Package ‘GCDkit’

August 16, 2019

Version 6.0
Date 2019-08-16
Title Geochemical Data Toolkit for Windows
Author Vojtech Janousek <vojtech.janousek@geology.cz>
        Colin Farrow <colinfarrow537@gmail.com>
        Vojtech Erban <vojtech.erban@geology.cz>
        Jean-Francois Moyen <jfmoyen@gmail.com>
Maintainer Vojtech Janousek <vojtech.janousek@geology.cz>
Depends R (>= 3.6.0), tcltk
Imports stats, methods, utils, graphics, grDevices, MASS, grid, lattice, foreign, RODBC, R2HTML, sp, compiler
Suggests XML, rgdal, tkrplot, curl
Description A program for recalculation of geochemical data from igneous and metamorphic rocks. With complete graphical user interface (GUI) it runs under Windows Vista/7/8/10, complete functionality/stability under 2000/XP cannot be guaranteed. The current version should also work on Mac OS X (release 10.6 and above) and various distributions of Linux (Debian, RedHat, SUSE, Ubuntu).
License GPL (>= 2)
URL http://www.gcdkit.org

R topics documented:

.claslist ...................................................... 6
about .......................................................... 6
accessVar ...................................................... 7
Add contours ................................................... 8
addResults ...................................................... 9
addResultsIso .................................................. 9
AFM ............................................................ 10
ageEps ........................................................ 12
Agrawal ........................................................ 14
Ague ............................................................. 16
appendSingle .................................................. 18
apSaturation ................................................... 19
R topics documented:

ArcMapSetup .................................................. 21
assign1col ................................................... 22
assign1symb .................................................. 23
assignCoILab .................................................. 24
assignCoIVar .................................................. 25
assignSymbGroup ............................................ 26
assignSymbLab ................................................ 27
assignSymbLett ............................................... 28
atacazo ......................................................... 29
Batchelor ....................................................... 30
binary .......................................................... 32
binaryBoxplot ............................................... 33
blatna .......................................................... 35
Boolean conditions ......................................... 35
bpplot2 ......................................................... 37
Cabanis ......................................................... 38
calc ............................................................ 40
calcAnomaly ................................................... 41
calcCore ......................................................... 43
Catanorm ........................................................ 44
CIPW ............................................................. 45
classify ........................................................ 47
clr.transform ................................................... 49
cluster .......................................................... 51
contourGroups ............................................... 52
coplotByGroup ............................................... 54
coplotTri ........................................................ 56
correlationCoefPlot ......................................... 58
Cox ............................................................... 59
crosstab ........................................................ 62
customScript .................................................. 63
cutMy ............................................................ 64
Debon ............................................................ 65
DebonBMgNo .................................................... 67
DebonBQF ........................................................ 69
DebonKNaB ..................................................... 71
DebonQB .......................................................... 73
deleteSingle ................................................... 74
EarthChem ....................................................... 75
Edit labels ...................................................... 77
Edit numeric data ............................................ 78
editLabFactor .................................................. 78
elemIso .......................................................... 79
epsEps ............................................................ 81
Export to Access ............................................. 82
Export to DBF .................................................. 83
Export to Excel ............................................... 84
Export to HTML tables ..................................... 85
F-M-W diagram ............................................... 87
FeMiddlemost .................................................. 89
figAdd ........................................................... 90
figaro.identify ................................................. 94
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>figCol</td>
<td>95</td>
</tr>
<tr>
<td>figEdit</td>
<td>96</td>
</tr>
<tr>
<td>figGbo</td>
<td>97</td>
</tr>
<tr>
<td>figLoad</td>
<td>98</td>
</tr>
<tr>
<td>figMulti</td>
<td>98</td>
</tr>
<tr>
<td>figOverplot</td>
<td>101</td>
</tr>
<tr>
<td>figOverplotDiagram</td>
<td>103</td>
</tr>
<tr>
<td>figRedraw</td>
<td>105</td>
</tr>
<tr>
<td>figSave</td>
<td>106</td>
</tr>
<tr>
<td>figScale</td>
<td>107</td>
</tr>
<tr>
<td>figUser</td>
<td>108</td>
</tr>
<tr>
<td>figZoom</td>
<td>109</td>
</tr>
<tr>
<td>filledContourFig</td>
<td>110</td>
</tr>
<tr>
<td>Frost</td>
<td>111</td>
</tr>
<tr>
<td>Frost_2008</td>
<td>114</td>
</tr>
<tr>
<td>gcdOptions</td>
<td>116</td>
</tr>
<tr>
<td>graphicsOff</td>
<td>119</td>
</tr>
<tr>
<td>Grebennikov</td>
<td>120</td>
</tr>
<tr>
<td>groupsByCluster</td>
<td>121</td>
</tr>
<tr>
<td>groupsByDiagram</td>
<td>122</td>
</tr>
<tr>
<td>groupsByLabel</td>
<td>123</td>
</tr>
<tr>
<td>Harris</td>
<td>124</td>
</tr>
<tr>
<td>Hastie</td>
<td>125</td>
</tr>
<tr>
<td>Hollucher</td>
<td>127</td>
</tr>
<tr>
<td>ID</td>
<td>130</td>
</tr>
<tr>
<td>info</td>
<td>131</td>
</tr>
<tr>
<td>isochron</td>
<td>131</td>
</tr>
<tr>
<td>isocon</td>
<td>133</td>
</tr>
<tr>
<td>Jensen</td>
<td>136</td>
</tr>
<tr>
<td>joinGroups</td>
<td>137</td>
</tr>
<tr>
<td>Jung</td>
<td>138</td>
</tr>
<tr>
<td>Laroche</td>
<td>140</td>
</tr>
<tr>
<td>LaRocheCalc</td>
<td>143</td>
</tr>
<tr>
<td>LaurentSource</td>
<td>144</td>
</tr>
<tr>
<td>loadData</td>
<td>146</td>
</tr>
<tr>
<td>Maniar</td>
<td>149</td>
</tr>
<tr>
<td>mergeData</td>
<td>152</td>
</tr>
<tr>
<td>Meschede</td>
<td>153</td>
</tr>
<tr>
<td>Mesonorm</td>
<td>154</td>
</tr>
<tr>
<td>Middlemost</td>
<td>155</td>
</tr>
<tr>
<td>millications</td>
<td>157</td>
</tr>
<tr>
<td>mzs2deg</td>
<td>158</td>
</tr>
<tr>
<td>Misc</td>
<td>159</td>
</tr>
<tr>
<td>Miyashiro</td>
<td>160</td>
</tr>
<tr>
<td>Mode</td>
<td>161</td>
</tr>
<tr>
<td>Molecular weights</td>
<td>163</td>
</tr>
<tr>
<td>Mullen</td>
<td>164</td>
</tr>
<tr>
<td>MullerK</td>
<td>166</td>
</tr>
<tr>
<td>Multiple plots</td>
<td>169</td>
</tr>
<tr>
<td>mzSaturation</td>
<td>171</td>
</tr>
<tr>
<td>NaAIK</td>
<td>172</td>
</tr>
<tr>
<td>Niggli</td>
<td>174</td>
</tr>
<tr>
<td>R topics documented:</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>OConnor</td>
<td>175</td>
</tr>
<tr>
<td>overplotDataset</td>
<td>177</td>
</tr>
<tr>
<td>oxide2oxide</td>
<td>180</td>
</tr>
<tr>
<td>oxide2ppm</td>
<td>181</td>
</tr>
<tr>
<td>pairsCorr</td>
<td>181</td>
</tr>
<tr>
<td>Paulick</td>
<td>183</td>
</tr>
<tr>
<td>pdfAll</td>
<td>185</td>
</tr>
<tr>
<td>Pearce and Cann</td>
<td>186</td>
</tr>
<tr>
<td>Pearce and Norry</td>
<td>187</td>
</tr>
<tr>
<td>Pearce Nb-Th-Yb</td>
<td>189</td>
</tr>
<tr>
<td>Pearce Nb-Ti-Yb</td>
<td>191</td>
</tr>
<tr>
<td>Pearce1982</td>
<td>193</td>
</tr>
<tr>
<td>Pearce1996</td>
<td>194</td>
</tr>
<tr>
<td>PearceDestructive1</td>
<td>196</td>
</tr>
<tr>
<td>PearceDestructive2</td>
<td>198</td>
</tr>
<tr>
<td>PearceDestructive3</td>
<td>199</td>
</tr>
<tr>
<td>PearceEtAl</td>
<td>201</td>
</tr>
<tr>
<td>PearceGranite</td>
<td>202</td>
</tr>
<tr>
<td>PeceTaylor</td>
<td>204</td>
</tr>
<tr>
<td>peekDataset</td>
<td>206</td>
</tr>
<tr>
<td>peterplot</td>
<td>207</td>
</tr>
<tr>
<td>phasePropPlot</td>
<td>209</td>
</tr>
<tr>
<td>Plate</td>
<td>211</td>
</tr>
<tr>
<td>Plate editing</td>
<td>213</td>
</tr>
<tr>
<td>plateAddReservoirs</td>
<td>215</td>
</tr>
<tr>
<td>plateLabelSlots</td>
<td>217</td>
</tr>
<tr>
<td>plotPlate</td>
<td>219</td>
</tr>
<tr>
<td>plotWithCircles</td>
<td>220</td>
</tr>
<tr>
<td>pokeDataset</td>
<td>222</td>
</tr>
<tr>
<td>ppm2oxide</td>
<td>223</td>
</tr>
<tr>
<td>prComp</td>
<td>224</td>
</tr>
<tr>
<td>printSamples</td>
<td>225</td>
</tr>
<tr>
<td>printSingle</td>
<td>226</td>
</tr>
<tr>
<td>profiler</td>
<td>227</td>
</tr>
<tr>
<td>projbiplot</td>
<td>230</td>
</tr>
<tr>
<td>psAll</td>
<td>232</td>
</tr>
<tr>
<td>purgeDatasets</td>
<td>233</td>
</tr>
<tr>
<td>QANOR</td>
<td>233</td>
</tr>
<tr>
<td>QAPF</td>
<td>235</td>
</tr>
<tr>
<td>quitGCDkit</td>
<td>238</td>
</tr>
<tr>
<td>r2clipboard</td>
<td>238</td>
</tr>
<tr>
<td>recast</td>
<td>239</td>
</tr>
<tr>
<td>reciprocalIso</td>
<td>240</td>
</tr>
<tr>
<td>Regular expressions</td>
<td>241</td>
</tr>
<tr>
<td>Ross</td>
<td>243</td>
</tr>
<tr>
<td>rtSaturation</td>
<td>245</td>
</tr>
<tr>
<td>saveData</td>
<td>247</td>
</tr>
<tr>
<td>saveResults</td>
<td>247</td>
</tr>
<tr>
<td>saveResultsIso</td>
<td>248</td>
</tr>
<tr>
<td>sazava</td>
<td>249</td>
</tr>
<tr>
<td>Schandl</td>
<td>250</td>
</tr>
<tr>
<td>selectAll</td>
<td>252</td>
</tr>
<tr>
<td>R topics documented:</td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---</td>
</tr>
<tr>
<td>selectByDiagram</td>
<td>253</td>
</tr>
<tr>
<td>selectByLabel</td>
<td>254</td>
</tr>
<tr>
<td>selectColumnLabel</td>
<td>255</td>
</tr>
<tr>
<td>selectColumnsLabels</td>
<td>256</td>
</tr>
<tr>
<td>selectNorm</td>
<td>258</td>
</tr>
<tr>
<td>selectPalette</td>
<td>260</td>
</tr>
<tr>
<td>selectSubset</td>
<td>262</td>
</tr>
<tr>
<td>setCex</td>
<td>264</td>
</tr>
<tr>
<td>setSilent</td>
<td>265</td>
</tr>
<tr>
<td>setTransparency</td>
<td>266</td>
</tr>
<tr>
<td>Shand</td>
<td>267</td>
</tr>
<tr>
<td>Shervais</td>
<td>269</td>
</tr>
<tr>
<td>showColours</td>
<td>270</td>
</tr>
<tr>
<td>showLegend</td>
<td>271</td>
</tr>
<tr>
<td>showSymbols</td>
<td>272</td>
</tr>
<tr>
<td>spider</td>
<td>273</td>
</tr>
<tr>
<td>spider2norm</td>
<td>279</td>
</tr>
<tr>
<td>spiderBoxplot</td>
<td>281</td>
</tr>
<tr>
<td>spiderByGroupFields</td>
<td>283</td>
</tr>
<tr>
<td>spiderByGroupPatterns</td>
<td>284</td>
</tr>
<tr>
<td>srnd</td>
<td>285</td>
</tr>
<tr>
<td>statsByGroup</td>
<td>287</td>
</tr>
<tr>
<td>statsByGroupPlot</td>
<td>288</td>
</tr>
<tr>
<td>statsIso</td>
<td>288</td>
</tr>
<tr>
<td>strip</td>
<td>291</td>
</tr>
<tr>
<td>stripBoxplot</td>
<td>292</td>
</tr>
<tr>
<td>Subset by range</td>
<td>294</td>
</tr>
<tr>
<td>summaryAll</td>
<td>294</td>
</tr>
<tr>
<td>summaryByGroup</td>
<td>296</td>
</tr>
<tr>
<td>summarySingle</td>
<td>297</td>
</tr>
<tr>
<td>summarySingleByGroup</td>
<td>299</td>
</tr>
<tr>
<td>Sylvester</td>
<td>300</td>
</tr>
<tr>
<td>TAS</td>
<td>301</td>
</tr>
<tr>
<td>TASMiddlemost</td>
<td>304</td>
</tr>
<tr>
<td>ternary</td>
<td>307</td>
</tr>
<tr>
<td>tetrad</td>
<td>310</td>
</tr>
<tr>
<td>threeD</td>
<td>311</td>
</tr>
<tr>
<td>tkSelectVariable</td>
<td>313</td>
</tr>
<tr>
<td>tk_winDialog</td>
<td>314</td>
</tr>
<tr>
<td>tk_winDialogString</td>
<td>314</td>
</tr>
<tr>
<td>trendTicks</td>
<td>315</td>
</tr>
<tr>
<td>Verma</td>
<td>317</td>
</tr>
<tr>
<td>Villaseca</td>
<td>319</td>
</tr>
<tr>
<td>Wedge</td>
<td>321</td>
</tr>
<tr>
<td>Whalen</td>
<td>324</td>
</tr>
<tr>
<td>WinFloyd1</td>
<td>326</td>
</tr>
<tr>
<td>WinFloyd2</td>
<td>327</td>
</tr>
<tr>
<td>Wood</td>
<td>329</td>
</tr>
<tr>
<td>YbN vs. LaN/YbN</td>
<td>331</td>
</tr>
<tr>
<td>zrSaturation</td>
<td>332</td>
</tr>
</tbody>
</table>

Index 335
.claslist  List of available classification schemes

Description
The function returns a list of classification diagrams available in the system.

Usage
.claslist()

Value
A matrix with two columns:

<table>
<thead>
<tr>
<th>menu</th>
<th>menu items</th>
</tr>
</thead>
<tbody>
<tr>
<td>function</td>
<td>the attached functions</td>
</tr>
</tbody>
</table>

Author(s)
Vojtech Erban, <erban@sopky.cz>

about  About GCDkit

Description
Prints short information about the current version of GCDkit and contact addresses of its authors.

Usage
about()

Arguments
None.

Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>
accessVar

Accessing data in memory of R

Description

Loads data already present in memory of R into GCDkit.

Usage

accessVar(var=NULL,GUI=FALSE)

Arguments

var 
a text string specifying the variable to be accessed

GUI 
logical; is the function called from GUI (or from the command line)?

Details

This function makes possible to access a variable, already present in R, most importantly the sample data sets. Firstly these need to be made available using the command data.

Value

WR 
numeric matrix: all numeric data

labels 
data frame: all at least partly character fields; labels$Symbol contains plotting symbols and labels$Colour the plotting colours

The function prints a short summary about the attached data. It also loads and executes the Plugins, i.e. all the R code that is currently stored in the subdirectory '\Plugin'.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

Examples

data(swiss)
accessVar("swiss")
binary("Catholic","Education")

data(sazava)
accessVar("sazava")
binary("SiO2","Ba")
Add contours

Description

Superposes contour lines to a Figaro-compatible plot.

Usage

```
addContours(GUI = FALSE, bandwidth = "auto", ...)
```

Arguments

- **GUI**: logical; is the function called from GUI (or in a direct mode)?
- **bandwidth**: vector of bandwidths for x and y directions provided to the function `kde2d`. See Details.
- **...**: additional parameters passed to the underlying function `contour`. Typically plotting parameters.

Details

This is, in principle, a front end to the standard R function `contour`. It will work on both the stand-alone Figaro-compatible plot or a plate thereof. The bandwidth should be a positive number or ‘auto’, whereby the higher value corresponds to a smoother result. The necessary calculations are done by the function `kde2d`.

Value

None.

Author(s)

Vojtech Erban, <erban@sopky.cz> Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

`filled.contour`, `kde2d`, `par`, `figaro`

Examples

```
data(sazava)
accessVar("sazava")
plotDiagram("CoxPlut",FALSE,TRUE)
addContours(col="darkblue",lty="dashed",bandwidth=10)
addContours(col="darkgreen",lty="dotted",bandwidth=5)
multiple("SiO2","Al2O3,MgO,CaO,K2O")
plateCex(2)
plateCexLab(1.5)
addContours(col="darkgreen",lty="dashed")
```
**addResults**

Appending results to data

**Description**

Appends the most recently calculated results to the data stored in memory.

**Usage**

```r
addResults(what="results", save=TRUE, overwrite=TRUE, GUI=FALSE)
```

**Arguments**

- `what` character; the name of variable to be appended.
- `save` logical; Append to the data matrix `WR`?
- `overwrite` logical; overwrite any matching items in the matrix `WR`?
- `GUI` logical; Is the function called within the GUI environment?

**Details**

This function appends the variable `results` (a matrix or vector) returned by most of the calculation algorithms to the numeric data stored in the matrix `WR`.

In case that any items of the same name are already present in the matrix `WR`, the user is asked whether they should be overwritten (GUI). In batch mode, they can be overwritten silently if `overwrite=TRUE`.

**Value**

Modifies the matrix `WR`.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

---

**addResultsIso**

Append Sr-Nd isotopic data

**Description**

Appends the calculated isotopic parameters stored in the matrix `init` to the numeric data already in the system.

**Usage**

```r
addResultsIso()
```
Value

Modifies the numeric data matrix('WR') to which it appends the following columns:

- **Age (Ma)**: Age in Ma
- **$^{87}\text{Sr}/^{86}\text{Sr}$**: Initial $^{87}\text{Sr}/^{86}\text{Sr}$ ratios
- **$^{143}\text{Nd}/^{144}\text{Nd}$**: Initial $^{143}\text{Nd}/^{144}\text{Nd}$ ratios
- **EpsNd**: Initial $\epsilon(\text{Nd})$ values
- **TDM**: Single-stage depleted-mantle Nd model ages (Liew & Hofmann, 1988)
- **TDM.Gold**: Single-stage depleted-mantle Nd model ages (Goldstein et al., 1988)
- **TDM.2stg**: Two-stage depleted-mantle Nd model ages (Liew & Hofmann, 1988)

Plugin

SrNd.r

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

`'addResults'`

Description

Assigns data for AFM ternary diagram into Figaro template (list `sheet`) and appropriate values into `x.data` and `y.data`.

Usage

`AFM(equ=FALSE)`

Arguments

equ Logical: Should the template use boundary defined by equation?
Details

The AFM diagram is a triangular plot with apices A, F and M defined as follows:

\[ A = (K_2O + Na_2O) \text{ wt. %} \]
\[ F = \text{FeOtot wt. %} \]
\[ M = \text{MgO wt. %} \]
\[ A + F + M = 100 \% \]

The classification diagram divides data into 'tholeiite series' and 'calc-alkaline series' as proposed by Irvine & Baragar (1971). For extreme values linear extrapolation of boundary curve is employed.

![AFM Diagram](image)

Value

<table>
<thead>
<tr>
<th>sheet</th>
<th>list with Figaro Style Sheet data</th>
</tr>
</thead>
<tbody>
<tr>
<td>x.data, y.data</td>
<td>A, F, M values (see details) transformed into 2D</td>
</tr>
</tbody>
</table>

Author(s)

Vojtech Erban, <erban@sopky.cz>
& Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

classify figaro plot Diagram
Examples

#Within GCDkit, AFM is called using following auxiliary functions:
#To Classify data stored in WR (Groups by diagram)
classify("AFM")

#To plot data stored in WR or its subset (menu Classification)
plotDiagram("AFM", FALSE)

---

**ageEps**

*Plot Sr or Nd growth lines*

Description

Plots Nd or Sr growth curves in the binary diagram age-$\epsilon(Nd)$ or age-Sr isotopic ratio.
ageEps

Usage

ageEps(GUI=FALSE, . . .)

ageEps2(GUI=FALSE, . . .)

ageSr(GUI=FALSE, . . .)

Arguments

GUI logical; is the function called from the GUI?

. . . optional parameters to the underlying function {plotWithLimits}

Details

The Nd growth curves in individual samples can be plotted using either a single- or two-stage (Liew & Hofmann 1988) models.

In case of Nd are shown growth curves for the two main mantle reservoirs, CHUR and Depleted Mantle (DM) (the latter in two modifications, after Goldstein et al. (1988) and Liew & Hofmann (1988)).

For Sr only uniform reservoir (UR) development is calculated using parameters of Faure (1986 and references therein).

The small ticks, or rugs, on x axis correspond to Nd model ages, on y axis to initial $\epsilon(Nd)$ values.

This function is Figaro compatible.

Value

None.

Plugin

SrNd.r

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

The actual plotting is done by the function plotWithLimits.
Agrawal | Trace-element based discrimination plots for (ultra-)basic rocks (Agrawal et al. 2008)

Description

Plots data stored in ‘WR’ into discrimination plots proposed by Agrawal et al. (2008) for (ultra-)basic rocks ($SiO_2 < 52$ wt. %).

Usage

Agrawal(plot.txt = getOption("gcd.plot.text"), GUI=FALSE)

Arguments

- plot.txt: logical, annotate fields by their names?
- GUI: logical, is the function called from a GUI?

Details

Suite of five diagrams for discrimination of geotectonic environment of ultrabasic and basic rocks, proposed by Agrawal et al. (2008). It is based on linear discriminant analysis applied to log-transformed concentration ratios of five trace elements (La, Sm, Yb, Nb, and Th), i.e., using four ratios $ln(La/Th)$, $ln(Sm/Th)$, $ln(Yb/Th)$, and $ln(Nb/Th)$. The two discriminant functions, DF1 and DF2, are mathematically designed to maximize the separation between the groups and account for 100 percent of the variance in the data.

Note that only samples with $SiO_2 < 52$ wt. % are plotted.

Also note that each diagram applies only to environments explicitly mentioned. Samples from the environment not taken into account will be misinterpreted (the CRB + OIB + MORB diagram is not designed for IAB etc.) See the Agrawal et al (2008) for further details.

Following geotectonic settings may be deduced:

<table>
<thead>
<tr>
<th>Abbreviation used</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAB</td>
<td>island arc basic rocks</td>
</tr>
<tr>
<td>CRB</td>
<td>continental rift basic rocks</td>
</tr>
<tr>
<td>OIB</td>
<td>ocean-island basic rocks</td>
</tr>
<tr>
<td>MORB</td>
<td>mid-ocean ridge basic rocks</td>
</tr>
</tbody>
</table>
Value

None.

Note

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots.

See Plate, Plate editing and figaro for details.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

Verma, Plate, Plate editing, plotPlate, figaro

Examples

# plot the diagrams
plotPlate("Agrawal")
Description

Implementation of Concentration ratio diagrams after Ague (1994) used for judging the mobility of elements or oxides in course of various geochemically open-system processes such as alteration or partial melting.

Usage

Ague(x = NULL, 
whichelems = "SiO2,TiO2,Al2O3,FeOt,MnO,MgO,CaO,Na2O,K2O,P2O5", 
immobile = NULL, bars = NULL, plot = TRUE)

Arguments

x two sample names for analyses of the protolith and altered rock compositions, respectively.

whichelems list of elements to be plotted.

immobile list of (one or more) elements considered as immobile.

bars optional name of the variable containing 1σ errors for plotting error bars.

plot logical, should be the diagram plotted or just the results calculated?

Details

The Concentration ratio diagram shows concentration ratio of each geochemical species of interest (element or oxide) in the 'altered rock' to that in its presumed 'protolith'. These ratios are plotted on the y-axis, and the elements are arranged in any convenient order along x.

Following an open-system geological process, any of the perfectly immobile constituents i should ideally have exactly the same concentration ratio \( r_{inv} \) defined as (Ague 2003):

\[
r_{inv} = \frac{c_A^i}{c_0^i}
\]

where \( c_i \) is the concentration of the species \( i \), \( 0 \) refers to the 'protolith' and \( A \) to the 'altered rock'.

This ratio, however, would only exceptionally equate unity, when the mass of the whole system is conserved. Using the presumably immobile species \( i \) as the geochemical reference frame, the change in the rock mass can be defined as Ague (1994):

\[
\Delta_{Mass} = \frac{c_0^i}{c_A^i} - 1
\]

Thus \( r_{inv} > 1 \) indicates overall rock mass loss due to removal of mobile constituents; this has the effect of increasing the concentrations of the immobile species ('residual enrichment'). Conversely, \( r_{inv} < 1 \) shows an overall rock mass gain ('residual dilution').

The mass change of any mobile constituent \( j \) can be expressed as (Ague 1994):
\[ \Delta_j = \frac{1}{r_{inv}} \frac{c_j^A}{c_0^A} - 1 \]

Mobile species \( j \) that have \( \frac{c_j^A}{c_0^A} \) ratios greater than \( r_{inv} \) have been added to the system, and those with ratios lower than \( r_{inv} \) have been lost.

In the GCDkit’s implementation of the Concentration ratio diagrams, firstly the parental and altered rock samples can be chosen interactively from a binary plot \( MgO - SiO_2 \), if not specified at the function call. Then the user is prompted for the elements/oxides to be plotted.

If not provided as a comma delimited list among the arguments, the presumably immobile elements are to be specified. To facilitate this choice, printed and plotted as barplots are ordered ratios of the elemental concentrations in the ‘altered rock’ to that in the ‘protolith’ \( \left( \frac{c_j^A}{c_0^A} \right) \).

Finally the concentration ratio diagram is plotted. If the parameter bars is given, error bars are also shown corresponding to \(+/- 1 \sigma\).

Spider plot – normalized by sample Po–4

Value

Returns a matrix ‘results’ with the following columns:

Altered/Protolith

concentration ratios of the given geochemical species in the ‘altered rock’ to that in the ‘protolith’ - primary y axis of the plot
Gain/loss in %  relative gains (positive) or losses (negative) corrected for the rock mass change
- secondary y axis of the plot

Plugin
Isocon.r

Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>

References

See Also
Wedge, isocon

Examples
```r
data<-loadData("sazava.data",sep="\t")
Ague(c("Po-4","Po-1"),
    "SiO2,TiO2,Al2O3,FeOt,MgO,CaO,Rb,Sr,Ba,Zr,La,Nd,Eu,Yb,Y",
    "TiO2,FeOt")
```

---

**appendSingle**

*Append empty label or variable*

**Description**

Appends an empty numeric data column or a new label to the current data set.

**Usage**

```r
appendSingle()
```

**Value**

Returns the corrected version of the data frame 'labels' or numeric matrix 'WR'.

apSaturation

Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>

Description
Calculates apatite saturation temperatures for observed whole-rock major-element compositions. Prints also phosphorus saturation levels for the given major-element compositions and assumed magma temperature.

Usage
apSaturation(Si = WR[, "SiO2"], ACNK = WR[, "A/CNK"], P2O5 = WR[, "P2O5"], T = 0)

Arguments
- Si: SiO2 contents in the melt (wt. %)
- ACNK: vector with A/CNK (mol %) values
- P2O5: vector with P2O5 concentrations
- T: assumed magma temperature in °C

Details
* Calculates phosphorus saturation levels following Harrison & Watson (1984):

\[ \ln(D_P) = \frac{8400 + 26400(SiO_2 - 0.5)}{T} - 3.1 - 12.4(SiO_2 - 0.5) \]

\[ P_{2O5,HW} = \frac{42}{D_P} \]

where 'T' = absolute temperature (K), 'D_P' = distribution coefficient for phosphorus between apatite and melt and 'SiO2' is the weight fraction of silica in the melt, SiO2 wt. %/100.

These formulae were shown to be valid only for metaluminous rocks, i.e. A/CNK < 1, and were modified for peraluminous rocks (A/CNK > 1) by Bea et al. (1992):

\[ P_{2O5,Bea} = P_{2O5,HW} e^{\frac{6428(A/CNK-1)}{(T-293.15)}} \]

and Pichavant et al. (1992):

\[ P_{2O5, PV} = P_{2O5, HW} + (A/CNK - 1)e^{\frac{-5800}{T}} - 3.22SiO_2 + 9.31 \]

Note that the phosphorus saturation concentrations are not returned by the function but printed only.

* Calculates saturation temperatures in °C using the observed P2O5 concentrations (Harrison & Watson, 1984):
\[ T_{HW} = \frac{8400 + 26400(SiO_2 - 0.5)}{ln(\frac{42}{P_2O_5})} + 3.1 + 12.4(SiO_2 - 0.5) - 273.15 \]

for peraluminous rocks \((A/CNK > 1)\) the equation of Bea et al. (1992) needs to be solved for \(T\) (in K) by iterations:

\[ P_2O_5,Bea = \frac{42}{e^{8400+26400(SiO_2-0.5)}-3.1-12.4(SiO_2-0.5)} e^{\frac{6429(A/CNK-1)}{T-273.15}} \]

as is that of Pichavant et al. (1992):

\[ P_2O_5, PV = \frac{42}{e^{8400+26400(SiO_2-0.5)}-3.1-12.4(SiO_2-0.5)} + (A/CNK-1)e^{-5900(T-3.22SiO_2+9.31)} \]

Value

Returns a matrix 'results' with the following columns:

- A/CNK
- A/CNK values
- Tap. sat. C. HW
- saturation T of Harrison & Watson (1984) in °C
- Tap. sat. C. Bea
- saturation T of Bea et al. (1992) in °C, peraluminous rocks only
- Tap. sat. C. Pich
- saturation T of Pichavant et al. (1992) in °C, peraluminous rocks only

Plugin

Saturation.r

Author(s)

Vojtěch Janoušek, <vojtech.janousek@geology.cz>

References


Description

This function provides a rudimentary support for drawing Arc GIS-compatible shape files (.shp).

Usage

\[
\text{ArcMapSetup}(\text{object}, \text{layers = NULL, map.col = NULL, map.palett = } \text{"heat.colours"}, \text{labels.txt = FALSE, col.txt = } \text{"black"}, \text{cex.txt = 0.5, axes = TRUE, longlat = TRUE, xlab = } \text{"Longitude"}, \text{ylab = } \text{"Latitude"})
\]

Arguments

- **object**: name of the object to be drawn, normally GCDmap.
- **layers**: names of layers to be drawn.
- **map.col**: a vector with colors specified for each of the polygons.
- **map.palett**: name of a palette to fill the individual polygons by a random colour.
- **labels.txt**: logical; label the individual polygons?
- **col.txt**: colour of these textual labels.
- **cex.txt**: relative size of these textual labels.
- **axes**: logical; should be the axes drawn?
- **longlat**: logical; should be long-lat grid added?
- **xlab**: label for the x axis.
- **ylab**: label for the y axis.

Details

By default, the \texttt{loadData} function of the \texttt{GCDkit} system loads a shape (*.shp) file into a list object called \texttt{GCDmap}. Each layer represents one item.

If required, the longitude-latitude grid is also drawn using the function \texttt{llgridlines}.

Value

None. It just modifies properties of a Figaro object (a map).

Author(s)

Vojtech Janousek, \texttt{<vojtech.janousek@geology.cz>}

This code relies heavily on rgdal and sp packages that were written by Roger Bivand, Edzer Pebesma and their co-workers.

References

None.
Examples

# Example of a public-domain World map
shp.file<="world_country_admin_boundary_shapefile_with_fips_codes.shp"
setwd(earthchem.dir)
loadData(shp.file)
figRedraw()

ArcMapSetup(GCDmap,map.palette="heat.colors",labels.txt=TRUE,col.txt="darkblue",cex.txt=0.8,axes=TRUE,longlat=FALSE)
figRedraw()

# Scaling (not precise clipping, as it needs to preserve the aspect ratio)
figXlim(c(-77,-50))
figYlim(c(0,30))

# Other Figaro functions should be finally working, too
figMain("Caribbean and adjacent South America")
figColMain("darkred")

---

**assign1col**

Uniform colours

**Description**

Assigns the same plotting colour to all samples.

**Usage**

```r
assign1col(col=-1)
```

**Arguments**

- `col` numeric; code of the colour.

**Details**

This function sets the same colour to all of the plotting symbols. If `col` = -1 (the default), the user is prompted to specify its code.

**Value**

Sets `labels$Colour` to code of the selected plotting colour.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>
assign1symb

See Also
To display the current legend use `showLegend`. Symbols and colours by a single label can be assigned by `assignSymbLab` and `assignColLab` respectively, symbols and colours by groups simultaneously by `assignSymbGroup`. Uniform symbols are obtained by `assign1symb`. Table of available plotting symbols is displayed by `showSymbols` and colours by `showColours`.

<table>
<thead>
<tr>
<th>assign1symb</th>
<th>Uniform symbols</th>
</tr>
</thead>
</table>

**Description**
Assigns the same plotting symbol to all samples.

**Usage**
```r
assign1symb(pch=-1)
```

**Arguments**
- `pch` numeric; code of the plotting symbol.

**Details**
This function sets the same plotting symbol to all the data points. If `pch` = -1 (the default), the user is prompted to specify its code.

**Value**
Sets `labels$Symbol` to code of the selected plotting symbol.

**Author(s)**
Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**
To display the current legend use `showLegend`. Symbols and colours by a single label can be assigned by `assignSymbLab` and `assignColLab` respectively, symbols and colours by groups simultaneously by `assignSymbGroup`. Uniform colours are obtained by `assign1col`. Table of available plotting symbols is displayed by `showSymbols` and colours by `showColours`. 
**assignColLab**

### Description

Assigns plotting colours according to the levels of the chosen label or, alternatively, sample names.

### Usage

```r
assignColLab(lab = NULL, pal = NULL, colours = NULL, display.legend = FALSE)
```

### Arguments

- **lab**: specification of the variable to be used for colours assignment. See Details.
- **pal**: character; name of the palette to be used when no colours are specified directly. Batch mode only.
- **colours**: a vector with codes of colours to be assigned. Batch mode only.
- **display.legend**: logical; should be the legend displayed? Batch mode only.

### Details

If called from in interactive mode (from GUI), the variable (sample names or label) can be selected using the function `selectColumnLabel`.

In batch mode, `lab` can be an integer (1 for sample names, or a sequence number of the column in the `labels` plus 1). Alternatively, it can contain the full name of a column in `labels`. See examples.

If in batch mode, either `colours` or `palette` have to be specified for the correct colour assignment.

### Value

Sets `leg.col` to a sequence number of column in `labels` that is to be used to build the legend or -1 if sample numbers are to be used; `labels$Colour` contains the codes of the desired plotting colours.

### Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

### See Also

- To display the current legend use `showLegend`
- Symbols by a single label can be assigned by `assignSymbLab`
- Uniform colours and symbols are obtained by `assign1symb` and `assign1col`
- Table of available plotting symbols is displayed by `showSymbols` and colours by `showColours`
- Selecting a label: `selectColumnLabel`
- Selecting a palette: `selectPalette`
assignColVar

Examples

data(sazava)
accessVar("sazava")

assignColLab()  # Interactive mode

# Sample names, standard GCDkit colours palette
assignColLab(1,colours=palette.gcdkit,display.legend=TRUE)

# Standard palettes
assignColLab(3,pal="jet.colours",display.legend=TRUE)  # Second column in labels
assignColLab("Locality",pal="jet.colours",display.legend=TRUE)  # Ditto (here Locality)

# User-defined palette
my.palette<-colorRampPalette(c("black", "darkgreen", "red"),space = "rgb")
assignColLab("Locality",pal="my.palette",display.legend=TRUE)

assignColVar  Colours by a variable

Description
Assigns plotting colours according to the values of the variable.

Usage
assignColVar(what=NULL,pal="heat.colours",save=TRUE,n=15,quant=0,eq.classes=FALSE,alt.leg=FALSE)

Arguments
what variable name or a formula; if NULL a dialogue is displayed
pal character; name of a palette
save logical; should the newly picked colours be assigned to 'labels'?
n desired approximate number of colours to be assigned.
quant numeric, 0-50; quantile to be potentially used to get rid of outliers. See details.
eq.classes logical; should classes contain equal number of values?
alt.leg logical; should the alternative (continuous) legend shown? See Examples.

Details
For selection of the variable is employed the function 'selectColumnLabel'. The user can specify either existing data column in the 'WR' or a formula. The colours can be optionally (default behaviour) assigned globally, so that all the plots will use these from this point on. If not specified upon function call, the palette is picked using selectPalette. The possible values are: 'grays', 'reds', 'blues', 'greens', 'cyans', 'violets', 'yellows', 'cm.colors', 'heat.colors', 'terrain.colors', 'topo.colors', 'rainbow' and 'jet.colors'.

Also, user-defined palette functions are supported now. See Examples.
The analyses with no data available for the colours assignment will remain black.
If quant differs from the default value of zero, the data are trimmed to an interval \((\text{quant}, 100-\text{quant})\)-th quantile of the dataset and all values out of it plotted in gray.

Setting `eq.classes=TRUE` allows to have classes with equal number of values (as opposed to equal intervals). This option is best suited for very skewed datasets (lots of points with similar values, some outliers).

**Value**

A list of two components, `col` and `leg`. The former are the plotting colours, the latter contains information needed to build a legend. If `save = TRUE`, `labels$Colour` will acquire the codes of desired plotting colours.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>
Jean-Francois Moyen, <jfmoyen@gmail.com>

**See Also**

*quantile* Colours by a single variable can be assigned by `assignColLab`, symbols and colours by groups simultaneously by `assignSymbGroup`. Uniform colours are obtained by `assign1col`. Table of available plotting colours is obtained by `showColours`.

**Examples**

```r
assignColVar("Na2O/K2O","greens")
plotDiagram("PeceTaylor",FALSE,TRUE)

my.palette=colorRampPalette(c("black","darkgreen","red"),space="rgb")
assignColVar("SiO2","my.palette")
plotDiagram("PeceTaylor",FALSE,TRUE)

assignColVar("SiO2","my.palette",n=7,quant=5)
plotDiagram("PeceTaylor",FALSE,TRUE)
showLegend()
showLegend(alt.leg=TRUE)
```

---

**Description**

Lets the user to assign plotting symbols and colours according to the levels of the defined groups.

**Usage**

```r
assignSymbGroup()
```

**Arguments**

None.
assignSymbLab

Value

Sets 'leg.col' and 'leg.pch' to zero, 'labels$Symbol' contains the codes of desired plotting symbols, 'labels$Colour' of plotting colours.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

To display the current legend use showLegend. Symbols by a single label can be assigned by assignSymbLab, colours using assignColLab. Uniform colours and symbols are obtained by assign1symb and assign1col. Table of available plotting symbols is displayed by showSymbols and colours by showColours.

<table>
<thead>
<tr>
<th>assignSymbLab</th>
<th>Symbols by label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Assigns plotting symbols according to the levels of the chosen label or, alternatively, sample names.</td>
</tr>
<tr>
<td>Usage</td>
<td>assignSymbLab(lab = NULL, symbols = NULL, display.legend = FALSE)</td>
</tr>
<tr>
<td>Arguments</td>
<td>lab: specification of the variable to be used for symbols assignment. See Details.</td>
</tr>
<tr>
<td></td>
<td>symbols: a vector with codes of plotting symbols to be assigned. Batch mode only.</td>
</tr>
<tr>
<td></td>
<td>display.legend: logical; should be the legend displayed? Batch mode only.</td>
</tr>
<tr>
<td>Details</td>
<td>If called from in interactive mode (from GUI), the variable (sample names or label) can be selected using the function 'selectColumnLabel'.</td>
</tr>
<tr>
<td></td>
<td>In batch mode, 'lab' can be an integer (1 for sample names, or a sequence number of the column in the 'labels' plus 1). Alternatively, it can contain the full name of a column in 'labels'. See examples.</td>
</tr>
<tr>
<td></td>
<td>If in batch mode, 'symbols' have to be specified for the correct plotting symbols assignment.</td>
</tr>
<tr>
<td>Value</td>
<td>Sets 'leg.pch' to a sequence number of column in 'labels' that is to be used to build the legend or -1 if sample numbers are to be used; 'labels$Symbol' contains the codes for desired plotting symbols.</td>
</tr>
<tr>
<td>Author(s)</td>
<td>Vojtech Janousek, <a href="mailto:vojtech.janousek@geology.cz">vojtech.janousek@geology.cz</a></td>
</tr>
</tbody>
</table>
See Also
To display the current legend use showLegend.

Using the function assignSymbLett, initial letters of the respective levels of the chosen label can be assigned to the plotting symbols.

Colours by a single label can be assigned by assignColLab, symbols and colours by groups simultaneously by assignSymbGroup. Uniform colours and symbols are obtained by assign1symb and assign1col. Table of available plotting symbols is displayed by showSymbols and colours by showColours.

Selecting a label: selectColumnLabel.

Examples

data(sazava)
accessVar("sazava")

assignSymbLab()  # Interactive mode

# Sample names, standard GCDkit colours palette
assignSymbLab(1,symbols=1:nrow(WR),display.legend=TRUE)

assignSymbLab(2,symbols=c("+","*","@"),display.legend=TRUE)  # First column in labels
assignSymbLab("Intrusion",symbols=c(12,15,17),display.legend=TRUE)  # Ditto (here Intrusion)

assignSymbLett  

Symbols by label - initial letters

Description
Assigns plotting symbols to initial letters of the respective levels of the chosen label.

Usage
assignSymbLett(lab = NULL, display.legend = FALSE)

Arguments
lab specification of the variable to be used for symbols assignment. See Details.
display.legend logical; should be the legend displayed? Batch mode only.

Details
If called from in interactive mode (from GUI), the variable (sample names or label) can be selected using the function 'selectColumnLabel'.

In batch mode, 'lab' can be an integer (a sequence number of the column in the 'labels'). Alternatively, it can contain the full name of a column in 'labels'. See examples.

Value
Sets 'leg.pch' to a sequence number of column in 'labels' that is to be used to build the legend; 'labels$Symbol' contains the plotting symbols, which correspond to initial letters for the levels of the specified label.
atacazo

Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also
To display the current legend use `showLegend`. Symbols by a single label can be assigned by `assignSymbLab`, colours by `assignColLab`, symbols and colours by groups simultaneously by `assignSymbGroup`. Uniform colours or symbols are achieved by `assign1symb` and `assign1col`. Table of available plotting symbols is displayed by `showSymbols` and colours by `showColours`.

Examples
data(sazava)
accessVar("sazava")
assignSymbLett() # Interactive mode
assignSymbLett(2,display.legend=TRUE) # Second column in labels
assignSymbLett("Locality",display.legend=TRUE) # The same (here Locality)

atacazo Whole-rock composition of lavas from the Atacazo and Ninahuilca volcanoes, Ecuador

Description
This data set gives the whole-rock major- and trace-element contents, together with Sr and Nd isotopic compositions of lavas from two volcanic complexes in Ecuador: the Atacazo and the Ninahuilca (Hidalgo, 2006; Hidalgo et al., 2008). This dataset is used in a worked example (chapter 25) of Janousek et al.’s book (2016).

Note that this data set contains information on symbols and colours to be used in `GCDkit`, as well as labels (Volcano) that can be used for grouping or similar purposes. It also includes 87Sr/86Sr and 143Nd/144Nd. Therefore, if the SrNd plugin for `GCDkit` is installed, these columns will automatically be recognized as Sr and Nd initial isotopic ratios when loading it into `GCDkit` (via `accessVar("atacazo")`, allowing variables such as TDM to be calculated and isotope-based diagrams to be plotted. As no Age column is supplied, the user will be prompted for the emplacement age; the volcanoes being Quaternary in age (220-71 ka for Atacazo and 71-2 ka for Ninahuilca), the age correction is insignificant and a small value (of 0.1 for instance) is adequate.

Usage
data(atacazo)

Format
A data frame containing 110 observations of 38 variables.

Source
data by Silvana Hidalgo, <shidalgo@igepn.edu.ec>,
formatted by Jean-François Moyen, <jfmoyen@gmail.com>
References


Examples

data(atacazo)
accessVar("atacazo")
binary("SiO2","Ba")
ageEps()
Value

- **sheet**
  - list with Figaro Style Sheet data
- **x.data**
  - \( R_1 = 4 \times \text{Si} - 11 \times (\text{Na} + \text{K}) - 2 \times (\text{Fe[total as bivalent]} + \text{Ti}) \), all in millications; as calculated by the function `LaRoche`
- **y.data**
  - \( R_2 = 6 \times \text{Ca} + 2 \times \text{Mg} + \text{Al} \), all in millications; as calculated by the function `LaRoche`

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

LaRoche figaro plotDiagram

Examples

- `#plot the diagram`
- `plotDiagram("Batchelor", FALSE)`
Description

These functions display data as a binary plot.

Usage

binary(x=NULL,y=NULL,log="",samples=rownames(WR),
new=TRUE, ...)

plotWithLimits(x.data, y.data,
digits.x=NULL, digits.y=NULL,log = "",new = TRUE,
xmin=.round.min.down(x.data,dec.places=digits.x,expand=TRUE),
xmax=.round.max.up(x.data,dec.places=digits.x,expand=TRUE),
ymin=.round.min.down(y.data,dec.places=digits.y,expand=TRUE),
ymax=.round.max.up(y.data,dec.places=digits.y,expand=TRUE),
xlab = "", ylab = "", fousy = "",
IDlabels=getOption("gcd.ident"), fit = FALSE, main = "",
pch = labels[names(x.data), "Symbol"],
col = labels[names(x.data), "Colour"],
cex=labels[names(x.data),"Size"],title=NULL,xaxs="i",yaxs="i",interactive=FALSE)

Arguments

x, y character; specification of the plotting variables (formulae OK).
log a vector ',x','y' or 'xy' specifying which of the axes are to be logarithmic
samples character or numeric vector; specification of the samples to be plotted.
new logical; should be opened a new plotting window?
... Further parameters to the function 'plotWithLimits'.
x.data a numerical vector with the x data.
y.data a numerical vector with the y data.
digits.x Precision to which should be rounded the x axis labels.
digits.y Precision to which should be rounded the y axis labels.
xmin, xmax limits of the x axis.
ymin, ymax limits of the y axis.
xlab, ylab labels for the x and y axes, respectively.
fousy numeric vector: if specified, vertical error bars are plotted at each data point.
IDlabels labels that are to be used to identify the individual data points
fit logical, should the data fitted by a least squares line?
main main title for the plot.
pch plotting symbols.
col plotting colours.
cex relative size of the plotting symbols.
title title for the plotting window.
xaxs, yaxs type of the x and y axes.
interactive logical; for internal use by our French colleagues.
**binaryBoxplot**

**Details**

The function `plots.with.limits` sets up the axes, labels them, plots the data and, if desired, enables the user to identify the data points interactively.

`'binary'` is the user interface to `plotWithLimits`.

The variables to be plotted are selected using the function `selectColumnLabel`. In the specification of the variables can be used also arithmetic expressions, see `calcCore` for the correct syntax.

The samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSubset` for details.

The functions are Figaro-compatible.

**Value**

None.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

plot

**Examples**

```r
binary("K2O/Na2O","Rb")

binary("Rb/Sr","Ba/Rb",log="xy",samples=1:10,col="red",pch="+",main="My plot")

plotWithLimits(WR[,"SiO2"]/10,WR[,"Na2O"]+WR[,"K2O"],xlab="SiO2/10", ylab="alkalis")

plotWithLimits(WR[,"Rb"],WR[,"Sr"],xlab="Rb",ylab="Sr",log="xy")

plotWithLimits(WR[,"SiO2"],WR[,"Rb"],fousy=WR[,"Rb"]*0.05,xlab="SiO2", ylab="Rb",fit=TRUE)
```

---

**binaryBoxplot**

**Binary boxplot**

**Description**

A binary plot combined with boxplots for both variables.

**Usage**

```r
binaryBoxplot(xaxis="",yaxis="")
```

**Arguments**

`xaxis`, `yaxis` specification of the variables. Formulae are OK.
Details

Unless specified in the call, the variables to be plotted are selected using the function `selectColumnLabel`. In the specification of the variables can be used also arithmetic expressions, see `calcCore` for the correct syntax.

The samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSubset` for details.

Value

None.

Warning

This function IS NOT Figaro-compatible.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

plot boxplot

Examples

`binaryBoxplot("SiO2/10","Na2O+K2O")`
Whole-rock composition of the Blatná suite, Central Bohemian Plutonic Complex

Description

This data set gives the whole-rock major- and trace-element contents in selected samples (monzo-gabbros, quartz monzonites and granodiorites) of the c. 345 My old high-K calc-alkaline Blatná suite of the Variscan Central Bohemian Plutonic Complex (Bohemian Massif, Czech Republic).

