Kevin's Sandbox

Kevin Hanna

November 23, 2018

```r
library(knitr)
library(kableExtra)
library(car)
```

```
## Loading required package: carData
```

```r
codebook <- read.csv('codebook.csv')
crime <- read.csv('crime_v2.csv')

# Convert columns to factors and logical.
crime$county <- as.factor(crime$county)
crime$year <- as.factor(crime$year)
crime$west <- as.logical(crime$west)
crime$central <- as.logical(crime$central)
crime$urban <- as.logical(crime$urban)

# Create a log of the dependent variable
crime$logcrmrte <- log(crime$crmrte)

#reorder to place logcrmte next to crmrte
crime <- crime[,c(1,2,3,26,4:25)]

# Delete the 6 empty observations at the end, including the row with the apostrophe.
# We can use complete.cases to do this as these 6 observations are the only incomplete observations.
crime = crime[complete.cases(crime), ]

# Fix prbconv which is a factor rather than numeric due to the apostrophe
# Convert from factor to numeric
crime$prbconv = as.numeric(as.character(crime$prbconv))

# county 193 is duplidated, remove one
crime = crime[!duplicated(crime), ]

# Create a column exluding prbconv > 1 values
#crime$prbconv_fix = crime$prbconv
#crime[crime$prbconv_fix > 1, 'prbconv_fix'] = NA
```

Preliminary Infomations (not intended to be left in)

From the assignment:

- 1. What do you want to measure? Make sure you identify variables that will be relevant to the concerns of the political campaign.
- 2. What transformations should you apply to each variable? This is very important because transformations can reveal linearities in the data, make our results relevant, or help us meet model

assumptions.

- 3. Are your choices supported by EDA? You will likely start with some general EDA to detect anomalies (missing values, top-coded variables, etc.). From then on, your EDA should be interspersed with your model building. Use visual tools to guide your decisions.
- 4. What covariates help you identify a causal effect? What covariates are problematic, either due to multicollinearity, or because they will absorb some of a causal effect you want to measure?

Variables:

1. Target

- crmrte

2. Label

- county

3. Geographic:

- density (likely related to others, especially urban)
- west
- central
- urban

Correlation between logcrmrte and urban: 0.491 and with density 0.633.
Correlation between urban and density is 0.820
Correlation between logcrmrte and west is -0.414 west is also negatively correlated with density.

I think density is an important variable (more so than urban). This would be logical as low income housing is often high-density.

```
# Geographic
#foo2 = lm(crmrte ~ urban + central + west + density, data = crime)
#foo2$coefficients
#vcov(foo2)

foo2log = lm(logcrmrte ~ urban + central + west + density, data = crime)
foo2log$coefficients
```

```
## (Intercept)   urbanTRUE centralTRUE    westTRUE     density
##  -3.6949892  -0.2841904  -0.2604751  -0.5223082   0.2818198
```

