Final Course Project (FCP)

Multivariate Analysis on 1985 Auto Imports Database

Luciano Boas | Nathan Lara | Daniel Martinez

Multivariate Analysis
Texas Tech University, Fall 2019

Table of Contents

Group Members

Name	Contact
Luciano Boas	
Nathan Lara	
Daniel Martinez	

1. Final Project Summary

For our project, we decided to study the "1985 Auto Imports Database" that has some interesting variables related to cars. This dataset has information covering the main makers around the world and important car details such as horsepower, and price, for example. We extracted this data from UCI Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Automobile). Our data was pulled from: https://archive.ics.uci.edu/ml/machine-learning-databases/autos/ . Who's parent link is: https://archive.ics.uci.edu/ml/datasets/Automobile

The parent link gives a general description, but the first link is where our data and subsequent description material is located. Once the files are downloaded the file types are converted to csv for viewing and ETL purposes.

This data set consists of three types of entities:

1. First, the specification of an auto in terms of various characteristics its assigned insurance risk rating as compared to other cars.

2. Second rating corresponds to the degree to which the auto is more risky than its price indicates. Cars are initially assigned a risk factor symbol associated with its price. Then adjusted by moving it up (or down) the scale. Actuarians call this process "symboling". A value of +3 indicates that the auto is risky.

3. Third factor is the relative average loss payment per insured vehicle year. This value is normalized for all autos within a particular size classification (two-door small sports/speciality per year.

The dataset has a total of 26 variables (15 continuous, 10 nominal, and 1 integer) and 205 rows. Missing values are denoted by "?". Headers (attributes) are also missing, and we will be addressing in the cleaning section (1.2).

1.1. Introduction

American automotive industry has an immense impact on the domestic economy. Since the invention of the automobile and the mass production techniques, the American economy, and culture, have been transformed by this key element in its prosperity. While some analysts are uncertain of how this industry will perform in the future, we all know that cars will be around us in the long run anyways.

Upon initial investigation of the dataset variables, we saw that there was an assigned insurance risk rating for each car, which was found in the column labeled "symbolling", due to the actuarial method referenced.

Our goal was to determine which vehicles were "riskier" and why – by analyzing the other variables provided. We categorized these into two groups: Dimensions and Performance. The Dimension group

held data regarding various sizing measurements of the many vehicles, and the Performance group was made up of performance-based metrics, like horsepower and stroke.

1.2. Data Cleaning & Visualization

We had to divide our cleaning in two phases: Phase 1 was what we used for the Update 1, which does not include Clustering. In Phase 2, because we had to run clustering analysis, we had to transform "Make" column into rows. The codes for Phase 2 cleaning can be found in the Cluster Analysis (sec. 1.4).

As stated previously, our data did not come very clean or ready to analysis. We had to work in this following sequence in order to clean it and get it ready for analysis (Phase 1):

1) *First, the data was read-in using the read.csv function on the data.csv file.*
2) *Next headers (column names) were manually extracted from the imports-85.csv file and stored as a list, in a txt file named headers.txt*
    a. *This file was read in using the readLines function and stored and added as headers to our data set.*
3) *Next we replaced all of the ?'s in the data with NA's, so that we could run the is.na function to remove all NA's in our data. This resulted in removing only 10 of 205 rows, leaving us with a 195 row data set.*
4) *We then separated our string based data from our numeric data and stored them as independent vectors.*
5) *Finally, we wanted to add rownames to help with analysis, but the column we chose to use was vehicle make. This presented a problem since R would not allow us to use duplicates as row names.*
    a. *Our solution was to create a numerical vector from 1-195, and merge it with our "make" column, and finally to store the merged result as our rownames. While this presented its own challenges, we believed this was the best solution for this data set, considering our initial goal of analyzing all of the other variables against vehicle make.*

Below are our code-steps for cleaning the data:

#read in data

```
data <- read.csv("0LUCIANO/Data Science/08_Multivariate Analysis/0_Project/imports-85.data.csv")
head(data)
```

1) **Create header vector:**

#files pulled from the website contained a description of the data

#these were stored in a .txt file to be used as headers

```
headers <- readLines("0LUCIANO/Data Science/08_Multivariate Analysis/0_Project/Data and Report/headers.txt")
```

2) Add headers:

```
colnames(data) <- headers
```

3) Create vector with numerical only data:

```
idata <- data[,c(10:14, 17, 19:26 )]
```

4) Replace all ? with NA:

```
idx <- idata == "?"
is.na(idata) <- idx
```

5) Data without na:

```
icdata <- na.omit(idata)
```

Now that our data was cleaned and is ready to analyze, we will run the following visualizations in order to get a big picture of how our data looks like.

Here we want to check our data types (after cleaning it):

```
#check data types
typeof(icdata)
```

Now we create a matrix:

```
#create matrix
mdata <- data.matrix(icdata)
```

Check Covariance and Correlation to decide whether the dataset can be used for the project or not:

```
#covariance
cov(mdata)
#correlation
cor(mdata)
```

Table 1 – Correlation Table on All Numerical Variables

	wheel-base	length	width	height	curb-weight	engine-size	bore	stroke	compression-ratio	horsepower	peak-rpm	city-mpg	highway-mpg	price
wheel-base	1	0.882	0.819	0.5852	0.788	0.5744	0.50679	0.1584	0.24704	0.3801	-0.3568	-0.5094	-0.5759	0.5904
length	0.882	1	0.858	0.4975	0.882	0.6879	0.61134	0.1156	0.15962	0.5846	-0.2816	-0.6924	-0.7215	0.6957
width	0.819	0.858	1	0.3108	0.869	0.7419	0.54793	0.1804	0.19011	0.6189	-0.253	-0.652	-0.6965	0.7558
height	0.585	0.497	0.311	1	0.311	0.0322	0.19737	-0.0769	0.26103	-0.0832	-0.2696	-0.1113	-0.1593	0.1403
curb-weight	0.788	0.882	0.869	0.3111	1	0.8576	0.64631	0.1741	0.15538	0.7604	-0.279	-0.7732	-0.8134	0.8357
engine-size	0.574	0.688	0.742	0.0322	0.858	1	0.58337	0.2143	0.0247	0.8427	-0.219	-0.7113	-0.7326	0.8889
bore	0.507	0.611	0.548	0.1974	0.646	0.5834	1	-0.0625	0.00385	0.5684	-0.2773	-0.5912	-0.5994	0.5472
stroke	0.158	0.116	0.18	-0.0769	0.174	0.2143	-0.06251	1	0.19886	0.1029	-0.0709	-0.0344	-0.0421	0.0948
compression-ratio	0.247	0.16	0.19	0.261	0.155	0.0247	0.00385	0.1989	1	-0.2142	-0.4451	0.3308	0.2674	0.0696
horsepower	0.38	0.585	0.619	-0.0832	0.76	0.8427	0.56844	0.1029	-0.21416	1	0.1059	-0.8345	-0.8131	0.8111
peak-rpm	-0.357	-0.282	-0.253	-0.2696	-0.279	-0.219	-0.27732	-0.0709	-0.44506	0.1059	1	-0.0703	-0.0176	-0.1043
city-mpg	-0.509	-0.692	-0.652	-0.1113	-0.773	-0.7113	-0.59124	-0.0344	0.33085	-0.8345	-0.0703	1	0.9723	-0.7034
highway-mpg	-0.576	-0.722	-0.696	-0.1593	-0.813	-0.7326	-0.59944	-0.0421	0.26736	-0.8131	-0.0176	0.9723	1	-0.7161
price	0.59	0.696	0.756	0.1403	0.836	0.8889	0.54717	0.0948	0.06956	0.8111	-0.1043	-0.7034	-0.7161	1

Based on the correlation table, we believe that this data set can provide us with some good variables for analysis. We will now proceed to plot this data into a scatter plot.

#color scatterplots for visual analysis

```
plot(icdata[,1:7], col = icdata[,1])
label = levels(icdata[,1])

plot(icdata[,8:14], col = icdata[,1])
label = levels(icdata[,1])
plot(icdata[,1:14], col = icdata[,1])
label = levels(icdata[,1])
```

