# M6120 – 2. cvičení : **M6120cv02** (*Práce s daty v R*)

## A. Načtení vstupních dat:

### I. Čtení dat pomocí příkazu `scan`

Příklad 1

U náhodně vybraných 10ti-letých chlapců a dívek byly změřeny hodnoty IQ:

chlapci 113, 115, 103, 80, 92, 109, 109, 128, 117, 88, 103, 100
dívky 94, 101, 109, 116, 128, 100, 75, 75, 123, 82, 123, 94, 92

Nejprve zkopírujeme do schránky data, která se týkají chlapců. Pomocí funkce `scan` načteme data ze schránky do proměnné `chlapci`.

```
> chlapci <- scan(file = "clipboard", sep = ",")
```

Obdobně zkopírujeme do schránky data, která se týkají dívek a opět použijeme funkci `scan`.

```
> divky <- scan(file = "clipboard", sep = ",")
```

Z proměnných `chlapci` a `divky` vytvoříme datový rámec (datovou tabulku, datový soubor) pomocí příkazu `data.frame` a vypíšeme jeho obsah.

```
> dataIQ <- data.frame(IQ = c(divky, chlapci), sex = c(rep(0, length(divky)),
    rep(1, length(chlapci))))
> dataIQ
```

```
    IQ sex
1   94   0
2  101   0
3  109   0
4  116   0
5  128   0
6  100   0
7   75   0
8   75   0
9  123   0
10  82   0
11 123   0
12  94   0
13  92   0
14 113   1
15 115   1
16 103   1
17  80   1
18  92   1
19 109   1
20 109   1
21 128   1
22 117   1
23  88   1
24 103   1
25 100   1
```

## II. Čtení dat pomocí příkazu `read.table`

PŘÍKLAD 2

Načteme datový soubor pomocí příkazu `read.table`. Vzhledem k tomu, že v prvním řádku obsahuje datový soubor názvy proměnných, v přikazu `read.table` nesmíme zapomenout nastavit `header=TRUE`. Příkazem `str` vypíšeme strukturu datového rámce, příkazem `header` se vypíše prvních šest řádků.

```
> fileDat <- paste(data.library, "PorodniHmotnostDelka.dat", sep = "")
> data <- read.table(fileDat, header = TRUE)
> str(data)

'data.frame':        99 obs. of  2 variables:
 $ porodni_hmotnost: int  3200 3720 4100 3960 3700 3700 2860 3420 3820 2900 ...
 $ porodni_delka   : int  50 52 53 53 52 51 50 51 52 50 ...

> head(data)

  porodni_hmotnost porodni_delka
1             3200            50
2             3720            52
3             4100            53
4             3960            53
5             3700            52
6             3700            51
```

## III. Čtení dat pomocí příkazu `readLines`

PŘÍKLAD 3

Při zpracování jednotlivých dat budeme často mít k dispozici dva soubory: v jednom souboru bude jeho popis, ve druhém samotná data. Díky tomu, že první soubor bude obsahovat pouze text, tj. proměnné typu charakter, k jeho načtení budeme používat příkaz `readLines`. Protože je příkaz v závorkách, ihned se zobrazí obsah proměnné `popis`.

```
> fileTxt <- paste(data.library, "pryz.inf", sep = "")
> con <- file(fileTxt)
> (popis <- readLines(con))

 [1] "Zavislost odolnosti pryze v oderu na jejim slozeni"
 [2] "================================================="
 [3] "Byl studovan vztah mezi odolnosti pryze v oderu \"y\" na obsahu"
 [4] "kremikoveho plniva \"X1\" a lepici substance \"X2\". Plnivo"
 [5] "zvysuje odolnost v oderu a lepici substance chemicky poji castecky"
 [6] "plniva k pryzi a zvysuje tak jejich ucinnost."
 [7] "Odhadnete parametry beta_1, beta_2 a beta_3 linearniho regresniho"
 [8] "modelu."
 [9] "[1] Y  odolnost pryze v oderu"
[10] "[2] X1 obsah kremikoveho plniva"
[11] "[3] X2 obsah lepici substance"

> close(con)
```

