```
### Combined code with all scripts
### Written by Jon V. Strøm and Emil B. Skei

################################################################################

### Title: Tickers.R
### Used to import tickers, clean tickers, and download SVI-data.

if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

library(readxl)
library(gtrendsR)

excel_sheets("sp500_tickers.xlsx")

sp_current <- read_excel("sp500_tickers.xlsx", sheet = 1, col_names = TRUE)
sp_changes <- read_excel("sp500_tickers.xlsx", sheet = 2, col_names = TRUE)

sp_current$`Date first added` <- as.Date.numeric(sp_current$`Date first added`, origin = "1899-12-
30")
sp_changes$Date <- as.Date(sp_changes$Date, format = "%B %d, %Y")

#subset data based on dates: 2014-01-01 - 2019-01-01

sp_changes <- subset(sp_changes, sp_changes$Date >= as.Date("2014-01-01")
                     & sp_changes$Date < as.Date("2019-01-01"))

#subset data which is needed: tickers, date inn/out
```

```
sp_current <- sp_current[c("Symbol", "Date first added")]
sp_changes <- sp_changes[c("Rem_ticker", "Date")]

#rename columns

colnames(sp_current) <- c("Ticker", "Date")
colnames(sp_changes) <- c("Ticker", "Date")

#merge data frames
merged <- rbind(sp_current, sp_changes)

#remove duplicates
length(unique(merged$Ticker))
merged = merged[!duplicated(merged$Ticker),]

#remove NAs and store only tickers in a new vector
tickers <- na.omit(merged$Ticker)

### For-loops: Connecting to GT API and downloading SVI data

tickers2 <- tickers[1:155]

SVI_1 <- list()

for (i in 1:length(tickers2)) {

  SVI_1[[i]] <- gtrends(keyword = tickers2[i], geo = "US", time = "2014-01-01 2018-12-31")

}

saveRDS(SVI_1, file = "SVI_1.rds")
```

```
#

tickers2 <- tickers[156:310]

SVI_2 <- list()

for (i in 1:length(tickers2)) {

  SVI_2[[i]] <- gtrends(keyword = tickers2[i], geo = "US", time = "2014-01-01 2018-12-31")

}

saveRDS(SVI_2, file = "SVI_2.rds")

#

tickers2 <- tickers[311:465]

SVI_3 <- list()

for (i in 1:length(tickers2)) {

  SVI_3[[i]] <- gtrends(keyword = tickers2[i], geo = "US", time = "2014-01-01 2018-12-31")

}

saveRDS(SVI_3, file = "SVI_3.rds")

#


tickers2 <- tickers[466:619]
```

```
SVI_4 <- list()

for (i in 1:length(tickers2)) {

  SVI_4[[i]] <- gtrends(keyword = tickers2[i], geo = "US", time = "2014-01-01 2018-12-31")

}

saveRDS(SVI_4, file = "SVI_4.rds")

################################################################################

### Title: SVI.R
### Used to combine data from four SVI-files into one data frame.

if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
#setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

library(readxl)
library(gtrendsR)

### Load RDS-files

SVI_1 <- readRDS("SVI_1.rds")
SVI_2 <- readRDS("SVI_2.rds")
SVI_3 <- readRDS("SVI_3.rds")
SVI_4 <- readRDS("SVI_4.rds")
```

```
### Load tickers

tickers <- read.csv(file = "tickers.csv", header = TRUE)
colnames(tickers) <- c("", "Ticker")
tickers <- as.character(tickers$Ticker)

### Combine SVI_1 data in a data frame

mylist = list()

for (i in 1:length(1:155)) {

  mylist[[i]] = SVI_1[[i]][["interest_over_time"]][["hits"]]

}

SVI_1_combined <- data.frame(SVI_1[[1]][["interest_over_time"]][["date"]])
names(SVI_1_combined) <- "Date"
SVI_1_combined[2:156] <- data.frame(matrix(unlist(mylist), ncol = 155, byrow = F),
stringsAsFactors = FALSE)
names(SVI_1_combined)[2:156] <- tickers[1:155]

### Combine SVI_2 data in a data frame

mylist = list()

for (i in 1:length(1:155)) {

  mylist[[i]] = SVI_2[[i]][["interest_over_time"]][["hits"]]

}
```

```
SVI_2_combined <- data.frame(matrix(unlist(mylist), ncol = 155, byrow = F), stringsAsFactors = FALSE)

colnames(SVI_2_combined) <- tickers[156:310]


### Combine SVI_3 data in a data frame


mylist = list()


for (i in 1:length(1:155)) {


  mylist[[i]] = SVI_3[[i]][["interest_over_time"]][["hits"]]


}


SVI_3_combined <- data.frame(matrix(unlist(mylist), ncol = 155, byrow = F), stringsAsFactors = FALSE)

colnames(SVI_3_combined) <- tickers[311:465]


### Combine SVI_4 data in a data frame


mylist = list()


for (i in 1:length(1:154)) {


  mylist[[i]] = SVI_4[[i]][["interest_over_time"]][["hits"]]


}


SVI_4_combined <- data.frame(matrix(unlist(mylist), ncol = 154, byrow = F), stringsAsFactors = FALSE)

colnames(SVI_4_combined) <- tickers[466:619]
```

```
### Combine all SVI data together

SVI_master <- data.frame(SVI_1_combined,
                         SVI_2_combined,
                         SVI_3_combined,
                         SVI_4_combined)

### Export to CSV
write.csv(SVI_master, file = "SVI_master.csv")

################################################################################

### Title: CRSP_3.R
### Extract data from "CRSP_data_3.csv" containing remaining CRSP data from 22 tickers.

if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

CRSP_data_3 <- read.csv("CRSP_data_4.csv")

### Remove tickers which are not among the 22 additional

unitick <- as.character(unique(CRSP_data_3$TICKER))

tickers <- c("UA", "MKC", "LEN", "TAP", "STZ", "CBS",
             "HLT", "KHC", "BRK", "F", "BF", "ROL", "CSC",
             "LNT", "LB", "PGR", "EQR", "MAC", "DO", "BXP",
             "WYNN", "WIN")
```

```
difftick <- setdiff(unitick, tickers)

for (i in 1:length(difftick)) {

  CRSP_data_3 <- subset(CRSP_data_3, CRSP_data_3$TICKER != difftick[i])

}

### Clean data

# Replace Dividends NAs with zeros

CRSP_data_3$DIVAMT[is.na(CRSP_data_3$DIVAMT)] <- 0

rows_with_na <- apply(CRSP_data_3, 1, function(x){any(is.na(x))})
sum(rows_with_na)
CRSP_data_3 <- CRSP_data_3[!rows_with_na,]

# Fix dates

CRSP_data_3$date <- as.Date(as.character(CRSP_data_3$date), format = "%Y%m%d")

# Export to csv

write.csv(CRSP_data_3, file = "CRSP_remaining_data_3.csv")



################################################################################