#create a grid of plots

```
plot(icdata$length, icdata$cityMpg)
```

Image 1 – Scatterplot for All Numeric Data

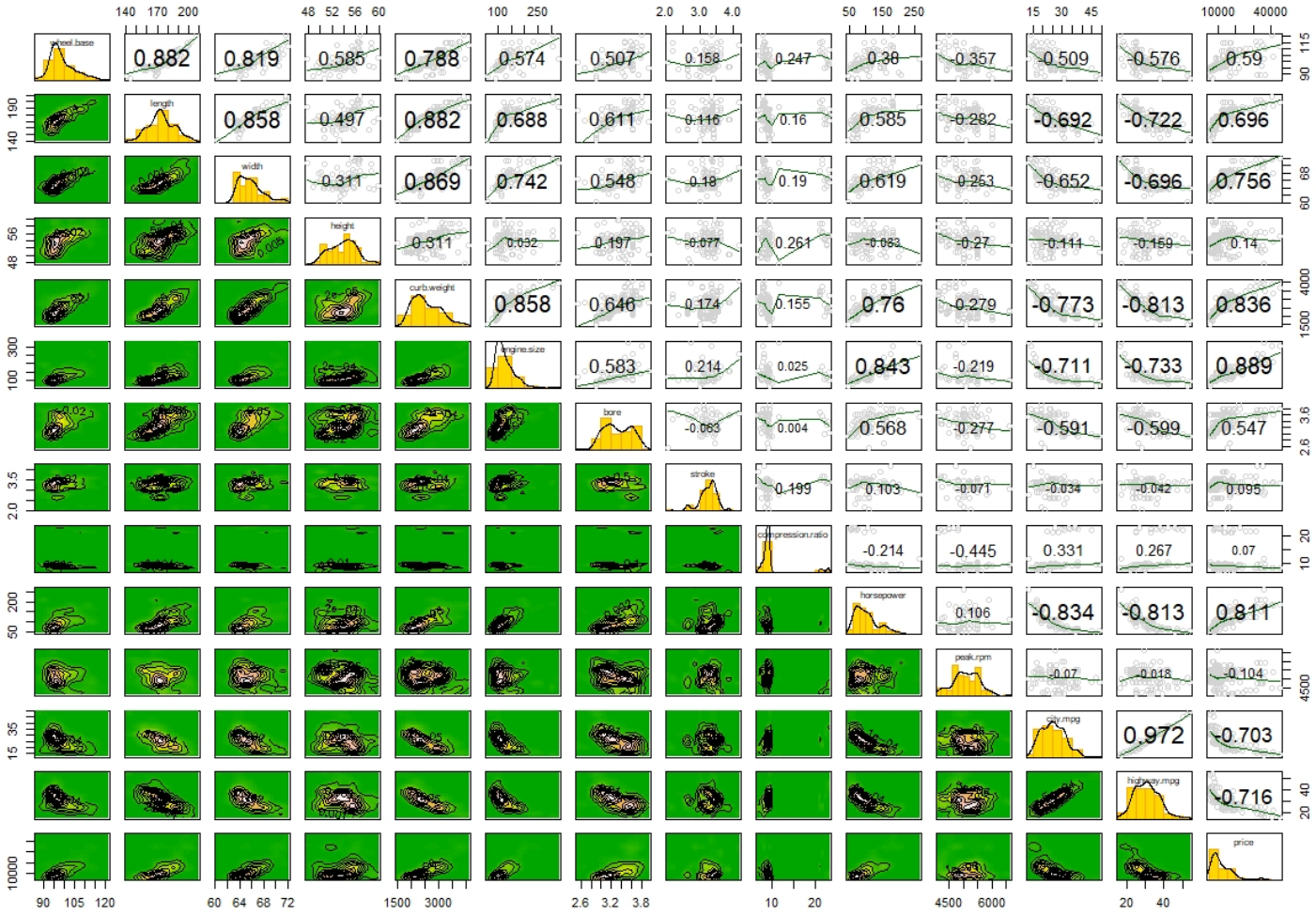
*Please, note that this image is just as sneak peak of the data. While this image has all numeric data available, it is hard to read some detail like names. We will be breaking this matrix down for further analysis.

Next, we thought on using kdpairs matrix to visualize our numeric data and see the interactions among variables. This next plot can give us a better and a better understanding without being so crowded.

#Plot "kdepairs" Matrix:

```
install.packages("ResourceSelection")
library(ResourceSelection)
kdepairs(mdata)
```

Image 2 – Kdepairs of All Numeric Data

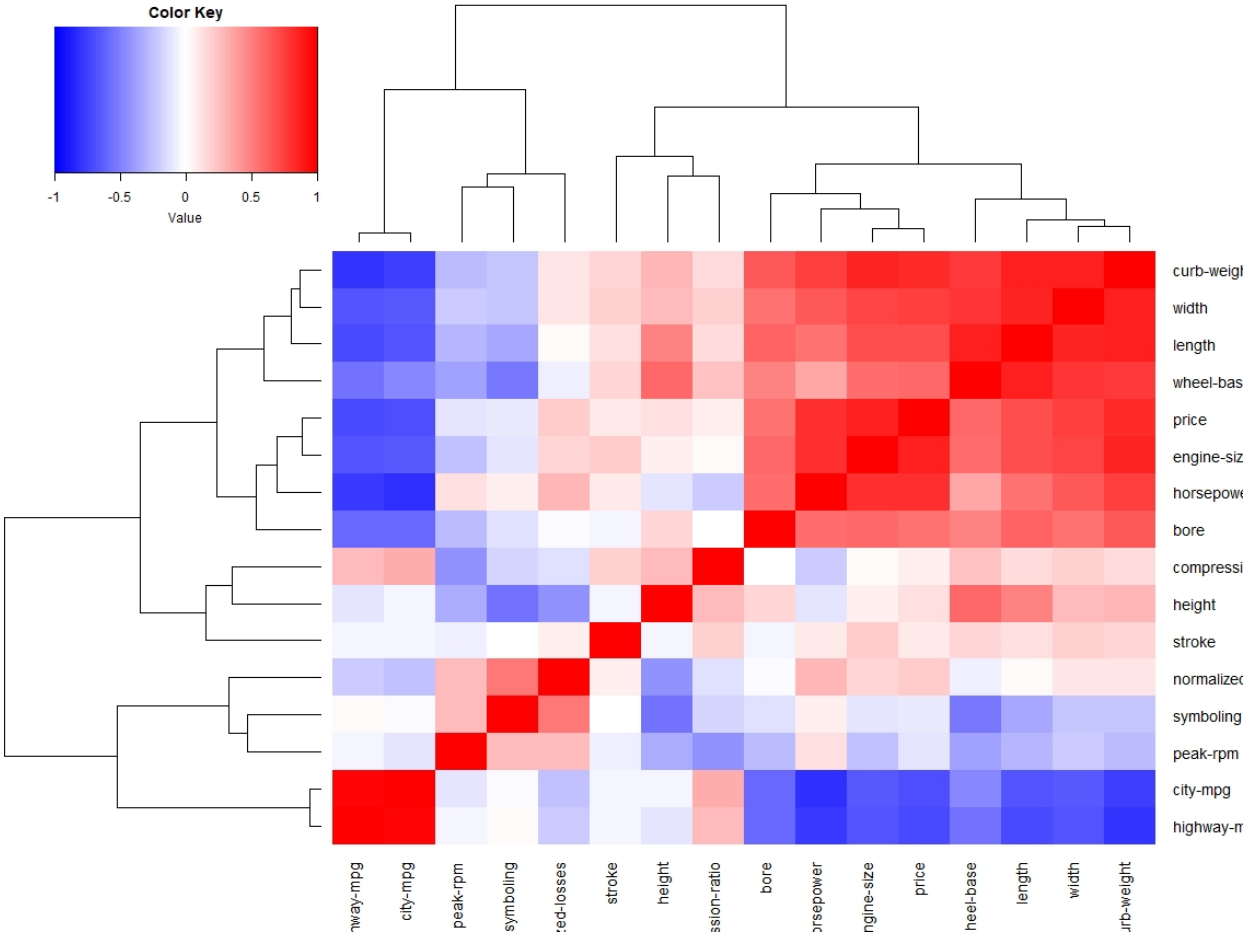


You will see next a heatmap with dendrograms. That will show us, not only the correlation among variables, but also the potential clusters and relationship in this data.

#Heatmap with Dendrograms:

```
install.packages("gplots")
library("gplots")
heatmap.2(corr.p, scale = "none", col = bluered(100),
        trace = "none", density.info = "none")
```

Image 3 – Heatmap with Dendrograms for All Numeric Data



These were all our initial analysis for this data set and what we presented in the Update 1. After, getting green light on it, we moved to analyze the data according to what was requested in the rubric of this project.

At this point, this was our initial ideas to further analyses on this data, and some of our findings:

1) Comparison among following variables: make, fuel type, aspiration, #of doors, drive;
2) Clusters in curbWeight and price;
3) As expected we see correlation between mpg and weight;
4) Negative linear correlation between city/highway mpg and engine size;

5) Positive linear correlation between wheelbase, length, width, height, curbWeight.

We will move now to the second part, which is Update 2, and run more advanced analyses in this data.

1.3. Dimension Reduction Analysis

For the dimension reduction analysis, we first ran PCA to make sure we had two strong components. Our result shows that PC1 and PC2 represent 70%, so we are good to move on.

