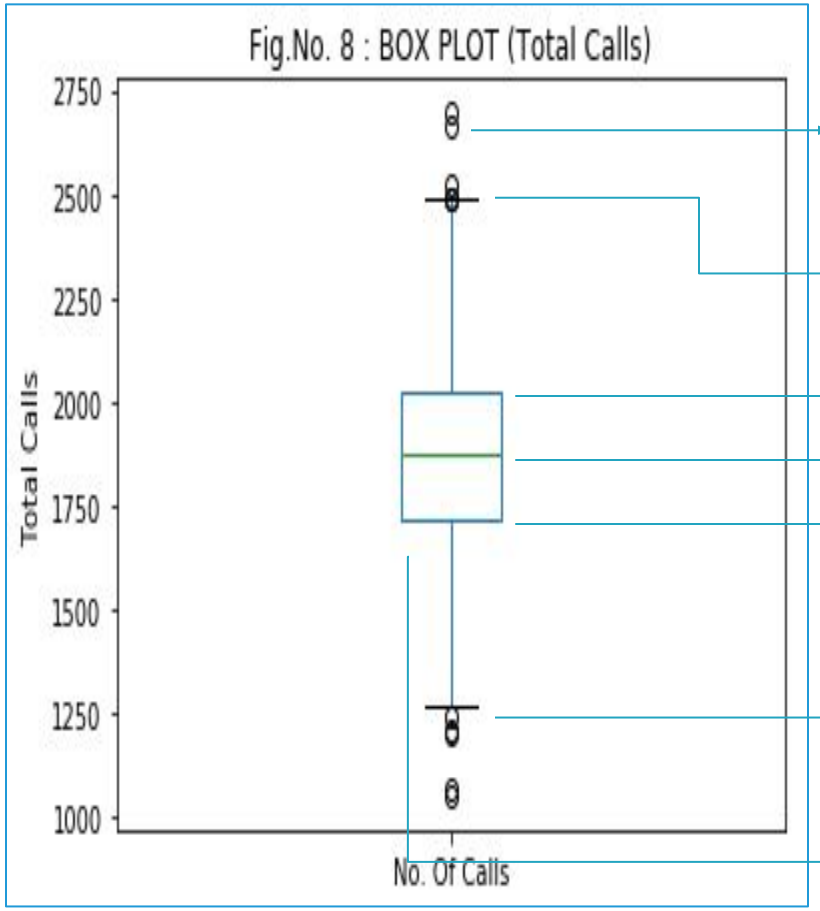
Other Basic Graphs with Python

Contents

Box – Whisker Plot



Outliers : Extreme Data values

Maximum value in the data : Q3+1.5*IQR

Q3 : Third Quartile

Q2 : Second Quartile (Median)

Q1 : First Quartile

Minimum value in the data : Q1-1.5*IQR

Box: Represents middle 50% Data

This plot shows that the distribution of total call is very much symmetric & there exists few outliers in the data.

* The minimum and maximum values are the ones excluding the outliers

Case Study

To get a better understanding of the subject, we shall consider the below case as an example.

Background

A telecom service provider has the Demographic and Transactional information of their customers

Objective

To visualise the distribution of their customer database
To see how the Calls and Amount are distributed across customers

Sample Size

1000

* Here we continue to use previous data for our further analysis.

Data Snapshot

telecom data

Variables

Observations

CustID	Age	Gender	PinCode	Active	Calls	Minutes	Amt	AvgTime	Age_Group
1001	29	F	186904	Yes	2247	18214	3168.76	8.105919	18-30

Columns	Description	Type	Measurement	Possible values
CustID	Customer ID	Numeric	-	-
Age	Age of the Customer	Numeric	-	-
Gender	Gender of the Customer	Categorical	M, F	2
PinCode	Pincode of area	Numeric	-	-
Active	Active usage of telecom	Categorical	Yes, No	2
Calls	Number of Calls made	Numeric	-	positive values
Minutes	Number of minutes spoken	Numeric	minutes	positive values
Amt	Amount charged	Continuous	Rs.	positive values
AvgTime	Mean Time per call	Continuous	minutes	positive values
Age_Group	Age Group of the Customer	Categorical	18-30, 30-45, >45	3