Usage

data(blatna)

Format

A data frame containing 11 observations.

Source

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


Examples

data(blatna)
accessVar("blatna")
binary("SiO2","Ba")

Boolean conditions

Select subset by Boolean condition

Description

Selecting subsets of the current dataset using Boolean conditions that can query both numeric fields and labels. Regular expressions can be employed to search the labels.
Details

The menu item 'Select subset by Boolean', connected to the function `selectSubset`, enables the user to query by any combination of the numeric columns and labels in the whole dataset. The current data will be replaced by its newly chosen subset.

First, the user is prompted to enter a search pattern which can contain conditions that may employ most of the comparison operators common in R, i.e. < (lower than), > (greater than), <= (lower or equal to), >= (greater or equal to), = or == (equal to), != (not equal to). The character strings should be quoted. The conditions can be combined together by logical and, or and brackets.

Logical and can be expressed as .and. .AND. &
Logical or can be expressed as .or. .OR. |

Please note that at the moment no extra spaces can be handled (apart from in quoted character strings).

Value

Overwrites the data frame ‘labels’ and numeric matrix ‘WR’ by subset that fulfills the search criteria.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

regular.expressions regex

Examples

```r
## Not run:

# Valid search patterns
Intrusion="Rum"
# Finds all analyses from Rum
Intrusion="Rum".and.SiO2>65
Intrusion="Rum".AND.SiO2>65
Intrusion="Rum"&SiO2>65
# All analyses from Rum with silica greater than 65
# (all three expressions are equivalent)

MgO>10&(Locality="Skye"|Locality="Islay")
# All analyses from Skye or Islay with MgO greater than 10
MgO>10&(Locality!="Skye"&Locality!="Islay")
# All analyses from any locality except Skye and Islay with MgO greater
# or equal to 10
Locality="^S"
# All analyses from any locality whose name starts with capital S

## End(Not run)
```
**bpplot2**  
*Box-Percentile Plot*

**Description**

Displays statistical distribution each of the variables in a data frame using a box-percentile plot (*Esty & Banfield 2003*).

**Usage**

```r
bpplot2(x, main="Box-Percentile Plot", sub="", xlab = "", ylab="", log="y", col="lightgray", horizontal=FALSE, ylim = NULL, axes=TRUE,...)
```

**Arguments**

- `x` data frame with the data to be plotted
- `main` main title for the plot
- `sub` sub title for the plot
- `xlab` label for x axis
- `ylab` label for y axis
- `log` which of the axes is to be logarithmic?
- `col` colour to fill the boxes
- `horizontal` logical, should be the orientation horizontal?
- `ylim` optional; limits for the y axis
- `axes` logical; should be the axis drawn?
- `...` additional plotting parameters

**Details**

The box-percentile plot is analogous to a `boxplot` but the width of the box is variable, mimicking the distribution of the given variable. As in boxplots, the median and two quartiles are marked by horizontal lines.
Value

None.

Warning

This function IS NOT Figaro-compatible. It means that the set of diagrams cannot be further edited in GCDkit (e.g. tools in "Plot editing" menu are inactive).

Author(s)

The code represents a modified function 'bpplot' from the package 'Hmisc' by Frank E Harrell Jr. (originally designed by Jeffrey Banfield). Adopted for GCDkit by Vojtech Janousek, <vojtech.janousek@geology.cz>.

References


---


Description

Assigns data for a La/10-Y/15-Nb/8 ternary diagram into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

Usage

Cabanis()
Cabanis

Arguments

None.

Details

The ternary plot La/10-Y/15-Nb/8 designed by Cabanis and Lecolle (1989) serves for distinguishing magmas that have originated (1) at orogenic, compressive, destructive plate boundaries (calc-alkaline, closer to the La apex and tholeiitic, closer to the Y apex); (2) in anorogenic, distensive inter-plate domains (including NMORB/EMORB and alkaline rocks); and, in between, (3) in either compressive or distensive, intra-continental, late- to post- orogenic zones. Y/Nb serves here as the "alkalinity index" and La/Y as the "calc-alkaline index". See the original paper for details.

The diagram can also serve for recognition of magmas contaminated by continental crust or resulting from magma mixing.

Value

<table>
<thead>
<tr>
<th>sheet</th>
<th>list with Figaro Style Sheet data</th>
</tr>
</thead>
<tbody>
<tr>
<td>x . data</td>
<td>x coordinates</td>
</tr>
<tr>
<td>y . data</td>
<td>y coordinates</td>
</tr>
</tbody>
</table>

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>
References


Coordinates and graph layout are taken from website of Kurt Hollocher.

See Also

figaro plotDiagram

Examples

plotDiagram("Cabanis",FALSE,TRUE)

calc Calculate a new variable

Description

Calculates a single numeric variable and appends it to the data.

Usage

calc()

Details

The formula can invoke any combination of names of existing numerical columns, with the constants, brackets, arithmetic operators +-*/* and R functions. See calcCore for a correct syntax.

If the result is a vector of the length corresponding to the number of the samples in the system, the user is prompted for the name of the new data column. Unless a column with the specified name already exists or the given name is empty, the newly calculated column is appended to the data in memory ("WR").

Value

results numerical vector with the results

Modifies, if appropriate, the numeric matrix "WR".

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

selectColumnLabel.
Examples

```r
## Not run:
# examples of valid formulae....
(Na2O+K2O)/CaO
Rb^2
log10(Sr)
mean(SiO2)/10

# ... but this command is in fact a simple R shell -
# meaning lots of fun for power users!
summary(Rb,na.rm=T)
cbind(SiO2/2,TiO2,Na2O+K2O)
cbind(major)
hist(SiO2,col="red")
boxplot(Rb~factor(groups))

# possibilities are endless
plot(Rb,Sr,col="blue",pch="*",xlab="Rb (ppm)",ylab="Sr (ppm)",log="xy")

## End(Not run)
```

calcAnomaly

Anomaly on a spiderplot

Description

Calculates a magnitude of an anomaly on a spiderplot, based on concentrations of selected neighboring elements.

Usage

```r
calcAnomaly(which.elem="Eu",dataset=WR,ref="^REE Boynton",left="Sm", right="Gd")
```

Arguments

- **which.elem** character; which element is being examined?
- **dataset** character; name of variable holding the whole-rock data.
- **ref** character; a specification of the normalization scheme.
- **left** character; a name of element to the left, used for extrapolation.
- **right** character; a name of element to the right, used for extrapolation.

Details

This is a general function that calculates a magnitude of an anomaly on a spiderplot. For the given element it is a ratio of its normalized contents divided by an extrapolated value (denoted by a star). The extrapolation is performed from two neighboring elements, one to the left and one to the right, of the examined one. But these two elements used for extrapolation do not need to be immediately adjacent.

The best known and the most commonly used is the Eu anomaly on chondrite-normalized REE plots expressed as:
The spiderplot is selected using the parameter ‘ref’ which can contain a substring (or a regular expression) specifying the name of the normalizing scheme stored in the file ‘spider.data’ of the main GCDkit directory. For details and examples, see selectNorm.

Value

A numeric matrix with a single row, containing the calculated values.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

selectNorm spider

Examples

calcAnomaly() # Eu anomaly on chondrite-normalized REE plot after Boynton (1984).

# Nb anomaly, Nb/Nb*, based on immobile NMORB spiderplot after Pearce (1984)
NbNb<-calcAnomaly(which.elem="Nb",dataset=WR, ref="^NMORB immobile",left="Th",right="La")
WR<-addOn("Nb/Nb*",as.vector(NbNb),where=WR) # Append to the current data set
Description

Calculates a user-defined parameter specified by the equation.

Usage

calcCore(equation, where = "WR", redo = TRUE)

Arguments

equation a text string to be evaluated.
where which matrix should be used?
redo logical; should be the routine called again and again?

Details

This is a core calculation function.

The expression specified by 'equation' can involve any combination of names of existing numerical columns in the matrix 'where', numbers (i.e. constants), arithmetic operators +-*^ and R functions.

The most useful of the latter are 'sqrt' (square root), 'log' (natural logarithm), 'log10' (common logarithm), 'exp' (exponential function), 'sin', 'cos' and 'tan' (trigonometric functions).

Potentially useful can be also min (minimum), max (maximum), length (number of elements/cases), 'sum' (sum of the elements), 'mean' (mean of the elements), and 'prod' (product of the elements).

However, any user-defined function can be also invoked here.

For most statistical functions, an useful parameter 'na.rm=T' can be specified. This makes the function to calculate the result from the available data only, ignoring the not determined value (see Examples).

The quotation marks in 'equation' need to preceded by a backslash. Option 'redo' specifies whether the routine should be called repeatedly until some meaningful result is obtained. Otherwise 'NA' is returned.

Value

A list of three items:

equation equation as entered by the user
results numeric vector with the results or NA if none can be calculated
formula the unevaluated expression corresponding to the 'equation'

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>
Examples

calcCore("SiO2/10")

calcCore("Na2O+K2O")

calcCore("log10(Na2O+K2O)")

# dividing by the built-in molecularWeight, NB the backslashes

calcCore("SiO2/MW["SiO2"]")

calcCore("length(MgO)")

# na.rm is a safety measure in case some missing values are present
# otherwise the result would be 'NA'

Catanorm(Niggli’s Molecular Norm (Catanorm))

Description

Calculates the Niggli’s Molecular Norm (Catanorm) using the algorithm given by Hutchison (1974).

Usage

Catanorm(WR, precision=getOption("gcd.digits"))

Arguments

WR a numerical matrix; the whole-rock data to be normalized.

precision precision of the result.

Details

Normative minerals of the Catanorm

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Full name</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>Quartz</td>
<td>SiO$_2$</td>
</tr>
<tr>
<td>C</td>
<td>Corundum</td>
<td>AlO$_{1.5}$</td>
</tr>
<tr>
<td>Or</td>
<td>Orthoclase</td>
<td>KO$<em>{0.5}$.AlO$</em>{1.5}$.3SiO$_2$</td>
</tr>
<tr>
<td>Plag</td>
<td>Plagioclase</td>
<td>Ab$<em>x$.An$</em>{100-x}$</td>
</tr>
<tr>
<td>Ab</td>
<td>(Albite)</td>
<td>NaO$<em>{1.5}$.AlO$</em>{1.5}$.3SiO$_2$</td>
</tr>
<tr>
<td>An</td>
<td>(Anorthite)</td>
<td>CaO.2AlO$_{1.5}$.2SiO$_2$</td>
</tr>
<tr>
<td>Lc</td>
<td>Leucite</td>
<td>KO$<em>{0.5}$.AlO$</em>{1.5}$.2SiO$_2$</td>
</tr>
<tr>
<td>Ne</td>
<td>Nepheline</td>
<td>NaO$<em>{0.5}$.AlO$</em>{1.5}$.SiO$_2$</td>
</tr>
<tr>
<td>Kp</td>
<td>Kaliophilite</td>
<td>KO$<em>{0.5}$.AlO$</em>{1.5}$.SiO$_2$</td>
</tr>
<tr>
<td>Ac</td>
<td>Acmite</td>
<td>NaO$<em>{0.5}$.FeO$</em>{1.5}$.2SiO$_2$</td>
</tr>
<tr>
<td>Ns</td>
<td>Sodium metasilicate</td>
<td>2NaO$_{0.5}$.SiO$_2$</td>
</tr>
<tr>
<td>Ks</td>
<td>Potassium metasilicate</td>
<td>2KO$_{0.5}$.SiO$_2$</td>
</tr>
<tr>
<td>Hy</td>
<td>Hypersthene</td>
<td>En$<em>x$.Fs$</em>{100-x}$</td>
</tr>
<tr>
<td>Di</td>
<td>Diopside</td>
<td>Wo$_{50}$.En$<em>x$.Fs$</em>{50-x}$</td>
</tr>
<tr>
<td>Wo</td>
<td>(Wollastonite)</td>
<td>CaO.SiO$_2$</td>
</tr>
</tbody>
</table>
### Value

A numeric matrix 'results'.

### Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

### References


### Examples

```r
data(sazava)
accessVar("sazava")
Catanorm(WR)
```

### Description

Calculates various modifications of the CIPW norm.

### Usage

```r
CIPW(wrdata, precision = getOption("gcd.digits"), normsum = FALSE, cancrinite = FALSE, spinel = FALSE, complete.results = FALSE)
CIPWhb(wrdata, precision = getOption("gcd.digits"), normsum = FALSE, cancrinite = FALSE, spinel = FALSE, complete.results = FALSE)
```
Arguments

- **wrdata**: a numerical matrix; the whole-rock data to be normalized.
- **precision**: precision of the result.
- **normsum**: logical; shall be the normative minerals recast to 100%?
- **cancrinite**: logical; is cancrinite present/to be calculated?
- **spinel**: logical; is spinel to be calculated (for ultrabasic rocks, i.e. for samples with $SiO_2 < 45\%$ only)?
- **complete.results**: logical; should be returned more extensive list of minerals, including the end members making up Di, Hy, Ol, Bi and Hbl?

Details

The method adopted for 'classic' CIPW norm calculation is that of Hutchison (1974, 1975). The function 'CIPWHB' is its modification with biotite and hornblende (Hutchison 1975).

### Normative minerals of the standard CIPW norm

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Full name</th>
<th>Formula</th>
<th>Molecular weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>Quartz</td>
<td>$SiO_2$</td>
<td>60.08</td>
</tr>
<tr>
<td>C</td>
<td>Corundum</td>
<td>$Al_2O_3$</td>
<td>101.96</td>
</tr>
<tr>
<td>Or</td>
<td>Orthoclase</td>
<td>$K_2O.Al_2O_3.6SiO_2$</td>
<td>556.64</td>
</tr>
<tr>
<td>Ab</td>
<td>Albite</td>
<td>$Na_2O.Al_2O_3.6SiO_2$</td>
<td>524.42</td>
</tr>
<tr>
<td>An</td>
<td>Anorthite</td>
<td>$CaO.Al_2O_3.2SiO_2$</td>
<td>278.20</td>
</tr>
<tr>
<td>Lc</td>
<td>Leucite</td>
<td>$K_2O.Al_2O_3.4SiO_2$</td>
<td>436.48</td>
</tr>
<tr>
<td>Ne</td>
<td>Nepheline</td>
<td>$Na_2O.Al_2O_3.2SiO_2$</td>
<td>284.10</td>
</tr>
<tr>
<td>Kp</td>
<td>Kaliophilite</td>
<td>$K_2O.Al_2O_3.2SiO_2$</td>
<td>316.32</td>
</tr>
<tr>
<td>Nc</td>
<td>Sodium carbonate</td>
<td>$Na_2O.CO_2$</td>
<td>105.99</td>
</tr>
<tr>
<td>Ac</td>
<td>Acmite</td>
<td>$Na_2O.FeO_2Si_2O_4$</td>
<td>461.99</td>
</tr>
<tr>
<td>Ns</td>
<td>Sodium metasilicate</td>
<td>$Na_2O.SiO_2$</td>
<td>122.06</td>
</tr>
<tr>
<td>Ks</td>
<td>Potassium metasilicate</td>
<td>$K_2O.SiO_2$</td>
<td>154.28</td>
</tr>
<tr>
<td>Di</td>
<td>Diopside</td>
<td>$(MgDi) CaO.MgO.2SiO_2$</td>
<td>216.55</td>
</tr>
<tr>
<td></td>
<td>(FeDi)</td>
<td>$(Fe-diopside) CaO.FeO.2SiO_2$</td>
<td>248.09</td>
</tr>
<tr>
<td>Wo</td>
<td>Wollastonite</td>
<td>$CaO.SiO_2$</td>
<td>116.16</td>
</tr>
<tr>
<td>Hy</td>
<td>Hypersthene</td>
<td>$(En) MgO.SiO_2$</td>
<td>100.39</td>
</tr>
<tr>
<td></td>
<td>(Fs)</td>
<td>$(Ferrosilite) FeO.SiO_2$</td>
<td>131.93</td>
</tr>
<tr>
<td>Ol</td>
<td>Olivine</td>
<td>$(Fo) 2MgO.SiO_2$</td>
<td>140.70</td>
</tr>
<tr>
<td></td>
<td>(Fa)</td>
<td>$(Fayalite) 2FeO.2SiO_2$</td>
<td>203.78</td>
</tr>
<tr>
<td>Dcs</td>
<td>Dicalcium silicate</td>
<td>$2CaO.SiO_2$</td>
<td>172.24</td>
</tr>
<tr>
<td>Mt</td>
<td>Magnetite</td>
<td>$FeO.Fe_2O_3$</td>
<td>231.54</td>
</tr>
<tr>
<td>Il</td>
<td>Ilmenite</td>
<td>$FeO.TiO_2$</td>
<td>151.75</td>
</tr>
<tr>
<td>Hm</td>
<td>Hematite</td>
<td>$Fe_2O_3$</td>
<td>159.69</td>
</tr>
<tr>
<td>Tn</td>
<td>Sphene</td>
<td>$CaO.TiO_2.2SiO_2$</td>
<td>196.06</td>
</tr>
<tr>
<td>Pf</td>
<td>Perovskite</td>
<td>$CaO.TiO_2$</td>
<td>135.98</td>
</tr>
<tr>
<td>Ru</td>
<td>Rutile</td>
<td>TiO_2.SiO_2</td>
<td>79.90</td>
</tr>
<tr>
<td>Ap</td>
<td>Apatite</td>
<td>$3CaO.P_2O_5.1/3CaF_2$</td>
<td>336.21</td>
</tr>
<tr>
<td>Fr</td>
<td>Fluorite</td>
<td>$CaF_2$</td>
<td>78.08</td>
</tr>
<tr>
<td>Py</td>
<td>Pyrite</td>
<td>$FeS_2$</td>
<td>119.98</td>
</tr>
<tr>
<td>Sp</td>
<td>Spinel</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
classify

\[ (\text{MgSp}) \quad \text{(Mg-spinel; spinel s. s.)} \quad CaO.MgO.2SiO_2 \quad 142.27 \]
\[ (\text{FeSp}) \quad \text{(Fe-spinel; hercynite)} \quad CaO.FeO.2SiO_2 \quad 173.81 \]

\[ \text{Cc} \quad \text{Calcite} \quad CaO.CO_2 \quad 100.09 \]

**Additional minerals of the modification with hornblende and biotite**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Full name</th>
<th>Formula</th>
<th>Molecular weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bi</td>
<td>Biotite</td>
<td>( KO_{0.5}MgO.AlO_{1.5}SiO_2 )</td>
<td>798.50</td>
</tr>
<tr>
<td>( (\text{MgBi}) )</td>
<td>( (\text{Phlogopite}) )</td>
<td>( KO_{0.5}MgO.AlO_{1.5}SiO_2 )</td>
<td>798.50</td>
</tr>
<tr>
<td>( (\text{FeBi}) )</td>
<td>( (\text{Annite}) )</td>
<td>( KO_{0.5}FeO.AlO_{1.5}SiO_2 )</td>
<td>987.74</td>
</tr>
<tr>
<td>Hbl</td>
<td>Hornblende</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Act</td>
<td>Actinolite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( (\text{MgAct}) )</td>
<td>( (\text{Tremolite}) )</td>
<td>( 2CaO.5MgO.8SiO_2 )</td>
<td>794.35</td>
</tr>
<tr>
<td>( (\text{FeAct}) )</td>
<td>( (\text{Ferroactinolite}) )</td>
<td>( 2CaO.5FeO.8SiO_2 )</td>
<td>952.05</td>
</tr>
<tr>
<td>Ed</td>
<td>Edenite</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( (\text{MgEd}) )</td>
<td>( (\text{Edenite}) )</td>
<td>( NaO_{0.5}.2CaO.5MgO.AlO_{1.5}SiO_2 )</td>
<td>1632.48</td>
</tr>
<tr>
<td>( (\text{FeEd}) )</td>
<td>( (\text{Ferroedenite}) )</td>
<td>( NaO_{0.5}.2CaO.5FeO.AlO_{1.5}SiO_2 )</td>
<td>1947.88</td>
</tr>
<tr>
<td>Ri</td>
<td>Riebeckite</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Value**

A numeric matrix 'results'.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**References**


**Examples**

```r
data(sazava)
accessVar("sazava")
CIPW(WR)
```

classify **Generic Classification Algorithm**

**Description**

Classifies rocks using specified diagram.
Usage

classify(diagram = NULL, grp = TRUE, labs = FALSE, source.sheet = TRUE, overlap = FALSE, X = x.data, Y = y.data, silent = FALSE, clas=sheet$d$t, ...)

Arguments

diagram name of diagram to be used, see details for more info
grp logical: if TRUE, results are assigned to the variable 'groups'
labs logical: if TRUE, yes/no dialogue for results assignment into the matrix 'labels' appears
source.sheet logical: if TRUE, the sheet for diagram is newly assigned
overlap logical: if TRUE, possible overlap between polygons of diagram is expected, and duplicate positive result for one sample is treated as polygon intersection
X vector of values for abscissa
Y vector of values for ordinate
silent logical: if TRUE, informative outputs are reduced to minimum
clas classification template to be used
... any additional graphical parameters

Details

Function looks for the name of the polygon within the classification diagram, into which falls the rock analysis represented by the coordinates [x.data, y.data].

In some cases (TAS diagram, Winchester & Floyd’s diagram) additional computations are performed. The argument 'diagram' may acquire one of following values:

'AFM', 'PeceTaylor', 'Shand', 'TAS', 'CoxPlut', 'CoxVolc',
'Jensen', 'LarochePlut', 'LarocheVolc', 'WinFloyd1',
'WinFloyd2', 'TASMiddlemostPlut', 'TASMiddlemostVolc',
'DebonPQ', 'DebonBA', 'MiddlemostPlut', 'QAPFPlut',
'QAPFVolc', 'OConnorPlut', 'QAPFVolc', 'OConnorVolc',
'Miyashiro', 'Hastie', 'Pearce1996', 'Villaseca', 'NaAIK'.

The function is based on the sp package.

Value

Vector of resulting rock names is stored in a variable 'results'. If 'grp = TRUE' results are also assigned to the 'groups' and 'grouping' is set to -1 (as if called from the menu 'Data handling'). If rock projection falls on the boundary between two or more fields, rock names in question are merged together with comment 'boundary between ...'.
Author(s)

The sp package was written by Edzer Pebesma, Roger Bivand and others.
Vojtech Erban, <erban@sopky.cz>

See Also

plotDiagram
claslist
figaro
AFM, PecTaylor, Shand, NaAlK, TAS, Cox, TASMiddlemost, Jensen, Laroche, WinFloyd1, WinFloyd2, DebonPQ, DebonBA, Middlemost, QAPF, OConnor, Miyashiro, Hastie, Pearce1996, Villaseca

clr.transform

Centered-log-ratio transformation

Description

Implementation of centred-log-ratio (clr) transformation for compositional data.

Usage

clr.trans(comp.data=NULL, GUI=FALSE)
pr.comp.clr(comp.data="SiO2, TiO2, Al2O3, FeOt, MnO, MgO, CaO, Na2O, K2O", cor=TRUE, GUI=FALSE)

Arguments

comp.data a numerical matrix; the data to be normalized. Or just names of variables in the data matrix 'WR'.
cor logical; should be the correlation matrix used instead of covariance matrix?
GUI logical; is the function called from a menu (GUI)?

Details

Compositional data - i.e., multivariate data in which all the components sum up to some constant (e.g. 1 or 100, for percentages) - are widespread in the geosciences. A typical example represent major-element analyses from whole-rock samples.

Numerous workers have argued that much of correlation in such closed datasets is spurious, due to the so-called constant sum or closure effect (e.g., Chayes 1960; Rock 1988; Rollinson 1992, 1993).

This effect arises from the fact that such components in the compositional datasets cannot vary independently. If one oxide, for instance SiO2 that dominates the whole-rock analyses of many igneous rocks, increases in abundance, all other oxides must decrease. Therefore, everything must be anti-correlated with silica.

For their correct statistical treatment, compositional data have to be transformed, or 'opened'. A classic remedy to the closure effect are log-ratio transformations (Aitchison 1986; Buccianti et al. eds 2006).
The functions ‘clr.trans’ and ‘pr.comp.clr’ implement the so-called centred-log-ratio (clr) transformation. Data opening in this case is done by dividing each value of a variable by the geometric mean of all the variables for that sample and then taking logarithms. It is critical of course that all the variables are expressed in the same measurement unit.

For instance, for MgO, the centred-log-ratio transformed version is given as:

\[
MgO_{\text{clr}} = \ln \left( \frac{C_{\text{MgO}}}{\sqrt[n]{\prod_{i=1}^{n} C_i}} \right)
\]

where ‘ln’ is natural logarithm, ‘C’ concentration in wt. % of the selected variable (oxide) and the denominator a geometric mean of all variables being transformed (e.g., Pawlowsky-Glahn & Egozcue 2006)).

Value

A numeric matrix ‘results’. The names of components are preserved, and supplemented by a suffix ‘_clr’.

Plugin
disclosure.r

Author(s)

Vojtěch Janoušek, <vojtech.janousek@geology.cz>
Vladimír Kusbach, <kusbach@gmail.com>

References


See Also

prComp


Examples

data(sazava)
accessVar("sazava")

ox<-c("SiO2","Al2O3","FeOt","MgO","CaO")
clr.trans(ox)
addResults() # Needed to append the clr-transformed data to the matrix 'WR'

multiple(x = "SiO2_clr", y = "Al2O3_clr,FeOt_clr,MgO_clr,CaO_clr")
plateCex(2)
plateCexLab(1.3)

pr.comp.clr(ox)

---

**cluster**

Statistics: Hierarchical clustering

Description

Hierarchical cluster analysis on a set of dissimilarities.

Usage

cluster(elems = "SiO2,TiO2,Al2O3,FeOt,MnO,MgO,CaO,Na2O,K2O", method = "average")

Arguments

elems numerical columns to be used for cluster analysis, typically major elements

method the agglomeration method to be employed. This should be one of (or an unambiguous abbreviation thereof): 'ward', 'single', 'complete', 'average', 'mcquitty', 'median', 'centroid'.
Details

The samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSamples` for details.

Even though a list of major elements is assumed as a default, different variables can be specified by the function `selectColumnsLabels`.

The user is also asked to specify a label for the individual samples, default are their names.

After the dendrogram is drawn, the individual clusters can be identified. For each sample falling into the given group, specified information (e.g. Locality, Rock Type and/or Author) can be printed.

For further details on the clustering algorithm, see the R manual entry of `hclust`.

Value

None.

Warning

Names of existing numeric data columns and not formulae involving these can be handled at this stage. Only complete cases are used for the cluster analysis.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

`hclust`

---

**contourGroups**  
*Outline individual groups in a binary plot*

Description

The functions outline the individual clusters of data (groups by default) on a binary plot. Implemented methods are the convex hull or contours. This can be useful for a quick appreciation of the data distribution, e.g. in classification diagrams.

Usage

```r
contourGroups(clusters=groups,border=NULL,fill=FALSE,precision=50, ...)
chullGroups(clusters=groups,border=NULL,fill=FALSE,...)
```

Arguments

- `clusters` grouping information for each of the samples.
- `border` outline colours.
- `fill` logical; should be the polygons filled by the border colour?
- `precision` a number indicating how tight the contours should be.
- `...` additional parameters to the functions `contour` and `polygon`, respectively.
Details

If not specified, the colours are selected as the most frequently occurring one defined among samples within each group.

For the function `contourGroups`, the shape of the contours drawn can be controlled using the parameter (`precision`). The higher it is, the smoother contours result.

Value

None.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

`chull`, `contour`, `polygon`

Examples

```r
data<-loadData("sazava.data",sep="\t")
groupsByLabel("Intrusion")
plotDiagram("PeceTaylor",FALSE,FALSE)
chullGroups()

chullGroups(fill=TRUE)

plotDiagram("PeceTaylor",FALSE,FALSE)
contourGroups()
```
coplotByGroup

Coplot by groups

Description

Plots a series of binary plots, for each of the groups separately.

Usage

    coplotByGroup(xaxis = "", yaxis = "", show.leg = ")

Arguments

- **xaxis**: Name of the data column to be used as x axis.
- **yaxis**: Name of the data column to be used as y axis.
- **show.leg**: Logical: are the levels of the conditioning variable ('groups') to be shown?

Details

For examination of large datasets split into user-defined subsets serves in R function `coplot`. It produces a set of binary diagrams with the data filtered out according to the values of the third (conditioning) variable. In case of the function 'coplotByGroup' it is done by groups.
If no parameters 'xlab', 'ylab' and 'show.leg' are given, the user is prompted to specify them. The variables to be plotted are selected using the function `selectColumnLabel`. See manual entry for `coplot` for further details.

**Value**

None.

**Warning**

Please note that no formulae can be handled at this stage.

This function IS NOT Figaro-compatible.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz> & Vojtech Erban, <erban@sopky.cz>
coplotTri

See Also

'coplot'

Examples

coplotByGroup("SiO2","Na2O",show.leg=TRUE)

coplotTri(xaxis = "", yaxis = "", zaxis = "", int = "")

Arguments

xaxis Name of the data column to be used as x axis.
yaxis Name of the data column to be used as y axis.
zaxis Name of the data column with the conditioning variable.
int The specification of the intervals: either 'auto' or a list of break points separated by commas.

Details

For examination of large datasets split into user-defined subsets serves in R the function 'coplot'. It displays a series of binary diagrams with the data filtered out according to the values of the third (conditioning) variable.
If no parameters 'xlab', 'ylab' and 'zlab' are given, the user is prompted to specify them. The variables to be plotted are selected using the function `selectColumnLabel`. After this is done, the user is prompted to enter a comma-delimited list of at least one break point defining the intervals. The default includes the mean, that will be automatically supplemented by minimum and maximum (i.e. two intervals). See manual entry for `coplot` for further details.

**Value**

None.

**Warning**

Please note that no formulae can be handled at this stage. This function IS NOT Figaro-compatible.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz> & Vojtech Erban, <erban@sopky.cz>
correlationCoefPlot

See Also

' coplot '

Examples

coplotTri("SiO2","Na2O","MgO","auto")

coplotTri("MgO","Na2O","SiO2","50,60")
# the intervals of the conditioning variable, SiO2,
# will be (min(SiO2) - 50),(50 - 60) and (60 - max(SiO2))

Description

Produces, for each group a separate, set of plots of correlation coefficient patterns.

Usage

correlationCoefPlot(elems = NULL)

Arguments

elems list of desired elements

Details

The utility of correlation coefficient patterns was discussed by Rollinson (1993 and references therein). Basically similarity in correlation patterns between two or more elements means their analogous geochemical behaviour, potentially controlled by the same geochemical process (fractional crystallization, partial melting, weathering, hydrothermal alteration...).
The variables are selected using the function `selectColumnsLabels`.

**Value**

None.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**References**


**Examples**

`correlationCoefPlot(elems="K,Rb,Sr,Cr,Nb,Ti")`

---

**TAS diagram (Cox et al. 1979)**

**Description**

Assigns data for Cox’s diagram into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.
Usage

CoxVolc(alkline=TRUE)

CoxPlut(alkline=TRUE)

Arguments

alkline  Logical: Should the boundary between alkaline and subalkaline rocks (Irvine & Baragar 1971) be drawn?

Details

TAS diagram, as proposed by Cox et al. (1979) for volcanic rocks and adapted by Wilson (1989) for plutonic rocks.

For volcanic rocks, the following diagram is plotted:

![TAS diagram](image)

And the version for plutonic rocks contains the following fields:

<table>
<thead>
<tr>
<th>volcanic rocks</th>
<th>plutonic rocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>basalt</td>
<td>gabbro</td>
</tr>
<tr>
<td>basaltic andesite</td>
<td>undefined</td>
</tr>
<tr>
<td>andesite</td>
<td>diorite</td>
</tr>
<tr>
<td>dacite</td>
<td>quartz diorite (granodiorite)</td>
</tr>
<tr>
<td>rhyolite</td>
<td>alkali granite/granite</td>
</tr>
<tr>
<td>hawaiite</td>
<td>gabbro</td>
</tr>
<tr>
<td>trachyandesite</td>
<td>undefined</td>
</tr>
<tr>
<td>basanite/tephrite</td>
<td>undefined</td>
</tr>
<tr>
<td>mugearite</td>
<td>syeno-diorite</td>
</tr>
</tbody>
</table>
Value

   sheet          list with Figaro Style Sheet data
   x.data          SiO2 weight percent
   y.data          Na2O+K2O weight percent

Warning

Note that, unlike in the standard TAS plot, the diagram is based on standard analyses (i.e. it is not recalculated on the volatile-free basis).

Author(s)

Vojtech Erban, <erban@sopky.cz>
& Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also
classify figaro plotDiagram TAS

Examples

#TAS diagram is called using following auxiliary functions:
#Classifies data stored in WR (Groups by diagram)
classify("CoxVolc")
#or
classify("CoxPlut")

#Plots data stored in WR or its subset (menu Classification)
plotDiagram("CoxVolc", FALSE)
#or
plotDiagram("CoxPlut", FALSE)

crosstab

Cross table of labels

Description

Prints a cross table (contingency table) for 1-3 labels.

Usage

crosstab(plot = TRUE)

Arguments

plot logical; should be also a barplot plotted?

Details

This command prints a frequency distribution (for a single label) or a contingency table (for 2-3 labels) useful for inspection of the data structure. Optionally a barplot is plotted (for 1-2 labels).

Just press Enter (enter an empty field), when the desired number of variables is reached.

Value

results the frequency/contingency table

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>
customScript

Add a new variable to a script

Description

Adds a formula to calculate a single numeric variable to the specified *.r file (a R script).

Usage

customScript()

Details

A formula can be entered that can involve any combination of names of existing numerical columns, with the constants, brackets, arithmetic operators \(+-*/^\) and R functions. See calcCore for a correct syntax.

Then the user is prompted for the name of the variable an any comments that should appear in the file.

The filename is chosen interactively, the default suffix for the R programs is .r. If the file exists already, the script is appended to its end.

If desired, the calculated variable can be, after the script is executed, added automatically to the numeric data, i.e. the numeric matrix \(WR\). If not, the contents of the calculated variable can be viewed by simply typing its name in the R Console window.

The script can be run at a later time using the R command File|Source. Alternatively, it can be placed among the so-called plugins into the subdirectory Plugin. All files placed here with a suffix *.r are executed each time when the new data file is being loaded into the GCDkit.

Value

None.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

Examples

```r
## Not run:
# examples of valid formulae....
(Na2O+K2O)/CaO
Rb^2
log10(Sr)
mean(SiO2)/10

# ... but this command is in fact a simple R shell -
# meaning lots of fun for power users!
summary(Rb,na.rm=T)
cbind(SiO2/2,TiO2,Na2O+K2O)
cbind(major)
hist(SiO2,col="red")
boxplot(Rb~factor(groups))
```
# possibilities are endless
plot(Rb,Sr,col="blue",pch="*",xlab="Rb (ppm)",ylab="Sr (ppm)",log="xy")

## End(Not run)

---

**cutMy**

*Groups by numerical variable*

### Description

Grouping the data according to the interval of a single numerical variable it falls into.

### Usage

```r
cutMy(where=NULL,int=NULL,int.lab=NULL,na.lab="Unclassified")
```

### Arguments

- **where**: Numeric data column in `WR` - the basis of the classification.
- **int**: Boundaries of intervals.
- **int.lab**: Labels for the intervals.
- **na.lab**: Labels for samples that cannot be classified.

### Details

The numeric data column is selected using the function `selectColumnLabel`.

After this is done, the user is prompted to enter a comma-delimited list or at least one break point defining the intervals. The default includes the mean, that will be automatically supplemented by minimum and maximum (i.e. two intervals).

Then the names of the individual groups are to be specified; values out of range are automatically labeled as 'Unclassified'. The vector containing the information on the current groups can be appended to the data frame 'labels'.

### Value

- **groups**: character vector: the grouping information;
- **grouping**: If the new column was appended the data frame `labels`, sequence number of this column; if not appended, though, this variable is set to -100.

### Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

### See Also

- `cut`
Description

Assigns data for Debon & Le Fort’s B-A and P-Q diagrams into Figaro template (list `sheet`) and appropriate values into `x.data` and `y.data`.

Usage

```r
DebonBA(reference.rocks=FALSE)
DebonPQ(reference.rocks=FALSE)
```

Arguments

- `reference.rocks` logical; should be the ideal rock compositions also plotted?

Details

The B-A diagram as proposed by *Debon and Le Fort (1983)* defines six sectors (I - VI), reflecting alumina balance of samples. Following minerals are characteristic for individual sectors:

<table>
<thead>
<tr>
<th>Sector</th>
<th>Domain</th>
<th>Characteristic Minerals</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Peraluminous</td>
<td>muscovite &gt; biotite</td>
</tr>
<tr>
<td>II</td>
<td>Bio-ferroaluminous</td>
<td>biotite &gt; muscovite</td>
</tr>
<tr>
<td>III</td>
<td>Bio-basaltic</td>
<td>biotite (+- minor amphibole)</td>
</tr>
<tr>
<td>IV</td>
<td>Metaluminous</td>
<td>biotite, amphibole, +- pyroxene</td>
</tr>
</tbody>
</table>
V  clinopyroxene, +/- amphibole, +/- biotite
VI  unusual mineral associations (carbonatites . . . )

Layout of the P-Q diagram of the same authors corresponds to cationic proportions of quartz, K-feldspar and plagioclase. Abbreviations used as classification output represent following rocks groups:

<table>
<thead>
<tr>
<th>label</th>
<th>plutonic rocks</th>
<th>volcanic rocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>go</td>
<td>gabbro, diorite, anorthosite</td>
<td>basalt, andesite, kenningite</td>
</tr>
<tr>
<td>mgo</td>
<td>monzogabbro, monzodiorite</td>
<td>latibasalt, latiandesite</td>
</tr>
<tr>
<td>mz</td>
<td>monzonite</td>
<td>laite</td>
</tr>
<tr>
<td>s</td>
<td>syenite</td>
<td>trachyte</td>
</tr>
<tr>
<td>dq</td>
<td>qtz diorite, qtz gabbro, qtz anorthosite</td>
<td>qtz andesite, qtz basalt</td>
</tr>
<tr>
<td>mzdq</td>
<td>qtz monzodiorite, qtz monzogabbro</td>
<td>qtz latiandesite, qtz latibasalt</td>
</tr>
<tr>
<td>mzq</td>
<td>quartz monzonite</td>
<td>quartz latite</td>
</tr>
<tr>
<td>sq</td>
<td>quartz syenite</td>
<td>quartz trachyte</td>
</tr>
<tr>
<td>to</td>
<td>tonalite, trondhjemite</td>
<td>dacite</td>
</tr>
<tr>
<td>gd</td>
<td>granodiortie, granogabbro</td>
<td>rhyodacite</td>
</tr>
<tr>
<td>ad</td>
<td>adamelite</td>
<td>dellenite</td>
</tr>
<tr>
<td>gr</td>
<td>granite</td>
<td>rhyolite</td>
</tr>
</tbody>
</table>

Parameters for the diagram are calculated by the function ‘DebonCalc’. All of them are based on millications (1000 gram-atoms per 100 grams).

P = K - (Na + Ca)
Q = Si / 3 - (K + Na + 2 * Ca / 3)
A = Al - (K + Na + 2 * Ca)
B = Fe + Mg + Ti

Note that the diagrams B-A and P-Q are recommended as complementary, i.e. resulting names should be used in conjunction (granite II etc.). For details, see Debon & Le Fort (1983) or (1988).

Value

- sheet: list with Figaro Style Sheet data
- x.data: P or B value. See details.
- y.data: Q or A value. See details.

Author(s)

Vojtech Erban, <erban@sopky.cz>
& Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

classify figaro plotDiagram DebonCalc DebonQB DebonKNaB DebonBMgNo DebonBQF

Examples

data(blatna)
accessVar("blatna")
plotDiagram("DebonBA",FALSE,reference.rocks=TRUE)
figCol("red")

options("gcd.plot.text"=FALSE)
plotDiagram("DebonPQ",FALSE,reference.rocks=TRUE)
figCol("red")
options("gcd.plot.text"=TRUE)

DebonBMgNo

B vs. Mg number diagram (Debon + Le Fort 1988)

Description

Assigns data for Debon & Le Fort’s B vs. Mg/(Fe + Mg) diagram into Figaro template (list ’sheet’) and appropriate values into ’x.data’ and ’y.data’.

Usage

DebonBMgNo(reference.rocks=TRUE)
Arguments

reference.rocks

logical; should be the ideal rock compositions also plotted?.

Details

The B vs. Mg/(Fe + Mg) diagram was proposed by Debon and Le Fort (1983) to distinguish magnesian and ferriferrous associations.

Parameters for the diagram are calculated by the function `DebonCalc`. All of them are based on millications (1000 gram-atoms per 100 grams).

\[ B = Fe + Mg + Ti \] [m aficity]

\[ Mg/(Fe + Mg) \] [Mg number, Fe is total iron]

Optionally, if, `reference.rocks = TRUE`, twelve average chemical compositions of common igneous rocks are also plotted, as follows:

<table>
<thead>
<tr>
<th>label</th>
<th>plutonic rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>go</td>
<td>gabbro</td>
</tr>
<tr>
<td>mzgo</td>
<td>monzogabbro</td>
</tr>
<tr>
<td>nz</td>
<td>monzonite</td>
</tr>
<tr>
<td>s</td>
<td>syenite</td>
</tr>
<tr>
<td>dq</td>
<td>quartz diorite</td>
</tr>
<tr>
<td>mzdq</td>
<td>quartz monzodiorite</td>
</tr>
<tr>
<td>mzq</td>
<td>quartz monzonite</td>
</tr>
<tr>
<td>sq</td>
<td>quartz syenite</td>
</tr>
<tr>
<td>to</td>
<td>tonalite</td>
</tr>
<tr>
<td>gd</td>
<td>granodiorite</td>
</tr>
</tbody>
</table>
DebonBQF

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ad</td>
<td>adamellite</td>
</tr>
<tr>
<td>gr</td>
<td>granite</td>
</tr>
</tbody>
</table>

For details, see Debon & Le Fort (1983) and (1988).

Value

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>sheet</td>
<td>list with Figaro Style Sheet data</td>
</tr>
<tr>
<td>x.data</td>
<td>Q value. See details.</td>
</tr>
<tr>
<td>y.data</td>
<td>B value. See details.</td>
</tr>
</tbody>
</table>

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

classify figaro plotDiagram DebonPQ DebonBA DebonQB DebonKNaB DebonBQF DebonCalc

Examples

data(blatna)
accessVar("blatna")
selectSubset("SiO2>50")
plotDiagram("DebonBMgNo",FALSE,reference.rocks=TRUE)
figCol("red")

DebonBQF

BQF ternary diagram (Debon + Le Fort 1983)

Description

Assigns data for Debon & Le Fort’s BQF ternary diagram into Figaro template (list ’sheet’) and appropriate values into ’x.data’ and ’y.data’.

Usage

DebonBQF(reference.rocks=TRUE)

Arguments

reference.rocks

logical; should be the ideal rock compositions also plotted?.
Details

The BQF diagram as proposed by *Debon and Le Fort (1983)* expresses balance of three main groups of rock-forming minerals, dark minerals (B), quartz (Q) and feldspars with muscovite (F).

Parameters for the diagram are calculated by the function `DebonCalc`. All of them are based on millications (1000 gram-atoms per 100 grams) and named as follows:

- Q (quartz)
- B (dark m.)
- F (feldsp)

Optionally, if `reference.rocks = TRUE`, twelve average chemical compositions of common igneous rocks are also plotted, as follows:

<table>
<thead>
<tr>
<th>label</th>
<th>plutonic rock</th>
</tr>
</thead>
<tbody>
<tr>
<td>go</td>
<td>gabbro</td>
</tr>
<tr>
<td>mzgo</td>
<td>monzogabbro</td>
</tr>
<tr>
<td>mz</td>
<td>monzonite</td>
</tr>
<tr>
<td>s</td>
<td>syenite</td>
</tr>
<tr>
<td>dq</td>
<td>quartz diorite</td>
</tr>
<tr>
<td>mzdq</td>
<td>quartz monzodiorite</td>
</tr>
<tr>
<td>mzq</td>
<td>quartz monzonite</td>
</tr>
<tr>
<td>sq</td>
<td>quartz syenite</td>
</tr>
<tr>
<td>to</td>
<td>tonalite</td>
</tr>
<tr>
<td>gd</td>
<td>granodiorite</td>
</tr>
<tr>
<td>ad</td>
<td>adamellite</td>
</tr>
<tr>
<td>gr</td>
<td>granite</td>
</tr>
</tbody>
</table>

For details, see *Debon & Le Fort (1983)* or (1988).

Value

- `sheet` list with Figaro Style Sheet data
DebonKNaB

x.data, y.data  Q(quartz), B(dark m.) and F(feldsp) values (see details) transformed into 2D

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

classify figaro plotDiagram DebonPQ DebonBA DebonQB DebonKNaB DebonBMgNo DebonCalc

Examples

data(blatna)
accessVar("blatna")
plotDiagram("DebonBQF",FALSE,reference.rocks=TRUE)
figCol("red")

DebonKNaB  K/(Na+K) vs. B diagram (Debon + Le Fort 1983)

Description

Assigns data for Debon & Le Fort’s K/(Na+K) vs. B diagram into Figaro template (list ‘sheet’) and appropriate values into ‘x.data’ and ‘y.data’.

Usage

DebonKNaB()

Details

The K/(Na+K) vs. B diagram was proposed by Debon and Le Fort (1988) for aluminous magmatic suites. It defines three associations based on the balance of the two alkalis, namely potassic, sodipotassic and sodic (note that the x axis is reversed). Moreover, leucocratic, subleucocratic and mesocratic associations are distinguished based on maficity of the samples.

Parameters for the diagram are calculated by the function `DebonCalc`. All of them are based on millications (1000 gram-atoms per 100 grams).

K/(Na + K) [alkali ratio]
B = Fe + Mg + Ti [maficity]
For details, see Debon & Le Fort (1983) or (1988).

Value

- **sheet**: list with Figaro Style Sheet data
- **x.data**: K/(Na + K) value. See details.
- **y.data**: B value. See details.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

classify figaro plotDiagram DebonPQ DebonBA DebonQB DebonBQF DebonBMgNo DebonCalc

Examples

data(blatna)
accessVar("blatna")
selectSubset("SiO2>50")
DebonQB

Q vs. B diagram (Debon + Le Fort 1983)

Description

Assigns data for Debon & Le Fort’s Q vs. B diagram into Figaro template (list ‘sheet’) and appropriate values into ‘x.data’ and ‘y.data’.

Usage

DebonQB()

Details

The Q vs. B diagram was proposed by Debon and Le Fort (1983) for aluminous associations. It defines three associations, reflecting maficity of samples: leucocratic, subleucocratic and mesocratic.

Parameters for the diagram are calculated by the function ‘DebonCalc’. All of them are based on millications (1000 gram-atoms per 100 grams).

Q = Si / 3 - (K + Na + 2 * Ca / 3) [cationic proportion of quartz]
B = Fe + Mg + Ti [maficity]

For details, see Debon & Le Fort (1983) and (1988).
Value

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sheet</td>
<td>list with Figaro Style Sheet data</td>
</tr>
<tr>
<td>x.data</td>
<td>Q value. See details.</td>
</tr>
<tr>
<td>y.data</td>
<td>B value. See details.</td>
</tr>
</tbody>
</table>

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

classify figaro plotDiagram DebonPQ DebonBA DebonKNaB DebonBMgNo DebonBQF DebonCalc

Examples

data(blatna)
accessVar("blatna")
selectSubset("SiO2>50")
plotDiagram("DebonQB",FALSE)
figCol("red")

deleteSingle

Description

Deletes a single numeric variable or a label.

Usage

deleteSingle()

Details

The variables to be deleted is selected using the function 'selectColumnLabel'. In any case, a confirmation is required before a variable is deleted from the system. Note that some variables are required by the system and cannot be deleted.

Value

Returns the corrected version of the data frame 'labels' or numeric matrix 'WR'.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>
Description

This function serves for importing the whole-rock geochemical data from EarthChem.org online database. Since 2010, EarthChem has been part of IEDA (Integrated Earth Data Applications), the National Science Foundation (NSF)-funded data facility for solid earth geoscience data. The data are gathered from several publicly available databases such as PetDB, SedDB, NA VDAT, or are contributed, and can be accessed through the EarthChem’s REST Search Service utilized by our function.

Usage

EarthChem(x)

Arguments

x  a list of parameters, given below, detailed account of which is to be found at the web page with the EarthChem REST Server Documentation. See also Examples.

Details

The function EarthChem imports the specified data taking advantage of the EarthChem REST Search Service, which accepts GET string variables that determine search criteria. The results are returned using html or xml protocols, and can be then imported to the GCDkit using the library 'XML'.

