Each team member should indicate either A or B.

A. I affirm that my contribution(s) to the lab work is my own, produced without help from any AI tool(s)

B. I affirm that my contribution(s) to the lab work has been produced with the help from AI tool(s)

Full Name	Date	A or B
Samuel Chan	8/11/24	A
Vijaya Madhulika	8/11/24	B
Zhou Runhe	8/11/24	A
Ramirez Sevilla Javier	9/11/24	B
Sheryl Ng Wenhui	9/11/24	A

By including this information in your jupyter notebook, you declare that the above affirmation made is true and that you have read and understood NTU's policy on the use of AI tools.

If any team member answered B, the team member(s) must indicate and replicate the table below for every instance AI tool(s) is used.

Declaration - Ramirez Sevilla Javier:	
Name of AI tool	ChatGPT
Input prompt	Given my for loop:

```
for row in Rows[1:]:
    values = Rows.strip().split(',')
    row_dictionary = {}
```

	I get the following error on this line: values = Rows.strip().split(','), AttributeError: 'list' object has no attribute 'strip'. What does that mean ?
Date generated	23/10/2024
Output generated	The error message **`AttributeError: 'list' object has no attribute 'strip'`** means that you're trying to use the `.strip()` method on an object of type `list`, which is not allowed. The `.strip()` method is used to remove leading and trailing whitespace from a string, but `Rows` is a list, not a string. That's why the error occurs. **Breakdown of the code:**

python

Copy code

```
for row in Rows[1:]:  # Iterating over the list
"Rows", skipping the first element

    values = Rows.strip().split(',')  # Attempting
to use .strip() on "Rows"

    row_dictionary = {}
```

What's happening here?

- **`Rows`** is a list, so you cannot directly apply `.strip()` on it.
- You're trying to split each element in `Rows` (which seems to be a list of strings) by commas, but you're mistakenly applying `.strip()` to the entire list, not the individual strings inside the list.

Fixing the issue:

You need to apply `.strip()` to **each row** inside the loop, since `row` is the individual element (string) of the `Rows` list.

Here is the corrected version of your code:

python

Copy code

```
for row in Rows[1:]:  # Iterating over the list
"Rows", skipping the first element

    values = row.strip().split(',')  # Apply
.strip() to the individual row (a string)

    row_dictionary = {}
```

Explanation:

- **`for row in Rows[1:]`**: This loops through all elements in `Rows`, skipping the first element (because of the `[1:]` slicing).

- **`row.strip().split(',')`**: This applies `.strip()` to remove any leading or trailing whitespace from the individual `row` (which is a string). Then `.split(',')` splits the string by commas into a list of values.

Example:

Suppose `Rows` is:

python

Copy code

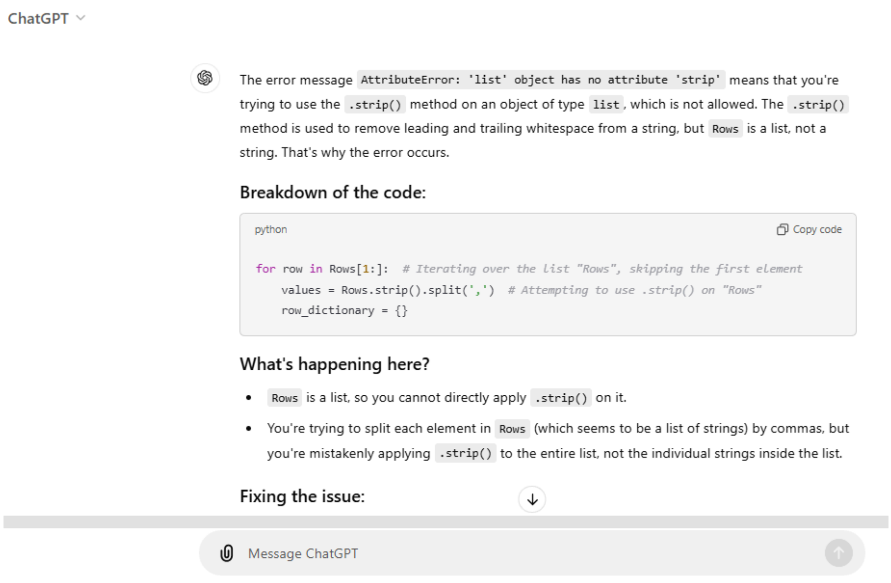
```
Rows = [

    "Name,Age,Location",

    "Alice,30,New York",

    "Bob,25,Los Angeles"

]
```

The loop will process each row, strip it of extra spaces (if any), and split it by commas:

- `"Alice,30,New York"` becomes `["Alice", "30", "New York"]`
- `"Bob,25,Los Angeles"` becomes `["Bob", "25", "Los Angeles"]`

Final thoughts:

You should always ensure that you apply methods like `.strip()` to strings, not to lists or other non-string objects.

