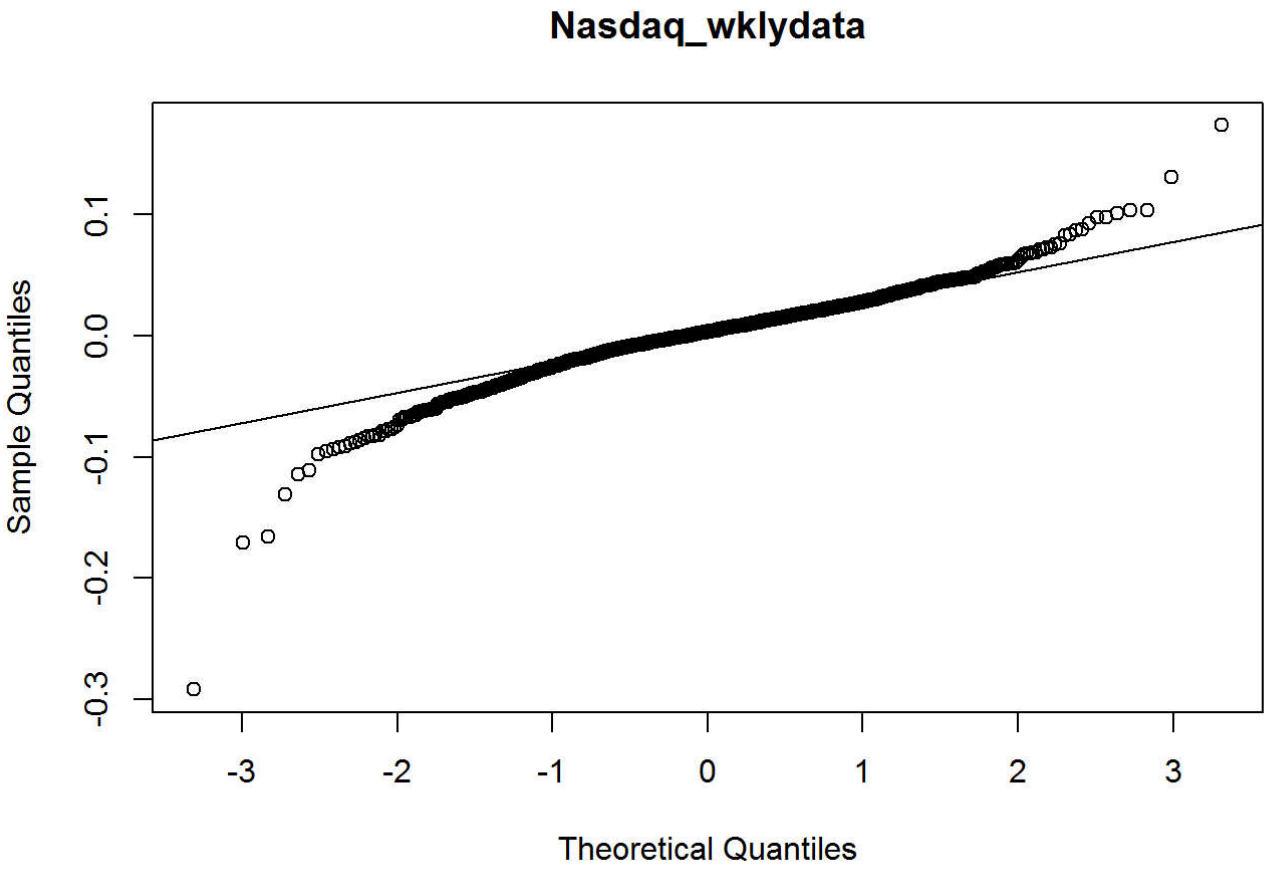
509HW4

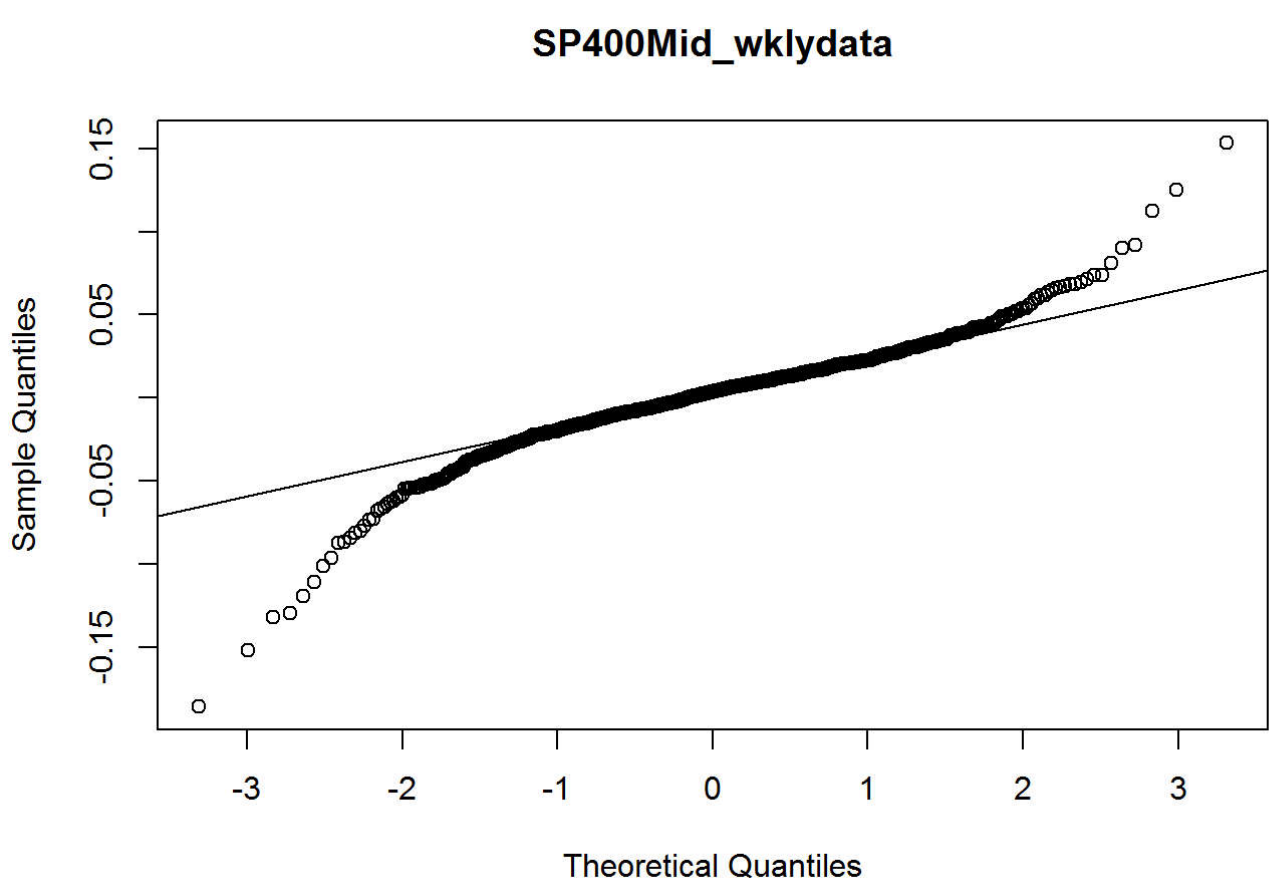
Yuankang Xiong

```
a=read.csv("Nasdaq_wklydata_92-12.csv")
b=read.csv("SP400Mid_wkly_92-12.csv")
NADQadjclose=rev(a[,7])
SP400adjclose=rev(b[,7])
NADQRet=diff(log(NADQadjclose))
SP400Ret=diff(log(SP400adjclose))
```

```
data = cbind(NADQRet,SP400Ret)
qqnorm(data[,1], ylab = "Sample Quantiles",main = "Nasdaq_wklydata")
qqline(data[,1])
```




```
qqnorm(data[,2], ylab = "Sample Quantiles",main = "SP400Mid_wklydata")
qqline(data[,2])
```



*From the QQ-plot we can easily observe that both Nasdaq weekly return and SP 400 weekly return are having heavier tails than normal distribution have.

```
## Warning: package 'copula' was built under R version 3.4.3
```

```
## Warning: package 'fGarch' was built under R version 3.4.3
```

```
## Loading required package: timeDate
```

```
## Warning: package 'timeDate' was built under R version 3.4.3
```

```
## Loading required package: timeSeries
```

```
## Warning: package 'timeSeries' was built under R version 3.4.3
```

```
## Loading required package: fBasics
```

