Report

Adam Foster, Maciej Staniszewski

2024-01-20

## Approach

For the Quantitative Strategies on High Frequency Data project, two sets of strategies were developed for two groups of assets.

For group 1, the asset considered was the futures contract on the S&P500 index. Multiple combinations within single EMA and crossover EMA strategies were tested before arriving at the best variant based on in-sample data across all available quarters.

For group 2, futures contracts on gold and silver were considered. Multiple combinations within level-based and return-based volatility breakout pair trading strategies were tested. Similarly, the best variant was chosen based on in-sample data across all available quarters.

More details are provided in subsequent sections.

This report contains sufficient R code to effectively run all the models and analyis, ensuring full reproducibility of results.

## Input

The working directory and quarters were defined. All quarterly data had to be stored in a subfolder called ‘data’ within the working directory.

#setwd("C:\\Users\\afost\\Documents\\UW\\Y2\\S1\\Quantitative Strategies High Frequency Data\\hfd\_strat")  
  
selected\_quarters <- c("2021\_Q1", "2021\_Q3", "2021\_Q4",   
 "2022\_Q2", "2022\_Q4",   
 "2023\_Q1", "2023\_Q2")

## Setup

Libraries, functions were loaded and system settings were set.

library(xts)  
library(chron)  
library(TTR)  
library(tseries)  
library(knitr)  
library(kableExtra)  
library(quantmod)  
library(caTools)  
library(lubridate)  
library(dplyr)  
library(lattice)  
library(grDevices)  
library(ggplot2)  
library(cowplot)  
  
options(scipen=999)  
Sys.setlocale("LC\_TIME", "English")

## [1] ""

Sys.setenv(TZ = 'America/New\_York')  
  
par(oma=c(0,0,2,0)) # allocate space for plot margins  
  
mySR <- function(x, scale) {  
 sqrt(scale) \* mean(coredata(x), na.rm = TRUE) /   
 sd(coredata(x), na.rm = TRUE)  
}   
  
myCalmarRatio <- function(x, # x = series of returns  
 # scale parameter = Nt  
 scale) {  
 scale \* mean(coredata(x), na.rm = TRUE) /   
 maxdrawdown(cumsum(x))$maxdrawdown  
   
}  
  
positionVB\_new <- function(signal,   
 lower,   
 upper,   
 pos\_flat,   
 strategy)  
{  
 require(xts)  
   
 # lets check thevalue of the strategy parameter  
 if (! strategy %in% c("mom", "mr"))  
 { print("Strategy parameter incorrect. Please use 'mom' or 'mr'!")  
 stop  
 }  
   
 # convert inputs to simpler objects   
 signal = coredata(signal)  
 lower = coredata(lower)  
 upper = coredata(upper)  
 pos\_flat = coredata(pos\_flat)  
   
   
 # lets first create a vector of 0s  
 position <- rep(0, length(signal))  
   
 for (i in 2:length(signal))  
 {  
 if ( pos\_flat[i] == 1 ) position[i] <- 0   
 else  
 { # check if values are nonmissing (otherwise calculations not possible)  
 if (!is.na(signal[i-1]) &   
 !is.na(upper[i-1]) &   
 !is.na(lower[i-1]))  
 {   
 # what if previous position was 0  
 if (position[i-1] == 0){  
 if (signal[i-1] > upper[i-1]){position[i] <- -1}  
 if (signal[i-1] < lower[i-1]){position[i] <- 1}  
 } else if (position[i-1]==-1){  
 # what if previous position was -1  
 if (signal[i-1] > lower[i-1]){position[i] <- -1}  
 if (signal[i-1] < lower[i-1]){position[i] <- 1}  
 } else if (position[i-1]==1){  
 # what if previous position was 1  
 if (signal[i-1] < upper[i-1]){position[i] <- 1}  
 if (signal[i-1] > upper[i-1]){position[i] <- -1}  
 }  
 } else position[i] <- position[i-1]  
 # if anything is missing, keep previous position  
 }  
 }  
 # reverse the position if we use a momentum ("mom") strategy  
 if(strategy == "mom") position <- (-position)  
   
 # return() function clearly indicates   
 # what the function should return  
 return(position)  
}  
  
plotHeatmap <- function(data\_plot, # dataset (data.frame) with calculations  
 col\_vlabels, # column name with the labels for a vertical axis (string)  
 col\_hlabels, # column name with the labels for a horizontal axis (string)  
 col\_variable, # column name with the variable to show (string)  
 main, # title  
 label\_size = 6, # size of labels  
 save\_graph = FALSE, # whether to save the graph  
 width = 12,  
 height = 8,  
 file\_name = NULL) { # filename for saving  
   
 require(ggplot2)  
 require(dplyr)  
   
   
 data\_plot$labels\_ <- round(data\_plot[, col\_variable], 2)  
 data\_plot[, col\_hlabels] <- as.factor(data\_plot[, col\_hlabels])  
 data\_plot[, col\_vlabels] <- as.factor(data\_plot[, col\_vlabels])  
   
   
 p1 <- ggplot(data\_plot,   
 aes\_string(x = col\_hlabels,   
 y = col\_vlabels)) +  
 geom\_raster(aes\_string(fill = col\_variable)) +  
 theme\_bw() +  
 xlab(col\_hlabels) +  
 ylab(col\_vlabels) +  
 ggtitle(main) +  
 scale\_fill\_gradient2(low = "red",  
 high = "darkgreen",  
 mid = "white",  
 midpoint = 0) +  
 geom\_label(aes\_string(label = "labels\_"),  
 size = label\_size) +  
 theme(legend.position = "bottom",   
 legend.key.width = unit(2, "cm"))  
   
 if(save\_graph) {   
 if(is.null(file\_name)) stop("Please provide the file\_name= argument") else  
 ggsave(filename = file\_name,   
 plot = p1,   
 units = "in",  
 width = width,   
 height = height)  
 }  
   
 return(p1)  
}

## Group 1

Group 1 data was limited to the S&P500 index. NA values were inserted in the first and last 10 minutes of the trading session to avoid trading on volatile and reactive parts of the day and to focus on intraday trading.

Exponential Moving Average (EMA) was the metric of choice in the algorithm. The time series is univariate and EMA places greater emphasis on recent prices than SMA. Both single EMA and crossover EMA were tested for entry and exit. Single EMA triggered a long position whenever price exceeded the EMA and short position otherwise; crossover EMA triggered a long position whenever fast EMA exceeded slow EMA and short position otherwise. This strategy was termed the momentum strategy being in the market. A corresponding set of results was generated for the mean reversion strategy.

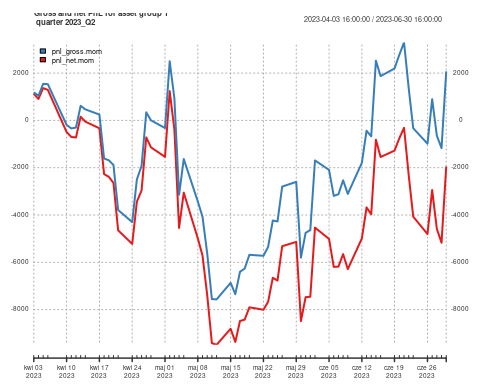
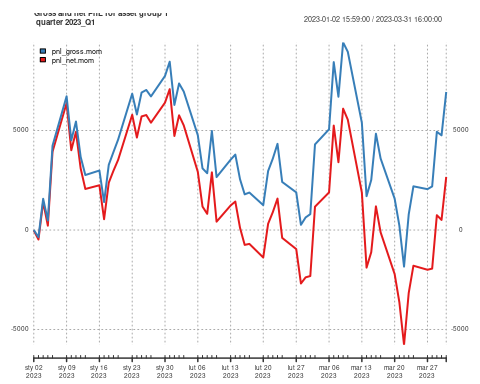
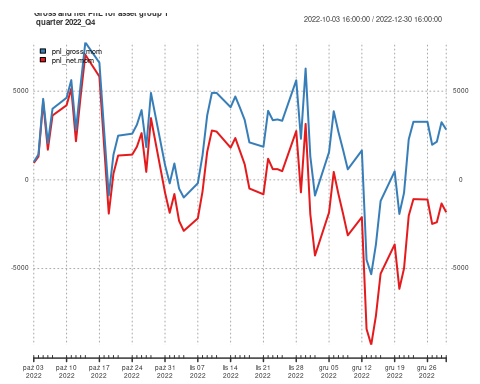
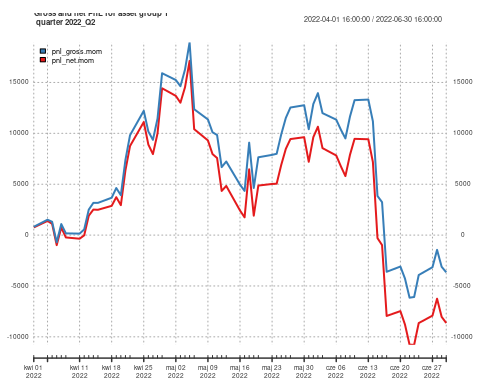
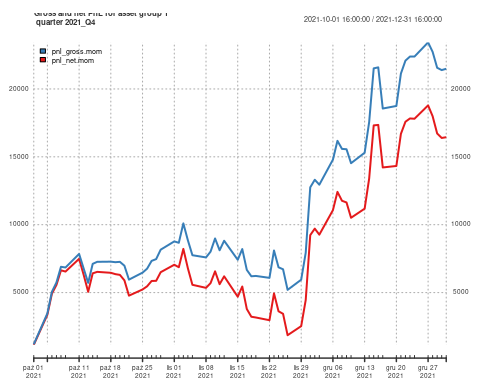
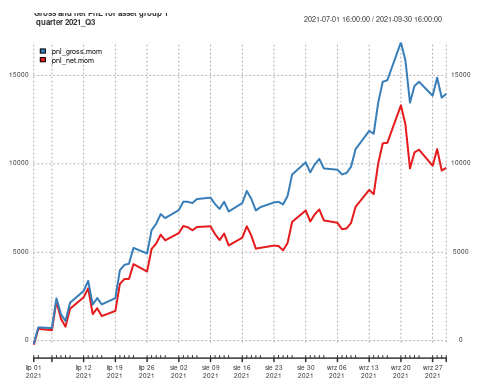
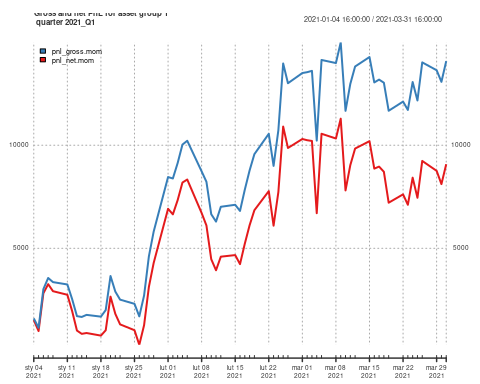
In order to determine the optimal parameters for the models, multiple combinations of EMA time horizons were tested: 10-80 days for single EMA and every combination of 10-80 days constrained by fast always having to be fewer days than slow for crossover EMA.

All positions were exited in the first 25 and last 20 minutes of the trading session and in between sessions. Any remaining missing value was populated with the previous value for completeness.

Gross P&L (accounting for point values), net P&L (reflecting transaction costs) and number of trades were calculated. The data was then aggregated to daily level and the following metrics were calculated: Sharpe ratio, Calmar ratio, average number of trades, cumulative sum of gross P&L, cumulative sum of net P&L and a final test statistic.

The entire process was repeated for all in-sample quarters.

heatmap\_list\_single <- list()  
heatmap\_list\_cross <- list()  
heatmap\_list\_single\_mr <- list()  
heatmap\_list\_cross\_mr <- list()  
sensitivities\_single <- list()  
sensitivities\_cross <- list()  
sensitivities\_single\_mr <- list()  
sensitivities\_cross\_mr <- list()  
  
for (selected\_quarter in selected\_quarters) {  
   
 message(selected\_quarter)  
   
 filename\_ <- paste0("data/data1\_", selected\_quarter, ".RData")  
 load(filename\_)  
   
 # create index of times for this quarter  
 data.group1 <- get(paste0("data1\_", selected\_quarter))  
 times\_ <- substr(index(data.group1), 12, 19)  
   
 # Keep S&P500  
 data.group1 <- data.group1[, !colnames(data.group1) %in% c("NQ")]  
   
 # the following common assumptions were defined:  
 # 1. do not use in calculations the data from the first   
 # and last 10 minutes of the session (9:31--9:40 and 15:51--16:00)  
 # – put missing values there,  
   
 # lets put missing values for these periods  
 data.group1["T09:31/T09:40",] <- NA   
 data.group1["T15:51/T16:00",] <-NA  
   
 myTheme <- chart\_theme()  
 myTheme$col$line.col <- "darkblue"  
 layout(matrix(1:1, 1, 1))  
 print(chart\_Series(data.group1$SP, theme = myTheme))  
 layout(matrix(1))  
  
  
 # Momentum  
   
 # Single EMA  
   
 EMA\_pairs = list(c(1, 10), c(1, 20), c(1, 30), c(1, 40), c(1, 50), c(1, 60), c(1, 70), c(1, 80))  
 data.group1a <- data.group1  
   
 for (pair in EMA\_pairs) {  
 # lets calculate EMAfast and EMAslow for SP  
 data.group1a$SP\_EMA <- EMA(na.locf(data.group1a$SP), pair[2])  
   
 # put missing value whenever the original price is missing  
 data.group1a$SP\_EMA[is.na(data.group1a$SP)] <- NA  
   
 # lets calculate the position for the MOMENTUM strategy  
 # if price(t-1) > MA(t-1) => pos(t) = 1 [long]  
 # if price(t-1) <= MA(t-1) => pos(t) = -1 [short]  
 # caution! this strategy is always in the market !  
 data.group1a$positionSP.mom <- ifelse(lag.xts(data.group1a$SP) >  
 lag.xts(data.group1a$SP\_EMA),  
 1, -1)  
   
 # lets apply the remaining assumptions  
 # - exit all positions 20 minutes before the session end, i.e. at 15:40  
 # - do not trade within the first 25 minutes of stocks quotations (until 9:55)  
 data.group1a$positionSP.mom[times(times\_) <= times("09:55:00") |   
 times(times\_) > times("15:40:00")] <- 0  
   
   
 # lets also fill every missing position with the previous one  
   
 data.group1a$positionSP.mom <- na.locf(data.group1a$positionSP.mom, na.rm = FALSE)  
   
 # calculating gross pnl  
   
 data.group1a$pnl\_grossSP.mom <- data.group1a$positionSP.mom \* diff.xts(data.group1a$SP) \* 50  
   
