# STATS 509 HOMEWORK 4

*Yuan Yin*

*2/4/2018*

## Problem 3

### (a)

First we read the data and delete the missing value:

```
data1 = read.csv("Nasdaq_wklydata_92-12.csv", header = T)
data2 = read.csv("SP400Mid_wkly_92-12.csv", header = T)
idx1 = which(is.na(data1$Adj.Close) == FALSE)
idx2 = which(is.na(data2$Adj.Close) == FALSE)
nas = rev(data1$Adj.Close[idx1])
sp400 = rev(data2$Adj.Close[idx2])
```

Then we compute the log-return:

```
nas_lreturn = diff(log(nas))
sp400_lreturn = diff(log(sp400))
```

We first compute the sample mean and covariance matrix with MLE:

```
multi_vector = cbind(nas_lreturn, sp400_lreturn)
mean_lreturn = signif(colMeans(multi_vector))
cov_lreturn = signif(cov(multi_vector))
mean_lreturn
```

```
##   nas_lreturn sp400_lreturn
##    0.00154143    0.00176529
```

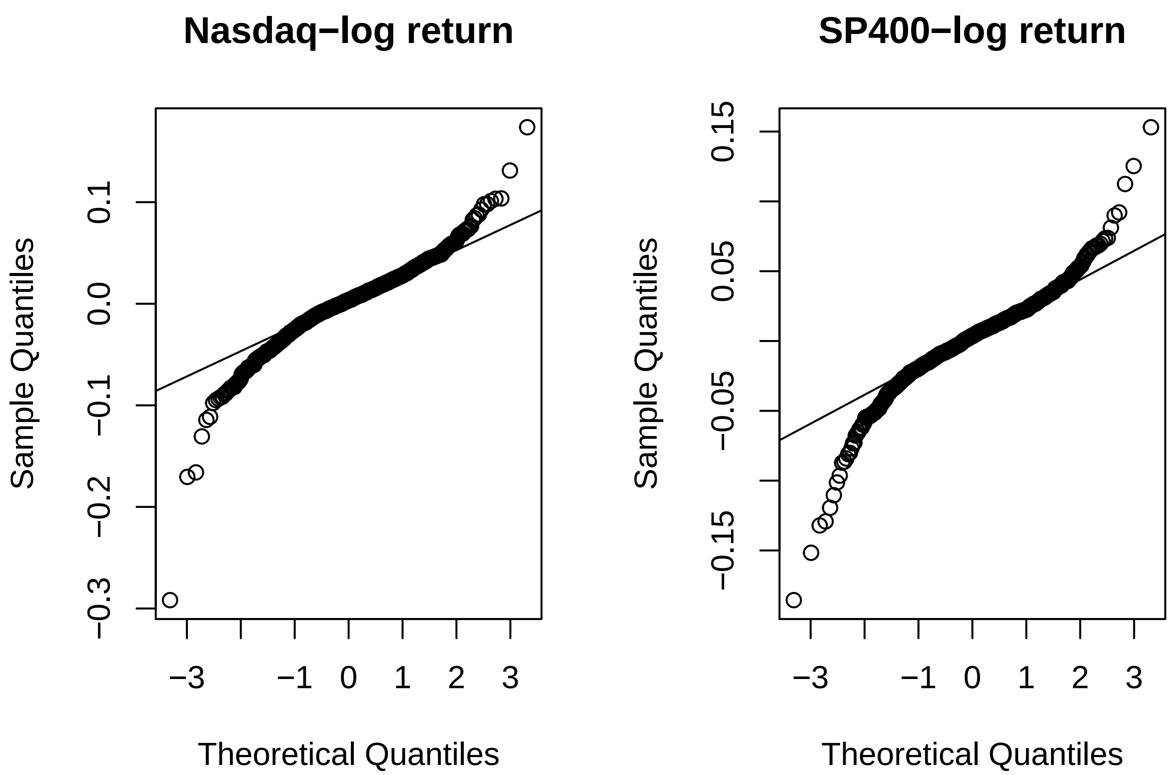
```
cov_lreturn
```

```
##                nas_lreturn sp400_lreturn
## nas_lreturn     0.00110335   0.000784460
## sp400_lreturn  0.00078446   0.000752856
```

In this way, we compute the sample mean and covariance matrix as above. Now we want to fit our data with normal distribution and finding the correlation.