Possible parameters are (in square brackets are GCDkit default values)

Data source specification (reference(s))
author author

title title of the article

journal journal

doi Digital Object Identifier (not always available)

minpubyear minimum publication year of the citation article

maxpubyear maximum publication year (reqd with the former option)

exactpubyear exact year of publication

keyword free-text generic descriptor field

Sample ID, location or age

sampleid sample number/identifier from the original database

polygon geographic region, specified by geographic coordinates

north, east, south, west coordinates of a geographic envelope, all to be provided together

minage minimum age of the sample (Ma)

maxage maximum age of the sample (Ma)

exactage age of the sample (Ma)

gеologicalage geological age

material either ‘bulk’, ‘whole rock’, ‘glass’ or ‘inclusion’

Output format

searchtype type of search, only 'rowdata' (table of items matching the criteria) is implemented so far [rowdata].
outputtype: either 'html' or 'xml' [html]
outputlevel: either 'sample' or 'method' [sample]
startrow: sequence number of the first output row minus 1 [0]
endrow: sequence number of the last output row minus 1 [number_of_hits-1]
standarditems: logical; output just the standard items? [yes]
outputitems: comma-separated list of output items
showcolumnnames: logical; import the names of variables? [yes]

Value

(In invisibly) the query string. If no hits were found, the function returns (again invisibly) the value -1.

Imports the data into the GCDkit system. Stores the imported dataset into memory (i.e., the variable WRCube) together with the fields source that contains the string "EarthChem" and date with time of the search, EarthChem.query.var with the query variable (a list) and EarthChem.query.url with URL sent to the web service.

Then, the previously active dataset becomes the current one.

Warning

XML library is required.

Author(s)

Function by Vojtech Janousek, <vojtech.janousek@geology.cz> (with helpful assistance from Jason Ash, <jasonash@ku.edu>).
Tcl/Tk GUI by Oscar Laurent, <oscar.laurent@erdw.ethz.ch>
The XML package was written by Duncan Temple Lang.

See Also

For further details, see the EarthChem REST Server Documentation.

Examples

# Some of these examples are based on original examples
# from http://ecp.iedadata.org/rest_search_documentation

EarthChem(list(author="smith",outputtype="html",showcolumnnames="yes",startrow=0,endrow=100, outputitems="sample_id,source,longitude,latitude"))

EarthChem(list(author="janousek",outputtype="xml",showcolumnnames="yes",outputitems="sample_id, source,longitude,latitude",standarditems="yes"))

EarthChem(list(author="moyen",outputtype="html",showcolumnnames="yes",outputitems= "sample_id,source,longitude,latitude",standarditems="yes"))

EarthChem(list(geologicalage="cambrian",outputtype="html",startrow=0,endrow=100, outputitems="sample_id,source,longitude,latitude"))

# Read a map directly into R
#shell.exec(query) # easy solution, gets to browser
filename<-paste(gcdx.dir,"map.jpg",sep="/")
download.file(query,filename,method="internal",mode="wb")
shell.exec(filename)

---

**Edit labels**

**Description**

Simultaneous editing of all labels using a spreadsheet-like interface.

**Usage**

`editLabels()`

**Arguments**

none.

**Details**

The function invokes a spreadsheet-like interface that enables the user to edit the labels for individual samples. When all the desired changes have been performed, close button is to be clicked.

**Value**

Returns the corrected version of the data frame 'labels'.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

`data.entry`
**Description**

Simultaneous editing of all numeric data using a spreadsheet-like interface.

**Usage**

```r
editData(x = WR)
```

**Arguments**

- `x` data frame/numeric matrix to be edited; default is `WR`, i.e. numeric data

**Details**

The function invokes a spreadsheet-like interface that enables the user to edit the numeric data for individual samples. When all the desired changes have been performed, close button is to be clicked.

The system then performs some recalculations as if the data set was loaded from the disc afresh (calling `Gcdkit.r`).

**Value**

Returns the corrected version of the numeric matrix `WR`.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

`data.entry`

---

**Description**

Global replacement each of the discrete values (levels) for a selected label.

**Usage**

```r
editLabFactor()
```
**Details**

The label to be edited is selected using the function `selectColumnLabel`.
Then the function invokes a spreadsheet-like interface that enables the user to overwrite directly any of the discrete values for the given label, in the R jargon called `levels`. When all the desired changes have been performed, close button is to be clicked.

**Value**

Returns the corrected version of the data frame labels.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

`data.entry`

---

**elemIso**  
*Binary plot of a WR geochemical parameter vs isotopic ratio*

**Description**

Plots a diagram of a selected whole-rock geochemical parameter vs initial Sr isotopic ratios or initial $\epsilon(Nd)$ for selected samples.
Usage

elemIso(xlab=NULL, what=NULL, GUI=FALSE, ...)

Arguments

- **xlab**: variable name or a formula for the x axis; if NULL a dialogue is displayed
- **what**: name of the desired isotopic parameter
- **GUI**: logical; is the function called from the GUI?
- **...**: optional parameters to the underlying function `{plotWithLimits}`

Details

The variable to be plotted as x axis is based on whole-rock geochemical data. If not specified upon the function call, it is selected using the function `selectColumnLabel`. In the specification of the variable can be used also an arithmetic expression, see `calcCore` for the correct syntax.

The plotted isotopic parameters (y axis) can be one of:

<table>
<thead>
<tr>
<th>Menu item</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>87Sr/86Sr</td>
<td>Initial Sr isotopic ratios</td>
</tr>
<tr>
<td>143Nd/144Nd</td>
<td>Initial Nd isotopic ratios</td>
</tr>
<tr>
<td>EpsNd</td>
<td>Initial $\epsilon$($Nd$) values</td>
</tr>
<tr>
<td>1 stg DM model ages (Goldstein et al. 1988)</td>
<td>Single-stage DM Nd model ages</td>
</tr>
<tr>
<td>1 stg DM model ages (Liew &amp; Hofmann 1988)</td>
<td>Single-stage DM Nd model ages</td>
</tr>
<tr>
<td>2 stg DM model ages (Liew &amp; Hofmann 1988)</td>
<td>Two-stage DM Nd model ages</td>
</tr>
</tbody>
</table>

If called from GUI, the samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSamples` for details.

Value

None.

Plugin

SrNd.r

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

The actual plotting is done by the function `plotWithLimits`. 
epsEps

Description

Plots a diagram of initial $^{87}\text{Sr}/^{86}\text{Sr}$ ratios vs. initial $\epsilon(\text{Nd})$ values for selected samples.

Usage

epsEps(GUI=FALSE,...)

Arguments

GUI logical; is the function called from the GUI?
...
optional parameters to the underlying function (plotWithLimits)
Details

If in GUI, the samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSamples` for details.

Value

None.

Plugin

SrNd.r

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

The actual plotting is done by the function `plotWithLimits`.

Description

This function serves for exporting the specified data into MDB (MS Access) format (via the ODBC interface).

Usage

```r
accessExport(what=cbind(labels, WR), tablename=NULL, transpose=FALSE, dec.places=NULL)
```

Arguments

- `what`: a matrix, data frame or a list
- `tablename`: name of the data table
- `transpose`: logical; transpose the data?
- `dec.places`: numeric; number of decimal places

Details

The function `accessExport` outputs the specified data via Microsoft’s ODBC interface, taking an advantage of the library ‘RODBC’. Unlike for the function ‘excelExport’, ODBC makes possible opening a new file.

If the argument ‘what’ is a matrix or data frame, the name of the table can be specified using the optional parameter ‘tablename’.

For a list, several tables are created, their number and names corresponding to the items present.
**Export to DBF**

**Value**

None.

**Warning**

This function is not available on 64-bit systems!

**Author(s)**

The RODBC package was written by Brian Ripley.
Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

'excelExport' 'dbfExport'

**Examples**

```r
accessExport(results) # Saves the last calculated results
```

---

**Description**

This function serves for exporting the specified data into DBF (dBase III) format (using the function 'write.dbf' of the package 'foreign').

**Usage**

```r
dbfExport(what=cbind(labels,WR), transpose=FALSE)
```

**Arguments**

- **what**: a matrix or data frame
- **transpose**: logical; transpose the data frame?

**Details**

The function dbfExport outputs the specified data. Note that it cannot handle lists.

**Value**

None.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

'write.dbf' 'excelExport' 'accessExport'
Export to Excel

Examples

dbfExport(results) # Saves the last calculated results

Export to Excel

Description

This function serves for exporting the specified data into XLS or XLSX (MS Excel) formats (via the ODBC interface).

Usage

excelExport(what=cbind(labels, WR), tablename =NULL, transpose=FALSE, dec.places=NULL)

excel2007Export(what=cbind(labels, WR), tablename =NULL, transpose=FALSE, dec.places=NULL)

Arguments

what a matrix, data frame or a list
tablename name of the data sheet
transpose logical; transpose the data?
dec.places numeric; number of decimal places

Details

The functions excelExport and excel2007Export output the specified data via Microsoft’s ODBC interface, taking an advantage of the library ‘RODBC’.

If the argument ‘what’ is a matrix or data frame, the name of the sheet can be specified using the optional parameter ‘tablename’.

For a list, several sheets are attached, their number and names corresponding to the items present.

Value

None.

Warning

These functions are not available on 64-bit systems!

Unfortunately the way the ODBC is programmed by Microsoft does not make opening a new Excel file possible. Thus only adding new sheet(s) to a pre-existing spreadsheet file is feasible.

Author(s)

The RODBC package was written by Brian Ripley.

Vojtech Janousek, <vojtech.janousek@geology.cz>
See Also

'accessExport' 'dbfExport'

Examples

    excelExport(results) # Saves the last calculated results in XLS format
    excel2007Export(results) # Saves the last calculated results in XLSX (or XLS) format

Description

Outputs the specified data with (optional) labels into HTML. This format is useful for importing into spreadsheets, word processors or publishing on the WWW.

Usage

    HTMLTableMain(what,digits=2,desc=NULL,title=" ",sum.up=FALSE,open=TRUE,
        close=TRUE,browse=TRUE,filename=paste(data.dir,"R2HTML/htmltable" ,sep="/" ),
        rotate=FALSE)

    HTMLtableOrdered(what,which=rownames(what),labs=labels,digits=2,desc=NULL,
        title=" ",sum.up=FALSE,key1=NULL,key2=NULL,
        filename=paste(data.dir,"R2HTML/htmltable" ,sep="/" ),split.by=25,rotate=TRUE)

    HTMLTableWR(filename="htmltable")

    HTMLTableResults(filename="htmltable")

Arguments

    what    numeric matrix; data to be exported
    digits  required precision
    desc    name of the columns within 'labels' to be attached to the table
    title   main title
    sum.up  logical; should be a sum calculated?
    open    logical; should be opened a new HTML file?
    close   logical; should be the HTML file closed when finished?
    browse  logical; should be the HTML file finally opened in the default browser?
    filename optional name for the file produced
    rotate  logical, should be the table transposed, with samples in columns and variables in rows?
    which   (optional) sample names in numeric matrix 'what' for the output
    labs    name of variable with textual labels
    key1    is a variable in numeric matrix 'what'
    key2    is a grouping information (name of a column in 'labs')
    split.by maximal number of data columns per page
Details

HTMLTableWR and HTMLTableResults are GUI front ends to HTMLTableMain, the former enabling the user to choose samples (rows) and columns for the output using the searching mechanisms common in the GCDkit.

HTMLTableWR outputs the numeric data (with optional labels and sum) stored in the data matrix 'WR'.

HTMLTableOrdered also outputs the numeric data stored in the numeric matrix specified by parameter 'what'. Optional argument 'which' gives the list of sample names (rows) in the matrix to be saved. The data are first sorted based on 'key2', which typically gives a grouping information (name of a column in 'labs'). Within each of the groups, the data are further sorted based on the numeric variable 'key1'. See example.

HTMLTableResults outputs the results of the most recent calculation (with optional labels and sum) as stored in the variable 'results'.

The plugin attempts to format sub- and superscripts in the names of variables.

The created file 'filename' is placed in the subdirectory 'R2HTM' of the current working directory; when finished, it is previewed in a browser. The style for the table is determined by the cascade style file R2HTML.css in the subdirectory 'Plugin'.

Value

None.
Warning

The plugin uses R2HTML library, which must be downloaded from CRAN and properly installed. Its presence is checked before the code is executed.

Author(s)

The R2HTML package was written by Eric Lecoutre.
Vojtech Janousek, <vojtech.janousek@geology.cz>

Examples

# Works on the 'sazava' test data set
setwd(paste(gcdx.dir,"Test_data",sep="/"))
loadData("sazava.data")

HTMLTableMain(WR[,c("SiO2","MgO","FeOt")],digits=2,desc="Intrusion",title="Sazava [wt.%]")

HTMLtableOrdered(WR[,LILE],digits=1,key1="SiO2",key2="Intrusion",title="Large Ion Lithophile Elements (ppm)",split.by=3)

F-M-W diagram

Ohta + Arai (2007) FMW weathering index

Description

Assigns data for the F-M-W diagram by Ohta & Arai (2007) into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

This function is meant to be used with plotDiagram.

Usage

OhtaArai(fixTi=F)

Arguments

fixTi logical, if TRUE, the TiO2 value of samples where this is not determined (or 0) will be replaced by FeOt/7.

Details

This (ternary) diagram has been proposed by Ohta & Arai (2007) to identify chemically weathered igneous rocks. It turns out to be also very useful to separate para- from orthogneisses (Moyen et al. 2017).

The values of the apices are complex combination of oxides defining three end-members: M (mafic igneous rocks), F (felsic igneous rocks) and W (chemical weathering):

\[
M = \exp \left( -0.395 \times \ln(SiO_2) + 0.206 \times \ln(TiO_2) - 0.316 \times \ln(Al_2O_3) \\
+ 0.160 \times \ln(Fe_2O_3t) + 0.246 \times \ln(MgO) + 0.368 \times \ln(CaO) \\
+ 0.073 \times \ln(Na_2O) - 0.342 \times \ln(K_2O) + 2.266 \right)
\]
\[ F = \exp \left( 0.191 \times \ln(SiO_2) - 0.397 \times \ln(TiO_2) + 0.020 \times \ln(Al_2O_3) \\
- 0.375 \times \ln(Fe_2O_3t) - 0.243 \times \ln(MgO) + 0.079 \times \ln(CaO) \\
+ 0.392 \times \ln(Na_2O) + 0.333 \times \ln(K_2O) - 0.892 \right) \]

\[ W = \exp \left( 0.203 \times \ln(SiO_2) + 0.191 \times \ln(TiO_2) + 0.296 \times \ln(Al_2O_3) \\
+ 0.215 \times \ln(Fe_2O_3t) - 0.002 \times \ln(MgO) - 0.448 \times \ln(CaO) \\
- 0.464 \times \ln(Na_2O) + 0.008 \times \ln(K_2O) - 1.374 \right) \]

**Value**

A list containing the Figaro template. In addition the following global variables are modified:

- `sheet` list with Figaro Style Sheet data
- `x.data` x coordinates in ternary projection
- `y.data` y coordinates in ternary projection

**Author(s)**

Jean-Francois Moyen, <jfmoyen@gmail.com>

**References**


See Also

figaro plotDiagram

Examples

# plot the diagram
# assuming a dataset is loaded, of course!
## Not run:
plotDiagram("OhtaArai",FALSE)
## End(Not run)

FeMiddlemost

Adjustment of Fe oxidation ratio (Middlemost 1989))

Description

Auxiliary function performing adjustment of the iron-oxidation ratio as proposed by Middlemost (1989).

Usage

FeMiddlemost(anhydrous = TRUE)

Arguments

anhydrous logical; should be returned major-element analyses recast to anhydrous basis?

Details

This function performs an adjustment of the iron-oxidation ratio for individual volcanic rock types as proposed by Middlemost (1989).

The classification is based on TAS classification (Le Bas et al. 1986, Le Maitre et al. 1989).

The $Fe_2O_3/FeO$ ratios for individual rock types, based on Verma et al. (2002) (Fig. 1), are as follows:

- foidite, $Na_2O + K_2O \leq 3$: 0.15
- foidite, $3 < Na_2O + K_2O \leq 7$: 0.2
- foidite, $7 < Na_2O + K_2O \leq 10$: 0.3
- foidite, $Na_2O + K_2O > 10$: 0.4
- picrobasalt: 0.15
- basalt: 0.2
- basaltic andesite: 0.3
- andesite: 0.35
- dacite: 0.4
- rhyolite: 0.5
- trachybasalt: 0.3
basaltic trachyandesite 0.35  
trachyandesite 0.4  
trachyte/trachydacite 0.5  
tephrite/basanite, $Na_2O + K_2O <= 6$ 0.2  
tephrite/basanite, $Na_2O + K_2O > 6$ 0.3  
phonotephrite 0.35  
tephriphonolite 0.4  
phonolite 0.5

If the parameter ‘anhydrous’ is set, returned are the major-element data recast to 100 % anhydrous basis.

**Value**

A matrix with adjusted whole-rock chemical data.

No permanent changes to either ‘WR’ or ‘WRanh’ are made.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**References**


**See Also**

TAS Verma

---

**figAdd**  
Plot editing: Add

**Description**

These functions enable adding new components to Figaro-compatible plots.
Usage

figTicks(major=-0.5, minor=0.25, x mj r=NULL, x m jr=NULL, ymj r=NULL, ymin=NULL)

figGrid(x.int=NULL, y.int=NULL, lty="dotted", col="gray30", GUI=FALSE)

figLegend()

figAddReservoirs(autoscale=FALSE, var.name=NULL, sample.names=NULL, reserv.condition=NULL, labs=NULL, pch="*", col="darkred", cex=1, type="p",...)

figAddText()

figAddArrow()

figAddBox()

figAddFit()

figAddCurve(equation=NULL)

Arguments

major length of the major tick marks.

minor length of the minor tick marks.

x mj r, ymj r intervals for the major tick marks.

xmin, ym in intervals for the minor tick marks.

x.int intervals for the grid, x axis component.

y.int intervals for the grid, y axis component.

GUI logical; is the function called from GUI?

lty grid line type.

col plotting colour.

autoscale logical; should be the scaling changed so that all the plotted data fit in?

var.name text; either 'reservoirs.data', 'idealmins.data' or a name of a global variable. See Details.

col

sample.names character vector; names of reservoirs, ideal minerals or samples to be plotted.

reserv.condition text; regular expression specifying reservoirs compositions of which are to be plotted.

labs text; optional abbreviated labels for the individual reservoirs

pch plotting symbol.

cex numeric; relative size of the plotting symbol.

t ype character; plot type; see plot.default.

... additional parameters to the plotting function. See figOverplot.

equation text; equation expressed as a function of x; see curve.
Details

'figTicks' adds major and minor tick marks for the x and y axes. Their length is specified as a fraction of the height of a line of text. Negative numbers imply outward and positive inward pointing ticks. The user is prompted for four numbers separated by commas, xmr, xmin, ymin, ymr. These specify the intervals of major and minor ticks for x and y axes, respectively. Not implemented to logarithmic plots and spiderplots yet.

'figGrid' adds grid lines for x and/or y axes.

'figLegend' adds legend(s) on specified location.

'figAddReservoirs' overplots compositions of selected geochemical reservoirs (taken from the file 'reservoirs.data', see selectNorm for the file structure as well as relevant references) or ideal minerals (from the file 'idealmins.data'). Alternatively, if the name of a numeric matrix or dataframe in the global environment is provided via the argument 'var.name', the selection of data from this object is used (see Examples). The selection is specified by either 'sample.names' or by 'reserv.condition' parameters. Optional argument 'labs' can specify alternative, perhaps abbreviated textual labels to the points plotted.

Please note that the function 'figAddReservoirs' is available so far for simple spiderplots, binary and ternary plots only. Technically, the function invokes 'figOverplot' setting just.draw=FALSE, and thus the overplotted dataset is added permanently. If just.draw=FALSE, the points for the reference dataset do not become a part of the template, and thus will vanish upon redrawing, zooming .... See Examples.

'figAddText' adds text on specified location. The parameters are the text style ('n' = normal, 'b' = bold, 'i' = italic and 'bi' = bold italic), colour and relative size.

'figAddArrow' adds arrow on specified location. The parameters are colour and line style ('solid', 'dashed', 'dotted' and 'dotdash').

'figAddBox' adds box on specified location (click bottom left and then top right corner).

'figAddFit' adds either a single least-squares fit to all data, or several fit lines, for each of the groups separately. The parameters are colour and line style ('solid', 'dashed', 'dotted' and 'dotdash'). The equation of each fit line is plotted at the user-defined location.

'figAddCurve' adds a curve, specified as a function of variable 'x'. The parameters are colour and line style ('solid', 'dashed', 'dotted' and 'dotdash').

The colours can be specified both by their code (see table under menu 'Data handling|Show available colours') or R name (see Examples).

The additional two menu items, available for binary and ternary plots, allow adding contours or convex hulls outlining individual groups of data. See contourGroups and chullGroups.

Value

For 'figAddReservoirs', a numeric matrix with the overplotted analyses from the reference dataset.

Warning

Most of these functions serve to adding some extra components/annotations immediately before the graph is printed/exported. Note that, except for 'figAddReservoirs', all user-defined components added via 'Plot editing: Add' will be lost upon redrawing, zooming ....

Author(s)

Colin M. Farrow, <colinfarrow537@gmail.com>

Vojtech Janousek, <vojtech.janousek@geology.cz>
See Also

par showColours colours figaro selectNorm
countGroups chullGroups
figOveplot figOveplotDiagram overplotDataset curve

Examples

## figTicks and figGrid
data(blatna)
accessVar("blatna")
setCex(1.5)

binary("Zr/Nb","Ba/La")
figTicks(major=-0.5, minor=0.25,10,1,10,1)
figGrid(,5,col="darkblue") # just y axis (second parameter)

figRedraw()
figGrid(2,5,col="darkblue")

## figAddReservoirs
data(blatna)
accessVar("blatna")
setCex(1.5)

# binary
binary("Zr/Nb","Ba/La")
# Sun & McDonough mantle reservoirs, Taylor & McLennan 1995 Upper and Lower Crust
reserv.names<-c("NMORB","EMORB","OIB","UCC","LCC")
figAddReservoirs(TRUE,"reservoirs.data",reserv.condition=reserv,labs=reserv.names)

# ternary
ternary("SiO2/10","MgO","FeOt")
figAddReservoirs(var.name="idealmins.data",sample.names=c("Or","Bt","Ph"))

# spider
spider(WR,"NMORB..Sun",field=TRUE,colour="gray",field.colour=TRUE,ymin=0.1,ymax=100)
figAddReservoirs(var.name="reservoirs.data",reserv.condition="Continental Crust", autoscale=TRUE,col=c("red","black","darkblue"),pch=1:3)

# Calculate Rayleigh-type fractionation trend
ff<-seq(1,0.1,-0.1) # F, amount of melt left
x<-80*ff^(1.2-1) # cL for three elements, arbitrary D of 1.2, 2.0 and 1.3
y<-550*ff^(2.0-1)
z<-1000*ff^(1.3-1)
my.trend<-cbind(x,y,z)
colnames(my.trend)<-c("Rb","Sr","Ba")
rownames(my.trend)<-ff

# By default, the overplotted information is added permanently
binary("Rb","Sr",log="xy")
figAddReservoirs(var.name="my.trend",pch=","col="blue",autoscale=TRUE,type="o", labs=rownames(my.trend))
figXlim(c(10,500))
figaro.identify

Plot editing: Identification of plotted symbols

Description

These functions allow the user to identify points in Figaro-compatible plots.

Usage

figIdentify()

highlightSelection()

Details

'figIdentify' identifies points closest to a mouse pointer, if a mouse button is pressed. For binary and ternary plots, sample names are plotted; for spider plots the function prints the sample name, concentration of the given element (in ppm) and highlights the whole pattern. The identification is terminated by pressing the right button and selecting 'Stop' from the menu.

'highlightSelection' allows the selected analyses to be highlighted. The samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see selectSubset for details.

If the search results are empty or embrace all samples, the user is given a chance to select the samples from the list of their names. Press Ctrl+click to select multiple ones.

For binary and ternary plots, Press Esc in the Console window to stop the points blinking. In spider plots are shown overall ranges of normalized concentrations (by a gray field) with superimposed patterns for selected samples.

Author(s)

Colin M. Farrow, <colinfarrow537@gmail.com>

and Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

identify selectSubset 'figaro'
Description

These functions enable altering colours for titles or all plotting symbols in Figaro-compatible plots.

Usage

```r
figCol(col=NULL)
figColMain(col=NULL)
figColSub(col=NULL)
figBw()
```

Arguments

- `col` colour specification

Details

The colours can be specified both by their code (see table under Data handling|Show available colours) or R name (see Examples).

- `figBw` sets the whole plot (main title and subtitle, axes and plotting symbols) in black and white, making them ready for printing/exporting.

Author(s)

Colin M. Farrow, <colinfarrow537@gmail.com>
& Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

- `showColours`
- `colours`
- `figaro`

Examples

```r
colours() # prints the list of available colour names

plotDiagram("TAS",FALSE) # example of a classification plot
figSub(txt="My TAS diagram")
figCol(col="green")
figColMain(col="red")
figColSub(col="blue")

figBw()

spider(WR,selectNorm("Boynton"),0.1,1000,pch=labels$Symbol,col=labels$Colour)
figMain(txt="My REE plot")
figSub(txt="Normalized by Boynton (1989)")
```
figCol(col="green")
figColMain(col="red")
figColSub(col="blue")

**figEdit**  \hspace{1cm} *Plot editing: Changing titles and axis labels*

**Description**

These functions enable altering titles and axis labels of binary (figXlab, figYlab) and ternary (figAlab, figBlab, figClab), Figaro-compatible plots.

**Usage**

```r
figMain(txt=NULL)
figSub(txt=NULL)
figXlab(txt=NULL)
figYlab(txt=NULL)
figAlab(txt=NULL)
figBlab(txt=NULL)
figClab(txt=NULL)
```

**Arguments**

- `txt` text

**Details**

If specified, the parameter `txt` will be passed to the function `annotate` to guess the correct reformatting to sub- and superscripts for production of "publication quality" plots. Otherwise, the current value (titles or labels for axes/apices) are edited.

**Author(s)**

Colin M. Farrow, <colinfarrow537@gmail.com>
and Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

*annotate*
*figaro*
Examples

plotDiagram("TAS",FALSE) # example of a classification plot
figMain(txt="My TAS diagram")
figSub(txt="test")
figXlab(txt="Silica")
figYlab(txt="Total alkalis")

Description

Interactive definition of groups on any Figaro-compatible plot.

Usage

figGbo(x.tol = 0, y.tol = 0, max.points = 100, max.polygons = 25)

Arguments

x.tol, y.tol  tolerance for the automatic closing of polygons.
max.points   maximum number of vertices for a single polygon.
max.polygons maximum number of polygons.

Details

Each of the groups is defined by clicking vertices of a polygon with the corresponding data points. The polygons are closed automatically. To finish, right click anywhere on the plot and select 'Stop'. The groups are numbered consecutively, points falling into two or more fields form extra groups, as do unclassified samples.

Author(s)

Vojtech Erban, <erban@sopky.cz> & Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

'figaro'
figLoad

Loading a Figaro plot

Description

Loads a Figaro-compatible plot (both the template and the data) stored in a file.

Usage

figLoad()

Arguments

None.

Details

The default suffix for the saved diagrams is 'fgr'. Note that only the data needed for the plotting ('x.data', 'y.data') are stored in the 'fgr' files. Thus the data set currently in memory (e.g., variables 'WR', 'labels', . . . ) is unaffected by the function 'figLoad'.

Author(s)

Colin Farrow, <colinfarrow537@gmail.com>
and Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

figSave figaro

figMulti

Figaro: Multiple plot by groups

Description

Displays multiple plots, for each of the groups one, based on a most recently plotted Figaro-compatible template. For spiderplots, the colour field denotes the total variation with the whole dataset.

Usage

figMulti(x=x.data,y=y.data,nrow=NULL,ncol=NULL,xlab=sheet$demo$call$xlab, ylab=sheet$demo$call$ylab,pch=NA,col=NA,cex = NULL,plot.symb=NA,shaded.col="gray",rotate.xlab=TRUE, offset=TRUE,centered=FALSE,title=NULL,...)
Arguments

\( x, y \) data to be plotted

\( nrow, ncol \) dimensions of the plots’ matrix

\( xlab, ylab \) labels for the axes

\( pch \) plotting symbols

\( col \) plotting colours

\( cex \) relative size of the plotting symbols

\( plot.symb \) logical, spiders. Shall be shown also plotting symbols or just lines?

\( shaded.col \) (spiders) Colour for the field portraying the overall variability in the dataset.

\( rotate.xlab \) logical, spiders. Shall be the element names on x axis rotated?

\( offset \) logical, spiders. Shall be the names for odd and even elements shifted relative to each other?

\( centered \) logical, spiders. Shall be the element names on x axis plotted in between tick marks?

\( title \) optional title for the whole plate. If not provided, it is taken from the title of the Figaro template.

\( ... \) any additional graphical parameters

Note

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots. See Plate, Plate editing and figaro for details.

Details

The function can handle any Figaro-compatible plots, including binary, ternary or spiderplots. For classification plots, it may be advantageous to switch off the field names using the function 'plateAnnotationsRemove', (see the figure below as well as the Examples).
**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

and Colin M. Farrow, <colinfarrow537@gmail.com>

**See Also**

figaro, Plate, Plate editing binary, ternary, spider

**Examples**

# Note that groups should have been defined before running these.

# switch on the field names (default, valid globally for the whole system)
options("gcd.plot.text"=TRUE)
```r
plotDiagram("TAS",FALSE)
figMulti()

# switch off the field names
options("gcd.plot.text"=FALSE)
plotDiagram("LarochePlut",FALSE)
figMulti(col="black",pch="x",cex=2)
options("gcd.plot.text"=TRUE)

spider(WR,selectNorm("Boynton"),0.1,1000,pch=labels$Symbol,
col=labels$Colour,cex=labels$cex)
figMulti(plot.symb=TRUE,cex=2)

figMulti(col="red",plot.symb=FALSE,rotate.xlab=FALSE,offset=TRUE)
figMulti(col="red",plot.symb=FALSE,rotate.xlab=FALSE,offset=FALSE,shaded.col="khaki")
```

---

**figOverplot**  
*Overplotting data onto pre-existing binary, ternary or spider plots*

**Description**  
This function allows overplotting new data points onto Figaro-compatible binary or ternary plots, or patterns onto spiderplots. It is most useful in adding selected data from typical geochemical reservoirs (e.g., Upper Continental Crust, MORB ...), ideal mineral compositions, results of petrogenetic modelling or just another dataset used for comparison (any of these will be henceforth referred to as a reference dataset).

**Usage**  
```
figOverplot(var.name, mat=NULL, sample.names=NULL, condition=NULL,
labs=NULL, autoscale=FALSE, pch="x", col="darkred", cex=1,
type="p", just.draw = FALSE, overplotDataset = FALSE,...)
```

**Arguments**  
- **var.name**: either `reservoirs.data`, `idealmins.data` or a quoted name of a global variable.
- **mat**: matrix with data for all reservoirs available for overplotting. Meant mainly for internal use of the `GCDkit` system.
- **sample.names**: character vector; list of names of desired reservoirs, ideal minerals or samples in the reference dataset to be overplotted.
- **condition**: text; regular expression specifying names of desired reservoirs, ideal minerals or samples in the reference dataset.
- **labs**: text; optional (typically abbreviated) labels for the overplotted data from the reference dataset.
- **autoscale**: logical; should be the scaling changed so that all the plotted analyses fit in?
- **pch**: plotting symbol(s) for the reference dataset.
- **col**: plotting colour(s) for the reference dataset.
- **cex**: numeric; relative size of the plotting symbol(s) for the reference dataset.
figOverplot

- **type**: character; plot type; see `plot.default`. For obvious reasons, not implemented for spiderplots.
- **just.draw**: logical; if FALSE, the overplotted bit is added permanently, i.e. the Figaro template is also affected.
- **overplotDataset**: logical; for internal use by the system only.
- ... additional parameters to the underlying plotting function(s). See Details.

### Details

If called directly, the function is employed to overplot data from a reference dataset, either real-world data or a numeric matrix spanning, for instance, from petrogenetic modelling. The data originate from a two-dimensional variable in the global environment, whose name is provided via the obligatory argument `var.name`.

Argument `mat` is meant for internal use by the system and does not need to be specified by the user as the data frame/matrix `mat` is generated automatically by the function `figOverplot`.

In both cases, the selection from the numeric matrix or dataframe `mat` is based on a list of desired `sample.names` or on a regular expression yielding their subset (`condition`). Of course, from this selection, only analyses with data sufficient to be plotted on the current diagram are used.

If neither `sample.names` nor `condition` is provided, all samples are shown.

For plotting are used functions `points`, `triplotadd` and `spider` for binary plots, ternary plots and spiderplots, respectively. Argument `...` can supply additional parameters to these low-level plotting functions.

Optional parameter `labs` can specify alternative, typically abbreviated textual labels to the points plotted.

Logical argument `autoscale` determines whether the plot should be rescaled to accommodate both the original data points and the reference dataset. Clearly, it does not make sense for a ternary plot.

By default, the overplotted information is added permanently but this behaviour is controlled by the argument `just.draw`.

### Value

A numeric matrix with the overplotted analyses from the reference dataset.

### Note

Within the `GCDkit` system, this function is invoked by `figAddReservoirs` to overplot selected compositions from typical geochemical reservoirs (system file `reservoirs.data`) or chemistries of ideal minerals (system file `idealmins.data`).

### Warning

If `just.draw=FALSE`, the points for the reference dataset do not become a part of the template, and thus will vanish upon redrawing, zooming ... See Examples.

### Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>
figOverplotDiagram

Overplotting data onto classification or geotectonic plots

Description

This function allows overplotting new data points onto single Figaro-compatible templates defined for classification or geotectonic plots (binary or ternary, designed as stand alone or extracted from plates).

Usage

figOverplotDiagram(overplot.dataset, bg.dataset=NULL, diagram=NULL, which=NULL, xlim=NULL, ylim=NULL, pch="*", col="darkred", cex=1, labs=NULL, type="p", lwd=1, lty="solid", transp=0, just.draw=TRUE, source.first = TRUE, source.plugins = TRUE,...)
Arguments

- **overplot.dataset**
  (obligatory) name of the main (foreground) dataset stored in memory, or global variable name.

- **bg.dataset** (optional) name of the background dataset stored in memory.

- **diagram** character; existing diagram name.

- **which** which plot is to be extracted (if belonging to a plate)?

- **xlim** new limits of the x axis.

- **ylim** new limits of the y axis.

- **pch** plotting symbol(s) for the foreground dataset.

- **col** plotting colour(s) for the foreground dataset.

- **cex** numeric; relative size of the plotting symbol(s) for the foreground dataset.

- **labs** text; optional labels for the overplotted data.

- **type** character; see 'points'.

- **lwd, lty** parameters for connecting line, if drawn; see 'par'.

- **transp** numeric; transparency for the background set, 0-1.

- **just.draw** logical; NOT FUNCTIONAL, kept just for compatibility sake.

- **source.first** logical; should be also the .First function sourced upon loading the new dataset?

- **source.plugins** logical; indicates whether all plugins should be sourced upon loading the new dataset.

- **...** additional parameters to the underlying plotting function(s). See Details.

Details

The function 'figOverplotDiagram' can be employed in two ways.

If quoted names of two datasets in memory are provided ('bg.dataset' and 'overplot.dataset'), a new plot is created, whereby the background dataset is plotted using either the function 'plotDiagram' (for stand-alone plots) or 'plateExtract' (for one of diagrams extracted from a plate).

If only a single name of dataset is given, then the data are overplotted onto the current (preexisting) diagram.

Optional plotting parameters 'pch', 'col', 'cex', 'type', 'lwd' and 'lty' can be defined for the overplotted (foreground) dataset.

Argument '...' can supply additional parameters to the original plotting functions (e.g., 'TAS') invoked by 'plotDiagram' or 'plateExtract'.

Value

None.

Warning

This function serves to add extra components/annotations immediately before the graph (a spider-plot, simple binary or ternary plot) is printed or exported. Note that the points for the overplotted dataset are not part of the template, and thus will vanish upon redrawing, zooming ....

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>
See Also

figOverplot figAddReservoirs overplotDataset
derplotDataset plotDiagram plateExtract
figaro par

Examples

data(sazava)
accessVar("sazava")

data(blatna)
accessVar("blatna")
setCex(2)
pokeDataset("blatna", overwrite.warn=FALSE)

## Two datasets
# stand alone plot
figOverplotDiagram("sazava","blatna","DebonBA",pch=15,col="darkred",
cex=2,transp=0.5)

# plateExtract
figOverplotDiagram("sazava","blatna","PearceGranite",col="darkred",
cex=2,transp=0.5,which=2)

## Overplotting on existing plot - plotDiagram
peekDataset("blatna")
plotDiagram("DebonPQ",FALSE,TRUE)
figCex(2)
figRemove()
figOverplotDiagram("sazava",pch=15,col="darkred",cex=2,transp=0.6)

## Overplotting of existing plot - plateExtract
peekDataset("blatna")
plateExtract("PearceGranite",which=2)
figXlim(c(1,100))
figYlim(c(1,300))
figCex(2)
figOverplotDiagram("sazava",pch=15,col="darkred",cex=2,transp=0.6)

figRedraw
Redrawing/refreshing a Figaro plot

Description

This function redraws/refreshes a Figaro-compatible plot.

Usage

figRedraw(x=x.data, y=y.data, zoom=NULL, bw=FALSE, title=NULL)

refreshFig()
figSave

Arguments

- **x**: vector of x coordinates
- **y**: vector of y coordinates
- **zoom**: logical; redraw while zooming?
- **bw**: logical; should be the output black and white?
- **title**: character; optional title for the plotting window.

Warning

Note that all user-defined components added via 'Plot editing: Add' (legend, lines, text, boxes, ...) - will be lost.

Author(s)

Colin M. Farrow, <colinfarrow537@gmail.com>
and Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

figaro

---

**figSave**

*Saving a Figaro plot*

Description

Saves the current Figaro-compatible plot, both the template and the data needed for the plotting ('x.data','y.data').

Usage

```r
figSave()
```

Arguments

None.

Details

The default suffix for the saved diagrams is `fgr`.

Author(s)

Colin M. Farrow, <colinfarrow537@gmail.com>
and Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

figLoad figaro
Description

These functions enable changing a size of titles, axis labels or plotting symbols of Figaro-compatible plots. The size is relative to 1 (the original).

Usage

figCex(x=NULL)
figCexLab(x=NULL)
figCexMain(x=NULL)
figCexSub(x=NULL)

Arguments

x numeric: scaling factor.

Author(s)

Colin M. Farrow, <colinfarrow537@gmail.com>
and Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

'figaro'

Examples

plotDiagram("TAS",FALSE) # example of a classification plot
figSub(txt="My TAS diagram")
figCex(2)
figCexMain(1.5)
figCexSub(0.5)

spider(WR,selectNorm("Boynton"),0.1,1000,pch=labels$Symbol,col=labels$Colour)
figMain(txt="My REE plot")
figSub(txt="Normalized by Boynton (1989)")
figCex(2)
figCexMain(1.5)
figCexSub(0.5)
**figUser**  
*Plot editing: User defined parameter*

**Description**
Enables the power users to modify the plot parameters directly.

**Usage**

```r
figUser(expression=NULL, redraw=TRUE)
```

**Arguments**

- `expression` character; expression to be evaluated
- `redraw` logical; should be modified Figaro template redrawn?

**Details**

The parameters can be specified at the function call. If not, they are chosen by a dialogue. Several of these can be entered simultaneously, as a semicolon delimited list. The most useful might be:

- `main` Main title
- `sub` Sub title
- `xlab` Label of x axis
- `ylab` Label of y axis
- `xlim` Limits for the x axis
- `ylim` Limits for the y axis
- `bg` Colour of background
- `pch` Plotting symbols
- `col` Colour of plotting symbols
- `cex` Relative size of plotting symbols
- `log` Which of the axes is logarithmic? ("", "x", "y" or "xy")

If no parameters are entered, they can be chosen from a list (still experimental!)

**Menu**

Plot editing: User defined parameter

**Warning**

If requesting a logarithmic plot, make sure that the axis ranges are positive. See Examples or invoke menu items 'Plot editing: Scale x axis' and 'Plot editing: Scale y axis'.

**Author(s)**

Colin M. Farrow, <colinfarrow537@gmail.com>

and Vojtech Janousek, <vojtech.janousek@geology.cz>
See Also

par figaro

Examples

plotDiagram("TAS")
figUser()

figUser("pch=1; col=2")
figUser("pch="+""")
figUser("col="darkblue"")
figUser("bg="khaki",cex=1.5") # for camouflage purposes
figUser("main="My plot"; las=2; font.main=4; cex.main=2; col.main="blue"")

figZoom

Plot editing: Zooming

Description

These functions zoom in and out Figaro-compatible plots.

Usage

figZoom()
figUnzoom()
figXlim(range=NULL)
figYlim(range=NULL)
figFixLim(no.action.warn=TRUE)

Arguments

range numeric: two limits, minimum and maximum, for the given axis.
no.action.warn logical: should be a warning shown if there is no action needed?

Details

'figZoom' zooms the specified rectangular area (click bottom left and then top right corner) in a new window. The zoomed area is highlighted in the old window.
'figUnzoom' closes the new window with blown up portion of the plotting window and returns to the original window.
'figXlim' and 'figYlim' allow to change the plotting limits (as a list of two components, separated by commas).
'figFixLim' extends the scales of both axes of a binary plot automatically if necessary to accommodate all the data points.
Warning
If requesting a logarithmic plot, make sure that the axis ranges are positive.

Author(s)
Colin M. Farrow, <colinfarrow537@gmail.com>
and Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also
'figaro'

Examples
## Not run:
# requires a preexisting Figaro-compatible plot
plot.diagram("Shand",select.samples=FALSE)
figXlim(c(0.6,1.2))
figYlim(c(0.8,3))
## End(Not run)

filledContourFig  Filled contours plot

Description
Generates a frequency plot on the basis of the most recently plotted Figaro template.

Usage
filledContourFig(xlab=sheet$demo$call$xlab,ylab=sheet$demo$call$ylab,
xlim=sheet$demo$call$xlim,ylim=sheet$demo$call$ylim,
annotate.fields=FALSE,...)

Arguments
xlab character vector; label for the x axis
ylab character vector; label for the y axis
xlim limits for the x axis
ylim limits for the y axis
annotate.fields logical; should be the plotted fields labeled by their names?
... additional plotting parameters
Details

This is a somewhat modified version of the R function `filled.contour` that produces a frequency plot on the basis of a Figaro template and superimposes, if desired, selected data points.

First the user is prompted, how many intervals should be each of the axes split into. This corresponds to a density of grid, in which are the individual points classified into. Then a colour scheme (palette) can be chosen. Lastly, after the frequency plot is generated, selected analyses can be plotted (‘overplot’). In the latter case, standard GCDkit routine is used to selectSamples.

Value

None.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

`addContours` `selectSubset` `figaro`

---

**Frost**

*Frost et al. (2001)*

---

**Description**

Classification of granitic rocks proposed by *Frost et al (2001).*
Usage

Frost(plot.txt = getOption("gcd.plot.text"),
clssf = FALSE, GUI = FALSE)

Arguments

plot.txt logical, annotate fields by their names?
clssf logical, should the samples be classified?
GUI logical, is the function called from a GUI?

Details

Classification scheme proposed by Frost et al. (2001). It consists of three diagrams:

- \( Fe \) number vs. \( SiO_2 \) (in wt. %). NB that the Fe-number is calculated as weight proportion of \( FeO/(FeO + MgO) \) (or \( FeO_{tot}/(FeO_{tot} + MgO) \)). See also Warning below.

- \( SiO_2 \) (in wt. %) vs. Modified alkali-lime index (MALI), defined as \( Na_2O + K_2O - CaO \) (in wt. %).

- \( SiO_2 \) (in wt. %) vs. Aluminium saturation index (ASI), defined as molecular \( Al_2O_3/(Na_2O + K_2O + CaO - 3:33P_2O_5) \).

  In fact, this is the A/CNK parameter of Shand (1943), corrected for the Ca content in apatite. See also Warning below.

The classification is designed to work both with analyses distinguishing between ferrous and ferric iron (preferred) and those with total iron only. The dialogue box lets the user decide, whether to use the ferrous iron value or the total iron.

Similarly, if some \( P_2O_5 \) concentrations are missing in the dataset, the user is prompted whether the missing values should be replaced with zero. If not, the problematic analyses are not plotted/classified.

The following associations are distinguished:

- ferroan, magnesian
- alkalic, alkali-calcic, calc-alkalic, calcic
- peralkaline, metaluminous, peraluminous

In fact, the ASI vs. A/NK diagram (based on Shand (1943)) was not plotted in the paper, but it replaces the conditions mentioned in the text and is, in our view, more instructive.
Value

The function returns table of calculated coefficients (Fe-Number, MALI, ASI).

There are two values for the ASI: that labeled ’ASI’ is calculated from molecular proportions of oxides, and is used for plotting and classification. The other one is labeled ’ASI_orig’, and is calculated exactly as stated in the original paper (i.e. $Al/(Ca - 1.67P + Na + K)$).

Note

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots.

See Plate, Plate editing and figaro for details.

Due to the specific design of this scheme (combination of multiple diagrams), the classification option is not available via the pull-down menus. Currently, the only way to apply Frost’s classification in GCDkit on individual samples is to call the function manually from the Console: Frost_2008(clssf = TRUE).

Warning

Note that the Fe-number is calculated as weight proportion of $FeO/(FeO + MgO)$ (or $FeO_{tot}/(FeO_{tot} + MgO)$). The approach used here should not be confused with the more common usage of the term "Fe-number" (as well as "Mg-number") for molecular proportions.

As approved by one of the authors (C. Barnes, pers. comm., 2008), the equation for ASI in the original work (Frost et al. 2001) was stated erroneously as molecular proportions of elements, instead of oxides.

Note

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots. See Plate, Plate editing and figaro for details.

Author(s)

Vojtech Erban, <erban@sopky.cz>

& Vojtech Janousek <vojtech.janousek@geology.cz>

References


See Also

Frost_2008 Shand classify Plate Plate editing plotPlate figaro
Examples

```r
# plot the diagrams
plotPlate("Frost")

# Classify the samples, suppress the graphical output
Frost(clssf=TRUE, GUI=TRUE)
```

---

**Description**

Classification of feldspathic igneous rocks proposed by *Frost and Frost (2008)*.

**Usage**

```r
Frost_2008(plot.txt = getOption("gcd.plot.text"),
          clssf = FALSE, GUI = FALSE)
```

**Arguments**

- `plot.txt` logical, annotate fields by their names?
- `clssf` logical, should the samples be classified?
- `GUI` logical, is the function called from a GUI?

**Details**

Classification scheme originally proposed by *Frost et al. (2001)* for (subalkaline) granitic rocks. It was modified and expanded for alkaline and/or mafic rocks by *Frost and Frost (2008)*. It consists of four diagrams:

- **Fe number** \(FeO_{\text{tot}}/(FeO_{\text{tot}} + MgO)\) vs. \(SiO_2\) (both parameters in wt. %). See *Warning* below.
- **\(SiO_2\)** (in wt. %) vs. Modified alkali-lime index (MALI), defined as \(Na_2O + K_2O - CaO\) (in wt. %).
- **\(SiO_2\)** (in wt. %) vs. Aluminium saturation index (ASI), defined as molecular \(Al_2O_3/(Na_2O + K_2O + CaO - 3.33P_2O_5)\).
  In fact, this is the A/CNK parameter of *Shand (1943)*, corrected for the Ca content in apatite. See also *Warning* below.
  If some \(P_2O_5\) concentrations are missing in the dataset, they are replaced by zero.
- **Feldspathoid silica-saturation index (FSSI) vs. Alkalinity index (AI)**, the former based on CIPW-normative minerals: \((Q - (Lc + 2(Ne + Kp))/100\) and the latter defined as molecular \(Al_2O_3 - (Na_2O + K_2O)\).

The following associations are distinguished:

- ferroan, magnesian
- alkalic, alkali-calcic, calc-alkalic, calcic
- peralkaline, metaluminous, peraluminous
- silica-undersaturated, silica-saturated
Value

The function returns a table of calculated coefficients (Fe-number, MALI, ASI, FSSI).

Note

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots.

See Plate, Plate editing and figaro for details.

Due to the specific design of this scheme (combination of multiple diagrams), the classification option is not available via the pull-down menus. Currently, the only way to apply Frost’s classification in GCDkit on individual samples is to call the function manually from the Console: Frost_2008(clssf = TRUE).
Warning

Note that the Fe-number is calculated as weight proportion of $\frac{FeO_{tot}}{(FeO_{tot} + MgO)}$. The approach used here should not be confused with the more common usage of the term "Fe-number" (as well as "Mg-number") for molecular proportions.

The equation for ASI in both original works of *Frost et al. (2001)* and *Frost and Frost (2008)* was stated erroneously as molecular proportions of elements, instead of oxides.

The equation for FSSI in *Frost and Frost (2008)* has apparently misplaced the outermost brackets, having been originally stated as: $Q\cdot[Qc+2(Ne+Kp)]/100$.

