Task 1 ---- Boston data in the ISLR2 package

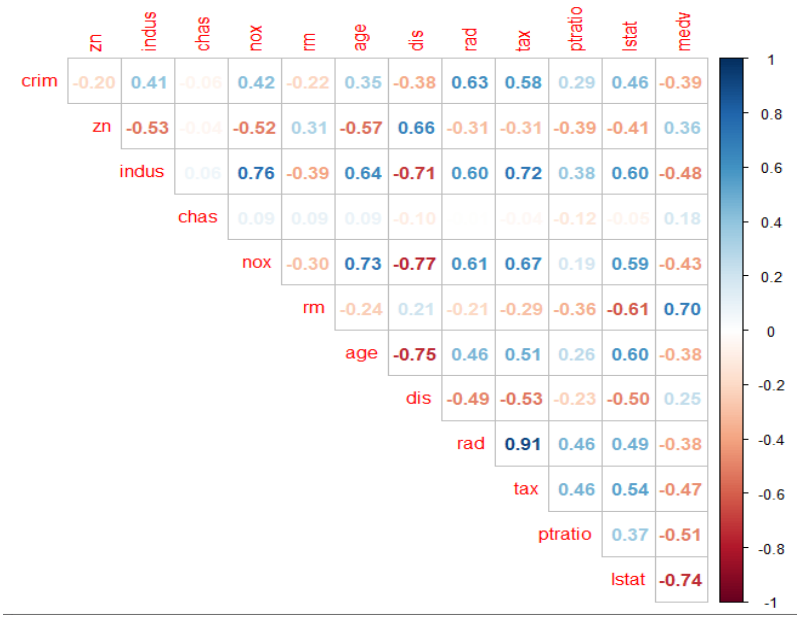
No null or missing data is there in Boston data of ISLR2 package

```
> sum(is.na(Boston))
[1] 0
>
> I
>
> is.null(Boston)
[1] FALSE
> |
```

No Duplicate Data in Boston:

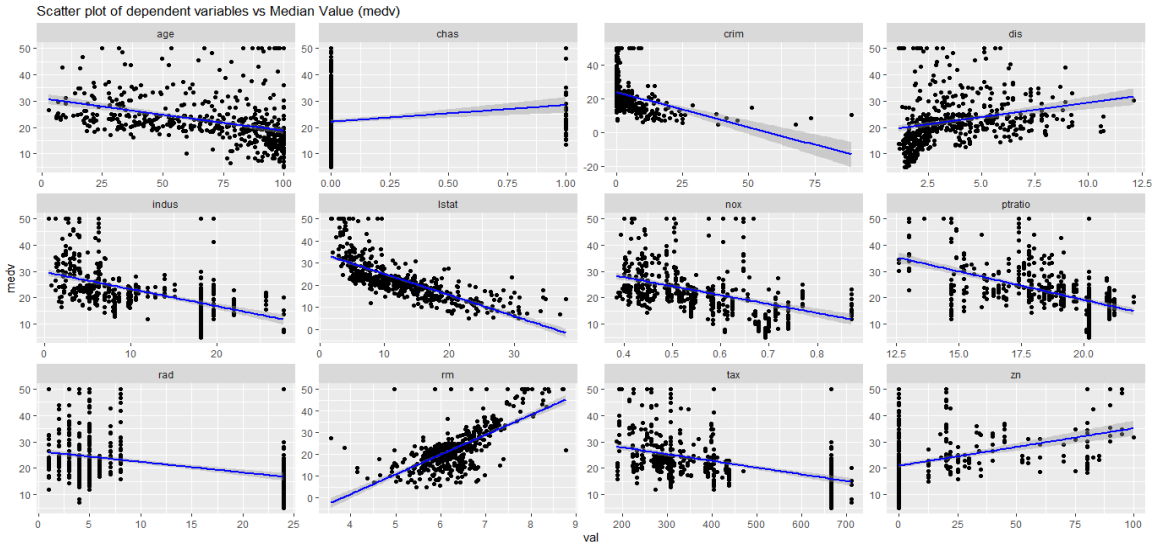
```
> sum(duplicated(Boston))
[1] 0
> |
```

Data Exploration

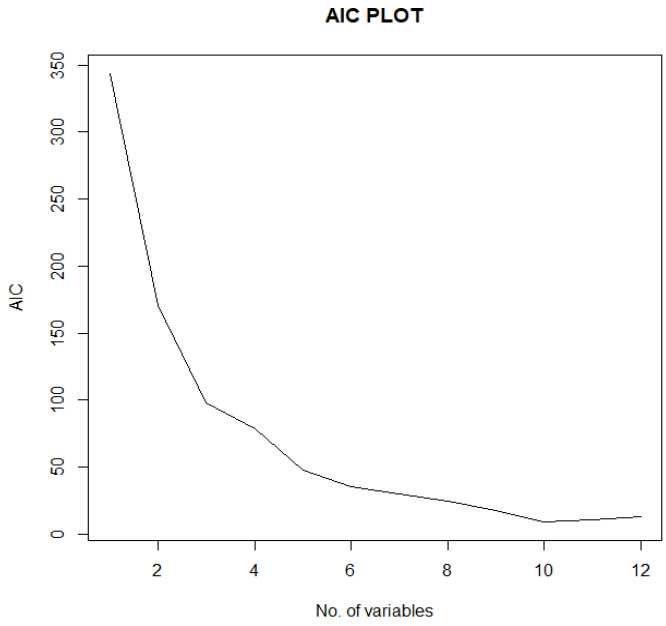



- Median value of owner-occupied homes (in 1000$) increases as average number of rooms per dwelling increases, and it decreases if percent of lower status population in the area increases
- nox or nitrogen oxides concentration (ppm) increases with increase in proportion of non-retail business acres per town and proportion of owner-occupied units built prior to 1940.
- rad and tax have a strong positive correlation of 0.91 which implies that as accessibility of radial highways increases, the full value property-tax rate per $10,000 also increases.
- crim is strongly associated with variables rad and tax which implies as accessibility to radial highways increases per capita crime rate increases.
- indus has strong positive correlation with nox, which supports the notion that nitrogen oxides concentration is high in industrial areas.

