

Project 1

MSIT 423, Spring 2019

Due: April 27, 2:00pm

Jessica Qin, Yang Hong, Yunzi Zhang, Yifan Chen, Ray Liu

```
library(car)
```

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

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

0.1 Loading the data from the csv file

```
setwd("~/Desktop/2019/NU/2019-spring/MSIT423/project1")
```

```
bike<-read.csv("bike.csv")
```

```
bike$avgbf= (bike$Limited_Business_License+bike$Retail_Food_Establishment)/2
```

```
dmg <- bike[,c(3:5,8,10:13,45,47)]
```

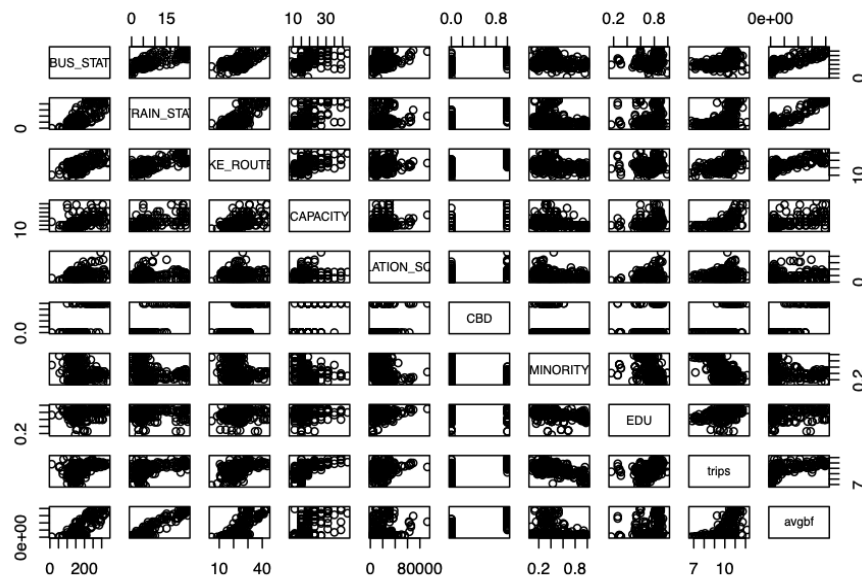
```
cor(dmg)
```

```
##              CTA_BUS_STATIONS CTA_TRAIN_STATIONS BIKE_ROUTES
## CTA_BUS_STATIONS      1.0000000      0.76391087  0.57678715
## CTA_TRAIN_STATIONS    0.7639109      1.00000000  0.68212020
## BIKE_ROUTES           0.5767872      0.68212020  1.00000000
## CAPACITY             0.4670364      0.60617220  0.46837336
## POPULATION_SQ_MILE    0.3038109      0.08970660  0.09750206
## CBD                  0.7233554      0.85270533  0.63539809
## MINORITY             -0.1816221     -0.23396665 -0.25718405
## EDU                   0.1119116      0.09380398  0.12139609
## trips                0.4300325      0.52622037  0.51112087
## avgbf                0.7953145      0.94037689  0.72820843
##              CAPACITY POPULATION_SQ_MILE      CBD  MINORITY
## CTA_BUS_STATIONS    0.4670364      0.30381094  0.7233554 -0.1816221
## CTA_TRAIN_STATIONS  0.6061722      0.08970660  0.8527053 -0.2339666
## BIKE_ROUTES         0.4683734      0.09750206  0.6353981 -0.2571841
## CAPACITY            1.0000000      0.11211830  0.6093845 -0.2152348
## POPULATION_SQ_MILE  0.1121183      1.00000000  0.1817645 -0.2667885
## CBD                 0.6093845      0.18176454  1.0000000 -0.1686462
## MINORITY            -0.2152348     -0.26678850 -0.1686462  1.0000000
## EDU                 0.1899403      0.26077076  0.1216539 -0.3768247
## trips              0.5944283      0.21655885  0.5269037 -0.6369958
## avgbf              0.6075820      0.08185719  0.8826539 -0.2018209
##              EDU      trips      avgbf
```

```
## CTA_BUS_STATIONS    0.11191158  0.4300325  0.79531447
## CTA_TRAIN_STATIONS  0.09380398  0.5262204  0.94037689
## BIKE_ROUTES         0.12139609  0.5111209  0.72820843
## CAPACITY            0.18994029  0.5944283  0.60758200
## POPULATION_SQ_MILE  0.26077076  0.2165588  0.08185719
## CBD                 0.12165392  0.5269037  0.88265393
## MINORITY            -0.37682469 -0.6369958 -0.20182091
## EDU                 1.00000000  0.3584473  0.14569653
## trips               0.35844727  1.0000000  0.56766421
## avgbf               0.14569653  0.5676642  1.00000000
```

```
plot(dmg)
library(MASS)
#tran1=cbind(log(com1.2[,c(1:6,9)]),com1.2[,7:8],com1.2[,10:11])
library(glmnet)
```

```
## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-16
```



