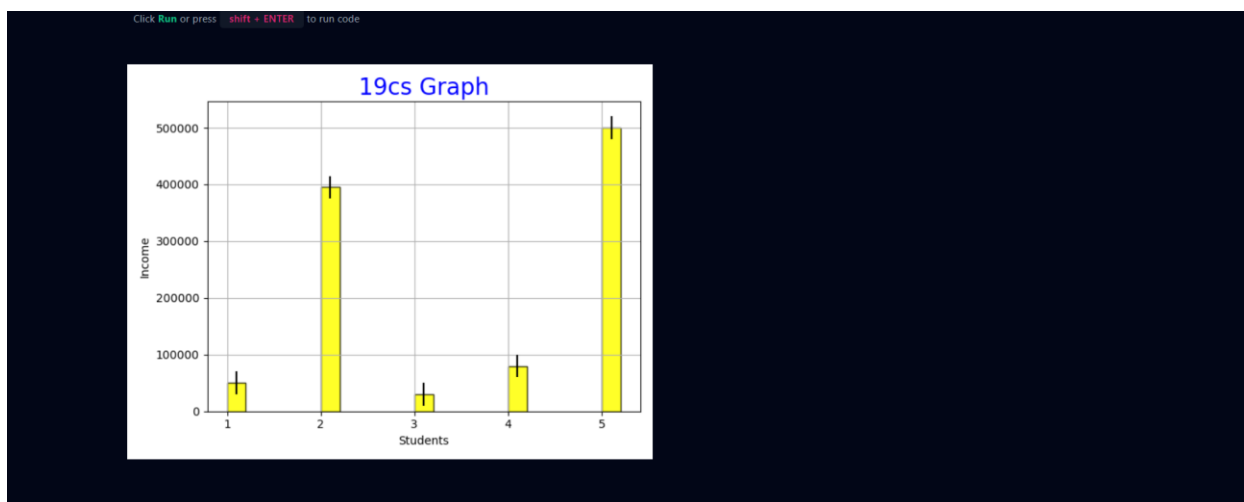
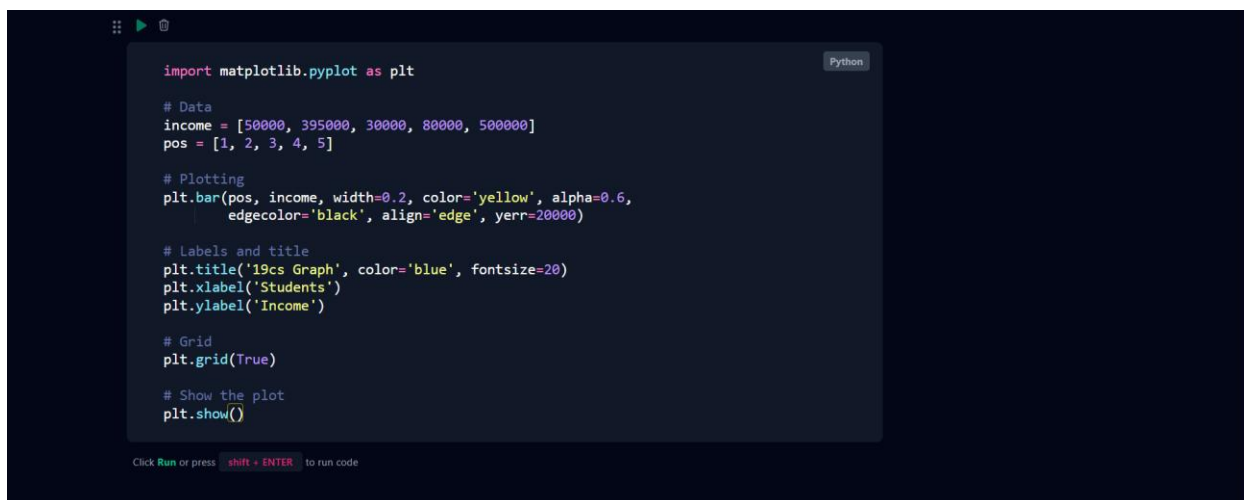
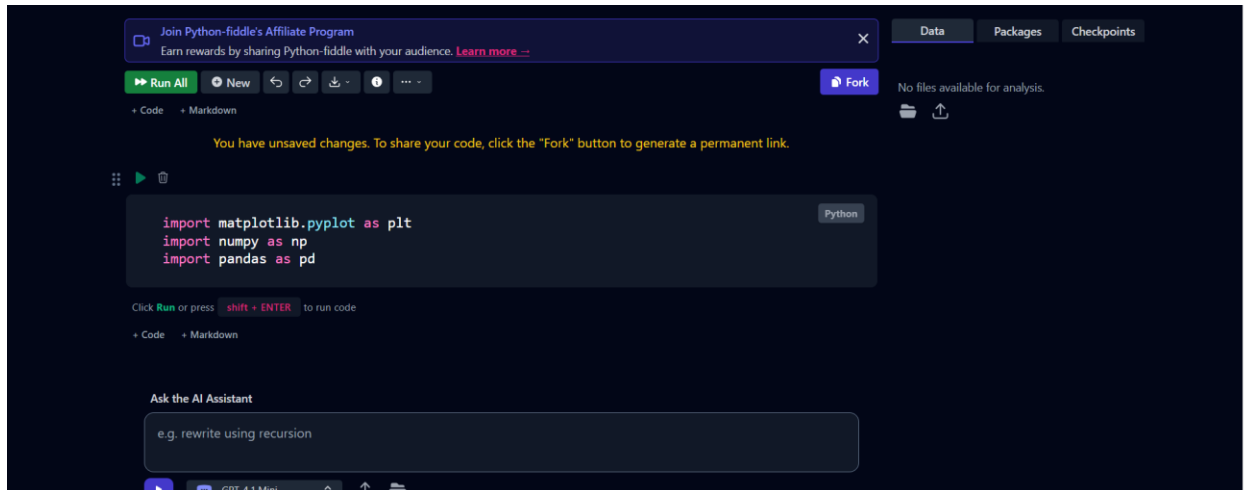


