HW 1: Returns

Sergey Pavlov, Group 141

21 september 2017

library(xts)  
library(quantmod)  
library(fBasics)  
library(car)  
library(tseries)  
library(normtest)  
library(moments)

For this assignment I’ve chosen the following datasets:

1. Microsoft inc. (MSFT) daily stock prices from Yahoo finance.
2. Electronic Arts (EA) daily stock prices from Google finance.
3. Crude Oil Prices: Brent - Europe (DCOILBRENTEU) daily price index from Federal Reserve. (PCU31141131141117) 4) S&P (^GSPC) daily index from Yahoo finance.

All of the datasets are filtered to be from 2010-01-01 to 2017-01-01.

Initially we have to download them:

#Obtain the data:  
BRENT = getSymbols('DCOILBRENTEU', src = 'FRED', auto.assign = FALSE, from = as.Date("2010-01-01"), to = as.Date("2017-01-01"))  
# Filters data on given dates, because the basic filter doesn't work for some reason.  
BRENT = BRENT[paste("2010-01-01","2017-01-01",sep="::")]   
EA = getSymbols('EA', src = 'google', auto.assign = FALSE, from = as.Date("2010-01-01"), to = as.Date("2017-01-01"))  
MSFT = getSymbols('MSFT', src = 'yahoo', auto.assign = FALSE, from = as.Date("2010-01-01"), to = as.Date("2017-01-01"))  
GSPC = getSymbols('^GSPC', src = 'yahoo', auto.assign = FALSE, from = as.Date("2010-01-01"), to = as.Date("2017-01-01"))

# a)Now lets count the NA values

#count the quantities of the NA values:  
sum(is.na(BRENT))

## [1] 61

sum(is.na(EA))

## [1] 0

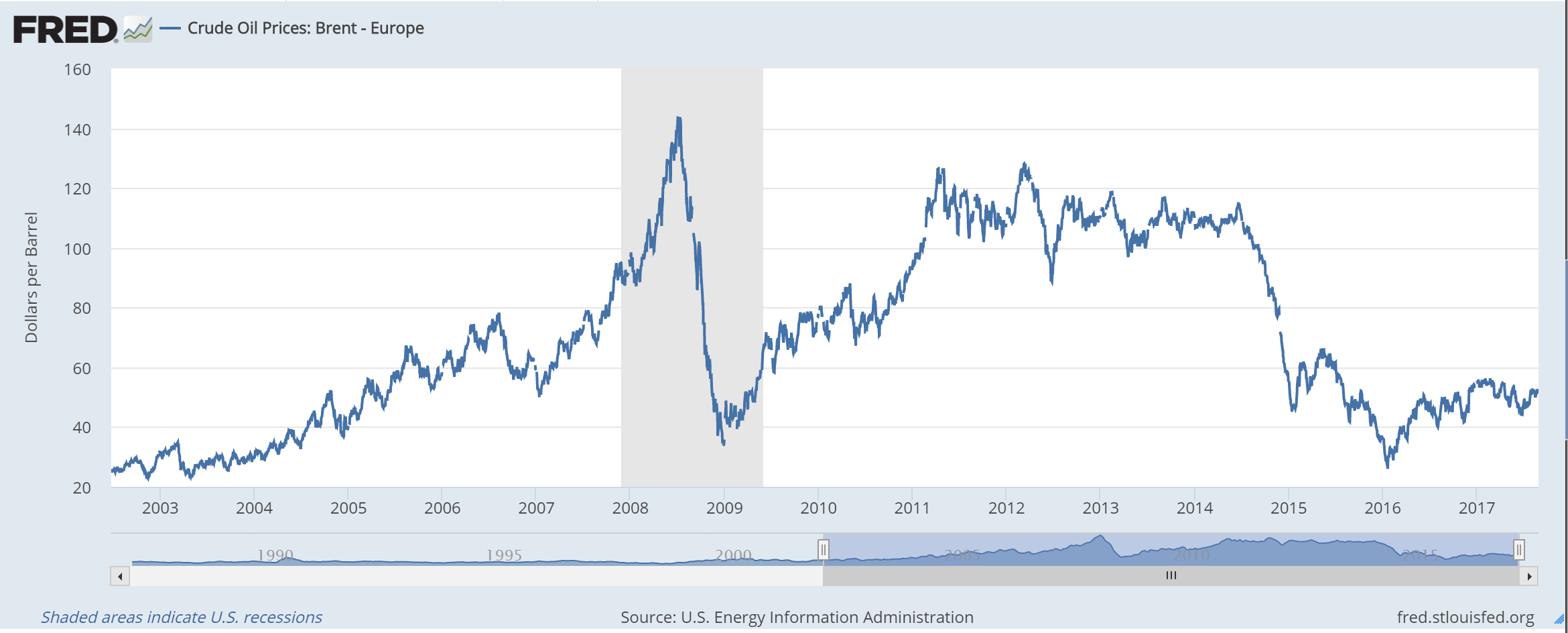
sum(is.na(MSFT))

