STAT 560 Final Exam

Amin Baabol

12/3/2020

Question 1:

(5 points) Plot the crime rate data vs the year.

Discussion:

The plot indicates a steady rise in crime rate from 1984 up to around 1992, at which point there is a significant drop in crime rate.

```
library(readxl)
library(ggplot2)
library(forecast)
```

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

```
library(tseries)
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

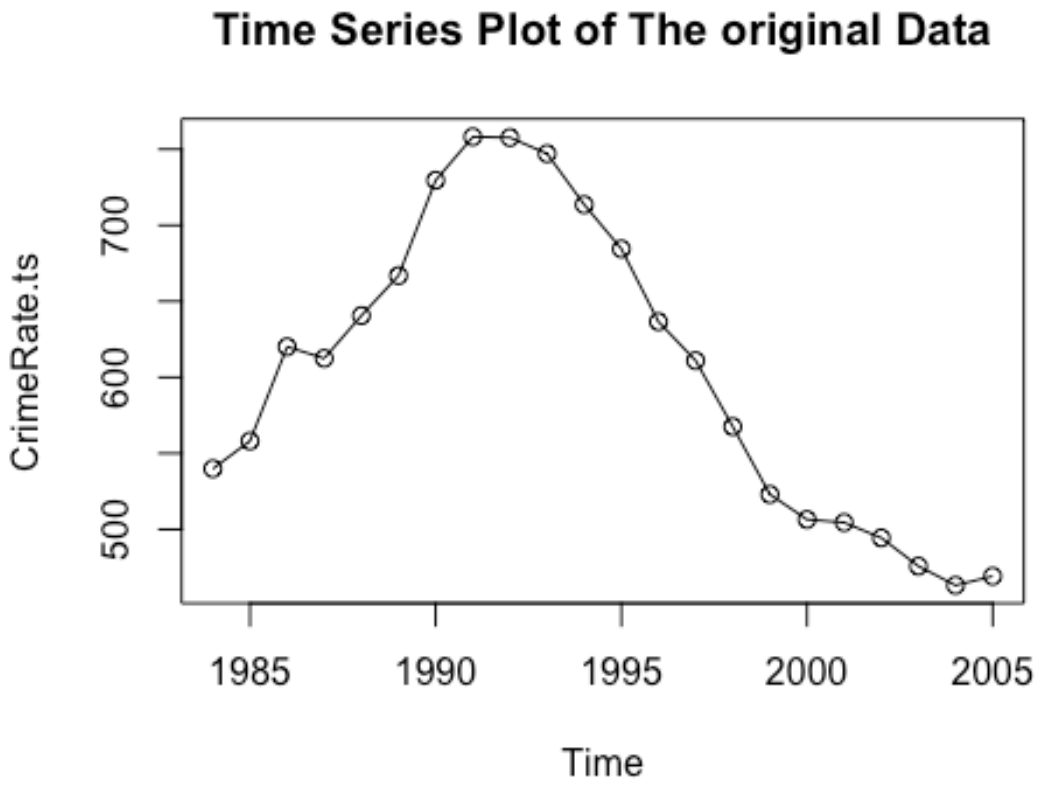
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(gridExtra)
```

```
Final_data <- data.frame(read_excel("~/Desktop/GradSchool/STATS 560 Time
Series Analysis/Exams/Final_data.xlsx"))
head(Final_data)
```

```
##   Year  Rate
## 1 1984 539.9
## 2 1985 558.1
## 3 1986 620.1
## 4 1987 612.5
## 5 1988 640.6
## 6 1989 666.9
```

```
#Converting it to time series
CrimeRate.ts = ts(Final_data[,2], start = 1984, end = 2005,frequency = 1)
CrimeRate.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

#year vs. crime rate time series plot
Original.ts <- plot(CrimeRate.ts, type = "o",
                    main = "Time Series Plot of The original Data")
```




```
Original.ts

## NULL
```

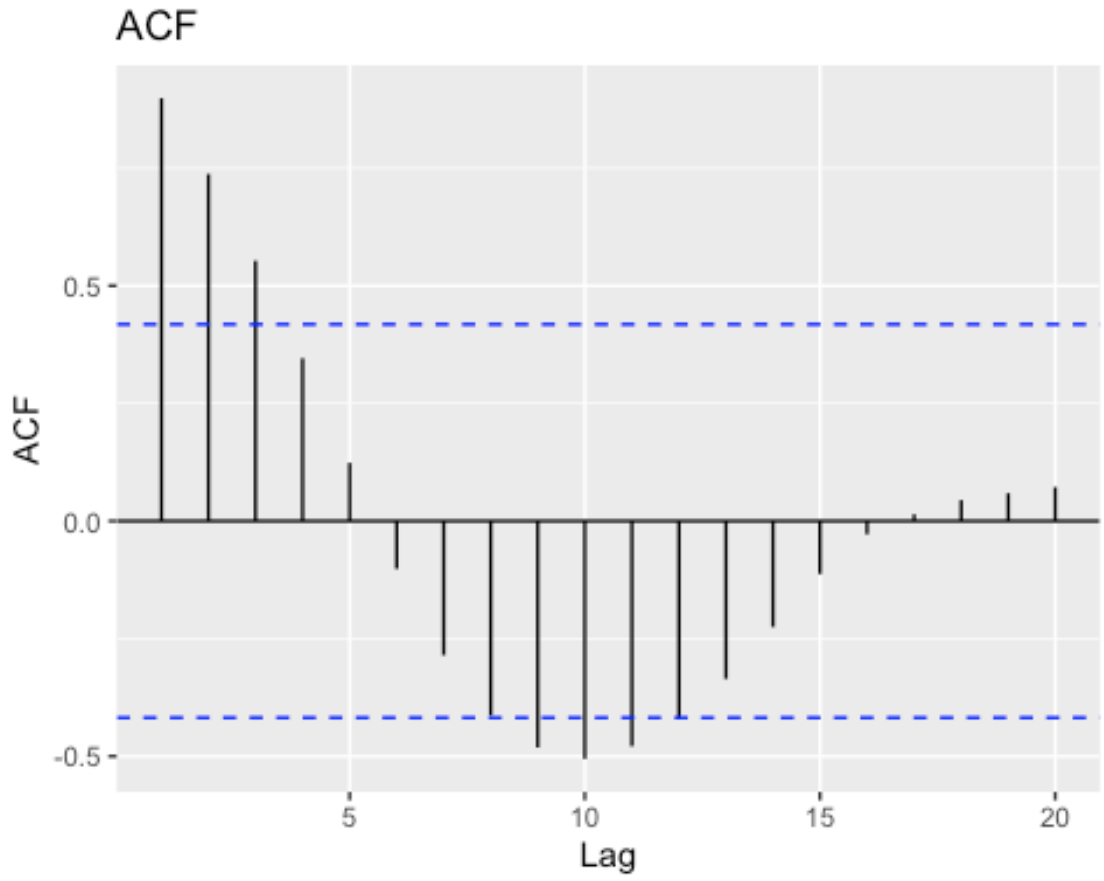
Question 2:

(10 points) Calculate and plot the sample autocorrelation function (ACF) and variogram.

