CQF Exam One

Boyan Davidov

March 12, 2020

Optimal Portfolio Allocation

Question 1

- We start by formulating the Lagrange function for the minimum variance optimization problem using two lagrange multipliers λ and γ:

$L(w,\, \lambda,\gamma)= \frac{1}{2}w^T\Sigma w + \lambda(m - \mu^T w) + \gamma(1 - \boldsymbol{1^T} w)$
The partial derivatives are as follows:

$\frac{\partial L}{\partial w}(w, \lambda, \gamma) = w^T\Sigma - \lambda\mu^T - \gamma\boldsymbol{1^T}$

$\frac{\partial L}{\partial \lambda}(w, \lambda, \gamma) = m - \mu^T w$

$\frac{\partial L}{\partial \gamma}(w, \lambda, \gamma) = 1 - \boldsymbol{1^T} w$

Further we make use of the formulae derived from Portfolio Optmization Lecture. The optimal allocations vector w^*(after solving for the first order conditions) is:

$w^* = \sum^{-1}(\lambda\mu + \gamma\mathbf{1})$

where

$$\begin{cases} \lambda = \frac{Am-B}{AC-B^2} \\ \gamma = \frac{C-Bm}{AC-B^2} \end{cases}$$

and

$$\begin{cases} A = & \boldsymbol{1^T}\sum\mathbf{1} \\ B = & \mu^T\sum\mathbf{1} \\ C = & \mu^T\sum^{-1}\mu \end{cases}$$

To compute the optimal allocation when m=0.045 for the initial correlation matrix and for the stressed ones, we make use of R. Let's call mu.vec the vector of mean returns, stdev.mat - a diagonal matrix of respective assets' standard deviations, ro.mat - correlation matrix.

```
>   asset.names <- c("A", "B", "C", "D")
>   mu.vec = c(0.02, 0.07, 0.15, 0.2)
>   names(mu.vec) = asset.names
>   ro.mat = matrix(c(1, 0.3, 0.3, 0.3,
+                     0.3, 1, 0.6, 0.6,
+                     0.3, 0.6, 1, 0.6,
+                     0.3, 0.6, 0.6, 1),
+                   nrow=4, ncol=4)
>   stdev.mat = matrix(c(0.05, 0, 0, 0,
+                        0, 0.12, 0, 0,
+                        0, 0, 0.17, 0,
+                        0, 0, 0, 0.25),
+                      nrow=4, ncol=4)
>   sigma.mat <- stdev.mat%*%ro.mat%*%stdev.mat
>   dimnames(sigma.mat) = list(asset.names, asset.names)
>   ones_vector<-rep(1,4)
>   A<-as.numeric(t(ones_vector)%*%(solve(sigma.mat))%*%ones_vector)
>   B<-as.numeric(t(mu.vec)%*%(solve(sigma.mat))%*%ones_vector)
>   C<-as.numeric(t(mu.vec)%*%(solve(sigma.mat))%*%mu.vec)
>   m<-0.045
>   lambda<-(A*m-B)/(A*C-B^2)
>   gamma<-(C-B*m)/(A*C-B^2)
```

```
>   w_optim<-solve(sigma.mat)%*%(lambda*mu.vec+gamma*ones_vector)
>   w_optim
```

The optimal weights are:

```
        [,1]
A 0.78511066
B 0.05386419
C 0.13355472
D 0.02747042
```

```
>   portfolio_risk<-sqrt(as.numeric(t(w_optim)%*%sigma.mat%*%w_optim))
>   portfolio_risk
```

```
[1] 0.05840091
```

Next, we stress the correlation matrices. We multiply the whole correlation matrix by 1.25 and then subtract from the diagonal elements 0.25 to ensure that the correlation of an asset with itself is 1.

```
> #########
> #stressed*1.25
> #########
>
> ro.mat_stressed1<-ro.mat*1.25-0.25*diag(diag(ro.mat))
> ro.mat_stressed1
```

```
      [,1]  [,2]  [,3]  [,4]
[1,] 1.000 0.375 0.375 0.375
[2,] 0.375 1.000 0.750 0.750
[3,] 0.375 0.750 1.000 0.750
[4,] 0.375 0.750 0.750 1.000
```

```
> sigma.mat_stressed1 <- stdev.mat%*%ro.mat_stressed1%*%stdev.mat
> A<-as.numeric(t(ones_vector)%*%(solve(sigma.mat_stressed1))%*%ones_vector)
> B<-as.numeric(t(mu.vec)%*%(solve(sigma.mat_stressed1))%*%ones_vector)
> C<-as.numeric(t(mu.vec)%*%(solve(sigma.mat_stressed1))%*%mu.vec)
> lambda<-(A*m-B)/(A*C-B^2)
> gamma<-(C-B*m)/(A*C-B^2)
> w_optim_stressed1<-solve(sigma.mat_stressed1)%*%(lambda*mu.vec+gamma*ones_vector)
> w_optim_stressed1
```

```
             [,1]
[1,]  0.818189444
[2,] -0.009403019
[3,]  0.178965849
[4,]  0.012247725
```

```
> portfolio_risk_stressed1<-sqrt(as.numeric(t(w_optim_stressed1)%*%sigma.mat_stressed1%*%w_optim_str
> portfolio_risk_stressed1
```

```
[1] 0.0607102
```

