

# Quantitative Finance

## Lab - “Portfolio Management”

Fiona Martin, Hedi Sagar, Nathan Sanglier

Version: 03 Mar 2024

### I/ Black-Litterman Model

1. Read He and Litterman’s article carefully.
2. Using the course notes, reproduce two examples from the article and compare the results with those obtained with the BLCOP package.
3. Compare with a traditional MV allocation.

#### Data

```
# Correlation Matrix (raw format)
corr_data =
'1,0.4880,0.4780,0.5150,0.4390,0.5120,0.4910
 0.4880,1,0.6640,0.6550,0.3100,0.6080,0.7790
 0.4780,0.6640,1,0.8610,0.3550,0.7830,0.6680
 0.5150,0.6550,0.8610,1,0.3540,0.7770,0.6530
 0.4390,0.3100,0.3550,0.3540,1,0.4050,0.3060
 0.5120,0.6080,0.7830,0.7770,0.4050,1,0.6520
 0.4910,0.7790,0.6680,0.6530,0.3060,0.6520,1'

# Correlation Matrix (matrix format)
corr_mat = matrix(as.double(spl(gsub('\n', ',', corr_data), ',')),
                  nrow = length(spl(corr_data, '\n')),
                  byrow=TRUE)

# Number of Assets in the Universe
N = nrow(corr_mat)

# Names of Assets in the Universe
asset_names = c('Australia', 'Canada', 'France', 'Germany', 'Japan', 'UK', 'USA')

# Standard Deviations of Assets Returns
stdevs_array = c(16.0, 20.3, 24.8, 27.1, 21.0, 20.0, 18.7)/100

# Prior Covariance Matrix of returns
Sigma = corr_mat * (stdevs_array %*% t(stdevs_array))
```

```

# Uncertainty over CAPM prior
tau = 0.05

# Risk-Aversion Parameter
delta = 2.5

# Market Capitalization Weights at Equilibrium
w_eq = c(1.6, 2.2, 5.2, 5.5, 11.6, 12.4, 61.5)/100

```

## Question 2.

### 1st Example :

In this first example, we assume the German market will outperform the remaining European market by 5% a year, and the Canadian equities will outperform US equity by 3% a year.

Let us suppose the returns on the assets is a Gaussian random vector :  $R \sim \mathcal{N}(\mu, \Sigma)$ , with  $\mu$  the parameter to estimate.

Assume investors all have quadratic utility function :  $U(w) = w^T \Pi - \frac{\delta}{2} w^T \Sigma w$ .

Then, by maximizing Utility, we find that the equilibrium risk premiums on the assets are  $\Pi = \delta \Sigma w_{eq}$ .

```

# Equilibrium Risk Premiums
PI = delta * Sigma %*% w_eq

```

We consider we start at the equilibrium, i.e.  $\mu = \Pi + \varepsilon^{(e)}$  with  $\varepsilon^{(e)} \sim \mathcal{N}(0, \tau \Sigma)$ ,  $\tau$  representing the uncertainty over CAPM prior.

Table 1: Data.  $\sigma$ : Asset Volatility,  $w_{eq}$ : Initial weights at CAPM equilibrium,  $\Pi$ : Equilibrium Risk Premium

	$\sigma$	$w_{eq}$	$\Pi$
Australia	16.0	1.6	3.9
Canada	20.3	2.2	6.9
France	24.8	5.2	8.4
Germany	27.1	5.5	9.0
Japan	21.0	11.6	4.3
UK	20.0	12.4	6.8
USA	18.7	61.5	7.6

Now, we'll add some information to the prior and suppose we have the following views on some portfolios (it can be summarized as  $P\mu = Q + \varepsilon^{(v)}$  where each row of  $P$  is a portfolio,  $Q$  contains the returns expected by the analyst on these portfolios, and  $\varepsilon^{(v)} \sim \mathcal{N}(0, \Omega)$ ) :

```

# Investor Views
p1 = c(0, 0, -29.5, 100, 0, -70.5, 0)/100
q1 = 5/100

p2 = c(0, 100, 0, 0, 0, 0, -100)/100
q2 = 3/100

```

```
# Portfolios of Investor Views
P = rbind(p1, p2)
Q = c(q1, q2)
```

To simplify, we assume the covariance matrix of noise on views is diagonal (even if it is not true in reality) with diagonal terms corresponding to the return prior variance on each portfolio view, affected by the uncertainty coefficient  $\tau$  :  $\Omega = \text{Diag}(P(\tau\Sigma)P^T)$ .

```
# Covariance Matrix of Noise on Views
Omega = matrix(0, nrow = nrow(P), ncol = nrow(P))
diag(Omega) = diag(tau * P %%% Sigma %%% t(P))
```

Moreover, let us assume  $\varepsilon^{(e)}$  and  $\varepsilon^{(v)}$  are independent, thus

$$\begin{pmatrix} \varepsilon^{(e)} \\ \varepsilon^{(v)} \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} \tau\Sigma & 0 \\ 0 & \Omega \end{pmatrix}\right)$$

Then, rearranging the terms, we can combine the 2 equations :

$$\begin{pmatrix} \Pi \\ Q \end{pmatrix} = \begin{pmatrix} I \\ P \end{pmatrix} \mu + \begin{pmatrix} \varepsilon^{(e)} \\ \varepsilon^{(v)} \end{pmatrix}$$

We can apply the Generalized Least Squares method (note that Ordinary Least Squares method does not work here as the error term has heteroscedasticity) to find a non-biased estimation  $\hat{\mu}$  of  $\mu$  :  $\hat{\mu} = \left(\frac{1}{\tau}\Sigma^{-1} + P^T\Omega^{-1}P\right)^{-1} \left(\frac{1}{\tau}\Sigma^{-1}\Pi + P^T\Omega^{-1}Q\right)$ .