SKILLCRAFT TECHNOLOGY INTERNSHIP TASK-1

Create a bar chart or histogram to visualize the distribution of a categorical or continuous variable such as the distribution of ages or genders in a population



Command Prompt

```
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\dhill>pip install openpyxl
Collecting openpyxl
  Downloading openpyxl-3.1.5-py2.py3-none-any.whl.metadata (2.5 kB)
Collecting et_xmlfile (from openpyxl)
  Downloading et_xmlfile-2.0.0-py3-none-any.whl.metadata (2.7 kB)
Downloading openpyxl-3.1.5-py2.py3-none-any.whl (250 kB)
Downloading et_xmlfile-2.0.0-py3-none-any.whl (18 kB)
Installing collected packages: et_xmlfile, openpyxl
Successfully installed et_xmlfile-2.0.0 openpyxl-3.1.5

C:\Users\dhill>
```

```
import pandas as pd

# Load Excel file (Make sure the path is correct for your environment)
df = pd.read_excel('16csdata.xlsx') # Change path as needed

# Display the first two rows
df.head(2)
```

	rno	name	age	cgpa	address	income	gender	scholarship amount
0	16cs01	Hooria Tahir	22	2.95	bahawalpur	22000	1	80000
1	16CS02	Sammrah Hanif	22	3.64	bahawalpur	0	1	13000

```
import pandas as pd

# Assuming df is already loaded and contains a 'gender' column
labels = ['male', 'female']

# Get value counts for 'gender' column
classes = df['gender'].value_counts() # default: sort=True, ascending=False

# Display the result
print(classes)
print(type(classes))
```

python

male 3

female 1

Name: gender, dtype: int64

<class 'pandas.core.series.Series'>

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

# Load Excel file (ensure the file is in the same directory)
df = pd.read_excel('16csdata.xlsx') # You can update the path as needed

