Project: Analysis and Modelling of AutoMPG Dataset

StudentID: 20153780 Name: Jeevanpreet Singh

## Introduction

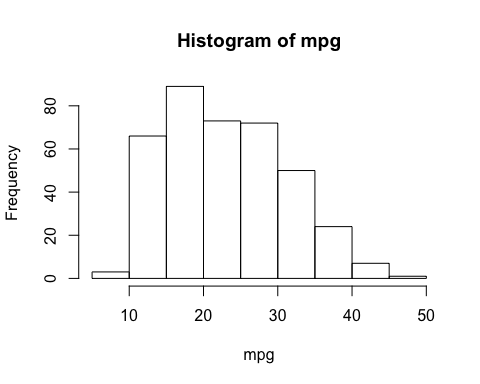
## This project will focalize on a specific dataset, undertaking multiple methods of data analysis and several applications of statistical modelling to identify the best method of predictive modelling, given the attributes of the response variable. The project will consist of 4 parts: Exploratory Analysis, Simple Linear Regression Analysis, Multiple Linear Regression Analysis, and a Conclusion. Results for each modelling method will be interspersed within the analysis of the data. The dataset I’ve decided to use is AutoMPG, provided by the lecturer. AutoMPG is a dataset involving specs of specific cars manufactured between 1970 and 1982. It’s a modified data set that is taken from the library of and currently maintained by Carnegie Mellon University (Ortiz, 2016). We will hone in on specific subgroups that make distinctions between the car characteristics, and specifically each of their individual impacts on car fuel consumption. This is a dataset of interest as there is a distinct response variable to be tested and multiple variations of data types that may have an effect on the response variable (continuous, discrete, string etc.). Furthermore, this study may become of importance to car purchasers or manufacturers in the future who want to identify the most efficient cars that can be utilised in the modern day. Utilising the AutoMPG dataset, I am tasked to identify the relationship between fuel consumption and multiple characteristics of a car. I am to thoroughly investigate the data through exploratory analysis, conduct simple linear regression and multiple linear regression, and decide the most suitable predictive model for this specific dataset. I have to consider and develop a multivariable model between fuel consumption and model year, and indicate any effects of relationships between key explanatory variables. View the beginning of the appendix to see how points are referenced to code chunks.

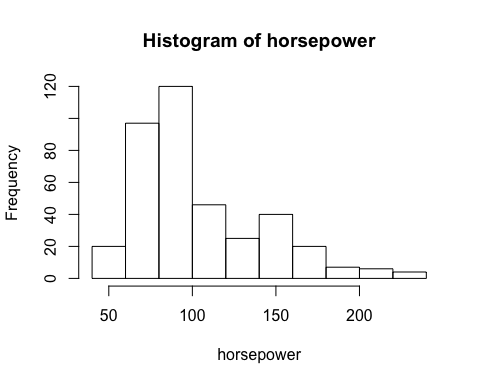
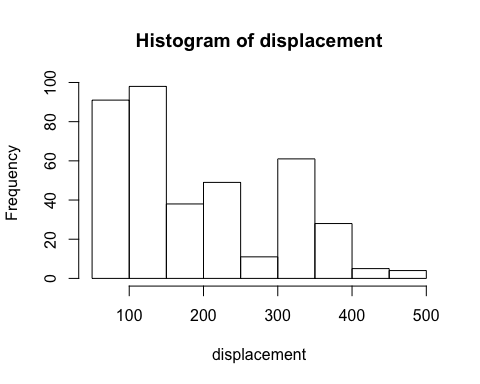
## Exploratory Analysis

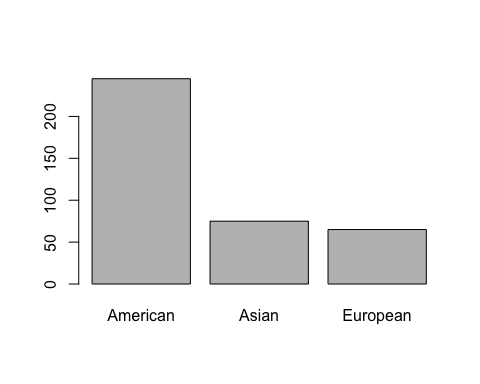
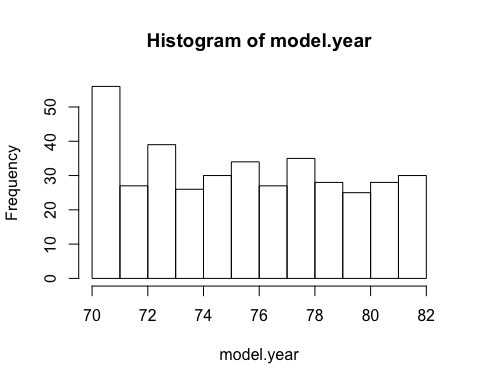
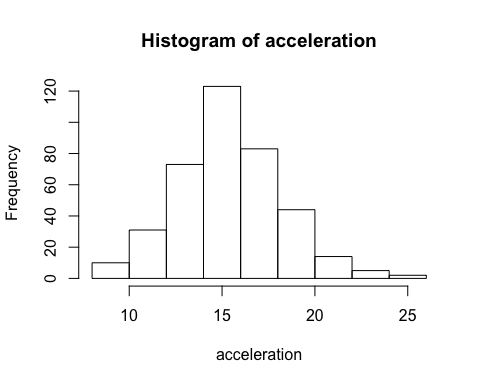
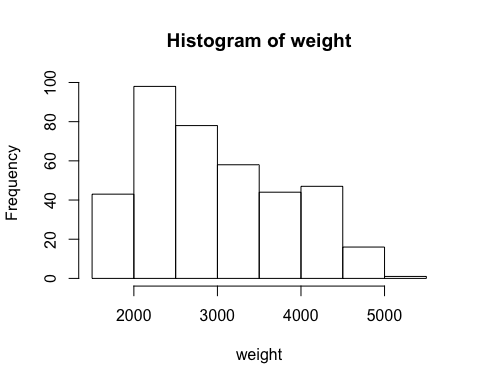
Exploratory Analysis involves the first steps of statistical analysis. It is the best way to understand the variables of the dataset, find points of interests, create tables to indicate significant values and give insight on the approach we should take to apply a statistical model to the data. In other words, we make inferences on how the data appears and the significance of certain variables over others in the dataset. As the project involves the effects of car characteristics **on** fuel consumption, we shall consider mpg (the measure of fuel consumption) as the main response variable, and the remaining variables will be considered the explanatory variables. Firstly (assuming data is fully loaded), we must consider the nature of the variables, and then make certain assumptions about the response variable. There are 385 observations within 9 variables. All of them are numerical variables except origin and car, which are character variables describing the country of origin of the car and the car name respectively. Cylinders and model year are both categorical variables (as year is a discretisation of time and cylinders are naturally discreet and limited). In other words, exactly 385 cars manufactured between 1970 and 1982 are being displayed, along with certain vehicle characteristics such as weight, horsepower, acceleration, etc. The data types are as follows: mpg: miles per gallon, cylinders: multivalued discrete, displacement: continuous, horsepower: continuous, weight: continuous, acceleration: continuous, model year: multivalued discrete, origin: multivalued discreet, car: unique strings.

## Analysing the Data

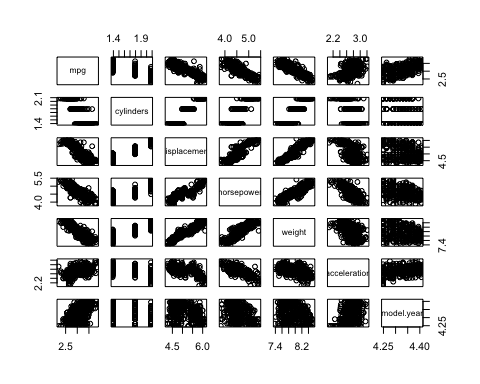
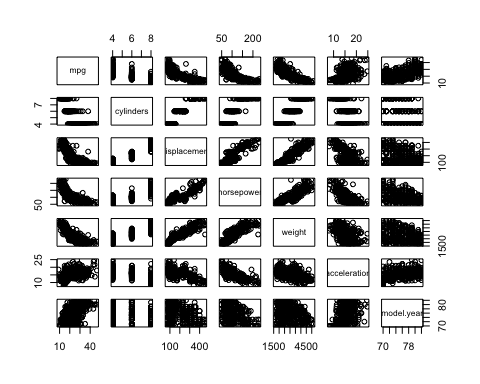
We begin by identifying the distribution of each variable present in the dataset (SEE 1.2). It is optional to use boxplots or histograms, however I’ve chosen to use histograms as it’s easier to interpret the general shape of the distributions. Additionally, we create a subgroup for numerical variables below. For now, we consider identifying the data only as univariate. In other words, we only currently focus on the individual values in each variable rather than the comparison of two or more variables in the data set. A barplot is used for the categorical variable origin, which had to be transformed into a table to allow plotting to occur.

Chart, histogram

Description automatically generated

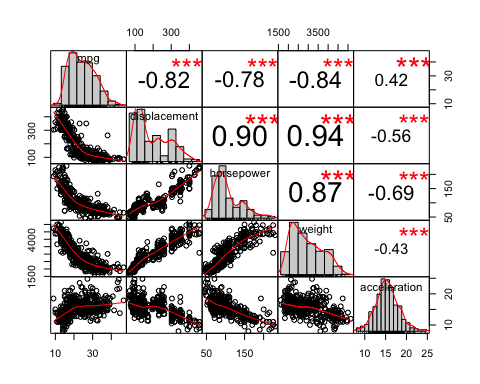


Considering these histograms, mpg is considered to have a distribution skewed to the right. This is an important distinction to remember when it comes to exploring the dataset further through linear regression. All continuous (numerical) variables have a distinctive skewed to the right distribution, except for acceleration, which seems to have a fairly normal distribution. As for the categorical variables (cylinder, model year, origin), they’re also skewed to the right, however we do not interpret the similarly to the numerical variables. For example, origin’s skewed distribution only occurs as vehicles are characterised within their manufacture countries. From an initial exploratory analysis between the explanatory variables and the response variables (SEE 1.3), there seems to be some indication of skewed distributions and nonconstant variance (looking at the top row). A log transformation appears to strengthen the assumptions of nonconstant variation and normal distribution. This is a possible transformation that we could consider moving forward with the dataset.