Box Plot in Python

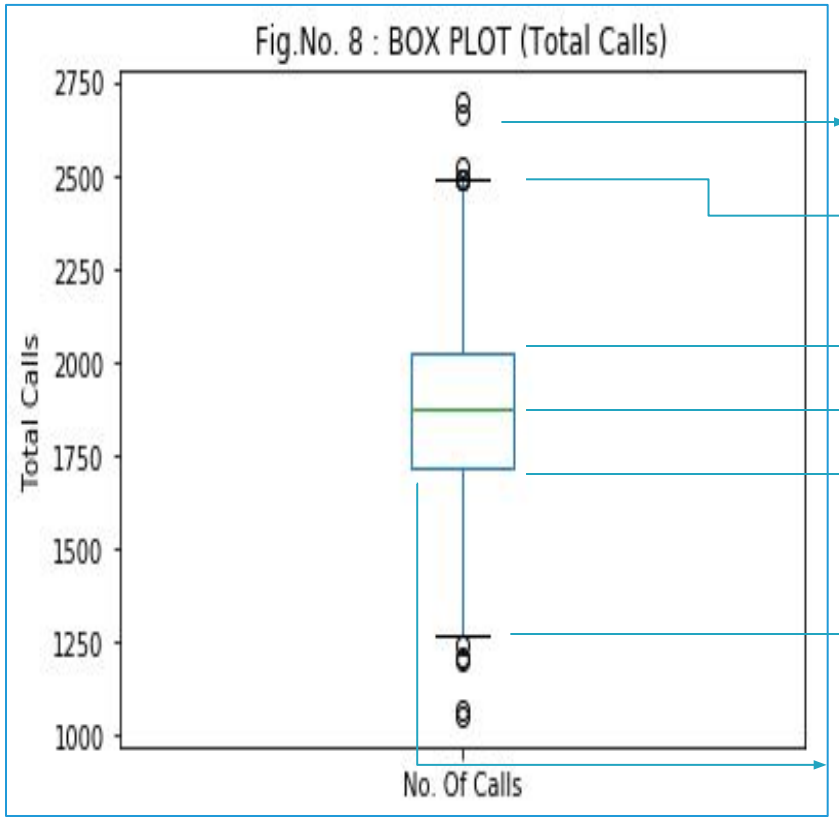
```
#Importing Data
import pandas as pd
telecom = pd.read_csv("telecom.csv")

#BoxPlot – Total Calls
import matplotlib.pyplot as plt
telecom.Calls.plot.box(label='No. Of Calls');plt.title('Fig.No. 8 :
BOX PLOT (Total Calls)');plt.ylabel('Total Calls')
```

- **box()** in pandas yields a different types of box chart
- **Calls** specfies vector (column) for which the box plot needs to be plotted
- **label=** provides a user defined label for the variable on X axis
- **ylabel** provides a user defined label for the variable on Y axis

Box Plot in Python

```
# Output
```



Outliers : Extreme Data values

Maximum value in the data : Q3+1.5*IQR

Q3 : Third Quartile

Q2 : Second Quartile (Median)

Q1 : First Quartile

Minimum value in the data : Q1-1.5*IQR

Box: Represents middle 50% Data

Interpretation :

- This plot shows that the distribution of total call is very much symmetric & there exists few outliers in the data.

Box Plot in Python

```
#BoxPlot for different categories of Age_Group
telecom.boxplot(column='Calls', by='Age_Group', grid=False,
patch_artist=True);plt.title('Fig.No. 9 : BOXPLOT – Average Call
Time');plt.suptitle('');plt.ylabel('Total Calls')
```

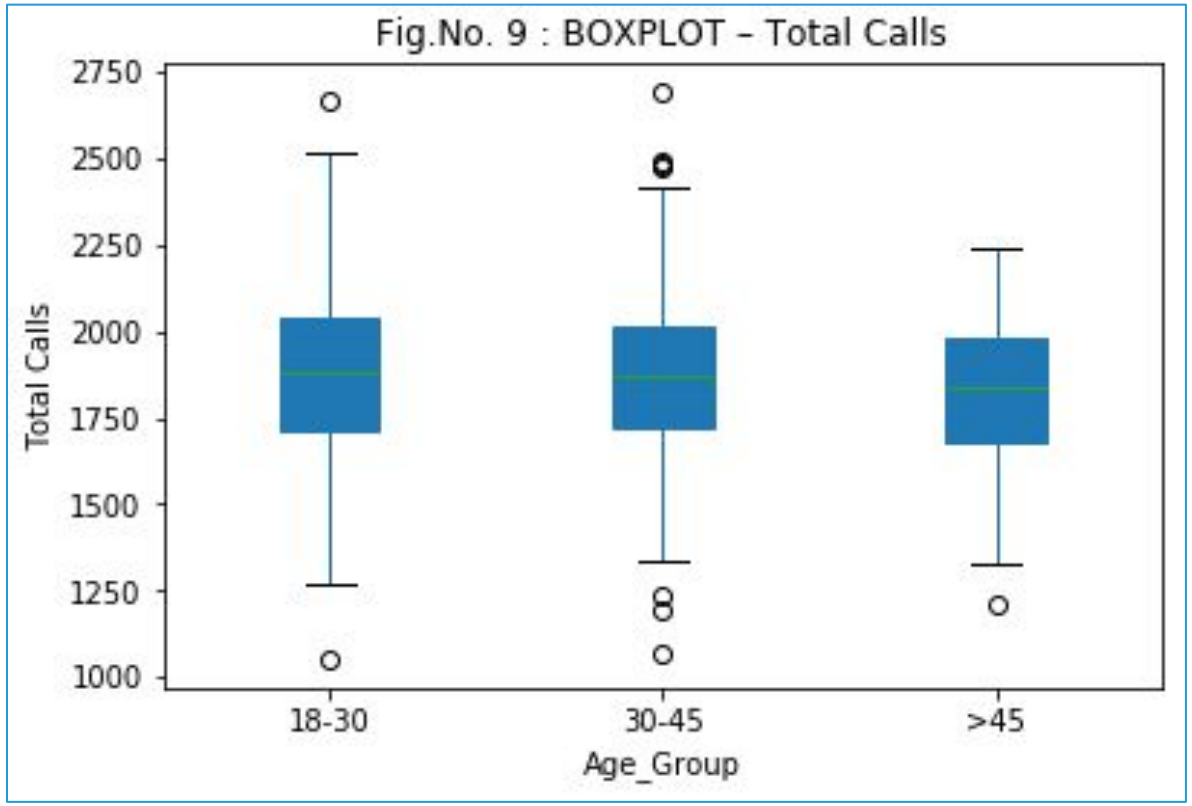
Difference between previous boxplot & this boxplot code is,

