Coursework 1

- Programme: Msc Data Science
- Name: Marco Catania
- Student ID: 13129252

1. Statistical learning methods

(a) The sample size n is extremely large, and the number of predictors p is small.

- Generally expected performance of a flexible statistical learning method over an inflexible method:

Better

- Reason:

Flexible methods will have in general a lower Mean Squared Error (MSE) as the function will tend to follow the observations, match the true unknow form of f and give a better prediction of the response than an inflexible method. But if the sample has a high level of noise (errors), in that case flexible functions will be subject to overfitting (follow the errors) and underperform.

(b) The number of predictors p is extremely large, and the number of observations n is small.

- Generally expected performance of a flexible statistical learning method over an inflexible method:

Worse

- Reason:

A flexible method will tend to overfit the small number of observation of the sample and give a poor prediction of the response when the model is applied to the test observations.

(c) The relationship between the predictors and response is highly non-linear.

- Generally expected performance of a flexible statistical learning method over an inflexible method:

Better

- Reason:

A flexible method, has more degrees of freedom than an inflexible method. Therefore when the relationship predictors-response is highly non-linear, the flexible method will have the lowest MSE as it will fit better the observations.

(d) The variance of the error terms is extremely high.

- Generally expected performance of a flexible statistical learning method over an inflexible method:

Worse

- Reason:

A high variance of the error terms means the data set has a high level of noise (errors). Therefore a flexible method will follow the noise closely (overfitting) and make a poor prediction of the response.

2. Descriptive analysis

```
oral <- c(4, 1, 4, 5, 3, 2, 3, 4, 3, 5, 2, 2, 4, 3, 5, 5, 1, 1, 1, 2)
written <- c(2, 3, 1, 4, 2, 5, 3, 1, 2, 1, 2, 2, 1, 1, 2, 3, 1, 2, 3, 4)
```

(a) Use R to calculate the mean, the mode, the median, the variance and the standard deviation of the oral and written exams separately and together as well.

- Mean Oral exam

```
mean(oral)
```

```
## [1] 3
```

- Mode Oral exam

```
sort(table(oral))
```

```
## oral
## 1 2 3 4 5
## 4 4 4 4 4
```

The frequency table shows that all values are same frequency, therefore are all modes.

- Median Oral exam

```
median(oral)
```

```
## [1] 3
```

- Variance Oral exam

```
var(oral)
```

```
## [1] 2.105263
```

- Standard deviation Oral exam

```
sd(oral)
```

```
## [1] 1.450953
```

- Mean Written exam

```
mean(written)
```

```
## [1] 2.25
```

- Mode Written exam

```
names(sort(-table(written)))[1]
```

```
## [1] "2"
```

- Median Written exam

```r
median(written)
```

```
## [1] 2
```

- Variance Written exam

```r
 var(written)
```

```
## [1] 1.355263
```

- Standard deviation Oral exam

```r
sd(written)
```

```
## [1] 1.164158
```

- Mean Oral & Written exams

```r
mean(c(oral, written))
```

```
## [1] 2.625
```

- Mode Oral & Written exams

```r
names(sort(-table(c(oral, written))))[1]
```

```
## [1] "2"
```

- Median Oral & Written exams

```r
median(c(oral, written))
```

```
## [1] 2
```

- Variance Oral & Written exams

```r
var(c(oral, written))
```

```
## [1] 1.830128
```

- Standard deviation Oral & Written exams

```r
sd(c(oral, written))
```

```
## [1] 1.352822
```

(b) Find the covariance and correlation between the oral and written exam scores.

- Covariance

```r
cov(oral, written)
```

```
## [1] -0.3157895
```

- Correlation

```r
cor(oral, written)
```

```
## [1] -0.1869531
```

(c) Is there a positive or negative or no correlation between the two ?

There is low negative correlation between the oral and written exam scores.

(d) Is there causation between the two ? Justify your answers.

No. Even if a correlation (low negative, here) is observed between two variables, that does not imply a causation between them. Therefore we cannot say there is a causation between the oral and written exam scores.

3. Descriptive analysis

```
library("ISLR")
attach(Auto)
```

(a) Which of the predictors are quantitative , and which are qualitative ?

Quantitative predictors: - mpg - cylinders - displacement - horsepower - weight - acceleration - year

Qualitative predictors: - origin - name

(b) What is the range of each quantitative predictor?

```
sapply(Auto[, 1:7], range)
```

```
##       mpg cylinders displacement horsepower weight acceleration year
## [1,]  9.0         3           68         46   1613          8.0   70
## [2,] 46.6         8          455        230   5140         24.8   82
```

(c) What is the mean and standard deviation of each quantitative predictor?

```
sapply(Auto[, 1:7], mean)
```

```
##          mpg    cylinders displacement   horsepower       weight
##    23.445918     5.471939   194.411990   104.469388  2977.584184
## acceleration         year
##    15.541327    75.979592
```

```
sapply(Auto[, 1:7], sd)
```

```
##          mpg    cylinders displacement   horsepower       weight
##     7.805007     1.705783   104.644004    38.491160   849.402560
## acceleration         year
##     2.758864     3.683737
```