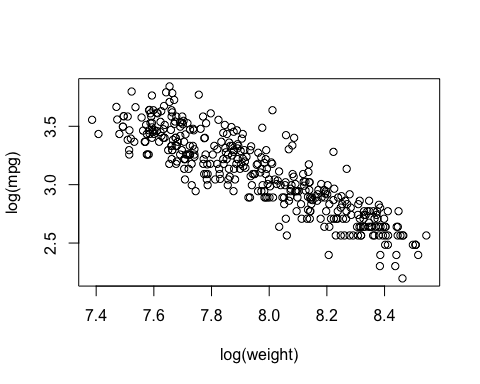
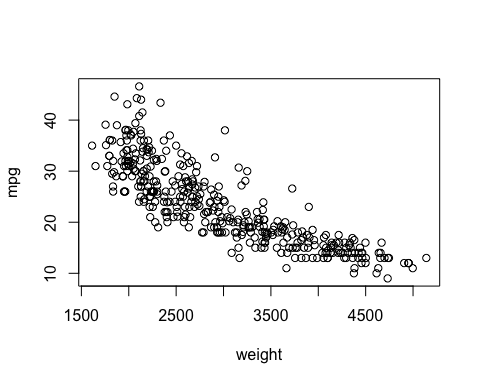
Relationship between variables Relationship between log variables

Now, we will identify bivariate relationships within the dataset. This is done through correlation charting (SEE 1.4), in which both, the graphical representation within each set of variables and their correlation coefficients, are displayed. We now consider the correlation between numerical variables, excluding categorical variables (model year and cylinders).

 The figure above outputs very interesting results for interpretations. The response variable mpg has a considerable strong negative correlation with all the explanatory variables, except for acceleration. This could be explained due to the normal distribution appearance of acceleration we had viewed above, in which every other variable fitted a skewed to the right distribution, similar to the mpg response variable. The explanatory numerical variable with the strongest correlation with mpg is weight. We should consider this as the best predictor moving forward with our regression analysis. Before we do that, we can also consider the discrete values and their impact on the response variable too (SEE 1.5), as well as on each other. Again, considering they are discreet variables, it would be inappropriate to test their correlation considering that is only utilized for continuous variables. Even considering their correlation, it is not as strong as the correlation of the weight variable, which seems to be the most appropriate predictor to undergo simple linear regression. The cylinders variables appears to have too little a sample and model year has skewed to the right distribution (indicating most cars in this dataset were produced at the beginning of the 1970-82 period).

If we take a closer look at the bivariate relationship between weight and mpg (SEE 1.6), we can see that although the relationship is moderately to strongly negative (higher values of one variable leads to lower values of the other), the line appears to be non-linear. This indicates that transformation must occur to allow a better fit for the data. In terms of outliers, there aren’t any values that significantly disturb the general pattern of the data, as the scatter seems consistent throughout the graph, although there is more scatter occuring at the lower weight than there is at heavier weights. The scale of the plot seems sufficient for the values being presented. We can quickly validate the correlation of these variables through hypothesis testing. As previously mentioned, the log transformation appears to eliminate the curve relationship and produce a far more linear relationship (SEE 1.7).

Mpg vs Weight log(Mpg vs Weight)

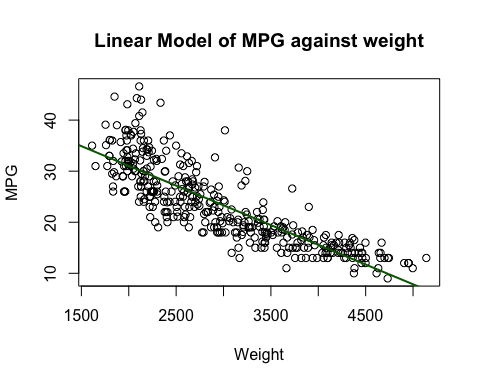


When discussing the correlation between the numerical explanatory variables and the response variable, it is important to identify whether assumptions of correlation are met. As previously displayed, mpg follows a skewed distribution, not a Normal distribution, thus violates the assumption of bivariate normality, in which two variables can be correlated dependent on their normal distributions matching up to result in another normal distribution. There are three correlation methods; Pearson, Spearman, and Kendall. They all lie within -1 (extreme negative correlation) and +1 (extreme positive correlation). As Pearson’s assumption of bivariate normality is violated, we turn to Spearman. We will consider a single numerical explanatory variable to show the process of the 6 steps of hypothesis testing. Weight is most appropriate (SEE 1.8). We see a correlation coefficient of -0.8864761, which is a fairly strong negative correlation. We may perform a hypothesis testing according to these results. Step 1:: mpg and weight are independent, : mpg and weight are correlated. Step 2: Test statistic = 17942350 (seeing R output) Step 3: Sampling distribution based on ranks. Step 4: P-value: 0 (or infinitely small) Step 5: P-value < 0.05 (5% significance level). Step 6: We reject the Null hypothesis, there is correlation between the two variables.

As we conclude our exploratory analysis of our dataset, now we must approach simple linear regression, taking into account everything that we have learnt from our explanatory variables and our response variable. We have successfully identified correlation, however this is different from simple linear regression in the sense that a correlation analysis identifies the strength and direction of a bivariate relationship, while simple linear regression is utilised to identify parameters in a linear model which can estimate the values of a variable in response to another.

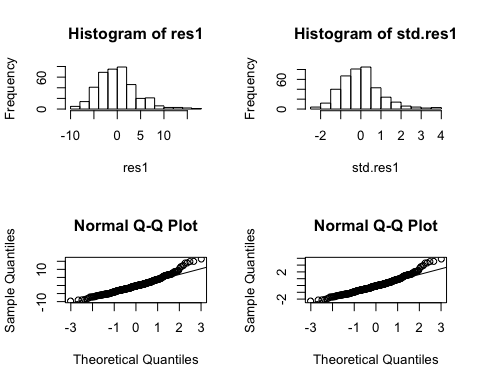
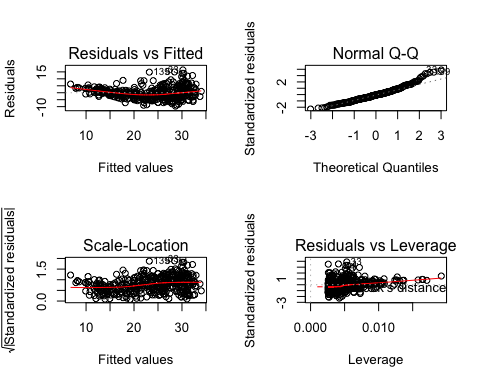
**Simple Linear Regression:**

There are a couple things that need to be done in Simple Linear Regression. There has to be univariate modelling of at least one key explanatory model (which was identified in the explanatory anlysis). Furthermore, we must distinguish applications towards differing variables, taking into account any transformations or removal of outliers that may further improve the fit of the regression, extending into diagnostics checking. Although we have considered weight to be the strongest correlated explanatory variable to the mpg, we can further consider it as the best model to pick through our R-squared values when undertaking linear modelling for each numerical preditor variable. The R-squared values gives insight on how much a certain predictor variable can explain the response variable (from 0-100%). We will begin by performing simple linear regression and diagnostics checking on the weight predictor variable, and compare it to the alternative predictor variables.

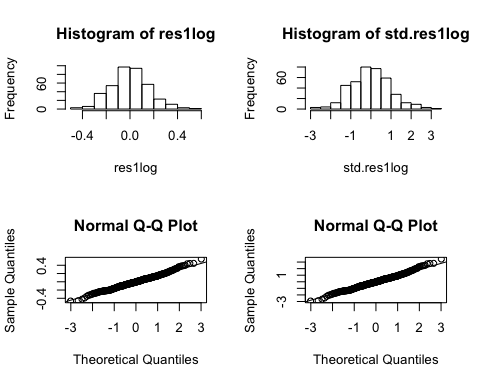
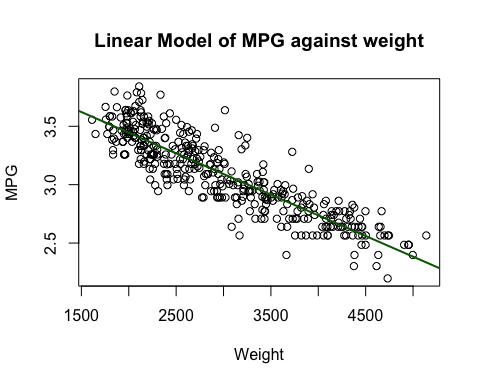


As provided from the Linear model above (SEE 2.1), we have produced a regression slope as well as provided summary statistics to help us further accentuate the relationship between the two variables. We have the y-intercept of 46.5027 and slope (gradient) value of -0.0077. In other words, for any vehicle we may have, the average miles per gallon will be 46.5027 (however this is not interpretable at 0 weight). The slope however indicates that whenever an additional unit of weight is added, it is predicted to reduce 0.0077 miles per gallon for the vehicle. Furthermore, we can calculate the 95% confidence interval of each value. As such, we are 95% confident that the intercept lies between 44.96 and 48.04 while the slope lies between -0.0082 and -0.0072. We can do a hypothesis testing utilising the regression line to verify the relationship between mpg and weight, the null hypothesis being the slope is 0, and the alternative hypothesis being the slop is less than 0. As the p-value is far less than 0.05 (SEE 2.2), we reject the null hypothesis and conclude that there is a statistically significant negative relationship between weight and mpg.

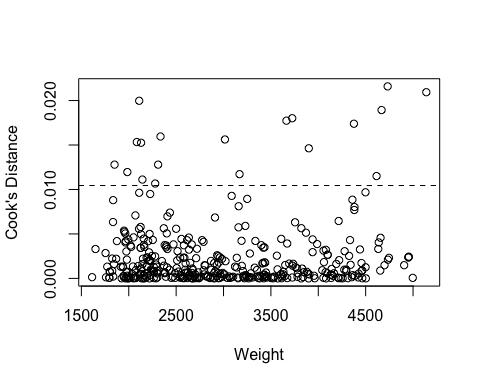
Now, we will consider the R-squared values for the simple linear regression model of other predictor variables (SEE 2.3). We can now compare the R-squared values within the numerical predictor variables. Displacement, horsepower and acceleration variables output an R-squared value of 0.6689, 0.6088 and 0.1758 respectively. It is evident that the R-squared value of weight surpasses all of the other predictor variables, making it the best candidate for the regression modelling (weight achieved an R-squared value of 0.7101, indicating that 71% of the mpg is effectively explained by the weight.) Diagnostics checking can now occur to possible find a better fit for the data.



