Crime and Communities

Justin Han

```
CC <- read_csv("../data/crime_and_communities_data.csv")
```

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

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

```
CC <- as.data.frame(CC)
```

Dataset exploration

After the dataset has been imported, we want to preview it to get an idea of what we will be working with.

```
head(CC, 10)
```

```
##    population householdsize racepctblack racePctWhite racePctAsian racePctHisp
## 1       11980          3.10         1.37        91.78         6.50        1.88
## 2       23123          2.82         0.80        95.57         3.44        0.85
## 3       29344          2.43         0.74        94.33         3.43        2.35
## 4       16656          2.40         1.70        97.35         0.50        0.70
## 5      140494          2.45         2.51        95.65         0.90        0.95
## 6       28700          2.60         1.60        96.57         1.47        1.10
## 7       59459          2.45        14.20        84.87         0.40        0.63
## 8       74111          2.46         0.35        97.11         1.25        0.73
## 9      103590          2.62        23.14        67.60         0.92       16.35
## 10      31601          2.54        12.63        83.22         0.77        4.39
##    agePct12t21 agePct12t29 agePct16t24 agePct65up numbUrban pctUrban medIncome
## 1        12.47       21.44       10.93      11.33     11980      100     75122
## 2        11.01       21.30       10.48      17.18     23123      100     47917
## 3        11.36       25.88       11.01      10.28     29344      100     35669
## 4        12.55       25.20       12.19      17.57         0        0     20580
## 5        18.09       32.89       20.04      13.26    140494      100     21577
## 6        11.17       27.41       12.76      14.42     28700      100     42805
## 7        15.31       27.93       14.78      14.60     59449      100     23221
## 8        16.64       35.16       20.33       8.58     74115      100     25326
## 9        19.88       34.55       21.62      13.12    103590      100     17852
## 10       15.73       28.57       15.16      14.26     31596      100     24763
##    pctWWage pctWFarmSelf pctWInvInc pctWSocSec pctWPubAsst pctWRetire medFamInc
## 1     89.24         1.55      70.20      23.62        1.03      18.39     79584
## 2     78.99         1.11      64.11      35.50        2.75      22.85     55323
## 3     82.00         1.15      55.73      22.25        2.94      14.56     42112
## 4     68.15         0.24      38.95      39.48       11.71      18.33     26501
## 5     75.78         1.00      41.15      29.31        7.12      14.09     27705
```

```
## 6     79.47         0.39      47.70      30.23        5.41      17.23     50394
## 7     71.60         0.67      35.74      32.58        8.81      22.59     28901
## 8     83.69         2.93      47.11      19.30        4.21      10.31     34269
## 9     74.20         0.86      30.98      29.09        9.06      13.99     24058
## 10    73.92         1.54      37.36      32.68        7.02      15.20     29509
##    perCapInc whitePerCap blackPerCap indianPerCap AsianPerCap OtherPerCap
## 1      29711       30233       13600         5725       27101        5115
## 2      20148       20191       18137            0       20074        5250
## 3      16946       17103       16644        21606       15528        5954
## 4      10810       10909        9984         4941        3541        2451
## 5      11878       12029        7382        10264       10753        7192
## 6      18193       18276       17342        21482       12639       21852
## 7      12161       12599        9820         6634        8802        7428
## 8      13554       13727        8852         5344        8011        5332
## 9      10195       12126        5715        11313        5770        7320
## 10     12929       14051        7496         9126        9107        5267
##    HispPerCap NumUnderPov PctPopUnderPov PctLess9thGrade PctNotHSGrad
## 1       22838         227           1.96            5.81         9.90
## 2       12222         885           3.98            5.61        13.72
## 3        8405        1389           4.75            2.80         9.09
## 4        4391        2831          17.23           11.05        33.68
## 5        8104       23223          17.78            8.76        23.03
## 6       22594        1126           4.01            4.49        13.89
## 7        6187       10320          17.98           10.09        28.67
## 8        5174        9603          13.68            5.52        11.27
## 9        6984       27767          28.68           13.01        31.62
## 10       4542        4698          15.61            9.07        24.86
##    PctBSorMore PctUnemployed PctEmploy PctEmplManu PctEmplProfServ PctOccupManu
## 1        48.18          2.70     64.55       14.65           28.82         5.49
## 2        29.89          2.43     61.96       12.26           29.28         6.39
## 3        30.13          4.01     69.80       15.95           21.52         8.79
## 4        10.81          9.86     54.74       31.22           27.43        26.76
## 5        20.66          5.72     59.02       14.31           26.83        14.72
## 6        27.01          4.85     65.42       14.02           27.17         8.50
## 7        12.00          8.19     56.59       27.00           21.54        21.92
## 8        30.24          4.18     68.51        6.89           31.55        11.37
## 9        17.02          8.39     51.37       15.73           29.06        16.43
## 10       19.23          7.19     57.76       25.33           27.59        15.74
##    PctOccupMgmtProf MalePctDivorce MalePctNevMarr FemalePctDiv TotalPctDiv
## 1             50.73           3.67          26.38         5.22        4.47
## 2             37.64           4.23          27.99         6.45        5.42
## 3             32.48          10.10          25.78        14.76       12.55
## 4             22.71          10.98          28.15        14.47       12.91
## 5             23.42          11.40          33.32        14.46       13.04
## 6             32.78           5.97          36.05         9.06        7.64
## 7             18.02          13.28          28.34        16.33       14.94
## 8             29.43           7.29          40.87         9.94        8.64
## 9             24.30          11.07          38.49        14.66       12.97
## 10            27.28          11.48          27.60        15.26       13.53
##    PersPerFam PctFam2Par PctKids2Par PctYoungKids2Par PctTeen2Par
## 1        3.22      91.43       90.17            95.78       95.81
## 2        3.11      86.91       85.33            96.82       86.46
## 3        2.95      78.54       78.85            92.37       75.72
## 4        2.98      64.02       62.36            65.38       67.43
```

```
## 5        2.89      71.94       69.79            79.76       75.33
## 6        3.14      79.53       79.76            92.05       77.12
## 7        2.95      62.56       58.70            69.89       62.76
## 8        3.00      79.35       79.70            86.60       80.70
## 9        3.11      61.65       54.56            68.85       61.69
## 10       2.99      68.41       64.64            75.18       70.94
##    PctWorkMomYoungKids PctWorkMom NumKidsBornNeverMar PctKidsBornNeverMar
## 1                44.56      58.88                  31                0.36
## 2                51.14      62.43                  43                0.24
## 3                66.08      74.19                 164                0.88
## 4                59.59      70.27                 561                3.84
## 5                62.96      70.52                1511                1.58
## 6                65.16      72.81                 263                1.18
## 7                63.08      72.44                2368                4.66
## 8                74.32      78.51                 751                1.64
## 9                60.80      69.23                3537                4.71
## 10               67.43      72.96                 603                2.47
##    NumImmig PctImmigRecent PctImmigRec5 PctImmigRec8 PctImmigRec10
## 1      1277           8.69        13.00        20.99         30.93
## 2      1920           5.21         8.65        13.33         22.50
## 3      1468          16.42        23.98        32.08         35.63
## 4       339          13.86        13.86        15.34         15.34
## 5      2091          21.33        30.56        38.02         45.48
## 6      2637          11.38        16.27        23.93         27.76
## 7       517          13.15        22.82        28.24         33.08
## 8      1474          23.68        33.58        46.68         53.93
## 9      4793          15.54        23.08        35.32         49.82
## 10      938          17.91        35.39        56.08         65.46
##    PctRecentImmig PctRecImmig5 PctRecImmig8 PctRecImmig10 PctSpeakEnglOnly
## 1            0.93         1.39         2.24          3.30            85.68
## 2            0.43         0.72         1.11          1.87            87.79
## 3            0.82         1.20         1.61          1.78            93.11
## 4            0.28         0.28         0.31          0.31            94.98
## 5            0.32         0.45         0.57          0.68            96.87
## 6            1.05         1.49         2.20          2.55            89.98
## 7            0.11         0.20         0.25          0.29            97.43
## 8            0.47         0.67         0.93          1.07            95.21
## 9            0.72         1.07         1.63          2.31            85.72
## 10           0.53         1.05         1.66          1.94            94.85
##    PctNotSpeakEnglWell PctLargHouseFam PctLargHouseOccup PersPerOccupHous
## 1                 1.37           4.81             4.17             2.99
## 2                 1.81           4.25             3.34             2.70
## 3                 1.14           2.97             2.05             2.42
## 4                 0.56           3.93             2.56             2.37
## 5                 0.60           3.08             1.92             2.28
## 6                 0.60           5.08             3.46             2.55
## 7                 0.28           3.85             2.55             2.36
## 8                 0.43           2.59             1.54             2.32
## 9                 2.51           6.70             4.10             2.45
## 10                0.81           3.66             2.51             2.42
##    PersPerOwnOccHous PersPerRentOccHous PctPersOwnOccup PctPersDenseHous
## 1               3.00              2.84           91.46             0.39
## 2               2.83              1.96           89.03             1.01
## 3               2.69              2.06           64.18             2.03
```

```
## 4              2.51              2.20           58.18             1.21
## 5              2.37              2.16           57.81             2.11
## 6              2.89              2.09           64.62             1.47
## 7              2.42              2.27           65.29             1.90
## 8              2.77              1.91           57.42             1.67
## 9              2.47              2.44           46.82             6.14
## 10             2.50              2.31           59.76             3.41
##    PctHousLess3BR MedNumBR HousVacant PctHousOccup PctHousOwnOcc
## 1           11.06        3         64        98.37         91.01
## 2           23.60        3        240        97.15         84.88
## 3           47.46        3        544        95.68         57.79
## 4           45.66        3        669        91.19         54.89
## 5           53.19        2       5119        91.81         55.50
## 6           47.35        3        566        95.11         56.96
## 7           56.30        2       2051        92.22         63.82
## 8           59.32        2       1562        95.07         48.10
## 9           59.96        2       5606        87.57         46.51
## 10          52.11        2       1807        87.33         57.83
##    PctVacantBoarded PctVacMore6Mos MedYrHousBuilt PctHousNoPhone PctWOFullPlumb
## 1              3.12          37.50           1959           0.00           0.28
## 2              0.00          18.33           1958           0.31           0.14
## 3              0.92           7.54           1976           1.55           0.12
## 4              2.54          57.85           1939           7.00           0.87
## 5              2.09          26.22           1966           6.13           0.31
## 6              1.41          34.45           1956           0.69           0.28
## 7              6.39          56.36           1954           8.42           0.49
## 8              0.45          25.61           1971           2.66           0.19
## 9              5.64          37.57           1960          11.74           0.33
## 10             2.77          42.34           1965           7.89           0.30
##    OwnOccLowQuart OwnOccMedVal OwnOccHiQuart OwnOccQrange RentLowQ RentMedian
## 1          215900       262600        326900       111000      685       1001
## 2          136300       164200        199900        63600      467        560
## 3           74700        90400        112000        37300      370        428
## 4           36400        49600         66500        30100      195        250
## 5           37700        53900         73100        35400      215        280
## 6          155100       179000        215500        60400      463        669
## 7           26300        37000         52400        26100      186        253
## 8           54500        70300         93700        39200      241        321
## 9           28600        43100         67400        38800      192        281
## 10          32200        49800         73600        41400      234        305
##    RentHighQ RentQrange MedRent MedRentPctHousInc MedOwnCostPctInc
## 1       1001        316    1001              23.8             21.1
## 2        672        205     627              27.6             20.7
## 3        520        150     484              24.1             21.7
## 4        309        114     333              28.7             20.6
## 5        349        134     340              26.4             17.3
## 6        824        361     736              24.4             20.8
## 7        325        139     338              26.3             15.1
## 8        387        146     355              25.2             20.7
## 9        369        177     353              29.6             19.4
## 10       376        142     380              23.8             17.1
##    MedOwnCostPctIncNoMtg NumInShelters NumStreet PctForeignBorn
## 1                   14.0            11         0         10.66
## 2                   12.5             0         0          8.30
```

```
## 3                   11.6            16         0           5.00
## 4                   14.5             0         0           2.04
## 5                   11.7           327         4           1.49
## 6                   12.5             0         0           9.19
## 7                   12.2            21         0           0.87
## 8                   12.8           125        15           1.99
## 9                   13.0            43         4           4.63
## 10                  12.9             1         0           2.97
##    PctBornSameState PctSameHouse85 PctSameCity85 PctSameState85 LemasSwornFT
## 1             53.72          65.29         78.09          89.14           NA
## 2             77.17          71.27         90.22          96.12           NA
## 3             44.77          36.60         61.26          82.85           NA
## 4             88.71          56.70         90.17          96.24           NA
## 5             64.35          42.29         70.61          85.66           NA
## 6             77.30          63.45         82.23          93.53           NA
## 7             73.70          54.85         85.55          91.51           NA
## 8             58.82          40.72         67.97          81.39           NA
## 9             75.59          42.33         74.05          92.12          198
## 10            65.73          44.95         74.82          88.66           NA
##    LemasSwFTPerPop LemasSwFTFieldOps LemasSwFTFieldPerPop LemasTotalReq
## 1               NA                NA                   NA            NA
## 2               NA                NA                   NA            NA
## 3               NA                NA                   NA            NA
## 4               NA                NA                   NA            NA
## 5               NA                NA                   NA            NA
## 6               NA                NA                   NA            NA
## 7               NA                NA                   NA            NA
## 8               NA                NA                   NA            NA
## 9           183.53               187               173.33         73432
## 10              NA                NA                   NA            NA
##    LemasTotReqPerPop PolicReqPerOffic PolicPerPop RacialMatchCommPol
## 1                 NA               NA          NA                NA
## 2                 NA               NA          NA                NA
## 3                 NA               NA          NA                NA
## 4                 NA               NA          NA                NA
## 5                 NA               NA          NA                NA
## 6                 NA               NA          NA                NA
## 7                 NA               NA          NA                NA
## 8                 NA               NA          NA                NA
## 9            68065.1            370.9       183.5             89.32
## 10                NA               NA          NA                NA
##    PctPolicWhite PctPolicBlack PctPolicHisp PctPolicAsian PctPolicMinor
## 1             NA            NA           NA            NA            NA
## 2             NA            NA           NA            NA            NA
## 3             NA            NA           NA            NA            NA
## 4             NA            NA           NA            NA            NA
## 5             NA            NA           NA            NA            NA
## 6             NA            NA           NA            NA            NA
## 7             NA            NA           NA            NA            NA
## 8             NA            NA           NA            NA            NA
## 9          78.28         11.11        10.61             0         21.72
## 10            NA            NA           NA            NA            NA
##    OfficAssgnDrugUnits NumKindsDrugsSeiz PolicAveOTWorked LandArea PopDens
## 1                   NA                NA               NA      6.5  1845.9
```

```
## 2                   NA                NA                NA     10.6  2186.7
## 3                   NA                NA                NA     10.6  2780.9
## 4                   NA                NA                NA      5.2  3217.7
## 5                   NA                NA                NA     70.4  1995.7
## 6                   NA                NA                NA     10.9  2643.5
## 7                   NA                NA                NA     39.2  1515.3
## 8                   NA                NA                NA     30.9  2399.3
## 9                   13                12              60.2     78.5  1319.3
## 10                  NA                NA                NA     38.7   816.1
##    PctUsePubTrans PolicCars PolicOperBudg LemasPctPolicOnPatr
## 1            9.63        NA            NA                  NA
## 2            3.84        NA            NA                  NA
## 3            4.37        NA            NA                  NA
## 4            3.31        NA            NA                  NA
## 5            0.97        NA            NA                  NA
## 6            9.62        NA            NA                  NA
## 7            0.70        NA            NA                  NA
## 8            1.41        NA            NA                  NA
## 9            0.76       100       9315474               94.44
## 10           0.00        NA            NA                  NA
##    LemasGangUnitDeploy LemasPctOfficDrugUn PolicBudgPerPop ViolentCrimesPerPop
## 1                   NA                0.00              NA               41.02
## 2                   NA                0.00              NA              127.56
## 3                   NA                0.00              NA              218.59
## 4                   NA                0.00              NA              306.64
## 5                   NA                0.00              NA              442.95
## 6                   NA                0.00              NA              226.63
## 7                   NA                0.00              NA              439.73
## 8                   NA                0.00              NA              115.31
## 9                   10                6.57         86346.3             1544.24
## 10                  NA                0.00              NA              722.02
```

