Principal Component Analysis (PCA)

R Programming: Principal Component Analysis

Example

```r
## Example
# ---
# Perform and visualize PCA in the given mtcars dataset
# ---
df <- mtcars
head(df)
```

```
##                    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
```

```r
# Selecting the numerical data (excluding the categorical variables vs and am)
# ---
#
df <- mtcars[,c(1:7,10,11)]
head(df)
```

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

```r
# We then pass df to the prcomp(). We also set two arguments, center and scale,
# to be TRUE then preview our object with summary

mtcars.pca <- prcomp(mtcars[,c(1:7,10,11)], center = TRUE, scale. = TRUE)
summary(mtcars.pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
```

```
## Standard deviation     2.3782 1.4429 0.71008 0.51481 0.42797 0.35184 0.32413
## Proportion of Variance 0.6284 0.2313 0.05602 0.02945 0.02035 0.01375 0.01167
## Cumulative Proportion  0.6284 0.8598 0.91581 0.94525 0.96560 0.97936 0.99103
##                           PC8     PC9
## Standard deviation     0.2419 0.14896
## Proportion of Variance 0.0065 0.00247
## Cumulative Proportion  0.9975 1.00000
```

- Here we have 9 principal components
- Each principal component explain a proportion of the total variation in the dataset
- PC1 explains almost 63% of the variance in the dataset - This means that nearly two-thirds of the information in the dataset (9 variables) can be encapsulated by just one principal component

```r
# Calling str() to have a look at your PCA object

str(mtcars.pca)
```

```
## List of 5
##  $ sdev    : num [1:9] 2.378 1.443 0.71 0.515 0.428 ...
##  $ rotation: num [1:9, 1:9] -0.393 0.403 0.397 0.367 -0.312 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:9] "mpg" "cyl" "disp" "hp" ...
##   .. ..$ : chr [1:9] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:9] 20.09 6.19 230.72 146.69 3.6 ...
##   ..- attr(*, "names")= chr [1:9] "mpg" "cyl" "disp" "hp" ...
##  $ scale   : Named num [1:9] 6.027 1.786 123.939 68.563 0.535 ...
##   ..- attr(*, "names")= chr [1:9] "mpg" "cyl" "disp" "hp" ...
##  $ x       : num [1:32, 1:9] -0.664 -0.637 -2.3 -0.215 1.587 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:32] "Mazda RX4" "Mazda RX4 Wag" "Datsun 710" "Hornet 4 Drive" ...
##   .. ..$ : chr [1:9] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

```r
# Here we note that our pca object: The center point ($center), scaling ($scale),
# standard deviation(sdev) of each principal component.
# The relationship (correlation or anticorrelation, etc)
# between the initial variables and the principal components ($rotation).
# The values of each sample in terms of the principal components ($x)
```

```r
# We will now plot our pca. This will provide us with some very useful insights i.e.
# which cars are most similar to each other

# Installing our ggbiplot visualisation package
library(devtools)
```

```
## Warning: package 'devtools' was built under R version 4.0.4
```

```
## Loading required package: usethis
```

```
## Warning: package 'usethis' was built under R version 4.0.4
```

```r
install_github("vqv/ggbiplot")
```

```
## WARNING: Rtools is required to build R packages, but is not currently installed.
##
## Please download and install Rtools 4.0 from https://cran.r-project.org/bin/windows/Rtools/.
```

```
## Skipping install of 'ggbiplot' from a github remote, the SHA1 (7325e880) has not changed since last i
##   Use `force = TRUE` to force installation
```

```r
# Then Loading our ggbiplot library
#
library(ggbiplot)
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.0.4
```

