FTS-HomeWork4

Name: Venkata Avinash

We have run seasonal arma model on the unemployment rate data. Now consider adding the weekly initial jobless claim to predict the monthly unemployment rates. The data is from 1967 to 2010. The data is given in "munrateic.txt". As shown below, the column "rate" denotes the unemployment rate, and the 5th to 8th column denotes the weekly jobless claims from week 1 to week 4 of the month and the last column-"icm1" is the total number of initial jobless claim of the month.

Question1. Perform a simple regression of unrate with the monthly claim numbers using the following commands. Comment on the results of the regression and the acf/pacf plots.

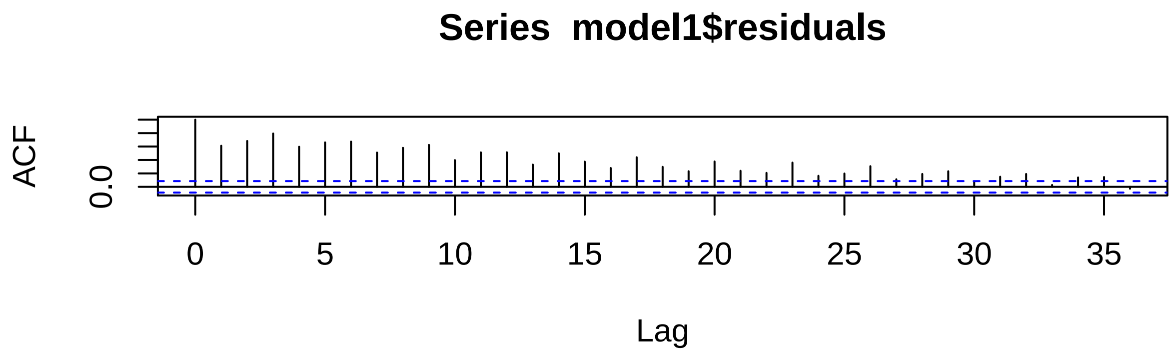
```
data=read.table("m-unrateic.txt", header=T)
head(data)
```

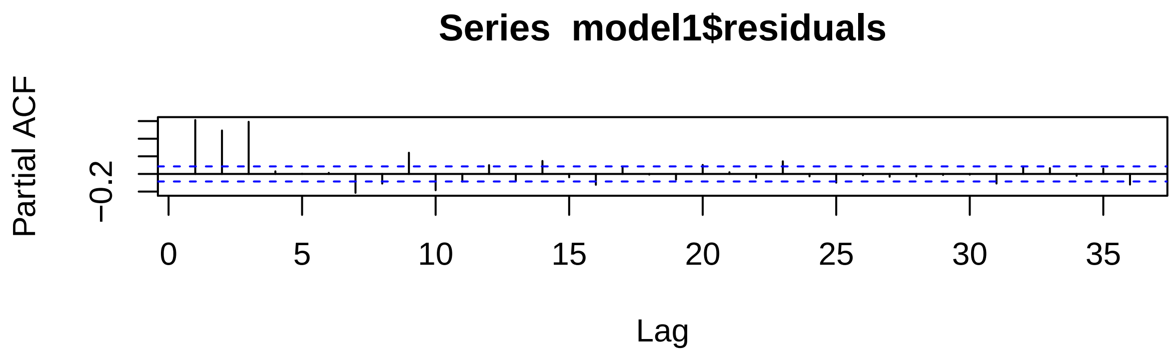
```
##   year mon dd rate w1m1 w2m1 w3m1 w4m1 icm1
## 1 1967   2  1  3.8  208  207  217  204  836
## 2 1967   3  1  3.8  216  229  229  242  916
## 3 1967   4  1  3.8  310  241  245  247 1043
## 4 1967   5  1  3.8  259  257  299  245 1315
## 5 1967   6  1  3.9  254  231  230  228  943
## 6 1967   7  1  3.8  248  238  224  218  928
```

```
unrate=data$rate
x=data[,5:9]/100
model1=lm(unrate~icm1, data=x)
summary(model1)
```

```
##
## Call:
## lm(formula = unrate ~ icm1, data = x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8638 -0.7008 -0.1299  0.7366  3.1746
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.52020    0.17847   8.518   <2e-16 ***
## icm1         0.29047    0.01097  26.475   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.051 on 522 degrees of freedom
## Multiple R-squared:  0.5731, Adjusted R-squared:  0.5723
## F-statistic: 700.9 on 1 and 522 DF,  p-value: < 2.2e-16
```

```
par(mfcol=c(2,1))
acf(model1$residuals, lag=36)
pacf(model1$residuals, lag=36)
```





ARMA can be implemented due to the significance of ACF and PACF, Moving avg can be estimate using ACF and AR can be estimated using PACF We need to find the movement of ACF and signal over differencing. And also find any significant peaks in ACF and PACF to estimte model order. Analysis on ACF and PACF suggests many models

