Matplotlib

Load Necessary Librariesm

In [100…

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

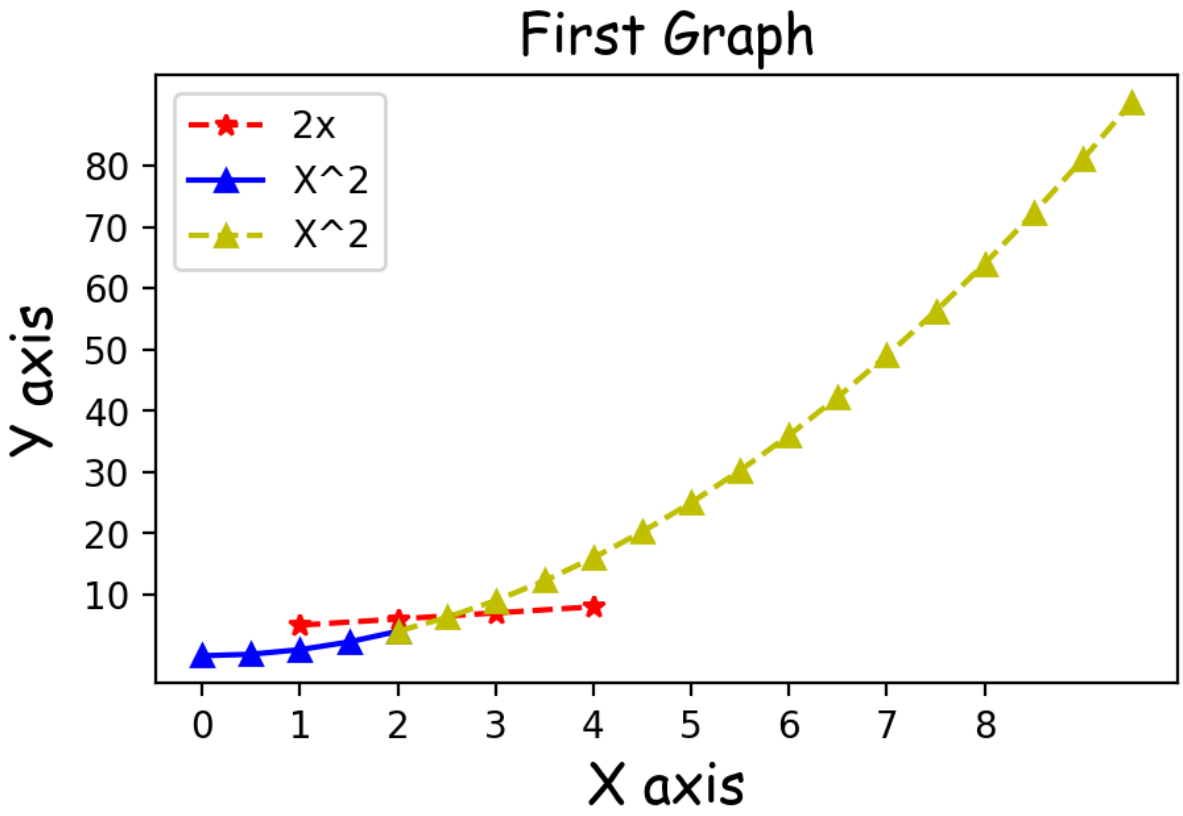
Basic Graph

In [102…

```
x = [1,2,3,4]
y = [5,6,7,8]

plt.figure(figsize=(5,3),dpi=200) # Plot size with Dots per inch or pixels per inch

#Keyword Argument notation
#plt.plot(x,y, label='2x',color='blue',linewidth = 3,marker='.',markersize = 20,mar

#Use shorthand notation of color,marker,linestyle
#fmt = '[color][marker][line]'
plt.plot(x,y,'r*--' ,label='2x') # General plot with short hand notation

#Select intervals we want to plot
x2 = np.arange(0,10.0,0.5)
print(x2)

#Plot part of the graph as line
plt.plot(x2[:5],x2[:5]**2,'b^-',label='X^2')

#Plot remainder of graph as dotted line
plt.plot(x2[4:],x2[4:]**2,'y^--',label='X^2')

#Adding a title (specify font parameters with fontdict)
plt.title("First Graph",fontdict={'fontname':'Comic Sans MS','fontsize':15}) #Title

#Adding x y labels
plt.xlabel('X axis',fontdict={'fontname':'Comic Sans MS','fontsize':15}) #xlabel of
plt.ylabel('Y axis',fontdict={'fontname':'Comic Sans MS','fontsize':15}) #ylabel of

#scaling the graph
plt.xticks([0,1,2,3,4,5,6,7,8]) # Points on x axis
plt.yticks([10,20,30,40,50,60,70,80]) # Points on y axis

#Add a legend
plt.legend()

#Saving a graph (dpi 300 is good enough which has high resolution)
plt.savefig('Graphs/Firstgraph.png',dpi=300)
```

```
plt.show() #To remove this line [<matplotlib.lines.Line2D at 0x22ae67da650>]
```

```
[0.  0.5 1.  1.5 2.  2.5 3.  3.5 4.  4.5 5.  5.5 6.  6.5 7.  7.5 8.  8.5
 9.  9.5]
```