# Define gender labels
labels = ['male', 'female']

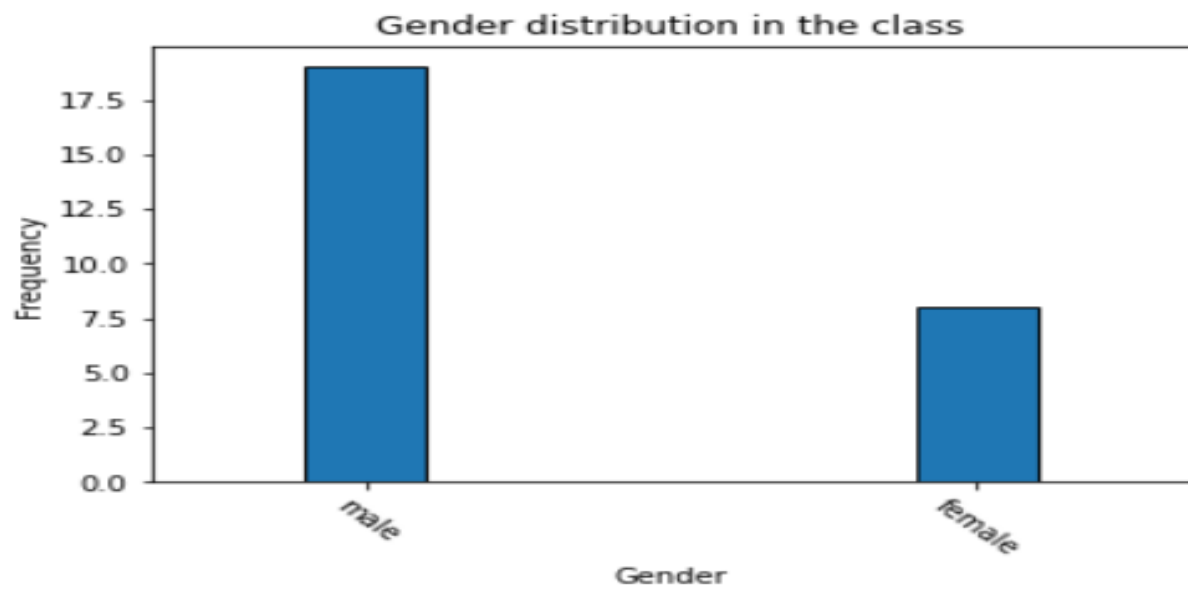
# Count gender frequencies
classes = df['gender'].value_counts()

# Plot
classes.plot(kind='bar', rot=315, width=0.2, edgecolor='black')

# Title and labels
plt.title("Gender distribution in the class")
plt.xticks(ticks=range(len(labels)), labels=labels)
plt.xlabel("Gender")
plt.ylabel("Frequency")

# Show the plot
plt.tight_layout()
plt.show()
```

```
Text(0, 0.5, 'Frequency')
```



```
import pandas as pd

# Load data (adjust the file name/path if needed)
df = pd.read_excel('16csdata.xlsx') # Make sure the file exists and has data

# Print the shape
print(df.shape)
```

```

import pandas as pd
import matplotlib.pyplot as plt

# Load your data
df = pd.read_excel('16csdata.xlsx') # Make sure this file exists

# Count non-null values per column
counts = df.count()

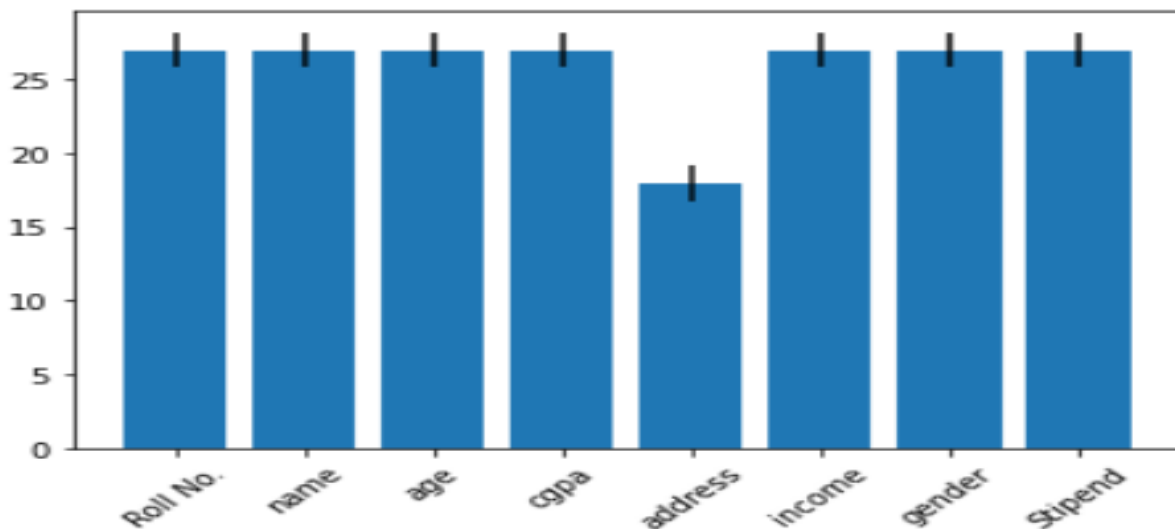
# Plot the bar chart
plt.bar(range(df.shape[1]), counts, yerr=1.2, edgecolor='black')