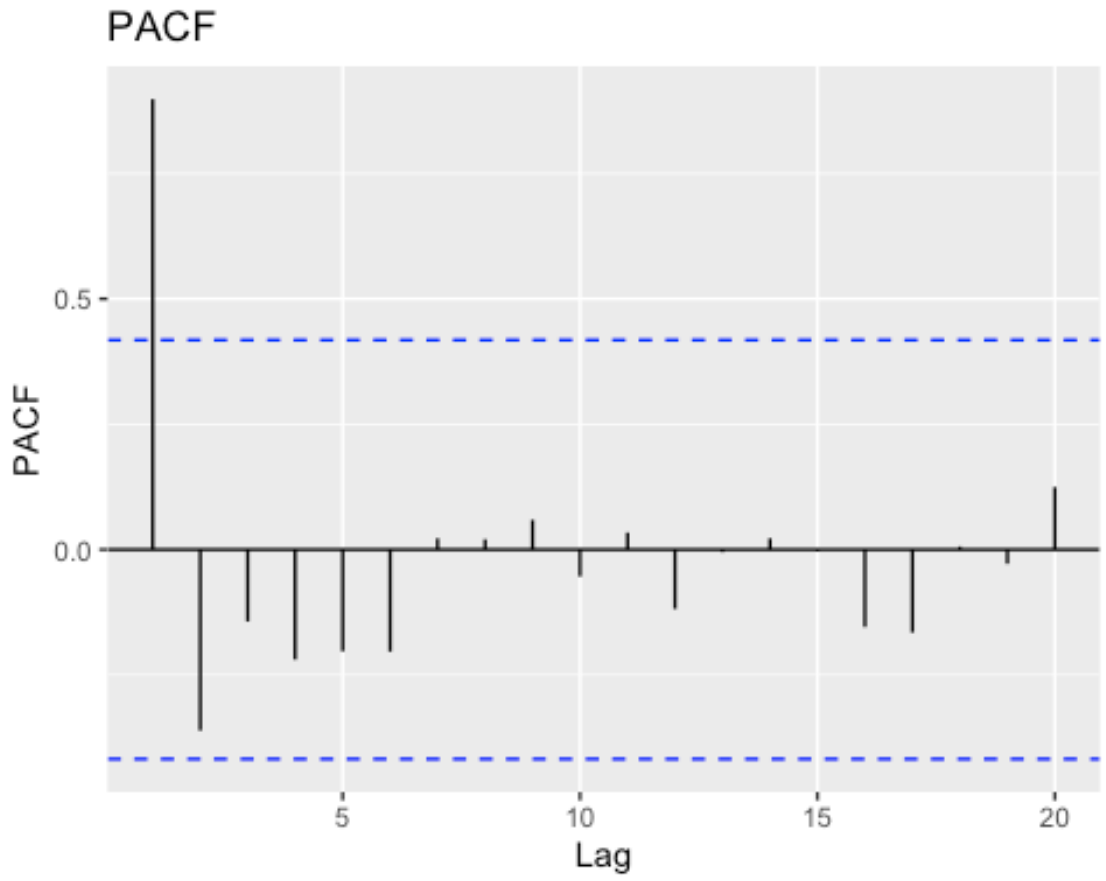
Discussion:

The ACF shows a sinusoidal pattern and significant higher lags.This indicates the suspected nonstationary time series.Therefore, differencing this time series is recommended moving forward.

```
set.seed(1242)
#Calcuation
ACF1.calculation <- acf(CrimeRate.ts, plot = FALSE)
PACF1.Calculation <- pacf(CrimeRate.ts, plot = FALSE)


#ACF plot
Original.ACF <- ggAcf(CrimeRate.ts,lag.max = 20)+labs(title = "ACF")
Original.PACF <- ggPacf(CrimeRate.ts,lag.max = 20)+labs(title = "PACF")
Original.ACF
```

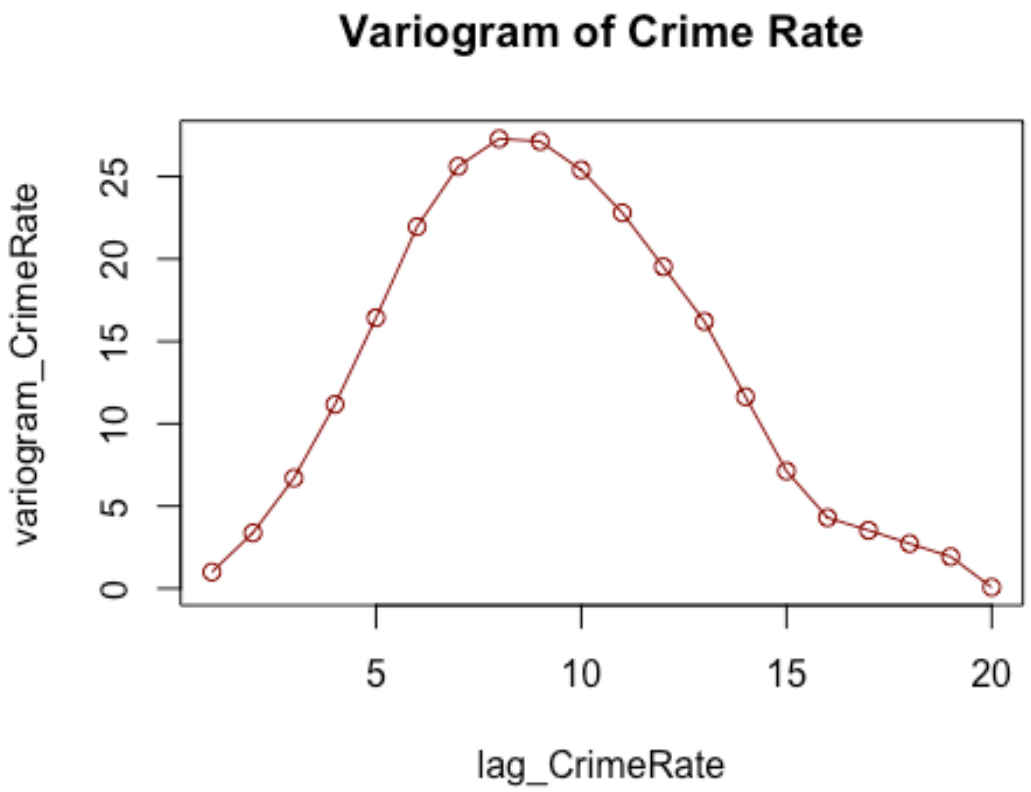



```
Original.PACF
```




```r
#Variogram
# Define the variogram function:from Dr. Fan's slides
variogram_func <- function(x, lag) {
  x <- as.matrix(x)
  Lag <- NULL
  var_k <- NULL
  vario <- NULL
  for (k in 1:lag) {
    Lag[k] <- k
    var_k[k] <- sd(diff(x, k))^2
    vario[k] <- var_k[k] / var_k[1]
    }
  return(as.data.frame(cbind(Lag, vario)))
}


x <- CrimeRate.ts
lag_length <- 20
lag_CrimeRate <- 1:lag_length
z <- variogram_func(x, lag_length)
variogram_CrimeRate <- z$vario
variogram_CrimeRate
```

```
##  [1]  1.00000000  3.37766165  6.70389263 11.17362914 16.42973876
21.95785351
##  [7] 25.61664540 27.29136513 27.13142750 25.40494395 22.80079606
19.52618728
## [13] 16.21070146 11.63960678  7.12772729  4.29925348  3.53141389
2.72671167
## [19]  1.93177934  0.07423935

#Crime rate variogram plot
Orginal.variogram <- plot(lag_CrimeRate, variogram_CrimeRate,
                          type = "o",
                          col = "dark red",
                          main = "Variogram of Crime Rate")
```




```
Orginal.variogram

## NULL

#frist 10 ACF and variogram values
paste("First 10 ACF Values")

## [1] "First 10 ACF Values"

ACF1.calculation[1:10]
```

```
##
## Autocorrelations of series 'CrimeRate.ts', by lag
##
##      1      2      3      4      5      6      7      8      9     10
##  0.898  0.737  0.553  0.346  0.124 -0.102 -0.285 -0.413 -0.481 -0.505
```

```
paste("First 10 PACF Values")
```

```
## [1] "First 10 PACF Values"
```

```
PACF1.Calculation[1:10]
```

```
##
## Partial autocorrelations of series 'CrimeRate.ts', by lag
##
##      1      2      3      4      5      6      7      8      9     10
##  0.898 -0.361 -0.144 -0.220 -0.203 -0.204  0.022  0.021  0.059 -0.054
```

```
paste("First 10 variogram Values")
```

```
## [1] "First 10 variogram Values"
```

```
variogram_CrimeRate[1:10]
```

```
##  [1]  1.000000  3.377662  6.703893 11.173629 16.429739 21.957854 25.616645
##  [8] 27.291365 27.131428 25.404944
```

