

Chapter 2

Data Manipulation and Simple Calculations

2.1 Loading and Manipulating Data

When starting a new session in R, with or without *GCDkit*, the first task is to import data. In plain R, tabular data, common in igneous geochemistry, can be loaded most conveniently by the `read.table` command. The other possibility is to access one of the built-in datasets using the `data` command. In *GCDkit* the analyses are usually read by the `loadData` function, or copied from any Windows application (such as a spreadsheet) via the clipboard.

Seldom are all the values available for all the samples. There are two types of missing data: concentrations below the detection limit and those not determined. If statistical evaluation is desired, especially by multivariate methods, information that elemental concentration is lower than a certain threshold fundamentally differs from the situation when it is completely unknown and can attain any value. One of the available strategies to deal with the first case (Rock 1988; van den Bogaard and Tolosana-Delgado 2013) is to replace the data below the detection limit by its half (Reimann et al. 2008). Clearly, thus estimated values should not represent a high proportion of the given variable. Regarding the not analyzed data, the R language has facilities to handle appropriately any completely missing values (denoted NA), e.g. during plotting or mean calculations.

Once data are loaded, the most common tasks include display, subsetting (modest database functionality) and simple arithmetics. Below we are going to practice such skills on vectors and data frames, the most common data types.



Exercise 2.1: Subsetting a numeric vector, vector operations

GCDkit contains several built-in datasets, normally needed just for correct system functioning. One of these is atomic weights, stored in the named vector `mw`. We will use it to show some simple vector operations.

- Display the whole vector. What is the atomic weight of Rb?
- What is the average value of the whole vector?
- Which atoms have the atomic weight higher than 170?
- Display the names of six lightest and six heaviest elements in the dataset.



```
a) GCDkit-> mw
      Ag      Al      As      Au      B      Ba
107.86820  26.98154  74.92160  196.96650  10.81100  137.33000
      Be      Bi      Br      C      Ca      Cd
  9.01218 208.98040  79.90400  12.01100  40.07800  112.41000 ...

GCDkit-> mw["Rb"]
      Rb
85.4678

b) GCDkit-> mean(mw)
[1] 107.9206

c) GCDkit-> names(mw)[mw>170]
[1] "Au" "Bi" "Hf" "Hg" "Ir" "Lu" "Np" "Os" "Pb" "Pt" "Pu"
[12] "Re" "Ta" "Th" "Tl" "U"  "W"  "Yb"

d) GCDkit-> sort(mw)[1:6]
      H      Li      Be      B      C      N
  1.00797  6.94100  9.01218  10.81100  12.01100  14.00670
GCDkit-> rev(sort(mw))[1:6]
      Pu      U      Np      Th      Bi      Pb
244.0640 238.0290 237.0482 232.0381 208.9804 207.2000
```



Exercise 2.2: Loading files, matrix/data frame manipulations

The file *sazava.data* contains selected major- and trace-element analyses from the ~354 Ma old Sázava suite of the Central Bohemian Plutonic Complex (CBPC; Bohemian Massif, Czech Republic) (Janoušek et al. 2000, 2004).

- Read analyses stored in the tab-delimited data file into a data frame `WR`.
- Find out the names of available variables (= column names).
- What is the MgO content of sample Sa-1?
- Show all available numeric data for samples Po-1 and Po-4.
- Calculate the total of the column "Na₂O".
- Display names of three samples with the lowest and the highest SiO₂ contents.
- Calculate averages of all variables.
- Display a table with three columns: SiO₂, MgO and Na₂O/K₂O.