```
## Loading required package: plyr
```

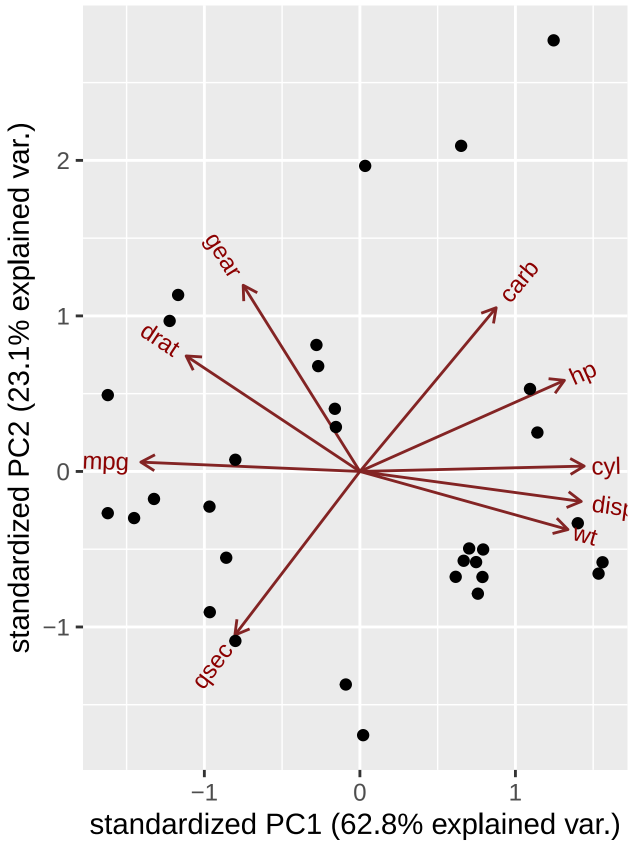
```
## Warning: package 'plyr' was built under R version 4.0.4
```

```
## Loading required package: scales
```

```
## Warning: package 'scales' was built under R version 4.0.4
```

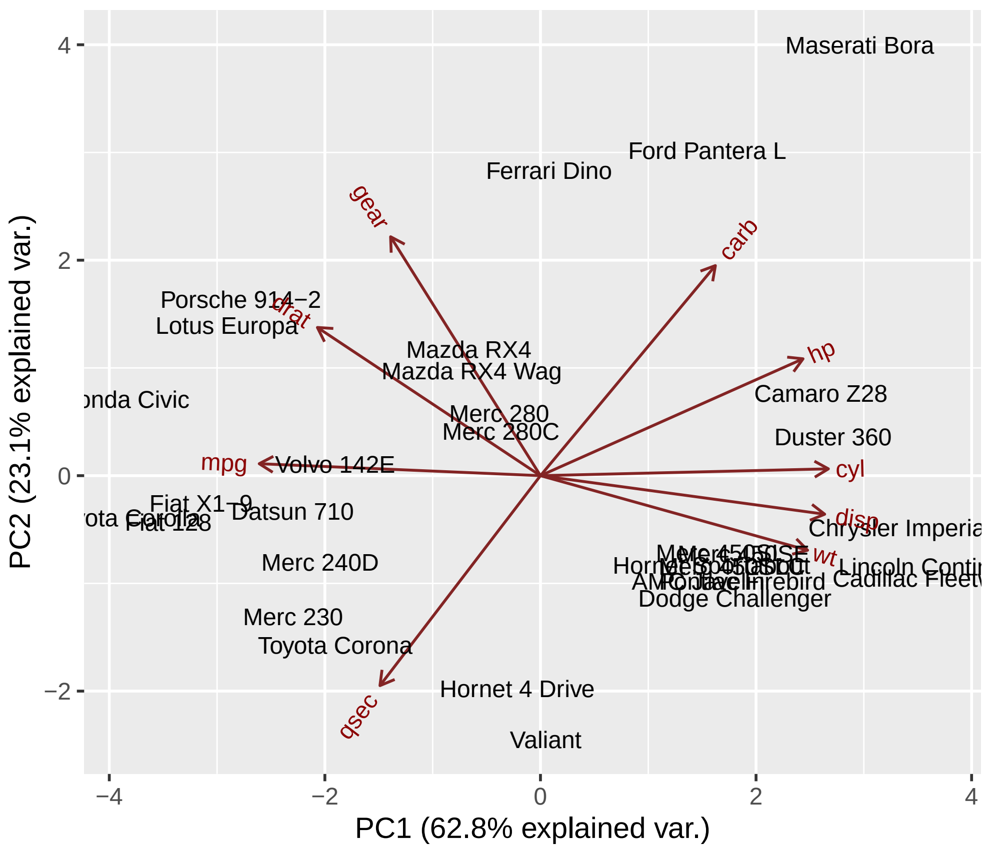
```
## Loading required package: grid
```

