**PCA: Principal Component Analysis — How to Get Superior Results with Fewer Dimensions?**

While PCA is often referred to as a dimensionality reduction technique, it is actually a data transformation.

Nevertheless, PCA makes it very easy to use the resulting principal components to reduce the number of dimensions as it ranks them from “most useful” (captures a lot of the data variance) to “least useful” (captures very little of the data variance).

Hence, there is really no harm in putting it under the dimensionality reduction group of algorithms within the unsupervised learning branch of ML.

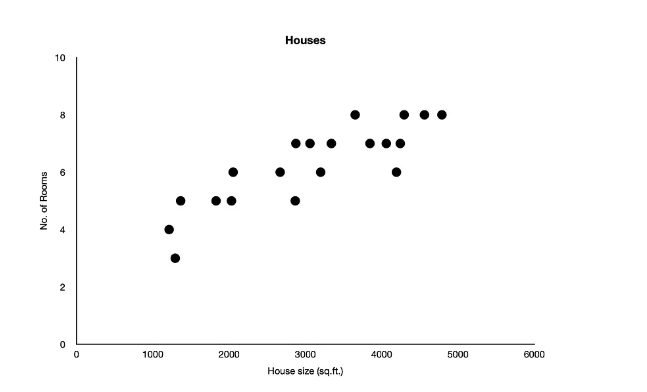
# How does Principal Component Analysis (PCA) work?

## Preparing the data

In simple terms, PCA helps us find new axes (principal components) of our dimensions that can better capture the variance of the data.

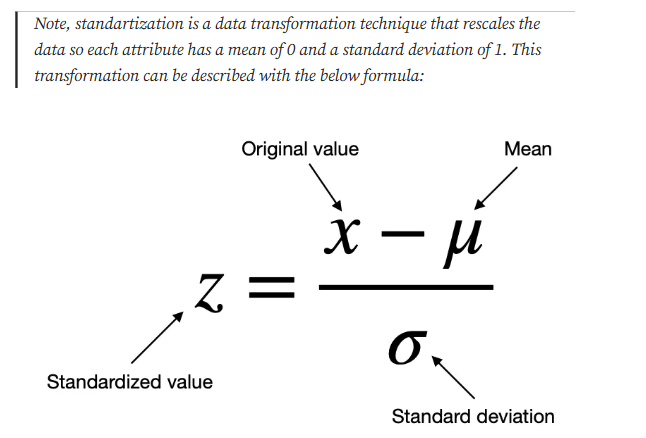
For example, you may have two attributes — house size (sq.ft.) and the number of rooms in that house. Not surprisingly, these two attributes are highly correlated as bigger houses tend to have more rooms.

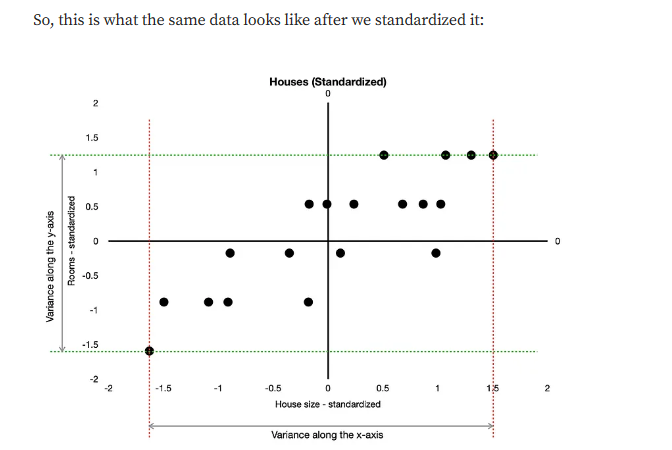
Let’s visualize our made-up example by plotting the two attributes on a 2D scatterplot:



We can see that there is a positive correlation between house size and the number of rooms. Also, there seems to be more variance in house sizes when compared to the number of rooms.

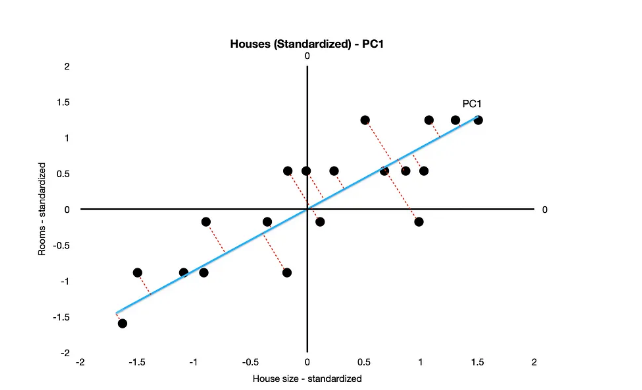
However, before we make any premature conclusions, let’s pause for a minute and take a look at the scale of x and y. We can see that these two attributes use two different scales; hence the above picture is not really representative.





