02. Datové objekty

Harmonogram

- 1. Vector
- 2. Factor
- 3. Matrix
- 4. Data Frame

Co je to objekt?

DEC. 9, 2015 AT 10:29 AM

A Complete Catalog Of Every Time Someone Cursed Or Bled Out In A Quentin Tarantino Movie

By Oliver Roeder

Filed under Word Count

Get the data on GitHub

RECOMMENDED

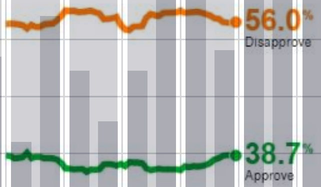
Students At Most Colleges Don't Pick 'Useless' Majors

The GOP Establishment Got What It Wanted (Sorta) In Alabama's Senate Primary

Al Gore's New Movie Exposes The Big Flaw In Online Movie Ratings

Trump Approval Ratings

UPDATED 15 HOURS AGO



Vector

Vector je <u>jednoduchý</u> datový <u>objekt</u> o **různé délce** obsahující **hodnoty**

- c("Reservoir Dogs", "Pulp Fiction", "Inglorious Basterds")
- c(421, 469, 51)
- c(421, "Reservoir Dogs", "death", FALSE, 10)

Počet cursing words dle filmů

- Words_Movie = c(421, 469, 57, 51)

Co je co aneb **pojmenování** vektorů

- names(Words_Movie) = c("Reservoir Dogs", "Pulp Fiction", "Kill Bill 1", "Inglorious Basterds")

Výběr hodnot(y) z vektoru

- Words_Movie[c(1, 4)]
- Words_Movie[c("Reservoir Dogs", "Inglorious Basterds")]

Vektorová **aritmetika**

Sčítání vektorů

- Hell = c(12, 5, 3, 4)
- Goddamn = c(10, 28, 7, 8)
- Spirituality = Hell + Goddamn

Součet hodnot ve vektoru

- Words_N <- sum(Spirituality)

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

Ve kterých filmech padlo více cursing words, než byl jejich průměrný počet za osm filmů?
- Words_Movie > 213

Zaznívalo ve filmech více slovo "Hell" nebo "Goddamn"?
- Hell < Goddamn

Porovnání hodnot(y) <u>mezi vektory</u>
- names(Hell) = c("Reservoir Dogs", "Pulp Fiction", "Kill Bill 1", "Inglorious Basterds")
- names(Goddamn) = c("Reservoir Dogs", "Pulp Fiction", "Kill Bill 1", "Inglorious Basterds")
- Hell[c(1, 4)] > Goddamn[c(1, 4)]
- Hell[c("Reservoir Dogs", "Inglorious Basterds")] != Goddamn[c("Reservoir Dogs", "Inglorious Basterds")]
- names(Spirituality) = c("Reservoir Dogs", "Pulp Fiction", "Kill Bill 1", "Inglorious Basterds")
- PulpFiction_Celkem <- Spirituality[c(2)] > 50

Factor

```
Filmy = c("Kill Bill 1", "Reservoir Dogs", "Inglorious Basterds", "Pulp Fiction")
class(Filmy)
```

Nominální kategorie

```
Factor_Filmy = as.factor(Filmy)
class(Factor_Filmy)
levels(Factor_Filmy) <- c("Reservoir Dogs", "Pulp Fiction", "Kill Bill 1", "Inglorious Basterds")
```

Ordinalizace

```
Factor_Filmy <- factor(Filmy, order = TRUE, levels = c("Reservoir Dogs", "Pulp Fiction", "Kill Bill
1", "Inglorious Basterds")
```

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

O cursing words v Tarantinových filmech už něco víme. Co ale počet mrtvých?

Budeme se věnovat Pulp Fiction, Inglorious Basterds a Django Unchained spolu s počtem zesnulých postav. Přidáme k tomu známý počet cursing words v příslušných filmech:

```
Pulp_Fiction = c(7, 469)
Inglorious_Basterds = c(48, 58)
Django_Unchained = c(47, 262)

Filmy <- matrix(c(Pulp Fiction, Inglorious Basterds, Django Unchained), nrow = 3,
byrow = TRUE)
view(Filmy)
```

Pojmenování řádků/sloupců

```
rownames(Filmy) <- c("Pulp_Fiction", "Inglorious_Basterds", "Django_Unchained")
colnames(Filmy) <- c("Deaths", "Words")
View(Filmy)
```

Matrix

```
Death_Curse = c(7, 48, 47, 469, 58, 262)
Death_Curse_Matrix = matrix(Death_Curse, nrow = 3, byrow = FALSE,
dimnames = list(c("Pulp_Fiction", "Inglorious_Basterds", "Django_Unchained")))

colnames(Death_Curse_Matrix) <- c("Deaths", "Curses")
colSums(Death_Curse_Matrix)
```

Jak do matice přidat sloupec / řádek?
Skrze příkaz **cbind() / rbind()**

Filmy si rozdělíme z hlediska období tvorby (90s, 00s a 10s) s kódy "0", "1" a "2":

```
Period = c(0, 1, 2)
Period_Matrix = matrix(Period)
rownames(Period_Matrix) = c("90s", "00s", "10s")
colnames(Period_Matrix) <- c("Period")

Death_Curse_Period = cbind(Death_Curse_Matrix, Period_Matrix)
Death_Curse_Period
```

Matrix

<u>Jak příkazem zjistit aktivní objekty?</u>
ls()

<u>Jak vybrat konkrétní prvky z matice?</u>
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:
    - Death_Curse_Period[1,2] selects the element at the first row and second column.
    - Death_Curse_Period[1:3,2:3] results in a matrix with the data on the rows 1, 2, 3 and columns 2 and 3.

If you want to select all elements of a row or a column, no number is needed before or after the comma, respectively:

- Death_Curse_Period[,1] selects all elements of the first column.
- Death_Curse_Period[1,] selects all elements of the first row.

Matrix

Jaký byl průměrný počet mrtvých ve sledovaných filmech?

```
Mean_Dead = Death_Curse_Period[,1]
mean(Mean_Dead)
```

Jaký je Tarantino index (tj. počet mrtvých na počet nadávek) pro Inglorious Basterds?

```
Dead_Curse = data.frame(Death_Curse_Period[2,1:2])
Dead_Curse[2,1]/Dead_Curse[1,1]
View(Dead_Curse[2,1])
```

I DON'T SEE THE CODE ANYMORE.
ALL I SEE IS DATA FRAMES, LISTS, AND OPTIONAL ARGUMENTS.

Data Frame

Data Frame je **matice** tak, <u>jak ji chápeme při analýze dat</u>

- 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)

View(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

RESPECT MY AUTHORITAH!

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)

NEVER FORGET PLUTO

(1930 -2006)

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

- order(Planets$diameter)
- Planets[order("diameter")]

Zdroje

Cornelissen, J. (n.d.) Introduction to R. Dostupné online na:

https://www.datacamp.com/courses/free-introduction-to-r