hw_week3

Ben Wilson

May 31, 2019

Week 3 Homework

Time-Series and Basic Regression

Setup

The packages included to run this notebook are:

- tidyverse
- forecast

7.1

Describe a situation or problem from your job, everyday life, current events, etc., for which exponential smoothing would be appropriate. What data would you need? Would you expect the value of α (the first smoothing parameter) to be closer to 0 or 1, and why?

I work in a SaaS company and we stay alive by making sure our customers are deriving value from the product. A good way to know if they are getting value is looking at the usage data of our customers Even something as simple as did a user login today is a good indicator of if the customer is getting value. So, how many active users of their licensed users logged into our product on a given day is a valuable metric for the company. If we notice their active user usage dropping off, we can reach out to see if the product is no longer meeting their needs, and if not, why!?

Setting alpha closer to 0 means the system has a lot of randomness, so you only want to rely more on the historic data.
Conversly, setting alpha closer to 1 means the system is pretty stable, so you can act on the latest information.

Monitoring a login for a particular user is very reliable, so the alpha would be set closer to 1. We can trust the latest data.

However, we definitely need a cyclical component to the time series model. We make business software, so there is a big drop off in usage on the weekends. SHOCKER!!
We also work in a ternd component. Q4 is a slow down overall for business as people travel to spend time with their famalies and enjoy the holidays.

7.2

Using the 20 years of daily high temperature data for Atlanta, build and use an exponential smoothing model to help make a judgment of whether the unofficial end of summer has gotten later over the 20 years.
(Part of the point of this assignment is for you to think about how you might use exponential smoothing to answer this question. Feel free to combine it with other models if you'd like to. There's certainly more than one reasonable approach.)

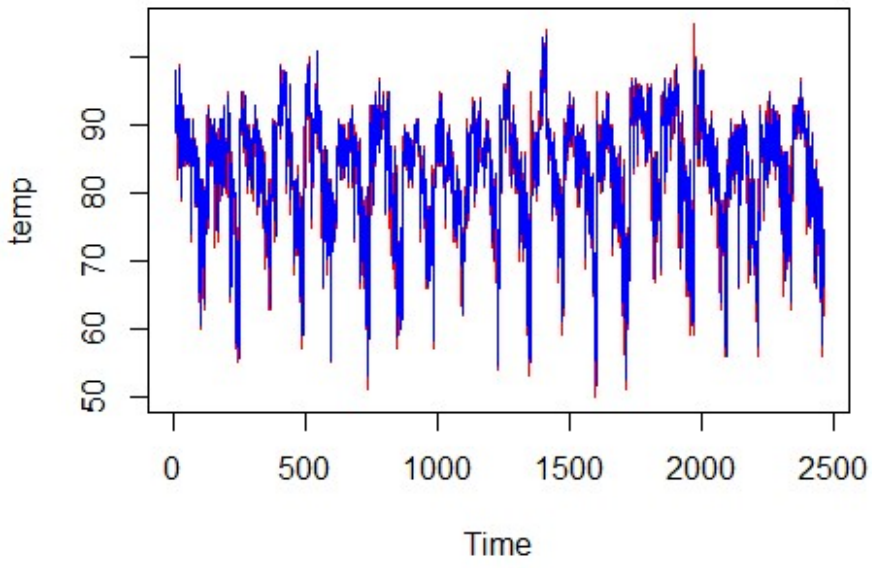
First let's format the data and plot out the data.
Since we know we are using an exponential smoothing model, we can train that model quickly too.

```
temps <- read.delim("data/temps.txt")

temps_ts <- temps %>%
  gather(key = "year", value = "temp", 2:21) %>%
  select( temp ) %>%
  as.ts()

# By default this Arima function uses the Box-Cox transformation to
set a lambda.
# I set it manually for completness of the assignment and,
# I set it towards 1 because yesterday's high is typically close to
today's high.
temps_exp <- Arima(temps_ts, order = c(0, 1, 1), lambda = 0.7)

# Not the prettiest plot, but we can see a bit of red poking through.
# The red is actuals and blue is the fitted exponential smoothing
model.
plot(temps_exp$x, col = "red")
lines(temps_exp$fitted, col = "blue")
```



Not the prettiest plot, but we can see a bit of red poking through. So we know the exponential smoothing is working.

