Lab 3

W203 Statistics for Data Science

Section 6 - Team 1

QUESTIONS TO ASK 1. Should we have pick variables and explain why we chose then, and perform regression on that set? Or, should do stepwise regression on all variables, let the model pick the best ones, and explain why those might be the best ones? 2. When are transformations most useful? Should we use background knowledge, or use residual information to diagnose which is an outlier? 3. Imputing variables: for missing values, take median for example; don't do this ##

Paul's OH: Measure ideally causal Take natural logs Transformations: when to transform variables? Probability greater than 1 is ok; not really probability due to collection process Quantile regression for conditional median (OLS is the conditional mean) ###

Model 1a: things that make people afrad (4 probability variables) that deters crime,

For these three models: %location could be important %density/urban absorbs too much causality %male absorbs the causality because then it will only reflect that %police also absorbs too much because people more police means more crime is detected. %tax could reflect how people vote; see paper %demographics (minority, male) certainly increase prediction but even playing ground first (good moderator variable, but not mediator variable) %mix shouldn't have too much effect.

Model 2: ability for people to succeed without having to commit crime. Add in wage to improve prediction. However, use BACKWARDS AIC to optimize the AIC of the model in order to achieve parsimony.

Model 3: demographics added in; tax added in for attitude ###

Strategy: 1. eliminate variables (year, county) 2. look at categorical variables (urban, west, central)

3. Urban and west very important (density directly related to this, so won't consider it); central is not; HOW TO DEAL WITH THESE? regress on them directly? Or separate them out into two different groups?

4. Regression of each variable on crime rate; do outlier detection (Cook's distance) and deal with outliers using domain knowledge

OUTLIERS!

WHICH APPROACH? Two approaches: 1. hand pick variables and run regression

2. Run AIC minimization (multiple times) and explain why the main variables may come about. train function in caret. NEED TO DO EDA on each variable. Check that standarized residuals fall within 1 STD. Outlier detection with Cook's distance.

BALANCE BETWEEN THROWING AWAY HIGHLY CORRELATED VARIABLES and KEEPING THEM.

WHEN TO TRANSFORM VARIABLES?

7. train/dev/test? on model generalization? 1000 fold validation?

Three models:

Model 1: KEY interests – should not perform very well; density, prbarr, police Model 2: BEST model; add in wage variables, demographics, others? We each need to experiment until we find the best model. Then we will reason. Model 3: Show robust to changes. Adding new variables do not influence prediction anymore.

START OF REPORT BELOW:

Introduction

Initial Data Cleaning

First, we need to omit the last rows of the csv since they do not contain data. When we check for duplicated rows, there was one county (193) which was repeated twice. We also checked that all of the values were numerical. This was not the case for the prbconv variable (it was stored as a factor due to omitted rows having non-numerical values). This was converted to numerical.

```
library(dplyr)
library(ggplot2)
library(tidyr)
library(caret)
library(MASS)
select <- dplyr::select #unmask select from dplyr
library(stargazer)
library(tibble)

setwd("~/Desktop/Stone/Berkeley_MIDS/Statistics/Labs/Lab_3")
full_data <- read.csv('crime_v2.csv')
data <- na.omit(full_data)

#check for duplicated data and remove
sum(duplicated(data))
```

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

```
data <- distinct(data, .keep_all=T)

#convert prbconv factor in numeric
data$prbconv <- as.numeric(levels(data$prbconv))[data$prbconv]

#check that all fields are numerical
for (field in names(data)) {
  stopifnot(class(data[,field]) %in% c("numeric", "integer"))
  }
```