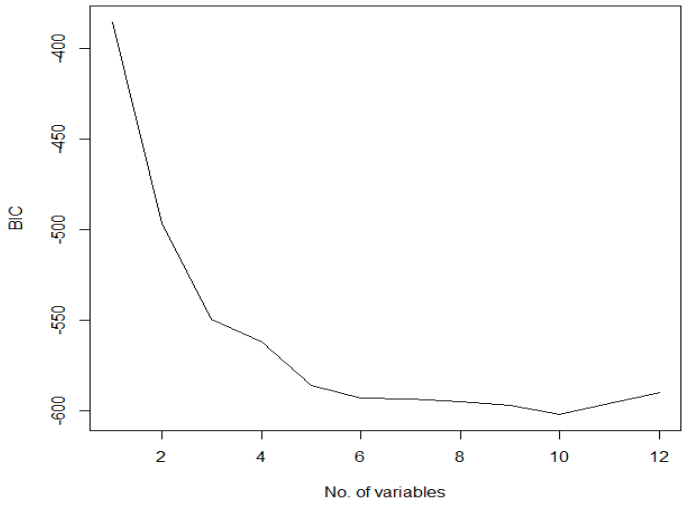
Data Visualization



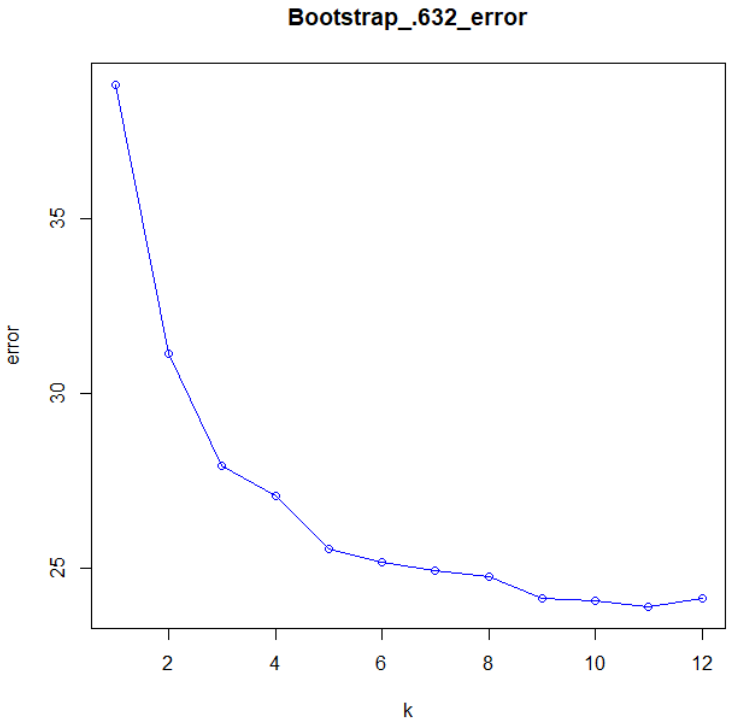

**Observations** * Proportion of owner-occupied units built prior to 1940 (age) and proportion of blacks by town (black) is heavily skewed to left, while per capita crime rate in town (crim) and weighted mean of distances to five Boston employment centres (dis) is heavily skewed to right. * rm is normally distributed with mean of approximately 6. * Most of the properties are situated close to the five Boston employment centres (dis skewed to right) * There is a high proportion of owner-occupied units built prior to 1940 (age skewed to left) and blacks in town (black skewed to right) * From scatter plots, it is seen that lstat and rm show strong correlation with medv. * 93% of the properties are away from Charles River. The properties bordering the river seems to have higher median prices.



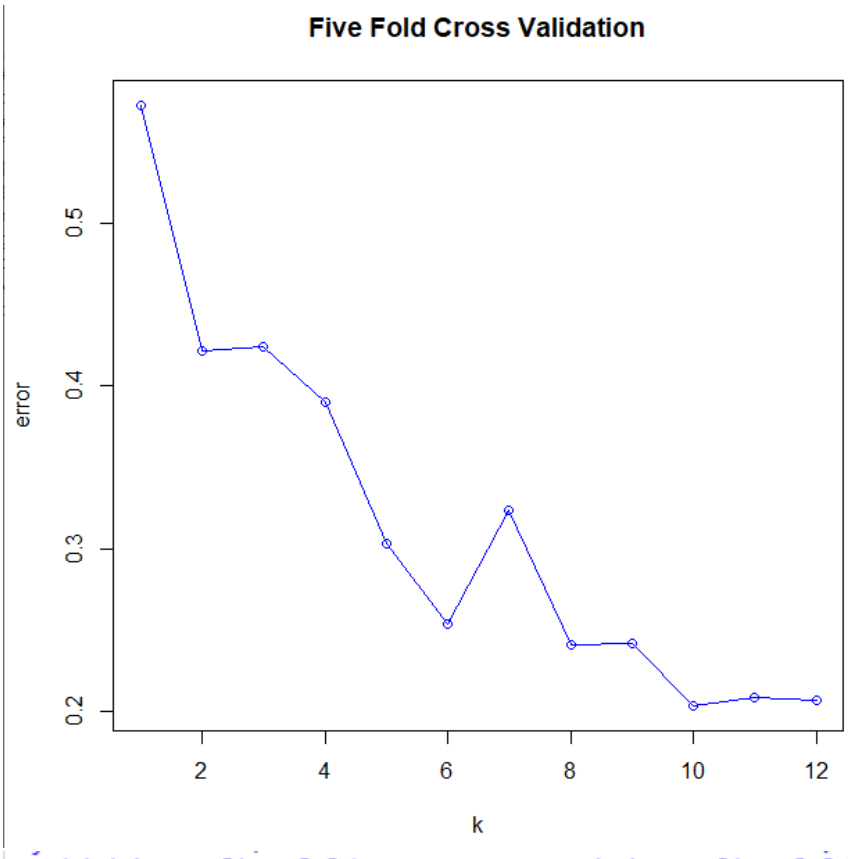

```
> which(my_sum$cp == min(my_sum$cp))
[1] 10
>
```