Next, we will learn more about our dataset's nuances. Some of the attributes we would like to pay attention to include the dimension, the number of categorical versus numerical variables, and how many missing values there are. Below is a high level overview which includes the attributes of interest as well as some other useful to know information about our data, such as memory. This can be done by using the `glimpse` function from the `dplyr` package and the `introduce` function from the `DataExploration` package.

```
#This is like a transposed version of print: columns run down the page, and data runs across. This make
glimpse(CC)
```

```
## Observations: 1,994
## Variables: 125
## $ population            <dbl> 11980, 23123, 29344, 16656, 140494, 28700, 59...
## $ householdsize         <dbl> 3.10, 2.82, 2.43, 2.40, 2.45, 2.60, 2.45, 2.4...
## $ racepctblack          <dbl> 1.37, 0.80, 0.74, 1.70, 2.51, 1.60, 14.20, 0....
## $ racePctWhite          <dbl> 91.78, 95.57, 94.33, 97.35, 95.65, 96.57, 84....
## $ racePctAsian          <dbl> 6.50, 3.44, 3.43, 0.50, 0.90, 1.47, 0.40, 1.2...
## $ racePctHisp           <dbl> 1.88, 0.85, 2.35, 0.70, 0.95, 1.10, 0.63, 0.7...
## $ agePct12t21           <dbl> 12.47, 11.01, 11.36, 12.55, 18.09, 11.17, 15....
## $ agePct12t29           <dbl> 21.44, 21.30, 25.88, 25.20, 32.89, 27.41, 27....
## $ agePct16t24           <dbl> 10.93, 10.48, 11.01, 12.19, 20.04, 12.76, 14....
```

```
## $ agePct65up            <dbl> 11.33, 17.18, 10.28, 17.57, 13.26, 14.42, 14....
## $ numbUrban             <dbl> 11980, 23123, 29344, 0, 140494, 28700, 59449,...
## $ pctUrban              <dbl> 100.00, 100.00, 100.00, 0.00, 100.00, 100.00,...
## $ medIncome             <dbl> 75122, 47917, 35669, 20580, 21577, 42805, 232...
## $ pctWWage              <dbl> 89.24, 78.99, 82.00, 68.15, 75.78, 79.47, 71....
## $ pctWFarmSelf          <dbl> 1.55, 1.11, 1.15, 0.24, 1.00, 0.39, 0.67, 2.9...
## $ pctWInvInc            <dbl> 70.20, 64.11, 55.73, 38.95, 41.15, 47.70, 35....
## $ pctWSocSec            <dbl> 23.62, 35.50, 22.25, 39.48, 29.31, 30.23, 32....
## $ pctWPubAsst           <dbl> 1.03, 2.75, 2.94, 11.71, 7.12, 5.41, 8.81, 4....
## $ pctWRetire            <dbl> 18.39, 22.85, 14.56, 18.33, 14.09, 17.23, 22....
## $ medFamInc             <dbl> 79584, 55323, 42112, 26501, 27705, 50394, 289...
## $ perCapInc             <dbl> 29711, 20148, 16946, 10810, 11878, 18193, 121...
## $ whitePerCap           <dbl> 30233, 20191, 17103, 10909, 12029, 18276, 125...
## $ blackPerCap           <dbl> 13600, 18137, 16644, 9984, 7382, 17342, 9820,...
## $ indianPerCap          <dbl> 5725, 0, 21606, 4941, 10264, 21482, 6634, 534...
## $ AsianPerCap           <dbl> 27101, 20074, 15528, 3541, 10753, 12639, 8802...
## $ OtherPerCap           <dbl> 5115, 5250, 5954, 2451, 7192, 21852, 7428, 53...
## $ HispPerCap            <dbl> 22838, 12222, 8405, 4391, 8104, 22594, 6187, ...
## $ NumUnderPov           <dbl> 227, 885, 1389, 2831, 23223, 1126, 10320, 960...
## $ PctPopUnderPov        <dbl> 1.96, 3.98, 4.75, 17.23, 17.78, 4.01, 17.98, ...
## $ PctLess9thGrade       <dbl> 5.81, 5.61, 2.80, 11.05, 8.76, 4.49, 10.09, 5...
## $ PctNotHSGrad          <dbl> 9.90, 13.72, 9.09, 33.68, 23.03, 13.89, 28.67...
## $ PctBSorMore           <dbl> 48.18, 29.89, 30.13, 10.81, 20.66, 27.01, 12....
## $ PctUnemployed         <dbl> 2.70, 2.43, 4.01, 9.86, 5.72, 4.85, 8.19, 4.1...
## $ PctEmploy             <dbl> 64.55, 61.96, 69.80, 54.74, 59.02, 65.42, 56....
## $ PctEmplManu           <dbl> 14.65, 12.26, 15.95, 31.22, 14.31, 14.02, 27....
## $ PctEmplProfServ       <dbl> 28.82, 29.28, 21.52, 27.43, 26.83, 27.17, 21....
## $ PctOccupManu          <dbl> 5.49, 6.39, 8.79, 26.76, 14.72, 8.50, 21.92, ...
## $ PctOccupMgmtProf      <dbl> 50.73, 37.64, 32.48, 22.71, 23.42, 32.78, 18....
## $ MalePctDivorce        <dbl> 3.67, 4.23, 10.10, 10.98, 11.40, 5.97, 13.28,...
## $ MalePctNevMarr        <dbl> 26.38, 27.99, 25.78, 28.15, 33.32, 36.05, 28....
## $ FemalePctDiv          <dbl> 5.22, 6.45, 14.76, 14.47, 14.46, 9.06, 16.33,...
## $ TotalPctDiv           <dbl> 4.47, 5.42, 12.55, 12.91, 13.04, 7.64, 14.94,...
## $ PersPerFam            <dbl> 3.22, 3.11, 2.95, 2.98, 2.89, 3.14, 2.95, 3.0...
## $ PctFam2Par            <dbl> 91.43, 86.91, 78.54, 64.02, 71.94, 79.53, 62....
## $ PctKids2Par           <dbl> 90.17, 85.33, 78.85, 62.36, 69.79, 79.76, 58....
## $ PctYoungKids2Par      <dbl> 95.78, 96.82, 92.37, 65.38, 79.76, 92.05, 69....
## $ PctTeen2Par           <dbl> 95.81, 86.46, 75.72, 67.43, 75.33, 77.12, 62....
## $ PctWorkMomYoungKids   <dbl> 44.56, 51.14, 66.08, 59.59, 62.96, 65.16, 63....
## $ PctWorkMom            <dbl> 58.88, 62.43, 74.19, 70.27, 70.52, 72.81, 72....
## $ NumKidsBornNeverMar   <dbl> 31, 43, 164, 561, 1511, 263, 2368, 751, 3537,...
## $ PctKidsBornNeverMar   <dbl> 0.36, 0.24, 0.88, 3.84, 1.58, 1.18, 4.66, 1.6...
## $ NumImmig              <dbl> 1277, 1920, 1468, 339, 2091, 2637, 517, 1474,...
## $ PctImmigRecent        <dbl> 8.69, 5.21, 16.42, 13.86, 21.33, 11.38, 13.15...
## $ PctImmigRec5          <dbl> 13.00, 8.65, 23.98, 13.86, 30.56, 16.27, 22.8...
## $ PctImmigRec8          <dbl> 20.99, 13.33, 32.08, 15.34, 38.02, 23.93, 28....
## $ PctImmigRec10         <dbl> 30.93, 22.50, 35.63, 15.34, 45.48, 27.76, 33....
## $ PctRecentImmig        <dbl> 0.93, 0.43, 0.82, 0.28, 0.32, 1.05, 0.11, 0.4...
## $ PctRecImmig5          <dbl> 1.39, 0.72, 1.20, 0.28, 0.45, 1.49, 0.20, 0.6...
## $ PctRecImmig8          <dbl> 2.24, 1.11, 1.61, 0.31, 0.57, 2.20, 0.25, 0.9...
## $ PctRecImmig10         <dbl> 3.30, 1.87, 1.78, 0.31, 0.68, 2.55, 0.29, 1.0...
## $ PctSpeakEnglOnly      <dbl> 85.68, 87.79, 93.11, 94.98, 96.87, 89.98, 97....
## $ PctNotSpeakEnglWell   <dbl> 1.37, 1.81, 1.14, 0.56, 0.60, 0.60, 0.28, 0.4...
## $ PctLargHouseFam       <dbl> 4.81, 4.25, 2.97, 3.93, 3.08, 5.08, 3.85, 2.5...
```

```
## $ PctLargHouseOccup     <dbl> 4.17, 3.34, 2.05, 2.56, 1.92, 3.46, 2.55, 1.5...
## $ PersPerOccupHous      <dbl> 2.99, 2.70, 2.42, 2.37, 2.28, 2.55, 2.36, 2.3...
## $ PersPerOwnOccHous     <dbl> 3.00, 2.83, 2.69, 2.51, 2.37, 2.89, 2.42, 2.7...
## $ PersPerRentOccHous    <dbl> 2.84, 1.96, 2.06, 2.20, 2.16, 2.09, 2.27, 1.9...
## $ PctPersOwnOccup       <dbl> 91.46, 89.03, 64.18, 58.18, 57.81, 64.62, 65....
## $ PctPersDenseHous      <dbl> 0.39, 1.01, 2.03, 1.21, 2.11, 1.47, 1.90, 1.6...
## $ PctHousLess3BR        <dbl> 11.06, 23.60, 47.46, 45.66, 53.19, 47.35, 56....
## $ MedNumBR              <dbl> 3, 3, 3, 3, 2, 3, 2, 2, 2, 2, 2, 2, 3, 3, 2, ...
## $ HousVacant            <dbl> 64, 240, 544, 669, 5119, 566, 2051, 1562, 560...
## $ PctHousOccup          <dbl> 98.37, 97.15, 95.68, 91.19, 91.81, 95.11, 92....
## $ PctHousOwnOcc         <dbl> 91.01, 84.88, 57.79, 54.89, 55.50, 56.96, 63....
## $ PctVacantBoarded      <dbl> 3.12, 0.00, 0.92, 2.54, 2.09, 1.41, 6.39, 0.4...
## $ PctVacMore6Mos        <dbl> 37.50, 18.33, 7.54, 57.85, 26.22, 34.45, 56.3...
## $ MedYrHousBuilt        <dbl> 1959, 1958, 1976, 1939, 1966, 1956, 1954, 197...
## $ PctHousNoPhone        <dbl> 0.00, 0.31, 1.55, 7.00, 6.13, 0.69, 8.42, 2.6...
## $ PctWOFullPlumb        <dbl> 0.28, 0.14, 0.12, 0.87, 0.31, 0.28, 0.49, 0.1...
## $ OwnOccLowQuart        <dbl> 215900, 136300, 74700, 36400, 37700, 155100, ...
## $ OwnOccMedVal          <dbl> 262600, 164200, 90400, 49600, 53900, 179000, ...
## $ OwnOccHiQuart         <dbl> 326900, 199900, 112000, 66500, 73100, 215500,...
## $ OwnOccQrange          <dbl> 111000, 63600, 37300, 30100, 35400, 60400, 26...
## $ RentLowQ              <dbl> 685, 467, 370, 195, 215, 463, 186, 241, 192, ...
## $ RentMedian            <dbl> 1001, 560, 428, 250, 280, 669, 253, 321, 281,...
## $ RentHighQ             <dbl> 1001, 672, 520, 309, 349, 824, 325, 387, 369,...
## $ RentQrange            <dbl> 316, 205, 150, 114, 134, 361, 139, 146, 177, ...
## $ MedRent               <dbl> 1001, 627, 484, 333, 340, 736, 338, 355, 353,...
## $ MedRentPctHousInc     <dbl> 23.8, 27.6, 24.1, 28.7, 26.4, 24.4, 26.3, 25....
## $ MedOwnCostPctInc      <dbl> 21.1, 20.7, 21.7, 20.6, 17.3, 20.8, 15.1, 20....
## $ MedOwnCostPctIncNoMtg <dbl> 14.0, 12.5, 11.6, 14.5, 11.7, 12.5, 12.2, 12....
## $ NumInShelters         <dbl> 11, 0, 16, 0, 327, 0, 21, 125, 43, 1, 20, 28,...
## $ NumStreet             <dbl> 0, 0, 0, 0, 4, 0, 0, 15, 4, 0, 49, 2, 0, 1, 1...
## $ PctForeignBorn        <dbl> 10.66, 8.30, 5.00, 2.04, 1.49, 9.19, 0.87, 1....
## $ PctBornSameState      <dbl> 53.72, 77.17, 44.77, 88.71, 64.35, 77.30, 73....
## $ PctSameHouse85        <dbl> 65.29, 71.27, 36.60, 56.70, 42.29, 63.45, 54....
## $ PctSameCity85         <dbl> 78.09, 90.22, 61.26, 90.17, 70.61, 82.23, 85....
## $ PctSameState85        <dbl> 89.14, 96.12, 82.85, 96.24, 85.66, 93.53, 91....
## $ LemasSwornFT          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 198, NA, NA, ...
## $ LemasSwFTPerPop       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 183.53, NA, N...
## $ LemasSwFTFieldOps     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 187, NA, NA, ...
## $ LemasSwFTFieldPerPop  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 173.33, NA, N...
## $ LemasTotalReq         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 73432, NA, NA...
## $ LemasTotReqPerPop     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 68065.1, NA, ...
## $ PolicReqPerOffic      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 370.9, NA, NA...
## $ PolicPerPop           <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 183.5, NA, NA...
## $ RacialMatchCommPol    <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 89.32, NA, NA...
## $ PctPolicWhite         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 78.28, NA, NA...
## $ PctPolicBlack         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 11.11, NA, NA...
## $ PctPolicHisp          <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 10.61, NA, NA...
## $ PctPolicAsian         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 0.00, NA, NA,...
## $ PctPolicMinor         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 21.72, NA, NA...
## $ OfficAssgnDrugUnits   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 13, NA, NA, N...
## $ NumKindsDrugsSeiz     <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 12, NA, NA, N...
## $ PolicAveOTWorked      <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 60.2, NA, NA,...
## $ LandArea              <dbl> 6.5, 10.6, 10.6, 5.2, 70.4, 10.9, 39.2, 30.9,...
## $ PopDens               <dbl> 1845.9, 2186.7, 2780.9, 3217.7, 1995.7, 2643....
```

```
## $ PctUsePubTrans        <dbl> 9.63, 3.84, 4.37, 3.31, 0.97, 9.62, 0.70, 1.4...
## $ PolicCars             <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 100, NA, NA, ...
## $ PolicOperBudg         <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 9315474, NA, ...
## $ LemasPctPolicOnPatr   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 94.44, NA, NA...
## $ LemasGangUnitDeploy   <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 10, NA, NA, N...
## $ LemasPctOfficDrugUn   <dbl> 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.0...
## $ PolicBudgPerPop       <dbl> NA, NA, NA, NA, NA, NA, NA, NA, 86346.3, NA, ...
## $ ViolentCrimesPerPop   <dbl> 41.02, 127.56, 218.59, 306.64, 442.95, 226.63...
```

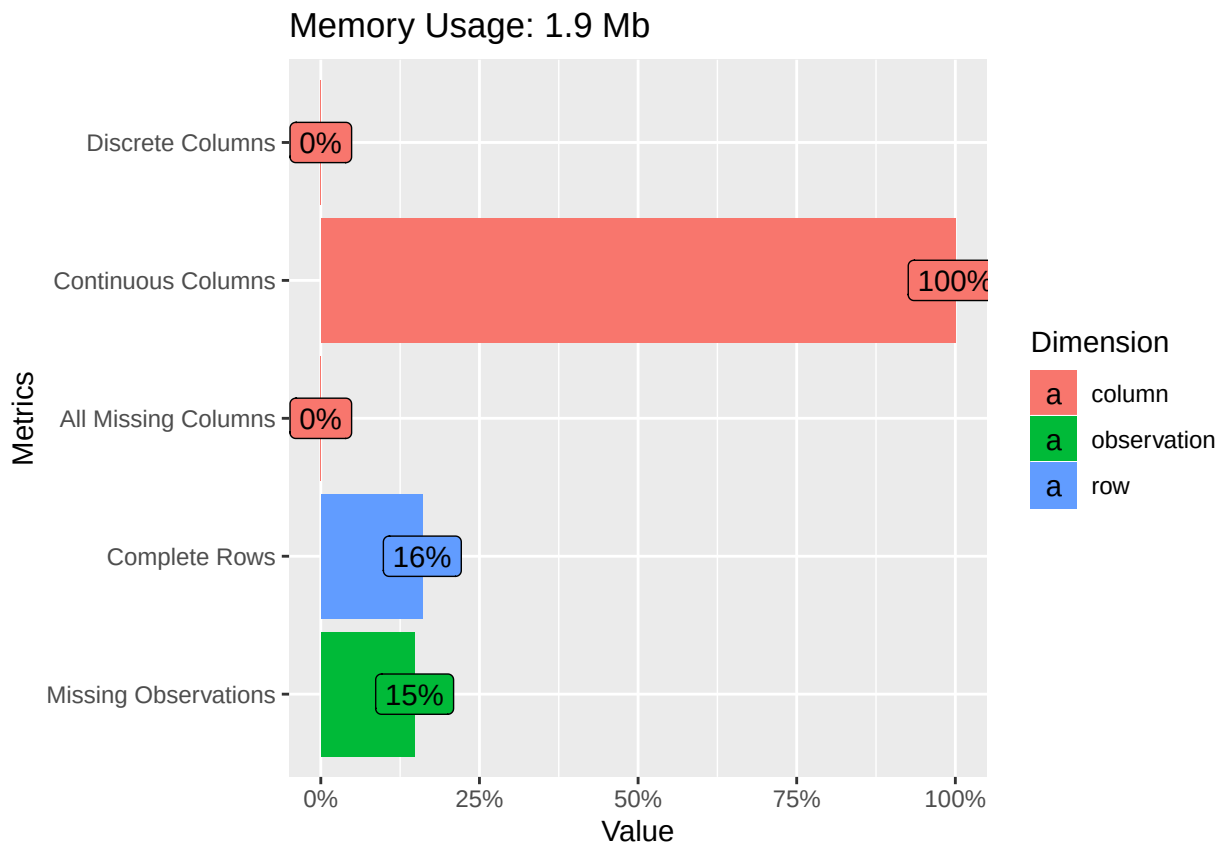
```
#summarize key attributes of the data
introduce(CC)
```

```
##   rows columns discrete_columns continuous_columns all_missing_columns
## 1 1994     125                0                 125                   0
##   total_missing_values complete_rows total_observations memory_usage
## 1                36851           319              249250      2020864
```

Here, "complete_rows" refers to the number of rows without any missing values, "all_missing_columns" refers to the number of missing columns (the entire column is NA), "total_observations" refers to each value in the dataset (including missing values), and "discrete_columns" refers to the number of categorical variables in our data. From our findings, we see that there are no cateogorial variables. Additionally, out of the 249250 values, there are a total of 36851 missing values.

We can visualize our findings with a barplot that gives us the proportions of each attribute. The `plot_intro` function from the `DataExploration` package lets us do just that.

```
plot_intro(CC)
```



From our visualization, it becomes clear that:
1. Only roughly 16% of all rows are not completely missing
2. About 15% of the values in the dataset are missing
Missing values definitely will cause problems. Hence the next step will be to take a closer look at what is missing and what we can do to alleviate the problem.

The following code breaks down the number of missing values by variable with a for loop, looping over each coloumn in our dataset.

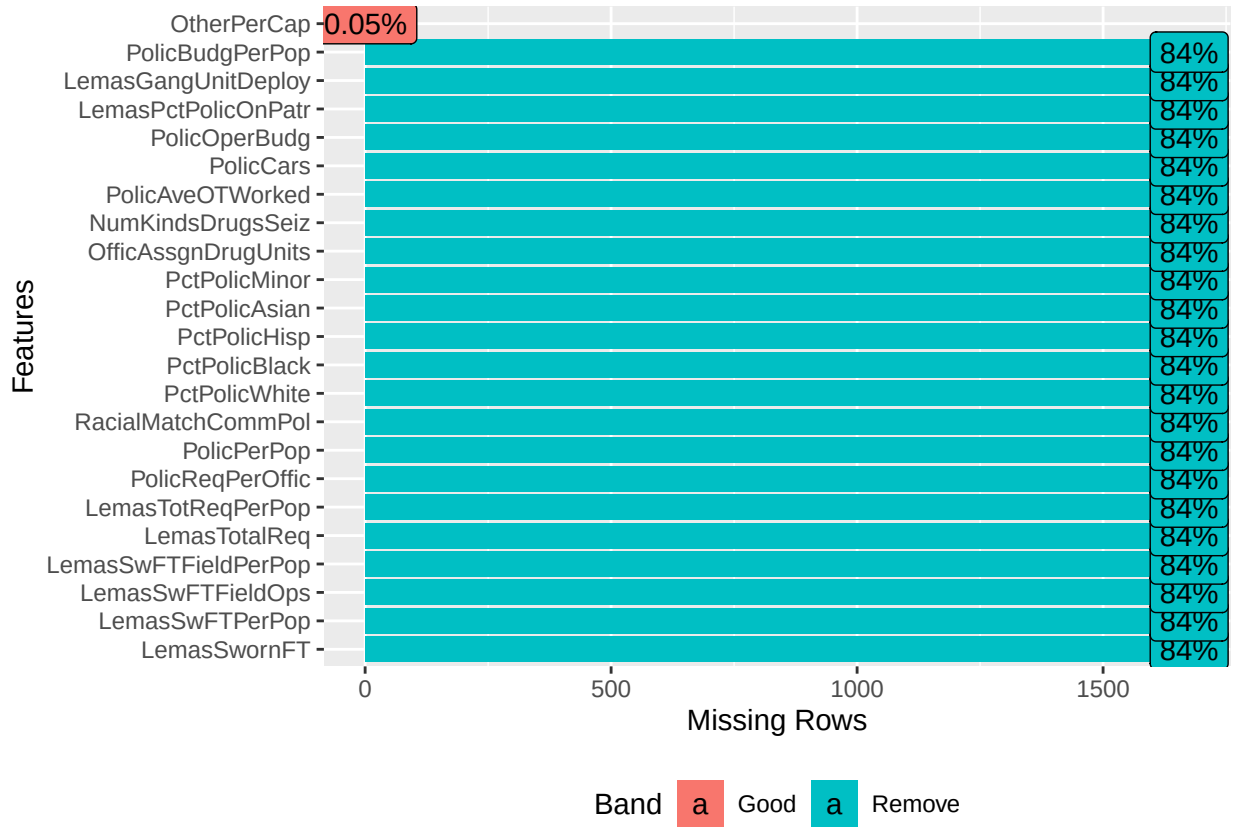
```
na_per_col <- c()
for(i in 1:ncol(CC)){
  na_per_col[i] <- sum(is.na(CC[,i]))
}
names(na_per_col) <- names(CC)
na_per_col
```

```
##            population         householdsize           racepctblack
##                     0                     0                      0
##          racePctWhite          racePctAsian           racePctHisp
##                     0                     0                      0
##           agePct12t21           agePct12t29           agePct16t24
##                     0                     0                      0
##           agePct65up             numbUrban               pctUrban
##                     0                     0                      0
##             medIncome              pctWWage          pctWFarmSelf
##                     0                     0                      0
##            pctWInvInc            pctWSocSec           pctWPubAsst
##                     0                     0                      0
##            pctWRetire             medFamInc              perCapInc
##                     0                     0                      0
##           whitePerCap           blackPerCap           indianPerCap
##                     0                     0                      0
##           AsianPerCap           OtherPerCap             HispPerCap
##                     0                     1                      0
##           NumUnderPov        PctPopUnderPov       PctLess9thGrade
##                     0                     0                      0
##          PctNotHSGrad           PctBSorMore         PctUnemployed
##                     0                     0                      0
##             PctEmploy           PctEmplManu       PctEmplProfServ
##                     0                     0                      0
##          PctOccupManu      PctOccupMgmtProf         MalePctDivorce
##                     0                     0                      0
##        MalePctNevMarr          FemalePctDiv            TotalPctDiv
##                     0                     0                      0
##            PersPerFam            PctFam2Par            PctKids2Par
##                     0                     0                      0
##      PctYoungKids2Par           PctTeen2Par    PctWorkMomYoungKids
##                     0                     0                      0
##            PctWorkMom    NumKidsBornNeverMar    PctKidsBornNeverMar
##                     0                     0                      0
##              NumImmig        PctImmigRecent           PctImmigRec5
##                     0                     0                      0
##           PctImmigRec8         PctImmigRec10         PctRecentImmig
##                     0                     0                      0
```

```
##          PctRecImmig5          PctRecImmig8         PctRecImmig10
##                     0                     0                     0
##      PctSpeakEnglOnly   PctNotSpeakEnglWell       PctLargHouseFam
##                     0                     0                     0
##     PctLargHouseOccup      PersPerOccupHous     PersPerOwnOccHous
##                     0                     0                     0
##    PersPerRentOccHous       PctPersOwnOccup      PctPersDenseHous
##                     0                     0                     0
##        PctHousLess3BR             MedNumBR           HousVacant
##                     0                     0                     0
##          PctHousOccup         PctHousOwnOcc      PctVacantBoarded
##                     0                     0                     0
##        PctVacMore6Mos        MedYrHousBuilt        PctHousNoPhone
##                     0                     0                     0
##        PctWOFullPlumb        OwnOccLowQuart          OwnOccMedVal
##                     0                     0                     0
##         OwnOccHiQuart          OwnOccQrange              RentLowQ
##                     0                     0                     0
##            RentMedian             RentHighQ           RentQrange
##                     0                     0                     0
##               MedRent     MedRentPctHousInc      MedOwnCostPctInc
##                     0                     0                     0
## MedOwnCostPctIncNoMtg         NumInShelters             NumStreet
##                     0                     0                     0
##        PctForeignBorn      PctBornSameState        PctSameHouse85
##                     0                     0                     0
##         PctSameCity85        PctSameState85          LemasSwornFT
##                     0                     0                  1675
##       LemasSwFTPerPop     LemasSwFTFieldOps  LemasSwFTFieldPerPop
##                  1675                  1675                  1675
##         LemasTotalReq     LemasTotReqPerPop      PolicReqPerOffic
##                  1675                  1675                  1675
##           PolicPerPop    RacialMatchCommPol         PctPolicWhite
##                  1675                  1675                  1675
##         PctPolicBlack          PctPolicHisp         PctPolicAsian
##                  1675                  1675                  1675
##         PctPolicMinor   OfficAssgnDrugUnits     NumKindsDrugsSeiz
##                  1675                  1675                  1675
##      PolicAveOTWorked              LandArea               PopDens
##                  1675                     0                     0
##        PctUsePubTrans             PolicCars         PolicOperBudg
##                     0                  1675                  1675
##   LemasPctPolicOnPatr   LemasGangUnitDeploy   LemasPctOfficDrugUn
##                  1675                  1675                     0
##       PolicBudgPerPop   ViolentCrimesPerPop
##                  1675                     0
```