# Automatically get column names for x-axis
plt.xticks(range(df.shape[1]), df.columns, rotation=45)

# Add labels
plt.title("Non-null Values per Column")
plt.xlabel("Columns")
plt.ylabel("Count")

# Display the plot
plt.tight_layout()
plt.show()

```



```

import pandas as pd
import matplotlib.pyplot as plt

# Load data
df = pd.read_excel('16csdata.xlsx') # Ensure this file exists and is readable

# Bar positions
x_pos = [1, 3, 5, 7, 9, 11, 13, 15]
rwidth = 0.6

# Data to plot: non-null count per column
counts = df.count()

# Ensure that the number of bars matches the number of positions
if len(counts) != len(x_pos):
    raise ValueError("Number of x positions and number of DataFrame columns must match.")

# Plot the bar chart
plt.bar(x_pos, counts, width=rwidth, align='center', color='cyan',
        alpha=0.6, edgecolor='black', yerr=1.2, label='Non-null Counts')

# Set x-tick labels
plt.xticks(x_pos, ['Roll No.', 'name', 'age', 'cgpa', 'address',
                    'income', 'gender', 'Stipend'], rotation=45)

# Add legend
plt.legend()

```

```

# Ensure that the number of bars matches the number of positions
if len(counts) != len(x_pos):
    raise ValueError("Number of x positions and number of DataFrame columns must match.")

# Plot the bar chart
plt.bar(x_pos, counts, width=rwidth, align='center', color='cyan',
        alpha=0.6, edgecolor='black', yerr=1.2, label='Non-null Counts')

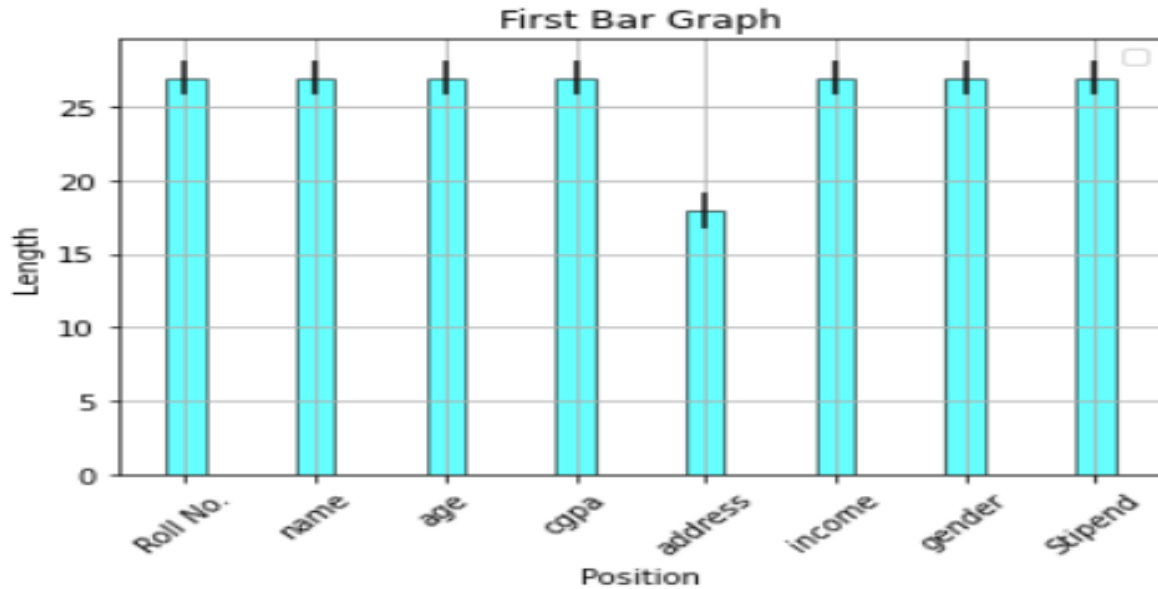
# Set x-tick labels
plt.xticks(x_pos, ['Roll No.', 'name', 'age', 'cgpa', 'address',
                    'income', 'gender', 'Stipend'], rotation=45)

