

Mean Variance Portfolio Allocation

Yu-Ting Huang

2022-01-01

- Computing Daily Returns From Price Data
 - Modifying prices by exchange rates
 - Calculating daily returns
 - Calculating log daily return
 - Mean
 - Standard deviation
 - Skewness
 - Kurtosis
 - Autocorrelation
 - Correlation Matrix
 - Summary statistics
 - Summary
 - Normality: Evaluating whether it is reasonable to approximate the return process by i.i.d. Gaussian distribution.
- Mean Variance Analysis
 - Situation 1: short selling allowed
 - Situation 2: short selling disallowed
 - Efficient frontier
- Mean Variance Analysis with 5-year Rolling Window
 - Find the minimum variance portfolio (Short sell allowed)
 - Find the minimum variance portfolio (Short sell prohibited)
 - Examine if this portfolio is significantly time-varying
 - Compare Market-based weights and Mean-Variance weights

Computing Daily Returns From Price Data

First of all, the original data were cleaned and sorted as data frame. In addition, index prices were converted from local currencies to USD for better manipulation in the following questions.

Hide

```
# load libraries
library(readxl)
library(xts)
library(openxlsx)
library(dplyr)
library(quantmod)
library(zoo)
library(mnormt)
library(moments)
```

Modifying prices by exchange rates

Hide

```
# original price
price <- read_excel("PS1_data.xls", sheet = "hard_copy", skip = 2)
```

New names:

- * `` -> ...1
- * `` -> ...2
- * `` -> ...3
- * `` -> ...13
- * `` -> ...14
- * ...

[Hide](#)

```
price <- price[c(4:12)]

# exchange rate
exchange_rate <- read_excel("PS1_data.xls", sheet = "hard_copy", skip = 2)
```

New names:

- * `` -> ...1
- * `` -> ...2
- * `` -> ...3
- * `` -> ...13
- * `` -> ...14
- * ...

[Hide](#)

```
exchange_rate <- exchange_rate[c(14:21)]
exchange_rate <- cbind(price[,1],exchange_rate)

price <- as.data.frame(price)
exchange_rate <- as.data.frame(exchange_rate)
exchange_rate <- setNames(exchange_rate, c("date","TSX","CAC","DAX","Euro50","NIKKEI","FTSE",
"SP500","IBOVESPA"))
#head(exchange_rate)
#head(price)

# get total date
n = as.numeric(nrow(price))

# convert price to dollar
price$TSX <- price$TSX*exchange_rate$TSX
price$CAC <- price$CAC*exchange_rate$CAC
price$DAX <- price$DAX*exchange_rate$DAX
price$Eurostoxx50 <- price$Eurostoxx50*exchange_rate$Euro50
price$NIKKEI225 <- price$NIKKEI225*exchange_rate$NIKKEI
price$FTSE100 <- price$FTSE100*exchange_rate$FTSE
price$SP500 <- price$SP500*exchange_rate$SP500
price$IBOVESPA <- price$IBOVESPA*exchange_rate$IBOVESPA
# head(price)
```

Calculating daily returns

Since the stock market do not open on weekends and other holidays, we need to adjust those values by deleting the rows where the returns are 0.

[Hide](#)

```
daily_return <- (price[-1,-1]-price[-n,-1])/price[-n,-1]
daily_return <- cbind(price$date[-1],daily_return)
daily_return <- daily_return[!rowSums(daily_return[,-1]) == 0,]
colnames(daily_return)[1]="date"
daily_return$date <- as.Date(daily_return$date, "%Y/%m/%d")
```

```
unknown timezone '%Y/%m/%d'
```

[Hide](#)

```
#daily_return <- replace(daily_return, daily_return==0, NA)
#head(daily_return)
```

Calculating log daily return

As required, daily returns were converted to log returns and below showed daily log returns for each index, using the following formula:

$$\log_{\text{return}}_t = \log\left(\frac{p_t}{p_{t-1}}\right)$$

[Hide](#)

```
log_daily_return <- log(price[-1,-1]/price[-n,-1])
log_daily_return <- cbind(price$date[-1],log_daily_return)
log_daily_return <- log_daily_return[!rowSums(log_daily_return[,-1]) == 0,]

colnames(log_daily_return)[1]="date"
log_daily_return$date <- as.Date(log_daily_return$date, "%Y/%m/%d")
```

```
unknown timezone '%Y/%m/%d'
```

[Hide](#)

```
#log_daily_return <- replace(log_daily_return, log_daily_return==1, NA)
```

Mean

From the results shown below, TSX, DAX, SP500 and IBOVESPA have average log return greater than 1 meaning that their average returns are positive. In contrast, the other indexes have negative returns.

[Hide](#)

```
mean <- colMeans(log_daily_return[, -1])
mean
```

TSX	CAC	DAX	Eurostoxx50	NIKKEI225	FTSE100	S
P500						
1.017803e-04	-5.204957e-05	1.091897e-04	-8.958102e-05	-4.743869e-05	-5.839154e-05	7.502743e-05

IBOVESPA
3.457150e-05

Standard deviation

Standard deviation for SP500 is the smallest among all indexes, implying that SP500 tends to be less volatile than the other indexes.

[Hide](#)

```
std_dev = apply(log_daily_return[,-1], 2, sd)
std_dev
```

TSX	CAC	DAX	Eurostoxx50	NIKKEI225	FTSE100	SP500	IBOVESPA
PA							
0.01438993	0.01647415	0.01667664	0.01667836	0.01554133	0.01369582	0.01242561	0.02481279

Skewness

Apart from CAC, the other indexes are negatively skewed which indicates that investors may encounter frequent small gains from most indexes but there are possibilities for huge negative returns.

[Hide](#)

```
ske <- apply(log_daily_return[,-1], 2, skewness)
ske
```

TSX	CAC	DAX	Eurostoxx50	NIKKEI225	FTSE100	SP500
-0.716079187	0.021234718	-0.011294537	-0.006166816	-0.313879062	-0.143823011	-0.181856378
IBOVESPA						
-0.147429680						

Kurtosis

All indexes have kurtosis being far from 0 which demonstrate that distributions for returns have fat tails for all indexes. Therefore, under high kurtosis distribution, it is possible for investors to experience extreme returns.

