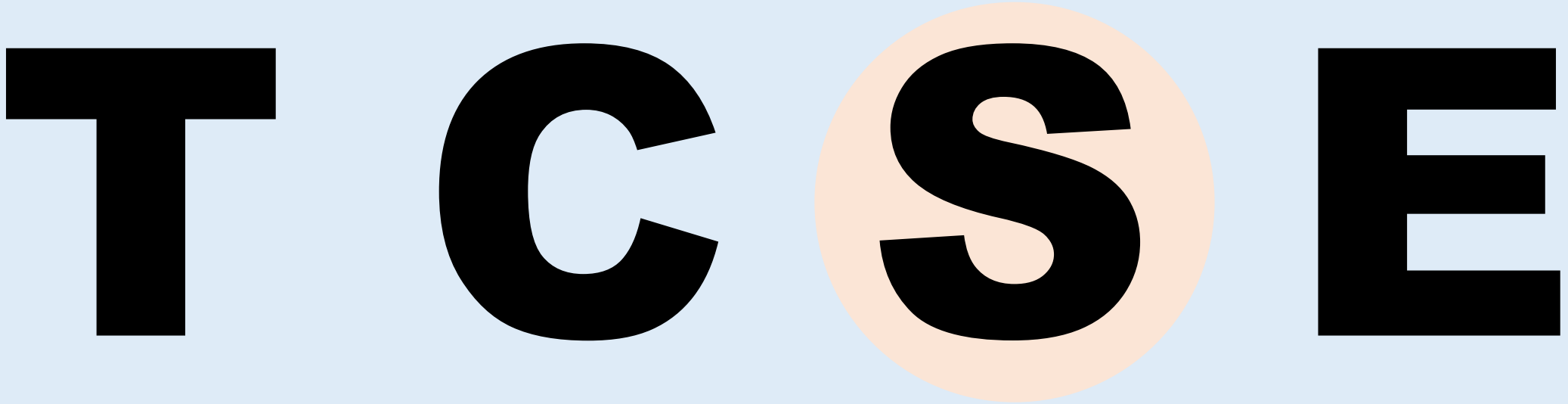
Feature Engineering

Feature Engineering



Transformation

Construction

Scaling

Extraction

Feature Engineering

- Scaling
  - Normalisation
  - Standardisation
  - Robust Scaler

  This Scales the data
- Encoding
  - Ordinal Encoding
  - OneHot Encoding

  This Encodes the data
- Mathematical Transformations
  - Log Transform
  - Reciprocal Transform
  - Square/Square Root Transform
  - Power Transform

  This Normalises the data

What is Feature Scaling?

Feature Scaling is a technique to standardise the independent feature present in the data

Types of Feature Scaling

- Normalization (With MinMaxScaler)
- Standardization (StandardScaler)
- Robust Scaling (RobustScaler)

Normalization

$$x_{normalised} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

The goal of normalisation is to change the values of numeric columns in the data set to a common scale without distorting the data or losing information

The data transformed in a 0 to 1 scale

Standardisation / Z Score Normalisation

$$x_{standardised} = \frac{x - x_{mean}}{standard\ deviation}$$

The new data formed will have its mean equal to zero and standard deviation equal to one

When to apply Scaling

- K-Means
- KNN
- PCA
- ANN
- Gradient Descent

Few Points to Remember

- Normalization is good to use when you know that the distribution of your data does not follow a Gaussian distribution. This can be useful in algorithms that do not assume any distribution of the data like K-Nearest Neighbors and Neural Networks.

- Standardization, on the other hand, can be helpful in cases where the data follows a Gaussian distribution. However, this does not have to be necessarily true. Also, unlike normalization, standardization does not have a bounding range. So, even if you have outliers in your data, they will not be affected by standardization.

- There is no any thumb rule to tell you when to normalize or standardize your data. You can always start by fitting your model to raw, normalized and standardized data and compare the performance for best results.

