Regression Competition

Claire Jellison

9/25/2019

```r
d <- read.csv("http://andrewpbray.github.io/data/crime-train.csv")
#summary(d)
library(ggplot2)
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-16
```

Fitting the model

```r
group_D_fit <- function(training_data) {
  # this function should be self-contained, so include
  # any packages you need and any data processing that
  # you do.

  # run lm() to fit your model.

  # on the last line, simply put m1, your final model.
  # this will return it as output.
  m11 <- lm(data = d, ViolentCrimesPerPop ~  factor(state) + racePctWhite + pctUrban  + PctEmploy + Male
  m11
}
```

Computing MSE

```r
group_D_MSE <- function(model, data){
  n <- nrow(data)
  ys <- data$ViolentCrimesPerPop
  y_hats <- predict(model, data)
  residuals <- y_hats - ys
  MSE <- sum(residuals^2)/n
  MSE
}
```

```r
#install.packages("Hmisc")
#library("Hmisc")
bool <- sapply(d, is.numeric)
num_only <- d[,bool]

matrix <- cor(num_only)
matrix <- matrix[,"ViolentCrimesPerPop"]
matrix
```

```
##                state          population       householdsize
##          -0.19867891          0.37066532         -0.03861389
##         racepctblack        racePctWhite        racePctAsian
##           0.61396953         -0.68885560          0.07698748
##          racePctHisp         agePct12t21         agePct12t29
##           0.37526637          0.03858255          0.14025298
##          agePct16t24         agePct65up            numbUrban
##           0.08207684          0.06033189          0.37483077
##             pctUrban           medIncome            pctWWage
##           0.15474311         -0.41384940         -0.29962918
##         pctWFarmSelf          pctWInvInc          pctWSocSec
##          -0.19174832         -0.57620572          0.12082657
##          pctWPubAsst          pctWRetire           medFamInc
##           0.59227263         -0.08997653         -0.42398363
##            perCapInc         whitePerCap         blackPerCap
##          -0.34862287         -0.22204504         -0.26352732
##          indianPerCap         AsianPerCap         OtherPerCap
##          -0.09625260         -0.16996957         -0.10880239
##           HispPerCap         NumUnderPov      PctPopUnderPov
##          -0.25883292          0.45390778          0.53779772
##      PctLess9thGrade        PctNotHSGrad        PctBSorMore
##           0.45567237          0.51567720         -0.33150986
##        PctUnemployed           PctEmploy         PctEmplManu
##           0.53731986         -0.32740291          0.01761068
##       PctEmplProfServ        PctOccupManu    PctOccupMgmtProf
##          -0.10331277          0.33476343         -0.36141065
##       MalePctDivorce       MalePctNevMarr        FemalePctDiv
##           0.54018366          0.31010841          0.56419032
##          TotalPctDiv          PersPerFam          PctFam2Par
##           0.56515609          0.17194429         -0.70528060
##          PctKids2Par     PctYoungKids2Par          PctTeen2Par
##          -0.73929315         -0.66897706         -0.65329249
##  PctWorkMomYoungKids          PctWorkMom            NumIlleg
##          -0.06341328         -0.19196402          0.49542797
##             PctIlleg            NumImmig      PctImmigRecent
##           0.74352441          0.32492255          0.17054361
##         PctImmigRec5        PctImmigRec8       PctImmigRec10
##           0.20884691          0.26657334          0.31277465
##       PctRecentImmig         PctRecImmig5         PctRecImmig8
##           0.28087817          0.29501202          0.30279580
##        PctRecImmig10     PctSpeakEnglOnly PctNotSpeakEnglWell
##           0.31827795         -0.32244668          0.38597995
##      PctLargHouseFam    PctLargHouseOccup    PersPerOccupHous
##           0.43062883          0.33953706         -0.01870530
##     PersPerOwnOccHous   PersPerRentOccHous     PctPersOwnOccup
##          -0.08508545          0.25020053         -0.53377983
##       PctPersDenseHous      PctHousLess3BR            MedNumBR
##           0.50540251          0.49694082         -0.39714251
##           HousVacant         PctHousOccup       PctHousOwnOcc
##           0.40353909         -0.26979027         -0.48147523
##      PctVacantBoarded       PctVacMore6Mos      MedYrHousBuilt
##           0.50895955          0.02399628         -0.17513688
##        PctHousNoPhone       PctWOFullPlumb      OwnOccLowQuart
##           0.48488275          0.38798494         -0.19113369
```

```
##          OwnOccMedVal          OwnOccHiQuart              RentLowQ
##           -0.17423559            -0.16223122           -0.23616004
##            RentMedian              RentHighQ               MedRent
##           -0.22219135            -0.21184794           -0.22753962
##       MedRentPctHousInc      MedOwnCostPctInc MedOwnCostPctIncNoMtg
##            0.32846021             0.06093440            0.04860191
##          NumInShelters              NumStreet         PctForeignBorn
##            0.39694511             0.35737606            0.25346691
##       PctBornSameState         PctSameHouse85          PctSameCity85
##           -0.06845675            -0.11254574            0.13192565
##         PctSameState85               LandArea               PopDens
##            0.03032880             0.18847682            0.34196833
##         PctUsePubTrans   LemasPctOfficDrugUn    ViolentCrimesPerPop
##            0.21336825             0.36681475            1.00000000
```

Started by examining the correlation between the predictor variables and ViolentCrimesPerPop.

Used a lasso regression approach to decide on the best predictors to include in the model.

LASSO

```
#set.seed(489)
#x_vars <- model.matrix(ViolentCrimesPerPop~. , num_only)[,-1]
#y_var <- num_only$ViolentCrimesPerPop
#lambda_seq <- 10^seq(2, -2, by = -.1)
#train = sample(1:nrow(x_vars), nrow(x_vars)/2)
#test = (-train)
#ytest = y[test]
#cv_output <- cv.glmnet(x_vars[train,], y_var[train],
                       #alpha = 1, lambda = lambda_seq)
#best_lam <- cv_output$lambda.min
#lasso.mod <- glmnet(x_vars[train,], y_var[train], alpha = 1, lambda = lambda)
#lasso.pred <- predict(lasso.mod, s = bestlam, newx = x_vars[test,])
#x <- cor(num_only)
#lasso_best <- glmnet(x_vars[train,], y_var[train], alpha = 1, lambda = best_lam)
#pred <- predict(lasso_best, s = best_lam, newx = x_vars[test,])
#final <- cbind(y_var[test], pred)
#head(final)
```

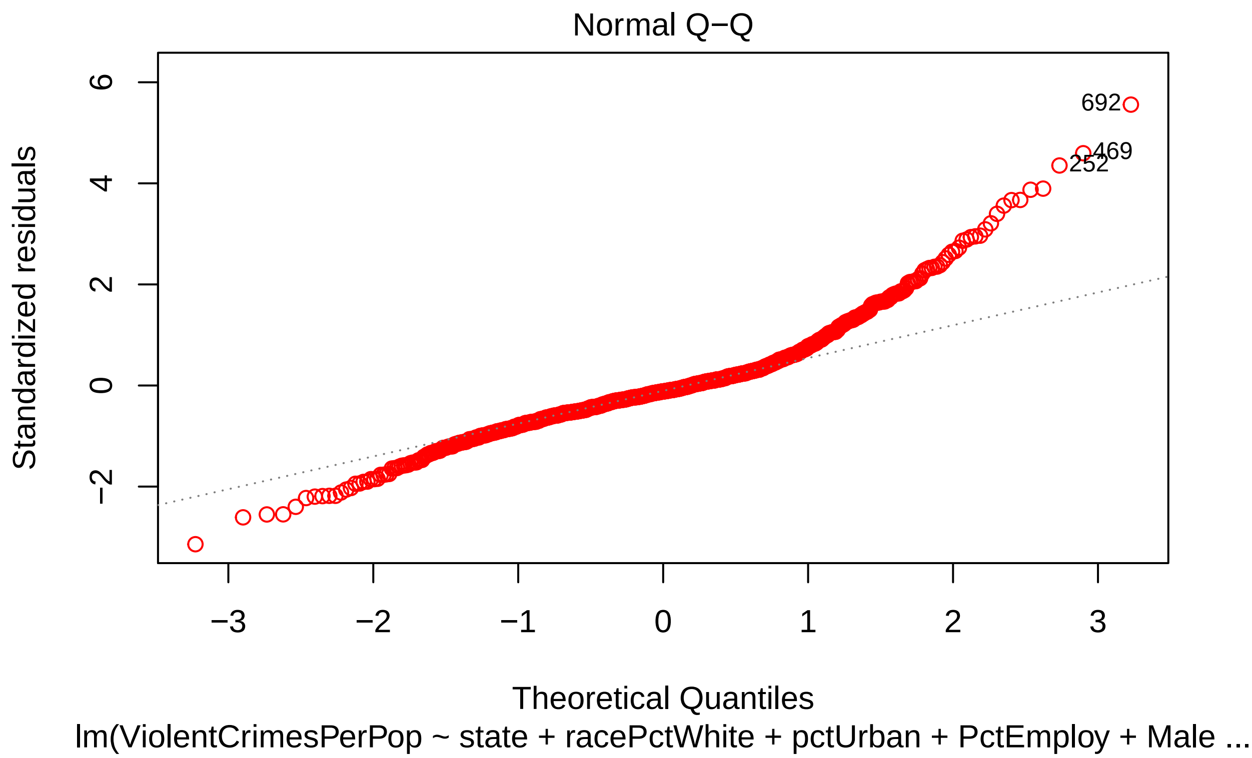
Checked for non linear relationships with the chosen variables and added a couple squared terms where it appeared appropriate.

```
MalePctDivorce2 <- (d$MalePctDivorce)^2
PctKids2Par2 <-(d$PctKids2Par)^2
PctImmigRec82 <- (d$PctImmigRec8)^2
#pctdensehouse2 <- (d$PctPersDenseHous)^2
#m2 <- lm(data = d, ViolentCrimesPerPop ~ state + racePctWhite + pctUrban + PctUnemployed + PctEmploy +
#m3 <- lm(data = d, ViolentCrimesPerPop ~  state + racePctWhite + pctUrban + PctUnemployed + MalePctDiv
#m4 <- lm(data = d, ViolentCrimesPerPop ~  state + racePctWhite + pctUrban + PctUnemployed + MalePctDiv
#ggplot(d, (aes(x = LemasPctOfficDrugUn , y = ViolentCrimesPerPop))) + geom_point(position = "jitter")
```

