# Matplotlib Tutorial

October 12, 2023

### 0.0.1 Load Necessary Libraries

[1]:

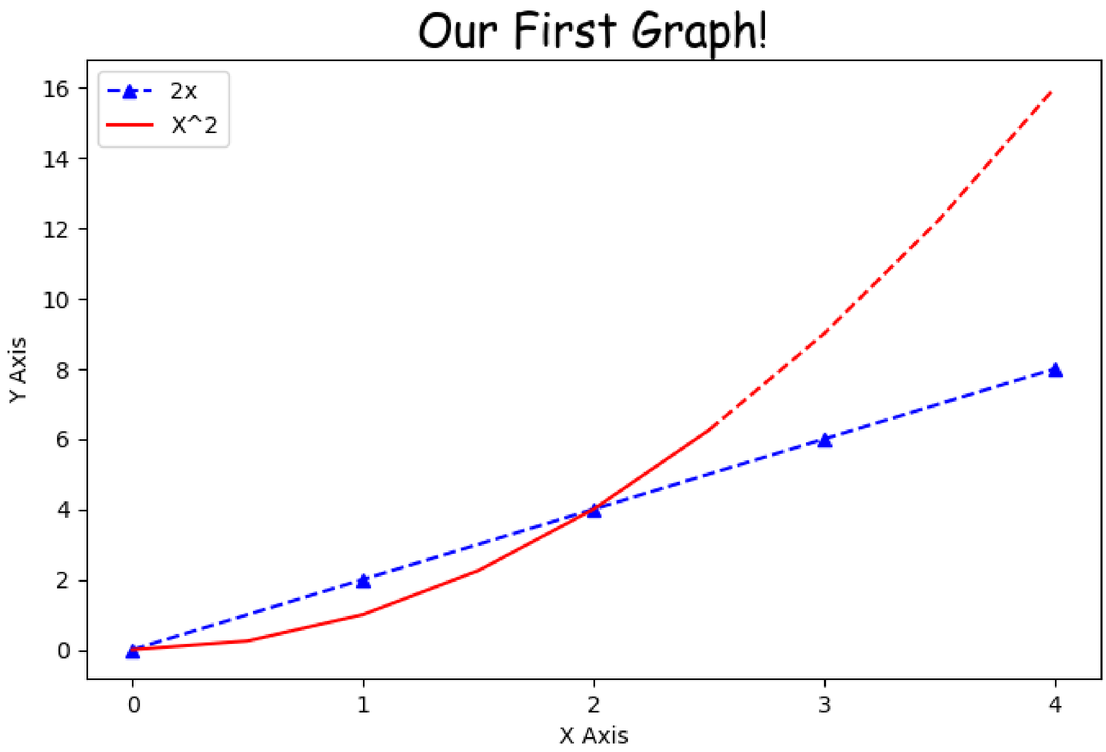
```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

### 0.0.2 Basic Graph

[2]:

```python
x = [0,1,2,3,4]
y = [0,2,4,6,8]

# Resize your Graph (dpi specifies pixels per inch. When saving probably should use 300 if possible)
plt.figure(figsize=(8,5), dpi=100)

# Line 1

# Keyword Argument Notation
#plt.plot(x,y, label='2x', color='red', linewidth=2, marker='.', linestyle='--', markersize=10, markeredgecolor='blue')

