Dimensionality Reduction using Principal Component Analysis and Multi-dimensional scaling

Prabin Kharel

Let's use the 'mtcars' dataset for Principal Component Analysis and multidimensional scaling. Let's view the structure of the dataset.

```
#first check the dataset
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
#let's check the structure of dataset
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

Since we have a lot of components in the mtcars dataset, assuming lesser influence of the binary variables, we get rid of vs and am from the mtcars dataset and create a subset with remaining components.

```
#dropping two binary variable and making a subset "mtcars.subset"
mtcars.subset <- subset(mtcars, select = -c(vs,am))
```

Let's fit PCA in the subset and get summary.

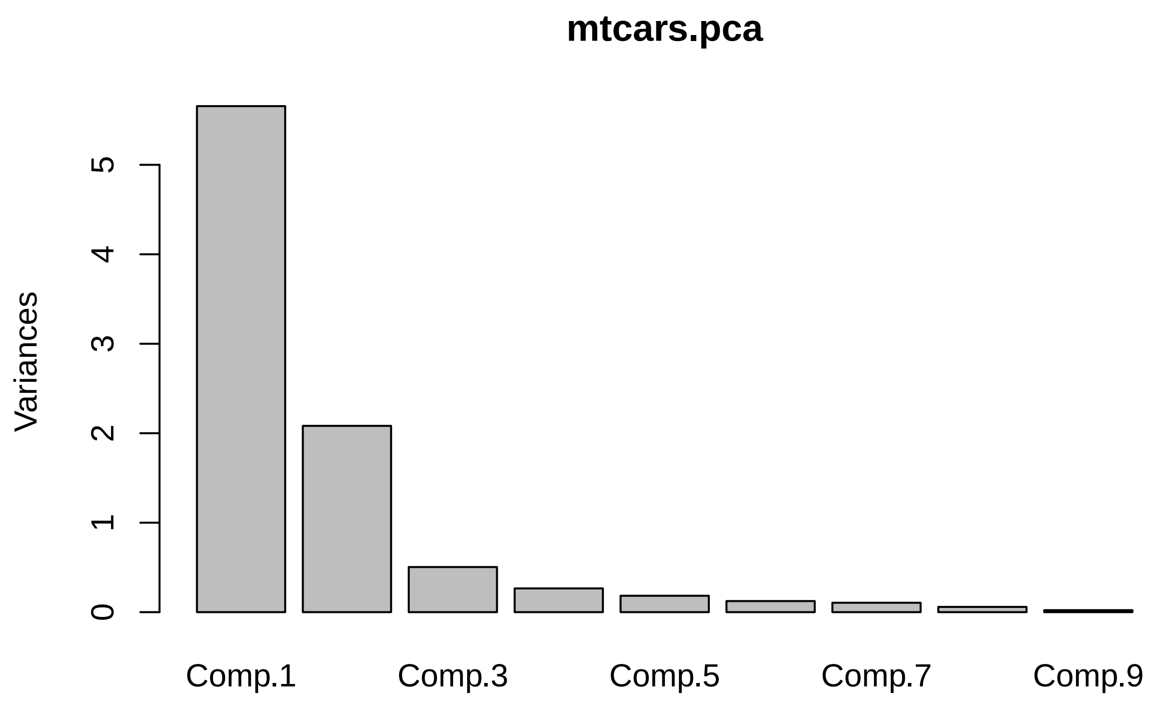
```
#fitting PCA in the mtcars.subset as mtcars.pca using the princomp() function
mtcars.pca <- princomp(mtcars.subset, cor = T, scores = T)
```