- **boxplot()** in pandas yields different types of box chart. It's a different way of writing **plot.box()**
- **column** specifies vector (variable) for which the box plot needs to be plotted
- **by** Specifies the vector (column) by which the distribution should be plotted.
- **ylabel** provides a user defined label for the variable on Y axis
- [illegible] in each plot

* Note : Re – order the levels of Variable Age_Group as explained in previous ppt before you execute the boxplot code, Age_Group wise.

Box Plot in Python

```
# Output
```

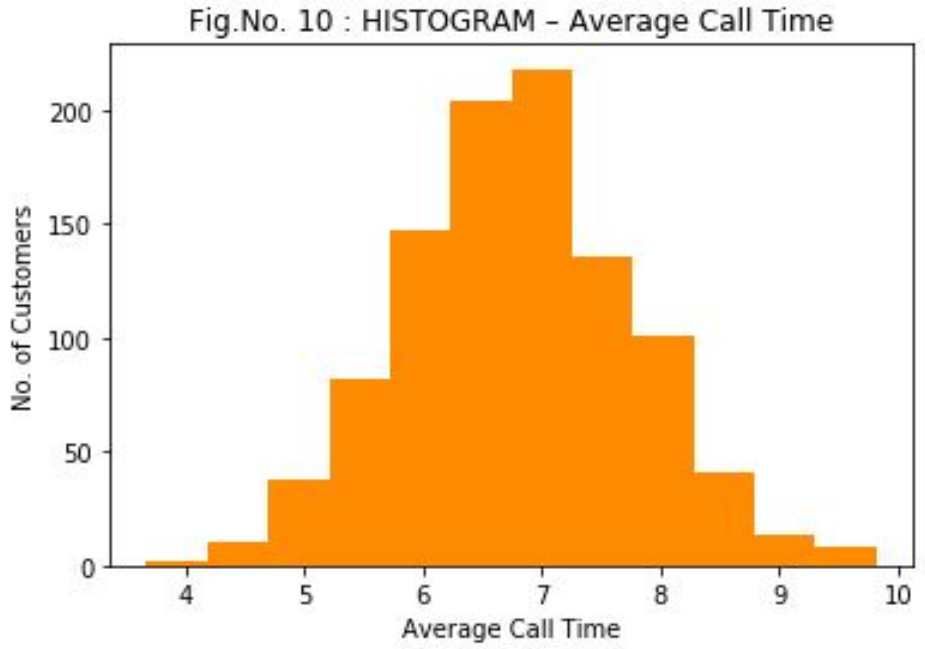


Interpretation :

- Here we can observe that the spread of total calls is higher in the age group 18-30.
- The number of outliers is higher in 30 – 45 age group.

Histogram

- A Histogram is similar to a bar chart but is used to display continuous data. Therefore we will use a continuous scale with no 'gaps' between the bars.
- It is generally used to check the Normality of the data.



- This plot shows that the distribution of Average Call Time is very much symmetric.

