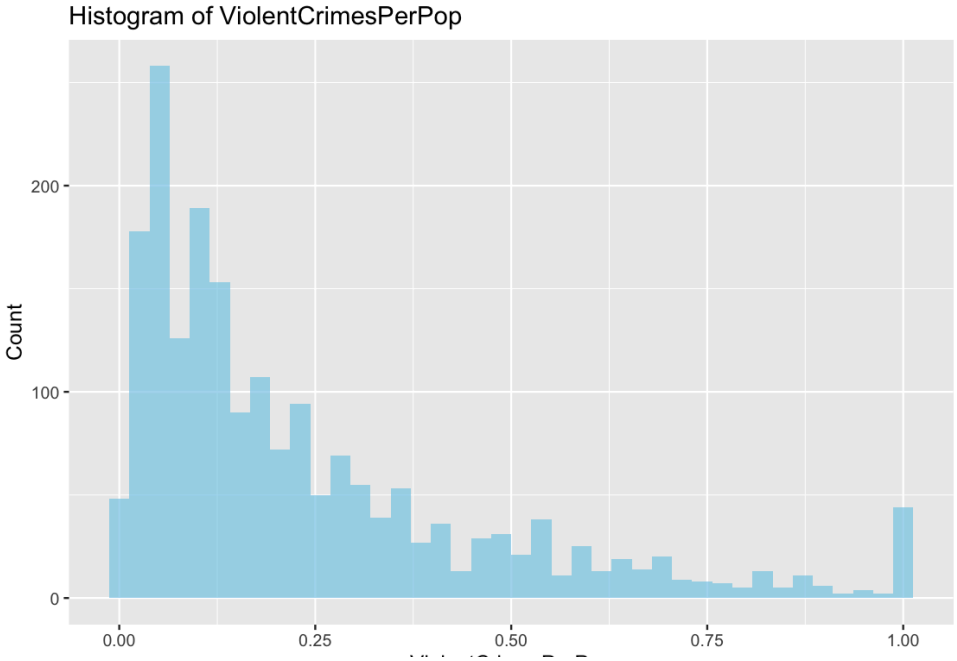
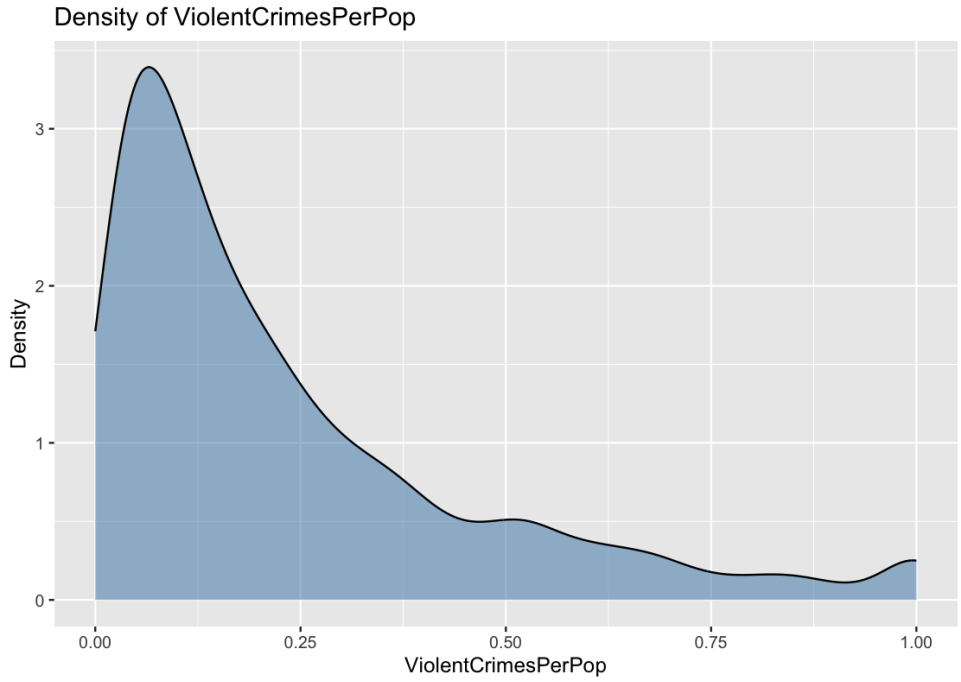
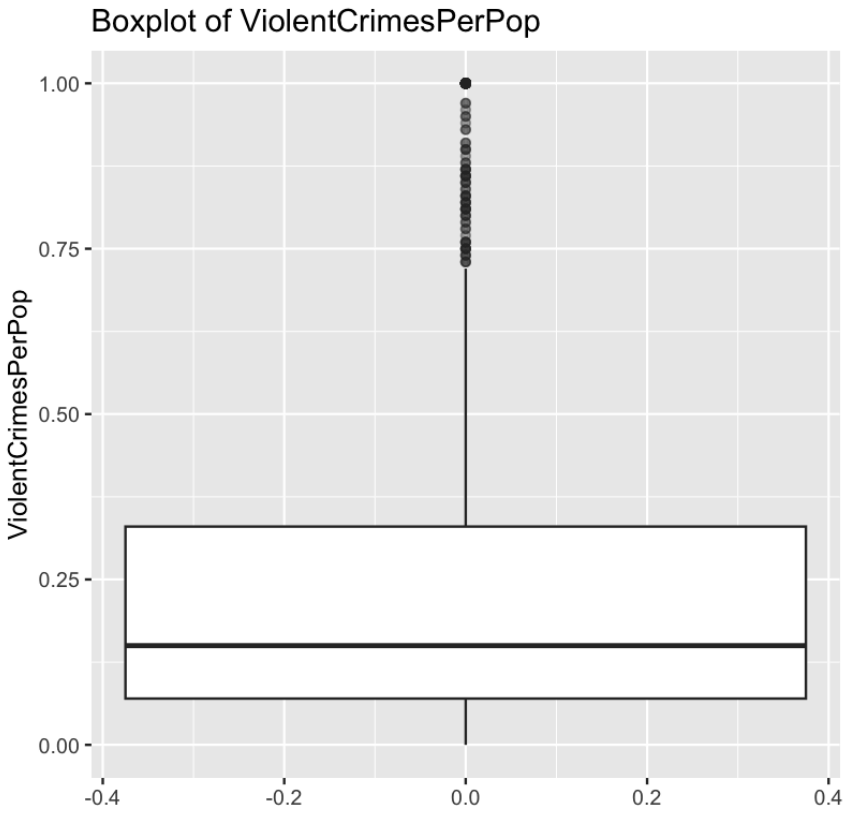
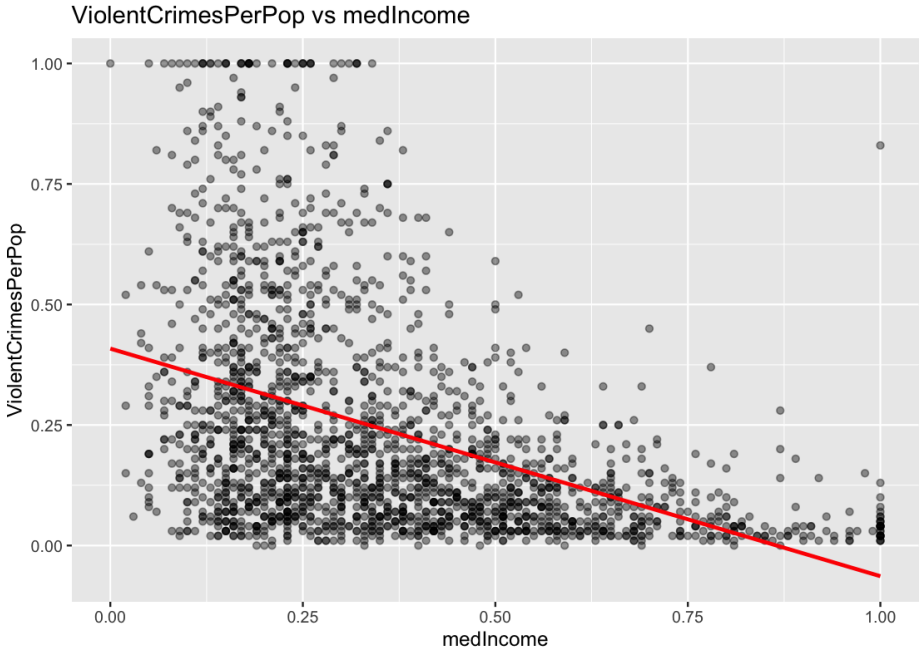
1. Exploratory Data Analysis (3%) Explore the statistical aspects of the dataset. Analyze the distributions and provide summaries of the relevant statistics. Perform any cleaning, transformations, interpolations, smoothing, outlier detection/ removal, etc. required on the data. Include figures and descriptions of this exploration and a short description of what you concluded (e.g. nature of distribution, indication of suitable model approaches you would try, etc.) Min.1 page text + graphics (required).