Question2. Assume now that the residual follows a seasonal ARIMA model. For sim- plicity, assume that our model is (p, 0, q) × (1, 0, 1)12. Also assume that 2 p 4and2 q 5. Findout-thebestmodelbyperformregression with time series errors, and checking the estimated coefficients as well as AIC scores. Check if the model is adequate. A sample code is model2=arima(unrate, order=c(2,0,2),xreg=x[,5],seasonal=list(order=c(1,0,1),period=12))

```
model2=arima(unrate, order=c(2,0,2),xreg=x[,5],seasonal=list(order=c(1,0,1),
period=12))
model2
```

```
##
## Call:
## arima(x = unrate, order = c(2, 0, 2), seasonal = list(order = c(1, 0, 1), period = 12),
##     xreg = x[, 5])
##
## Coefficients:
##          ar1      ar2      ma1     ma2    sar1     sma1  intercept  x[, 5]
##       1.9123  -0.9145  -0.9100  0.1860  0.6465  -0.8483     6.1111  0.0078
## s.e.  0.0283   0.0282   0.0527  0.0479  0.0823   0.0591     0.3748  0.0021
##
## sigma^2 estimated as 0.02426:  log likelihood = 226.97,  aic = -435.93
```

```
model3=arima(unrate, order=c(2,0,3),xreg=x[,5],seasonal=list(order=c(1,0,1),
period=12))
model3
```

```
##
## Call:
## arima(x = unrate, order = c(2, 0, 3), seasonal = list(order = c(1, 0, 1), period = 12),
##     xreg = x[, 5])
##
## Coefficients:
##          ar1      ar2      ma1     ma2     ma3    sar1     sma1  intercept
##       1.8997  -0.9021  -0.8932  0.1458  0.0555  0.6501  -0.8520     6.0373
## s.e.  0.0332   0.0331   0.0543  0.0565  0.0466  0.0824   0.0586     0.3705
##       x[, 5]
##       0.0077
## s.e.  0.0021
##
## sigma^2 estimated as 0.02419:  log likelihood = 227.7,  aic = -435.39
```

```r
model4=arima(unrate, order=c(2,0,4),xreg=x[,5],seasonal=list(order=c(1,0,1),
period=12))
model4
```

```
##
## Call:
## arima(x = unrate, order = c(2, 0, 4), seasonal = list(order = c(1, 0, 1), period = 12),
##     xreg = x[, 5])
##
## Coefficients:
##          ar1      ar2      ma1     ma2     ma3     ma4    sar1     sma1
##       1.8793  -0.8819  -0.8679  0.1486  0.0162  0.0588  0.6650  -0.8629
## s.e.  0.0415   0.0414   0.0612  0.0585  0.0567  0.0497  0.0782   0.0550
##       intercept  x[, 5]
##          6.0476  0.0077
## s.e.     0.3786  0.0021
##
## sigma^2 estimated as 0.02412:  log likelihood = 228.39,  aic = -434.78
```

```r
model5=arima(unrate, order=c(2,0,5),xreg=x[,5],seasonal=list(order=c(1,0,1),
period=12))
model5
```

```
##
## Call:
## arima(x = unrate, order = c(2, 0, 5), seasonal = list(order = c(1, 0, 1), period = 12),
##     xreg = x[, 5])
##
## Coefficients:
##          ar1      ar2      ma1     ma2     ma3     ma4      ma5    sar1
##       1.8893  -0.8918  -0.8783  0.1481  0.0129  0.0724  -0.0230  0.6612
## s.e.  0.0457   0.0454   0.0646  0.0593  0.0581  0.0582   0.0533  0.0807
##          sma1  intercept  x[, 5]
##       -0.8597     6.0367  0.0077
## s.e.   0.0571     0.3715  0.0021
##
## sigma^2 estimated as 0.02411:  log likelihood = 228.48,  aic = -432.97
```

```r
model3a=arima(unrate, order=c(3,0,2),xreg=x[,5],seasonal=list(order=c(1,0,1),period=12),method="CSS")
model3a
```

```
##
```

```
## Call:
## arima(x = unrate, order = c(3, 0, 2), seasonal = list(order = c(1, 0, 1), period = 12),
##     xreg = x[, 5], method = "CSS")
##
## Coefficients:
##          ar1      ar2      ar3      ma1     ma2    sar1     sma1
##       1.9088  -0.9059  -0.0064  -0.9015  0.1822  0.4695  -0.6927
## s.e.  0.2638   0.5154   0.2530   0.2694  0.2266  0.0790   0.0714
##       intercept  x[, 5]
##          6.3938  0.0079
## s.e.     0.3631  0.0022
##
## sigma^2 estimated as 0.02502:  part log likelihood = 222.7
```

```
model3b=arima(unrate, order=c(3,0,3),xreg=x[,5],seasonal=list(order=c(1,0,1),
period=12))
model3b
```

```
##
## Call:
## arima(x = unrate, order = c(3, 0, 3), seasonal = list(order = c(1, 0, 1), period = 12),
##     xreg = x[, 5])
##
## Coefficients:
##         ar1     ar2      ar3     ma1      ma2     ma3    sar1     sma1
##       0.931  0.9604  -0.8962  0.0755  -0.7203  0.1985  0.6567  -0.8459
## s.e.  0.031  0.0284   0.0321  0.0539   0.0474  0.0487  0.0891   0.0646
##       intercept  x[, 5]
##          6.0169  0.0086
## s.e.     0.3620  0.0023
##
## sigma^2 estimated as 0.02409:  log likelihood = 228.17,  aic = -434.33
```

```
model3c=arima(unrate, order=c(3,0,4),xreg=x[,5],seasonal=list(order=c(1,0,1),
period=12))
model3c
```

```
