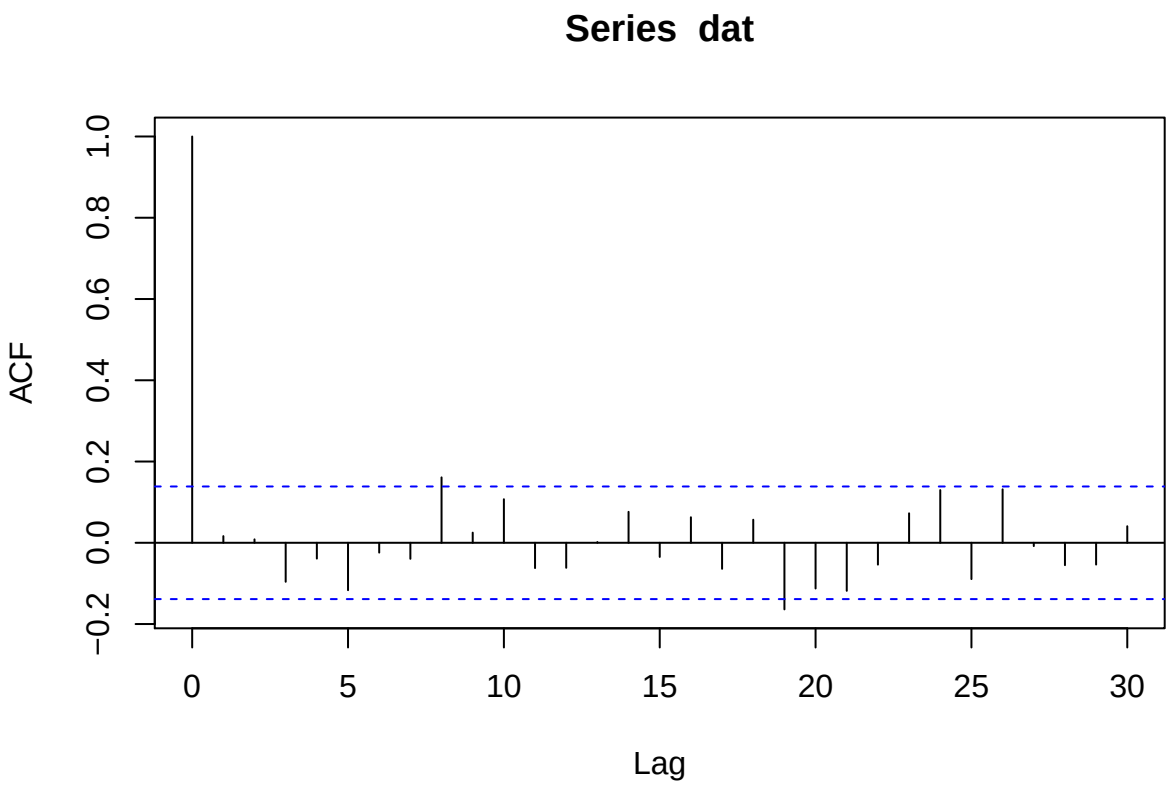
STAT443_Assignment1_Yuting_Wen

yuting wen

3/9/2020

1. (10 marks) This question concerns a test to determine whether it seems likely that a time series is a realization of a white noise process. As usual, we assume we have observed a series x(1);. . . ;x(N) from a stochastic process X (t); t = 1;2;. . . Use the rnorm command in R to generate an i.i.d. series of size 200 from the N (0; 22) distribution. Find the acf of your data and find out how to extract the values from R up to a given lag (you will find help(acf) useful).

```
set.seed(443)
N <- 200
dat <- rnorm(N,mean=0,sd=2)
dat_acf <- acf(dat, lag.max=30)
```



(a) One test for white noise is the portmanteau lack-of-fit test. Perform this test for two different values of M for the data you generated, selecting the values of M at random between 15 and 30 inclusive. Quote the P-value of the test on each occasion, and comment on your results.

```
dat_acf <- dat_acf$acf
dat_acf <- dat_acf^2
M <- sample(15:30, 2)
# the M we use
show(M)
```

```
## [1] 27 15
```

```
Q1 <- N*sum(dat_acf[2:(M[1]+1)])
Q2 <- N*sum(dat_acf[2:(M[2]+1)])
```

```r
P1 <- pchisq(Q1,M[1], lower.tail = FALSE)
P2 <- pchisq(Q2,M[2], lower.tail = FALSE)
# P values we got
show(P1)
```

```
## [1] 0.06394779
```