Through residual analysis (SEE 2.4), the histogram displays a slightly skewed to the right, but we can consider the distribution to appear symmetrical, thus normality is satisfied. The quantile plots show that the points mainly stay with the line aside from the larger values which drift off. Considering the residuals, there is clear evidence that the variance of errors increases with x, this causes an issue with our model in terms of avoiding relationships between residual and our explanatory variables. The plots of residuals and standardised residuals (SEE 2.5) show clear evidence of a relationship between the explanatory variable and its variance of errors. This is indicated by the non-constant scatter at different values of the weight variable. We must now identify any influencing points and eliminate them to enhance our fit, hopefully resulting in an improved R-squared value. We can precede this by attempting transformations (SEE 2.6) and applying diagnostics checking with it (SEE 2.7)..



A logarithmic transformation of both variables increased the R-squared value output to 0.7617, significantly improving the fit of the linear regression model we initially displayed. The intercept changes to 4.154 (with 95% confidence of points lying between 4.096 and 4.212) and the slope changes to -0.000354 (with 95% confidence of points lying between -0.000373 and -0.000335). We can confirm the better fit by analysing the residuals for this new model. The histogram appears closer to normal distribution (confirming Normality), the quantile plots fit more points on the line and the scatter is far more constant across the fitted values (bottom left), ie. constant variance. As such, we consider this model more appropriate. We can now look to remove any influential points that significantly impact the fit of the SLR model between our predictor variable and the response variable. We successfully figure out the 19 outliers/leverage points that influence the model. (SEE 2.8). We compare Cook’s distances again and we see that the high leverage points are removed. We will see the effect it will have on the linear model, as well as the R-squared output indicating the goodness of the fit.

Chart, scatter chart

Description automatically generated

Through the log transformation and removal of outliers, diagnostics checking efficiently improved the R-squared value from an initial 0.7101 to a value of 0.8114. In other words, 81% (SEE 2.9) of the mpg is explained by the weight, due to our newer and improved linear regression model.

**Multiple Linear Regression:**

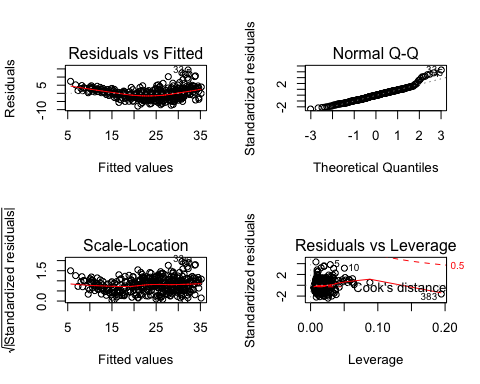
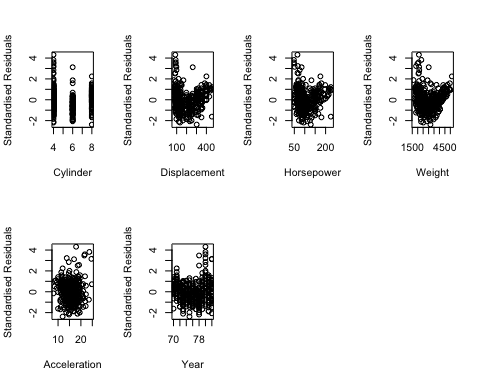
For this dataset, we now want to consider the relationship between the response variable and a multitude of explanatory variables. The previous pair scatter plots are not enough to determine the relationships between predictors, which may have a significant effect on the regression model. We can begin our Multiple Linear Regression by first creating a model and then investigating parameter estimates as well as their covariance (SEE 3.1). We have created a multivariabe model including year, cylinders, displacement, horsepower, weight and acceleration. The R-squared value indicates that 82.08% of mpg can be identified by this regression model. To interpret the slope, we must consider that all other variables are held constant while in calculation. Our initial regression model is:

mpg = -12.32 - 0.4863 Cylinders + 0.0051 Displacement + 0.004435 Horsepower - 0.0066 Weight + 0.02260 Acceleration + 0.7404 Years.

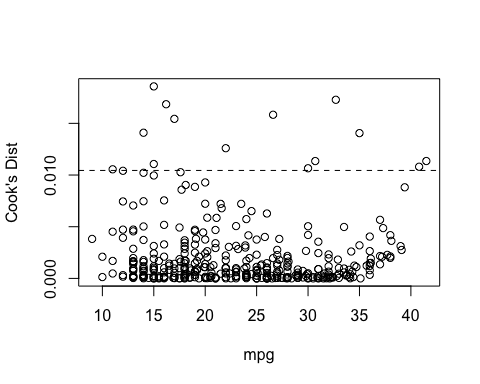
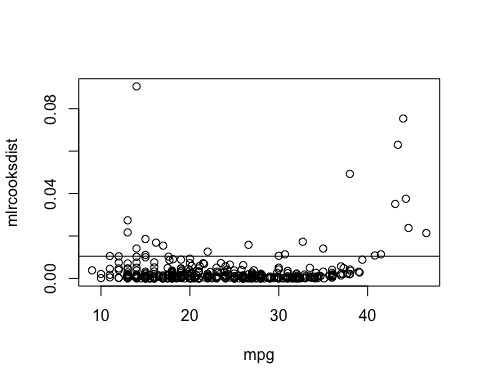
We can consider acceleration as the most statistically significant predictor variable in relation to the response variable due to its regression coefficient being the highest(slope). Matrix implementation of the regression model has also been done (SEE 3.2), solving for variance (which is possibly useful for anova F-testing). We will not alter or remove less statistically significant variables at this time. We must now consider the tradeoff between limiting the sum of squares error and the number of explanatory variables present in the regression model. Removing statistically insignificant variables will decrease error however it is necessary to distinguish whether all predictor variables must be present to allow the most accurate MLR Regression model. This can be applied through ANOVA analysis and Partial F tests, which compares the ratios of the sum of squares. We will consider horsepower as the variable to be omitted and tested against for this F-test. Before we begin, we must undergo analysis of variance of the initial MLR model to prove at least one variable is statistically significant in the regression model (SEE 3.3).

mpg.lmm=lm(mpg~1,data=auto1) ## model with intercept only  
anova(mpg.lmm, mpg.lm1)

The null hypothesis is that all intercepts equal zero while the alternative hypothesis states the values are not 0. Due to the p-value below the 5% significance level, we reject the null hypothesis and conclude that at least one of the variables is statistically significant to the response mpg variable. The outputted analysis variance (comparing the model with the intercept) indicates the following values: n=385, p=6, SSR=19358,F=288.53,p-value < 2.2^10-6. We may now conduct the Partial F-test to consider the additional variable horsepower (SEE 3.4). The P-value (0.7434) fails to reject the null hypothesis that horsepower is statistically significant in our multiple linear regression model, thus it is not necessary to include it. We can begin the final step of our modelling process, diagnostics checking. This will done through utilizing residual analysis (SEE 3.5). The Normality, constant variance and independence assumptions are met for the residuals of the model. However there does seem to be some relationship between standardised residuals and fitted values. Furthermore, the q-q plot shows more significant residuals at the higher end than the lower. We can further plot standardised residuals against our predictor variables as part of our diagnostics checking (SEE 3.6)

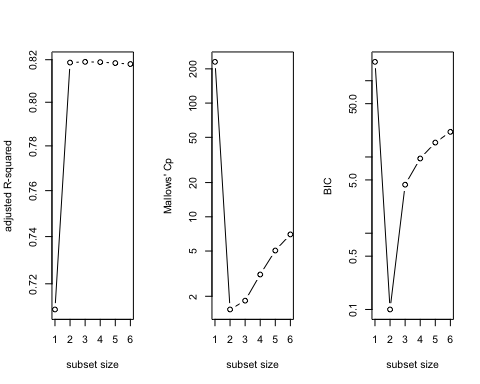
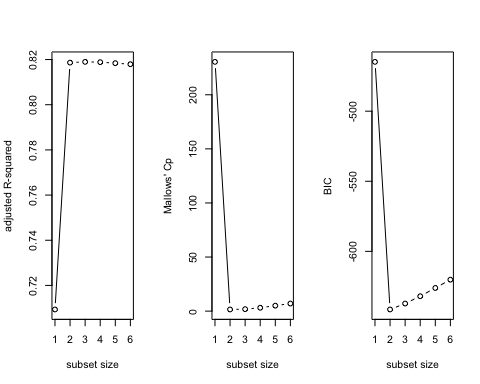
.

We can determine that outliers must be omitted for a better regression model fit (SEE 3.7).

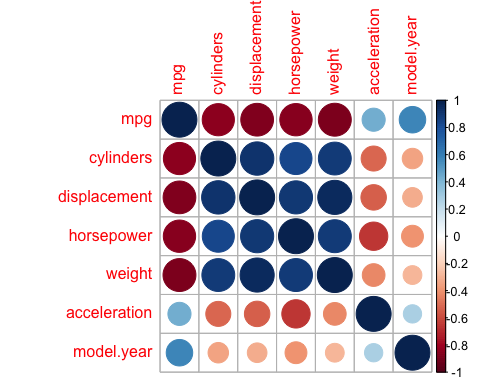


By omitting the top 10 outliers, it has successfully increased the R-squared value (SEE 3.8), and thus enhanced the fit of the regression model. Completing diagnostics checking, we may want to see if any categorical value is of interest, specifically, the origin of manufacturer (SEE 3.9). The slope demonstrates the difference in the statistical significance of each manufacturer origin and its impact on the mpg response variable. Considering Asian cars have the highest regression coefficient, they are the most statistically significant to the response variable at hand.

Relating back to the main linear regression model and dataset, we must utilise model-building techniques to identify which combination of explanatory variables provide increased predictive ability. All subsets will be utilised as it can optimise any function we want to limit or increase (SSE, R-squared, etc.) (SEE 3.10).



We can interpret the results of this all-subsets modelling technique. (Relating to the **right** 3 plots), as subset size increases (number of predictor variables in the model), the adjusted R-squared value increases heavily yet slowly decreases as more are added. As subset size increases, the Mallow’s CP and BIC are both rapidly decreasing, yet rise as the subset size increases further. BIC rises more significantly than Mallow’s CP. (Relating to the **left** 3 plots) A logarithmic transformation returns the same trends(SEE 3.11), except the Mallow’s CP and BIC increase more rapidly as the subset increases past 2 variables. Finally, we must consider the correlation between the multiple explanatory variables we have chosen. Multi-collinearity may have adverse impacts on the fitting of regression models that we must consider. Highly correlated variables tend to have inflated variances and biased parameter estimates (SEE 3.12).