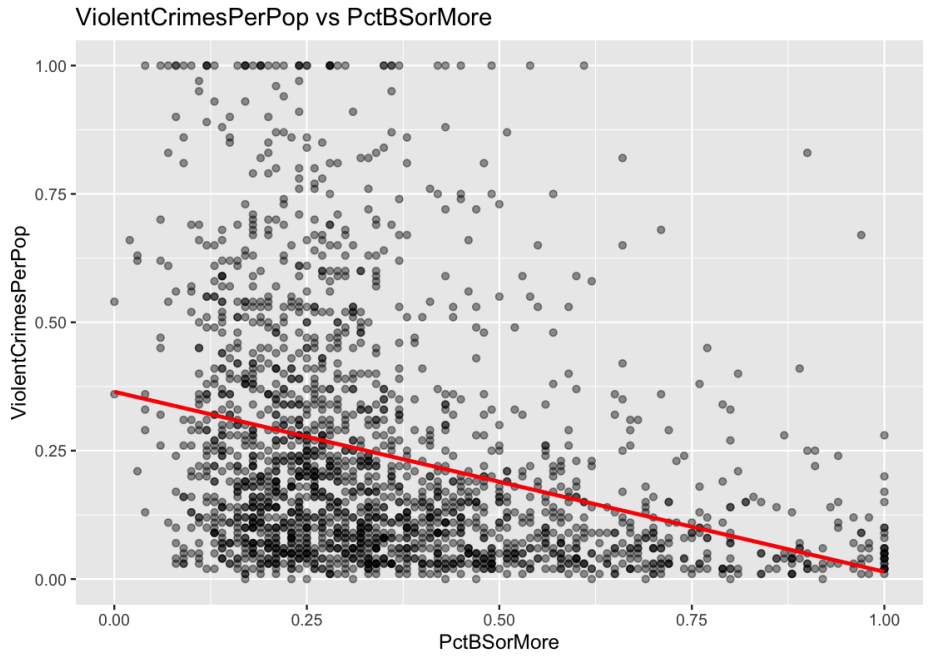


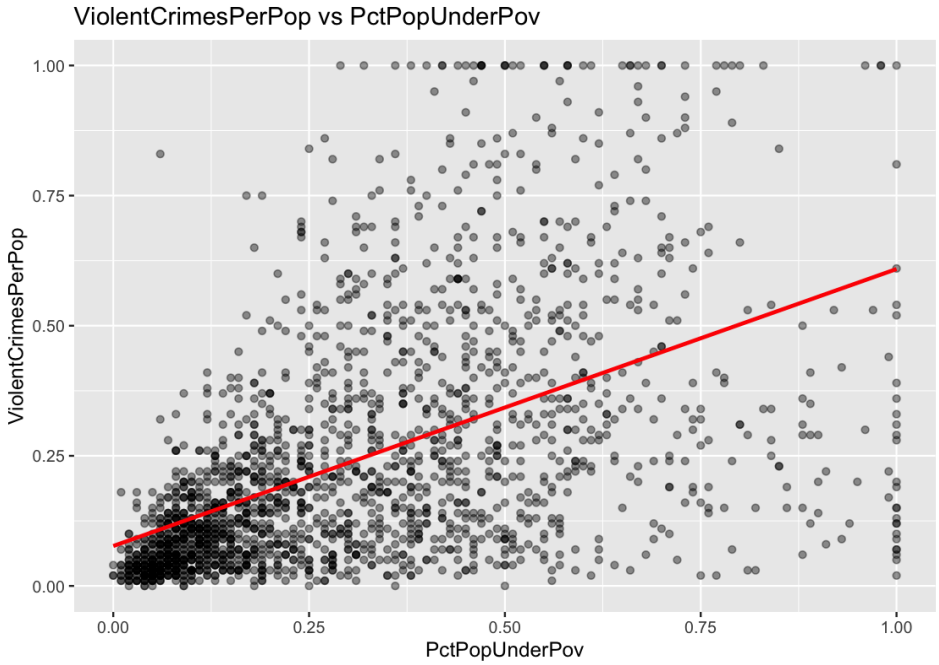


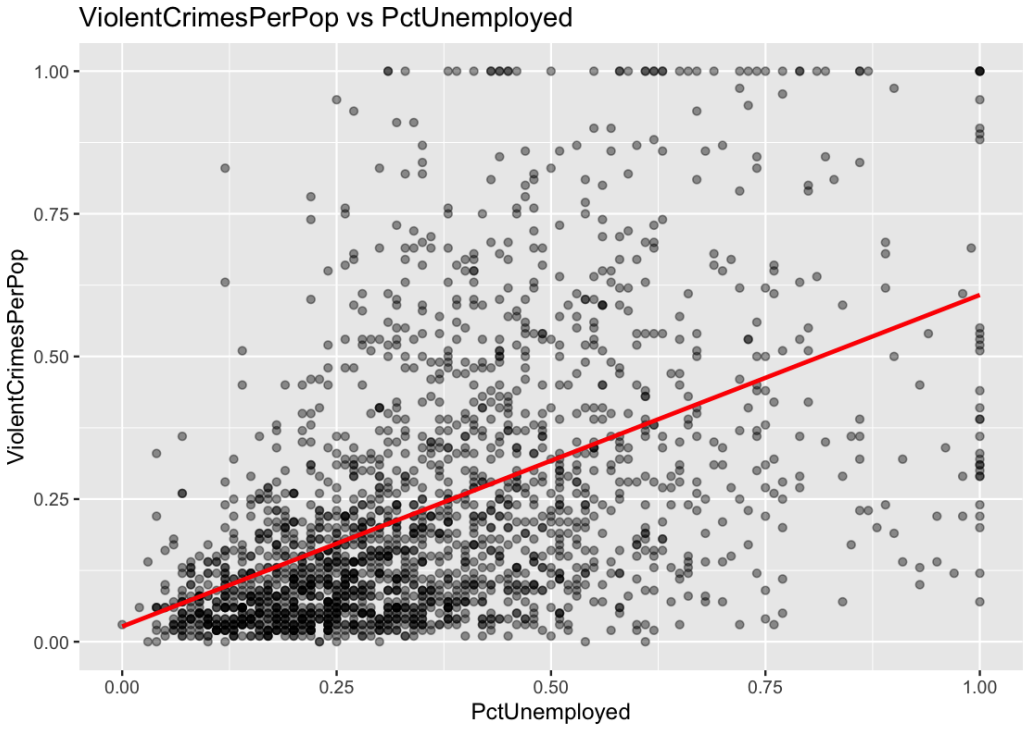

Boxplot of ViolentCrimesPerPop
ViolentCrimesPerPop
1.00
0.75
0.50
0.25
0.00
-0.4
-0.2
0.0
0.2
0.4

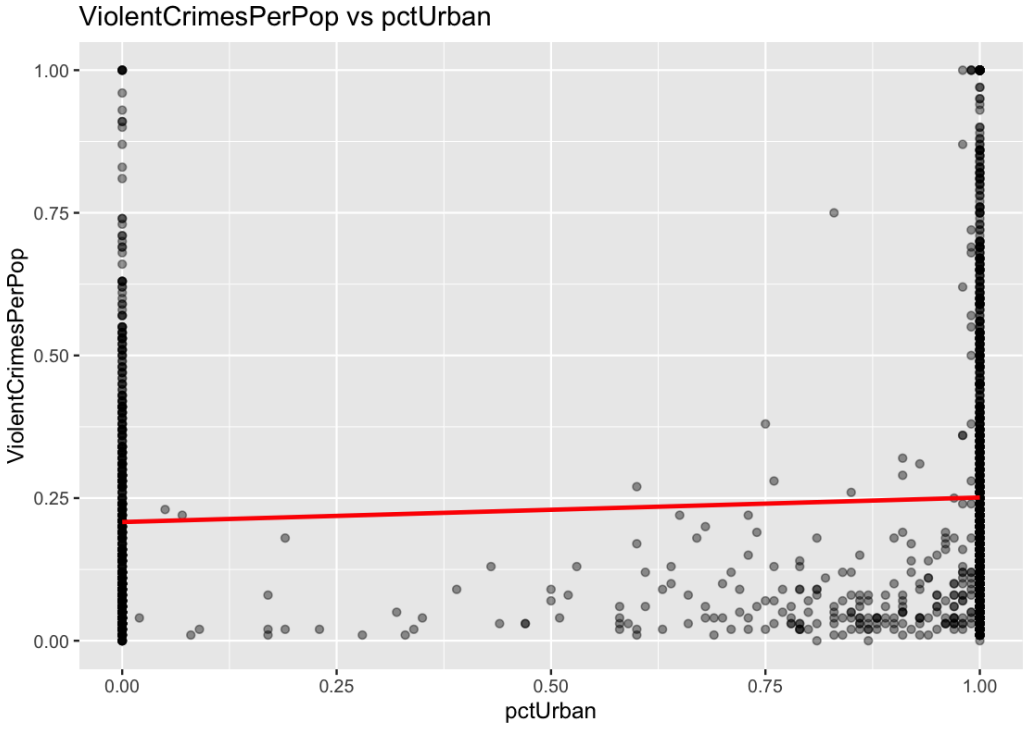

ViolentCrimesPerPop vs medIncome
ViolentCrimesPerPop
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
medIncome