This can be visualized as well with a barplot which illustrates the proportion of missing values for each variable given that the variable contains missing values. The `plot_missing` function from the `DataExploration` package allows us to accomplish this.

```
plot_missing(CC[,na_per_col!= 0])
```



From our illustration, we observe that out of the 23 variables that contain missing values, all of them except for "OtherPerCap" has a significant portion of missing values. Because those variables are mostly missing values, we can drop them using the `drop_columns` function since they probably will not provide us with much information. Doing this drop reduces the number of variables in our dataset to 103 (previously 125) as shown with the `dim` function.

```
cleaned_CC <- drop_columns(CC, names(na_per_col)[na_per_col == 1675])
cleaned_CC <- as.data.frame(cleaned_CC)
dim(cleaned_CC)
```

```
## [1] 1994  103
```

Now that we have learned and cleaned our dataset a little bit, we can get a general summary of our data using the `summary` function. This will provide us with some useful summary statistics of our variables as well as give us insight as to how they are distributed. We can visualize these distributions by plotting the histograms corresponding to each variable.

```
summary(cleaned_CC)
```

```
##    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
```

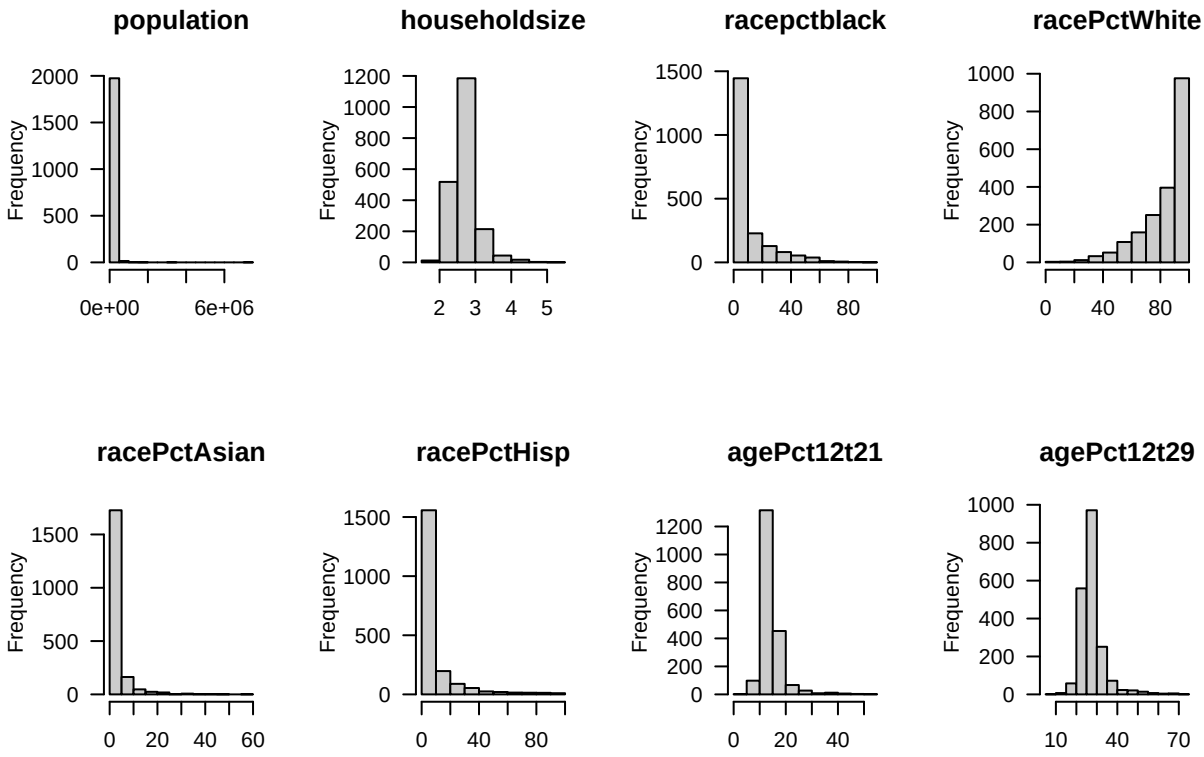
```
##  Max.   :7322564   Max.   :5.280   Max.   :96.67   Max.   :99.63
##
##   racePctAsian      racePctHisp      agePct12t21     agePct12t29
##  Min.   : 0.0300   Min.   : 0.120   Min.   : 4.58   Min.   : 9.38
##  1st Qu.: 0.6125   1st Qu.: 0.920   1st Qu.:12.23   1st Qu.:24.38
##  Median : 1.2400   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
```

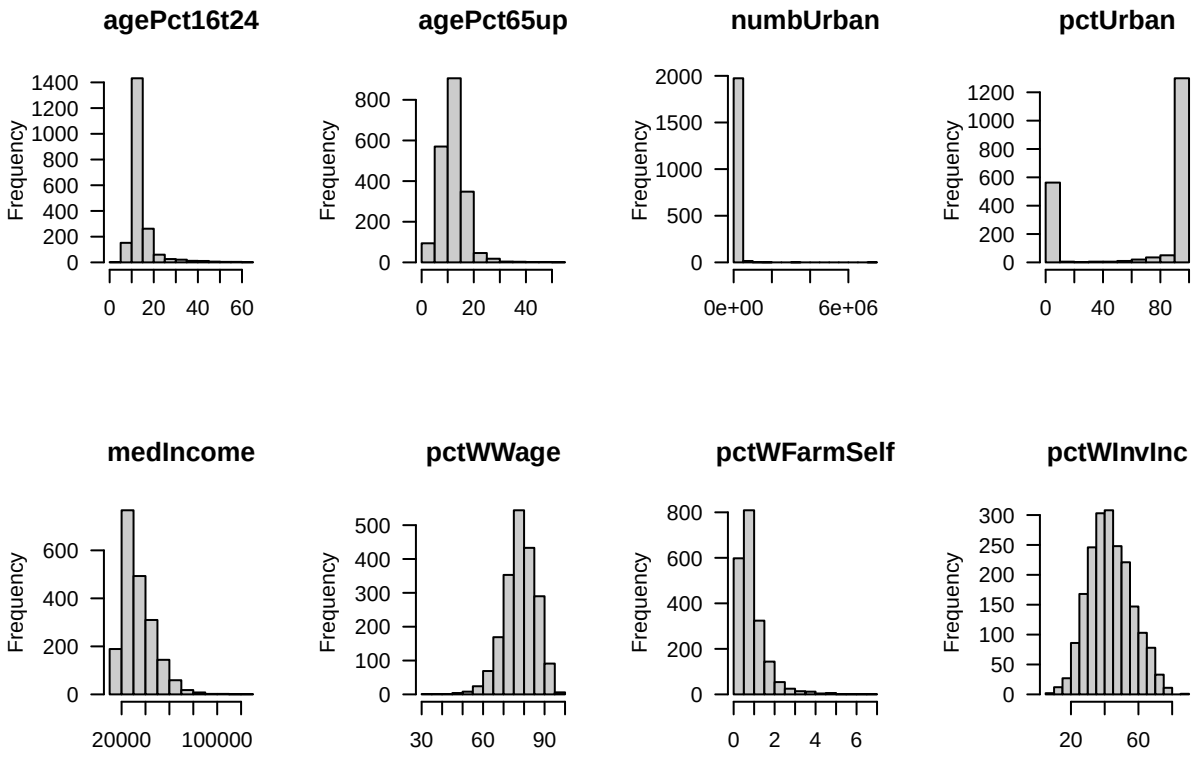
```
##  Mean   :11.796   Mean   : 9.444   Mean   :22.70   Mean   :22.99
##  3rd Qu.:17.078   3rd Qu.:12.245   3rd Qu.:29.66   3rd Qu.:28.93
##  Max.   :48.820   Max.   :49.890   Max.   :73.66   Max.   :73.63
##
##  PctUnemployed      PctEmploy      PctEmplManu    PctEmplProfServ
##  Min.   : 1.320   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
```

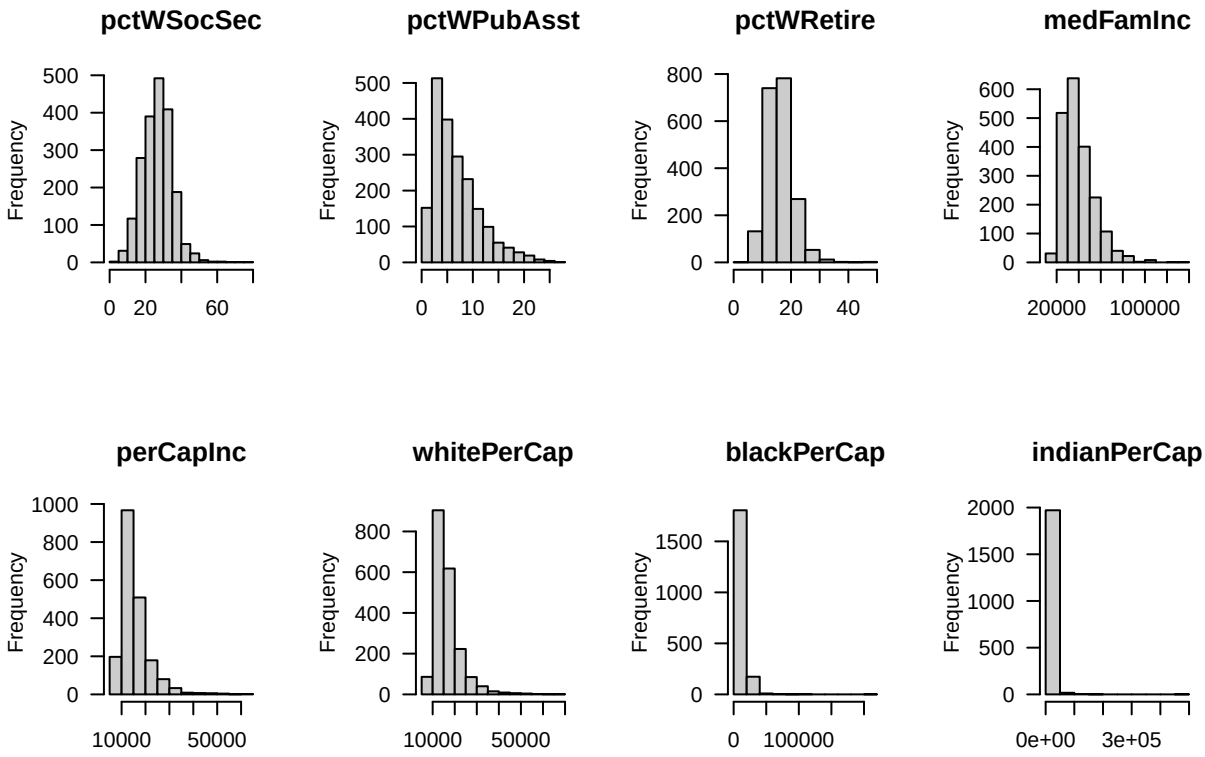
```
##  1st Qu.: 0.180   1st Qu.: 0.290   1st Qu.: 0.410   1st Qu.: 0.540
##  Median : 0.530   Median : 0.780   Median : 1.080   Median : 1.380
##  Mean   : 1.149   Mean   : 1.781   Mean   : 2.424   Mean   : 3.094
##  3rd Qu.: 1.370   3rd Qu.: 2.180   3rd Qu.: 2.870   3rd Qu.: 3.680
##  Max.   :13.710   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
##
```

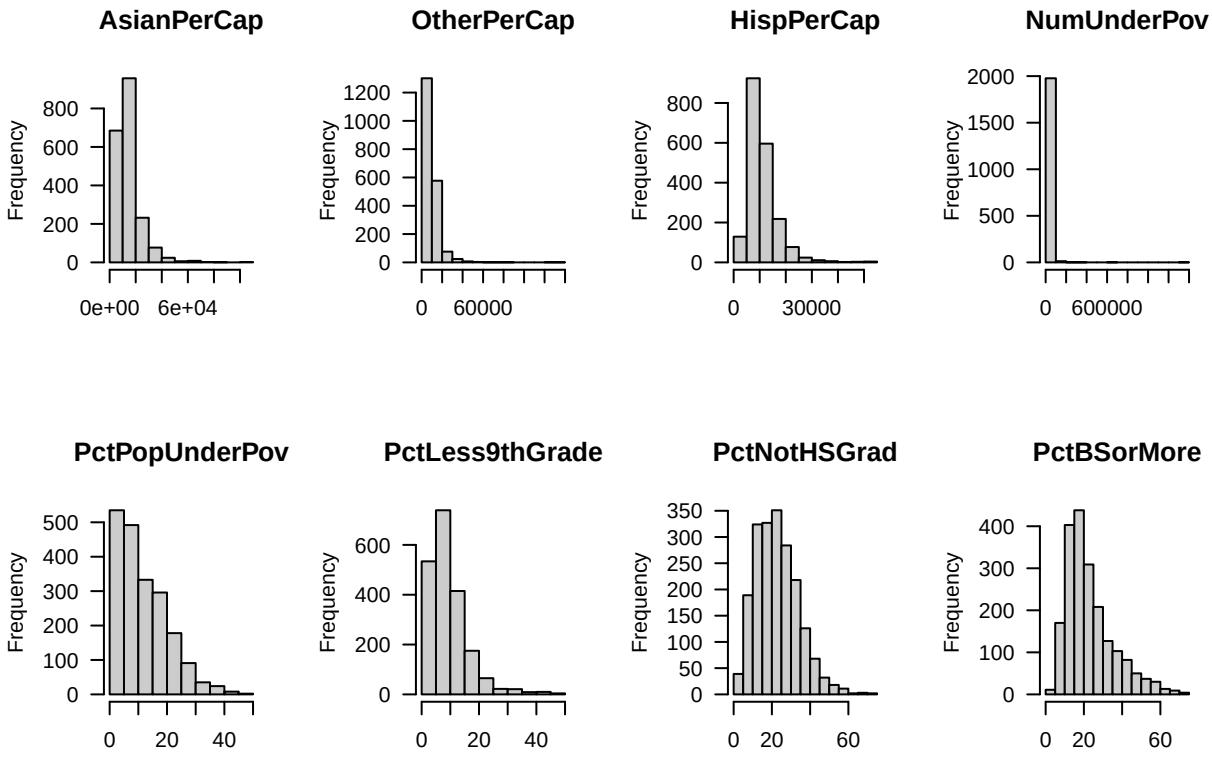
```
##    RentMedian       RentHighQ        RentQrange       MedRent
##  Min.   : 120.0   Min.   : 182.0   Min.   :  0.0   Min.   : 192.0
##  1st Qu.: 286.0   1st Qu.: 361.2   1st Qu.:139.0   1st Qu.: 363.0
##  Median : 394.0   Median : 484.0   Median :173.0   Median : 467.0
##  Mean   : 428.4   Mean   : 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     LandArea          PopDens
##  Min.   :27.95   Min.   :32.83   Min.   :   0.90   Min.   :   10
##  1st Qu.:71.92   1st Qu.:84.73   1st Qu.:   7.40   1st Qu.: 1171
##  Median :79.31   Median :89.64   Median :  13.70   Median : 1996
##  Mean   :77.11   Mean   :87.73   Mean   :  27.96   Mean   : 2790
##  3rd Qu.:84.70   3rd Qu.:92.73   3rd Qu.:  25.77   3rd Qu.: 3270
##  Max.   :96.59   Max.   :99.90   Max.   :3569.80   Max.   :44230
##
##  PctUsePubTrans   LemasPctOfficDrugUn ViolentCrimesPerPop
##  Min.   : 0.000   Min.   : 0.00       Min.   :   0.0
##  1st Qu.: 0.350   1st Qu.: 0.00       1st Qu.: 161.7
##  Median : 1.220   Median : 0.00       Median : 374.1
##  Mean   : 3.063   Mean   : 1.01       Mean   : 589.1
##  3rd Qu.: 3.377   3rd Qu.: 0.00       3rd Qu.: 794.4
##  Max.   :54.330   Max.   :48.44       Max.   :4877.1
##
```

