**Please note** that the tables are in another folder named "tables", therefore you can copy paste this code in Rstudio to draw the graphs and open the different tables on the side in order observe both at the same time.

**Colour keys**:

-red:the different steps/exercises that were in the original document.

-blue:the modifications we did to the original document.

-green:the lines we used to clarify the exercises by deleting objects or values that are unnecessary for the following exercises.

============================================

library(eventstudies)

library(quantmod)

library(tidyverse)

library(tidyquant)

#======== Define your portfolio========#

#PortfolioTickers

portfolioTickers <- c("CAT","CHL","CVX","GS","CVS")

# Benchmark

benchmarkTicker <- c("^NYA")

#We choose the period for the assets, we chose 4 years before and after the event.

getSymbols(portfolioTickers,

src="yahoo",

from = "2012-11-08",

to = "2020-11-08")

getSymbols(benchmarkTicker,

src="yahoo",

from = "2012-11-08",

to = "2020-11-08")

#We merge the returns (in percentage, hence \*100) in a data base that we will convert in a #zoo object, after we replaced the missing returns by 0 (as they are 0 in mean).

returns\_m <- merge( weeklyReturn(CAT$CAT.Close,leading = TRUE)\*100,

weeklyReturn(CHL$CHL.Close,leading = TRUE)\*100,

weeklyReturn(CVX$CVX.Close,leading = TRUE)\*100,

weeklyReturn(GS$GS.Close,leading = TRUE)\*100,

weeklyReturn(CVS$CVS.Close,leading = TRUE)\*100

)

returns\_m[is.na(returns\_m)] <- 0

names(returns\_m) <- c("CAT","CHL","CVX","GS","CVS")

returns\_m <- as.zoo(returns\_m)

#========= same weights portfolio=========#

rm(benchmarkTicker)

rm(CHL)

rm(CVS)

rm(CVX)

rm(GS)

wts\_portfolio <- c(0.5,0.2,0.2,0.05,0.05)

same\_w\_portfolio <- as.zoo(t(t(returns\_m) \* wts\_portfolio), order.by = index(returns\_m))

#We define the event date

event\_df <- data.frame(name = names(returns\_m),

when = as.Date(rep("2016-11-08",5)

,"%Y-%m-%d"),

stringsAsFactors=FALSE)

#We create the event study, the event window is 26 weeks, this is 6 months before and after the event.

es\_same\_w\_portfolio <- eventstudy(firm.returns = same\_w\_portfolio,

event.list = event\_df,

event.window = 26,

type = "None",

to.remap = TRUE,

remap = "cumsum",

inference = TRUE,

inference.strategy = "bootstrap")

#We draw the graph of the event study

plot(es\_same\_w\_portfolio)

#======== same weight portfolio, market model ==========#

rm(es\_same\_w\_portfolio)

#We merge the returns of the benchmark and Caterpillar.

NYA <- merge(weeklyReturn(NYA$NYA.Close,leading = TRUE)\*100,

weeklyReturn(CAT$CAT.Close,leading = TRUE)\*100)

#We merge the returns (in percentage, hence \*100) in a data base that we will convert in a #zoo object, after we replaced the missing returns by 0 (as they are 0 in mean).

NYA[is.na(NYA)] <- 0

names(NYA) <- c("NYA","CAT")

NYA <- as.zoo(NYA)

#We do the event study with the market model (we regress with the benchmark)

rm(CAT)

es\_same\_w\_portfolio\_mm <- eventstudy(firm.returns = same\_w\_portfolio,

event.list = event\_df,

event.window = 26,

type = "marketModel",

to.remap = TRUE,

remap = "cumsum",

inference = TRUE,

inference.strategy = "bootstrap",

model.args = list(market.returns = NYA$NYA)

)

#We make the graph

plot(es\_same\_w\_portfolio\_mm)

#========== Portfolio with same weight=========#

rm(es\_same\_w\_portfolio\_mm)

#We make a matrix with the returns of the portfolio.

assetsReturns <- portfolioTickers %>%

tq\_get(get = "stock.prices",

from = "2012-11-08",

to = "2020-11-08") %>%

group\_by(symbol) %>%

tq\_transmute(select = close,

mutate\_fun = periodReturn,

period = "weekly",

col\_rename = "Return\_assets")

#We define the weights and associate them with returns associated with the respected assets.

wts\_portfolio <- c(0.5,0.2,0.2,0.05,0.05)

portfolioReturns <- assetsReturns %>%

tq\_portfolio(assets\_col = symbol,

returns\_col = Return\_assets,

weights = wts\_portfolio,

col\_rename = "Returns\_port")

#We add the dates and we define the returns in percentage

row.names(portfolioReturns) <- portfolioReturns$date

names(portfolioReturns) <- c("date","Portfolio")

portfolioReturns$date <- as.Date(portfolioReturns$date,"%Y-%m-%d")

portfolioReturns$Portfolio <- portfolioReturns$Portfolio\*100

portfolioReturns <- zoo(portfolioReturns,order.by = portfolioReturns$date)

#We define the event date in the portfolio

port\_event <- data.frame(name = "Portfolio",

when = as.Date(rep("2016-11-08",1)