# Shorthand notation
# fmt = '[color][marker][line]'
plt.plot(x,y, 'b^--', label='2x')

## Line 2

# select interval we want to plot points at
x2 = np.arange(0,4.5,0.5)

# Plot part of the graph as line
plt.plot(x2[:6], x2[:6]**2, 'r', label='X^2')

# Plot remainder of graph as a dot
plt.plot(x2[5:], x2[5:]**2, 'r--')
```

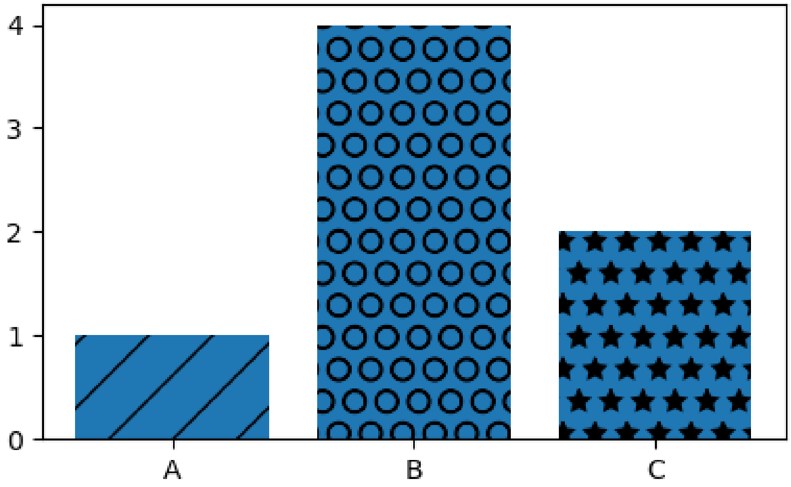
```
# Add a title (specify font parameters with fontdict)
plt.title('Our First Graph!', fontdict={'fontname': 'Comic Sans MS', 'fontsize':
 ↪ 20})

# X and Y labels
plt.xlabel('X Axis')
plt.ylabel('Y Axis')

# X, Y axis Tickmarks (scale of your graph)
plt.xticks([0,1,2,3,4,])
#plt.yticks([0,2,4,6,8,10])

# Add a legend
plt.legend()

# Save figure (dpi 300 is good when saving so graph has high resolution)
plt.savefig('mygraph.png', dpi=300)

# Show plot
plt.show()
```



### 0.0.3 Bar Chart

[3]:

```
labels = ['A', 'B', 'C']
values = [1,4,2]

plt.figure(figsize=(5,3), dpi=100)

bars = plt.bar(labels, values)

patterns = ['/', 'O', '*']
for bar in bars:
    bar.set_hatch(patterns.pop(0))

plt.savefig('barchart.png', dpi=300)

plt.show()
```



# 1 Real World Examples

Download data from my Github (gas_prices.csv & fifa_data.csv)

### 1.0.1 Line Graph

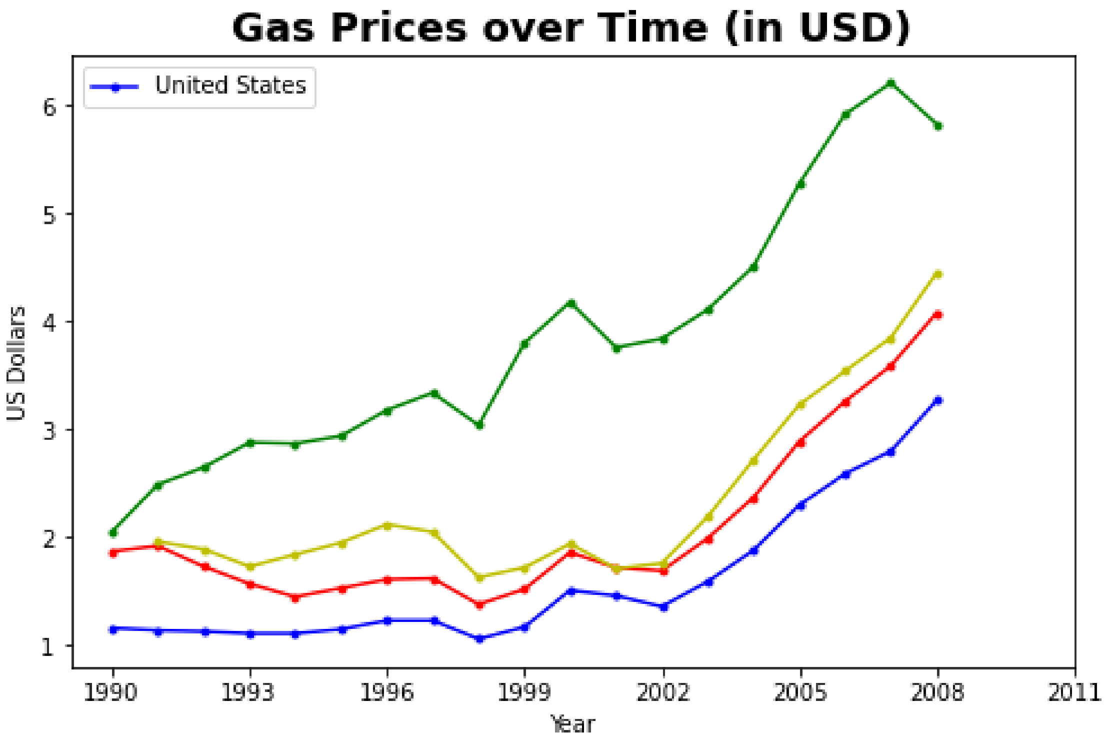
[30]:

```
# import warnings
# warnings.filterwarnings("ignore")

gas = pd.read_csv('gas_prices.csv')
```

```
plt.figure(figsize=(8,5))

plt.title('Gas Prices over Time (in USD)', fontdict={'fontweight':'bold',␣
 ↪'fontsize': 18})

# print (gas.USA[1,:])
print (gas.USA[0:])
print (np.expand_dims(gas.USA, axis=0)[0])
print (np.ndim(gas.USA))
# Canada South Korea  Australia
plt.plot(np.expand_dims(gas.Year, axis=0)[0], np.expand_dims(gas.USA,␣
 ↪axis=0)[0], 'b.-', label='United States')
plt.plot(np.expand_dims(gas.Year, axis=0)[0],  np.expand_dims(gas.Canada,␣
 ↪axis=0)[0], 'r.-')
plt.plot(np.expand_dims(gas.Year, axis=0)[0], np.expand_dims(gas['South␣
 ↪Korea'], axis=0)[0], 'g.-')
plt.plot(np.expand_dims(gas.Year, axis=0)[0], np.expand_dims(gas.Australia,␣
 ↪axis=0)[0], 'y.-')

# Another Way to plot many values!
# countries_to_look_at = ['Australia', 'USA', 'Canada', 'South Korea']
# for country in gas:
#     if country in countries_to_look_at:
#         plt.plot(gas.Year, gas[country], marker='.')

plt.xticks(gas.Year[::3].tolist()+[2011])

plt.xlabel('Year')
plt.ylabel('US Dollars')

plt.legend()

plt.savefig('Gas_price_figure.png', dpi=300)

plt.show()
```

```
0     1.16
1     1.14
2     1.13
3     1.11
4     1.11
5     1.15
6     1.23
7     1.23
8     1.06
9     1.17
```

```
10    1.51
11    1.46
12    1.36
13    1.59
14    1.88
15    2.30
16    2.59
17    2.80
18    3.27
Name: USA, dtype: float64
[1.16 1.14 1.13 1.11 1.11 1.15 1.23 1.23 1.06 1.17 1.51 1.46 1.36 1.59
 1.88 2.3  2.59 2.8  3.27]
1
```



### 1.0.2 Load Fifa Data

```
[5]: fifa = pd.read_csv('fifa_data.csv')

     fifa.head(5)
```

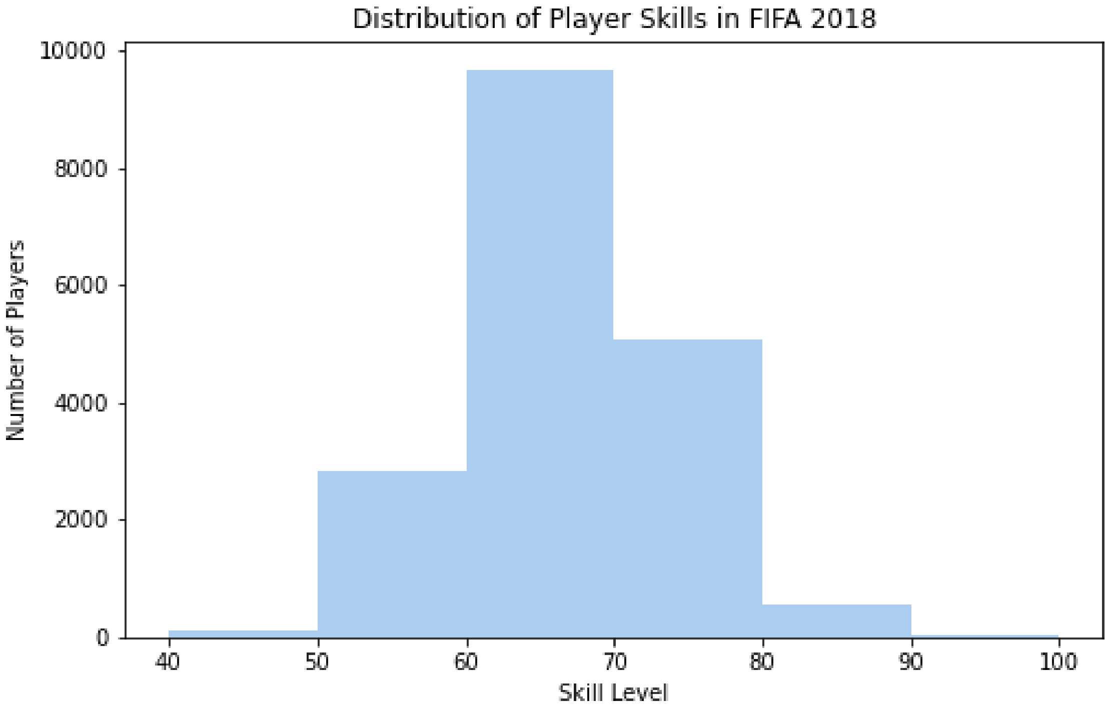
```
[5]:    Unnamed: 0      ID               Name  Age  \
    0            0  158023           L. Messi   31
```