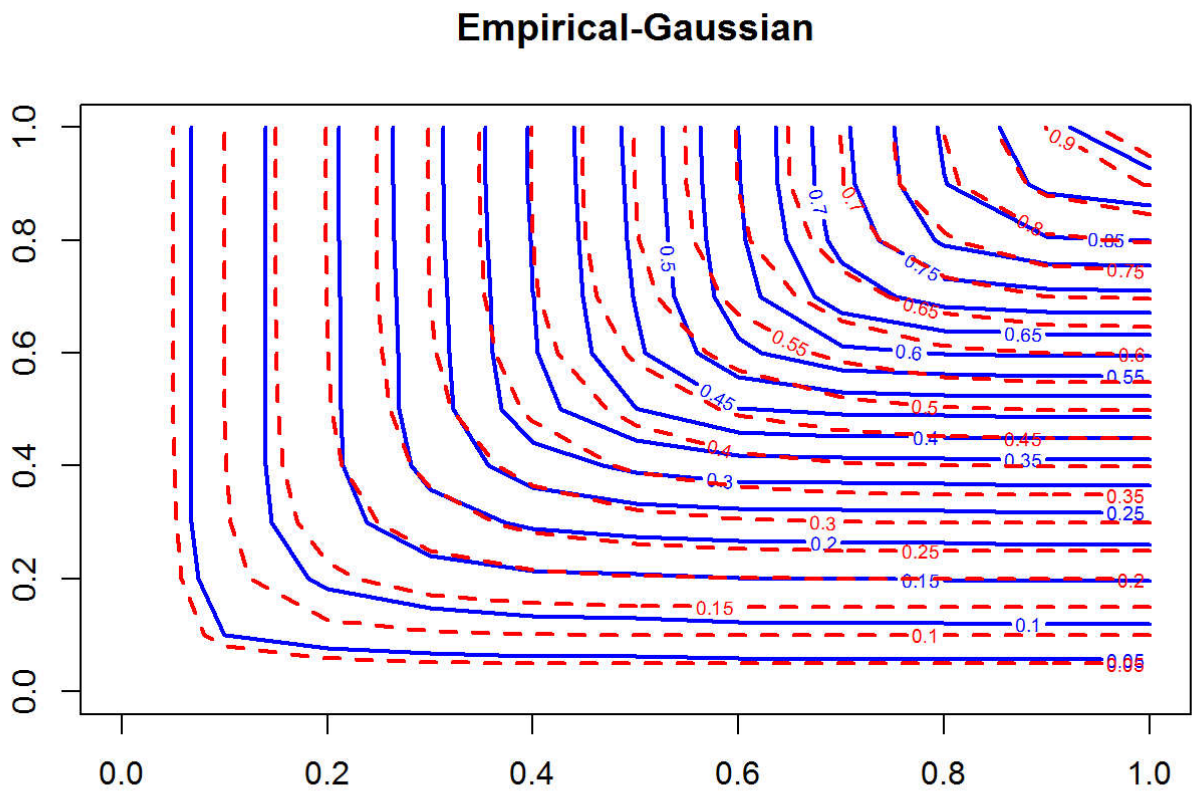
```
## Warning: package 'fBasics' was built under R version 3.4.3
```

```
## Warning: package 'fCopulae' was built under R version 3.4.3
```

```
## Loading required package: fMultivar
```

```
## Warning: package 'fMultivar' was built under R version 3.4.3
```

```
est.nadq = as.numeric(fitdistr(NADQRet,"normal")$estimate)
est.sp400 = as.numeric(fitdistr(SP400Ret,"normal")$estimate)
data1 = cbind(pnorm(NADQRet,mean=est.nadq[1],sd=est.nadq[2]), pnorm(SP400Ret,mean=est.sp400[1],sd=
est.sp400[2]))
u1=data1[,1]
u2=data1[,2]
dem=pempiricalCopula(u1,u2)
contour(dem$x,dem$y,dem$z,main="Empirical-Gaussian",col='blue',lty=1,lwd=2,nlevel=20)

fnorm = fitCopula(data=data1,copula=normalCopula(.3,dim=2),optim.method="BFGS",start=0.5)
cn <- normalCopula(fnorm@estimate[1], 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 empirical and theoretical do not agree with each other a lot especially when the two returns are close to each other(at the turning point of each contour).

