

CODE

```
data<-read.csv("dataset.csv")
summary(data)

#####
#### 2. Descriptive Statistics ####
#####

library("PerformanceAnalytics")
asset = data[,-1]
netreturn = asset[2:(nrow(asset)),]/asset[1:(nrow(asset)-1),]-1
date1 = data$Date[-1]
data3 = round(netreturn,3)
data2 = cbind(date1, data3)
dim(asset)
dim(netreturn)
rf=0.07*10^(-2)

AMD<-netreturn[,1]
AAPL<-netreturn[,2]
BAC<-netreturn[,3]
COKE<-netreturn[,4]
FCX<-netreturn[,5]
F<-netreturn[,6]
GE<-netreturn[,7]
MDT<-netreturn[,8]
MRO<-netreturn[,9]
PFE<-netreturn[,10]
SIRI<-netreturn[,11]
SBUX<-netreturn[,12]
X<-netreturn[,13]
VALE<-netreturn[,14]
VZ<-netreturn[,15]
sp500<-netreturn[,ncol(netreturn)]

# mean
apply(netreturn,2,mean)
# std
apply(netreturn,2,sd)
# skewness
apply(netreturn,2,skewness)
# kurtosis
apply(netreturn,2,kurtosis)
```

```

# annualized statistics
round(apply(netreturn,2,mean)*12,2)
round(apply(netreturn,2,sd)*sqrt(12),2)

# beta
betai<-rep(0,ncol(netreturn))
for(i in 1:ncol(netreturn)){
  # betai[i]<-((netreturn[,i]-rf)/(netreturn[,ncol(netreturn)]-rf))
  betai[i]<-cov(netreturn[,i],netreturn[,ncol(netreturn)])/var(netreturn[,ncol(netreturn)])
}
betai

# plot monthly prices
plot(asset[,16],type="l",lty=23,ylim=c(min(asset[,16]),max(asset[,16])),xlab="Time Series",ylab="S&P500
Price",main="Monthly Closing Price for S&P 500")

color1<-c("red","blue","green","darkgrey","orange","pink","black")
plot(asset[,1],type="l",col="red",ylim=c(0,220),main="Monthly Closing Price for 7
Assets",ylab="Price",xlab="Time Series")
lines(asset[,2],col="blue")
lines(asset[,3],col="green")
lines(asset[,4],col="darkgrey")
lines(asset[,5],col="orange")
lines(asset[,6],col="pink")
lines(asset[,7],col="black")
legend("topleft",c("AMD","AAPL","BAC","COKE","FCX","F","GE"),lty=1,col=color1)

color2<-c("red","blue","green","grey","orange","pink","lightblue","darkgreen")
plot(asset[,8],type="l",col="red",ylim=c(0,100),main="Monthly Closing Price for 8
Assets",ylab="Price",xlab="Time Series")
lines(asset[,9],col="blue")
lines(asset[,10],col="green")
lines(asset[,11],col="grey")
lines(asset[,12],col="orange")
lines(asset[,13],col="pink")
lines(asset[,14],col="lightblue")
lines(asset[,15],col="darkgreen")
legend("topleft",c("MDT","MRO","PFE","SIRI","SBUX","X","VALE","VZ"),lty=1,col=color2)

# plot monthly returns
# par(mfrow=c(1,3))
color<-c("red","blue")

plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="AMD Monthly Returns")

