W203: Statistics for Data Science

LAB 3: Reducing Crime

Robert Louka   Ryan Sawasaki   Joshua Noble   Praveen Joseph

```
## Loading required package: carData

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

##
## Please cite as:

##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
#read in the data
#data <- read.csv(file = 'H:/ROL/MIDS/W203 Stats/lab_3/crime_v2.csv') #Robert

data <- read.csv(file = '~/Desktop/W203/w203_lab3-master/crime_v2.csv') #Praveen
```

Ryan's section

1. Introduction

Your introduction should present a research question and explain the concept that you're attempting to measure and how it will be operationalized. This section should pave the way for the body of the report, preparing the reader to understand why the models are constructed the way that they are. It is not enough to simply say "We are looking for determinants of crime." Your introduction must do work for you, focusing the reader on a specific measurement goal, making them care about it, and propeling the narrative forward. This is also good time to put your work into context, discuss cross-cutting issues, and assess the overall appropriateness of the data.

Robert's section

2. A Model Building Process

Exploratory Data Analaysis

We started by conducting exploratory data analysis. First, we read the original paper to get a better understanding of each variable. We defined the variables in the table below and grouped them into five groups in order to get a better handle on them.

```r
crime_count <- c(1:25)
data_variables <- c("county","year","crmrte","prbarr","prbconv","prbpris","avgsen","polpc","density","ta
data_description <- c("county identifier","1987","crimes committed per person","'probability' of arrest
data_group <- c("Control","","","Deterrent","Deterrent","Deterrent","Deterrent","Deterrent","Demographi
data_notes <- c("","","ratio of FBI index crimes to county population","ratio of arrests to offenses","
)
data_headers <- c("Variable", "Description", "Group", "Note")
data_table <- data.frame(data_variables, data_description, data_group, data_notes)
kable(data_table, col.names = data_headers, caption = "Descriptions and Groups of Variables")
```

Table 1: Descriptions and Groups of Variables

Variable	Description	Group	Note
county	county identifier	Control	
year	1987		
crmrte	crimes committed per person		ratio of FBI index crimes to county population
prbarr	‘probability’ of arrest	Deterrent	ratio of arrests to offenses
prbconv	‘probability’ of conviction	Deterrent	ratio of convictions to arrests
prbpris	‘probability’ of prison sentence	Deterrent	proportion of total convictions resulting in prison sentences
avgsen	avg. sentence, days	Deterrent	average sentence in days
polpc	police per capita	Deterrent	
density	people per sq. mile	Demographic	country population divided by county land area
taxpc	tax revenue per capita	Demographic	
west	=1 if in western N.C.	Region	dummy
central	=1 if in central N.C.	Region	dummy
urban	=1 if in SMSA	Urban	dummy
pctmin80	perc. minority, 1980	Demographic	proportion of country population that is minority or nonwhite
wcon	weekly wage, construction	Wages	average weekly wage in that sector
wtuc	wkly wge, trns, util, commun	Wages	average weekly wage in that sector
wtrd	wkly wge, whlesle, retail trade	Wages	average weekly wage in that sector
wfir	wkly wge, fin, ins, real est	Wages	average weekly wage in that sector
wser	wkly wge, service industry	Wages	average weekly wage in that sector
wmfg	wkly wge, manufacturing	Wages	average weekly wage in that sector
wfed	wkly wge, fed employees	Wages	average weekly wage in that sector
wsta	wkly wge, state employees	Wages	average weekly wage in that sector
wloc	wkly wge, local gov emps	Wages	average weekly wage in that sector
mix	offense mix: face-to-face/other	Demographic	ratio of face-to-face crimes (robbery, assault, rape) to non-face
pctymle	percent young male	Demographic	proportion of country population that is male between 15 and

To get a better sense of the data set the summary function was run.

```
summary(data)
```

```
##      county           year        crmrte             prbarr
##  Min.   :  1.0   Min.   :87   Min.   :0.005533   Min.   :0.09277
##  1st Qu.: 52.0   1st Qu.:87   1st Qu.:0.020927   1st Qu.:0.20568
##  Median :105.0   Median :87   Median :0.029986   Median :0.27095
##  Mean   :101.6   Mean   :87   Mean   :0.033400   Mean   :0.29492
##  3rd Qu.:152.0   3rd Qu.:87   3rd Qu.:0.039642   3rd Qu.:0.34438
##  Max.   :197.0   Max.   :87   Max.   :0.098966   Max.   :1.09091
##  NA's   :6       NA's   :6    NA's   :6          NA's   :6
##         prbconv      prbpris           avgsen           polpc
##             : 5   Min.   :0.1500   Min.   : 5.380   Min.   :0.000746
##  0.588859022: 2   1st Qu.:0.3648   1st Qu.: 7.340   1st Qu.:0.001231
##  `          : 1   Median :0.4234   Median : 9.100   Median :0.001485
##  0.068376102: 1   Mean   :0.4108   Mean   : 9.647   Mean   :0.001702
##  0.140350997: 1   3rd Qu.:0.4568   3rd Qu.:11.420   3rd Qu.:0.001877
##  0.154451996: 1   Max.   :0.6000   Max.   :20.700   Max.   :0.009054
##  (Other)    :86   NA's   :6        NA's   :6        NA's   :6
##     density            taxpc             west            central
##  Min.   :0.00002   Min.   : 25.69   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.54741   1st Qu.: 30.66   1st Qu.:0.0000   1st Qu.:0.0000
```

```
##  Median :0.96226   Median : 34.87   Median :0.0000   Median :0.0000
##  Mean   :1.42884   Mean   : 38.06   Mean   :0.2527   Mean   :0.3736
##  3rd Qu.:1.56824   3rd Qu.: 40.95   3rd Qu.:0.5000   3rd Qu.:1.0000
##  Max.   :8.82765   Max.   :119.76   Max.   :1.0000   Max.   :1.0000
##  NA's   :6         NA's   :6        NA's   :6        NA's   :6
##      urban            pctmin80            wcon            wtuc
##  Min.   :0.00000   Min.   : 1.284   Min.   :193.6   Min.   :187.6
##  1st Qu.:0.00000   1st Qu.: 9.845   1st Qu.:250.8   1st Qu.:374.6
##  Median :0.00000   Median :24.312   Median :281.4   Median :406.5
##  Mean   :0.08791   Mean   :25.495   Mean   :285.4   Mean   :411.7
##  3rd Qu.:0.00000   3rd Qu.:38.142   3rd Qu.:314.8   3rd Qu.:443.4
##  Max.   :1.00000   Max.   :64.348   Max.   :436.8   Max.   :613.2
##  NA's   :6         NA's   :6        NA's   :6       NA's   :6
##       wtrd            wfir            wser             wmfg
##  Min.   :154.2   Min.   :170.9   Min.   : 133.0   Min.   :157.4
##  1st Qu.:190.9   1st Qu.:286.5   1st Qu.: 229.7   1st Qu.:288.9
##  Median :203.0   Median :317.3   Median : 253.2   Median :320.2
##  Mean   :211.6   Mean   :322.1   Mean   : 275.6   Mean   :335.6
##  3rd Qu.:225.1   3rd Qu.:345.4   3rd Qu.: 280.5   3rd Qu.:359.6
##  Max.   :354.7   Max.   :509.5   Max.   :2177.1   Max.   :646.9
##  NA's   :6       NA's   :6       NA's   :6        NA's   :6
##       wfed            wsta            wloc            mix
##  Min.   :326.1   Min.   :258.3   Min.   :239.2   Min.   :0.01961
##  1st Qu.:400.2   1st Qu.:329.3   1st Qu.:297.3   1st Qu.:0.08074
##  Median :449.8   Median :357.7   Median :308.1   Median :0.10186
##  Mean   :442.9   Mean   :357.5   Mean   :312.7   Mean   :0.12884
##  3rd Qu.:478.0   3rd Qu.:382.6   3rd Qu.:329.2   3rd Qu.:0.15175
##  Max.   :598.0   Max.   :499.6   Max.   :388.1   Max.   :0.46512
##  NA's   :6       NA's   :6       NA's   :6       NA's   :6
##     pctymle
##  Min.   :0.06216
##  1st Qu.:0.07443
##  Median :0.07771
##  Mean   :0.08396
##  3rd Qu.:0.08350
##  Max.   :0.24871
##  NA's   :6
```

This function provides a high level view of each variable. Six rows have missing values for all variables. In addition, there is one duplicate row. Also the variable prbconv is loaded as a factor, so it needs to be converted to numeric. These issues are handled below to create the initial data set.

```r
#eliminate N/A's (6 rows of NA were removed)
data_crmrte <- data[!is.na(data$crmrte),]

#remove duplicates (1 duplicate record was found)
data_crmrte <- data_crmrte %>% distinct()

#prbconv was defined as factor , we will convert it to numeric
data_crmrte$prbconv <- as.numeric(as.character(data_crmrte$prbconv))
class(data_crmrte$prbconv)
```

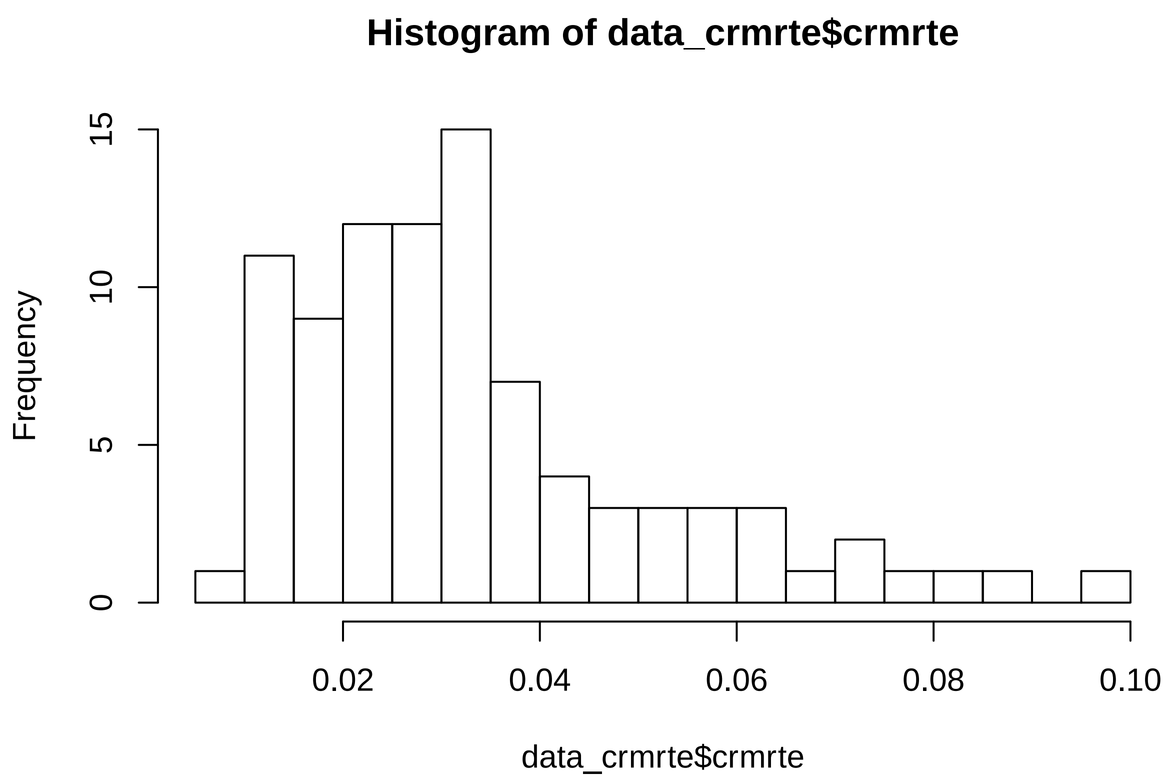
```
## [1] "numeric"
```

With 25 original variables in the data set the natural place to start is with the dependent variable, crmrte. To get a better sense of this variable, the distribution is graphed below.

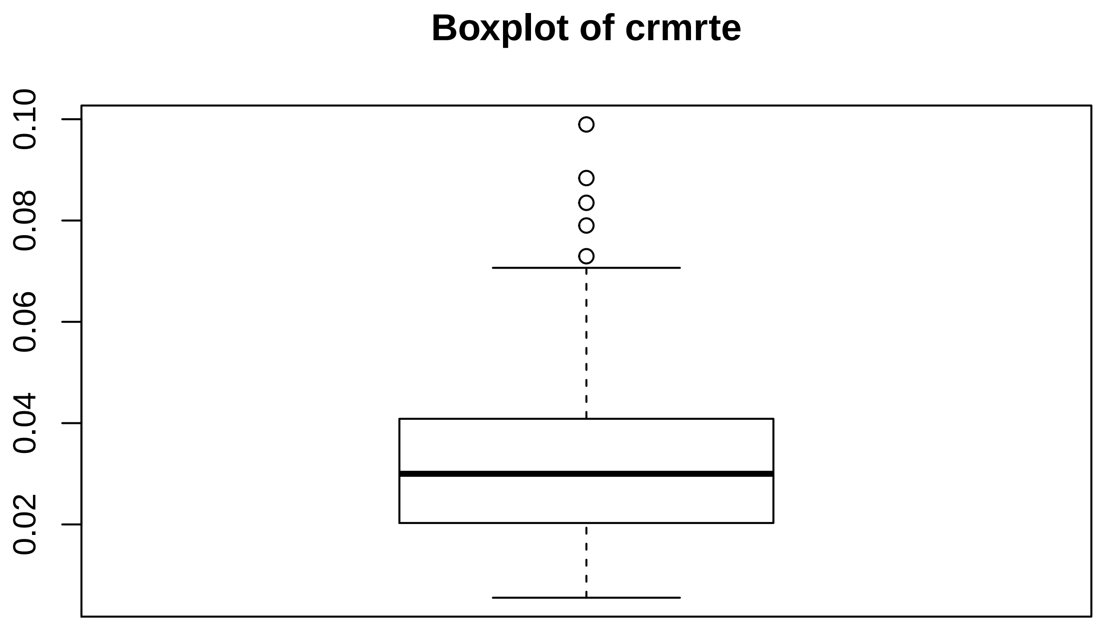
```r
quantile(data_crmrte$crmrte, c(0, .01, .05, .10, .25, .50,  .75, .90, .95, .99, 1.0))
```

```
##         0%         1%         5%        10%        25%        50%        75% 
## 0.00553320 0.01006330 0.01235660 0.01418007 0.02060425 0.03000200 0.04024925 
##        90%        95%        99%       100% 
## 0.06054659 0.07191830 0.08954881 0.09896590
```

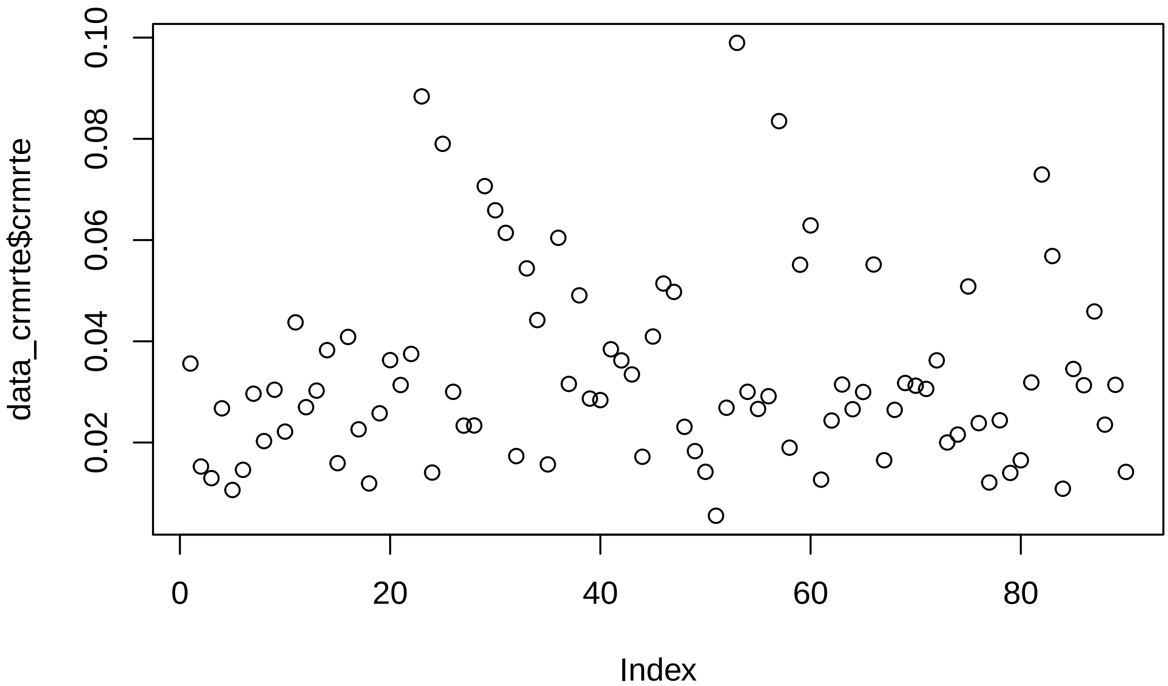
```r
hist(data_crmrte$crmrte,breaks=20)
```