```
est.nadq = as.numeric(fitdistr(NADQRet,"t")$estimate)
```

```
## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs
```

```
## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs
```

```
## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs
```

```
## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs
```

```
est.nadq
```

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

```
est.sp400 = as.numeric(fitdistr(SP400Ret,"t")$estimate)
```

```
## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs
```

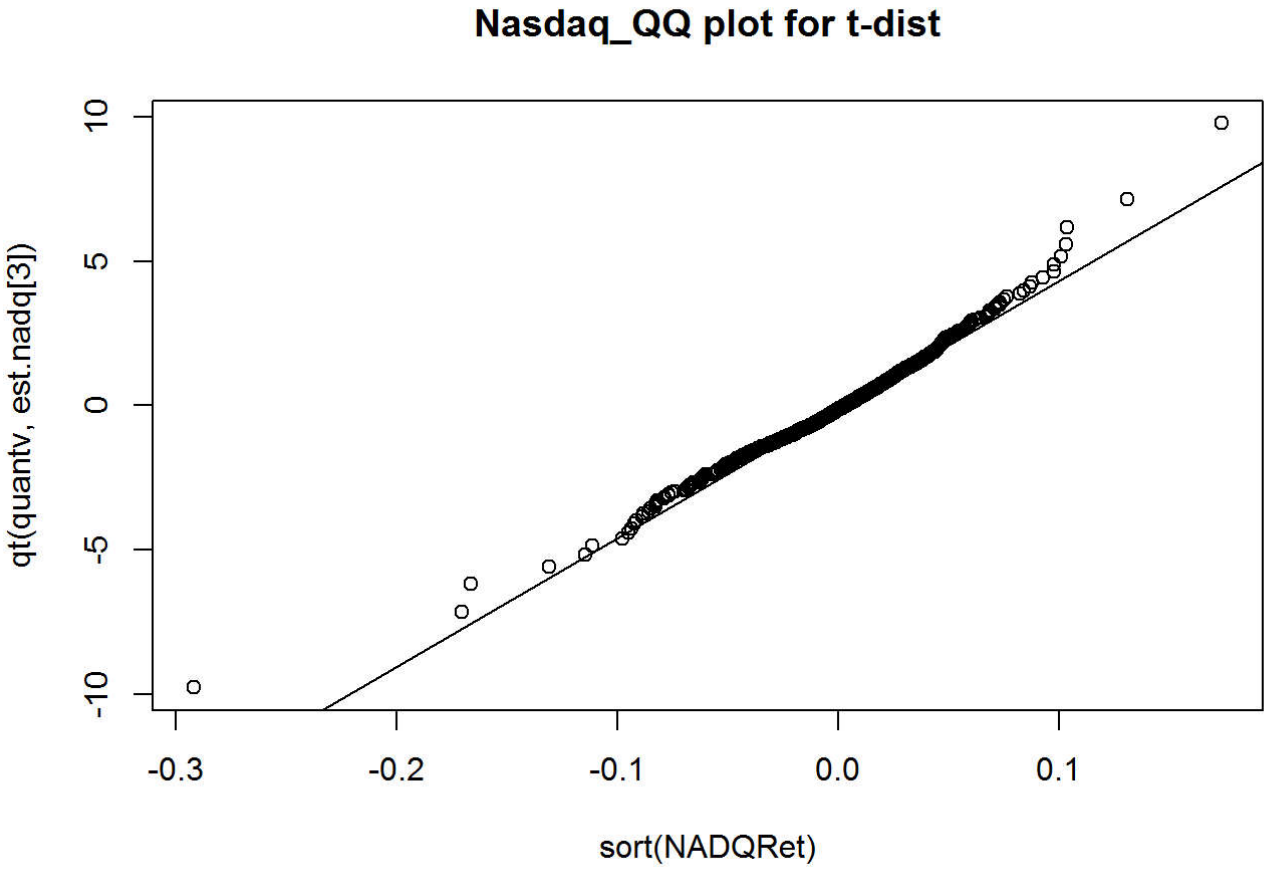
```
## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs
```