```
summary(mtcars.pca)
```

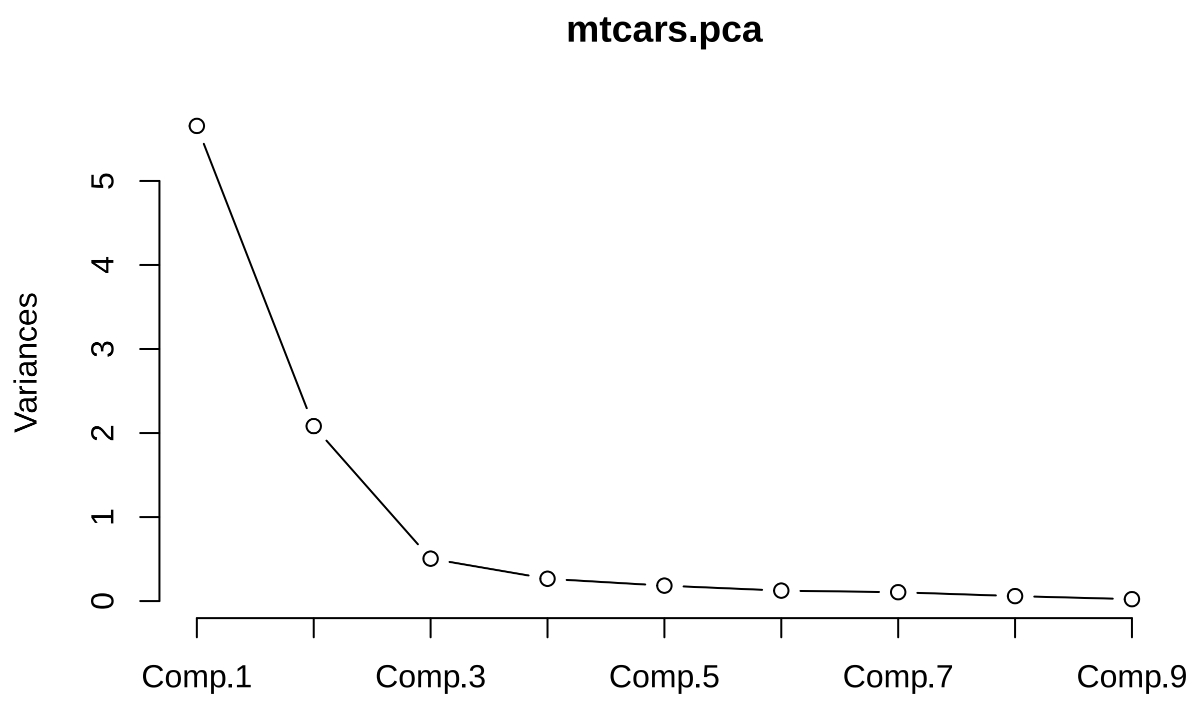
```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3    Comp.4     Comp.5
## Standard deviation     2.3782219 1.4429485 0.71008086 0.5148082 0.42797037
## Proportion of Variance 0.6284377 0.2313445 0.05602387 0.0294475 0.02035096
## Cumulative Proportion  0.6284377 0.8597822 0.91580607 0.9452536 0.96560453
##                           Comp.6     Comp.7      Comp.8      Comp.9
## Standard deviation     0.3518426 0.32413257 0.241896155 0.148964367
## Proportion of Variance 0.0137548 0.01167355 0.006501528 0.002465598
## Cumulative Proportion  0.9793593 0.99103287 0.997534402 1.000000000
```

Visualization of the PCs

```
plot(mtcars.pca)
```



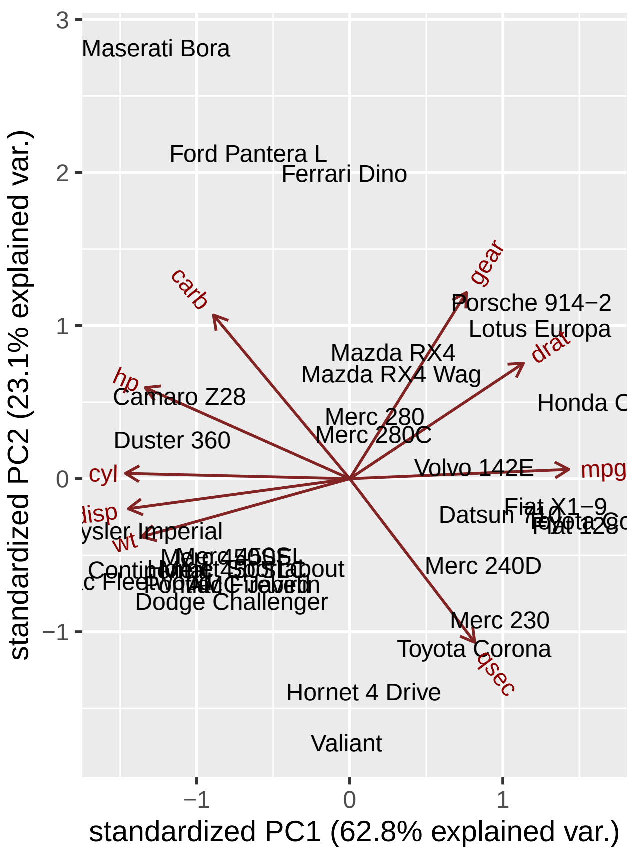
```
# scree plot
plot(mtcars.pca, type = "l")
```



```
library(ggbiplot)
```

```
## Loading required package: ggplot2
## Loading required package: plyr
## Loading required package: scales
## Loading required package: grid
```

```
ggbiplot(mtcars.pca, labels = rownames(mtcars))
```



In Principal Component Analysis, we do not determine how many components should be used. Rather we first find out the number of components to be used for final PCA by using scree plot and eigenvalues.

