UCLA College | Social Sciences
Economics

Day 9: Visualization

matplotlib

Paul Schumacher, MSc Quantitative Economics

Announcements

Attendance

HW2 due

Wednesday 11pm

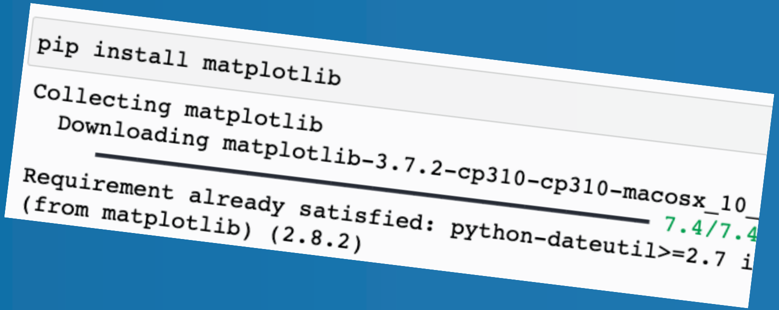
Final

Friday 4pm – 6pm

Class exercise: Pandas

- “PLF Day 8 Worksheet Pandas.ipynb”
- Data set: Hotel Reservations.csv
- Breakout rooms
- Time: 20 min

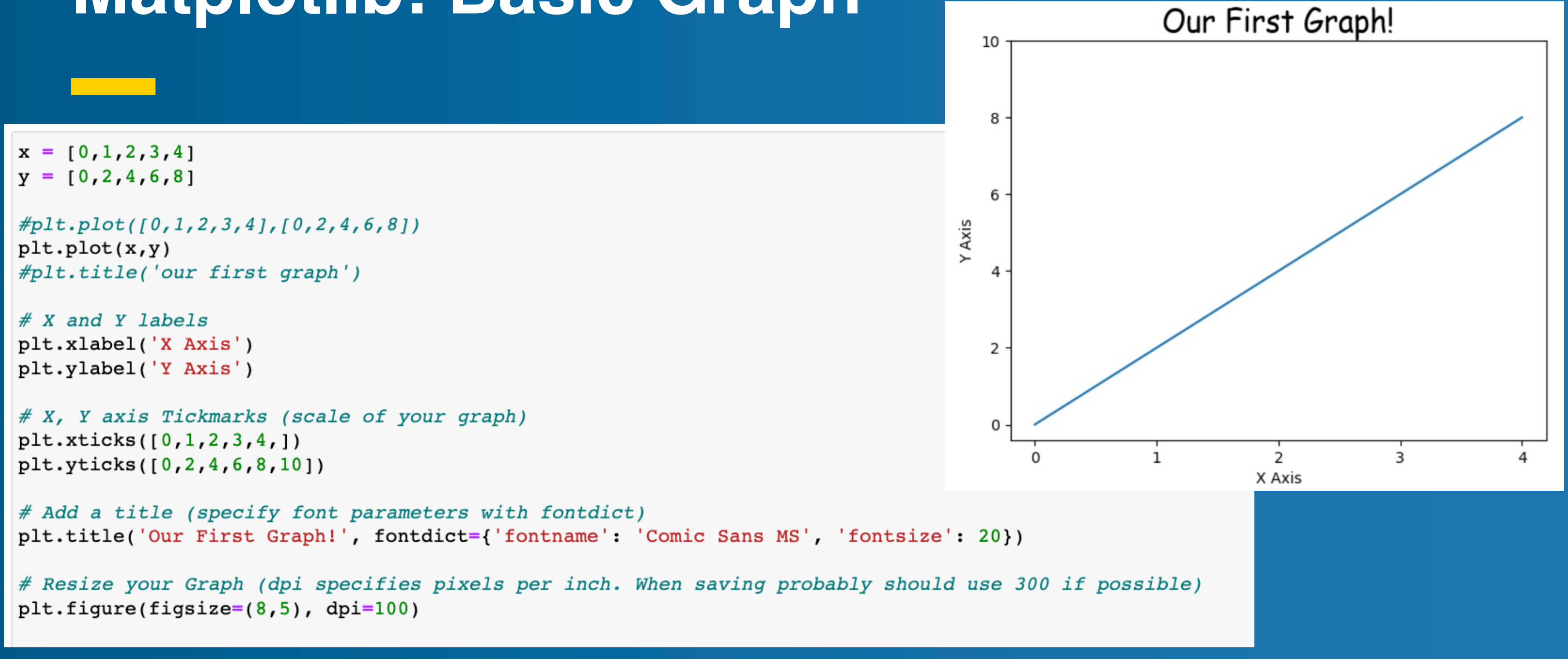
Visualization: Matplotlib



```
pip install matplotlib
Collecting matplotlib
  Downloading matplotlib-3.7.2-cp310-cp310-macosx_10_
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.4/7.4
Requirement already satisfied: python-dateutil>=2.7 i
(from matplotlib) (2.8.2)
```

