Crime and Communities

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.

```
library(readr)
CC <- read_csv("crime_and_communities_data.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

```
print(dim(CC))
```

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

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

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?

```
type_vec <- vector(length = ncol(X))
for (i in 1:ncol(X)) {
  type_vec[i] <- typeof(X[[1]])
}
sum(type_vec == 'double')
```

```
## [1] 124
```

```
ncol(X)
```

```
## [1] 124
```

Looking over the data dictionary provided on the website that provides the data and checking the type of each column all of the variables are numeric as they are stored as a double, I would say that the variables are numerical as well. The values such as population, percentage urban, median number of bedrooms, and number of police cars are all numbers that pertain to each community in the dataset. Although none of the variables are categorical, a lot of them are based on categorical things such as per capita income there is an overall measure for the entire population, but there are also other variables that have the per capita income specifically for caucasians, african americans, and other race/ethnicity groups. This pattern follows for some other variables as well such as percent of police given race/ethnicity, percentage of population, and a couple more.

```
summary_of_vars <- summary(X)
summary_of_vars
```

```
##    population      householdsize    racepctblack    racePctWhite
##  Min.   :  10005   Min.   :1.600   Min.   : 0.00   Min.   : 2.68
##  1st Qu.:  14359   1st Qu.:2.490   1st Qu.: 0.94   1st Qu.:75.88
##  Median :  22681   Median :2.650   Median : 3.15   Median :89.61
##  Mean   :  52251   Mean   :2.707   Mean   : 9.51   Mean   :83.49
##  3rd Qu.:  43154   3rd Qu.:2.850   3rd Qu.:11.96   3rd Qu.:95.99
##  Max.   :7322564   Max.   :5.280   Max.   :96.67   Max.   :99.63
##
##   racePctAsian      racePctHisp      agePct12t21     agePct12t29
##  Min.   : 0.0300   Min.   : 0.120   Min.   : 4.58   Min.   : 9.38
##  1st Qu.: 0.6125   1st Qu.: 0.920   1st Qu.:12.23   1st Qu.:24.38
##  Median : 1.2400   Median : 2.340   Median :13.62   Median :26.77
##  Mean   : 2.7508   Mean   : 8.482   Mean   :14.43   Mean   :27.62
##  3rd Qu.: 2.7375   3rd Qu.: 8.610   3rd Qu.:15.39   3rd Qu.:29.18
##  Max.   :57.4600   Max.   :95.290   Max.   :54.40   Max.   :70.51
##
##   agePct16t24      agePct65up       numbUrban          pctUrban
##  Min.   : 4.64   Min.   : 1.660   Min.   :      0   Min.   :  0.00
##  1st Qu.:11.34   1st Qu.: 8.922   1st Qu.:      0   1st Qu.:  0.00
##  Median :12.54   Median :11.855   Median :  17348   Median :100.00
##  Mean   :13.99   Mean   :12.005   Mean   :  46672   Mean   : 69.62
##  3rd Qu.:14.36   3rd Qu.:14.547   3rd Qu.:  41932   3rd Qu.:100.00
##  Max.   :63.62   Max.   :52.770   Max.   :7322564   Max.   :100.00
##
##    medIncome         pctWWage      pctWFarmSelf      pctWInvInc
##  Min.   : 11576   Min.   :31.68   Min.   :0.0000   Min.   : 7.91
##  1st Qu.: 23597   1st Qu.:73.22   1st Qu.:0.4700   1st Qu.:34.19
##  Median : 30896   Median :78.38   Median :0.7000   Median :42.38
##  Mean   : 33699   Mean   :78.08   Mean   :0.8933   Mean   :43.36
##  3rd Qu.: 41215   3rd Qu.:83.70   3rd Qu.:1.1100   3rd Qu.:52.07
##  Max.   :123625   Max.   :96.62   Max.   :6.5300   Max.   :89.04
##
##    pctWSocSec     pctWPubAsst       pctWRetire      medFamInc
##  Min.   : 4.81   Min.   : 0.500   Min.   : 3.46   Min.   : 13785
##  1st Qu.:20.98   1st Qu.: 3.362   1st Qu.:12.99   1st Qu.: 29307
##  Median :26.79   Median : 5.720   Median :15.66   Median : 36010
##  Mean   :26.66   Mean   : 6.806   Mean   :16.06   Mean   : 39553
##  3rd Qu.:31.84   3rd Qu.: 9.150   3rd Qu.:18.78   3rd Qu.: 46683
##  Max.   :76.39   Max.   :26.920   Max.   :45.51   Max.   :131315
##
##    perCapInc      whitePerCap     blackPerCap      indianPerCap
##  Min.   : 5237   Min.   : 5472   Min.   :     0   Min.   :     0
```

