STA457 Assignment 3

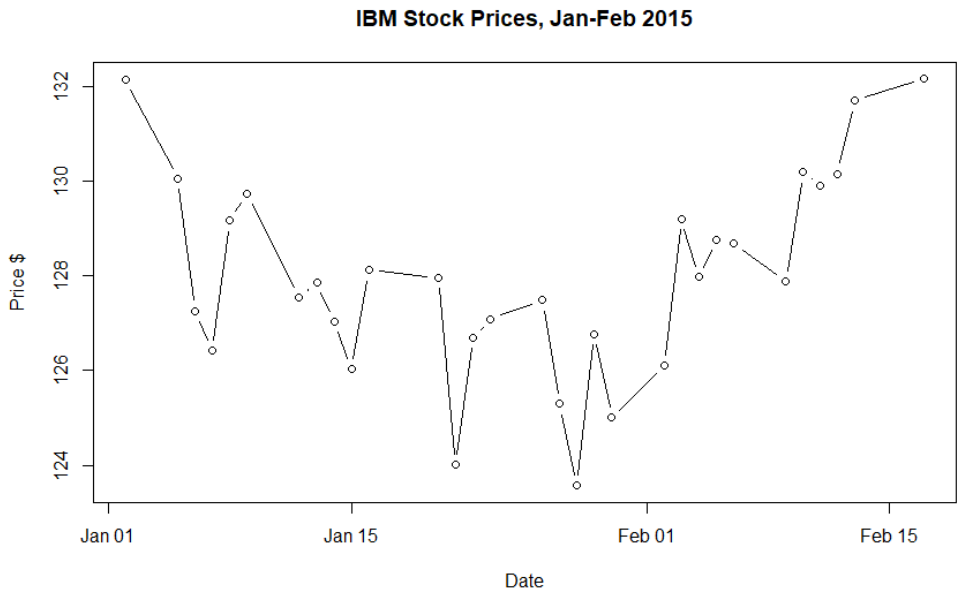
Brian Lee 1002750855

1

Read in the data set for the specified time period. Identify the length of your data and display the first month of the time series data (roughly 30 data points don't go wild)
Answer:

```
fileRead <- read.csv("IBM.csv")
IBMdata <- fileRead[c(203:1208),]
prices <- IBMdata$Adj.Close
dates <- as.Date(IBMdata$Date, format='%Y-%m-%d')

length <- length(prices[]) #length of dataset

#Displaying first 30 data points, roughly 1 1/2 months
firstMonth <- IBMdata[c(976:1006),]
fmDates <- as.Date(firstMonth$Date, format='%Y-%m-%d')
fmPrices <- as.numeric(firstMonth$Adj)
plot(fmDates, fmPrices, type="b", main="IBM Stock Prices,
Jan-Feb 2015", xlab="Date", ylab="Price $")
```

From this code, we see that the length of our data is 1006 data points from 2015-01-02 to 2018-12-31. Furthermore, the time series plot of the first 30 data points, which is approximately the first 1 1/2 months is given below:

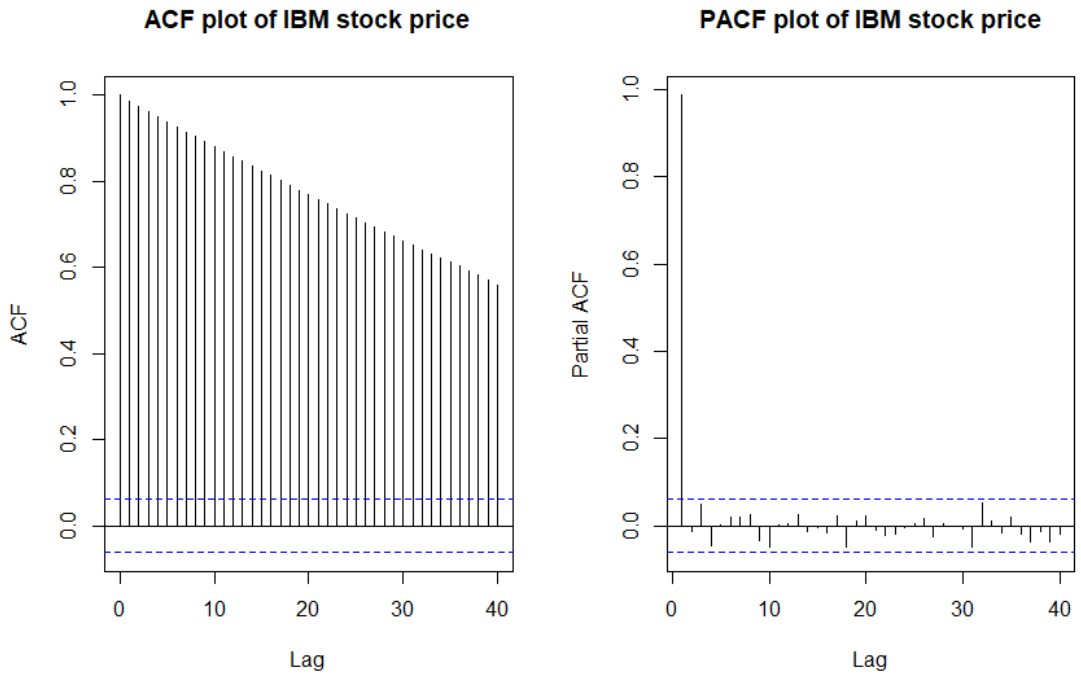


2

Plot ACF and PACF of IBM stock price and change in IBM stock price up to lag 40. Based on auto correlations and the partial auto correlations, if you want to fit an AR(p) model, what order p would you pick?