 # Run PCA Analysis:

```
cars.pca <- princomp(mdata, cor =T)
summary(cars.pca, loading = T)
```

Image 4 – PCA Result for All Numeric Data

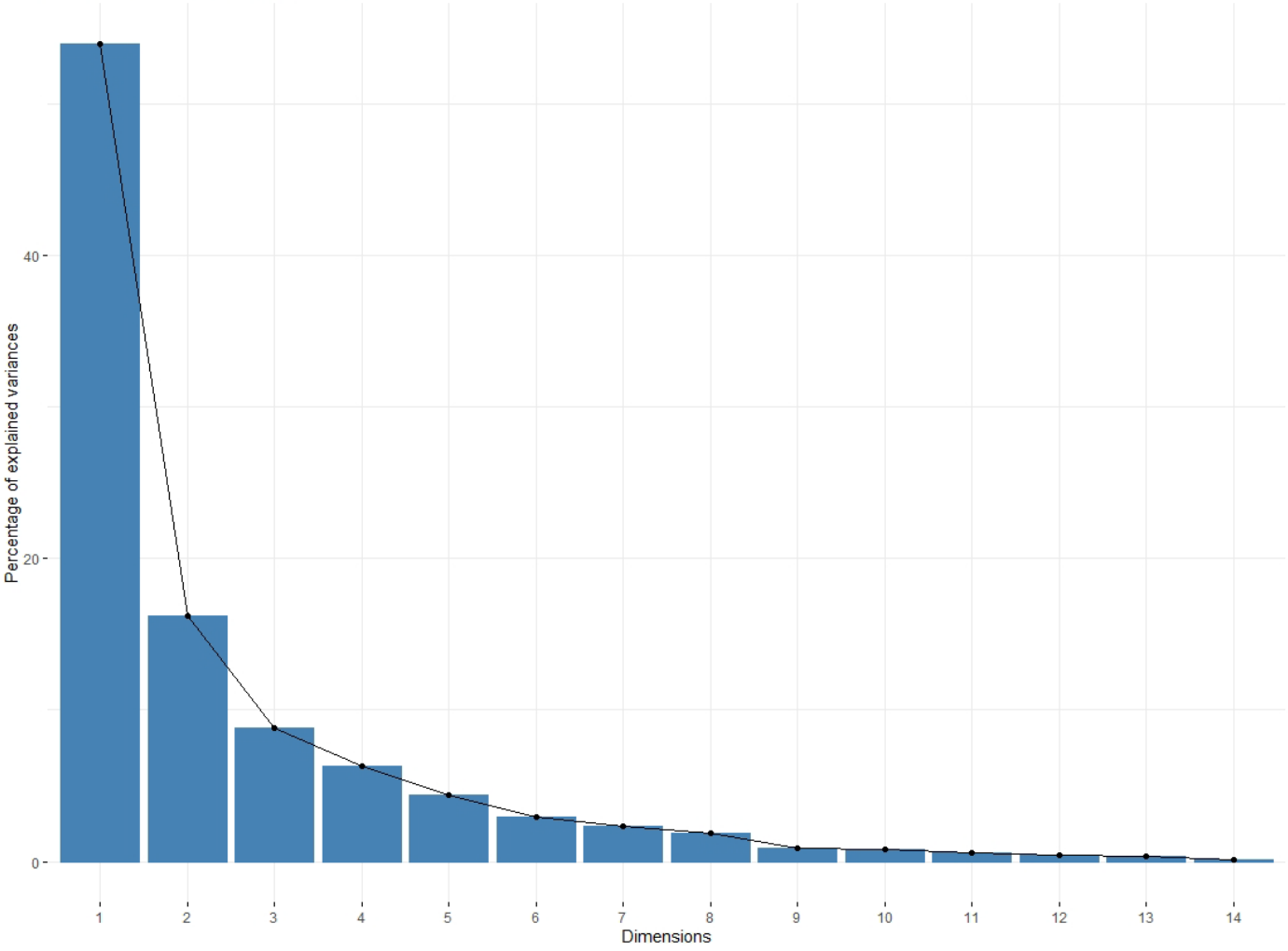
```
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4    Comp.5     Comp.6     Comp.7     Comp.8      Comp.9     Comp.10     Comp.11
Standard deviation     2.7475837 1.5060881 1.1104814 0.94159725 0.7816626 0.64530997 0.56830695 0.51521308 0.347413053 0.332878974 0.284104545
Proportion of Variance 0.5392297 0.1620215 0.0880835 0.06332896 0.0436426 0.02974464 0.02306948 0.01896032 0.008621131 0.007914887 0.005765385
Cumulative Proportion  0.5392297 0.7012512 0.7893347 0.85266368 0.8963063 0.92605092 0.94912040 0.96808073 0.976701858 0.984616745 0.990382130
                          Comp.12     Comp.13     Comp.14
Standard deviation     0.252271049 0.226431931 0.140492285
Proportion of Variance 0.004545763 0.003662244 0.001409863
Cumulative Proportion  0.994927893 0.998590137 1.000000000

Loadings:
               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9 Comp.10 Comp.11 Comp.12 Comp.13 Comp.14
wheel-base      0.290  0.288  0.128  0.237                0.313  0.281  0.331  0.407   0.323   0.415   0.115
length          0.328  0.163  0.124  0.153                0.225         0.432 -0.303  -0.640  -0.203   0.141  -0.165
width           0.324  0.122                -0.131  0.126  0.466  0.165 -0.669 -0.349   0.117
height          0.112  0.398  0.479  0.389                -0.605        -0.181 -0.160   0.109
curb-weight     0.352                                             -0.133  0.181          0.178  -0.194  -0.854   0.109
engine-size     0.321         -0.248 -0.187         0.175 -0.257  0.252  0.211          0.341  -0.563   0.374
bore            0.259          0.160 -0.399  0.321 -0.761         0.217
stroke                 0.101 -0.704  0.486  0.429 -0.193 -0.139
compression-ratio      0.522 -0.287 -0.151 -0.498 -0.319        -0.480                                  0.169
horsepower      0.297 -0.303 -0.137        -0.133        -0.239         0.177 -0.535   0.167   0.591
peak-rpm              -0.453         0.519 -0.489 -0.442  0.129  0.127                          -0.172
city-mpg       -0.309  0.271 -0.115        -0.157                0.458        -0.143   0.154           -0.183  -0.698
highway-mpg    -0.319  0.220 -0.115        -0.142                0.471  0.118 -0.288  -0.157                   0.665
price           0.318         -0.132 -0.111 -0.367  0.108 -0.322  0.287 -0.264  0.432  -0.469   0.191  -0.118
```

Result shows that we got 70% in the first two Components. These two components are more related with the dimensions of the cars. In the sequence, we will plot a bar chart that shows the cumulative perceptual just as a way to better visualize our result.

 # Plot Barchart to vizualize PCA cumulative percent:

```
install.packages("factoextra")
library(factoextra)
fviz_screeplot(cars.pca, main="Barchart of Cumulative Porportion",ncp=50)
```

Image 5 – Barchart of Cumulative PCA for All Numeric Data

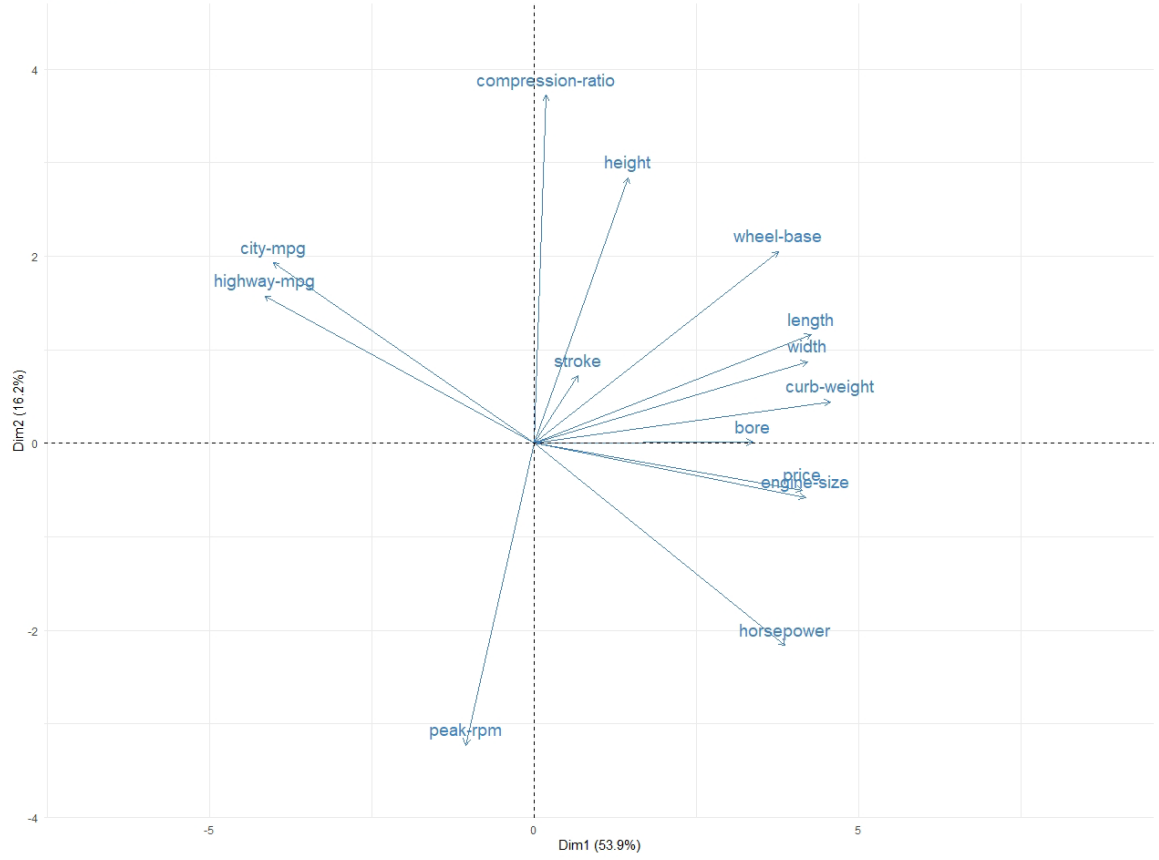


The barchart facilitates to see how the dimensions are proportionally distributed. That is a high-level visualization of the PCA. Next, we plot a biplot to check how variables are related.

 # Plot biplot - Two Principal Components:

```
fviz_pca_biplot(cars.pca, invisible = "ind", habillage ="none", geom = "text", labelsize=5) +
theme_minimal()
```

Image 6 – PCA Biplot for All Numeric Data



From a quick analysis on the above biplot, we see how the variables are correlated. For example, “Price” and “Engine” sized are very close to each other, suggesting strong relationship. On the other hand, we see that “Horsepower” is not closer to “Height”, for instance, and that suggests that they are nor dependable.

To complement this PCA study, we will also run an EFA to see if the data is consistent. We will see that based on the parallel graph (coming next), we picked four factors to study. Factor1 is explain and cover a lot of our variables, main ones being curb-weight, engine size, and price. Factor2 touches wheel-base and height, and its main results. Facto3 is compression-ratio, and bore. And Factor4 is about compression-ration and engine size. This result is consistent with the PCA, which validates our analysis.