```
#squares of standard deviation gives us the eigenvalue
eigenvalues <- mtcars.pca$sdev^2
eigenvalues
```

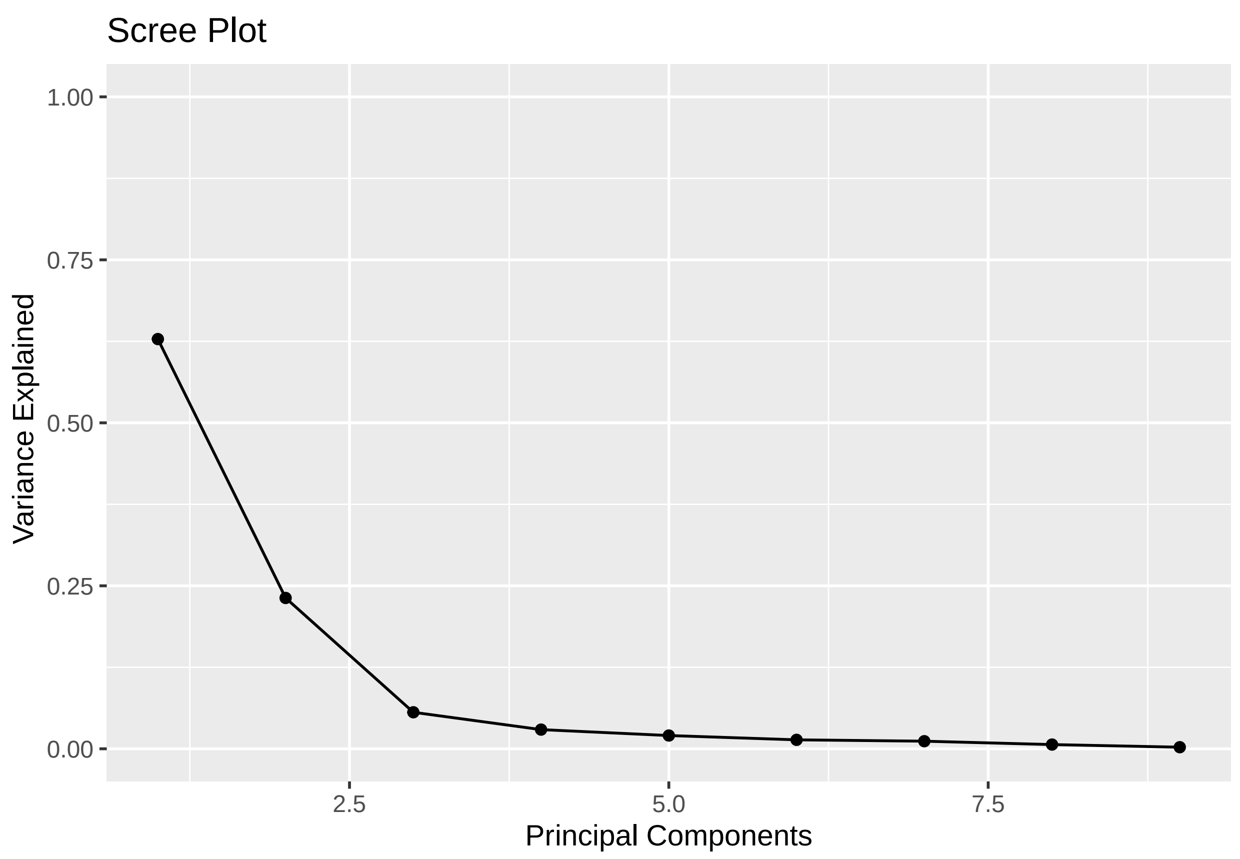
```
##     Comp.1     Comp.2     Comp.3     Comp.4     Comp.5     Comp.6     Comp.7
## 5.65593947 2.08210029 0.50421482 0.26502753 0.18315864 0.12379319 0.10506192
##     Comp.8     Comp.9
## 0.05851375 0.02219038
```

Kaiser's criteria states that any principal components with eigenvalues greater or equal to one must be retained for latent variable. Here, we see PC1 and PC2 are the only PC's that we should use for the latent variable.

Let's draw a scree plot and chose the number of components best on "first bend" of this plot.