The covariance matrix of the estimator is given by :  $\text{Var}[\hat{\mu}] = M^{-1} = \left(\frac{1}{\tau}\Sigma^{-1} + P^T\Omega P\right)^{-1}$ .

Thus, according to the Law of Total Variance, the posterior distribution of returns is given by :  $R \sim \mathcal{N}(\hat{\mu}, \bar{\Sigma})$  with  $\bar{\Sigma} = \Sigma + M^{-1}$ .

```
Sigma_inv = solve(Sigma)
Omega_inv = solve(Omega)
M = 1/tau * Sigma_inv + t(P) %%% Omega_inv %%% P
M_inv = solve(M)
mu_hat = M_inv %%% (1/tau * Sigma_inv %%% PI + t(P) %%% Omega_inv %%% Q)
Sigma_bar = Sigma + M_inv
```

Then, we can find the optimal weights for the  $N$  assets using the classical Mean-Variance Optimization, here maximizing utility :  $\max U(w) = w^T \hat{\mu} - \frac{\delta}{2} w^T \bar{\Sigma} w$ .

We find  $w^* = \frac{1}{\delta} \bar{\Sigma}^{-1} \hat{\mu}$ .

```
# Optimal Weights
w_star = 1/delta * solve(Sigma_bar, mu_hat)
```

Moreover, using Matrix Inversion Lemma, it can be shown that  $\bar{\Sigma}M^{-1} = \frac{\tau}{1+\tau} \left(I - P^T A^{-1} P \frac{\Sigma}{1+\tau}\right)$ , where  $A = \frac{\Omega}{\tau} + P \frac{\Sigma}{1+\tau} P^T$ .

Then, we can deduce  $w^* = \frac{1}{1+\tau} (w_e q + P^T \Lambda)$  where  $\Lambda = \tau \Omega^{-1} \frac{Q}{\delta} - A^{-1} P \frac{\Sigma}{1+\tau} w_{eq} - A^{-1} P \frac{\Sigma}{1+\tau} P^T \tau \Omega^{-1} \frac{Q}{\delta}$ .

Thus, the investor optimal portfolio is represented by the initial equilibrium portfolio and then changed by a weighted sum of the views portfolios whose weights are given by the array  $\Lambda$ , the whole scaled by  $\frac{1}{1+\tau}$ .

Let us calculate the array  $\Lambda$  as it gives the weight of each portfolio view :

```

A = Omega/tau + 1/(1+tau) * P %*% Sigma %*% t(P)
A_inv = solve(A)
# Weight of each portfolio view in the optimal weights
Lambda = tau * Omega_inv %*% Q/delta - 1/(1+tau) * A_inv %*% P %*% Sigma %*% w_eq
Lambda = Lambda - tau/(1+tau) * A_inv %*% P %*% Sigma %*% t(P) %*% Omega_inv %*% Q/delta

```

In summary, we obtain the following results (Table 5 in the article) :

Table 2: Solution with View 1.  $P$ : view matrix,  $\bar{\mu}$ : ex-post expected return,  $w^*$ : optimal weights,  $\frac{W_{eq}}{1+\tau}$ : Scaled equilibrium weights

	$P_1$	$P_2$	$\hat{\mu}$	$w^*$	$w^* - \frac{W_{eq}}{1+\tau}$
Australia	0.000	0.000	4.4	1.5	0.0
Canada	0.000	100.000	8.7	41.9	39.8
France	-29.500	0.000	9.5	-3.4	-8.4
Germany	100.000	0.000	11.2	33.6	28.3
Japan	0.000	0.000	4.6	11.0	0.0
UK	-70.500	0.000	7.0	-8.2	-20.0
USA	0.000	-100.000	7.5	18.8	-39.8
$q$	5.000	3.000	NA	NA	NA
$\omega/\tau$	0.021	0.017	NA	NA	NA
$\lambda$	0.298	0.418	NA	NA	NA

We can see that the weights  $w^*$  have changed compared to the initial ones based on the additional information given by the views. Compared to equilibrium weights, we see we especially allocate more wealth to Germany and Canada, which is logic since they are expected to outperform in the portfolios they are defined. Moreover, we remark the weight on the second portfolio view (given by  $\lambda$  values of  $\Lambda$ ) is higher than the one on the first view, in part due to the fact that the confidence in this view is better (represented by  $\frac{\omega}{\tau}$ ,  $\omega$  being the diagonal values of  $\Omega$ ).

Now, let's check we retrieve the same results using directly the BLCOP package.