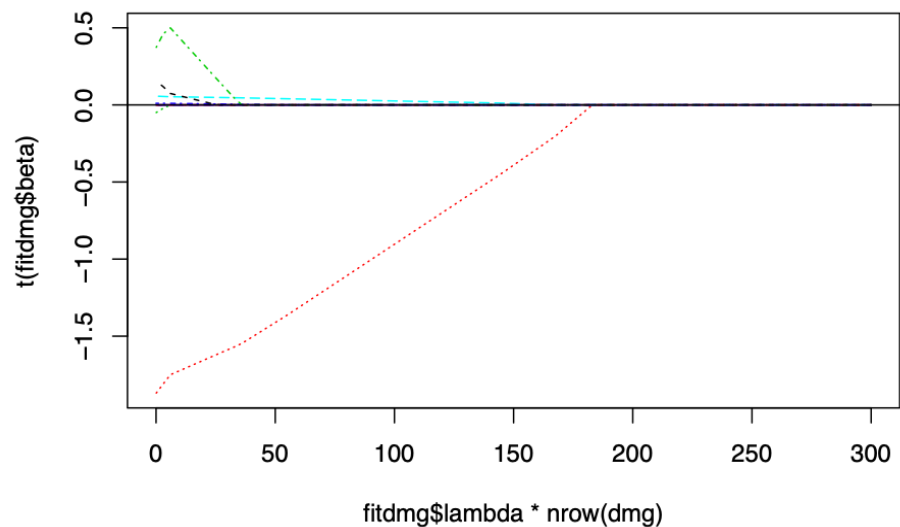
```
lam = seq(0,300,length=101)/nrow(dmg)
x = model.matrix(trips~., dmg)
fitdmg=glmnet(x,dmg$trips, alpha=1, lambda = lam)
cv.lasso=cv.glmnet(x, dmg$trips, alpha=1, lambda = lam)
cv.lasso$lambda.min
```

```
## [1] 0
```

```
coef(cv.lasso,s="lambda.min")
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##              1
## (Intercept)  9.251243e+00
## (Intercept)  .
## CTA_BUS_STATIONS -1.391762e-03
## CTA_TRAIN_STATIONS -4.999291e-02
## BIKE_ROUTES  9.881654e-03
## CAPACITY  5.672128e-02
## POPULATION_SQ_MILE 1.988521e-06
## CBD  1.555312e-01
## MINORITY -1.870577e+00
## EDU  3.708170e-01
## avgbf  4.329040e-06
```

```
matplot(fitdmg$lambda*nrow(dmg), t(fitdmg$beta), type="l"); abline(h=0)
```



```
#fit the model
```

```
fit2.1= lm(trips~ CTA_BUS_STATIONS+ CTA_TRAIN_STATIONS + BIKE_ROUTES + CAPACITY +I(MINORITY~
summary(fit2.1)
```

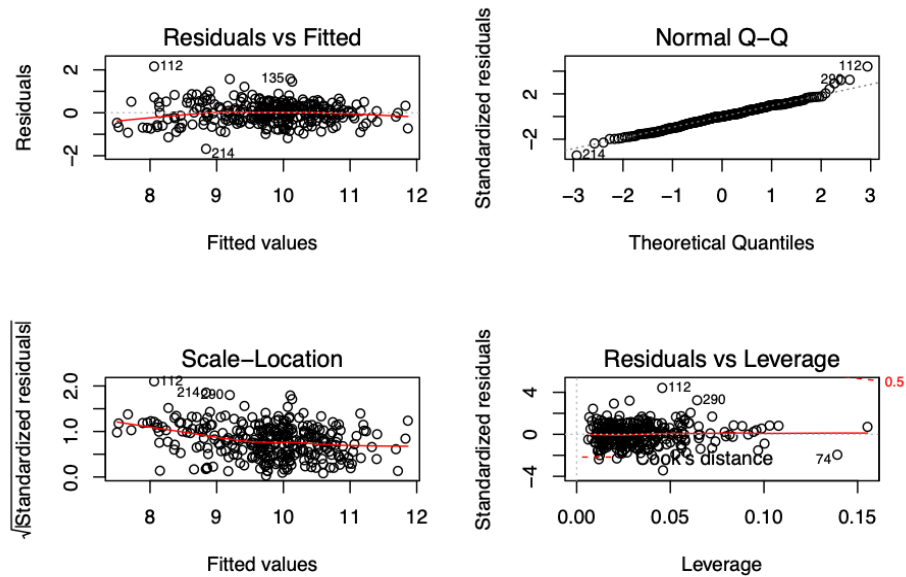
```
##
## Call:
## lm(formula = trips ~ CTA_BUS_STATIONS + CTA_TRAIN_STATIONS +
##     BIKE_ROUTES + CAPACITY + I(MINORITY^2) + EDU + CBD + POPULATION_SQ_MILE +
##     log(avgbf), data = dmg)
##
## Residuals:
```

```
##      Min      1Q   Median      3Q      Max
## -1.68025 -0.30942  0.00193  0.31565  2.16113
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.271e+00  5.185e-01  10.165 < 2e-16 ***
## CTA_BUS_STATIONS -1.380e-03  9.308e-04  -1.482  0.13932
## CTA_TRAIN_STATIONS -2.980e-02  1.092e-02  -2.730  0.00672 **
## BIKE_ROUTES      3.492e-03  6.559e-03   0.532  0.59483
## CAPACITY         5.235e-02  7.089e-03   7.385 1.63e-12 ***
## I(MINORITY^2)    -1.512e+00  1.227e-01 -12.322 < 2e-16 ***
## EDU              4.379e-01  2.533e-01   1.729  0.08494 .
## CBD              2.210e-01  1.495e-01   1.479  0.14034
## POPULATION_SQ_MILE 2.638e-06  2.609e-06   1.011  0.31280
## log(avgbf)       3.684e-01  5.388e-02   6.837 4.78e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5009 on 290 degrees of freedom
## Multiple R-squared:  0.7357, Adjusted R-squared:  0.7275
## F-statistic: 89.71 on 9 and 290 DF, p-value: < 2.2e-16
```

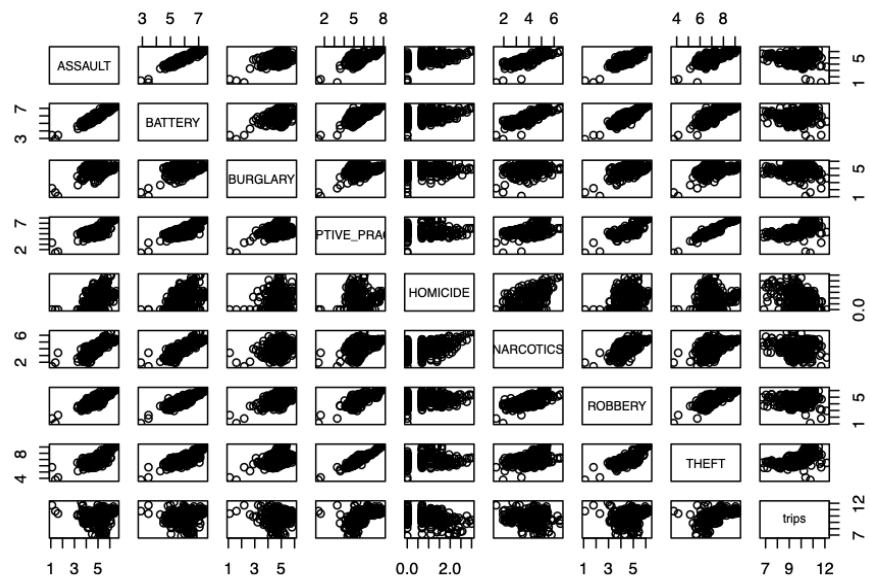
```
vif(fit2.1)
```

```
##      CTA_BUS_STATIONS CTA_TRAIN_STATIONS      BIKE_ROUTES
##           3.035598           5.908774           2.265607
##           CAPACITY      I(MINORITY^2)           EDU
##           1.732919           1.407974           1.232317
##           CBD POPULATION_SQ_MILE      log(avgbf)
##           4.380889           1.334459           4.860235
```

```
#diagnostic
library(car)
par(mfrow=c(2,2))
plot(fit2.1)
```



```
#crime
crime <- bike[,c(15,16,17,22,24,31,40,43,45)]
plot(crime)
```



```
dmgcrime<- cbind(dmg[,c(1:10)], bike[,c(15,16,17,22,24,31,40,43)])
cor(dmgcrime)
```