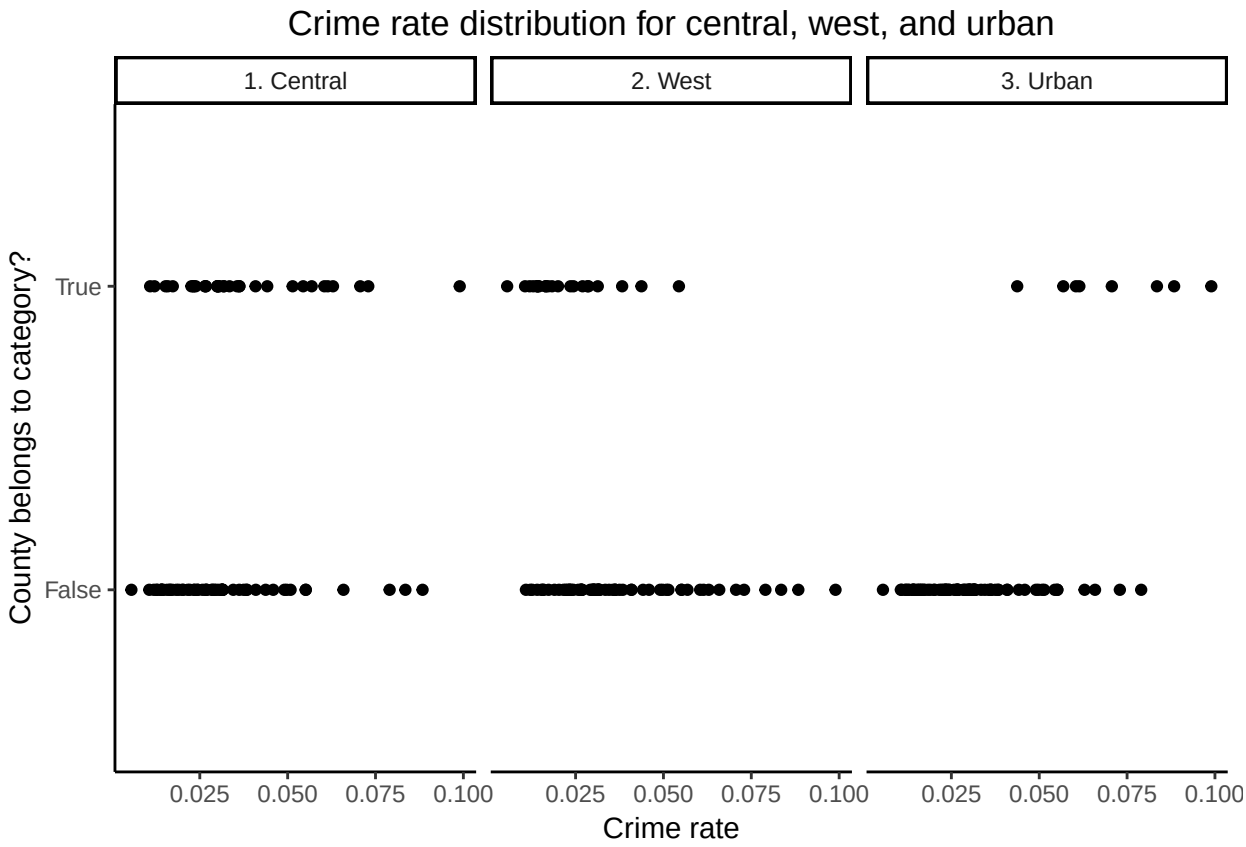
Next, our independent variable of interest is crime rate (crmrte). Our goal is to find the best possible causal predictors for crime rate. We first identify columns for which believe may be possible candidates.

County and year are identifiers and are not relevant to the study. West/Central/Urban are categorical variables which have been one-hot encoded. These can be used directly in the regression. We can first see whether the distribution of crime rate is different depending on the location (west vs central) and whether the county is urban.

```
C <- data[, c('west', 'central', 'urban')]

df.categorical <- data[, c('crmrte', 'west', 'central', 'urban')]
colnames(df.categorical) <- c('crmrte', '2. West', '1. Central', '3. Urban')
dt_long <- gather(df.categorical, key, value, -crmrte)

ggplot(dt_long, aes(x = crmrte, y = factor(value))) +
  geom_point() +
  facet_grid(. ~ key) +
  theme_classic()+
  theme(plot.title = element_text(hjust = 0.5))+
  labs(title="Crime rate distribution for central, west, and urban",
       x='Crime rate',
       y = "County belongs to category?") +
  scale_y_discrete(breaks=c(0,1), labels=c("False","True"))
```