##
## Call:
## arima(x = unrate, order = c(3, 0, 4), seasonal = list(order = c(1, 0, 1), period = 12),
##     xreg = x[, 5])
##
## Coefficients:
##          ar1     ar2      ar3     ma1      ma2     ma3     ma4    sar1
##       0.9161  0.9666  -0.8877  0.0937  -0.7396  0.2098  0.0466  0.6518
## s.e.  0.0360  0.0274   0.0351  0.0575   0.0547  0.0476  0.0472  0.0882
##          sma1  intercept  x[, 5]
##       -0.8457     6.0006  0.0085
## s.e.   0.0639     0.3724  0.0023
##
## sigma^2 estimated as 0.02404:  log likelihood = 228.63,  aic = -433.27
```

```
model3d=arima(unrate, order=c(3,0,5),xreg=x[,5],seasonal=list(order=c(1,0,1),
period=12))
model3d
```

```
##
```

```
## Call:
## arima(x = unrate, order = c(3, 0, 5), seasonal = list(order = c(1, 0, 1), period = 12),
##     xreg = x[, 5])
##
## Coefficients:
##          ar1     ar2      ar3     ma1      ma2     ma3     ma4     ma5
##       0.8963  0.9585  -0.8604  0.1196  -0.7031  0.1682  0.0669  0.0682
## s.e.  0.0438  0.0310   0.0452  0.0629   0.0625  0.0556  0.0497  0.0503
##         sar1     sma1  intercept  x[, 5]
##       0.6736  -0.8604     6.0194  0.0084
## s.e.  0.0837   0.0594     0.3813  0.0022
##
## sigma^2 estimated as 0.02396:  log likelihood = 229.54,  aic = -433.08
```

```
model4a=arima(unrate, order=c(4,0,2),xreg=x[,5],seasonal=list(order=c(1,0,1),
period=12))
```

```
## Warning in arima(unrate, order = c(4, 0, 2), xreg = x[, 5], seasonal =
## list(order = c(1, : possible convergence problem: optim gave code = 1
```

```
model4a
```

```
##
## Call:
## arima(x = unrate, order = c(4, 0, 2), seasonal = list(order = c(1, 0, 1), period = 12),
##     xreg = x[, 5])
##
## Coefficients:
##          ar1     ar2      ar3      ar4     ma1      ma2    sar1     sma1
##       0.7256  1.1625  -0.6967  -0.1969  0.2999  -0.6938  0.6613  -0.8485
## s.e.  0.0645  0.0578   0.0623   0.0487  0.0553   0.0523  0.0946   0.0675
##       intercept  x[, 5]
##          6.0379  0.0084
## s.e.     0.3452  0.0023
##
## sigma^2 estimated as 0.02419:  log likelihood = 227.17,  aic = -432.35
```

```
model4b=arima(unrate, order=c(4,0,3),xreg=x[,5],seasonal=list(order=c(1,0,1),
period=12))
```

```
## Warning in arima(unrate, order = c(4, 0, 3), xreg = x[, 5], seasonal =
## list(order = c(1, : possible convergence problem: optim gave code = 1
```

```
model4b
```

```
##
## Call:
## arima(x = unrate, order = c(4, 0, 3), seasonal = list(order = c(1, 0, 1), period = 12),
##     xreg = x[, 5])
##
## Coefficients:
##          ar1      ar2      ar3     ar4      ma1      ma2     ma3    sar1
##       1.7133  -0.2487  -0.7584  0.2915  -0.7062  -0.3080  0.3318  0.6668
## s.e.  0.4958   1.0036   0.6549  0.2001   0.4939   0.5316  0.2175  0.0793
##          sma1  intercept  x[, 5]
##       -0.8627     6.0356  0.0077
## s.e.   0.0554     0.3882  0.0021
```

```
##
## sigma^2 estimated as 0.02413:  log likelihood = 228.31,  aic = -432.62
```

```
model4c=arima(unrate, order=c(4,0,4),xreg=x[,5],seasonal=list(order=c(1,0,1),
period=12))
model4c
```

```
##
## Call:
## arima(x = unrate, order = c(4, 0, 4), seasonal = list(order = c(1, 0, 1), period = 12),
##     xreg = x[, 5])
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##          ar1      ar2      ar3     ar4     ma1     ma2     ma3     ma4
##       1.5218  -0.1982  -0.3303  0.0032  -0.511  -0.173  0.0696  0.0781
## s.e.     NaN      NaN      NaN     NaN     NaN     NaN  0.0489  0.0466
##         sar1     sma1  intercept  x[, 5]
##       0.6626  -0.8605     6.0317  0.0077
## s.e.     NaN      NaN     0.3757  0.0021
##
## sigma^2 estimated as 0.02411:  log likelihood = 228.52,  aic = -431.04
```

```
model4d=arima(unrate, order=c(4,0,5),xreg=x[,5],seasonal=list(order=c(1,0,1),
period=12))
model4d
```

```
##
## Call:
## arima(x = unrate, order = c(4, 0, 5), seasonal = list(order = c(1, 0, 1), period = 12),
##     xreg = x[, 5])
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##          ar1     ar2      ar3      ar4     ma1      ma2     ma3     ma4
##       0.8616  0.9895  -0.8248  -0.0311  0.1528  -0.6980  0.1442  0.0724
## s.e.     NaN     NaN      NaN      NaN  0.0397   0.0422  0.0490  0.0445
##          ma5    sar1     sma1  intercept  x[, 5]
##       0.0689  0.6686  -0.8583     6.2518  0.0085
## s.e.  0.0437  0.0117   0.0279     0.4549  0.0022
##
## sigma^2 estimated as 0.02398:  log likelihood = 229.37,  aic = -430.74
```

