homework3_isye6501

Zach Olivier

5/31/2018

Question 7.1

Question:

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?

Answer:

Any problem trying to forecast response data over time would be appropriate for an exponential smoothing model. In my work experience, I have tried to develop a time series based forecast on retail sales by day based on the current trend and seasonality. Exponential smoothing could have been used to deliver these forecasts. All the data we need would be our response retail sales by day for a long time horizon, preferably back more than two years.

The tuning of the alpha parameter would depend on what the goal of the forecasts are. If providing a estimate and the baseline - de-seasonalized forecast - I would choose a value closer to 1 to smooth out the fit as much as possible. Other applications, like using the forecasts to generate actions in other systems, may need a less smooth forecast to account for the typical fluctuations.

Question 7.2

Question:

Using the 20 years of daily high temperature data for Atlanta (July through October) from Question 6.2 (file temps.txt), 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.)

Answer:

I will apply the Holt Winter exponential smoothing algorithm to analyze if our temperature data has any overall trends in the end of summer temperature in the last 20 years.

To do this first first read in the data and format it as a time series object to feed the Holt Winters algorithm. I also quickly look at the temperatures by year to see if I can visibly inspect any trends. By looking at the plot of temperature over time - nothing stands out as a trend, randomness can fluctuation seem like a constant pattern.

The base implementation of Holt Winters in the stats packages automatically fits a triple exponential smoothing model, with overall smoothing, trend smoothing, and seasonality smoothing parameters (Alpha, Beta, Gamma respectively).

We can fit the Holt Winters model and examine the Beta parameter to determine if any trending exists in our data.

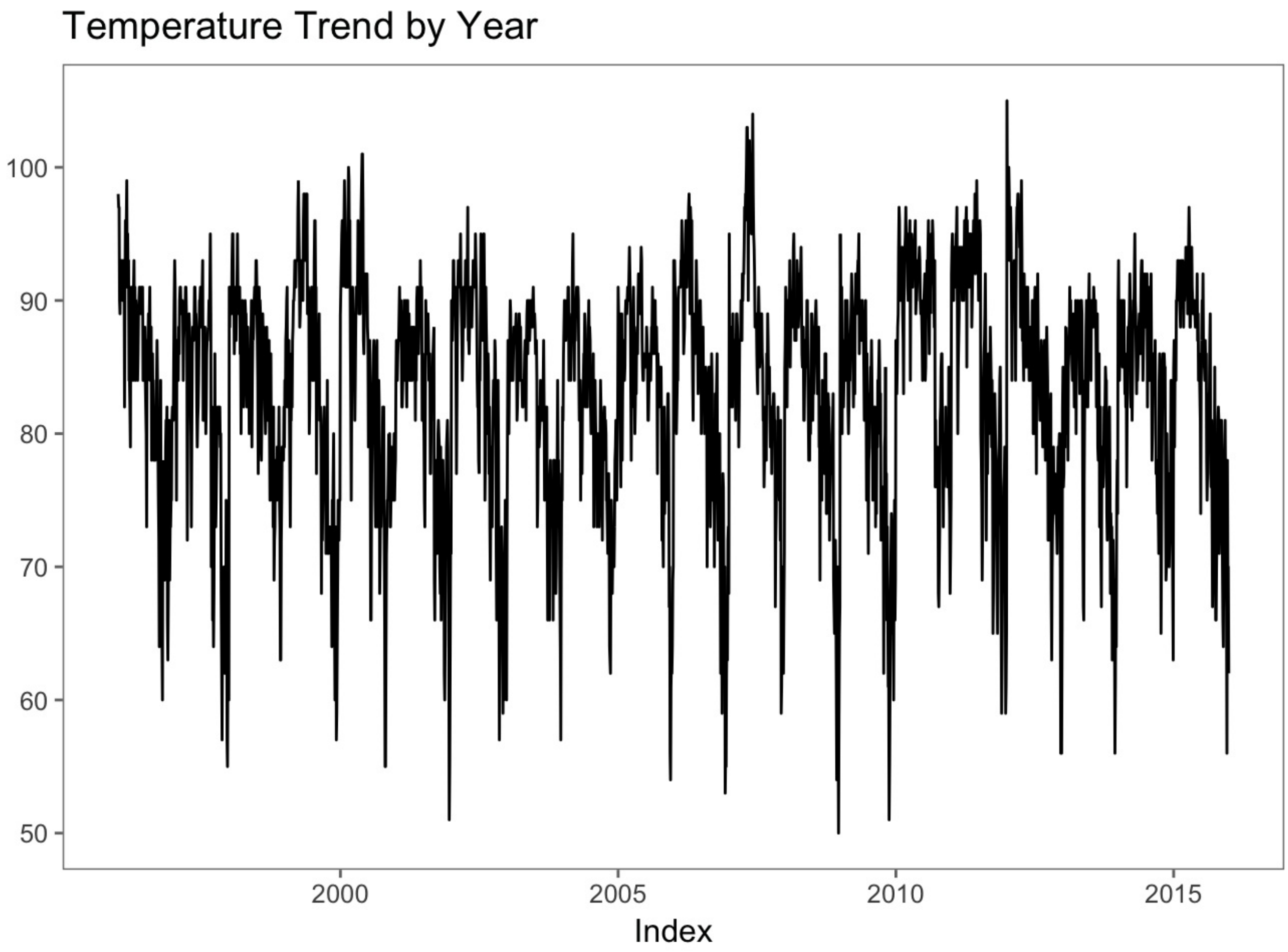
Upon fitting the data we see a Beta parameter of 0 and can conclude our temperature data does not have any trending in the last 20 years.

