Crime and Communities

Group Member 1 Name: Khang V. Tran **Group Member 1 SID:** 25181590

Group Member 2 Name: Christian Philip Hoeck **Group Member 2 SID:** ____________________

The crime and communities dataset contains crime data from communities in the United States. The data combines socio-economic data from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR. More details can be found at https://archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized.

The dataset contains 125 columns total; $p = 124$ predictive and 1 target (ViolentCrimesPerPop). There are $n = 1994$ observations. These can be arranged into an $n \times p = 1994 \times 127$ feature matrix $\mathbf{X}$, and an $n \times 1 = 1994 \times 1$ response vector $\mathbf{y}$ (containing the observations of ViolentCrimesPerPop).

Once downloaded (from bCourses), the data can be loaded as follows.

```r
library(readr)
CC <- read_csv("../data_files/crime_and_communities_data.csv")
print(dim(CC))
```

```
## [1] 1994  125
```

```r
y <- CC$ViolentCrimesPerPop
X <- subset(CC, select = -c(ViolentCrimesPerPop))
```

```r
# library(dplyr)
# head(CC)
# names(CC)
# glimpse(CC)
# dim(CC)
# is.na(CC)
# str(CC)
```

Dataset exploration

In this section, you should provide a thorough exploration of the features of the dataset. Things to keep in mind in this section include:

- Which variables are categorical versus numerical?
- What are the general summary statistics of the data? How can these be visualized?
- Is the data normalized? Should it be normalized?
- Are there missing values in the data? How should these missing values be handled?
- Can the data be well-represented in fewer dimensions?

YOUR CODE GOES HERE

Examine Categorical vs. Quantitative data

Let's look at the structure of the data

```r
str(X)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    1994 obs. of  124 variables:
##  $ population          : num  11980 23123 29344 16656 140494 ...
##  $ householdsize       : num  3.1 2.82 2.43 2.4 2.45 2.6 2.45 2.46 2.62 2.54 ...
##  $ racepctblack        : num  1.37 0.8 0.74 1.7 2.51 ...
##  $ racePctWhite        : num  91.8 95.6 94.3 97.3 95.7 ...
##  $ racePctAsian        : num  6.5 3.44 3.43 0.5 0.9 1.47 0.4 1.25 0.92 0.77 ...
##  $ racePctHisp         : num  1.88 0.85 2.35 0.7 0.95 ...
##  $ agePct12t21         : num  12.5 11 11.4 12.6 18.1 ...
##  $ agePct12t29         : num  21.4 21.3 25.9 25.2 32.9 ...
##  $ agePct16t24         : num  10.9 10.5 11 12.2 20 ...
##  $ agePct65up          : num  11.3 17.2 10.3 17.6 13.3 ...
##  $ numbUrban           : num  11980 23123 29344 0 140494 ...
##  $ pctUrban            : num  100 100 100 0 100 100 100 100 100 100 ...
##  $ medIncome           : num  75122 47917 35669 20580 21577 ...
##  $ pctWWage            : num  89.2 79 82 68.2 75.8 ...
##  $ pctWFarmSelf        : num  1.55 1.11 1.15 0.24 1 0.39 0.67 2.93 0.86 1.54 ...
##  $ pctWInvInc          : num  70.2 64.1 55.7 39 41.1 ...
##  $ pctWSocSec          : num  23.6 35.5 22.2 39.5 29.3 ...
##  $ pctWPubAsst         : num  1.03 2.75 2.94 11.71 7.12 ...
##  $ pctWRetire          : num  18.4 22.9 14.6 18.3 14.1 ...
##  $ medFamInc           : num  79584 55323 42112 26501 27705 ...
##  $ perCapInc           : num  29711 20148 16946 10810 11878 ...
##  $ whitePerCap         : num  30233 20191 17103 10909 12029 ...
##  $ blackPerCap         : num  13600 18137 16644 9984 7382 ...
##  $ indianPerCap        : num  5725 0 21606 4941 10264 ...
##  $ AsianPerCap         : num  27101 20074 15528 3541 10753 ...
##  $ OtherPerCap         : num  5115 5250 5954 2451 7192 ...
##  $ HispPerCap          : num  22838 12222 8405 4391 8104 ...
##  $ NumUnderPov         : num  227 885 1389 2831 23223 ...
##  $ PctPopUnderPov      : num  1.96 3.98 4.75 17.23 17.78 ...
##  $ PctLess9thGrade     : num  5.81 5.61 2.8 11.05 8.76 ...
##  $ PctNotHSGrad        : num  9.9 13.72 9.09 33.68 23.03 ...
##  $ PctBSorMore         : num  48.2 29.9 30.1 10.8 20.7 ...
##  $ PctUnemployed       : num  2.7 2.43 4.01 9.86 5.72 4.85 8.19 4.18 8.39 7.19 ...
##  $ PctEmploy           : num  64.5 62 69.8 54.7 59 ...
##  $ PctEmplManu         : num  14.7 12.3 15.9 31.2 14.3 ...
##  $ PctEmplProfServ     : num  28.8 29.3 21.5 27.4 26.8 ...
##  $ PctOccupManu        : num  5.49 6.39 8.79 26.76 14.72 ...
##  $ PctOccupMgmtProf    : num  50.7 37.6 32.5 22.7 23.4 ...
##  $ MalePctDivorce      : num  3.67 4.23 10.1 10.98 11.4 ...
##  $ MalePctNevMarr      : num  26.4 28 25.8 28.1 33.3 ...
##  $ FemalePctDiv        : num  5.22 6.45 14.76 14.47 14.46 ...
##  $ TotalPctDiv         : num  4.47 5.42 12.55 12.91 13.04 ...
##  $ PersPerFam          : num  3.22 3.11 2.95 2.98 2.89 3.14 2.95 3 3.11 2.99 ...
##  $ PctFam2Par          : num  91.4 86.9 78.5 64 71.9 ...
##  $ PctKids2Par         : num  90.2 85.3 78.8 62.4 69.8 ...
##  $ PctYoungKids2Par    : num  95.8 96.8 92.4 65.4 79.8 ...
##  $ PctTeen2Par         : num  95.8 86.5 75.7 67.4 75.3 ...
##  $ PctWorkMomYoungKids : num  44.6 51.1 66.1 59.6 63 ...
```

```
##  $ PctWorkMom           : num  58.9 62.4 74.2 70.3 70.5 ...
##  $ NumKidsBornNeverMar  : num  31 43 164 561 1511 ...
##  $ PctKidsBornNeverMar  : num  0.36 0.24 0.88 3.84 1.58 1.18 4.66 1.64 4.71 2.47 ...
##  $ NumImmig             : num  1277 1920 1468 339 2091 ...
##  $ PctImmigRecent       : num  8.69 5.21 16.42 13.86 21.33 ...
##  $ PctImmigRec5         : num  13 8.65 23.98 13.86 30.56 ...
##  $ PctImmigRec8         : num  21 13.3 32.1 15.3 38 ...
##  $ PctImmigRec10        : num  30.9 22.5 35.6 15.3 45.5 ...
##  $ PctRecentImmig       : num  0.93 0.43 0.82 0.28 0.32 1.05 0.11 0.47 0.72 0.53 ...
##  $ PctRecImmig5         : num  1.39 0.72 1.2 0.28 0.45 1.49 0.2 0.67 1.07 1.05 ...
##  $ PctRecImmig8         : num  2.24 1.11 1.61 0.31 0.57 2.2 0.25 0.93 1.63 1.66 ...
##  $ PctRecImmig10        : num  3.3 1.87 1.78 0.31 0.68 2.55 0.29 1.07 2.31 1.94 ...
##  $ PctSpeakEnglOnly     : num  85.7 87.8 93.1 95 96.9 ...
##  $ PctNotSpeakEnglWell  : num  1.37 1.81 1.14 0.56 0.6 0.6 0.28 0.43 2.51 0.81 ...
##  $ PctLargHouseFam      : num  4.81 4.25 2.97 3.93 3.08 5.08 3.85 2.59 6.7 3.66 ...
##  $ PctLargHouseOccup    : num  4.17 3.34 2.05 2.56 1.92 3.46 2.55 1.54 4.1 2.51 ...
##  $ PersPerOccupHous     : num  2.99 2.7 2.42 2.37 2.28 2.55 2.36 2.32 2.45 2.42 ...
##  $ PersPerOwnOccHous    : num  3 2.83 2.69 2.51 2.37 2.89 2.42 2.77 2.47 2.5 ...
##  $ PersPerRentOccHous   : num  2.84 1.96 2.06 2.2 2.16 2.09 2.27 1.91 2.44 2.31 ...
##  $ PctPersOwnOccup      : num  91.5 89 64.2 58.2 57.8 ...
##  $ PctPersDenseHous     : num  0.39 1.01 2.03 1.21 2.11 1.47 1.9 1.67 6.14 3.41 ...
##  $ PctHousLess3BR       : num  11.1 23.6 47.5 45.7 53.2 ...
##  $ MedNumBR             : num  3 3 3 3 2 3 2 2 2 2 ...
##  $ HousVacant           : num  64 240 544 669 5119 ...
##  $ PctHousOccup         : num  98.4 97.2 95.7 91.2 91.8 ...
##  $ PctHousOwnOcc        : num  91 84.9 57.8 54.9 55.5 ...
##  $ PctVacantBoarded     : num  3.12 0 0.92 2.54 2.09 1.41 6.39 0.45 5.64 2.77 ...
##  $ PctVacMore6Mos       : num  37.5 18.33 7.54 57.85 26.22 ...
##  $ MedYrHousBuilt       : num  1959 1958 1976 1939 1966 ...
##  $ PctHousNoPhone       : num  0 0.31 1.55 7 6.13 ...
##  $ PctWOFullPlumb       : num  0.28 0.14 0.12 0.87 0.31 0.28 0.49 0.19 0.33 0.3 ...
##  $ OwnOccLowQuart       : num  215900 136300 74700 36400 37700 ...
##  $ OwnOccMedVal         : num  262600 164200 90400 49600 53900 ...
##  $ OwnOccHiQuart        : num  326900 199900 112000 66500 73100 ...
##  $ OwnOccQrange         : num  111000 63600 37300 30100 35400 60400 26100 39200 38800 41400 ...
##  $ RentLowQ             : num  685 467 370 195 215 463 186 241 192 234 ...
##  $ RentMedian           : num  1001 560 428 250 280 ...
##  $ RentHighQ            : num  1001 672 520 309 349 ...
##  $ RentQrange           : num  316 205 150 114 134 361 139 146 177 142 ...
##  $ MedRent              : num  1001 627 484 333 340 ...
##  $ MedRentPctHousInc    : num  23.8 27.6 24.1 28.7 26.4 24.4 26.3 25.2 29.6 23.8 ...
##  $ MedOwnCostPctInc     : num  21.1 20.7 21.7 20.6 17.3 20.8 15.1 20.7 19.4 17.1 ...
##  $ MedOwnCostPctIncNoMtg: num  14 12.5 11.6 14.5 11.7 12.5 12.2 12.8 13 12.9 ...
##  $ NumInShelters        : num  11 0 16 0 327 0 21 125 43 1 ...
##  $ NumStreet            : num  0 0 0 0 4 0 0 15 4 0 ...
##  $ PctForeignBorn       : num  10.66 8.3 5 2.04 1.49 ...
##  $ PctBornSameState     : num  53.7 77.2 44.8 88.7 64.3 ...
##  $ PctSameHouse85       : num  65.3 71.3 36.6 56.7 42.3 ...
##  $ PctSameCity85        : num  78.1 90.2 61.3 90.2 70.6 ...
##  $ PctSameState85       : num  89.1 96.1 82.8 96.2 85.7 ...
##  $ LemasSwornFT         : num  NA NA NA NA NA NA NA NA 198 NA ...
##   [list output truncated]
```

The structure of the data is partialy ommited due to the high number of features. Let's try getting the class of each feature

```r
apply(X = X, MARGIN = 2, FUN = class)
```

```
##            population         householdsize           racepctblack 
##             "numeric"             "numeric"              "numeric" 
##          racePctWhite           racePctAsian            racePctHisp 
##             "numeric"             "numeric"              "numeric" 
##           agePct12t21            agePct12t29            agePct16t24 
##             "numeric"             "numeric"              "numeric" 
##            agePct65up              numbUrban               pctUrban 
##             "numeric"             "numeric"              "numeric" 
##             medIncome               pctWWage            pctWFarmSelf 
##             "numeric"             "numeric"              "numeric" 
##            pctWInvInc             pctWSocSec             pctWPubAsst 
##             "numeric"             "numeric"              "numeric" 
##            pctWRetire              medFamInc              perCapInc 
##             "numeric"             "numeric"              "numeric" 
##           whitePerCap            blackPerCap            indianPerCap 
##             "numeric"             "numeric"              "numeric" 
##           AsianPerCap            OtherPerCap             HispPerCap 
##             "numeric"             "numeric"              "numeric" 
##           NumUnderPov          PctPopUnderPov        PctLess9thGrade 
##             "numeric"             "numeric"              "numeric" 
##          PctNotHSGrad           PctBSorMore          PctUnemployed 
##             "numeric"             "numeric"              "numeric" 
##             PctEmploy            PctEmplManu        PctEmplProfServ 
##             "numeric"             "numeric"              "numeric" 
##          PctOccupManu       PctOccupMgmtProf         MalePctDivorce 
##             "numeric"             "numeric"              "numeric" 
##         MalePctNevMarr          FemalePctDiv            TotalPctDiv 
##             "numeric"             "numeric"              "numeric" 
##            PersPerFam            PctFam2Par             PctKids2Par 
##             "numeric"             "numeric"              "numeric" 
##       PctYoungKids2Par            PctTeen2Par    PctWorkMomYoungKids 
##             "numeric"             "numeric"              "numeric" 
##            PctWorkMom     NumKidsBornNeverMar    PctKidsBornNeverMar 
##             "numeric"             "numeric"              "numeric" 
##              NumImmig         PctImmigRecent          PctImmigRec5 
##             "numeric"             "numeric"              "numeric" 
##          PctImmigRec8          PctImmigRec10         PctRecentImmig 
##             "numeric"             "numeric"              "numeric" 
##          PctRecImmig5           PctRecImmig8          PctRecImmig10 
##             "numeric"             "numeric"              "numeric" 
##       PctSpeakEnglOnly     PctNotSpeakEnglWell       PctLargHouseFam 
##             "numeric"             "numeric"              "numeric" 
##       PctLargHouseOccup      PersPerOccupHous      PersPerOwnOccHous 
##             "numeric"             "numeric"              "numeric" 
##     PersPerRentOccHous      PctPersOwnOccup      PctPersDenseHous 
##             "numeric"             "numeric"              "numeric" 
##         PctHousLess3BR               MedNumBR             HousVacant 
##             "numeric"             "numeric"              "numeric" 
##           PctHousOccup          PctHousOwnOcc       PctVacantBoarded 
##             "numeric"             "numeric"              "numeric"
```

```
##        PctVacMore6Mos        MedYrHousBuilt        PctHousNoPhone
##             "numeric"             "numeric"             "numeric"
##        PctWOFullPlumb        OwnOccLowQuart          OwnOccMedVal
##             "numeric"             "numeric"             "numeric"
##         OwnOccHiQuart          OwnOccQrange              RentLowQ
##             "numeric"             "numeric"             "numeric"
##            RentMedian             RentHighQ            RentQrange
##             "numeric"             "numeric"             "numeric"
##               MedRent     MedRentPctHousInc      MedOwnCostPctInc
##             "numeric"             "numeric"             "numeric"
## MedOwnCostPctIncNoMtg          NumInShelters             NumStreet
##             "numeric"             "numeric"             "numeric"
##        PctForeignBorn      PctBornSameState        PctSameHouse85
##             "numeric"             "numeric"             "numeric"
##         PctSameCity85        PctSameState85          LemasSwornFT
##             "numeric"             "numeric"             "numeric"
##       LemasSwFTPerPop     LemasSwFTFieldOps  LemasSwFTFieldPerPop
##             "numeric"             "numeric"             "numeric"
##         LemasTotalReq     LemasTotReqPerPop      PolicReqPerOffic
##             "numeric"             "numeric"             "numeric"
##           PolicPerPop    RacialMatchCommPol         PctPolicWhite
##             "numeric"             "numeric"             "numeric"
##         PctPolicBlack          PctPolicHisp         PctPolicAsian
##             "numeric"             "numeric"             "numeric"
##         PctPolicMinor   OfficAssgnDrugUnits     NumKindsDrugsSeiz
##             "numeric"             "numeric"             "numeric"
##      PolicAveOTWorked              LandArea               PopDens
##             "numeric"             "numeric"             "numeric"
##        PctUsePubTrans             PolicCars         PolicOperBudg
##             "numeric"             "numeric"             "numeric"
##   LemasPctPolicOnPatr    LemasGangUnitDeploy   LemasPctOfficDrugUn
##             "numeric"             "numeric"             "numeric"
##       PolicBudgPerPop
##             "numeric"
```

Neither str() nor apply(class) shows any factor. Just to be certain, I examine the documentation from the source (UC Irvine): https://archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized

There exist in the original data the feature of states, county code, and community code, which are catergorical. However, they are not included in the given data. On the other hads, all other quantitative features in the original data are. We can say that the data set is entirely quantitative.

Missing Data Processing

check if target y contains missing data

```r
any(is.na(y))
```

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

check if any of the features contains missing data

```r
any(is.na(X))
```

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

Now that we have detected there is NA in some the features, we decide to replace it by the median of other existing data in that corresponding feature

```r
X <- X %>% mutate_all(function(x) ifelse(is.na(x), median(x, na.rm = TRUE), x))
any(is.na(X))
```

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

Data Normalization - Scaling

After the previous step of examination, it is obvious that many features are different in nature. For example, some feautures are Percentage (PctForeignBorn, PctBornSameState). Some are counts (NumInShelters, population). Some are in US Dollars (MedRent, . . .). Each of the features have different range, scale, and unit. Such condition will affect how much each of the feature influence the predition later on .Therefore, it is highly crucial that we normalize the features.

```r
X <- scale(X)
```

Regression task

In this section, you should use the techniques learned in class to develop a model to predict ViolentCrimesPerPop using the 124 features (or some subset of them) stored in **X**. Remember that you should try several different methods, and use model selection methods to determine which model is best. You should also be sure to keep a held-out test set to evaluate the performance of your model.

YOUR CODE GOES HERE