# Quantitative Finance

Lab - "Portfolio Management"

Fiona Martin, Romain Pepin, Hedi Sagar, Nathan Sanglier

Version: 12 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))
```

```r
# 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, with $\mu$ the array of expected returns to estimate. Notice that in this part about Black-Litterman Model, when we talk about "returns", we mean excess return w.r.t. risk-free rate.

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 (i.e. expected returns of assets at equilibrium) on the assets are $\Pi = \delta\Sigma w_{eq}$. Notice that at equilibrium, we have $R \sim \mathcal{N}(\Pi, \Sigma)$.

```r
# 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 Weight 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 we 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 $\varepsilon^{(v)} \sim \mathcal{N}(0, \Omega)$) :

```r
# 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 that 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 $\tau$ : $\Omega = Diag\left(P\left(\tau\Sigma\right)P^T\right)$.

```
# 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 is heteroscedastic) 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 : $Var[\hat{\mu}] = M^{-1} = \left(\frac{1}{\tau}\Sigma^{-1} + P^T\Omega^{-1}P\right)^{-1}$.

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_view1 = M_inv %*% (1/tau * Sigma_inv %*% PI  + t(P) %*% Omega_inv %*% Q)
Sigma_bar_view1 = 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_view1, mu_hat_view1)
```

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

Then, we can deduce $w^* = \frac{1}{1+\tau}\left(w_eq + P^T\Lambda\right)$ 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 to which we add a weighted sum of the views portfolios whose weights are given by the array $\Lambda$, and finally we scale the entire expression 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. $P = [P_1\ P_2]$: View Matrix, $\hat{\mu}$: Ex-post Expected Return, $w^*$: Optimal Weight, $\frac{w_{eq}}{1+\tau}$: Scaled Equilibrium Weight

| | $P_1$ | $P_2$ | $\hat{\mu}$ | $w^*$ | $w^* - \frac{w_{eq}}{1+\tau}$ |
|---|---|---|---|---|---|
| Australia | 0 | 0 | 4.4 | 1.5 | 0 |
| Canada | 0 | 100 | 8.7 | 41.9 | 39.8 |
| France | -29.5 | 0 | 9.5 | -3.4 | -8.4 |
| Germany | 100 | 0 | 11.2 | 33.6 | 28.3 |
| Japan | 0 | 0 | 4.6 | 11 | 0 |
| UK | -70.5 | 0 | 7 | -8.2 | -20 |
| USA | 0 | -100 | 7.5 | 18.8 | -39.8 |
| $q$ | 5 | 3 | | | |
| $\omega/\tau$ | 0.021 | 0.017 | | | |
| $\lambda$ | 0.298 | 0.418 | | | |

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 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 higher (represented by $\frac{\omega}{\tau}$, $\omega$ being the diagonal values of $\Omega$). Be careful, in the examples shown in this section, the weights do not sum up to 1 as we did not defined such constraint when calculating $w^*$. Indeed, it is not very important, as the goal of these examples is just to show the impact of investor views on assets allocation.

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.

```
# General functions to compute the estimation of Sigma and mu using BLCOP
solve_with_BLCOP = function(V, S, Pi, tau){
  # Posterior Distribution of returns
```

```r
  post_distribution = posteriorEst(views=V, sigma=S, mu=as.numeric(Pi), tau=tau)
  Sigma_bar_bl_view = post_distribution@posteriorCovar
  mu_hat_bl_view = post_distribution@posteriorMean
  return(list(Sigma_bar_bl_view = Sigma_bar_bl_view, mu_hat_bl_view = mu_hat_bl_view))

}
res = solve_with_BLCOP(views, Sigma, PI, tau)
Sigma_bar_bl_view1 = res$Sigma_bar_bl_view
mu_hat_bl_view1 = res$mu_hat_bl_view
```

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

```r
# Optimal Weights
w_star = 1/delta * solve(Sigma_bar_bl_view1, mu_hat_bl_view1)
```

Let's verify the results :

Table 3: BLCOP Solution with View 1. $P = [P_1\ P_2]$: View Matrix, $\hat{\mu}$: Ex-post Expected Return, $w^*$: Optimal Weight, $\frac{w_{eq}}{1+\tau}$: Scaled Equilibrium Weight

| | $P_1$ | $P_2$ | $\hat{\mu}$ | $w^*$ | $w^* - \frac{w_{eq}}{1+\tau}$ |
|---|---|---|---|---|---|
| Australia | 0 | 0 | 4.4 | 1.5 | 0 |
| Canada | 0 | 100 | 8.7 | 41.9 | 39.8 |
| France | -29.5 | 0 | 9.5 | -3.4 | -8.4 |
| Germany | 100 | 0 | 11.2 | 33.6 | 28.3 |
| Japan | 0 | 0 | 4.6 | 11 | 0 |
| UK | -70.5 | 0 | 7 | -8.2 | -20 |
| USA | 0 | -100 | 7.5 | 18.8 | -39.8 |
| $q$ | 5 | 3 | | | |
| $\omega/\tau$ | 0.021 | 0.017 | | | |
| $\lambda$ | 0.298 | 0.418 | | | |

We obtain the same results as when we do not use the BLCOP package.