##	CTA_BUS_STATIONS	CTA_TRAIN_STATIONS	BIKE_ROUTES	
## CTA_BUS_STATIONS	1.0000000	0.763910868	0.57678715	
## CTA_TRAIN_STATIONS	0.7639109	1.000000000	0.68212020	
## BIKE_ROUTES	0.5767872	0.682120205	1.00000000	
## CAPACITY	0.4670364	0.606172200	0.46837336	
## POPULATION_SQ_MILE	0.3038109	0.089706600	0.09750206	
## CBD	0.7233554	0.852705331	0.63539809	
## MINORITY	-0.1816221	-0.233966650	-0.25718405	
## EDU	0.1119116	0.093803985	0.12139609	
## trips	0.4300325	0.526220374	0.51112087	
## avgbf	0.7953145	0.940376886	0.72820843	
## ASSAULT	0.7422541	0.521785160	0.39011575	
## BATTERY	0.7480613	0.477795269	0.40225401	
## BURGLARY	0.2820593	0.005800537	0.04504342	
## DECEPTIVE_PRACTICE	0.8421596	0.747351310	0.68669188	
## HOMICIDE	0.1548896	-0.076293964	-0.04640741	
## NARCOTICS	0.5716115	0.368524372	0.21314223	
## ROBBERY	0.7034665	0.493954187	0.43069032	
## THEFT	0.8285750	0.712237374	0.67737098	
##	CAPACITY	POPULATION_SQ_MILE	CBD	MINORITY
## CTA_BUS_STATIONS	0.46703643	0.30381094	0.72335537	-0.181622091
## CTA_TRAIN_STATIONS	0.60617220	0.08970660	0.85270533	-0.233966650
## BIKE_ROUTES	0.46837336	0.09750206	0.63539809	-0.257184052
## CAPACITY	1.00000000	0.11211830	0.60938449	-0.215234763
## POPULATION_SQ_MILE	0.11211830	1.00000000	0.18176454	-0.266788498
## CBD	0.60938449	0.18176454	1.00000000	-0.168646219
## MINORITY	-0.21523476	-0.26678850	-0.16864622	1.000000000
## EDU	0.18994029	0.26077076	0.12165392	-0.376824694
## trips	0.59442833	0.21655885	0.52690369	-0.636995823
## avgbf	0.60758200	0.08185719	0.88265393	-0.201820906
## ASSAULT	0.27377761	0.21899064	0.50168689	0.211183759
## BATTERY	0.30094054	0.29395427	0.48945661	0.119227137
## BURGLARY	-0.06941903	0.37339738	-0.07374079	-0.271585779
## DECEPTIVE_PRACTICE	0.51195387	0.30829189	0.72777513	-0.317739027
## HOMICIDE	-0.09992196	-0.04727424	-0.05253500	0.540184937
## NARCOTICS	0.17894224	0.28818816	0.35433147	0.254880261
## ROBBERY	0.27292261	0.27898949	0.45414679	-0.001395701
## THEFT	0.48344579	0.30971845	0.69537643	-0.323389957
##	EDU	trips	avgbf	ASSAULT
## CTA_BUS_STATIONS	0.11191158	0.43003247	0.79531447	0.74225410
## CTA_TRAIN_STATIONS	0.09380398	0.52622037	0.94037689	0.52178516
## BIKE_ROUTES	0.12139609	0.51112087	0.72820843	0.39011575
## CAPACITY	0.18994029	0.59442833	0.60758200	0.27377761
## POPULATION_SQ_MILE	0.26077076	0.21655885	0.08185719	0.21899064
## CBD	0.12165392	0.52690369	0.88265393	0.50168689
## MINORITY	-0.37682469	-0.63699582	-0.20182091	0.21118376
## EDU	1.00000000	0.35844727	0.14569653	-0.03318352
## trips	0.35844727	1.00000000	0.56766421	0.03970275