## [1] 0

sum(is.na(GSPC))

## [1] 0

Except Brent Oil process, the chosen datasets have no “NA” values. Brent has 61 “voids” in throughout the whole period, which is seen on the following graph on the FRED website:



It is easily avoided by using the na.omit function in R:

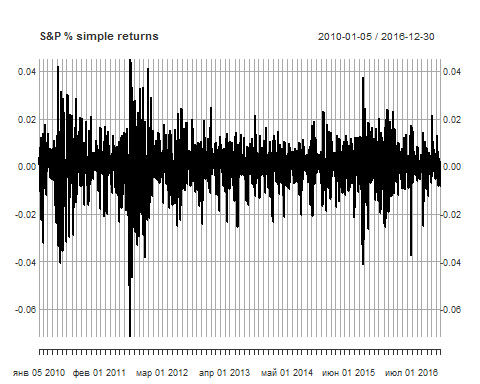
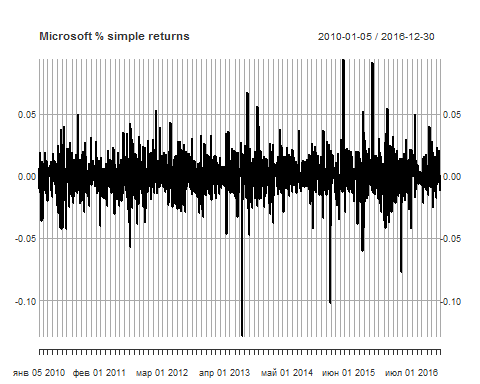
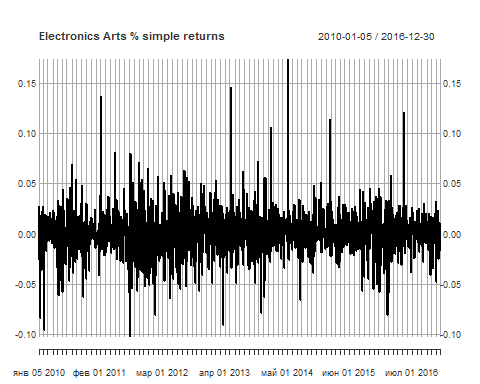
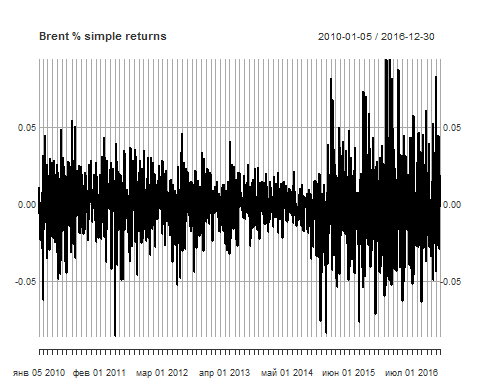
#Clear the datasets from the NA values:  
BRENT = na.omit(BRENT)  
EA = na.omit(EA)  
MSFT = na.omit(MSFT)  
GSPC = na.omit(GSPC)

# b) Daily stock returns:

Calculate returns:

BrentRet <- na.omit(diff(BRENT)/BRENT)  
EARet <- na.omit(diff(EA$EA.Close)/EA$EA.Close)  
MSFTRet <- na.omit(diff(MSFT$MSFT.Adjusted)/MSFT$MSFT.Adjusted)  
GSPCRet <- na.omit(diff(GSPC$GSPC.Adjusted)/GSPC$GSPC.Adjusted)

Now lets plot the data:



Brent: We can see the decreasing volatility trend up to the end of the year 2014 with a small-volatility cluster during the last year. After that and all the way to the end of observed period we can see a huge volatility cluster with lots of outliers. can be explained by the oil prices dip in 2014.