 # RUN EFA Analysis

Find the number (range) of potential factors for analysis - PLOT Parallel Scree Plot:

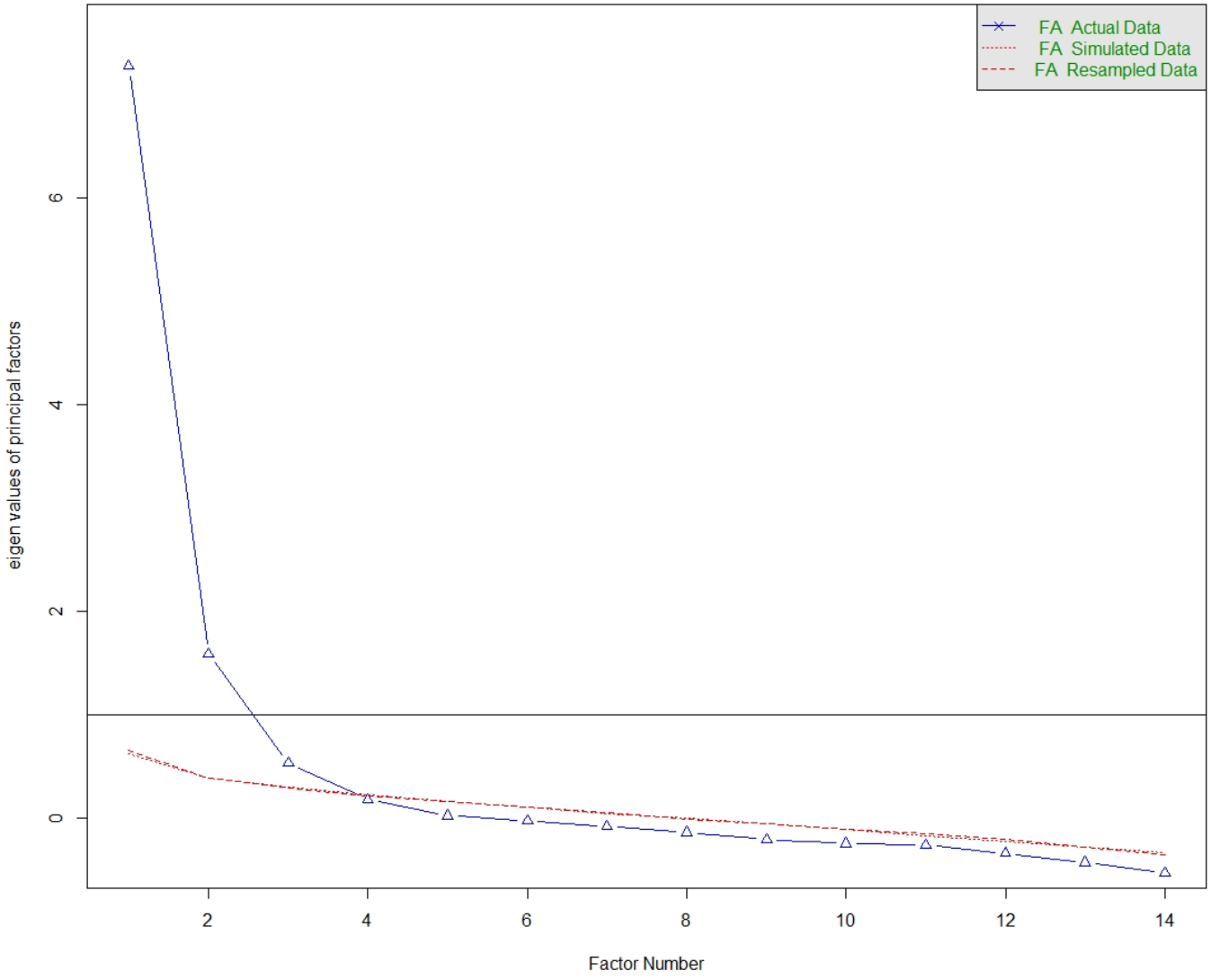
install.packages('psych') -> library(psych)

Plot parallel scree plot

parallel.c <- fa.parallel(mdata, fm = 'minres', fa = 'fa')

parallel.c

Image 7 – Parallel Analysis Scree Plots



 # Run EFA:

car.fa4 <- factanal(mdata, factors=4) # Four should be enough for this set

car.fa4

Image 8 – EFA Results

```
Call:
factanal(x = mdata, factors = 4)

Uniquenesses:
       wheel-base            length             width            height       curb-weight       engine-size              bore            stroke
            0.095             0.086             0.167             0.458             0.057             0.068             0.518             0.929
compression-ratio        horsepower          peak-rpm          city-mpg       highway-mpg             price
            0.486             0.089             0.005             0.005             0.042             0.148

Loadings:
                  Factor1 Factor2 Factor3 Factor4
wheel-base         0.486   0.760   0.146   0.265
length             0.675   0.643           0.192
width              0.690   0.501           0.315
height                     0.725   0.125
curb-weight        0.835   0.392   0.126   0.276
engine-size        0.871           0.117   0.400
bore               0.625   0.213   0.209
stroke                                     0.246
compression-ratio -0.225   0.279   0.322   0.531
horsepower         0.926          -0.153   0.146
peak-rpm                  -0.201  -0.962  -0.166
city-mpg          -0.943  -0.177   0.106   0.253
highway-mpg       -0.928  -0.237           0.188
price              0.827   0.114           0.393

                Factor1 Factor2 Factor3 Factor4
SS loadings       6.366   2.196   1.195   1.090
Proportion Var    0.455   0.157   0.085   0.078
Cumulative Var    0.455   0.612   0.697   0.775

Test of the hypothesis that 4 factors are sufficient.
The chi square statistic is 139.68 on 41 degrees of freedom.
The p-value is 1.08e-12
```

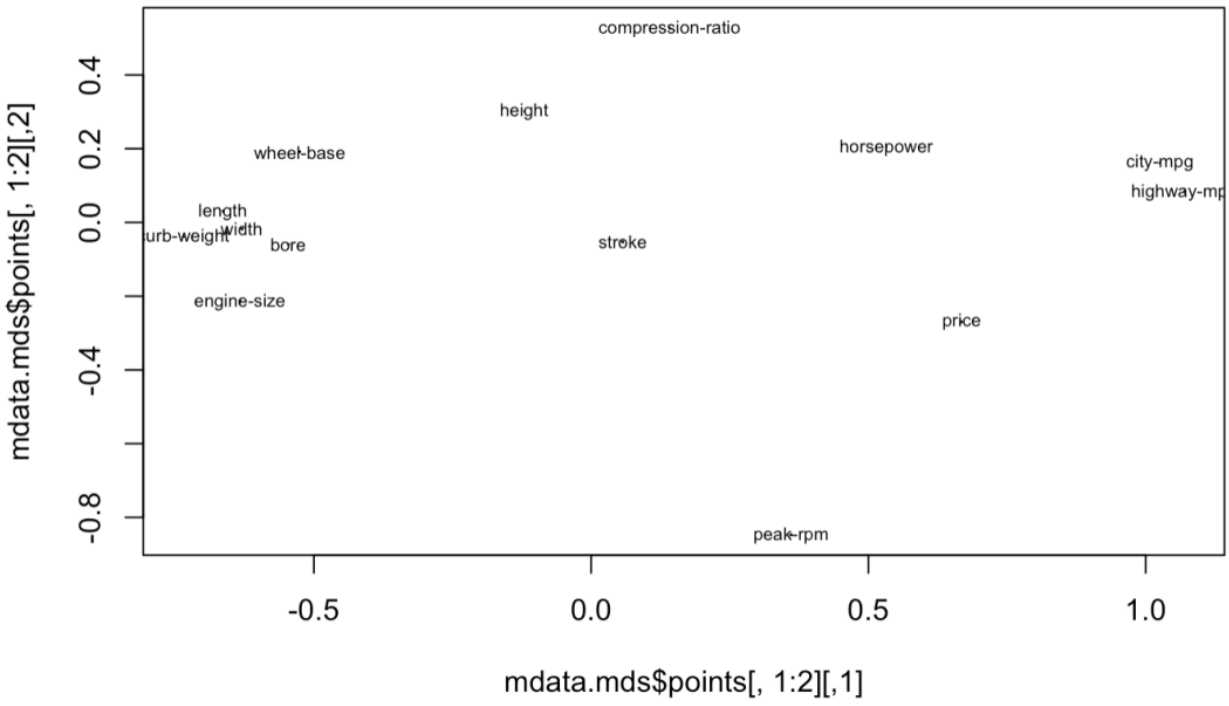
For Dimension Reduction Analysis we also decided to use Multi-Dimensional Scaling. Instead of using the distances for the data, we used the distances for the variables by using the correlations matrix as input. From the MDS plot, we can see that the variables for Length, Width, and Curb-Weight are very close. Engine size, Bore, and Wheel-Base are also closely correlated but not as much. Price was closer to the MPG and Horsepower, than to the size and weight of the vehicles. Height, Stroke, Compression-Ratio, and Peak-RPM fall somewhere in the middle. We can conclude that price was not an important factor for the risk of imported vehicles in 1985. Also, the size of the vehicle was not closely related to the MPG as one would expect them to be. It seems that MPG had more impact on the Price than the Curb-Weight of the vehicles.

 # Create MDS from scaled distance of the variables

```
mdata.mds = cmdscale(dist(scale(1-cor(mdata))), eig = T)

#extract the eigen values
eignv <- mdata.mds$eig

#find proportionate value of eigen values
cumsum(eignv[1:14])/sum(eignv[1:14])
##  [1] 0.6895254 0.8403142 0.9202359 0.9505621 0.9762342 0.9905708 0.9959189
##  [8] 0.9984158 0.9995397 0.9997676 0.9999282 0.9999938 1.0000000 1.0000000
```

#The fist two coordinates provide more than 84% of the overall eigen values so we can use a 2-dimension plot.

Image 9 – Plot MDS



1.4. Cluster Analysis

We started our cluster analysis using the Model-Based technique, because we want to know, right of the bat, the optimal number of clusters for our data. At this point, we just want to have a big picture of that. It's also important to mention, that we brought the variable "make" into this clustering analysis.

Bringing this column (make) back to the dataframe means that we have to clean the data again in order to prevent error (e.g. Coercion error). A possible proactive solution for this problem would be creating a new csv file with all cleaned data and ready to work since the begging, but we did not have the time to do that as we came across this issue later down in the road. Below are our codes to cleaning that and getting data ready to run the analysis:

We anticipate that the MBC gives four clusters as output.

