ISYE 6501

Week 3 Homework

Please see the appendix for full code

Thank you for reading!

Question 7.1

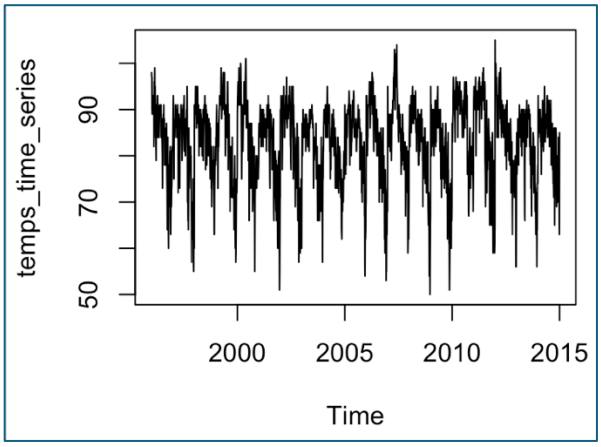
I was in charge of forecasting how many bikes would be rented during a specific time period for a city's bicycle-sharing system. I needed the number of rentals during each hour of the day for some time, which was the previous two months in this case. Since there were many different minor peaks in the data due to variation during different hours of the day, exponential smoothing would be useful to determine any trend and randomness. I would expect the value of α to be closer to 1 because the historical data provided was recent from the past two months. Furthermore, more recent observations are more important and weighted more due to the current baseline estimate. This means randomness is not as impactful compared to recent data.

Question 7.2

For exponential smoothing models, we use time series data, so I converted the imported data into a vector into a time series object.

```
# load in data
temps <- read.table("~/Downloads/temps.txt", header=TRUE)
# vector of temps data
temps_v <- as.vector(unlist(temps[,2:21]))
# vector --> time series
temps_time_series <- ts(data=temps_v,start=1996,end=2015,frequency=123)
```

I decided to plot the time series of temps to see what the data looks like.



I decided to test different exponential smoothing models: single, double, and triple.

For single exponential smoothing, there is no trend or seasonality, so there are no beta or gamma parameters.

```
Holt-Winters exponential smoothing without trend and without seasonal component.

Call:
HoltWinters(x = temps_time_series, beta = FALSE, gamma = FALSE)

Smoothing parameters:
 alpha: 0.8396301
 beta : FALSE
 gamma: FALSE

Coefficients:
      [,1]
a 81.62444
```

We get an alpha value of 0.8396301 and an SSE (sum of squared errors) of 53704.15.

Double exponential smoothing has trend and no seasonality in this instance, so we make our gamma parameter false.

```
Holt-Winters exponential smoothing with trend and without seasonal component.

Call:
HoltWinters(x = temps_time_series, gamma = FALSE)

Smoothing parameters:
 alpha: 0.8455303
 beta : 0.003777803
 gamma: FALSE

Coefficients:
          [,1]
a 81.729657393
b -0.004838906
```

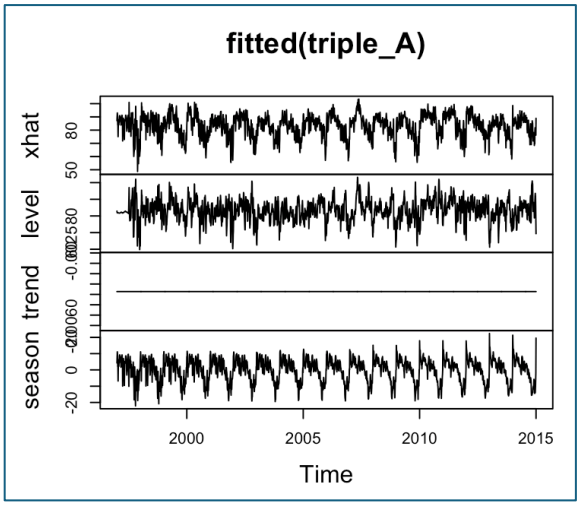
We get an alpha value of 0.8455303 and an SSE of 54071.22. We can also see that coefficient b and beta values are near 0, signifying that there is not a significant trend.