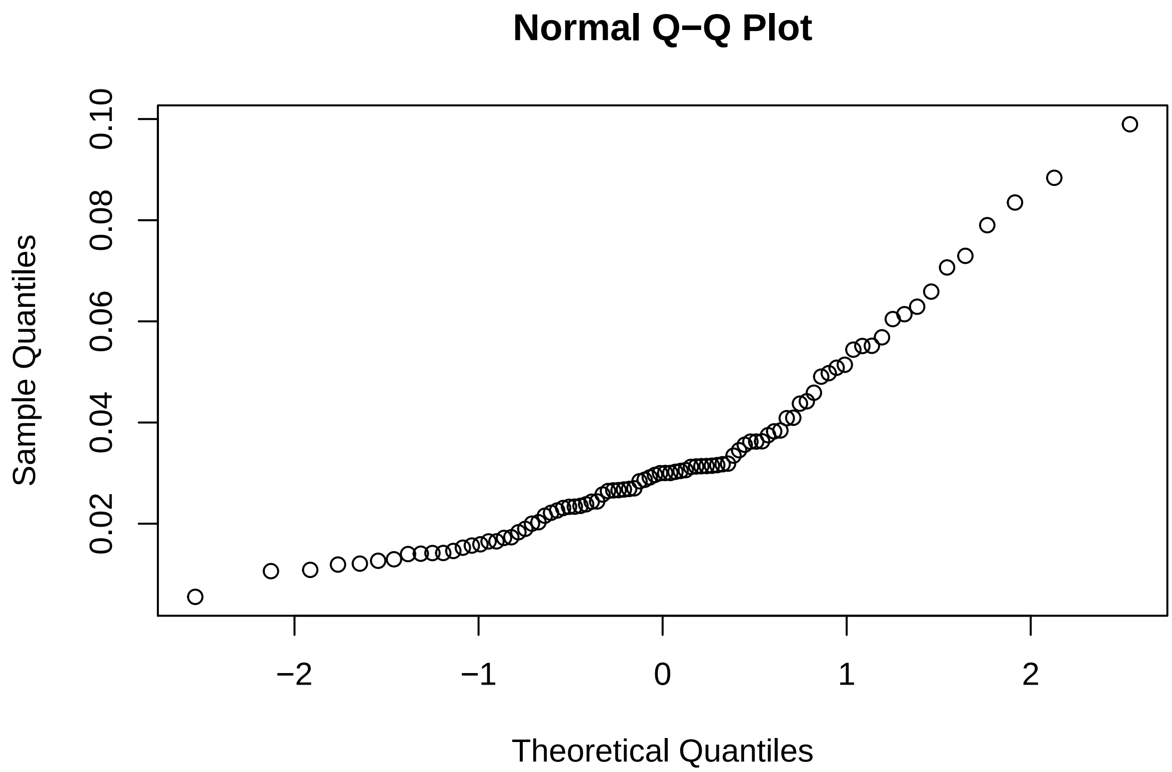

```r
boxplot(data_crmrte$crmrte, main="Boxplot of crmrte")
```




```r
plot(data_crmrte$crmrte)
```




```r
qqnorm(data_crmrte$crmrte)
```




```
shapiro.test(data_crmrte$crmrte) # Shapiro-wilk test confirms non-normality
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_crmrte$crmrte
## W = 0.89162, p-value = 1.741e-06
```

Outlier Analysis
There are several outliers in the variable crmrte and the distribution is right skewed. We have ninety observations so non-normality is not a top concern but this distribution is not perfectly normal. we analyse outliers for crime rate that are > 1.96*Std-dev from the mean crime rate (~0.07)

The largest outliers (6 counties) on the right side of the distribution are examined. There are some insights: - 1. 4 of out of the 6 outliers are in urban areas - 2. The average demographic density for the outlier set is greater than 3 times the average density for the sample - 3. We also observe that data ppt 53 which has the highest crime rate, also has the highest density amongst the outliers and is a urban area

This is not very surprising as we expect urban areas with high density of population to have more crimes.

we will continue to monitor the impact of the outliers and conisder the treatment of these outlier in later part of the report.

```
data_crmrte[data_crmrte$crmrte > 0.07,]
```

```
##    county year    crmrte    prbarr  prbconv  prbpris avgsen      polpc    density
```

```
## 23     51   87 0.0883849 0.155248 0.259833 0.407628  11.93 0.00190802 3.9345510
## 25     55   87 0.0790163 0.224628 0.207831 0.304348  13.57 0.00400962 0.5115089
## 29     63   87 0.0706599 0.133225 0.459216 0.363636  11.51 0.00237609 5.6744967
## 53    119   87 0.0989659 0.149094 0.347800 0.486183   7.13 0.00223135 8.8276520
## 57    129   87 0.0834982 0.236601 0.393413 0.415158   9.57 0.00255849 6.2864866
## 82    181   87 0.0729479 0.182590 0.343023 0.548023   7.06 0.00172948 1.5702811
##        taxpc west central urban pctmin80     wcon     wtuc     wtrd     wfir
## 23  35.69936    0       0     1 37.77920 283.6695 412.4720 213.7524 324.8357
## 25 119.76145    0       0     0  6.49622 309.5238 445.2762 189.7436 284.5933
## 29  50.19918    0       1     1 38.22300 349.3267 548.9865 238.9154 435.1107
## 53  75.67243    0       1     1 28.54600 436.7666 548.3239 354.6761 509.4655
## 57  67.67963    0       0     1 23.04410 315.5760 392.0999 220.4530 363.2880
## 82  27.59179    0       1     0 44.62830 244.8362 365.4716 279.2273 325.0271
##        wser   wmfg   wfed   wsta   wloc        mix    pctymle
## 23 257.3344 441.72 433.94 367.34 333.71 0.10474319 0.14223780
## 25 221.3903 319.21 338.91 361.68 326.08 0.08437271 0.07613807
## 29 391.3081 646.85 563.77 415.51 362.58 0.07585382 0.09468981
## 53 354.3007 494.30 568.40 329.22 379.77 0.16869897 0.07916495
## 57 292.7027 464.49 548.49 421.36 319.08 0.07871422 0.08109921
## 82 213.5822 290.69 453.53 317.23 286.45 0.10003893 0.07977433
```

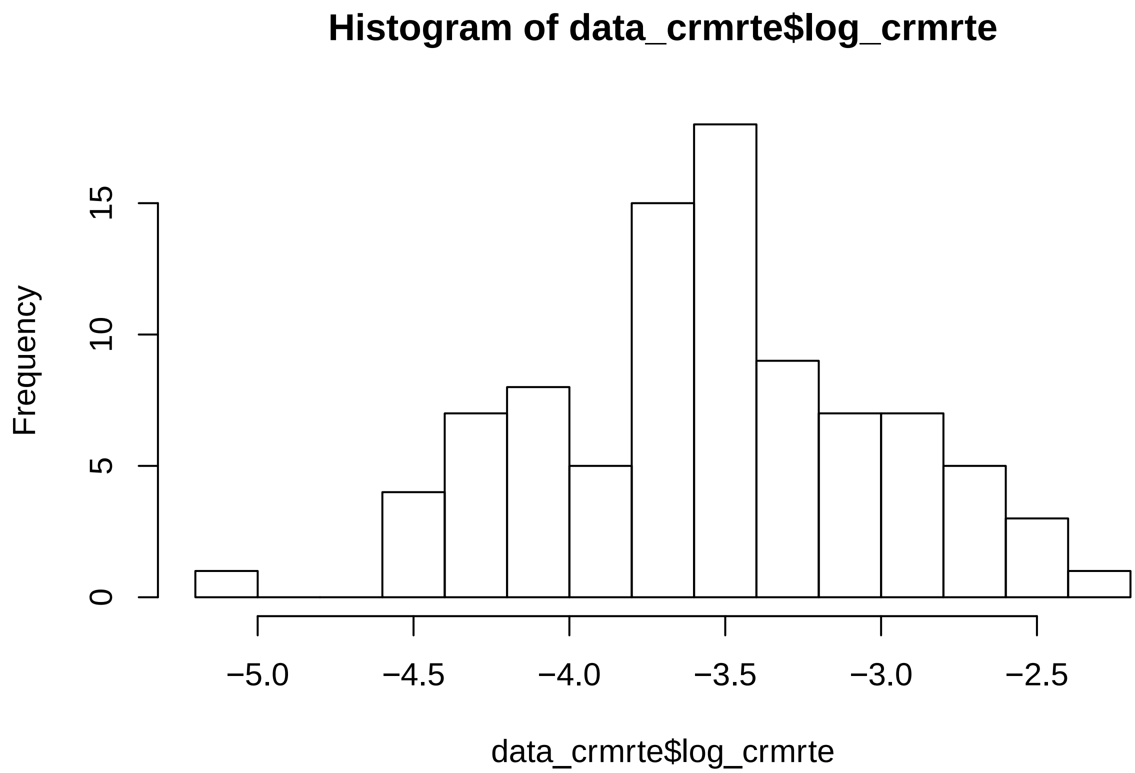
We also look at the lower range of outliers and find only data pt 51 (county 115) which has crime rate $<$ 0.01. This outlier has some significant outlier effects and will be explored further later in the report.

```r
data_crmrte[data_crmrte$crmrte < 0.01,]
```

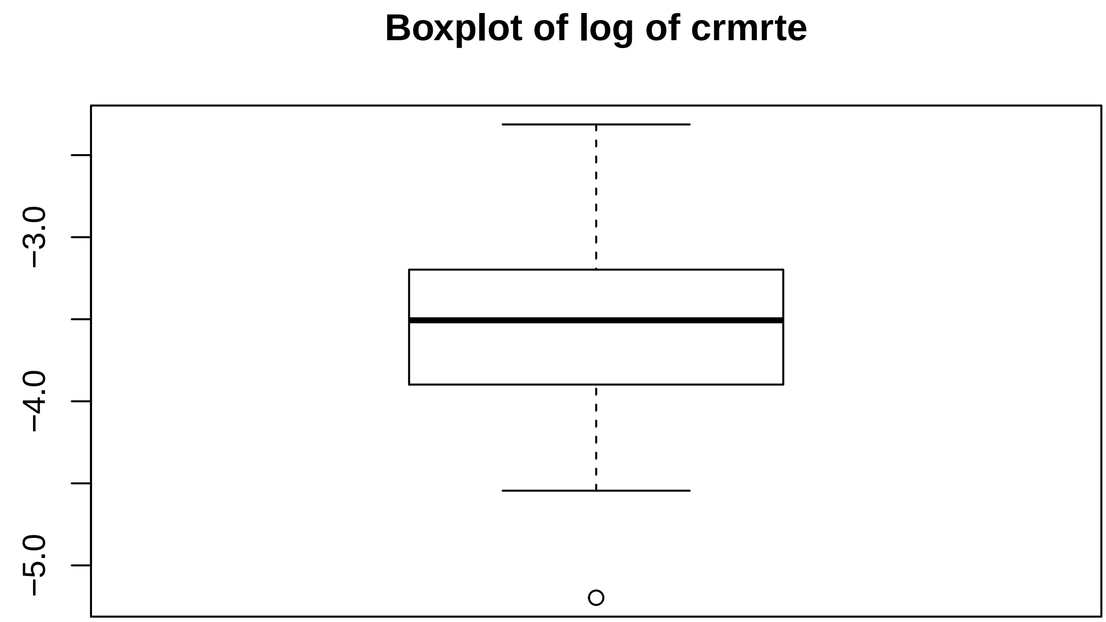
```
##    county year    crmrte   prbarr prbconv prbpris avgsen      polpc   density
## 51    115   87 0.0055332 1.09091     1.5     0.5   20.7 0.00905433 0.3858093
##      taxpc west central urban pctmin80     wcon     wtuc     wtrd     wfir
## 51 28.1931    1       0     0  1.28365 204.2206 503.2351 217.4908 342.4658
##        wser   wmfg  wfed   wsta   wloc mix    pctymle
## 51 245.2061 448.42 442.2 340.39 386.12 0.1 0.07253495
```

For campaign purposes, we want to predict crime. We want our candidate to be able to say that he or she can reduce crime in order to win votes. What is the most effective way to convey that? Using crime rate as it appears in the data set is using the level of crime rate and would suggest the following statement as a campaign slogan - “I can reduce crime to this rate by doing x, y, and z”. Transforming crime rate into the log of crime rate allows for the statement “I can reduce crime by n% by doing x, y, and z.” We find the latter more powerful and meaningful to voters since voters have no idea about the level of crime rates. In addition, we will show that the transformation of crime rate imrpoves the normaility and distribution of the variable, which will often reduce skew in the errors as well.

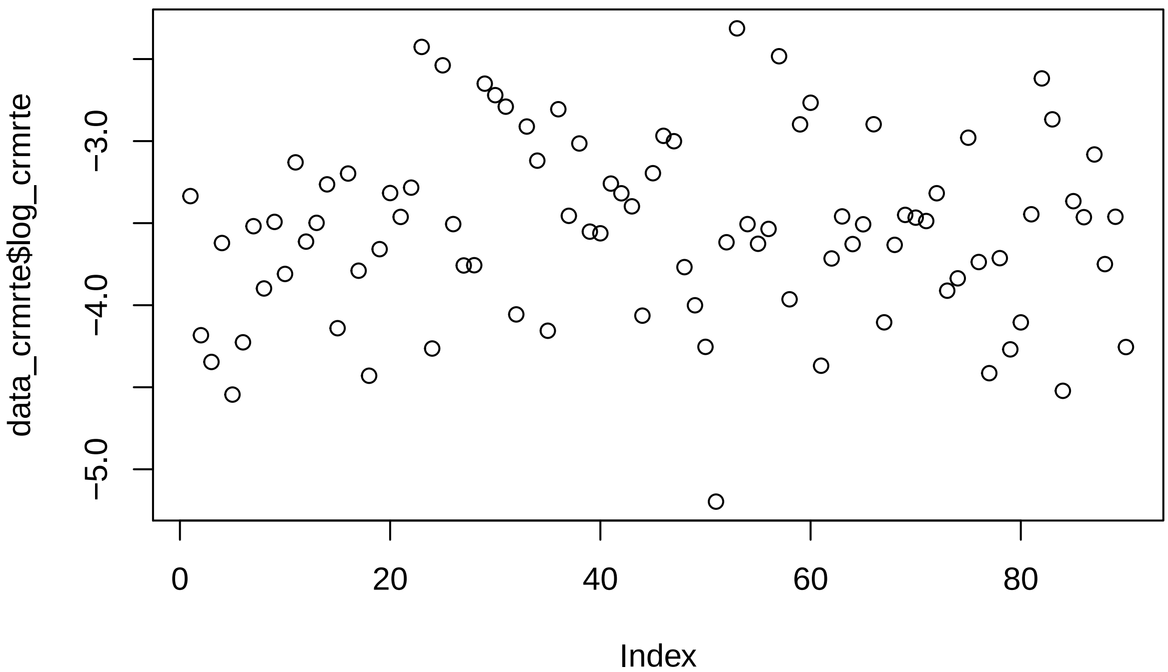
```r
data_crmrte$log_crmrte <- log(data_crmrte$crmrte)
hist(data_crmrte$log_crmrte,breaks=20)
```