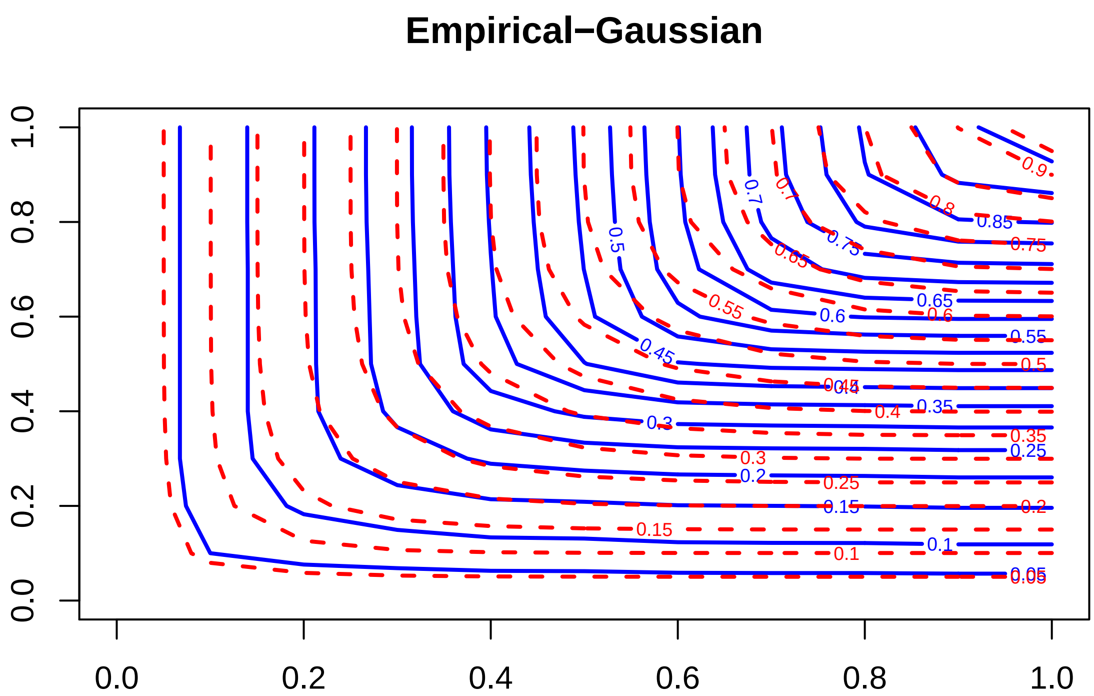
Now we polt the QQ-plot to see the goodness of our assumption.

```
par(mfrow = c(1,2))
qqnorm(nas_lreturn, main = "Nasdaq-log return")
qqline(nas_lreturn)
qqnorm(sp400_lreturn, main = "SP400-log return")
qqline(sp400_lreturn)
```




It shows that both log-returns has a significantly heavier tail than normal distribution. Now let's plot the empirical vs. theoretical cdf and look at the difference.

```
corr = cov_lreturn[1,2]/(sqrt(cov_lreturn[1,1])*sqrt(cov_lreturn[2,2]))
dat1 = cbind(pnorm(nas_lreturn, mean = mean_lreturn[1], sd = sqrt(cov_lreturn[1,1])),
             pnorm(sp400_lreturn, mean =mean_lreturn, sd = sqrt(cov_lreturn[2,2])))
# empirical copula
u1 = dat1[,1]
u2 = dat1[,2]
dem = pempiricalCopula(u1,u2)
contour(dem$x, dem$y, dem$z, main = "Empirical-Gaussian",
        col = 'blue', lty = 1, lwd = 2, nlevel = 20)
# theoretical copula
cn = normalCopula(corr, dim = 2, dispstr = "un")
utdis = rCopula(100000, cn)
demt = pempiricalCopula(utdis[,1], utdis[,2])
contour(demt$x, demt$y, demt$z, main = "Gauss",
        col = 'red', lty = 2, lwd = 2, add = TRUE, nlevel = 20)
```




The correlation we find is $\rho = 0.8607127$. Also from the plot above we can find that the difference of two cdf is quite obvious on tails data. Both marginal and joint distribution show that our data doesn't fit well under normal distribution. We should find some other distribution fits better.

## (b)

If we want to fit into a t-distribution, we need to find suitable degree of freedom. Let's do all things again under t-distribution. First we use profile likelihood to find suitable degree of freedom.

