INWT Statistics

INWT Statistics GmbH
The Beauty of R Programming

Verena Pflieger @ RLadies | 17. April 2018

INWT Statistics GmbH

INWT Statistics specializes in intelligent data analysis and delivers solutions in the fields of online marketing, CRM, data management and business intelligence/reporting.

Online Marketing

- Customer Journey Analysis
- Conversion Optimization
- Fraud Detection
- TV Impact

CRM

- Customer Lifetime Value
- Customer Segmentation
- Churn Management
- 360° Brand

BI/Reporting

- Data Management
- Data Consolidation
- Dashboards

Selected Customers:

CHARITÉ UNIVERSITÄTSMEDIZIN BERLIN

BIOMET®

CosmosDirekt. Die Versicherung.

OTTO

SIXT rent a car

tipico

Webtrekk

zalando

Agenda

1 Functions

2 Scoping Rules

3 Control Structures

4 Some Members of the apply Family

Agenda

Copy and Paste

```
[...]

round(mean(x), 3)
round(max(x), 2)

round(mean(y), 3)
round(max(y), 2)

round(mean(z), 3)
round(max(x), 2)
```

A Function

```
mySummary <- function(a) {
  print(round(mean(a), 3))
  print(round(max(a), 2))
}

# Apply this function to vectors ..., x, y, and z:
[...]
mySummary(x)
mySummary(y)
mySummary(z)
```

The Beauty of the Function

- easier to maintain
- less error prone
- flexible
- fewer lines of code

Functions can be...

- passed as arguments to other functions,
- returned by other functions,
- stored in a list, and
- nested, so that a function can be defined inside of another function.

R Functions

```
someFunction <- function( <arguments> ) {
  <expression>
}
```

- body
- formals
- environment

Function Arguments

- exact,
- partial, and
- positional matching

The ... Argument

```
myplot <- function(x, y, type = "l", ...) {
  plot(x, y, type = type, ...)
}
```

Lazy evaluation

- arguments are evaluated only as needed
- functions create own environment

Environments

- binds set of names to values
- global environment / function environment

- reference semantics: modifying environment, modifys every copy
- have parents up to the empty environment
- objects in environments must have names

Functions and Environments

- temporal environment
- global environment is not modified
- self-containment: return depends only on input values

Agenda

Scoping Rules

Set of rules about how `R` finds objects.

- Lexical scoping: where
- Dynamic lookup: when

Name Masking

- Objects with same name
- search path ‘search()’
- package::fct_name

Packages

- write packages!

Agenda

If-else

```
ifelse(condition, yes, no)
```

```
if (condition) {
  ## do something
} else {
  ## do something else
}
```

More If-else

```
if (condition1) {
  ## do something
} else if (condition2) {
  ## do something different
} else {
  ## do something even more different
}
```

switch

```
mySwitchNum <- function(x) {
  switch(EXPR = x,
         "first alternative", "alternative number 2", "third alternative")
}
mySwitchNum(1)
```

```
## [1] "first alternative"
```

```
mySwitchNum(2)
```

```
## [1] "alternative number 2"
```

```
mySwitchNum(4)
```

Loops

- for
- while
- repeat

for

```r
x <- c("a", "b", "c", "d")

for (i in 1:4) {
  print(x[i])
}
```

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

while

```
number <- 0

while (number < 5) {
  print(number)
  number <- number + 1
}
```

```
## [1] 0
## [1] 1
## [1] 2
## [1] 3
## [1] 4
```

repeat

- the infinite loop!

```
x0 <- 1
tol <- 1e-8

repeat {
  x1 <- computeEstimate(x0)
  if (abs(x1 - x0) < tol) {
    break
  } else {
    x0 <- x1
  }
}
```

next

```
for (i in 1:100) {
  if (i <= 20) {
    ## Skip the first 20 iterations
    next
  }
  ## Do something here
}
```

Agenda

1 Functions

2 Scoping Rules

3 Control Structures

4 Some Members of the apply Family

apply

```
x <- matrix(rnorm(20), nrow = 5, ncol = 4)
x
```

```
##             [,1]       [,2]       [,3]        [,4]
## [1,]  0.9608811  0.3020303  0.1817257 -2.75762574
## [2,]  0.8602888  0.5106651  0.0692807  1.15305677
## [3,]  1.3842613 -1.4839198  0.4938446  0.09614295
## [4,] -1.2362002  0.4490325  1.8775407 -0.14237074
## [5,] -0.2639524 -0.5299821 -0.1606260  0.68157064
```

```
apply(x, 2, mean)
```

```
## [1]  0.3410557 -0.1504348  0.4923531 -0.1938452
```

lapply

```
x
```

```
## $a
## [1] 1 2 3 4 5
## $b
##  [1] -0.6264538  0.1836433 -0.8356286  1.5952808  0.3295078 -0.8204684
##  [7]  0.4874291  0.7383247  0.5757814 -0.3053884
```

```
lapply(x, mean)
```

```
## $a
## [1] 3
## $b
## [1] 0.1322028
```

mapply

```
students <- c("Peter", "Paul", "Mary", "Max", "Moritz")
grades <- c(3, 2, 1, 1, 5)
mapply(paste, students, grades, USE.NAMES = FALSE)
```

```
## [1] "Peter 3"  "Paul 2"   "Mary 1"   "Max 1"    "Moritz 5"
```

tapply

```
tapply(airquality$Temp,
       INDEX = airquality$Month,
       mean,
       na.rm = TRUE)
```

```
##        5        6        7        8        9
## 65.54839 79.10000 83.90323 83.96774 76.90000
```

INWT Statistics

Thanks for your attention!

INWT Statistics GmbH
Hauptstraße 8
Meisenbach Höfe, Aufgang 3a
10827 Berlin

+49 30 12082310

info@inwt-statistics.de

www.inwt-statistics.de