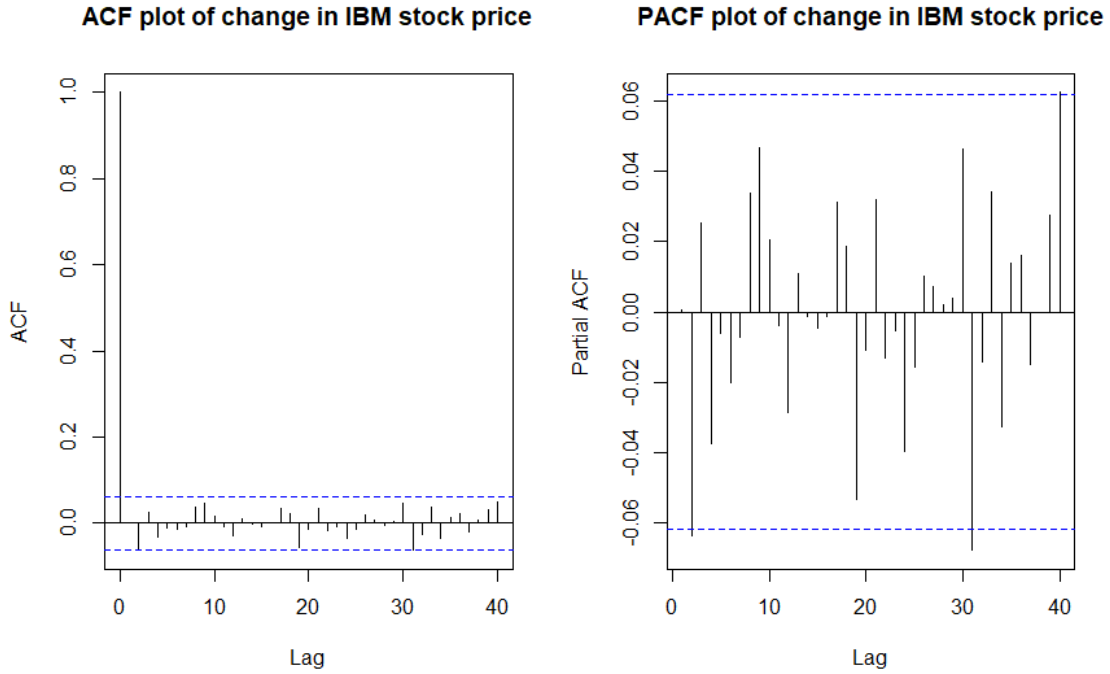
Answer:

```
priceDelta <- -diff(prices)
#ACF and PACF plots of stock price
acf(prices, 40, main="ACF plot of IBM stock price")
pacf(prices, 40, main="PACF plot of IBM stock price")

#ACF and PACF plots of change in stock price
acf(priceDelta, 40)
pacf(priceDelta, 40)
```



The ACF plot of the stock price decays slowly to zero, which potentially indicates nonstationarity of stationarity with long memory dependence. Looking at the PACF plot, we see that there is a large spike in the first lag, followed by non-significant values for all subsequent lags. This pattern indicates that there is an autoregressive term of order 1, thus an AR(1) model would be a good choice to fit the stock price to.

On the other hand, looking at the change in stock price, we see that the ACF plot only has a significant spike at lag 0, and non-significant autocovariance at all positive lags, which is an indicator of stationarity. Further supporting this is the PACF plot, where we see that almost all of the autocorrelations lie within the confidence band, and the ones that lie outside are not significantly outside of the confidence interval range. Due to this, it seems that an AR(p) model is not an appropriate model to fit to the change in stock price.



3

Perform a simultaneous test for $\rho(1) = \rho(2) = \ldots = \rho(K)$ where $K = 5$ for the IBM stock price. Comment on the results.

Answer:

```
#Ljung-Box test
Box.test(prices, lag=5, type='Ljung-Box')

Output:
Box-Ljung test

data:  prices
X-squared = 4680.7, df = 5, p-value < 2.2e-16
```

The very small p-value indicates that we can reject the null hypothesis, which is that $\rho(1) = \rho(2) = \ldots = \rho(K) = 0$. The result is strong evidence against the data being white noise, as we're seeing significant serial correlation at positive lags. However, result is not enough to conclude stationarity or non-stationarity of the data.

4

Fit an AR(1) model. Perform diagnostic test on this model (doest this model fit well?) and comment. Write down the AR(1) model with the estimated parameters.

Answer:

```
#Fitting an AR(1) model to stock price data
fitAR1 = arima(prices, order=c(1,0,0))
print(fitAR1)


Output:
```

```
Call:
arima(x = prices, 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
```

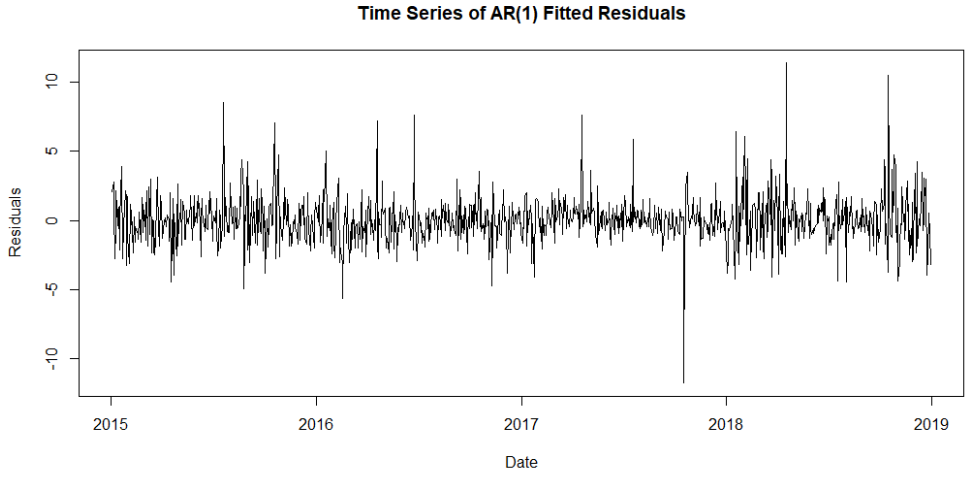
Fitting an AR(1) model, we get estimates of the parameters:

$$\widehat{\phi} = 0.9905, \quad \widehat{\mu} = 132.6845, \quad \widehat{\sigma}^2 = 2.915$$

