Homework 7

Amin Baabol

11/19/2020

Problem 5.12:

```r
#Packages
library(tseries)

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

library(forecast)
library(zoo)

##
## Attaching package: 'zoo'

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

Problem 5.12:

##Instructions:

Part 5.2a

a) "Fit an ARIMA model" – specify the model and give the estimators of the coefficients. For example, if you use a AR(2) model to fit the data, then you need to give the model like y_t = 30 + 0.5 y_{t-1} + 0.3 y_{t-2} + _t. Your answer should include the ACF and PACF of the data, then determine your "best" model. You can use the auto.arima() function to double check.

"Investigate the model adequacy" – show the 4-in-1 residual plots, and ACF and PACF plots for the residuals. Then interpret your outputs/results.

"Explain how this model would be used for forecasting" – show the forecast model. To specify the forecast model, you need to show how to get estimators of the Ψ_i's. Show at least the first 4 Ψ_i's. Similar to what we did for Example 5.3, 5.4, and 5.5 on page 381-382 in the class. The general forecast model is shown as the equation of (5.88) on page 379.

###Answer:

From the standard time series plot we can see an irregular, cyclical pulses with no apparently defined pattern.The non-stationarity of the original time series data is implicated by its ACF plot which seems to be undergoing an exponential decay while the

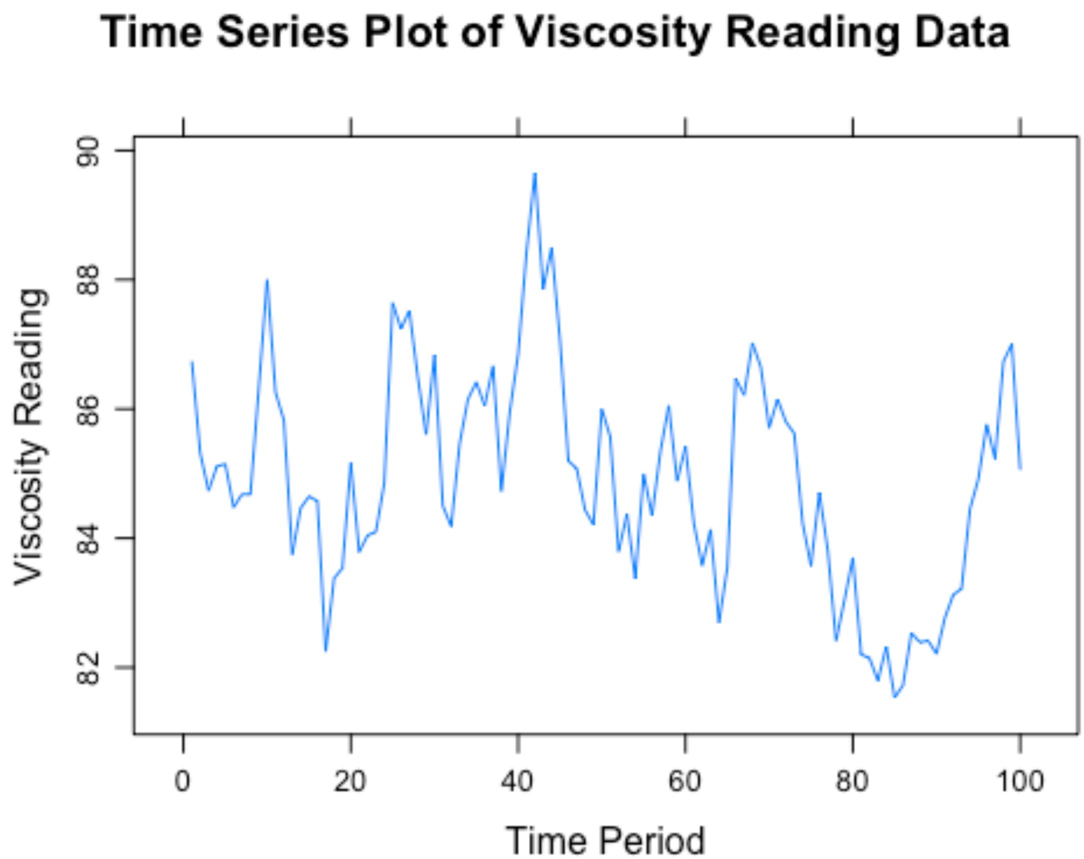
PACF plot quickly drops to zero after the first lag.The Augmented Dickey-Fuller Test supports this non-stationarity indication hinted by the ACF and the PACF plots with its large p-value of 0.2603.Hence,we fail to reject the null hypothesis, that is the time series is non-stationary.

After the first differencing, the stationarity of the difference time series is validated by the ACF and PACF plots which indicate perhaps 1 or less significant lags.I propose using ARIMA(1,0,0) of the difference times series as well as ARIMA(0,1,0) of the original time time series.

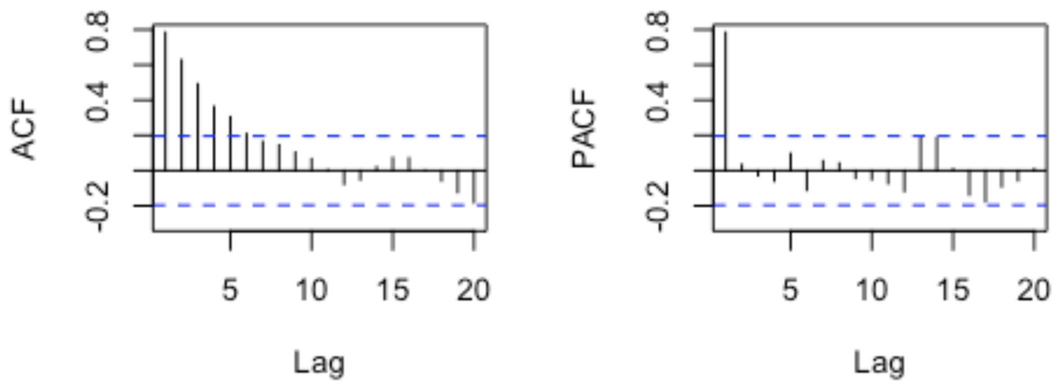
```
Viscosity.Reading <- read.csv("~/Desktop/GradSchool/STATS 560 Time Series Analysis/Lecture/ViscosityData.csv",
                              header = TRUE)
Viscosity.Reading

##     Reading
## 1   86.7418
## 2   85.3195
## 3   84.7355
## 4   85.1113
## 5   85.1487
## 6   84.4775
## 7   84.6827
## 8   84.6757
## 9   86.3169
## 10  88.0006
## 11  86.2597
## 12  85.8286
## 13  83.7500
## 14  84.4628
## 15  84.6476
## 16  84.5751
## 17  82.2473
## 18  83.3774
## 19  83.5385
## 20  85.1620
## 21  83.7881
## 22  84.0421
## 23  84.1023
## 24  84.8495
## 25  87.6416
## 26  87.2397
## 27  87.5219
## 28  86.4992
## 29  85.6050
## 30  86.8293
## 31  84.5004
## 32  84.1844
## 33  85.4563
## 34  86.1511
```