[Hide](#)

```
kurtosis = apply(log_daily_return[,-1], 2, kurtosis)
kurtosis
```

TSX	CAC	DAX	Eurostoxx50	NIKKEI225	FTSE100	SP500	IBOVESPA
PA							
10.830890	8.463801	7.864342	7.873810	8.748768	10.072936	11.423217	8.302262

Autocorrelation

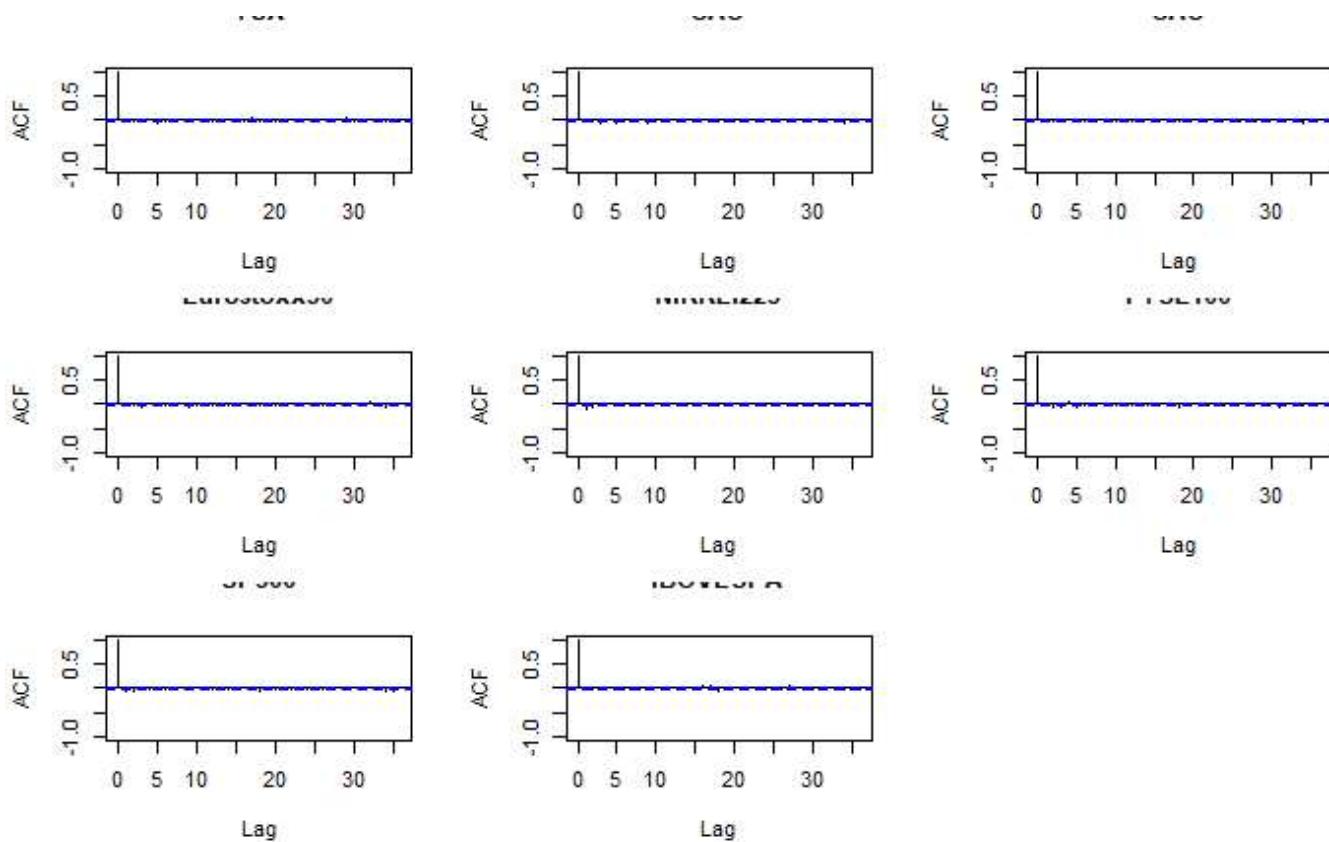
Graphs show positive and substantial first-order correlations and weak correlations for long term.

[Hide](#)

```
# autocorrelation for each index
# auto_cor = apply(log_daily_return[,-1], 2, acf)

# Convert data frame to time series
data_xts <- xts(log_daily_return, order.by = as.Date(log_daily_return$date),format = '%Y-%m-%d')
returns_daily_xts <- na.omit(data_xts[,-1])

# Plot autocorrelation for each index
title <- c( "TSX","CAC","CAC","Eurostoxx50", "NIKKEI225","FTSE100","SP500","IBOVESPA")
par(mfrow = c(3, 3))
for (i in 1:8) {
  acf(as.numeric(returns_daily_xts[,i]), plot=T,ylim = c(-1,1),main = title[i])
}
```


[Hide](#)

```
# acf(as.numeric(returns_daily_xts[,1]), plot=T,ylim = c(-0.2,0.2),main= "TSX")
# acf(as.numeric(returns_daily_xts[,2]), plot=T,ylim = c(-0.2,0.2),main= "CAC")
# acf(as.numeric(returns_daily_xts[,3]), plot=T,ylim = c(-1,1),main="CAC")
# acf(as.numeric(returns_daily_xts[,4]), plot=T,ylim = c(-0.2,0.2),main= "Eurostoxx50")
# acf(as.numeric(returns_daily_xts[,5]), plot=T,ylim = c(-0.2,0.2),main= "NIKKEI225")
# acf(as.numeric(returns_daily_xts[,6]), plot=T,ylim = c(-0.2,0.2),main= "FTSE100")
# acf(as.numeric(returns_daily_xts[,7]), plot=T,ylim = c(-1,1),main= "SP500")
# acf(as.numeric(returns_daily_xts[,8]), plot=T,ylim = c(-0.2,0.2),main= "IBOVESPA")
```

Correlation Matrix

Correlations are all positive and most of them are highly correlated except for NIKKEI225. NIKKEI225 is weakly correlated with the rest of the indexes especially SP500 which has correlation of 0.00064 with NIKKEI225. Overall, these indexes tend to move toward the same directions.

[Hide](#)

```
corr_matrix = cor(log_daily_return[,-1])
corr_matrix
```

	TSX	CAC	DAX	Eurostoxx50	NIKKEI225	FTSE100	SP500
TSX	1.0000000	0.6054480	0.6054209	0.6056300	0.1257192562	0.6268727	0.7396837963
CAC	0.6054480	1.0000000	0.9116861	0.9814592	0.2115118634	0.8709373	0.5466897845
DAX	0.6054209	0.9116861	1.0000000	0.9450664	0.1882519477	0.8151868	0.5929462576
Eurostoxx50	0.6056300	0.9814592	0.9450664	1.0000000	0.2002665797	0.8675177	0.5608888817
NIKKEI225	0.1257193	0.2115119	0.1882519	0.2002666	1.0000000000	0.2270514	0.0006400986
FTSE100	0.6268727	0.8709373	0.8151868	0.8675177	0.2270513698	1.0000000	0.5385025220
SP500	0.7396838	0.5466898	0.5929463	0.5608889	0.0006400986	0.5385025	1.0000000000
IBOVESPA	0.6036287	0.4828383	0.4974479	0.4873130	0.0593528591	0.4995873	0.5930348537
	IBOVESPA						
TSX	0.60362871						
CAC	0.48283827						
DAX	0.49744793						
Eurostoxx50	0.48731299						
NIKKEI225	0.05935286						
FTSE100	0.49958728						
SP500	0.59303485						
IBOVESPA	1.00000000						

Summary statistics

Below provides a table that summarises mean, standard deviation, skewness and kurtosis for each index.