Načteme i datový soubor, který nemá v prvním řádku názvy proměnných. Příkazem `str` vypíšeme jeho strukturu a uvedením jména datového rámce také i jeho obsah.

```
> fileDat <- paste(data.library, "pryz.txt", sep = "")
> data.pryz <- read.table(fileDat, header = FALSE)
> str(data.pryz)
```

```
'data.frame':        11 obs. of  3 variables:
 $ V1: int  83 113 92 82 100 96 98 95 80 100 ...
 $ V2: num  1 1 -1 -1 0 0 0 0 0 1.5 ...
 $ V3: num  -1 1 1 -1 0 0 0 1.5 -1.5 0 ...
```

```
> data.pryz
```

```
    V1   V2   V3
1   83  1.0 -1.0
2  113  1.0  1.0
3   92 -1.0  1.0
4   82 -1.0 -1.0
5  100  0.0  0.0
6   96  0.0  0.0
7   98  0.0  0.0
8   95  0.0  1.5
9   80  0.0 -1.5
10 100  1.5  0.0
11  92 -1.5  0.0
```

# B. Úprava datových souborů:

## I. Přejmenování sloupců

Příklad 4

V minulém příkladu jsme načetli datový soubor, který neměl v prvním řádku jména proměnných. Jazyk R automaticky proměnné po řadě nazval jmény *V1*, *V2*, *V3*. Chceme–li jiná jména, použijeme příkaz `names` (popř. `colnames`).

```
> names(data.pryz) <- c("odolnost_pryze", "obsah_kremikoveho_plniva",
    "obsah_lepici_substance")
> data.pryz
```

```
   odolnost_pryze obsah_kremikoveho_plniva obsah_lepici_substance
1              83                     1.0                   -1.0
2             113                     1.0                    1.0
3              92                    -1.0                    1.0
4              82                    -1.0                   -1.0
5             100                     0.0                    0.0
6              96                     0.0                    0.0
7              98                     0.0                    0.0
8              95                     0.0                    1.5
9              80                     0.0                   -1.5
10            100                     1.5                    0.0
11             92                    -1.5                    0.0
```

## II. Vytvoření proměnné typu faktor

Příklad 5

V prvním příkladu jsme vytvořili datový soubor `dataIQ`, ve kterém je druhá proměná, která nabývá hodnot 0 a 1. Z této proměnné vytvoříme faktor, jehož kategorie nazveme $F$ a $M$ ($F$ $(female) = 0$, $M$ $(male) = 1$):

```
> dataIQ$sex <- factor(dataIQ$sex, labels = c("F", "M"))
> head(dataIQ)
```

```
   IQ sex
1  94   F
2 101   F
3 109   F
4 116   F
5 128   F
6 100   F
```

```
> tail(dataIQ)
```

```
    IQ sex
20 109   M
21 128   M
22 117   M
23  88   M
24 103   M
25 100   M
```

## III. Spojování souborů

Příklad 5

Předpokládejme, že máme k dispozici dva datové soubory, první (`lifeforms.txt`) charakterizuje rostlinu a druhý (`fltimes.txt`) informuje o době kvetení.

Poznamenejme, že rostlina má ve jménu nejdříve rodové pojmenování (*genus*), potom druhové jméno (*species*), za ním případně následuje *subspécie* (*subsp.*), případně *varieta* (*var.*). Teprve pak, obvykle ale jen v literatuře, ne na jmenovkách, jsou údaje o jménu botanika.

Oba dva soubory nejprve načteme. Pokud je příkaz v kulatých závorkách, vypíše se obsah proměnné.

```
> fileDat <- paste(data.library, "lifeforms.txt", sep = "")
> (lifeforms <- read.table(fileDat, header = TRUE))
```

```
   Genus     species lifeform
1   Acer platanoides     tree
2   Acer    palmatum     tree
3  Ajuga     reptans     herb
4 Conyza sumatrensis   annual
5 Lamium       album     herb
```

```
> fileDat <- paste(data.library, "fltimes.txt", sep = "")
> (flowering <- read.table(fileDat, header = TRUE))
```

```
      Genus       species flowering
1      Acer   platanoides       May
2     Ajuga       reptans      June
3  Brassica         napus     April
4 Chamerion angustifolium      July
5    Conyza     bilbaoana    August
6    Lamium         album   January
```