```

```
lines(AMD,col="blue",lty=23)
legend("bottomright",c("sp500", "AMD"),lty=c(1,23),col=color)
```

```
plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="AAPL Monthly Returns")
lines(AAPL,col="blue")
legend("bottomright",c("sp500", "AAPL"),lty=c(1,23),col=color)
```

```
plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="BAC Monthly Returns")
lines(BAC,col="blue")
legend("bottomright",c("sp500", "BAC"),lty=c(1,23),col=color)
```

```
plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="COKE Monthly Returns")
lines(COKE,col="blue")
legend("bottomright",c("sp500", "COKE"),lty=c(1,23),col=color)
```

```
plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="FCX Monthly Returns")
lines(FCX,col="blue")
legend("bottomright",c("sp500", "FCX"),lty=c(1,23),col=color)
```

```
plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="F Monthly Returns")
lines(F,col="blue")
legend("bottomright",c("sp500", "F"),lty=c(1,23),col=color)
```

```
plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="GE Monthly Returns")
lines(GE,col="blue")
legend("bottomright",c("sp500", "GE"),lty=c(1,23),col=color)
```

```
plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="MDT Monthly Returns")
lines(MDT,col="blue")
legend("bottomright",c("sp500", "MDT"),lty=c(1,23),col=color)
```

```
plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="MRO Monthly Returns")
lines(MRO,col="blue")
legend("bottomright",c("sp500", "MRO"),lty=c(1,23),col=color)
```

```
plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="PFE Monthly Returns")
lines(PFE,col="blue")
legend("bottomright",c("sp500", "PFE"),lty=c(1,23),col=color)
```

```
plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="SIRI Monthly Returns")
lines(SIRI,col="blue")
legend("bottomright",c("sp500", "SIRI"),lty=c(1,23),col=color)
```

```
plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="SBUX Monthly Returns")
```

```
lines(SBUX,col="blue")
legend("bottomright",c("sp500","SBUX"),lty=c(1,23),col=color)
```

```
plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="X Monthly Returns")
lines(X,col="blue")
legend("bottomright",c("sp500","X"),lty=c(1,23),col=color)
```

```
plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="VALE Monthly Returns")
lines(VALE,col="blue")
legend("bottomright",c("sp500","VALE"),lty=c(1,23),col=color)
```

```
plot(sp500,type='l',lty=1,col="red",ylim=c(min(netreturn),max(netreturn)),main="VZ Monthly Returns")
lines(VZ,col="blue")
legend("bottomright",c("sp500","VZ"),lty=c(1,23),col=color)
```

```
# Equity Curve
EquityCurve<-cumprod(1+netreturn)
```

```
par(mfrow=c(1,1))
colorEC<-c("red","blue","green","darkgrey","orange","pink","darkgreen","lightblue","purple","darkblue","magenta",
,"slateblue4","red3","grey","green3")
name<-c("SP500","AMD","AAPL","BAC","COKE","FCX","F","GE","MDT","MRO","PFE","SIRI","SBUX","X",
,"VALE","VZ")
plot(EquityCurve[,ncol(EquityCurve)],type="l",lwd=2.5,lty=2,main="Equity
Curve",ylim=c(min(EquityCurve),max(EquityCurve)),ylab="Return of 1 dollar", xlab="Time Series")
for(i in 1:(ncol(EquityCurve)-1)){
  lines(EquityCurve[,i],col=colorEC[i])
}
legend("topleft",name,lty=c(2,rep(1,15)),col=c("black",colorEC),cex=0.5)
```

```
# separate plot of equity curve
par(mfrow = c(2,3))
colorEC2<-c("red","blue")
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="AMD",ylim=c(min(EquityCurve[,16],EquityCurve[,1]),max(
EquityCurve[,16],EquityCurve[,1])),ylab="Return of 1 dollar", xlab="Time Series",col="red")
lines(EquityCurve[,1],col="blue")
legend("topleft",c("sp500","AMD"),lty=c(2,1),col=colorEC2)
```

```
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="AAPL",ylim=c(min(EquityCurve[,16],EquityCurve[,2]),max(
EquityCurve[,16],EquityCurve[,2])),ylab="Return of 1 dollar", xlab="Time Series",col="red")
lines(EquityCurve[,2],col="blue")
legend("topleft",c("sp500","AAPL"),lty=c(2,1),col=colorEC2)
```

```
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="BAC",ylim=c(min(EquityCurve[,16],EquityCurve[,3]),max(EquityCurve[,16],EquityCurve[,3])),ylab="Return of 1 dollar", xlab="Time Series",col="red")
lines(EquityCurve[,3],col="blue")
legend("topleft",c("sp500","AMD"),lty=c(2,1),col=colorEC2)
```

```
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="COKE",ylim=c(min(EquityCurve[,16],EquityCurve[,4]),max(EquityCurve[,16],EquityCurve[,4])),ylab="Return of 1 dollar", xlab="Time Series",col="red")
lines(EquityCurve[,4],col="blue")
legend("topleft",c("sp500","COKE"),lty=c(2,1),col=colorEC2)
```

```
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="FCX",ylim=c(min(EquityCurve[,16],EquityCurve[,5]),max(EquityCurve[,16]+1,EquityCurve[,5]+1))),ylab="Return of 1 dollar", xlab="Time Series",col="red")
lines(EquityCurve[,5],col="blue")
legend("topleft",c("sp500","FCX"),lty=c(2,1),col=colorEC2)
```

```
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="F",ylim=c(min(EquityCurve[,16],EquityCurve[,6]),max(EquityCurve[,16],EquityCurve[,6])),ylab="Return of 1 dollar", xlab="Time Series",col="red")
lines(EquityCurve[,6],col="blue")
legend("topleft",c("sp500","F"),lty=c(2,1),col=colorEC2)
```

```
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="GE",ylim=c(min(EquityCurve[,16],EquityCurve[,7]),max(EquityCurve[,16],EquityCurve[,7])),ylab="Return of 1 dollar", xlab="Time Series",col="red")
lines(EquityCurve[,7],col="blue")
legend("topleft",c("sp500","GE"),lty=c(2,1),col=colorEC2)
```

```
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="MDT",ylim=c(min(EquityCurve[,16],EquityCurve[,8]),max(EquityCurve[,16],EquityCurve[,8])),ylab="Return of 1 dollar", xlab="Time Series",col="red")
lines(EquityCurve[,8],col="blue")
legend("topleft",c("sp500","MDT"),lty=c(2,1),col=colorEC2)
```

```
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="MRO",ylim=c(min(EquityCurve[,16],EquityCurve[,9]),max(EquityCurve[,16],EquityCurve[,9])),ylab="Return of 1 dollar", xlab="Time Series",col="red")
lines(EquityCurve[,9],col="blue")
legend("topleft",c("sp500","MRO"),lty=c(2,1),col=colorEC2)
```

```
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="PFE",ylim=c(min(EquityCurve[,16],EquityCurve[,10]),max(EquityCurve[,16],EquityCurve[,10])),ylab="Return of 1 dollar", xlab="Time Series",col="red")
lines(EquityCurve[,10],col="blue")
legend("topleft",c("sp500","PFE"),lty=c(2,1),col=colorEC2)
```

```
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="SIRI",ylim=c(min(EquityCurve[,16],EquityCurve[,11]),max(EquityCurve[,16],EquityCurve[,11])),ylab="Return of 1 dollar", xlab="Time Series",col="red")
lines(EquityCurve[,11],col="blue")
```

```
legend("topleft",c("sp500","SIRI"),lty=c(2,1),col=colorEC2)
```

```
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="SBUX",ylim=c(min(EquityCurve[,16],EquityCurve[,12]),max(EquityCurve[,16],EquityCurve[,12])),ylab="Return of 1 dollar", xlab="Time Series",col="red")
```

```
lines(EquityCurve[,12],col="blue")
```

```
legend("topleft",c("sp500","SBUX"),lty=c(2,1),col=colorEC2)
```

```
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="X",ylim=c(min(EquityCurve[,16],EquityCurve[,13]),max(EquityCurve[,16],EquityCurve[,13])),ylab="Return of 1 dollar", xlab="Time Series",col="red")
```

```
lines(EquityCurve[,13],col="blue")
```

```
legend("topleft",c("sp500","X"),lty=c(2,1),col=colorEC2)
```

```
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="VALE",ylim=c(min(EquityCurve[,16],EquityCurve[,14]),max(EquityCurve[,16],EquityCurve[,14])),ylab="Return of 1 dollar", xlab="Time Series",col="red")
```

```
lines(EquityCurve[,14],col="blue")
```

```
legend("topleft",c("sp500","VALE"),lty=c(2,1),col=colorEC2)
```

```
plot(EquityCurve[,16],type="l",lwd=2.5,lty=2,main="VZ",ylim=c(min(EquityCurve[,16],EquityCurve[,15]),max(EquityCurve[,16],EquityCurve[,15])),ylab="Return of 1 dollar", xlab="Time Series",col="red")
```

```
lines(EquityCurve[,15],col="blue")
```

```
legend("topleft",c("sp500","VZ"),lty=c(2,1),col=colorEC2)
```

```
#rainbow color
```

```
# pie(rep(1, times = 1000), labels = "", col = rainbow(1000), border = rainbow(1000),main = "rainbow1000")
```

```
# Histogram
```

```
par(mfrow=c(2,3))
```

```
hist(AMD,main="AMD")
```

```
hist(AAPL,main="AAPL")
```

```
hist(BAC,main="BAC")
```

```
hist(COKE,main="COKE")
```

```
hist(FCX,main="FCX")
```

```
hist(F,main="F")
```

```
hist(GE,main="GE")
```

```
hist(MDT,main="MDT")
```

```
hist(MRO,main="MRO")
```

```
hist(PFE,main="PFE")
```

```
hist(SIRI,main="SIRI")
```

```
hist(SBUX,main="SBUX")
```

```
hist(X,main="X")
hist(VALE,main="VALE")
hist(VZ,main="VZ")
```

```
# Boxplot
library("plotly")
library("ggplot2")
```

```
plot_ly(netreturn,y=AMD,type="box",name="AMD") %>%
  add_trace(y=AAPL,type="box",name="AAPL") %>%
  add_trace(y=BAC,type="box",name="BAC") %>%
  add_trace(y=COKE,type="box",name="COKE") %>%
  add_trace(y=FCX,type="box",name="FCX") %>%
  add_trace(y=F,type="box",name="F") %>%
  add_trace(y=GE,type="box",name="GE") %>%
  add_trace(y=MDT,type="box",name="MDT") %>%
  add_trace(y=MRO,type="box",name="MRO") %>%
  add_trace(y=PFE,type="box",name="PFE") %>%
  add_trace(y=SIRI,type="box",name="SIRI") %>%
  add_trace(y=SBUX,type="box",name="SBUX") %>%
  add_trace(y=X,type="box",name="X") %>%
  add_trace(y=VALE,type="box",name="VALE") %>%
  add_trace(y=VZ,type="box",name="VZ") %>%
  layout(title="Boxplot of assets' returns")
```

```
# plot_ly(netreturn,y=AMD,type="box",name="AMD") %>%
#   add_trace(y=AAPL,type="box",name="AAPL") %>%
#   add_trace(y=BAC,type="box",name="BAC") %>%
#   add_trace(y=COKE,type="box",name="COKE") %>%
#   add_trace(y=FCX,type="box",name="FCX") %>%
#   layout(title="Boxplot of five assets' returns")
#
# plot_ly(netreturn,y=F,type="box",name="F") %>%
#   add_trace(y=GE,type="box",name="GE") %>%
#   add_trace(y=MDT,type="box",name="MDT") %>%
#   add_trace(y=MRO,type="box",name="MRO") %>%
#   add_trace(y=PFE,type="box",name="PFE") %>%
#   layout(title="Boxplot of five assets' returns")
#
# plot_ly(netreturn,y=SIRI,type="box",name="SIRI") %>%
#   add_trace(y=SBUX,type="box",name="SBUX") %>%
```

```

#   add_trace(y=X,type="box",name="X") %>%
#   add_trace(y=VALE,type="box",name="VALE") %>%
#   add_trace(y=VZ,type="box",name="VZ") %>%
#   layout(title="Boxplot of five assets' returns")

# Outlier test
boxplot(netreturn)

outlier = boxplot.stats(data2[,2])$out
for (j in 1:length(outlier)) {
  print(subset(data2, data2[,2] == outlier[j], select = c(1,2)))
}

outlier = boxplot.stats(data2[,3])$out
for (j in 1:length(outlier)) {
  print(subset(data2, data2[,3] == outlier[j], select = c(1,3)))
}

outlier = boxplot.stats(data2[,4])$out
for (j in 1:length(outlier)) {
  print(subset(data2, data2[,4] == outlier[j], select = c(1,4)))
}

outlier = boxplot.stats(data2[,5])$out
for (j in 1:length(outlier)) {
  print(subset(data2, data2[,5] == outlier[j], select = c(1,5)))
}

outlier = boxplot.stats(data2[,6])$out
for (j in 1:length(outlier)) {
  print(subset(data2, data2[,6] == outlier[j], select = c(1,6)))
}

outlier = boxplot.stats(data2[,7])$out
for (j in 1:length(outlier)) {
  print(subset(data2, data2[,7] == outlier[j], select = c(1,7)))
}

outlier = boxplot.stats(data2[,8])$out
for (j in 1:length(outlier)) {
  print(subset(data2, data2[,8] == outlier[j], select = c(1,8)))
}

outlier = boxplot.stats(data2[,9])$out