```
## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs
```

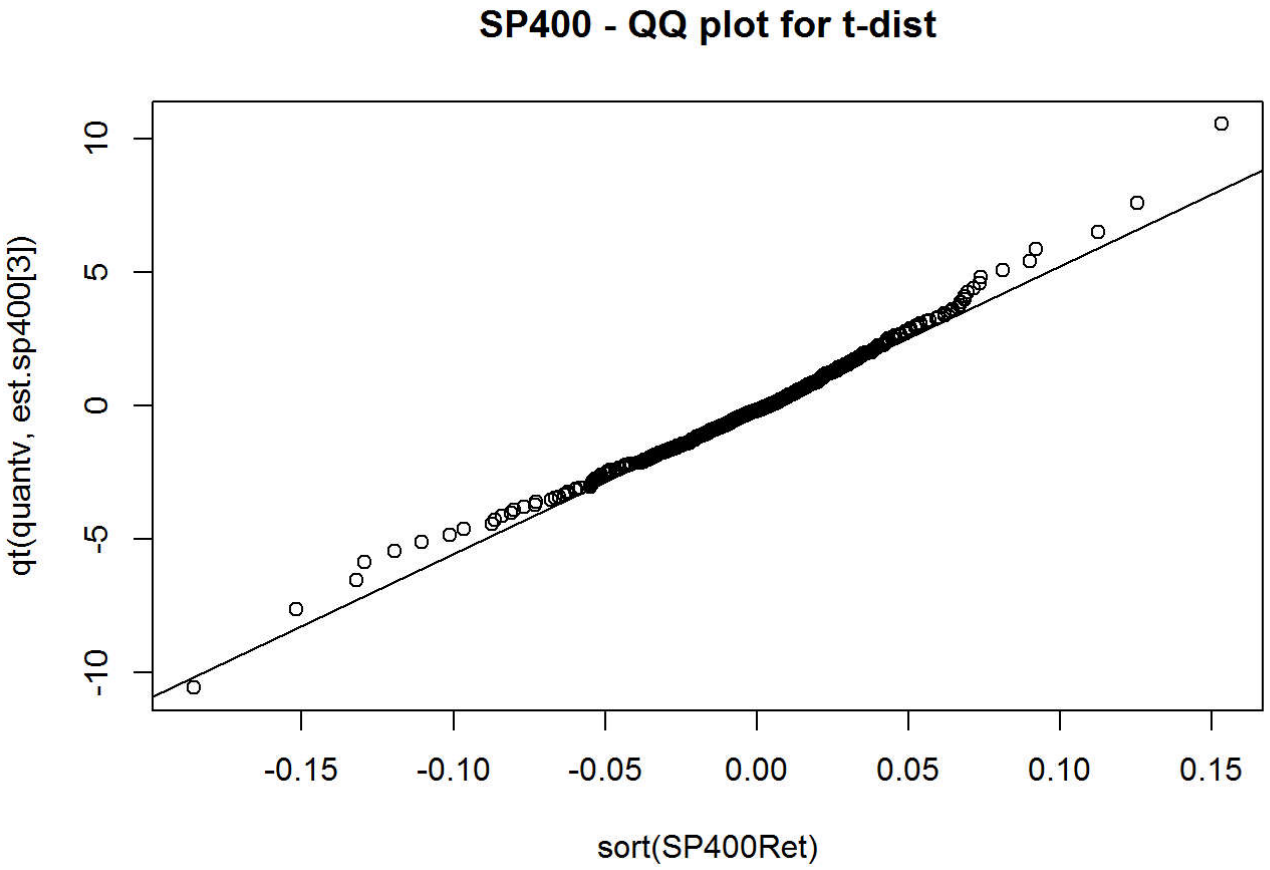
```
est.sp400
```

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

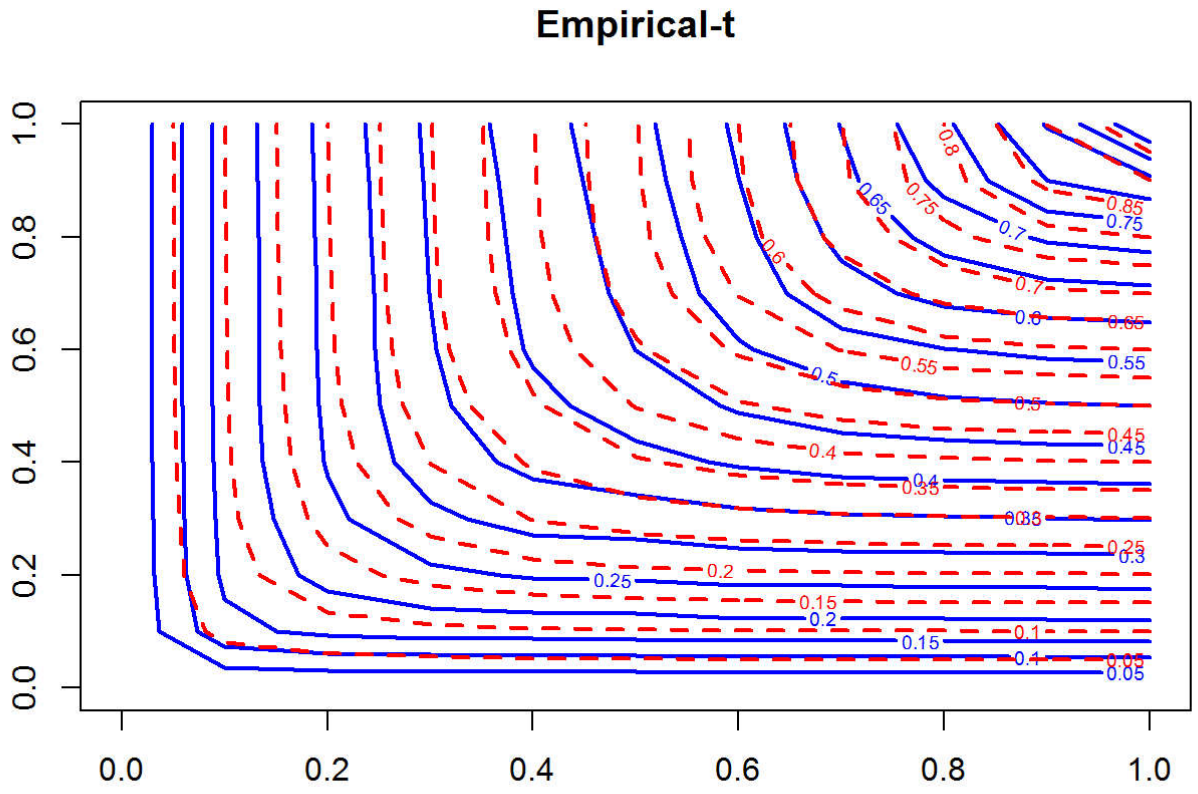
```
N = length(NADQRet)
quantv = (1/N)*seq(.5,N-.5,1)
qqplot(sort(NADQRet),qt(quantv,est.nadq[3]),main='Nasdaq_QQ plot for t-dist')
abline(lm(qt(c(.25,.75),est.nadq[3])~quantile(NADQRet,c(.25,.75))))
```




