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

Problem 1

1

Let us generate the moment generating function for r_t

Moment generating function Mg(s) = $\mathbb{E}(expr_t s)$

$= \mathbb{E}(e^{[\mu+\sigma\epsilon_t+\beta_t(\mu_j+\sigma_j\delta_t)]s})$

$= \mathbb{E}(e^{(\mu+\sigma\epsilon_t)s}e^{(\beta_t(\mu_j+\sigma_j\delta_t))s})$

$= \mathbb{E}(e^{(\mu+\sigma\epsilon_t)s})\mathbb{E}(e^{\beta_t(\mu_j+\sigma_j\delta_t)s})$

$= e^{s\mu+\frac{1}{2}\sigma^2s^2}(p\mathbb{E}(e^{(\mu_j+\sigma_j\delta_t)s}) + (1-p)e^0)$

$= \mathbf{e^{s\mu+\frac{1}{2}\sigma^2s^2}[pe^{s\mu_j+\frac{1}{2}{\sigma_j}^2s^2} + (1-p)]}$

Mean

Lets keep A = $s\mu + \frac{1}{2}\sigma^2s^2 + s\mu_j + \frac{1}{2}{\sigma_j}^2s^2$, B = $s\mu + \frac{1}{2}\sigma^2s^2$,

so Mg(s) = $pe^A + (1-p)e^B$

We know Mean = 1st Moment (M_1) = $\frac{\partial Mg}{\partial s}(s=0)$

$\frac{\partial Mg}{\partial s} = pe^A\frac{\partial A}{\partial s} + (1-p)e^B\frac{\partial B}{\partial s}$

$\frac{\partial A}{\partial s} = \mu + \sigma^2s + \mu_j + {\sigma_j}^2s$
$\frac{\partial B}{\partial s} = \mu + \sigma^2s$

So, $\frac{\partial A}{\partial s}(s=0) = \mu + \mu_j$, $\frac{\partial B}{\partial s}(s=0) = \mu$

$Mean(\mu_{r_t}) = M_1 = \frac{\partial Mg}{\partial s}(s=0) = \mu + \mathbf{p\mu_j}$

Variance

We know Variance = M_2 - ${\mu_{r_t}}^2$

$M_2 = \frac{\partial^2 Mg}{\partial s^2}(s=0)$

$\frac{\partial^2 Mg}{\partial s^2} = pe^A{\frac{\partial A}{\partial s}}^2 + pe^A\frac{\partial^2 A}{\partial s^2} + e^B{\frac{\partial B}{\partial s}}^2 + e^B\frac{\partial^2 B}{\partial s^2} - pe^A{\frac{\partial B}{\partial s}}^2 - pe^B\frac{\partial^2 B}{\partial s^2}$

So $\frac{\partial^2 A}{\partial s^2} = \sigma^2 + {\sigma_j}^2$, $\frac{\partial^2 B}{\partial s^2} = \sigma^2$

So, $M_2 = p{\mu_j}^2 + 2p\mu\mu_j + p\sigma^2 + p{\sigma_j}^2 + \mu^2 + \sigma^2$

Variance $({\sigma_{r_t}}^2) = \mathbf{\sigma^2 + p{\sigma_j}^2 + p{\mu_j}^2 - p^2{\mu_j}^2}$

Skewness

We know Skewness = $\frac{M_3 - 3\mu_{r_t}{\sigma_{r_t}}^2 - {\mu_{r_t}}^3}{{\sigma_{r_t}}^{\frac{3}{2}}}$

$M_3 = \frac{\partial^3 Mg}{\partial s^3}$

$\frac{\partial^3 Mg}{\partial s^3} = pe^A{\frac{\partial A}{\partial s}}^3 + 3pe^A\frac{\partial A}{\partial s}\frac{\partial^2 A}{\partial s^2} + pe^A\frac{\partial^3 A}{\partial s^3} + e^B{\frac{\partial B}{\partial s}}^3 + 3e^B\frac{\partial B}{\partial s}\frac{\partial^2 B}{\partial s^2} + e^B\frac{\partial^3 B}{\partial s^3} - pe^B(\frac{\partial B}{\partial s})^3 - 3pe^B\frac{\partial B}{\partial s}\frac{\partial^2 B}{\partial s^2} - pe^B\frac{\partial^3 B}{\partial s^3}$

We can calculate $\frac{\partial^3 A}{\partial s^3} = 0$, $\frac{\partial^3 B}{\partial s^3} = 0$

$M_3 = p(\mu+\mu_j)^3 + 3p(\mu+\mu_j)(\sigma^2+{\sigma_j}^2) + (1-p)(\mu^3 + 3\mu\sigma^2)$

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

$= \frac{\mu_j\mathbf{p}(\mu_j{}^2(\mathbf{2p^2-3p+1})-\mathbf{3}\sigma_j{}^2(\mathbf{p-1}))}{(\sigma^2+\mathbf{p}\sigma_j{}^2+\mathbf{p}\mu_j{}^2-\mathbf{p^2}\mu_j{}^2)^{\frac{3}{2}}}$

