Financial Econometrics
Problem Set
Problem set 1, 2021

Exercise 1. *Describing Covid* [4]
In the lecture notes you find a picture of the (log) increase of confirmed cases of covid for selected countries.

An alternative way of presenting the data is to look at (say) weekly *changes* (i.e. new cases).

Show the weekly changes in new cases for the following countries: Norway, Sweden, U.K. and the U.S.

Exercise 2. *Crypto - describe* [3]
In this problem we will look at the two crypto currencies Bitcoin (BTC) and Ethereum (ETH). Pull the complete time series of these two currencies. (There is some R code you may find helpful).

- Describe the two time series.
- Plot the two time series in a single plot. Hint: You may want to use different axes.
- Calculate annual return. Describe the returns.

Exercise 3. *Exposure estimation* [4]
From Yahoo finance, download the following data

- Four Norwegian equities
  - NHY.OL (Norsk Hydro)
  - EQNR.OL (Equinor)
  - NAS.OL (Norwegian Air Shuttle)
  - GSF.OL (Grieg Seafood)
- The OSE stock market index (ÔSEAX)

From Fred, download

- Oil prices (DCOILBRENTEU)
- Price index for seafood (PSALMUSDM)

Calculate monthly returns for all these variables.

Conside the regressions

$$r_{it} = a + b_m r_{mt} + b_{oil}\Delta Oil + \varepsilon_{it},$$

and

$$r_{it} = a + b_m r_{mt} + b_{oil}\Delta Oil_t + b_{fish}\Delta Fish_t + \varepsilon_{it}.$$

Here r_{it} is the stock return, r_{mt} the stock market index return, ΔOil_t return for oil prices, and $\Delta Fish_t$ the return for the seafood prices.

1. Use data for 2013-2019. Perform and report these regressions for the four norwegian equities.
2. Comment on the results. What is the interpretation of the coefficients b_{oil} and b_{fish}?.

Exercise 4. *Updating Blindern Weather and the OSE.* [6]
You have seen the evidence linking Oslo weather to the return of the Oslo Stock Exchange. In this exercise we will update this evidence. The homepage contains new Blindern observations for 1980–2021. The format has changed slightly due to updates for the weather data. It is now daily observations. Update the estimates linking returns to measures of good or bad weather. You may have to redefine a couple of the weather measures.

Exercise 5. *Trump and Hillary* [2]

Prediction markets (also known as predictive markets, information markets, decision markets, idea futures, event derivatives, or virtual markets) are exchange-traded markets created for the purpose of trading the outcome of events. The market prices can indicate what the crowd thinks the probability of the event is. A prediction market contract trades between 0 and 100%. It is a binary option that will expire at the price of 0 or 100%.

The asset traded on in a prediction market pays off one unit if an event occurs, nothing if it does not. In option terms this is called a *binary option*. For the more theoretically inclided this is a *state price*. Since the events traded is an enumeration of all possible future events, we can treat them using the rules of state pricing, as discussed in financial theory. If we let x be the future payoff, and p the current price of the binary option, we can price the option as follows

$$p = E[mx]$$

where m is the *pricing kernel*. Financial theory can be used to specify properties of m.

Let us consider one example, *political prediction markets*. Such contracts are for example traded at the University of Iowa's business school, the IEM, Iowa Electronic Markets. In 2016, the US Presidental election ended up being between Hillary Clinton and Donald Trump. Long before their nominations became clear, contracts for Republican and Democratic winners were traded. On the course homepage you will find data for these contracts.

1. Plot, in one picture, prices of the two contracts for Democratic and Republican president.

Exercise 6. *Disney and Star Wars* [2]

The Latest Star Wars film (episode 7) was released on December 17, 2015. It became the fastest film ever to one billion very soon afterwards. Investigate how this affected the Stock Price of Disney: Download Stock Prices of Disney (from Yahoo finance, or somewhere similar).

Plot the evolution of the Stock Price of Disney through December of 2015.

Comment

Exercise 7. *NIBOR* [5]

The "fixing" of the Norwegian Equivalent of LIBOR, NIBOR, has recently moved to the Oslo Stock Exchange. This is a reaction to the "LIBOR Scandal", they need more public scrutiny of the process of setting the interest rate. The "fixing" is done as an average of quotes by a number of financial institutions. At the OSE (and the course homepage) you will find data on this fixing for 2013-2017.

Investigate whether there are any systematic biases across member institutions, such as one institution constantly quoting a lower rate than others.

Hint: Ways to investigate this:

- Histogram of differences between the institutions and the fixing.
- Testing for differences between the fixing mean and the institutional mean.

