Fuel vs Tyre Analysis

1. Overview

This notebook uses k-means clustering to group fuel usage to tire spend.

Problem Statement for Project-1-RJP For a medium sized fleet of a grounds maintenance company there is a robust methodology in use for creating a fuel use projection for the coming year. However tyre costs fluctuate and previous budget projections have been inaccurate. Can we use clustering to provide a model of a relationship between fuel costs and tyre costs for the previous year, which could be used to improve the budget projection for future tyre costs? For this exercise solely focus on fuel cost and tyre cost relationship

2. Load libraries

The Tidyverse is a collection of libraries that incudes all of the libraries that are required to complete this analysis

```
# install.packages("tidyverse")

library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.5.2
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0
```

```
## Warning: package 'tidyr' was built under R version 3.5.2
```

```
## Warning: package 'readr' was built under R version 3.5.2
```

```
## Warning: package 'purrr' was built under R version 3.5.2
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
## Warning: package 'forcats' was built under R version 3.5.2
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

3. Load Data

```
ruth <- read.csv("Data.csv")

head(ruth)
```

```
##   Registration X2018.Fuel.Quantity Tyre.Spend
## 1      ET15LBP             4082.60      18.61
## 2      EU67NYR             3321.70      22.39
## 3      EY68UEM              505.95      22.39
## 4      DP17UMG             3379.71      23.00
## 5      EO62WOM             1489.02      24.26
```

```
## 6      EO63XMZ             1351.59      33.00
```

4. Explore and tansform data

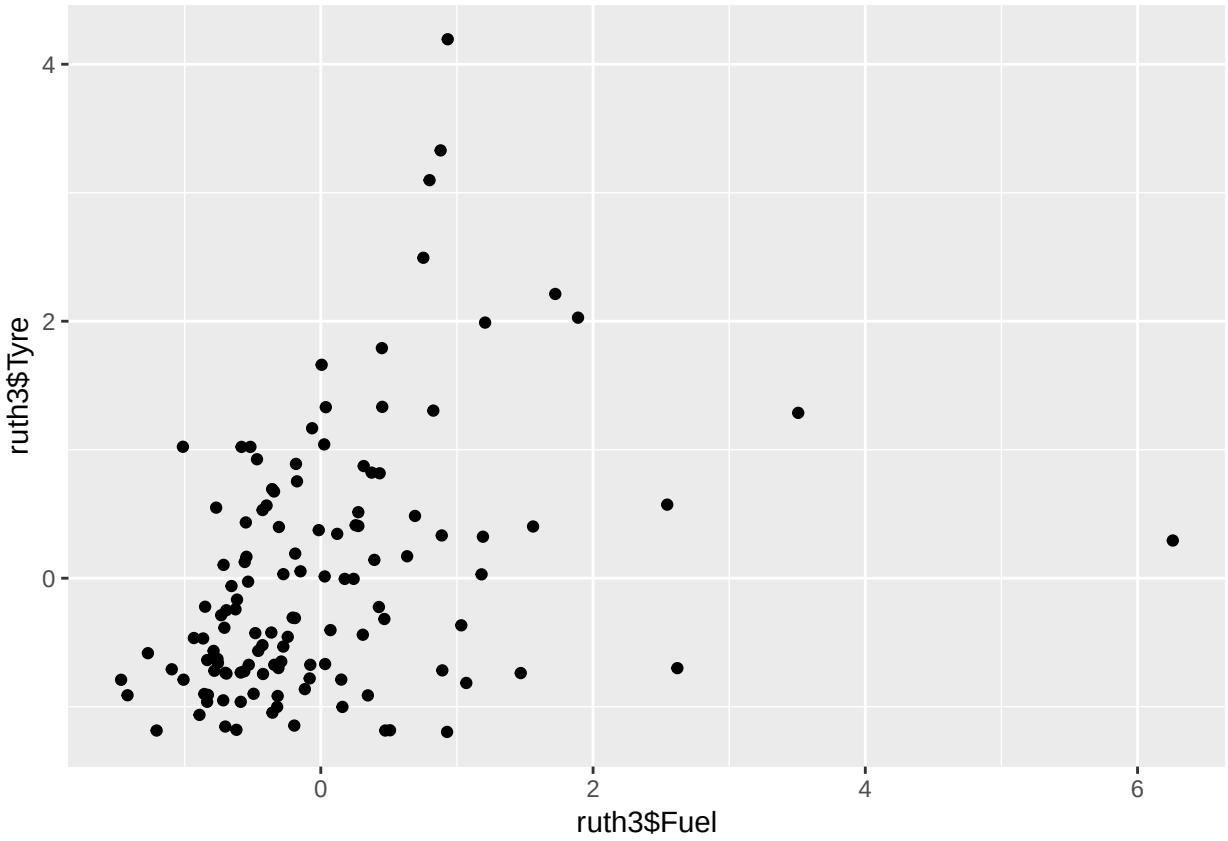
Change titles