```r
ggbiplot(mtcars.pca)
```




```r
# From the graph we will see that the variables hp, cyl and disp contribute to PC1,
# with higher values in those variables moving the samples to the right on the plot.
```

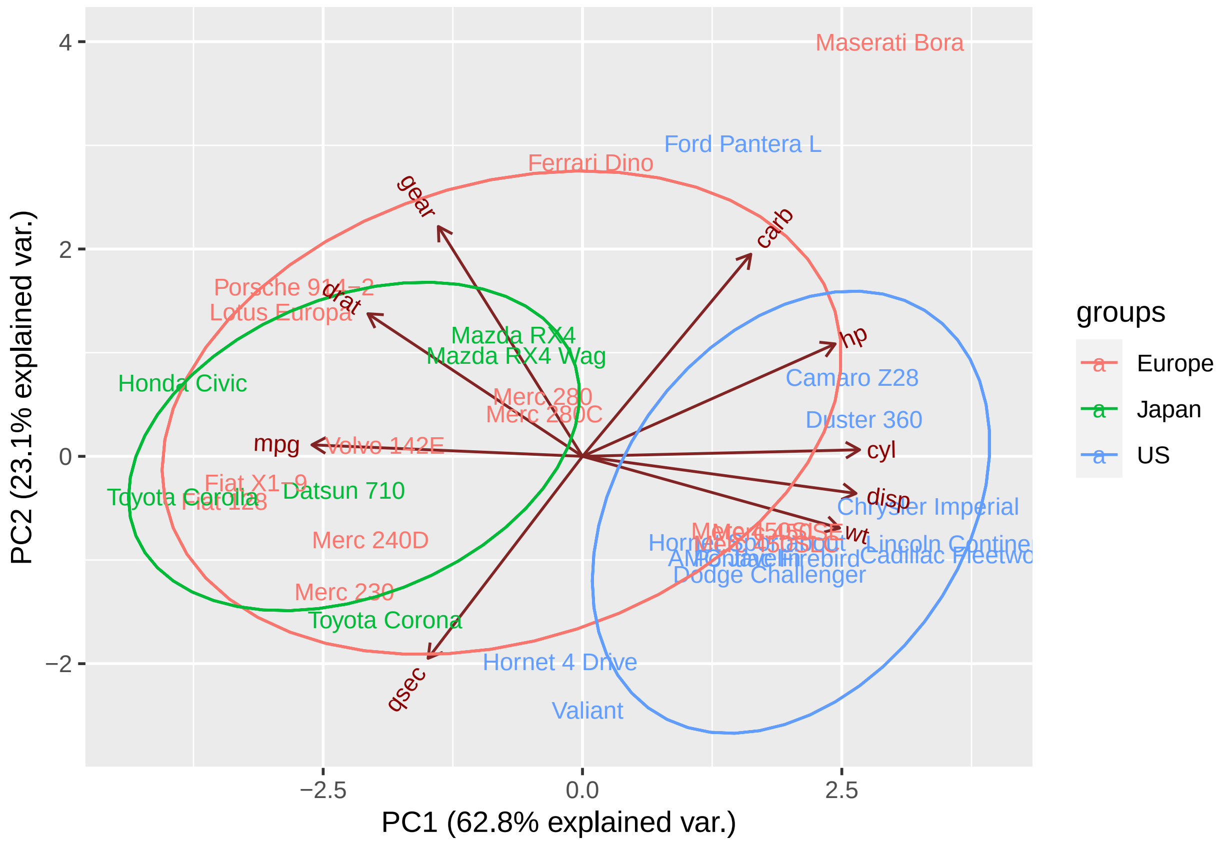
```r
# Adding more detail to the plot, we provide arguments rownames as labels
#
ggbiplot(mtcars.pca, labels=rownames(mtcars), obs.scale = 1, var.scale = 1)
```




