PCA with FactoMineR and factoextra

FactoMineR (for multivariate data analysis) and factoextra (for visualisation of PCA results)

Basics

PCA (Principal Component Analysis) is a dimension-reduction method. It finds principal factors - orthogonal linear combinations of original variables that explain maximum amount of variance.

$$W_{n \times q} = X_{n \times p} R_{p \times q}$$

The p dimensional input data **X** is projected into a q dimensional subspace by a linear transformation defined by **R**. New q dimensional data **W** has orthogonal variables. The transformation may be done through SVD decomposition or eigen value decomposition.

The Example

This example uses data about Hollywood action movies from 2015. Six quantitative variables with movie ratings scrapped from Rotten Tomato and Metacritic websites.

```
> head(movies2015)
                  Rotten              Rotten  Metacritic
               Tomatoes Metacritic Audience   Audience
Spectre              64         60       65         67
Furious 7            81         67       84         68
Terminator Genisys   25         38       59         63
San Andreas          50         43       56         55
Point Break           9         38       37         22
```

Use the **FactoMineR ::PCA()** function for PCA with supplementary quantitative and categorical variables. Missing values will be replaced by colMeans.

```
> library("FactoMineR")
> model <- PCA(movies2015)
> summary(model)
Eigenvalues
                Dim.1   Dim.2   Dim.3   Dim.4  Dim.5
Variance        4.474   0.355   0.131   0.040   0.00
% of var.      89.481   7.093   2.627   0.798   0.00
Cumulative % of var.  89.481  96.574  99.202 100.000 100.00

Individuals
                    Dist    Dim.1     ctr    cos2
Spectre         |  1.077 |  0.989   2.184   0.842 |
Furious 7       |  2.408 |  2.321  12.045   0.930 |
Terminator Genisys | 1.694 | −1.394   4.341   0.677 |
San Andreas     |  0.811 | −0.704   1.108   0.754 |
Point Break     |  3.643 | −3.461  26.767   0.902 |
Run All Night   |  1.192 |  0.842   1.584   0.499 |
No Escape       |  1.076 | −0.508   0.577   0.223 |
...

Variables
                    Dim.1    ctr    cos2    Dim.2
Rotten.Tomatoes   |  0.988 21.836  0.977 | −0.059
Metacritic        |  0.931 19.389  0.867 | −0.330
Average.critics   |  0.986 21.721  0.972 | −0.156
Rotten.Tomatoes.Audience | 0.943 19.885 0.890 | 0.135
Metacritic.Audience |  0.876 17.169  0.768 |  0.447
```

Scree plot

Use the **factoextra::get_eig()** function to extract information about eigenvalues. The **factoextra::fviz_screeplot()** function will plot the percentage of variance explained by each principal factor.

```
> get_eig(model)
         eigenvalue variance.percent cum.variance.percent
Dim.1  4.474039e+00       8.9480e+01                89.48
Dim.2  3.546706e-01       7.0934e+00                96.57
Dim.3  1.313722e-01       2.6273e+00                99.20
Dim.4  3.991824e-02       7.9836e-01               100.00
Dim.5  5.256294e-32       1.0512e-30               100.00
> fviz_screeplot(model)
```