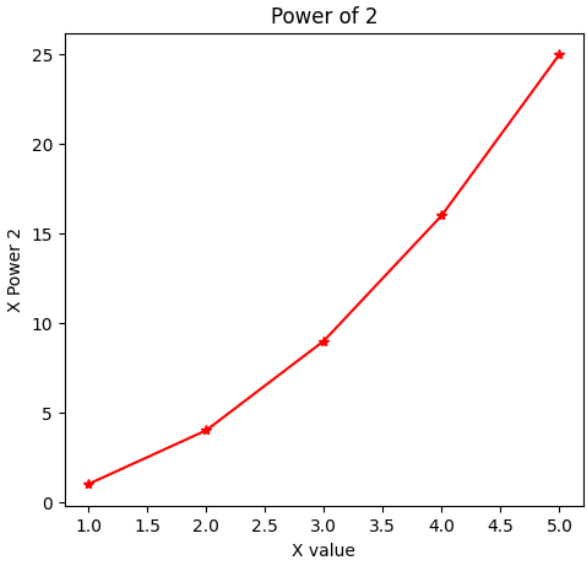
Plotting plot side by side

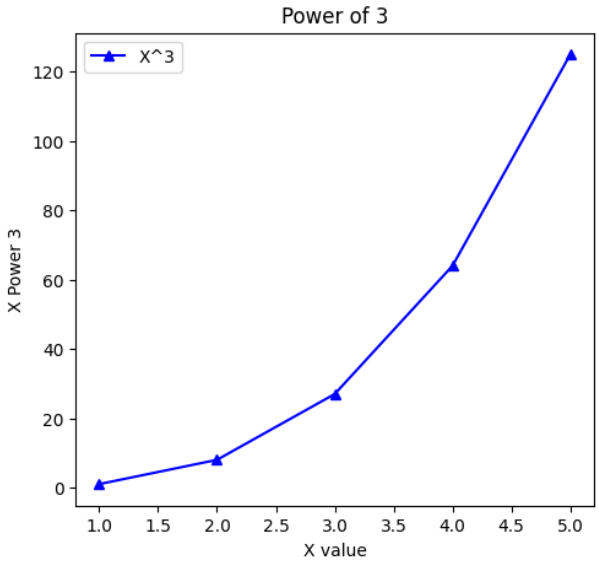
In [103…

```
fig, axes = plt.subplots(1,2,figsize=(12,5))

x = np.array([1,2,3,4,5])

axes[0].plot(x,x**2,'r*-',label='X^2')
axes[0].set_title("Power of 2")
axes[0].set_xlabel('X value')
axes[0].set_ylabel('X Power 2')


axes[1].plot(x,x**3,'b^-',label='X^3')
axes[1].set_title("Power of 3")
axes[1].set_xlabel('X value')
axes[1].set_ylabel('X Power 3')

plt.savefig('Graphs/Powerof3.png',dpi=300)

plt.legend()
```

Out[103…

```
<matplotlib.legend.Legend at 0x22af1be72e0>
```





Real World Examples

Data used here is gas_prices.csv

Line Graphm

In [106…
```
gas = pd.read_csv('data/gas_prices.csv')
```

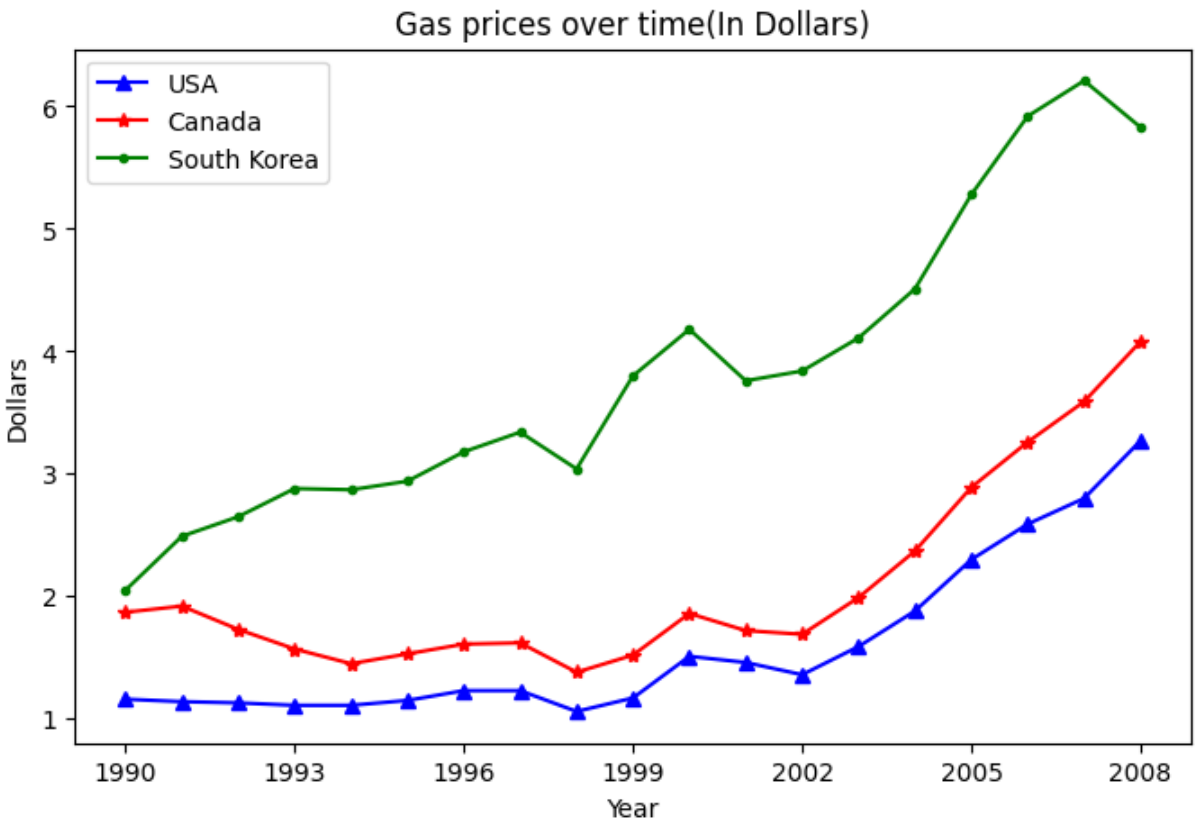
In [108…
```
gas # Gas prices of Different countries from year 1990 to 2008
```