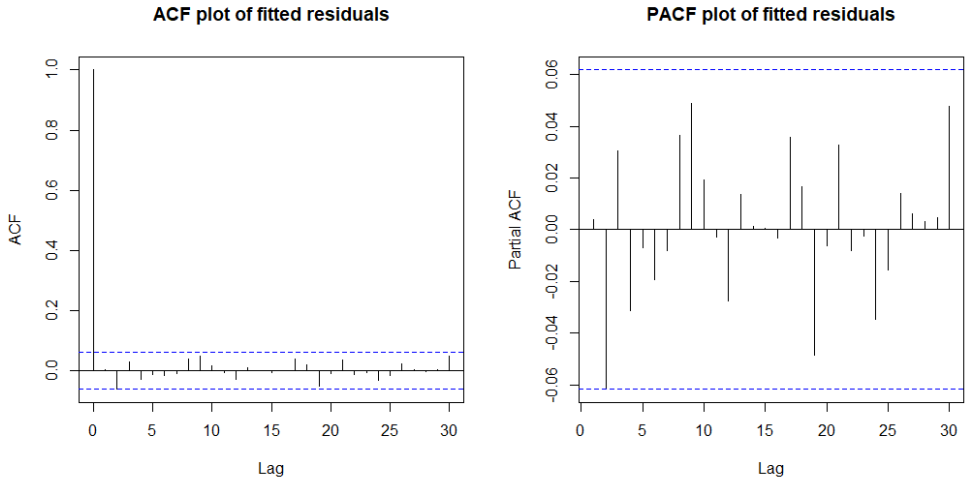
Hence we can write down the model as

$$Y_t - 132.6845 = 0.9905(Y_{t-1} - 132.6845) + \epsilon_t$$

where ϵ_t is WN(0,$\widehat{\sigma}^2$)=WN(0,2.915). Firstly, we should check to make sure that the residuals are indeed white noise.



Looking at the time series plot, we see approximately constant variance and mean with no apparent autocovariance, which supports the hypothesis that the residuals are white noise.



The ACF and PACF plots further support our hypothesis, as we see values staying inside the blue confidence band, indicating none or very little autocovariance. Finally, we use the Ljung-Box test to check independence and further confirm our hypothesis

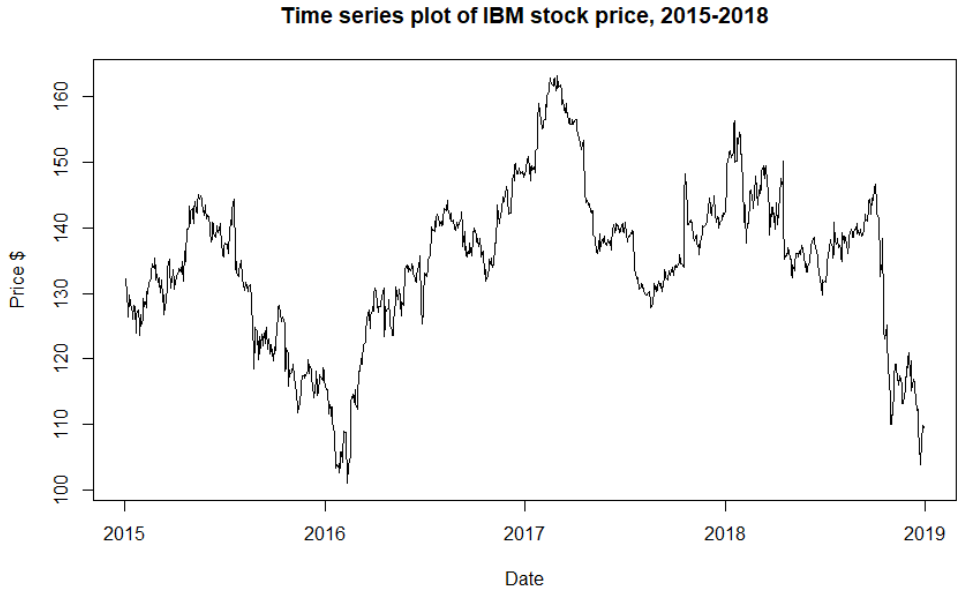
```
Box.test(fitRes, lag=5, type="Ljung-Box")

Output:
Box-Ljung test

data:  fitRes
X-squared = 5.5574, df = 5, p-value = 0.3517
```

The large p-value supports the null hypothesis of white noise.

