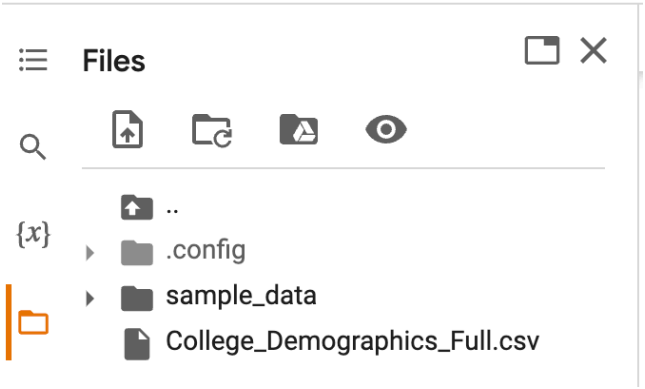
1. Downloading & uploading the dataset

1. Open the dataset by clicking on this link
2. Download the dataset by clicking on the download icon
3. To upload the dataset on Google Colab, first click the folder icon near the top left of this page
4. Click the upload button and select the "College_Demographics_Full.csv" file you previously downloaded

Once you've completed these steps, your Files should look like this:



↓ Click "run" to clean the dataset

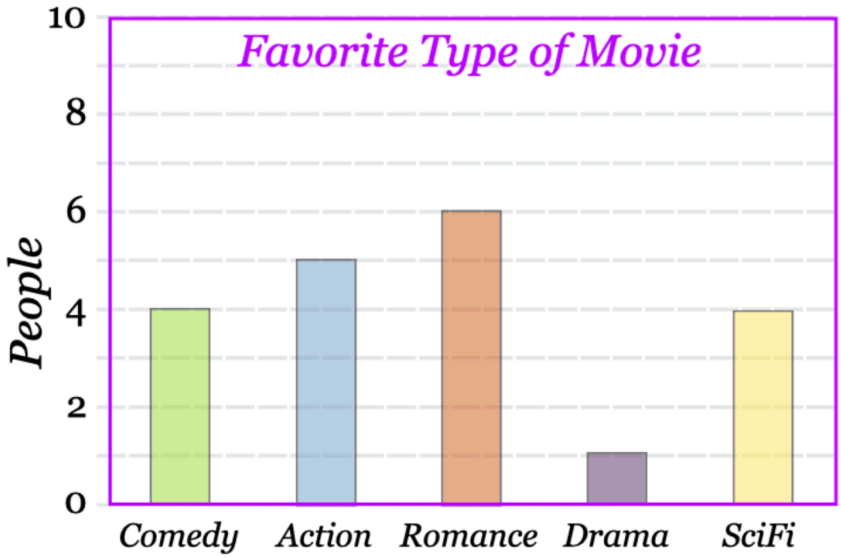
Show code

```
# visualize a dataset below
df_hbcu
```

2. Data Visualization

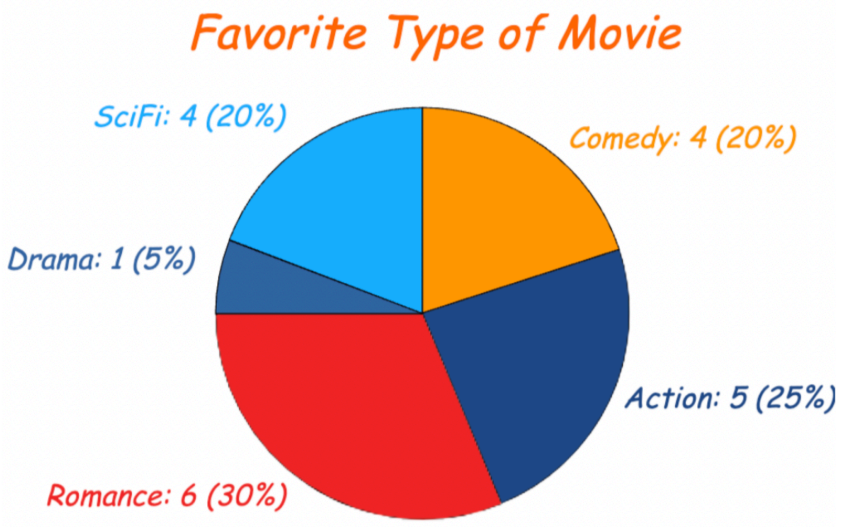
Bar Graph

A bar graph shows data using rectangular bars with varying heights (values).



