HW4

October 23, 2024

1 STOR 320 Homework 4 Visualization

Please submit the solution to gradescope by 11:59 PM, Oct 24, Thursday.

Name: Ivy Nangalia

PID: 730670491

[ ]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
import requests
from bs4 import BeautifulSoup
```

1.1 Problem 1: Visualization of research areas for UNC School of Data Science and Society (25 points)

In this problem, you will use wordcloud to visualize the research areas of faculty members at UNC SDSS

1.1 Use web scraping techniques to collect the research interests of all faculty members in the UNC School of Data Science and Society from the webpage: `https://datascience.unc.edu/people/?wpv_view_count=743&wpv-people-category=faculty&wpv_post_search=` (10 points)

- Create a `pd.DataFrame` that contains two columns: `faculty_name` and `research areas`.
- For example, one row of the dataframe is `David Adalsteinsson` and `biology, interface problems, spatial modeling, algorithm development, creating tools for scientific computing and visualization`.
- For faculty members without listed research interests, use `np.nan` to represent missing values.

1.2 Notice that there are 4 pages of faculty members in total. Use web scraping to extract the research areas from all 4 pages and store them in the same format as described in 1.1. (2 points)

- Hint: You can use `'https://datascience.unc.edu/people/?wpv_view_count=743&wpv-people-category` `+ str(i)` as the URL to access the content on page `i`.

1.3 How many rows do you obtain in the 1.2? How many rows have missing research areas? (2 points)

Hint: You can use `notnull()` or `isnull()` to check if cells within a DataFrame have missing data.

1.4 From the DataFrame in 1.2, select rows with research areas. Combine all the research areas into a single string called `text`. Remove the separator characters: `,`, `;` and `and`. (6 points)

1.5 Create a wordcloud use function `WordCloud`. Set the title as **Research Areas for UNC SDSS**. Based on your observation, what are the main research topics for UNC SDSS? (5 points)

```
[ ]: #1.1
     url_1 = "https://datascience.unc.edu/people/?
      ↪wpv_view_count=743&wpv-people-category=faculty&wpv_post_search="

     # Call the GET request
     response_1 = requests.get(url_1)

     soup = BeautifulSoup(response_1.text, "html.parser")
```

```
[ ]: scrape = ".col-md-8 p , h2"

     faculty_name_1 = []
     research_areas_1 = []

     for a in soup.select(scrape):
         a = a.get_text().strip()
         #print(a)
         if "," in a:
             research_areas_1.append(a)
         elif ";" in a:
             research_areas_1.append(a)
         else:
             faculty_name_1.append(a)
             if len(faculty_name_1) != len(research_areas_1) + 1:
                 research_areas_1.append(np.NAN)

     faculty_1 = pd.DataFrame({"faculty_name": faculty_name_1, "research_areas":␣
      ↪research_areas_1})
     faculty_1
```

```
[ ]:                  faculty_name  \
     0         David Adalsteinsson
     1                  Stan Ahalt
     2                   Jay Aikat
     3           Amarjit Budhiraja
     4             Iain Carmichael
     5                    Can Chen
     6             Anita Crescenzi
     7   Snehalkumar ‘Neil’ Gaikwad
     8             Melissa Haendel
     9             Thomas Hofweber
```

```
                                      research_areas
0  biology, interface problems, spatial modeling,…
1  Signal, image, and video processing, high-perf…
2  Experimental methods and models in networking …
3  large deviations, stochastic control, stochast…
4  Computational pathology, deep learning for med…
5  control theory, network science, tensor algebr…
6  impact of time constraints and time pressure o…
7  human-AI alignment; AI and decision making; AI…
8                                                NaN
9  metaphysics, the philosophy of language, the f…
```

```
[ ]: #1.2

     faculty_name = [] # initializing the lists outside the loop
     research_areas = []

     for i in range(1, 5):  # not inclusive of upper bound

         url = "https://datascience.unc.edu/people/?
      ↪wpv_view_count=743&wpv-people-category=faculty&wpv_post_search=&wpv_paged="␣
      ↪+ str(i)
         response = requests.get(url)
         soup = BeautifulSoup(response.text, "html.parser")

         scrape = ".col-md-8 p , h2"

         index: int = 0

         for a in soup.select(scrape):
             a = a.get_text().strip()
             #print(a)
             if "," in a:
                 research_areas.append(a)
             elif ";" in a:
                 research_areas.append(a)
             else:
                 faculty_name.append(a)
                 if len(faculty_name) != len(research_areas) + 1:
                     research_areas.append(np.NAN)

     faculty = pd.DataFrame({"faculty_name": faculty_name, "research_areas":␣
      ↪research_areas})
     faculty
```

```
[ ]:                   faculty_name  \
     0          David Adalsteinsson
```