Histogram in Python

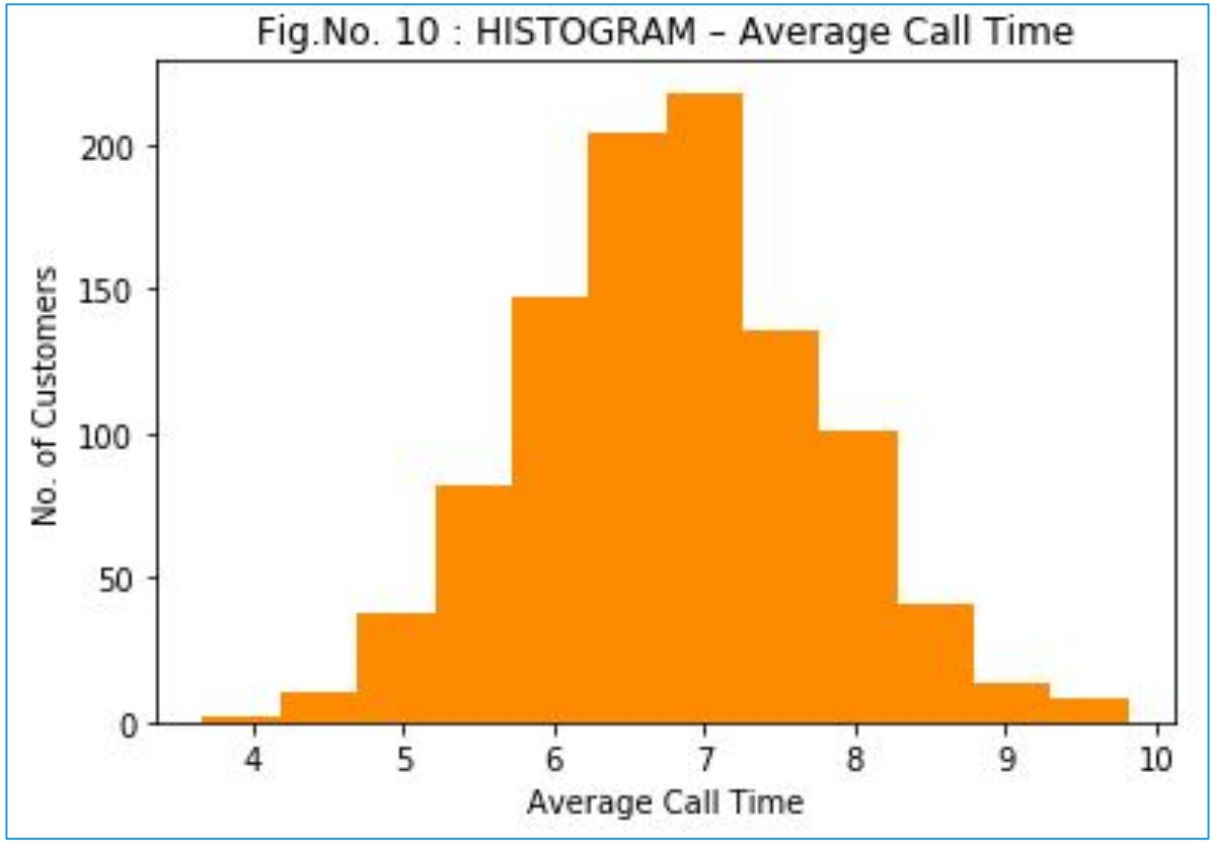
```
# Histogram – Average Call Time

telecom.AvgTime.hist(bins=12,grid=False, color = 'darkorange');
plt.title('Fig.No. 10 : HISTOGRAM – Average Call Time');
plt.xlabel('Average Call Time');plt.ylabel('No. of Customers')
```

- **hist()** yields a histogram
- **bins** specifies the width of each bar
- **xlabel** provides a user defined label for the variable on X axis
- **ylabel** provides a user defined label for the variable on Y axis
- **color** can be used to input your choice of color to the bars