Pie Chart

A pie chart shows a divided circle with each sector representing various proportions of the whole (value).



To plot graphs and charts we import the `matplotlib.pyplot` library.

We have already imported this library in the first cell, but you can check that you have imported it below:

↓ Have you imported the correct libraries? (run this cell)

Show code

↓ When should we use a bar graph to visualize our data? (run this cell)

input_answer: Choose an answer

Show code

3. Visualizing College Admissions Data

EXAMPLE: If we want to create a bar plot of the total undergraduate enrollment within HBCUs we would write the following code:

```
# use the df_hbcu DataFrame that we created in the first code cell
ax = df_hbcu.plot.bar(x = "Name", y = "ungrad total")
ax.set_xlabel("University Name")
ax.set_ylabel("Undergrad Enrollment Total")
ax.set_title("Students enrolled in HBCUs")

# visualize the graph (ax creates the axes)
ax
```

Practice: Adding Color

We can also color each column in the bar graph by creating a list of colors:

```
'''
Exercise 1)
'''
# list of colors
colors = ["orange", "green", "red", "cyan", "purple", "Goldenrod", "blue"]

# use the df_hbcu DataFrame that we created in the first code cell
ax = ?.plot.bar(x = ?, y = ?, color = colors) # set x and y equal to the column t
?.set_xlabel("University Name")
?.set_ylabel("Undergrad Enrollment Total")
?.set_title("Students enrolled in HBCUs")

# visualize the graph
ax.get_legend().remove() # remove the legend
ax
```

↓ Which *character* is used for the color black? (run this cell)

input_answer: Choose an answer

Show code

```
'''
Exercise 2) Create a bar graph with undergraduate totals for colleges in Washingt
'''
# use the df_wash DataFrame that we created in the first code cell
? = ?.plot.bar(x = ?, y = ?)
wash_plot.?("University Name") # use the function set_xlabel
wash_plot.?("Undergrad Enrollment Total") # use the function set_ylabel
wash_plot.?("Students enrolled in Washington State Colleges") # use the function

# visualize the graph
wash_plot
```

↓ Which University has the largest undergraduate enrollment? (run this cell)

input_answer: Choose an answer

Show code

↓ Which University has the smallest undergraduate enrollment? (run this cell)

input_answer: Choose an answer

Show code

Practice: Adding color from the `matplotlib.pyplot` list of colors:

```
'''
Exercise 3) Create a colored bar graph with undergraduate totals for colleges in \
'''
# create a list of colors, you can add as many colors as you want to the list
colors = [?, ?, ?, ... ?]

# use the df_wash DataFrame that we created in the first code cell
colors_wash_plot = ?.plot.?(x = ?, y = ?, ? = ?)
colors_wash_plot.?("University Name") # use the function set_xlabel
colors_wash_plot.?("Undergrad Enrollment Total") # use the function set_ylabel
colors_wash_plot.?("Students enrolled in Washington State Colleges") # use the fu

# visualize the graph
colors_wash_plot.get_legend().remove() # remove the legend
colors_wash_plot
```

4. Creating Pie Charts

↓ When should we use a pie chart to visualize our data? (run this cell)

input_answer: Choose an answer

Show code

EXAMPLE: If we want to create a pie chart of the total undergraduate enrollment in HBCUs we would write the following code:

```
# use the df_hbcu DataFrame that we created in the first code cell
ax = df_hbcu.plot.pie(y = "ungrad total")

# visualize the graph
ax
```

Practice: Do the same thing but for the `df_wash` DataFrame undergraduate enrollment.

```
'''
Exercise 4) Create a pie chart for the df_wash DataFrame for total undergraduate
'''
# use the df_wash DataFrame that we created in the first code cell
wash_pie = ?.plot.pie(y = ?)

# visualize the graph
?
```