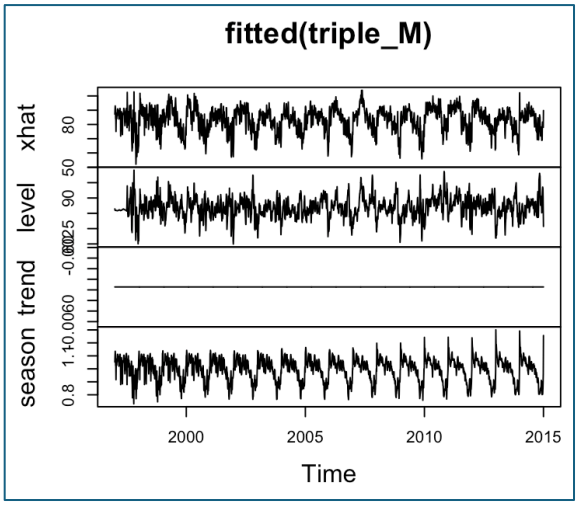
For triple exponential smoothing, there can be additive seasonality or multiplicative seasonality.

```
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = temps_time_series, seasonal = "additive")

Smoothing parameters:
 alpha: 0.6677614
 beta : 0
 gamma: 0.6297674

Coefficients:
              [,1]
a     66.739214602
b     -0.004362918
```

```
Holt-Winters exponential smoothing with trend and multiplicative seasonal component.

Call:
HoltWinters(x = temps_time_series, seasonal = "multiplicative")

Smoothing parameters:
 alpha: 0.6219732
 beta : 0
 gamma: 0.5521032

Coefficients:
             [,1]
a    69.063940901
b    -0.004362918
```

Additive seasonality gives us an SSE of 63025.97, and multiplicative seasonality gives us 65648.65, which tells us that additive seasonality has slightly less variation and performs a little better.





Plotting the triple exponential smoothing models, we can see there's a straight line, implying that there is no trend, which is supported by the smoothing parameters and beta being close to 0. In other words, there's no specific evidence that shows a significant change, so we cannot conclude that the unofficial end of summer has gotten later over the past 20 years.

Overall, the single exponential smoothing model gives us the least variation and randomness, as its α is closer to 1.

Question 8.1

A common circumstance where a linear regression model would be appropriate is optimizing pricing for hotel rooms to maximize revenue. Various factors affect the prices at which rooms can be sold successfully, as consumer behavior can play a significant role.
Some predictors I would use:

- Time of year: different seasons have different numbers of room bookings
- Day of the week: weekend booking frequencies often vary greatly from weekday frequencies
- Special events: occasional entertainment events like sports or concerts can spur hotel demand during that time period
- Room features: room prices can be affected by room size/offerings, amenities, etc.
- Competitors: nearby competitors can set more appealing prices to attract customers

Question 8.2

I conducted a linear model for the imported data and produced a summary that displayed the significance of each factor.

```
Call:
lm(formula = Crime ~ ., data = crime)

Residuals:
    Min      1Q  Median      3Q     Max
-395.74  -98.09   -6.69  112.99  512.67

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.984e+03  1.628e+03  -3.675 0.000893 ***
M            8.783e+01  4.171e+01   2.106 0.043443 *
So          -3.803e+00  1.488e+02  -0.026 0.979765
Ed           1.883e+02  6.209e+01   3.033 0.004861 **
Po1          1.928e+02  1.061e+02   1.817 0.078892 .
Po2         -1.094e+02  1.175e+02  -0.931 0.358830
LF          -6.638e+02  1.470e+03  -0.452 0.654654
M.F          1.741e+01  2.035e+01   0.855 0.398995
Pop         -7.330e-01  1.290e+00  -0.568 0.573845
NW           4.204e+00  6.481e+00   0.649 0.521279
U1          -5.827e+03  4.210e+03  -1.384 0.176238
U2           1.678e+02  8.234e+01   2.038 0.050161 .
Wealth       9.617e-02  1.037e-01   0.928 0.360754
Ineq         7.067e+01  2.272e+01   3.111 0.003983 **
Prob        -4.855e+03  2.272e+03  -2.137 0.040627 *
Time        -3.479e+00  7.165e+00  -0.486 0.630708
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 209.1 on 31 degrees of freedom
Multiple R-squared:  0.8031,    Adjusted R-squared:  0.7078
F-statistic: 8.429 on 15 and 31 DF,  p-value: 3.539e-07
```

Off the bat, we can see a few factors are more significant, which we will keep in mind for later.