(d) Now remove the 10th through 85th observations. What is the range, mean, and standard deviation of each predictor in the subset of the data that remains ?

```
newFrame = Auto[-(10:85),]
```

```
sapply(newFrame[, 1:7], range)
```

```
##       mpg cylinders displacement horsepower weight acceleration year
## [1,] 11.0         3           68         46   1649          8.5   70
## [2,] 46.6         8          455        230   4997         24.8   82
```

```
sapply(newFrame[, 1:7], mean)
```

```
##          mpg    cylinders displacement   horsepower       weight
##    24.404430     5.373418   187.240506   100.721519  2935.971519
## acceleration         year
##    15.726899    77.145570
```

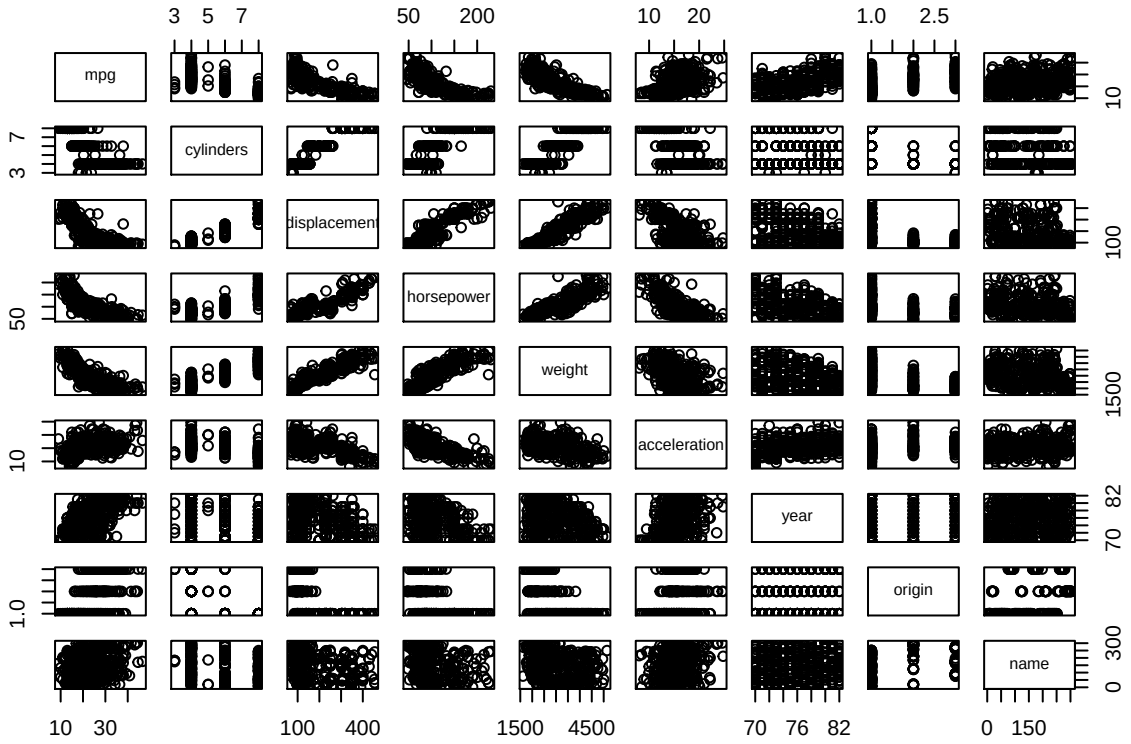
```
sapply(newFrame[, 1:7], sd)
```

```
##          mpg    cylinders displacement   horsepower       weight
##     7.867283     1.654179    99.678367    35.708853   811.300208
## acceleration         year
##     2.693721     3.106217
```

