Econ\_HW4

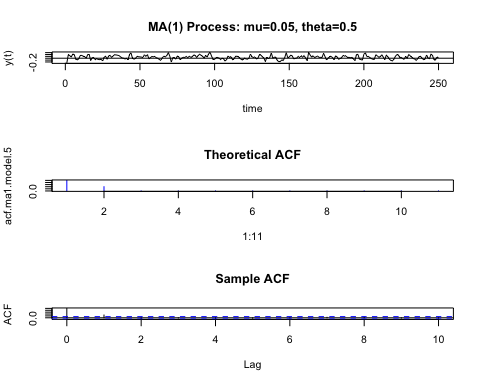
Arash Barmas

2/28/2019

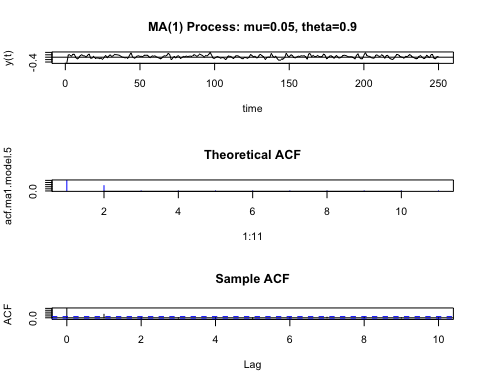
R Exercise A Question 1 We have an MA(1) model Simulate and plot 250 obser- vations of the MA(1)

1 a)

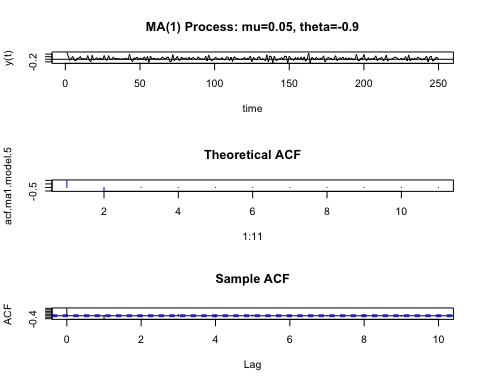
theta = 0.5  
mu = 0.05  
ma1.model.5 = list(ma=theta)  
  
set.seed(123)  
ma1.sim.5 = mu + arima.sim(model=ma1.model.5, n=250,  
 innov=rnorm(n=250, mean=0, sd=0.1))  
  
acf.ma1.model.5 = ARMAacf(ma=theta, lag.max=10)  
par(mfrow=c(3,1))  
 plot(ma1.sim.5, main="MA(1) Process: mu=0.05, theta=0.5",xlab="time",ylab="y(t)")  
 abline(h=0)  
 plot(1:11, acf.ma1.model.5, type="h", col="blue", main="Theoretical ACF")  
 tmp=acf(ma1.sim.5, lag.max=10, main="Sample ACF")



theta = 0.9  
mu = 0.05  
ma1.model.5 = list(ma=theta)  
  
set.seed(123)  
ma1.sim.5 = mu + arima.sim(model=ma1.model.5, n=250,  
 innov=rnorm(n=250, mean=0, sd=0.1))  
  
acf.ma1.model.5 = ARMAacf(ma=theta, lag.max=10)  
par(mfrow=c(3,1))  
 plot(ma1.sim.5, main="MA(1) Process: mu=0.05, theta=0.9",xlab="time",ylab="y(t)")  
 abline(h=0)  
 plot(1:11, acf.ma1.model.5, type="h", col="blue", main="Theoretical ACF")  
 tmp=acf(ma1.sim.5, lag.max=10, main="Sample ACF")



theta = -0.9  
mu = 0.05  
ma1.model.5 = list(ma=theta)  
  
set.seed(123)  
ma1.sim.5 = mu + arima.sim(model=ma1.model.5, n=250,  
 innov=rnorm(n=250, mean=0, sd=0.1))  
  
acf.ma1.model.5 = ARMAacf(ma=theta, lag.max=10)  
par(mfrow=c(3,1))  
 plot(ma1.sim.5, main="MA(1) Process: mu=0.05, theta=-0.9",xlab="time",ylab="y(t)")  
 abline(h=0)  
 plot(1:11, acf.ma1.model.5, type="h", col="blue", main="Theoretical ACF")  
 tmp=acf(ma1.sim.5, lag.max=10, main="Sample ACF")

