INTRODUCTION TO DATA SCIENCE

ASSIGNMENT 1:

NAME: **HAMMAD ZAFAR**

REG#: **SP20-BCS-136-B**

SUBMITTED TO: **Dr. Muhammad Sharjeel**

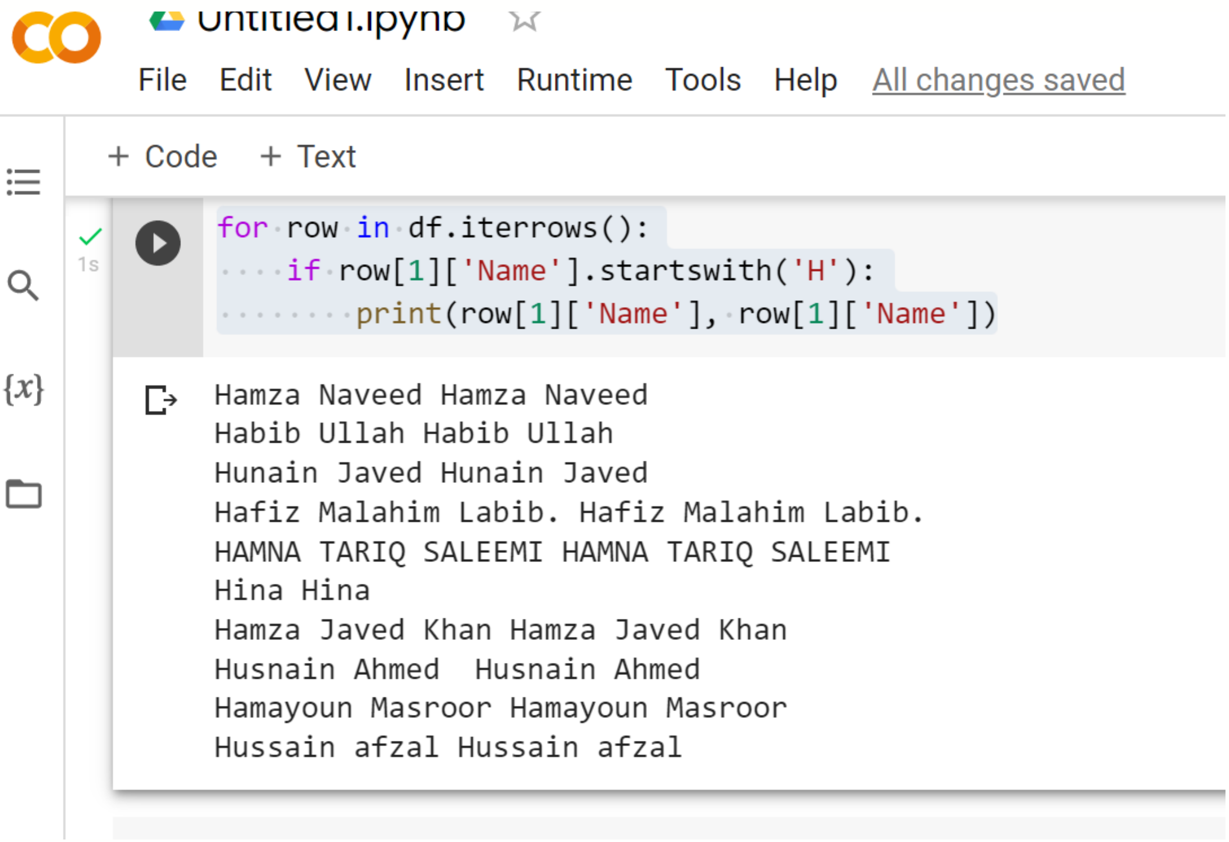
OCT 21,2022
CUI LAHORE

Load the dataset (csv file) into a Pandas DataFrame.

```
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
from scipy.stats import linregress
from scipy import stats
from google.colab import drive

drive.mount('/content/drive/')

df = pd.read_csv('/content/drive/My Drive/Colab Notebooks/the-hello-dataset-fa22.csv')
```

Print the list of all students whose first name starts with letter the 'H

```
for row in df.iterrows():
    if row[1]['Name'].startswith('H'):
        print(row[1]['Name'], row[1]['Name'])
```



```
for row in df.iterrows():
    if row[1]['Name'].startswith('H'):
        print(row[1]['Name'], row[1]['Name'])
```

```
Hamza Naveed Hamza Naveed
Habib Ullah Habib Ullah
Hunain Javed Hunain Javed
Hafiz Malahim Labib. Hafiz Malahim Labib.
HAMNA TARIQ SALEEMI HAMNA TARIQ SALEEMI
Hina Hina
Hamza Javed Khan Hamza Javed Khan
Husnain Ahmed  Husnain Ahmed
Hamayoun Masroor Hamayoun Masroor
Hussain afzal Hussain afzal
```