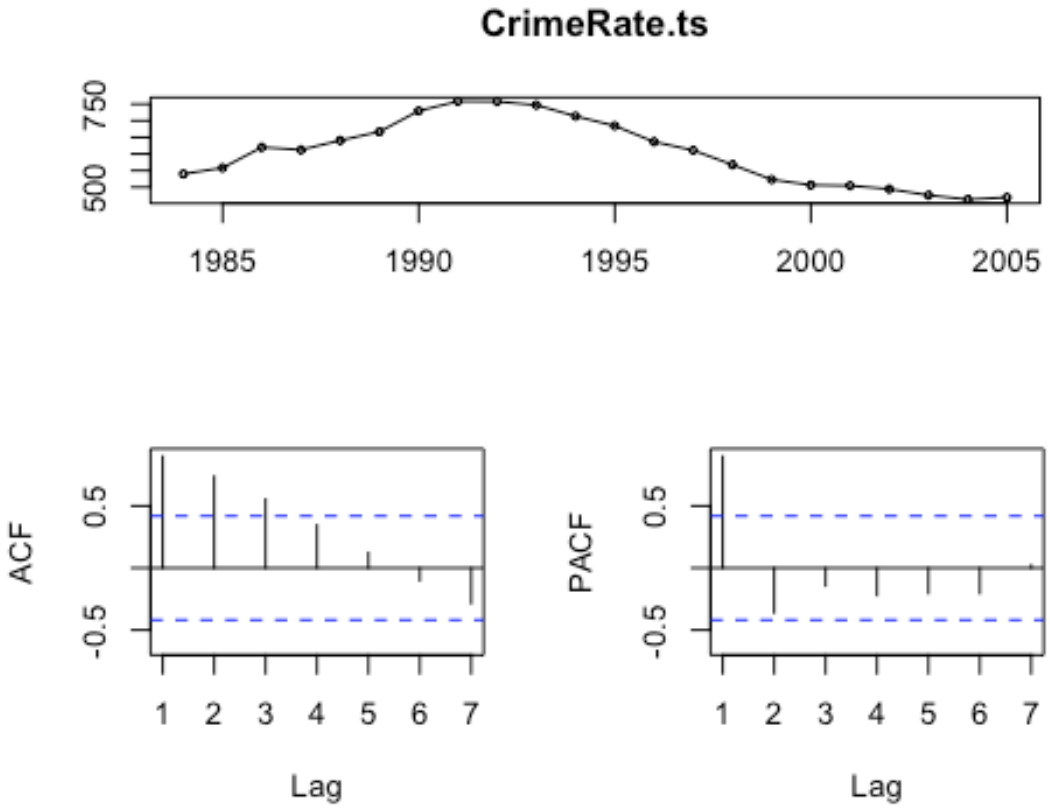
Question 3:

#(5 points) Is there an indication of nonstationary behavior in the time series? Why or why not?

Discussion:

The time series plot, ACF and variogram all agree that the crime rate is non-stationary time series.There is an apparent decreasing trend.The ACF plot in particular,shows an oscillating trend crossing the significant threshold at the first three lags as well as latter lags.While the variogram shows the increasing,decreasing trend. The Dicke-Fuller test also supports this interpretation having a p-value of 0.5932. This p-value is not statistically significant enough to reject the null hypothesis that the time series is non-stationary.

```
tsdisplay(CrimeRate.ts)
```



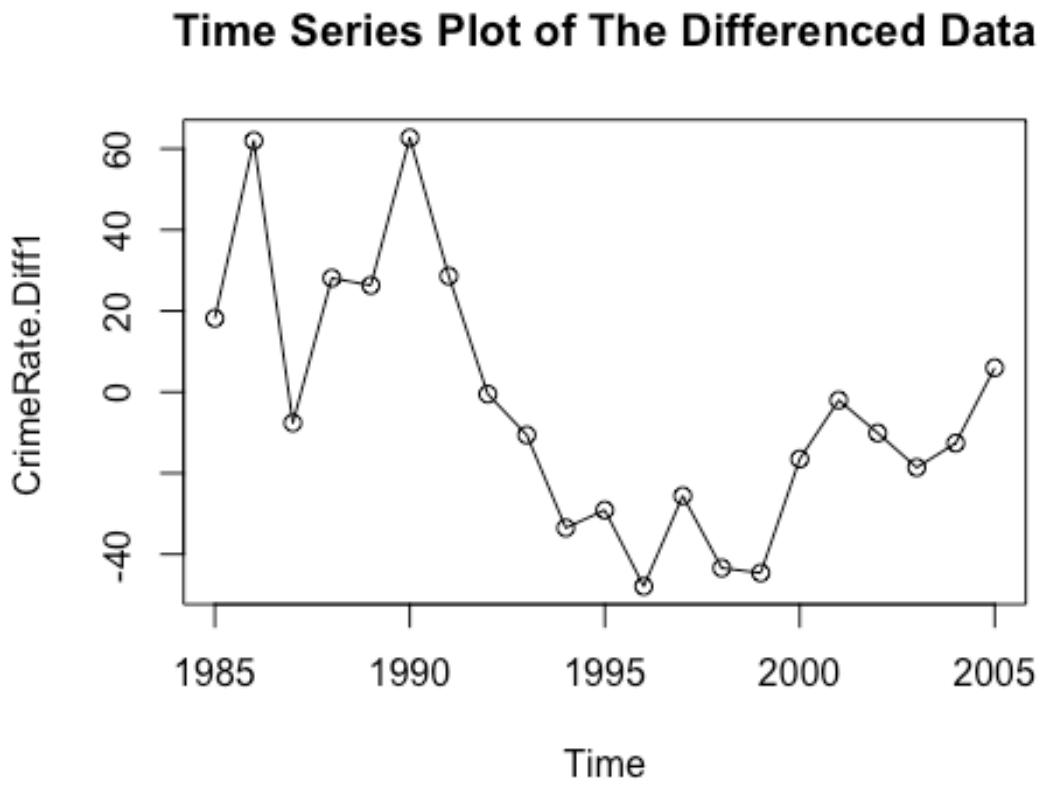

```
#Stationarity check :Dicke-Fuller test
adf.test(CrimeRate.ts)

##
##  Augmented Dickey-Fuller Test
##
## data:  CrimeRate.ts
## Dickey-Fuller = -1.9454, Lag order = 2, p-value = 0.5932
## alternative hypothesis: stationary
```

Question 4:

(10 points) Calculate and plot the first difference of the time series. Show the first 10 differences.

```
#calculating the first differencing
CrimeRate.Diff1 <- diff(ts(Final_data[,2],start = 1984, end = 2005,frequency
= 1),
                        differences = 1)
CrimeRate.Diff1

## Time Series:
## Start = 1985
## End = 2005
```

```
## Frequency = 1
##  [1]  18.2  62.0  -7.6  28.1  26.3  62.7  28.6  -0.5 -10.6 -33.5 -29.1 -
47.9
## [13] -25.6 -43.4 -44.6 -16.5  -2.0 -10.1 -18.6 -12.6   6.0

# plot time series of the first difference
First.Diff <- plot(CrimeRate.Diff1, type = "o",
                   main = "Time Series Plot of The Differenced Data")
```

