**2.1.1 Základy programovacího jazyka R (II.)**

## 2.1. Soubory

### 2.1.1 Vstup dat ze souborů

***read.table (file, header = FALSE, sep="", na.strings="NA", quote = "\"", dec=".")***

Funkce `read.table()` načte z disku soubor `filename`, který je rozdělený oddělovačem `sep` na jednotlivé položky. Separátorem mohou být mj.: „,“ – čárka, „\t“ – tabulátor, „\n“ – konec řádku (= new line).

Pokud R nenajde specifikovaný soubor, je pravděpodobně potřeba nastavit pracovní adresář příkazem *File|Change dir...*

Normálně se první řádek interpretuje jako jména sloupců a první položka na druhém a následujících řádcích jako jména řádek (`header` je nastaven na `TRUE`). Z toho vyplývá, že délka prvního řádku by měla být o jednu položku kratší, než všech následujících. Například:

```
    SiO2   TiO2   Al2O3  Fe2O3  FeO
Li1 51.73  1.48   16.01  1.03   7.06
Li2 51.88  1.48   15.93  0.99   6.85
```

*Poznámka: data načtená příkazem* read.table() *jsou uložena do proměnné typu data frame. Pokud ji chceme konvertovat na matici, použijeme funkce* `as.matrix()`:

```
> x<-read.table(filename,sep="\t")
> x<-as.matrix(x)
```

### 2.1.2 Ukládání datových souborů

***write.table(x, file = "", append = FALSE, quote = TRUE, sep = " ", eol = "\n", na = "NA", dec = ".", row.names = TRUE, col.names = TRUE***

Tato funkce uloží objekt `x` (vektor, matici, data frame) do souboru, přičemž jednotlivé položky jsou odděleny oddělovačem `sep`. Podobně jako pro `read.table`, lze specifikovat řetězce interpretované jako chybějící data a znak pro desetinnou tečku/čárku. Navíc existují logické parametry, řídící, zda mají být uložena také jména řádků nebo sloupců (`row.names,col.names`) a zda data mají být připojena k již existujícím nebo zda je mají přepsat (`append`).

***Cvičení 2.1***

*Soubor* sazava.data *obsahuje analýzy hlavních a stopových prvků vybraných granitoidních hornin sázavské suity středočeského plutonu. Tato data použijeme pro ukázku přístupu k datovým souborům a grafických možností jazyka R.*

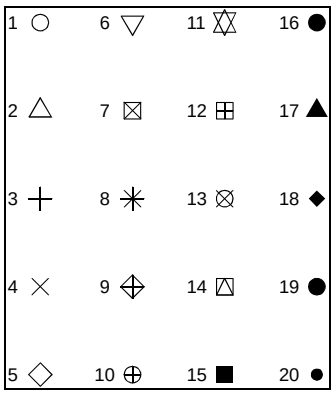
- načtěte do proměnné `WR` analýzy uložené v souboru `sazava.data`
- zobrazte v tabulce hodnoty $SiO_2$, MgO a $Na_2O/K_2O$
  (použijte funkce `cbind()`; nezapomeňte správně pojmenovat sloupce).

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

```
> x<-cbind(WR[,"SiO2"],WR[,"MgO"],WR[,"Na2O"]/WR[,"K2O"])
> colnames(x)<-c("SiO2","MgO","Na2O/K2O")
> rownames(x)<-rownames(WR)
> x
        SiO2  MgO Na2O/K2O
Sa-1   59.98 3.21 1.008000
Sa-2   55.17 3.67 1.976471
Sa-3   55.09 3.52 1.387255
… atd.
```

## 2.2. Grafické funkce

Jednou z velkých předností R je celá řada funkcí, určených pro vytváření a export grafů v prezentační kvalitě. Existují dva typy grafických funkcí, ty které otevírají nové grafické okno s úplně novým grafem **(high-level functions)** a ty, které jenom přidávají informaci nebo modifikují diagramy stávající **(low-level functions)**.