Considering 5 is the usual cut-off value, we consider the regression coefficient to be highly correlated due to their extremely inflated variance values (looking at the vif function). Furthermore, the corrplot function has identified strong correlation within the explanatory variables except for acceleration, which was initially determined in the exploratory analysis section and now confirmed.

**Conclusion Discussion:**

Through the explanatory analysis, simple linear regression and multiple linear regression of the AutoMPG dataset, I have successfully interpreted the multiple models utilized for investigating large datasets. Exploratory analysis involved the general understanding of data variables and correlations, as well as providing the key predictor variable and its relationship with the response variable. SLR (Simple Linear Regression) extends upon the explanatory analysis by creating a linear model within to numerical variables to determine the best relationship possible, in which diagnostics checking and general comparisons between variable subgroups enhanced the predictive ability of the specified model. Multiple Linear Regression followed similarly to SLR except the inclusion of multiple explanatory variables in the regression model to determine which may be the best predictor variable according to their correlation coefficient (further diagnostics checking was undertaken to improve the model as for SLR). Transformations and removal of outliers was considered in every section and their impacts on the regression model accuracy were also accounted for, while MLR required the use of further subset selection to determine which combination of variables provided the strongest relationship, also taking into account the multicollinearity within predictor variables which may impact the results of the regression model.

It is particularly insightful to understand what this kind of study on this specific dataset will do for the vehicle industry. Car purchasers may consider these variables (ie. car characteristics) to select a car with the optimum fuel efficiency, while car manufacturers may utilize the models to predict future trends in cars or apply predictive models to create the most efficient car in the future. Limitations of the study include the age of the dataset, as modern cars are hardly a comparison to vehicles manufactured in the 70s. Furthermore, the classification of vehicles (such as origin, model year, etc) will have limited analysis due to numerical variables being best suited to the regression models being displayed. Finally, there are a number of differing opinions or methods to find the most suitable predictive model. Although the results interspersed within the data analysis shows clear indication of certain predictor variables (car characteristics) being far more statistically significant over others (in relation to the miles per gallon response variable), this may not always be the case in real life. A majority of regression models will not give a perfect R-squared value, and thus no model can predict whether a response variable can be 100% predicted by a certain variable or subgroup of variables.

**References:**

Regression code and data analysis are derived from:

Practical: [2020-S2-STAT1006-Week5.docx] from Curtin University| Non-Parametric and Regression Inference, STAT1007, accessed on 2020

Practical: [2020-S2-STAT1006-Week6.docx] from Curtin University| Non-Parametric and Regression Inference, STAT1007, accessed on 2020

Practical: [2020-S2-STAT1006-Week7.docx] from Curtin University| Non-Parametric and Regression Inference, STAT1007, accessed on 2020

Practical: [2020-S2-STAT1006-Week8.docx] from Curtin University| Non-Parametric and Regression Inference, STAT1007, accessed on 2020

Practical: [2020-S2-STAT1006-Week9.docx] from Curtin University| Non-Parametric and Regression Inference, STAT1007, accessed on 2020

Practical: [2020-S2-STAT1006-Week10.docx] from Curtin University| Non-Parametric and Regression Inference, STAT1007, accessed on 2020

Practical: [2020-S2-STAT1006-Week11.docx] from Curtin University| Non-Parametric and Regression Inference, STAT1007, accessed on 2020

Lecture: [2020-Wk5-Correlation1.pdf] from Curtin University| Non-Parametric and Regression Inference, STAT1007, accessed on 2020

Lecture: [2020-FinalWk6-SLR.pdf]from Curtin University| Non-Parametric and Regression Inference, STAT1007, accessed on 2020

Lecture: [2020-Wk7-SLR-Pred.pdf] from Curtin University| Non-Parametric and Regression Inference, STAT1007, accessed on 2020

Lecture: [2020-Wk8-SLR-Diag.pdf] from Curtin University| Non-Parametric and Regression Inference, STAT1007, accessed on 2020

Lecture: [2020-Wk9-MLR.pdf] from Curtin University| Non-Parametric and Regression Inference, STAT1007, accessed on 2020

Lecture: [2020-Wk10-MLR\_Part1.pdf] from Curtin University| Non-Parametric and Regression Inference, STAT1007, accessed on 2020

Lecture: [2020-Wk11-Var Selection.pdf] from Curtin University| Non-Parametric and Regression Inference, STAT1007, accessed on 2020

Ortiz, J (2016). *Auto MPG Data Set.* Data.world. <https://data.world/databeats/auto-mpg-data-set>

StackExchange(2015). *Removing outliers based on cook's distance in R Language.* StackExchange. <https://stats.stackexchange.com/questions/164099/removing-outliers-based-on-cooks-distance-in-r-language>

Project Appendix

To interpret the appendix, each chunk that I will refer to in the main project will begin with a number from #1.1 to #3.12. It will appear as (SEE #().()) in the main project.

EXPLORATORY ANALYSIS

#1.1

(load('AutoMPG.RData'))

## [1] "AutoMPG"

auto <- AutoMPG  
head(auto) #familiarise myself with the data

## mpg cylinders displacement horsepower weight acceleration model.year  
## 1 26.0 4 97 46 1835 20.5 70  
## 2 26.0 4 97 46 1950 21.0 73  
## 3 43.1 4 90 48 1985 21.5 78  
## 4 44.3 4 90 48 2085 21.7 80  
## 5 43.4 4 90 48 2335 23.7 80  
## 6 29.0 4 68 49 1867 19.5 73  
## origin car  
## 1 European "volkswagen 1131 deluxe sedan"  
## 2 European "volkswagen super beetle"  
## 3 European "volkswagen rabbit custom diesel"  
## 4 European "vw rabbit c (diesel)"  
## 5 European "vw dasher (diesel)"  
## 6 European "fiat 128"

summary(auto)

## mpg cylinders displacement horsepower weight   
## Min. : 9.00 Min. :4.000 Min. : 68.0 Min. : 46.0 Min. :1613   
## 1st Qu.:17.00 1st Qu.:4.000 1st Qu.:105.0 1st Qu.: 75.0 1st Qu.:2223   
## Median :23.00 Median :4.000 Median :151.0 Median : 93.0 Median :2807   
## Mean :23.45 Mean :5.501 Mean :196.1 Mean :104.7 Mean :2983   
## 3rd Qu.:29.00 3rd Qu.:8.000 3rd Qu.:302.0 3rd Qu.:129.0 3rd Qu.:3630   
## Max. :46.60 Max. :8.000 Max. :455.0 Max. :230.0 Max. :5140   
## acceleration model.year origin car   
## Min. : 8.00 Min. :70.00 Length:385 Length:385   
## 1st Qu.:13.90 1st Qu.:73.00 Class :character Class :character   
## Median :15.50 Median :76.00 Mode :character Mode :character   
## Mean :15.54 Mean :75.96   
## 3rd Qu.:17.00 3rd Qu.:79.00   
## Max. :24.80 Max. :82.00

str(auto)

## 'data.frame': 385 obs. of 9 variables:  
## $ mpg : num 26 26 43.1 44.3 43.4 29 31 29 32.8 44 ...  
## $ cylinders : num 4 4 4 4 4 4 4 4 4 4 ...  
## $ displacement: num 97 97 90 90 90 68 76 85 78 97 ...  
## $ horsepower : num 46 46 48 48 48 49 52 52 52 52 ...  
## $ weight : num 1835 1950 1985 2085 2335 ...  
## $ acceleration: num 20.5 21 21.5 21.7 23.7 19.5 16.5 22.2 19.4 24.6 ...  
## $ model.year : num 70 73 78 80 80 73 74 76 78 82 ...  
## $ origin : chr "European" "European" "European" "European" ...  
## $ car : chr "\"volkswagen 1131 deluxe sedan\"" "\"volkswagen super beetle\"" "\"volkswagen rabbit custom diesel\"" "\"vw rabbit c (diesel)\"" ...

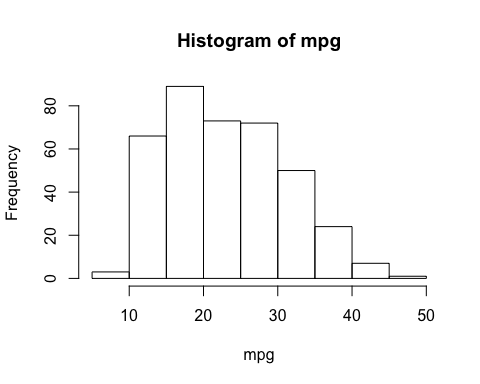
dim(auto)

## [1] 385 9

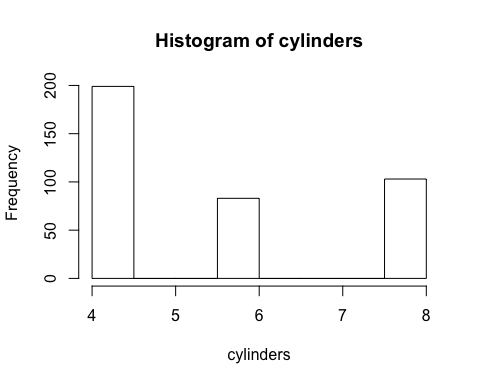
mpg <- AutoMPG$mpg #separating data  
cylinders <- AutoMPG$cylinders  
displacement <- AutoMPG$displacement  
horsepower <- AutoMPG$horsepower  
weight <- AutoMPG$weight  
acceleration <- AutoMPG$acceleration  
model.year <- AutoMPG$model.year  
origin <- AutoMPG$origin  
car <- AutoMPG$car  
#View(auto)

#1.2

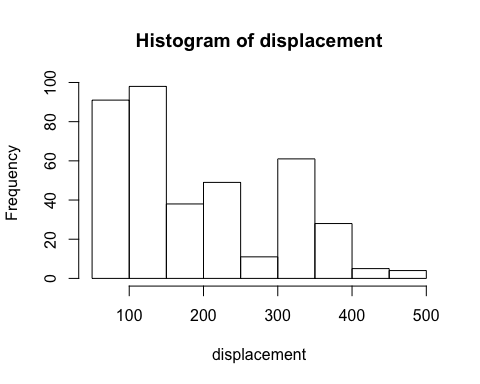
hist(mpg)