```r
# We now see which cars are similar to one another.
# The sports cars Maserati Bora, Ferrari Dino and Ford Pantera L all cluster together at the top
```

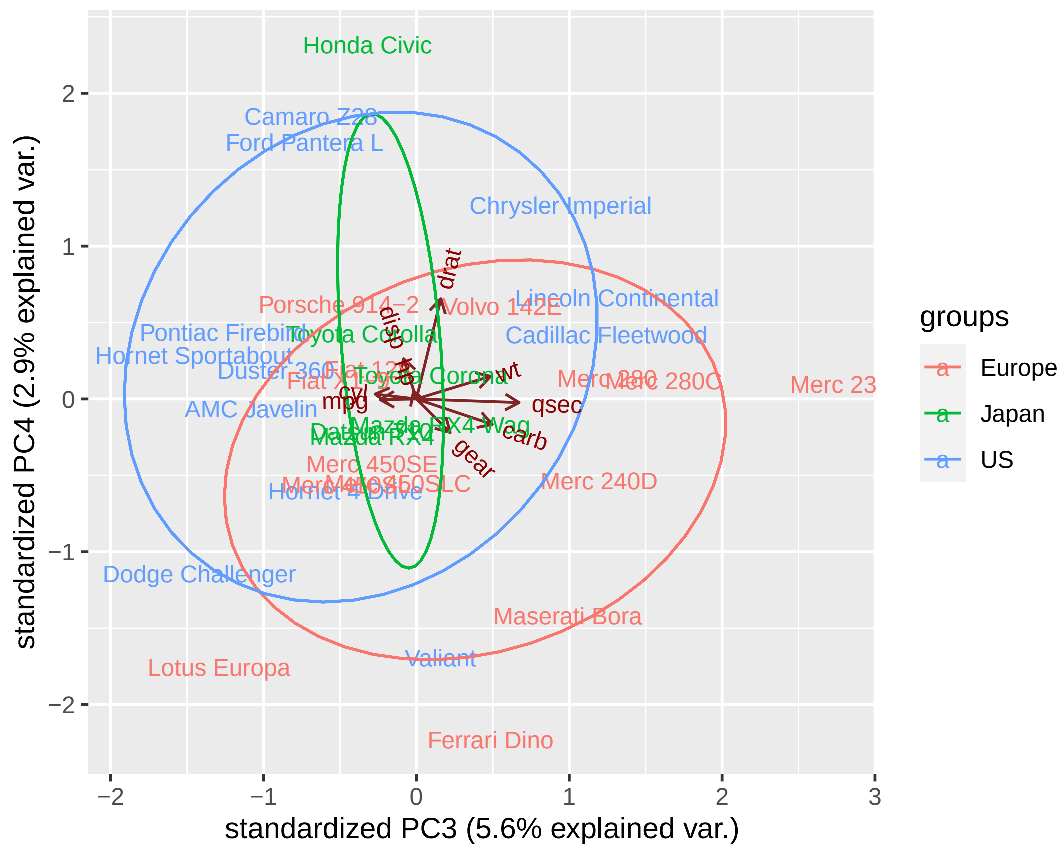
```r
# We can also look at the origin of each of the cars by putting them
# into one of three categories i.e. US, Japanese and European cars.
#
mtcars.country <- c(rep("Japan", 3), rep("US",4), rep("Europe", 7),rep("US",3), "Europe", rep("Japan", 3

ggbiplot(mtcars.pca,ellipse=TRUE,  labels=rownames(mtcars), groups=mtcars.country, obs.scale = 1, var.sc
```




```
# We get to see that US cars for a cluster on the right.
# This cluster is characterized by high values for cyl, disp and wt.
# Japanese cars are characterized by high mpg.
# European cars are somewhat in the middle and less tightly clustered that either group.
```

```
# We now plot PC3 and PC4
ggbiplot(mtcars.pca,ellipse=TRUE,choices=c(3,4),   labels=rownames(mtcars), groups=mtcars.country)
```