### Title: CRSP_5.R
### Clean remaining CRSP data from 22 tickers.
```

```
if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

library(readxl)

CRSP_remaining_data <- read_excel("CRSP_remaining_data_3.xlsx", sheet = 1, col_names = TRUE)

# Replace Dividends NAs with zeros

CRSP_remaining_data$DIVAMT[is.na(CRSP_remaining_data$DIVAMT)] <- 0

# Fix dates

CRSP_remaining_data$date <- as.Date(as.character(CRSP_remaining_data$date), format = "%Y-%m-%d")

# Subset data to prepare for-loop

CRSP_price <- data.frame(CRSP_remaining_data[c("date", "TICKER", "PRC")])
CRSP_vol <- data.frame(CRSP_remaining_data[c("date", "TICKER", "VOL")])
CRSP_shrout <- data.frame(CRSP_remaining_data[c("date", "TICKER", "SHROUT")])
CRSP_div <- data.frame(CRSP_remaining_data[c("date", "TICKER", "DIVAMT")])

tickers <- as.character(unique(CRSP_remaining_data$TICKER))
dates <- data.frame(as.Date(as.character(unique(CRSP_remaining_data$date))))
colnames(dates) <- c("date")

### For-loop: Prices
```

```
library(dplyr)

df_prices <- data.frame()

temp <- data.frame(subset(CRSP_price, CRSP_price$TICKER == tickers[1]))
temp <- temp[-2]
df_prices <- left_join(dates, temp, by = "date")

for (i in 1:length(tickers)) {

  temp <- data.frame(subset(CRSP_price, CRSP_price$TICKER == tickers[i+1]))
  temp <- temp[-2]
  temp2 <- left_join(dates, temp, by = "date")
  df_prices[,i+2] <- temp2$PRC

}

# Delete last column (only NAs)
df_prices <- df_prices[,-24]

# Set colnames for df_prices
tickers_colnames <- c("Dates")
tickers_colnames[2:23] <- tickers
colnames(df_prices) <- tickers_colnames

### For-loop: Volume

df_volume <- data.frame()

temp <- data.frame(subset(CRSP_vol, CRSP_vol$TICKER == tickers[1]))
temp <- temp[-2]
```

```
df_volume <- left_join(dates, temp, by = "date")

for (i in 1:length(tickers)) {

  temp <- data.frame(subset(CRSP_vol, CRSP_vol$TICKER == tickers[i+1]))
  temp <- temp[-2]
  temp2 <- left_join(dates, temp, by = "date")
  df_volume[,i+2] <- temp2$VOL

}
# Delete last column (only NAs)
df_volume <- df_volume[,-24]

# Set colnames for df_volume
colnames(df_volume) <- tickers_colnames

### For-loop: Shares outstanding

df_shrout <- data.frame()

temp <- data.frame(subset(CRSP_shrout, CRSP_shrout$TICKER == tickers[1]))
temp <- temp[-2]
df_shrout <- left_join(dates, temp, by = "date")

for (i in 1:length(tickers)) {

  temp <- data.frame(subset(CRSP_shrout, CRSP_shrout$TICKER == tickers[i+1]))
  temp <- temp[-2]
  temp2 <- left_join(dates, temp, by = "date")
  df_shrout[,i+2] <- temp2$SHROUT

}
```

```
# Delete last column (only NAs)
df_shrout <- df_shrout[,-24]

# Set colnames for df_shrout
colnames(df_shrout) <- tickers_colnames



### For-loop: Dividends

df_div <- data.frame()

temp <- data.frame(subset(CRSP_div, CRSP_div$TICKER == tickers[1]))
temp <- temp[-2]
df_div <- left_join(dates, temp, by = "date")

for (i in 1:length(tickers)) {

  temp <- data.frame(subset(CRSP_div, CRSP_div$TICKER == tickers[i+1]))
  temp <- temp[-2]
  temp2 <- left_join(dates, temp, by = "date")
  df_div[,i+2] <- temp2$DIVAMT

}
# Delete last column (only NAs)
df_div <- df_div[,-24]

# Set colnames for df_div
colnames(df_div) <- tickers_colnames

# replace NA with 0
df_div[is.na(df_div)] <- 0
```

```
### Save each data frame as its own .csv-file

write.csv(df_prices, file = "Remaining_Prices.csv")
write.csv(df_volume, file = "Remaining_Volume.csv")
write.csv(df_shrout, file = "Remaining_Shrout.csv")
write.csv(df_div, file = "Remaining_Div.csv")

################################################################################

### Title: CRSP_5.R
### Import and clean CRSP dataset. Need the following files:
## CRSP_data_final.csv
## Remaining_Prices.csv
## Remaining_Volume.csv
## Remaining_Shrout.csv
## Remaining_Div.csv
### (Can be obtained by running CRSP_5.R seperately)

if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

CRSP_data <- read.csv("CRSP_data_final.csv")

# Replace Dividends NAs with zeros

CRSP_data$DIVAMT[is.na(CRSP_data$DIVAMT)] <- 0

rows_with_na <- apply(CRSP_data, 1, function(x){any(is.na(x))})
```

```
sum(rows_with_na)
CRSP_data <- CRSP_data[!rows_with_na,]

### remove some stocks. Common among these: They are longer than 1258

rem_tick <- c("UA", "MKC", "LEN", "TAP", "STZ", "CBS",
              "HLT", "KHC", "BRK", "F", "BF", "ROL", "CSC",
              "LNT", "LB", "PGR", "EQR", "MAC", "DO", "BXP",
              "WYNN", "WIN")

for (i in 1:length(rem_tick)) {

  CRSP_data <- subset(CRSP_data, CRSP_data$TICKER != rem_tick[i])

}

# Fix dates

CRSP_data$date <- as.Date(as.character(CRSP_data$date), format = "%Y%m%d")

### Remove stocks which are not among the 619 original ones
# Import the original tickers list

tickers_svi <- read.csv("tickers.csv", header = FALSE)
tickers_svi <- as.character(tickers_svi$V1)
tickers_CRSP <- as.character(unique(CRSP_data$TICKER))

# Check for additional tickers and save these
tickers_diff <- setdiff(tickers_CRSP, tickers_svi)

# Remove additional tickers
```

```
for (i in 1:length(tickers_diff)) {

  CRSP_data <- subset(CRSP_data, CRSP_data$TICKER != tickers_diff[i])

}

# Subset data into new data frames

CRSP_price <- data.frame(CRSP_data[c("date", "TICKER", "PRC")])
CRSP_vol <- data.frame(CRSP_data[c("date", "TICKER", "VOL")])
CRSP_shrout <- data.frame(CRSP_data[c("date", "TICKER", "SHROUT")])
CRSP_div <- data.frame(CRSP_data[c("date", "TICKER", "DIVAMT")])

tickers <- as.character(unique(CRSP_data$TICKER))
dates <- data.frame(as.Date(as.character(unique(CRSP_data$date))))
colnames(dates) <- c("date")

### For-loop: Prices

library(dplyr)

df_prices <- data.frame()

temp <- data.frame(subset(CRSP_price, CRSP_price$TICKER == tickers[1]))
temp <- temp[-2]
df_prices <- left_join(dates, temp, by = "date")

for (i in 1:length(tickers)) {

  temp <- data.frame(subset(CRSP_price, CRSP_price$TICKER == tickers[i+1]))
  temp <- temp[-2]
  temp2 <- left_join(dates, temp, by = "date")
```

```
  df_prices[,i+2] <- temp2$PRC

}

#tickers[598]
#subset_temp <- subset(CRSP_data, CRSP_data$TICKER == "WIN")

# Delete last column (only NAs)
df_prices <- df_prices[,-599]

# Set colnames for df_prices
tickers_colnames <- c("Dates")
tickers_colnames[2:598] <- tickers
colnames(df_prices) <- tickers_colnames



### For-loop: Volume

df_volume <- data.frame()

temp <- data.frame(subset(CRSP_vol, CRSP_vol$TICKER == tickers[1]))
temp <- temp[-2]
df_volume <- left_join(dates, temp, by = "date")

for (i in 1:length(tickers)) {