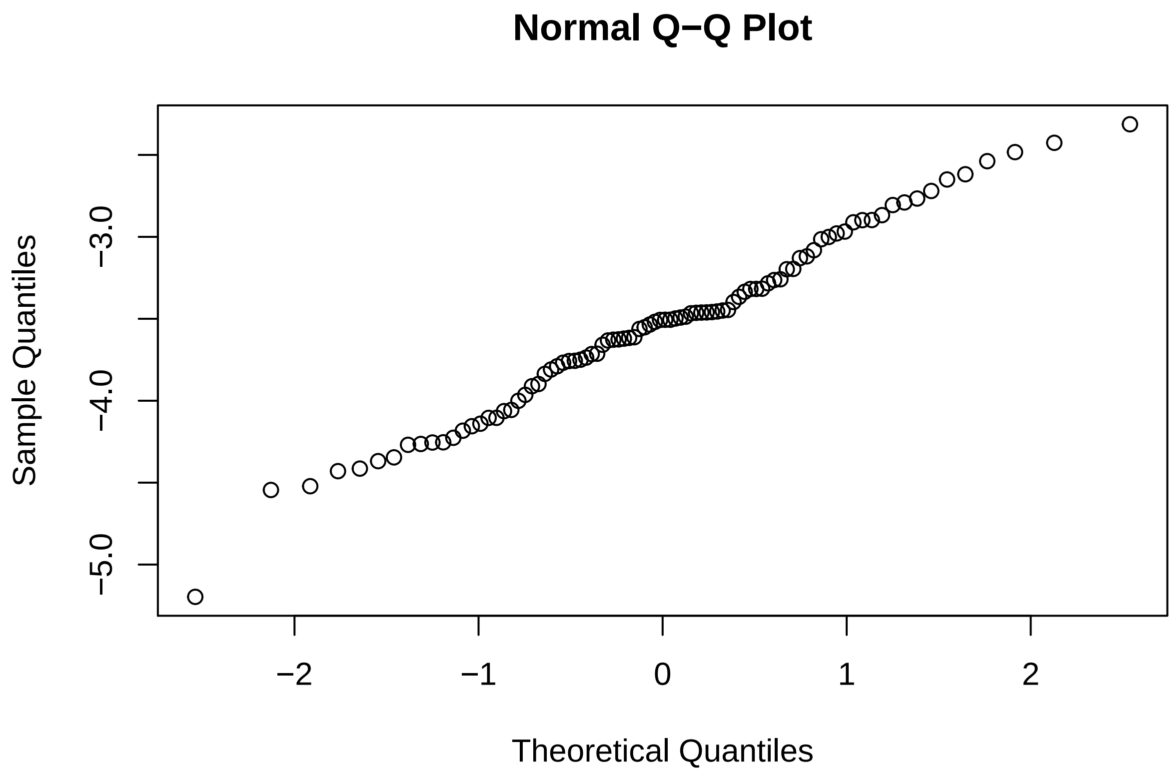

```
boxplot(data_crmrte$log_crmrte, main="Boxplot of log of crmrte")
```




```r
plot(data_crmrte$log_crmrte)
```




```
qqnorm(data_crmrte$log_crmrte)
```




```r
shapiro.test(data_crmrte$crmrte) # Shapiro-wilk test confirms persistance of non-normality
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data_crmrte$crmrte
## W = 0.89162, p-value = 1.741e-06
```

The histogram of the transformed crime rate is much more symmetrical and shows much less right skew. The box plot shows all of the outliers on the high end have been removed, though outlier 51 (countty 115) on the low end has been become more prominent.

The scatter plot looks much more normal, and the Q-Q plot is much closer to normal with the data points hugging the 45 degree line much more closely. Given the stronger argument for the political campaign and the benefits to normality we have chosen to model the tranformation of crime rate as opposed to crime rate.

Groupings

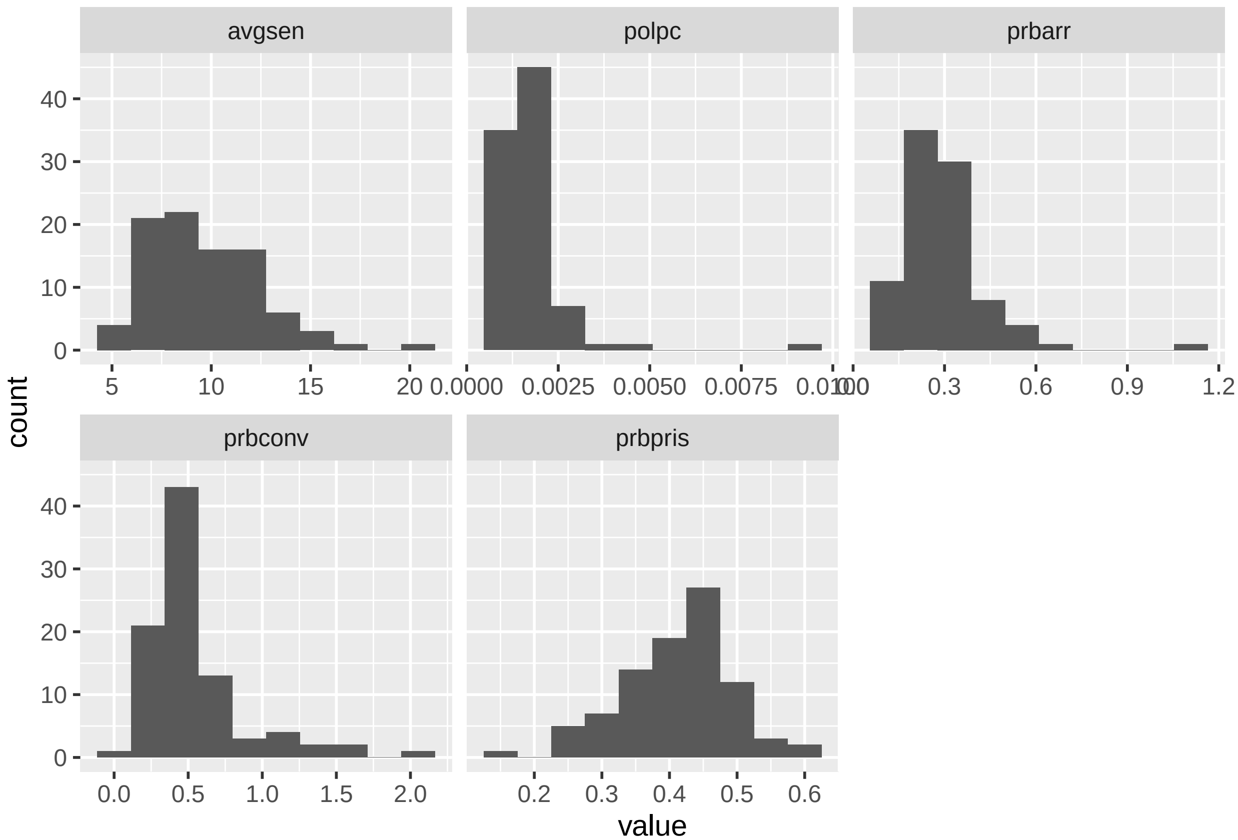
In order to digest the data in the data set we decided to group the variables into five groups: deterrent, wages, demographic, region, and urban. We performed exploratory data analysis on all of these variables.

The group is deterrent data. As cited in the original paper, these variables were hypothesized to reduce crime rate through disincentivizing crime. Essentially, as the probability of getting caught increases, criminals' desire to commit crimes decreases.

Deterrent Data

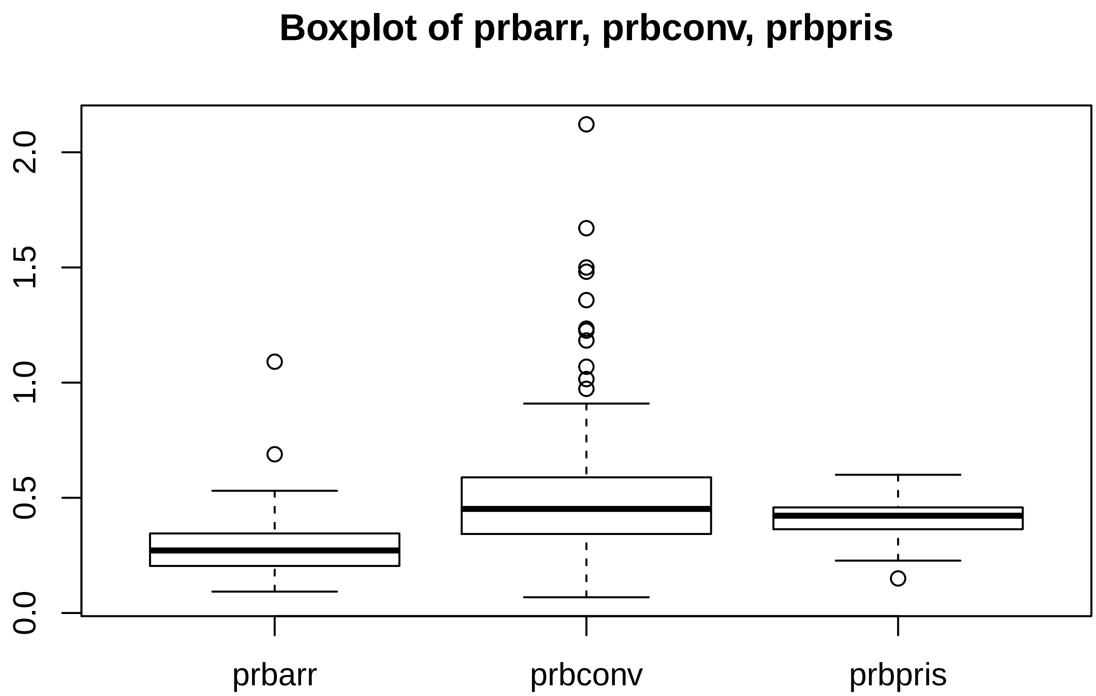
```
deterrent_data <- data_crmrte[,c('prbarr','prbconv','prbpris','crmrte',
                              'avgsen','polpc')]

ggplot(gather(deterrent_data[,c('prbarr','prbconv','prbpris',
                                     'avgsen','polpc')]), aes(value)) +
geom_histogram(bins = 10) + facet_wrap(~key, scales = 'free_x')
```




```
my_vars1 <- c("prbarr","prbconv","prbpris")
deterrent_data2 <- deterrent_data[my_vars1]
my_vars2 <- c("polpc")
deterrent_data3 <- deterrent_data[my_vars2]

boxplot(deterrent_data2, main="Boxplot of prbarr, prbconv, prbpris")
```



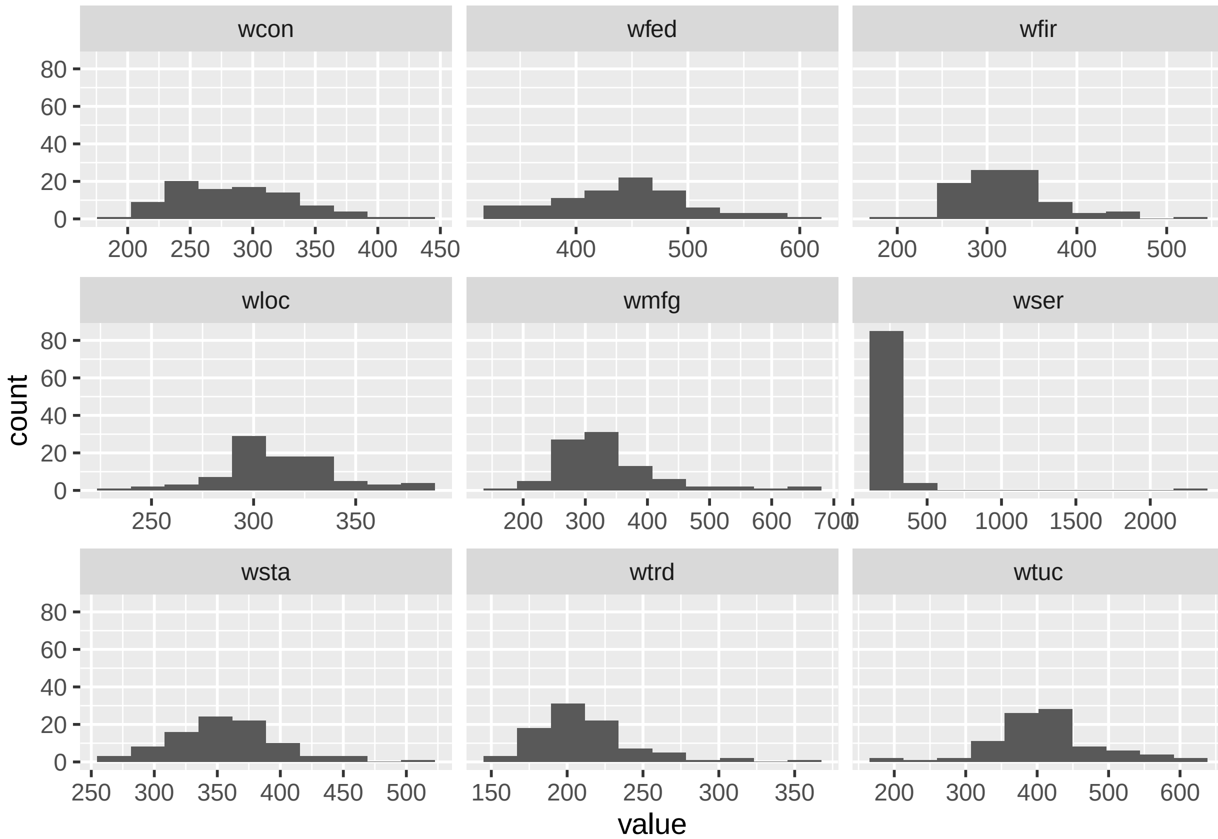

```r
#boxplot(deterrent_data3, main="Boxplot of polpc")
#boxplot(deterrent_data$avgsen, main="Boxplot of avgsen")
```

The first four histograms show right skew while prbpris shows left skew. The biggest outlier is observation 51. This observation has the lowest crime rate in the data set, obviously the highest polpc (police per capita), the highest avg sentence, the third highest prbconv, and the lowest pctmin80. This observation is likely to affect many of the regressions so it will need to be examined further. These variables are candidates to be transformed.

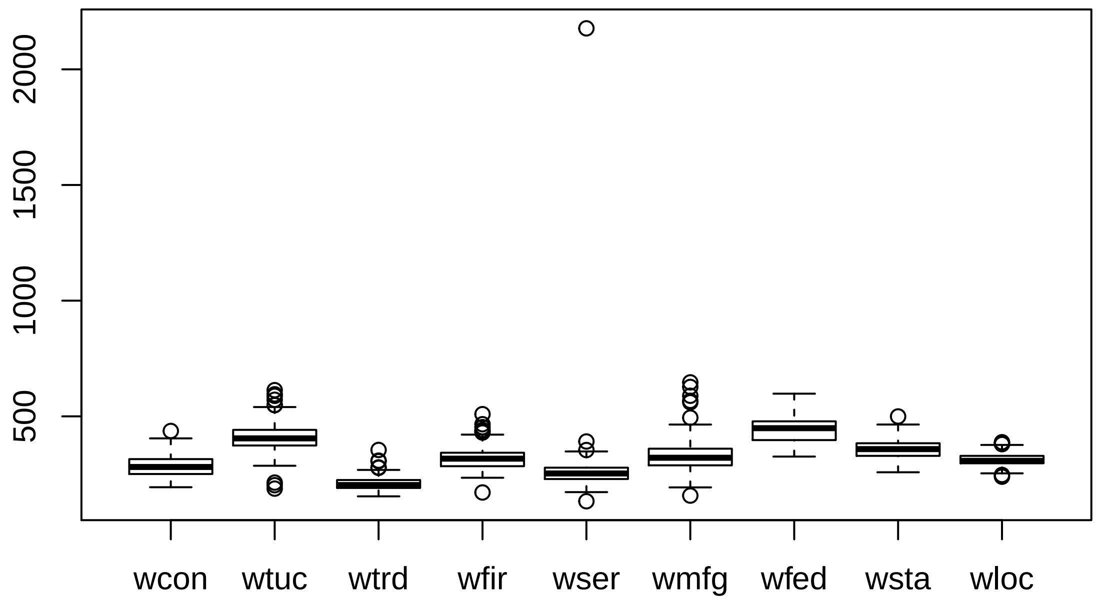
Wages Data

```r
#create a dataframe of just the wage variables
wages_data <- data_crmrte[,c('wcon','wtuc','wtrd','wfir', 'wser',
                          'wmfg','wfed', 'wsta', 'wloc')]

#plot histograms of just the wage variables
ggplot(gather(wages_data), aes(value)) +
  geom_histogram(bins = 10) +
  facet_wrap(~key, scales = 'free_x')
```




```r
#generate boxplots of just the wage variables
boxplot(wages_data)
```



There is an obvious outlier for wser in data pt 84 (County 185) . The mean services wage across all the counties is $275 (with a std dev of 206) and 84 has wser of 2177 (~9sd from mean), which seems like a measurement or typographical error. The next highest average weekly wage in any sector is 646 versus the value of 2177.It is very possible that this data point might add measurement error and we will revisit this later.