We can see that the data is now centered around the origin because both standardized attributes have a mean of 0. At the same time, it is now easier to visually compare the spread (variance) of the two attributes. As previously speculated, there is indeed a wider spread in house sizes than the number of rooms.

Now that we have the data standardized and centered around the origin, we can look for the “best-fit” line. This line must go through the origin, and we can find it by minimizing the distances from data points to their projections on the line. It looks like this:



Interestingly, this “best-fit” line also happens to be the axis for Principal Component 1 (PC1). This is because minimizing the distances from the line also **maximizes the spread of data point projections on that same line**.

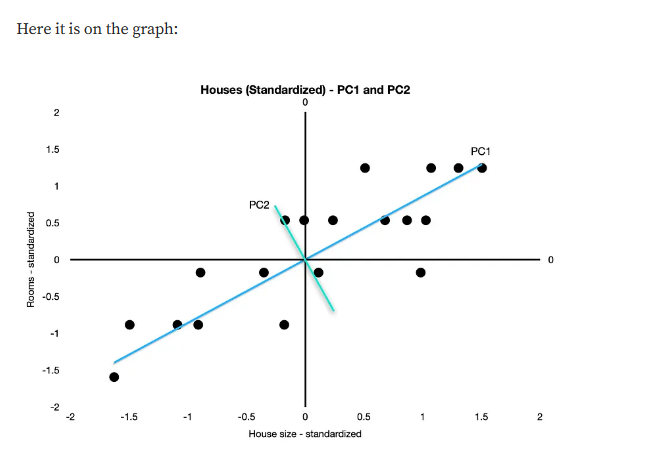
**In other words, we have found a new axis that captures the maximum amount of variance of the data in that dimension.**

If you decide to study the mathematics of this process separately, you will come across terms like **eigenvectors** and **eigenvalues**. They, in a sense, describe this “best-fit” line and are mathematically found through doing **eigendecomposition** of the covariance matrixor through **Singular Value Decomposition** analysis (SVD for short).

## **Finding principal components — PC2 and others**

Remember, Principal Component Analysis is not just about dimensionality reduction. Hence, we can find as many principal components as there are attributes (dimensions) in our data. Said that, let’s find PC2.

Since we only have two attributes, in this example, it is straightforward to find PC2 once we know PC1. It is simply an orthogonal (90 degrees angle) line to PC1 that goes through the origin.



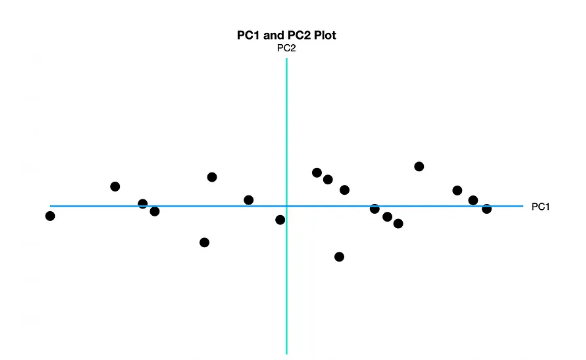
Note, if we had three attributes, then PC2 would be the line that goes through the origin, is orthogonal to PC1, and minimizes distances from data points to their projections on the PC2 line. Then PC3 would be the line that goes through the origin and is orthogonal to both PC1 and PC2.

## **Summarizing principal components**

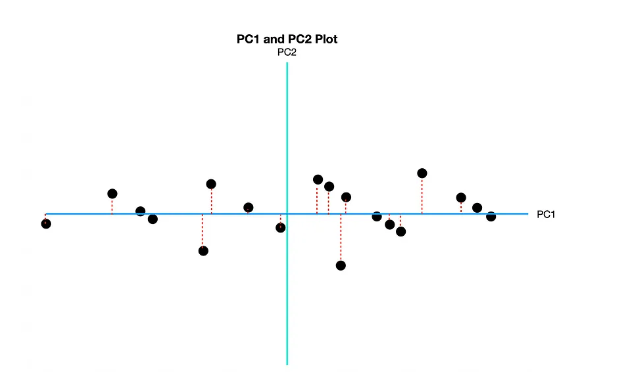
If we were to list each principal component, PC1 would be the dimension that captures the highest proportion of the data variance, with PC2 being the dimension that captures the highest proportion of the **remaining** variance that PC1 could not capture. Similarly, PC3 would be the dimension capturing the highest proportion of the remaining variance that PC1 and PC2 could not capture, etc.

The goal of PCA is to transform the data in a way that enables us to capture the maximum amount of variance in each subsequent dimension.