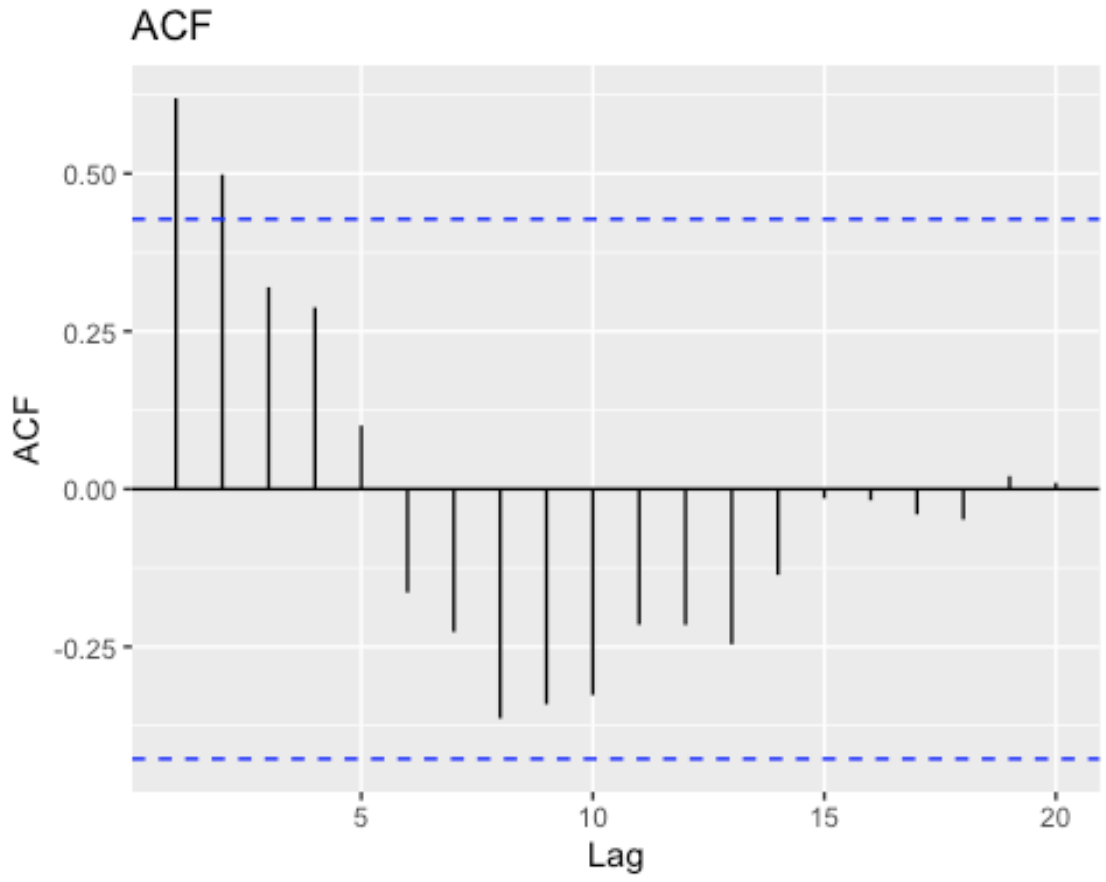



```
First.Diff

## NULL

#First 10 differences
paste("First ten differences")

## [1] "First ten differences"

head(CrimeRate.Diff1,10)

## Time Series:
## Start = 1985
## End = 1994
## Frequency = 1
##  [1]  18.2  62.0  -7.6  28.1  26.3  62.7  28.6  -0.5 -10.6 -33.5
```

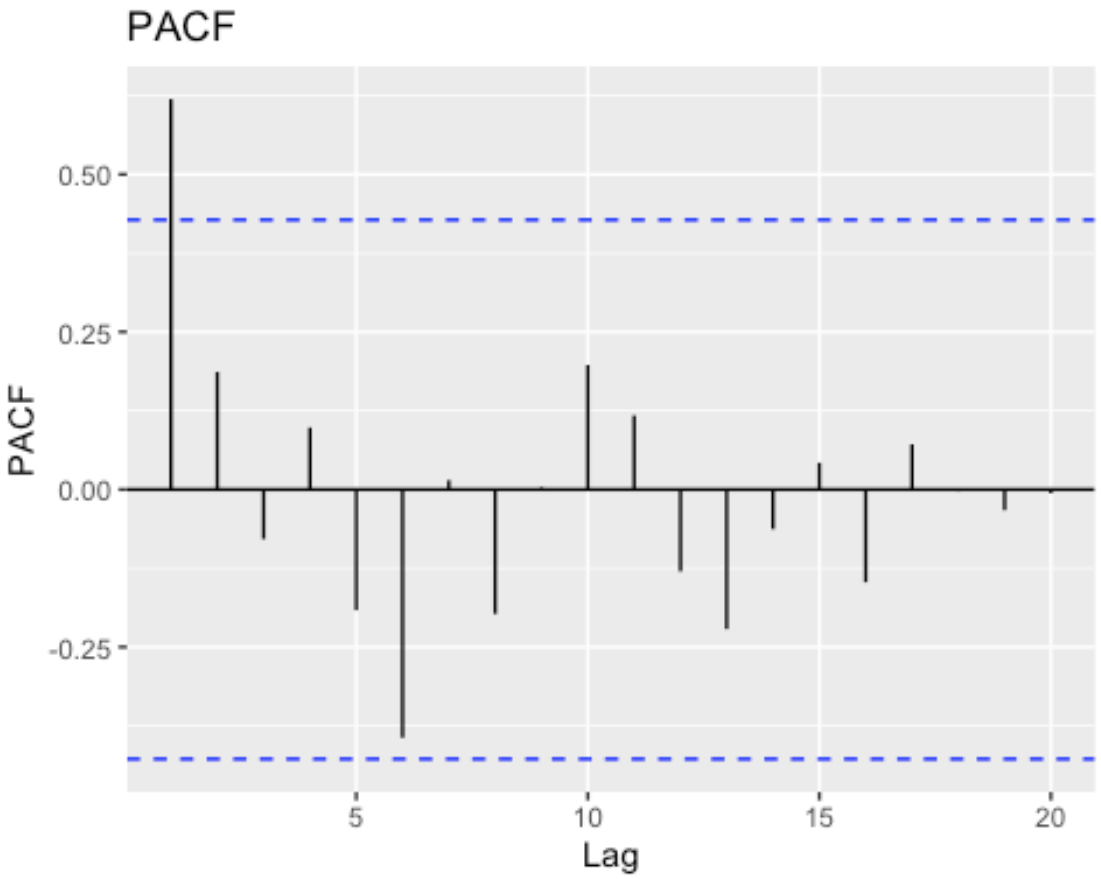
Question 5:

(10 points) Compute the sample autocorrelation function (ACF) and variogram of the first differences.

```
#Stationarity check:ACF and PACF plots
ACF.diff1 <- acf(CrimeRate.Diff1, plot = FALSE)
PACF.diff1 <- pacf(CrimeRate.Diff1, plot = FALSE)
ACF.Diff1.Plot <- ggAcf(CrimeRate.Diff1,lag.max = 20)+labs(title = "ACF")
PACF.Diff1.Plot <- ggPacf(CrimeRate.Diff1,lag.max = 20)+labs(title = "PACF")

ACF.Diff1.Plot
```