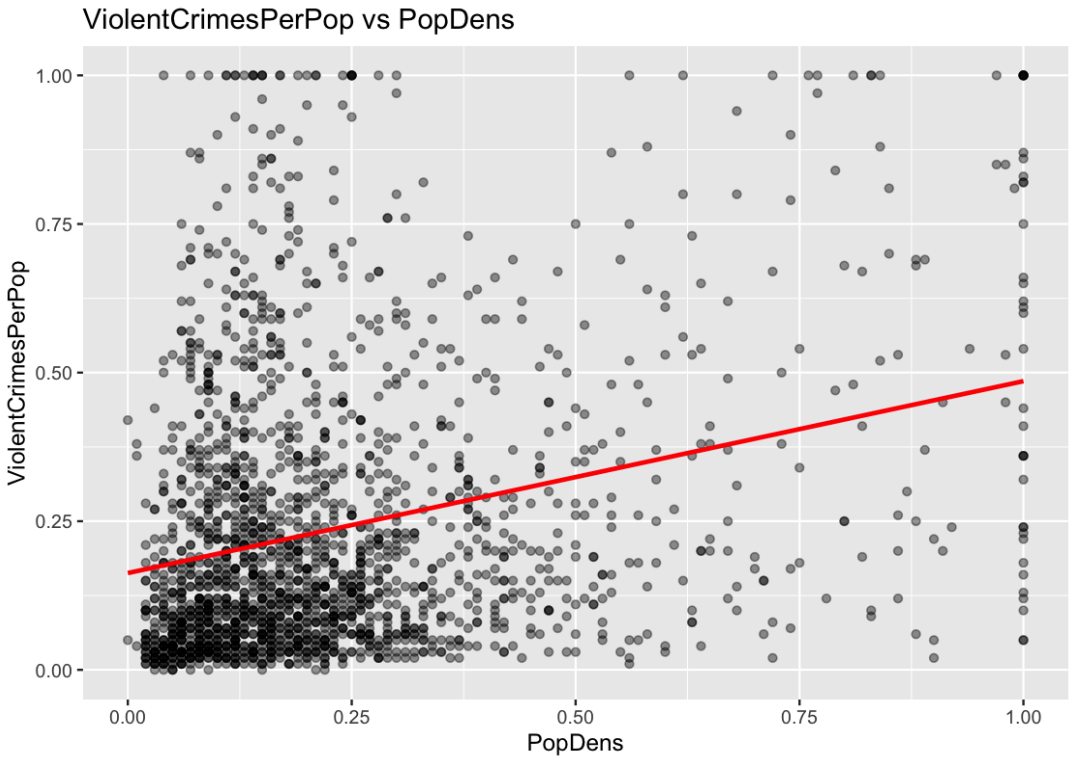
‘

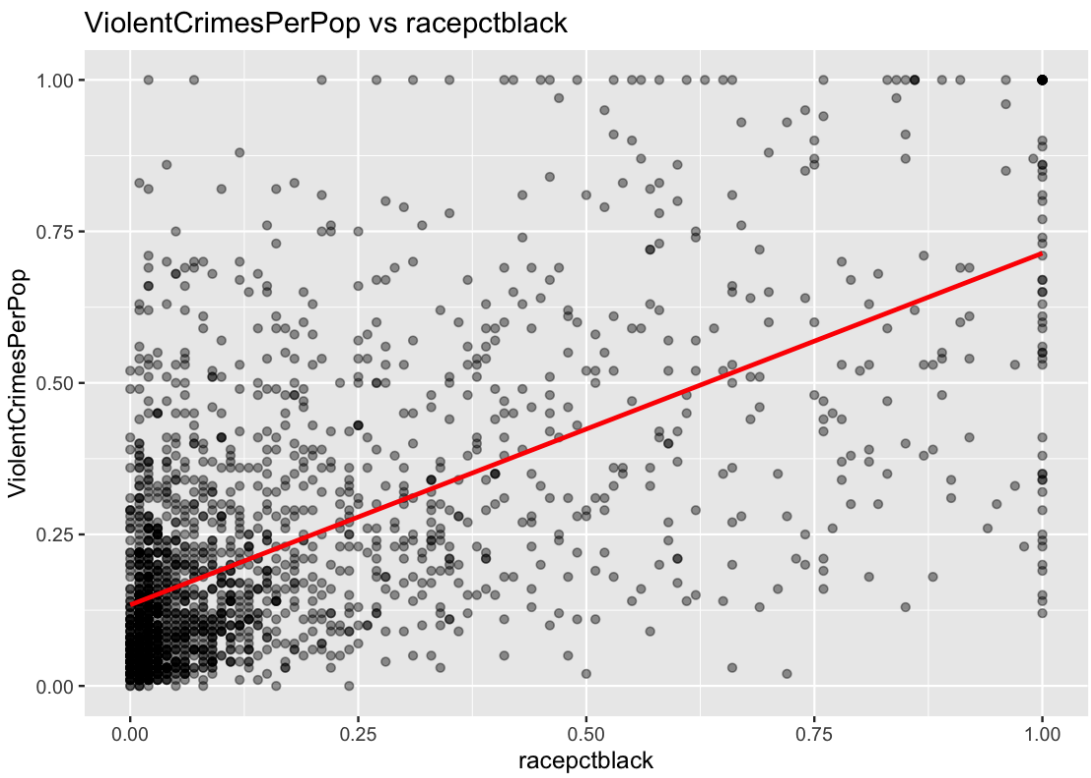

ViolentCrimesPerPop vs PctBSorMore
ViolentCrimesPerPop
PctBSorMore
0.00
0.25
0.50
0.75
1.00

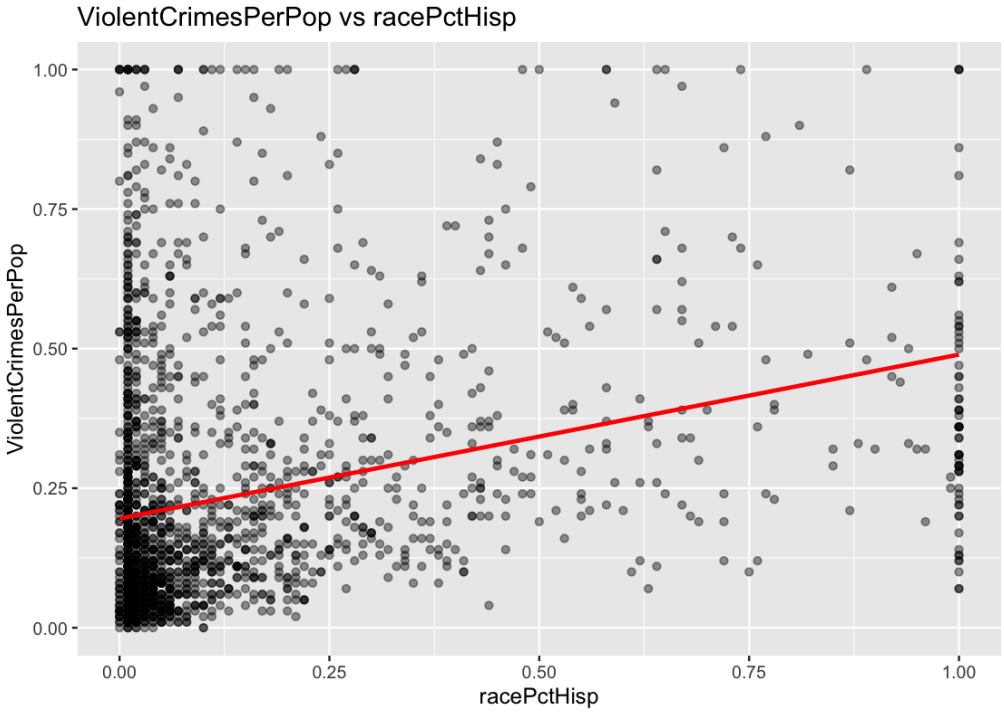

ViolentCrimesPerPop vs PctPopUnderPov
ViolentCrimesPerPop
PctPopUnderPov
0.00
0.25
0.50
0.75
1.00

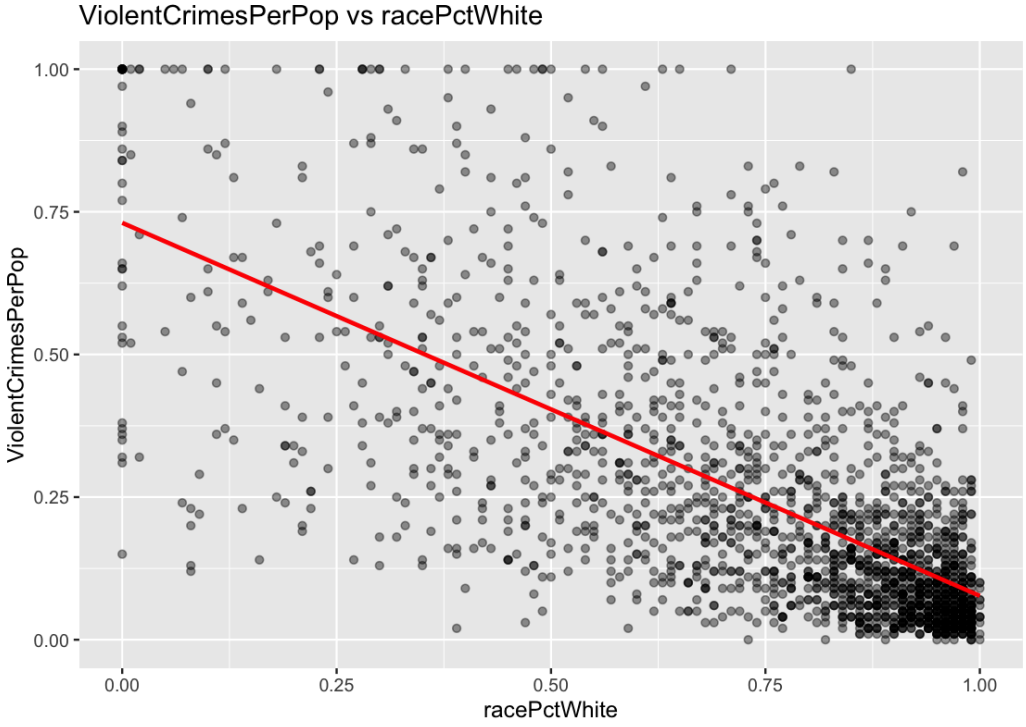

ViolentCrimesPerPop vs PctUnemployed
ViolentCrimesPerPop
PctUnemployed
0.00
0.25
0.50
0.75
1.00

