STA457#3

jonathanwai

28/11/2019

install.packages("readxl")

```
library("readxl")
IBM = read_excel("C:\\Users\\Jonathan\\Desktop\\IBM457.xlsx")
```

```
IBM
```

```
## # A tibble: 1,006 x 1
##    Adj_Close
##        <dbl>
##  1      110.
##  2      109.
##  3      110.
##  4      108.
##  5      104.
##  6      107.
##  7      109.
##  8      112.
##  9      113.
## 10      112.
## # ... with 996 more rows
```

```
mean(IBM$Adj_Close)
```

```
## [1] 134.768
```

Question 1

```
length(IBM$Adj_Close)
```

```
## [1] 1006
```

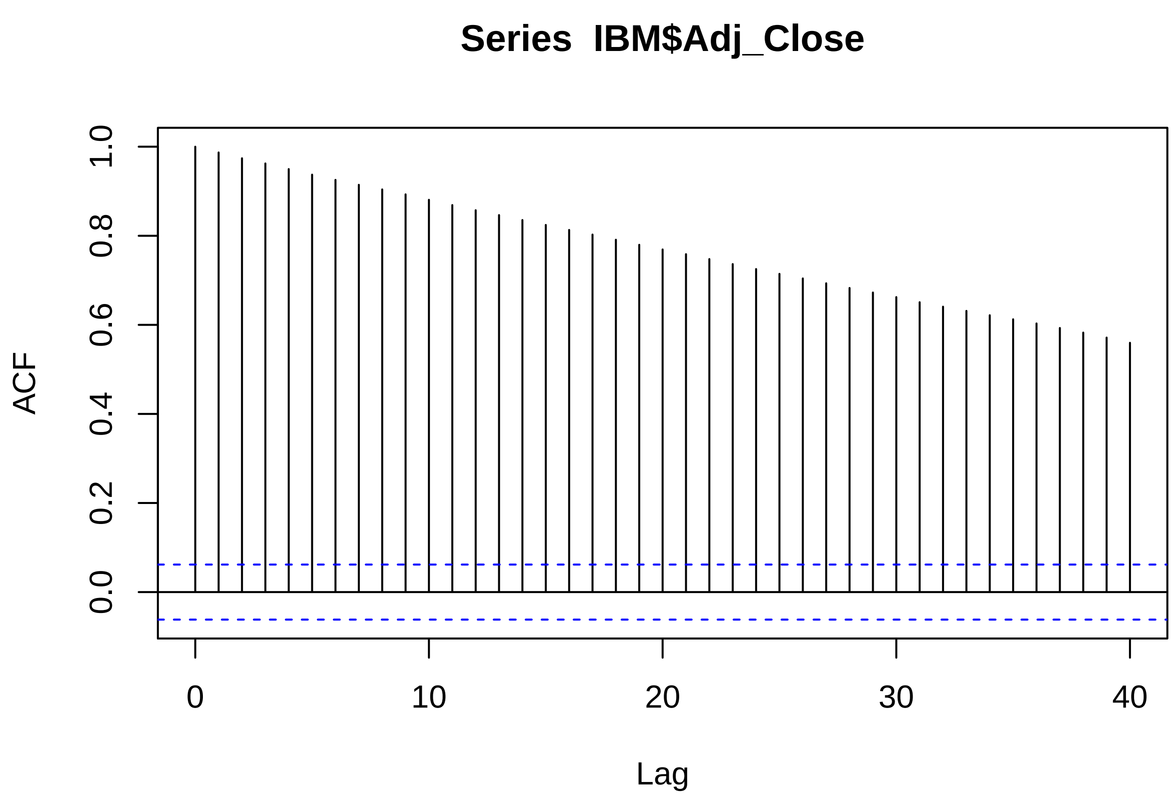
```
head(IBM,30)
```

```
## # A tibble: 30 x 1
##    Adj_Close
##        <dbl>
##  1      110.
##  2      109.
##  3      110.
##  4      108.
##  5      104.
```