EXAMPLE: You can also add percentages to the pie chart using the `autopct` parameter:

```
# use the df_hbcu DataFrame that we created in the first code cell
ax = df_hbcu.plot.pie(y = "ungrad total", autopct = '%1.0f%%')

# visualize the graph
ax
```

Practice: Add the percentages for the Washington state universities using the `df_wash` DataFrame.

```
'''
Exercise 5) Create a pie chart for the df_wash DataFrame for total undergraduate
            using the autopct parameter to show the percentages of each sector of
'''
# use the df_wash DataFrame that we created in the first code cell
? = ?.plot.?(? = ?, autopct = ?)

# visualize the graph
wash_pie
```

EXAMPLE: The legend is blocking the view of our beautiful pie chart, let's remove it!

```
# use the df_hbcu DataFrame that we created in the first code cell
ax = df_hbcu.plot.pie(y = "ungrad total", autopct ='%1.0f%%')

# visualize the graph
ax.get_legend().remove()
ax
```

Practice: Remove the legend for the Washington pie chart.

```
'''
Exercise 6) Create a pie chart for the df_wash DataFrame for total undergraduate
            using the autopct parameter to show the percentages of each sector of
            and removing the legend
'''
# use the df_wash DataFrame that we created in the first code cell
wash_pie = ?.?.pie(y = ?, autopct =?)

# visualize the graph
wash_pie.?.? # remove the legend
wash_pie
```

EXAMPLE: The ungrad totals don't help us understand our chart very well. Let's add labels for the names of the colleges instead, and a title:

```
# use the df_hbcu DataFrame that we created in the first code cell
ax = df_hbcu.plot.pie(y = "ungrad total", autopct = '%1.0f%%', labels = df_hbcu.l

# visualize the graph
ax.get_legend().remove() # remove the legend
ax
```

Practice: Let's remove the ungrad totals, add labels for the names of the colleges in Wahington, and add a title.

```
'''
Exercise 7) Create a pie chart for the df_wash DataFrame for total undergraduate
            using the autopct parameter to show the percentages of each sector of
            removing the legend, and adding a title
'''
# use the df_wash DataFrame that we created in the first code cell
wash_pie = df_wash.?.?(y = "ungrad total", autopct = '%1.0f%%', labels = ?.loc[:,

# visualize the graph
? # remove the legend
wash_pie
```

5. More Fun with Charts!

You just learned a lot about bar graphs and pie charts.

Coding more pie charts

Let's see if we can create a pie chart with the average percentages of males and females enrolled in Washington state colleges.

First, we'll need to calculate the average percentages of male and female students enrolled in Washington state colleges:

```
'''
Exercise 8) Calculate the average percentage of men and women enrolled in Washing
'''
# locate the column from df_wash for percentage of men enrolled
df_wash_men_avg = df_wash.loc[:, ?].mean()
print(df_wash_men_avg)

# locate the column from df_wash for percentage of women enrolled
df_wash_women_avg = df_wash.loc[:, ?].mean()
print(df_wash_women_avg)

# create a seperate DataFrame with the averages
averages = [df_wash_men_avg, df_wash_women_avg] # list out the averages here
df_average_gender = pd.DataFrame(averages, columns=["Averages"])
```

Now we can create a pie chart using the new `df_average_gender` DataFrame:

```
# create a list for labeling the genders (example: Men, Women)
gender_labels = [?, ?]

# create a pie chart using the df_average_gender DataFrame
wash_pie_gender = ?.plot.?(y = ?, autopct ='%1.0f%%', labels = ?, ylabel='', titl

# visualize the pie chart
wash_pie_gender.get_legend().remove() # remove the legend
wash_pie_gender
```

Let's see if we can create a similar pie chart, but with the average percentages of each race enrolled in Washington state colleges.