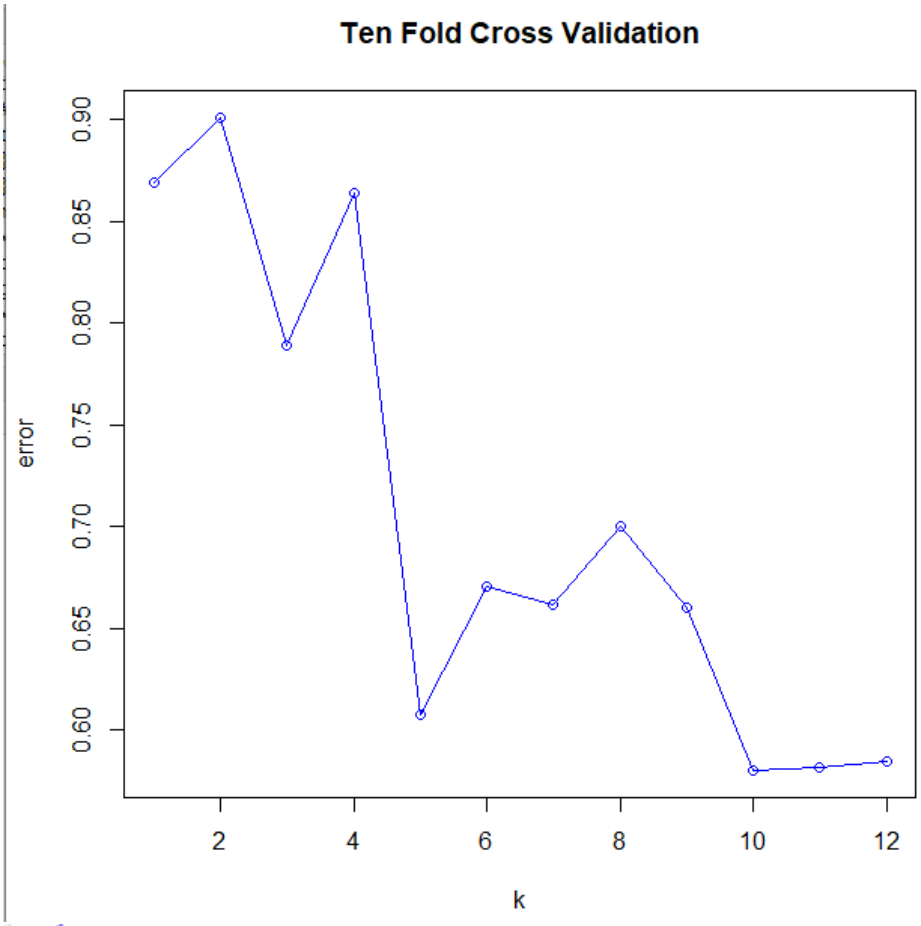

```
> which(my_sum$bic == min(my_sum$bic))
[1] 10
>
```




```
> which(Bootstrap_.632_error == min(Bootstrap_.632_error))
[1] 11
>
```




```
> which(mean_five_fold_cv_errors == min(mean_five_fold_cv_errors))
10
```




```
> which(mean_ten_fold_cv_errors == min(mean_ten_fold_cv_errors))
10
```

From AIC and BIC and cross validation plots it seems mostly the minimum error value is attained for 10 variable model. All of these error selection procedures agree on this point as seen from the above plots that ten variable model is the optimal model .

Task2)Since  crim is a continuous variable so we have to make a new dummy variable to use classifiers. The variable chas is also supposed to be a factor so let's change that before we go on.