```
qqplot(sort(SP400Ret),qt(quantv,est.sp400[3]),main='SP400 - QQ plot for t-dist')
abline(lm(qt(c(.25,.75),est.sp400[3])~quantile(SP400Ret,c(.25,.75))))
```



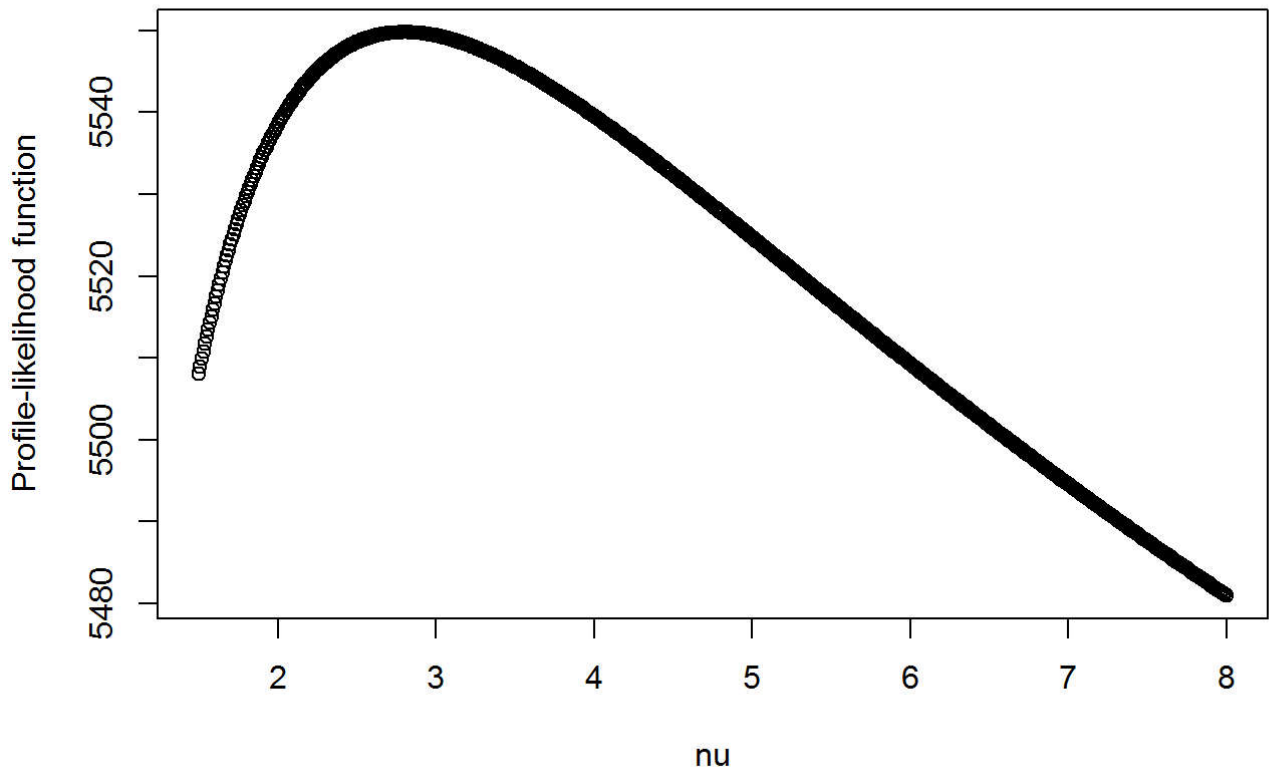

```
data1 = cbind(pstd(NADQRet,mean=est.nadq[1],sd=est.nadq[2],nu=est.nadq[3]),pstd(SP400Ret,mean=est.
sp400[1],sd=est.sp400[2]))
u1=data1[,1]
u2=data1[,2]
dem=pempiricalCopula(u1,u2)
contour(dem$x,dem$y,dem$z,main="Empirical-t",col='blue',lty=1,lwd=2,nlevel=20)

# Generating initial estimate of correlation for t-copula
cor_tau = cor(NADQRet,SP400Ret,method="spearman")
omega = cor_tau
n = length(NADQRet)
cop_t_dim2 = tCopula(omega, dim = 2, dispstr = "un", df = 4)
ft1 = fitCopula(cop_t_dim2, optim.method="L-BFGS-B", data=data1,start=c(omega,5))
ct = tCopula (ft1@estimate[1], dim = 2, dispstr = "un", df = ft1@estimate[2])
utdis = rCopula(100000,ct)
demt2=pempiricalCopula(utdis[,1],utdis[,2])
contour(demt2$x,demt2$y,demt2$z,main="t",col='red',lty=2,lwd=2,add=TRUE,nlevel=20)
```



From the QQ-plot we can easily observe that both Nasdaq weekly return and SP 400 weekly return are more likely to be t-distributed because they are quite on the theoretical line. The empirical and theoretical agree with each other a lot especially when the two returns are close to each other(at the turning point of each contour) but they do not agree with the NADQRET distribution(on the vertical contour when the y have less influence, the distribution of x is misspecified). In general the multivariate t fits better.

**confidence interval