Like before, we'll first need to calculate the average percentages of each race of students enrolled in Washington state colleges:

```
'''
Exercise 9) Calculate the average race demographics of students enrolled in Washi
'''
# create multiple different variables for the averages of each race
df_native_american = df_wash.loc[:,?].mean()
df_asian = df_wash.loc[:, ?].mean()
df_black = df_wash.loc[:, ?].mean()
df_latino = df_wash.loc[:, ?].mean()
df_islander = df_wash.loc[:, ?].mean()
df_white = df_wash.loc[:, ?].mean()
df_two_races = df_wash.loc[:, ?].mean()
df_unknown = df_wash.loc[:, ?].mean()

averages = [?, ?, ?, ?, ?, ?, ?, ?] # list out the averages here
df_average_race = pd.DataFrame(?, columns=['Averages']) # put the name of your li
```

Now we can create a pie chart using our new `df_average_race` DataFrame:

```
# create a list for the labels
race_labels = [?, ?, ?, ?, ?, ?, ?, ?] # create a list for labeling each race

# create a pie chart using the df_average_race DataFrame
wash_pie_race = ?.?.?(y = ?, ? ='%1.0f%%', labels = ?, ylabel='', ? = "Average Ra

# visualize the pie chart
wash_pie_race.get_legend().remove() # remove the legend
wash_pie_race
```

Coding more bar graphs

Let's create a bar graph with the average percentages of males and females enrolled in HBCUs.

First, we'll need to calculate the average percentages of male and female students enrolled in HBCUs.

```
'''
Exercise 10) Calculate the average percentage of men and women enrolled in HBCUs
'''
# locate the column from df_hbcu for percentage of men enrolled
df_hbcu_men_mean = df_hbcu.loc[:,?].mean()

# locate the column from df_hbcu for percentage of women enrolled
df_hbcu_women_mean = df_hbcu.loc[:, ?].mean()

# create a seperate DataFrame with the averages
averages = {'Gender' : ['Men', 'Women'],
            'Averages' :[df_hbcu_men_mean, df_hbcu_women_mean]}
df_average_gender_hbcu = pd.DataFrame(averages)

# visualize the new DataFrame
df_average_gender_hbcu
```

Now we can create a pie chart using our new `df_average_gender_hbcu` DataFrame:

```
# use the df_average_gender_hbcu DataFrame to create the bar graph
hbcu_bar_gender = ?.plot.?(x = ?, y = ?)
hbcu_bar_gender.?("Gender") # use the function set_xlabel
hbcu_bar_gender.?("Percent of Enrollment (%)") # use the function set_ylabel
hbcu_bar_gender.?("Average Gender Demographics in HBCUs") # use the function set_

# visualize the graph
hbcu_bar_gender.get_legend().remove() # remove the legend
hbcu_bar_gender
```

Now, make another bar graph showing the races of students enrolled in HBCUs

```
'''
Exercise 11) Calculate the average race demographics of students enrolled in HBCU
'''
# create multiple different variables for the averages of each race using df_hbcu
df_native_american = ?.loc[:,?].mean()
df_asian = ?.loc[:, ?].mean()
df_black = ?.loc[:, ?].mean()
df_latino = ?.loc[:, ?].mean()
df_islander = ?.loc[:, ?].mean()
df_white = ?.loc[:, ?].mean()
df_two_races = ?.loc[:, ?].mean()
df_unknown = ?.loc[:, ?].mean()

averages = {'Race' : [?, ?, ?, ?, ?, ?, ?, ?],  # add the names of the races as S
            'Averages' : [?, ?, ?, ?, ?, ?, ?, ?]} # add the variables for the av
df_average_race_hbcu = pd.DataFrame(averages) # put the name of your list here


# create the bar graph using the df_average_race_hbcu DataFrame
hbcu_bar_race = ?.plot.?(x = ?, y = ?)

# visualize the pie chart
hbcu_bar_race.get_legend().remove() # remove the legend
hbcu_bar_race
```

6. Making your own visualizations (optional)

Now you can try creating different types of bar graphs and pie charts on your own.

There are many creative ways to graph data, learn more here!

FOR THE NEXT LESSON CLICK HERE