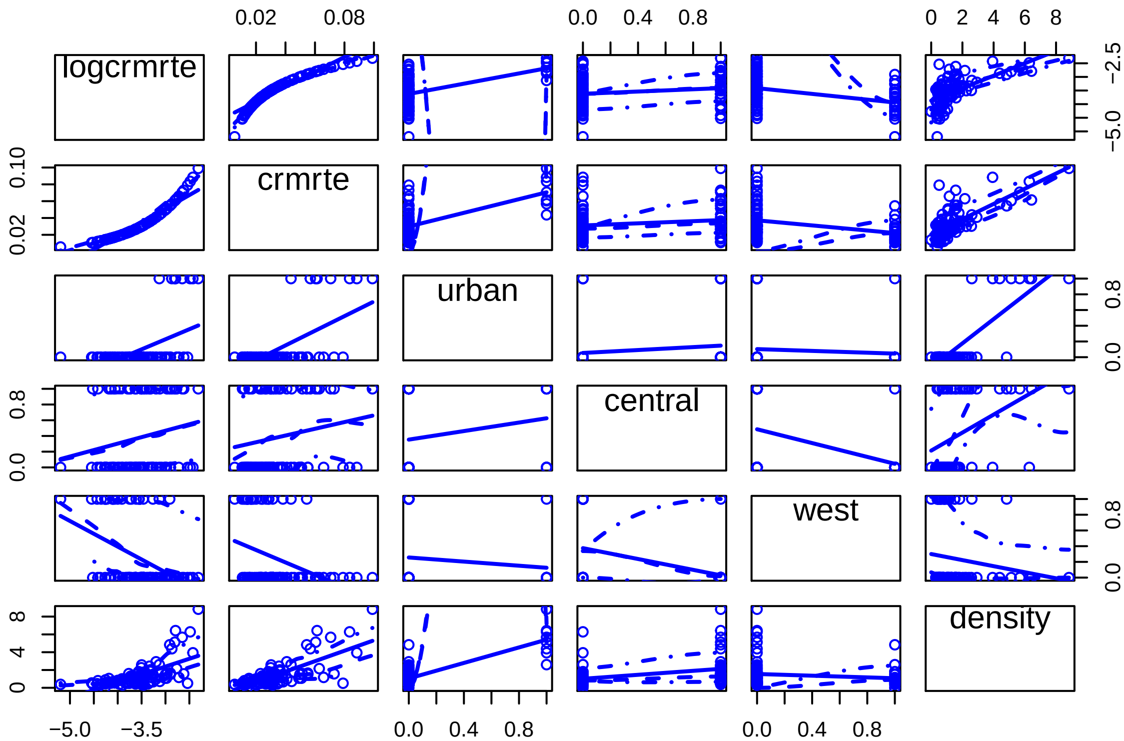
```
#vcov(foo2log)

foo2rows = c("logcrmrte", "crmrte", "urban", "central", "west", "density")

round(cor(crime[foo2rows]), 3)
```

```
##           logcrmrte crmrte  urban central   west density
## logcrmrte     1.000  0.942  0.491   0.185 -0.414   0.633
```

```
## crmrte        0.942  1.000  0.615   0.166 -0.346   0.728
## urban         0.491  0.615  1.000   0.159 -0.087   0.820
## central       0.185  0.166  0.159   1.000 -0.390   0.358
## west         -0.414 -0.346 -0.087  -0.390  1.000  -0.136
## density       0.633  0.728  0.820   0.358 -0.136   1.000
```

```
scatterplotMatrix(crime[,foo2rows], diagonal = FALSE)
```

```
## Warning in smoother(x[subs], y[subs], col = smoother.args$col[i], log.x =
## FALSE, : could not fit smooth

## Warning in smoother(x[subs], y[subs], col = smoother.args$col[i], log.x =
## FALSE, : could not fit smooth

## Warning in smoother(x[subs], y[subs], col = smoother.args$col[i], log.x =
## FALSE, : could not fit smooth

## Warning in smoother(x[subs], y[subs], col = smoother.args$col[i], log.x =
## FALSE, : could not fit smooth

## Warning in smoother(x[subs], y[subs], col = smoother.args$col[i], log.x =
## FALSE, : could not fit smooth

## Warning in smoother(x[subs], y[subs], col = smoother.args$col[i], log.x =
## FALSE, : could not fit smooth

## Warning in smoother(x[subs], y[subs], col = smoother.args$col[i], log.x =
## FALSE, : could not fit smooth

## Warning in smoother(x[subs], y[subs], col = smoother.args$col[i], log.x =
## FALSE, : could not fit smooth

## Warning in smoother(x[subs], y[subs], col = smoother.args$col[i], log.x =
## FALSE, : could not fit smooth
```



4. Cost of doing crime:

Probabilities:

- prbconv
- prbpris
- prbarr

Both prbarr and prbconv are negativly correlated to logcrmrte (-0.473 and -0.447 respectively). prbconv is less reliable (unless we can explain the > 1 values.)

```
# Probabilities

#foo1 = lm(crmrte ~ prbarr + prbconv + prbpris, data = crime)
#foo1$coefficients
#vcov(foo1)
foo1log = lm(logcrmrte ~ prbarr + prbconv +  prbpris, data = crime )
foo1log$coefficients
```

```
## (Intercept)      prbarr     prbconv     prbpris
##  -2.6846297  -1.9991732  -0.7364431   0.3380481
```