Histogram in Python

This plot shows the distribution of Average Call Time

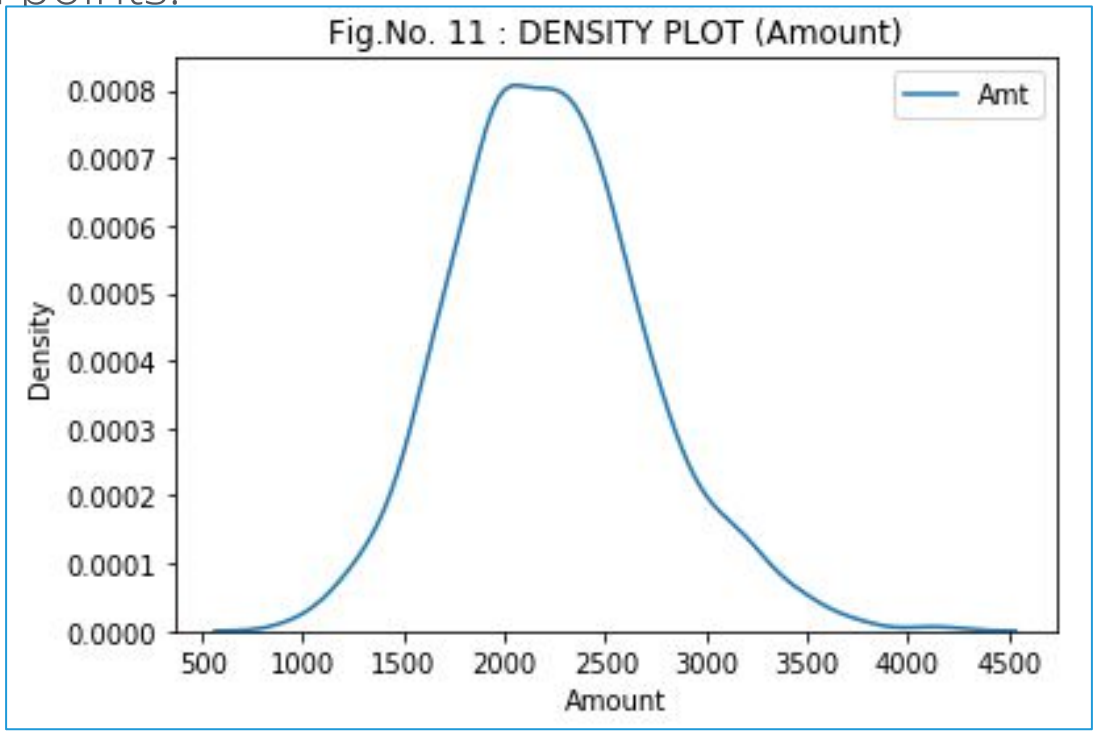


Interpretation :

- This plot shows that the distribution of Average Call Time is quite symmetric.

Density Plot

- A Density Plot is similar to a histogram which plots the probability.
- It is generally used to check the Normality of the data when there are higher data points.



- This plot shows that the distribution of amount is very slightly positively skewed.

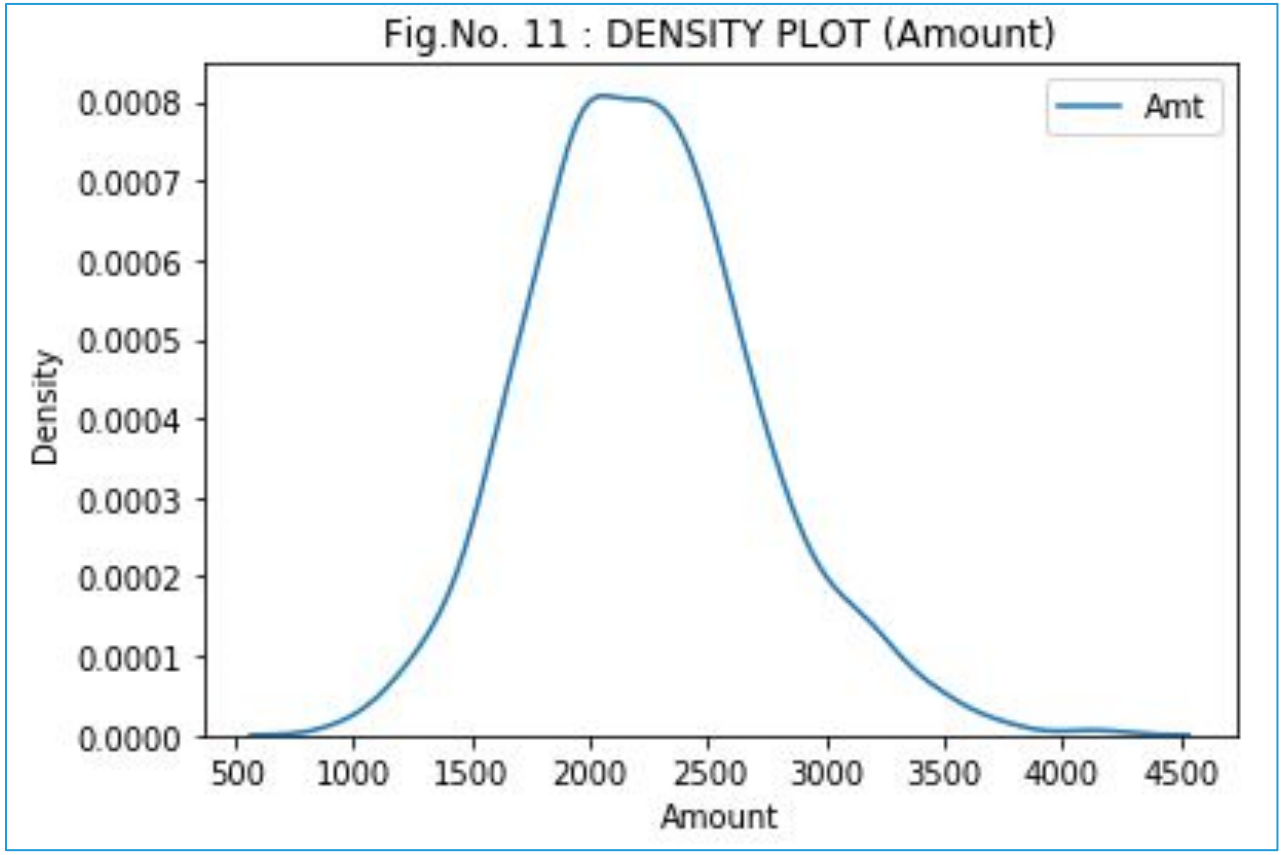
Density Plot in Python

```
# Density Plot - Amount
(telecom['Amt']).plot.kde();plt.title('Fig.No. 11 : DENSITY PLOT
(Amount)');plt.xlabel('Amount')
```

