Assigment 1
Mgmt 237E: Empirical Methods
Ian Laker, PrasanthKumar, Nitish Ramkumar

Problem 1

1

We know the jump at time t is $J_t = B_t(\mu_j + \sigma_j\delta_t)$

where δ_t = N(0,1)

and the log returns $r_t = \mu + \sigma\epsilon_t + J_t$

Mean
$\mathbb{E}(J_t) = \mathbb{E}[B_t(\mu_j + \sigma_j\delta_t)] = \mathbb{E}(B_t)\mathbb{E}(\mu_j + \sigma_j\delta_t) = \mathbf{p\mu_j}$

$\mathbb{E}(r_t) = \mathbb{E}(\mu) + \mathbb{E}(\sigma\epsilon_t) + \mathbb{E}(J_t) = \underline{\mu + \mathbf{p\mu_j}}$

Variance
Var(J_t) = $\mathbb{E}(J_t^2)$ - $(\mathbb{E}(J_t))^2$
$= \mathbb{E}({B_t}^2(\mu_j + \sigma_j\delta_t)^2) - {\mu_j}^2p^2$
$= \underline{\mathbb{E}({B_t}^2)\mathbb{E}((\mu_j + \sigma_j\delta_t)^2)} - {\mu_j}^2p^2$

$\underline{\mathbb{E}({B_t}^2)} = Var(B_t) + (E(B_t))^2 = p(1-p) + p^2 = p$
$\underline{\mathbb{E}((\mu_j + \sigma_j\delta_t)^2)} = Var(\mu_j + \sigma_j\delta_t) + (\mathbb{E}(\mu_j + \sigma_j\delta_t))^2 = {\sigma_j}^2 + {\mu_j}^2$

So, Var(J_t) = $\mathbf{p{\sigma_j}^2 + p{\mu_j}^2 - {\mu_j}^2p^2}$

Var(r_t) = Var$(\mu + \sigma\epsilon_t)$ + Var(J_t) = $\underline{\boldsymbol{\sigma^2} + \mathbf{p{\sigma_j}^2 + p{\mu_j}^2 - {\mu_j}^2p^2}}$

Skewness
If Skew in terms of mean and variance = $\frac{\mathbb{E}({J_t}^3) - 3\mathbb{E}(J_t)Var(J_t) - (\mathbb{E}(J_t))^3}{(Var(J_t))^{3/2}}$
Then 3rd moment $\mu_3(J_t) = \mathbb{E}({J_t}^3) - 3\mathbb{E}(J_t)Var(J_t) - (\mathbb{E}(J_t))^3$

$\mathbb{E}({J_t}^3) = \mathbb{E}({B_t}^3(\mu_j + \sigma_j\delta_t)^3) = \underline{\mathbb{E}({B_t}^3)\mathbb{E}((\mu_j + \sigma_j\delta_t)^3)}$

We can find $\underline{\mathbb{E}({B_t}^3)}$ by using bernoulli distribution mean, variance and skew($\frac{1-2p}{\sqrt{p(1-p)}}$). Substitute in skew formula in terms of mean and variance
$\mathbb{E}({B_t}^3) = \frac{(1-2p)(p(1-p))^{3/2}}{\sqrt{p(1-p)}} + 3p^2(1-p) - p^3$
$= (1-2p)(p-p^2) + 3p^2 - 3p^3 + p^3 = \mathbf{p}$

We can find $\underline{\mathbb{E}((\mu_j + \sigma_j\delta_t)^3)}$ by using normal distribution mean, variance and skew (0). Substitute in skew formula in terms of mean and variance

$\mathbb{E}((\mu_j + \sigma_j\delta_t)^3) = \mathbf{3\mu_j{\sigma_j}^2 + {\mu_j}^3}$

so $\mathbb{E}({J_t}^3) = \mathbf{p(3\mu_j{\sigma_j}^2 + {\mu_j}^3})$

so $\mu_3(J_t) = \mathbf{3\mu_j{\sigma_j}^2p + {\mu_j}^3p - 3p^2\mu_j{\sigma_j}^2 - 3p^2{\mu_j}^3 + 2p^3{\mu_j}^3}$

