PCA using R: Visualization

Rei Sanchez-Arias, Ph.D.

Principal Component Analysis (PCA) in R

Pre-requisites

Checklist

☑ Load the `tidyverse` package

```r
library(tidyverse)
```

PCA calculation and visualization

To perform Principal Component Analysis (PCA) you will be using the function `prcomp()` from the `stats` package (you don't need to install any package to use it, since it comes with your R installation)

☑ For data visualization you will be using the `factoextra` package

```r
library(factoextra)
```

Example: 1974 Motor Trend US Magazine

Dreamer-Garage.Tumblr.com

`mtcars` dataset

Our data set will be **`mtcars`**, a built-in dataset in `R` with Motor Trend Car Road Tests. The data comprises **fuel consumption** and 10 aspects of automobile design and performance for 32 automobiles (1973 - 1974 models).

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1

Previous 1 2 3 4 5 … 8 Next

PCA and visualization

When clustering data using principal component analysis, it is often of interest to visually inspect how well the data points separate in 2D space based on principal component scores.

> With `prcomp()` the calculation is done by a singular value decomposition of the (centered and possibly scaled) data matrix, not by using `eigen()` on the covariance matrix. This is generally the preferred method for numerical accuracy.

PCA in R

```
pcaCars <- prcomp(mtcars, scale = T)
```

A quick summary of the principal component calculation shows that with only the first two principal components we can explain about 84% of the variance in the data.

```
summary(pcaCars)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6    PC7     PC8    PC9
## Standard deviation     2.5707 1.6280 0.79196 0.51923 0.47271 0.46000 0.3678 0.35057 0.2776
## Proportion of Variance 0.6008 0.2409 0.05702 0.02451 0.02031 0.01924 0.0123 0.01117 0.0070
## Cumulative Proportion  0.6008 0.8417 0.89873 0.92324 0.94356 0.96279 0.9751 0.98626 0.9933
```

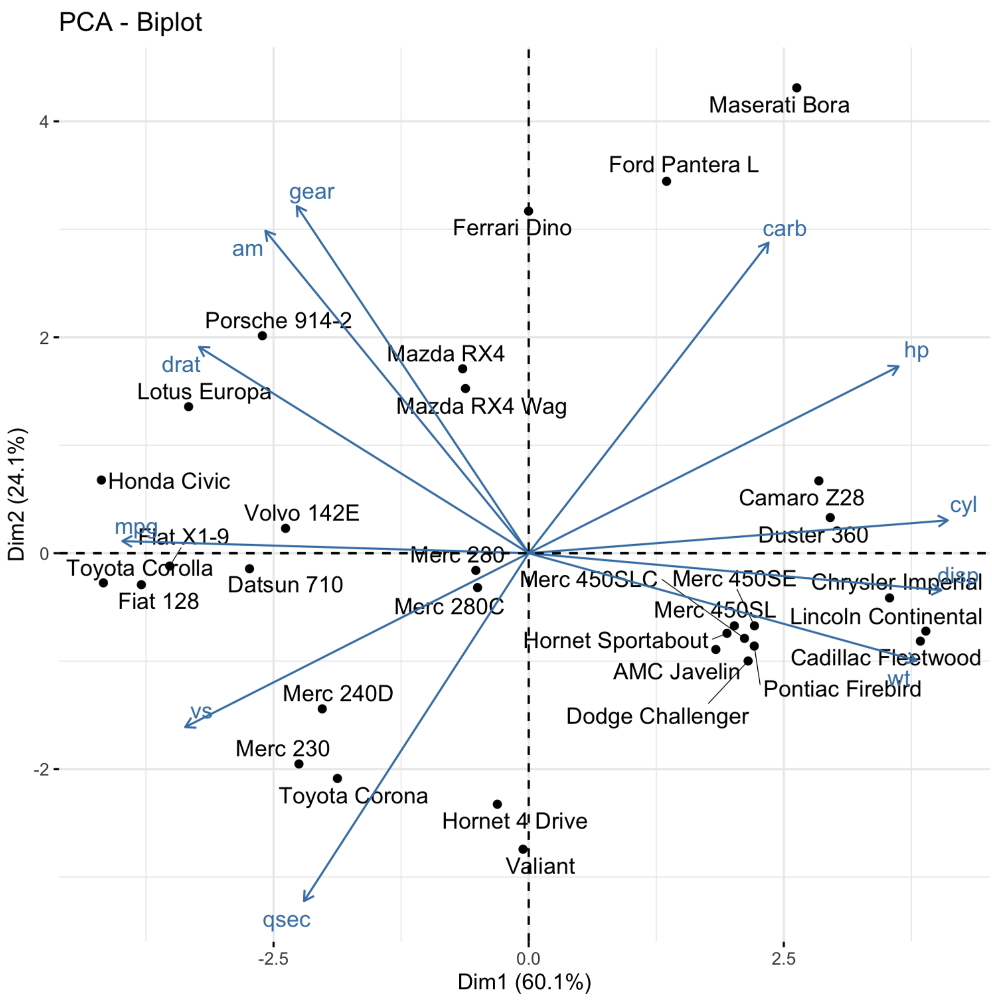
Percentage of variance explained

```
fviz_screeplot(pcaCars, addLabels = TRUE) +
  labs(y = "", title = "", x = "Principal Components")
```

