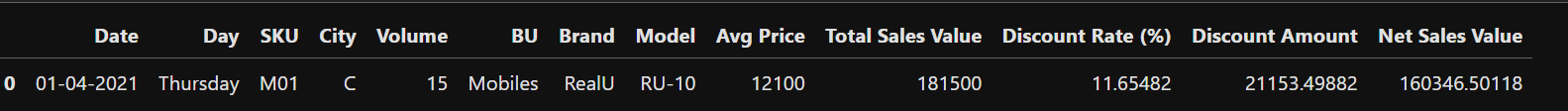
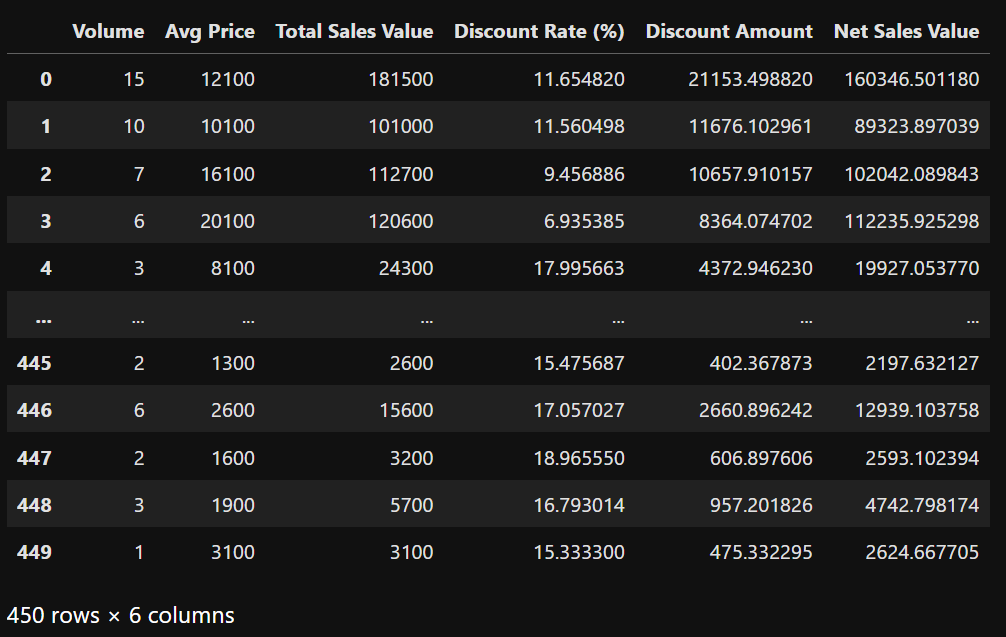
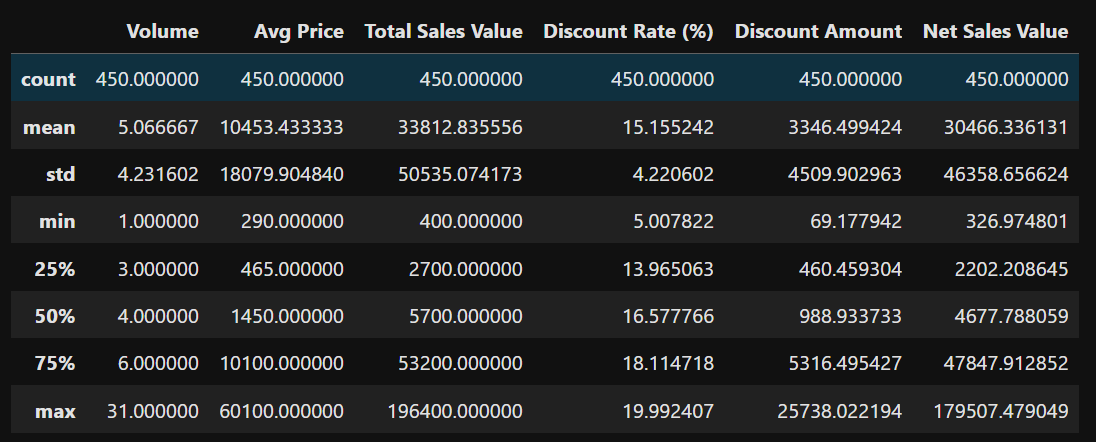
.**BASIC STATISTICS LEVEL 1 ASSIGMENTS**