```
##  1st Qu.:11548   1st Qu.:12596   1st Qu.:  6706   1st Qu.:  6336
##  Median :13977   Median :15028   Median :  9664   Median :  9834
##  Mean   :15522   Mean   :16535   Mean   : 11472   Mean   : 12257
##  3rd Qu.:17774   3rd Qu.:18610   3rd Qu.: 14464   3rd Qu.: 14690
##  Max.   :63302   Max.   :68850   Max.   :212120   Max.   :480000
##
##   AsianPerCap      OtherPerCap       HispPerCap     NumUnderPov
##  Min.   :     0   Min.   :     0   Min.   :    0   Min.   :     78.0
##  1st Qu.:  8441   1st Qu.:  5500   1st Qu.: 7253   1st Qu.:    936.2
##  Median : 12331   Median :  8144   Median : 9676   Median :   2217.5
##  Mean   : 14284   Mean   :  9375   Mean   :10989   Mean   :   7398.4
##  3rd Qu.: 17346   3rd Qu.: 11378   3rd Qu.:13360   3rd Qu.:   5097.5
##  Max.   :106165   Max.   :137000   Max.   :54648   Max.   :1384994.0
##                   NA's   :1
##  PctPopUnderPov   PctLess9thGrade   PctNotHSGrad    PctBSorMore
##  Min.   : 0.640   Min.   : 0.200   Min.   : 2.09   Min.   : 1.63
##  1st Qu.: 4.692   1st Qu.: 4.770   1st Qu.:14.20   1st Qu.:14.09
##  Median : 9.650   Median : 7.920   Median :21.66   Median :19.62
##  Mean   :11.796   Mean   : 9.444   Mean   :22.70   Mean   :22.99
##  3rd Qu.:17.078   3rd Qu.:12.245   3rd Qu.:29.66   3rd Qu.:28.93
##  Max.   :48.820   Max.   :49.890   Max.   :73.66   Max.   :73.63
##
##  PctUnemployed      PctEmploy      PctEmplManu    PctEmplProfServ
##  Min.   : 1.320   Min.   :24.82   Min.   : 2.05   Min.   : 8.69
##  1st Qu.: 4.090   1st Qu.:56.35   1st Qu.:11.94   1st Qu.:20.11
##  Median : 5.485   Median :62.27   Median :16.66   Median :23.41
##  Mean   : 6.024   Mean   :61.78   Mean   :17.79   Mean   :24.58
##  3rd Qu.: 7.430   3rd Qu.:67.50   3rd Qu.:22.75   3rd Qu.:27.63
##  Max.   :23.830   Max.   :84.67   Max.   :50.03   Max.   :62.67
##
##   PctOccupManu    PctOccupMgmtProf MalePctDivorce   MalePctNevMarr
##  Min.   : 1.370   Min.   : 6.48    Min.   : 2.130   Min.   :12.06
##  1st Qu.: 9.072   1st Qu.:21.92    1st Qu.: 7.162   1st Qu.:25.41
##  Median :13.040   Median :26.30    Median : 9.240   Median :29.00
##  Mean   :13.747   Mean   :28.25    Mean   : 9.180   Mean   :30.67
##  3rd Qu.:17.465   3rd Qu.:32.89    3rd Qu.:11.110   3rd Qu.:33.47
##  Max.   :44.270   Max.   :64.97    Max.   :19.090   Max.   :76.32
##
##   FemalePctDiv    TotalPctDiv      PersPerFam      PctFam2Par
##  Min.   : 3.35   Min.   : 2.83   Min.   :2.290   Min.   :32.24
##  1st Qu.: 9.94   1st Qu.: 8.64   1st Qu.:2.990   1st Qu.:67.67
##  Median :12.63   Median :11.04   Median :3.095   Median :74.77
##  Mean   :12.40   Mean   :10.88   Mean   :3.129   Mean   :73.90
##  3rd Qu.:14.80   3rd Qu.:13.06   3rd Qu.:3.220   3rd Qu.:81.64
##  Max.   :23.46   Max.   :19.11   Max.   :4.640   Max.   :93.60
##
##   PctKids2Par    PctYoungKids2Par  PctTeen2Par    PctWorkMomYoungKids
##  Min.   :26.11   Min.   : 27.43   Min.   :30.64   Min.   :24.42
##  1st Qu.:63.62   1st Qu.: 74.42   1st Qu.:69.92   1st Qu.:55.45
##  Median :72.06   Median : 83.77   Median :76.67   Median :60.70
##  Mean   :70.91   Mean   : 81.75   Mean   :75.34   Mean   :60.43
##  3rd Qu.:79.82   3rd Qu.: 91.44   3rd Qu.:82.52   3rd Qu.:65.80
##  Max.   :92.58   Max.   :100.00   Max.   :97.34   Max.   :87.97
##
```

```