The best model after checking all the AIC values is(3,0,2) with minimum of AIC -436.1

Question3. Now we use the weekly initial jobless claims. First run a multiple regression between unrate with the four weekly numbers w1m1 w2m1 w3m1 w4m1 and the monthly cliam "icm1". Select the significant variables from this regression. (Note, this is just to judge the which of the weeks are important to include, not to obtain estimates).

```
fit1<- lm(unrate~ w1m1+w2m1+w3m1+w4m1+icm1,data=data)
summary(fit1)
```

```
##
## Call:
```

```
## lm(formula = unrate ~ w1m1 + w2m1 + w3m1 + w4m1 + icm1, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.38093 -0.65802 -0.05042  0.64144  2.51668
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.5156244  0.1651958   3.121  0.00190 **
## w1m1         0.0065148  0.0021157   3.079  0.00218 **
## w2m1         0.0094341  0.0028865   3.268  0.00115 **
## w3m1        -0.0026863  0.0028226  -0.952  0.34169
## w4m1         0.0015164  0.0020751   0.731  0.46525
## icm1         0.0001453  0.0002176   0.668  0.50464
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8846 on 518 degrees of freedom
## Multiple R-squared:  0.7001, Adjusted R-squared:  0.6972
## F-statistic: 241.8 on 5 and 518 DF,  p-value: < 2.2e-16
```

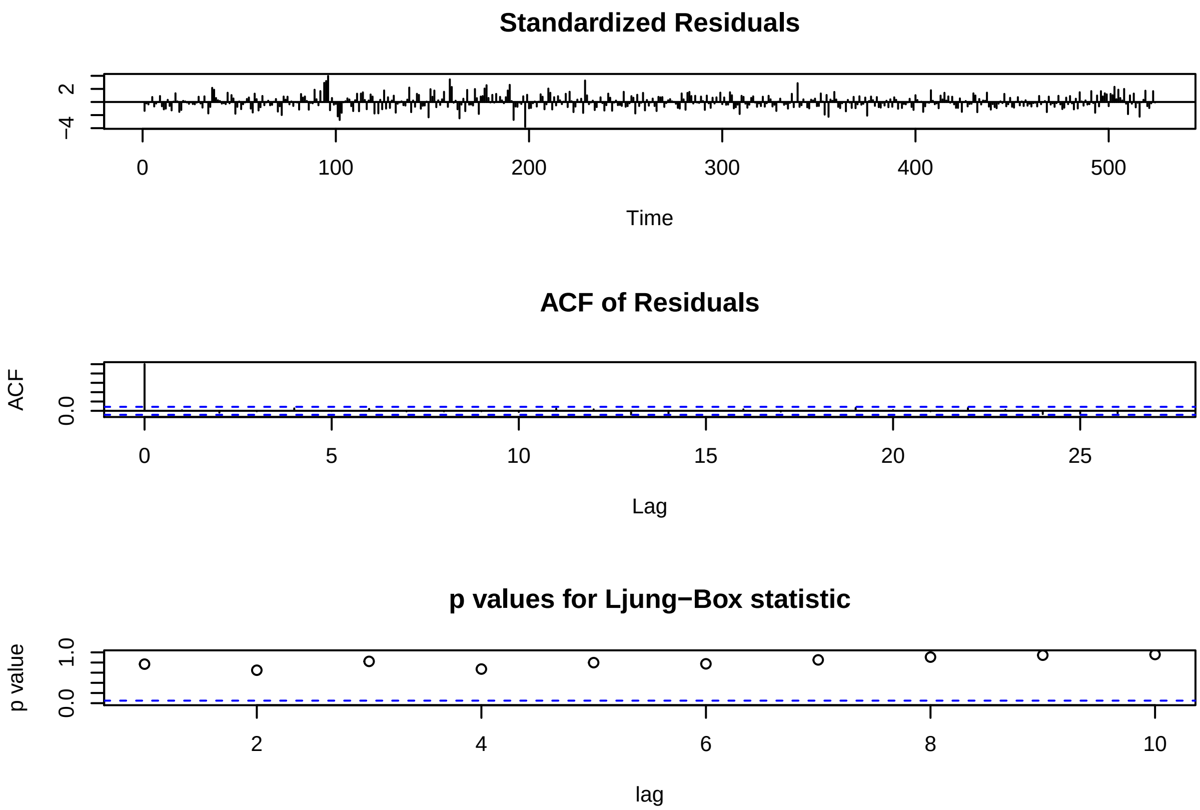
From above we can observe w1m1 and w2m1 are significant variables.

Question4. Run a time series regression of unrate with the selected variables from the last step using the arima command. For simplicity, assume that the residuals follow (2, 0, 2) ? (1, 0, 1)12. Perform model diagnosis, and check model adequacy. (You can certainly run a loop to select the proper p and q values for the ARMA. But it's not required for the homework, you can simply use 2 and 2.)

```r
model_ad=arima(unrate, order=c(2,0,2),xreg=x[,1:2],seasonal=list(order=c(1,0,1),
period=12))
model_ad
```

```
##
## Call:
## arima(x = unrate, order = c(2, 0, 2), seasonal = list(order = c(1, 0, 1), period = 12),
##     xreg = x[, 1:2])
##
## Coefficients:
##          ar1      ar2      ma1     ma2    sar1     sma1  intercept    w1m1
##       1.9172  -0.9197  -0.9958  0.2532  0.6111  -0.7915     5.6555  0.0426
## s.e.  0.0269   0.0268   0.0563  0.0507  0.1119   0.0883     0.3912  0.0272
##         w2m1
##       0.0969
## s.e.  0.0321
##
## sigma^2 estimated as 0.024:  log likelihood = 230.29,  aic = -440.59
```

```r
tsdiag(model_ad)
```