It is a bit hard to determine any trends from looking at the full data set all in one timeline though. Let's break it down into each year.

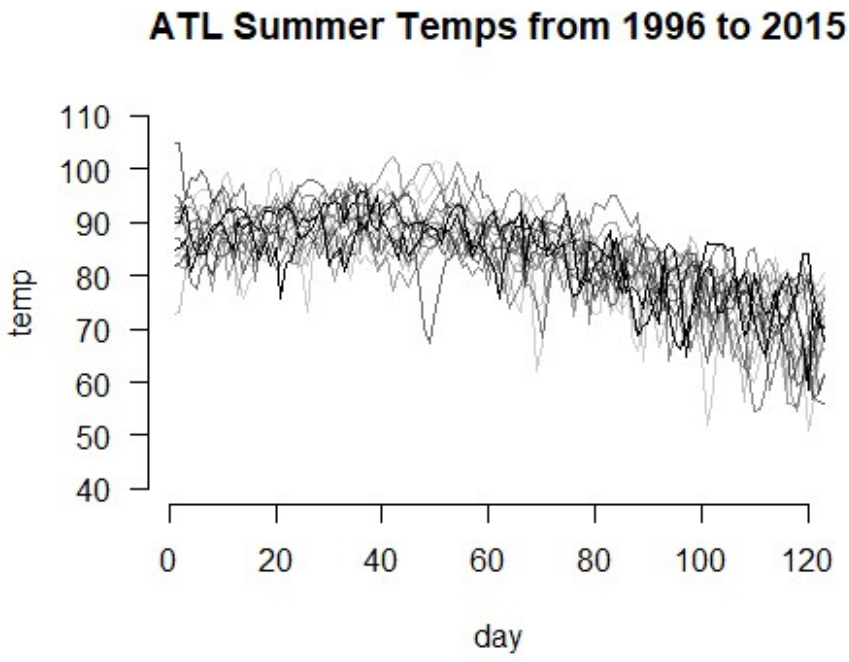
```
plot_yearly_exponential_smoothing <- function(year, transparency) {
  year_exp <- year %>%
    as.ts() %>%
    Arima(order = c(0, 1, 1), lambda = 0.7)

  if (transparency == 100) {
    transparency = ""
  }

  return(lines(year_exp$fitted, col=paste0("#000000", transparency)))
}
plot(0, 0, xlim=c(1,123), ylim=c(40, 110), type="n", las=1, xlab =
"day", ylab = "temp", bty="n", main = "ATL Summer Temps from 1996 to
2015")

for (i in 2:length(temps)) {
  transparency = 10 * as.integer(i/2)
  plot_yearly_exponential_smoothing(temps[i], transparency)
```

```
}
```



In this plot we increase the opacity as we get more recent data.

If we compare the different years, it looks as if the downward trends towards Fall starts around the same time, but more recent years have been sligthly warmer than they used to be.

8.1

Describe a situation or problem from your job, everyday life, current events, etc., for which a linear regression model would be appropriate. List some (up to 5) predictors that you might use.

I am currently house shopping and look at a lot of "Zestimates". I believe linear regression could be used to come up with a home price prediction. Useful features for the model could be:

- Zip Code
- Square Footage
- Lot Size

- Number of Bedrooms
- Number of Bathrooms

8.2

Using crime data from `uscrime.txt`, use regression (`lm` or `glm`) to predict the observed crime rate in a city with the following data:

Please refrence the HW file for sample predictor data

Show your model (factors used and their coefficients), the software output, and the quality of fit.

```
uscrime <- read.delim("data/uscrime.txt")
```

Let's start with a simple `lm` to see what the function does for us.

```
crime_lm1 <- lm(Crime ~ M, data = uscrime)
summary(crime_lm1)
```