Similarly, we stress the correlation matrix by multiplying by 1.5.

```
> #########
> #stressed*1.5
> #########
>
> ro.mat_stressed2<-ro.mat*1.5-0.5*diag(diag(ro.mat))
> ro.mat_stressed2
```

```
     [,1] [,2] [,3] [,4]
[1,] 1.00 0.45 0.45 0.45
[2,] 0.45 1.00 0.90 0.90
[3,] 0.45 0.90 1.00 0.90
[4,] 0.45 0.90 0.90 1.00
```

```
> sigma.mat_stressed2 <- stdev.mat%*%ro.mat_stressed2%*%stdev.mat
> A<-as.numeric(t(ones_vector)%*%(solve(sigma.mat_stressed2))%*%ones_vector)
> B<-as.numeric(t(mu.vec)%*%(solve(sigma.mat_stressed2))%*%ones_vector)
> C<-as.numeric(t(mu.vec)%*%(solve(sigma.mat_stressed2))%*%mu.vec)
> lambda<-(A*m-B)/(A*C-B^2)
> gamma<-(C-B*m)/(A*C-B^2)
> w_optim_stressed2<-solve(sigma.mat_stressed2)%*%(lambda*mu.vec+gamma*ones_vector)
> w_optim_stressed2
```

```
           [,1]
[1,]  0.8761765
[2,] -0.1461295
[3,]  0.3257015
[4,] -0.0557484
```

```
> portfolio_risk_stressed2<-sqrt(as.numeric(t(w_optim_stressed2)%*%sigma.mat_stressed2%*%w_optim_str
> portfolio_risk_stressed2
```

```
[1] 0.06109091
```

I have written a function that performs this optimization. It takes as input mean return vector, the desired return, vector of assets standard deviations, correlation matrix, minimum and maximum portfolio weights allowed for a position in an asset, and finally a boolean kind of variable to specify if portfolio weight should be in the constrained minimum and maximum weight or between 0 an 1.

```
> # a vector named mu containing the expected returns of the assets must be defined globally
> # a vector of standard deviations and correlation matrix must be defined
>
> # mu0 is the demanded expected return
> # fixed is a vector that contains either -1, 0, 1 for each asset
> # fixed[i] == -1 :  variable i is still free, i.e., betreen 0 and maxw
> # fixed[i] ==  0 :  variable i fixed at 0
> # fixed[i] ==  1 :  variable i between minw and maxw
>
>
> library(quadprog) #to use solve.QP
>
> opt<-function(mu0,mu,stdev.vector, ro.mat,minw,maxw,fixed){
+   CovMatrix<-diag(stdev.vector)%*%ro.mat%*%diag(stdev.vector)
+   minVarPortfolio<-function(mu0,fixed){
+     Constr <- makeConstr(fixed)
+     dvec <- rep(0,times=nrow(CovMatrix))
+     # Constraints matrix (add the constraint mu=mu0)
+     A <- rbind(matrix(mu,nrow=1),Constr$Constr)
+     bvec <- rbind(mu0,Constr$bvec)
+     xx <- solve.QP(2*CovMatrix,dvec,t(A),meq=2+sum(fixed==0),bvec=bvec)
+     return(xx)
+   }
+
+   makeConstr <- function(fixed){
+     # this is the portfolio constraint
+     Constr <- matrix(1,ncol=length(mu),nrow=1)
+     bvec <- matrix(1,nrow=1)
+     # first those that are fixed to w=0, i.e., fixed == 0;
```

```
+     sel <- (1:length(mu))[fixed==0]
+     for (i in sel){
+       Constr <- rbind(Constr,0)
+       Constr[nrow(Constr),i] <- 1
+       bvec <- rbind(bvec,0)
+     }
+     # then those with weights between minx and maxw, fixed = 1;
+     sel <-  (1:length(fixed))[fixed==1]
+     for (i in sel){
+       # the maximum constraint
+       Constr <- rbind(Constr,0)
+       Constr[nrow(Constr),i] <- -1
+       bvec <- rbind(bvec,-maxw)
+       # the minimum constraint
+       Constr <- rbind(Constr,0)
+       Constr[nrow(Constr),i] <- 1
+       bvec <- rbind(bvec,minw)
+     }
+     # undetermined = free, fixed = -1;
+     sel = (1:length(fixed))[fixed==-1]
+     for (i in sel){
+       # the maximum constraint
+       Constr <- rbind(Constr,0)
+       Constr[nrow(Constr),i] <- -1
+       bvec <- rbind(bvec,-maxw)
+       # the minimum constraint
+       Constr <- rbind(Constr,0)
+       Constr[nrow(Constr),i] <- 1
+       bvec <- rbind(bvec,0)
+     }
+     return(list(Constr=Constr,bvec=bvec))
+   }
+   sol <- try(minVarPortfolio(mu0,fixed),silent=TRUE)
+   if((is(sol))[1]=="try-error"){
+     "infeasible"
+   } else {
+     sol
+   }
+   return(list(Weights=sol$solution,Return=crossprod(sol$solution,mu),
+               Risk_St_Deviation=sqrt(t(sol$solution)%*%CovMatrix%*%(sol$solution)), Variance=sol$v
+ }
> #desired return mu0, we use the value of m
> m

[1] 0.045

> #vector of mean returns
> mu.vec

   A    B    C    D
0.02 0.07 0.15 0.20

> #standard deviation vector
> stdev.vector

[1] 0.05 0.12 0.17 0.25

> # correlation matrix
> ro.mat
```

```
     [,1] [,2] [,3] [,4]
[1,]  1.0  0.3  0.3  0.3
[2,]  0.3  1.0  0.6  0.6
[3,]  0.3  0.6  1.0  0.6
[4,]  0.3  0.6  0.6  1.0
```