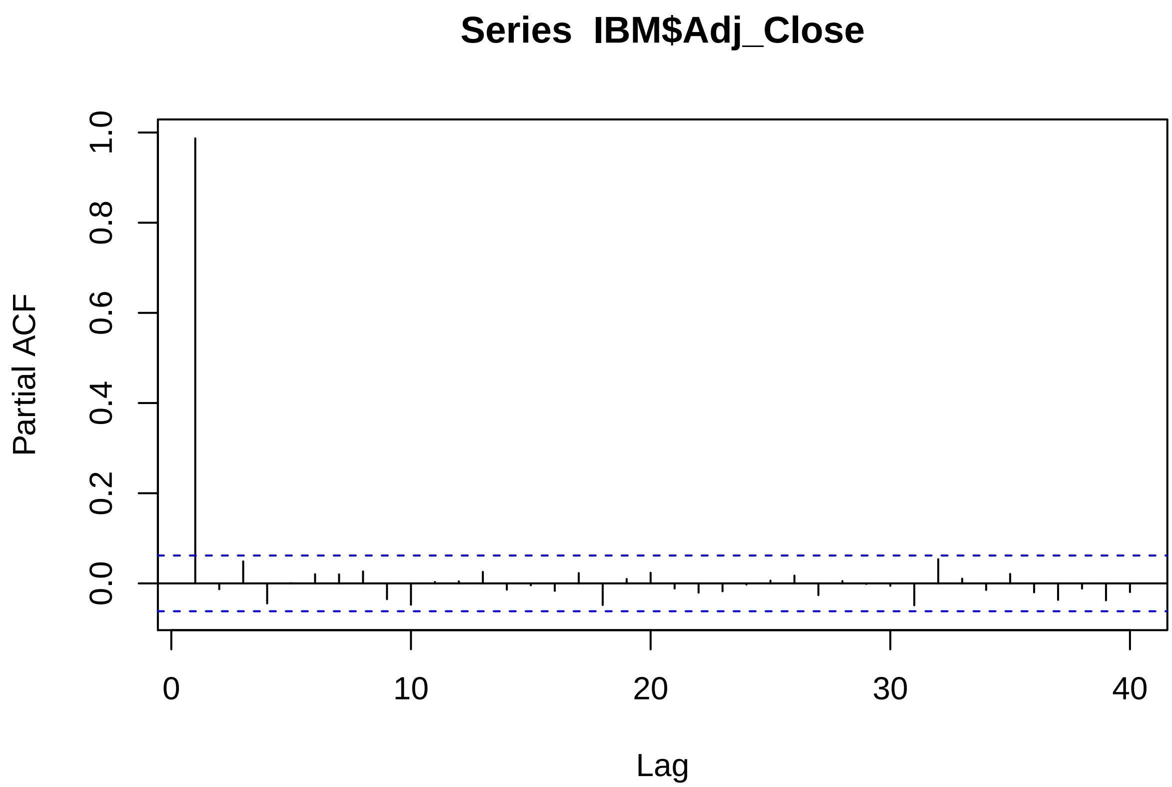
```
##   6       107.
##   7       109.
##   8       112.
##   9       113.
## 10       112.
## # ... with 20 more rows
```

Question 2

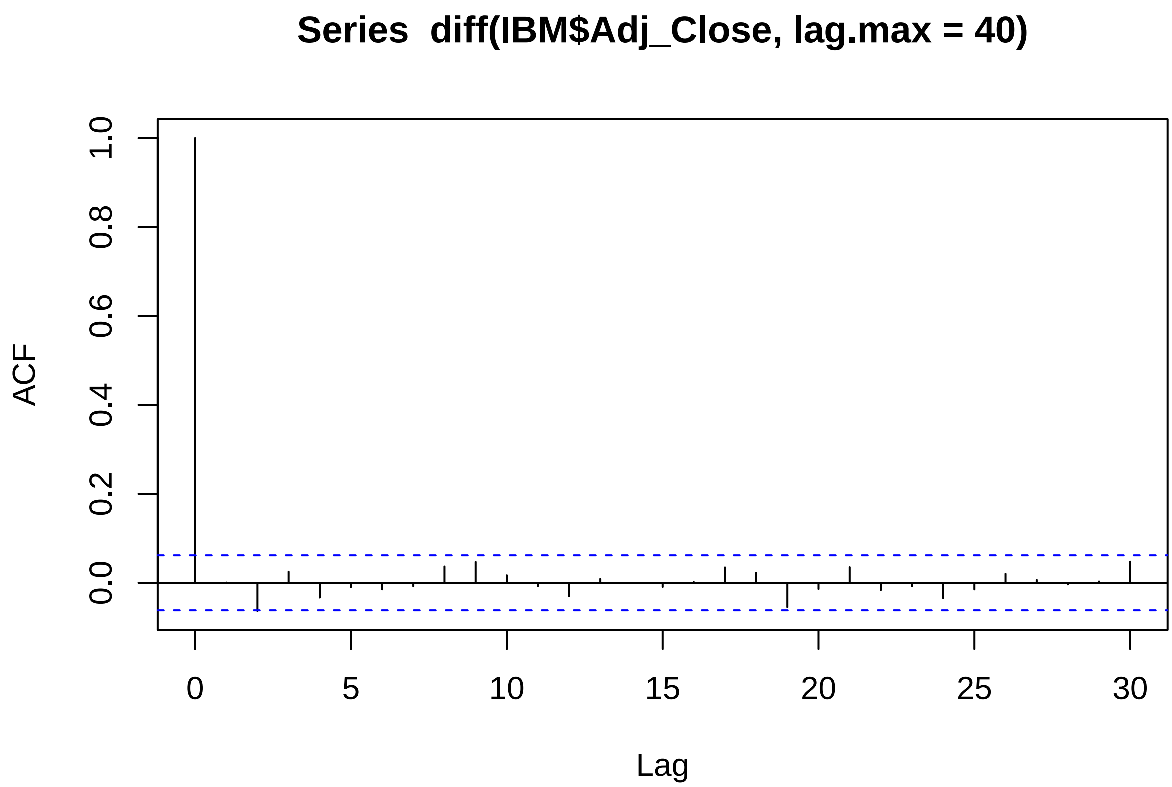
```r
acf(IBM$Adj_Close, lag.max =40)
```