With the given data in the homework, we will create a test set to compare against the linear model created.

```
59  comparison <- data.frame(M=14.0,So=0,Ed=10.0,Po1=12.0,Po2=15.5,LF=0.640,M.F=94.0,
60                           Pop=150,NW=1.1,U1=0.120,U2=3.6,Wealth=3200,Ineq=20.1,
61                           Prob=0.04,Time=39.0)
```

To compare the two data sets, we can run a prediction estimate against them.

```
> prediction
       1
155.4349
```

The prediction resulted in 155.4349, which cannot be accurate since our data’s lowest value is 342. It could be a result of overfitting, so we need to adjust the factors.

```
> range(crime$Crime)
[1]  342 1993
```

Earlier, we saw that there were many insignificant factors when we conducted the summary of the linear model, so now we can try to exclude such variables to better the estimate.

```
71  lm_model2 <- lm(Crime ~ M+Ed+Po1+U2+Ineq+Prob,data=crime)
```

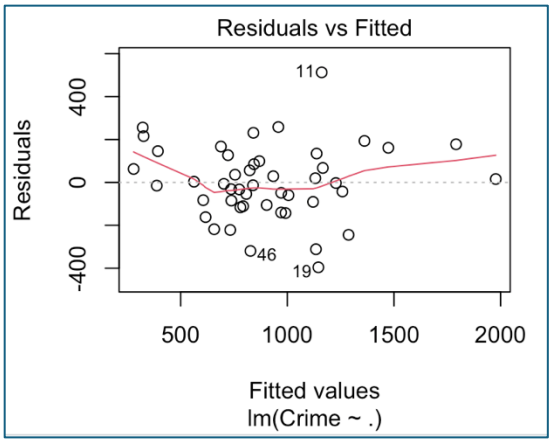
```
Call:
lm(formula = Crime ~ M + Ed + Po1 + U2 + Ineq + Prob, data = crime)

Residuals:
    Min      1Q  Median      3Q     Max
-470.68  -78.41  -19.68  133.12  556.23

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5040.50     899.84  -5.602 1.72e-06 ***
M             105.02      33.30   3.154  0.00305 **
Ed            196.47      44.75   4.390 8.07e-05 ***
Po1           115.02      13.75   8.363 2.56e-10 ***
U2             89.37      40.91   2.185  0.03483 *
Ineq           67.65      13.94   4.855 1.88e-05 ***
Prob        -3801.84    1528.10  -2.488  0.01711 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 200.7 on 40 degrees of freedom
Multiple R-squared:  0.7659,    Adjusted R-squared:  0.7307
F-statistic: 21.81 on 6 and 40 DF,  p-value: 3.418e-11
```

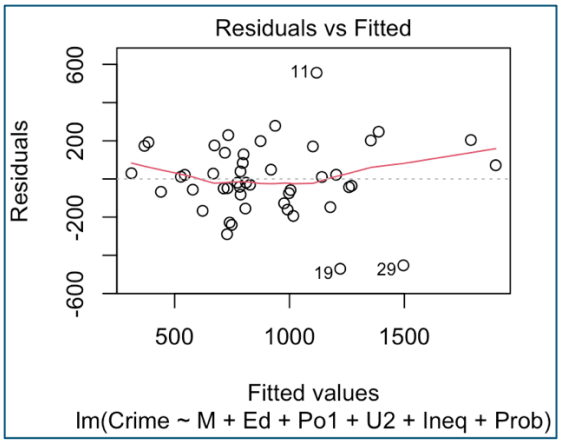
The summary of this new linear model shows that all factors utilized are significant, and there is also a higher adjusted R-squared value, indicating a stronger fit.
Now, we will test it to see if the prediction improves.

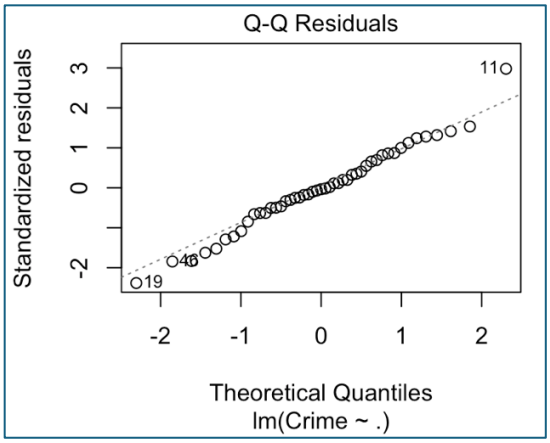
```
> prediction2
       1
1304.245
```

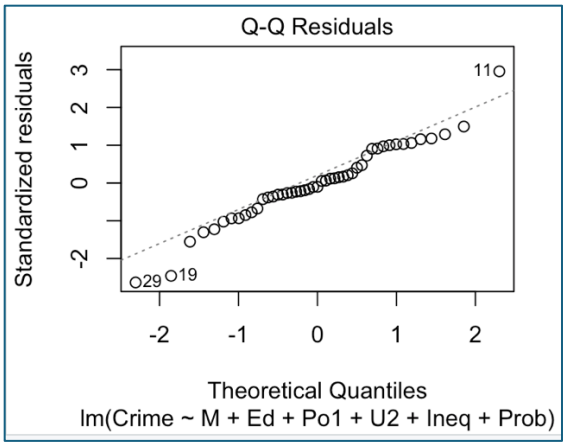
This prediction makes a lot more sense, considering that our data ranges from 342 to 1993. So, getting rid of non-significant components made our model more practical in this scenario, as backed up by the higher adjusted R-squared value.