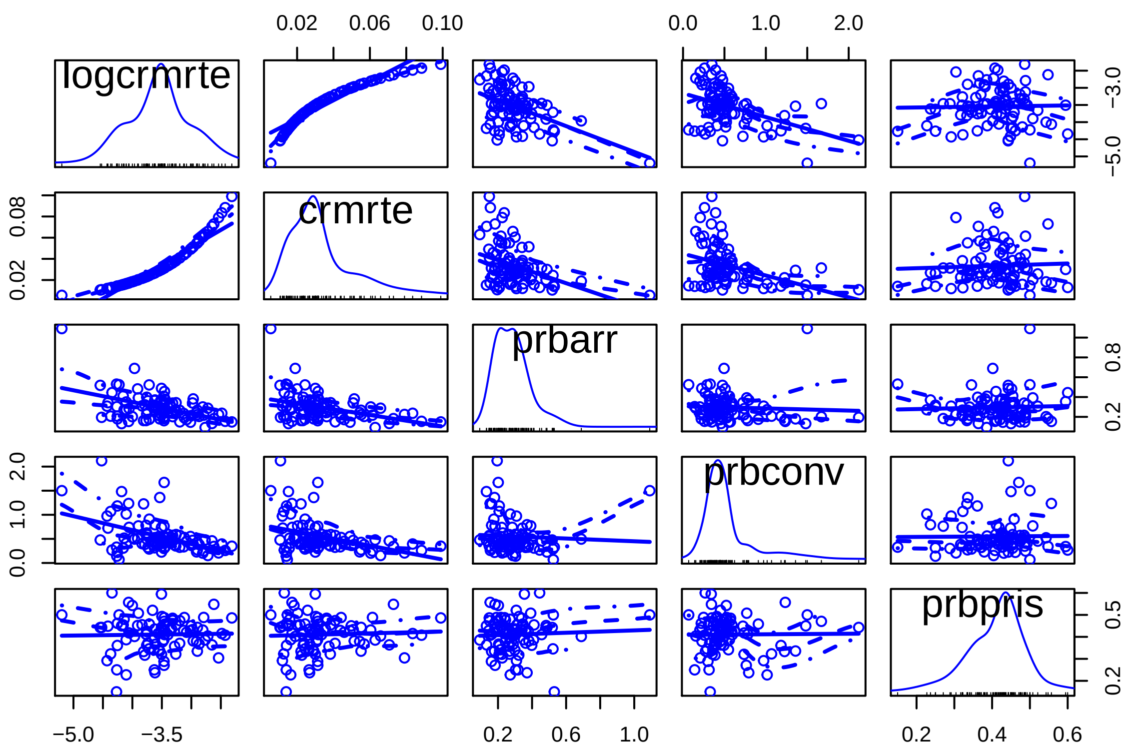
```
#vcov(foo1log)

foo1rows = c("logcrmrte", "crmrte", "prbarr", "prbconv", "prbpris")
```

```r
round(cor(crime[foo1rows]), 3)
```

```
##           logcrmrte crmrte prbarr prbconv prbpris
## logcrmrte     1.000  0.942 -0.473  -0.447   0.021
## crmrte        0.942  1.000 -0.395  -0.386   0.048
## prbarr       -0.473 -0.395  1.000  -0.056   0.046
## prbconv      -0.447 -0.386 -0.056   1.000   0.011
## prbpris       0.021  0.048  0.046   0.011   1.000
```

```r
scatterplotMatrix(crime[,foo1rows], diagonal = "histogram")
```

```
## Warning in applyDefaults(diagonal, defaults = list(method =
## "adaptiveDensity"), : unnamed diag arguments, will be ignored
```




```r
#crime_tmp = crime[complete.cases(crime), ]
#foo7log = lm(logcrmrte ~ prbarr + prbconv_fix +  prbpris, data = crime_tmp )
#foo7log$coefficients
#vcov(foo1log)

#foo7rows = c("logcrmrte", "crmrte", "prbarr", "prbconv_fix", "prbpris")

#round(cor(crime_tmp[foo7rows]), 3)
#scatterplotMatrix(crime_tmp[,foo7rows], diagonal = "histogram")

#remove(crime_tmp)
```

Sentence and police

- avgsen

- polpc (likely related to prbconv)

polpc has a huge correlation, it makes sense, but it's still so high we should be very cautious.

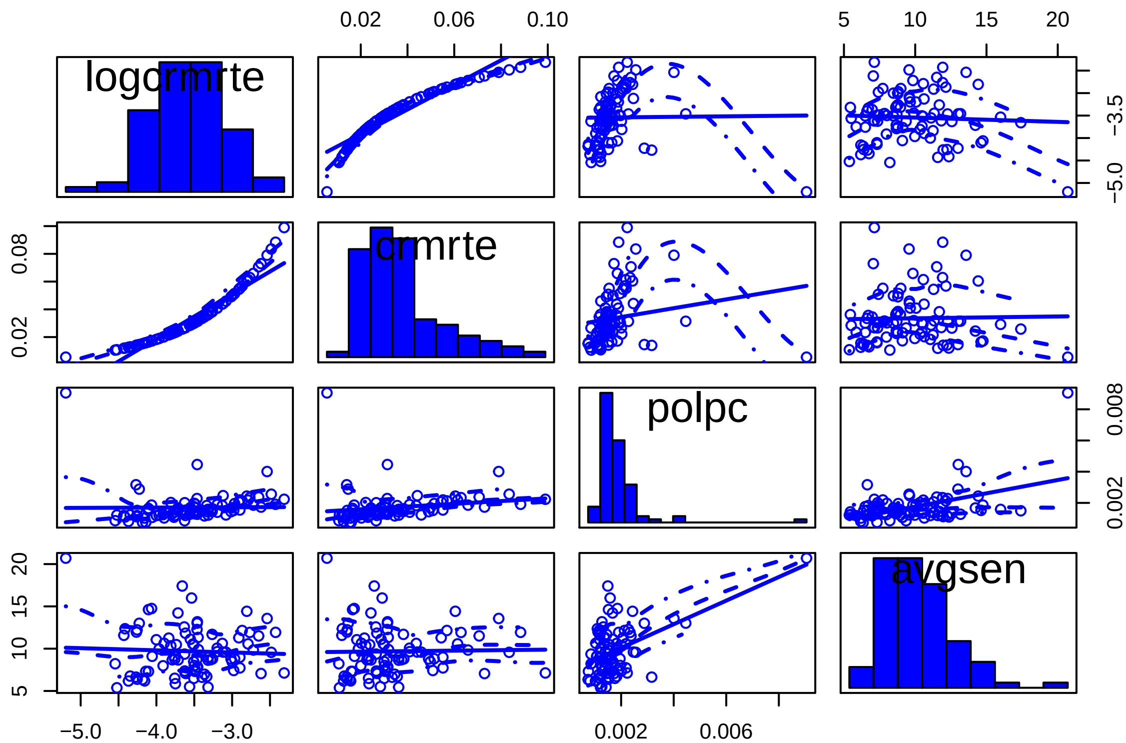
```
# Sentence and police

#foo3 = lm(crmrte ~ polpc + avgsen, data = crime)
#foo3$coefficients
#vcov(foo3)

foo3log = lm(logcrmrte ~ polpc + avgsen, data = crime)
foo3log$coefficients
```

```
## (Intercept)       polpc      avgsen
## -3.45048112 25.08600936 -0.01383982
```

```
#vcov(foo3log)

foo3rows = c("logcrmrte", "crmrte", "polpc", "avgsen")

round(cor(crime[foo3rows]), 3)
```

```
##           logcrmrte crmrte polpc avgsen
## logcrmrte     1.000  0.942 0.010 -0.049
## crmrte        0.942  1.000 0.167  0.020
## polpc         0.010  0.167 1.000  0.488
## avgsen       -0.049  0.020 0.488  1.000
```

```
scatterplotMatrix(crime[,foo3rows], diagonal=list(method ="histogram", breaks="FD"))
```