##    PctWorkMom    NumKidsBornNeverMar PctKidsBornNeverMar    NumImmig
##  Min.   :41.95   Min.   :     0.0    Min.   : 0.000      Min.   :     20
##  1st Qu.:64.96   1st Qu.:   146.2    1st Qu.: 1.083      1st Qu.:    407
##  Median :69.25   Median :   361.0    Median : 2.080      Median :   1040
##  Mean   :68.80   Mean   :  2041.5    Mean   : 3.140      Mean   :   6314
##  3rd Qu.:73.34   3rd Qu.:  1070.2    3rd Qu.: 3.980      3rd Qu.:   3389
##  Max.   :89.37   Max.   :527557.0    Max.   :24.190      Max.   :2082931
##
##  PctImmigRecent    PctImmigRec5    PctImmigRec8   PctImmigRec10
##  Min.   : 0.000   Min.   : 0.00   Min.   : 0.00   Min.   : 0.00
##  1st Qu.: 6.942   1st Qu.:11.70   1st Qu.:17.91   1st Qu.:23.54
##  Median :12.440   Median :19.64   Median :27.46   Median :35.58
##  Mean   :13.734   Mean   :20.83   Mean   :28.12   Mean   :35.48
##  3rd Qu.:18.090   3rd Qu.:27.69   3rd Qu.:37.07   3rd Qu.:46.81
##  Max.   :64.290   Max.   :76.16   Max.   :80.81   Max.   :88.00
##
##  PctRecentImmig    PctRecImmig5     PctRecImmig8    PctRecImmig10
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 0.180   1st Qu.: 0.290   1st Qu.: 0.410   1st Qu.: 0.540
##  Median : 0.530   Median : 0.780   Median : 1.080   Median : 1.380
##  Mean   : 1.149   Mean   : 1.781   Mean   : 2.424   Mean   : 3.094
##  3rd Qu.: 1.370   3rd Qu.: 2.180   3rd Qu.: 2.870   3rd Qu.: 3.680
##  Max.   :13.710   Max.   :19.930   Max.   :25.340   Max.   :32.630
##
##  PctSpeakEnglOnly PctNotSpeakEnglWell PctLargHouseFam  PctLargHouseOccup
##  Min.   : 6.15    Min.   : 0.000      Min.   : 0.960   Min.   : 0.440
##  1st Qu.:83.70    1st Qu.: 0.510      1st Qu.: 3.390   1st Qu.: 2.360
##  Median :91.78    Median : 0.955      Median : 4.290   Median : 3.050
##  Mean   :86.55    Mean   : 2.538      Mean   : 5.465   Mean   : 3.975
##  3rd Qu.:95.41    3rd Qu.: 2.467      3rd Qu.: 5.957   3rd Qu.: 4.280
##  Max.   :98.98    Max.   :38.330      Max.   :34.870   Max.   :30.870
##
##  PersPerOccupHous PersPerOwnOccHous PersPerRentOccHous PctPersOwnOccup
##  Min.   :1.580    Min.   :1.610     Min.   :1.580      Min.   :13.93
##  1st Qu.:2.400    1st Qu.:2.540     1st Qu.:2.120      1st Qu.:56.56
##  Median :2.560    Median :2.700     Median :2.290      Median :64.99
##  Mean   :2.614    Mean   :2.734     Mean   :2.382      Mean   :65.50
##  3rd Qu.:2.770    3rd Qu.:2.890     3rd Qu.:2.540      3rd Qu.:75.30
##  Max.   :4.520    Max.   :4.480     Max.   :4.730      Max.   :96.59
##
##  PctPersDenseHous PctHousLess3BR     MedNumBR       HousVacant
##  Min.   : 0.050   Min.   : 3.06   Min.   :1.000   Min.   :    36.0
##  1st Qu.: 1.300   1st Qu.:37.93   1st Qu.:2.000   1st Qu.:   310.0
##  Median : 2.470   Median :46.78   Median :3.000   Median :   582.5
##  Mean   : 4.325   Mean   :45.84   Mean   :2.626   Mean   :  1733.0
##  3rd Qu.: 4.920   3rd Qu.:54.09   3rd Qu.:3.000   3rd Qu.:  1280.5
##  Max.   :59.490   Max.   :95.34   Max.   :4.000   Max.   :172768.0
##
##   PctHousOccup   PctHousOwnOcc   PctVacantBoarded PctVacMore6Mos
##  Min.   :37.47   Min.   :16.86   Min.   : 0.000   Min.   : 3.12
##  1st Qu.:90.98   1st Qu.:54.09   1st Qu.: 0.780   1st Qu.:24.74
##  Median :93.98   Median :62.08   Median : 1.740   Median :34.52
##  Mean   :92.71   Mean   :62.63   Mean   : 2.791   Mean   :35.15
##  3rd Qu.:95.91   3rd Qu.:71.59   3rd Qu.: 3.520   3rd Qu.:44.26
```