Out[108...

	Year	Australia	Canada	France	Germany	Italy	Japan	Mexico	South Korea	UK	USA
0	1990	NaN	1.87	3.63	2.65	4.59	3.16	1.00	2.05	2.82	1.16
1	1991	1.96	1.92	3.45	2.90	4.50	3.46	1.30	2.49	3.01	1.14
2	1992	1.89	1.73	3.56	3.27	4.53	3.58	1.50	2.65	3.06	1.13
3	1993	1.73	1.57	3.41	3.07	3.68	4.16	1.56	2.88	2.84	1.11
4	1994	1.84	1.45	3.59	3.52	3.70	4.36	1.48	2.87	2.99	1.11
5	1995	1.95	1.53	4.26	3.96	4.00	4.43	1.11	2.94	3.21	1.15
6	1996	2.12	1.61	4.41	3.94	4.39	3.64	1.25	3.18	3.34	1.23
7	1997	2.05	1.62	4.00	3.53	4.07	3.26	1.47	3.34	3.83	1.23
8	1998	1.63	1.38	3.87	3.34	3.84	2.82	1.49	3.04	4.06	1.06
9	1999	1.72	1.52	3.85	3.42	3.87	3.27	1.79	3.80	4.29	1.17
10	2000	1.94	1.86	3.80	3.45	3.77	3.65	2.01	4.18	4.58	1.51
11	2001	1.71	1.72	3.51	3.40	3.57	3.27	2.20	3.76	4.13	1.46
12	2002	1.76	1.69	3.62	3.67	3.74	3.15	2.24	3.84	4.16	1.36
13	2003	2.19	1.99	4.35	4.59	4.53	3.47	2.04	4.11	4.70	1.59
14	2004	2.72	2.37	4.99	5.24	5.29	3.93	2.03	4.51	5.56	1.88
15	2005	3.23	2.89	5.46	5.66	5.74	4.28	2.22	5.28	5.97	2.30
16	2006	3.54	3.26	5.88	6.03	6.10	4.47	2.31	5.92	6.36	2.59
17	2007	3.85	3.59	6.60	6.88	6.73	4.49	2.40	6.21	7.13	2.80
18	2008	4.45	4.08	7.51	7.75	7.63	5.74	2.45	5.83	7.42	3.27

In [133...

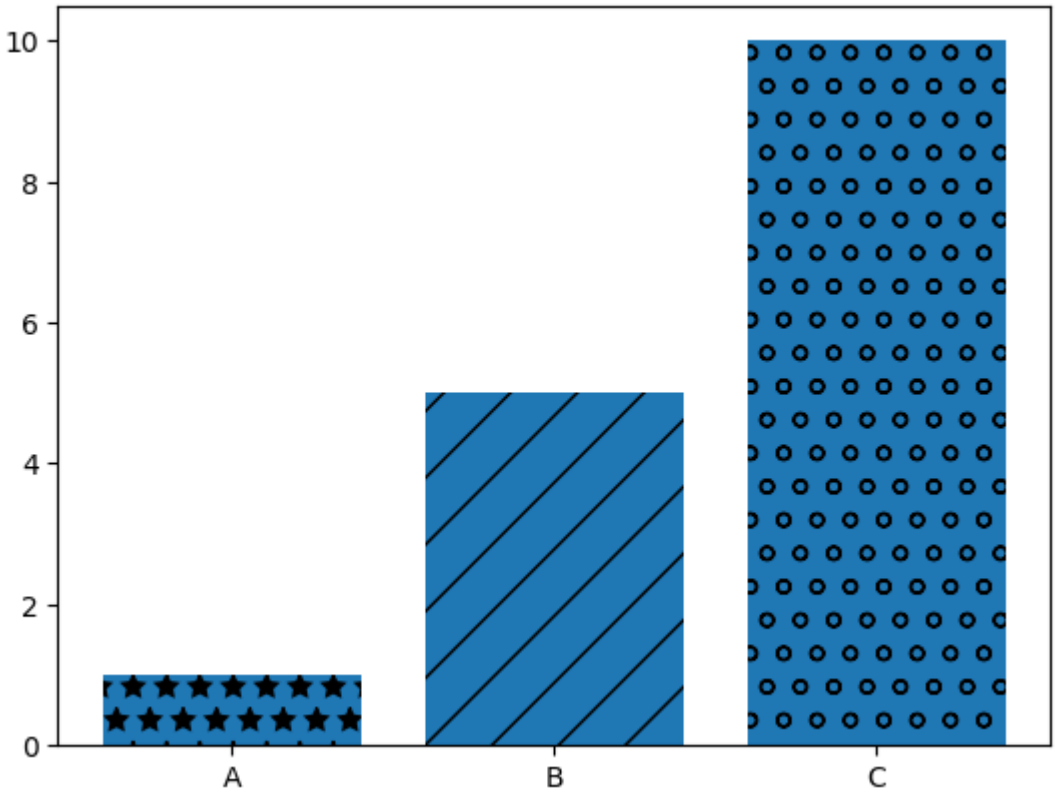
```
plt.figure(figsize=(8,5))
plt.title('Gas prices over time(In Dollars)')
plt.plot(gas.Year,gas.USA,'b^-',label='USA')
plt.plot(gas.Year,gas.Canada,'r*-',label='Canada')
plt.plot(gas.Year,gas['South Korea'],'g.-',label='South Korea')
# for country in gas:
#     if country != 'Year':
#         plt.plot(gas.Year,gas[country],marker='.',label = country)
plt.xlabel('Year')
plt.ylabel('Dollars')
plt.xticks(gas.Year[::3])
plt.legend()
plt.savefig('Graphs/lineplot_gasprices.png',dpi=300)
plt.show()
```