```
## 35  86.4142
## 36  86.0498
## 37  86.6642
## 38  84.7289
## 39  85.9523
## 40  86.8473
## 41  88.4250
## 42  89.6481
## 43  87.8566
## 44  88.4997
## 45  87.0622
## 46  85.1973
## 47  85.0767
## 48  84.4362
## 49  84.2112
## 50  85.9952
## 51  85.5722
## 52  83.7935
## 53  84.3706
## 54  83.3762
## 55  84.9975
## 56  84.3495
## 57  85.3395
## 58  86.0503
## 59  84.8839
## 60  85.4176
## 61  84.2309
## 62  83.5761
## 63  84.1343
## 64  82.6974
## 65  83.5454
## 66  86.4714
## 67  86.2143
## 68  87.0215
## 69  86.6504
## 70  85.7082
## 71  86.1504
## 72  85.8032
## 73  85.6197
## 74  84.2339
## 75  83.5737
## 76  84.7052
## 77  83.8168
## 78  82.4171
## 79  83.0420
## 80  83.6993
## 81  82.2033
## 82  82.1413
## 83  81.7961
## 84  82.3241
```

```
## 85  81.5316
## 86  81.7280
## 87  82.5375
## 88  82.3877
## 89  82.4159
## 90  82.2102
## 91  82.7673
## 92  83.1234
## 93  83.2203
## 94  84.4510
## 95  84.9145
## 96  85.7609
## 97  85.2302
## 98  86.7312
## 99  87.0048
## 100 85.0572

#Convert to time series
ViscosityReading.TS <- ts(Viscosity.Reading)


#time series plot
library(lattice)
xyplot.ts(ViscosityReading.TS,
          main = 'Time Series Plot of Viscosity Reading Data',
          xlab = 'Time Period',
          ylab = 'Viscosity Reading')
```



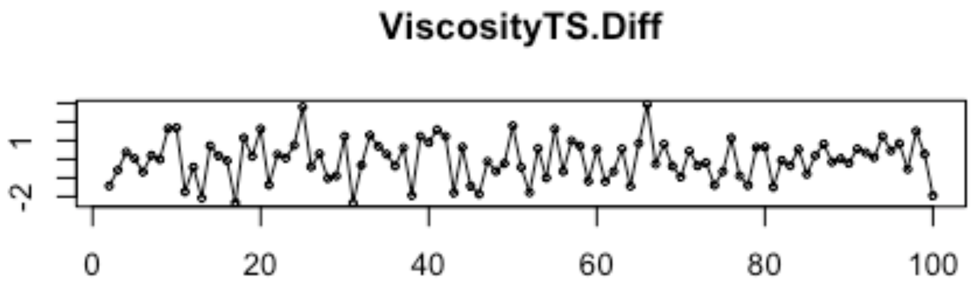

```r
#Stationarity check of the original time series
#ACF and PACF plots
tsdisplay(ViscosityReading.TS)
```

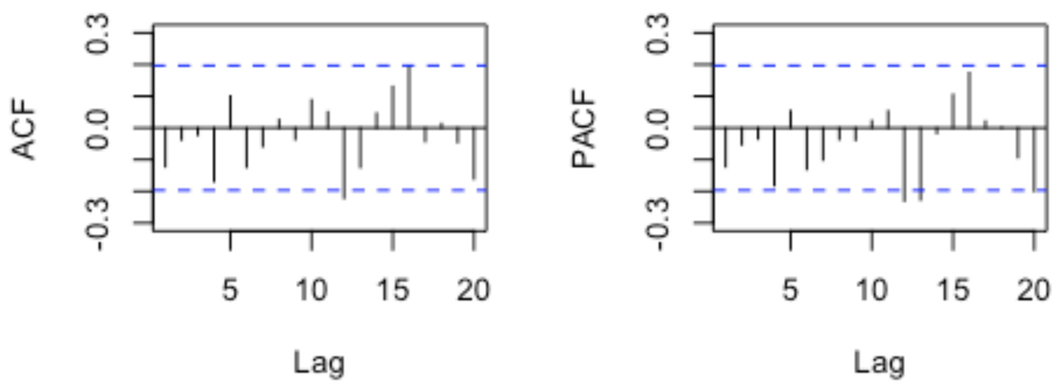



```
#Stationarity:Dicke-Fuller test
adf.test(ViscosityReading.TS)

##
##  Augmented Dickey-Fuller Test
##
## data:  ViscosityReading.TS
## Dickey-Fuller = -2.7637, Lag order = 4, p-value = 0.2603
## alternative hypothesis: stationary
```

The ACF and PACF plots of the differenced time series shows no significant autocorrelation spiking. Moreover,The Augmented Dickey-Fuller Test shows a statistically significant p-value of less then 0.01. Hence, we reject the null hypothesis that the time series is non-stationary in favor of the alternative hypothesis that the time series data is now stationary.

```
#First order differencing
ViscosityTS.Diff <- diff(ViscosityReading.TS, differences = 1)

#Stationarity check of the differenced time series
#ACF and PACF plots
tsdisplay(ViscosityTS.Diff)
```






```
#Stationarity:Dicke-Fuller test
adf.test(ViscosityTS.Diff)

## Warning in adf.test(ViscosityTS.Diff): p-value smaller than printed p-
value

##
##  Augmented Dickey-Fuller Test
##
## data:  ViscosityTS.Diff
## Dickey-Fuller = -4.8866, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