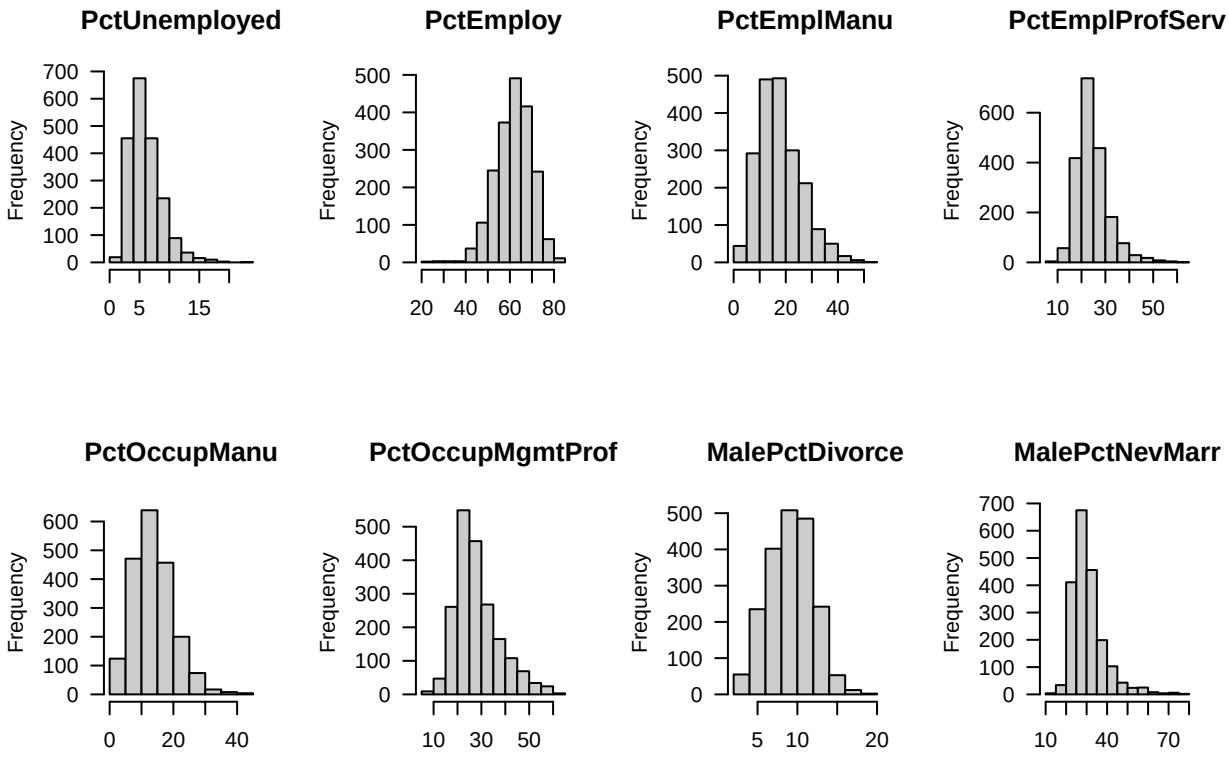
```
par(mfrow=c(2,4))
for (name in names(cleaned_CC)){
  hist(cleaned_CC[ ,name], col ='gray80', main =paste(name), xlab ='', las = 1)
}
```

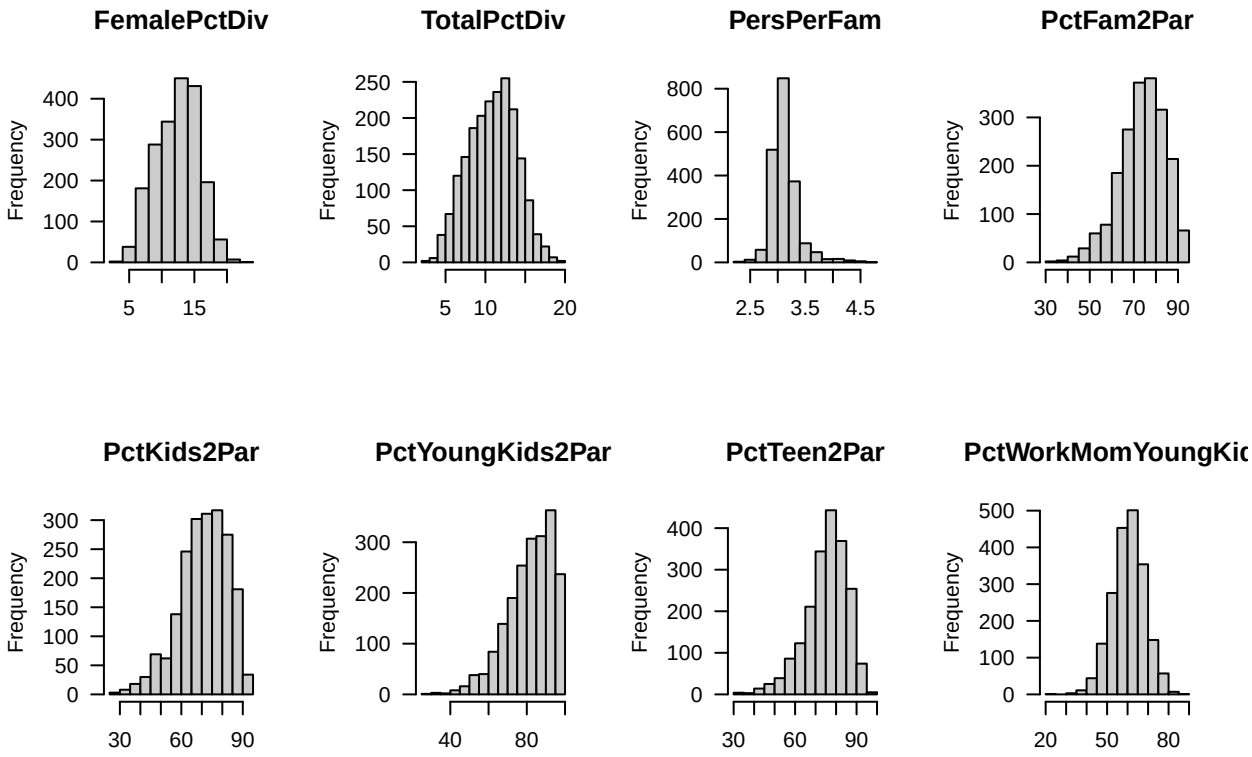

population
Frequency
0e+00
6e+06
householdsize
2
3
4
5
racepctblack
0
40
80
racePctWhite
racePctAsian
0
20
40
60
racePctHisp
agePct12t21
agePct12t29
10
40
70

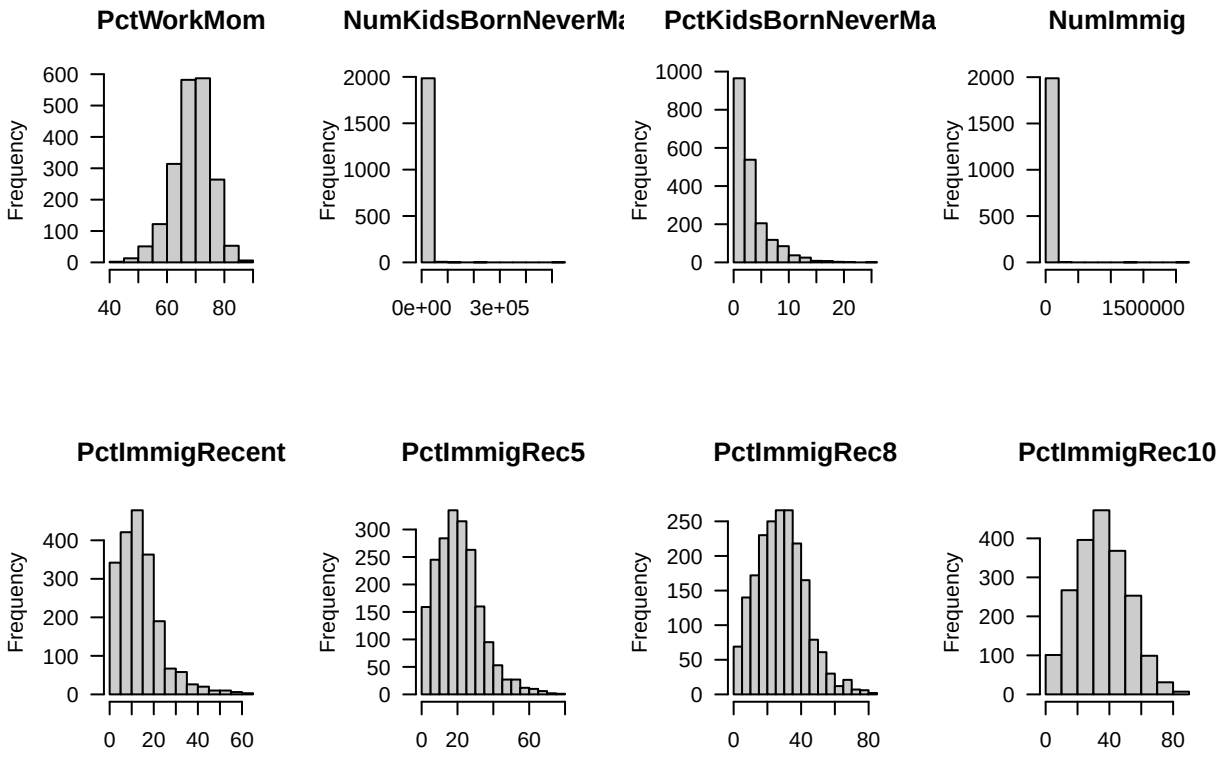

agePct16t24
Frequency
0
200
400
600
800
1000
1200
1400
0
20
40
60
agePct65up
Frequency
0
200
400
600
800
0
20
40
numbUrban
Frequency
0
500
1000
1500
2000
0e+00
6e+06
pctUrban
Frequency
0
200
400
600
800
1000
1200
0
40
80
medIncome
Frequency
0
200
400
600
20000
100000
pctWWage
Frequency
0
100
200
300
400
500
30
60
90
pctWFarmSelf
Frequency
0
200
400
600
800
0
2
4
6
pctWInvInc
Frequency
0
50
100
150
200
250
300
20
60

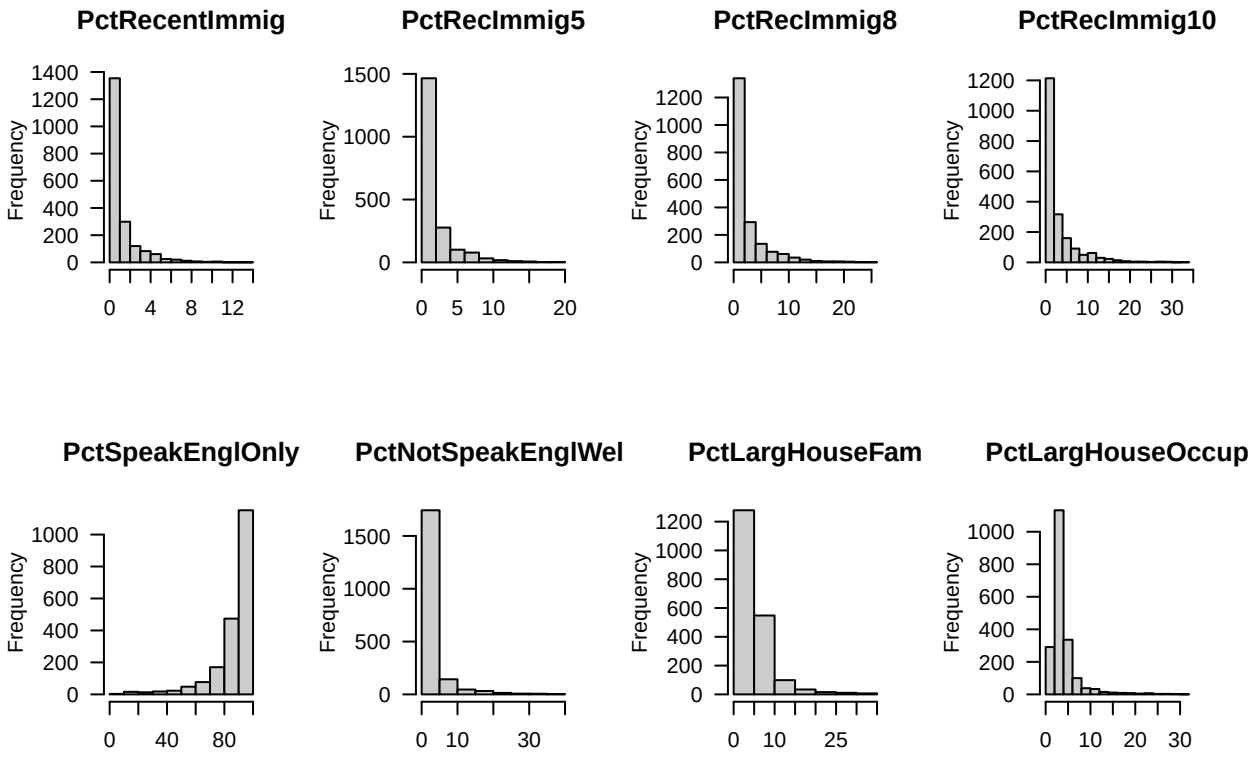

pctWSocSec
Frequency
0 100 200 300 400 500
0 20 60
pctWPubAsst
Frequency
0 100 200 300 400 500
0 10 20
pctWRetire
Frequency
0 200 400 600 800
0 20 40
medFamInc
Frequency
0 100 200 300 400 500 600
20000 100000
perCapInc
Frequency
0 200 400 600 800 1000
10000 50000
whitePerCap
Frequency
0 200 400 600 800
10000 50000
blackPerCap
Frequency
0 500 1000 1500
0 100000
indianPerCap
Frequency
0 500 1000 1500 2000
0e+00 3e+05

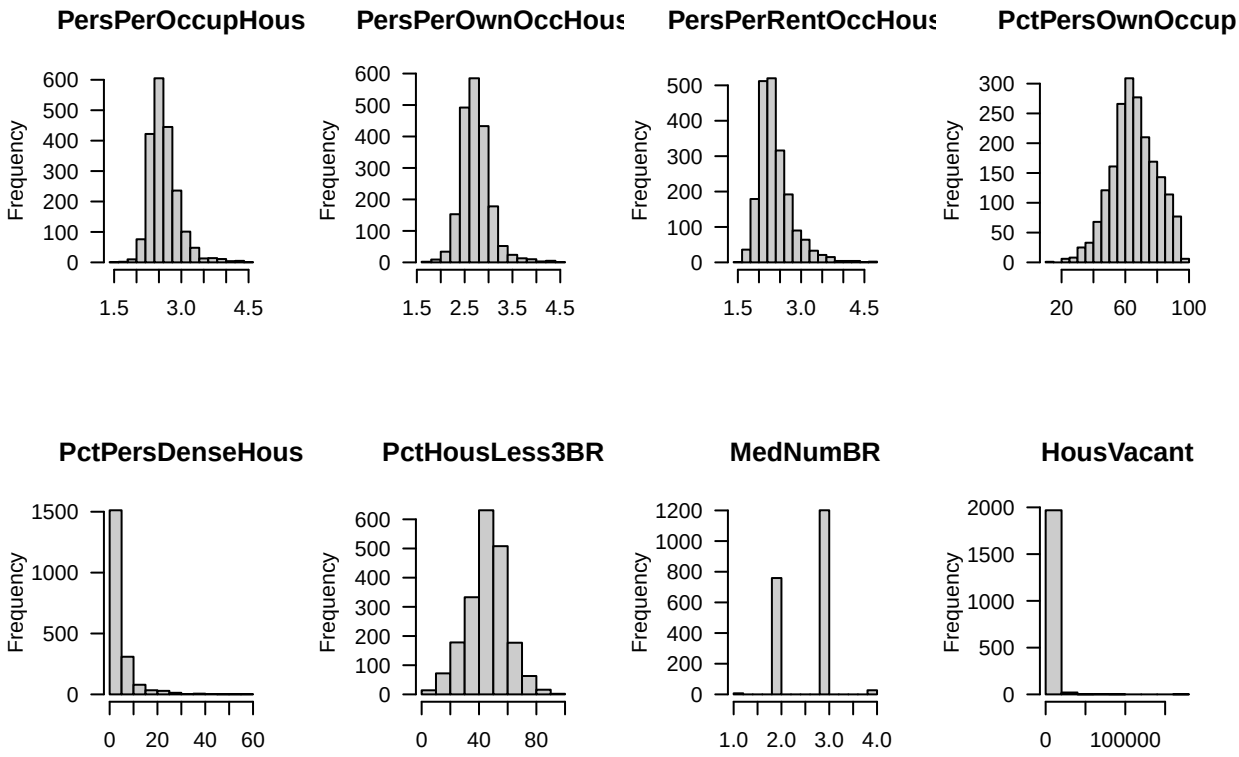

AsianPerCap
Frequency
0
200
400
600
800
0e+00
6e+04
OtherPerCap
Frequency
0
200
400
600
800
1000
1200
0
60000
HispPerCap
Frequency
0
200
400
600
800
0
30000
NumUnderPov
Frequency
0
500
1000
1500
2000
0
600000
PctPopUnderPov
Frequency
0
100
200
300
400
500
0
20
40
PctLess9thGrade
Frequency
0
200
400
600
0
20
40
PctNotHSGrad
Frequency
0
50
100
150
200
250
300
350
0
20
60
PctBSorMore
Frequency
0
100
200
300
400
0
20
60

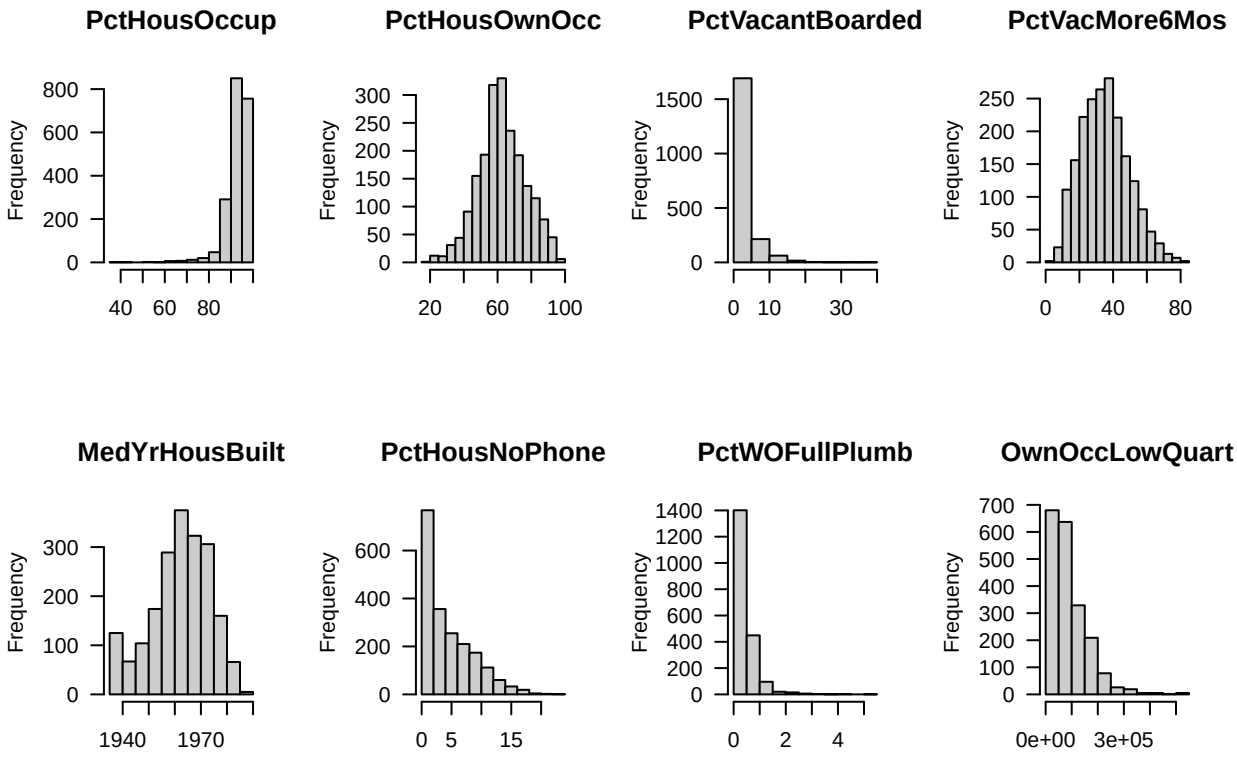

PctUnemployed
PctEmploy
PctEmplManu
PctEmplProfServ
PctOccupManu
PctOccupMgmtProf
MalePctDivorce
MalePctNevMarr
Frequency