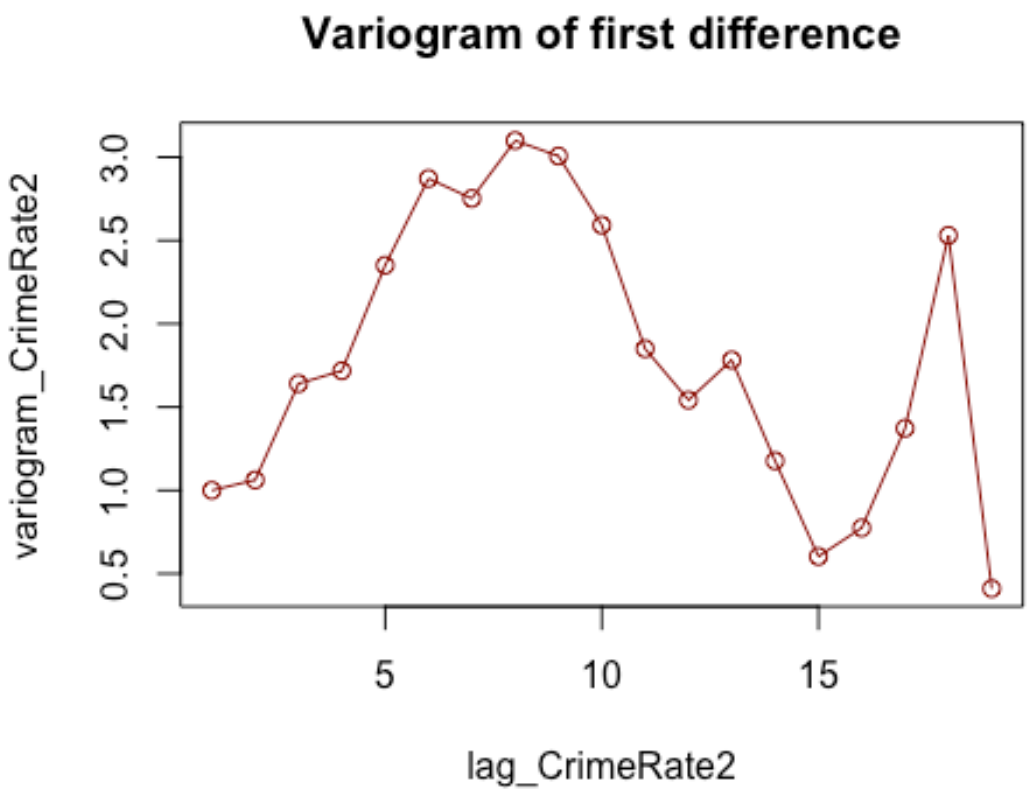

```
PACF.Diff1.Plot
```




```
#Variogram of the first differences

x2 <- CrimeRate.Diff1
lag_length2 <- 19
lag_CrimeRate2 <- 1:lag_length2
z2 <- variogram_func(x2, lag_length2)
variogram_CrimeRate2 <- z2$vario
variogram_CrimeRate2
```

```
##  [1] 1.0000000 1.0625431 1.6405991 1.7179228 2.3514208 2.8718687 2.7523149
##  [8] 3.0997290 3.0068767 2.5919156 1.8502498 1.5418690 1.7822678 1.1771419
## [15] 0.6030835 0.7744814 1.3727564 2.5304664 0.4103459
```

```
# First difference variogram plot
first.diff.variogram <- plot(lag_CrimeRate2, variogram_CrimeRate2,
                             type = "o",
                             col = "dark red",
                             main = "Variogram of first difference")
```




```
first.diff.variogram
```

```
## NULL
```

```
#frist 10 ACF and variogram values fo the differences time seires
paste("First 10 ACF values of the differences time series")
```

```
## [1] "First 10 ACF values of the differences time series"
```

```
ACF.diff1[1:10]
```

```
##
## Autocorrelations of series 'CrimeRate.Diff1', by lag
##
##      1      2      3      4      5      6      7      8      9     10
##  0.619  0.499  0.320  0.288  0.101 -0.164 -0.227 -0.364 -0.341 -0.327
```

```
paste("First 10 PACF values of the differenced time series")
```

```
## [1] "First 10 PACF values of the differenced time series"
```

```
PACF.diff1[1:10]
```

```
##
## Partial autocorrelations of series 'CrimeRate.Diff1', by lag
```

```
##
##      1      2      3      4      5      6      7      8      9     10
##  0.619  0.186 -0.079  0.098 -0.192 -0.394  0.015 -0.198  0.003  0.197
```

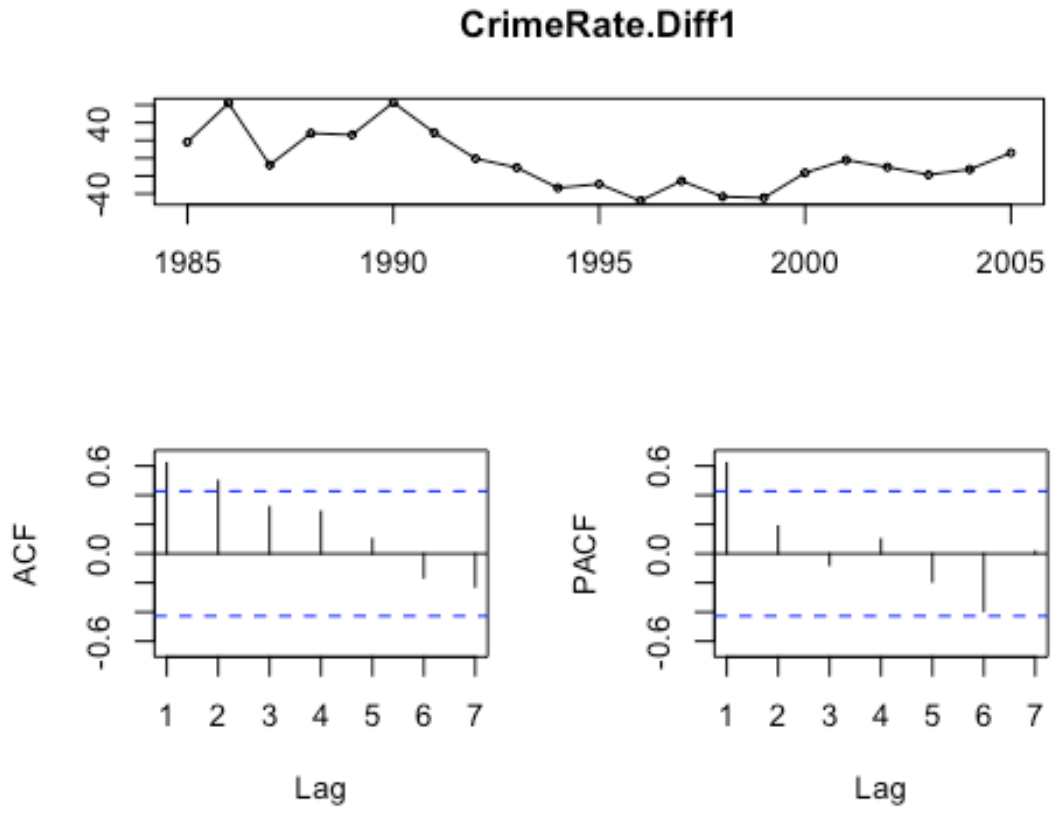
```
paste("First 10 variogram Values of differenced time series")
```

```
## [1] "First 10 variogram Values of differenced time series"
```

```
variogram_CrimeRate2[1:10]
```

```
##  [1] 1.000000 1.062543 1.640599 1.717923 2.351421 2.871869 2.752315
3.099729
##  [9] 3.006877 2.591916
```

```
#Further stationarity checks
tsdisplay(CrimeRate.Diff1)
```




```
adf.test(CrimeRate.Diff1);
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  CrimeRate.Diff1
## Dickey-Fuller = -1.3807, Lag order = 2, p-value = 0.8083
## alternative hypothesis: stationary
```

```
pp.test(CrimeRate.Diff1);

##
##  Phillips-Perron Unit Root Test
##
## data:  CrimeRate.Diff1
## Dickey-Fuller Z(alpha) = -11.055, Truncation lag parameter = 2, p-value
## = 0.4045
## alternative hypothesis: stationary

kpss.test(CrimeRate.Diff1)

##
##  KPSS Test for Level Stationarity
##
## data:  CrimeRate.Diff1
## KPSS Level = 0.42191, Truncation lag parameter = 2, p-value = 0.06771
```

Question 6:

(5 points) What impact has differencing had on the time series?

Discussion:

The differencing was intending to remove the changing levels of the original time time series data, so it detrended the time series.The ACF plot immediately lost the oscillation seen in the original time series.My only concern is that there is still random fluctuations shown by the first differenced time series plot as well as the variogram plot.This indicates further diferrencing may be required moving forward.