  temp <- data.frame(subset(CRSP_vol, CRSP_vol$TICKER == tickers[i+1]))
  temp <- temp[-2]
  temp2 <- left_join(dates, temp, by = "date")
  df_volume[,i+2] <- temp2$VOL

}
```

```
# Delete last column (only NAs)
df_volume <- df_volume[,-599]

# Set colnames for df_volume
colnames(df_volume) <- tickers_colnames



### For-loop: Shares outstanding

df_shrout <- data.frame()

temp <- data.frame(subset(CRSP_shrout, CRSP_shrout$TICKER == tickers[1]))
temp <- temp[-2]
df_shrout <- left_join(dates, temp, by = "date")

for (i in 1:length(tickers)) {

  temp <- data.frame(subset(CRSP_shrout, CRSP_shrout$TICKER == tickers[i+1]))
  temp <- temp[-2]
  temp2 <- left_join(dates, temp, by = "date")
  df_shrout[,i+2] <- temp2$SHROUT

}
# Delete last column (only NAs)
df_shrout <- df_shrout[,-599]

# Set colnames for df_shrout
colnames(df_shrout) <- tickers_colnames



### For-loop: Dividends
```

```
df_div <- data.frame()

temp <- data.frame(subset(CRSP_div, CRSP_div$TICKER == tickers[1]))
temp <- temp[-2]
df_div <- left_join(dates, temp, by = "date")

for (i in 1:length(tickers)) {

  temp <- data.frame(subset(CRSP_div, CRSP_div$TICKER == tickers[i+1]))
  temp <- temp[-2]
  temp2 <- left_join(dates, temp, by = "date")
  df_div[,i+2] <- temp2$DIVAMT

}
# Delete last column (only NAs)
df_div <- df_div[,-599]

# Set colnames for df_div
colnames(df_div) <- tickers_colnames

# replace NA with 0
df_div[is.na(df_div)] <- 0

### Join data with remaining CRSP-data (22 tickers)

# Import the data and remove unwanted columns
rem_prices <- read.csv("Remaining_Prices.csv")
rem_prices <- rem_prices[-1]
rem_prices <- rem_prices[-1]

rem_vol <- read.csv("Remaining_Volume.csv")
```

```
rem_vol <- rem_vol[-1]
rem_vol <- rem_vol[-1]

rem_shrout <- read.csv("Remaining_Shrout.csv")
rem_shrout <- rem_shrout[-1]
rem_shrout <- rem_shrout[-1]

rem_div <- read.csv("Remaining_Div.csv")
rem_div <- rem_div[-1]
rem_div <- rem_div[-1]

# Join the data with existing CRSP data

Prices <- cbind(df_prices, rem_prices)
Volume <- cbind(df_volume, rem_vol)
Shrout <- cbind(df_shrout, rem_shrout)
Div <- cbind(df_div, rem_div)

# Write master data

write.csv(Prices, file = "Master_Prices.csv")
write.csv(Volume, file = "Master_Volume.csv")
write.csv(Shrout, file = "Master_Shrout.csv")
write.csv(Div, file = "Master_Div.csv")



################################################################################

### Title: Weekly_Return.R
### Create weekly returns using endpoints

if(!is.null(dev.list())) dev.off()
```

```
cat("\014")
rm(list=ls())
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

library(dplyr)

# Import price data
prices <- read.csv("Master_Prices.csv")
prices <- prices[-1]
prices$Dates <- as.Date(prices$Dates)

# Import shares outstanding data
shrout <- read.csv("Master_Shrout.csv")
shrout <- shrout[-1]
shrout$Dates <- as.Date(shrout$Dates)

# Import dividends data
div <- read.csv("Master_Div.csv")
div <- div[-1]
div$Dates <- as.Date(div$Dates)

# Create daily, total returns (formula checked in Excel)

daily_returns <- data.frame(rep(0, 1510))

for (i in 1:619) {

  share_ratio <- lead(shrout[,i+1]) / shrout[,i+1]
  temp <- (lead(prices[,i+1]) + lead(div[,i+1])) / prices[,i+1]
  temp <- temp * share_ratio
  daily_returns[,i] <- temp
```

```
}

dates <- prices$Dates
prices <- prices[,-1]
colnames(daily_returns) <- colnames(prices)

# Create xts object
dates <- dates[-1]
daily_returns <- daily_returns[-1510,]
rownames(daily_returns) <- dates
#daily_returns <- daily_returns[,-1]
daily_returns <- as.xts(daily_returns)

# Import endpoints
ep <- read.csv("End_points.csv")
ep <- ep -1
end_points <- c(0, ep$X0)

# Calculate weekly returns (TEST: ORCL)

test <- period.apply(daily_returns$ORCL, end_points, prod)
test <- test - 1
test <- test[-314,]

# Save the date vector
dates_short <- as.character(index(test))
dates_short <- as.Date(dates_short)

# Calculate weekly returns

weekly_returns <- data.frame(rep(0, 314))
```

```
for (i in 1:ncol(daily_returns)) {

  weekly_returns[,i] <- period.apply(daily_returns[,i], end_points, prod)

}

weekly_returns <- weekly_returns - 1
weekly_returns <- weekly_returns[-314,]
rownames(weekly_returns) <- dates_short
colnames(weekly_returns) <- colnames(prices)

write.csv(weekly_returns, file = "Weekly_returns.csv")

################################################################################

### Title: FamaFrench.R
### Import Fama French factors
### Need the following files to run the script:
### F-F_Research_Data_Factors_weekly.csv
### Prices_master_2.csv

if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

library(dplyr)
library(roll)

#fama_french <- read.csv("F-F_Research_Data_5_Factors_2x3_daily.csv", skip = 3)
```

```
fama_french <- read.csv("F-F_Research_Data_Factors_weekly.csv", skip = 3)

names(fama_french)[names(fama_french) == "X"] <- "Dates"

# Format as dates
fama_french$Dates <- as.Date(as.character(fama_french$Dates), format = "%Y%m%d")

# Subset based on dates
fama_french <- subset(fama_french, fama_french$Dates >= as.Date("2013-01-01")
                      & fama_french$Dates < as.Date("2019-01-01"))

# Divide by zero to obtain decimal form
fama_french$Mkt.RF <- fama_french$Mkt.RF/100
fama_french$SMB <- fama_french$SMB/100
fama_french$HML <- fama_french$HML/100
fama_french$RF <- fama_french$RF/100

### Test regression

# Import weekly returns

weeklyreturns <- read.csv("Weekly_returns.csv")
dates <- as.Date(weeklyreturns$X.1)
rownames(weeklyreturns) <- dates
weeklyreturns <- weeklyreturns[,-1]

# Make weeklyreturns into xts object

weeklyreturns <- as.xts(weeklyreturns)

#colnames(weeklyreturns) <- unique(colnames(test))
```

```
# Do not need to change for each firm

#fama_french <- fama_french[-1,]

rownames(fama_french) <- dates

fama_french <- fama_french[,-1]

FF_ind_var <- as.xts(data.frame(fama_french$Mkt.RF, fama_french$SMB, fama_french$HML), dates)


# For-loop with the rolling regression


# Create excess returns

excessreturns <- weeklyreturns - fama_french$RF


rr <- list()


for (i in 1:ncol(excessreturns)) {


  rr[[i]] <- roll_lm(FF_ind_var, excessreturns[,i], 52)

  #coeffs[,i] <- data.frame(rr$coefficients)


}


# Delist, calculate expected returns and save in data frame


alpha <- data.frame()


alpha <- as.data.frame(as.numeric(rr[[1]][["coefficients"]]$`(Intercept)`[1]))


for (j in 1:619) {


  for (i in 1:312) {


    alpha[i+1,j] <- as.data.frame(as.numeric(rr[[j]][["coefficients"]]$`(Intercept)`[i]))