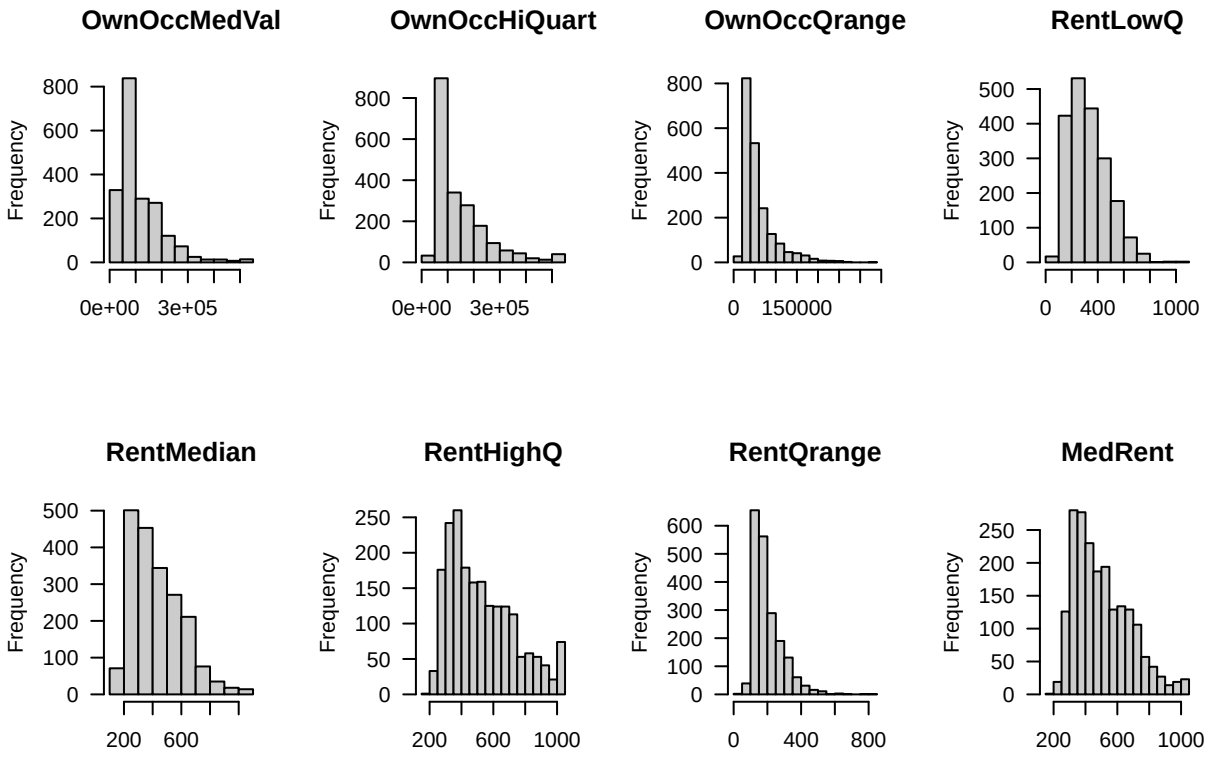

FemalePctDiv
Frequency
0
100
200
300
400
5
15
TotalPctDiv
Frequency
0
50
100
150
200
250
5
10
20
PersPerFam
Frequency
0
200
400
600
800
2.5
3.5
4.5
PctFam2Par
Frequency
0
100
200
300
30
50
70
90
PctKids2Par
Frequency
0
50
100
150
200
250
300
30
60
90
PctYoungKids2Par
Frequency
0
100
200
300
40
80
PctTeen2Par
Frequency
0
100
200
300
400
30
60
90
PctWorkMomYoungKid
Frequency
0
100
200
300
400
500
20
50
80

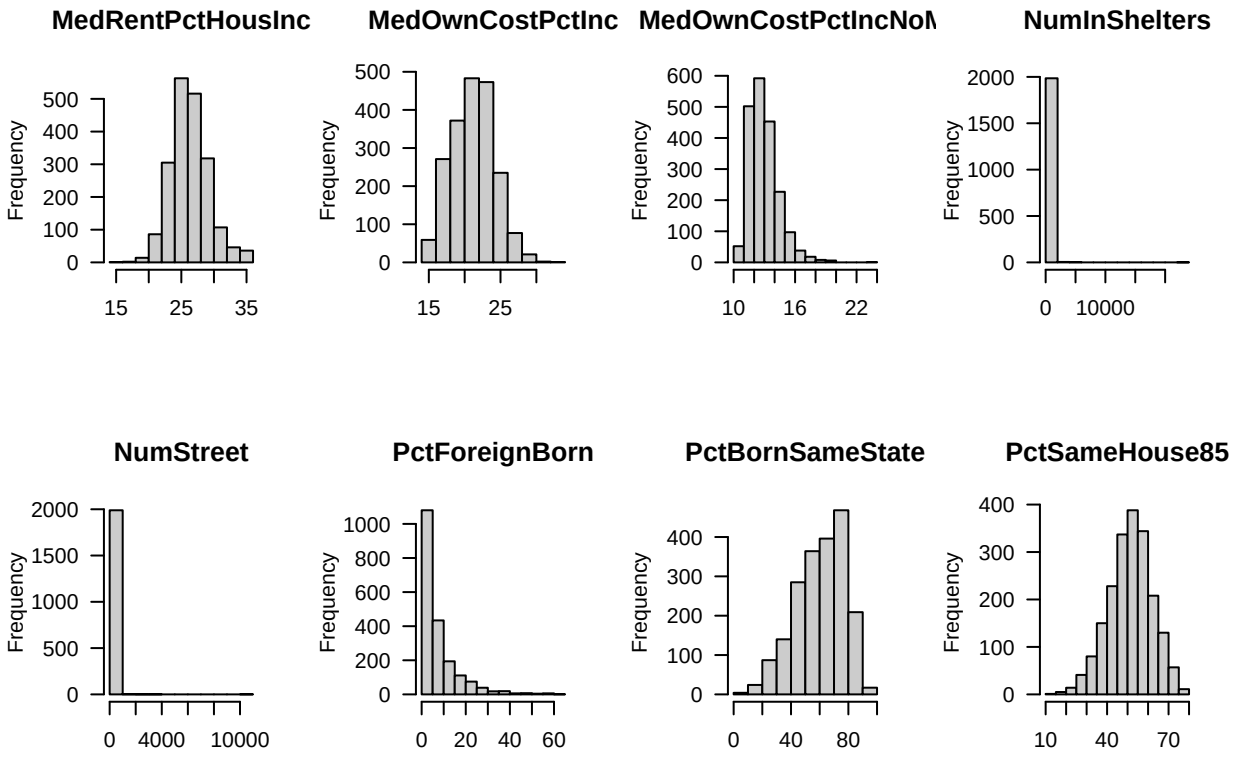

PctWorkMom
NumKidsBornNeverMa
PctKidsBornNeverMa
NumImmig
PctImmigRecent
PctImmigRec5
PctImmigRec8
PctImmigRec10
Frequency

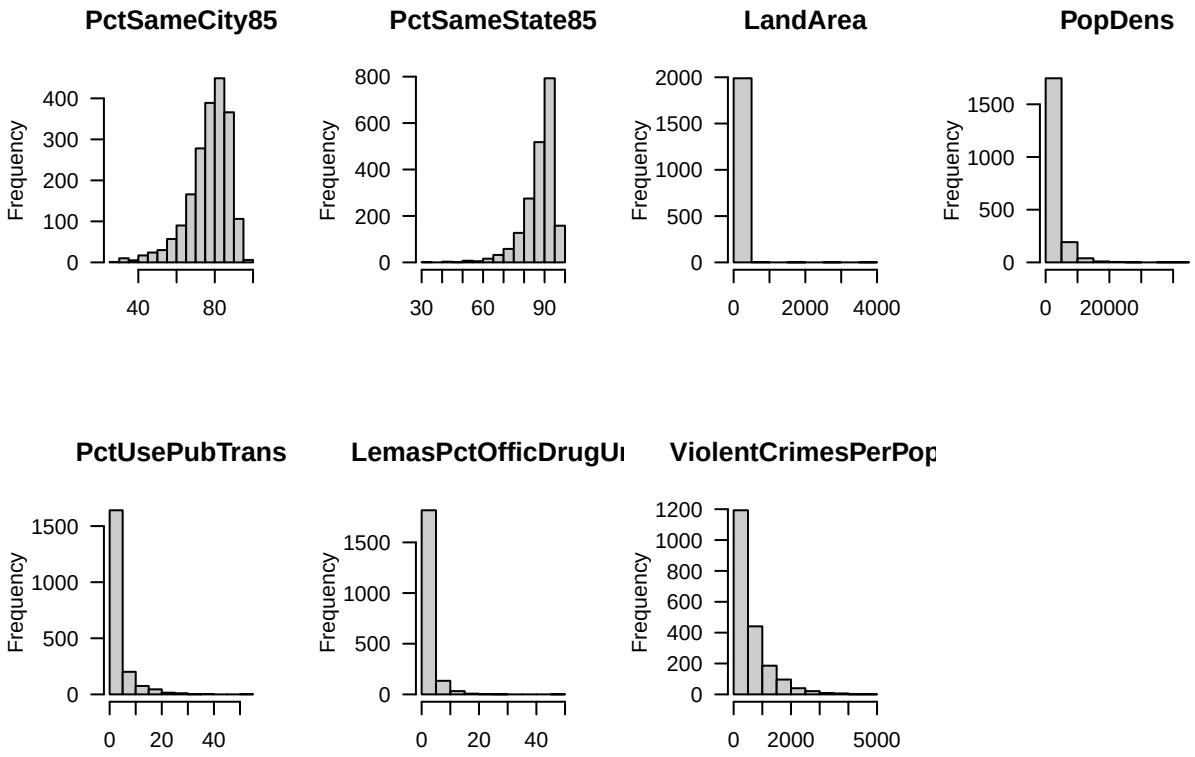

PctRecentImmig
PctRecImmig5
PctRecImmig8
PctRecImmig10
PctSpeakEnglOnly
PctNotSpeakEnglWel
PctLargHouseFam
PctLargHouseOccup
Frequency


PersPerOccupHous
PersPerOwnOccHous
PersPerRentOccHous
PctPersOwnOccup
PctPersDenseHous
PctHousLess3BR
MedNumBR
HousVacant
Frequency


PctHousOccup
PctHousOwnOcc
PctVacantBoarded
PctVacMore6Mos
MedYrHousBuilt
PctHousNoPhone
PctWOFullPlumb
OwnOccLowQuart
Frequency


OwnOccMedVal
OwnOccHiQuart
OwnOccQrange
RentLowQ
RentMedian
RentHighQ
RentQrange
MedRent
Frequency
0e+00
3e+05
0
150000
200
600
1000
400
800


MedRentPctHousInc
MedOwnCostPctInc
MedOwnCostPctIncNoM
NumInShelters
NumStreet
PctForeignBorn
PctBornSameState
PctSameHouse85
Frequency



We notice that many of the variables have a skewed distribution as depicted by the shapes of the histograms (this indicates that we might have to scale our data which will be addressed later). Having seen how each variable behaves on its own, we can even go one step further and examine how each variable behaves with respect to one another. Learning which variables are correlated can help us gain intuition behind why certain variables can be dropped. Below is a matrix of correlations (since the matrix is really large, we will use the `head` function to just view a small portion of it).

```
head(cor(cleaned_CC))
```

```
##                population householdsize racepctblack racePctWhite racePctAsian
## population     1.00000000   -0.01998124   0.11985409   -0.1691664   0.09020787
## householdsize -0.01998124    1.00000000  -0.05641415   -0.2427810   0.19010502
## racepctblack   0.11985409   -0.05641415   1.00000000   -0.8038940  -0.09006910
## racePctWhite  -0.16916638   -0.24278100  -0.80389404    1.0000000  -0.28996778
## racePctAsian   0.09020787    0.19010502  -0.09006910   -0.2899678   1.00000000
## racePctHisp    0.08698228    0.50089241  -0.07905903   -0.4141658   0.19388596
##                racePctHisp  agePct12t21 agePct12t29 agePct16t24  agePct65up
## population     0.08698228 -0.008922560  0.04603586  0.01748179 -0.04535452
## householdsize  0.50089241  0.486331416  0.37694662  0.30814265 -0.57727020
## racepctblack  -0.07905903  0.089593377  0.11905687  0.08754291  0.03501101
## racePctWhite  -0.41416581 -0.140683973 -0.21562068 -0.12389602  0.13680899
## racePctAsian   0.19388596 -0.008304097  0.07313797  0.03613463 -0.21793309
## racePctHisp    1.00000000  0.117924145  0.16616652  0.05601813 -0.20332624
##                 numbUrban     pctUrban   medIncome     pctWWage pctWFarmSelf
## population     0.99899475  0.114537414 -0.04902656 -0.007606969  -0.06468539
```

```
## householdsize -0.02052241 -0.014662170  0.18105895  0.417834151   0.16431834
## racepctblack   0.11906300  0.004526951 -0.34025272 -0.211859282  -0.14507967
## racePctWhite  -0.16957554 -0.054400107  0.29346975  0.108838173   0.09486210
## racePctAsian   0.09801341  0.227859902  0.31561770  0.259413552  -0.07956796
## racePctHisp    0.08736676  0.032882668 -0.15056995  0.009222161   0.07617366
##                pctWInvInc  pctWSocSec pctWPubAsst  pctWRetire   medFamInc
## population    -0.07920605 -0.05825732  0.10271673 -0.05905109 -0.05290946
## householdsize -0.16463318 -0.43464444  0.13502665 -0.31819863  0.09156280
## racepctblack  -0.48809256  0.10083477  0.43830333 -0.06974238 -0.34099531
## racePctWhite   0.59341547  0.07109588 -0.58551408  0.22263949  0.32767722
## racePctAsian   0.12784966 -0.31285796 -0.02913192 -0.15870111  0.29717196
## racePctHisp   -0.42104452 -0.13674881  0.44011865 -0.25614138 -0.21156285
##                 perCapInc whitePerCap blackPerCap indianPerCap AsianPerCap
## population    -0.02172376  0.03041797 -0.02529795 -0.009580455 -0.03695617
## householdsize -0.13058976 -0.12895857  0.03287737 -0.003111265 -0.05825344
## racepctblack  -0.26389697 -0.10109669 -0.19326730 -0.037701703 -0.10470790
## racePctWhite   0.28791549  0.11754179  0.12989914  0.033979863  0.13984378
## racePctAsian   0.25178076  0.28881002  0.18682018  0.060575837  0.05749987
## racePctHisp   -0.23610945 -0.20335442 -0.01488340 -0.009598256 -0.10456218
##               OtherPerCap  HispPerCap NumUnderPov PctPopUnderPov
## population             NA -0.04852552  0.98844390     0.08687736
## householdsize          NA -0.07778172 -0.01217924     0.07858127
## racepctblack           NA -0.13602581  0.15561213     0.47335079
## racePctWhite           NA  0.19643357 -0.19509753    -0.53178950
## racePctAsian           NA  0.15782486  0.05269314    -0.14309254
## racePctHisp            NA -0.21792719  0.10357355     0.34569067
##               PctLess9thGrade PctNotHSGrad  PctBSorMore PctUnemployed
## population          0.0341362   0.04520605 -0.003314509    0.08159569
## householdsize       0.2453181   0.13656123 -0.039715817    0.17072483
## racepctblack        0.2317974   0.34782645 -0.176438038    0.38426340
## racePctWhite       -0.4565570  -0.48549824  0.215510874   -0.51925170
## racePctAsian       -0.1116814  -0.18329153  0.256835320   -0.12349254
## racePctHisp         0.6399916   0.50212299 -0.258944239    0.46934622
##                 PctEmploy PctEmplManu PctEmplProfServ PctOccupManu
## population    -0.01436350 -0.05767265     0.013357012  -0.01558050
## householdsize  0.08192078  0.02998270    -0.071258377   0.08769569
## racepctblack  -0.26395386 -0.01813612     0.098149205   0.22408021
## racePctWhite   0.25107068  0.03526150     0.019681114  -0.26962446
## racePctAsian   0.19844967 -0.07100244     0.006181762  -0.22587980
## racePctHisp   -0.16450296 -0.03245830    -0.222755296   0.24086108
##               PctOccupMgmtProf MalePctDivorce MalePctNevMarr FemalePctDiv
## population        -0.006115214     0.09818571      0.1259982   0.12086191
## householdsize     -0.087931228    -0.43051333      0.1980953  -0.32918255
## racepctblack      -0.193809485     0.39593651      0.2633142   0.42418309
## racePctWhite       0.258503112    -0.33843081     -0.3475338  -0.44451726
## racePctAsian       0.229080506    -0.10700149      0.1657728  -0.02724117
## racePctHisp       -0.289367195     0.02119139      0.1504020   0.14757123
##               TotalPctDiv  PersPerFam PctFam2Par PctKids2Par PctYoungKids2Par
## population      0.1117944  0.05483077 -0.1436528  -0.1477879       -0.1176591
## householdsize  -0.3920254  0.83427003  0.2593835   0.1848487        0.1785928
## racepctblack    0.4273592  0.04352565 -0.6987393  -0.7308858       -0.6560017
## racePctWhite   -0.4057335 -0.39621600  0.6410727   0.7020485        0.6020396
## racePctAsian   -0.0706462  0.20304202  0.1264430   0.1056274        0.1603422
## racePctHisp     0.0858183  0.65260879 -0.1247147  -0.1933914       -0.1290443
```

```
##               PctTeen2Par PctWorkMomYoungKids  PctWorkMom NumKidsBornNeverMar
## population    -0.13880404        -0.048911713 -0.07832886          0.96919546
## householdsize  0.24793655        -0.225003314 -0.27454503         -0.01382065
## racepctblack  -0.68928536         0.172996713  0.08223719          0.18128400
## racePctWhite   0.61672564         0.006455434  0.15530871         -0.20145070
## racePctAsian   0.10197458        -0.107829302 -0.13979304          0.04170485
## racePctHisp   -0.08922259        -0.285707381 -0.41993518          0.06913143
##               PctKidsBornNeverMar    NumImmig PctImmigRecent PctImmigRec5
## population              0.1918587  0.93698544     0.05969798   0.07400148
## householdsize           0.0424901  0.02446960     0.04096913   0.06311299
## racepctblack            0.8045245  0.04345595     0.15684744   0.18712843
## racePctWhite           -0.7982256 -0.12413352    -0.21928174  -0.28559015
## racePctAsian           -0.0445678  0.11977194     0.14309005   0.17894882
## racePctHisp             0.2177592  0.12170301     0.06053260   0.11772738
##               PctImmigRec8 PctImmigRec10 PctRecentImmig PctRecImmig5
## population      0.09096036     0.1055186     0.14177309   0.14457513
## householdsize   0.07525378     0.1082819     0.29947542   0.32161070
## racepctblack    0.21907883     0.2431958    -0.04405507  -0.03533397
## racePctWhite   -0.34792615    -0.4104704    -0.39610968  -0.41945931
## racePctAsian    0.24065764     0.2711773     0.57871625   0.57841982
## racePctHisp     0.15336367     0.2252240     0.60912950   0.63959849
##               PctRecImmig8 PctRecImmig10 PctSpeakEnglOnly PctNotSpeakEnglWell
## population      0.15028179    0.15128376       -0.1120853           0.1169453
## householdsize   0.32768383    0.34005924       -0.4508026           0.4539802
## racepctblack   -0.03133008   -0.02679911        0.1010608          -0.0536535
## racePctWhite   -0.43596004   -0.44658474        0.4046849          -0.4338322
## racePctAsian    0.61794808    0.61283928       -0.4069128           0.3197273
## racePctHisp     0.64653593    0.66857076       -0.9142703           0.8923207
##               PctLargHouseFam PctLargHouseOccup PersPerOccupHous
## population          0.1007336        0.06256921      -0.01239029
## householdsize       0.6842431        0.74163139       0.87884973
## racepctblack        0.1454162        0.07447286      -0.10083585
## racePctWhite       -0.5431399       -0.46718488      -0.22907158
## racePctAsian        0.2181128        0.20671411       0.21429846
## racePctHisp         0.7590884        0.73735224       0.55762262
##               PersPerOwnOccHous PersPerRentOccHous PctPersOwnOccup
## population          0.006475298         0.02142278     -0.15269723
## householdsize       0.812931820         0.73248303      0.10591993
## racepctblack       -0.149839069         0.09853665     -0.36660021
## racePctWhite       -0.134574764        -0.46963328      0.50394418
## racePctAsian        0.198028064         0.23973473     -0.09457199
## racePctHisp         0.472150928         0.66649194     -0.30369714
##               PctPersDenseHous PctHousLess3BR    MedNumBR  HousVacant
## population           0.1146375     0.13519473 -0.11160762  0.92506684
## householdsize        0.5498120    -0.29198804  0.18800102 -0.09101512
## racepctblack         0.1121861     0.22738771 -0.13896697  0.18344377
## racePctWhite        -0.5924452    -0.36473283  0.28306768 -0.20031790
## racePctAsian         0.2985607     0.04941892 -0.07194458  0.03235154
## racePctHisp          0.8643040     0.32426794 -0.30307948  0.07268417
##               PctHousOccup PctHousOwnOcc PctVacantBoarded PctVacMore6Mos
## population     -0.02107012   -0.15552193       0.14496181   -0.032799859
## householdsize   0.21639449    0.16091300       0.05906204   -0.035390554
## racepctblack   -0.18598389   -0.33264409       0.47508949    0.165820489
## racePctWhite    0.12343669    0.43829466      -0.44598166    0.005199846
```

```
## racePctAsian    0.18929321   -0.07703753      -0.11257058   -0.329860451
## racePctHisp    -0.05624573   -0.24200338       0.15654664   -0.139280497
##                MedYrHousBuilt PctHousNoPhone PctWOFullPlumb OwnOccLowQuart
## population       -0.05245494     0.03540845     0.07661476     0.01309797
## householdsize     0.23576331     0.01158037     0.17729330     0.10948380
## racepctblack     -0.08891917     0.46355039     0.25372037    -0.26722203
## racePctWhite      0.02160577    -0.47222110    -0.37774264     0.08656218
## racePctAsian      0.06365386    -0.25752141    -0.05578593     0.51168731
## racePctHisp       0.08225862     0.29802016     0.42852090     0.04405002
##                OwnOccMedVal OwnOccHiQuart OwnOccQrange      RentLowQ
## population       0.02186620    0.03763290  0.073552241 -0.0005833201
## householdsize    0.09842372    0.08492910  0.027318381  0.1372430501
## racepctblack    -0.25182349   -0.23572706 -0.139428213 -0.3078488853
## racePctWhite     0.07195308    0.06216770  0.008856967  0.1228098864
## racePctAsian     0.50903905    0.50198842  0.396457596  0.5055629358
## racePctHisp      0.04921780    0.04620829  0.041787865  0.0371101404
##                 RentMedian   RentHighQ  RentQrange      MedRent
## population     0.001442043  0.01469968  0.03545392 -0.007353084
## householdsize  0.153729058  0.16773905  0.16168116  0.165025196
## racepctblack  -0.293102246 -0.28356527 -0.14531907 -0.269213448
## racePctWhite   0.123818603  0.11901415  0.07178750  0.123516168
## racePctAsian   0.480286908  0.46678128  0.24122397  0.448396868
## racePctHisp    0.026300909  0.03252414  0.01362826  0.001199020
##                MedRentPctHousInc MedOwnCostPctInc MedOwnCostPctIncNoMtg
## population            0.06105356       0.04605370          -0.007638429
## householdsize         0.13527095       0.24506090          -0.100861914
## racepctblack          0.19330933      -0.06033315           0.217050630
## racePctWhite         -0.33748613      -0.17496918          -0.065463790
## racePctAsian          0.15282841       0.34479739          -0.204903686
## racePctHisp           0.26945343       0.31486880          -0.095597760
##                NumInShelters   NumStreet PctForeignBorn PctBornSameState
## population        0.93032799  0.92103806     0.13311280      -0.06776498
## householdsize    -0.03336532 -0.01217337     0.33805885      -0.06011572
## racepctblack      0.11256032  0.06126485    -0.09464446       0.09589610
## racePctWhite     -0.14207182 -0.09979246    -0.37735606       0.11169349
## racePctAsian      0.06594532  0.06870114     0.59404355      -0.34866914
## racePctHisp       0.05269424  0.05869189     0.69121593      -0.23186484
##                PctSameHouse85 PctSameCity85 PctSameState85      LandArea
## population        -0.03340836    0.01686028  -0.0281548497  0.1963973642
## householdsize     -0.07219896   -0.12507445  -0.0497000116 -0.0049557789
## racepctblack      -0.03431448    0.06374576  -0.0006933544  0.0403043730
## racePctWhite       0.15740343   -0.02602928   0.0328709611 -0.0457945937
## racePctAsian      -0.14974903   -0.14166234  -0.1480290209  0.0005761101
## racePctHisp       -0.12105218    0.04648453   0.0163896829 -0.0052344098
##                    PopDens PctUsePubTrans LemasPctOfficDrugUn
## population     0.21394125     0.31966609          0.20139178
## householdsize  0.03158455    -0.05373092         -0.08377101
## racepctblack   0.09431397     0.17293606          0.24596833
## racePctWhite  -0.32032835    -0.22887554         -0.25760969
## racePctAsian   0.29643645     0.22793683          0.06364783
## racePctHisp    0.36858895     0.08412792          0.07654516
##                ViolentCrimesPerPop
## population              0.21235381
## householdsize          -0.02011018
```

```
## racepctblack           0.62836776
## racePctWhite          -0.67684882
## racePctAsian           0.03194948
## racePctHisp            0.25359641
```

