**Data Science Maths**

**Descriptive Statistics:**

* **Mean:** It is described as the arithmetic average of all the data points. It can be calculated by diving the sum of total number of points / number of data points.

**Note:** *Mean is affected by extreme data points.*

* **Median:** It is described as the middle value in a sequence of data points ordered by rank.

If the total number of data points in the list is odd, then the median is the middle values.

Else, if the list is even, then the middle value is calculated by average the two middle values.

**Note:** *Median is not affected by extreme data points.*

* **Mode:** It is the data with the highest frequency. Unlike mean and median, Mode can be used with alphabets too.

**Variance and Standard Deviation:**

* **Variance:** It is a measure of how the data points are spread throughout the dataset.

It basically gives a measure of how far the data points are from each other and the mean.

Low variance indicates that the points are close to each other and to the mean.

High variance indicates that the points are far from each other and from the mean.

To calculate the mean the formula is:

Square of the distance of each data point from the mean/n where n is the number of points.

* **Standard Deviation**: It is the measure of how much variation from the mean exists.

A picture containing table

Description automatically generated

**Visualising Statistical Measures:**

* A ***frequency histogram*** gives us information about the frequency of occurrence of the data interval.
* ***Density plots*** give us information about the percentage of data lying within the interval.

**Calculating percentiles:**

* Percentile basically denotes that the number of values in data that lie below a certain value (Data needs to be sorted first)

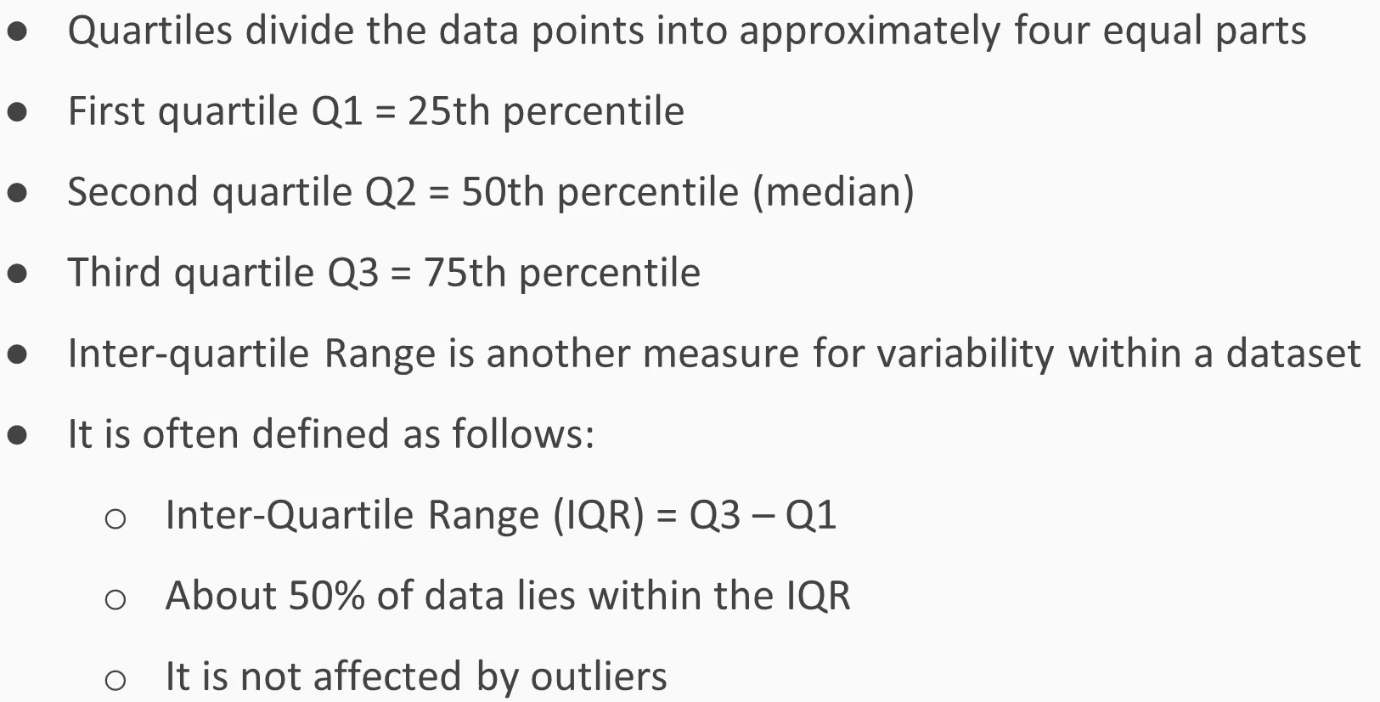
Text

Description automatically generated

Number of values before 5/ total number of values \* 100.