Numerical Summary

Having recoded the crim variable into a factor, I make a table that summarizes each variable for each group in the response variable after first normalizing each variable.

| Variable | crime_factor | Q10 | Q25 | median | mean | Q75 | Q90 |
|---|---|---|---|---|---|---|---|
| age | High | -0.234 | 0.470 | 0.846 | 0.613 | 1.042 | 1.116 |
| age | Low | -1.782 | -1.317 | -0.713 | -0.613 | 0.118 | 0.750 |
| black | High | -2.942 | -0.298 | 0.292 | -0.351 | 0.421 | 0.441 |
| black | Low | 0.222 | 0.356 | 0.405 | 0.351 | 0.441 | 0.441 |
| dis | High | -1.108 | -0.970 | -0.786 | -0.616 | -0.355 | 0.098 |
| dis | Low | -0.631 | -0.202 | 0.628 | 0.616 | 1.275 | 1.915 |
| indus | High | -0.720 | -0.180 | 1.015 | 0.603 | 1.015 | 1.231 |
| indus | Low | -1.306 | -1.132 | -0.801 | -0.603 | -0.376 | 0.247 |
| lstat | High | -0.914 | -0.290 | 0.352 | 0.453 | 1.076 | 1.931 |
| lstat | Low | -1.149 | -0.964 | -0.580 | -0.453 | -0.045 | 0.457 |
| medv | High | -1.319 | -0.971 | -0.504 | -0.263 | 0.051 | 0.995 |

Now I apply summarizing functions to the numerical summary to show the absolute differences between groups for each of the summary variables.

Files Plots Packages Help Viewer

Zoom Export Publish

| | Between-Groups Differences | | | | | |
|---|---|---|---|---|---|---|
| Variable | diff_Q10 | diff_Q25 | diff_med | diff_mean | diff_Q75 | diff_Q90 |
| age | 1.548 | 1.787 | 1.560 | 1.227 | 0.924 | 0.366 |
| black | -3.165 | -0.654 | -0.113 | -0.702 | -0.020 | 0.000 |
| dis | -0.477 | -0.768 | -1.414 | -1.231 | -1.630 | -1.817 |
| indus | 0.586 | 0.952 | 1.816 | 1.205 | 1.391 | 0.984 |
| lstat | 0.235 | 0.674 | 0.933 | 0.906 | 1.122 | 1.474 |
| medv | -0.883 | -0.728 | -0.565 | -0.526 | -0.598 | -0.350 |
| nox | 0.844 | 0.975 | 1.510 | 1.445 | 1.597 | 1.645 |
| ptratio | -0.231 | 0.831 | 1.016 | 0.507 | 0.508 | 0.370 |
| rad | 0.230 | 0.230 | 2.297 | 1.238 | 2.182 | 2.067 |
| rm | -0.666 | -0.322 | -0.258 | -0.312 | -0.364 | -0.007 |

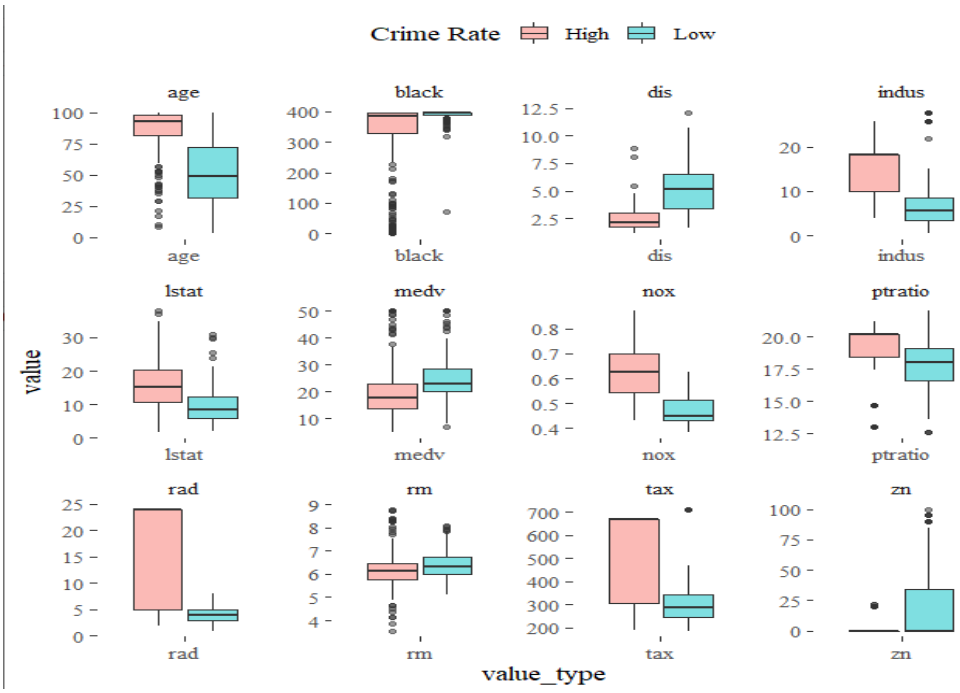
Then I take the mean of each summary variable and arrange them in descending order. What we have now is a table of the mean of the absolute normalized differences in variables between each group.

Files | Plots | Packages | Help | Viewer

Zoom | Export | Publish

| Variable | Absolute Mean Differences |
|---|---|
| rad | 1.374 |
| nox | 1.336 |
| tax | 1.300 |
| age | 1.235 |
| dis | 1.223 |
| indus | 1.156 |
| zn | 0.960 |
| lstat | 0.890 |
| black | 0.776 |
| medv | 0.608 |
| ptratio | 0.500 |