 # Run Model-Based Clustering

```
# Create df with "Make" column
newdata <- data.frame(idata, data$make)


# Replace all ? with NA
idx <- newdata == "?"
is.na(newdata) <- idx
```

# Data without na:
newdata <- na.omit(newdata)
install.packages("mclust") -> library(mclust)
 # Run Mcluster:
mc <- Mclust(newdata)
mc

Image 10 – Model-Based Clustering Output

```
'Mclust' model object: (VVE,4)

Available components:
 [1] "call"           "data"           "modelName"      "n"              "d"              "G"              "BIC"
 [8] "bic"            "loglik"         "df"             "hypvol"         "parameters"     "z"              "classification"
[15] "uncertainty"
```

Here we will check the number of items per group:

 # Check number of items on each group:
table(mc$classification)

Result is:

```
1  2  3  4
37 59 78 20
```

Next, we will plot the fitted mclust for all variables to get a big picture, and a sense of how they interact in this scenario. First, we will run the “classification” and then “uncertainty”. As expected, we can easily “price” and “engine size” is positive, the correlation between “city mpg” and “highway mpg” are also positive.

 # Plot fitted mclust:
plot(mc, what="classification")
plot(mc, what="uncertainty")

Image 11 – Mclust – Classification

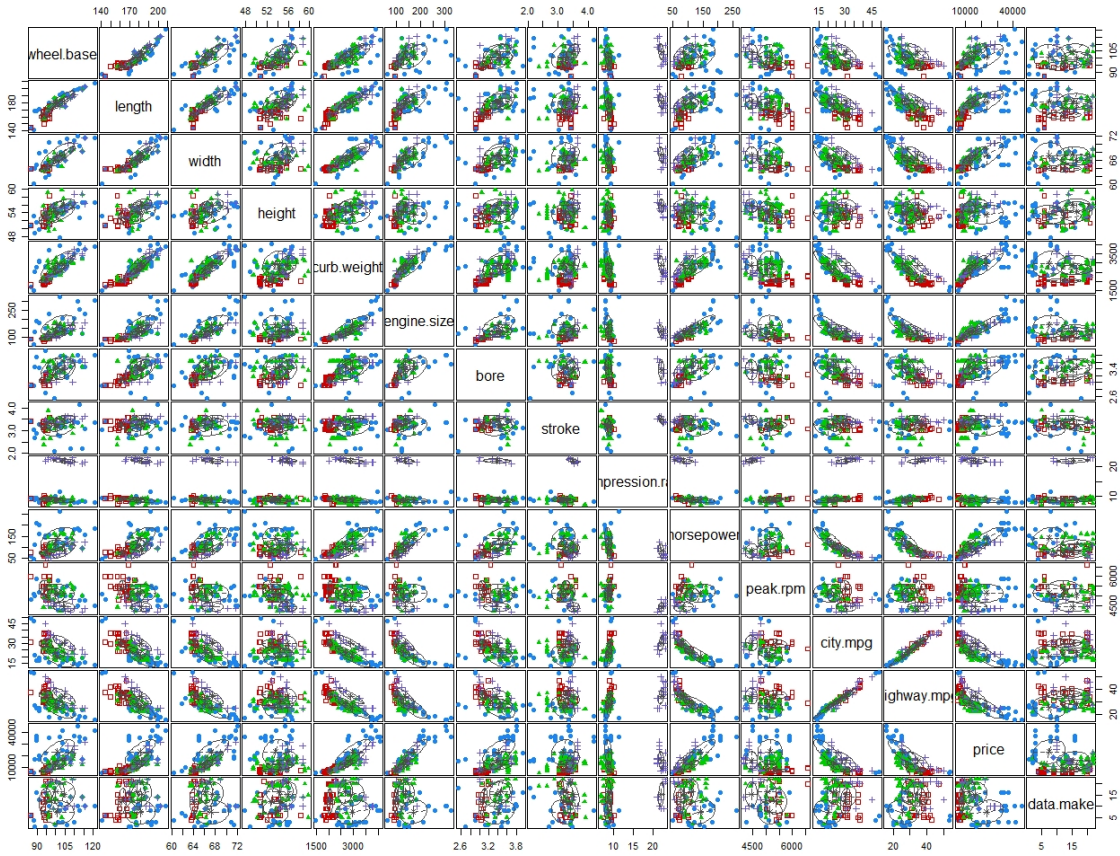
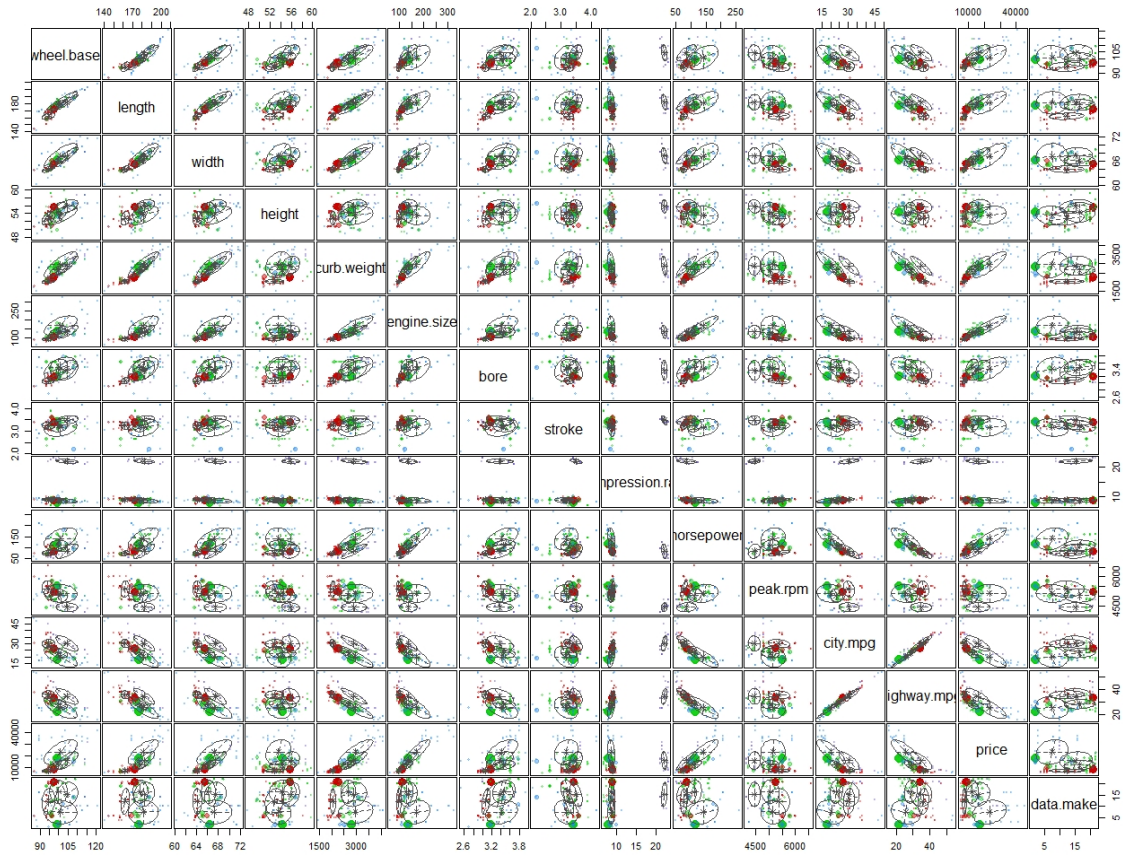


Image 12 – Mclust – Uncertainty



Now that we have a big picture, we want to run some more in deep analysis and test some clustering techniques. As we stated in the Cleaning section, we will now transform the “Make” column into rows. Below are the code for the transformation and Phase 2 of cleaning:

Read in and clean data

```
data <- read.csv('C:/users/nalara/Desktop/TTU/2 Fall/MA/Project/data.csv', header = FALSE)

#create header vector
#files pulled from the website contained a description of the data
#these were stored in a .txt file to be used as headers
headers <- readLines('C:/users/nalara/Desktop/TTU/2 Fall/MA/Project/headers.txt')

#add headers
colnames(data) <- headers

#replace all ? with NA to be cleaned next
idxd <- data == "?"
is.na(data) <- idxd

# create dataframe without cols 1,2,6
# after observing those variables there are too many NA's
# following this we will remove all NA;s
minus.5.data <- data[, c(3:5, 7:26)]

#  remove all na's
cleaned.data <- na.omit(minus.5.data)
```

Separate numeric data for analysis and add “Make” variable as row names

```
#create vector with string data
strdata <- cleaned.data[,c(1:6, 12:15)]
#create vector with numerical only data
numdata <- cleaned.data[,c(7:11, 14, 16:23 )]

# Setting row names
#create new col to merge with make
newcol <- c(1:195)
# merge using paste function
strdata$makeplus <- paste(strdata$make, newcol)
#observe col
strdata$makeplus  ### Because this result is too big, we decided to omit it
```

```
#set index (row names) of data to make
rownames(numdata) <- strdata$makeplus
```