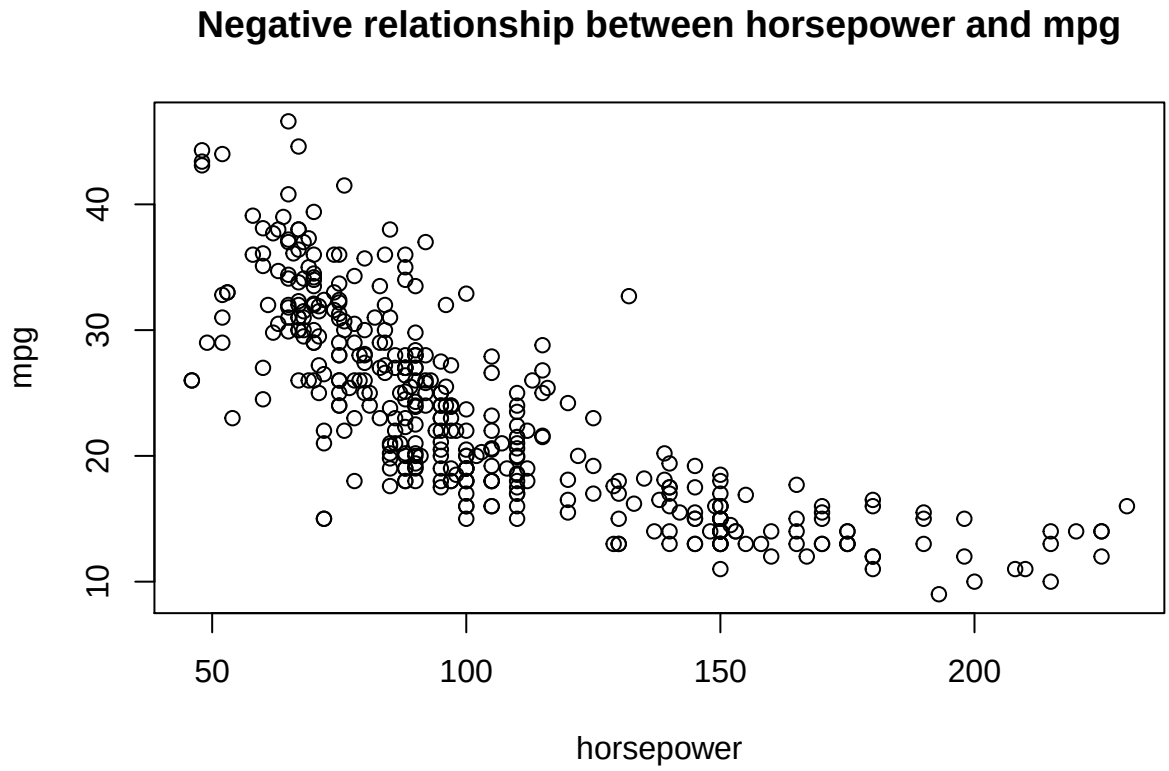
(e) Using the full data set, investigate the predictors graphically, using scatterplots or other tools of your choice. Create some plots highlighting the relationships among the predictors. Comment on your findings.

- Comments are on plot titles

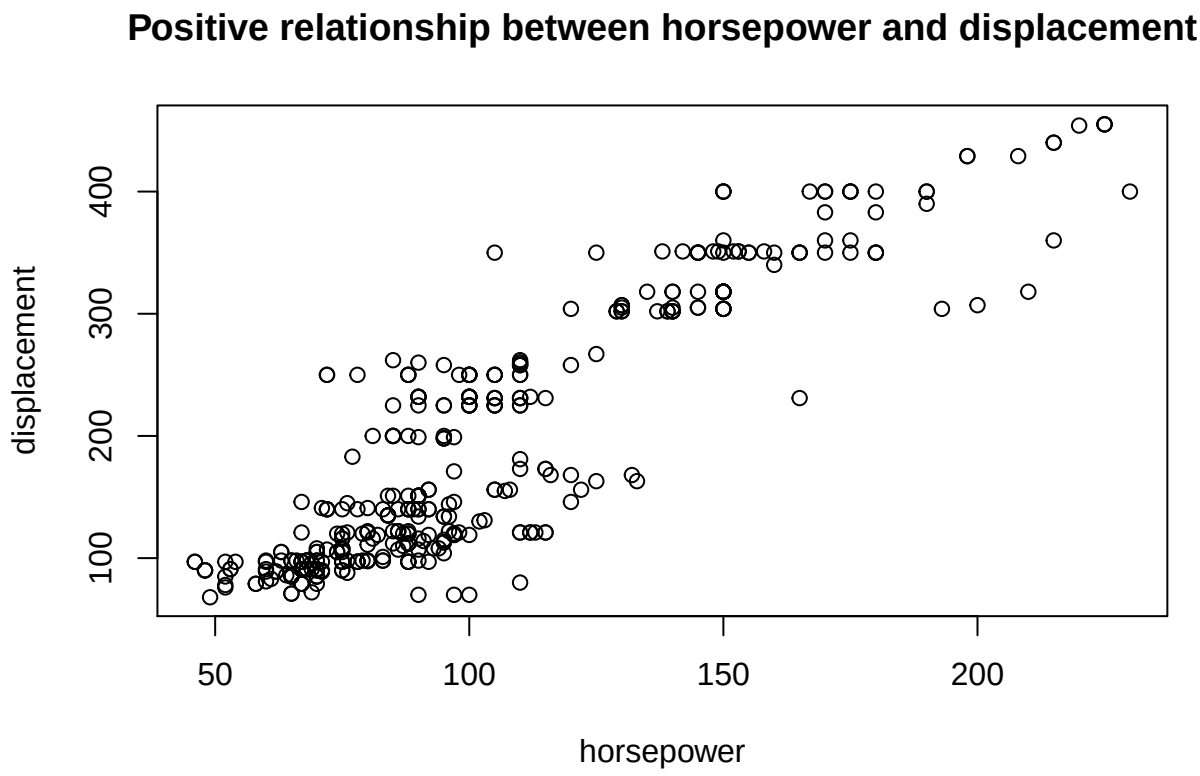
```
pairs(Auto)
```




```
plot(horsepower, mpg,
     title("Negative relationship between horsepower and mpg"))
```