5

Is the data set stationary? Support your claim with an appropriate test.
Answer:

```
plot(dates, prices, type="l", main="Time series plot of IBM stock price, 2015-2018",
xlab="Date", ylab="Price $")
```



Looking at the time-series plot, visually we can see some variation in the mean and variance giving some evidence of nonstationarity; however, such observations are not sufficient to make a conclusion. By looking at the ACF plot in problem 2, we see that there is significant autocovariance, but slow decay going to zero as lag increases; this may indicate nonstationarity, but the large spike in the first lag in the PACF suggests that an AR(1) model may be a good fit, and the ACF plot pattern display that of an AR(1) model with ϕ close to, but less than 1, which is stationary. Indeed, in problem 4 we saw that an AR(1) model of $\phi = 0.99$ was fitted to the data; however, we note that 0.99 is very close to 1 (also taking into account error), and an AR(1) model with $\phi \geq 1$ is nonstationary, hence this may be a sign of nonstationarity. Furthermore, fitting an arima model to the data using auto.arima() using AIC as a criterion (done later in problem 7), an AR(2) model is fit to the model with $\widehat{\phi}_1 = 0.9963$ and $\widehat{\phi}_2 = -0.0062$, and the roots of the polynomial $1 - \widehat{\phi}_1 x - \widehat{\phi}_2 x^2$ is given by

```
polyroot(c(1, -0.9963, 0.0062))
[1]   1.010063+0i 159.683486+0i
```

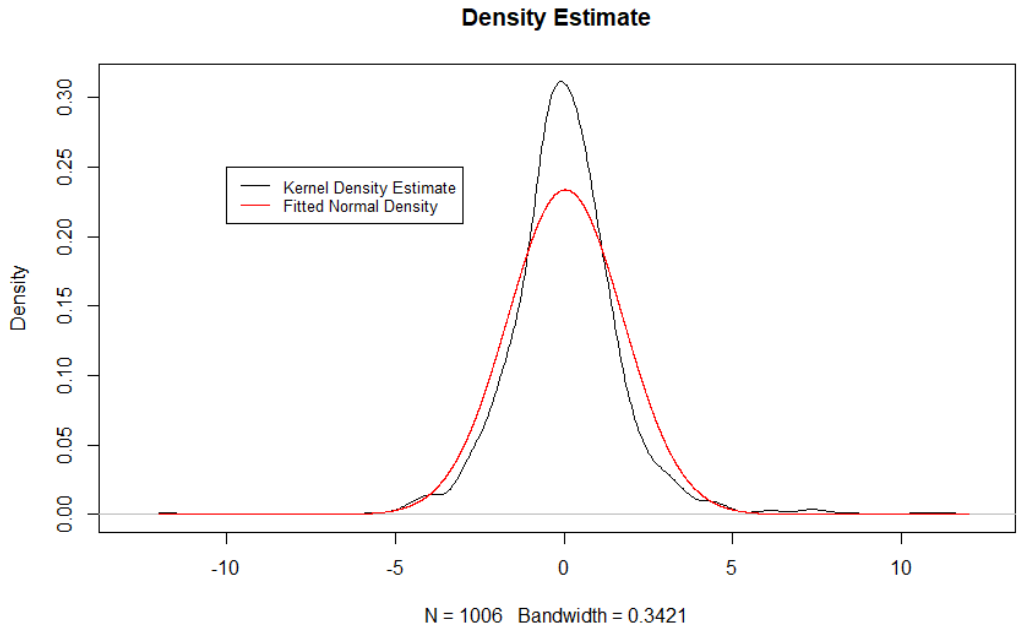
Although both of these roots have magnitude greater than zero, which is supposed to be a sign of stationarity, we see that the first root is 1.010063 which is very close to 1; taking error into account, this unit root test supports the nonstationarity of this data. Taking all of these tests and observations into account, we conclude that the data set is not stationary.

6

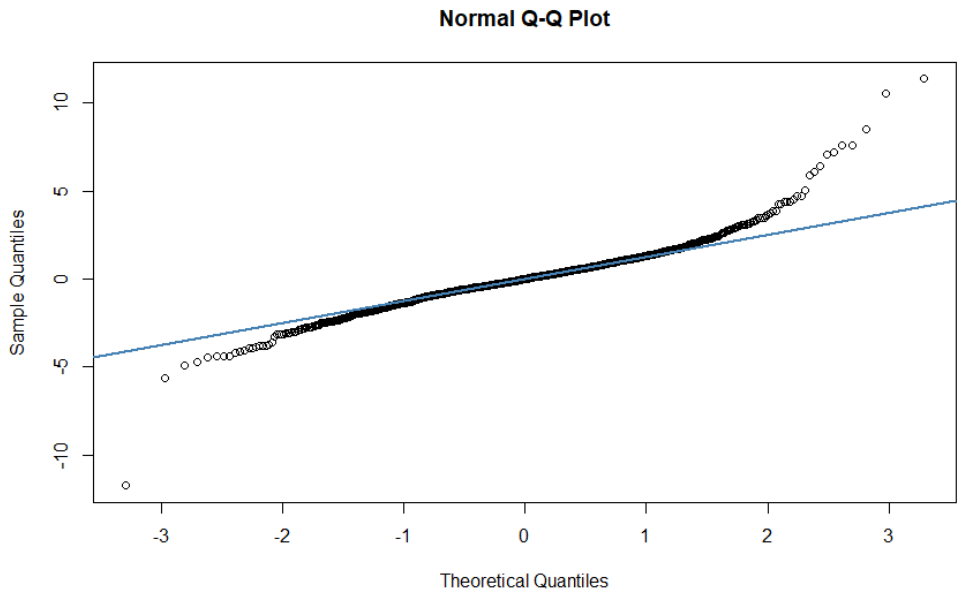
Is the noise ϵ_i in the AR(1) model Gaussian?
Answer:

```
#Plotting KDE and fitted normal distribution
d <- density(fitRes, adjust=1.2)
plot(d, main="Density estimate of IBM stock price")
legend(-10, 0.25, legend=c("Kernel Density Estimate", "Fitted Normal Density"),
col=c("black", "red"), lty=1:1, cex=0.8)
normfit = fitdistr(fitRes, densfun="normal")
fitMean = as.numeric(normfit$estimate[1])
fitSd = as.numeric(normfit$estimate[2])
xval = seq(-12, 12, length=10000)
yval = dnorm(xval, mean=fitMean, sd = fitSd)
lines(xval, yval, col="red", lwd=1.5)

#Normal QQ plot
qqnorm(fitRes)
qqline(fitRes, col = "steelblue", lwd=2)

#Shapiro-wilk test of normality
shapiro.test(prices)
```

We first look at the kernel density estimate of the residuals, and compare it to the density of a fitted normal distribution.



