Berkstats_Day2.Rmd

Ulrich Matter

22 February 2017

Working with Data in R: Data Structures and Indeces

Vectors and Lists

```r
# A vector containing numeric (or integer) values
numeric_vector <- 10:20
numeric_vector[2]
```

```
## [1] 11
```

```r
numeric_vector[2:5]
```

```
## [1] 11 12 13 14
```

```r
# A string vector ('a vector containing text')
string_vector <- c("a", "b", "c")
string_vector[-3]
```

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

```r
# Lists
# A list can contain different types of elements, for example a numeric vector and a string_vector
mylist <- list(numbers = numeric_vector, letters = string_vector)
mylist
```

```
## $numbers
##  [1] 10 11 12 13 14 15 16 17 18 19 20
##
## $letters
## [1] "a" "b" "c"
```

```r
# We can access the elements of a list in various ways
# with the element's name
mylist$numbers
```

```
##  [1] 10 11 12 13 14 15 16 17 18 19 20
```

```r
mylist["numbers"]
```

```
## $numbers
##  [1] 10 11 12 13 14 15 16 17 18 19 20
```

```r
# via the index
mylist[1]
```

```
## $numbers
##  [1] 10 11 12 13 14 15 16 17 18 19 20
```

```r
# with [[]] we can access directly the content of the element
mylist[[1]]
```

```
##  [1] 10 11 12 13 14 15 16 17 18 19 20
```

```r
# lists can also be nested (list of lists of lists....)
mynestedlist <- list(a = mylist, b = 1:5)
```

Matrices and Data Frames

```r
# matrices
mymatrix <- matrix(numeric_vector, nrow = 4)
```

```
## Warning in matrix(numeric_vector, nrow = 4): data length [11] is not a sub-
## multiple or multiple of the number of rows [4]
```

```r
# get the second row
mymatrix[2,]
```

```
## [1] 11 15 19
```

```r
# get the first two columns
mymatrix[, 1:2]
```

```
##      [,1] [,2]
## [1,]   10   14
## [2,]   11   15
## [3,]   12   16
## [4,]   13   17
```

```r
# data frames ("lists as columns")
mydf <- data.frame(Name = c("Alice", "Betty", "Claire"), Age = c(20, 30, 45))
mydf
```

```
##     Name Age
## 1  Alice  20
## 2  Betty  30
## 3 Claire  45
```

```r
# select the age column
mydf$Age
```

```
## [1] 20 30 45
```

```r
mydf[, "Age"]
```

```
## [1] 20 30 45
```

```r
mydf[, 2]
```

```
## [1] 20 30 45
```

```r
# select the second row
mydf[2,]
```

```
##    Name Age
## 2 Betty  30
```

Classes and Data Structure

```r
# have a look at what kind of object you are dealing with
class(mydf)
```

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

```
class(mymatrix)
```

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

```
# have a closer look at the data structure
str(mydf)
```

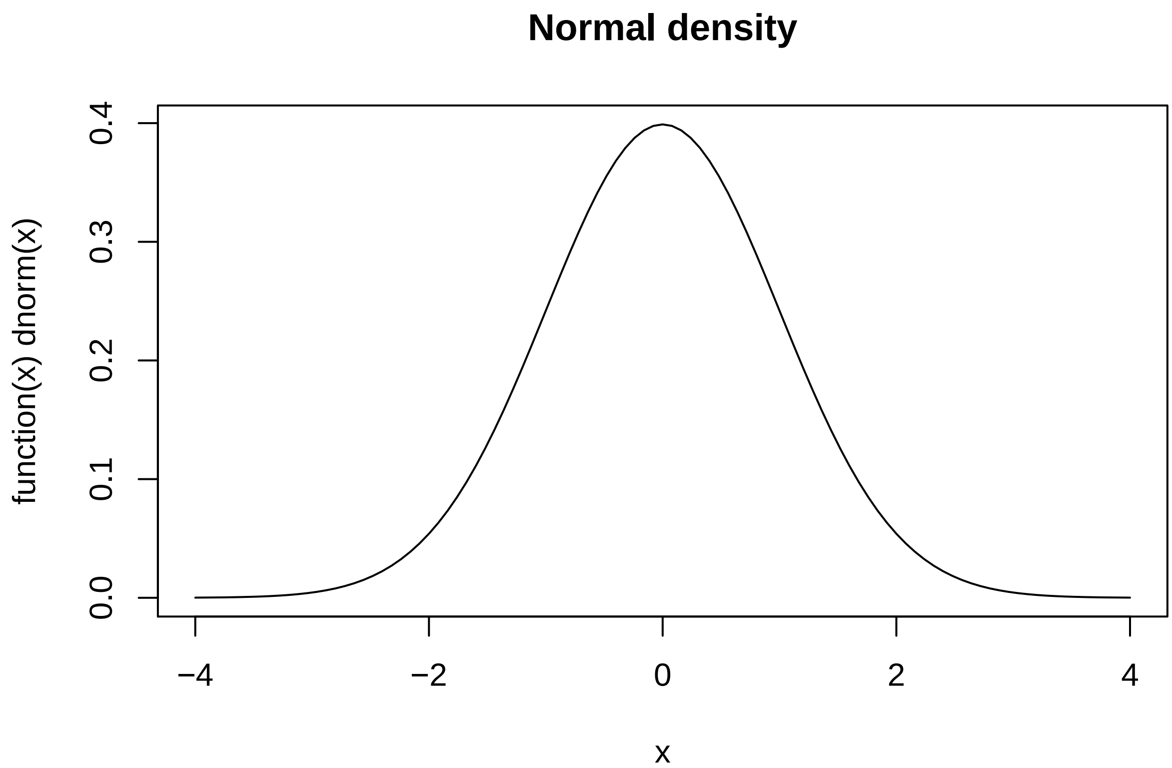
```
## 'data.frame':    3 obs. of  2 variables:
##  $ Name: Factor w/ 3 levels "Alice","Betty",..: 1 2 3
##  $ Age : num  20 30 45
```