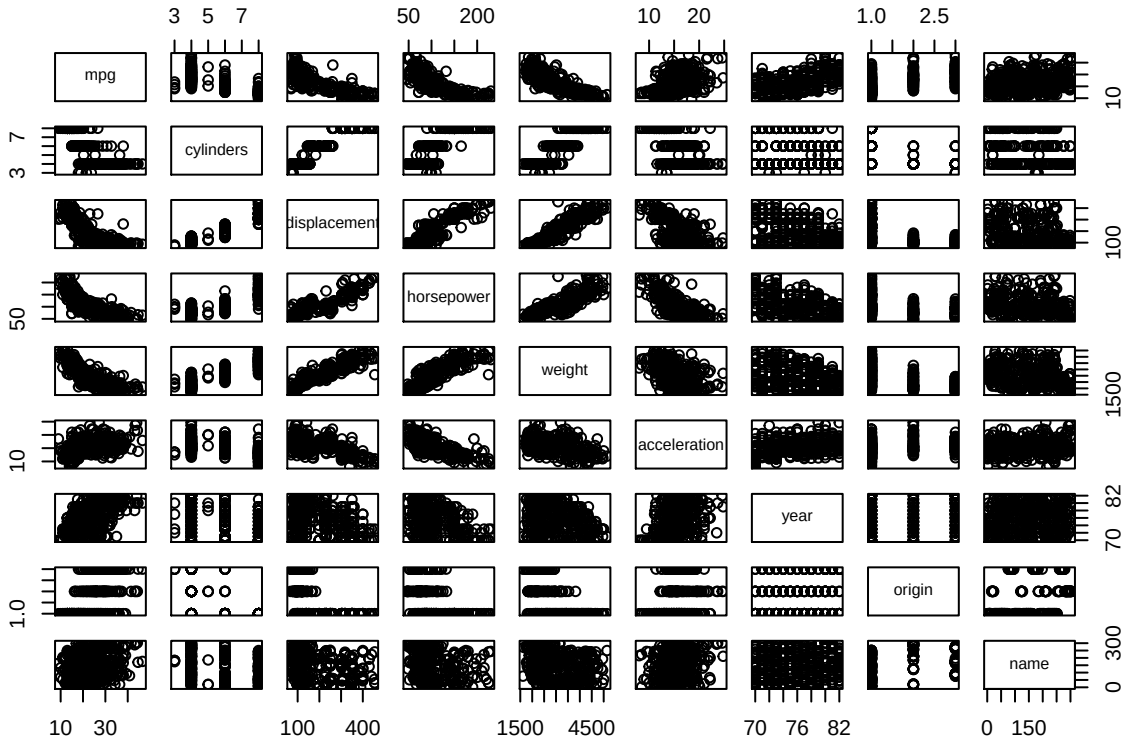



```
plot(horsepower, displacement,
     title("Positive relationship between horsepower and displacement"))
```



(f) Suppose that we wish to predict gas mileage (mpg) on the basis of the other variables. Do your plots suggest that any of the other variables might be useful in predicting mpg? Justify your answer.

```
pairs(Auto)
```



The scatterplot matrix suggests that all the variables show some relationship with mpg (some strong, some weak), therefore they might be useful in predicting mpg.

4. Linear regression

(a) Use the lm() function to perform a simple linear regression with mpg as the response and horsepower as the predictor. Use the summary() function to print the results. Comment on the output.

```
lm.fit = lm(mpg~horsepower)
summary(lm.fit)
```

```
##
## 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
```

i. Is there a relationship between the predictor and the response ?

With a p-value (<2e-16) for the predictor inferior to the criterion of 0.05, we reject the null hypothesis. Hence, there is a relationship between the predictor and the response.

ii. How strong is the relationship between the predictor and the response ?

The p-value for the predictor is so small that it is inferior to the lowest criterion of 0.01. Also Hence there is a particularly strong relationship between the predictor and the response.

iii. Is the relationship between the predictor and the response positive or negative ?

The predictor has a negative coefficient (-0.157845). It means the relationship between the predictor and the response is negative. For instance a raise in horsepower of 100 will decrease mpg of approximately 15.78.

iv. What is the predicted mpg associated with a horsepower of 98? What are the associated 95% confidence and prediction intervals?

```
predict(lm.fit, data.frame(horsepower=c(98)),
        interval = 'confidence')
```

```
##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108
```

```
predict(lm.fit, data.frame(horsepower=c(98)),
        interval = 'prediction')
```

```
##        fit     lwr      upr
## 1 24.46708 14.8094 34.12476
```