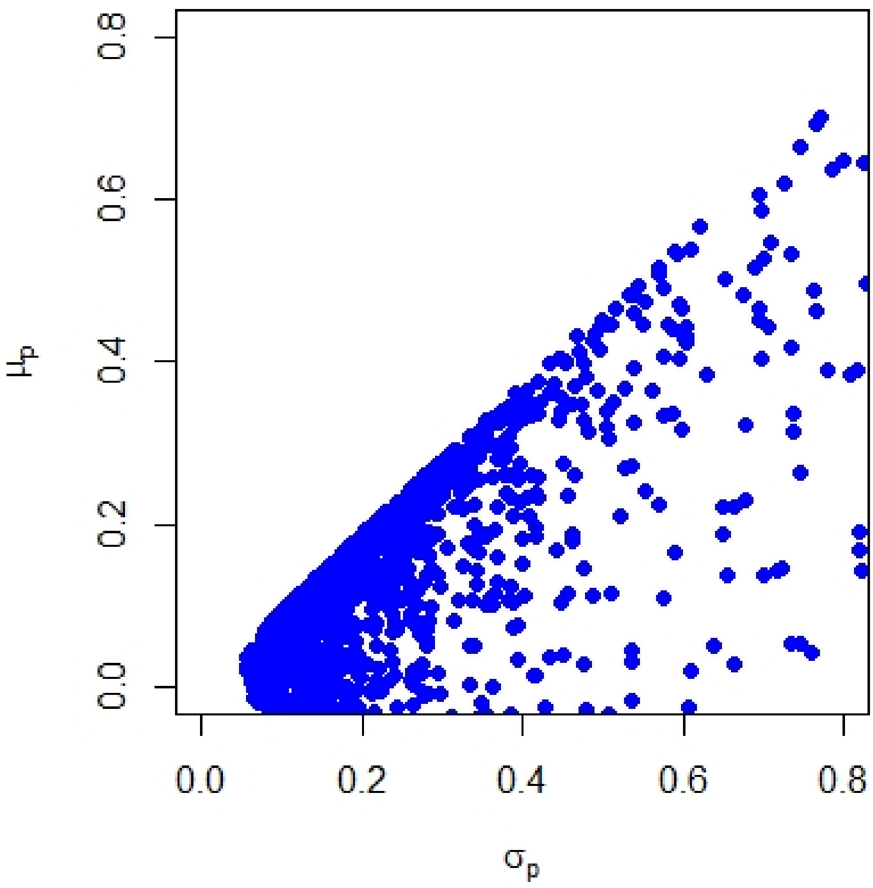
```
> # one can be up to 100% of the portfolio short in an asset
> minw=-1
> #one can be up 100% of the portfolio long in an asset
> maxw=1
> #the assets weights must be between minw and maxw as defined above
> fixed=1
> opt(m,mu.vec,stdev.vector, ro.mat ,minw,maxw,fixed)
```

```
$Weights
[1] 0.78511066 0.05386419 0.13355472 0.02747042

$Return
      [,1]
[1,] 0.045

$Risk_St_Deviation
           [,1]
[1,] 0.05840091

$Variance
[1] 0.003410666
```

We obtained exactly the same results as for the portfolio with the original (non-stressed) correlation matrix.

Finally, we generate 2001 portfolios with random allocations (i.e. random weights of the four assets). For the purpose we use the r function runif. Since no optimizations is required, the problem is reduced to calculate portfolio return (cross product of the vector with random asset weigths and the mean vector mu.vec) and the volatility of the portfolio (this repeated 2001 times). Below is a simple code in r to plot the resulting frontier.

```
>    mu_random_portfolio<-rep(0,2001)
>    sig_random_portfolio<-rep(0,2001)
>    for (i in 1:2001) {
+    w<-0
+    w_random<-c()
+    w_random<-runif(4,-1,1)   #generating a vector containing random numbers between -1 and 1
+    w_random<-w_random/sum(w_random) # this is to ensure that the weights sum up to 1
+    mu_random_portfolio[i]<-crossprod(w_random,mu.vec) # this is the return of portfolio i
+    sig_random_portfolio[i]<-sqrt(t(w_random)%*%sigma.mat%*%w_random) # this is the volatility of p
+    plot(sig_random_portfolio, mu_random_portfolio, ylim=c(0, 0.8), xlim=c(0, 0.8),
+         pch=16,col="blue",ylab=expression(mu[p]),
+         xlab=expression(sigma[p]))
+    }
```



Question 2

- Formulate optimization, the Lagrangian function and give its partial derivatives only.