First, let's express our views. Notice that "confidences" parameter represents the inverse of  $\Omega$  diagonal values, i.e. it is the inverse of variance associated with each view. This means that our  $\Omega$  matrix will be the same as the one we computed before.

```

# Expression of views
views = BLViews(P=P, q=Q, confidences=1/diag(Omega), assetNames=asset_names)

```

Then, let's compute the posterior distribution of returns.

```

# Posterior Distribution of returns
# post_distribution = posteriorEst(views=views, sigma=Sigma, mu=Pi, tau=tau)
# Sigma_bar = post_distribution@posteriorCovar
# mu_hat = post_distribution@posteriorMean

```

Finally, let's calculate the optimal weights, as before :

Let's verify the results :

Table 3: BLCOP Solution with View 1.  $P$ : view matrix,  $\bar{\mu}$ : ex-post expected return,  $w^*$ : optimal weights,  $\frac{W_{eq}}{1+\tau}$ : scaled equilibrium weights

	$P_1$	$P_2$	$\hat{\mu}$	$w^*$	$w^* - \frac{W_{eq}}{1+\tau}$
Australia	0.000	0.000	4.4	1.5	0.0
Canada	0.000	100.000	8.7	41.9	39.8
France	-29.500	0.000	9.5	-3.4	-8.4
Germany	100.000	0.000	11.2	33.6	28.3
Japan	0.000	0.000	4.6	11.0	0.0
UK	-70.500	0.000	7.0	-8.2	-20.0
USA	0.000	-100.000	7.5	18.8	-39.8
$q$	5.000	3.000	NA	NA	NA
$\omega/\tau$	0.021	0.017	NA	NA	NA
$\lambda$	0.298	0.418	NA	NA	NA

CHECK THE RESULTS !!!!!!!!!!!!!!!

## 2nd Example :

In this second example, we assume the German market will outperform the European market by 5% a year, and the Canadian equities will outperform US equity by 4% a year (instead of 3% in the previous example).

Let's repeat the same steps as before (what changes here is  $Q$ , and all the variables derived from it), and we get the results (Table 6 in the article).

```

q2 = 4/100
Q = c(q1, q2)

mu_hat = M_inv %*% (1/tau * Sigma_inv %*% PI + t(P) %*% Omega_inv %*% Q)
Sigma_bar = Sigma + M_inv

w_star = 1/delta * solve(Sigma_bar, mu_hat)

Lambda = tau * Omega_inv %*% Q/delta - 1/(1+tau) * A_inv %*% P %*% Sigma %*% w_eq
Lambda = Lambda - tau/(1+tau) * A_inv %*% P %*% Sigma %*% t(P) %*% Omega_inv %*% Q/delta

```

Table 4: Solution with View 2.  $P$ : view matrix,  $\bar{\mu}$ : ex-post expected return,  $w^*$ : optimal weights,  $\frac{W_{eq}}{1+\tau}$ : deviation from scaled equilibrium weights

	$P_1$	$P_2$	$\hat{\mu}$	$w^*$	$w^* - \frac{W_{eq}}{1+\tau}$
Australia	0.000	0.000	4.4	1.5	0.0
Canada	0.000	100.000	9.1	53.3	51.3
France	-29.500	0.000	9.5	-3.3	-8.2
Germany	100.000	0.000	11.3	33.1	27.8
Japan	0.000	0.000	4.6	11.0	0.0
UK	-70.500	0.000	7.0	-7.8	-19.6
USA	0.000	-100.000	7.3	7.3	-51.3
$q$	5.000	4.000	NA	NA	NA
$\omega/\tau$	0.021	0.017	NA	NA	NA
$\lambda$	0.292	0.538	NA	NA	NA

We can see that the weights  $w^*$  have changed compared to the previous example. We can notice that the weight on Canada has increased, which is logic since we're more bullish about it than in the previous example (we expect Canada to outperform USA by 4% instead of 3% previously).

Then, let's use the BLCOP package :

```
views = BLViews(P=P, q=Q, confidences=1/diag(Omega), assetNames=asset_names)

# post_distribution = posteriorEst(views=views, sigma=Sigma, mu=as.numeric(Pi), tau=tau)
# Sigma_bar = post_distribution@posteriorCovar
# mu_hat = post_distribution@posteriorMean

w_star = 1/delta * solve(Sigma_bar, mu_hat)
```

Table 5: BLCOP Solution with View 2.  $P$ : view matrix,  $\bar{\mu}$ : ex-post expected return,  $w^*$ : optimal weights,  $\frac{W_{eq}}{1+\tau}$ : scaled equilibrium weights

	$P_1$	$P_2$	$\hat{\mu}$	$w^*$	$w^* - \frac{W_{eq}}{1+\tau}$
Australia	0.000	0.000	4.4	1.5	0.0
Canada	0.000	100.000	9.1	53.3	51.3
France	-29.500	0.000	9.5	-3.3	-8.2
Germany	100.000	0.000	11.3	33.1	27.8
Japan	0.000	0.000	4.6	11.0	0.0
UK	-70.500	0.000	7.0	-7.8	-19.6
USA	0.000	-100.000	7.3	7.3	-51.3
$q$	5.000	4.000	NA	NA	NA
$\omega/\tau$	0.021	0.017	NA	NA	NA
$\lambda$	0.292	0.538	NA	NA	NA

CHECK THE RESULTS !!!!!!!!!!!!!

### Question 3.

WHAT DO WE NEED TO DO ??

## II/ Multi-Factor Mean-Variance Model

To remedy the fragility of a covariance matrix estimated on historical data, we propose to explore various techniques for obtaining a more robust estimate, and to observe the effect of these estimates on the solution of a classical mean-variance model.