```
names(ruth) <- c("Reg","Fuel","Tyre")
#Create DataFrame with only 2 numeric columns

ruth2 <-(ruth[ , c("Fuel","Tyre")])

#scale the data to make a more sensible output, not skewed by one variable

ruth2 <-scale(ruth2)

#make this scaled matrix into a dataframe again

ruth3 <- as.data.frame(ruth2)

head(ruth3)
```

```
##         Fuel      Tyre
## 1  0.9277173 -1.196167
## 2  0.4737596 -1.185175
## 3 -1.2061343 -1.185175
## 4  0.5083687 -1.183401
## 5 -0.6196287 -1.179738
## 6 -0.7016203 -1.154323
```

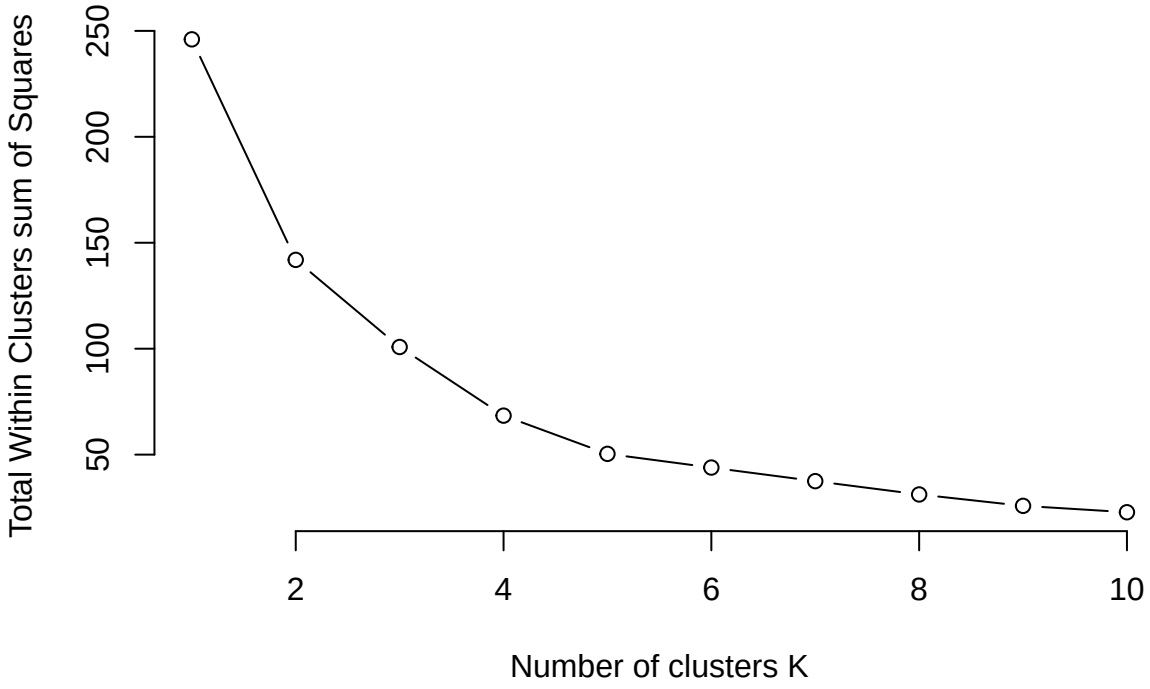
5.Visualise data