Also see whether this behaviour is changing over time, by doing the above year by year, and looking at changes.

Financial Econometrics

Solutions

Problem set 1, 2021

Exercise 1. *Describing Covid* [4]

Exercise 2. *Crypto - describe* [3]

```
library(jsonlite)
library(anytime)
library(xts)

get_crypto <- function(ticker){
    call <- paste0("https://min-api.cryptocompare.com/data/histoday?fsym=",
                   ticker,
                   "&tsym=USD&allData=true&e=CCCAGG")
    hist_crypto <-fromJSON(call)
    time <- anytime(hist_crypto$Data$time)
    dates <- anydate(time)
    crypto_close <- xts(hist_crypto$Data$close, order.by=dates)
    crypto_close <- crypto_close[crypto_close>0]
    return (crypto_close)
}
```

Doing the descriptives

```
library(chron)
library(quantmod)
library(stargazer)

source ("../2021_09_get_bitcoin_data/get_crypto.R")
outdir <- "../../results/2021_09_describe_crypto/"

library(xtable)

btc <- get_crypto("BTC")
eth <- get_crypto("ETH")

yearlist <- 2010:2021
firstyear <- yearlist[1]

tabl <- matrix(nrow=length(yearlist),ncol=4)
rownames(tabl) <- as.character(yearlist)
colnames(tabl)  <- c("min","mean","med","max")
for (year in yearlist){
    i <- 1+(year-firstyear)
    btc_this_year <- btc[years(index(btc))==year]
    tabl[i,1] <- min(btc_this_year)
    tabl[i,2] <- mean(btc_this_year)
    tabl[i,3] <- median(btc_this_year)
    tabl[i,4] <- max(btc_this_year)
}


xtabl <- xtable(tabl,digits=1)
```

```
filename <- paste0(outdir,"btc_describe_prices.tex")
print.xtable(xtabl, file=filename, floating=FALSE)

yearlist <- 2015:2021
firstyear <- yearlist[1]

for (year in yearlist){
    i <- 1+(year-firstyear)
    eth_this_year <- eth[years(index(eth))==year]
    tabl[i,1] <- min(eth_this_year)
    tabl[i,2] <- mean(eth_this_year)
    tabl[i,3] <- median(eth_this_year)
    tabl[i,4] <- max(eth_this_year)
}

xtabl <- xtable(tabl,digits=1)
filename <- paste0(outdir,"eth_describe_prices.tex")
print.xtable(xtabl, file=filename, floating=FALSE)

btc_returns <- na.omit(annualReturn(btc,leading=FALSE))
names(btc_returns) <- "BTC"
eth_returns <- na.omit(annualReturn(eth,leading=FALSE))
names(eth_returns) <- "ETH"


filename <- paste0(outdir,"btc_eth_returns.tex")
rets <- merge(btc_returns, eth_returns)*100
stargazer(rets,
          digits=0,
          out=filename,
          float=FALSE,
          summary=TRUE)
```

Summary of prices, BTC

	min	mean	med	max
2010	0.0	0.1	0.1	0.4
2011	0.3	5.6	3.3	29.6
2012	4.2	8.3	6.8	13.7
2013	13.3	200.1	116.4	1237.5
2014	111.6	522.9	485.5	1014.7
2015	164.9	272.0	248.2	463.2
2016	368.0	567.0	583.1	972.2
2017	785.4	3981.1	2583.8	19345.5
2018	3232.5	7547.0	6915.7	17172.3
2019	3375.3	7379.3	7807.4	12913.3
2020	4916.8	11117.8	9715.6	28972.4
2021	29388.9	44567.6	45058.0	63558.5

	min	mean	med	max
2015	0.4	0.9	0.9	2.8
2016	0.9	9.8	10.9	20.6
2017	8.2	221.6	240.9	812.5
2018	84.1	481.4	451.1	1385.0
2019	103.7	181.3	173.9	334.4
2020	110.2	307.5	245.2	752.5
2021	730.6	2272.4	2148.1	4178.0

Describe returns

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
BTC	11	849	1,749	−73	46.2	820.5	5,866
ETH	6	1,768	3,607	−82	85	693	9,104

Doing the plot

```
source ("../2021_09_get_bitcoin_data/get_crypto.R")
outdir <- "../../results/2021_09_plot_btc/"
btc <- get_crypto("BTC")
eth <- get_crypto("ETH")

filename <- paste0(outdir,"daily_bitcoin_and_eth_usd.png")
png(filename,width=800,height=400)
plot(btc,
     main="Bitcoin and ETH",
     bty="n",
     xlab="time",
     col="royalblue",
     ylab="Price",
     bty="n")
box(bty="L")
par(new=TRUE)
plot(eth,
     xaxt="n",
     yaxt="n",
     main="",
     xlab="",
     ylab="",
     col = "green",
     type="l",
     bty="n"
     )
axis(4,col="green")
mtext("ETH",side=4,line=0)
legend("topleft",c("BTC","ETH"),
       col = c("blue","green"),
       lty=c(1,1))
dev.off()
```