   
 # number of transactions  
 data.group1a$ntransSP.mom <- abs(diff.xts(data.group1a$positionSP.mom))  
   
 data.group1a$ntransSP.mom[1] <- 0  
   
 # net pnl  
 data.group1a$pnl\_netSP.mom <- data.group1a$pnl\_grossSP.mom -  
 data.group1a$ntransSP.mom \* 10 # 10$ per transaction  
   
 # total for strategy  
   
 data.group1a$pnl\_gross.mom <- data.group1a$pnl\_grossSP.mom  
 data.group1a$pnl\_net.mom <- data.group1a$pnl\_netSP.mom  
   
   
 # aggregate pnls and number of transactions to daily  
 my.endpoints <- endpoints(data.group1a, "days")  
   
 data.group1a.daily <- period.apply(data.group1a[,c(grep("pnl", names(data.group1a)),  
 grep("ntrans", names(data.group1a)))],  
 INDEX = my.endpoints,   
 FUN = function(x) colSums(x, na.rm = TRUE))  
   
 # summarize the strategy for this quarter  
   
 # SR  
 grossSR = mySR(x = data.group1a.daily$pnl\_gross.mom, scale = 252)  
 netSR = mySR(x = data.group1a.daily$pnl\_net.mom, scale = 252)  
 # CR  
 grossCR = myCalmarRatio(x = data.group1a.daily$pnl\_gross.mom, scale = 252)  
 netCR = myCalmarRatio(x = data.group1a.daily$pnl\_net.mom, scale = 252)  
   
 # average number of transactions  
 av.daily.ntrades = mean(data.group1a.daily$ntransSP.mom, na.rm = TRUE)  
 # PnL  
 grossPnL = sum(data.group1a.daily$pnl\_gross.mom)  
 netPnL = sum(data.group1a.daily$pnl\_net.mom)  
 # stat  
 stat = netCR \* max(0, log(abs(netPnL/1000)))  
   
 # summary of a particular strategy  
 summary\_ <- data.frame(Close = 1,  
 EMA = pair[2],  
 period = selected\_quarter, # "2016-08-16 - 2016-11",  
 gross.SR = grossSR,  
 net.SR = netSR,  
 gross.PnL = grossPnL,  
 net.PnL = netPnL,  
 av.daily.ntrans = av.daily.ntrades,  
 stringsAsFactors = FALSE)  
   
 # putting all summaries together  
   
 if(!exists("summary.pair.trading")) summary.pair.trading <- summary\_ else  
 summary.pair.trading <- rbind(summary.pair.trading, summary\_)  
   
 # deleting working files not needed any more  
 rm(grossSR, netSR, netCR,  
 grossPnL, netPnL,  
 av.daily.ntrades, stat,  
 summary\_)  
 }  
   
 # net.SR - spread av\_ratio  
 heatmap\_sr\_single <- plotHeatmap(data\_plot = summary.pair.trading, # dataset (data.frame) with calculations  
 col\_vlabels = "Close", # column name with the labels for a vertical axis (string)  
 col\_hlabels = "EMA", # column name with the labels for a horizontal axis (string)  
 col\_variable = "net.SR", # column name with the variable to show (string)  
 main = paste(selected\_quarter, "Sensitivity analysis for momentum stategy based on single EMA", sep = ": "),  
 label\_size = 3)  
   
 sensitivities\_single[[selected\_quarter]] <- summary.pair.trading  
 rm(summary.pair.trading)  
 heatmap\_list\_single[[selected\_quarter]] <- heatmap\_sr\_single  
   
   
 # EMA Crossover  
   
 EMA\_pairs = list(c(10, 20), c(10, 30), c(10, 40), c(10, 50), c(10, 60), c(10, 70), c(10, 80),  
 c(20, 30), c(20, 40), c(20, 50), c(20, 60), c(20, 70), c(20, 80),  
 c(30, 40), c(30, 50), c(30, 60), c(30, 70), c(30, 80),  
 c(40, 50), c(40, 60), c(40, 70), c(40, 80),  
 c(50, 60), c(50, 70), c(50, 80),  
 c(60, 70), c(60, 80),  
 c(70, 80))  
   
 data.group1b <- data.group1  
   
 # pair <- c(60, 70)  
 for (pair in EMA\_pairs) {  
 # lets calculate EMAfast and EMAslow for SP  
 data.group1b$SP\_EMAfast <- EMA(na.locf(data.group1b$SP), pair[1])  
 data.group1b$SP\_EMAslow <- EMA(na.locf(data.group1b$SP), pair[2])  
   
 # put missing value whenever the original price is missing  
 data.group1b$SP\_EMAfast[is.na(data.group1b$SP)] <- NA  
 data.group1b$SP\_EMAslow[is.na(data.group1b$SP)] <- NA  
   
 # lets calculate the position for the MOMENTUM strategy  
 # if fast MA(t-1) > slow MA(t-1) => pos(t) = 1 [long]  
 # if fast MA(t-1) <= slow MA(t-1) => pos(t) = -1 [short]  
 # caution! this strategy is always in the market !  
 data.group1b$positionSP.mom <- ifelse(lag.xts(data.group1b$SP\_EMAfast) >  
 lag.xts(data.group1b$SP\_EMAslow),  
 1, -1)  
   
 # lets apply the remaining assumptions  
 # - exit all positions 20 minutes before the session end, i.e. at 15:40  
 # - do not trade within the first 25 minutes of stocks quotations (until 9:55)  
 data.group1b$positionSP.mom[times(times\_) <= times("09:55:00") |   
 times(times\_) > times("15:40:00")] <- 0  
   
   
 # lets also fill every missing position with the previous one  
   
 data.group1b$positionSP.mom <- na.locf(data.group1b$positionSP.mom, na.rm = FALSE)  
   
 # calculating gross pnl  
   
 data.group1b$pnl\_grossSP.mom <- data.group1b$positionSP.mom \* diff.xts(data.group1b$SP) \* 50  
   
   
 # number of transactions  
 data.group1b$ntransSP.mom <- abs(diff.xts(data.group1b$positionSP.mom))  
   
 data.group1b$ntransSP.mom[1] <- 0  
   
 # net pnl  
 data.group1b$pnl\_netSP.mom <- data.group1b$pnl\_grossSP.mom -  
 data.group1b$ntransSP.mom \* 10 # 10$ per transaction  
   
 # total for strategy  
   
 data.group1b$pnl\_gross.mom <- data.group1b$pnl\_grossSP.mom  
 data.group1b$pnl\_net.mom <- data.group1b$pnl\_netSP.mom  
   
   
 # aggregate pnls and number of transactions to daily  
 my.endpoints <- endpoints(data.group1b, "days")  
   
 data.group1b.daily <- period.apply(data.group1b[,c(grep("pnl", names(data.group1b)),  
 grep("ntrans", names(data.group1b)))],  
 INDEX = my.endpoints,   
 FUN = function(x) colSums(x, na.rm = TRUE))  
   
 # summarize the strategy for this quarter  
   
 # SR  
 grossSR = mySR(x = data.group1b.daily$pnl\_gross.mom, scale = 252)  
 netSR = mySR(x = data.group1b.daily$pnl\_net.mom, scale = 252)  
 # CR  
 grossCR = myCalmarRatio(x = data.group1b.daily$pnl\_gross.mom, scale = 252)  
 netCR = myCalmarRatio(x = data.group1b.daily$pnl\_net.mom, scale = 252)  
   
 # average number of transactions  
 av.daily.ntrades = mean(data.group1b.daily$ntransSP.mom, na.rm = TRUE)  
 # PnL  
 grossPnL = sum(data.group1b.daily$pnl\_gross.mom)  
 netPnL = sum(data.group1b.daily$pnl\_net.mom)  
 # stat  
 stat = netCR \* max(0, log(abs(netPnL/1000)))  
   
 #selected\_quarter = "2021\_Q1"  
 #pair <- c(60, 70)  
 # collecting all statistics for a particular quarter  
 if(pair[1] == 60 & pair[2] == 70) {  
 quarter\_stats <- data.frame(quarter = selected\_quarter,  
 assets.group = 1,  
 gross.SR = grossSR,  
 net.SR = netSR,  
 gross.CR = grossCR,  
 net.CR = netCR,  
 gross.PnL = grossPnL,  
 net.PnL = netPnL,  
 av.daily.ntrans = av.daily.ntrades,  
 stat,  
 stringsAsFactors = FALSE  
 )  
   
 # collect summaries for all quarters  
 if(!exists("quarter\_stats.all.group1")) quarter\_stats.all.group1 <- quarter\_stats else  
 quarter\_stats.all.group1 <- rbind(quarter\_stats.all.group1, quarter\_stats)  
   
 # create a plot of gros and net pnl and save it to png file  
 print( # when plotting in a loop you have to use print()  
 plot(cbind(cumsum(data.group1b.daily$pnl\_gross.mom),  
 cumsum(data.group1b.daily$pnl\_net.mom)),  
 multi.panel = FALSE,  
 main = paste0("Gross and net PnL for asset group 1 \n quarter ", selected\_quarter),   
 col = c("#377EB8", "#E41A1C"),  
 major.ticks = "weeks",   
 grid.ticks.on = "weeks",  
 grid.ticks.lty = 3,  
 legend.loc = "topleft",  
 cex = 0.3)  
 )  
  
 # remove all unneeded objects for group 1  
 rm(pnl.gross.d, pnl.net.d, quarter\_stats)  
   
 gc()  
 }  
   
 # summary of a particular strategy  
 summary\_ <- data.frame(EMA.fast = pair[1],  
 EMA.slow = pair[2],  
 period = selected\_quarter, # "2016-08-16 - 2016-11",  
 gross.SR = grossSR,  
 net.SR = netSR,  
 gross.PnL = grossPnL,  
 net.PnL = netPnL,  
 av.daily.ntrans = av.daily.ntrades,  
 stringsAsFactors = FALSE)  
   
 # putting all summaries together  
   
 if(!exists("summary.pair.trading")) summary.pair.trading <- summary\_ else  
 summary.pair.trading <- rbind(summary.pair.trading, summary\_)  
   
 # deleting working files not needed any more  
 rm(grossSR, netSR, netCR,  
 grossPnL, netPnL,  
 av.daily.ntrades, stat,  
 summary\_)  
 }  
   
 # net.SR - spread av\_ratio  
 heatmap\_sr\_cross <- plotHeatmap(data\_plot = summary.pair.trading, # dataset (data.frame) with calculations  
 col\_vlabels = "EMA.fast", # column name with the labels for a vertical axis (string)  
 col\_hlabels = "EMA.slow", # column name with the labels for a horizontal axis (string)  
 col\_variable = "net.SR", # column name with the variable to show (string)  
 main = paste(selected\_quarter, "Sensitivity analysis for momentum stategy based on EMA crossover", sep = ": "),  
 label\_size = 3)  
   
 sensitivities\_cross[[selected\_quarter]] <- summary.pair.trading  
 rm(summary.pair.trading)  
 heatmap\_list\_cross[[selected\_quarter]] <- heatmap\_sr\_cross  
   
   
 # Mean reversion  
   
 # Single EMA  
   
 EMA\_pairs = list(c(1, 10), c(1, 20), c(1, 30), c(1, 40), c(1, 50), c(1, 60), c(1, 70), c(1, 80))  
 data.group1a\_mr <- data.group1  
   
 for (pair in EMA\_pairs) {  
 # lets calculate EMAfast and EMAslow for SP  
 data.group1a\_mr$SP\_EMA <- EMA(na.locf(data.group1a\_mr$SP), pair[2])  
   
 # put missing value whenever the original price is missing  
 data.group1a\_mr$SP\_EMA[is.na(data.group1a\_mr$SP)] <- NA  
   
 # lets calculate the position for the MOMENTUM strategy  
 # if price(t-1) > MA(t-1) => pos(t) = 1 [long]  
 # if price(t-1) <= MA(t-1) => pos(t) = -1 [short]  
 # caution! this strategy is always in the market !  
 data.group1a\_mr$positionSP.mr <- ifelse(lag.xts(data.group1a\_mr$SP) >  
 lag.xts(data.group1a\_mr$SP\_EMA),  
 -1, 1)  
   
 # lets apply the remaining assumptions  
 # - exit all positions 20 minutes before the session end, i.e. at 15:40  
 # - do not trade within the first 25 minutes of stocks quotations (until 9:55)  
 data.group1a\_mr$positionSP.mr[times(times\_) <= times("09:55:00") |   
 times(times\_) > times("15:40:00")] <- 0  
   
   
 # lets also fill every missing position with the previous one  
   
 data.group1a\_mr$positionSP.mr <- na.locf(data.group1a\_mr$positionSP.mr, na.rm = FALSE)  
   
 # calculating gross pnl  
   
 data.group1a\_mr$pnl\_grossSP.mr <- data.group1a\_mr$positionSP.mr \* diff.xts(data.group1a\_mr$SP) \* 50  
   
   
 # number of transactions  
 data.group1a\_mr$ntransSP.mr <- abs(diff.xts(data.group1a\_mr$positionSP.mr))  
   
 data.group1a\_mr$ntransSP.mr[1] <- 0  
   
 # net pnl  
 data.group1a\_mr$pnl\_netSP.mr <- data.group1a\_mr$pnl\_grossSP.mr -  
 data.group1a\_mr$ntransSP.mr \* 10 # 10$ per transaction  
   
 # total for strategy  
   
 data.group1a\_mr$pnl\_gross.mr <- data.group1a\_mr$pnl\_grossSP.mr  
 data.group1a\_mr$pnl\_net.mr <- data.group1a\_mr$pnl\_netSP.mr  
   
   
 # aggregate pnls and number of transactions to daily  
 my.endpoints <- endpoints(data.group1a\_mr, "days")  
   
 data.group1a\_mr.daily <- period.apply(data.group1a\_mr[,c(grep("pnl", names(data.group1a\_mr)),  
 grep("ntrans", names(data.group1a\_mr)))],  
 INDEX = my.endpoints,   
 FUN = function(x) colSums(x, na.rm = TRUE))  
   
 # summarize the strategy for this quarter  
   
 # SR  
 grossSR = mySR(x = data.group1a\_mr.daily$pnl\_gross.mr, scale = 252)  
 netSR = mySR(x = data.group1a\_mr.daily$pnl\_net.mr, scale = 252)  
 # CR  
 grossCR = myCalmarRatio(x = data.group1a\_mr.daily$pnl\_gross.mr, scale = 252)  
 netCR = myCalmarRatio(x = data.group1a\_mr.daily$pnl\_net.mr, scale = 252)  
   
