5CS037 Concepts and Technologies of AI Workshop-2 Data Visualizations with Matplotlib.

Siman Giri

December 5, 2023

1. Data Visualization: Introduction.

1.1 EDA: Exploratory Data Analysis

"The greatest value of a picture is when it forces us to notice what we never expected to see"

John Tukey

Disclaimer: EDA is not about knowing how to plot a graph but all about knowing which graph to plot, and what information to extract.

Siman Giri

1.2 Visualization Goals.

- **Communicate(Explanatory)**
    - Present data and ideas.
    - Explain and inform.
    - Provide evidence and support.
    - Influence and persuade.
- **Analyze (Exploratory)**
    - Explore the data.
    - Assess a situation.
    - Determine how to proceed.
    - Decide what to do.

1.3 Recommended EDA Workflow

1. **Build** a DataFrame from the data (ideally, put all in this object)
2. **Clean** the DataFrame. For example:
   - Each row describes a single object.
   - Each column describes a property of that object.
   - Columns are numeric whenever appropriate.
   - Columns contain atomic properties that can not be further decomposed.
3. Explore **global properties**.Summarize a whole data using plots like histogram, scatter plots etc.
4. Explore **group properties**. Use groupby and small multiples to compare subsets of the data.

1.4 Effective Visualizations.

Followings are must for an effective visualizations:

1. Have a graphical integrity.
2. Keep it simple.
3. Use the right display.
4. Use color sensibly.

1.4 Tools for Visualizations

Following three tools are the most popular tools used for **Data Visualizations and Exploration.**
They are inter-mixable.

- Pandas Visualization module.
- Matplotlib
- Seaborn

```
    import pandas as pd
    import seaborn as sns
    import matplotlib.pyplot as plt

```

2.Getting Started with Matplotlib.

2.1 Plotting Your First Figure

- **Style:1**

```python
import matplotlib.pyplot as plt
days = ["Mon","Tue","Wed","Thu","Fri","Sat","Sun"]
steps_walked = [8934,14902,3409,25672,12300,2023,6890]
plt.plot(steps_walked)

```

- **Style:2**

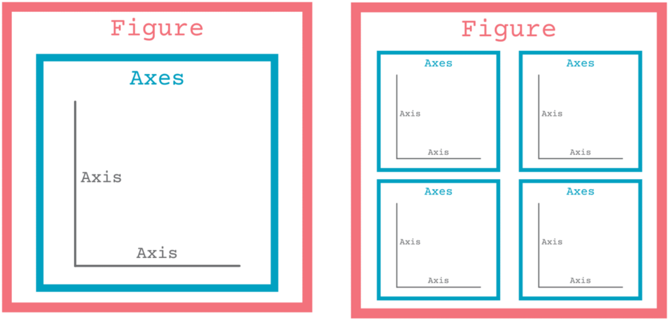
```python
import matplotlib.pyplot as plt
days = ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"]
steps_walked = [8934, 14902, 3409, 25672, 12300, 2023, 6890]
plt.plot(days, steps_walked)
plt.show()

```

Observe the difference between above two outputs.

A. Anatomy of Matplotlib Figure.

- When working with data visualisation in Python, you'll want to have control over all aspects of your figure.
- In this section, you'll learn about the main components that make up a figure in Matplotlib.

Figure: **Components of Matplotlib Figure**



A.1 Anatomy of Matplotlib Figure.

Everything in Python is an object, and therefore, so is a Matplotlib figure. In fact, a Matplotlib figure is made up of several objects of different data types.
There are three main parts to a Matplotlib figure:

- **Figure:** This is the whole region of space that's created when you create any figure. The Figure object is the overall object that contains everything else.
- **Axes:** An Axes object is the object that contains the x-axis and y-axis for a 2D plot. Each Axes object corresponds to a plot or a graph. You can have more than one Axes object in a Figure, as you'll see later on in this Chapter.
- **Axis:**An Axis object contains one of the axes, the x-axis or the y-axis for a 2D plot.