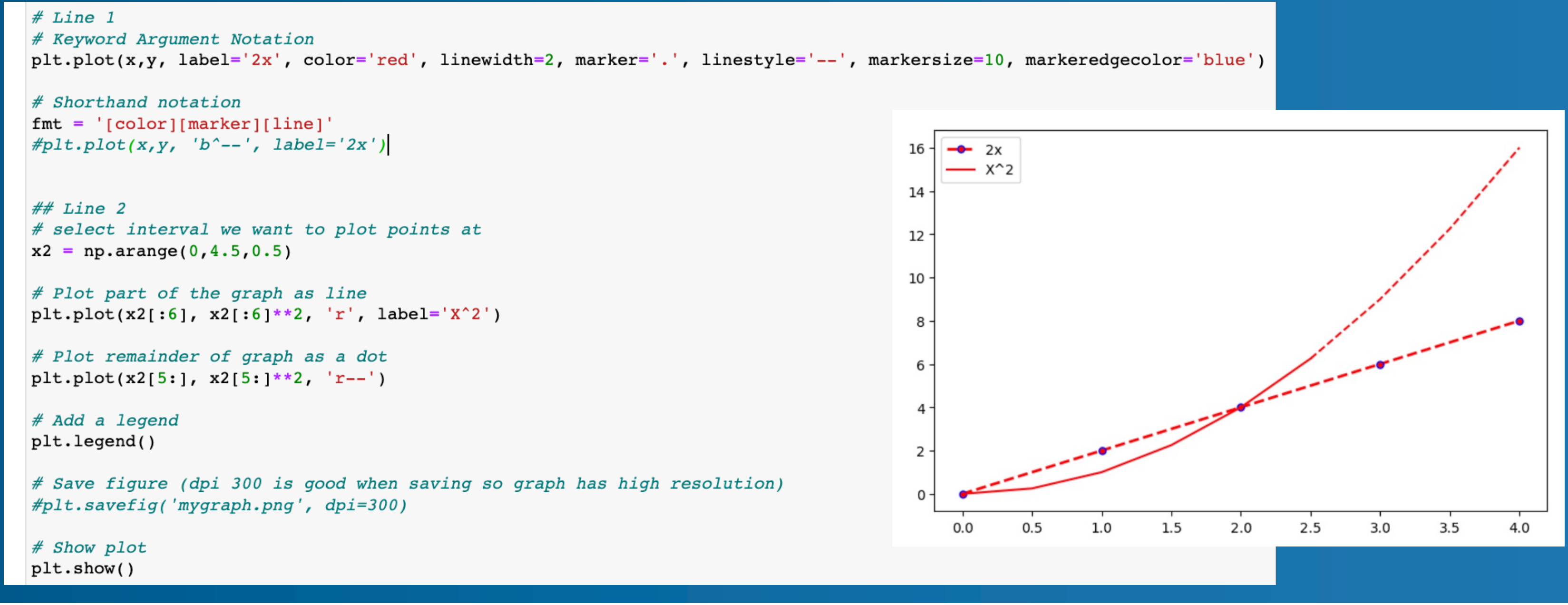
Documentation:
https://matplotlib.org/3.5.3/api/_as_gen/matplotlib.pyplot.html