Author(s)

Vojtech Janousek <vojtech.janousek@geology.cz>
& Vojtech Erban, <erban@sopky.cz>

References


See Also

Frost Shand CIPW classify Plate Plate editing plotPlate figaro

Examples

# plot the diagrams
plotPlate("Frost_2008")

# Classify the samples, suppress the graphical output
Frost_2008(clssf = TRUE, GUI=TRUE)

---

**gcdOptions**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A graphical user interface (GUI, programmed in Tcl/Tk) for setting the main options controlling the behaviour of the GCDkit.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>gcdOptions(permanent.only=FALSE)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>permanent.only</td>
</tr>
</tbody>
</table>
**gcdOptions**

### Details

The settings are stored permanently in the file `gcdkit.xxx` residing in the main GCDkit directory. They are loaded upon start up. If is missing or damaged, this file is created anew based on the default values.

The panel connected to the function `gcdOptions` serves to change several parameters. Most of them are passed to a list accessible in a way similar to the standard R `options`. See the corresponding manual page for details and Examples for their implementation. Only a few are stored in dedicated variables (see below).

Firstly, the default working directory can be set (and stored in the global variable `data.dir`).

Secondly, the default Graphical User Interface (GUI) including the menu system can be specified using the option `gcd.menus`. It can attain one of the three values: "", "win", "tcltk". If GCDkit is run under Windows OS using RGui (standard behaviour), a default value of the `gcd.menus` parameter is set to "win" automatically.

Otherwise (any other operating system, Windows in batch mode using the RTerm window), the default value is "" unless it has been modified previously in the configuration file. In these cases, "tcltk" would be the correct setting.

From within GCDkit under any operating system, Tcl/Tk interface can be (re)started anytime using the `menuet` command.

The parameter 'Minimize output on screen?' is linked to the option `gcd.shut.up`. It controls excessive output to the Console window. Its default value is FALSE, meaning that detailed information is to be printed. This, however, may become not viable on slower systems and/or for extensive data sets.

The preferred precision of the numeric values that need to be rounded off are controlled by the parameter 'Precision of results' (option `gcd.digits`).

Using the parameter 'Plotting symbols magnification', linked to the option `gcd.cex`, one can define a factor, by which are multiplied the plotting sizes defined for individual analyses upon startup and stored in the variable `labels[,"Size"]`. Please note that this is effective for the next plot if the GUI frontend is used to set this parameter, otherwise it will work for data files loaded from now on.

In this way, the magnification is maintained proportional to the original sizes. If uniform plotting symbols sizes are desired, one should use the function `setCex` invoked from the menu

Plot settings|Set uniform symbol size.

The parameter 'Annotate fields in discrimination plots?' toggles the labeling of the fields on and off, typically for classification or geotectonic diagrams. It is stored in a logical variable `gcd.plot.text`, whose default is TRUE. The language for the field annotations can be selected using the list box connected to the option `gcd.language`.

The next possibility is to alter the colours used, e.g. for texts or field boundaries on diagrams. There are in total three colours stored in the list `plt.col`. Alternatively, all the plots can be set to black and white (check box 'Set to BW?' linked to the option 'gcd.plot.bw'), excluding the data points. The default is FALSE (i.e. colour plotting).

The parameter 'Identify points?' toggles on and off the identification/labelling of individual data points on plots. In general, the identification can be either interactive (option `gcd.ident.each = TRUE`) or all the points can be labeled automatically as soon as the plotting is finished (option `gcd.ident.each = FALSE`). In the former case, the user may click the left mouse button near the points to be identified, pressing the right mouse button when finished.

The option `gcd.ident` determines whether identification should take place at all (the default value is zero, which means no identification). If the identification is on, the option `gcd.ident` attains
either 1 (identification by sample name), or the sequential number of the column in the data frame
'labels' increased by one (identification by a label).

The identification by sample name for a current plot can be invoked also from the menu
'Plot editing|Identify points'. There can be also chosen alternative means of points identification
('Plot editing|Highlight multiple points').

Value

Sets the following options:

gcd.plot.text logical; should be fields on classification diagrams labeled by their names?
gcd.language language for these labels.
gcd.plot.bw logical; if TRUE, plots are produced as black and white.
gcd.shut.up logical; determines whether extensive textual output is to be printed.
gcd.ident numeric; if zero, no identification takes place after plotting each diagram. If
    greater than zero, indicates the variable used to identify individual data points.
    See Details.
gcd.ident.each logical; are the data points to be identified individually?
gcd.digits preferred number of digits for rounding off the numeric values.
gcd.cex a factor by which are multiplied all symbol sizes previously defined.

Remaining options changed by GCDkit which cannot be altered via the GUI, though:

prompt "GCDkit-> ">
windowsBuffered FALSE
locatorBell FALSE
scipen 20
max.print 99999999
If necessary they can be set directly in the file `gcdkit.xxx`.
Apart from that the GUI panel sets the variables `data.dir` (default data directory) and `plt.col` (colours for Figaro-compatible plots).

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

`options identify ID figaro setCex menuet`

**Examples**

```r
bak <- options()  # backup the current options
options("gcd.ident"=1)  # identify by sample names
options("gcd.ident.each"=FALSE)  # to label by sample names automatically,
                                # i.e. without the user interference
plotDiagram("TAS",FALSE,TRUE)  # to turn off the identification completely
options("gcd.plot.bw"=TRUE)  # to set the diagram to black and white
plotDiagram("TAS",FALSE,TRUE)

options("gcd.cex"=2)  # make the plotting symbols double as big
                        # (effective for the data files loaded from now on;
                        # for immediate result use the GUI front end)
getOption("gcd.plot.bw")  # printing the current value of the given option
options(bak)  # restore the previous options
```

---

**Description**

Closes all graphic windows.

**Usage**

`graphicsOff()`

**Arguments**

None.

**Details**

Under Windows 95/98/ME, the R system may become install, failing to redraw graphical windows if too many of them are being open. It is always a good idea to close the unnecessary ones, for instance using this function.

**See Also**

`'dev.off'`
Grebennikov (2014) $5Fe_2O_3t - Na_2O + K_2O - 5(CaO + MgO)$

**Description**

Assigns data for $5Fe_2O_3t - Na_2O + K_2O - 5(CaO + MgO)$ ternary diagram into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

**Usage**

Grebennikov()

**Arguments**

None.

**Details**

The ternary plot $5Fe_2O_3t - Na_2O + K_2O - 5(CaO + MgO)$ of Grebennikov (2014) serves for classification of A-type granites and related felsic volcanic rocks ($SiO_2 > 67$ wt.%).

**Value**

- `sheet` list with Figaro Style Sheet data
- `x.data` x coordinates
- `y.data` y coordinates
groupsByCluster

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

figaro plotDiagram

Examples

# plot the diagram
# assuming a dataset is loaded, of course!
## Not run:
plotDiagram("Grebennikov",FALSE,TRUE)
## End(Not run)

groupByCluster

Groups by cluster analysis

Description

Grouping the data using the cluster analysis.

Usage

groupsByCluster(elems=
"SiO2,TiO2,Al2O3,FeOt,MnO,MgO,CaO,Na2O,K2O",
method="ave")

Arguments

elems numerical columns to be used for cluster analysis, typically major elements
method the agglomeration method to be employed. This should be one of (or an unambiguous abbreviation thereof): 'ward','single','complete', 'average','mcquitty','median','centroid'.

Details

After the dendrogram is drawn, the user is asked how many clusters is the dataset to be broken into. The vector containing the information on the current groups can be appended to the data frame 'labels'. The groups are initially numbered but this can be changed readily using the function editLabFactor. For further details on the clustering algorithm, see the R manual entry of 'hclust'.


Value

- groups: character vector: the grouping information
- grouping: set to -1.

Author(s)

- Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

- classify
- groupsByLabel
- groupsByDiagram

---

Description

Grouping the data on a basis of selected classification diagram.

Usage

```r
groupsByDiagram(fun = NULL, silent = TRUE)
```

Arguments

- fun: character; name of the classification function available in the system.
- silent: logical; should be echoed the information about classification each of the samples?

Value

- groups: character vector: the grouping information
- grouping: set to -1.

Author(s)

- Vojtech Erban, <erban@sopky.cz>
- Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

- classify
- groupsByLabel
- groupsByCluster

Examples

```r
data(sazava)
accessVar("sazava")

groupsByDiagram("TASMiddlemostPlut") # Function called "TASMiddlemostPlut"
groupsByDiagram("^TAS$") # Function called "TAS"
```
groupsByLabel

Groups by label

Description

Grouping the data according to the levels of a single label.

Usage

groupsByLabel(lab=NULL)

Arguments

lab name or sequence number of the label

Details

Sets the groups on the selected column within the data frame 'labels'. If not specified at the function call, the appropriate label is selected by the function 'selectColumnLabel'.

Value

groups character vector: the grouping information

grouping the sequence number of the column in the data frame 'labels' used for grouping

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

classify groupsByCluster groupsByDiagram

Examples

data(sazava)
accessVar("sazava")
groupsByLabel("Intrusion")
Description

Assigns data for the Hf-Rb/30-Ta*3 ternary diagram of Harris et al. (1986) into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

Usage

Harris()

Details

Triangular diagram with apices Hf, Rb/30 and Ta*3, proposed by Harris et al. (1986) for classification of collisional granites.

Following fields are defined:

- **VA**: Volcanic-Arc granites (Group 1, VA)
- **WP**: Within-Plate granites (Group 4, WP)
- Group 2
- Group 3
Quoting from their abstract:

(i) Group 1 - Pre-collision calc-alkaline (volcanic-arc) intrusions which are mostly derived from mantle modified by a subduction component and which are characterized by selective enrichments in LIL elements.

(ii) Group 2 - Syn-collision peraluminous intrusions (leucogranites) which may be derived from the hydrated bases of continental thrust sheets and which are characterized by high Rb/Zr and Ta/Nb and low K/Rb ratios.

(iii) Group 3 - Late or post-collision calc-alkaline intrusions which may be derived from a mantle source but undergo extensive crustal contamination and can only be distinguished from volcanic-arc intrusions by their higher ratios of Ta/Hf and Ta/Zr.

(iv) Group 4 - Post-collision alkaline intrusions which may be derived from mantle lithosphere beneath the collision zones and which carry high concentrations of both LIL and HFS elements.

Value

<table>
<thead>
<tr>
<th>sheet</th>
<th>list with Figaro Style Sheet data</th>
</tr>
</thead>
<tbody>
<tr>
<td>x.data, y.data</td>
<td>Th, Hf/3 and Ta in ppm recalculated into two dimensions</td>
</tr>
</tbody>
</table>

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

figaro plotDiagram

Examples

#plot the diagram
plotDiagram("Harris", FALSE)

Hastie

Co-Th diagram (Hastie et al. 2007)

Description

Assigns data for Co vs. Th (ppm) diagram into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'

Usage

Hastie()
Details

Diagram in Co vs. Th space, proposed by Hastie et al. (2007) for subdivision of volcanic arc rocks. This is thought to be a more robust replacement for SiO$_2$ vs. K$_2$O plot of Peccerillo & Taylor (1976) for altered/weathered volcanic rocks. The decreasing Co concentrations are used as an index of fractionation (as a proxy for SiO$_2$), the Th contents mimic those of K$_2$O.

The following fields are defined:

- **Tholeiite Series**
- **Calc-alkaline Series**
- **High-K Calc-alkaline and Shoshonite Series**

Rocks with composition falling beyond defined boundaries are labeled 'undefined' by the 'classify' function.

In addition, the diagram discriminates between the following rock types:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full name</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>basalt</td>
</tr>
<tr>
<td>BA/A</td>
<td>basaltic andesite and andesite</td>
</tr>
<tr>
<td>D/R*</td>
<td>dacite and rhyolite*</td>
</tr>
</tbody>
</table>

* latites and trachytes also fall in the D/R fields

Value

sheet list with Figaro Style Sheet data
Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

classify figaro plotDiagram

Examples

#Within GCDkit, the plot is called using following auxiliary functions:
#To Classify data stored in WR (Groups by diagram)
classify("Hastie")

#To plot data stored in WR or its subset (menu Classification)
plotDiagram("Hastie", FALSE)

---

**Hollocher**

*Hollocher et al. (2012)* La/Yb vs. Nb/La or Th/Nb

---

**Description**

Assigns data for La/Yb vs. Nb/La or La/Yb vs. Th/Nb binary diagrams into Figaro template (list ‘sheet’) and appropriate values into ‘x.data’ and ‘y.data’.

**Usage**

Hollocher1()
Hollocher2()

**Arguments**

None.
The two binary plots, La/Yb vs. Nb/La and La/Yb vs. Th/Nb, of Hollocher et al. (2012) serve for geotectonic discrimination of basalts or basaltic amphibolite units. These diagrams can distinguish between the MORB, enriched ocean island basalts, and the near continuum defined by oceanic, continental, and alkaline arcs.

However, the authors have noted that basalts from back-arc basins have a wide range of compositions caused by basalt source region variation between depleted N-MORB, ocean-island (hot spot), and subduction component-enriched (sub-volcanic arc) mantle end members. See also their Fig. 16.
Value

- sheet: list with Figaro Style Sheet data
- x.data: x coordinates
- y.data: y coordinates

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


Coordinates and graph layout are taken from website of Kurt Hollocher.

See Also

figaro plotDiagram
Examples

plotDiagram("Hollocher1",FALSE,TRUE)
plotDiagram("Hollocher2",FALSE,TRUE)

<table>
<thead>
<tr>
<th>ID</th>
<th>Sample identification</th>
</tr>
</thead>
</table>

Description
Identification/labelling of individual data points on plots.

Usage
ID(x, y, labs = getOption("gcd.ident"), offset = 0.4,
col = "gray30", cex = 1)

Arguments
- x, y: vector with x-y coordinates of the data points
- labs: text to label individual data points, see details
- offset: distance (in char widths) between label and identified points.
- col: colour of the text
- cex: its size

Details
In GCDkit, the option ‘ident’ determines whether the user wishes to identify data points on binary and ternary plots. The default is zero, which means no identification.
If ‘ident’ differs from zero, internal function ‘ID’ can be invoked. Its parameter labs is either a single number, or character vector.
In the former case, the variable ‘labs’ contains either 1 (identification by sample name), or the sequential number of the column in the data frame ‘labels’ increased by one (identification by a user-defined label).
Alternatively, a character vector labs can be used to specify the text directly.

Value
None.

Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also
identify, gcdOptions, options

Examples
getOption("ident") # yields the current value of the given option
Description

Prints information about the current dataset (and its selected subset, if applicable).

Usage

info()

Details

This function prints comprehensive information about the current dataset. For each of the labels, individual levels and their frequencies are given. The number of numeric columns is printed, and for each of the variables number of available values. Moreover, the information concerning the total number of samples, the names of the samples in the selected subset (or all samples if none is defined) and the current grouping are shown.

Value

None

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

isochron

Rb-Sr and Sm-Nd isochrons

Description

Plots a Rb-Sr or Sm-Nd isochron diagram and calculates a simple linear fit to the selected data.
Usage

`isochron()`

Arguments

None.

Details

The samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSamples` for details.

If empty list is given, all the samples for which the required isotopic data are available are plotted and the user can choose their subset interactively. Then the isochron diagram is redrawn only with those samples.

The data are fitted by simple least-squares linear fit, from which the age and initial ratio are calculated.
Value

Returns a numeric vector with the calculated age and initial ratio.

Plugin

SrNd.r

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

isocon

Isocon plots (Grant 1986)

Description

Implementation of isocon plot after Grant (1986, 2005) widely used for quantitative estimates of changes in mass/volume/concentration of elements or oxides in course of various open-system geochemical processes such as alteration or partial melting.

Usage

isocon(x = NULL, whichelems = NULL, immobile = NULL, atomic = FALSE, plot = TRUE)
isoconAtoms()
isoconOxides()

Arguments

x numeric matrix with the chemical data
whichelems list of elements for plotting, separated by commas
immobile list of presumed immobile elements, separated by commas
atomic logical; should be atomic wt. % used for oxides?
plot logical; is the graphical output desirable?

Details

Isocon plot (Grant 1986, 2005) spans from the theoretical quantitative treatment of losses or gains of geochemical species (elements or oxides). It is applicable to balancing mass, volume and/or concentration changes in course of open-system processes such as weathering, hydrothermal alteration, metasomatic addition/leaching or migmatitization.

According to Grant (2005 and references therein) the equation for composition/volume changes in open-system process can be written as:

\[ c_i^A = \frac{M^0}{MA} c_i^0 + \Delta c_i \]
where $c_i$ is the concentration of the species $i$, $0$ refers to the original rock and $A$ to the altered rock, $M^0$ is the equivalent mass before and $M^A$ after alteration.

For immobile element ($\Delta c_i = 0$) the ratio $\frac{M^0}{M^A}$ reflecting the overall change in mass can be obtained. This can be done graphically in the plot of the analytical data for presumed protolith ($c_0^i$) and altered rock ($c_A^i$). Such a straight line passing through the origin is termed isocon, the equation of which is:

$$
c_A^i = \left(\frac{M^0}{M^A}\right)c_0^i
$$

Species plotting above the isocon were gained, whereas those plotting below were lost, and the gain or loss is according to Grant (2005):

$$
\frac{\Delta c_i}{c_i^-} = \frac{M^A}{M^0} \frac{c_A^i}{c_0^i} - 1
$$

where $\frac{c_A^i}{c_0^i}$ is the slope of the tie line from the origin to the data point.

In the GCDkit’s implementation of the function ‘isocon’, firstly the parental and altered rock samples are to be chosen interactively from a binary plot $MgO – SiO_2$. Then the user is prompted for the elements/oxides to be used in the isocon analysis. Printed and plotted in the form of barplots are ordered slopes for each data point in the isocon diagram.

The user can choose the presumably immobile elements. These can be either provided as a comma delimited list, or, if empty, chosen interactively from the isocon plot. Finally are plotted two isocons, as well as a blue equiline (a straight line with the slope 1).

Implemented are two methods for assessing the change in mass of the system. Traditionally used has been the slope of the isocon line, obtained by linear regression of the presumably immobile data (dark green). However, this depends on the scaling of the isocon plot, which is arbitrary. In particular, the data plotted close to the origin may appear erroneously to lie on an isocon (Baumgartner & Olsen, 1995).

More objectively, the change in the mass can be estimated by clustering slopes to data points, deciphering the elements/oxides with a similar behaviour and averaging the slopes for the selected presumably immobile species.

Functions ‘isoconAtoms’ and ‘isoconOxides’ are frontends to the function ‘isocon’, providing different default values. See Arguments above.
Value

Returns a list 'results' with the following components:

- **slope.avg**: slope of the isocon obtained as an average of the slopes for the individual presumably 'immobile' species
- **slope.regression**: slope obtained by linear regression
- **balance**: numeric matrix; balance of individual species. This matrix contains the following columns:
  - XXX=orig.: composition of the parental (unaltered) rock
  - XXX=alt.: composition of the altered rock
- **Slope data point**: slope of the line connecting the data point with origin
- **G/L rel.(LQ)**: relative mass gain/loss, isocon slope by least-squares fit
- **G/L rel.(avg)**: relative mass gain/loss, averaged slopes for immobile elements
- **G/L wt%/ppm(LQ)**: absolute mass gain/loss, isocon slope by least-squares fit
- **G/L wt%/ppm(avg)**: absolute mass gain/loss, averaged slopes for immobile elements

Plugin

Isocon.r

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


Examples

# Grant (2005) - see Tab. 1, Fig. 1
x<-matrix(c(46.45,1.29,14.30,11.05,0.17,5.28,12.14,2.93,0.49,3.00,3.29,42,327,313,67,77,100,170,29,80,45.62,1.30,14.74,8.20,0.15,3.89,8.29,2.09,3.12,2.18,10.96,39,305,282,42,75,72,214,17,140), byrow=TRUE,nrow=2)
y<"SiO2,TiO2,Al2O3,Fe2O3,MnO,MgO,CaO,Na2O,K2O,H2O,CO2,Sc,V,Cr,Ni,Cu,Zn,Sr,Y,Ba"
colnames(x)<-unlist(strsplit(y","))
rownames(x)<c("UA","401")
isocon(x,y,atomic=FALSE,plot=TRUE,immobile="Al2O3,SiO2,TiO2,Cu,Sc")
isocon(x, y, atomic=TRUE, plot=FALSE)

Jensen

**Jensen cation plot (1976)**

**Description**
Assigns data for Jensen’s cation plot into Figaro template (list ‘sheet’) and appropriate values into ‘x.data’ and ‘y.data’.

**Usage**
Jensen()

**Details**
Jensen’s cation plot, proposed by *Jensen* (1976) and modified by *Jensen & Pyke* (1982). The triangular diagram is defined on the basis of millifications as follows:
left apex: Al
upper apex: $Fe^T + Ti$
right apex: Mg

The diagram defines following rock series and names:

*Komatiite series (KOMATIITE) Komatiite*
joinGroups

**Komatitic basalt**

*Tholeiite series (TH)*
- Rhyolite
- Dacite
- Andesite
- High-Fe tholeiite basalt
- High-Mg tholeiite basalt

*Calc-alkaline series (CA)*
- Rhyolite
- Dacite
- Andesite
- Basalt

**Value**

- `x.data`, `y.data` Values for the three apices transformed into 2D space
- `sheet` list with Figaro Style Sheet data

**Author(s)**

Vojtech Erban, <erban@sopky.cz>
& Vojtech Janousek, <vojtech.janousek@geology.cz>

**References**


**See Also**

classify, figaro, plotDiagram

**Examples**

```r
#plot the diagram
plotDiagram("Jensen", FALSE)
```

**joinGroups**

*Merge groups*

**Description**

Enables merging several groups into a single one.

**Usage**

`joinGroups()`
Arguments

None.

Details

This function is the most useful to merge several groups, defined e.g. on the basis of a classification plot. A simple spreadsheet is invoked with two columns, the first ('Old') containing the old levels of groups and the second, 'New', which can be edited. Finally, groups with identical names will be merged into a single one.

Optionally, the vector containing the information on the current groups can be appended to the data frame `labels`.

Value

groups character vector: the grouping information

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

Description

This function estimates the temperature of a granitic magma based on measured $\text{Al}_2\text{O}_3/\text{TiO}_2$ ratio and experimental constraints. The regression formulae were defined by Jung & Pfänder (2007).

Usage

Jung(model = NULL, plot = TRUE)

Arguments

model specification of the model

plot logical; should be shown a $\text{Al}_2\text{O}_3/\text{TiO}_2$ vs. $\text{CaO}/\text{Na}_2\text{O}$ plot?

Details

As shown by Sylvester (1998), the $\text{Al}_2\text{O}_3/\text{TiO}_2$ ratio in the granitic magmas is temperature sensitive, decreasing with the increasing temperature of the crustal anatexis. This probably reflects an increasing instability of Ti-bearing phases with progressive crustal fusion.

Jung & Pfänder (2007) compiled the available experimental data and defined a set of regression formulae (linear, power law and exponential) for several types of protoliths.

Any of the following models can be chosen: pelite melting, psammite melting, igneous rock melting, A-type granite melting, amphibolite melting after Rapp & Watson (1995) and amphibolite melting after Patino Douce & Beard (1995).
 Optionally, also $Al_2O_3/TiO_2$ vs. $CaO/Na_2O$ plot could be displayed with three secondary axes annotated by the calculated temperatures.

![Temperature (°C) after Jung and Pfänder (2007)](image)

**Value**

Returns a matrix `results` with the following columns:

- $Al_{203}/TiO_2$ wt. % ratio of $Al_2O_3/TiO_2$
- $T_{Al/Ti}.power.C$ temperature in C, power law calibration
- $T_{Al/Ti.exp.C}$ temperature in C, exponential calibration
- $T_{Al/Ti.linear.C}$ temperature in C, linear calibration
- $T_{Al/Ti.mean.C}$ mean temperature in C, based on the above three models

**Plugin**

`Jung.r`
Erratum

As pointed out by S. Jung (pers. com. 2009), in Table 1 of their original paper were printed wrongly several of the regression coefficients. These are:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A-type</td>
<td>power law</td>
<td>$B = 0.992$</td>
<td>$B = 9.921$</td>
</tr>
<tr>
<td>amphibolite (Rapp and Watson 1995)</td>
<td>power law</td>
<td>$A = 2.82 \times 10^3$</td>
<td>$A = 2.82 \times 10^3$</td>
</tr>
</tbody>
</table>

The function implements these corrected values.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


doi: 10.1016/S0024-4937(98)00024-3

Examples

Jung()
Jung("A-type")
Jung("psammit",plot=FALSE)

Laroche R1-R2 diagram (De la Roche et al. 1980)

Description

Assigns data for the $R_1 - R_2$ diagram into Figaro template (list `sheet`) and appropriate values into `x.data` and `y.data`.

Usage

LarocheVolc()
LarochePlut()
Details

$R_1 - R_2$ plot, as proposed by De La Roche et al. (1980) for volcanic, as well as plutonic rocks.
nepheline  iolite
andesite-andesite  diorite
laves-lavas  tonalite
hawaiite  syenogabbro
tholeiite  gabbro
basalt  gabbro
eka-ska  alkaline gabbro
basanite  theralite
ankaratite  melteigite
picrite  ultramafic rock

\[ R_1 = 4 \cdot Si - 11 \cdot (Na + K) - 2 \cdot (Fe^{\text{total as bivalent}} + Ti) \]
\[ R_2 = 6 \cdot Ca + 2 \cdot Mg + Al \]

Value

sheet list with Figaro Style Sheet data
x.data R1 = 4 * Si - 11 * (Na + K) - 2 * (Fe[total as bivalent] + Ti), all in millications; as calculated by the function 'LaRocheCalc()' 
y.data R2 = 6 * Ca + 2 * Mg + Al, all in millications; as calculated by the function 'LaRocheCalc()'
Author(s)
Vojtech Erban, <erban@sopky.cz> & Vojtech Janousek, <vojtech.janousek@geology.cz>

References

See Also
classify figaro LaRocheCalc millications plotDiagram

Examples
#Within GCDkit, the plot is called using following auxiliary functions:
#To classify data stored in WR (Groups by diagram)
classify("LarocheVolc")
#or
classify("LarochePlut")

#To plot data stored in WR or its subset (menu Classification)
plotDiagram("LarocheVolc", FALSE)
#or
plotDiagram("LarochePlut", FALSE)

LaRocheCalc Calculation: De la Roche

Description
Recalculates whole-rock data into $R_1 - R_2$ values of De La Roche et al. (1980).

Usage
LaRocheCalc(rock=WR,precision=getOption("gcd.digits"))

Arguments

rock a numeric matrix with whole-rock data to be recalculated.

precision precision of the result.

Details
$R_1 - R_2$ parameters, as proposed by De La Roche et al. (1980):

\[ R_1 = 4 \times \text{Si} - 11 \times (\text{Na} + \text{K}) - 2 \times (\text{Fe} \, \text{total as bivalent} + \text{Ti}), \] all in millications
\[ R_2 = 6 \times \text{Ca} + 2 \times \text{Mg} + \text{Al}, \] all in millications

Value

results numeric matrix with the two above specified parameters
LaurentSource

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

LaRoche

Examples

data(sazava)
accessVar("sazava")
LaRocheCalc()

LaurentSource  Laurent et al. (2014) granitoid sources

Description

Assigns data for a Laurent et al. (2014) triangular diagram into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

Usage

LaurentSource()

Details

Triangular diagram of Laurent et al. (2014, Fig. 6) with apices: $\frac{Al_2O_3}{(FeOt + MgO)}$, $3CaO$ and $5K_2O/Na_2O$ (all in wt. %), serves to distinguish prospective sources of granitoids. It is based on compilation of a large database of experimental data - see Laurent et al. (2014 and references therein) for the citations of the original works.
The following sources can be distinguished:

- Low-K mafic rocks
- High-K mafic rocks
- Tonalites
- Metasediments

**Value**
- sheet list with Figaro Style Sheet data
- x.data, y.data Al2O3/(FeOt+MgO) - 3CaO - 5K2O/Na2O values recalculated into two dimensions

**Author(s)**
- Jean-Francois Moyen, <jfmoyen@gmail.com>
- Oscar Laurent, <oscarlaurent86@gmail.com>

**References**
**Example**

```r
# plot the diagram
plotDiagram("LaurentSource", FALSE)
```

---

### Description

Loads data from a file (or, alternatively, a clipboard) into GCDkit. The files may contain plain text, or, if library RODBC (has been installed, can be in the dBase III/IV (*.dbf), Excel (*.xls), Access (*.mdb), PetroGraph (*.peg), IgPet or NewPet (*.roc) formats.

### Usage

```r
loadData(filename = NULL, separators = c("\t", ",", ";", " ", " "),
    na.strings = c("NA", ",", "bd", "b.d.", "bd1", "b.d.1.", "N.A.", "n.d."),
    clipboard = FALSE, merging = FALSE);
loadDataOdbc(filename = NULL, na.strings = c("NA", ",", "bd",
    "b.d.", "bd1", "b.d.1.", "N.A.", "n.d."), merging = FALSE,
    ODBC.choose = TRUE)
```

### Arguments

- **filename**: fully qualified name of the file to be loaded, including suffix.
- **separators**: strings that should be tested as prospective delimiters separating individual items in the data file.
- **na.strings**: strings that will be interpreted, together with empty items, zeros and negative numbers, as missing values (NA).
- **clipboard**: logical; is clipboard to be read instead of a file?
- **merging**: logical; is the function invoked during merging of two data files?
- **ODBC.choose**: logical; if TRUE, ODBC channel can be chosen interactively.

### Details

If library RODBC is available, the functions attempt to establish an ODBC connection to the selected file, and open it as dBase III/IV (*.dbf), Excel (*.xls) or Access (*.mdb) format. The DBF files are used to store data by other popular geochemical packages, such as IgPet (Carr, 1995) or MinPet (Richard, 1995).

Another format that can be imported is *.csv. It is employed by geochemical database systems such as GEOROC (http://georoc.mpch-mainz.gwdg.de/georoc/) and PETDB (http://www.petdb.org/).

The import filter for the *.csv files has been tailored to keep the structure of these databases in mind.
The package PetroGraph (Petrelli et al. 2005) saves data into *.peg files that are also, in principle, *.csv files compatible with the GCDKit.

Data files *.roc are yet another variant of *.csv files, used by NewPet (Clarke et al. 1994). This is not to be confused with the *.roc format designed for IgPet (Carr, 1995). This is a text file with a quite complex structure, whose import is still largely experimental. DBF files are to be preferred for this purpose.

If not successful, the function 'loadData' assumes that it is dealing with a simple text file.

On the other hand 'loadDataOdbc' allows an ODBC channel to be specified interactively if 'ODBC.choose=TRUE'.

Plain text files can be delimited by tabs, commas or semicolons (the delimiter is recognized automatically). Alternative separators list can be specified by the optional 'separators' parameter. The Windows clipboard is just taken as a special kind of a tab-delimited text file.

In the text file, the first line contains names for the data columns (except for the first one that is automatically assumed to contain the sample names); hence the first line may (or may not) have one item less than the following ones. The data rows start with sample name and do not have to be all of the same length (the rest of the row is filled by 'NA' automatically).

Missing values ('NA') are allowed anywhere in the data file (naturally apart from sample and column names); any of 'NA', 'N.A.', '-', 'b. d.', 'bd', 'b. d. l.', 'bd/l' or 'n. d.' are also treated as such, as specified by the parameter na.strings.

While loading, the values '#WHATEVER!' (Excel error messages) are also replaced by 'NA' automatically.

Please note that the function 'loadDataOdbc', due to the current limitations of the RODBC package, cannot handle correctly columns of mixed numeric and textual data. In such a column all textual information is converted to 'NA' and this unfortunately concerns the sample names as well. If encountering any problems, please use import from text file or via clipboard, which are much more robust.

The negative numbers and values '<x' (used by some authors to indicate items below detection limit) can be either replaced by their half (i.e. half of the detection limit) or 'NA'. User is prompted which of these options he prefers.

Alternatively, the negative values can be viewed either as missing ('NA') or can be imported, as may be desirable for instance for stable isotope data in the delta notation.

Decimal commas, if present in text file, are converted to decimal points.

The data files can be practically freeform, i.e. no specified oxides/elements are required and no exact order of these is to be adhered to. Analyses can contain as many numeric columns as necessary, the names of oxides and trace elements are self-explanatory (e.g. "Si02", "Fe2O3", "Rb", "Nd").

In the text files (or if pasting from clipboard), any line starting with the hash symbol ('#') is ignored and can be used to introduce comments or to prevent the given analysis from loading temporarily.

Note that names of variables are case sensitive in R. However, any of the fully upper case names of the oxides/elements that appear in the following list are translated automatically to the appropriate capitalization:

SiO2, TiO2, Al2O3, Fe2O3, FeO, MnO, MgO, CaO, Na2O, FeOt, Fe2O3t, Li2O, mg#, Ac, Ag, Al, As, At, Au, Ba, Be, Bi, Br, Ca, Cd, Ce, Cl, Co, Cr, Cs, Cu, Dy, Er, Eu, Fe, Ga, Gd, Ge, Hf, Hg, Ho, In, Ir, La, Li, Lu,
Mg, Mn, Mo, Na, Nb, Nd, Ne, Ni, Np, Os, Pa, Pb, Pd, Pr, Pt, Pu, Re, Rh, Ru, S, Sb, Sc, Se, Si, Sm, Sn, Sr, Ta, Tb, Te, Th, Ti, Tl, Tm, Yb, Zn, Zr.

Total iron, if given, should be expressed either as ferrous oxide ('FeOt', 'FeOT', 'FeOtot', 'FeOTOT' or 'FeOx') or ferric oxide ('Fe2O3t', 'Fe2O3T', 'Fe2O3tot', 'Fe2O3TOT' or 'Fe2O3x').

Structurally bound water can be named 'H2O.Plus', 'H2O+', 'H2OPLUS', 'H2O' or 'H2O_Plus'.

Upon loading, all the completely empty columns are removed first. Any non-numeric items found in a data column with one of the names listed in the above dictionary are assumed to be typos and replaced by 'NA', after a warning appears. At the next stage all fully numeric data columns are stored in a numeric data matrix 'WR'.

For any missing major- and minor-element data (SiO2, TiO2, Al2O3, Fe2O3, FeO, MnO, MgO, CaO, Na2O, K2O, H2O.Plus, CO2, P2O5, F, S), an empty (NA) column is created automatically.

The remaining, that is all at least partly textual data columns are transferred to the data frame 'labels'. To this are also attached a column whose name starts with 'Symbol' (if any) that is taken as containing plotting symbols and a column whose name is 'Colour' or 'Color' (if any, capitalization does not matter) that may contain plotting colours specification. The relative size of the individual plotting symbols may be specified in a column named 'Size' or 'cex' that is also to be attached to the 'labels'.

The plotting symbols can be given either by their code (see showSymbols) or directly as strings of single characters.

The colours can be specified as codes (1-49) or English names (see showColours or type 'colours()' into the Console window).

If specifications of the plotting symbols and colours are missing completely, and at least one non-numeric variable is present, the user is prompted whether he does not want to have the symbols and colours assigned automatically, from 1 to n, according to the levels of the selected label. Otherwise default symbols (empty black circles) are used.

The default grouping is set on the basis of plotting symbols 'labels$Symbol' or the data column used to autoassign the plotting symbols and colours.

Lastly, a backup copy of the data is stored in the list 'WRCube' using the function 'pokeDataset'. It is stored either under the name of the file, or, if it already exists, under the file name with a time stamp attached.

Value

<table>
<thead>
<tr>
<th>WR</th>
<th>numeric matrix: all numeric data</th>
</tr>
</thead>
<tbody>
<tr>
<td>labels</td>
<td>data frame: all at least partly character fields; labels$Symbol contains plotting symbols and labels$Colour the plotting colours</td>
</tr>
</tbody>
</table>

The function prints a short summary about the loaded file. It also loads and executes the Plugins, i.e. all the R code (*.r) that is currently stored in the subdirectory '\Plugin'. Finally, the system performs some recalculations (calling 'Gcdkit.r').
Note

In order to ensure the database functionality, duplicated column (variable) names are not allowed. This concerns, to a large extent, also the sample names. The only exception are CSV files - if duplicated samples are found, sequence numbers are assigned instead. All completely empty rows and columns in both labels and numeric data are ignored.

Author(s)

The RODBC package was written by Brian Ripley.
Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

'saveData' 'mergeData' 'pokeDataset' 'showColours' 'showSymbols' 'read.table' 'getwd' 'setwd'

Examples

# Sets the working path and loads the 'sazava' test data set
setwd(paste(gcdx.dir,"Test_data",sep="/"))
loadData("sazava.data")

Description

Plots data stored in 'WR' (or its subset) into Maniar and Piccoli’s series of diagrams.

Usage

Maniar(plot.txt = getOption("gcd.plot.text"))

Arguments

plot.txt logical, annotate fields by their names?

Details

Collection of six binary diagrams, based on major elements chemistry, developed by Maniar & Piccoli (1989) for tectonic discrimination of granitic rocks. Shand's (1943) diagram is also used. Diagrams are defined as follows:
**Maniar**

**x axis**

\[ \text{SiO}_2 \]

\[ \text{SiO}_2 \]

\[ \text{SiO}_2 \]

M and F proportion in the AFM system

C and F proportion in the ACF system

A/CNK (molar)

**y axis**

\[ \text{K}_2\text{O} \]

\[ \text{Al}_2\text{O}_3 \]

\[ \frac{100\times \text{MgO}}{(\text{FeO(T)}+\text{MgO})} \]

\[ \frac{100+\text{MgO}}{(\text{Al}_2\text{O}_3+\text{Na}_2\text{O}+\text{K}_2\text{O}+\text{FeO(T)}+\text{MgO})} \]

\[ \frac{100+\text{FeO(T)}}{(\text{Al}_2\text{O}_3+\text{Na}_2\text{O}+\text{K}_2\text{O}+\text{FeO(T)}+\text{MgO})} \]

\[ \frac{100+\text{FeO(T)}}{(\text{Al}_2\text{O}_3+\text{Na}_2\text{O}+\text{K}_2\text{O}+\text{FeO(T)}+\text{MgO}+\text{CaO})} \]

\[ \frac{100+\text{FeO(T)}+\text{MgO}}{(\text{Al}_2\text{O}_3+\text{Na}_2\text{O}+\text{K}_2\text{O}+\text{FeO(T)}+\text{MgO}+\text{CaO})} \]

A/NK (molar)

Abbreviations used in diagrams represent granitoids from following geotectonic environments:

- **IAG** Island Arc Granitoids
- **CAG** Continental Arc Granitoids
- **CCG** Continental Collision Granitoids
- **POG** Post-orogenic Granitoids
- **RRG** Rift-related Granitoids
- **CEUG** Continental Epeirogenic Uplift Granitoids
- **OP** Oceanic Plagiogranites
Peralkaline, Metaluminous and Peraluminous rocks are defined in the last (Shand’s) diagram.

Note

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots. See Plate, Plate editing and figaro for details.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

Plate, Plate editing, figaro, plotPlate,

Examples

#plot the diagrams
plotPlate("Maniar")
mergeData

Appending data to a current data set

Description

These functions append new data to the analyses currently stored in the memory of the GCDkit.

Usage

mergeDataRows()
mergeDataCols(all.rows=NULL)

Arguments

all.rows logical; should be all samples preserved, even those missing in one of the datasets?

Details

The function 'mergeDataRows' appends new samples (i.e. new rows). The structures of both datafiles are, as much as possible, matched against each other, and, if necessary, new empty columns are introduced to the original data file, if they are missing. If any duplicated sample names are found, they are replaced by sequence numbers and a new column 'old.ID' is appended to the labels. Also appended is a column named 'file' containing the name of the file the particular sample originated from.

'mergeDataCols' adds new data (i.e. new data columns) to the samples stored in the memory. If desired ('all.rows' is 'TRUE'), included are also samples that occur solely in one of the files.

For the guidelines on correct formatting of the data files see loadData.

Value

WR numeric matrix: all numeric data
labels data frame: all at least partly character fields; labels$Symbol contains plotting symbols and labels$Colour the plotting colours

The function prints a short summary about the loaded file.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

'loadData' 'saveData' 'merge'
Meschede (1986) Zr/4-2Nb-Y

Description
Assigns data for a Meschede's (1986) triangular diagram into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

Usage
Meschede()

Details
Triangular diagram with apices Zr/4, 2Nb and Y, proposed by Meschede (1986). The plot serves primarily for tectonic discrimination of tholeiitic basalts.

Abbreviations used in diagram represent following geotectonic settings:

- AI-AII: Within-Plate Alkaline Basalts
- AII-C: Within-Plate Tholeiites
- B: P-type Mid-Ocean Ridge Basalts
- D: N-type Mid-Ocean Ridge Basalts
- C-D: Volcanic Arc Basalts
### Value
- **sheet** list with Figaro Style Sheet data
- **x.data, y.data** Zr/4, 2Nb and Y values recalculated into two dimensions

### Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>

### References

### See Also
- figaro plotDiagram

### Examples
```r
#plot the diagram
plotDiagram("Meschede",FALSE)
```

---

### Description
*Improved Mesonorm for granitoid rocks*

Calculates *eine bessere* Mesonorm for granitoids of Mielke & Winkler (1979).

### Usage
```r
Mesonorm(WR, GUI = FALSE, precision = getOption("gcd.digits"))
```

### Arguments
- **WR** a numerical matrix; the whole-rock data to be normalized.
- **GUI** logical, is the function called from the GUI?
- **precision** precision of the result.

### Details
This method of norm calculation should yield mineral proportions close to the actual mode of granitoid rocks. The calculated components are:

- Orthoclase, Albite, Anorthite, Quartz, Apatite, Magnetite, Hematite, Ilmenite, Biotite, Amphibole, Calcite, Corundum, Rest

If desired, the function plots the Q'-ANOR diagram of Streckeisen & Le Maitre (1979) using the function QANOR.
Value

A numeric matrix 'results'.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

QANOR

Examples

data(sazava)
accessVar("sazava")
Mesonorm(WR)

Description

Assigns data for Middlemost’s diagram into Figaro template (list ‘sheet’) and appropriate values into ‘x.data’ and ‘y.data’.

Usage

MiddlemostPlut()

Details

Classification diagram, as proposed by Middlemost (1985) for plutonic rocks.
Value

- sheet: list with Figaro Style Sheet data
- x.data: SiO2 weight percent
- y.data: Na2O+K2O weight percent
- results: matrix with classification results
- groups: vector with classification results
- grouping: set to -1

Author(s)

Vojtech Erban, <erban@sopky.cz>
& Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

classify figaro plotDiagram
Examples

# Within GCDkit, the plot is called using following auxiliary functions:
# To classify data stored in WR (Groups by diagram)
classify("MiddlemostPlut")

# To plot data stored in WR or its subset (menu Classification)
plotDiagram("MiddlemostPlut", FALSE)

Millications

Description

Returns millications.

Usage

millications(x=WR,print=TRUE,save=FALSE,precision=getOption("gcd.digits"))

Arguments

x matrix or vector with major-element data
print logical: print the result?
save logical: should the results assigned globally?
precision precision of the result.

Details

The millications are used for many plots of the French school, e.g. De la Roche et al. (1980) or
The calculated values are Si, Ti, Al, Fe3, Fe2, Fe, Mn, Mg, Ca, Na, K, P.

\[ Element_i = 1000 \frac{Oxide_i(\text{wt.\%})}{MW(Oxide_i)} \times x(Element_i) \]

Where: MW = molecularWeight of the Oxide[i], x = number of atoms of Element[i] in its formula

Value

Numeric matrix (or vector) with the millications. If 'save=TRUE', 'results' and 'milli' are assigned globally.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>
References


Examples

data(sazava)
accessVar("sazava")
millications()

mins2deg

Recasts degrees and minutes to degrees (with decimal places).

Description

The functions serves to recast strings in the form 48°53.6814’ to decimal values in degrees. As separators are used the 'degree' sign (unicode + 00B0) and apostrophe, respectively. There are no spaces.

Usage

mins2deg(x="Easting",varname="XX")

Arguments

x
  a text string, to be interpreted directly, or a colname of variable in 'labels'

varname
  a name of a numeric variable in 'WR' to store the recalculated output

Value

Returns, invisibly, the converted numbers. Appends the recalculated coordinate to the data matrix 'WR'.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

Examples

data(sazava)
accessVar("sazava")
WR<-WR[1:5,]
x<-c("48°53.6814'","48°53.6814'","48°53.6814'","48°53.6814'","48°53.6814'")
mins2deg(x,"E")
Miscellaneous geochemical indexes

Description

Calculates a series of useful geochemical indexes.

Usage

Misc(WR)

Arguments

WR a numerical matrix; the whole-rock data to be recalculated.

Details

Various petrochemical indexes are calculated, such as:

- total iron as $Fe_2O_3$
- $Fe_2O_3/FeO$, $Na_2O/K_2O$ and $K_2O/Na_2O$ ratios
- Larsen’s DI - Differentiation index (Larsen 1938)
- Kuno’s SI - Solidification index (Kuno 1959)
- Agpaitic index (Ussing 1912)

Value

A numeric matrix ‘results’.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References

Larsen E S (1938) Some new variation diagrams for groups of igneous rocks. J Geol 46: 505-520
Miyashiro  

**SiO2-FeOt/MgO diagram (Miyashiro 1974)**

**Description**

Assigns data for $\text{SiO}_2$ vs. $\text{FeO}_t/\text{MgO}$ diagram into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

**Usage**

Miyashiro()

**Details**

Diagram in $\text{SiO}_2$ vs. $\text{FeO}_t/\text{MgO}$ space, proposed by Miyashiro (1974), defines the following fields:

- **Tholeiite Series**
- **Calc-alkaline Series**

As the boundary was defined by Akiho Miyashiro as straight line passing through two specific points, no limits of diagram validity for ultrabasic and high-silica rocks were given. Thus, the boundary implemented in GCDkit script spreads from $\text{FeO}_t/\text{MgO} = 0$ to $\text{SiO}_2 = 100\%$. 
**Mode**

**Value**

<table>
<thead>
<tr>
<th>sheet</th>
<th>list with Figaro Style Sheet data</th>
</tr>
</thead>
<tbody>
<tr>
<td>x.data</td>
<td>SiO2 weight percent</td>
</tr>
<tr>
<td>y.data</td>
<td>FeOt/MgO weight percent</td>
</tr>
</tbody>
</table>

**Author(s)**

Vojtech Erban, <erban@sopky.cz>
& Vojtech Janousek, <vojtech.janousek@geology.cz>

**References**


**See Also**

classify figaro plotDiagram

**Examples**

#Within GCDkit, the plot is called using following auxiliary functions:
#To Classify data stored in WR (Groups by diagram)
classify("Miyashiro")

#To plot data stored in WR or its subset (menu Classification)
plotDiagram("Miyashiro", FALSE)

---

**Mode**

**Approximating the mode by least-squares method**

**Description**

The functions ‘Mode’ and ‘ModeC’ calculate the best approximations of the mode given major-element compositions of the rock and its main mineral constituents. Function ‘WRComp’ does the opposite, i.e. yields the whole-rock composition given the chemistry of individual minerals and their modal proportions.

**Usage**

ModeMain(WR,sample.id="",select.oxides=TRUE,select.minerals=TRUE)
Mode(rock, mins,sample.id="")
ModeC(rock, mins,sample.id="")
ModeAll(WR)
WRComp(mins, f)
Arguments

WR  a numerical matrix; the whole-rock data to be normalized.
rock whole-rock composition of the given sample.
sample.id (optional) sample name.
select.oxides (logical) should be selected oxides used for calculation?
select.minerals (logical) should be selected minerals used for calculation?
mins composition of its main rock-forming minerals.
f their modal proportions.

Details

'Mode' uses unconstrained least-squares method taking advantage of the standard R function 
'lsfit(mins,rock,intercept=F)'. It produces results that generally do not sum up to 100 % due to
the presence of elements not used in calculation (such as water), and, or, analytical noise.
'ModeC' is the constrained variation whose output ought to sum up to 100 % by definition (Albarede
1995). As such it seems to be more appropriate in most applications.

In both cases, the printed output involves the input data, calculated modal proportions of the individual minerals, the calculated composition of the rock (using the auxiliary function 'WRComp') and
differences between the approximated and the real data (residuals).
The sum of squared residuals is a measure of fit (as a rough guide it should be less than ca. 1).
The mineral compositions are provided by a tab-delimited ASCII file, whose first row contains the names of the determined oxides, the following ones start with the mineral abbreviation and the numeric data (hence the first row has one item less than the following ones).

'ModeMain' is entry point to both 'Mode' and 'ModeC' that enables the user to read the mineral data file, select the oxides and minerals to be used in the calculation.
The options 'select.oxides=FALSE' and 'select.minerals=FALSE' read the mineral file in its entirety, using all minerals and oxides present.

'ModeAll' is a front end that performs the constrained least squares calculation for samples specified by the function selectSamples.

Value

'ModeMain', 'Mode' and 'ModeC' return a list with two items. The first of them ('table') is a matrix with the real composition of the rock and its minerals, the calculated whole-rock composition and the residuals. The second ('(un)constrained') returns calculated mineral proportions and sum of squared residuals.

'ModeAll' returns a simple matrix listing, for each rock sample, calculated proportions of rock-forming minerals and the sum of squared residuals.

'WRComp' yields a vector with the calculated whole-rock composition.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References

Molecular weights

See Also

For example of the mineral data, see file `Test_data\sazava mins.data`.