```
library(ggplot2)
#we need to calculate total variance explained by each principal components
var_explained <- mtcars.pca$sdev^2 / sum(mtcars.pca$sdev^2)

#making a scree plot
qplot(c(1:9),var_explained) + geom_line() + xlab("Principal Components") + ylab(" Variance Explained") +
```



In the plot, the first bend occurs on 2, so we choose only 2 components.

Since from Kaiser's rule, we retained only the components with eigenvalues greater than 1 for the latent variable. In our case, there were two principal components with eigenvalues greater than 1, namely PC1 and PC2. Also, the first bend on the scree plot suggested us to take only 2 components into account. So, in agreement to both the Kaiser's rule and Scree plot, we should retain 2 components.

Now, we fit the final PCA model based on the retained components. For this we use the library "psych".

```
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:scales':
##
##     alpha, rescale

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
final.pca <- psych::principal(mtcars.subset, nfactors=2, rotate= "none")
final.pca
```

```
## Principal Components Analysis
## Call: psych::principal(r = mtcars.subset, nfactors = 2, rotate = "none")
## Standardized loadings (pattern matrix) based upon correlation matrix
##        PC1   PC2   h2    u2 com
## mpg  -0.93  0.04 0.88 0.124 1.0
## cyl   0.96  0.02 0.92 0.083 1.0
## disp  0.94 -0.13 0.91 0.091 1.0
## hp    0.87  0.39 0.91 0.087 1.4
## drat -0.74  0.49 0.79 0.207 1.7
## wt    0.89 -0.25 0.85 0.150 1.2
## qsec -0.53 -0.70 0.77 0.227 1.9
## gear -0.50  0.79 0.88 0.120 1.7
## carb  0.58  0.70 0.83 0.173 1.9
##
##                        PC1  PC2
## SS loadings           5.66 2.08
## Proportion Var        0.63 0.23
## Cumulative Var        0.63 0.86
## Proportion Explained  0.73 0.27
## Cumulative Proportion 0.73 1.00
##
## Mean item complexity =  1.4
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.05
##  with the empirical chi square  6.04  with prob <  1
##
## Fit based upon off diagonal values = 0.99
```

Of the total variance explained, PC1 accounts for 63% and PC2 accounts for 23% of the variance, summing to 86% of the total variance. So, these two were sufficient for our PCA.

Now we can see which cars are similar to one another. Here the Maserati Bora, Ferrari Dino and Ford Pantera L all cluster together at the top which makes sense, as all of these are sports cars.

Let's compare the loadings from the first and final PCA's. Loadings are the coefficients of the linear combination of the initial variables from which the principal components are constructed. When the loadings has positive value, it means higher scores are correlated to higher component scores and negative value in the loadings mean that higher score are correlated to lower component scores.