Kurtosis

We know Kurtosis $= \frac{M_4-4\mu_{r_t}M_3+6{\mu_{r_t}}^2M_2-3{\mu_{r_t}}^4}{{\sigma_{r_t}}^2}$

$M_4 = \frac{\partial^4 Mg}{\partial s^4}$

$\frac{\partial^4 Mg}{\partial s^4} = pe^A\frac{\partial^1 Mg}{\partial s^1}^4 + 6pe^A\frac{\partial^1 Mg}{\partial s^1}^2\frac{\partial^2 Mg}{\partial s^2} + 4pe^A\frac{\partial^1 Mg}{\partial s^1}\frac{\partial^3 Mg}{\partial s^3} + 3pe^A\frac{\partial^2 Mg}{\partial s^2}^2 + pe^A\frac{\partial^4 Mg}{\partial s^4}$

We can calculate, $\frac{\partial^4 A}{\partial s^4} = 0, \frac{\partial^4 B}{\partial s^4} = 0$

$M_4 = p(\mu+\mu_j)^4 + 6p(\mu+\mu_j)^2(\sigma^2+{\sigma_j}^2) + 3p(\sigma^2+{\sigma_j}^2)^2 + (1-p)(\mu^4+6\mu^2\sigma^2+3\sigma^3)$

So Kurtosis $= \frac{\mathbf{3}\sigma_j{}^4\mathbf{p(1-p)}+\mu_j{}^4\mathbf{p}(\mathbf{-6p^3+12p^2-7p+1})+\mu_j{}^2\sigma_j{}^2\mathbf{p(12p^2-18p+6)}}{(\sigma^2+\mathbf{p}\sigma_j{}^2+\mathbf{p}\mu_j{}^2-\mathbf{p^2}\mu_j{}^2)^2}$

2

We observed that the daily and monthly log stock market returns are negatively skewed and have fat tails compared to a normal. This Bernoulli-normal mixture model can take care of the skewness and fat tails of the log returns by appropriately choosing the parameters $\sigma, \mu, \sigma_j, \mu_j$ since the skewness and excess kurtosis of this distribution is a function of these parameters as obtained in part1.

3

Lets use WRDS to get the SP500 monthly data which can be used as baseline

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

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

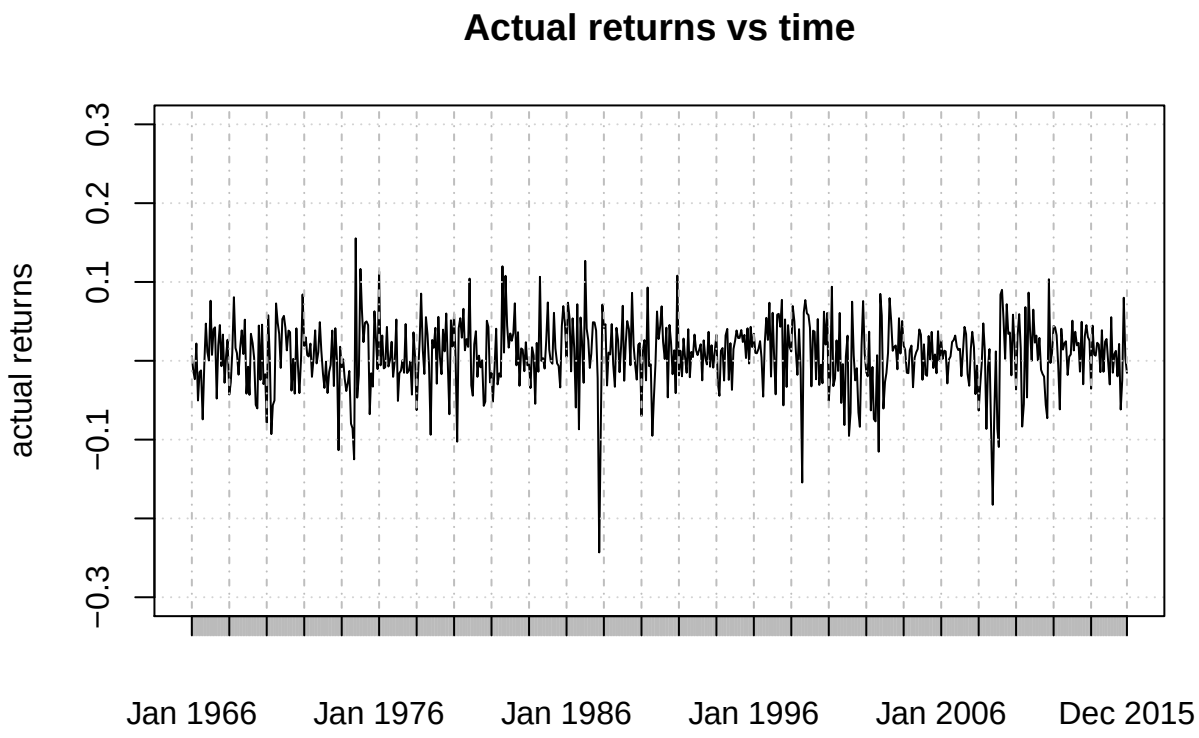
```
## Warning: package 'zoo' was built under R version 3.3.2
```