```
1                   Stan Ahalt
2                    Jay Aikat
3            Amarjit Budhiraja
4              Iain Carmichael
5                     Can Chen
6              Anita Crescenzi
7   Snehalkumar ‘Neil’ Gaikwad
8              Melissa Haendel
9              Thomas Hofweber
10                 Hsun-Ta Hsu
11                 Dan Kessler
12             Shahar Kovalsky
13                  Harlin Lee
14                  Youzuo Lin
15                   Yifei Lou
16              Terry Magnuson
17               Richard Marks
18                Steve Marron
19                 Alex McAvoy
20               Julie McMurry
21                Lina Montoya
22           Santiago Olivella
23             Courtney Rivard
24           Rei Sanchez-Arias
25               Keriayn Smith
26               Jack Snoeyink
27                 Justin Sola
28         Matthew G. Springer
29                  Huaxiu Yao
30                 David Yokum
31               Weitong Zhang
32                 Chudi Zhong

                                       research_areas
0   biology, interface problems, spatial modeling,…
1   Signal, image, and video processing, high-perf…
2   Experimental methods and models in networking …
3   large deviations, stochastic control, stochast…
4   Computational pathology, deep learning for med…
5   control theory, network science, tensor algebr…
6   impact of time constraints and time pressure o…
7   human-AI alignment; AI and decision making; AI…
8                                                NaN
9   metaphysics, the philosophy of language, the f…
10                                               NaN
11  statistical analysis of networks, post-selecti…
12  optimization, geometry, computer graphics and …
```

```
13  Networks, manifolds, optimal transport, noncon…
14  physics-informed machine learning, deep learni…
15  Image processing, sparse signal recovery, nume…
16  mammalian genetics, genomics, chromatin remode…
17     AR/VR, machine learning, context-aware systems
18  statistics, data science and machine learning,…
19  evolutionary dynamics, game theory, multi-agen…
20                                                NaN
21  causal inference, precision health/policy, (op…
22  statistical models, particularly Bayesian grap…
23          rhetoric, composition, digital humanities
24  Data mining, machine learning algorithm develo…
25  RNA biology, noncoding RNAs, gene editing, bio…
26  Computational geometry; algorithms for geograp…
27  gun ownership, trends in social research, the …
28                                                NaN
29                                                NaN
30                                                NaN
31           reinforcement learning, machine learning
32  machine learning, optimization, human-model in…
```

[ ]:

```
#1.3
missing = faculty["research_areas"].isnull().sum()

len(faculty), missing
```

[ ]:

```
(33, 6)
```

There are 33 rows in the DataFrame from 1.2 in total, and 6 of those rows have null values.

[ ]:

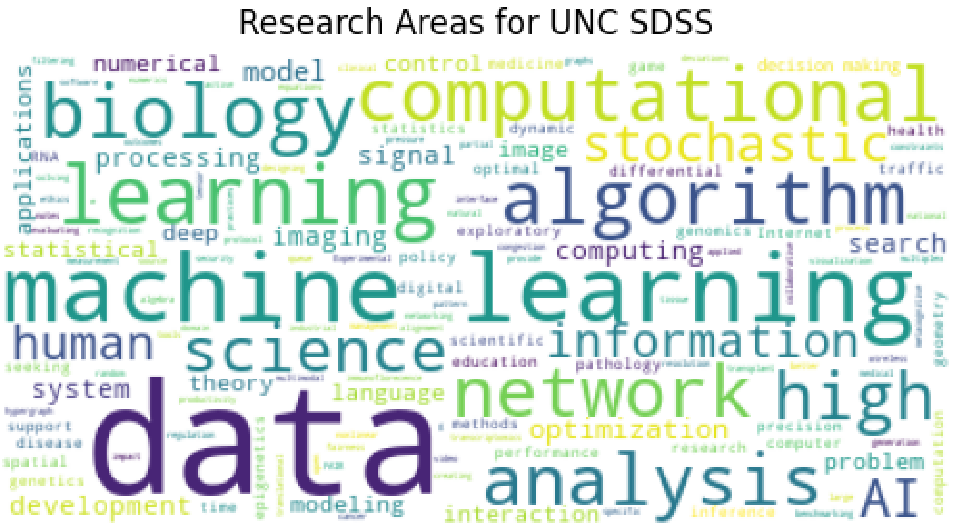
```
#1.4
research_areas = faculty["research_areas"].dropna()
text = ", ".join(research_areas).replace(",", "").replace(";", "").replace(" and", "")
text[0:200] # sliced to truncate output
```

[ ]:

```
'biology interface problems spatial modeling algorithm development creating
tools for scientific computing visualization Signal image video processing high-
performance scientific industrial computing p'
```

[ ]:

```
wordcloud = WordCloud(background_color="white")
wordcloud.generate(text)
plt.imshow(wordcloud)
plt.title("Research Areas for UNC SDSS")
plt.axis("off")
plt.show()
```



It seems like machine learning, algorithms, “data”, and biology are the biggest research focuses for SDSS staff.

1.2 Problem 2. (20 points).

Given a function $f(x) = \sin(5x) + \cos(3x) + x^2$, complete the following tasks:

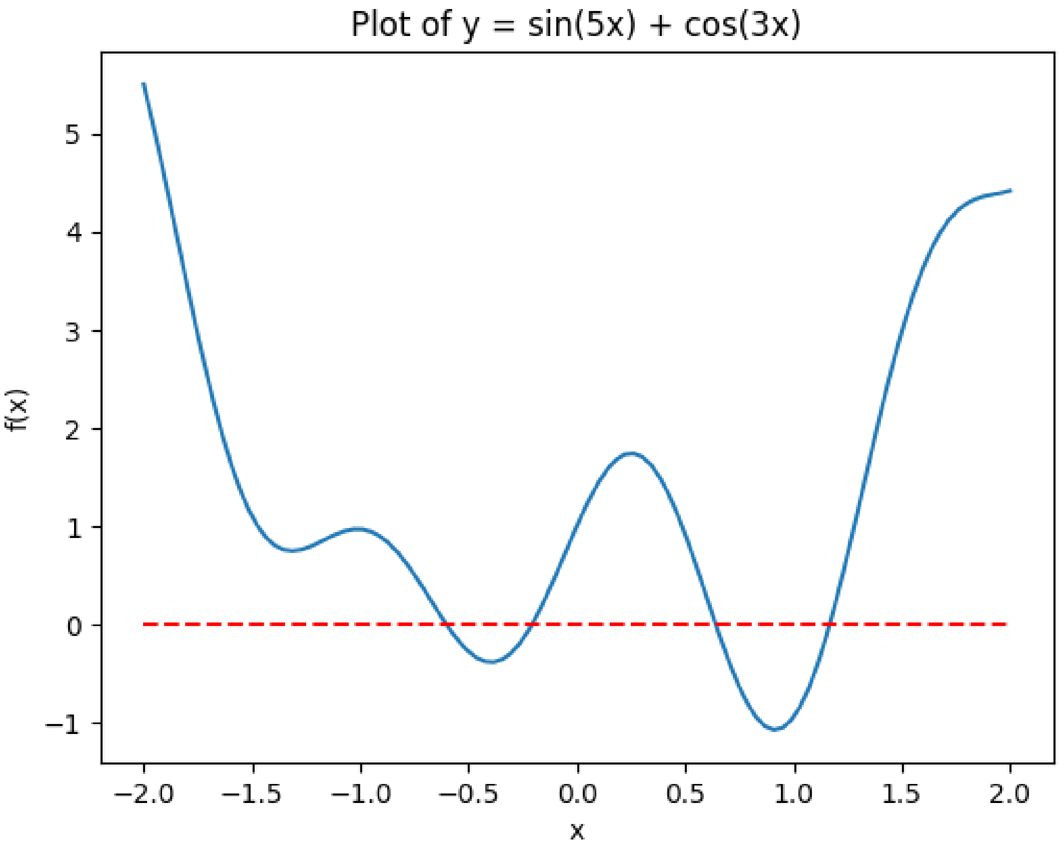
- 2.1 visualize this function for $x \in [-2, 2]$ using `plt`. Add a title to the plot. (5 points)
- 2.2 Add a horizontal line at $y = 0$ to the graph in 2.1 (2 points)

Hint: You can use `plt.hlines()` to draw a horizontal line.

- 2.3 How many intersections of the function and the horizontal line do you have? (3 points)
- 2.3 Based on the plot in 2.2, create another plot that zooms in on the area where x is between `[0.6, 0.7]`. Add a title to the plot. (5 points)
- 2.4 Create another plot to find the intersections of the function and the horizontal line. Mark them with red dots. You may find the intersections by observation. (5 points)

[ ]:
```
def f(x):
    return np.sin(5*x) + np.cos(3*x) + x**2

#2.1
x = np.linspace(-2, 2, 100)
y = f(x)