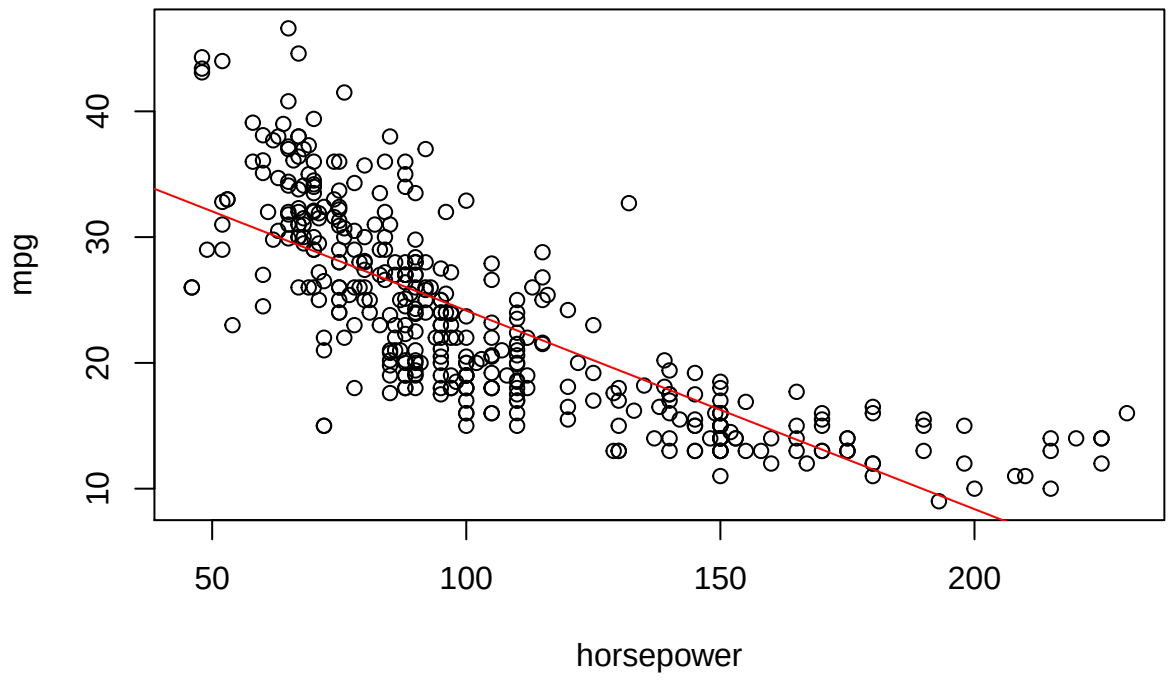
The predicted mpg associated with a horsepower of 98 is 24.47 (2 decimal places).

The 95% confidence interval (2 decimal places) is [23.97(lower), 24.96(upper)]

The 95% prediction interval is (2 decimal places) [14.81(lower), 34.12(upper)]

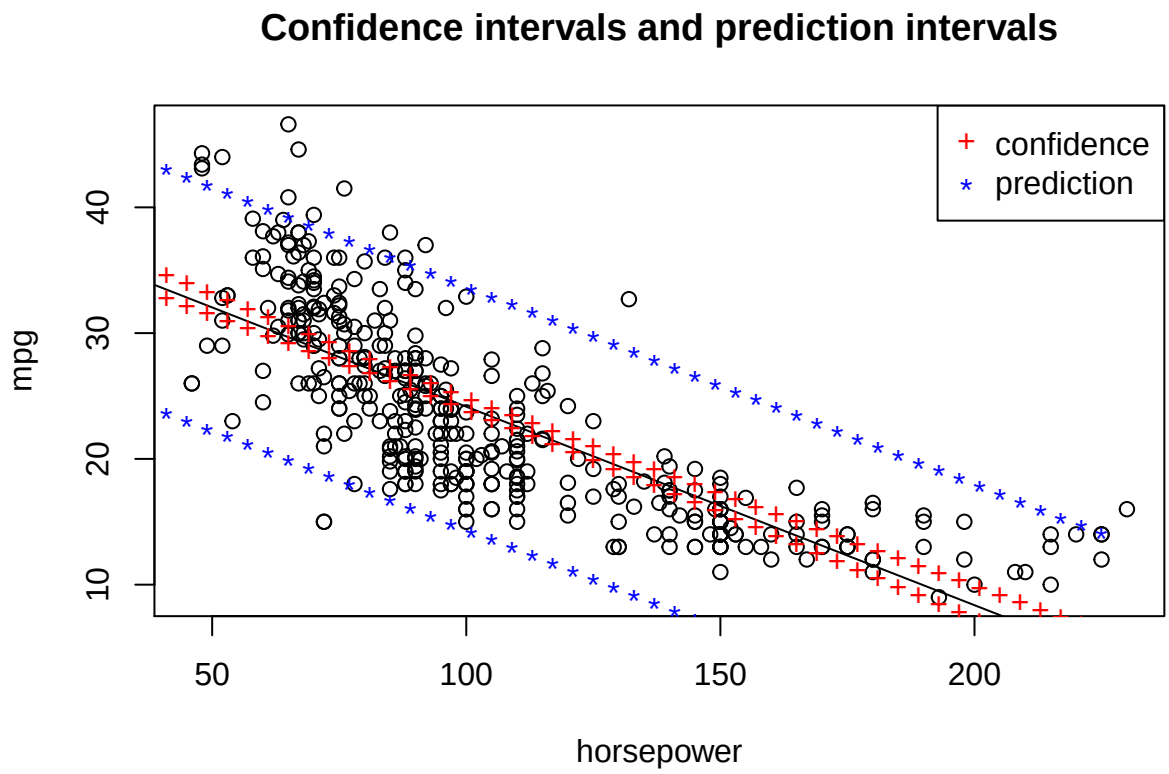
(b) Plot the response and the predictor. Use abline() function to display the least squares regression line.

```
plot(horsepower, mpg)
abline(lm.fit, col='red')
```