Question 7:

Develop an appropriate exponential smoothing forecasting procedure for the first-differencing data by answering the questions below.

Part a:

(10 points) Assume the first-difference data is a constant process. For R user, use the HoltWinters() function to find the optimum value of λ to smooth the data. For JMP user, specify the λ given by the software.

##Discussion Holtwinter's method indicates a lambda value of 0.592 as the optimum lambda which gives us a sum error squared of 12382.3.

```
CrimeRate.fit <- HoltWinters(CrimeRate.Diff1, beta=FALSE, gamma=FALSE)

CrimeRate.fit$alpha

## [1] 0.5924839
```

Part b:

(10 points) Show the fitted values and corresponding SSE by using the λ obtained in part a.

##Discussion The fitted values are printed down and the sum squared error is 12382.3.

```
set.seed(34378)
Holt.Model1 <- HoltWinters(CrimeRate.Diff1, beta=FALSE, gamma=FALSE)
Holt.Model1$fitted[,2]

## Time Series:
## Start = 1986
## End = 2005
## Frequency = 1
##  [1]  18.2000000  44.1507967  13.4892806  22.1458972  24.6071364
47.1765465
##  [7]  36.1702410  14.4437120  -0.3942853 -20.0088897 -25.3952266 -
38.7289435
## [13] -30.9502553 -38.3265291 -42.0434599 -26.9093700 -12.1509682 -
10.9358025
## [19] -15.4767165 -13.7723081

sse <- sum((CrimeRate.Diff1-Holt.Model1$fitted[,2])^2)
sse

## [1] 12382.3

Holt.Model1$SSE

## [1] 12382.3
```

Part c:

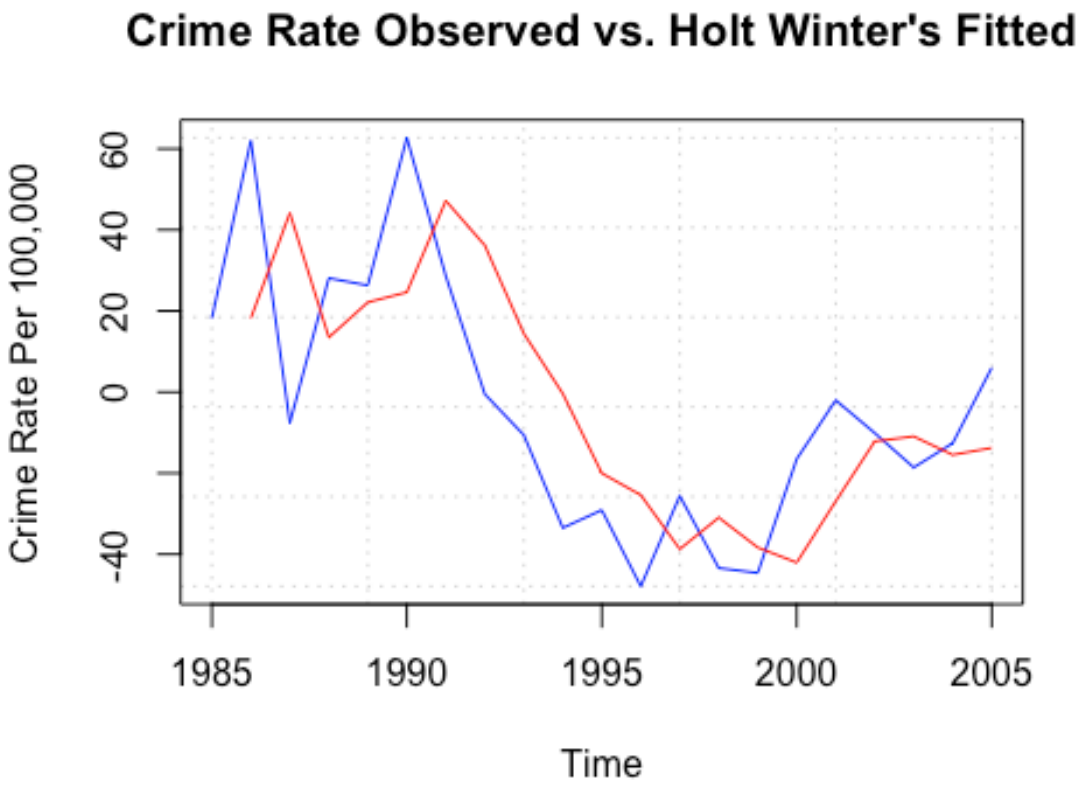
(5 points) Plot the fitted values and original values in a same plot.

Discussion

This particular Holtwinter's method does not seem to be doing a great job of fitting. The fitted values seem to be close to the boundary of the confidence interval.It does alright in capturing the general patterns but not so well in fitting with good accuracy.This is because we are applying a data with a trend on a model meant for a univariate data without a trend or season.

```
#
plot(CrimeRate.Diff1, type = "l",
     pch = 16, cex = 0.5,col = "blue",
     xlab = "Time",
     ylab = "Crime Rate Per 100,000",
     main = "Crime Rate Observed vs. Holt Winter's Fitted",
     panel.first=grid())
lines(Holt.Model1$fitted[,1],type = "l",
```

```
      cex = 0.5,
      col = "red")
legend(19,64, legend=c("Original", "Fitted"),
       col=c("blue", "red"), lty = 1:1, cex = 0.8)
```



Part d:

(5 points) Assume the first-difference data shows a trend. Calculate the SSE. You can get it from the HoltWinters() function. Then compare the SSE with that of obtained in part b. What can you tell from the comparison?

Discussion:

During the construction of this second HoltWinters model gamma was set to "FALSE" because seasonality or cyclical fluctuations was not observed in the the first difference time series. However, there was a trend which require Holt-Winters exponential smoothing. So the default function optimized the best smoothing parameters as: alpha: 0.7354783 beta : 0.44757 and sse of 21089.49 which is higher than the sse we obtained by treating the first time difference data as a constant process.The sse we obtained from the first model was 12382.3. This fitted model seems to better predict or capture the trend in the time series than the previous model where the time series was assumed to be a constant process. Although this model is still needs quite a bit more tuning before it's ready for deployment.

```
Holt.Model2 <- HoltWinters(CrimeRate.Diff1,gamma = FALSE)
Holt.Model2
```

```
## Holt-Winters exponential smoothing with trend and without seasonal
component.
##
## Call:
## HoltWinters(x = CrimeRate.Diff1, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.7354783
##  beta : 0.44757
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 1.200133
## b 6.525726
```