 # average number of transactions  
 av.daily.ntrades = mean(data.group1a\_mr.daily$ntransSP.mr, na.rm = TRUE)  
 # PnL  
 grossPnL = sum(data.group1a\_mr.daily$pnl\_gross.mr)  
 netPnL = sum(data.group1a\_mr.daily$pnl\_net.mr)  
 # stat  
 stat = netCR \* max(0, log(abs(netPnL/1000)))  
   
 # summary of a particular strategy  
 summary\_ <- data.frame(Close = 1,  
 EMA = pair[2],  
 period = selected\_quarter, # "2016-08-16 - 2016-11",  
 gross.SR = grossSR,  
 net.SR = netSR,  
 gross.PnL = grossPnL,  
 net.PnL = netPnL,  
 av.daily.ntrans = av.daily.ntrades,  
 stringsAsFactors = FALSE)  
   
 # putting all summaries together  
   
 if(!exists("summary.pair.trading")) summary.pair.trading <- summary\_ else  
 summary.pair.trading <- rbind(summary.pair.trading, summary\_)  
   
 # deleting working files not needed any more  
 rm(grossSR, netSR, netCR,  
 grossPnL, netPnL,  
 av.daily.ntrades, stat,  
 summary\_)  
 }  
   
 # net.SR - spread av\_ratio  
 heatmap\_sr\_single\_mr <- plotHeatmap(data\_plot = summary.pair.trading, # dataset (data.frame) with calculations  
 col\_vlabels = "Close", # column name with the labels for a vertical axis (string)  
 col\_hlabels = "EMA", # column name with the labels for a horizontal axis (string)  
 col\_variable = "net.SR", # column name with the variable to show (string)  
 main = paste(selected\_quarter, "Sensitivity analysis for mean reversion stategy based on single EMA", sep = ": "),  
 label\_size = 3)  
   
 sensitivities\_single\_mr[[selected\_quarter]] <- summary.pair.trading  
 rm(summary.pair.trading)  
 heatmap\_list\_single\_mr[[selected\_quarter]] <- heatmap\_sr\_single\_mr  
   
   
 # EMA Crossover  
   
 EMA\_pairs = list(c(10, 20), c(10, 30), c(10, 40), c(10, 50), c(10, 60), c(10, 70), c(10, 80),  
 c(20, 30), c(20, 40), c(20, 50), c(20, 60), c(20, 70), c(20, 80),  
 c(30, 40), c(30, 50), c(30, 60), c(30, 70), c(30, 80),  
 c(40, 50), c(40, 60), c(40, 70), c(40, 80),  
 c(50, 60), c(50, 70), c(50, 80),  
 c(60, 70), c(60, 80),  
 c(70, 80))  
   
 data.group1b\_mr <- data.group1  
   
 for (pair in EMA\_pairs) {  
 # lets calculate EMAfast and EMAslow for SP  
 data.group1b\_mr$SP\_EMAfast <- EMA(na.locf(data.group1b\_mr$SP), pair[1])  
 data.group1b\_mr$SP\_EMAslow <- EMA(na.locf(data.group1b\_mr$SP), pair[2])  
   
 # put missing value whenever the original price is missing  
 data.group1b\_mr$SP\_EMAfast[is.na(data.group1b\_mr$SP)] <- NA  
 data.group1b\_mr$SP\_EMAslow[is.na(data.group1b\_mr$SP)] <- NA  
   
 # lets calculate the position for the MOMENTUM strategy  
 # if fast MA(t-1) > slow MA(t-1) => pos(t) = 1 [long]  
 # if fast MA(t-1) <= slow MA(t-1) => pos(t) = -1 [short]  
 # caution! this strategy is always in the market !  
 data.group1b\_mr$positionSP.mr <- ifelse(lag.xts(data.group1b\_mr$SP\_EMAfast) >  
 lag.xts(data.group1b\_mr$SP\_EMAslow),  
 -1, 1)  
   
 # lets apply the remaining assumptions  
 # - exit all positions 20 minutes before the session end, i.e. at 15:40  
 # - do not trade within the first 25 minutes of stocks quotations (until 9:55)  
 data.group1b\_mr$positionSP.mr[times(times\_) <= times("09:55:00") |   
 times(times\_) > times("15:40:00")] <- 0  
   
   
 # lets also fill every missing position with the previous one  
   
 data.group1b\_mr$positionSP.mr <- na.locf(data.group1b\_mr$positionSP.mr, na.rm = FALSE)  
   
 # calculating gross pnl  
   
 data.group1b\_mr$pnl\_grossSP.mr <- data.group1b\_mr$positionSP.mr \* diff.xts(data.group1b\_mr$SP) \* 50  
   
   
 # number of transactions  
 data.group1b\_mr$ntransSP.mr <- abs(diff.xts(data.group1b\_mr$positionSP.mr))  
   
 data.group1b\_mr$ntransSP.mr[1] <- 0  
   
 # net pnl  
 data.group1b\_mr$pnl\_netSP.mr <- data.group1b\_mr$pnl\_grossSP.mr -  
 data.group1b\_mr$ntransSP.mr \* 10 # 10$ per transaction  
   
 # total for strategy  
   
 data.group1b\_mr$pnl\_gross.mr <- data.group1b\_mr$pnl\_grossSP.mr  
 data.group1b\_mr$pnl\_net.mr <- data.group1b\_mr$pnl\_netSP.mr  
   
   
 # aggregate pnls and number of transactions to daily  
 my.endpoints <- endpoints(data.group1b\_mr, "days")  
   
 data.group1b\_mr.daily <- period.apply(data.group1b\_mr[,c(grep("pnl", names(data.group1b\_mr)),  
 grep("ntrans", names(data.group1b\_mr)))],  
 INDEX = my.endpoints,   
 FUN = function(x) colSums(x, na.rm = TRUE))  
   
 # summarize the strategy for this quarter  
   
 # SR  
 grossSR = mySR(x = data.group1b\_mr.daily$pnl\_gross.mr, scale = 252)  
 netSR = mySR(x = data.group1b\_mr.daily$pnl\_net.mr, scale = 252)  
 # CR  
 grossCR = myCalmarRatio(x = data.group1b\_mr.daily$pnl\_gross.mr, scale = 252)  
 netCR = myCalmarRatio(x = data.group1b\_mr.daily$pnl\_net.mr, scale = 252)  
   
 # average number of transactions  
 av.daily.ntrades = mean(data.group1b\_mr.daily$ntransSP.mr, na.rm = TRUE)  
 # PnL  
 grossPnL = sum(data.group1b\_mr.daily$pnl\_gross.mr)  
 netPnL = sum(data.group1b\_mr.daily$pnl\_net.mr)  
 # stat  
 stat = netCR \* max(0, log(abs(netPnL/1000)))  
   
 # summary of a particular strategy  
 summary\_ <- data.frame(EMA.fast = pair[1],  
 EMA.slow = pair[2],  
 period = selected\_quarter, # "2016-08-16 - 2016-11",  
 gross.SR = grossSR,  
 net.SR = netSR,  
 gross.PnL = grossPnL,  
 net.PnL = netPnL,  
 av.daily.ntrans = av.daily.ntrades,  
 stringsAsFactors = FALSE)  
   
 # putting all summaries together  
   
 if(!exists("summary.pair.trading")) summary.pair.trading <- summary\_ else  
 summary.pair.trading <- rbind(summary.pair.trading, summary\_)  
   
 # deleting working files not needed any more  
 rm(grossSR, netSR, netCR,  
 grossPnL, netPnL,  
 av.daily.ntrades, stat,  
 summary\_)  
 }  
   
 # net.SR - spread av\_ratio  
 heatmap\_sr\_cross\_mr <- plotHeatmap(data\_plot = summary.pair.trading, # dataset (data.frame) with calculations  
 col\_vlabels = "EMA.fast", # column name with the labels for a vertical axis (string)  
 col\_hlabels = "EMA.slow", # column name with the labels for a horizontal axis (string)  
 col\_variable = "net.SR", # column name with the variable to show (string)  
 main = paste(selected\_quarter, "Sensitivity analysis for mean reversion stategy based on EMA crossover", sep = ": "),  
 label\_size = 3)  
   
 sensitivities\_cross\_mr[[selected\_quarter]] <- summary.pair.trading  
 rm(summary.pair.trading)  
 heatmap\_list\_cross\_mr[[selected\_quarter]] <- heatmap\_sr\_cross\_mr  
}



The results were depicted using heatmaps containing Sharpe ratios, considering the return per unit of volatility. The largest value indicated which parameter combination generated the best performance on the in-sample data.

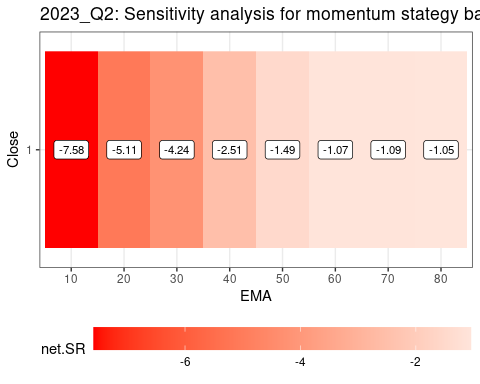
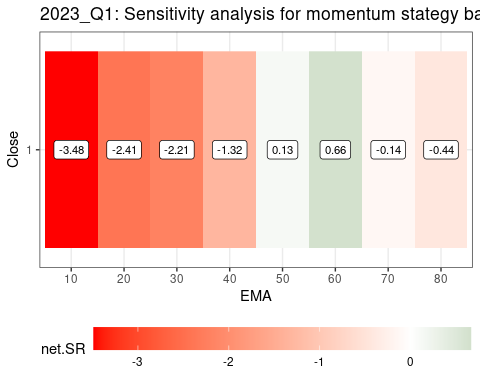
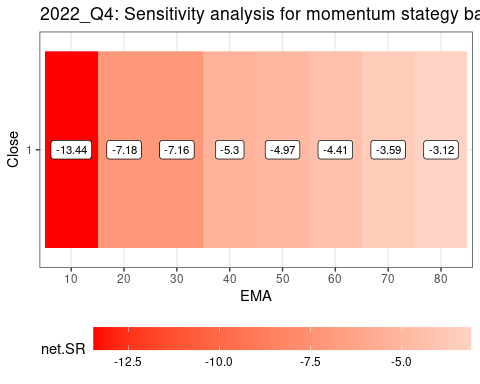
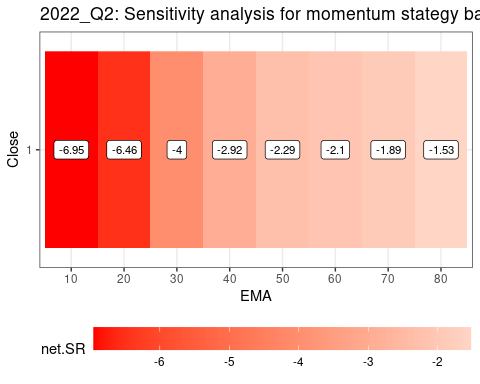
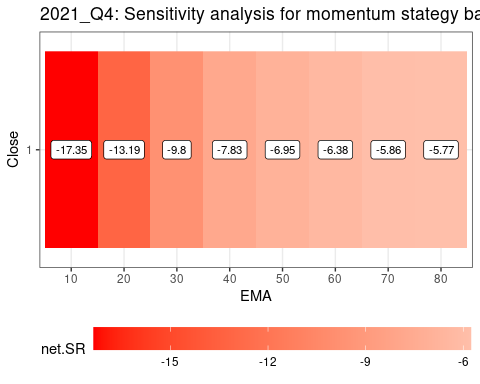
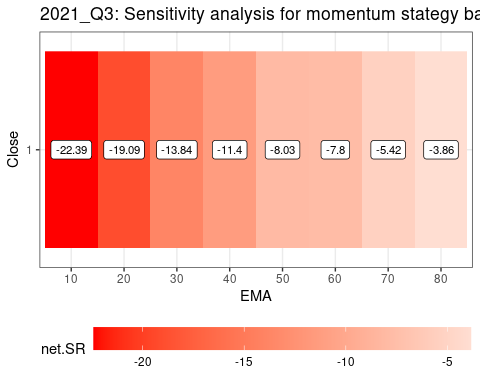
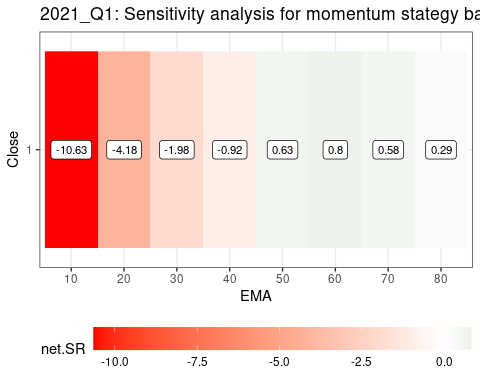
The optimal strategy was crossover EMA where fast EMA was 60 days and slow EMA was 70 days. The P&L graphs above display the performance of this particular strategy.

Heatmaps supporting the choice of strategy are shown below. Quarterly outputs, as well as the mean across quarters are shown.

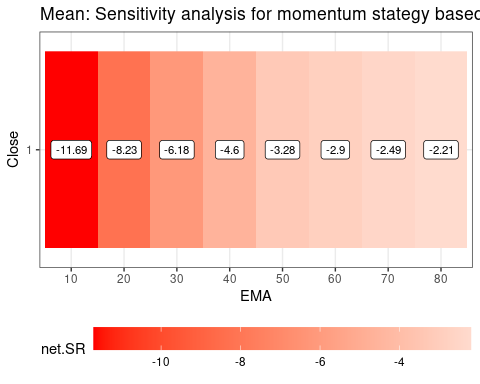
### Single EMA Momentum

Single EMA momentum strategies performed poorly. All combinations yielded negative Sharpe ratios, especially at fewer days.