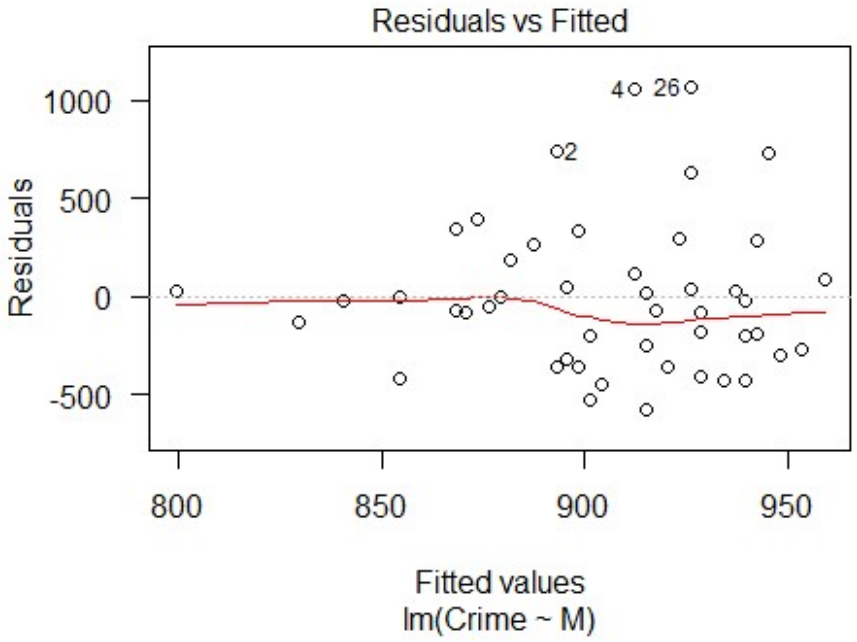
```
##
## Call:
## lm(formula = Crime ~ M, data = uscrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -572.93 -283.22  -50.38  153.97 1067.06
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1286.65     635.72   2.024   0.0489 *
## M             -27.53      45.69  -0.603   0.5498
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 389.5 on 45 degrees of freedom
## Multiple R-squared:  0.008005,   Adjusted R-squared:  -0.01404
## F-statistic: 0.3631 on 1 and 45 DF,  p-value: 0.5498
```

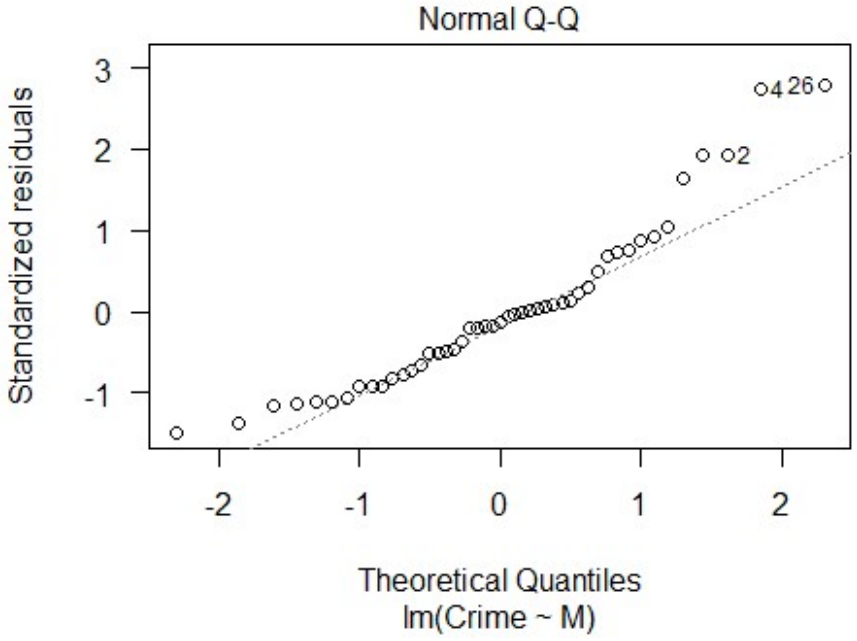
The `summary()` of the linear model gives us some good information talked about duirng the lesson.
In the coefficients table, the final column labeled `Pr(>|t|)` is the p-score. The software highlights important predictors. In this model the intercept is the most important factor… that's never a good sign.
If we continue on to look at the R-Squared, we see that this model does not even

