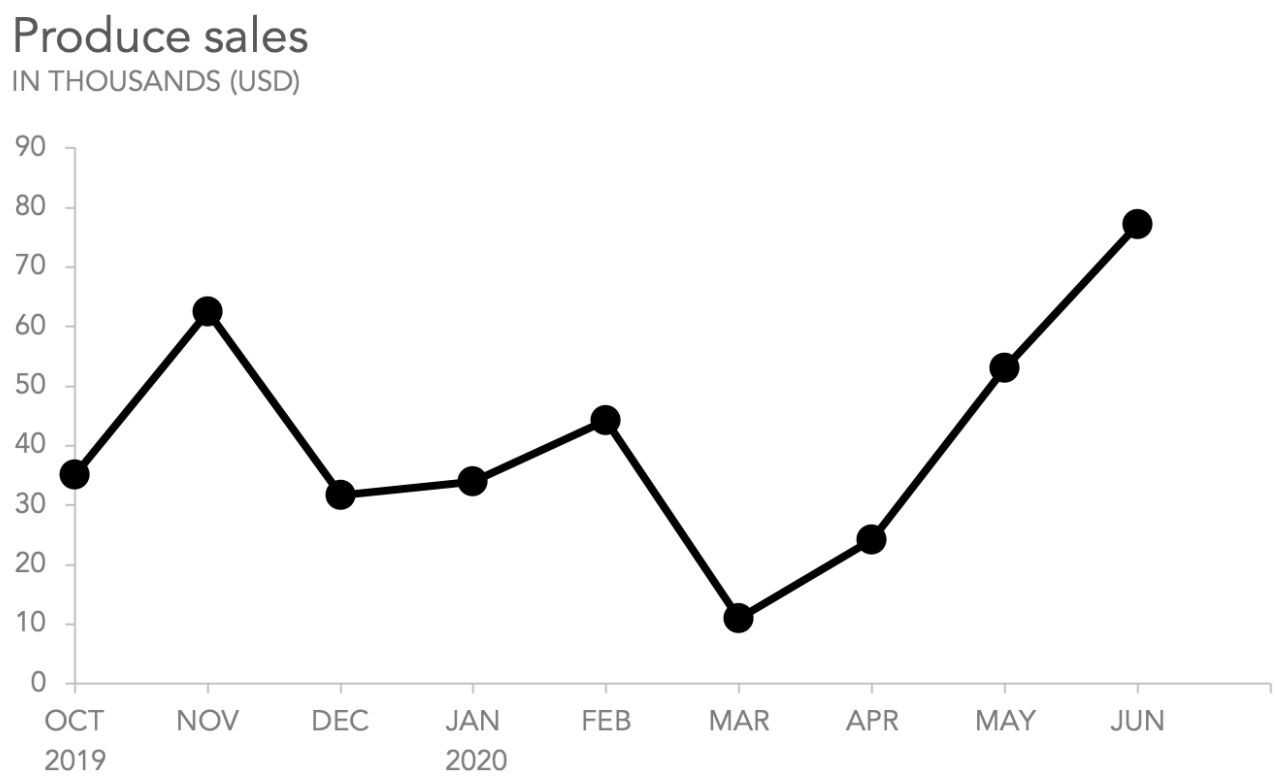
Types of Data

- Numerical Data
- Categorical Data

```
# import the library
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

plt.style.use('default')
```

2D Line plot



- Bivariate Analysis
- categorical -> numerical and numerical -> numerical
- Use case - Time series data

```
# plotting a simple function
price = [48000,54000,57000,49000,47000,45000]
year = [2015,2016,2017,2018,2019,2020]

plt.plot(year,price)
```

```
# from a pandas dataframe
batsman = pd.read_csv('/content/sharma-kohli.csv')
batsman
```

```
plt.plot(batsman['index'],batsman['V Kohli'])
```

```
# plotting multiple plots
plt.plot(batsman['index'],batsman['V Kohli'])
plt.plot(batsman['index'],batsman['RG Sharma'])
```

```
# labels title
plt.plot(batsman['index'],batsman['V Kohli'])
plt.plot(batsman['index'],batsman['RG Sharma'])

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')
```

```
# colors(hex) and line(width and style) and marker(size)
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F')
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6')

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')
```

```
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F',linestyle='solid',linewidth=3)
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6',linestyle='dashdot',linewidth=2)

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')
```

```
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F',linestyle='solid',linewidth=3,marker='D',markersize=10)
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6',linestyle='dashdot',linewidth=2,marker='o')

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')
```

```
# legend -> location
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F',linestyle='solid',linewidth=3,marker='D',markersize=10,label='V
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6',linestyle='dashdot',linewidth=2,marker='o',label='Rohit')

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')

plt.legend(loc='upper right')
```

```
# limiting axes
price = [48000,54000,57000,49000,47000,45000,4500000]
year = [2015,2016,2017,2018,2019,2020,2021]

plt.plot(year,price)
plt.ylim(0,75000)
plt.xlim(2017,2019)
```

```
# grid
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F',linestyle='solid',linewidth=3,marker='D',markersize=10)
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6',linestyle='dashdot',linewidth=2,marker='o')

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')

plt.grid()
```

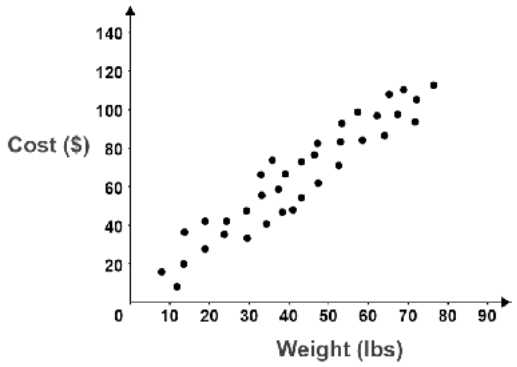
```
# show
plt.plot(batsman['index'],batsman['V Kohli'],color='#D9F10F',linestyle='solid',linewidth=3,marker='D',markersize=10)
plt.plot(batsman['index'],batsman['RG Sharma'],color='#FC00D6',linestyle='dashdot',linewidth=2,marker='o')

plt.title('Rohit Sharma Vs Virat Kohli Career Comparison')
plt.xlabel('Season')
plt.ylabel('Runs Scored')

plt.grid()

plt.show()
```

Scatter Plots



- Bivariate Analysis
- numerical vs numerical
- Use case - Finding correlation

```
# plt.scatter simple function
x = np.linspace(-10,10,50)

y = 10*x + 3 + np.random.randint(0,300,50)
y
```

```
array([199.        ,  70.08163265,  13.16326531,  25.24489796,
       198.32653061,  40.40816327, -64.51020408, 206.57142857,
       -60.34693878, -28.26530612,  23.81632653,  29.89795918,
         6.97959184, 166.06122449, 136.14285714, 156.2244898 ,
        -8.69387755, 204.3877551 ,  66.46938776,  85.55102041,
       203.63265306, 182.71428571, 139.79591837, 164.87755102,
        67.95918367,  57.04081633, 190.12244898,  51.20408163,
       101.28571429,  84.36734694,  31.44897959,  47.53061224,
       223.6122449 , 145.69387755, 278.7755102 , 122.85714286,
       258.93877551, 174.02040816, 315.10204082, 338.18367347,
       363.26530612, 242.34693878, 342.42857143, 376.51020408,
        98.59183673, 376.67346939,  95.75510204, 268.83673469,
       309.91836735, 324.        ])
```

```
plt.scatter(x,y)
```

```
# plt.scatter on pandas data
df = pd.read_csv('/content/batter.csv')
df = df.head(50)
df
```

```
plt.scatter(df['avg'],df['strike_rate'],color='red',marker='+')
plt.title('Avg and SR analysis of Top 50 Batsman')
plt.xlabel('Average')
plt.ylabel('SR')
```

```
# marker
```

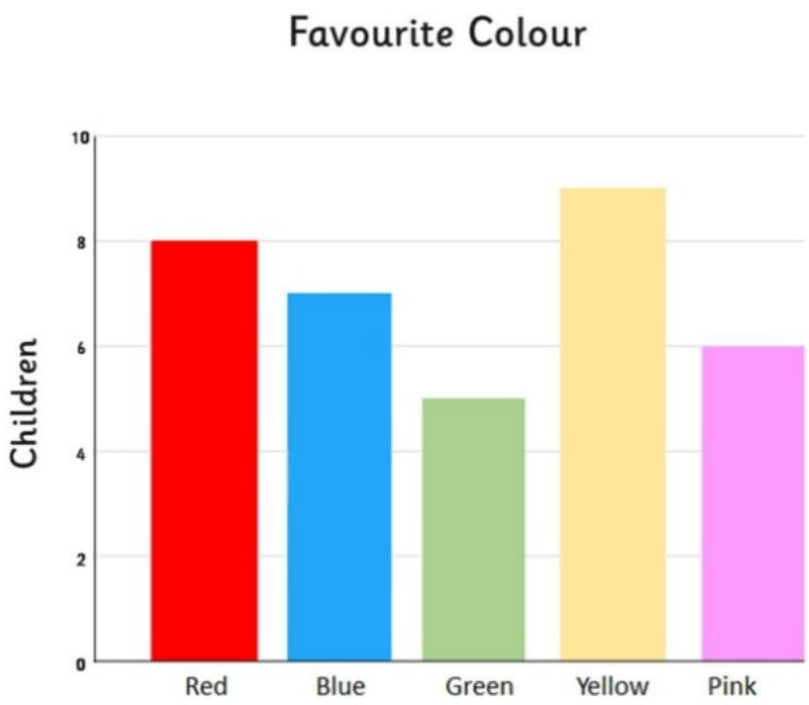
```
# size
tips = sns.load_dataset('tips')
```

```
# slower
plt.scatter(tips['total_bill'],tips['tip'],s=tips['size']*20)
```

```
# scatterplot using plt.plot
# faster
plt.plot(tips['total_bill'],tips['tip'],'o')
```

```
# plt.plot vs plt.scatter
```

Bar chart



- Bivariate Analysis
- Numerical vs Categorical
- Use case - Aggregate analysis of groups

```
# simple bar chart
children = [10,20,40,10,30]
colors = ['red','blue','green','yellow','pink']

plt.bar(colors,children,color='black')
```

```
# bar chart using data
```

```
# horizontal bar chart
plt.barh(colors,children,color='black')
```

```
# color and label
df = pd.read_csv('/content/batsman_season_record.csv')
df
```

```
plt.bar(np.arange(df.shape[0]) - 0.2,df['2015'],width=0.2,color='yellow')
plt.bar(np.arange(df.shape[0]),df['2016'],width=0.2,color='red')
plt.bar(np.arange(df.shape[0]) + 0.2,df['2017'],width=0.2,color='blue')

plt.xticks(np.arange(df.shape[0]), df['batsman'])

plt.show()
```

```
np.arange(df.shape[0])
```

```
array([0, 1, 2, 3, 4])
```

```
# Multiple Bar charts
```

```
# xticks
```

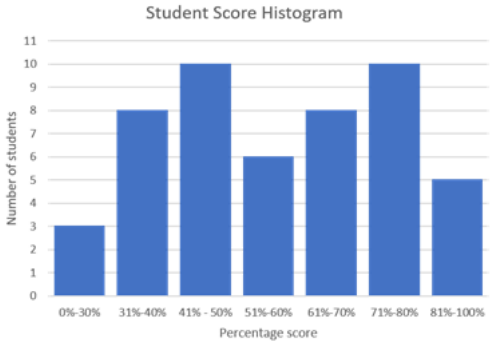
```
# a problem
children = [10,20,40,10,30]
colors = ['red red red red red red','blue blue blue blue','green green green green green','yellow yellow yellow yellow ','pi

plt.bar(colors,children,color='black')
plt.xticks(rotation='vertical')
```

```
# Stacked Bar chart
plt.bar(df['batsman'],df['2017'],label='2017')
plt.bar(df['batsman'],df['2016'],bottom=df['2017'],label='2016')
plt.bar(df['batsman'],df['2015'],bottom=(df['2016'] + df['2017']),label='2015')

plt.legend()
plt.show()
```

Histogram



- Univariate Analysis
- Numerical col
- Use case - Frequency Count

```
# simple data

data = [32,45,56,10,15,27,61]

plt.hist(data,bins=[10,25,40,55,70])
```

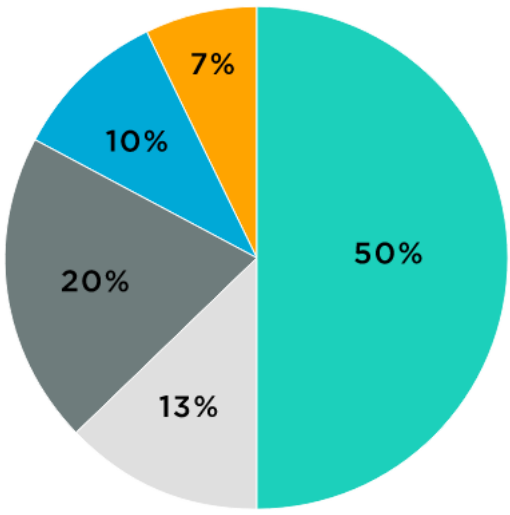
```
# on some data
df = pd.read_csv('/content/vk.csv')
df
```

```
plt.hist(df['batsman_runs'],bins=[0,10,20,30,40,50,60,70,80,90,100,110,120])
plt.show()
```

```
# handling bins


# logarithmic scale
arr = np.load('/content/big-array.npy')
plt.hist(arr,bins=[10,20,30,40,50,60,70],log=True)
plt.show()
```

Pie Chart



- Univariate/Bivariate Analysis
- Categorical vs numerical
- Use case - To find contibution on a standard scale

```
# simple data
data = [23,45,100,20,49]
subjects = ['eng','science','maths','sst','hindi']
plt.pie(data,labels=subjects)

plt.show()
```

```
# dataset
df = pd.read_csv('/content/gayle-175.csv')
df
```

```
plt.pie(df['batsman_runs'],labels=df['batsman'],autopct='%0.1f%%')
plt.show()
```

```
# percentage and colors
plt.pie(df['batsman_runs'],labels=df['batsman'],autopct='%0.1f%%',colors=['blue','green','yellow','pink','cyan','brown'])
plt.show()
```

```
# explode shadow
plt.pie(df['batsman_runs'],labels=df['batsman'],autopct='%0.1f%%',explode=[0.3,0,0,0,0,0.1],shadow=True)
plt.show()
```

Changing styles

```
plt.style.available
```

```
['Solarize_Light2',
 '_classic_test_patch',
 'bmh',
 'classic',
 'dark_background',
 'fast',
 'fivethirtyeight',
 'ggplot',
 'grayscale',
 'seaborn',
 'seaborn-bright',
 'seaborn-colorblind',
 'seaborn-dark',
 'seaborn-dark-palette',
 'seaborn-darkgrid',
 'seaborn-deep',
 'seaborn-muted',
 'seaborn-notebook',
 'seaborn-paper',
 'seaborn-pastel',
 'seaborn-poster',
 'seaborn-talk',
 'seaborn-ticks',
 'seaborn-white',
 'seaborn-whitegrid',
 'tableau-colorblind10']
```

```
plt.style.use('dark_background')
```

```
arr = np.load('/content/big-array.npy')
plt.hist(arr,bins=[10,20,30,40,50,60,70],log=True)
plt.show()
```

Save figure

```
arr = np.load('/content/big-array.npy')
plt.hist(arr,bins=[10,20,30,40,50,60,70],log=True)

plt.savefig('sample.png')
```