Exercise 3. *Exposure estimation* [4]

There is one issue that is actually an inconsistency in my specification. The Norwegian data is return in NOK. The others are returns in USD. One can argue that one should have returns in same currency. If anyone translated the currency, extra points. Otherwise one can proceed by assuming the monthly currency changes will only add noise, and my problem specification is without mention of this currency inconsistency.

Read equity data

```
library(quantmod)
fdate <- as.Date("2010-01-01")
                                        # norsk hydro
nhy <- getSymbols("NHY.OL",
                  from=fdate,
                  source="yahoo",
                  auto.assign=FALSE)
```

```
nhy_daily_prices <- na.omit(nhy$NHY.OL.Adjusted)
names(nhy_daily_prices) <- "nhy_daily_prices"
                                        # equinor
eqnr <- getSymbols("EQNR.OL",
                   from=fdate,
                   source="yahoo",
                   auto.assign=FALSE)
eqnr_daily_prices <- na.omit(eqnr$EQNR.OL.Adjusted)
names(eqnr_daily_prices) <- "eqnr_daily_prices"
                                        # Norwegian
nas <- getSymbols("NAS.OL",
           from=fdate,
           source="yahoo",
           auto.assign=FALSE)
nas_daily_prices <- na.omit(nas$NAS.OL.Adjusted)
names(nas_daily_prices) <- "nas_daily_prices"

gsf <- getSymbols("GSF.OL",
                  from=fdate,
                  source="yahoo",
           auto.assign=FALSE)
gsf_daily_prices <- na.omit(gsf$GSF.OL.Adjusted)
names(gsf_daily_prices) <- "gsf_daily_prices"

summary(gsf_daily_prices)

oseax <- getSymbols("^OSEAX",
                    from=fdate,
                    source="yahoo",
                    auto.assign=FALSE)
print(summary(oseax))
oseax_daily_prices = na.omit(oseax$OSEAX.Adjusted)
names(oseax_daily_prices) <- "oseax_daily_prices"

nhy_monthly_returns   <- monthlyReturn(nhy_daily_prices,  leading=FALSE)
oseax_monthly_returns <- monthlyReturn(oseax_daily_prices,leading=FALSE)
nas_monthly_returns   <- monthlyReturn(nas_daily_prices,  leading=FALSE)
eqnr_monthly_returns  <- monthlyReturn(eqnr_daily_prices, leading=FALSE)
gsf_monthly_returns   <- monthlyReturn(gsf_daily_prices,  leading=FALSE)

names(nhy_monthly_returns)   <- "nhy_monthly_returns"
names(oseax_monthly_returns) <- "oseax_monthly_returns"
names(nas_monthly_returns)   <- "nas_monthly_returns"
names(eqnr_monthly_returns)  <- "eqnr_monthly_returns"
names(gsf_monthly_returns)   <- "gsf_monthly_returns"

index(nhy_monthly_returns)   <- as.yearmon(index(nhy_monthly_returns))
index(oseax_monthly_returns) <- as.yearmon(index(oseax_monthly_returns))
index(nas_monthly_returns)   <- as.yearmon(index(nas_monthly_returns))
index(eqnr_monthly_returns)  <- as.yearmon(index(eqnr_monthly_returns))
index(gsf_monthly_returns)   <- as.yearmon(index(gsf_monthly_returns))
```

Read oil data

```
library(quantmod)
```

```
fdate <- as.Date("2010-01-01")

                                        # brent prices, in dollars
brent <- getSymbols("DCOILBRENTEU",
                    from=fdate,
                    src="FRED",
                    auto.assign=FALSE)
daily_brent_oil_price        <- na.omit(brent)
names(daily_brent_oil_price) <- "daily_brent_oil_price"

monthly_oil_returns        <- monthlyReturn(daily_brent_oil_price,leading=FALSE)
names(monthly_oil_returns) <- "monthly_oil_returns"
monthly_oil_returns        <- na.omit(monthly_oil_returns )
index(monthly_oil_returns) <- as.yearmon(index(monthly_oil_returns))
```