```
##  Max.   :99.00   Max.   :96.36   Max.   :39.890   Max.   :82.13
##
##  MedYrHousBuilt PctHousNoPhone   PctWOFullPlumb   OwnOccLowQuart
##  Min.   :1939   Min.   : 0.000   Min.   :0.0000   Min.   : 15700
##  1st Qu.:1956   1st Qu.: 0.980   1st Qu.:0.1800   1st Qu.: 41800
##  Median :1964   Median : 3.090   Median :0.3300   Median : 65900
##  Mean   :1963   Mean   : 4.446   Mean   :0.4377   Mean   : 91116
##  3rd Qu.:1971   3rd Qu.: 7.080   3rd Qu.:0.5700   3rd Qu.:126800
##  Max.   :1987   Max.   :23.630   Max.   :5.3300   Max.   :500001
##
##   OwnOccMedVal    OwnOccHiQuart     OwnOccQrange       RentLowQ
##  Min.   : 26600   Min.   : 36700   Min.   :     0   Min.   :  99.0
##  1st Qu.: 56700   1st Qu.: 74800   1st Qu.: 32925   1st Qu.: 210.0
##  Median : 84600   Median :109500   Median : 44250   Median : 305.0
##  Mean   :116102   Mean   :149007   Mean   : 57891   Mean   : 328.1
##  3rd Qu.:156250   3rd Qu.:192850   3rd Qu.: 67475   3rd Qu.: 420.0
##  Max.   :500001   Max.   :500001   Max.   :331000   Max.   :1001.0
##
##    RentMedian       RentHighQ        RentQrange       MedRent
##  Min.   : 120.0   Min.   : 182.0   Min.   :  0.0   Min.   : 192.0
##  1st Qu.: 286.0   1st Qu.: 361.2   1st Qu.:139.0   1st Qu.: 363.0
##  Median : 394.0   Median : 484.0   Median :173.0   Median : 467.0
##  Mean   : 428.4   Mean   : 528.4   Mean   :200.3   Mean   : 502.7
##  3rd Qu.: 547.8   3rd Qu.: 667.8   3rd Qu.:241.0   3rd Qu.: 621.0
##  Max.   :1001.0   Max.   :1001.0   Max.   :803.0   Max.   :1001.0
##
##  MedRentPctHousInc MedOwnCostPctInc MedOwnCostPctIncNoMtg NumInShelters
##  Min.   :14.90     Min.   :14.10    Min.   :10.10         Min.   :    0.00
##  1st Qu.:24.30     1st Qu.:19.10    1st Qu.:11.90         1st Qu.:    0.00
##  Median :26.20     Median :21.20    Median :12.80         Median :    0.00
##  Mean   :26.33     Mean   :21.21    Mean   :13.03         Mean   :   67.72
##  3rd Qu.:28.10     3rd Qu.:23.30    3rd Qu.:13.80         3rd Qu.:   24.00
##  Max.   :35.10     Max.   :32.70    Max.   :23.40         Max.   :23383.00
##
##    NumStreet        PctForeignBorn   PctBornSameState PctSameHouse85
##  Min.   :    0.00   Min.   : 0.180   Min.   : 6.75    Min.   :11.83
##  1st Qu.:    0.00   1st Qu.: 2.080   1st Qu.:48.87    1st Qu.:44.68
##  Median :    0.00   Median : 4.490   Median :62.52    Median :51.87
##  Mean   :   18.71   Mean   : 7.606   Mean   :60.50    Mean   :51.32
##  3rd Qu.:    1.00   3rd Qu.: 9.585   3rd Qu.:74.38    3rd Qu.:58.51
##  Max.   :10447.00   Max.   :60.400   Max.   :93.14    Max.   :78.56
##
##  PctSameCity85   PctSameState85   LemasSwornFT     LemasSwFTPerPop
##  Min.   :27.95   Min.   :32.83   Min.   :   65.0   Min.   :  29.4
##  1st Qu.:71.92   1st Qu.:84.73   1st Qu.:  131.0   1st Qu.: 149.1
##  Median :79.31   Median :89.64   Median :  173.0   Median : 196.0
##  Mean   :77.11   Mean   :87.73   Mean   :  458.7   Mean   : 248.1
##  3rd Qu.:84.70   3rd Qu.:92.73   3rd Qu.:  314.0   3rd Qu.: 260.8
##  Max.   :96.59   Max.   :99.90   Max.   :25655.0   Max.   :3437.2
##                                  NA's   :1675      NA's   :1675
##  LemasSwFTFieldOps LemasSwFTFieldPerPop LemasTotalReq     LemasTotReqPerPop
##  Min.   :   14.0   Min.   :  19.21      Min.   :   8100   Min.   :   2705
##  1st Qu.:  113.5   1st Qu.: 130.43      1st Qu.:  49864   1st Qu.:  65486
##  Median :  152.0   Median : 170.16      Median :  89205   Median :  91035
```