We can see that for Central versus not Central N.C., the crime rate distribution is relatively even. Counties in Western N.C. appear to have less crime on average than those labeled as not Western. Counties labeled as Urban have more crime on average than those not. We see definitive summaries below for the urban and west variables below.

```
u <- data %>%
  group_by(urban) %>%
  summarise(mean_crime_rate = mean(crmrte)) %>%
  as.data.frame()
u$urban <- c('N', 'Y')
w <- data %>%
```

```
  group_by(west) %>%
  summarise(mean_crime_rate = mean(crmrte)) %>%
  as.data.frame()
w$west <- c('N', 'Y')
print(u)
```

```
##   urban mean_crime_rate
## 1     N      0.02990170
## 2     Y      0.07049427
```

```
print(w)
```

```
##   west mean_crime_rate
## 1    N      0.03720145
## 2    Y      0.02209975
```

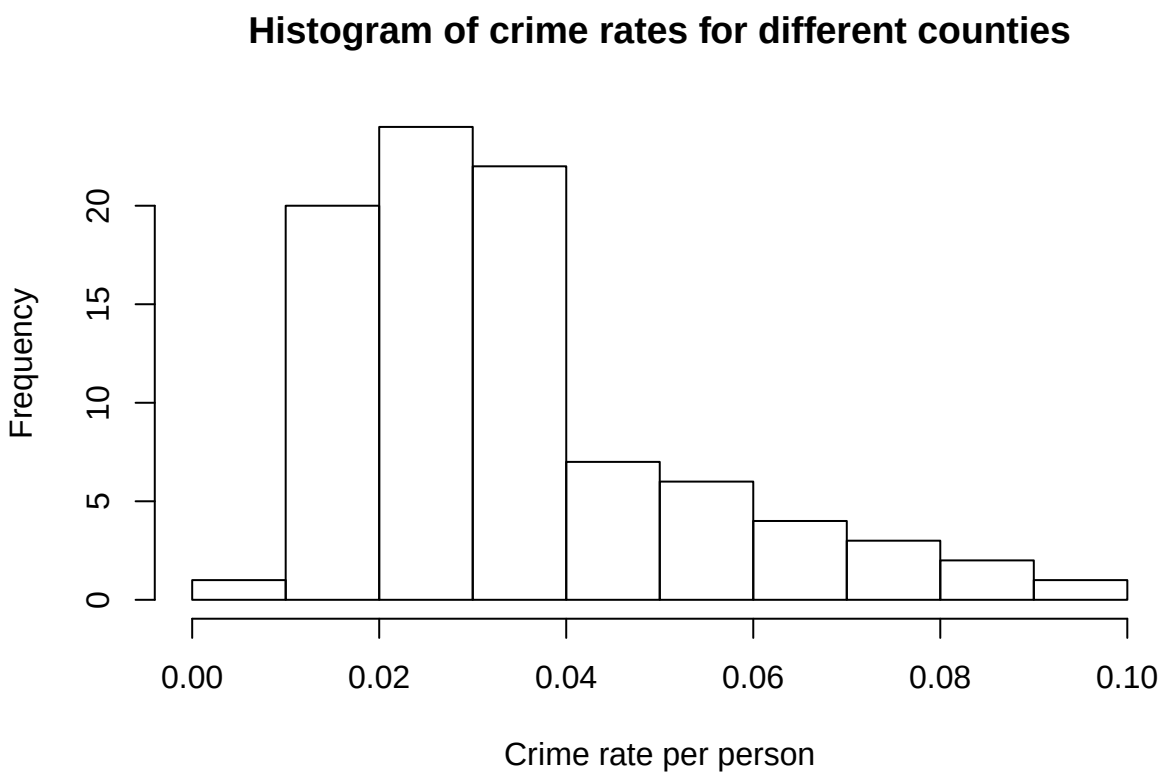
All other variables are numeric, and are also possible candidates for influencing crime rate.

We now examine the crime rate variable on its own.

```
summary(data$crmrte)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.005533 0.020604 0.030002 0.033510 0.040249 0.098966
```

```
hist(data$crmrte,
     main='Histogram of crime rates for different counties',
     xlab = 'Crime rate per person')
```