sazava.data



```
a) > sazava <- read.table("sazava.data", sep="\t")
```

```
GCDkit-> loadData("sazava.data") # Alternative in GCDkit1
GCDkit-> sazava <- cbind(labels,WR)

b) > colnames(sazava)
[1] "Intrusion" "Locality" "Petrology" "Outcrop"
[5] "Symbol" "Colour" "SiO2" "TiO2"
[9] "Al2O3" "FeO" "Fe2O3" "MnO"...
```

```
c) > sazava["Sa-1", "MgO"]
[1] 3.21
```

```
d) > sazava <- sazava[, -(1:6)] # Stripping 1st six columns
> sazava[c("Po-1", "Po-4"), ]
      SiO2 TiO2 Al2O3 FeO Fe2O3 MnO MgO CaO Na2O K2O
Po-1 62.95 0.28 20.02 1.65 0.67 0.05 0.55 6.61 3.91 1.99
Po-4 71.09 0.30 15.09 2.12 0.38 0.06 0.52 3.75 3.68 1.87 ...
```

```
e) > sum(sazava[, "Na2O"])
[1] 39.13
```

```
f) > silica <- sazava[, "SiO2"]
> names(silica) <- rownames(sazava)
> names(sort(silica)) [1:5]
[1] "Gbs-2" "Gbs-1" "Sa-4" "Gbs-20" "SaD-1"
> names(rev(sort(silica))) [1:5]
[1] "Po-5" "Po-4" "Po-3" "Po-1" "Sa-1"
```

```
g) > apply(sazava, 2, mean, na.rm=TRUE)
      SiO2      TiO2      Al2O3      FeO
57.95285714 0.63928571 16.94285714 4.73071429
      Fe2O3      MnO      MgO      CaO
1.74642857 0.13785714 3.57000000 8.16000000 ...
```

```
h) > x <- cbind(sazava[, "SiO2"], sazava[, "MgO"],
+   sazava[, "Na2O"]/sazava[, "K2O"])
> colnames(x) <- c("SiO2", "MgO", "Na2O/K2O")
> rownames(x) <- rownames(sazava)
> x
      SiO2 MgO Na2O/K2O
Sa-1 59.98 3.21 1.008000
Sa-2 55.17 3.67 1.976471 ...
```

2.2 Linking Whole-Rock Chemistry with Mineral Stoichiometry

After loading, a common task is to recast the bulk geochemical analyses into several indexes, related to the mineralogy of the rocks. This is often followed by

¹ Caution, two variables will be created in this case. WR will contain only the numeric values, all textual information will be transferred into data frame labels. See Appendix B for details.

normative recalculations. The aim is to better understand the modal chemistry, classification and, together with statistical methods, the distribution of elements within the dataset. R, as a statistical language, is well suited to such a task. Furthermore, user-defined functions serve to add new features tailored to our needs.

2.2.1 Basic Indexes

The calculations in R can be best demonstrated on an example of simple geochemical indexes. Arguably the most used, and quite powerful, are the recalculations of Fe as total ferrous or ferric oxides, and two types of Mg numbers²:

$$FeOt = FeO + 0.89981 \times Fe_2O_3[wt. \%] \quad (2.1)$$

$$mg\# = 100 \frac{MgO}{FeO + MgO} [mol. \%] \quad (2.2)$$

$$Mg\# = 100 \frac{MgO}{FeOt + MgO} [mol. \%] \quad (2.3)$$

Many major-element based diagrams are constructed using some conversion of wt. % oxides into cation numbers; this allows easy comparison with mineral formulae. For instance, the popular alumina saturation index, A/CNK (Shand 1943) (sometimes also abbreviated as ASI) mimics the stoichiometry of feldspars:

$$A/CNK = \frac{Al_2O_3}{CaO + Na_2O + K_2O} [mol. \%] \quad (2.4)$$

A similar index, distinguishing peralkaline rocks (with excess alkalis), is:

$$A/NK = \frac{Al_2O_3}{Na_2O + K_2O} [mol. \%] \quad (2.5)$$

If $A/CNK > 1$, there is excess Al over the amount needed to form feldspars. Such rocks are termed peraluminous, while those with $A/CNK < 1$ and $A/NK > 1$ are metaluminous and those with $A/CNK \sim 1$ subaluminous (Shand 1943). The A/CNK value has a direct link to modal mineralogy—presence of Ca- and/or alkali-rich phases such as amphiboles and pyroxenes indicates an Al deficit and the host rock is metaluminous. Biotite is weakly peraluminous and thus its occurrence points to weakly peraluminous nature of the rock (Miller 1985). Strongly peraluminous granitoids (*sensu* Miller 1985) contain additional more peraluminous phases like muscovite, or even aluminosilicates (kyanite, sillimanite or andalusite), cordierite, garnet, tourmaline, topaz or corundum (Clarke 1981). However, the

² The Fe numbers (Frost et al. 2001; Frost and Frost 2008) are defined analogously.

definition of the ASI does differ between authors—sometimes the Ca is corrected for apatite³, or even the definition is misleading, not reflecting the feldspars stoichiometry at all (Frost et al. 2001)⁴.