**Obr. 2.1.** Značky, které mohou být zobrazeny grafickými funkcemi R prostřednictvím parametru `pch`

***plot (y)***

Tato funkce vynese seřazené hodnoty vektoru `y`

***plot (x,y)***

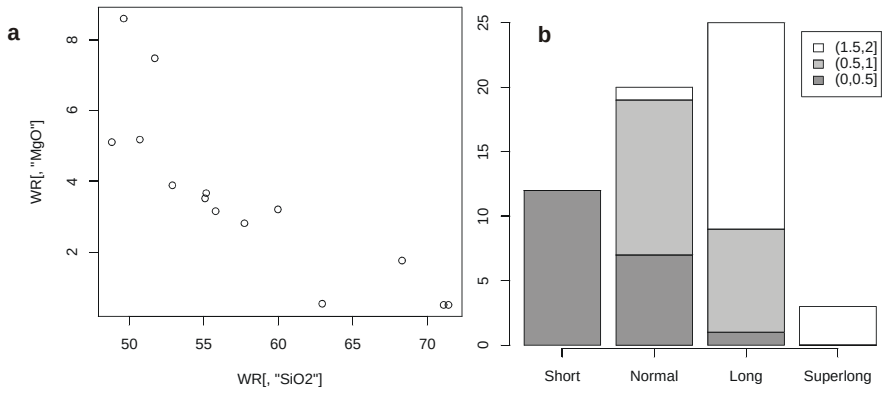
Pokud `x` a `y` jsou vektory, dostáváme binární xy graf (scatterplot):

```
> plot(WR[,"SiO2"],WR[,"MgO"]) # Obr. 2.2a
```

***plot (f); plot (f1, f2)***

Pokud `f` je faktor, `plot()` zobrazí sloupcový graf pro jeho jednotlivé hodnoty (*levels*). Pokud jsou uvedeny faktory dva, graf vypadá obdobně jako na Obr. 2.2b, získaném pro faktory z minulé lekce (soubor `ToothGrowth`):

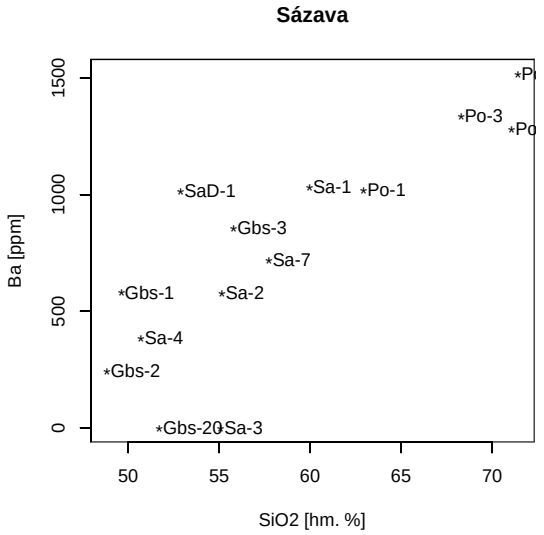
```
> plot(zuby,davka) # Obr. 2.2b
```




**Obr. 2.2.a** Graf $SiO_2$–MgO pro granitoidy sázavské suity; **b** Sloupcový graf ukazující závislost délky zubů morčat na dávce vitamínu C (mg)

Při volání funkce `plot()` může být uvedeno velké množství parametrů.
Mezi nejpoužívanější patří (s příklady):

| | |
|---|---|
| *xlab, ylab* | popis os x a y: `xlab="SiO2[wt. %]"` |
| *xlim, ylim* | rozsah os x a y: `xlim=c(50,70)` |
| *log* | která z os má být logaritmická? `log="xy"` |
| *pch* | kód pro značky (Obr. 2.1) nebo přímo symbol jako textový znak: `7, "+", "o", "Q"` |
| *col* | barva: `0, "black"`[1] |
| *type* | typ diagramu ("`p`" = body (points), "`l`" = čáry (lines), "`b`" = oboje (both), "`o`" = přepis (overplot), "`n`" = nic (no plotting)) |
| *main* | hlavní titulek diagramu (nahoře) |
| *sub* | podtitulek diagramu (dole) |
| *cex* | relativní velikost textu nebo symbolů (např. 2 = dvojnásobná) |
| *lty* | typ čáry (číslo 1–6, `1` = normální (solid), `2` = čárkovaná (dashed), `3` = tečkovaná (dotřes), `4` = čerchovaná (dotdash)) |
| *lwd* | relativní tloušťka čáry (např. 2 = dvojnásobná) |




