# The University of British Columbia

*Data Science 570 Predictive Modelling*

Lab Assignment 3

The TA will demonstrate exercises 1 through 3. You are expected to submit answers to exercise 7. Instructions: Please use a png, pdf or html file for submission.

1. Life Auto Data

   Today we'll be learning how to plot and interpret several diagnostic plots. To begin, lets start by fitting a linear regression model to the Auto dataset available in the ISLR package. For more details see ?Auto. fit1 attempts to explain gas mileage (in mpg mpg) using the predictor variable of the engines horsepower horsepower.

```
install.packages('ISLR', repos='http://cran.us.r-project.org')

## Installing package into 'C:/Users/xshi/Documents/R/win-library/4.0'
## (as 'lib' is unspecified)

## package 'ISLR' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\xshi\AppData\Local\Temp\RtmpW6fYm4\downloaded_packages

library(ISLR)

## Warning:  package 'ISLR' was built under R version 4.0.5

attach(Auto)
head(Auto)

##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                        name
## 1 chevrolet chevelle malibu
## 2         buick skylark 320
## 3        plymouth satellite
## 4             amc rebel sst
## 5               ford torino
## 6          ford galaxie 500

fit1 <- lm(mpg~horsepower)
summary(fit1)

##
## Call:
```

```
## lm(formula = mpg ~ horsepower)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5710  -3.2592  -0.3435   2.7630  16.9240
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.935861   0.717499   55.66   <2e-16 ***
## horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.906 on 390 degrees of freedom
## Multiple R-squared:  0.6059,Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

Judging by the very low p-values for $\beta_1$ (slope), it appears that there a relationship between the predictor and the response. Since the sign for slope is negative ( $\hat{\beta}_1$=-0.1578) we expect this relationship to be negative, that is, as the horsepower of our engine increase, gas mileage tends to go down. This relationship would also appear to be relatively strong as indicated by the high $R^2$ value = 0.6049 (ie approximately 60% of the variance in the response is being explained by the simple model.

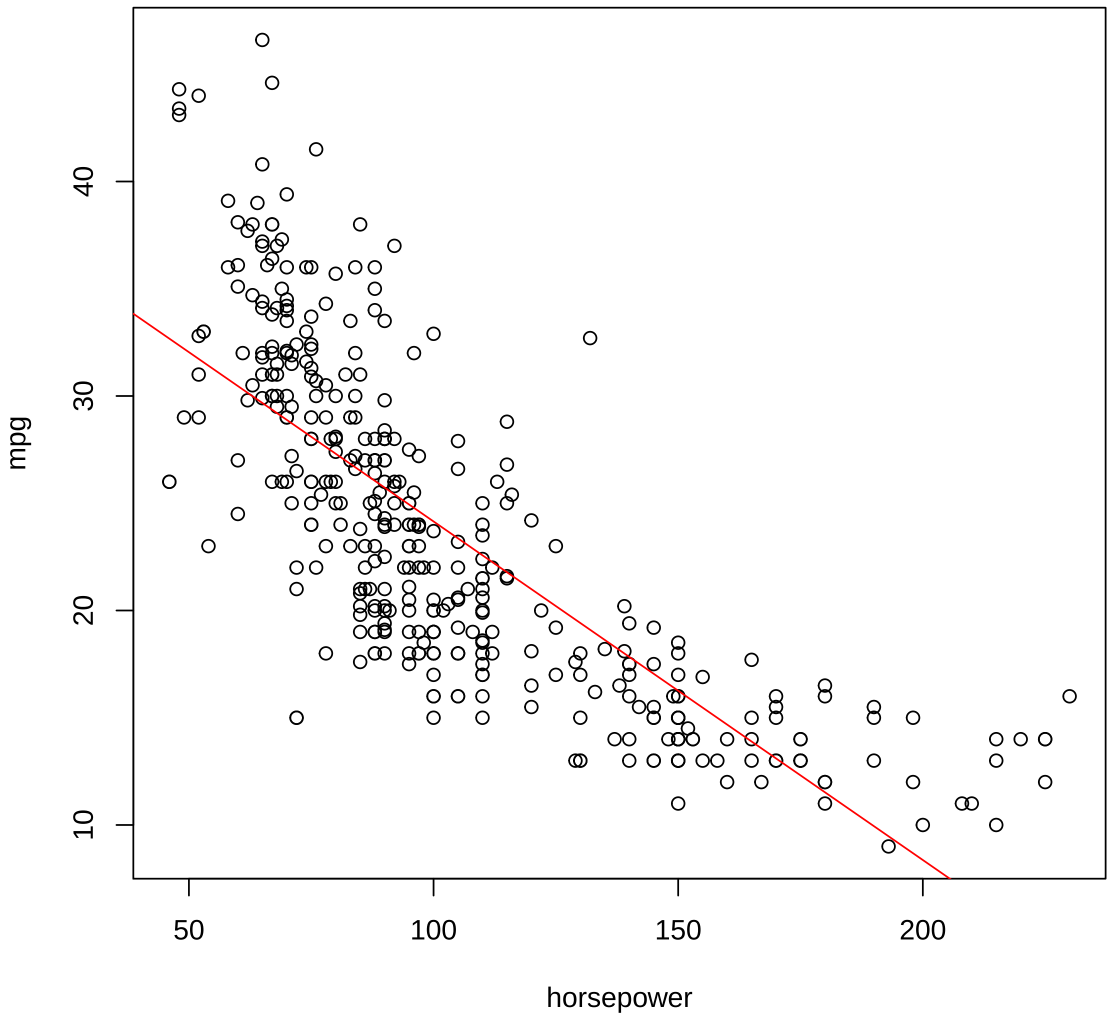
2. Residual analysis

   While the conclusions we made above may seem appropriate based on the R output, we would be amiss to not check the assumptions of this model before we jump to any conclusions. Recall that the assumptions for a linear regression model are that:

   - There exists an (at least approximate) linear relationship between Y and X.
   - The distribution of $\varepsilon_i$ has constant variance.
   - $\varepsilon_i$ is normally distributed.
   - $\varepsilon_i$ are independent of one another. For example, $\varepsilon_2$ is not affected by $\varepsilon_1$.

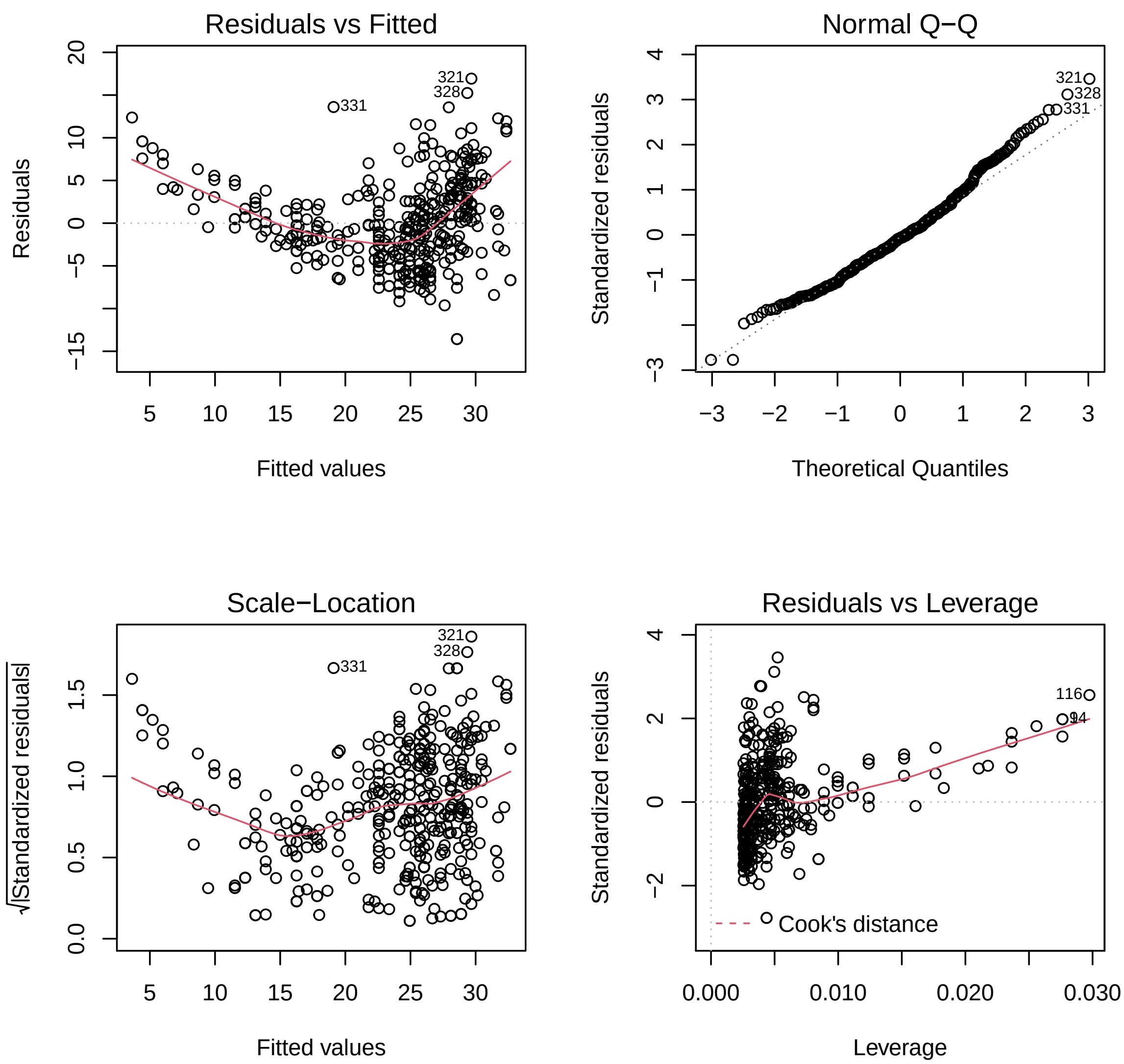
   To check the first assumption, we may like to look at the scatterplot with the line of best fit. As demonstrated below, it would appear that the response and predictor variable have a curved relationship rather than a linear one.