Exercise 2.3: Calculating simple indexes

On the Sázava dataset we can demonstrate how to define a function calculating a geochemical index. In this way the system can be enriched, quickly and efficiently.

a) Given the molecular weights below, design a function to calculate *mg* number.

FeO	MgO	Al ₂ O ₃	CaO	Na ₂ O	K ₂ O
71.85	40.31	101.96	56.08	61.98	94.20

b) Write a function returning Shand's indexes (A/CNK and A/NK).

c) Calculate all these values for the Sázava dataset.

d) Recast the major-element oxides on 100% volatile-free basis.



sazava.data



```
> sazava <- read.table("sazava.data", sep="\t")
> MW <- c(71.85, 40.31, 101.96, 56.08, 61.98, 94.20)
> oxides <- c("FeO", "MgO", "Al2O3", "CaO", "Na2O", "K2O")
> names(MW) <- oxides
> # Transpose as the division of a matrix by a vector
> # proceeds along columns, not rows.
> mol <- t(sazava[, oxides]) / MW[oxides]

a) > mgno <- function() {
>   mg <- 100 * mol["MgO",] / (mol["FeO",] + mol["MgO",])
>   return(mg)
> }

b) > ank <- function() {
>   ANK <- mol["Al2O3",] / (mol["Na2O",] + mol["K2O",])
>   return(ANK)
> }
> acnk <- function() {
>   ACNK <- mol["Al2O3",] / (mol["Na2O",] + mol["K2O",] +
+     mol["CaO",])
>   return(ACNK)
> }

c) > # Calculate the indexes
> x <- cbind(mgno(), acnk(), ank())
> colnames(x) <- c("mg.no", "A/CNK", "A/NK")
```

³ NB that *GCDkit* does not perform this correction.

⁴ See *GCDkit* help to the function `Frost`.

```
> x
      mg.no      A/CNK      A/NK
Sa-1  51.16987  0.8355806  2.396569
Sa-2  55.42955  0.7619109  2.307463
Sa-3  51.92059  0.8079150  2.562820 ...

d) > major <- c("SiO2", "TiO2", "Al2O3", "Fe2O3", "FeO", "MnO",
+              "MgO", "CaO", "Na2O", "K2O", "P2O5")
> sums <- apply(sazava[,major], 1, sum)
> anh <- sazava[,major]/sums*100
> anh
```

	SiO2	TiO2	Al2O3	Fe2O3	FeO
Sa-1	60.30565	0.6334205	16.50915	1.3573296	5.489644
Sa-2	56.25000	0.7238989	17.33279	2.7120718	5.362969
Sa-3	56.03133	0.7628153	17.89056	2.1663954	5.909276



Calculation of molecular weights in *GCDkit* utilises the function `molecularWeight`. It returns also the number of cations and oxygens per formula and thus the result needs to be subset, e.g. as follows:

```
GCDkit-> molecularWeight("Al2O3")[1]
      MW
101.9613
```



Simple geochemical indexes in *GCDkit*

Upon loading new data, several useful petrological indexes are calculated automatically, including *FeOt*, *mg#* and *Mg#*, as well as *A/CNK* and *A/NK* values that are then available, e.g. for plotting. Moreover, a matrix *WRanh* contains the major-element oxides recast to 100% anhydrous basis.

Therefore, the exercise a–c has a simple *GCDkit* solution:

```
GCDkit-> loadData("sazava.data")
GCDkit-> WR[,c("mg#", "A/CNK", "A/NK")]
GCDkit-> WRanh
```

2.2.2 Cationic Parameters

Niggli (1948) stressed the importance of simple cationic values for petrogenetic interpretation of igneous rocks. Several of the Niggli's cationic values, *si*, *al*, *fm*, *c*, *alk*, *k*, *mg*, *ti*, *p*, *c/fm*, and *qz* are still in use. The concept was further elaborated in multicationic parameters of the French authors, based on millications:

$$milli_{\alpha} = n_{\alpha} \times \frac{C_{\alpha}}{MW_{\alpha}} \times 1000 \quad (2.6)$$

where MW_{α} is molecular weight and n_{α} number of atoms in the oxide formula (e.g., the later is 1 for CaO, 2 for Na₂O, and again 2 for Al₂O₃). Note that even if the original

analyses (wt. % oxides) did sum up to 100, there is no reason for the total millications to attain a specific value. For some applications (e.g. zircon saturation, Sect. 13.1.1) it is required to normalize this total to 1.