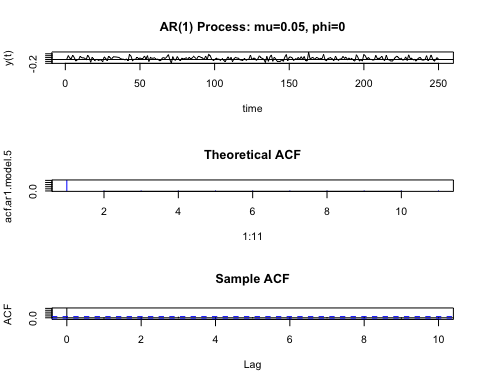


1 b) As we know our model is MA(1), so Y(t) is a linear combination of mu, e(t), and e(t-1). Also, the sign of the correlations of two RV with lag 1 (Y(t) and Y(t-1)) is determined by theta. By increasing theta, the first lag on the ACF didn’t change much.

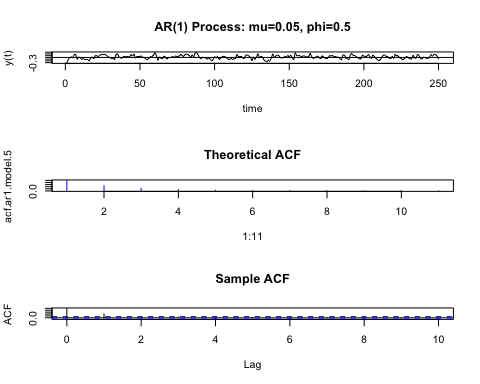
Question 2

2 a)

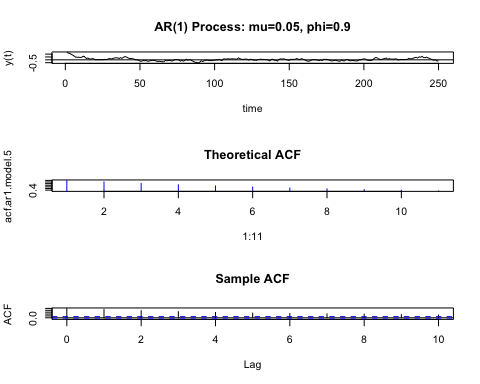
phi = 0   
mu = 0.05  
ar1.model.5 = list(ar=phi)  
  
set.seed(123)  
ar1.sim.5 = suppressWarnings(mu\*(1-phi) + arima.sim(model=ar1.model.5, n = 250,  
 innov=rnorm(n=250, mean=0, sd=0.1)))  
acf.ar1.model.5 = ARMAacf(ar=phi, lag.max=10)  
  
par(mfrow=c(3,1))  
 plot(ar1.sim.5, main="AR(1) Process: mu=0.05, phi=0",  
 xlab="time",ylab="y(t)")  
 abline(h=0)  
 plot(1:11, acf.ar1.model.5, type="h", col="blue", main="Theoretical ACF")  
 tmp=acf(ar1.sim.5, lag.max=10, main="Sample ACF")



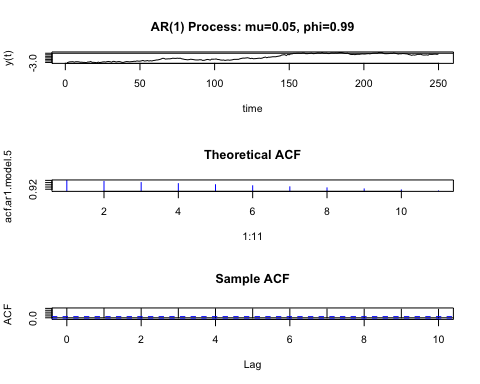
phi = 0.5  
mu = 0.05  
ar1.model.5 = list(ar=phi)  
  
set.seed(123)  
ar1.sim.5 = mu\*(1-phi) + arima.sim(model=ar1.model.5, n = 250,  
 innov=rnorm(n=250, mean=0, sd=0.1))  
acf.ar1.model.5 = ARMAacf(ar=phi, lag.max=10)  
  
par(mfrow=c(3,1))  
 plot(ar1.sim.5, main="AR(1) Process: mu=0.05, phi=0.5",  
 xlab="time",ylab="y(t)")  
 abline(h=0)  
 plot(1:11, acf.ar1.model.5, type="h", col="blue", main="Theoretical ACF")  
 tmp=acf(ar1.sim.5, lag.max=10, main="Sample ACF")