[Hide](#)

```
summary_stat <- rbind(mean, std_dev,ske, kurtosis)
summary_stat
```

	TSX	CAC	DAX	Eurostoxx50	NIKKEI225	FTSE100
mean	0.0001017803	-5.204957e-05	0.0001091897	-8.958102e-05	-4.743869e-05	-5.839154e-05
std_dev	0.0143899267	1.647415e-02	0.0166766397	1.667836e-02	1.554133e-02	1.369582e-02
ske	-0.7160791867	2.123472e-02	-0.0112945368	-6.166816e-03	-3.138791e-01	-1.438230e-01
kurtosis	10.8308901815	8.463801e+00	7.8643416194	7.873810e+00	8.748768e+00	1.007294e+01
	SP500	IBOVESPA				
mean	7.502743e-05	0.0000345715				
std_dev	1.242561e-02	0.0248127897				
ske	-1.818564e-01	-0.1474296795				
kurtosis	1.142322e+01	8.3022617454				

Summary

[Hide](#)

```
summary(log_daily_return[, -1])
```

TSX	CAC	DAX	Eurostoxx50
Min. :-0.1270422	Min. :-1.147e-01	Min. :-0.0933772	Min. :-1.019e-01
1st Qu.:-0.0063287	1st Qu.:-8.234e-03	1st Qu.:-0.0080654	1st Qu.:-8.532e-03
Median : 0.0008451	Median : 2.865e-04	Median : 0.0005260	Median : 2.244e-04
Mean : 0.0001018	Mean :-5.205e-05	Mean : 0.0001092	Mean :-8.958e-05
3rd Qu.: 0.0071587	3rd Qu.: 8.650e-03	3rd Qu.: 0.0085780	3rd Qu.: 8.600e-03
Max. : 0.0866889	Max. : 1.248e-01	Max. : 0.1270912	Max. : 1.230e-01
NIKKEI225	FTSE100	SP500	IBOVESPA
Min. :-1.371e-01	Min. :-9.731e-02	Min. :-9.470e-02	Min. :-1.808e-01
1st Qu.:-8.149e-03	1st Qu.:-6.431e-03	1st Qu.:-5.126e-03	1st Qu.:-1.250e-02
Median : 3.386e-04	Median : 2.138e-04	Median : 1.818e-04	Median : 1.861e-04
Mean :-4.744e-05	Mean :-5.839e-05	Mean : 7.503e-05	Mean : 3.457e-05
3rd Qu.: 8.305e-03	3rd Qu.: 6.907e-03	3rd Qu.: 5.687e-03	3rd Qu.: 1.338e-02
Max. : 1.318e-01	Max. : 1.110e-01	Max. : 1.096e-01	Max. : 2.134e-01

Normality: Evaluating whether it is reasonable to approximate the return process by i.i.d. Gaussian distribution.

To examine whether it is reasonable to approximate the return process by i.i.d Gaussian distribution, distributions of log return and original return are plotted below. In addition, shapiro wilk test was conducted and according to the insignificant p-value resulting from the test, null hypothesis stating that return data are normally distributed can be rejected. In addition, distribution of returns are mostly negative skewed with high kurtosis which illustrates that the return data of the given indexes do not follow normal distribution. Therefore, it is not reasonable to approximate the return process by i.i.d Gaussian distribution.

[Hide](#)

```
# hgA <- hist(log_daily_return[,2], plot = FALSE) # Save first histogram data
# hgB <- hist(log_daily_return[,3], plot = FALSE) # Save 2nd histogram data
# plot(hgA, col = "red50") # Plot 1st histogram using a transparent color
# plot(hgB, col = "blue50", add = TRUE) # Add 2nd histogram using different color
# Log return plot
plot(density(log_daily_return[,2]), xlim= c(-0.2,0.2), ylim = c(0,100))
lines(density(log_daily_return[,3]))
```

[Hide](#)

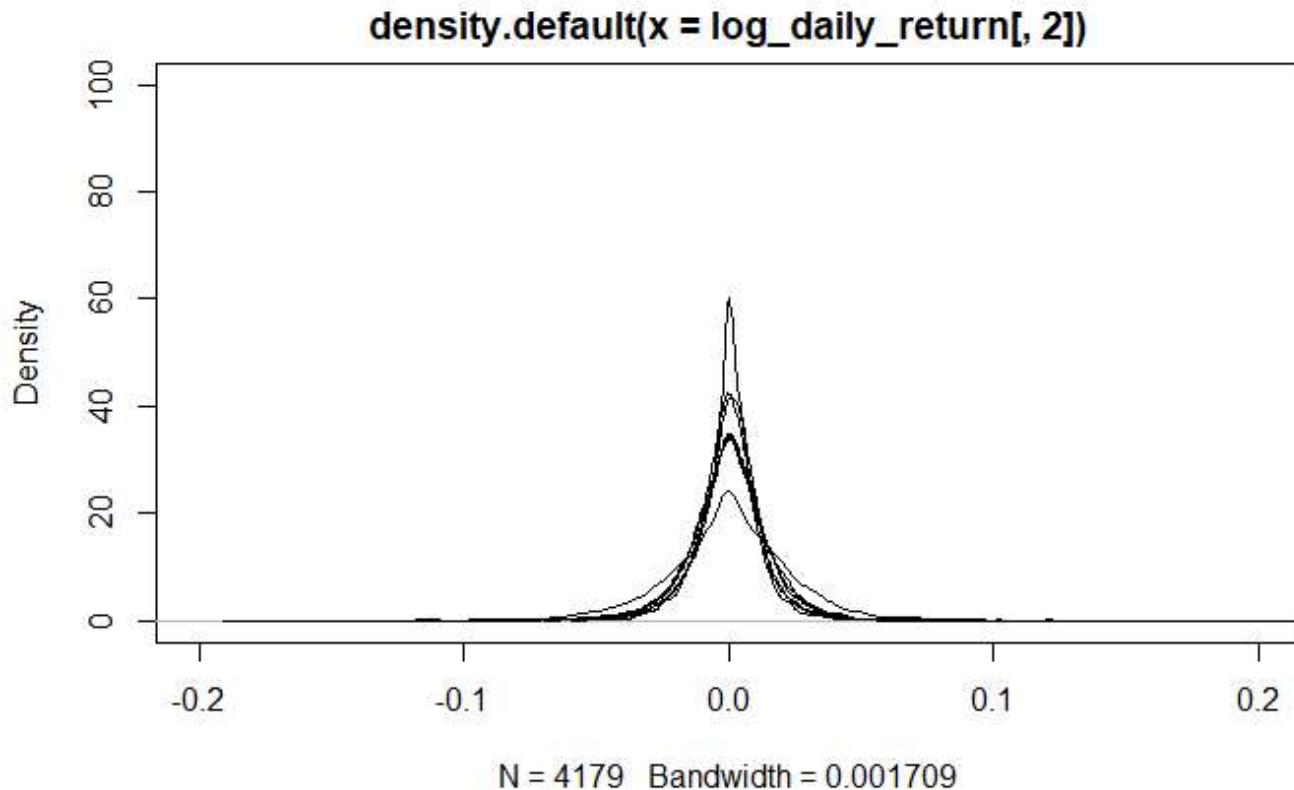
```
lines(density(log_daily_return[,4]))
lines(density(log_daily_return[,5]))
```