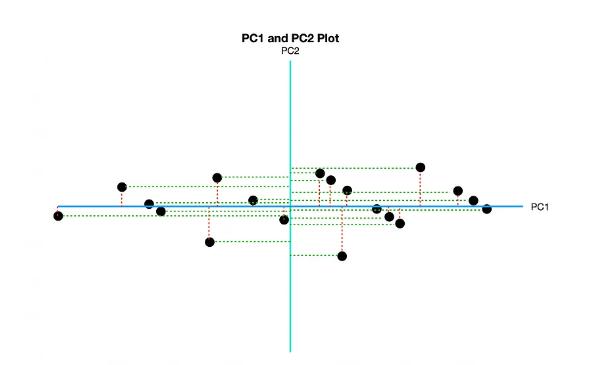
Now that we have found PC1 and PC2 in our two-dimensional example, we can rotate the graph and make PC1 and PC2 our new axes:



Also, if we wish, we can perform dimensionality reduction by projecting points onto PC1 and getting rid of PC2:



Finally, if, for whatever reason, we did not like PC1, we could reduce dimensions by projecting points onto PC2 instead, enabling us to get rid of PC1. **However, we would lose a lot of the data variance if we did that.**



Essentially, in PCA we make a transition from one variable space to another, with the new space containing fewer variables (n\_component), where the new variables are **uncorrelated** and are the weighted sum of the old variables.  
As a result we get m variables: {PC1, PC2, PC3... PCm} , where PC1 will receive the most information(maximum sample variance), PC2 - less, and so on (A variable is considered informative if it has a high sample variance).

**Note**: PCA is a linear transformation, and linearity is sensitive to the scale of data. Therefore, PCA works best when all data values are on the same scale. This can be done by subtracting the column mean from its values and dividing the result by its standard deviation. That is called data standardization. Prior to using PCA, make sure the data is scaled!

After getting the covariance matrix, PCA tries to find a linear combination of features that best explains it - it fits linear models until it identifies the one that explains the ***maximum*** **amount of variance**.

One important detail to take into consideration here is that, due to its linear nature, PCA will concentrate most of the explained variance in the first principal components. So, when looking at the explained variance, usually our first two components will suffice.

In a nutshell, this is what PCA is all about: Finding the directions of maximum variance in high-dimensional data and project it onto a smaller dimensional subspace while retaining most of the information. So main advantages of PCA are data compression (reduce memory, speed up learning) and visualization.

As humans, we can only visualize things in 2-dimensions or 3-dimensions. For data, this rule does not apply! Data can have an infinite amount of dimensions, but this is where the curse of dimensionalitycomes into play.

For example let’s say that we have two variables height and weight. To represent these 2 lines, PCA combines both height and weight to create two brand new variables. It could be 30% height and 70% weight, or 87.2% height and 13.8% weight, or any other combinations depending on the data that we have. **PCA creates new variables with the original features trying to include the majority of the variance into the principal components.**

**A rule of thumb here is that the cumulative variance explained by the components should be at least 70%.**

PCA creates new features trying to collect the major variance as possible within the data given in order to not lose the main information of the data.

Since PCA is a linear method, it has a main disadvantage in order to make it works properly.

**PCA needs features, variables to be highly correlated, before doing the reduction.**

* However there is a solution to solve this disadvantage and are the algorithms called **Non-linear methods (Manifold learning). That I will post later.**

Notice how the steps in principal component analysis such as computing the covariance matrix, performing eigendecomposition or singular value decomposition on the covariance matrix to get the principal components have all been abstracted away when we use scikit-learn’s implementation of PCA.

PCA takes advantage of existing correlations between the input variables in the dataset and combines those correlated variables into a new set of uncorrelated variables.

**PCA is an unsupervised machine learning algorithm as it does not require labels in the data.**

Important considerations:

* Each new variable is a linear combination of the original variables.
* The first new variable is chosen to capture the maximum amount of variance in the data set.
* The second will account for the maximum variance not captured by the first.
* The pth new variable will account for the maximum variance in the data not captured by the p−1 new variables.
* **The new variables are not correlated.**

Choosing the ideal number of components:

One alternative is to conduct a PCA for all or an arbitrarily large number of components, then visualize the percentage of explained variance or the eigenvalues per component using a scree plot. Based on the graph, the optimal number can be selected.

A Scree Plot is a simple line segment plot that shows the eigenvalues for each individual PC. It shows the eigenvalues on the y-axis and the number of factors on the x-axis. It always displays a downward curve. Most scree plots look broadly similar in shape, starting high on the left, falling rather quickly, and then flattening out at some point. This is because the first component usually explains much of the variability, the next few components explain a moderate amount, and the latter components only explain a small fraction of the overall variability. The scree plot criterion looks for the “elbow” in the curve and selects all components just before the line flattens out.