```
1           1   20801  Cristiano Ronaldo   33
2           2  190871          Neymar Jr   26
3           3  193080             De Gea   27
4           4  192985       K. De Bruyne   27

                                            Photo Nationality  \
0  https://cdn.sofifa.org/players/4/19/158023.png   Argentina
1   https://cdn.sofifa.org/players/4/19/20801.png    Portugal
2  https://cdn.sofifa.org/players/4/19/190871.png      Brazil
3  https://cdn.sofifa.org/players/4/19/193080.png       Spain
4  https://cdn.sofifa.org/players/4/19/192985.png     Belgium

                                  Flag  Overall  Potential  \
0  https://cdn.sofifa.org/flags/52.png       94         94
1  https://cdn.sofifa.org/flags/38.png       94         94
2  https://cdn.sofifa.org/flags/54.png       92         93
3  https://cdn.sofifa.org/flags/45.png       91         93
4   https://cdn.sofifa.org/flags/7.png       91         92

                  Club  … Composure Marking StandingTackle  SlidingTackle  \
0         FC Barcelona  …      96.0    33.0           28.0           26.0
1             Juventus  …      95.0    28.0           31.0           23.0
2  Paris Saint-Germain  …      94.0    27.0           24.0           33.0
3    Manchester United  …      68.0    15.0           21.0           13.0
4      Manchester City  …      88.0    68.0           58.0           51.0

  GKDiving  GKHandling  GKKicking  GKPositioning GKReflexes Release Clause
0      6.0        11.0       15.0           14.0        8.0        €226.5M
1      7.0        11.0       15.0           14.0       11.0        €127.1M
2      9.0         9.0       15.0           15.0       11.0        €228.1M
3     90.0        85.0       87.0           88.0       94.0        €138.6M
4     15.0        13.0        5.0           10.0       13.0        €196.4M

[5 rows x 89 columns]
```

### 1.0.3 Histogram

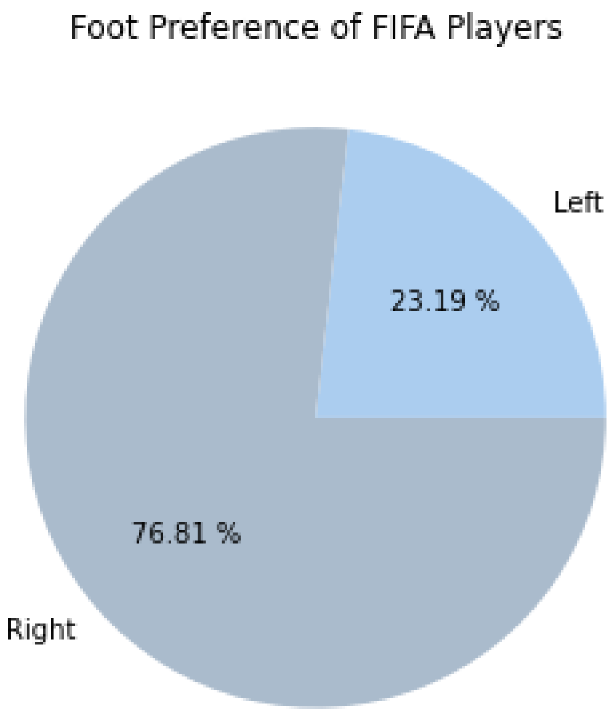
[6]:

```
bins = [40,50,60,70,80,90,100]

plt.figure(figsize=(8,5))

plt.hist(fifa.Overall, bins=bins, color='#abcdef')

plt.xticks(bins)

plt.ylabel('Number of Players')
plt.xlabel('Skill Level')
plt.title('Distribution of Player Skills in FIFA 2018')

plt.savefig('histogram.png', dpi=300)

plt.show()
```



### 1.0.4 Pie Chart

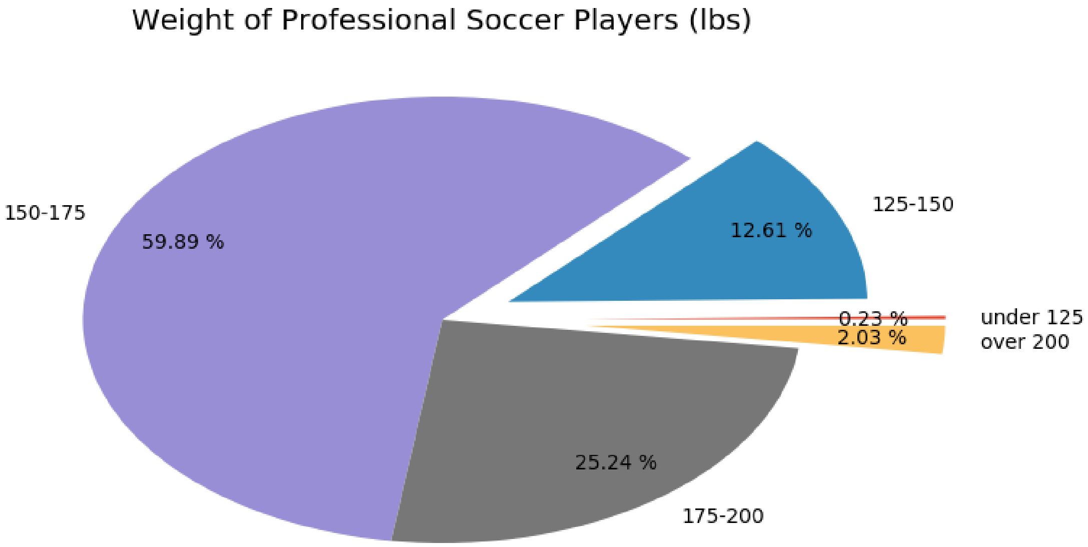
[7]:
```
left = fifa.loc[fifa['Preferred Foot'] == 'Left'].count()[0]
right = fifa.loc[fifa['Preferred Foot'] == 'Right'].count()[0]

plt.figure(figsize=(8,5))

labels = ['Left', 'Right']
colors = ['#abcdef', '#aabbcc']

plt.pie([left, right], labels = labels, colors=colors, autopct='%.2f %%')

plt.title('Foot Preference of FIFA Players')