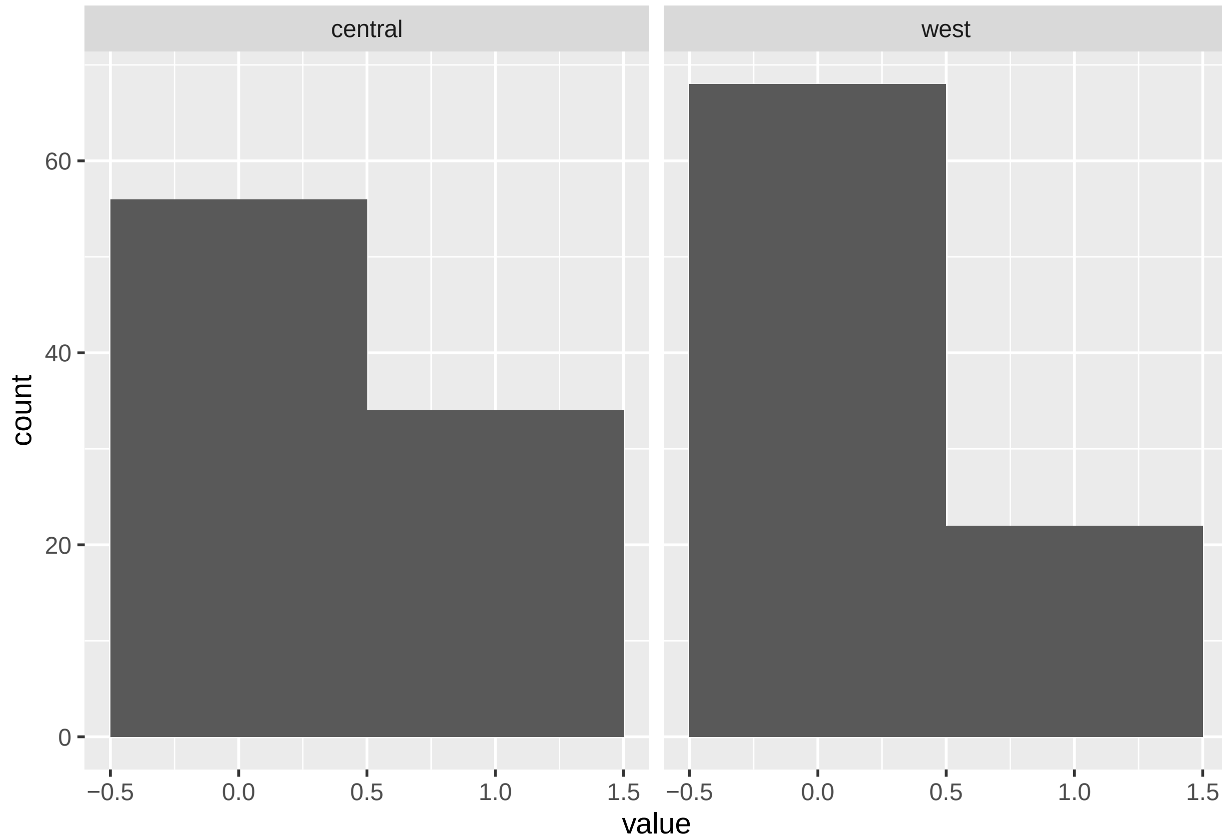
For now, we create an additional variable that is the median of all wage variables for each observation. If it conveys as much information, it has the benefit of increasing our degress of freedom and removing the effect of the outlier.

```
data_crmrte$median_wage <- apply(data_crmrte[c("wcon", "wtuc", "wtrd",
                                               "wfir", "wser", "wmfg",
                                               "wfed", "wsta", "wloc")],
                                 1, FUN=median, na.rm=TRUE)
```

Region Data___

```
#create a dataframe of just the wage variables
dummies_data <- data_crmrte[,c('west','central')]

#plot histograms of just the dummy variables
ggplot(gather(dummies_data), aes(value)) +
  geom_histogram(bins = 2) +
  facet_wrap(~key)
```




```r
#just a quick check that there is no overlap
region_check <- data_crmrte[which(data_crmrte$west == 1 && data_crmrte$central == 1)]

summary(region_check)
```

```
## < table of extent 0 x 0 >
```

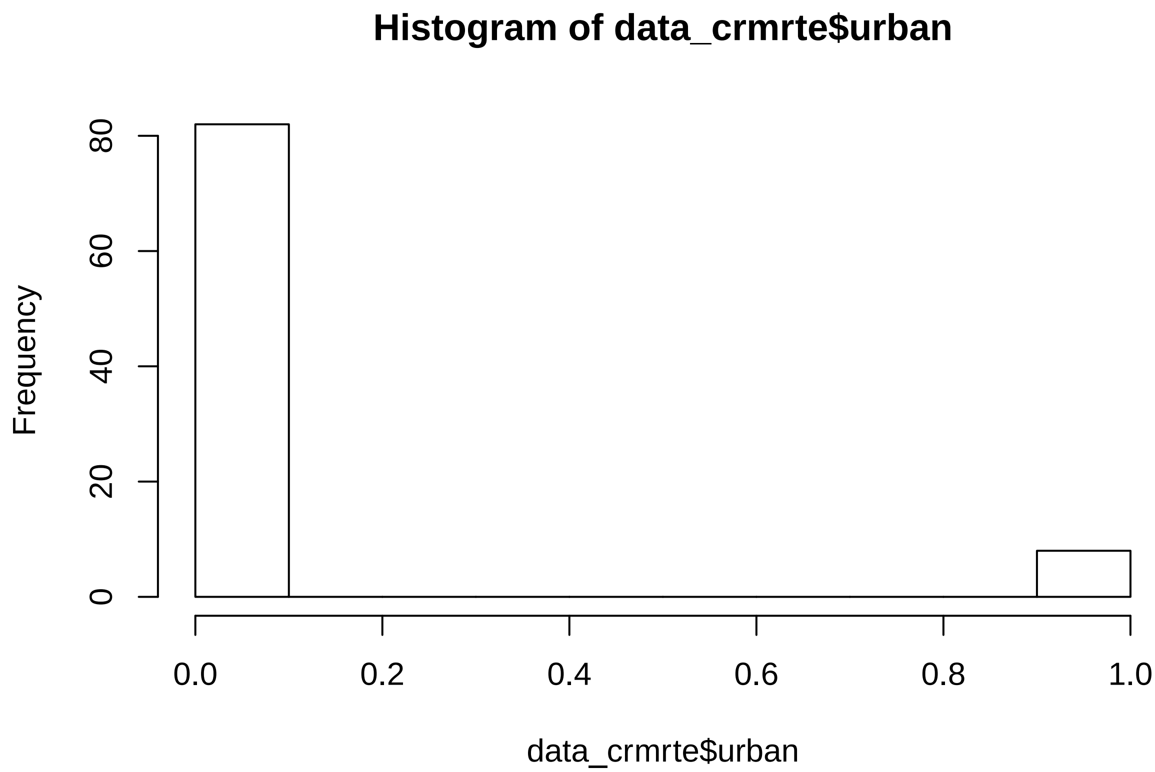
The regions are broken up into central, west, and east. East is left out of the data set and it's effect as the final level of the indicator variable will move to the intercept.

Urban Data

```r
#plot histograms of just the wage variables
sum(data_crmrte$urban) # There are only 8 Urban areas out of 90 counties
```

```
## [1] 8
```

```r
hist(data_crmrte$urban)
```

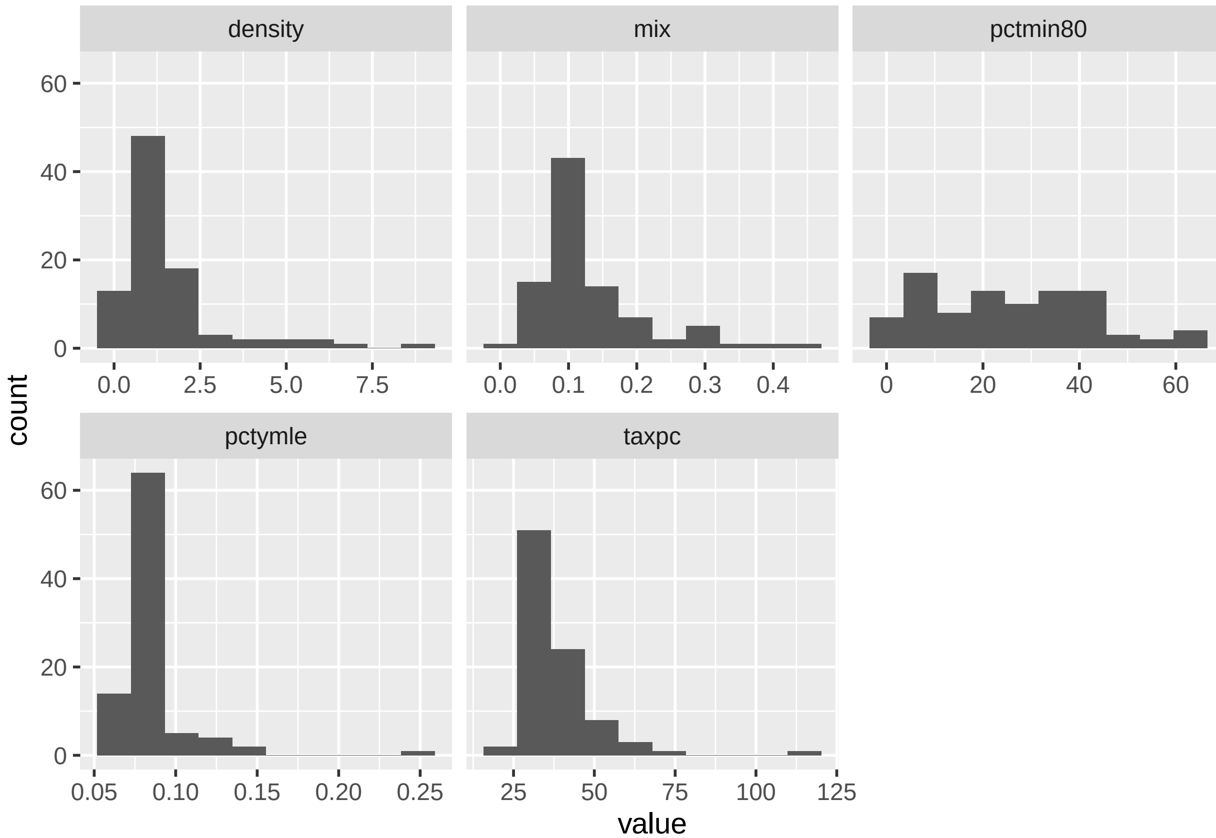


Urban did not fit into a great grouping so we left this variable on its own. A histogram shows that the state has relatively few urban counties, something to keep in mind when analyzing other variables such as density.

Demographic Data

```
#create a dataframe of just the demographic variables
demographic_data <- data_crmrte[,c('density', 'taxpc', 'pctmin80',
                                   'mix', 'pctymle')]

#plot histograms of just the demographic variables
ggplot(gather(demographic_data), aes(value)) +
  geom_histogram(bins = 10) +
  facet_wrap(~key, scales = 'free_x')
```



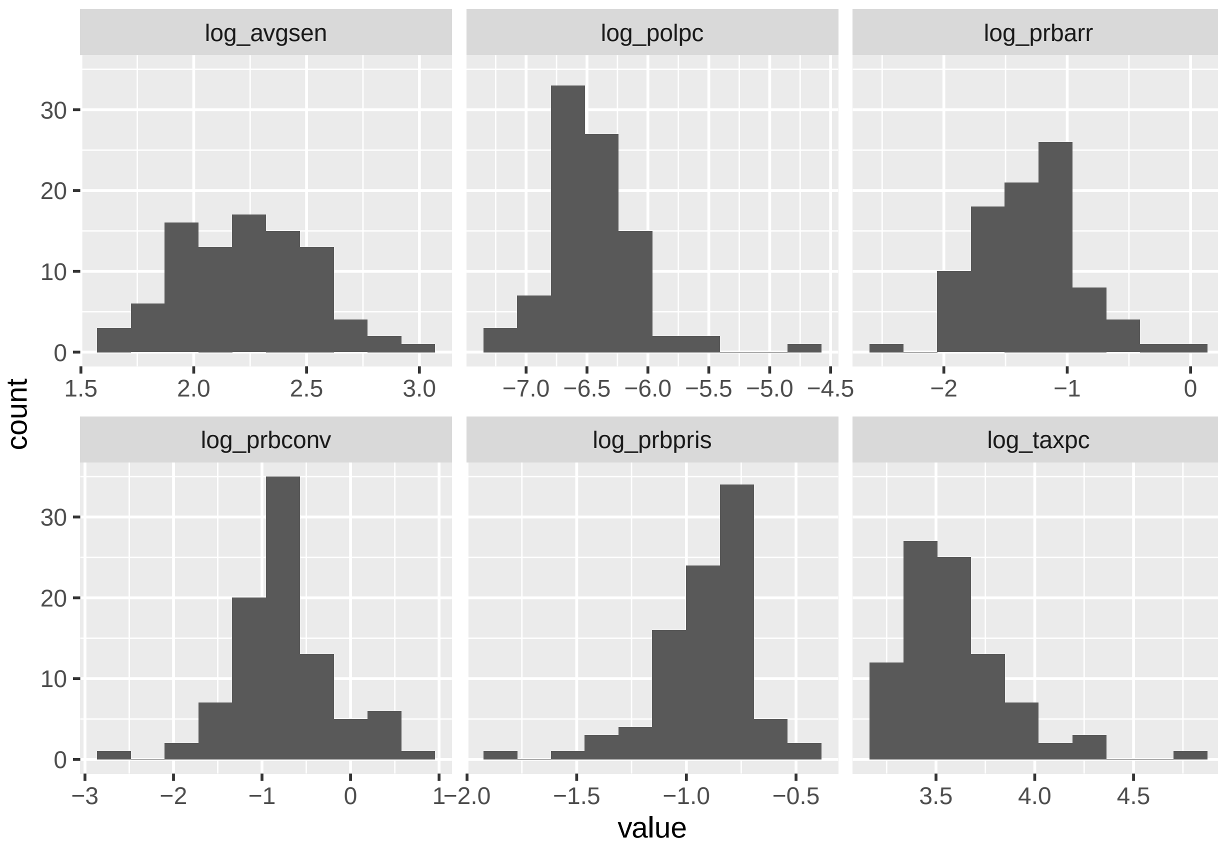

```
#Lots of skewed distributions above, particularly in pctymle and taxpc
# #generate boxplots of just the demographic variables
# demographic_data2 <- demographic_data[c("taxpc", "pctmin80")]
# boxplot(demographic_data2)
# demographic_data3 <- demographic_data[c("pctymle")]
# boxplot(demographic_data3)
# demographic_data4 <- demographic_data[c("density")]
# boxplot(demographic_data4)
# demographic_data5 <- demographic_data[c("mix")]
# boxplot(demographic_data5)
```

Once again we see a lot of right skewed distributions in the historgams and in the box plots.