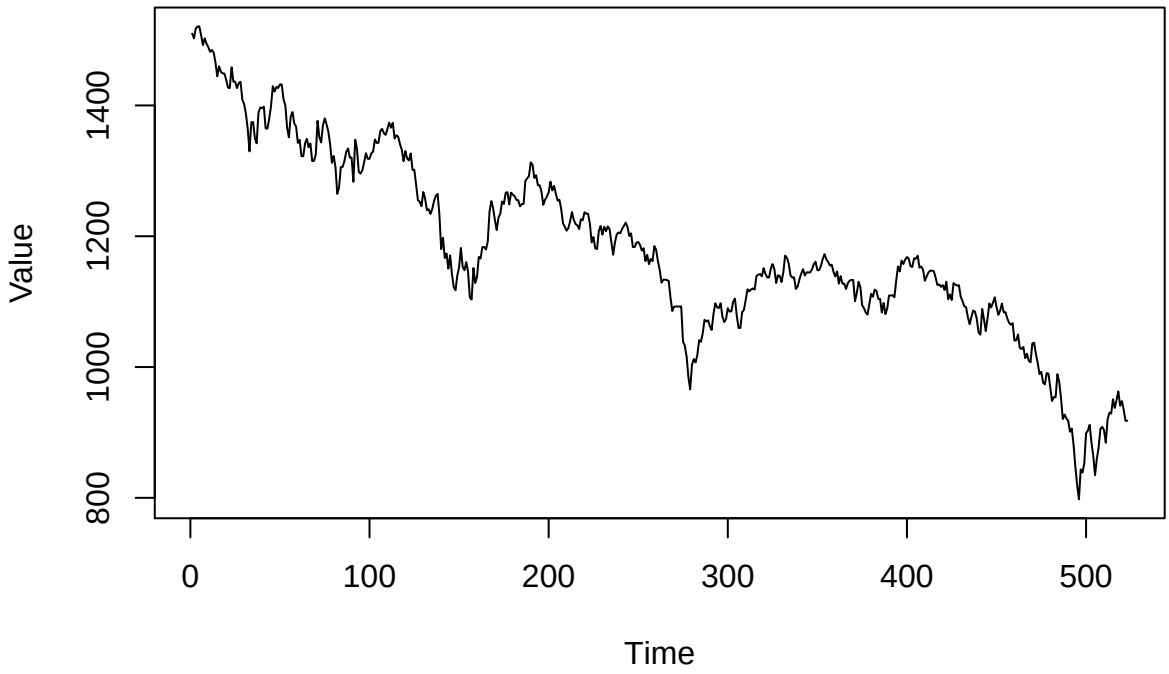
```r
show(P2)
```

```
## [1] 0.3874471
```

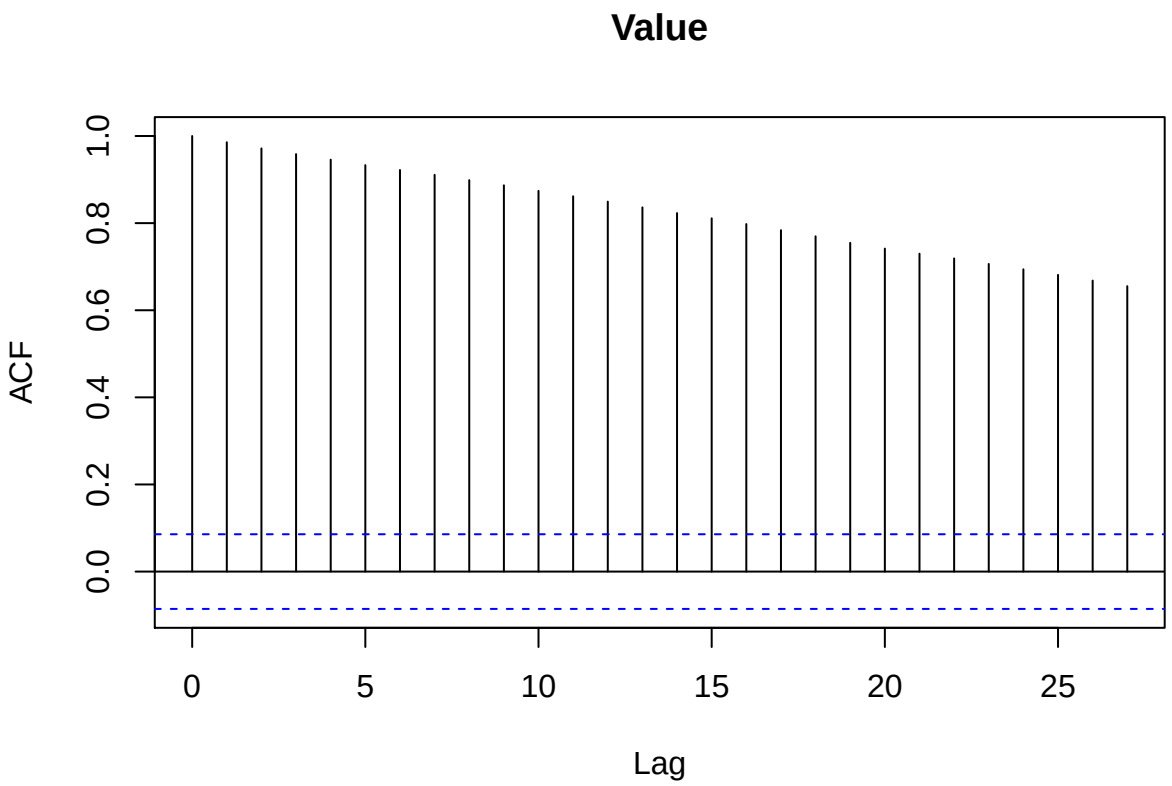
Since the P-value we got are both larger than 0.05, indicates weak evidence against the null hypothesis, so we fail to reject the null hypothesis, hence we say that the observations are from white noise process.

