# Principal Component Analysis

## What is PCA ?

- As the dimensions of data increases, the difficulty to visualize it and perform computations on it also increases. So, how to reduce the dimensions of a data-
  * Remove the redundant dimensions
  * Only keep the most important dimensions
- Principal component analysis is a dimensionality reduction technique for feature extraction.
- Want to learn more about type dimensionality reduction and feature extraction look at our previous posts.
- PCA combines our input variables in a specific way, then we can drop the “least important” variables while still retaining the most valuable parts of all of the variables.
- As an added benefit, each of the “new” variables after PCA are all independent of one another. Because of this there will be no correlation. This is mostly used for Linear models, because linear model assumes that all the variables are independent.

# Principal Component Analysis

## What is PCA ?

- **PCA finds a new set of dimensions (or a set of basis of views) such that all the dimensions are orthogonal (and hence linearly independent) and ranked according to the variance of data along them. It means more important principle axis occurs first. (more important = more variance/more spread out data)**

# Principal Component Analysis

## When to use PCA?

- **Do you want to reduce the number of variables, but aren't able to identify variables to completely remove from consideration?**
- **Do you want to ensure your variables are independent of one another?**
- **Are you comfortable making your independent variables less interpretable?**

**If you answered "yes" to all three questions, then PCA is a good method to use. If you answered "no" to question 3, you should not use PCA.**

# Principal Component Analysis

## How does PCA work?

**The main goal of PCA is to reduce the dimensions of data, Where the reduced data will represent the original dataset. Because we should not lose the important variables and it also keeps the variables as independent without any correlation. we can do PCA in two ways**

**1. Maximizing the variance**

**2. minimizing distance**

- **The first step is to normalize the data and decide whether to standardize the data or not. This is a very important step in PCA because if we use data(features here) of different scales, we get misleading components.**
- **Let the matrix be "Z". Take the matrix Z, transpose it, and multiply the transposed matrix by Z. The resulting matrix is the covariance matrix of Z, up to a constant.**

# Principal Component Analysis

## How does PCA work?

- Calculate the eigenvectors and their corresponding eigenvalues of $Z^TZ$. This is quite easily done in most computing packages— in fact, the eigendecomposition of $Z^TZ$ is where we decompose $Z^TZ$ into $PDP^{-1}$,where P is the matrix of eigenvectors and D is the diagonal matrix with eigenvalues on the diagonal and values of zero everywhere else. The eigenvalues on the diagonal of D will be associated with the corresponding column in P i.e the first element of D is $\lambda_1$ and the corresponding eigenvector is the first column of P. This holds for all elements in D and their corresponding eigenvectors in P.
- Take the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p$ and sort them from largest to smallest. In doing so, sort the eigenvectors in P accordingly. Depending on the computing package, this may be done automatically. Call this sorted matrix of eigenvectors P*.

# Principal Component Analysis

## How does PCA work?

- Calculate Z* = ZP*. This new matrix, Z*, is a centred/ standardized version of original data but now each observation is a combination of the original variables, where the weights are determined by the eigenvector. As a bonus, because our eigenvectors in P* are independent of one another, each column of Z* is also independent of one another.
- The principal components are perpendicular to one another. In fact, every principal component will ALWAYS be orthogonal to every other principal component.
- Because our principal components are orthogonal to one another, they are statistically independent of one another. which is why our columns of Z* are independent of one another!

# Principal Component Analysis

## How to determine the no of features we want?

**There are 3 major methods to do that**

- **Method 1: We select how many dimensions we want to keep. If I want to visually represent things in two dimensions, so I may only keep two features. This is use-case dependent and there isn't a hard-and-fast rule for how many features I should pick.**
- **Method 2: Calculate the proportion of variance explained for each feature, pick a threshold, and add features until you hit that threshold.**
- **Method3: sort features by the proportion of variance explained and plot the cumulative proportion of variance explained as you keep more features. One can pick how many features to include by identifying the point where adding a new feature has a significant drop in variance explained relative to the previous feature, and choosing features up until that point. you can use elbow mothod.**

# Principal Component Analysis

## What do we get from PCA?

- A measure of how each variable is associated with one another. (Covariance matrix.)
- The directions in which our data are dispersed. (Eigenvectors.)
- The relative importance of these different directions. (Eigenvalues.)
- PCA combines our predictors and allows us to drop the eigenvectors that are relatively unimportant.

# Principal Component Analysis

## Limitations of PCA

- PCA is a linear algorithm. It will not be able to interpret complex polynomial relationship between features. On the other hand, t-SNE is based on probability distributions with random walk on neighborhood graphs to find the structure within the data.
- A major problem with, linear dimensionality reduction algorithms is that they concentrate on placing dissimilar data points far apart in a lower dimension representation. But in order to represent high dimension data on low dimension, non-linear manifold, it is important that similar datapoints must be represented close together, which is not what linear dimensionality reduction algorithms do.
- Local approaches seek to map nearby points on the manifold to nearby points in the low-dimensional representation. Global approaches on the other hand attempt to preserve geometry at all scales, i.e mapping nearby points to nearby points and far away points to far away points
- It is important to know that most of the nonlinear techniques other than t-SNE are not capable of retaining both the local and global structure of the data at the same time.