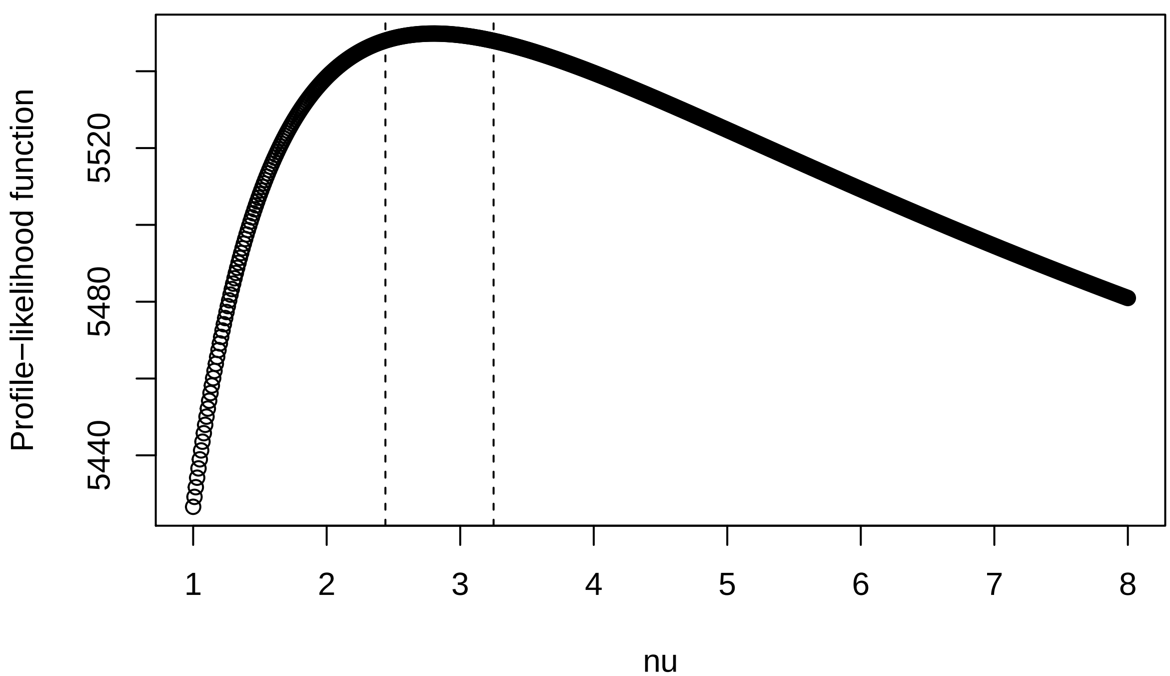
```
library(mnormt)
df = seq(1, 8, 0.01)
n = length(df)
loglik_max = rep(0, n)
for(i in 1:n){
  fit = cov.trob(multi_vector, nu = df[i])
  mu = as.vector(fit$center)
  sigma = matrix(fit$cov, nrow = 2)
  loglik_max[i] = sum(log(dmt(multi_vector, mean = fit$center, S = fit$cov, df = df[i])))
}
plot(df, loglik_max, xlab = 'nu', ylab = 'Profile-likelihood function')
nuest = df[which.max(loglik_max)]
nuest
```

```
## [1] 2.8
```

```r
# using the code:
# which(abs(loglik_max - loglik_max[which.max((loglik_max))]) <= 1.92)
# and we find that the interval should begin with 145 and end with 226
# where 1.92 is computed by 1/2*chi-square(95%)
c(df[145], df[226])
```

```
## [1] 2.44 3.25
```

```r
CI_left=df[145]
CI_right=df[226]
abline(v = CI_left, lty = 2)
abline(v = CI_right, lty = 2)
```




Finding that the df which maximize profile likelihood is 2.8, with confidence interval of [2.44, 3.25]

```r
fitfinal = cov.trob(cbind(nas_lreturn, sp400_lreturn), nu = nuest)
mu = fitfinal$center
sigma = fitfinal$cov
mu
```

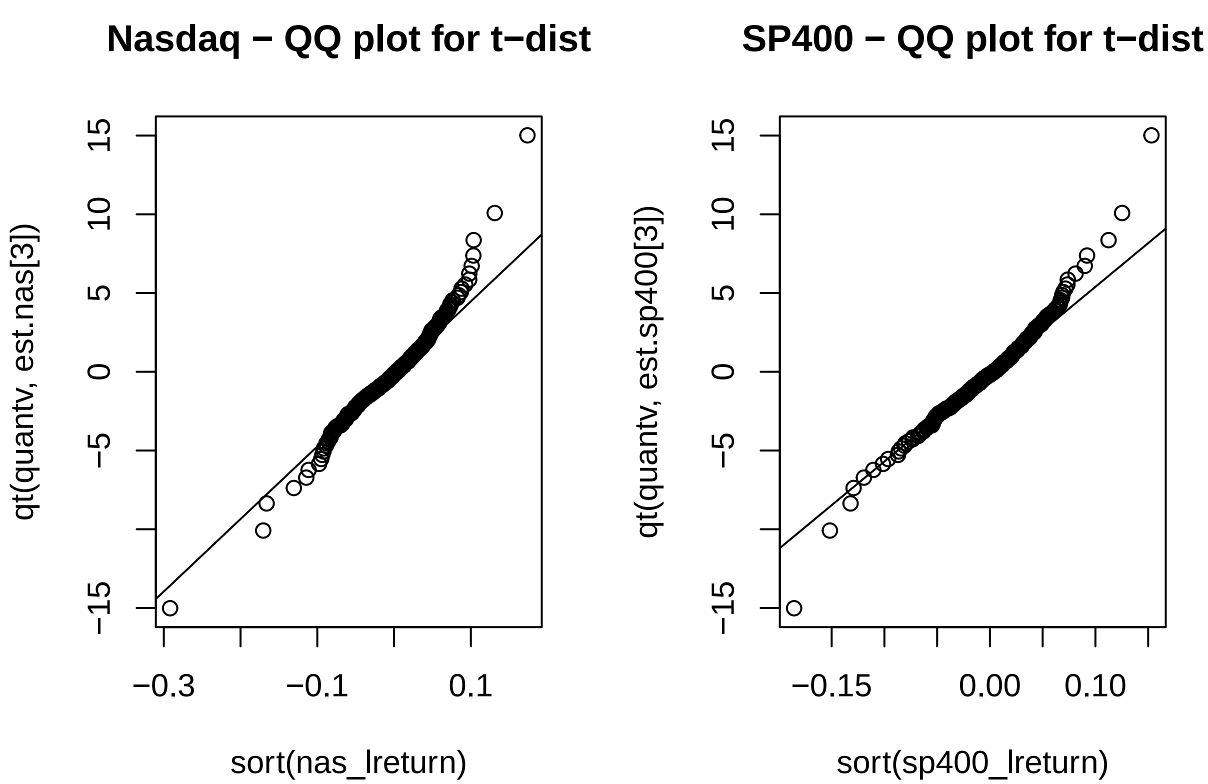
```
##   nas_lreturn sp400_lreturn
##   0.003894330   0.003342333
```

```r
sigma
```

```
##                nas_lreturn sp400_lreturn
## nas_lreturn   0.0004257728  0.0003237975
## sp400_lreturn 0.0003237975  0.0003135155
```

Thus we fit our data with t-distribution with profile likelihood method. The mean and covariance is as above Now let's look at what QQ-plot looks like.