# Combined  
for (output in heatmap\_list\_single) {  
 print(output)  
}



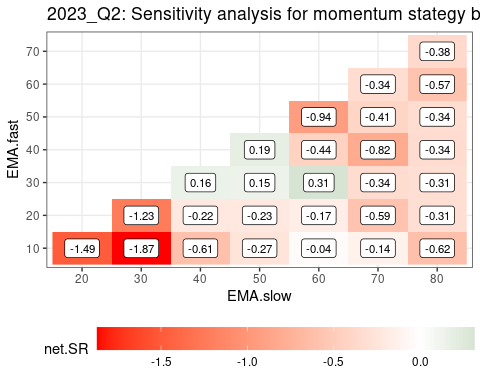
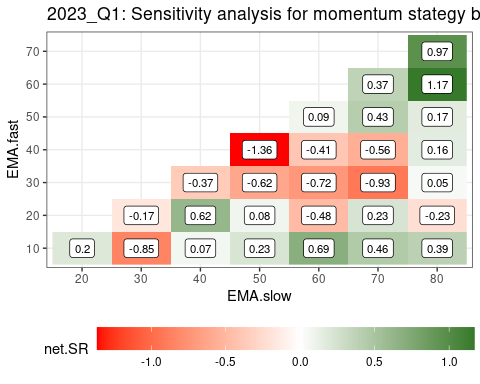
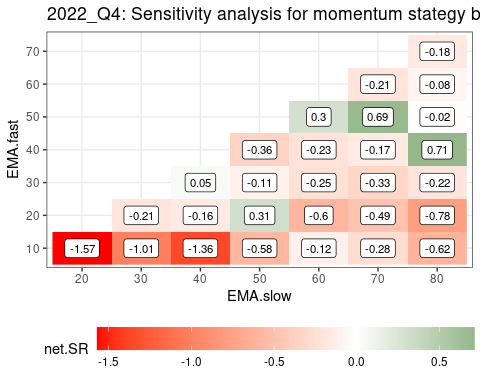
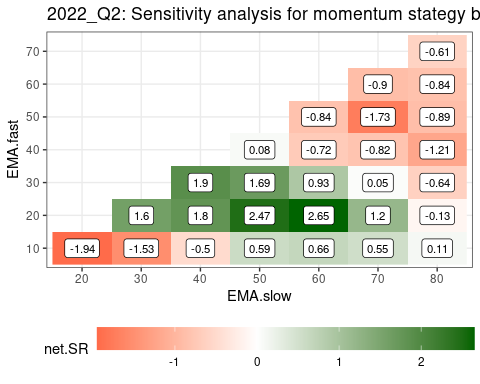
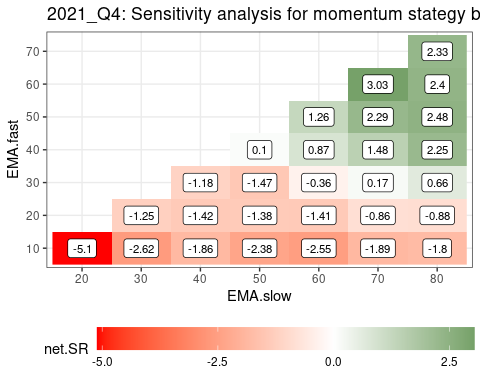
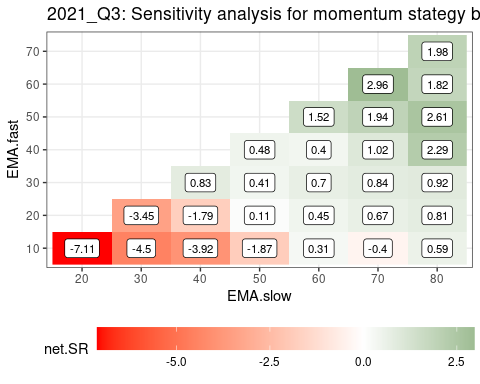
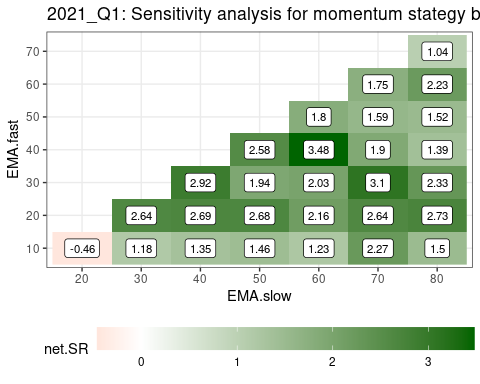
# Mean  
net\_srs\_single <- list()  
  
for(i in 1:length(sensitivities\_single)) {  
 net\_srs\_single[[i]] <- as.list(sensitivities\_single[[i]][c("net.SR")])[[1]]  
}  
  
average\_net\_sr\_single <- sapply(seq\_along(net\_srs\_single[[1]]), function(i) {  
 mean(sapply(net\_srs\_single, function(x) x[[i]]))  
})  
average\_net\_sr\_single <- data.frame(net.SR = average\_net\_sr\_single)  
  
sensitivities\_average\_single <- sensitivities\_single[[1]][c("Close", "EMA")]  
sensitivities\_average\_single <- cbind(sensitivities\_average\_single, "net.SR" = average\_net\_sr\_single)  
  
heatmap\_sr\_mean\_single <- plotHeatmap(data\_plot = sensitivities\_average\_single, # dataset (data.frame) with calculations  
 col\_vlabels = "Close", # column name with the labels for a vertical axis (string)  
 col\_hlabels = "EMA", # column name with the labels for a horizontal axis (string)  
 col\_variable = "net.SR", # column name with the variable to show (string)  
 main = paste("Mean", "Sensitivity analysis for momentum stategy based on single EMA", sep = ": "),  
 label\_size = 3)  
  
heatmap\_sr\_mean\_single



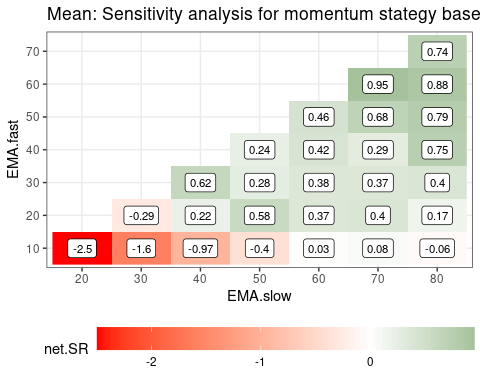
### Crossover EMA Momentum

Crossover EMA momentum strategies performed well. Most combinations produced positive Sharpe ratios with a clear tendency to outperform at longer day combinations. The best strategy was 60-70 with a Sharpe ratio of 0.95 and was selected as the strategy for group 1.

# Combined  
for (output in heatmap\_list\_cross) {  
 print(output)  
}



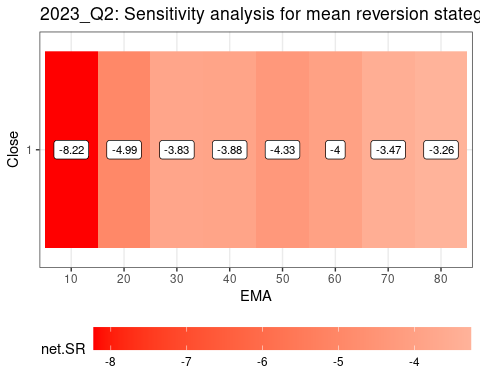
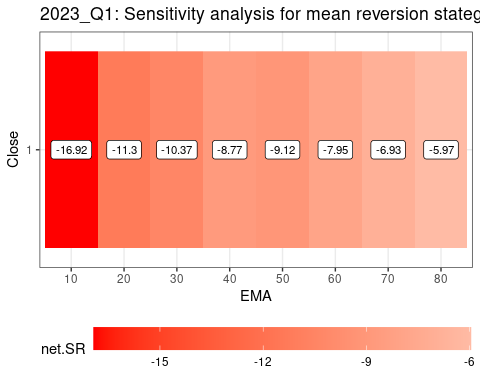
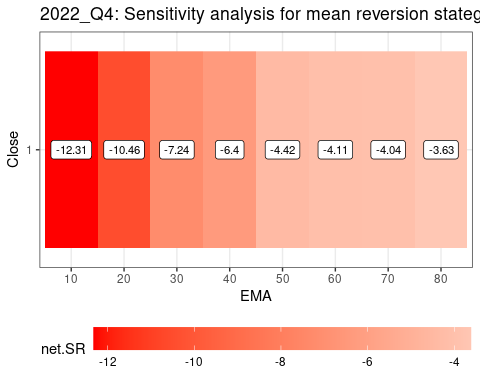
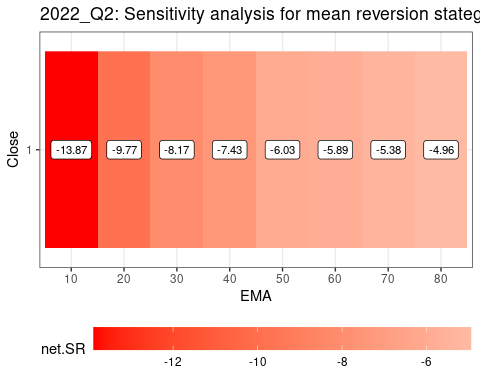
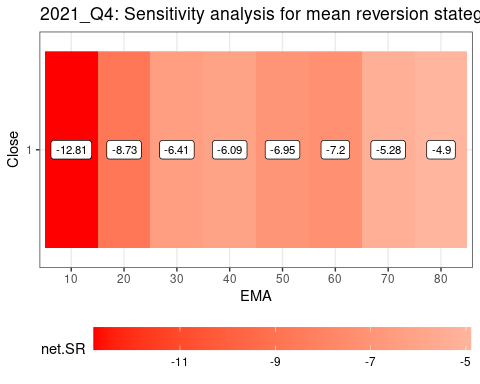
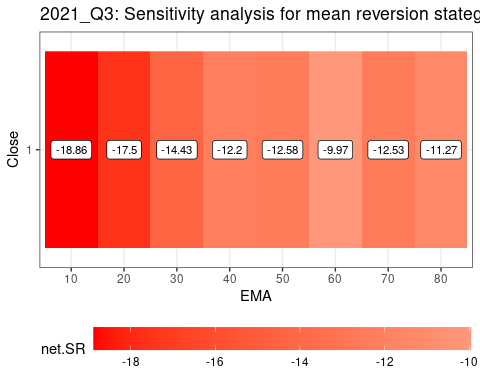
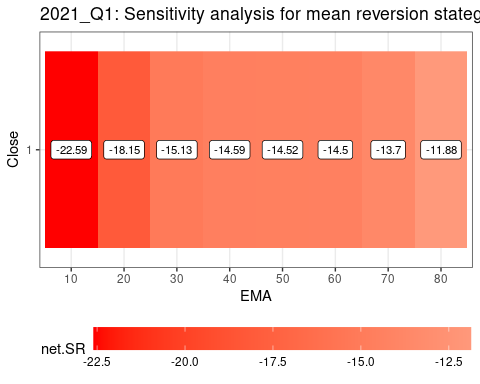
# Mean  
net\_srs\_cross <- list()  
  
for(i in 1:length(sensitivities\_cross)) {  
 net\_srs\_cross[[i]] <- as.list(sensitivities\_cross[[i]][c("net.SR")])[[1]]  
}  
  
average\_net\_sr\_cross <- sapply(seq\_along(net\_srs\_cross[[1]]), function(i) {  
 mean(sapply(net\_srs\_cross, function(x) x[[i]]))  
})  
average\_net\_sr\_cross <- data.frame(net.SR = average\_net\_sr\_cross)  
  
sensitivities\_average\_cross <- sensitivities\_cross[[1]][c("EMA.fast", "EMA.slow")]  
sensitivities\_average\_cross <- cbind(sensitivities\_average\_cross, "net.SR" = average\_net\_sr\_cross)  
  
heatmap\_sr\_mean\_cross <- plotHeatmap(data\_plot = sensitivities\_average\_cross, # dataset (data.frame) with calculations  
 col\_vlabels = "EMA.fast", # column name with the labels for a vertical axis (string)  
 col\_hlabels = "EMA.slow", # column name with the labels for a horizontal axis (string)  
 col\_variable = "net.SR", # column name with the variable to show (string)  
 main = paste("Mean", "Sensitivity analysis for momentum stategy based on EMA crossover", sep = ": "),  
 label\_size = 3)  
  
heatmap\_sr\_mean\_cross



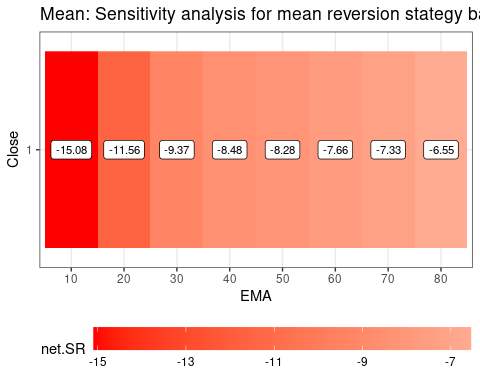
### Single EMA Mean Reversion

Single EMA mean reversion strategies performed even worse than their momentum equivalent.

