STATS 509 HOMEWORK 5

Yuan Yin

2/11/2018

Question 2

first import and deal with the data:

```
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])
nas_lreturn = diff(log(nas))
sp400_lreturn = diff(log(sp400))
data = cbind(nas_lreturn,sp400_lreturn)
```

Then we compute the mean and variance of two datas by t-distribution via MLE method.

```
est.nas = as.numeric(fitdistr(nas_lreturn, "t")$estimate); est.nas
```

```
## [1] 0.003350693 0.023018961 3.674359674
```

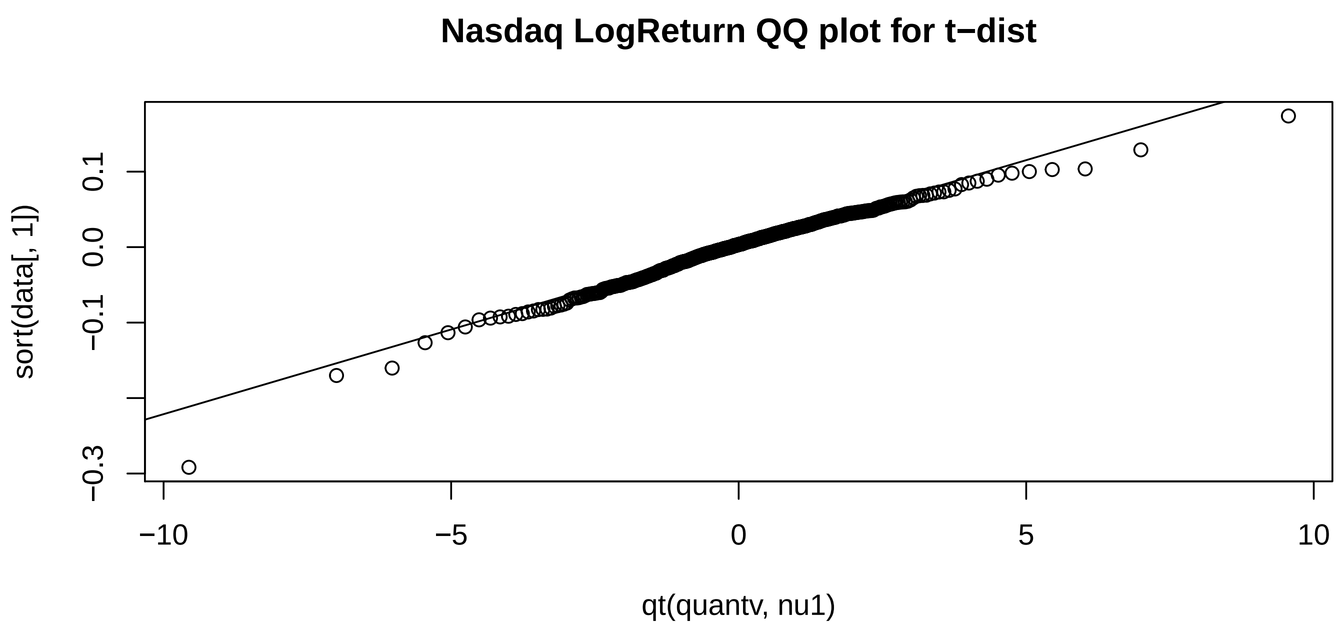
```
est.sp400 = as.numeric(fitdistr(sp400_lreturn, "t")$estimate); est.sp400
```

```
## [1] 0.003084286 0.018450673 3.472334962
```