```

```
  }

}


colnames(alpha) <- colnames(excessreturns)

# Delete first 52 observations and add dates as rownames

alpha <- alpha[-(1:52),]
row_names <- rownames(fama_french)
row_names <- row_names[-(1:52)]
rownames(alpha) <- row_names

write.csv(alpha, file = "Abnormal_returns.csv")

################################################################################

### Title: Compustat.R
### Import and clean Compustat dataset

if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

library(readxl)
compustat_data <- read_excel("Compustat_data.xlsx", sheet = 1, col_names = TRUE)

compustat_data <- compustat_data[c("fyear", "tic", "sale", "xad")]
```

```
# set observations from xad with NA to 0

compustat_data$xad[is.na(compustat_data$xad)] <- 0
compustat_data$sale[is.na(compustat_data$sale)] <- 0

tickers <- as.character(unique(compustat_data$tic))

### Compare tickers with original list

tickers_619 <- read.table("Tickers_final.txt")
tickers_619 <- as.character(tickers_619$V1)

tickers_diff <- setdiff(tickers_619, tickers)
# 20 tickers mising

comp_sale <- data.frame(compustat_data[c("fyear", "tic", "sale")])
comp_adv <- data.frame(compustat_data[c("fyear", "tic", "xad")])
fyear <- data.frame(unique(compustat_data$fyear))
colnames(fyear) <- c("fyear")

# For-loop: Sales

library(dplyr)

df_sale <- data.frame()

temp <- data.frame(subset(comp_sale, comp_sale$tic == tickers[1]))
temp <- temp[-2]
df_sale <- left_join(fyear, temp, by = "fyear")

for (i in 1:length(tickers)) {
```

```
  temp <- data.frame(subset(comp_sale, comp_sale$tic == tickers[i+1]))
  temp <- temp[-2]
  temp2 <- left_join(fyear, temp, by = "fyear")
  df_sale[,i+2] <- temp2$sale

}

# Delete last column (only NAs)
df_sale <- df_sale[,-601]

# Set colnames for df_sale
tickers_colnames <- c("Year")
tickers_colnames[2:600] <- tickers
colnames(df_sale) <- tickers_colnames

# For-loop: Advertisement cost

df_adv <- data.frame()

temp <- data.frame(subset(comp_adv, comp_adv$tic == tickers[1]))
temp <- temp[-2]
df_adv <- left_join(fyear, temp, by = "fyear")

for (i in 1:length(tickers)) {

  temp <- data.frame(subset(comp_adv, comp_adv$tic == tickers[i+1]))
  temp <- temp[-2]
  temp2 <- left_join(fyear, temp, by = "fyear")
  df_adv[,i+2] <- temp2$xad

}
```

```
# Delete last column (only NAs)
df_adv <- df_adv[,-601]

# Set colnames for df_adv
colnames(df_adv) <- tickers_colnames

### Final cleaning and calculation

# Delete 2012 rows
df_sale <- df_sale[-7,]
df_adv <- df_adv[-7,]

# set NA to 0
df_sale[is.na(df_sale)] <- 0
df_adv[is.na(df_adv)] <- 0

# Convert data from character to numerical

df_sale[] <- lapply(df_sale, function(x) as.numeric(x))
df_adv[] <- lapply(df_adv, function(x) as.numeric(x))

# Calculate advexp/sales

df_ratio <- data.frame()

df_ratio <- df_sale[1]

for (i in 1:length(tickers)) {

  df_ratio[,i+1] <- data.frame(df_adv[i+1] / df_sale[i+1])

}
```

```
# Set NA to 0
df_ratio[is.na(df_ratio)] <- 0

# Add missing tickers with zero values

# create for-loop to duplicate zeros vector
zeros <- data.frame(rep(0, 6))
df <- data.frame(c("2013", "2014", "2015", "2016", "2017", "2018"))

for (i in 1:length(tickers_diff)) {

  df[,i] <- zeros

}

colnames(df) <- tickers_diff

# Combine data

df_ratio <- cbind(df_ratio, df)

write.csv(df_ratio, file = "Advsales.csv")

################################################################################

### Title: Compustat.R
### Calculate abnormal turnover

if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
```

```
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

library(xts)

# Import volume data

vol <- read.csv("Master_Volume.csv")
vol <- vol[,-1]
vol$Dates <- as.Date(vol$Dates)

# Delete first observation (due to deletion from other calculations)
vol <- vol[-1,]

# Import endpoints

ep <- read.csv("End_points.csv")
ep <- ep -1
end_points <- c(0, ep$X0)

# Create weekly volume data (mean, using period.apply() and endpoints)

#weekly_vol <- period.apply(vol$ORCL, end_points, mean)

weekly_vol <- data.frame(rep(0, 314))

for (i in 1:619) {

  weekly_vol[,i] <- period.apply(vol[,i+1], end_points, mean)

}
```

```
# Delete last observation (outside the dataset)
weekly_vol <- weekly_vol[-314,]

colnames(weekly_vol) <- colnames(vol[c(2:620)])

# Create abnormal turnover (formula from Bijl et al)
# Use roll.apply()

# Subtract the mean of the last 52 observations from the weekly volume and divide by the std.dev. of the last 52 observations

rw_mean <- rollapply(weekly_vol, 52, mean)
rw_sd <- rollapply(weekly_vol, 52, sd)

# Delete last observations (outside the dataset)
rw_mean <- rw_mean[-262,]
rw_sd <- rw_sd[-262,]

# Cut the weekly volume

weekly_vol <- weekly_vol[-c(1:52),]

# Calculate abnormal turnover

abnormal_turnover <- (weekly_vol - rw_mean) / rw_sd

write.csv(abnormal_turnover, file = "Abnormal_turnover.csv")

################################################################################

### Title: ASVI.R
### Calculate ASVI
```

```
if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

### Import SVI_master

library(readxl)

SVI_master <- read_excel("SVI_master.xlsx", sheet = 1, col_names = TRUE)
SVI_master <- SVI_master[-1]

# Convert from tibble to data frame
SVI_master <- data.frame(SVI_master)

SVI_master[SVI_master == "<1"] <- 0
SVI_master[SVI_master == 0] <- 1
SVI_master[] <- lapply(SVI_master, function(x) as.integer(as.character(x)))

### ASVI calculation

ASVI <- data.frame()

for(j in 1:ncol(SVI_master))
{
  for(k in 9:nrow(SVI_master))
  {
    ASVI[k,j] <- log(SVI_master[k,j]) - log(median(SVI_master[((k-1):(k-8)),j]))
  }
}
```

```
### Rename ASVI variables

names <- colnames(SVI_master)
colnames(ASVI) <- names

### Export to CSV

write.csv(ASVI, file = "ASVI.csv")

################################################################################

### Title: MarketCap.R
### Calculate Market cap, log(Market cap) and log(Market cap) x ASVI

if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

library(readxl)
library(xts)

# Import price data
prices <- read.csv("Master_Prices.csv")
dates_long <- as.Date(prices$Dates)
prices <- prices[-1]
prices <- prices[-1]

# Import shares outstanding data
shrout <- read.csv("Master_Shrout.csv")
shrout <- shrout[-1]
```

```
shrout <- shrout[-1]

# Create market cap
market_cap <- prices * shrout
#market_cap <- log(market_cap)

# Import endpoints

end_points <- read.csv("End_points.csv")
end_points <- as.numeric(end_points$X0)