```



```

for (j in 1:length(outlier)) {
  print(subset(data2, data2[,9] == outlier[j], select = c(1,9)))
}

outlier = boxplot.stats(data2[,10])$out
for (j in 1:length(outlier)) {
  print(subset(data2, data2[,10] == outlier[j], select = c(1,10)))
}

outlier = boxplot.stats(data2[,11])$out
for (j in 1:length(outlier)) {
  print(subset(data2, data2[,11] == outlier[j], select = c(1,11)))
}

outlier = boxplot.stats(data2[,12])$out
for (j in 1:length(outlier)) {
  print(subset(data2, data2[,12] == outlier[j], select = c(1,12)))
}

outlier = boxplot.stats(data2[,13])$out
for (j in 1:length(outlier)) {
  print(subset(data2, data2[,13] == outlier[j], select = c(1,13)))
}

outlier = boxplot.stats(data2[,14])$out
for (j in 1:length(outlier)) {
  print(subset(data2, data2[,14] == outlier[j], select = c(1,14)))
}

outlier = boxplot.stats(data2[,15])$out
for (j in 1:length(outlier)) {
  print(subset(data2, data2[,15] == outlier[j], select = c(1,15)))
}

outlier = boxplot.stats(data2[,16])$out
for (j in 1:length(outlier)) {
  print(subset(data2, data2[,16] == outlier[j], select = c(1,16)))
}

# qqplot
library("fBasics")

```

```
par(mfrow=c(2,3))
qqnormPlot(AMD,title=FALSE)
title(main="AMD",ylab=NULL)
```

```
qqnormPlot(AAPL,title=FALSE)
title(main="AAPL",ylab=NULL)
```

```
qqnormPlot(BAC,title=FALSE)
title(main="BAC",ylab=NULL)
```

```
qqnormPlot(COKE,title=FALSE)
title(main="COKE",ylab=NULL)
```

```
qqnormPlot(FCX,title=FALSE)
title(main="FCX",ylab=NULL)
```

```
qqnormPlot(F,title=FALSE)
title(main="F",ylab=NULL)
```

```
qqnormPlot(GE,title=FALSE)
title(main="GE",ylab=NULL)
```

```
qqnormPlot(MDT,title=FALSE)
title(main="MDT",ylab=NULL)
```

```
qqnormPlot(MRO,title=FALSE)
title(main="MRO",ylab=NULL)
```

```
qqnormPlot(PFE,title=FALSE)
title(main="PFE",ylab=NULL)
```

```
qqnormPlot(SIRI,title=FALSE)
title(main="SIRI",ylab=NULL)
```

```
qqnormPlot(SBUX,title=FALSE)
title(main="SBUX",ylab=NULL)
```

```
qqnormPlot(X,title=FALSE)
title(main="X",ylab=NULL)
```

```
qqnormPlot(VALE,title=FALSE)
title(main="VALE",ylab=NULL)
```

```
qqnormPlot(VZ,title=FALSE)
```

```
title(main="vZ",ylab=NULL)
```

```
# Stationarity Test
```

```
# Priestley-Subba Rao
```

```
library(fractal)
```

```
s1<-stationarity(netreturn[,1])
```

```
attr(s1,"pvals")
```

```
s2<-stationarity(netreturn[,2])
```

```
attr(s2,"pvals")
```

```
s3<-stationarity(netreturn[,3])
```

```
attr(s3,"pvals")
```

```
s4<-stationarity(netreturn[,4])
```

```
attr(s4,"pvals")
```

```
s5<-stationarity(netreturn[,5])
```

```
attr(s5,"pvals")
```

```
s6<-stationarity(netreturn[,6])
```

```
attr(s6,"pvals")
```

```
s7<-stationarity(netreturn[,7])
```

```
attr(s7,"pvals")
```

```
s8<-stationarity(netreturn[,8])
```

```
attr(s8,"pvals")
```

```
s9<-stationarity(netreturn[,9])
```

```
attr(s9,"pvals")
```

```
s10<-stationarity(netreturn[,10])
```

```
attr(s10,"pvals")
```

```
s11<-stationarity(netreturn[,11])
```

```
attr(s11,"pvals")
```

```
s12<-stationarity(netreturn[,12])
```

```
attr(s12,"pvals")
```

```
s13<-stationarity(netreturn[,13])
```

```
attr(s13,"pvals")
```

```
s14<-stationarity(netreturn[,14])
```

```
attr(s14,"pvals")
```

```
s15<-stationarity(netreturn[,15])
```

```
attr(s15,"pvals")
```

```
s16<-stationarity(netreturn[,16])
```

```
attr(s16,"pvals")
```

```
# Normality Test
```

```
ks.test(netreturn[,1],"pnorm")
```

```
ks.test(netreturn[,2],"pnorm")
```

```

ks.test(netreturn[,3], "pnorm")
ks.test(netreturn[,4], "pnorm")
ks.test(netreturn[,5], "pnorm")
ks.test(netreturn[,6], "pnorm")
ks.test(netreturn[,7], "pnorm")
ks.test(netreturn[,8], "pnorm")
ks.test(netreturn[,9], "pnorm")
ks.test(netreturn[,10], "pnorm")
ks.test(netreturn[,11], "pnorm")
ks.test(netreturn[,12], "pnorm")
ks.test(netreturn[,13], "pnorm")
ks.test(netreturn[,14], "pnorm")
ks.test(netreturn[,15], "pnorm")
# Source: http://www.statosphere.com.au/check-time-series-stationary-r/

# Fit Distributions
library("fitdistrplus")
library("fGarch")
AICnorm<-rep(0,ncol(netreturn)-1)
BICnorm<-rep(0,ncol(netreturn)-1)
for (i in 1:(ncol(netreturn)-1)){
  fitnorm<-fitdist(netreturn[,i], "norm")
  AICnorm[i]<-fitnorm$aic
  BICnorm[i]<-fitnorm$bic
}
AICnorm
BICnorm

loglik<-rep(0,ncol(netreturn)-1)
AICt<-rep(0,ncol(netreturn)-1)
for(i in 1:(ncol(netreturn)-1)){

fitt<-fitdistr(netreturn[,i], "t", df=ncol(netreturn)-1-2, list(m=stdFit(netreturn[,i])$par[1], s=stdFit(netreturn[,i])$par[2]), lower=c(-1, 0.0001, 1))
  loglik[i]<-fitt$loglik
  AICt[i]<-AIC(fitt)
}
AICt

# Density Plot

par(mfrow = c(1,5))

plot(density(netreturn$AMD, adjust = 0.3, kernel = "gaussian"), main = "Density of AMD (Adjust = 0.3)")