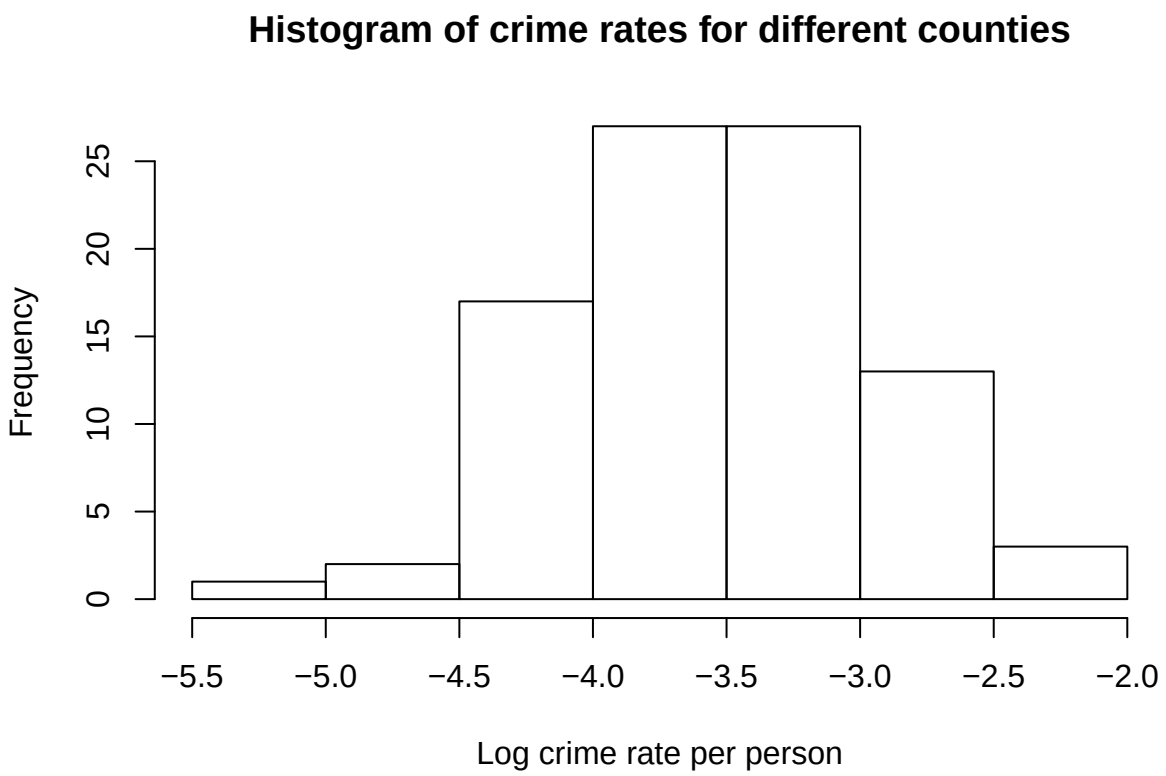


We can see that crime rate is greater than 0 for all counties. We can also see from this plot that crime rate is highly skewed toward larger values. Since we are performing inference analysis, we would like to interpret our model coefficients as how changes in explanatory variables affecs changes in crime rate. However, since

the baseline crime is different for different counties, it makes more sense to transform crime rate into the log of crime rate This changes the interpretation from absolute changes in crime rates to percent changes (at least for small changes in crime rate since the percent interpretation is only accurate for differentially small changes), which makes comparisons across counties more comparable. For example, a 0.01 change in crime rate for the lowest county (0.005) is a much larger change than for the largest county (0.099), but a 1% change is comparable. As a result, we perform inference on the log of crime rate.

We can visualize the transformed crime rate variable again.

```
data$crmrte_abs <- data$crmrte
data$crmrte <- log(data$crmrte)
hist(data$crmrte,
     main='Histogram of crime rates for different counties',
     xlab = 'Log crime rate per person')
```



explain and generate boxplot

Model 1

Key Variables

For our initial model, we would like to focus on factors that intuition says should influence crime. We believe that there are four variables which represent deterrents to crime: probability variables of arrest, conviction, prison sentence, and the severity of punishment in average sentence days. We will call these variables the "fear factors;" namely, we believe the higher the chance someone believes they will be arrested, convicted, or sent to prison, the less likely they will commit a crime. Also, the more severely they believe the punishment to be (prison day sentences), the less likely they will commit a crime. Out of these four, we believe probability of arrest and probability of conviction will have the greatest effects. The reason is that a single arrest or convinction can permanently damage someone's record. For most people who have never committed crimes before, just the idea of possibly getting in trouble with the police could be enough to deter them. In addition,

there are many crimes that result in fines, community service, and other forms of punishment that does not involve prison. Many criminals are likely not thinking about possibility or severity of prison sentences because they might feel even if arrested they can talk their way out of it. For heavy repeat offenders, they will likely prison sentence into account, but possibility of arrest is still a heavy influencer. As a result, we believe arrest and convinction are the most relevant variables.

