Project no : 4

HIRING PROCESS ANALYTICS

Project Description:

The Data Analysis for Hiring Trends program is designed to analyse a dataset provided by a multinational company (MNC) to gain insights into their hiring process. My task is to analyse the dataset and provide a detailed report that answers key questions and draws meaningful conclusions.

The program follows a series of steps for exploratory data analysis (EDA) and utilizes statistical formulas and Pandas to derive necessary conclusions. Here is an overview of the program's steps:

Tech-Stack Used: Excel, Python, Pandas

Approach :

1. **Understanding data columns and data:**

   The program begins by exploring the dataset's columns and understanding the available data, such as gender, salary, department, and other relevant fields.

2. **Checking for missing data:**

   The program identifies any missing data points in the dataset and handles them appropriately, ensuring data integrity for accurate analysis.

3. **Clubbing columns with multiple categories:**

   If there are columns with multiple categories, such as different post tiers, the program combines them to create a consolidated representation for analysis.

4. **Checking for outliers:**

   The program identifies potential outliers in the dataset, which could skew analysis results, and flags them for further investigation.

5. **Removing outliers:**

   After identifying outliers, the program applies outlier removal techniques to eliminate or correct extreme values, ensuring more reliable analysis outcomes.

6. **Drawing Data Summary:**

   The program generates a summary of the dataset, including key statistics and metrics, to provide a comprehensive overview of the hiring trends.

7. **Answering specific questions:**

   The program addresses specific questions provided by the company, such as determining the count of males and females hired and calculating the average salary offered.

8. **Drawing class intervals for salary:**

   The program defines class intervals for salary by considering the range of salaries and selecting an appropriate interval width. It creates intervals and presents them in a table or histogram.

-Harsh Tikone

9. Visualizing data using charts and plots:

The program incorporates charts and plots to visualize data, such as drawing a pie chart or bar graph to depict the proportion of people working in different departments.

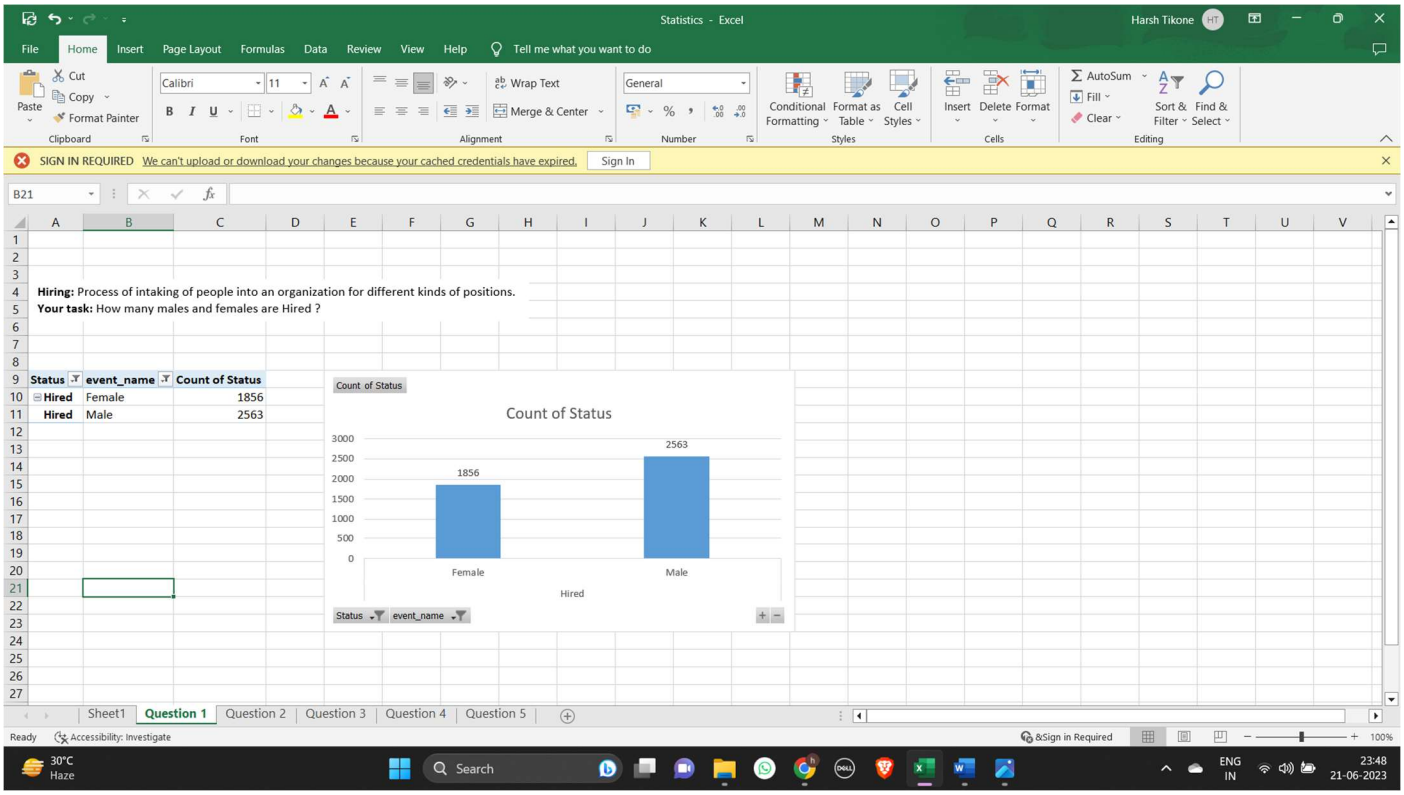
10. Representing different post tiers using charts/graphs:

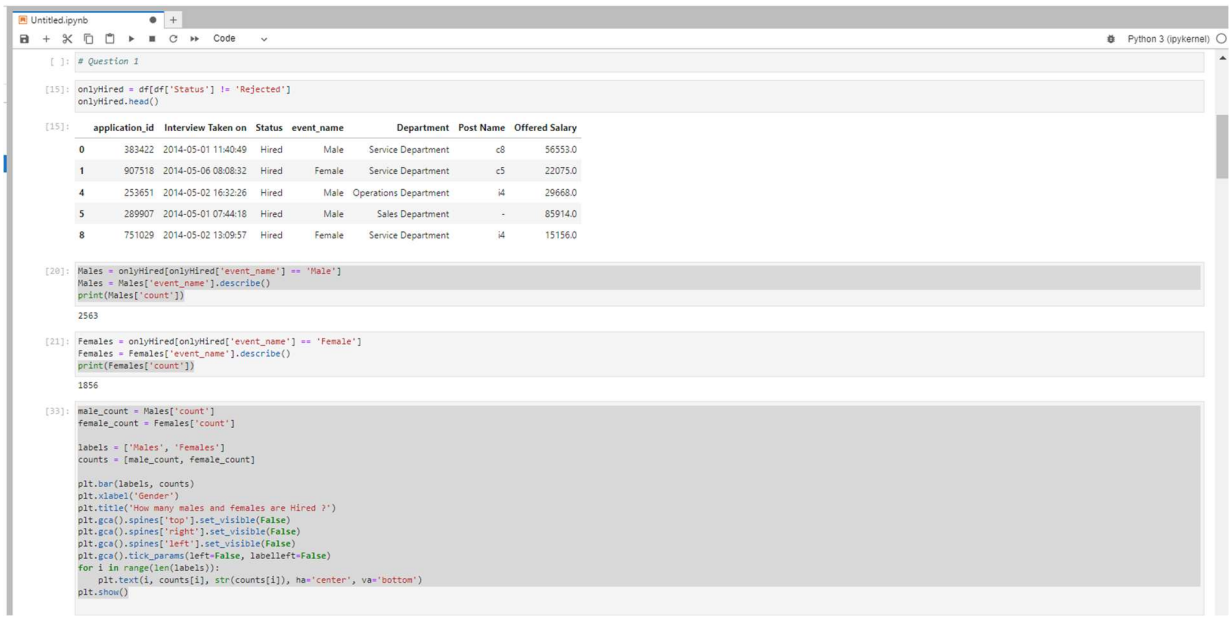
The program creates visual representations, such as bar graphs or stacked bar graphs, to showcase the distribution of employees across different post tiers.

Result and Insights :

1) Hiring: Process of intaking of people into an organization for different kinds of positions.

Your task: How many males and females are Hired?