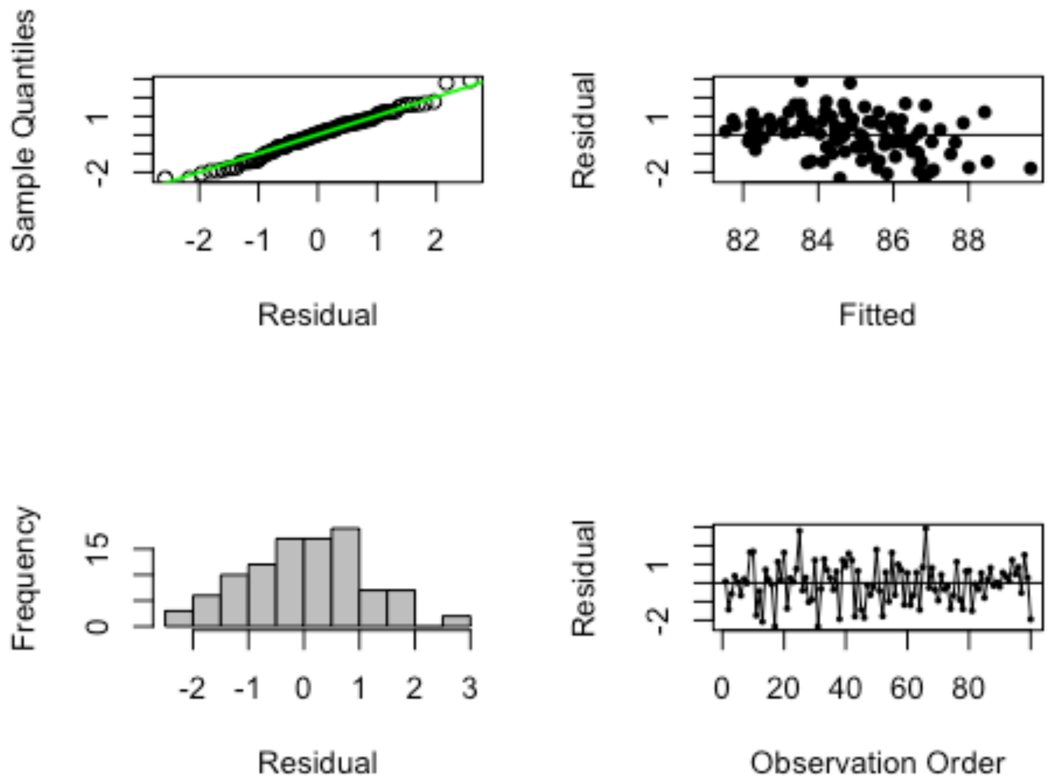
The first proposed model, ARIMA(0,1,0) has relatively low AIC score of 300.27. Checking its residuals we see that the QQ-plot shows the normal distribution of the residuals isn't violated. The fitted vs the residuals shows a fairly good degree of randomness centered around zero with a +/-2 margin, while the histogram indicates relative normal distribution of the residuals with some skewness.Lastly, the residual vs. observation order plot show that residuals exhibit normal random noise around 0 which suggests t hat there is no serial autocorrelation.

After checking the residual ACF and PACF of the ARIMA(0,1,0) model, we can see that both plots indicate there's no significant autocorrelation remaining in the residuals.

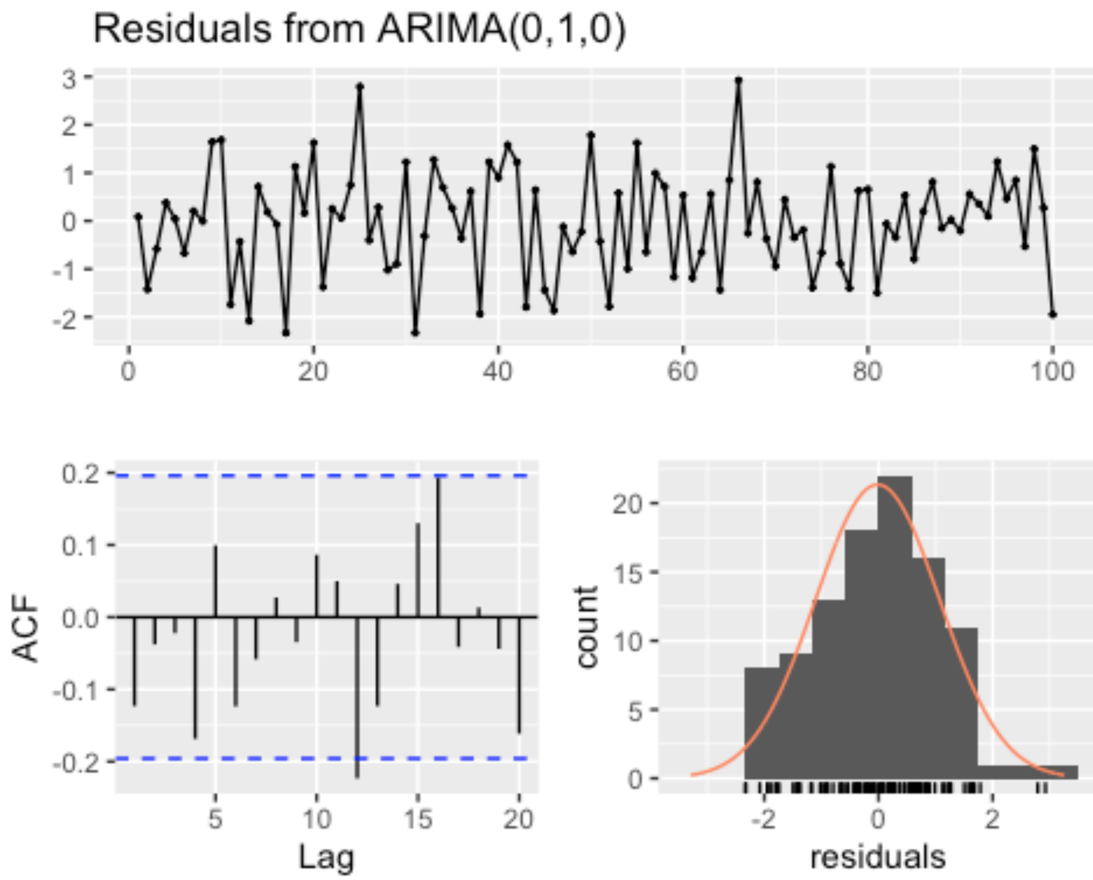
```r
arima010 <- Arima(ViscosityReading.TS, order = c(0,1,0))
arima010
```

```
## Series: ViscosityReading.TS
## ARIMA(0,1,0)
##
## sigma^2 estimated as 1.191:  log likelihood=-149.12
## AIC=300.23   AICc=300.27   BIC=302.83
```

```r
fitted<-as.vector(fitted(arima010))


#Model diagnosticso
#4-in-1 plot of the residuals
par(mfrow=c(2,2),oma=c(0,0,0,0))
qqnorm(arima010$residuals,
       pch=1,
       xlab='Residual',
       main='')
qqline(arima010$residuals,
       col="green",
       lwd=1.5)
plot(fitted,
     arima010$residuals,
     pch=16,
     xlab='Fitted',
ylab='Residual')
abline(h=0)
hist(arima010$residuals,
     col="gray",
     xlab='Residual',
     main='')
plot(arima010$residuals,
     type="l",
     xlab='Observation Order',
ylab='Residual')
points(arima010$residuals,pch=16,cex=.5)
abline(h=0)
```




```
#Residual acf and pacf
checkresiduals(arima010)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,1,0)
## Q* = 8.9696, df = 10, p-value = 0.535
##
## Model df: 0.   Total lags used: 10
```