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

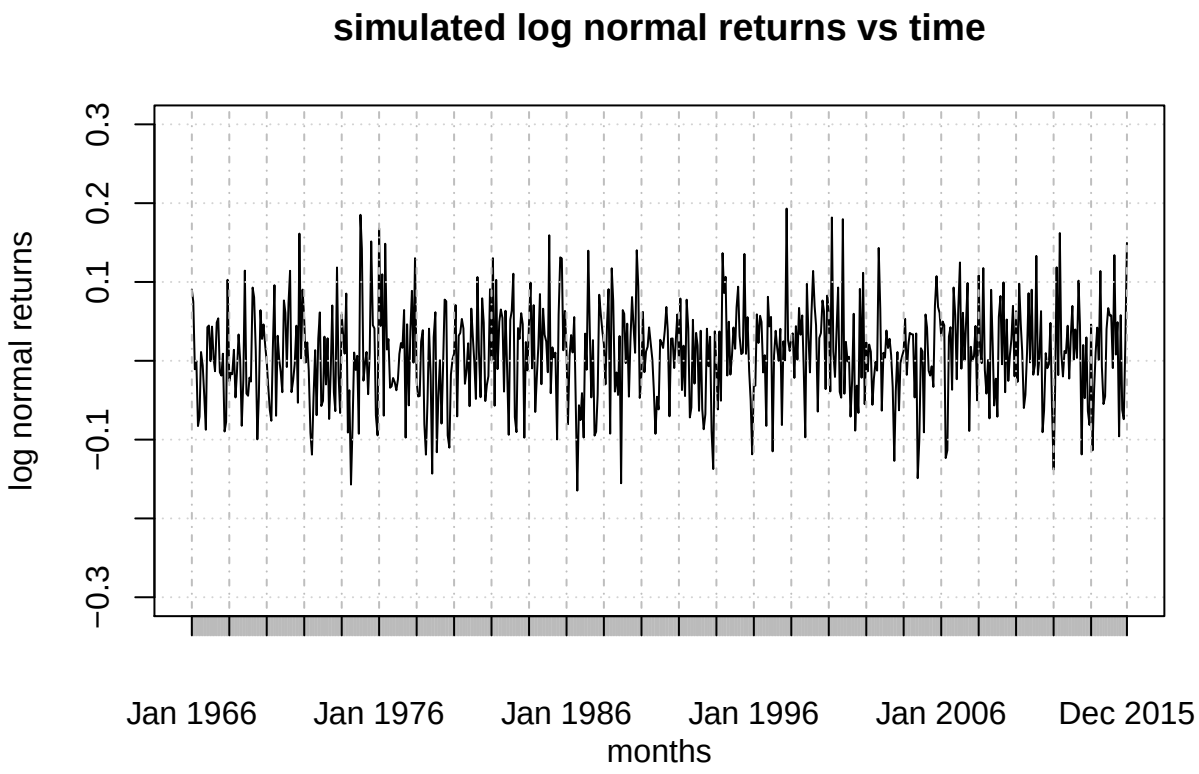
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