The tangency portfolio is the one with the highest Sharpe ratio. The optimization problem is:

$\underset{w}{max}\frac{w^T\mu-r_f}{(w^T\sum w)^{\frac{1}{2}}}$ with a constraint of $w^T\mathbf{1}$=1

The Lagrangian for this problem is:

$L(w,\lambda) = (w^T\mu - r_f)(w^T\sum w)^{-\frac{1}{2}} + \lambda(w^T\mathbf{1} - 1)$

The partial derivatives are:

$\frac{\partial L(w,\lambda)}{\partial w} = \mu(w^T\sum w)^{-\frac{1}{2}} - (w^T\mu - r_f)(w^T\sum w)^{-\frac{3}{2}}\sum w + \lambda\mathbf{1}$

$\frac{\partial L(w,\lambda)}{\partial \lambda} = w^T\mathbf{1} - 1$

Alternatively, one can use the fact that the tangency portfolio maximises the of slope of the CML and the problem can be interpreted as finding the tangency portfolio between a line going through the point $(0, r_f)$ and the efficient frontier (where r_f denotes the risk free rate). The optimization problem can be reduced to:

$\min_{w} \frac{1}{2} w^T \sum w$

subject to $r_f + (\mu - r_f \mathbf{1})^T w = m$

Thus, the Lagrange function would be simply:

$L(w, \lambda) = \frac{1}{2} w^T \sum w + \lambda(m - r_f + (\mu - r_f \mathbf{1})^T w)$

The partial derivatives are:

$\frac{\partial L}{\partial w} = w^T \sum - \lambda(\mu - r_f \mathbf{1})^T$

$\frac{\partial L}{\partial \lambda} = m - r_f + (\mu - r_f \mathbf{1})^T w$

- For the range of tangency portfolios given by $r_f = 50bps, 100bps, 150bps, 175bps$ compute optimal allocations and σ_Π. Present results in a table.

Using the second approach (tangency portfolio maximises the of slope of the CML) and Setting the partial derivatives to 0 (first order conditions) and solving for the optimal weight vector we get:

$w^* = \lambda \sum^{-1}(\mu - r_f \mathbf{1})$

We know that this is the optimal weight vector, since the Hessian of the objective function (the covariance matix) is positive definite. From the constraint equation by substituting w^*,we can obtain expression for λ:

$\lambda = \frac{m - r^*}{(\mu - r_f \mathbf{1})^T \sum^{-1}(\mu - r_f \mathbf{1})}$, then $w^*$is reduced to:

$w^* = \frac{(m - r_f) \sum^{-1}(\mu - r_f \mathbf{1})}{(\mu - r_f \mathbf{1})^T \sum^{-1}(\mu - r_f \mathbf{1})}$

Recalling that the tangency portfolio is 100% invested in risky assets , i.e. $\mathbf{1}^T w = 1$, we finally arrive at the weights of the tangency portfolio:

$w_t = \frac{\sum^{-1}(\mu - r_f \mathbf{1})}{B - A r_f}$

```
> #case1 rf=50bps
>
> rf<-0.005
> ones_vector<-rep(1,4)
> A<-as.numeric(t(ones_vector)%*%(solve(sigma.mat))%*%ones_vector)
> A

[1] 423.615

> B<-as.numeric(t(mu.vec)%*%(solve(sigma.mat))%*%ones_vector)
> B

[1] 6.807021

> #case1 rf=50bps
>
> rf<-0.005
> w_tangency1<-(solve(sigma.mat)%*%(mu.vec-rf*ones_vector))/(B-A*rf)
> w_tangency1
```

```
        [,1]
A  0.0168352
B -0.2293670
C  0.8143403
D  0.3981915

> return_portfolio1<-as.numeric(crossprod(mu.vec,w_tangency1))
> return_portfolio1

[1] 0.1860704

> sigma_portfolio1<-as.numeric(sqrt(t(w_tangency1)%*%sigma.mat%*%w_tangency1))
> sigma_portfolio1

[1] 0.1965106

> #case2 rf=100bps
>
> rf<-0.01
> w_tangency2<-(solve(sigma.mat)%*%(mu.vec-rf*ones_vector))/(B-A*rf)
> w_tangency2

        [,1]
A -0.7459371
B -0.5105694
C  1.4902493
D  0.7662571

> return_portfolio2<-as.numeric(crossprod(mu.vec,w_tangency2))
> return_portfolio2

[1] 0.3261302

> sigma_portfolio2<-as.numeric(sqrt(t(w_tangency2)%*%sigma.mat%*%w_tangency2))
> sigma_portfolio2

[1] 0.3506654

> #case3 rf=150bps
>
> rf<-0.015
> w_tangency3<-(solve(sigma.mat)%*%(mu.vec-rf*ones_vector))/(B-A*rf)
> w_tangency3

       [,1]
A -8.644854
B -3.422571
C  8.489651
D  4.577774

> return_portfolio3<-as.numeric(crossprod(mu.vec,w_tangency3))
> return_portfolio3

[1] 1.776525

> sigma_portfolio3<-as.numeric(sqrt(t(w_tangency3)%*%sigma.mat%*%w_tangency3))
> sigma_portfolio3

[1] 1.972392
```

```
> #case3 rf=175bps
>
>
> rf<-0.0175
> w_tangency4<-(solve(sigma.mat)%*%(mu.vec-rf*ones_vector))/(B-A*rf)
> w_tangency4

       [,1]
A  8.103502
B  2.751851
C -6.351431
D -3.503922

> return_portfolio4<-as.numeric(crossprod(mu.vec,w_tangency4))
> return_portfolio4

[1] -1.298799

> sigma_portfolio4<-as.numeric(sqrt(t(w_tangency4)%*%sigma.mat%*%w_tangency4))
> sigma_portfolio4

[1] 1.473515
```

This is obviuosly not the return we would like to get when investing. Clearly, the optimization has failed. The reason why is that the global minimum portfolio (the most left point on the efficient frontier, i.e. with smallest variation) has a return which is smaller than the risk free rate. We can check that:

```
> aMat  <- array(1, dim = c(1,4))
> bVec  <- 1
> zeros <- array(0, dim = c(4,1))
> solQP <- solve.QP(sigma.mat, zeros, t(aMat), bVec, meq = 1)
> solQP$solution

[1]  0.942670800  0.111950052 -0.006062729 -0.048558123
```