-Harsh Tikone

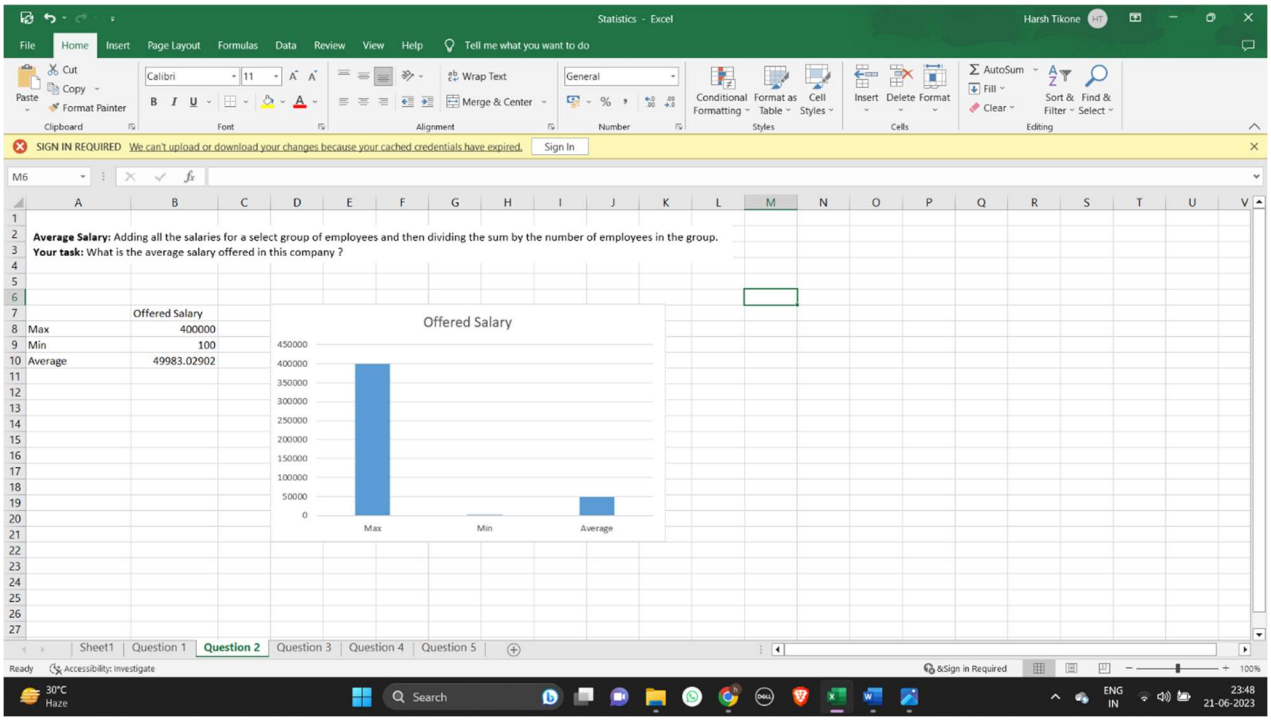
2) Average Salary: Adding all the salaries for a select group of employees and then dividing the sum by the number of employees in the group.

Your task: What is the average salary offered in this company? x

Salary

```
[ ]: # Question 3

[59]: upper_limit = df['Offered Salary'].max()
      lower_limit = df['Offered Salary'].min()
      print(upper_limit, lower_limit)

      400000.0 100.0

[84]: salary_values = [value for value in salary_values if not math.isnan(value)]

      range_sal = upper_limit - lower_limit
      col = 5
      in_wid = range_sal/col

      class_intervals = [( lower_limit + i * in_wid, lower_limit + (i + 1) * in_wid)
                         for i in range(col)]

      frequency = [0] * col
      for value in salary_values:
              for i, interval in enumerate(class_intervals):
                  if interval[0] <= value <= interval[1]:
                      frequency[i] += 1
                      break
      x_labels = [f"{interval[0]} - {interval[1]}" for interval in class_intervals]

      plt.bar(x_labels, frequency)

      plt.xlabel('Salary Intervals')
      plt.ylabel('Frequency')
      plt.title('Salary Distribution by Intervals')

      for i, freq in enumerate(frequency):
              plt.text(i, freq, str(freq), ha='center', va='bottom')

      plt.xticks(rotation=20)

      plt.show()
```