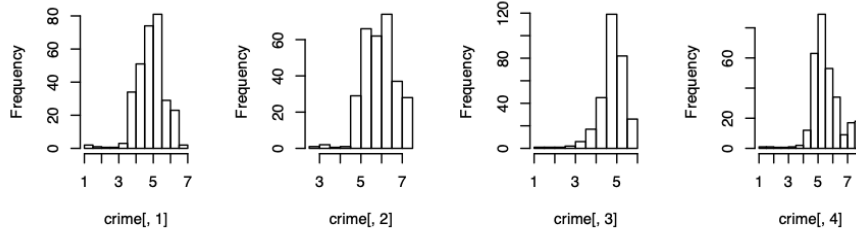
## avgbf	0.14569653	0.56766421	1.00000000	0.52695955
## ASSAULT	-0.03318352	0.03970275	0.52695955	1.00000000
## BATTERY	0.05513185	0.13209566	0.48812407	0.95044632
## BURGLARY	0.13222405	-0.01963150	-0.02669805	0.40054925
## DECEPTIVE_PRACTICE	0.19035762	0.52870467	0.76061970	0.74865873
## HOMICIDE	-0.16807883	-0.38873064	-0.01966505	0.49335299
## NARCOTICS	-0.03492525	-0.05937600	0.31785731	0.82507026
## ROBBERY	0.06050953	0.17675697	0.48578539	0.85239606
## THEFT	0.21617174	0.55132574	0.74548015	0.72866970
##	BATTERY	BURGLARY	DECEPTIVE_PRACTICE	HOMICIDE
## CTA_BUS_STATIONS	0.74806126	0.282059270	0.84215964	0.15488955
## CTA_TRAIN_STATIONS	0.47779527	0.005800537	0.74735131	-0.07629396
## BIKE_ROUTES	0.40225401	0.045043419	0.68669188	-0.04640741
## CAPACITY	0.30094054	-0.069419026	0.51195387	-0.09992196
## POPULATION_SQ_MILE	0.29395427	0.373397382	0.30829189	-0.04727424
## CBD	0.48945661	-0.073740794	0.72777513	-0.05253500
## MINORITY	0.11922714	-0.271585779	-0.31773903	0.54018494
## EDU	0.05513185	0.132224050	0.19035762	-0.16807883
## trips	0.13209566	-0.019631495	0.52870467	-0.38873064
## avgbf	0.48812407	-0.026698048	0.76061970	-0.01966505
## ASSAULT	0.95044632	0.400549254	0.74865873	0.49335299
## BATTERY	1.00000000	0.440713898	0.77256304	0.45614052
## BURGLARY	0.44071390	1.000000000	0.36014661	0.11070927
## DECEPTIVE_PRACTICE	0.77256304	0.360146614	1.00000000	0.03784259
## HOMICIDE	0.45614052	0.110709265	0.03784259	1.00000000
## NARCOTICS	0.83870461	0.241867382	0.50869141	0.52605669
## ROBBERY	0.88774347	0.551816547	0.77885944	0.28496565
## THEFT	0.77242350	0.422622699	0.94960488	0.04357397
##	NARCOTICS	ROBBERY	THEFT	
## CTA_BUS_STATIONS	0.57161154	0.703466477	0.82857496	
## CTA_TRAIN_STATIONS	0.36852437	0.493954187	0.71223737	
## BIKE_ROUTES	0.21314223	0.430690316	0.67737098	
## CAPACITY	0.17894224	0.272922615	0.48344579	
## POPULATION_SQ_MILE	0.28818816	0.278989495	0.30971845	
## CBD	0.35433147	0.454146791	0.69537643	
## MINORITY	0.25488026	-0.001395701	-0.32338996	
## EDU	-0.03492525	0.060509535	0.21617174	
## trips	-0.05937600	0.176756974	0.55132574	
## avgbf	0.31785731	0.485785392	0.74548015	
## ASSAULT	0.82507026	0.852396057	0.72866970	
## BATTERY	0.83870461	0.887743468	0.77242350	
## BURGLARY	0.24186738	0.551816547	0.42262270	
## DECEPTIVE_PRACTICE	0.50869141	0.778859440	0.94960488	
## HOMICIDE	0.52605669	0.284965650	0.04357397	
## NARCOTICS	1.00000000	0.696017454	0.48488441	
## ROBBERY	0.69601745	1.000000000	0.80600736	
## THEFT	0.48488441	0.806007365	1.00000000	

```
x2 = model.matrix(trips~., dmgcrime)
withcrime.lasso=cv.glmnet(x2, dmgcrime$trips, alpha=1, lambda = lam)
coef(withcrime.lasso, s="lambda.min")
```

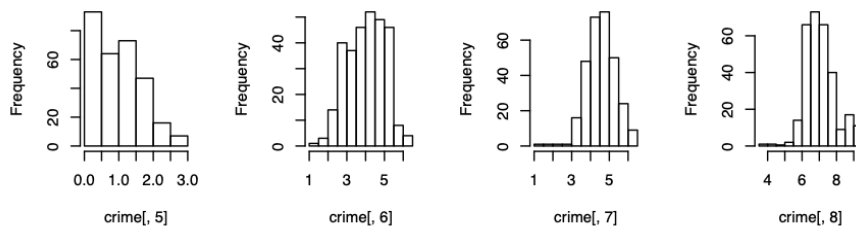
```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##              1
## (Intercept)  8.373237e+00
## (Intercept)  .
## CTA_BUS_STATIONS  .
## CTA_TRAIN_STATIONS -3.655162e-03
## BIKE_ROUTES  .
## CAPACITY  4.677113e-02
## POPULATION_SQ_MILE  3.415981e-06
## CBD  .
## MINORITY  -1.327583e+00
## EDU  3.631000e-01
## avgbf  7.700012e-07
## ASSAULT  -3.193913e-01
## BATTERY  .
## BURGLARY  -3.217129e-01
## DECEPTIVE_PRACTICE  .
## HOMICIDE  -1.835976e-02
## NARCOTICS  -7.280841e-03
## ROBBERY  .
## THEFT  5.597133e-01
```

```
#tran1=cbind(log(com1.2[,c(1:6,9)]),com1.2[,7:8],com1.2[,10:11])
par(mfrow=c(2,4))
hist(crime[,1])
hist(crime[,2])
hist(crime[,3])
hist(crime[,4])
hist(crime[,5])
hist(crime[,6])
hist(crime[,7])
hist(crime[,8])
```


Histogram of crime[, 1] Histogram of crime[, 2] Histogram of crime[, 3] Histogram of crime[, 4]



Histogram of crime[, 5] Histogram of crime[, 6] Histogram of crime[, 7] Histogram of crime[, 8]



```
#find the best performance
set.seed(12345)
train = runif(nrow(dmgsrime))<.5
fitall= lm(trips~ ASSAULT + BATTERY
+ log(DECEPTIVE_PRACTICE) + log(ROBBERY) + log(THEFT) + log((HOMICIDE+2)*5) + sqrt(NARCOTICS
+ CTA_BUS_STATIONS+ CTA_TRAIN_STATIONS +
BIKE_ROUTES + CAPACITY +I(MINORITY^2) + EDU +CBD + POPULATION_SQ_MILE
, data=dmgsrime, subset = train)
fitstepwise= step(fitall)

## Start: AIC=-197.38
## trips ~ ASSAULT + BATTERY + log(DECEPTIVE_PRACTICE) + log(ROBBERY) +
## log(THEFT) + log((HOMICIDE + 2) * 5) + sqrt(NARCOTICS) +
## BURGLARY + log(avgbf) + CTA_BUS_STATIONS + CTA_TRAIN_STATIONS +
## BIKE_ROUTES + CAPACITY + I(MINORITY^2) + EDU + CBD + POPULATION_SQ_MILE
##
##
```