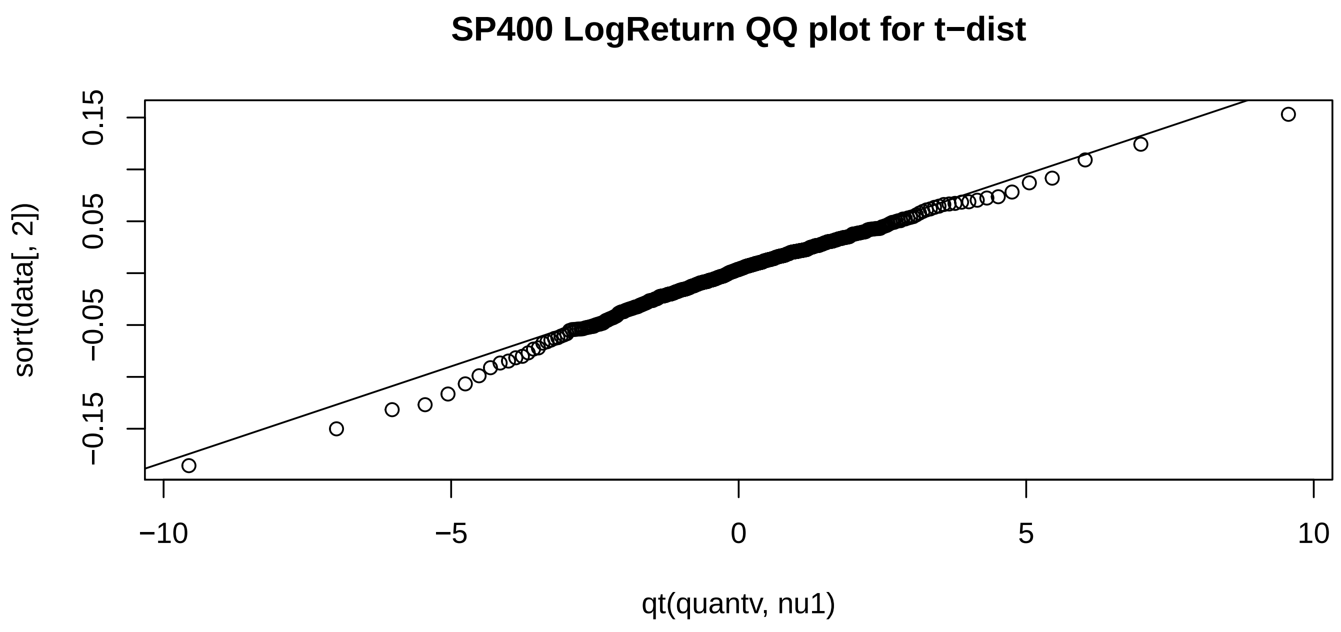
```
# Need to convert to standard deviation for incorporating within "ptsd"
est.nas[2] = est.nas[2] * sqrt(est.nas[3] / (est.nas[3] - 2))
est.sp400[2] = est.sp400[2] * sqrt(est.sp400[3] / (est.sp400[3] - 2))
```

As we see above, the mean for Nasdaq is $est.nas[1] = 0.003350693$ and for SP400 is $est.sp400[1] = 0.003084286$, and the variance for Nasdaq is $est.nas[2] = 0.023018961$ and for SP400 is $est.sp400[2] = 0.018450673$. Also, we can see that the estimatie of degree of freedom separately is $df_{Nasdaq} = est.nas[3] = 3.674359674$, $df_{SP400} = est.sp400[3] = 3.472334962$

```
N = 1000
quantv = (1/N)*seq(.5,N-.5,1)
## QQ_Plot Nasdaq
nu1 = est.nas[3]
qqplot(qt(quantv,nu1),sort(data[,1]),main='Nasdaq LogReturn QQ plot for t-dist')
a = lm(qt(c(.25,.75),nu1)~quantile(data[,1],c(.25,.75)))$coefficients[1]
b = lm(qt(c(.25,.75),nu1)~quantile(data[,1],c(.25,.75)))$coefficients[2]
abline(-a/b,1/b)
```



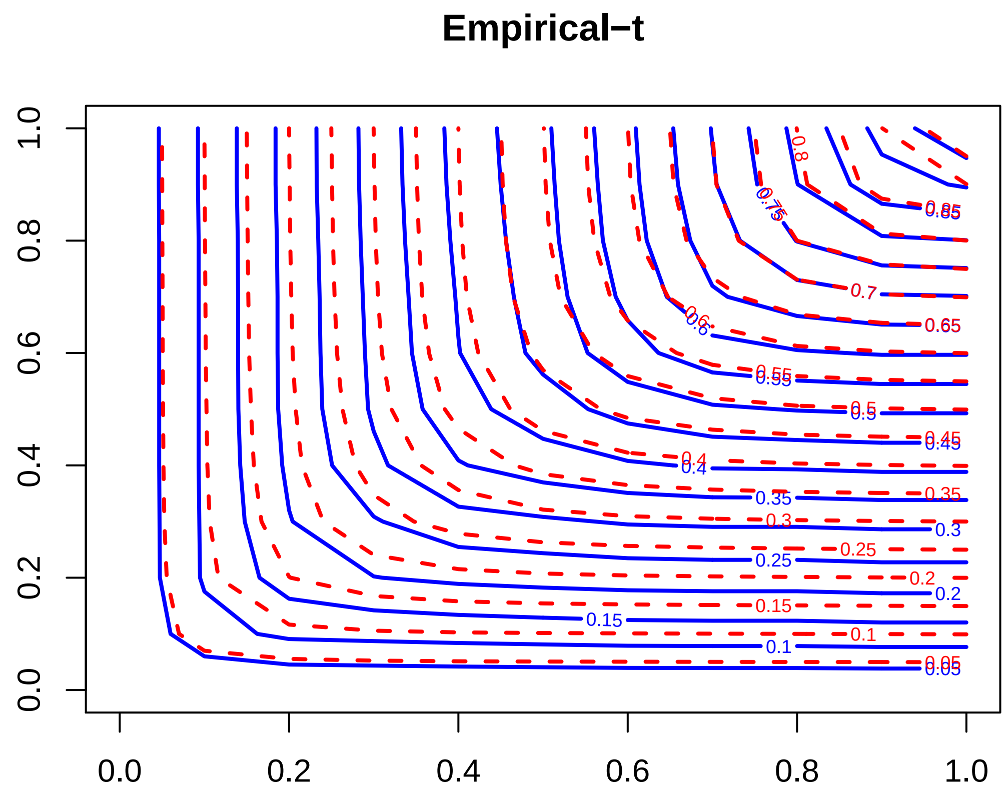
```
## QQ-Plot SP400
nu2 = est.sp400[3]
qqplot(qt(quantv,nu1),sort(data[,2]),main='SP400 LogReturn QQ plot for t-dist')
c = lm(qt(c(.25,.75),nu2)~quantile(data[,2],c(.25,.75)))$coefficients[1]
d = lm(qt(c(.25,.75),nu2)~quantile(data[,2],c(.25,.75)))$coefficients[2]
abline(-c/d,1/d)
```



