Box Plot

Box Plot is a graphical method to visualize data distribution for gaining insights and making informed decisions. Box plot is a type of chart that depicts a group of numerical data through their quartiles.

In this article, we are going to discuss components of a box plot, how to create a box plot, uses of a Box Plot, and how to compare box plots.

Table of Content

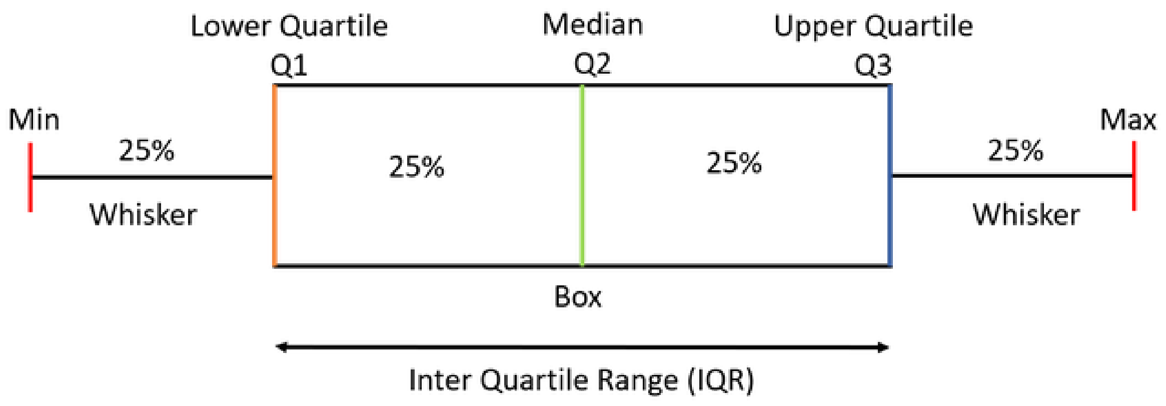
What is a Box Plot?

The idea of box plot was presented by John Tukey in 1970. He wrote about it in his book "Exploratory Data Analysis" in 1977. Box plot is also known as a whisker plot, box-and-whisker plot, or simply a box-and whisker diagram. Box plot is a graphical representation of the distribution of a dataset. It displays key summary statistics such as the median, quartiles, and potential outliers in a concise and visual manner. By using Box plot you can provide a summary of the distribution, identify potential and compare different datasets in a compact and visual manner.

Elements of Box Plot

A box plot gives a five-number summary of a set of data which is-

- Minimum – It is the minimum value in the dataset excluding the outliers.
- First Quartile (Q1) – 25% of the data lies below the First (lower) Quartile.
- Median (Q2) – It is the mid-point of the dataset. Half of the values lie below it and half above.
- Third Quartile (Q3) – 75% of the data lies below the Third (Upper) Quartile.
- Maximum – It is the maximum value in the dataset excluding the outliers.



Note: The box plot shown in the above diagram is a perfect plot with no skewness. The plots can have skewness and the median might not be at the center of the box.

The area inside the box (50% of the data) is known as the Inter Quartile Range. The IQR is calculated as –

IQR = Q3-Q1

Outlies are the data points below and above the lower and upper limit. The lower and upper limit is calculated as –

Lower Limit = Q1 - 1.5*IQR

Upper Limit = Q3 + 1.5*IQR

The values below and above these limits are considered outliers and the minimum and maximum values are calculated from the points which lie under the lower and upper limit.

How to create a box plots?

Let us take a sample data to understand how to create a box plot.

Here are the runs scored by a cricket team in a league of 12 matches – *100, 120, 110, 150, 110, 140, 130, 170, 120, 220, 140, 110.*

To draw a box plot for the given data first we need to arrange the data in ascending order and then find the minimum, first quartile, median, third quartile and the maximum.

Ascending Order

100, 110, 110, 110, 120, 120, 130, 140, 140, 150, 170, 220

Median (Q2) = (120+130)/2 = 125; Since there were even values

To find the First Quartile we take the first six values and find their median.

Q1 = (110+110)/2 = 110

For the Third Quartile, we take the next six and find their median.