i.i.d and normally distributed

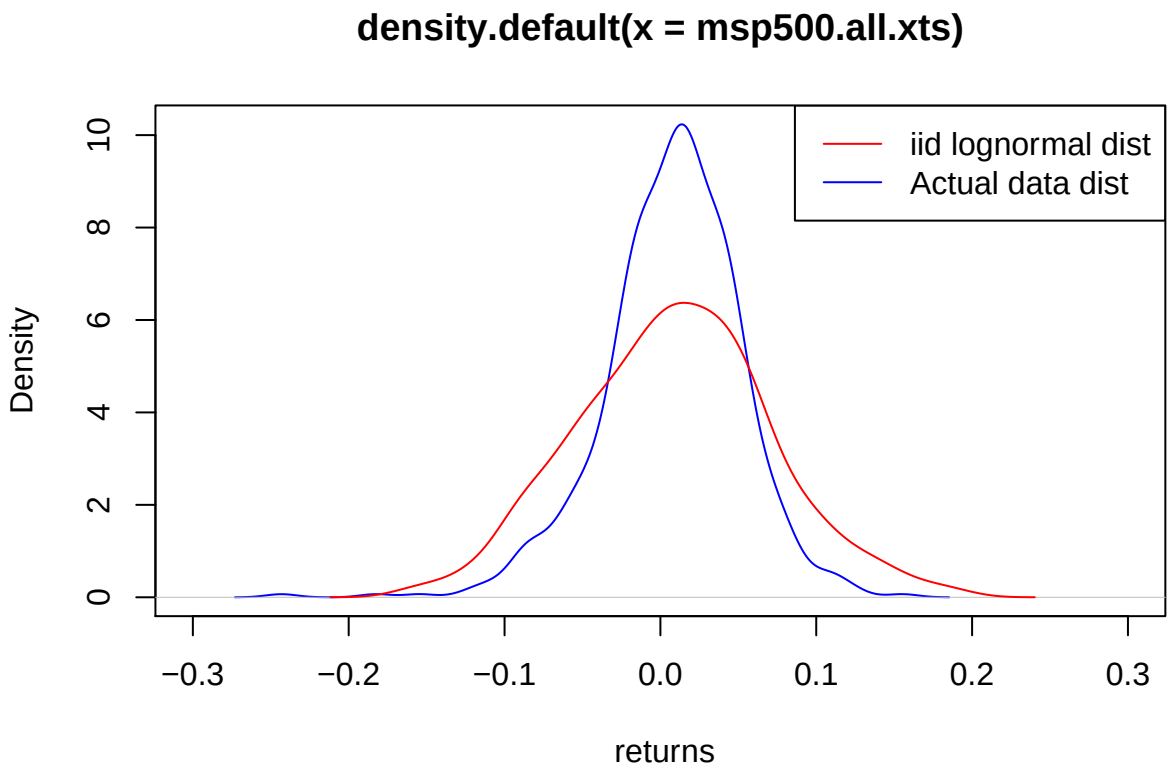
```
#plotting ACTUAL DATE vs TIME
plot(msp500.all.xts,ylab="actual returns",main="Actual returns vs time",ylim=c(-0.3,0.3))
```




```r
#simulating iid log normal returns
logret_normal=rnorm(n = 600,mean = 0.008,sd = 0.063)
logret_normal = xts(logret_normal,order.by = as.Date(index(msp500.all.xts)))
#plotting LOG NORMAL RETURNS vs TIME
plot(logret_normal,ylab="log normal returns",xlab="months",main="simulated log normal returns vs time",
```




```
#plotting distributions of simulated log normal returns and the actual data
plot(density(msp500.all.xts),col="blue", type="l",xlim=c(-0.3,0.3),xlab=" returns",ylab = "Density")
lines(density(logret_normal),col="red")
legend("topright",c("iid lognormal dist","Actual data dist"),col=c("red","blue"),lty = 1)
lines(x=0.008,y = 10)
```




```
library(moments)
```

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

```
stats1 <- matrix(nrow=4,ncol=2)
stats1[1,1] <- mean(msp500.all.xts)
stats1[1,2] <- mean(logret_normal)
stats1[2,1] <- sd(msp500.all.xts)
stats1[2,2] <- sd(logret_normal)
stats1[3,1] <- skewness(msp500.all.xts)
stats1[3,2] <- skewness(logret_normal)
stats1[4,1] <- kurtosis(msp500.all.xts)
stats1[4,2] <- kurtosis(logret_normal)
colnames(stats1) <- c("SP500 Log Returns","Normal Log Returns")
row.names(stats1) <- c("Mean","Standard Deviation","Skewness","Kurtosis")
stats1
```

```
##                    SP500 Log Returns Normal Log Returns
## Mean                       0.007760569        0.008546028
## Standard Deviation         0.043790891        0.062655367
## Skewness                  -0.645904943        0.059301979
## Kurtosis                   5.483924103        2.978619510
```

We can observe that the variance of the last 50 years of log returns from SP500 doesn't match the variance of the normal log distribution. But we can notice the fat left tail (kurtosis) of the SP500 log returns, which is not represeneted by the iid log normal distribution.