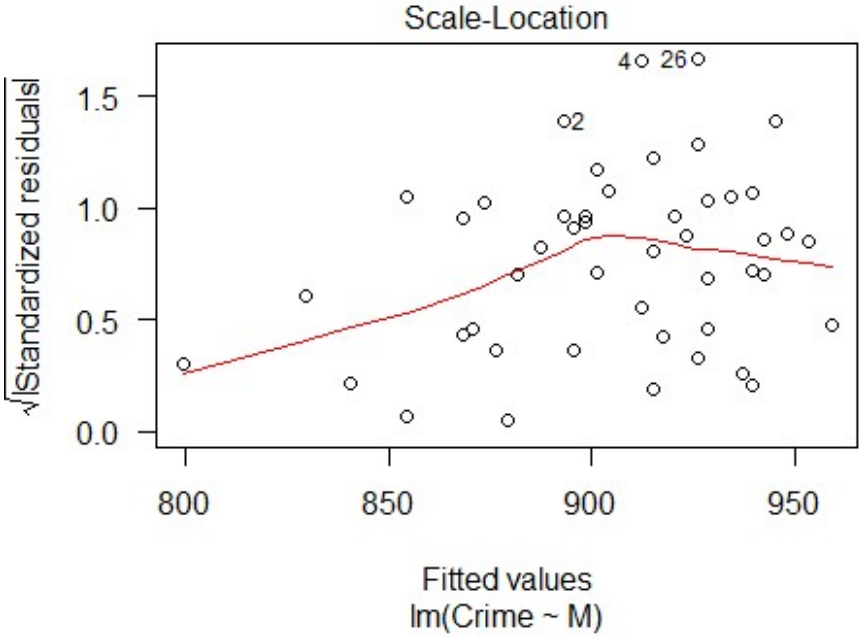
explain 1% of the variance in the model.

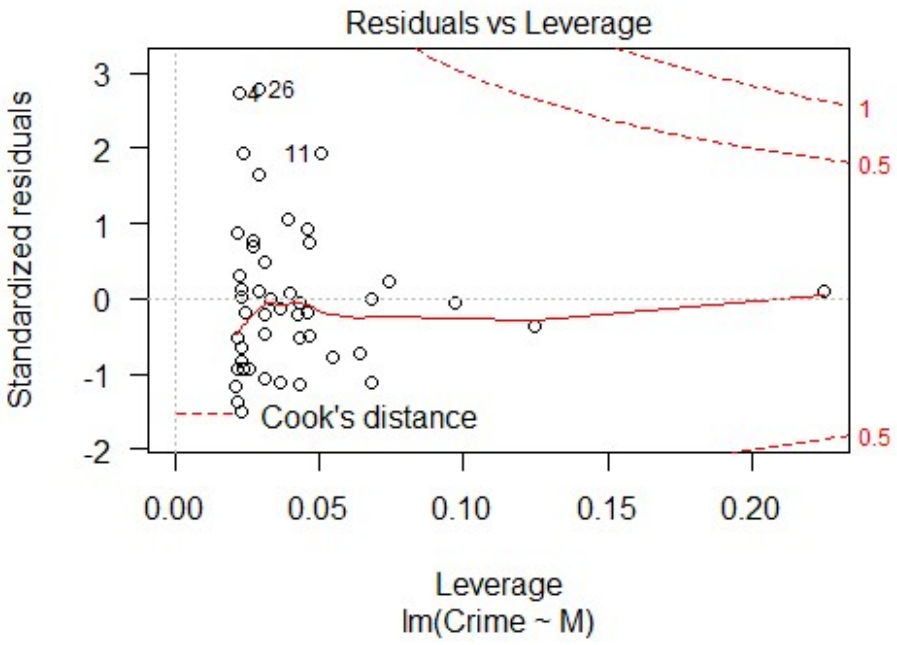
Still, plotting the data may give us a hint at why `M` is not a good predictor of crime.

```
plot(crime_lm1, las = 1)
```




Normal Q-Q
Standardized residuals
3
2
1
0
-1
4
26
2
-2
-1
0
1
2
Theoretical Quantiles
lm(Crime ~ M)


Scale-Location
√|Standardized residuals|
1.5
1.0
0.5
0.0
4
26
2
800
850
900
950
Fitted values
lm(Crime ~ M)



Plotting the model returns 4 different plots.

The first plot is similar to a scatter plot. Here the residuals - *the model's error* - are plotted against the prediction.