Seeing that the QQ-plot looks better than previous homework.

Next, we want to transform our fit into the estimated t-cdfs, and fit a t-copula to the data.

```r
nuest = 2.8
fitfinal = cov.trob(cbind(nas_lreturn, sp400_lreturn), nu = 2.8)
mu = fitfinal$center
sigma = fitfinal$cov
est.nas1 = as.numeric(c(mu[1], sqrt(sigma[1,1]), nuest))
est.sp4001 = as.numeric(c(mu[2], sqrt(sigma[2,2]), nuest))
# Need to convert to standard deviation for incorporating within "pstd"
est.nas1[2] = est.nas1[2] * sqrt(est.nas1[3] / (est.nas1[3]-2))
est.sp4001[2] = est.sp4001[2] * sqrt(est.sp4001[3] / (est.sp4001[3]-2))
corr = sigma[1,2]/(sqrt(sigma[1,1])*sqrt(sigma[2,2]))
dat2 = cbind(pstd(nas_lreturn, mean = est.nas1[1], sd = est.nas1[2], nu = est.nas1[3]),
             pstd(sp400_lreturn, mean = est.sp4001[1],
                  sd = est.sp4001[2], nu = est.sp4001[3]))
# empirical cdf
u1 = dat2[,1]
u2 = dat2[,2]
dem = pempiricalCopula(u1,u2)
contour(dem$x, dem$y, dem$z, main = "Empirical-t",
        col = 'blue', lty = 1, lwd = 2, nlevel = 20)
# Generating initial estimate of correlation for t-copula
n = length(nas_lreturn)
ct = tCopula(corr, dim = 2, dispstr = "un", df = 2.8)
utdis = rCopula(100000, ct)
demt = pempiricalCopula(utdis[,1], utdis[,2])
contour(demt$x, demt$y, demt$z, main = "t",
        col = 'red', lty = 2, lwd = 2, add = TRUE, nlevel = 20)
```



```r
# Generating initial estimate of correlation for t-copula
cor_tau = cor(nas_lreturn, sp400_lreturn, method = "spearman"); cor_tau
```