	Df	Sum of Sq	RSS	AIC
## - BIKE_ROUTES	1	0.0115	23.013	-199.32
## - EDU	1	0.0404	23.042	-199.15
## - CTA_BUS_STATIONS	1	0.0483	23.050	-199.10
## - CTA_TRAIN_STATIONS	1	0.0487	23.051	-199.10
## - POPULATION_SQ_MILE	1	0.1530	23.155	-198.50
## - CBD	1	0.2688	23.271	-197.84
## - log(THEFT)	1	0.2754	23.277	-197.80
## <none>			23.002	-197.38
## - log(DECEPTIVE_PRACTICE)	1	0.3522	23.354	-197.36

```

## - log((HOMICIDE + 2) * 5) 1 0.4668 23.469 -196.71
## - log(ROBBERY) 1 0.6687 23.671 -195.57
## - sqrt(NARCOTICS) 1 0.9605 23.962 -193.94
## - ASSAULT 1 1.3746 24.377 -191.66
## - BATTERY 1 1.9149 24.917 -188.75
## - BURGLARY 1 2.2661 25.268 -186.89
## - log(avgbf) 1 2.5418 25.544 -185.44
## - CAPACITY 1 4.4908 27.493 -175.66
## - I(MINORITY^2) 1 8.0944 31.096 -159.28
##
## Step: AIC=-199.32
## trips ~ ASSAULT + BATTERY + log(DECEPTIVE_PRACTICE) + log(ROBBERY) +
## log(THEFT) + log((HOMICIDE + 2) * 5) + sqrt(NARCOTICS) +
## BURGLARY + log(avgbf) + CTA_BUS_STATIONS + CTA_TRAIN_STATIONS +
## CAPACITY + I(MINORITY^2) + EDU + CBD + POPULATION_SQ_MILE
##
## Df Sum of Sq RSS AIC
## - CTA_TRAIN_STATIONS 1 0.0433 23.057 -201.07
## - EDU 1 0.0454 23.059 -201.06
## - CTA_BUS_STATIONS 1 0.0584 23.072 -200.98
## - POPULATION_SQ_MILE 1 0.1505 23.164 -200.45
## - CBD 1 0.2667 23.280 -199.79
## - log(THEFT) 1 0.2761 23.290 -199.73
## - log(DECEPTIVE_PRACTICE) 1 0.3464 23.360 -199.33
## <none> 23.013 -199.32
## - log((HOMICIDE + 2) * 5) 1 0.4575 23.471 -198.70
## - log(ROBBERY) 1 0.6701 23.683 -197.50
## - sqrt(NARCOTICS) 1 0.9828 23.996 -195.76
## - ASSAULT 1 1.5251 24.538 -192.78
## - BATTERY 1 1.9632 24.977 -190.43
## - BURGLARY 1 2.2648 25.278 -188.83
## - log(avgbf) 1 2.7475 25.761 -186.32
## - CAPACITY 1 4.4793 27.493 -177.66
## - I(MINORITY^2) 1 8.2626 31.276 -160.52
##
## Step: AIC=-201.07
## trips ~ ASSAULT + BATTERY + log(DECEPTIVE_PRACTICE) + log(ROBBERY) +
## log(THEFT) + log((HOMICIDE + 2) * 5) + sqrt(NARCOTICS) +
## BURGLARY + log(avgbf) + CTA_BUS_STATIONS + CAPACITY + I(MINORITY^2) +
## EDU + CBD + POPULATION_SQ_MILE
##
## Df Sum of Sq RSS AIC
## - EDU 1 0.0396 23.096 -202.84
## - CTA_BUS_STATIONS 1 0.1072 23.164 -202.45
## - POPULATION_SQ_MILE 1 0.2066 23.263 -201.88
## - log(THEFT) 1 0.2852 23.342 -201.43
## <none> 23.057 -201.07
## - log(DECEPTIVE_PRACTICE) 1 0.3690 23.426 -200.96

```

```

## - log((HOMICIDE + 2) * 5) 1 0.4235 23.480 -200.65
## - CBD 1 0.5623 23.619 -199.86
## - log(ROBBERY) 1 0.6426 23.699 -199.41
## - sqrt(NARCOTICS) 1 1.2706 24.327 -195.93
## - ASSAULT 1 1.5859 24.643 -194.22
## - BURGLARY 1 2.2743 25.331 -190.56
## - BATTERY 1 2.2872 25.344 -190.49
## - log(avgbf) 1 2.7569 25.814 -188.05
## - CAPACITY 1 4.5909 27.648 -178.92
## - I(MINORITY^2) 1 8.2668 31.323 -162.32
##
## Step: AIC=-202.84
## trips ~ ASSAULT + BATTERY + log(DECEPTIVE_PRACTICE) + log(ROBBERY) +
## log(THEFT) + log((HOMICIDE + 2) * 5) + sqrt(NARCOTICS) +
## BURGLARY + log(avgbf) + CTA_BUS_STATIONS + CAPACITY + I(MINORITY^2) +
## CBD + POPULATION_SQ_MILE
##
## Df Sum of Sq RSS AIC
## - CTA_BUS_STATIONS 1 0.0986 23.195 -204.27
## - POPULATION_SQ_MILE 1 0.1736 23.270 -203.84
## - log(THEFT) 1 0.2892 23.386 -203.19
## <none> 23.096 -202.84
## - log(DECEPTIVE_PRACTICE) 1 0.3647 23.461 -202.76
## - log((HOMICIDE + 2) * 5) 1 0.4513 23.548 -202.27
## - CBD 1 0.5900 23.686 -201.49
## - log(ROBBERY) 1 0.6730 23.769 -201.02
## - sqrt(NARCOTICS) 1 1.2478 24.344 -197.84
## - ASSAULT 1 1.5687 24.665 -196.10
## - BATTERY 1 2.3071 25.403 -192.18
## - BURGLARY 1 2.3076 25.404 -192.18
## - log(avgbf) 1 2.7399 25.836 -189.93
## - CAPACITY 1 4.5629 27.659 -180.86
## - I(MINORITY^2) 1 8.2712 31.367 -164.13
##
## Step: AIC=-204.27
## trips ~ ASSAULT + BATTERY + log(DECEPTIVE_PRACTICE) + log(ROBBERY) +
## log(THEFT) + log((HOMICIDE + 2) * 5) + sqrt(NARCOTICS) +
## BURGLARY + log(avgbf) + CAPACITY + I(MINORITY^2) + CBD +
## POPULATION_SQ_MILE
##
## Df Sum of Sq RSS AIC
## - POPULATION_SQ_MILE 1 0.1482 23.343 -205.43
## - log(THEFT) 1 0.3501 23.545 -204.28
## <none> 23.195 -204.27
## - log(DECEPTIVE_PRACTICE) 1 0.4615 23.656 -203.65
## - log((HOMICIDE + 2) * 5) 1 0.4628 23.658 -203.65
## - CBD 1 0.6548 23.850 -202.57
## - log(ROBBERY) 1 0.6707 23.866 -202.48

