(v2) Data Visualisation – I

Contents

About Data Visualisation

What is Data Visualisation?

It is the visual representation of data in the form of graphs and plots.

Why is it important?

It enables us to

- See the data and get insights in one glance
- Allows us to grasp difficult / complex data in an easy manner
- Helps us to identify patterns or trends easily. Also shows distribution, correlation and causality in data.

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

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

Simple Bar Diagram

A Bar Chart is the simplest and the most basic form of graph. In this graph, for each data item, we simply draw a ‘bar’ showing its value.

Simple Bar Chart: It is a type of chart which shows the values of different categories of data as rectangular bars with different lengths. The values are generally :

- Frequency
- Mean
- Totals
- Percentages

Diagrams in Python

```
#Importing the
Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

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

```
#Aggregating Data
telecom1 = telecom.groupby('Age_Group')['Calls'].sum()
telecom1
```

```
             Calls
Age_Group
18-30       943187
30-45       798721
>45         128870
```

For plotting a bar chart in Python, it is important to aggregate the data using **groupby()** to get required vector/matrix]

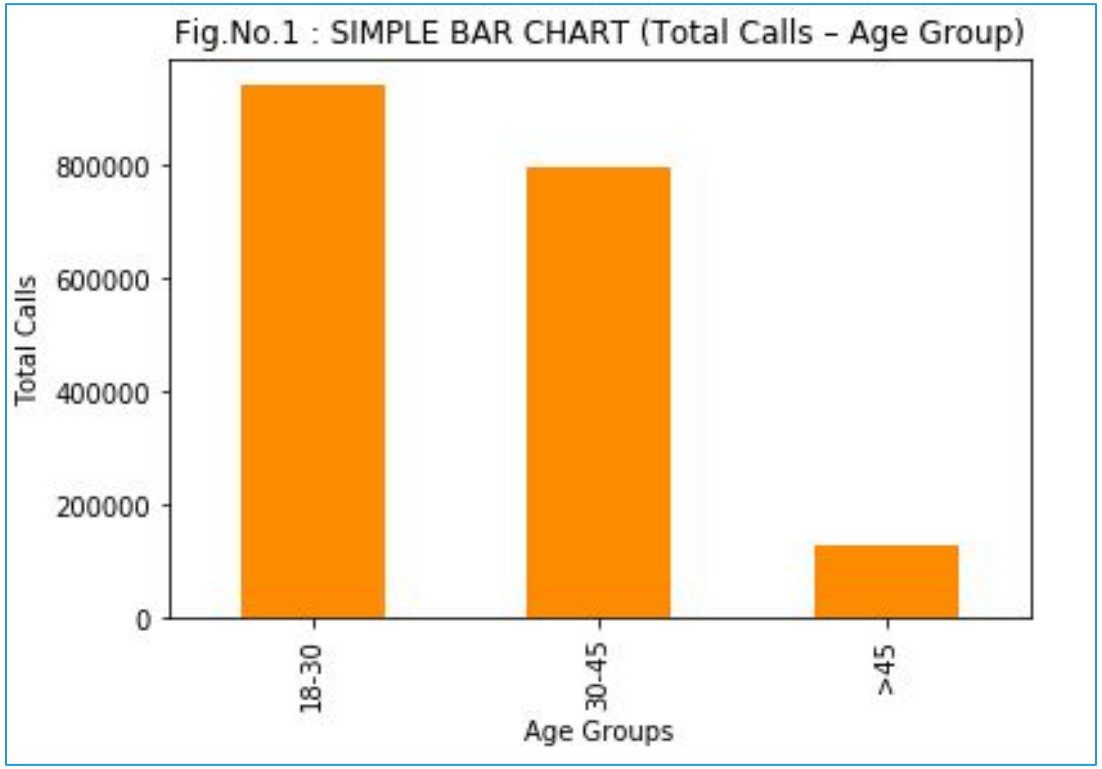
Simple Bar Chart in Python

```
#Simple Bar Chart – Total Calls for different Age Groups
plt.figure(); telecom1.plot.bar(title='Fig.No.1 : SIMPLE BAR CHART (Total Calls – Age Group)', color='darkorange'); plt.xlabel('Age Groups'); plt.ylabel('Total Calls')
```

- **plt.figure()** function is a convenient method to plot all columns with labels.
- **Plot.bar()** plots a bar chart. Can also be called by passing the argument **kind ='bar'** in plot.
- **title** is a string argument to give the plot a title.
- **color** argument specifies the plot colour. Accepts strings, hex numbers and colour code.
- **plt.xlabel** function/method to specify the x label.
- **plt.ylabel** function/method to specify the y label.

Simple Bar Chart in Python

This graph simply gives the distribution of the **Total number of calls** across different **Age Groups**.



Interpretation :

- Number of calls made by young age group (18-30) is slightly higher than mid age group (30-45) and very high than age group >45.

Simple Bar Chart in Python

```
# Simple Bar Chart – Mean Calls for different Age Groups
telecom2 = telecom.groupby('Age_Group')['Calls'].mean()
telecom2
```