Read fish data

```
library(quantmod)
fdate <- as.Date("2010-01-01")

fish <- getSymbols("PSALMUSDM",
                   from=fdate,
                   src="FRED",
                   auto.assign=FALSE)
daily_fish_prices <- na.omit(fish)
names(daily_fish_prices) <- "daily_fish_prices"

monthly_fish_returns <- monthlyReturn(daily_fish_prices,leading=FALSE)
monthly_fish_returns <- na.omit(monthly_fish_returns)
names(monthly_fish_returns) <- "monthly_fish_returns"
index(monthly_fish_returns) <- as.yearmon(index(monthly_fish_returns))
```

Exposure estimation

```
library(stargazer)
outdir <- "../../results/2021_09_exposure_estimation/"

source("read_equity_data.R")
source("read_oil_data.R")
source("read_fish_data.R")

fdate <- as.Date("2013-01-01")
ldate <- as.DAte("2019-12-31")

data <- na.omit(merge(nhy_monthly_returns,
                      oseax_monthly_returns,
                      monthly_oil_returns,
                      all=FALSE))
data <- window(data, start=fdate, end=ldate)
nhy_regr <- lm(data$nhy_monthly_return
               ~ data$oseax_monthly_returns
               + data$monthly_oil_returns)
summary(nhy_regr)
```

```
data <- na.omit(merge(eqnr_monthly_returns,
                      oseax_monthly_returns,
                      monthly_oil_returns,
                      all=FALSE))
data <- window(data, start=fdate, end=ldate)

eqnr_regr <- lm(data$eqnr_monthly_return
                ~ data$oseax_monthly_returns
                + data$monthly_oil_returns)
summary(eqnr_regr)

data <- na.omit(merge(nas_monthly_returns,
                      oseax_monthly_returns,
                      monthly_oil_returns,
                      all=FALSE))
data <- window(data, start=fdate, end=ldate)
nas_regr <- lm(data$nas_monthly_return
               ~ data$oseax_monthly_returns
               + data$monthly_oil_returns)
summary(nas_regr)

data <- na.omit(merge(gsf_monthly_returns,
                      oseax_monthly_returns,
                      monthly_oil_returns,
                      all=FALSE))
data <- window(data, start=fdate, end=ldate)
gsf_regr <- lm(data$gsf_monthly_return
                ~ data$oseax_monthly_returns
                + data$monthly_oil_returns)
summary(gsf_regr)

filename <- paste0(outdir,"oil_exposure_regression.tex")
stargazer(nhy_regr,
          eqnr_regr,
          nas_regr,
          gsf_regr,
          out=filename,
          model.numbers=FALSE,
          dep.var.labels=c("NHY","EQNR","NAS","GSF"),
          covariate.labels=c("Rm","Oil","Constant"),
          omit.stat=c("rsq","ser","f"),
          header=FALSE,
          float=FALSE
          )


data <- na.omit(merge(nhy_monthly_returns,
                      oseax_monthly_returns,
                      monthly_oil_returns,
                      monthly_fish_returns,
                      all=FALSE))
data <- window(data, start=fdate, end=ldate)
nhy_regr <- lm(data$nhy_monthly_return
               ~ data$oseax_monthly_returns
               + data$monthly_oil_returns
               + data$monthly_fish_returns)
```

```
summary(nhy_regr)

data <- na.omit(merge(eqnr_monthly_returns,
                      oseax_monthly_returns,
                      monthly_oil_returns,
                      monthly_fish_returns,
                      all=FALSE))
data <- window(data, start=fdate, end=ldate)
eqnr_regr <- lm(data$eqnr_monthly_return
                ~ data$oseax_monthly_returns
                + data$monthly_oil_returns
                + data$monthly_fish_returns)
summary(eqnr_regr)

data <- na.omit(merge(nas_monthly_returns,
                      oseax_monthly_returns,
                      monthly_oil_returns,
                      monthly_fish_returns,
                      all=FALSE))
data <- window(data, start=fdate, end=ldate)
nas_regr <- lm(data$nas_monthly_return
               ~ data$oseax_monthly_returns
               + data$monthly_oil_returns
               + data$monthly_fish_returns)
summary(nas_regr)

data <- na.omit(merge(gsf_monthly_returns,
                      oseax_monthly_returns,
                      monthly_oil_returns,
                      monthly_fish_returns,
                      all=FALSE))
data <- window(data, start=fdate, end=ldate)
gsf_regr <- lm(data$gsf_monthly_return
                ~ data$oseax_monthly_returns
                + data$monthly_oil_returns
                + data$monthly_fish_returns)
summary(gsf_regr)

library(stargazer)
filename <- paste0(outdir,"oil_fish_exposure_regression.tex")
stargazer(nhy_regr,
          eqnr_regr,
          nas_regr,
          gsf_regr,
          out=filename,
          model.numbers=FALSE,
          dep.var.labels=c("NHY","EQNR","NAS","GSF"),
          covariate.labels=c("Rm","Oil","Fish","Constant"),
          omit.stat=c("rsq","ser","f"),
          float=FALSE,
          header=FALSE
          )
```