The second plot is a Q-Q plot. This type of plot looks at the distributions of two variables. If they are distributed the same then all the dots will be along the diagonal dotted line.

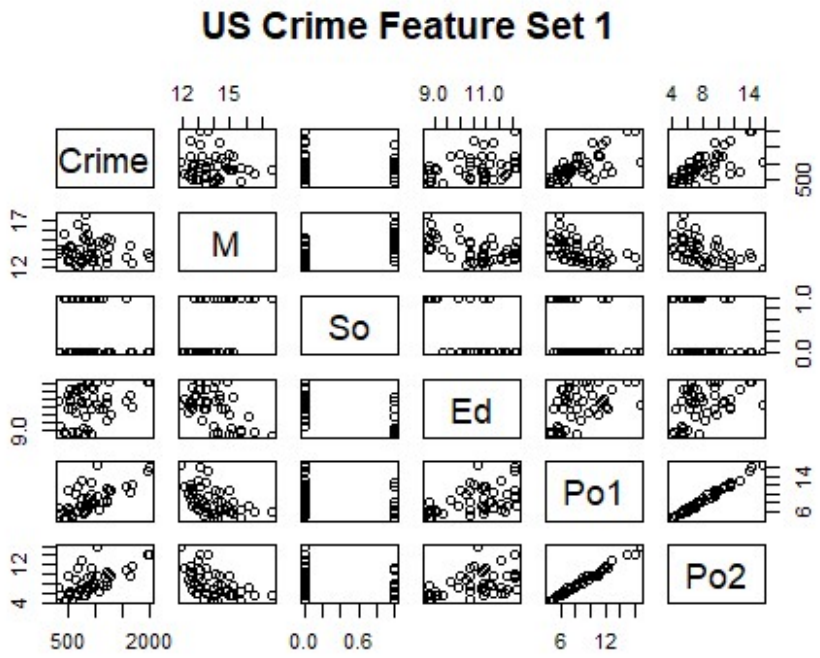
The third plot looks at the RMSE against the predictions. The closer to the red line, the better. So we have further evidence of this not being a great model.

The last plot is a leverage plot. I am still learning to inerpret this one. My general understanding is if a plotted residual is beyond the 0.5 and 1 thresholds is that these points are likely outliers that are having a strong impact on the model. For this model we do not have any points that the plot highlights as outliers, but there is a wide range of residuals.

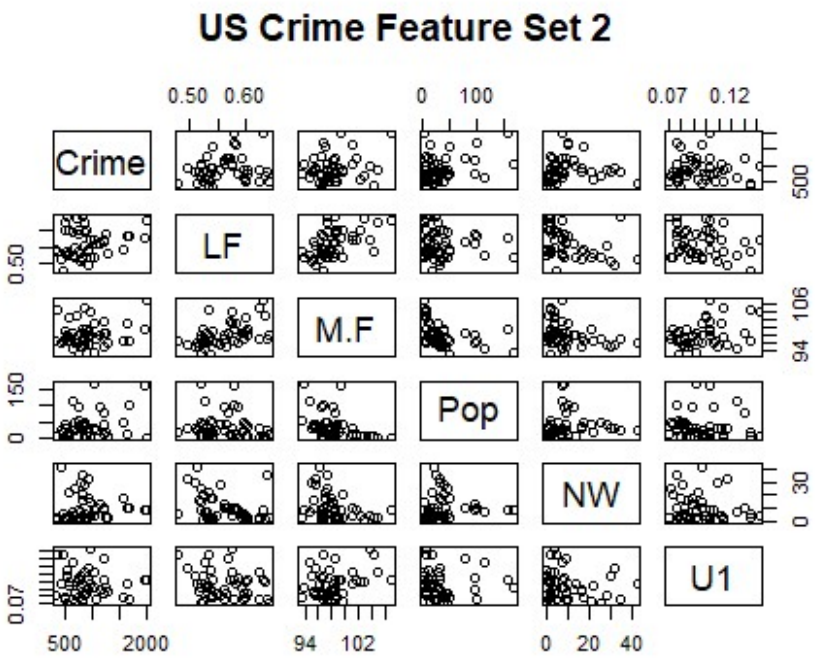
Since this model was not good, let's look at some scatter plots of features to see which ones might make for a good model.

```
pairs(
```