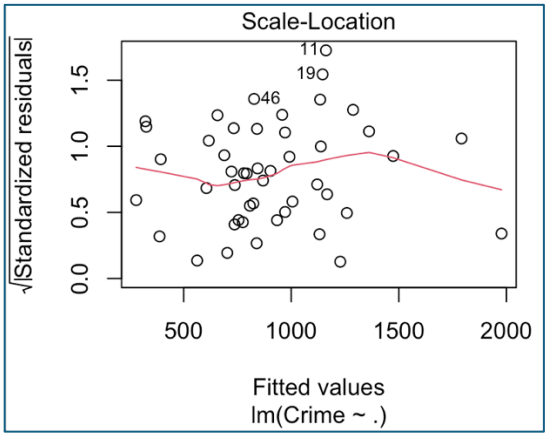
Here are some visualizations to show that the newer model fits the data better.

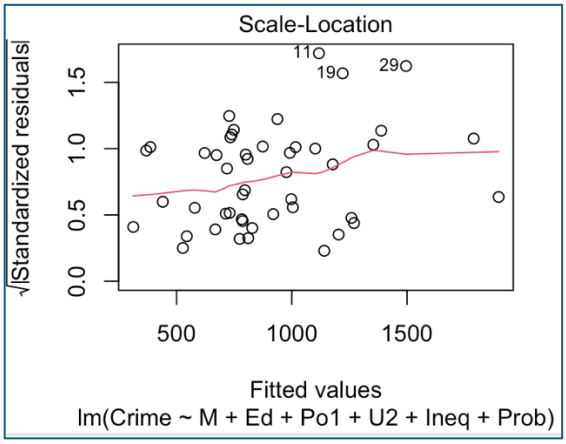


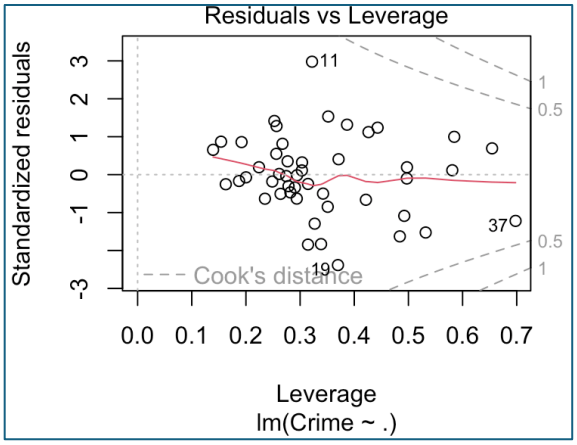


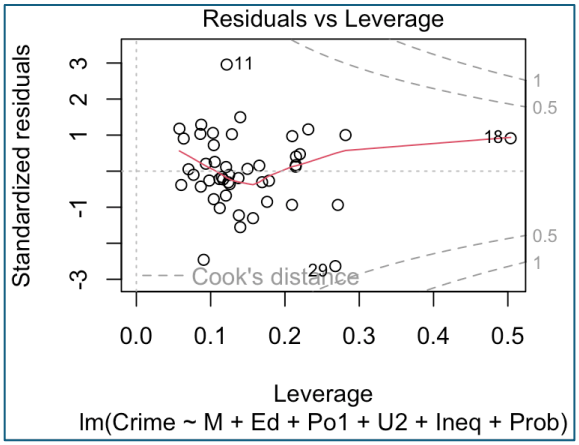












However, we still need to conduct cross-validation to ensure model accuracy.
I used recursive feature elimination (rfe), which removes the most insignificant factors until it reaches the set number we decide. I chose 10 features.

```
Recursive feature selection

Outer resampling method: Cross-Validated (10 fold, repeated 25 times)

Resampling performance over subset size:

 Variables  RMSE Rsquared   MAE RMSESD RsquaredSD  MAESD
         1 355.4   0.3294 287.3 132.22     0.2872 101.32
         2 341.3   0.3437 273.8 133.82     0.2865 107.36
         3 350.5   0.3064 283.4 131.54     0.2802 107.17
         4 328.3   0.4220 273.8  98.19     0.3299  85.35
         5 307.9   0.5265 253.7  97.26     0.3298  84.22
         6 299.5   0.5369 248.9  95.90     0.3228  82.97
         7 300.9   0.5342 249.4  92.38     0.3216  81.07
         8 271.9   0.5762 226.9  91.85     0.3034  75.80
         9 239.6   0.6516 195.4 101.53     0.2915  84.35
        10 231.2   0.6698 189.8  93.25     0.2737  76.43
        11 235.9   0.6466 192.6  97.53     0.2801  77.60
        12 245.8   0.6218 200.0  98.04     0.2923  80.75
        13 249.1   0.6136 202.5  98.47     0.2955  81.25
        14 255.8   0.6042 207.1 100.74     0.2960  83.89
        15 263.7   0.6034 214.8  99.24     0.2874  84.65
```

Based on the RMSE R-squared values, it seems like 10 features was a decent guess, as there are also higher R-squared values. These are the 10 features selected from the cross-validation.

```
> predictors(lm_model_rfe)
 [1] "U1"   "Prob" "LF"   "Po1"  "Ed"   "U2"   "Po2"  "M"
 [9] "Ineq" "So"
```

This is the final model with the 10 components selected.

```
Call:
lm(formula = y ~ ., data = tmp)

Coefficients:
(Intercept)           U1         Prob           LF
   -5099.78     -2925.15     -4000.57       531.84
        Po1           Ed           U2          Po2
     177.49       210.85       150.25       -79.51
          M         Ineq           So
     100.88        58.80        78.48
```

Once again, the prediction estimate resulted in roughly 870, which is reasonable within our dataset's range.

```
> prediction_cv