	Dependent variable:			
	NHY	EQNR	NAS	GSF
Rm	1.440***	1.123***	1.967***	0.614
	(0.294)	(0.145)	(0.692)	(0.441)
Oil	−0.271***	0.190***	−0.476**	0.036
	(0.097)	(0.048)	(0.227)	(0.145)
Constant	−0.005	−0.001	−0.021	0.031**
	(0.008)	(0.004)	(0.018)	(0.012)
Observations	81	81	81	81
Adjusted R^2	0.215	0.685	0.073	0.021
Note:				*p<0.1; **p<0.05; ***p<0.01

	Dependent variable:			
	NHY	EQNR	NAS	GSF
Rm	1.437***	1.123***	1.960***	0.612
	(0.295)	(0.146)	(0.695)	(0.444)
Oil	−0.272***	0.190***	−0.478**	0.035
	(0.097)	(0.048)	(0.229)	(0.146)
Fish	0.054	−0.012	0.112	0.025
	(0.090)	(0.044)	(0.211)	(0.135)
Constant	−0.005	−0.001	−0.022	0.031**
	(0.008)	(0.004)	(0.018)	(0.012)
Observations	81	81	81	81
Adjusted R^2	0.209	0.681	0.065	0.009
Note:				*p<0.1; **p<0.05; ***p<0.01

Exercise 4. *Updating Blindern Weather and the OSE.* [6]

Exercise 5. *Trump and Hillary* [2]

This R code reads in the data

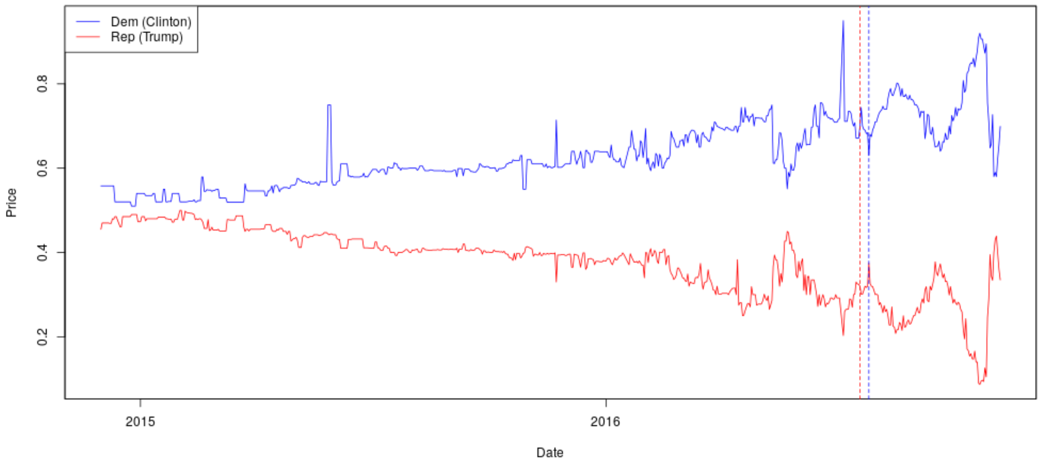
```
library(zoo)
indir <- "/home/bernt/data/2016/iowa_markets/"
filename <- paste0(indir,"PRES16WTA.txt")
data <- read.csv(filename,sep=";",skip=6,header=TRUE)
data$Date <- as.Date(data$Date,,format="%m/%d/%y")
REPData <- data[(data$Contract=="REP16_WTA"),]
REPPrices <- na.omit(zoo(REPData$LastPrice,order.by=REPData$Date))
DEMData <- data[(data$Contract=="DEM16_WTA"),]
DEMPrices <- na.omit(zoo(DEMData$LastPrice,order.by=DEMData$Date))
```

And this generates the plot

```
outdir <- "../../results/2016_10_describe_iowa_president_2016/"
```

```
source ("/home/bernt/data/2016/iowa_markets/read_PRES16WTA.R")
data <- merge(DEMPrices,REPPrices)
filename <- paste0(outdir,"presidential_election_2016.eps")
postscript(file=filename,horizontal=FALSE,width=10,height=5)
plot(data,screens=c(1,1),col=c("blue","red"))
abline(v=as.Date("2016-07-18"),col="red",lty=2)
abline(v=as.Date("2016-07-25"),col="blue",lty=2)
legend("topleft",c("Dem (Clinton)","Rep (Trump)"),
       col=c("blue","red"),
       lty=c(1,1))
dev.off()
```