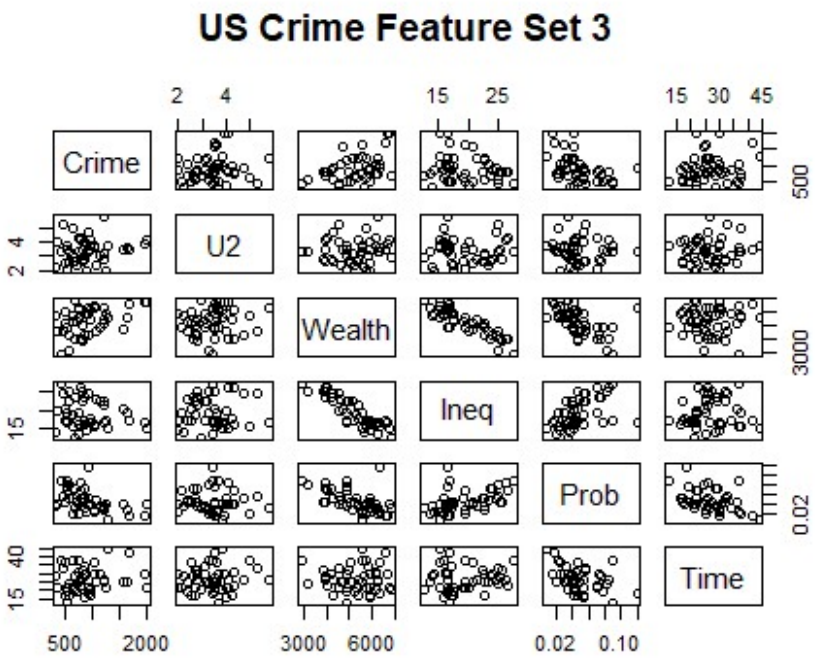
```
  ~Crime + M + So + Ed + Po1 + Po2,
  data=uscrime,
  main="US Crime Feature Set 1"
)
```




```
pairs(
  ~Crime + LF + M.F + Pop + NW + U1,
  data=uscrime,
  main="US Crime Feature Set 2"
)
```




```r
pairs(
  ~Crime + U2 + Wealth + Ineq + Prob + Time,
  data=uscrime,
  main="US Crime Feature Set 3"
)
```



In the scatter plots we are looking for a good diagonal relationship between Crime and a predictor. Glancing through the plots, it looks like there are a few predictors worth exploring with a model.