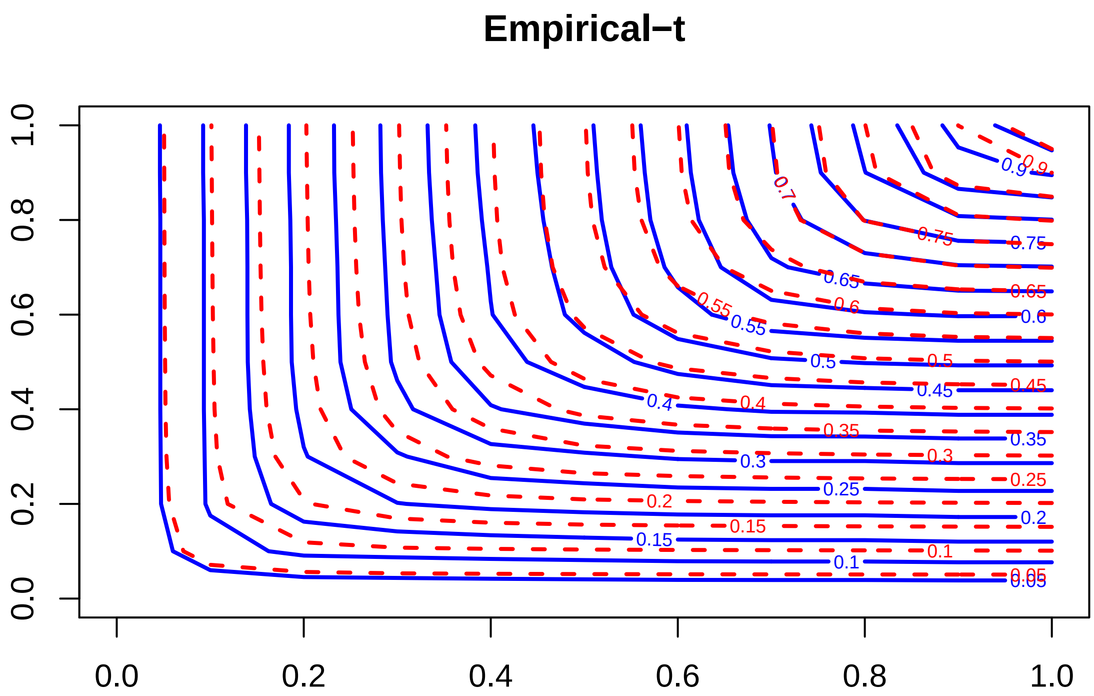
```
est.nas = as.numeric(c(mu[1], sqrt(sigma[1,1]), nuest))
est.sp400 = as.numeric(c(mu[2], sqrt(sigma[2,2]), nuest))
# Need to convert to standard deviation for incorporating within "pstd"
est.nas[2] = est.nas[2] * sqrt(est.nas[3] / (est.nas[3]-2))
est.sp400[2] = est.sp400[2] * sqrt(est.sp400[3] / (est.sp400[3]-2))
N = length(nas_lreturn)
quantv = (1/N)*seq(0.5, N - 0.5, 1)
par(mfrow = c(1,2))
qqplot(sort(nas_lreturn), qt(quantv, est.nas[3]), main = 'Nasdaq - QQ plot for t-dist')
abline(lm(qt(c(0.25, 0.75), est.nas[3]) ~ quantile(nas_lreturn, c(0.25, 0.75))))
qqplot(sort(sp400_lreturn), qt(quantv, est.sp400[3]), main = 'SP400 - QQ plot for t-dist')
abline(lm(qt(c(0.25, 0.75), est.sp400[3])~quantile(sp400_lreturn, c(0.25, 0.75))))
```




We can see that t-distribution has better linear QQ-plot than normal distribution, although the tails fit still not that good, where Nasdaq is a little heavy tail and the right tail has a little lighter tail. Now we compare the theorical cdf and empirical cdf.

```
corr = sigma[1,2]/(sqrt(sigma[1,1])*sqrt(sigma[2,2]))
dat2 = cbind(pstd(nas_lreturn, mean = est.nas[1], sd = est.nas[2], nu = est.nas[3]),
             pstd(sp400_lreturn, mean = est.sp400[1],
                  sd = est.sp400[2], nu = est.sp400[3]))
# empirical cdf
u1 = dat2[,1]
u2 = dat2[,2]
```

```r
dem = pempiricalCopula(u1,u2)
contour(dem$x, dem$y, dem$z, main = "Empirical-t",
        col = 'blue', lty = 1, lwd = 2, nlevel = 20)
# Generating initial estimate of correlation for t-copula
n = length(nas_lreturn)
ct = tCopula(corr, dim = 2, dispstr = "un", df = nuest)
utdis = rCopula(100000, ct)
demt = pempiricalCopula(utdis[,1], utdis[,2])
contour(demt$x, demt$y, demt$z, main = "t",
        col = 'red', lty = 2, lwd = 2, add = TRUE, nlevel = 20)
```




Thus using maximum pseudo-likelihood method, we get the estimate of correlation under t-distribution: $\rho = 0.8862476$, and the estimate of degree of freedom is 2.8 with CI = [2.44, 3.25]. Also we can see the difference between empircal cdf and t-distribution cdf, there is less difference than fitting in normal distribution.