```
                Calls
Age_Group
18-30      1882.608782
30-45      1866.170561
>45        1815.070423
```

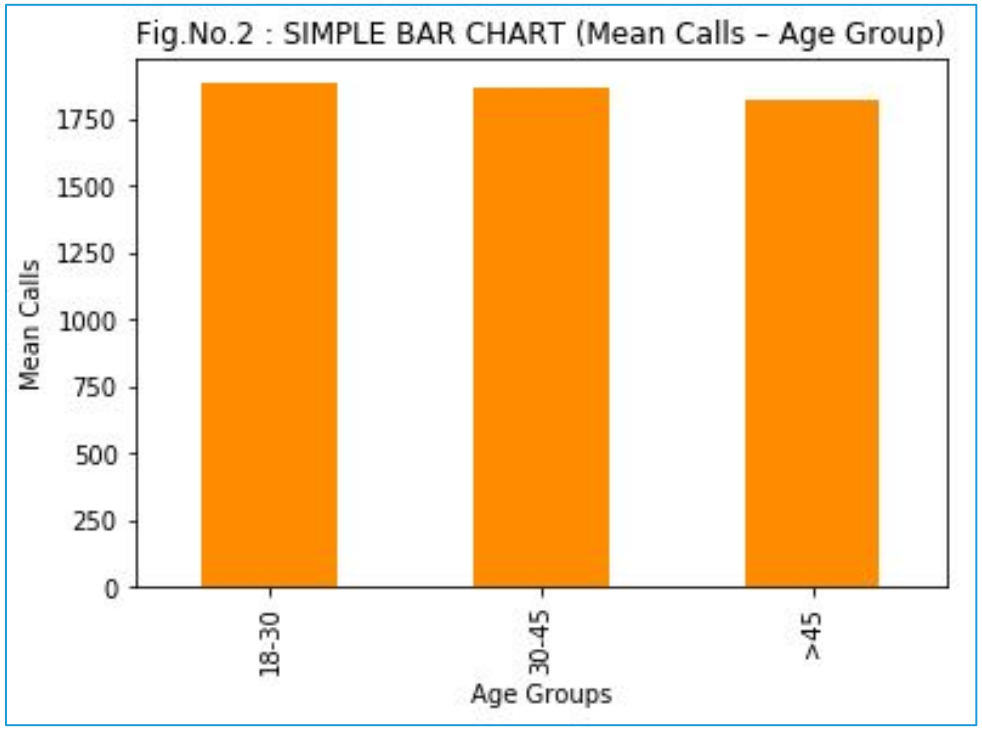
```
plt.figure(); telecom2.plot.bar(title='Fig.No.2 : SIMPLE BAR CHART (Mean Calls – Age Group)', color='darkorange'); plt.xlabel('Age Groups'); plt.ylabel('Mean Calls')
```

Note :

- The barplot code remains the same with respect to previous barplot code, the only difference is while aggregating the data.
- In previous plot aggregation function was **“sum”** & in this plot aggregation function is **“mean”**.

Simple Bar Chart in Python

This graph simply gives the distribution of the **Mean calls** across different **Age Groups**.



Interpretation :

- By plotting the average calls we can see that, though there is quite a difference in total calls in each age group, **the average number of calls across age groups is similar**

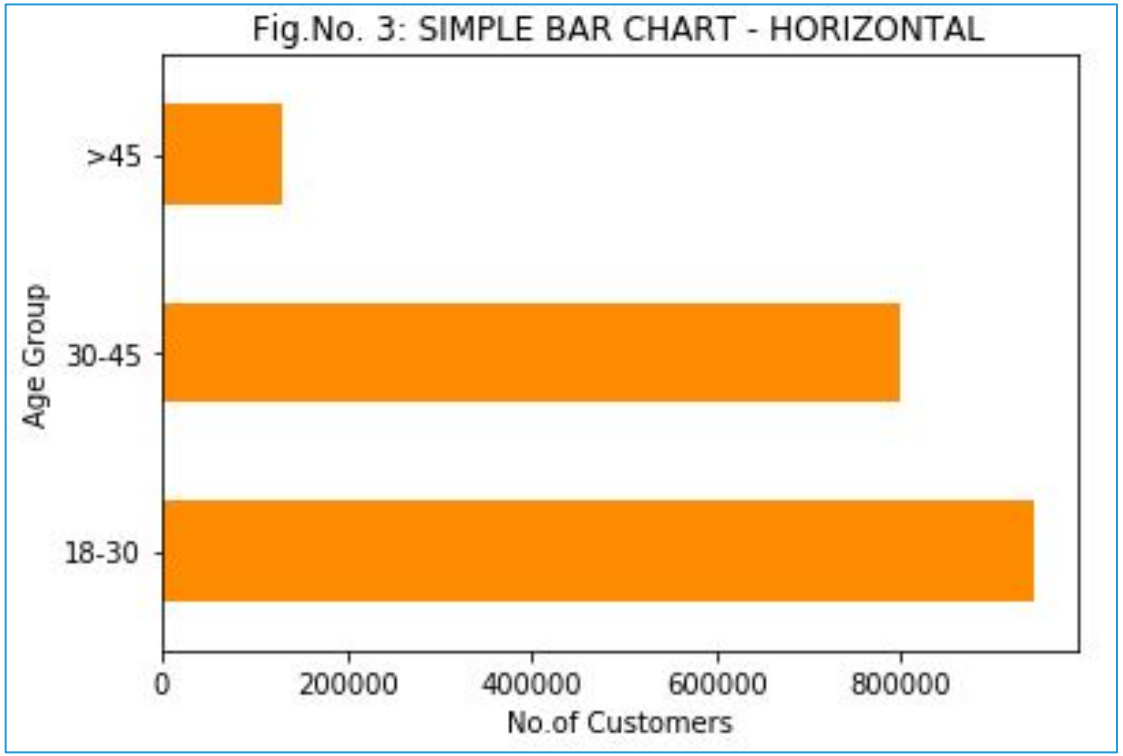
Simple Bar Chart in Python

```
# Simple Bar Chart in Horizontal orientation
plt.figure(); telecom1.plot.barh(title='Fig.No. 3: SIMPLE BAR CHART
- HORIZONTAL', color='darkorange'); plt.xlabel('No.of Customers');
plt.ylabel('Age Group')
```

- **barh()** gives horizontal orientation to the bars.

Simple Bar Chart in Python

This graph displays the number of customers across age group.



Interpretation :

- This is horizontal view of figure 1. Both these graphs are describing the same thing that, there are very few customers for age group >45 as compared to other two age groups.
- This graph is generally useful when there are negative frequency values in the data.

Stacked Bar Chart in Python

```
# Stacked Bar Chart
telecom3=pd.pivot_table(telecom, index=['Age_Group'], columns=['Gender'],
values=['CustID'], aggfunc='count')
telecom3
```

	CustID	
Gender	F	M
Age_Group		
18-30	256	245
30-45	221	207
>45	32	39

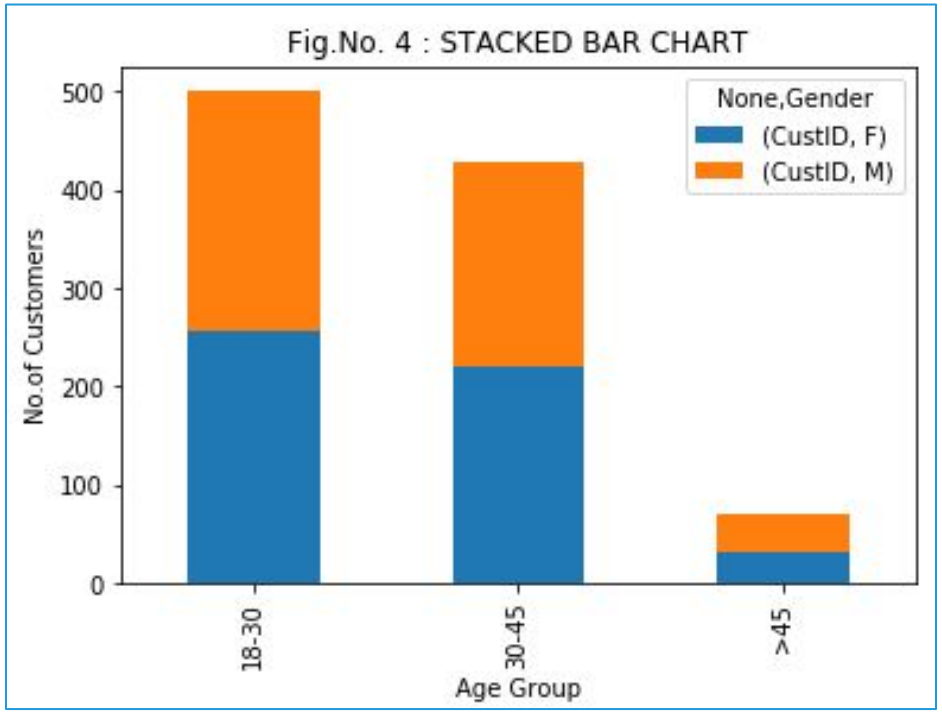
- **pivot_table()** reshapes the data and aggregates according to function specified. Here, we are aggregating the number of calls made by gender and age group.
- **index** is the column or array to group by on the x axes (pivot table rows).
- **columns** is the column or array to group by on the y axes (pivot table column).
- **values** is the column to aggregate
- **aggfunc** specifies a function to aggregate by.