* **Importing libraries** :-
  + Code:
    - import **pandas** as **pd**
    - import **numpy** as **np**
    - import **matplotlib.pyplot** as **plt**
* **Loading a Data Set :-**
  + Code:
    - df = pd.read\_csv('sales\_data\_with\_discounts.csv')
    - df.head(1)
  + result:-

**fig1**

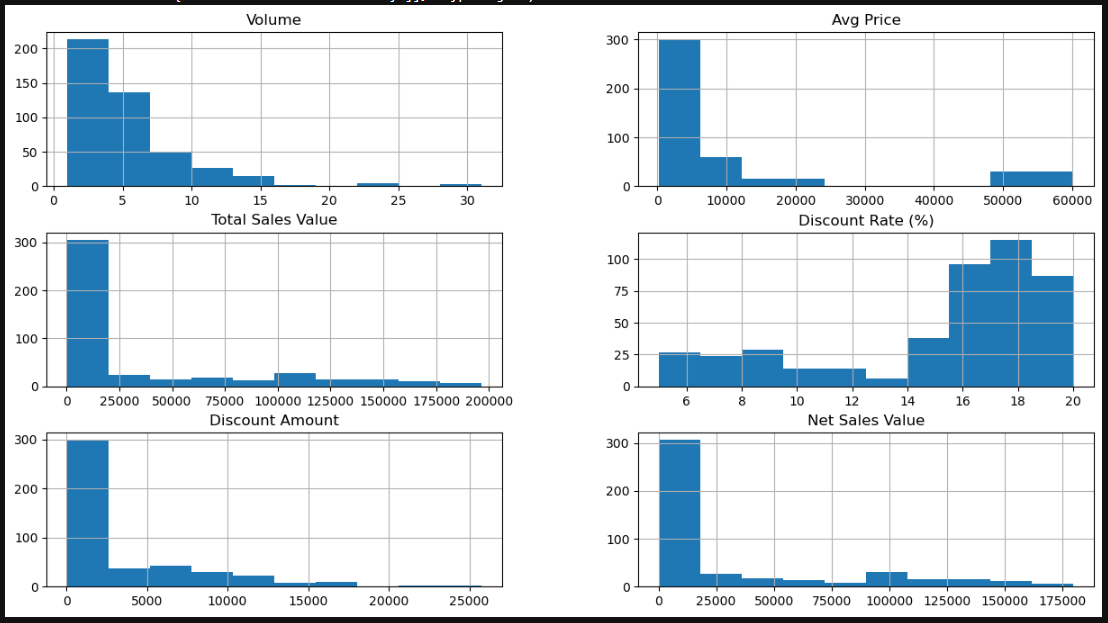
* **Numerical columns in Dataset:-**
  + Code:
    - df.select\_dtypes( ‘int64’ , ’float64’)
  + ****result:- Number of numerical columns in dataset=6

**fig2**

* **Calculating mean, median, mode, and standard deviation for these columns.**
  + Code:
    - df.describe()
  + ****result:-

**fig3**

* **Plot histograms for each numerical column:-**
  + Code:
    - df.select\_dtypes(['int64','float64']).hist(figsize=(15,8))
  + result:-

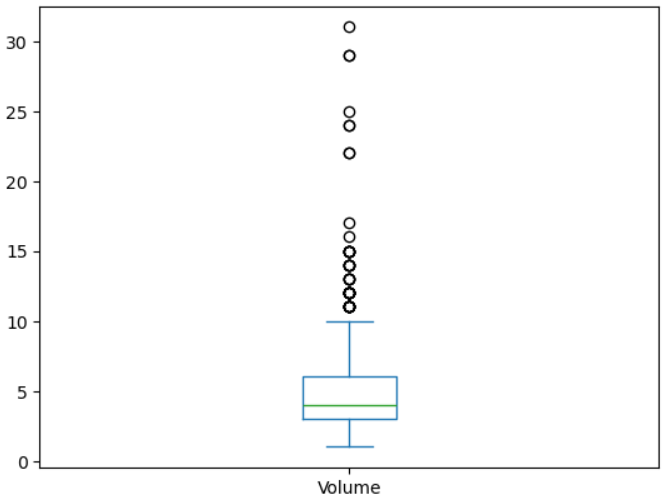


**fig4**:

* **Create boxplots for numerical variables to identify outliers and the interquartile range**
  + Code:
    - For Volume:

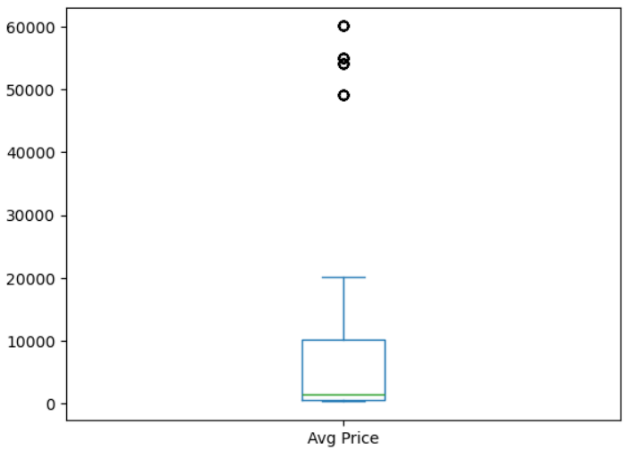
**df['Volume'].plot(kind='box')**

* + - * result:

**fig5** 

* + - For Ang Price:

**df['Avg Price'].plot(kind='box')**

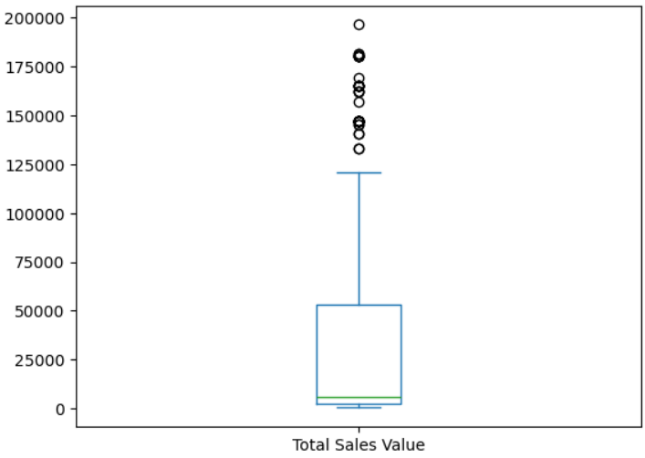
* + - * **result:**

**fig6**

* + - For Total Sales Value:

**df[‘Total Sales Value’].plot(kind='box')**

* + - * Result:

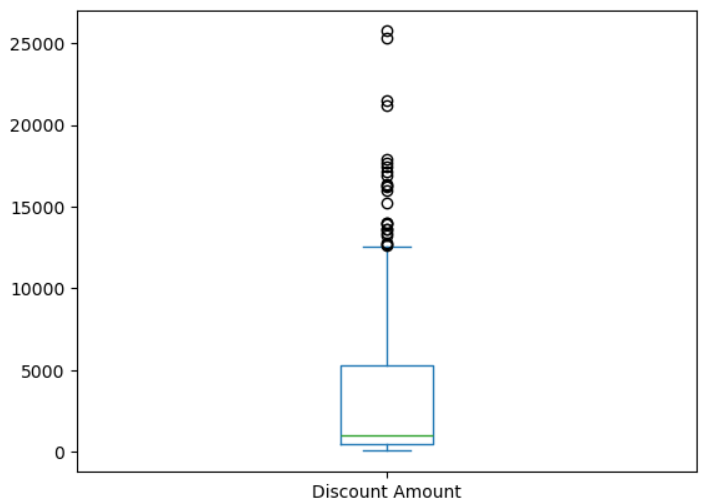


**fig7**

* + - For Discount Amount:

**df[‘Discount Amount’].plot(kind=’box’)**

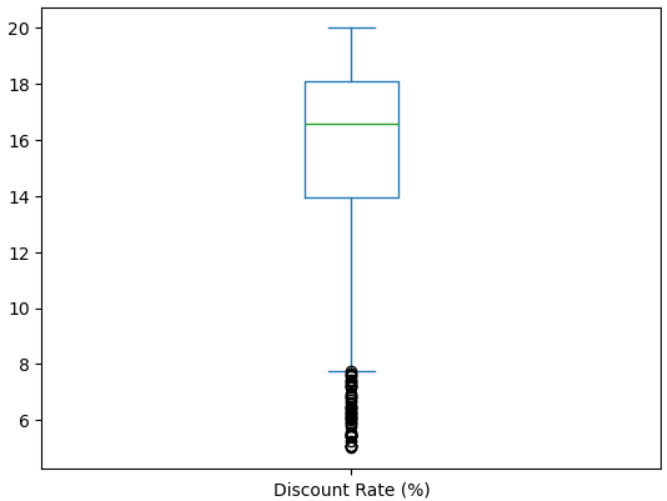
* + - * result:

**fig 8** 

* + - For Discount Range (%)

**df.[‘Discount Rate (%)’].plot(kind=’box’)**

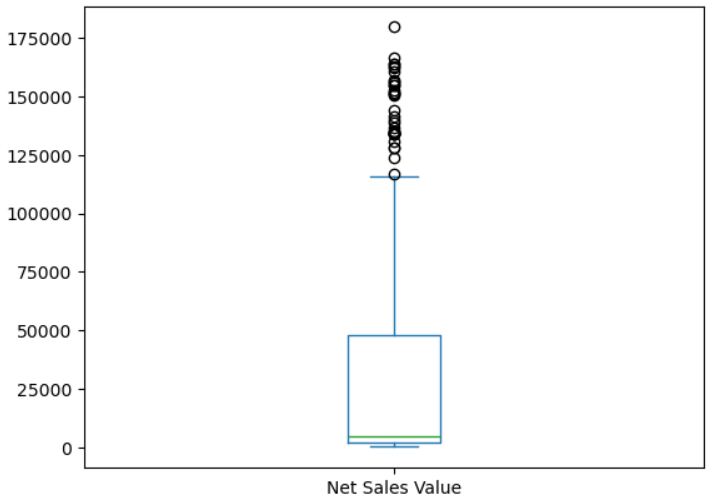
* + - * result:

**fig 9** 

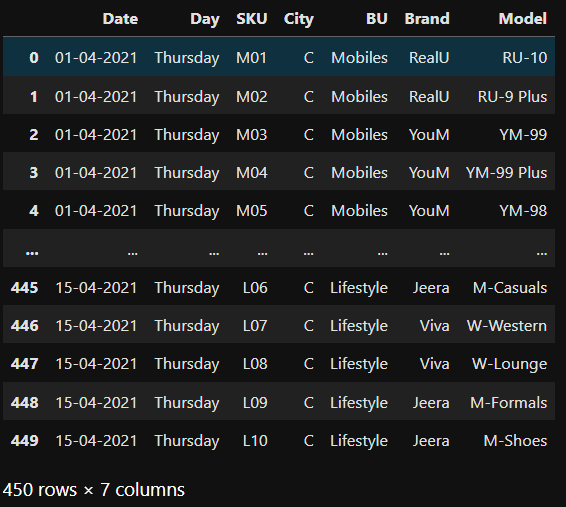
* + - For Net Sales Value

**df[‘Net Sales Value’].plot(kind=’box’)**

* + - * result

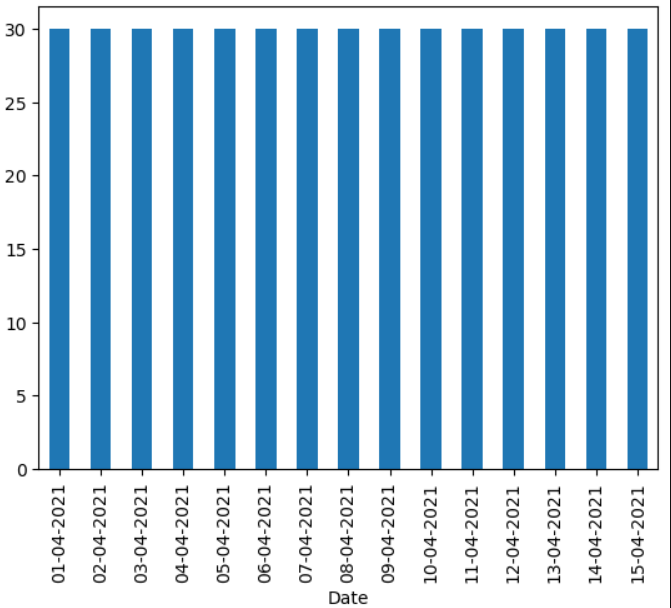
**fig10** 

* **Identify categorical columns in the dataset:**
  + Code:
    - **df.select\_dtypes([‘object’])**
  + result:

**fig11**

* Create bar charts to visualize the frequency or count of each category.
  + For Date:

**df[‘Date’].value\_counts().plot(kind=’bar’)**

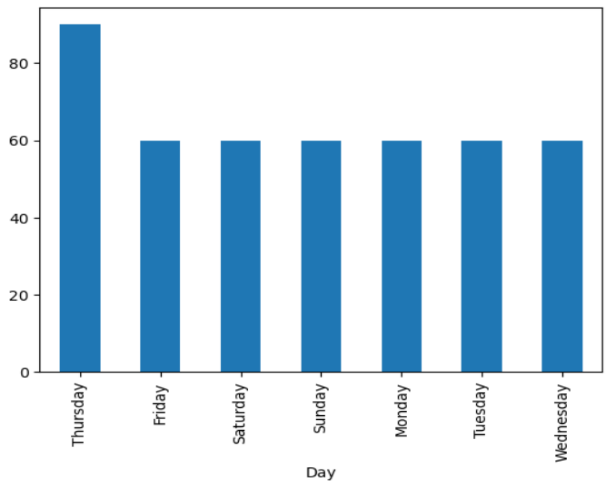
* + - result

**fig12**

* + For Day

**df[‘Day’].value\_counts().plot(kind=’bar’)**

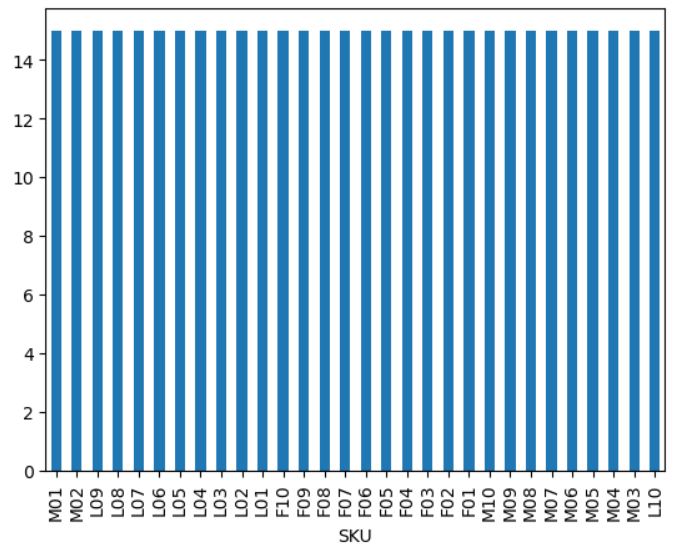
* + - Result



**fig13**

* + For SKU

**df[‘SKU’].value\_counts().plot(kind=’bar’)**

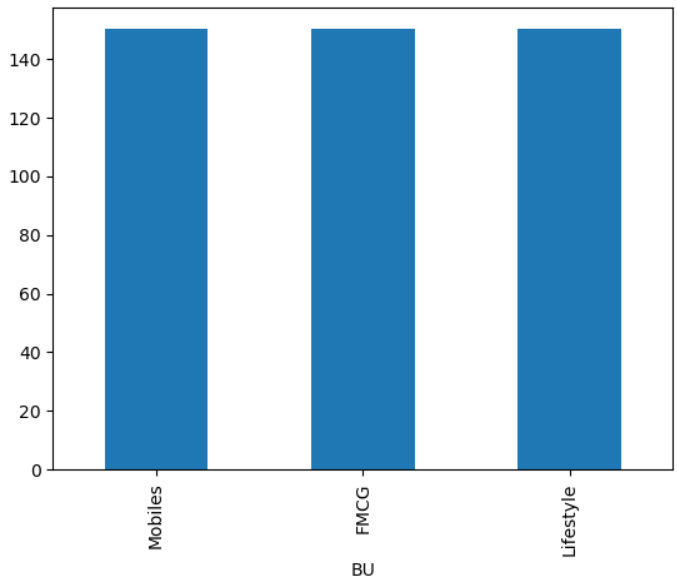
* + - result:

**fig14**

* + For BU

**df[‘BU’].value\_counts().plot(kind=’bar’)**

* + - Result

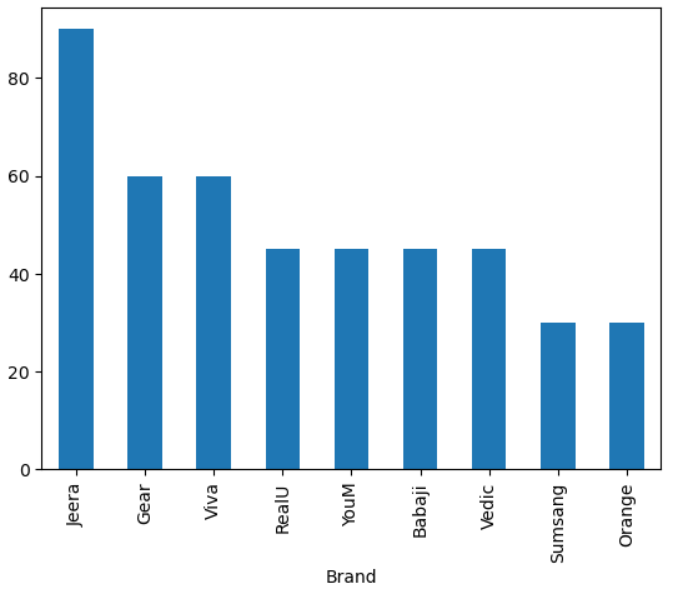


**fig15**

* + For Brands

**df[‘Brands’].value\_counts().plot(kind=’bar’)**

result:

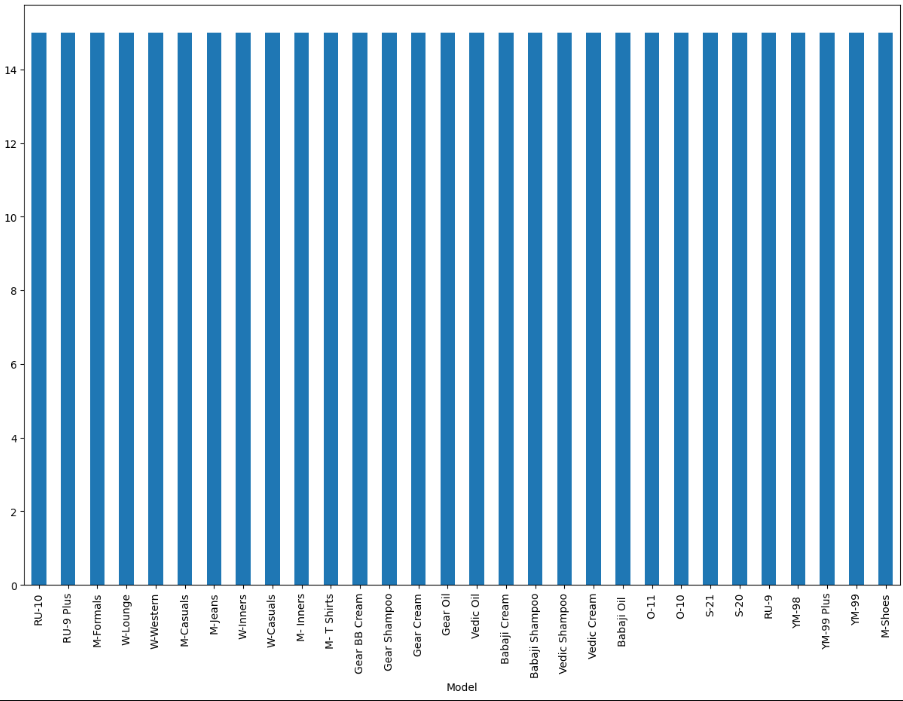


**fig16**

* + For Model

**df[‘Model’].value\_counts().plot(kind=’bar’)**

result:



**fig17**

**Standardization of Numerical Variables:**

Standardization transforms/scaling data into a standard format, making it easier for computers to use and understand.

**z-score normalisation:**

Z score standardization uses mean and standard deviation from given data to standardize.

If z-score is 0 than data point is at mean.