We see that the estimated empirical density is indeed symmetric, and has similar tails as the normal distribution. However, we see a much sharper and taller peak for the KDE compared to the fitted normal distribution; although sharper peaks are a characteristic of KDE's, the difference is drastic enough in the peaks and the shoulders to warrant concern. Furthermore, we look at the normal QQ plot:



We see although we see some linearity in the pattern, there is enough deviation from linearity to support non-normality. Furthermore, we observe a general concave-convex pattern, indicating that the underlying distribution of the residuals may be lighter tailed than a normal distribution. From these, it appears that the residuals are not Gaussian; to further support this, we run a Shapiro-Wilk test for normality.

```
shapiro.test(prices)

Output:
Shapiro-Wilk normality test

data:  prices
W = 0.98142, p-value = 5.007e-10
```

We see that the p-value is $5.007e^{-10}$ which is very small, hence we reject the null hypothesis that the data is normally distributed. This further supports our conclusion that the residuals are not Gaussian.

7

Based on AIC what is the best ARIMA model? Write down the model with the fitted parameters.

Answer:

```
auto.arima(prices, max.p = 20, max.q = 0, d=0, ic="aic")
fitAR2 = arima(prices, order=c(2,0,0), optim.control=list(maxit = 1000))
print(fitAR2)
fitRes2 = residuals(fitAR2)
acf(fitRes2)
pacf(fitRes2)
Box.test(fitRes2, lag=10, type="Ljung-Box")
```

Fitting an ARIMA model to the IBM stock prices using auto.arima(), we get

```
print(fitAR2)

Output:
Series: prices
ARIMA(2,0,0) with non-zero mean

Coefficients:
ar1      ar2      mean
0.9963  -0.0062  133.9117
s.e.  0.0315   0.0316    5.0050

sigma^2 estimated as 2.924:  log likelihood=-1967.6
AIC=3943.19   AICc=3943.23   BIC=3962.85
```

thus based on AIC, an AR(2) model is the best ARIMA model fitting the data. The fitted/estimated parameters of this AR(2) model are

$$\widehat{\phi}_1 = 0.9963, \quad -\widehat{\phi}_2 = -0.0062, \quad \widehat{\mu} = 133.0117, \quad \widehat{\sigma}^2 = 2.924$$

Hence we can write the process $\{Y_t\}$ as

$$Y_t = (1 - \widehat{\phi}_1 - \widehat{\phi}_2)\widehat{\mu} + \widehat{\phi}_1 Y_{t-1} + \widehat{\phi}_2 Y_{t-2} + \epsilon_t = 1.3181 + 0.9963 Y_{t-1} - 0.0062 Y_{t-2} + \epsilon_t$$

8

Fit a t-distriution to the noise ϵ_t for the ARIMA model.
Answer:

```
tFit2 = fitdistr(fitRes2, densfun="t", start =list(m=mean(fitRes2), s=sd(fitRes2), df=3),
lower=c(-1,0.001,1))
print(tFit2)

Output:
m             s             df
-0.01819289    1.13659041    3.47206905
( 0.04308557) ( 0.04603383) ( 0.41089425)
```

Fitting a t distribution to the residuals of the AR(2) model, we estimate a $t_\nu(\mu, \sigma^2)$ distribution with estimated parameters $\widehat{\nu} = 3.4721$, $\widehat{\sigma} = 1.13659041$, and $\widehat{\mu} = -0.01819$.

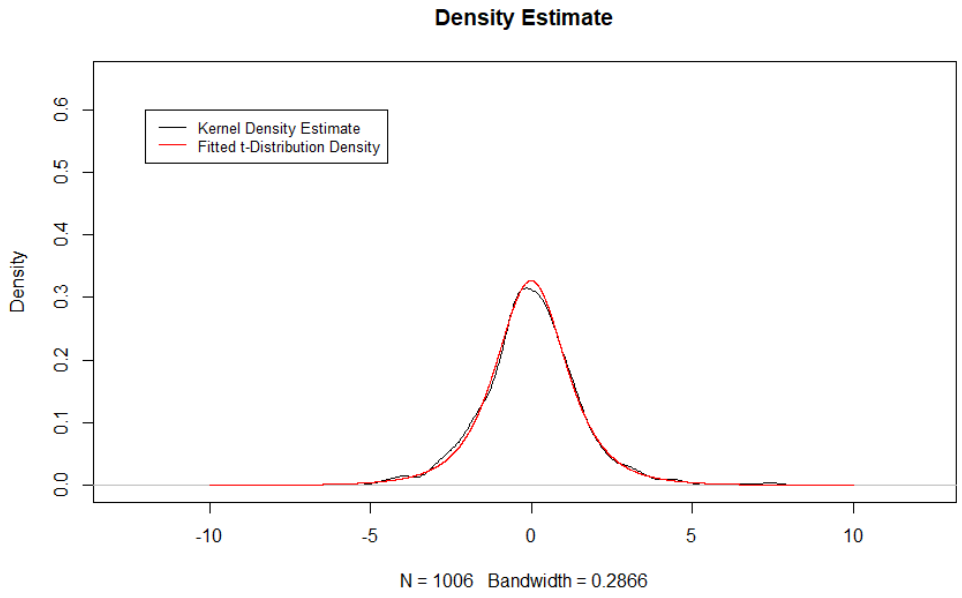
9

Evaluate how well the fitted t-distribution by comparing emprical density with fitted t-density functions (in the same plot and label properly) and a qunatile plot.
Answer:

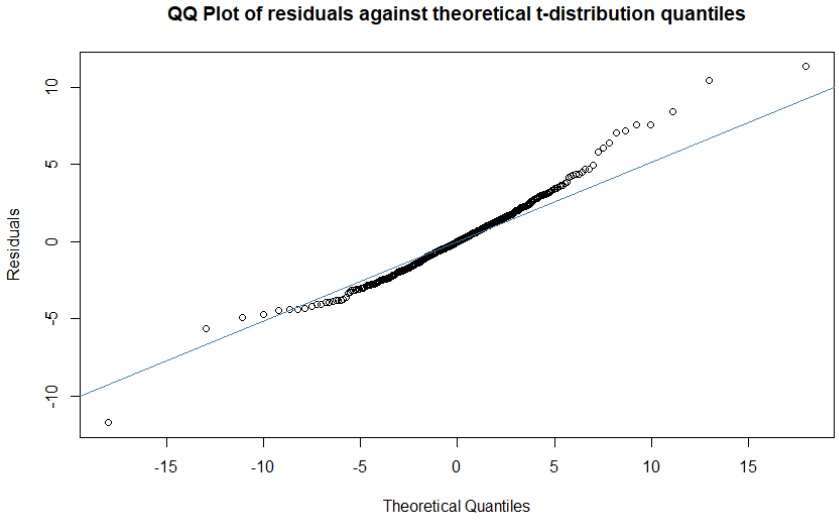
```
library(sgt)
tMean <- as.numeric(tFit$par[1])
tSd <- as.numeric(tFit$par[2])
tNu <- as.numeric(tFit$par[3])

#Extracting parameters
tMean2 <- as.numeric(tFit2$estimate[1])
tSd2 <- as.numeric(tFit2$estimate[2])
tNu2 <- as.numeric(tFit2$estimate[3])

#Density Estimate
d1 <- density(fitRes2, adjust=1)
```

```
plot(d1, lwd=1.5, col="black", ylim=c(0,0.65), main="Density Estimate")
legend(-12, 0.6, legend=c("Kernel Density Estimate", "Fitted t-Distribution Density"),
col=c("black", "red"), lty=1:1, cex=0.8)
xval1 = seq(-10, 10, length=10000)
yval1 = dsgt(xval1, mu=tMean, sigma=tSd, lambda=0, p=2, q=tNu/2)
lines(xval1, yval1, col="red", lwd=1.5)

##QQPLOT
scaledQuan = qsgt(pppoints(1006))
qqplot(scaledQuan, fitRes2, xlab="Theoretical Quantiles", ylab="Residuals", main="QQ Plot of res
qqline(scaledQuan, fitRes2, col="steelblue", lwd=1.5)
```

First we look at the kernel density estimate of the residuals, and compare it to the fitted t-distribution density



We see that that fitted t distribution very closely follows the shape of the kernel density estimate, except with slight difference in the peak (possibly due to slight skew in the data, which is natural since this comes from real data, thus with only 1000 data points cannot be expected to be perfectly symmetric), and some minor bumps in the tails/shoulders which is expected. This indicates that the t-distribution is a good fit to the residuals. Furthermore, we look at the QQ plot comparing the residuals to theoretical t-distribution quantiles:



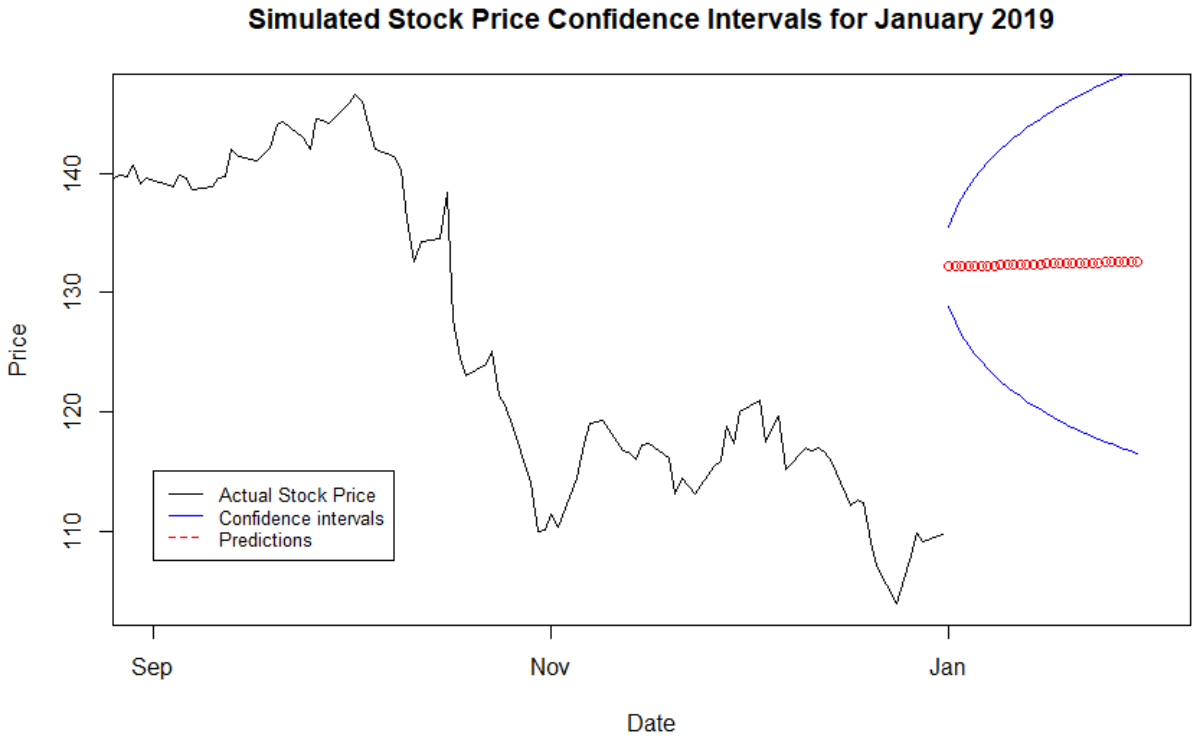
The theoretical quantiles used were that of the fitted t-distribution. Again, we see a generally linear trend, supporting our conclusion from the density estimate plots that the t distribution is a good fit. However, we slight deviance from linearity in either extremes, displaying some concavity/convexity. This may be an indication that perhaps the tail weight needs to be find tuned in the fit, although the t-distribution seems to still be a good idea.