With resulting plot



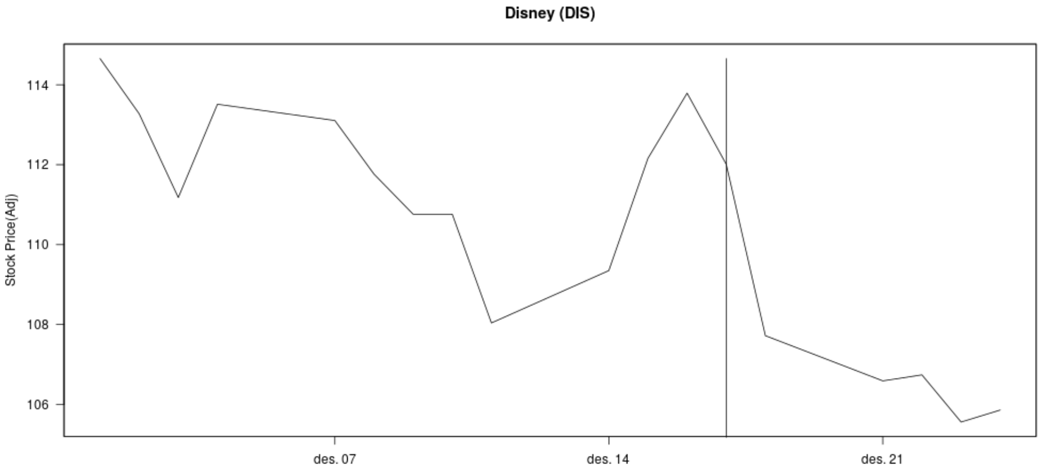
Note the vertical lines which are the Republican and Democratic conventions.

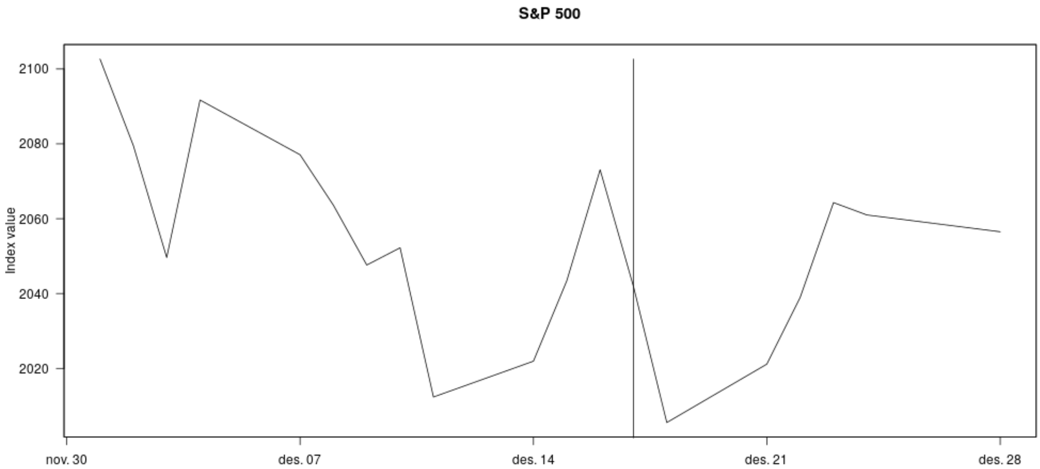
Exercise 6. *Disney and Star Wars* [2]

R code for doing plots.

```
outdir <- "../../results/2015_12_plot_disney/"

source("~/data/2015/yahoo_data/read.R")
disney15 <- window(disney,start=as.Date("2015-01-01"),end=as.Date("2015-12-31"))
disney1512 <- window(disney,start=as.Date("2015-12-01"),end=as.Date("2015-12-31"))
sp1512 <- window(sp500,start=as.Date("2015-12-01"),end=as.Date("2015-12-31"))

filename <- paste0(outdir,"sp500_dec_2015.png")
png(filename,width=1000,height=500)
plot.zoo(sp1512,main="S&P 500",xlab="time",ylab="Index value")
rect(as.Date("2015-12-17"),0,as.Date("2015-12-17"),max(sp1512))
dev.off()

plot.zoo(disney1512, main="Disney (DIS)",xlab="time",ylab="Stock Price(Adj)")
rect(as.Date("2015-12-17"),0,as.Date("2015-12-17"),max(disney1512))
```