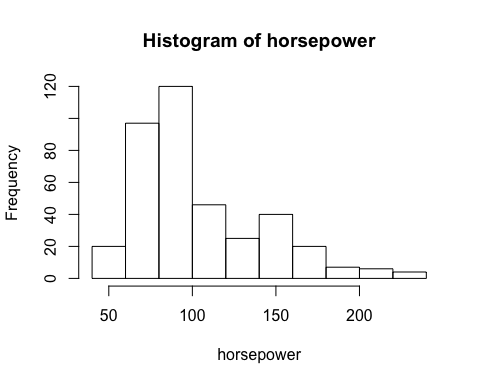
hist(cylinders)



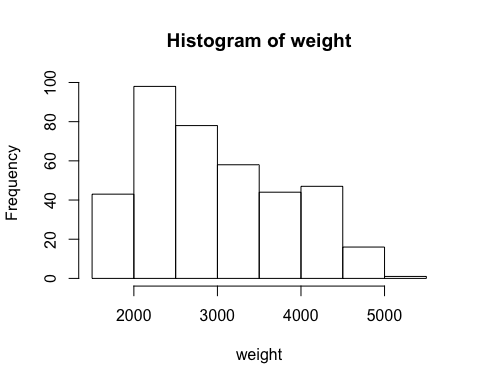
hist(displacement)



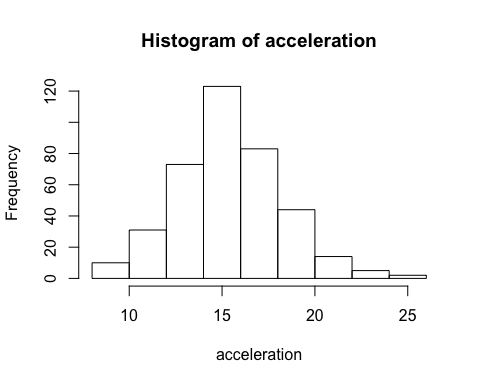
hist(horsepower)



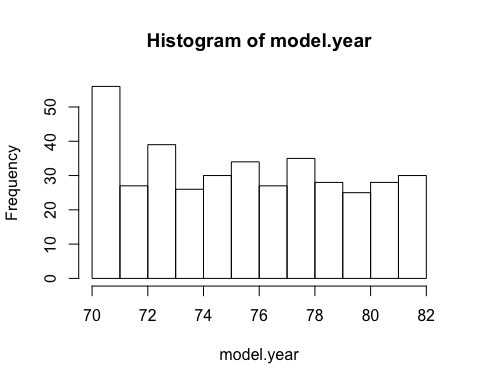
hist(weight)



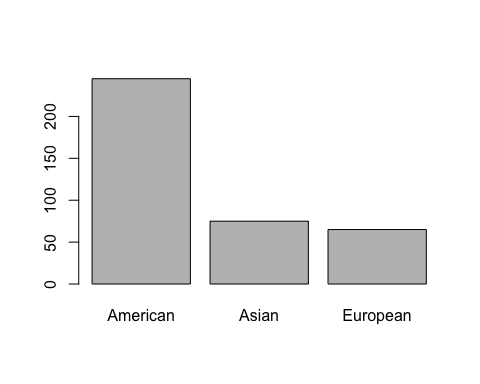
hist(acceleration)



hist(model.year)



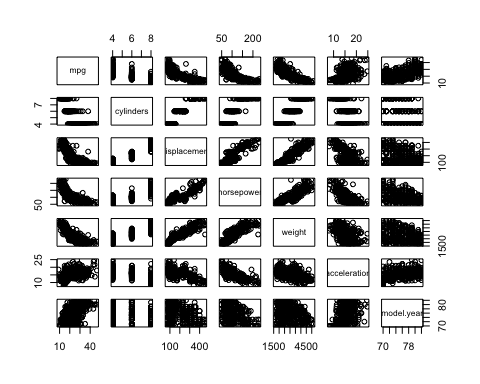
origintable <- table(AutoMPG$origin)  
barplot(origintable)



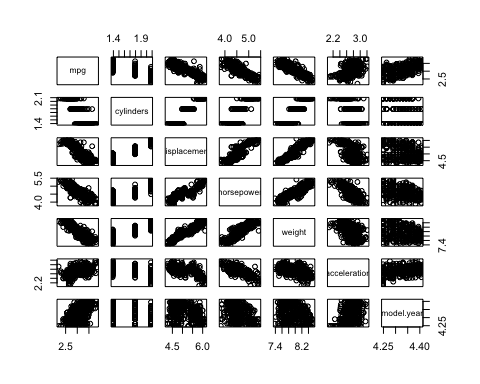
originf <- factor(auto$origin, labels = c("American", "Asian", "European"))  
#View(originf)  
#hist(originf)  
#Car isn't measured as it is a unique string.

#1.3

auto1<- (auto[, c(-8,-9)])  
pairs(auto1[, -mpg])



pairs(log(auto1[, -mpg]))



#1.4

autonum <- auto[,c(-2,-7,-8,-9)]  
autonum1 <- auto[, c(-8,-9)] #includes numerical categorical variables.  
library(PerformanceAnalytics)

## Loading required package: xts

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

##   
## Attaching package: 'PerformanceAnalytics'

## The following object is masked from 'package:graphics':  
##   
## legend

#View(autonum)  
pairs(autonum) #appendix

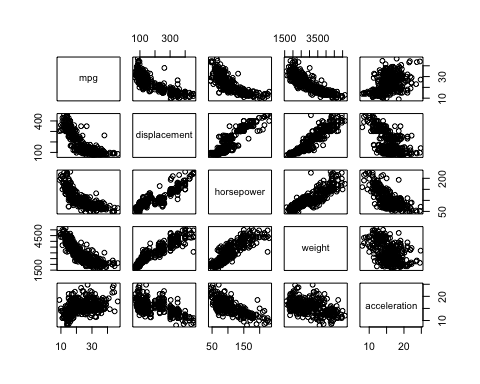
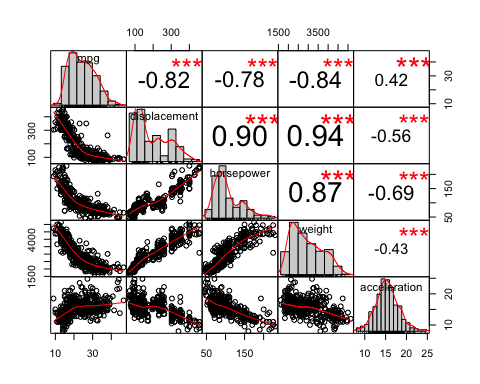


chart.Correlation(autonum, histogram = TRUE, pch = 19)



#1.5

autocat <- auto[,c(-3,-4,-5,-6,-8,-9)]   
#View(autocat)  
pairs(autocat)

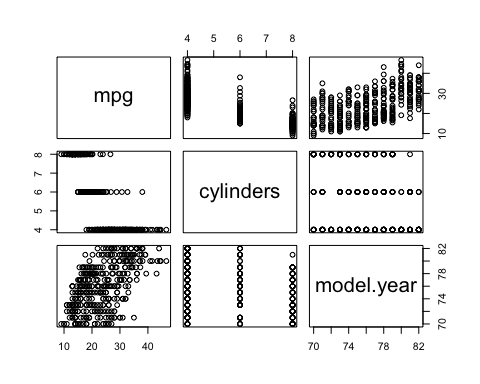
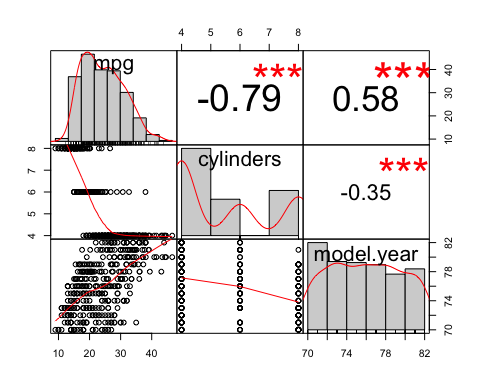
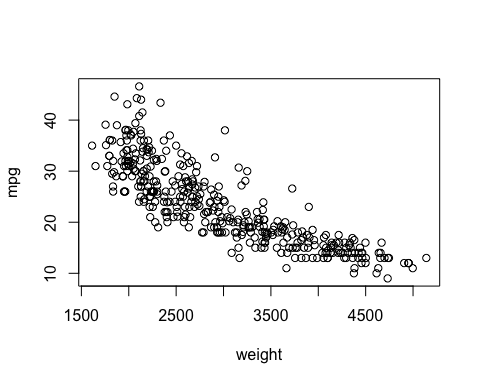


chart.Correlation(autocat)



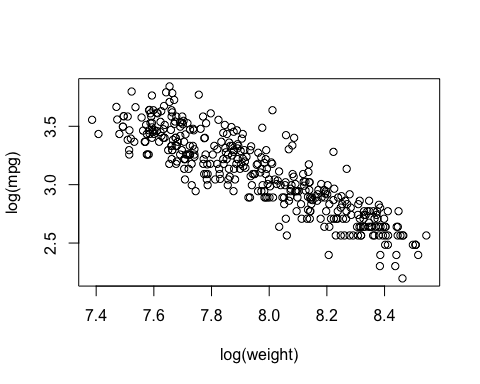
#1.6

plot(mpg ~ weight)



#1.7

plot(log(mpg) ~ log(weight))



#1.8

cor.test(mpg,weight, method="spearman", exact=F)

##   
## Spearman's rank correlation rho  
##   
## data: mpg and weight  
## S = 17942350, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## -0.8864761

pval = pt(17942350, df = 383, lower.tail = F)  
pval

## [1] 0

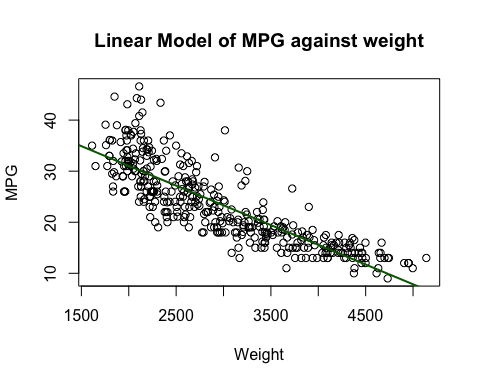
SIMPLE LINEAR REGRESSION

#2.1

mpgweight.lm <- lm(mpg ~ weight)  
summary(mpgweight.lm)