```
qplot(x = ruth3$Fuel, y = ruth3$Tyre,)
```



6. fine tuning kmeans

```
kclustermax <- 10
wss <- sapply(1:kclustermax, function(k){kmeans(ruth3[c("Fuel","Tyre")],k)$tot.withinss})

plot(1:kclustermax,wss, type="b", pch = 21, frame = FALSE, xlab="Number of clusters K", ylab = "Total W
```

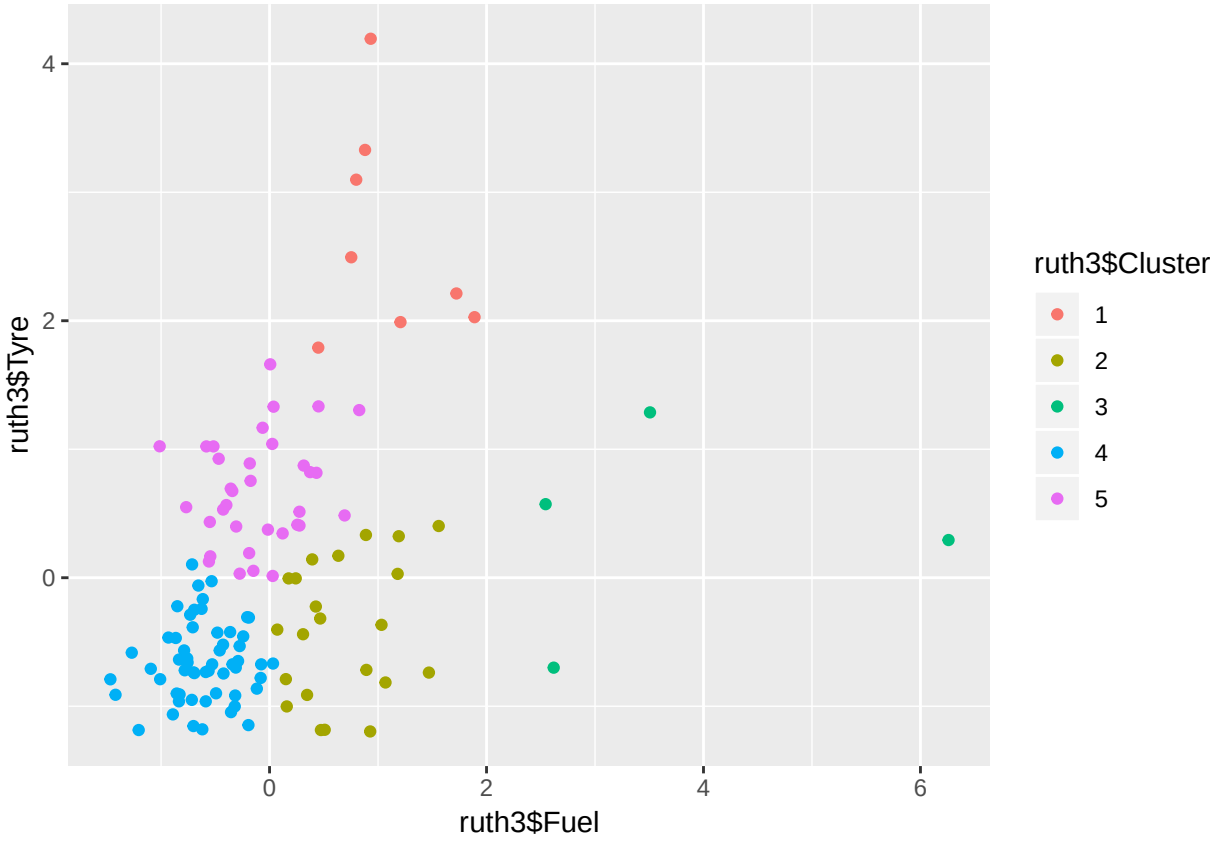


7. Modelling (cllustering)

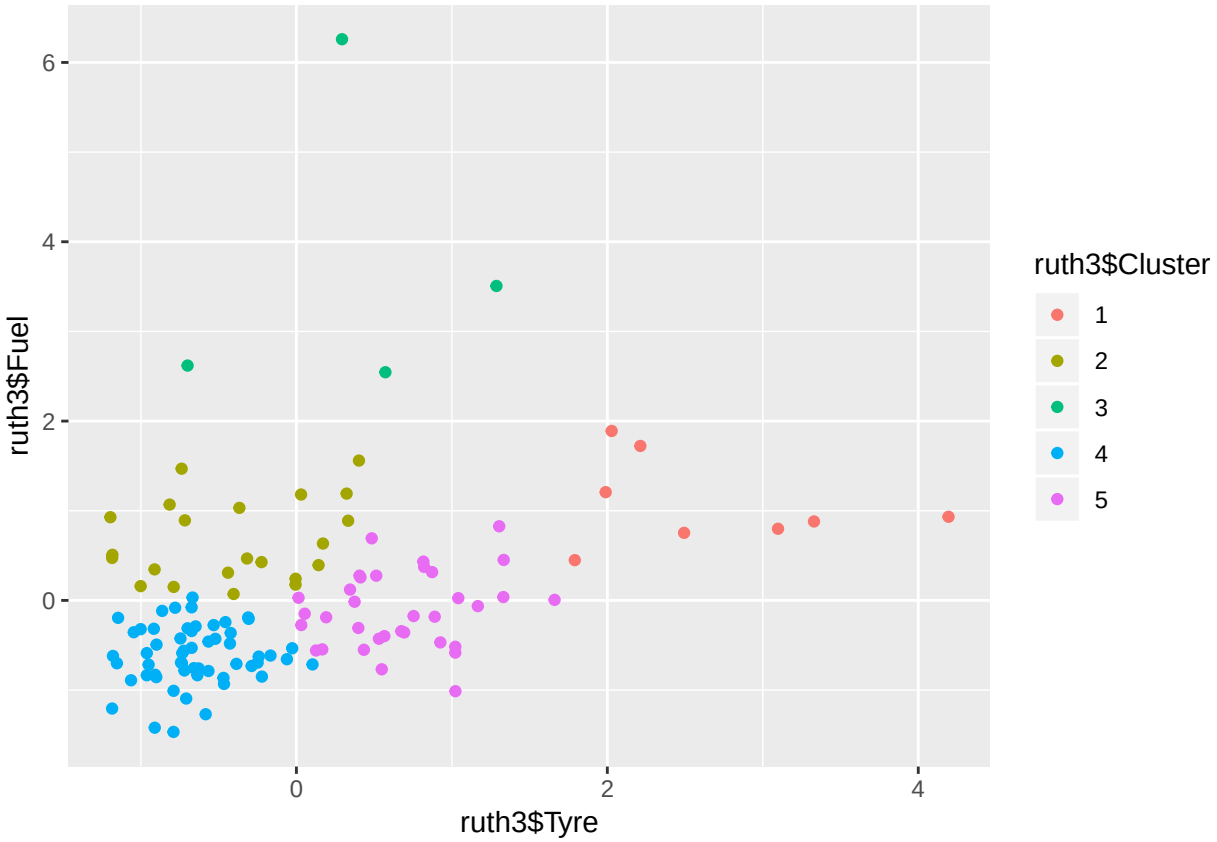
```
# From The above we can assume our dataset has 5 clusers