(b) The data SP500.txt contains 523 consecutive closing values for Standard and Poor's 500 Index. Read the data set into R, and coerce the data into a time series object. Create a plot of the data, and of the acf of the series. Comment on what you observe.

```
dat_SP <- read.delim("~/Desktop/STAT443/SP500.TXT")
dat_SP <- as.ts(dat_SP)
plot(dat_SP)
```




```
dat_SP_acf <- acf(dat_SP)
```



Observation: - plot: it has decreasing trend, with clear cyclical effect. - acf: looks non-stationary, since it decays very slowly, and are all above significance level.

(c) Use a portmanteau lack-of-fit test with M = 25 to decide whether the series appears to be a realization from a white noise process.

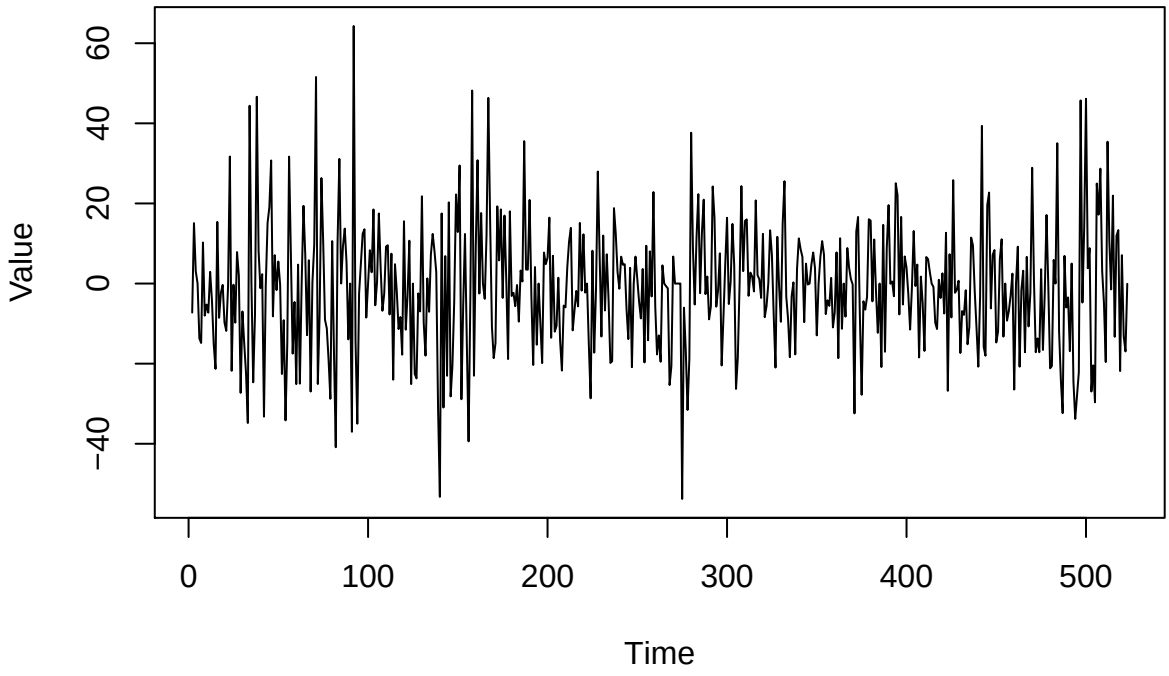
```
M <- 25
dat_SP_acf <- dat_SP_acf$acf
dat_SP_acf <- dat_SP_acf^2
Q <- length(dat_SP)*sum(dat_SP_acf[2:M+1])
P <- 1-pchisq(Q,M)
# P value we got
P
```

```
## [1] 0
```