Let's calculate the return:

```
> crossprod(solQP$solution,mu.vec)

           [,1]
[1,] 0.01606889
```

Now the variance:

```
> t(solQP$solution)%*%sigma.mat%*%solQP$solution

            [,1]
[1,] 0.002360634
```

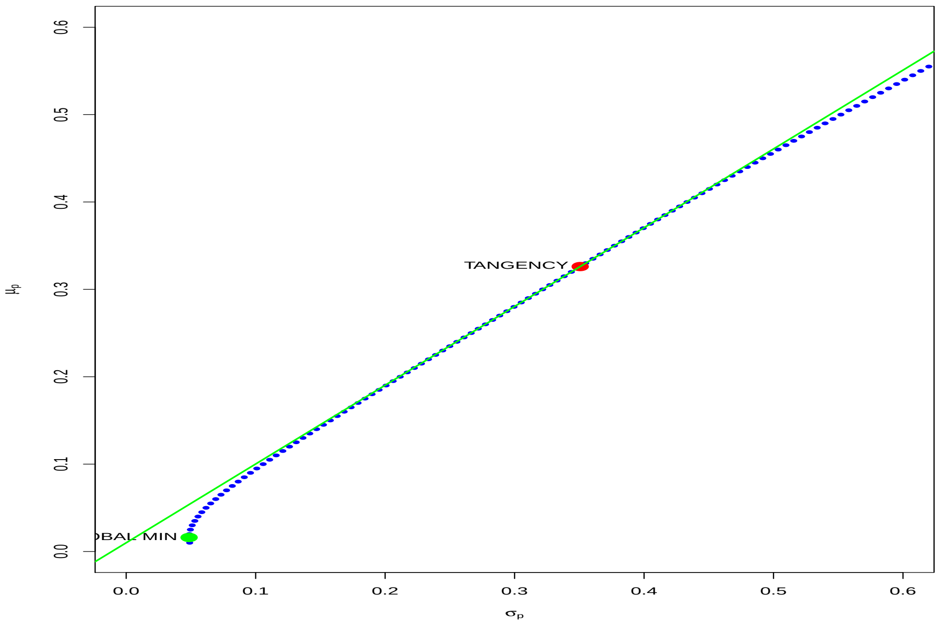
So investing in risky portfolio gives a return of appr. 0.016, whereas investing in a risk-free asset would achieve return of 0.0175. Clearly, when calculating tangency portfolio one should stop the optimization when average return of the global minimum variance portfolio becomes less than the risk free rate. In this case the optimal portfolio would be risk free rate. If we assume that the risk-free asset is not investable than the global minimum portfolio should be the optimal.

```
> return_portfolio4<-crossprod(solQP$solution,mu.vec)
> sigma_portfolio4<-sqrt(t(solQP$solution)%*%sigma.mat%*%solQP$solution)

> table_results <- cbind(c(return_portfolio1,return_portfolio2,return_portfolio3,return_portfolio4),
+                        c(sigma_portfolio1,sigma_portfolio2,sigma_portfolio3,sigma_portfolio4))
> colnames(table_results) <- c("Return","Standard Deviation")
> rownames(table_results) <- c("rf=50bps","rf=100bps","rf=150bps", "rf=175bps")
> results <- as.table(table_results)
> results
```

```
              Return Standard Deviation
rf=50bps   0.1860704          0.1965106
rf=100bps  0.3261302          0.3506654
rf=150bps  1.7765254          1.9723917
rf=175bps  0.0160688          0.0485863
```

To plot the efficient frontier one can use the global minimum portfolio and construct with it the rest of the efficient frontier (the the set of efficient portfolios of risky assets can be computed as a convex combination of any two efficient portfolios). Alternatively we can make use of the opt function coded in Question1, which finds optimal portfolio allocations for given target return. One can run optimizations for low discrepancy of target returns to build the efficient frontier. In case of rf=100bps we have:

```
> mu0<-seq(0.01,1, by=0.005)
> solution<-c()
> risk_vector<-rep(0,length(mu0))
> returns<-rep(0,length(mu0))
> for (i in 1:length(mu0)) {
+   solution<-opt(mu0[i],mu.vec,stdev.vector, ro.mat ,-1,1,1)
+   risk_vector[i]<-solution$Risk_St_Deviation
+   returns[i]<-solution$Return
+   plot(risk_vector, returns, ylim=c(0, 0.6), xlim=c(0, 0.6),
+        pch=20,col="blue",ylab=expression(mu[p]),
+        xlab=expression(sigma[p]))
+ }
> points(gmin.port$sd, gmin.port$er, col="green", pch=16, cex=2)
> points(tan.port$sd, tan.port$er, col="red", pch=16, cex=2)
> text(gmin.port$sd, gmin.port$er, labels="GLOBAL MIN", pos=2)
> text(tan.port$sd, tan.port$er, labels="TANGENCY", pos=2)
> sr.tan = (tan.port$er - r.free)/tan.port$sd
> abline(a=r.free, b=sr.tan, col="green", lwd=2)
>
```



This is still concave function, but rather steep. The reason for that could be the relative high positive correlation between the four assets.