EA: Lots of outliers and a huge volatility during the whole period.

Microsoft: Less volatility than in the case of EA. It can be explained by the fact, that EA is a game-development company and due to that fact is less sustainable than Microsoft which products are constantly demanded.

S&P: Has some huge-volatility clusters in summer 2010, late 2011 and late 2015. Might be explained by overall economical situation.

# c) Statistics

The required statistics can be obtained by using the basicStats command from the fbasics package. The result on all of the examined stocks can be seen below:

basicStats(BrentRet)  
basicStats(EARet)  
basicStats(MSFTRet)  
basicStats(GSPCRet)

## DCOILBRENTEU EA.Close MSFT.Adjusted GSPC.Adjusted  
## nobs 1773.000000 1773.000000 1773.000000 1773.000000  
## NAs 9.000000 13.000000 12.000000 12.000000  
## Minimum -0.085947 -0.101928 -0.128663 -0.071392  
## Maximum 0.094222 0.173785 0.094631 0.045261  
## 1. Quartile -0.009885 -0.010671 -0.007347 -0.003885  
## 3. Quartile 0.009015 0.011923 0.008094 0.005210  
## Mean -0.000392 0.000594 0.000395 0.000339  
## Median 0.000000 0.000698 0.000223 0.000532  
## Sum -0.690800 1.045439 0.695095 0.596345  
## SE Mean 0.000458 0.000519 0.000348 0.000234  
## LCL Mean -0.001290 -0.000424 -0.000288 -0.000120  
## UCL Mean 0.000507 0.001612 0.001077 0.000798  
## Variance 0.000370 0.000474 0.000213 0.000096  
## Stdev 0.019247 0.021779 0.014597 0.009820  
## Skewness 0.131604 0.544791 -0.332310 -0.526705  
## Kurtosis 2.759312 6.372926 7.829041 4.407089

Brent: -Mean: small decreasing trend. Seems to be cause by the oil prices crisis. -Std deviation: respectively huge due to the crisis fluctuations. -Skewness: slight skew to the left -Excess kurtosis: a respectively small trend towards fat tails.

EA: -Mean: respectively huge increasing trend. Almost 2 times bigger than the S&P trend. -Std deviation: respectively huge due the the instability of the company. -Skewness: respectively huge skew to the left -Excess kurtosis: fat tails trend.

Microsoft: -Mean: small increasing trend. Fitts well into the S&P trend -Std deviation: respectively smal due the the stability of the company. Bigger than the S&P's though, -Skewness: medium skew to the right -Excess kurtosis: fat tails trend.