## (c)

Based on results in (a) and (b), the QQ-plot seems better of t-distribution. Also the cdf of t-distribution seems better when considering tails. Thus, we should choose t-distribution.

```r
AIC_norm = -2*sum(log(dmvnorm(x = cbind(nas_lreturn, sp400_lreturn), mean = mean_lreturn, si
AIC_t = -2*max(loglik_max) + 2*6
AIC_norm;AIC_t
```

```
## [1] -10451.28
```

```
## [1] -11087.61
```

Seeing that AIC for t-distribution is less than normal distribution, which confirm our idea.

## (d)

First we use model derived in (a) which is a normal distribution model.

```
set.seed(12345678)
w = 0.5
mu1 = w*mean_lreturn[1]+(1-w)*mean_lreturn[2]
sd1 = sqrt(0.5^2*cov_lreturn[1,1] + 0.5^2*cov_lreturn[2,2]+2*0.5^2*cov_lreturn[1,2])
VaR1 = -(exp(qnorm(0.001, mu1, sd1)) - 1)*10^7
```

We get VaR = 849475.6

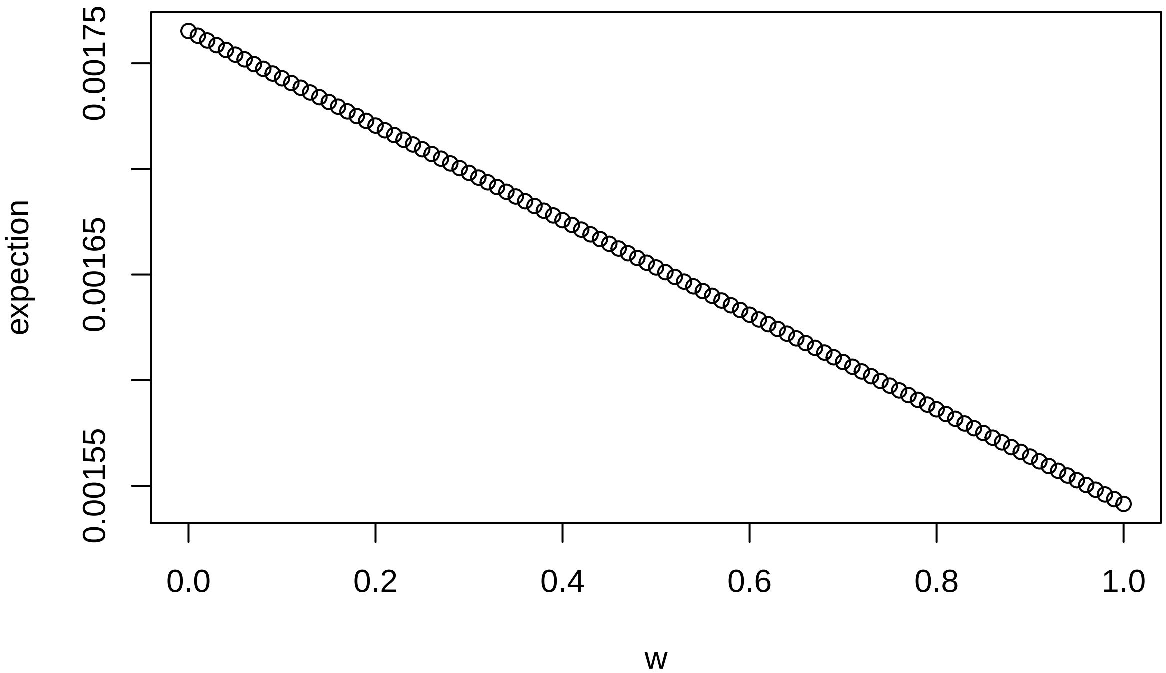
Then we use model derived in (b) which is a t-distribution model.

```
set.seed(12345678)
w = 0.5
mu_t = w*mu[1]+(1-w)*mu[2]
sd_t = sqrt(0.5^2*sigma[1,1] + 0.5^2*sigma[2,2]+2*0.5^2*sigma[1,2])
VaR = -(exp(qt(0.001,nuest)*sd_t+mu_t)-1)*10^7
```