In concrete terms, we propose to compare three approaches to constructing the covariance matrix: - classical estimation using historical data - robust estimation using statistical factors - estimation using Fama-French factors

### Data

#### Fama-French Factors

The monthly factors of the classic three-factor model are available on the K. French website:

```
# Load source file
FF.file <- file.path(get.data.folder(), "FFdownload.rda")
if(!file.exists(FF.file)) {
  tempf <- tempfile(fileext = ".RData")
  inputlist <- c("F-F_Research_Data_Factors")
  FFdownload(output_file = FF.file, inputlist=inputlist)
}
load(FF.file)

# Recover Fama-French 3 factors monthly returns since 1960
ts.FF <- FFdownload$x_F-F_Research_Data_Factors$monthly$Temp2["1960-01-01/", c("Mkt.RF", "SMB", "HML")]
ts.FF <- timeSeries(ts.FF, as.Date(time(ts.FF)))
```

#### NASDAQ Returns

Our study will focus on the universe of NASDAQ assets. We'll remove assets with too much volatility in order to keep good data.

```
# Get daily returns from source folder
folder <- 'NASDAQ'
tickers <- get.tickers(folder)
ts.all <- get.all.ts(folder, tickers, dt.start = dmy('01Mar2007'), combine = TRUE)

# We remove assets with too much volatility

# Assets volatilities
sigma = colSds(ts.all)

# Above 3 standard deviations of mean volatility,
# we consider the assets as outliers and remove them
idx <- which((sigma-mean(sigma)) > 3*sqrt(var(sigma)))
while(length(idx)>0) {
  ts.all <- ts.all[,-idx]
  sigma = colSds(ts.all)
  idx <- which((sigma-mean(sigma)) > 3*sqrt(var(sigma)))
}
```

```
# Number of assets
N = ncol(ts.all)
```

## Risk-free Asset

The riskfree asset returns are given on the FED website.

```
# Get returns from source
file.path <- file.path(get.data.folder(), "DP_LIVE_01032020211755676.csv")
tmp <- read.csv(file.path, header=TRUE, sep=";")[, c("TIME", "Value")]
dt <- ymd(paste(tmp$TIME, "-01", sep=""))

rf_rate <- timeSeries(data=tmp$Value/(100.0*12), dt)
colnames(rf_rate) <- "Rf"
```

## II.a) Mean-Variance Model with Historical Covariance Matrix

We'll do all our calculations on monthly data.

1. Convert daily returns into monthly ones.
2. Choose a 36-month interval and calculate the covariance matrix. Check that the matrix is positive definite and make any necessary corrections.
3. Calculate the tangent portfolio and present the solution numerically and graphically. What do you think ?

### Question 1.

Let's convert our daily returns to monthly returns, using "PerformanceAnalytics" library.

```
month_rets = apply.monthly(ts.all, FUN=colSums)
```

### Question 2.

We'll select the last 36 months of NASDAQ observations to calculate our covariance matrix. Be careful, the end date of the riskless asset returns data is not the same as the ones for the NASDAQ. Moreover, the riskless asset returns data is already monthly and the dates are alike "YYYY-MM-01". Thus we need to change the dates of our monthly returns to correspond to the way the ones of the risk-free rates are displayed, and select the 36 months corresponding for both datasets (notice there are not missing values in riskfree rate dataset, and also in monthly returns dataset by construction).

```
# Window of monthly returns selected
month_rets_select = tail(month_rets, 36)
# Flooring dates of monthly returns
dates_window = floor_date(ymd(time(month_rets_select)), 'month')
time(month_rets_select) = as.Date(dates_window)

# Window of riskfree rates selected
rf_row_dates <- as.Date(row.names(rf_rate))
rf_rate_select = rf_rate[rf_row_dates %in% as.Date(dates_window), ]
```



```
# Covariance Matrix
Sigma = cov(month_rets_select)
```

Let's check that the covariance matrix is positive definite :

```
res = is.positive.definite(Sigma)
if(res) {
  print("The matrix is positive definite.")
} else {
  print("The matrix is not positive definite.")
}
```

```
## [1] "The matrix is not positive definite."
```

The matrix is not positive definite, thus let's add a small perturbation term  $\epsilon I$  to make it positive definite.

```
epsilon = 10^(-13)
Sigma = Sigma + epsilon * diag(nrow=nrow(Sigma), ncol=nrow(Sigma))
res = is.positive.definite(Sigma)
if(res) {
  print("The matrix is positive definite.")
} else {
  print("The matrix is not positive definite.")
}
```

```
## [1] "The matrix is positive definite."
```

Now, our covariance matrix is well defined.

### Question 3.

In the Mean-Variance framework, the tangent portfolio is the one that maximizes the Sharpe Ratio :

$$\max SR(w) = \frac{w^T \mu - R_f}{\sqrt{(w^T \Sigma w)}}, \text{ st } \mathbf{1}^T w = 1.$$

However this problem is not quadratic, so instead let's notice that any portfolio on the CML is a combination of tangent portfolio and riskless asset. Since we're interested in finding the tangent portfolio, we can find it by finding any portfolio on the CML with an excess return wrt risk-free rate  $\tilde{\mu}_p : w^T \mu + (1 - \mathbf{1}^T w) R_f - R_f = \tilde{\mu}_p \Leftrightarrow w^T (\mu - R_f \mathbf{1}) = \tilde{\mu}_p$ .