10

Based on the previous results, use Model-based resampling to simulate a 95% confidence band forecasting the month of January 2019. Write down a detailed description for each step and plot the confidence band.

Answer:

```
#Forecasting 30 days based on fitted model
predictions = predict(fitAR2, n.ahead=30)

#Plotting the last 150 days of the time series
lastDates = dates[c(1:150)]
lastPrices = prices[c(1:150)]
plot(lastDates, lastPrices, type='l', xlim=as.Date(c("2018-05-01","2019-02-01")))

#Plotting predictions and confidence intervals
dateseq = seq(as.Date("2019/01/01"), by = "day", length.out = 30)
addDates = c(lastDates, dateseq)
lines(dateseq, predictions$pred, col="red")
lines(dateseq, predictions$pred + 1.96*predictions$se, col="blue")
lines(dateseq, predictions$pred - 1.96*predictions$se, col="blue")-
```



We used the predict() function to simulate a 95% confidence band forecasting the month of January 2019. The predict function takes in the fitted model and generates predictions and standard error, which allowed us to compute confidence bands for the predictions; this could be done because in the previous questions, we had verified that the reisduals ϵ_t of the model is well fitted to a t-distribution, so we can get our confidence intervals for the predictions by

$$\widehat{x} \pm \widehat{s} * z_{\alpha/2}$$

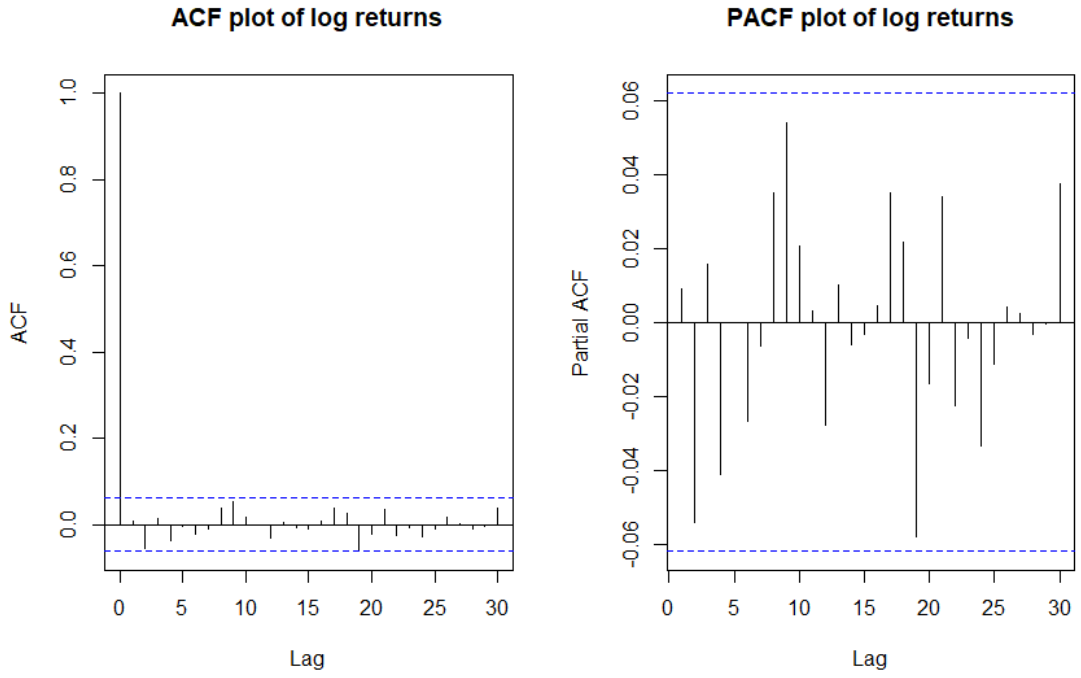
where $\widehat{x}$ is our predicted value, $\widehat{s}$ the standard errror, and z_α quantiles of the standard Gaussian. Since the arima fit also computed a standard error, we were able to do this. Although the predictions appears to be off and much higher than where the real data ends, this is because of a recent dip, while on average the time series was higher, near where the prediction is.