       1
870.6834
```

With cross-validation, we can conclude that there is a better model, as seen with a higher R-squared value. There could be better models, but our evaluation has shown that this is acceptable since it is reasonably inside our data minimum and maximum values.

```
# ISYE 6501
# Week 3 Homework

# libraries
library(caret)



# Q 7.2
# reproducibility
set.seed(42)

# load in data
temps <- read.table("~/Downloads/temps.txt", header=TRUE)
# vector of temps data
temps_v <- as.vector(unlist(temps[,2:21]))
# vector --> time series
temps_time_series <- ts(data=temps_v,start=1996,end=2015,frequency=123)

plot(temps_time_series)

# single exponential smoothing
single <- HoltWinters(temps_time_series,beta=FALSE,gamma=FALSE)
single
single$SSE  # 53704.15

# double exponential smoothing
double <- HoltWinters(temps_time_series,gamma=FALSE)
double
double$SSE  # 54071.22

# triple exponential smoothing
# additive seasonality
triple_A <- HoltWinters(temps_time_series,seasonal="additive")
triple_A
triple_A$SSE  # 63025.97
# multiplicative seasonality
triple_M <- HoltWinters(temps_time_series,seasonal="multiplicative")
triple_M
triple_M$SSE  # 65648.65

plot(fitted(triple_A))
plot(fitted(triple_M))



# Q 8.2
# reproducibility
set.seed(42)

# read in data
crime <- read.table("~/Downloads/uscrime.txt", header=TRUE)

# linear model
lm_model <- lm(Crime ~ .,data=crime)
summary(lm_model)

plot(lm_model)

# comparison set of given data
comparison <- data.frame(M=14.0,So=0,Ed=10.0,Po1=12.0,Po2=15.5,LF=0.640,M.F=94.0,
                         Pop=150,NW=1.1,U1=0.120,U2=3.6,Wealth=3200,Ineq=20.1,
                         Prob=0.04,Time=39.0)
```

```
# prediction
prediction <- predict(lm_model,comparison)
prediction

#range
range(crime$Crime)

# only using significant vars
lm_model2 <- lm(Crime ~ M+Ed+Po1+U2+Ineq+Prob,data=crime)
summary(lm_model2)

plot(lm_model2)

# prediction
prediction2 <- predict(lm_model2,comparison)
prediction2

# cross validation for model accuracy
control <- rfeControl(functions=lmFuncs,method="repeatedcv",number=10,
                      repeats=25,verbose=FALSE)
lm_model_rfe <- rfe(crime[,-16],crime[[16]],sizes=c(1:15),rfeControl=control)
lm_model_rfe

# best vars
predictors(lm_model_rfe)

# model with those vars
lm_model_rfe$fit

# prediction
prediction_cv <- predict(lm_model_rfe,comparison)
prediction_cv
```