Bar chat

In [137…

```
labels = ['A','B','C']
values = [1,5,10]

#Adding patterns inside barplots
bars = plt.bar(labels,values)
patterns = ['*','/','o']
for bar in bars:
    bar.set_hatch(patterns.pop(0))

# bars[0].set_hatch('/')
# bars[1].set_hatch('*')

plt.savefig('Graphs/barplot.png',dpi=300)
plt.show()
```



Histograms

Frequency data

In [185…

```
fifa = pd.read_csv('data/fifa_data.csv')
fifa.head(5)
```

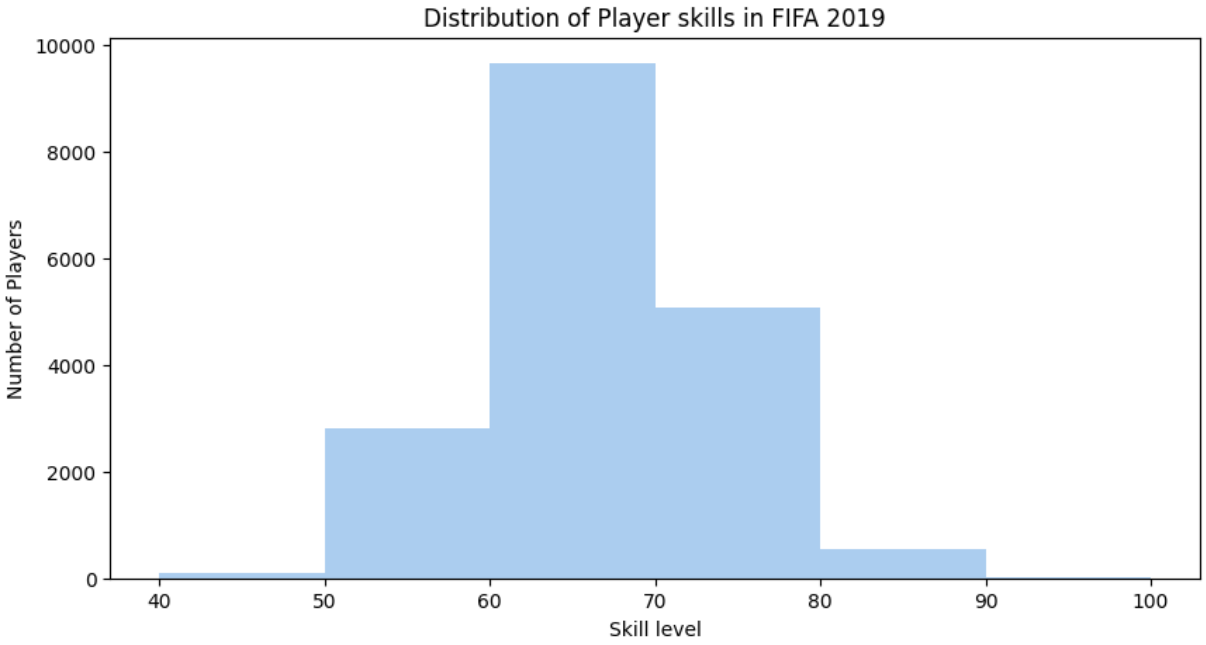
Out[185…

	Unnamed: 0	ID	Name	Age	Photo	Nationali
0	0	158023	L. Messi	31	https://cdn.sofifa.org/players/4/19/158023.png	Argentir
1	1	20801	Cristiano Ronaldo	33	https://cdn.sofifa.org/players/4/19/20801.png	Portug
2	2	190871	Neymar Jr	26	https://cdn.sofifa.org/players/4/19/190871.png	Braz
3	3	193080	De Gea	27	https://cdn.sofifa.org/players/4/19/193080.png	Spa
4	4	192985	K. De Bruyne	27	https://cdn.sofifa.org/players/4/19/192985.png	Belgiu