Matplotlib: Basic Graph



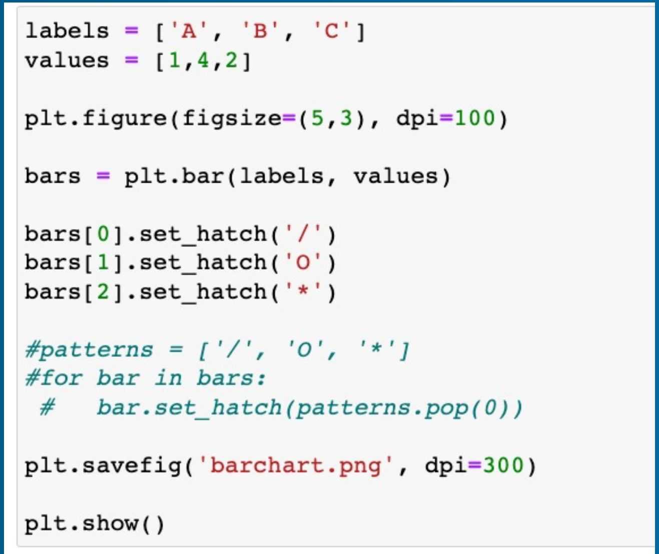
```
x = [0,1,2,3,4]
y = [0,2,4,6,8]

#plt.plot([0,1,2,3,4],[0,2,4,6,8])
plt.plot(x,y)
#plt.title('our first graph')

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

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

Matplotlib: Basic Graph

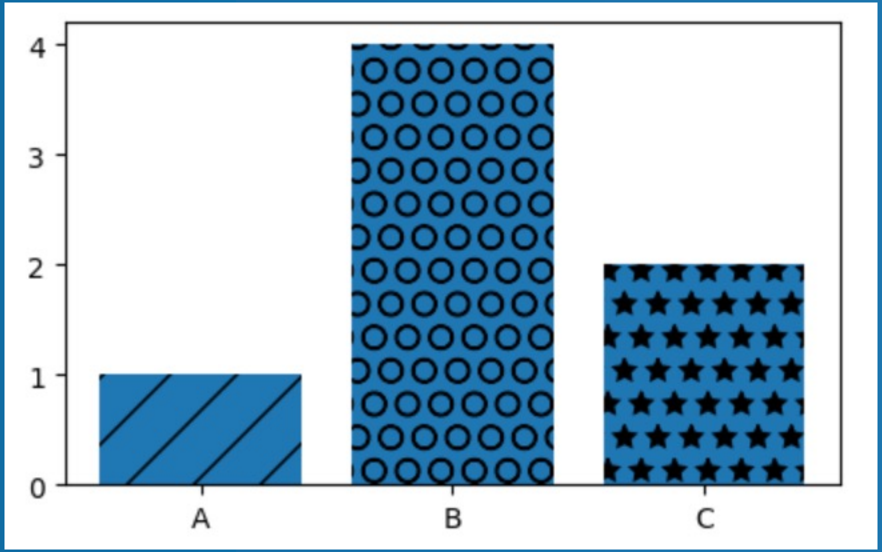


```
# Line 1
# Keyword Argument Notation
plt.plot(x,y, label='2x', color='red', linewidth=2, marker='.', linestyle='--', markersize=10, markeredgecolor='blue')

# 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 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()
```

Matplotlib: Bar Chart



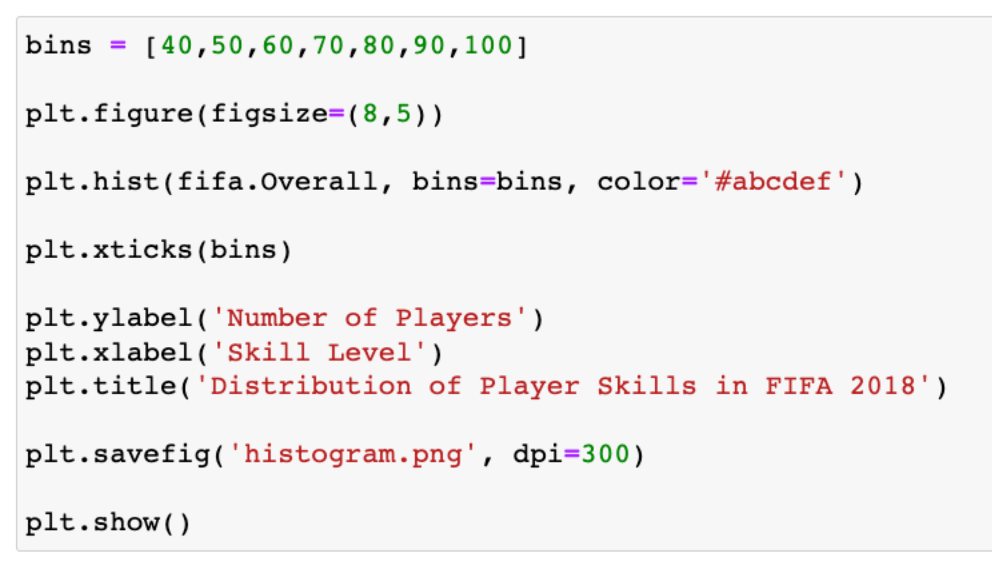
```
labels = ['A', 'B', 'C']
values = [1,4,2]