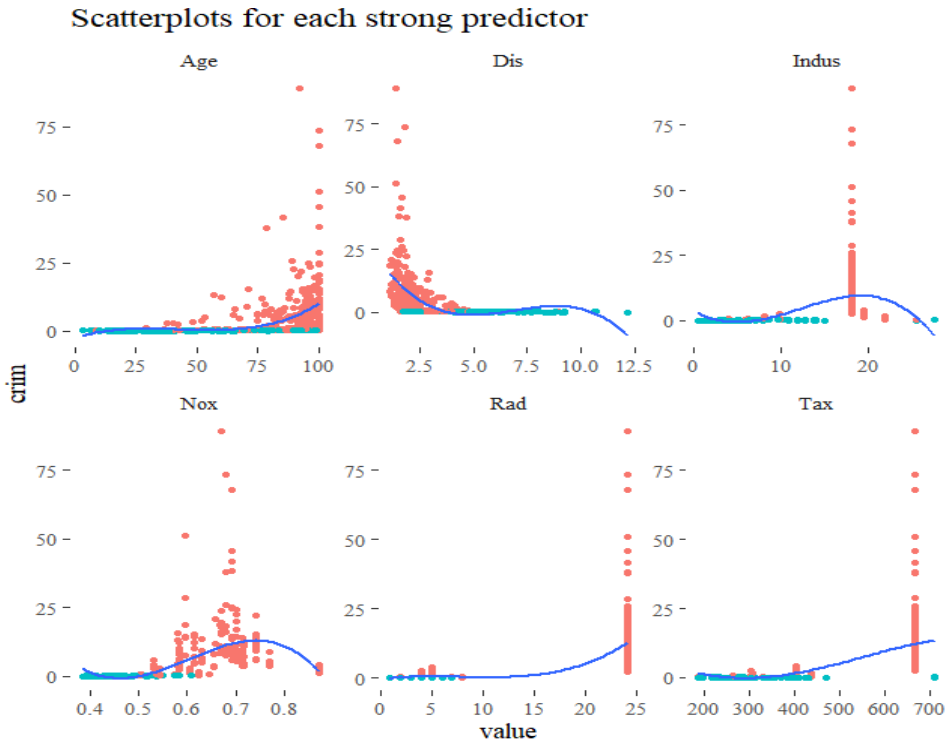
Faceted Boxplots




The following covariates seem to separate the data well on the boxplots:

- rad
- age
- indus
- nox
- tax

These variables are likely to be strong predictors. Now I will plot these variables against the crim variable and color the points according to whether they are above or below the median. This will give us a good visual understanding of the problem.




Confusion matrix after prediction with logistic regression:

| Predicted | Observed | |
|---|---|---|
| | High | Low |
| High | 58 | 14 |
| Low | 6 | 49 |

[a] 0.84251968503937

Confusion Matrix after prediction with LDA

Files Plots Packages Help Viewer

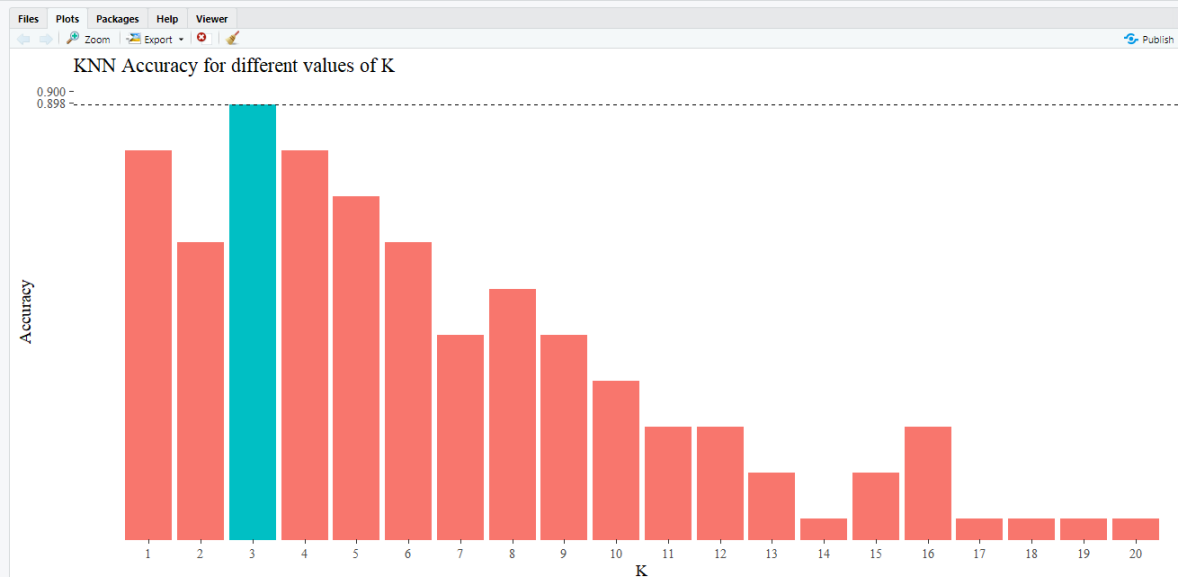
Zoom Export Publish

| Predicted | Observed | |
|---|---|---|
| | High | Low |
| High | 49 | 1 |
| Low | 15 | 62 |

[a] 0.874015748031496

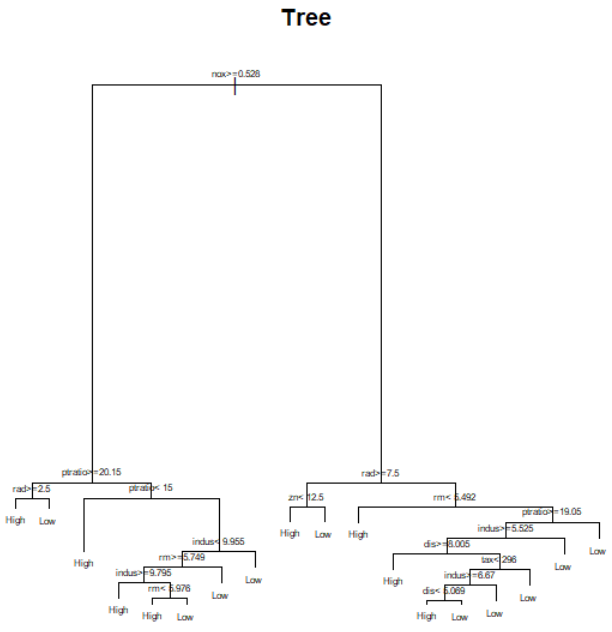
The LDA models do well, better than the logistic regression model.

