Tvorba datových objektů a manipulace s nimi

Vít Gabrhel

vit.gabrhel@mail.muni.cz

FSS MU,

26. 9. 2016

Harmonogram

0. Rekapitulace předchozí hodiny

1. Vector

2. Matrix

3. Factor

4. Data Frame

Co je to objekt?

Rekapitulace

Skupinu tvoří celkem tři přátelé, Vytvořte proto proměnnou "Přátelé" , která bude obsahovat hodnotu "3". Výsledek reportujte.

```
Pratele <- length(Pratele_Jmena)
Pratele
class(Pratele)


Pratele = 3
class(Pratele)

# Mojmír se stydí

                          Luděk > 0
is.numeric(Luděk)         Eleanora > 0
is.numeric(Eleanora)      Vladislav > 0
is.numeric(Vladislav)

                          c(Luděk, Eleanora,Vladi
as.logical(Luděk > 0)     slav) > 0
as.logical(Eleanora > 0)
as.logical(Vladislav > 0)
```

Vector

Vector je jednoduchý datový objekt o **různé délce** obsahující **hodnoty**

- c("Luděk", "Eleonora", "Vladislav")
- c(15, 13, 16)
- c(15, "Vladislav", "Eleonora", FALSE, 13)

Počet uchazečů dle oborů

- Uchazeči_FSS = c(35, 50, 150, 15)

Kdo je kdo aneb **pojmenování** vektorů

- names(Uchazeči_FSS) = c("POL", "BSS", "GEN", "PSY")

Výběr hodnot(y) z vektoru

- Uchazeči_FSS[c(1, 4)]
- Uchazeči_FSS[c("POL", "PSY")]

Vektorová **aritmetika**

Sčítání vektorů

- Uchazeči_FSS_Letos = c(35, 50, 150, 15)
- Uchazeči_FSS_Loni = c(70, 30, 140, 30)
- Uchazeči_Celkem = Uchazeči_FSS_Letos + Uchazeči_FSS_Loni

Součet hodnot ve vektoru

- Uchazeči_N <- sum(Uchazeči_Celkem)

Vector

Logické operátory

< for less than
> for greater than
<= for less than or equal to
>= for greater than or equal to
== for equal to each other
!= not equal to each other

Hlásilo se více uchazečů dle oborů více loni nebo letos?

- Uchazeči_FSS_Letos < Uchazeči_FSS_Loni

Hlásilo se více uchazečů dle oborů než bylo předpokládáno?

- Uchazeči_N > 400

Porovnání hodnot(y) mezi vektory

- Uchazeči_FSS_Loni[c(1, 4)] > Uchazeči_FSS_Letos[c(1, 4)]
- Uchazeči_FSS_Loni[c("POL", "PSY")] != Uchazeči_FSS_Letos[c("POL", "PSY")]
- FSS_GEN_Celkem <- Uchazeči_Celkem[c(3)] > 50

Factor

```
Známky = c("A", "C", "F", "B", "D", "F")
class(Známky)
```

Nominální kategorie

```
Factor_Známky = as.factor(Známky)
class(Factor_Známky)
levels(Factor_Známky) <- c("A", "B", "C", "D", "E", "F")
```

Ordinalizace

```
Factor_Známky <- factor(Známky, order = TRUE, levels = c("A", "B", "C", "D", "E", "F"))
```

Factor

Logické operátory

Tvorba vektoru

```
Modus_Známek_Letos = c("A", "C", "B", "D")
Modus_Známek_Loni = c("A", "B", "D", "A")
```

Faktorizace a ordinalizace

```
Modus_Známek_Loni_Factor <- factor(Modus_Známek_Loni, order =  TRUE, levels = c("A", "B", "C", "D", "E", "F"))

Modus_Známek_Letos_Factor <- factor(Modus_Známek_Letos,  order = TRUE, levels = c("A", "B", "C", "D", "E", "F"))
```

Srovnání vybraných hodnot mezi vektory

```
Modus_Známek_Loni_Factor[3] > Modus_Známek_Letos_Factor[3]
```

Matrix

In R, a **matrix** is

- a collection of elements of the same data type *(numeric, character, or logical)*
- arranged into a fixed number of **rows** and **columns**.
- Since you are only working with rows and columns, a matrix is called **two-dimensional**.

You can construct a matrix in R with the **matrix()** function. Consider the following example:

- matrix(1:9, byrow = TRUE, nrow = 3)

The **first argument** is the collection of elements that R will arrange into the rows and columns of the matrix. Here, we use 1:9 which is a shortcut for c(1, 2, 3, 4, 5, 6, 7, 8, 9).

The argument **byrow** indicates that the matrix is *filled by the rows*. If we want the matrix to be filled by the columns, we just place byrow = FALSE.