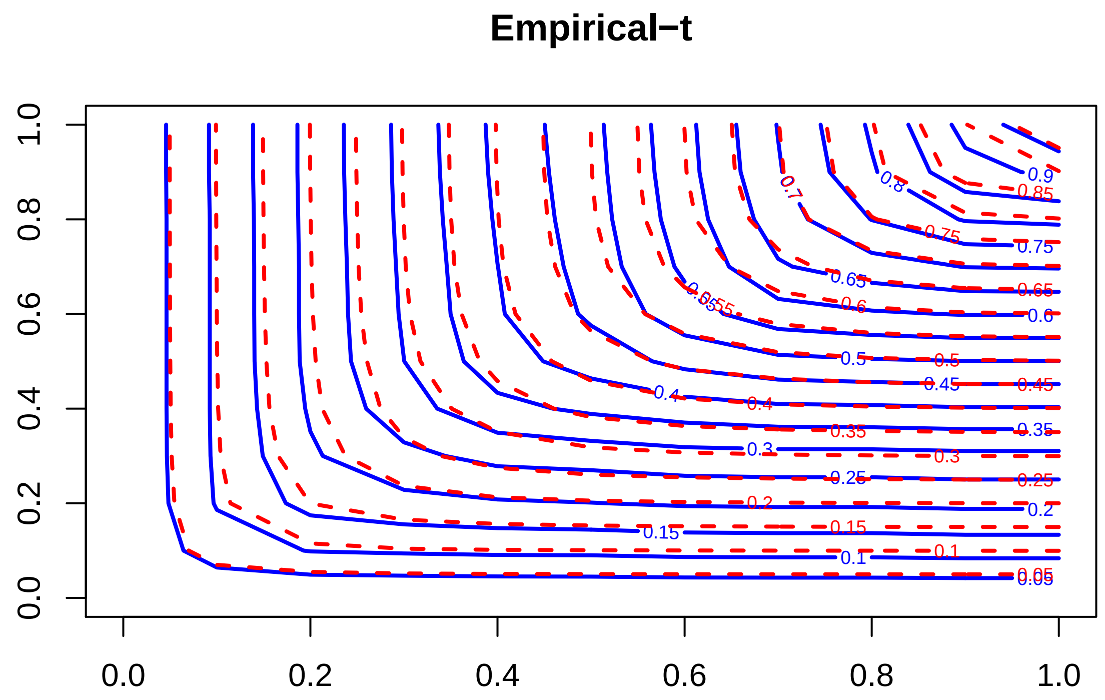
```
## [1] 0.867582
```

```r
data1 = 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]))
cop_t_dim2 = tCopula(cor_tau, dim = 2, dispstr = "un", df = 4)
ft1 = fitCopula(cop_t_dim2, optim.method = "L-BFGS-B", data = data1); ft1
```

```
## Call: fitCopula(copula, data = data, optim.method = "L-BFGS-B")
## Fit based on "maximum pseudo-likelihood" and 1081 2-dimensional observations.
## Copula: tCopula
##  rho.1     df
## 0.8883 3.8299
## The maximized loglikelihood is 849.4
## Optimization converged
```

```r
u1 = data1[,1]; u2 = data1[,2]
dem = pempiricalCopula(u1, u2)
contour(dem$x, dem$y, dem$z, main = "Empirical-t",
        col = 'blue', lty = 1, lwd = 2, nlevel = 20)
ct = tCopula(ft1@estimate[1], dim = 2, dispstr = "un", df = ft1@estimate[2])
```

```
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, we use t-copula to fit our model, and we get the correlation is $\rho = 0.8883$, and we get the degree of freedom of multivariate distribution is $df = 3.8299$. Comparing with the cdf plot of last homework, we can find that this cdf plot fits better on tails than last time. What's more, we can also see what AIC criteria shows of these to models.

```
marginal1 = - 2 * fitdistr(nas_lreturn, "t")$loglik + 2*3
marginal2 = -2 * fitdistr(sp400_lreturn, "t")$loglik + 2*3
AIC_t_copula = -2 * ft1@loglik + 2*2 + marginal1 + marginal2; AIC_t_copula
```