##   
## Call:  
## lm(formula = mpg ~ weight)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -9.6452 -2.7225 -0.3547 2.0839 16.4087   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 46.5027485 0.7829598 59.39 <2e-16 \*\*\*  
## weight -0.0077305 0.0002524 -30.63 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.225 on 383 degrees of freedom  
## Multiple R-squared: 0.7101, Adjusted R-squared: 0.7094   
## F-statistic: 938.2 on 1 and 383 DF, p-value: < 2.2e-16

plot(mpg ~ weight, xlab='Weight', ylab = 'MPG', main = 'Linear Model of MPG against weight')  
abline(mpgweight.lm, lwd = 2, col = 'darkgreen')



confint(mpgweight.lm)

## 2.5 % 97.5 %  
## (Intercept) 44.963310698 48.042186251  
## weight -0.008226782 -0.007234315

cor(mpg, weight)

## [1] -0.8426809

#2.2

pt(-0.0077305/0.0002524, 383, lower.tail = T)

## [1] 2.558678e-105

#2.3

mpgdisplacement.lm <- lm(mpg ~ displacement)  
summary(mpgdisplacement.lm)

##   
## Call:  
## lm(formula = mpg ~ displacement)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -10.0397 -2.9823 -0.5086 2.2177 18.5902   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 35.44553 0.48890 72.50 <2e-16 \*\*\*  
## displacement -0.06121 0.00220 -27.82 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.515 on 383 degrees of freedom  
## Multiple R-squared: 0.6689, Adjusted R-squared: 0.6681   
## F-statistic: 773.9 on 1 and 383 DF, p-value: < 2.2e-16

mpghorsepower.lm <- lm(mpg ~ horsepower)  
summary(mpghorsepower.lm)

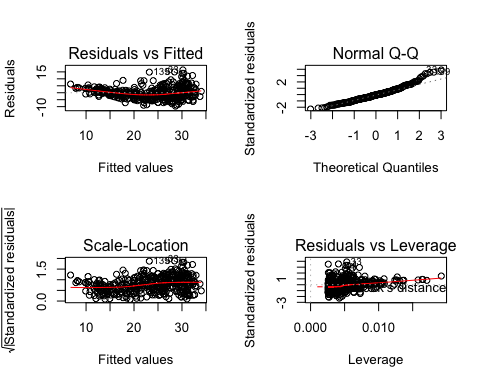
##   
## Call:  
## lm(formula = mpg ~ horsepower)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -13.6041 -3.2976 -0.3198 2.7382 16.8915   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 39.963858 0.721350 55.40 <2e-16 \*\*\*  
## horsepower -0.157775 0.006462 -24.41 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.908 on 383 degrees of freedom  
## Multiple R-squared: 0.6088, Adjusted R-squared: 0.6078   
## F-statistic: 596 on 1 and 383 DF, p-value: < 2.2e-16

mpgacceleration.lm <- lm(mpg ~ acceleration)  
summary(mpgacceleration.lm)

##   
## Call:  
## lm(formula = mpg ~ acceleration)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -17.975 -5.731 -1.186 4.796 23.231   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.907 2.083 2.356 0.019 \*   
## acceleration 1.193 0.132 9.040 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.124 on 383 degrees of freedom  
## Multiple R-squared: 0.1758, Adjusted R-squared: 0.1737   
## F-statistic: 81.72 on 1 and 383 DF, p-value: < 2.2e-16

#2.4

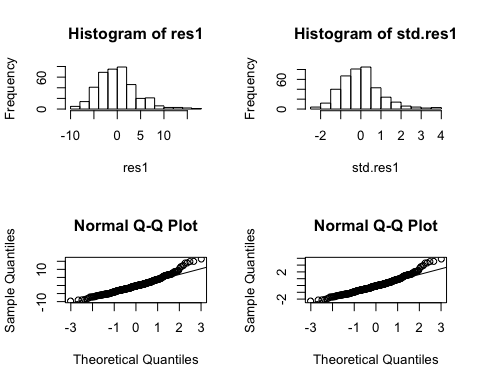
par(mfrow=c(2,2))  
plot(mpgweight.lm)



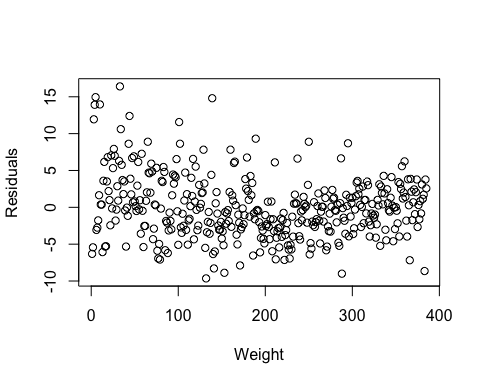
names(mpgweight.lm) ## checking the output of the model

## [1] "coefficients" "residuals" "effects" "rank"   
## [5] "fitted.values" "assign" "qr" "df.residual"   
## [9] "xlevels" "call" "terms" "model"

res1=mpgweight.lm$residuals  
std.res1=rstandard(mpgweight.lm) ## standardised residuals  
par(mfrow=c(2,2)) ## plotting 4 plots to checkk normality  
hist(res1)  
hist(std.res1)  
qqnorm(res1)  
qqline(res1)  
qqnorm(std.res1)  
qqline(std.res1)

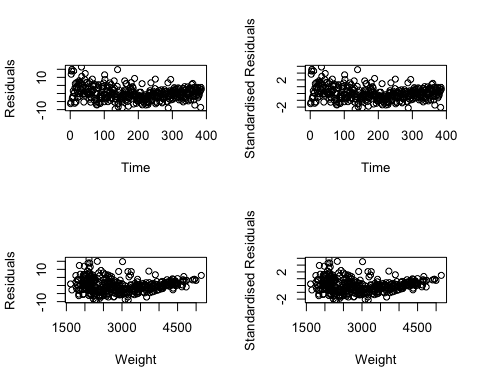


par(mfrow=c(1,1))  
plot(res1, xlab="Weight", ylab="Residuals")



#2.5

par(mfrow=c(2,2))  
plot(res1, xlab="Time", ylab="Residuals") ## Residuals vs Time  
plot(std.res1,xlab="Time", ylab="Standardised Residuals")  
plot(weight,res1, xlab="Weight", ylab="Residuals") # Residuals vs x  
plot(weight,std.res1, xlab="Weight", ylab="Standardised Residuals")

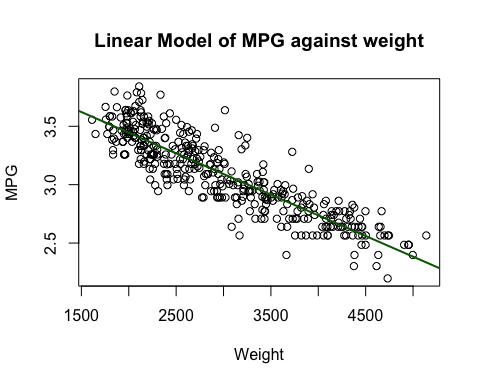


#2.6

logmpgweight.lm <- lm(log(mpg) ~ (weight))  
summary(logmpgweight.lm)

##   
## Call:  
## lm(formula = log(mpg) ~ (weight))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.46681 -0.10062 -0.00702 0.09944 0.55131   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.154e+00 2.955e-02 140.54 <2e-16 \*\*\*  
## weight -3.540e-04 9.527e-06 -37.16 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1595 on 383 degrees of freedom  
## Multiple R-squared: 0.7829, Adjusted R-squared: 0.7823   
## F-statistic: 1381 on 1 and 383 DF, p-value: < 2.2e-16

plot(log(mpg) ~ (weight), xlab='Weight', ylab = 'MPG', main = 'Linear Model of MPG against weight')  
abline(logmpgweight.lm, lwd = 2, col = 'darkgreen')



confint(logmpgweight.lm)

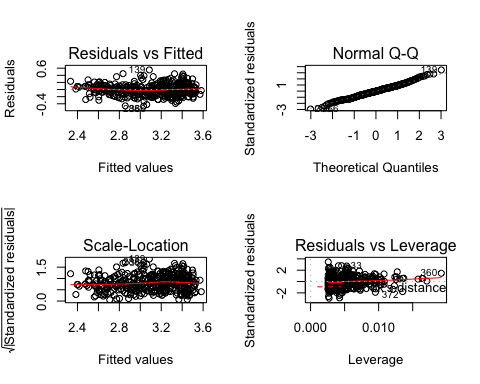
## 2.5 % 97.5 %  
## (Intercept) 4.0955545161 4.2117746062  
## weight -0.0003727569 -0.0003352937

cor(log(mpg), (weight)) #ALL APPENDIX

## [1] -0.8847983

#2.7

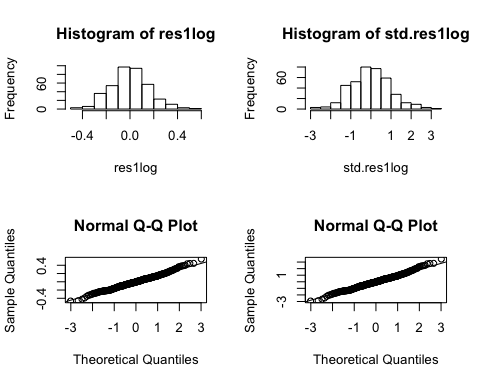
par(mfrow=c(2,2))  
plot(logmpgweight.lm)



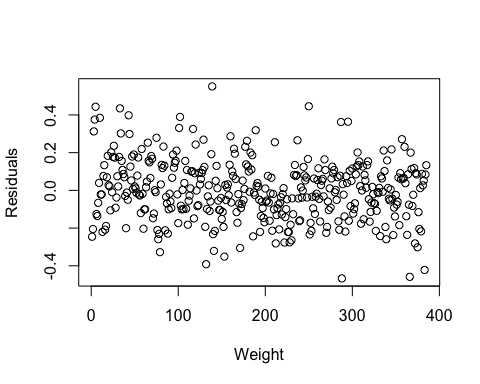
names(logmpgweight.lm) ## checking the output of the model