# Add legend
plt.legend()

# Set labels and title
plt.xlabel('Position')
plt.ylabel('Length')
plt.title('First Bar Graph')

# Grid and layout
plt.grid(True)
plt.tight_layout()
plt.show()

```



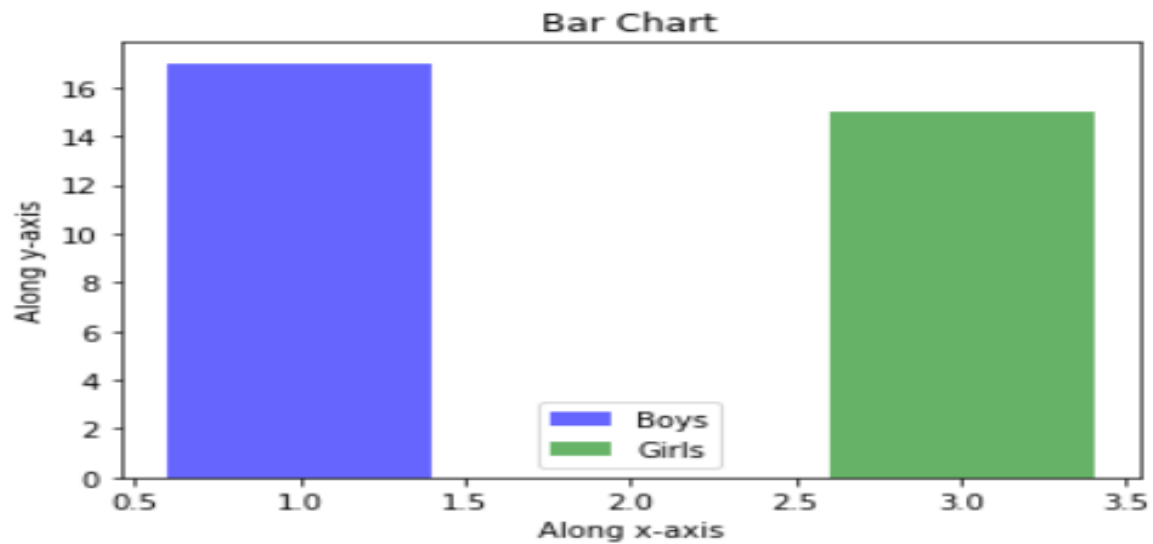
```
import matplotlib.pyplot as plt

# Data
x = [1]; boys = [17]
x2 = [3]; girls = [15]

# Plotting bars
plt.bar(x, boys, label="Boys", color='blue', alpha=0.6)
plt.bar(x2, girls, label="Girls", color='green', alpha=0.6)