After exploring all of the variables we decided to tranform the other variables that are potentially under a politican's control - the deterrent variables. This gives us our final data set andso we can start running regressions.

```
data_crmrte$prbconv <- as.numeric(as.character(data_crmrte$prbconv))
data_crmrte$log_prbarr <- log(data_crmrte$prbarr)
data_crmrte$log_prbconv <- log(data_crmrte$prbconv)
data_crmrte$log_prbpris <- log(data_crmrte$prbpris)
data_crmrte$log_avgsen <- log(data_crmrte$avgsen)
data_crmrte$log_polpc <- log(data_crmrte$polpc)
data_crmrte$log_taxpc <- log(data_crmrte$taxpc)
```

```
#plot histograms of just the demographic variables
ggplot(gather(data_crmrte[,c('log_prbarr', 'log_prbconv', 'log_prbpris', 'log_avgsen', 'log_polpc', 'lo
  geom_histogram(bins = 10) +
  facet_wrap(~key, scales = 'free_x')
```



Though the distribution of the variables still exhibits skew, the skew does seem to be reduced.

Log Tranformed Dependent Variable Comparison

In order to settle on the final data set we compare an all-in log-log model with an all-in log-linear to see which dependent variables are more suitable.

```
###### Initial Models #####
all_in_model <- lm(crmrte ~ prbarr + prbconv + prbpris
                   + avgsen + polpc + density
                   + taxpc + west + central + urban + pctmin80 + wcon
                   + wtuc + wtrd + wfir + wser + wmfg
                   + wfed + wsta + wloc
                   + mix + pctymle,
                   data = data_crmrte)
se.all_in_model = sqrt(diag(vcovHC(all_in_model)))
coeftest(all_in_model, vcov = vcovHC)
```

```
##
```

```
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  1.3853e-02  3.0755e-02  0.4504 0.6538622
## prbarr      -5.1466e-02  1.5689e-02 -3.2805 0.0016467 **
## prbconv     -1.8633e-02  6.5853e-03 -2.8295 0.0061464 **
## prbpris      3.1727e-03  1.3586e-02  0.2335 0.8160642
## avgsen      -3.9858e-04  5.5361e-04 -0.7200 0.4740570
## polpc        6.9679e+00  2.9536e+00  2.3591 0.0212406 *
## density      5.3314e-03  1.4895e-03  3.5793 0.0006464 ***
## taxpc        1.6240e-04  2.8408e-04  0.5717 0.5694537
## west        -2.5652e-03  4.4698e-03 -0.5739 0.5679579
## central     -4.2416e-03  3.7423e-03 -1.1334 0.2610725
## urban       -9.6498e-05  8.2752e-03 -0.0117 0.9907307
## pctmin80     3.2542e-04  1.3849e-04  2.3497 0.0217429 *
## wcon         2.3025e-05  3.2876e-05  0.7004 0.4861334
## wtuc         6.1914e-06  1.9862e-05  0.3117 0.7562178
## wtrd         2.8767e-05  8.7294e-05  0.3295 0.7427756
## wfir        -3.5455e-05  3.5699e-05 -0.9932 0.3242068
## wser        -1.7158e-06  9.9447e-05 -0.0173 0.9862856
## wmfg        -8.9675e-06  1.7469e-05 -0.5133 0.6094087
## wfed         2.9075e-05  3.7780e-05  0.7696 0.4442480
## wsta        -2.2302e-05  3.6828e-05 -0.6056 0.5468431
## wloc         1.4456e-05  8.5367e-05  0.1693 0.8660410
## mix         -1.8693e-02  2.2922e-02 -0.8155 0.4176761
## pctymle      1.0125e-01  4.7826e-02  2.1170 0.0379748 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(all_in_model)
```

```
## [1] -585.5858
```

```r
all_in_model_log_level <- lm(log_crmrte ~ prbarr + prbconv + prbpris
                             + avgsen + polpc + density
                             + taxpc + west + central + urban
                             + pctmin80 + wcon
                             + wtuc + wtrd + wfir + wser + wmfg
                             + wfed + wsta + wloc
                             + mix + pctymle,
                             data = data_crmrte)
se.all_in_model_log_level = sqrt(diag(vcovHC(all_in_model_log_level)))
coeftest(all_in_model_log_level, vcov = vcovHC)
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept) -4.0261e+00  8.4822e-01 -4.7466 1.128e-05 ***
## prbarr      -1.8891e+00  3.7955e-01 -4.9773 4.770e-06 ***
## prbconv     -6.5603e-01  1.7443e-01 -3.7611 0.0003579 ***
## prbpris     -9.3077e-02  3.9921e-01 -0.2332 0.8163542
## avgsen      -7.8769e-03  1.6125e-02 -0.4885 0.6267962
```

```
## polpc        1.5484e+02  8.6523e+01  1.7895 0.0780510 .
## density      1.1653e-01  5.4037e-02  2.1566 0.0346326 *
## taxpc        3.3224e-03  7.2890e-03  0.4558 0.6500012
## west        -1.1492e-01  1.2509e-01 -0.9187 0.3615403
## central     -1.0078e-01  9.2053e-02 -1.0948 0.2775232
## urban       -1.6923e-01  2.2872e-01 -0.7399 0.4619535
## pctmin80     9.9770e-03  3.0480e-03  3.2733 0.0016833 **
## wcon         4.6001e-04  8.3564e-04  0.5505 0.5838140
## wtuc         1.0174e-04  6.0187e-04  0.1690 0.8662750
## wtrd         2.5964e-04  1.7638e-03  0.1472 0.8834136
## wfir        -1.1015e-03  1.1960e-03 -0.9210 0.3603557
## wser        -1.3142e-04  1.5060e-03 -0.0873 0.9307193
## wmfg        -2.0528e-04  5.1630e-04 -0.3976 0.6921878
## wfed         2.3405e-03  1.0820e-03  2.1632 0.0340968 *
## wsta        -1.1357e-03  8.9769e-04 -1.2651 0.2102213
## wloc         5.8983e-04  2.4003e-03  0.2457 0.8066400
## mix         -2.3924e-01  6.2632e-01 -0.3820 0.7036869
## pctymle      2.7706e+00  1.4330e+00  1.9334 0.0574191 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(all_in_model_log_level)
```

```
## [1] 21.354
```

```r
all_in_model_log_log <- lm(log_crmrte ~ log_prbarr + log_prbconv
                             + log_prbpris + log_avgsen + log_polpc
                             + density+ log_taxpc + west + central
                             + urban + pctmin80 + wcon
                             + wtuc + wtrd + wfir
                             + wser + wmfg + wfed + wsta + wloc
                             + mix + pctymle,
                             data = data_crmrte)
se.all_in_model_log_log = sqrt(diag(vcovHC(all_in_model_log_log)))
coeftest(all_in_model_log_log, vcov = vcovHC)
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value Pr(>|t|)
## (Intercept) -3.36882669  2.97497990 -1.1324 0.261508
## log_prbarr  -0.52143620  0.16459898 -3.1679 0.002313 **
## log_prbconv -0.33101341  0.15365522 -2.1543 0.034820 *
## log_prbpris -0.06569465  0.19741379 -0.3328 0.740342
## log_avgsen  -0.19652151  0.18205821 -1.0794 0.284261
## log_polpc    0.29132794  0.27176129  1.0720 0.287567
## density      0.12320127  0.06040422  2.0396 0.045335 *
## log_taxpc    0.06158051  0.30979897  0.1988 0.843040
## west        -0.18453792  0.16353910 -1.1284 0.263174
## central     -0.10789292  0.09991865 -1.0798 0.284100
## urban       -0.14767055  0.26670745 -0.5537 0.581641
## pctmin80     0.00956927  0.00358175  2.6717 0.009466 **
## wcon         0.00078953  0.00090745  0.8701 0.387376
```

```
## wtuc         0.00010106  0.00075559  0.1337 0.894001
## wtrd         0.00029022  0.00177967  0.1631 0.870952
## wfir        -0.00108230  0.00125937 -0.8594 0.393186
## wser        -0.00042887  0.00096365 -0.4451 0.657718
## wmfg        -0.00014147  0.00061356 -0.2306 0.818343
## wfed         0.00224918  0.00136611  1.6464 0.104363
## wsta        -0.00102039  0.00106131 -0.9614 0.339787
## wloc         0.00017815  0.00261968  0.0680 0.945986
## mix         -0.44834658  0.77846459 -0.5759 0.566587
## pctymle      2.00755501  2.60186976  0.7716 0.443075
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(all_in_model_log_log)
```

```
## [1] 44.17803
```

```r
BIC(all_in_model_log_log)
```

```
## [1] 104.1735
```

```r
#Not comparing r-squared, just looking at significant variables
stargazer(all_in_model, all_in_model_log_level,
          all_in_model_log_log,
          type = "text", omit.stat = "f",
          se = list(se.all_in_model, se.all_in_model_log_level,
                    se.all_in_model_log_log),
          star.cutoffs = c(0.05, 0.01, 0.001))
```

```
##
## ==============================================================
##                                    Dependent variable:
##                               ------------------------------
##                                 crmrte        log_crmrte
##                                  (1)        (2)        (3)
## --------------------------------------------------------------
## prbarr                         -0.051**  -1.889***
##                                (0.016)    (0.380)
##
## prbconv                        -0.019**  -0.656***
##                                (0.007)    (0.174)
##
## prbpris                         0.003      -0.093
##                                (0.014)    (0.399)
##
## avgsen                         -0.0004     -0.008
##                                (0.001)    (0.016)
##
## polpc                           6.968*    154.835
##                                (2.954)    (86.523)
##
## log_prbarr                                           -0.521**
```

```
##                                                     (0.165)
##
## log_prbconv                                         -0.331*
##                                                     (0.154)
##
## log_prbpris                                         -0.066
##                                                     (0.197)
##
## log_avgsen                                          -0.197
##                                                     (0.182)
##
## log_polpc                                            0.291
##                                                     (0.272)
##
## density                       0.005***    0.117*    0.123*
##                               (0.001)    (0.054)    (0.060)
##
## taxpc                         0.0002      0.003
##                              (0.0003)    (0.007)
##
## log_taxpc                                            0.062
##                                                     (0.310)
##
## west                          -0.003     -0.115     -0.185
##                               (0.004)    (0.125)    (0.164)
##
## central                       -0.004     -0.101     -0.108
##                               (0.004)    (0.092)    (0.100)
##
## urban                         -0.0001    -0.169     -0.148
##                               (0.008)    (0.229)    (0.267)
##
## pctmin80                      0.0003*    0.010**    0.010**
##                               (0.0001)   (0.003)    (0.004)
##
## wcon                          0.00002     0.0005     0.001
##                              (0.00003)   (0.001)    (0.001)
##
## wtuc                          0.00001     0.0001    0.0001
##                              (0.00002)   (0.001)    (0.001)
##
## wtrd                          0.00003     0.0003    0.0003
##                               (0.0001)   (0.002)    (0.002)
##
## wfir                          -0.00004   -0.001     -0.001
##                              (0.00004)   (0.001)    (0.001)
##
## wser                          -0.00000   -0.0001    -0.0004
##                               (0.0001)   (0.002)    (0.001)
##
## wmfg                          -0.00001   -0.0002    -0.0001
##                              (0.00002)   (0.001)    (0.001)
##
## wfed                          0.00003     0.002*     0.002
```

```
##                               (0.00004)   (0.001)    (0.001)
##
## wsta                           -0.00002    -0.001     -0.001
##                               (0.00004)   (0.001)    (0.001)
##
## wloc                            0.00001     0.001     0.0002
##                                (0.0001)   (0.002)    (0.003)
##
## mix                             -0.019     -0.239     -0.448
##                                (0.023)    (0.626)    (0.778)
##
## pctymle                         0.101*      2.771      2.008
##                                (0.048)    (1.433)    (2.602)
##
## Constant                        0.014    -4.026***   -3.369
##                                (0.031)    (0.848)    (2.975)
##
## ----------------------------------------------------------------
## Observations                      90         90         90
## R2                              0.855      0.854      0.812
## Adjusted R2                     0.807      0.806      0.750
## Residual Std. Error (df = 67)   0.008      0.242      0.275
## ================================================================
## Note:                            *p<0.05; **p<0.01; ***p<0.001
```

```
# #r-squared comparison of final two models
# yhat_level_level <- predict(all_in_model)
#
# #get the coefficients
# for (b in coef(all_in_model_log_level))
# {
#   beta_log_level <- c(beta_log_level, b)
# }
# #calculate the predictions
# for (b in coef(all_in_model_log_level))
# {
#   beta_log_level <- c(beta_log_level, b)
# }
#
# data_crmrte$log_level_yhat <- exp(--3.36882669
#                                   -0.5214362*data_crmrte$log_prbarr
#                                   -0.33101341*data_crmrte$log_prbconv
#                                   -0.06569465*data_crmrte$log_prbpris
#                                   -0.19652151*data_crmrte$log_avgsen
#                                   +0.29132794*data_crmrte$log_polpc
#                                   +0.12320127*data_crmrte$density
#                                   +0.06158051*data_crmrte$log_taxpc
#                                   -0.18453792*data_crmrte$west
#                                   -0.10789292*data_crmrte$central
#                                   -0.14767055*data_crmrte$urban
#                                   +0.00956927*data_crmrte$pctmin80
#                                   +0.00078953*data_crmrte$wcon
#                                   +0.00010106*data_crmrte$wtuc
#                                   +0.00029022*data_crmrte$wtrd
```

```
#                                   -0.0010823*data_crmrte$wfir
#                                   -0.00042887*data_crmrte$wser
#                                   -0.00014147*data_crmrte$wmfg
#                                   +0.00224918*data_crmrte$wfed
#                                   -0.00102039*data_crmrte$wsta
#                                   +0.00017815*data_crmrte$wloc
#                                   -0.44834658*data_crmrte$mix
#                                   +2.00755501*data_crmrte$pctymle
#                                   )
# r_squared_level_level <- cor(data_crmrte$crmrte, yhat_level_level)
# r_squared_log_level <- cor(data_crmrte$crmrte, data_crmrte$log_level_yhat)
# (r_squared_level_level)
# (r_squared_log_level)
```

Model 1: Simple Model

In order to create a simple model we decided to build using a bottom up approach. We looked at a correlation matrix

Anyone know how to print this better?

```
round(cor(data_crmrte$log_crmrte,data_crmrte)*100,2) #Corr Matrix as % for reading clarity
```

```
## Warning in cor(data_crmrte$log_crmrte, data_crmrte): the standard deviation is
## zero
```

```
##      county year crmrte prbarr prbconv prbpris avgsen polpc density taxpc
## [1,]   2.38   NA  94.15 -47.28  -44.68    2.15  -4.94  1.04    63.3 35.83
##        west central urban pctmin80  wcon  wtuc  wtrd  wfir   wser  wmfg  wfed
## [1,] -41.44   18.47 49.15    23.29 39.37 20.15 39.38 29.32 -11.31 30.75 52.33
##       wsta  wloc    mix pctymle log_crmrte median_wage log_prbarr log_prbconv
## [1,] 16.97 28.86 -12.47   27.82        100       45.44     -43.58      -37.25
##      log_prbpris log_avgsen log_polpc log_taxpc
## [1,]        6.96       2.34     28.45     33.98
```

In the above correlation matrix, focusing on the correlations between the log_crmrte and all other variables, denisty has the highest correlation. This variable makes intuitive sense. As a single variable it might encompass a lot of other factors. Lower income people with more incentive to commit crimes tend to live in more highly populated areas. Below is the simple regression.

```
simple_regression_model <- lm(log_crmrte ~ density, data = data_crmrte)
se.simple_regression_model = sqrt(diag(vcovHC(simple_regression_model)))
coeftest(simple_regression_model, vcov = vcovHC)
```

```
##
## t test of coefficients:
##
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -3.869488   0.068563 -56.4366  < 2e-16 ***
```

```
## density      0.228298   0.030439   7.5003  4.8e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(simple_regression_model)
```

```
## [1] 106.2991
```

```r
stargazer(simple_regression_model,
          type = "text", omit.stat = "f",
          se = list(se.simple_regression_model),
          star.cutoffs = c(0.05, 0.01, 0.001))
```

```
##
## ===================================================
##                          Dependent variable:
##                      -----------------------------
##                               log_crmrte
## ---------------------------------------------------
## density                        0.228***
##                                 (0.030)
##
## Constant                       -3.869***
##                                 (0.069)
##
## ---------------------------------------------------
## Observations                       90
## R2                               0.401
## Adjusted R2                      0.394
## Residual Std. Error         0.427 (df = 88)
## ===================================================
## Note:                 *p<0.05; **p<0.01; ***p<0.001
```

```r
#plot(simple_regression_model)
```

The variable density explains 40.1% of the variation in the log of crime rate. As density increases by 1 unit (as the county population divided by the county land area increases by 1%) crime increases by 22%.