The third argument nrow indicates that the matrix should have **three rows**.

- Analogicky **"ncol"**

Matrix

Vedoucí katedry se rozhodne zjistit, jak si studenti stojí po 1. semestru studia v povinných předmětech.

Náhodně vylosuje Bořivoje, Jarmilu a Boženu s jejich body ze tří povinných předmětů (Úvodu, Kognice a Fyziologie). Jedno/víceoborový studenti

- Bořivoj = c(90, 70, 68)
- Jarmila = c(80, 100, 65)
- Božena = c(100, 80, 95)

Známky <- matrix(c(Bořivoj, Jarmila, Božena), nrow = 3, byrow = TRUE)

View(Známky**)**

Pojmenování řádků/sloupců

rownames(Známky) <- c("Bořivoj", "Jarmila", "Božena")
colnames(Známky) <- c("Úvod", "Kognice", "Fyziologie")

View(Známky)

Matrix

Jak je to u jedno- a dvouoborových studentů?

```
Známky_Subobory = c(80, 90, 70, 85, 75, 90)
Známky_Subobory_Matice = matrix(Známky_Subobory, nrow = 2, byrow = TRUE,
                                dimnames = list(c("Jednobor", "Dvouobor"), c("Úvod", "Kognice",
"Fyziologie")))

rowSums(Známky_Subobory_Matice)
```

Jak do matice přidat sloupec / řádek?

Skrze příkaz **cbind()** / **rbind()**

```
Obory = c(0, 1, 0)
Obory_Matrix = matrix(Obory)
rownames(Obory_Matrix) = c("Jednoobor", "Dvouobor", "Jednoobor")
colnames(Obory_Matrix) <- c("Jedno/Dvouobor")

Známky_Vše = cbind(Známky, Obory_Matrix)
Známky_Vše
```

Matrix

Jak příkazem zjistit aktivní objekty?

ls()

Jak vybrat konkrétní prvky z matice?

Similar to vectors, you can use the square brackets [ ] to select one or multiple elements from a matrix.

- Whereas vectors have one dimension, matrices have two dimensions. You should therefore use a comma to separate that what to select from the rows from that what you want to select from the columns. For example:
  - Známky_Vše[1,2] selects the element at the first row and second column.
  - Známky_Vše[1:3,2:4] results in a matrix with the data on the rows 1, 2, 3 and columns 2, 3, 4.
- If you want to select all elements of a row or a column, no number is needed before or after the comma, respectively:
  - Známky_Vše[,1] selects all elements of the first column.
  - Známky_Vše[1,] selects all elements of the first row.

Matrix

Jaký bodový průměr získali studenti průměrně z Fyziologie?

```
Body_Celkem_Fyziologie = Známky_Vše[,3]
mean(Body_Celkem_Fyziologie)
```

Jaký bodový průměr získal Bořivoj?

```
Body_Bořivoj = Známky[1,1:3]
mean(Body_Bořivoj)
```

Data Frame

Data Frame je **matice** tak, jak ji chápeme při analýze dat

- A data frame has the **variables** of a data set as **columns** and the **observations** as **rows**

V čem se v R "*Data Frame*" liší od "*Matrix*"?

- All the elements that you put in a matrix should be of the same type

Data Frame

Vyvolání Data Frame z R

data()

data(USArrests)

??(USArrests)

Jak se zorientovat v Data Frame?

- head() - show the first observations of a data frame
- tail() - prints out the last observations in your data set
- str() - struktura dat

Data Frame

Tvorba vlastní Data Frame

data.frame()

 # Planety - definování vektorů

- name <- c("Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptune")
- type <- c("Terrestrial planet", "Terrestrial planet", "Terrestrial planet","Terrestrial planet", "Gas giant", "Gas giant", "Gas giant", "Gas giant")
- diameter <- c(0.382, 0.949, 1, 0.532, 11.209, 9.449, 4.007, 3.883)
- rotation <- c(58.64, -243.02, 1, 1.03, 0.41, 0.43, -0.72, 0.67)
- rings <- c(FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, TRUE, TRUE)

Planety

- Planets = data.frame("name", "type", "diameter", "rotation", "rings")

Struktura dat

- str(Planets)
- head(Planets)
- tail(Planets)

Data Frame

Výběr prvků

- Planets[1:3,1]
- Planets[1:3,"name"]
- Planets$name
- Planets[rings, "name"]

Subsoubory

- subset(Planets, subset = rings)
- subset(Planets, subset = (diameter < 4))

Seřazování

- subset(Planets, subset = rings)
- subset(Planets, subset = (diameter < 4))

Zdroje

Cornelissen, J. (n.d.) Introduction to R. Dostupné online na: https://www.datacamp.com/courses/free-introduction-to-r

Cvičení

PSY232

/

PSY532