**Obr. 2.3.** Diagram $SiO_2$–Ba pro granitoidy sázavské suity ukazující použití některých parametrů funkce `plot()` a funkce `text()` pro popis vynesených bodů

### *text (x, y, labels)*

Tento příkaz zobrazí daný text o souřadnicích `[x,y]`. Lze ho použít např. pro popis jednotlivých bodů xy grafů:

```
> plot(WR[,"SiO2"],WR[,"Ba"],
  xlab="SiO2[wt. %]",
  ylab="Ba[ppm]",
  pch="*",main="Sázava")
> text(WR[,"SiO2"]+0.2,WR[,"Ba"],
  rownames(WR),adj=0,col="red")
> # Obr. 2.3
```

Parametr `adj` udává způsob zarovnání textu (0 — vlevo, 1 — vpravo, 0.5 — na střed)

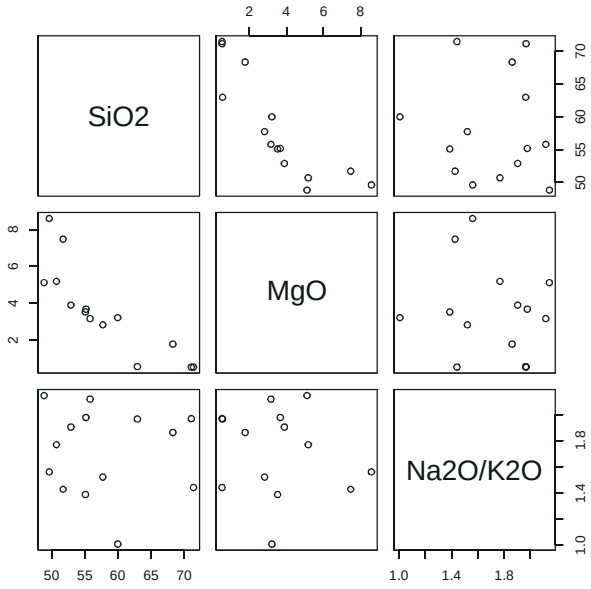
### *abline ()*

Příkaz `abline()` slouží pro vynášení přímek:

*abline (a ,b)*
sklon b, úsek vyťatý na ose y a

*abline(h=y)*
horizontální přímka

*abline (v=x)*
vertikální přímka




**Obr. 2.4.** Korelace mezi třemi proměnnými v souboru sazava.data jak je zobrazuje funkce `pairs()`

[1] Funkce `colors()` zobrazí všechna symbolická jména barev dostupných v R.

***hist (x)***

zobrazí histogram četností hodnot vektoru x

***pairs (x)***

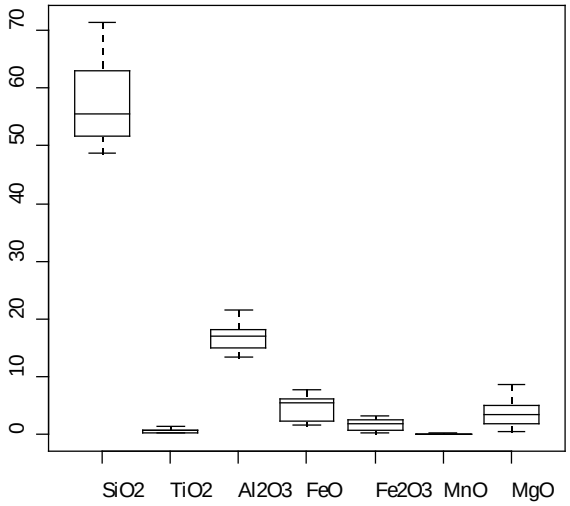
vykreslí binární diagramy pro všechny možné kombinace sloupců matice (dataframe) x:

```
> x<-cbind(WR[,"SiO2"],WR[,"MgO"],WR[,"Na2O"]/WR[,"K2O"])
> colnames(x)<-c("SiO2","MgO","Na2O/K2O")
> pairs(x) # Obr. 2.4
```