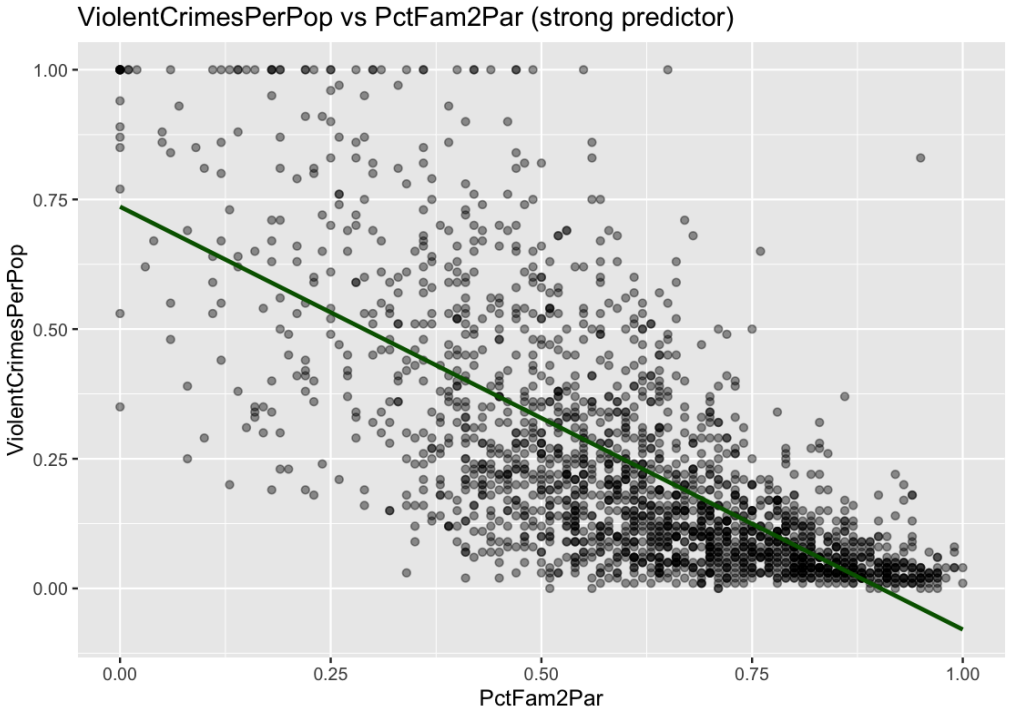

ViolentCrimesPerPop vs pctUrban
ViolentCrimesPerPop
pctUrban
0.00
0.25
0.50
0.75
1.00

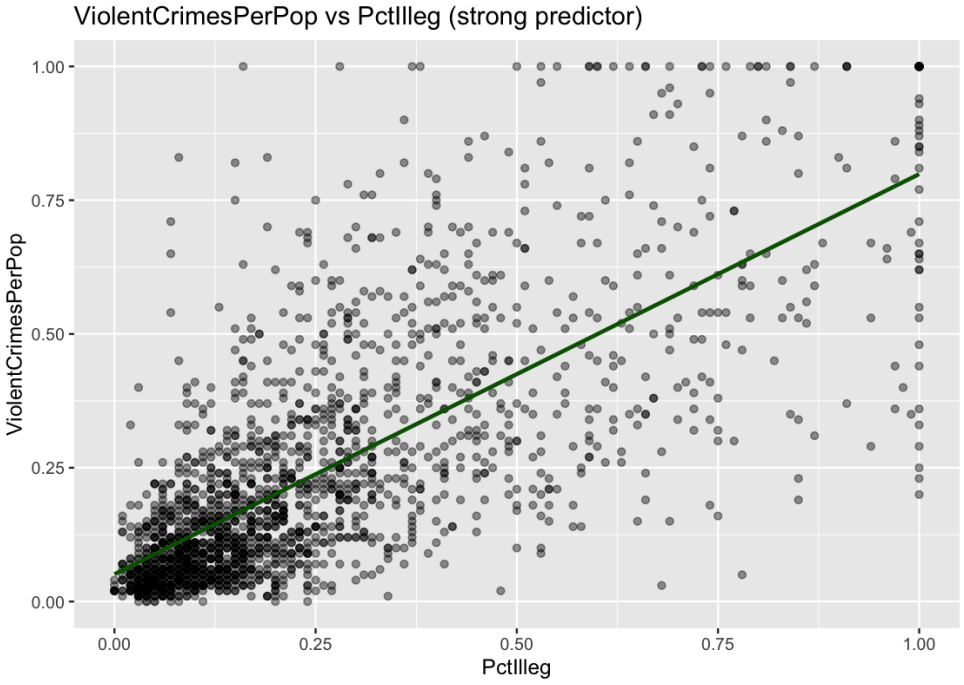

ViolentCrimesPerPop vs PopDens
ViolentCrimesPerPop
PopDens
0.00
0.25
0.50
0.75
1.00

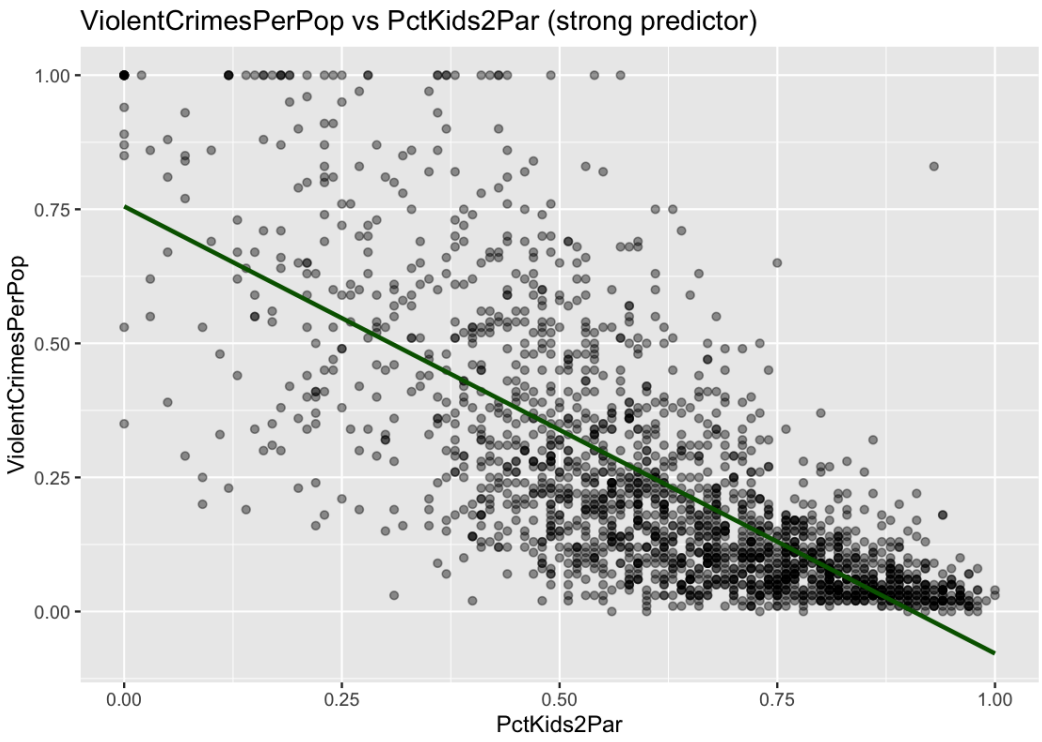

ViolentCrimesPerPop vs racepctblack
ViolentCrimesPerPop
racepctblack
0.00
0.25
0.50
0.75
1.00


ViolentCrimesPerPop vs racePctHisp
ViolentCrimesPerPop
racePctHisp
0.00
0.25
0.50
0.75
1.00


ViolentCrimesPerPop vs racePctWhite
ViolentCrimesPerPop
racePctWhite
0.00
0.25
0.50
0.75
1.00


ViolentCrimesPerPop vs PctFam2Par (strong predictor)
ViolentCrimesPerPop
PctFam2Par
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00