- **kde()** returns the density values of the variable (kernel density estimation)
- **plot()** plots the line graph of the specified variable
- **title** provides the user defined name of the chart. It is to be put in double quotes
- **xlabel** provides a user defined label for the variable on X axis

Density Plot in Python

This plot shows the distribution of Amount

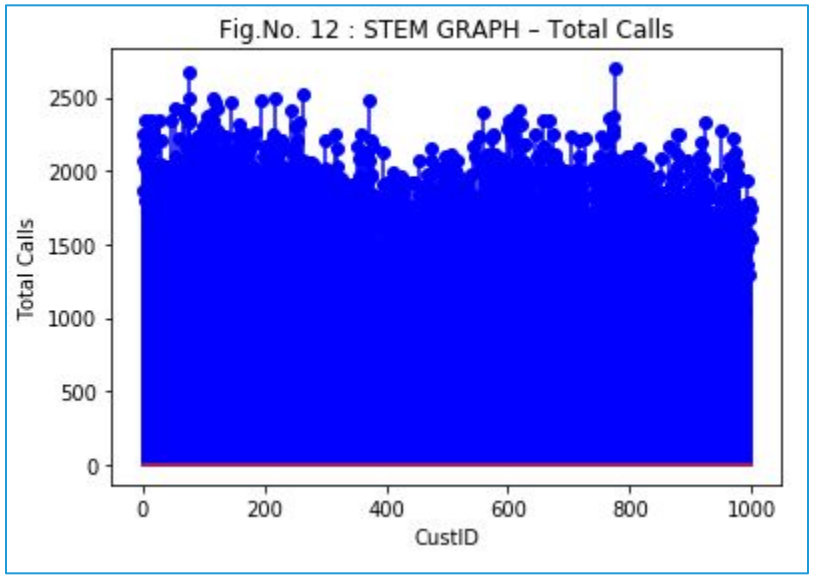


Interpretation :

- This plot shows that the distribution of Amount is slightly positively symmetric as smaller amount has slightly high frequency count of customers.

Stem and Leaf Plot

- A Stem and Leaf diagram can, again, be an alternative to a histogram.
- It is a special table where each numeric value is split into a stem (First digit(s)) and a leaf (last Digit)
- Stem and leaf diagrams show the shape of the distribution (like bar charts) but have the advantage of not losing the detail of the original data.
- Arranging the leaves in numerical order, will allow us to use the diagram to find the middle value (the median) and the values that are a quarter and three-quarters of the way through the data (the lower and upper quartiles).