```



```
plot(density(netreturn$MDT, adjust = 3, kernel = "gaussian"), main = "Density of MDT (Adjust = 3)")
plot(density(netreturn$MDT, adjust = 7, kernel = "gaussian"), main = "Density of MDT (Adjust = 7)")
```

```
plot(density(netreturn$MRO, adjust = 0.3, kernel = "gaussian"), main = "Density of MRO (Adjust = 0.3)")
plot(density(netreturn$MRO, adjust = 0.7, kernel = "gaussian"), main = "Density of MRO (Adjust = 0.7)")
plot(density(netreturn$MRO, adjust = 1, kernel = "gaussian"), main = "Density of MRO (Adjust = 1)")
plot(density(netreturn$MRO, adjust = 3, kernel = "gaussian"), main = "Density of MRO (Adjust = 3)")
plot(density(netreturn$MRO, adjust = 7, kernel = "gaussian"), main = "Density of MRO (Adjust = 7)")
```

```
plot(density(netreturn$PFE, adjust = 0.3, kernel = "gaussian"), main = "Density of PFE (Adjust = 0.3)")
plot(density(netreturn$PFE, adjust = 0.7, kernel = "gaussian"), main = "Density of PFE (Adjust = 0.7)")
plot(density(netreturn$PFE, adjust = 1, kernel = "gaussian"), main = "Density of PFE (Adjust = 1)")
plot(density(netreturn$PFE, adjust = 3, kernel = "gaussian"), main = "Density of PFE (Adjust = 3)")
plot(density(netreturn$PFE, adjust = 7, kernel = "gaussian"), main = "Density of PFE (Adjust = 7)")
```

```
plot(density(netreturn$SIRI, adjust = 0.3, kernel = "gaussian"), main = "Density of SIRI (Adjust = 0.3)")
plot(density(netreturn$SIRI, adjust = 0.7, kernel = "gaussian"), main = "Density of SIRI (Adjust = 0.7)")
plot(density(netreturn$SIRI, adjust = 1, kernel = "gaussian"), main = "Density of SIRI (Adjust = 1)")
plot(density(netreturn$SIRI, adjust = 3, kernel = "gaussian"), main = "Density of SIRI (Adjust = 3)")
plot(density(netreturn$SIRI, adjust = 7, kernel = "gaussian"), main = "Density of SIRI (Adjust = 7)")
```

```
plot(density(netreturn$SBUX, adjust = 0.3, kernel = "gaussian"), main = "Density of SBUX (Adjust = 0.3)")
plot(density(netreturn$SBUX, adjust = 0.7, kernel = "gaussian"), main = "Density of SBUX (Adjust = 0.7)")
plot(density(netreturn$SBUX, adjust = 1, kernel = "gaussian"), main = "Density of SBUX (Adjust = 1)")
plot(density(netreturn$SBUX, adjust = 3, kernel = "gaussian"), main = "Density of SBUX (Adjust = 3)")
plot(density(netreturn$SBUX, adjust = 7, kernel = "gaussian"), main = "Density of SBUX (Adjust = 7)")
```

```
plot(density(netreturn$X, adjust = 0.3, kernel = "gaussian"), main = "Density of X (Adjust = 0.3)")
plot(density(netreturn$X, adjust = 0.7, kernel = "gaussian"), main = "Density of X (Adjust = 0.7)")
plot(density(netreturn$X, adjust = 1, kernel = "gaussian"), main = "Density of X (Adjust = 1)")
plot(density(netreturn$X, adjust = 3, kernel = "gaussian"), main = "Density of X (Adjust = 3)")
plot(density(netreturn$X, adjust = 7, kernel = "gaussian"), main = "Density of X (Adjust = 7)")
```

```
plot(density(netreturn$VALE, adjust = 0.3, kernel = "gaussian"), main = "Density of VALE (Adjust = 0.3)")
plot(density(netreturn$VALE, adjust = 0.7, kernel = "gaussian"), main = "Density of VALE (Adjust = 0.7)")
plot(density(netreturn$VALE, adjust = 1, kernel = "gaussian"), main = "Density of VALE (Adjust = 1)")
plot(density(netreturn$VALE, adjust = 3, kernel = "gaussian"), main = "Density of VALE (Adjust = 3)")
plot(density(netreturn$VALE, adjust = 7, kernel = "gaussian"), main = "Density of VALE (Adjust = 7)")
```

```
plot(density(netreturn$VZ, adjust = 0.3, kernel = "gaussian"), main = "Density of VZ (Adjust = 0.3)")
plot(density(netreturn$VZ, adjust = 0.7, kernel = "gaussian"), main = "Density of VZ (Adjust = 0.7)")
plot(density(netreturn$VZ, adjust = 1, kernel = "gaussian"), main = "Density of VZ (Adjust = 1)")
plot(density(netreturn$VZ, adjust = 3, kernel = "gaussian"), main = "Density of VZ (Adjust = 3)")
```

```
plot(density(netreturn$VZ, adjust = 7, kernel = "gaussian"), main = "Density of VZ (Adjust = 7)")
```

```
plot(density(netreturn$SP500, adjust = 0.3, kernel = "gaussian"), main = "Density of SP500 (Adjust = 0.3)")
```

```
plot(density(netreturn$SP500, adjust = 0.7, kernel = "gaussian"), main = "Density of SP500 (Adjust = 0.7)")
```

```
plot(density(netreturn$SP500, adjust = 1, kernel = "gaussian"), main = "Density of SP500 (Adjust = 1)")
```

```
plot(density(netreturn$SP500, adjust = 3, kernel = "gaussian"), main = "Density of SP500 (Adjust = 3)")
```

```
plot(density(netreturn$SP500, adjust = 7, kernel = "gaussian"), main = "Density of SP500 (Adjust = 7)")
```

```
par(mfrow = c(1,1))
```

```
# Scatter Plot
```

```
pairs(~AMD+AAPL+BAC+COKE+FCX+F+GE+MDT+MRO+PFE+SIRI+SBUX+X+VALE+VZ,data=netreturn,  
main="Pairwise Scatter Plots")
```

```
# Covariance Matrix
```

```
cov(netreturn)
```

```
cor(netreturn)
```

```
# Sharpe Ratio --- related to problem 3 ??? Annualize?
```

```
sr<-rep(0,ncol(netreturn))
```

```
for(i in 1:ncol(netreturn)){
```

```
  sr[i]<-(mean(netreturn[,i])*12-rf)/(sd(netreturn[,i])*sqrt(12))
```

```
}
```

```
sr
```

```
#####
```

```
#### 3. Portfolio Theory ####
```

```
#####
```

```
library(Ecdat)
```

```
library(quadprog)
```

```
#### Short sales allowed
```

```
mean_vect<-colMeans(netreturn[,-16])
```

```
cov_mat<-cov(netreturn[,-16])
```

```
sd_vect<-sqrt(diag(cov_mat))
```

```
Amat<-cbind(rep(1,15),mean_vect)
```

```
mup<-seq(min(mean_vect)+0.0001,max(mean_vect)-0.0001,length=300)
```

```
sdp<-mup
```

```
weights<-matrix(0,nrow=300,ncol=15)
```

```
for (i in 1:length(mup)){
```

```

bvec<-c(1,mup[i])
result<-solve.QP(Dmat=2*cov_mat,dvec=rep(0,15),Amat=Amat,bvec=bvec,meq=2)
sdp[i]<-sqrt(result$value)
weights[i,]<-result$solution
}

mufree<-rf/12
plot(sdp,mup,typ='l',xlim=c(0, 0.2),ylim=c(-0.05, 0.1),lty=3,lwd=2,main="Portfolio Management (Short Sales
Allowed)")
points(0,mufree,cex=1,pch="*")
text(0,mufree,sprintf("(%.2f,%.2f)",0,rf),cex=0.8,pos=4)

# Tangent weight
sharpe<-(mup-mufree)/sdp
ind<-(sharpe==max(sharpe))
options(digits=3)
tangent_weight<-weights[ind,]
tangent_weight

lines(c(0,2),mufree+c(0,2)*(mup[ind]-mufree)/sdp[ind],lwd=1,lty=1,col="blue")
points(sdp[ind],mup[ind],cex=1,pch="T")
text(sdp[ind],mup[ind],sprintf("(%.2f,%.2f)",sdp[ind],mup[ind]),cex=0.8,pos=4)

# MVP weight
ind2<-(sdp==min(sdp))
MVP_weight<-weights[ind2,]
MVP_weight
points(sdp[ind2],mup[ind2],cex=1,pch="+")
text(sdp[ind2],mup[ind2],sprintf("(%.2f,%.2f)",sdp[ind2],mup[ind2]),cex=0.8,pos=4)

ind3<-(mup>mup[ind2])
lines(sdp[ind3],mup[ind3],type='l',xlim=c(0,.25),ylim=c(0,.3),lwd=1,col='red')

for(i in 1:15){
  text(sd_vect[i],mean_vect[i],as.character(i),cex=0.5)
}

legend("topleft",
      c("risk-free asset","tangency portfolio","minimum variance portfolio (MVP)","efficient portfolio","efficient
frontier"),
      lty=c(NA,NA,NA,1,1),
      col=c("black","black","black","blue","red"),
      pch=c("","T","+","",""),
      pt.cex = c(1,1,1,1,1))