[Hide](#)

```
lines(density(log_daily_return[,6]))
lines(density(log_daily_return[,7]))
```

[Hide](#)

```
lines(density(log_daily_return[,8]))  
lines(density(log_daily_return[,9]))
```



[Hide](#)

```
# Normal return plot  
plot(density(daily_return[,2]),xlim= c(-0.2,0.2),ylim = c(0,100),col="red")  
lines(density(daily_return[,2]),col= "red")
```

[Hide](#)

```
lines(density(daily_return[,3]),col= "red")  
lines(density(daily_return[,4]),col= "red")
```

[Hide](#)

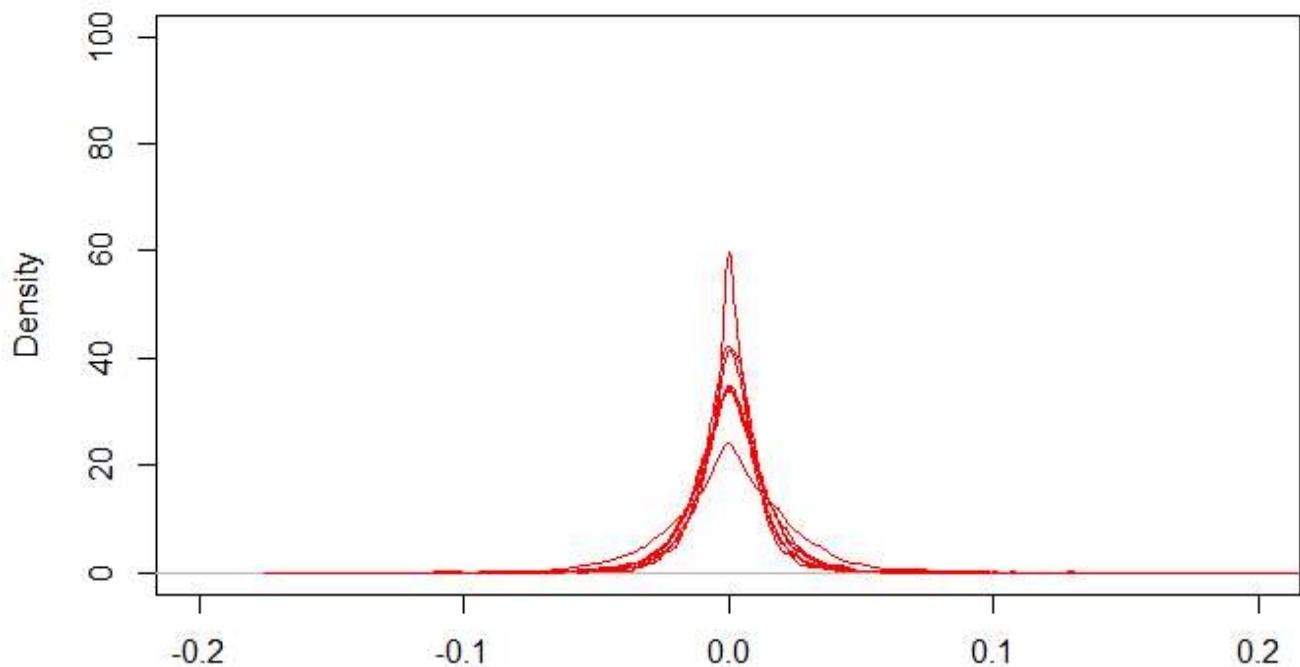
```
lines(density(daily_return[,5]),col= "red")  
lines(density(daily_return[,6]),col= "red")
```

[Hide](#)

```
lines(density(daily_return[,7]),col= "red")  
lines(density(daily_return[,8]),col= "red")
```

[Hide](#)

```
lines(density(daily_return[,9]),col= "red")
```

density.default(x = daily_return[, 2])[Hide](#)

```
# shapiro wilk test
for (i in 2:9){
  print(shapiro.test(log_daily_return[,i]))
}
```

```
Shapiro-Wilk normality test
```

```
data: log_daily_return[, i]  
W = 0.9113, p-value < 2.2e-16
```

```
Shapiro-Wilk normality test
```

```
data: log_daily_return[, i]  
W = 0.94319, p-value < 2.2e-16
```

```
Shapiro-Wilk normality test
```

```
data: log_daily_return[, i]  
W = 0.94853, p-value < 2.2e-16
```

```
Shapiro-Wilk normality test
```

```
data: log_daily_return[, i]  
W = 0.94624, p-value < 2.2e-16
```

```
Shapiro-Wilk normality test
```

```
data: log_daily_return[, i]  
W = 0.95254, p-value < 2.2e-16
```

```
Shapiro-Wilk normality test
```

```
data: log_daily_return[, i]  
W = 0.92929, p-value < 2.2e-16
```

```
Shapiro-Wilk normality test
```

```
data: log_daily_return[, i]  
W = 0.90964, p-value < 2.2e-16
```

```
Shapiro-Wilk normality test
```

```
data: log_daily_return[, i]  
W = 0.95354, p-value < 2.2e-16
```

Mean Variance Analysis

[Hide](#)

```
#install.packages("fPortfolio",repos = "http://cran.us.r-project.org")
#install.packages("fAssets",repos = "http://cran.us.r-project.org")
library(fAssets)
library(ggplot2)
library(fPortfolio)
library(quadprog) # for portfolio
```

Situation 1: short selling allowed

In this part, I assume risk free rate is 0 and there are 252 trading days each year.