(c) Plot the 95% confidence interval and prediction interval in the same plot as (b) using different colours and legends.

```
plot(horsepower, mpg,
xlab="horsepower", ylab = "mpg",
main = "Confidence intervals and prediction intervals")
abline(lm.fit)
newData <- data.frame(horsepower=seq(25,225,length=51))
p_conf <- predict(lm.fit,newData,interval="confidence")
p_pred <- predict(lm.fit,newData,interval="prediction")
lines(newData$horsepower,p_conf[,"lwr"],col="red", type="b",pch="+")
lines(newData$horsepower,p_conf[,"upr"],col="red", type="b",pch="+")
lines(newData$horsepower,p_pred[,"upr"],col="blue", type="b",pch="*")
lines(newData$horsepower,p_pred[,"lwr"],col="blue",type="b",pch="*")
legend("topright",
  pch=c("+","*"),
  col=c("red","blue"),
  legend = c("confidence","prediction"))
```



5. Logistic regression

```
library(MASS)
data <- Boston
attach(data)
crim_pred <- rep(0, nrow(data))
crim_pred[crim > median(crim)] <- 1
data$crim <- crim_pred
glm.fits.allpredictors = glm(crim~zn+indus+chas+nox+rm+age+
                               dis+rad+tax+ptratio+black+lstat+medv, data=data, family="binomial")
glm.fits.subset1 = glm(crim~zn+nox+dis+rad+tax+ptratio+
                         black+medv, data=data, family="binomial")
glm.fits.subset2 = glm(crim~indus+chas+rm+age+lstat, data=data, family="binomial")

detach(data)
```

```
summary(glm.fits.allpredictors)
```

```
##
## Call:
## glm(formula = crim ~ zn + indus + chas + nox + rm + age + dis +
##     rad + tax + ptratio + black + lstat + medv, family = "binomial",
##     data = data)
##
```

```
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3946  -0.1585  -0.0004   0.0023   3.4239
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -34.103704   6.530014  -5.223 1.76e-07 ***
## zn           -0.079918   0.033731  -2.369  0.01782 *
## indus        -0.059389   0.043722  -1.358  0.17436
## chas          0.785327   0.728930   1.077  0.28132
## nox          48.523782   7.396497   6.560 5.37e-11 ***
## rm           -0.425596   0.701104  -0.607  0.54383
## age           0.022172   0.012221   1.814  0.06963 .
## dis           0.691400   0.218308   3.167  0.00154 **
## rad           0.656465   0.152452   4.306 1.66e-05 ***
## tax          -0.006412   0.002689  -2.385  0.01709 *
## ptratio       0.368716   0.122136   3.019  0.00254 **
## black        -0.013524   0.006536  -2.069  0.03853 *
## lstat         0.043862   0.048981   0.895  0.37052
## medv          0.167130   0.066940   2.497  0.01254 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 701.46  on 505  degrees of freedom
## Residual deviance: 211.93  on 492  degrees of freedom
## AIC: 239.93
##
## Number of Fisher Scoring iterations: 9
```

The summary function for the model including all the predictors shows that the most correlated predictor to the response is nox with p-value = 5.37e-11 (well below 1% -> strong relationship predictor-response). nox has a positive coefficient (-> positive relationship) and also the highest: 48.523782. That means if the suburb has a high nox, it will have a strong likelihood to get a crime rate above the median.

```
summary(glm.fits.subset2)
```

```
##
## Call:
## glm(formula = crim ~ indus + chas + rm + age + lstat, family = "binomial",
##     data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.74307  -0.57644   0.01691   0.53593   2.37806
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -8.822186   1.609360  -5.482 4.21e-08 ***
## indus        0.155675   0.025179   6.183 6.30e-10 ***
## chas         0.096542   0.453121   0.213  0.83128
## rm           0.630210   0.222307   2.835  0.00458 **
## age          0.036517   0.006453   5.659 1.52e-08 ***
## lstat        0.047817   0.028372   1.685  0.09192 .
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 701.46  on 505  degrees of freedom
## Residual deviance: 423.97  on 500  degrees of freedom
## AIC: 435.97
##
## Number of Fisher Scoring iterations: 5
```

The use of various subsets of the predictors to fit the models highlights interesting patterns. When nox is excluded, other predictors that had a high p-value as indus and age, see the p-value decrease (showing a strong relationship with the response). So indus and age are surrogates for nox: they get "credit" for the effect of nox on crime rate. This can be explained by the fact that an high precentage of non-retail business acres and houses built prior to 1940 has an effect to increase the level of nox (industries and old houses pollute more). The correlation between those two predictors and nox is high.

```
cor(data$indus, data$nox)
```

```
## [1] 0.7636514
```

```
cor(data$age, data$nox)
```

```
## [1] 0.7314701
```

So indus and age are not relevant in the model including nox because a suburb bearing a high level of nox doesn't have necessarily industries and old houses (new suburbs with high density of population for example).