phi = 0.9  
mu = 0.05  
ar1.model.5 = list(ar=phi)  
  
set.seed(123)  
ar1.sim.5 = mu\*(1-phi) + arima.sim(model=ar1.model.5, n = 250,  
 innov=rnorm(n=250, mean=0, sd=0.1))  
acf.ar1.model.5 = ARMAacf(ar=phi, lag.max=10)  
  
par(mfrow=c(3,1))  
 plot(ar1.sim.5, main="AR(1) Process: mu=0.05, phi=0.9",  
 xlab="time",ylab="y(t)")  
 abline(h=0)  
 plot(1:11, acf.ar1.model.5, type="h", col="blue", main="Theoretical ACF")  
 tmp=acf(ar1.sim.5, lag.max=10, main="Sample ACF")



phi = 0.99  
mu = 0.05  
ar1.model.5 = list(ar=phi)  
  
set.seed(123)  
ar1.sim.5 = mu\*(1-phi) + arima.sim(model=ar1.model.5, n = 250,  
 innov=rnorm(n=250, mean=0, sd=0.1))  
acf.ar1.model.5 = ARMAacf(ar=phi, lag.max=10)  
  
par(mfrow=c(3,1))  
 plot(ar1.sim.5, main="AR(1) Process: mu=0.05, phi=0.99",  
 xlab="time",ylab="y(t)")  
 abline(h=0)  
 plot(1:11, acf.ar1.model.5, type="h", col="blue", main="Theoretical ACF")  
 tmp=acf(ar1.sim.5, lag.max=10, main="Sample ACF")

 2 b) By observing the ACFs, we ifer that the phi effects the correlation, and hence dependence in out process. Also, for large values of phi, the process looks like a non-stationary process. For phi =0, the process will turn to Y(t) = mu + e(t), thus ACF graph doesn’t show any correlation.

R Exercise B

Question 1

options(digits=4, width=70)  
library(PerformanceAnalytics)

## Loading required package: xts

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

##   
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':  
##   
## legend

library(zoo)  
library(tseries)  
VBLTX.prices = get.hist.quote(instrument="vbltx", start="1998-01-01",  
 end="2009-12-31", quote="AdjClose",  
 provider="yahoo", origin="1970-01-01",  
 compression="m", retclass="zoo")

## 'getSymbols' currently uses auto.assign=TRUE by default, but will  
## use auto.assign=FALSE in 0.5-0. You will still be able to use  
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")  
## and getOption("getSymbols.auto.assign") will still be checked for  
## alternate defaults.  
##   
## This message is shown once per session and may be disabled by setting   
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

##   
## WARNING: There have been significant changes to Yahoo Finance data.  
## Please see the Warning section of '?getSymbols.yahoo' for details.  
##   
## This message is shown once per session and may be disabled by setting  
## options("getSymbols.yahoo.warning"=FALSE).

## time series ends 2009-12-01

index(VBLTX.prices) = as.yearmon(index(VBLTX.prices))  
  
FMAGX.prices = get.hist.quote(instrument="fmagx", start="1998-01-01",  
 end="2009-12-31", quote="AdjClose",  
 provider="yahoo", origin="1970-01-01",  
 compression="m", retclass="zoo")

## time series ends 2009-12-01

index(FMAGX.prices) = as.yearmon(index(FMAGX.prices))  
  
SBUX.prices = get.hist.quote(instrument="sbux", start="1998-01-01",  
 end="2009-12-31", quote="AdjClose",  
 provider="yahoo", origin="1970-01-01",  
 compression="m", retclass="zoo")

## time series ends 2009-12-01

index(SBUX.prices) = as.yearmon(index(SBUX.prices))  
#merged data  
lab4Prices.z = merge(VBLTX.prices, FMAGX.prices, SBUX.prices)  
colnames(lab4Prices.z) = c("VBLTX", "FMAGX", "SBUX")  
lab4Returns.z = diff(log(lab4Prices.z))  
  
start(lab4Returns.z)

## [1] "Feb 1998"

end(lab4Returns.z)

## [1] "Dec 2009"

colnames(lab4Returns.z)

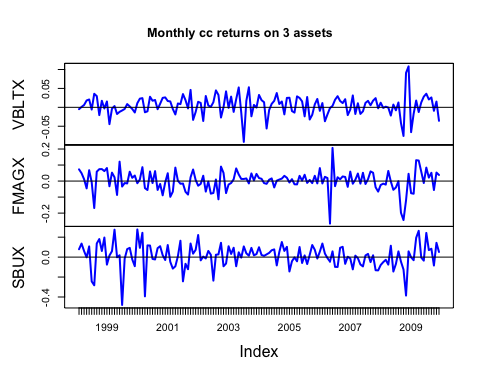
## [1] "VBLTX" "FMAGX" "SBUX"

head(lab4Returns.z)

## VBLTX FMAGX SBUX  
## Feb 1998 -0.004881 0.073069 0.078858  
## Mar 1998 0.001037 0.049066 0.135702  
## Apr 1998 0.006181 0.011513 0.060219  
## May 1998 0.018136 -0.045728 -0.002601  
## Jun 1998 0.020749 0.067677 0.107312  
## Jul 1998 -0.005911 -0.007508 -0.243824

1 a)

my.panel <- function(...) {  
 lines(...)  
 abline(h=0)  
}  
plot(lab4Returns.z, col="blue", lwd=2, main="Monthly cc returns on 3 assets",  
 panel=my.panel)



The scale for each graph is different. The graph of VBLTX has lowest volatility, and SBUX has highest volatility if we conisder the sclae difference. In 2018, VBlTX went up, but other two funds went down.

chart.TimeSeries(lab4Returns.z, legend.loc="bottom", main="")

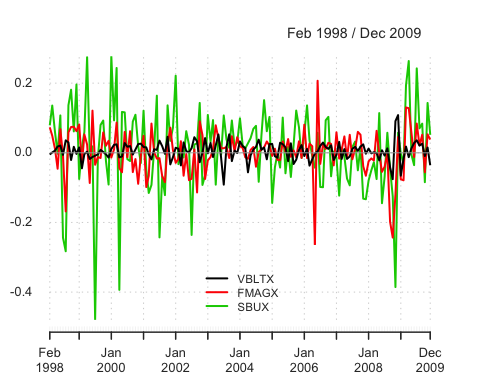
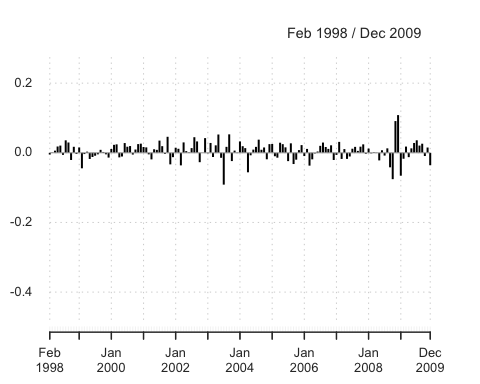
 We can observe the volatility better by comparing them all on the same graph.

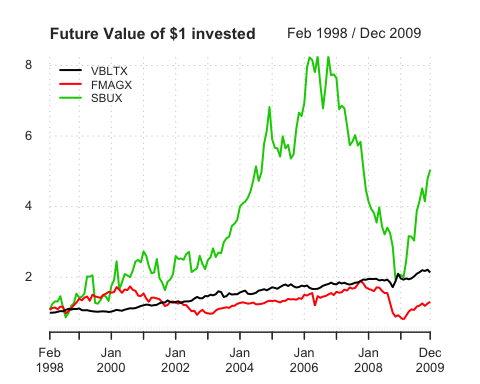
chart.Bar(lab4Returns.z, main="")

## Warning in plot.xts(x = y, y = NULL, ..., col = colorset, type = type,  
## lty = lty, : only the univariate series will be plotted



1 b)

chart.CumReturns(diff(lab4Prices.z)/lag(lab4Prices.z, k=-1),   
 legend.loc="topleft", wealth.index=TRUE,  
 main="Future Value of $1 invested")

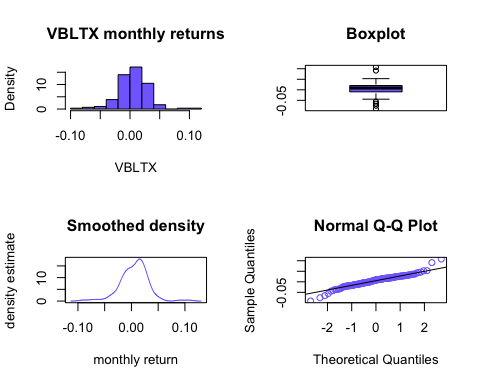


ret.mat = coredata(lab4Returns.z)

Over the investment horizon, SBUX had the best future value, but also had highest volatility VBLTX had steady return. FMAGX had the worst future value.

1 c)

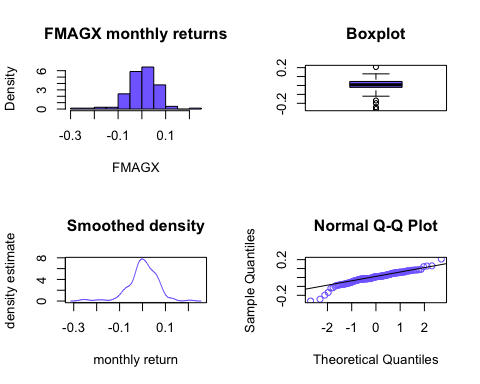
par(mfrow=c(2,2))  
 hist(ret.mat[,"VBLTX"],main="VBLTX monthly returns",  
 xlab="VBLTX", probability=T, col="slateblue1")  
 boxplot(ret.mat[,"VBLTX"],outchar=T, main="Boxplot", col="slateblue1")  
 plot(density(ret.mat[,"VBLTX"]),type="l", main="Smoothed density",  
 xlab="monthly return", ylab="density estimate", col="slateblue1")  
 qqnorm(ret.mat[,"VBLTX"], col="slateblue1")  
 qqline(ret.mat[,"VBLTX"])



par(mfrow=c(1,1))

The distribution is relatively symmetric with fatter tails on the left side as we see it on box-plot and hist density. The Q-Q plot indicates some differences on the tails, but pretty much like normal on the middle.

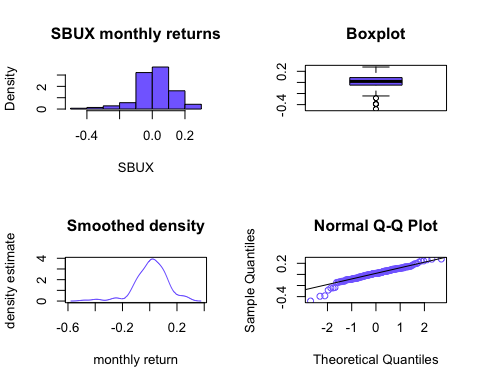
par(mfrow=c(2,2))  
 hist(ret.mat[,"FMAGX"],main="FMAGX monthly returns",  
 xlab="FMAGX", probability=T, col="slateblue1")  
 boxplot(ret.mat[,"FMAGX"],outchar=T, main="Boxplot", col="slateblue1")  
 plot(density(ret.mat[,"FMAGX"]),type="l", main="Smoothed density",  
 xlab="monthly return", ylab="density estimate", col="slateblue1")  
 qqnorm(ret.mat[,"FMAGX"], col="slateblue1")  
 qqline(ret.mat[,"FMAGX"])



par(mfrow=c(1,1))

The distribution doesn’t look normal.(more data on the right, negatively skewed) The Q-Q plot indicates fat tail on the left side than normal, but simillar to normal on the middle.

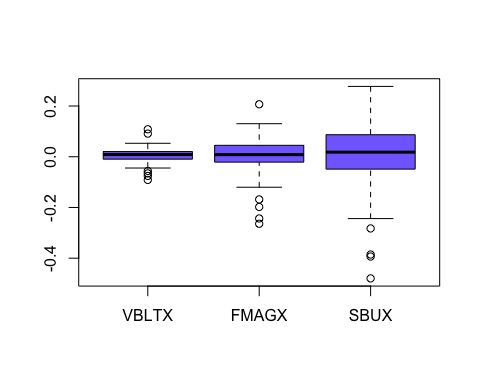
par(mfrow=c(2,2))  
 hist(ret.mat[,"SBUX"],main="SBUX monthly returns",  
 xlab="SBUX", probability=T, col="slateblue1")  
 boxplot(ret.mat[,"SBUX"],outchar=T, main="Boxplot", col="slateblue1")  
 plot(density(ret.mat[,"SBUX"]),type="l", main="Smoothed density",  
 xlab="monthly return", ylab="density estimate", col="slateblue1")  
 qqnorm(ret.mat[,"SBUX"], col="slateblue1")  
 qqline(ret.mat[,"SBUX"])



par(mfrow=c(1,1))

The distribution is not very symmetric with(somehow negatively skewed) The Q-Q plot indicated left tail doesn’t look like normal but right is more simillar

boxplot(ret.mat[,"VBLTX"], ret.mat[,"FMAGX"], ret.mat[,"SBUX"],  
 names=colnames(ret.mat), col="slateblue1")

 This box-plot graph will help us to compare them all together b/c they are on the same scale. As we can see, VBLTX has the lowest volatility, and SBUX has the highest.

1 d)

summary(ret.mat)

## VBLTX FMAGX SBUX   
## Min. :-0.09138 Min. :-0.26430 Min. :-0.4797   
## 1st Qu.:-0.00958 1st Qu.:-0.02124 1st Qu.:-0.0488   
## Median : 0.00855 Median : 0.00825 Median : 0.0182   
## Mean : 0.00530 Mean : 0.00186 Mean : 0.0113   
## 3rd Qu.: 0.02038 3rd Qu.: 0.04503 3rd Qu.: 0.0868   
## Max. : 0.10827 Max. : 0.20686 Max. : 0.2773

print("Mean")

## [1] "Mean"

apply(ret.mat, 2, mean)

## VBLTX FMAGX SBUX   
## 0.005302 0.001856 0.011318

print("var")

## [1] "var"

apply(ret.mat, 2, var)

## VBLTX FMAGX SBUX   
## 0.0006903 0.0041077 0.0142545

print("SD")

## [1] "SD"

apply(ret.mat, 2, sd)

## VBLTX FMAGX SBUX   
## 0.02627 0.06409 0.11939

print("skewness")

## [1] "skewness"

apply(ret.mat, 2, skewness)

## VBLTX FMAGX SBUX   
## -0.1470 -0.9284 -0.9061

print("kurtosis")

## [1] "kurtosis"

apply(ret.mat, 2, kurtosis)

## VBLTX FMAGX SBUX   
## 2.887 3.398 2.695

SBUX has the highest mean, and all the means are positive. Variance of SUBUX is the highest, and VBLTX has the lowest variance which we observed it in the previous parts. Skeweness is negative for all of them. Meaning that distributions are all negatively skewed. Kurtosis (excess) indicates that they all have fatter tails than nomral.

1 e) and f)

# annualized cc mean   
12\*apply(ret.mat, 2, mean)

## VBLTX FMAGX SBUX   
## 0.06363 0.02227 0.13582

# annualized simple mean  
exp(12\*apply(ret.mat, 2, mean)) - 1

## VBLTX FMAGX SBUX   
## 0.06569 0.02252 0.14548

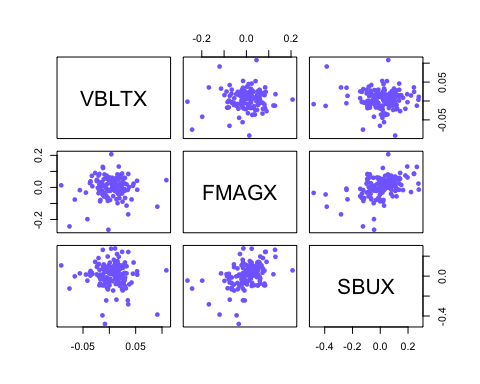
# annualized sd values  
sqrt(12)\*apply(ret.mat, 2, sd)

## VBLTX FMAGX SBUX   
## 0.09101 0.22202 0.41359

first row is the cc return. SBUX has the highest(but with high sd). VBLTX has the lowest mean. The data makes sense because stock the most volatile(more than portfolios), and portfolios are also more volatile than bonds. sd value for SBUX is the highest, and lowest for VBlTX.

1 g)

pairs(ret.mat, col="slateblue1", pch=16)

 No clear relation exists except between FMAGX and SBUX which a linear relationship can be seen.

