

# Gas Mileage Prediction

Aymen Rumi

5/5/2020

## Overview

We will predict whether a given car gets high or low gas mileage given data from Auto dataset

```
Auto<-Auto%>%mutate(mileage=ifelse(mpg > median(mpg),1,0))
Auto$mileage<-as.factor(Auto$mileage)
attach(Auto)
```

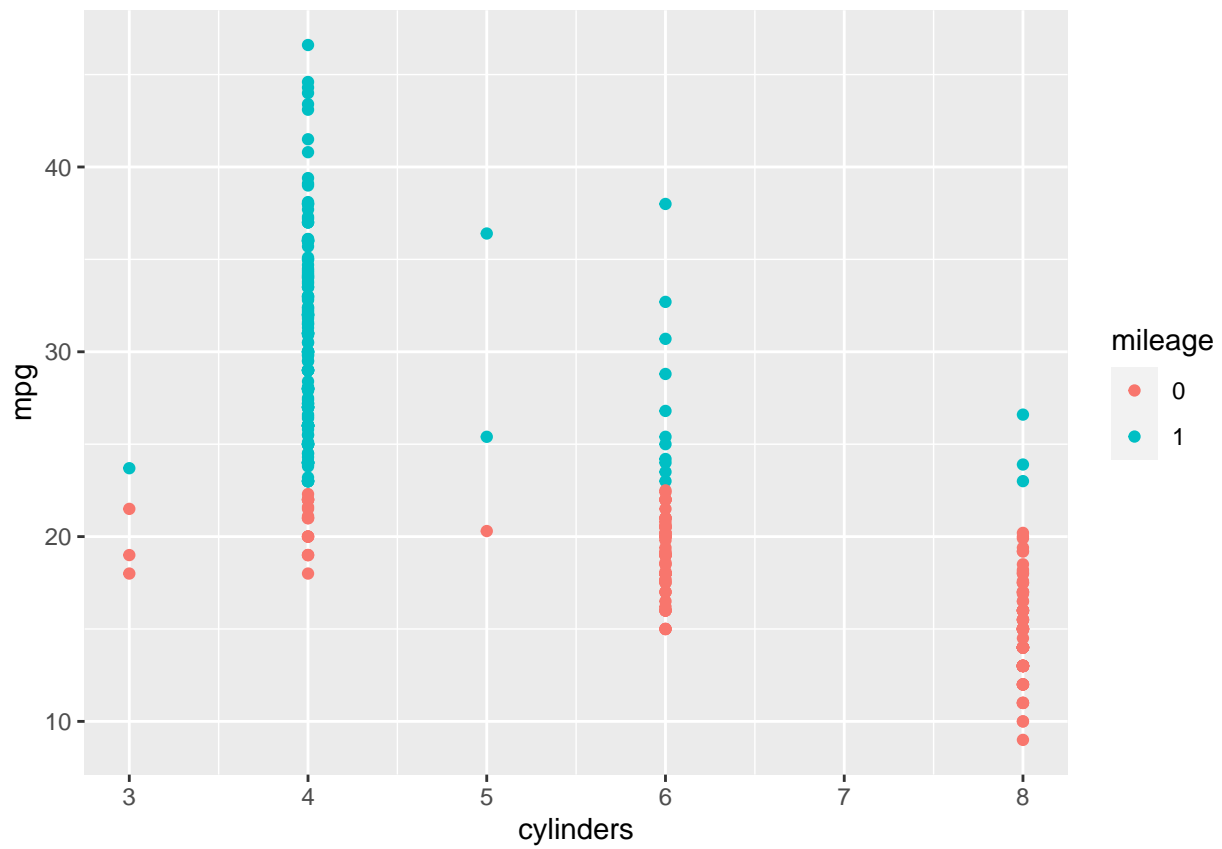
The following object is masked from package:ggplot2:

mpg

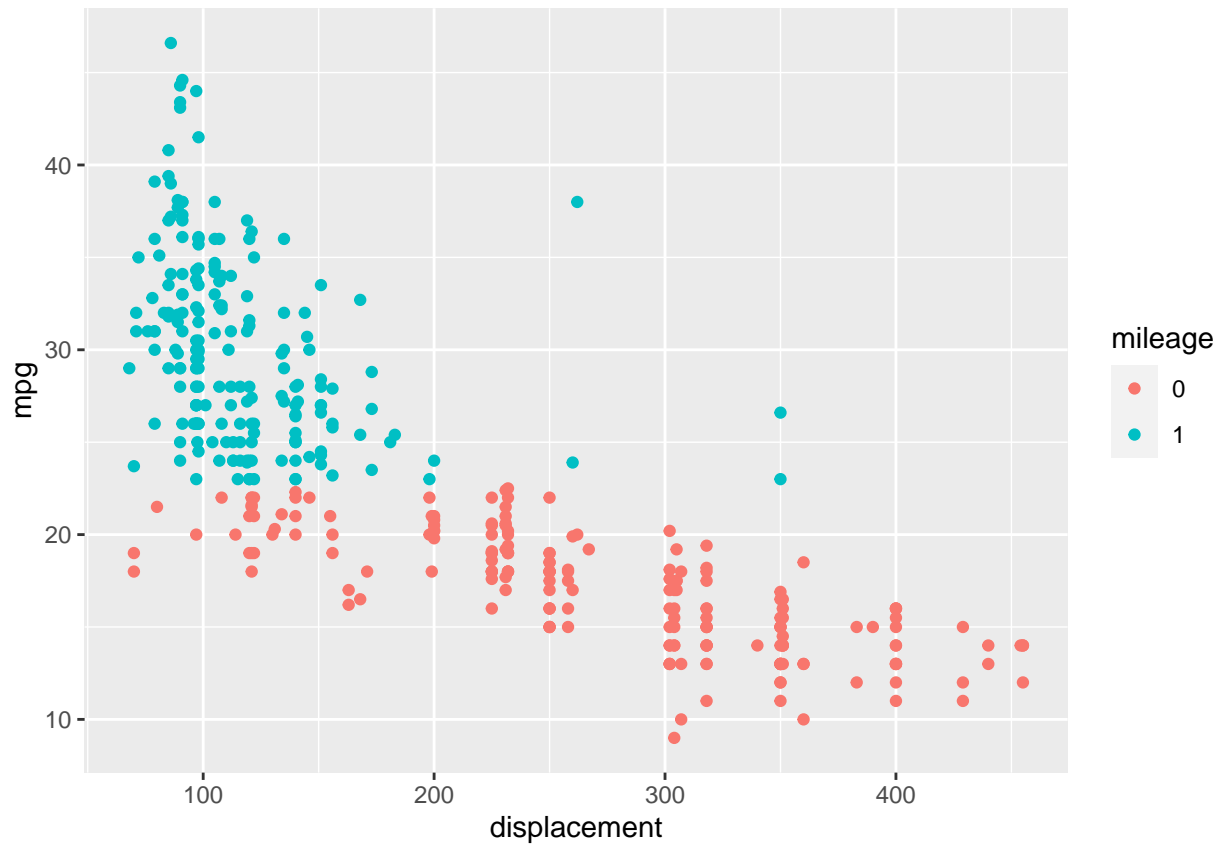
```
names(Auto)
```

```
[1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"
[6] "acceleration" "year"         "origin"       "name"         "mileage"
```