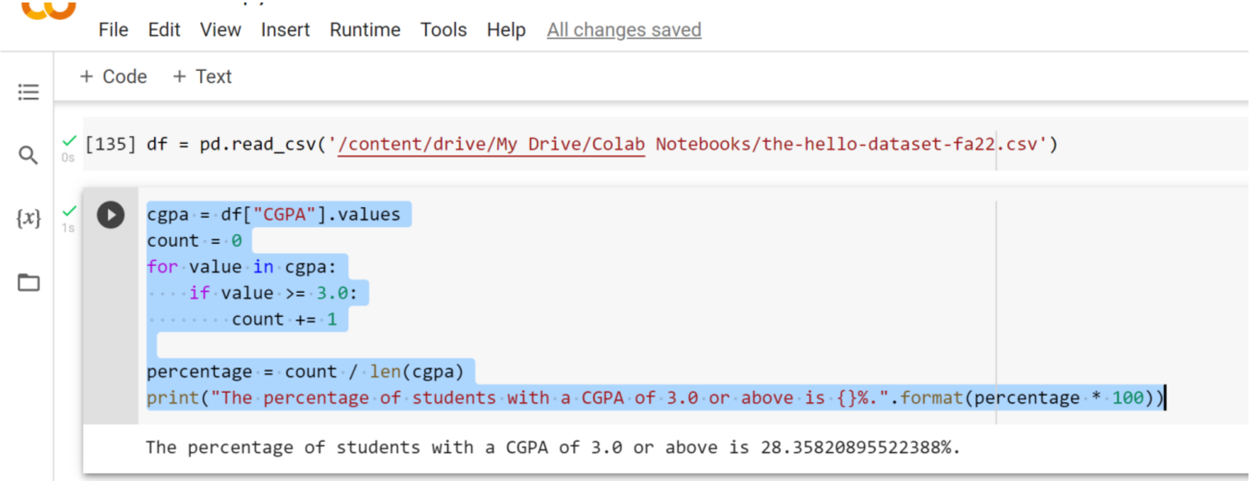
Print the total number of students who have a three words name (first-middle-surname).

```
print(len(df[df['Name'].str.split().str.len() == 3]))
```

```
print(len(df[df['Name'].str.split().str.len() == 4]))
```

Print the percentage of students who have a CGPA of 3.0 or above.

```
cgpa = df["CGPA"].values
count = 0
for value in cgpa:
    if value >= 3.0:
        count += 1

percentage = count / len(cgpa)
print("The percentage of students with a CGPA of 3.0 or above is {}%.".format(percentage * 100))
```



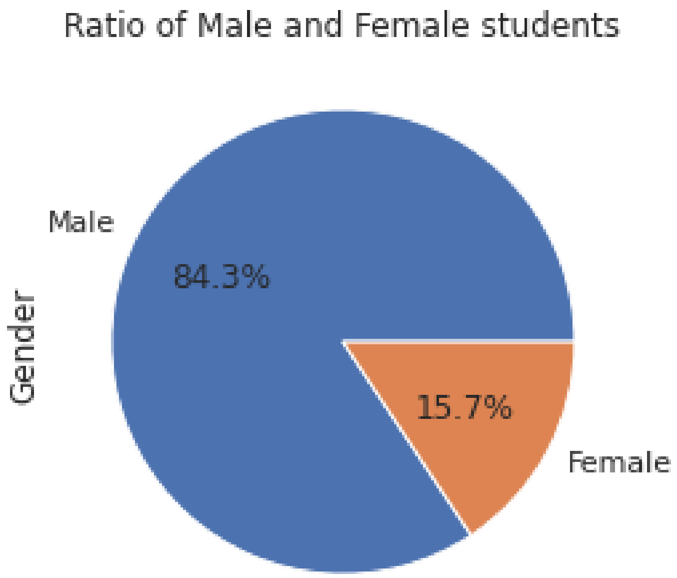
```
[135] df = pd.read_csv('/content/drive/My Drive/Colab Notebooks/the-hello-dataset-fa22.csv')
```

```
cgpa = df["CGPA"].values
count = 0
for value in cgpa:
    if value >= 3.0:
        count += 1

percentage = count / len(cgpa)
print("The percentage of students with a CGPA of 3.0 or above is {}%.".format(percentage * 100))
```

```
The percentage of students with a CGPA of 3.0 or above is 28.35820895522388%.
```