5. Economics

- taxpc
- wcon
- wtuc
- wtrd
- wfir
- wser
- wmfg
- wfed
- wsta
- wloc

There's a lot to take in, however the negative relatinship to wser (wage service worker) is initially the most interesting.

```
# Economics
#foo4 = lm(crmrte ~ taxpc + wcon + wtuc + wtrd + wfir + wser + wmfg + wfed + wsta + wloc, data = crime)
#foo4$coefficients
#vcov(foo4)

foo4log = lm(logcrmrte ~ taxpc + wcon + wtuc + wtrd + wfir + wser + wmfg + wfed + wsta + wloc, data = c
foo4log$coefficients
```

```
##    (Intercept)          taxpc           wcon           wtuc           wtrd
## -6.2657436983  0.0139749059  0.0015560093 -0.0003859724  0.0017671261
```

```
##          wfir          wser          wmfg          wfed          wsta
## -0.0018814706 -0.0003771697  0.0001090428  0.0046852095  0.0017895087
##          wloc
## -0.0016300505
```

```r
#vcov(foo4log)

foo4rows = c("logcrmrte", "crmrte", "taxpc", "wcon", "wtuc", "wtrd", "wfir", "wser", "wmfg", "wfed", "w

round(cor(crime[foo4rows]), 2)
```

```
##           logcrmrte crmrte taxpc  wcon  wtuc  wtrd wfir  wser wmfg wfed
## logcrmrte      1.00   0.94  0.36  0.39  0.20  0.39 0.29 -0.11 0.31 0.52
## crmrte         0.94   1.00  0.45  0.39  0.24  0.43 0.34 -0.05 0.35 0.49
## taxpc          0.36   0.45  1.00  0.26  0.17  0.18 0.13  0.08 0.26 0.06
## wcon           0.39   0.39  0.26  1.00  0.41  0.56 0.49 -0.01 0.35 0.51
## wtuc           0.20   0.24  0.17  0.41  1.00  0.35 0.33 -0.02 0.47 0.40
## wtrd           0.39   0.43  0.18  0.56  0.35  1.00 0.67 -0.02 0.37 0.64
## wfir           0.29   0.34  0.13  0.49  0.33  0.67 1.00  0.01 0.50 0.62
## wser          -0.11  -0.05  0.08 -0.01 -0.02 -0.02 0.01  1.00 0.01 0.02
## wmfg           0.31   0.35  0.26  0.35  0.47  0.37 0.50  0.01 1.00 0.52
## wfed           0.52   0.49  0.06  0.51  0.40  0.64 0.62  0.02 0.52 1.00
## wsta           0.17   0.20 -0.03 -0.02 -0.15  0.01 0.24  0.04 0.05 0.19
## wloc           0.29   0.36  0.22  0.52  0.33  0.58 0.55  0.08 0.45 0.52
##            wsta wloc
## logcrmrte  0.17 0.29
## crmrte     0.20 0.36
## taxpc     -0.03 0.22
## wcon      -0.02 0.52
## wtuc      -0.15 0.33
## wtrd       0.01 0.58
## wfir       0.24 0.55
## wser       0.04 0.08
## wmfg       0.05 0.45
## wfed       0.19 0.52
## wsta       1.00 0.16
## wloc       0.16 1.00
```

```r
#scatterplotMatrix(crime[,foo4rows], diagonal = "histogram")
```

6. Demographics

- pctmin80
- pctymle

pctymle is strongly correlated.

```r
# Demographics
#foo5 = lm(crmrte ~ pctmin80 + pctymle, data = crime)
#foo5$coefficients
#vcov(foo5)

foo5log = lm(logcrmrte ~pctmin80 + pctymle, data = crime)
foo5log$coefficients
```

```
##  (Intercept)     pctmin80      pctymle
```