Output screenshot	

The error message `AttributeError: 'list' object has no attribute 'strip'` means that you're trying to use the `.strip()` method on an object of type `list`, which is not allowed. The `.strip()` method is used to remove leading and trailing whitespace from a string, but `Rows` is a list, not a string. That's why the error occurs.

Breakdown of the code:

```python
for row in Rows[1:]:  # Iterating over the list "Rows", skipping the first element
    values = Rows.strip().split(',')  # Attempting to use .strip() on "Rows"
    row_dictionary = {}
```

What's happening here?

- `Rows` is a list, so you cannot directly apply `.strip()` on it.
- You're trying to split each element in `Rows` (which seems to be a list of strings) by commas, but you're mistakenly applying `.strip()` to the entire list, not the individual strings inside the list.

Fixing the issue:

Impact on submission	< Briefly explain which part of your submitted work was ChatGPT's response applied > My code changed from:

```
for row in Rows[1:]:
        values = Rows.strip().split(',')
        row_dictionary = {}
To:
for row in Rows[1:]:
        values = row.strip().split(',')
        row_dictionary = {}
```

As for my part : `Rows.strip().split(',')`, was giving an error message.

Name of AI tool	ChatGPT
Input prompt	Given my code to display a pie chart for Gender distribution:

```
Gender_frequency = {"Male": 0, "Female": 0}
for row in Team_data:
    gender = row["Gender"].strip()
    if gender == "M":
        Gender_frequency["Male"] += 1
    elif gender == "F":
        Gender_frequency["Female"] += 1

plt.pie(Gender_frequency.values(), labels =
Gender_frequency.keys(), colors=['lightblue', 'pink'])
plt.title(f"{Selected_Group}'s Gender Distribution in {team}")
```

	How do I use matplotlib plt.pie() to change the display from frequency to percentages and display it in a good way for visualisation purposes?
Date generated	03/11/2024
Output generated	To display percentages instead of raw frequencies in the pie chart, you can use the `autopct` parameter in `plt.pie()` to format the labels as percentages. Here's how you can adjust your code to include percentages: python Copy code

```
import matplotlib.pyplot as plt

# Given data
Gender_frequency = {"Male": 0, "Female": 0}
for row in Team_data:
    gender = row["Gender"].strip()
    if gender == "M":
        Gender_frequency["Male"] += 1
    elif gender == "F":
        Gender_frequency["Female"] += 1

# Plotting the pie chart with percentages
plt.pie(
    Gender_frequency.values(),
    labels=Gender_frequency.keys(),
    colors=['lightblue', 'pink'],
    autopct='%1.1f%%',  # Displays percentages with one decimal place
    startangle=140     # Rotates the start of the pie chart for better visualization
)

plt.title(f"{Selected_Group}'s Gender Distribution in {team}")
plt.show()
```

Explanation of the Changes

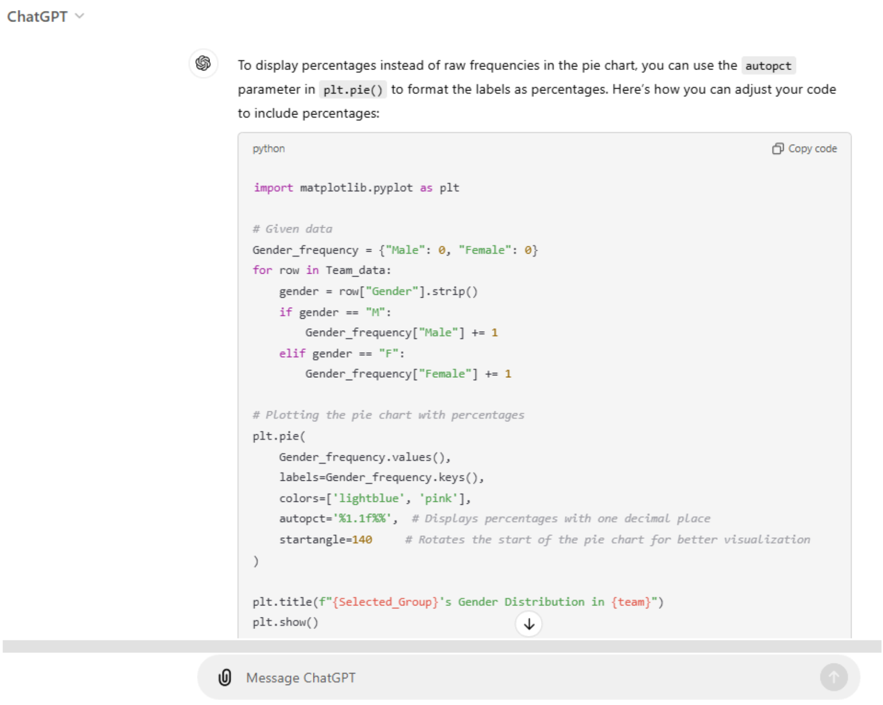
- **`autopct='%1.1f%%'`**: This adds the percentage labels to each slice. The format `%1.1f%%` means that the percentage will be displayed with one decimal place, followed by the percent symbol (`%%` escapes the `%` symbol in strings).