```

```

## - sqrt(NARCOTICS)          1    1.2172 24.412 -199.47
## - ASSAULT                  1    1.7282 24.923 -196.72
## - BATTERY                  1    2.2579 25.453 -193.92
## - BURGLARY                 1    2.6700 25.865 -191.78
## - log(avgbf)               1    2.8206 26.015 -191.01
## - CAPACITY                 1    4.5428 27.738 -182.49
## - I(MINORITY^2)           1    8.2897 31.485 -165.63
##
## Step:  AIC=-205.43
## trips ~ ASSAULT + BATTERY + log(DECEPTIVE_PRACTICE) + log(ROBBERY) +
##       log(THEFT) + log((HOMICIDE + 2) * 5) + sqrt(NARCOTICS) +
##       BURGLARY + log(avgbf) + CAPACITY + I(MINORITY^2) + CBD
##
##              Df Sum of Sq    RSS    AIC
## <none>                23.343 -205.43
## - log(THEFT)          1    0.3751 23.718 -205.31
## - log(DECEPTIVE_PRACTICE) 1    0.4158 23.759 -205.08
## - log((HOMICIDE + 2) * 5) 1    0.5086 23.852 -204.56
## - log(ROBBERY)        1    0.6273 23.970 -203.90
## - CBD                 1    0.6371 23.980 -203.84
## - sqrt(NARCOTICS)     1    1.1017 24.445 -201.29
## - ASSAULT             1    1.8920 25.235 -197.06
## - BATTERY             1    2.2505 25.594 -195.19
## - BURGLARY            1    2.5222 25.865 -193.78
## - log(avgbf)          1    2.7231 26.066 -192.75
## - CAPACITY            1    4.5626 27.906 -183.68
## - I(MINORITY^2)       1    8.3664 31.709 -166.69
yhatstw = predict(fitstepwise, dmcrime[!train,])
mean((dmcrime$trips[!train] - yhatstw)^2)      # MSE=0.2378

## [1] 0.2362196
summary(fitstepwise)

##
## Call:
## lm(formula = trips ~ ASSAULT + BATTERY + log(DECEPTIVE_PRACTICE) +
##     log(ROBBERY) + log(THEFT) + log((HOMICIDE + 2) * 5) + sqrt(NARCOTICS) +
##     BURGLARY + log(avgbf) + CAPACITY + I(MINORITY^2) + CBD, data = dmcrime,
##     subset = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.20405 -0.25034  0.04595  0.27144  1.36659
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.134946   1.466084   4.185 5.47e-05 ***

```

```
## ASSAULT -0.643555 0.206354 -3.119 0.002275 **
## BATTERY 0.839063 0.246683 3.401 0.000911 ***
## log(DECEPTIVE_PRACTICE) -1.534933 1.049856 -1.462 0.146344
## log(ROBBERY) 1.065775 0.593497 1.796 0.075050 .
## log(THEFT) 2.078658 1.496852 1.389 0.167500
## log((HOMICIDE + 2) * 5) -0.378426 0.234029 -1.617 0.108503
## sqrt(NARCOTICS) -0.795019 0.334066 -2.380 0.018897 *
## BURGLARY -0.399642 0.110985 -3.601 0.000462 ***
## log(avgbf) 0.277958 0.074291 3.741 0.000282 ***
## CAPACITY 0.045608 0.009417 4.843 3.85e-06 ***
## I(MINORITY^2) -1.613821 0.246078 -6.558 1.44e-09 ***
## CBD -0.334516 0.184839 -1.810 0.072835 .
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4411 on 120 degrees of freedom
## Multiple R-squared: 0.8113, Adjusted R-squared: 0.7924
## F-statistic: 42.99 on 12 and 120 DF, p-value: < 2.2e-16
```

```
vif(fitstepwise)
```

```
##          ASSAULT          BATTERY log(DECEPTIVE_PRACTICE)
##          15.247552          22.402693          17.763554
##          log(ROBBERY)          log(THEFT) log((HOMICIDE + 2) * 5)
##          5.765965          20.683153          2.666346
##          sqrt(NARCOTICS)          BURGLARY          log(avgbf)
##          4.634383          2.809277          5.901447
##          CAPACITY          I(MINORITY^2)          CBD
##          1.860583          3.295204          4.175639
```

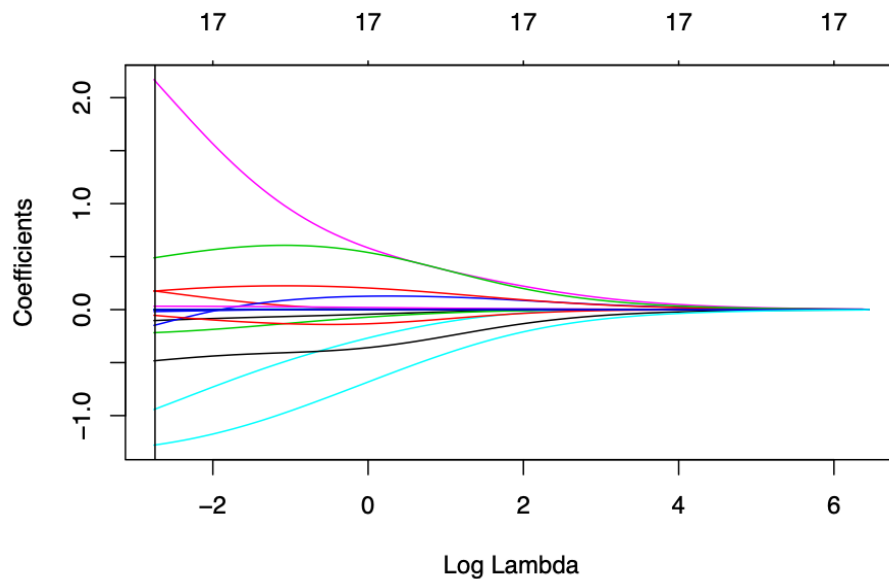
```
fitred= lm(trips~
# ASSAULT
+ BATTERY
#+ log(DECEPTIVE_PRACTICE)
+ ROBBERY
# +log(THEFT)
+ log((HOMICIDE+2)*5)
+ sqrt(NARCOTICS)
+BURGLARY
+log(avgbf)
#+ CAPACITY +I(MINORITY^2) + EDU +CBD
, data=dmgcrime, subset = train)
fitred
```

```
##
## Call:
## lm(formula = trips ~ +BATTERY + ROBBERY + log((HOMICIDE + 2) *
## 5) + sqrt(NARCOTICS) + BURGLARY + log(avgbf), data = dmcrime,
## subset = train)
```