Question5. Compare the AIC scores of the model obtained in step 2 and 4. AIC value=-440.59 sothe model is better

Use quantmod to obtain adjusted daily closing pricing of "SPY". Use dates from the first trading day of 2012 todate.

```r
library(quantmod)
```

```
## Loading required package: xts

## Warning: package 'xts' was built under R version 3.4.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.4.3

##
## Attaching package: 'zoo'

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

## Loading required package: TTR

## Warning: package 'TTR' was built under R version 3.4.3

## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```r
start <- as.POSIXct("2012-01-01")
```

```
## Warning in strptime(xx, f <- "%Y-%m-%d %H:%M:%OS", tz = tz): unknown
## timezone 'zone/tz/2018c.1.0/zoneinfo/America/New_York'
```

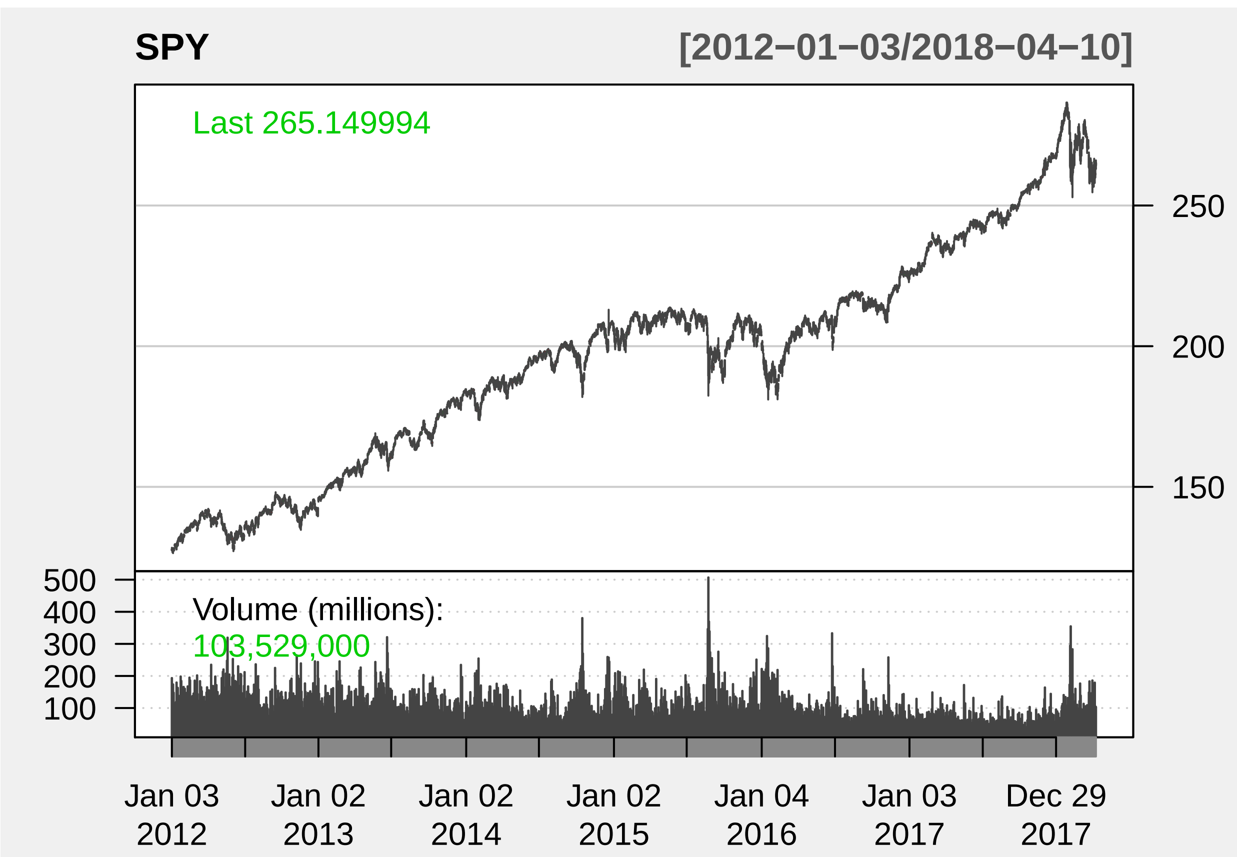
```
end <- as.POSIXct("2018-04-11")
getSymbols(Symbols = "SPY",src = "yahoo", from = start, to = end)
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
##
## WARNING: There have been significant changes to Yahoo Finance data.
## Please see the Warning section of '?getSymbols.yahoo' for details.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.yahoo.warning"=FALSE).
```

```
## [1] "SPY"
```

```
candleChart(SPY,close.col="black",dn.col="red",theme = "white")
```




```
spyclose<-as.xts(data.frame(SPYClose = SPY$"SPY.Close"))
```

Compute the daily log return.