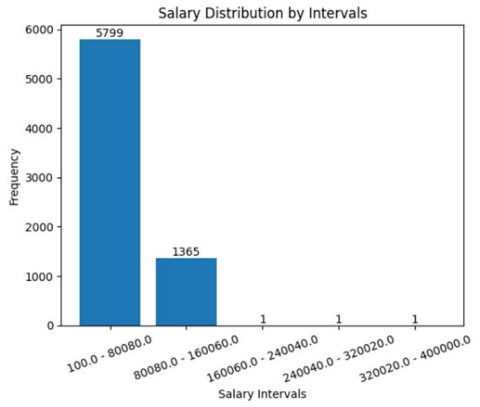
```
[42]: Average_sal = df['Offered Salary'].sum()/df['Offered Salary'].count()
      print("What is the average salary offered in this company ?")
      print(f"The Average Salary offered in this company is {Average_sal.round()}")

      What is the average salary offered in this company ?
      The Average Salary offered in this company is 49983.0

[44]: labels = ['Minimum', 'Average','Maximum' ]
      counts = [ df['Offered Salary'].min(),Average_sal.round(),df['Offered Salary'].max()]

      plt.bar(labels, counts)
      plt.xlabel('Salary')
      plt.title('What is the average salary offered in this company ?')
      plt.gca().spines['top'].set_visible(False)
      plt.gca().spines['right'].set_visible(False)
      plt.gca().spines['left'].set_visible(False)
      plt.gca().tick_params(left=False, labelleft=False)
      for i in range(len(labels)):
          plt.text(i, counts[i], str(counts[i]), ha='center', va='bottom')
      plt.show()
```