Examples

```r
# Albarede (1995) - page 7
# Calculate WRComposition of olivine gabbro containing 40 % olivine,
# 30 % diopside and 30 % plagioclase.
mins<-matrix(c(40.01,0.00,14.35,45.64,0.00,0.00,54.69,0.00,3.27,16.51,
               25.52,0.00,48.07,33.37,0.00,0.00,16.31,2.25),3,6,byrow=TRUE)
rownames(mins)<-c("ol","di","plg")
colnames(mins)<-c("SiO2","Al2O3","FeO","MgO","CaO","Na2O")
print(mins)

f<-c(0.4,0.3,0.3)
names(f)<-c("ol","di","plg")
print(f)

rock<-WRComp(mins,f)
print(rock)

# Reverse
model<-Mode(rock,mins)
mode2<-ModeC(rock,mins)
```

### Molecular weights Calculating molecularWeights of oxides

**Description**

These functions plot multiple binary plots with a common x axis, such as Harker plots.

**Usage**

```r
molecularWeight(formula)
```

**Arguments**

- `formula` a character vector of length 1, a formula of the oxide.

**Details**

So far only simple oxide formulae in form of $A_xO_y$ (where x, y are optional indexes) can be handled. The atomic weights are stored in a file `MW.data`. The atomic weights come from official CIAAW web site [http://www.ciaaw.org](http://www.ciaaw.org).

**Value**

A list with items:

- `MW` molecularWeight
- `x.atoms` number of atoms in the formula
- `x.oxygen` number of oxygens
Mullen

Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz> Vojtech Erban, <erban@sopky.cz>

References

Examples

```r
molecularWeight("SiO2")
molecularWeight("SiO2")[[1]]
oxides<-c("SiO2","TiO2","Al2O3","Fe2O3","FeO")
sapply(oxides,molecularWeight)
```

---

Mullen (1983) 10MnO-TiO2-10P2O5

Description
Assigns data for the diagram of Mullen (1983) into Figaro template (list `sheet`) and appropriate values into `x.data` and `y.data`.

Usage

```r
Mullen()
```

Details
Abbreviations used in diagram represent following geotectonic settings:

- **CAB**  
  *Calc-Alkaline Basalts*

- **IAT**  
  *Island Arc Tholeiites*

- **MORB**  
  *Mid-Ocean Ridge Basalts*

- **OIA**  
  *Ocean Island Andesites*

- **OIT**  
  *Ocean Island Tholeiites*

**Value**

- **sheet**  
  list with Figaro Style Sheet data

- **x.data, y.data**  
  $10\text{MnO}$, $TiO_2$ and $10P_2O_5$ in wt. % recalculated to 2D

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**References**


**See Also**

figaro plotDiagram
Examples

# plot the diagram
plotDiagram("Mullen", FALSE)

MullerK

Muller et al. (1992) potassic igneous rocks discrimination

Description

Assigns Figaro templates to geotectonic diagrams for potassic igneous rocks of Muller et al. (1992) into the list 'plate') and appropriate values into the list 'plate.data' for subsequent plotting.

Usage

MullerKbinary(plot.txt=getOption("gcd.plot.text"))
MullerKternary(plot.txt=getOption("gcd.plot.text"))

Arguments

plot.txt logical, annotate fields by their names?

Details

Suite of binary and ternary diagrams for discrimination of geotectonic environment of potassic igneous rocks, proposed by Muller et al. (1992) and Muller & Groves (1995). Following geotectonic settings may be deduced:

<table>
<thead>
<tr>
<th>Abbreviation used</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAP</td>
<td>Continental Arc</td>
</tr>
<tr>
<td>PAP</td>
<td>Postcollisional Arc</td>
</tr>
<tr>
<td>IOP</td>
<td>Initial Oceanic Arc</td>
</tr>
<tr>
<td>LOP</td>
<td>Late Oceanic Arc</td>
</tr>
<tr>
<td>WIP</td>
<td>Within Plate</td>
</tr>
</tbody>
</table>
Geotectonic classification of potassic rocks – Muller et al. (1992)
Note

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots. See Plate, Plate editing and figaro for details.

Author(s)

Vojtěch Janoušek, <vojtech.janousek@geology.cz>

References


See Also

Plate, Plate editing, plotPlate, figaro

Examples

plotPlate("MullerKbinary")

plotPlate("MullerKternary")
Multiple plots

Description

These functions plot multiple binary plots with a common x axis, such as Harker plots.

Usage

```
multiple(x,y=paste(colnames(WR),sep=","),
samples=rownames(WR),pch=labels$Symbol,
   col=labels$Colour,xmin=NULL,xmax=NULL,GUI=FALSE,nrow=NULL,ncol=NULL,...)
```

```
multipleMjr(x = "", y = "SiO2,TiO2,Al2O3,FeOt,MgO,CaO,Na2O,K2O,P2O5",
pch = labels$Symbol, col = labels$Colour, ...)
```

```
multipleTrc(x = "", y = "Rb,Sr,Ba,Cr,Ni,La,Ce,Y,Zr,mg#,A/CNK,K2O/Na2O",
pch = labels$Symbol, col = labels$Colour, ...)
```

Arguments

- **x**
a character vector, name of the common x axis. Formulae are OK.
- **y**
a character vector, names of oxides/elements to be plotted as y axes separated by commas. Formulae are OK.
- **nrow, ncol**
dimensions of the plots’ matrix
- **samples**
  character or numeric vector; specification of the samples to be plotted.
- **pch**
  plotting symbols.
- **col**
  plotting colours.
- **xmin, xmax**
  minimum and maximum for the x axis.
- **GUI**
  logical; is the call being made from within GCDkit GUI or not?
- **...**
  further graphical parameters: see `help(par)` for details.

Details

If x axis occurs among the arguments to be plotted as y axes, it is skipped.

Functions `multipleMjr` and `multipleTrc` are entry points supplying the default lists for major- and trace elements.

Even though as a default is assumed a list of major (SiO2,TiO2,Al2O3,FeOt,MnO,MgO,CaO,Na2O,K2O) or trace (Rb,Sr,Ba,Cr,Ni,La,Ce,Y,Zr and mg#) elements, the variable(s) to be displayed can be specified.

The easiest way is to type directly the names of the columns, separated by commas. Alternatively can be used their sequence numbers or ranges. Also built-in lists can be employed, such as `LILE`, `REE`, `major` and `HFSE` or their combinations with the column names.

These lists are simple character vectors, and additional ones can be built by the user (see Examples). Note that currently only a single, stand-alone, user-defined list can be employed as a search criterion.
In the specification of the x axis or any of the y axes can be used also arithmetic expressions, see `calcCore` for the correct syntax.

Lastly, the user is asked to enter the limits for the x axis, two numbers separated by a comma. Note that the scaling takes into account the size of the plotting symbols, i.e. the axes are extended somewhat.

**Value**

None.

**Note**

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots. See `Plate`, `Plate editing` and `figaro` for details.
mzSaturation

Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also
figaro, Plate, Plate editing

Examples
multipleMjr("SiO2")

multiple("Na2O+K2O",LILE,xmin=0)
# Plots the LILE against the sum of alkalis

multiple("FeOt/MgO","SiO2,CaO,Na2O+K2O,TiO2",pch="+",col="red",samples=1:10,cex=2.5)

multipleTrc("Zr")
# Plots the default trace-element set against the Zr

Description
Monazite saturation (Montel 1993)

Calculates monazite saturation temperatures for given major-element compositions and LREE contents of the magma.

Usage
mzSaturation(cats = milli,
REE = filterOut(WR, c("La", "Ce", "Pr","Nd","Sm","Gd"), 1),
H2O = 3, Xmz = 0)

Arguments
cats numeric matrix; whole-rock data recast to millications
REE numeric matrix with LREE concentrations - only complete set of La-Gd, excluding Eu
H2O assumed water contents of the magma
Xmz mole fractions of the REE-phosphates in monazite

Details
This function uses saturation model of Montel (1993). The formulae are as follows:

\[ LREE = \sum \left( \frac{REE_i}{at\, weight(REE_i)} \right) \]

where \( REE_i \): La, Ce, Pr, Nd, Sm, Gd.

\[ Dmz = \frac{Na + K + 2Ca}{Al} \cdot \frac{1}{Al + Si} \]
\[ T_{\text{mz.sat.C}} = \frac{13318}{9.5 + 2.34 D_{\text{mz}} + 0.3879 \sqrt{H_2O} - \ln(LREE)} - 273.15 \]

Value

Returns a matrix 'results' with the following components:

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dmz</td>
<td>distribution coefficient</td>
</tr>
<tr>
<td>Tmz.sat.C</td>
<td>monazite saturation temperature of Montel (1993) in °C</td>
</tr>
<tr>
<td>FM</td>
<td>cationic ratios</td>
</tr>
</tbody>
</table>

Plugin

Saturation.r

Author(s)

Vojtěch Janoušek, <vojtech.janousek@geology.cz>

References


NaAlK

\[ Na2O - Al2O3 - K2O \] (mol. %) diagram

Description

Assigns data for ternary diagram \( Na_2O - Al_2O_3 - K_2O \) (mol. %) into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'. Calculates molar concentrations of alkalis and alumina, as well as several molar ratios involving these three oxides.

Usage

NaAlK()}
Details

Ternary plot $Na_2O - Al_2O_3 - K_2O$ (mol. %). Dashed lines define the following compositional fields (all oxides are expressed in mol. %):

- Peraluminous + Metaluminous (Shand 1943) \( \frac{(Na_2O + K_2O)}{Al_2O_3} < 1 \)
- Peralkaline (Shand 1943) \( \frac{(Na_2O + K_2O)}{Al_2O_3} > 1 \)
- Perpotassic \( K_2O/Al_2O_3 > 1 \) and \( K_2O/Na_2O > 1 \)
- Potassic \( 1 < K_2O/Na_2O < 3 \)
- Ultrapotassic \( K_2O/Na_2O >= 3 \)

The molar ratio of $K_2O/Na_2O >= 3$, is equivalent to $K_2O/Na_2O >= 2$ in wt. %, i.e. to the definition of ultrapotassic igneous rocks by Foley et al. (1987).

Value

<table>
<thead>
<tr>
<th>sheet</th>
<th>list with Figaro Style Sheet data</th>
</tr>
</thead>
<tbody>
<tr>
<td>x.data, y.data</td>
<td>$Na_2O, Al_2O_3$ and $K_2O$ contents in mol.%, transformed into 2D</td>
</tr>
<tr>
<td>Na20</td>
<td>$Na_2O$ in mol. %</td>
</tr>
<tr>
<td>Al203</td>
<td>$Al_2O_3$ in mol. %</td>
</tr>
<tr>
<td>K20</td>
<td>$K_2O$ in mol. %</td>
</tr>
<tr>
<td>(Na20+K20)/Al203</td>
<td>molecular ratio $\frac{(Na_2O + K_2O)}{Al_2O_3}$</td>
</tr>
<tr>
<td>K20/Al203</td>
<td>molecular ratio $\frac{K_2O}{Al_2O_3}$</td>
</tr>
<tr>
<td>K20/Na20</td>
<td>molecular ratio $\frac{K_2O}{Na_2O}$</td>
</tr>
</tbody>
</table>
Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


Shand (1943) Eruptive Rocks. John Wiley & Sons

See Also

classify figaro plotDiagram Shand

Examples

#Within GCDkit, the plot is called using following auxiliary functions:
#To Classify data stored in WR (Groups by diagram)
classify("NaAlK")

#To plot data stored in WR or its subset (menu Classification)
plotDiagram("NaAlK", FALSE)

Niggli  Niggli’s values

Description

Calculates cationic parameters of Niggli (1948).

Usage

Niggli(WR, precision = getOption("gcd.digits"))

Arguments

WR a numerical matrix; the whole-rock data to be normalized.

precision precision of the result.

Details

The calculated parameters are:

si, al, fm, c, alk, k, mg, ti, p, c/fm, qz

Value

A numeric matrix ‘results’.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>
References


Examples

```r
data(sazava)
accessVar("sazava")
Niggli(WR)
```

---

**OConnor**

*Classification diagram for siliceous igneous rocks, based on Fsp composition (O’Connor 1965)*

---

**Description**

Assigns data for O’Connor’s triangular diagram into Figaro template (list ‘sheet’) and appropriate values into ‘x.data’ and ‘y.data’.

**Usage**

```r
OConnorVolc()
```

```r
OConnorPlut(ab=NULL, an=NULL, or=NULL)
```

**Arguments**

```r
ab, an, or
```

character; specification of the plotting variables.

**Details**

The O’Connor’s triangular diagram is based on combination of Albite, Anorthite and K-feldspar modal or normative data. While the function ‘OConnorPlut’ can plot either modal or normative diagrams for plutonic rocks, ‘OConnorVolc’ is to be used exclusively with normative data computed from chemical compositions of volcanic rocks.
In fact, the triangle represents projection of the Quartz - K-feldspar - Albite - Anorthite tetrahedron. All three diagrams are designed for quartz-rich rocks, i.e. those with quartz contents higher than 10 such silica-rich samples, the rock type can be determined purely on the basis of the feldspars’ proportions.

As the specific version of the normative calculation is not mentioned in the original paper by 'O’Connor (1965)', the function 'CIPw', designed after 'Hutchison (1974, 1975)' was implemented.
as a default calculation scheme.

Alternatively, the plotting variables can be present already in the dataset (variable WR). The variables to be plotted can be then specified upon call or can be selected using the function 'selectColumnLabel'. The samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see selectSubset for details.

Value

- sheet: list with Figaro Style Sheet data
- x.data, y.data: An, Ab and Or data (see details) transformed to orthogonal coordinates

Author(s)

Vojtech Erban, <erban@sopky.cz>
Vojtech Janousek, <vojtech.janousek@geology.cz>
Jean-Francois Moyen, <jfmoyen@gmail.com>

References


See Also

classify, figaro, CIPW, plotDiagram

Examples

```r
plotDiagram("OConnorVolc",FALSE)
classify("OConnorVolc")
results<-Mesonorm(WR)
addResults()
plotDiagram("OConnorPlut",FALSE,ab="Albite",an="Anorthite",or="Orthoclase")
```

---

**overplotDataset**  Adding another dataset to the current plot

**Description**

This function allows overplotting new data points stored in the memory onto any type of single Figaro-compatible plots (or their plates). This can be done either into foreground or into background.
overplotDataset

Usage

overplotDataset(reference.dataset=NULL, underplotting=FALSE, transp=0, pch=NULL, col=NULL, cex=NULL, ...) underplotDataset(reference.dataset=NULL, transp=0, ...)

Arguments

reference.dataset
  object name (given as a character string or unquoted); the dataset to be added to the current diagram. See Details.
underplotting logical; should be the reference dataset added at the background?
transp numeric, 0-1; transparency of the background dataset (in underplotting).
pch plotting symbol(s) for the foreground dataset.
col plotting colour(s) for the foreground dataset.
cex numeric; relative size of the plotting symbol(s) for the foreground dataset.
... additional parameters to the underlying plotting function(s). See Details.

Details

These are front-ends to the functions 'figOverplot' and 'figOverplotDiagram', invoked as appropriate. However, the functions 'overplotDataset' and 'underplotDataset' work correctly also on plates.

Also underplotDataset is just a convenience function, calling overplotDataset with the parameter underplotting=TRUE.

Most typically, reference.dataset is a (quoted) name of a dataset stored in memory. Alternatively, a (unquoted) name of a global variable can be specified.

Plotting parameters 'pch', 'col' and 'cex' are available only for overplotting.

On the other hand, transparency can be set only in underplotting. See 'setTransparency' for further info.

Argument '...' can supply additional parameters to the original plotting functions (e.g., 'TAS') invoked by 'plotDiagram' or 'plateExtract'.

Value

(Invisibly) name of the reference dataset.

Note

This function is a front-end, truly a 'mother of all' specialized and less versatile overplotting functions such as 'figOverplot', 'figOverplotDiagram' or 'figAddReservoirs(just.draw=TRUE)'. Please use 'overplotDataset' instead, unless permanent addition to the plot is required. For such cases, 'figAddReservoirs' 'plateAddReservoirs' in their default form, i.e. with argument just.draw=FALSE, are the functions of choice.

Warning

NB that the points for the overplotted dataset do not form a part of the template, and thus will vanish upon redrawing, zooming .....

overplotDataset
Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

'figOverplot' 'figOverplotDiagram' 'figAddReservoirs' 'plateAddReservoirs'
'peekDataset' 'pokeDataset' 'purgeDatasets'
'setTransparency' 'plotDiagram' 'plateExtract'

Examples

# Loading two testing datasets
data(sazava)
accessVar("sazava")
data(blatna)
accessVar("blatna")
setCex(1.5)
pokeDataset("blatna", overwrite.warn=FALSE) # Store a version with larger symbols

# Single plots
peekDataset("blatna")
plotDiagram("DebonPQ",FALSE,TRUE)
figRemove()
overplotDataset("sazava",cex=2,col="darkred",pch=15)

plotDiagram("DebonPQ",FALSE,TRUE)
figRemove()
underplotDataset("sazava",transp=0.5)

plateExtract("PearceGranite",2,main=" ")
overplotDataset("sazava")

# Spiderplots
peekDataset("blatna")
spider(WR,"Boynton",1,1000,cex=0,join=TRUE,offset=TRUE,
        centered=FALSE,xrotate=FALSE,xaxs="r")
overplotDataset("sazava")

spider(WR,"Boynton",0.1,1000,field=TRUE,fill.col=TRUE,shaded.col="gray")
# Blatna as gray field
overplotDataset("sazava")

# A simple plate
peekDataset("blatna")
multiple("SiO2",y="TiO2,Al2O3,FeOt,MgO,CaO,Na20,K20,P2O5",nrow=3,ncol=3,main=" ")
plateCex(1.8)
plateCexLab(1.3)
overplotDataset("sazava")

# A plate of classification diagrams
peekDataset("blatna")
multiplePerPage(4,nrow=2,ncol=2,title="A classification plate")
plotDiagram("DebonPQ",FALSE,FALSE,main=" ")
plotDiagram("DebonBA",FALSE,FALSE,main=" ")
oxide2oxide

Recalculation of one oxide to a different one

Description

Returns a factor needed to multiply concentrations of an element given as an oxide (in wt %) to a
different target oxide (of the same element).

Usage

oxide2oxide(formula1, formula2)

Arguments

formula1 character: the oxide which is to be recalculated
formula2 character: the target oxide

Value

A factor for recalculation.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

oxide2ppm, ppm2oxide, molecularWeight

Examples

oxide2oxide("FeO","Fe2O3")
oxide2oxide("Mn2O3","MnO")
oxide2ppm

Description
Recasts concentrations of an oxide (in wt. %) to that of appropriate cation (in ppm).

Usage
oxide2ppm(formula, where="WR")

Arguments
formula character: the oxide which is to be recalculated
where character: a name of matrix or dataframe with the data to be recalculated

Value
A numeric matrix with one column containing the recalculated concentrations of the given cation (ppm) for individual samples.

Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also
ppm2oxide, oxide2oxide, molecularWeight

Examples
data(sazava)
accessVar("sazava")
oxide2ppm("K2O")

pairsCorr

Statistics: Correlation

Description
Plots a matrix of scatterplots in the lower panel and one of other pre-defined panel functions in the upper.

Usage
pairsCorr(elems = major)
pairsMjr()
pairsTrc()
Arguments
elems  list of desired elements

Details
The samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSamples` for details.
Even though a list of major elements is assumed as a default, different variables can be specified by the function `selectColumnsLabels`.

The upper panels to choose from are:

- `panel.corr`  Prints correlations, with size proportional to the correlations;
- `panel.cov`    Prints covariances;
- `panel.smooth` Fits smooth trendlines;
- `panel.hist`   Plots frequency histograms.
Value

None.

Warning

Names of existing numeric data columns and not formulae involving these can be handled at this stage.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

Examples

pairsCorr(LILE)
pairsMjr()
pairsTrc()

# user-defined list
my.elems<-c("Rb","Sr","Ba")
pairsCorr(my.elems)

Description

Assigns data for $\text{Al}_2\text{O}_3/\text{SiO}_2$ vs. $\text{MgO}/\text{SiO}_2$ binary diagram into Figaro template (list ”sheet”) and appropriate values into ”x.data” and ”y.data”.

Usage

Paulick()

Arguments

None.

Details

According to Paulick et al. (2006), the global analyses of mantle peridotites form a ”Terrestrial Array” in the binary plot $\text{Al}_2\text{O}_3/\text{SiO}_2$ vs. $\text{MgO}/\text{SiO}_2$. This linear correlation reflects the successive magmatic depletion of a primitive mantle and highly depleted compositions are characterized by low $\text{Al}_2\text{O}_3/\text{SiO}_2$ values (<0.01; Jagoutz et al. 1979; Hart and Zindler 1986).
Value

sheet  list with Figaro Style Sheet data
x.data x coordinates
y.data y coordinates

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

figaro plotDiagram
pdfAll

Examples

# plot the diagram
# assuming a dataset is loaded, of course!
## Not run:
plotDiagram("Paulick",FALSE,TRUE)
## End(Not run)

pdfAll filename=NULL

Save all graphics to PDF

Description

Saves all graphical windows to a single PDF file.

Usage

pdfAll(filename=NULL)

Arguments

filename a name of file for saving the output.

Details

The function prompts for filename under which it saves all graphical windows, each on a separate page. PDF is the most portable format, that should preserve practically the same layout on all platforms.

Individual diagram can be saved from a menu that appears after clicking on the appropriate graphical window ("File|Save as|PDF").

Value

None.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

`psAll` `pdf`
Description

Plots data stored in 'WR' (or its subset) into Pearce and Cann’s diagrams.

Usage

Cann(plot.txt = getOption("gcd.plot.text"))

Arguments

plot.txt logical, annotate fields by their names?

Details

Set of two triangular and one binary diagram, proposed by Pearce & Cann (1973).

Following abbreviations are used:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAT</td>
<td>Low-K Tholeiites</td>
</tr>
<tr>
<td>MORB</td>
<td>Ocean Floor Basalts</td>
</tr>
<tr>
<td>CAB</td>
<td>Island Arc Basalts</td>
</tr>
<tr>
<td>WPB</td>
<td>Within Plate Basalts</td>
</tr>
</tbody>
</table>

Note

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots. See Plate, Plate editing and figaro for details.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>
References


See Also

Plate, Plate editing, plotPlate, figaro

Examples

#plot the diagrams
plotPlate("Cann")

Description

Assigns data for the diagram of Pearce & Norry (1979) into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

Usage

Norry()

Details

Diagram proposed by Pearce & Norry (1979) for geotectonic discrimination between basaltic rocks from distinct geotectonic positions:

Within-plate Basalts
Island-arc basalts
Mid-ocean Ridge Basalts
Value
  sheet       list with Figaro Style Sheet data
  x.data      Zr ppm
  y.data      Zr/Y by weight

Author(s)
  Vojtech Janousek, <vojtech.janousek@geology.cz>

References

See Also
  figaro plotDiagram

Examples
  #plot the diagram
  plotDiagram("Norry",FALSE)
Description

Assigns data for a Th/Yb vs. Nb/Yb diagram into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

Usage

PearceNbThYb(reservoirs=TRUE,xmin=0.1,xmax=1000,ymin=0.01,ymax=100)

Arguments

reservoirs logical, should be plotted average NMORB, EMORB and OIB?

xmin, xmax numeric, limits for the x axis.

ymin, ymax numeric, limits for the y axis.

Details

This diagram (Th/Yb vs. Nb/Yb) has been developed by J. Pearce in the 2000s to characterize (and discriminate) arc magmatism. The current version is based on paper by Pearce (2008) dealing with oceanic basalts, though. According to this author, Th-Nb serves as a ‘crustal input proxy’ and hence for demonstrating an oceanic, non-subduction setting.

The 'MORB-OIB array' at the bottom extends from N-MORB to OIB (plotted for reference are average compositions of NMORB, EMORB and OIB taken from Sun and McDonough (1989). Melting of the metasomatized mantle yields trends parallel to the mantle array.

Arc lavas, formed by fluxed melting of the mantle, are shifted above the mantle array; the same effects have mantle-derived magma-crust interactions. The top dashed line is the outer limit of typical arc lavas, but there is a great deal of variation.
Value

- sheet: list with Figaro Style Sheet data
- x.data: Nb/Yb
- y.data: Th/Yb

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz> and Jean-Francois Moyen, <jfmoyen@gmail.com>

References


See Also

- figaro plotDiagram PearceNbTiYb
Examples

```r
# Plot the diagram
plotDiagram("PearceNbThYb",FALSE,FALSE,reservoirs=TRUE)
plotDiagram("PearceNbThYb",FALSE,FALSE,reservoirs=FALSE)
```

---

Pearce Nb-Ti-Yb  

*Pearce (2008) Nb/Yb-TiO2/Yb diagram*

---

Description

Assigns data for a TiO$_2$/Yb vs. Nb/Yb diagram into Figaro template (list ‘sheet’) and appropriate values into ‘x.data’ and ‘y.data’.

Usage

```r
PearceNbTiYb(reservoirs=TRUE,xmin=0.1,xmax=100,ymin=0.1,ymax=10)
```

Arguments

- `reservoirs` logical, should be plotted average NMORB, EMORB and OIB?
- `xmin,xmax` numeric, limits for the x axis.
- `ymin,ymax` numeric, limits for the y axis.

Details

The diagram TiO$_2$/Yb vs. Nb/Yb serves as ‘melting depth proxy’ and hence for indicating mantle temperature and thickness of the conductive lithosphere (*Pearce 2008*). It distinguishes basalts, which have originated by shallow melting, out of garnet stability field (‘MORB array’) from those spanning from deep melting with garnet in the residue (‘OIB array’). Plotted for reference are average compositions of NMORB, EMORB and OIB taken from *Sun and McDonough (1989)*.
Value

- Sheet: list with Figaro Style Sheet data
- x.data: Nb/Yb
- y.data: TiO2/Yb

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

figaro plotDiagram PearceNbThYb
Examples

```r
# Plot the diagram
plotDiagram("PearceNbTiYb", FALSE, FALSE, reservoirs=TRUE)
plotDiagram("PearceNbTiYb", FALSE, FALSE, reservoirs=FALSE)
```

Description

Assigns data for the diagram of *Pearce (1982)* into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

![Zr-Ti diagram](image)

Usage

`Pearce1982()`

Details

Diagram proposed by *Pearce (1982)* for geotectonic discrimination between lavas from distinct geotectonic positions:

- **Within-plate lavas**
- **Island-arc lavas**
- **Mid-ocean Ridge Basalts**
Value

sheet list with Figaro Style Sheet data
x.data Zr ppm
y.data Ti ppm

Author(s)

Jean-Francois Moyen, <jfmoyen@gmail.com>

References


See Also

figaro plotDiagram

Examples

#plot the diagram
plotDiagram("Pearce1982",FALSE)

---

Pearce1996

Nb/Y - Zr/Ti diagram (Winchester + Floyd 1977, modified by Pearce 1996)

Description

Assigns data for Nb/Y vs. Zr/Ti diagram into Figaro template (list `sheet`) and appropriate values into `x.data` and `y.data`.

Usage

Pearce1996()

Details

Classification diagram proposed by Winchester & Floyd (1977) using incompatible element ratios (Nb/Y vs. Zr/Ti). As the original plot has been designed prior to the publication of the TAS diagram Le Bas et al. 1986, the field definition has been subsequently modified by Pearce (1996).
The following fields are defined:

(Subalkaline) Basalt
Alkali basalt
Foidite
Andesite/Basaltic andesite
Trachyandesite
Tephriphonolite
Rhyolite/Dacite
Trachyte
Phonolite
Alkali Rhyolite

Value

sheet list with Figaro Style Sheet data
x.data Nb/Y wt. % ratio
y.data Zr/Ti wt. % ratio

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>
References


See Also

WinFloyd1 classify figaro plotDiagram

Examples

#Within GCDkit, the plot is called using following auxiliary functions:
#To Classify data stored in WR (Groups by diagram)
classify("Pearce1996")

#To plot data stored in WR or its subset (menu Classification)
plotDiagram("Pearce1996", FALSE)

---

PearceDestructive1  
**Ta/Yb - K2O/Yb (Pearce 1982)**

Description

Assigns data for Ta/Yb vs. \( K_2O/Yb \) binary diagram into Figaro template (list ‘sheet’) and appropriate values into ‘x.data’ and ‘y.data’.

Usage

PearceDestructive1()

Arguments

None.

Details

The binary plot Ta/Yb vs. \( K_2O/Yb \) of *Pearce* (1982) serves for geotectonic discrimination of volcanic-arc basalts. In addition, it can distinguish between the tholeiitic, calc-alkaline and shoshonitic types.
Value

- `sheet` : list with Figaro Style Sheet data
- `x.data` : x coordinates
- `y.data` : y coordinates

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


Coordinates and graph layout are taken from website of Kurt Hollocher.

See Also

- figaro plotDiagram

Examples

```r
# plot the diagram
# assuming a dataset is loaded, of course!
```
## Not run:
plotDiagram("PearceDestructivel",FALSE,TRUE)
## End(Not run)

**PearceDestructive2**  
**Ta/Yb - Th/Yb (Pearce 1982)**

### Description
Assigns data for Ta/Yb vs. Th/Yb binary diagram into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

### Usage
PearceDestructive2()

### Arguments
None.

### Details
The binary plot Ta/Yb vs. Th/Yb of **Pearce (1982)** serves for geotectonic discrimination of volcanic-arc basalts. In addition, it can distinguish between the tholeiitic, calc-alkaline and shoshonitic types.
Value

```
  sheet  list with Figaro Style Sheet data
  x.data  x coordinates
  y.data  y coordinates
```

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


Coordinates and graph layout are taken from website of Kurt Hollocher.

See Also

figaro plotDiagram

Examples

```
# plot the diagram
# assuming a dataset is loaded, of course!
## Not run:
plotDiagram("PearceDestructive3",FALSE,TRUE)

## End(Not run)
```

---

PearceDestructive3  Nb/Y - Ti/Y (Pearce 1982)

Description

Assigns data for Nb/Y vs. Ti/Y binary diagram into Figaro template (list ‘sheet’) and appropriate values into ‘x.data’ and ‘y.data’.

Usage

`PearceDestructive3()`

Arguments

None.
Details

The binary plot Nb/Y vs. Ti/Y of Pearce (1982) serves for geotectonic discrimination of volcanic-arc basalts. In addition, it can distinguish between the tholeiitic, calc-alkaline and shoshonitic types.

Value
(sheet list with Figaro Style Sheet data
x.data x coordinates
y.data y coordinates

Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>

References

Coordinates and graph layout are taken from website of Kurt Hollocher.

See Also
figaro plotDiagram
Examples

# plot the diagram
# assuming a dataset is loaded, of course!
## Not run:
plotDiagram("PearceDestructive3",FALSE,TRUE)
## End(Not run)

## Description

Assigns data for the MgO-FeOt-Al$_2$O$_3$ triangle proposed by Pearce et al. (1977) into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

Usage

PearceEtAl()

Details

Triangular diagram with apices MgO, FeOt and Al$_2$O$_3$, proposed by Pearce et al. (1977). The boundaries were defined solely for subalkaline volcanic rocks with SiO$_2$ between 51-56 wt %. Following geotectonic positions may be identified using the diagram:
**Spread**ing Center Island (or inter-plate island) - oceanic islands adjacent to ocean-ridge spreading, such as Iceland or Galapagos; the authors ‘do not consider this field well established’.

**Orogenic**

**Ocean Ridge and Floor**

**Ocean Island**

**Continental**

**Value**

- **sheet** : list with Figaro Style Sheet data
- **x.data, y.data** : MgO, FeOt and $Al_2O_3$ in wt. % recalculated to two dimensions

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**References**


**See Also**

figaro plotDiagram

**Examples**

```r
# Plot the diagram
plotDiagram("PearceEtAl",FALSE)
```

**Description**

Assigns Figaro templates to Pearce’s geotectonic diagrams for granitoids into the list ‘plate’) and appropriate values into the list ‘plate.data’ for subsequent plotting.

**Usage**

`PearceGranite(plot.txt = getOption("gcd.plot.text"))`

**Arguments**

- **plot.txt** : logical, annotate fields by their names?
Details

Suite of four diagrams for discrimination of geotectonic environment of granitoid rocks, proposed by Pearce et al. (1984). It is based on combination of five trace elements (namely Y, Nb, Rb, Yb and Ta).

Following geotectonic settings may be deduced:

<table>
<thead>
<tr>
<th>Abbreviation used</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORG</td>
<td>Ocean Ridge Granites</td>
</tr>
<tr>
<td>VAG</td>
<td>Volcanic Arc Granites</td>
</tr>
<tr>
<td>WPG</td>
<td>Within Plate Granites</td>
</tr>
<tr>
<td>COLG</td>
<td>Collision Granites</td>
</tr>
</tbody>
</table>
Note
This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots. See Plate, Plate editing and figaro for details.

Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>

References

See Also
Plate, Plate editing, plotPlate, figaro

Examples
plotPlate("PearceGranite")
To employ boundaries as originally defined by Peccerillo & Taylor (1976), change the value of variable 'extrapolated' to 'FALSE' in the file '[R-root]\library\GCDkit\Diagrams\Classification\PeceTaylor.r'. Also note that the second value for the middle boundary (i.e. [52,1.5]) is in the original paper obviously misquoted as 1.3.

Rocks with composition falling beyond defined boundaries are labeled 'undefined' by the 'classify' function.

For comparison with similar diagrams used by other authors see Rickwood (1989).

**Value**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sheet</td>
<td>list with Figaro Style Sheet data</td>
</tr>
<tr>
<td>x.data</td>
<td>SiO2 weight percent</td>
</tr>
<tr>
<td>y.data</td>
<td>K2O weight percent</td>
</tr>
</tbody>
</table>

**Author(s)**

Vojtech Erban, <erban@sopky.cz>
& Vojtech Janousek, <vojtech.janousek@geology.cz>

**References**


peekDataset

Description

Both functions restore the previously stored dataset and make it current.

Usage

peekDataset(which.dataset=NULL)
selectDataset()

Arguments

which.dataset numeric or character; a sequence number or name of the stored dataset.

Details

The function 'peekDataset' restores a dataset saved previously into memory by the function 'pokeDataset'. This means that it assigns all global variables specified by individual items of the list 'WRCube'.

These typically are: 'WR', 'WRanh', 'milli', 'labels', 'filename', 'groups' and 'grouping'.

The function 'selectDataset' provides a graphical interface to 'peekDataset', i.e. shows a list box filled by the names of datasets currently stored in the memory.

Value

None. But several global variables, among others 'WR', 'WRanh', 'milli' and 'labels', are affected. The name of the current dataset is stored in 'dataset.name'.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

'pokeDataset', 'purgeDatasets'
Examples

data(sazava)
accessVar("sazava")
# stored as sazava in WRCube

assignColVar("MgO","blues")
assign1symb(15)
# store a new copy in the WRCube
pokeDataset("coloured sazava")

data(swiss)
accessVar("swiss")
# stored as swiss in WRCube

peekDataset("sazava")
binary("SiO2","Ba")

peekDataset("coloured sazava")
binary("SiO2","Ba")

peekDataset("swiss")
binary("Catholic","Education",pch=15,col="darkgreen")

peekDataset(2)
binary("SiO2","Sr")

peterplot

Anomaly plot

Description

This function plots a conventional binary diagram but the type and size of the plotting symbols is assigned according to the distribution of a third, conditioning variable.

Usage

peterplot(xaxis = "", yaxis = "", zaxis = "", ident = FALSE,
   scaling.small = labels[1,"Size"], scaling.big = 2 * scaling.small,
   assign.symbols = FALSE)

Arguments

xaxis, yaxis character; specification of the axes
zaxis character; conditioning variable
ident logical; identify the individual points?
scaling.small scaling factor for the smaller plotting symbols
scaling.big scaling factor for the larger plotting symbols
assign.symbols logical; should be the plotting symbols and their sizes assigned permanently?
Details

If no parameters `xaxis`, `yaxis` and `zaxis` are specified, the user is prompted to do so interactively. The plotting symbols are assigned as follows: the values within 25 quartiles) obtain a dot, the higher ones are denoted by '+' and lower ones by '-' . If the given value is an outlier, its plotting size is doubled.

Optionally, the user can assign the plotting symbols and their sizes permanently, for use in other diagrams throughout the system.

Value

May modify the variable `cex`, as well as the codes of plotting symbols stored in the data frame labels.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


Examples

```
peterplot("SiO2","MgO","K2O")
peterplot("SiO2","MgO","K2O",assign.symbols=TRUE)
plotDiagram("TAS",FALSE)
```
phasePropPlot

Stacked barplot of temperature vs. phase proportions.

Description

This function makes a stacked barplot of phase proportions, typically of minerals with, or without, melt.

Usage

phasePropPlot(mat, renormalize = TRUE, col = NULL, palette = "jet.colors", leg.pos = "bottomleft", leg.bg = "#FFFFFFAA", xlab = expression(Temperature\degree{\textdegree}C), ylab = "Phase proportions", xlim = NULL, ylim = c(0, 1), border = "white", main="")

Arguments

mat

a numeric matrix with phase proportions in columns, and temperature in C in rows.

renormalize

logical, should the data in mat renormalized to a sum of 1?

col

list of colours for each of the phases.

palette

palette name.

leg.pos

position of the legend.

leg.bg

background colour for the legend.

xlab

character or expression; label for the x axis.

ylab

character or expression; label for the y axis.

xlim

limits for the x axis.

ylim

limits for the y axis.

border

colour for the border for each of the bars.

main

character; main title for the plot.

Details

The input is a matrix with phase proportions in columns, their names in colnames and variable (by default a temperature in °C) in rownames.

If `col = NULL` and `palette` is specified, then the corresponding number of colours are taken therefrom. Then the first column of data, typically a melt, is shown in gray.
The function assigns data for the diagram into a Figaro template (list 'sheet'), centers of intervals into 'x.data' (not used for the x axis labeling) and the plotting matrix into 'y.data'. The values for labeling the x axis are taken from rownames of 'y.data'.

**Value**

- **sheet**: list with Figaro Style Sheet data.
- **x.data**: See Details.
- **y.data**: See Details.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

- figaro

**Examples**

```r
min.prop<-matrix(c(0.1,0.2,0.5,
                   0.2,0.25,0.25,
                   0.5,0.4,0.15,
                   0.2,0.15,0.1),
                   nrow=3,ncol=4,dimnames=list(seq(750,850,by=50),c("Liq","Cpx","Opx","Pl")))
```
```r
phasePropPlot(min.prop, palette="jet.colors", ylab="vol. percent",
    main="Plot of mineral proportions")

phasePropPlot(min.prop, col=1:4)

phasePropPlot(min.prop, col=heat.colors(4))
data(blatna)
accessVar("blatna")
windows(10, 5)
i<-names(sort(WR[,"SiO2"]))
phasePropPlot(WR[i, major])
```

### Description

Functions to set up, save or load a so-called 'plate', i.e. a regular grid of slots to accommodate (any mixture of) binary or ternary plots, spiderplots or such alike. For instance, Harker plots are implemented using the plate concept.

### Usage

- `multiplePerPage(which=NULL, nrow=NULL, ncol=NULL, title="Plate", dummy=FALSE)`
- `Plate(scr=NULL)`
- `plateRedraw(device="windows", filename=NULL, colormodel="rgb")`
- `platePS(colormodel="rgb", filename=NULL)`
- `plateSave()`
- `plateLoad()`

### Arguments

- `which` : total number of slots to be occupied by individual diagrams.
- `nrow` : number of rows in the plots' matrix.
- `ncol` : number of columns in the plots' matrix.
- `title` : title for the whole plate.
- `dummy` : logical; if TRUE, dummy plots are shown. See Details.
- `scr` : (optional) number of screen to be selected.
- `device` : output device; either 'windows' or 'postscript'.
- `filename` : name of file if output is to be redirected to Postscript.
- `colormodel` : color mode for Postscript; 'rgb' or 'gray'.
The function `multiplePerPage` serves to setting up a matrix of slots, each of which could be taken by a single Figaro-compatible diagram (a binary plot, a ternary plot, a spiderplot, ...). If 'which' is NULL, the function asks for their number, and then suggests number of rows (`nrow`) and columns (`ncol`) for the matrix arrangement.

If desired, the slots can be filled by the so-called 'dummy plots', i.e. gray boxes showing the exact position and the size of each of them.

If 'which' is an integer, specified number of slots is allocated. Alternatively, this argument may represent a vector containing any mixture of names of diagrams that can be plotted by the function `plotDiagram` or even plotting commands themselves used to fill the individual slots directly. See Examples.

Once set up, a single slot can be selected for further work using the function `Plate`. The function can be called directly, with the number of the screen desired. If none is specified, a red box-like cursor appears in the graphical window, which can be moved around using the cursor keys, Spacebar or by mouse. The appropriate slot can be chosen by left mouse button or by pressing Enter. Right-click anywhere on the plate invokes a context menu which enables several actions:

<table>
<thead>
<tr>
<th>Menu item</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduce plot</td>
<td>Select a new Figaro-compatible diagram for this slot.</td>
</tr>
<tr>
<td>Plot editing</td>
<td>Modify the existing diagram (like the menu Plot editing for stand alone plots).</td>
</tr>
<tr>
<td>Plate editing</td>
<td>Functions to modify the overall plate properties or all its diagrams simultaneously.</td>
</tr>
</tbody>
</table>

The function `plateRedraw` serves for reploting a 'clean!' version of the whole plate, e.g. for saving/printing. For this purpose, its output can be redirected to Postscript, either in colour or as black and white. As a wrapper for the Postscript output serves the function `platePS`.

The functions `plateSave` and `plateLoad` are designed to save and retrieve definitions of plates (Figaro sheets and the relevant data) for later use. The default suffix for the saved plates is 'mgr'. Note that only the data needed for the plotting ('x.data', 'y.data') are stored in the 'mgr' files. Thus the data set currently in memory (e.g., variables 'WR', 'labels', ...) is unaffected by the function `plateLoad`.

Starting with GCDKit version 3, the plates concept is used by some built-in functions, such as 'Multiple plots' (function `multiple`) or 'Multiple plots by groups' (function `figMulti`).

### Value

- `plate`: list of Figaro definitions for individual diagrams
- `plate.data`: list containing 'x.data' and 'y.data' for each of them

### Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

### See Also

Plate editing, plotPlate, multiple, figMulti, plot, binary, ternary, spider, figaro, figLoad, figSave

### Examples

```r
data<-loadData("sazava.data", sep="\t")
multiplePerPage(which=cbinary("K2O/Na2O", "K2O/Na2O"),
```
Plate editing

A collection of functions to modify the properties of a plate (or all its diagrams) simultaneously.

Usage

plateXLim(xlim=NULL)
plateYLim(ylim=NULL)
plate0YLim()
plateCex(n=NULL)
plateCexLab(n=NULL)
plateCexMain(n=NULL)
plateAnnotationsRemove()
platePch(pch=NULL)
plateCol(col=NULL)
plateBW()
plateExpand(scr=NULL)
plateExtract(diagram, which=NULL, main=NULL, calc.only=FALSE,...)

Arguments

xlim scaling for the x axis
ylim scaling for the y axis
n relative size (use n = 1 for normal one).
pch plotting symbol specification, either as string or a numeric code (showSymbols).
col colour specification, either by its English name, or by a numeric code (showColours).
scr number of screen to be expanded.
Plate editing

Diagram

name of the function plotting a plate.

Which

sequential number of plot in its definition.

Main

optional alternative main title to the diagram.

Calc.only

logical; should be performed only calculations, without plotting?

...additional parameters to the diagram (plate) plotting function.

Details

The functions serve to change properties of all particular diagrams forming the given plate. They can be used to set up a uniform size of plotting symbols (`plateCex`), specify a main title (`plateCexMain`), set up a uniform size of the axes’ labels (`plateCexLab`), remove the annotations of classification fields (`plateAnnotationsRemove`), specifying a uniform plotting symbol (`platePch`) and/or colour (`plateCol`) to all plots, or set them into black and white (`plateBW`). If the same variable is plotted as x or y axis in all diagrams forming the plate (e.g., on Harker plots), it can be scaled by means of the functions `plateXLim` and `plateYLim`. Using the command `plateYLim` it is possible to set the origin of all non-logarithmic y axes to zero. The function `plateExpand` displays a zoomed up version of the selected diagram in a separate window.

The function `plateExtract` extracts a Figaro definition of a single plot from a plate plotted by the function `diagram`. If `calc.only` is `FALSE`, the diagram is plotted, either into a separate window or into current slot, if the active plot is a plate.

Value

None.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

Plate, plotPlate, figaro, figScale, figCol, showSymbols, showColours

Examples

data<-loadData("sazava.data",sep="\t")
showSymbols()
showColours()
multiplePerPage(which=c("binary("K20/Na2O","Rb","DebonPQ","AFM","PeceTaylor","Shand"))

plateCex(0.5)

plateCex(2)

platePch(11)
plateAddReservoirs

Plate editing: plateAddReservoirs

Description

This function enables adding selected data from typical geochemical reservoirs (e.g., Upper Continental Crust, MORB . . . ), ideal mineral compositions, results of petrogenetic modelling or just another dataset used for comparison to a plate of Figaro-compatible plots.

Usage

plateAddReservoirs(autoscale=FALSE, var.name="reservoirs.data", sample.names=NULL, reserv.condition=NULL, labs=NULL, pch="*", col="darkblue", cex=1, type="p", just.draw=FALSE,...)

Arguments

autoscale logical; should the scaling be changed so that all the plotted data fit in?
var.name text; either 'reservoirs.data', 'idealmins.data' or a name of a global variable. See Details.
sample.names character vector; names of reservoirs, ideal minerals or samples to be plotted.
reserv.condition text; regular expression specifying names of reservoirs, ideal minerals or samples to be plotted.
labs text; optional labels for the individual reservoirs.
pch plotting symbols.
col  plotting colours.
cex  numeric; relative size of the plotting symbols.
type  character; plot type; see plot.default.
just.draw  logical; if FALSE, the overplotted bit is added permanently, i.e. the Figaro template is also affected.
...  additional parameters to the plotting function. See figOverplot.

Details

The function `plateAddReservoirs` overplots compositions of selected geochemical reservoirs (from the file `reservoirs.data`, see selectNorm for the file structure as well as relevant references) or ideal minerals (from the file `idealmins.data`) onto a current plate. Alternatively, if the name of a numeric matrix or dataframe in the global environment is provided via the argument `var.name`, data from this object are used (see Examples). The selection of samples is governed either by `sample.names` or by `reserv.condition` parameters.

Optional argument `labs` can provide alternative, perhaps abbreviated textual labels to the points plotted.

Please note that this function is so far available for spiderplots, binary and ternary plots only and no special indexes, e.g. for Debon and Le Fort’s plots, are calculated.

By default, the overplotted information is added permanently but this behaviour is controlled by the argument `just.draw`.

Value

A list of numeric matrices with the overplotted analyses from the reference dataset.

Warning

If `just.draw=FALSE`, the points for the reference dataset do not become a part of the template, and thus will vanish upon redrawing, zooming .... See Examples.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

figAddReservoirs selectNorm overplotDataset figOverplot figOverplotDiagram

Examples

data(blatna)
accessVar("blatna")

# Simple binary plots
multiplePerPage(2,ncol=2,nrow=1,title="Testing plateAddReservoirs",dummy=FALSE)
screen(1)
binary("Ba","Sr",new=FALSE,log="xy")
screen(2)
binary("Ba","Rb/Sr",new=FALSE)
plateCex(2)
plateCexLab(1.5)
# Temporary overplotting with the selected reservoirs
# Sun & McDonough 1989 mantle reservoirs, Taylor & McLennan 1995 Upper/Lower Crust
reserv<-c("(MORB|EMORB|OIB) McDonough","Upper Crust Taylor 1995","Lower Crust Taylor 1995")
reserv.names<-c("NMORB","EMORB","OIB","UCC","LCC")

plateAddReservoirs(TRUE,"reservoirs.data",reserv.condition=reserv,
    labs=reserv.names,cex=1.2,col="darkblue",just.draw=TRUE)
plateRedraw()

# Permanent overplotting with a modelled trend
# Calculate Rayleigh-type fractionation trend and store in a global variable
ff<-seq(1,0.1,-0.1) # F, amount of melt left
x<-80*ff^(1.2-1)    # cL for three elements, arbitrary D of 1.2, 2.0 and 1.3
y<-550*ff^(2.0-1)
z<-1000*ff^(1.3-1)
my.trend<-cbind(x,y,z)
colnames(my.trend)<-c("Rb","Sr","Ba")
rownames(my.trend)<-ff

plateAddReservoirs(TRUE,var.name="my.trend",type="o",col="darkgreen",just.draw(FALSE)
plateRedraw()

# Spider plots
eee<-spider(WR,"NMORB immobile",0.1,1000,pch=1:14,col=1:14,legend=TRUE)
groupsByLabel("Suite")
figMulti(nrow=1,ncol=3,plot.symb=TRUE)

reserv<-c("OIB .* McDonough","Lower Crust Taylor 1995")
reserv.names<-c("OIB","LCC")
plateAddReservoirs(FALSE,"reservoirs.data",reserv.condition=reserv,
    labs=reserv.names,cex=1.2,col="darkgreen")

---

plateLabelSlots

**Annotate individual slots by letters or Roman numerals**

---

**Description**

Annotates individual slots in a plate by letters or Roman numerals. For instance (a), (b), (c)... or (i), (ii), (iii), (iv), (v)...