6. Resampling methods

Question:

Suppose that we use some statistical learning method to make a prediction for the response Y for a particular value of the predictor X. Carefully describe how we might estimate the standard deviation of our prediction.

Answer:

The standard deviation of a predictor, in the case of linear regression, is given by R automatically. But for other statistical learning methods, it can be difficult to obtained and must be computed. It can be obtained using Bootstrap. Bootstrap randomly select n observations from the data set to build a bootstrap data set (sampling with replacement - the same observation can occur more than once in the bootstrap data set) from which the model is fitted. The procedure is repeated B times for some large value of B in order to produce B different bootstrap data sets and B corresponding coefficient estimates. The standard deviation of these bootstrap estimates can be then computed by the root square of the mean square of errors (MSE) of the estimates for all B models.

7. Resampling methods

(a) Generate a simulated data set as follows. In this data set, what is n and what is p? Write out the model used to generate the data in equation form.

```
set.seed(500)
y = rnorm(500)
```

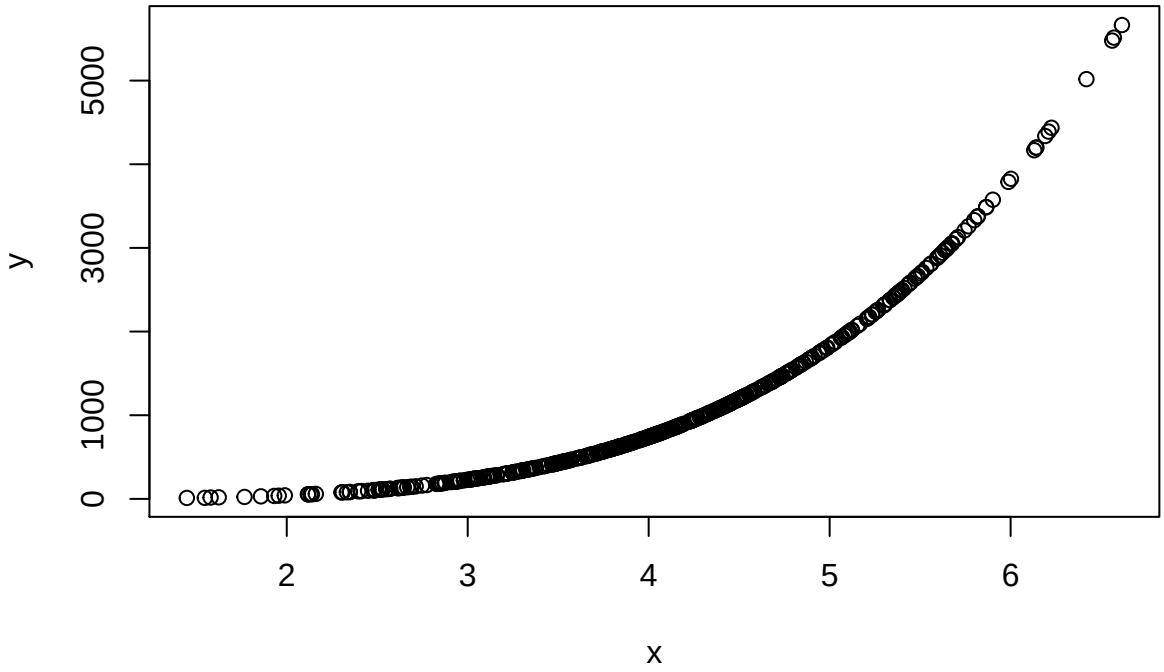
```
x = 4 - rnorm(500)
y = x - 2*x^2 + 3*x^4 + rnorm(500)
```

n is the number of observations in the data set: n = 500 p is the number of predictors inthe data set. p = 3

The model used to generate the data in equation form: Y = X - 2X^2 + 3X^4 + epsilon

(b) Create a scatterplot of X against Y. Comment on what you find.

```
plot(x,y)
```



The plot shows a positive non linear relationship between X and Y.

(c) Set the seed to be 23, and then compute the LOOCV and 10-fold CV errors that result from fitting the following four models using least squares:

```
set.seed(23)
frame <- data.frame(x, y)
library(boot)
```

i. Y = beta_0 + beta_1X + epsilon

- LOOCV:

```
glm.fit = glm(y~x,data=frame)
cv.glm(frame, glm.fit)$delta
```

```
## [1] 155977.1 155974.1
```

- 10-fold CV:

```
glm.fit=glm(y~x,data=frame)
cv.glm(frame,glm.fit,K=10)$delta
```

```
## [1] 155666.9 155528.5
```

ii. Y = beta_0 + beta_1X + beta_2X^2 + epsilon

-LOOCV:

```
glm.fit=glm(y~poly(x, 2), data=frame)
cv.glm(frame,glm.fit)$delta
```

```
## [1] 7400.428 7399.967
```

