STA 160 - Data Science Practices
Midterm Report
Seeds and Automobile Data sets

Author: Jordan Brennan

I. INTRODUCTION

The goal of this report is to explore two multivariate data sets from the UCI Machine Learning Repository, Seeds and Automobiles. Points of interest include but are not limited to, comparing multiple labels with respect to one single feature, measuring associative relations between a categorical response variable and multiple covariate features, and determining which variables are pertinent for classification. We will begin with analysis of the seeds data, containing all continuous variables, and then discuss the automobiles data set which has both categorical and continuous variables.

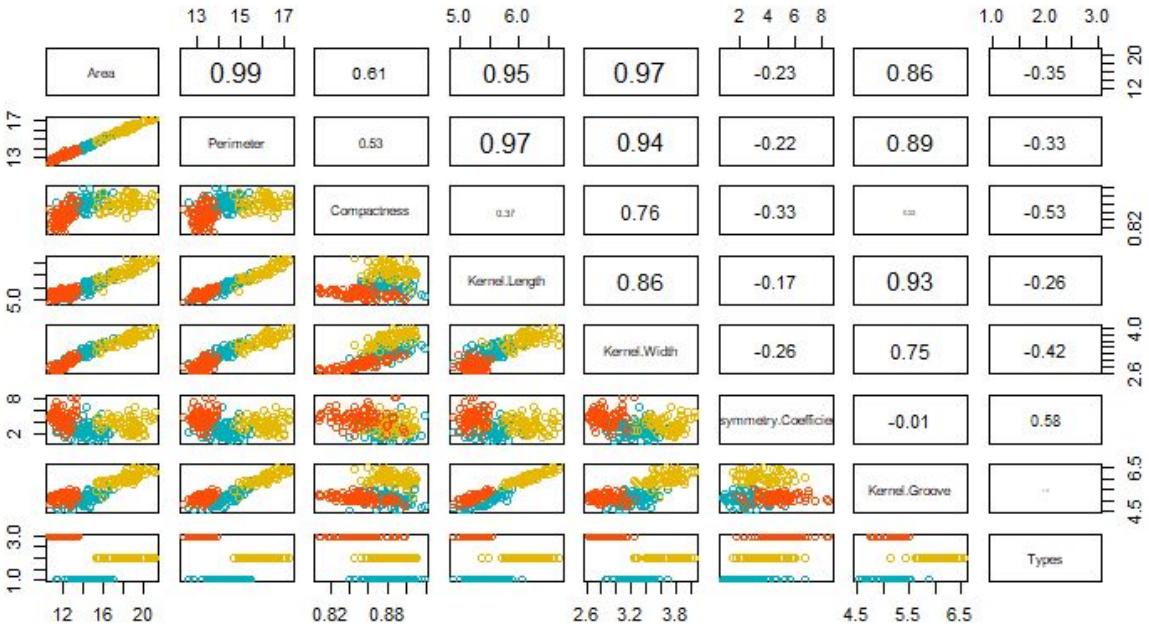
II. SEEDS

We will be first taking a look at the Seeds data, where the examined group comprised kernels belonging to three different varieties of wheat: Kama, Rosa and Canadian, 70 elements each, randomly selected for the experiment. The data set can be used for the tasks of classification and cluster analysis. The seven predictor variables that are used in classifying wheat type are shown in (**Table 1**).

Table 1: Seeds Dataset

Area	Perimeter	Compactness	Kernel.Length	Kernel.Width	Asymmetry.Coefficient	Kernel.Groove	Types
15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
14.88	14.57	0.8811	5.554	3.333	1.018	4.956	1
14.29	14.09	0.9050	5.291	3.337	2.699	4.825	1
13.84	13.94	0.8955	5.324	3.379	2.259	4.805	1
16.14	14.99	0.9034	5.658	3.562	1.355	5.175	1

Figure 1: Scatterplot matrix of Seeds Data



Since all variables are continuous, let us first take a look at the relationship of all of the variables in the scatterplot matrix (**Fig. 1**), with three different colors corresponding to the types of seeds in the lower panel. There seems to be some distinguishable clusters between certain variables, and we can start to get an idea of what characteristics each type of seed has and its relationship to the others. Additionally, (**Fig. 1**) shows the Pearson correlation (1), in the upper panel, which measures the linear dependence between each set of variables.