Regression Task

Our goal is to develop a model to predict "ViolentCrimesPerPop"

Since there are so many variables in our dataset, we hope to find an appropriate model using dimension reduction methods or shrinkage methods so that we can predict our desired response in a "friendly" way. In particular, we will use methods such as PC regression, PLS regression, Ridge regression, and Lasso regression. Addtionally, we will use cross-validation to select tuning parameters and utilize the three-way holdout method to perform model selection and model assessment.

We begin by defining our response vector and feature matrix. Also, recall that there is one column ("OtherPerCap") which contains a single missing value of which we chose to keep. To deal with this missing value, we will impute the missing value with its respective column mean.

```
#response vector
y <- cleaned_CC$ViolentCrimesPerPop
response <- "y"

#feature matrix with replaced missing value
X <- subset(cleaned_CC, select = -c(ViolentCrimesPerPop))
X$OtherPerCap[is.na(X$OtherPerCap)] <- mean(X$OtherPerCap, na.rm = TRUE)
predictors <- names(X)

#combine feature matrix and response vector for convenience
dat <- as.data.frame(cbind(X,y))
```

Three-way hold-out method

Before we begin implementing any of the regularization methods, we first split the data into three different parts:
1. Training set: 60% of the data (chosen at random)
2. Validation set: 20% of the data (i.e. one half of the remaining 40% not in training, chosen at random)
3. Test set: 20% of the data (i.e. the other half of the remaining 40% not in training, chosen at random)

```
set.seed(1)
train_prop <- 0.6
total <- nrow(dat)

#training set
train <- sample(1:total, floor(total*train_prop))
testval <- c(1:total)[-train]

#validation set
validation <- sample(testval, size = floor(length(testval)/2))

#test set
test <- c(1:total)[-c(train, validation)]
```

PCA

After successfully prepping our data, we standardize it in order to ensure that the significance of each variable is properly captured relative to one another. The first thing that we would like to do is attempt to reduce the dimensionality by performing PCA.

```
#standardize feature matrix
X <- scale(X)

#compute variance-covariance matrix
n <- nrow(X)
S <- (1/(n-1)) * t(as.matrix(X)) %*% as.matrix(X)

#perform eigenvalue decomposition
EVD <- eigen(S)

#eigenvalues of S (the variances of each PC)
evalues <- EVD$values
head(evalues)
```

```
## [1] 24.343530 16.298249  9.134802  7.922403  6.406749  4.377326
```

```
#eigenvectors of S (loadings)
evectors <- EVD$vectors
head(evectors)
```

```
##              [,1]         [,2]        [,3]         [,4]        [,5]
## [1,] -0.026577636  0.065099858  0.14579982  0.184552471  0.25791833
## [2,]  0.004568478  0.131887283 -0.19031340 -0.083595496  0.14163395
## [3,] -0.102306123  0.006402538  0.08253572  0.028133461 -0.02983111
## [4,]  0.109552493 -0.123011775 -0.02356669 -0.024602880  0.04069955
## [5,]  0.047371094  0.144805274  0.03377442 -0.009803065 -0.04907106
## [6,] -0.066472116  0.172216454 -0.12716760  0.022964970 -0.01394782
##              [,6]        [,7]         [,8]         [,9]       [,10]       [,11]
## [1,]  0.017920849  0.04300980 -0.007308468  0.009270909 -0.01752143  0.01658662
## [2,] -0.092275175 -0.06945836 -0.066195076 -0.077128587 -0.04949304  0.08685429
## [3,] -0.033944385 -0.33980979 -0.072452598  0.005294221 -0.08859645  0.13479344
## [4,]  0.002376779  0.24042461  0.079127005 -0.037051010  0.04986506 -0.11844207
## [5,]  0.049042273  0.02310039  0.041174163  0.056528524  0.01405839  0.14956301
## [6,]  0.016825353  0.11177358 -0.065674167  0.017439324  0.07143277 -0.12478334
##             [,12]       [,13]       [,14]       [,15]       [,16]        [,17]
## [1,]  0.02417096 -0.00448563 -0.02054244 -0.01101617  0.01064100 -0.010208790
## [2,]  0.05320303 -0.01817154  0.04133420 -0.02792225 -0.05029930 -0.005984993
## [3,] -0.11358764 -0.11989426  0.21991616 -0.03876096 -0.05829349 -0.014462900
## [4,]  0.06307187  0.18213135 -0.14092898  0.02204136  0.05965231 -0.011517265
## [5,]  0.12567539 -0.16057485 -0.03681884  0.01337671 -0.12607560  0.139261012
## [6,] -0.05238597 -0.08298295 -0.07303860  0.00633800  0.10536800 -0.064508144
##              [,18]        [,19]         [,20]       [,21]        [,22]
## [1,]  0.012335047  0.008066279 -0.0006572833  0.01659684 -0.001498696
## [2,]  0.005616012  0.024631501 -0.0098504806 -0.08147366  0.059616601
## [3,] -0.021166101 -0.051420843  0.0044448381  0.03361956 -0.082371854
## [4,]  0.094606541  0.028545366  0.0256470516 -0.10357670  0.057285983
## [5,] -0.368869950  0.136818197 -0.1125299892  0.40007712 -0.034907653
## [6,]  0.103539563 -0.078308040 -0.0072723209 -0.04762336  0.055394553
```

```
##             [,23]        [,24]        [,25]        [,26]        [,27]
## [1,] -0.004304147  0.009933551  0.016030376  0.004640294 -0.009244075
## [2,] -0.042951091 -0.052796559 -0.077902485  0.032137798  0.004682224
## [3,]  0.177470262 -0.021563738  0.032877321  0.017927307 -0.066890249
## [4,] -0.173724157  0.027148074 -0.018740252 -0.026497756  0.099104044
## [5,]  0.096386783  0.027101988 -0.135746170 -0.117658885 -0.242247503
## [6,] -0.035728242  0.071051067 -0.002599899  0.016553087  0.140606586
##             [,28]       [,29]        [,30]       [,31]        [,32]       [,33]
## [1,]  0.01224319 -0.02381306  0.004598176 -0.02509612 -0.005139776 -0.01579424
## [2,] -0.03292691 -0.05412032  0.054488233  0.03172119  0.097821856 -0.02885663
## [3,] -0.08472306  0.02996253  0.035017291 -0.01015335 -0.004234876  0.19607923
## [4,]  0.10006823  0.04050458 -0.063466950 -0.02164266  0.044015928 -0.11529137
## [5,] -0.18529955 -0.03588542 -0.095768115  0.02631169 -0.011947679 -0.35861926
## [6,]  0.03135547 -0.15842536  0.132196520  0.06262484 -0.112271073  0.02118127
##             [,34]        [,35]        [,36]       [,37]       [,38]        [,39]
## [1,]  0.002647875 -0.007538536 -0.008024600 -0.01472617  0.01532305 -0.03279090
## [2,]  0.089952283  0.014390879 -0.050565445  0.07829235 -0.01354886  0.10814440
## [3,]  0.078259478 -0.116064876  0.053349695  0.06356608  0.11547131 -0.17954767
## [4,] -0.053360982  0.019965303  0.017147220 -0.03558513 -0.15230900  0.18953870
## [5,] -0.022223962  0.094632339  0.002748044 -0.09177312 -0.08614383 -0.08123203
## [6,] -0.009500068  0.121085398 -0.094885050  0.02055191  0.15719307 -0.10550619
##             [,40]       [,41]       [,42]       [,43]       [,44]       [,45]
## [1,]  0.03938087  0.01414469 -0.05741996  0.08223016  0.00287909  0.03512732
## [2,] -0.11904009  0.13202800 -0.05745720  0.12837136 -0.08353355 -0.15623845
## [3,]  0.01356902 -0.15992657 -0.10272887  0.06854872 -0.13588351 -0.09664213
## [4,] -0.13498945  0.05555136  0.02599796 -0.07538281  0.05756422  0.21995838
## [5,]  0.10382761  0.13636457  0.12889909  0.01332579  0.11072867 -0.10722184
## [6,]  0.18613851  0.17453111  0.11316429  0.04719531  0.09406834 -0.14040867
##             [,46]        [,47]       [,48]        [,49]       [,50]       [,51]
## [1,] -0.09208309 -0.008014516 -0.02614619 -0.023247759 -0.01595838 -0.01391626
## [2,]  0.02254294 -0.167543636 -0.12697078  0.016487177 -0.21233950  0.17841960
## [3,]  0.06009740  0.059699318  0.05337934  0.004733562 -0.05733635  0.04460733
## [4,] -0.01140995 -0.040932350 -0.17112909  0.045943860 -0.02616258 -0.01103662
## [5,] -0.04427226 -0.074888913  0.03545273  0.101399918  0.05776887 -0.15577436
## [6,]  0.01172869  0.139841036  0.15907314 -0.238334209 -0.04606000  0.17378738
##             [,52]        [,53]       [,54]       [,55]        [,56]       [,57]
## [1,] -0.08546778  0.055491757  0.05098578 -0.04218347 -0.016462811  0.00358729
## [2,]  0.30988896  0.002275641  0.16817313 -0.16816489  0.171070473  0.04861430
## [3,]  0.03859940  0.162938780 -0.06028919  0.07356701  0.022912234  0.01708291
## [4,] -0.03853784 -0.051991648 -0.02661532 -0.05800092 -0.107937072 -0.08145447
## [5,] -0.02013483 -0.053583447  0.04106358 -0.07741997 -0.022233053 -0.05592863
## [6,] -0.05144988 -0.046973997  0.03194063  0.06019650 -0.004860287 -0.08282579
##             [,58]        [,59]        [,60]       [,61]       [,62]       [,63]
## [1,] -0.00598417  0.033847177 -0.002253565 -0.01918254 -0.01942746  0.01205204
## [2,] -0.05948579  0.228635163 -0.044489180  0.09824357  0.08518101  0.13713744
## [3,] -0.07465325 -0.075434074 -0.073696451  0.01171874 -0.14042785 -0.05718693
## [4,]  0.00302883 -0.009003527 -0.044732715  0.04357659  0.16870732 -0.09845800
## [5,] -0.02813981  0.031045785  0.094647529 -0.02304627  0.11724966 -0.02220296
## [6,] -0.08541486  0.037179623  0.033837258 -0.04670896  0.01659687  0.01312556
##              [,64]       [,65]        [,66]        [,67]       [,68]
## [1,]  0.0193433090 -0.01007596 -0.008759277  0.036588780 -0.03507907
## [2,]  0.0008658202 -0.10301268  0.100263015 -0.001539129  0.02960534
## [3,] -0.1362979196 -0.18156326  0.074904983 -0.067149629  0.02149141
## [4,] -0.1507786298 -0.10370603  0.069060940 -0.148617185 -0.04340101
```

```
## [5,]  0.0264535029  0.01535314 -0.020576799 -0.021054941 -0.01480823
## [6,]  0.0621279896  0.10933976 -0.210319544  0.255603187 -0.03139649
##            [,69]         [,70]       [,71]       [,72]        [,73]
## [1,]  0.01681108 -0.0031317617 -0.01799005  0.03155739  0.061873322
## [2,] -0.07585506  0.0530965236 -0.06316023  0.03123251 -0.013741543
## [3,]  0.02782936 -0.0002594133  0.02629149  0.01546533 -0.036253282
## [4,]  0.08266508  0.0126065223 -0.01669295 -0.02517491 -0.009506757
## [5,] -0.04706139 -0.0206470179  0.02225068 -0.02188407 -0.056592960
## [6,]  0.05698817 -0.0319708319 -0.01593871  0.01643605 -0.028572526
##             [,74]        [,75]       [,76]        [,77]       [,78]
## [1,] -0.071716716  0.040418870 -0.07116238 -0.010167522 -0.10881923
## [2,] -0.189098138 -0.006858964  0.21504799 -0.389885210  0.05285137
## [3,] -0.053522116 -0.009368924 -0.09265473  0.043871395 -0.13967937
## [4,] -0.005148836  0.054709405  0.01100692  0.005541169 -0.12136304
## [5,] -0.053152678 -0.061828640  0.04519890  0.009302802 -0.05326495
## [6,] -0.062233447  0.069696700  0.12540680  0.046476732  0.03960356
##             [,79]        [,80]        [,81]       [,82]        [,83]
## [1,]  0.101997711 -0.072849287 -0.028313428  0.46409781 -0.009196756
## [2,]  0.001181653  0.008829965 -0.047619067  0.01364931  0.013420261
## [3,]  0.090392083 -0.227656660 -0.008063569 -0.19458386 -0.186374083
## [4,]  0.040586701 -0.321383915 -0.026871904 -0.18564652 -0.197329649
## [5,]  0.013096439 -0.070204488  0.031742092 -0.07778473 -0.052279059
## [6,] -0.132901885 -0.060663887  0.337004526 -0.06214878 -0.178214854
##            [,84]       [,85]      [,86]        [,87]      [,88]        [,89]
## [1,] -0.08265288 -0.01570085 0.01060036  0.004893220 0.01182844  0.194270122
## [2,] -0.06020545  0.08419881 0.05414943  0.180025100 0.03906225 -0.008069885
## [3,] -0.41340174 -0.01049703 0.08694862 -0.016737024 0.04628290  0.052443598
## [4,] -0.42613970 -0.04718964 0.11504841  0.032455641 0.02060317  0.041509420
## [5,] -0.14059341  0.03095404 0.10808799 -0.004715453 0.01164414  0.025370632
## [6,] -0.20301483  0.11493089 0.20617902 -0.049874803 0.01360616  0.062448865
##              [,90]       [,91]        [,92]        [,93]        [,94]
## [1,]  0.0951919643 0.010476679  0.020437789 -0.007806031  0.009477949
## [2,] -0.0358581498 0.006957098  0.020341569 -0.010408041 -0.001017762
## [3,] -0.0005616237 0.057065263 -0.014759082 -0.032051504 -0.040322506
## [4,]  0.0559086203 0.057536178 -0.032966279 -0.068948886 -0.031878264
## [5,] -0.0124768349 0.016957818 -0.009194687 -0.013737141 -0.040722084
## [6,] -0.0282286047 0.058613320 -0.003037201 -0.034661777 -0.011213528
##             [,95]        [,96]         [,97]         [,98]        [,99]
## [1,] -0.006723752  0.007160879  0.0086604257  5.342341e-03  0.058083089
## [2,]  0.011061589  0.012545210 -0.0055693641 -1.421555e-02 -0.015048108
## [3,] -0.019926131 -0.020567838 -0.0133201903 -2.131431e-03  0.004255648
## [4,] -0.044595208 -0.018484079 -0.0131029244 -7.091007e-03  0.002633125
## [5,] -0.016973724 -0.022581098  0.0072926885 -3.685843e-03 -0.002139027
## [6,] -0.004650588 -0.005709771 -0.0002122037  2.573254e-06 -0.004727481
##           [,100]        [,101]       [,102]
## [1,] 0.708202898 -1.936865e-13 0.000000e+00
## [2,] 0.004830813  9.010694e-14 2.554240e-14
## [3,] 0.002653897 -2.256466e-14 5.473834e-14
## [4,] 0.003067335 -8.354587e-16 3.844750e-14
## [5,] 0.001612752  1.971652e-14 1.877639e-14
## [6,] 0.002455213  2.709642e-14 9.350326e-15
```

```
#principal components (each PC is a linear combination of each of the original features)
pcs <- X %*% evectors
```

```r
colnames(pcs) <-paste0('PC', 1:ncol(pcs))
head(pcs[,1:4])
```

```
##         PC1        PC2        PC3       PC4
## 1 11.137044  2.3507545 -1.7893031  1.807111
## 2  6.343140 -1.6726233 -1.8967130  2.255265
## 3  2.555256 -1.2286008  1.8883608 -2.155570
## 4 -4.796155 -4.3618166 -0.8038962  2.299482
## 5 -2.752111 -2.4552902  2.5989538 -2.062738
## 6  4.699144 -0.0739867  0.2789733  1.227586
```

```r
#high level summary of doing PCA
pc <- princomp(X, scores = TRUE)
summary(pc)
```