Robust Scaler

$$x_{scaled} = \frac{x_i - x_{median}}{IQR}$$

$$IQR = Q_3 - Q_1$$

It is called Robust because it is Robust to outliers

Feature Engineering

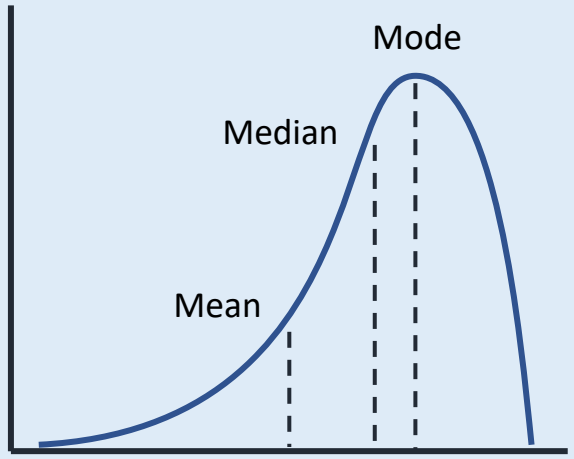
T C S E

Transformation Construction Scaling Extraction

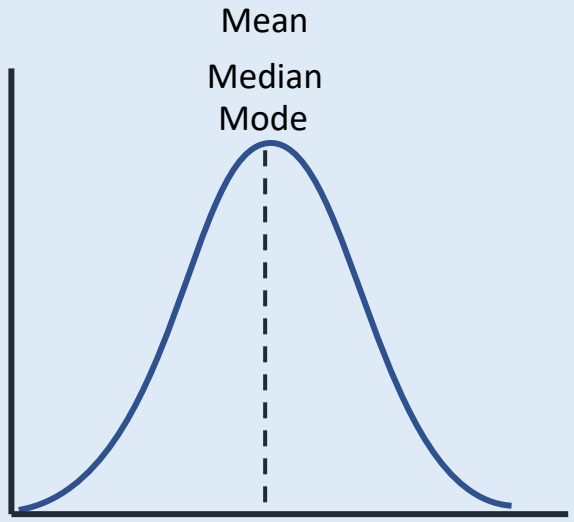
Mathematical Transformation

- These Transformations are used when the data is not normally distributed
- How to check if the data is normally distributed or not?
  - Use sns.distplot
  - Used pandas skew method
  - QQ Plots
  - Hypothesis Testing

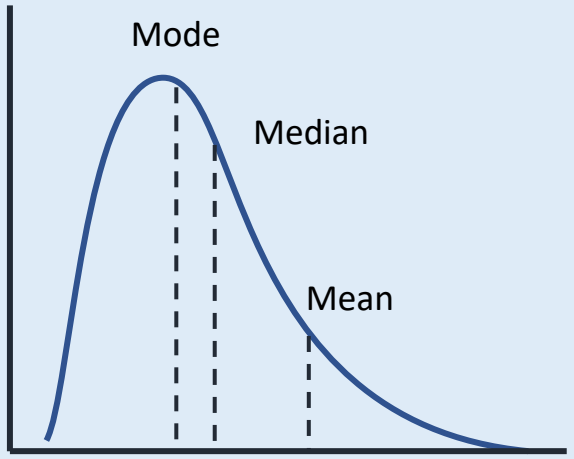
What is data skewness?