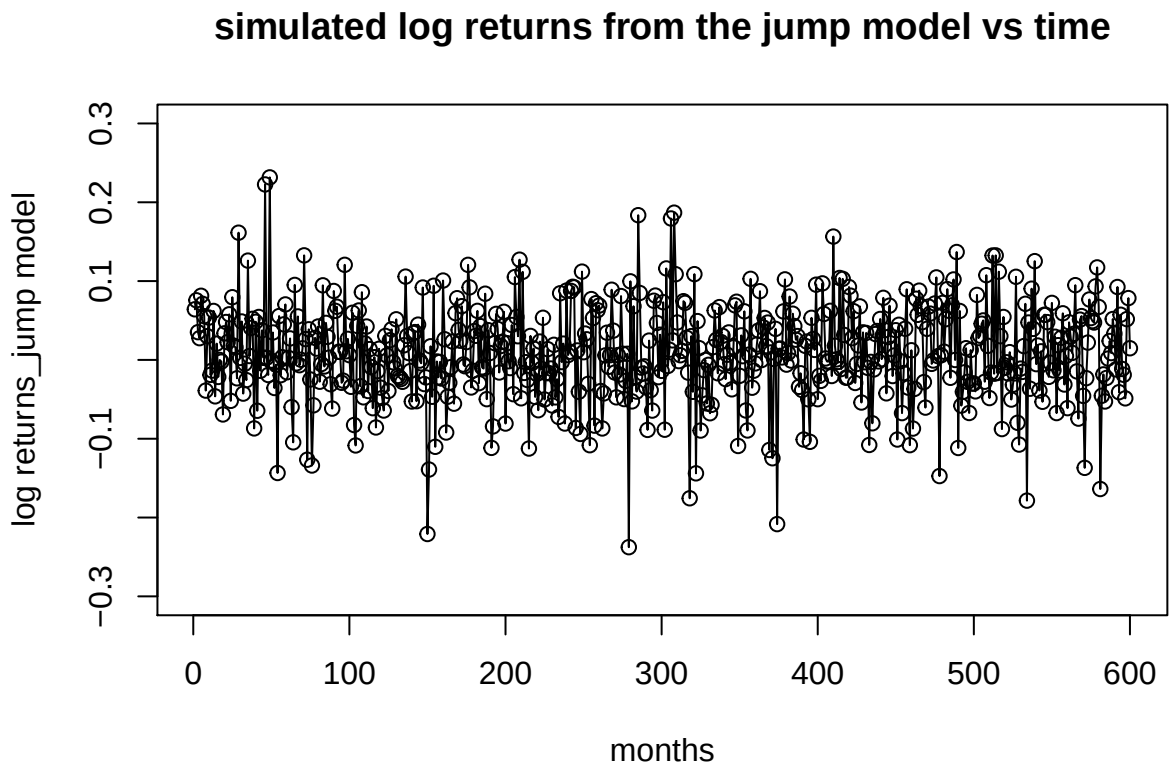
Also, we would expect the SP500 log returns to have skewness as compared to 0 skewness (almost) in a log normal distribution. This shows that failure of log normal distribution to explain the skewness of the actual data.

Bernoulli-normal mixture model

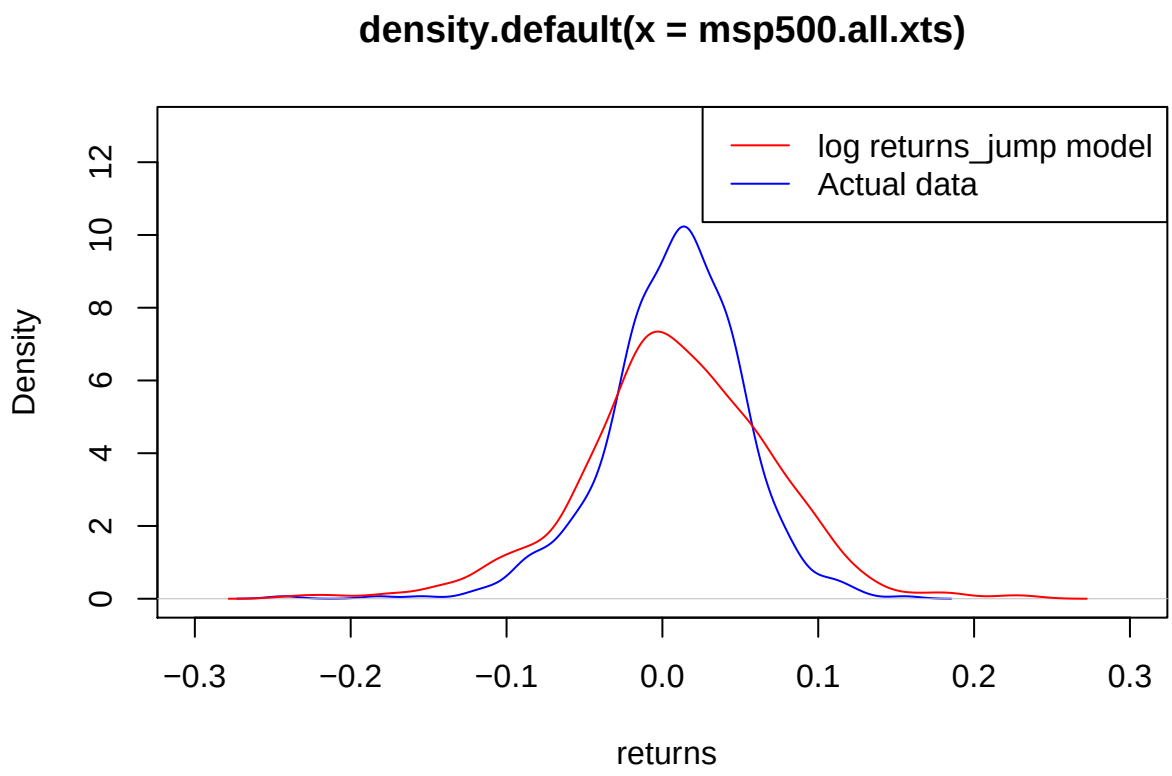
```
#normalBernoullimix
normalBernoulliMix <- function(normalMean,normalSD,bernProb,jumpMean,jumpSD,n)
{
  SecondTerm <- jumpMean + jumpSD*rnorm(n)
  jt <- rbinom(n,1,bernProb)*(SecondTerm)
  normalMean + normalSD * rnorm(n) + jt
}

#mean, SD, skewness, Kurtosis
m=0.012
sig=0.05
p=0.15
mj=-0.03
sigj=0.1
normalBernoulliDist <- normalBernoulliMix(m,sig,p,mj,sigj,600)

#plotting log returns from the jump model over time
plot(normalBernoulliDist,ylab="log returns_jump model",xlab="months",main="simulated log returns from t
lines(normalBernoulliDist)
```



```r
#plotting distributions of simulated log returns from the jump model and the actual data
plot(density(msp500.all.xts),col="blue", type="l",xlim=c(-0.3,0.3),xlab=" returns",ylab = "Density",yli
lines(density(normalBernoulliDist),col="red")
legend("topright",c("log returns_jump model","Actual data"),col=c("red","blue"),lty = 1)
```




```r
stats2 <- matrix(nrow=4,ncol=2)
stats2[1,1] <- mean(msp500.all.xts)
stats2[1,2] <- mean(normalBernoulliDist)
stats2[2,1] <- sd(msp500.all.xts)
stats2[2,2] <- sd(normalBernoulliDist)
stats2[3,1] <- skewness(msp500.all.xts)
stats2[3,2] <- skewness(normalBernoulliDist)
stats2[4,1] <- kurtosis(msp500.all.xts)
stats2[4,2] <- kurtosis(normalBernoulliDist)
colnames(stats2) <- c("SP500 Log Returns","Bernoulli Normal Mix Model")
row.names(stats2) <- c("Mean","Standard Deviation","Skewness","Kurtosis")
stats2
```

```
##                    SP500 Log Returns Bernoulli Normal Mix Model
## Mean                       0.007760569                0.008799751
## Standard Deviation         0.043790891                0.061097528
## Skewness                  -0.645904943               -0.224854184
## Kurtosis                   5.483924103                4.248106284
```