***boxplot (x)***

Diagram zvaný též "box-and-whiskers plot", v němž každý sloupec dataframu x (pro matici jsou všechna data vynesena dohromady do jednoho boxplotu) je zobrazen jako obdélník. Jeho vodorovné strany odpovídají dvěma kvartilům, zobrazena je dále vodorovná příčka (pro medián) a dvě svislé linie (spojující extrémy bez odlehlých hodnot, které bývají vyneseny zvlášť).

```
> boxplot(WR[,3:9]) # Obr. 2.5
```




**Obr. 2.5** Data pro některé oxidy hlavních prvků sázavské suity, jak je zobrazuje funkce `boxplot()`

Informace podobné funkci `boxplot()`v textové formě poskytne funkce `summary()`:

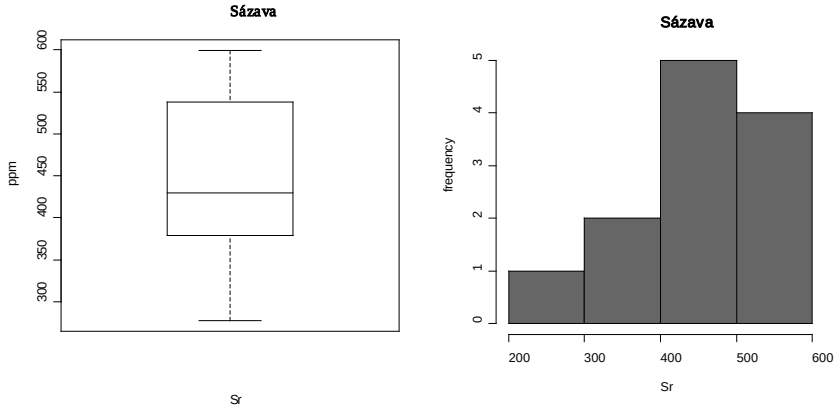
```
> summary(WR[,3:9])
```

```
      SiO2            TiO2            Al2O3            FeO        
 Min.   :48.84   Min.   :0.2800   Min.   :13.34   Min.   :1.650  
 1st Qu.:52.02   1st Qu.:0.3100   1st Qu.:15.14   1st Qu.:2.493  
 Median :55.49   Median :0.6900   Median :16.99   Median :5.445  
 Mean   :57.95   Mean   :0.6393   Mean   :16.94   Mean   :4.731  
 3rd Qu.:62.21   3rd Qu.:0.7900   3rd Qu.:18.07   3rd Qu.:6.118  
 Max.   :71.42   Max.   :1.3500   Max.   :21.64   Max.   :7.650  
     Fe2O3             MnO              MgO        
 Min.   :0.3800   Min.   :0.0400   Min.   :0.520  
 1st Qu.:0.7525   1st Qu.:0.0750   1st Qu.:2.032  
 Median :1.8000   Median :0.1550   Median :3.365  
 Mean   :1.7460   Mean   :0.1379   Mean   :3.570  
 3rd Qu.:2.6050   3rd Qu.:0.1675   3rd Qu.:4.805  
 Max.   :3.2800   Max.   :0.2500   Max.   :8.590
```

***Cvičení 2.2***

Nyní již máme k dispozici nástroje pro poměrně podrobnou grafickou analýzu souboru *sazava.data*. Podívejme se tedy v detailu na distribuci některých prvků.