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

bars = plt.bar(labels, values)

bars[0].set_hatch('/')
bars[1].set_hatch('O')
bars[2].set_hatch('*')

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

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

plt.show()
```



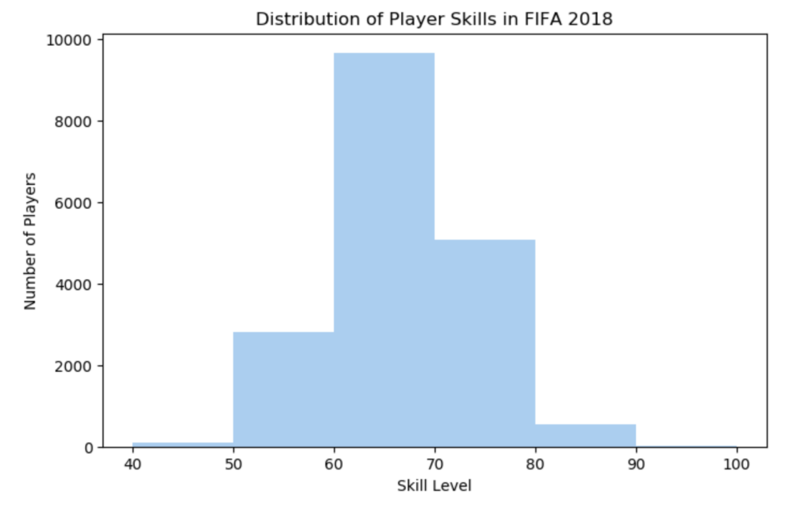
Class exercise: Matplotlib

- “PLF Day 9 Worksheet Matplotlib.ipynb”
- Data set: gas_prices.csv
- Breakout rooms
- Time: 10 min

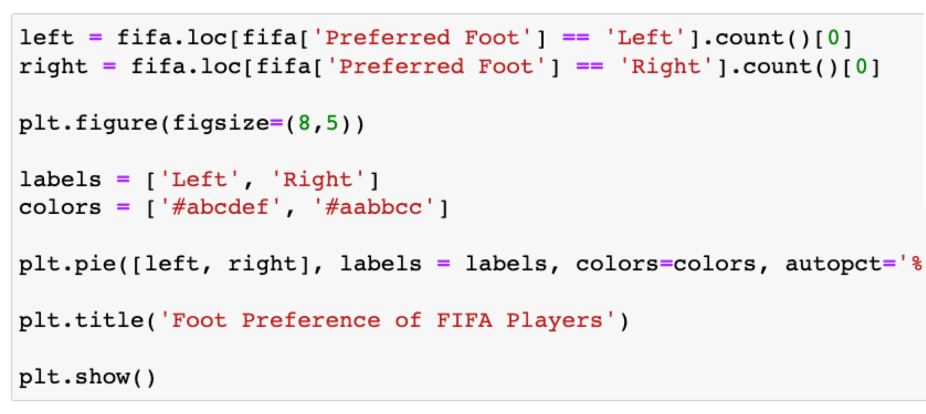
Matplotlib: Histogram



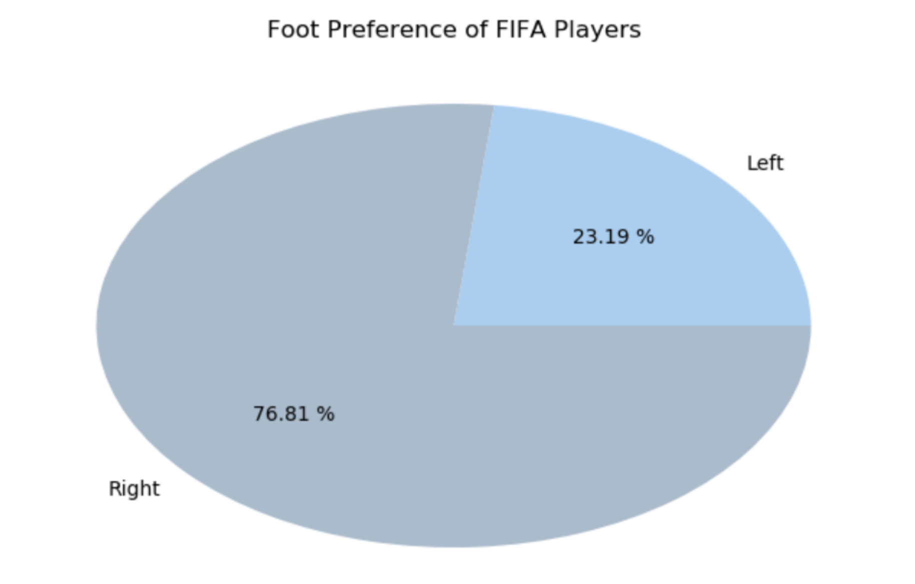
```
bins = [40,50,60,70,80,90,100]