The SP500 data doesn't have exactly the same statistical parameters as compared to the bernoulli normal

mix model. But it can be clearly observed that the bernoulli normal mix has non-zero skewness and excess kurtosis which is similar to the SP500 log returns data. This wasn't explained by the log normal model.

The unconditional mean, standard deviation, skewness and kurtosis are as below:

```
mean_jumpmodel= m + p*mj
variance_jumpmodel = sig^2 + p*(sigj^2) + p*(mj^2) - (p^2)*(mj^2)
sd_jumpmodel = sqrt(variance_jumpmodel)

skewness_jumpmodel= (p*mj*(mj^2*(2*p^2 - 3*p + 1) - 3*sigj^2*(p-1)))/(variance_jumpmodel)^(3/2)
kurtosis_jumpmodel = (3*sigj^4*p*(1-p) + mj^4*p*(-6*p^3 + 12*p^2 - 7*p + 1) + mj^2*sigj^2*p*(12*p^2 - 18

statsJump <- c(mean_jumpmodel,sd_jumpmodel,skewness_jumpmodel,kurtosis_jumpmodel)
names(statsJump) <- c("Mean","Sd","Skewness","Kurtosis")
statsJump
```

```
##        Mean          Sd    Skewness    Kurtosis
##  0.00750000  0.06414632 -0.44387763  2.54523319
```

One aspect which is not explained by the bernoulli normal mix model is the difference in kurtosis of the two tails involved. the negative tail has more risk consequences, which should be modelled.

Problem 2

Firstly get the necessary data

```
setwd("C:/_UCLA/237E_Empirical/Assigments/Assignment1")
library("readxl")
```