The wage variables can either deter or motivate individuals to commit a crime and were excluded from this base model. We believe that the more satisfied someone is with their income, the less likely they will commit a crime because they are more likely to attain their desires without having to pursue illegal routes. Along the same lines, unemployment is likely to lead to increased crime rates. Too high of a wage, especially in blue collar jobs, means me employees can be "priced out" some employees. As wage goes up, individuals paid that wage are expected to do more, lowering the amount of workforce necessary, leading to greater unemployment. As a result, we believe wage can go either way. For our base model, we will look at only what we consider traditional blue collar jobs: construction and manufacturing. We also take the log of these variables: this is common practice as we want to measure percent changes in salary, and not absolute changes.

Before performing EDA on the variables listed above, we note why we have chose to exclude the other variables in our base model:

We believe that density should be a positive predictor of crime. Highly dense population areas present more opportunities for crime. There also tends to be a larger wage and wealth gap in these areas, which increases the rate of crime as people will be tantalized to use illegal ways to get to the top. However, this may absorb too much of the causal effect.

We also believe that the police per capita is a key variable that would absorb too much of the model. Namely, more police are required for regions of greater crime, and so counties with more crime are more likely to have more police. In addition, the fact that there's more police could mean that more crime is detected and responded to, increasing the recorded number of criminal cases and perceived rate of crime. However, we also expect there to be a tipping point. If the density of police is extremely high, that likely acts as a major deterrent for criminals. As a result, police and crime rate are very intricately linked and want to avoid this for our base model.

Location could be important. Different geographic areas may be more prone to crime due to cultural and socioeconomic differences. However, we would like a model that can be generalized.

Male absorbs the causality because then it will only reflect that.

Tax could reflect how people vote; see paper

Demographics (minority, male) certainly increase prediction but even playing ground first (good moderator variable, but not mediator variable)

Mix shouldn't have too much effect.

Explain above further

```
#function for outlier detection
#we will define outliers as 1.5*IQR above the 1st quartiler or 1.5*IQR below the third
outlier_detector <- function(df, outlier_field) {
  iqr <- IQR(data[,outlier_field])
  upper <- quantile(df[, outlier_field], 0.75)
  lower <- quantile(df[, outlier_field], 0.25)
  mask <- (df[, outlier_field] > (upper + 1.5 * iqr)) | (df[, outlier_field] < (lower - 1.5 * iqr))
  return (df[mask, ])
}
```

Univariate analysis of each of the 4 variables

```
#first model
model_1 <- lm(crmrte ~ prbarr + prbconv + log(wcon) + log(wmfg), data = data)
print(model_1)
```

```
##
## Call:
## lm(formula = crmrte ~ prbarr + prbconv + log(wcon) + log(wmfg),
##     data = data)
##
## Coefficients:
## (Intercept)       prbarr      prbconv    log(wcon)    log(wmfg)
##     -8.4338      -1.6815      -0.7070       0.4696       0.5408
```

Coefficient interpretation

For our first model, the coefficient in density represents the percent change in crime per person per change in people per square mile, while keeping the other three variables constant. This is positive as expected, meaning that more density locations tend to have more crime. The other interpretations are analogous: prbarr is negative, meaning the greater the chance someone gets arrested, the less likely they will commit a crime. FINISH THIS UP. Interestingly, the wage variables appears to have played very little role with very small coefficients.