```



```

# MVP
A<-100000

MVP_weight
round(MVP_weight*100,2)

mean_MPV<-MVP_weight%%mean_vect
A_mean_MPV<-mean_MPV*12

sd_MPV<-sqrt(MVP_weight%%cov_mat%%MVP_weight)
A_sd_MPV<-sd_MPV*sqrt(12)
# sd_MPV<-sqrt(t(as.matrix(MVP_weight))%%cov_mat%%as.matrix(MVP_weight))

MVP_data<-rowSums(as.matrix(netreturn[,-16]))%%diag(MVP_weight)
VaR_MVP<--quantile(MVP_data,0.05)*A
ES_MVP<--sum(MVP_data*(MVP_data<quantile(MVP_data,0.05)))/sum(MVP_data<quantile(MVP_data,0.05))*
A
VaR_MVP2<--A*(mean(MVP_data)+qnorm(0.05)*sd(MVP_data))
ES_MVP2<-A*(-mean(MVP_data)+sd(MVP_data)*dnorm(qnorm(0.05))/0.05)

B=10000
VaR_MVP_a<-rep(0,B)
ES_MVP_a<-rep(0,B)
for (i in 1:B){
  data_a<-sample(MVP_data,length(MVP_data),replace = TRUE)
  VaR_MVP_a[i]<--quantile(data_a,0.05)*A
  ES_MVP_a[i]<--sum(data_a*(data_a<quantile(data_a,0.05)))/sum(data_a<quantile(data_a,0.05))*A
}
CIU_VaR_MVP_a<-2*VaR_MVP-quantile(VaR_MVP_a,0.025)
CIL_VaR_MVP_a<-2*VaR_MVP-quantile(VaR_MVP_a,0.975)
CIU_ES_MVP_a<-2*ES_MVP-quantile(ES_MVP_a,0.025,na.rm=TRUE)
CIL_ES_MVP_a<-2*ES_MVP-quantile(ES_MVP_a,0.975,na.rm=TRUE)

# Sharpe Ratio
MVP_data<-rowSums(as.matrix(netreturn[,-16]))%%diag(MVP_weight)
sr_MVP<-(mean(MVP_data)*12-rf)/(sd(MVP_data)*sqrt(12))

# MVP_data<-sweep(as.matrix(netreturn[,-16]),2,MVP_weight, '*')

# Tangent
tangent_weight
round(tangent_weight*100,2)

```

```

mean_tangent<-tangent_weight%%mean_vect
A_mean_tangent<-mean_tangent*12

sd_tangent<-sqrt(t(as.matrix(tangent_weight))%%cov_mat%%as.matrix(tangent_weight))
A_sd_tangent<-sd_tangent*sqrt(12)

var_tangent<-sd_tangent^2
A_sd_tangent<-var_tangent*12

tangent_data<-rowSums(as.matrix(netreturn[, -16])%%diag(tangent_weight))
VaR_tangent<--quantile(tangent_data,0.05)*A
ES_tangent<--sum(tangent_data*(tangent_data<quantile(tangent_data,0.05)))/sum(tangent_data<quantile(tangent_data,0.05))*A

VaR_tangent2<--A*(mean(tangent_data)+qnorm(0.05)*sd(tangent_data))
ES_tangent2<-A*(-mean(tangent_data)+sd(tangent_data)*dnorm(qnorm(0.05))/0.05)

B=10000
VaR_tangent_b<-rep(0,B)
ES_tangent_b<-rep(0,B)
for (i in 1:B){
  data_b<-sample(tangent_data,length(tangent_data),replace = TRUE)
  VaR_tangent_b[i]<--quantile(data_b,0.05)*A
  ES_tangent_b[i]<--sum(data_b*(data_b<quantile(data_b,0.05)))/sum(data_b<quantile(data_b,0.05))*A
}
CIU_VaR_tangent_b<-2*VaR_tangent-quantile(VaR_tangent_b,0.025)
CIL_VaR_tangent_b<-2*VaR_tangent-quantile(VaR_tangent_b,0.975)
CIU_ES_tangent_b<-2*ES_tangent-quantile(ES_tangent_b,0.025,na.rm=TRUE)
CIL_ES_tangent_b<-2*ES_tangent-quantile(ES_tangent_b,0.975,na.rm=TRUE)

# sharpe ratio
tangent_data<-rowSums(as.matrix(netreturn[, -16])%%diag(tangent_weight))
sr_tangent<-(mean(tangent_data)*12-rf)/(sd(tangent_data)*sqrt(12))

#### Short sales not allowed
mean_vect<-colMeans(netreturn[, -16])
cov_mat<-cov(netreturn[, -16])
sd_vect<-sqrt(diag(cov_mat))

Amat_no<-cbind(rep(1,15),mean_vect,diag(1,nrow=15))
mup_no<-seq(min(mean_vect)+0.0001,max(mean_vect)-0.0001,length=300)
sdp_no<-mup_no
weights_no<-matrix(0,nrow=300,ncol=15)