```
head(mtcars.pca$loadings)
```

```
##          Comp.1      Comp.2      Comp.3       Comp.4     Comp.5      Comp.6
## mpg   0.3931477  0.02753861  0.22119309  0.006126378  0.3207620  0.72015586
## cyl  -0.4025537  0.01570975  0.25231615 -0.040700251 -0.1171397  0.22432550
## disp -0.3973528 -0.08888469  0.07825139 -0.339493732  0.4867849 -0.01967516
## hp   -0.3670814  0.26941371  0.01721159 -0.068300993  0.2947317  0.35394225
## drat  0.3118165  0.34165268 -0.14995507 -0.845658485 -0.1619259 -0.01536794
## wt   -0.3734771 -0.17194306 -0.45373418 -0.191260029  0.1874822 -0.08377237
##           Comp.7      Comp.8      Comp.9
## mpg   0.38138068  0.12465987  0.11492862
## cyl   0.15893251 -0.81032177  0.16266295
## disp  0.18233095  0.06416707 -0.66190812
## hp   -0.69620751  0.16573993  0.25177306
## drat -0.04767957 -0.13505066  0.03809096
## wt    0.42777608  0.19839375  0.56918844
```

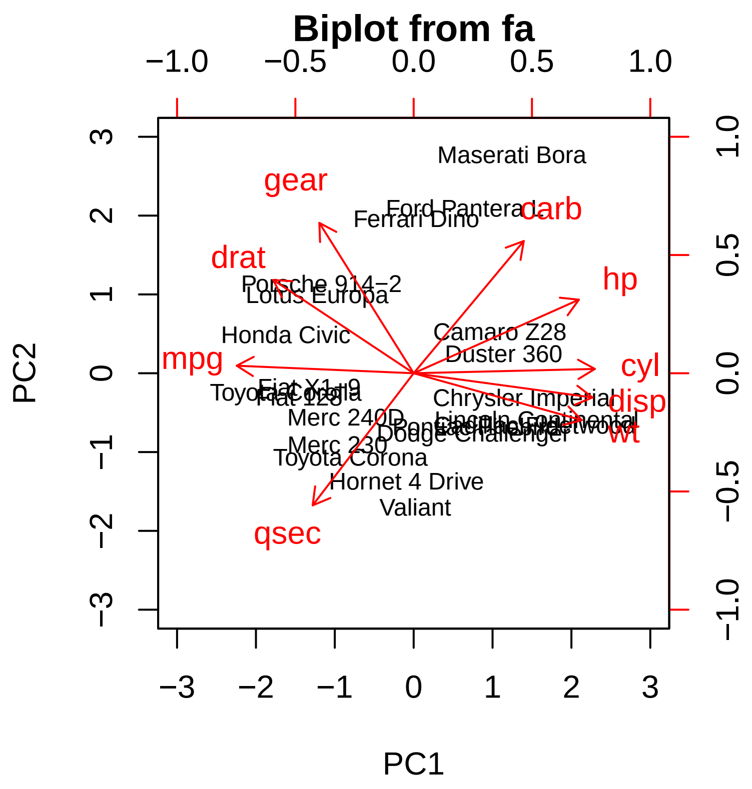
```
head(final.pca$loadings)
```

```
##             PC1         PC2
## mpg  -0.9349924  0.03973679
## cyl   0.9573620  0.02266837
## disp  0.9449932 -0.12825603
## hp    0.8730011  0.38875010
## drat -0.7415688  0.49298721
## wt    0.8882114 -0.24810498
```

Here, PC1 has strong negative loadings for mpg and drat and strong positive loadings for cyl and disp. This likely represents that axis 1 is more correlated to cyl and disp and less correlated to mpg and drat. Similarly we can analyze for the 2nd axis as well.

let's look at the biplot of these two component loadings.

```
biplot(final.pca,labels = rownames(mtcars.subset))
```



We see that mpg and cyl have very high values but are moving in the opposite direction. Similar thing was observed from their loading values in PC1. Although, loadings values of PC2 does not show much of difference but since PC1 does explain the major portion of the variance we can see the plot also gives similar result.

Now let's compare the scores of first and final PCA's. Scores are just a linear combination of the raw data, with weighting given by the loadings.This means the scores on all the PCs are just the linear combination of the original value and the weigthing given by loadings.