Performance with Knn for different k values



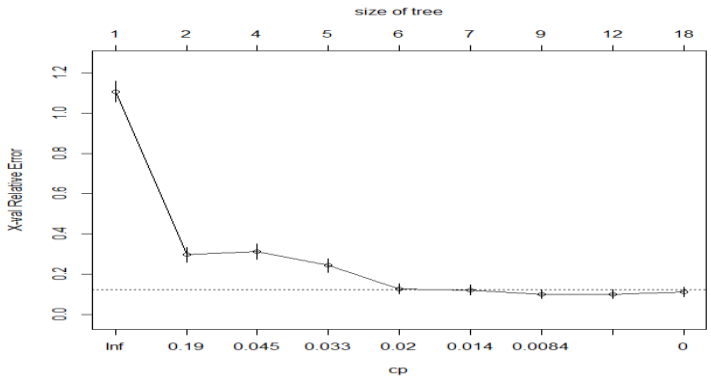

KNN achieves a best accuracy of 0.913 with K = 1.

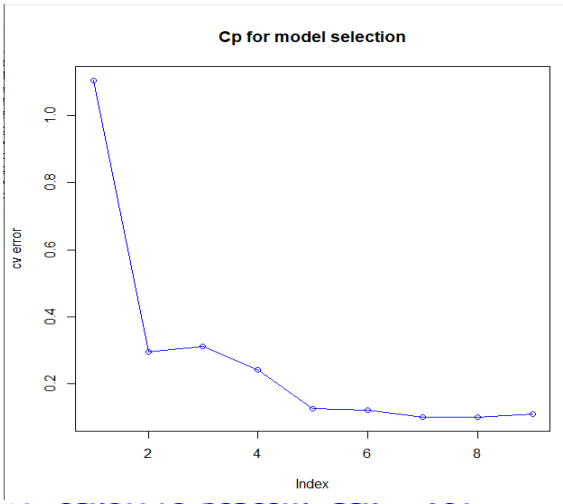
Training on CART model – Full Tree Model Represenation



```
> fit_boston$cptable
           CP nsplit  rel error    xerror       xstd
1 0.724867725      0 1.00000000 1.1058201 0.05122904
2 0.047619048      1 0.27513228 0.2962963 0.03655223
3 0.042328042      3 0.17989418 0.3121693 0.03734389
4 0.026455026      4 0.13756614 0.2433862 0.03363718
5 0.015873016      5 0.11111111 0.1269841 0.02508640
6 0.013227513      6 0.09523810 0.1216931 0.02459277
7 0.005291005      8 0.06878307 0.1005291 0.02247743
8 0.001763668     11 0.05291005 0.1005291 0.02247743
9 0.000000000     17 0.04232804 0.1111111 0.02356513
> |
```

The CP table as well as the CP table plot below indicates that we can reduce the actual tree model complexity of nsplits=8 where the cross validation error is minimum.

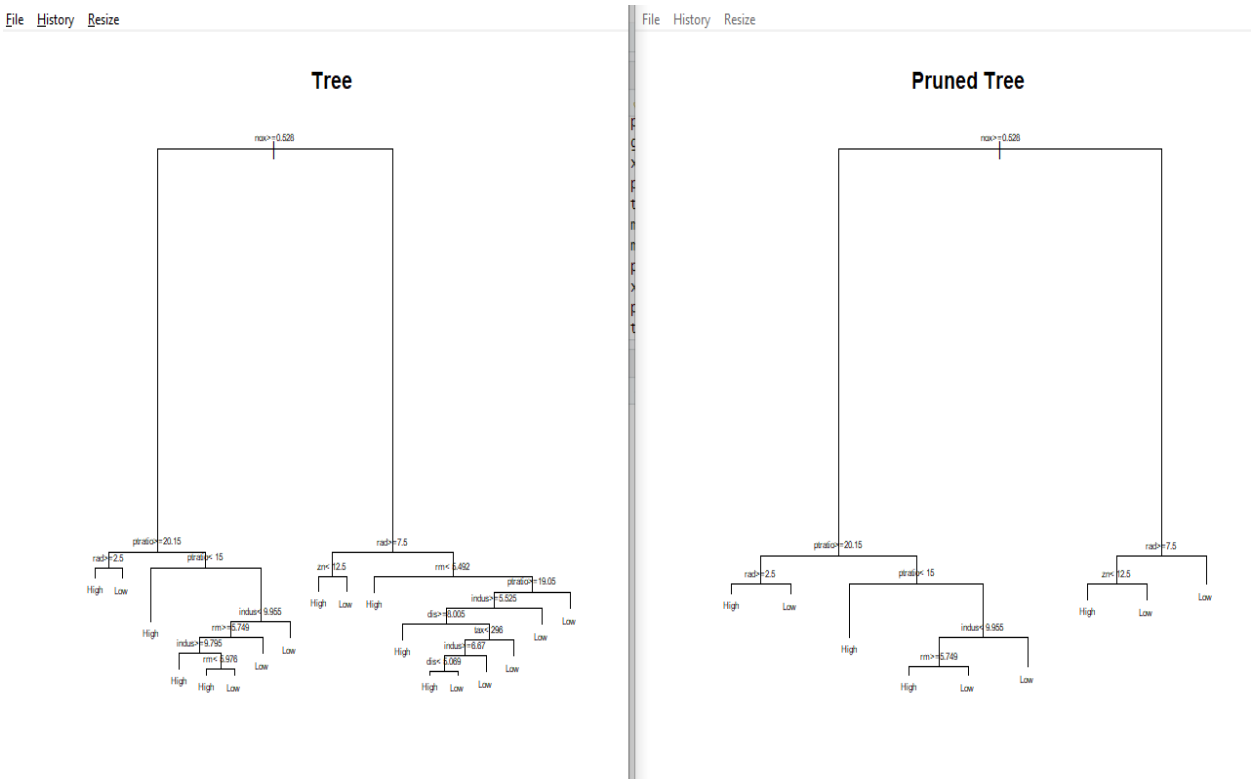



```
> min_cp = which.min(fit_boston$cptable[,4])
> min_cp
7
7
> |
```