**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).

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

mu_hat_view2 = M_inv %*% (1/tau * Sigma_inv %*% PI  + t(P) %*% Omega_inv %*% Q)
Sigma_bar_view2 = Sigma + M_inv

w_star = 1/delta * solve(Sigma_bar_view2, mu_hat_view2)
```

```
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. $P = [P_1\ P_2]$: View Matrix, $\hat{\mu}$: Ex-post Expected Return, $w^*$: Optimal Weight, $\frac{w_{eq}}{1+\tau}$: Scaled Equilibrium Weight

| | $P_1$ | $P_2$ | $\hat{\mu}$ | $w^*$ | $w^* - \frac{w_{eq}}{1+\tau}$ |
|---|---|---|---|---|---|
| Australia | 0 | 0 | 4.4 | 1.5 | 0 |
| Canada | 0 | 100 | 9.1 | 53.3 | 51.3 |
| France | -29.5 | 0 | 9.5 | -3.3 | -8.2 |
| Germany | 100 | 0 | 11.3 | 33.1 | 27.8 |
| Japan | 0 | 0 | 4.6 | 11 | 0 |
| UK | -70.5 | 0 | 7 | -7.8 | -19.6 |
| USA | 0 | -100 | 7.3 | 7.3 | -51.3 |
| $q$ | 5 | 4 | | | |
| $\omega/\tau$ | 0.021 | 0.017 | | | |
| $\lambda$ | 0.292 | 0.538 | | | |

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 on 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)

res = solve_with_BLCOP(views, Sigma, PI, tau)
Sigma_bar_bl_view2 = res$Sigma_bar_bl_view
mu_hat_bl_view2 = res$mu_hat_bl_view