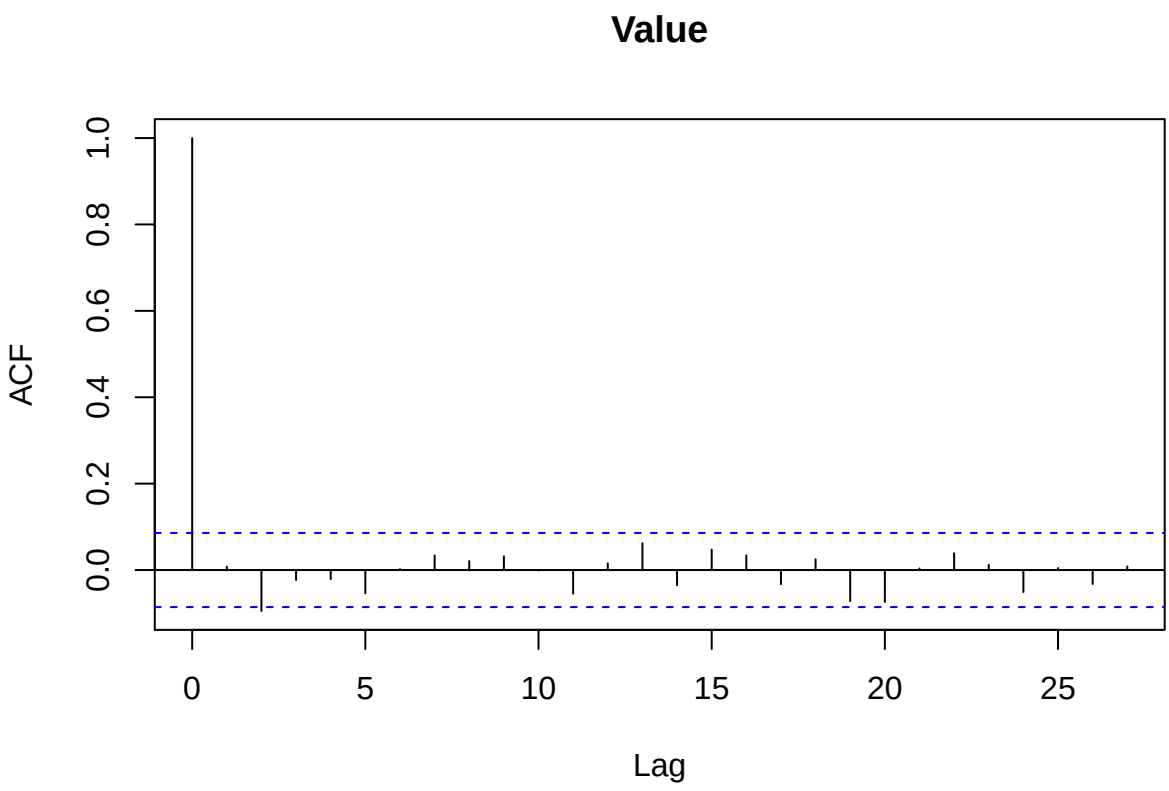
Conclusion: Since the P-value (0) is smaller than 0.05, indicates strong evidence against the null hypothesis, hence we say that the data is not from a white noise process.

(d) Apply an operator to the SP500 series that might reasonably be expected to remove the non-stationary component. Plot the new series and its acf, and comment.

```
# we take first difference
dat_SP_1 <- diff(dat_SP,lag=1)
plot(dat_SP_1)
```




```
dat_SP_1_acf <- acf(dat_SP_1)
```



Operator: take first difference as new time series Observation: - plot: oscillates up and down, also looks random - acf: matches with white noise process since acf cuts off at lag = 1

(e) Use a portmanteau lack-o-fit test with M = 25 to decide whether the series you created in (d) appears to be a realization from a white noise process.

```
M <- 25
dat_SP_1_acf <- dat_SP_1_acf$acf
dat_SP_1_acf <- dat_SP_1_acf^2
Q <- length(dat_SP_1)*sum(dat_SP_1_acf[2:M+1])
P <- 1-pchisq(Q,M)
# P value we got
P
```

```
## [1] 0.5852165
```

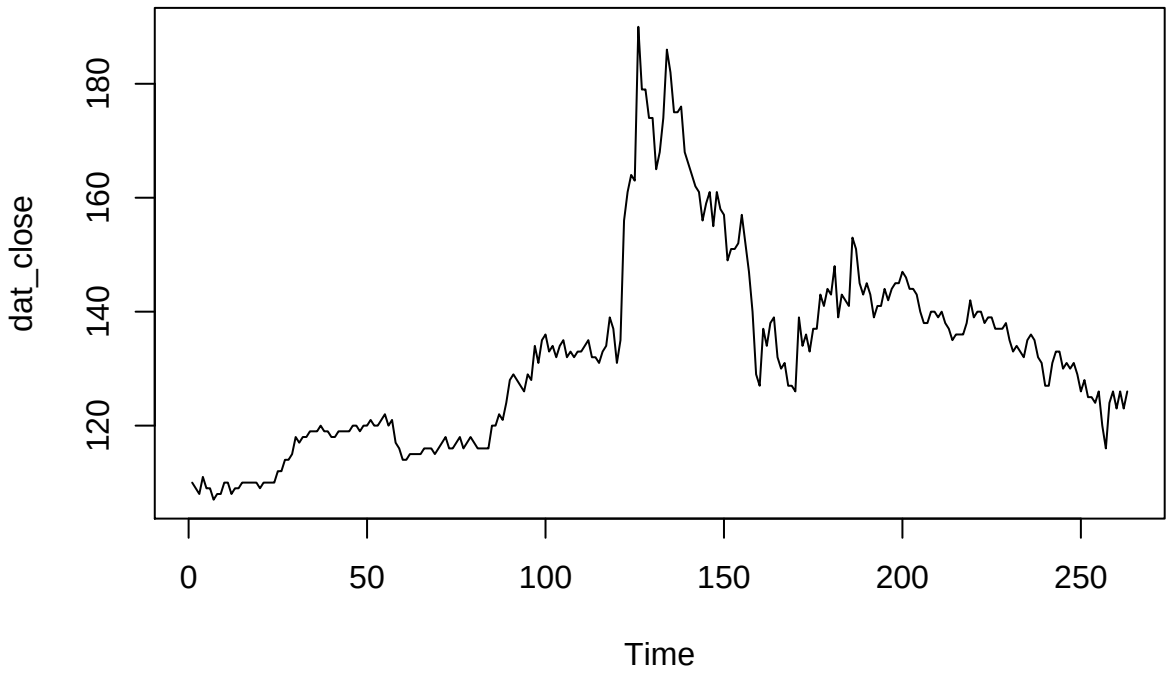
Since the P-value we got is larger than 0.05, indicates weak evidence against the null hypothesis, so we fail to reject the null hypothesis, hence we say that the data with operator is from white noise process.