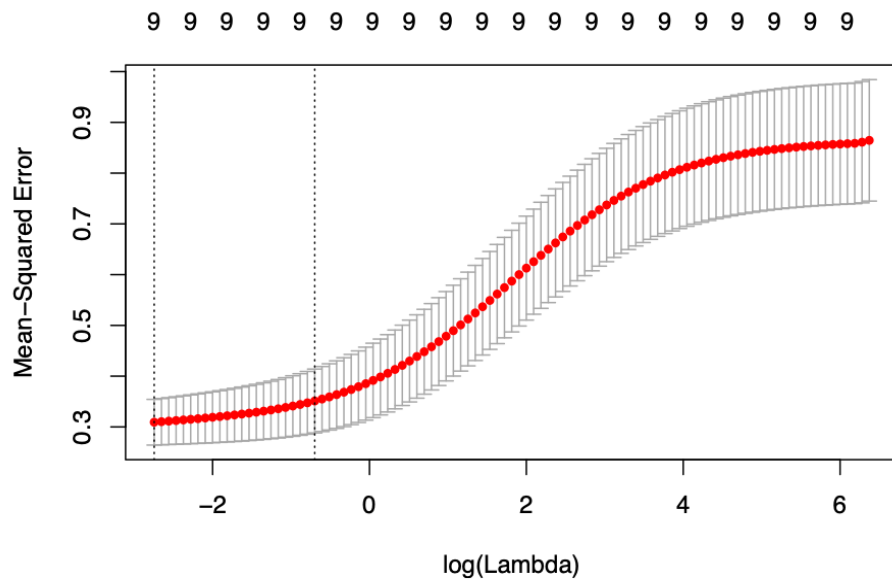
```
##
## Coefficients:
##          (Intercept)          BATTERY          ROBBERY
##          6.58935          0.23567         -0.01873
## log((HOMICIDE + 2) * 5)      sqrt(NARCOTICS)      BURGLARY
##          -0.93399          -0.95357          0.13329
##          log(avgbf)
##          0.52297

#Ridge
dmgtrans=cbind(dmg[,c(1:6,8:10)],dmg[,7]^2)
dmgcrime2= cbind(crime[,1:3],log(crime[,c(4,7:8)]),log((crime[,5]+2)*5),sqrt(crime[,6]),dmgt
train2 = runif(nrow(dmgcrime2))<.5
xr = model.matrix(trips ~ ., dmgcrime2)
fit.ridge = glmnet(xr[train2,], dmgcrime2$trips[train2], alpha=0)
plot(fit.ridge, xvar="lambda")
fit.cv.rd = cv.glmnet(x[train2,], dmgcrime2$trips[train2], alpha=0) # find optimal lambda
fit.cv.rd$lambda.min          # optimal value of lambda

## [1] 0.06425702
abline(v=log(fit.cv.rd$lambda.min))
```



```
plot(fit.cv.rd)          # plot MSE vs. log(lambda)
```