```

```

for (i in 1:length(mup_no)){
  bvec_no<-c(1,mup_no[i],rep(0.000000000001,15))
  result_no<-solve.QP(Dmat=2*cov_mat,dvec=rep(0,15),Amat=Amat_no,bvec=bvec_no,meq=2)
  sdp_no[i]<-sqrt(result_no$value)
  weights_no[i,<-result_no$solution
}

mufree<-rf/12
plot(sdp_no,mup_no,typ='l',xlim=c(0, 0.2),ylim=c(-0.05, 0.1),lty=3,lwd=2,main="Portfolio Management (Short
Sales Not Allowed)")
points(0,mufree,cex=1,pch="*")
text(0,mufree,sprintf("(% .2f,% .2f)",0,rf),cex=0.8,pos=4)

# Tangent weight
sharpe_no<-(mup_no-mufree)/sdp_no
ind_no<-(sharpe_no==max(sharpe_no))
options(digits=3)
tangent_weight_no<-weights_no[ind_no,]
tangent_weight_no

lines(c(0,2),mufree+c(0,2)*(mup_no[ind_no]-mufree)/sdp_no[ind_no],lwd=1,lty=1,col="blue")
points(sdp_no[ind_no],mup_no[ind_no],cex=1,pch="T")
text(sdp_no[ind_no],mup_no[ind_no],sprintf("(% .2f,% .2f)",sdp_no[ind_no],mup_no[ind_no]),cex=0.8,pos=4)

# MVP weight
ind2_no<-(sdp_no==min(sdp_no))
MVP_weight_no<-weights_no[ind2_no,]
MVP_weight_no
points(sdp_no[ind2_no],mup_no[ind2_no],cex=1,pch="+")
text(sdp_no[ind2_no],mup_no[ind2_no],sprintf("(% .2f,% .2f)",sdp_no[ind2_no],mup_no[ind2_no]),cex=0.8,pos=4)

ind3_no<-(mup_no>mup_no[ind2_no])
lines(sdp_no[ind3_no],mup_no[ind3_no],type='l',xlim=c(0,.25),ylim=c(0,.3),lwd=1,col='red')

for(i in 1:15){
  text(sd_vect[i],mean_vect[i],as.character(i),cex=0.5)
}

legend("topleft",
      c("risk-free asset","tangency portfolio","minimum variance portfolio (MVP)","efficient portfolio","efficient
frontier"),
      lty=c(NA,NA,NA,1,1),
      col=c("black","black","black","blue","red"),

```

```
pch=c("","T","+","",""),
pt.cex = c(1,1,1,1,1))
```

```
# MVP_no
A<-100000
```

```
MVP_weight_no
round(MVP_weight_no*100,2)
```

```
mean_MPV_no<-MVP_weight_no%%mean_vect
A_mean_MPV_no<-mean_MPV_no*12
```

```
sd_MPV_no<-sqrt(MVP_weight_no%%cov_mat%%MVP_weight_no)
A_sd_MPV_no<-sd_MPV_no*sqrt(12)
# sd_MPV<-sqrt(t(as.matrix(MVP_weight))%%cov_mat%%as.matrix(MVP_weight))
```

```
MVP_data_no<-rowSums(as.matrix(netreturn[, -16])%%diag(MVP_weight_no))
VaR_MVP_no<--quantile(MVP_data_no,0.05)*A
ES_MVP_no<--sum(MVP_data_no*(MVP_data_no<quantile(MVP_data_no,0.05)))/sum(MVP_data_no<quantile(
MVP_data_no,0.05))*A
VaR_MVP_no2<-A*(mean(MVP_data_no)+qnorm(0.05)*sd(MVP_data_no))
ES_MVP_no2<-A*(-mean(MVP_data_no)+sd(MVP_data_no)*dnorm(qnorm(0.05))/0.05)
```

```
B=10000
VaR_MVP_c<-rep(0,B)
ES_MVP_c<-rep(0,B)
for (i in 1:B){
  data_c<-sample(MVP_data_no,length(MVP_data_no),replace = TRUE)
  VaR_MVP_c[i]<--quantile(data_c,0.05)*A
  ES_MVP_c[i]<--sum(data_c*(data_c<quantile(data_c,0.05)))/sum(data_c<quantile(data_c,0.05))*A
}
CIU_VaR_MVP_c<-2*VaR_MVP_no-quantile(VaR_MVP_c,0.025)
CIL_VaR_MVP_c<-2*VaR_MVP_no-quantile(VaR_MVP_c,0.975)
CIU_ES_MVP_c<-2*ES_MVP_no-quantile(ES_MVP_c,0.025,na.rm=TRUE)
CIL_ES_MVP_c<-2*ES_MVP_no-quantile(ES_MVP_c,0.975,na.rm=TRUE)
```

```
# Sharpe Ratio
MVP_data_no<-rowSums(as.matrix(netreturn[, -16])%%diag(MVP_weight_no))
sr_MVP_no<-(mean(MVP_data_no)*12-rf)/(sd(MVP_data_no)*sqrt(12))
```

```
# MVP_data<-sweep(as.matrix(netreturn[, -16]),2,MPV_weight, '*')
```

```
# Tangent_no
```

```

tangent_weight_no
round(tangent_weight_no*100,3)

mean_tangent_no<-tangent_weight_no%%mean_vect
A_mean_tangent_no<-mean_tangent_no*12

sd_tangent_no<-sqrt(t(as.matrix(tangent_weight_no))%%cov_mat%%as.matrix(tangent_weight_no))
A_sd_tangent_no<-sd_tangent_no*sqrt(12)

var_tangent_no<-sd_tangent_no^2
A_sd_tangent_no<-var_tangent_no*12

tangent_data_no<-rowSums(as.matrix(netreturn[,-16])%%diag(tangent_weight_no))
VaR_tangent_no<--quantile(tangent_data_no,0.05)*A
ES_tangent_no<--sum(tangent_data_no*(tangent_data_no<quantile(tangent_data_no,0.05)))/sum(tangent_data_no
<quantile(tangent_data_no,0.05))*A
VaR_tangent_no2<--A*(mean(tangent_data_no)+qnorm(0.05)*sd(tangent_data_no))
ES_tangent_no2<-A*(-mean(tangent_data_no)+sd(tangent_data_no)*dnorm(qnorm(0.05))/0.05)

B=10000
VaR_tangent_d<-rep(0,B)
ES_tangent_d<-rep(0,B)
for (i in 1:B){
  data_d<-sample(tangent_data_no,length(tangent_data_no),replace = TRUE)
  VaR_tangent_d[i]<--quantile(data_d,0.05)*A
  ES_tangent_d[i]<--sum(data_d*(data_d<quantile(data_d,0.05)))/sum(data_d<quantile(data_d,0.05))*A
}
CIU_VaR_tangent_d<-2*VaR_tangent_no-quantile(VaR_tangent_d,0.025)
CIL_VaR_tangent_d<-2*VaR_tangent_no-quantile(VaR_tangent_d,0.975)
CIU_ES_tangent_d<-2*ES_tangent_no-quantile(ES_tangent_d,0.025,na.rm=TRUE)
CIL_ES_tangent_d<-2*ES_tangent_no-quantile(ES_tangent_d,0.975,na.rm=TRUE)

# Sharpe ratio
tangent_data_no<-rowSums(as.matrix(netreturn[,-16])%%diag(tangent_weight_no))
sr_tangent_no<-(mean(tangent_data_no)*12-rf)/(sd(tangent_data_no)*sqrt(12))

#####
#### 4. Asset Allocation ####
#####

target<-0.005 #0.5% per month, 6% per year

# No T-Bills
mean_vect<-colMeans(netreturn[,-16])