Thus we get VaR = 1877063

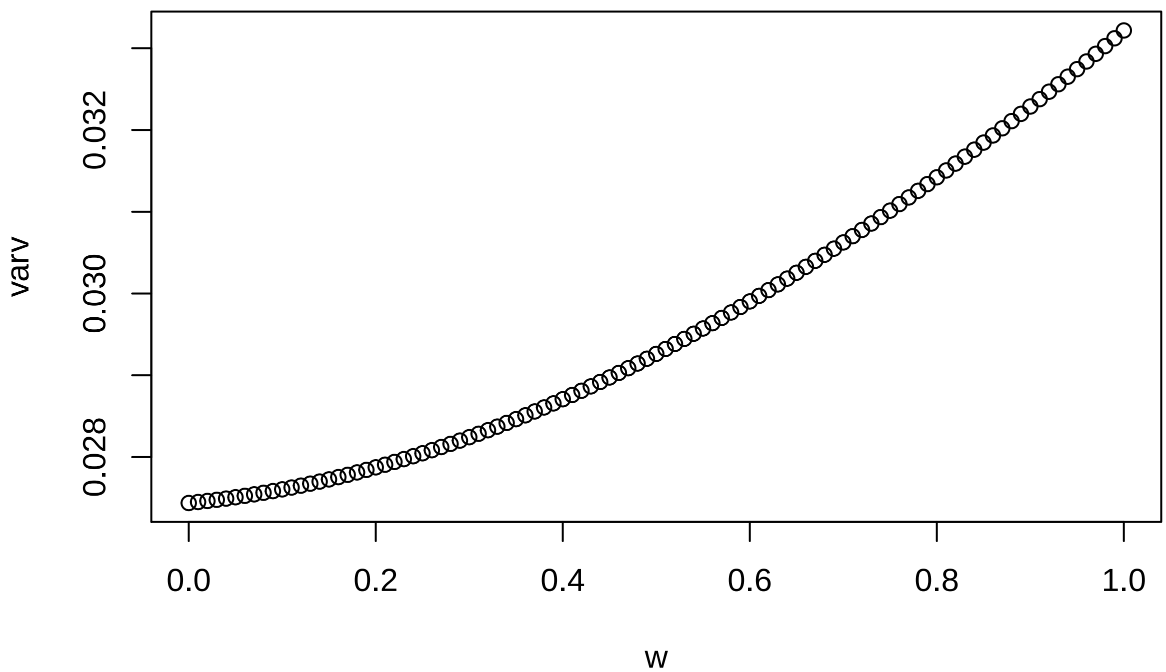
## (e)

```
dist_portpolio = function(w){
mu_portfoilio = w*mean_lreturn[1]+(1-w)*mean_lreturn[2]
sd_portfoilio = sqrt(w^2*cov_lreturn[1,1]
+(1-w)^2*cov_lreturn[2,2]+2*w*(1-w)*cov_lreturn[2,1])
return(c(mu_portfoilio,sd_portfoilio))
}
set.seed(12345678)
w = seq(0, 1, 0.01)
n = length(w)
VaRv = rep(0, n)
varv = rep(0, n)
expection = rep(0, n)
for (i in 1:n){
  x = dist_portpolio(w[i])
  expection[i] = dist_portpolio(w[i])[1]
  varv[i] = dist_portpolio(w[i])[2]
  VaRv[i] = -(exp(qnorm(0.002, dist_portpolio(w[i])[1], dist_portpolio(w[i])[2]))-1)
}
plot(w, expection)
```



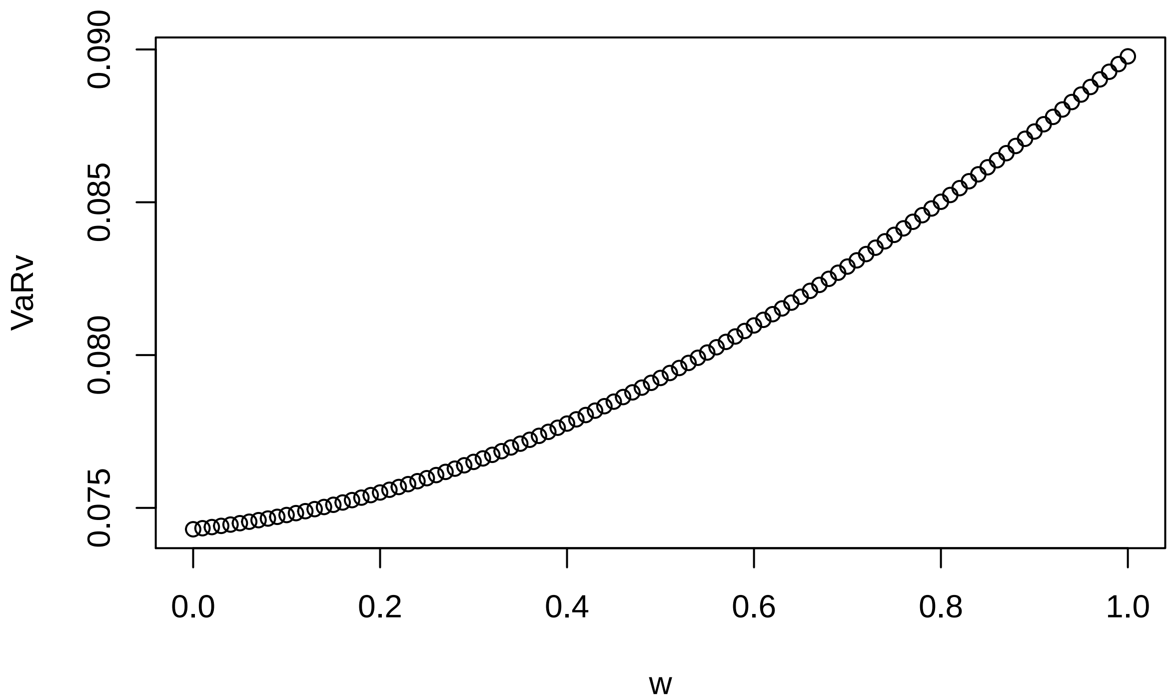
0.00155
0.00165
0.00175
expection
0.0
0.2
0.4
0.6
0.8
1.0
w


```r
plot(w, varv)
```