```
head(mtcars.pca$scores)
```

```
##                        Comp.1     Comp.2     Comp.3      Comp.4      Comp.5
## Mazda RX4          0.67485176  1.1922239  0.2075865  0.12803392 -0.76404073
## Mazda RX4 Wag      0.64739389  0.9925769 -0.1125504  0.08704801 -0.66719589
## Datsun 710         2.33653412 -0.3318150  0.2135122  0.11036335  0.07744294
## Hornet 4 Drive     0.21874167 -2.0084411  0.3347400  0.31299155  0.24782081
## Hornet Sportabout -1.61236724 -0.8419890  1.0495215 -0.14974247  0.22626754
## Valiant           -0.05039885 -2.4858332 -0.1135663  0.88549481  0.12776087
##                        Comp.6      Comp.7       Comp.8       Comp.9
## Mazda RX4         -0.12706898  0.43035095  0.003311314 -0.169724100
## Mazda RX4 Wag     -0.06725355  0.45566941 -0.057549593 -0.072737641
## Datsun 710        -0.57600804 -0.39230244  0.205268522  0.116337227
## Hornet 4 Drive     0.08516563 -0.03352155  0.024093561 -0.147579996
## Hornet Sportabout  0.18572940  0.05886501 -0.154780223 -0.157120566
## Valiant           -0.23411734 -0.22810775 -0.100241845 -0.004301647
```

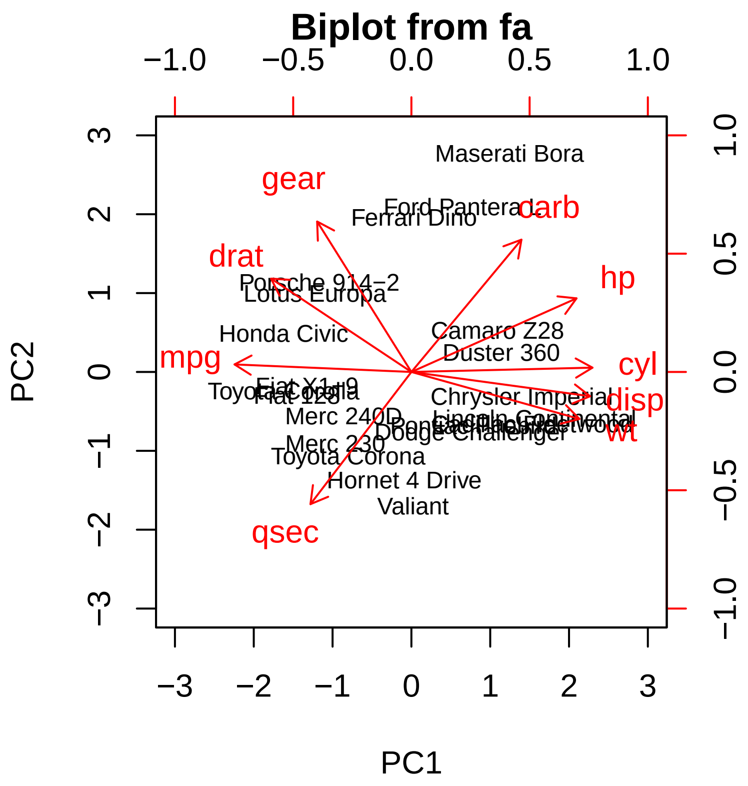
```
head(final.pca$scores)
```

```
##                           PC1        PC2
## Mazda RX4         -0.27929417  0.8132290
## Mazda RX4 Wag     -0.26793045  0.6770476
## Datsun 710        -0.96699807 -0.2263347
## Hornet 4 Drive    -0.09052843 -1.3699797
## Hornet Sportabout  0.66729435 -0.5743299
## Valiant            0.02085807 -1.6956141
```

This means the scores on PC1 and PC2 are just the linear combination of the original value and the weighting given by loadings.

Following figure shows the biplot of these two component scores.

```
biplot(final.pca,rownames(mtcars.subset))
```



Here, we see that mpg and cyl have very high values but are moving in the opposite direction. Since scores are the linear combinations of raw data and loadings, this might be because of their higher loading values in opposite directions.

NOw, let's get dissimilar distance of all the variables of mtcars data as mtcars.dist and fit classical multi-dimensional scaling model with the mtcars.dist in 2-dimensional state as cars.mds.2d

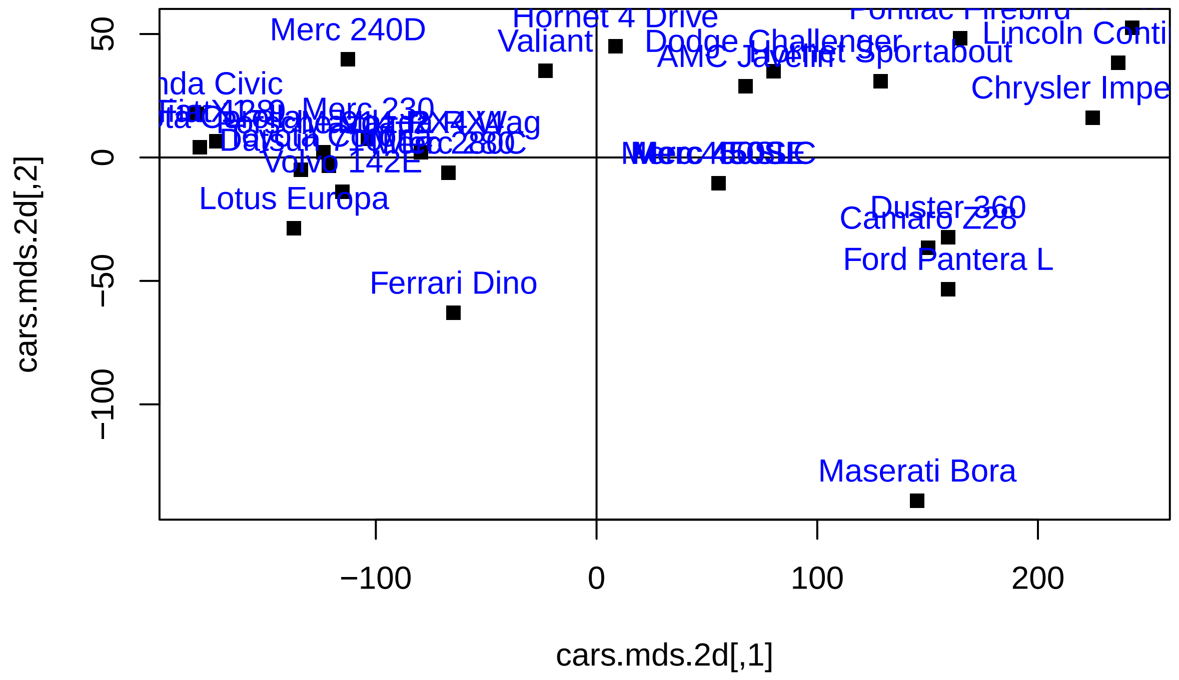
```
mtcars.dist<-dist(mtcars.subset)
head(mtcars.dist)
```

