# Homework #4

*Sumer Vaid*

First, lets load some packages:

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.5
```

```
library(lattice)
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.2.5
```

```
library(class)
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 3.2.5
```

```
library(SDMTools)
```

```
##
## Attaching package: 'SDMTools'

## The following objects are masked from 'package:caret':
##
##     sensitivity, specificity
```

Question 1a): I replace the "?" values with NA values.

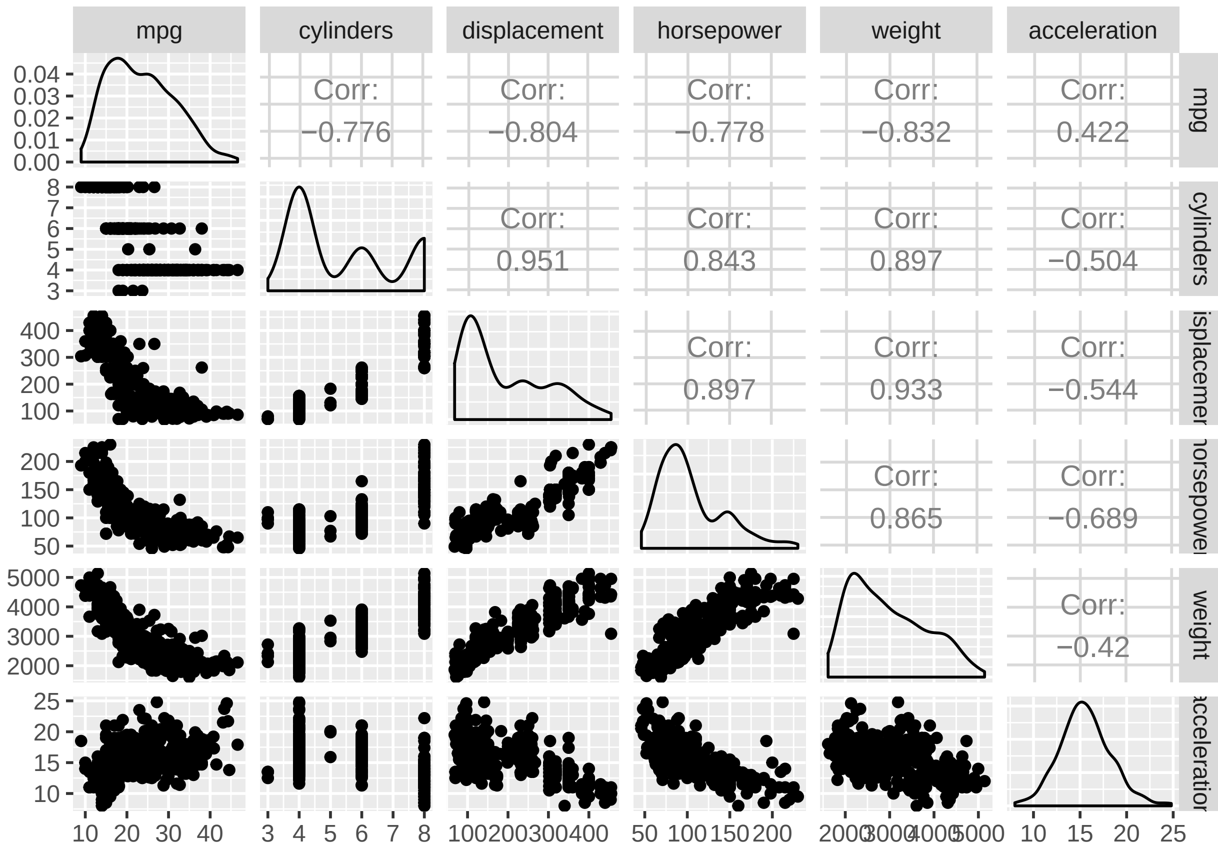
```
auto<-read.csv("auto.csv", na.strings = "?")
```

Question 1b):

```
scatterplot_matrix<-ggpairs(auto[,1:6])
print(scatterplot_matrix)
```

```
## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 5 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 5 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 5 rows containing missing values

## Warning: Removed 5 rows containing missing values (geom_point).

## Warning: Removed 5 rows containing missing values (geom_point).

## Warning: Removed 5 rows containing missing values (geom_point).

## Warning: Removed 5 rows containing non-finite values (stat_density).

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 5 rows containing missing values

## Warning in (function (data, mapping, alignPercent = 0.6, method =
## "pearson", : Removed 5 rows containing missing values
```

```
## Warning: Removed 5 rows containing missing values (geom_point).