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()
```



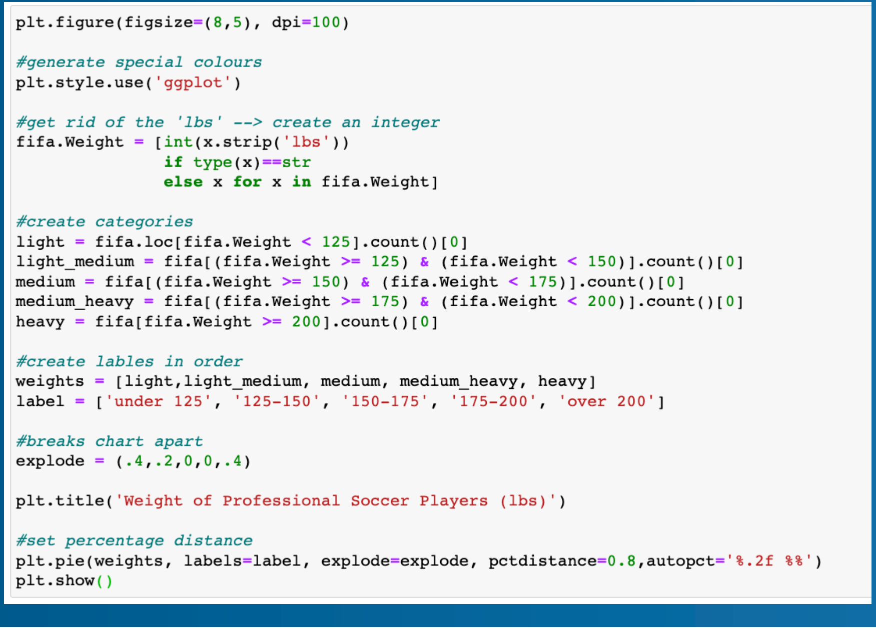
Matplotlib: Pie Chart



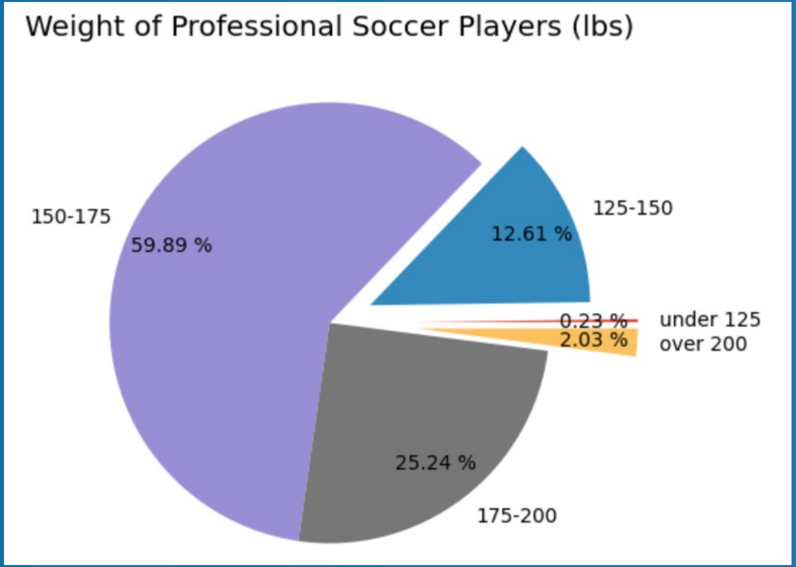
```
left = fifa.loc[fifa['Preferred Foot'] == 'Left'].count()[0]
right = fifa.loc[fifa['Preferred Foot'] == 'Right'].count()[0]