# Combined  
for (output in heatmap\_list\_single\_mr) {  
 print(output)  
}



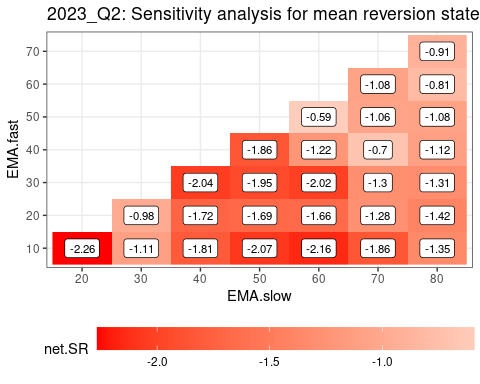
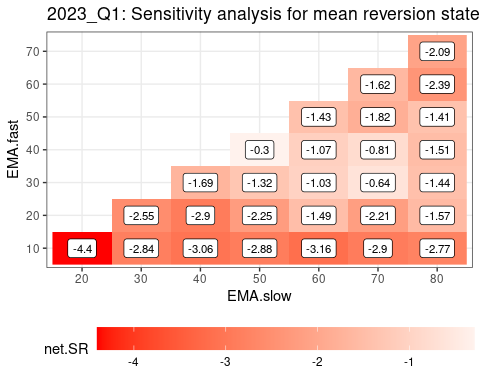
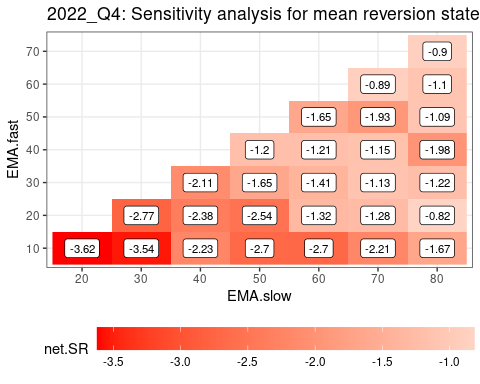
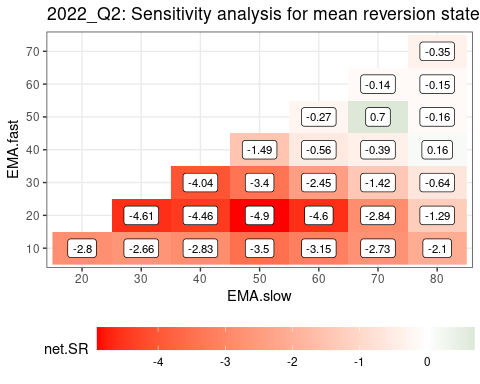
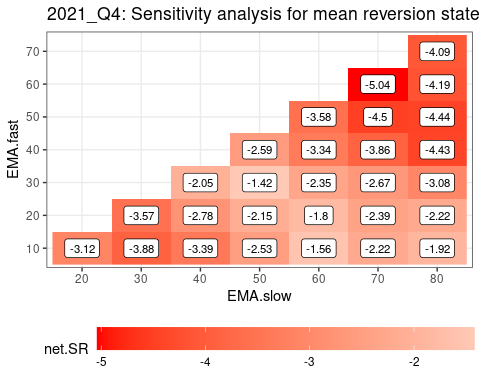
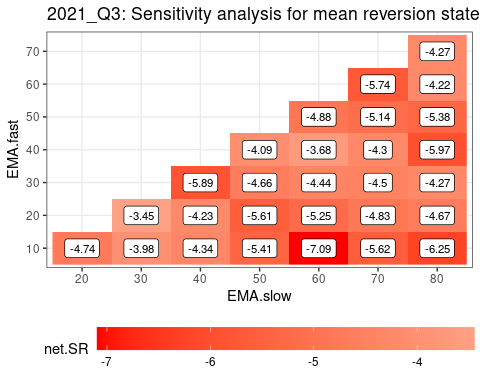
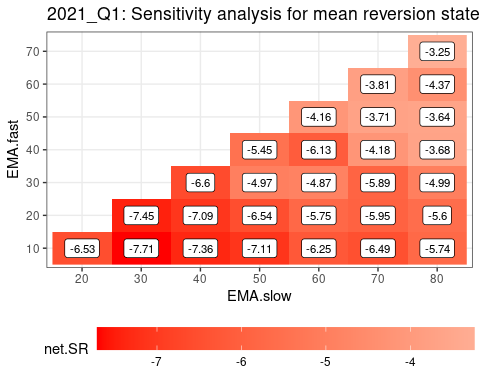
# Mean  
net\_srs\_single\_mr <- list()  
  
for(i in 1:length(sensitivities\_single\_mr)) {  
 net\_srs\_single\_mr[[i]] <- as.list(sensitivities\_single\_mr[[i]][c("net.SR")])[[1]]  
}  
  
average\_net\_sr\_single\_mr <- sapply(seq\_along(net\_srs\_single\_mr[[1]]), function(i) {  
 mean(sapply(net\_srs\_single\_mr, function(x) x[[i]]))  
})  
average\_net\_sr\_single\_mr <- data.frame(net.SR = average\_net\_sr\_single\_mr)  
  
sensitivities\_average\_single\_mr <- sensitivities\_single\_mr[[1]][c("Close", "EMA")]  
sensitivities\_average\_single\_mr <- cbind(sensitivities\_average\_single\_mr, "net.SR" = average\_net\_sr\_single\_mr)  
  
heatmap\_sr\_mean\_single\_mr <- plotHeatmap(data\_plot = sensitivities\_average\_single\_mr, # dataset (data.frame) with calculations  
 col\_vlabels = "Close", # column name with the labels for a vertical axis (string)  
 col\_hlabels = "EMA", # column name with the labels for a horizontal axis (string)  
 col\_variable = "net.SR", # column name with the variable to show (string)  
 main = paste("Mean", "Sensitivity analysis for mean reversion stategy based on single EMA", sep = ": "),  
 label\_size = 3)  
  
heatmap\_sr\_mean\_single\_mr



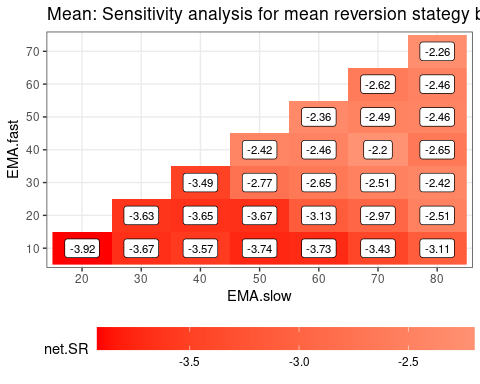
### Crossover EMA Mean Reversion

Similarly, crossover EMA mean reversion strategies produced negative Sharpe ratios for all combinations.

# Combined  
for (output in heatmap\_list\_cross\_mr) {  
 print(output)  
}



# Mean  
net\_srs\_cross\_mr <- list()  
  
for(i in 1:length(sensitivities\_cross\_mr)) {  
 net\_srs\_cross\_mr[[i]] <- as.list(sensitivities\_cross\_mr[[i]][c("net.SR")])[[1]]  
}  
  
average\_net\_sr\_cross\_mr <- sapply(seq\_along(net\_srs\_cross\_mr[[1]]), function(i) {  
 mean(sapply(net\_srs\_cross\_mr, function(x) x[[i]]))  
})  
average\_net\_sr\_cross\_mr <- data.frame(net.SR = average\_net\_sr\_cross\_mr)  
  
sensitivities\_average\_cross\_mr <- sensitivities\_cross\_mr[[1]][c("EMA.fast", "EMA.slow")]  
sensitivities\_average\_cross\_mr <- cbind(sensitivities\_average\_cross\_mr, "net.SR" = average\_net\_sr\_cross\_mr)  
  
heatmap\_sr\_mean\_cross\_mr <- plotHeatmap(data\_plot = sensitivities\_average\_cross\_mr, # dataset (data.frame) with calculations  
 col\_vlabels = "EMA.fast", # column name with the labels for a vertical axis (string)  
 col\_hlabels = "EMA.slow", # column name with the labels for a horizontal axis (string)  
 col\_variable = "net.SR", # column name with the variable to show (string)  
 main = paste("Mean", "Sensitivity analysis for mean reversion stategy based on EMA crossover", sep = ": "),  
 label\_size = 3)  
  
heatmap\_sr\_mean\_cross\_mr



## Group 2

Group 2 data was limited to gold and silver. NA values were inserted in the first and last 10 minutes of the trading session to avoid trading on volatile and reactive parts of the day and to focus on intraday trading.

The two assets were grouped for pair trading and volatility breakout was the approach of choice in the algorithm. Being part of the same precious metals category of commodities, it was plausible to assume some level of common price movement. A spread was calculated, which formed the basis of signals for trading (volatility bands were set around the spread). Both levels and returns were tested for the spread. A short position was initially triggered whenever the signal exceeded the upper bound and then continued whilst the signal remained above the lower bound and a long position otherwise. This strategy was always a mean reversion strategy being against the market.

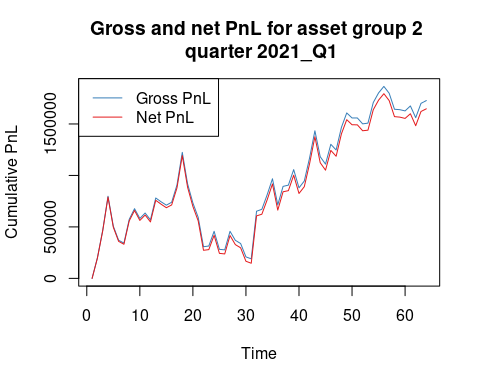
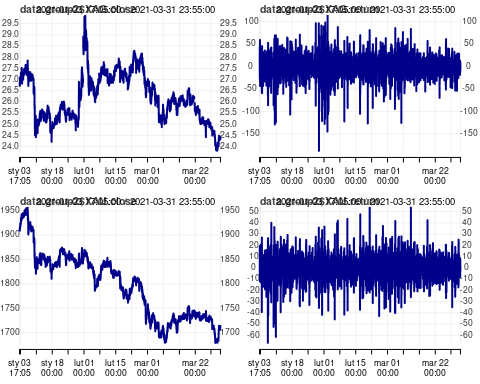
In order to determine the optimal parameters for the models, multiple combinations of volatility memories and multipliers were tested: every combination of 60-180 day volatility memories and 0.5-3.5 multipliers.

All positions were exited in the first 10 and last 10 minutes of the trading session and in between sessions. Any remaining missing ratio value was populated with the previous value for completeness.

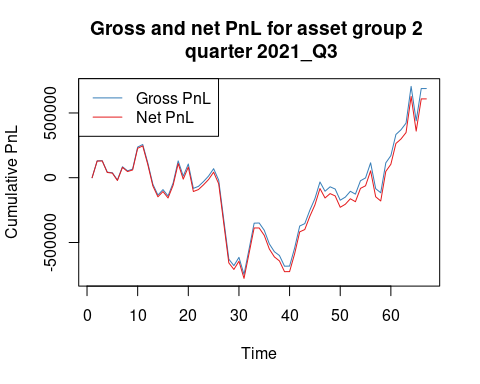
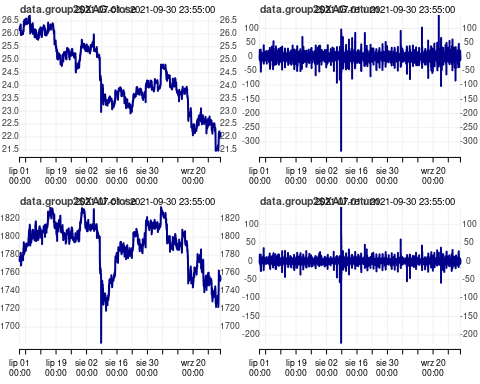
Gross P&L (accounting for point values), net P&L (reflecting transaction costs) and number of trades were calculated. The data was then aggregated to daily level and the following metrics were calculated: Sharpe ratio, Calmar ratio, average number of trades, cumulative sum of gross P&L, cumulative sum of net P&L and a final test statistic.

The entire process was repeated for all in-sample quarters.

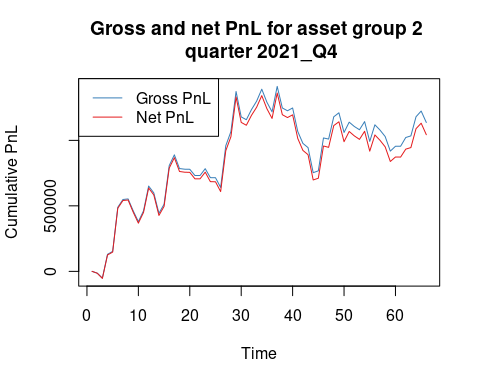
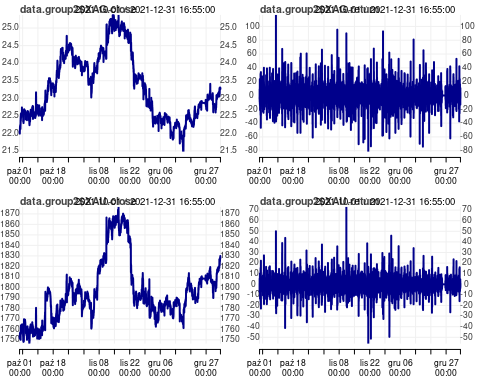
heatmap\_list <- list()  
heatmap\_list2 <- list()  
sensitivities <- list()  
sensitivities2 <- list()  
  
for (selected\_quarter in selected\_quarters) {  
   
 message(selected\_quarter)  
   
 filename\_ <- paste0("data/data2\_", selected\_quarter, ".RData")  
 load(filename\_)  
   
 data.group2 <- get(paste0("data2\_", selected\_quarter))  
 times\_ <- substr(index(data.group2), 12, 19)  
   
 # Keep gold and silver  
 data.group2 <- data.group2[, !colnames(data.group2) %in% c("AUD","CAD")]  
 names(data.group2)[1:2] <- c("XAG.close","XAU.close")  
   
 data.group2.return <- 10000\*diff.xts(log(data.group2))  
 names(data.group2.return)[1:2] <- c("XAG.return","XAU.return")  
   
 data.group2 <- merge(data.group2[, c("XAG.close", "XAU.close")],  
 data.group2.return[, c("XAG.return", "XAU.return")])  
   
 myTheme <- chart\_theme()  
 myTheme$col$line.col <- "darkblue"  
  
 # the following common assumptions were defined:  
 # 1. do not use in calculations the data from the first and last 10 minutes of the session (18:01--18:10 and 16:51--17:00) – put missing values there,  
   
 # lets put missing values for these periods  
 data.group2["T18:01/T18:10",] <- NA   
 data.group2["T16:51/T17:00",] <- NA  
   
 layout(matrix(1:4, 2, 2))  
 print(chart\_Series(data.group2$XAG.close, theme = myTheme))  
 print(chart\_Series(data.group2$XAU.close, theme = myTheme))  
 print(chart\_Series(data.group2$XAG.return, theme = myTheme))  
 print(chart\_Series(data.group2$XAU.return, theme = myTheme))  
 layout(matrix(1))  
   
 ###################################################################  
 # lets formulate a spread: P1 - m \* P2 (P\_NASDAQ - m \* P\_AAPL)   
 # where m = m1/m2 is based on average ratio between the prices  
 # on the PREVIOUS day  
   
 # spread is a signal to our model, which shows whether to take   
 # position or not (volatility bands around the spread)  
   
 # CAUTION! we assume the mean reverting behavior of the spread!  
   
 ####################################################################  
 # lets calculate average ratio of prices on the daily basis  
   
 index\_posix <- index(data.group2)  
 time\_component <- format(index\_posix, format = "%H:%M:%S")  
 target\_time <- "17:00:00"  
 indices <- which(time\_component == target\_time)  
  
 cmd.av.ratio <- period.apply(data.group2,  
 INDEX = indices,  
 function(x) mean(x$XAU.close/x$XAG.close,   
 na.rm = TRUE)  
 )  
   
 names(cmd.av.ratio) <- "av.ratio"  
   
 # chart\_Series(cmd.av.ratio)  
   
 # about 64-74 XAG units per each unit of XAU (future)  
   
 # head(cmd.av.ratio)  
   
 # but calculations based on the first day  
 # will be used on the second day, etc.  
 # lets adjust the dataset accordingly  
 # by moving the time index to 18:00 of the next trading day (same day)  
   
 #index(cmd.av.ratio)  
   
 # lets use functions from lubridate  
 # ceiling\_date() rounds the date up to midnight  
 # (in fact start of the next day)  
 # hours(n), minutes(n) - create a period object  
 # with specified values  
   
