Introduction to R

FGCZ Protein Informatics Training

Witold Wolski wew@fgcz.ethz.ch

31 May, 2021

introduction to R

Where can I learn R?

- Online Courses :
  - www.edx.org; udemy.com; coursera.org; codeacademy.org
- Online books:
  - R Programming for Data Science
  - Advanced R
  - ...
- Introducing Variables, Types and Operators

Variables, Types and Operators

```
2 + 2
```

```
## [1] 4
```

```
a = 2; b = 3
a + 2
```

```
## [1] 4
```

```
a * b
```

```
## [1] 6
```

```
a == b
```

```
## [1] FALSE
```

```
l ← a == b
```

```
s ← 'a'
S ← "helloworld"
s ≠ S
```

```
## [1] TRUE
```

```
class(a); class(l); class(s); class(S)
```

```
## [1] "numeric"
```

```
## [1] "logical"
```

```
## [1] "character"
```

```
## [1] "character"
```

- a, b, s and S are variables
- R is case sensitive `s ≠ S`
- variables have `types`, e.g. `numeric`, `character`.
- Arithmetic Operators : `+`, `-`, `*`, `/`
- Relational Operators : `<`, `>`, `⩽`, `==`

Variables, Types and Operators

```
a = 1
class(a)
```

```
## [1] "numeric"
```

```
a = 'a'
class(a)
```

```
## [1] "character"
```

```
a == s
```

```
## [1] TRUE
```

```
?`==`
```

- Dynamic typing
- `?` getting help function, e.g. `?==`

- NA - not available

```
a ← 1;length(a);class(a)
```

```
## [1] 1
```

```
## [1] "numeric"
```

```
b ← numeric();length(b);class(b)
```

```
## [1] 0
```

```
## [1] "numeric"
```

```
a + b
```

```
## numeric(0)
```

```
c ← NA; class(c)
```

```
## [1] "logical"
```

Composed data types - arrays and matrices

```
av ← c(1,2,3,4)
bv ← 1:4
class(av); class(bv)
```

```
## [1] "numeric"
```

```
## [1] "integer"
```

```
av * 2
```

```
## [1] 2 4 6 8
```

```
length(av)
```

```
## [1] 4
```

```
av[1];av[2:4];av[-1]
```

```
## [1] 1
```

```
## [1] 2 3 4
```

```
m ← matrix(1:4, ncol = 2)
m
```

```
##      [,1] [,2]
## [1,]    1    3
## [2,]    2    4
```

```
m[1,]; m[,2]
```

```
## [1] 1 3
```

```
## [1] 3 4
```

```
class(m);class(m[1,]);class(m[1,1])
```

```
## [1] "matrix" "array"
```

```
## [1] "integer"
```

```
## [1] "integer"
```

Composed data types - lists

```
a ← c(a = 1:4, b = c('a','b'));a
```

```
##  a1  a2  a3  a4  b1  b2
## "1" "2" "3" "4" "a" "b"
```

```
l ← list(a = 1:4, b = c('a','b'));class(l) ;l
```

```
## [1] "list"

## $a
## [1] 1 2 3 4
##
## $b
## [1] "a" "b"
```

```
length(l)
```

```
## [1] 2
```

```
l$a;l$b; class(l$a)
```

```
## [1] 1 2 3 4

## [1] "a" "b"

## [1] "integer"
```

```
l[1];class(l[1]); length(l[1])
```

```
## $a
## [1] 1 2 3 4

## [1] "list"

## [1] 1
```

```
l[[1]];class(l[[1]]); length(l[[1]])
```

```
## [1] 1 2 3 4

## [1] "integer"
```

Composed data types - data.frames

```
df ← data.frame(a = 1:5,
                b = c("a","b","b","c","d"))
df;
```

```
##   a b
## 1 1 a
## 2 2 b
## 3 3 b
## 4 4 c
## 5 5 d
```

```
colnames(df)
```

```
## [1] "a" "b"
```

```
class(df);class(df[1]);class(df[[1]])
```

```
## [1] "data.frame"

## [1] "data.frame"

## [1] "integer"
```

```
df[1]
```

```
##   a
## 1 1
## 2 2
## 3 3
## 4 4
## 5 5
```

```
df$a
```

```
## [1] 1 2 3 4 5
```

```
df[,1]
```

```
## [1] 1 2 3 4 5
```

```
df[[1]]
```

```
## [1] 1 2 3 4 5
```

Composed data types - data.frames

```
?read.csv
?readr:::read_tsv
?readxl::read_xlsx
```

- csv, tsv, excel files are imported into R as `data.frames`
- many functions in base R, e.g. `subset`, `aggregate`
- readr & tidyr & dplyr

```
subset(df, b == "b")
```

```
##   a b
## 2 2 b
## 3 3 b
```

```
subset(df,a < 2)
```

```
##   a b
## 1 1 a
```

```
aggregate(df, by = list(c = df$b), paste0)
```

```
##   c    a    b
## 1 a    1    a
## 2 b 2, 3 b, b
## 3 c    4    c
## 4 d    5    d
```

```
aggregate(df, by = list(c = df$b), mean)
```

Functions and Scopes

```
rm(list=ls())
a ← 23

myfun ← function(x){
  a ← 3
  x ← x + 1
  return(x)
}

b ← myfun(a)
a == 23
```

```
## [1] TRUE
```

```
b
```

```
## [1] 24
```

```
myfun
```

```
## function(x){
##   a ← 3
##   x ← x + 1
##   return(x)
## }
```

```
?myfun
```

```
## No documentation for 'myfun' in specified packages and
## you could try '??myfun'
```

```
?print
```

Thank you for your attention