[Hide](#)

```
nor_daily_return<- daily_return[-c(1)]
# annualised covariance matrix
nor_cov_matrix <- 252*cov(na.omit(nor_daily_return))
# annualised mean
nor_mean <- ((1+colMeans(nor_daily_return,na.rm = TRUE))^252)-1
```

Tangency portfolio is then calculated in two ways. (1) Manual calculation

[Hide](#)

```
knitr:::include_graphics("tangencyportfolioformula.png")
```

Tangency Portfolio Formula

$$w^* = \frac{\Sigma^{-1}\mu}{I\Sigma^{-1}\mu}$$

[Hide](#)

```
# Find the tangency portfolio
# using matrix to calculate portfolio weights
Sigma <- nor_cov_matrix
R <- nor_mean
w <- solve(Sigma) %*% R
w <- w/sum(w)
w
```

	[,1]
TSX	0.9587117
CAC	7.5191451
DAX	8.1501355
Eurostoxx50	-14.6189464
NIKKEI225	0.2028951
FTSE100	-1.0974022
SP500	-0.4806745
IBOVESPA	0.3661357

2. R package

[Hide](#)

```
# use solve.QP to get the weights
# solve quadratic problem
Dmat <- nor_cov_matrix
dvec <- matrix(nor_mean,nrow = 8,ncol = 1)
Amat <- t(matrix(1,1,8))
bvec <- c(1)
optimal_weights1 <-solve.QP(Dmat,dvec,Amat,bvec,meq = 1)
optimal_weights1$solution
```

```
[1] 0.9111773 7.1437011 7.7408865 -13.8975268 0.2091928 -1.0235026 -0.4276055 0.3
436772
```

As we can see, these two ways give us similar results of the tangency portfolio. Values of some assets are extremal. For example, Eurostoxx50 has almost 1400% short position. Then we should consider to add no short selling constraint.

Situation 2: short selling disallowed

[Hide](#)

```
# minimum variance portfolio
Amat_2 <- cbind(Amat,diag(8))
bvec_2 <- c(1,rep(0,8))

optimal_weights2 <- solve.QP(Dmat,dvec,Amat=Amat_2 ,bvec=bvec_2,meq = 1)
optimal_weights2$solution
```

```
[1] 2.298115e-01 -5.449416e-18 4.598038e-01 2.347856e-16 1.966925e-02 -4.320922e-16
[7] 4.676425e-17 2.907154e-01
```

As we can see, the constrained tangency portfolio is more realistic with reasonable weights of some assets comparing to the unconstrained tangency portfolio.

[Hide](#)

```
# Tangency portfolio
time_series_return=as.timeSeries(daily_return,na.rm = TRUE)
tangencyPortfolio(as.timeSeries(na.omit(time_series_return)),constraints = "LongOnly")
```

Title:

```
MV Tangency Portfolio
Estimator: covEstimator
Solver: solveRquadprog
Optimize: minRisk
Constraints: LongOnly
```

Portfolio Weights:

	TSX	CAC	DAX	Eurostoxx50	NIKKEI225	FTSE100	SP500	IBOVES
PA	0.2929	0.0000	0.3861	0.0000	0.1118	0.0000	0.0000	0.20
92								

Covariance Risk Budgets:

	TSX	CAC	DAX	Eurostoxx50	NIKKEI225	FTSE100	SP500	IBOVES
PA	0.2548	0.0000	0.4068	0.0000	0.0347	0.0000	0.0000	0.30
37								

Target Returns and Risks:

mean	Cov	CVaR	VaR
0.0002	0.0137	0.0326	0.0216

Description:

Sat Jan 01 20:29:27 2022 by user: Yuting Huang

Efficient frontier

The efficient frontier is plotted as follow. The blue dot represents the tangency portfolio and the red dot represents the minimum variance portfolio.

[Hide](#)

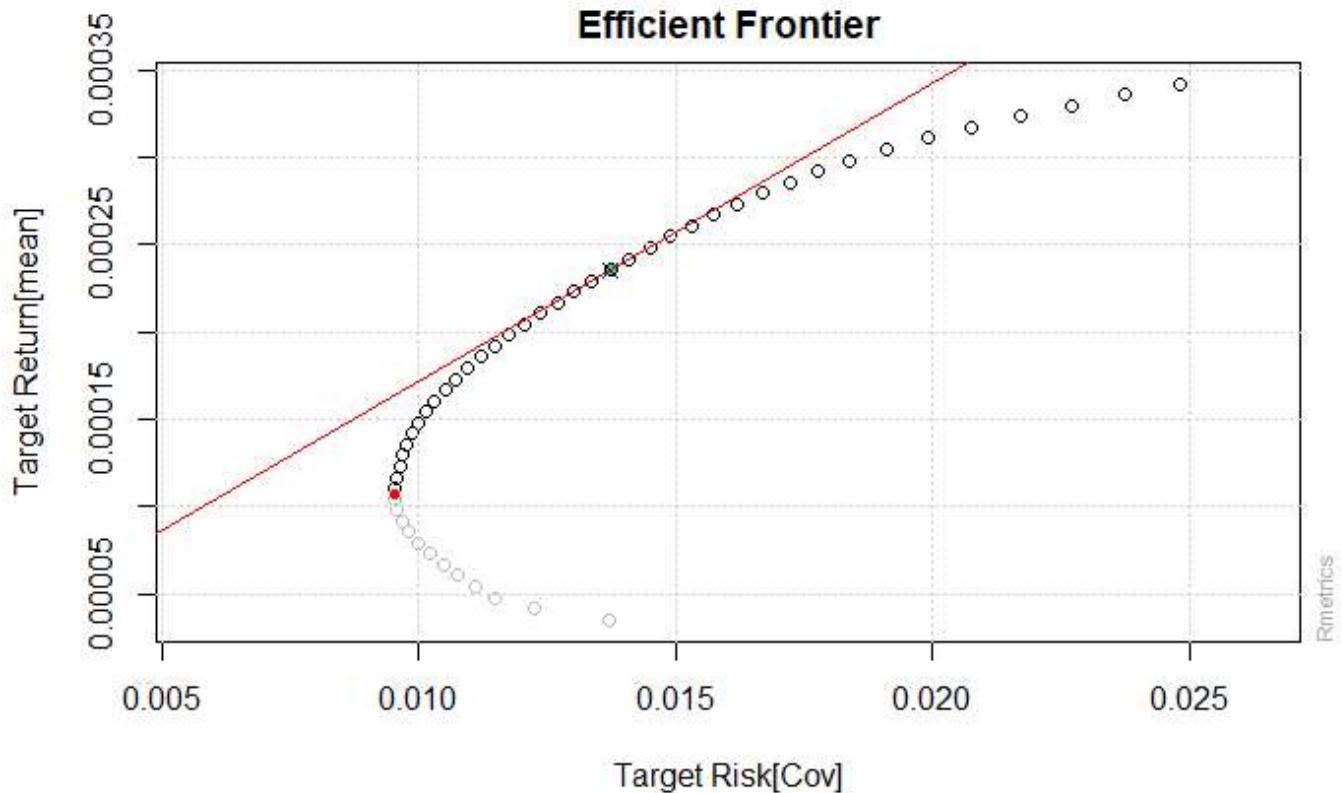
```
# plot Efficient frontier
frontier2 <- minvariancePortfolio(na.omit(time_series_return))
frontier_plot <- frontierPlot(portfolioFrontier(data = na.omit(time_series_return)))
grid()
```