5 rows × 89 columns

In [154…

```
plt.figure(figsize=(10,5))

bins = [40,50,60,70,80,90,100]
plt.title("Distribution of Player skills in FIFA 2019")

plt.hist(fifa.Overall,bins = bins,color='#abcdef')
plt.xlabel('Skill level')
plt.ylabel('Number of Players')
plt.xticks(bins)
plt.savefig('Graphs/playerskillsdis.png',dpi=300)
plt.show()
```

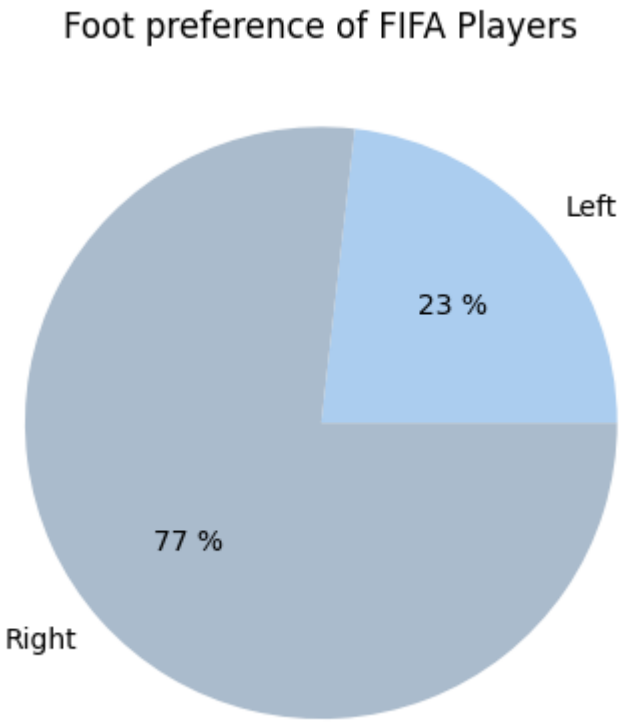


Pie Charts

In [166…

```
left = fifa.loc[fifa['Preferred Foot']=='Left'].count()[0]
right = fifa.loc[fifa['Preferred Foot']=='Right'].count()[0]
print(left,right)

labels = ['Left','Right']
colors = ['#abcdef','#aabbcc']
plt.pie([left,right],labels=labels,colors=colors,autopct = '%.0f %%')

plt.title("Foot preference of FIFA Players")

plt.savefig('Graphs/FootPereference.png',dpi=300)

plt.show()
```

```
C:\Users\saite\AppData\Local\Temp\ipykernel_22320\2967012679.py:1: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
  left = fifa.loc[fifa['Preferred Foot']=='Left'].count()[0]
C:\Users\saite\AppData\Local\Temp\ipykernel_22320\2967012679.py:2: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will always be treated as labels (consistent with DataFrame behavior). To access a value by position, use `ser.iloc[pos]`
  right = fifa.loc[fifa['Preferred Foot']=='Right'].count()[0]
4211 13948
```



In [186…
```
fifa.Weight = [int(x.strip('lbs')) if type(x) == str else x for x in fifa.Weight]
```

In [187…
```
fifa.Weight
```

Out[187…
```
0        159.0
1        183.0
2        150.0
3        168.0
4        154.0
         ...
18202    134.0
18203    170.0
18204    148.0
18205    154.0
18206    176.0
Name: Weight, Length: 18207, dtype: float64
```

In [188…
```
light = fifa.loc[fifa.Weight < 125].count()[0]
light_medium = fifa.loc[(fifa.Weight >= 125) & (fifa.Weight < 150)].count()[0]
```

```
medium = fifa.loc[(fifa.Weight >= 150) & (fifa.Weight < 175)].count()[0]
medium_heavy = fifa.loc[(fifa.Weight >= 175) & (fifa.Weight < 200)].count()[0]
heavy = fifa.loc[fifa.Weight >= 200].count()[0]
```

```
C:\Users\saite\AppData\Local\Temp\ipykernel_22320\3158887285.py:1: FutureWarning: Se
ries.__getitem__ treating keys as positions is deprecated. In a future version, inte
ger keys will always be treated as labels (consistent with DataFrame behavior). To a
ccess a value by position, use `ser.iloc[pos]`
  light = fifa.loc[fifa.Weight < 125].count()[0]
C:\Users\saite\AppData\Local\Temp\ipykernel_22320\3158887285.py:2: FutureWarning: Se
ries.__getitem__ treating keys as positions is deprecated. In a future version, inte
ger keys will always be treated as labels (consistent with DataFrame behavior). To a
ccess a value by position, use `ser.iloc[pos]`
  light_medium = fifa.loc[(fifa.Weight >= 125) & (fifa.Weight < 150)].count()[0]
C:\Users\saite\AppData\Local\Temp\ipykernel_22320\3158887285.py:3: FutureWarning: Se
ries.__getitem__ treating keys as positions is deprecated. In a future version, inte
ger keys will always be treated as labels (consistent with DataFrame behavior). To a
ccess a value by position, use `ser.iloc[pos]`
  medium = fifa.loc[(fifa.Weight >= 150) & (fifa.Weight < 175)].count()[0]
C:\Users\saite\AppData\Local\Temp\ipykernel_22320\3158887285.py:4: FutureWarning: Se
ries.__getitem__ treating keys as positions is deprecated. In a future version, inte
ger keys will always be treated as labels (consistent with DataFrame behavior). To a
ccess a value by position, use `ser.iloc[pos]`
  medium_heavy = fifa.loc[(fifa.Weight >= 175) & (fifa.Weight < 200)].count()[0]
C:\Users\saite\AppData\Local\Temp\ipykernel_22320\3158887285.py:5: FutureWarning: Se
ries.__getitem__ treating keys as positions is deprecated. In a future version, inte
ger keys will always be treated as labels (consistent with DataFrame behavior). To a
ccess a value by position, use `ser.iloc[pos]`
  heavy = fifa.loc[fifa.Weight >= 200].count()[0]
```