Skew(r_t) = Skew($\mu + \sigma\epsilon_t + J_t$)

We know, Skew(A + B) = $\frac{\mu_3(A)+\mu_3(B)}{(Var(A)+Var(B))^{3/2}}$,

Substitute A = $\mu + \sigma\epsilon_t$ and B = J_t

Skew(r_t) = $\frac{0+\mu_3(B)}{\sigma^2+p{\sigma_j}^2+p{\mu_j}^2-{\mu_j}^2p^2)^{3/2}}$

$$= \frac{\boldsymbol{\frac{3\mu_j{\sigma_j}^2p+{\mu_j}^3p-3p^2\mu_j{\sigma_j}^2-3p^2{\mu_j}^3+2p^3{\mu_j}^3}{(\sigma^2+p{\sigma_j}^2+p{\mu_j}^2-{\mu_j}^2p^2)^{3/2}}}}{}$$

2

The general stock returns are distributed with skew and thick tails. A log normal distribution doesn't cover the skew and the thick tails which are necessary to represent general stock returns.

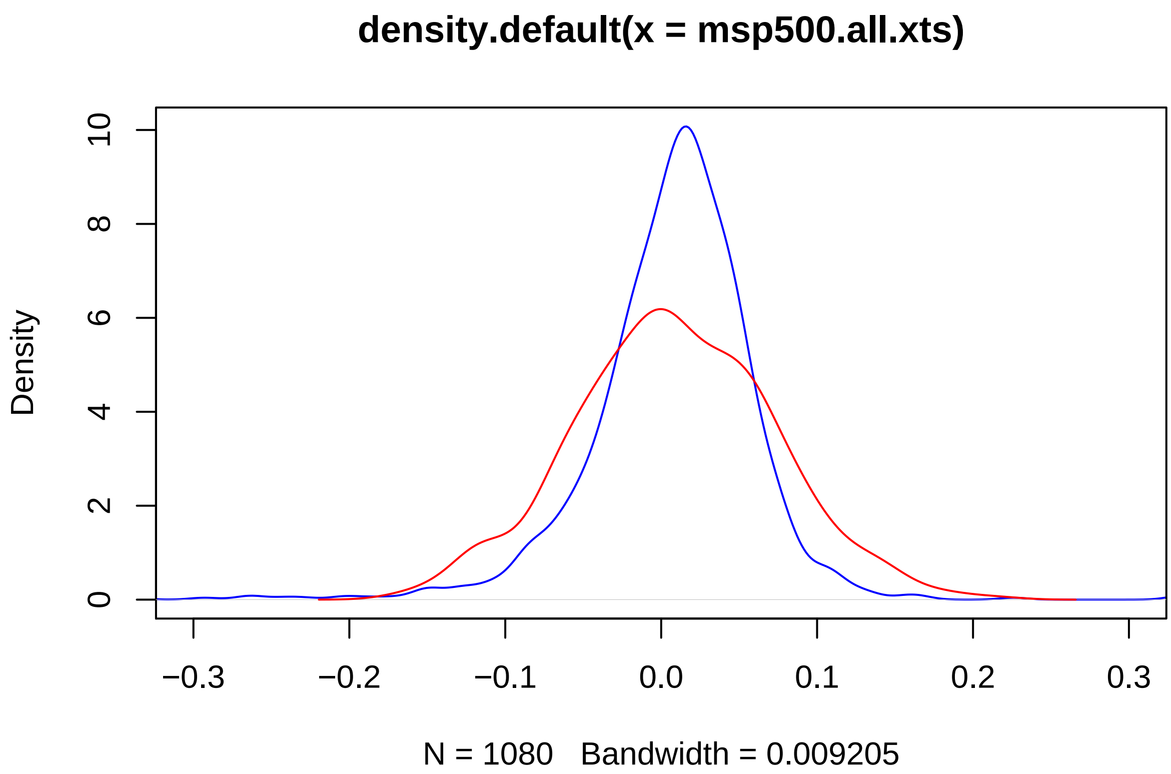
3

Get the SP500 monthly data to use as baseline

```
# Function to get data #
getData <- function(sql, n = -1){
  #setup connection
  res <- dbSendQuery(wrds, sql)
  dbHasCompleted(res)

  #perform fetch
  returnData <- fetch(res, n)

  #clear memory
  dbClearResult(res)
  return(returnData)
}


sql3 <- "SELECT * FROM CRSP.MSP500"
msp500.all <- getData(sql3)
msp500.all$caldt <- as.Date(msp500.all$caldt)
msp500.all.xts <- xts::xts(log(1+msp500.all$vwretd[-1]),order.by = msp500.all$caldt[-1])
colnames(msp500.all.xts) <- "vwretd"
```

If the returns are normally distributed, the distribution will look like this compared to the monthly SP returns.

```
plot(density(msp500.all.xts),col="blue", type="l",xlim=c(-0.3,0.3))

#only normal
lines(density(rnorm(n = 600,mean = 0.008,sd = 0.063)),col="red",type="l")
```



If we take the bernoulli-normal mix, the distribution will look like this compared to the monthly SP returns.

```
#normalBernoullimix
normalBernoulliMix <- function(normalMean,normalSD,bernProb,jumpMean,jumpSD,n)
{
  SecondTerm <- jumpMean + jumpSD*rnorm(n)
  jt <- rbinom(n,1,bernProb)*(SecondTerm)
  normalMean + normalSD * rnorm(n) + jt
}

normalBernoulliDist <- normalBernoulliMix(0.012,0.05,0.15,-0.03,0.1,600)

#mean, SD, skewness, Kurtosis
mean(normalBernoulliDist)
```

```
## [1] 0.007948848
```

```
sd(normalBernoulliDist)
```

```
## [1] 0.0623559
```

```
library(moments)
```

```
## Warning: package 'moments' was built under R version 3.3.2
```

