Useful basics and descriptive statistics

Andreas Alfons[1] Pieter Schoonees[2] Dennis Fok[1]

[1]Erasmus School of Economics, Erasmus Universiteit Rotterdam

[2]Rotterdam School of Management, Erasmus Universiteit Rotterdam

EQI Programming & Visualization for Business Analytics,
September 7, 2018

Software requirements

$\longrightarrow$ Data from package euR are used

```
R> library("euR")
```

Content

Basics

Vectors

Combine values into a vector with function `c()`:

```
R> c(2, 4, 5, 21, 65)
[1]  2  4  5 21 65
R> c("foo", "bar")
[1] "foo" "bar"
```

Sequences of values:

```
R> 3:7
[1] 3 4 5 6 7
R> seq(0, 1, by = 0.2)
[1] 0.0 0.2 0.4 0.6 0.8 1.0
```

Assigning values

Assign values to an object with <-:

```
R> numbers <- c(2, 4, 5, 21, 65)
R> numbers
[1]  2  4  5 21 65
R> text <- c("foo", "bar")
R> text
[1] "foo" "bar"
```

⟶ If values are assigned to an object, they are not printed

⟶ Print values by typing the name of the object

Special values

`NA`	Not available (represents missing value)
`NaN`	Not a number (usually result of division 0/0)
`Inf`	Positive infinity
`-Inf`	Negative infinity
`NULL`	Represents undefined value

Basic math

Operator or function	Operation	Example
`+`	addition	`x + y`
`-`	subtraction	`x - y`
	univariate minus	`-x`
`*`	multiplication	`x * y`
`/`	division	`x / y`
`^`	exponentiation	`x ^ y`
`abs()`	absolute value	`abs(x)`
`sqrt()`	square root	`sqrt(x)`
`log()`	logarithm	`log(x)`
`exp()`	exponential function	`exp(x)`

$\longrightarrow$ Vectorized arithmetic: operations are performed elementwise

Data manipulation

Data dimensions

Number of observations and columns together:

```
R> dim(PhDPublications)
[1] 915   6
```

Number of observations and columns separately:

```
R> nrow(PhDPublications)
[1] 915
R> ncol(PhDPublications)
[1] 6
```

Names

Variable names:

```
R> colnames(PhDPublications)
[1] "articles" "gender"   "married"  "kids"     "prestige"
[6] "mentor"
```

Row names:

```
R> rownames(PhDPublications)
```

Extracting a variable

Extract a variable from a data frame with $:

```
R> articles <- PhDPublications$articles
R> mentor <- PhDPublications$mentor
```

Length of a vector:

```
R> length(articles)
[1] 915
```

Categorizing a numeric vector

Use function `cut()` for categories between breakpoints:

```
R> prestige <- PhDPublications$prestige
R> b <- c(0, 2.5, 3.5, 5)
R> prcat <- cut(prestige, breaks = b)
```

The frequencies can be counted with function `table()`:

```
R> table(prcat)
prcat
  (0,2.5] (2.5,3.5]   (3.5,5]
      279       284       352
```

⟶ Categorization yields loss of information

⟶ Always keep the original variable

Categorization revisited

$\longrightarrow$ Use labels in the categorization:

```
R> b <- c(0, 2.5, 3.5, 5)
R> l <- c("low", "average", "high")
R> prcat <- cut(prestige, breaks = b, labels = l)
R> table(prcat)
prcat
    low average    high
    279     284     352
```

Descriptive statistics

Some useful statistical functions

Minimum and maximum:

```
R> min(articles)
[1] 0
R> max(articles)
[1] 19
```

Default quantiles:

```
R> quantile(articles)
  0%  25%  50%  75% 100%
   0    0    1    2   19
```

Mean and median:

```
R> mean(articles)
[1] 1.692896
R> median(articles)
[1] 1
```

Some useful statistical functions

Standard deviation and variance:

```
R> sd(articles)
[1] 1.926069
R> var(articles)
[1] 3.709742
```

Contingency tables

```
R> married <- PhDPublications$married
R> kids <- PhDPublications$kids
```

One-way contingency table:

```
R> table(kids)
kids
  0   1   2   3
599 195 105  16
```

Two-way contingency table:

```
R> table(married, kids)
       kids
married   0   1   2   3
    no  309   0   0   0
    yes 290 195 105  16
```

Accessing variables with `with()`

⟶ Variables of a data frame do not have to be accessed with $

⟶ Useful if computations require multiple variables

```
R> with(PhDPublications, cor(articles, mentor))
[1] 0.3058616
R> with(PhDPublications, table(married, kids))
       kids
married   0   1   2   3
    no  309   0   0   0
    yes 290 195 105  16
```