## Warning: Removed 5 rows containing missing values (geom_point).
```




Question 1c):

```r
auto$mpg<-as.numeric(auto$mpg)
auto$cylinders<-as.numeric(auto$cylinders)
auto$displacement<-as.numeric(auto$displacement)
auto$horsepower<-as.numeric(auto$horsepower)
cor_matrix<-cor(auto[,1:8], method="pearson", use="pairwise")
print(cor_matrix)
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7762599   -0.8044430 -0.7784268 -0.8317389
## cylinders    -0.7762599  1.0000000    0.9509199  0.8429834  0.8970169
## displacement -0.8044430  0.9509199    1.0000000  0.8972570  0.9331044
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8317389  0.8970169    0.9331044  0.8645377  1.0000000
## acceleration  0.4222974 -0.5040606   -0.5441618 -0.6891955 -0.4195023
## year          0.5814695 -0.3467172   -0.3698041 -0.4163615 -0.3079004
## origin        0.5636979 -0.5649716   -0.6106643 -0.4551715 -0.5812652
##              acceleration       year     origin
## mpg             0.4222974  0.5814695  0.5636979
## cylinders      -0.5040606 -0.3467172 -0.5649716
## displacement   -0.5441618 -0.3698041 -0.6106643
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4195023 -0.3079004 -0.5812652
```

```
## acceleration    1.0000000  0.2829009  0.2100836
## year            0.2829009  1.0000000  0.1843141
## origin          0.2100836  0.1843141  1.0000000
```

Question 1d):

```
fit <-lm(auto$mpg~auto$cylinders+auto$displacement+auto$horsepower+auto$weight+auto$acceleration+auto$y
print(summary(fit))
```

```
##
## Call:
## lm(formula = auto$mpg ~ auto$cylinders + auto$displacement +
##     auto$horsepower + auto$weight + auto$acceleration + auto$year +
##     auto$origin)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -17.218435   4.644294  -3.707  0.00024 ***
## auto$cylinders     -0.493376   0.323282  -1.526  0.12780
## auto$displacement   0.019896   0.007515   2.647  0.00844 **
## auto$horsepower    -0.016951   0.013787  -1.230  0.21963
## auto$weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## auto$acceleration   0.080576   0.098845   0.815  0.41548
## auto$year           0.750773   0.050973  14.729  < 2e-16 ***
## auto$origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

1d i) Weight, year, origin and displacement are statistically significant at the 1% signifiance level.

1d ii) Cylinders, horsepower, acceleration are not significant at the 10% significance level.

1d iii) A change in one unit of the year variable - 1 year - corresponds with a 0.75 change in miles per gallon, given that all other variables are controlled for. Question 1e) According to the scatterplot matrix, it looks like acceleration, horsepower and displacement may have a non-linear relationship with mpg.

1e i)

```
acc2<-auto$acceleration^2
hors2<-auto$horsepower^2
displacement2<-auto$displacement^2
fit2<-lm(mpg~cylinders+displacement2+hors2+weight+acc2+year+origin+acceleration+horsepower+displacement
print(summary(fit2))
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement2 + hors2 + weight +
##     acc2 + year + origin + acceleration + horsepower + displacement,
##     data = auto)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5788 -1.5511 -0.0461  1.5622 11.9010
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.985e+00  6.004e+00   1.163   0.2454
## cylinders      7.388e-01  3.099e-01   2.384   0.0176 *
## displacement2  1.164e-04  2.847e-05   4.090 5.27e-05 ***
## hors2          5.802e-04  1.369e-04   4.237 2.85e-05 ***
## weight        -2.925e-03  6.695e-04  -4.368 1.62e-05 ***
## acc2           3.306e-02  1.566e-02   2.111   0.0354 *
## year           7.495e-01  4.483e-02  16.716  < 2e-16 ***
## origin         5.737e-01  2.683e-01   2.138   0.0332 *
## acceleration  -1.352e+00  5.378e-01  -2.514   0.0124 *
## horsepower    -2.221e-01  3.939e-02  -5.638 3.36e-08 ***
## displacement  -6.999e-02  1.616e-02  -4.332 1.90e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.919 on 381 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.8637, Adjusted R-squared:  0.8602
## F-statistic: 241.5 on 10 and 381 DF,  p-value: < 2.2e-16
```

1e ii) The adjusted R Square term is 0.8602. This is greater than the initial adjusted R Square term of 0.8182 obtained from the model without square terms.

1e iii) Both displacement and its squared term have coefficients that are statistically significant at a level of 1%. Furthermore, the coefficient value went from being positive to negative of the displacement coeffiecient.

Question 1 f)

```
new_df<-0
new_df$cylinders<-6
```

```
## Warning in new_df$cylinders <- 6: Coercing LHS to a list
```

```
new_df$displacement<-200
new_df$horsepower<-100
new_df$weight<-3100
new_df$acceleration<-15.1
new_df$year<-99
new_df$origin<-1
new_df$displacement2<-(new_df$displacement^2)
new_df$hors2<-(new_df$horsepower^2)
new_df$acc2<-(new_df$acceleration^2)
print(predict.lm(fit2, new_df, response=TRUE))
```

```
##        1
## 38.49804
```

The mpg of the specified vechile would be 38.49804 miles per gallon.