Scale the data for analysis

SCALE DATA, first convert to 1 data type

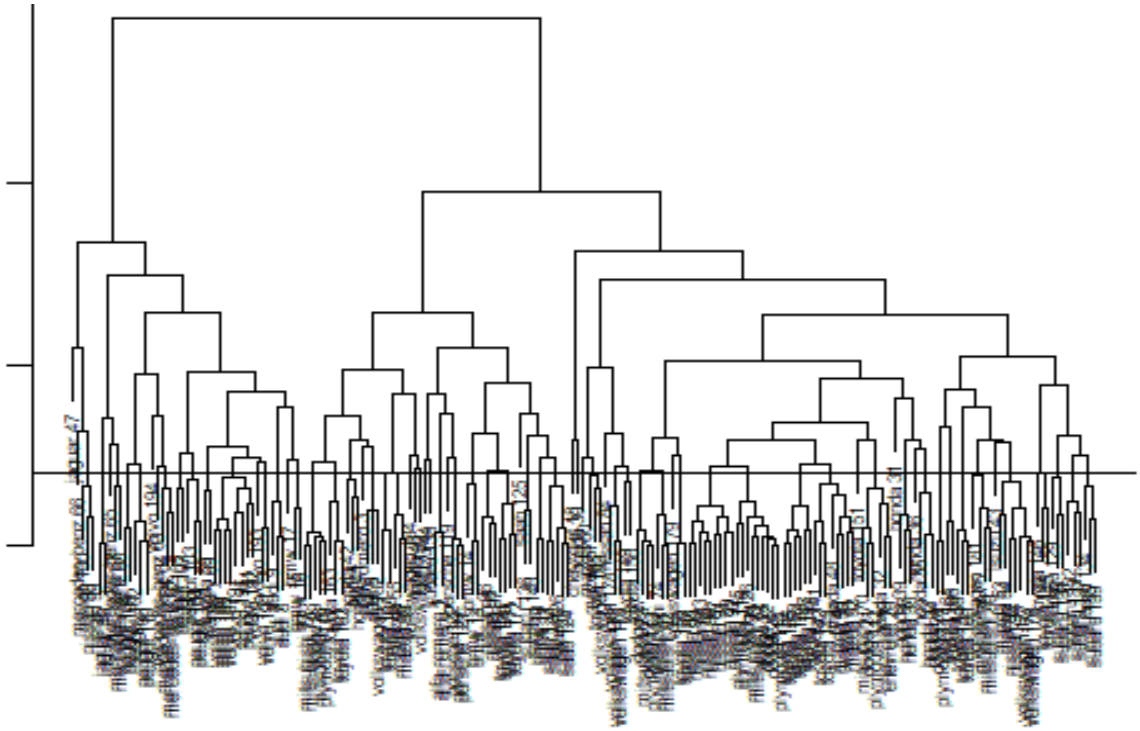
```
#convert all data types to a single type
numdata[, c(1:14)] <- sapply(numdata[, c(1:14)], as.integer)
bestdata <- numdata

#scale data
bestdata.s <- scale(bestdata)
```

Perform Cluster Analysis

Clustering Analysis

```
#create distance matrix
dismat.bestdata.s <- dist(bestdata.s)
hclust.bestdata.s <- hclust(dismat.bestdata.s, "complete")

#plot Dendogram
par("mar")
## [1] 5.1 4.1 4.1 2.1

par(mar=c(1,1,1,1))

plot(hclust.bestdata.s, cex = 0.5, main = "Complete Linkage HC Dendogram")
abline(h=2)
```

Image 13 – Complete Linkage HC Dendogram

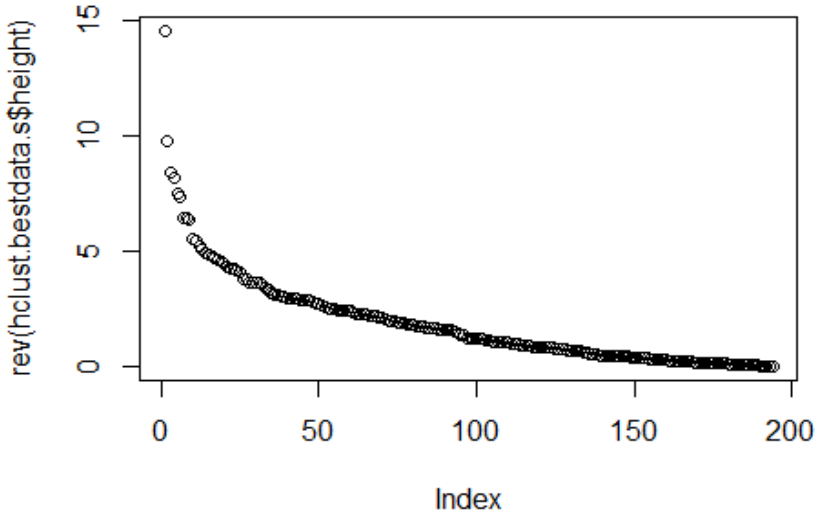


We noticed that it is impossible to actually read the dendogram labels, but we can get an idea of its clustering dynamics.

#observe appropriate number of clusters in hierarchical clustering using scree plot

```
par("mar")
## [1] 1 1 1 1
par(mar=c(5.1, 4.1, 4.1, 2.1))
plot(rev(hclust.bestdata.s$height))
```

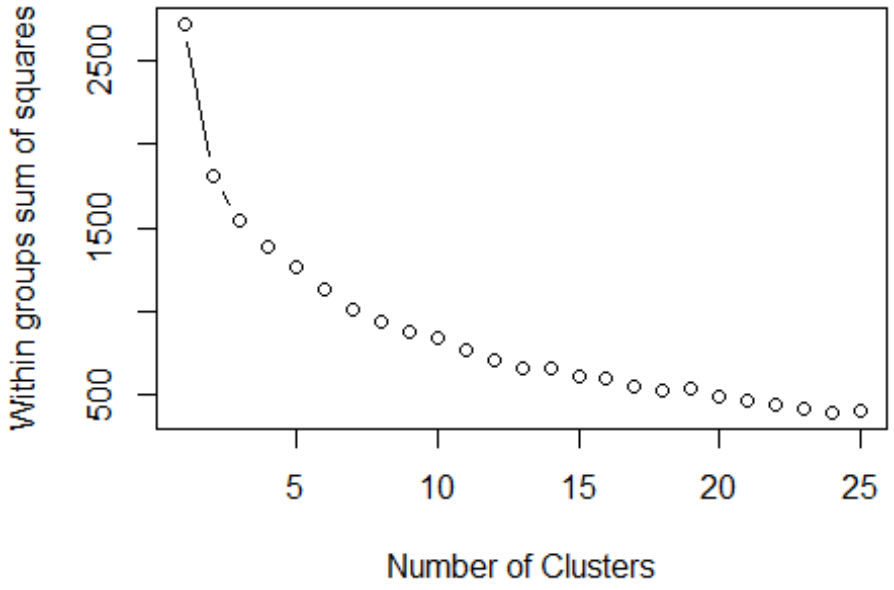
Image 14 – Scree plot



#Plot WGSS

```
plot.wgss = function(best.data, maxc) {
  wss = numeric(maxc)
  for (i in 1:maxc)
    wss[i] = kmeans(best.data,centers=i, nstart = 10)$tot.withinss
  plot(1:maxc, wss, type="b", xlab="Number of Clusters",
       ylab="Within groups sum of squares", main="Scree Plot")
}
#use the scaled data
plot.wgss(bestdata.s, 25)
```

Image 15 – WGSS Scree plot



Based on viewing the above wgss scree plot it appears there may be 3 or 4 clusters, which matches the mclust analysis we made previously. To be consistent, we will be now using 4 clusters for perfoming K-means analysis and plotting visuals.

Perform K-means Clustering Analysis and plotting for visual analysis

```
#perform k-means clustering
#use an increased value for nstart for a more reliable output

kms4 <- kmeans(bestdata.s, centers=4, nstart=10)
```

```
table(kms4$cluster)
##
##  1  2  3  4
## 50 36 95 14
```

#output carmake/group
kms4**$**cluster ### Because this result is too big, we decided to omit it.

#observe centroids

kms4$centers

Image 16 – Centroids

```
##    wheelBase     length       width     height curbWeight engineSize
## 1 -0.2185344  0.1582740  0.09312534 -0.3336900  0.2364328  0.2886895
## 2  1.2964934  1.3175359  1.28823387  0.6292349  1.3081155  1.1257655
## 3 -0.5891074 -0.7400687 -0.72049605 -0.2083268 -0.7954582 -0.6570142
## 4  1.4441544  1.0686807  1.24388847  0.9873637  1.1896236  0.5324512
##         bore       stroke compressionRatio horsepower    peakRpm
## 1  0.4412048  0.008846236      -0.34987384 -0.6975783  0.0735199
## 2  0.8181733 -0.183851313      -0.38581634 -0.4332039  0.1014125
## 3 -0.6265194 -0.093399509      -0.08830187  0.4951315  0.1703413
## 4  0.5717758  1.074949198       2.84083985  0.2454828 -1.6792337
##      cityMpg  highwayMpg      price
## 1 -0.6052379 -0.50093246 -0.8906281
## 2 -1.1172801 -1.16441204 -0.5899428
## 3  0.7045122  0.71071712  0.8043169
## 4  0.2539516 -0.03947644 -0.7600544
```

In our analysis of the clusters above, we see the following details:

Cluster1:
bore/hp/mpg/price

Cluster2:
length/width/curbWeight/bore

Cluster3:
wheelBase/length/width/height/curbWeight/engineSize/bore/stroke/compressionRatio

