The "Live" Code of Lecture 6

```
##############################################
#      LECTURE 6: Data analysis              #
##############################################

# We will first go through (almost) the same steps as in
# Lecture 5. This time, we will organize the code in a more
# efficient way, such that it becomes more "reusable"
# You will also learn how to merge two data sets.
# The most important thing today are scatter plots
# and regression analysis. These are key tools for
# data analysis.

rm(list = ls()) # Empty workspace to start with a "clean sheet"

# REPLACE THE WORKING DIRECTORY BELOW WITH THE ONE FOR YOUR DEVICE
setwd("D:/Dropbox/Mac&Surf/Programmierkurs Dropb/Data")
#load("rawExpImp (1).RData")

# Read files, adjust file names to your situation!

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

rawAussen = read.csv(file =
    "SNB Aussenhandel downloaded clean noEmptyCol.csv")

# If you are having troubles with the data, use the following

# load("rawXrates.RData")
# load("rawExpImp.RData")

# You get these data from the online script on
# https://binswanger.github.io/practicaldata_hs16/#Problems--What-if-some-numbers-are-automatically-con
# Put these files in your working directory

# SET PARAMETERS HERE!
#######################

startYear = 2000
curr = "USD1"
tradeDir = "A"  # Direction of trade
                # Values are E (Einfuhr)
                # and A (Ausfuhr)
                # run unique(rawAussen$D0) for overview
goodsType = "MAE" # run unique(rawAussen$D1)

measure = "R" # run unique(rawAussen$D2)


unique(rawAussen$D0)
```

```
## [1] E A H
## Levels: A E H
```

```
unique(rawAussen$D1)
```

```
##  [1] T    MAE  PUB  C    TBS0 F    M    U    P    TBS1
## Levels: C F M MAE P PUB T TBS0 TBS1 U
```

```
unique(rawAussen$D2)
```

```
## [1] WMF N   R
## Levels: N R WMF
```

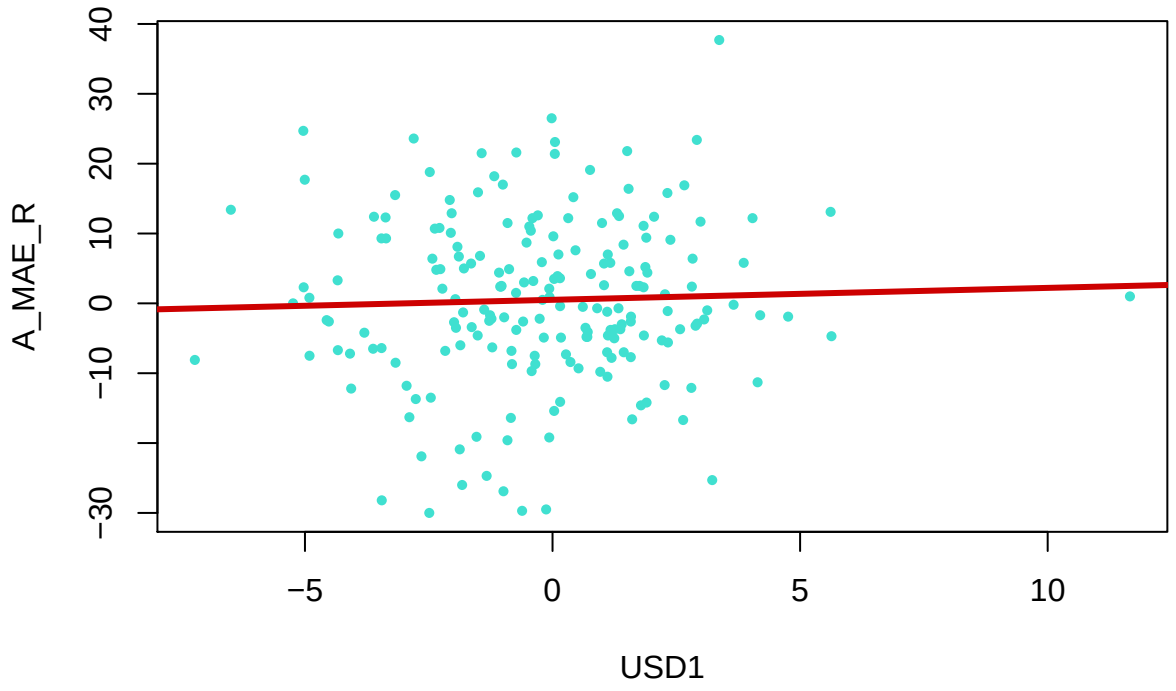
```
# Functions
###########


toGrowth = function(x){
  out = (tail(x, -1)/head(x, -1) -1 )*100
  # This is one element shorter than input,
  # add an NA as first value
  out = c(NA, out)
  return(out)
}



# Customize the data
####################

# A unique time identifier for BOTH data
library(stringr)

rawXrates$year = as.numeric(
  substr(rawXrates$Date, start = 1, stop = 4)  )

rawXrates$month = as.numeric(
  substr(rawXrates$Date, start = 6, stop = 7)  )

rawXrates$timeID = rawXrates$year +
  (rawXrates$month-1)/12


rawAussen$year = as.numeric(
  substr(rawAussen$Date, start = 1, stop = 4)  )

rawAussen$month = as.numeric(