- Po1
- Po2
- M.F
- Wealth

Let's try a few combinations of these predictors to see how much we can improve the model.

```
# All candidate features
crime_lm2 <- lm(Crime ~ Po1 + Po2 + M.F + Wealth, data = uscrime)
summary(crime_lm2)
```

```
##
## Call:
## lm(formula = Crime ~ Po1 + Po2 + M.F + Wealth, data = uscrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -613.51 -183.63   -0.85  149.79  578.11
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.440e+03  1.334e+03  -1.829   0.0744 .
## Po1          2.233e+02  1.181e+02   1.891   0.0656 .
## Po2         -1.078e+02  1.280e+02  -0.843   0.4042
## M.F          3.064e+01  1.397e+01   2.193   0.0339 *
## Wealth      -1.331e-01  6.989e-02  -1.905   0.0637 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 267.5 on 42 degrees of freedom
## Multiple R-squared:  0.5634, Adjusted R-squared:  0.5218
## F-statistic: 13.55 on 4 and 42 DF,  p-value: 3.554e-07
```

```
# M.F and Wealth
crime_lm3 <- lm(Crime ~ M.F + Wealth, data = uscrime)
summary(crime_lm3)
```

```
##
## Call:
## lm(formula = Crime ~ M.F + Wealth, data = uscrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -594.62 -267.58  -61.88  210.48  797.53
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.767e+03  1.726e+03  -1.024  0.31159
## M.F          1.826e+01  1.784e+01   1.024  0.31154
## Wealth       1.669e-01  5.448e-02   3.063  0.00373 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 350.7 on 44 degrees of freedom
## Multiple R-squared:  0.2135, Adjusted R-squared:  0.1777
## F-statistic: 5.972 on 2 and 44 DF,  p-value: 0.005075
```

```
# Po1 and M.F
crime_lm4 <- lm(Crime ~ Po1 + M.F, data = uscrime)
summary(crime_lm4)
```

```
##
## Call:
## lm(formula = Crime ~ Po1 + M.F, data = uscrime)
##
## Residuals:
##    Min     1Q Median     3Q    Max
```

```
## -598.2 -186.0  -12.8  200.1  497.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2311.68    1365.10  -1.693   0.0974 .
## Po1            88.65      13.75   6.446 7.45e-08 ***
## M.F            25.06      13.87   1.807   0.0777 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 277 on 44 degrees of freedom
## Multiple R-squared:  0.5092, Adjusted R-squared:  0.4869
## F-statistic: 22.83 on 2 and 44 DF,  p-value: 1.584e-07
```

```
# Po1 and wealth
crime_lm5 <- lm(Crime ~ Po1 + Wealth, data = uscrime)
summary(crime_lm5)
```

```
##
## Call:
## lm(formula = Crime ~ Po1 + Wealth, data = uscrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -687.46 -140.11    3.37  141.37  553.68
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 469.17785  247.51328   1.896   0.0646 .
## Po1         116.42016   22.51731   5.170 5.48e-06 ***
## Wealth       -0.10538    0.06935  -1.520   0.1358
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 279.9 on 44 degrees of freedom
## Multiple R-squared:  0.4991, Adjusted R-squared:  0.4763
## F-statistic: 21.92 on 2 and 44 DF,  p-value: 2.482e-07
```

```
# Po1, M.F, and Wealth
crime_lm6 <- lm(Crime ~ Po1 + M.F + Wealth, data = uscrime)
summary(crime_lm6)
```

```
##
## Call:
## lm(formula = Crime ~ Po1 + M.F + Wealth, data = uscrime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -581.83 -169.25   17.68  158.53  563.19
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.578e+03  1.319e+03  -1.954   0.0572 .
## Po1          1.255e+02  2.179e+01   5.759 8.18e-07 ***
## M.F          3.234e+01  1.378e+01   2.347   0.0236 *
## Wealth      -1.451e-01  6.819e-02  -2.128   0.0391 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 266.6 on 43 degrees of freedom
## Multiple R-squared:  0.556,  Adjusted R-squared:  0.525
## F-statistic: 17.95 on 3 and 43 DF,  p-value: 1.06e-07
```

Looking at the summary statistics from all the trained models, I am going to select model 6 as the winner. Even though it did not achieve the best R-Squared stat of the models, it is simpler (only 3 predictors), and each predictor is significant.

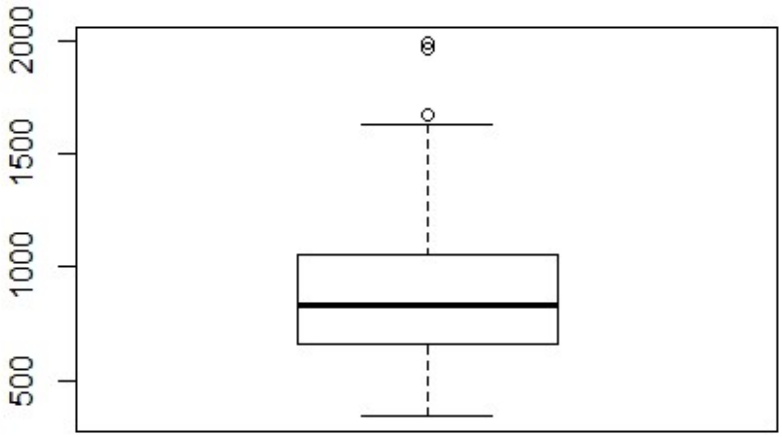
The final step of the assignment was to make a prediction based on a fictional city. Let's take a look!

```
# Data points for new city pulled from HW guidelines
new_city <- data.frame(Po1 = 12.0, M.F = 94.0, Wealth = 3200)
new_city_crime <- predict(crime_lm6, new_city)
print(new_city_crime)
```

```
##        1
## 1503.259
```

Our hypothetical city is predicted to have 1503 crimes. This puts the city well into the top quartile when you compare it to the box plot of our crime data.

```
boxplot(uscrime$Crime)
```


2000
1500
1000
500