Q3 = (140+150)/2 = 145

Note: If the total number of values is odd then we exclude the Median while calculating Q1 and Q3. Here since there were two central values we included them. Now, we need to calculate the Inter Quartile Range.

IQR = Q3-Q1 = 145-110 = 35

We can now calculate the Upper and Lower Limits to find the minimum and maximum values and also the outliers if any.

Lower Limit = Q1-1.5*IQR = 110-1.5*35 = 57.5

Upper Limit = Q3+1.5*IQR = 145+1.5*35 = 197.5

So, the minimum and maximum between the range [57.5,197.5] for our given data are –

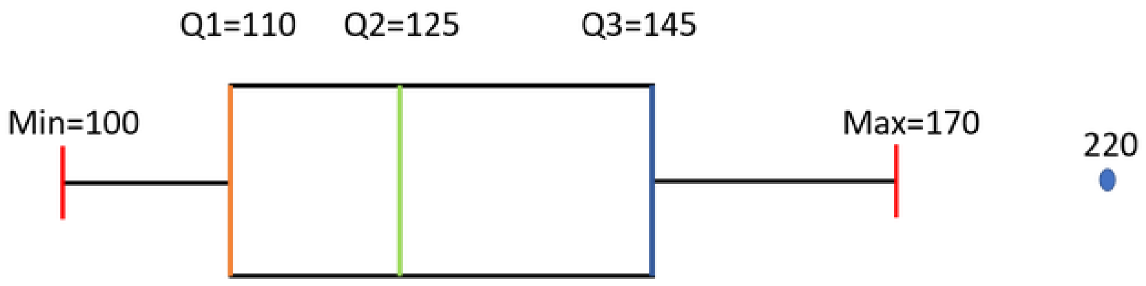
Minimum = 100

Maximum = 170

The outliers which are outside this range are –

Outliers = 220

Now we have all the information, so we can draw the box plot which is as below-



We can see from the diagram that the Median is not exactly at the center of the box and one whisker is longer than the other. We also have one Outlier.

Use-Cases of Box Plot

- Box plots provide a visual summary of the data with which we can quickly identify the average value of the data, how dispersed the data is, whether the data is skewed or not (skewness).
- The Median gives you the average value of the data.
- Box Plots shows Skewness of the data-

a) If the Median is at the center of the Box and the whiskers are almost the

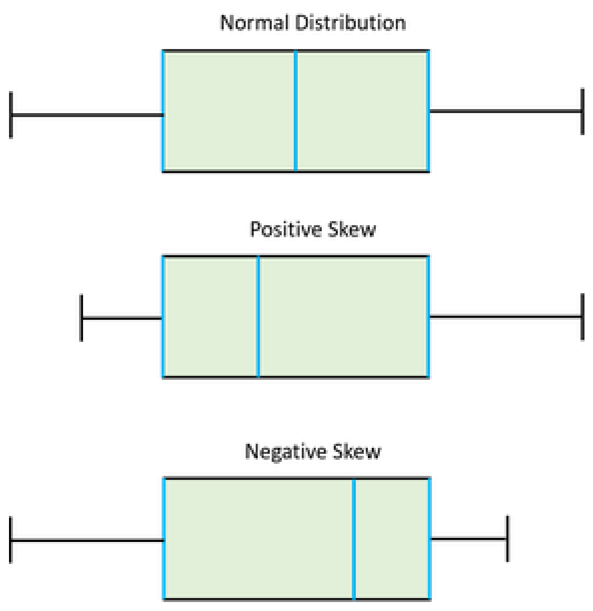
same on both the ends then the data is Normally Distributed.

b) If the Median lies closer to the First Quartile and if the whisker at the lower

end is shorter (as in the above example) then it has a Positive Skew (Right Skew).

c) If the Median lies closer to the Third Quartile and if the whisker at the

upper end is shorter than it has a Negative Skew (Left Skew).