In [190…

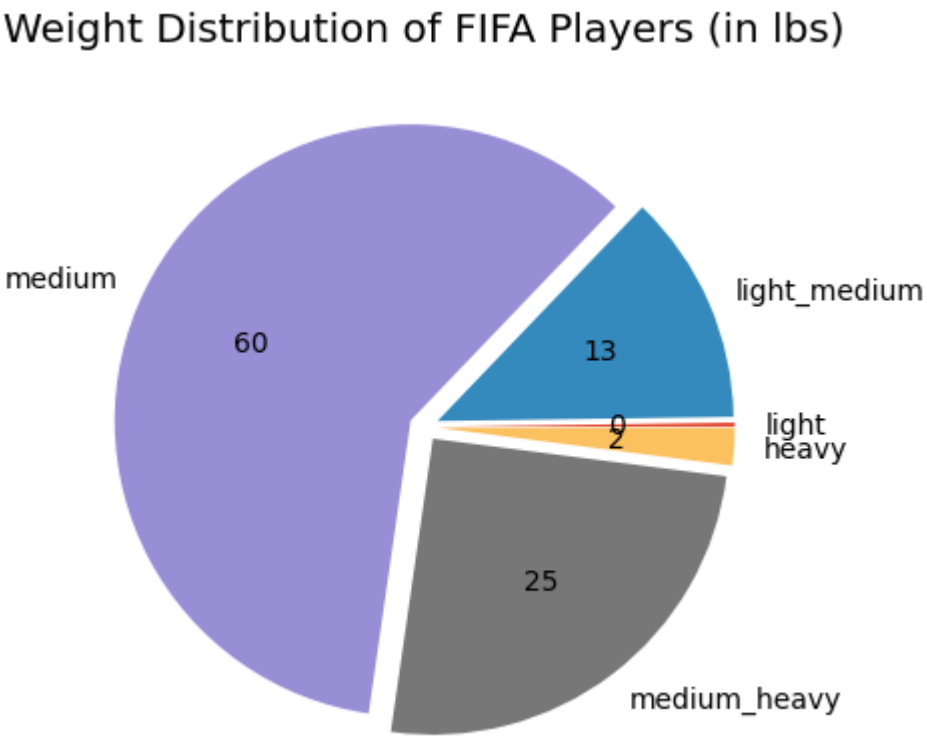
```
medium_heavy
```

Out[190…

```
4583
```

In [199…

```
plt.style.use('ggplot')
labels = ['light','light_medium','medium','medium_heavy','heavy']
weights = [light,light_medium,medium,medium_heavy,heavy]
explode = (0.05,0.05,0.05,0.05,0.05)
plt.title("Weight Distribution of FIFA Players (in lbs)")
plt.pie(weights,labels=labels,autopct='%.f',explode = explode)
plt.savefig('Graphs/Piechart_weights_fifa.png',dpi = 300)
plt.show()
```



