The “Live” Code of Lecture 4

```
############################################
# LECTURE 4, Part 1: Completing function   #
# from last time with an optional argument #
############################################

# During Lecture 3 we worked on a function with an optional
# or "NULL" argument: if the argument get.out.as.text is not NULL,
# then the function saveFun would get the output as text.

# In this function, we introduced two new R functions:
# sprintf, and the conditional.


# sprintf
#########


x = 5000; T = 30; r = 4
out = round(  x/(1+r/100)^T  )
# Note that you can run several commands on one line,
# separated by semicolons


sprintf("If you want to spend %s after %s years
and the interest rate is %s percent,
you have to save %s.", x, T, r, out)
```

```
## [1] "If you want to spend 5000 after 30 years\nand the interest rate is 4 percent, \nyou have to save
```

```
# Note: The s in "%s" means "string". See Section 8.2 in
# "R for Everyone".

# If you want get rid of the editing symbols like "" or \n, you have
# to use the cat function


cat(sprintf("If you want\t to spend %s\n after %s years
and the\n interest\t rate is %s percent,
you have to save %s.", x, T, r, out))
```

```
## If you want   to spend 5000
##  after 30 years
## and the
##  interest     rate is 4 percent,
## you have to save 1542.
```

```
# Conditionals
##############
```

```r
# See Chapter 9 in "R for Everyone"

arg = "yes"

if (arg ==  "no"){
  print("I have nothing to say :-(")
}


# So what if arg = "yes"?

arg = "yes"

if (arg ==  "no"){
  print("I have nothing to say :-(")
}

if (arg ==  "yes"){
  print("I have nothing to say :-)")
}
```

```
## [1] "I have nothing to say :-)"
```

```r
arg = "no"

if (arg ==  "no"){
  print("I have nothing to say :-(")
} else if (arg=="yes"){
  print(":-))")
}
```

```
## [1] "I have nothing to say :-("
```

```r
# Be very careful with the positions of the curly brackets
# If they are not in the right position, you will get
# an error. This can sometimes be quite tricky.


# Now let's make a function of this

saySomething = function(arg){

  #copy/paste from above
  if (arg ==  "no"){
    print("I have nothing to say :-(")
  } else if (arg=="yes"){
    print(":-))")
  }
}

saySomething("no")
```

```
## [1] "I have nothing to say :-("
saySomething("yes")
```

```
## [1] ":-))"
```

```
# Now we go back to our savings function
########################################

a = NULL
b = "yes"
is.null(a)
```

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

```
is.null(b)
```

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

```
get.out.as.text = "yes"

saveFun = function(spending = 5000,
                   interestRate = 4,
                   horizon = 30,
                   get.out.as.text = NULL){
  x = spending
  r = interestRate
  T = horizon
  out = round(  x/(1+r/100)^T  )

  if (is.null(get.out.as.text)){
    return(out)
    # everything in a function that comes after return is not executed
    # if return is executed...

  } else if (get.out.as.text == "yes"){
    cat(sprintf("If you want to spend %s after %s years
and the interest rate is %s percent,
you have to save %s!", x, T, r, out))
  }
}

saveFun()
```

```
## [1] 1542
```

```
saveFun(get.out.as.text = "yes")
```

```
## If you want to spend 5000 after 30 years
## and the interest rate is 4 percent,
## you have to save 1542!
```

```r
saveFun(spending = 5000,
        interestRate = 1,
        horizon = 30,
        get.out.as.text = "yes")
```

```
## If you want to spend 5000 after 30 years
## and the interest rate is 1 percent,
## you have to save 3710!
```

```r
saveFun(spending = 5000,
        interestRate = -0.5,
        horizon = 30)
```

```
## [1] 5811
```

```r
####################################################
# LECTURE 4, Part 2: Reading data from csv files   #
####################################################




# See Section 6.1 and 5.1 in "R for Everyone".

rm(list = ls())

# Set the working directory to the folder
# where you have the csv files from the SNB


# On a Mac it may look like this
#setwd("/Users/Thomas/Dropbox/Programmierkurs/Data")

# On Windows it may look like this
#setwd("D:/Programmierkurs/Data")

# Note the forward slashes in the directory!!!!!


# Load the data... Which one
# works for you?

rawData = read.csv(file = "data/SNB Xrates downloaded.csv")

rawXrates = read.csv(file = "data/SNB Xrates downloaded clean.csv")

rawXrates =
  read.csv(file =
             "data/SNB Xrates downloaded clean.csv",
           sep = ",")

rawXrates$XX = NA
```

```r
# In my case, there are still the empty rows and columns.
# However, even if you do not have them, you can execute
# the below commands


# What is the type of rawXrates?
class(rawXrates)
```

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

```r
# Get the names of the columns ("variables"
# in the statistical sense)
names(rawXrates)
```

```
##  [1] "Date"  "X"     "X.1"   "D0"    "X.2"   "X.3"   "X.4"   "X.5"
##  [9] "D1"    "X.6"   "X.7"   "X.8"   "X.9"   "Value" "XX"
```

```r
# You can use the names to get a column



head(rawXrates["Date"])
```

```
##      Date
## 1 1914-01
## 2 1914-01
## 3 1914-01
## 4 1914-01
## 5 1914-01
## 6 1914-01
```

```r
# Use this trick to select only the variables we are interested in

varList = c("Date", "D0", "D1", "Value")

rawXrates = rawXrates[varList]

head(rawXrates)
```

```
##      Date D0     D1 Value
## 1 1914-01 M0   EUR1    NA
## 2 1914-01 M0   GBP1 25.28
## 3 1914-01 M0 DKK100    NA
## 4 1914-01 M0 NOK100    NA
## 5 1914-01 M0 CZK100    NA
## 6 1914-01 M0 HUF100    NA
```