**Usage**

plateLabelSlots(text=letters,style="()",cex=1.5,pos="topright")

**Arguments**

text  desired type of labels; see Details.

style  optional character strings before and after label, typically brackets.

cex  relative size of the text compared to the current codepar("cex").

pos  character; position of the label relative to the plot.
Details

The argument 'what' may acquire one of following values:

'letters' 'LETTERS' 'numbers' 'roman' 'ROMAN'

or can be user-defined character string of longer or of the same length as is the number of slots to be annotated (see the last example).

Possible positions (parameter pos) are:

'bottomright' 'bottom' 'bottomleft' 'left'
'topleft' 'top' 'topright' 'right' 'center'

Value

none

Note

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots. See Plate, Plate editing and figaro for details.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

Plate, Plate editing, figaro

Examples

multipleMjr("SiO2")

plateLabelSlots("letters","",pos="bottomleft")

plateLabelSlots("ROMAN","{}")

my_labs<-c("1st","2nd","3rd","4th","5th","6th","7th","8th","9th")

plateLabelSlots(my_labs)
Description

Plots a plate of diagrams, based on the Figaro style sheets.

Usage

plotPlate(diagram, where="WR", ...)

Arguments

diagram a valid name of the function that uses the plate concept to plot the given diagram. See Details.
where name of the data matrix/data frame, columns of which are to be used for plotting.
... optional parameters for the diagram function call.

Details

The argument 'diagram' may acquire one of following values:

'Maniar' 'Frost' 'PearceGranite' 'Schandl'
'Verma' 'Agrawal' 'Cann' 'Wood'

Value

none

Note

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots. See Plate, Plate editing and figaro for details.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

Plate, Plate editing, figaro

Examples

plotPlate("PearceGranite")
plotWithCircles(xaxis = "", yaxis = "", zaxis = "", colour = "heat.colors", scaling.factor = NULL, bins = NULL, ident = getOption("gcd.ident"))

Arguments

xaxis Name of the data column to be used as x axis.
yaxis Name of the data column to be used as y axis.
zaxis Name of the data column to determine the size/colour of the circles.
colour colour scheme for the circles.
scaling.factor a factor determine the size of the circles.
bins number of intervals for the legend.
ident Logical: should be the individual samples identified?

Details

If no parameters 'xlab', 'ylab' and 'zlab' are given, the user is prompted to specify them.

The variables are selected using the function 'selectColumnLabel'.

In the specification of the apices can be used also arithmetic expressions, see calcCore for the correct syntax.
The samples to be plotted can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSubset` for details.

The legal colour schemes are: "grays", "reds", "blues", "greens", "cyans", "violets", "yellows" "cm.colors", "heat.colors", "terrain.colors", "topo.colors", "rainbow", "jet.colors".

**Value**

None.

**Warning**

This function IS NOT Figaro-compatible.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>
& Vojtech Erban, <erban@sopky.cz>

**Examples**

```r
plotWithCircles("SiO2","Na2O+K2O","MgO+FeOt",colour="rainbow")
plotWithCircles("SiO2","MgO","K2O",colour="grays",scaling.factor=0.5,ident=TRUE)
```
pokeDataset

Storing a dataset into memory for later use

Description
Saves the current dataset into memory so that it can be later re-stored.

Usage
```r
pokeDataset(which.dataset=NULL,
  par.list="WR,WRanh,milli,labels,filename,groups,grouping,init,age",
  overwrite.warn=TRUE)
```

Arguments
- `which.dataset` character; a name of the stored dataset.
- `par.list` list of global variables to be stored.
- `overwrite.warn` logical, warn if a dataset is going to be rewritten in 'WRCube'. See Details.

Details
This function stores the global variables specified by `par.list`, typically 'WR', 'WRanh', 'milli', 'labels', 'filename', 'groups' and 'grouping' into the list 'WRCube'.

If no `which.dataset` is provided upon the call, it can be typed in or selected from the list of existing datasets.

Please note that `pokeDataset` is also invoked when a new dataset is loaded into memory using the functions `loadData` or `accessVar`. In the former case it is stored under the name of the file, in the latter under the variable name. If such a name already exists in 'WRCube', a time stamp is attached.

For restoring the stored variables serve functions `peekDataset` and `selectDataset`. The function `purgeDatasets` removes all older datasets, apart from the most recent copy of the current one.

Value
None.

Warning
If not called from a GUI, no warning is issued upon rewriting the existing dataset.

Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also
- `peekDataset`
- `selectDataset`
- `purgeDatasets`
- `loadData`
- `accessVar`
Examples

data(sazava)
accessVar("sazava")
# stored as sazava in WRCube
assignColVar("MgO","blues")
assign1symb(15)
# store a new copy in the WRCube
pokeDataset("coloured sazava")

data(swiss)
accessVar("swiss")
# stored as swiss in WRCube

peekDataset("sazava")
binary("SiO2","Ba")

peekDataset("coloured sazava")
binary("SiO2","Ba")

peekDataset("swiss")
binary("Catholic","Education",pch=15,col="darkgreen")

---

**ppm2oxide**  
*Calculation of wt% of the given oxide from ppm of atom*

Description

Recasts concentrations of a cation (in ppm) to those of the selected oxide (in wt%).

Usage

`ppm2oxide(formula,where="WR")`

Arguments

- `formula`: character: the oxide which is to be recalculated
- `where`: character: a name of matrix or dataframe with Te data to be recalculated

Value

A numeric matrix with one column containing the recalculated concentrations of the given oxide (in wt %) for individual samples.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

`oxide2ppm, oxide2oxide, molecularWeight`
prComp

Statistics: Principal components

**Examples**

```r
data(sazava)
accessVar("sazava")
ppm2oxide("K2O")

oxide2ppm("FeOt")
oxide2ppm("FeO")+oxide2ppm("Fe2O3")
```

**Description**

Performs principal components analysis (scaled variables, covariance or correlation matrix) and plots a biplot (Gabriel, 1971).

**Usage**

```r
prComp(comp.data=NULL,use.cov=FALSE,scale=TRUE,GUI=FALSE)
```

**Arguments**

- **comp.data**: a numerical matrix; the data to be normalized. Or just names of variables in the data matrix 'WR'.
- **use.cov**: logical; should be the covariance matrix used instead of correlation matrix?
- **scale**: logical; the scalings applied to each variable.
- **GUI**: logical; is the function called from a menu (GUI)?

**Details**

Biplot aims to represent both the observations and variables of a data matrix on a single bivariate plot (Gabriel, 1971; Buccianti & Peccerillo, 1999).

In the biplots, the length of the individual arrows is proportional to the relative variation of each variable. A comparable direction of two arrows implies that both variables are positively correlated; the opposite one indicates a strong negative correlation. When two links are perpendicular it indicates independence of the two variables (Buccianti & Peccerillo, 1999).

If called from menu (GUI version), a list of major elements (SiO2, TiO2, Al2O3, FeOt, MnO, MgO, CaO, Na2O, K2O) is assumed as a default, but different variables can be specified by the function 'selectColumnsLabels'.

The samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see selectSamples for details.

**Value**

Vector of the scores of the supplied data on the principal components is stored in a variable 'results'. Returns invisibly the complete output from the underlying function 'princomp'.

**Warning**

Names of existing numeric data columns and not formulae involving these can be handled at this stage. Only complete cases are used for the principal components analysis.
Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

For further details on the used principal components algorithm and biplots, see the R manual entries of `princomp` and `biplot.princomp`.

printSamples  Display samples

Description

Displays specified combination of numeric variable(s) and/or labels for selected range of samples.

Usage

printSamples(elems=NULL, which=NULL, select.samples=FALSE, print=TRUE)

Arguments

elems    list of variables to be printed
which     list of samples, useful only for select.samples=FALSE
select.samples logical: if TRUE, samples can be chosen using the appropriate dialogue
print     logical: should the result indeed printed or just returned for further evaluation?

Details

This function prints the desired numerical columns, textual labels, or their combinations, for selected samples.

The samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSamples` for details.

The variables to be printed are chosen by the function `selectColumnsLabels`. In the specification of the variable can be used also arithmetic expressions, see `calcCore` for the correct syntax.

Value

results    data matrix with the desired data for the specified samples
printSingle

Display a variable

Description

Displays a single numeric variable or a result of a calculation.

Usage

printSingle(default="")

Arguments

default character: list of default column names, separated by commas.
profiler

**Details**

The variable to be printed is selected using the function `selectColumnLabel`. In the specification of the variable can be used also arithmetic expressions, see `calcCore` for the correct syntax.

In the specification of the variable can be used also arithmetic expressions, see `calcCore` for the correct syntax.

**Value**

| results | numerical vector/matrix with the results |

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**Examples**

```r
## Not run:
# examples of valid formulae....
(Na2O+K2O)/CaO
Rb^2
log10(Sr)
mean(SiO2)/10

# ... but this command is in fact a simple R shell -
# meaning lots of fun for power users!
summary(Rb,na.rm=TRUE)
cbind(SiO2/2,TiO2,Na2O+K2O)
cbind(major)
hist(SiO2,col="red")
boxplot(Rb~factor(groups))

# possibilities are endless
plot(Rb,Sr,col="blue",pch="*",xlab="Rb (ppm)",ylab="Sr (ppm)",log="xy")

## End(Not run)
```

---

**profiler**

*Profile plotting*

**Description**

Plotting geochemical profiles. As a x axis can be specified an arbitrary variable or a numerical interval (for equidistant measurements).

**Usage**

```r
profiler(x = NULL, y = NULL, method = "Variable", legend = FALSE, pch = 1, col = "black", cex = 1, xaxs = "r", yaxs = "i", main = "",xmin = NULL, xmax = NULL)
```
Arguments

x character; optional name of variable to be plotted as x axis.
y character; name(s) of variable(s) for individual profiles.
method character; which of the methods is to be used? Valid are "Variable","Equidistant" or "From-To".
legend logical; should be plotted also legend (in a separate window)?
pch plotting symbols specification.
col plotting colour(s).
cex numeric; relative size of the plotting symbols.
xaxs, yaxs character; type of the axes. See par for details.
main character; main title for the plot
xmin, xmax range of the x axis (for methods 'Variable' and 'From/To'))

Details

The function 'profiler' serves for plotting three different types of profiles involving a single or several geochemical parameters.

![My plot](image)

The first one, 'Variable' uses any numeric variable as the x axis (e.g., SiO2 contents, depth...). It is in fact a special type of a binary plot, in which the data points are, for each of the y-axis variables, joined by a line.

The remaining two methods are very similar to each other. The x axis is in both cases equidistant, and the order of the individual samples follows from their sequence in the data set.

The method 'Equidistant' uses simply the sequence number of the individual samples in the data set. It does not label the x-axis, just prints the number of samples used for plotting.
The method 'From/To' serves for drawing equidistant profiles, where the x axis can be specified by an interval.

In the specification of the x axis (for the method 'Variable') or any of the y variables (all methods) can be used also arithmetic expressions, see calcCore for the correct syntax.

If not called from the command prompt, the samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see selectSubset for details.

The easiest way to specify the variable(s) to be plotted on individual profile(s) is to type directly the names of the columns, separated by commas. Alternatively can be used their sequence numbers or ranges. Also built-in lists can be employed, such as 'LILE', 'REE', 'major' and 'HFSE' or their combinations with the column names.

These lists are simple character vectors, and additional ones can be built by the user (see Examples). Note that currently only a single, stand-alone, user-defined list can be employed as a search criterion.

If the function is not called from the command prompt, and it desired so, the symbols and colours for each of the profiles can be specified separately in a simple spreadsheet-like interface.

If x axis occurs among the arguments to be plotted as y axes, it is skipped.

Likewise the relative scaling of the plotting symbols and the scale of the y axis can be specified.

Lastly, the user is asked to enter the limits for the axes, which are always two numbers separated by a comma.

**Value**

results numeric matrix with the values for individual profiles.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**Examples**

# Profiles of SiO2 versus (scaled) TiO2, MgO and K2O
# if x is specified, method="Variable" assumed automatically
profiler("Na2O+K2O",c("TiO2","6*MgO","SiO2"),pch=c("+","o","@"),col=c("red","blue","darkgreen"),
          xmin=2,xmax=10)

# Equidistant profiles of (scaled) MgO, CaO, and Al2O3 (in sample sequence)
# with default symbols and scaling
profiler(y=c("MgO","3*CaO","2*Al2O3"),method="Equidistant",col=c("red","blue","darkgreen"))

# Equidistant profiles of two calculated variables in custom colour
# and user-defined plotting symbols; range of the x axis will be specified
# interactively
profiler(y=c("2*MgO","10*(Na2O+K2O)"),method="From-To",pch=1:10,
       col=c("blue","red"),cex=1.5,main="My plot",xmin=10,xmax=30)

Value

results numeric matrix with the values for individual profiles.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

Examples

# Profiles of SiO2 versus (scaled) TiO2, MgO and K2O
# if x is specified, method="Variable" assumed automatically
profiler("Na2O+K2O",c("TiO2","6*MgO","SiO2"),pch=c("+","o","@"),col=c("red","blue","darkgreen"),
          xmin=2,xmax=10)

# Equidistant profiles of (scaled) MgO, CaO, and Al2O3 (in sample sequence)
# with default symbols and scaling
profiler(y=c("MgO","3*CaO","2*Al2O3"),method="Equidistant",col=c("red","blue","darkgreen"))

# Equidistant profiles of two calculated variables in custom colour
# and user-defined plotting symbols; range of the x axis will be specified
# interactively
profiler(y=c("2*MgO","10*(Na2O+K2O)"),method="From-To",pch=1:10,
       col=c("blue","red"),cex=1.5,main="My plot",xmin=10,xmax=30)
projbiplot

Calculates ternary coordinates projected from biotite (and plots the ternary diagram).

Description

The function projbiocoords calculates the coordinates used to define the projection, whereas projbioplot calls projbiocoords and does the other operations needed to define a template that will be used by plotDiagram.

Usage

projbiocoords(where=WR, add=FALSE)
projbioplot(mins=FALSE, addWR=FALSE, ticks=FALSE, xmin=-2, xmax=1, ymin=-0.5, ymax=0.5)

Arguments

projbiocoords:
A data matrix containing whole rock analyses, to be projected. By default WR

where Boolean. If TRUE, the results of the calculations will be added to WR as new columns with colnames = c("ms1","fsp","CaAl","bio")

projbioplot:

mins Boolean. If TRUE, the composition of ideal minerals will be plotted on the diagram, namely "q","fsp","an","an50","cz","Ep","sill","opx","cpx","olv","grs-Gt","Gt","NaCrd","Crd","bio","ms","MgHbl","Edn" and "Pgs".

addWR Boolean. If TRUE, the results of the calculations will be added to WR as new columns with colnames = c("ms1","fsp","CaAl","bio")

ticks Boolean. If TRUE, tick marks will be added to the side of the diagram (similar to the ticks option in ternary)

xmin, xmax, ymin, ymax
Numeric. Bounds of the plotting area, in rectangular coordinates. Note that the vertical axis (left side) is at x=0; ms1 = 3 al + 2 (Na + K) is at x=0 and y=0.5, and CaAl = Ca + Al is at x=0 and y=-0.5; the fsp (=Al + (Na + K)) is at x = √3/2 and y=0. Plagioclase an50 is at x = −√3/2 and y=0, and most points should fall to the right of it. The defaults are therefore sensible.

Details

The "projection from biotite" of Moyen et al. (2017) is based on four coordinates based on molar proportions:

\[ ms1 = Al - Ca - NK \]
\[ fsp = -2Al - 2Ca + 3NK - 1/3FM \]
\[ CaAl = Ca \]
\[ bio = 1/3FM \]

where Al = molar Al

Ca = molar Ca
FM = molar Fe + molar Mg
NK = molar Na + molar K
molar proportions are calculated by \texttt{millications}.
They are primarily used for plotting the relevant diagram (ms1-fsp-CaAl, note that bio is dropped during plotting as this is projected from bio).
\texttt{projbiocoords} calculates the coordinates. It also has the side effect of (globally) affecting its results to \texttt{results}, where they are available to \texttt{addResults}, \texttt{r2clipboard(results)}, \texttt{HTMLTableResults}, etc.

\texttt{projbioplot} is used purely for defining a Figaro template. The values of the four coordinates are calculated (and (globally) assigned to \texttt{results}). If called with \texttt{plotDiagram}, the diagram is plotted, which is the real use of the function.
If used as a plugin, a GUI function \texttt{.projbioGUI} is also supplied (and linked to the menu item), in charge of gathering the missing arguments and calling \texttt{plotDiagram}.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{Projected_from_biotite_onto_A-C-NK_plane.png}
\caption{Projected from biotite, onto A-C-NK plane}
\end{figure}

\textbf{Value}
For \texttt{projbioplot}, nothing. This function is meant only to be called via \texttt{plotDiagram}.
\texttt{projbiocoords} returns a matrix of 4 columns containing the new coordinates, (ms1,fsp,CaAl,bio).

\textbf{Author(s)}
Jean-François Moyen <jfmoyen@gmail.com>

\textbf{References}
and in particular the supplementary item SE4 "Multivariate statistics and projection for granitic rocks".

\textbf{See Also}
\texttt{plotDiagram} \texttt{millications}
Examples

data(sazava)
accessVar("sazava")
projbiocoords()
plotDiagram("projbioplot")

plotDiagram("projbioplot",mins=T,xmin=-1)

psAll

Description

Saves all graphical windows to Postscript files.

Usage

psAll(filename=NULL)

Arguments

filename a name of file for saving the output.

Details

The function prompts for a common root of the filenames and then saves all graphical windows, each in a separate file, numbering them sequentially. Postscript is the best export format from R, preserving the necessary quality as well as the possibility to be imported by most graphical editors (such as Corel Draw!) for retouching.

Otherwise individual diagram can be saved from a menu that appears after clicking on the appropriate graphical window ("File|Save as|Postscript").

Value

None.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

'pdfAll' 'postscript'
purgeDatasets

Removing stored datasets from the memory

Description
Removes all the stored datasets (apart from the current one) in order to save memory.

Usage
purgeDatasets(GUI=FALSE)

Arguments
GUI logical; is the function called from GUI?

Details
This function removes all older datasets, regardless whether stored automatically by the functions 'loadData' or 'accessVar', as well as on demand by 'pokeDataset'.
Only the most recent copy of the current dataset is preserved (i.e. the last item within the list 'WRCube').

Value
None.

Warning
If not called from a GUI, no warning is issued and all but the current dataset are deleted immediately.

Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also
'pokeDataset' 'peekDataset' 'selectDataset'

QANOR

Q'-ANOR diagram (Streckeisen + Le Maitre 1979)

Description
Plots normative Q'-ANOR diagram of Streckeisen & Le Maitre (1979).

Usage
QANOR(mesonorm=NULL,new=TRUE)
Arguments

mesonorm  numeric, a matrix containing normative minerals calculated, most appropriately, by the function Mesonorm.

new    logical, is a new plotting window to be opened?

Details

This function plots Q'-ANOR diagram of Streckeisen & Le Maitre (1979) where \[ Q' = \frac{100Qz}{Qz + Or + Ab + An} \] and \[ ANOR = \frac{100An}{Or + An} \] based on eine bessere Mesonorm for granitoids of Mielke & Winkler (1979).

The fields in this diagram are labeled as follows:

2  alkali feldspar granite
3  granite
4  granodiorite
5  tonalite
6* quartz alkali feldspar syenite
7* quartz syenite
QAPF

8* quartz monzonite
9* quartz monzodiorite/quartz monzogabbro
10* quartz diorite/quartz gabbro
6 alkali feldspar syenite
7 syenite
8 monzonite
9 monzodiorite/monzogabbro
10 diorite/gabbro

Value

A numeric matrix 'results'.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>
& Jean-François Moyen <jfmoyen@gmail.com>

References


See Also

classify figaro plotDiagram Mesonorm

Examples

data(blatna)
accessVar("blatna")
plotDiagram("QANOR",FALSE)

<table>
<thead>
<tr>
<th>QAPF</th>
<th>QAPF diagram (Streckeisen 1974, 1978)</th>
</tr>
</thead>
</table>

Description

Assigns data for Streckeisen’s diagram into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'. The Q, A, P and F coordinates are assigned into matrix 'results'.

Usage

QAPFVolc()
QAPFPPlut()
Following the IUGS recommendation (Le Maitre et al. 2002), the QAPF diagram should be the prime classification scheme for holocrystalline plutonic and volcanic rocks containing at least 10% of felsic minerals.

The apices are defined as follows:

Q = Quartz modal %

A = Alkali feldspar modal %

P = Plagioclase modal %

F = Feldspathoid modal %

Q + A + P + F = 100 %

As the whole QAPF diagram is rather complicated, GCDkit plots just the appropriate triangle if the dataset contains only Si-oversaturated or only Si-undersaturated rock samples. If both kinds of rock samples are present, the whole double triangle is shown. This behaviour may be changed in the source code of the diagram (in file 'QAPFPlut.r' or 'QAPFVolc.r', stored in the subdirectory GCDkit\Diagrams\Classification, change the 'triangle<="auto"' to 'triangle<="both"' and complete double triangle will be always plotted).
Value

- sheet: list with Figaro Style Sheet data
- x.data, y.data: Q, A, P and F data (see details) transformed to orthogonal coordinates

Author(s)

Vojtech Erban, <erban@sopky.cz>

References

doi: 10.1007/BF01820841


See Also

classify figaro plotDiagram

Examples

#plots the QAPF diagram for current dataset
plotDiagram("QAPFVolc", FALSE)
plotDiagram("QAPFPlut", FALSE)

# classifies the current dataset using the QAPF diagram
classify("QAPFVolc")
classify("QAPFPlut")

### quitGCDkit

**Exit GCDkit**

**Description**

Exits GCDkit (nicely).

**Usage**

```r
quitGCDkit()
```

**Arguments**

None.

**Details**

By invoking this command the user is not prompted whether he wants to save his unfinished work in the 'Workspace image', i.e. file `.RData` in the main GCDkit directory.

**Menu**

GCDkit: Exit GCDkit

**See Also**

'quit'

---

### r2clipboard

**Copy results to clipboard**

**Description**

Copies the most recently calculated results to a clipboard.

**Usage**

```r
r2clipboard(what=results)
```
Arguments

- **what**: A variable to be copied, can be either a vector, a matrix, a list or a table.

Details

Copies the variable `results` returned by most of the calculation algorithms to the Windows clipboard.

Value

None.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

---

**recast**

**Recast to given sum**

Description

Recasts the selected data to a fixed sum.

Usage

```r
recast(total = 100)
normalize2total(what = NULL, total = 100)
```

Arguments

- **what**: Numeric matrix or character vector with a list of column names to be normalized, separated by commas.
- **total**: A sum the data should be normalized to.

Details

Both functions return the selected elements/oxides (columns in the data matrix 'WR') normalized to the required sum. The function 'recast' is front-end to 'normalize2total'. If 'what' is a comma delimited list, the corresponding columns from the data matrix 'WR' are selected. If 'what' is empty, the user is prompted to supply the list of required column names via the function 'selectColumnLabels'.

Value

- **results**: Numerical vector/matrix with the results

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>
reciprocalIso

Binary plots of reciprocal element concentration vs initial isotopic composition

Description

Plots a diagram 1/Sr vs initial Sr isotopic ratios or 1/Nd vs initial $\epsilon(Nd)$ for selected samples.

Usage

reciprocalIso(what=NULL, GUI=FALSE, ...)

Arguments

what
name of the desired isotopic parameter

GUI
logical; is the function called from the GUI?

... optional parameters to the underlying function (plotWithLimits)
Details

The recognized types of diagrams (specified by 'what') are: 'Rb-Sr' and 'Sm-Nd' for the 1/Sr vs. $^{87}\text{Sr} / ^{86}\text{Sr}$ or 1/Nd vs. $\epsilon$(Nd) plots, respectively.

If called from GUI, the samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see selectSamples for details.

Value

None.

Plugin

SrNd.r

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

The actual plotting is done by the function plotWithLimits.

Regular expressions

Implementation of regular expressions in GCDkit

Description

Implementation of regular expressions in the searching patterns.

Details

Many enquiries in the GCDkit employ regular expressions. This is a quite powerful searching mechanism more familiar to people working in Unix. Put in simple terms, most characters, including all letters and digits, are regular expressions that match themselves. However, metacharacters with a special meaning ('?' '+' '{' '}' '|' '(' ')') must be preceded by a backslash.

<table>
<thead>
<tr>
<th>Regular expression</th>
<th>Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>.</td>
<td>Any character</td>
</tr>
<tr>
<td>^</td>
<td>Beginning of the expression</td>
</tr>
<tr>
<td>$</td>
<td>End of the expression</td>
</tr>
<tr>
<td>[ ]</td>
<td>Any of the characters given in square brackets</td>
</tr>
<tr>
<td>[m-n]</td>
<td>Any character in the range given by m and n</td>
</tr>
</tbody>
</table>

A subexpression is a regular expression enclosed in '()' and '\('. Two such subexpressions may be joined by the infix operator '|' (logical or); the resulting regular expression matches any string matching either of them. For instance:

\{South\} \{North\}Uist

yields both

South Uist and North Uist.
A regular expression may be followed by one of several repetition operators:

<table>
<thead>
<tr>
<th>Repetition operator</th>
<th>The preceding item will be matched</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>At most once (i.e. is optional)</td>
</tr>
<tr>
<td>*</td>
<td>Zero or more times</td>
</tr>
<tr>
<td>+</td>
<td>One or more times</td>
</tr>
<tr>
<td>{n}</td>
<td>Exactly n times</td>
</tr>
<tr>
<td>{n,m}</td>
<td>At least n times, but not more than m times</td>
</tr>
</tbody>
</table>

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

regex

Examples

```r
## Not run:
# Subset by label
The searched field corresponds to localities with the following levels:
Mull, Rum, Skye, Coll, Colonsay, Hoy, Westray, Sanday,
Stronsay, Tiree, Islay

Search pattern = ol
Coll, Colonsay

Search pattern = n.a
Colonsay, Sanday, Stronsay

Search pattern = ^S
Skye, Sanday, Stronsay

Search pattern = e$
Skye, Tiree

Search pattern = [ds]ay
Colonsay, Sanday, Stronsay

Search pattern = [p-s]ay
Colonsay, Westray, Stronsay

Search pattern = ol|oy
Coll, Colonsay, Hoy

Search pattern = l{2}
Mull, Coll
```

# Subset by sample name
The sample names are: Bl-1, Bl-3, Koz-1, Koz-2, Koz-5, Koz-11, KozD-1, Ri-1.
Ross

Search pattern = oz-[1-3]
Koz-1, Koz-2, Koz-11

Search pattern = oz-|Bl-
Bl-1, Bl-2, Bl-3, Koz-1, Koz-2, Koz-5, Koz-11

## End(Not run)

Ross + Bedard (2009) Zr/Y-Th/Yb

Description

Assigns data for a Zr/Y vs. Th/Yb binary diagram into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

Usage

Ross()

Arguments

None.

Details

The binary plot Zr/Y vs. Th/Yb designed by Ross and Bédard (2009) for classification of ancient subalkaline volcanic rocks into tholeiitic or calc-alkaline series. In these cases, the conventional AFM diagram tends to be of limited use due to the potential mobility of alkalis.
Value

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sheet</code></td>
<td>list with Figaro Style Sheet data</td>
</tr>
<tr>
<td><code>x.data</code></td>
<td>x coordinates</td>
</tr>
<tr>
<td><code>y.data</code></td>
<td>y coordinates</td>
</tr>
</tbody>
</table>

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


Coordinates and graph layout are taken from website of Kurt Hollocher.

See Also

figaro plotDiagram AFM

Examples

plotDiagram("Ross",FALSE,TRUE)
**Description**

Calculates rutile saturation temperatures for the observed major-element data and Ti concentrations. Also returns Ti saturation levels for the given major-element compositions and assumed magma temperature.

**Usage**

```r
tSaturation(cats=milli,T=0,P=0,Ti=filterOut(WR,"Ti",1))
```

**Arguments**

- **cats** numeric matrix; whole-rock data recast to millications
- **T** assumed temperature of the magma in °C
- **P** assumed pressure in kbar, Ryerson & Watson (1987) model only
- **Ti** numeric vector with Ti concentrations in ppm

**Details**

Ryerson & Watson (1987) have first formulated rutile saturation model for melts ranging in composition from basalt to rhyodacite. The distribution of $\text{TiO}_2$ between rutile and liquid was given as:

$$D_{\text{TiO}_2} = e^{(-3.16 + \frac{2.74}{T} + 0.026P - 0.152FM)}$$

where 'T' is the absolute temperature (K) of the magma, 'P' pressure (kbar) and 'FM' is a melt composition parameter:

$$FM = \frac{1}{Si} Na + K + 2(Ca + Mg + Fe) Al$$

The Ti saturation level then would be:

$$Ti.sat.RW = \frac{599342.9}{D_{\text{TiO}_2}} (ppm)$$

In turn, when the rutile saturation was reached, the magma temperature (in °C) can be calculated as:

$$TRt.sat.C.RW = \frac{9373}{(3.16 + ln(100/TiO2) - 0.026P + 0.152FM) - 273.15}$$

The Ti solubility in rutile-saturated hydrous siliceous melts was revisited by Hayden & Watson (2007). According to these authors, it can be expressed as:

$$Ti.sat.HW = 10^{(7.95 - \frac{539}{P} + 0.124FM)} (ppm)$$
where ‘T’ is the absolute temperature (K) of the magma, and ‘FM’ is the melt composition parameter defined above.

The temperature (in °C) for rutile-saturated magma can be calculated as:

\[ T_{Rt.sat.C.HW} = \frac{5305}{7.95 - \log(Ti)} + 0.124FM - 273.15 \]

Using these formulae, the function ‘rtSaturation’ calculates the rutile saturation levels, Ti activities and rutile saturation temperatures following both models.

The formulation of Ryerson & Watson (1987) may be more suitable for basic rocks, whereas the more recent model of Hayden & Watson (2007) seems to be appropriate for siliceous magmas. Please note also that the latter does not take into account effects of pressure (having been calibrated at 1 GPa; Hayden & Watson 2007).

**Value**

Returns a matrix ‘results’ with the following columns:

- **FM**: melt composition parameter
- **Ti**: observed Ti concentrations
- **Ti.sat.RW**: saturation levels of Ti for assumed temperature, Ryerson & Watson (1987)
- **aTi.RW**: activity of Ti (ratio of Ti/Ti.sat), Ryerson & Watson (1987)
- **TRt.sat.C.RW**: rutile saturation temperatures in °C, Ryerson & Watson (1987)
- **Ti.sat.HW**: saturation levels of Ti for assumed temperature, Hayden & Watson (2007)
- **aTi.HW**: activity of Ti (ratio of Ti/Ti.sat), Hayden & Watson (2007)

**Plugin**

Saturation.r

**Author(s)**

Vojtěch Janoušek, <vojtech.janousek@geology.cz>

**References**


saveData

**Save data file**

**Description**
Saves modified data set into a specified datafile.

**Usage**
saveData(sep="\t")

**Arguments**
- **sep**: delimiter separating individual items in the data file.

**Details**
Labels (stored in data frame `labels`) and numeric data (in numeric matrix `WR`) for the currently selected subset are glued together and saved under the specified filename. The format is such that the data can be retrieved again into GCDkit using the `loadData` command. Note that no mg numbers are currently saved.

**Value**
None.

**Author(s)**
Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**
`loadData`, `mergeData`, `showColours`, `colours`, `showSymbols`, `read.table`

---

saveResults

**Save results**

**Description**
Saves the most recently calculated results to a text file.

**Usage**
saveResults(what = results, sep = "\t", digits = 2)

**Arguments**
- **what**: a variable to be saved, can be either a vector, a matrix or a list.
- **sep**: separator; default is a tab-delimited file.
- **digits**: precision of the results to be saved.
**Details**

Saves the variable ‘results’ returned by most of the calculation algorithms to a tab-delimited ASCII file.

**Value**

None.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

---

**Description**

Saves the calculated isotopic parameters stored in the matrix ‘init’ to a text file.

**Usage**

```
saveResultsIso(digits = 6)
```

**Arguments**

- **digits**: precision of the results to be saved.

**Details**

Saves the data matrix `init` with the following columns:

- **Age (Ma)**: Age in Ma
- **87Sr/86Sri**: Initial Sr isotopic ratios
- **143Nd/144Nd**: Initial Nd isotopic ratios
- **EpsNd**: Initial ϵ(Nd) values
- **TDM**: Single-stage depleted-mantle Nd model ages *(Liew & Hofmann, 1988)*
- **TDM.Gold**: Single-stage depleted-mantle Nd model ages *(Goldstein et al., 1988)*
- **TDM.2stg**: Two-stage depleted-mantle Nd model ages *(Liew & Hofmann, 1988)*

**Value**

None.

**Plugin**

SrNd.r

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>
References


Goldstein S L, O’Nions R K & Hamilton P J (1984) A Sm-Nd isotopic study of atmospheric dusts and particulates from major river systems. Earth Planet Sci Lett 70: 221-236

See Also

'saveResults'

| sazava | Whole-rock composition of the Sazava suite, Central Bohemian Plutonic Complex |

Description

This data set gives the whole-rock major- and trace-element contents in selected samples (gabbros, quartz diorites, tonalites and trondhjemites) of the c. 355 My old calc-alkaline Sazava suite of the Variscan Central Bohemian Plutonic Complex (Bohemian Massif, Czech Republic).

Usage

data(sazava)

Format

A data frame containing 14 observations.

Source

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


Examples

data(sazava)
accessVar("sazava")
binary("SiO2","Ba")
Description

Plots data stored in 'WR' (or its subset) into the classification diagrams after Schandl and Gorton (2002).

Usage

Schandl(plot.txt = getOption("gcd.plot.text"))

Arguments

plot.txt logical, annotate fields by their names?

Details

Suite of four diagrams for geotectonic environment discrimination of felsic volcanic rocks (rhyolites), proposed by Schandl and Gorton (2002). It is based on combination of four presumably little immobile trace elements (namely Ta, Yb, Th, and Hf). Diagrams were designed to decipher the geotectonic setting of felsic volcanic suites, specifically those associated with the volcanogenic massive sulphide (VMS) deposits. a) Ta/Yb versus Th/Yb diagram from Gorton and Schandl (2000) is divided into three fields: Oceanic Arcs, Active Continental Margins (ACM) and Within-Plate Volcanic Zones (WPVZ). The Within-Plate Basalts (WPB) and Mid-Ocean Ridge Basalts (MORB) represent compositions previously determined by Pearce (1982, 1983). b) Ta vs. Th diagram demonstrates the Th enrichment of felsic volcanic rocks at post-Archaean VMS deposits (and of some unmineralized Archaean rhyolites) with respect to Ta. c) Graph of Ta/Hf vs Th/Hf ratios shows the similar incompatibility between Th and Ta in two different tectonic environments: Active Continental Margins and Within-Plate Volcanic Zones. d) Yb vs. Th/Ta diagram with fields for associations of Oceanic Arcs, Active Continental Margins, Within Plate Volcanic Zones and MORB.
Taken together, the following geotectonic settings may be deduced:

<table>
<thead>
<tr>
<th>Rock Association</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oceanic Arcs</td>
<td></td>
</tr>
<tr>
<td>Active Continental Margins</td>
<td>ACM</td>
</tr>
<tr>
<td>Within-Plate Volcanic Zones</td>
<td>WPVZ</td>
</tr>
</tbody>
</table>

Further abbreviations used on the plots:

<table>
<thead>
<tr>
<th>Rock Association</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mid-Oceanic Ridge Basalts</td>
<td>MORB</td>
</tr>
<tr>
<td>Within-Plate Basalts</td>
<td>WPB</td>
</tr>
</tbody>
</table>
Note

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots. See Plate, Plate editing and figaro for details.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

Plate, Plate editing, plotPlate, figaro

Examples

# plot the diagrams
plotPlate("Schandl")

selectAll

Select whole dataset

Description

Restores data for all samples as they were loaded from a data file.

Usage

selectAll(GUI=FALSE)

Arguments

GUI logical; was the function called from the GUI?
selectByDiagram

Details

When a datafile is loaded into GCDkit using the `loadData` function, the data and their backup copy are stored in the memory.

The subsets of the current dataset can be chosen using the functions `selectByLabel` and `selectSubset` (menus 'Select subset by sample name or label', 'Select subset by range' , 'Select subset by Boolean') and the current data will be replaced by their newly chosen subset.

The backup copy is kept intact ever since the `loadData` function has been invoked and can be uploaded any time in place of the current data set using the function 'selectAll'. Note that all changes made e.g. to plotting symbols, grouping, newly calculated variables etc. will be lost.

Value

None.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

selectByDiagram  Selecting subset by diagram

Description

This function enables selecting samples that plot into certain field(s) of the given classification diagram.

Usage

`selectByDiagram(diagram = select.list(claslist[, "menu"]))`

Arguments

diagram  one of the valid diagram names that appear in `.claslist()`

Details

The diagram can be chosen from a list (the default) or specified directly as an argument. Clicking onto a field toggles its inclusion/exclusion - the currently selected fields are cyan.

Value

None.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz> & Vojtech Erban, <erban@sopky.cz>

See Also

'`selectByLabel', 'selectSubset', 'selectAll' and 'classify'.
Examples

```r
.claslist() # names of existing diagrams
selectByDiagram("TAS")
```

---

**selectByLabel**

*Select subset by sample name or label*

**Description**

Selecting subsets of the data stored in memory by searching sample names or a single label.

**Usage**

`selectByLabel()`

**Details**

This function enables the user to query a single textual column, a label, chosen using the function `selectColumnLabel()`. The current data will be replaced by its newly chosen subset. These enquiries employ regular expressions.

**Value**

Overwrites the data frame `labels` and numeric matrix `WR` by subset that fulfills the search criteria.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**Examples**

```r
## Not run:
# Subset by label
The searched field corresponds to localities with the following levels:
Mull, Rum, Skye, Coll, Colonsay, Hoy, Westray,
Sanday, Stronsay, Tiree, Islay

Search pattern = ol
Coll, Colonsay

Search pattern = n.a
Colonsay, Sanday, Stronsay

Search pattern = ^S
Skye, Sanday, Stronsay

Search pattern = e$
Skye, Tiree

Search pattern = [ds]ay
Colonsay, Sanday, Stronsay
```
selectColumnLabel

Description

This is an auxiliary function invoked by many others to select a single variable.

Usage

```r
selectColumnLabel(where = colnames(labels),
message = "Select the variable\nor press ENTER to pick from a list",
default = ",

Arguments

where

- `where`: names of data columns to choose from
- `message`: prompt
- `default`: comma delimited list of default names
- `sample.names`: logical; should be the sample names listed
- `silent`: logical, echo on/off
- `print`: logical, echo on/off
- `empty.ok`: is empty selection ok?

## End

# Subset by sample name
The sample names are: Bl-1, Bl-3, Koz-1, Koz-2, Koz-5, Koz-11, KozD-1, Ri-1.

Search pattern = oz-[1-3]
Koz-1, Koz-2, Koz-11

Search pattern = oz-[Bl-]
Bl-1, Bl-2, Bl-3, Koz-1, Koz-2, Koz-5, Koz-11

## End(Not run)
The easiest way for specification of the variable is to type directly the name of the numerical column in the data matrix \texttt{WR} (e.g., \texttt{SiO2}) or its sequence number (2 for the second column). However, it is not necessary to enter the name in its entirety. Only a substring that appears somewhere in the column name or other forms of \texttt{regular.expressions} can be specified.

If the result is ambiguous, the correct variable has to be selected by mouse from the list of the multiple matches. Ultimately, empty response invokes list of all variables available in the memory.

**Value**

A numeric index of the selected column.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

\texttt{selectColumnsLabels}

---

**Description**

An auxiliary function invoked by many others to select several variables simultaneously.

**Usage**

\begin{verbatim}
selectColumnsLabels(where = colnames(WR),
message = "Select variable(s), e.g. \texttt{SiO2,TiO2,MgO}
or press ENTER to pick from a list", default = ",", print = TRUE,
exact.only = TRUE)
\end{verbatim}

**Arguments**

\begin{verbatim}
where vector of names for data columns to choose from
message prompt
default comma delimited list of default names
print logical, echo on/off
exact.only logical, should be the input checked for correctness?
\end{verbatim}
Details

The variable(s) can be specified in several ways. The easiest is to type directly the name(s) of the column(s), separated by commas. Alternatively can be used their sequence numbers or ranges. Also built-in lists can be employed, such as 'LILE', 'REE', 'major' and 'HFSE' or their combinations with the column names.

These lists are simple character vectors, and additional ones can be built by the user (see Examples). Note that currently only a single, stand-alone, user-defined list can be employed as a search criterion.

Empty response invokes list of all variables available. The correct variables have to be selected by mouse + SHIFT from this list.

If exact.only=TRUE, the individual items in the input line are checked against the list of existing column/variable names (i.e. components in the vector 'where').

Value

Vector with the selected column names.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

Examples

```r
## Not run:
# Querying names of numeric data columns
Search pattern = SiO2, MgO, CaO
Search pattern = major
SiO2, TiO2, Al2O3, FeO, MnO, MgO, CaO, Na2O, K2O, P2O5
Search pattern = LILE
Rb, Sr, Ba, K, Cs, Li
Search pattern = HFSE
Nb, Zr, Hf, Ti, La, Ce, Y, Ga, Sc, Th, U
Search pattern = REE
La, Ce, Pr, Nd, Sm, Eu, Gd, Tb, Dy, Ho, Er, Tm, Yb, Lu
Search pattern = Locality, SiO2, LILE, HFSE
Locality, SiO2, Rb, Sr, Ba, K, Cs, Li, Nb, Zr, Hf, Ti, La, Ce, Y, Ga, Sc, Th, U
Search pattern = 1:5, 7
Numeric data columns number 1, 2, ...5, 7

# User-defined list
my.elems<-c("Rb","Sr","Ba")
Search pattern = my.elems
Rb, Sr, Ba

## End(Not run)
```
selectNorm

Selecting the normalization data for spiderplots

Description
Displays available normalization schemes and lets the user to choose one interactively.

Usage

selectNorm(ref=NULL,elems = "Rb,Sr,Ba,Cr,Ni,La,Ce,Y,Zr",REE.only=FALSE,
multiple=FALSE)

Arguments

ref character: a specification of the normalization scheme.
elems character: a default list of elements.
REE.only logical: should be only listed normalization schemes for REE?
multiple logical: is a result with several normalizing schemes allowed?

Details

A search pattern can be specified directly (in batch mode) in order to query the available normalizing model names. The corresponding parameter ’ref’ can contain a substring appearing in the name of the normalizing scheme (or even a regular expression).

Alternatively, the parameter ’ref’ can refer to a name of a sample to be used for normalization, or even a regular expression if average of several of them is desired.

The function fails if no matches are found or the search in names of normalizing schemes is ambiguous (returns more than a single match), unless ’multiple = TRUE’.

The second possibility is to pick an option from the list of available normalizing schemes via GUI. The first option therein offers normalization by a single sample. Its name can be typed in or, after pressing the Enter key, picked from a list.

The second option is similar but it allows to normalize by average concentrations in a group of samples specified by one of the three searching mechanisms as above (see selectSubset).

Then the user is prompted to specify the list and order of elements/oxides that should appear on the plot. The easiest way is to type directly the names of the columns, separated by commas. Alternatively can be used their sequence numbers or ranges. Also built-in lists can be employed, such as ’LILE’, ’REE’, ’major’ and ’HFSE’ or their combinations with the column names. These lists are simple character vectors, and additional ones can be built by the user (see Examples). Note that currently only a single, stand-alone, user-defined list can be employed as a search criterion.

The samples to be plotted can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see selectSubset for details.

The composition of various standards available for normalization and subsequent plotting of spider diagrams is stored in the file ’spider.data’ in the main GCDkit directory. It is a comma delimited file such as:
Normalization data used for spiderplots

MORB (Pearce 1983)
Sr, K, Rb, Ba, Th, Ta, Nb, Ce, P, Zr, Hf, Sm, Ti, Y, Yb
120, 1245, 2, 20, .2, .18, 3.5, 10, 534, .90, 2.4, 3.3, 8992, 30, 3.4

REE chondrite (Boynton 1984)
La, Ce, Pr, Nd, Pm, Sm, Eu, Gd, Tb, Dy, Ho, Er, Tm, Yb, Lu
.31, .808, .122, .6, 1.195, .0735, .2590, .0474, .322, .0718, .21, 0.0324, .209, .0322

ORG (PearceEtAl.1984)
K2O, Rb, Ba, Th, Ta, Nb, Ce, Hf, Zr, Sm, Y, Yb
0.4, 4, 50, 0.8, 0.7, 10, 35, 9, 340, 9, 70, 8, 0

The first row is always skipped and can contain any comments. The following ones have a fixed
structure. For each normalization scheme, the first row contains the title and reference. If title
starts with 'REE', the normalization is supposed to be for REE only and special parameters, such as
'Eu/Eu*'. are calculated. The second line gives a comma delimited list of elements in the order they
should appear on the plot. The last line is a comma delimited list of normalization values. There
are empty lines left between the normalization schemes.

As the file 'spider.data' is read every time 'selectNorm' is called, the user can add or delete
normalization schemes on his will using a text editor.

Value
A numeric matrix with one row, containing the normalizing values. The row name contains
the name of the model and reference.

Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>

References
Implemented spiderplots:
selectPalette

**Description**

Picks given number of colours from one of the available palettes.

**Usage**

```r
selectPalette(n, colour.palette=NULL, GUI=TRUE)
```
selectPalette

Arguments

- **n**: desired number of colours
- **colour.palette**: one of the colour palette names, see Details
- **GUI**: logical; is the function called from GUI?

Details

The desired number of colours has to be given in any case.

The possible palettes are: 'grays', 'reds', 'blues', 'greens', 'cyans', 'violets', 'yellows', 'cm.colors', 'heat.colors', 'terrain.colors', 'topo.colors', 'rainbow' and 'jet.colors'. Also, user-defined palette functions are supported now. See Examples.

If not specified upon function call, the colour palette can be picked from list of available ones. Optionally (if GUI = TRUE) it plots a chart with their preview.