Model 2: Kitchen Sink Model
Still, we can do better in predicting the log crime rate than simply using one variable. We know examine a "kitchen sink" model. This model includes all of the variables in the data set except county (which has too many values to be a useful indicator variable) and year, which is a constant (1987). Below are the results.

```r
all_in_model_log_level <- lm(log_crmrte ~ prbarr + prbconv + prbpris
                             + avgsen + polpc + density
                             + taxpc + west + central + urban
                             + pctmin80 + wcon
                             + wtuc + wtrd + wfir + wser + wmfg
                             + wfed + wsta + wloc
                             + mix + pctymle,
                             data = data_crmrte)
se.all_in_model_log_level = sqrt(diag(vcovHC(all_in_model_log_level))) #HC White SE
coeftest(all_in_model_log_level, vcov = vcovHC)
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept) -4.0261e+00  8.4822e-01 -4.7466 1.128e-05 ***
## prbarr      -1.8891e+00  3.7955e-01 -4.9773 4.770e-06 ***
## prbconv     -6.5603e-01  1.7443e-01 -3.7611 0.0003579 ***
## prbpris     -9.3077e-02  3.9921e-01 -0.2332 0.8163542
## avgsen      -7.8769e-03  1.6125e-02 -0.4885 0.6267962
## polpc        1.5484e+02  8.6523e+01  1.7895 0.0780510 .
## density      1.1653e-01  5.4037e-02  2.1566 0.0346326 *
## taxpc        3.3224e-03  7.2890e-03  0.4558 0.6500012
## west        -1.1492e-01  1.2509e-01 -0.9187 0.3615403
## central     -1.0078e-01  9.2053e-02 -1.0948 0.2775232
## urban       -1.6923e-01  2.2872e-01 -0.7399 0.4619535
## pctmin80     9.9770e-03  3.0480e-03  3.2733 0.0016833 **
## wcon         4.6001e-04  8.3564e-04  0.5505 0.5838140
## wtuc         1.0174e-04  6.0187e-04  0.1690 0.8662750
## wtrd         2.5964e-04  1.7638e-03  0.1472 0.8834136
## wfir        -1.1015e-03  1.1960e-03 -0.9210 0.3603557
## wser        -1.3142e-04  1.5060e-03 -0.0873 0.9307193
## wmfg        -2.0528e-04  5.1630e-04 -0.3976 0.6921878
## wfed         2.3405e-03  1.0820e-03  2.1632 0.0340968 *
## wsta        -1.1357e-03  8.9769e-04 -1.2651 0.2102213
## wloc         5.8983e-04  2.4003e-03  0.2457 0.8066400
## mix         -2.3924e-01  6.2632e-01 -0.3820 0.7036869
## pctymle      2.7706e+00  1.4330e+00  1.9334 0.0574191 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(all_in_model_log_level)
```

```
## [1] 21.354
```

```r
stargazer(simple_regression_model, all_in_model_log_level,
          type = "text", omit.stat = "f",
          se = list(se.simple_regression_model, se.all_in_model_log_level),
          star.cutoffs = c(0.05, 0.01, 0.001))
```

```
##
## ===================================================
##                          Dependent variable:
##                     -------------------------------
##                               log_crmrte
##                          (1)             (2)
## ---------------------------------------------------
## prbarr                                -1.889***
##                                        (0.380)
##
## prbconv                               -0.656***
##                                        (0.174)
##
## prbpris                                -0.093
```

```
##                                     (0.399)
##
## avgsen                              -0.008
##                                     (0.016)
##
## polpc                               154.835
##                                     (86.523)
##
## density              0.228***        0.117*
##                      (0.030)        (0.054)
##
## taxpc                                0.003
##                                     (0.007)
##
## west                                -0.115
##                                     (0.125)
##
## central                             -0.101
##                                     (0.092)
##
## urban                               -0.169
##                                     (0.229)
##
## pctmin80                            0.010**
##                                     (0.003)
##
## wcon                                0.0005
##                                     (0.001)
##
## wtuc                                0.0001
##                                     (0.001)
##
## wtrd                                0.0003
##                                     (0.002)
##
## wfir                                -0.001
##                                     (0.001)
##
## wser                                -0.0001
##                                     (0.002)
##
## wmfg                                -0.0002
##                                     (0.001)
##
## wfed                                0.002*
##                                     (0.001)
##
## wsta                                -0.001
##                                     (0.001)
##
## wloc                                 0.001
##                                     (0.002)
##
## mix                                 -0.239
```

```
##                                       (0.626)
##
## pctymle                                2.771
##                                       (1.433)
##
## Constant                -3.869***       -4.026***
##                          (0.069)        (0.848)
##
## ---------------------------------------------------
## Observations               90             90
## R2                        0.401          0.854
## Adjusted R2               0.394          0.806
## Residual Std. Error 0.427 (df = 88) 0.242 (df = 67)
## ===================================================
## Note:                 *p<0.05; **p<0.01; ***p<0.001
```

Unsuprisingly, the r-squared of the "kitchen sink" model is substantially higher (85.4% vs. 40.1%). More importantly, the adjusted r-squared which accounts for the number of variables in the models, is also higher (80.6% vs 39.4%). Interestingly, density is no longer the variable with the highest statistical significance. The coefficients show the effect after all the other variables have been controlled for (partialled out). In the "kitchen sink" model prbarr and prbconv both have the lowest p-values.

Model 3: Balanced Model

We took two approaches to building the balanced model. We used a bottom up approach that relied on both the correlation matrix and stepwide regression. We also used a top down approach that started with the "kitchen sink" model and excluded variables. Both methods are discussed below. Both approaches relied on our categories of variables to simplify the process.

```r
base_forward = lm(log_crmrte ~ density,
                             data = data_crmrte)
# forward_step = step(base_forward, scope = formula(all_in_model_log_level), direction = "forward")
```

With the top down approach, we started with model 3 and looked to exclude variables that weren't as predictive. We ran hypothesis testing on all five groups, one group at a time.

```r
#deterrent
linearHypothesis(all_in_model_log_level,
                 c("prbarr = 0", "prbconv = 0", "prbpris = 0",
                   "avgsen = 0", "polpc = 0"),
                 vcov = vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## prbarr = 0
## prbconv = 0
## prbpris = 0
## avgsen = 0
## polpc = 0
##
## Model 1: restricted model
```

```
## Model 2: log_crmrte ~ prbarr + prbconv + prbpris + avgsen + polpc + density +
##     taxpc + west + central + urban + pctmin80 + wcon + wtuc +
##     wtrd + wfir + wser + wmfg + wfed + wsta + wloc + mix + pctymle
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F    Pr(>F)
## 1     72
## 2     67  5 6.0582 0.0001101 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#wage
linearHypothesis(all_in_model_log_level,
                 c("wcon = 0", "wtuc = 0", "wtrd = 0",
                   "wfir = 0", "wser = 0", "wmfg = 0",
                   "wfed = 0", "wsta = 0", "wloc = 0"),
                 vcov = vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## wcon = 0
## wtuc = 0
## wtrd = 0
## wfir = 0
## wser = 0
## wmfg = 0
## wfed = 0
## wsta = 0
## wloc = 0
##
## Model 1: restricted model
## Model 2: log_crmrte ~ prbarr + prbconv + prbpris + avgsen + polpc + density +
##     taxpc + west + central + urban + pctmin80 + wcon + wtuc +
##     wtrd + wfir + wser + wmfg + wfed + wsta + wloc + mix + pctymle
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df     F Pr(>F)
## 1     76
## 2     67  9 1.372 0.2185
```

```
#region
linearHypothesis(all_in_model_log_level,
                 c("west = 0", "central = 0"),
                 vcov = vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## west = 0
## central = 0
```

```
##
## Model 1: restricted model
## Model 2: log_crmrte ~ prbarr + prbconv + prbpris + avgsen + polpc + density +
##     taxpc + west + central + urban + pctmin80 + wcon + wtuc +
##     wtrd + wfir + wser + wmfg + wfed + wsta + wloc + mix + pctymle
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df     F Pr(>F)
## 1     69
## 2     67  2 0.623 0.5394
```

```r
#urban
linearHypothesis(all_in_model_log_level,
                 c("urban = 0"),
                 vcov = vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## urban = 0
##
## Model 1: restricted model
## Model 2: log_crmrte ~ prbarr + prbconv + prbpris + avgsen + polpc + density +
##     taxpc + west + central + urban + pctmin80 + wcon + wtuc +
##     wtrd + wfir + wser + wmfg + wfed + wsta + wloc + mix + pctymle
##
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F Pr(>F)
## 1     68
## 2     67  1 0.5474  0.462
```

```r
#demographic
linearHypothesis(all_in_model_log_level,
                 c("density = 0", "taxpc = 0", "pctmin80 = 0",
                   "mix = 0", "pctymle = 0"),
                 vcov = vcovHC)
```

```
## Linear hypothesis test
##
## Hypothesis:
## density = 0
## taxpc = 0
## pctmin80 = 0
## mix = 0
## pctymle = 0
##
## Model 1: restricted model
## Model 2: log_crmrte ~ prbarr + prbconv + prbpris + avgsen + polpc + density +
##     taxpc + west + central + urban + pctmin80 + wcon + wtuc +
##     wtrd + wfir + wser + wmfg + wfed + wsta + wloc + mix + pctymle
##
```

```
## Note: Coefficient covariance matrix supplied.
##
##   Res.Df Df      F   Pr(>F)
## 1     72
## 2     67  5 3.9627 0.003298 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

/ The hypothesis tests below show that of the five groups the only groups that are jointly significant are the deterrent data and the demographic data. These tests measure whether removing all the variables within a group reduces the r-squared by s attsitically signficant amount. We will re-run the models and compare.

```
balanced_model_top_1 <- lm(log_crmrte ~ prbarr + prbconv + prbpris
                             + avgsen + polpc + density
                             + taxpc + pctmin80 + mix + pctymle,
                             data = data_crmrte)
se.balanced_model_top_1 = sqrt(diag(vcovHC(balanced_model_top_1)))
coeftest(balanced_model_top_1, vcov = vcovHC)
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  -3.3522918   0.3558217 -9.4213 1.467e-14 ***
## prbarr       -1.9627484   0.4014808 -4.8888 5.228e-06 ***
## prbconv      -0.7672158   0.1366862 -5.6130 2.846e-07 ***
## prbpris      -0.0764993   0.4732818 -0.1616  0.872005
## avgsen       -0.0044749   0.0140406 -0.3187  0.750789
## polpc       176.1347220  82.5884550  2.1327  0.036056 *
## density       0.1135225   0.0351279  3.2317  0.001796 **
## taxpc         0.0020988   0.0055753  0.3764  0.707593
## pctmin80      0.0125062   0.0016215  7.7128 3.155e-11 ***
## mix          -0.7304967   0.5396416 -1.3537  0.179702
## pctymle       1.3832565   1.6211791  0.8532  0.396105
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(balanced_model_top_1)
```

```
## [1] 27.4514
```

```
#stargazer(simple_regression_model, all_in_model_log_level, balanced_model_top_1,
#          type = "text", omit.stat = "f",
#          se = list(se.simple_regression_model, se.all_in_model_log_level, #se.balanced_model_top_1),
#          star.cutoffs = c(0.05, 0.01, 0.001))
```

Our adjusted r-squared has only fallen from 80.6% to 77.6% but we have dropped 12 variables. This is a much more parisimous model. In order to double check wages, we decided to try to one more model that included just the median wage from all industries. The fundamental concept behind this is that the median could capture all opportunity for potential criminals, and it has the benefit of not being affected by the outlier in wser.

RESULT: Unfortunately, though it was much better, it was still not predictive.

```r
balanced_model_top_2 <- lm(log_crmrte ~ prbarr + prbconv + prbpris
                             + avgsen + polpc + density
                             + taxpc + pctmin80 + mix + pctymle
                             + median_wage,
                             data = data_crmrte)
se.balanced_model_top_2 = sqrt(diag(vcovHC(balanced_model_top_2)))
coeftest(balanced_model_top_2, vcov = vcovHC)
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  -3.7018614   0.6241512 -5.9310 7.799e-08 ***
## prbarr       -1.9379154   0.4176289 -4.6403 1.381e-05 ***
## prbconv      -0.7650603   0.1360985 -5.6214 2.826e-07 ***
## prbpris      -0.1114352   0.4509889 -0.2471  0.805487
## avgsen       -0.0046181   0.0137443 -0.3360  0.737770
## polpc       167.9127417  85.2311288  1.9701  0.052378 .
## density       0.0977596   0.0343625  2.8450  0.005671 **
## taxpc         0.0020705   0.0056750  0.3648  0.716214
## pctmin80      0.0123893   0.0016202  7.6467 4.534e-11 ***
## mix          -0.5896970   0.5806401 -1.0156  0.312961
## pctymle       1.5670818   1.9109317  0.8201  0.414680
## median_wage   0.0011536   0.0014133  0.8162  0.416848
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(balanced_model_top_2)
```

```
## [1] 28.07462
```

```r
# stargazer(simple_regression_model, all_in_model_log_level, balanced_model_top_1, balanced_model_top_2
#           type = "text", omit.stat = "f",
#           se = list(se.simple_regression_model, se.all_in_model_log_level,
#                     se.balanced_model_top_1, se.balanced_model_top_2),
#           star.cutoffs = c(0.05, 0.01, 0.001))
```

Three of the five groups have been eliminated, with only the deterrent and demographic groups remaining. We will use step wise regression to evaluate.