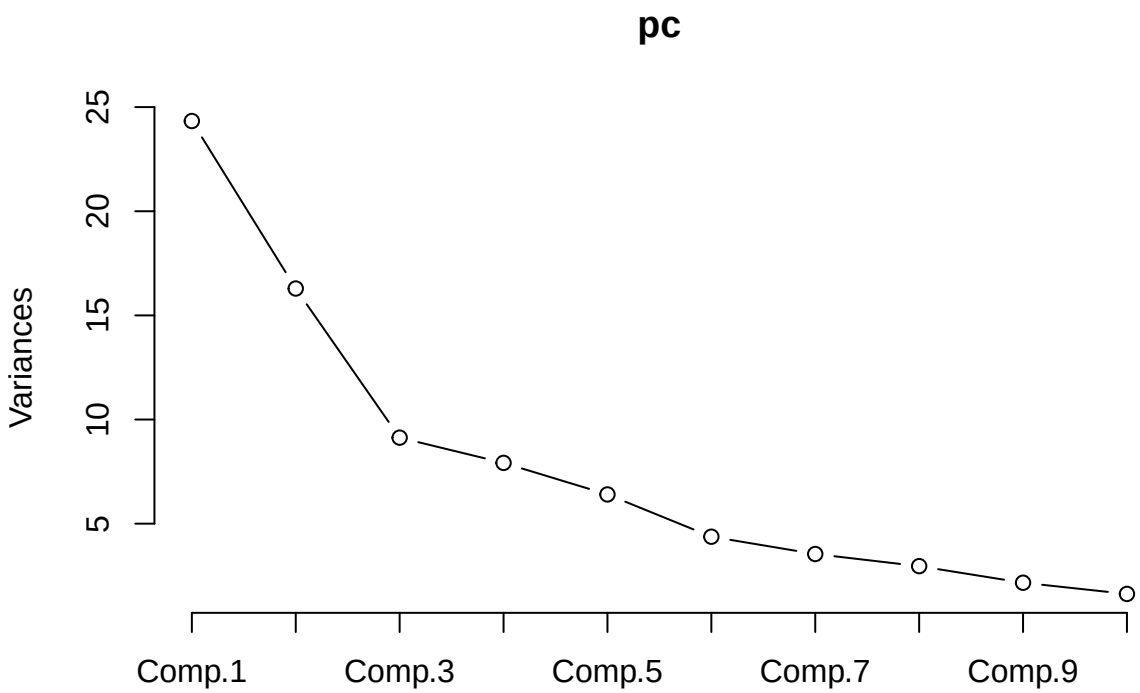
```
## Importance of components:
##                            Comp.1    Comp.2     Comp.3     Comp.4     Comp.5
## Standard deviation     4.9326789 4.0360965 3.02162558 2.81397046 2.53052082
## Proportion of Variance 0.2386621 0.1597868 0.08955688 0.07767062 0.06281126
## Cumulative Proportion  0.2386621 0.3984488 0.48800569 0.56567631 0.62848757
##                            Comp.6    Comp.7     Comp.8     Comp.9    Comp.10
## Standard deviation     2.09168135 1.8823422 1.72071511 1.47110777 1.27572454
## Proportion of Variance 0.04291496 0.0347548 0.02904261 0.02122788 0.01596362
## Cumulative Proportion  0.67140253 0.7061573 0.73519995 0.75642783 0.77239145
##                           Comp.11    Comp.12    Comp.13    Comp.14    Comp.15
## Standard deviation     1.26639012 1.21586732 1.20654739 1.04850815 1.01969247
## Proportion of Variance 0.01573087 0.01450074 0.01427928 0.01078354 0.01019897
## Cumulative Proportion  0.78812232 0.80262306 0.81690234 0.82768588 0.83788485
##                            Comp.16     Comp.17     Comp.18     Comp.19
## Standard deviation     0.989307306 0.976271643 0.937987851 0.914970955
## Proportion of Variance 0.009600196 0.009348868 0.008630026 0.008211685
## Cumulative Proportion  0.847485042 0.856833910 0.865463936 0.873675622
##                            Comp.20     Comp.21     Comp.22     Comp.23
## Standard deviation     0.875911957 0.865512451 0.850786030 0.780493078
## Proportion of Variance 0.007525556 0.007347918 0.007100001 0.005975246
## Cumulative Proportion  0.881201178 0.888549096 0.895649097 0.901624343
##                            Comp.24     Comp.25     Comp.26    Comp.27
## Standard deviation     0.769692094 0.728701908 0.708472385 0.70300746
## Proportion of Variance 0.005811012 0.005208558 0.004923382 0.00484772
## Cumulative Proportion  0.907435354 0.912643912 0.917567294 0.92241501
##                            Comp.28     Comp.29    Comp.30    Comp.31
## Standard deviation     0.686829233 0.668760688 0.65251133 0.62607171
## Proportion of Variance 0.004627168 0.004386914 0.00417632 0.00384473
## Cumulative Proportion  0.927042182 0.931429096 0.93560542 0.93945015
##                            Comp.32     Comp.33     Comp.34     Comp.35
## Standard deviation     0.614053732 0.603214708 0.579554833 0.565048753
## Proportion of Variance 0.003698541 0.003569123 0.003294631 0.003131768
## Cumulative Proportion  0.943148688 0.946717811 0.950012441 0.953144209
##                            Comp.36     Comp.37     Comp.38    Comp.39   Comp.40
## Standard deviation     0.538861573 0.520473095 0.508690135 0.50557907 0.4844127
## Proportion of Variance 0.002848211 0.002657139 0.002538191 0.00250724 0.0023017
## Cumulative Proportion  0.955992420 0.958649559 0.961187750 0.96369499 0.9659967
```

```
##                            Comp.41     Comp.42    Comp.43     Comp.44
## Standard deviation     0.481134399 0.465598776 0.44850228 0.435680860
## Proportion of Variance 0.002270652 0.002126382 0.00197309 0.001861893
## Cumulative Proportion  0.968267341 0.970393723 0.97236681 0.974228707
##                            Comp.45     Comp.46     Comp.47    Comp.48
## Standard deviation     0.42850672 0.420775769 0.397115542 0.38737787
## Proportion of Variance 0.00180108 0.001736677 0.001546862 0.00147193
## Cumulative Proportion  0.97602979 0.977766464 0.979313325 0.98078526
##                            Comp.49     Comp.50    Comp.51     Comp.52
## Standard deviation     0.364008940 0.352446149 0.34947045 0.322576014
## Proportion of Variance 0.001299696 0.001218437 0.00119795 0.001020662
## Cumulative Proportion  0.982084952 0.983303389 0.98450134 0.985522001
##                            Comp.53      Comp.54      Comp.55      Comp.56
## Standard deviation     0.321055708 0.3142163085 0.3052621659 0.2853168174
## Proportion of Variance 0.001011064 0.0009684454 0.0009140367 0.0007984954
## Cumulative Proportion  0.986533064 0.9875015095 0.9884155462 0.9892140416
##                             Comp.57      Comp.58      Comp.59      Comp.60
## Standard deviation     0.2737585853 0.2690977660 0.2599109128 0.2561377774
## Proportion of Variance 0.0007351114 0.0007102935 0.0006626233 0.0006435243
## Cumulative Proportion  0.9899491531 0.9906594466 0.9913220699 0.9919655942
##                             Comp.61      Comp.62      Comp.63      Comp.64
## Standard deviation     0.2529292131 0.2397154092 0.2362733634 0.2248870324
## Proportion of Variance 0.0006275028 0.0005636501 0.0005475795 0.0004960741
## Cumulative Proportion  0.9925930970 0.9931567471 0.9937043267 0.9942004007
##                             Comp.65      Comp.66      Comp.67      Comp.68
## Standard deviation     0.2173380176 0.2080171493 0.1986859627 0.1951771870
## Proportion of Variance 0.0004633286 0.0004244397 0.0003872149 0.0003736593
## Cumulative Proportion  0.9946637293 0.9950881690 0.9954753839 0.9958490432
##                             Comp.69      Comp.70      Comp.71      Comp.72
## Standard deviation     0.1856936330 0.1800098925 0.1784278362 0.1727328029
## Proportion of Variance 0.0003382297 0.0003178414 0.0003122791 0.0002926627
## Cumulative Proportion  0.9961872728 0.9965051142 0.9968173933 0.9971100560
##                             Comp.73      Comp.74      Comp.75      Comp.76
## Standard deviation     0.1659213685 0.1619551449 0.1518608298 0.1510578655
## Proportion of Variance 0.0002700364 0.0002572807 0.0002262087 0.0002238228
## Cumulative Proportion  0.9973800924 0.9976373731 0.9978635817 0.9980874045
##                             Comp.77      Comp.78      Comp.79      Comp.80
## Standard deviation     0.1443429543 0.1389526415 0.1326806473 0.1262231991
## Proportion of Variance 0.0002043661 0.0001893875 0.0001726763 0.0001562774
## Cumulative Proportion  0.9982917706 0.9984811581 0.9986538345 0.9988101118
##                             Comp.81      Comp.82      Comp.83      Comp.84
## Standard deviation     0.1235752103 0.1212765575 0.1170376777 0.1110082965
## Proportion of Variance 0.0001497892 0.0001442685 0.0001343597 0.0001208728
## Cumulative Proportion  0.9989599010 0.9991041695 0.9992385292 0.9993594020
##                             Comp.85      Comp.86      Comp.87      Comp.88
## Standard deviation     0.1057593846 1.009005e-01 8.309721e-02 7.888263e-02
## Proportion of Variance 0.0001097123 9.986286e-05 6.773148e-05 6.103522e-05
## Cumulative Proportion  0.9994691143 9.995690e-01 9.996367e-01 9.996977e-01
##                             Comp.89      Comp.90      Comp.91      Comp.92
## Standard deviation     0.0745019065 7.400372e-02 6.520519e-02 6.049967e-02
## Proportion of Variance 0.0000544443 5.371861e-05 4.170442e-05 3.590241e-05
## Cumulative Proportion  0.9997521882 9.998059e-01 9.998476e-01 9.998835e-01
##                             Comp.93      Comp.94      Comp.95      Comp.96
## Standard deviation     5.818417e-02 4.970809e-02 4.677072e-02 4.339695e-02
```

```
## Proportion of Variance 3.320683e-05 2.423661e-05 2.145684e-05 1.847294e-05
## Cumulative Proportion  9.999167e-01 9.999410e-01 9.999624e-01 9.999809e-01
##                            Comp.97      Comp.98      Comp.99     Comp.100
## Standard deviation     3.192576e-02 2.233391e-02 1.563978e-02 1.363459e-02
## Proportion of Variance 9.997699e-06 4.892685e-06 2.399269e-06 1.823483e-06
## Cumulative Proportion  9.999909e-01 9.999958e-01 9.999982e-01 1.000000e+00
##                           Comp.101     Comp.102
## Standard deviation     3.670186e-08 1.477783e-08
## Proportion of Variance 1.321277e-17 2.142097e-18
## Cumulative Proportion  1.000000e+00 1.000000e+00
```

We can summarize our findings visually with a plot which will give us a sense of what PCs are important and the amount of variance each captures.

```
plot(pc, type = "l")
```



The plot indicates that we do not need to use all of the PCs to get good results since the first couple of PCs already capture most of the variance of our original data. In fact, if we look at the cumulative proportions, we notice that the first 17 PCs already capture roughly 85% of the total variance and the first 34 captures about 95%! This is a big step forward since it shows a significant drop in the number of PCs we would need (total of 102 PCs), reducing dimensionality while still capturing as much variability of the original data as possible. We can build our model using just a few PCs and still get good results; this ultimately sets the stage for dimension reduction techniques like PCR as well as PLSR and gives us a reason as to why we should utilize them.

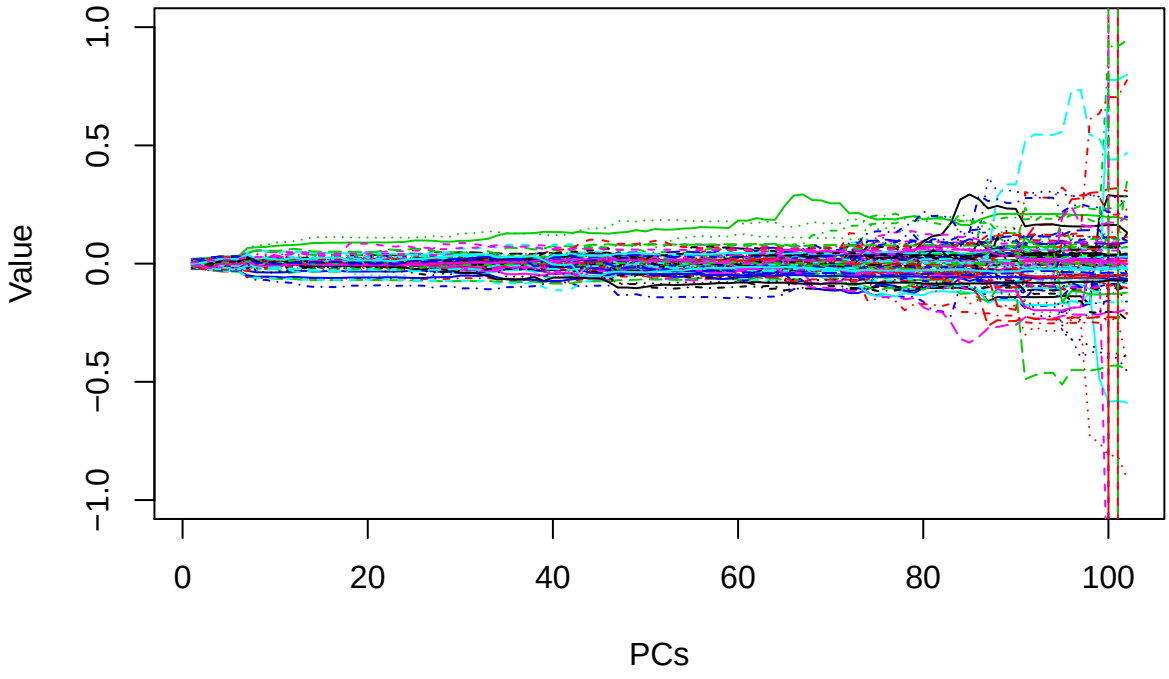
Principal Component Regression

In the first method, we will use Principal Component Regression (PCR). The idea is that we can fit a least squares model on a small number of PCs and obtain coefficients that give us a good model to use in predicting our response. We use the training set to fit PCR. The pcr function returns the regression coefficients in terms of the X variables, which simplifies interpretation. We can then visualize how the coefficients grow in terms of the number of retained PCs.

```
formula <- y ~.
Xtrain <- scale(dat[train, c(predictors, response)])
Xtrain <- as.data.frame(Xtrain)

#PCR on training data
pcr_fit <- pcr(formula, data = Xtrain, scale = FALSE, ncomp = ncol(X))

PCR <- apply(pcr_fit$coefficients, MARGIN = 3, function(u) u)
matplot(t(PCR), type = "l", xlim = c(1, 102), ylim = c(-1, 1), ylab = "Value", xlab = "PCs")
```



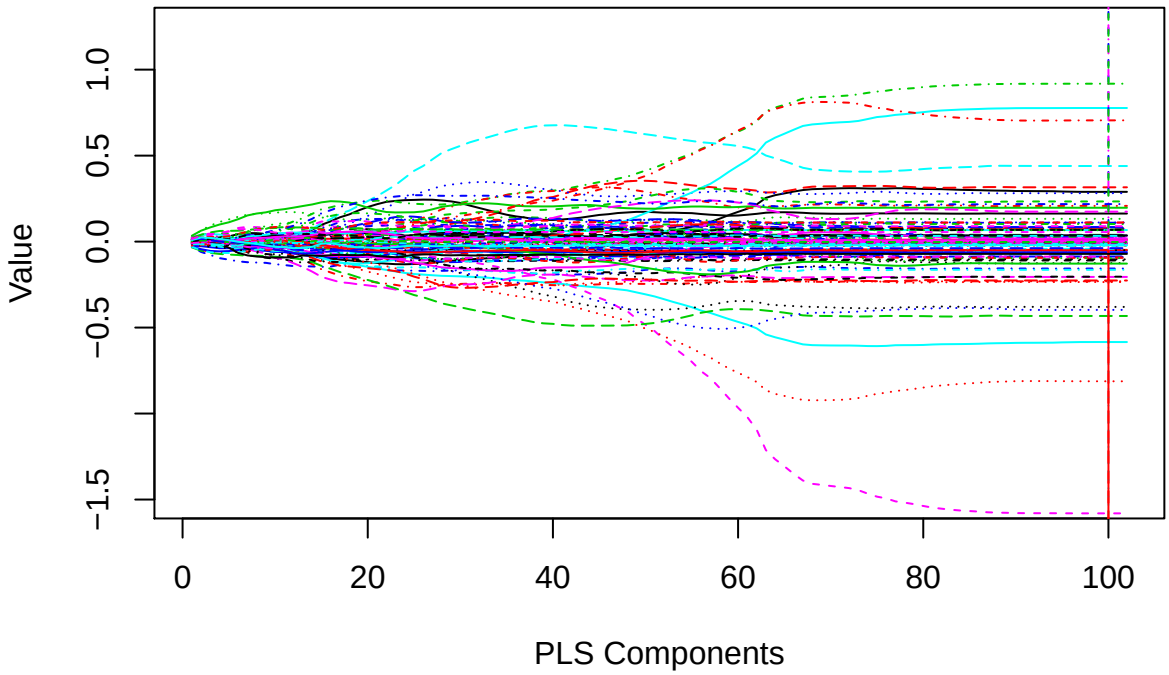
The graph consists of a different line for each coefficient (total of 102 because there are 102 features). Each line captures how each of the coefficients change as the number of retained PCs increases. The coefficients start off stable but as the number of PCs increase, we can see that the values of the coefficients start to grow both in the postive and negative directions. We observe something interesting when the number of PCs reach 60. We can see that at that point, the values of the coefficients begin to fluctuate really rapidly, leading to significant differences between them.