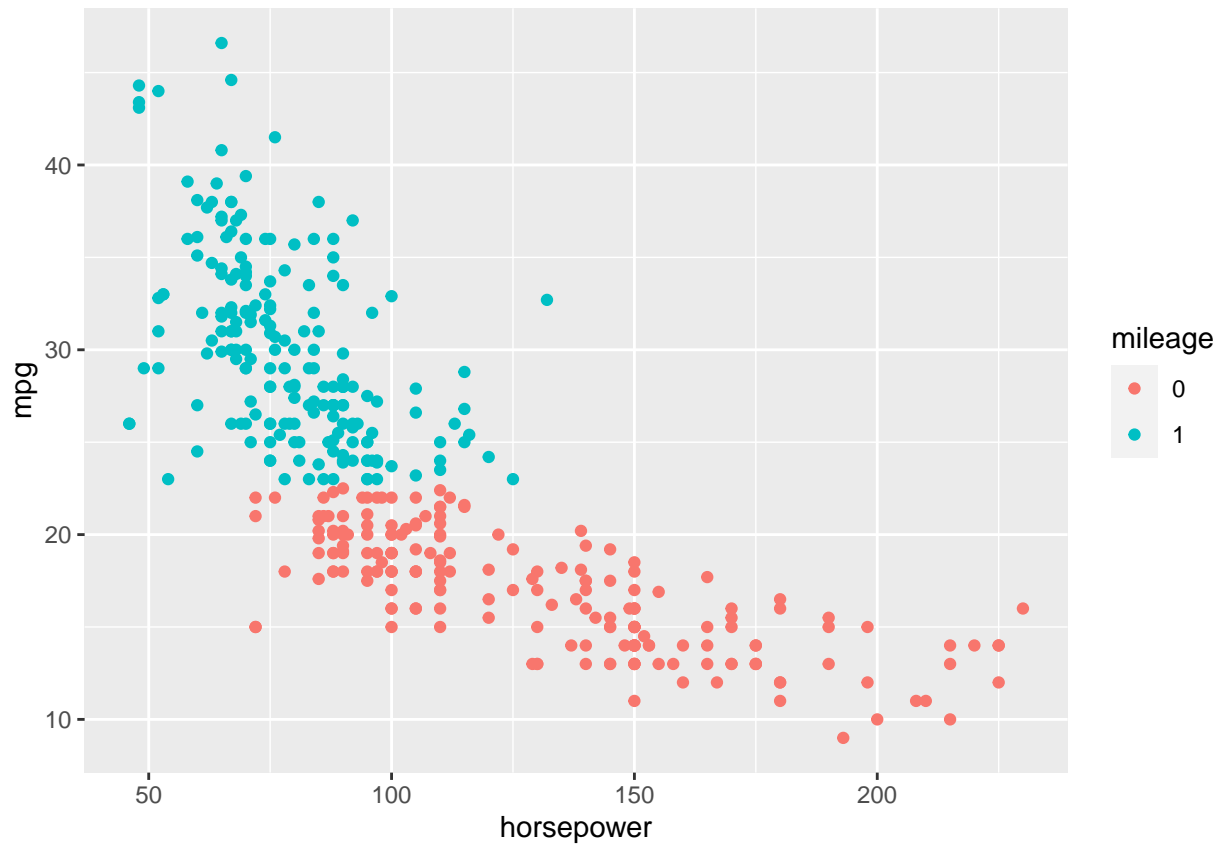
```
ggplot(data=Auto,aes(y=mpg,x=cylinders,color=mileage))+geom_point()
```



```
ggplot(data=Auto,aes(y=mpg,x=displacement,color=mileage))+geom_point()
```