## [1] "coefficients" "residuals" "effects" "rank"   
## [5] "fitted.values" "assign" "qr" "df.residual"   
## [9] "xlevels" "call" "terms" "model"

res1log=logmpgweight.lm$residuals  
std.res1log=rstandard(logmpgweight.lm) ## standardised residuals  
par(mfrow=c(2,2)) ## plotting 4 plots to checkk normality  
hist(res1log)  
hist(std.res1log)  
qqnorm(res1log)  
qqline(res1log)  
qqnorm(std.res1log)  
qqline(std.res1log)

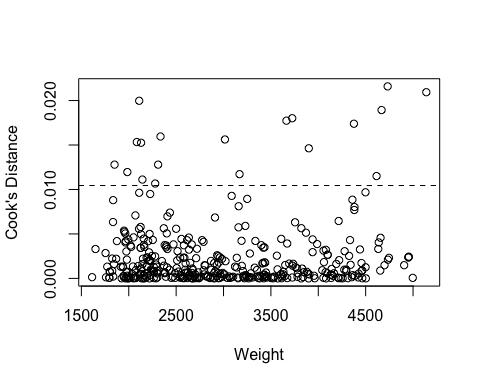


par(mfrow=c(1,1))  
plot(res1log, xlab="Weight", ylab="Residuals")



#2.8

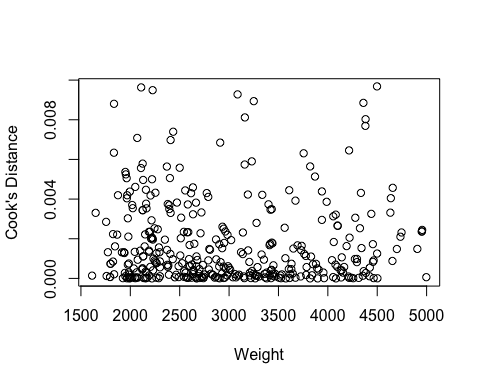
logcooksdweight <- cooks.distance(logmpgweight.lm)  
plot(logcooksdweight ~ weight, xlab = 'Weight', ylab = "Cook's Distance")  
n <- nrow(auto)  
abline(h=4/(n-2), lty = 2)



weightinfluential <- as.numeric(names(sort(logcooksdweight, decreasing = TRUE)[1:19]))  
weightinfluential #StackExchange(2015) REFERENCE

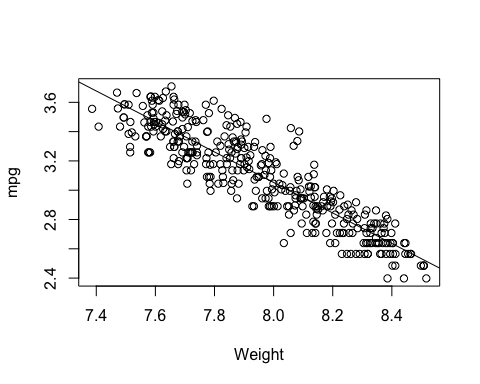
## [1] 372 360 33 357 250 366 375 5 139 4 10 287 44 132 3 288 379 101 153

plot(logcooksdweight ~ weight, xlab = 'Weight', ylab = "Cook's Distance", subset = -weightinfluential)  
abline(h=4/(n-2), lty = 2)



#2.9

newplot <- plot(log(mpg) ~ log(weight), ylab = "mpg",   
 xlab = "Weight", subset = -weightinfluential) # remove the outliers  
lognewmpgweight.lm <- lm(log(mpg) ~ log(weight), subset = -weightinfluential)  
abline(lognewmpgweight.lm)



summary(lognewmpgweight.lm)

##   
## Call:  
## lm(formula = log(mpg) ~ log(weight), subset = -weightinfluential)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.37726 -0.09730 -0.00692 0.09715 0.43882   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 11.38585 0.20962 54.32 <2e-16 \*\*\*  
## log(weight) -1.04168 0.02632 -39.57 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1403 on 364 degrees of freedom  
## Multiple R-squared: 0.8114, Adjusted R-squared: 0.8109   
## F-statistic: 1566 on 1 and 364 DF, p-value: < 2.2e-16

MULTIPLE LINEAR REGRESSION

#3.1

mpg.lm1 <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + model.year)  
summary(mpg.lm1)

##   
## Call:  
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +   
## acceleration + model.year)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.8602 -2.3113 -0.1169 1.9357 14.3790   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.232e+01 4.682e+00 -2.631 0.00885 \*\*   
## cylinders -4.863e-01 3.267e-01 -1.488 0.13752   
## displacement 5.089e-03 7.260e-03 0.701 0.48374   
## horsepower 4.435e-03 1.353e-02 0.328 0.74335   
## weight -6.576e-03 6.659e-04 -9.874 < 2e-16 \*\*\*  
## acceleration 2.260e-02 1.004e-01 0.225 0.82202   
## model.year 7.404e-01 5.155e-02 14.364 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.344 on 378 degrees of freedom  
## Multiple R-squared: 0.8208, Adjusted R-squared: 0.8179   
## F-statistic: 288.5 on 6 and 378 DF, p-value: < 2.2e-16

#3.2

auto1<- (auto[, c(-8,-9)]) #removes character variables  
mlrmatrix<- as.matrix(auto1[, -1])  
mlrmatrix <- cbind(1, mlrmatrix)  
Bhat<- solve(t(mlrmatrix) %\*% mlrmatrix) %\*% t(mlrmatrix)%\*% auto1[, 1]  
round(Bhat, 6)

## [,1]  
## -12.320404  
## cylinders -0.486273  
## displacement 0.005089  
## horsepower 0.004434  
## weight -0.006576  
## acceleration 0.022602  
## model.year 0.740381

print(solve(t(mlrmatrix) %\*% mlrmatrix), digits = 4)

## cylinders displacement horsepower weight  
## 1.961e+00 -2.730e-02 3.320e-04 -2.696e-03 5.432e-05  
## cylinders -2.730e-02 9.548e-03 -1.375e-04 3.928e-05 -2.602e-06  
## displacement 3.320e-04 -1.375e-04 4.714e-06 -2.131e-06 -1.954e-07  
## horsepower -2.696e-03 3.928e-05 -2.131e-06 1.638e-05 -3.489e-07  
## weight 5.432e-05 -2.602e-06 -1.954e-07 -3.489e-07 3.966e-08  
## acceleration -1.871e-02 1.943e-04 8.004e-06 7.483e-05 -2.911e-06  
## model.year -1.925e-02 3.118e-05 2.391e-06 1.396e-05 -5.031e-07  
## acceleration model.year  
## -1.871e-02 -1.925e-02  
## cylinders 1.943e-04 3.118e-05  
## displacement 8.004e-06 2.391e-06  
## horsepower 7.483e-05 1.396e-05  
## weight -2.911e-06 -5.031e-07  
## acceleration 9.016e-04 3.825e-05  
## model.year 3.825e-05 2.376e-04

sigma <- summary(mpg.lm1)$sigma  
round(sigma \* sqrt(diag(solve(t(mlrmatrix) %\*% mlrmatrix))), 6)

## cylinders displacement horsepower weight acceleration   
## 4.682150 0.326743 0.007260 0.013534 0.000666 0.100405   
## model.year   
## 0.051546

#3.3

mpg.lmm=lm(mpg~1,data=auto1) ## model with intercept only  
anova(mpg.lmm, mpg.lm1)

## Analysis of Variance Table  
##   
## Model 1: mpg ~ 1  
## Model 2: mpg ~ cylinders + displacement + horsepower + weight + acceleration +   
## model.year  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 384 23584.2   
## 2 378 4226.7 6 19358 288.53 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

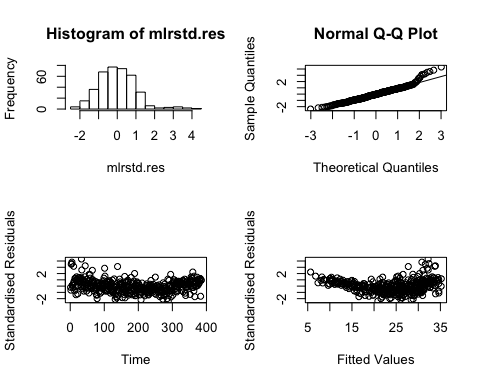
#3.4

mpg.lm2 <- lm(mpg ~ cylinders + displacement + weight + acceleration + model.year)  
anova(mpg.lm1, mpg.lm2)

## Analysis of Variance Table  
##   
## Model 1: mpg ~ cylinders + displacement + horsepower + weight + acceleration +   
## model.year  
## Model 2: mpg ~ cylinders + displacement + weight + acceleration + model.year  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 378 4226.7   
## 2 379 4227.9 -1 -1.2004 0.1074 0.7434

#3.5

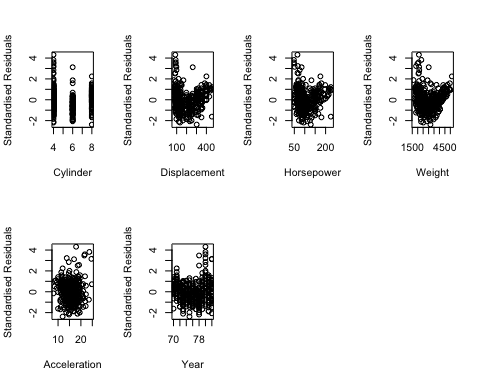
mlrres=mpg.lm1$residuals #Extracting residuals from model  
mlrstd.res=rstandard(mpg.lm1) #residuals are standardised  
par(mfrow=c(2,2)) ## Normality and constance variance check  
hist(mlrstd.res)  
qqnorm(mlrstd.res)  
qqline(mlrstd.res)  
plot(mlrstd.res,xlab="Time", ylab="Standardised Residuals")  
plot(mpg.lm1$fitted.values,mlrstd.res, xlab="Fitted Values", ylab="Standardised Residuals")



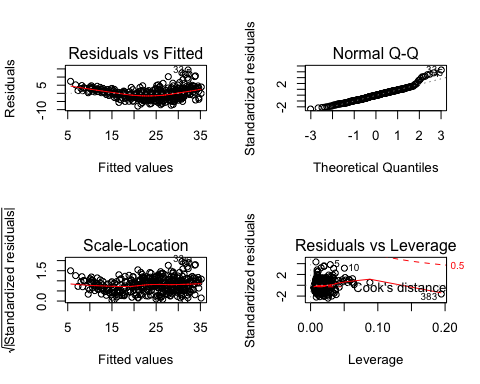
par(mfrow=c(1,1))