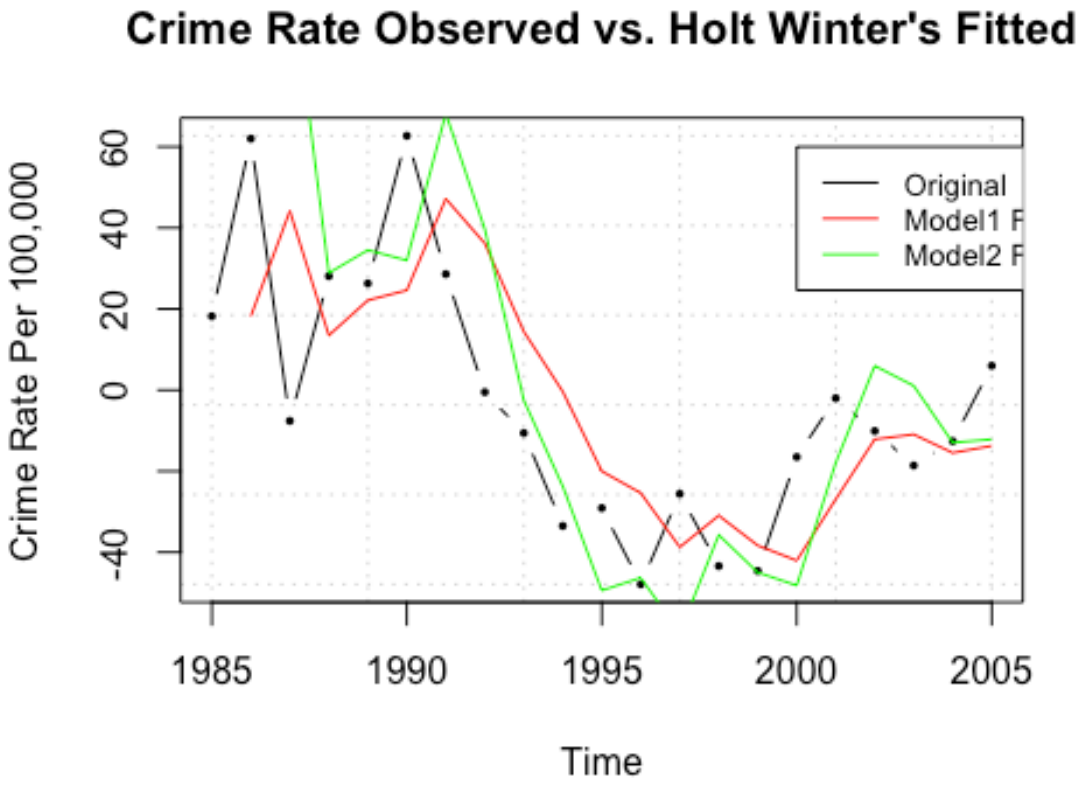
```
Holt.Model2$fitted
```

```
## Time Series:
## Start = 1987
## End = 2005
## Frequency = 1
##            xhat      level       trend
## 1987 105.800000  62.000000  43.8000000
## 1988  28.867962  22.396756   6.4712065
## 1989  34.521553  28.303143   6.2184102
## 1990  31.986834  28.474779   3.5120553
## 1991  68.197859  54.575703  13.6221561
## 1992  39.661900  39.074491   0.5874093
## 1993  -2.509315  10.123692 -12.6330077
## 1994 -23.756122  -8.459839 -15.2962837
## 1995 -49.426288 -30.922533 -18.5037545
## 1996 -46.289530 -34.476743 -11.8127862
## 1997 -59.816913 -47.473996 -12.3429178
## 1998 -35.730575 -34.651115  -1.0794601
## 1999 -44.975338 -41.371271  -3.6040668
## 2000 -48.179799 -44.699285  -3.4805138
## 2001 -17.932211 -24.879993   6.9477813
## 2002   5.977901  -6.214415  12.1923160
## 2003   1.052776  -5.847047   6.8998235
## 2004 -12.970855 -13.401415   0.4305604
## 2005 -12.145461 -12.698099   0.5526376
```

```
sse <- sum((CrimeRate.Diff1 - Holt.Model2$fitted[,1])^2)
sse
```

```
## [1] 21089.49
```

```
Holt.Model2$SSE

## [1] 21089.49

plot(CrimeRate.Diff1, type = "b",
     pch = 16, cex = 0.5,col = "black",
     xlab = "Time",
     ylab = "Crime Rate Per 100,000",
     main = "Crime Rate Observed vs. Holt Winter's Fitted",
     panel.first=grid())
lines(Holt.Model1$fitted[,1],type = "l",
      cex = 0.5,
      col = "red")
lines(Holt.Model2$fitted[,1],type = "l",
      cex = 0.5,
      col = "green")
legend(2000,60, legend=c("Original", "Model1 Fitted","Model2 Fitted"),
       col=c("black", "red" , "green"), lty = 1:1, cex = 0.8)
```



Part e:

(5 points) Suppose the first-difference is a constant process. Give the forecasts of the crime rate for years from 2006 to 2010.

Discussion:

The second model with the trend smoothing parameter seems to have larger prediction interval and larger coefficients.

```
Holt.Model1.Forecast <- predict(Holt.Model1, n.ahead = 5,
                                prediction.interval = TRUE)
Holt.Model2.Forecast <- predict(Holt.Model2, n.ahead = 5,
                                prediction.interval = TRUE)
Holt.Model1.Forecast
```

```
## Time Series:
## Start = 2006
## End = 2010
## Frequency = 1
##            fit      upr       lwr
## 2006 -2.057533 47.85900 -51.97407
## 2007 -2.057533 55.96252 -60.07758
## 2008 -2.057533 63.06536 -67.18043
## 2009 -2.057533 69.46629 -73.58136
## 2010 -2.057533 75.33964 -79.45471
```

```
Holt.Model2.Forecast
```

```
## Time Series:
## Start = 2006
## End = 2010
## Frequency = 1
##            fit       upr        lwr
## 2006  7.725858  73.73177  -58.28005
## 2007 14.251584 110.66300  -82.15983
## 2008 20.777310 154.04172 -112.48710
## 2009 27.303035 202.49918 -147.89311
## 2010 33.828761 255.28352 -187.62600
```

Question 8:

Part a:

a. (10 points) Develop an appropriate ARIMA model and a procedure for forecasting for the crime rate data. Specify the model and estimated parameters in the model. Hint: You can use the auto.arima() and forecast() functions to answer this question

##Discussion:

The specified model is an arima(1,2,0) which can be written as arima(1,0,0) after 2nd differencing.This follows logic because there was still a trend left in the first differenced time series. The 1 in the p parameter suggests that the autoregressive component is heavy handed in this model.The AR(1) coefficient is -0.4671 with mu of -1.6561.The AIC, AICc and BIC are respectively,AIC=190.48 AICc=191.98
BIC=193.46.The ACF plot indicates there is no significant autocorrelation in the lags. The

residuals are normally distributed with some skewness.The AR(1) model's coefficient can be used for forecasting in the following manner:

$$y_t = 1.6561 - 0.4533 * y_{t-1} + \epsilon_t$$

$= -0.4671 $

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

```
set.seed(54842)
auto.arima(Final_data[,2])
```

```
## Series: Final_data[, 2]
## 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
```

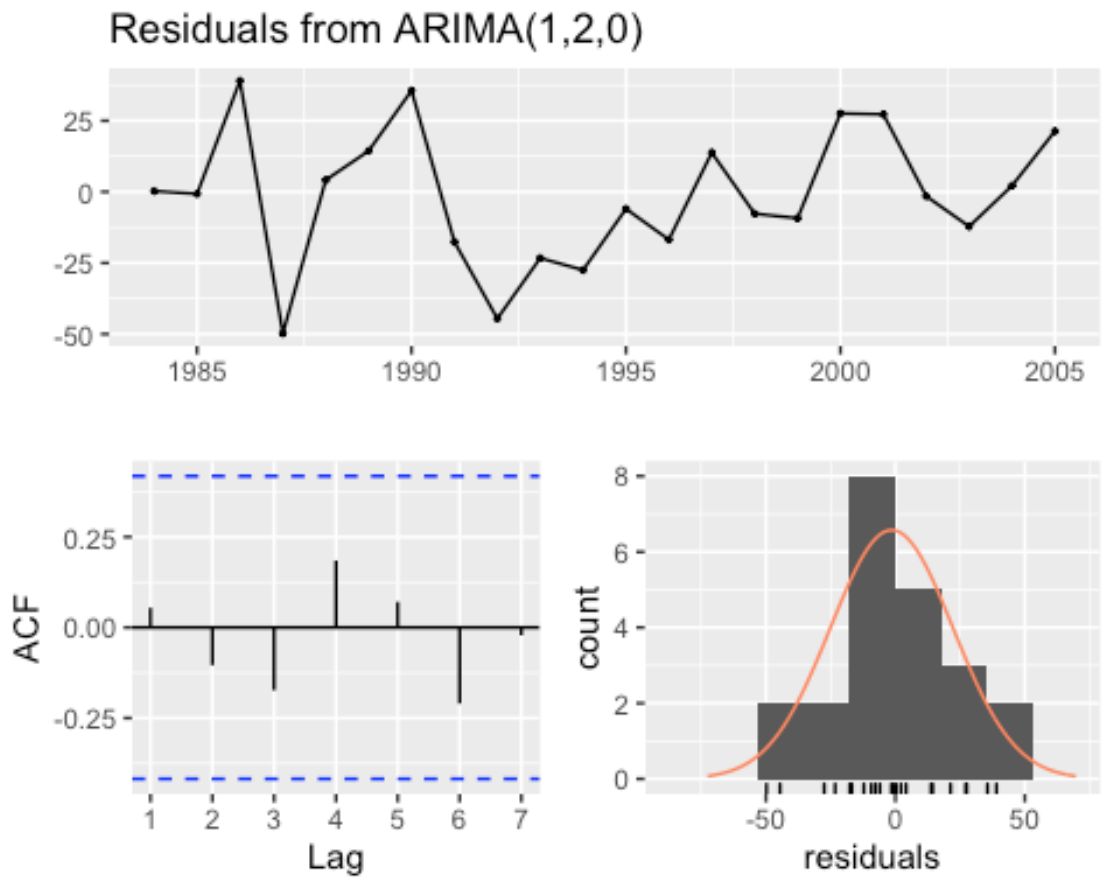
```
#ARIMA1
arima.120 <- arima(CrimeRate.ts, order=c(1,2,0))
arima.120
```

```
##
## Call:
## arima(x = CrimeRate.ts, order = c(1, 2, 0))
##
## Coefficients:
##           ar1
##       -0.4533
## s.e.   0.2091
##
## sigma^2 estimated as 592.3:  log likelihood = -92.33,  aic = 188.67
```

```
fitted<-as.vector(fitted(arima.120))
fitted
```

```
##  [1] 539.6585 558.7837 581.0593 662.2452 636.4501 652.5170 694.0160
775.7997
##  [9] 802.2577 770.3912 741.0784 690.4807 653.4055 597.2222 575.2913
532.2688
## [17] 478.9440 477.2621 495.9271 487.9718 461.0531 447.8802
```