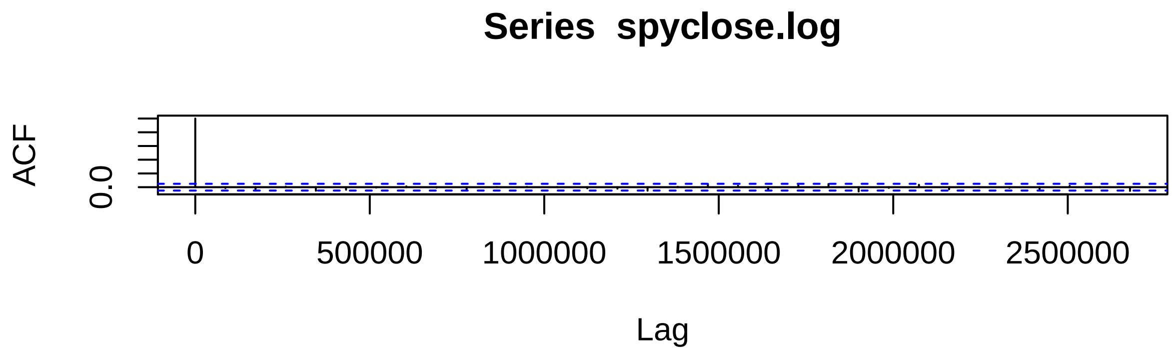
```r
spyclose.log<- log(dailyReturn(spyclose)+1)
class(spyclose.log)
```

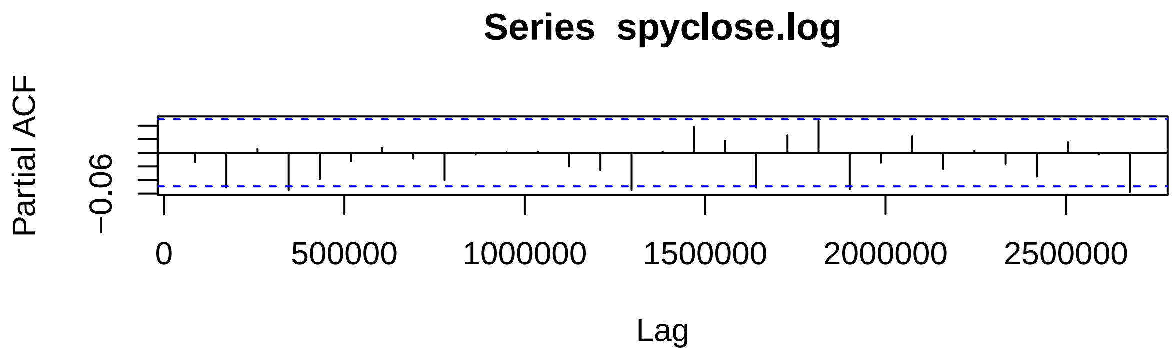
```
## [1] "xts" "zoo"
```

```r
# Time series conversion

# Mean equation
spyclose.log.m1 <- arima(spyclose.log, order = c(0, 0, 2), include.mean = F)
```

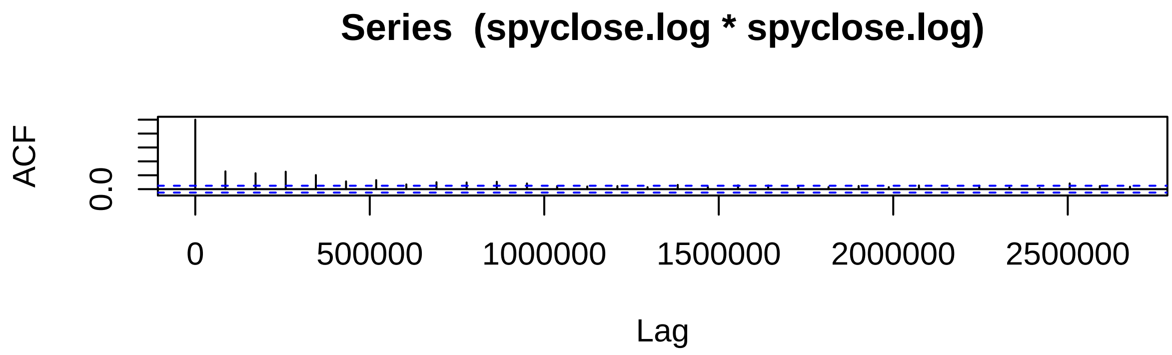
```r
#acf&pacf
par(mfcol = c(2, 1))
acf(spyclose.log)
pacf(spyclose.log)
```

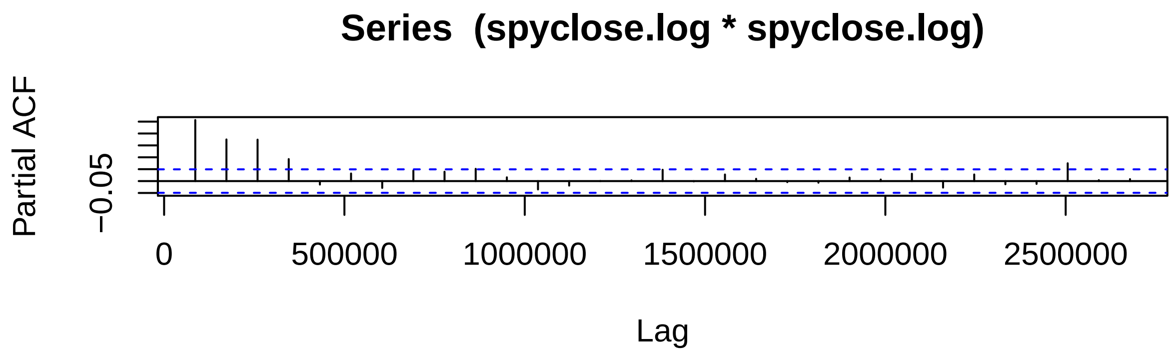





```r
par(mfcol = c(1, 1))
```

```r
par(mfcol = c(2, 1))
acf((spyclose.log*spyclose.log))
pacf((spyclose.log*spyclose.log))
```






```r
par(mfcol = c(1, 1))
```

```r
# Ljung-Box test
Box.test(spyclose.log, lag = 10, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  spyclose.log
## X-squared = 13.501, df = 10, p-value = 0.197
```

If the p value is greater than 0.05 then the residuals are independent which we want for the model to be correct. The model is correct.

(2) Select the best ARMA model to get rid of serial correlations. For simplicity, consider only the AR order, which you can use the ar() command to get the proper order.

```r
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.4.4
```

```r
fit2<- ar(spyclose, seasonal=TRUE)
fit2
```

```
##
## Call:
## ar(x = spyclose, seasonal = TRUE)
##
## Coefficients:
##     1
## 0.997
##
```

```
## Order selected 1  sigma^2 estimated as  8.466
```

```r
model2=arima(spyclose.log, order=c(1,1,1))
```

(3) Check if there exists any ARCH effect in the daily log returns of "SPY"? Why?

```r
Box.test(residuals(spyclose.log.m1)^2, lag = 10, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  residuals(spyclose.log.m1)^2
## X-squared = 444.15, df = 10, p-value < 2.2e-16
```

p-value<0.05 and is significant Reject null hypothesis and there is ARCH effect.

(4) Fit a AR+GARCH(1,1) model using fGarch package for the log return of "SPY" using Gaussian distribution for the innovations. Perform model checking, and write down the fitted model. Check the significance of the AR coeffi- cients

```r
# Gaussian ARMA-GARCH model to the log return series
library(fGarch)
```

```
## Loading required package: timeDate

## Loading required package: timeSeries

##
## Attaching package: 'timeSeries'

## The following object is masked from 'package:zoo':
##
##     time<-

## Loading required package: fBasics

##
## Attaching package: 'fBasics'

## The following object is masked from 'package:TTR':
##
##     volatility
```

```r
ms.log.m2 <- garchFit(~arma(0, 2) + garch(1, 1), data = spyclose.log, trace = F,
                      include.mean = F)
ms.log.m2
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(0, 2) + garch(1, 1), data = spyclose.log,
##     include.mean = F, trace = F)
##
## Mean and Variance Equation:
##  data ~ arma(0, 2) + garch(1, 1)
## <environment: 0x7fae64144c30>
##  [data = spyclose.log]
##
## Conditional Distribution:
```