This matches our intuition from the initial plot of the temperature data - and gives us confidence that the Holt Winters model is picking up on the patterns (or lack of patterns) we can visibly see in the data.

```
library(zoo)

 # setup summer data as a time series object to feed into holt winters expo smoothi
ng
temps_df <- read_delim('7.2tempsSummer2018.txt', delim = '\t') %>%
        dplyr::select(., -DAY) %>%
        unlist() %>%
        as.vector() %>%
        ts(start = 1996, frequency = 123)

# quickly explore data
autoplot(as.zoo(temps_df), geom = 'line') + theme_few() +
        labs(title = 'Temperature Trend by Year') +
        ylab('')
```



```
# triple exponential smoothing w/ holt winters
(holt_fit = HoltWinters(temps_df))
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = temps_df)
##
## Smoothing parameters:
##  alpha: 0.6610618
##  beta : 0
##  gamma: 0.6248076
##
## Coefficients:
##               [,1]
## a     71.477236414
## b     -0.004362918
## s1    18.590169842
## s2    17.803098732
## s3    12.204442890
## s4    13.233948865
## s5    12.957258705
## s6    11.525341233
## s7    10.854441534
## s8    10.199632666
## s9     8.694767348
## s10    5.983076192
## s11    3.123493477
## s12    4.698228193
## s13    2.730023168
## s14    2.995935818
## s15    1.714600919
## s16    2.486701224
## s17    6.382595268
## s18    5.081837636
## s19    7.571432660
## s20    6.165047647
## s21    9.560458487
## s22    9.700133847
## s23    8.808383245
## s24    8.505505527
## s25    7.406809208
## s26    6.839204571
## s27    6.368261304
## s28    6.382080380
## s29    4.552058253
## s30    6.877476437
## s31    4.823330209
## s32    4.931885957
## s33    7.109879628
## s34    6.178469084
## s35    4.886891317
## s36    3.890547248
## s37    2.148316257
## s38    2.524866001
## s39    3.008098232
## s40    3.041663870
## s41    2.251741386
## s42    0.101091985
## s43   -0.123337548
## s44   -1.445675315
## s45   -1.802768181
## s46   -2.192036338
## s47   -0.180954242
## s48    1.538987281
## s49    5.075394760
## s50    6.740978049
## s51    7.737089782
## s52    8.579515859
## s53    8.408834158
## s54    4.704976718
## s55    1.827215229
## s56   -1.275747384
## s57    1.389899699
## s58    1.376842871
## s59    0.509553410
## s60    1.886439429
## s61   -0.806454923
## s62    5.221873550
## s63    5.383073482
## s64    4.265584552
## s65    3.841481452
## s66   -0.231239928
## s67    0.542761270
## s68    0.780131779
## s69    1.096690727
## s70    0.690525998
## s71    2.301303414
## s72    2.965913580
## s73    4.393732595
## s74    2.744547070
## s75    1.035278911
## s76    1.170709479
## s77    2.796838283
## s78    2.000312540
## s79    0.007337449
## s80   -1.203916069
## s81    0.352397232
## s82    0.675108103
## s83   -3.169643942
## s84   -1.913321175
## s85   -1.647780450
## s86   -5.281261301
## s87   -5.126493027
## s88   -2.637666754
## s89   -2.342133004
## s90   -3.281910970
## s91   -4.242033198
## s92   -2.596010530
## s93   -7.821281290
## s94   -8.814741200
## s95   -8.996689798
## s96   -7.835655534
## s97   -5.749139155
## s98   -5.196182693
## s99   -8.623793296
## s100 -11.809355220
## s101 -13.129428554
## s102 -16.095143067
## s103 -15.125436350
## s104 -13.963606549
## s105 -12.953304848
## s106 -16.097179844
## s107 -15.489223470
## s108 -13.680122300
## s109 -11.921434142
## s110 -12.035411347
## s111 -12.837047727
## s112  -9.095808127
## s113  -5.433029341
## s114  -6.800835107
## s115  -8.413639598
## s116 -10.912409484
## s117 -13.553826535
## s118 -10.652543677
## s119 -12.627298331
## s120  -9.906981556
## s121 -12.668519900
## s122  -9.805502547
## s123  -7.775306633
```

```
# gather the different components of the time series model
holt_SSE = holt_fit$SSE
holt_overall = holt_fit$alpha
holt_trend = holt_fit$beta
holt_season = holt_fit$gamma

# output stats from holt winters nicely to screen
print(paste0('Holt Winters SSE: ', holt_SSE))
```

```
## [1] "Holt Winters SSE: 66244.2504058465"
```

```
print(paste0('Holt Winters Overall Smmothing (Alpha): ', holt_overall))
```

```
## [1] "Holt Winters Overall Smmothing (Alpha): 0.661061754684708"
```

```
print(paste0('Holt Winters Trend (Beta): ', holt_trend))
```

```
## [1] "Holt Winters Trend (Beta): 0"
```

```
print(paste0('Holt Winters Trend (Gamma): ', holt_season))
```

```
## [1] "Holt Winters Trend (Gamma): 0.624807621487671"
```