ViolentCrimesPerPop vs PctIlleg (strong predictor)
ViolentCrimesPerPop
PctIlleg
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00




```
--- Summary of ViolentCrimesPerPop ---
> print(summary(comm[[target]]))
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.070   0.150   0.238   0.330   1.000
>
```

```
--- Summaries of selected predictors ---
> print(summary(eda.df))
 ViolentCrimesPerPop   medIncome      PctPopUnderPov  PctUnemployed
 Min.   :0.000       Min.   :0.0000   Min.   :0.000   Min.   :0.0000
 1st Qu.:0.070       1st Qu.:0.2000   1st Qu.:0.110   1st Qu.:0.2200
 Median :0.150       Median :0.3200   Median :0.250   Median :0.3200
 Mean   :0.238       Mean   :0.3611   Mean   :0.303   Mean   :0.3635
 3rd Qu.:0.330       3rd Qu.:0.4900   3rd Qu.:0.450   3rd Qu.:0.4800
 Max.   :1.000       Max.   :1.0000   Max.   :1.000   Max.   :1.0000
  PctBSorMore      racepctblack     racePctWhite     racePctHisp
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.000
 1st Qu.:0.2100   1st Qu.:0.0200   1st Qu.:0.6300   1st Qu.:0.010
 Median :0.3100   Median :0.0600   Median :0.8500   Median :0.040
 Mean   :0.3617   Mean   :0.1796   Mean   :0.7537   Mean   :0.144
 3rd Qu.:0.4600   3rd Qu.:0.2300   3rd Qu.:0.9400   3rd Qu.:0.160
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.000
    pctUrban         PopDens
 Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.0000   1st Qu.:0.1000
 Median :1.0000   Median :0.1700
 Mean   :0.6963   Mean   :0.2329
 3rd Qu.:1.0000   3rd Qu.:0.2800
 Max.   :1.0000   Max.   :1.0000
```

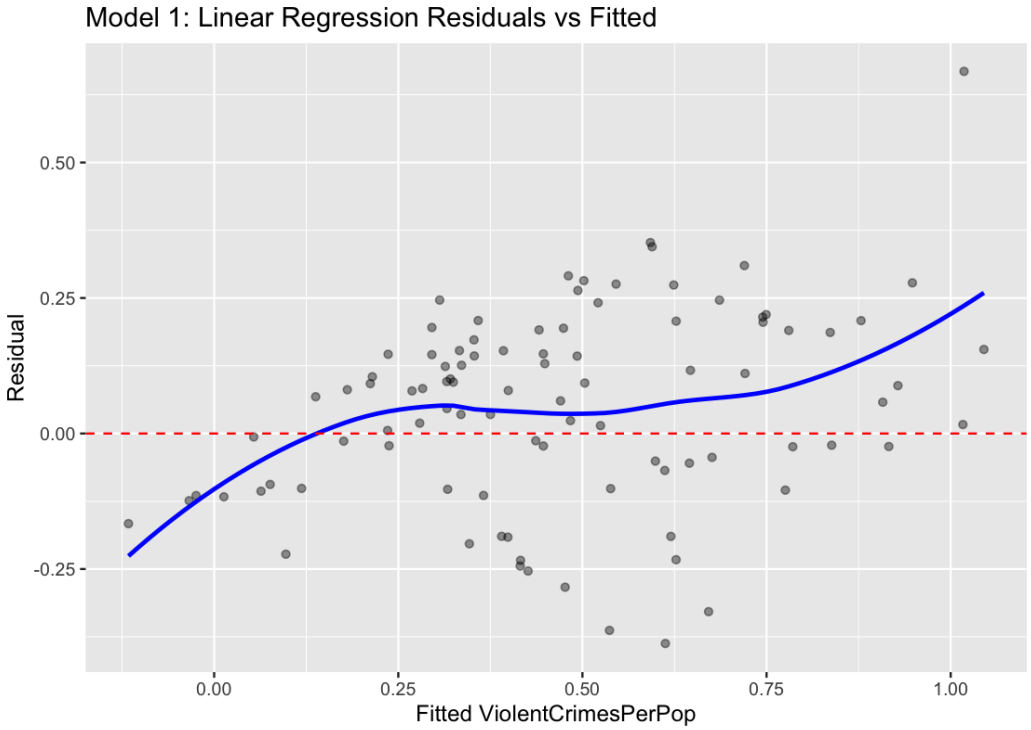
In the exploratory stage of this project, I began by examining the statistical structure of *ViolentCrimesPerPop*, not only to understand where crime intensity is concentrated, but also to develop an instinct for how the distribution might influence later modeling choices. Summary statistics, paired with histogram, density, and boxplot visualizations, reveal a distinctly right-skewed distribution: most communities lie in the lower-crime regime, while a long heavy tail reaches the maximum bound of 1.000, marking the existence of highly affected regions that cannot be dismissed as noise. Instead of treating these high values as anomalies, I preserved them, acknowledging that they represent real sociological extremities rather than measurement defects. I then turned toward relationship-level structure, generating twelve scatterplots that map crime against a range of socioeconomic indicators. Rather than interpreting each in isolation, the collection as a whole illustrates a clear pattern—variables tied to stability and opportunity tend to depress crime, while those reflecting poverty, unemployment, or family instability move sharply in the opposite direction. Strong monotonic trends emerge in features such as median income, bachelor-degree attainment, two-parent household proportion, and birth-to-unmarried-mother percentage, while other predictors such as population density or urban proportion display far weaker gradients. Race-linked variables introduce additional statistical contrast: *racePctWhite* declines as crime rises, while *racepctblack* and *racePctHisp* move upward, though I treat these as demographic reflections of underlying socioeconomic conditions rather than causal mechanisms. Missing values were reviewed and addressed, the dataset retained high-crime outliers to preserve distributional truth, and scaling was prepared for later non-linear models such as k-NN. Taken together, the distributional asymmetry, variance across predictors, and visible nonlinear structure suggest that while linear models may capture broad tendencies, flexible methods like Random Forest and neighbor-based classifiers may more faithfully reflect the layered social complexity embedded in this dataset.

2. Model Development, Validation and Optimization (10% 4000-level / 7% 6000-level)
Develop and evaluate three (4000-level) or four (6000-level) or more ☺ models. **If possible**, these models should cover more than one objective, i.e. regression, classification, clustering. Consider the effect of dimension reduction of the dataset on model performance. Different models means different combinations of an algorithm and a formula (input and output features). The choice of independent and response variables is up to you.

Explain why you chose them. Construct the models, test/ validate them. Briefly explain the validation approach. You can use any method(s) covered in the course. Include your code in your submission. Compare model results if applicable. Report the results of the model (fits, coefficients, sample trees, other measures of fit/ importance, etc., predictors and summary statistics). Min. 2 pages of text + graphics (required).

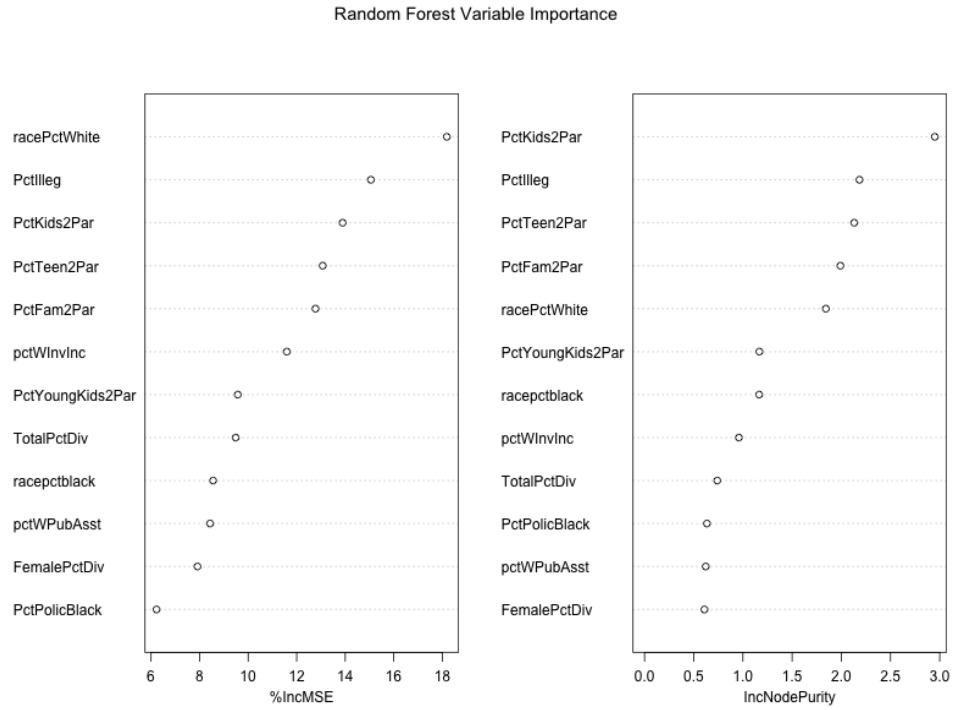
```
[LM] Summary:
> print(summary(lm.mod))

Call:
lm(formula = form.lm, data = train.reg)

Residuals:
     Min       1Q   Median       3Q      Max
-0.49718 -0.10792 -0.01830  0.09558  0.43049

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)       0.46151    0.26961   1.712   0.0884 .
PctKids2Par      -0.48093    0.43437  -1.107   0.2695
PctIlleg          0.17285    0.14055   1.230   0.2201
PctFam2Par       -0.22363    0.48317  -0.463   0.6440
racePctWhite     -0.27646    0.10665  -2.592   0.0102 *
PctYoungKids2Par  0.38181    0.21694   1.760   0.0799 .
PctTeen2Par       0.05780    0.21930   0.264   0.7924
racepctblack     -0.02945    0.10936  -0.269   0.7879
pctWInvInc       -0.18189    0.14237  -1.278   0.2028
pctWPubAsst       0.05805    0.10591   0.548   0.5842
FemalePctDiv     -0.70592    0.50377  -1.401   0.1626
TotalPctDiv       1.08890    0.49149   2.215   0.0278 *
PctPolicBlack     0.04604    0.09573   0.481   0.6311
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1662 on 210 degrees of freedom
Multiple R-squared:  0.6696,    Adjusted R-squared:  0.6507
F-statistic: 35.47 on 12 and 210 DF,  p-value: < 2.2e-16
```

```
> cat("Linear Regression RMSE:", round(rmse.lm, 4),
+     " MAE:", round(mae.lm, 4), "\n")
Linear Regression RMSE: 0.1862  MAE: 0.1518
```




```
> print(rf.reg)

Call:
 randomForest(formula = form.lm, data = train.reg, ntree = 500,      mtry = max(1, floor(sqrt(p))), importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 0.02811615
                    % Var explained: 64.28
>
> pred.rf <- predict(rf.reg, newdata = test.reg)
> err.rf  <- pred.rf - test.reg[[target]]
>
> rmse.rf <- rmse(err.rf)
> mae.rf  <- mae(err.rf)
>
> cat("Random Forest RMSE:", round(rmse.rf, 4),
+     " MAE:", round(mae.rf, 4), "\n")
Random Forest RMSE: 0.1802  MAE: 0.1419
```

```
Random Forest variable importance (sorted by X.IncMSE ):
> print(head(rf.imp.df, 10))
           Variable  X.IncMSE IncNodePurity
4      racePctWhite 18.185981     1.8432765
2          PctIlleg 15.058811     2.1845044
1       PctKids2Par 13.893399     2.9502804
6       PctTeen2Par 13.072396     2.1313653
3        PctFam2Par 12.776323     1.9906125
8        pctWInvInc 11.594021     0.9588933
5  PctYoungKids2Par  9.573050     1.1672982
11      TotalPctDiv  9.487657     0.7390265
7      racepctblack  8.557329     1.1652254
9       pctWPubAsst  8.432496     0.6221694
```