$$r = \frac{\sum (x - m_x)(y - m_y)}{\sqrt{\sum (x - m_x)^2 \sum (y - m_y)^2}} \quad (1)$$

Where m_x is the mean of x, and m_y is the mean of y, and the significance level, or p-value, can be evaluated in one of the following two ways:

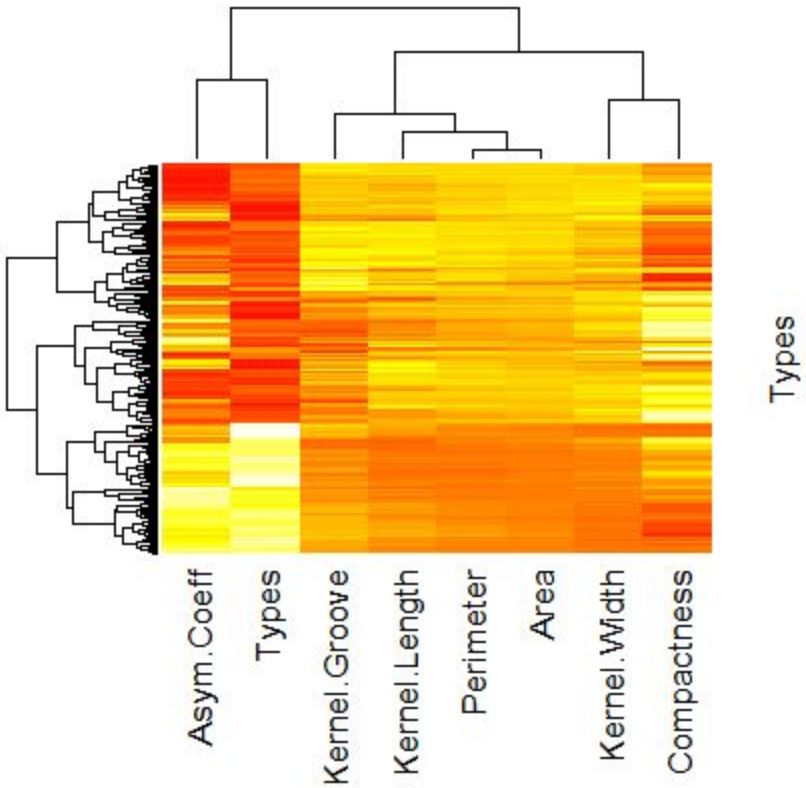
1. Using the correlation coefficient table with df = n - 2.

2. Or by calculating the t-value as follows:

$$t = \frac{r}{\sqrt{1 - r^2}} \sqrt{n - 2}$$

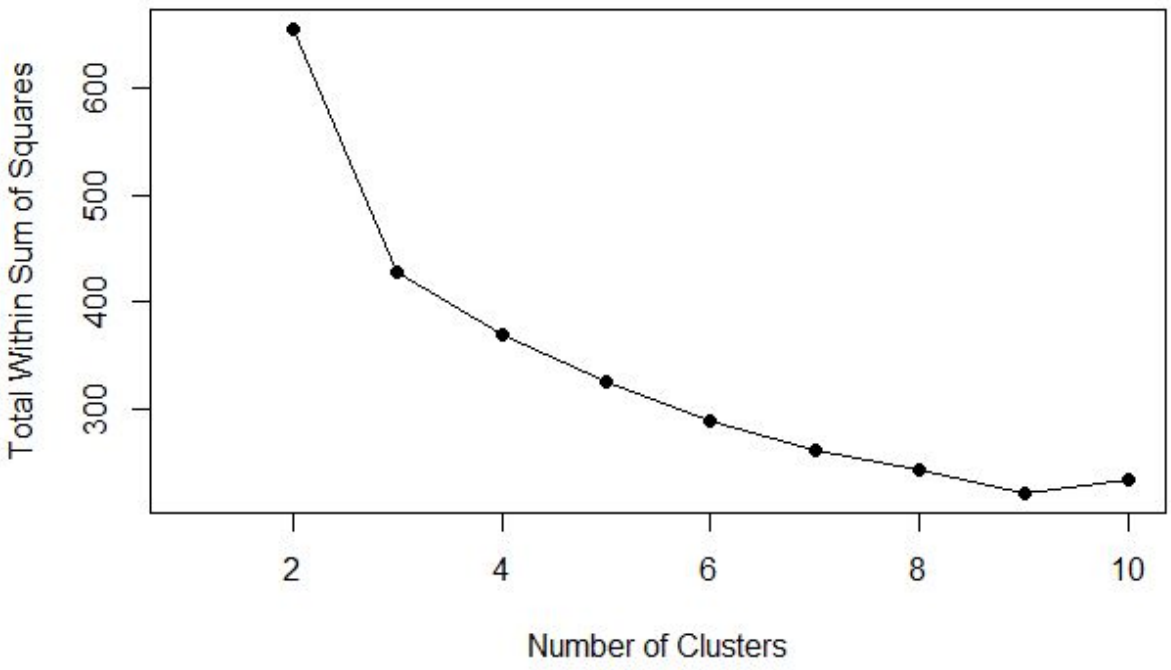
The high R-values between Area, Perimeter, Length, and Groove reveal the dependence of one variable on another. We can see that in a lot of the cases above, that Type 2 wheat generally has higher-valued clusters, except when it comes to Asymmetry Coefficient and Compactness. Consider (**Fig. 2**), and first notice that the claim above holds true when looking at the heat levels of the variables. Then, draw your attention to the dendrogram on the top that shows which chunks of variables are more similar to each other at varying levels. When looking at the dendrogram from the top down, we can see how Asymmetry Coefficient and Types are substantially different from the remaining variables because they are on their own branch. This implies that every chunk within that branch is more similar to each other than to any chunks that join at a higher level. The only higher level branch is the one that compares the rest of the variables with each other. The height of the branches represent the strength of the similarity between the chunks, which reflects some of the R-values shown in (**Fig.1**).