```
# We find it difficult to derive insights from the given plot mainly because PC3 and PC4
# explain very small percentages of the total variation, thus it would be surprising
# if we found that they were very informative and separated the groups or revealed apparent patterns.
```

Having performed PCA using this dataset, if we were to build a classification model to identify the origin of a car (i.e. European, Japanese, US) the variables cyl, disp, wt and mpg would be significant variables as seen in our PCA analysis

Challenges

```
## Challenge 1
# ---
# Question: Perform and plot PCA to the give Iris dataset. Reduce 4 dimensinal data into 2 or three dim
# Provide remarks on your analysis.
# ---
# Dataset url = http://bit.ly/IrisDataset
# ---
print(paste("IRIS DATA"))
```

```
## [1] "IRIS DATA"
```

```r
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```r
print(paste("PCA"))
```

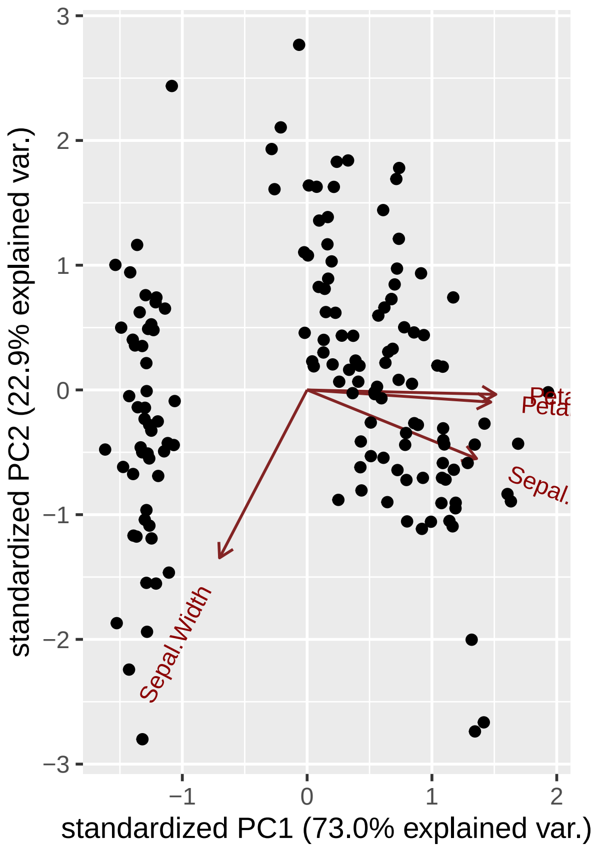
```
## [1] "PCA"
```