```
## [1]   0.6153251  54.8994991  98.1023283 210.3350625  65.4564955 241.4055778
```

```
cars.mds.2d <- cmdscale(mtcars.dist)
summary(cars.mds.2d)
```

```
##        V1                 V2
##  Min.   :-181.07   Min.   :-139.047
##  1st Qu.:-116.69   1st Qu.: -10.373
##  Median : -43.99   Median :   2.144
##  Mean   :   0.00   Mean   :   0.000
##  3rd Qu.: 132.85   3rd Qu.:  29.375
##  Max.   : 242.81   Max.   :  52.503
```

```
plot(cars.mds.2d, pch=15)
abline(h=0,v=0)
text(cars.mds.2d, pos = 3, labels = rownames(mtcars.subset),col="blue")
```

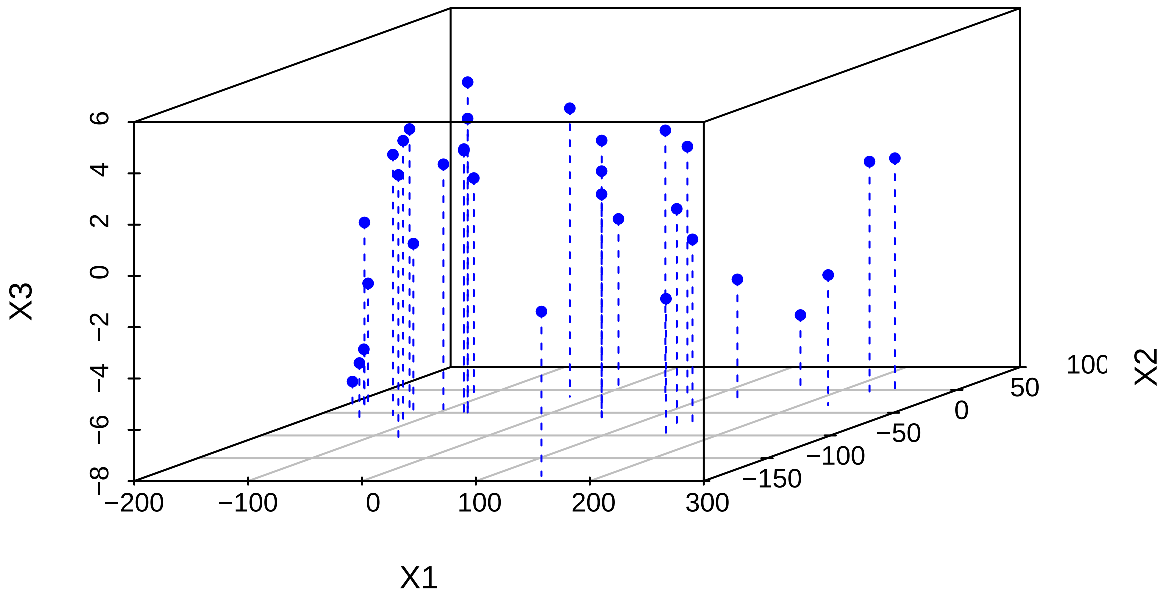


let's fit the same mtcars.dist again with MDS but this time in 3-dimension state cars.mds.3d

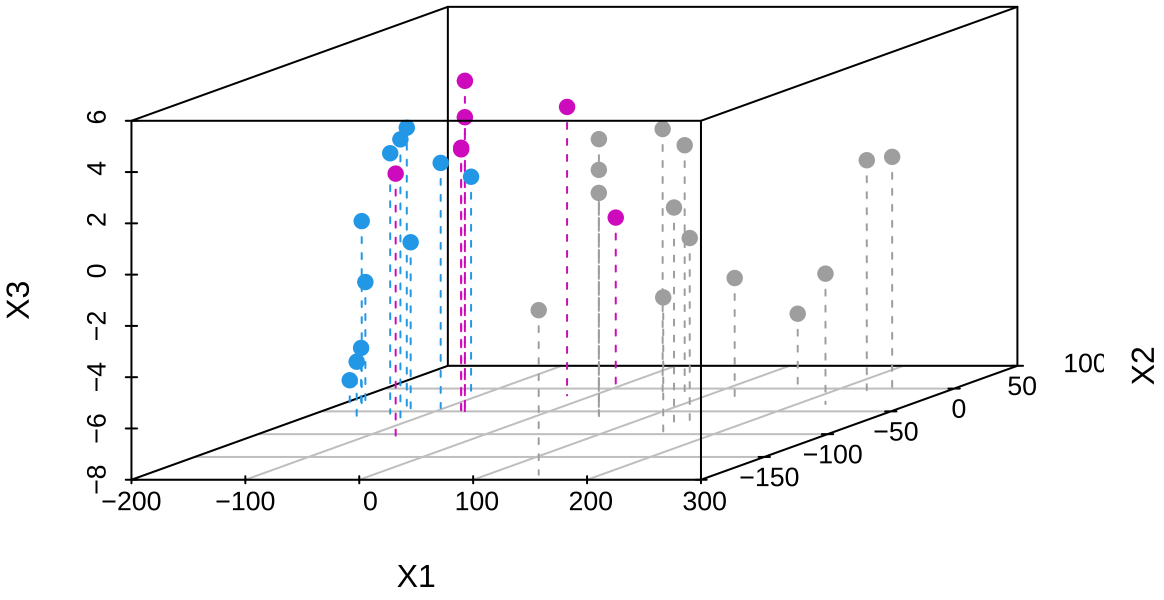
```
cars.mds.3d <- cmdscale(mtcars.dist,k=3)
summary(cars.mds.3d)
```

```
##        V1                 V2                  V3
##  Min.   :-181.07   Min.   :-139.047   Min.   :-6.8611
##  1st Qu.:-116.69   1st Qu.: -10.373   1st Qu.:-1.8374
##  Median : -43.99   Median :   2.144   Median : 0.8492
##  Mean   :   0.00   Mean   :   0.000   Mean   : 0.0000
##  3rd Qu.: 132.85   3rd Qu.:  29.375   3rd Qu.: 2.2806
##  Max.   : 242.81   Max.   :  52.503   Max.   : 5.0029
```

```
library(scatterplot3d)
cars.mds.3d <- data.frame(cars.mds.3d)
scatterplot3d(cars.mds.3d, type="h", pch=20, lty.hplot=2,color = "blue")
```



```
library(scatterplot3d)
cars.mds.3d <- data.frame(cmdscale(mtcars.dist, k = 3))
scatterplot3d(cars.mds.3d, type = "h", pch = 19, lty.hplot = 2, color = mtcars$cyl)
```



Here, we have the V1,V2 and V3 of the multi dimensional scaling model represented by different colors for easier visualization.

While PCA retains the important dimensions for us (PC1 and PC2) with help of eigenvalues and the scree plot, MDS gives us configuration to m-dimensions and reproduces dissimilarities on the map more directly and accurately than PCA. Also for PCA we do not define how many dimensions we need (we got 2 from eigenvalues and scree plot and not because we wanted to make it 2) but in MDS we can define how many dimensions do we want (we made a multi-dimensional scaling model with 2 and 3 dimensions).