```
<table>
<thead>
<tr>
<th>colours</th>
</tr>
</thead>
<tbody>
<tr>
<td>grays</td>
</tr>
<tr>
<td>reds</td>
</tr>
<tr>
<td>blues</td>
</tr>
<tr>
<td>greens</td>
</tr>
<tr>
<td>cyans</td>
</tr>
<tr>
<td>violets</td>
</tr>
<tr>
<td>yellows</td>
</tr>
<tr>
<td>cm.colors</td>
</tr>
<tr>
<td>heat.colors</td>
</tr>
<tr>
<td>terrain.colors</td>
</tr>
<tr>
<td>topo.colors</td>
</tr>
<tr>
<td>rainbow</td>
</tr>
<tr>
<td>jet.colors</td>
</tr>
</tbody>
</table>
```

Value

Returns a matrix with a single row of hexadecimal codes. Its rownames represent the name of the palette selected.

Note

Note that UK spelling of "colours" in names of palettes is fixed automatically to the US "colors".
selectSubset

Description

Selects samples corresponding to given criteria.

Usage

```r
selectSubset(what=NULL, where=cbind(labels,WR), save=TRUE, multiple=TRUE,
            text="Press ENTER for all samples, or specify search pattern 
            by sample name, range or Boolean condition",
            range=FALSE, GUI=FALSE, all.nomatch=TRUE)
```

```r
selectSamples(what=NULL, print=TRUE, multiple=TRUE, text=NULL)
```

Arguments

- `what` search pattern
- `where` data to be searched
- `save` should the newly selected subset replace the data in memory, i.e. 'labels' and 'WR'
- `multiple` logical, can be multiple items selected?
- `text` text prompt
- `range` logical: is the search pattern to be interpreted as a range of samples?
- `GUI` logical: is the function called from within GUI?
- `all.nomatch` logical: return all samples when there is no match?
- `print` logical: should the chosen samples ID printed?
Details

The function 'selectSubset' has two purposes.

1. If 'save=TRUE', it is a core function used in selecting subsets of the current data set by ranges (see `subsetRange`) or Boolean conditions (see `subsetBoolean`).

2. If `save=FALSE`, no permanent subsetting takes place. This is useful for temporary selections of the data, e.g. in determining which samples are to be plotted on a diagram.

In this case, the samples can be selected based on combination of three searching mechanisms. The search pattern is first tested whether it obeys a syntax of a valid regular expression that could be interpreted as a query directed to the sample name(s).

If not, the syntax of the search pattern is assumed to correspond to a selection of sample sequence numbers.

At the last resort, the search pattern is interpreted as a Boolean condition that may employ most of the comparison operators common in R, i.e. < (lower than), > (greater than), <= (lower or equal to), >= (greater or equal to), = or == (equal to), != (not equal to). The character strings should be quoted. Regular expressions can be employed to search the textual labels.

The conditions can be combined together by logical and, or and brackets. Logical and can be expressed as .and., .AND. &

Logical or can be expressed as .or., .OR. |

The function 'selectSamples' is a front-end to 'selectSubset'.

Value

If 'save=TRUE', the function overwrites the data frame 'labels' and numeric matrix 'WR' by subset that fulfills the search criteria. Otherwise names of samples fulfilling the given criteria are returned.

Warning

So far only names of existing numeric data columns and not formulae involving these can be handled.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

regex, selectByLabel and selectAll

Examples

# permanent selection, the variables 'WR' and 'labels' affected
selectSubset("SiO2>70")

# back to the complete, originally loaded dataset
selectAll()

# both expressions below return only sample names of analyses fulfilling
# the given criteria, variables 'WR' and 'labels' NOT affected
selectSamples("SiO2<70&MgO>5")
selectSubset("SiO2<70&MgO>5", save=FALSE)
## Not run:

#EXAMPLES OF SEARCHING PATTERNS

# Searching by sample name

The sample names are: Bl-1, Bl-3, Koz-1, Koz-2, Koz-5, Koz-11, KozD-1, Ri-1.

oz-[!-3]

# Samples Koz-1, Koz-2, Koz-11

oz-|Bl-

# Samples Bl-1, Bl-2, Bl-3, Koz-1, Koz-2, Koz-5, Koz-11

# Searching by range

1:5

# First to fifth samples in the data set

1,10

# First and tenth samples

1:5, 10:11, 25

# Samples number 1, 2, ...5, 10, 11, 25

# Searching by Boolean

-------------------

Intrusion="Rum"

# Finds all analyses from Rum

Intrusion="Rum".and.SiO2>65

Intrusion="Rum".AND.SiO2>65

Intrusion="Rum"&SiO2>65

# All analyses from Rum with silica greater than 65

# (all three expressions are equivalent)

MgO>10&(Locality="Skye"|Locality="Islay")

# All analyses from Skye or Islay with MgO greater than 10

Locality="^S"

# All analyses from any locality whose name starts with capital S

## End(Not run)

---

**setCex**  
*Set uniform symbols size*

**Description**

Defines the default relative size of plotting symbols.

**Usage**

`setCex(x)`
**setShutUp**

**Arguments**

- `x` numeric; scaling for the plotting symbols.

**Details**

The coefficient determining the plotting symbols expansion is stored in a variable `labels[, "Size"]`, the default is 1.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

`gcdOptions`

**Examples**

```r
setCex(2) # double size
plotDiagram("TAS", FALSE)

setCex(0.5) # half the size
plotDiagram("TAS", FALSE)
```

---

**setShutUp**

**Quiet mode?**

**Description**

Determines whether extensive textual output is to be printed.

**Usage**

```r
setShutUp()
```

**Arguments**

None.

**Details**

The control option is `shut.up`, whose default is `FALSE`, meaning that detailed information is to be printed. This, however, may become not viable on slower systems and/or for extensive data sets. This can be set from the menu 'GCDkit|Options' by setting the checkbox 'Minimize output on screen?' or directly, from the command line (see Examples).

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

`gcdOptions` `options`
Examples

getOption("shut.up") # query the current value of the given option
options("shut.up"=TRUE) # reduce the printed output to a minimum

setTransparency

Setting transparency of plotting colours

Description

Sets transparency of plotting colours for selected samples. Alternatively, it just returns the hexadecimal code(s) of specified colour(s) with the desired degree of transparency.

Usage

setTransparency(which.samples=NULL,transp=NULL,alpha=NULL,
col.in="black",save=TRUE,GUI=FALSE)

Arguments

which.samples list of samples; if NULL a dialogue is displayed
transp numeric; transparency to be set
alpha character; alpha value to be set (opacity)
col.in numeric or character vector; colour specification(s)
save logical; should be the result saved into labels$Colour?
GUI logical; is the function called form within GUI?

Details

The transparency value has to fall between 1 (completely transparent) to 0 (opaque).
Alternatively, the so-called alpha channel can be specified, which can attain any hexadecimal number between 0 (completely transparent) to ff (opaque).
if GUI = TRUE, the samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see selectSamples for details.

Value

Returns (invisibly) hexadecimal codes of the colours with desired degree of transparency If 'save=TRUE' it also assigns 'labels$Colour' producing the new, partly transparent colour.

Warning

As a side product, plotting colours are converted to hexadecimal values, which are not easy to translate back to symbolic names.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>
Shand

See Also

Colours by a single variable can be assigned by \texttt{assignColLab}, symbols and colours by groups simultaneously by \texttt{assignSymbGroup}. Uniform colours are obtained by \texttt{assign1col}. Table of available plotting colours is obtained by \texttt{showColours}.

Examples

\begin{verbatim}
# Affects the colour of plotting symbols in the system (save=TRUE by default)
setTransparency(transp=0)
setTransparency(transp=0.5)
setTransparency(which.samples=c("Sa-1","Sa-2","Sa-3"),transp=0.5)
setTransparency(which.samples=c("Sa-1","Sa-2","Sa-3"),alpha=6a)

# No labels assigned
setTransparency(col=2,transp=0.5,save=FALSE)
setTransparency(col=c("blue","red"),transp=0.5,save=FALSE)
\end{verbatim}

Shand

\textit{A/CNK-A/NK diagram (Shand 1943)}

Description

Assigns data for Shand’s diagram into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'

Usage

\texttt{Shand()}

Details

Classic Shand’s diagram (1943). Three rock types are defined in the A/CNK vs A/NK plot:

\begin{itemize}
  \item \textit{Peralkaline}
  \item \textit{Metaluminous}
  \item \textit{Peraluminous}
\end{itemize}
Value

- sheet: list with Figaro Style Sheet data
- x.data: molecular ratio $A/\text{CNK}=\frac{\text{Al}_2\text{O}_3}{(\text{CaO} + \text{Na}_2\text{O} + \text{K}_2\text{O})}$
- y.data: molecular ratio $A/\text{NK}=\frac{\text{Al}_2\text{O}_3}{(\text{Na}_2\text{O} + \text{K}_2\text{O})}$

Author(s)

Vojtech Erban, <erban@sopky.cz>
& Vojtech Janousek, <vojtech.janousek@geology.cz>

References

Shand (1943) Eruptive Rocks. John Wiley & Sons

See Also

classify figaro plotDiagram NaAlK

Examples

#Within GCDkit, the plot is called using following auxiliary functions:
#To Classify data stored in WR (Groups by diagram)
classify("Shand")

#To plot data stored in WR or its subset (menu Classification)
plotDiagram("Shand", FALSE)
Description

Assigns data for the diagram of Shervais (1982) into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

Usage

Shervais()

Details

Discrimination diagram for basalts, as proposed by Shervais (1982) is based on variability of the Ti/V ratio under different oxygen fugacity.

Following environments may be distinguished:

ARC  Arc Tholeiites
OFB  Ocean Floor Basalts
showColours

Author(s)

Vojtech Erban, <erban@sopky.cz>
& Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

figaro plotDiagram

Examples

#plot the diagram
plotDiagram("Shervais", FALSE)

showColours

Show available colours

Description

Display colours available for plotting.

Usage

showColours(n=49)
showColours2(n=64)

Arguments

n numeric: number of colours to display

Details

The function ‘showColours’ displays a palette of plotting colours which can be specified by their numeric codes (1-49). On the other hand, ‘showColours2’ demonstrates the colours which can be given by their English names (there are some 657 of them).
Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also
'colours'

**showLegend**

<table>
<thead>
<tr>
<th>showLegend</th>
<th>Display legend</th>
</tr>
</thead>
</table>

Description
Displays a graphical legend(s) with assignment of plotting symbols and colours used by majority of the diagrams.

Usage

```r
showLegend(pch=labels$Symbol, col=labels$Colour, new.plot=TRUE, alt.leg=FALSE, just.colours=FALSE, GUI=FALSE)
```

Arguments

- **pch**: numeric or character: plotting symbols.
- **col**: numeric: code for their colour.
- **new.plot**: logical: shall be opened a new plotting window for the legend?
showSymbols

```
alt.leg  logical; should be the alternative (continuous) legend shown? See details.
just.colours  logical; in cases when two legends would be created, should be only that for
              plotting colours shown?
GUI  logical; Is teh function called from GUI (and not batch mode)?
```

Details

The internal variables ‘leg.col’ and ‘leg.pch’ are set to zero, if the current assignment is on the
basis of ‘groups’. Otherwise they contain the sequential number(s) of column(s) in the data frame
‘labels’ whose levels are to be used to build the legend(s).

If both variables differ, two legends are created, for plotting symbols and colours separately. This
is done unless ‘just.colours’ is set, when only legend for colours is displayed.

If both variables equal zero, the current grouping information is used.

If a complete colour scale is used for plotting symbols, for instance that created by the assignColVar
function, an alternative (continuous) legend can be drawn.

Value

None.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

Symbols and colours by a single label can be assigned by functions assignSymbLab and assignColLab
respectively, symbols and colours by groups simultaneously by assignSymbGroup. Symbols can
be colour-coded according to a variable using the function assignColVar. Uniform symbols are
obtained by assign1symb, uniform colours by assign1col. Table of available plotting symbols is
displayed by showSymbols and colours by showColours.

Examples

```
showLegend()
```

```
showSymbols  Show available symbols
```

Description

Shows numeric codes of symbols available for plotting:
**Usage**

```
showSymbols()
```

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

---

**Description**

Normalization of trace-element data by the given standard and spiderplot plotting.

**Usage**

```
spider.individual(new=TRUE)
```

```
spider.contour(chondrit = selectNorm(),what=NULL,
colour.palette = "heat.colors", ymin = 0, ymax = 0,
cex = 1,join = TRUE,pch = 15,
main = "",sub = "",offset = TRUE,centered = FALSE,
xrotate = FALSE, xaxs = "r", new = TRUE, legend = TRUE)
```

```
spider(rock=WR, chondrit = selectNorm(), ymin = 0,
ymax = 0, cex = NULL, plot = TRUE, join = TRUE,
field = FALSE, legend = FALSE, add = FALSE,
pch = NULL, col = NULL, shaded.col = "gray",
density = 0.02, angle = 0, main = "", sub = "",
offset = FALSE, centered = FALSE, xrotate = FALSE,
xaxs = "r", fill.col = TRUE, log = "y", new = TRUE, ...)
```
Arguments

new logical; if true, new plotting window is opened.
chondrit a numeric matrix with one row; the normalizing values.
what variable name or formula.
colour.palette variable name or formula.
rock a numeric matrix; the whole-rock data from which will be filtered out those to be normalized.
ymin, ymax y range of the diagram.
cex magnification of the plotting symbols.
plot logical; if set to FALSE, individual patterns are not plotted.
join logical; if TRUE, the NAs are extrapolated so that the patterns are unbroken.
field logical; if TRUE, a shaded field denoting the overall data span is plotted
legend logical; if TRUE, room for legend is reserved.
add logical; if FALSE, a new plot is started (otherwise overplot).
pch a vector specifying the plotting symbols.
col a numeric vector; colour of the plotting symbols and connecting lines.
fill.col logical; should be the field of overall variability filled by solid colour?
shaded.col numeric: colour for the cross-hatched or solid fill.
density numeric: density of the fill pattern (fraction of the whole plotting range).
gle angle numeric: angle of the fill pattern (in degrees).
main character: the main title for the plot.
sub character: the subtitle for the plot.
xrotate logical; shall be the element names on x axis rotated?
offset logical; shall be the names for odd and even elements shifted relative to each other?
centered logical; shall be the element names on x axis plotted in between tick marks?
xaxs style of the xaxis: see `help(par)` for details.
log which of the axes should be logarithmic?
... further graphical parameters: see `help(par)` for details.

Details

This is a quite flexible function, a true Mother of All Spiderplots, that can be used in a number of ways. It is employed by functions of the GCDkit system for normalization and plotting individual patterns for selected samples (`'spiderplot.r'`) or each of the groups (`'spider by group individual.r'`). In `spiderplot.r` is stored a user interface to `spider` for plotting individual patterns.
Function `spider` can also serve for plotting the overall compositional ranges (shown as cross-hatched fields or, optionally, semitransparent filled polygons) in a manner similar to function `spider by group.r`.
In `spiderplot_contour.r` is stored a user interface to `spider` for plotting individual patterns, in which the plotting symbols is uniform and colour reflects distribution of an independent variable, such as silica contents. The variable (or formula) can be specified using the parameter 'what', the colour scheme by 'colour.palette'. The legal colour schemes are: "grays", "reds", "blues", "greens", "cyans", "violets", "yellows", "cm.colors", "heat.colors", "terrain.colors", "topo.colors", "rainbow", "jet.colors".
The samples to be plotted can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSamples` for details. For choosing the correct normalization values serves the auxiliary function `selectNorm`. Then the user is prompted whether to use the currently assigned plotting symbols. If desired so, the symbols and colours can be specified in a simple spreadsheet-like interface. Likewise the scale of the y axis can be specified. The exact appearance of the labels to the x axis can be fine tuned by the arguments 'rotate.xlab', 'offset' and 'centered'. See examples.

If `plot=FALSE`, not plotting is done, and only the normalized values are returned.

### Value

- **results** numeric matrix with normalized concentrations.

### Note

If not specified, the parameters `pch`, `col` and `cex` are set up by default to 0 (circle), black and 1 numeric matrix, respectively. The only exception occurs when the plotting object is `WR`, when the missing plotting parameters are sought in the dataframe `labels`, i.e. among the standard plotting properties.

### Author(s)

- Vojtech Janousek, <vojtech.janousek@geology.cz>
- Vojtech Erban, <erban@sopky.cz>, contributed the algorithm hatching closed polygons

### See Also

For the syntax of the setup file with normalizing values and adding new normalization schemes see `selectNorm`; for further applications of `spider` see `spider2norm`, `spiderByGroupPatterns` and `spiderByGroupFields`.
Examples

```r
# spider.contour
ee <- spider.contour("Boynton","SiO2","reds",pch="*",cex=2,ymin=0.01,ymax=1000)

# the ee- construction redirects the textual output

# spider(WR,"Boynton",0.1,1000,pch="*",col="red",cex=2)

# Shade the background field portraying the overall variation

# Shade the background field portraying the overall variation
```

```r
# Custom normalization scheme
chon<-c(0.4,4,50,0.8,7,10,35,9,340,9,70,8.0)
chon<-matrix(chon,nrow=1)
colnames(chon)<-c("K2O","Rb","Ba","Th","Ta","Nb","Ce","Hf","Zr","Sm","Y","Yb")
rownames(chon)<-"ORG (Pearce et al. 1984)"
spider(WR,chon,ymin=0.01,col="navy",ymax=1000)
```

```r
# Possible styles for x axis
multiplePerPage(8,nrow=2,ncol=4,"Possible x axis styles", dummy=FALSE)
```

```r
ee<-spider(WR,"Boynton",0.1,1000,pch=labels$Symbol,col=labels$Colour,cex=labels$Size, 
offset=F,xrotate=F,centered=F,
main="offset=F, xrotate=F, centered=F",new=F)
ee<-spider(WR,"Boynton",0.1,1000,pch=labels$Symbol,col=labels$Colour,cex=labels$Size, 
offset=F,xrotate=T,centered=F,
main="offset=F, xrotate=T, centered=F",new=F)
ee<-spider(WR,"Boynton",0.1,1000,pch=labels$Symbol,col=labels$Colour,cex=labels$Size, 
offset=F,xrotate=F,centered=T,
main="offset=F, xrotate=F, centered=T",new=F)
ee<-spider(WR,"Boynton",0.1,1000,pch=labels$Symbol,col=labels$Colour,cex=labels$Size, 
offset=F,xrotate=T,centered=T,
main="offset=F, xrotate=T, centered=T",new=F)
ee<-spider(WR,"Boynton",0.1,1000,pch=labels$Symbol,col=labels$Colour,cex=labels$Size, 
offset=T,xrotate=F,centered=F,
main="offset=T, xrotate=F, centered=F",new=F)
ee<-spider(WR,"Boynton",0.1,1000,pch=labels$Symbol,col=labels$Colour,cex=labels$Size, 
offset=T,xrotate=F,centered=F,
main="offset=T, xrotate=F, centered=F",new=F)
ee<-spider(WR,"Boynton",0.1,1000,pch=labels$Symbol,col=labels$Colour,cex=labels$Size, 
offset=T,xrotate=T,centered=F,
main="offset=T, xrotate=T, centered=F",new=F)
ee<-spider(WR,"Boynton",0.1,1000,pch=labels$Symbol,col=labels$Colour,cex=labels$Size, 
offset=T,xrotate=T,centered=T,
main="offset=T, xrotate=T, centered=T",new=F)
```
spider2norm

spider(WR,"Boynton",plot=FALSE) # Calculation only

spider2norm

Spider plot(s): Selected samples, double normalized

Description

Plots a double normalized spiderplot. Trace-element data are first normalized by the given standard, as usual (see spider). Then the normalized concentrations are multiplied by a factor needed to adjust the normalized content of the selected element in each analysis to a desired value (such as unity). The goal is to eliminate effects of fractional crystallization (Thompson et al. 1983, Pearce et al. 2005, Pearce and Stern 2006).

Usage

spider2norm(rock=WR,norm=NULL,norm2=NULL,ymin=0,ymax=0,which=rep(TRUE,nrow(rock)),
legend=FALSE,pch=labels$Symbol,col=labels$Colour,plot=TRUE,join=TRUE,
shaded.col="gray",density=-1,angle=0,xaxs="r",fill.col=FALSE,field=FALSE,
add=FALSE,...)

Arguments

rock a numeric matrix; the whole-rock data from which will be filtered out those to be normalized.
norm a character string specifying the model.
norm2 name of the variable for the second normalization.
ymin, ymax y range of the diagram.
which specification of the samples to be plotted.
legend logical; if TRUE, room for legend is reserved.
pch a vector specifying the plotting symbols.
col a numeric vector; colour of the plotting symbols and connecting lines.
plot logical; if set to FALSE, individual patterns are not plotted.
join logical; if TRUE, the NAs are extrapolated so that the patterns are unbroken.
shaded.col numeric; colour for the cross-hatched fill.
density numeric; density of the fill pattern (fraction of the whole plotting range).
angle numeric; angle of the fill pattern (in degrees).
xaxs style of the xaxis: see `help(par)` for details.
fill.col colour for solid fill
field logical; if TRUE, a shaded field denoting the overall data span is plotted
add logical; if TRUE, a new plot is started (otherwise overplot).
... further graphical parameters: see `help(par)` for details.
**Details**

The parameter ‘*norm’ is an optional search pattern to query the available normalizing model names. It can contain a substring or even a regular expression. For choosing the correct normalization values serves the auxiliary function `selectNorm`. The function fails if no matches are found or the search is ambiguous. See `selectNorm` for details.

The samples to be plotted can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSamples` for details.

Then the user is prompted whether to use the currently assigned plotting symbols. If desired so, the symbols and colours can be specified in a simple spreadsheet-like interface. Likewise the scale of the y axis can be specified interactively.

**Value**

- **results** numeric matrix with normalized concentrations

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**References**


**See Also**

For the syntax of the setup file with normalizing values and adding new normalization schemes see `selectNorm`; for further variants of spiderplots, see `spider`, `spiderByGroupPatterns` and `spiderByGroupFields`.

**Examples**

```r
ee<-spider2norm(WR,"Boynton","Yb",0.1,1000,pch="*",col="red",cex=2)
# the ee<- construction redirects the textual output

ee<-spider2norm(WR,"Boynton","Yb",field=TRUE,density=0.05,angle=60,col="red",0.1,1000)

ee<-spider2norm(WR,"Boynton","Yb",field=TRUE,density=0.02,angle=45,col="gray",add=TRUE)
# Shade the background field portraying the overall variation
```

```r
# Shade the background field portraying the overall variation
```
spiderBoxplot

Spider plot(s): Selected samples - summary boxplot

Description
Normalization of geochemical data by the given standard (optionally also one of the samples) and spiderplot plotting. No individual patterns are drawn; instead, the statistical distribution of each element is portrayed by a boxplot.

Usage
spiderBoxplot(norm = NULL, which = rep(TRUE, nrow(WR)),
doublenorm = FALSE, norm2 = "",
ymin = NULL, ymax = NULL, bpplot = TRUE,
col = "lightgray", log = TRUE)

Arguments
norm a character string specifying the model.
which specification of the samples to be plotted.
doublenorm logical; should be the normalization employed? See details.
norm2 name of the variable for the second normalization.
ymin, ymax y range of the diagram.
bpplot logical; if FALSE, boxplot box (instead of box and percentile plot) is shown.
col fill colour.
log logical; should be the y axis scaled logarithmically?
Details

The parameter 'norm' is an optional search pattern to query the available normalizing model names. It can contain a substring or even a regular expression. The function fails if no matches are found or the search is ambiguous. See selectNorm for details.

The samples to be plotted can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see selectSamples for details.

For choosing the correct normalization values serves the auxiliary function selectNorm, which is the same as in ordinary spiderplots. If the user desires so, the data can be normalized by a sample present in the dataset. Then the elements to be plotted and their order is to be specified, as well.

Optionally, double normalization can be used. Trace-element data are first normalized by the given standard, then by the normalized content of the selected element in each analysis to eliminate effects of fractional crystallization (Thompson et al. 2003, Pearce et al. 2005, Pearce and Stern 2006). See spider2norm for details.

Distributions of individual normalized elements are plotted in the form of boxplot or box and percentile plot (Esty and Banfield 2003).

In both cases the box denotes 50% of the population (both quartiles), the horizontal line in the middle is a median and the whiskers denote the overall range. For boxplot this is without outliers. See manual entry for 'boxplot' and 'bpplot.my' for further details.

Printed are number of observations, missing values, mean, standard deviation, minimum, 25% quartile, median (=50% quartile), 75% quartile and maximum.

Value

results numeric matrix with statistical data for individual elements.

Warning

This function IS NOT Figaro-compatible. It means that the set of diagrams cannot be further edited in GCDkit (e.g. tools in "Plot editing" menu are inactive).
Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

For the syntax of the setup file with normalizing values and adding new normalization schemes see selectNorm; for further applications of `spider` see spiderByGroupPatterns, spider2norm and spiderByGroupFields.

Examples

spiderBoxplot("Boynton",col="yellow",bpplot=FALSE)
spiderBoxplot("Primordial Wood",doublenorm=TRUE,norm2="y",
  col="khaki",ymin=0.05,ymax=1000,bpplot=TRUE)

spiderByGroupFields  Spider plot(s) - by group fields

Description

Plots a series of spiderplots, for each group one, outlining the overall distribution as a field.

Usage

spiderByGroupFields(rock = WR, norm = NULL,
  bw = FALSE, fill = FALSE, ymin = 0, ymax = 0,
  xrotate = FALSE, offset = TRUE, centered = FALSE)

Arguments

  rock        a numeric matrix; the whole-rock data from which will be filtered out those to be normalized.
  norm        a character string specifying the model.
  bw          logical; should be the plot black and white?
  fill         logical; should be the fields filled by solid colour (and not hatched)?
spiderByGroupPatterns

ymin, ymax | y range of the diagram.
xrotate  | logical; shall be the element names on x axis rotated?
offset   | logical; shall be the names for odd and even elements shifted relative to each other?
centered | logical; shall be the element names on x axis plotted in between tick marks?

Details

The parameter 'norm' is an optional search pattern to query the available normalizing model names. It can contain a substring or even a regular expression. For choosing the correct normalization values serves the auxiliary function selectNorm. The function fails if no matches are found or the search is ambiguous. See selectNorm for details.

A series of spiderplots is plotted, for each group one, in which the whole variation range is outlined as filled/cross-hatched fields.

Value

None.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>;
Vojtech Erban, <erban@sopky.cz>, contributed the algorithm hatching closed polygons

See Also

For the syntax of the setup file with normalizing values and adding new normalization schemes see selectNorm. This function is based on spider.

Examples

data(sazava)
accessVar("sazava")
groupsByLabel("Intrusion")
spiderByGroupFields(norm="Boynton",ymin=1,ymax=1000)

spiderByGroupFields(norm="Boynton",bw=TRUE,ymin=1,ymax=1000,xrotate=TRUE,offset=FALSE)
spiderByGroupFields(norm="Boynton",fill=TRUE,ymin=1,ymax=1000)

spiderByGroupPatterns  Spider plot(s) - by group patterns

Description

Plots a series of spiderplots, for each group one, in which individual patterns are shown.

Usage

spiderByGroupPatterns(rock = WR, norm = NULL, bw = FALSE,
ymin = 0, ymax = 0, xrotate = FALSE, offset = TRUE, centered = FALSE)
Arguments

- **rock**: a numeric matrix; the whole-rock data from which will be filtered out those to be normalized.
- **norm**: a character string specifying the model.
- **bw**: logical; should the plot be black and white?
- **ymin, ymax**: y range of the diagram.
- **xrotate**: logical; shall the element names on x axis be rotated?
- **offset**: logical; shall the names for odd and even elements be shifted relative to each other?
- **centered**: logical; shall the element names on x axis be plotted in between tick marks?

Details

Firstly, the normalization scheme is chosen and scaling for all the plots is specified. Then, a series of spiderplots is plotted, for each group one, in which patterns for individual samples are shown.

Value

Returns a list 'results' with the normalized values, and, in case of REE, some extra parameters.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

For the syntax of the setup file with normalizing values and adding new normalization schemes see `selectNorm`. This function is based on `spider`.

Examples

```r
# Get the data ready
data(sazava)
accessVar("sazava")
groupsByLabel("Intrusion")

# Plot
spiderByGroupPatterns(norm="Boynton",ymin=1,ymax=1000)
spiderByGroupPatterns(norm="Boynton",bw=TRUE,ymin=1,ymax=1000,xrotate=TRUE,offset=FALSE)
```

---

**srnd**

Recalculations of the Sr-Nd isotopic data

Description

Age-corrects the Sr-Nd isotopic data to a given age; calculates initial $\epsilon(Nd)$ values and Nd model ages.
srnd

Usage

srnd(age="")

initial(x,age,system="Nd")

epsilon(WR,age)

DMage(WR)

DMGage(WR)

DMLHage(WR,age,RCC=0.12)

Arguments

- age: age in Ma: if empty, the user is prompted to enter a value
- x, WR: isotopic data to be recalculated
- system: character; which isotopic system Sr or Nd?
- RCC: numeric; the 147Sm/144Nd ratio of the intermediate crustal reservoir for calculation of the two-stage Nd model ages

Details

Recalculates the Sr-Nd isotopic data and returns them in the numeric matrix init with the following columns (DM = Depleted Mantle):

- Age (Ma): Age in Ma
- 87Sr/86Sr: Initial 87Sr/86Sr ratios
- 143Nd/144Nd: Initial 143Nd/144Nd ratios
- EpsNd: Initial \( \epsilon(Nd) \) values
- TDM: Single-stage DM Nd model ages (Liew & Hofmann, 1988), function DMage
- TDM.Gold: Single-stage DM Nd model ages (Goldstein et al., 1988), function DMGage
- TDM.2stg: Two-stage DM Nd model ages (Liew & Hofmann, 1988), function DMLHage

Value

init: numeric matrix with the results

Plugin

SrNd.r

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


Examples

```r
# recalculation to 500 Ma
srnd(500)

# print the isotopic parameters currently in the memory
init
```

### statsByGroup

**Statistics by groups**

**Description**

Calculates simple descriptive statistics for individual columns of the given data matrix; optionally this can be done for each of the groups separately.

**Usage**

```r
statsByGroup(data = WR, groups = groups)
```

**Arguments**

- `data` numeric data matrix.
- `groups` a vector, in which is specified, for each sample, a group it belongs to.

**Details**

The function returns a list containing the calculated statistical parameters respecting the desired grouping. The statistical summary involves number of observations, missing values, mean, standard deviation, minimum, 25% quartile, median (= 50% quartile), 75% quartile and maximum. This is a core function invoked both by `summarySingle` and `summarySingleByGroup`.

**Value**

- `results` a list with the results for individual groups

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

- `summarySingle`
- `statistics`
- `summaryAll`
- `summaryByGroup`
**Examples**

```r
statsByGroup(WR)

statsByGroup(WR[,LILE])
```

---

**Description**

Plots crosses in a binary diagram denoting means and standard deviations for individual groups.

**Usage**

```r
statsByGroupPlot()
```

**Details**

Displays a binary diagram of two elements/oxides in which are plotted averages for the individual groups with whiskers corresponding to their standard deviations.

The variables are entered via the function `selectColumnLabel`. In the specification of the variables can be used also arithmetic expressions, see `calcCore` for the correct syntax.

**Value**

- `results` a matrix with the results for individual groups and selected two elements/oxides

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

---

**statsIso**

*Statistical plots of isotopic ratios/model ages*

**Description**

Plots a boxplot or stripplot for a given isotopic parameter, respecting groups.

**Usage**

```r
boxplotIso(what=NULL)

stripplotIso(what=NULL)
```

**Arguments**

- `what` the variable name; see Details.
The boxplot portrays realistically a statistical distribution of the data. The box represents, for each of the groups, the two quartiles, the line inside is a median, the whiskers show the whole range without outliers.

Stripplot shows 1D scatter plots for each of the groups, with some artificial noise (jitter) added to make the individual points better visible. Stripplots are a good alternative to boxplots when sample sizes are small.
The variables to choose from are:

<table>
<thead>
<tr>
<th>Menu item</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>87Sr/86Sri</td>
<td>Initial Sr isotopic ratios</td>
</tr>
<tr>
<td>143Nd/144Nd</td>
<td>Initial Nd isotopic ratios</td>
</tr>
<tr>
<td>EpsNd</td>
<td>Initial ( \epsilon(Nd) ) values</td>
</tr>
<tr>
<td>1 stg DM model ages (Goldstein et al. 1988)</td>
<td>Single-stage DM Nd model ages</td>
</tr>
<tr>
<td>1 stg DM model ages (Liew &amp; Hofmann 1988)</td>
<td>Single-stage DM Nd model ages</td>
</tr>
<tr>
<td>2 stg DM model ages (Liew &amp; Hofmann 1988)</td>
<td>Two-stage DM Nd model ages</td>
</tr>
</tbody>
</table>

In addition, any variable names starting with the text "delta" will appear in this list.

The variable names can be specified also upon the function call, as the parameter "what". The possibilities are "87Sr/86Sri", "143Nd/144Nd", "EpsNd", "TDM.Gold", "TDM" or "TDM.2stg".

**Value**

a list object with data produced by the function `boxplot`.

**Plugin**

SrNd.r

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>
strip

291

References
and particulates from major river systems. Earth Planet Sci Lett 70: 221-236 doi: 10.1016/0012821X(84)90007-4
Liew T C & Hofmann A W (1988) Precambrian crustal components, plutonic associations, plate
environment of the Hercynian Fold Belt of Central Europe: indications from a Nd and Sr isotopic
See Also
’boxplot’

strip

Statistics: Stripplot by groups

Description
Stripplot for selected samples and variable, respecting the grouping.
Usage
strip(xlab = "", ...)
Arguments
xlab
...

variable name
additional parameters to stripplot

Details
Stripplot shows 1D scatter plots for each of the groups, with some artificial noise (jitter) added to
make the individual points better visible. Stripplots are a good alternative to boxplots when sample
sizes are small.
If no variable is specified as ’xlab’, the user can enter it using the function ’selectColumnLabel’.
In the specification of the variable can be used also arithmetic expressions, see calcCore for the
correct syntax.
Value
None.
Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>
See Also
stripplot, stripBoxplot
Examples
strip("(Na2O+K2O)/Al2O3")


stripBoxplot

Statistics: Stripplot by groups - with boxplots

Description
Stripplot for selected variable, respecting the grouping. Each of the stripplots for the individual groups are underlain by a boxplot, so that the median, quartiles and range are immediately apparent. Optionally, the data points can be replaced by variously sized/coloured circles, depicting a distribution of a second variable.

Usage

```r
stripBoxplot(yaxis=",zaxis="0",ymin=NULL,ymax=NULL,pal="heat.colors",ident=FALSE,
scaling.factor=NULL,boxplot.data=NULL,
pch=NULL,col=NULL,cex=NULL,silent=TRUE,add=FALSE)
```

Arguments

- `yaxis` specification of the variable used for stripplots/boxplots.
- `zaxis` (optional) specification of the variable depicted by the circles.
- `ymin, ymax` minimum and maximum of the y axis.
- `pal` name of predefined palette.
- `ident` logical; should be the samples identified interactively after plotting?
- `scaling.factor` numeric; relative size of the plotted symbols.
- `boxplot.data` a list; data for the underlying boxplots (if different from those used for the stripplots). See Details.
- `pch` plotting symbols.
- `col` plotting colours.
- `cex` relative size of the plotting symbols.
- `silent` logical, should be some of the above parameters chosen by the appropriate dialogues?
- `add` logical; should be the diagram added to a preexisting plot (rather than a new plotting window opened)?

Details
Stripplot shows 1D scatter plots for each of the groups, with some artificial noise (jitter) added to make the individual points better visible. Stripplots are a good alternative to boxplots when sample sizes are small.

If no variable is specified as an argument 'yaxis', and the function is invoked in interactive regime (silent = FALSE), the user can enter it using the function 'selectColumnLabel'.

If 'zaxis' is zero, assigned plotting symbols, colours and symbol sizes are used.

If 'zaxis' refers to a valid variable name, the data points are shown as circles, the size and colours of which correspond to this second variable. In the batch mode, the relative size of the circles plotted can be specified using the parameter scaling.factor.
In the specification of the variable(s) can be used also arithmetic expressions, see calcCore for the correct syntax.

The colour scheme can be specified by 'pal'. The legal colour schemes are: "grays", "reds", "blues", "greens", "cyans", "violets", "yellows", "cm.colors", "heat.colors", "terrain.colors", "topo.colors", "rainbow" and "jet.colors".

Normally, the stripplots are underlain by boxplots portraying the statistical distribution of the same data, as used for construction of stripplots for each of the groups. However, with caution, one can specify via boxplot.data a list containing the alternative data to be shown on background. Clearly, the number of components in the list, as well as their order, needs to exactly match the individual groups (the levels).

Value

None.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

stripplot, boxplot, strip.plotWithCircles

Examples

stripBoxplot("(Na2O+K2O)/Al2O3")
Subset by range  

Select subset by range

Description

Selecting subsets of the data stored in memory by their range.

Details

The menu item 'Select subset by range' is connected to the function `selectSubset`. The search pattern is treated as a selection of sample sequence numbers (effectively a list separated by commas that may also contain ranges expressed by colons). The current data will be replaced by its newly chosen subset.

Value

Overwrites the data frame 'labels' and numeric matrix 'WR' by subset that fulfills the search criteria.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

Examples

```r
## Not run:
Search pattern = 1:5
# First to fifth samples in the data set

Search pattern = 1,10
# First and tenth samples

Search pattern = 1:5, 10:11, 25
# Samples number 1, 2, ...5, 10, 11, 25

## End(Not run)
```

summaryAll

Statistics: Statistical summaries for the whole data set or its subset

Description

The function 'summaryAll' prints statistical summary for selected list of elements (majors as a default) and the current dataset (or its part). Functions 'summaryMajor' and 'summaryTrace' are entry points supplying the default lists for major- and trace elements.

Usage

```r
summaryAll(elems = major, where = NULL, show.boxplot = FALSE,
            show.hist = FALSE, silent=TRUE)
summaryMajor()
summaryTrace()
```
summaryAll

Arguments

elems list of desired elements
where list of desired samples to be processed
show.boxplot logical, should be plotted the boxplots?
show.hist logical, should be plotted the histograms?
silent logical, should be the above chosen by the appropriate dialogues?

Details

The statistical summary involves number of observations, missing values, mean, standard deviation, minimum, 25% quartile, median (= 50% quartile), 75% quartile and maximum. The function also plots summary boxplots and histograms, if desired so.

The samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see selectSamples for details.

Even though as a default are assumed majors (SiO2, TiO2, Al2O3, FeOt, MnO, MgO, CaO, Na2O, K20 for 'summaryMajor') or selected trace (Rb, Sr, Ba, Cr, Ni, La, Eu, Y, Zr for 'summaryTrace') elements, the variable(s) to be displayed can be modified/specified in all cases. To this purpose serves the function 'selectColumnsLabels'.

In the specification of the variable can be used also arithmetic expressions, see calcCore for the correct syntax.

Value

results numeric matrix with the results

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

statistics summarySingle summarySingleByGroup summaryByGroup

Examples

summaryAll(LILE)
summaryAll(LILE, show.hist=TRUE)
summaryAll(LILE, show.boxplot=TRUE)

# user-defined list
my.elems<-c("Rb","Sr","Ba")
summaryAll(my.elems)

## Not run:
# summaryMajor()
# summaryTrace()

## End(Not run)
**summaryByGroup**

*Statistics: Statistical summaries by groups*

**Description**

The function `summaryByGroup` prints a statistical summary for selected list of elements (majors as a default) and the whole dataset or its selection, respecting the current grouping. Functions `summaryByGroupMjr` and `summaryByGroupTrc` are entry points supplying the default lists for major- and trace elements. The function `summaryByGroupTrc` returns only ranges of the given parameter(s).

**Usage**

```r
summaryByGroup(elems = major, where = NULL, show.boxplot = FALSE, show.hist = FALSE, silent = TRUE)

summaryByGroupMjr()

summaryByGroupTrc()

summaryRangesByGroup(elems=major, where=NULL, silent=TRUE)
```

**Arguments**

- **elems** list of desired elements
- **where** list of desired samples to be processed
- **show.boxplot** logical, should be plotted the boxplots?
- **show.hist** logical, should be plotted the histograms?
- **silent** logical, should be the above chosen by the appropriate dialogues?

**Details**

The statistical summary involves number of observations, missing values, mean, standard deviation, minimum, 25% quartile, median (= 50% quartile), 75% quartile and maximum. The function also plots a summary boxplots and histograms, if desired so.

The samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSamples` for details.

The defaults are lists of major (SiO2, TiO2, Al2O3, FeOt, MnO, MgO, CaO, Na2O, K2O) or trace (Rb, Sr, Ba, Cr, Ni, La, Eu, Y, Zr) elements, respectively.

The desired variables are selected using the function `selectColumnsLabels`.

In the specification of the variable can be used also arithmetic expressions, see `calcCore` for the correct syntax.

**Value**

- **results** a list with the results for individual groups

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>
Examples

summaryByGroup(LILE)

summaryByGroup(LILE,show.hist=TRUE)

summaryByGroup(LILE,show.boxplot=TRUE)

# user-defined list
my.elems<-c("Rb","Sr","Ba/Sr")
summaryByGroup(my.elems)

## Not run:
summaryByGroupTrc()
summaryByGroupMjr()
summaryRangesByGroup(elems="Rb/Sr,Na2O+K2O")

## End(Not run)

summarySingle

Statistics: Single variable all/selection

Description

Prints statistical summary for a single variable and the current dataset (or its part).

Usage

summarySingle(xlab="")

Arguments

xlab variable name

Details

The statistical summary involves number of observations, missing values, mean, standard deviation, minimum, 25% quartile, median (=50% quartile), 75% quartile and maximum. The function also plots a summary boxplot and histogram.

In addition the statistical distribution of the given variable is shown as a boxplot, a box-percentile plot and two variants of histograms.
If no variable is specified as an argument `xlab`, the user can enter it using the function `selectColumnLabel`. In the specification of the variable can be used also arithmetic expressions, see `calcCore` for the correct syntax.

The samples can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSamples` for details.

**Value**

- `results`: numeric matrix/vector with the results

**Author(s)**

- Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

- `boxplot`
- `bpplot2`
- `statistics`
**summarySingleByGroup**

```r
summarySingleByGroup(xlab="")
```

**Arguments**

- `xlab` variable name

**Description**

Prints statistical summary for a single variable and the whole dataset, divided by groups.

**Usage**

```r
summarySingleByGroup(xlab="")
```

**Details**

The statistical summary involves number of observations, missing values, mean, standard deviation, minimum, 25\% quartile, median (= 50\% quartile), 75\% quartile and maximum. The function also plots a summary boxplot and histogram.

If no variable is specified as an argument `xlab`, the user can enter it using the function `selectColumnLabel`. In the specification of the variable can be used also arithmetic expressions, see `calcCore` for the correct syntax.

**Value**

- `results` numeric matrix with the results

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

- `boxplot` `summarySingle` `statistics` `summaryAll` `summaryByGroup`

**Examples**

```r
summarySingleByGroup("(Na2O+K2O)/Al2O3")
```
Description

Assigns data for a binary plot \((Al_2O_3 + CaO)/(FeOt + Na_2O + K_2O)\) vs. \(100 \times (MgO + FeOt + TiO_2)/SiO_2\), proposed by Sylvester (1989) to distinguish the alkaline collision-related alkaline granites into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

Usage

Sylvester()

Details

In the plot \((Al_2O_3 + CaO)/(FeOt + Na_2O + K_2O)\) vs. \(100 \times (MgO + FeOt + TiO_2)/SiO_2\) of Sylvester (1989) can be distinguished 'Alkaline' collision-related granites, from 'Calc-alkaline & Strongly peraluminous' types (solid line). The strongly fractionated calc-alkaline varieties are separated by the dashed line.

Note that only samples with \(SiO_2 > 68\) wt. % are plotted.
Value

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sheet</td>
<td>list with Figaro Style Sheet data</td>
</tr>
<tr>
<td>x.data</td>
<td>((\text{Al}_2\text{O}_3+\text{CaO})/(\text{FeO}_t+\text{Na}_2\text{O}+\text{K}_2\text{O})) [wt. %]</td>
</tr>
<tr>
<td>y.data</td>
<td>(100*(\text{MgO}+\text{FeO}_t+\text{TiO}_2)/\text{SiO}_2) [wt. %]</td>
</tr>
</tbody>
</table>

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

figaro plotDiagram

Examples

```r
#plot the diagram
plotDiagram("Sylvester", FALSE)
```

Description

Assigns data for IUGS recommended TAS diagram into Figaro template (list ‘sheet’) and appropriate values into ‘x.data’ and ‘y.data’

Usage

TAS(cutoff=95)

Arguments

cutoff numeric; the minimal sum of the analysis to be considered for classification

Details

TAS diagram, as proposed by Le Bas et al. (1986), codified by Le Maitre et al. (1989) and slightly modified by Le Bas (2000).
The diagram (in its basic form) defines the following fields:

- foidite
- picrobasalt
- basalt
- basaltic andesite
- andesite
- dacite
- rhyolite
- trachybasalt
- basaltic trachyandesite
- trachyandesite
- trachyte/trachydacite
- tephrite/basanite
- phonotephrite
- tephriphonolite
- phonolite

This primary division is further enhanced by the 'TASadd' routine (called automatically by 'classify'). Following actions are carried out:

- Analyses with $H_2O > 2$ and $CO_2 > 0.5$ (weight percent) are filtered out
- Trachybasalt is subdivided into hawaiite and potassic trachybasalt
- Basaltic trachyandesite is subdivided into mugearite and shoshonite
- Trachyandesite is subdivided into benmoreite and latite
TAS

- High-Mg rocks are split into *picrite*, *komatiite*, *meimechite* and *boninite*

Note that systematics of high-Mg rocks follows revised IUGS Recommendations (*Le Bas et al., 2000; Le Maitre et al., 2002*) which differ from their 1st edition (*Le Maitre et al. 1989*). Further subdivisions recommended by *Le Maitre et al. (1989)* are not implemented in GCDkit, mainly for poorly defined CIPW version used by the Subcommission.

The boundary between the subalkaline and alkaline domains is based on (*Irvine & Baragar 1971*).

**Value**

- `x.data` SiO2 data recast to anhydrous sum (matrix 'WRanh')
- `y.data` Na2O+K2O data recast to anhydrous sum (matrix 'WRanh')
- `sheet` list with Figaro Style Sheet data
- `results` matrix with classification results
- `groups` vector with classification results
- `grouping` set to -1

**Warning**

Note that, in accordance with the IUGS recommendation, the diagram is based on analyses recalculated on volatile-free basis.

**Author(s)**

Vojtech Erban, `<erban@sopky.cz>`

& Vojtech Janousek, `<vojtech.janousek@geology.cz>`

**References**


**See Also**

- `classify figaro plotDiagram`
Examples

# Within GCDkit, the plot is called using following auxiliary functions:
# To classify data stored in WR (Groups by diagram)
classify("TAS")

# To plot data stored in WR or its subset (menu Classification)
plotDiagram("TAS", FALSE)

TASMiddlemost

Middlemost’s modification of TAS diagram

Description

Assigns data for Middlemost’s modification of the TAS diagram into Figaro template (list ‘sheet’) and appropriate values into ‘x.data’ and ‘y.data’.

Usage

TASMiddlemostVolc()

TASMiddlemostPlut()

Details

Middlemost’s variation of classic IUGS-recommended TAS diagram, originally proposed by Le Bas et al. (1986). Boundaries of foidite, phonolite, trachyte, trachydacite and rhyolite fields are defined, as inferred from the phase relations in the TAS system. Moreover, the trachyte + trachydacite field is split into trachyte and trachydacite fields, silexite and sodalitite + nephelinolith + leucitolith fields are defined.
The same diagram layout is applied also to plutonic rocks as follows:

<table>
<thead>
<tr>
<th>Plutonic Rocks</th>
<th>Volcanic Rocks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peridot Gabbro</td>
<td>Picrobasalt</td>
</tr>
<tr>
<td>Gabbro</td>
<td>Basalt</td>
</tr>
<tr>
<td>Gabbroic Diorite</td>
<td>Basaltic Andesite</td>
</tr>
<tr>
<td>Diorite</td>
<td>Andesite</td>
</tr>
<tr>
<td>Granodiorite</td>
<td>Dacite</td>
</tr>
<tr>
<td>Granite</td>
<td>Rhyolite</td>
</tr>
<tr>
<td>Quartzolite</td>
<td>Silexite</td>
</tr>
<tr>
<td>Monzogabbro</td>
<td>Trachybasalt</td>
</tr>
<tr>
<td>Monzodiorite</td>
<td>basaltic Trachyandesite</td>
</tr>
<tr>
<td>Monzonite</td>
<td>Trachyandesite</td>
</tr>
<tr>
<td>Quartzmonzonite</td>
<td>Trachydacite</td>
</tr>
<tr>
<td>Syenite</td>
<td>Trachyte</td>
</tr>
<tr>
<td>Foid Gabbro</td>
<td>Tephrite</td>
</tr>
<tr>
<td>Foid Monzodiorite</td>
<td>Phonotephrite</td>
</tr>
<tr>
<td>Foid Monzosyenite</td>
<td>Tephriphonolite</td>
</tr>
<tr>
<td>Foid Syenite</td>
<td>Phonolite</td>
</tr>
<tr>
<td>Foidolite</td>
<td>Foidite</td>
</tr>
<tr>
<td>Tawite/Urtilte/Italite</td>
<td>sodalite/nephelinolith/leucitolith</td>
</tr>
</tbody>
</table>
Value

sheet  list with Figaro Style Sheet data
x.data  SiO2 weight percent
y.data  Na2O+K2O weight percent

Warning

Note that, unlike in the standard TAS plot, the diagram is based on standard analyses (i.e. it is not recalculated on the volatile-free basis).

Author(s)

Vojtech Erban, <erban@sopky.cz>

References


See Also

classify TAS Cox figaro plot Diagram
Examples

#Within GCDkit, the plot is called using following auxiliary functions:
#To Classify data stored in WR (Groups by diagram)
classify("TASMiddlemostVolc")
# or
classify("TASMiddlemostPlut")

#To plot data stored in WR or its subset (menu Classification)
plotDiagram("TASMiddlemostVolc", FALSE)
# or
plotDiagram("TASMiddlemostPlut", FALSE)

description

These functions plot/add data to a ternary plot.

Usage

ternary(x = NULL, y = NULL, z = NULL, samples = rownames(WR),
new = TRUE, grid = FALSE, ticks = TRUE, ...)

triplot(aa, bb, cc, alab, blab, clab, title = "", grid.int = 0,
tick.int = 0, label.axes = FALSE, line = FALSE,
pch = labels[names(aa), "Symbol"],
col = labels[names(aa), "Colour"],
cex = labels[names(aa), "Size"],
identify = getOption("gcd.ident"),
new = TRUE,...)

triplotadd(aa, bb, cc,
pch=labels[names(aa), "Symbol"],
col=labels[names(aa), "Colour"],
cex = labels[names(aa), "Size"],
labs=NULL, identify = FALSE, lines = FALSE, lty = "solid", type="p", lwd = 1)

Arguments

x character; specification of the plotting variable for the bottom left apex (formulae OK).
y character; specification of the plotting variable for the top apex (formulae OK).
z character; specification of the plotting variables for the bottom right apex (formulae OK).
grid logical; should be grid plotted?
ticks logical; should be ticks plotted?
samples character or numeric vector; specification of the samples to be plotted.
new logical; should be opened a new plotting window?
Further parameters to the functions 'ternary' and 'triplot'.

aa
a numerical vector, bottom left apex.

bb
a numerical vector, top apex.

c
a numerical vector, bottom right apex.

alab, blab, clab labels for the apices.

title
title for the whole diagram.

grid.int
interval of grid lines (0-1); if set to zero (default value), no grid is drawn.

tick.int
interval of ticks on axes (0-1); if set to zero (default value), no ticks are drawn.

label.axes
logical; if set to TRUE, axes are labeled by percentages of the components.

type
logical; if set to TRUE, lines are drawn instead of plotting points.

lty
line type.

lwd
line width.

pch plotting symbols.

col
plotting colours.

cex relative size of plotting symbols.

identify
logical; should be samples identified?

labs character; optional text to label the points.

type character; plot type; see plot.default.

Details

The function 'ternary' is the user interface to 'triplot'. The latter sets up the axes, labels the apices, plots the data and, if desired, enables the user to identify the data points interactively.

If 'new=TRUE', new plot window is opened.
The values for 'label.axes' are chosen according to 'tick.int' or 'grid.int'; if these are not available, labels are drawn by 10%.

'triplotadd' adds data points/lines to pre-existing ternary plot.

The variables to be plotted are selected using the function 'selectColumnLabel'.

In the specification of the apices can be used also arithmetic expressions, see calcCore for the correct syntax.

The functions are Figaro-compatible.

Value

A numeric matrix with coordinates of the data points recast to a sum of 1.

Author(s)

Jakub Smid <smid@prfdec.natur.cuni.cz> & Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

plot

Examples

ternary("Ba", "Rb*10", "Sr", col="red", pch="+")
ternary("SiO2/10","2*FeOt","K2O+5",samples=1:10,grid=TRUE)

triplot(WR[,"SiO2"]/10,WR[,"Na2O"]+WR[,"K2O"],WR[,"MgO"],"SiO2","A","MgO",
tick.int=0.1)

triplot(WR[,"Rb"]*10,WR[,"Sr"],WR[,"Ba"],"Rb","Sr","Ba",tick.int=0.05,
grid.int=0.1,pch="+",col="darkblue",label.axes=TRUE)

tetrad

Description
Lanthanide tetrad effect

Calculates lanthanide tetrad effect following the method of Irber (1999).

Usage

tetrad(method=NULL)

Arguments

method Normalization scheme.

Details

The method indicates which normalization scheme is to be used. It can be either 'Boynton' or 'Nakamura'. If not specified, the user is prompted to choose it interactively by the function spider.

The anomalies of individual elements are calculated as follows for the first tetrad:

\[
Ce/Cet = \frac{Ce_N}{La_N^{\frac{4}{3}} * Nd_N^{\frac{1}{3}}}
\]

\[
Pr/Prt = \frac{Pr_N}{La_N^{\frac{4}{3}} * Nd_N^{\frac{1}{3}}}
\]

\[
t_1 = \sqrt{Ce/Cet * Pr/Prt}
\]

By analogy, one can define for the third tetrad:

\[
Tb/Tbt = \frac{Tb_N}{Gd_N^{\frac{2}{3}} * Ho_N^{\frac{1}{3}}}
\]

\[
Dy/Dyt = \frac{Dy_N}{Gd_N^{\frac{2}{3}} * Ho_N^{\frac{1}{3}}}
\]

\[
t_3 = \sqrt{Tb/Tbt * Dy/Dyt}
\]

The magnitude of the tetrad effect is then calculated as a geometric mean:

\[
t_3 = \sqrt{t_1 * t_3}
\]
threeD

Value

Returns a matrix 'results' with the following columns:

- Ce/Cet  Ce anomaly
- Pr/Prt  Pr anomaly
- t1      first tetrad
- Tb/Tbt  Tb anomaly
- Dy/Dyt  Dy anomaly
- t3      third tetrad
- TE1-3   degree of lanthanide tetrad effect, geometric mean of t1 and t3

Plugin

tetrad.r

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

spider

Examples

tetrad("Boynton")

threeD  3D plot

Description

Plots a 3-D plot of three specified variables.

Usage

threeD(xlab='', ylab='', zlab='')

Arguments

- xlab  Name of the data column to be used as x axis.
- ylab  Name of the data column to be used as y axis.
- zlab  Name of the data column to be used as z axis.
Details

This function displays three variables in a form of 3D plot. The plot can be rotated interactively, if required so.

The samples to be plotted can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSubset` for details.

If no parameters `xlab`, `ylab` and `zlab` are given, the user is prompted to specify them.

The variables are selected using the function `selectColumnLabel`.

In the specification of the apices can be used also arithmetic expressions, see `calcCore` for the correct syntax.

See manual entry for `cloud` for further details.

Value

None.

Warning

This function IS NOT Figaro-compatible.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz> & Vojtech Erban, <erban@sopky.cz>

Examples

`threeD("SiO2", "Na2O+K2O", "MgO+FeOt")`
tkSelectVariable

TclTk GUI: Select a single variable

Description

Function to select a single variable using the Tcl/Tk-based Graphical User Interface (GUI).

Usage