kclusters <- 5

kmeansresult <- kmeans(ruth3[c("Fuel","Tyre")], centers = kclusters)

#increase number of colu mns in c notation for increased variables

ruth3$Cluster <- kmeansresult$cluster

ruth3$Cluster <- as.factor(ruth3$Cluster)

head(ruth3)
```

```
##         Fuel      Tyre Cluster
## 1  0.9277173 -1.196167       2
## 2  0.4737596 -1.185175       2
## 3 -1.2061343 -1.185175       4
## 4  0.5083687 -1.183401       2
## 5 -0.6196287 -1.179738       4
## 6 -0.7016203 -1.154323       4
```

```
ggplot(ruth3, aes(x = ruth3$Fuel, y = ruth3$Tyre)) + geom_point(aes(colour = ruth3$Cluster))
```




```
qplot(x = ruth3$Tyre, y = ruth3$Fuel, color = ruth3$Cluster)
```



8. Show K Means Values

```
print(kmeansresult)
```

```
## K-means clustering with 5 clusters of sizes 8, 22, 4, 56, 34
##
## Cluster means:
##         Fuel       Tyre
## 1  1.0791292  2.6420416
## 2  0.6620887 -0.4044242
## 3  3.7324809  0.3632819
## 4 -0.6135540 -0.6543964
## 5 -0.1108789  0.6751197
##
## Clustering vector:
##   [1] 2 2 4 2 4 4 4 4 4 2 4 4 4 4 4 2 4 4 4 4 4 2 4 4 2 4 4 4 2 4 4 4 4 2 4
##  [36] 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 2 4 2 2 4 4 4 4 4 2 4 4 4 4
##  [71] 2 2 5 2 5 5 4 5 2 5 2 5 3 2 2 5 5 5 2 5 5 5 5 5 5 5 5 3 5 5 5 5 5 5 5
## [106] 5 5 5 5 5 5 3 5 5 5 5 1 1 1 1 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1]  6.554984 10.098680 11.114586 10.968154 11.631570
##  (between_SS / total_SS =  79.5 %)
##
## Available components:
```

```
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```