Source: TDS [[LINK](https://towardsdatascience.com/exploratory-data-analysis-86eb12060eac)]

Exploratory Data Analysis

# Complex Relationships

* Sometimes you need to compare two or more variables in unison to spot outliers.
  + You would plot one variable in x-axis, and the other in y-axis, and see what “sticks out”.
  + For a third variable, there are options to plot a z-variable.
  + However, what if you wanted to compare more than 3 variables?
* There are several techniques to reduce the dimensionality of your data and visualize patterns.
  + One common technique is **PCA (Principal Components Analysis)**

# Data Visualizations

* Scatter plots:
  + Relationship btwn two variables
  + Color:
    - Color the top n categories and lump the rest into ‘Other’.
  + size and shape
* Bar plots:
  + discrete values, e.g., count of categories.
* Histograms:
  + summarizing distributions
* Line plots:
  + continuous sequence of values

Why, How and When to Scale your Features

Source: [[LINK](https://medium.com/greyatom/why-how-and-when-to-scale-your-features-4b30ab09db5e)]

# Methods to perform Feature Scaling

1. Standardization:

xi = x – mean(x)

standard deviation

Replaces the values by their Z-scores  
The feature will have mean = 0, and standard deviation = 1

1. Mean Normalization:

xi = x – mean(x)

max(x) – min(x)

The distribution will have values between -1 and 1 with mean = 0

NOTE: **Standardization** and **Mean Normalization** can be used for algorithms that assumes zero centric data like Principal Component Analysis (PCA)

1. Min-Max Scaling:

xi = x – min(x)

max(x) – min(x)

This scaling brings the value between 0 and 1

1. Unit Vector:

xi = x .

||x||

scaling is done considering the whole feature vector to be of unit length

NOTE: **Min-Max Scaling** and **Unit Vector** techniques produces values of rage [0,1].   
When dealing with features with hard boundaries this is quite useful.  
For e.g., when dealing with image data, the color can range form only 0 to 255.

# When to Scale

* **Rule of thumb: Any algorithm that computes distance or assume normality, scale your features!!!**
* Examples of where feature scaling matters and does not:
  + k-nearest neighbor
  + PCA
  + Can speed up gradient descent by scaling
  + **Tree based models** are not distance based and can handle varying ranges of features. Hence, Scaling is **NOT** required
  + Linear Discriminant Analysis (LDA), Naïve Bayes are by design equipped to handle this, and gives weights to features accordingly.  
    Scaling may **NOT** have much effect.