  substr(rawAussen$Date, start = 6, stop = 7)  )

rawAussen$timeID = rawAussen$year +
  (rawAussen$month-1)/12
```

```
xrates = subset(rawXrates,
        D1 == curr &  # NOTE the apparance of the VARIABLE xrate!!!
           D0 == "M0" &
          timeID >= startYear , # NOTE the apparance of the VARIABLE startYear!!!
        select = c("timeID", "D1", "Value"))

aussen =
  subset(rawAussen,

         D0 == tradeDir  &
         D1 == goodsType &
         D2 == measure   &
         timeID >= startYear ,

         select = c("timeID", "D0","D1","D2", "Value"))


# Bring data into wide format
library(reshape2)
xrates = dcast(xrates, timeID ~ D1, value.var = "Value")

aussen = dcast(aussen, timeID ~ D0 + D1 + D2, value.var = "Value")

# Merge the two data sets (NEW!!!!)
# DA stands for "Data for Analysis"
DA = merge(xrates, aussen, by = "timeID")
# The "by" argument contains the so-called "key".


# [FOR LATER] convert to growth rates
DA[[curr]] = toGrowth(DA[[curr]])


# ANALYSIS
############

# Variable name for exports/imports
names(DA)
```

```
## [1] "timeID"  "USD1"    "A_MAE_R"
```

```
vn = paste(tradeDir, goodsType, measure, sep = "_")



# Make a plot
plot(DA[[curr]], DA[[vn]],
        # NOTE: This code works for all types of export/import
        # data and exchange rates!!! They are captured in the
        # variable names

        pch = 16, # data points as dots; google "r plot pch"
```

```
        cex = .7, # the size of the point (exam question from last semester :-)

        xlab = curr, ylab = vn, # axis labels

        col = c("turquoise") # color of dots

    )  # The closing paranthesis

# Add a regression line

abline(  lm(DA[[vn]]  ~   DA[[curr]]
         ), col="red3", lwd = 3)
```




```
# Run a regression
reg = lm(DA[[vn]]~DA[[curr]])
summary(reg)
```

```
##
## Call:
## lm(formula = DA[[vn]] ~ DA[[curr]])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.103  -6.448  -1.142   8.119  36.618
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5077     0.8411   0.604    0.547
## DA[[curr]]    0.1707     0.3372   0.506    0.613
##
## Residual standard error: 11.79 on 196 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.001307,   Adjusted R-squared:  -0.003789
## F-statistic: 0.2564 on 1 and 196 DF,  p-value: 0.6131
```

```
# How calculate changes?

test = 1:10
head(test, -1)
```

```
## [1] 1 2 3 4 5 6 7 8 9
```

```
tail(test, -1)
```

```
## [1]  2  3  4  5  6  7  8  9 10
```

```
dTest = (tail(test, -1)/head(test, -1) -1)*100
```