```r
base_backward = lm(log_crmrte ~ prbarr + prbconv + prbpris
                             + avgsen + polpc + density
                             + taxpc + pctmin80 + mix + pctymle,
                             data = data_crmrte)
# backward_step = step(base_backward, scope = formula(base_backward), direction = "backward")
```

```r
balanced_model_top_3 <- lm(log_crmrte ~ mix + density
                           + polpc + pctmin80
                           + prbarr + prbconv,
```

```
                              data = data_crmrte)
se.balanced_model_top_3 = sqrt(diag(vcovHC(balanced_model_top_3)))
coeftest(balanced_model_top_2, vcov = vcovHC)
```

```
##
## t test of coefficients:
##
##                 Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)   -3.7018614   0.6241512 -5.9310 7.799e-08 ***
## prbarr        -1.9379154   0.4176289 -4.6403 1.381e-05 ***
## prbconv       -0.7650603   0.1360985 -5.6214 2.826e-07 ***
## prbpris       -0.1114352   0.4509889 -0.2471  0.805487
## avgsen        -0.0046181   0.0137443 -0.3360  0.737770
## polpc        167.9127417  85.2311288  1.9701  0.052378 .
## density        0.0977596   0.0343625  2.8450  0.005671 **
## taxpc          0.0020705   0.0056750  0.3648  0.716214
## pctmin80       0.0123893   0.0016202  7.6467 4.534e-11 ***
## mix           -0.5896970   0.5806401 -1.0156  0.312961
## pctymle        1.5670818   1.9109317  0.8201  0.414680
## median_wage    0.0011536   0.0014133  0.8162  0.416848
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(balanced_model_top_3, k=2)
```

```
## [1] 21.39003
```

```
# stargazer(simple_regression_model, all_in_model_log_level, balanced_model_top_1, balanced_model_top_3
#           type = "text", omit.stat = "f",
#           se = list(se.simple_regression_model, se.all_in_model_log_level,
#                     se.balanced_model_top_1, se.balanced_model_top_3),
#           star.cutoffs = c(0.05, 0.01, 0.001))
```

The difference between the backward and forward model is that the backward model chooses variables for exclusion based on comparing significance while the forward model looks for significance in inclusion. We also used the f-tests (hypothesis tests) to give the backward stepwise regression a head start.

The backward stepwise regression yielded a more reasonable model so that is the model we are choosing for our balanced model. This model strikes a nice balance between parsimony and explanatory power. The variables included are prbarr, prbconv, polpc, density, pctmin80, and mix. Six out of the original 24 independent variables are included. The adjust r-squared is only 3% lower (77.6% vs. 80.6%). It includes a blend of actionable items for the campaign in the deterrent data as well as demographic variables that perhaps can focus the campaign's efforts.

Praveen's section

3. An Assessment of the CLM Assumptions

We choose our balanced model for the complete assessment of all 6 classical linear model assumptions.

Assumptions	Ways Assumption Fails	Diagnostic	Conseq of Failed Assump	Solution
Assumption MLR.1 Linear in Parameters The model in the population can be written as $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + u,$ [3.31] where $\beta_0, \beta_1, \ldots, \beta_k$ are the unknown parameters (constants) of interest and u is an unobserved random error or disturbance term.	Fit linear model to nonlinear data	plot of observed vs. predicted values or plot of residuals vs. predicted values	Bad predictions, particularly out of range of the sample data	Apply a nonlinear transformation to either independent or dependent variables; add another regressor that is a nonlinear function of another variable; add a possibly omitted variable
84 PART 1 Regression Analysis with Cross-Sectional Data **Assumption MLR.2** Random Sampling We have a random sample of n observations, $\{(x_{i1}, x_{i2}, \ldots, x_{ik}, y_i): i = 1, 2, \ldots, n\}$, following the population model in Assumption MLR.1.	1.) Clustering (researchers can only access a limited number 2.) Autocorrelation common for time series data)	1.) Use knowledge of where data comes from 2.) Durbin Watson Statistic	1.) Observing less variation than actually exists in the population; betas still unbiased but estimates are much less precise	1.) Use clustered standard errors 2.) No simple fix for serial correlation - use time series model
Assumption MLR.3 No Perfect Collinearity In the sample (and therefore in the population), none of the independent variables is constant, and there are no exact linear relationships among the independent variables.	Extremely high or perfect multicollinearity (assumption only rules out perfect multicollinearity); often from lagged variables of another variable, a shared common time trend, or variables that capture similar phenomena	Correlation matrix (though difficult for several variables); VIF's (multicollinearity likely between 5 and 10, problem > 10)	When variables are highly correlated but not perfectly collinear, OLS works but estimates will be much less precise. R-squared may be high but t-stats are low; regression becomes sensitive to small changes in specification and adding or removing a variable changes betas a lot; you might get nonsensical coefficient signs and magnitudes; confidence intervals might be very wide	Drop redundant variables
Assumption MLR.4 Zero Conditional Mean The error u has an expected value of zero given any values of the independent variables. In other words, $E(u \mid x_1, x_2, \ldots, x_k) = 0.$ [3.36]	The error exhibits a pattern that is not in a fairly constant band around zero or it shows a pattern that results in nonzero errors for different x's	For one variable, plot residuals vs predictor (should see flat average line around zero). For multiple regression, plot residuals vs. fitted values (predicted values). Should again see a flat band or line. Also use domain knowledge on any omitted variables.	Endogeneity is a violation of zero-conditional mean and results in OLS coefficients that are biased and inconsistent. If the explanatory variables are uncorrelated with the error term they are exogenous	1.) Change the functional form (log of independent or dependent variable, x and x^2, etc.); might lose interpretability though. 2.) Adding new variables. 3.) Decide we can't meet zero conditional mean but we can meet exogeneity. If we satisfy the first three assumptions and exogeneity (Cov(x,u) = 0 for all x, dependent variables) then OLS estimators are consistent (unbiased as n->infinity)

Figure 1: MLR 1-4'

MLR.1: The model is linear in parameters (and the error term)

we haven't constrained the error term, so the model can be any joint distribution. Therefore the linear model assumption is not violated

```
balanced_model_top_3 <- lm(log_crmrte ~ mix + density
                          + polpc + pctmin80
                          + prbarr + prbconv,
                            data = data_crmrte)
```

MLR.2: Random sampling

First thing to note is that we are dealing with a single cross-section (1987) of a multi-year panel data.

Secondly this is observational data and not experiemental so perfect random sampling is hard to achieve.

CORNWELL – TRUMBULL (1994) specifically state they choose panel data because cross – section data were not able to capture the real effect of the crime rate on several independent regressors.

The authors identify that the time-series component of the panel data is able to identify specific characteristics of county heterogeneity, which is correlated with the criminal justice variables.

While the balanced model achieves high level of statistical significance for the co-efficients, it's important to be mindful of the limitations of the dataset.

```
se.balanced_model_top_3 = sqrt(diag(vcovHC(balanced_model_top_3)))
coeftest(balanced_model_top_3, vcov = vcovHC)
```

```
##
## t test of coefficients:
##
##                Estimate  Std. Error  t value  Pr(>|t|)
## (Intercept)  -3.1966143   0.2371960 -13.4767 < 2.2e-16 ***
## mix          -0.7447751   0.4742158  -1.5705  0.120094
## density       0.1134850   0.0265199   4.2792 4.997e-05 ***
## polpc       190.5494683  71.9365456   2.6489  0.009666 **
## pctmin80      0.0127752   0.0014745   8.6643 3.067e-13 ***
## prbarr       -2.0998396   0.4356245  -4.8203 6.398e-06 ***
## prbconv      -0.8094922   0.1261063  -6.4191 8.051e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

MLR.3: No perfect multicollinearity

Multicollinearity refers to a situation in which two or more explanatory variables in a multiple regression model are highly linearly related. We have perfect multicollinearity if, for example as in the equation above, the correlation between two independent variables is equal to 100% or negative 100%.

As seen from the correlation matrix below, there is no perfect multicollinearity in the model but we observe some meaningful correlations between (Prbarr, mix) and (Prbarr,density)

These linear realtionships among the X's don't invlaidate the MLR parameters but they lower precision and increase the std-errors in the mdoel

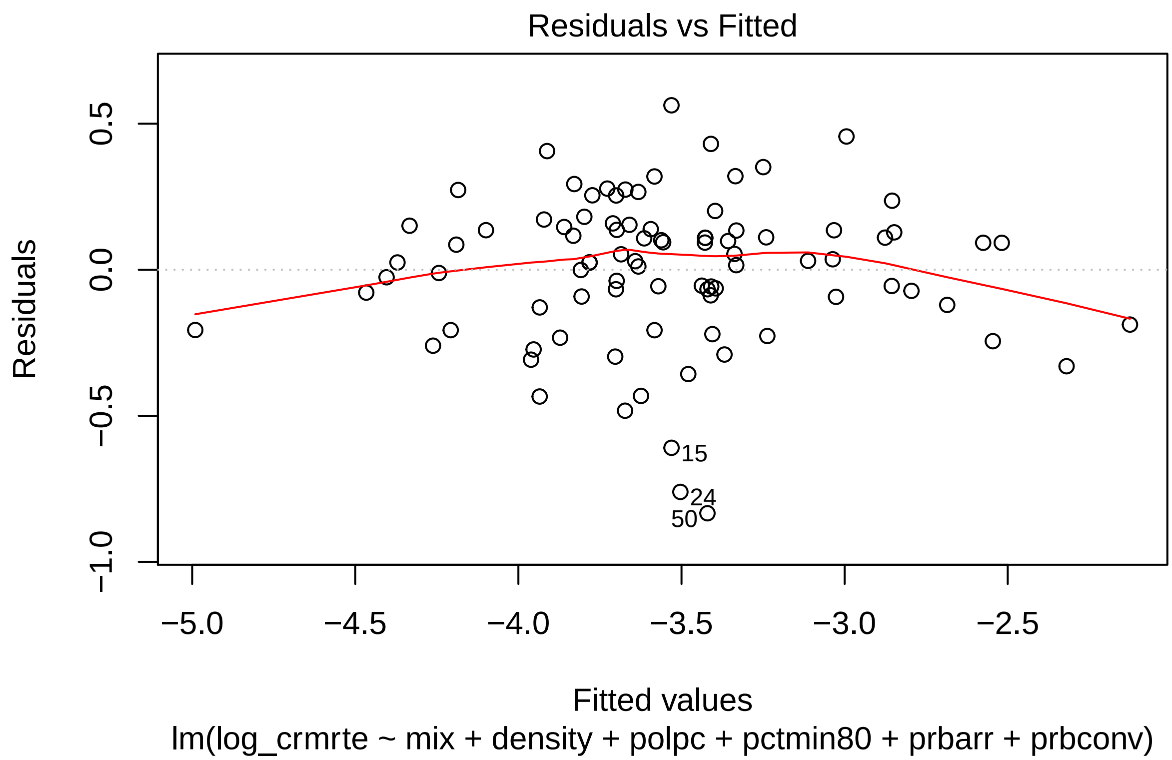
```
balanced_model <- c( "mix", "density","polpc", "pctmin80","prbarr","prbconv")
balanced_model_data <- data_crmrte[balanced_model]
round(cor(balanced_model_data)*100,2) # correlations displayed as % for convenience
```

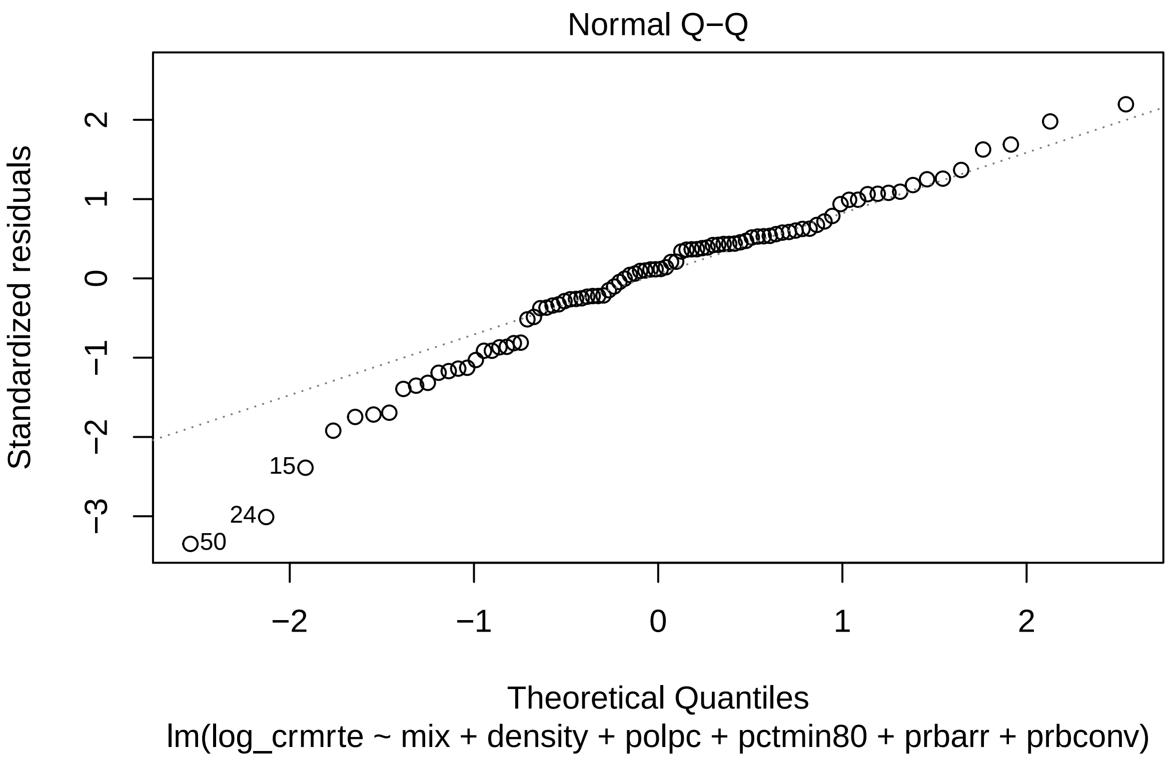
```
##              mix density  polpc pctmin80 prbarr prbconv
## mix       100.00  -13.69   2.41    20.12  41.29  -30.43
## density   -13.69  100.00  15.91    -7.46 -30.27  -22.67
## polpc       2.41   15.91 100.00   -16.91  42.60   17.19
## pctmin80   20.12   -7.46 -16.91   100.00   4.91    6.25
## prbarr     41.29  -30.27  42.60     4.91 100.00   -5.58
## prbconv   -30.43  -22.67  17.19     6.25  -5.58  100.00
```

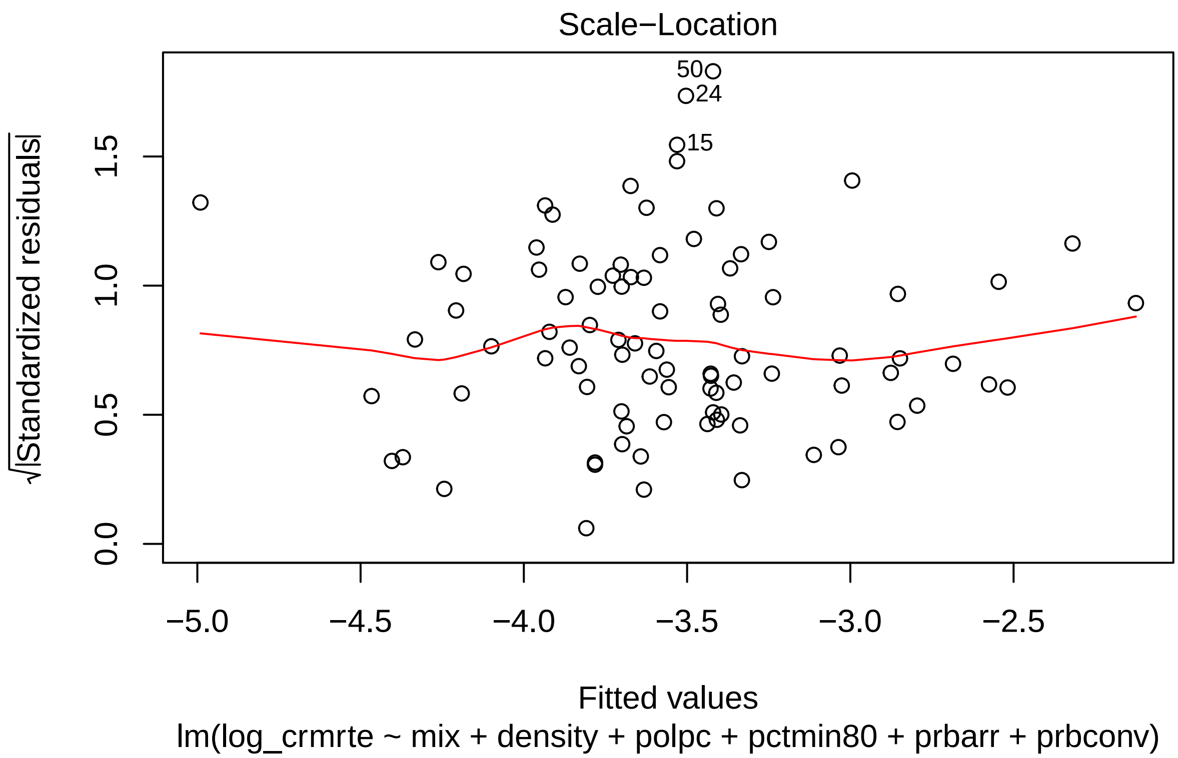
MLR.4: Zero Conditional Mean / exogeneity