end_points <- c(0, end_points)

rownames(market_cap) <- dates_long
market_cap <- as.xts(market_cap)

weekly_market_cap <- period.apply(market_cap, end_points, mean)

weekly_market_cap <- weekly_market_cap[-(1:52),]
weekly_market_cap <- weekly_market_cap[-262,]

weekly_market_cap <- log(weekly_market_cap)
weekly_market_cap <- data.frame(weekly_market_cap)

# Import ASVI data
ASVI <- read.csv("ASVI.csv")
ASVI <- ASVI[,-1]

# Sort ASVI and weekly market cap the same way

ASVI <- ASVI[, order(colnames(ASVI))]
weekly_market_cap <- weekly_market_cap[, order(colnames(weekly_market_cap))]
```

```
# Combine with ASVI
# Assumption: The ASVI data (Sundays) is matched with Friday/last trading day data

market_cap_ASVI <- weekly_market_cap * ASVI

# Save the data

write.csv(weekly_market_cap, file = "Log_market_cap.csv")
write.csv(market_cap_ASVI, file = "Log_market_cap_ASVI.csv")

################################################################################

### Title: Num_Analysts.R
### Clean number of analysts dataset

if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

# Import data

library(readxl)

num_analysts <- read_excel("Num_Analysts.xlsx", sheet = 1, col_names = TRUE)
num_analysts$`Number of Analysts` = as.numeric(as.character(num_analysts$`Number of
Analysts`))

# Fix dates

num_analysts$Date <- as.Date(num_analysts$Date, format = "%Y%m%d")
```

```
# Subset by dates

num_analysts <- subset(num_analysts, num_analysts$Date >= as.Date("2014-01-01")
                       & num_analysts$Date < as.Date("2019-01-01"))

# Create dates vector

dates <- data.frame(as.Date(unique(num_analysts$Date)))
colnames(dates) <- "Date"

# Tickers

tickers <- as.character(unique(num_analysts$Ticker))

# For-loop

df <- data.frame(rep(0,nrow(dates)))

for (i in 1:length(tickers)) {

  temp <- data.frame(subset(num_analysts, num_analysts$Ticker == tickers[i]))
  temp <- temp[!duplicated(temp$Date),]
  temp <- temp[,-1]
  temp2 <- left_join(dates, temp, by = "Date")
  df[,i] <- temp2$Number.of.Analysts

}

df <- data.frame(dates, df)
colnames(df) <- c("Dates", tickers)

# Order by dates, increasing
```

```
df <- df[order(df$Dates, decreasing = FALSE),]

# Make endpoints

dates <- data.frame(dates[order(dates$Date, decreasing = FALSE),])
colnames(dates) <- "Date"

cut_data <- data.frame(
  df,
  cut_Date = cut(as.Date(df$Dates), "week"),
  cut_POSIXt = cut(as.POSIXct(df$Dates), "week"),
  stringsAsFactors = FALSE)

ep <- cut_data[!rev(duplicated(rev(cut_data$cut_POSIXt))),]

ep_dates <- data.frame(ep$Dates, rep(1,nrow(ep)))
colnames(ep_dates) <- c("Date", "Index")

ep_index <- left_join(dates, ep_dates, by = "Date")

# Find the relative position for each "1" in the ep_index

end_points <- which(ep_index$Index > 0)
end_points <- c(0, end_points)

# Create xts object

rownames(df) <- df[,1]
df <- df[,-1]
df <- as.xts(df)
```

```
# period.apply()

clean_data <- data.frame(rep(0, 261))

for (i in 1:ncol(df)) {

  clean_data[,i] <- period.apply(df[,i], end_points, colMeans, na.rm=TRUE)

}

clean_data <- na.locf(clean_data, na.rm = FALSE)

clean_data <- clean_data + 1
clean_data <- log(clean_data)

clean_data <- data.frame(ep_dates$Date, clean_data)
colnames(clean_data) <- c("Dates", colnames(df))

# Add missing tickers with NA for all observations
# Create for-loop to duplicate zeros vector

zeros <- data.frame(rep(NaN, nrow(clean_data)))
df2 <- data.frame(zeros)

tickers_619 <- read.table("Tickers_final.txt")
tickers_619 <- as.character(tickers_619$V1)

tickers_diff <- setdiff(tickers_619, tickers)

for (i in 1:length(tickers_diff)) {

  df2[,i] <- zeros
```

```
}

colnames(df2) <- tickers_diff

# Combine data

clean_data <- cbind(clean_data, df2)

# Save as .csv

write.csv(clean_data, file = "Log_Num_Analysts.csv")

################################################################################

### Title: MasterData.R
### Combine all data together
### 07.06.19: Removes noisy tickers if you want it to

if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

library(dplyr)
library(readxl)

# Import all the data

abnormal_returns <- read.csv("Abnormal_returns.csv")
abnormal_turnover <- read.csv("Abnormal_turnover.csv")
```

```
ASVI <- read.csv("ASVI.csv")
log_market_cap <- read.csv("Log_market_cap.csv")
market_cap_ASVI <- read.csv("Log_market_cap_ASVI.csv")
advsales <- read.csv("Advsales.csv")
log_num_analysts <- read.csv("Log_Num_Analysts.csv")
market_cap <- read.csv("Market_cap.csv")
num_analysts <- read.csv("Num_Analysts_desc.csv")

# Save date and year column

dates <- as.Date(abnormal_returns$X.1)
years <- as.numeric(advsales$Year)
years <- years[-1]

# Delete the first column for each data frame

abnormal_returns <- abnormal_returns[,-1]
abnormal_turnover <- abnormal_turnover[,-1]
ASVI <- ASVI[,-1]
log_market_cap <- log_market_cap[,-1]
market_cap_ASVI <- market_cap_ASVI[,-1]
advsales <- advsales[,-1] # index
advsales <- advsales[,-1] # years
log_num_analysts <- log_num_analysts[,-1] # index
log_num_analysts <- log_num_analysts[,-1] # years
market_cap <- market_cap[,-1]
num_analysts <- num_analysts[,-1] # index
num_analysts <- num_analysts[,-1] # years

# Delete the 2018 data for advsales

advsales <- advsales[-6,]
```

```
### Order data alphabetically

abnormal_returns <- abnormal_returns[, order(colnames(abnormal_returns))]
abnormal_turnover <- abnormal_turnover[, order(colnames(abnormal_turnover))]
ASVI <- ASVI[, order(colnames(ASVI))]
log_market_cap <- log_market_cap[, order(colnames(log_market_cap))]
market_cap_ASVI <- market_cap_ASVI[, order(colnames(market_cap_ASVI))]
advsales <- advsales[, order(colnames(advsales))]
log_num_analysts <- log_num_analysts[, order(colnames(log_num_analysts))]
market_cap <- market_cap[, order(colnames(market_cap))]
num_analysts <- num_analysts[, order(colnames(num_analysts))]

# Remove noisy tickers

#noisy <- read_excel("NoisyTickers.xlsx", sheet = 1, col_names = TRUE)
#noisy <- noisy[,-(2:3)]

#noisy <- noisy[order(noisy$Tickers),]
#noisy_tickers <- subset(noisy, noisy$Remove == 1)
#noisy_tickers <- as.character(noisy_tickers$Tickers)

# Remove companies with only NA data

rem_tick <- c("BXLT", "CPGX", "JEF", "EVRG", "BKNG", "CBRE")
#rem_tick <- c(rem_tick, noisy_tickers)

tickers <- as.character(unique(colnames(abnormal_returns)))
index <- as.numeric(rep(0, length(rem_tick)))

for (i in 1:length(rem_tick)) {
```

```
  index[i] <- as.numeric(which(tickers == rem_tick[i]))

}