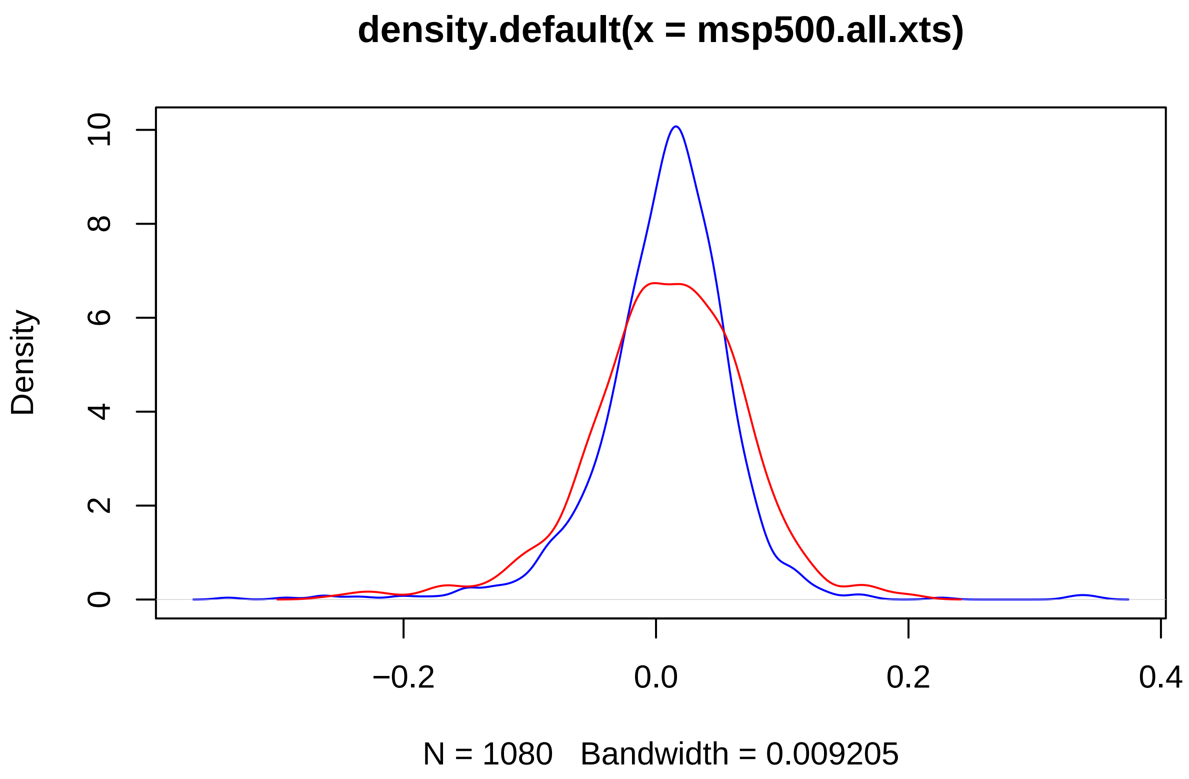
```r
skewness(normalBernoulliDist)
```

```
## [1] -0.5621593
```

```r
kurtosis(normalBernoulliDist)
```

```
## [1] 4.771089
```

```r
#plot sp500
plot(density(msp500.all.xts),col="blue", type="l")

#density
lines(density(normalBernoulliDist),type="l",col="red", ylim=c(0,9))

#actual values
lines(normalBernoulliDist,type="l",col="blue")
```



Problem 2

Get the necessary data

```r
#this.dir <- dirname(rstudioapi::getActiveDocumentContext()$path)
setwd("C:/_UCLA/237E_Empirical/Assigments/Assignment1")
library("readxl")
```

```
## Warning: package 'readxl' was built under R version 3.3.2
```

```r
library("xts")
```

```
## Warning: package 'xts' was built under R version 3.3.2
```

```
## Loading required package: zoo
```

```
## Warning: package 'zoo' was built under R version 3.3.2
```

```
##
## Attaching package: 'zoo'
```