 # but some of the dates might be Fridays and in this case  
 # we would move the index to 18:00 on Sunday  
   
 # 6 = Friday  
   
 # lets use if\_else() from dplyr instead  
 # lets apply the changes in our data object  
   
 index(cmd.av.ratio) <-   
 ceiling\_date(index(cmd.av.ratio), "day") -   
 hours(6) +   
 minutes(0) +  
 if\_else(wday(index(cmd.av.ratio)) == 6,   
 days(2),  
 days(0))   
   
 ###################################################################  
 # alternative spread based on RETURNS:  
 # r1 - ms \* r2 (r\_NASDAQ - ms \* r\_AAPL)   
 # where ms = s1/s2 is based on the ratio of standard  
 # deviations of returns on the PREVIOUS day  
   
 cmd.sds.ratio <- period.apply(data.group2,  
 INDEX = indices,  
 function(x) sd(x$XAU.return, na.rm = TRUE) /  
 sd(x$XAG.return, na.rm = TRUE)  
 )  
   
 names(cmd.sds.ratio) <- "sds.ratio"  
   
 # chart\_Series(cmd.sds.ratio)  
   
 # between 0.2 and 0.65 XAG units   
 # per each unit of XAU (future)  
   
 # lets move the index to 18:00 of the next trading day (same day)  
   
 index(cmd.sds.ratio) <-   
 ceiling\_date(index(cmd.sds.ratio), "day") -  
 hours(6) +   
 minutes(0) +  
 if\_else(wday(index(cmd.sds.ratio)) == 6,   
 days(2),   
 days(0))  
   
 #-----------------------------------------------------------  
 # we need to merge our basic 5 min data with daily calculations  
   
 data.group2b <- merge(data.group2,  
 cmd.av.ratio,   
 cmd.sds.ratio)  
   
 # there are a lot of missings in a the last 2 columns  
 # which should be filled with the last non-missing value  
 # (last multiplier is used until there is a new one)  
   
 data.group2b$av.ratio <- na.locf(data.group2b$av.ratio,  
 na.rm = FALSE)  
 data.group2b$sds.ratio <- na.locf(data.group2b$sds.ratio,  
 na.rm = FALSE)  
   
 # lets make sure that we exclude weekends from our data  
   
 table(wday(data.group2b))  
   
 # there are no rows with 7 (Saturday)  
   
 # now we can calculate the spread (in 2 variants)  
 data.group2b$spread\_avratio <-   
 data.group2b$XAU.close -  
 data.group2b$av.ratio \* data.group2b$XAG.close  
   
 data.group2b$spread\_sdsratio <-   
 data.group2b$XAU.return -   
 data.group2b$sds.ratio \* data.group2b$XAG.return  
   
 # plot both spreads  
   
 # lets check it on the plot  
 # layout(matrix(1:2, 2, 1))  
 # chart\_Series(data.group2b$spread\_avratio, theme = myTheme)  
 # abline( h = 0)  
 # chart\_Series(data.group2b$spread\_sdsratio, theme = myTheme)  
 # abline( h = 0)  
 # layout(matrix(1))  
   
 # we assume that spread mean reverts to 0,  
 # which is not that clear on the top panel...  
   
   
 # lets assume we do not trade within the first 10-mins of the day  
 # and exit all positions 10 minutes before the end of quotations  
   
 # lets create a pos\_flat vector and fill it with 0s  
 pos\_flat <- xts(rep(0, nrow(data.group2b)), index(data.group2b))  
   
 # we do not trade within the first 10 mins (18:00-18:10)   
 # but also before that time when session was inactive  
 # and last 10 mins of the session (16:51-17:00)  
 # but also after this time when session was inactive  
   
 pos\_flat["T16:51/T18:10"] <- 1  
   
 # note this covers Fridays and Sundays as the series goes from 17:00 Friday to 17:05 Sunday  
   
 # !!! there are no weekends in our data, so we do not need   
 # to control for that in pos\_flat  
   
   
 # # lets apply the volatility breakout model  
 #   
 # # standard deviation of the spread  
 # # runsd - efficient function for rolling standard deviation  
 #   
 # data.group2b$spread\_avratio\_rollsd120 <-  
 # runsd(data.group2b$spread\_avratio,  
 # 120,  
 # endrule = "NA",  
 # align = "right")  
 #   
 # data.group2b$spread\_sdsratio\_rollsd120 <-   
 # runsd(data.group2b$spread\_sdsratio,   
 # 120,   
 # endrule = "NA",  
 # align = "right")  
 #   
 # # lets put missings whenever XAG price is missing  
 #   
 # data.group2b$spread\_avratio\_rollsd120[is.na(data.group2b$XAG.close)] <- NA  
 # data.group2b$spread\_sdsratio\_rollsd120[is.na(data.group2b$XAG.close)] <- NA  
 #   
 # #---------------------------------  
 # # applying a volatility breakout model  
 # # sample upper and lower bounds for spreads  
 # # for a volatility multiplier of 3  
 # # (here we put the upper and lower band along zero)  
 #   
 # data.group2b$upper <- 3 \* data.group2b$spread\_avratio\_rollsd120  
 # data.group2b$lower <- (-3 \* data.group2b$spread\_avratio\_rollsd120)  
 #   
 # # lets see it on the plot  
 # chart\_Series(data.group2b$spread\_avratio, theme = myTheme)  
 # add\_TA(data.group2b$upper, col = "red", on = 1)  
 # add\_TA(data.group2b$lower, col = "red", on = 1)  
 # abline(h = 0, lty = 2, col = "gray")  
 #   
 # # the same for spread\_sdsratio  
 #   
 # data.group2b$upper2 <- 3 \* data.group2b$spread\_sdsratio\_rollsd120  
 # data.group2b$lower2 <- (-3 \* data.group2b$spread\_sdsratio\_rollsd120)  
 #   
 # # lets see it on the plot  
 # chart\_Series(data.group2b$spread\_sdsratio, theme = myTheme)  
 # add\_TA(data.group2b$upper2, col = "red", on = 1)  
 # add\_TA(data.group2b$lower2, col = "red", on = 1)  
 # abline(h = 0, lty = 2, col = "gray")  
 #   
 # ### position based on relation of the spread to volatility bands  
 #   
 # # lets assume we do not trade within the first 10-mins of the day  
 # # and exit all positions 10 minutes before the end of quotations  
 #   
 # # lets create a pos\_flat vector and fill it with 0s  
 # pos\_flat <- xts(rep(0, nrow(data.group2b)), index(data.group2b))  
 #   
 # # we do not trade within the 10 mins quarter (18:00-18:10)   
 # # but also before that time when session was inactive  
 # # and last 10 mins of the session (16:51-17:00)  
 # # but also after this time when session was inactive  
 #   
 # pos\_flat["T16:51/T18:10"] <- 1  
 #   
 # # note this covers Fridays and Sundays as the series goes from 17:00 Friday to 17:05 Sunday  
 #   
 # # !!! there are no weekends in our data, so we do not need   
 # # to control for that in pos\_flat  
 #   
 # # lets use the positionVB\_new() function from previous labs  
 #   
 # data.group2b$pos\_strategy <- positionVB\_new(signal = data.group2b$spread\_avratio,  
 # lower = data.group2b$lower,  
 # upper = data.group2b$upper,  
 # pos\_flat = pos\_flat,  
 # strategy = "mr" # important !!!  
 # )  
 #   
 # # lets create a vector of number of transactions  
 #   
 # data.group2b$ntrans <- abs(diff.xts(data.group2b$pos\_strategy))  
 #   
 # # caution !!!  
 # # our strategy pnl would be position\*(pnl of the spread)  
 # # pnl of the spread = pos\*[diff(XAU.close)\*$100 - m\*diff(XAG.close)\*$5000]  
 #   
 # data.group2b$gross.pnl <- (data.group2b$pos\_strategy) \*  
 # (diff.xts(data.group2b$XAU.close) \* 100 -  
 # data.group2b$av.ratio \* diff.xts(data.group2b$XAG.close) \* 5000)  
 # # 100 is point value of XAU and 5000 is the point value of XAG so multiply by those  
 # # pnl after costs  
 # # costs = $7 for XAG and $12 for XAU = (12+m\*7) in total  
 # # there is NO minus "-" in the costs - they are always positive !!!  
 #   
 # data.group2b$net.pnl <- data.group2b$gross.pnl -  
 # data.group2b$ntrans \* (12 + data.group2b$av.ratio \* 7)  
 #   
 #   
 # data.group2b$cum.gross.pnl <- cumsum(ifelse(is.na(data.group2b$gross.pnl),  
 # 0,  
 # data.group2b$gross.pnl))  
 #   
 # data.group2b$cum.net.pnl <- cumsum(ifelse(is.na(data.group2b$net.pnl),  
 # 0,  
 # data.group2b$net.pnl))  
 #   
 # # lets see if it was profitable  
 #   
 # chart\_Series(data.group2b$cum.gross.pnl,  
 # theme = myTheme)  
 # add\_TA(data.group2b$cum.net.pnl,  
 # on = 1,  
 # col = "red")  
 # abline(h = 0, lty = 2, col = "gray")  
   
 # lets do a comparison within a loop for spread and spread2  
  
 for(volat.sd in c(60, 90, 120, 150, 180)) { # different volatility memories  
 for(m\_ in c(0.5, 1, 1.5, 2, 2.5, 3, 3.5)) { # different multipliers  
   
 message(paste0("volat.sd = ", volat.sd,  
 ", m\_ = ", m\_))   
   
 # calculating elements of the strategy  
 XAU\_price <- coredata(data.group2b$XAU.close)  
 XAG\_price <- coredata(data.group2b$XAG.close)  
   
 signal <- coredata(data.group2b$spread\_avratio)  
 signal2 <- coredata(data.group2b$spread\_sdsratio)  
   
 upper <- m\_ \* runsd(signal, volat.sd,   
 endrule = "NA",   
 align = "right")  
 lower <- -m\_ \* runsd(signal, volat.sd,   
 endrule = "NA",   
 align = "right")  
   
 upper2 <- m\_ \* runsd(signal2, volat.sd,   
 endrule = "NA",   
 align = "right")  
 lower2 <- -m\_ \* runsd(signal2, volat.sd,   
 endrule = "NA",   
 align = "right")  
   
 # position for mean-reverting strategy  
 pos.mr <- positionVB\_new(signal, lower, upper,  
 pos\_flat = pos\_flat,  
 strategy = "mr" # important !!!  
 )  
 pos.mr2 <- positionVB\_new(signal2, lower2, upper2,  
 pos\_flat = pos\_flat,  
 strategy = "mr" # important !!!  
 )  
 # number of transactions  
 ntrans <- abs(diff.xts(pos.mr))  
 ntrans2 <- abs(diff.xts(pos.mr2))  
   
 # gross pnl  
 gross.pnl <- (pos.mr) \*  
 (diff.xts(XAU\_price) \* 100 # point value for XAU  
 - coredata(data.group2b$av.ratio) \* diff.xts(XAG\_price) \* 5000) # point value for XAG  
   
 gross.pnl2 <- (pos.mr2) \*  
 (diff.xts(XAU\_price) \* 100 # point value for XAU  
 - coredata(data.group2b$sds.ratio) \* diff.xts(XAG\_price) \* 5000) # point value for XAG  
   
 # pnl after costs  
 # costs = $7 for XAG and $12 for XAU = (12+m\*7) in total  
 # there is NO minus "-" in the costs - they are always positive !!!  
   
 net.pnl <- gross.pnl - ntrans \* (12 + coredata(data.group2b$av.ratio) \* 7)  
 net.pnl2 <- gross.pnl2 - ntrans2 \* (12 + coredata(data.group2b$sds.ratio) \* 7)  
   
 # aggregate to daily  
 # ends\_ <- endpoints(data.group2b, "days")  
   
 pnl.gross.d <- period.apply(gross.pnl, INDEX = indices,   
 FUN = function(x) sum(x, na.rm = TRUE))  
 pnl.gross2.d <- period.apply(gross.pnl2, INDEX = indices,   
 FUN = function(x) sum(x, na.rm = TRUE))  
 pnl.net.d <- period.apply(net.pnl, INDEX = indices,  
 FUN = function(x) sum(x, na.rm = TRUE))  
 pnl.net2.d <- period.apply(net.pnl2, INDEX = indices,  
 FUN = function(x) sum(x, na.rm = TRUE))  
 ntrans.d <- period.apply(ntrans, INDEX = indices,   
 FUN = function(x) sum(x, na.rm = TRUE))  
 ntrans2.d <- period.apply(ntrans2, INDEX = indices,   
 FUN = function(x) sum(x, na.rm = TRUE))  
   
 # calculate summary measures  
 gross.SR <- mySR(pnl.gross.d, scale = 252)  
 gross.SR2 <- mySR(pnl.gross2.d, scale = 252)  
 net.SR <- mySR(pnl.net.d, scale = 252)  
 net.SR2 <- mySR(pnl.net2.d, scale = 252)  
   
 gross.CR <- myCalmarRatio(pnl.gross.d, scale = 252)  
 gross.CR2 <- myCalmarRatio(pnl.gross2.d, scale = 252)  
 net.CR <- myCalmarRatio(pnl.net.d, scale = 252)  
 net.CR2 <- myCalmarRatio(pnl.net2.d, scale = 252)  
   
 gross.PnL <- sum(pnl.gross.d, na.rm = TRUE)  
 gross.PnL2 <- sum(pnl.gross2.d, na.rm = TRUE)  
 net.PnL <- sum(pnl.net.d, na.rm = TRUE)  
 net.PnL2 <- sum(pnl.net2.d, na.rm = TRUE)  
   
 av.daily.ntrans <- mean(ntrans.d, na.rm = TRUE)  
 av.daily.ntrans2 <- mean(ntrans2.d, na.rm = TRUE)   
   
 stat = net.CR \* max(0, log(abs(net.PnL/1000)))  
 stat2 = net.CR2 \* max(0, log(abs(net.PnL2/1000)))  
   
 # collecting all statistics for a particular quarter  
 if(volat.sd == 180 & m\_ == 1) {  
 quarter\_stats <- data.frame(quarter = selected\_quarter,  
 assets.group = 2,  
 gross.SR,  
 net.SR,  
 gross.CR,  
 net.CR,  
 gross.PnL,  
 net.PnL,  
 av.daily.ntrans,  
 stat,  
 stringsAsFactors = FALSE  
 )  
   
 quarter\_stats2 <- data.frame(quarter = selected\_quarter,  
 assets.group = 2,  
 gross.SR2,  
 net.SR2,  
 gross.CR2,  
 net.CR2,  
 gross.PnL2,  
 net.PnL2,  
 av.daily.ntrans2,  
 stat2,  
 stringsAsFactors = FALSE  
 )  
   