```

```

cov_mat<-cov(netreturn[, -16])
sd_vect<-sqrt(diag(cov_mat))

Amat_t<-cbind(rep(1,15),mean_vect,diag(1,nrow=15))
mup_t<-seq(0.005,0.005,length=300)
sdp_t<-mup_t
weights_t<-matrix(0,nrow=300,ncol=15)

for (i in 1:length(mup_t)){
  bvec_t<-c(1,mup_t[i],rep(0.000000000001,15))
  result_t<-solve.QP(Dmat=2*cov_mat,dvec=rep(0,15),Amat=Amat_t,bvec=bvec_t,meq=2)
  sdp_t[i]<-sqrt(result_t$value)
  weights_t[i,]<-result_t$solution
}

target_weight<-weights_t[1,]
mean_t<-target_weight%%mean_vect
A_mean_t<-mean_t*12

sd_t<-sqrt(t(as.matrix(target_weight))%%cov_mat%%as.matrix(target_weight))
A_sd_t<-sd_t*sqrt(12)

t_data<-rowSums(as.matrix(netreturn[, -16])%%diag(target_weight))
VaR_t<--quantile(t_data,0.05)*A
ES_t<--sum(t_data*(t_data<quantile(t_data,0.05)))/sum(t_data<quantile(t_data,0.05))*A
VaR_t2<--A*(mean(t_data)+qnorm(0.05)*sd(t_data))
ES_t2<-A*(-mean(t_data)+sd(t_data)*dnorm(qnorm(0.05))/0.05)

B=10000
VaR_t_e<-rep(0,B)
ES_t_e<-rep(0,B)
for (i in 1:B){
  data_e<-sample(t_data,length(t_data),replace = TRUE)
  VaR_t_e[i]<--quantile(data_e,0.05)*A
  ES_t_e[i]<--sum(data_e*(data_e<quantile(data_e,0.05)))/sum(data_e<quantile(data_e,0.05))*A
}
CIU_VaR_t_e<-2*VaR_t-quantile(VaR_t_e,0.025)
CIL_VaR_t_e<-2*VaR_t-quantile(VaR_t_e,0.975)
CIU_ES_t_e<-2*ES_t-quantile(ES_t_e,0.025,na.rm=TRUE)
CIL_ES_t_e<-2*ES_t-quantile(ES_t_e,0.975,na.rm=TRUE)

# Sharpe ratio
t_data<-rowSums(as.matrix(netreturn[, -16])%%diag(target_weight))
sr_t<-(mean(t_data)*12-rf)/(sd(t_data)*sqrt(12))

```

```

# using T-Bills, no short sales
# wp*miup+(1-wp)*miufree=0.005
wp_no<-(0.005-mufree)/(mean_tangent_no-mufree)
round(wp_no*tangent_weight_no*100,2)
round((1-wp_no)*100,2) # risk free proportion

mean_t_no<-(1-wp_no)*mufree+wp_no*mean_tangent_no
A_mean_t_no<-mean_t_no*12
sd_t_no<-sqrt(wp_no^2*sd_tangent_no^2)
A_sd_t_no<-sd_t_no*sqrt(12)

t_data_no<-rowSums(as.matrix(netreturn[,-16])%*%diag(wp_no*tangent_weight_no))+mufree*(1-wp_no)
VaR_t_no<--quantile(t_data_no,0.05)*A
ES_t_no<--sum(t_data_no*(t_data_no<quantile(t_data_no,0.05)))/sum(t_data_no<quantile(t_data_no,0.05))*A
VaR_t_no2<--A*(mean(t_data_no)+qnorm(0.05)*sd(t_data_no))
ES_t_no2<-A*(-mean(t_data_no)+sd(t_data_no)*dnorm(qnorm(0.05))/0.05)

B=10000
VaR_t_f<-rep(0,B)
ES_t_f<-rep(0,B)
for (i in 1:B){
  data_f<-sample(t_data_no,length(t_data_no),replace = TRUE)
  VaR_t_f[i]<--quantile(data_f,0.05)*A
  ES_t_f[i]<--sum(data_f*(data_f<quantile(data_f,0.05)))/sum(data_f<quantile(data_f,0.05))*A
}
CIU_VaR_t_f<-2*VaR_t_no-quantile(VaR_t_f,0.025)
CIL_VaR_t_f<-2*VaR_t_no-quantile(VaR_t_f,0.975)
CIU_ES_t_f<-2*ES_t_no-quantile(ES_t_f,0.025,na.rm=TRUE)
CIL_ES_t_f<-2*ES_t_no-quantile(ES_t_f,0.975,na.rm=TRUE)

# Sharpe ratio
t_data_no<-rowSums(as.matrix(netreturn[,-16])%*%diag(wp_no*tangent_weight_no))+mufree*(1-wp_no)
sr_t_no<-(mean(t_data_no)*12-rf)/(sd(t_data_no)*sqrt(12))

sr_tangent_no<-(A_mean_t_no-rf)/A_sd_t_no
# (mean(tangent_data_no)*12-rf)/(sd(tangent_data_no)*sqrt(12))

#####
#### 6. Risk Management ####
#####

```

```
A=100000
```

```
# Normal Distribution
```

```
VaR15<-rep(0,15)
```

```
ES15<-rep(0,15)
```

```
for (i in 1:15){
```

```
  # VaR15[i]<--VaR(netreturn[,i],0.05,method="gaussian")*A
```

```
  # ES15[i]<--ES(netreturn[,i],0.05,method="gaussian")*A
```

```
  VaR15[i]<--A*(mean(netreturn[,i])+qnorm(0.05)*sd(netreturn[,i]))
```

```
  ES15[i]<-A*((-mean(netreturn[,i])+sd(netreturn[,i])*dnorm(qnorm(0.05))/0.05))
```

```
}
```

```
VaR15
```

```
ES15
```

```
VaR15[which.max(VaR15)]
```

```
which.max(VaR15)
```

```
ES15[which.max(ES15)]
```

```
which.max(ES15)
```

```
VaR15[which.min(VaR15)]
```

```
which.min(VaR15)
```

```
ES15[which.min(ES15)]
```

```
which.min(ES15)
```

```
# -A*(mean(AAPL)+qnorm(0.05)*sd(AAPL))
```

```
# Nonparametric
```

```
VaR15_non<-rep(0,15)
```

```
ES15_non<-rep(0,15)
```

```
for(i in 1:15){
```

```
  VaR15_non[i]<--quantile(netreturn[,i],0.05)*A
```

```
  ES15_non[i]<--sum(netreturn[,i]*(netreturn[,i]<quantile(netreturn[,i],0.05)))/sum(netreturn[,i]<quantile(netreturn[,i],0.05))*A
```

```
}
```

```
VaR15_non
```

```
ES15_non
```

```
VaR15_non[which.max(VaR15_non)]
```



```

which.max(VaR15_non)
ES15_non[which.max(ES15_non)]
which.max(ES15_non)

VaR15_non[which.min(VaR15_non)]
which.min(VaR15_non)
ES15_non[which.min(ES15_non)]
which.min(ES15_non)

# Bootstrap
B=10000
sample_boot<-rep(0,dim(netreturn)[1])
data_boot<-matrix(0,ncol=ncol(netreturn)-1,nrow=nrow(netreturn))
VaR15_boot<-matrix(0,ncol=15,nrow=B)
ES15_boot<-matrix(0,ncol=15,nrow=B)

for ( i in 1:B){
  for(j in 1:15){
    sample_boot<-sample(netreturn[,j],size=dim(netreturn)[1],replace=TRUE)
    data_boot[,j]<-sample_boot
    VaR15_boot[i,j]<--quantile(sample_boot,0.05)*A

    ES15_boot[i,j]<--sum(sample_boot*(sample_boot<quantile(sample_boot,0.05)))/sum(sample_boot<quantile(sample
e_boot,0.05))*A
  }

}

SE_VaR_boot<-apply(VaR15_boot,2,sd)
SE_ES_boot<-apply(ES15_boot,2,sd)

CIU_VaR<-2*VaR15_non-apply(VaR15_boot,2,quantile,0.025)
CIL_VaR<-2*VaR15_non-apply(VaR15_boot,2,quantile,0.975)

CIU_ES<-2*ES15_non-apply(ES15_boot,2,quantile,0.025,na.rm=TRUE)
CIL_VaR<-2*ES15_non-apply(ES15_boot,2,quantile,0.975,na.rm=TRUE)

#####
#### 5. PCA          #####
#####

# Correlation
library(corrplot)
mat<-cor(netreturn[,16])