Figure 2: Heatmap of Seeds Data



Because of the distinction between Kernel types across some variables, using K-means clustering for this data set may be suitable. We want to find which, if any of these geometric properties, influences predicting Kernel type. "K- means algorithm is an iterative algorithm that tries to partition the dataset into *K*pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group" [1]. In order to ensure k = 3 is the ideal number of clusters, we take a look at the total within-cluster sum of squares plot, displayed in (**Fig 3.**) and we can see that indeed k =3 is the appropriate amount.

Figure 3: Total Within Sum of Squares of Seeds Data

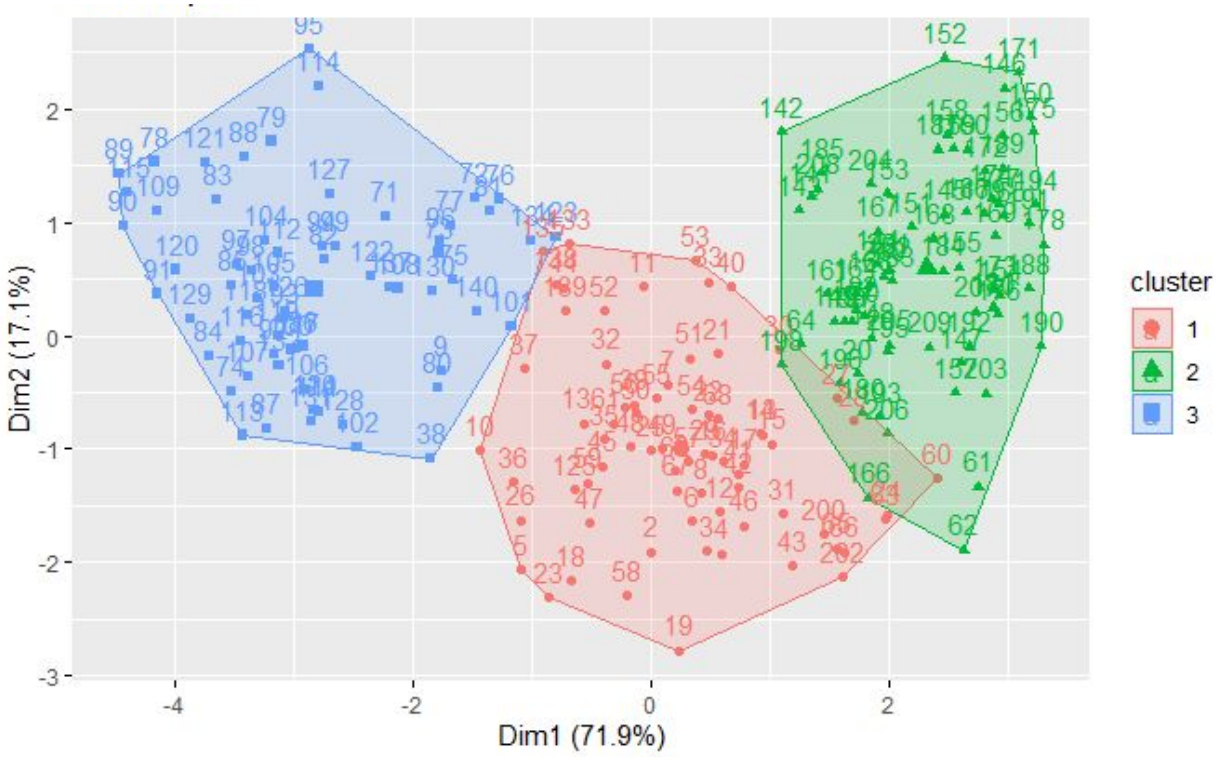


We then proceed with the K-means algorithm, initialized by randomly selecting points from the data set. The idea is to designate random cluster centroids, one for each cluster. Next, the algorithm assigns data points to a cluster, so that the sum of the squared distance between the data points and the cluster's centroid, the arithmetic mean of all the data points that belong to that cluster, is at the minimum.