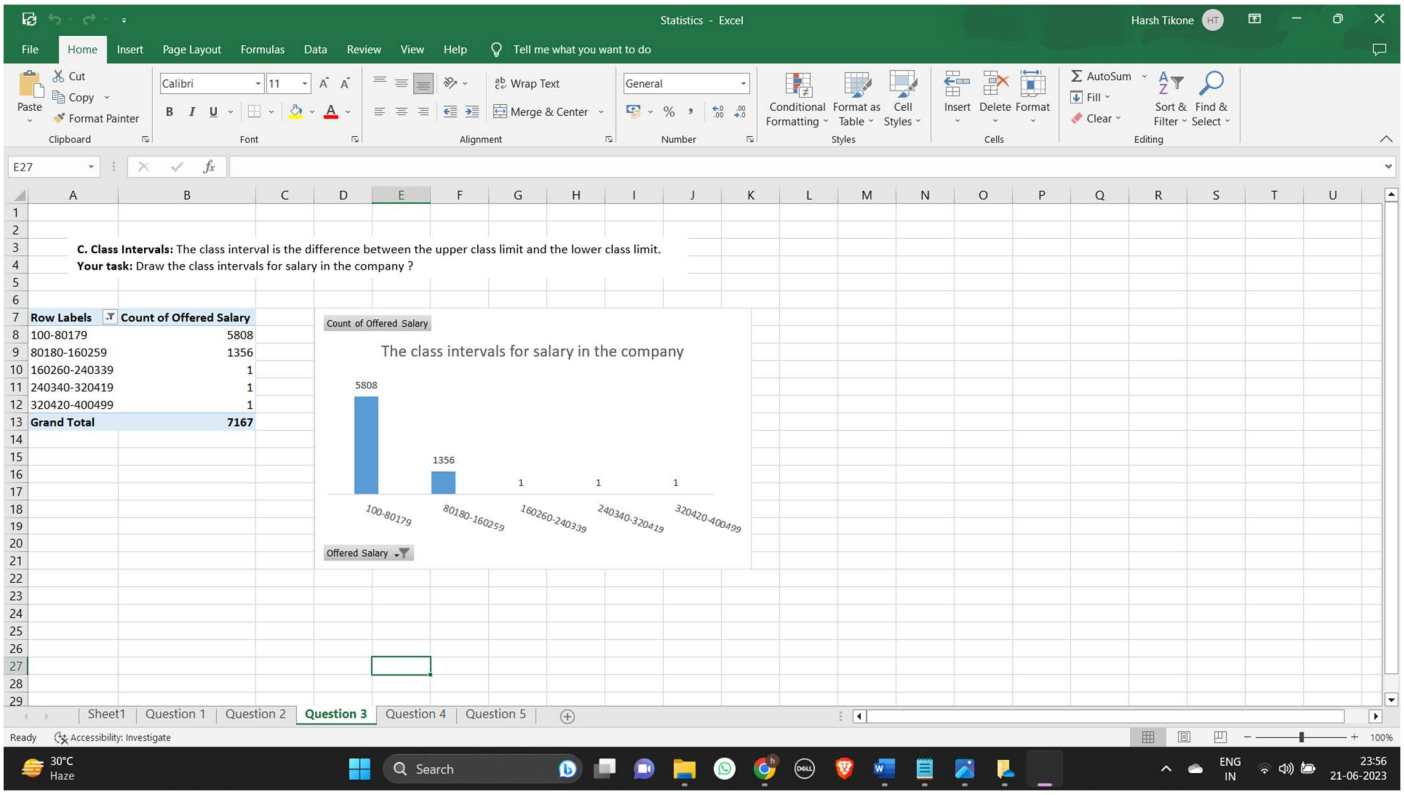


-Harsh Tikone

3) Class Intervals: The class interval is the difference between the upper class limit and the lower class limit.

Your task: Draw the class intervals for salary in the company ?