0.028
0.030
0.032
varv
0.0
0.2
0.4
0.6
0.8
1.0
w

```r
plot(w, VaRv)
```




```r
w[which.max(expection)]
```

```
## [1] 0
```

```r
w[which.min(varv)]
```

```
## [1] 0
```

```r
w[which.min(VaRv)]
```

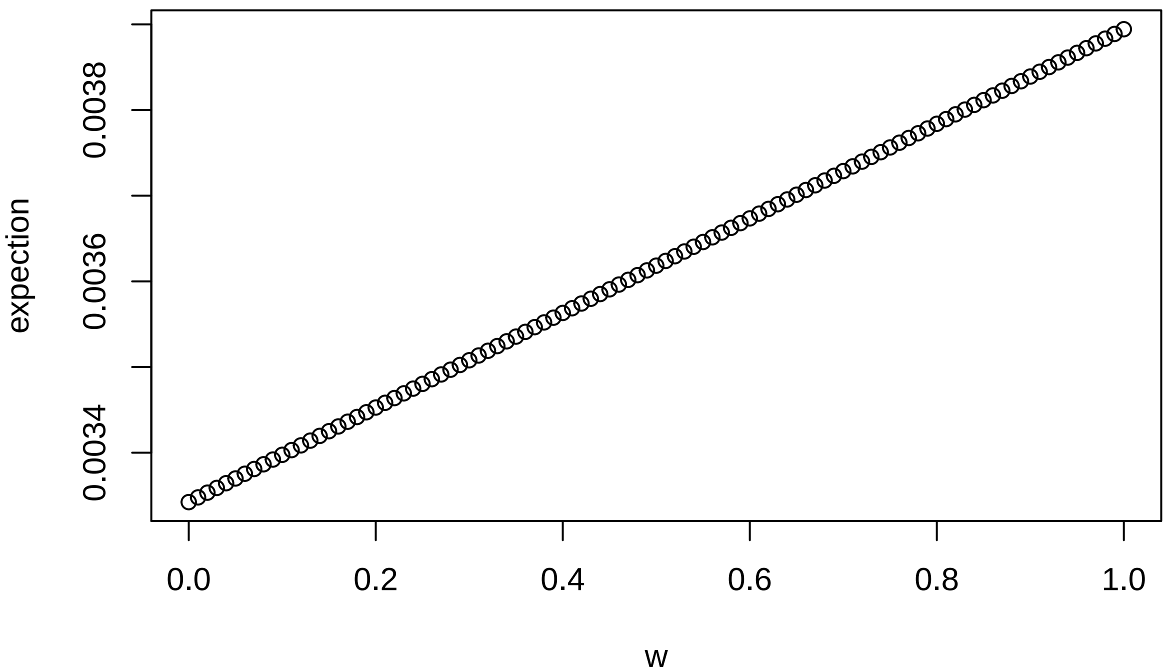
```
## [1] 0
```

Thus when we want to minize volatility or minimize VaR at 0.002 or we want to maximize expected return, we should all choose w = 0.