Table 2: Mean Values of Clusters from K-means

cluster	Area	Perimeter	Compactness	Kernel.Length	Kernel.Width	Asym.Coeff	Kernel.Groove
1	11.85694	13.24778	0.8482528	5.231750	2.849542	4.742389	5.101722
2	18.49537	16.20343	0.8842104	6.175687	3.697537	3.632373	6.041702
3	14.43789	14.33775	0.8815972	5.514577	3.259225	2.707341	5.120803

Figure 3: Cluster Plot of Seeds Data



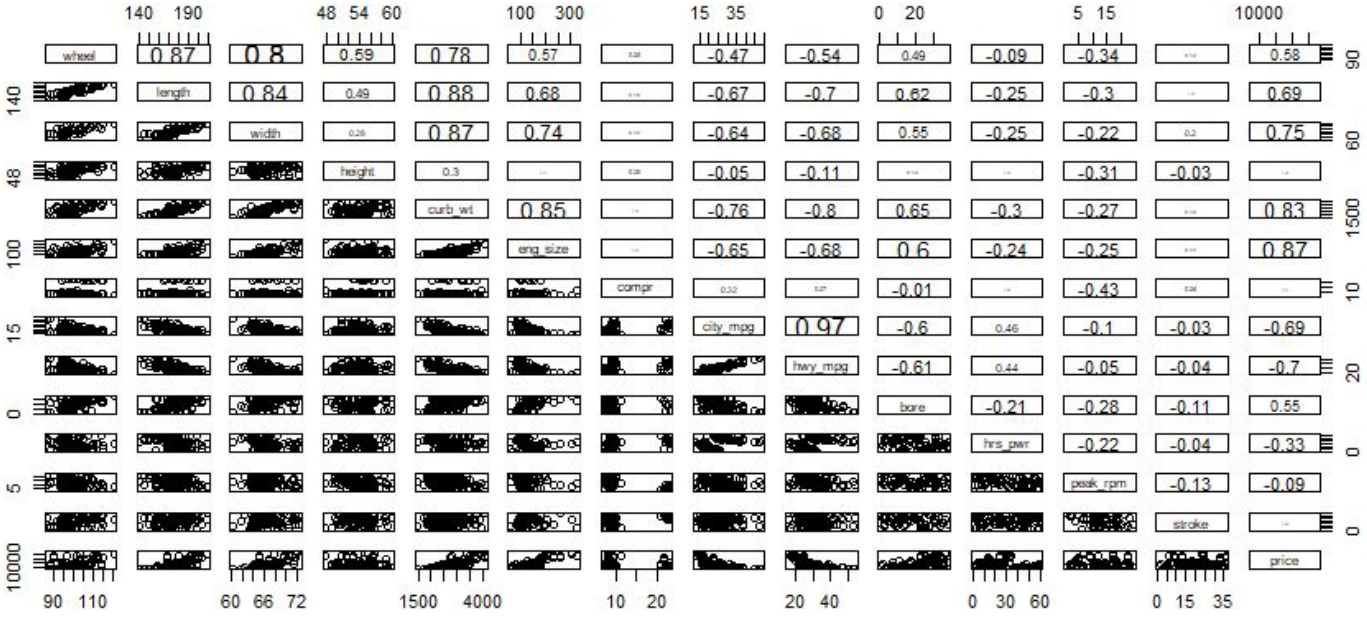
From the K-means algorithm, it formed three clusters of sizes 71, 73, and 66, which is slightly different from what the actual data set had given. (**Table 2**) shows the means of each variable for the cluster's formed by the k-means algorithm, and (**Fig. 3**) shows an illustration of these clusters. (**Table 2**) supports the notion that Type 2 has the highest means of most of the variables across the board, with the exception to Asymmetric Coefficient. In addition, there seems to be some overlap between clusters 1 and 2, and just a slight overlapping between clusters 1 and 3, shown in (**Fig. 3**). From our analysis, k-means worked sufficiently on this data