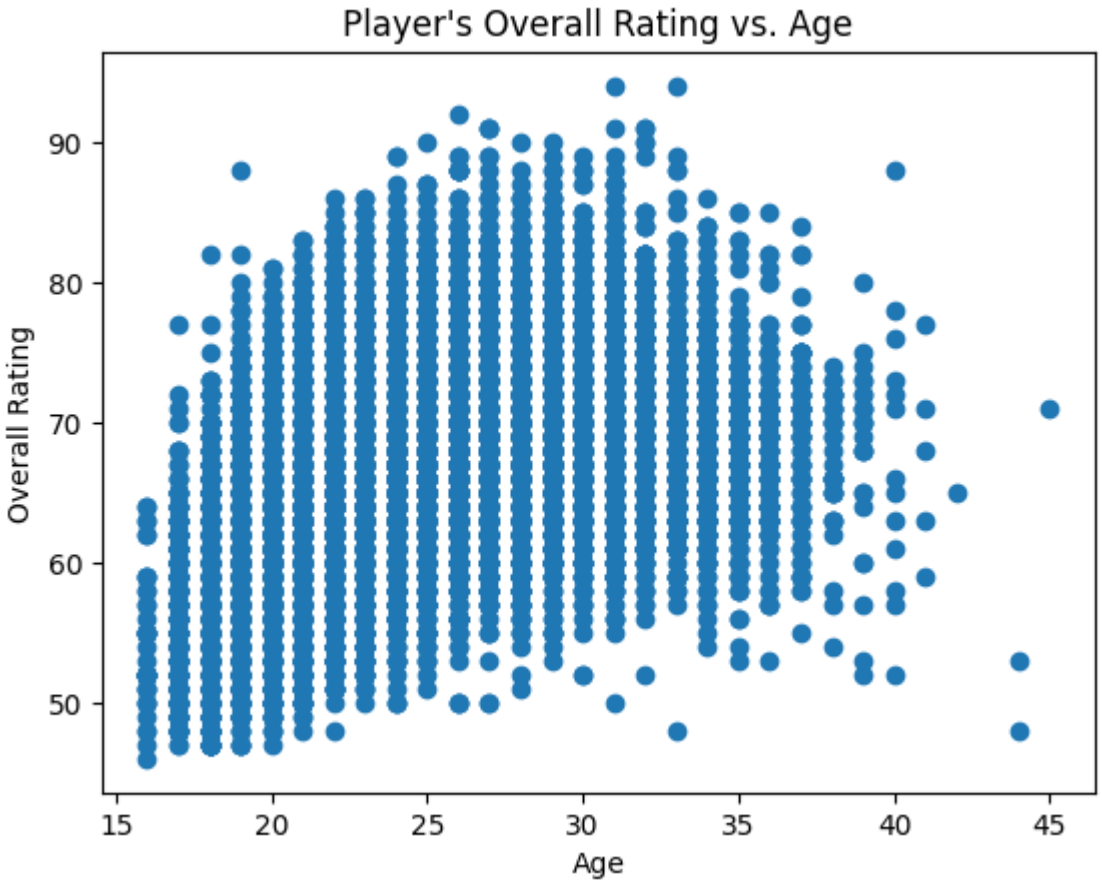
Scatter plot

Relationships between two numeric variables

In [223…
```
price_fcb = fifa.loc[fifa.Club =='FC Barcelona']['Value']
price_rm = fifa.loc[fifa.Club =='Real Madrid']['Value']
```

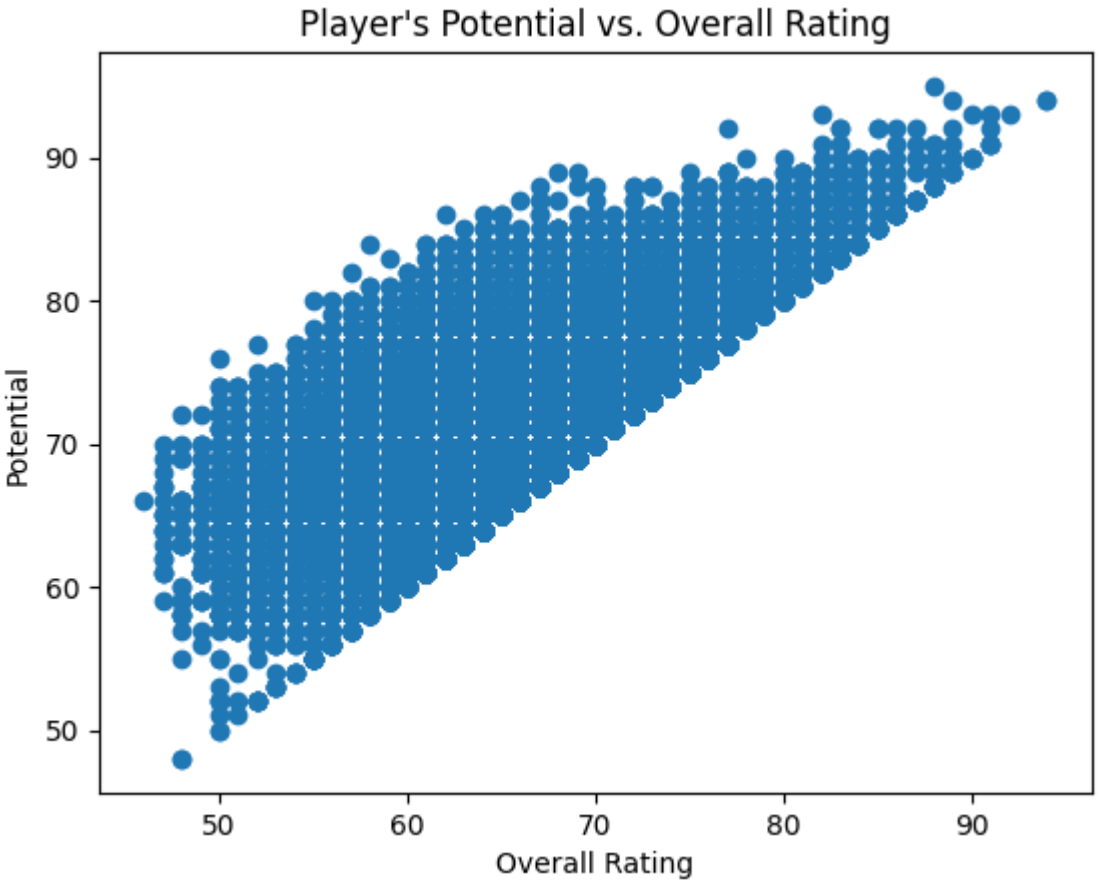
In [224…
```
barcelona = fifa.loc[fifa.Club == 'FC Barcelona']['Overall']
madrid = fifa.loc[fifa.Club == 'Real Madrid']['Overall']
```