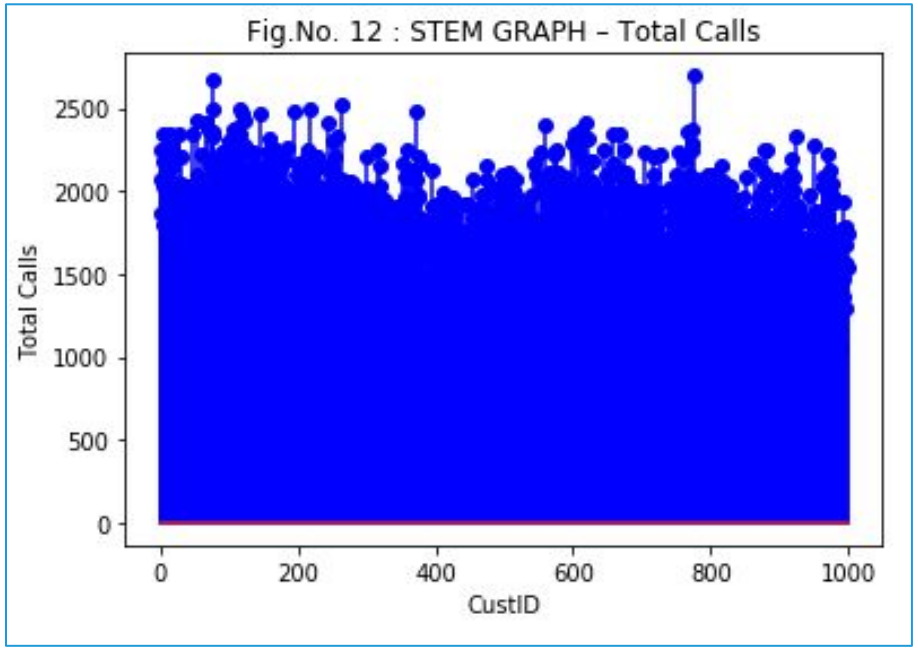
Stem and Leaf Plot in Python

```
# Stem Plot in Python
plt.stem(telecom.Calls);plt.title('Fig.No. 12 : STEM GRAPH – Total
Calls'); plt.xlabel('CustID'); plt.ylabel('Total Calls')
```

- **stem()** in base Python yields a stem chart
- **telecom.Calls** specfies vector (variable) for which the stem plot needs to be plotted

Stem and Leaf Plot in Python

```
# Output
```

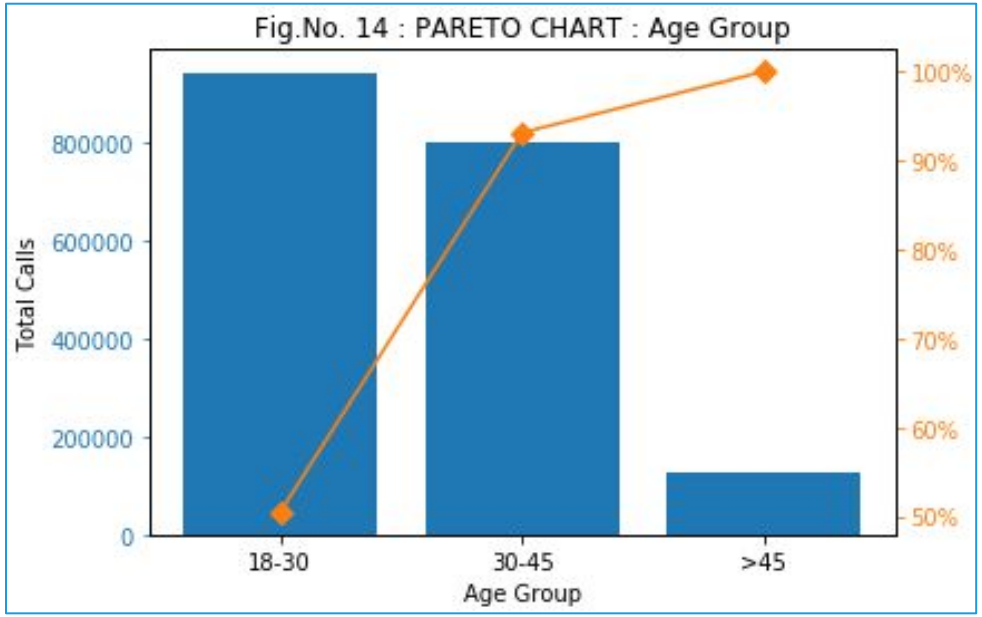


Interpretation :

- The stem and leaf plot of overall calling data shows that, calls values are symmetrically distributed and there exists few outliers also in the data.

Pareto Chart

- Pareto chart, named after Vilfredo Pareto, is a type of chart that contains both a bar and a line graph, where individual values are represented in descending order by bars. In this way the chart visually depicts which categories are more significant. The cumulative total is represented by the line.
- There needs to be at least one categorical variable to plot this chart.



- From the above chart we can interpret that 50% of the Total calls made come from age group 18-30.
- Another 42% calls are made by age group 30-45, only 8% calls are made by customers > 45 .