```

```
print(round(cor(data3[,-16]),3))
```

```
par(mfrow = c(1,2))  
corrplot.mixed(mat)
```

```
cor.mtest <- function(mat, conf.level = 0.95){  
  mat <- as.matrix(mat)  
  n <- ncol(mat)  
  p.mat <- lowCI.mat <- uppCI.mat <- matrix(NA, n, n)  
  diag(p.mat) <- 0  
  diag(lowCI.mat) <- diag(uppCI.mat) <- 1  
  for(i in 1:(n-1)){  
    for(j in (i+1):n){  
      tmp <- cor.test(mat[,i], mat[,j], conf.level = conf.level)  
      p.mat[i,j] <- p.mat[j,i] <- tmp$p.value  
      lowCI.mat[i,j] <- lowCI.mat[j,i] <- tmp$conf.int[1]  
      uppCI.mat[i,j] <- uppCI.mat[j,i] <- tmp$conf.int[2]  
    }  
  }  
  return(list(p.mat, lowCI.mat, uppCI.mat))  
}
```

```
res1 <- cor.mtest(netreturn[,-16],0.95)
```

```
# specialized the insignificant value according to the significant level  
corrplot(mat, p.mat = res1[[1]], sig.level=0.05)  
par(mfrow = c(1,1))
```

```
## PCA
```

```
pca<-prcomp(mat,scale=TRUE)  
pca$rotation  
summary(pca)
```

```
# plot the scree plot
```

```
eig = (pca$sdev)^2  
variance = eig*100/sum(eig)  
cumvar = cumsum(variance)  
p.eig <- data.frame(eig = eig, variance = variance, cumvariance = cumvar)  
p.eig  
barplot(p.eig[, 2], names.arg=1:nrow(p.eig),main = "Variances explained by PC's",ylim =c(0,30),  
        xlab = "Principal Components", ylab = "Percentage of variances",col ="darkblue")  
# Add connected line segments to the plot  
p.lab=p.eig[,2]
```

```

p.label=round(p.lab, digits = 2)
lines(x = 1:nrow(p.eig), p.eig[, 2], type="b", pch=19, col = "red")
textxy(1:nrow(p.eig), p.eig[, 2],
      labs = c(p.label),cex = 0.8, offset = c(0.5, 0.8),col="red")

# # install.packages("devtools")
# # devtools::install_github("kassambara/factoextra")
# library("devtools")
# library("factoextra")
# fviz_screplot(pca,ncp=15,addlabels=T)
#
# # PCA Analysis on yield curve
# f1<-pca$rotation[,1]
# f2<-pca$rotation[,2]
# f3<-pca$rotation[,3]
# plot(f1,type="b",main="PCA on yield
curve",ylim=c(min(f1,f2,f3)-.5,max(f1,f2,f3)+.5),xlab="Term",ylab="Factor Loadings")
# lines(f2,col="blue",type="b")
# lines(f3,col="red",type="b")
# legend.factor<-c('1st PC','2nd PC','3rd PC')
# legend("bottomright",lty=c(1,1,1),legend=legend.factor,col=c("black","blue","red"))

# Plot the correlation circle
a <- seq(0, 2*pi, length = 100)
plot( cos(a), sin(a), type = 'l', col="gray",
      xlab = "PC1 (27.6%)",  ylab = "PC2 (19.6%)",
      main = "Projection of the First Two Pricipal Components")
abline(h = 0, v = 0, lty = 2)
# Add active variables
var_cor_func <- function(var.loadings, comp.sdev){
  var.loadings*comp.sdev
}
loadings <- pca$rotation
sdev <- pca$sdev
var.coord <- t(apply(loadings, 1, var_cor_func, sdev))
arrows(0, 0, var.coord[, 1], var.coord[, 2],
      length = 0.1, angle = 15, code = 2)
text(var.coord, labels=rownames(var.coord), cex = 1, adj=1)

fviz_pca_var(pca, col.var="contrib") +
  scale_color_gradient2(low="green", mid="royalblue2",
                        high="red", midpoint=50) + theme_minimal()

```

```
fviz_pca_contrib(pca, choice = "var", axes = 1, top = 9)
fviz_pca_contrib(pca, choice = "var", axes = 2, top = 7)
fviz_pca_contrib(pca, choice = "var", axes = 3, top = 7)
```

```
## Factor Models
```

```
library("psych")
library("GPArotation")
describe(netreturn[,-16])
cor.plot(mat,numbers=TRUE,main="Correlation Matrix Heatmap")
```

```
ft<-fa(netreturn[,-16],3)
fa.diagram(ft)
```

```
ft2<-factanal(netreturn[,-16],factors=2,rotation="varimax")
print(ft2,digits=3,sort=TRUE)
```

```
ft3<-factanal(netreturn[,-16],factors=3,rotation="varimax")
print(ft3,digits=3,sort=TRUE)
```

```
ft4<-factanal(netreturn[,-16],factors=4,rotation="varimax")
print(ft4,digits=3,sort=TRUE)
```

```
ft5<-factanal(netreturn[,-16],factors=5,rotation="varimax")
print(ft5,digits=2,sort=TRUE)
```

```
ft4.load = ft4$loadings[1:15,]
ft4.loading =as.data.frame(ft4.load)
```

```
plot(ft4.load,type = "p",pch=19, col = "red", main = "Loadings for Factor1 and Factor2")
textxy(ft4.loading$Factor1,ft4.loading$Factor2,
       labs = row.names(ft4.load),cex = 0.8, offset =0.5,srt = 30,col="royalblue2")
```

```
plot(ft4.load[,3:4],type = "p",pch=19, col = "red", main = "Loadings for Factor3 and Factor4")
textxy(ft4.loading$Factor3,ft4.loading$Factor4,
       labs = row.names(ft4.load),cex = 0.8, offset =0.5,srt = 15,col="royalblue2")
```

```
#####
```

```
#### 7. Copula      ####
```

```
#####
```

```
library("copula")
# library("VineCopula")
```

```

# Gaussian, t, archimedean, clayton, gumbel

cop.norm = normalCopula(dim=15)
fit.copnorm = fitCopula(cop.norm,pobs(netreturn[,-16]),method="ml")
fit.copnorm
AIC(fit.copnorm)
BIC(fit.copnorm)
logLik(fit.copnorm)

cop.t = tCopula(dim=15)
fit.copt = fitCopula(cop.t,pobs(netreturn[,-16]),method = "ml")
fit.copt
AIC(fit.copt)
BIC(fit.copt)
logLik(fit.copt)

rho<-coef(fit.copt)[1]
df<-coef(fit.copt)[2]
persp(tCopula(dim=2,rho,df=df),dCopula)
u<-rCopula(4000,tCopula(dim=15,rho,df=df))
qplot(u[,1],u[,2],colour = u[,1], main="t copula random samples", xlab = "u", ylab = "v")
# plot(u[,1],u[,2],pch="*",col="blue")

cop.clayton = claytonCopula(dim=15)
fit.copclayton = fitCopula(cop.clayton,pobs(netreturn[,-16]),method = "ml")
fit.copclayton
AIC(fit.copclayton)
BIC(fit.copclayton)
logLik(fit.copclayton)

cop.gumbel = gumbelCopula(dim=15)
fit.copgumbel = fitCopula(cop.gumbel,pobs(netreturn[,-16]),method = "ml")
fit.copgumbel
AIC(fit.copgumbel)
BIC(fit.copgumbel)
logLik(fit.copgumbel)

cop.archm = archmCopula(family = "frank",dim=15)
fit.archm = fitCopula(cop.archm, pobs(netreturn[,-16]),method = "ml")
fit.archm
AIC(fit.archm)
BIC(fit.archm)
logLik(fit.archm)

```