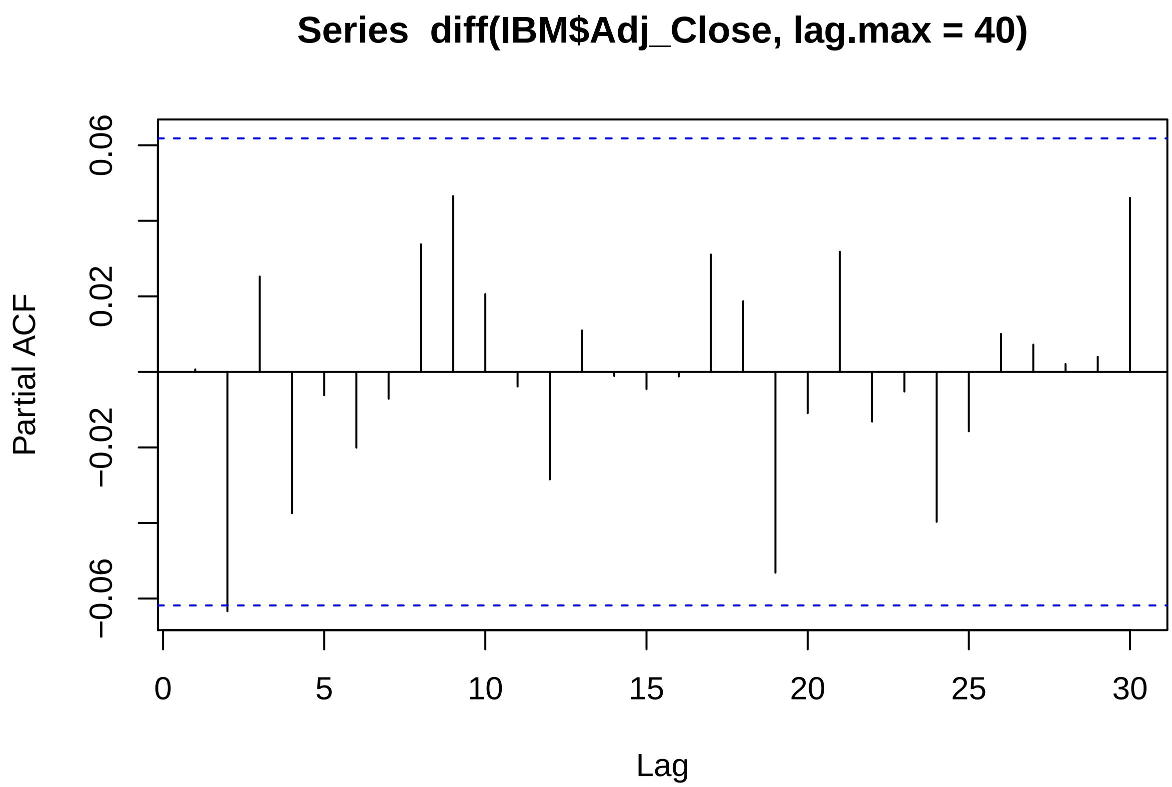

```r
pacf(IBM$Adj_Close, lag.max =40)
```




```
acf(diff(IBM$Adj_Close, lag.max =40))
```




```r
pacf(diff(IBM$Adj_Close, lag.max =40))
```



ACF decreases slowly. pacf cuts off at lag 1. we would fit an ar1 model.

Question 3

```r
Box.test(IBM$Adj_Close, lag=5, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  IBM$Adj_Close
## X-squared = 4680.7, df = 5, p-value < 2.2e-16
```

P value is less than 0.05, we reject the null hypothesis.

Null hypothesis: p(1) =. . . = p(5) = 0 for some K ={5}, i.e white noise hypothesis

Alt hypothesis: one or more of p(1), . . . ,p(5) is nonzero

Question 4

```
ar_model=arima(IBM$Adj_Close, order=c(1,0,0))
Box.test(ar_model$resid, lag=5, type="Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  ar_model$resid
## X-squared = 5.5574, df = 5, p-value = 0.3517
```

```
ar_model
```

```
##
## Call:
## arima(x = IBM$Adj_Close, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.9905   132.6845
## s.e.  0.0043     5.1762
##
## sigma^2 estimated as 2.915:  log likelihood = -1967.57,  aic = 3941.14
```