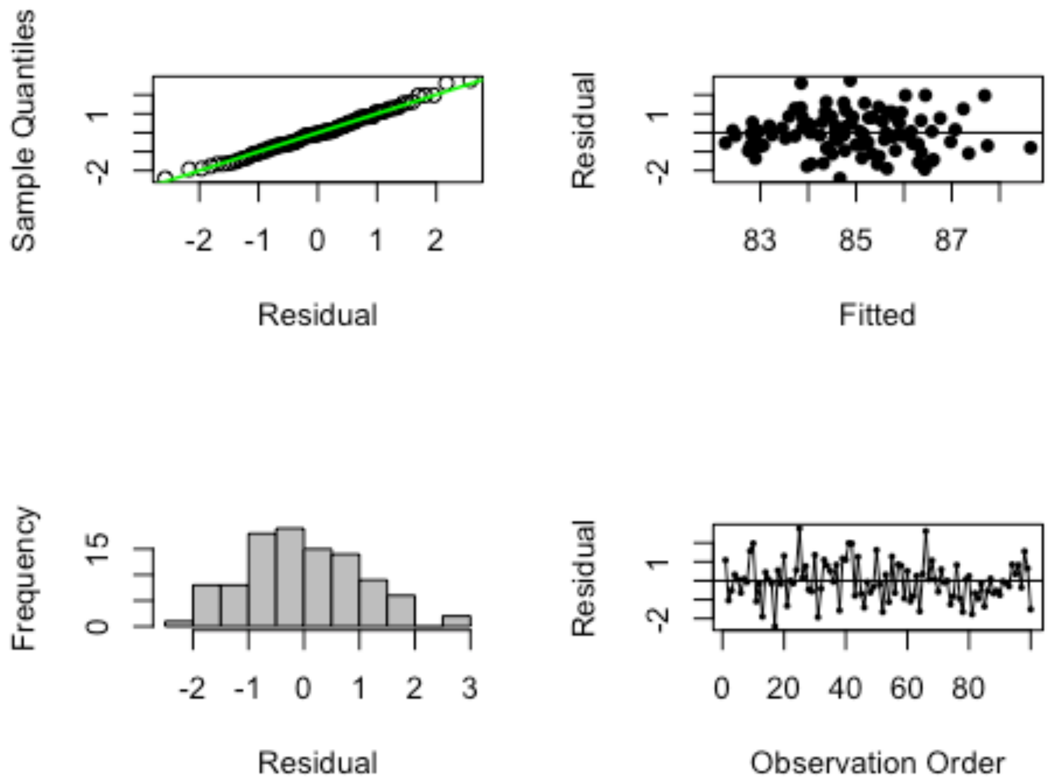
The ARIMA(1,0,0) model seems to be doing better than the previous mode.The residual ACF and PACF plots show no significant autocorrelation remaining.The AIC score at 296 is lower than the previous model which had an AIC of 300 and the auto.arima() model seems to be close to ARIMA(1,0,0) model's AIC. Furthermore,The qq-plot,histogram, fitted vs. residuals all seems to be suggesting relative normal distribution with good random dispersion of the residuals.

```
arima100 <- Arima(ViscosityReading.TS, order = c(1,0,0))
arima100
```

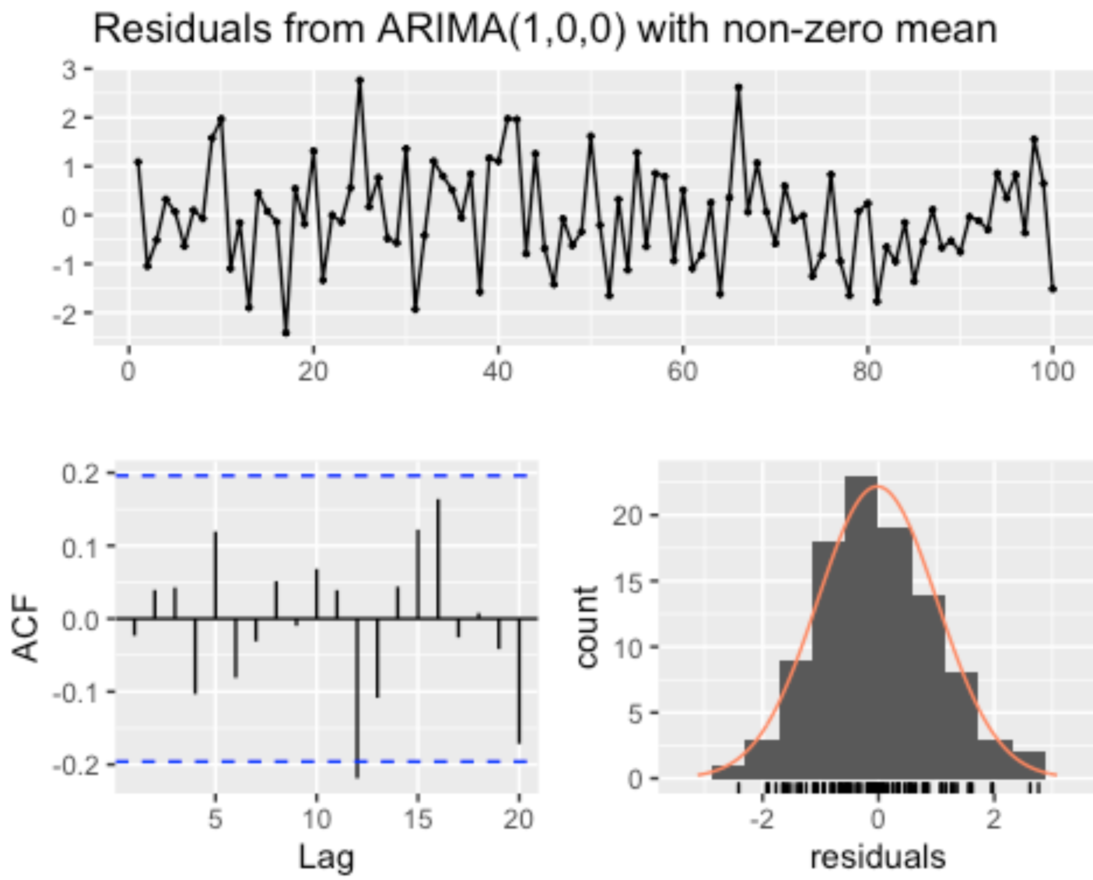
```
## Series: ViscosityReading.TS
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1     mean
```

```
##        0.7859  84.9827
## s.e.  0.0605   0.4649
##
## sigma^2 estimated as 1.082:  log likelihood=-145.32
## AIC=296.64   AICc=296.89   BIC=304.45
```

```
fitted<-as.vector(fitted(arima100))



#Model diagnostics

#4-in-1 plot of the residuals
par(mfrow=c(2,2),oma=c(0,0,0,0))
qqnorm(arima100$residuals,
       pch=1,
       xlab='Residual',
       main='')
qqline(arima100$residuals,
       col="green",
       lwd=1.5)
plot(fitted,
     arima100$residuals,
     pch=16,
     xlab='Fitted',
ylab='Residual')
abline(h=0)
hist(arima100$residuals,
     col="gray",
     xlab='Residual',
     main='')
plot(arima100$residuals,
     type="l",
     xlab='Observation Order',
ylab='Residual')
points(arima100$residuals,pch=16,cex=.5)
abline(h=0)
```




```
#Residual acf and pacf
checkresiduals(arima100)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with non-zero mean
## Q* = 4.7363, df = 8, p-value = 0.7854
##
## Model df: 2.   Total lags used: 10
```

```
#check against auto.arima()
auto.arima(ViscosityReading.TS,
           trace = TRUE,
           stepwise = FALSE,
           approximation = FALSE)
```

```
##
##  ARIMA(0,1,0)                    : 300.2736
##  ARIMA(0,1,0) with drift         : 302.3333
##  ARIMA(0,1,1)                    : 300.6087
##  ARIMA(0,1,1) with drift         : 302.7194
##  ARIMA(0,1,2)                    : 302.2848
##  ARIMA(0,1,2) with drift         : 304.4448
##  ARIMA(0,1,3)                    : 303.8227
##  ARIMA(0,1,3) with drift         : 306.0288
```

```
##  ARIMA(0,1,4)                    : 301.9459
##  ARIMA(0,1,4) with drift         : 304.1183
##  ARIMA(0,1,5)                    : 304.2136
##  ARIMA(0,1,5) with drift         : Inf
##  ARIMA(1,1,0)                    : 300.8023
##  ARIMA(1,1,0) with drift         : 302.9112
##  ARIMA(1,1,1)                    : 296.4466
##  ARIMA(1,1,1) with drift         : Inf
##  ARIMA(1,1,2)                    : 298.5492
##  ARIMA(1,1,2) with drift         : Inf
##  ARIMA(1,1,3)                    : 300.5931
##  ARIMA(1,1,3) with drift         : Inf
##  ARIMA(1,1,4)                    : 302.6575
##  ARIMA(1,1,4) with drift         : 306.2916
##  ARIMA(2,1,0)                    : 302.6249
##  ARIMA(2,1,0) with drift         : 304.7821
##  ARIMA(2,1,1)                    : 298.5426
##  ARIMA(2,1,1) with drift         : Inf
##  ARIMA(2,1,2)                    : 299.9143
##  ARIMA(2,1,2) with drift         : Inf
##  ARIMA(2,1,3)                    : 302.8241
##  ARIMA(2,1,3) with drift         : Inf
##  ARIMA(3,1,0)                    : 304.6947
##  ARIMA(3,1,0) with drift         : 306.8989
##  ARIMA(3,1,1)                    : 300.6552
##  ARIMA(3,1,1) with drift         : Inf
##  ARIMA(3,1,2)                    : Inf
##  ARIMA(3,1,2) with drift         : Inf
##  ARIMA(4,1,0)                    : 303.4839
##  ARIMA(4,1,0) with drift         : 305.7388
##  ARIMA(4,1,1)                    : 305.0721
##  ARIMA(4,1,1) with drift         : 307.3754
##  ARIMA(5,1,0)                    : 305.4589
##  ARIMA(5,1,0) with drift         : 307.7625
## 
## 
## 
##  Best model: ARIMA(1,1,1)
```

```
## Series: ViscosityReading.TS 
## ARIMA(1,1,1) 
## 
## Coefficients:
##          ar1      ma1
##       0.7930  -0.9820
## s.e.  0.0788   0.0465
## 
## sigma^2 estimated as 1.106:  log likelihood=-145.1
## AIC=296.19   AICc=296.45   BIC=303.98
```