Question 8.1

Question:

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.

Answer:

Linear regression can be helpful for all types of problems. I particularly enjoy fitting simple linear models for different aspects of my work to learn more about the system by examining the coefficients and the residuals of a simple model.

A recent example involved trying to quantify the value of having a strong brand - what a prestigious and well-respected brand image does for sales and profitability. The initial project started with a 8-variable linear regression with a 'Brand' indicator as one of the variables. This allowed us for an easy interpretation of how each much each typical nameplate is 'worth' to profitability / or sales after controlling for all other variables in the model. All this insight came from a relatively simple implementation of linear regression.

This was a powerful project that helped sell our 'customer engagement' and 'brand value management' investments to the entire organization as extremely critical. Of course, regression can predict and describe, but the actions needed to get from a Tier 3 brand to Tier 1 brand will not come out of the model. However, model linear models could help us even understand what a Tier 1 brand does well, and we can model our programs to support these key brand actions.

Overall linear regression has been a powerful tool in my toolbox throughout my work career - fitting models to understand the business is quick and easy to interpret. I am an advocate of fitting simple models as a part of any exploratory data analysis phase in any problem you are trying to solve.

Question 8.2

Question:

Using crime data from http://www.statsci.org/data/general/uscrime.txt (file uscrime.txt, description at http://www.statsci.org/data/general/uscrime.html), use regression (a useful R function is lm or glm) to predict the observed crime rate in a city with the following data:

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

Show your model (factors used and their coefficients), the software output, and the quality of fit. Note that because there are only 47 data points and 15 predictors, you'll probably notice some over fitting. We'll see ways of dealing with this sort of problem later in the course.

Answer:

Fitting a linear model starts with loading the data and quickly exploring the data. I take a look at the density of the response and examine the correlation between predictors and correlation with predictors and the response. Typically we do not want to include predictors correlated with each other and include only the variables that explain a lot of the response variation.