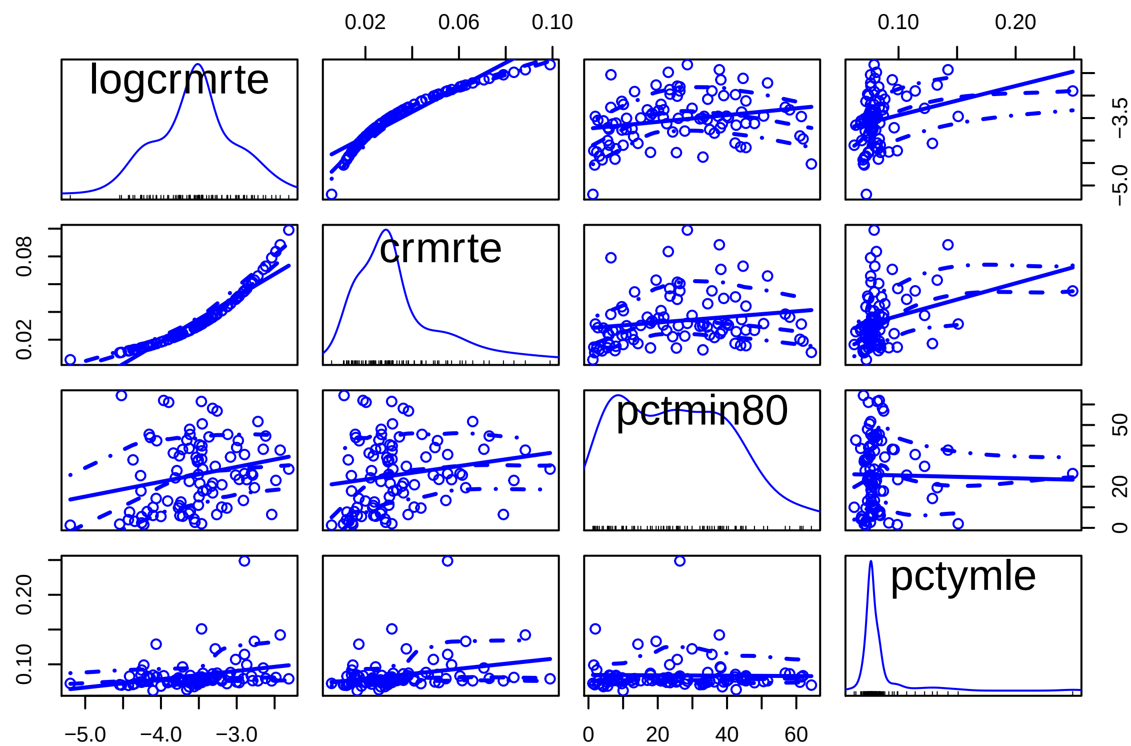
```
## -4.295710411  0.007701047  6.616597353
```

```r
#vcov(foo5log)

foo5rows = c("logcrmrte", "crmrte", "pctmin80", "pctymle")

round(cor(crime[foo5rows]), 3)
```

```
##           logcrmrte crmrte pctmin80 pctymle
## logcrmrte     1.000  0.942    0.233   0.278
## crmrte        0.942  1.000    0.182   0.290
## pctmin80      0.233  0.182    1.000  -0.019
## pctymle       0.278  0.290   -0.019   1.000
```

```r
scatterplotMatrix(crime[,foo5rows], diagonal = "histogram")
```

```
## Warning in applyDefaults(diagonal, defaults = list(method =
## "adaptiveDensity"), : unnamed diag arguments, will be ignored
```



7. Crime types

The higher the ratio of face-to-face crimes ends up with fewer crimes. I suspect this is the result of a small police force that doesn't have as much time to go after less significant crimes, so I added that variable in too. They're not strongly correlated.

```r
# Crime Types
foo6log = lm(logcrmrte ~ mix + polpc, data = crime)
```

```r
foo6log$coefficients
```