Resulting plots



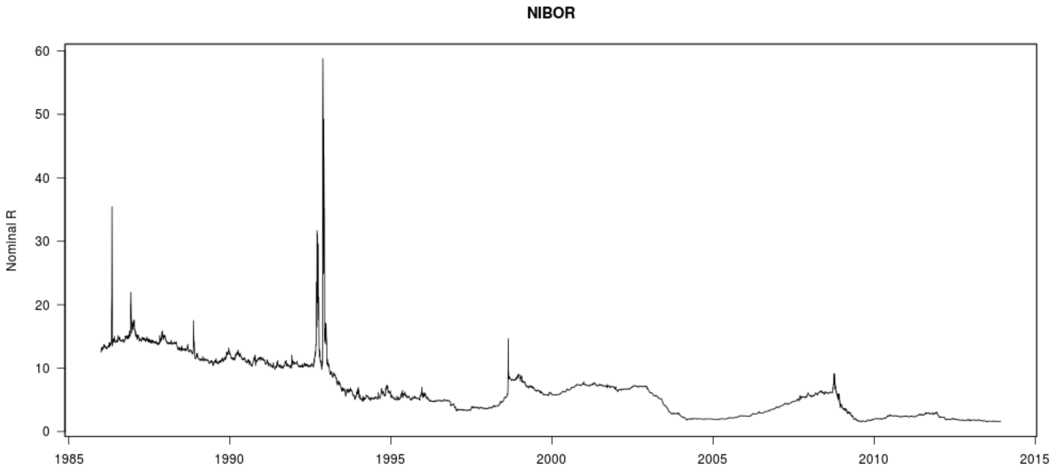


Comment: You don't necessarily expect much of a reaction, the expected performance of Star Wars will already be in the stock price. Any reaction will only happen if the film does it *unexpectedly* worse or better.

Exercise 7. *NIBOR* [5]

The Norwegian equivalent of LIBOR.

Set in a similar manner to LIBOR, as an average of quotes provided by financial institutions.



In 2013 the fixing of the NIBOR was moved to the Oslo Stock Exchange. On their homepage they provide the fixing, as well as the quotes by the member institutions.

Want to look at the components of the fixing.

There are any number of maturities to investigate. The following shows results for the one month interest rate.

Reading the data

```
indir <- "/home/bernt/data/2017/nibor/2017_01/"
filename <- paste0(indir,"NIBOR1M.csv")
nib1m1216  <- read.zoo(filename, header=TRUE, format="%d.%m.%y", sep=",")
indir <- "/home/bernt/data/2018/nibor/2018_01/"
filename <- paste0(indir,"nibor1m.csv")
nib1m17  <- read.zoo(filename, header=TRUE, format="%d.%m.%y", sep=",")
nib1m1216 <- window(nib1m1216,end=as.Date("2016-12-31"))
nib1m17  <- window(nib1m17, start=as.Date("2017-01-01"))
nib1m <- rbind(nib1m1216,nib1m17)
```

Object of interest is the difference between the fixing and the quotes provided by the six banks.

```
> ## 1 month
> diffDDB <- na.omit(as.matrix(nib1m$Fixing-nib1m$DDB))
```

```
> diffDNM <- na.omit(as.matrix(nib1m$Fixing-nib1m$DNM))
> diffNDA <- na.omit(as.matrix(nib1m$Fixing-nib1m$NDA))
> diffSEB <- na.omit(as.matrix(nib1m$Fixing-nib1m$SEB))
> diffSHB <- na.omit(as.matrix(nib1m$Fixing-nib1m$SHB))
> diffSWD <- na.omit(as.matrix(nib1m$Fixing-nib1m$SWD))
```

Test for difference, DDB (Danske Bank):

```
> t.test(diffDDB)

One Sample t-test

data:  diffDDB
t = -28.838, df = 1313, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.02829377 -0.02468949
sample estimates:
  mean of x
-0.02649163
```

Pretty clear. Lets do for all

```
> means <- c(mean(diffDDB),
+            mean(diffDNM),
+            mean(diffNDA),
+            mean(diffSEB),
+            mean(diffSHB),
+            mean(diffSWD))
> print(means)
[1] -0.026491629 -0.006012130 -0.001250948  0.005689394  0.034045455
[6] -0.005223654
> ttests <-c(t.test(diffDDB)$statistic,
+            t.test(diffDNM)$statistic,
+            t.test(diffNDA)$statistic,
+            t.test(diffSEB)$statistic,
+            t.test(diffSHB)$statistic,
+            t.test(diffSWD)$statistic)
> pvals <- c(t.test(diffDDB)$p.value,
+            t.test(diffDNM)$p.value,
+            t.test(diffNDA)$p.value,
+            t.test(diffSEB)$p.value,
+            t.test(diffSHB)$p.value,
+            t.test(diffSWD)$p.value)
```