11

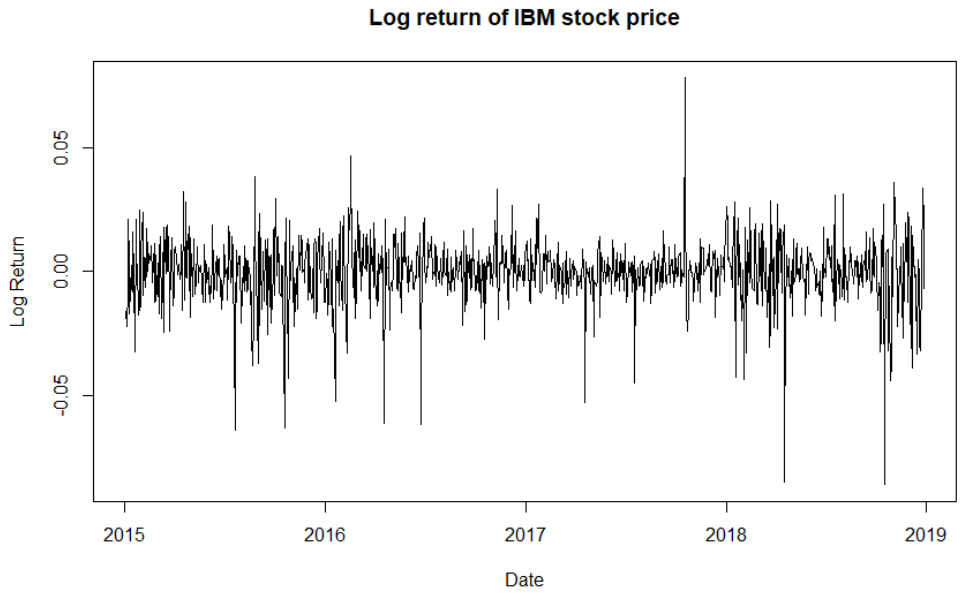
Instead of Price use the log return of the data set to plot ACF and PACF and emprical density. Comment on why log returns are widely used in financial time series analyses over raw asset prices analysis.

```
#Calculating log returns
returns = -diff(prices)/prices[-length(prices)]
logReturns = log(1+returns)

par(mfrow=c(1,2))
acf(logReturns, main="ACF plot of log returns")
pacf(logReturns, main="PACF plot of log returns")

#Time series plot of log returns
plot(dates[-c(1)], logReturns, type="l", main="Log return of IBM stock price", xlab="Date", ylab
```

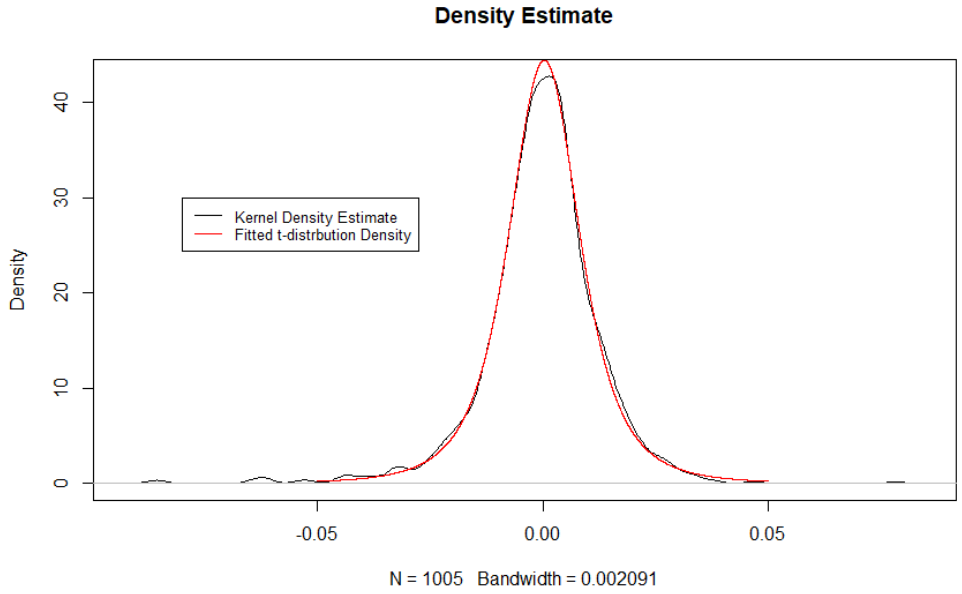
```r
#KDE of the log returns
d2 <- density(logReturns, adjust=1)
plot(d2, lwd=1.5, col="black", main="Density Estimate")

#Fitting a t-distribution to log returns
tFit3 = stdFit(logReturns)
fitMean3 = tFit3$par[1]
fitSd3 = tFit3$par[2]
fitNu3 = tFit3$par[3]

#Comparing fitted t-distribution with KDE
xval3 = seq(-0.05, 0.05, length=10000)
yval3 = dsgt(xval3, mu=fitMean3, sigma=fitSd3, lambda=0, p=2, q=fitNu3/2)
lines(xval3, yval3, col="red", lwd=1.5)
legend(-0.08, 30, legend=c("Kernel Density Estimate", "Fitted t-distrbution Density"),
col=c("black", "red"), lty=1:1, cex=0.8)
```



We see from in the ACF plot that there is no significant autocovariance at any positive lags, as all of the spikes except at lag 0 are well within the blue confidence band. We also see in the PACF that there are no spikes outside of the confidence band. These are good indicators of stationarity of the log returns, which is evident when looking at a time series plot of the data.



We see that the mean appears to be constant throughout the time series, and the variance, except for some spikes, remains relatively constant. Thus, there is strong support that the log returns are weakly stationary. Furthermore, we look at the empirical density estimate and compare it with a fitted t-distribution



We note that the estimated density is very similar to the fitted t-distribution, thus giving strong support to the hypothesis that the underlying distribution of the log returns is a t-distribution.
Based on these characteristics, we see that the log returns are a much nicer object to work with compared to raw asset prices, in the sense of processes. Raw asset prices are often not stationary and have much more complicated behavior, requiring a more complex model to try and model the data. However, the log returns are often stationary, and follow a t-distribution; both of these characteristics make analysis much easier.