```
## [1] -11065.22
```

```
AIC_t_ple = -2*5549.806 + 2*6; AIC_t_ple
```

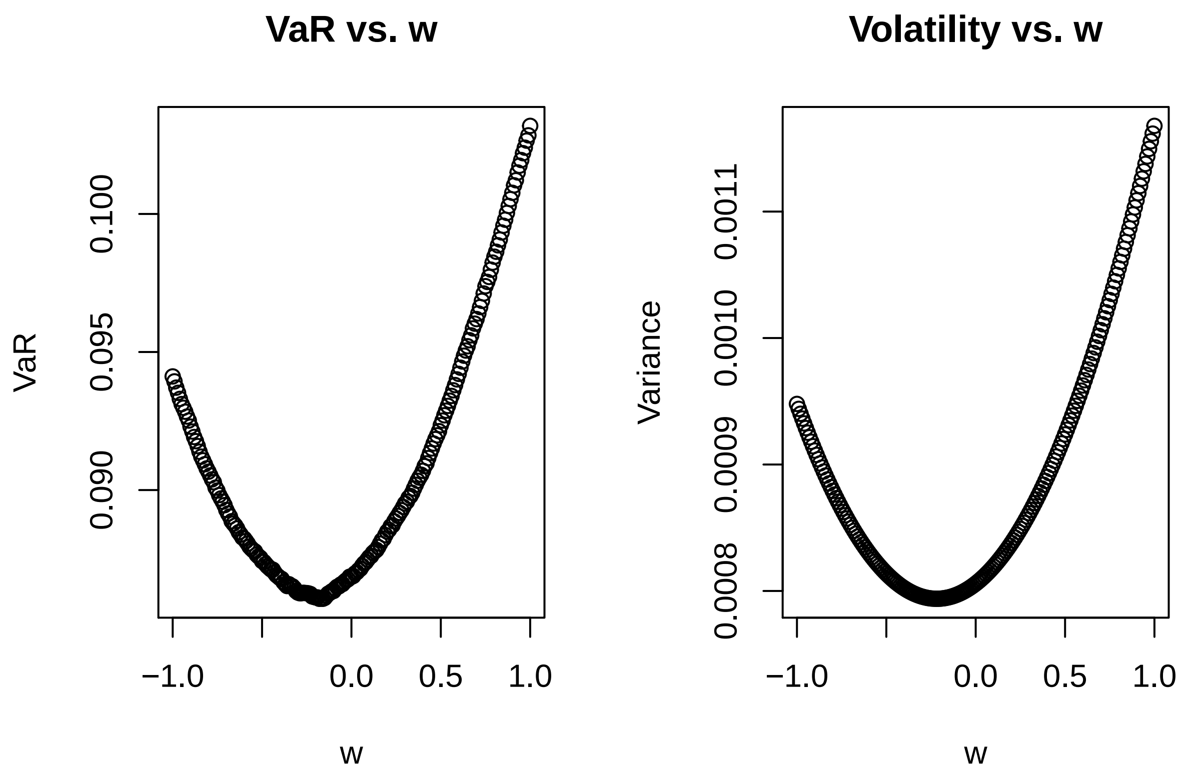
```
## [1] -11087.61
```

Comparing AIC value of these two method, we can see that the cdf plot in this homework seems better than the homework before, but the AIC critrerion shows that last time is better. Thus, I believe that we can choose both method to fit our data.

Question 3

(a) & (b)

```r
set.seed(12345678)
uvsim = rCopula(1000000, ct)
w = seq(-1, 1, 0.01); n = length(w)
VaRv = rep(0, n); variancev = rep(0, n); expectionv = rep(0, n)
data_sim = cbind(qstd(uvsim[,1], mean = est.nas[1], sd = est.nas[2], nu = est.nas[3]),
                 qstd(uvsim[,2], mean = est.sp400[1], sd = est.sp400[2], nu = est.sp400[3]))
for (i in 1:n){
  datat = w[i] *data_sim[,1] + (1 - w[i]) * data_sim[,2];
  VaRv[i] = -(exp(quantile(datat, 0.005))-1)
  variancev[i] = var(datat)
  expectionv[i] = mean(datat)
}
par(mfrow = c(1,2))
plot(w, VaRv, xlab = 'w', ylab = 'VaR', main = "VaR vs. w")
plot(w, variancev, xlab = 'w', ylab = 'Variance', main = "Volatility vs. w")
```



```r
wmin_VaR = w[which.min(VaRv)]
VaR = VaRv[which.min(VaRv)]
paste("The minimum of VaR is ", VaR, " where w =", wmin_VaR)
```

```
## [1] "The minimum of VaR is  0.0860621588251295  where w = -0.16"
```

```r
wmin_vol = w[which.min(variancev)]
vol = variancev[which.min(variancev)]
paste("The minimum of variance is ", vol, "where w =", wmin_vol)
```

```
## [1] "The minimum of variance is  0.000793825659819305 where w = -0.22"
```

```r
set.seed(12345678)
N = length(data_sim[,1]); s1 = 0; n1 = 0
data2 = wmin_VaR *data_sim[,1] + (1 - wmin_VaR) * data_sim[,2]
VaR1 = -(exp(quantile(data2, 0.005))-1)
loss = -(exp(data2) - 1)
below = which(loss > VaR1)
expection1 = -mean(exp(data2[below]) - 1); expection1
```

```
## [1] 0.1211164
```

Thus, if the portfolio is 1, the expected shortfall is 0.1211164.

```r
count = 0;
VaR_1 = -(exp(quantile(data_sim[,1], 0.003))-1)
VaR_2 = -(exp(quantile(data_sim[,2], 0.003))-1)
for (j in 1:N){
  if ((exp(data_sim[j,1])-1) < -VaR_1 & (exp(data_sim[j,2])-1) < -VaR_2){
    count = count + 1
  }
}
prob = count/N
```

Thus the probability that both of the returns are below relative VaRs at q = 0.003 is 0.1877%.