set, although it didn't reflect the actual kernel types exactly. Other methods of classification on this data set may produce higher accuracy, however, k-means still reveals a lot about the data. Perhaps more criteria, besides geometric measurements, are needed to successfully classify kernel types.

III. AUTOMOBILES

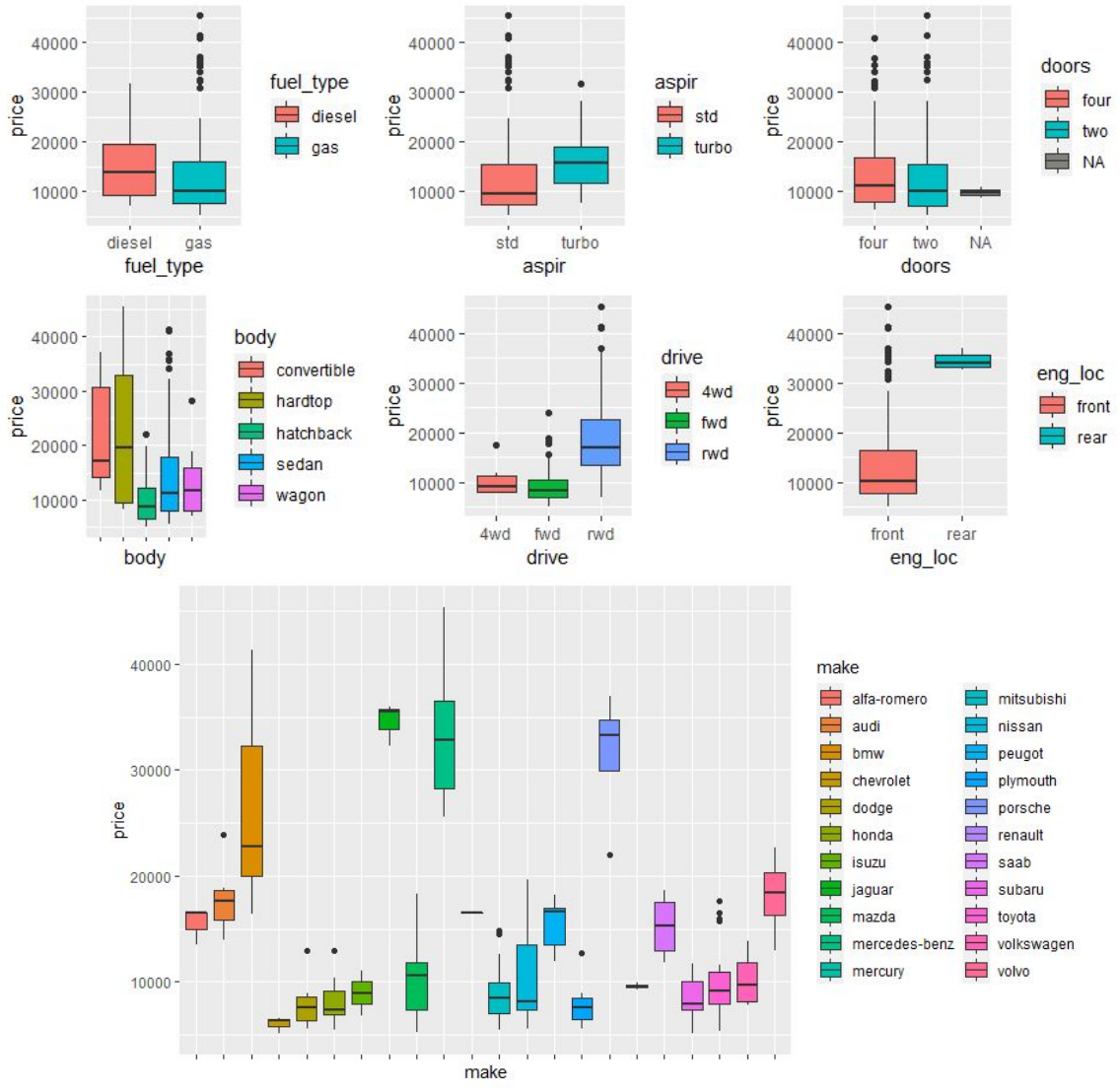
We next will be looking at the Automobile data set, consisting of three types of entities: (a) the specification of an auto in terms of various characteristics, (b) its assigned insurance risk rating, (c) its normalized losses in use as compared to other cars. The objective here is how to deal with categorical features, and how to measure associative relations between multivariate, categorical features with the response, price. This data set consists of 26 variables, and for the purpose of the study, we are going to drop the symboling and normalized-loss values. For the rest of the variables, we will characterize them as either continuous or categorical. Starting with the continuous variables, we will address this like we did with the Seeds data, by creating a scatterplot matrix with Pearson R-values in the upper panel of (**Fig. 4**).