abnormal_returns <- abnormal_returns[,-index]
abnormal_turnover <- abnormal_turnover[,-index]
ASVI <- ASVI[,-index]
log_market_cap <- log_market_cap[,-index]
market_cap_ASVI <- market_cap_ASVI[,-index]
advsales <- advsales[,-index]
log_num_analysts <- log_num_analysts[,-index]
market_cap <- market_cap[,-index]
num_analysts <- num_analysts[,-index]

# Update tickers vector

tickers <- as.character(colnames(abnormal_returns))

# Need to add the date vector to each data frame (and year vector to advsales)

abnormal_returns <- data.frame(dates, abnormal_returns)
advsales <- data.frame(years, advsales)

# Create Year-column for one dataset, in order to match compustat data with the rest: Year <-
as.numeric(format(date,'%Y'))

abnormal_returns$years <- as.numeric(format(abnormal_returns$dates, "%Y"))

# Combine abnormal_returns with advsales using the years column

advsales_new <- data.frame(rep(0, 261))

years_long <- data.frame(abnormal_returns$years)
```

```
colnames(years_long) <- "years"

# with noisy tickers: 1:613
# without: 1:516
for (i in 1:ncol(ASVI)) {

  temp <- data.frame(advsales[,1], advsales[,i+1])
  colnames(temp) <- c("years", "data")
  temp2 <- left_join(years_long, temp, by = "years")
  advsales_new[,i] <- temp2$data

}

colnames(advsales_new) <- tickers

# Delete date and year column

abnormal_returns <- abnormal_returns[,-1]
abnormal_returns <- abnormal_returns[,-614] #with noisy tickers: 614. Without: 517

# Create absolute abnormal returns

abs_abnormal_returns <- abs(abnormal_returns)

### Create lag_ASVI

lag_ASVI <- data.frame(rep(0,261))
for (i in 1:length(ASVI)) {lag_ASVI[,i] <- lag(ASVI[,i])}

### Create lag_log_market_cap_ASVI

lag_log_market_cap_ASVI <- data.frame(rep(0,261))
```

```
for (i in 1:length(market_cap_ASVI)) {lag_log_market_cap_ASVI[,i] <- lag(market_cap_ASVI[,i])}

### Create lag_log_market_cap

lag_log_market_cap <- data.frame(rep(0,261))
for (i in 1:length(log_market_cap)) {lag_log_market_cap[,i] <- lag(log_market_cap[,i])}

### Create lag_absolute abnormal return

lag_abs_abnormal_returns <- data.frame(rep(0,261))
for (i in 1:length(abs_abnormal_returns)) {lag_abs_abnormal_returns[,i] <-
lag(abs_abnormal_returns[,i])}

### Create lag_advsales

lag_advsales <- data.frame(rep(0,261))
for (i in 1:length(advsales_new)) {lag_advsales[,i] <- lag(advsales_new[,i])}

### Create lag_log_num_analysts

lag_log_num_analysts <- data.frame(rep(0,261))
for (i in 1:length(log_num_analysts)) {lag_log_num_analysts[,i] <- lag(log_num_analysts[,i])}

### Create lag_abnormal_turnover

lag_abnormal_turnover <- data.frame(rep(0,261))
for (i in 1:length(abnormal_turnover)) {lag_abnormal_turnover[,i] <- lag(abnormal_turnover[,i])}

###



# create one data frame before the for-loop, then rbind() the frames for each for-loop
```

```
col_names <- c("dates", "ticker", "abnormal_returns", "abs_abnormal_returns", "abnormal_turnover",
"ASVI", "log_market_cap",
        "log_market_cap_ASVI", "advsales", "log_num_analysts", "market_cap", "num_analysts",
"lag_ASVI", "lag_log_market_cap_ASVI",
        "lag_log_market_cap", "lag_abs_abnormal_returns", "lag_advsales",
"lag_log_num_analysts", "lag_abnormal_turnover")


master_data <- data.frame(
  dates,
  tickers[1],
  abnormal_returns[,1],
  abs_abnormal_returns[,1],
  abnormal_turnover[,1],
  ASVI[,1],
  log_market_cap[,1],
  market_cap_ASVI[,1],
  advsales_new[,1],
  log_num_analysts[,1],
  market_cap[,1],
  num_analysts[,1],
  lag_ASVI[,1],
  lag_log_market_cap_ASVI[,1],
  lag_log_market_cap[,1],
  lag_abs_abnormal_returns[,1],
  lag_advsales[,1],
  lag_log_num_analysts[,1],
  lag_abnormal_turnover[,1])


colnames(master_data) <- col_names


# with noisy tickers: 612
# without: 515
```

```
for (i in 1:612) {

  temp3 <- data.frame(
    dates,
    tickers[i+1],
    abnormal_returns[,i+1],
    abs_abnormal_returns[,i+1],
    abnormal_turnover[,i+1],
    ASVI[,i+1],
    log_market_cap[,i+1],
    market_cap_ASVI[,i+1],
    advsales_new[,i+1],
    log_num_analysts[,i+1],
    market_cap[,i+1],
    num_analysts[,i+1],
    lag_ASVI[,i+1],
    lag_log_market_cap_ASVI[,i+1],
    lag_log_market_cap[,i+1],
    lag_abs_abnormal_returns[,i+1],
    lag_advsales[,i+1],
    lag_log_num_analysts[,i+1],
    lag_abnormal_turnover[,i+1])

  colnames(temp3) <- col_names

  master_data <- rbind(master_data, temp3)

}

write.csv(master_data, file = "Master_data.csv")

#############################################################################
```

```
### Title: PanelDataRegressions.R
### Run panel data regressions and tests

if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

library(plm)
library(lmtest)
library(tseries)
library(sandwich)
library(CADFtest)

master_data <- read.csv("Master_data_test.csv")
master_data <- master_data[,-1]

# Remove variables we don't use
remove <- c(3, 8, 11, 12, 13, 14, 15, 16, 17, 18, 19)
master_data <- master_data[,-remove]

# Remove rowes with NA
rows_with_na <- apply(master_data, 1, function(x){any(is.na(x))})
sum(rows_with_na)
master_data <- master_data[!rows_with_na,]

# Panel regression model
pdata <- pdata.frame(master_data, index = c("ticker", "dates"))

model <- plm(ASVI ~ abs_abnormal_returns + abnormal_turnover + log_market_cap +
log_num_analysts + advsales,
```

```
        data = pdata, model = "within", effect = "time")

summary(model)

# Double-clustering formula (Thompson, 2011)
vcovDC <- function(x, ...){
  vcovHC(x, cluster = "group", ...) + vcovHC(x, cluster = "time", ...) -
    vcovHC(x, method = "white1", ...)
}

coeftest(model, vcov = function(x) vcovDC(x, type = "HC0")) # Double clustering, heteroscedasticity consistent

###



# Breusch-Pagan LM test for cross-sectional dependence in panels.
# Null: Residuals across entities are not correlated.
# Result: p-value < 2.22e-16
# Reject null and infer that there is correlation.

pcdtest(model, test = c("lm"))

# Pesaran CD test for cross-sectional dependence in panels
# Null: Residuals across entities are not correlated.
# Result: p-value < 2.2e-16
# Reject null and infer that there is correlation.

pcdtest(model, test = c("cd"))

# Breusch-Godfrey test for serial correlation.
# Null: There is no serial correlation.
# Result: p-value < 2.2e-16
```

```
# Reject null and infer that there is serial correlation.

pbgtest(model)

# Dickey-Fuller test to check for stochastic trends
# Check each variable separately
# Null: The series has a unit root (i.e. non-stationary).

adf.test(pdata$abnormal_returns, k = 1)