After the exploratory data analysis I think I found a good set of predictors to use for the model.

```
    - M = percentage of males aged 14-24 in state population
    - Ed = means years of schooling of the population 25 years or over
    - Po1 = per capita expenditure on police protection in 1959
    - Pop = state population in 1960
    - NW = percentage of non-whites in the population
    - Wealth = median value of transferable assets or family income
    - Ineq = income inequality
    - Prob = probability of imprisionment
```

Before fitting the model we need to pre-process the data. I am going to take advantage of the caret package to scale, center, remove zero variance predictors and apply the Box Cox transformation method to the sample data. This will ensure all variables are of the same magnitude, there are no arbitrary predictors included in the model (variables with the same observation), and all variables are transformed to be closer to normal distribution. All of these should held stabilize our model’s fit.

The model summary gives and R^2 metric of 80%. This means that around 80% of the total variance in crime is explained by our model. This may seem like a good model - but as we learned training set performance is not always an indicator of model quality. To further gague the model quality we should perform some type of cross validation or at least test the model on an independent dataset.

The model summary also shows that 7 out of the 9 variables have significant coefficient estimates - this means that the true value of the coefficient is not zero - each of these significant variables has some effect on the response.

Here are the final coefficient estimates. Note that becuase I scaled, centered and transformed the data, these coefficents are not on the original scale of the original data. This makes interpreation a little more difficult but should give us more accurate estimates of the coefficients. To interpret these values we could apply the reversing transformation steps to get back to our original scale.

```
    - (Intercept)   905.08511
    - Ineq          322.59345
    - Prob         -108.01137
    - Po1           312.07599
    - M         83.20118
    - Wealth    145.49971
    - Pop          -93.00281
    - Ed            114.56050
    - NW            84.10634
```

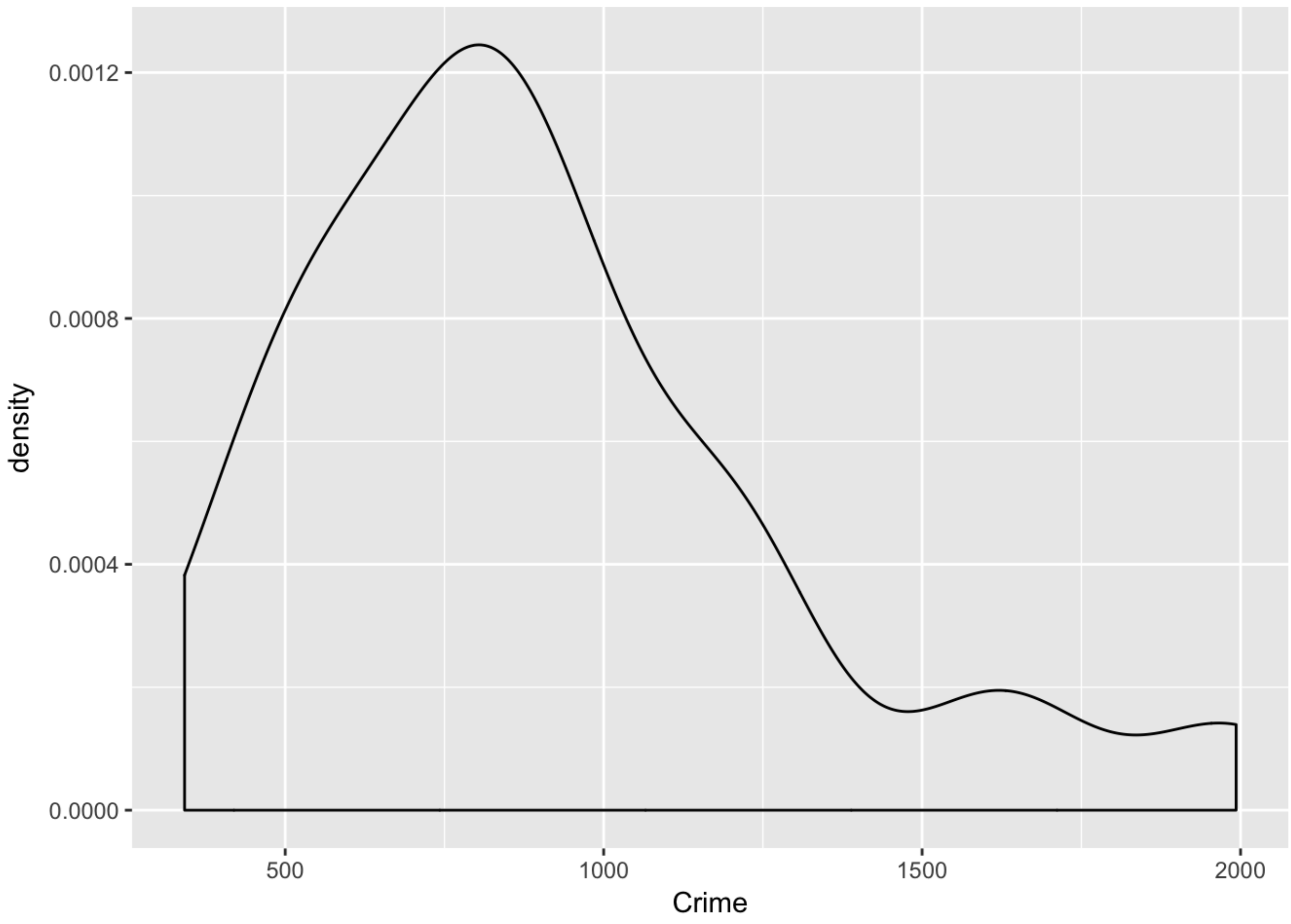
Finally I set up the new state date to use the model to predict a crime value for the given variables. To do this we need to apply the same centering, scaling and transformation steps to this new state observation. Once this is done we feed the relevant data to the model and obtain a prediction using the predict function. The crime estimate for the new state is:

```
    - "New State Crime Prediction:  791.838478897169"
```