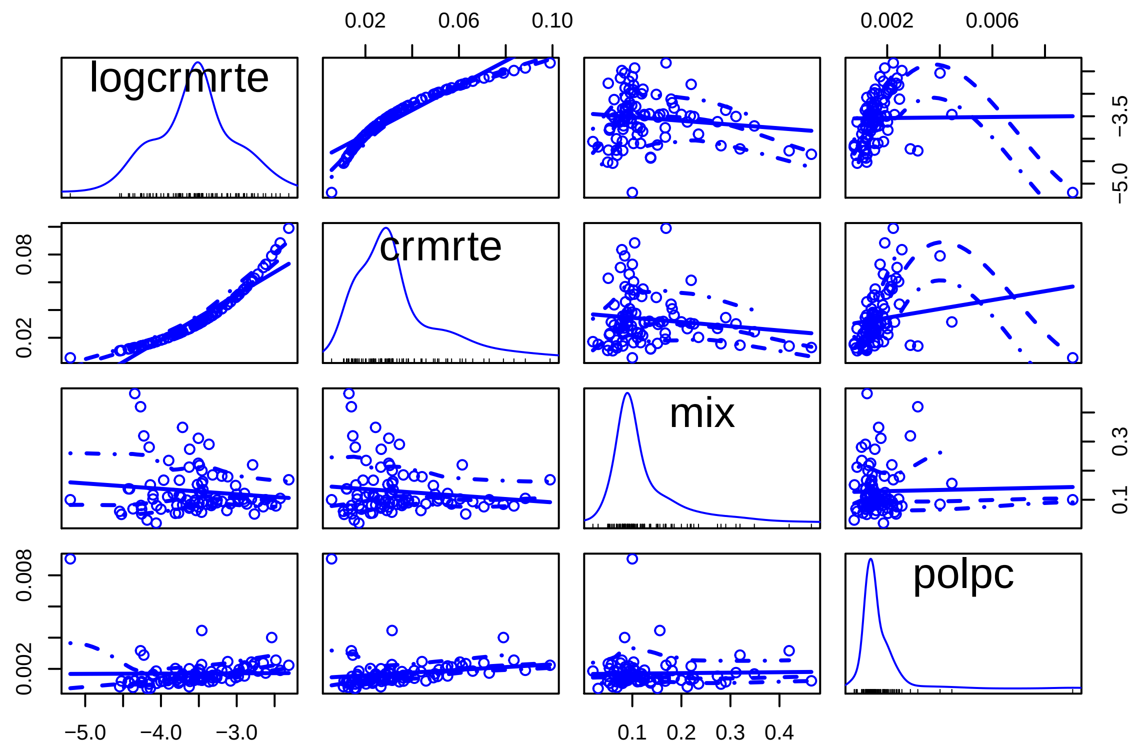
```
## (Intercept)         mix       polpc
##  -3.4461127  -0.8393071   7.4322745
```

```r
#vcov(foo5log)

foo6rows = c("logcrmrte", "crmrte", "mix", "polpc")

round(cor(crime[foo6rows]), 3)
```

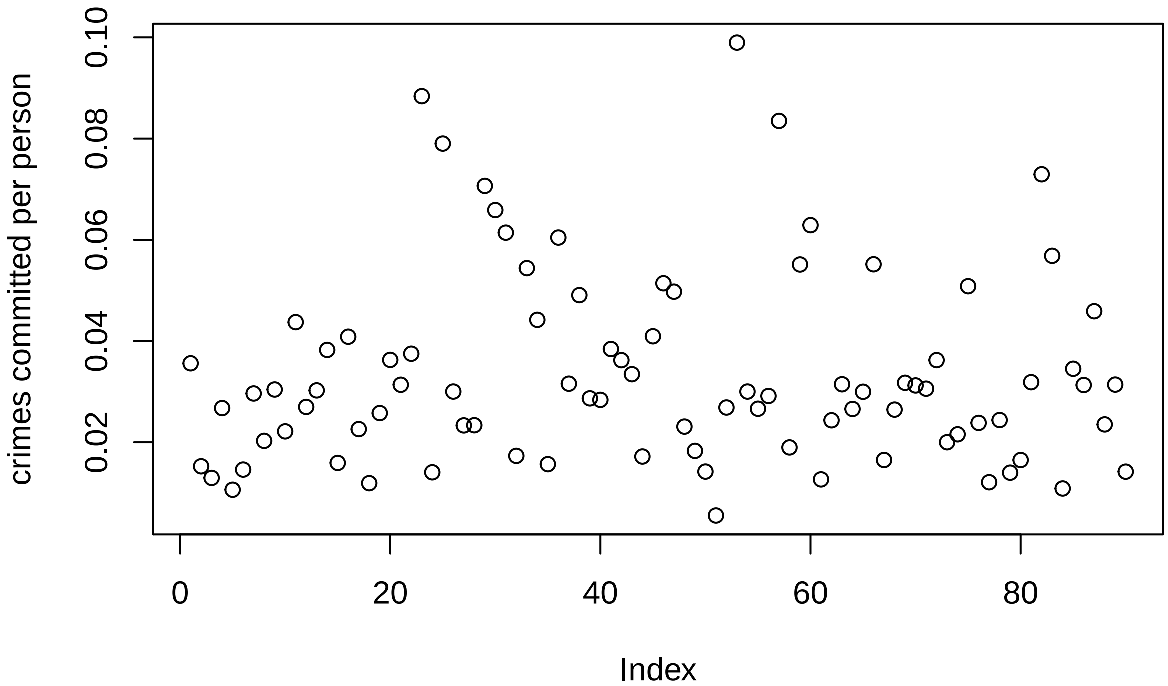
```
##           logcrmrte crmrte    mix polpc
## logcrmrte     1.000  0.942 -0.125 0.010
## crmrte        0.942  1.000 -0.132 0.167
## mix          -0.125 -0.132  1.000 0.024
## polpc         0.010  0.167  0.024 1.000
```

```r
scatterplotMatrix(crime[,foo6rows], diagonal = "histogram")
```