# Labels and title
plt.xlabel('Along x-axis')
plt.ylabel('Along y-axis')
plt.title('Bar Chart')

# Show legend and plot
plt.legend()
plt.show()
```



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

# Load data (make sure the file 'iris.txt' exists in the same directory)
df = pd.read_csv("iris.txt")

# Width of bars
rwidth = 0.2

# Count of non-null values per column
counts = df.count()

# Check if color list matches the number of columns
colors = ['blue', 'red', 'green', 'white', 'yellow']
if len(colors) < len(counts):
    raise ValueError("Not enough colors provided for the number of columns.")

# Plot bar chart
plt.bar(range(len(counts)), counts, width=rwidth, bottom=0,
        color=colors[:len(counts)], alpha=0.6, edgecolor='black')

# Set x-tick labels to column names
plt.xticks(range(len(counts)), df.columns, rotation=45)
```



```

# Width of bars
rwidth = 0.2

# Count of non-null values per column
counts = df.count()

# Check if color list matches the number of columns
colors = ['blue', 'red', 'green', 'white', 'yellow']
if len(colors) < len(counts):
    raise ValueError("Not enough colors provided for the number of columns.")

# Plot bar chart
plt.bar(range(len(counts)), counts, width=rwidth, bottom=0,
        color=colors[:len(counts)], alpha=0.6, edgecolor='black')

# Set x-tick labels to column names
plt.xticks(range(len(counts)), df.columns, rotation=45)

# Labels and title
plt.xlabel('Columns')
plt.ylabel('Non-null Count')
plt.title('Non-null Data Counts in iris.txt')

# Layout adjustment
plt.tight_layout()
plt.show()

```

```

import matplotlib.pyplot as plt
population_ages = [10,20,87, 65, 24, 37, 21, 9,7,4, 100, \
                  115, 53, 70, 75, 43, 54,\
                  33, 39, 51, 59, 30]
plt.hist(population_ages, rwidth=0.2)

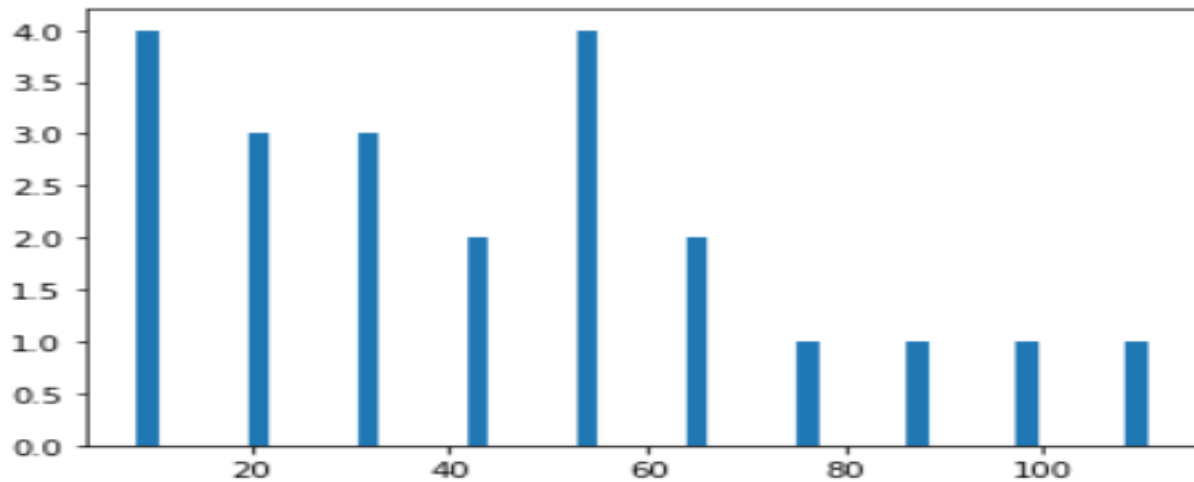
```

Click **Run** or press **shift + ENTER** to run code

```

array([4., 3., 3., 2., 4., 2., 1., 1., 1., 1.])
array([ 4. , 15.1, 26.2, 37.3, 48.4, 59.5, 70.6, 81.7, 92.8,
       103.9, 115. ])
<BarContainer object of 10 artists>

```



```
import matplotlib.pyplot as plt

# Data
population_ages = [10, 20, 87, 65, 24, 37, 21, 9, 7, 4, 100,
                   115, 53, 70, 75, 43, 54,
                   33, 39, 51, 59, 30]

# Define bins
bins = [0, 10, 20, 30, 40, 50, 70, 80, 90, 100, 110, 120]

# Plot histogram with label
plt.hist(population_ages, bins=bins, histtype='bar', color='g',
         alpha=0.75, label="Age Group Count")

# Add ticks, labels, title, legend
plt.xticks([0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120], rotation=45)
plt.xlabel('Along x-axis')
plt.ylabel('Along y-axis')
plt.title('Population Age Distribution')
plt.legend(title="Histogram")

# Show plot
plt.tight_layout()
plt.show()
```

Population Age Distribution