2. (10marks)The variable Close in the data file FeedOneCloseJan15Feb16 gives closing prices (in Yen) of Feed One Co. Ltd., as listed on the Tokyo Stock Exchange, 5th January 2015 to 29th January 2016.

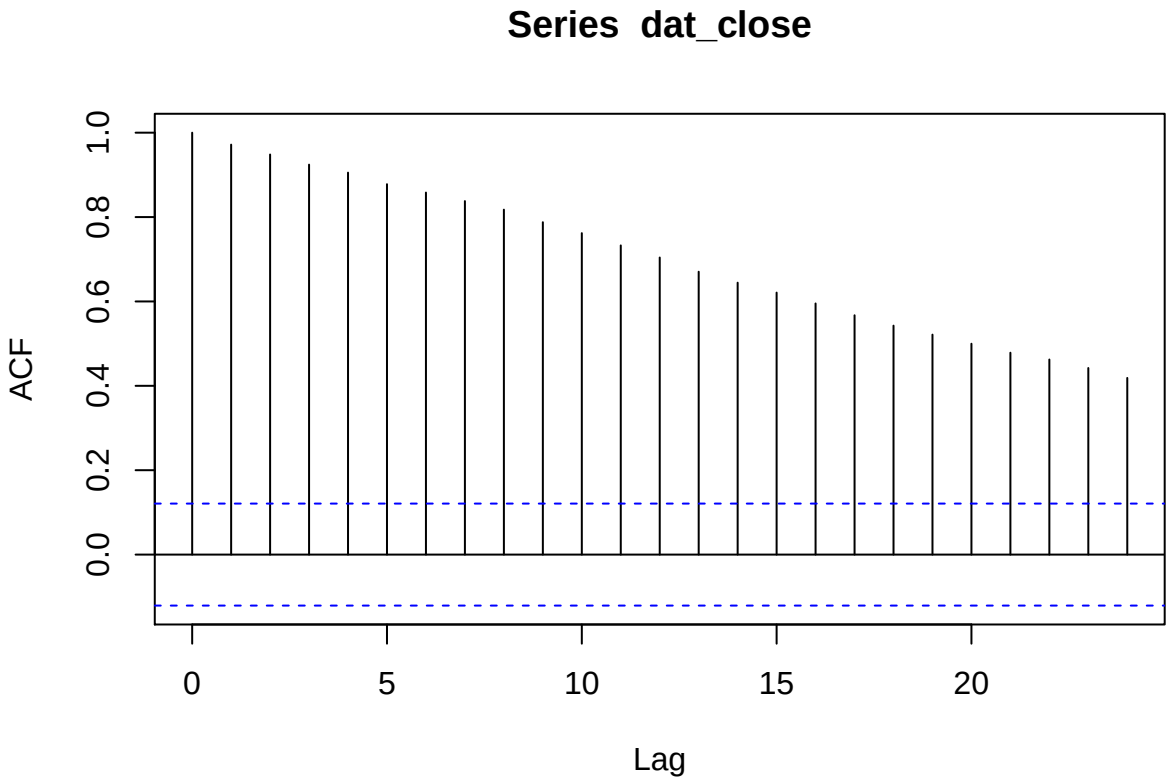
```
dat_close <- read.csv("~/Desktop/STAT443/FeedOneCloseJan15Feb16.csv")
# reverse the data since the order is reversed
dat_close <- dat_close[nrow(dat_close):1,]
dat_close <- dat_close$Close
```

(a) Plot the time series and the acf of the series, and comment on what you observe. Does the closing price time series appear to be stationary?

```
dat_close <- as.ts(dat_close)
plot(dat_close)
```