kNN

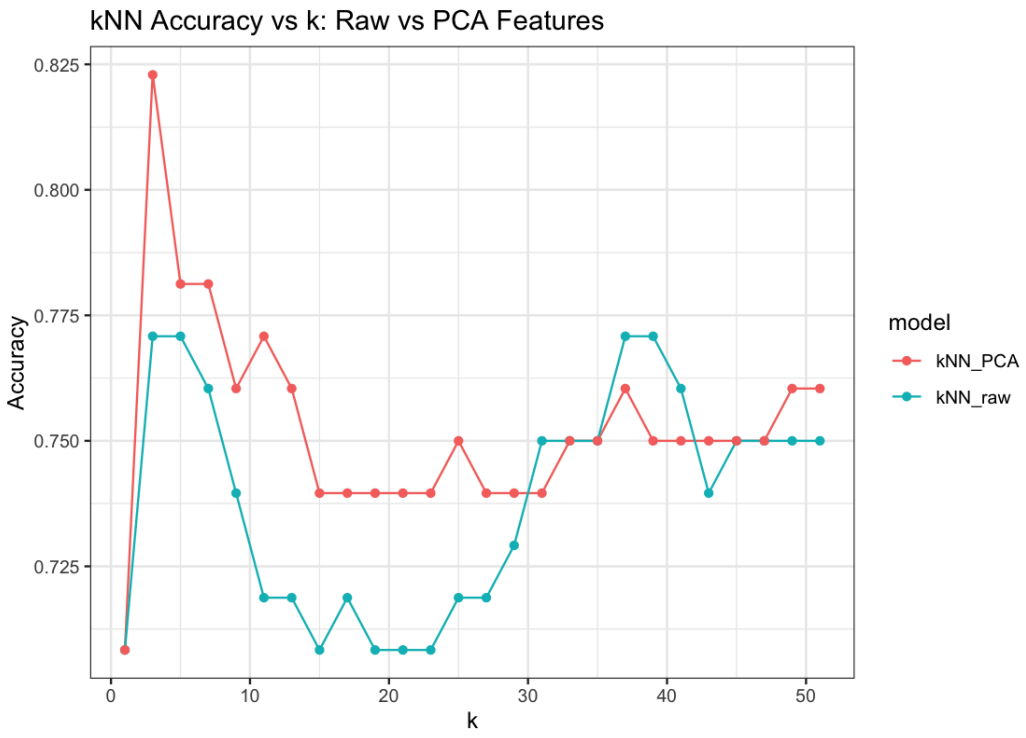
Classification:

```
> cat("Best k (by accuracy) =", best.k,
+     "with accuracy =", round(max(acc.grid), 4), "\n")
Best k (by accuracy) = 3 with accuracy = 0.7708
>
> pred.knn <- knn(train = X.train, test = X.test, cl = y.train, k = best.k)
> tab.knn  <- table(predicted = pred.knn, actual = y.test)
>
> cat("\n[Model 3] Confusion matrix (kNN, raw features):\n")

[Model 3] Confusion matrix (kNN, raw features):
> print(tab.knn)
         actual
predicted low high
     low   41   12
     high  10   33
> cat("Accuracy:", round(acc_from_tab(tab.knn), 4),
+     " MacroP:", round(macro_precision(tab.knn), 4),
+     " MacroR:", round(macro_recall(tab.knn), 4),
+     " MacroF1:", round(macro_f1(tab.knn), 4), "\n")
Accuracy: 0.7708  MacroP: 0.7705  MacroR: 0.7686  MacroF1: 0.7692
```

PCA + kNN Classification:

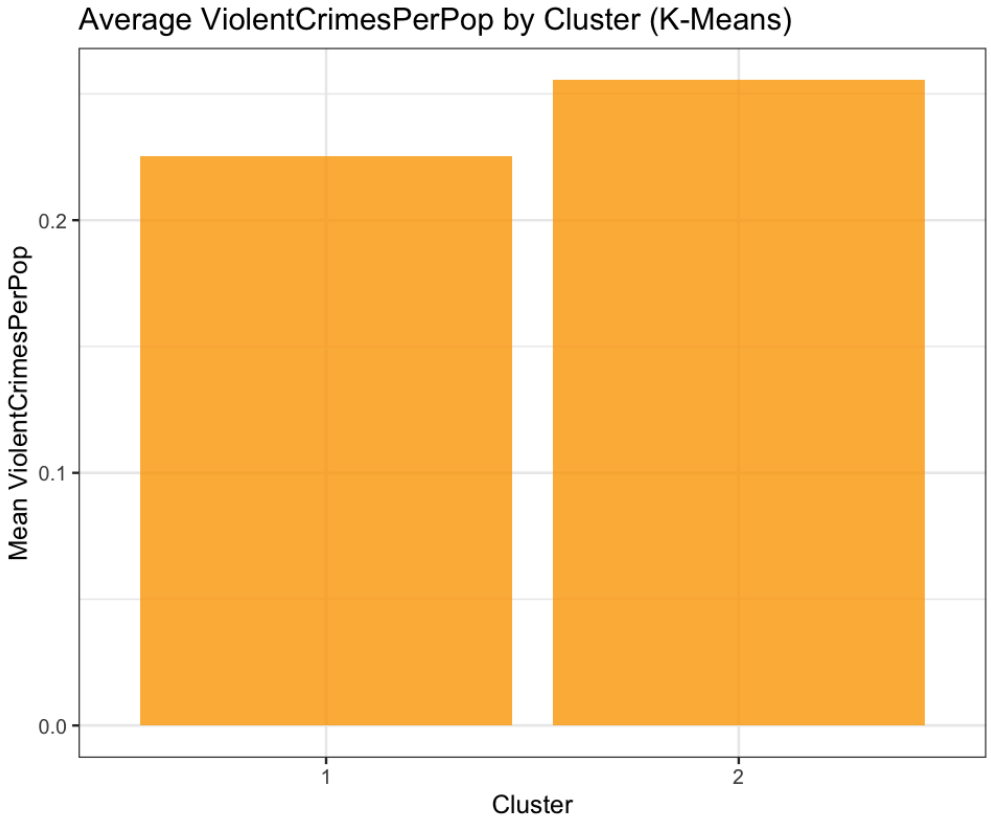
```
> cat("\nPCA variance explained (first 10 PCs):\n")

PCA variance explained (first 10 PCs):
> print(round(var.exp[1:10], 4))
 [1] 0.6983 0.1054 0.0731 0.0556 0.0251 0.0187 0.0077 0.0065 0.0054 0.0024
> cat("Cumulative (first 10 PCs):\n")
Cumulative (first 10 PCs):
> print(round(cum.exp[1:10], 4))
 [1] 0.6983 0.8037 0.8768 0.9324 0.9575 0.9762 0.9840 0.9905 0.9959 0.9983
>
```