Partial Least Squares Regression

The next method is Partial Least Squares Regression (PLSR). It is very similar to PCR except in PLS, it makes use of the response in order to identify new features (PLS components) that not only approximate the old features well, but also that are related to the response. The function `plsr` returns the regression coefficients in terms of the X variables, which simplifies interpretation yet again. Similar to what we did with PCR, We can visualize how the coefficients grow in terms of the number of retained PLS components.

```
#PLSR on training data
plsr_fit <-plsr(formula, data = Xtrain, scale = FALSE, ncomp = ncol(X))

PLSR <- apply(plsr_fit$coefficients, MARGIN = 3,function(u) u)
matplot(t(PLSR), type = "l", xlim = c(1, 102), ylim = c(-1.5, 1.25), ylab = "Value", xlab = "PLS Compon
```



The graph consists of a different line for each coefficient (total of 102 because there are 102 features). Each line captures how each of the coefficients change as the number of retained PLS components increases. Unlike the coefficients in PCR, we notice that the cofficients in PLS experience much greater changes and are much more sensitive to the number of components retained.

Ridge Regression

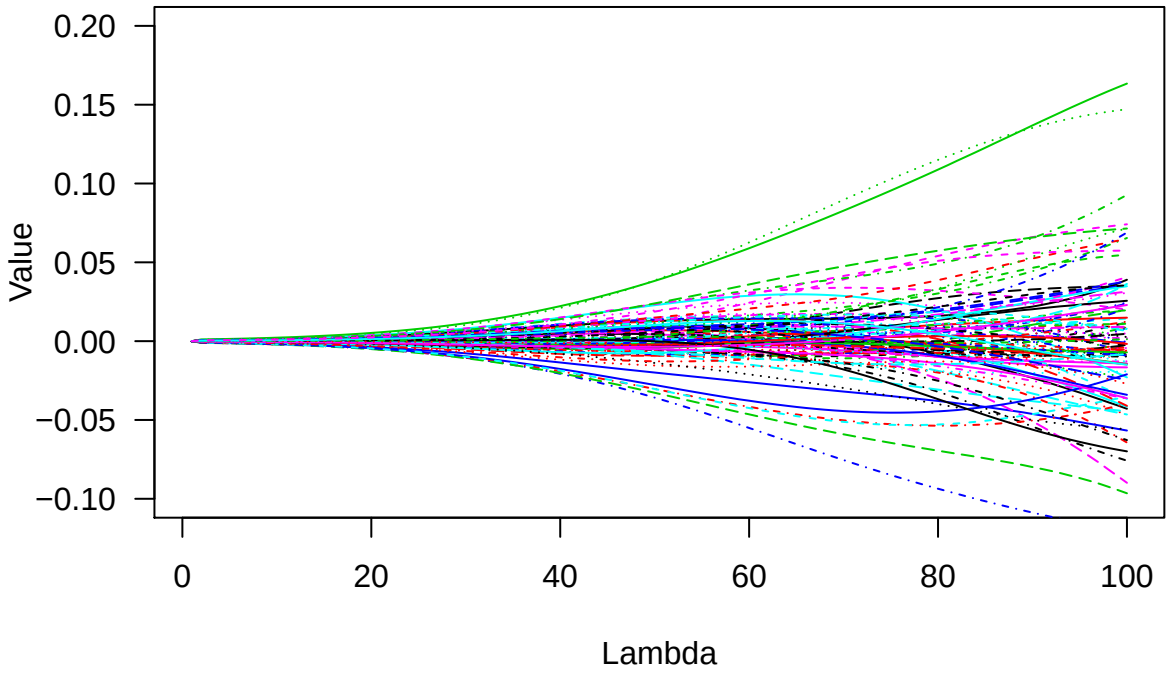
So far, we fitted two dimension reduction methods on our training data and observed how the regression coefficients change when we retain different numbers of components for PCR and PLSR. We can also use shrinkage methods like Ridge regression (RR) and Lasso regression to examine some interesting effects that the tuning parameter, lambda, has on the regression coefficients. This way, we will be able to have multiple sources of comparision, providing us

more insight about the different regularization techniques used. Below is a graph of how the regression coefficients change in terms of lambda for Ridge regression.

```
#ridge regression on training data
x <- as.matrix(Xtrain[ ,predictors])
y <- Xtrain[ ,response]

ridge_fit <- glmnet(x, y, alpha = 0)

matplot(t(ridge_fit$beta), type = "l", las = 1, ylim = c(-0.1, 0.2), ylab = "Value", xlab = "Lambda")
```



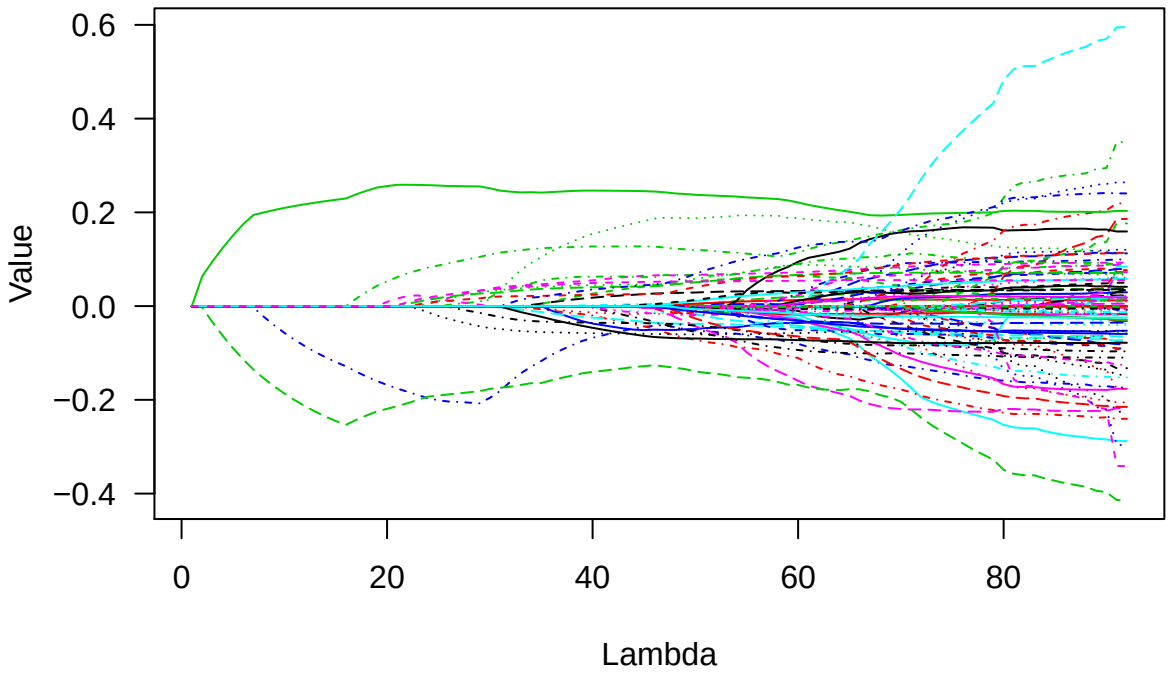
The graph consists of a different line for each coefficient (total of 102 because there are 102 features). Each line captures how each of the coefficients change with varying values of lambda. We notice that the change in value of the cofficients for RR follow a similar shape as that of PLSR and PCR.

Lasso Regression

Lasso regression is similar to Ridge regression in the sense that both are shrinkage methods that involve the parameter, lambda. However, one disadvantage that Ridge regression has is the fact that RR will include all features (in this case 102) in the final model. Although this may not be a problem for prediction accuracy, it may make model interpretation somewhat challenging especially when the number of features we are working with is large. Lasso regression helps overcome this issue by slightly adjusting the way the penalty is set up. Below is a graph of how the regression coefficients change in terms of lambda for Lasso regression.

```r
#lasso regression on training data
lasso_fit <- glmnet(x, y, alpha = 1)

matplot(t(lasso_fit$beta), type = "l", las = 1, ylab = "Value", xlab = "Lambda")
```



The graph consists of a different line for each coefficient (total of 102 because there are 102 features). Each line captures how each of the coefficients change with varying values of lambda. Again, we notice here that the change in value of the cofficients for Lasso follow a similar shape as that of PLSR and PCR. The one thing that stands out about the regression coefficients for Lasso is how sensitive it is to lambda early on.

Cross-Validation

The question to ask now is: what is the best number of components to have? What is the best lambda to use? The answers to these questions can be found by using k-fold cross validation. The size of k is somewhat arbitrary, but typically 5 or 10 is good. We will go with k=5 here. That means we will have to create folds by splitting the training data into 5 subsets of similar size and then train on the first k-1 folds and reserve the kth fold for "testing". We repeat this process for each fold, training on all folds except for the one that has been reserved for "testing".

```r
#create folds
folds <- 5
if ((length(train)%%folds)==0){
  cuts <- seq(1,length(train), by = length(train)/folds)
} else {
```

```
    cuts <- seq(1,length(train), by = floor(length(train)/folds))
    cuts[folds+1] <-length(train)
}

#starting and ending positions of folds
fold_starts <- cuts[-length(cuts)]
fold_ends <- c(cuts[2:(length(cuts)-1)]-1, cuts[length(cuts)])

folds_list <-as.list(1:folds)
for(fold in 1:folds) {
  folds_list[[fold]] <- fold_starts[fold]:fold_ends[fold]
}
```

Once we have created the folds, we will train our methods on the training data and compute cross-validation MSEs for each value of a tuning parameter (the number of components and lambda are examples of tuning parameters). We note the tuning parameters that yield the lowest cross-validation MSE for each model (training MSE for each). The following code uses a for loop to do this process for PCR and PLSR. We also make use of the `cv.glmnet` function to do this process for Ridge regression and Lasso regression.

```
mse_cv_pcr <- c()
mse_cv_plsr <- c()

#pcr and plsr
for(tune in 1:ncol(X)) {
  mse_pcr_aux <- c()
  mse_plsr_aux <- c()
  for(fold in 1:folds) {
    #train with k-1 folds
    Xtrain_folds <- scale(Xtrain[folds_list[[fold]], ])
    Xtrain_folds <- as.data.frame(Xtrain_folds)
    x_aux <- as.matrix(Xtrain_folds[, predictors])
    y_aux <- Xtrain_folds[,response]
    pcr_aux <- pcr(formula, data = Xtrain_folds, scale = FALSE, ncomp = tune)
    plsr_aux <-plsr(formula, data = Xtrain_folds, scale = FALSE, ncomp = tune)

    # predict holdout fold
    Xfoldout <- scale(Xtrain[unlist(folds_list[-fold]), predictors])
    Xfoldout <- as.data.frame(Xfoldout)
    yfoldout <- Xtrain[unlist(folds_list[-fold]), response]

    pcr_hat <- predict(pcr_aux, Xfoldout, ncomp = tune)
    plsr_hat <- predict(plsr_aux, Xfoldout, ncomp = tune)

    #measure performance
    mse_pcr_aux[fold] <- mean((pcr_hat[,,1]-yfoldout)^2)
    mse_plsr_aux[fold] <- mean((plsr_hat[,,1]-yfoldout)^2)
  }
  mse_cv_pcr[tune] <- mean(mse_pcr_aux)
  mse_cv_plsr[tune] <- mean(mse_plsr_aux)
}

lasso <- cv.glmnet(as.matrix(Xtrain[,c(1:102)]), Xtrain[,response], alpha = 1, type.measure = "mse", nf
ridge <- cv.glmnet(as.matrix(Xtrain[,c(1:102)]), Xtrain[,response], alpha = 0, type.measure = "mse", nf
```

```r
#lasso
lasso_keep <- lasso$nzero[lasso$lambda == lasso$lambda.min]

#ridge
ridge_keep <- ridge$nzero[ridge$lambda == ridge$lambda.min]

#show the best tuning parameter for each method
cv_mse <- data.frame(pcr = mse_cv_pcr, plsr = mse_cv_plsr)
selected_pcr_plsr <- apply(cv_mse, 2, which.min)
keep <- c(selected_pcr_plsr, "ridge" = ridge$lambda.min, "lasso" = lasso$lambda.min)
keep
```

```
##         pcr        plsr       ridge       lasso
## 14.00000000  4.00000000  0.32492694  0.01223329
```

This table summarizes, for every method, the best number of components to keep as well as the best lambda to choose (determined by the lowest cross-validation MSE for each). Now that we have found the best tuning parameters for each model, the next question to ask is: which model is the best one? We will use the validation set to determine the answer. In the next step, we will fit our ideal models (those with the best tuning parameters that we just found for each) on our validation set and compare how each model does.

```r
Xval <- scale(dat[validation, c(predictors, response)])
x_val <- as.matrix(Xval[,predictors])
y_val <- Xval[, response]

pcr_hat <- predict(pcr_fit, x_val, ncomp = keep[1])
plsr_hat <- predict(plsr_fit, x_val, ncomp = keep[2])
las_hat <- predict(lasso_fit, newx = x_val, s = keep[3])
rr_hat <- predict(ridge_fit, newx = x_val, s = keep[4])

mse_val <- c(
  "pcr" = mean((pcr_hat[,,1]-y_val)^2),
  "plsr" = mean((plsr_hat[,,1]-y_val)^2),
  "ridge" = mean((rr_hat[,1]-y_val)^2),
  "lasso" = mean((las_hat[,1]-y_val)^2)
)

mse_val
```

```
##       pcr      plsr     ridge     lasso
## 0.3624519 0.3588764 0.3476880 0.5369280
```

```r
which.min(mse_val)
```

```
## ridge
##     3
```

The model that gives us the lowest validation MSE is the best model out of the bunch; in our case, Ridge Regression yields the lowest validation MSE. Once we have identified the best model, we need to assess its performance using the test set.

Final Model and its Performance

```
#test the performance of the best model
Xbest <- scale(dat[c(train, validation),c(predictors, response)])
Xtest <- scale(dat[test,c(predictors, response)])
x_test <- as.matrix(Xtest[, predictors])
y_test <- Xtest[,response]

ridge_best <- glmnet(Xbest[,predictors], Xbest[,response], alpha = 0, lambda = keep[4])
y_hat_test <- predict(ridge_fit, newx = x_test, s = keep[4])
mean((y_hat_test-y_test)^2)
```

```
## [1] 0.331372
```

The resulting quantity above shows the test MSE for our best model (Ridge regression). We note that the test MSE is greater than the validation MSE, but by a very small amount. We can conclude that our model did a fairly good job in predicting "ViolentCrimesPerPop". For our last step, we will use our entire dataset (training set, validation set, and test set) to fit our crowned model and observe its coefficients.

```
Xall <- scale(dat)
ridge_model <- glmnet(Xall[,predictors], Xall[,response], alpha = 0, lambda = keep[4])
ridge_model$beta
```

```
## 102 x 1 sparse Matrix of class "dgCMatrix"
##                                  s0
## population            -0.0491638432
## householdsize         -0.0296575851
## racepctblack           0.1360736818
## racePctWhite          -0.0711683487
## racePctAsian          -0.0270917885
## racePctHisp            0.0008447537
## agePct12t21            0.0474468570
## agePct12t29           -0.1631222262
## agePct16t24            0.0149398041
## agePct65up            -0.0220666184
## numbUrban             -0.0662126258
## pctUrban               0.0782010845
## medIncome              0.0025570737
## pctWWage              -0.1230538106
## pctWFarmSelf           0.0204778255
## pctWInvInc            -0.0846178506
## pctWSocSec             0.0441072657
## pctWPubAsst            0.0578071799
## pctWRetire            -0.0769737452
## medFamInc             -0.0105337681
## perCapInc             -0.0488610155
## whitePerCap           -0.0144640937
## blackPerCap           -0.0117364695
## indianPerCap          -0.0015510813
## AsianPerCap            0.0332978613
## OtherPerCap            0.0333035847
## HispPerCap             0.0116231328
```

```
## NumUnderPov           -0.0294094010
## PctPopUnderPov        -0.0899341263
## PctLess9thGrade       -0.1630281039
## PctNotHSGrad           0.1008233839
## PctBSorMore            0.0348004401
## PctUnemployed         -0.0287455401
## PctEmploy              0.0927356630
## PctEmplManu           -0.0577608813
## PctEmplProfServ       -0.0143637236
## PctOccupManu           0.0273220063
## PctOccupMgmtProf       0.0263912145
## MalePctDivorce         0.1572377452
## MalePctNevMarr         0.1145609615
## FemalePctDiv          -0.0852828205
## TotalPctDiv           -0.0272195207
## PersPerFam            -0.0223951462
## PctFam2Par            -0.0268378529
## PctKids2Par           -0.1644580429
## PctYoungKids2Par       0.0111063241
## PctTeen2Par           -0.0018010256
## PctWorkMomYoungKids    0.0302839580
## PctWorkMom            -0.1053712451
## NumKidsBornNeverMar   -0.0722701022
## PctKidsBornNeverMar    0.2087152419
## NumImmig               0.0605466888
## PctImmigRecent         0.0224669666
## PctImmigRec5          -0.0130889453
## PctImmigRec8          -0.0122816049
## PctImmigRec10          0.0177751472
## PctRecentImmig        -0.0145640151
## PctRecImmig5          -0.0368117995
## PctRecImmig8          -0.0078361943
## PctRecImmig10          0.0283682677
## PctSpeakEnglOnly       0.0125285763
## PctNotSpeakEnglWell   -0.0456592215
## PctLargHouseFam        0.0081864184
## PctLargHouseOccup     -0.0760607938
## PersPerOccupHous       0.1568150205
## PersPerOwnOccHous     -0.0288321659
## PersPerRentOccHous    -0.0266881439
## PctPersOwnOccup       -0.1048253263
## PctPersDenseHous       0.1333692545
## PctHousLess3BR         0.0404593300
## MedNumBR               0.0117927734
## HousVacant             0.1516442887
## PctHousOccup          -0.0348632729
## PctHousOwnOcc          0.0523937702
## PctVacantBoarded       0.0767204224
## PctVacMore6Mos        -0.0480007455
## MedYrHousBuilt         0.0020052541
## PctHousNoPhone         0.0324747898
## PctWOFullPlumb        -0.0062906739
## OwnOccLowQuart        -0.0079906551
## OwnOccMedVal           0.0160187159
```

```
## OwnOccHiQuart         -0.0139603868
## OwnOccQrange          -0.0249140880
## RentLowQ              -0.0999785921
## RentMedian             0.0014985799
## RentHighQ             -0.0533925818
## RentQrange             0.0431898420
## MedRent                0.1232714127
## MedRentPctHousInc      0.0121121724
## MedOwnCostPctInc      -0.0103880424
## MedOwnCostPctIncNoMtg -0.0744301120
## NumInShelters          0.0789941307
## NumStreet             -0.0039622510
## PctForeignBorn         0.1349230872
## PctBornSameState      -0.0249750695
## PctSameHouse85         0.0069380749
## PctSameCity85          0.0259725524
## PctSameState85         0.0148322542
## LandArea               0.0020638694
## PopDens               -0.0531134884
## PctUsePubTrans        -0.0041650996
## LemasPctOfficDrugUn    0.0451312598
```