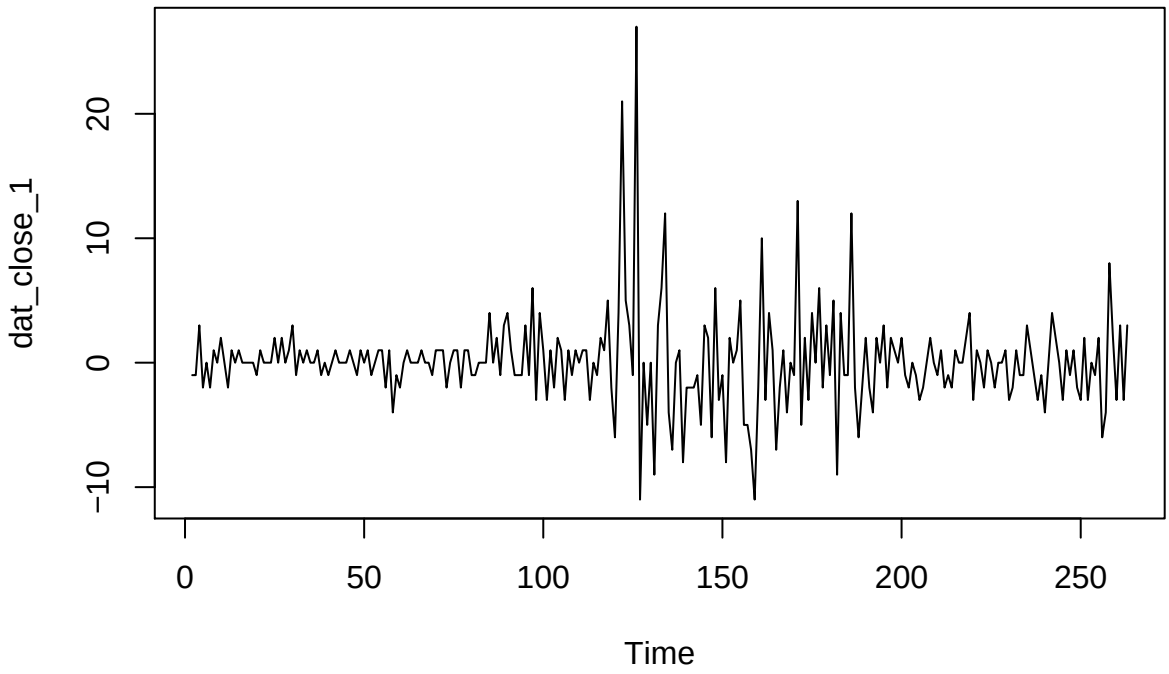

```
acf(dat_close)
```



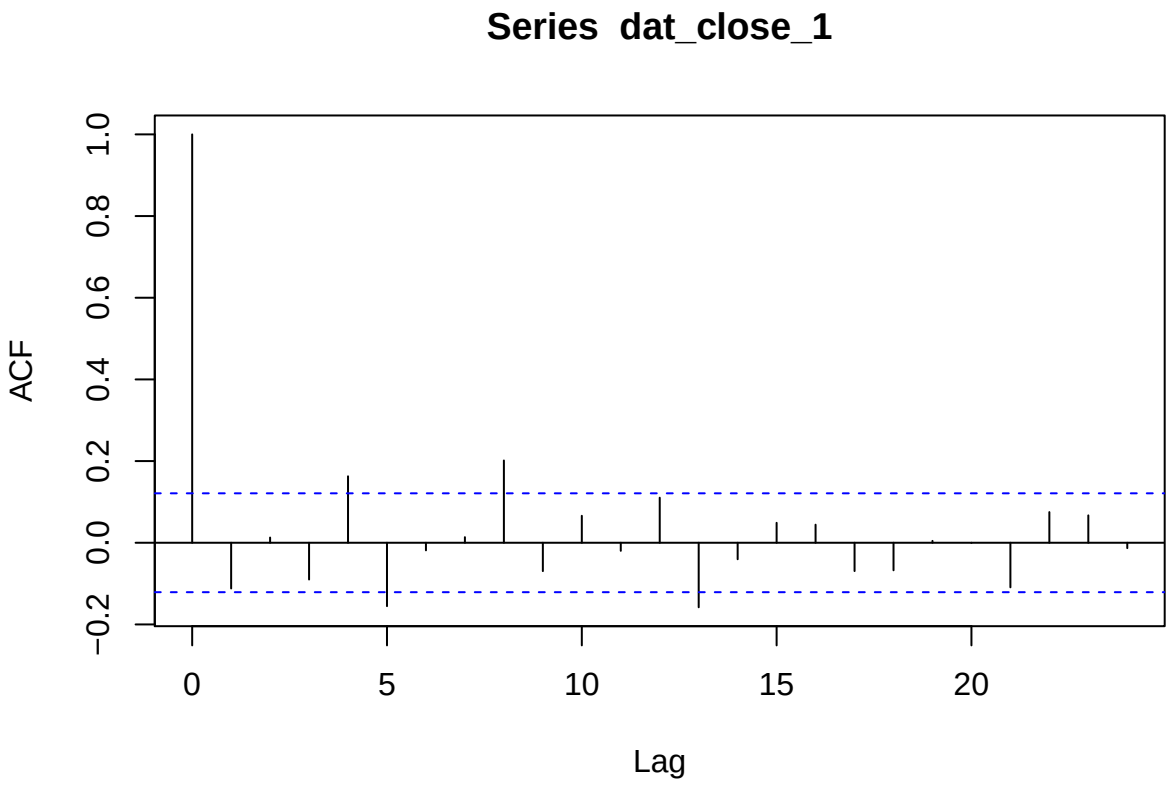
Observations: - plot: it shows increasing then decreasing trend with a peek, also shows cyclical effect - acf: non-stationary, since acf decays very slowly, and are above significance level for a large lag

(b) Apply the first difference operator to the time series. Plot the new series, the acf, and pacf of the new series, and comment on their patterns. Does the new series appear to be stationary?

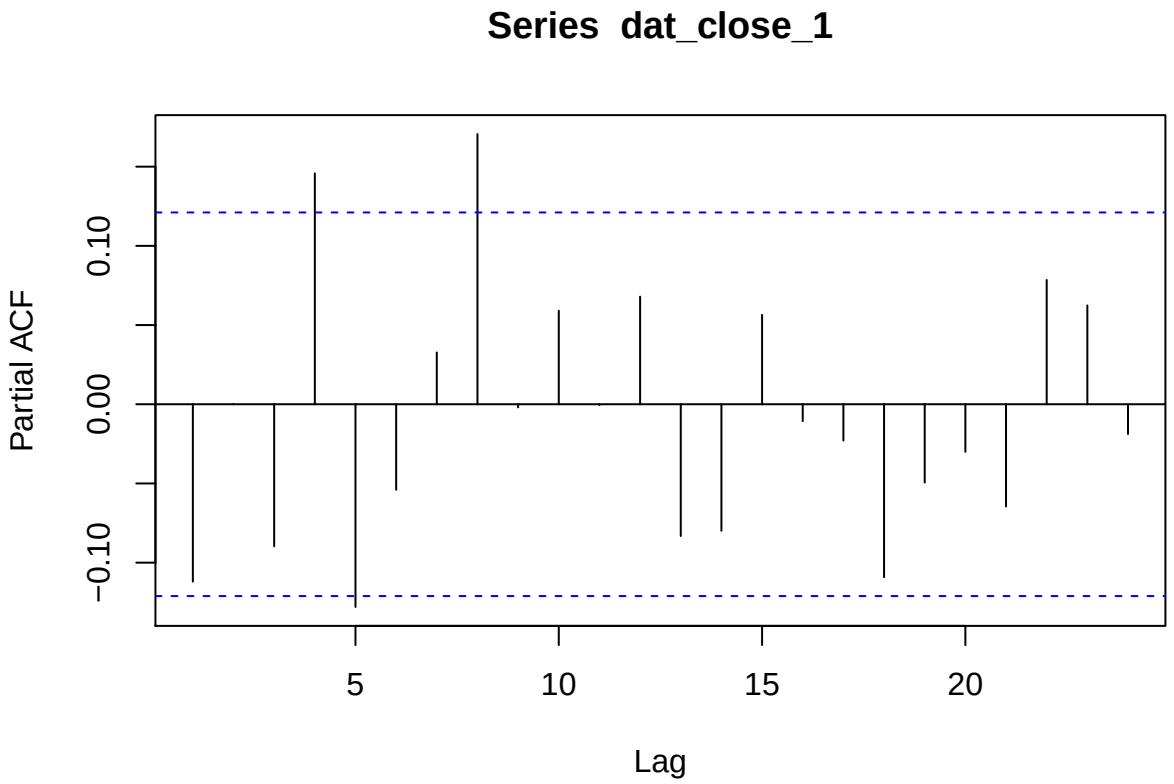
```
dat_close_1 <- diff(dat_close, lag=1)
plot(dat_close_1)
```




```
acf(dat_close_1)
```



```
pacf(dat_close_1)
```



Observations: - plot: oscillates up and down, no additional effect shown - acf: decays quickly, looks stationary, some values are above significance level at some lags - pacf: shows down and up flow, some values are above significance level at some lags

(c) Fit various models from the ARMA(p,q) family to the new series. You should only consider models for which p + q <= 6; and you should state which three models you would consider to be the best options and clarify why you chose those three models. Write out your three possible models in full, providing all the parameter estimates.

```
# we went through all combinations with checking their Ljung-Box stat
# since we have too many graphs here,
# I chose to set eval=FALSE of this R chunk to hide the result,
# you can set it to true if interested in the graphs
for (p in 0:6) {
  for (q in 0:6) {
    if (p+q <= 6) {
      model <- arima(dat_close_1, order=c(p,0,q), include.mean=TRUE)
      tsdiag(model)
    }
  }
}
```

based on P-value for Ljung-Box statistic, some p-q combiantions models have p-values that are lower than significance level, which means we reject the null hypothesis that residuals follow white noise process:

we reject those p-q combinations

p	q
0	0
0	1
0	2
0	3
0	4
1	0
1	1
1	2
2	0
2	1
2	2
3	0

```
# then we went through models after filtering to check their aic, log-likelihood, sigma^2 stats
paramList <- list()
tmp <- 1
for (p in 0:6) {
  for (q in 0:6) {
    if ((p==0&q==5)|(p==0&q==6)|
        (p==1&q==3)|(p==1&q==4)|(p==1&q==5)|
        (p==2&q==3)|(p==2&q==4)|
        (p==3&q==1)|(p==3&q==2)|(p==3&q==3)|
        (p==4&q==0)|(p==4&q==1)|(p==4&q==2)|
        (p==5&q==0)|(p==5&q==1)|
        (p==6&q==0)) {
      model <- arima(dat_close_1, order=c(p,0,q), include.mean=TRUE)
      paramList[[tmp]] <- c(p,q,model$aic,model$loglik,model$sigma2)
      tmp <- tmp+1
    }
  }
```

```
}
# paramList: p, q, aic, log-likelihood, sigma^2
# uncomment next line if you want to see the aic result
# paramList
```