```
data=cbind(NADQRet,SP400Ret)
df = seq(1.5,8,.01)
n = length(df)
loglik_max = rep(0,n)
for(i in 1:n){
fit = cov.trob(data,nu=df[i])
mu = as.vector(fit$center)
sigma = matrix(fit$cov,nrow = 2)
loglik_max[i] =sum(log(dmt(data,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)]
paste("The estimated df is",nuest)
```

```
## [1] "The estimated df is 2.8"
```

```
CI = which(loglik_max>max(loglik_max)-0.5*qchisq(0.95,1))
lbound = df[min(CI)]
ubound = df[max(CI)]
paste("The 95% CI of df is [",lbound,",",ubound,"]")
```

```
## [1] "The 95% CI of df is [ 2.44 , 3.25 ]"
```

c) From the graphs we can see that the multivariate t fits the data better. It not only does better in marginal distribution but also well describe the multivariate distribution. We can also take a look at AIC.

```
AIC_gauss = -2*fnorm@loglik+2*1
AIC_t = -2*ft1@loglik+2*2
c(AIC_gauss,AIC_t)
```

```
## [1] -1457.640 -1843.849
```

**From both AIC and the graph, multivariate t is better.

d) model in a

```
est.nadq = as.numeric(fitdistr(NADQRet,"normal")$estimate)
est.sp400 = as.numeric(fitdistr(SP400Ret,"normal")$estimate)
cn = normalCopula(fnorm@estimate[1],dim = 2, dispstr = "un")
uvsim1 = rCopula(100000,cn)
data_sim1 = cbind(qnorm(uvsim1[,1],mean=est.nadq[1],sd=est.nadq[2]),qnorm(uvsim1[,2],mean=est.sp40
0[1],sd=est.sp400[2]))
datan = 1/2*(data_sim1[,1]+data_sim1[,2])
rVaR = -(exp(quantile(datan,0.001))-1)
VaR = rVaR * 1e7
VaR
```

```
##     0.1%
## 838225.4
```

**model in b

```
est.nadq = as.numeric(fitdistr(NADQRet,"t")$estimate)
```

```
## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs
```

```
## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs
```

```
## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs
```

```
## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs
```

```
est.sp400 = as.numeric(fitdistr(SP400Ret,"t")$estimate)
```

```
## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs

## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs
```

```
## Warning in dt((x - m)/s, df, log = TRUE): 产生了NaNs
```

```
## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs

## Warning in log(s): 产生了NaNs
```

```
ct = tCopula(ft1@estimate[1],dim=2,dispstr="un",df=ft1@estimate[2])
uvsim2 = rCopula(100000,ct)
data_sim2 = cbind(qstd(uvsim2[,1],mean=est.nadq[1],sd=est.nadq[2],nu=est.nadq[3]),
qstd(uvsim2[,2],mean=est.sp400[1],sd=est.sp400[2],nu=est.sp400[3]))
datat = 1/2*(data_sim2[,1]+data_sim2[,2])
rVaR = -(exp(quantile(datat,0.001))-1)
VaR = rVaR * 1e7
VaR
```

```
##    0.1%
## 1022471
```

e) model in a)