Z-Scores and the Standard Normal Distribution

```
# The Z-Score formula in R
# define the parameter values
X <- 10
mu <- 12
sigma <- 2
# compute the z-score
z <- (X - mu) / sigma
z
```

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

```
# Plot the Standard Normal Distribution
plot(function(x) dnorm(x), -4, 4, main = "Normal density")
```



```r
# Get the area under the curve (probability of observing a value of a certain size)
pnorm(-1)
```

```
## [1] 0.1586553
```

```r
pnorm(-2)
```

```
## [1] 0.02275013
```

```r
pnorm(-1) - pnorm(-2)
```

```
## [1] 0.1359051
```

Standard Errors

```r
# forumla for standard error
# define parameter values
s <- 20
n <- 100
# compute the standard error (of the mean)
se <- s / sqrt(n)
se
```

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

```r
# write your own standard error (of the mean) function
se <- function(x) {
     s <- sd(x)
     n <- length(x)
     se <- s / sqrt(n)

     return(se)
}

# draw a random sample of size 100 and compute the mean and its estimated standard error
mysample <- rnorm(100)
mean(mysample)
```

```
## [1] 0.03004297
```

```r
se(mysample)
```

```
## [1] 0.09441269
```

```r
# repeat this but this time with a larger sample
mysample <- rnorm(1000)
mean(mysample)
```

```
## [1] -0.0006847108
```

```r
se(mysample)
```

```
## [1] 0.03058479
```

Hypothesis Testing: the T-Statistic

Reproduce the example from the presentation

```
# define parameters
mu <- 39000
sample_mean <- 37000
sample_sd <- 6150
n <- 100

# calculate the  standard error of the sample mean
se <- sample_sd / sqrt(n)

# compute the t-statistic (and compare it with the critical value)
t <- (sample_mean - mu) / se

# look up the p-value
# (the fraction of the mass under the standard normal distribution)
2*pnorm(-abs(t))
```

```
## [1] 0.001145829
```

Extended Example

```
# I) Compute the t-value step by step with our own implementation while controlling the properties of t

# define size of sample
n <- 100
# draw the random sample from a normal distribution with mean 10 and sd 2
sample <- rnorm(n, mean = 10, sd = 2)

# compute the sample mean
sample_mean <- mean(sample)
# compute the sample sd
sample_sd <- sd(sample)
# estimated standard error of the mean
mean_se <- sample_sd/sqrt(length(sample))

# compute the t-statistic for the null hypothesis: H0: mu = 9
t <- (sample_mean - 10) / mean_se
t
```

```
## [1] 0.1539256
```

```
# get the p value
2*pnorm(-abs(t))
```

```
## [1] 0.8776684
```

```
# II) Apply the R-function t.test
t.test(sample, mu = 10)
```

```
##
##  One Sample t-test
```

```
##
## data:  sample
## t = 0.15393, df = 99, p-value = 0.878
## alternative hypothesis: true mean is not equal to 10
## 95 percent confidence interval:
##   9.612311 10.452898
## sample estimates:
## mean of x
##   10.0326
```