plt.plot(x, y)
plt.title("Plot of y = sin(5x) + cos(3x)")
plt.xlabel("x")
plt.ylabel("f(x)")
```

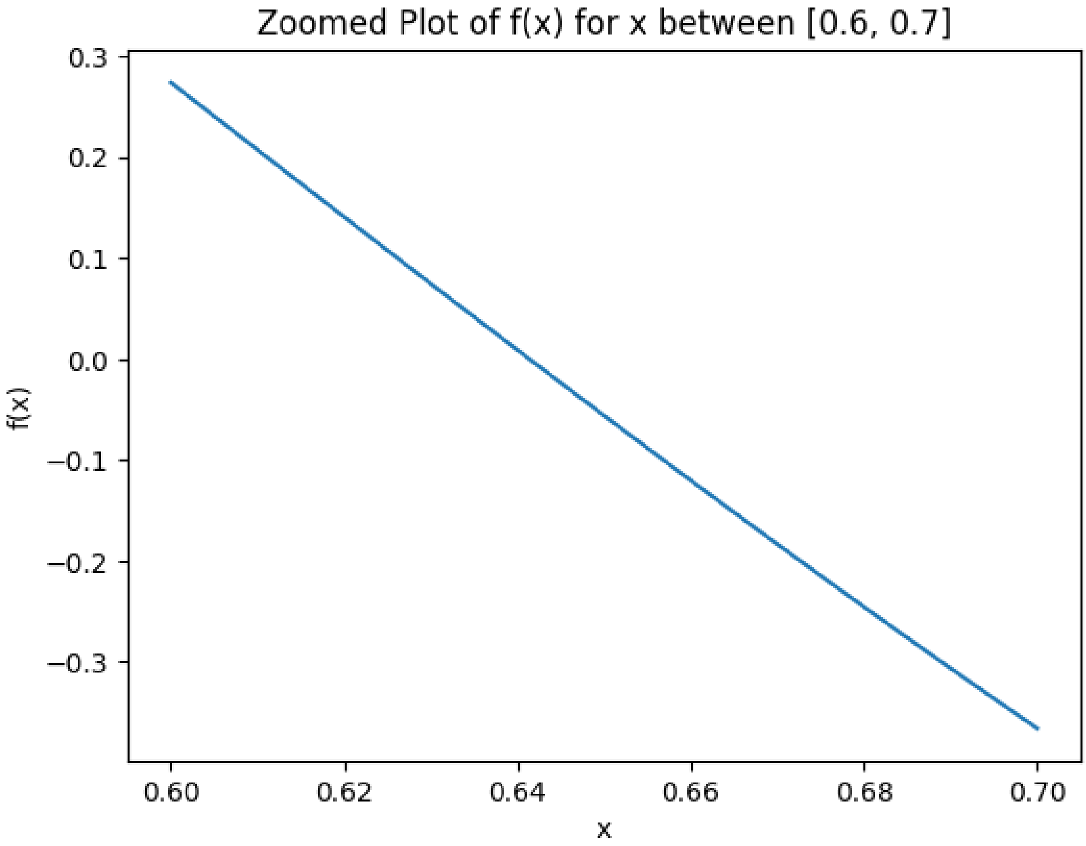
```
# 2.2
plt.hlines(0, -2, 2, colors="red", linestyles="dashed")

plt.show()
```



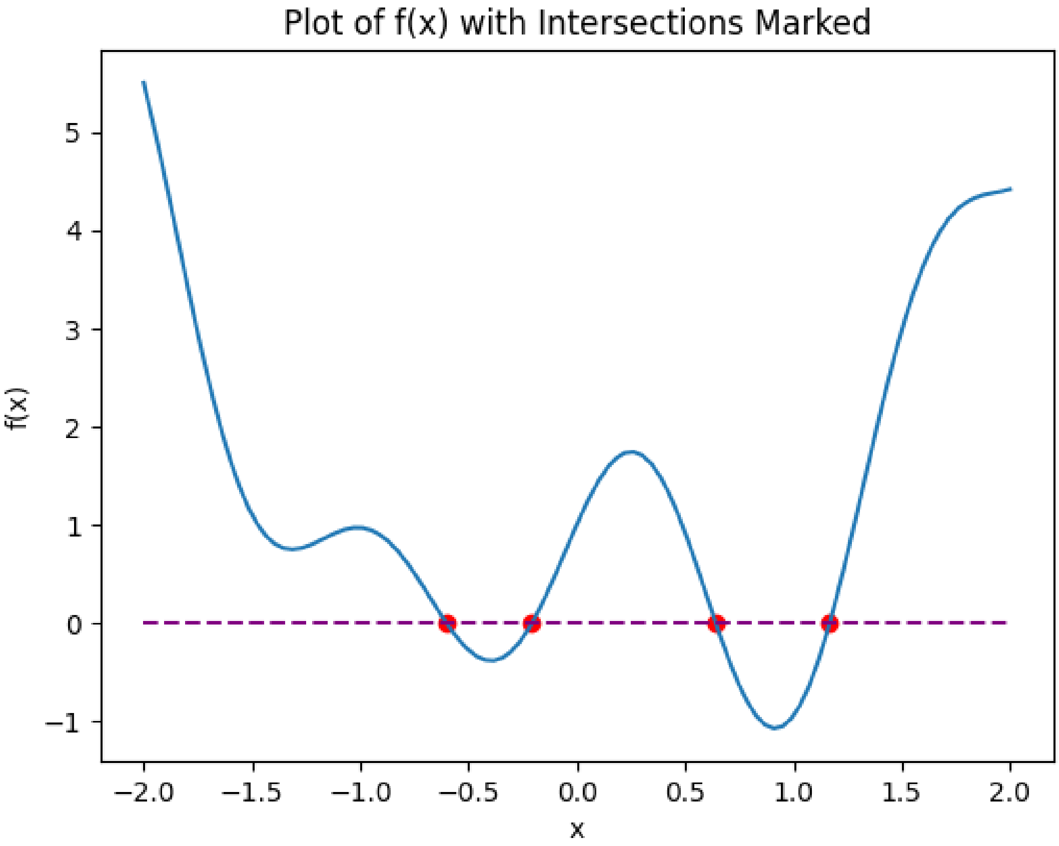
1.2.1 2.3

There are four intersections of f(x) and the horizontal line at x = 0.