If z-score is 1. Tells data point lies 1standard deviation more than mean

If z-score is -2. Tells data point lies 2standard deviation less than mean

Z = x -µ /

X -data point, µ - mean ,  – Standard deviation

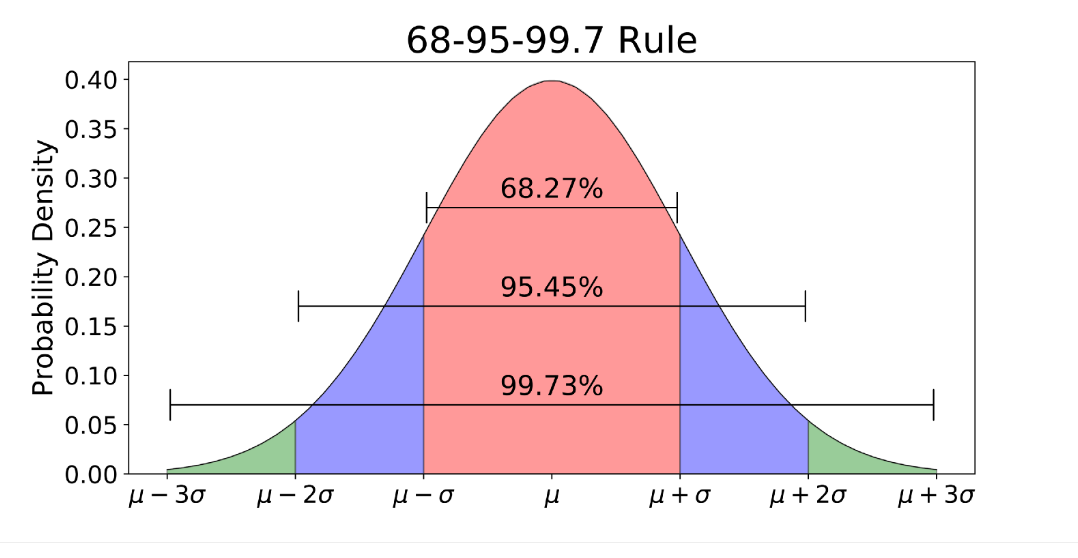
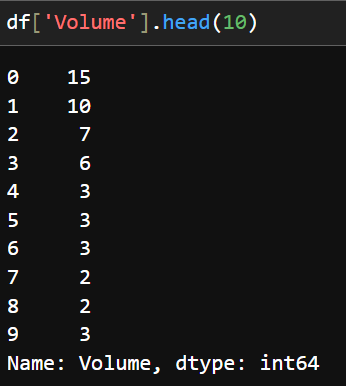
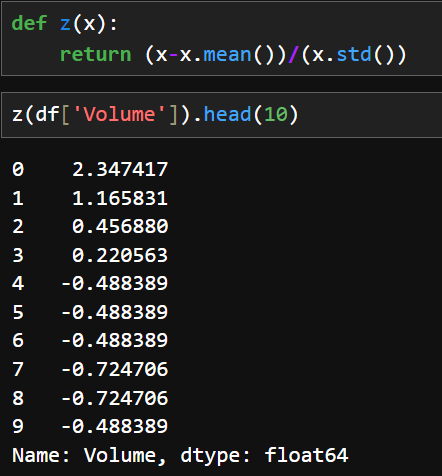


fig-18



Volume data set before normalisation



Volume data set after normalisation

**Dummy Variables:**

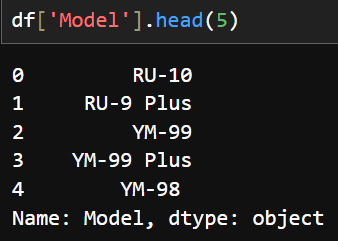
Dummy Variable are use to convert categorical data into numerical variable. It helps in applying algorithms for Machine Learning.

One-hot encoding is the process of creating dummy variable using binary number (0,1).

The active data will be 1 and in active will be 0

for example:

below represents 5 row on Model dataset having categorical data of 4 different types.



After applying one-hot encoding the above dataset divides into 5 columns of each catogeries.

If the data is in row is present it replace denotes with 1.

If not it place 0.

Syntax: pd.get\_dummies(df, dtype=’int’)

df = dataframe ,

dtype = which datatype. If data type is not given it takes bool dtype.

Result:

