Principal Component Analysis (PCA)

# Introduction to PCA

* Also known as general factor analysis
* Is an unsupervised learning algorithm that helps examine the relationship between a set of variables.
* It helps determine multiple lines of best fit that are perpendicular (orthogonal or at right angle) to each other unlike regression which finds only one line of best fit.
* The number of possible dimensions is equal to the number of variables.
* Components are a linear transformation that chooses the greatest variance lies on the first axis, the second variance lies on the second axis.
* The lines are uncorrelated as the lines are orthogonal to each other.
* It helps to compress the explained variable of a large data set to just a few variables.
* The data is used for analysis and requires scaling the data.
* It essentially helps spot the variables that explain the greatest variance.

# PCA with Python

* Import the data analysis and visualization libraries
* Import data and clean up
* From sklearn.preprocessing import StandardScaler
* Instantiate the Standard Scaler
* import PCA from sklearn.decomposition
* instantiate PCA with n\_components expected e.g., pca = PCA(n\_components = 3)
* fit and transform the data to pca and set it to a variable
* confirm that the data has been transformed by checking the shape of the new variable
* plot the new transformed data.
* Interpreting the components is difficult as the components don’t correspond with any attributes directly but are combination of a number of components. This can be verified by checking pca.components\_
* This can further be confirmed by visualizing with a heatmap as follows:
* Set the pca.components\_ into a data frame with the feature names as the column titles e.g., df\_pca = pd.DataFrame(pca.components\_, columns=data[‘Featurenames’])
* Plot the heatmap of the data frame.