[Hide](#)

```
minvariancePoints(frontier2,col="red",pch=20) # highlight the minimum variance point
tangencyPoints(frontier2,col="green", pch=20) # highlight the tangency portfolio point
```

[Hide](#)

```
cmlPoints(frontier2,col="blue",pch=4) # highlight the capital market portfolio point
tangencyLines(frontier2,col="red")
```



By adding the non-shorting-selling constraint, we get a more realistic efficient frontier with the tangency portfolio.

Mean Variance Analysis with 5-year Rolling Window

[Hide](#)

```
## Clean and sort data
library(quadprog) # for portfolio
## Compute list of means and list of covariance matrix based on 5-year rolling window
begindates = c("2000-01-01", "2001-01-01", "2002-01-01", "2003-01-01", "2004-01-01", "2005-01-01",
"2006-01-01", "2007-01-01", "2008-01-01", "2009-01-01", "2010-01-01", "2011-01-01")
enddates = c("2004-12-31", "2005-12-31", "2006-12-31", "2007-12-31", "2008-12-31", "2009-12-31", "2
010-12-31", "2011-12-31", "2012-12-31", "2013-12-31", "2014-12-31", "2015-12-31")
listofreturns = list()
for (i in 1:length(begindates))
  listofreturns[[i]] = daily_return %>% filter(daily_return$date >= begindates[i] & daily_re
turn$date <= enddates[i])

listofmeans = list()
for (i in 1:length(listofreturns))
{
  listofmeans[[i]] = colMeans(as.matrix(listofreturns[[i]][-c(1)]), na.rm = T)
}

listofcov = list()
for (i in 1:length(listofreturns))
{
  listofcov[[i]] = cov(na.omit(as.matrix(listofreturns[[i]][-c(1)])))
}

## Allow short sell
w_roll <- matrix(ncol = 12, nrow = 8) # Create an empty matrix
for (i in 1:12){
  w1 <- solve(listofcov[[i]])%*%listofmeans[[i]]
  w_roll[,i] <- w1
}
colnames(w_roll) <- c("Period 1", "Period 2", "Period 3", "Period 4", "Period 5", "Period 6", "Peri
od 7", "Period 8", "Period 9", "Period 10", "Period 11", "Period 12")
rownames(w_roll) <- c("TSX", "CAC", "DAX", "Eurostoxx50", "NIKKI225", "FTSE100", "SP500", "IBOVESPA"
)
w_roll
```

	Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Perio
d 7							
TSX 050	4.1840087	6.3974365	7.2659628	8.4241219	2.7342048	2.394710617	1.62840
CAC 567	3.2137957	5.0830877	14.6841832	8.9908462	15.6176545	10.648165542	10.18279
DAX 691	0.6202310	2.6962803	6.1199835	20.0588513	18.8296079	17.064565442	18.74386
Eurostoxx50 572	-3.4419838	-8.7137168	-21.0172040	-25.0190095	-29.4162234	-25.210619712	-27.28264
NIKKI225 139	-1.2671768	0.2690770	0.9411132	0.7322032	-0.5415366	-0.005925061	0.01905
FTSE100 580	-1.1568947	0.5091171	1.8495296	-2.1633700	-5.8136971	-3.549116418	-1.76735
SP500 503	-2.9786167	-4.1536484	-4.0829787	-8.1005873	-6.0492109	-6.077927089	-4.47375
IBOVESPA 296	0.7040895	1.0432900	1.4802570	3.3821111	2.1720535	3.133905022	2.62601
	Period 8	Period 9	Period 10	Period 11	Period 12		
TSX	1.413388	0.3529955	-2.324409	-6.8706565	-12.310494		
CAC	4.104049	9.2737123	1.429624	-0.7927085	8.824187		
DAX	13.035034	12.3584316	16.710219	17.0132637	12.475383		
Eurostoxx50	-17.381261	-21.3759792	-20.322297	-17.4776880	-19.949928		
NIKKI225	-0.163443	0.1703271	1.951409	1.6201618	1.917759		
FTSE100	-0.545854	-0.1778036	4.227455	1.7778940	0.422225		
SP500	-3.137504	-0.2906377	8.214361	17.7830449	18.775873		
IBOVESPA	1.950721	0.3819400	-1.266655	-3.5335960	-3.558255		

Find the minimum variance portfolio (Short sell allowed)

Hide

```
mv_roll <- matrix(ncol = 8, nrow = 12) # Create an empty matrix
for (i in 1:12){
  Dmat2 <- listofcov[[i]]
  dvec2 <- matrix(listofmeans[[i]],nrow = 8,ncol = 1)
  Amat2 <- t(matrix(1,1,8))
  bvec2 <- c(1)
  optimal_weights2 <- solve.QP(Dmat2,dvec2,Amat2,bvec2,meq = 1)
  mv_roll[i,] <- optimal_weights2$solution
}
rownames(mv_roll) <- c("2004-01-01","2005-01-01","2006-01-01","2007-01-01","2008-01-01","2009-01-01","2010-01-01","2011-01-01","2012-01-01","2013-01-01","2014-01-01","2015-01-01")
colnames(mv_roll) <- c("TSX","CAC","DAX","Eurostoxx50","NIKKI225","FTSE100","SP500","IBOVESPA")
mv_roll
```