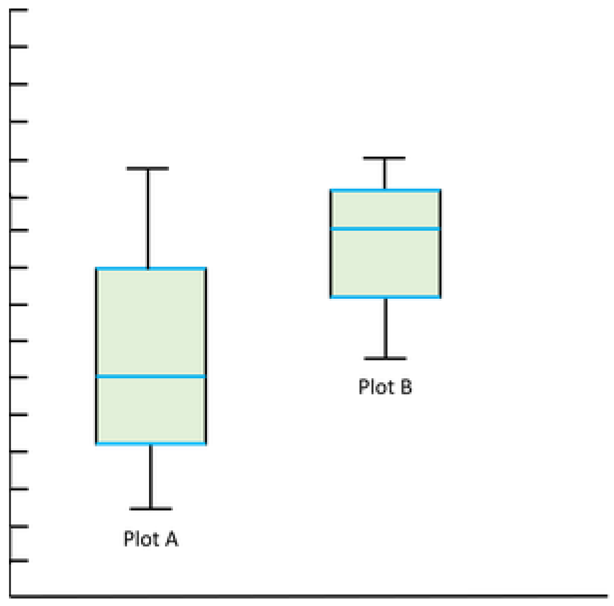


- The dispersion or spread of data can be visualized by the minimum and maximum values which are found at the end of the whiskers.
- The Box plot gives us the idea of about the Outliers which are the points which are numerically distant from the rest of the data.

How to compare box plots?

As we have discussed at the beginning of the article that box plots make comparing characteristics of data between categories very easy. Let us have a look at how we can compare different box plots and derive statistical conclusions from them.

Let us take the below two plots as an example: –



- Compare the Medians — If the median line of a box plot lies outside the box of the other box plot with which it is being compared, then we can say that there is likely to be a difference between the two groups. Here the Median line of the plot B lies outside the box of Plot A.

- Compare the Dispersion or Spread of data — The Inter Quartile range (length of the box) gives us an idea about how dispersed the data is. Here Plot A has a longer length than Plot B which means that the dispersion of data is more in plot A as compared to plot B. The length of whiskers also gives an idea of the overall spread of data. The extreme values (minimum &maximum) give the range of data distribution. Larger the range more scattered the data. Here Plot A has a larger range than Plot B.
- Comparing Outliers — The outliers give the idea of unusual data values which are distant from the rest of the data. More number of Outliers means the prediction will be more uncertain. We can be more confident while predicting the values for a plot which has less or no outliers.
- Compare Skewness — Skewness gives us the direction and the magnitude of the lack of symmetry. We have discussed above how to identify skewness. Here Plot A is Positive or Right Skewed and Plot B is Negative or Left Skewed.

This is all for Box Plots. Now you might have got the idea of Box Plots how to make them and how to derive information from them. For any queries do leave a comment down below.

Box plot – Frequently Asked Questions (FAQs)

What do you mean by box plot?

A box plot, also known as a box-and-whisker plot, is a graphical representation of the distribution of a dataset. It summarizes key statistics such as the median, quartiles, and outliers, providing insights into the spread and central tendency of the data.

Box Plot is used for which type of data?

Box Plots gives a visual summary of the variability of values of dataset. Boxplots usually shows the numeric data values, especially is you want to compare multiple groups.

What information cannot be found in a box plot?

Information that are missed in a box plot is the detailed shape of the distribution. It is quite difficult to find the mean as it is visual representation of the data.

Is Box Plot vertical or horizontal?

Box Plot can either be drawn horizontally or vertically. It depends on the estimate L-estimators, range, mid-range and trimean.

Here's a complete roadmap for you to become a developer: Learn DSA -> Master Frontend/Backend/Full Stack -> Build Projects -> Keep Applying to Jobs

ML | T-distributed Stochastic Neighbor Embedding (t-SNE) Algorithm

T-distributed Stochastic Neighbor Embedding (t-SNE) is a nonlinear dimensionality reduction technique well-suited for embedding high-dimensional data for visualization in a low-dimensional space of two or three dimensions.

What is Dimensionality Reduction?

Dimensionality Reduction represents n-dimensions data(multidimensional data with many features) in 2 or 3 dimensions. An example of dimensionality reduction can be discussed as a classification problem i.e. student will play football or not that relies on both temperature and humidity can be collapsed into just one underlying feature since both features are correlated to a high degree. Hence, we can reduce the number of features in such problems. A 3-D classification problem can be hard to visualize, whereas a 2-D one can be mapped to simple 2-dimensional space and a 1-D problem to a simple line.