For classification purposes, De La Roche et al. (1980) used a projection of two parameters, $R_1 = 4Si - 11(Na + K) - 2(Fe + Ti)$ and $R_2 = 6Ca + 2Mg + Al$, thus incorporating all major-element oxides. Besides that, the R_1 – R_2 plot has petrogenetic and geotectonic implications (Batchelor and Bowden 1985).

In the complex classification system of Debon and Le Fort (1983, 1988), arguably the most useful parameters are $A = Al - (K + Na + 2Ca)$ (reflecting peraluminosity), $B = Fe + Mg + Ti$ (maficity), $P = K - (Na + Ca)$ (proportion of K-feldspar among feldspars) and $Q = Si/3 - (K + Na + 2Ca/3)$ (quartz content).



Millications and related classification schemes

Upon loading a new data into *GCDkit*, the analyses are all recalculated to millications and stored in a data matrix `milli`. There are also functions calculating millications-based indexes, as well as generating some of the related plots (De La Roche et al. 1980; Debon and Le Fort 1983, 1988; Batchelor and Bowden 1985; Villaseca et al. 1998). See help for `LaRocheCalc`, `LaRoche`, `DebonCalc`, `Debon`, `Batchelor` and `Villaseca` to find out more.

2.2.3 Normative Calculations and Classification of Igneous Rocks

The norms, even though introduced early in the history of igneous petrology, are not obsolete. For instance, the CIPW norm (designed by Cross et al. 1902) remains important part of the TAS classification of volcanic rocks, where it serves for distinguishing some rock types (Le Bas et al. 1986; Le Maitre 2002). The calculation involves a hierarchical list of rules that tend to be often ambiguous and giving, to a varied extent, different results (Hutchison 1974, 1975; Verma et al. 2002, 2003). Another shortcoming of the CIPW norm is that it does not include hydrous minerals and therefore yields phases often not matching modal mineralogy in the studied igneous rocks, especially acidic ones.



Calculating norms in *GCDkit*

Most of the *GCDkit*'s normative recalculation schemes have been adopted from its predecessor, NORMAN (Janoušek 2001). Available are modules for the CIPW norm, including the modification with Bt and Hbl (Hutchison 1974, 1975), Catanorm (Hutchison 1974 and references therein), and Improved Granite Mesonorm (Mielke and Winkler 1979). The curious reader can type `CIPW` (without brackets!) at the *GCDkit* prompt, and look at the code of the function. See help for `CIPW`, `CIPWhb`, `Catanorm` and `Mesonorm` for details.



Dealing with results of *GCDkit* calculations

The most recent values calculated are always stored in the variable `results`. The results (i.e., the namesake variable) can be copied to the clipboard, appended to the data (to the data matrix `WR`) or saved into a variety of formats from a menu that appears after right-clicking the *R-Console* window.

2.3 Statistics

Early in the interpretation of a newly acquired geochemical dataset it is handy to examine descriptive statistics for selected elements or oxides. R contains a plethora of statistical tools, either built in, or provided as additional modules (packages). At this stage, however, simple functions such as `mean`, `median`, `sd` (standard deviation) and `summary` (a statistical overview) suffice. Revealing are also simple graphical tools such as boxplots (box-and-whiskers plots; function `boxplot`) and histograms (`hist`). Scatter matrices (`pairs`) serve to spot potentially significant correlations. See Appendix B for syntax and further details; the specific problem of dealing with a more complex data set containing several groups of data (using factors) is dedicated to Section 2.4.



Statistics in *GCDkit*

The command line interface of the standard R environment on the one hand allows much control for experienced users but on the other tends to discourage many scientists, accustomed to menu-driven software. *GCDkit* builds on the diverse functionality of the R language by providing a graphical user interface (GUI) to at least some of the most commonly used statistical functions, including simple descriptive statistics, histograms, boxplots, strip plots, correlation diagrams as well as more sophisticated methods of multivariate statistics (such as hierarchical clustering and principal components analysis). This interface is accessible from the menu *Calculations/Statistics*. Nevertheless the *R-Console* is still available for standard commands.