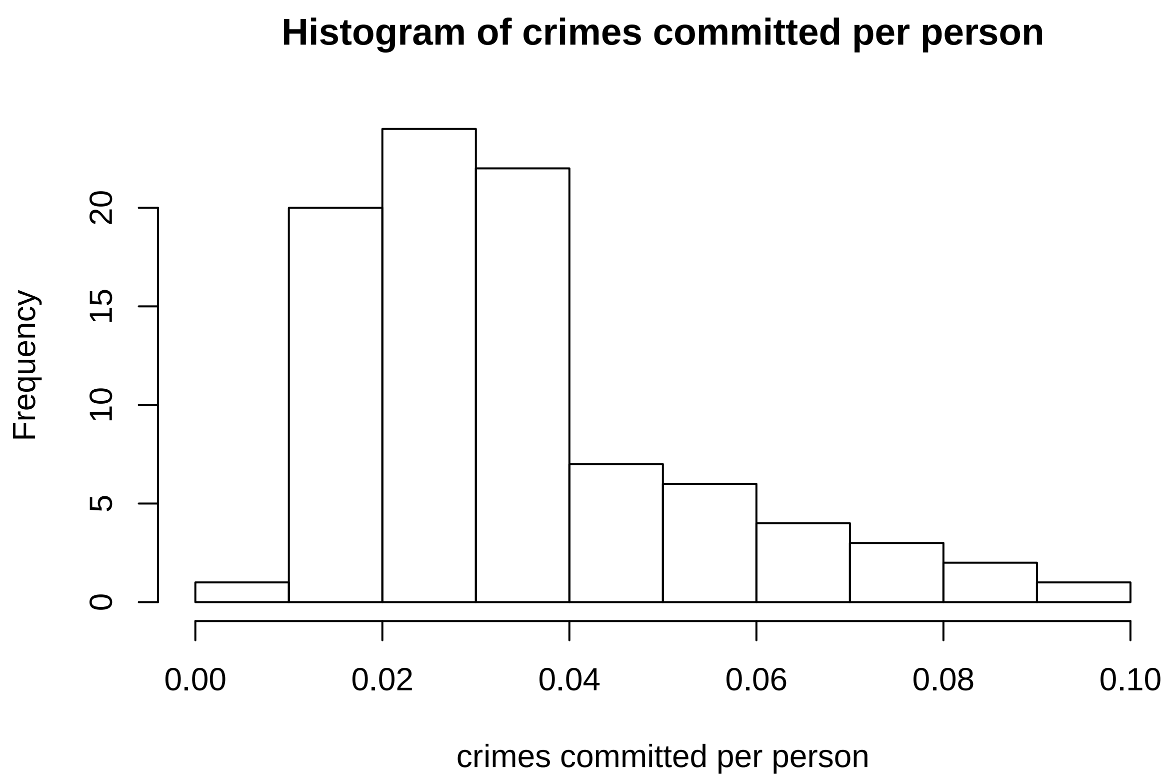
```
## Warning in applyDefaults(diagonal, defaults = list(method =
## "adaptiveDensity"), : unnamed diag arguments, will be ignored
```




```r
plot(crime$crmrte, ylab = 'crimes committed per person')
```




```
hist(crime$crmrte, xlab = 'crimes committed per person', main = 'Histogram of crimes committed per perso
```



parsimoneous model

```
model1 <- lm(logcrmrte ~ density + prbarr + polpc + wser + mix + pctmin80 + pctymle, data = crime)
(model1$coefficients)
```

```
##   (Intercept)       density        prbarr         polpc          wser
## -3.8526048249  0.1822832403 -1.6660083568 88.2812037246 -0.0006698481
##           mix      pctmin80       pctymle
##  0.0494596366  0.0119206523  3.1157342337
```

```
summary(model1)
```

```
##
## Call:
## lm(formula = logcrmrte ~ density + prbarr + polpc + wser + mix +
##     pctmin80 + pctymle, data = crime)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.75119 -0.19258  0.01882  0.19391  1.07098
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.8526048  0.1895520 -20.325  < 2e-16 ***
## density      0.1822832  0.0255094   7.146 3.34e-10 ***
## prbarr      -1.6660084  0.3475995  -4.793 7.23e-06 ***
## polpc       88.2812037 43.0685205   2.050  0.04358 *
## wser        -0.0006698  0.0001769  -3.787  0.00029 ***
```

```
## mix          0.0494596  0.4901234   0.101  0.91987
## pctmin80     0.0119207  0.0021983   5.423 5.79e-07 ***
## pctymle      3.1157342  1.5311643   2.035  0.04509 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3285 on 82 degrees of freedom
## Multiple R-squared:  0.6699, Adjusted R-squared:  0.6417
## F-statistic: 23.77 on 7 and 82 DF,  p-value: < 2.2e-16
```

```r
AIC(model1)
```

```
## [1] 64.62546
```

```r
model2 <- lm(logcrmrte ~ density + prbarr + polpc + pctymle, data = crime)
(model2$coefficients)
```

```
## (Intercept)     density      prbarr       polpc     pctymle
##  -3.8052783   0.1845146  -1.2429850  29.7931639   3.7457232
```

```r
summary(model2)
```

```
##
## Call:
## lm(formula = logcrmrte ~ density + prbarr + polpc + pctymle,
##     data = crime)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.84438 -0.25617  0.02812  0.24318  1.04735
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.80528    0.20242 -18.799  < 2e-16 ***
## density      0.18451    0.03029   6.092 3.14e-08 ***
## prbarr      -1.24299    0.37256  -3.336  0.00126 **
## polpc       29.79316   49.39663   0.603  0.54802
## pctymle      3.74572    1.81373   2.065  0.04195 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3905 on 85 degrees of freedom
## Multiple R-squared:  0.5164, Adjusted R-squared:  0.4936
## F-statistic: 22.69 on 4 and 85 DF,  p-value: 8.955e-13
```

```r
AIC(model2)
```

```
## [1] 92.99971
```