```r
iris_pca <- prcomp(iris[,c(1:4)], center = TRUE, scale. = TRUE)
summary(iris_pca)
```

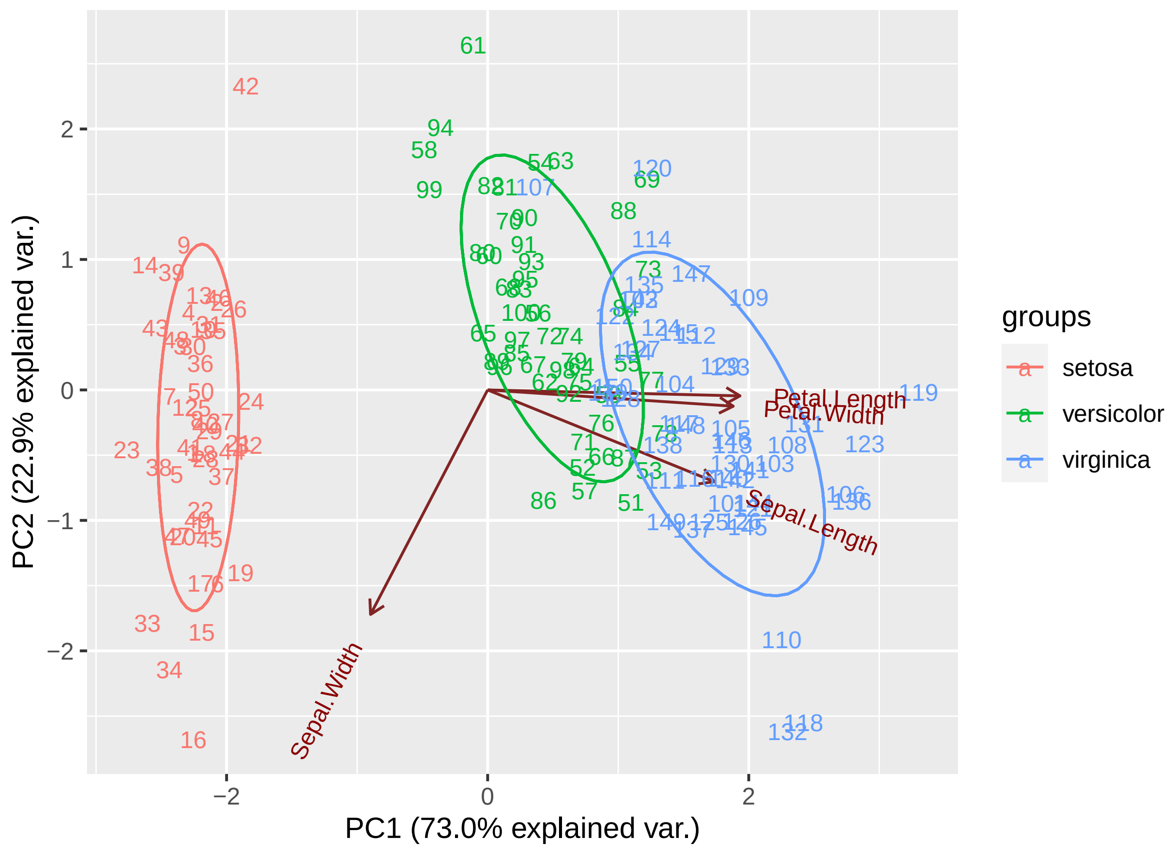
```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

Principal Component 1 (PC1) accounts for almost 73% variability in the dataset PC1 and PC2 would acount for 95.81% variability in the dataset

```r
# Since the plotting library has been loaded above, we will depend on it to do the plot
ggbiplot(iris_pca)
```




```
ggbiplot(iris_pca, ellipse=TRUE,  labels=rownames(iris), groups=iris$Species, obs.scale = 1, var.scale =
```




```r
## Challenge 2
# ---
# Question: Perform and plot PCA on the given dataset.
# ---

url = "http://bit.ly/WisconsinDataset"
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.0.4
```

```r
df2 = read.csv(url)
head(df2)
```

```
##                                X..DOCTYPE.HTML.PUBLIC.....W3C..DTD.HTML.4.01.Transitional..EN..
## 1                                                                                        <html>
## 2                                                                                        <head>
## 3 <title>UCI Machine Learning Repository: Breast Cancer Wisconsin (Diagnostic) Data Set</title>
## 4                                                                      <!-- Stylesheet link -->
## 5                                     <link rel=stylesheet type=text/css href=../assets/ml.css />
## 6                                              <script language=JavaScript type=text/javascript>
```

```r
## Challenge 3
# ---
# Question: Perform and plot the given housing dataset. Provide remarks to your analysis.
```

```r
# ---
# Dataset url = http://bit.ly/BostonHousingDataset
# ---
library(data.table)
df3 <- fread("http://bit.ly/BostonHousingDataset")
head(df3)
```

```
##       crim zn indus chas   nox    rm  age    dis rad tax ptratio      b lstat
## 1: 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2: 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3: 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4: 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5: 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6: 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##    medv
## 1: 24.0
## 2: 21.6
## 3: 34.7
## 4: 33.4
## 5: 36.2
## 6: 28.7
```

```r
# Checking the data types
str(df3)
```

```
## Classes 'data.table' and 'data.frame':   506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : int  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ b      : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

All the variables are numeric therefore fit for pca