Pokud budeme chtít oba dva soubory spojit pomocí příkazu `merge` je třeba si uvědomit, že je třeba ošetřit situaci chybějících variant. Napíšeme–li

```
> (both1 <- merge(flowering, lifeforms))
```

```
   Genus     species flowering lifeform
1   Acer platanoides       May     tree
2  Ajuga     reptans      June     herb
3 Lamium       album   January     herb
```

vidíme, že výstupní soubor obsahuje pouze 3 řádky, kdežto pokud uvedeme

```
> (both2 <- merge(flowering, lifeforms, all = TRUE))
```

```
      Genus       species flowering lifeform
1      Acer   platanoides       May     tree
2      Acer      palmatum      <NA>     tree
3     Ajuga       reptans      June     herb
4  Brassica         napus     April     <NA>
5    Conyza     bilbaoana    August     <NA>
6    Conyza   sumatrensis      <NA>   annual
7 Chamerion angustifolium      July     <NA>
8    Lamium         album   January     herb
```

výsledný soubor obsahuje všechny varianty. Protože oba dva výchozí soubory měly stejné názvy proměnných, spojování souborů bylo velmi jednoduché.

Nyní načteme další soubor, který obsahuje informace o váze semen

```
> fileDat <- paste(data.library, "seedwts.txt", sep = "")
> (seeds <- read.table(fileDat, header = TRUE))
```

```
      name1         name2 seed
1      Acer   platanoides 32.0
2    Lamium         album 12.0
3     Ajuga       reptans  4.0
4 Chamerion angustifolium  1.5
5    Conyza     bilbaoana  0.5
6  Brassica         napus  7.0
7      Acer      palmatum 21.0
8    Conyza   sumatrensis  0.6
```

V případě, kdy jména proměnných neodpovídají, chceme–li k souboru `both2` připojit informaci o váze semene, musíme psát

```
> (AllInf <- merge(both2, seeds, by.x = c("Genus", "species"), by.y = c("name1",
    "name2")))
```

```
      Genus       species flowering lifeform seed
1      Acer      palmatum      <NA>     tree 21.0
2      Acer   platanoides       May     tree 32.0
3     Ajuga       reptans      June     herb  4.0
4  Brassica         napus     April     <NA>  7.0
5    Conyza     bilbaoana    August     <NA>  0.5
6    Conyza   sumatrensis      <NA>   annual  0.6
7 Chamerion angustifolium      July     <NA>  1.5
8    Lamium         album   January     herb 12.0
```

Všimněme si ještě, že jména proměných výstupního souboru odpovídají jménům souboru `both2`.

## IV. Třídění souborů

Příklad 6

Načteme následující soubor

```
> fileDat <- paste(data.library, "worms.txt", sep = "")
> (worms <- read.table(fileDat, header = TRUE))
```

```
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
8            Ashurst  2.1     0     Arable     4.8 FALSE            4
9        The.Orchard  1.9     0    Orchard     5.7 FALSE            9
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
11       Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6
16      Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
17         Cheapside  2.2     8      Scrub     4.7  TRUE            4
18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5
19        Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
20         Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
```

Chceme-li soubor setřídit (vzestupně) podle jedné proměnné, např. sklonu, napíšeme

```
> worms[order(worms$Slope), ]
```

```
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
8            Ashurst  2.1     0     Arable     4.8 FALSE            4
9        The.Orchard  1.9     0    Orchard     5.7 FALSE            9
15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6
16      Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
19        Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
17         Cheapside  2.2     8      Scrub     4.7  TRUE            4
11       Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
20         Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
```

Chceme-li soubor setřídit sestupně podle jiné proměnné, např. plochy, píšeme

```
> worms[rev(order(worms$Area)), ]
```

```
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5
15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6
16      Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
19        Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
11       Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
17         Cheapside  2.2     8      Scrub     4.7  TRUE            4
8            Ashurst  2.1     0     Arable     4.8 FALSE            4
9        The.Orchard  1.9     0    Orchard     5.7 FALSE            9
14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
20         Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
```