Scree plot
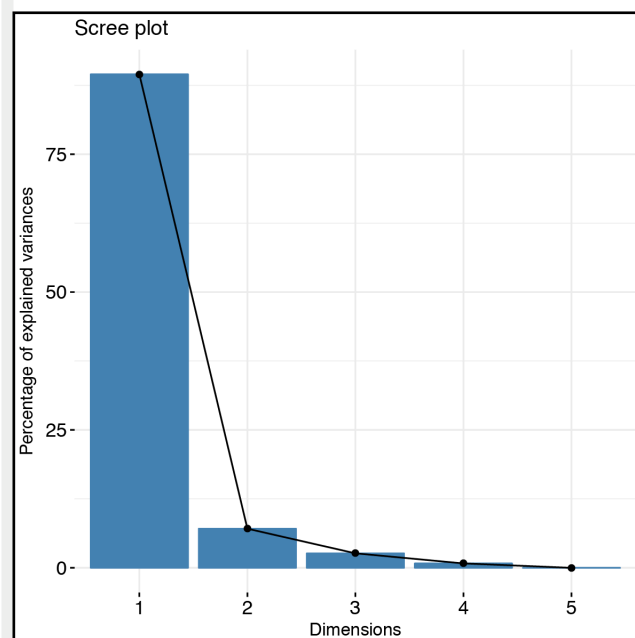
PCA variables' plot

Use the **factoextra::fviz_pca_var()** function to plot contribution of original variables into selected (the **axes** argument) principal components . Show variables through text labels or arrows (the **geom** argument). Result of this function is the **ggplot2** plot.
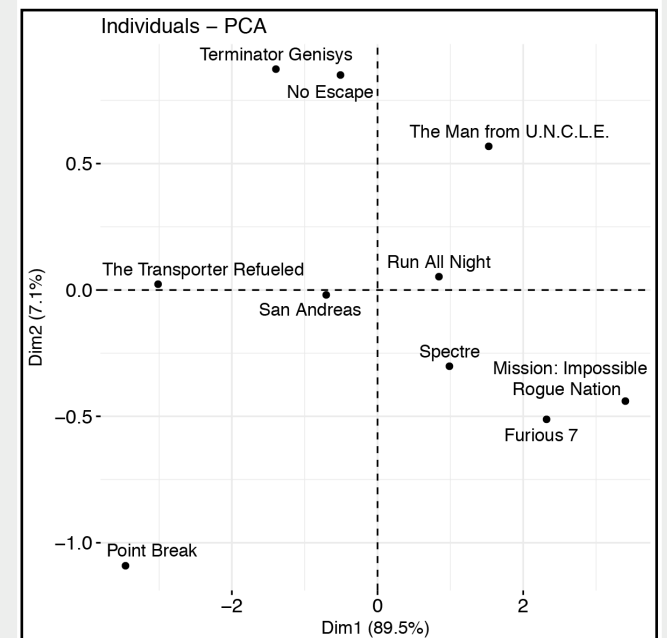
```
> fviz_pca_var(model)
```


Variables – PCA
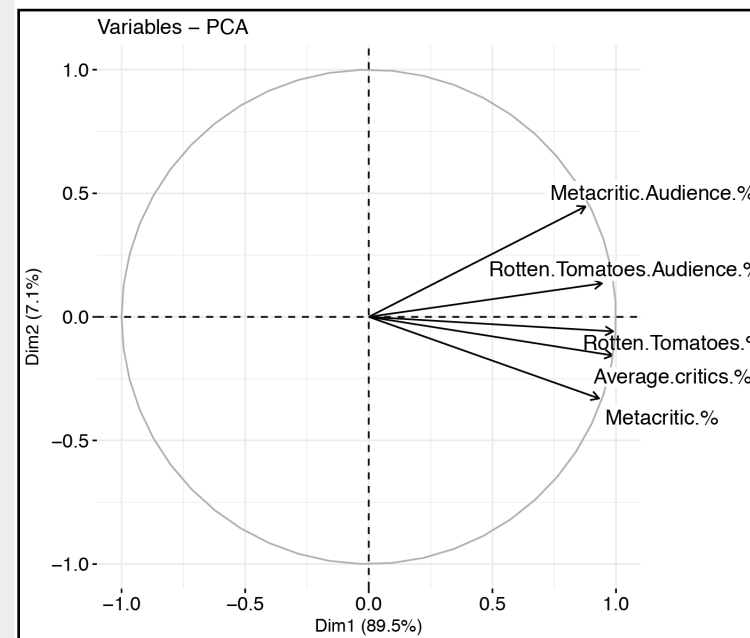
PCA individuals' plot

Use the **factoextra::fviz_pca_ind()** function to plot observations with selected (the **axes** argument) principal coordinates. With the **habillage** argument one can select a grouping variable which will be color-coded in the plot. Use **addEllipses** to plot ellipses for each group.

```
> fviz_pca_ind(model)
```


Individuals – PCA

PCA - Biplot

Use the **factoextra::fviz_pca_biplot()** function to combine results for individuals and variables into a single bi-plot.

With the **habillage** argument one can select a grouping variable which will be color-coded in the plot. Use **addEllipses** to plot ellipses for each group.

In the presented example, the first principal coordinate is highly correlated with average rating from all sources (audience and critics) while the second principal coordinate discriminate between audience and critics. Thus one can easily identify movies that are preferred by critics and these preferred by audience.

```
> fviz_pca_biplot(model, habillage = filmy2015$script.type) +
    theme(legend.position = "top")
```


PCA – Biplot