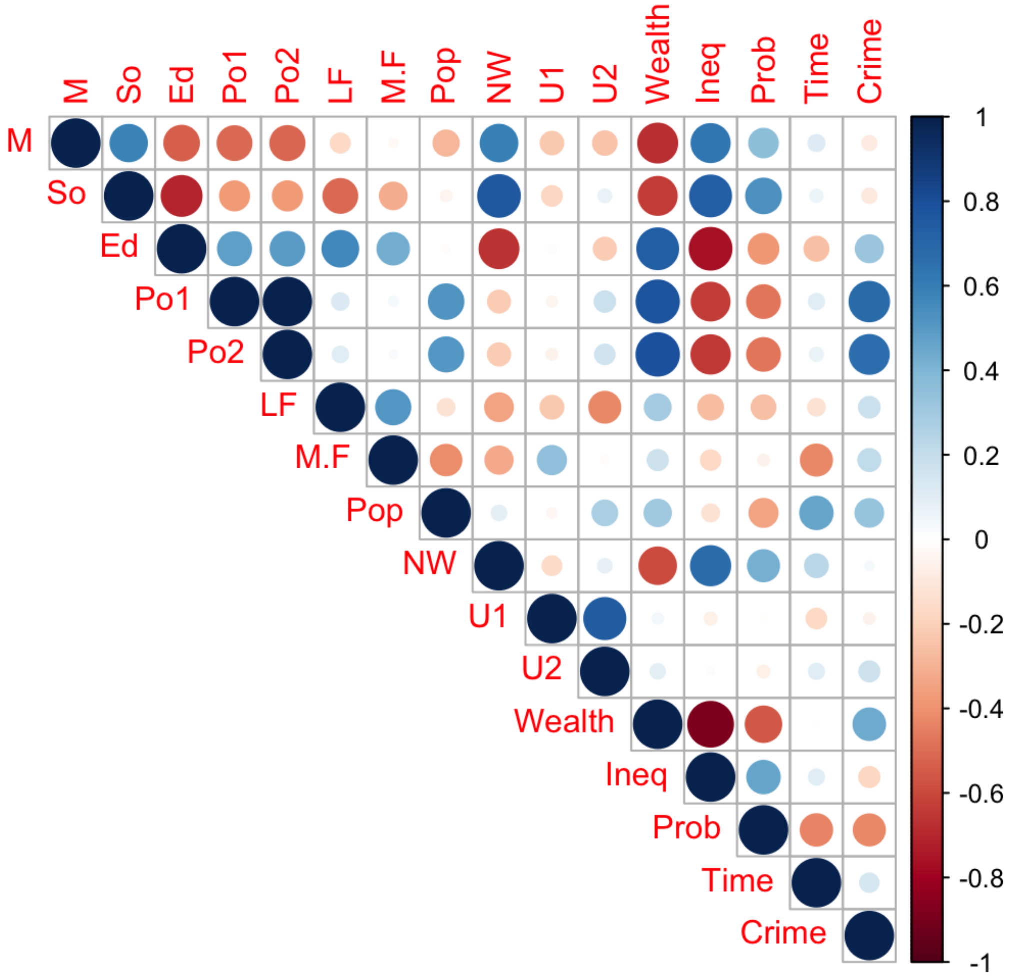
```
# read in the crime data
crime = read_delim('8.2uscrimeSummer2018.txt', delim = '\t') %>%
        as_tibble()
```

```
## Parsed with column specification:
## cols(
##   M = col_double(),
##   So = col_integer(),
##   Ed = col_double(),
##   Po1 = col_double(),
##   Po2 = col_double(),
##   LF = col_double(),
##   M.F = col_double(),
##   Pop = col_integer(),
##   NW = col_double(),
##   U1 = col_double(),
##   U2 = col_double(),
##   Wealth = col_integer(),
##   Ineq = col_double(),
##   Prob = col_double(),
##   Time = col_double(),
##   Crime = col_integer()
## )
```