Now, since the excess return is fixed, we just need to  $\min \frac{1}{2} w^T \Sigma w$ , with the constraint above. Solving this quadratic problem, we get  $w = \tilde{\mu}_p \frac{\Sigma^{-1} \tilde{\mu}}{\tilde{\mu}^T \Sigma^{-1} \tilde{\mu}}$ , where  $\tilde{\mu} = \mu - R_f \mathbf{1}$ .

Then, since the tangency portfolio is 100% invested in risky assets (as it is on the efficient frontier) but is also on the CML, then we have for this portfolio :  $\mathbf{1}^T w = 1$ . Thus, we can deduce that  $w = \frac{\Sigma^{-1} \tilde{\mu}}{\mathbf{1}^T \Sigma^{-1} \tilde{\mu}}$ .

```
# Expected Returns of Assets
mu = colMeans(month_rets_select)

# Risk-free rate
Rf = mean(rf_rate_select)
```

```

Sigma_inv = solve(Sigma)
mu_tilde = as.matrix(mu - Rf)
array_ones = array(1, dim = c(length(mu), 1))

# Weights of Tangent Portfolio
w_T = Sigma_inv %*% mu_tilde / as.numeric(t(array_ones) %*% Sigma_inv %*% mu_tilde)

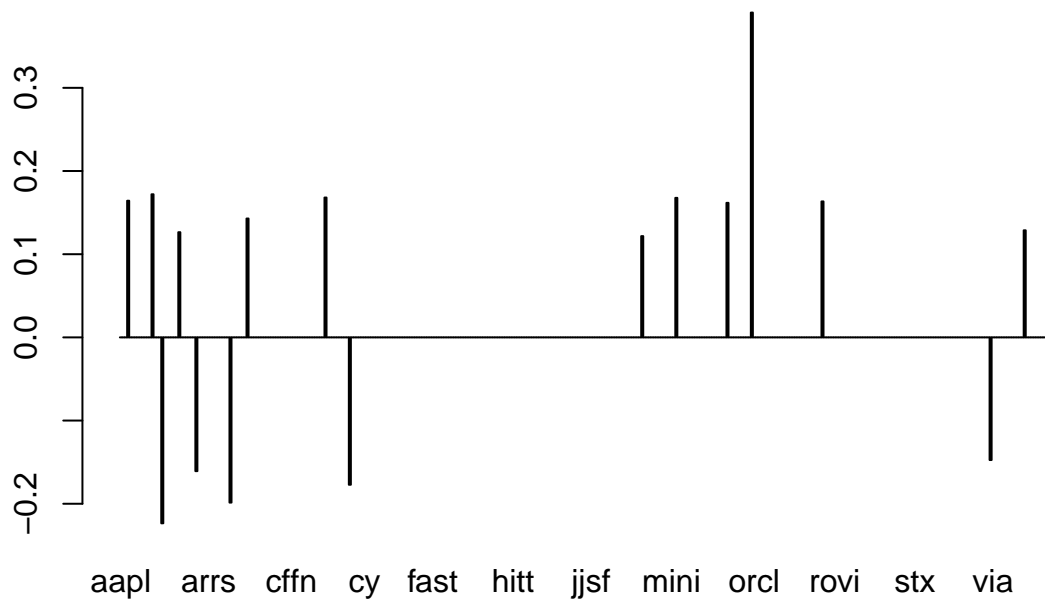
```

Let us keep only the weights that are significant, i.e. let's say above 3% in absolute value.

```

w_T[abs(w_T) < 0.03] = 0
w_T = w_T/sum(w_T)

```



The weights of tangent portfolio found are diversified only accross about 20 assets.

## II.b) Mean-Variance Model with Statistical Factors

We propose to use factors derived from a Principal Components Analysis (PCA) to model the covariance between securities. In practice, we will use the “Diagonizable Model of Covariance” described by Jacobs, Levy & Markowitz (2005).

With data previously selected,

1. Perform a PCA and identify the significant factors.
2. Calculate the factors returns time series  $f(t)$ .
3. Model assets returns by a regression on factors returns and estimate the coefficients.