The more sophisticated tools are beyond the scope of the current text and the interested reader is referred to R/S documentation or special publications (e.g., Chambers and Hastie 1992; Venables and Ripley 1999; Maindonald and Braun 2003; Reimann et al. 2008; van den Bogaard and Tolosana-Delgado 2013).



Exercise 2.4: Simple statistics

- Compute means for all columns (variables) in the file `sazava.data`.
- Display boxplot for strontium, and find out all the main statistical parameters characterizing its distribution (the range, median, number of observations and not determined cases...).
- Plot all the possible combinations of binary diagrams (a scatterplot matrix) for the following oxides: SiO_2 , MgO , CaO , Na_2O , K_2O , and P_2O_5 .



sazava.data



```
> sazava <- read.table("sazava.data", sep="\t")
> sazava <- sazava[, -(1:6)]
# or sazava[, 7:ncol(x)] to get solely the numeric data

a) > result <- apply(sazava, 2, mean, na.rm=TRUE)
> round(result, 2)
      SiO2      TiO2      Al2O3      FeO      Fe2O3      MnO
57.95     0.64     16.94     4.73     1.75     0.14 ...

b) > boxplot(sazava[, "Sr"], xlab="Sr", ylab="ppm")
> summary(sazava[, "Sr"])
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
278.0   392.5   430.0   443.0   537.5   599.0     2.0

c) > oxides <- c("SiO2", "MgO", "CaO", "Na2O", "K2O", "P2O5")
> pairs(sazava[, oxides])
```

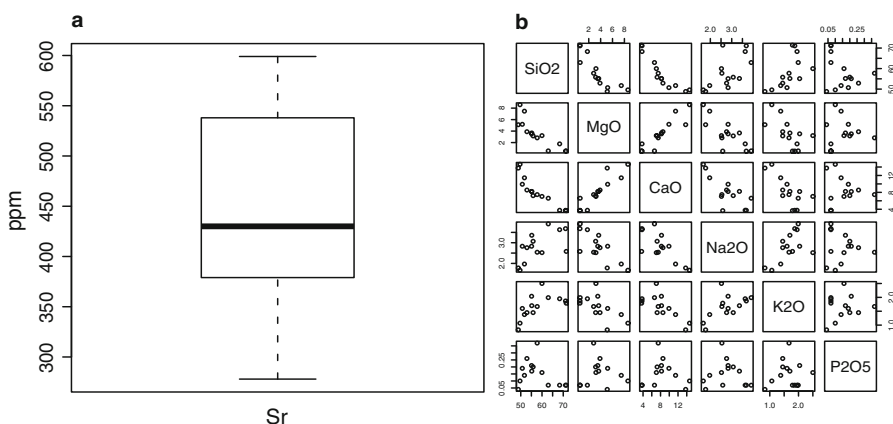


Fig. 2.1 **a** Boxplot of Sr distribution in the Sázava dataset (Exercise 2.4). **b** Scatterplot matrix for selected major-element oxides, plotted using the function `pairs`.

2.4 Classification and Grouping—Using Factors

Imagine that the studied plutonic complex consists of several igneous suites. Then whole-rock analysis for each sample can be accompanied by an indication as to which suite it belongs. A factor collecting this classification information enables, for instance, calculating an average A/CNK value for each of the suites separately. We should first demonstrate the definition of factors and then use them for increasingly difficult statistical and classification tasks.

2.4.1 Statistical Examination of Complex Data Sets

Statistical examination of complex geochemical data sets including, for instance, analyses for several intrusions, is tedious. Fortunately factors in R, in connection with the function `tapply`, offer a very flexible and elegant solution.



Exercise 2.5: Using factors to deal with complex datasets I

- For the Sázava dataset define a factor `intrusion` based on the specification given in the column 'Intrusion' that splits the suite into three groups: `basic` (quartz diorites to Amp gabbros of numerous smaller bodies), `Sazava` (Sázava intrusion proper: mainly Amp–Bt tonalites to quartz diorites) and `Pozary` (Požáry trondhjemite).
- Display all possible values (levels) of this factor.
- Using the factor `intrusion`, calculate the mean SiO_2 contents for each of the rock groups in the Sázava dataset.
- Analogously, calculate the mean concentrations of Ba.