```
plot(mpg~horsepower, data=Auto)
abline(fit1, col="red")
```




We could also explore the diagonostic plots produced when call plot() on the output of an lm() object. To see them all in one plot, let's change the panel layout of our plot window to store 4 figures (in this case in a 2 by 2 matrix) (for more details see ?par). In other words we could call:

```
par(mfrow=c(2,2))
plot(fit1)
```




As seen in the Scale-Location graph, it appears there may also been some outliers in our data. Using the ISLR recommendation, we can check the observations whose studentized residuals and observe which ones are greater than 3.

```
which(rstudent(fit1)>3)

## 321 328
## 321 328

## 321 328
```

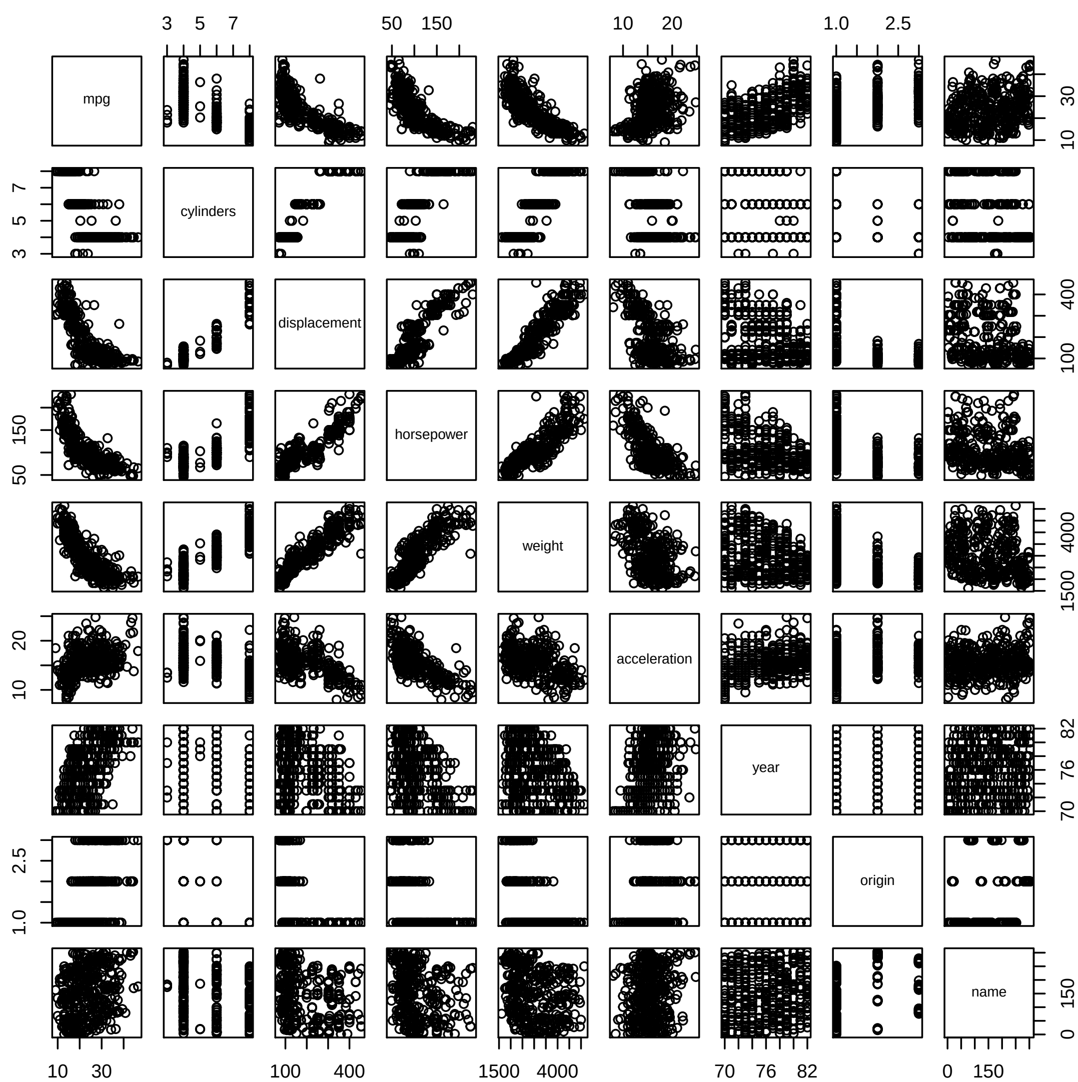
It appears from the Residuals vs Leverage graph that a number of points could be high leverage points. To investigate the top, say 5, observations having the highest cooks distance, we could do the following:

```
order(cooks.distance(fit1), decreasing = TRUE)[1:5]

## [1] 116   9  14   7  95

## 116   9  14   7  95
```

3. Collinearity

   Now lets consider a MLR model using more predictors from the Auto data from above. We discussed in lecture how some of these predictor variables might be correlated with one another. To investigate this, lets have a look at a scatterplot matrix which includes all of the variables in the data set using the pairs() function.

```
pairs(Auto)
```



We can see that horespower and displacement for example are highly correlated. We might also decide to look at the matrix of correlations between the variables using the function cor().

```
# remove "name" (since it is not numeric)
cor(Auto[,names(Auto)!="name"])

##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
## displacement   -0.5438005 -0.3698552 -0.6145351
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

Notice that the correlation between horespower and displacement is very high (0.8972570) as is the correlation between cylinders and displacement is very high (0.9508233). As discussed in lecture, we can go a step further than the bivariate correlations and compute the so-called VIF vales. The car package will help us do this.

```
require(car)

## Loading required package:  car

## Loading required package:  carData

## Loading required package: car
## Loading required package: carData
lm.fit <- lm(mpg ~ .-name, data=Auto)
summary(lm.fit)

##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## 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 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## 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
## Multiple R-squared:  0.8215,Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

```
vif(lm.fit)
```

```
##    cylinders displacement   horsepower       weight acceleration         year
##    10.737535    21.836792     9.943693    10.831260     2.625806     1.244952
##       origin
##     1.772386
```

4. Consider the simple regression model fit to the lawn roller data.

```
install.packages('DAAG', repos='http://cran.us.r-project.org')
```

```
## Installing package into 'C:/Users/xshi/Documents/R/win-library/4.0'
## (as 'lib' is unspecified)
## also installing the dependencies 'rbibutils', 'Rdpack'
```

```
##
##   There are binary versions available but the source versions are later:
##           binary source needs_compilation
## rbibutils  2.2.8 2.2.16              TRUE
## Rdpack       2.3    2.5             FALSE
## DAAG        1.24 1.25.4             FALSE
##
##   Binaries will be installed
## package 'rbibutils' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\xshi\AppData\Local\Temp\RtmpW6fYm4\downloaded_packages
```

```
## installing the source packages 'Rdpack', 'DAAG'
```

```
library(DAAG)
```

```
##
## Attaching package:  'DAAG'
## The following object is masked from 'package:car':
##
##     vif

x=roller$weight
y=roller$depression
```

4.1 2 marks Plot the data points and fitted line.

4.2 2 marks Call plot() on the output of an lm() object.

5. (2 marks) Consider the blood pressure data again. Call pairs() and cor() to show the bivariate correlations and compute the so-called VIF vales.

6. (3 marks) Simulate 100 Poisson random numbers using rpois(100, $\lambda$), where $\lambda = e^{\beta_0+\beta_1 x}$, $\beta_0 = 1$, $\beta_1 = -1$, and x is from a standard normal distribution. Use glm() function to fit Poisson regression and print the coefficients.

7. (3 marks) Simulate 100 Bernoulli random numbers using rbinom(100, 1, prob = p), where $p = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$, $\beta_0 = 1$, $\beta_1 = -1$, and x is from a standard normal distribution. Use glm() function to fit logistic regression and print the coefficients.