Therefore, I selected to move forward with ARIMA(1,0,0).

Model equation:

$$y_t = 84.9827 + 0.7859 * y_{t-1} + \epsilon_t$$

Forecast model

$$Y_t - \mu = \phi(Y_{t-1} - \mu) + \epsilon_t$$

3.

$$E(\epsilon_t + 1 | Y_1, Y_2, \ldots, Y_t) = 0$$

Forecasting

$$Y_{t+1}$ $$

(1) = + (Y_t-)$$

Subsequently, we get

$$\hat{Y}(l) = \mu + \phi^l(Y_t - \mu)$$

Step-ahead forecast

1-step ahead

$$\hat{Y}(1) = \mu + \phi^1(Y_t - \mu)$$

2-step ahead

$$\hat{Y}(2) = \mu + \phi^2(Y_t - \mu)$$

3-step ahead

$$\hat{Y}(3) = \mu + \phi^3(Y_t - \mu)$$

4-step ahead

$$\hat{Y}(4) = \mu + \phi^4(Y_t - \mu)$$

$$Y_{t+1}$

Part 5.2b

##Instructions: "Forecast the last 20 observations." – No matter what model that you got in part a, to answer question b, please use auto.arima() to get the best model. (This is just for the grading purpose.) Then use the forecast() to get the forecast values. See the R code on page 408. This will give you a 1- to 20- step ahead forecasts.

```
optimal.arima <- auto.arima(ViscosityReading.TS,
                            trace = TRUE,
```

```
                                stepwise = FALSE,
                                approximation = FALSE)

##
##  ARIMA(0,1,0)                    : 300.2736
##  ARIMA(0,1,0) with drift         : 302.3333
##  ARIMA(0,1,1)                    : 300.6087
##  ARIMA(0,1,1) with drift         : 302.7194
##  ARIMA(0,1,2)                    : 302.2848
##  ARIMA(0,1,2) with drift         : 304.4448
##  ARIMA(0,1,3)                    : 303.8227
##  ARIMA(0,1,3) with drift         : 306.0288
##  ARIMA(0,1,4)                    : 301.9459
##  ARIMA(0,1,4) with drift         : 304.1183
##  ARIMA(0,1,5)                    : 304.2136
##  ARIMA(0,1,5) with drift         : Inf
##  ARIMA(1,1,0)                    : 300.8023
##  ARIMA(1,1,0) with drift         : 302.9112
##  ARIMA(1,1,1)                    : 296.4466
##  ARIMA(1,1,1) with drift         : Inf
##  ARIMA(1,1,2)                    : 298.5492
##  ARIMA(1,1,2) with drift         : Inf
##  ARIMA(1,1,3)                    : 300.5931
##  ARIMA(1,1,3) with drift         : Inf
##  ARIMA(1,1,4)                    : 302.6575
##  ARIMA(1,1,4) with drift         : 306.2916
##  ARIMA(2,1,0)                    : 302.6249
##  ARIMA(2,1,0) with drift         : 304.7821
##  ARIMA(2,1,1)                    : 298.5426
##  ARIMA(2,1,1) with drift         : Inf
##  ARIMA(2,1,2)                    : 299.9143
##  ARIMA(2,1,2) with drift         : Inf
##  ARIMA(2,1,3)                    : 302.8241
##  ARIMA(2,1,3) with drift         : Inf
##  ARIMA(3,1,0)                    : 304.6947
##  ARIMA(3,1,0) with drift         : 306.8989
##  ARIMA(3,1,1)                    : 300.6552
##  ARIMA(3,1,1) with drift         : Inf
##  ARIMA(3,1,2)                    : Inf
##  ARIMA(3,1,2) with drift         : Inf
##  ARIMA(4,1,0)                    : 303.4839
##  ARIMA(4,1,0) with drift         : 305.7388
##  ARIMA(4,1,1)                    : 305.0721
##  ARIMA(4,1,1) with drift         : 307.3754
##  ARIMA(5,1,0)                    : 305.4589
##  ARIMA(5,1,0) with drift         : 307.7625
##
##
##
##  Best model: ARIMA(1,1,1)
```

```r
forecast <-as.array(forecast(optimal.arima,h = 20))
paste("Last 20 observations")
```

```
## [1] "Last 20 observations"
```

```r
forecast
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 101       84.99210 83.64377 86.34043 82.93001 87.05419
## 102       84.94047 83.20384 86.67710 82.28452 87.59641
## 103       84.89953 82.94669 86.85236 81.91292 87.88613
## 104       84.86706 82.78105 86.95308 81.67678 88.05735
## 105       84.84132 82.66887 87.01377 81.51885 88.16379
## 106       84.82090 82.59028 87.05153 81.40945 88.23235
## 107       84.80471 82.53371 87.07571 81.33152 88.27791
## 108       84.79187 82.49206 87.09169 81.27461 88.30914
## 109       84.78169 82.46073 87.10266 81.23208 88.33130
## 110       84.77362 82.43670 87.11054 81.19961 88.34763
## 111       84.76722 82.41792 87.11651 81.17428 88.36016
## 112       84.76214 82.40298 87.12131 81.15411 88.37017
## 113       84.75811 82.39087 87.12536 81.13772 88.37851
## 114       84.75492 82.38088 87.12896 81.12414 88.38570
## 115       84.75239 82.37250 87.13228 81.11267 88.39211
## 116       84.75038 82.36535 87.13542 81.10279 88.39798
## 117       84.74879 82.35913 87.13845 81.09412 88.40346
## 118       84.74753 82.35364 87.14141 81.08640 88.40866
## 119       84.74653 82.34872 87.14433 81.07940 88.41365
## 120       84.74573 82.34424 87.14723 81.07297 88.41850
```