VaR Backtesting

Question 3

The calculations and the backtesting of the two VaR methods can be found in the excel file provided. Here are presented the results with some explanations. I have called the first method Simple VaR and the second EWMA VaR. The table below summarizes the findings[1] :

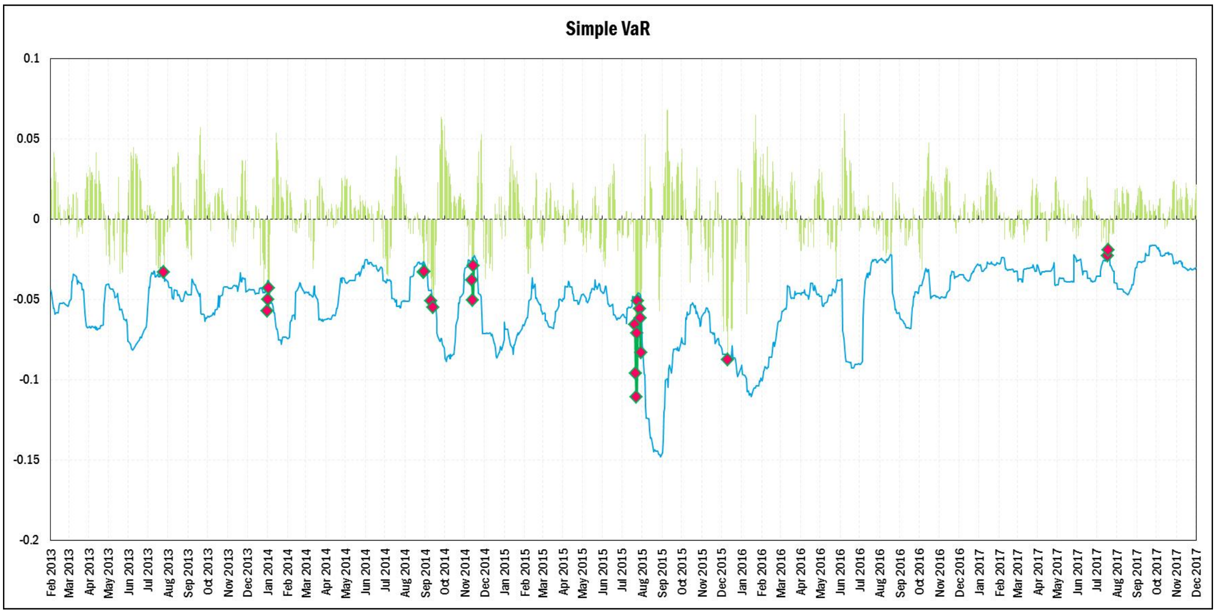
Stats						*Unconditional coverage*			*Independence*			*Conditional coverage*		
Method Code	Method	N	NOS	POF	VaR CF	Kstat	K crit 99%	K Rejected	n00	Ind crit 99%	Ind rejected	Ucstat	UC crit 99%	UC rejected
1	Simple VaR	1220	22	1.80%	1.00%	6.42	6.63	FALSE	1188	6.63	TRUE	90.95	9.21	TRUE
2	EWMA VaR	1220	31	2.54%	1.00%	20.51	6.63	TRUE	1172	6.63	TRUE	96.80	9.21	TRUE

As we can see the Simple VaR detects 22 overshootings for 1220 observations, or 1.80%. In this case, using the famous Kupiec test, we wouldn't reject the hypothesis that the model is correct. Rejection of the null hypothesis, indicates statistically significant deviation between the observed and expected overshootings pattern behaviour for the model (frequency - Kupiec, independence - Christoffersen). The Christoffersen test shows however that overshootings are not independent. Indeed, one can see that there are several periods of consecutive overshootings (for instance in august 2015)[2] :

Table 1: Consecutive days of overshootings - Simple VaR

Date of overshootings	**Number of consecutive days of overshootings**
21.01.2014	3
18.09.2014	2
02.12.2014	3
10.08.2015	10
07.08.2017	2

Below, we plot the overshootings:



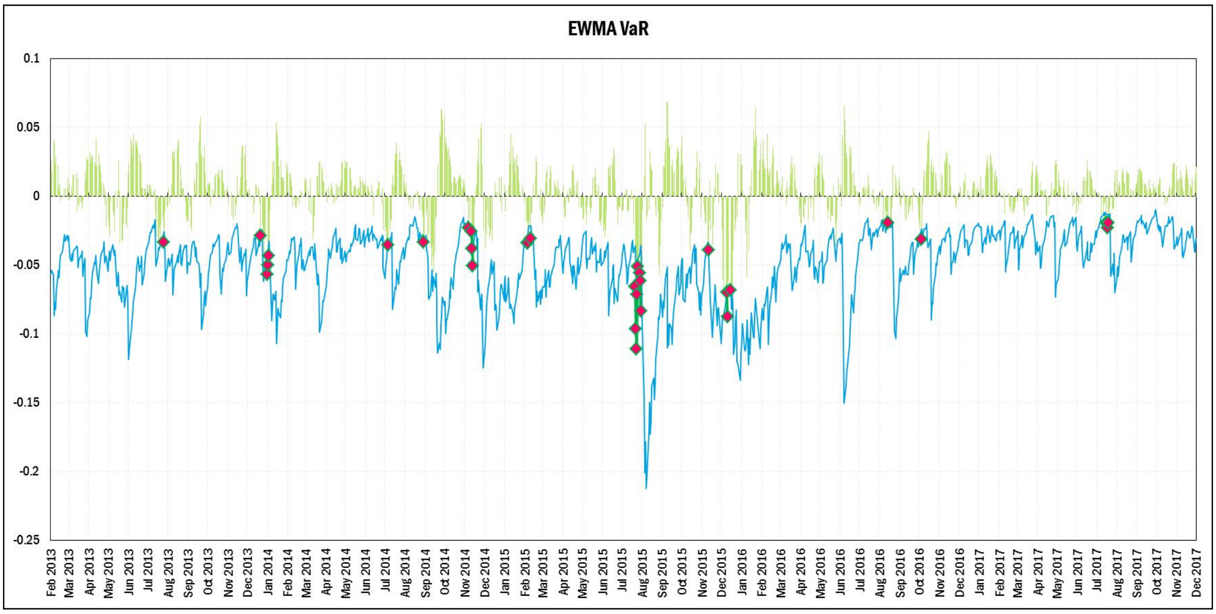
[1]The table can be found in sheet DATA of the excel file provided
[2]..the yuan depreciation