	TSX	CAC	DAX	Eurostoxx50	NIKKI225	FTSE100	SP500
2004-01-01	4.4488166	3.459438	0.5018619	-3.803494	-1.02411852	-0.66482609	-2.61039837
2005-01-01	5.6868965	4.558579	2.9584256	-7.867628	-0.09697338	-0.47024509	-4.84745381
2006-01-01	5.6416082	13.616658	6.8955886	-18.604377	-0.16334633	-1.22798896	-6.89795928
2007-01-01	7.7599313	9.213608	21.0751974	-25.260479	-0.43706372	-3.87823071	-11.53355214
2008-01-01	2.4867823	15.031633	18.6846877	-28.922962	0.66174838	-5.03135466	-3.53658731
2009-01-01	2.1239531	10.741604	17.2292726	-25.802660	0.98809597	-2.84631171	-4.16748648
2010-01-01	1.5000756	10.136891	18.9843258	-27.702938	0.52576004	-1.35716754	-3.49226706
2011-01-01	1.2242739	3.701335	13.1035551	-17.497430	0.54523063	0.08918588	-1.85246372
2012-01-01	0.3134525	9.207376	12.3584639	-21.393429	0.29654158	-0.06009808	-0.06120247
2013-01-01	-1.5535800	2.908276	17.0782895	-19.401934	-0.67738301	0.66905048	2.47758872
2014-01-01	-6.7900370	2.768637	17.4983364	-18.207010	-0.91272075	-2.22036861	11.86629534
2015-01-01	-12.4001222	10.374604	13.2202134	-20.594530	0.24702599	-1.84215651	15.19525707
	IBOVESPA						
2004-01-01	0.6927199						
2005-01-01	1.0783989						
2006-01-01	1.7398174						
2007-01-01	4.0605889						
2008-01-01	1.6260525						
2009-01-01	2.7335324						
2010-01-01	2.4053203						
2011-01-01	1.6863130						
2012-01-01	0.3388962						
2013-01-01	-0.5003073						
2014-01-01	-3.0031325						
2015-01-01	-3.2002923						

Find the minimum variance portfolio (Short sell prohibited)

Hide

```
mv_roll_noshort <- matrix(ncol = 8, nrow = 12) # Create an empty matrix
for (i in 1:12){
  Dmat2 <- listofcov[[i]]
  dvec2 <- matrix(listofmeans[[i]],nrow = 8,ncol = 1)
  A_roll <- cbind(Amat2,diag(8))
  b_roll <- c(1,rep(0,8))
  optimal_weights_roll <- solve.QP(Dmat2,dvec2,Amat=A_roll ,bvec=b_roll,meq = 1)
  mv_roll_noshort[i,] <- optimal_weights_roll$solution
}
rownames(mv_roll_noshort) <- c("2004-01-01","2005-01-01","2006-01-01","2007-01-01","2008-01-01",
"2009-01-01","2010-01-01","2011-01-01","2012-01-01","2013-01-01","2014-01-01","2015-01-01")
colnames(mv_roll_noshort) <- c("TSX","CAC","DAX","Eurostoxx50","NIKKI225","FTSE100","SP500",
"IBOVESPA")
mv_roll_noshort
```

	TSX	CAC	DAX	Eurostoxx50	NIKKI225	FTSE100
0						
2004-01-01	7.345322e-01	0.000000e+00	8.739756e-18	9.292506e-17	0.000000e+00	4.175243e-18
2005-01-01	3.322113e-01	4.021482e-16	0.000000e+00	9.930987e-17	0.000000e+00	9.390240e-17
2006-01-01	0.000000e+00	-2.782394e-16	0.000000e+00	1.023092e-15	1.006762e-18	-1.064751e-15
2007-01-01	0.000000e+00	-8.530112e-16	0.000000e+00	-3.552714e-15	-2.220446e-16	1.511744e-15
2008-01-01	3.145934e-17	-3.127955e-16	0.000000e+00	-4.267272e-16	0.000000e+00	-1.007101e-16
2009-01-01	1.386231e-16	3.527187e-16	-6.282960e-16	-4.501200e-16	5.551115e-17	-7.940919e-17
2010-01-01	1.401850e-16	-3.692480e-16	-1.889899e-16	3.452144e-15	0.000000e+00	2.298619e-16
2011-01-01	-1.194884e-16	-2.048884e-16	-1.092725e-17	1.149127e-16	6.811785e-02	4.200896e-17
2012-01-01	6.479541e-17	2.710073e-16	1.353495e-18	3.540906e-15	3.856086e-01	2.581787e-17
2013-01-01	1.886016e-16	1.217723e-15	2.420195e-01	0.000000e+00	0.000000e+00	-2.470229e-16
2014-01-01	-1.161364e-17	2.143906e-15	-3.036325e-17	9.089410e-17	0.000000e+00	2.340144e-15
2015-01-01	2.494700e-15	1.852011e-16	-6.592304e-17	0.000000e+00	0.000000e+00	2.892569e-16
	SP500	IBOVESPA				
2004-01-01	-1.251173e-16	2.654678e-01				
2005-01-01	-1.670777e-16	6.677887e-01				
2006-01-01	1.409136e-16	1.000000e+00				
2007-01-01	4.808872e-16	1.000000e+00				
2008-01-01	0.000000e+00	1.000000e+00				
2009-01-01	9.529079e-16	1.000000e+00				
2010-01-01	-3.213580e-16	1.000000e+00				
2011-01-01	-7.396681e-17	9.318821e-01				
2012-01-01	3.994546e-01	2.149368e-01				
2013-01-01	7.579805e-01	3.989790e-18				
2014-01-01	1.000000e+00	-7.703584e-16				
2015-01-01	1.000000e+00	3.682661e-16				

Examine if this portfolio is significantly time-varying

From the graph below, it is obvious that index weights within mean-variance portfolio that allows short selling vary to a larger extent than non-shortselling portfolio. Moreover, ranges of weights of NIKKI225 and Eurostoxx50 are a lot greater than the rest. For non-shortselling portfolio, index weights from 2006 to 2011 appeared to be relatively stable while for shortselling portfolio, weights became stable after 2008.

```
### M-V (no short selling)
# Organise data
mv_roll_noshort_df <- as.data.frame(mv_roll_noshort)
mydf <- cbind(rownames(mv_roll_noshort_df), mv_roll_noshort_df)
rownames(mydf) <- NULL
colnames(mydf) <- c("Year", "TSX", "CAC", "DAX", "Eurostoxx50", "NIKKI225", "FTSE100", "SP500", "IBOVESPA")
# convert to timeseries
mv_roll_noshort_xts <- xts(mydf, order.by = as.Date(mydf$Year), format = '%Y-%m-%d' )
zoo.basket <- as.zoo(mv_roll_noshort_xts)
tsRainbow <- rainbow(ncol(zoo.basket))
plot(x = zoo.basket, ylab = "Weight", main = "Weight for non-short selling portfolio", col = tsRainbow, screens = 1, ylim = c(-3,3))
```