- 10-fold CV:

```
glm.fit=glm(y~poly(x,2),data=frame)
cv.glm(frame,glm.fit,K=10)$delta
```

```
## [1] 7363.821 7341.746
```

iii. Y = beta_0 + beta_1X + beta_2X^2 + beta_3X^3 + epsilon

-LOOCV:

```
glm.fit=glm(y~poly(x, 3), data=frame)
cv.glm(frame,glm.fit)$delta
```

```
## [1] 64.00805 63.99983
```

- 10-fold CV:

```
glm.fit=glm(y~poly(x,3),data=frame)
cv.glm(frame,glm.fit,K=10)$delta
```

```
## [1] 64.03821 63.60979
```

iv. Y = beta_0 + beta_1X + beta_2X^2 + beta_3X^3 + beta_4X^4 + epsilon

-LOOCV:

```
glm.fit=glm(y~poly(x, 4), data=frame)
cv.glm(frame,glm.fit)$delta
```

```
## [1] 0.9261812 0.9261552
```

- 10-fold CV:

```
glm.fit=glm(y~poly(x,4),data=frame)
cv.glm(frame,glm.fit,K=10)$delta
```

```
## [1] 0.9391511 0.9370416
```

(d) Repeat (c) using random seed 46, and report your results. Are you results the same as what you got in (c)? Why?

```
set.seed(46)
```

i. Y = beta_0 + beta_1X + epsilon

- LOOCV:

```
glm.fit = glm(y~x,data=frame)
cv.glm(frame, glm.fit)$delta
```

```
## [1] 155977.1 155974.1
```

- 10-fold CV:

```
glm.fit=glm(y~x,data=frame)
cv.glm(frame,glm.fit,K=10)$delta
```

```
## [1] 154238 154175
```

ii. Y = beta_0 + beta_1X + beta_2X^2 + epsilon

-LOOCV:

```
glm.fit=glm(y~poly(x, 2), data=frame)
cv.glm(frame,glm.fit)$delta
```

```
## [1] 7400.428 7399.967
```

- 10-fold CV:

```
glm.fit=glm(y~poly(x,2),data=frame)
cv.glm(frame,glm.fit,K=10)$delta
```

```
## [1] 7261.817 7245.260
```

iii. Y = beta_0 + beta_1X + beta_2X^2 + beta_3X^3 + epsilon

-LOOCV:

```
glm.fit=glm(y~poly(x, 3), data=frame)
cv.glm(frame,glm.fit)$delta
```

```
## [1] 64.00805 63.99983
```

- 10-fold CV:

```
glm.fit=glm(y~poly(x,3),data=frame)
cv.glm(frame,glm.fit,K=10)$delta
```

```
## [1] 62.21704 61.89179
```

iv. Y = beta_0 + beta_1X + beta_2X^2 + beta_3X^3 + beta_4X^4 + epsilon

-LOOCV:

```
glm.fit=glm(y~poly(x, 4), data=frame)
cv.glm(frame,glm.fit)$delta
```

```
## [1] 0.9261812 0.9261552
```

- 10-fold CV:

```
glm.fit=glm(y~poly(x,4),data=frame)
cv.glm(frame,glm.fit,K=10)$delta
```

```
## [1] 0.9236741 0.9224638
```

LOOCV is the same. Since it evaluates each observations of the data set, there is no randomness in the training/validation set splits.

10-fold CV is different. Because there is randomness in the selection of observations as they are splitted in folds.

(e) Which of the models in (c) had the smallest LOOCV and 10-fold CV error?

The model that had the smallest LOOCV and 10-fold CV errors is the model with the highest degree of polynomial: Y = beta_0 + beta_1X + beta_2X^2 + beta_3X^3 + beta_4X^4 + epsilon

This result was expected as the plot of X and Y showed a non linear relationship. Therefore increasing the degrees of polynomial makes the linear function used to fit the data more flexible. This reduces the error because the function increasingly fit better the data.

(f) Comment on the statistical significance of the coefficient estimates that results from fitting each of the models in (c) using least squares. Do these results agree with the conclusions drawn based on the cross-validation results?

```
summary(glm(y~poly(x,4),data=frame))
```

```
##
## Call:
## glm(formula = y ~ poly(x, 4), data = frame)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.88395  -0.61614   0.02642   0.64776   2.70284
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.071e+03  4.267e-02 25086.5   <2e-16 ***
## poly(x, 4)1 2.003e+04  9.542e-01 20990.7   <2e-16 ***
## poly(x, 4)2 8.547e+03  9.542e-01  8956.9   <2e-16 ***
## poly(x, 4)3 1.857e+03  9.542e-01  1945.8   <2e-16 ***
## poly(x, 4)4 1.664e+02  9.542e-01   174.3   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for gaussian family taken to be 0.9105089)
##
##     Null deviance: 4.777e+08  on 499  degrees of freedom
## Residual deviance: 4.507e+02  on 495  degrees of freedom
## AIC: 1379
##
## Number of Fisher Scoring iterations: 2
```

p-values of linear and quadatric terms are well below 1%. These results agree with the conclusions drawn based on the cross-validation results (that the highest quadratric model had the smallest LOOCV and 10-fold CV errors).