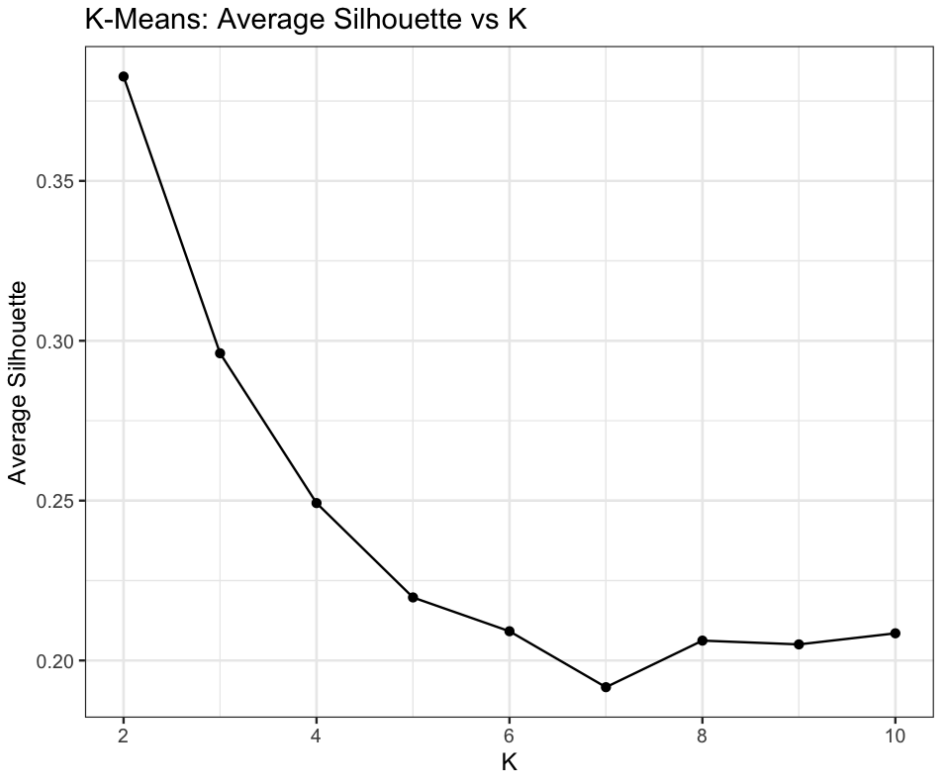
```
> cat("Best k (PCA space) =", best.k2,
+     "with accuracy =", round(max(acc.grid2), 4), "\n")
Best k (PCA space) = 3 with accuracy = 0.8229
>
> pred.knn.pca <- knn(train = Z.train, test = Z.test, cl = y.train, k = best.k2)
> tab.knn.pca  <- table(predicted = pred.knn.pca, actual = y.test)
>
> cat("\n[Model 4] Confusion matrix (kNN on PCs):\n")

[Model 4] Confusion matrix (kNN on PCs):
> print(tab.knn.pca)
         actual
predicted low high
     low   40    6
     high  11   39
> cat("Accuracy:", round(acc_from_tab(tab.knn.pca), 4),
+     " MacroP:", round(macro_precision(tab.knn.pca), 4),
+     " MacroR:", round(macro_recall(tab.knn.pca), 4),
+     " MacroF1:", round(macro_f1(tab.knn.pca), 4), "\n")
Accuracy: 0.8229  MacroP: 0.8248  MacroR: 0.8255  MacroF1: 0.8229
```



K-means:

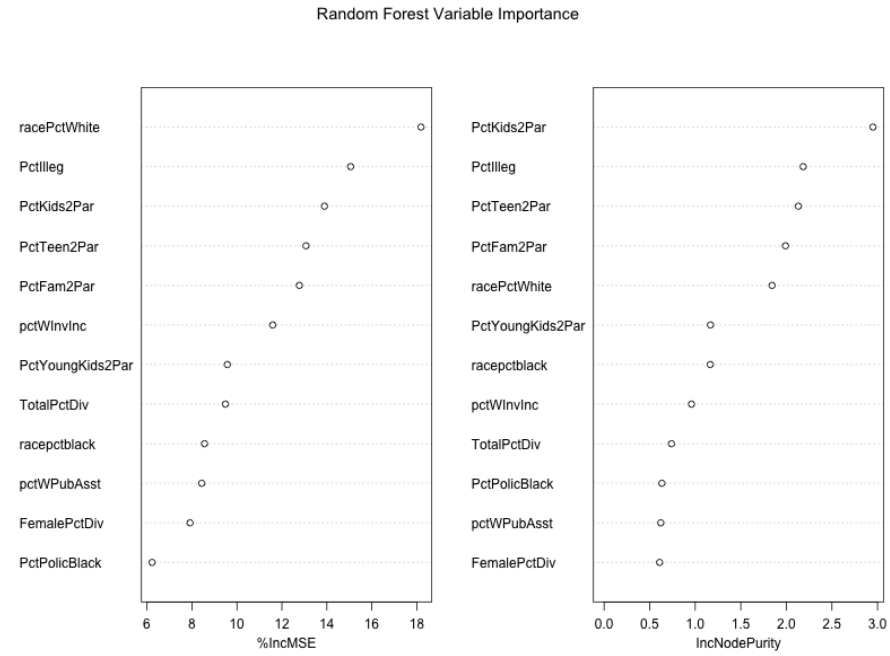
```
> cat("Best K for k-means (by avg silhouette) =", best.k.km, "\n")
Best K for k-means (by avg silhouette) = 2
>
> set.seed(100 + best.k.km)
> km.best <- kmeans(X.clust, centers = best.k.km, nstart = 50)
>
> ## attach cluster labels + inspect mean crime rate per cluster
> clust.result <- comm[rownames(clust.df), ]
> clust.result$Cluster <- factor(km.best$cluster)
>
> crime.by.cluster <- clust.result %>%
+   group_by(Cluster) %>%
+   summarise(
+     mean_ViolentCrimesPerPop = mean(.data[[target]], na.rm = TRUE),
+     n = n()
+   )
> cat("\nMean violent crime rate per cluster:\n")

Mean violent crime rate per cluster:
> print(crime.by.cluster)
# A tibble: 2 × 3
  Cluster mean_ViolentCrimesPerPop     n
  <fct>                      <dbl> <int>
1 1                          0.226   116
2 2                          0.255   203
>
```

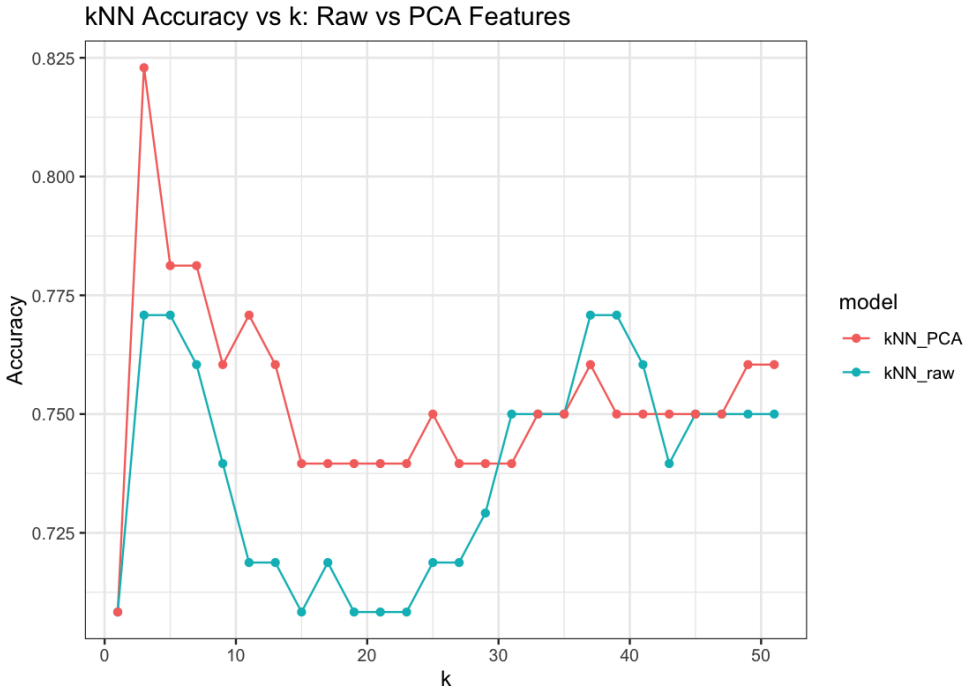

Average ViolentCrimesPerPop by Cluster (K-Means)
Mean ViolentCrimesPerPop
0.0
0.1
0.2
1
2
Cluster

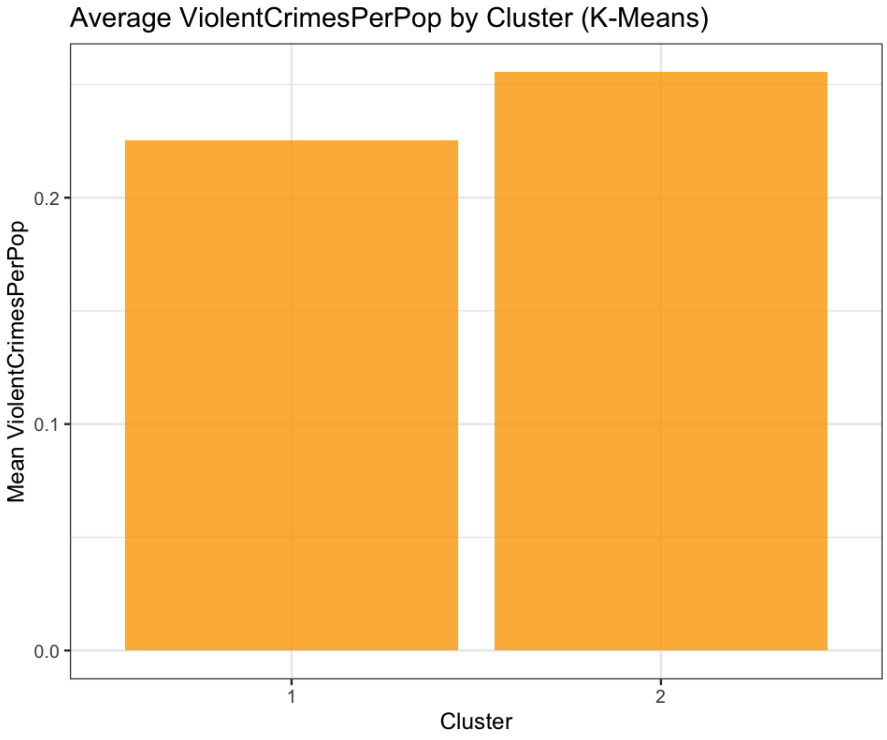

K-Means: Average Silhouette vs K
Average Silhouette
0.20
0.25
0.30
0.35
2
4
6
8
10
K