Part 5.2c

##Instructions:

"Show how to obtain prediction intervals…" — The 80% and 95% Prediction Intervals should have been obtained from part b if you use forecast() function. You don't have to calculate the Prediction intervals again. For this question, just show the prediction interval formula.

Formula for 95% prediction interval

$$\hat{y}_{T+\tau}(T) \pm (1.96) * \sigma(\tau)$$

Formula for 80% prediction interval

$$\hat{y}_{T+\tau}(T) \pm (1.282) * \sigma(\tau)$$

Part 5.33

##Instructions:

Develop an ARIMA model and a procedure for forecasting for these data."— Show the model and compute the 1- to 10-step ahead forecasts. Similar to 5.12, but you can skip the calculation of Ψ_i's.

“Explain how prediction intervals would be computed.” — show the prediction intervals formula. Compute the 95% prediction intervals for the 1- to 10-step ahead forecasts.

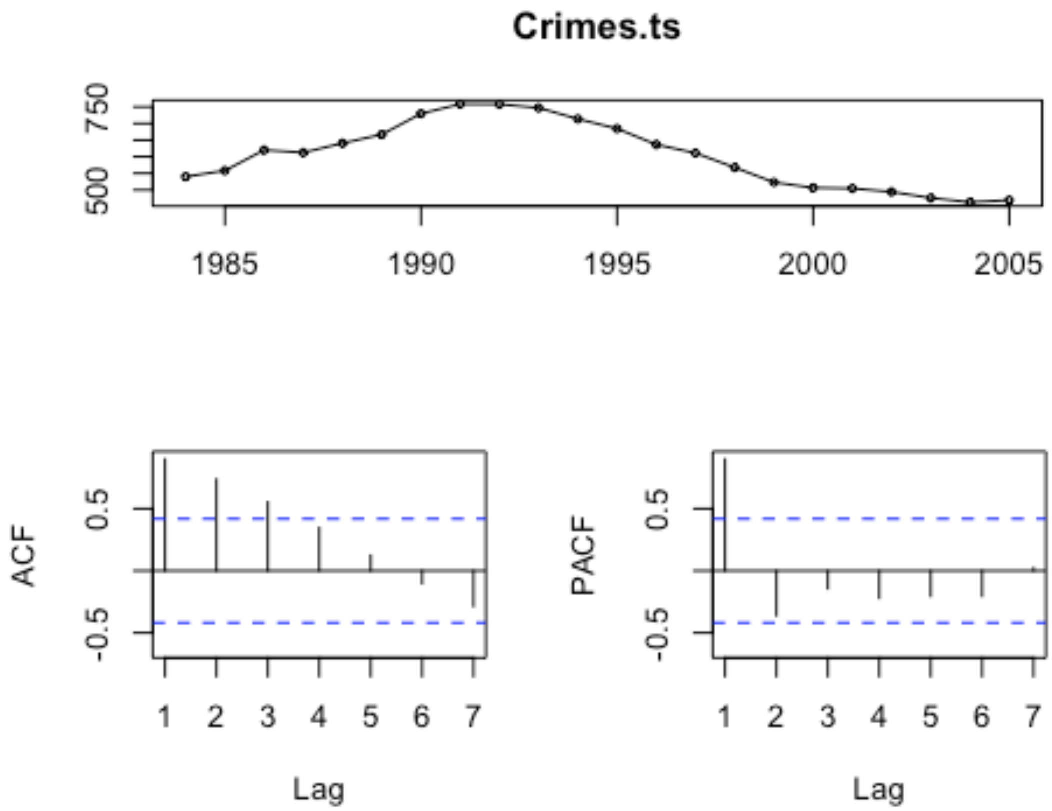
##Answer The initial time series data turns out to be non-stationary.So, we differenced the time series twice in order to get it to a stationary state. The second difference time series ACF and PACF plots show no clear significant spike.

```
Crimes.data <- read.csv("~/Desktop/GradSchool/STATS 560 Time Series 
Analysis/Lecture/TableB15.csv",
                              header = TRUE)

#Convert to time series
Crimes.ts <- ts(Crimes.data[,2],
                start = 1984)
Crimes.ts

## Time Series:
## Start = 1984 
## End = 2005 
## Frequency = 1 
##  [1] 539.9 558.1 620.1 612.5 640.6 666.9 729.6 758.2 757.7 747.1 713.6 
684.5
## [13] 636.6 611.0 567.6 523.0 506.5 504.5 494.4 475.8 463.2 469.2

#first check auto.arima()
auto.arima(Crimes.ts,
           trace = TRUE,
           stepwise = FALSE,
           approximation = FALSE)

## 
##  ARIMA(0,2,0)                    : 190.9899
##  ARIMA(0,2,1)                    : 190.0552
##  ARIMA(0,2,2)                    : 192.6437
##  ARIMA(0,2,3)                    : Inf
##  ARIMA(0,2,4)                    : Inf
##  ARIMA(0,2,5)                    : Inf
##  ARIMA(1,2,0)                    : 189.3732
##  ARIMA(1,2,1)                    : 191.2572
##  ARIMA(1,2,2)                    : 193.8158
##  ARIMA(1,2,3)                    : 193.4443
##  ARIMA(1,2,4)                    : Inf
##  ARIMA(2,2,0)                    : 191.8749
##  ARIMA(2,2,1)                    : 194.4068
##  ARIMA(2,2,2)                    : Inf
##  ARIMA(2,2,3)                    : Inf
##  ARIMA(3,2,0)                    : 193.1617
```