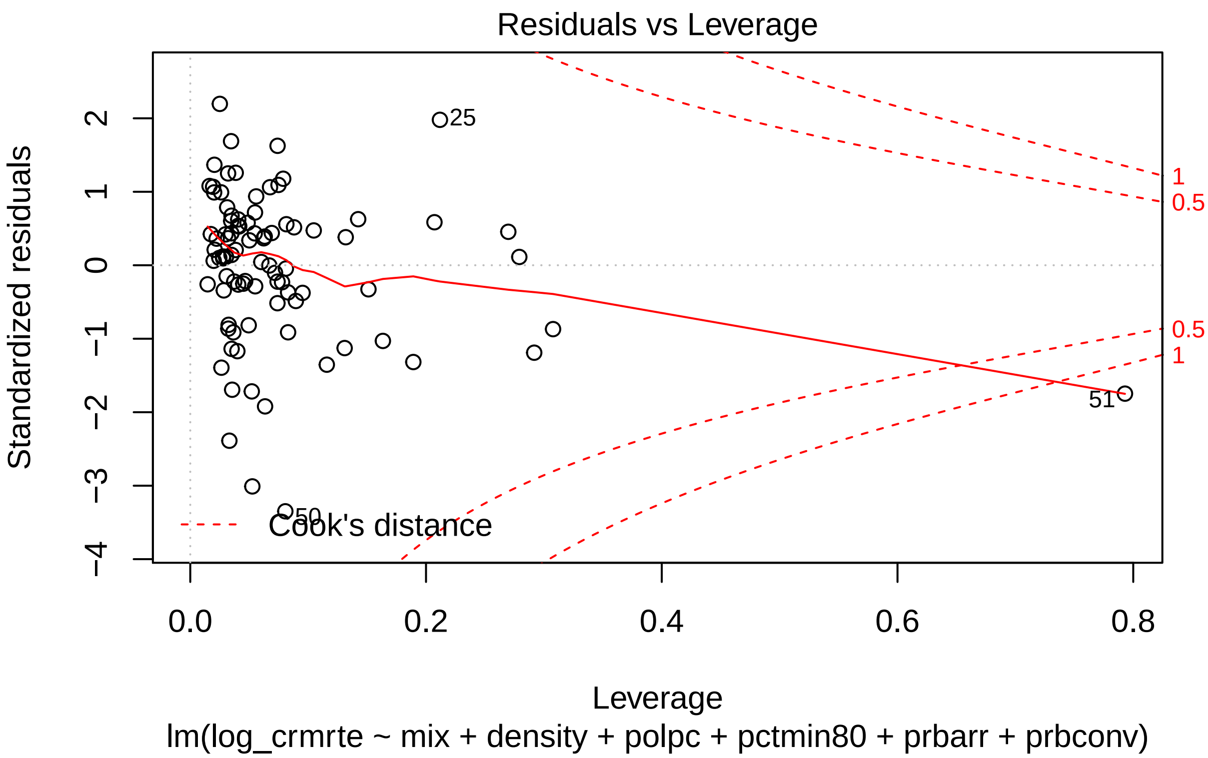
ZCM is best analysed by studying the regression plots of the residuals. Let's start by looking at the regression plots of the balanced model

```
plot(balanced_model_top_3)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(log_crmrte ~ mix + density + polpc + pctmin80 + prbarr + prbconv)
50
24
15
−3
−2
−1
0
1
2
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
50
24
15
−5.0
−4.5
−4.0
−3.5
−3.0
−2.5
Fitted values
lm(log_crmrte ~ mix + density + polpc + pctmin80 + prbarr + prbconv)



CLM assumptions analysis from plots

- Plot 1. The residuals vs. fitted plot indicates that the zero conditional mean assumpotion is NOT perfectly satisfied but the red line is close enough to zero, a big improvement compared to some of the other models we tested. The non-uniform thickness of the residuals indicates possible hetroskedasticity.
- Plot 2. The Q-Q plot shows that the residuals are not perfectly normally distributed, but the log transform of the crime rate improved the +ve skew in the data but has introduced some negative skew
- Plot 3. The scale location plot indicated the presence of hetroskedasticity especially in the middle where the thickness of the band varies and outliers such as '50' and '24' are generating large standardized residuals
- Plot 4. The residuals vs leverage plot shows some of the outliers we had discussed earlier (51, 25, 84) but the most significantly outlier is 51 (having high leverage and Cook's distance >1). This outlier 51 significantly affects our model estimte and might be worth removing form the data to improve model accuracy.

```
round(cor(balanced_model_top_3$residuals, balanced_model_data)*100,5)
```

```
##      mix density polpc pctmin80 prbarr prbconv
## [1,]   0       0     0        0      0       0
```

Finally we check the correlation between the X's and the errors in the model to ensure there is no endogeneity in the model. There is a more extensive discussion of omitted variable and their implication on model endogeniety in section 5 of the report.

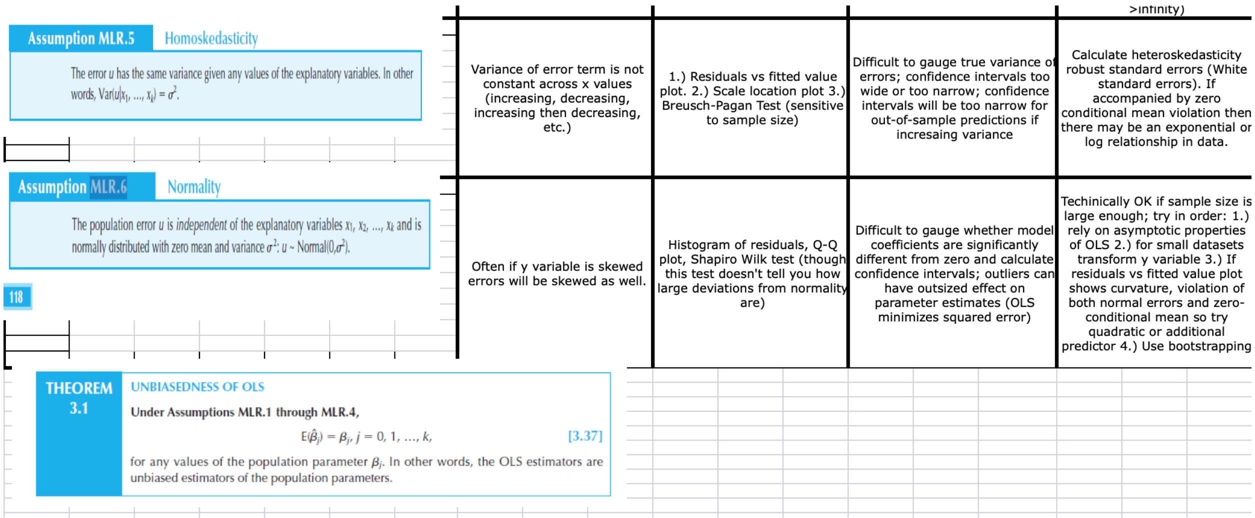


Figure 2: MLR 5 & 6'

MLR.5: Homoskedasticity

Homoscedasticity describes a situation in which the error term has the same variance across all values of the independent variables.

The regression plots indicate the presence of some hetreskedasticity in the erros let's test if they are statistically significant using the Breusch-Pagan test.

- The Breusch-Pagan test below allows us to test for hetroskedasticity under the

$H_0 : Thereishomeskedasticity$

```
bptest(balanced_model_top_3)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  balanced_model_top_3
## BP = 8.6648, df = 6, p-value = 0.1933
```

From the BP test, surprisingly we find p-value is not statistically significant, therefore we fail to reject $H_0 : Thereishomeskedasticity$

However, we will still choose to be more conservative and use HC consistent std-errors (Huber-white Std-errors) using coeftest function from the sandwich package in R. This conservative approach we have taken througout this report in our model selection process in choosing regressors for different models

MLR.6: Normality of the error term

Often, if the Y variable is skewed, the error terms will be skewed as well.

We can check the normality using the Q-Q plot to vizualize the distribution of residuals.

We saw in the earlier section that the crime rate has some positive skew, but we were able to reduce the skew by applying log transform to the crime rate.

We can also run a Shapiro - wilk test for normality $H_0 : Errorsarenormal$

```
shapiro.test(balanced_model_top_3$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  balanced_model_top_3$residuals
## W = 0.96118, p-value = 0.008754
```

The p-value is significant, therefore we reject $H_0 : Errorsarenormal$

The non-normality of the residuals is statisitically significant for this model.

There is some negative skew from outlier 51 in the transformed variable but However since we have n>30 under CLT we have OLS estimators are normally distributed.

4. A Regression Table

The results were displayed in stargazer using HC standard errors as part of model selction

- This sections has been fully covered under section 2 of the report
- We hav include statistical F-tests besides the standard t-tests for regression coefficients to check model validity.
- Additionally the practical significance of the model variable chosen have also been discussed in detail
- Below is the summary of the regression models and the AIC & BIC scores which provides a parsimony adjusted measure of fit

```
stargazer(simple_regression_model, all_in_model_log_level, balanced_model_top_1, balanced_model_top_3,
          type = "text", omit.stat = "f",
          se = list(se.simple_regression_model, se.all_in_model_log_level,
                    se.balanced_model_top_1, se.balanced_model_top_3),
          star.cutoffs = c(0.05, 0.01, 0.001))
```

```
##
## ===================================================================================
##                                           Dependent variable:
##                       -------------------------------------------------------------
##                                               log_crmrte
##                             (1)             (2)             (3)             (4)
## -----------------------------------------------------------------------------------
## prbarr                                   -1.889***       -1.963***       -2.100***
##                                           (0.380)         (0.401)         (0.436)
##
## prbconv                                  -0.656***       -0.767***       -0.809***
##                                           (0.174)         (0.137)         (0.126)
##
## prbpris                                   -0.093          -0.076
##                                           (0.399)         (0.473)
##
## avgsen                                    -0.008          -0.004
##                                           (0.016)         (0.014)
```

```
##
## polpc                                   154.835        176.135*        190.549**
##                                        (86.523)        (82.588)        (71.937)
##
## density                0.228***         0.117*          0.114**        0.113***
##                         (0.030)         (0.054)         (0.035)         (0.027)
##
## taxpc                                    0.003           0.002
##                                         (0.007)         (0.006)
##
## west                                    -0.115
##                                         (0.125)
##
## central                                 -0.101
##                                         (0.092)
##
## urban                                   -0.169
##                                         (0.229)
##
## pctmin80                                0.010**        0.013***        0.013***
##                                         (0.003)         (0.002)         (0.001)
##
## wcon                                    0.0005
##                                         (0.001)
##
## wtuc                                    0.0001
##                                         (0.001)
##
## wtrd                                    0.0003
##                                         (0.002)
##
## wfir                                    -0.001
##                                         (0.001)
##
## wser                                    -0.0001
##                                         (0.002)
##
## wmfg                                    -0.0002
##                                         (0.001)
##
## wfed                                    0.002*
##                                         (0.001)
##
## wsta                                    -0.001
##                                         (0.001)
##
## wloc                                     0.001
##                                         (0.002)
##
## mix                                     -0.239          -0.730          -0.745
##                                         (0.626)         (0.540)         (0.474)
##
## pctymle                                  2.771           1.383
##                                         (1.433)         (1.621)
```

```
##
## Constant                 -3.869***       -4.026***       -3.352***       -3.197***
##                           (0.069)         (0.848)         (0.356)         (0.237)
##
## ------------------------------------------------------------------------------------
## Observations                90              90              90              90
## R2                         0.401           0.854           0.796           0.791
## Adjusted R2                0.394           0.806           0.770           0.776
## Residual Std. Error 0.427 (df = 88) 0.242 (df = 67) 0.263 (df = 79) 0.260 (df = 83)
## ====================================================================================
## Note:                                                *p<0.05; **p<0.01; ***p<0.001
```

Parismony adjusted model performance

Though AIC and BIC are both Maximum Likelihood estimate driven and penalize free parameters in an effort to combat overfitting, they do so in ways that result in significantly different behavior. Lets look at one commonly presented version of the methods (which results form stipulating normally distributed errors and other well behaving assumptions):

AIC = -2*ln(likelihood) + 2*k, and BIC = -2*ln(likelihood) + ln(N)*k,

where: k = model degrees of freedom (K=2 is default for OLS) N = number of observations

The quick explanation is:

- AIC is best for prediction as it is asymptotically equivalent to cross-validation.
- BIC is best for explanation as it is allows consistent estimation of the underlying data generating process.

When N is large the two models will produce quite different results. Then the BIC applies a much larger penalty for complex models, and hence will lead to simpler models than AIC for very large N.

So we check both IC for our model and in both cases a lower value implies a better parsimony adjusted outcome.

```r
AIC(simple_regression_model)
```

```
## [1] 106.2991
```

```r
AIC(all_in_model_log_level)
```

```
## [1] 21.354
```

```r
AIC(balanced_model_top_1)
```

```
## [1] 27.4514
```

```r
AIC(balanced_model_top_3)
```

```
## [1] 21.39003
```

```r
BIC(simple_regression_model)
```

```
## [1] 113.7985
```

```r
BIC(all_in_model_log_level)
```

```
## [1] 81.34943
```

```r
BIC(balanced_model_top_1)
```

```
## [1] 57.44912
```

```r
BIC(balanced_model_top_3)
```

```
## [1] 41.38851
```

Josh's section

5. A Discussion of Omitted Variables

We've identified several key omitted variables that we feel most influence the crime rate but are not represented in the data here.

1. Unemployment Rate - Unemployment is a key indicator for crime rate. We may be able to infer some indication of the frequency of seasonal or part-time work in the construction or service industries from the `wcon` or `wser` variables as they shows an average weekly wage which mght indicate how often workers are employed. However, this estimate is likely not accurate enough to be considered meaningful. The unemployment rate among youth 18-30 would also be meaningful as criminal activity among young adults is higher than that of older adults.

2. Inflation Rate (Consumer Price Index) - Inflation and crime rates are correlated with a positive relationship and the causal link is from inflation and unemployment to crime. Link. Inflation causes the purchasing power to reduce and cost of living to increase, consequently crime rates rise as the inflation rate rises. Because of the lag between price and wage adjustments, inflation lowers the real income of low-skilled labor, but rewards property criminals due to the rising demand and subsequent high profits in the illegal market. Inflation in the year represented, 1987, would not be sufficient though as the reduction in purchasing power does not happen immediately, it takes time for inflation to gradually reduce purchasing power. None of the data provided in the study gives us an indication of the inflation rate in a time period before the study. We would expect that this variable would show a positive bias towards crime rate and that it would likely be a large bias.

3. Childhood Blood Lead Levels (with 18 year offset) - The lead–crime hypothesis is the proposed link between elevated blood lead levels in children and increased rates of crime, delinquency, and recidivism later in life. Studies linking blood lead levels (BLL) in children to crime rate typically seek to quantify the BLL 17-18 years before the examined crime rate. One such study used a unique dataset linking preschool blood lead levels (BLLs), birth, school, and detention data for 120,000 children born 1990-2004 in Rhode Island, to estimate the impact of lead on behavior Link. We expect that this variable would show a positive bias and that it would likely be a small bias but still significant for any given year as there may be other underlying phenomena driving crime rate in a particular county. There are no variables in the provided data set that would give any insight into this.

4. Abortion Rates (with 18 year time lag) - Multiple studies have shown a correlation between legalized abortion rates and crime. One study by Donohoe and Leavitt estimated that crime fell roughly 20% between 1997 and 2014 due to legalized abortion. Link While it may be difficult to ascertain which counties residents accessing abortion services lived in, we expect that measures of employment and poverty could be correlated to show how a negative bias of abortion rates potentially offset other variables with a positive bias. We estimate that the bias may be small as it could present difficulties in localizing it effectively, but we still believe that it would be significant. There are no variables in the provided data set that would give any insight into this.

5. Income Inequality metrics: There are several measures of income inequaity that could be included in the data: Mean Log Deviation or Theil Index or Gini Index for each of the counties. Income inequality has been shown to have a significant effect on violent crime in particular. One World Bank report states that inequality predicts about half of the variance in murder rates between American states and between countries around the world. Link Income inequality measures are often measured as 0 (perfectly equal income distribution) to 1 (perfectly unequal income distribution, or 1 household has all the income). We would thus expect these to have a positive bias, in that an increase in income inequality would lead to an increase in violent crime. We expect that the bias would be somewhat smaller as income inequality is correlated specifically with violent crime less than property crime. There are no variables in the provided data set that would give any insight into this.

Ryan's section

6. Conclusion

Make sure that you end your report with a discussion that relates your results to concerns of the political campaign.