```
plt.figure(); telecom3.plot.bar(title='Fig.No. 4 : STACKED BAR CHART',
stacked=True); plt.xlabel('Age Group'); plt.ylabel('No.of Customers')
```

- **Stacked** returns a stacked chart. Default is False.

Stacked Bar Chart in Python

This graph divides the number of customers in each age group by Gender.



Interpretation :

- This graph shows that, though there are more young customers in data but, almost equal number of Males and Females are present in each age group.

Percentage Bar Chart in Python

```
# Percentage Bar Chart
```

```
telecom4=telecom3.div(telecom3.sum(1).astype(float), axis=0)
telecom4
```

- **div()** creates percentage values by dividing the count data by column sum.

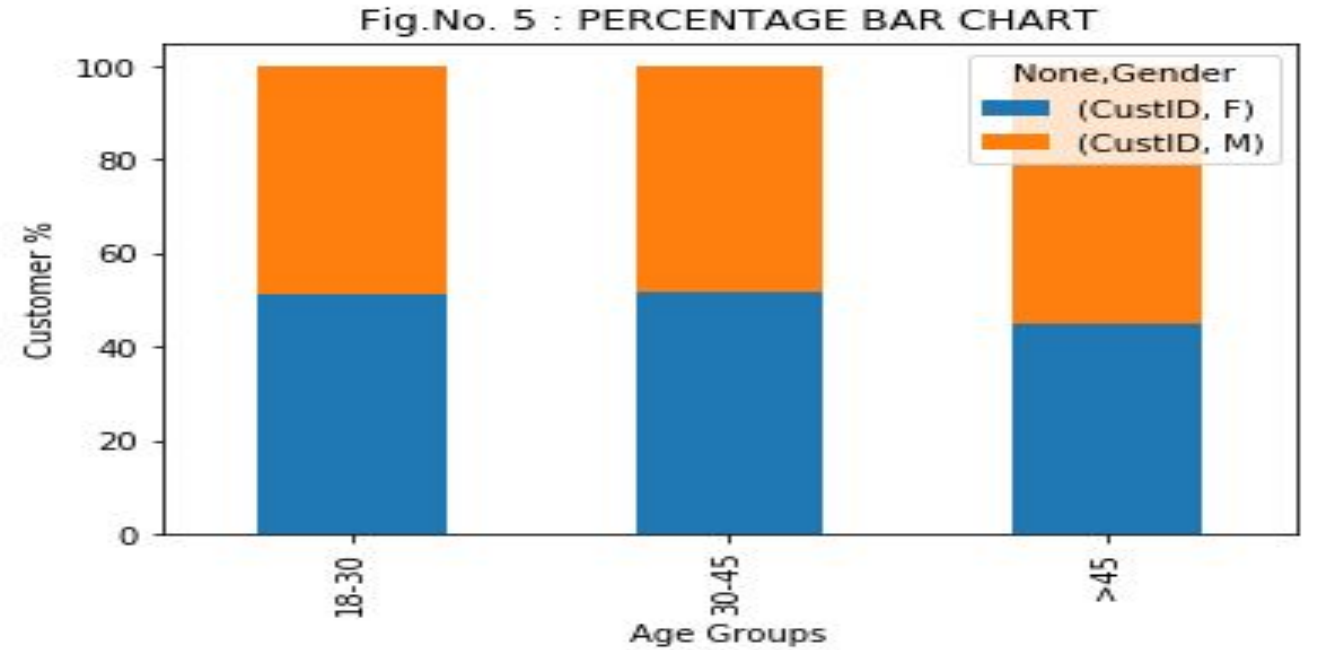
```
              CustID
Gender             F         M
Age_Group
18-30       0.510978  0.489022
30-45       0.516355  0.483645
>45         0.450704  0.549296
```

```
plt.figure();(telecom4*100).plot.bar(title='Fig.No. 5 : PERCENTAGE BAR
CHART', stacked=True); plt.xlabel('Age Groups'); plt.ylabel('Customer %')
```

- **telecom4*100** has to be a vector or matrix for which the bar chart needs to be plotted. *100 would display percentage scale on y-axis.

Percentage Bar Chart in Python

```
# Output for gender wise distribution of number of customers across the
# Age Groups.
```



Interpretation :

- Data contains almost equal proportion of Male and Female callers across three different age groups.
- Plotting a percentage stacked graph makes it efficient to compare the gender wise distribution of the number of customers across the Age Groups.

Multiple Bar Chart in Python

```
# Multiple Bar Chart
telecom5=pd.pivot_table(telecom, index=['Age_Group'], columns=['Gender'],
values=['Calls'], aggfunc='sum')
telecom5
```

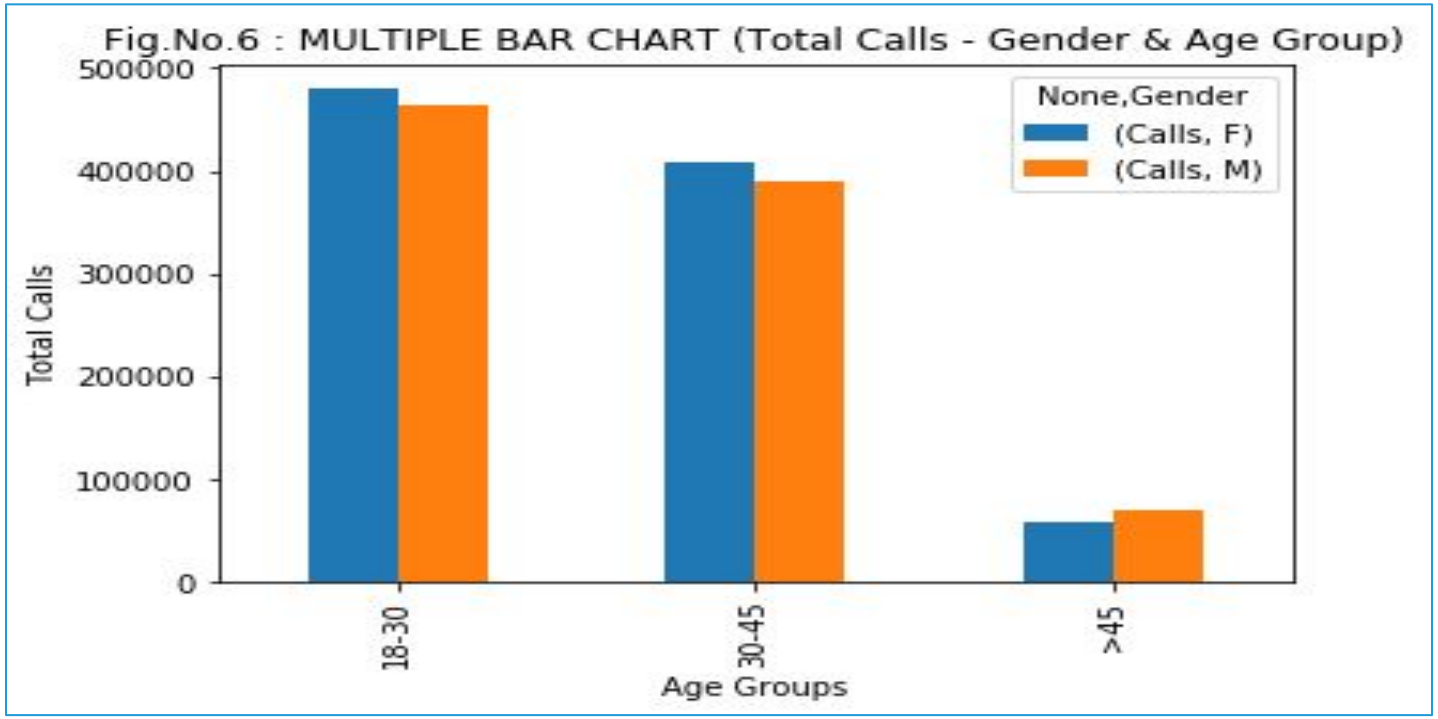
	Calls	
Gender	F	M
Age_Group		
18-30	480235	462952
30-45	408184	390537
>45	58310	70560

- **pivot_table()** is used to cross tabulate the categories of more than one variables using another numeric variable which results in total of each category