`sazava.data`



```
> sazava <- read.table("sazava.data", sep="\t")

a) > intrusion <- factor(sazava[, "Intrusion"])
> intrusion
[1] Sazava Sazava Sazava Sazava Sazava basic basic basic
[9] basic basic Pozary Pozary Pozary Pozary
Levels: basic Pozary Sazava

b) > levels(intrusion)
[1] "basic" "Pozary" "Sazava"

c) > tapply(sazava[, "SiO2"], intrusion, mean)
basic Pozary Sazava
51.778 68.440 55.738

d) > tapply(sazava[, "Ba"], intrusion, mean)
basic Pozary Sazava
NA 1291.25 NA
```

In the last command, two of three groups gave NA because there are some missing values present:

```
> tapply(sazava[, "Ba"], intrusion, is.na)
$basic
[1] FALSE FALSE TRUE FALSE FALSE
$Pozary
[1] FALSE FALSE FALSE FALSE
```

```
$Sazava
[1] FALSE FALSE TRUE FALSE FALSE
```

If the missing values are to be ignored and the mean for the remaining analyses calculated, we can pass the parameter `na.rm=TRUE` to the function `mean`:

```
> tapply(sazava[, "Ba"], intrusion, mean, na.rm=TRUE)
      basic Pozary Sazava 
676.25 1291.25  682.25
```

The R language provides additional, arguably even more powerful tools. For instance, `aggregate` applies a given function to each of the variables (i.e., columns) of a numeric matrix or data frame \times respecting grouping (defined by a factor or list of factors). Analogous is the function `by`, which splits a data frame into several smaller ones based on a factor (or list of factors).



Exercise 2.6: Using factors to deal with complex datasets II

- Utilizing the function `summary`, calculate basic statistical parameters for SiO_2 distribution in each of the rock groups of the Sázava suite (factor `intrusion`).
- What are the means for selected trace elements (Ba, Rb, Sr and Zr) in individual intrusions?
- Using the function `by`, print basic statistical summaries for major-element oxides in each of the rock groups.



sazava.data



```
> sazava <- read.table("sazava.data", sep="\t")
> intrusion <- factor(sazava[, "Intrusion"])
```

```
a) > tapply(sazava[, "SiO2"], intrusion, summary)
$basic
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 48.84  49.63   51.72   51.78   52.90   55.80
$Pozary
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 62.95  66.96   69.69   68.44   71.17   71.42
$Sazava
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 50.72  55.09   55.17   55.74   57.73   59.98
```

```
b) > trace <- c("Rb", "Sr", "Ba", "Zr")
> aggregate(sazava[, trace], list(Rock=intrusion), mean,
+          na.rm=TRUE)
   Rock  Rb      Sr      Ba      Zr
1 basic 34.5 346.25 676.25  65.75
2 Pozary 59.5 460.75 1291.25 157.25
3 Sazava 60.5 522.00  682.25  61.00
```

```
c) > by(sazava[,7:17],list(Rock=intrusion),summary)
Rock: basic
      SiO2          TiO2          Al2O3
Min.   :48.84   Min.   :0.340   Min.   :13.34
1st Qu.:49.63   1st Qu.:0.670   1st Qu.:14.17 ... etc.
```



Groups in *GCDkit*

GCDkit allows each of the samples to be assigned to a group and these groups are subsequently utilised by statistical and plotting functions. Groups can be defined on the basis of a single label, a value of a numerical variable, position in classification diagram (e.g., TAS) or cluster analysis. In command line mode or a batch file it is simplest to use the *GCDkit* function `groupsByLabel`. The information regarding current grouping is stored in a vector `groups`; default grouping after loading a new data file or selecting a subset is on plotting symbol.

```
# GCDkit solution
GCDkit-> loadData("sazava.data")
GCDkit-> groupsByLabel("Intrusion")
      Sa-1      Sa-2      Sa-3      Sa-4      Sa-7      SaD-1
"Sazava" "Sazava" "Sazava" "Sazava" "Sazava"  "basic"...
Assigned groups:
  basic Pozary Sazava
    5      4      5
```