1 h)

# compute 3 x 3 covariance and correlation matrices  
var(ret.mat)

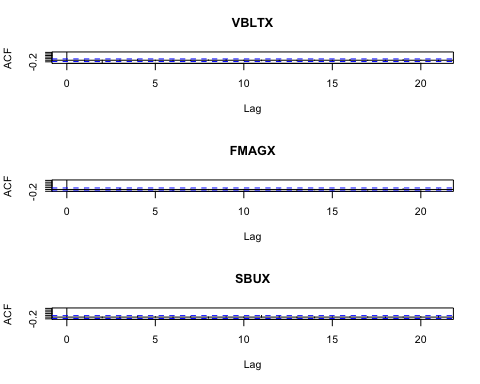
## VBLTX FMAGX SBUX  
## VBLTX 0.0006903 0.0001074 -0.0001761  
## FMAGX 0.0001074 0.0041077 0.0032432  
## SBUX -0.0001761 0.0032432 0.0142545

cor(ret.mat)

## VBLTX FMAGX SBUX  
## VBLTX 1.00000 0.06376 -0.05614  
## FMAGX 0.06376 1.00000 0.42384  
## SBUX -0.05614 0.42384 1.00000

cov between VBLTX and FMAGX is positive (positive linear relation) cov between FMAGX and SBUX is positive (positive linear relation) cov between VBLTX and SBUX is negative (negative linear relation)

par(mfrow=c(3,1))  
 acf.msft = acf(ret.mat[,"VBLTX"], main="VBLTX")  
 acf.sbux = acf(ret.mat[,"FMAGX"], main="FMAGX")  
 acf.sp500 = acf(ret.mat[,"SBUX"], main="SBUX")



par(mfrow=c(1,1))

It shows that it is almost uncorrelated for all 3 assets.

2 a)

muhat.vals = apply(ret.mat, 2, mean)  
muhat.vals

## VBLTX FMAGX SBUX   
## 0.005302 0.001856 0.011318

sigma2hat.vals = apply(ret.mat, 2, var)  
sigma2hat.vals

## VBLTX FMAGX SBUX   
## 0.0006903 0.0041077 0.0142545

sigmahat.vals = apply(ret.mat, 2, sd)  
sigmahat.vals

## VBLTX FMAGX SBUX   
## 0.02627 0.06409 0.11939

cov.mat = var(ret.mat)  
cov.mat

## VBLTX FMAGX SBUX  
## VBLTX 0.0006903 0.0001074 -0.0001761  
## FMAGX 0.0001074 0.0041077 0.0032432  
## SBUX -0.0001761 0.0032432 0.0142545

cor.mat = cor(ret.mat)  
cor.mat

## VBLTX FMAGX SBUX  
## VBLTX 1.00000 0.06376 -0.05614  
## FMAGX 0.06376 1.00000 0.42384  
## SBUX -0.05614 0.42384 1.00000

covhat.vals = cov.mat[lower.tri(cov.mat)]  
rhohat.vals = cor.mat[lower.tri(cor.mat)]  
names(covhat.vals) <- names(rhohat.vals) <-   
c("VBLTX,FMAGX","VBLTX,SBUX","FMAGX,SBUX")  
covhat.vals

## VBLTX,FMAGX VBLTX,SBUX FMAGX,SBUX   
## 0.0001074 -0.0001761 0.0032432

rhohat.vals

## VBLTX,FMAGX VBLTX,SBUX FMAGX,SBUX   
## 0.06376 -0.05614 0.42384

cbind(muhat.vals,sigma2hat.vals,sigmahat.vals)

## muhat.vals sigma2hat.vals sigmahat.vals  
## VBLTX 0.005302 0.0006903 0.02627  
## FMAGX 0.001856 0.0041077 0.06409  
## SBUX 0.011318 0.0142545 0.11939

cbind(covhat.vals,rhohat.vals)

## covhat.vals rhohat.vals  
## VBLTX,FMAGX 0.0001074 0.06376  
## VBLTX,SBUX -0.0001761 -0.05614  
## FMAGX,SBUX 0.0032432 0.42384

As we can see, SBUX has the highest mean and SD. (make sense b/c it’s individual stock) VBLTX has the lowest SD b/c it’s a bond. FAMAGX is a portfolio which mean the lowest.