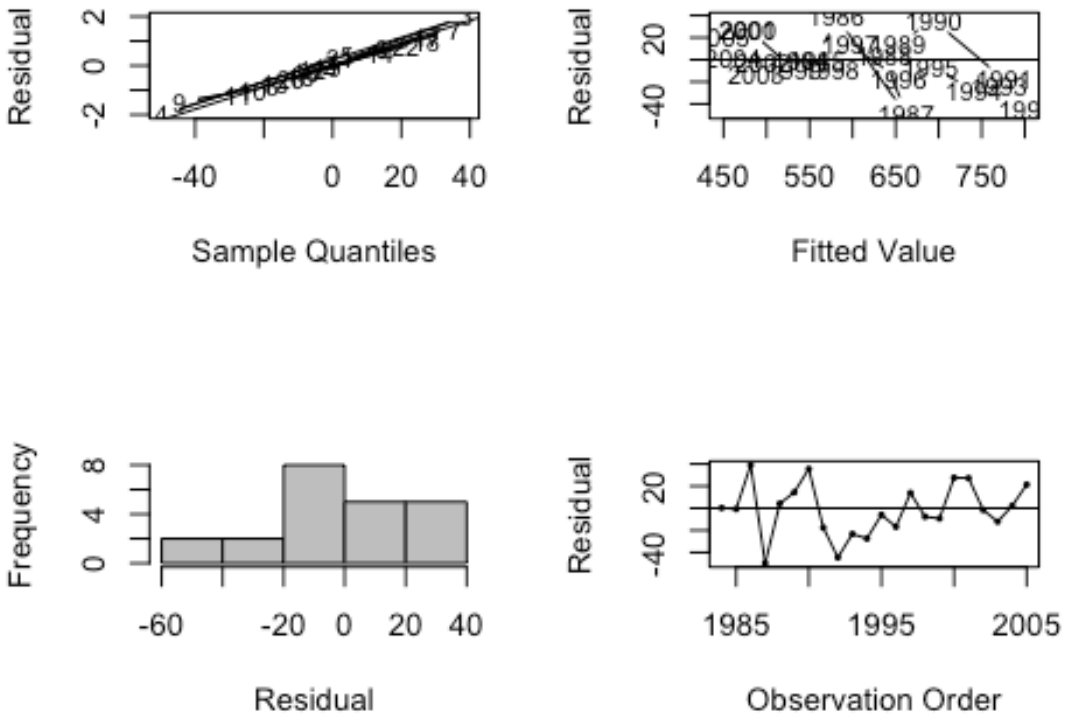
```
#Model Adequacy
checkresiduals(arima.120)
```




```
##
##  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
```

```
#4-in-1 plot of the residuals
par(mfrow = c(2,2),oma = c(0,0,0,0))
qqnorm(arima.120$residuals,
       datax = TRUE,
       pch = 16,
       xlab = 'Residual',
       main = '')
qqline(arima.120$residuals,
       datax = TRUE)
plot(fitted(arima.120),
     arima.120$residuals,
     pch = 16,
     xlab = 'Fitted Value',
     ylab = 'Residual')
abline(h = 0)
```

```
hist(arima.120$residuals,
     col = "gray",
     xlab = 'Residual',
     main = '')
plot(arima.120$residuals,
     type = "l",
     xlab = 'Observation Order',
     ylab = 'Residual')
points(arima.120$residuals,
       pch = 16,
       cex = .5)
abline(h = 0)
```




```
#forecast
forecast(arima.120,5)

##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2006       466.7685 435.5795 497.9575 419.0690 514.4680
## 2007       468.1591 410.7148 525.6033 380.3057 556.0124
## 2008       467.8171 375.9439 559.6902 327.3092 608.3249
## 2009       468.2604 338.0624 598.4584 269.1398 667.3811
## 2010       468.3478 295.2114 641.4842 203.5586 733.1370
```

Part b:

(5 points) Compare the AIC obtained from part a with that of obtained from ARIMA(0,1,0) model. Which model has a smaller AIC? What can you tell by this comparison?

Discussion:

The ARIMA(1,2,0) has a lower AIC of 188.6674, compared to ARIMA(0,1,0)'s AIC of 205.9326. This makes sense because the arima(0,1,0) is a first order differencing. We know from our earlier analysis that this time series data requires a second order differencing. Hence, ARIMA(1,2,0) model is more adequate in properly characterizing the behavior of the time series data or the residual trend after taking the first differencing.

```
#part a AIC
AIC(arima.120)

## [1] 188.6674

#Arima(0,1,0) AIC
ARIMA.2 <- Arima((Final_data[,2]), order = c(0,1,0))
AIC(ARIMA.2)

## [1] 205.9326
```

Part C:

(5 points) Show the 1- to 5- step ahead forecasts and corresponding 95% prediction intervals for the crime rate. Show only the results/outputs. Calculation process or formula are not required.

Discussion:

We estimate the prediction error variance using the forecasting equation and plug it into the standard prediction interval equation.

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

Regarding the forecasted values the lower bound of the prediction interval seems to be getting smaller the farther ahead we predict into the future. This is valid since error rate increases with with long-term forecasting.

```
step.ahead.forecast <- as.data.frame(forecast(arima.120,5))
step.ahead.forecast

##      Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 2006       466.7685 435.5795 497.9575 419.0690 514.4680
## 2007       468.1591 410.7148 525.6033 380.3057 556.0124
## 2008       467.8171 375.9439 559.6902 327.3092 608.3249
## 2009       468.2604 338.0624 598.4584 269.1398 667.3811
## 2010       468.3478 295.2114 641.4842 203.5586 733.1370
```

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

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

```
##   step.ahead.forecast..Lo.95.
## 1                     419.0690
## 2                     380.3057
## 3                     327.3092
## 4                     269.1398
## 5                     203.5586
```

```r
paste("Lower bound 95% prediction inverval")
```

```
##   step.ahead.forecast..Hi.95.
## 1                     514.4680
## 2                     556.0124
## 3                     608.3249
## 4                     667.3811
## 5                     733.1370
```