Null hypothesis: p(1) =… = p(5) = 0 for some K ={5}, i.e white noise hypothesis

Alt hypothesis:one or more of p(1), … ,p(5) is nonzero

P is greater than 0.05, therefore we fail to reject null hypothesis

Find no evidence that any autocorrelation are 0

AR(1) model with the esitmated parameters

132.6845_0.9905yt-1 + error t

Question 5

install.packages("tseries")

```
library("tseries")
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
pp.test(IBM$Adj_Close)
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  IBM$Adj_Close
## Dickey-Fuller Z(alpha) = -13.637, Truncation lag parameter = 7,
## p-value = 0.349
## alternative hypothesis: stationary
```

Is similar to ADF test

```
adf.test(IBM$Adj_Close)
```

```
##
##  Augmented Dickey-Fuller Test
```

```
##
## data:  IBM$Adj_Close
## Dickey-Fuller = -2.9457, Lag order = 10, p-value = 0.178
## alternative hypothesis: stationary
```

There is a unit root

since the p value is greater than 0.05, fail to reject null hypothesis

the series might contain a unit root

```r
kpss.test(IBM$Adj_Close)
```

```
## Warning in kpss.test(IBM$Adj_Close): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  IBM$Adj_Close
## KPSS Level = 1.9266, Truncation lag parameter = 7, p-value = 0.01
```

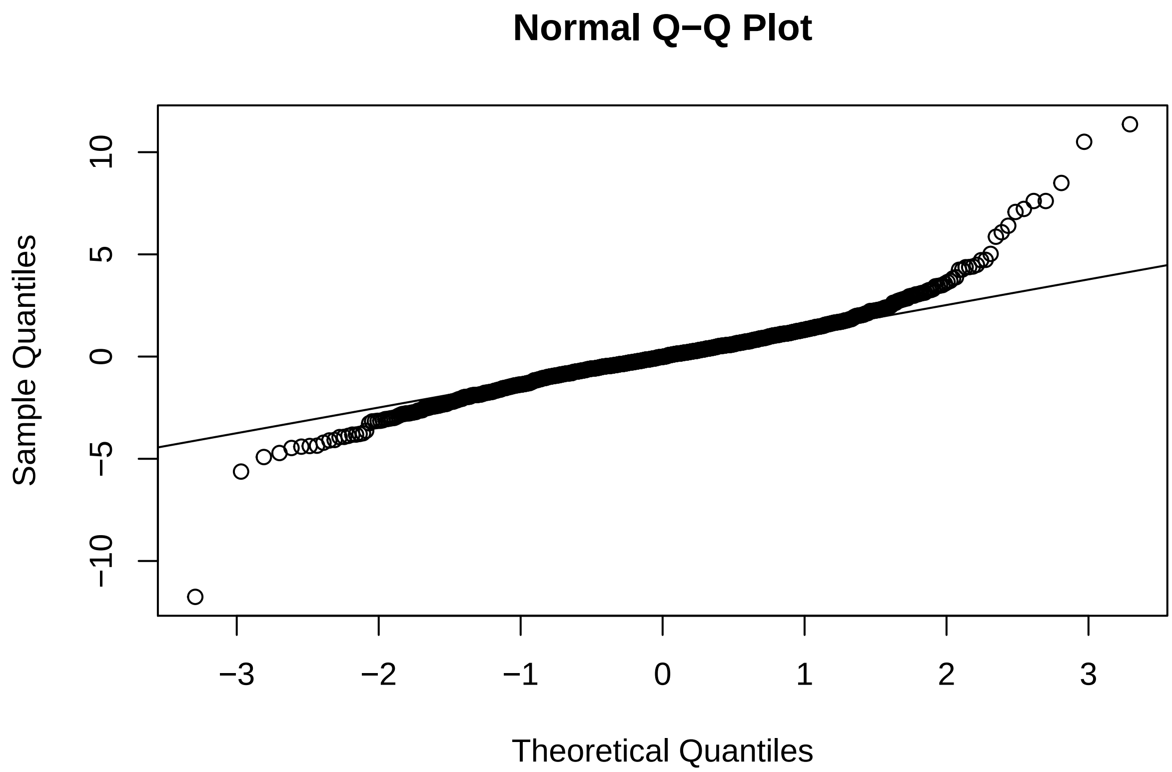
Null hypothesis is that it is stationary

Alternative it is not stationary

p value smaller than 0.05, reject null

Question 6

```r
noise=resid(ar_model)
qqnorm(noise)
qqline(noise)
```




```
shapiro.test(noise)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  noise
## W = 0.92283, p-value < 2.2e-16
```

QQ plot shows. Not normally distributed, skewed.