Biplot

Recall that we can also produce a biplot to **project** every data point (a car's information in this case) onto the PC1-PC2 coordinate place, along with the *loading vectors* for each attribute

```
fviz_pca(pcaCars, geom = "point",
         geom.ind = c("point", "text"),
         repel = TRUE,
         label = c("var", "ind"))
```

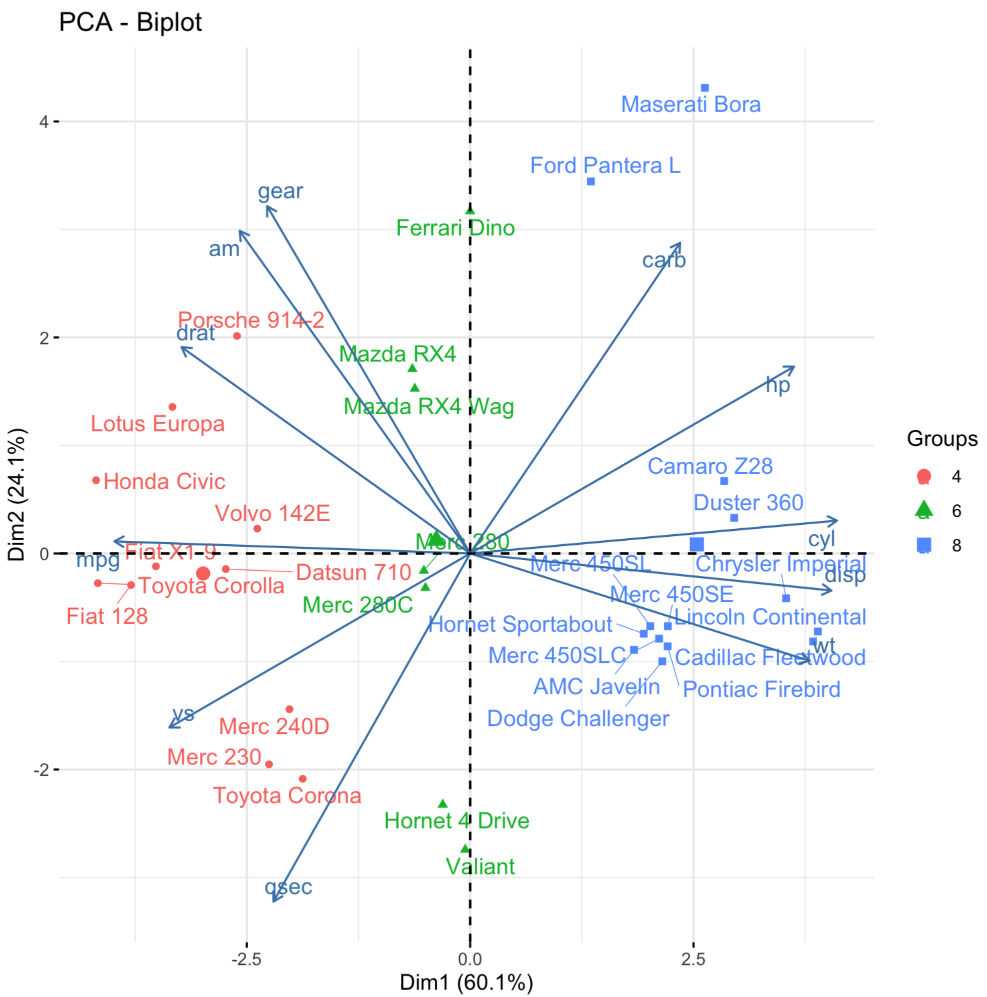


Biplot (2)

In this case, we can also color every data point based on the variable `cyl` (when understood as a category)

```
fviz_pca(pcaCars, geom = "point",
         geom.ind = c("point", "text"),
         repel = TRUE,
         label = c("var", "ind"),
         habillage = factor(mtcars$cyl))
```

Notice the clear separation (in this lower dimensional space) of the different type of cars based on the number of cylinders.



Contributions to first 2 PCs

```
fviz_contrib(pcaCars, choice = "var", axes =
```

```
fviz_contrib(pcaCars, choice = "ind", axes =
```