Jestliže chceme soubor setřídit (vzestupně) podle více proměnných, např. uvnitř proměnné `Vegetation` podle proměnné `Soil.pH`, píšeme

```
> worms[order(worms$Vegetation, worms$Soil.pH), ]
```

```
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5
8            Ashurst  2.1     0     Arable     4.8 FALSE            4
2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
19        Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
16      Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6
9        The.Orchard  1.9     0    Orchard     5.7 FALSE            9
5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
17         Cheapside  2.2     8      Scrub     4.7  TRUE            4
20         Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
11       Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
```

## V. Přidání řádků a sloupců

PŘÍKLAD 7

Načteme soubor, ve kterém jsou pro jednotlivé osoby uvedeny prodeje za jaro, léto podzim a zimu.

```
> fileDat <- paste(data.library, "salesRB.txt", sep = "")
> (sales <- read.table(fileDat, header = TRUE))
```

```
             name spring summer autumn winter
1      Jane.Smith     14     18     11     12
2    Robert.Jones     17     18     10     13
3     Dick.Rogers     12     16      9     14
4 William.Edwards     15     14     11     10
5     Janet.Jones     11     17     11     16
```

Naším úkolem bude přidat novou proměnnou, která bude pro každého jednotlivce (tj. pro každý řádek) obsahovat odchylku jeho ročního průměru od celkového průměru. Nejprve spočítáme řádkové a sloupcové průměry.

```
> (RadekPrum <- rowMeans(sales[, 2:5]))
```

```
[1] 13.75 14.50 12.75 12.50 13.75
```

```
> (SloupecPrum <- colMeans(sales[, 2:5]))
```

```
spring summer autumn winter
  13.8   16.6   10.4   13.0
```

Spočítáme odchylky

```
> (devPoeple <- RadekPrum - mean(RadekPrum))
```

```
[1]  0.30  1.05 -0.70 -0.95  0.30
```

```
> (devSeasons <- SloupecPrum - mean(SloupecPrum))
```

```
spring summer autumn winter
  0.35   3.15  -3.05  -0.45
```

Přidáme sloupec, tj. novou proměnnou

```
> (sales1 <- cbind(sales, devPoeple))
```

```
             name spring summer autumn winter devPoeple
1      Jane.Smith     14     18     11     12      0.30
2    Robert.Jones     17     18     10     13      1.05
3     Dick.Rogers     12     16      9     14     -0.70
4 William.Edwards     15     14     11     10     -0.95
5     Janet.Jones     11     17     11     16      0.30
```

Na závěr přidáme ještě jeden řádek, kde do proměnné `name` dáme název proměnné, dále sezónní odchylky a do poslední proměnné, tj. do `devPoeple`, dosadíme nulu.

Přidáme sloupec, tj. novou proměnnou

```
> newRow <- sales1[1, ]
> newRow[1] <- "devSeasons"
> newRow[2:5] <- devSeasons
> newRow[6] <- 0
> (salesNew <- rbind(sales1, newRow))
```

```
             name spring summer autumn winter devPoeple
1      Jane.Smith  14.00  18.00  11.00  12.00      0.30
2    Robert.Jones  17.00  18.00  10.00  13.00      1.05
3     Dick.Rogers  12.00  16.00   9.00  14.00     -0.70
4 William.Edwards  15.00  14.00  11.00  10.00     -0.95
5     Janet.Jones  11.00  17.00  11.00  16.00      0.30
6      devSeasons   0.35   3.15  -3.05  -0.45      0.00
```

## C. Úkoly:

1. **Světová populace**
   Vytvořte datový soubor s proměnnými `country` a `population`, ve kterých bude název země a počet obyvatel (minimálně 25 zemí).
2. **Doplnění datového souboru**
   Předchozí datový soubor doplňte o proměnnou `continent`.
3. **Třídění dat**
   Setřiďte data nejprve podle počtu obyvatel, pak totéž, ale v rámci svědadílů.
4. **Výběr dat a nový soubor**
   Vytvořte nový datový soubor, ve kterém bude název kontinentu a počet obyvatel.