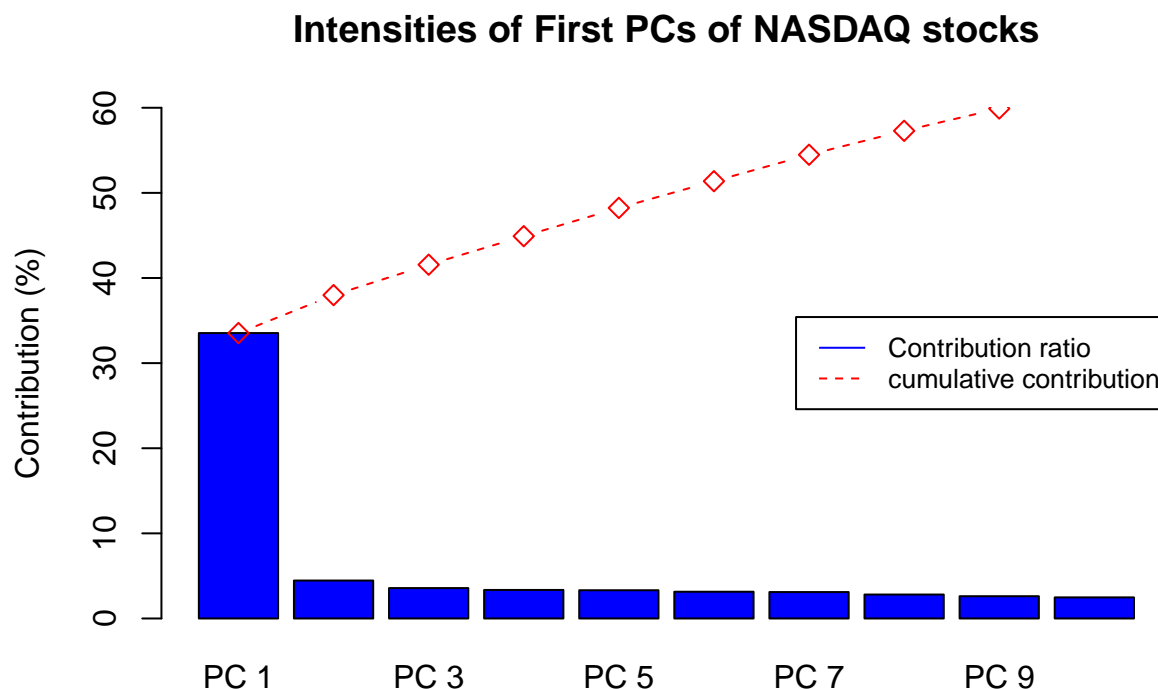
4. Calculate covariance matrices of factors returns and error terms.
5. Formulate and solve the quadratic program whose solution is the tangent portfolio, and compare this solution to the previous one.

### Question 1.

The Principal Components Analysis enables to project data (here our monthly returns selected) in an orthogonal basis, such that the  $i$ -th component is the one that explains the most the variance not explained by the  $i - 1$  previous vectors. It can be shown that the Principal Components can be found using eigenvalue decomposition of returns covariance matrix according to Spectral Theorem :  $\Sigma = QDQ^T$ , with columns of  $Q$  being the eigenvectors of eigenvalues associated, but also the Principal Components (scaled by the number of observations).

```
# Result of PCA (scale=TRUE to scale data for PCA)
res_pca = prcomp(month_rets_select, scale=TRUE)

# Eigenvalues associated with each eigenvector
eigvals = res_pca$sdev^2
# Normalized eigenvalues
eigvals_normed = eigvals/sum(eigvals)
```



Notice that the eigenvalues and eigenvectors associated are already ordered in decreasing order of importance. We'll consider we keep the first 3 components.

## Question 2.

Then, we construct the factors returns time series. A factor return is a linear combination of the assets returns, which is given by the values in the corresponding eigenvector.

```
# Number of factors to keep
nb_factors = 3

# Factors returns series
factors_rets = res_pca$x[,1:nb_factors]
```

## Question 3.

Let us assume the assets return follow the stochastic relation :  $R(t) = \mu + Bf(t) + \varepsilon(t)$ , with  $\varepsilon(t) \sim \mathcal{N}(0, \sigma^2 I)$ ,  $f(t)$  being the factors returns, and  $\mu$  and  $B$  being coefficients to estimate.

Since we have calculated  $f(t)$ , we can estimate by Ordinary Least Squares (OLS) method  $\mu$  and  $B$ , as the noise has homoscedasticity.

```
ols_model = lm(month_rets_select ~ factors_rets)
mu = ols_model$coefficients[1,]
B = t(ols_model$coefficients[2:(nb_factors+1),])
```

## Question 4.

The covariance matrix of returns is then given by  $\Sigma = \mathbb{V} [R(t)R(t)^T] = \mathbb{V} [(\mu + Bf(t) + \varepsilon(t)) (\mu^T + f(t)^T B + \varepsilon(t)^T)] = B\Sigma_f B^T + \Sigma_\varepsilon$ , assuming that  $\varepsilon(t)$  is not correlated with the factors returns, and  $\Sigma_f$  and  $\Sigma_\varepsilon$  being the covariance matrices of  $f$  and  $\varepsilon$ .

```
# Covariance Matrix of Factors Returns
Sigma_f = cov(factors_rets)

# Covariance matrix of noise
Sigma_eps = matrix(0, nrow = N, ncol = N)
diag(Sigma_eps) = diag(cov(ols_model$residuals))

# Final Covariance Matrix
Sigma = B %*% Sigma_f %*% t(B) + Sigma_eps
```

Let's check the matrix is positive definite.

```
res = is.positive.definite(Sigma)
if(res) {
  print("The matrix is positive definite.")
} else {
  print("The matrix is not positive definite.")
}
```

```
## [1] "The matrix is positive definite."
```

### Question 5.

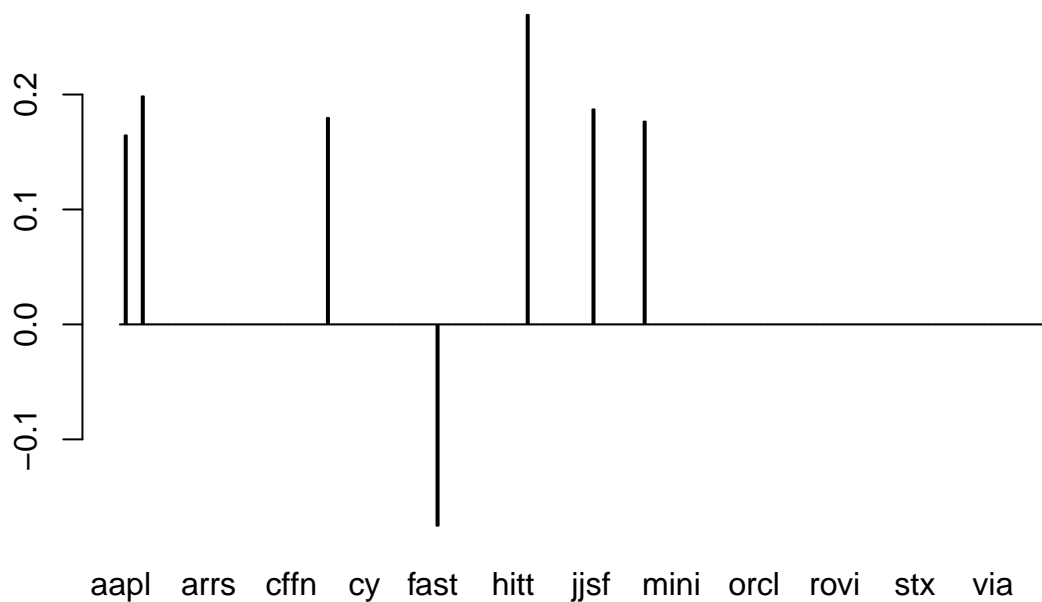
Since we now have all the data, let's find the tangent portfolio, the formula being the same as previously.

```
Sigma_inv = solve(Sigma)

# Weights of Tangent Portfolio
w_T = Sigma_inv %*% mu_tilde / as.numeric(t(array_ones) %*% Sigma_inv %*% mu_tilde)
```

Let us keep only the weights that are significant, i.e. above 3% in absolute value.

```
w_T[abs(w_T) < 0.03] = 0
w_T = w_T/sum(w_T)
```



We can see the weights are scattered among less assets than previously.

## II.c) Mean-Variance Model with Fama-French Factors

Here, we won't use statistical properties to identify factors and estimate their returns. Instead, we'll consider explicit microeconomic factors whose returns are directly observable.

Fama-French have identified 3 factors in their model :

$R(t) = \mu + B_M f_M(t) + B_{SMB} f_{SMB}(t) + B_{HML} f_{HML}(t) + \varepsilon(t)$ , with  $\varepsilon(t) \sim \mathcal{N}(0, \sigma^2 I)$ ,  $f_M(t)$ ,  $f_{SMB}(t)$ ,  $f_{HML}(t)$  being the considered factors returns, and  $\mu$  and  $B_M$ ,  $B_{SMB}$ ,  $B_{HML}$  being coefficients to estimate.

The *SMB* factor (Small Minus Big, Capitalization factor) and *HML* factor (High Minus Low, Valorisation factor) are arbitrage portfolios defined as linear combinations of 4 portfolios which represent a segmentation of assets market capitalization and book-to-market value ratio.

The *M* factor (Market factor) is the factor present in Sharpe's 1 Factor model, i.e. the market excess return wrt risk-free rate.

We proceed in the same way as previously, considering the 3 Fama-French factors instead of the statistical factors.

1. Compare the solution to the previous one.
2. Compare the first PCA factor with the Fama-French market factor.

### Question 1.

First, let's get the fama-french factors returns for the time window selected. Moreover, the fama french returns data is already monthly, dates are already alike "YYYY-MM-01", and there are no missing values.

```
print(paste("Missing values : ", sum(is.na(ts.FF))))
```

```
## [1] "Missing values : 0"
```

```
ff_row_dates <- as.Date(row.names(ts.FF))
ff_factors_rets = ts.FF[ff_row_dates %in% as.Date(dates_window), ]
nb_factors = ncol(ff_factors_rets)
```

Then, let's calculate the covariance matrix, as previously.

```
# Regression to estimate coefficients
ols_model = lm(month_rets_select ~ ff_factors_rets)
mu = ols_model$coefficients[1,]
B = t(ols_model$coefficients[2:(nb_factors+1),])

# Covariance Matrix of Factors Returns
Sigma_f = cov(ff_factors_rets)

# Covariance matrix of noise
Sigma_eps = matrix(0, nrow = N, ncol = N)
diag(Sigma_eps) = diag(cov(ols_model$residuals))

# Final Covariance Matrix
Sigma = B %*% Sigma_f %*% t(B) + Sigma_eps

res = is.positive.definite(Sigma)
if(res) {
  print("The matrix is positive definite.")
} else {
  print("The matrix is not positive definite.")
}
```

```
## [1] "The matrix is positive definite."
```

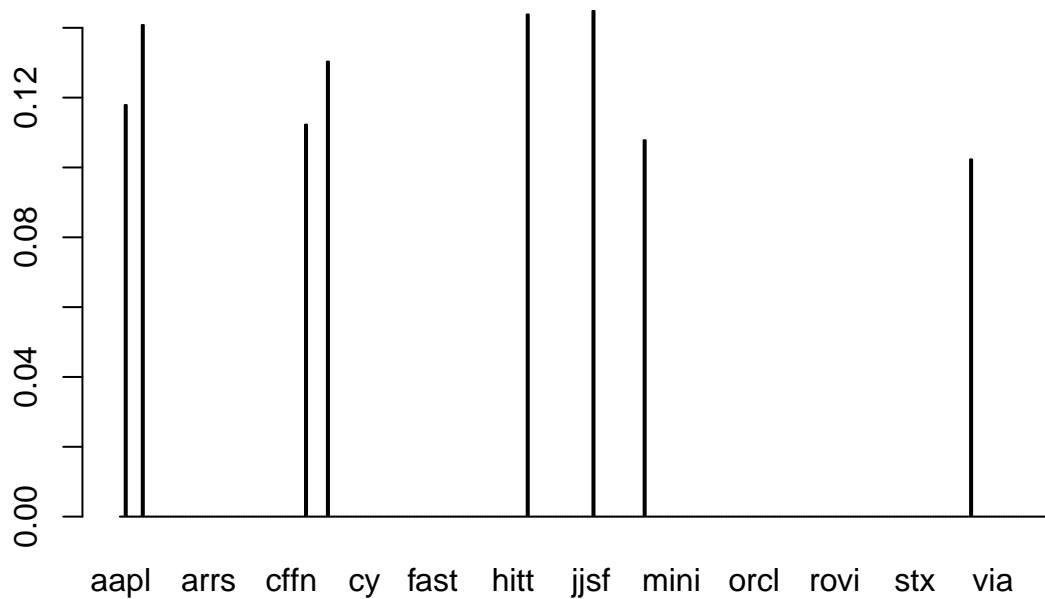
Now, let's find the tangent portfolio.

```
Sigma_inv = solve(Sigma)

# Weights of Tangent Portfolio
w_T = Sigma_inv %*% mu_tilde / as.numeric(t(array_ones) %*% Sigma_inv %*% mu_tilde)
```

Let us keep only the weights that are significant, i.e. let's say above 3% in absolute value.

```
w_T[abs(w_T) < 0.03] = 0
w_T = w_T/sum(w_T)
```



We are more diversified than in the case where factors came from ACP, but still less than when we calculated the historical covariance matrix. Moreover, we can notice there are no short sales of stocks here.

## Question 2.

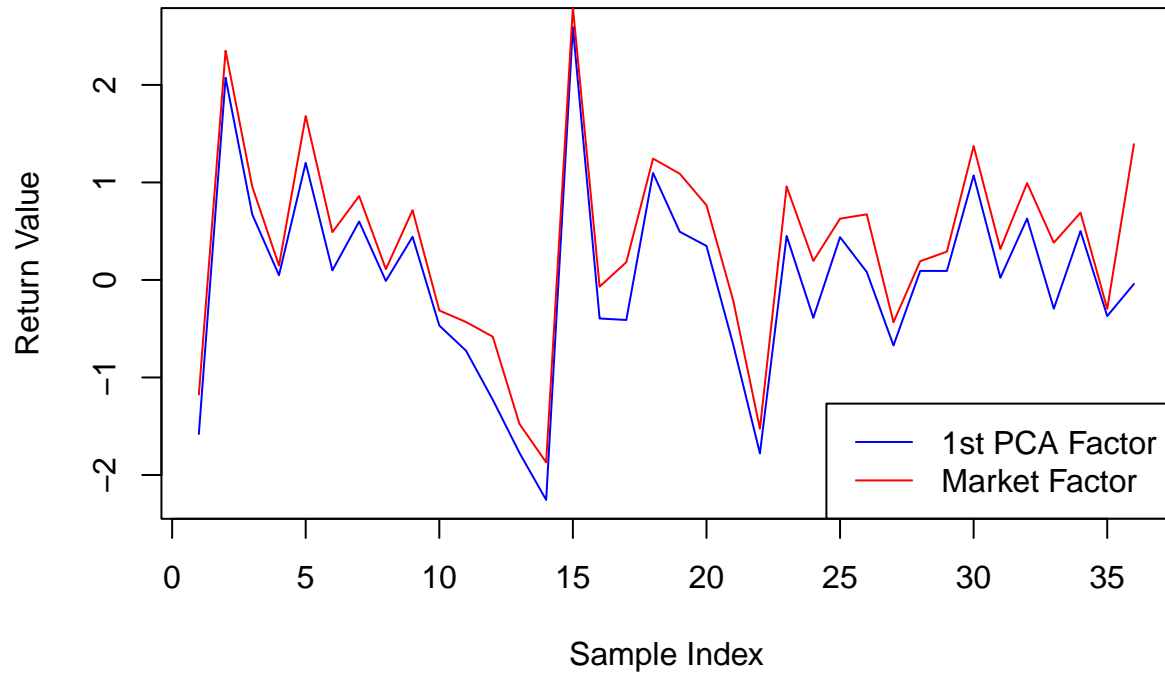
Let us compare the first PCA factor with the Fama-French market factor (normalized so that they are at the same scale).

```
# Market Factor returns normalized
mkt_factor_ret = ff_factors_rets[,1]
mkt_factor_ret_norm = mkt_factor_ret / sd(mkt_factor_ret)

# 1st PCA Factor returns normalized
```

```
pca1_factor_ret = factors_rets[:,1]
pca1_factor_ret_norm = pca1_factor_ret / sd(pca1_factor_ret)
```

### 1st PCA Factor VS FF Market Factor



We can see that the market factor is highly positively correlated with the 1st PCA Factor. It means that the excess return of the market wrt risk-free asset is indeed the most important factor statistically.