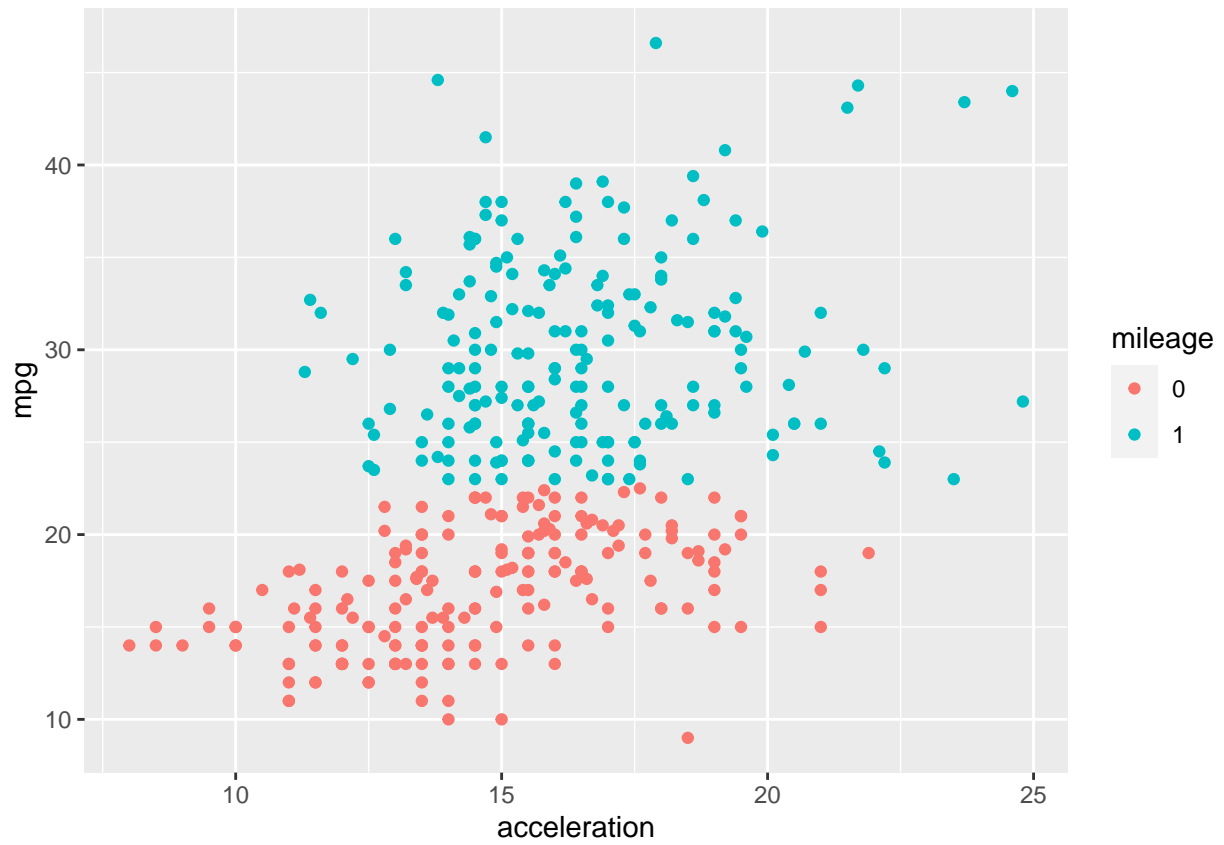
```
ggplot(data=Auto,aes(y=mpg,x=displacement,color=mileage))+geom_point()
```



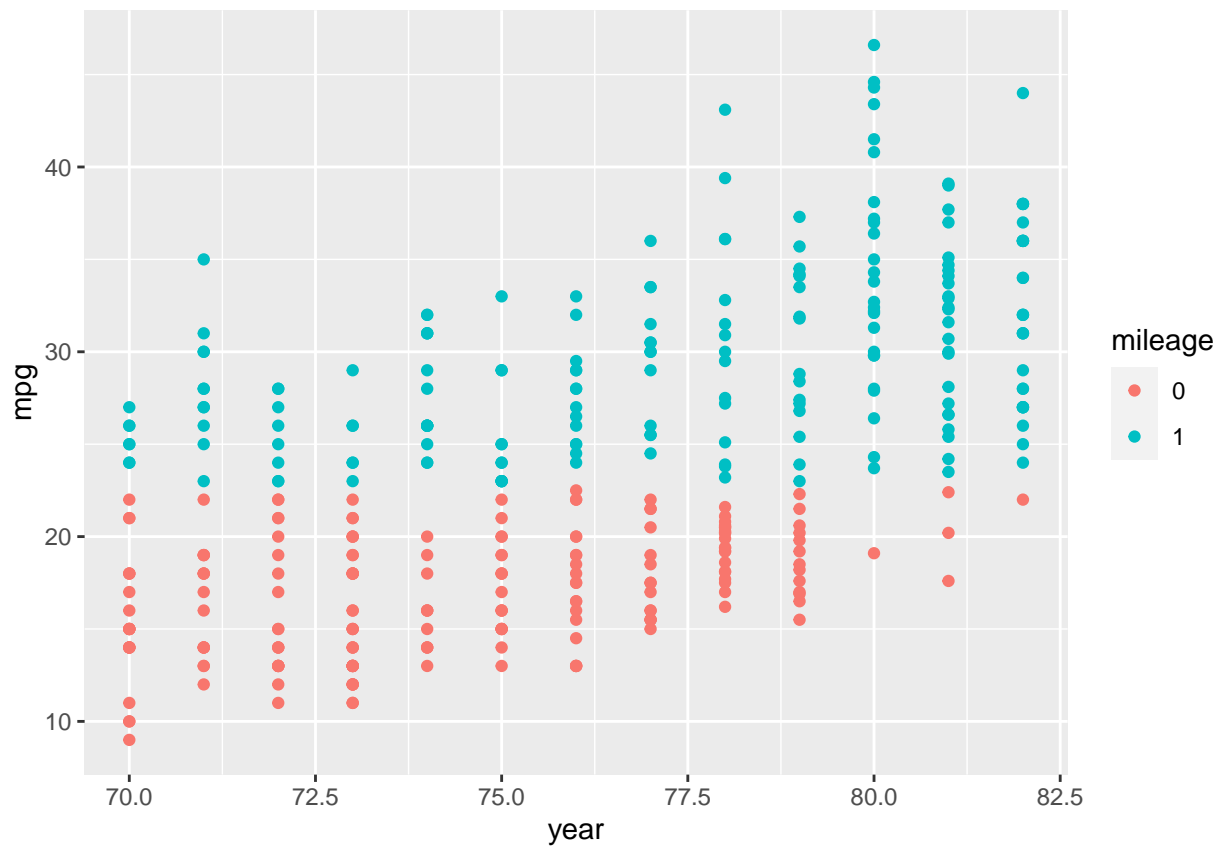
```
ggplot(data=Auto,aes(y=mpg,x=weight,color=mileage))+geom_point()
```