- `startangle=140`: This rotates the starting angle of the pie chart for a more aesthetically pleasing layout, especially useful if one category is much larger than the other.

Additional Tips for Visualization

- If you have a very small category that's hard to label, you can increase the font size using `plt.rcParams['font.size'] = 12`.
- Adjust `colors` to reflect color themes that are appropriate for the data

Output screenshot



To display percentages instead of raw frequencies in the pie chart, you can use the `autopct` parameter in `plt.pie()` to format the labels as percentages. Here's how you can adjust your code to include percentages:

```python
import matplotlib.pyplot as plt

# Given data
Gender_frequency = {"Male": 0, "Female": 0}
for row in Team_data:
    gender = row["Gender"].strip()
    if gender == "M":
        Gender_frequency["Male"] += 1
    elif gender == "F":
        Gender_frequency["Female"] += 1

# Plotting the pie chart with percentages
plt.pie(
    Gender_frequency.values(),
    labels=Gender_frequency.keys(),
    colors=['lightblue', 'pink'],
    autopct='%1.1f%%',  # Displays percentages with one decimal place
    startangle=140     # Rotates the start of the pie chart for better visualization
)

plt.title(f"{Selected_Group}'s Gender Distribution in {team}")
plt.show()
```

Impact on submission

My code changed from:

```
Gender_frequency = {"Male": 0, "Female": 0}
for row in Team_data:
    gender = row["Gender"].strip()
    if gender == "M":
        Gender_frequency["Male"] += 1
    elif gender == "F":
        Gender_frequency["Female"] += 1
```

```
plt.pie(Gender_frequency.values(), labels =
Gender_frequency.keys(), colors=['lightblue', 'pink'])
plt.title(f"{Selected_Group}'s Gender Distribution in
{team}")
```

To:

```
Gender_frequency = {"Male": 0, "Female": 0}
for row in Team_data:
   gender = row["Gender"].strip()
   if gender == "M":
      Gender_frequency["Male"] += 1
   elif gender == "F":
      Gender_frequency["Female"] += 1

plt.pie(Gender_frequency.values(), labels =
Gender_frequency.keys(), autopct='%.0f%%',
colors=['lightblue', 'pink'], startangle=90)
plt.title(f"{Selected_Group}'s Gender Distribution in
{team}")
```

Note that ChatGPT gave me:

```
plt.pie(Gender_frequency.values(), labels =
Gender_frequency.keys(), autopct='%1.1f%%',
colors=['lightblue', 'pink'], startangle=140)
```

However, I decided to change “`autopct=` “ and “`startangle=`” according to my own preference.

Name of AI tool	ChatGPT

Input prompt

Given my code to display a bar chart for schools:

```
School_frequency = {}
for row in Team_data:
    school = row["School"]
    if school in School_frequency:
        School_frequency[school] += 1
    else:
        School_frequency[school] = 1

plt.bar(School_frequency.keys(), School_frequency.values(),
color='green')
plt.title(f"{Selected_Group}'s School Distribution in {team}")
plt.xlabel('School')
```

	plt.ylabel('Number of Students') The graph displayed is too narrow, is difficult to see the labelled schools, is there a way to make the graph wider for better visualisation.
Date generated	06/11/2024
Output generated	Yes, you can adjust the size of the graph to make it wider by using the `figsize` parameter of `plt.figure()` before plotting the bar chart. Additionally, you can rotate the x-axis labels if they overlap, which is often the case with long or numerous labels. Here's how you can modify your code:

Adjusting the Figure Size and Rotating Labels

python
Copy code

```python
import matplotlib.pyplot as plt

School_frequency = {}
for row in Team_data:
    school = row["School"]
    if school in School_frequency:
        School_frequency[school] += 1
    else:
        School_frequency[school] = 1

# Set the figure size to make the chart wider
plt.figure(figsize=(12, 6))  # 12 units wide, 6 units tall

# Plot the bar chart
plt.bar(School_frequency.keys(), School_frequency.values(), color='green')

# Add title and labels
plt.title(f"{Selected_Group}'s School Distribution in {team}")
plt.xlabel('School')
plt.ylabel('Number of Students')
```

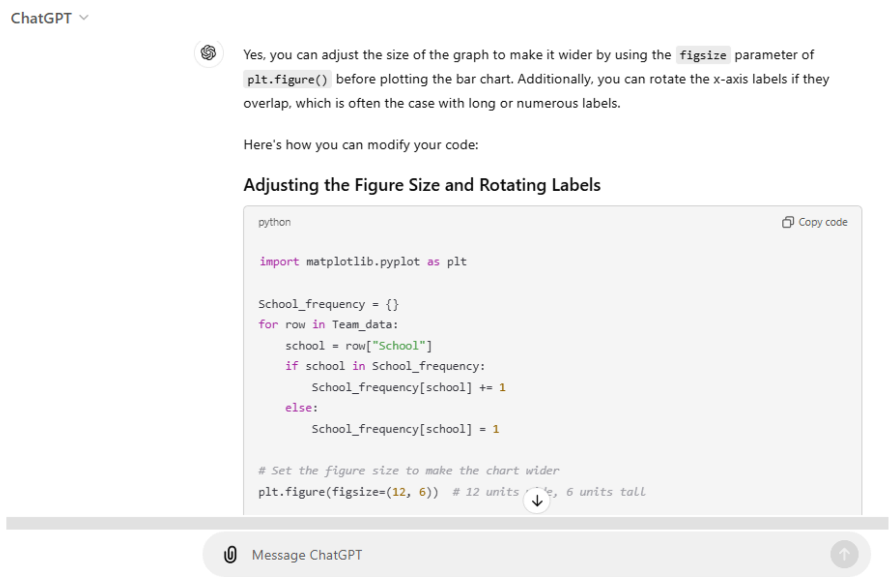
```
# Rotate x-axis labels for better visibility
plt.xticks(rotation=45, ha='right')  # Rotate labels by 45 degrees and align them to the right

# Display the chart
plt.show()
```

Key Changes:

1. **`plt.figure(figsize=(12, 6))`**: Adjust the size of the figure, making it 12 units wide and 6 units tall. You can tweak these values as per your needs.
2. **`plt.xticks(rotation=45, ha='right')`**: Rotates the x-axis labels by 45 degrees to ensure they fit well and are easier to read. The `ha='right'` argument helps align the text to the right of the labels, preventing them from overlapping with the bars.

This should make the chart wider and the labels clearer for better visualization!

Output screenshot



Yes, you can adjust the size of the graph to make it wider by using the `figsize` parameter of `plt.figure()` before plotting the bar chart. Additionally, you can rotate the x-axis labels if they overlap, which is often the case with long or numerous labels.

Here's how you can modify your code:

Adjusting the Figure Size and Rotating Labels

```python
import matplotlib.pyplot as plt

School_frequency = {}
for row in Team_data:
    school = row["School"]
    if school in School_frequency:
        School_frequency[school] += 1
    else:
        School_frequency[school] = 1

# Set the figure size to make the chart wider
plt.figure(figsize=(12, 6))  # 12 units wide, 6 units tall
```

Impact on submission

My code changed from:

```
School_frequency = {}
for row in Team_data:
    school = row["School"]
    if school in School_frequency:
        School_frequency[school] += 1
    else:
        School_frequency[school] = 1

plt.bar(School_frequency.keys(),
School_frequency.values(), color='green')
plt.title(f"{Selected_Group}'s School Distribution in
{team}")
plt.xlabel('School')
plt.ylabel('Number of Students')
```

To:

```
School_frequency = {}
for row in Team_data:
   school = row["School"]
   if school in School_frequency:
      School_frequency[school] += 1
   else:
      School_frequency[school] = 1

plt.figure(figsize=(10, 5))
plt.bar(School_frequency.keys(),
School_frequency.values(), color='green')
plt.title(f"{Selected_Group}'s School Distribution in
{team}")
plt.xlabel('School')
plt.ylabel('Number of Students')
```

Note that ChatGPT gave me:

`plt.figure(figsize=(12, 6))`

However I decided to change “`figsize=`” according to my own preference.

I also decided to not use:
“`plt.xticks(rotation=45,ha='right')`” at all, as I didn’t need the code ChatGPT suggested.