Question 2

2 a)

```r
knn<-data.frame(c(1,2,3,4,5,6))
knn$X1<-c(0,2,0,0,-1,1)
knn$X2<-c(3,0,1,1,0,1)
knn$X3<-c(0,0,3,2,1,1)
knn$Y<-c("Red", "Red", "Red", "Green", "Green", "Red")
knn$dist<-sqrt(knn$X1^2+knn$X2^2+knn$X3^2)
```

2 b) Since the distance is shortest to observation 4 (distance=1.414), I predict that the response variable will be green.

2 c) I will pick those values that correspond to the three closest neighbors to the origin point. This distance is shortest for observation 2 (distance=2), observation 5 (distance=1.414) and observation 6 (distance=1.732).

2 d) A K increases, the patterns of results become more linear. As such, if the Bayes decision boundary is extremely non-linear, a k-value smaller in magnitude will be better.

2 e)

```r
pred<-c(1,1,1)
labels<-knn$Y
test_pred<-knn(knn[,2:4],pred,cl=labels,k=2)
print(test_pred)
```

```
## [1] Red
## Levels: Green Red
```

The KNN classifier is Green.

Question 3)

```r
auto$mpg_high<-0
auto$mpg_high[auto$mpg<median(auto$mpg)]<-0
auto$mpg_high[auto$mpg>median(auto$mpg)]<-1
auto$constant<-1 #adds a constant term
```

Question 3a)

```r
fit3<-glm(mpg_high~cylinders+displacement+horsepower+weight+acceleration+year+origin, family=binomial(li
summary(fit3)
```

```
##
## Call:
## glm(formula = mpg_high ~ cylinders + displacement + horsepower +
##     weight + acceleration + year + origin, family = binomial(link = "logit"),
##     data = auto)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.41620  -0.09337  -0.00041   0.18644   2.53708
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -2.271e+01  6.140e+00  -3.700 0.000216 ***
## cylinders    -6.329e-02  4.366e-01  -0.145 0.884756
## displacement -2.199e-04  1.306e-02  -0.017 0.986568
## horsepower   -3.987e-02  2.464e-02  -1.618 0.105725
## weight       -4.816e-03  1.224e-03  -3.935 8.34e-05 ***
## acceleration -1.777e-02  1.407e-01  -0.126 0.899483
```

```
## year          5.196e-01  8.422e-02   6.169 6.87e-10 ***
## origin        4.990e-01  3.603e-01   1.385 0.166066
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 542.60  on 391  degrees of freedom
## Residual deviance: 148.43  on 384  degrees of freedom
##   (5 observations deleted due to missingness)
## AIC: 164.43
##
## Number of Fisher Scoring iterations: 8
```

Weight and year are both significant at a level of 10%.

Question 3b)

```
#random_state=10 from Python in R:
set.seed(10)
#train_test_split in python in R:
breakdata<-createDataPartition(auto$mpg, p=0.5, list=FALSE, times=1)
train<-auto[breakdata,]
test<-auto[-breakdata,]
```

Question 3c)

```
fit4<-glm(mpg_high~cylinders+displacement+horsepower+weight+acceleration+year+origin,
          family=binomial(link='logit'), data=train)

summary(fit4)
```

```
##
## Call:
## glm(formula = mpg_high ~ cylinders + displacement + horsepower +
##     weight + acceleration + year + origin, family = binomial(link = "logit"),
##     data = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.39831  -0.12488  -0.00304   0.25372   2.23403
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -22.636484   8.366129  -2.706  0.00682 **
## cylinders     -0.640609   0.590332  -1.085  0.27785
## displacement   0.009321   0.017057   0.546  0.58476
## horsepower    -0.014035   0.031119  -0.451  0.65199
## weight        -0.004401   0.001584  -2.779  0.00546 **
## acceleration  -0.001086   0.171317  -0.006  0.99494
## year           0.489362   0.109314   4.477 7.58e-06 ***
## origin         0.338258   0.430314   0.786  0.43182
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 272.851  on 196  degrees of freedom
## Residual deviance:  86.982  on 189  degrees of freedom
##   (2 observations deleted due to missingness)
## AIC: 102.98
##
## Number of Fisher Scoring iterations: 7
```

The estimates are: B0=-22.63 (intercept) B1=-0.640609 B2=0.009321 B3=-0.014035 B4=-0.004401 B5=-0.001086 B6=0.489362 B7=0.338258

Question 3d)

```r
predicted<-predict(fit4,test,type="response")
actual<-train$mpg_high
actual<-actual[-199]
confm<-confusion.matrix(actual, predicted)
```

```
## Warning in confusion.matrix(actual, predicted): 3 data points removed due
## to missing data
```

```r
print(confm)
```

```
##     obs
## pred  0  1
##    0 70 28
##    1 33 64
## attr(,"class")
## [1] "confusion.matrix"
```

As the confusion matrix indicates, the model is (roughly) equally good at classifyig both 0s and 1s. Therefore, it is equally good at classifying the presence or absence of a high mpg.