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

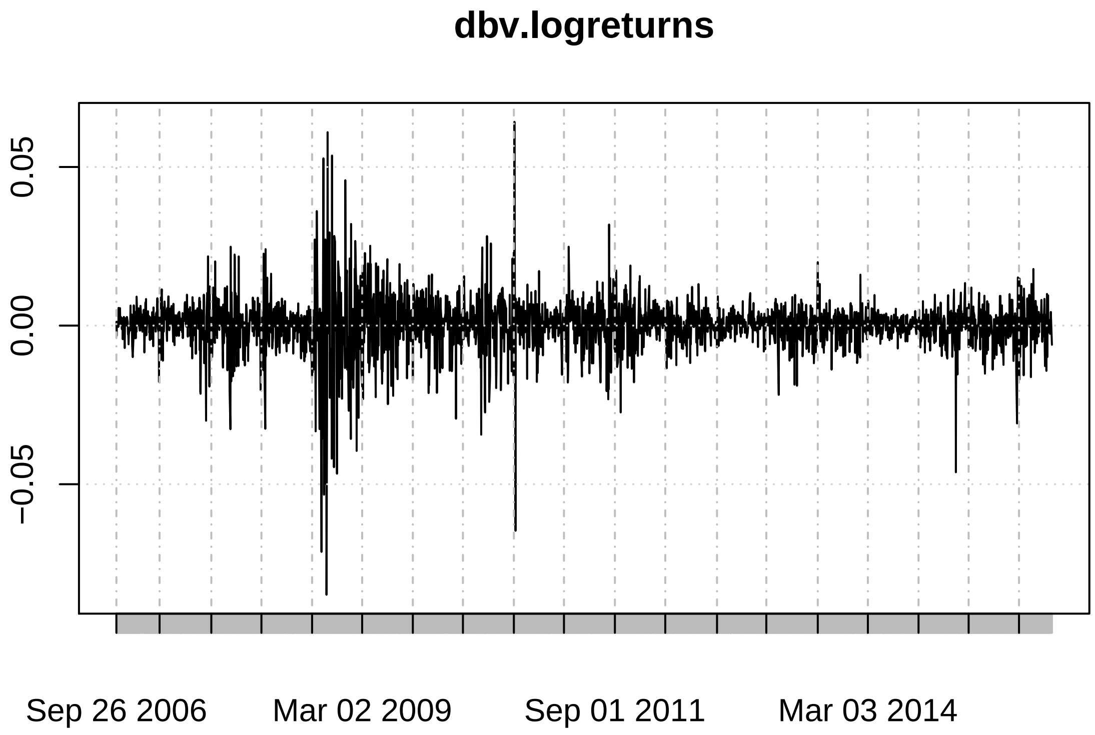
```r
dbv <- read_excel("DBV.xlsx")
gspc <- read_excel("GSPC.xlsx")

dbv$Date <- as.Date(dbv$Date)
gspc$Date <- as.Date(gspc$Date)
dbv.xts <- xts(dbv[,-1],order.by = dbv$Date)