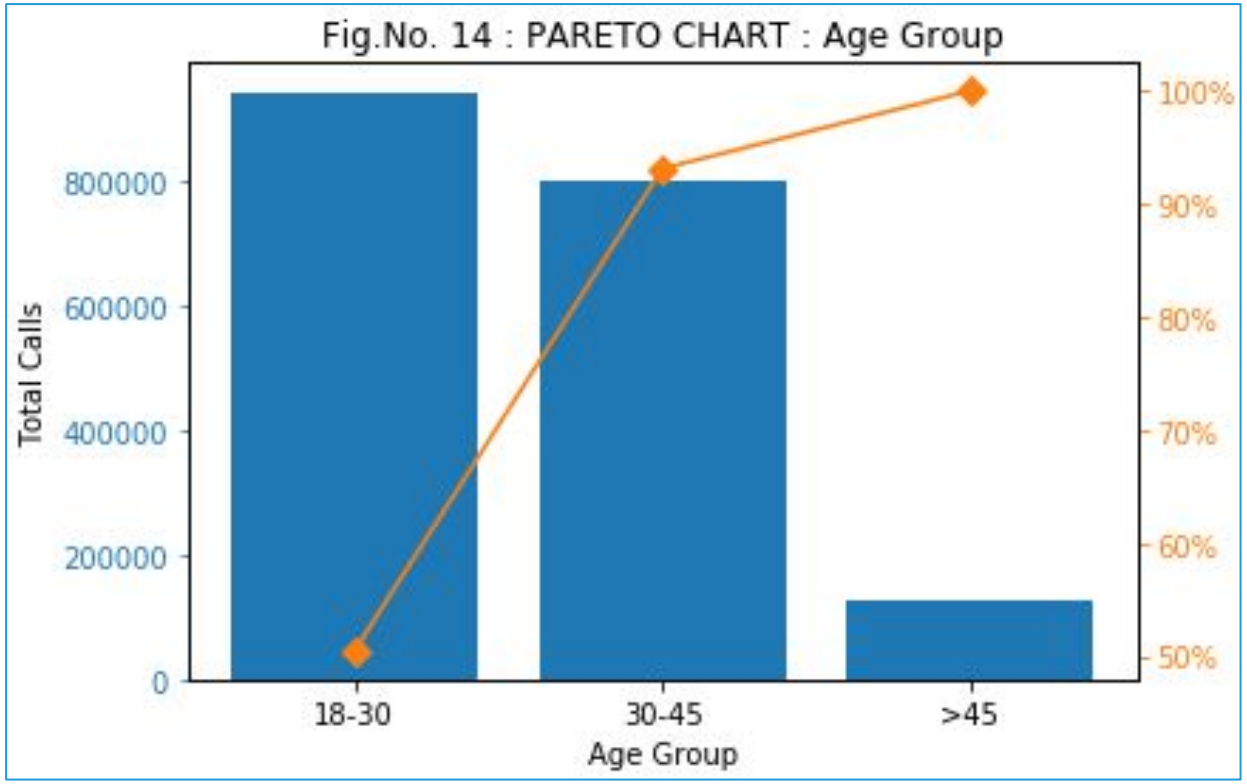
Pareto Chart in Python

```
# Pareto Chart – Age Group
telecom1 = telecom.groupby('Age_Group')['Calls'].sum()
telecom1
telecom1 = telecom1.to_frame()
telecom1["cumpercentage"]=telecom1["Calls"].cumsum()/telecom1["Calls"].sum(
)*100
fig, ax = plt.subplots()
ax.bar(telecom1.index, t
ax2.plot(telecom1.index,
ms=7);ax2.yaxis.set_majo
ax.tick_params(axis="y",
colors="C1");ax.set_xlab
Calls");ax.set_title("Fi
```

- **to_frame()** function is used to convert the given series object to a dataframe
- **plt.subplots** method provides a way to plot multiple plots on a single figure. Given the number of rows and columns , it returns a **tuple (fig , ax)**, giving a single figure fig with an array of axes ax
- **telecom1.index** is the argument that allows the bars to be named according the row names in the variable mentioned
- **telecom1[“Calls”]** specifies vector (variable) for which the Pareto chart needs to be plotted
- **ax2.twinx()** Create a twin Axes sharing the X axis
- **set_major_formatter(PercentFormatter())** sets percentage format on y axis for pareto chart
- **ax.tick_param** provides axis ticks the chart. It has to be put in double quotes
- **colors** can be used to input your choice of color to the bars
- **ax.set_xlabel, ax.set_ylabel** provides a user defined label for the variable on X and Y axes

Pareto Chart in Python

```
# Output
```



Interpretation :

- 50% of the Total calls made come from age group 18-30.
- Another 42% calls are made by age group 30-45, only 8% calls are made by customers > 45

Get an Edge!

Graphs for different types of Variables

Type of Variable	Chart
Discrete	Bar Graph
Continuous	Histogram/Boxplot/Density Plot
Categorical	Bar Graph/Pie Chart/Pareto Chart
Dichotomous	Multiple/ Stacked Bar Chart

Quick Recap

In this session, we learnt data visualisation using basic graphs

Chart Types and Functions in Python	
	• Box-Whisker Plot – `box(),boxplot()` • Histogram - `hist()` • Density Plot - `kde()` • Stem Plot- `stem()` • Pareto Chart – `bar() + twinx() + plot()`