Null hypothesis is that the data is normally distributed

Alternative is that it is not normally distributed

our p value is less than 0.05, we reject the null, therefore data is not normally distributed. Confirmed by qq plot

Question 7

install.packages("forecast")

```r
library(forecast)
```

```
## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

```r
arima_model=auto.arima(IBM$Adj_Close, ic=c("aic"))
arima_noise=resid(arima_model)
```

```r
arima_model
```

```
## Series: IBM$Adj_Close
## ARIMA(0,1,0)
##
## sigma^2 estimated as 2.928:  log likelihood=-1965.86
## AIC=3933.72   AICc=3933.73   BIC=3938.63
```

AR order 0, MA order 0. D is 1

Yt = Yt-1 + et

Question 8

install.packages("MASS")

```r
library(MASS)
fit <- fitdistr(noise, densfun="t")
```

```
## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced

## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced

## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced

## Warning in dt((x - m)/s, df, log = TRUE): NaNs produced
```

```r
fit
```

```
##         m               s               df
##   -0.007168365    1.136027403    3.471574750
##  ( 0.043065636) ( 0.045999682) ( 0.410663801)
```

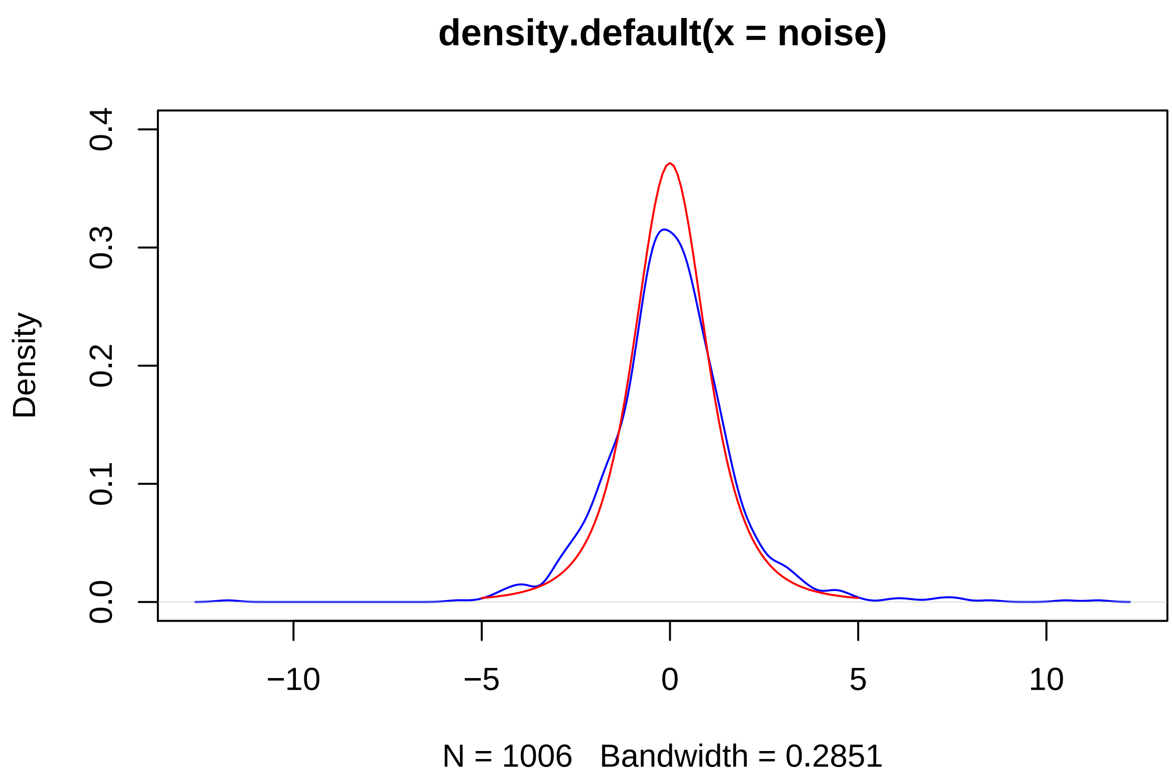
Question 8

```r
fit
```

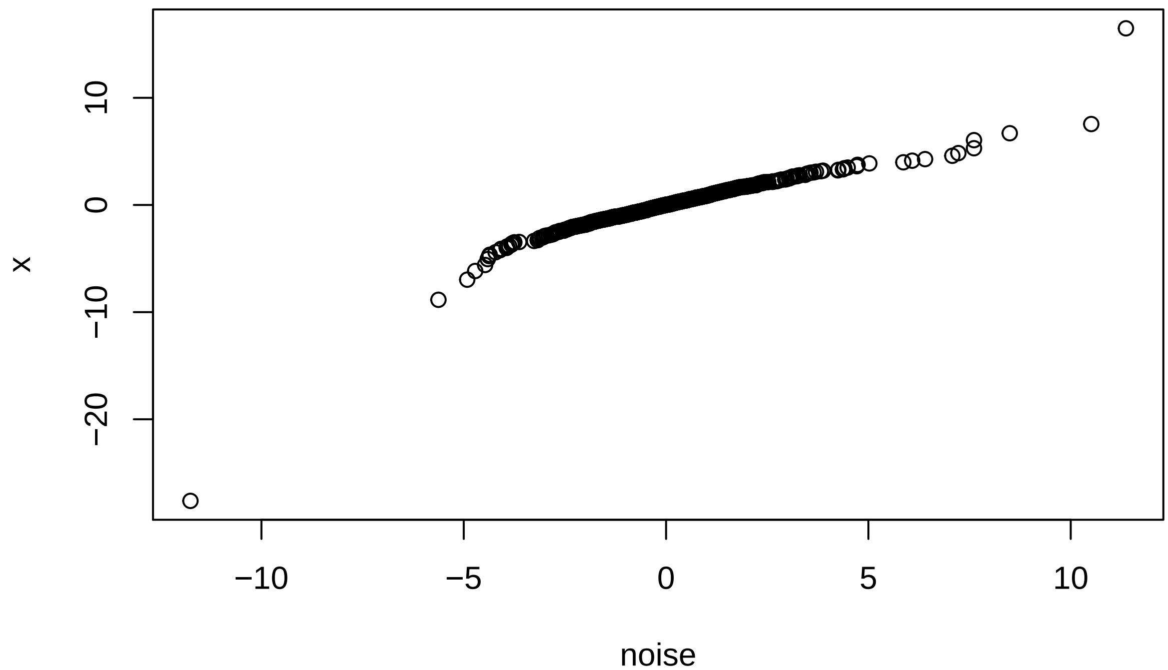
```
##          m               s               df     
##   -0.007168365    1.136027403    3.471574750 
##  ( 0.043065636) ( 0.045999682) ( 0.410663801)
```

Question 9

```r
plot(density(noise), ylim = c(0, 0.4),col="blue")
curve(dt(x,fit$estimate[3]),from = -5, to = 5, add=TRUE, col="red")
```




```r
x=rt(10000,fit$estimate[3])
qqplot(noise,x)
```

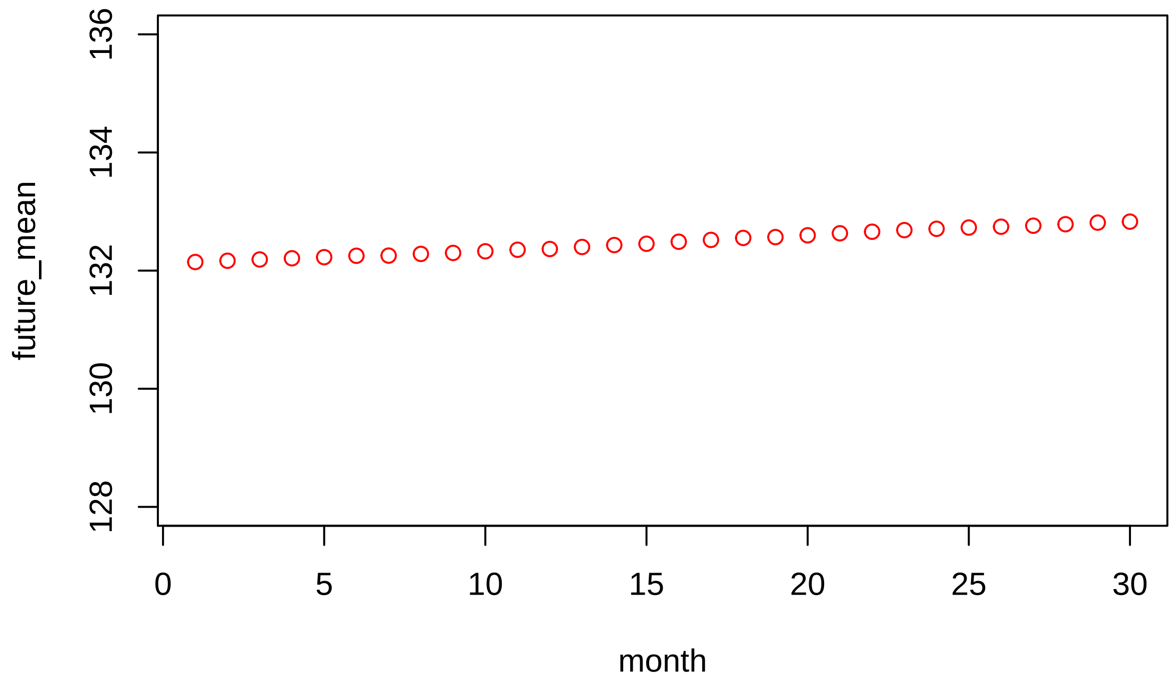


Looking at density plot, it shows a reasonable fit

Question 10

```
n=length(IBM$Adj_Close)
niter=50000
n.ahead=30
#Creating an empty matrix for future observations
futureobs=matrix(0,nrow=niter, ncol=n.ahead)
for (i in 1:niter) {
  #Generating errors from ARIMA's residuals
  errors=sample(arima_noise,n.ahead,replace=TRUE)
  for (j in 1:n.ahead) {
    #Assigning values to future observations based on the ARIMA model
    if (j==1){futureobs[i,j]=IBM$Adj_Close[n]+errors[j]}
     else {futureobs[i,j]=futureobs[i,j-1]+errors[j]}
  }
}
#Calculating mean values for each day of the month
future_mean=apply(futureobs,2,mean)
#Calculating the confidence bands
u1=0*(1:n.ahead)
```