EWMA VaR shows 31 out of 1220 observations or 2.54%. This time Kupiec has been rejected, i.e. one can accept that the risk management model is correct (unless there are exceptional reasons). Christoffersen test has been also rejected. The table below presents the periods of consecutive overshootings:

Table 2: Consecutive days of overshootings - EWMA VaR

Date of overshootings	**Number of consecutive days of overshootings**
21.01.2014	3
01.12.2014	2
25.02.2015	2
10.08.2015	10
29.12.2015	2
07.08.2017	2

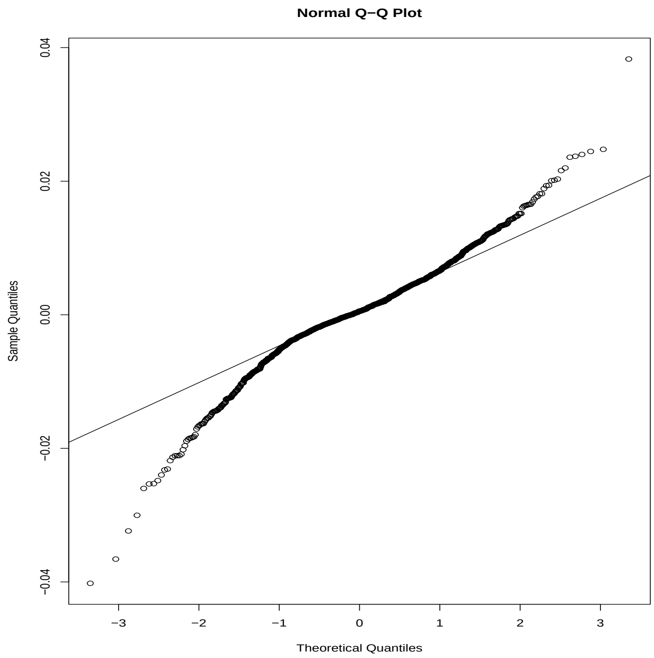
Below, the plot of the overshootings:



The calculation (sheet SP500 VaR Calculations) for the EWMA VaR has been done with a VBA function EWMAV (this is 1D VaR, with Prices - array of price observations, Lambda - decay factor, dConf - confidence power. One must execute the function for 21 price observations on a rolling principle, then multiply by $\sqrt{10}$ when comparing to 10D holding period returns):

```
Function EWMAV(Prices As range, Lambda As Double, dConf As Double) As Double
    Dim vPrices As Variant
    Dim dSumWtdRtn As Double
    Dim i As Long
    Dim dLogRtn As Double, dRtnSQ As Double, dWT As Double, dWtdRtn As Double

vPrices = Prices
For i = 2 To UBound(vPrices, 1)
    dLogRtn = Log(vPrices(i, 1) / vPrices(i - 1, 1))
    dRtnSQ = dLogRtn ^ 2
    dWT = (1 - Lambda) * Lambda ^ (UBound(vPrices, 1) - i)
    dWtdRtn = dWT * dRtnSQ
    dSumWtdRtn = dSumWtdRtn + dWtdRtn
Next i

EWMAV = (dSumWtdRtn ^ (1 / 2)) * Application.WorksheetFunction.Norm_S_Inv(dConf)
End Function
```

The Simple VaR on 99% confidence test would conclude that the model is well calibrated (there are 1.80% exceptions which is almost in line with 1% leeway left from 99% VaR). The problem of this method is the weighting structure (equal weight for every return) which creates the potential for "ghost" effects. The equal-weight structure also presumes that each observation in the sample period is equally likely and independent of the others over time, which as we have seen from the Christoffersen's test is not the case). Overall, as far as I am concerned, the EWMA VaR proves to be more robust model (other terms equal - eg. no exceptional market events). However, the question is whether the normal distribution assumption is appropriate at all. From a simple Q-Q plot, one can easily conclude that the timeseries of S&P exhibits fat tails. To accommodate the tail risk, a more sophisticated method like Filtered Historical Simulation (that combines the benefits of Historical Simulation and conditional volatility approach) could be used.