Figure 4: Scatterplot Matrix of Automobile Data



This scatterplot matrix allows us to see the strengths of the relationships between the continuous variables in this data set. We notice strong relationships between some of the cosmetic features of the car, length, width, height, curb weight, etc. In addition, make an observation of the variables with the highest R-values regarding price, engine size ~ 0.87, curb weight ~0.83, and width of the car ~ 0.75. Then, take a look at how city and highway mpg negatively affect the price of the car at R-values of ~ -0.69 and ~ -0.7, respectively. These are early indicators of what continuous variables affect price the most. Next, we will address the categorical variables of the data set, and use boxplots to display each of the categorical variables against price, shown in (**Fig.5**).

Figure 5: Boxplots of Categorical Variables of Automobile Data



The boxplots above can tell us a lot about each categorical variable's relationship with price, and see which feature in that variable is more or less pricey. For fuel type and doors, there is not a huge difference in price between diesel and gas or in the amount of doors the car has. As far as for aspiration, the mean price of the car is generally higher than for a standard. When looking at the body types of cars, convertibles and hardtops have the highest mean prices, while the hatchback's mean price is the least at just under 10k. Then, consider the boxplot for drive train, in which rear-wheel drive clearly has the highest mean price, while two-wheel drive has the least mean price with four-wheel drive just above it. Engine location has a huge discrepancy in price when it comes to being in front or in the back of the car. The mean value of cars with the engine located in the back is just under 35k, while the mean value of cars with the engine located in the front is just above 10k. Lastly, comparing the makes of the cars, jaguars, mercedes-benz, and porsche have the highest mean prices of cars, while chevrolet, honda, and plymouth have the lowest.

The next part in the analysis of this data set is going to be conducting an ANOVA test in order to try and find predictors that influence price. After manipulating the data, and adjusting which variables to include in the ANOVA test, I found a combination of variables that suits the data fairly well. Here, we used a mix of continuous and categorical variables based on what the plots above had suggested. The goal was to use as little predictors as possible to show the strength of the relationship with price. The following seven variables were used in this exploration: engine size, curb weight, make, drive train, engine location, engine type, and fuel system which yielded (**Table 3**).