plt.show()
```



### 1.0.5 Pie Chart #2

[11]:

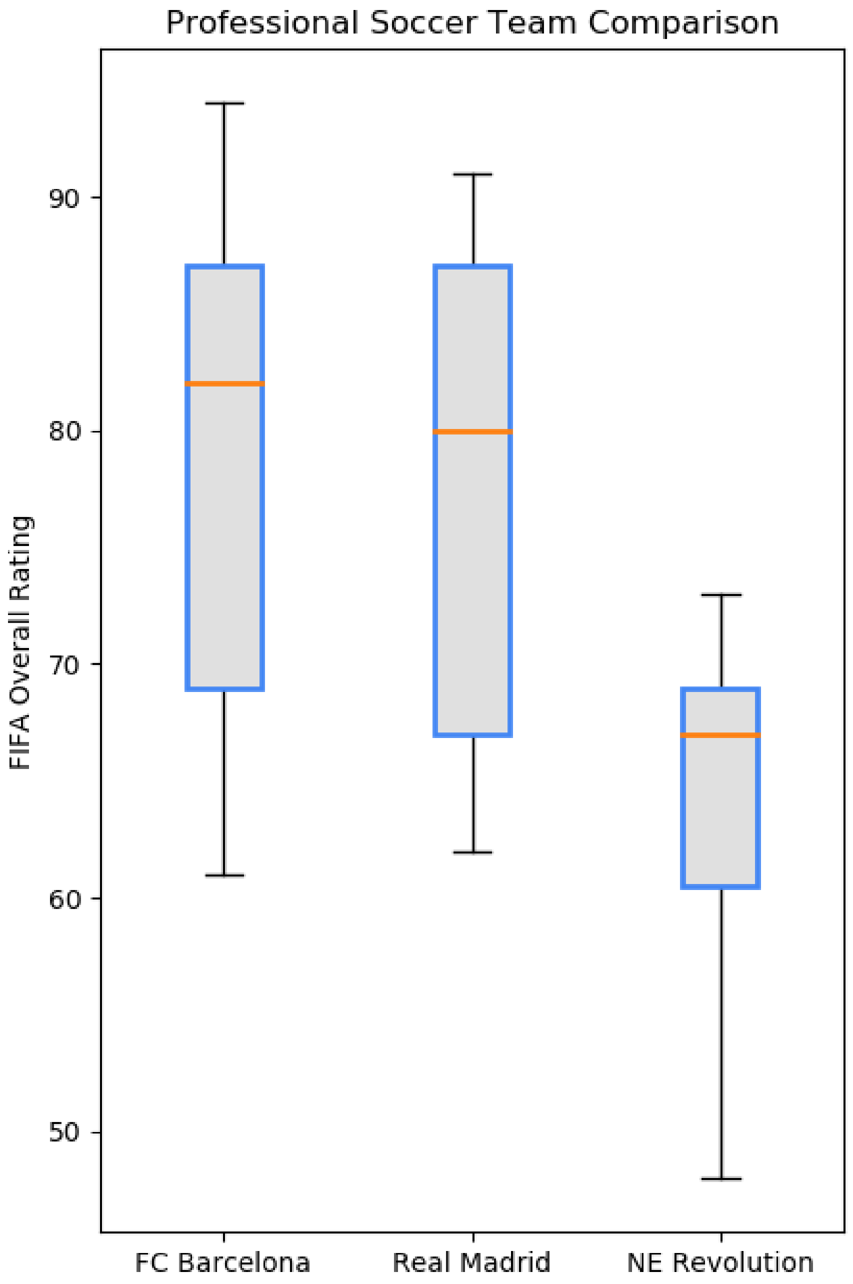
```
plt.figure(figsize=(8,5), dpi=100)

plt.style.use('ggplot')

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

light = fifa.loc[fifa.Weight < 125].count()[0]
light_medium = fifa[(fifa.Weight >= 125) & (fifa.Weight < 150)].count()[0]
medium = fifa[(fifa.Weight >= 150) & (fifa.Weight < 175)].count()[0]
medium_heavy = fifa[(fifa.Weight >= 175) & (fifa.Weight < 200)].count()[0]
heavy = fifa[fifa.Weight >= 200].count()[0]

weights = [light,light_medium, medium, medium_heavy, heavy]
label = ['under 125', '125-150', '150-175', '175-200', 'over 200']
explode = (.4,.2,0,0,.4)

plt.title('Weight of Professional Soccer Players (lbs)')
plt.pie(weights, labels=label, explode=explode, pctdistance=0.8,autopct='%.2f␣
↪%%')
plt.show()
```



### 1.0.6 Box and Whiskers Chart

[12]:

```
plt.figure(figsize=(5,8), dpi=100)

plt.style.use('default')

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']

#bp = plt.boxplot([barcelona, madrid, revs], labels=['a','b','c'],␣
↪boxprops=dict(facecolor='red'))
bp = plt.boxplot([barcelona, madrid, revs], labels=['FC Barcelona','Real␣
↪Madrid','NE Revolution'], patch_artist=True, medianprops={'linewidth': 2})

plt.title('Professional Soccer Team Comparison')
plt.ylabel('FIFA Overall Rating')

for box in bp['boxes']:
    # change outline color
    box.set(color='#4286f4', linewidth=2)
    # change fill color
    box.set(facecolor = '#e0e0e0' )
    # change hatch
    #box.set(hatch = '/')

plt.show()
```


Professional Soccer Team Comparison
FIFA Overall Rating
FC Barcelona
Real Madrid
NE Revolution