Steps for evaluating variables

Leverage (and Influence if required)
Goodness-of-Fit : AIC
Omitted variable bias
MSE
E[theta hat] = theta

```r
crime$urban + crime$west + crime$central
```

```
##  [1] 1 1 1 1 1 1 0 0 0 0 2 1 1 1 1 1 1 1 0 1 0 0 1 0 0 1 1 0 2 0 2 1 2 1 0
## [36] 2 0 0 1 1 0 0 1 1 0 1 0 1 1 1 1 0 2 1 1 0 1 0 0 1 0 0 0 0 1 0 1 1 1 0
## [71] 1 1 1 0 0 1 1 1 1 1 1 1 2 1 0 1 0 1 0 1
```

```r
crime$urban + crime$west
```

```
##  [1] 0 0 1 0 1 1 0 0 0 0 2 1 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0
## [36] 1 0 0 1 1 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0
## [71] 0 0 1 0 0 0 0 1 1 1 0 0 1 0 0 1 0 1 0 1
```

```r
crime$urban + crime$central
```

```
##  [1] 1 1 0 1 0 0 0 0 0 0 1 0 1 0 1 1 1 0 0 1 0 0 1 0 0 1 1 0 2 0 2 1 1 1 0
## [36] 2 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 2 1 1 0 1 0 0 1 0 0 0 0 1 0 0 1 1 0
## [71] 1 1 0 0 0 1 1 0 0 0 1 1 2 1 0 0 0 0 0 0
```

```r
crime$west + crime$central
```

```
##  [1] 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 0 1 0 1 1 2 1 0
## [36] 1 0 0 1 1 0 0 1 1 0 1 0 1 1 1 1 0 1 1 1 0 0 0 0 1 0 0 0 0 1 0 1 1 1 0
## [71] 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0 1 0 1 0 1
```

```r
crime$avgwage = (crime$wcon + crime$wtuc + crime$wtrd + crime$wfir + crime$wser + crime$wmfg + crime$wf

cn = colnames(crime)
cnlen = length(cn)
results = data.frame()

for (i in 5:cnlen) {

  if (!0 %in% crime[,cn[i]] & !FALSE %in% crime[,cn[i]]) {

  var = crime[,cn[i]]
  varlog = log(crime[,cn[i]])

  print(cn[i])
  mod1 = lm(crime$crmrte ~ var)
  mod2 = lm(crime$logcrmrte ~ var)
  mod3 = lm(crime$logcrmrte ~ varlog)
  mod4 = lm(crime$crmrte ~ varlog)
#  mod2  = lm(as.formula(paste("logcrmrte ~", cn[i])), data=crime)
#  mod3  = lm(as.formula(paste("logcrmrte ~", log(cn[i]))), data=crime)
#  mod4  = lm(as.formula(paste("crmrte ~", log(cn[i]))), data=crime)

  results = rbind(results, data.frame(var=cn[i],
                                      rsquared_level_level=summary(mod1)[8],
                                      rsquared_log_level=summary(mod2)[8],
                                      rsquared_log_log=summary(mod3)[8],
                                      rsquared_level_log=summary(mod4)[8],
                                      tvalue_level_level=summary(mod1)$coefficients[2,4],
                                      tvalue_log_level=summary(mod2)$coefficients[2,4],
                                      tvalue__log_log=summary(mod3)$coefficients[2,4],
                                      tvalue_level_log=summary(mod4)$coefficients[2,4]
                                      ))
  }
```

```
}
```

```
## [1] "prbarr"
## [1] "prbconv"
## [1] "prbpris"
## [1] "avgsen"
## [1] "polpc"
## [1] "density"
## [1] "taxpc"
## [1] "pctmin80"
## [1] "wcon"
## [1] "wtuc"
## [1] "wtrd"
## [1] "wfir"
## [1] "wser"
## [1] "wmfg"
## [1] "wfed"
## [1] "wsta"
## [1] "wloc"
## [1] "mix"
## [1] "pctymle"
## [1] "avgwage"
```