**Value at Risk**

indf<- bbts500cos2[,c("amzn","trip","msft","nflx","aapl")]

indf

indf<- na.omit(indf)

indf

**fVar\_custom1**<- function(df,amt,p){

p<- as.vector(p)

**fpctcng**<-function(x,n){

# returns the percentage difference between the current value and the previous value of a vector

numVec<-x[(n+1):length(x)]

denomVec<-x[1:(length(x)-n)]

# numVec/denomVec-1

c(rep(NA,n),numVec/denomVec-1)}

vdf<-df

vdf<-as.data.frame(df)

vdf<- na.omit(vdf)

pordf<-vdf

pordf<- na.omit(pordf)

nSecs<-length(names(pordf))

nSecs

#sapply(pordf,fpctcng,1)

rrdf<- data.frame(sapply(pordf,fpctcng,1))

rrdf<- na.omit(rrdf)

dimrrdf<- dim(rrdf)

mult<- function(x)

{

x<- as.data.frame(x)

return(p\*amt\*x)

}

dollarReturnDF<- as.data.frame(apply(rrdf,1,mult))

#dfdim<- dim(dollarReturnDF)

#dollarReturnVector<- sapply(1:dfdim[1],function(i){sum(dollarReturnDF[i,],na.rm = T)})

dollarReturnVector<- as.vector(apply(dollarReturnDF,2,sum))

#dollarReturnVector == dollarReturVector1

#while(i <= nSecs){

# z =0

# z<- matrix(z,nrow = dim[1], ncol = 1)

# z<- rrdf[,i]\*p[i]\*amt # in millions

#dollarReturnDF= (z+dollarReturnDF)

#i=i+1

#}

#dollarReturnVector<-apply(dollarReturnDF,1,sum,na.rm=T) # in millions

#5% and 1% parametric VAR

p\_var5<-(mean(dollarReturnVector,na.rm=T)-1.65\*sd(dollarReturnVector,na.rm=T))

p\_var1<-(mean(dollarReturnVector,na.rm=T)-2.33\*sd(dollarReturnVector,na.rm=T))

#5% and 1% historical VAR

n<-length(dollarReturnVector)

h\_var5<-sort(dollarReturnVector)[round(n\*.05,0)]

h\_var1<-sort(dollarReturnVector)[round(n\*.01,0)]

varResultsdf1<-data.frame(matrix(2,2))

varResultsdf1[1,1]<-p\_var1

varResultsdf1[1,2]<-p\_var5

varResultsdf1[2,1]<-h\_var1

varResultsdf1[2,2]<-h\_var5

names(varResultsdf1)<- c("1%","5%")

row.names(varResultsdf1)<-c("parametric","historical")

return(varResultsdf1)

p\_var5

}

df

rrdf

d<- c(0.1,0.1,0.1,0.2,0.5)

fVar\_custom1(indf,10,d)

fVar(df)

**Custom Functions**

fskew<-function(x){

# custom skewness function

3\*mean(x,na.rm=T)/sd(x,na.rm=T)}

fkurt<-function(x){NA}

# function to return descriptive statistics on a dataframe

fdesstat<-function(inputDataFrame){

meanVec<-apply(inputDataFrame,2,mean,na.rm=TRUE)

medVec<-apply(inputDataFrame,2,median,na.rm=TRUE)

sdVec<-apply(inputDataFrame,2,sd,na.rm=TRUE)

skewVec<-apply(inputDataFrame,2,fskew)

kurtVec<-apply(inputDataFrame,2,fkurt)

minVec<-apply(inputDataFrame,2,min,na.rm=TRUE)

maxVec<-apply(inputDataFrame,2,max,na.rm=TRUE)

nVec<-apply(inputDataFrame,2,length)

resultsDF<-data.frame(cbind(meanVec,medVec,sdVec,skewVec,kurtVec,minVec,maxVec,nVec))

names(resultsDF)<-c("mean","med","stdev","skew","kurt","min","max","n")

return(resultsDF)}

fplot3<-function(x,y,z){

plot(x,y,type="n"); text(x,y,z,cex=.4)