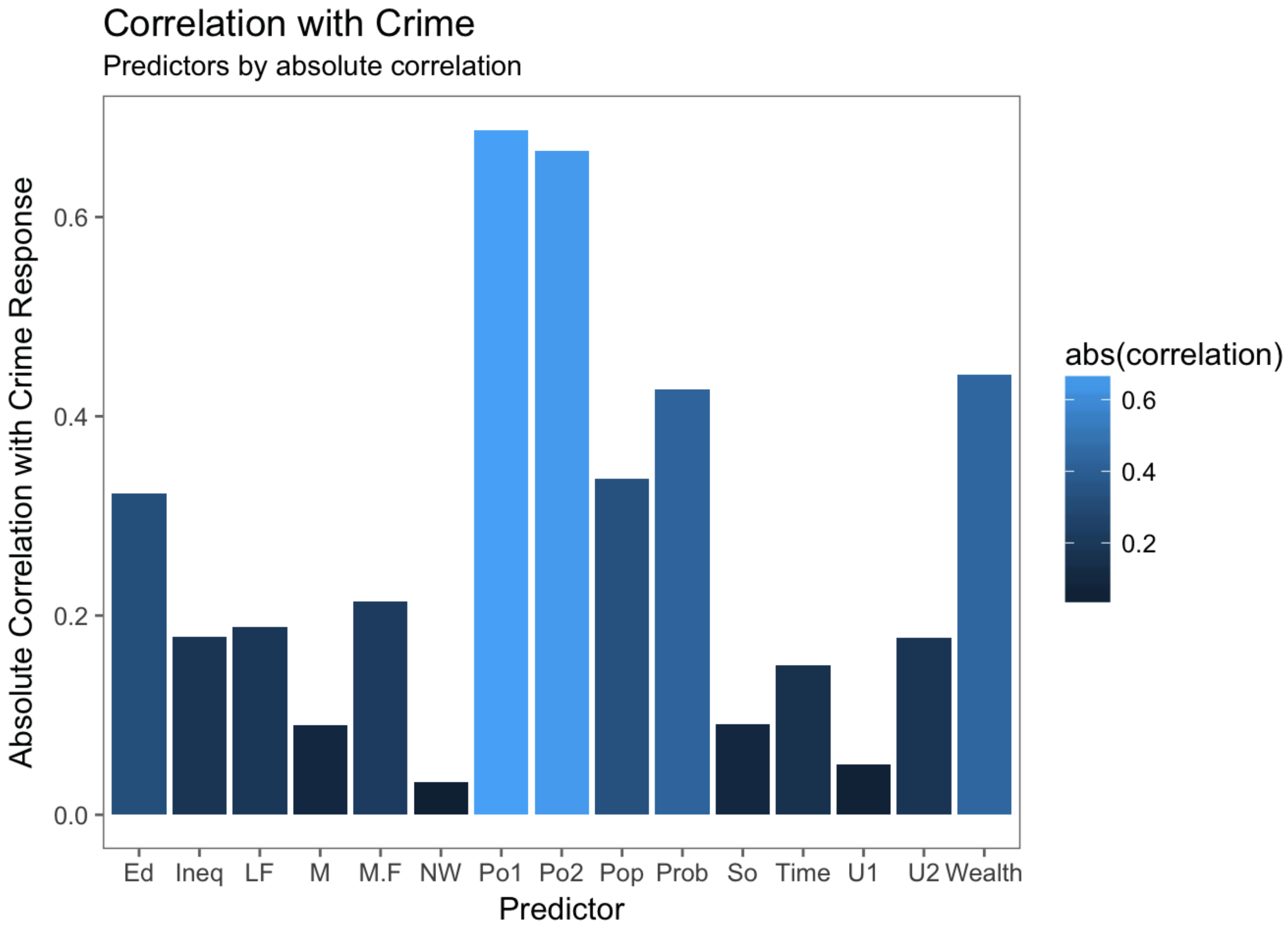
```
# quick eda - density of response
ggplot(data = crime, aes(x = Crime)) +
        geom_density()
```



```
# quick eda - correlation between predictors
corrplot(cor(crime), type = 'upper', diag = T)
```



```
# plot relationship between highest predictors
cor(crime$Crime, crime[,1:15]) %>%
        as_tibble() %>%
        gather(predictor, correlation) %>%
        arrange(-correlation) %>%
        ggplot(data = ., aes(x = predictor, y = abs(correlation), fill = abs(corre
lation))) +
        geom_col() +
        theme_few() +
        labs(title = 'Correlation with Crime',
             subtitle = 'Predictors by absolute correlation') +
        xlab('Predictor') +
        ylab('Absolute Correlation with Crime Response')
```



```
# select the variables with highest correlation for modeling - exclude Po2 due to covariance with Po1
crime_df = crime %>%
        dplyr::select(Crime, Ineq, Prob, Po1, M, Wealth,  Pop, Ed, NW)



# develop predictor data and response labels
crime_x = data.frame(crime_df) %>% mutate_all(as.numeric) %>% dplyr::select(., -Crime)
crime_y = data.frame(crime_df) %>% mutate_all(as.numeric) %>% dplyr::select(Crime)


# using cool caret functions to transform predictor data - including box cox transformation
transform_df = caret::preProcess(crime_x,  method = c('center', 'scale', 'nzv', 'BoxCox'))

# transformed df
crime_x_proc = predict(transform_df, crime_x) %>% cbind(crime_y = crime_y$Crime)


# fit linear model
crime_mod = lm(crime_y ~ . , data = crime_x_proc)

summary(crime_mod, cor = F)
```

```