What is t-SNE Algorithm?

t-Distributed Stochastic Neighbor Embedding is a dimensionality reduction. This algorithm uses some randomized approach to reduce the dimensionality of the dataset at hand non-linearly. This focuses more on retaining the local structure of the dataset in the lower dimension as well.

This helps us explore high dimensional data as well by mapping it into lower dimensionss as the local structures are retained in the dataset we can get a feel of the same by ploting it and visualizing it in the 2D or may be 3D plane.

What is the difference between PCA and t-SNE algorithm?

Even though PCA and t-SNE both are unsupervised algorithms that are used to reduce the dimensionality of the dataset. PCA is a deterministic algorithm to reduce the dimensionality of the algorithm and the t-SNE algorithm a randomized non-linear method to map the high dimensional data to the lower dimensional. The data that is obtained after reducing the dimensionality via the t-SNE algorithm is generally used for visualization purpose only.

One more thing that we can say is an advantage of using the t-SNE data is that it is not effected by the outliers but the PCA algorithm is highly affected by the outliers because the methodologies that are used in the two algorithms is different. While we try to preserve the variance in the data using PCA algorithm we use t-SNE algorithm to retain teh local structure of the dataset.

How does t-SNE work?

t-SNE a non-linear dimensionality reduction algorithm finds patterns in the data based on the similarity of data points with features, the similarity of points is calculated as the conditional probability that point A would choose point B as its neighborr.

It then tries to minimize the difference between these conditional probabilities (or similarities) in higher-dimensional and lower-dimensional space for a perfect representation of data points in lower-dimensional space.

Space and Time Complexity

The algorithm computes pairwise conditional probabilities and tries to minimize the sum of the difference of the probabilities in higher and lower dimensions. This involves a lot of calculations and computations. So the algorithm takes a lot of time and space to compute. t-SNE has a quadratic time and space complexity in the number of data points.

Python Code Implementation of t-SNE on MNIST Dataset

Now let's use the sklearn implementation of the t-SNE algorithm on the MNIST dataset which contains 10 classes that are for the 10 different digits in the mathematics.

- Python3

```
# Importing Necessary Modules.

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.manifold import TSNE

from sklearn.preprocessing import StandardScaler
```

Now let's load the MNIST dataset into pandas dataframe. You can download this dataset from here.

- Python3

```
# Reading the data using pandas

df = pd.read_csv('mnist_train.csv')


# print first five rows of df

print(df.head(4))


# save the labels into a variable l.

l = df['label']


# Drop the label feature and

# store the pixel data in d.

d = df.drop("label", axis=1)
```

Output:

```
   label  pixel0  pixel1  pixel2  pixel3  pixel4  pixel5  pixel6  pixel7
0      1       0       0       0       0       0       0       0       0
1      0       0       0       0       0       0       0       0       0
2      1       0       0       0       0       0       0       0       0
3      4       0       0       0       0       0       0       0       0

   pixel8    ...     pixel774  pixel775  pixel776  pixel777  pixel778  \
0       0    ...            0         0         0         0         0
1       0    ...            0         0         0         0         0
2       0    ...            0         0         0         0         0
3       0    ...            0         0         0         0         0

   pixel779  pixel780  pixel781  pixel782  pixel783
0         0         0         0         0         0
1         0         0         0         0         0
2         0         0         0         0         0
3         0         0         0         0         0

[4 rows x 785 columns]
```

First five rows of the MNIST dataset

Before applying the t-SNE algorithm on the dataset we must standardize the data. As we know that the t-SNE algorithm is a complex algorithm which utilizes some comples non-linear methodologies to map the high dimensional data the lower dimensional it help us save some of the time complexity that will be needed to complete the process of reduction.