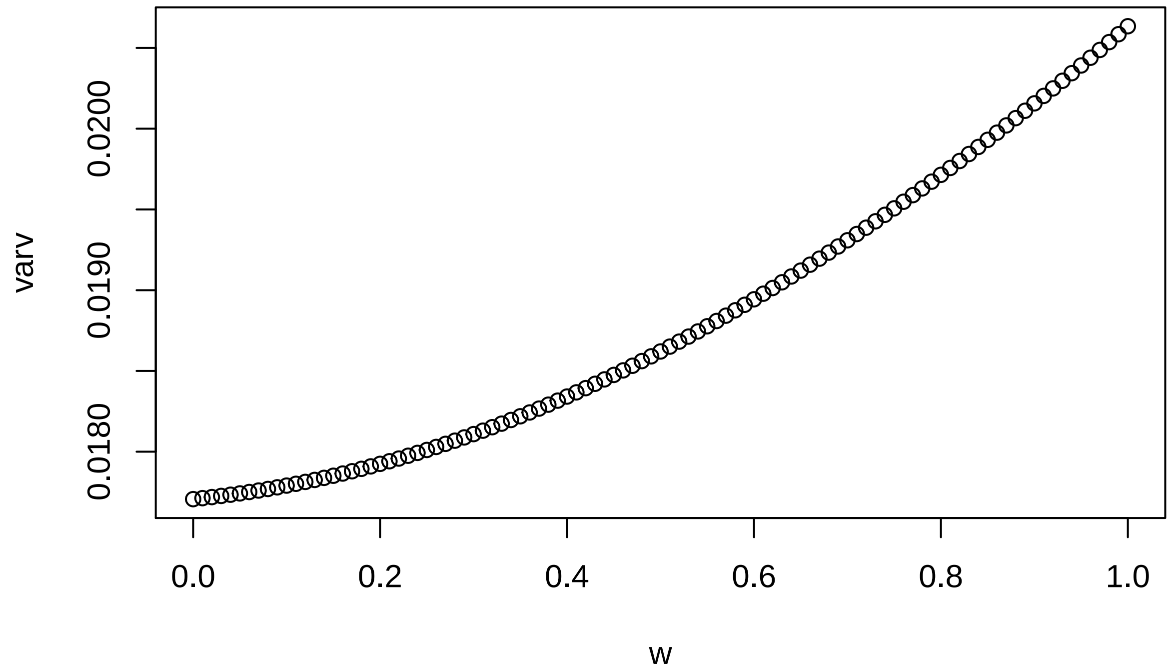
for t distribution

```r
dist_portpolio = function(w){
mu_portfoilio = w*mu[1]+(1-w)*mu[2]
sd_portfoilio = sqrt(w^2*sigma[1,1]
+(1-w)^2*sigma[2,2]+2*w*(1-w)*sigma[2,1])
return(c(mu_portfoilio,sd_portfoilio))
}
set.seed(12345678)
w = seq(0, 1, 0.01)
n = length(w)
VaRv = rep(0, n)
```

```r
varv = rep(0, n)
expection = rep(0, n)
for (i in 1:n){
  x = dist_portpolio(w[i])
  expection[i] = dist_portpolio(w[i])[1]
  varv[i] = dist_portpolio(w[i])[2]
  VaRv[i] = -(exp(qt(0.002, df = nuest)*x[2]+x[1])-1)
}
plot(w, expection)
```

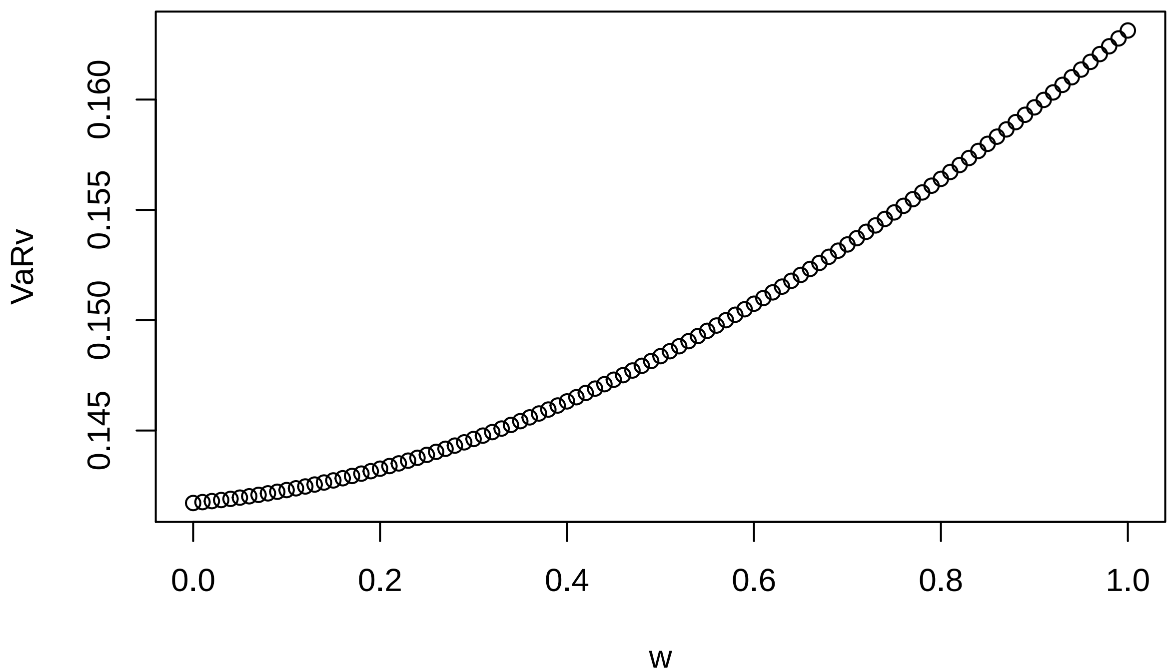



```r
plot(w, varv)
```



varv
0.0180
0.0190
0.0200
0.0
0.2
0.4
0.6
0.8
1.0
w


```r
plot(w, VaRv)
```



VaRv
0.145
0.150
0.155
0.160
0.0
0.2
0.4
0.6
0.8
1.0
w

```r
w[which.max(expection)]
```

```
## [1] 1
```

```r
w[which.min(varv)]
```

```
## [1] 0
```

```r
w[which.min(VaRv)]
```

```
## [1] 0
```

Thus when we want to minize volatility or minimize VaR at 0.002, we should choose w = 0, which means we only invest in SP400, if we want to maximize expected return, we should choose w = 1.