```
l1=u1
for (k in 1:n.ahead){
  u1[k]=quantile(futureobs[,k],0.975)
  l1[k]=quantile(futureobs[,k],0.025)
}
month=seq(1:30)
plot(month,future_mean,col="red", ylim=c(128,136))
```

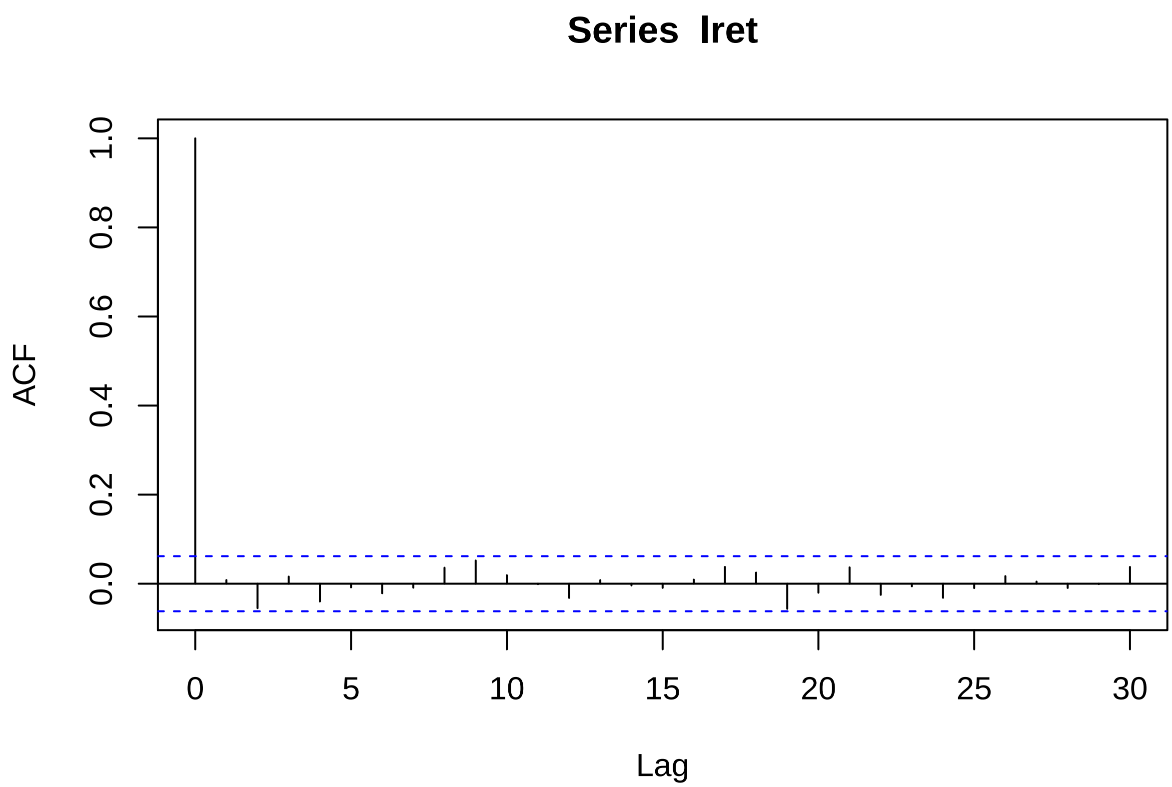


Question 11

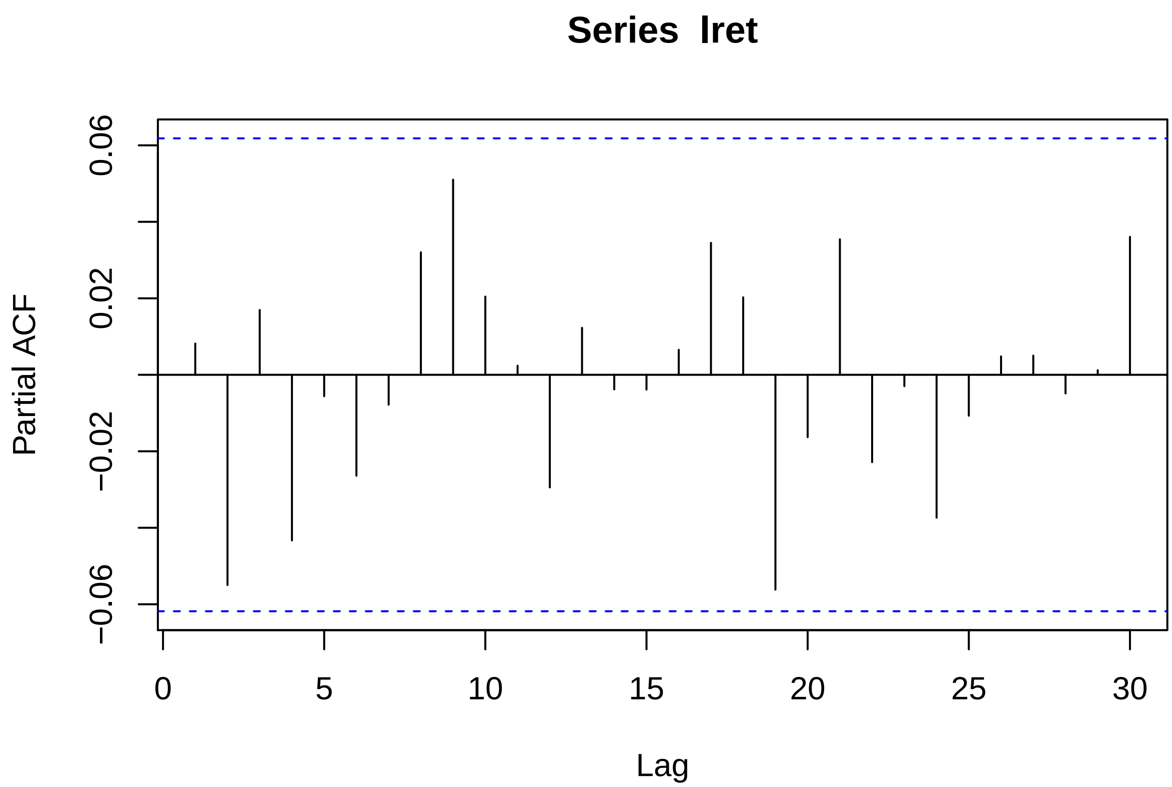
```
n=length(IBM$Adj_Close)
n
```