```r
tkSelectVariable(top.frame = NULL, where = colnames(WR), preselect = 2,
pack = FALSE, message = "Select a variable", background = "wheat",
variable = "x", on.leave = function() {}, row = 0, column = 0, height = 15,
width = 50, buttons = FALSE, state = "normal")
```

Arguments

- `top.frame` name of the parental frame
- `where` character; names of variables to be chosen from
- `preselect` numeric; which item is to be preselected
- `pack` logical; pack the frame?
- `message` character; textual prompt
- `background` colour for the frame background
- `variable` character; variable name with the output
- `on.leave` function to be invoked upon leave
- `row, column` coordinates within the parental frame
- `height, width` size of the frame
- `buttons` logical; should the frame have also buttons?
- `state` ???

Details

The buttons are: Reset, SortUp, SortDown, OK, Cancel.

Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also

tcltk-package
**tk_winDialog**

**Description**

Tcl/Tk replacement for the MS Windows-specific function 'winDialog'.

**Usage**

```tcl
tk_winDialog(type="ok",message="")
```

**Arguments**

- **type** Character; the type of the dialogue box.
- **message** Character. The information field of the dialogue box.

**Details**

This is a platform-independent implementation of the MS Windows-specific function 'winDialog', written using the Tcl/Tk. Possible types of the dialogue box are: ok, okcancel, yesno and yesnocancel.

**Value**

A character string giving the name of the button pressed (in capitals).

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

`winDialog tkmessageBox tk_winDialogString tcltk-package`

**Examples**

```tcl
tk_winDialog(type="yesnocancel",message="Are you sure?")
```

---

**tk_winDialogString**

**Description**

Tcl/Tk replacement for the MS Windows-specific function 'winDialogString'.

**Usage**

```tcl
tk_winDialogString(message="Enter variable",default="",returnValOnCancel=NULL)
```

---
**trendTicks**

**Arguments**

- **message** Character. The information field of the dialog box.
- **default** Character; the default string.
- **returnValOnCancel** Character; a value to be returned when the dialogue is canceled.

**Details**

This is a platform-independent implementation of the MS Windows-specific function `winDialogString`, written using the Tcl/Tk.

**Value**

A character string giving the contents of the text box when Ok was pressed, or value specified by `returnValOnCancel` if Cancel was pressed.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**See Also**

`winDialogString tkentry tk_winDialog tcltk-package`

**Examples**

```
tk_winDialogString(message="Enter x value",default="15.7")
```

---

**trendTicks**

**Petrogenetic trends**

**Description**

Adding a trend with arrow and tick marks to a pre-existing GCDkit plot.

**Usage**

```
trendTicks(equation, x, xmin = par("usr")[1], xmax = par("usr")[2],
            tick = abs(par("tcl")), text = FALSE, col = "blue", lty = "solid",
            lwd = 1, arrow = FALSE, autoscale = TRUE)
```

**Arguments**

- **equation** character or expression; a valid formula expressed as a function of `x`.
- **x** numeric; `x` values where the ticks are to be drawn.
- **xmin** numeric; beginning of the trend.
- **xmax** numeric; end of the trend.
- **tick** numeric; length of a tick as a fraction of the height of a line of text.
- **text** logical; should the tick marks annotated by `text`?
trendTicks

- **col**: text or numeric; plotting colour specification.
- **lty**: text or numeric; the line type.
- **lwd**: numeric; the line width, a positive number, defaulting to 1.
- **arrow**: logical; should be also an arrow head shown?
- **autoscale**: logical; should the plot be autosized in order to accommodate the whole trend as well as all data points?

**Details**

Using the function `curve`, the function `trendTicks` adds to an existing GCDkit plot a linear or curved trend with tick marks and (optionally) arrow head. It is required that the trend is defined as a function of `x`. The slope of the individual tick marks is then determined using a derivation of the main function at the respective points.

**Value**

A list with two components, `x` and `y`, with coordinates of the tick marks.

**Warning**

Autoscaling will work only with Figaro compatible plots!
Author(s)
Vojtech Janousek, <vojtech.janousek@geology.cz>

See Also
par

Examples

```r
binary("Ba","Sr",xmin=200,xmax=2000,ymin=10,ymax=400)
equation<="x/8+200"
x<-seq(2000,500,by=-100)
trendTicks(equation,x,min(x),max(x),col="darkred",lty="solid",lwd=2,arrow=TRUE,text=FALSE)

plot(1,1,type="n",xlim=c(0.01,1),ylim=c(0,1),xlab="Rb",ylab="Sr",log="x")
equation<="6*x/8"
x<-seq(0.01,1,by=0.1)
trendTicks(equation,x,min(x),max(x),col=2,lwd=2,arrow=FALSE,text=FALSE,autoscale=FALSE)
```

Verma Major-element based discrimination plots for (ultra-)basic rocks (Verma et al. 2006)

Description
Plots data stored in 'WR' (or its subset) into discrimination plots proposed by Verma et al. (2006) for (ultra-) basic rocks ($\text{SiO}_2 < 52$ wt. %).

Usage

```r
Verma(FeMiddlemost=FALSE,GUI=FALSE)
```

Arguments

- `FeMiddlemost` logical, should be iron adjusted according to Middlemost (1989)?
- `GUI` logical, is the function called from a GUI?

Details
Suite of five diagrams for discrimination of geotectonic environment of ultrabasic and basic rocks ($\text{SiO}_2 < 52$ wt. %), proposed by Verma et al. (2006). It is based on log-transformed concentration ratios of major-element oxides. Note that prior to the transformation, the analyses are recast to 100% anhydrous basis. Each diagram is a plot of two discriminant functions, DF1 and DF2, respectively in x- and y-axes. Only samples with $\text{SiO}_2 < 52$ wt. % are plotted. To work properly, the major element analysis should be complete ($\text{SiO}_2, \text{TiO}_2, \text{Al}_2\text{O}_3, \text{Fe}_2\text{O}_3, \text{FeO}, \text{MnO}, \text{MgO}, \text{CaO}, \text{Na}_2\text{O}, \text{K}_2\text{O}, \text{P}_2\text{O}_5$). Following the recommendation by Verma et al. (2006), prior to the plotting can be performed an adjustment of the iron-oxidation ratio as proposed by Middlemost (1989) (see 'FeMiddlemost').

For the $\text{Fe}_2\text{O}_3/\text{FeO}$ ratios implemented for individual rock types (based on TAS classification), see Verma et al. (2002) (Fig. 1).
Following geotectonic settings may be deduced:

<table>
<thead>
<tr>
<th>Abbreviation used</th>
<th>Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAB</td>
<td>island arc basic rocks</td>
</tr>
<tr>
<td>CRB</td>
<td>continental rift basic rocks</td>
</tr>
<tr>
<td>OIB</td>
<td>ocean-island basic rocks</td>
</tr>
<tr>
<td>MORB</td>
<td>mid-ocean ridge basic rocks</td>
</tr>
</tbody>
</table>

**Value**

None.

**Note**

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots. See Plate, Plate editing and figaro for details.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**References**


See Also

FeMiddlemost Agrawal Plate Plate editing plotPlate figaro

Examples

# plot the diagrams
plotPlate("Verma")

---

Villaseca | B-A plot (modified by Villaseca et al. 1998)

Description

The B-A diagram as proposed by Debon and Le Fort (1983) with classification fields for various types of peraluminous rocks designed by Villaseca et al. (1998).

Usage

Villaseca()

Details

Plots modified B-A diagram (designed originally by Debon and Le Fort 1983) with fields for various peraluminous rock types after Villaseca et al. (1998). Assigns data for the B-A diagram into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

The following fields are defined:

- l-P low peraluminous
- m-P moderately peraluminous
- h-P highly peraluminous
- f-P felsic peraluminous
- metaluminous

Rocks with composition falling beyond defined boundaries are labeled ‘undefined’ by the ‘classify’ function.

Parameters for the diagram are calculated by the function ‘DebonCalc’. All of them are based on millications (1000 gram-atoms per 100 grams).

\[
A = Al - (K + Na + 2 Ca)
\]
\[
B = Fe + Mg + Ti
\]

For details, see Debon & Le Fort (1983) or (1988).

**Value**

- sheet list with Figaro Style Sheet data
- x.data B value. See details.
- y.data A value. See details.
Author(s)

Vojtech Janousek, <vojtech.janousek@geology.cz>

References


See Also

classify figaro plotDiagram DebonCalc Debon

Examples

#plot the diagram
plotDiagram("Villaseca",FALSE)

Wedge diagrams (Ague 1994)

Implementation of Wedge diagrams after Ague (1994) and Bucholz and Ague (2010) used for judging the mobility of elements or oxides in course of various geochemically open-system processes such as alteration or partial melting.

Usage

Wedge(x = "Ti", y = NULL, protolith = NULL, outline = "chull", precision = 10, plotAltered = TRUE, xmin = 0, ymin = 0, xmax = NULL, ymax = NULL, fun = NULL)

Arguments

x a single geochemical species presumably immobile during the given rock transformation.
y list of elements/oxides for plotting, separated by commas.
protolith Boolean search pattern to specify the protolith samples in the data file.
outline method for contouring the clusters of protolith and product compositions, see Details.
precision precision of contours drawn, if ‘outline’="contour", see Details.
plotAltered logical; should be the altered analyses plotted or just contoured?
xmin, xmax (optional) limits for shared x axes of the individual plots.
ymin (optional) minimum for all of the y axes of the plots.
ymax (optional) upper limits for each of the y axes of the plots.
fund panel function to be applied to each of the individual plots.
Details

Wedge diagrams (Ague 1994) enable qualitative treatment of losses/gains of geochemical species (elements or oxides) during open-system geological processes, such as alteration, metamorphism or partial melting. As such they represent a viable alternative to the isocon plots (Grant 1986, 2005) or concentration ratio diagrams (Ague 1994). However, the Wedge diagrams have an advantage in that they take into account the overall variability of the whole dataset (both of the putative protolith and the altered product) and not just a selected whole-rock pair.

Wedge diagrams are simple binary plots of a potentially mobile element $j$ versus a reference (immobile) element $i$. The compositionally heterogeneous protolith samples yield a cloud of points. The outer edges of this cloud define a wedge-shaped region that converges towards the origin. As shown by Bucholz and Ague (2010), the altered samples that plot above and to the left of this wedge are thought to have gained the mobile species $j$, whereas those falling below and to the right suffered its loss. The samples that remain in the wedge but moved upwards are thought to record residual enrichment, and those shifted downwards to have underwent a residual dilution.
The samples defining the protolith variation can be selected based on combination of three searching mechanisms (by sample name/label, range or a Boolean condition) - see `selectSamples` for details. Implemented are two methods for outlining the clusters of the protolith and altered compositions (as specified by the argument 'outline'), convex hull (`chull`) and contour (`contour`). For the latter, the shape of the contours drawn can be controlled using the parameter (`precision`). The higher it is, the smoother contours result. See `contourGroups` and `chullGroups` for further details.

Optionally, the individual data points for the altered samples may be replaced by contours portraying their density, if `plotAltered = FALSE`. Parameters `xmin, xmax, ymin` and `ymax` are passed to the function `plotWithLimits` used for the actual data plotting.

Optionally, panel function specified by `fun` with two arguments, `xlab` and `ylab`, is applied to each of the plots.

**Value**

Returns a matrix 'results' of slopes of tie-lines from individual protolith samples to the origin (with a component for each diagram, i.e. for each species evaluated). Lines of maximum and minimum slopes are those which are plotted as dashed lines, thus defining the wedge of the protolith variation (see Details).

**Plugin**

`Isocon.r`

**Note**

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots. See `Plate, Plate editing` and `figaro` for details.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**References**


**See Also**

`Ague, isocon, Plate, Plate editing, chull, contour contourGroups chullGroups, plotWithLimits`
Examples

```r
data(sazava)
accessVar("sazava")
Wedge("Ti","SiO2,FeOt,MgO,CaO,Na2O,K2O",
protolith="Intrusion="Sazava"","chull")

# Using the default precision of 10
Wedge("Ti","Zr,Nb,Sr,Rb,Ba",protolith="Intrusion="Sazava"","contour")

Wedge("Ti","Zr,Nb,Sr,Rb,Ba",protolith="Intrusion="Sazava"","contour",precision=100)
```

---

**Whalen**

A type granitoids *(Whalen et al. 1987)*

---

**Description**

Set of discrimination plots to distinguish A-type granitoids as defined by *Whalen et al. (1987)*.

**Usage**

```r
Whalen(plot.txt = getOption("gcd.plot.text"))
```

**Arguments**

- `plot.txt` logical, annotate fields by their names?

**Details**

Set of binary plots proposed by *Whalen et al. (1987)* to distinguish A-type granitoids on the one hand from ordinary/fractionated I- and S-types on the other.

In total 12 diagrams are plotted split into two pages. Apart from fields for I and S type granites ('I & S'), sometimes split into ordinary ('OGT') and fractionated ('FG') domains, average composition of the A-type granites (labeled 'A') are shown. See Figs 1, 2 and 5 in the original paper *(Whalen et al. 1987)* for comparison.
The following diagrams are plotted: $Zr + Nb + Ce + Y$ vs. $FeOt/MgO$ and $(K_2O + Na_2O)/CaO$; $10000Ga/Al$ vs. $K_2O + Na_2O$, $(K_2O + Na_2O)/CaO$, $K_2O/MgO$ and $FeOt/MgO$; $10000Ga/Al$ vs. $Zr$, $Nb$, $Ce$, $Y$, $Zn$ and Agpaitic Index.

**Value**

To the matrix 'WR' are appended two columns, with Ga/Al ratios and values of the Agpaitic Index (labeled 'A.I.').

**Note**

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots. See Plate, Plate editing and figaro for details.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**References**


**See Also**

Plate Plate editing plotPlate figaro
Examples

# plot the diagrams
plotPlate("Whalen")

---

**WinFloyd1**

*Nb/Y - Zr/TiO2 diagram (Winchester + Floyd 1977)*

---

**Description**

Assigns data for Nb/Y vs. Zr/TiO2 diagram into Figaro template (list 'sheet') and appropriate values into 'x.data' and 'y.data'.

**Usage**

WinFloyd1()

**Details**

Classification diagram proposed by *Winchester & Floyd (1977)*.

---

Using incompatible element ratios (Nb/Y vs. Zr/TiO2), following fields are defined:

- Trachyandesite
- Alkali basalt

---

![Nb/Y - Zr/TiO2 plot (Winchester and Floyd 1977)](image-url)
**WinFloyd2**

- Basanite/Nephelinite
- Trachyte
- Phonolite
- Comendite/Pantellerite
- Rhyolite
- Rhyodacite/Dacite
- Andesite
- Andesite/Basalt
- Subalkaline basalt

**Value**

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sheet</td>
<td>list with Figaro Style Sheet data</td>
</tr>
<tr>
<td>x.data</td>
<td>Nb/Y wt. % ratio</td>
</tr>
<tr>
<td>y.data</td>
<td>(Zr/TiO₂)*0.0001 wt. % ratio</td>
</tr>
</tbody>
</table>

**Author(s)**

Vojtech Erban, <erban@sopky.cz>
& Vojtech Janousek, <vojtech.janousek@geology.cz>

**References**


**See Also**

classify figaro plotDiagram

**Examples**

#Within GCDkit, the plot is called using following auxiliary functions:
#To Classify data stored in WR (Groups by diagram)
classify("WinFloyd1")

#To plot data stored in WR or its subset (menu Classification)
plotDiagram("WinFloyd1", FALSE)

---

**WinFloyd2**

*Zr/TiO₂ - SiO₂ (Winchester + Floyd 1977)*

**Description**

Assigns data for Zr/TiO₂ vs. SiO₂ diagram into Figaro template (list `sheet`) and appropriate values into `x.data` and `y.data`

**Usage**

WinFloyd2()
Classification diagram proposed by Winchester & Floyd (1977).

Using incompatible element ratio and silica ($Zr/TiO_2$ vs. $SiO_2$), following fields are defined:

- Trachyandesite
- Basanite/Trachyte/Nephelinite
- Phonolite
- Trachyte
- Comendite/Pantellerite
- Rhyolite/Dacite
- Rhyodacite/Dacite
- Andesite
- Subalkaline basalt
- Alkaline basalt

Value

- sheet list with Figaro Style Sheet data
- y.data SiO2 wt. %
- x.data ($Zr/TiO_2$)*0.001 wt. % ratio
Author(s)

Vojtech Erban, <erban@sopky.cz>
& Vojtech Janousek, <vojtech.janousek@geology.cz>

References

Winchester J A & Floyd P A (1977) Geochemical discrimination of different magma series and their

See Also

classify figaro plotDiagram

Examples

#Within GCDkit, the plot is called using following auxiliary functions:
#To Classify data stored in WR (Groups by diagram)
classify("WinFloyd2")

#To plot data stored in WR or its subset (menu Classification)
plotDiagram("WinFloyd2", FALSE)

Wood

Wood (1980)

Description

Assigns Figaro templates to Wood’s geotectonic diagrams for basaltoids into the list 'plate' and
appropriate values into the list 'plate.data' for subsequent plotting.

Usage

Wood(ident = getOption("gcd.ident"),
     plot.txt = getOption("gcd.plot.text"))

Arguments

ident logical, identify?
plot.txt logical, annotate fields by their names?

Details

A series of triangular diagrams with apices Th-Hf/3-Ta, Th-Hf/3-Ta and Th-Zr/117-Nb/16, pro-
posed by Wood (1980).
Following fields are defined:

- IAT: Island-arc Tholeiites
- CAB: Calc-alkaline Basalts
- N-MORB: N-type Mid-ocean Ridge Basalts
- E-MORB: E-type Mid-ocean Ridge Basalts
- WPT: Within-plate Tholeiites
- WPA: Alkaline Within-plate Basalts

**Value**

- sheet: list with Figaro Style Sheet data
- x.data, y.data: Th, Hf/3 and Ta in ppm recalculated into two dimensions

**Note**

This function uses the plates concept. The individual plots can be selected and their properties/appearance changed as if they were stand alone Figaro-compatible plots. See Plate, Plate editing, and figaro for details.

**Author(s)**

Vojtech Janousek, <vojtech.janousek@geology.cz>

**References**


**See Also**

Plate, Plate editing, plotPlate, figaro
Examples

```
#plot the diagrams
plotPlate("Wood")
```

<table>
<thead>
<tr>
<th>YbN vs. LaN/YbN</th>
<th>YbN vs LaN/YbN (Martin 1986) TTG/adakite</th>
</tr>
</thead>
</table>

**Description**

Assigns data for the $Yb_N$ vs. $La_N/Yb_N$ diagram for adakite/TTG discrimination into Figaro template (list `sheet`) and appropriate values into `x.data` and `y.data`.

**Usage**

```
LaYb(ybrep=FALSE)
```

**Arguments**

- `ybrep` logical, should be missing Yb values replaced by Y/2.4?

**Details**

Diagram proposed by several authors for discriminating between adakitic (or TTG) and "ordinary" calc-alkaline rocks. The version used here is from Martin (1986).

Martin (1999) suggested that Yb (ppm) could be replaced by Y (ppm)/2.4. Notionally this could help with old data with missing values. However if Yb is missing La is also likely to be absent (or unreliable), so by default this replacement is not done; the user can access it by calling the function with `ybrep=TRUE` (not available from GUI).

Normalization values are $La = 0.33$ ppm, $Yb = 0.22$ ppm (Nakamura 1974).

**Value**

- `sheet` list with Figaro Style Sheet data
- `x.data` $La_N/Yb_N$ or $La/(Y/2.4)$
- `y.data` $Yb_N$

**Author(s)**

Jean-Francois Moyen, `<jfmoyen@gmail.com>`

**References**


zrSaturation

See Also

figaro plotDiagram

Examples

# plot the diagram
plotDiagram("LaYb",FALSE)

zrSaturation

Zircon saturation (Watson + Harrison 1983, Boehnke et al. 2013)

Description

Calculates zircon saturation temperatures for the observed major-element data and Zr concentrations. Returns also Zr saturation levels for the given major-element compositions and assumed magma temperature.

Usage

zrSaturation(cats = milli, T = 0, Zr = filterOut(WR, "Zr", 1))

Arguments

cats numeric matrix; whole-rock data recast to millications
T assumed temperature of the magma in °C
Zr numeric vector with Zr concentrations

Details

Calculates Zr saturation concentration at a given temperature. Given 'T' is the estimated absolute temperature (K) of the magma and 'M' is a cationic ratio:

\[
M = 100 \frac{Na + K + 2Ca}{Al.Si}
\]

it can be written (Watson & Harrison 1983):

\[
D_{Zr} = e^{(-3.8 - 0.85(M - 1) + \frac{12900}{T})}
\]

The Zr saturation level is then given by:

\[
Zr.sat = \frac{497644}{D_{Zr}}
\]

On the other hand, the saturation temperature can be obtained from the observed Zr concentration and magma composition (assuming no zircon inheritance)

\[
D_{Zr} = \frac{497644}{Zr}
\]
This model has been reformulated by Kelsey et al. (2008) as follows:

\[ D_{Zr} = e^{\left(\frac{12900}{\ln(D_{Zr}) + 3.8 + 0.85(M - 1)}\right)} - 273.15 \]

using the cationic ratio \( FM \)

\[ FM = \frac{Na + K + 2(Ca + Fe + Mg)}{Al + Si} \]

Utilizing the model of Kelsey et al. (2008), the zircon saturation temperature can be calculated as:

\[ TZr.sat.C = \frac{11574}{\ln(\frac{497644}{Zr}) + 0.679FM + 1.7965} - 273.15 \]

A re-calibration of the (Watson & Harrison 1983) model by Boehnke et al. (2013) has yielded:

\[ D_{Zr} = e^{\left(\frac{10108}{\ln(\frac{497644}{Zr}) + 1.16(M - 1) - 1.48}\right)} \]

and

\[ TZr.sat.C = \frac{10108}{\ln(\frac{497644}{Zr}) + 1.16(M - 1) + 1.48} - 273.15 \]

Value

Returns a matrix ‘results’ with the following columns:

- \( M \)  cationic ratios
- \( Zr \)  observed Zr concentrations
- \( Zr.sat \)  saturation levels of Zr after Watson & Harrison (1983) for assumed temperature
- \( TZr.sat.C \)  zircon saturation temperatures after Watson & Harrison (1983) in °C
- \( Zr.sat \) (Boehnke)  saturation levels of Zr after Boehnke et al. (2013) for assumed temperature
- \( TZr.sat.C \) (Boehnke)  zircon saturation temperatures after Boehnke et al. (2013) in °C

Plugin

Saturation.r

Author(s)

Vojíček Janoušek, <vojtech.janousek@geology.cz>
References


Index

*Topic aplot
Add contours, 8
assign1symb, 23
assignSymbGroup, 26
assignSymbLab, 27
assignSymbLett, 28
contourGroups, 52
figAdd, 90
figCol, 95
figEdit, 96
figOverplot, 101
figOverplotDiagram, 103
figScale, 107
figUser, 108
figZoom, 109
overplotDataset, 177
peterplot, 207
plateAddReservoirs, 215
showColours, 270
showSymbols, 272

*Topic cluster
cluster, 51
groupsByCluster, 121

*Topic color
assign1col, 22
assignColLab, 24
assignColVar, 25
figAdd, 90
figCol, 95
figOverplot, 101
figOverplotDiagram, 103
figUser, 108
plateAddReservoirs, 215
selectPalette, 260
setTransparency, 266
showColours, 270

*Topic database
Boolean conditions, 35
Regular expressions, 241
selectColumnLabel, 255
selectColumnsLabels, 256
selectSubset, 262
Subset by range, 294

*Topic datasets
atacazo, 29
blatna, 35
sazava, 249

*Topic device
figRedraw, 105
pdfAll, 185
psAll, 232

*Topic dplot
calcAnomaly, 41
selectNorm, 258
setCex, 264

*Topic file
customScript, 63
EarthChem, 75
Export to Access, 82
Export to DBF, 83
Export to Excel, 84
Export to HTML tables, 85
figLoad, 98
figSave, 106
loadData, 146
mergeData, 152
pdfAll, 185
peekDataset, 206
pokeDataset, 222
psAll, 232
purgeDatasets, 233
r2clipboard, 238
saveData, 247
saveResults, 247
saveResultsIso, 248

*Topic hplot
AFM, 10
ageEps, 12
Agrawal, 14
ArcMapSetup, 21
Batchelor, 30
binary, 32
binaryBoxplot, 33
bpplot2, 37
Cabanis, 38
cluster, 51
coplotByGroup, 54
coplotTri, 56
correlationCoefPlot, 58
Cox, 59
Debon, 65
DebonBmgNo, 67
DebonBQF, 69
DebonKNaB, 71
DebonQB, 73
elemIso, 79
epsEps, 81
F-M-W diagram, 87
figLoad, 98
figMulti, 98
figSave, 106
filledContourFig, 110
Frost, 111
Frost_2008, 114
graphicsOff, 119
Grebennikov, 120
groupsByDiagram, 122
Harris, 124
Hastie, 125
Hollocher, 127
isochron, 131
Jensen, 136
Laroche, 140
LaurentSource, 144
Maniar, 149
Meschede, 153
Middlemost, 155
mins2deg, 158
Miyashiro, 160
Molecular weights, 163
Mullen, 164
MullerK, 166
Multiple plots, 169
NaAlK, 172
OConnor, 175
pairsCorr, 181
Paulick, 183
Pearce and Cann, 186
Pearce and Norry, 187
Pearce Nb-Th-Yb, 189
Pearce Nb-Ti-Yb, 191
Pearce1982, 193
Pearce1996, 194
PearceDestructive1, 196
PearceDestructive2, 198
PearceDestructive3, 199
PearceEtAl, 201
PearceGranite, 202
PeceTaylor, 204
phasePropPlot, 209
Plate, 211
Plate editing, 213
plateLabelSlots, 217
plotPlate, 219
plotWithCircles, 220
profiler, 227
projbioplot, 230
QAPF, 235
reciprocalIso, 240
Ross, 243
Schandl, 250
Shand, 267
Shervais, 269
spider, 273
spider2norm, 279
spiderBoxplot, 281
spiderByGroupFields, 283
spiderByGroupPatterns, 284
statsByGroupPlot, 288
strip, 291
stripBoxplot, 292
summaryAll, 294
summaryByGroup, 296
summarySingle, 297
summarySingleByGroup, 299
Sylvester, 300
TAS, 301
TASMiddlemost, 304
ternary, 307
threeD, 311
Verma, 317
Villaseca, 319
Walchen, 324
WinFloyd1, 326
WinFloyd2, 327
Wood, 329
YbN vs. LaN/YbN, 331

*Topic iplot
contourGroups, 52
figAdd, 90
figaro.identify, 94
figCol, 95
figEdit, 96
figGbo, 97
figOverplot, 101
figOverplotDiagram, 103
figUser, 108
gcdOptions, 116
ID, 130
overplotDataset, 177
INDEX

plateAddReservoirs, 215
selectByDiagram, 253
showLegend, 271

*Topic manip
.claslist, 6
accessVar, 7
addResults, 9
addResultsIso, 9
Ague, 16
appendSingle, 18
apSaturation, 19
Boolean conditions, 35
calc, 40
calcCore, 43
Catanorm, 44
classify, 47
clr.transform, 49
customScript, 63
cutMy, 64
deleteSingle, 74
Edit labels, 77
Edit numeric data, 78
editLabFactor, 78
FeMiddlemost, 89
groupsByCluster, 121
groupsByDiagram, 122
groupsByLabel, 123
isocon, 133
joinGroups, 137
Jung, 138
LaRocheCalc, 143
Mesonorm, 154
millications, 157
Misc, 159
Mode, 161
mzSaturation, 171
Niggli, 174
oxide2oxide, 180
oxide2ppm, 181
ppm2oxide, 223
QANOR, 233
recast, 239
rtSaturation, 245
selectAll, 252
selectByDiagram, 253
selectByLabel, 254
selectColumnLabel, 255
selectColumnLabels, 256
selectSubset, 262
srnd, 285
Subset by range, 294
tetrad, 310
tk_winDialog, 314
tk_winDialogString, 314
tkSelectVariable, 313
trendTicks, 315
Wedge, 321
zrSaturation, 332

*Topic misc
.about, 6
crosstab, 62
info, 131
quitGCDkit, 238

*Topic multivariate
cluster, 51
groupsByCluster, 121
pairsCorr, 181
prComp, 224

*Topic print
printSamples, 225
printSingle, 226
setShutUp, 265

*Topic univar
statsByGroup, 287
statsByGroupPlot, 288
statsIso, 288
strip, 291
stripBoxplot, 292
summaryAll, 294
summaryByGroup, 296
summarySingle, 297
summarySingleByGroup, 299
.claslist, 6
87Sr/86Sr vs EpsNd1 (epsEps), 81
about, 6
accessExport, 83, 85
accessExport (Export to Access), 82
accessVar, 7, 222, 233
Add contours, 8
addContours, 111
addContours (Add contours), 8
addResults, 9, 10, 231
addResultsIso, 9
AFM, 10, 49, 122, 243, 244
ageEps, 12
ageEps2 (ageEps), 12
ageSr (ageEps), 12
Agrawal, 14, 319
Ague, 16, 323
annotate, 96
appendSingle, 18
apSaturation, 19
ArcMapSetup, 21
assign1col, 22, 23, 24, 26–29, 262, 267, 272
assign1symb, 23, 24, 27–29, 272
assignColLab, 23, 24, 26–29, 262, 267, 272
assignColVar, 22, 25, 262, 272
assignSymbGroup, 23, 24, 26, 28, 29, 267, 272
assignSymbLab, 23, 24, 27, 29, 272
assignSymbLett, 28, 28
atacazo, 29
Batchelor, 30
binary, 32, 100, 212
binaryBoxplot, 33
biplot.princomp, 223
blatna, 35
Boolean conditions, 35
boxplot, 34, 37, 282, 290, 291, 293, 298, 299
Boxplot isotopic ratios/model ages (statsIso), 288
boxplotIsso (statsIso), 288
bpplot2, 37, 298
Cabanis, 38
calc, 40
calcAnomaly, 41
calcCore, 33, 34, 40, 43, 63, 80, 170, 220, 225, 227, 229, 288, 291, 293, 295, 296, 298, 299, 309, 312
Cann (Pearce and Cann), 186
Catanorm, 44
chull, 53, 323
chullGroups, 92, 93, 323
chullGroups (contourGroups), 52
CIPW, 45, 116, 177
CIPWhb (CIPW), 45
cloud, 312
clr.trans (clr.transform), 49
clr.transform, 49
cluster, 51
colours, 93, 95, 247, 271
Colours: Plotting symbols (figCol), 95
Colours: set to B&W (FigCol), 95
Colours: subtitle (figCol), 95
Colours: title (figCol), 95
contour, 8, 52, 53, 323
contourGroups, 52, 92, 93, 323
coplot, 54–58
coplotByGroup, 54
coplotTri, 56
Correlation coefficient patterns (correlationCoefPlot), 58
correlog (correlationCoefPlot), 58
Cox, 49, 59, 122, 306
CoxPlut (Cox), 59
CoxVolc (Cox), 59
crosstab, 62
curve, 91, 93, 316
customScript, 63
cut, 64
cutMy, 64
data.entry, 77–79
dbfExport, 83, 85
dbfExport (Export to DBF), 83
Debon, 65, 321
DebonBA, 49, 69, 71, 72, 74, 122
DebonBA (Debon), 65
DebonBMgNo, 67, 67, 71, 72, 74
DebonBQF, 67, 69, 69, 72, 74
DebonCalc, 67, 69, 71, 72, 74, 321
DebonKNaB, 67, 69, 71, 71, 74
DebonPQ, 49, 69, 71, 72, 74, 122
DebonPQ (Debon), 65
DebonQB, 67, 67, 71, 72, 73
deleteSingle, 74
dev.off, 119
Dage (srnd), 285
DMGage (srnd), 285
DMLHage (srnd), 285
EarthChem, 75
Edit labels, 77
Edit numeric data, 78
Edit: subtitle (figEdit), 96
Edit: title (figEdit), 96
Edit: x label (figEdit), 96
Edit: y label (figEdit), 96
eeditData (Edit numeric data), 78
eeditLabels (Edit labels), 77
eeditLabFactor, 78, 121
elemIso, 79
epsEps, 81
epsilon (srnd), 285
excel2007Export (Export to Excel), 84
eexcelExport, 82, 83
eexcelExport (Export to Excel), 84
Export to Access, 82
Export to DBF, 83
Export to Excel, 84
Export to HTML tables, 85
F-M-W diagram, 87
INDEX

FeMiddlemost, 89, 317, 319
figAdd, 90
figAddArrow(figAdd), 90
figAddBox(figAdd), 90
figAddCurve(figAdd), 90
figAddFit(figAdd), 90
figAddReservoirs, 102, 103, 105, 179, 216
figAddReservoirs(figAdd), 90
figAddText(figAdd), 90
figAlab(figEdit), 96
figaro.identify, 94
figBlab(figEdit), 96
figBw(figCol), 95
figCex(figScale), 107
figCexLab(figScale), 107
figCexMain(figScale), 107
figCexSub(figScale), 107
figClab(figEdit), 96
figCol, 95, 214
figColMain(figCol), 95
figColSub(figCol), 95
figColSub(figCol), 95
figColSub(figCol), 95
figEdit, 96
figFixLim(figZoom), 109
figGbo, 97
figGrid(figAdd), 90
figIdentify(figaro.identify), 94
figLegend(figAdd), 90
figLoad, 98, 106, 212
figMain(figEdit), 96
figMulti, 98, 212
figOverplot, 91–93, 101, 105, 179, 216
figOverplotDiagram, 93, 103, 103, 179, 216
figRedraw, 105
figSave, 98, 106, 212
figScale, 107, 214
figSub(figEdit), 96
figTicks(figAdd), 90
figUnzoom(figZoom), 109
figUser, 108
figXlab(figEdit), 96
figXlim(figZoom), 109
figYlab(figEdit), 96
figYlim(figZoom), 109
figZoom, 109
figZooming(figZoom), 109
filled.contour, 8, 111
filledContourFig, 110
Frost, 111, 116
Frost_2008, 113, 114
gcdOptions, 116, 120, 265
getwd, 149
graphicsOff, 119
Grebennikov, 120
groupsByCluster, 121, 122, 123
groupsByDiagram, 122, 122, 123
groupsByLabel, 122, 123
Harris, 124
Hastie, 49, 122, 125
hclust, 52
Highlight multiple points
(figaro.identify), 94
highlightSelection(figaro.identify), 94
Hollocher, 127
Hollocher1 (Hollocher), 127
Hollocher2 (Hollocher), 127
HTMLTableMain(Export to HTML tables), 85
HTMLTableOrdered(Export to HTML tables), 85
HTMLTableResults, 231
HTMLTableResults(Export to HTML tables), 85
HTMLTableWR(Export to HTML tables), 85
ID, 119, 130
identify, 94, 119, 130
Identify points (figaro.identify), 94
info, 131
initial (srnd), 285
isochron, 131
isocon, 18, 133, 323
isoconAtoms (isocon), 133
isoconOxides (isocon), 133
Jensen, 49, 122, 136
joinGroups, 137
Jung, 138
kde2d, 8
LaRoche, 31, 144
LaRoche (Laroche), 140
INDEX

Laroche, 49, 122, 140
LaRocheCalc, 143, 143
LarochePlut (Laroche), 140
LarocheVolc (Laroche), 140
LaurentSource, 144
LaYb (YbN vs. LaN/YbN), 331
llgridlines, 21, 22
loadData, 21, 22, 146, 152, 222, 233, 247, 253
loadDataOdbc (loadData), 146

Maniar, 149
menuet, 117, 119
merge, 152
mergeData, 149, 152, 247
mergeDataCols (mergeData), 152
mergeDataRows (mergeData), 152
Mescede, 153
Mesonorm, 154, 235
Middlemost, 49, 122, 155
MiddlemostPlut (Middlemost), 155
mlications, 143, 157, 231
mins2deg, 158
Misc, 159
Miyashiro, 49, 122, 160
Mode, 161
ModeAll (Mode), 161
ModeC (Mode), 161
ModeMain (Mode), 161
Molecular weights, 163
molecularWeight, 180, 181, 223
molecularWeight (Molecular weights), 163
Mullen, 164
MullerK, 166
MüllerKbinary (MullerK), 166
MüllerKternary (MullerK), 166
multiple, 212
multiple (Multiple plots), 169
Multiple plots, 169
multipleMjr (Multiple plots), 169
multiplePerPage (Plate), 211
multipleTrc (Multiple plots), 169
mzSaturation, 171

NaAlK, 49, 122, 172, 268
Niggl1, 174
normalize2total (recast), 239
Norry (Pearce and Norry), 187
OConnor, 49, 122, 175
OConnorPlut (OConnor), 175
OConnorVolc (OConnor), 175
OhtaArai (F-M-W diagram), 87
options, 117–119, 130, 265

overplotDataset, 93, 103, 105, 177, 216
oxide2oxide, 180, 181, 223
oxide2ppm, 180, 181, 223
pairsCorr, 181
pairsMjr (pairsCorr), 181
pairsTrc (pairsCorr), 181
panel.cor (pairsCorr), 181
panel.cov (pairsCorr), 181
panel.hist (pairsCorr), 181
panel.smooth (pairsCorr), 181
par, 8, 93, 103–105, 109, 228, 317
Paulick, 183
pdf, 185
pdfAll, 185, 232
Pearce and Cann, 186
Pearce and Norry, 187
Pearce Nb-Th-Yb, 189
Pearce Nb-Ti-Yb, 191
Pearce1982, 193
Pearce1996, 49, 122, 194
PearceDestructive1, 196
PearceDestructive2, 198
PearceDestructive3, 199
PearceEtAl, 201
PearceGranite, 202
PearceNbThYb, 192
PearceNbThYb (Pearce Nb-Th-Yb), 189
PearceNbTiYb, 190
PearceNbTiYb (Pearce Nb-Ti-Yb), 191
Pecetaylor, 49, 122, 204
peekDataset, 179, 206, 206, 222, 233
peterplot, 207
phasePropPlot, 209
Plate, 15, 99, 100, 113, 115, 116, 151, 168,
170, 171, 186, 187, 204, 211, 214,
218, 219, 252, 318, 319, 323, 325,
330
Plate editing, 15, 99, 100, 113, 115, 116,
151, 168, 170, 171, 186, 187, 204,
212, 213, 218, 219, 252, 318, 319,
323, 325, 330
plate0YLim (Plate editing), 213
plateAddReservoirs, 103, 179, 215
plateAnnotationsRemove (Plate editing),
213
plateBW (Plate editing), 213
plateCex (Plate editing), 213
plateCexLab (Plate editing), 213
plateCexMain (Plate editing), 213
plateCol (Plate editing), 213
plateExpand (Plate editing), 213
plateExtract, 104, 105, 178, 179
INDEX

plateExtract (Plate editing), 213
plateLabel$Slots, 217
plateLoad(Plate), 211
platePch (Plate), 213
platePS (Plate), 211
plateRedraw (Plate), 211
plateSave (Plate), 211
plateXlim (Plate editing), 213
plateYlim (Plate editing), 213
plot, 33, 34, 212, 309
plot.default, 91, 102, 216, 308
plotPlate, 15, 113, 116, 151, 168, 187, 204, 212, 214, 219, 252, 319, 325, 330
plotWithCircles, 220, 293
plotWithLimits, 13, 80, 82, 241, 323
plotWithLimits (binary), 32
points, 102–104
pokeDataset, 148, 149, 179, 206, 222, 222, 233
polygon, 52, 53
postscript, 232
ppm2oxide, 180, 181, 223
pr.comp.clr (clr.transform), 49
prComp, 51, 224
princomp, 225
printSamples, 225
printSingle, 226
profiler, 227
projbiocoords(projbioplot), 230
projbioplot, 230
psAll, 185, 232
purgeDatasets, 179, 206, 222, 233
QANOR, 154, 155, 233
QAPF, 49, 122, 235
QAPFPlut (QAPF), 235
QAPFvoic (QAPF), 235
quantile, 26
quit, 238
quitGCDkit, 238
r2clipboard, 231, 238
read.table, 149, 247
readOGR, 22
recast, 239
recipcalliso, 240
refreshfig(figRedraw), 105
regex, 36, 242, 263
regular expression, 263
regular expression (Regular expressions), 241
regular expressions, 241
regular expressions (Regular expressions), 241
regular expression (Regular expressions), 241
regular expressions, 36, 256
regular expressions (Regular expressions), 241
Ross, 243
rtSaturation, 245
saveData, 149, 152, 247
saveResults, 247, 249
saveResultsIsso, 248
sazava, 249
Scale: axis labels (figScale), 107
Scale: subtitle (figScale), 107
Scale: symbols (figScale), 107
Scale: title (figScale), 107
Schandl, 250
selectAll, 252, 253, 263
selectById, 253
selectByLabel, 253, 254, 263
selectColumnLabel, 24, 25, 27, 28, 33, 34, 40, 55, 57, 64, 74, 79, 80, 123, 177, 220, 227, 254, 255, 288, 291, 292, 298, 299, 309, 312
selectColumnLabels, 52, 59, 182, 224, 225, 239, 256, 295, 296
selectDataset, 222, 233
selectDataset (peekDataset), 206
selectNorm, 42, 92, 93, 103, 216, 258, 277, 280, 282–285
selectPalette, 24, 25, 260
selectSamples, 52, 80, 82, 132, 162, 182, 224, 225, 241, 266, 277, 280, 281, 295, 296, 298, 323
selectSamples (selectSubset), 262
selectSubset, 33, 34, 36, 94, 111, 177, 221, 229, 253, 258, 262, 294, 312
setCex, 117, 119, 264
setShutUp, 265
setTransparency, 178, 179, 266
setwd, 149
Shand, 49, 113, 116, 122, 174, 267
Shervais, 269
showColours, 23, 24, 26–29, 93, 95, 148, 149, 213, 214, 247, 262, 267, 270, 272
showLegend, 23, 24, 27–29, 271
showSymbols, 23, 24, 27–29, 148, 149, 213, 214, 247, 272, 272
sp. 22
spider, 42, 100, 102, 103, 212, 258, 273, 279, 280, 284, 285, 310, 311
spider2norm, 277, 279, 282, 283
spider_double_norm.r (spider2norm), 279
spiderBoxplot, 281
spiderByGroupFields, 277, 280, 283, 283
spiderByGroupPatterns, 277, 280, 283, 284
spiderplot.r (spider), 273
srnd, 285
statistics, 287, 295, 298, 299
Statistics: Correlation: majors
(pairsCorr), 181
Statistics: Correlation: traces
(pairsCorr), 181
Statistics: Majors
summaryAll/selection (summaryAll), 294
Statistics: Majors summaryByGroups (summaryByGroup), 296
Statistics: Trace
summaryAll/selection (summaryAll), 294
Statistics: Trace summaryByGroups (summaryByGroup), 296
statsByGroup, 287
statsByGroupPlot, 288
statsIso, 288
strip, 291, 293
stripBoxplot, 291, 292
stripplot, 291, 293
Stripplot isotopic ratios/model ages
(statsIso), 288
stripplotIso (statsIso), 288
Subset by range, 294
subsetBoolean, 263
subsetBoolean (Boolean conditions), 35
subsetRange, 263
subsetRange (Subset by range), 294
summaryAll, 287, 294, 299
summaryByGroup, 287, 295, 296, 299
summaryByGroupMjr (summaryByGroup), 296
summaryByGroupTrc (summaryByGroup), 296
summaryMajor (summaryAll), 294
summaryRangesByGroup (summaryByGroup), 296
summarySingle, 287, 295, 297, 299
summarySingleByGroup, 287, 295, 299, 299
summaryTrace (summaryAll), 294
Sylvester, 300
TAS, 49, 61, 62, 90, 104, 122, 178, 301, 306
TASadd (TAS), 301
TASMiddlemost, 49, 122, 304
TASMiddlemostPlut (TASMiddlemost), 304
TASMiddlemostVolc (TASMiddlemost), 304
ternary, 100, 212, 230, 307
tetrad, 310
treeD, 311
tk_winDialog, 314, 315
tk_winDialogString, 314, 314
tkentry, 315
tkmessageBox, 314
tkSelectVariable, 313
trendTicks, 315
triplot (ternary), 307
triplotadd, 102, 103
triplotadd (ternary), 307
underplotDataset, 105
underplotDataset (overplotDataset), 177
Verma, 15, 90, 317
Villaseca, 49, 122, 319
Wedge, 18, 321
Whalen, 324
winDialog, 314
winDialogString, 315
WinFloyd1, 49, 122, 196, 326
WinFloyd2, 49, 122, 327
Wood, 329
WRComp (Mode), 161
write.dbf, 83
YbN vs. LaN/YbN, 331
Zooming: Scale x axis (figZoom), 109
Zooming: Scale y axis (figZoom), 109
Zooming: Zoom in (figZoom), 109
Zooming: Zoom out to original size (figZoom), 109
zrSaturation, 332