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

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

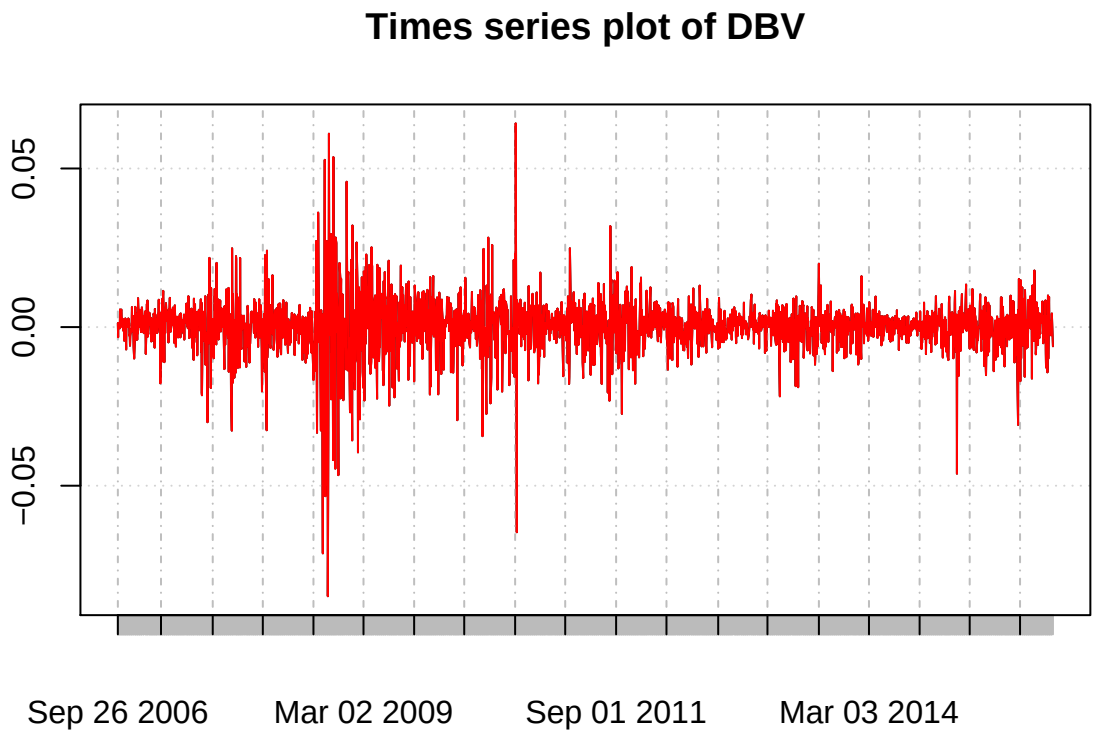
Calculate the log returns

```
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,])
```

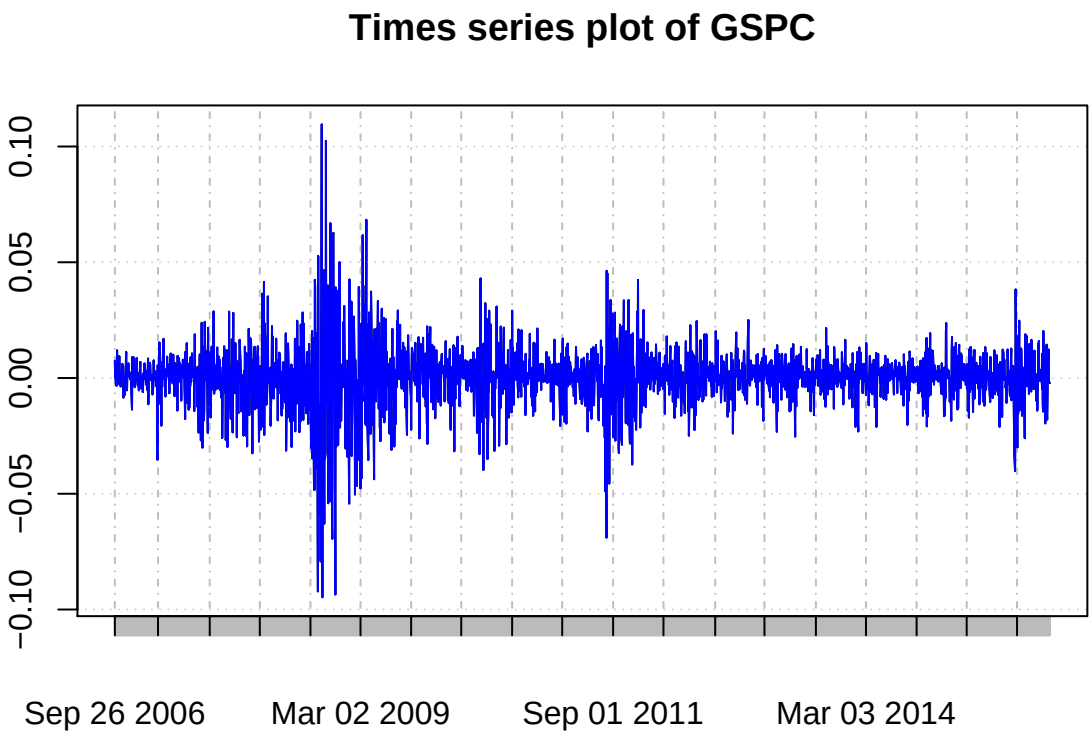
1

Time series for DBV and GSPC are below

```r
#1
plot(dbv.logreturns, main = "Times series plot of DBV")
lines(dbv.logreturns,col="red")
```