 # collect summaries for all quarters  
 if(!exists("quarter\_stats.all.group2")) quarter\_stats.all.group2 <- quarter\_stats else  
 quarter\_stats.all.group2 <- rbind(quarter\_stats.all.group2, quarter\_stats)  
   
 if(!exists("quarter\_stats2.all.group2")) quarter\_stats2.all.group2 <- quarter\_stats2 else  
 quarter\_stats2.all.group2 <- rbind(quarter\_stats2.all.group2, quarter\_stats2)  
   
 # create a plot of gross and net pnl and save it to png file  
 y\_range <- range(c(cumsum(pnl.gross.d), cumsum(pnl.net.d)))  
  
 print( # when plotting in a loop you have to use print()  
 plot(cumsum(pnl.gross.d),  
 type = "l",  
 main = paste0("Gross and net PnL for asset group 2 \n quarter ", selected\_quarter),  
 col = "#377EB8",  
 xlab = "Time",  
 ylab = "Cumulative PnL",  
 ylim = y\_range  
 )  
 )  
 lines(cumsum(pnl.net.d), col = "#E41A1C")  
 legend("topleft", legend = c("Gross PnL", "Net PnL"), col = c("#377EB8", "#E41A1C"), lty = 1, cex = 1)  
 }  
   
 # summary of a particular strategy  
 summary\_ <- data.frame(spread = "av.ratio",  
 volat.sd = volat.sd,  
 m = m\_,  
 period = selected\_quarter, # "2016-08-16 - 2016-11",  
 gross.SR,  
 net.SR,  
 gross.PnL,  
 net.PnL,  
 av.daily.ntrans,  
 stringsAsFactors = FALSE)  
  
 summary2\_ <- data.frame(spread = "sds.ratio",  
 volat.sd = volat.sd,  
 m = m\_,  
 period = selected\_quarter, # "2016-08-16 - 2016-11",  
 gross.SR = gross.SR2,  
 net.SR = net.SR2,  
 gross.PnL = gross.PnL2,  
 net.PnL = net.PnL2,  
 av.daily.ntrans = av.daily.ntrans2,  
 stringsAsFactors = FALSE)  
  
 # putting all summaries together  
  
 if(!exists("summary.pair.trading")) summary.pair.trading <- rbind(summary\_, summary2\_) else  
 summary.pair.trading <- rbind(summary.pair.trading, summary\_, summary2\_)  
   
 # deleting working files not needed any more  
 rm(gross.SR, gross.SR2, net.SR, net.SR2, net.CR, net.CR2,  
 gross.PnL, gross.PnL2, net.PnL, net.PnL2,  
 av.daily.ntrans, av.daily.ntrans2, stat, stat2,  
 pnl.gross.d, pnl.gross2.d, pnl.net.d, pnl.net2.d,   
 ntrans.d, ntrans2.d,  
 pnl.gross, pnl.gross2, pnl.net, pnl.net2,   
 ntrans, ntrans2,  
 pos.mr, pos.mr2, summary\_, summary2\_,  
 XAU\_price, XAG\_price,  
 signal, signal2, lower, lower2, upper, upper2)  
   
 } # end of loop for m\_  
 } # end of loop for volatility   
   
   
 # lets see the results on the heatmap graph  
   
 # net.SR - spread av\_ratio  
 heatmap\_sr <- plotHeatmap(data\_plot = summary.pair.trading[summary.pair.trading$spread == "av.ratio",], # dataset (data.frame) with calculations  
 col\_vlabels = "volat.sd", # column name with the labels for a vertical axis (string)  
 col\_hlabels = "m", # column name with the labels for a horizontal axis (string)  
 col\_variable = "net.SR", # column name with the variable to show (string)  
 main = paste(selected\_quarter, "Sensitivity analysis for pair trading - spread based on prices ratio", sep = ": "),  
 label\_size = 3)  
   
 # volat.sd = 180, m\_ = 3.5  
 # browser()  
 #out <- summary.pair.trading  
 heatmap\_sr2 <- plotHeatmap(data\_plot = summary.pair.trading[summary.pair.trading$spread == "sds.ratio",], # dataset (data.frame) with calculations  
 col\_vlabels = "volat.sd", # column name with the labels for a vertical axis (string)  
 col\_hlabels = "m", # column name with the labels for a horizontal axis (string)  
 col\_variable = "net.SR", # column name with the variable to show (string)  
 main = paste(selected\_quarter, "Sensitivity analysis for pair trading - spread based on returns ratio", sep = ": "),  
 label\_size = 3)  
   
 # net.Pnl - spread av\_ratio  
 # plotHeatmap(data\_plot = summary.pair.trading[summary.pair.trading$spread == "av.ratio",], # dataset (data.frame) with calculations  
 # col\_vlabels = "volat.sd", # column name with the labels for a vertical axis (string)  
 # col\_hlabels = "m", # column name with the labels for a horizontal axis (string)  
 # col\_variable = "net.PnL", # column name with the variable to show (string)  
 # main = "Sensitivity analysis for pair trading - spread based on prices ratio",  
 # label\_size = 3)  
   
 # av.daily.ntrans  
 # plotHeatmap(data\_plot = summary.pair.trading[summary.pair.trading$spread == "av.ratio",], # dataset (data.frame) with calculations  
 # col\_vlabels = "volat.sd", # column name with the labels for a vertical axis (string)  
 # col\_hlabels = "m", # column name with the labels for a horizontal axis (string)  
 # col\_variable = "av.daily.ntrans", # column name with the variable to show (string)  
 # main = "Sensitivity analysis for pair trading - spread based on prices ratio",  
 # label\_size = 3)  
   
 sensitivities[[selected\_quarter]] <- summary.pair.trading[summary.pair.trading$spread == "av.ratio",]  
 sensitivities2[[selected\_quarter]] <- summary.pair.trading[summary.pair.trading$spread == "sds.ratio",]  
 rm(summary.pair.trading)  
   
 # collect summaries for all quarters  
 # if(!exists("heatmaps.all.group2")) heatmaps.all.group2 <- heatmap\_sr else  
 # heatmaps.all.group2 <- rbind(heatmaps.all.group2, heatmap\_sr)  
   
 heatmap\_list[[selected\_quarter]] <- heatmap\_sr  
 heatmap\_list2[[selected\_quarter]] <- heatmap\_sr2  
}



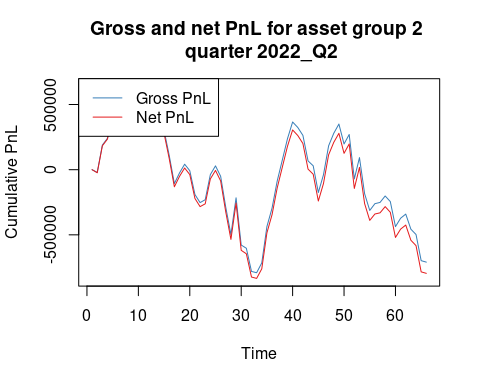
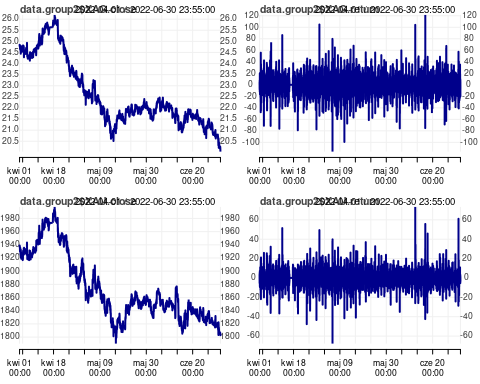
## NULL



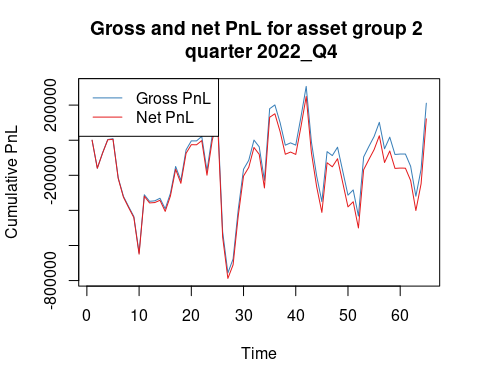
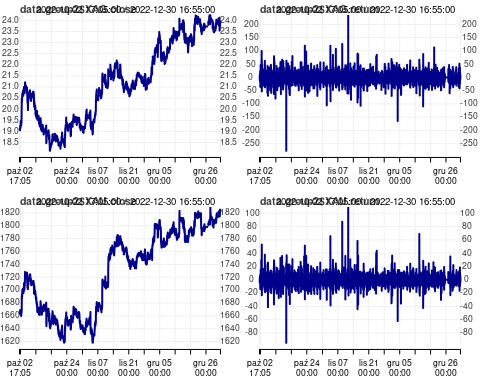
## NULL



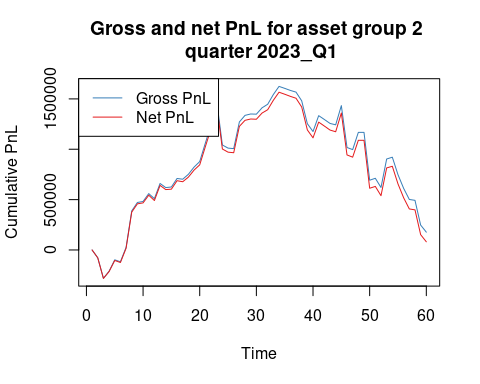
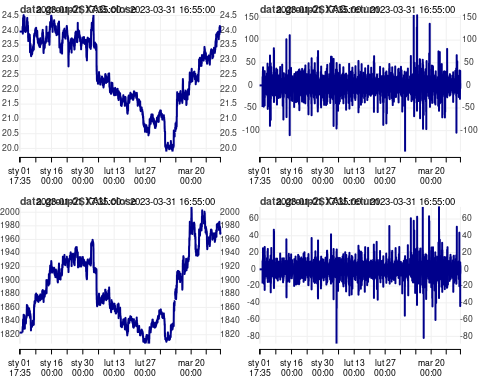
## NULL



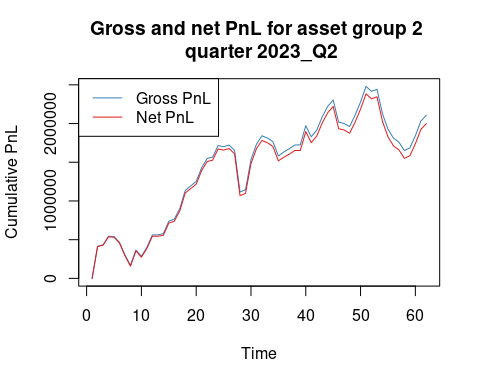
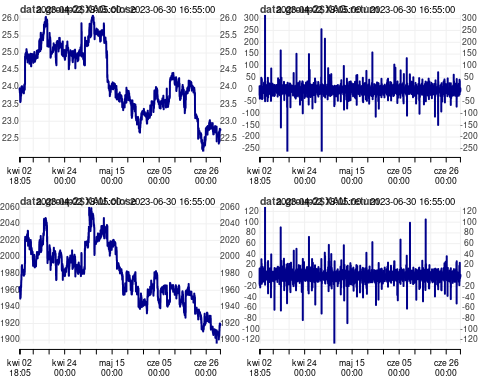
## NULL



## NULL



## NULL



## NULL

The results were depicted using heatmaps containing Sharpe ratios, considering the return per unit of volatility. The largest value indicated which parameter combination generated the best performance on the in-sample data.

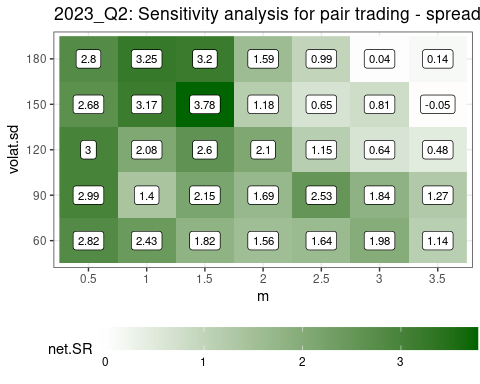
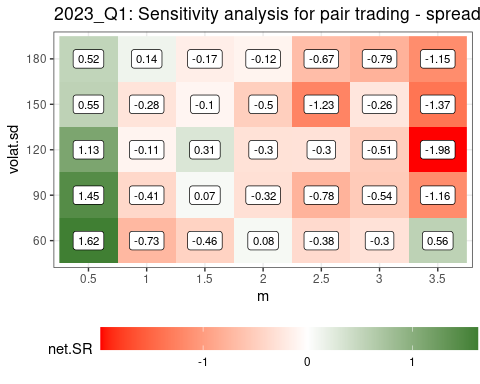
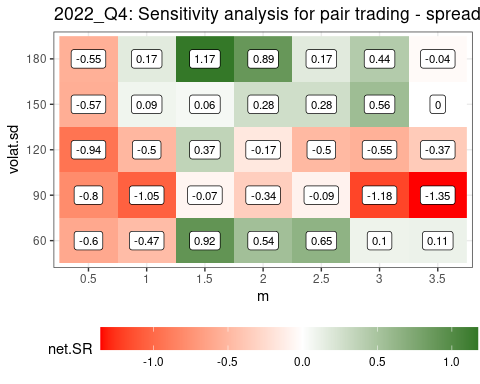
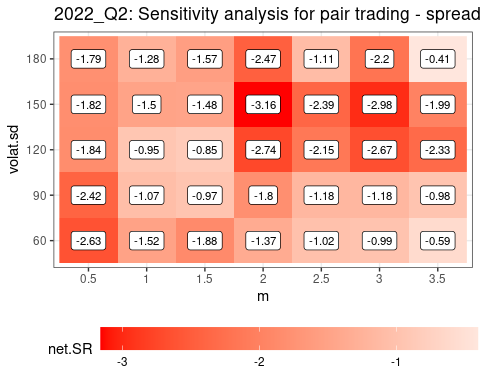
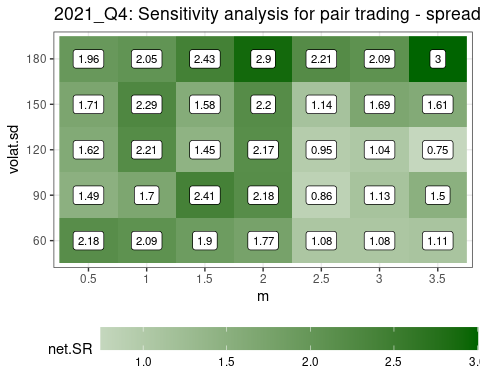
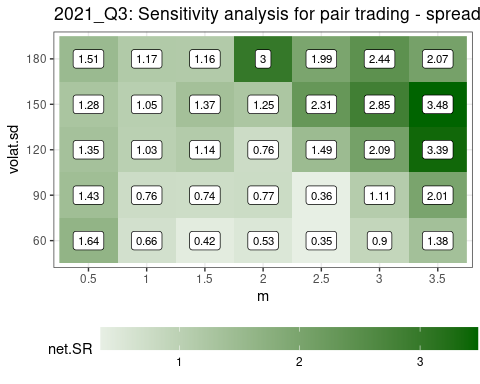
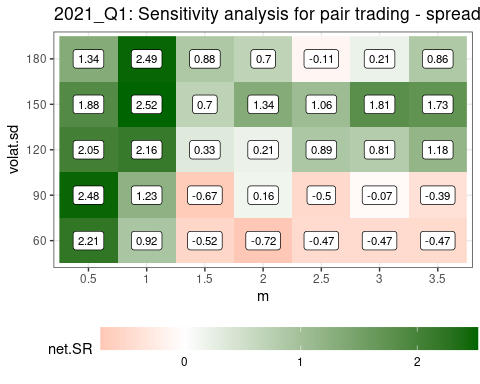
The optimal strategy was volatility breakout using levels where volatility memory was 180 days and the multiplier was 1. The P&L graphs above display the performance of this particular strategy.