# Breusch-Pagan test for heteroscedasticity
# Null: Homoskedasticity.
# Result: p-value < 2.2e-16
# Reject null and infer that there is heteroscedasticity

bptest(ASVI ~ abs_abnormal_returns + abnormal_turnover + log_market_cap +
       advsales + log_num_analysts, data = pdata, studentize = F)

# Hausman test

model_fix <- plm(ASVI ~ log_abs_abnormal_returns + abnormal_turnover + log_market_cap +
                 advsales + log_num_analysts, data = pdata, model = "within", effect = "time", cluster =
"group")

model_rnd <- plm(ASVI ~ log_abs_abnormal_returns + abnormal_turnover + log_market_cap +
                 advsales + log_num_analysts, data = pdata, model = "random")

#summary(model_rnd)

phtest(model, model_rnd)

################################################################################
```

```
### Title: FamaMacBeth.R
### Run Fama-MacBeth regressions

if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

library(plm)
library(lmtest)
library(tseries)
library(sandwich)

master_data <- read.csv("Master_data_test.csv", stringsAsFactors = FALSE)
master_data <- master_data[,-1]

# Panel regression model

# Activate if you want to subset
#master_data$dates <- as.Date(master_data$dates)
#master_data <- subset(master_data, master_data$dates < as.Date("2016-07-01"))
#master_data <- subset(master_data, master_data$dates > as.Date("2016-06-30"))

#master_data_split1 <- subset(master_data, dates < "2016-07-01")
#master_data_split2 = subset(master_data, dates > "2016-06-30")

#write.csv(master_data_split1, file = "master_data_split1.csv")
#write.csv(master_data_split2, file = "master_data_split2.csv")

#Scaled without lag
master_data <- master_data %>% mutate_each_(list(~scale(.) %>% as.vector),
```

```
                                vars=c("abs_abnormal_returns","abnormal_turnover"
                                     , "ASVI", "log_market_cap", "log_market_cap_ASVI",
                                     "advsales", "num_analysts"))


#scaled with lag 1
master_data <- master_data %>% mutate_each_(list(~scale(.) %>% as.vector),
                                vars=c("lag_abs_abnormal_returns","lag_abnormal_turnover"
                                     , "lag_ASVI", "lag_log_market_cap", "lag_log_market_cap_ASVI",
                                     "lag_advsales", "lag_log_num_analysts"))

#scaled with lag 2
master_data <- master_data %>% mutate_each_(list(~scale(.) %>% as.vector),
                                vars=c("lag2_abs_abnormal_returns","lag2_abnormal_turnover"
                                     , "lag2_ASVI", "lag2_log_market_cap",
"lag2_log_market_cap_ASVI",
                                     "lag2_advsales", "lag2_log_num_analysts"))

#scaled with lag 3
master_data <- master_data %>% mutate_each_(list(~scale(.) %>% as.vector),
                                vars=c("lag3_abs_abnormal_returns","lag3_abnormal_turnover"
                                     , "lag3_ASVI", "lag3_log_market_cap",
"lag3_log_market_cap_ASVI",
                                     "lag3_advsales", "lag3_log_num_analysts"))

#scaled with lag 4
master_data <- master_data %>% mutate_each_(list(~scale(.) %>% as.vector),
                                vars=c("lag4_abs_abnormal_returns","lag4_abnormal_turnover"
                                     , "lag4_ASVI", "lag4_log_market_cap",
"lag4_log_market_cap_ASVI",
                                     "lag4_advsales", "lag4_log_num_analysts"))
```

Fama-MacBeth regressions

```
fpmg <- pmg(abnormal_returns ~ lag_abs_abnormal_returns + lag_abnormal_turnover + lag_ASVI +
lag_log_market_cap
        + lag_log_market_cap_ASVI + lag_advsales + lag_log_num_analysts, data = master_data,
index = c("dates", "ticker"), model = "dmg")

summary(fpmg)
```

CCEMG estimator regressions

```
fpcce <- pcce(abnormal_returns ~ lag_abs_abnormal_returns + lag_abnormal_turnover + lag_ASVI +
lag_log_market_cap
         + lag_log_market_cap_ASVI + lag_advsales + lag_log_num_analysts, data = master_data,
index = c("dates", "ticker"), model = "mg")

summary(fpcce)
```

##

Title: DescriptiveStatistics.R

Make descriptive statistics table

```
if(!is.null(dev.list())) dev.off()
cat("\014")
rm(list=ls())
setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")
#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")

library(psych)
library(EnvStats)
library(readxl)
```

SVI

```
# Import SVI data

SVI <- read_excel("SVI_master.xlsx", sheet = 1, col_names = TRUE)
SVI <- SVI[-1]

# Convert from tibble to data frame
SVI <- data.frame(SVI)

# Set observations < 1 to 1
SVI[SVI == "<1"] <- 0
SVI[SVI == 0] <- 1
SVI[] <- lapply(SVI, function(x) as.integer(as.character(x)))

# Create descriptive statistics

temp <- data.frame()
desc_SVI <- describe(SVI[,1])

for (i in 2:ncol(SVI)) {

  temp <- describe(SVI[,i])
  desc_SVI <- rbind(desc_SVI, temp)

}

# Find average statistics

temp <- data.frame()
desc_SVI_mean <- mean(desc_SVI[,1])
```

```
for (i in 2:ncol(desc_SVI)) {

  temp <- mean(desc_SVI[,i])
  desc_SVI_mean <- cbind(desc_SVI_mean, temp)

}

desc_SVI_mean <- data.frame(desc_SVI_mean)
colnames(desc_SVI_mean) <- colnames(desc_SVI)

#write_tableHTML(tableHTML(desc_SVI_mean), file = "SVI.html")

### ASVI

# Import ASVI data

ASVI <- read.csv("ASVI.csv")
ASVI <- ASVI[,-1]

# Create descriptive statistics

temp <- data.frame()
desc_ASVI <- describe(ASVI[,1])

for (i in 2:ncol(ASVI)) {

  temp <- describe(ASVI[,i])
  desc_ASVI <- rbind(desc_ASVI, temp)

}

# Find average statistics
```

```
temp <- data.frame()
desc_ASVI_mean <- mean(desc_ASVI[,1])

for (i in 2:ncol(desc_ASVI)) {

  temp <- mean(desc_ASVI[,i])
  desc_ASVI_mean <- cbind(desc_ASVI_mean, temp)

}

desc_ASVI_mean <- data.frame(desc_ASVI_mean)
colnames(desc_ASVI_mean) <- colnames(desc_ASVI)

### Abnormal return (with 52 week rolling regression)

ab_ret <- read.csv("Abnormal_returns_test.csv")
ab_ret <- ab_ret[,-1]

# Remove certain tickers
rem_tick <- c("BXLT", "CPGX", "JEF", "EVRG", "BKNG", "CBRE")

tickers <- as.character(unique(colnames(ab_ret)))
index <- as.numeric(rep(0, length(rem_tick)))

for (i in 1:length(rem_tick)) {

  index[i] <- as.numeric(which(tickers == rem_tick[i]))

}

ab_ret <- ab_ret[,-index]
```

```
# Locate companies with less than 52 observations

temp <- data.frame(rep(0, 619))

for (i in 1:ncol(ab_ret)) {

  temp[i,] <- length(which(!is.na(ab_ret[,i])))

}

index <- which(temp < 52)

# Create descriptive statistics

temp <- data.frame()
desc_ab_ret <- describe(ab_ret[,1])

for (i in 2:ncol(ab_ret)) {

  temp <- describe(ab_ret[,i])
  desc_ab_ret <- rbind(desc_ab_ret, temp)

}

# Remove companies with less than 52 observations

desc_ab_ret <- desc_ab_ret[-index,]