Added some interaction terms that impoved the R^2, adjusted R^2 and MSE. Looked at plots of residuals.

```
m7 <- lm(data = d, ViolentCrimesPerPop ~  state + racePctWhite + pctUrban  + PctEmploy + MalePctDivorce


plot(m7, which=2, col=c("red"))
```




```
#summary(m7)
#group_D_MSE(m7, d)
```

Added back some variables that were not included in the lasso, but improved the model.

```
m11 <- lm(data = d, ViolentCrimesPerPop ~  factor(state) + racePctWhite + pctUrban  + PctEmploy + MaleP
summary(m11)
```

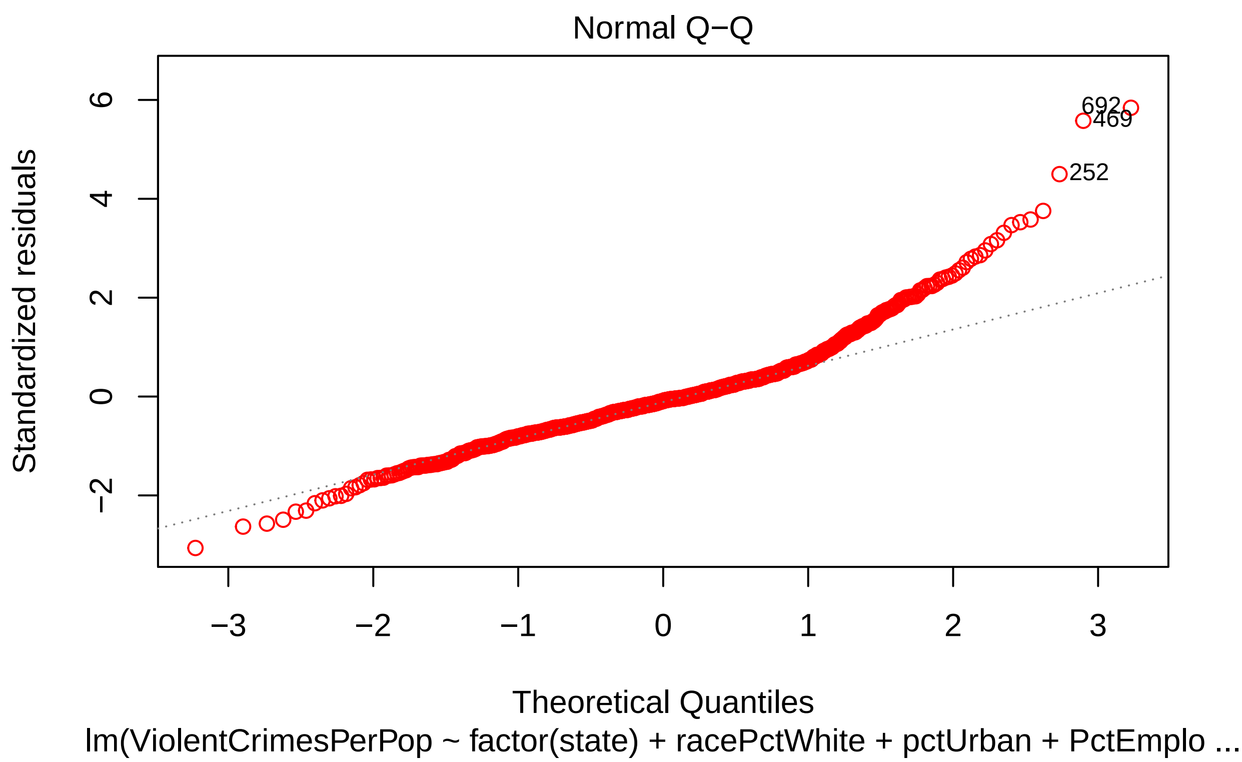
```
##
## Call:
## lm(formula = ViolentCrimesPerPop ~ factor(state) + racePctWhite +
##     pctUrban + PctEmploy + MalePctDivorce + MalePctDivorce^2 +
##     PctKids2Par + PctKids2Par^2 + PctWorkMom + PctPersDenseHous +
##     NumStreet + PctVacantBoarded + PctImmigRec8 + PctImmigRec8^2 +
##     PctIlleg + PctHousOccup + PctWorkMom * MalePctDivorce + pctUrban *
##     racePctWhite + PctEmploy * racePctWhite + pctUrban * PctHousOccup +
##     PctEmploy * pctUrban + PctIlleg * PctEmploy + PctImmigRec8 *
##     PctVacantBoarded + PctNotHSGrad + PctLess9thGrade + NumInShelters +
##     PctEmploy * pctUrban + PctIlleg * PctEmploy + PctImmigRec8 *
##     PctVacantBoarded + PctNotHSGrad + PctLess9thGrade + NumInShelters,
##     data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35269 -0.07361 -0.01086  0.04494  0.71466
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.4112001  0.1387024   2.965 0.003128 **
## factor(state)2               -0.1421297  0.1353742  -1.050 0.294108
## factor(state)4               -0.1029670  0.0637838  -1.614 0.106888
```

```
## factor(state)5                -0.1578449  0.0498027  -3.169 0.001591 **
## factor(state)6                -0.1027411  0.0386599  -2.658 0.008042 **
## factor(state)8                -0.1336471  0.0511594  -2.612 0.009175 **
## factor(state)9                -0.1822639  0.0415871  -4.383 1.34e-05 ***
## factor(state)12               -0.0189972  0.0415190  -0.458 0.647407
## factor(state)13               -0.1866278  0.0479136  -3.895 0.000107 ***
## factor(state)16               -0.1531316  0.0969082  -1.580 0.114497
## factor(state)18               -0.1345293  0.0431615  -3.117 0.001899 **
## factor(state)19               -0.0410532  0.0622479  -0.660 0.509774
## factor(state)21               -0.1454542  0.0531040  -2.739 0.006311 **
## factor(state)22               -0.2280464  0.0546191  -4.175 3.33e-05 ***
## factor(state)23               -0.2238731  0.0566277  -3.953 8.45e-05 ***
## factor(state)24               -0.1314355  0.0828686  -1.586 0.113153
## factor(state)25               -0.0808962  0.0380796  -2.124 0.033970 *
## factor(state)27               -0.1858728  0.0971838  -1.913 0.056188 .
## factor(state)28               -0.2550116  0.0527116  -4.838 1.60e-06 ***
## factor(state)29               -0.1091633  0.0468878  -2.328 0.020173 *
## factor(state)32               -0.1573377  0.0982231  -1.602 0.109620
## factor(state)33               -0.1858497  0.0551858  -3.368 0.000797 ***
## factor(state)34               -0.1871036  0.0353853  -5.288 1.64e-07 ***
## factor(state)36               -0.2154638  0.0436754  -4.933 1.00e-06 ***
## factor(state)37               -0.1385367  0.0435482  -3.181 0.001528 **
## factor(state)38               -0.2023313  0.0679039  -2.980 0.002981 **
## factor(state)39               -0.1931397  0.0380657  -5.074 4.94e-07 ***
## factor(state)40               -0.1926520  0.0457180  -4.214 2.82e-05 ***
## factor(state)41               -0.2018225  0.0563507  -3.582 0.000364 ***
## factor(state)42               -0.1627450  0.0398160  -4.087 4.84e-05 ***
## factor(state)44               -0.1548377  0.0505245  -3.065 0.002259 **
## factor(state)45               -0.0474661  0.0519182  -0.914 0.360886
## factor(state)46               -0.1363641  0.0981100  -1.390 0.164976
## factor(state)47               -0.0881303  0.0444084  -1.985 0.047566 *
## factor(state)48               -0.1218307  0.0382073  -3.189 0.001490 **
## factor(state)49               -0.1575874  0.0549081  -2.870 0.004222 **
## factor(state)50               -0.2356500  0.0962349  -2.449 0.014570 *
## factor(state)51               -0.2398559  0.0459355  -5.222 2.31e-07 ***
## factor(state)53               -0.1737736  0.0477565  -3.639 0.000293 ***
## factor(state)54               -0.1928352  0.0619917  -3.111 0.001939 **
## factor(state)55               -0.1666619  0.0426808  -3.905 0.000103 ***
## factor(state)56               -0.1785575  0.0827548  -2.158 0.031277 *
## racePctWhite                  -0.2385404  0.1107573  -2.154 0.031586 *
## pctUrban                       0.1952759  0.0580465   3.364 0.000808 ***
## PctEmploy                     -0.3151692  0.2056409  -1.533 0.125799
## MalePctDivorce                 0.2683803  0.0966151   2.778 0.005612 **
## PctKids2Par                   -0.0824103  0.0968535  -0.851 0.395115
## PctWorkMom                    -0.0172031  0.0766088  -0.225 0.822386
## PctPersDenseHous              -0.0284675  0.0552688  -0.515 0.606657
## NumStreet                      0.1009727  0.0536773   1.881 0.060352 .
## PctVacantBoarded               0.0192847  0.0532844   0.362 0.717515
## PctImmigRec8                   0.0009149  0.0372881   0.025 0.980432
## PctIlleg                       0.2518983  0.1244616   2.024 0.043341 *
## PctHousOccup                  -0.1160975  0.0540048  -2.150 0.031899 *
## PctNotHSGrad                   0.2016649  0.1011110   1.994 0.046467 *
## PctLess9thGrade               -0.0852119  0.0861635  -0.989 0.323010
## NumInShelters                  0.0843425  0.0623508   1.353 0.176563
```

```
## MalePctDivorce:PctWorkMom     -0.1260923  0.1554167  -0.811 0.417446
## racePctWhite:pctUrban         -0.2597827  0.0564834  -4.599 4.99e-06 ***
## racePctWhite:PctEmploy         0.4644071  0.1996022   2.327 0.020254 *
## pctUrban:PctHousOccup          0.1379663  0.0635187   2.172 0.030170 *
## pctUrban:PctEmploy            -0.0930252  0.0677950  -1.372 0.170433
## PctEmploy:PctIlleg             0.0394544  0.2279420   0.173 0.862628
## PctVacantBoarded:PctImmigRec8  0.1746203  0.1163871   1.500 0.133955
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1257 on 736 degrees of freedom
## Multiple R-squared:  0.7361, Adjusted R-squared:  0.7136
## F-statistic: 32.59 on 63 and 736 DF,  p-value: < 2.2e-16
```

```
plot(m11, which=2, col=c("red"))
```

```
## Warning: not plotting observations with leverage one:
##   27
```



Assessed final two models looking by two different criterion statistics.

```
AIC(m7)
```

```
## [1] -953.1708
```

```
BIC(m7)
```

```
## [1] -798.5786
```

```
AIC(m11)
```

```
## [1] -984.1071
```

```r
BIC(m11)
```

```
## [1] -679.6074
```