gspc.xts <- xts(gspc[,-1],order.by = gspc$Date)
```

Calculate the log returns

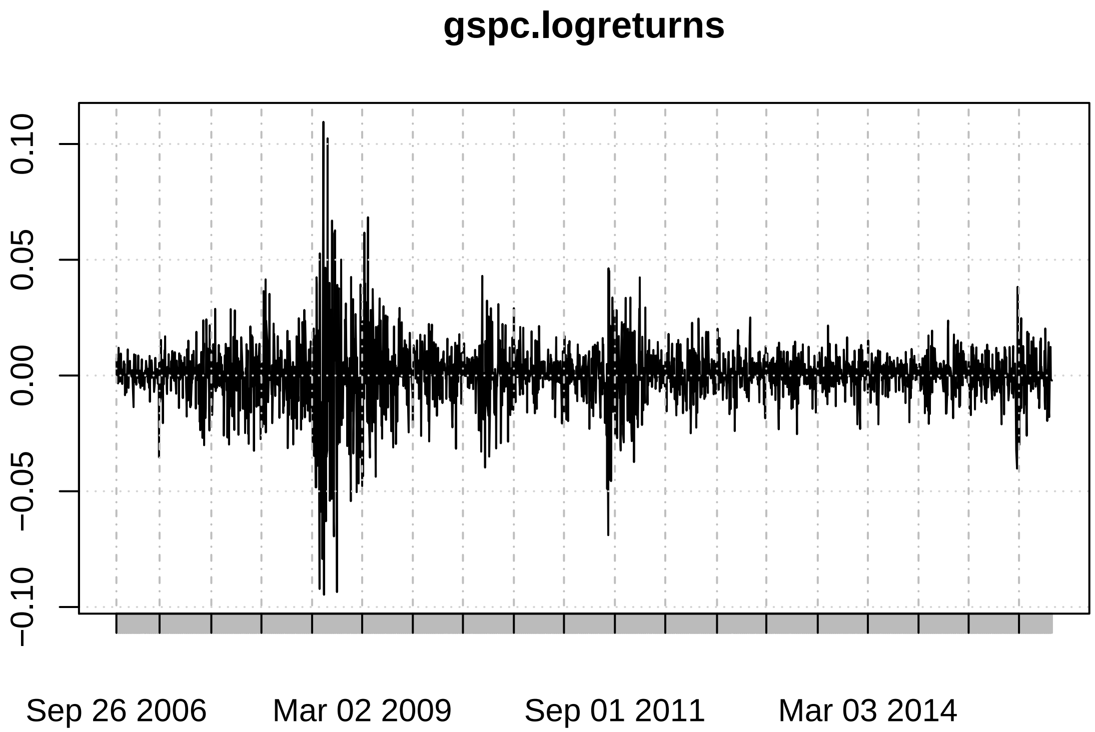
```r
dbv.logreturns <- log(dbv.xts$'Adj Close'[-1,]/lag(dbv.xts$'Adj Close')[-1,])
gspc.logreturns <- log(gspc.xts$'Adj Close'[-1,]/lag(gspc.xts$'Adj Close')[-1,])
```