Cluster4:
peakRpm/cityMpg/highwayMpg/price

PCA Analysis and Plotting

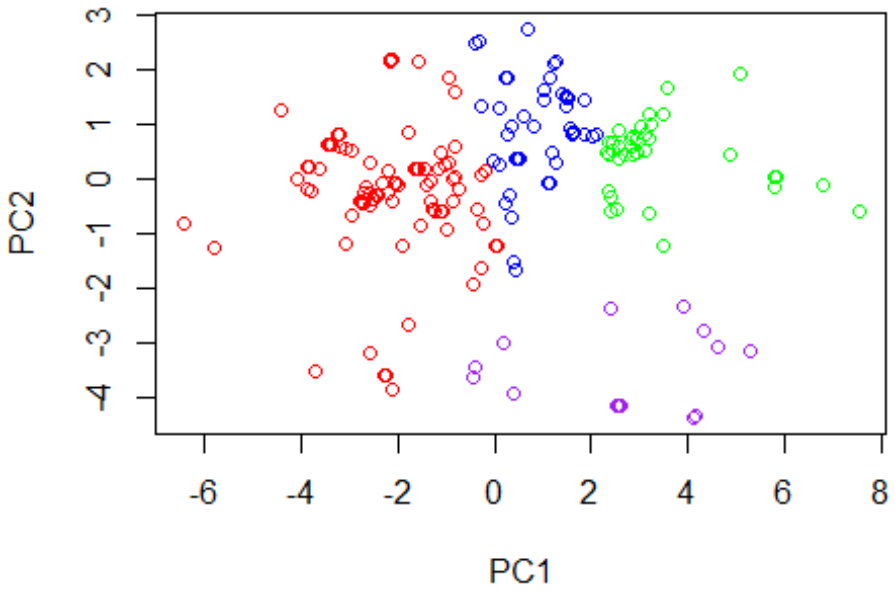
```
#org
#pca.bestdata <- princomp(best.data)
#pca.bestdata$loadings

#scaled
pca.bestdata.s <- princomp(bestdata.s)
pca.bestdata.s$loadings
```

Image 17 – PCA Output

```
## Loadings:
##                  Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
## wheelBase         0.325  0.239         0.144  0.230
## length            0.363  0.110         0.102  0.118
## width             0.349                                0.335
## height            0.143  0.333  0.446  0.472  0.302        -0.269
## curbWeight        0.376                                0.134
## engineSize        0.324        -0.216 -0.322                0.184  0.102
## bore              0.283         0.249 -0.313 -0.220        -0.265 -0.785
## stroke                  0.187 -0.751  0.129  0.284 -0.285 -0.406 -0.227
## compressionRatio        0.532 -0.226  0.103 -0.332  0.321  0.352 -0.225
## horsepower       -0.143  0.324  0.207 -0.402  0.381 -0.590  0.364 -0.114
## peakRpm          -0.104 -0.460         0.439  0.291         0.498 -0.470
## cityMpg          -0.324  0.315               -0.125
## highwayMpg       -0.338  0.267               -0.119                -0.113
## price            -0.196                -0.390  0.582  0.669
##                  Comp.9 Comp.10 Comp.11 Comp.12 Comp.13 Comp.14
## wheelBase         0.438          0.466   0.555   0.116   0.128
## length            0.190          0.341  -0.783   0.146  -0.173
## width             0.391  0.212  -0.736
## height           -0.336 -0.274  -0.300
## curbWeight       -0.167 -0.110                 -0.876
## engineSize       -0.194 -0.730                  0.344
## bore              0.136
## stroke
## compressionRatio -0.404  0.252   0.104           0.159
## horsepower       -0.153
## peakRpm                -0.120
## cityMpg           0.346 -0.340                 -0.189  -0.697
## highwayMpg        0.338 -0.351          -0.256  -0.111   0.680
## price
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
## SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var  0.071  0.071  0.071  0.071  0.071  0.071  0.071  0.071
## Cumulative Var  0.071  0.143  0.214  0.286  0.357  0.429  0.500  0.571
##                Comp.9 Comp.10 Comp.11 Comp.12 Comp.13 Comp.14
## SS loadings     1.000   1.000   1.000   1.000   1.000   1.000
## Proportion Var  0.071   0.071   0.071   0.071   0.071   0.071
## Cumulative Var  0.643   0.714   0.786   0.857   0.929   1.000
```

```
#plotting
vectcol4 <- c("blue","green","red", "purple")

#do PCA outside plot function
pca.best.s <-prcomp(bestdata.s)$x

#observe loadings
#pca.best.s$loadings

#plot
plot(pca.best.s, col = vectcol4[kms4$cluster])
```

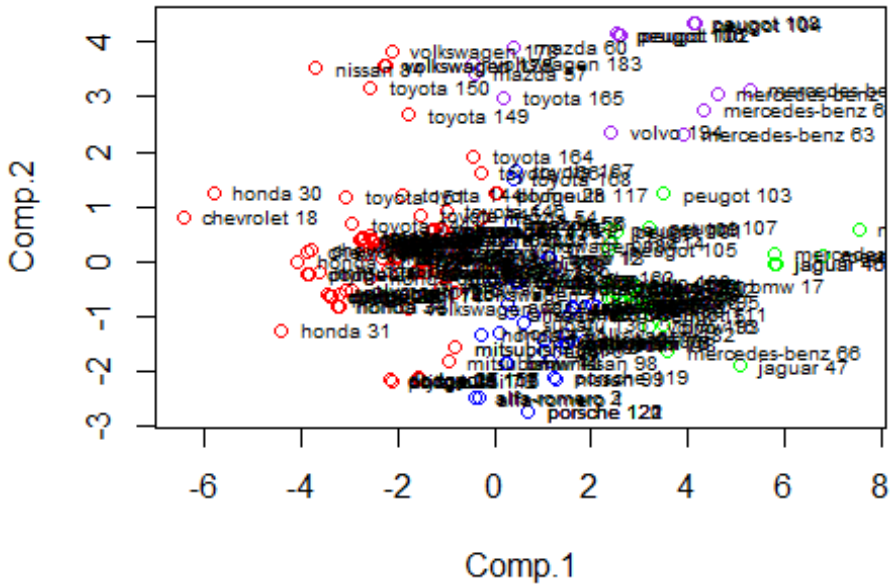
Image 18 – PCA Plot



Change PC's and plot various clusters

```
plot(pca.bestdata.s$score[, 1:2], col = vectcol4[kms4$cluster])
text(x=pca.bestdata.s$score[,1], y=pca.bestdata.s$score[,2], labels = (row.names(bestdata)), pos = 4,
cex = .6)
```

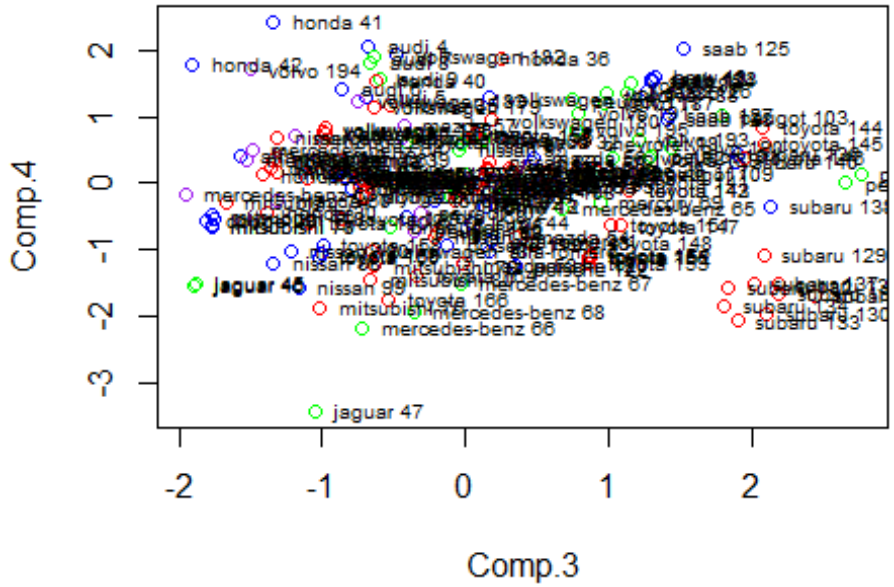
Image 19 – PCA Plot – Comp 1 and Comp 2



#plot(pca.bestdata.s$score[, 2:3], col = vectcol4[kms4$cluster])
#text(x=pca.bestdata.s$score[,2], y=pca.bestdata.s$score[,3], labels = (row.names(bestdata)), pos = 4, cex = .6)

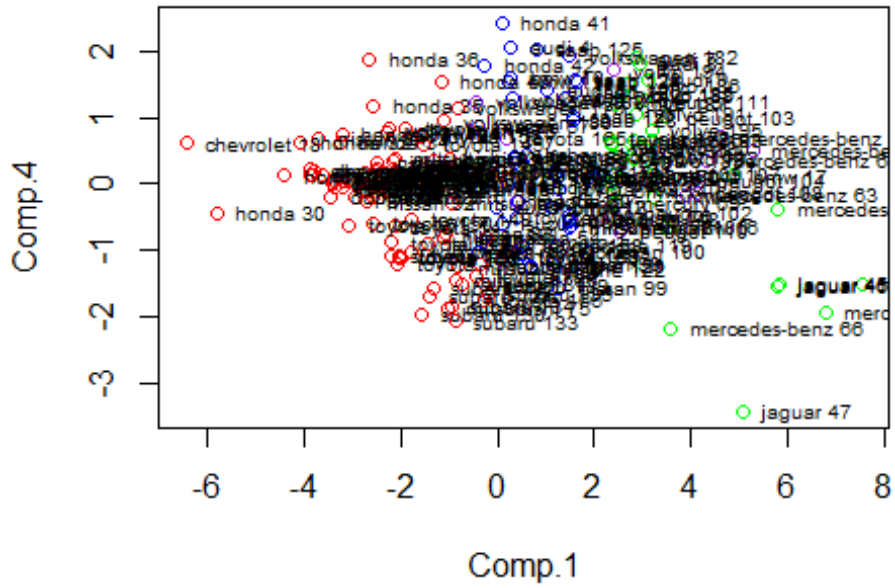
plot(pca.bestdata.s$score[, 3:4], col = vectcol4[kms4$cluster])
text(x=pca.bestdata.s$score[,3], y=pca.bestdata.s$score[,4], labels = (row.names(bestdata)), pos = 4, cex = .6)

Image 20 – PCA Plot – Comp 3 and Comp 4



plot(pca.bestdata.s$score[, c(1,4)], col = vectcol4[kms4$cluster])
text(x=pca.bestdata.s$score[,1], y=pca.bestdata.s$score[,4], labels = (row.names(bestdata)), pos = 4, cex = .6)

Image 21 – PCA Plot – Comp 1 and Comp 4



For 4 clusters observing 3:4, we see subarus cluster in a higher comp3, and low comp4.