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()
```



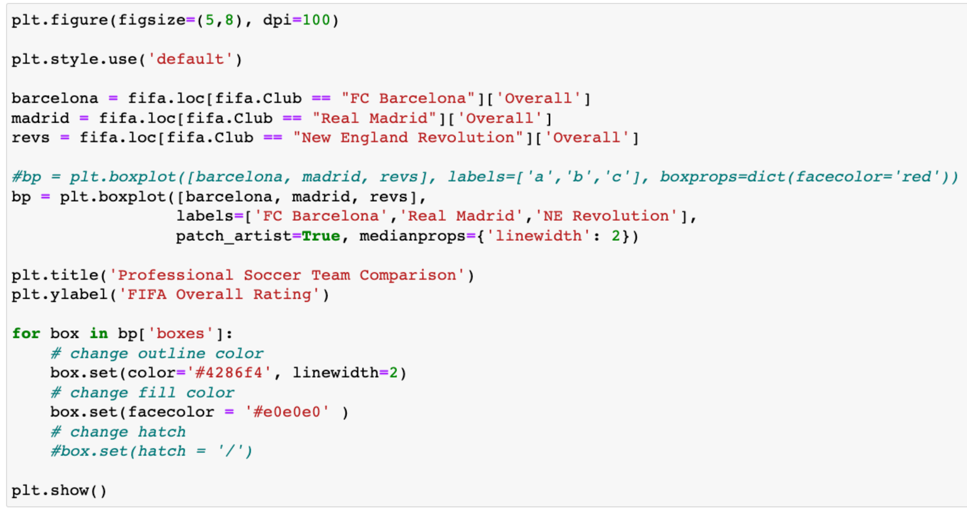
Matplotlib: Pie Chart II



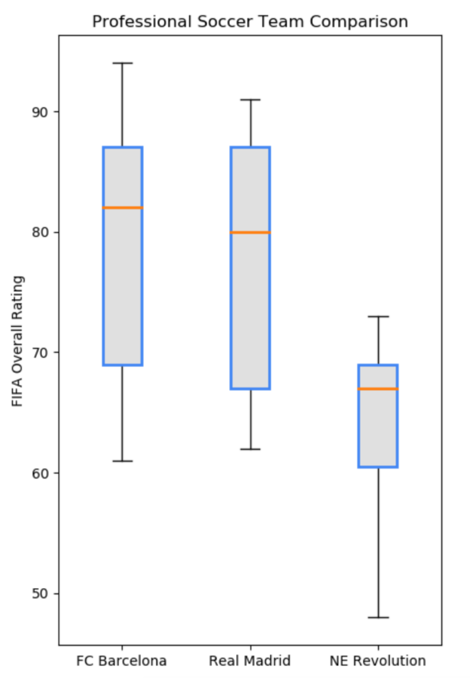
```python
plt.figure(figsize=(8,5), dpi=100)

#generate special colours
plt.style.use('ggplot')

#get rid of the 'lbs' --> create an integer
fifa.Weight = [int(x.strip('lbs'))
               if type(x)==str
               else x for x in fifa.Weight]

#create categories
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]

#create lables in order
weights = [light,light_medium, medium, medium_heavy, heavy]
label = ['under 125', '125-150', '150-175', '175-200', 'over 200']

#breaks chart apart
explode = (.4,.2,0,0,.4)

plt.title('Weight of Professional Soccer Players (lbs)')

#set percentage distance
plt.pie(weights, labels=label, explode=explode, pctdistance=0.8,autopct='%.2f %%')
plt.show()
```



Matplotlib: Box and Whiskers Chart



```
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()
```



Any Questions?