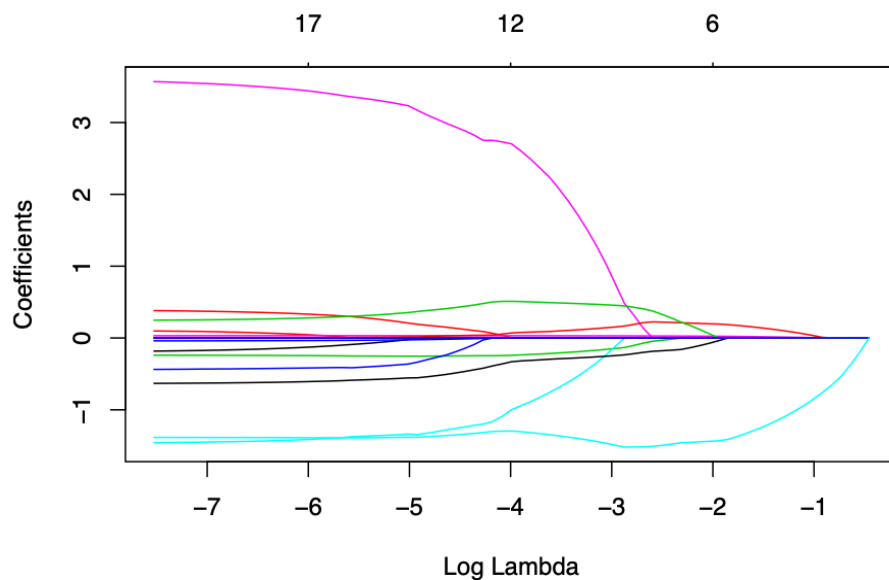


```
yhat = predict(fit.ridge, s=fit.cv.rd$lambda.min, newx=xr[!train2,]) # find yhat for best n
mean((dmgcrime2$trips[!train2] - yhat)^2) # MSE=0.27
```

```
## [1] 0.2722699
```

```
#Lasso
```

```
fit.lasso = glmnet(xr[train2,], dmgcrime2$trips[train2], alpha=1)
plot(fit.lasso, xvar="lambda")
```



```
fit.cv.la = cv.glmnet(xr[train2,], dmcrime2$trips[train2], alpha=1)
yhatla = predict(fit.lasso, s=fit.cv.la$lambda.min, newx=xr[!train2,])
mean((dmcrime2$trips[!train2] - yhatla)^2) # MSE=0.27177
```

```
## [1] 0.271772
```

```
summary(fit.lasso)
```

```
##          Length Class      Mode
## a0           77  -none-  numeric
## beta        1386 dgMatrix S4
## df           77  -none-  numeric
## dim           2  -none-  numeric
## lambda        77  -none-  numeric
## dev.ratio     77  -none-  numeric
## nulldev        1  -none-  numeric
## npasses        1  -none-  numeric
## jerr           1  -none-  numeric
## offset         1  -none- logical
## call           4  -none-   call
## nobs           1  -none-  numeric
```

```
coef(fit.cv.la,s="lambda.min")
```

```
## 19 x 1 sparse Matrix of class "dgMatrix"
##                                     1
## (Intercept)                        6.909980e+00
## (Intercept)                        .
```



```

## ASSAULT -1.362651e-01
## BATTERY 3.396608e-01
## BURGLARY -2.446829e-01
## DECEPTIVE_PRACTICE -4.197462e-01
## ROBBERY -1.420033e+00
## THEFT 3.459512e+00
## `log((crime[, 5] + 2) * 5)` -6.112912e-01
## `sqrt(crime[, 6])` 5.394357e-02
## CTA_BUS_STATIONS .
## CTA_TRAIN_STATIONS -3.634209e-02
## BIKE_ROUTES -7.454278e-03
## CAPACITY 3.229239e-02
## POPULATION_SQ_MILE 6.465691e-06
## CBD 2.681339e-03
## EDU 2.738475e-01
## avgbf 2.798785e-06
## `dmg[, 7]^2` -1.387952e+00

#random forest
library(gam)

## Loading required package: splines
## Loaded gam 1.16

library(randomForest)

## randomForest 4.6-14

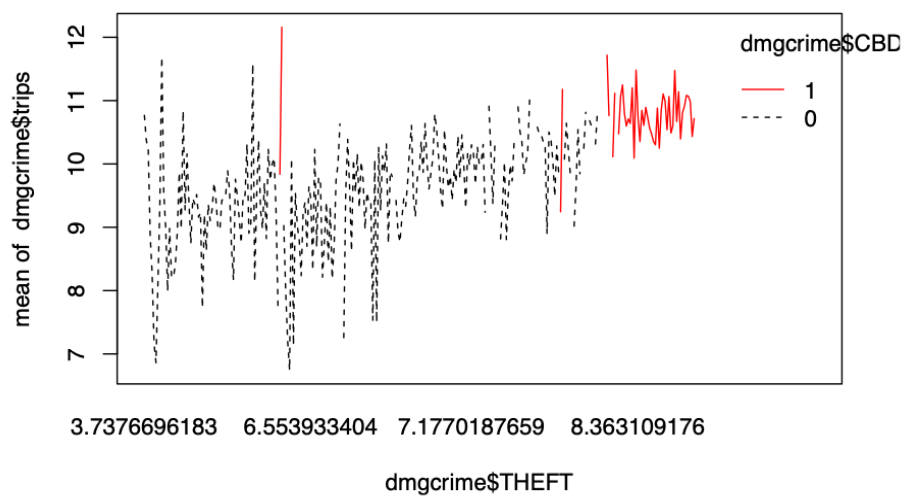
## Type rfNews() to see new features/changes/bug fixes.

colnames(dmgcrime2)[7] <- "homocidetrams"
colnames(dmgcrime2)[8] <- "narcoticstrams"
colnames(dmgcrime2)[18] <- "combineddmgtrams"
fitrf=randomForest(trips ~ ., data=dmgcrime2, importance=T)
fitrf

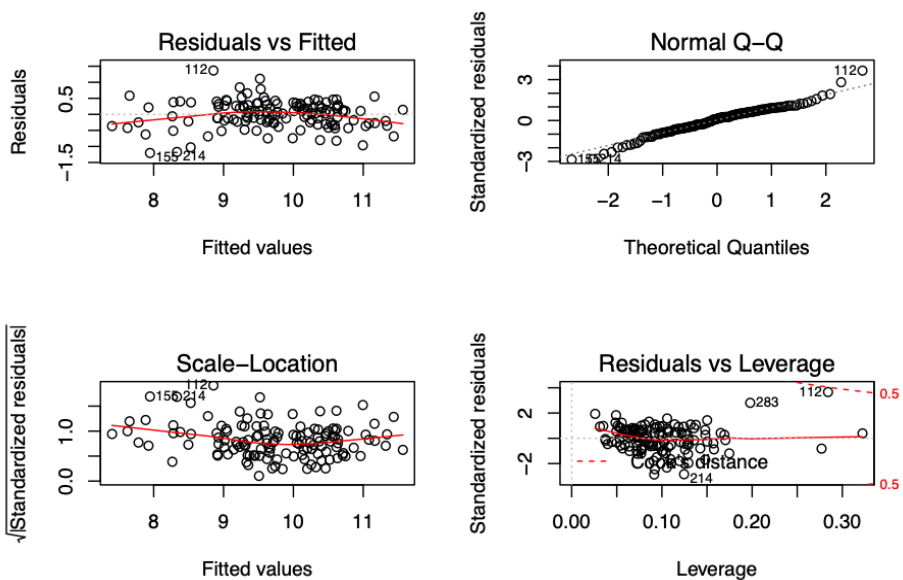
##
## Call:
## randomForest(formula = trips ~ ., data = dmgcrime2, importance = T)
## Type of random forest: regression
## Number of trees: 500
## No. of variables tried at each split: 5
##
## Mean of squared residuals: 0.2266479
## % Var explained: 75.3

```

```
interaction.plot(dmccrime$THEFT, dmccrime$CBD, dmccrime$strips, col=1:2)
```



```
par(mfrow=c(2,2))
plot(fitstepwise)
```



```
vif(fitstepwise)
```

```
##          ASSAULT          BATTERY log(DECEPTIVE_PRACTICE)
##          15.247552          22.402693          17.763554
```

##	log(ROBBERY)	log(THEFT)	log((HOMICIDE + 2) * 5)
##	5.765965	20.683153	2.666346
##	sqrt(NARCOTICS)	BURGLARY	log(avgbf)
##	4.634383	2.809277	5.901447
##	CAPACITY	I(MINORITY^2)	CBD
##	1.860583	3.295204	4.175639