1.5. Confirmatory Factor Analysis (CFA)

We attempted to run the CFA analysis with our data. Our initial plan was to reduce the EFA results from 4 factors, which we have had shown in previously to two factors so we could facilitate interpretation and easier the coding and txt file. However, up to now, we were unable to detect why we are getting error on our code. Due to the short deadline, we did not have the time to debug and find the issue. Nathan, and I tried to make our CFA, but neither of us were successful on that. Below is our code:

```
# RUN EFA Analysis First with 2 Factors
# Get Correlation:
mdata.cor
# Run EFA:
car.fa2 <- factanal(mdata, factors=2)
car.fa2
car.fa2$loadings
```

Image 22 – EFA Output for 2 Factors

```
Call:
factanal(x = mdata, factors = 2)

Uniquenesses:
       wheel-base            length             width            height       curb-weight       engine-size              bore            stroke
            0.166             0.102             0.174             0.709             0.089             0.348             0.545             0.945
compression-ratio        horsepower          peak-rpm          city-mpg       highway-mpg             price
            0.483             0.805             0.687             0.007             0.040             0.800

Loadings:
                  Factor1 Factor2
wheel-base         0.632   0.659
length             0.791   0.521
width              0.752   0.510
height             0.196   0.502
curb-weight        0.859   0.415
engine-size        0.769   0.247
bore               0.646   0.194
stroke                     0.226
compression-ratio -0.218   0.685
horsepower        -0.418   0.142
peak-rpm                  -0.559
city-mpg          -0.982   0.168
highway-mpg       -0.977
price             -0.412  -0.174

                 Factor1 Factor2
SS loadings        5.692   2.407
Proportion Var     0.407   0.172
Cumulative Var     0.407   0.579

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 348.5 on 64 degrees of freedom.
The p-value is 9.33e-41
> car.fa2$loadings

Loadings:
                  Factor1 Factor2
wheel-base         0.632   0.659
length             0.791   0.521
width              0.752   0.510
height             0.196   0.502
curb-weight        0.859   0.415
engine-size        0.769   0.247
bore               0.646   0.194
stroke                     0.226
compression-ratio -0.218   0.685
horsepower        -0.418   0.142
peak-rpm                  -0.559
city-mpg          -0.982   0.168
highway-mpg       -0.977
price             -0.412  -0.174

                 Factor1 Factor2
SS loadings        5.692   2.407
Proportion Var     0.407   0.172
Cumulative Var     0.407   0.579
```

Next, we proceed to create the text file based on two factor as following: we called factor one "Dimension" and factor 2 "Status".

Image 23 – Text File

```
Dimension             -> wheel-base, lambda1, NA
Dimension             -> length, lambda2, NA
Dimension             -> width, lambda3, NA
Dimension             -> height, lambda4, NA
Dimension             -> curb-weight, lambda5, NA
Dimension             -> engine-size, lambda6, NA
Dimension             -> bore, lambda7, NA
Status                -> stroke, lambda8, NA
Status                -> compression-ratio, lambda9, NA
Status                -> horsepower, lambda10, NA
Status                -> peak-rpm, lambda11, NA
Status                -> city-mpg, lambda12, NA
Status                -> highway-mpg, lambda13, NA
Status                -> price, lambda14, NA
Dimension            <-> Status, rho, NA
wheel-base           <-> wheel-base, theta1, NA
length               <-> length, theta2, NA
width                <-> width, theta3, NA
height               <-> height, theta4, NA
curb-weight          <-> curb-weight, theta5, NA
engine-size          <-> engine-size, theta6, NA
bore                 <-> bore, theta7, NA
stroke               <-> stroke, theta8, NA
compression-ratio    <-> compression-ratio, theta9, NA
horsepower           <-> horsepower, theta10, NA
peak-rpm             <-> peak-rpm, theta11, NA
city-mpg             <-> city-mpg, theta12, NA
highway-mpg          <-> highway-mpg, theta13, NA
price                <-> price, theta14, NA
Dimension            <-> Dimension, NA, 1
Status               <-> Status, NA, 1
```

```
install.packages("sem")
library("sem")
install.packages("semPlot")
library(semPlot)

#Bring txt file model:
car_txt <- read.table("~/0LUCIANO/Data Science/08_Multivariate Analysis/0_Project/car_project.txt")
cars_model <- specifyModel(file = "~/0LUCIANO/Data Science/08_Multivariate
Analysis/0_Project/car_project.txt")

#check number of roll?
nrow(mdata)

# check for potential NAs:
is.na(mdata)

# Run CFA model (correlation plus model text file plu # of rows)
cars_sem <- sem(cars_model, mdata.cor, 194)
```

The error we found:

```
Error in sem.default(ram, S = S, N = N, raw = raw, data = data, pattern.numbe
r = pattern.number,  :
  The model has negative degrees of freedom = -1
In addition: Warning message:
In sem.semmod(cars_model, mdata.cor, 194) :
  The following observed variables are in the input covariance or raw-moment
matrix but do not appear in the model:
wheel-base, curb-weight, engine-size, compression-ratio, peak-rpm, city-mpg,
highway-mpg
```

```
summary(cars_sem)
Error in summary(cars_sem) : object 'cars_sem' not found
```

2. Conclusions

2.1. Conclusions

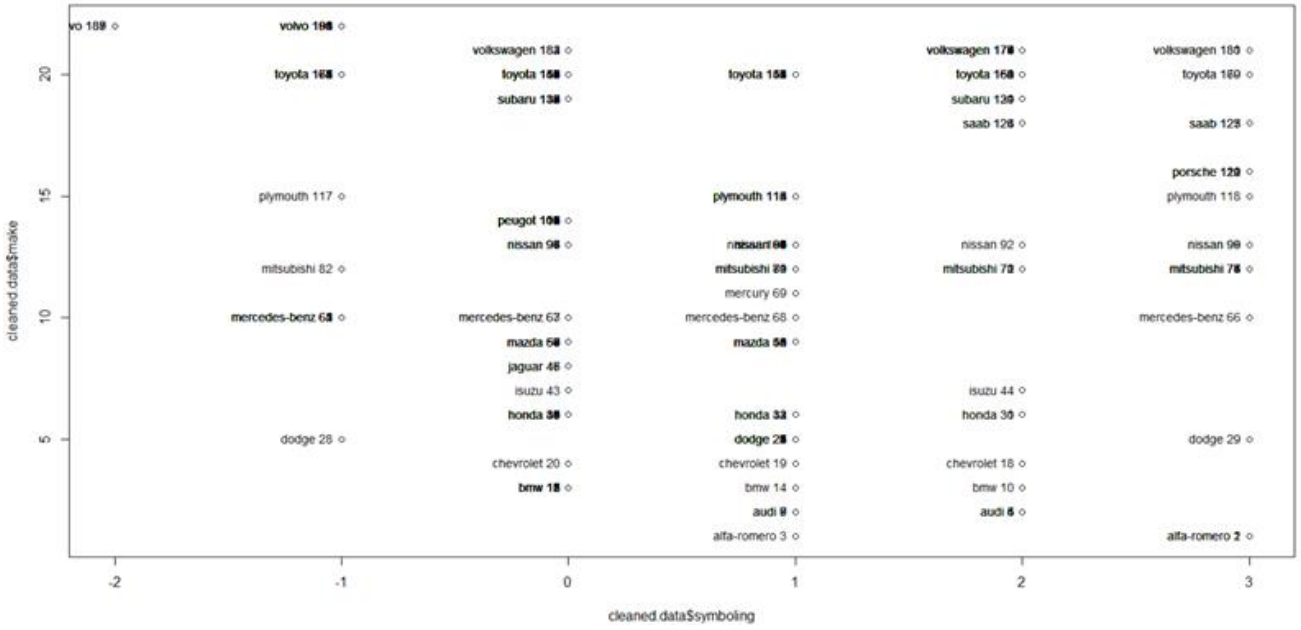
We were able to confirm some strong correlation between, for example, price and engine-size, and city-mpg and highway-mpg. We see that, nowadays, the auto industry is offering and selling bigger vehicles and from our analysis we confirm that there is also a correlation between price and width, which validates why vehicles are getting expensive today. Obviously, there are many other reasons for data. Our Heatmap with Dendrogram really shows those correlations very well and also gives a glance of how our clustering look like.

On the other hand, our MDS analysis shows that the size of the vehicle was not closely related to the MPG as one would expect them to be. It seems that MPG had more impact on the Price than the Curb-Weight of the vehicles. After running PCA, we see that 2 components can explain 70% of our data. Also, our WGSS analysis tells us that we have, potentially, up to 3 to 4 cluster in our data, which matches with our mclust result. We actually dedicated an important and big part of our project to run cluster analysis.

As a new analysis, we were also interested to see how risky or safer manufactures are. So we run the following analysis to help us to visualize that. The variables that measures the risk in this data set is "symboling". A value of +3 indicates that the auto is risky, -3 that it is probably pretty safe.

```
# code to plot symboling (risk) and make:
plot(cleaned.data$symboling, cleaned.data$make)
text(cleaned.data, labels = (row.names(bestdata)), pos = 4,  cex = .6)
```

Image 24 – Risk versus Make



From the above result; we see a tendency for the vehicles, in general, to be less safe than we should expected. We also see makes, like mercedez-benz, navigating across, basically, all ranges of risk. This may suggest that different models have different set ups, which might be correlated with the price of the vehicle. A good comparison would be seeing how those makes are working on mitigate risk nowadays, and see if the trend has shifted toward to safety.