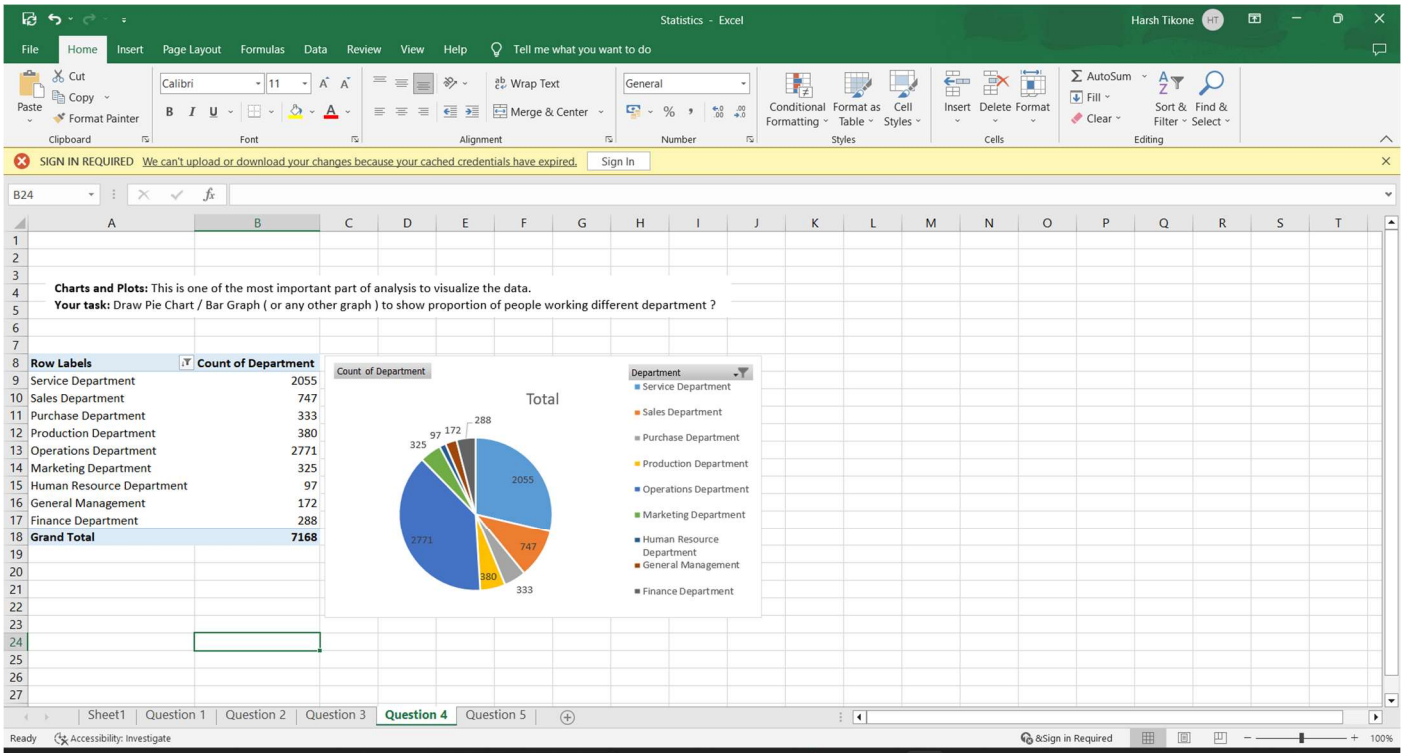


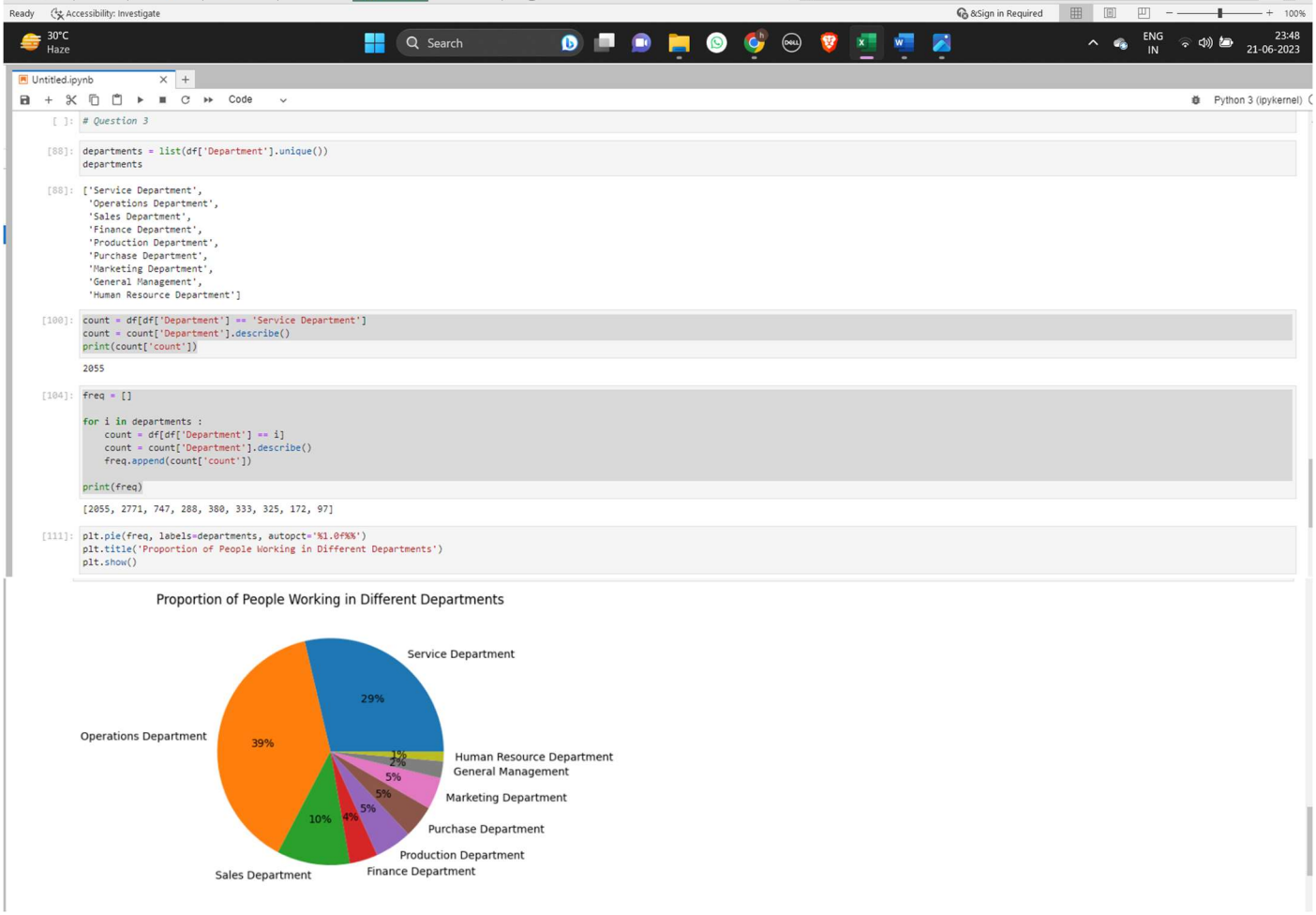


-Harsh Tikone

4) Charts and Plots: This is one of the most important part of analysis to visualize the data.