Model 2

model_2 <- lm(formula = crmrte ~ prbarr + prbconv + polpc + pctmin80 + log(wcon) + log(wmfg) + log(wfed) + log(wsta), data=data)

For model 2, we wanted to add in other covariates meant to increase accuracy of prediction. To do this, we wanted to first get a sense of crmrte correlation with all numeric variables. We also parse our data into X (numeric variables) and y (crmrte)

```
y <- data$crmrte
X <- data[,!names(data) %in% c('county',
                               'year',
                               'crmrte',
                               'density',
                               'west',
                               'central',
                               'urban')]


#correlate all variables and store in new dataframe
cor_df <- data.frame(variable = character(),
                     crmrte_cor = numeric())
for (x in names(X)) {
  crmrte_cor <- cor(y, data[,x])
  corr <- as.data.frame(crmrte_cor,
                        col.names = c('crmrte_cor')) %>%
                        add_column(variable = x, .before = 1)
  cor_df <- rbind(cor_df, corr)
```

```
}

cor_df <- arrange(cor_df, desc(crmrte_cor))
print(cor_df)
```

```
##      variable  crmrte_cor
## 1  crmrte_abs  0.94154646
## 2        wfed  0.52330585
## 3        wtrd  0.39379240
## 4        wcon  0.39371486
## 5       taxpc  0.35832339
## 6        wmfg  0.30753731
## 7        wfir  0.29324265
## 8        wloc  0.28856678
## 9     pctymle  0.27815466
## 10   pctmin80  0.23291821
## 11       wtuc  0.20146493
## 12       wsta  0.16970208
## 13    prbpris  0.02147024
## 14      polpc  0.01040580
## 15     avgsen -0.04936931
## 16       wser -0.11312801
## 17        mix -0.12473445
## 18    prbconv -0.44681361
## 19     prbarr -0.47276691
```

We took a closer look at wage variables:

```
nonwage_variables <- c('prbarr', 'prbconv', 'prbpris', 'avgsen',
                       'polpc', 'density', 'taxpc',
                       'pctymle', 'pctmin80', 'mix',
                       'urban', 'central', 'west')

wage_variables <- c('wtrd', 'wfir', 'wser', 'wfed', 'wsta', 'wloc',
                    'wcon', 'wtuc', 'wmfg')

X_non_wage <- data[, names(data) %in% nonwage_variables]
X_wage <- lapply(data[, names(data) %in% wage_variables], log)

X_wage_transformed <- cbind(X_non_wage, X_wage)

print(cor(as.data.frame(X_wage)))
```

```
##             wcon       wtuc        wtrd      wfir      wser       wmfg
## wcon  1.00000000  0.4008940  0.52846367 0.4356398 0.2126332 0.36664528
## wtuc  0.40089396  1.0000000  0.34494475 0.3110378 0.1717896 0.46370470
## wtrd  0.52846367  0.3449447  1.00000000 0.6399048 0.1822485 0.41394680
## wfir  0.43563981  0.3110378  0.63990481 1.0000000 0.2290153 0.53158146
## wser  0.21263319  0.1717896  0.18224850 0.2290153 1.0000000 0.24085776
## wmfg  0.36664528  0.4637047  0.41394680 0.5315815 0.2408578 1.00000000
## wfed  0.47612863  0.3858648  0.63075894 0.5940578 0.2545001 0.55748968
## wsta -0.03908865 -0.1763056 -0.01426043 0.2330851 0.0268477 0.02256164
## wloc  0.50083967  0.3040954  0.57419433 0.5134441 0.3138729 0.46671058
##           wfed        wsta      wloc
## wcon 0.4761286 -0.03908865 0.5008397
```

```
## wtuc 0.3858648 -0.17630564 0.3040954
## wtrd 0.6307589 -0.01426043 0.5741943
## wfir 0.5940578  0.23308507 0.5134441
## wser 0.2545001  0.02684770 0.3138729
## wmfg 0.5574897  0.02256164 0.4667106
## wfed 1.0000000  0.14065703 0.4970107
## wsta 0.1406570  1.00000000 0.1496427
## wloc 0.4970107  0.14964266 1.0000000
```

Interestingly, wsta is not correlated with anything very strongly, so that's a good strong independent predictor. wfed is the largest single univariate predictor. Both are government jobs with the potential to influence change. Explain why these might influence crime rates. Add back some government, white-collar jobs.

Also add back police. Reason might be more police there is more arrests going to be made.

Add back demographics back. Unfortunately certain people are more likely to be tagged for crimes. Young male won't be used because crime is specified. Young male can't commit some of the white collar crimes for example. If the data was only petty theft then maybe we would consider it.

Model 3

Adding back density no longer influences the adjusted r squared value! The model_2 is already very robust. 23 predictors increase the adjusted r squared by very little compared to using just 8.