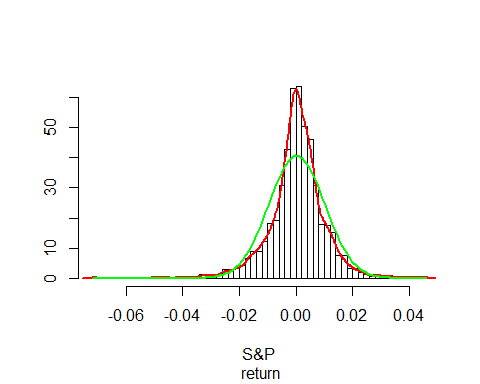
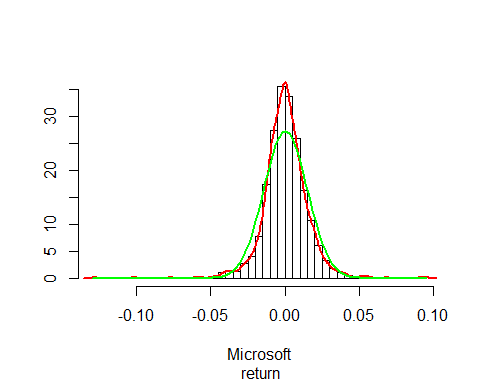
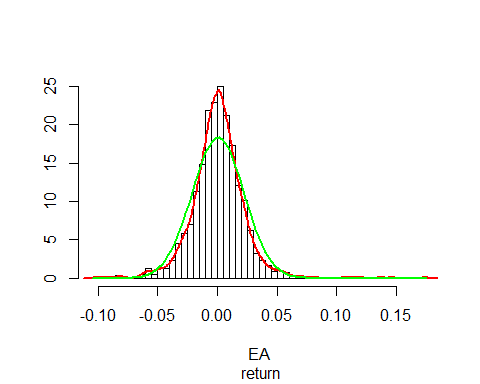
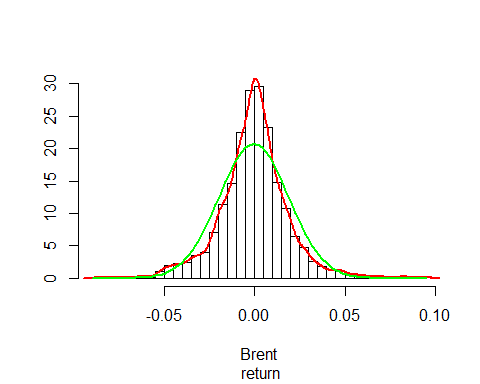
S&P: -Mean: small increasing trend -Std deviation: smaller than all of the observed stocks have, which is no surprise, because S&P index accumulates 500 companies and due to that fact is much more sustainable. -Skewness: respectively huge skew to the left. -Excess kurtosis: medium fat tails trend.

Minimum and Maximum: all of the stocks have almost the same minimum and maximum. Themselves these statistics are usually technical mistakes. However, we can see that the EA stocks have more significant minimum and maximum, which once again can be cause by the bigger volatility of its prices.

# d) Theoretical and empirical density comparison

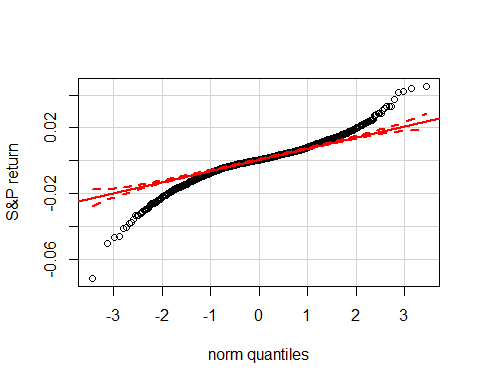
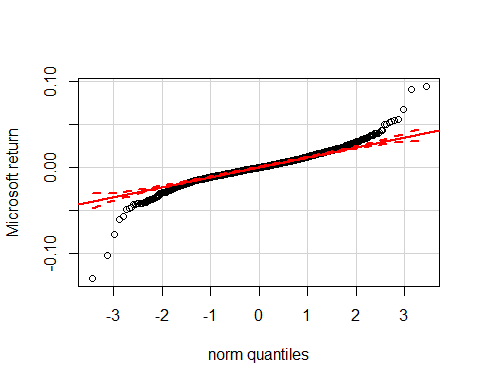
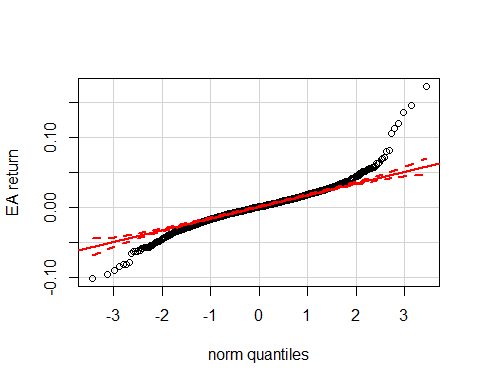
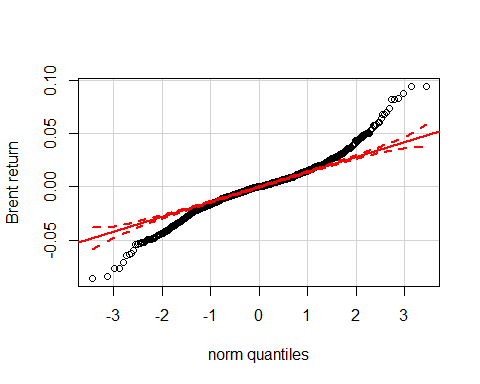
First of all lets create a function, which will simplify our further code:

MakeTheorAndEmpDensityGraph <- function(dataToPlot, stringDataName, addTheoretical)  
{  
 hist(dataToPlot, breaks=50, main="", xlab=cbind(stringDataName, " return"),  
 ylab="",prob=TRUE)  
 lines(density(dataToPlot,na.rm=TRUE),col=2,lwd=2)  
 if(addTheoretical)  
 {  
 curve(dnorm(x, mean(dataToPlot, na.rm=T), sd(dataToPlot, na.rm=T)), add=TRUE, col="green",lwd=2)  
 }  
}

Now lets build the graphs of the theoretical and empirical densities: 

On the first glance all the data seems to be normally distributed as they represent the typical form of “bell”, however the empirical and theorectical data graphs seem to differ too much. Therefore it should be assumed that the returns are not normally distributed.

# e) QQ plots for the stock returns.



In these graphs we see that the data exceeds the normal distribution area close to the edges of the graphs. Therefore all of the datasets aren’t normally distributed because they don’t fit into 95% confidence interval required for normal distribution.

# f) Jarque-Bera test

I have used the package tseries and its function jarque.bera.test(dataset) to perform the Jarque-Bera test.

jarque.bera.test(na.omit(BrentRet))

##   
## Jarque Bera Test  
##   
## data: na.omit(BrentRet)  
## X-squared = 567.37, df = 2, p-value < 2.2e-16

jarque.bera.test(na.omit(EARet))

##   
## Jarque Bera Test  
##   
## data: na.omit(EARet)  
## X-squared = 3075.6, df = 2, p-value < 2.2e-16

jarque.bera.test(na.omit(MSFTRet))

##   
## Jarque Bera Test  
##   
## data: na.omit(MSFTRet)  
## X-squared = 4544.1, df = 2, p-value < 2.2e-16

jarque.bera.test(na.omit(GSPCRet))

##   
## Jarque Bera Test  
##   
## data: na.omit(GSPCRet)  
## X-squared = 1512.1, df = 2, p-value < 2.2e-16

In all of the cases, the null hypothesis of the datasets being normally distributed is not accepted because the P-value never reached even the value of 0.05 (95% interval).

# g) Log-returns and statistics.

To compute log-returns we can use the previously computed simple returns by adding 1 to them to get them equal cur/previous, while currently they are: (cur-previous)/previous. Also just in case we have 0's we need to avoid infinity values.

# log returns   
BrentLog <- log(BrentRet+1)  
EALog <- log(EARet +1)  
MASFTLog <- log(MSFTRet +1)  
GSPCLog <- log(GSPCRet+1)  
  
#omitting na's and inf's  
BrentLog <- na.omit(BrentLog\*is.finite(BrentLog))  
EALog <- na.omit(EALog \*is.finite(EALog ))  
MASFTLog <- na.omit(MASFTLog\*is.finite(MASFTLog))  
GSPCLog <- na.omit(GSPCLog\*is.finite(GSPCLog))

now we can get the basic statistics:

basicStats(BrentLog)  
basicStats(EALog)  
basicStats(MASFTLog)  
basicStats(GSPCLog)

## DCOILBRENTEU EA.Close MSFT.Adjusted GSPC.Adjusted  
## nobs 1773.000000 1773.000000 1773.000000 1773.000000  
## NAs 9.000000 13.000000 12.000000 12.000000  
## Minimum -0.089866 -0.107505 -0.137726 -0.074068  
## Maximum 0.090044 0.160234 0.090418 0.044267  
## 1. Quartile -0.009934 -0.010728 -0.007374 -0.003892  
## 3. Quartile 0.008975 0.011852 0.008062 0.005196  
## Mean -0.000577 0.000358 0.000288 0.000290  
## Median 0.000000 0.000697 0.000223 0.000531  
## Sum -1.017527 0.630860 0.506765 0.511126  
## SE Mean 0.000458 0.000517 0.000349 0.000235  
## LCL Mean -0.001476 -0.000655 -0.000397 -0.000170  
## UCL Mean 0.000322 0.001372 0.000972 0.000750  
## Variance 0.000370 0.000470 0.000214 0.000097  
## Stdev 0.019248 0.021678 0.014643 0.009845  
## Skewness -0.003945 0.296960 -0.555502 -0.618631  
## Kurtosis 2.665725 5.423650 8.832975 4.679631

-Means saved their trends, while getting closer to the zero value. However, absolute brent mean value now is much bigger than the others.  
-Standard deviations have saved their values.  
-Skewness of the Brent stock has almost disappeared. EA skewnes shows a better value. However Microsoft and S&P skewness values became slighly worse: they skew more to the right after the lof-transformation.  
-Excess kurtosis of EA and EA became slightly better: they show a less fat-tailed trend, although that is not enough. Microsoft, S&P show a more fat-tailed trend.  
-Minimum and maximum: almost all of the data min and max values became a small, unsignificant fraction closer to zero.

# h) Mean is zero test (t.test)

To test the mean value on being a zero we can use t.test method from the stats standard library.

# log test mean == 0  
t.test(BrentLog, mu = 0, conf.level = 0.95)

##   
## One Sample t-test  
##   
## data: BrentLog  
## t = -1.2587, df = 1763, p-value = 0.2083  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.0014756551 0.0003219967  
## sample estimates:  
## mean of x   
## -0.0005768292

t.test(EALog , mu = 0, conf.level = 0.95)

##   
## One Sample t-test  
##   
## data: EALog  
## t = 0.69369, df = 1759, p-value = 0.488  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.0006550048 0.0013718911  
## sample estimates:  
## mean of x   
## 0.0003584432

t.test(MASFTLog, mu = 0, conf.level = 0.95)

##   
## One Sample t-test  
##   
## data: MASFTLog  
## t = 0.82473, df = 1760, p-value = 0.4096  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.0003965867 0.0009721288  
## sample estimates:  
## mean of x   
## 0.000287771

t.test(GSPCLog, mu = 0, conf.level = 0.95)

##   
## One Sample t-test  
##   
## data: GSPCLog  
## t = 1.2372, df = 1760, p-value = 0.2162  
## alternative hypothesis: true mean is not equal to 0  
## 95 percent confidence interval:  
## -0.0001698721 0.0007503674  
## sample estimates:  
## mean of x   
## 0.0002902477

In all of the cases we can see that the p-value is more than 0.05 (95% interval). Therefore we have no reason to decline null hypothesis of true mean being equal to 0.

# i) Jarque bera test

Once again we can use the Jaque-bera test:

# Jarque-Bera test  
jarque.bera.test(na.omit(BrentLog))

##   
## Jarque Bera Test  
##   
## data: na.omit(BrentLog)  
## X-squared = 524.82, df = 2, p-value < 2.2e-16

jarque.bera.test(na.omit(EALog))

##   
## Jarque Bera Test  
##   
## data: na.omit(EALog)  
## X-squared = 2190.7, df = 2, p-value < 2.2e-16

jarque.bera.test(na.omit(MASFTLog))