Question 4

From the text we have that the Liquidity-Adjusted VaR (LVAR) is computed as:

$$LVaR = VaR + \triangle_{Liquidity} = PortfolioValue \times \left[-\mu + Factor \times \sigma + \tfrac{1}{2}(\mu_{spread} + Factor \times \sigma_{spread})\right]$$

a)

From standard normal tables we know that 99% VAR corresponds to 2.326 times sigma. The formula for LVAR suggests the same factor (i.e. assuming standard normal distribution) to be used both for the VaR component and the liquidity component. Given that Portfolio Value of USD 16 million invested in a share with $\mu = 0.01$ and $\sigma = 0.03$, $\mu_{spread} = 0.0035$ and $\sigma_{spread} = 0.0150$ we can substitute in the formula for to find LVAR:

$$LVaR = 16 \times \left[-0.01 + 2.326 \times 0.03 + \tfrac{1}{2}(0.0035 + 2.326 \times 0.0150)\right] = 1.54272 \text{ million USD}$$

with attributions of VaR=$16 \times (-0.01 + 2.326 \times 0.03) = 0.95648$million USD and $\triangle_{Liquidity} = \frac{1}{2}(0.0035 + 2.326 \times 0.0150) = 0.58624$ million USD

We can also compute the ratio $\frac{LVaR}{VaR} = \frac{1.54}{0.95} \approx 1.62$

b)

Given Portfolio Value of USD 40 million invested in a UK gilts with $\mu = 0.01$ and$\sigma = 0.03$, and $\mu_{spread} = 0.0035$ and $\sigma_{spread} = 0.0150$ with no bid-ask spread volatility, the LVaR will be:

$40 \times (2.326 \times 0.03 + 0.5 \times 0.0035) \approx$ 2.86 million USD,

with VaR and Liquidity spread attribution of 2.79 mil. USD and 0.07 mil. USD. The LVaR of UK gilts porrtfiolio is higher than the of the tech stock portfolio, however, one can see that this mainly because of the size of the portfolio and the VaR component with relatively negligable contribution of the liquidity component. This can be seen also from the ratio $\frac{LVaR}{VaR} = 1.025$.

Given a bid-ask spread 125 bps, the LVaR will be:

$40 \times (2.326 \times 0.03 + 0.5 \times 0.0125) \approx$ 3.04 mil. USD

VaR and bid-ask spread attribution would be$40 \times 2.326 \times 0.03 = 2.79$ mil. USD and$40 \times 0.5 \times 0.0125 = 0.25$ mil.USD resp. The $\frac{LVaR}{VaR} \approx 1.09$. As expected, the ratio has risen, due to a higher proportion in LVaR of the liquidity component.

Question 5

a) Which types of p & L the firm wide VaR backtesting operates with?

As stated in paragraph 32.4 of the regulation MAR32, one-day holding period VaR is compared against each of the **actual P&L (APL)** and **hypothetical P&L (HPL)** over the prior 12 months.

The actual return corresponds to the actual P&L, taking into account intraday trades and other profit items such as fees, commissions, spreads, and net interest income.

The hypothetical return represents a frozen portfolio, obtained from fixed positions applied to the actual returns on all securities, measured from close to close.

An exception (or an outlier) is marked when either the actual loss or the hypothetical loss of the bank-wide trading book registered in a day from the backtesting period exceeds the corresponding daily VaR, as measured by the model. In case the P&L or VaR are not available (or not possible to be computed), an outlier is recorded. [3]

b) What are two key metrics chosen to compare risk-theoretical (RTPL) and hypothetical (HPL)

The two key test metrics utilized to compare RTPL and HPL are[4] :

1. the **Spearman correlation** metric - to assess correlation between RTPL and HPL; and
2. the **Kolmogorov-Smirnov (KS)** test metric - to assess similarity of the distributions of RTPL and HPL

For the calculation of these metrics, a bank must use time series of observations of RTPL and HPL from the most recent 250 days. To determine the Spearman correlation metrics, it must be:

- Produced two time series representing ranking based on the value of the P&L (one time series for R_{HPL} and similiarly one time series for R_{RTPL}). Lowest value receives a rank of 1, next lowest rank of 2 and so on

[3]MAR32, Paragraph 32.5
[4]MAR32, Paragraph 32.34

- Calculate the Spearman correlation coefficient of the R_{HPL} and R_{RTPL} using the formula:

$r_s = \frac{cov(R_{HPL} R_{RTPL})}{\sigma_{R_{HPL}} \times \sigma_{R_{RTPL}}}$

The process for determining Kolmogorov-Smirnov test metrics is as follows:

- Calculation of the empirical cumulative distribution function of RTPL. The empirical cumulative distribution function is the product of 0.004 and the number of RTPL observations less than or equal to the specified RTPL.
- Calculation of the empirical cumulative distribution function of HPL following the same procedure as the one for RTPL
- The KS metric is the largest absolute difference observed between the above mentioned empirical cumulative distributions function at any P&L value.

c) How many exceptions are allowed in a trading desk-level backtesting at 99th percentile before it will be forced to exercise the standardised approach.

Paragraph 32.19 of the MAR32 regulation states that if any trading desk experiences more than 12 exceptions at the 99th percentile in the most recent 12 month period, the capital requirement for all of the positions in the trading desk must be determined using the standardised approach. This assumes that there are no exceptional situations like regime shifting or significant cross-border financial market stress affecting several banks [5].

According to paragraph 32.43 of the MAR32 regulation, if a trading desk is in the PLA red zone, it is ineligible to use the IMA to determine market risk and must use the standardised approach. Hence, a trading desk could be forced to use the standardised approach also if the Spearman correlation is below 0.70 or the KS test is below 0.12 (p-value of 0.055).

[5]MAR32, 32.45, Treatment for exceptional situations