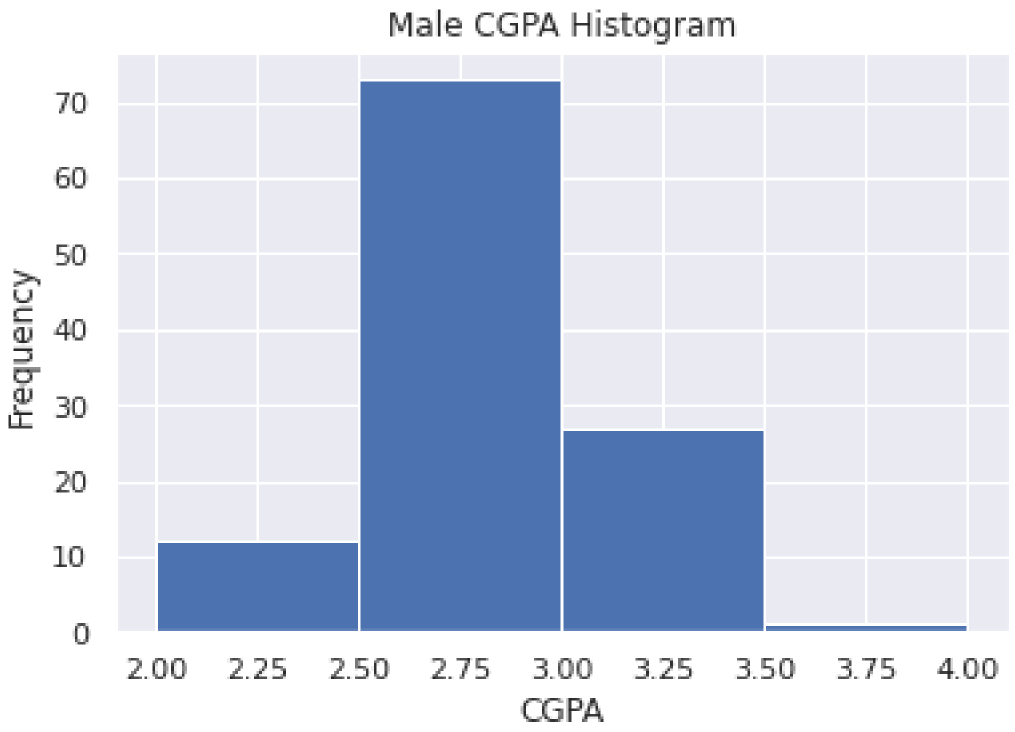
Plot a pie chart to show the ratio of male and female students.

```
df["Gender"].value_counts().plot(kind="pie",title="Ratio of Male and Female students",autopct="%1.1f%%")
```



Plot the CGPA of all male students on a histogram with intervals 2.0-2.5, 2.6-3.0, 3.1-3.5, 3.6-4.0.

```
plt.hist(df['CGPA'][df['Gender']=='Male'],bins=[2.0,2.5,3.0,3.5,4.0])
plt.xlabel("CGPA")
plt.ylabel("Frequency")
plt.title("Male CGPA Histogram")
plt.show()
```