}

fdiff<-function(x,n){

# creates the difference between the current value and the previous value of a vector

x-c(rep(NA,n),x[1:(length(x)-n)])}

fpctcng<-function(x,n){

# returns the percentage difference between the current value and the previous value of a vector

numVec<-x[(n+1):length(x)]

denomVec<-x[1:(length(x)-n)]

numVec/denomVec-1

c(rep(NA,n),numVec/denomVec-1)}

**Advance Regression**

spDF<- fs1500y

spDF<- fs1500y

spDF<- as.data.frame(spDF)

sort(spDF$tkr)

spDF$nameSht<- substring(spDF$name,1,10)

spDF$nameSht

spDF$year<- substring(spDF$date,1,4)

spDF$year<- as.numeric(spDF$year)

spDF$year

fsdf<- spDF

names(fsdf)

fsdf<- fsdf[fsdf$sector =="Consumer Discretionary" & fsdf$tkr=="PCLN" & fsdf$industry=="Internet & Direct Marketing Retail" & fsdf$io500==1,c("year","nameSht","price","eps","cfps","bvps","tkr")]

dim(fsdf)

fsdf<- na.omit(fsdf)

fsdf

#fsdf<- fsdf[fsdf$io500 ==1,]

#fsdf<- fsdf[fsdf$sector=="Financials",]

#fsdf<- fsdf[fsdf$industry=="Marine",]; dim(fsdf)

dim(fsdf)

#histograms

par(mfcol=c(2,2))

hist(fsdf$price,prob =1,main ="Fig. Hist of Price",xlab = "PRICE"); lines(density(fsdf$price))

hist(fsdf$eps,prob =1,main ="Fig. Hist of EPS",xlab = "EPS"); lines(density(fsdf$eps))

hist(fsdf$cfps,prob =1,main ="Fig. Hist of CFPS",xlab = "CFPS"); lines(density(fsdf$cfps))

hist(fsdf$bvps,prob =1,main ="Fig. Hist of BVPS",xlab = "BVPS"); lines(density(fsdf$bvps))

#scatterplots

par(mfcol=c(2,2))

plot(fsdf$price,fsdf$price)

plot(fsdf$eps,fsdf$price)

plot(fsdf$cfps,fsdf$price)

plot(fsdf$bvps,fsdf$price)

#scatterplots

scatter.smooth(fsdf$eps,fsdf$price,xlab="eps",ylab="price",type = "n")

text(fsdf$eps,fsdf$price,fsdf$year,cex=1)

scatter.smooth(fsdf$cfps,fsdf$price,xlab="cfps",ylab="price",type = "n")

text(fsdf$cfps,fsdf$price,fsdf$year,cex=1)

scatter.smooth(fsdf$bvps,fsdf$price,xlab="bvps",ylab="price",type="n")

text(fsdf$bvps,fsdf$price,fsdf$year,cex=1)

#descriptive statistics

names(fsdf)

desc<- describe(fsdf[,c("price","eps","cfps","bvps")])

#correlation matrix

cor(fsdf[,c("price","eps","cfps","bvps")],use="na.or.complete")

# linear regression

fit<-lm(price~eps+cfps+bvps,data=fsdf,na.action=na.omit)

summary(fit)

#residual dataframe

rdf<-data.frame(fsdf,r=fit$residuals,p=fit$fitted.values)

par(mfcol=c(2,2))

hist(rdf$p,probability = 1); lines(density(fsdf$price)) #hist of residuals

scatter.smooth(rdf$p,rdf$price,xlab="predicted",ylab="price",type="n")

text(rdf$p,rdf$price,rdf$year,cex=1)

scatter.smooth(rdf$eps,rdf$r)

scatter.smooth(rdf$cfps,rdf$r)

scatter.smooth(rdf$bvps,rdf$r)

# gam regression

fit<-gam(price~s(eps,2)+s(cfps,1)+s(bvps,4),na.action=na.omit,data=fsdf)

par(mfcol=c(2,2)); plot.gam(fit)

#residual dataframe

rdf<-data.frame(fsdf,r=fit$residuals,p=fit$fitted.values)

par(mfcol=c(2,2))

hist(fsdf$p,probability = 1); lines(density(fsdf$price)) #hist of residuals

scatter.smooth(rdf$p,rdf$price,xlab="predicted",ylab="price",type="n")

text(rdf$p,rdf$price,rdf$year,cex=.5)

cor(rdf$price,rdf$p)^2

par(mfcol=c(2,2))

scatter.smooth(rdf$eps,rdf$r)

scatter.smooth(rdf$cfps,rdf$r)

scatter.smooth(rdf$bvps,rdf$r)

**Bond Price**

fbondprice1<- function(faceval,cr,ytm,n){

cf<- faceval\*cr

a<- (cf\*(1-(1/(1+ytm)^n)))/ytm

b<- faceval/((1+ytm)^n)

#pvfv<- fpv(faceval,ytm,n)

#pvcf<- fpva(cf,ytm,n)

p<- a+b

return(p)

}