MAKE SURE ALSO THAT MODEL 2's coefficients don't change that much compared to model 3

```
#data.model3
```

other useful discussions

Other variables include the probability variables of arrest, conviction, prison sentence, as well as the severity of punishment in average sentence days. We will call these variables the "fear factors;" namely, we believe the higher the chance someone believes they will be arrested, convicted, or sent to prison, the less likely they will commit a crime. Also, the more severely they believe the punishment to be (prison day sentences), the less likely they will commit a crime.

We need to include more reasons why from the original research paper

First, we look at our data for probability variables, and average sentence days.

```
fear_factors <- c('prbarr', 'prbconv', 'prbpris', 'avgsen')
stargazer(data[, fear_factors], type='text',
          summary.stat = c("min", "p25", "median", "p75", "max", "median", "sd"))
```

```
##
## ==========================================================
## Statistic  Min  Pctl(25) Median Pctl(75)  Max   Median St. Dev.
## ----------------------------------------------------------
## prbarr    0.093  0.205   0.271   0.345   1.091  0.271   0.138
## prbconv   0.068  0.344   0.452   0.585   2.121  0.452   0.354
```

```
## prbpris   0.150  0.364   0.422   0.458   0.600  0.422   0.081
## avgsen    5.380  7.375   9.110   11.465  20.700 9.110   2.834
## ------------------------------------------------------------
```

Interestingly, we see that the probability of arrest and conviction variables can both exceed 1. Upon doing some external research, we believe this is ok.

PREVIOUS VERSION

need to perform automated residual detection

In order to detect outliers in our data that may greatly influence our prediction model, we will perform pairwise correlation for all numerical variables with the crimerate, and determine the Cook's distance for all data points in the regression. Any variable for which there contains one more data points with a Cook's distance greater than or equal to 1 are typically considered influential outliers. These will all be examined closely as part of our EDA.

```
cooks.outliers <- data.frame(cor_variable = character(),
                  distance = c(),
                  county = c())

for (var in names(X)) {
  mod <- lm(y ~ X[, var])
  cooks.distances <- cooks.distance(mod)
  influential_outliers <- names(cooks.distances)[cooks.distances >= 1]

  if (length(influential_outliers > 1)) {
  cooks.outliers <- rbind(cooks.outliers, data.frame(cor_variable = var,
                                    distance = cooks.distances[influential_outliers],
                                    county = influential_outliers))
  }
}

print(cooks.outliers)
```

```
##     cor_variable   distance county
## 51        avgsen   1.064429     51
## 511        polpc  21.573927     51
## 84          wser 137.563613     84
```

explain cook's distance a bit more

We can see that there are 3 influential outliers, that show up in separate correlations.

deal with each of these outliers

With these variables, to find the best optimal set, we first perform backward stepwise model selection with AIC as the criteria to find one possible solution.

EXPLAIN STEPWISE REGRESSION FURTHER.

EDA on each of these; some of the probabilities are greater than 1; explain that these actually aren't a probability. LOOK FOR OUTLIERS.

```
X$prbconv_offense <- X$prbarr * X$prbconv
X$prbpris_offense <- X$prbconv_offense * X$prbpris
X <- X[-2:-3] #remove original prbarr, prbpris variable in factor of transformed

train_X <- cbind(X, C)
nonwage_variables <- c('prbarr', 'prbconv', 'prbpris', 'avgsen',
                       'polpc',
                       'pctymle', 'pctmin80')

wage_variables <- c('wtrd', 'wfir', 'wser', 'wfed', 'wsta', 'wloc',
                    'wcon', 'wtuc', 'wmfg')

X_non_wage <- data[, names(data) %in% nonwage_variables]
X_wage <- lapply(data[, names(data) %in% wage_variables], log)

X_stepwise_set1 <- cbind(X_non_wage, X_wage)

model_upper <- lm(y ~ ., data=train_X)
model_lower <- lm(y ~ 1, data=train_X)

AIC.min_model <- stepAIC(model_upper,
                         trace = FALSE,
                         direction = 'backward',
                         scope = list(upper=model_upper,lower=model_lower))
summary(AIC.min_model)$r.squared
```

```
## [1] 0.9479289
```