```
##  Mean   :  395.9   Mean   : 211.32      Mean   : 240510   Mean   : 122280
##  3rd Qu.:  283.0   3rd Qu.: 226.81      3rd Qu.: 174171   3rd Qu.: 131894
##  Max.   :22496.0   Max.   :3290.62      Max.   :8328470   Max.   :1926282
##  NA's   :1675      NA's   :1675         NA's   :1675      NA's   :1675
##  PolicReqPerOffic  PolicPerPop     RacialMatchCommPol PctPolicWhite
##  Min.   :  41.4   Min.   :  29.4   Min.   : 42.15     Min.   :  1.60
##  1st Qu.: 342.9   1st Qu.: 149.2   1st Qu.: 79.44     1st Qu.: 76.36
##  Median : 444.8   Median : 196.0   Median : 87.95     Median : 86.18
##  Mean   : 526.8   Mean   : 248.1   Mean   : 85.49     Mean   : 82.53
##  3rd Qu.: 646.0   3rd Qu.: 260.8   3rd Qu.: 93.62     3rd Qu.: 93.09
##  Max.   :2162.5   Max.   :3437.2   Max.   :100.00     Max.   :100.00
##  NA's   :1675     NA's   :1675     NA's   :1675       NA's   :1675
##  PctPolicBlack     PctPolicHisp    PctPolicAsian     PctPolicMinor
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.0000   Min.   : 0.00
##  1st Qu.: 2.055   1st Qu.: 0.450   1st Qu.: 0.0000   1st Qu.: 5.05
##  Median : 4.840   Median : 2.110   Median : 0.0000   Median :11.39
##  Mean   : 8.983   Mean   : 5.683   Mean   : 0.7088   Mean   :15.20
##  3rd Qu.:13.355   3rd Qu.: 6.490   3rd Qu.: 0.6650   3rd Qu.:19.68
##  Max.   :67.310   Max.   :98.400   Max.   :18.5700   Max.   :98.40
##  NA's   :1675     NA's   :1675     NA's   :1675      NA's   :1675
##  OfficAssgnDrugUnits NumKindsDrugsSeiz PolicAveOTWorked    LandArea
##  Min.   :   0.00     Min.   : 1.000    Min.   :  0.0    Min.   :   0.90
##  1st Qu.:   6.00     1st Qu.: 7.000    1st Qu.: 55.1    1st Qu.:   7.40
##  Median :  12.00     Median : 9.000    Median : 99.0    Median :  13.70
##  Mean   :  25.87     Mean   : 8.784    Mean   :119.8    Mean   :  27.96
##  3rd Qu.:  23.00     3rd Qu.:10.500    3rd Qu.:153.6    3rd Qu.:  25.77
##  Max.   :1773.00     Max.   :15.000    Max.   :634.7    Max.   :3569.80
##  NA's   :1675        NA's   :1675      NA's   :1675
##     PopDens       PctUsePubTrans     PolicCars      PolicOperBudg
##  Min.   :   10   Min.   : 0.000   Min.   :  20.0   Min.   :2.380e+06
##  1st Qu.: 1171   1st Qu.: 0.350   1st Qu.:  54.0   1st Qu.:7.247e+06
##  Median : 1996   Median : 1.220   Median :  86.0   Median :1.075e+07
##  Mean   : 2790   Mean   : 3.063   Mean   : 177.3   Mean   :2.896e+07
##  3rd Qu.: 3270   3rd Qu.: 3.377   3rd Qu.: 191.0   3rd Qu.:2.047e+07
##  Max.   :44230   Max.   :54.330   Max.   :3187.0   Max.   :1.617e+09
##                                   NA's   :1675     NA's   :1675
##  LemasPctPolicOnPatr LemasGangUnitDeploy LemasPctOfficDrugUn PolicBudgPerPop
##  Min.   :10.85       Min.   : 0.000      Min.   : 0.00       Min.   :  15260
##  1st Qu.:83.87       1st Qu.: 0.000      1st Qu.: 0.00       1st Qu.:  86869
##  Median :89.44       Median : 5.000      Median : 0.00       Median : 114582
##  Mean   :86.77       Mean   : 4.404      Mean   : 1.01       Mean   : 154590
##  3rd Qu.:93.06       3rd Qu.:10.000      3rd Qu.: 0.00       3rd Qu.: 156961
##  Max.   :99.94       Max.   :10.000      Max.   :48.44       Max.   :2422367
##  NA's   :1675        NA's   :1675                            NA's   :1675
```

The website with the data also provides the summary statistics of all of the different variables which include the minimum, maximum, mean, standard deviation, correlation with ViolPerPop, median, mode, and missing. This information can be difficult to look at as just the common seperated lists that they are presented with on the website. I think a much better visualization for these values would be histograms for things such as population, household size, rent median, and other variables that are not split by categories like some of the previous variables mentioned such as per capita income for caucasians. For those variables I think side by side boxplots split by race/ethnicity might be a better visualization so that you would be able to compare the different metrics across the different race/ethnicity groups better.

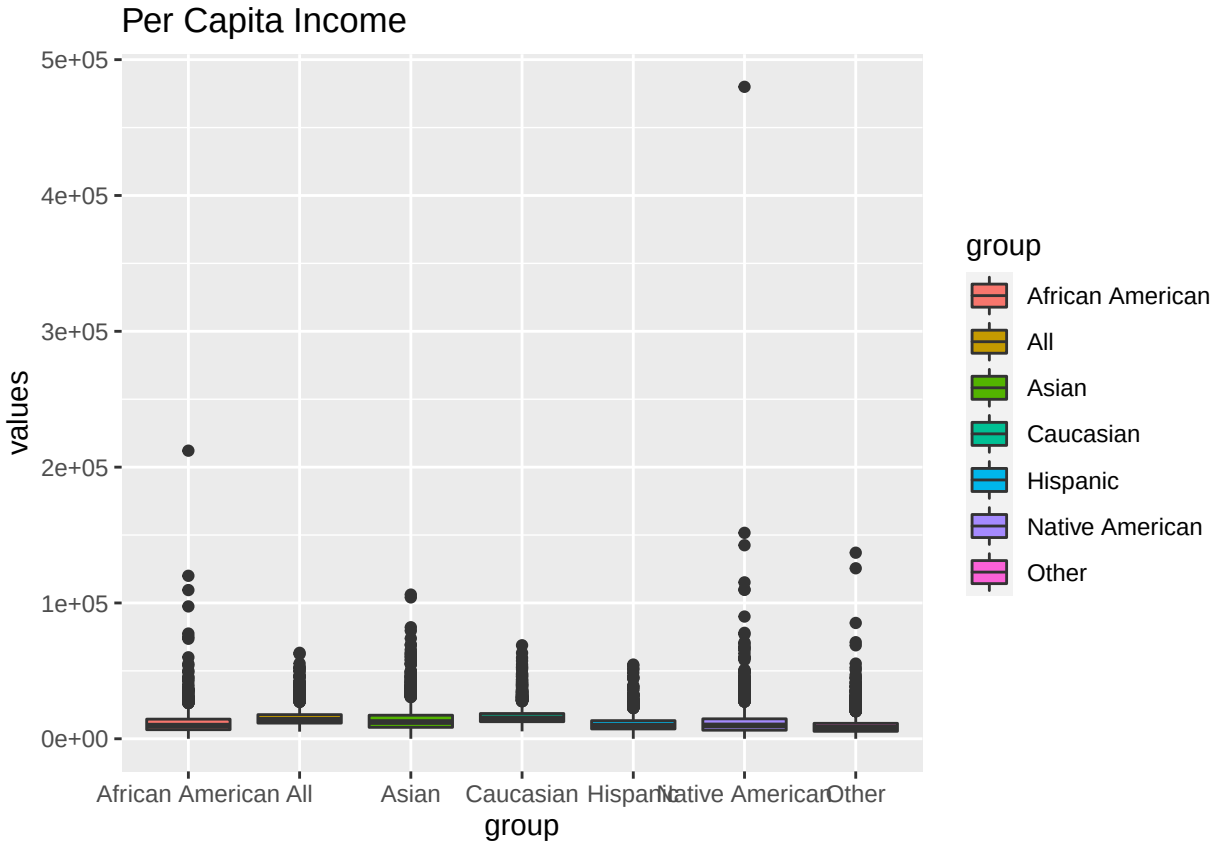
```
# visualizing summary statistics for per capitia income for all community residents, and then split by
library(ggplot2)
per_cap <- X[, c('perCapInc', 'whitePerCap', 'blackPerCap', 'indianPerCap', 'AsianPerCap', 'OtherPerCap
```

```
values <- c()
group <- c()
better_labels <- c('All', 'Caucasian', 'African American', 'Native American', 'Asian', 'Other', 'Hispani
i <- 1
for (c in names(per_cap)) {
  val <- per_cap[[c]]
  values <- c(values, val)
  group <- c(group, rep(better_labels[i], length(val)))
  i <- i + 1
}
```

```
per_cap_df <- data.frame(cbind(values, group), stringsAsFactors = FALSE)
per_cap_df[['values']] <- as.numeric(per_cap_df[['values']])
```

```
ggplot(data = per_cap_df, aes(x = group, y = values, fill = group)) + geom_boxplot() + ggtitle('Per Capi
```

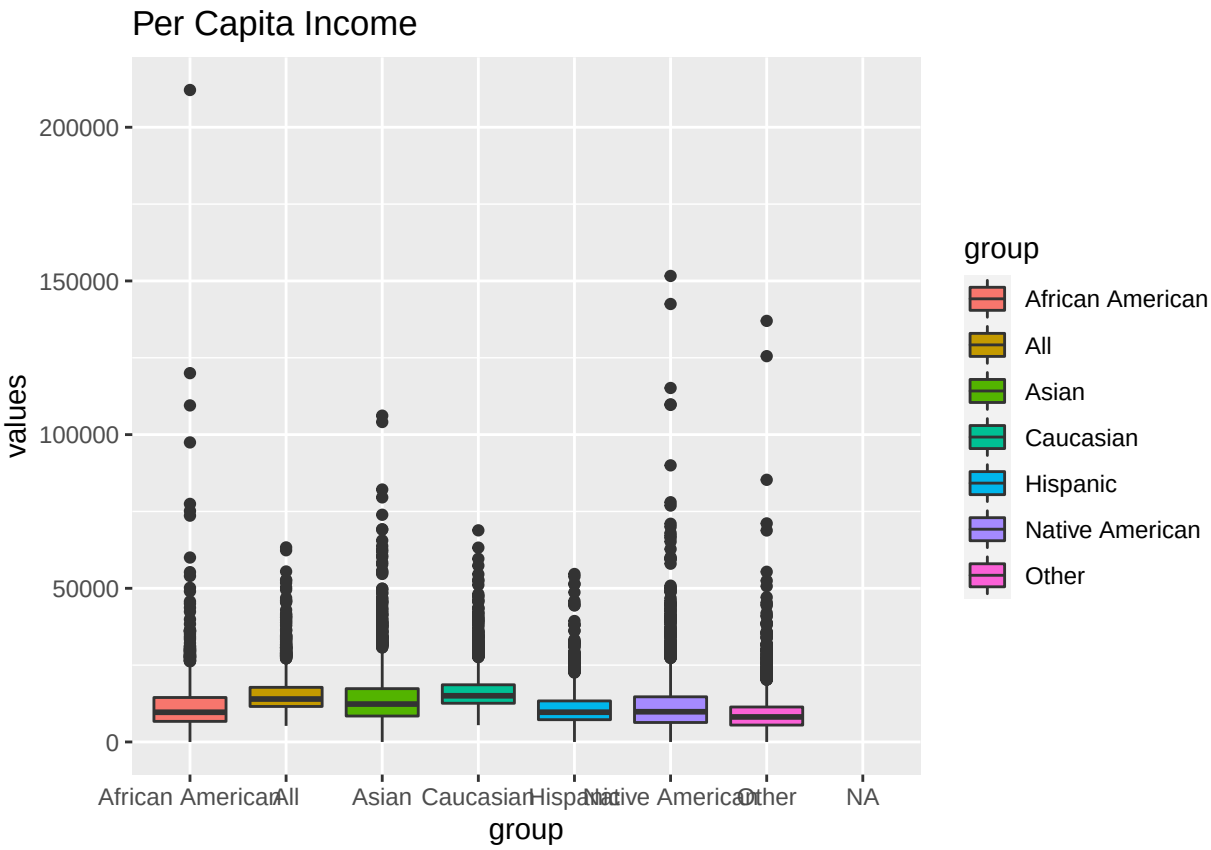
```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```




```
per_cap_df_wo <- per_cap_df[per_cap_df[['values']] != max(per_cap_df$values, na.rm = TRUE), ]
```

```
# remove large outlier in Native American Group in order to visualize things better
ggplot(per_cap_df_wo, aes(x = group, y = values, fill = group)) + geom_boxplot() + ggtitle('Per Capita I
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).
```



In terms of missing data, I think that simply filling the values with a predetermined value or removing the observations with missing values could skew the data or be getting rid of possibly meaningful data respectively. I think that maybe the best way to deal with a missing value in this situation would be to fill it with the the average of the column for which it belongs to so that the other variables for the community may be used without to much skewness hopefully. I chose this rather than completely ridding the dataset of rows for which there is some amount of missing data because there might be something in common with all of these communites with some amount of missing data and I would not want to completely disregard this and be getting rid of a specific subgroup of communities.

```
X_wo <- cbind()
for (c in names(X)) {
  col <- X[[c]]
  m <- mean(col, na.rm = TRUE)
  repl <- ifelse(is.na(col), m, col)
  X_wo <- cbind(X_wo, repl)
}
```

```
X_wo <- data.frame(X_wo)
names(X_wo) <- names(X)
X_wo_mat <- as.matrix(X_wo)
```

There are quite a few variables to look at specifically 124 and there are probably ones that matter more than others or ones that are correlated enough that having both of them seem redundant. In order to possibly recognize the more important variables for dimension reduction for overall more stable solutions for any type of prediction we are going to do. I think the use of LASSO for dimension reduction would be good. This would allow us to only use the variables that demonstrate the most variation in the data with less variance given so many features. Principal Component Analysis that also provides a form of dimension reduction, but we used that in the other part of the project and in this situation intuitively I think that certain features may

be more important in predicting ViolentCrimesPerPop and if we do not use PCA, it allows for our results to be more interpretable as the variables stay as themselves rather than linear combinations of all.

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.0
```

```
lambda_seq <- 10^seq(2, -2, by = -.1)
cv_lasso <- cv.glmnet(X_wo_mat, y, alpha = 1, lambda = lambda_seq)
best_lam <- cv_lasso$lambda.min
```

```
best_lasso <- glmnet(X_wo_mat, y, alpha = 1, lambda = best_lam)
```

```
coef_df <- data.frame(as.matrix(coef(best_lasso)))
coef_df[['coef']] <- rownames(coef_df)
rownames(coef_df) <- NULL
coef_df <- coef_df[coef_df[['s0']] != 0, ]
coef_df
```

```
##                s0                  coef
## 1    3.143717e+03           (Intercept)
## 4    5.672533e+00          racepctblack
## 5   -1.097555e+00          racePctWhite
## 9   -5.258953e+00           agePct12t29
## 13   8.762453e-01              pctUrban
## 15  -4.131689e+00              pctWWage
## 16   3.261988e-02          pctWFarmSelf
## 17  -3.479053e+00            pctWInvInc
## 18   7.574838e-01            pctWSocSec
## 20  -8.055636e+00            pctWRetire
## 24  -8.277611e-04           blackPerCap
## 26   1.718396e-03           AsianPerCap
## 27   2.495441e-03           OtherPerCap
## 30  -2.401485e-01        PctPopUnderPov
## 31  -5.151019e+00       PctLess9thGrade
## 32   6.956310e-01          PctNotHSGrad
## 33  -6.515949e-01           PctBSorMore
## 34  -1.279000e+00         PctUnemployed
## 36  -2.149525e+00           PctEmplManu
## 37  -1.293818e-01       PctEmplProfServ
## 40   1.538142e+01        MalePctDivorce
## 46  -1.097954e+01           PctKids2Par
## 50  -4.077470e+00            PctWorkMom
## 52   4.865603e+01   PctKidsBornNeverMar
## 70   8.395292e+00      PctPersDenseHous
## 71   6.350635e-01        PctHousLess3BR
## 74  -5.753450e+00          PctHousOccup
## 76   1.082684e+01      PctVacantBoarded
## 77  -9.843527e-01        PctVacMore6Mos
## 84  -1.725655e-04          OwnOccQrange
## 85  -3.322853e-02              RentLowQ
## 88   4.078740e-01            RentQrange
## 90   6.235518e-01     MedRentPctHousInc
## 92  -2.527283e+01 MedOwnCostPctIncNoMtg
```

```
## 93   1.454301e-01         NumInShelters
## 95   3.353918e+00        PctForeignBorn
## 98   5.218482e-01         PctSameCity85
## 102 -4.902644e-02     LemasSwFTFieldOps
## 103 -7.094853e-02  LemasSwFTFieldPerPop
## 104 -8.478423e-06         LemasTotalReq
## 106  1.258514e-01      PolicReqPerOffic
## 108 -5.415041e+00    RacialMatchCommPol
## 109 -1.536731e-01         PctPolicWhite
## 111  1.571526e-02          PctPolicHisp
## 112  7.530933e+00         PctPolicAsian
## 113  1.791260e-02         PctPolicMinor
## 114 -1.902049e+00   OfficAssgnDrugUnits
## 115 -2.487402e-01     NumKindsDrugsSeiz
## 116 -1.931828e-01      PolicAveOTWorked
## 118 -1.991977e-03               PopDens
## 120  5.496394e-01             PolicCars
## 123  8.562241e+00   LemasGangUnitDeploy
## 124  9.904745e+00   LemasPctOfficDrugUn
```

LASSO as a dimension reduction technique identified 39 predictors that would best help predict ViolentCrimesPerPop.

```
keep <- coef_df[['coef']][2:40]
keep
```

```
##  [1] "racepctblack"          "racePctWhite"          "agePct12t29"
##  [4] "pctUrban"              "pctWWage"              "pctWFarmSelf"
##  [7] "pctWInvInc"            "pctWSocSec"            "pctWRetire"
## [10] "blackPerCap"           "AsianPerCap"           "OtherPerCap"
## [13] "PctPopUnderPov"        "PctLess9thGrade"       "PctNotHSGrad"
## [16] "PctBSorMore"           "PctUnemployed"         "PctEmplManu"
## [19] "PctEmplProfServ"       "MalePctDivorce"        "PctKids2Par"
## [22] "PctWorkMom"            "PctKidsBornNeverMar"   "PctPersDenseHous"
## [25] "PctHousLess3BR"        "PctHousOccup"          "PctVacantBoarded"
## [28] "PctVacMore6Mos"        "OwnOccQrange"          "RentLowQ"
## [31] "RentQrange"            "MedRentPctHousInc"     "MedOwnCostPctIncNoMtg"
## [34] "NumInShelters"         "PctForeignBorn"        "PctSameCity85"
## [37] "LemasSwFTFieldOps"     "LemasSwFTFieldPerPop"  "LemasTotalReq"
```

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 $\mathbf{X}$. 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.

Linear Regression

Linear regression is our most inflexible model as it assumes a linear relationship between the predictors and the target variables. However it is one of the least computationally inexpensive

```
nrow(X_wo)
```

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

```
lasso_data <- X_wo[, keep]
lasso_data[['ViolentCrimesPerPop']] <- y
shuf_lasso_data <- lasso_data[sample(1:nrow(lasso_data), size = nrow(lasso_data)), ]
train_lasso_data <- shuf_lasso_data[1:1500, ]
test_lasso_data <- shuf_lasso_data[1501:nrow(lasso_data), ]
```

```
linreg_cv <- c()
for (i in 0:4) {
  ind <- ((300 * i) + 1):(300 * (i + 1))
  train <- train_lasso_data[-ind, ]
  test <- train_lasso_data[ind, ]
  true_val <- test[['ViolentCrimesPerPop']]
  fit <- lm(ViolentCrimesPerPop ~ ., data = train)
  pred <- predict(fit, test)
  linreg_cv <- c(linreg_cv, mean((pred - true_val) ^ 2))
}
linreg_cv
```

```
## [1] 146962.7 135231.7 138003.5 110546.6 149775.7
```

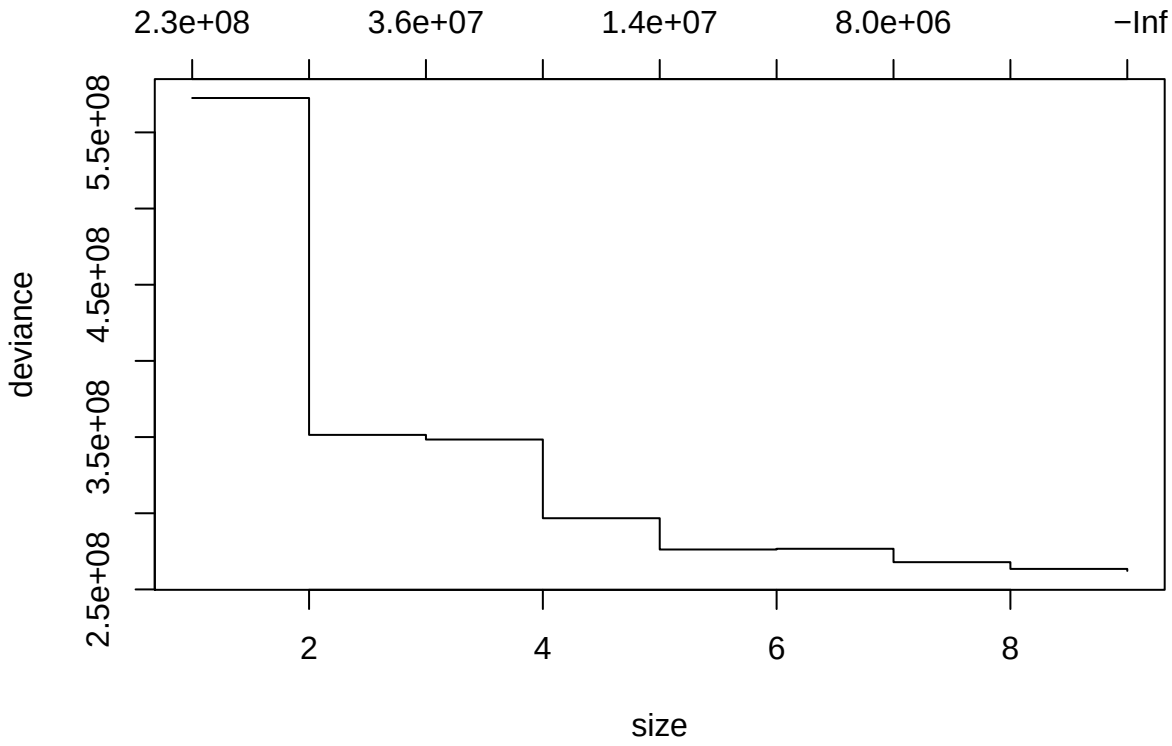
```
mean(linreg_cv)
```

```
## [1] 136104
```

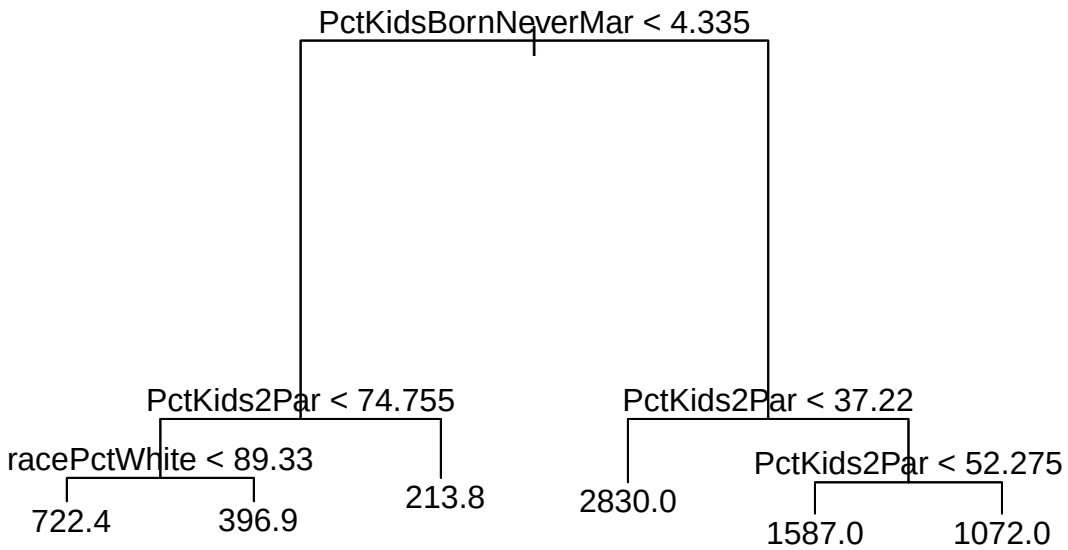
Decision Tree

Decision trees are really helpful because they provide a really nice visualization and they are easy to interpret. Some things that make it difficult on trainig a tree is deciding at what height we should prune the tree because we don't want to overfit the data and have the height to be so hight that the tree is no longer able to generalize for data that is not included in the training set.

```
library(tree)
tree_vcpp <- tree(ViolentCrimesPerPop ~., data = train_lasso_data)
cv_tree_vcpp <- cv.tree(tree_vcpp, FUN = prune.tree)
plot(cv_tree_vcpp)
```




```
pruned_tree_vcpp <- prune.tree(tree_vcpp, best = 6)
plot(pruned_tree_vcpp)
text(pruned_tree_vcpp, pretty = 0)
```




```
dec_tree_cv <- c()
for (i in 0:4) {
  ind <- ((300 * i) + 1):(300 * (i + 1))
  train <- train_lasso_data[-ind, ]
  test <- train_lasso_data[ind, ]
  true_val <- test[['ViolentCrimesPerPop']]
  pruned_tree <- prune.tree(tree(ViolentCrimesPerPop ~., data = train), best = 6)
  pred <- predict(pruned_tree, test)
  dec_tree_cv <- c(dec_tree_cv, mean((pred - true_val) ^ 2))
```

```
}
dec_tree_cv
```

```
## [1] 191555.9 180984.5 181293.5 181182.2 180065.2
```

```
mean(dec_tree_cv)
```

```
## [1] 183016.3
```

Random Forest

We could have improved some sort of accuracy for the prediction of the decision trees by bootstrapping the training data, fitting a tree, and then averaging the predictions as our final prediction. This process is called bagging, but all of these trees are correlated to each other. Random forests impove this by decorrelating the trees since at each node split a random sample of a given number of predictors is chosen for the elements to be split on.

```
round(sqrt(39))
```

```
## [1] 6
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
rf_cv <- c()
m <- round(sqrt(39))
for (i in 0:4) {
  ind <- ((300 * i) + 1):(300 * (i + 1))
  train <- train_lasso_data[-ind, ]
  test <- train_lasso_data[ind, ]
  true_val <- test[['ViolentCrimesPerPop']]
  fit <- randomForest(ViolentCrimesPerPop ~ ., data = train, mtry = m)
  pred <- predict(fit, test)
  rf_cv <- c(rf_cv, mean((pred - true_val) ^ 2))
}
rf_cv
```

```
## [1] 136324.5 122859.1 141216.5 114107.4 139337.7
```

```
mean(rf_cv)
```

```
## [1] 130769
```

After some evaluation, the best model that we found in this case for prediciting the violent crimes per 100k in the population was a random forest model. We used the common practice of allowing the square root of the total predictors to be the number of variables that we allow to be randomly selected at each node.

```
# fitting the final model and finding the average mean squared error
rf_final <- randomForest(ViolentCrimesPerPop ~ ., data = train_lasso_data, mtry = m)
```

```r
predictions <- predict(rf_final, test_lasso_data)
mean((predictions - test_lasso_data$ViolentCrimesPerPop) ^ 2)
```

```
## [1] 136373.7
```