Left Skewed / Negatively Skewed



Normally Distributed

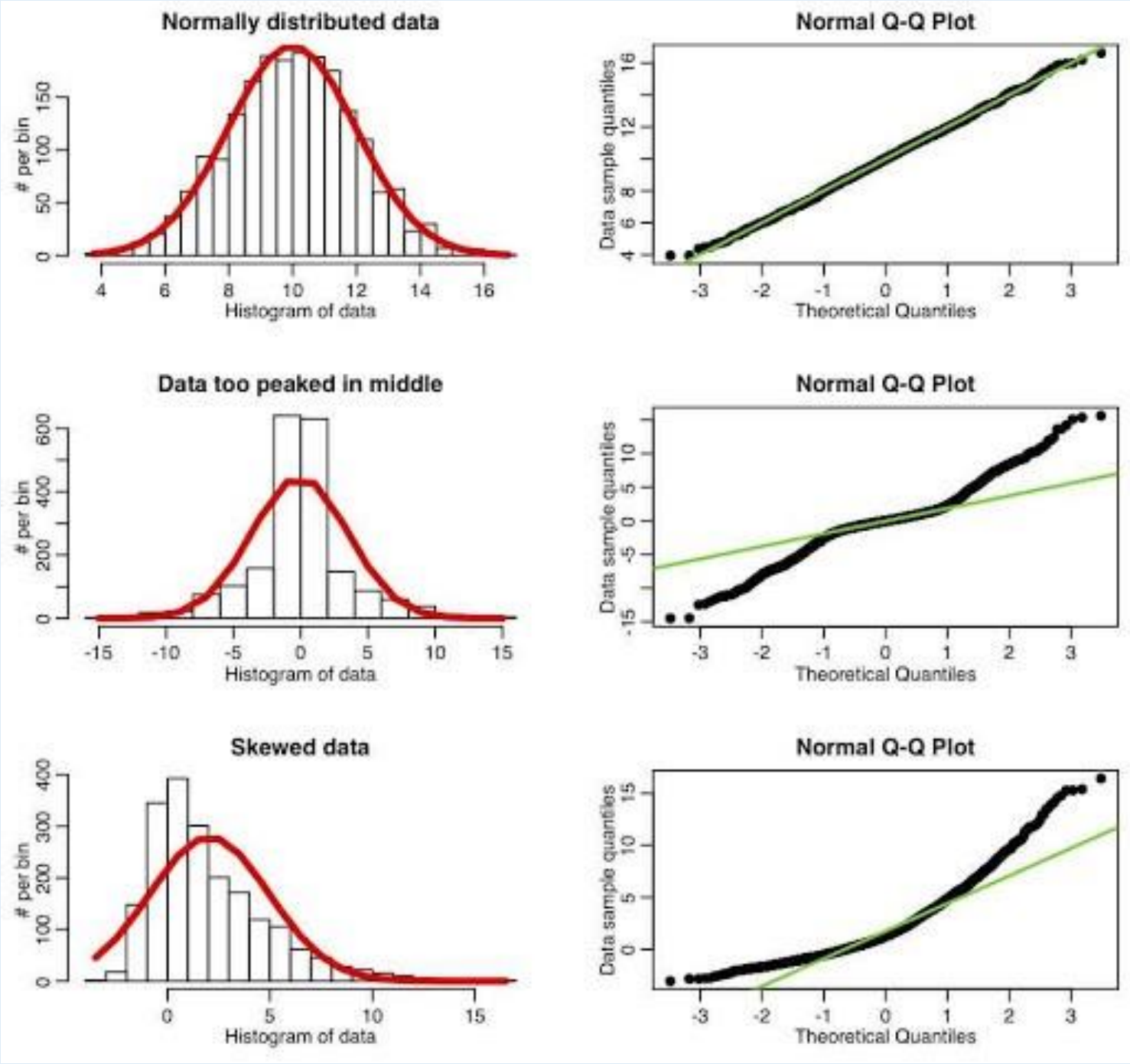


Right Skewed / Positively Skewed

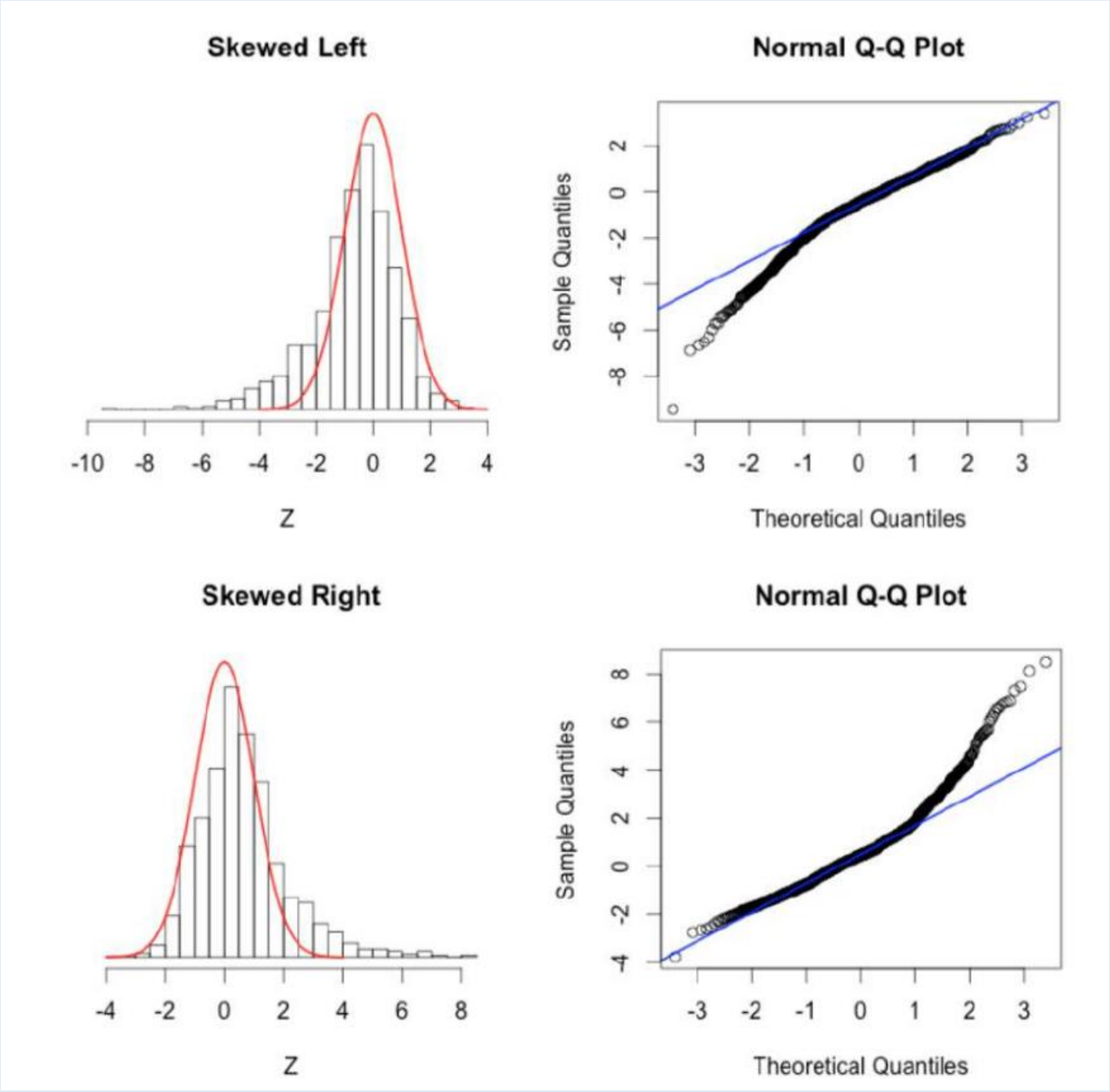
After Running pd.skew():

- If value is $-0.5\ to\ +0.5$ the data is fairly normally distributed
- If the value is $-1\ to\ -0.5$ the data is negatively skewed
- If the value is < -1 the data is highly negatively skewed
- If the value is $+0.5\ to\ 1.0$ the data is positively skewed distributed
- If the value $>\ +1$ the data is highly positively skewed

Data Distribution and QQ Plots



Data Distribution and QQ Plots



Hypothesis Testing

The statistical test conducted to check the normality of data are:

- Shapiro Test
- KS Test
- Normal Test

All these test assume the following null and alternate hypothesis as:

Null Hypothesis (H0): Data is normally distributed

Alternate Hypothesis (H1): Data is not normally distributed

We reject the null hypothesis if the P-Value is less than 0.05 and if it is greater than 0.05 technically we say that we do not have sufficient evidence to reject null hypothesis (that tantamount to rejecting the alternate and accepting the null hypothesis)

Which Mathematical Transformation to use?

- Log Transform if the data is right skewed
- Square Transform if the data is left skewed
- Reciprocal Transform (Should be experimented)
- Square root transform (Should be experimented)
- Cube Root Transform (Should be experimented)