Declaration - Vijaya Madhulika:	
Name of AI tool	ChatGPT

Input prompt

```
# Step 5: Plot Gender Distribution as a % Pie Chart

def gender_distribution(group_data):
  gender_counts = {'Male': 0, 'Female': 0}
  for row in group_data:
    gender_counts[row['Gender']] += 1
  plt.figure(figsize=(4, 4))
  plt.pie(
     gender_counts.values(),
     labels=['Male', 'Female'],
     colors=['Lightblue', 'Pink']
  )
   plt.title(f"Gender Distribution")
   plt.show()
```

Above is my code to generate a pie chart for the distribution of male and female population in each tutorial group. How do I use matplotlib plt.pie() to display the values in percentages?

Date generated

06/11/2024

Output generated

To display the values as percentages on the pie chart in `matplotlib.pyplot.pie()`, you need to use the `autopct` parameter. This parameter formats the text labels that display the percentage of each slice directly on the chart.

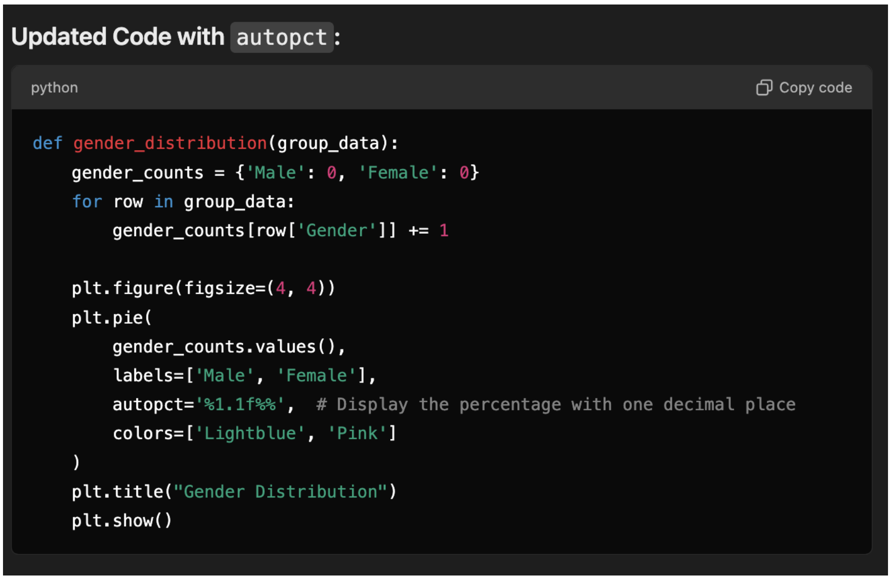
Explanation:

- **`autopct='%1.1f%%'`**:
    - `%1.1f`: Formats the number as a floating-point with one decimal place (e.g., `50.0`).
    - `%%`: Displays the `%` symbol in the label.

How It Works:

- The `autopct` parameter automatically calculates and displays the percentage of each slice of the pie based on the values provided. In your case, it calculates the percentage of male and female students in the `gender_counts` dictionary and displays these percentages on the chart.

Output screenshot



Updated Code with `autopct`:

```python
def gender_distribution(group_data):
    gender_counts = {'Male': 0, 'Female': 0}
    for row in group_data:
        gender_counts[row['Gender']] += 1

    plt.figure(figsize=(4, 4))
    plt.pie(
        gender_counts.values(),
        labels=['Male', 'Female'],
        autopct='%1.1f%%',  # Display the percentage with one decimal place
        colors=['Lightblue', 'Pink']
    )
    plt.title("Gender Distribution")
    plt.show()
```

Impact on submission

My code changed from

```
def gender_distribution(group_data):
  gender_counts = {'Male': 0, 'Female': 0}
  for row in group_data:
    gender_counts[row['Gender']] += 1
  plt.figure(figsize=(4, 4))
  plt.pie(
     gender_counts.values(),
     labels=['Male', 'Female'],
     colors=['Lightblue', 'Pink']
  )
   plt.title(f"Gender Distribution")
   plt.show()
```

To

```
def gender_distribution(group_data):
    gender_counts = {'Male': 0, 'Female': 0}
    for row in group_data:
        gender_counts[row['Gender']] += 1

    plt.figure(figsize=(4, 4))
    plt.pie(
        gender_counts.values(),
        labels=['Male', 'Female'],
        autopct='%1.1f%%',  # Display the percentage with one
decimal place
        colors=['Lightblue', 'Pink']
    )
    plt.title("Gender Distribution")
```

	plt.show()