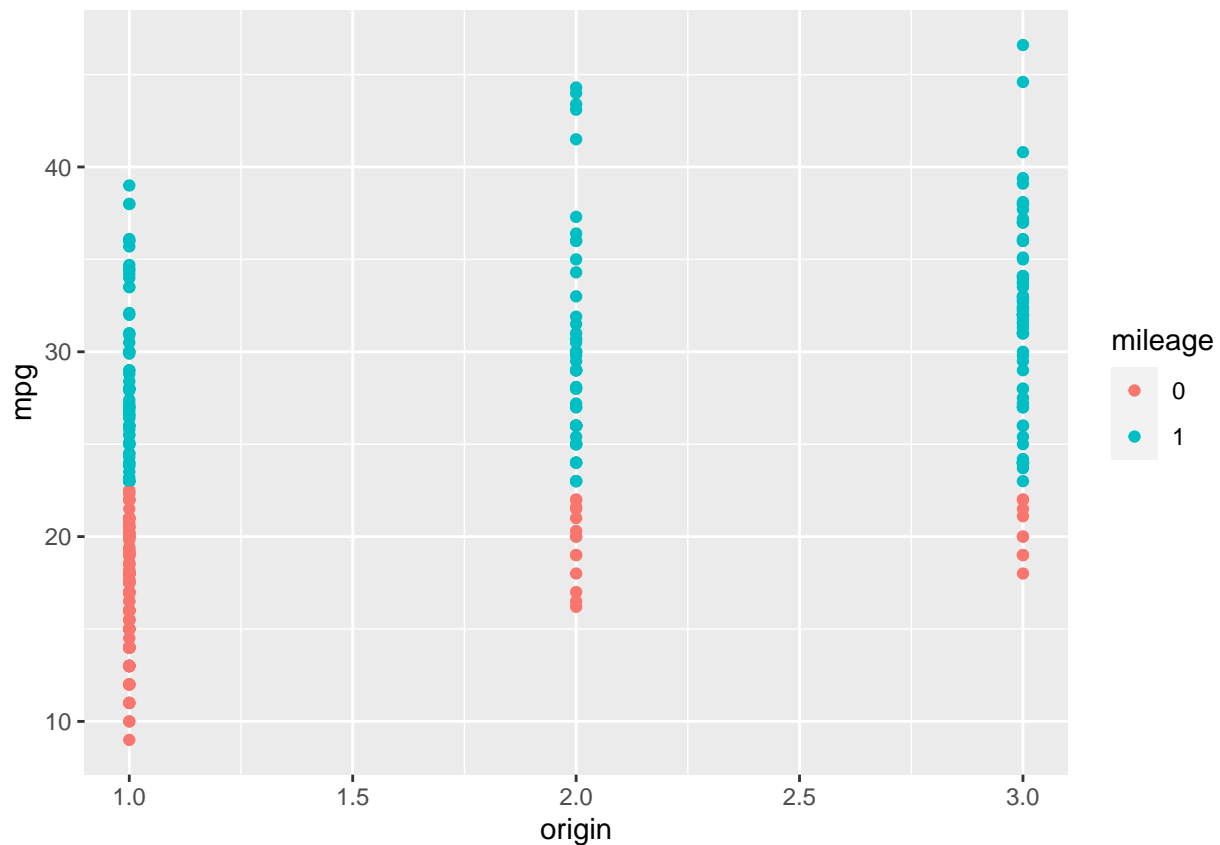
```
ggplot(data=Auto,aes(y=mpg,x=acceleration,color=mileage))+geom_point()
```



```
ggplot(data=Auto,aes(y=mpg,x=year,color=mileage))+geom_point()
```



```
ggplot(data=Auto,aes(y=mpg,x=origin,color=mileage))+geom_point()
```