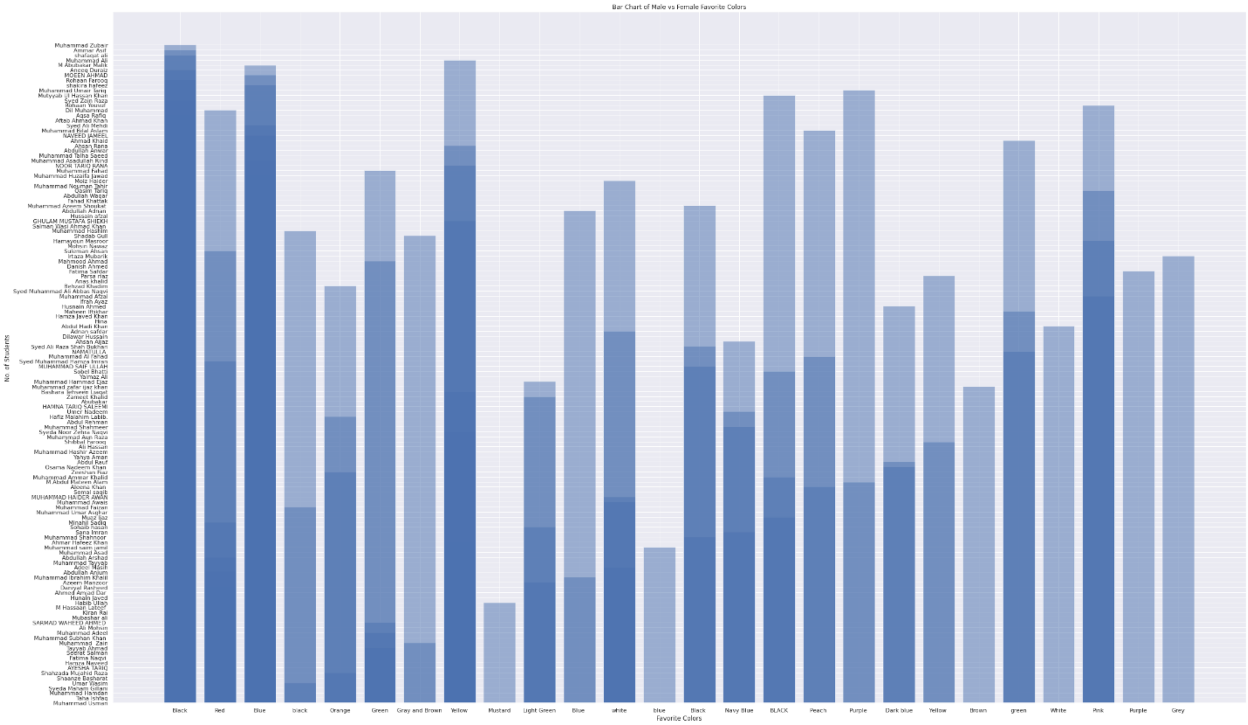
Plot the HSSC-1 marks of all male vs female students on a scatter plot.

```
male_students = df.loc[(df['Gender'] == 'M')]
female_students = df.loc[(df['Gender'] == 'F')]
plt.figure(figsize=(10,10))
plt.scatter(male_students['HSSC-
1'], male_students['Gender'], label = 'Male', color = 'b')
plt.scatter(female_students['HSSC-
1'], female_students['Gender'], label = 'Female', color = 'r')
plt.xlabel('Marks in HSSC-1')
plt.ylabel('Gender')
plt.title('Marks of Male and Female Students in HSSC-1')
plt.legend()
plt.show()
```

Plot the favorite colors of male vs female students on a bar chart.

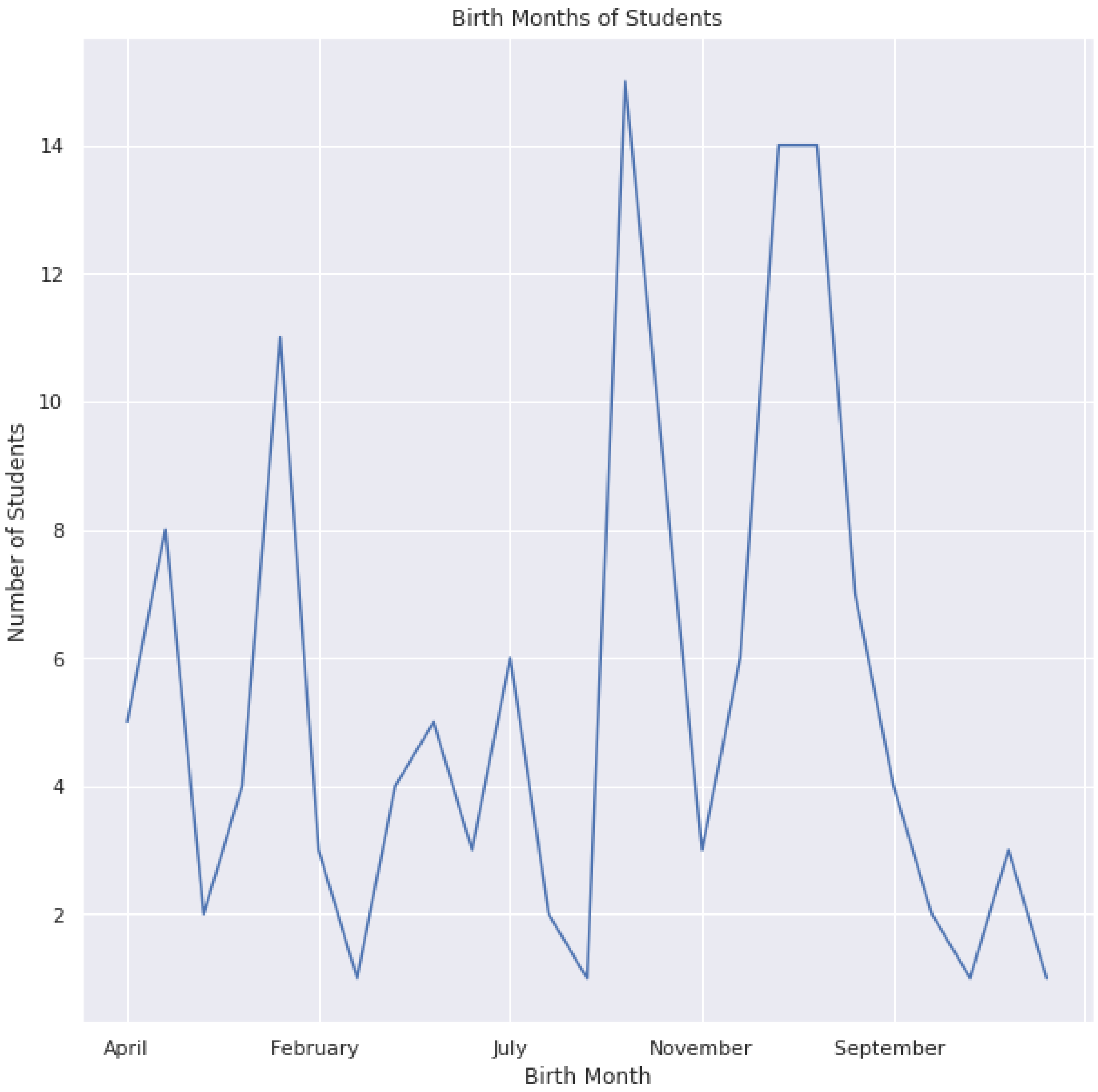
```
sns.set()
plt.figure(figsize=(40,25))
```

```
x = df['FavoriteColor']
y = df['Gender']
z = df['Name']

plt.bar(x, z, align='center', alpha=0.5)

plt.xlabel('Favorite Colors')
plt.ylabel('No. of Students')
plt.title('Bar Chart of Male vs Female Favorite Colors')

plt.show()
```



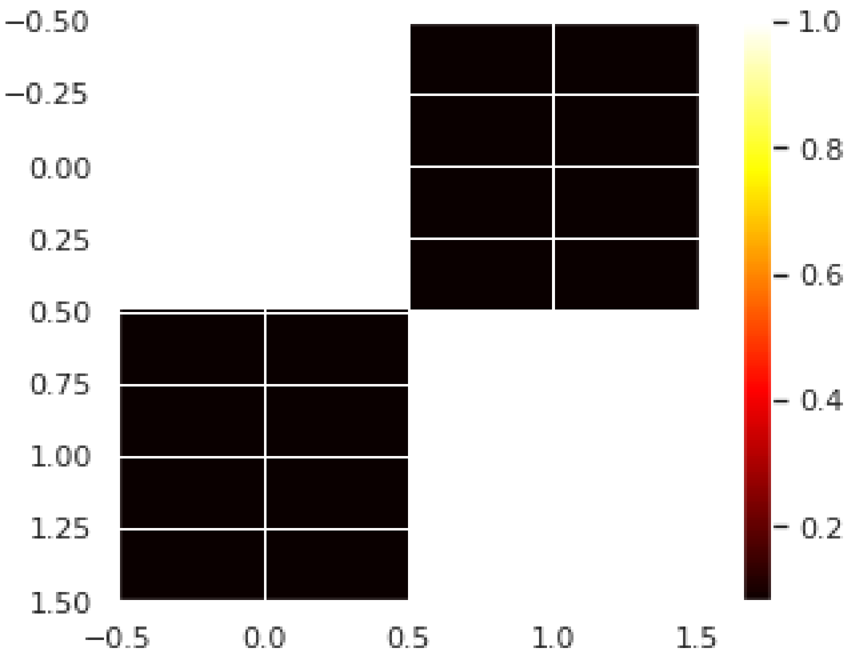
Plot line chart of students and their birth months

```
sns.set()
plt.figure(figsize=(10,10))

df['BirthMonth'].value_counts().sort_index().plot(kind='line')

plt.xlabel('Birth Month')

plt.ylabel('Number of Students')
plt.title('Birth Months of Students')

plt.show()
```



Create a correlation matrix between HSSC-1 and HSSC-2 marks and then plot on a heatmap

```
x = df['HSSC-1(Norm-Values)']
y = df['HSSC-2(Norm-Values)']

corr_matrix = np.corrcoef(x, y)

plt.imshow(corr_matrix, cmap='hot', interpolation='nearest')
plt.colorbar()
plt.show()
```


−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
1.25
1.50
−0.5
0.0
0.5
1.0
1.5
1.0
0.8
0.6
0.4
0.2