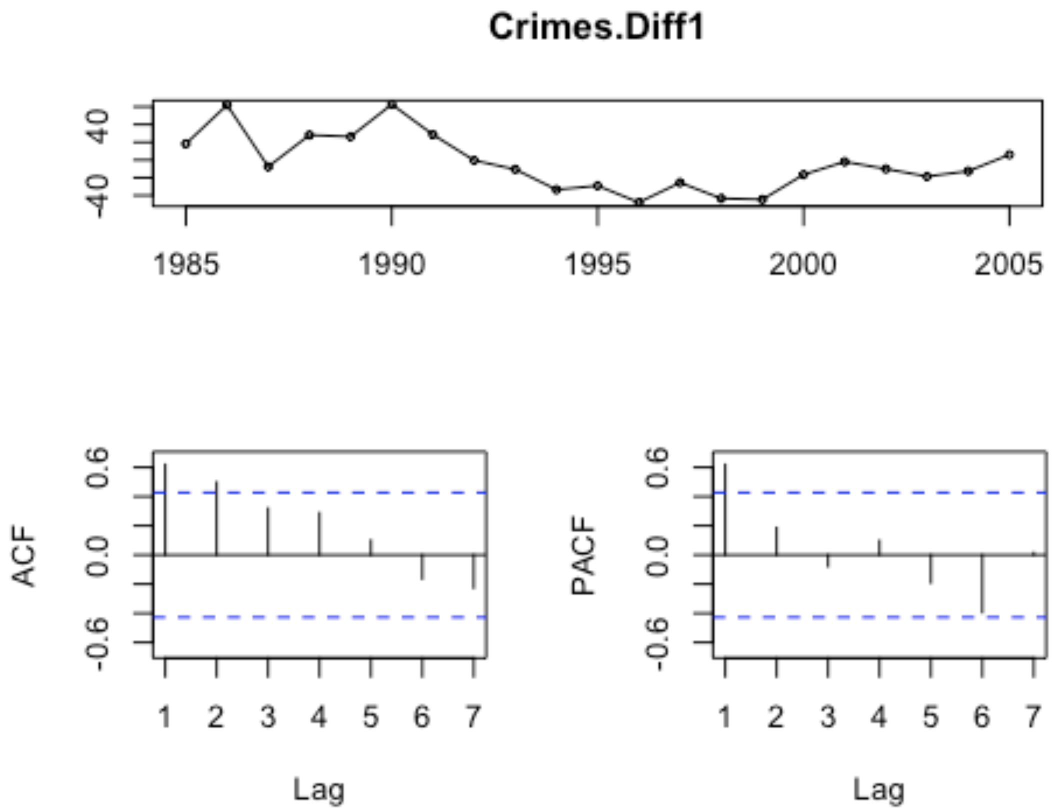
```
##  ARIMA(3,2,1)                    : 196.7312
##  ARIMA(3,2,2)                    : Inf
##  ARIMA(4,2,0)                    : 196.637
##  ARIMA(4,2,1)                    : 200.6166
##  ARIMA(5,2,0)                    : 199.5054
##
##
##
##  Best model: ARIMA(1,2,0)
```

```
## Series: Crimes.ts
## ARIMA(1,2,0)
##
## Coefficients:
##           ar1
##       -0.4533
## s.e.   0.2091
##
## sigma^2 estimated as 623.5:  log likelihood=-92.33
## AIC=188.67   AICc=189.37   BIC=190.66
```

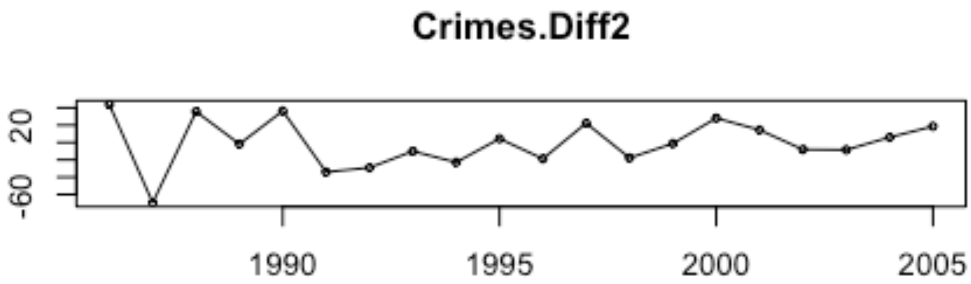
```
###Stationarity check
#ACF and PACF plots
tsdisplay(Crimes.ts)
```

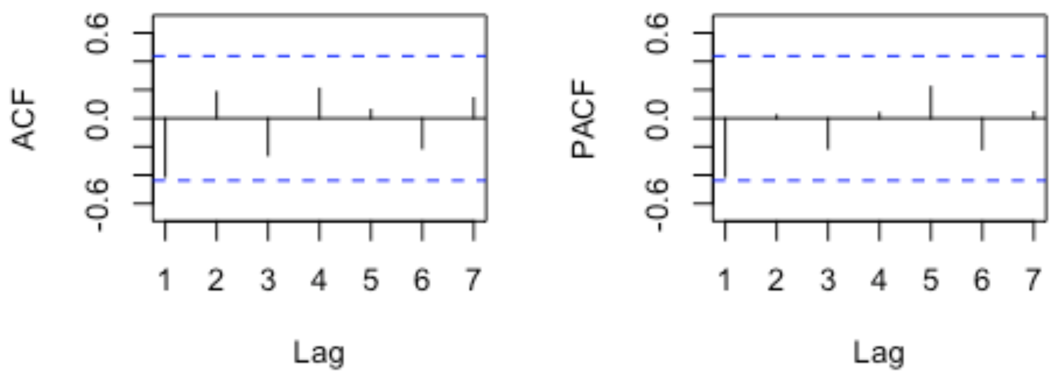



```
#First order differencing
Crimes.Diff1 <- diff(Crimes.ts, differences = 1)

###checking stationarity of the 1st order difference time series
#First difference ACF and PACF plots
tsdisplay(Crimes.Diff1)
```




```r
#Second order differencing
Crimes.Diff2 <- diff(Crimes.Diff1, differences = 1)

###checking stationarity of the 2nd order difference time series
#Second difference ACF and PACF plots
tsdisplay(Crimes.Diff2)
```