```
plt.figure(); telecom5.plot.bar(title='Fig.No.6 : MULTIPLE BAR CHART
(Total Calls - Gender & Age Group)'); plt.xlabel('Age Groups');
plt.ylabel('No. of Calls')
```

Multiple Bar Chart in Python

```
# Output for gender-wise distribution of number of calls across age groups
```



Interpretation :

- There is no significant difference between Male and Female in terms of number of calls made across three different age groups, the only difference is that, age group >45 has slightly more male customers than female customers as compared to other age groups.
- This can be used as an alternative way of representing a stacked bar graph.

Pie Chart in Python

```
# Pie Chart
telecom6 = telecom.groupby('Age_Group')['Calls'].sum()
telecom6 = telecom6.div(telecom6.sum().astype(float)).round(2)*100
telecom6
```

Here, we calculate the proportions for each category using **div()** function

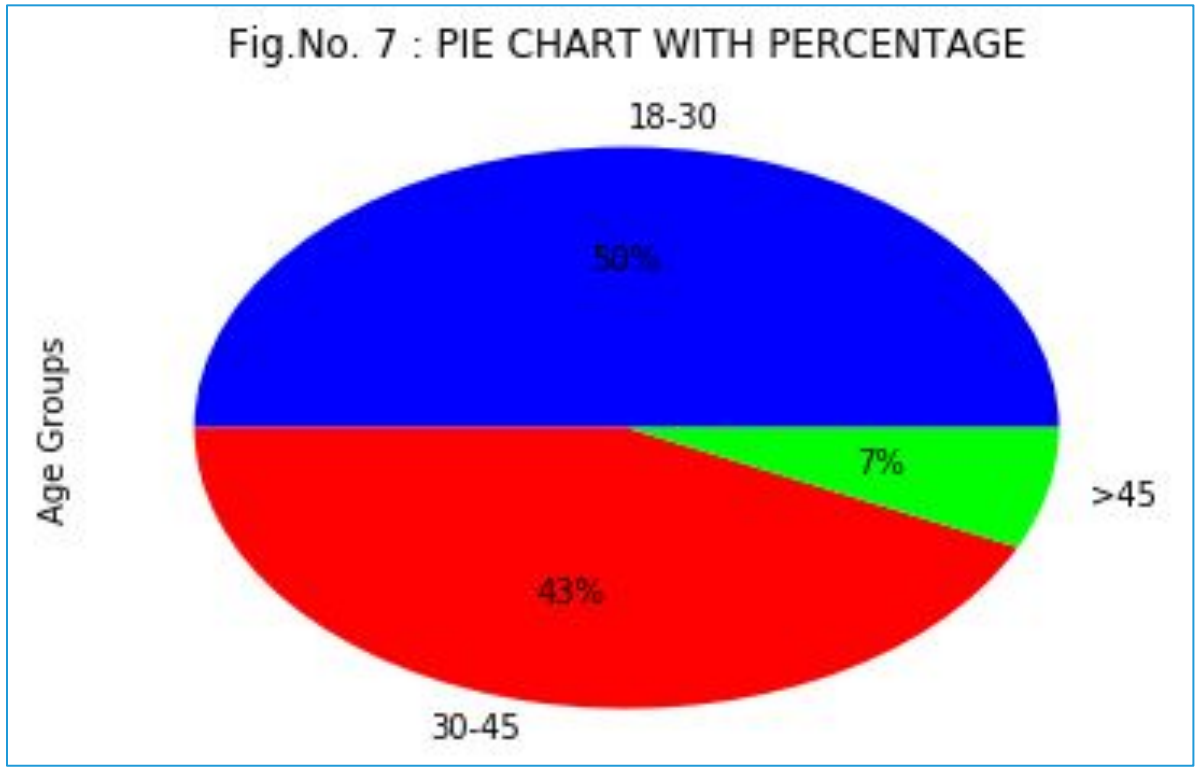
```
Age_Group
18-30    50.0
30-45    43.0
>45       7.0
```

```
telecom6.plot.pie(label=('Age Groups'), title = "Fig.No. 7 : PIE CHART WITH
PERCENTAGE",colormap='brg', autopct='%1.0f%%')
```

- **pie()** Used with plot create a pie chart
- **autopct** is used to display percentage values
- **labels=** provides a user defined label for the variable on X axis
- **title=** gives title of the plot
- **colormap=** can be used to input your choice of colors

Pie Chart in Python

```
# Output of Pie chart with percentage
```



Interpretation :

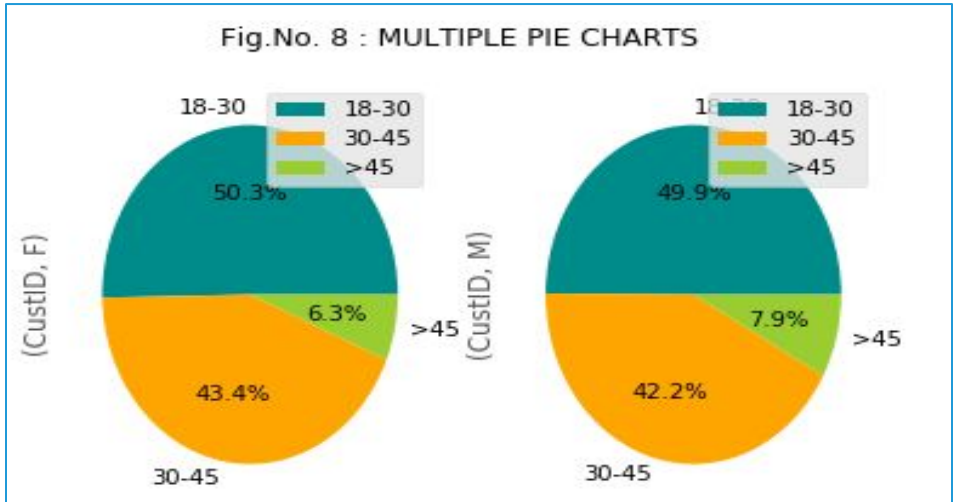
- **50%** of calls are made by Age_Group 18-30, **43%** by 30-45 & **only 7%** by >45 Age_Group.

Multiple Pie Chart in Python

```
#Pie Bar Chart – More than one
telecom7 = pd.pivot_table(telecom, index=['Age_Group'], columns=['Gender'],
values=['CustID'], aggfunc='count')
telecom7
```

	CustID	
Gender	F	M
Age_Group		
18-30	256	245
30-45	221	207
>45	32	39

```
plt.figure(); telecom7.plot.pie(title='Fig.No. 8 : MULTIPLE PIE CHARTS',
colors=['darkcyan','orange','yellowgreen'],autopct='%.1f%%', subplots=True)
```



- **subplots()** is default false, when 'True' plots multiple pie charts \

Get an Edge!

Important Principles of Data Visualisation

ACCENT is the principle of Data Visualization given for effective graphical display by D.A. Burn

Apprehension	Does the graph maximize the ability to correctly perceive relations among variables.?
Clarity	Is the graph able to visually distinguish all the elements of a graph and show the most important ones prominently?
Consistency	Are the elements, symbol shapes, and colors consistent with the previous graphs?
Efficiency	Is the graph able to portray complex relation in a simple and easy to interpret way?
Necessity	Is the graph more useful than the other ways to represent the data like a table/text?
Truthfulness	Are they accurately positioned and scaled such that the true values determinable by magnitude in terms of scale

Contents

About Data Visualisation

What is Data Visualisation?

It is the visual representation of data in the form of graphs and plots.

Why is it important?

It enables us to

- See the data and get insights in one glance
- Allows us to grasp difficult / complex data in an easy manner
- Helps us to identify patterns or trends easily. Also shows distribution, correlation and causality in data.

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

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

Simple Bar Diagram

A Bar Chart is the simplest and the most basic form of graph. In this graph, for each data item, we simply draw a ‘bar’ showing its value.

Simple Bar Chart: It is a type of chart which shows the values of different categories of data as rectangular bars with different lengths. The values are generally :

- Frequency
- Mean
- Totals
- Percentages

Diagrams in Python

```
#Importing the
Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

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

```
#Aggregating Data
telecom1 = telecom.groupby('Age_Group')['Calls'].sum()
telecom1
```

```
             Calls
Age_Group
18-30       943187
30-45       798721
>45         128870
```

For plotting a bar chart in Python, it is important to aggregate the data using **groupby()** to get required vector/matrix

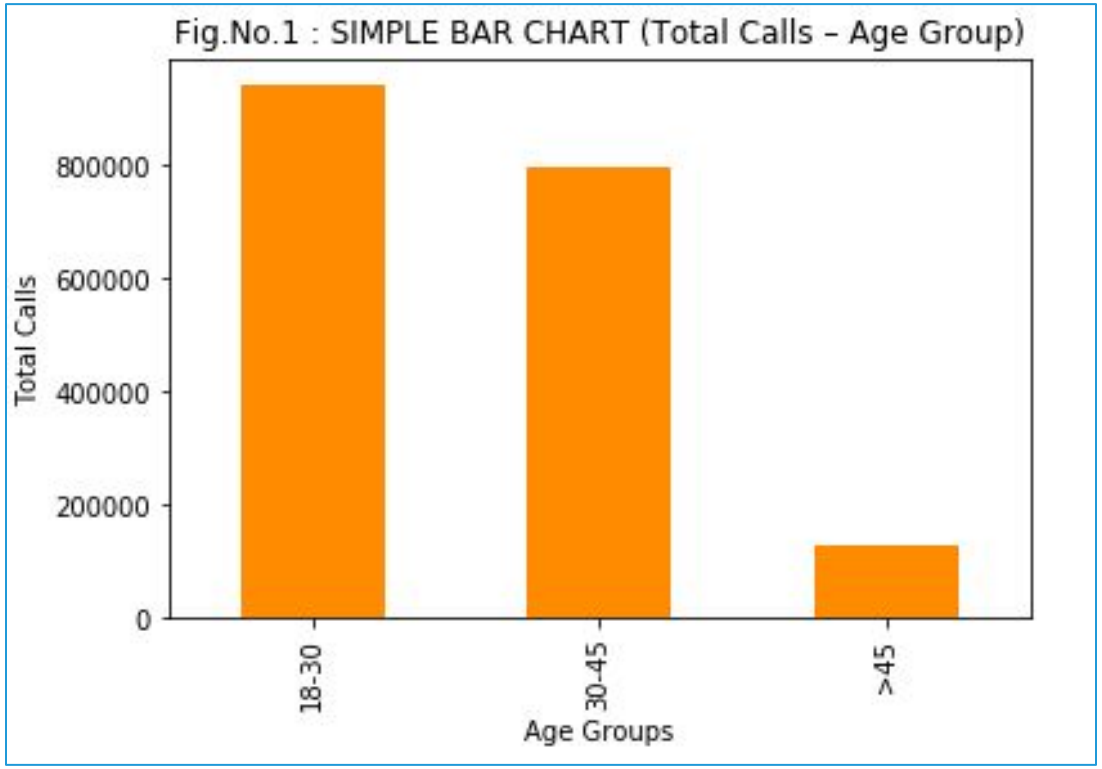
Simple Bar Chart in Python

```
#Simple Bar Chart – Total Calls for different Age Groups
plt.figure(); telecom1.plot.bar(title='Fig.No.1 : SIMPLE BAR CHART (Total Calls – Age Group)', color='darkorange'); plt.xlabel('Age Groups'); plt.ylabel('Total Calls')
```

- **plt.figure()** function is a convenient method to plot all columns with labels.
- **Plot.bar()** plots a bar chart. Can also be called by passing the argument **kind ='bar'** in plot.
- **title** is a string argument to give the plot a title.
- **color** argument specifies the plot colour. Accepts strings, hex numbers and colour code.
- **plt.xlabel** function/method to specify the x label.
- **plt.ylabel** function/method to specify the y label.

Simple Bar Chart in Python

This graph simply gives the distribution of the **Total number of calls** across different **Age Groups**.



Interpretation :

- Number of calls made by young age group (18-30) is slightly higher than mid age group (30-45) and very high than age group >45.

Simple Bar Chart in Python

```
# Simple Bar Chart – Mean Calls for different Age Groups
telecom2 = telecom.groupby('Age_Group')['Calls'].mean()
telecom2
```

```
                Calls
Age_Group
18-30      1882.608782
30-45      1866.170561
>45        1815.070423
```

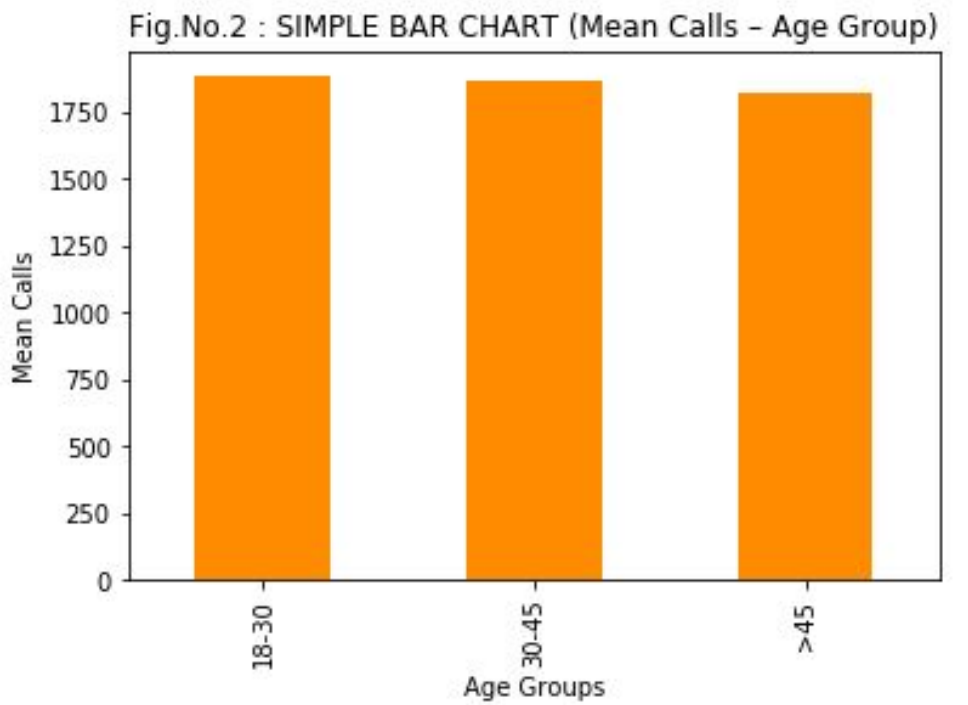
```
plt.figure(); telecom2.plot.bar(title='Fig.No.2 : SIMPLE BAR CHART (Mean
Calls – Age Group)', color='darkorange'); plt.xlabel('Age Groups');
plt.ylabel('Mean Calls')
```

Note :

- The barplot code remains the same with respect to previous barplot code, the only difference is while aggregating the data.
- In previous plot aggregation function was **“sum”** & in this plot aggregation function is **“mean”**.

Simple Bar Chart in Python

This graph simply gives the distribution of the **Mean calls** across different **Age Groups**.



Interpretation :

- By plotting the average calls we can see that, though there is quite a difference in total calls in each age group, **the average number of calls across age groups is similar**.

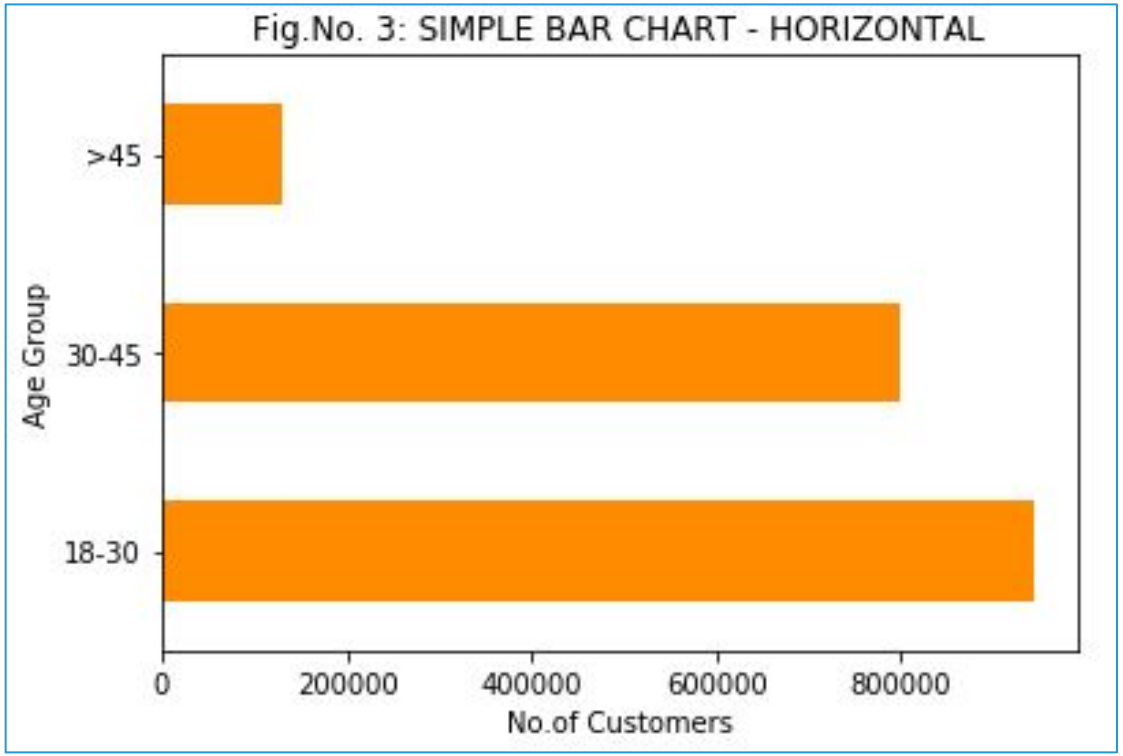
Simple Bar Chart in Python

```
# Simple Bar Chart in Horizontal orientation
plt.figure(); telecom1.plot.barh(title='Fig.No. 3: SIMPLE BAR CHART
- HORIZONTAL', color='darkorange'); plt.xlabel('No.of Customers');
plt.ylabel('Age Group')
```

- **barh()** gives horizontal orientation to the bars.

Simple Bar Chart in Python

This graph displays the number of customers across age group.



Interpretation :

- This is horizontal view of figure 1. Both these graphs are describing the same thing that, there are very few customers for age group >45 as compared to other two age groups.
- This graph is generally useful when there are negative frequency values in the data.

Stacked Bar Chart in Python

```
# Stacked Bar Chart
telecom3=pd.pivot_table(telecom, index=['Age_Group'], columns=['Gender'],
values=['CustID'], aggfunc='count')
telecom3
```

	CustID	
Gender	F	M
Age_Group		
18-30	256	245
30-45	221	207
>45	32	39

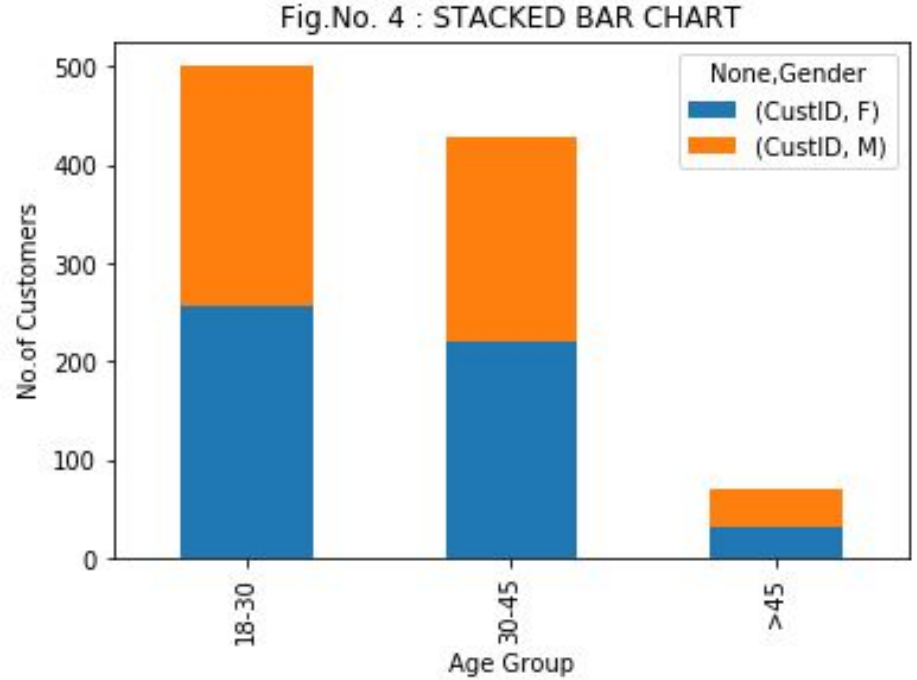
- **pivot_table()** reshapes the data and aggregates according to function specified. Here, we are aggregating the number of calls made by gender and age group.
- **index** is the column or array to group by on the x axes (pivot table rows).
- **columns** is the column or array to group by on the y axes (pivot table column).
- **values** is the column to aggregate
- **aggfunc** specifies a function to aggregate by.

```
plt.figure(); telecom3.plot.bar(title='Fig.No. 4 : STACKED BAR CHART',
stacked=True); plt.xlabel('Age Group'); plt.ylabel('No.of Customers')
```

- **Stacked** returns a stacked chart. Default is False.

Stacked Bar Chart in Python

This graph divides the number of customers in each age group by Gender.



Interpretation :

- This graph shows that, though there are more young customers in data but, almost equal number of Males and Females are present in each age group.

Percentage Bar Chart in Python

```
# Percentage Bar Chart
```

```
telecom4=telecom3.div(telecom3.sum(1).astype(float), axis=0)
telecom4
```

- **div()** creates percentage values by dividing the count data by column sum.

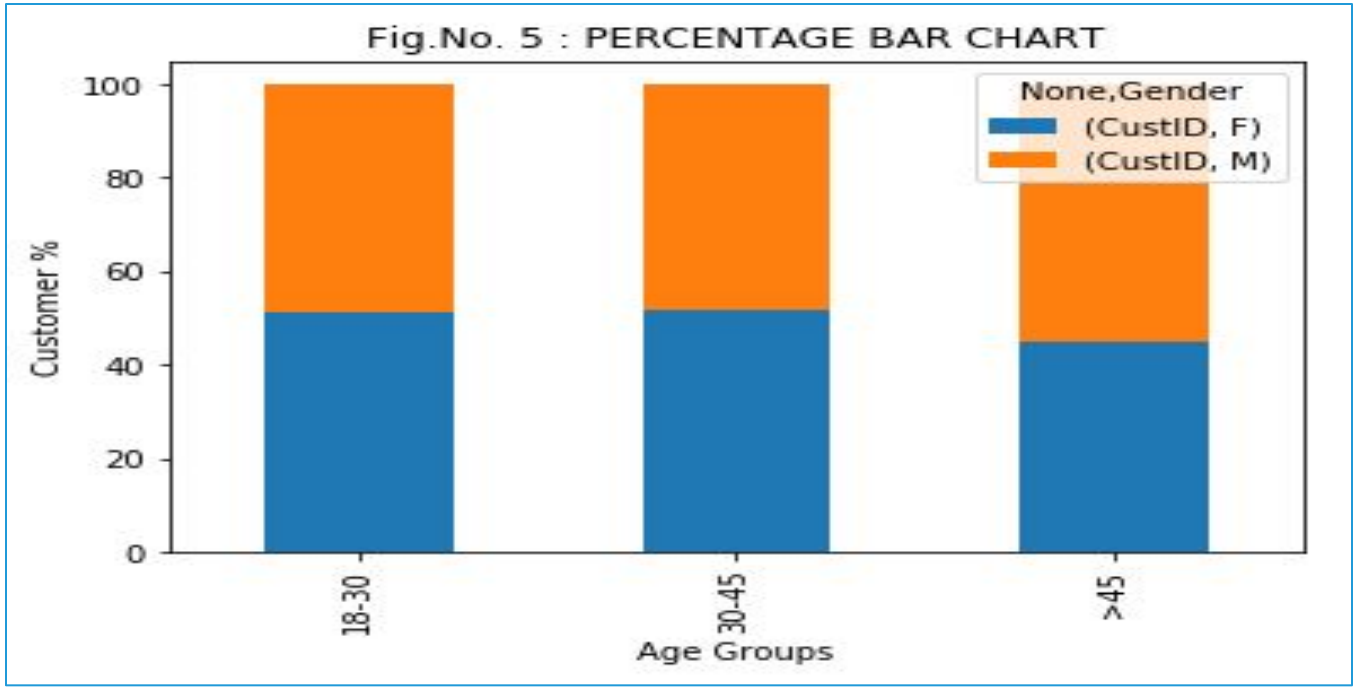
```
              CustID
Gender             F         M
Age_Group
18-30       0.510978  0.489022
30-45       0.516355  0.483645
>45         0.450704  0.549296
```

```
plt.figure();(telecom4*100).plot.bar(title='Fig.No. 5 : PERCENTAGE BAR
CHART', stacked=True); plt.xlabel('Age Groups'); plt.ylabel('Customer %')
```

- **telecom4*100** has to be a vector or matrix for which the bar chart needs to be plotted. *100 would display percentage scale on y-axis.

Percentage Bar Chart in Python

```
# Output for gender wise distribution of number of customers across the
# Age Groups.
```



Interpretation :

- Data contains almost equal proportion of Male and Female callers across three different age groups.
- Plotting a percentage stacked graph makes it efficient to compare the gender wise distribution of the number of customers across the Age Groups.

Multiple Bar Chart in Python

```
# Multiple Bar Chart
telecom5=pd.pivot_table(telecom, index=['Age_Group'], columns=['Gender'],
values=['Calls'], aggfunc='sum')
telecom5
```

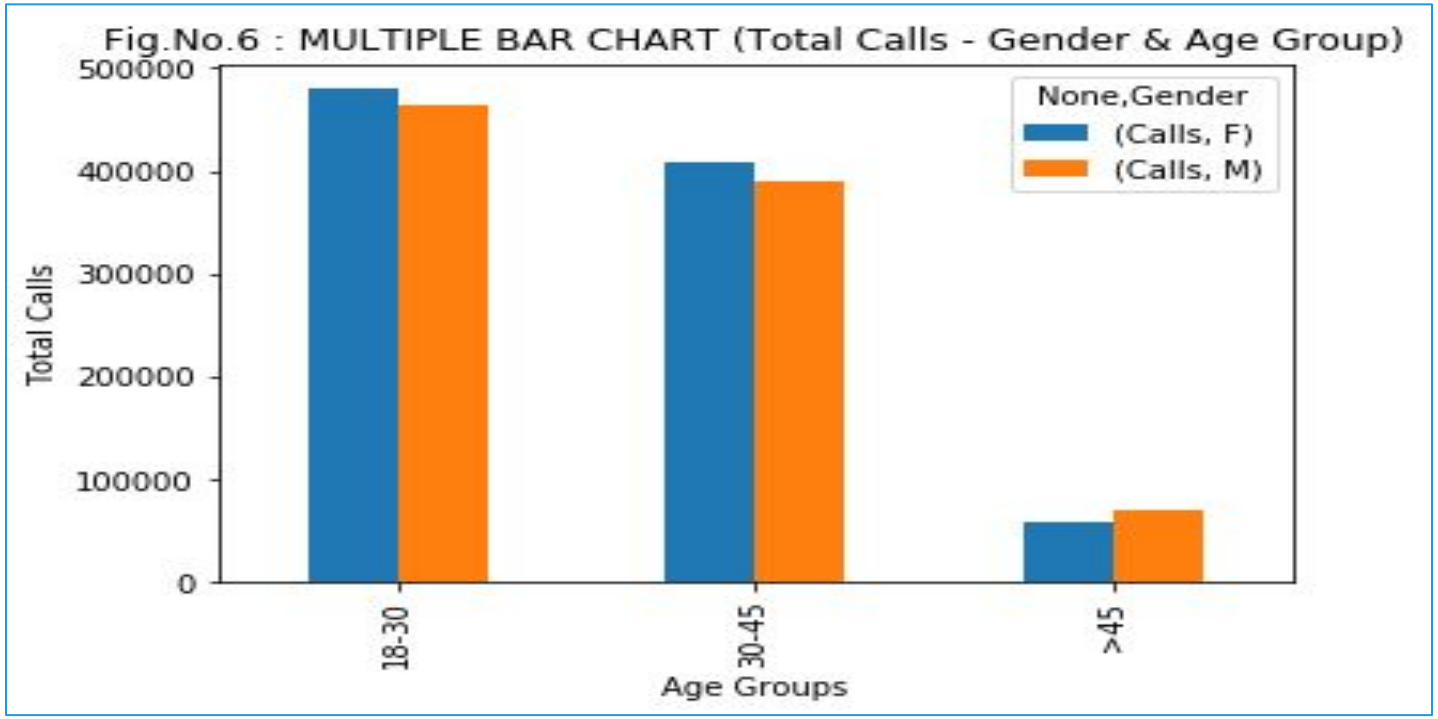
	Calls	
Gender	F	M
Age_Group		
18-30	480235	462952
30-45	408184	390537
>45	58310	70560

- **pivot_table()** is used to cross tabulate the categories of more than one variables using another numeric variable which results in total of each category

```
plt.figure(); telecom5.plot.bar(title='Fig.No.6 : MULTIPLE BAR CHART
(Total Calls - Gender & Age Group)'); plt.xlabel('Age Groups');
plt.ylabel('No. of Calls')
```

Multiple Bar Chart in Python

```
# Output for gender-wise distribution of number of calls across age groups
```



Interpretation :

- There is no significant difference between Male and Female in terms of number of calls made across three different age groups, the only difference is that, age group >45 has slightly more male customers than female customers as compared to other age groups.
- This can be used as an alternative way of representing a stacked bar graph.

Pie Chart in Python

```
# Pie Chart
telecom6 = telecom.groupby('Age_Group')['Calls'].sum()
telecom6 = telecom6.div(telecom6.sum().astype(float)).round(2)*100
telecom6
```

Here, we calculate the proportions for each category using **div()** function

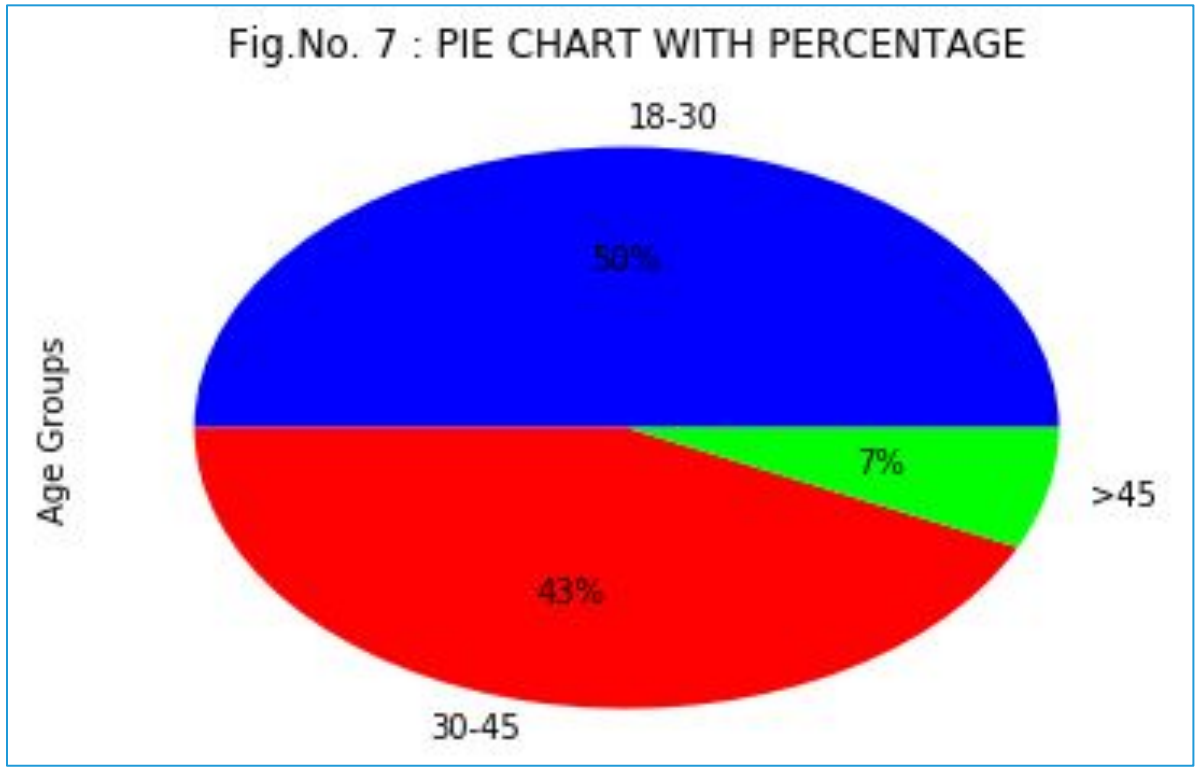
```
Age_Group
18-30    50.0
30-45    43.0
>45       7.0
```

```
telecom6.plot.pie(label=('Age Groups'), title = "Fig.No. 7 : PIE CHART WITH
PERCENTAGE",colormap='brg', autopct='%1.0f%%')
```

- **pie()** Used with plot create a pie chart
- **autopct** is used to display percentage values
- **labels=** provides a user defined label for the variable on X axis
- **title=** gives title of the plot
- **colormap=** can be used to input your choice of colors

Pie Chart in Python

```
# Output of Pie chart with percentage
```



Interpretation :

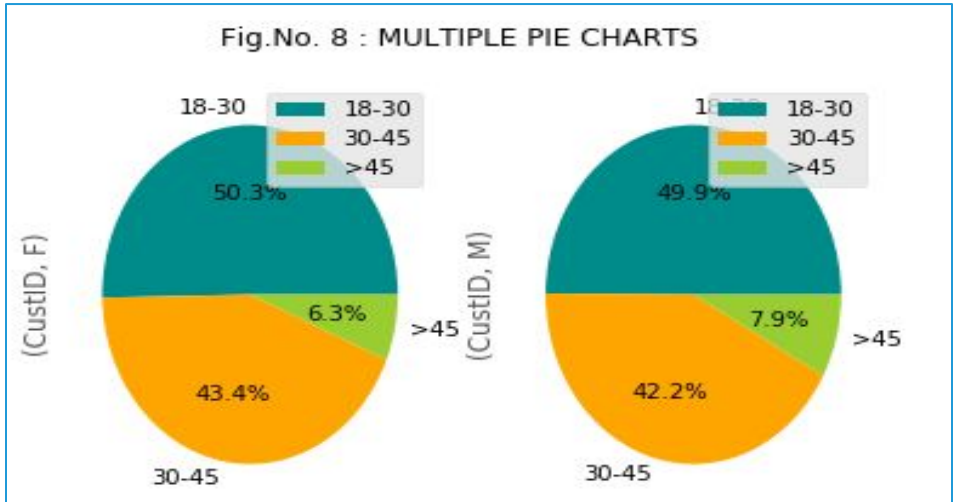
- **50%** of calls are made by Age_Group 18-30, **43%** by 30-45 & **only 7%** by >45 Age_Group.

Multiple Pie Chart in Python

```
#Pie Bar Chart – More than one
telecom7 = pd.pivot_table(telecom, index=['Age_Group'], columns=['Gender'],
values=['CustID'], aggfunc='count')
telecom7
```

	CustID	
Gender	F	M
Age_Group		
18-30	256	245
30-45	221	207
>45	32	39

```
plt.figure(); telecom7.plot.pie(title='Fig.No. 8 : MULTIPLE PIE CHARTS',
colors=['darkcyan','orange','yellowgreen'],autopct='%.1f%%', subplots=True)
```



- **subplots()** is default false, when 'True' plots multiple pie charts

Get an Edge!

Important Principles of Data Visualisation

ACCENT is the principle of Data Visualization given for effective graphical display by D.A. Burn

Apprehension	Does the graph maximize the ability to correctly perceive relations among variables.?
Clarity	Is the graph able to visually distinguish all the elements of a graph and show the most important ones prominently?
Consistency	Are the elements, symbol shapes, and colors consistent with the previous graphs?
Efficiency	Is the graph able to portray complex relation in a simple and easy to interpret way?
Necessity	Is the graph more useful than the other ways to represent the data like a table/text?
Truthfulness	Are they accurately positioned and scaled such that the true values determinable by magnitude in terms of scale

Quick Recap

In this session, we learnt data visualisation using basic graphs

Chart Types and Functions in Python	• Bar Diagrams - `plot.bar()` • Pie Chart - `plot.pie()`