2 b) c)

# compute estimated standard error for mean  
nobs = nrow(ret.mat)  
nobs

## [1] 143

se.muhat = sigmahat.vals/sqrt(nobs)  
se.muhat

## VBLTX FMAGX SBUX   
## 0.002197 0.005360 0.009984

cbind(muhat.vals,se.muhat)

## muhat.vals se.muhat  
## VBLTX 0.005302 0.002197  
## FMAGX 0.001856 0.005360  
## SBUX 0.011318 0.009984

# compute approx 95% confidence intervals  
mu.lower = muhat.vals - 2\*se.muhat  
mu.upper = muhat.vals + 2\*se.muhat  
cbind(mu.lower,mu.upper)

## mu.lower mu.upper  
## VBLTX 0.000908 0.009696  
## FMAGX -0.008863 0.012575  
## SBUX -0.008650 0.031286

# compute estimated standard errors for variance and sd  
se.sigma2hat = sigma2hat.vals/sqrt(nobs/2)  
se.sigma2hat

## VBLTX FMAGX SBUX   
## 8.163e-05 4.858e-04 1.686e-03

se.sigmahat = sigmahat.vals/sqrt(2\*nobs)  
se.sigmahat

## VBLTX FMAGX SBUX   
## 0.001554 0.003790 0.007060

cbind(sigma2hat.vals,se.sigma2hat)

## sigma2hat.vals se.sigma2hat  
## VBLTX 0.0006903 8.163e-05  
## FMAGX 0.0041077 4.858e-04  
## SBUX 0.0142545 1.686e-03

cbind(sigmahat.vals,se.sigmahat)

## sigmahat.vals se.sigmahat  
## VBLTX 0.02627 0.001554  
## FMAGX 0.06409 0.003790  
## SBUX 0.11939 0.007060

# compute approx 95% confidence intervals  
sigma2.lower = sigma2hat.vals - 2\*se.sigma2hat  
sigma2.upper = sigma2hat.vals + 2\*se.sigma2hat  
cbind(sigma2.lower,sigma2.upper)

## sigma2.lower sigma2.upper  
## VBLTX 0.000527 0.0008535  
## FMAGX 0.003136 0.0050793  
## SBUX 0.010883 0.0176260

sigma.lower = sigmahat.vals - 2\*se.sigmahat  
sigma.upper = sigmahat.vals + 2\*se.sigmahat  
cbind(sigma.lower,sigma.upper)

## sigma.lower sigma.upper  
## VBLTX 0.02317 0.02938  
## FMAGX 0.05651 0.07167  
## SBUX 0.10527 0.13351

# compute estimated standard errors for correlation  
se.rhohat = (1-rhohat.vals^2)/sqrt(nobs)  
se.rhohat

## VBLTX,FMAGX VBLTX,SBUX FMAGX,SBUX   
## 0.08328 0.08336 0.06860

cbind(rhohat.vals,se.rhohat)

## rhohat.vals se.rhohat  
## VBLTX,FMAGX 0.06376 0.08328  
## VBLTX,SBUX -0.05614 0.08336  
## FMAGX,SBUX 0.42384 0.06860

# compute approx 95% confidence intervals  
rho.lower = rhohat.vals - 2\*se.rhohat  
rho.upper = rhohat.vals + 2\*se.rhohat  
cbind(rho.lower,rho.upper)

## rho.lower rho.upper  
## VBLTX,FMAGX -0.1028 0.2303  
## VBLTX,SBUX -0.2229 0.1106  
## FMAGX,SBUX 0.2866 0.5610

The variance and SD are estimated more precisely. The mean is not estimated precisely. the cor of VBLTX is not very precise. the cor between FMAGX and SBUX is precise. The 95% for mean of SBUX and FMAGX have negative and positive values and it’s relatively wide, so it’s not good estimate The 95% for SD and var is not very wide, thus it give us more precise estimate The third line in the correkation estimate is narrow.

2 d)

Value.at.Risk = function(x,p=0.05,w=100000) {  
 x = as.matrix(x)  
 q = apply(x, 2, mean) + apply(x, 2, sd)\*qnorm(p)  
 VaR = (exp(q) - 1)\*w  
 VaR  
}  
  
Value.at.Risk(ret.mat,p=0.05,w=100000)

## VBLTX FMAGX SBUX   
## -3720 -9838 -16895

Value.at.Risk(ret.mat,p=0.01,w=100000)

## VBLTX FMAGX SBUX   
## -5429 -13692 -23389

As we can see, the largest VaR is for SBUX( makes sense). VBLTX has the lowest VaR value