ARIMA(1,2,0) model with estimator coefficient

$$w_t^2 = -0.4533 * w_{t-1} + e_t$$

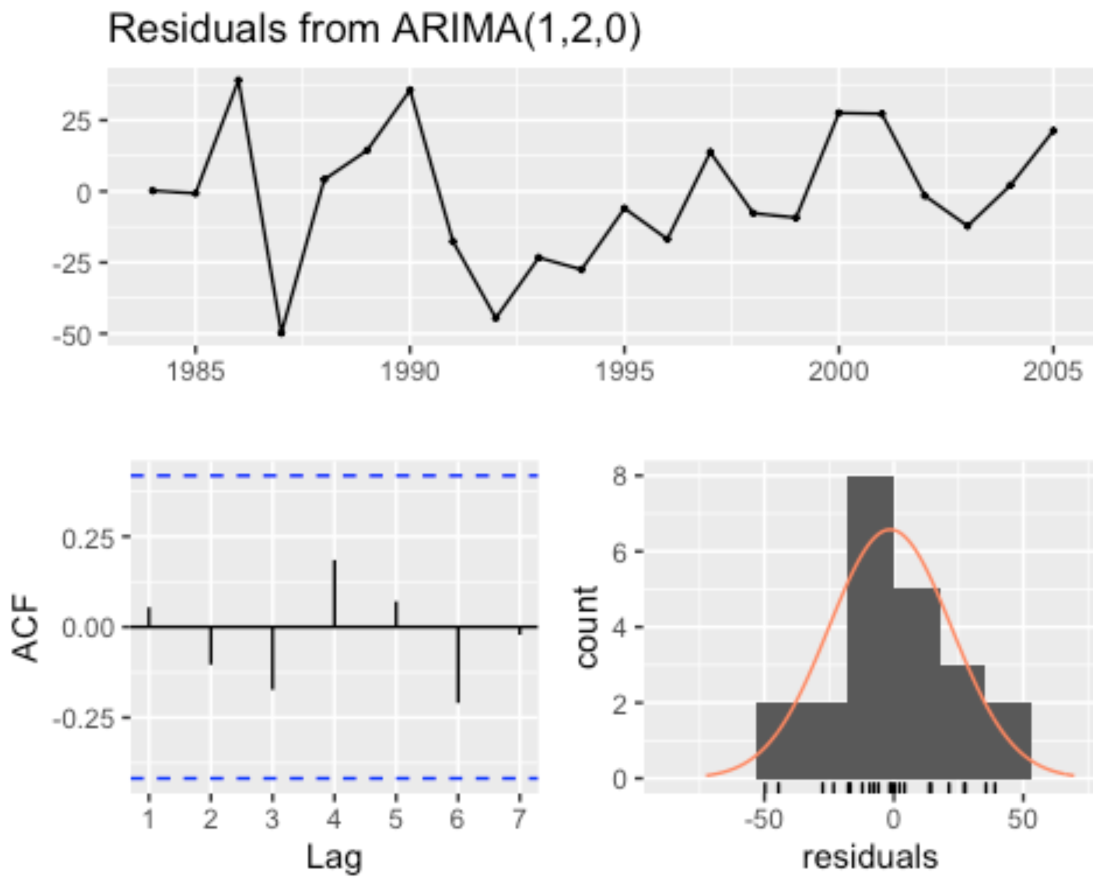
seems to be the most promising out of the proposed models. It has the lowest criteria information.The residual ACF doesn't show any spikes crossing the significant threshold boundary. The residual histogram looks fairly normally distributed with a hint of a left skewness.

Lastly,prediction interval can be computed by applying the variance operator to the forecast errors, meaning we estimate the prediction error variance using the forecasting equation and plug it into the standard prediction interval equation.

$$\hat{y}_{T+\tau}(T) \pm (1.96) * \sigma(\tau)$$

```r
arima.020 <- arima(Crimes.ts, order=c(0,2,0))
arima.120 <- arima(Crimes.ts, order=c(1,2,0))
arima.220 <- arima(Crimes.ts, order=c(2,2,0))
arima.021 <- arima(Crimes.ts, order=c(0,2,1))
arima.022 <- arima(Crimes.ts, order=c(0,2,2))
paste("ARIMA(0,2,0) AIC")

## [1] "ARIMA(0,2,0) AIC"
```

```r
AIC(arima.020)

## [1] 190.7677

paste("ARIMA(1,2,0) AIC")

## [1] "ARIMA(1,2,0) AIC"

AIC(arima.120)

## [1] 188.6674

paste("ARIMA(2,2,0) AIC")

## [1] "ARIMA(2,2,0) AIC"

AIC(arima.220)

## [1] 190.3749

paste("ARIMA(0,2,1) AIC")

## [1] "ARIMA(0,2,1) AIC"

AIC(arima.021)

## [1] 189.3494

paste("ARIMA(0,2,0) AIC")

## [1] "ARIMA(0,2,0) AIC"

AIC(arima.020)

## [1] 190.7677

#model diagnostics
checkresiduals(arima.120)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,2,0)
## Q* = 2.2032, df = 3, p-value = 0.5313
##
## Model df: 1.   Total lags used: 4
```

```
#1 to 10 step ahead Forecasting

forecast10 <- as.data.frame(forecast(arima.120))
forecast10
```

```
##      Point Forecast     Lo 80    Hi 80      Lo 95    Hi 95
## 2006       466.7685 435.57949 497.9575  419.06902 514.4680
## 2007       468.1591 410.71483 525.6033  380.30568 556.0124
## 2008       467.8171 375.94389 559.6902  327.30918 608.3249
## 2009       468.2604 338.06244 598.4584  269.13979 667.3811
## 2010       468.3478 295.21144 641.4842  203.55859 733.1370
## 2011       468.5965 248.80248 688.3906  132.45054 804.7425
## 2012       468.7721 198.69568 738.8486   55.72585 881.8184
## 2013       468.9809 145.31996 792.6418  -26.01578 963.9775
```

```
## 2014       469.1746  88.78690 849.5623 -112.57818 1050.9274
## 2015       469.3751  29.29789 909.4524 -203.66493 1142.4152

#Calculating 95% PI
paste("Lower bound 95% prediction inverval")

## [1] "Lower bound 95% prediction inverval"

data.frame(forecast10$`Lo 95`)

##    forecast10..Lo.95.
## 1           419.06902
## 2           380.30568
## 3           327.30918
## 4           269.13979
## 5           203.55859
## 6           132.45054
## 7            55.72585
## 8           -26.01578
## 9          -112.57818
## 10         -203.66493

paste("Lower bound 95% prediction inverval")

## [1] "Lower bound 95% prediction inverval"

data.frame(forecast10$`Hi 95`)

##    forecast10..Hi.95.
## 1            514.4680
## 2            556.0124
## 3            608.3249
## 4            667.3811
## 5            733.1370
## 6            804.7425
## 7            881.8184
## 8            963.9775
## 9           1050.9274
## 10          1142.4152
```

$$w_t^2 = -0.4533 * w_{t-1} + e_t$$