,"%Y-%m-%d"),

stringsAsFactors=FALSE)

#We do the event study for the portfolio

port\_es <- eventstudy(firm.returns = portfolioReturns,

event.list = port\_event,

event.window = 26,

type = "None",

to.remap = TRUE,

remap = "cumsum",

inference = TRUE,

inference.strategy = "bootstrap")

#We make the graph

plot(port\_es)

#========= Min variance weights =========#

rm(port\_es)

#We indicate the weights of the min var portfolio(step 2), for the sake of simplicity (and mostly because the summ of the given weights were not 1), we took rounded numbers for the four first assets and we chose the last weight by sustracting the sum of the four first assets to 1.

wts\_portfolio<-c(0.014,0.294,0.280,0.34,0.072)

same\_w\_portfolio <- as.zoo(t(t(returns\_m) \* wts\_portfolio), order.by = index(returns\_m))

#We define the event

event\_df <- data.frame(name = names(returns\_m),

when = as.Date(rep("2016-11-08",5)

,"%Y-%m-%d"),

stringsAsFactors=FALSE)

#We do the eventstudy

es\_min\_var <- eventstudy(firm.returns = same\_w\_portfolio,

event.list = event\_df,

event.window = 26,

type = "None",

to.remap = TRUE,

remap = "cumsum",

inference = TRUE,

inference.strategy = "bootstrap")

#We draw the graph

plot(es\_min\_var)

#========== Min variance portfolio, market model ============#

rm(es\_min\_var)

#We do the event study with the market model as we compare it to the benchmark

es\_min\_var\_mm <- eventstudy(firm.returns = same\_w\_portfolio,

event.list = event\_df,

event.window = 26,

type = "marketModel",

to.remap = TRUE,

remap = "cumsum",

inference = TRUE,

inference.strategy = "bootstrap",

model.args = list(market.returns = NYA$NYA)

)

plot(es\_min\_var\_mm)

#========= Portfolio with min variance ========#

rm(same\_w\_portfolio)

rm(es\_min\_var\_mm)

rm(NYA)

#We build the matrix of the returns

assetsReturns <- portfolioTickers %>%

tq\_get(get = "stock.prices",

from = "2012-11-08",

to = "2020-11-08") %>%

group\_by(symbol) %>%

tq\_transmute(select = close,

mutate\_fun = periodReturn,

period = "weekly",

col\_rename = "Return\_assets")

#We define the weights and associate them with returns associated with the respected assets.

wts\_portfolio<-c(0.014,0.294,0.280,0.34,0.072)

portfolioReturns <- assetsReturns %>%

tq\_portfolio(assets\_col = symbol,

returns\_col = Return\_assets,

weights = wts\_portfolio,

col\_rename = "Returns\_port")

#We add the dates

row.names(portfolioReturns) <- portfolioReturns$date

names(portfolioReturns) <- c("date","Portfolio")

portfolioReturns$date <- as.Date(portfolioReturns$date,"%Y-%m-%d")

portfolioReturns$Portfolio <- portfolioReturns$Portfolio\*100

portfolioReturns <- zoo(portfolioReturns,order.by = portfolioReturns$date)

#We define the event

port\_event <- data.frame(name = "Portfolio",

when = as.Date(rep("2016-11-08",1)

,"%Y-%m-%d"),

stringsAsFactors=FALSE)

#We do the event study

port\_es <- eventstudy(firm.returns = portfolioReturns,

event.list = port\_event,

event.window = 26,

type = "None",

to.remap = TRUE,

remap = "cumsum",

inference = TRUE,

inference.strategy = "bootstrap")

#We draw the graph

plot(port\_es)

**Please note that the second method is useless as the first method worked ! We let them in the code just in case**.

### Method 2 - With confidence intervals

rm(portfolioTickers)

rm(port\_es)

rm(assetsReturns)

rm(portfolioReturns)

rm(port\_event)

#Same weights

wts\_portfolio <- c(0.5,0.2,0.2,0.05,0.05)

portfolio <- as.zoo(t(t(returns\_m) \* wts\_portfolio), order.by = index(returns\_m))

portfolio\_es <- eventstudy(firm.returns = portfolio,

event.list = event\_df,

event.window = 26,

type = "None",

to.remap = TRUE,

remap = "cumsum",

inference = TRUE,

inference.strategy = "bootstrap")

plot(portfolio\_es)

#min variance weights

rm(portfolio\_es)

rm(wts\_portfolio)

rm(portfolio)

wts\_portfolio\_min\_var<-c(0.014,0.294,0.280,0.34,0.072)

portfolio\_min\_var<- as.zoo(t(t(returns\_m) \* wts\_portfolio\_min\_var), order.by = index(returns\_m))

portfolio\_min\_var\_es<- eventstudy(firm.returns = portfolio\_min\_var,

event.list = event\_df,

event.window = 26,

type = "None",

to.remap = TRUE,

remap = "cumsum",

inference = TRUE,

inference.strategy = "bootstrap")

plot(portfolio\_min\_var\_es)

rm(returns\_m)

rm(wts\_portfolio\_min\_var)

rm(event\_df)

rm(portfolio\_min\_var\_es)

rm(portfolio\_min\_var)