Heatmaps supporting the choice of strategy are shown below. Quarterly outputs, as well as the mean across quarters are shown.

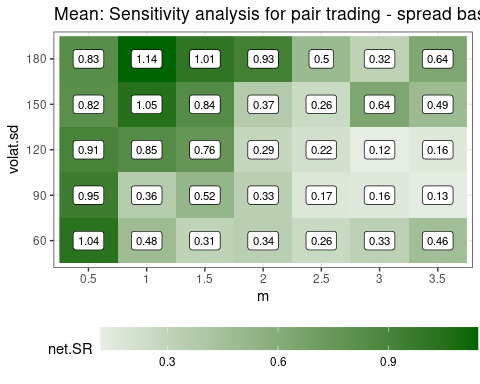
### Volatility Breakout - Levels

Volatility breakout using price levels generated positive Sharpe ratios across all combinations. The best strategy was chosen in this approach: 180 days volatility memory and a multiplier of 1 produced a Sharpe ratio of 1.14.

# Combined  
for (output in heatmap\_list) {  
 print(output)  
}



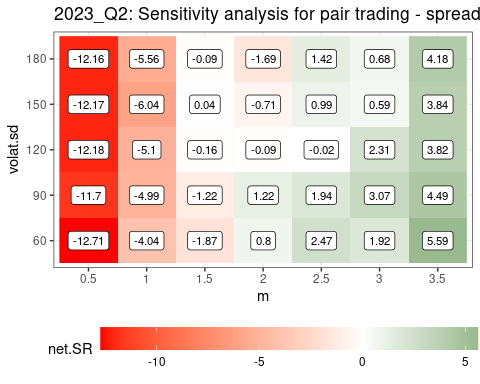
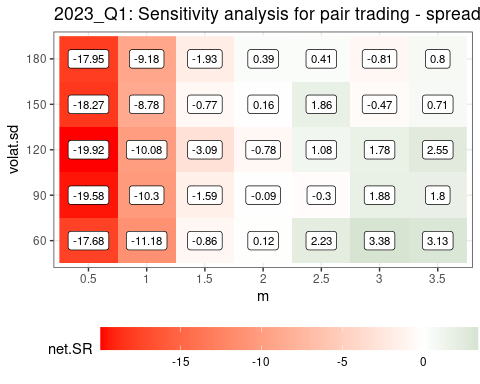
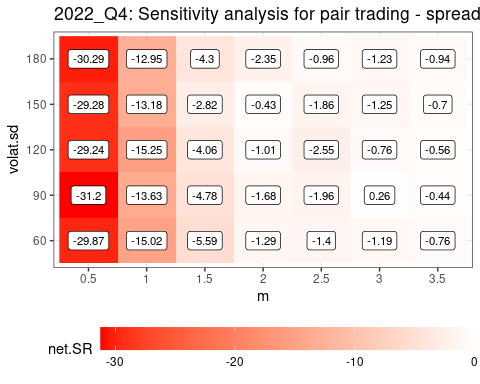
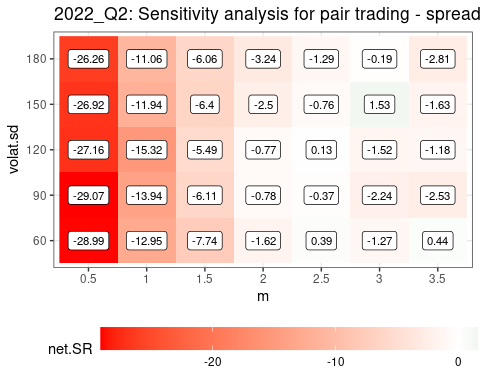
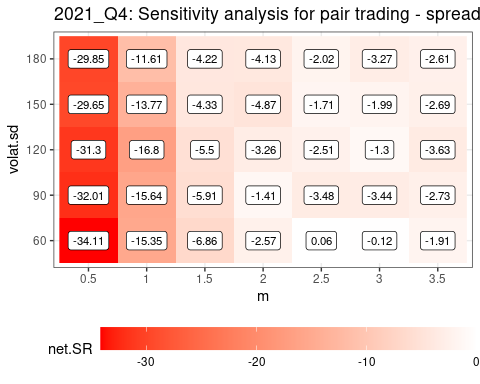
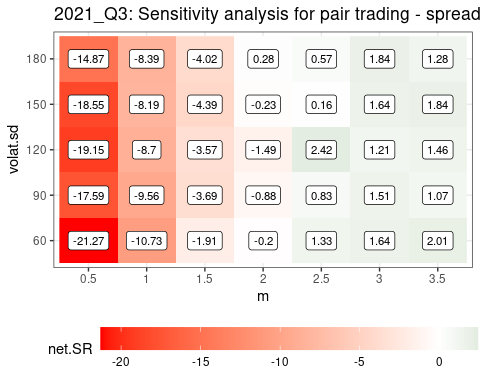
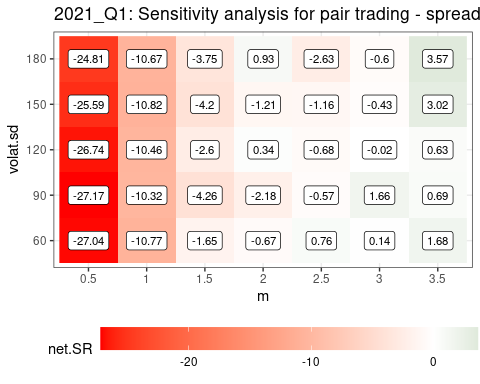
# Mean  
net\_srs <- list()  
  
for(i in 1:length(sensitivities)) {  
 net\_srs[[i]] <- as.list(sensitivities[[i]][c("net.SR")])[[1]]  
}  
  
average\_net\_sr <- sapply(seq\_along(net\_srs[[1]]), function(i) {  
 mean(sapply(net\_srs, function(x) x[[i]]))  
})  
average\_net\_sr <- data.frame(net.SR = average\_net\_sr)  
  
sensitivities\_average <- sensitivities[[1]][c("spread", "volat.sd", "m")]  
sensitivities\_average <- cbind(sensitivities\_average, "net.SR" = average\_net\_sr)  
  
heatmap\_sr\_mean <- plotHeatmap(data\_plot = sensitivities\_average, # dataset (data.frame) with calculations  
 col\_vlabels = "volat.sd", # column name with the labels for a vertical axis (string)  
 col\_hlabels = "m", # column name with the labels for a horizontal axis (string)  
 col\_variable = "net.SR", # column name with the variable to show (string)  
 main = paste("Mean", "Sensitivity analysis for pair trading - spread based on prices ratio", sep = ": "),  
 label\_size = 3)  
  
heatmap\_sr\_mean



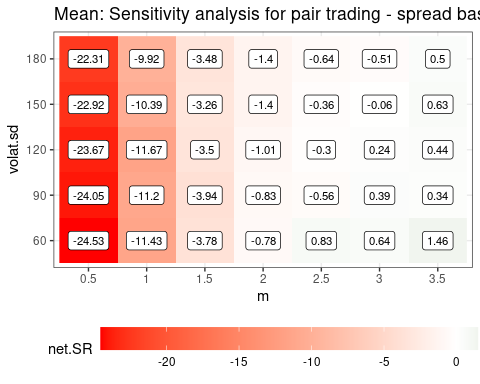
### Volatility Breakout - Returns

Volatility breakout using returns yielded worse Sharpe ratios, particularly at low levels of the multiplier.

# Combined  
for (output in heatmap\_list2) {  
 print(output)  
}



# Mean  
net\_srs2 <- list()  
  
for(i in 1:length(sensitivities2)) {  
 net\_srs2[[i]] <- as.list(sensitivities2[[i]][c("net.SR")])[[1]]  
}  
  
average\_net\_sr2 <- sapply(seq\_along(net\_srs2[[1]]), function(i) {  
 mean(sapply(net\_srs2, function(x) x[[i]]))  
})  
average\_net\_sr2 <- data.frame(net.SR = average\_net\_sr2)  
  
sensitivities\_average2 <- sensitivities2[[1]][c("spread", "volat.sd", "m")]  
sensitivities\_average2 <- cbind(sensitivities\_average2, "net.SR" = average\_net\_sr2)  
  
heatmap\_sr\_mean2 <- plotHeatmap(data\_plot = sensitivities\_average2, # dataset (data.frame) with calculations  
 col\_vlabels = "volat.sd", # column name with the labels for a vertical axis (string)  
 col\_hlabels = "m", # column name with the labels for a horizontal axis (string)  
 col\_variable = "net.SR", # column name with the variable to show (string)  
 main = paste("Mean", "Sensitivity analysis for pair trading - spread based on returns ratio", sep = ": "),  
 label\_size = 3)  
  
heatmap\_sr\_mean2



## Conclusion

### Group 1

The best strategy for group 1 was crossover EMA momentum with a fast EMA of 60 days and slow EMA of 70 days. There were quarters in which most combinations (even containing fewer days) generated positive Sharpe ratios. This was in line with continuing bull markets throughout such periods. In general, shorter EMA periods (particularly 10 days) often yielded negative Sharpe ratios, as the market tended to revert after such short bullish signals. A longer-term view is beneficial with the S&P500, as supported by the strategy of choice: a long-term signal relative to an even longer time horizon indicated there was a sustained trading opportunity - about **0.95** Sharpe ratio on average across all quarters. The single EMA entry and exit approach performed worse, as it was falsely identifying bull/bear markets based on price changes relative to one EMA. Mean reversion also performed worse for all variants.

quarter\_stats.all.group1

## quarter assets.group gross.SR net.SR gross.CR net.CR gross.PnL  
## 1 2021\_Q1 1 2.7643615 1.7515603 14.3639656 8.2525053 14079.20  
## 2 2021\_Q3 1 4.3221804 2.9608779 15.6427494 10.1261152 13957.15  
## 3 2021\_Q4 1 4.0216583 3.0336018 16.7524145 9.8777086 21485.55  
## 4 2022\_Q2 1 -0.3837697 -0.8968936 -0.5654044 -1.2004592 -3659.05  
## 5 2022\_Q4 1 0.3310653 -0.2104375 0.8403617 -0.4334056 2832.50  
## 6 2023\_Q1 1 0.9874473 0.3738055 2.3871303 0.8047851 6936.70  
## 7 2023\_Q2 1 0.3614862 -0.3350885 0.8106422 -0.7066736 2072.30  
## net.PnL av.daily.ntrans stat  
## 1 9079.20 7.936508 18.2049118  
## 2 9777.15 6.333333 23.0880291  
## 3 16445.55 7.636364 27.6581267  
## 4 -8639.05 7.661538 -2.5885414  
## 5 -1827.50 7.169231 -0.2613215  
## 6 2666.70 6.584615 0.7893668  
## 7 -1947.70 6.312500 -0.4711034

Cumulative net P&L amounted to **$25,554** and cumulative summary statistic was **66.4** using in-sample data.

### Group 2

The optimal strategy for group 2 was volatility breakout using levels with 180 days of volatility memory and a multiplier of 1. All combinations were generally profitable, indicating going against an increasing spread between gold and silver is a good strategy. There were, however, some quarters where this was not true and the divergence continued. The combination that consistently delivered the best results generated a Sharpe ratio of **1.14**. It indicated it was worth taking a longer view of volatility and a lower multiplier, meaning whenever the signal moved relatively little outside the long-term bounds, a trading opportunity was worthwhile. There were also fewer trades per day (hence lower trading costs) with the selected strategy in group 2. The return-based volatility breakout entry and exit approach performed worse, as it was unable to identify genuine moves away from gold and silver’s mean to trade against, particularly at low multiplier values when the bounds are more restrictive.

options(digits=2)  
quarter\_stats.all.group2

## quarter assets.group gross.SR net.SR gross.CR net.CR gross.PnL net.PnL  
## 1 2021\_Q1 2 2.60 2.49 6.58 6.20 1727629 1647371  
## 2 2021\_Q3 2 1.32 1.17 2.58 2.24 688515 608509  
## 3 2021\_Q4 2 2.22 2.05 6.59 6.00 1136272 1043029  
## 4 2022\_Q2 2 -1.14 -1.28 -1.89 -2.08 -709471 -795455  
## 5 2022\_Q4 2 0.29 0.17 0.89 0.51 210894 121714  
## 6 2023\_Q1 2 0.30 0.14 0.51 0.23 176808 80963  
## 7 2023\_Q2 2 3.41 3.25 10.34 9.76 2108033 1999241  
## av.daily.ntrans stat  
## 1 3.5 46.0  
## 2 3.2 14.3  
## 3 3.5 41.7  
## 4 3.2 -13.9  
## 5 3.4 2.5  
## 6 3.7 1.0  
## 7 4.0 74.2

Cumulative net P&L amounted to **$4,705,373** and cumulative summary statistic was **165.8** using in-sample data.

The results for the best models for groups 1 and 2 can be saved as csv - the following lines need to be uncommented.

# write.csv(quarter\_stats.all.group1,   
# "quarter\_stats.all.group1.csv",  
# row.names = FALSE)1  
  
# write.csv(quarter\_stats.all.group2,   
# "quarter\_stats.all.group2.csv",  
# row.names = FALSE)