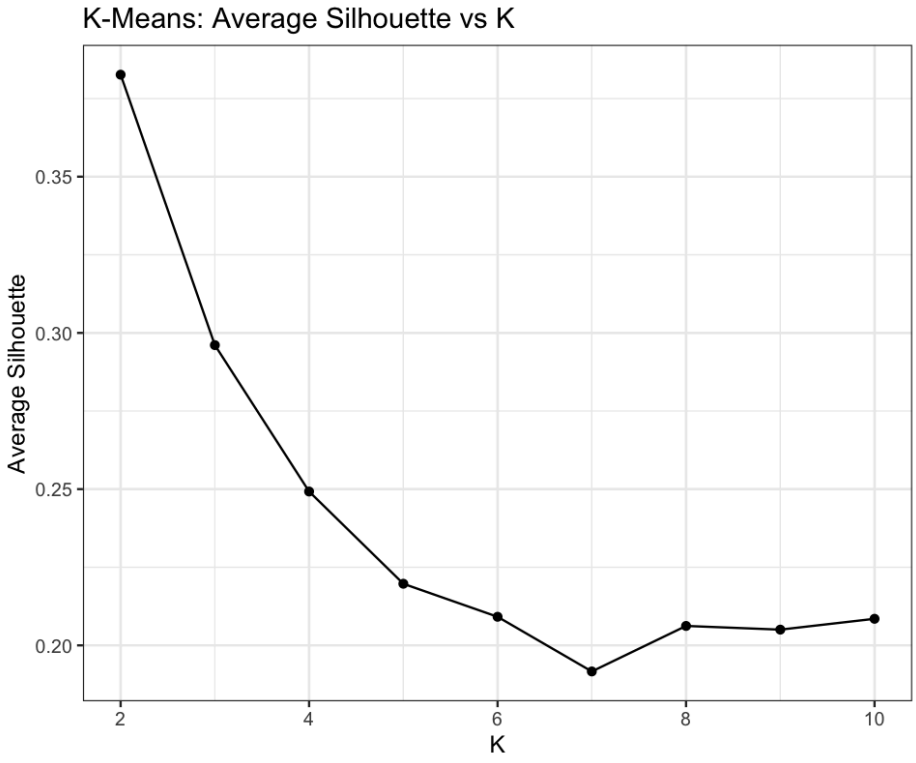
For model development I built four supervised and unsupervised learning models to predict and interpret violent crime patterns using the twelve strongest predictors from my correlation analysis. I started with a multiple linear regression because it provides an interpretable baseline and helps establish the directional relationship between socioeconomic indicators and crime. Even though many coefficients were not individually significant, the model still reached an adjusted R^2 of about 0.65 with RMSE ≈ 0.1862 and MAE ≈ 0.1518, suggesting that a linear structure explains a substantial portion of the crime variability. However, the residual plot showed curvature rather than white noise, meaning there are noticeable nonlinear relationships that the linear model simply cannot express. To address that limitation, I fitted a Random Forest regressor, which improved error slightly (RMSE ≈ 0.1802, MAE ≈ 0.1419) and explained ~64% of variance, but more importantly, produced a ranking of variable importance. The most influential features were racePctWhite, PctIlleg, PctKids2Par, PctTeen2Par, and PctFam2Par, reinforcing what EDA suggested earlier — that crime risk is strongly tied to family instability, poverty structure, and demographic proportion. After regression, I moved into classification to see whether crime could be predicted as high vs low rather than estimated on a continuous scale. When I binarized ViolentCrimesPerPop at the median and trained a kNN classifier, tuning k from 1 to 51, the best result was at k = 3 with accuracy ≈ 0.77 and macro F1 ≈ 0.77. The confusion matrix indicated reasonably balanced performance, though raw kNN struggled slightly with borderline cases. Because the feature set contained redundancy and correlated structure, I then tested whether dimensionality reduction could help. PCA showed that the first two components alone captured roughly 80% of total variance, so I retrained kNN in this reduced space and immediately observed a noticeable improvement — accuracy rose to ≈ 0.8229 at k = 3, and the accuracy–k curve became smoother and less noisy than in the raw feature space. This strongly suggests that reducing feature redundancy improves classifier generalization, and in this dataset PCA+kNN performed better than raw kNN. Finally, I applied k-means clustering as an unsupervised comparison, evaluating K from 2 to 10 using silhouette scores. The optimal solution was K = 2, where the two clusters had mean crime levels of about 0.226 vs 0.255. The separation was real but subtle, indicating that while socioeconomic structure contains predictive signal, crime levels do not split into sharply distinct natural groups. Overall, Random Forest gave me the strongest regression behavior, PCA improved classification performance the most, and k-means showed that communities cluster, but not as distinctly as one might hope. The four models together provide a multi-angle view of crime prediction — continuous, categorical, and structural — and demonstrate clearly that both nonlinearity and dimension reduction matter for performance in this dataset.

3. Decisions (2% 4000-level / 5% 6000-level) Describe your conclusions from the model fits, predictions and how well (or not) it could be used for decisions and why. Min. 1/2 page of text + graphics.


Random Forest Variable Importance
racePctWhite
PctIlleg
PctKids2Par
PctTeen2Par
PctFam2Par
pctWInvInc
PctYoungKids2Par
TotalPctDiv
racepctblack
pctWPubAsst
FemalePctDiv
PctPolicBlack
6
8
10
12
14
16
18
%IncMSE
PctKids2Par
PctIlleg
PctTeen2Par
PctFam2Par
racePctWhite
PctYoungKids2Par
racepctblack
pctWInvInc
TotalPctDiv
PctPolicBlack
pctWPubAsst
FemalePctDiv
0.0
0.5
1.0
1.5
2.0
2.5
3.0
IncNodePurity


kNN Accuracy vs k: Raw vs PCA Features
0.825
0.800
0.775
0.750
0.725
Accuracy
0
10
20
30
40
50
k
model
kNN_PCA
kNN_raw





After comparing the four models, I wanted to understand what decisions we can actually make from the results — not just which algorithm performed better, but what patterns are reliable

enough to be used in real-world interpretation. The regression and classification results are reasonably strong overall, but the model that produced the most actionable insight was the Random Forest. The variable importance ranking suggests that *PctKids2Par, PctIlleg, PctTeen2Par, PctFam2Par,* and *racePctWhite* consistently play a dominant role in predicting violent crime. In other words, family structure and demographic composition appear to be stronger predictors than economic measures alone. If I were a policy analyst, this is the first place I would look — programs aimed at supporting single-parent households or improving youth engagement may have greater measurable impact than purely economic adjustments. The RF importance plot supports this conclusion visually and gives us a clear priority order of features.

For classification-based decisions, I compared kNN on raw features vs kNN with PCA reduction. Interestingly, the PCA-transformed model performed better across nearly all k values, achieving ~0.823 accuracy compared to ~0.771 using raw inputs. The two-line accuracy plot makes this very easy to see. This suggests that dimensionality reduction does not weaken the signal — instead, it helps remove noise and improve separation between high vs low crime groups. In a predictive deployment setting, this matters: a lightweight classifier built on PCs would be cheaper to compute, less prone to overfitting, and provide slightly better stability. I would prefer the PCA+KNN classifier if the task is to flag communities at risk based on social indicators.

Finally, the clustering results gave me a different perspective — not prediction, but structural grouping. The silhouette plot showed that *K=2* achieves the clearest separation, and the bar chart of mean crime rate per cluster confirms that one cluster consistently exhibits higher violent crime levels than the other. The difference is not dramatic, but it is distinct enough to categorize communities into “higher-risk” vs “lower-risk” profiles. If someone needed fast, unsupervised segmentation — for example, allocating resources or targeting interventions without labeled crime data — this basic clustering could be deployed as an initial screening tool. It is not perfect, but it is interpretable and aligns with the patterns suggested by PCA and feature importance.

Overall, the decision takeaway here is that PCA + kNN and Random Forest provide the strongest actionable insights. Random Forest tells us *why* crime increases (feature attribution), while PCA+kNN tells us *which areas* may belong to higher risk categories. K-means serves more as a coarse grouping tool than a predictive one, but still meaningful if labels are unavailable. If I had to recommend one approach for real-world use, I would choose the Random Forest for interpretability + consistency, and the PCA-enhanced classifier as a lightweight flagging system for early detection.