This will give 0.25\*100 which means 25% of the data is less than 25

**Quartiles and Box plots:**

****

**Dealing with missing data:**

* First way to find the missing values is datasetName.describe()

Table

Description automatically generated

Count returns number of non-null values. So, we can subtract it by total number of entries for that column or the serial number of the data set.

Another way is datasetName.isna(). True means value is null.

Opposite of this is datasetName.notna(). False means value is null.

Another way is datasetName.isnull(). True means value is null.

#Finding count of missing values in each column

datasetName.isnull().sum() 🡪 Gives the count of missing values in each column

Note: In date time formats, missing values are denoted by NaT.

**Dealing with missing values:**

* **Technique 1:** If thenumber ofmissing values in the rows are less then we can remove them, but if they are more then removing them could result in bad input for the machine learning model.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A picture containing diagram

Description automatically generated

Note: This might not always be the case as the replacing value can not be a hundred percent sure.

* **Technique 2:**

Text

Description automatically generated

A picture containing chart

Description automatically generated

Dropna() drops the rows will null values

Another way is to add a condition in the dropna function. Thresh=4. By doing this we are saying that keep the rows that have at least four non null values.

* **Technique 2:**

Another way is to removing columns that have a certain percentage of null values.



In the above code axis =1 means look into columns and axis = 0 means look into rows. By default, it is 0. In the case above it will keep values where at least 40% of rows in the column are not empty.

* **Technique 3:**

Graphical user interface, text, application

Description automatically generated

In this as we see the axis is zero, so this code keeps the rows with at least 60 %of the columns filled with values.

* **Technique 4:**

**Imputing Values:** In the imagebelow, we see a ‘***fillna’*** method which fill the null values with the value entered in parameter.

Table

Description automatically generated

Note: In addition to the above method, you need to inplace method after that to implement the changes.

Graphical user interface, text

Description automatically generated

* **Technique 5:**

**Backward Fill:** datasetName.bfill() function fills the null value with the previous value from the back as displayed in the tables below.

Table

Description automatically generated

* **Technique 6:**

**Forward Fill:** This function does the same thing as the bfill with the only difference of replacing values from top to bottom manner.

* **Technique 7:**

**Filling with central tendencies:** In this method we take out the mean of the values in that column and other columns. Then store them all together in a dictionary and then by using the fillna method we use an argument called value which takes our dictionary and replaces all the values where necessary by the mean we took out for values of each column.

Graphical user interface, text, application, email

Description automatically generated

* **Technique 8:**

**Imputing values based on condition:**

In this case we will separate the male and female and then replace the missing values by the mean of that particular gender.



**Outliers:**

**Graphical user interface, text, application, email

Description automatically generated**

Graphical user interface, text, application, email

Description automatically generated

Table

Description automatically generated

Table

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Using Z score to find outliers:**

Text, application

Description automatically generated

Chart, diagram

Description automatically generated

This image is a standard deviation mean chart which shows the data points spread from -4std to 4std with mean in the middle. Here x axis is Values and Y axis is Probability.

So, the way we see outliers is by following a general rule of thumb i.e. if a point is located past 3std and -3std then it is an outlier.

As an observation we can also see that 99% of the data points are located between 3std and -3std.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Chart, box and whisker chart

Description automatically generated

So, any datapoint that lies past the left side of the whiskers or the right side of the whiskers can be considered as an outlier.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Diagram

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Chart

Description automatically generated

Note: *unique vs value\_counts: Unique just lists out the unique values and value\_counts gives the total number of each unique value.*