```
[ ]: #2.3
     x_zoom = np.linspace(0.6, 0.7, 100)
     y_zoom = f(x_zoom)

     plt.plot(x_zoom, y_zoom)
     plt.title("Zoomed Plot of f(x) for x between [0.6, 0.7]")
     plt.xlabel("x")
     plt.ylabel("f(x)")
     plt.show()
```



[ ]:
```
# 2.4
plt.plot(x, y)
plt.hlines(0, -2, 2, colors="purple", linestyles="dashed")
plt.title("Plot of f(x) with Intersections Marked")
plt.xlabel("x")
plt.ylabel("f(x)")
plt.scatter(1.16, 0, color="red")
plt.scatter(0.64, 0, color="red")
plt.scatter(-0.21, 0, color="red")
plt.scatter(-0.6, 0, color="red")
plt.show()
```



1.3 Problem 3. (20 points).

In this problem, you will draw a 3D ellipsoid in Matplotlib. To do so, you need to parameterize the ellipsoid equation in terms of two angles (azimuthal angle θ and polar angle ϕ and use the parametric equation of the ellipsoid.

The general equation for an ellipsoid is:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1,$$

where: a, b, c are the semi-axes of the ellipsoid along the x, y, and z directions, respectively.

To plot the ellipsoid, you can parametrize it in spherical coordinates as follows:

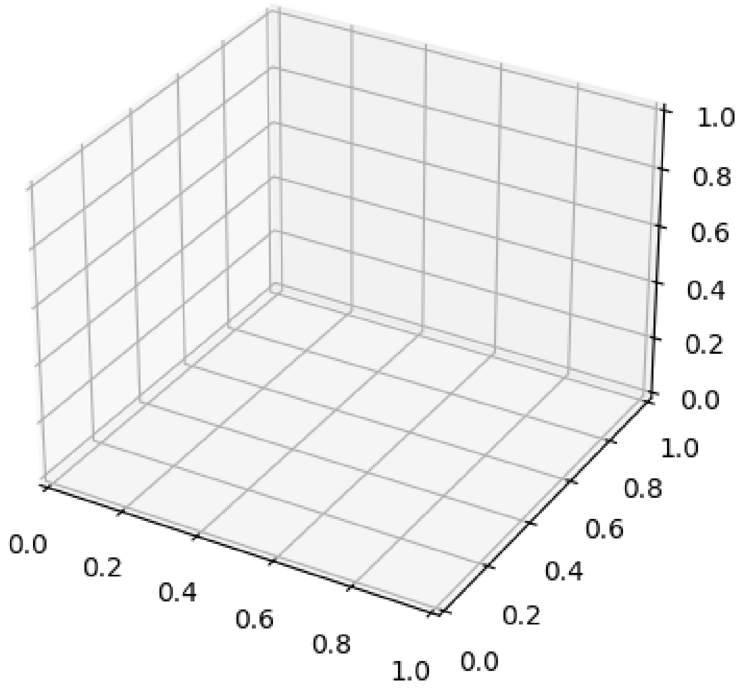
$$\begin{aligned} x = && a\sin(\phi)\cos(\theta) \\ y = && b\sin(\phi)\sin(\theta) \\ z = && c\cos(\phi), \end{aligned}$$

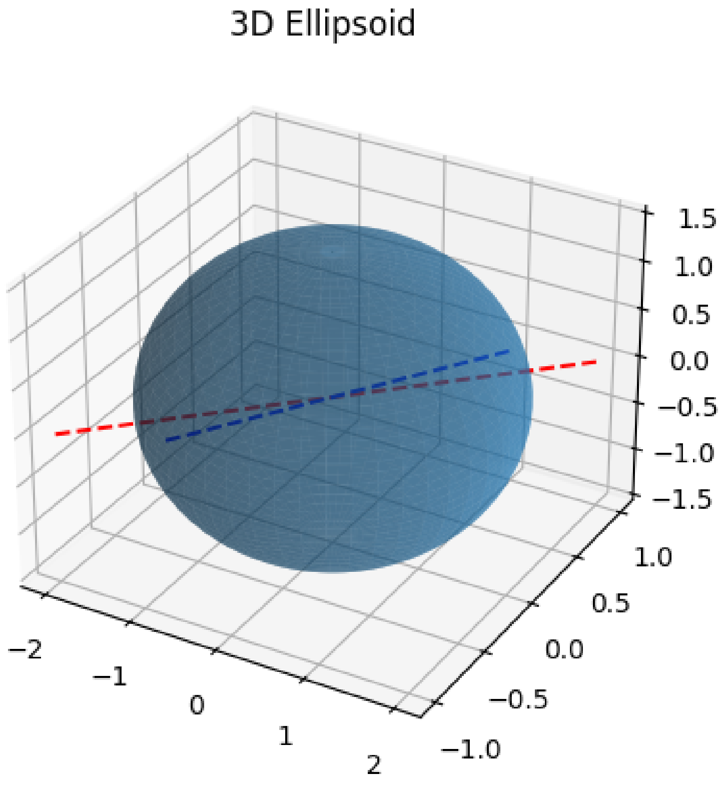
where: θ varies from 0 to 2π, ϕ varies from 0 to π.

Let $a = 2, b = 1, c = 1.5$. Make a 3D plot that - show the ellipsoid (10 points) - add appropriate title and labels for each axis (5 points) - add two planar cross-sections of the ellipsoid: one parallel to the XY-plane (at $z = 0$) and another parallel to the XZ-plane (at $y = 0$). Use dashed lines to indicate these cross-sections. (5 points)

[ ]:

```
ax = plt.axes(projection="3d")

a = 2
b = 1
c = 1.5

theta = np.linspace(0, 2*np.pi, 100)
phi = np.linspace(0, np.pi, 100)

theta, phi = np.meshgrid(theta, phi)

x = a * np.sin(phi) * np.cos(theta)
y = b * np.sin(phi) * np.sin(theta)
z = c * np.cos(phi)

fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")
ax.plot_surface(x, y, z, alpha=0.5)

x_cross = np.linspace(-a, a, 100)
y_cross = np.linspace(-b, b, 100)
z_cross = np.zeros_like(x_cross)

ax.plot(x_cross, y_cross, z_cross, linestyle="dashed", color="red")
ax.plot(x_cross, z_cross, y_cross, linestyle="dashed", color="blue")
ax.set_zlabel("Z")
ax.set_title("3D Ellipsoid")




plt.draw()
plt.show()
```


1.0
0.8
0.6
0.4
0.2
0.0
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


3D Ellipsoid
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
1.0
0.5
0.0
−0.5
−1.0
−2
−1
0
1
2

1.4 Problem 4 (20 points): You are provided with a dataset containing random information about 50 different cars. The dataset includes the following columns:

- Horsepower: The horsepower of the car.
- Weight: The weight of the car in pounds.
- MPG: The miles per gallon (fuel efficiency) of the car.
- ModelYear: The model year of the car.
- Brand

[ ]:

```
cars = {
    'Horsepower': np.random.randint(60, 200, 50),
    'Weight': np.random.randint(1500, 4000, 50),
    'MPG': np.random.uniform(10, 40, 50),
    'ModelYear': np.random.randint(1970, 2020, 50),
    'Brand': np.random.choice(['Toyota','Honda','KIA'],50)
}
cars = pd.DataFrame(cars)
cars.head(3)
```

[ ]:

```
   Horsepower  Weight        MPG  ModelYear   Brand
0         155    3752  27.942459       1996  Toyota
1         156    3725  29.392193       1982   Honda
2          60    3540  33.480513       2000  Toyota
```

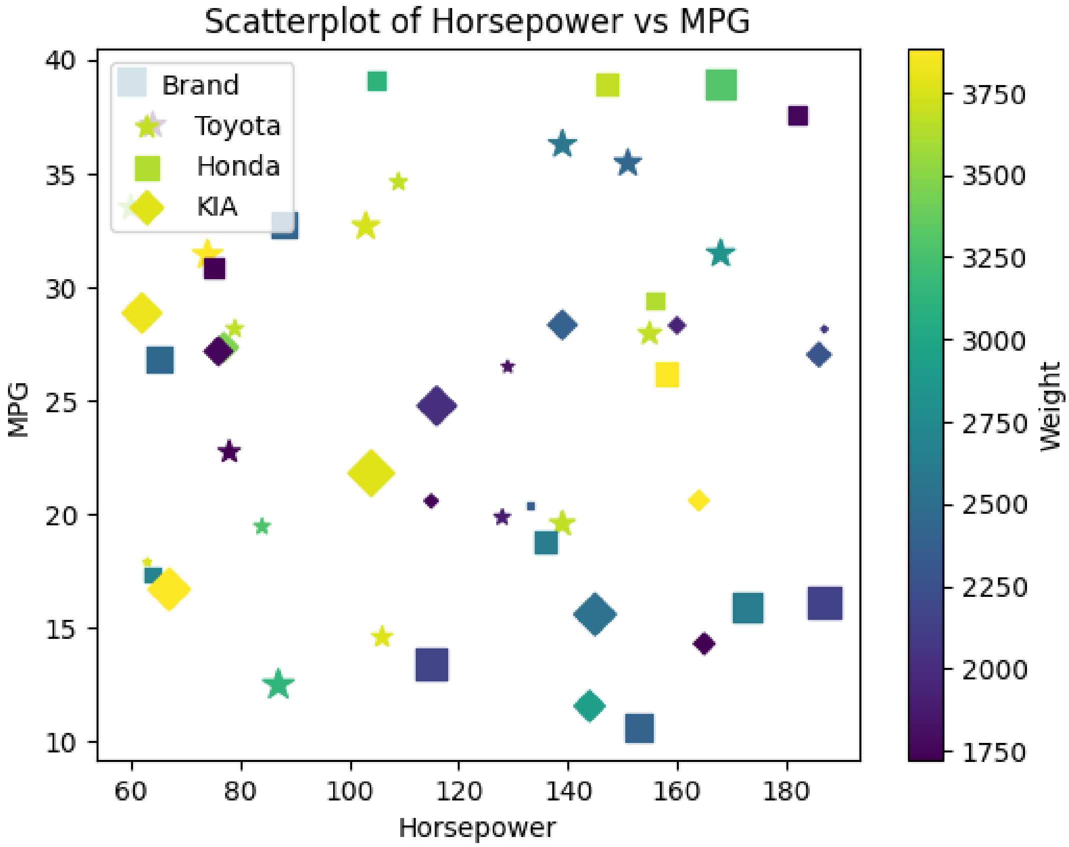
Create a scatter plot to include the information of all five columns in `cars`.

- Use the shape of markers to represent car brands (3 points)
- Use Color to represent the weight of the car (3 points)
- Use x-axis to represent Horsepower (3 points)
- Use y-axis to represent MPG (3 points)
- Use the size of markers to represent the ModelYear (3 points)

Add titles, legend, labels to the graph. (5 points)

[ ]:

```
marker_shapes = {"Toyota": "*", "Honda": "s", "KIA": "D"}


for brand, marker in marker_shapes.items():
    brand_data = cars[cars["Brand"] == brand]
    plt.scatter(
        x=brand_data["Horsepower"],
        y=brand_data["MPG"],
        c=brand_data["Weight"],
        s= 3 * (brand_data["ModelYear"] - (cars["ModelYear"].min()-1)),
        marker=marker,
        #cmap="viridis",
        #alpha=0.6,
        label=brand
    )

plt.title("Scatterplot of Horsepower vs MPG")
plt.xlabel("Horsepower")
plt.ylabel("MPG")
plt.colorbar(label="Weight")
plt.legend(marker_shapes.keys(), title="Brand", loc="upper left")

plt.show()
```



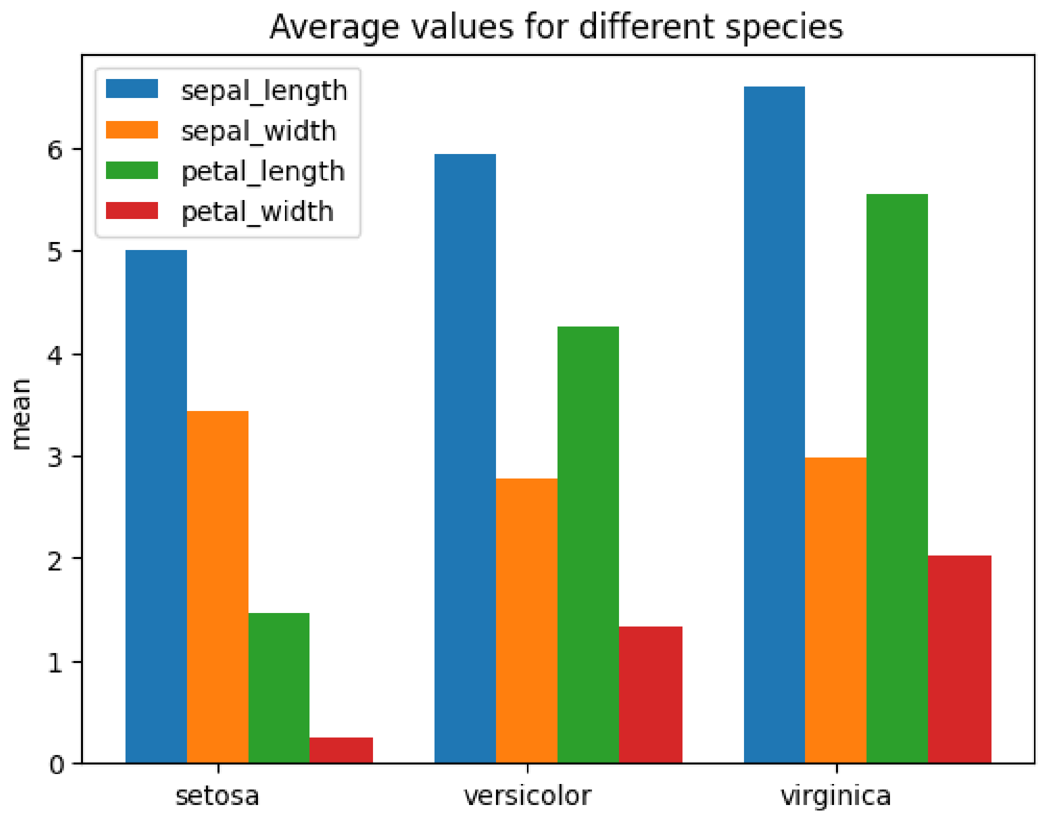
1.5 Problem 5: Recreate the same plot using sns (15 points)

Recall that in the lecture, we use the following code to create a bar plot for the iris dataset

[ ]:

```
df = pd.read_csv('iris.csv')
avg_iris = df.groupby("species").mean()
width=0.2
```

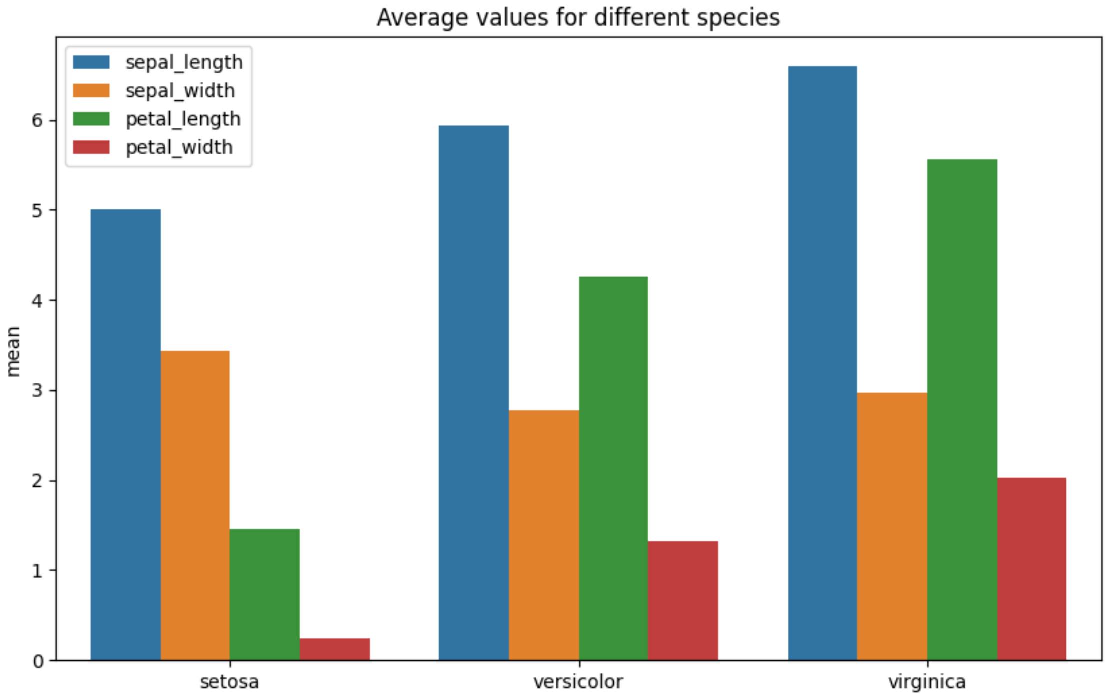
```
multiplier=0
x = np.arange(3)
for i in range(avg_iris.shape[1]):
    offset = width*multiplier
    plt.bar(x + offset, avg_iris.iloc[:,i], width, label=avg_iris.columns[i])
    multiplier += 1
plt.legend()
plt.ylabel('mean')
plt.title('Average values for different species')
plt.xticks(x + width, avg_iris.index);
```



Create the same plot using `sns.barplot` function

[ ]:

```
avg_iris_long = avg_iris.reset_index().melt(id_vars="species",␣
 ↪var_name="feature", value_name="mean")

plt.figure(figsize=(10,6))
sns.barplot(data=avg_iris_long, x="species", y="mean", hue="feature")


plt.title("Average values for different species")
plt.ylabel("mean")
plt.xlabel("")
plt.legend()
plt.show()
```



```
[ ]:
```