Statistics in complex datasets—the *GCDkit* way

Two functions provide basic statistical parameters for complex datasets with several groups, such as igneous suites (and optional plotting of histograms and/or boxplots): `summarySingleByGroup` (for a single variable) and `summaryByGroup` (several elements/oxides). If only the range of certain variable(s) is desired, use the function `summaryRangesByGroup`.

```
# Continuing from the previous example... (output is omitted)
GCDkit-> summarySingleByGroup("SiO2")

GCDkit-> trace <- c("Rb","Sr","Ba","Zr")
GCDkit-> summaryByGroup(trace)
GCDkit-> summaryRangesByGroup(trace)

GCDkit-> summaryByGroup(major)
```

2.4.2 Conversion of Numeric Vectors to Factors

The function `cut` splits a numeric vector `x` into given number of intervals and codes its individual items according to the rank they fall into. So this function can be used for simple classification purposes.



Exercise 2.7: Classification using factors

- Classify samples in the Sázava set according to SiO₂ contents (wt. %) in four groups, U (< 45), B (45–52), I (52–63) and A (> 63).



sazava.data



```
> sazava <- read.table("sazava.data", sep="\t")
> silica <- cut(sazava[, "SiO2"], breaks=c(0, 45, 52, 63, 100),
+   labels=c("U", "B", "I", "A"))
> silica
[1] I I I B I I B B B I I A A A
Levels: U B I A
```

Note that the levels that do not occur in the data at all (here the ultrabasic rocks, U) are not dropped. If we want to know the classification of individual samples, we convert the factor `silica` to a character vector:

```
> acidity <- as.vector(silica)
> names(acidity) <- rownames(sazava)
> acidity
Sa-1   Sa-2   Sa-3   Sa-4   Sa-7   SaD-1   Gbs-1   Gbs-20
  "I"    "I"    "I"    "B"    "I"    "I"    "B"    "B"
Gbs-2   Gbs-3   Po-1   Po-3   Po-4   Po-5
  "B"    "I"    "I"    "A"    "A"    "A"
```



Grouping according to a single numeric variable

Similar task, i.e. classification of samples into several groups according to values of a numeric variable, is done by *GCDkit* function `cutMy`.

```
# GCDkit solution
GCDkit-> loadData("sazava.data")
GCDkit-> cutMy("SiO2", c(0, 45, 52, 63, 100), c("U", "B", "I", "A"))
      SiO2 Interval
Sa-1   59.98      I
Sa-2   55.17      I
Sa-3   55.09      I
Sa-4   50.72      B
Sa-7   57.73      I ...
```

2.4.3 Frequency (Contingency) Tables

A nifty application of factors enables the creation of frequency tables.



Exercise 2.8: Frequency tables

Continuing from the previous exercise, we demonstrate making frequency tables.

- Using the factor `intrusion`, count the number of analyses obtained from each of the rock groups in the Sázava dataset.
- Analogously, count the number of ultrabasic, basic, intermediate and acid rocks (factor `silica` from the previous exercise).
- Set up a frequency table showing the dependence of `silica` on the rock type.



`sazava.data`



```
> intrusion <- factor(sazava[, "Intrusion"])
a) > table(intrusion)
```

```
intrusion
  basic Pozary Sazava
      5      4      5
```

```
b) > table(silica)
```

```
silica
  U B I A
0 4 7 3
```

```
c) > table(intrusion, silica)
```

```
      silica
intrusion U B I A
  basic   0 3 2 0
  Pozary  0 0 1 3
  Sazava  0 1 4 0
```

References

- Batchelor RA, Bowden P (1985) Petrogenetic interpretation of granitoid rock series using multicationic parameters. *Chem Geol* 48:43–55
- Chambers JM, Hastie TJ (1992) *Statistical models in S*. Chapman & Hall, London
- Clarke DB (1981) The mineralogy of peraluminous granites; a review. *Canad Mineral* 19:3–17
- Cross W, Iddings JP, Pirsson LV, Washington HS (1902) A quantitative chemico-mineralogical classification and nomenclature of igneous rocks. *J Geol* 10:555–690
- De La Roche H, Leterrier J, Grandclaude P, Marchal M (1980) A classification of volcanic and plutonic rocks using R_1R_2 -diagram and major element analyses—its relationships with current nomenclature. *Chem Geol* 29:183–210
- Debon F, Le Fort P (1983) A chemical-mineralogical classification of common plutonic rocks and associations. *Trans Roy Soc Edinb, Earth Sci* 73:135–149
- Debon F, Le Fort P (1988) A cationic classification of common plutonic rocks and their magmatic associations: principles, method, applications. *Bull Minéral* 111:493–510
- Frost BR, Frost CD (2008) A geochemical classification for feldspathic igneous rocks. *J Petrol* 49:1955–1969
- Frost BR, Barnes CG, Collins WJ, Arculus RJ, Ellis DJ, Frost CD (2001) A geochemical classification for granitic rocks. *J Petrol* 42:2033–2048