```r
colSums(is.na(df3))
```

```
##    crim      zn   indus    chas     nox      rm     age     dis     rad     tax
##       0       0       0       0       0       0       0       0       0       0
## ptratio       b   lstat    medv
##       0       0       0       0
```

There are no missing values in any of the columns

```r
df3.pca <- prcomp(df3, center = TRUE, scale. = TRUE, rank. = 10)
summary(df3.pca)
```

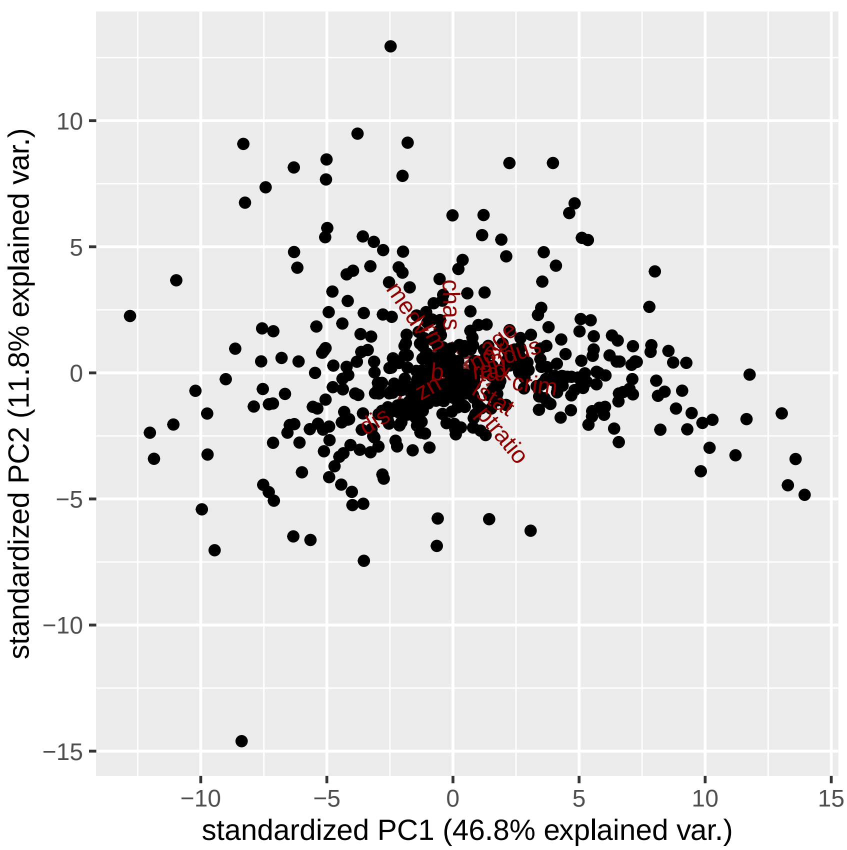
```
## Importance of first k=10 (out of 14) components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.5585 1.2843 1.16142 0.94156 0.92244 0.81241 0.73172
## Proportion of Variance 0.4676 0.1178 0.09635 0.06332 0.06078 0.04714 0.03824
## Cumulative Proportion  0.4676 0.5854 0.68174 0.74507 0.80585 0.85299 0.89123
##                            PC8    PC9    PC10
## Standard deviation     0.63488 0.5266 0.50225
## Proportion of Variance 0.02879 0.0198 0.01802
## Cumulative Proportion  0.92003 0.9398 0.95785
```

- 14 Principal Components have been created
- The first 10 principal components accounts for almost 96% variance in the data
- PC1 explains 46.76% variance in the data

```r
# Loading the ggbiplot libray for use in plotting the PCA components
library(ggbiplot)
ggbiplot(df3.pca)
```

```
## Warning in sweep(pcobj$x, 2, 1/(d * nobs.factor), FUN = "*"): STATS is longer
## than the extent of 'dim(x)[MARGIN]'

## Warning in sweep(v, 2, d^var.scale, FUN = "*"): STATS is longer than the extent
## of 'dim(x)[MARGIN]'
```



- The distinction is not very clear since PC1 & PC2 explains less than 60% of the variance