Construct a table

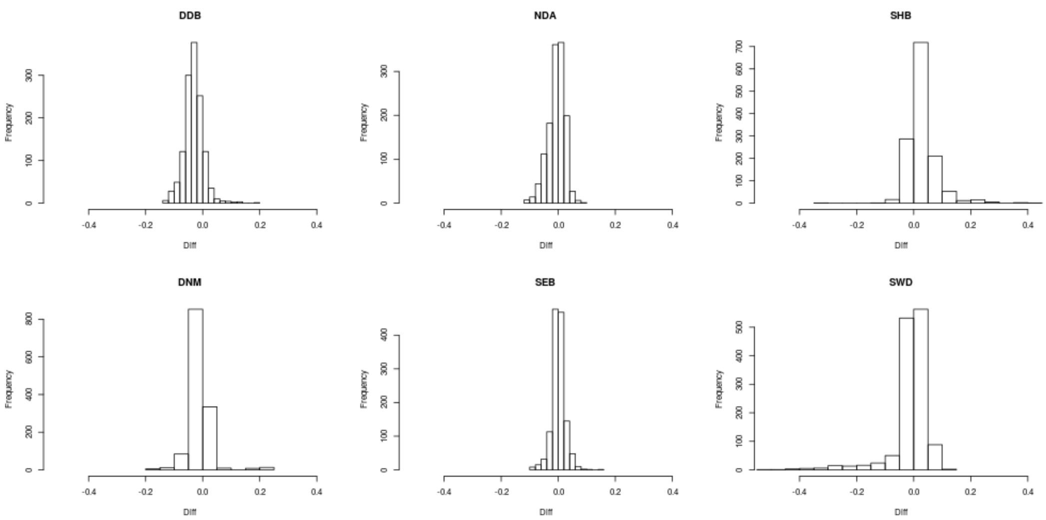
```
> tabl <- matrix(0,3,6)
> tabl[1,1:6] <- means
> tabl[2,1:6] <- ttests
> tabl[3,1:6] <- pvals
> rownames(tabl) <- c("Mean","T value","p value")
> colnames(tabl) <- c("DDB","DNM","NDA","SEB","SHB","SWD")
> filename  <- paste0(outdir,"test_differences_1_month.tex")
>
```

```
> stargazer(tabl,
+           out=filename,
+           float=FALSE,
+           title="Test Differences 1 month")
```

	DDB	DNM	NDA	SEB	SHB	SWD
Mean	-0.026	-0.006	-0.001	0.006	0.034	-0.005
T value	-28.838	-5.554	-1.545	8.694	25.078	-2.805
p value	0	0.00000	0.123	0	0	0.005

Construct histograms with same axes.

```
> alldata <-na.omit( c(diffDDB,diffDNM,diffNDA,diffSEB,diffSHB,diffSWD))
> mx <- max(alldata)
> mn <- min(alldata)
> filename <- paste0(outdir,"distribution_differences_1_month_same_axes.png")
> png(file=filename,width=1000,height=500)
> par(mfcol=c(2,3))
> hist(diffDDB,main="DDB",xlab="Diff",xlim=c(mn,mx))
> hist(diffDNM,main="DNM",xlab="Diff",xlim=c(mn,mx))
> hist(diffNDA,main="NDA",xlab="Diff",xlim=c(mn,mx))
> hist(diffSEB,main="SEB",xlab="Diff",xlim=c(mn,mx))
> hist(diffSHB,main="SHB",xlab="Diff",xlim=c(mn,mx))
> hist(diffSWD,main="SWD",xlab="Diff",xlim=c(mn,mx))
```



Doing the same estimates for subperiods

12–13:

	DDB	DNM	NDA	SEB	SHB	SWD
Mean	-0.036	-0.014	0.008	-0.001	0.030	0.020
T value	-35.704	-11.269	6.743	-0.759	15.924	15.855
p value	0	0	0	0.449	0	0

14–15:

	DDB	DNM	NDA	SEB	SHB	SWD
Mean	-0.034	-0.011	-0.010	0.009	0.028	0.017
T value	-23.713	-12.122	-7.174	11.656	21.649	8.227
p value	0	0	0	0	0	0

16–17:

	DDB	DNM	NDA	SEB	SHB	SWD
Mean	-0.013	0.004	0.002	0.006	0.043	-0.043
T value	-7.874	1.390	1.458	4.360	14.010	-11.550
p value	0	0.165	0.145	0.00002	0	0