1

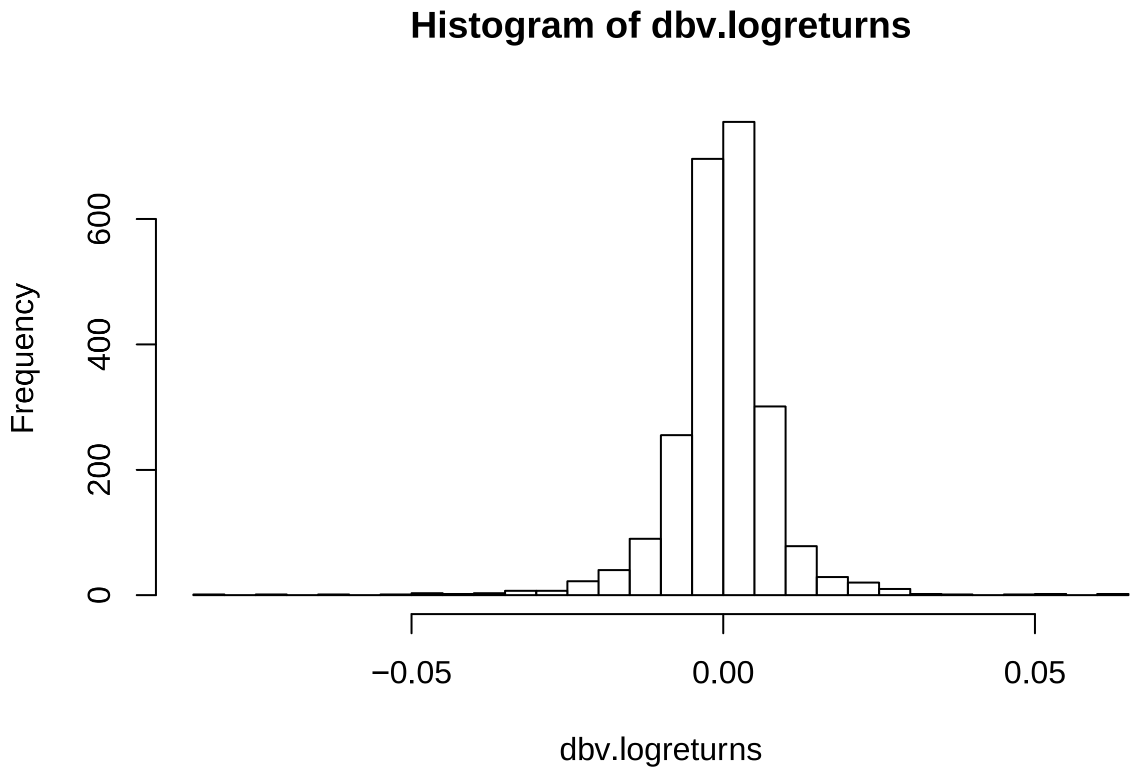
```r
#1
plot(dbv.logreturns)
```



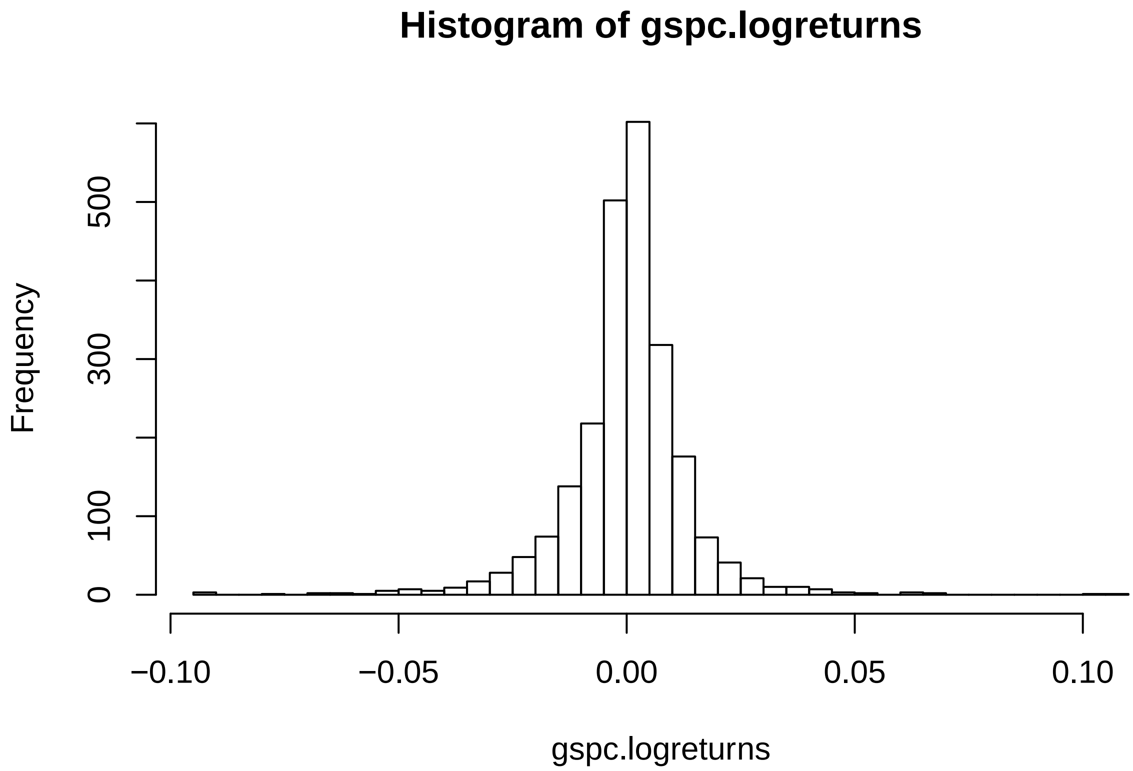
```
plot(gspc.logreturns)
```



```r
hist(dbv.logreturns,breaks = 50)
```



```r
hist(gspc.logreturns, breaks = 50)
```