- Hutchison CS (1974) Laboratory handbook of petrographic techniques. John Wiley & Sons, New York
- Hutchison CS (1975) The norm, its variations, their calculation and relationships. *Schweiz mineral petrogr Mitt* 55:243–256
- Janoušek V (2001) Norman, a QuickBasic programme for petrochemical re-calculation of whole-rock major-element analyses on IBM PC. *J Czech Geol Soc* 46:9–13
- Janoušek V, Bowes DR, Rogers G, Farrow CM, Jelfinek E (2000) Modelling diverse processes in the petrogenesis of a composite batholith: the Central Bohemian Pluton, Central European Hercynides. *J Petrol* 41:511–543
- Janoušek V, Braithwaite CJR, Bowes DR, Gerdes A (2004) Magma-mixing in the genesis of Hercynian calc-alkaline granitoids: an integrated petrographic and geochemical study of the Sázava intrusion, Central Bohemian Pluton, Czech Republic. *Lithos* 78:67–99
- Le Bas MJ, Le Maitre RW, Streckeisen A, Zanettin B (1986) A chemical classification of volcanic rocks based on the total alkali-silica diagram. *J Petrol* 27:745–750
- Le Maitre RW (2002) Igneous rocks: a classification and glossary of terms: recommendations of the International Union of Geological Sciences, Subcommittee on the Systematics of Igneous Rocks. Cambridge University Press, Cambridge
- Maindonald J, Braun J (2003) Data analysis and graphics using R. Cambridge University Press, Cambridge
- Mielke P, Winkler HGF (1979) Eine bessere Berechnung der Mesonorm für granitische Gesteine. *Neu Jb Mineral, Mh* 471–480
- Miller CF (1985) Are strongly peraluminous magmas derived from pelitic sources? *J Geol* 93:673–689
- Niggli P (1948) Gesteine und Mineralagerstätten. Birkhäuser, Basel
- Reimann C, Filzmoser P, Garrett R, Dutter R (2008) Statistical data analysis explained: applied environmental statistics with R. John Wiley & Sons, Chichester
- Rock NMS (1988) Numerical geology. A source guide, glossary and selective bibliography to geological uses of computers and statistics. Lecture Notes in Earth Sciences, vol 18. Springer, Berlin
- Shand SJ (1943) Eruptive rocks. Their genesis, composition, classification, and their relation to ore-deposits with a chapter on meteorite. John Wiley & Sons, New York
- van den Bogaard P, Tolosana-Delgado R (2013) Analyzing compositional data with R. Springer, Berlin
- Venables WN, Ripley BD (1999) Modern applied statistics with S-Plus. Springer, Berlin
- Verma SP, Torres-Alvarado IS, Sotelo-Rodriguez ZT (2002) SINCLAS: standard igneous norm and volcanic rock classification system. *Comput and Geosci* 28:711–715
- Verma SP, Torres-Alvarado IS, Velasco-Tapia F (2003) A revised CIPW norm. *Schweiz mineral petrogr Mitt* 83:197–216
- Villaseca C, Barbero L, Herreros V (1998) A re-examination of the typology of peraluminous granite types in intracontinental orogenic belts. *Trans Roy Soc Edinb, Earth Sci* 89:113–119

Geochemical Modelling of Igneous Processes –

Principles And Recipes in R Language

Bringing the Power of R to a Geochemical Community

Janousek, V.; Moyen, J.-F.; Martin, H.; Erban, V.; Farrow,
C.

2016, XXVIII, 346 p. 332 illus., 86 illus. in color.,

Hardcover

ISBN: 978-3-662-46791-6