The above graph is a representation of the cp table which provides the cross-validation information of the fitted tree model on Boston Train Dataset, and the information that we get from the above graph is that we have a minimum cross validation error at index 7 which corresponds to the 8$^{th}$ split in the cp table hence we can prune the model for a overall size of 8 splits.

Full Tree Model




Error acquired for Full Tree Model

```
> error<-length(x)/length(predicted[,3])
> error
[1] 0.03937008
>
```

Error acquired for Pruned  Tree Model

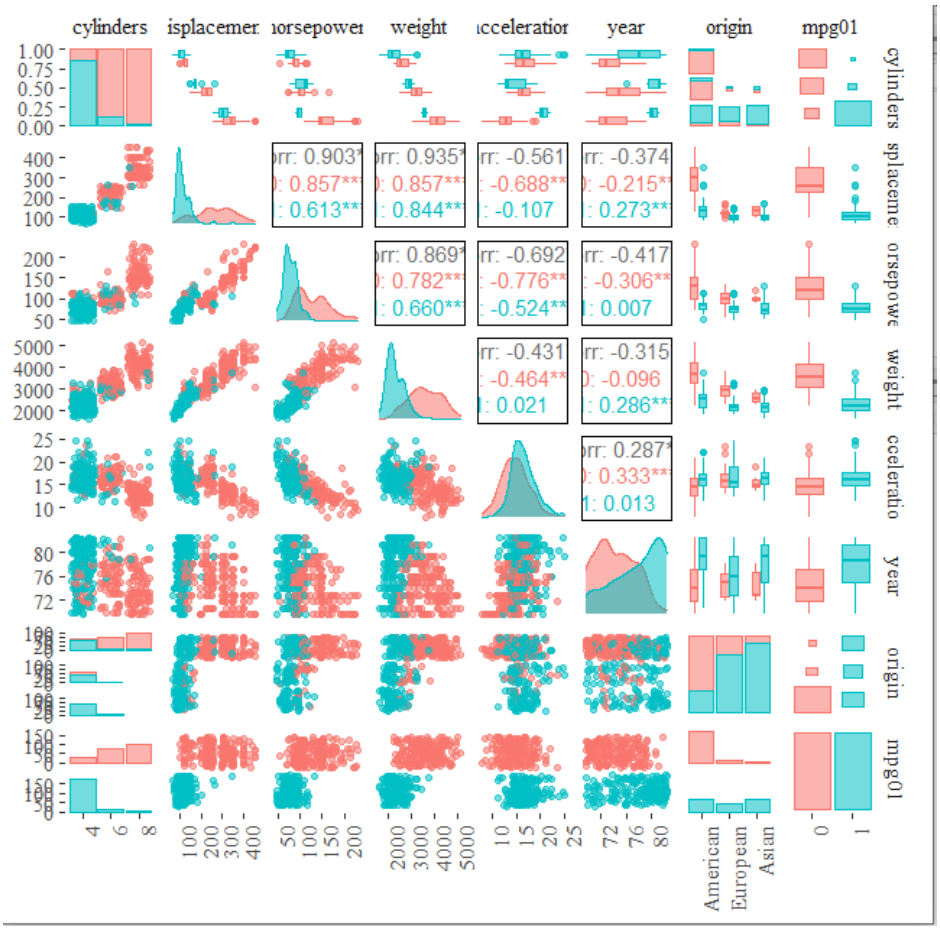
```
> y_true_test <- test$crime_factor
> x<- which(predicted_prune_tree[,3]!=y_true_test)
> error<-length(x)/length(predicted_prune_tree[,3])
> error
[1] 0.04724409
> 
```

It seems both Pruned Tree and Full Tree Model are performing good with an accuracy of 0.95 and 0.96 on test data set respectively. KNN lags behind Tree Models but got a accuracy of 0.913 for k=1 on test data set which is quite ok. LDA and Logistic Regression both lags behind KNN.