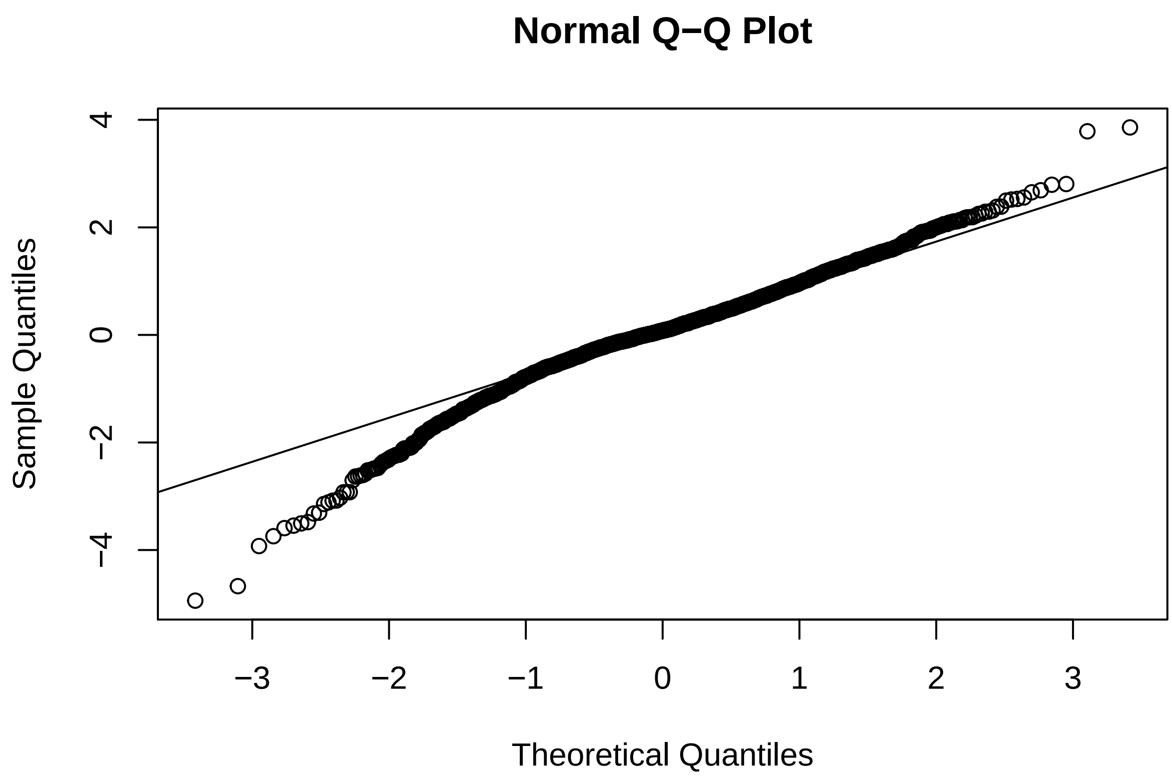
```
##  norm
##
## Coefficient(s):
##         ma1          ma2        omega       alpha1        beta1
## -3.1063e-02   7.6707e-03   4.6642e-06   1.7324e-01   7.5678e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## ma1    -3.106e-02   2.821e-02   -1.101    0.271
## ma2     7.671e-03   2.818e-02    0.272    0.785
## omega   4.664e-06   9.092e-07    5.130 2.90e-07 ***
## alpha1  1.732e-01   2.427e-02    7.137 9.55e-13 ***
## beta1   7.568e-01   3.000e-02   25.225  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  5540.757    normalized:  3.51348
##
## Description:
##  Fri Apr 13 16:17:28 2018 by user:
```

```
summary(ms.log.m2)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(0, 2) + garch(1, 1), data = spyclose.log,
##     include.mean = F, trace = F)
##
## Mean and Variance Equation:
##  data ~ arma(0, 2) + garch(1, 1)
## <environment: 0x7fae64144c30>
##  [data = spyclose.log]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         ma1          ma2        omega       alpha1        beta1
## -3.1063e-02   7.6707e-03   4.6642e-06   1.7324e-01   7.5678e-01
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## ma1    -3.106e-02   2.821e-02   -1.101    0.271
## ma2     7.671e-03   2.818e-02    0.272    0.785
## omega   4.664e-06   9.092e-07    5.130 2.90e-07 ***
```

```
## alpha1  1.732e-01   2.427e-02    7.137 9.55e-13 ***
## beta1   7.568e-01   3.000e-02   25.225  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  5540.757    normalized:  3.51348
##
## Description:
##  Fri Apr 13 16:17:28 2018 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  265.0454  0
##  Shapiro-Wilk Test  R    W      0.9760855 1.530006e-15
##  Ljung-Box Test     R    Q(10)  7.782648  0.650058
##  Ljung-Box Test     R    Q(15)  13.86843  0.5355305
##  Ljung-Box Test     R    Q(20)  21.95891  0.3427446
##  Ljung-Box Test     R^2  Q(10)  8.55502   0.5747884
##  Ljung-Box Test     R^2  Q(15)  11.97737  0.6807411
##  Ljung-Box Test     R^2  Q(20)  13.2816   0.8649774
##  LM Arch Test       R    TR^2   9.791501  0.6342463
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -7.020618 -7.003613 -7.020638 -7.014299
```

```r
# examine residuals for normality assumption
ms.log.m2.res <- residuals(ms.log.m2, standardize = T)
# Q-Q Plot
qqnorm(ms.log.m2.res); qqline(ms.log.m2.res)
```




```r
# shapiro test
shapiro.test(ms.log.m2.res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  ms.log.m2.res
## W = 0.97609, p-value = 1.53e-15
```

AR>0.05 so it is insignificant

(5) Fit the AR+GARCH(1,1) model again using the Student-t distribution for the innovations. Write down the fitted model. Check the significance of the AR coefficients.

```r
ms.log.m4 <- garchFit(~arma(0, 1) + garch(1, 1), data = spyclose.log, trace = F,
                      cond.dist = "std", include.mean = F)
summary(ms.log.m4)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(0, 1) + garch(1, 1), data = spyclose.log,
##     cond.dist = "std", include.mean = F, trace = F)
##
## Mean and Variance Equation:
##  data ~ arma(0, 1) + garch(1, 1)
## <environment: 0x7fae5f0b10a0>
##  [data = spyclose.log]
##
```

```
## Conditional Distribution:
##  std
##
## Coefficient(s):
##         ma1        omega       alpha1        beta1        shape
## -4.1810e-02   3.1525e-06   1.7821e-01   7.9052e-01   5.2497e+00
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## ma1    -4.181e-02   2.609e-02   -1.603  0.10904
## omega   3.152e-06   9.413e-07    3.349  0.00081 ***
## alpha1  1.782e-01   3.162e-02    5.636 1.74e-08 ***
## beta1   7.905e-01   3.350e-02   23.598  < 2e-16 ***
## shape   5.250e+00   7.549e-01    6.954 3.55e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  5582.164    normalized:  3.539736
##
## Description:
##  Fri Apr 13 16:17:29 2018 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  332.462   0
##  Shapiro-Wilk Test  R    W      0.9738797 0
##  Ljung-Box Test     R    Q(10)  7.248686  0.7017774
##  Ljung-Box Test     R    Q(15)  13.24535  0.5833536
##  Ljung-Box Test     R    Q(20)  20.90341  0.4028372
##  Ljung-Box Test     R^2  Q(10)  7.697611  0.6583467
##  Ljung-Box Test     R^2  Q(15)  13.62099  0.5544467
##  Ljung-Box Test     R^2  Q(20)  16.14213  0.7077685
##  LM Arch Test       R    TR^2   9.952003  0.6201716
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -7.073131 -7.056126 -7.073151 -7.066812
```

AR>0.05 so it is insignificant

(6) Between these two models, choose the one that is the best based on AIC scores reported in the output. AIC score of student T-distribution is less compared to Guassian Distribution, So T-distribution is the best model