##
## Call:
## lm(formula = crime_y ~ ., data = crime_x_proc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -359.12 -120.16    6.44   67.57  390.61
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      27.22  33.249  < 2e-16 ***
## Ineq          322.59      70.35   4.586 4.80e-05 ***
## Prob         -108.01      36.51  -2.958   0.0053 **
## Po1           312.08      63.52   4.913 1.74e-05 ***
## M              83.20      40.57   2.051   0.0472 *
## Wealth        145.50      87.07   1.671   0.1029
## Pop           -93.00      37.21  -2.500   0.0169 *
## Ed            114.56      47.65   2.404   0.0212 *
## NW             84.11      48.38   1.738   0.0902 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 186.6 on 38 degrees of freedom
## Multiple R-squared:  0.8077, Adjusted R-squared:  0.7672
## F-statistic: 19.95 on 8 and 38 DF,  p-value: 2.098e-11
```

```
coef(crime_mod) %>% as.data.frame()
```

```
##                     .
## (Intercept)  905.08511
## Ineq         322.59345
## Prob        -108.01137
## Po1          312.07599
## M             83.20118
## Wealth       145.49971
## Pop          -93.00281
## Ed           114.56050
## NW            84.10634
```

```
# predict new data
new_state = 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
)

# transform new_state date
new_state_transform = predict(transform_df, new_state) %>%
        dplyr::select(paste(names(crime_x)))

# crime prediction for new state based on model
crime_pred = predict(crime_mod, new_state_transform) %>%
        as_tibble()

print(paste('New State Crime Prediction: ', crime_pred$value))
```

```
## [1] "New State Crime Prediction:  791.838478897169"
```