```
## [1] 1006
```

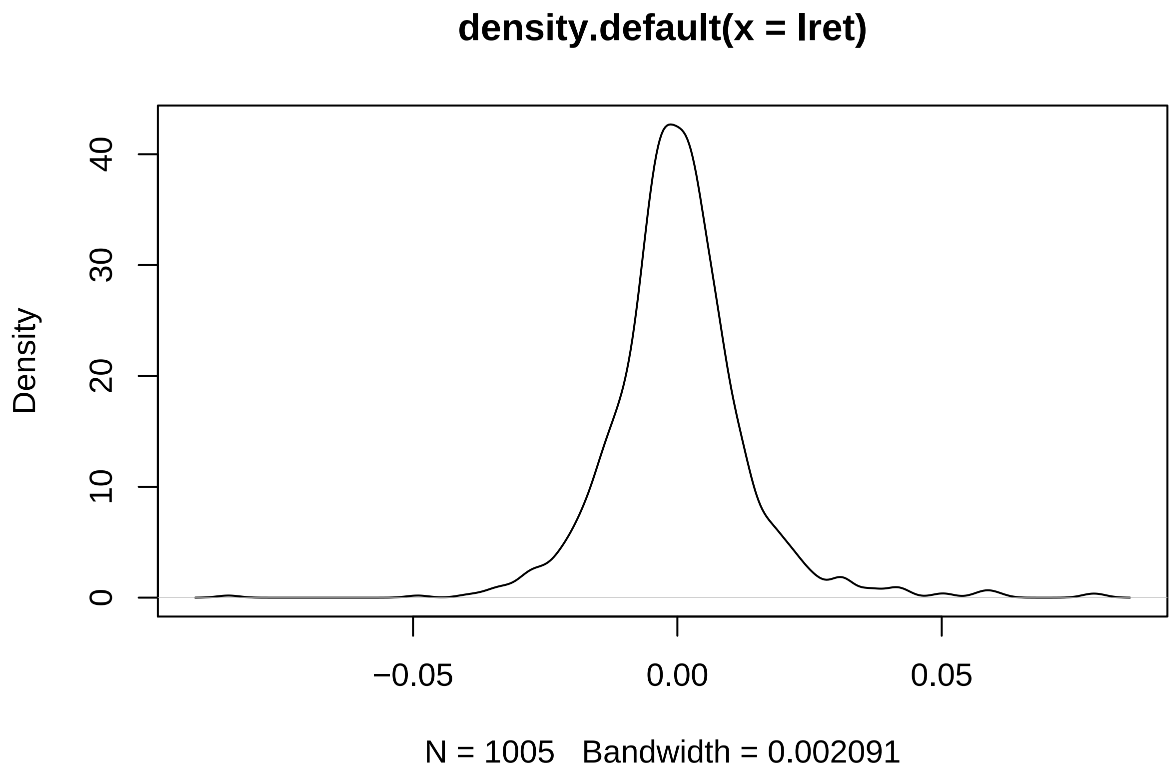
```
lret <- log(IBM$Adj_Close[-1]/IBM$Adj_Close[-n])
acf(lret)
```




```
pacf(lret)
```




```r
plot(density(lret))
```



Log returns show more normality