2.2 Customising the plots

- **Add a custom marker.**

```python
import matplotlib.pyplot as plt
days = ["Mon","Tue","Wed","Thu","Fri","Sat","Sun"]
steps_walked =
    [8934,14902,3409,25672,12300,2023,6890]
plt.plot(days, steps_walked, "o")
plt.show()

```

The third argument in plot() now indicates what marker you'd like to use. The string ”o” represents filled circles.
Also try following markers and observe the output:

```
plt.plot(t, t, ’r--’, t, t**2, ’bs’, t, t**3, ’g^’)
```

2.2 Customising the plots:Titles and labels

- **Adding titles,labels and legends.**

```python
import matplotlib.pyplot as plt
days = ["Mon","Tue","Wed","Thu","Fri","Sat","Sun"]
steps_walked = [8934,14902,3409,25672,12300,2023,6890]
steps_last_week = [9788,8710,5308,17630,21309,4002,5223]
plt.plot(days, steps_walked, "o-g")
plt.plot(days, steps_last_week, "v--m")
plt.title("Step count | This week and last week")
plt.xlabel("Days of the week")
plt.ylabel("Steps walked")
plt.grid(True)
plt.legend(["This week", "Last week"])
plt.show()

```

Observe the output.

2.3 Creating Subplots

```python
def f(t):
    return np.exp(-t) * np.cos(2*np.pi*t)
t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.02)
plt.figure()
plt.subplot(211)
plt.plot(t1, f(t1), ’bo’, t2, f(t2), ’k’)
plt.subplot(212)
plt.plot(t2, np.cos(2*np.pi*t2), ’r--’)
plt.show()
```

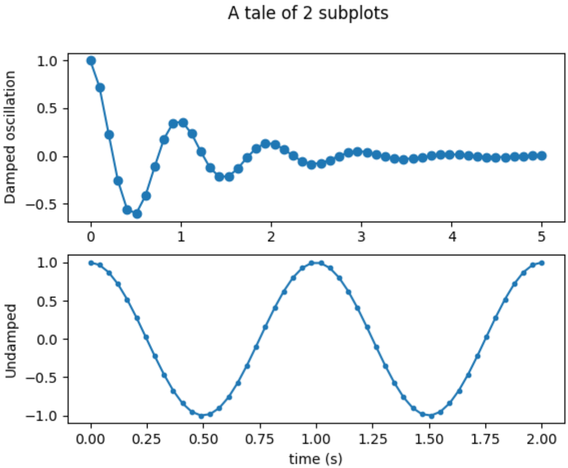
Observe and find what are the arguments for `plt.subplot()`.

Task-1

```python
import matplotlib.pyplot as plt
import numpy as np
# Create some fake data.
x1 = np.linspace(0.0, 5.0)
y1 = np.cos(2 * np.pi * x1) * np.exp(-x1)
x2 = np.linspace(0.0, 2.0)
y2 = np.cos(2 * np.pi * x2)
```

1. Rewrite above code.
2. Recreate subplots on slide 16.

Task-1


A tale of 2 subplots
Damped oscillation
1.0
0.5
0.0
−0.5
0
1
2
3
4
5
Undamped
1.0
0.5
0.0
−0.5
−1.0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
time (s)

3.Data Visualizations:

3.1 Data Visualizations: Task1

For the `bank.csv` from previous week perform following:

1. Plot histograms of age.
2. plot in bar showing number of subscriptions according to level of education.
3. Plot in bar showing number of subscriptions according to job.
4. Plot bar chart for age,balance.
5. Plot histograms of balance.

3.1 Data Visualizations: Task2

For the bank.csv from previous week perform following:

1. Check number of acceptance and rejection in target variable y and plot in pie chart
2. Check Total number of clients who subscribed according to the marital status
3. Check Total number of clients on the basis of education
4. Check the general level of education of people based on the provided data set
5. Calculate the ratio who said yes according to their education

3.1 Data Visualizations: Task3

For the `bank.csv` from previous week perform following:

- Split the Dataset into numeric column and Categorical Column.
- Explore the documentation of matplotlib and perform two unique plots each for Numerical Data and Categorical Data. {Unique: some interesting plots we have not discussed in class for example count plots etc.

3.1 Data Visualizations: Task4

For the `performance.csv` from previous week perform following:

- Write a code to show histogram plots and boxplots to visualize the distribution of the variables"age","absence",and,"G3". Interpret the results and comment on the distribution of each variable. Write your answer in text cell of jupyter notebook.
- Write a code to show a bar graph of the total number of students who passed the final term grouped according to the school that they belong to. Use proper labels in the graph and interpret the results.
- Write a code to show a pie chart of analysis of students' grades as per their school. Use proper labels in the graph and interpret the results.
- Write a code to show a bar graph with the relation of the father's occupation with the grade of the student. Use proper labels in the graph and interpret the results.