p	q	aic
3	3	1433.117
3	1	1438.96720
5	0	1438.59993

we chose there models since they have the lowest aic values (A good model is the one that has minimum AIC among all the other models)

```
# Parameters estimates of model ARMA(3,3)
arima(dat_close_1, order=c(3,0,3), include.mean=TRUE)
```

```
##
## Call:
## arima(x = dat_close_1, order = c(3, 0, 3), include.mean = TRUE)
##
## Coefficients:
##           ar1      ar2      ar3     ma1     ma2     ma3  intercept
##       -1.0998  -1.1029  -0.8561  1.0160  1.0185  0.7011     0.0566
## s.e.   0.0788   0.0838   0.0703  0.1092  0.1207  0.0963     0.2055
##
## sigma^2 estimated as 13.05:  log likelihood = -708.56,  aic = 1433.12
```

parameter estimates:

ar1	ar2	ar3	ma1	ma2	ma3	intercept
-1.0998	-1.1029	-0.8561	1.0160	1.0185	0.7011	0.0566

```
# Parameters estimates of model ARMA(3,1)
arima(dat_close_1, order=c(3,0,1), include.mean=TRUE)
```

```
##
## Call:
## arima(x = dat_close_1, order = c(3, 0, 1), include.mean = TRUE)
##
## Coefficients:
##           ar1      ar2      ar3     ma1  intercept
##       -0.8720  -0.0976  -0.1207  0.7900     0.0595
## s.e.   0.1176   0.0817   0.0659  0.1043     0.1951
##
## sigma^2 estimated as 13.57:  log likelihood = -713.48,  aic = 1438.97
```

parameter estimates:

ar1	ar2	ar3	ma1	intercept
-0.8720	-0.0976	-0.1207	0.7900	0.0595

```r
# Parameters estimates of model ARMA(4,2)
arima(dat_close_1, order=c(5,0,0), include.mean=TRUE)
```

```
##
## Call:
## arima(x = dat_close_1, order = c(5, 0, 0), include.mean = TRUE)
##
## Coefficients:
##           ar1      ar2      ar3     ar4      ar5  intercept
##       -0.0796  -0.0167  -0.0759  0.1345  -0.1289     0.0590
## s.e.   0.0612   0.0607   0.0606  0.0607   0.0610     0.1945
##
## sigma^2 estimated as 13.45:  log likelihood = -712.3,  aic = 1438.6
```

parameter estimates:

ar1	ar2	ar3	ar4	ar5	intercept
-0.0796	-0.0167	-0.0759	0.1345	-0.1289	0.0590

(d) An alternative, and modern, way of model selection is to split the data into a training set and a test set. The idea is that we use the training set to determine fit and explore competing models, and then assess how well the models perform when fitting values in the test set. A popular criterion to adopt in assessing a model's performance on the test set is mean squared error.

Using the differences in January 2016 closing prices as the test set for your models and the remaining data as the training set, compare the three models you selected in (c) for performance over the test set as above.

```
# sets up training set and test set
# first 18 values are test set
m <- 18
train <- 1:(length(dat_close_1)-m)
trainx <- dat_close_1[train]
testx <- dat_close_1[-train]

#fits model to training set,
# uses model to predict test set,
# computes squared errors over test set

# model 1: p=3, q=3
model1 <- arima(trainx, order = c(3, 0, 3))
foremodel1 = predict(model1, m)
error1 <- sum((testx - foremodel1$pred)^2)

# model 2: p=3, q=1
model2 <- arima(trainx, order = c(3, 0, 1))
foremodel2 = predict(model2, m)
error2 <- sum((testx - foremodel2$pred)^2)

# model 3: p=5, q=0
model3 <- arima(trainx, order = c(5, 0, 0))
foremodel3 = predict(model3, m)
error3 <- sum((testx - foremodel3$pred)^2)

# MSE of model 1
error1
```

```
## [1] 181.4014
```

```
# MSE of model 2
error2
```

```
## [1] 193.3461
```

```
# MSE of model 3
error3
```

```
## [1] 193.7725
```

As we can see, model 3 (p=3, q=3) has the lowest test error, hence we say ARMA(3,3) fits the first difference data best.

(e) Use your winning model from (d), when fitted to the entire series, to forecast the next two values in the differenced series. Hence forecast the next two values in the closing price series.

```
model_full <- arima(dat_close_1, order=c(3,0,3))
forecast_new <- predict(model_full, 2)
forecast_new
```

```
## $pred
## Time Series:
## Start = 264
## End = 265
## Frequency = 1
## [1] -1.8510896  0.9817795
##
## $se
## Time Series:
## Start = 264
## End = 265
## Frequency = 1
## [1] 3.612919 3.625583
```

based on the forecast values

next 2 values in the differenced series

next 1 value	next 2 value
- 1.8510896	0.9817795

next 2 values in the closing price series:

next 1 value	next 2 value
126 - 1.8510896 = 124.1489	124.1489 + 0.9817795 = 125.1307