ITCS159 Software Lab for Basic Scientific Problem Solving
Lab8: Graph representation (Matplotlib and Seaborn)

September 27, 2021

Introduction

In this lab, you will learn how to present scientific data visually using Python. We will create visualization in the form of graphs, which allows us to illustrate a large data set and get insight from it. There are two important Python libraries in this lab, *Matplotlib* and *Seaborn*.

Matplotlib

Matplotlib is one of the most popular Python's data visualization library. The library was originally created to offer plotting feature in Python to work similarly to plotting feature in MatLab, another programming language. Matplotlib supports visualizations for both 2D and 3D.

Exercise 1) Open Jupyter Notebook and create a new folder called "visualization". Create the file 'lab8' and then try your first plot.

```
import matplotlib.pyplot as plt
plt.plot(1, 3, 'bo')
plt.plot(2, 2, 'ro')
```

What is the result? What will happen if you edit those numbers? Then, try this.

```
plt.plot([1, 3, 12, 30])
plt.plot([1, 4, 3, 12])
```

Try another piece of code. Please feel free to adjust the numbers. Comparing to the previous codes, what are differences?

```
import matplotlib.pyplot as plt
plt.plot([0, 3, 5, 6], [0, 5, 15, 10])
plt.plot([0, 1, 2, 6], [10, 8, 1, 3])
```

Exercise 2) Plot decoration. After you know how to create a plot. You can adjust the appearance of it as well as adding the legend. Note ('-' solid line style, '–' dashed line style, '-.' dash-dot line style, ':' dotted line style)

```
plt.plot([0, 3, 5, 6], [0, 5, 15, 10], '--',label='A',color=
    "pink", linewidth=5)
plt.plot([0, 1, 2, 6], [10, 8, 1, 3], ':', label='B',color="
    lightblue", linewidth=2)
plt.legend()
```

Then, you can adjust the size of your graph. Figure is the area that your graph drawn on. You can make change of it using `figsize`. The size is in inch.

```
fig = plt.figure(figsize=(6,2))
plt.plot([0, 3, 5, 6], [0, 5, 15, 10], '--',label='A',color=
    "pink", linewidth=5)
plt.plot([0, 1, 2, 6], [10, 8, 1, 3], ':', label='B',color="
    lightblue", linewidth=2)
plt.legend()
```

Milestone 1) You have now completed the Milestone 1 for Python list section. You should get sign off by your lab assistant.

Exercise 3) Bar chart. Matplot allows you to create a bar chart using 'bar' for vertical chart and 'barh' for horizontal chart. Try the following code. What each number describe?

```
import matplotlib.pyplot as plt
plt.barh([1,2,3],[0,1,5])
```

You can add creativity to make your graph look interesting. Check this out.

```
n = 6
x = np.arange(n)
y1 = (1-x/float(n))
y2 = (1-x/float(n))
plt.bar(x, +y1, facecolor='blue', edgecolor='white')
plt.bar(x, -y2, facecolor='red', edgecolor='white')
```

What if you need to have multiple bar charts?

```
plt.barh([1,2,3],[0,1,5])
plt.barh([2,2,5],[1,1,1])
plt.bar([1,2,3],[3,4,5])
```

Exercise 4) You should see the overlapping problem in the previous exercise. You can solve it using `subplot`. The code below is a single chart with subplot.

```
fig = plt.figure()
ax = fig.add_subplot(111)
ax.barh([1,2,3],[0,1,5])
```

The digits in the number '111' refers to The three arguments: *the number of rows*, *the number of columns*, and *the plot number*. Then, you can add more subplots.

```
fig = plt.figure(figsize=(20,10))
ax1 = fig.add_subplot(131)
ax2 = fig.add_subplot(132)
ax3 = fig.add_subplot(133)
ax1.barh([1,2,3],[0,1,5])
ax2.barh([2,2,5],[1,1,1])
ax3.bar([1,2,3],[3,4,5])
```

How the result looks like?

Exercise 5) Then, let's see other types of plot. Scatter plot.

```
fig = plt.figure()
ax = fig.add_subplot(111)
N = 30
x = np.random.rand(N)
y = np.random.rand(N)
scat = ax.scatter(x, y, 200 ,edgecolors = 'k', facecolors='c
    ')
```

Note Color abbreviations are supported ('b' blue, 'g' green, 'r' red, 'c' cyan, 'm' magenta, 'y' yellow, 'k' black, 'w' white)

Exercise 6) Stack plot

```
rng = np.arange(50)
rnd = np.random.randint(0, 10, size=(3, rng.size))
yrs = 1950 + rng
fig, ax = plt.subplots(figsize=(5, 3))
ax.stackplot(yrs, rng + rnd, labels=['Eastasia', 'Eurasia',
    'Oceania'])
ax.set_title('Combined debt growth over time')
ax.legend(loc='upper left')
ax.set_ylabel('Total debt')
ax.set_xlim(xmin=yrs[0], xmax=yrs[-1])
fig.tight_layout()
```

Milestone 2) You have now completed the Milestone 2 for Python list section. You should get sign off by your lab assistant.

Seaborn

Seaborn is a library that is an extension of Matplotlib. The library illustrates related statistical information of a graph to allow us to gain more insight about a dataset. Seaborn also closely integrated with pandas data structures. Seaborn offers many visualization tools. We will study only some of them.

Exercise 7) Make sure that you are using the latest version of Seaborn. Open Anaconda Prompt and type in:

```
conda update seaborn
```

Exercise 8) Download the 'tips.csv' from MyCourse into the same location as your Jupyter Notebook. Then, briefly investigate the dataset. You will use Seaborn to visualize it.

Exercise 9) Try to visualize multiple column from the dataset using the following code.

```
import seaborn as sns
sns.set()
tips = sns.load_dataset("tips")
sns.relplot(x="total_bill", y="tip", hue="smoker", style="
    smoker", size="size", data=tips)
```

How many column from the dataset that are visualized in this graph? Then, you edit to add some more code.

```
sns.set()
tips = sns.load_dataset("tips")
sns.relplot(x="total_bill", y="tip", col="time", hue="smoker
    ", style="smoker", size="size", data=tips)
```

Now, how many visualized column? Let's adjust the appearance to make it is easier to analyze.

```
sns.relplot(x="total_bill", y="tip", col="time", hue="size",
     style="smoker", size="size", palette="YlGnBu", markers=[
    "D","o"], sizes=(10, 125), edgecolor=".2", linewidth=.5,
    alpha=.75,data=tips)
```

Is it easier to analyze? Then, try to interpret to get insight from the visualization. What do you get out of it? After that, edit the code to change the way to visualize.

```
sns.set()
tips = sns.load_dataset("tips")
sns.relplot(x="total_bill", y="tip", col="smoker", hue="sex"
    , kind="line", data=tips)
```

Analyze the result. Do you gain more insight from it? If you need you can use the following code to see some parts of the raw data.

```
tips.head()
```

Exercise 10) Try other available visualization. Categorical plots.

```
import seaborn as sns
sns.set()
tips = sns.load_dataset("tips")
sns.catplot(x="day", y="total_bill", hue="smoker", kind="
    swarm", data=tips)
```

How can you explain the result? Then, slightly change the appearance.

```
sns.catplot(x="day", y="total_bill", hue="smoker", kind="
    violin", split=True, data=tips)
```

Is it easier to interpret? How about this?

```
sns.catplot(x="day", y="total_bill", hue="smoker", kind="bar
    ", data=tips)
```

Milestone 3) You have now completed the Milestone 3. You should get sign off by your lab assistant.