```
w = seq(0,1,.01)
cn = normalCopula(fnorm@estimate[1],dim = 2, dispstr = "un")
uvsim = rCopula(100000,cn)
data_sim = cbind(qnorm(uvsim1[,1],mean=est.nadq[1],sd=est.nadq[2]),qnorm(uvsim1[,2],mean=est.sp400
[1],sd=est.sp400[2]))
n = length(w)
exp_ret = rep(0,n)
vol = rep(0,n)
rVaR = rep(0,n)
for(i in 1:n){
datan = w[i]*data_sim[,1]+(1-w[i])*data_sim[,2]
exp_ret[i] = mean(exp(datan) - 1)
vol[i] = var(exp(datan)-1)
rVaR[i] = -(exp(quantile(datan,0.002))-1)
}
par(mfrow=c(2,2))
plot(w,exp_ret,xlab='w',ylab='exp_ret',main="Expected Return v.s. w")
plot(w,vol,xlab='w',ylab='vol',main="Volatility v.s. w")
plot(w,rVaR,xlab='w',ylab='rVaR',main="rVaR v.s. w")
w1 = w[which.max(exp_ret)]
Exp_Return = exp_ret[which.max(exp_ret)]
paste("The maximum expected return is", Exp_Return, ", the corresponding w is", w1)
```

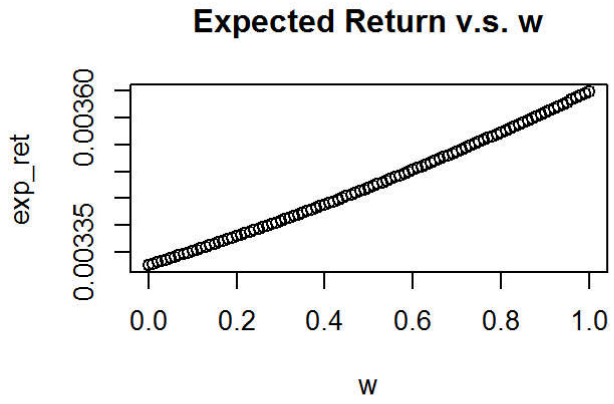
```
## [1] "The maximum expected return is 0.00364824790753866 , the corresponding w is 1"
```

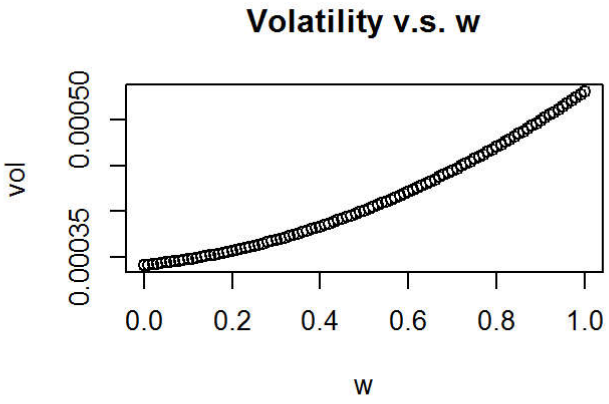
```
w2 = w[which.min(vol)]
Vol = vol[which.min(vol)]
paste("The minimum volatility is", Vol, ", the corresponding w is", w2)
```

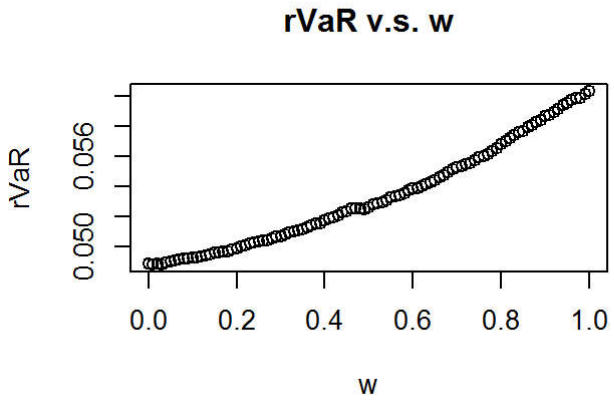
```
## [1] "The minimum volatility is 0.000341105263890725 , the corresponding w is 0"
```

```
w3 = w[which.min(rVaR)]
VaR = 1e7 * rVaR[which.min(rVaR)]
paste("The minimum VaR at q = .002 is", VaR, ", the corresponding w is", w3)
```

```
## [1] "The minimum VaR at q = .002 is 488222.559257068 , the corresponding w is 0.03"
```







**model in b)

```
ct = tCopula(ft1@estimate[1],dim=2,dispstr="un",df=ft1@estimate[2])
uvsim = rCopula(100000,ct)
data_sim= cbind(qstd(uvsim2[,1],mean=est.nadq[1],sd=est.nadq[2],nu=est.nadq[3]),
qstd(uvsim2[,2],mean=est.sp400[1],sd=est.sp400[2],nu=est.sp400[3]))
n = length(w)
exp_ret = rep(0,n)
vol = rep(0,n)
rVaR = rep(0,n)
for(i in 1:n){
datan = w[i]*data_sim[,1]+(1-w[i])*data_sim[,2]
exp_ret[i] = mean(exp(datan) - 1)
vol[i] = var(exp(datan)-1)
rVaR[i] = -(exp(quantile(datan,0.002))-1)
}
par(mfrow=c(2,2))
plot(w,exp_ret,xlab='w',ylab='exp_ret',main="Expected Return v.s. w")
plot(w,vol,xlab='w',ylab='vol',main="Volatility v.s. w")
plot(w,rVaR,xlab='w',ylab='rVaR',main="rVaR v.s. w")
w1 = w[which.max(exp_ret)]
Exp_Return = exp_ret[which.max(exp_ret)]
paste("The maximum expected return is", Exp_Return, ", the corresponding w is", w1)
```

```
## [1] "The maximum expected return is 0.00366113920340208 , the corresponding w is 1"
```

```
w2 = w[which.min(vol)]
Vol = vol[which.min(vol)]
paste("The minimum volatility is", Vol, ", the corresponding w is", w2)
```

```
## [1] "The minimum volatility is 0.00033908183886282 , the corresponding w is 0"
```

```
w3 = w[which.min(rVaR)]
VaR = 1e7 * rVaR[which.min(rVaR)]
paste("The minimum VaR at q = .002 is", VaR, ", the corresponding w is", w3)
```

```
## [1] "The minimum VaR at q = .002 is 770666.041911611 , the corresponding w is 0.08"
```