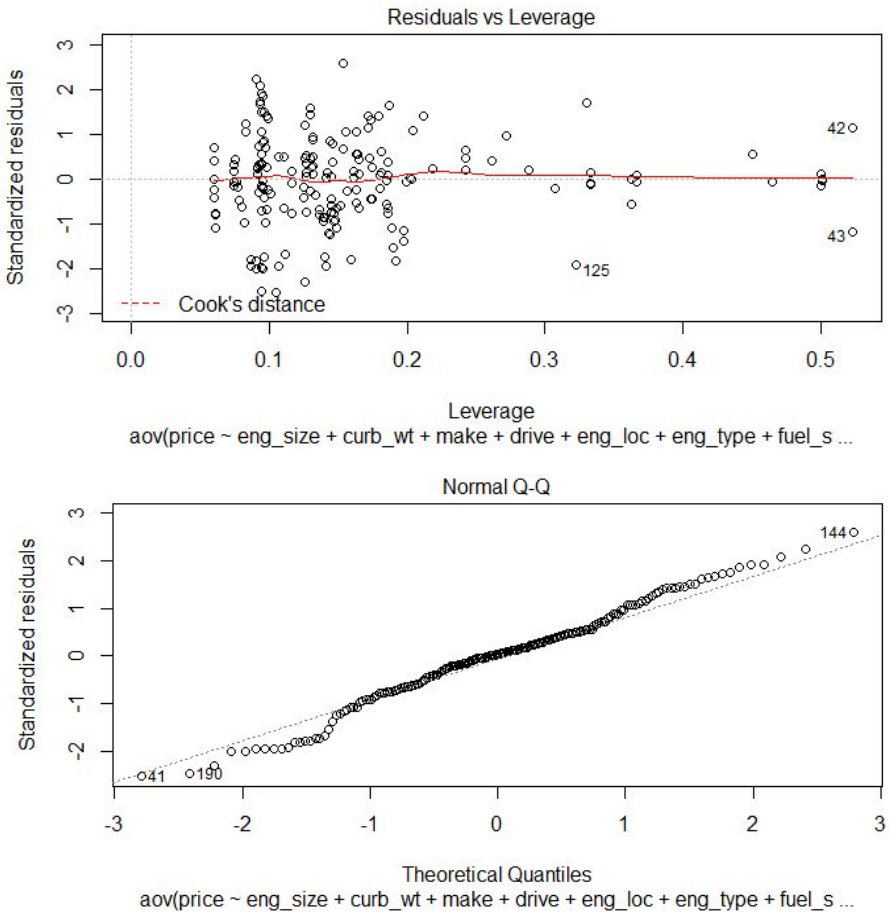
Table 3: Summary of Automobile ANOVA Test

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
eng_size	1	45750	45750	24.150	2.20e-06	***
curb_wt	1	76141	76141	40.193	2.27e-09	***
make	21	84496	4024	2.124	0.00474	**
drive	2	2988	1494	0.789	0.45623	
eng_loc	1	86	86	0.046	0.83109	
eng_type	3	1566	522	0.276	0.84294	
fuel_sys	6	31998	5333	2.815	0.01246	*
Residuals	159	301211	1894			

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0

From the table, we can note which variables had the biggest impact on price, engine size, curb weight, make, and fuel system all had higher F-values than their respective p-values. According to the residual plot and QQ-plot in (**Fig. 6**), the data set seems to be normal and homogeneity of variances is checked.

Figure 6: Automobile Residuals and QQ-Plot



IV. REFERENCES

[1] Dabbura, Imad. “K-Means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks.” *Medium*, Towards Data Science, 29 Apr. 2020, towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a.