Your task: Draw Pie Chart / Bar Graph (or any other graph) to show proportion of people working different department ?

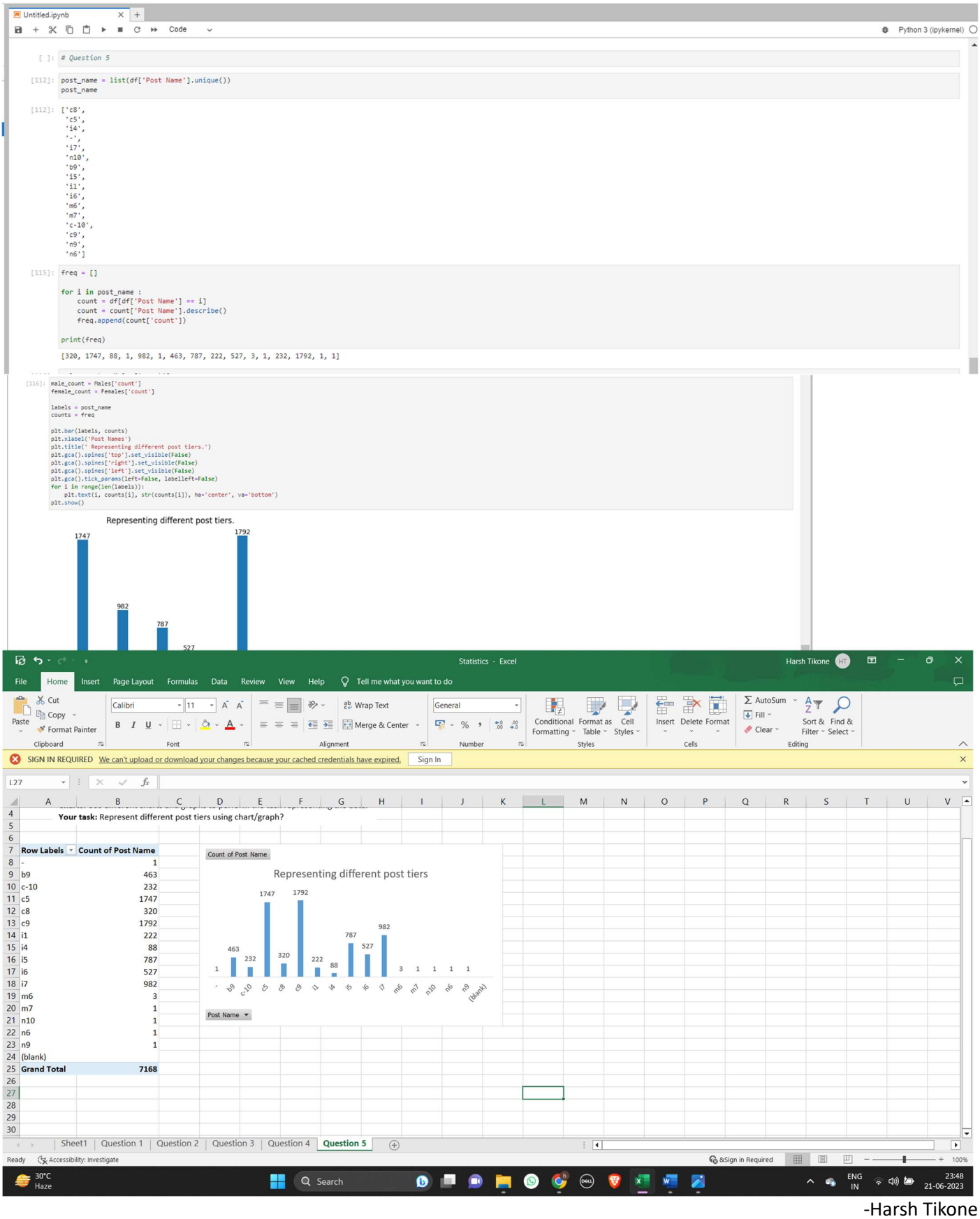




-Harsh Tikone

5) Charts: Use different charts and graphs to perform the task representing the data.

Your task: Represent different post tiers using chart/graph?



-Harsh Tikone