## Split Data to Train & Test

```
set.seed(1)
sample <- sample.split(Auto$mileage, SplitRatio = .75)
train <- subset(Auto, sample == TRUE)
test  <- subset(Auto, sample == FALSE)
```

## Logistic Regression

```
glm.fit<-glm(mileage~displacement+horsepower+weight+acceleration+year+cylinders+origin,family=binomial,
summary(glm.fit)
```

Call:

```
glm(formula = mileage ~ displacement + horsepower + weight +
    acceleration + year + cylinders + origin, family = binomial,
    data = train)
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.27305	-0.09392	0.00790	0.22766	3.10823



Coefficients:

	Estimate	Std. Error	z value	Pr(> z )	
(Intercept)	-18.919142	6.505128	-2.908	0.003633	**
displacement	-0.001618	0.013913	-0.116	0.907420	
horsepower	-0.024222	0.027116	-0.893	0.371704	
weight	-0.005248	0.001379	-3.806	0.000141	***
acceleration	0.076713	0.157582	0.487	0.626388	
year	0.434125	0.085768	5.062	4.16e-07	***
cylinders	0.271768	0.474154	0.573	0.566534	
origin	0.351906	0.396237	0.888	0.374476	

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 407.57 on 293 degrees of freedom  
Residual deviance: 120.99 on 286 degrees of freedom  
AIC: 136.99

Number of Fisher Scoring iterations: 8

```
glm.probs=predict(glm.fit,test,type="response")  
  
glm.prediction=rep(0,length(test$mileage))  
glm.prediction[glm.probs>0.5]=1  
  
table(glm.prediction,test$mileage)
```

```
glm.prediction  0  1  
               0 45  5  
               1  4 44
```

```
mean(glm.prediction==test$mileage)
```

```
[1] 0.9081633
```

```
mean(glm.prediction!=test$mileage)
```

```
[1] 0.09183673
```

## Linear Discriminant Analysis

```
lda.fit=lda(mileage~displacement+horsepower+weight+acceleration+year+cylinders+origin,data=train)  
  
lda.fit
```

Call:

```
lda(mileage ~ displacement + horsepower + weight + acceleration +
    year + cylinders + origin, data = train)
```

Prior probabilities of groups:

```
0 1
0.5 0.5
```

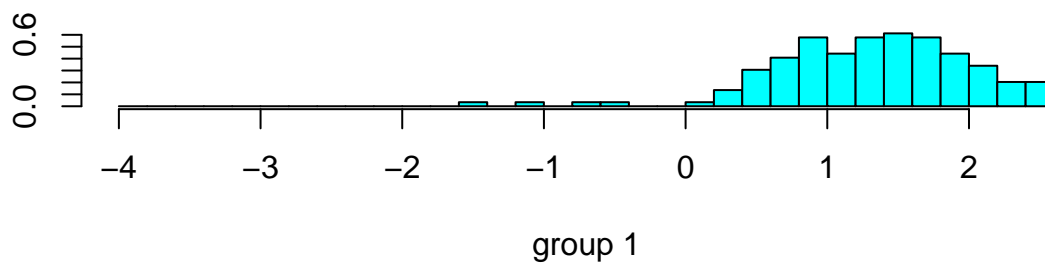
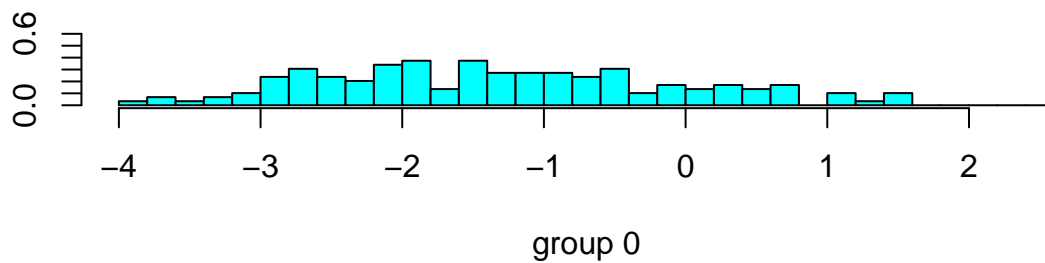
Group means:

	displacement	horsepower	weight	acceleration	year	cylinders	origin
0	268.9116	128.92517	3597.619	14.62925	74.38095	6.680272	1.190476
1	114.6633	79.34694	2313.395	16.39592	77.46259	4.190476	2.013605

Coefficients of linear discriminants:

	LD1
displacement	-0.0002351927
horsepower	0.0097066538
weight	-0.0014434378
acceleration	0.0161947564
year	0.1272136926
cylinders	-0.2642234671
origin	0.1644223915

```
plot(lda.fit)
```



```
lda.prediction=predict(lda.fit, test)
lda.class=lda.prediction$class
table(lda.class,test$mileage)
```

```
lda.class 0 1
          0 45 3
          1 4 46
```

```
mean(lda.class==test$mileage)
```

```
[1] 0.9285714
```

```
mean(lda.class!=test$mileage)
```

```
[1] 0.07142857
```

## Quadratic Discriminant Analysis

```
qda.fit=qda(mileage~displacement+horsepower+weight+acceleration+year+cylinders+origin,data=train)
```

```
qda.fit
```

Call:

```
qda(mileage ~ displacement + horsepower + weight + acceleration +
     year + cylinders + origin, data = train)
```

Prior probabilities of groups:

```
0 1
0.5 0.5
```

Group means:

	displacement	horsepower	weight	acceleration	year	cylinders	origin
0	268.9116	128.92517	3597.619	14.62925	74.38095	6.680272	1.190476
1	114.6633	79.34694	2313.395	16.39592	77.46259	4.190476	2.013605

```
qda.prediction=predict(qda.fit, test)
```

```
qda.class=qda.prediction$class
```

```
table(qda.class,test$mileage)
```

```
qda.class 0 1
          0 48 7
          1 1 42
```

```
mean(qda.class==test$mileage)
```

```
[1] 0.9183673
```

```
mean(qda.class!=test$mileage)
```

```
[1] 0.08163265
```

## K-Nearest Neighbors

```
knn.pred=knn(train[2:8],test[2:8],train$mileage,k=1)
table(knn.pred,test$mileage)
```

```
knn.pred  0  1
          0 47 10
          1  2 39
```

```
mean(knn.pred==test$mileage)
```

```
[1] 0.877551
```

```
knn.pred=knn(train[2:8],test[2:8],train$mileage,k=3)
table(knn.pred,test$mileage)
```

```
knn.pred  0  1
          0 46  9
          1  3 40
```

```
mean(knn.pred==test$mileage)
```

```
[1] 0.877551
```

```
knn.pred=knn(train[2:8],test[2:8],train$mileage,k=10)
table(knn.pred,test$mileage)
```

```
knn.pred  0  1
          0 44 10
          1  5 39
```

```
mean(knn.pred==test$mileage)
```

```
[1] 0.8469388
```

## Conclusions

We have compared the performance on test data for different classifiers trained on 75% of the dataset. The predictors we used to estimate our response were cylinders, displacement, horsepower, weight, acceleration, year & origin. The classification algorithms used were Logistic Regression, Linear Discriminant Analysis, Quadratic Discriminant Analysis & K-Nearest Neighbors. Linear Discriminant Analysis outperformed all the classifiers with a 92% accuracy rate and KNN underperformed with a still relatively high accuracy of 85%