w_star = 1/delta * solve(Sigma_bar_bl_view2, mu_hat_bl_view2)
```

Table 5: BLCOP Solution with View 2. $P = [P_1\ P_2]$: view matrix, $\hat{\mu}$: ex-post expected return, $w^*$: Optimal Weight, $\frac{w_{eq}}{1+\tau}$: Scaled Equilibrium Weight

| | $P_1$ | $P_2$ | $\hat{\mu}$ | $w^*$ | $w^* - \frac{w_{eq}}{1+\tau}$ |
|---|---|---|---|---|---|
| Australia | 0 | 0 | 4.4 | 1.5 | 0 |
| Canada | 0 | 100 | 9.1 | 53.3 | 51.3 |
| France | -29.5 | 0 | 9.5 | -3.3 | -8.2 |
| Germany | 100 | 0 | 11.3 | 33.1 | 27.8 |
| Japan | 0 | 0 | 4.6 | 11 | 0 |
| UK | -70.5 | 0 | 7 | -7.8 | -19.6 |
| USA | 0 | -100 | 7.3 | 7.3 | -51.3 |
| $q$ | 5 | 4 | | | |
| $\omega/\tau$ | 0.021 | 0.017 | | | |
| $\lambda$ | 0.292 | 0.538 | | | |

We obtain the same results as when we do not use the BLCOP package.

## Question 3.

To further illustrate the difference between prior and ex-post returns distribution, we will compare the tangent portfolio allocation when using prior returns distribution ($R \sim \mathcal{N}(\Pi, \Sigma)$) and ex-post returns distribution ($R \sim \mathcal{N}(\hat{\mu}, \bar{\Sigma})$, for the 2 examples presented before).

In the Mean-Variance framework, the tangent portfolio is the one that maximizes the Sharpe Ratio along the Efficient Frontier defined by the problem constraints. Here, we'll define the following constraints : $w^T\mathbf{1} = 1$ and $\forall i \in 1:N$, $w_i \geq 0$ (we do not allow short sales). Thus, we have : $\max SR(w) = \frac{w^T\mu - R_f}{\sqrt{(w^T\Sigma w)}}$, s.t. $\mathbf{1}^T w = 1$ and $\forall i \in 1:N$, $w_i \geq 0$, where $\mu$ is this time the true expected return of the asset, and not the asset expected excess return w.r.t. risk-free rate; $\Sigma$ is the covariance matrix considered.

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 solve $\min \frac{1}{2}w^T\Sigma w$, with the constraints $w^T(\mu - R_f\mathbf{1}) = \tilde{\mu}_p$ and $\forall i \in 1:N$, $w_i \geq 0$ (quadratic problem). Notice that we can take any value for $\tilde{\mu}_p$ here, so we'll take arbitrarily $\tilde{\mu}_p = 5\%$.

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 : $w^T\mathbf{1} = 1$. Thus, to find the tangency portfolio, we just need to normalize the optimal weights found by $w^T\mathbf{1} = 1$.

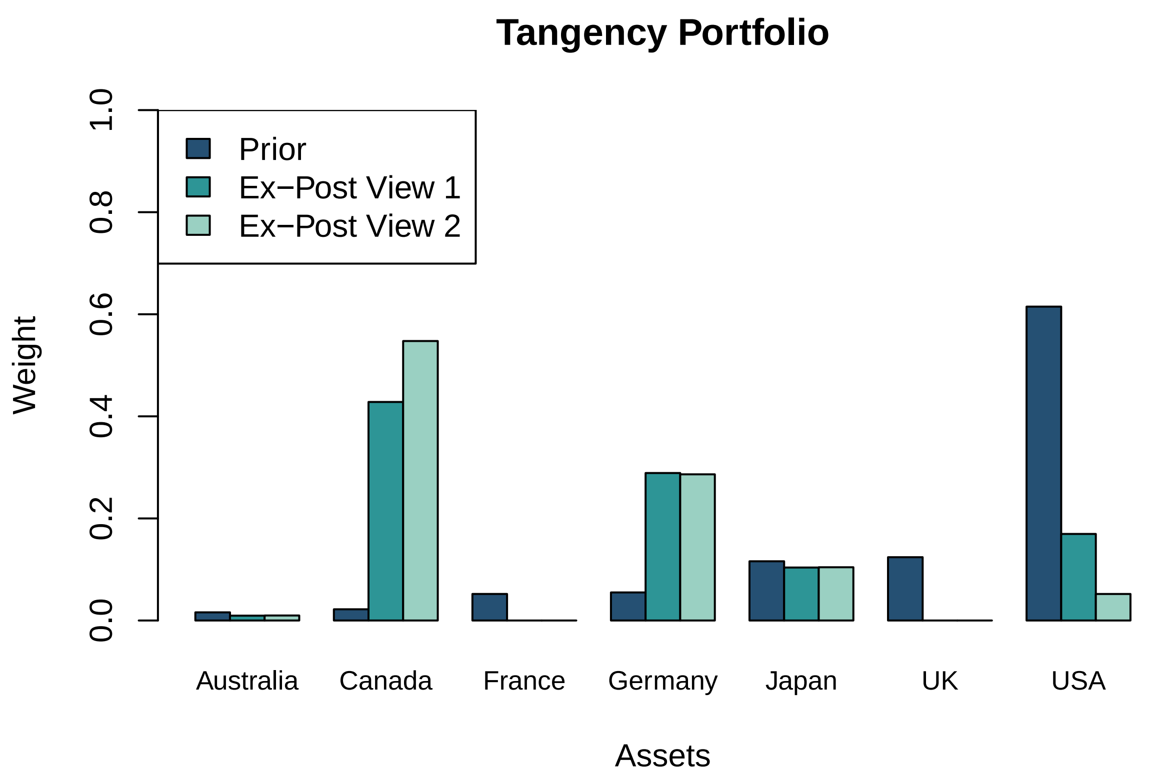
We will use the "quadprog" library that enables us to solve quadratic optimization problems (which is the case here). The function "solve.QP" solves the following quadratic problem : $\min -d^Tb + \frac{1}{2}b^TDb$ s.t. $A^Tb \geq b_0$. So in our case, $d = 0_{N,1}$, $b = w$, $D = \Sigma$, $A = \begin{bmatrix} \mu - R_f\mathbf{1} & I_N \end{bmatrix}$ and $b_0 = \begin{bmatrix} \tilde{\mu}_p \\ 0_{N,1} \end{bmatrix}$. Also notice that we'll set $m_{eq} = 1$, i.e. we only impose the first constraint as equality, the other ones being "greater than" inequalities.

Be careful, as we consider in this section expected returns as excess returns w.r.t. risk-free rate, there is no need here to define a risk-free rate because the first constraint will be written as $w^T((\mu + R_f\mathbf{1}) - R_f\mathbf{1}) = w^T\mu = \tilde{\mu}_p$., where $\mu$ is either $\Pi$ or $\hat{\mu}$.

```
# Array of ones
array_ones = array(1, dim = c(N, 1))

# Excess Return to reach (dummy variable)
mu_tilde = 0.05

d = rep(0, N)
b0 = c(mu_tilde, rep(0, N))

# General function to get w_star
get_w_star = function(S, m){
  D = S
  A = cbind(m, diag(N))
  w_star = solve.QP(D, d, A, b0, meq=1)$solution
  return(w_star / as.numeric(t(w_star) %*% array_ones))
}

# MV Optim Prior Distribution
w_star_prior = get_w_star(Sigma, PI)
```

```
# MV Optim Ex-post Distribution View 1
w_star_post1 = get_w_star(Sigma_bar_view1, mu_hat_view1)

# MV Optim Ex-post Distribution View 2
w_star_post2 = get_w_star(Sigma_bar_view2, mu_hat_view2)
```

Let us display the results.




Recall that in the first view, Germany outperforms remaining Europe by 5%, and Canada outperforms US by 3%. Thus, the graph observed is logic as in the first view, we put much more weight on Canada and much less on US, a little more weight on Germany and less weight on the remaining Europe. Moreover, recall that the second view is the same as the first view, but with Canada outperforming US by 4%, instead of 3%. Indeed, we can see that the weight on Canada is even higher and the weight on US even lower compared to the first view.

# 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)

# Retrieve Fama-French 3 factors monthly returns since 1960
ts.FF <- FFdownload$`x_F-F_Research_Data_Factors`
ts.FF <- ts.FF$monthly$Temp2["1960-01-01/", c("Mkt.RF","SMB","HML")]/100
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 (here, we'll remove an asset if its volatility over the considered period is above the average volatility of the stocks during this period plus 3 standard deviations of the stocks volatility during this period), 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.

```r
# 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 "apply.monthly" function of PerformanceAnalytics" library.

```r
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 in the format "YYYY-MM-01". Thus we need to change the dates of our monthly returns to match the way the ones of the risk-free rates are displayed, and select the 36 months corresponding for both datasets (notice there are no missing values in riskfree rate dataset, nor in monthly returns dataset by construction).

```r
# Window of monthly returns selected
obs.months = 36 # Number of observation months

all_dates = floor_date(ymd(time(month_rets)), 'month')
time(month_rets) = as.Date(all_dates)

dates_start = dmy("01Aug2010") # Here we'll study the last 36 months

idx_start = closest.index(month_rets, dates_start)
idx_end = idx_start + obs.months - 1
month_rets_select = month_rets[idx_start:idx_end,]

# Window of riskfree rates selected
```

```r
rf_rate_select = rf_rate[row.names(rf_rate) %in% row.names(month_rets_select), ]

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

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

```r
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.

```r
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.

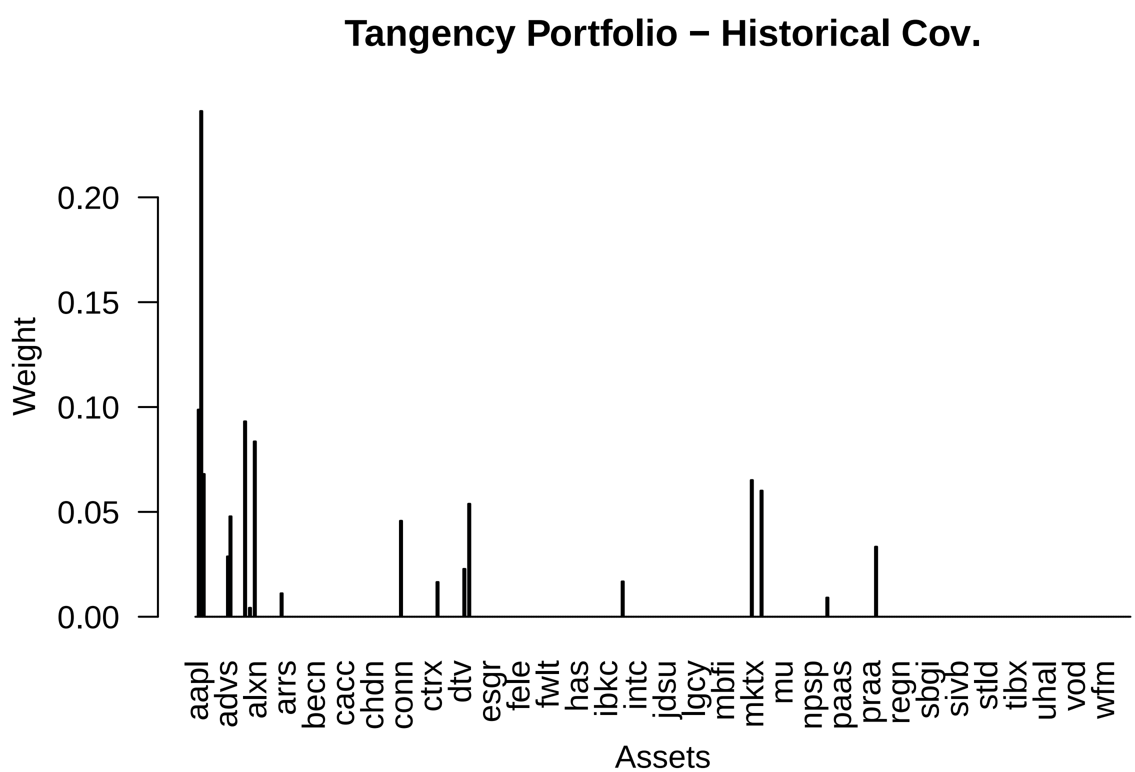
**Question 3.**

We will find the tangent portfolio as in Question 3. of first section, and we also do not allow short sales. Thus, using "quadprog" library, we have (see Question 3. of first section for details): $d = 0_{N,1}$, $b = w$, $D = \Sigma$, $A = \begin{bmatrix} \mu - R_f \mathbf{1} & I_N \end{bmatrix}$, $b_0 = \begin{bmatrix} \tilde{\mu}_p \\ 0_{N,1} \end{bmatrix}$ and $m_{eq} = 1$.

Moreover we choose $\mu$ as the simple averages of monthly returns for each asset and $R_f$ as the simple average of monthly returns for riskless asset on the period considered.

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

# Risk-free rate
Rf = mean(rf_rate_select)

# Array of ones
array_ones = array(1, dim = c(N, 1))

# Excess Return to reach
```

```r
mu_tilde = 0.05

d = rep(0, N)
b0 = c(mu_tilde, rep(0, N))
A = cbind(mu - Rf, diag(N))
D = Sigma

w_star = solve.QP(D, d, A, b0, meq=1)$solution
# Normalizing weights
w_star = w_star / as.numeric(t(w_star) %*% array_ones)
```




The weights of tangent portfolio found are diversified only accross a few 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,

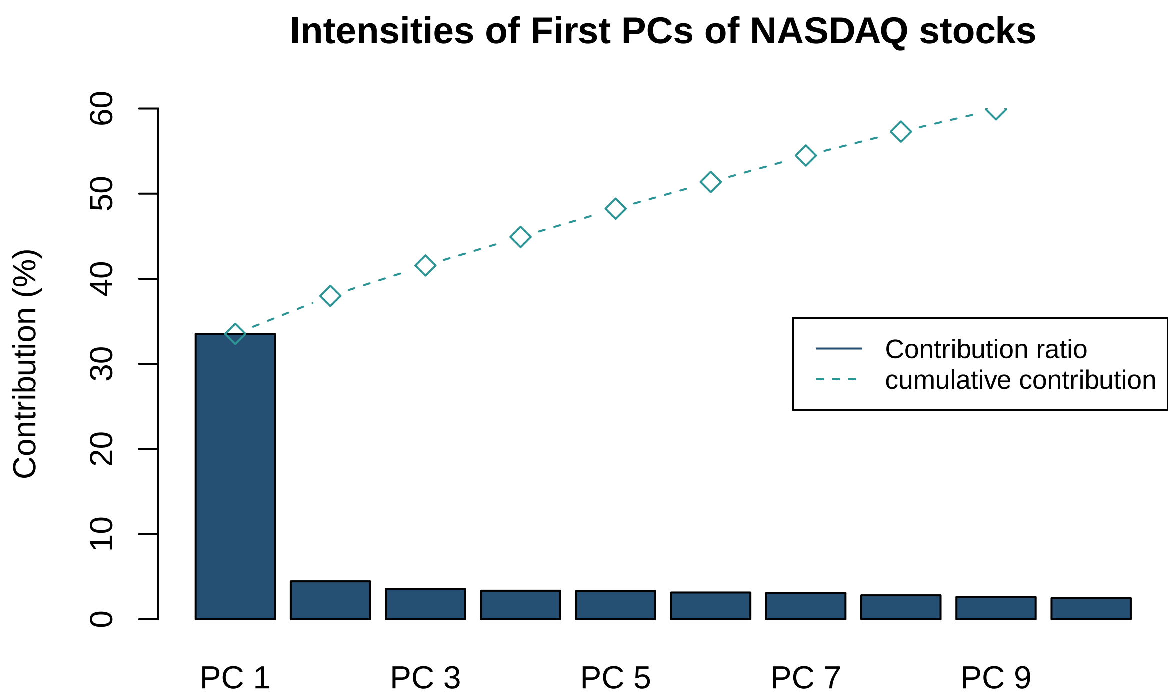
1. Perform a PCA and identify the significant factors.
2. Calculate the factors returns time series $f(t)$.
3. Model assets returns by a regression on factors returns and estimate the coefficients.
4. Calculate covariance matrices of factors returns and error terms.

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 the 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 4 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 = 4

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

**Question 3.**

Let us assume the assets returns 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. Also notice that $\mu$ represents the assets expected returns.

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

```
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

$$\Sigma = \mathbb{V}\left[R(t)R(t)^T\right] = \mathbb{V}\left[(\mu + Bf(t) + \varepsilon(t))\left(\mu^T + f(t)^T B + \varepsilon(t)^T\right)\right] = B\Sigma_f B^T + \Sigma_\varepsilon$$

, assuming that $\varepsilon(t)$ is not correlated with the factors returns, and $\Sigma_f$ and $\Sigma_\varepsilon$ being the covariance matrices of $f$ and $\varepsilon$. Notice that as Principal Components Analysis enables to project data in an orthogonal basis, then $\Sigma_f$ is diagonal.

```
# Covariance Matrix of Factors Returns
Sigma_f = matrix(0, nrow = nb_factors, ncol = nb_factors)
diag(Sigma_f) = diag(cov(factors_rets))

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

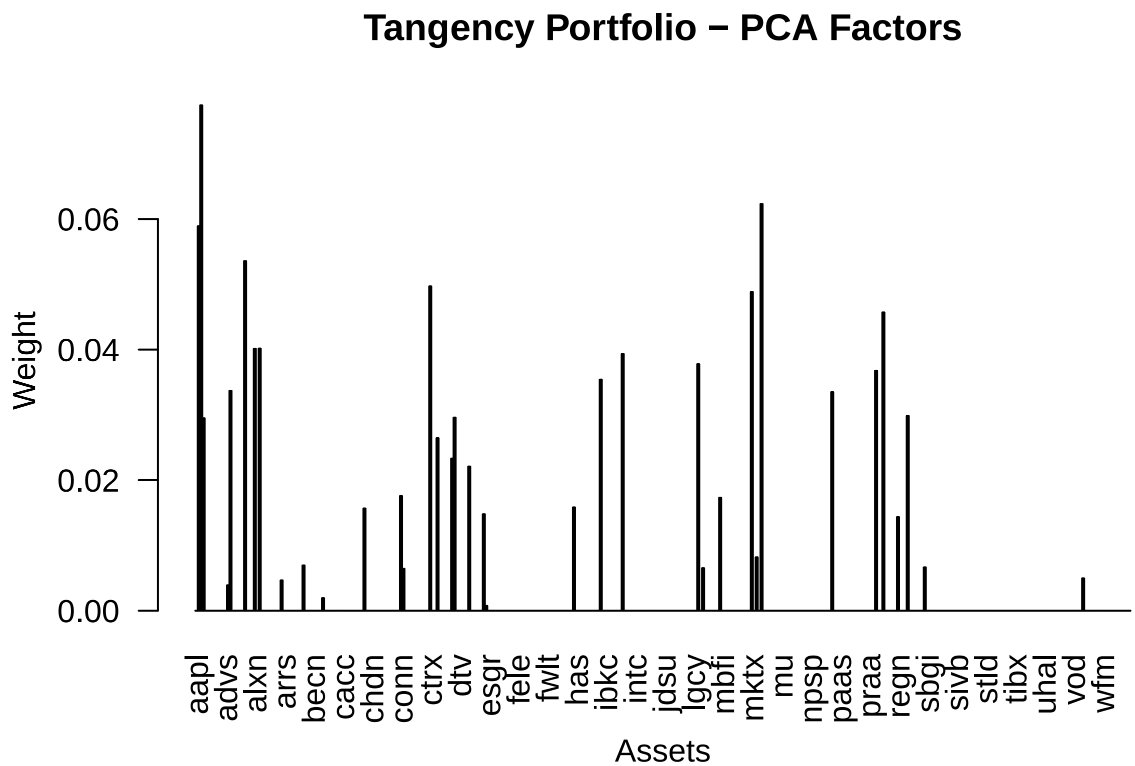
**Question 5.**

Now, as we have more securities than observations, it could be interesting to use a "fast algorithm" to estimate the tangency portfolio, instead of defining the classical mean variance problem with the estimated $\Sigma$. Indeed, Jacobs et al. have developed such algorithm for factors model of covariance, referred to as the "Diagonizable Model of Covariance". In our case, we use the version where short sales are not allowed.

First, recall our optimization problem such that (using notations of "quadprog" library, see previous section) : $d = 0_{N,1}$, $b = w$, $D = \Sigma$, $A = \begin{bmatrix}\mu - R_f\mathbf{1} & I_N\end{bmatrix}$ and $b_0 = \begin{bmatrix}\tilde{\mu}_p \\ 0_{N,1}\end{bmatrix}$.

Then, Jacobs et al. proved that it is computationally faster to add the following. Define K "fictitious" investments (weights) $v$ such that $v = B^T w$ and denote $u = \begin{bmatrix} w \\ v \end{bmatrix}$ the new weights vector. Then, let's add this last constraint to our constraints (see previous section on tangency portfolio calculation) : $A^{(new)} = \begin{bmatrix} B & A \\ -I_K & 0_{K,N+1} \end{bmatrix}$, $b_0^{(new)} = \begin{bmatrix} 0_{K,1} \\ b_0 \end{bmatrix}$ and $b^{(new)} = u$. Notice we need to put the new constraints in the first columns of $A^{(new)}$ as they are equality constraints. Thus, the first $K + 1$ columns of $A^{(new)}$ will be equality constraints. Now, the portfolio variance is $\sigma_p^2 = w^T B \Sigma_f B^T w + w^T \Sigma_\varepsilon w = v^T \Sigma_f v + w^T \Sigma_\varepsilon w = u^T M u$, with $M = \begin{bmatrix} \Sigma_\varepsilon & 0_{K,N} \\ 0_{N,K} & \Sigma_f \end{bmatrix}$. Thus, the original problem can be restated as the mean-variance optimization problem under the constraints defined above with $D^{(new)} = M$ and $d^{(new)} = 0_{N+K,1}$. Then, to find $w$, we just need to retrieve the $N$ first coefficients of $b^{(new)} = u$ and normalize them.

```r
# Function to calculate u_star
get_u_star = function(B, mu, Rf, N, nb_factors, b0, Sigma_f, Sigma_eps){
  A = cbind(mu - Rf, diag(N))

  d_new = rep(0, N+nb_factors)
  b0_new = c(rep(0, nb_factors), b0)
  A_1 = rbind(B, -diag(nb_factors))
  A_2 = rbind(A, matrix(0, nrow = nb_factors, ncol = ncol(A)))
  A_new = cbind(A_1, A_2)
  D_1 = rbind(Sigma_eps, matrix(0, nrow = nb_factors, ncol = N))
  D_2 = rbind(matrix(0, nrow = N, ncol = nb_factors), Sigma_f)
  D_new = cbind(D_1, D_2)

  u_star = solve.QP(D_new, d_new, A_new, b0_new, meq=nb_factors+1)$solution
  return(u_star)
}

u_star = get_u_star(B, mu, Rf, N, nb_factors, b0, Sigma_f, Sigma_eps)

# Retrieving assets weights
w_star = u_star[1:N]
# Normalizing weights
w_star = w_star / as.numeric(t(w_star) %*% array_ones)
```




We can see the weights are scattered among more assets than previously, i.e. the solution is more robust.

### 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 in the format "YYYY-MM-01", and there are no missing values.

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

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

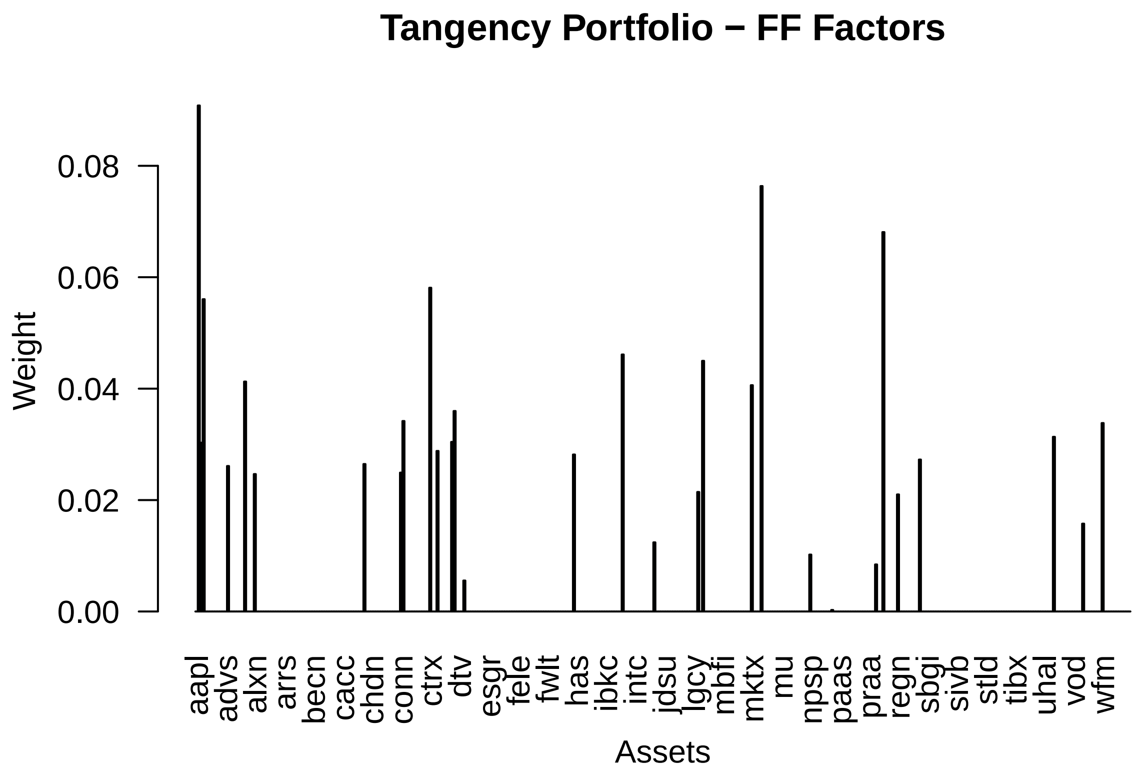
```
ff_factors_rets = ts.FF[row.names(ts.FF) %in% row.names(month_rets_select), ]
nb_factors = ncol(ff_factors_rets)
```

Then, let's calculate the covariance matrix of factors returns and error terms, as previously. Notice that here, $\Sigma_f$ is not diagonal anymore.

```
# 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))
```

Now, let's find the tangent portfolio, using the "Diagonizable Model of Covariance" as previously.

```
# Using the same function as above
u_star = get_u_star(B, mu, Rf, N, nb_factors, b0, Sigma_f, Sigma_eps)

# Retrieving assets weights
w_star = u_star[1:N]
# Normalizing weights
w_star = w_star / as.numeric(t(w_star) %*% array_ones)
```

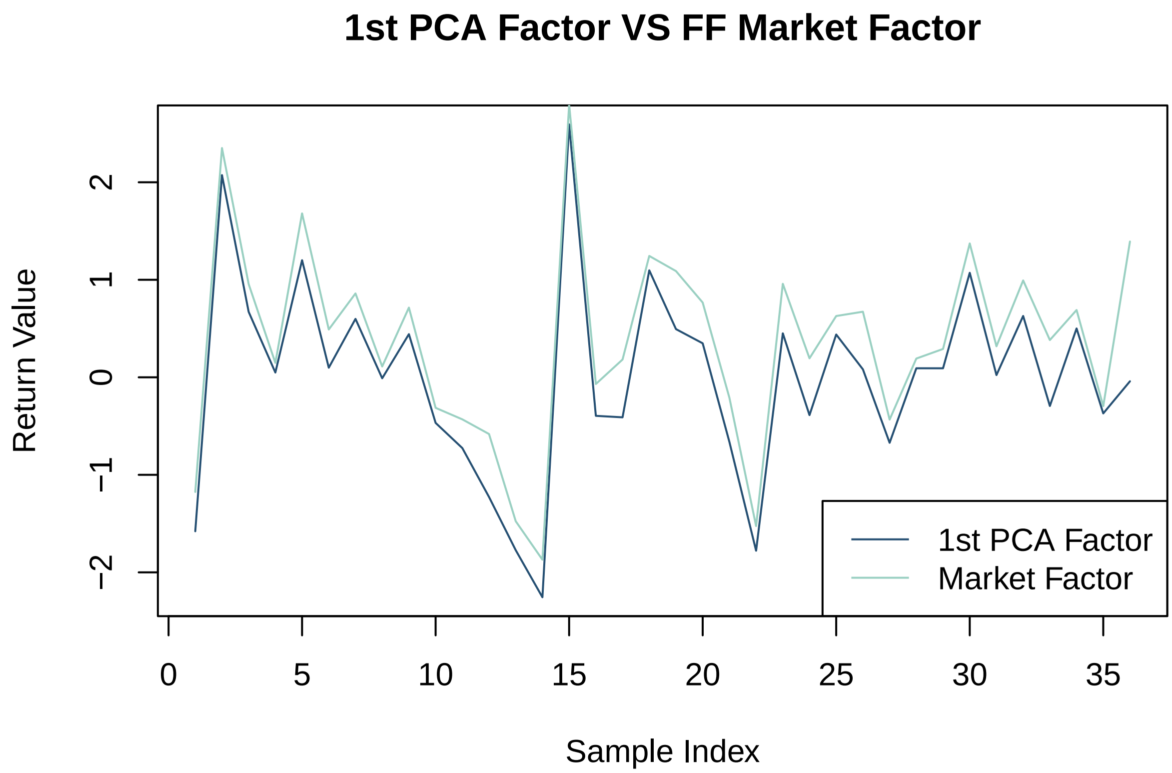



As for the situation where factors were estimated statistically, we can see that the weights are scattered among more assets than when using the historical covariance matrix, i.e. the solution is more robust. Moreover, diversification is of same order as for statistical factors.

**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)
```




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 w.r.t. risk-free asset is indeed the most important factor statistically.