V. APPENDIX

```
library(readr)
library(knitr)
library(tidyverse)
library(kableExtra)
library(GGally)
library(FactoMineR)
library(gridExtra)
library(dplyr)
library(factoextra)
library(psych)
library(ggplot2)

seeds <- read.csv("C:/Users/Jordan/Downloads/seeds_dataset.txt", header = FALSE,
sep = "")
names(seeds) <- (c("Area", "Perimeter", "Compactness", "Kernel.Length", "Kernel.Width",
"Asym.Coeff", "Kernel.Groove", "Types"))

kable(seeds[1:5, ]) %>%
  kable_styling("striped")

seeds$Types <- as.factor(seeds$Types)

# Correlation panel
panel.cor <- function(x, y){
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- round(cor(x, y, method = "pearson", use = "complete.obs"), digits=2)
  txt <- paste0(r)
  cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}
```

```
# Customize upper panel
lower.panel<-function(x, y){
  points(x,y, col = my_cols[seeds$Types])
}

# Create the plots
pairs(seeds,
      lower.panel = lower.panel,
      upper.panel = panel.cor)




seeds$cluster <- as.factor(seed_k$cluster)

seeds %>%
  group_by(cluster) %>%
  summarise_all(.,"mean") %>%
  kable %>%
  kable_styling("striped")

fviz_cluster(object = k_result,
             data = seed_z)

#### Automobile Data

library(ggplot2)
library(MASS)
library(tidyverse)
library(ggplot2)
library(MASS)
library(corrplot)
library(caret)
library(gridExtra)
library(cowplot)

autos <- read.csv("C:/Users/Jordan/Downloads/imports-85.data", header = FALSE)
autos <- autos %>% mutate(V2 = recode(V2, "?" = NA_character_))
autos <- autos %>% mutate(V6 = recode(V6, "?" = NA_character_))
autos <- autos %>% mutate(V19 = recode(V19, "?" = NA_character_))
autos <- autos %>% mutate(V20 = recode(V20, "?" = NA_character_))
autos <- autos %>% mutate(V26 = recode(V26, "?" = NA_character_))

autos <- autos[,3:26]
```

```
autos <- autos %>% rename(
  make =V3,
  fuel_type =V4,
  aspir =V5,
  doors =V6,
  body =V7,
  drive =V8,
  eng_loc =V9,
  wheel =V10,
  length =V11,
  width =V12,
  height =V13,
  curb_wt =V14,
  eng_type =V15,
  cyl =V16,
  eng_size =V17,
  fuel_sys =V18,
  bore =V19,
  stroke =V20,
  compr =V21,
  hrs_pwr =V22,
  peak_rpm =V23,
  city_mpg=V24,
  hwy_mpg =V25,
  price =V26)

autos_con <- autos[,sapply(autos,is.numeric)]
autos_cat <- autos[,sapply(autos,is.factor)]
autos_con <- autos_con %>% mutate(bore = autos$bore)
autos_con <- autos_con %>% mutate(hrs_pwr = autos$hrs_pwr)
autos_con <- autos_con %>% mutate(peak_rpm = autos$peak_rpm)
autos_con <- autos_con %>% mutate(stroke = autos$stroke)
autos_con <- autos_con %>% mutate(price = autos$price)

autos_con$bore <- as.numeric(autos_con$bore)
autos_con$hrs_pwr <- as.numeric(autos_con$hrs_pwr)
autos_con$peak_rpm <- as.numeric(autos_con$peak_rpm)
autos_con$stroke <- as.numeric(autos_con$stroke)
autos_con$price <- as.numeric(as.character(autos_con$price))

autos_cat$price <- as.numeric(as.character(autos_cat$price))
autos_cat <- autos_cat[-c(11,12,13,14)]
length(autos_cat)
length(autos_con)
```

```
# Correlation panel
panel.cor <- function(x, y){
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- round(cor(x, y, method = "pearson", use = "complete.obs"), digits=2)
  txt <- paste0(r)
  cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}

lower.panel<-function(x, y){
  points(x,y)
}
# Create the plots
pairs(autos_con,
      lower.panel = lower.panel,
      upper.panel = panel.cor)

p1 <- autos_cat %>% ggplot(aes(x=make, y=price, fill = make)) + geom_boxplot() +
theme(axis.text.x=element_blank())

p2 <- ggplot(data = autos_cat, aes(x=fuel_type, y=price, fill = fuel_type)) +
geom_boxplot()

p3 <- ggplot(data = autos_cat, aes(x=aspir, y=price, fill = aspir)) + geom_boxplot()

p4 <- ggplot(data = autos_cat, aes(x=doors, y=price, fill = doors)) + geom_boxplot()

p5 <- ggplot(data = autos_cat, aes(x=body, y=price, fill = body)) + geom_boxplot() +
theme(axis.text.x=element_blank())

p6 <- ggplot(data = autos_cat, aes(x=drive, y=price, fill = drive)) + geom_boxplot()

p7 <- ggplot(data = autos_cat, aes(x=eng_loc, y=price, fill = eng_loc)) + geom_boxplot()

p8 <- ggplot(data = autos_cat, aes(x=eng_type, y=price, fill = eng_type)) +
geom_boxplot() + theme(axis.text.x=element_blank())

p9 <- ggplot(data = autos_cat, aes(x=cyl, y=price, fill = cyl))+ geom_boxplot() +
theme(axis.text.x=element_blank())

p10 <- ggplot(data = autos_cat, aes(x=fuel_sys, y=price, fill = fuel_sys))+ geom_boxplot()
+ theme(axis.text.x=element_blank())

autos<-autos %>% na.omit
```

```
res.aov <- aov(price ~ eng_size + curb_wt + make + drive + eng_loc + eng_type +
fuel_sys, data = autos)
# Summary of the analysis
summary(res.aov)

plot(res.aov)
```