# Find average statistics

temp <- data.frame()
```

```
desc_ab_ret_mean <- mean(desc_ab_ret[,1])

for (i in 2:ncol(desc_ab_ret)) {

  temp <- mean(desc_ab_ret[,i])
  desc_ab_ret_mean <- cbind(desc_ab_ret_mean, temp)

}

desc_ab_ret_mean <- data.frame(desc_ab_ret_mean)
colnames(desc_ab_ret_mean) <- colnames(desc_ab_ret)

### Absolute abnormal return (with 52 week rolling regression)

abs_ret <- data.frame(abs(ab_ret))

# Locate companies with less than 52 observations

temp <- data.frame(rep(0, 619))

for (i in 1:ncol(abs_ret)) {

  temp[i,] <- length(which(!is.na(abs_ret[,i])))

}

index <- which(temp < 52)

# Create descriptive statistics

temp <- data.frame()
desc_abs_ret <- describe(abs_ret[,1])
```

```
for (i in 2:ncol(abs_ret)) {

  temp <- describe(abs_ret[,i])
  desc_abs_ret <- rbind(desc_abs_ret, temp)

}

# Remove companies with less than 52 observations

desc_abs_ret <- desc_abs_ret[-index,]

# Find average statistics

temp <- data.frame()
desc_abs_ret_mean <- mean(desc_abs_ret[,1])

for (i in 2:ncol(desc_abs_ret)) {

  temp <- mean(desc_abs_ret[,i])
  desc_abs_ret_mean <- cbind(desc_abs_ret_mean, temp)

}

desc_abs_ret_mean <- data.frame(desc_abs_ret_mean)
colnames(desc_abs_ret_mean) <- colnames(desc_abs_ret)

### Abnormal turnover

ab_turn <- read.csv("Abnormal_turnover.csv")
ab_turn <- ab_turn[,-1]
```

```
# Locate companies with less than 52 observations

temp <- data.frame(rep(0, 619))

for (i in 1:ncol(ab_turn)) {

  temp[i,] <- length(which(!is.na(ab_turn[,i])))

}

index <- which(temp < 52)

# Create descriptive statistics

temp <- data.frame()
desc_ab_turn <- describe(ab_turn[,1])

for (i in 2:ncol(ab_turn)) {

  temp <- describe(ab_turn[,i])
  desc_ab_turn <- rbind(desc_ab_turn, temp)

}

# Remove companies with less than 52 observations

desc_ab_turn <- desc_ab_turn[-index,]

# Find average statistics

temp <- data.frame()
desc_ab_turn_mean <- mean(desc_ab_turn[,1])
```

```
for (i in 2:ncol(desc_ab_turn)) {

  temp <- mean(desc_ab_turn[,i])
  desc_ab_turn_mean <- cbind(desc_ab_turn_mean, temp)

}

desc_ab_turn_mean <- data.frame(desc_ab_turn_mean)
colnames(desc_ab_turn_mean) <- colnames(desc_ab_turn)

### Market cap

mark_cap <- read.csv("Market_cap.csv")
mark_cap <- mark_cap[,-1]

# Locate companies with less than 52 observations

temp <- data.frame(rep(0, 619))

for (i in 1:ncol(mark_cap)) {

  temp[i,] <- length(which(!is.na(mark_cap[,i])))

}

index <- which(temp < 52)

# Create descriptive statistics

temp <- data.frame()
desc_mark_cap <- describe(mark_cap[,1])
```

```
for (i in 2:ncol(mark_cap)) {

  temp <- describe(mark_cap[,i])
  desc_mark_cap <- rbind(desc_mark_cap, temp)

}

# Remove companies with less than 52 observations

desc_mark_cap <- desc_mark_cap[-index,]

# Find average statistics

temp <- data.frame()
desc_mark_cap_mean <- mean(desc_mark_cap[,1])

for (i in 2:ncol(desc_mark_cap)) {

  temp <- mean(desc_mark_cap[,i])
  desc_mark_cap_mean <- cbind(desc_mark_cap_mean, temp)

}

desc_mark_cap_mean <- data.frame(desc_mark_cap_mean)
colnames(desc_mark_cap_mean) <- colnames(desc_mark_cap)

### Adv/sales

advsales <- read.csv("Advsales_desc.csv")
advsales <- advsales[,-1]
advsales[advsales == 0] <- NA
```

```
# Locate companies with less than 52 observations

temp <- data.frame(rep(0, 619))

for (i in 1:ncol(advsales)) {

  temp[i,] <- length(which(!is.na(advsales[,i])))

}

index <- which(temp < 52)

# Create descriptive statistics

temp <- data.frame()
desc_advsales <- describe(advsales[,1])

for (i in 2:ncol(advsales)) {

  temp <- describe(advsales[,i])
  desc_advsales <- rbind(desc_advsales, temp)

}

# Remove companies with less than 52 observations

desc_advsales <- desc_advsales[-index,]

# Remove rows with NA

rows_with_na <- apply(desc_advsales, 1, function(x){any(is.na(x))})
```

```
sum(rows_with_na)
desc_advsales <- desc_advsales[!rows_with_na,]

# Find average statistics

temp <- data.frame()
desc_advsales_mean <- mean(desc_advsales[,1])

for (i in 2:ncol(desc_advsales)) {

  temp <- mean(desc_advsales[,i])
  desc_advsales_mean <- cbind(desc_advsales_mean, temp)

}

desc_advsales_mean <- data.frame(desc_advsales_mean)
colnames(desc_advsales_mean) <- colnames(desc_advsales)

### Number of analysts

num_ana <- read.csv("Num_Analysts_desc.csv")
num_ana <- num_ana[,-1]
num_ana <- num_ana[,-1]

# Locate companies with less than 52 observations

temp <- data.frame(rep(0, 619))

for (i in 1:ncol(num_ana)) {

  temp[i,] <- length(which(!is.na(num_ana[,i])))
```

```
}

index <- which(temp < 52)

# Create descriptive statistics

temp <- data.frame()
desc_num_ana <- describe(num_ana[,1])

for (i in 2:ncol(num_ana)) {

  temp <- describe(num_ana[,i])
  desc_num_ana <- rbind(desc_num_ana, temp)

}

# Remove companies with less than 52 observations

desc_num_ana <- desc_num_ana[-index,]

# Find average statistics

temp <- data.frame()
desc_num_ana_mean <- mean(desc_num_ana[,1])

for (i in 2:ncol(desc_num_ana)) {

  temp <- mean(desc_num_ana[,i])
  desc_num_ana_mean <- cbind(desc_num_ana_mean, temp)

}
```

```
desc_num_ana_mean <- data.frame(desc_num_ana_mean)

colnames(desc_num_ana_mean) <- colnames(desc_num_ana)


### Bind the table together


desc_table <- rbind(desc_SVI_mean, desc_ASVI_mean, desc_ab_ret_mean,

                    desc_abs_ret_mean, desc_ab_turn_mean, desc_mark_cap_mean, desc_advsales_mean,
desc_num_ana_mean)



write_tableHTML(tableHTML(desc_table), file = "DescriptiveStat.html")


##############################################################################


### Title: Correlations.R

### Create correlation matrix


if(!is.null(dev.list())) dev.off()

cat("\014")

rm(list=ls())

setwd("C:/Users/A1310360/Dropbox/Dokumenter/Skole/BI/Master Thesis/Excel")

#setwd("/Users/emilbreivikasskei/Dropbox/Dokumenter/Skole/BI/Master\ Thesis/Excel/")


master_data <- read.csv("Master_data_test.csv")

master_data <- master_data[,-(1:3)]


# Sort data correctly and add logged variables


master_data_cor <- data.frame(master_data$ASVI, master_data$abnormal_returns,
master_data$abs_abnormal_return,

                              master_data$abnormal_turnover, master_data$log_market_cap,
master_data$advsales,

                              master_data$log_num_analysts)
```

```
# Calculate correlations

#cor_matrix_1 <- cor(master_data_cor, method = "pearson", use = "na.or.complete")
cor_matrix_2 <- cor(master_data_cor, method = "pearson", use = "pairwise.complete.obs")

# Write HTML-tables

#write_tableHTML(tableHTML(cor_matrix_1), file = "Cor_matrix_1.html")
write_tableHTML(tableHTML(cor_matrix_2), file = "Cor_matrix_2.html")

# end
################################################################################
```