NAs introduced by coercion

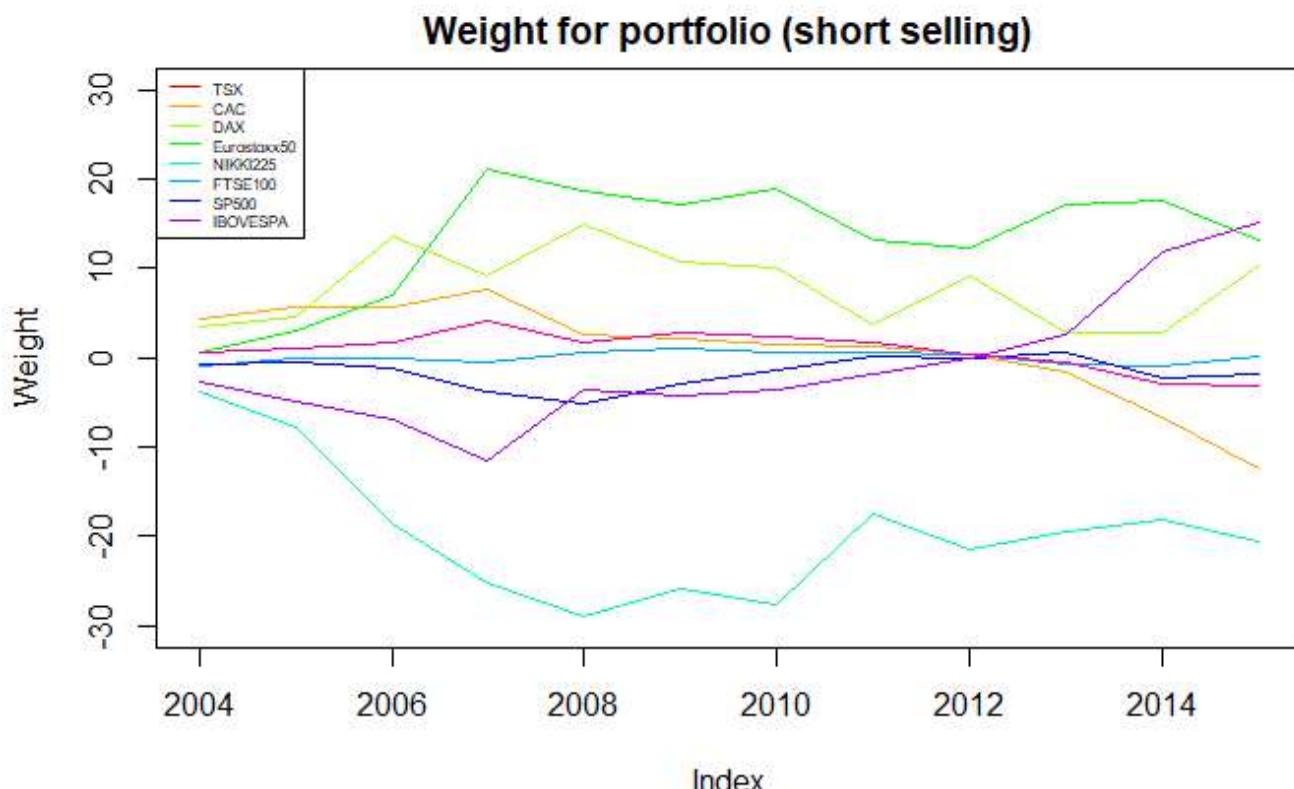
[Hide](#)

```
legend(x = 'topleft', legend = c("TSX", "CAC", "DAX", "Eurostoxx50", "NIKKI225", "FTSE100", "SP500", "IBOVESPA"),
       lty = 1, col = tsRainbow, cex = 0.5)
```



[Hide](#)

```
### M-V (short selling allowed)
# Organise data
mv_roll_df <- as.data.frame(mv_roll)
mydf2 <- cbind(rownames(mv_roll_df), mv_roll_df)
rownames(mydf2) <- NULL
colnames(mydf2) <- c("Year", "TSX", "CAC", "DAX", "Eurostoxx50", "NIKKI225", "FTSE100", "SP500", "IBOVESPA")
# convert to timeseries
mv_roll_xts <- xts(mydf2, order.by = as.Date(mydf2$Year), format = '%Y-%m-%d' )
zoo.basket2 <- as.zoo(mv_roll_xts)
tsRainbow2 <- rainbow(ncol(zoo.basket2))
plot(x = zoo.basket2, ylab = "Weight", main = "Weight for portfolio (short selling)", col = tsRainbow, screens = 1, ylim = c(-30,30))
legend(x = 'topleft', legend = c("TSX", "CAC", "DAX", "Eurostoxx50", "NIKKI225", "FTSE100", "SP500", "IBOVESPA"),
       lty = 1, col = tsRainbow2, cex = 0.5)
```



Compare Market-based weights and Mean-Variance weights

Weights calculated by market capital and mean-variance approaches varies in a large degree. By allowing short selling, composition weight of portfolio deviates from strategies that prohibit short selling even more.

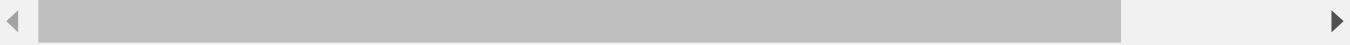
[Hide](#)

```
Mktcap_data <- read_excel("PS1_data.xls", sheet ="market_cap", skip = 1)
Mktcap_weight <- Mktcap_data[5,]

# Compare market-cap weights and mean-variance weights
MV_weight_short <- optimal_weights1$solution
MV_weight_noshort <- optimal_weights2$solution
comparison <- rbind(Mktcap_weight,MV_weight_short,MV_weight_noshort)
comparison <- as.data.frame(comparison)
rownames(comparison) <- c("Market-Cap Weight","Mean-Variance Weight (with short sale)","Mean-Variance Weight (without short sale)" )
comparison
```

	SPTSX INDEX <dbl>	CAC INDEX <dbl>	DAX INDEX <dbl>
Market-Cap Weight	0.04183375	0.04317955	0.03619743
Mean-Variance Weight (with short sale)	0.91117731	7.14370107	7.74088650
Mean-Variance Weight (without short sale)	-12.40012217	10.37460422	13.22021337

3 rows | 1-5 of 8 columns



[Hide](#)

NA