In [227…
```
plt.scatter(fifa['Age'], fifa['Overall'])
plt.xlabel('Age')
plt.ylabel('Overall Rating')
plt.title('Player\'s Overall Rating vs. Age')
plt.show()
```



In [228…

```
plt.scatter(fifa['Overall'], fifa['Potential'])
plt.xlabel('Overall Rating')
plt.ylabel('Potential')
plt.title('Player\'s Potential vs. Overall Rating')
plt.show()
```

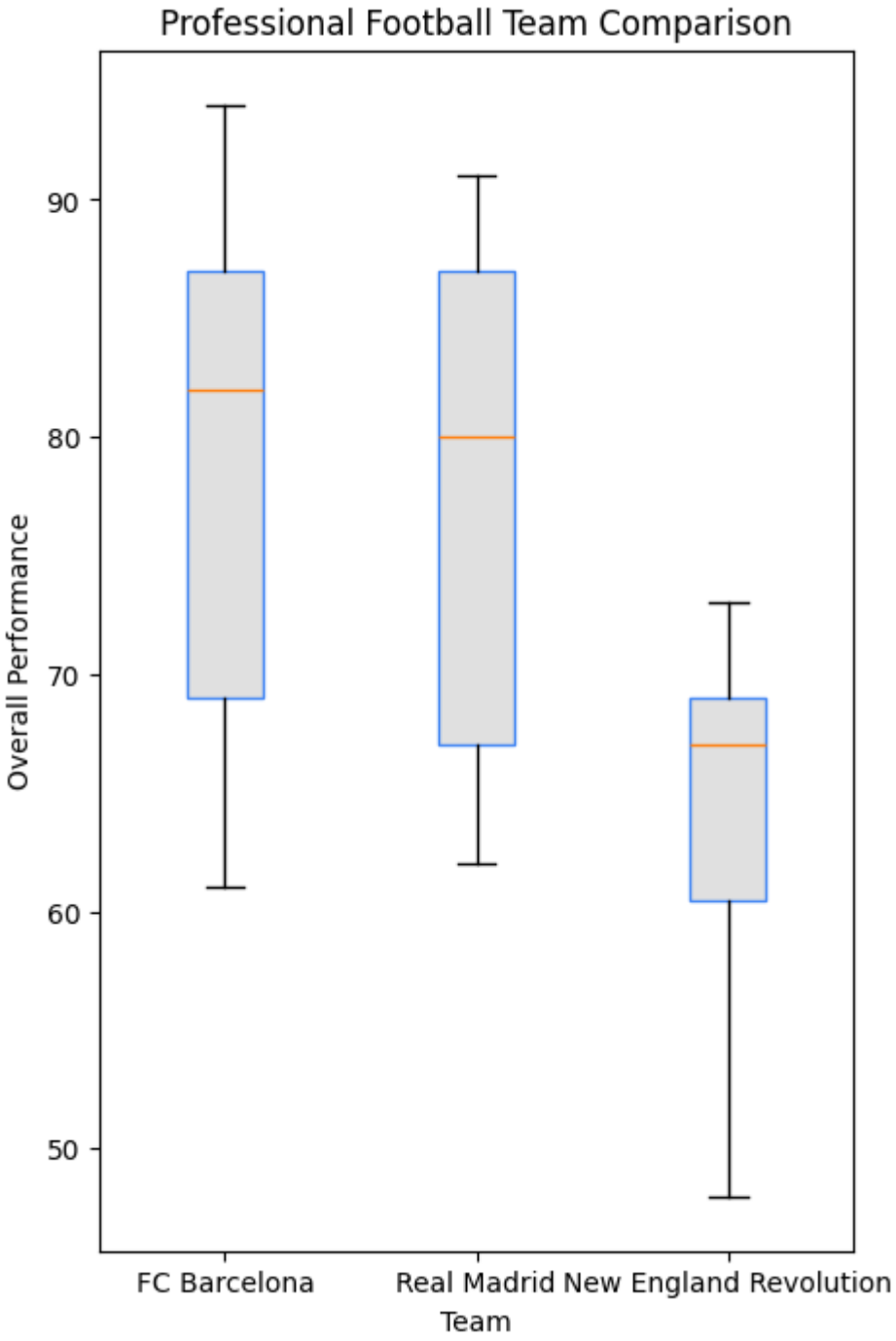


Comparing FIFA Teams to one another

Box & Whisker Plot

In [208…
```
barcelona = fifa.loc[fifa.Club == 'FC Barcelona']['Overall']
madrid = fifa.loc[fifa.Club == 'Real Madrid']['Overall']
revs = fifa.loc[fifa.Club=='New England Revolution']['Overall']
```

In [218…
```
plt.figure(figsize=(5,8))

plt.style.use('default')

labels = ['FC Barcelona','Real Madrid','New England Revolution']

boxes = plt.boxplot([barcelona,madrid,revs],labels=labels,patch_artist = True)

for box in boxes['boxes']:
    box.set(color='#4286f4')
    box.set(facecolor = '#e0e0e0')

plt.title('Professional Football Team Comparison')
plt.xlabel('Team')
plt.ylabel('Overall Performance')

plt.show()
```


Professional Football Team Comparison
Overall Performance
90
80
70
60
50
FC Barcelona
Real Madrid
New England Revolution
Team