- Python3

```
# Data-preprocessing: Standardizing the data
from sklearn.preprocessing import StandardScaler


standardized_data = StandardScaler().fit_transform(data)
print(standardized_data.shape)
```

Output:

(42000, 784)

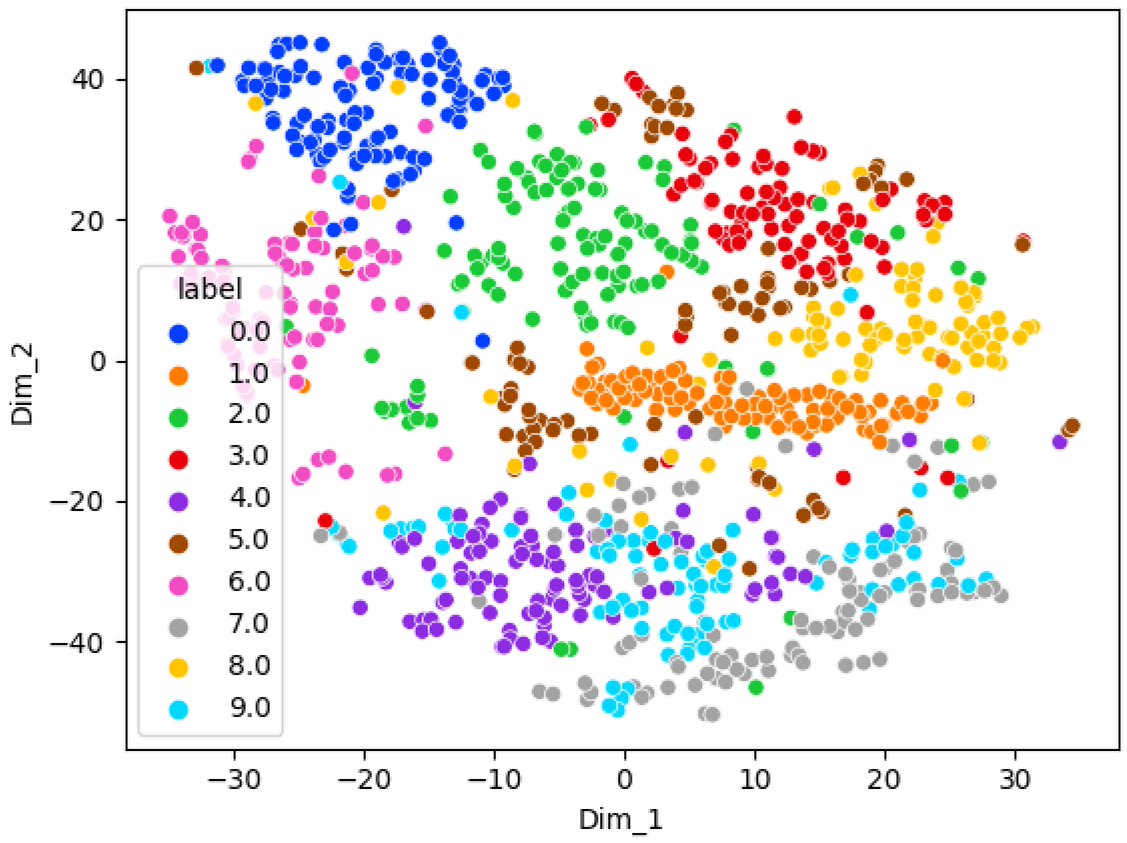
Now let's reduce the 784 columns data to 2 dimensions so, that we can create a scatter plot to visualize the same.

- Python3

```
# Picking the top 1000 points as TSNE
# takes a lot of time for 15K points
```

```
data_1000 = standardized_data[0:1000, :]
labels_1000 = labels[0:1000]


model = TSNE(n_components = 2, random_state = 0)
# configuring the parameters
# the number of components = 2
# default perplexity = 30
# default learning rate = 200
# default Maximum number of iterations
# for the optimization = 1000


tsne_data = model.fit_transform(data_1000)


# creating a new data frame which
# help us in plotting the result data
tsne_data = np.vstack((tsne_data.T, labels_1000)).T
tsne_df = pd.DataFrame(data = tsne_data,
    columns =("Dim_1", "Dim_2", "label"))


# Plotting the result of tsne
sn.scatterplot(data=tsne_df, x='Dim_1', y='Dim_2',
            hue='label', palette="bright")
plt.show()
```

Output:



MNIST data mapped to the 2D plane

Don't miss your chance to ride the wave of the data revolution! Every industry is scaling new heights by tapping into the power of data. Sharpen your skills and become a part of the hottest trend in the 21st century.

Dive into the future of technology - explore the Complete Machine Learning and Data Science Program by GeeksforGeeks and stay ahead of the curve.