2

a

```
skewNullCheck <- function(returns,alpha=.05){
  criticalt <- qt(1-(alpha/2),df = length(returns))
  skewcap <- skewness(returns)
  skewt <- skewcap/(sqrt(6/length(returns)))
  returnVal <- c(skewcap-3,skewt,abs(skewt) > criticalt) #TRUE, so reject normal distribution, no skewn
  names(returnVal) <- c("Sample Skewness","Skewness t","Reject Null?")
  returnVal
}

skewNullCheck(dbv.logreturns)
```

```
## Sample Skewness      Skewness t    Reject Null?
##       -3.847038      -16.691872        1.000000
```

```
skewNullCheck(gspc.logreturns)
```

```
## Sample Skewness      Skewness t    Reject Null?
##       -3.324043       -6.385641        1.000000
```

```
#b
kurtosisNullCheck <- function(returns,alpha=.05){
  criticalt <- qt(1-(alpha/2),df = length(returns))
  kurtosiscap <- kurtosis(returns)
  kurtosist <- (kurtosiscap-3)/(sqrt(24/length(returns)))
  returnVal <- c(kurtosiscap-3,kurtosist,abs(kurtosist) > criticalt) #TRUE, so reject normal distributi
  names(returnVal) <- c("Sample Kurtosis","Kurtosis t","Reject Null?")
  returnVal
}

kurtosisNullCheck(dbv.logreturns)
```

```
## Sample Kurtosis      Kurtosis t    Reject Null?
##        13.51674       133.18159         1.00000
```

```
kurtosisNullCheck(gspc.logreturns)
```

```
## Sample Kurtosis      Kurtosis t    Reject Null?
##        9.797934       96.539900        1.000000
```

```
#c
jbTest <- function(returns,alpha=.05){
  criticalchi <- qchisq(1-alpha,df = 2)
  skewcap <- skewness(returns)
  kurtosiscap <- kurtosis(returns)
  jbt <- skewcap^2/(6/length(returns)) + (kurtosiscap-3)^2/(24/length(returns))
  returnVal <- c(jbt,abs(jbt) > criticalchi)
  names(returnVal) <- c("JBt","Reject Null?")
  returnVal
}

jbTest(dbv.logreturns)
```

```
##          JBt Reject Null?
##     18015.95         1.00
```

```
jbTest(gspc.logreturns)
```

```
##          JBt Reject Null?
##     9360.729        1.000
```

3

```
sqlDaily <- "SELECT caldt, vwretd FROM CRSPQ.DSP500 WHERE year(caldt) >= 1973 AND year(caldt) < 2015"
dsp500 <- getData(sqlDaily)

skewKurtMat <- matrix(c(skewNullCheck(dbv.logreturns)[1],skewNullCheck(gspc.logreturns)[1],kurtosisNull(
colnames(skewKurtMat) <- c("Skewness","Kurtosis")
row.names(skewKurtMat) <- c("DBV","GSPC")
skewKurtMat
```

```
##       Skewness  Kurtosis
## DBV  -3.847038 13.516736
## GSPC -3.324043  9.797934
```

4

The expected return to standard deviation ratio covers only the first 2 moments of the return. It doesn't show the difference in skewness and kurtosis between the two investments.
As GSPC has lower negative skewness, there is lesser chance of getting a negative tail value.
It also has lower kurtosis, which means there is lesser chance of getting an extreme value

5

a

Both assumptions valid (i.e. homoskedastic and normal)

```
lm <- lm(dbv.logreturns ~ gspc.logreturns)
reg1Summ <- summary(lm)
reg1Summ$coefficients[,2]
```

```
##     (Intercept) gspc.logreturns
##    0.0001361433    0.0101101940
```

```
vcov(lm)
```

```
##                   (Intercept) gspc.logreturns
## (Intercept)      1.853500e-08   -1.923941e-08
## gspc.logreturns -1.923941e-08    1.022160e-04
```

```
library(DataAnalytics)
reg2Summ <- lmSumm(lm(dbv.logreturns ~ gspc.logreturns),HAC = T)
```

```
## Multiple Regression Analysis:
##     2 regressors(including intercept) and 2330 observations
##     with heteroskedastic|autocorrelation consistent standard errors
##     Lag truncation = 0
##
## lm(formula = dbv.logreturns ~ gspc.logreturns)
##
## Coefficients:
##                   Estimate Std Error t value p value
## (Intercept)     -9.579e-05  0.000137   -0.70   0.484
## gspc.logreturns  4.309e-01  0.018280   23.57   0.000
## ---
## Standard Error of the Regression:  0.006571
## Multiple R-squared:  0.438  Adjusted R-squared:  0.438
```

```
reg2Summ$coef.table[,2]
```

```
##     (Intercept) gspc.logreturns
##        0.000137        0.018280
```