Principal Component Analysis Crib Sheet

Prcomp()

This is the vital function which creates the PCA object from the data. It takes as an input a data frame containing only numerical data. It also has an optional argument called `scale` which you should usually set to TRUE.

The output object has several useful categories within it:

`$x`

These are the values of the principal components for each data item

`$sdev`

These are the standard deviations of each principal component. We could use these to find the proportion of variance explained by each principal component.

`$rotation`

This shows the coefficients of each (possibly scaled) original variable in the principal component.

Predict(output.pca, predictor)

takes a predictor data frame and transforms it into the

You can use the functions `biplot()`, `plot()` and `summary()` on the output from `Prcomp()` but I prefer the look of the outputs using a library called `factoextra`. The rest of the functions are taken from there. The arguments for all these functions are the output from `Prcomp()`.

fviz_scree()

This produces a scree plot to determine how many principal components are needed. `summary()` also does this well.

fviz_pca_var()

This produces a "correlation circle" to show how much each principal component correlates with each original variable.

If the angles between arrows are small then the variables are positively correlated. If the angle is about 90 degrees then the variables are uncorrelated. If the angle is about 180 degrees then the variables are negatively correlated.

If the arrow is very short then the variable is not very well represented by the two principal components. If it has length 1 (i.e. is on the circle) then the variable is completely represented by the two principal components.

fviz_pca_biplot()

Create a biplot showing both the transformed data and the correlation circle.