##   
## Jarque Bera Test  
##   
## data: na.omit(MASFTLog)  
## X-squared = 5833, df = 2, p-value < 2.2e-16

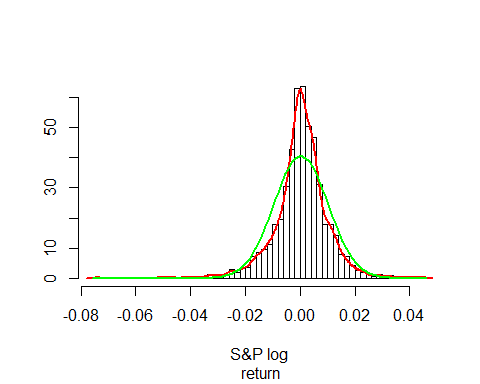
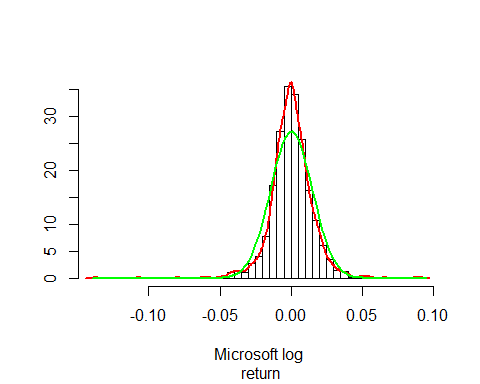
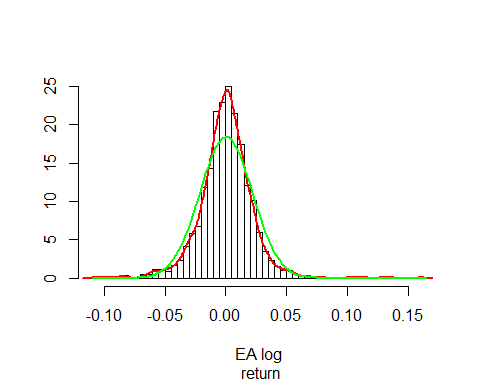
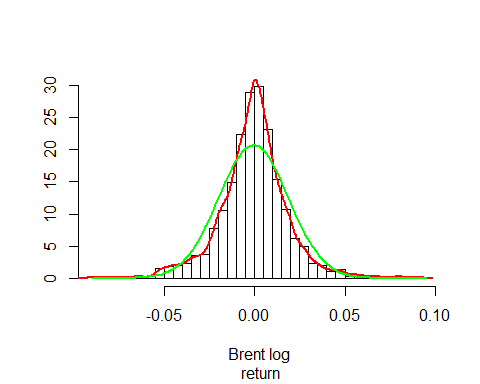
jarque.bera.test(na.omit(GSPCLog))

##   
## Jarque Bera Test  
##   
## data: na.omit(GSPCLog)  
## X-squared = 1725.4, df = 2, p-value < 2.2e-16

As we can see that although we've applied the log-transformation to the data, none of the datasets became normally distributed, because p-values are close to 0's (far away from 0.05 value).

# j) empirical and theoretical density plots

In order to get the density plots we can use the function, which we've declared earlier:



By comparing empirical and theoretical graphs we can see that the situation has remained almost the same and the graphs still aren't normally distributed.

# k) Correlations and scatter plots

Now lets calculate the three different correlations of the 3 stocks with the S&P index.

Pearson:

cor(GSPCLog[index(BrentLog)],BrentLog[index(GSPCLog)], method = "pearson")

## DCOILBRENTEU  
## GSPC.Adjusted 0.3247206

cor(GSPCLog[index(EALog)],EALog[index(GSPCLog)],method="pearson")

## EA.Close  
## GSPC.Adjusted 0.5268822

cor(GSPCLog[index(MASFTLog)],MASFTLog[index(GSPCLog)],method="pearson")

## MSFT.Adjusted  
## GSPC.Adjusted 0.6546166

Spearman:

cor(GSPCLog[index(BrentLog)],BrentLog[index(GSPCLog)], method = "spearman")

## DCOILBRENTEU  
## GSPC.Adjusted 0.2815792

cor(GSPCLog[index(EALog)],EALog[index(GSPCLog)],method="spearman")

## EA.Close  
## GSPC.Adjusted 0.5518318

cor(GSPCLog[index(MASFTLog)],MASFTLog[index(GSPCLog)],method="spearman")

## MSFT.Adjusted  
## GSPC.Adjusted 0.6620869

Kendall:

cor(GSPCLog[index(BrentLog)],BrentLog[index(GSPCLog)], method = "kendall")