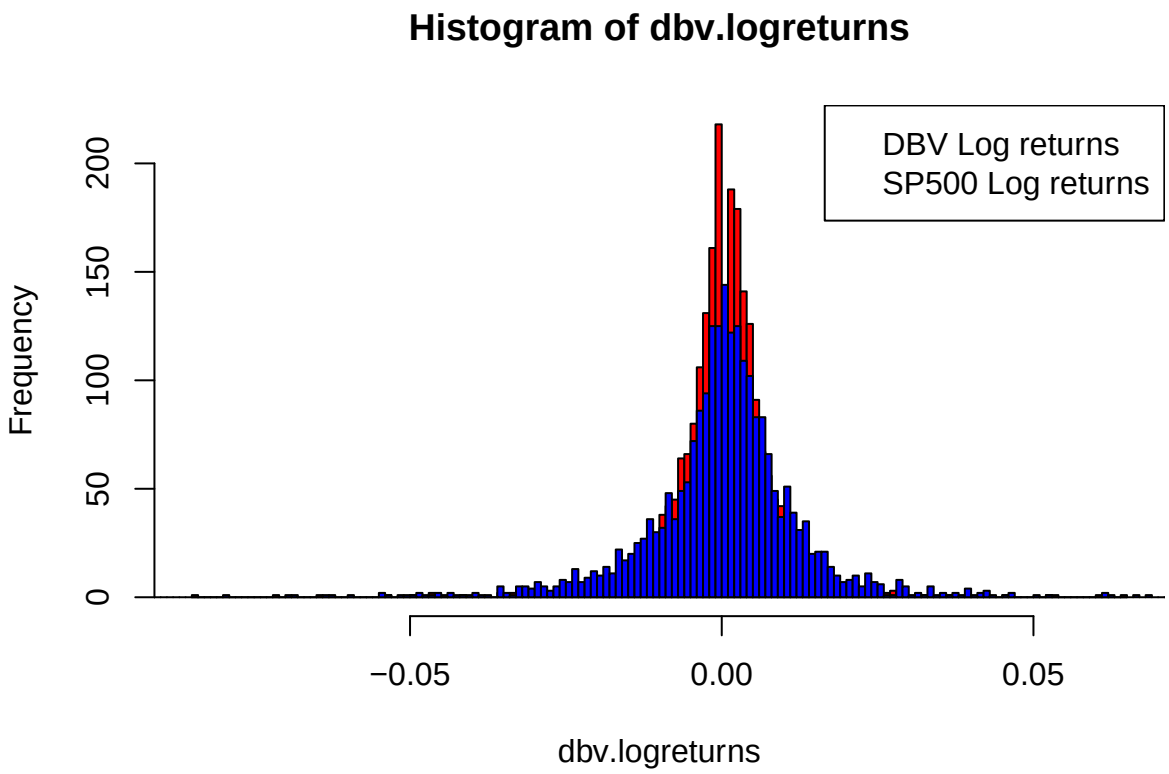



```r
plot(gspc.logreturns, main = "Times series plot of GSPC")
lines(gspc.logreturns,col="blue")
```



The histograms are stacked on one another

```
hist(dbv.logreturns,breaks = 200,col="red")
hist(gspc.logreturns, breaks = 200,col="blue",add=TRUE)
legend("topright",c("DBV Log returns","SP500 Log returns"),col=c("red","blue"))
```



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
}
```

When the sample skewness check is done for the two returns

```
skewNullCheck(dbv.logreturns)
```

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

We can see that this DBV sample skewness is significantly different from 0 with a confidence of 5%

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

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

We can see that this GSPC sample skewness is significantly different from 0 with a confidence of 5%

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
}
```

When the sample kurtosis check is done for the two returns

```
kurtosisNullCheck(dbv.logreturns)
```

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

We can see that this DBV sample excess kurtosis is significantly different from 3 with a confidence of 5%

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

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

We can see that this GSPC sample excess kurtosis is significantly different from 3 with a confidence of 5%

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
}
```

We can run the JB test for both the returns

```
jbTest(dbv.logreturns)
```

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

This confirms that the DBV returns is not normal with a confidence of 5%

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

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

This confirms that the GSPC returns is not normal with a confidence of 5%

3

```
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
```

As it can be seen from the table, GSPC has lower skewness and kurtosis as compared to DBV.

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.

Also, SP500 has 500 constituents, which allows it move towards a normal distribution as per central limit theorem, compared to the DBV currency portfolio, which only has 6 constituents. The currency portfolio also depends on international exchange and interest rates. This introduces an additional risk factor, mainly due to the presence of leverage.

5

a

Firstly, lets perform a regression where 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
```

Now, we can remove the assumptions. For heteroskedasticity, we can use the HAC value as part of the lmSumm function in Data Analytics package. This adjusts for the standard error due to heteroskedasticity.

With regards to non-normality, Asympotitic OLS will handle this case. Due to central limit theorem, we can say that OLS will be able to handle non-normal errors

```
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
```

Generally OLS with homoskedasticity underestimates the variability in error. This is the situation in this case. By removing the heteroskedasticity condition, we should expect the error in prediction to increase due to the previous underestimation.