- zobrazte všechny možné kombinace binárních diagramů následujících oxidů hlavních prvků: $SiO_2$, MgO, FeO, $Fe_2O_3$, CaO, $P_2O_5$
- vyneste binární diagram $SiO_2$–CaO, popište jeho osy a vynesené body. Zvolte vhodný rozsah pro osy `x` a `y`. Vytvořte dvě verze, nejprve vyneste všechny body jako modré čtverečky, potom přiřaďte symboly podle jednotlivých horninových typů (použijte pro to data ve sloupci `WR[,"Symbol"]`). Zobrazte linii $SiO_2$/CaO = 10 procházející počátkem
- vyneste boxplot pro distribuci stroncia a zjistěte příslušné základní statistické charakteristiky (rozsah, medián, počet ne-stanovených hodnot, …); zobrazte Sr data také ve formě histogramu
- vyneste diagram log(Zr)–log(Ba)




**Obr. 2.6.** Distribuce Sr v granitoidech sázavské suity (boxplot a histogram četností)

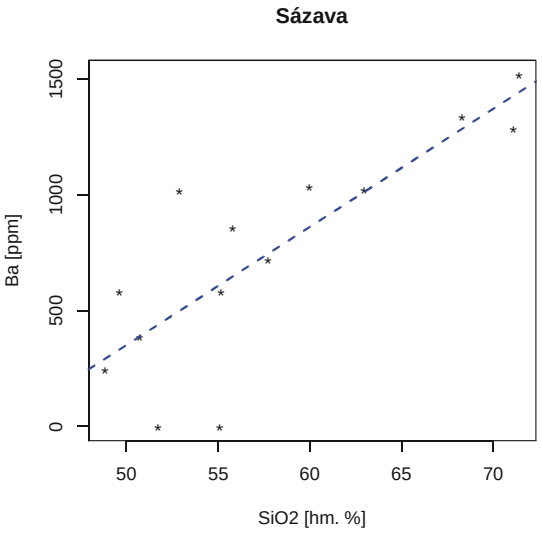
```
> WR<-read.table("sazava.data",sep="\t")
> WR<-as.matrix(WR[,-1])
> oxides<-c("SiO2","MgO","FeO","Fe2O3","CaO","P2O5")
> pairs(WR[,oxides])
> plot(WR[,"SiO2"],WR[,"CaO"],xlab="SiO2",ylab="CaO",pch=15,
  col="blue",xlim=c(49,75),ylim=c(0,15))
> text(WR[,"SiO2"]+0.3,WR[,"CaO"],rownames(WR),adj=0)
> plot(WR[,"SiO2"],WR[,"CaO"],xlab="SiO2",ylab="CaO",
  pch=WR[,"Symbol"],cex=1.5,xlim=c(49,75),ylim=c(0,15))
> abline(0,0.1)
> boxplot(WR[,"Sr"],main="Sázava",sub="Sr",ylab="ppm") # Obr. 2.6
> summary(WR[,"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
> mean(WR[,"Sr"],na.rm=T)
[1] 443
#  Pokud nespecifikujeme parametr na.rm, dostaneme NA – protože dvě koncentrace Sr
   nejsou stanoveny
> hist(WR[,"Sr"],main="Sázava",xlab="Sr",ylab="frequency",
  col="gray40",breaks=4)
> plot(WR[,"Zr"],WR[,"Ba"],xlab="Zr",ylab="Ba",pch=15,cex=1.5,
  log="xy")
```

### 2.2.1 Lineární regrese

V R jsou buď přímo zabudovány, anebo poskytovány ve formě přídavných balíčků (*packages*), mnohé statistické funkce. Většina jde mimo rámec tohoto kurzu a zájemce by se měl obrátit na specializované učebnice jazyků R/S (e.g., Venables & Ripley 1999). Po vytváření lineárních modelů slouží funkce `lm`:

***lm (model)***

Pokud je model specifikován jako "`y~x`", funkce `lm` provede lineární regresi. Například data v $SiO_2$–Ba diagramu pro sázavskou suitu mohou být proložena přímkou:




**Obr. 2.7** Diagram $SiO_2$–Ba sázavské suity proložený funkcí `lm(Ba~SiO2,data=WR)`

```
> plot(WR[,"SiO2"],WR[,"Ba"],xlab=
  expression(SiO[2]),ylab="Ba",
  pch="*",main="Sázava",cex=1.5)
> lq<-lm(WR[,"Ba"]~WR[,"SiO2"])
> lq<-lm(Ba~ SiO2,data=WR)
  # alternativa pro dataframes
> lq
```

```
Call:
lm(formula = WR[, "Ba"] ~ WR[, "SiO2"])

Coefficients:
 (Intercept)  WR[, "SiO2"]
   -1680.45         43.67
```

Pokud chceme vědět detaily, celý seznam (list) *lq* může být zobrazen funkcí `summary()`:

```
> summary(lq)
```

přičemž objekt `lq` je připraven k vynesení funkcí `abline`.

```
> abline(lq,lty=2,lwd=2)# Obr. 2.7
```

***Cvičení 2.3***

- vytvořte nový objekt `x`, jenž bude obsahovat pouze vzorky, jejichž $SiO_2 > 55$
- vyneste diagram $SiO_2$–CaO pro tyto vzorky a proložte data přímkou

```
> x<-WR[WR[,"SiO2"]>55,]
> plot(x[,"SiO2"],x[,"CaO"],xlab=expression(SiO[2]),ylab="CaO",
  pch=15)
> lq<-lm(x[,"CaO"]~ x[,"SiO2"])
> lq
```

```
Call:
lm(formula = CaO ~ SiO2, data = x)

Coefficients:
(Intercept)         SiO2
    23.4522      -0.2782
> abline(lq,lty=2)
```

### 2.2.2 Interakce s grafickými objekty

Velmi praktickou možností v R je interakce s grafy. Ta například umožňuje interaktivní výběr odlehlých hodnot a označit je názvem vzorku. Nebo lze vybrat určité vzorky jako koncové členy pro petrogenetické modelování.

***locator ()***

Funkce `locator()` vrací souřadnice bodů, na něž klikneme levým knoflíkem myši. Ukončení identifikace je pravým tlačítkem.

***identify(x, y, labels)***

Tato funkce slouží k interaktivní identifikaci bodů vynesených v diagramech. Typické použití ilustruje následující příklad, v němž uživatel popíše pouze body, které si sám zvolí (levým tlačítkem, pravým končí).

```
> plot(WR[,"SiO2"],WR[,"CaO"],xlab=expression(SiO2),ylab="CaO",
  col="darkred")
> identify(WR[,"SiO2"],WR[,"CaO"],rownames(WR))
```

### 2.2.3 Grafická zařízení

Fundamentálním problémem je, jak exportovat grafy z R do textového procesoru, případně grafického programu jako je CorelDraw! k dalším úpravám. Již existující graf lze kopírovat do schránky nebo uložit do souboru kliknutím pravým tlačítkem myši s volbou *Save as.* Na výběr je celá řada formátů, včetně populárních vektorových (Windows Metafile – WMF, PostScript[2]) a bitmapových (PNG, BMP, JPG). V nových verzích R je také k dispozici možnost výstupu do Adobe PDF. Alternativně lze graf uložit příslušnými položkami menu *File.*

Grafický výstup může být přesměrován na různá grafická zařízení, pod Microsoft Windows se to nejdůležitější jmenuje:

***windows()***

odstartuje nové grafické okno

***par (mfrow = c(nrow, ncol)); par(mfcol = c(nrow, ncol))***

rozdělí okno pro `nrow × ncol` dílčí grafy (zaplňovat se budou postupně po řádcích resp. po sloupcích)

***postscript(filename)***

grafický výstup bude uložen v PostScriptu do souboru `filename.` Kvalita je mnohem lepší, než při exportu do formátu Windows Metafile (WMF).

***graphics.off()***

zavře všechna grafická okna

[2] PostScript je jazyk, který slouží pro profesionální tisk z nejrůznějších programů a operačních systémů. Navíc ho lze bez problémů importovat do vyšších verzí CorelDraw! pro případnou další editaci.