## DCOILBRENTEU  
## GSPC.Adjusted 0.1921994

cor(GSPCLog[index(EALog)],EALog[index(GSPCLog)],method="kendall")

## EA.Close  
## GSPC.Adjusted 0.3962436

cor(GSPCLog[index(MASFTLog)],MASFTLog[index(GSPCLog)],method="kendall")

## MSFT.Adjusted  
## GSPC.Adjusted 0.4855205

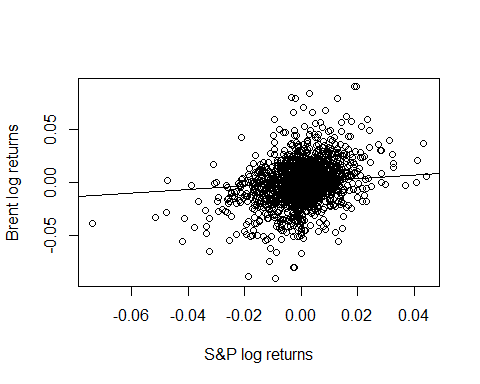
Analyzing the values achieved by the each test we can see that all of the methods show different, but somewhat close results. Being a commodity Brent stock prices aren't as stable as the summarized index of S&P 500 companies. Therefore the dependency between Brent and S&P index almost absents. On the other hand the other two companies have a normal correlation with that index being a part of it.

Now lets make a function which will make scatter plots for us:

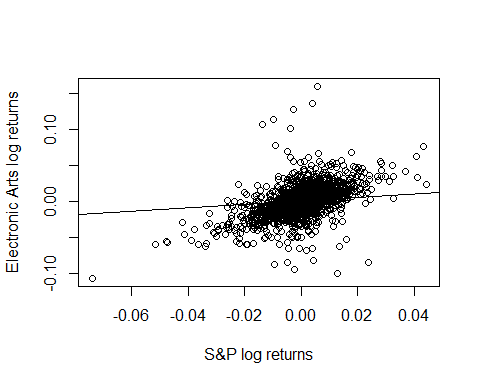
MakeScatterPlot <- function(FirstData, SecondData, nameFirstData, nameSecondData)  
{  
 plot(coredata(FirstData),coredata(SecondData), xlab = nameFirstData, ylab = nameSecondData)   
 LogLM=lm(coredata(FirstData[index(SecondData)])~coredata(SecondData[index(FirstData)]))   
 abline(a =summary(LogLM)$coefficients[1,1],b= summary(LogLM)$coefficients[2,1])  
}

Now we can do the plots themselves:

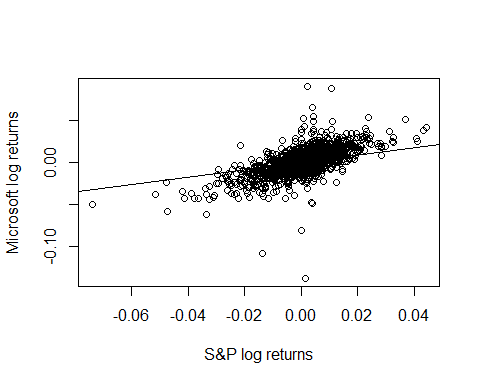
MakeScatterPlot(GSPCLog[index(BrentLog)], BrentLog[index(GSPCLog)], "S&P log returns", "Brent log returns")



MakeScatterPlot(GSPCLog[index(EALog)], EALog[index(GSPCLog)], "S&P log returns", "Electronic Arts log returns")



MakeScatterPlot(GSPCLog[index(MASFTLog)], MASFTLog[index(GSPCLog)], "S&P log returns", "Microsoft log returns")



By analyzing the plots we can assume, that the less correlated and sustainable market stock is the more graph looks like a round cloud. It can be easily seen on the graph of Brent. No particular trend can bee seen on that graph.  
A better situation can be seen on the EA graph. It truly correlates with S&P (probably because it's a part of S&P) in a positive way, although the prices are really volatile, so the graph still looks a bit cluttered.  
The best of all 3 situations is Mirosoft. Being correlated with S&P and having a relatively volatile stock returns it shows the positive correlation trend with S&P, which can be easily recognized on the plot.