#3.6

par(mfrow=c(2,4))  
plot(auto$cylinder,mlrstd.res,xlab="Cylinder", ylab="Standardised Residuals")  
plot(auto$displacement,mlrstd.res,xlab="Displacement", ylab="Standardised Residuals")  
plot(auto$horsepower,mlrstd.res,xlab="Horsepower", ylab="Standardised Residuals")  
plot(auto$weight,mlrstd.res,xlab="Weight", ylab="Standardised Residuals")  
plot(auto$acceleration,mlrstd.res,xlab="Acceleration", ylab="Standardised Residuals")  
plot(auto$model.year,mlrstd.res,xlab="Year", ylab="Standardised Residuals")



par(mfrow=c(2,2))  
plot(mpg.lm1)

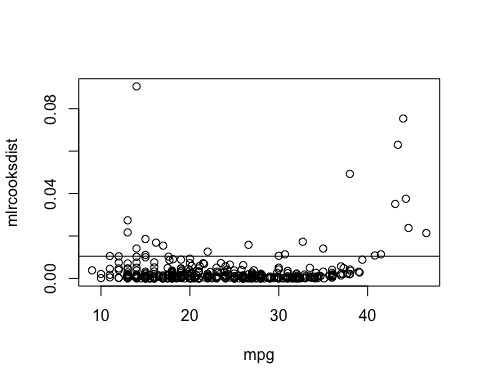


summary(mpg.lm1)

##   
## Call:  
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +   
## acceleration + model.year)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.8602 -2.3113 -0.1169 1.9357 14.3790   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.232e+01 4.682e+00 -2.631 0.00885 \*\*   
## cylinders -4.863e-01 3.267e-01 -1.488 0.13752   
## displacement 5.089e-03 7.260e-03 0.701 0.48374   
## horsepower 4.435e-03 1.353e-02 0.328 0.74335   
## weight -6.576e-03 6.659e-04 -9.874 < 2e-16 \*\*\*  
## acceleration 2.260e-02 1.004e-01 0.225 0.82202   
## model.year 7.404e-01 5.155e-02 14.364 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.344 on 378 degrees of freedom  
## Multiple R-squared: 0.8208, Adjusted R-squared: 0.8179   
## F-statistic: 288.5 on 6 and 378 DF, p-value: < 2.2e-16

#3.7

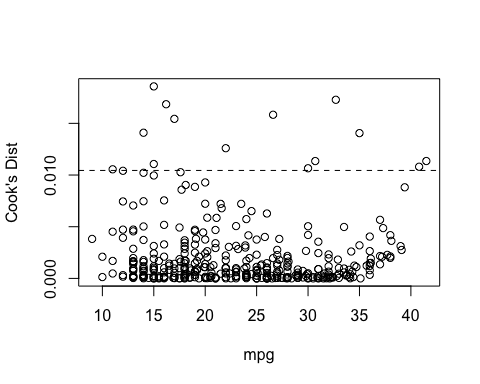
mlrcooksdist <- cooks.distance(mpg.lm1)  
plot(mlrcooksdist ~ mpg)  
abline(h=4/(n-2))



mlrinfluential <- as.numeric(names(sort(mlrcooksdist, decreasing = TRUE)[1:10]))  
mlrinfluential

## [1] 383 10 5 139 4 3 360 44 288 33

plot(mlrcooksdist ~ mpg, xlab='mpg', ylab="Cook's Dist", subset = -mlrinfluential)  
abline(h=4/(n-2), lty = 2)



#3.8

rmoutliermpg.lm1 <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + model.year, subset = -mlrinfluential)  
summary(rmoutliermpg.lm1)

##   
## Call:  
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +   
## acceleration + model.year, subset = -mlrinfluential)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.1647 -2.0283 -0.0458 1.8967 9.6673   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -6.4148935 4.1103152 -1.561 0.1195   
## cylinders -0.4810157 0.2844319 -1.691 0.0917 .   
## displacement 0.0016199 0.0065617 0.247 0.8051   
## horsepower 0.0064646 0.0121650 0.531 0.5955   
## weight -0.0063543 0.0006402 -9.925 <2e-16 \*\*\*  
## acceleration -0.1348512 0.0926962 -1.455 0.1466   
## model.year 0.6890952 0.0448855 15.352 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.87 on 368 degrees of freedom  
## Multiple R-squared: 0.8514, Adjusted R-squared: 0.849   
## F-statistic: 351.5 on 6 and 368 DF, p-value: < 2.2e-16

#3.9

originfit<-lm(mpg~origin,data=auto)  
summary(originfit)

##   
## Call:  
## lm(formula = mpg ~ origin, data = auto)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.979 -5.034 -1.034 3.466 18.966   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 20.0335 0.4049 49.473 < 2e-16 \*\*\*  
## originAsian 10.9452 0.8364 13.086 < 2e-16 \*\*\*  
## originEuropean 7.5804 0.8843 8.572 2.56e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.338 on 382 degrees of freedom  
## Multiple R-squared: 0.3493, Adjusted R-squared: 0.3459   
## F-statistic: 102.5 on 2 and 382 DF, p-value: < 2.2e-16

#3.10

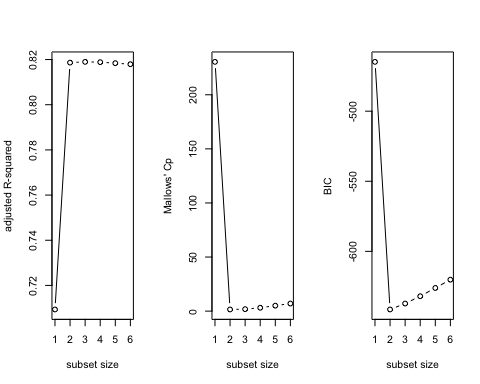
require(leaps)

## Loading required package: leaps

par(mfrow = c(1, 3))  
allsubsets <- regsubsets(mpg ~ ., nvmax=10, nbest=1, data = auto1)  
summary(allsubsets)

## Subset selection object  
## Call: regsubsets.formula(mpg ~ ., nvmax = 10, nbest = 1, data = auto1)  
## 6 Variables (and intercept)  
## Forced in Forced out  
## cylinders FALSE FALSE  
## displacement FALSE FALSE  
## horsepower FALSE FALSE  
## weight FALSE FALSE  
## acceleration FALSE FALSE  
## model.year FALSE FALSE  
## 1 subsets of each size up to 6  
## Selection Algorithm: exhaustive  
## cylinders displacement horsepower weight acceleration model.year  
## 1 ( 1 ) " " " " " " "\*" " " " "   
## 2 ( 1 ) " " " " " " "\*" " " "\*"   
## 3 ( 1 ) "\*" " " " " "\*" " " "\*"   
## 4 ( 1 ) "\*" "\*" " " "\*" " " "\*"   
## 5 ( 1 ) "\*" "\*" "\*" "\*" " " "\*"   
## 6 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

summary.allsubsets <- summary(allsubsets)  
plot(1:6, summary.allsubsets$adjr2, xlab = "subset size", ylab = "adjusted R-squared", type = "b")  
plot(1:6, summary.allsubsets$cp, xlab = "subset size", ylab = "Mallows' Cp", type = "b")  
plot(1:6, summary.allsubsets$bic, xlab = "subset size", ylab = "BIC", type = "b")

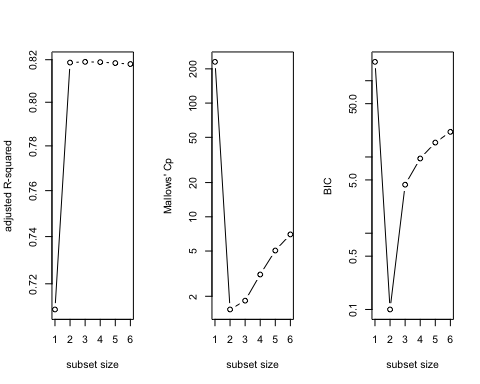


#3.11

require(leaps)  
par(mfrow = c(1, 3))  
allsubsets <- regsubsets(mpg ~ ., nvmax=10, nbest=1, data = auto1)  
summary(allsubsets)

## Subset selection object  
## Call: regsubsets.formula(mpg ~ ., nvmax = 10, nbest = 1, data = auto1)  
## 6 Variables (and intercept)  
## Forced in Forced out  
## cylinders FALSE FALSE  
## displacement FALSE FALSE  
## horsepower FALSE FALSE  
## weight FALSE FALSE  
## acceleration FALSE FALSE  
## model.year FALSE FALSE  
## 1 subsets of each size up to 6  
## Selection Algorithm: exhaustive  
## cylinders displacement horsepower weight acceleration model.year  
## 1 ( 1 ) " " " " " " "\*" " " " "   
## 2 ( 1 ) " " " " " " "\*" " " "\*"   
## 3 ( 1 ) "\*" " " " " "\*" " " "\*"   
## 4 ( 1 ) "\*" "\*" " " "\*" " " "\*"   
## 5 ( 1 ) "\*" "\*" "\*" "\*" " " "\*"   
## 6 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*"

summary.allsubsets <- summary(allsubsets)  
plot(1:6, summary.allsubsets$adjr2, xlab = "subset size", ylab = "adjusted R-squared", type = "b", log='y')  
plot(1:6, summary.allsubsets$cp, xlab = "subset size", ylab = "Mallows' Cp", type = "b", log='y')  
plot(1:6, summary.allsubsets$bic - min(summary.allsubsets$bic) + 0.1,, xlab = "subset size", ylab = "BIC", type = "b", log='y')

 A

#3.12

library(corrplot)

## corrplot 0.84 loaded

multico <- cbind(displacement,horsepower,weight,acceleration)  
multico2 <- cor(multico)  
round(multico2,3)

## displacement horsepower weight acceleration  
## displacement 1.000 0.903 0.935 -0.561  
## horsepower 0.903 1.000 0.869 -0.692  
## weight 0.935 0.869 1.000 -0.431  
## acceleration -0.561 -0.692 -0.431 1.000

library(car)

## Loading required package: carData

vif(mpg.lm1)

## cylinders displacement horsepower weight acceleration model.year   
## 10.619579 19.852587 9.448600 11.113743 2.627535 1.243800

corrplot(cor(auto1,method="spearman"))