Task3)

a)Created a binary variable mpg01

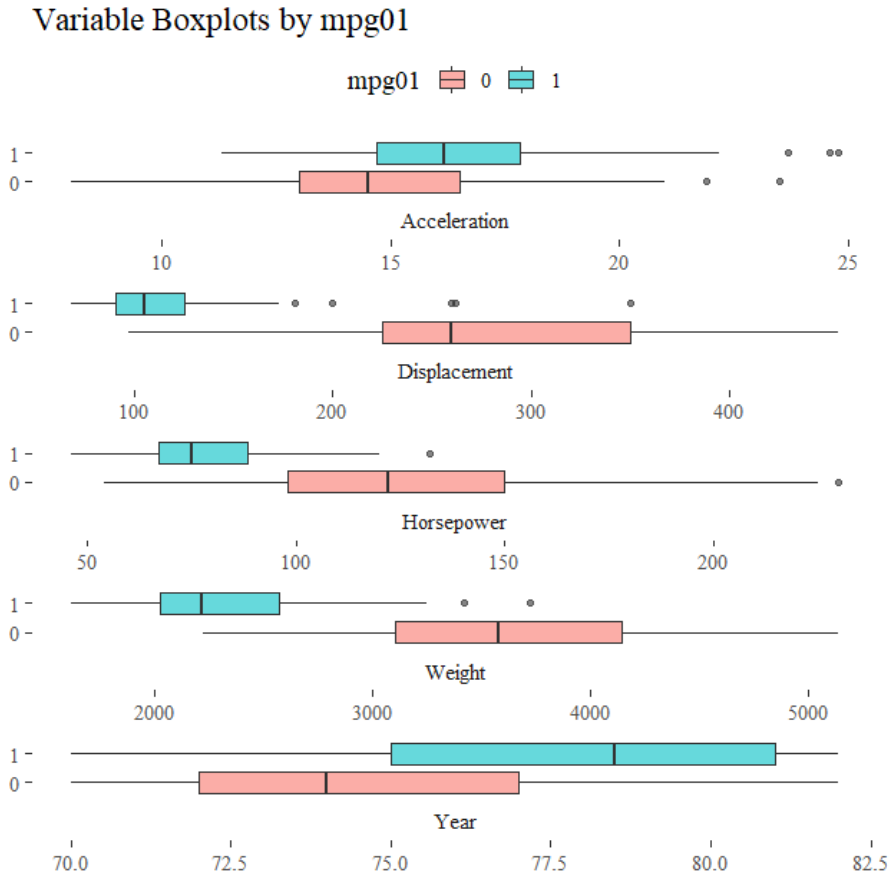
b)Graphical Exploration of Auto Data




From the faceted ggpairs plot it looks like most of the variables separate our target well. The best separators seem to be:

- cylinders
- displacement

- horsepower
- weight
- year




c)Splitting the data into test and train

d) Test error for LDA Model

```
> fmla <- as.formula('mpg01 ~ displacement + horsepower + weight + year + cylind
ers')
> lda_model <- lda(fmla, data = training)
>
> pred <- predict(lda_model, testing)
> table(pred$class, testing$mpg01)

    0  1
  0 48  5
  1  5 39
> mean(pred$class != testing$mpg01)
[1] 0.1030928
>
```

The LDA model does well. The prediction error is 0.103 for test data

e) Test error for QDA Model

```
> qda_model <- qda(fmla, data = training)
>
> pred <- predict(qda_model, testing)
> table(pred$class, testing$mpg01)

    0  1
  0 48  5
  1  5 39
> mean(pred$class != testing$mpg01)
[1] 0.1030928
> |
```

The QDA model performs same as LDA

f)Test error for logistic regression

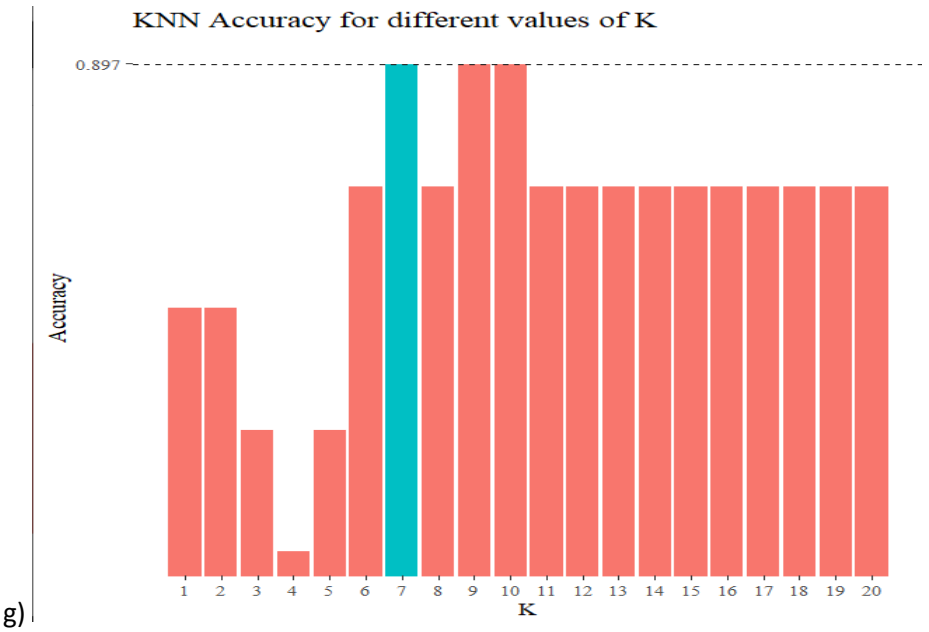
```
> log_reg <- glm(fmla, data = training, family = binomial)
>
> pred <- predict(log_reg, testing, type = 'response')
> pred_values <- round(pred)
> table(pred_values, testing$mpg01)

pred_values  0  1
           0 49  3
           1  4 41
> mean(pred_values != testing$mpg01)
[1] 0.07216495
> |
```

The logistic regression performs better than both LDA and QDA

h)

g)




The KNN model performs same as LDA and QDA with an accuracy of 0.897 and misclassification error of 0.103 for k=7

KNN error for different values of k: