

In this report, we conducted an inference attack on a dataset containing income information for 100 employees of a company. The primary objective was to estimate an employee's (Adam) income based on the given data in two different tasks.

Python Code:

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

# Load the data
data = pd.read_csv('dataset.csv', delimiter=",", header=None);

# Round to the nearest 1000
data = round(data, -3)

# Calculate the frequencies
df = data.value_counts(sort=False)

print(df.to_string())

# Frequency plot
fplot = plt.figure(1)
df.plot(kind='bar', x='Income', y='Count', color='lightblue')
plt.title('Frequency plot')
plt.xlabel('Income')
plt.ylabel('Count')
fplot.show()

# Calculate PMF
total_data = len(data)
```

CYBER 759 Lab – Inference Attack

Gunnar Yonker

```
pmf = df / total_data
```

```
# probability for pmf
```

```
pmfplot = plt.figure(2)
```

```
# PMF plot
```

```
pmfplot = plt.figure(2)
```

```
pmf.plot(kind='bar', color='lightblue')
```

```
plt.title('Probability Mass Function')
```

```
plt.xlabel('Income')
```

```
plt.ylabel('Probability')
```

```
pmfplot.show()
```

```
# Estimate for Adam's Salary based on PMF
```

```
adam_estimate = pmf.idxmax()[0]
```

```
print(f"Estimated Salary for Adam based on PMF: ${adam_estimate}")
```

```
print(f"Probability that the estimate is correct: {pmf.max()}")
```

```
# Task 2
```

```
# Filter data greater than $75,000
```

```
filtered_data_series = data[0][data[0] > 75000]
```

```
# Calculate frequencies for filtered data
```

```
df_filtered = filtered_data_series.value_counts(sort=False)
```

```
# Calculate PMF for filtered data
```

```
total_filtered_data = len(filtered_data_series)
```

```
pmf_filtered = df_filtered / total_filtered_data
```

CYBER 759 Lab – Inference Attack
Gunnar Yonker

```
# PMF plot for filtered data

pmfplot_filtered = plt.figure(3)

pmf_filtered.sort_index().plot(kind='bar', color='lightgreen')

plt.title('Filtered Probability Mass Function')

plt.xlabel('Income')

plt.ylabel('Probability')

pmfplot_filtered.show()

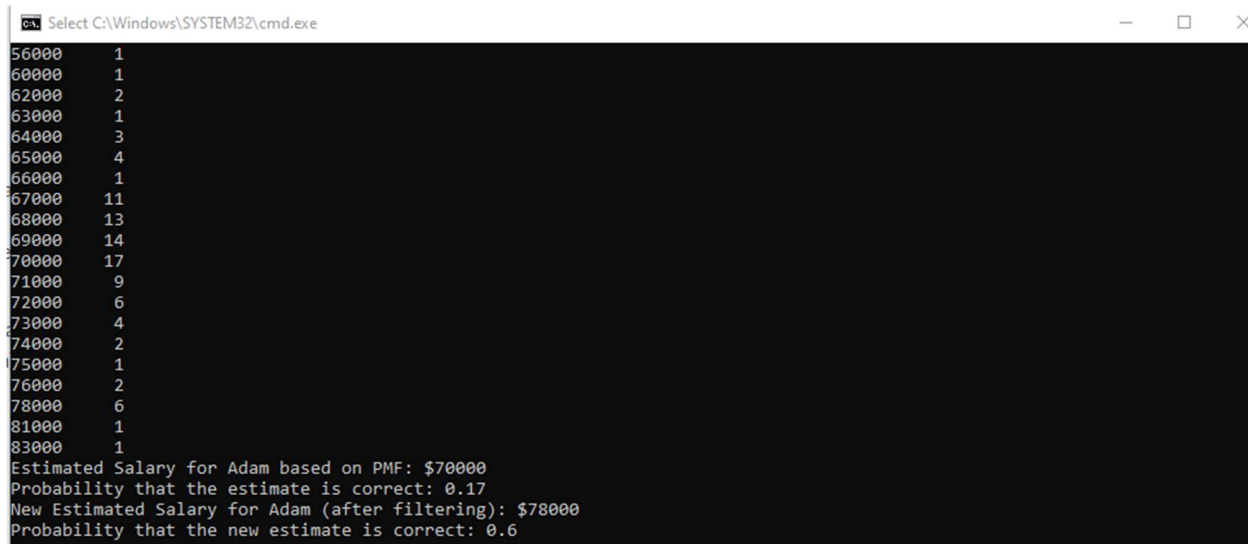

# New estimate for Adam's salary based on filtered PMF

adam_estimate_filtered = pmf_filtered.idxmax()

print(f"New Estimated Salary for Adam (after filtering): ${adam_estimate_filtered}")

print(f"Probability that the new estimate is correct: {pmf_filtered.max()}")


input()
```

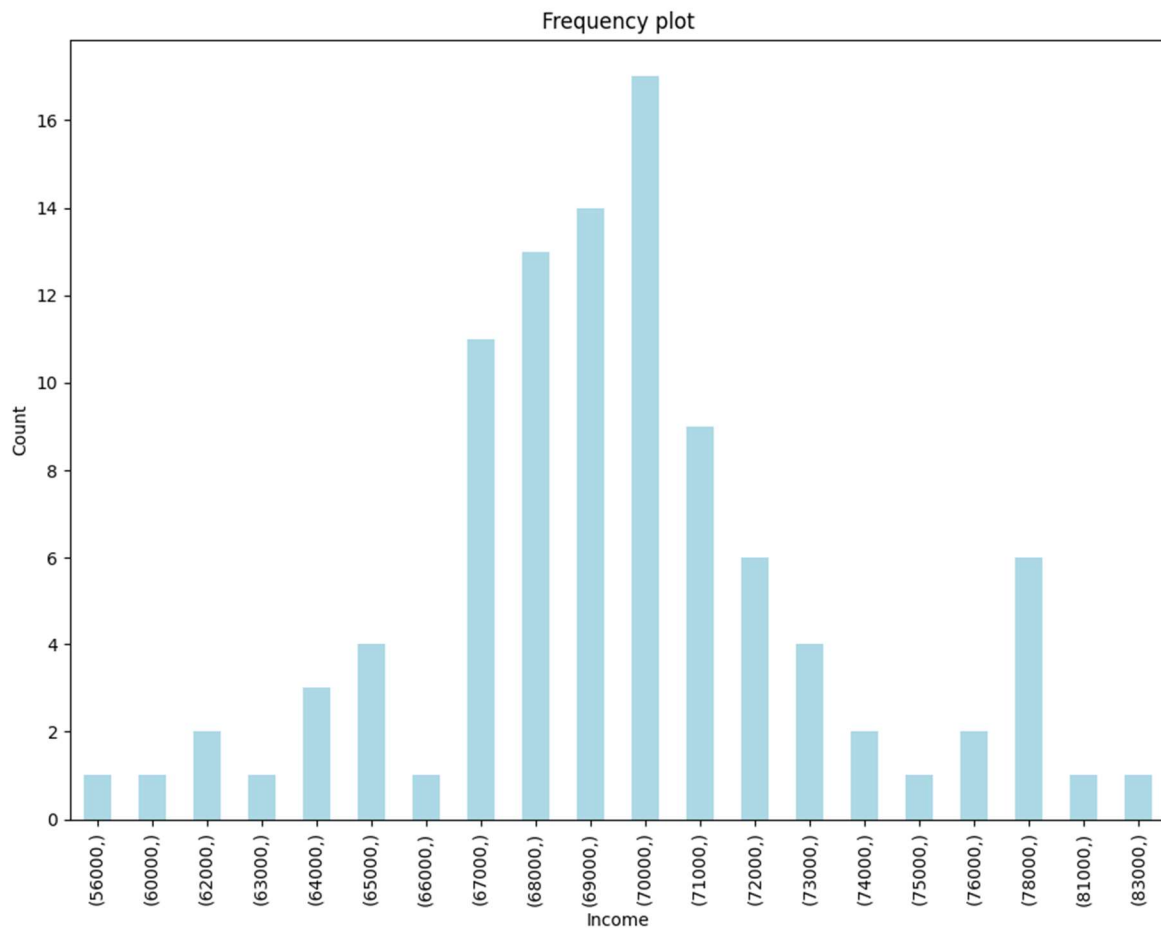


```
Select C:\Windows\SYSTEM32\cmd.exe
56000 1
60000 1
62000 2
63000 1
64000 3
65000 4
66000 1
67000 11
68000 13
69000 14
70000 17
71000 9
72000 6
73000 4
74000 2
75000 1
76000 2
78000 6
81000 1
83000 1
Estimated Salary for Adam based on PMF: $70000
Probability that the estimate is correct: 0.17
New Estimated Salary for Adam (after filtering): $78000
Probability that the new estimate is correct: 0.6
```

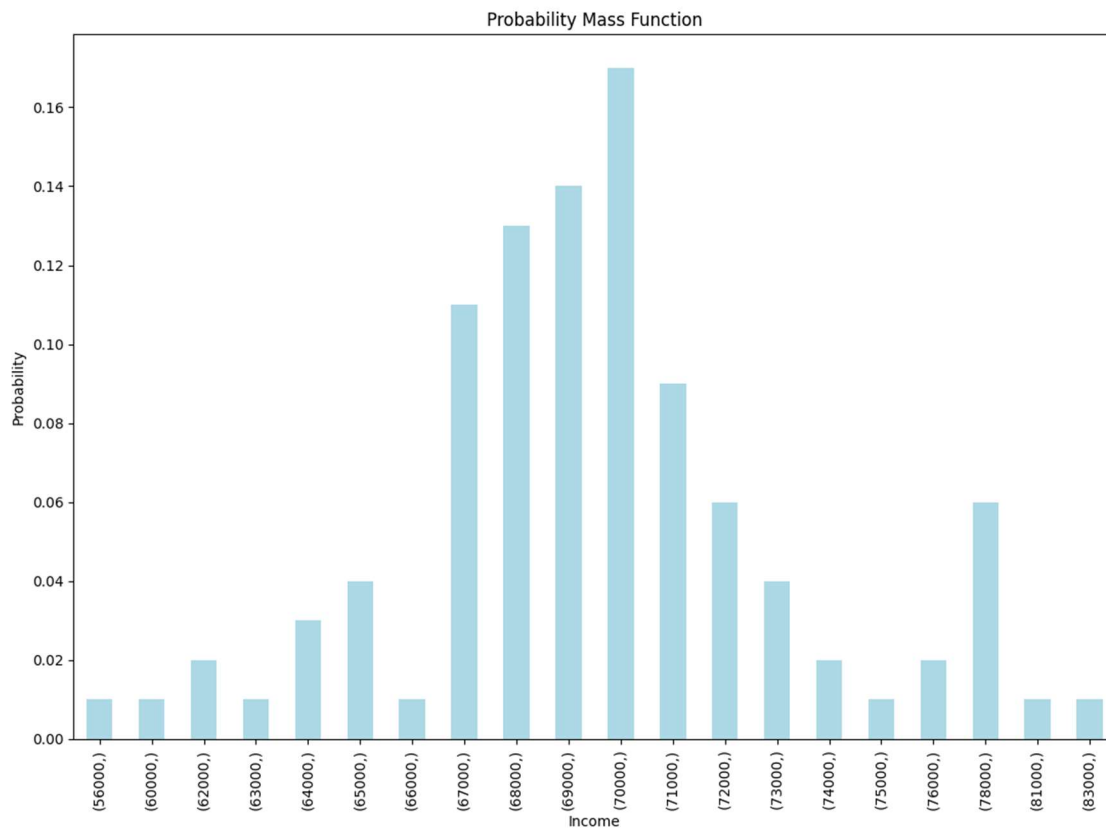
Analysis:

Task 1:

A frequency plot was generated using the matplotlib library to visualize the number of occurrences of each income value rounded to the nearest 1,000 in the dataset. The dataset was rounded to the nearest 1,000 for ease of analysis.



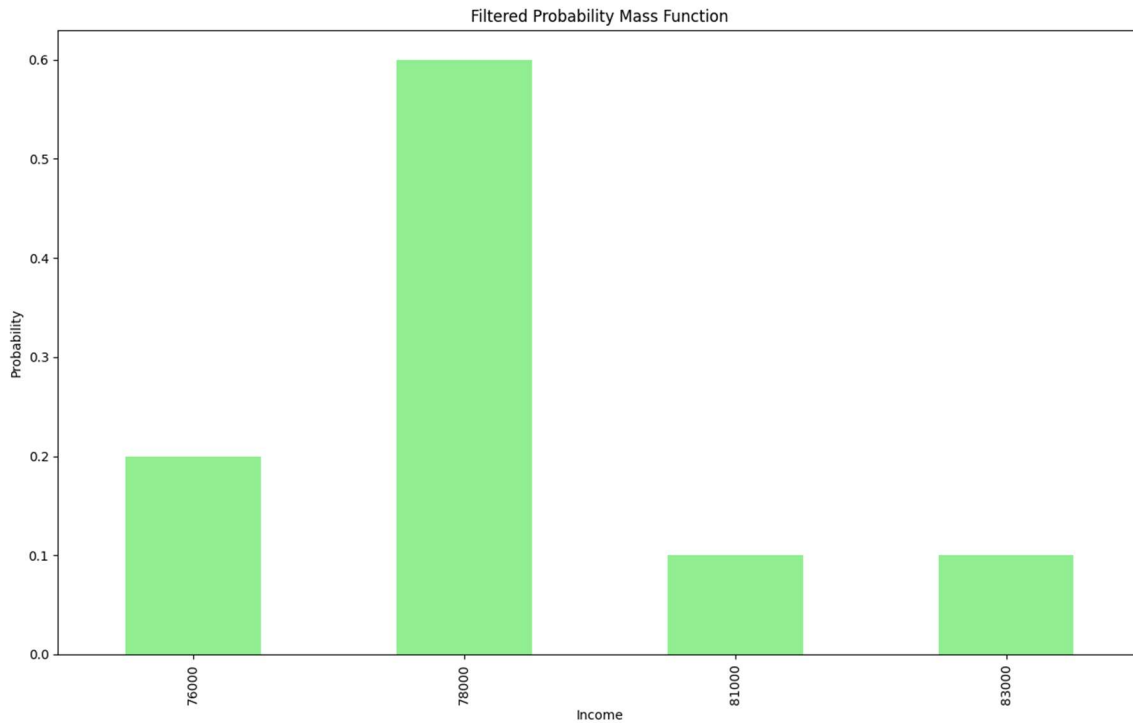
The probability mass function (PMF) was calculated to represent the probability of each income value rounded to the nearest 1,000.



Based on the probability distribution, Adam's salary can be estimated to be approximately \$70,000. The probability that this estimate is correct is 0.17 or 17%. This will most likely not be his exact salary since the data was rounded to the nearest 1,000 but with 17% certainty, we can say his salary is approximately \$70,000.

Task 2:

Now that we know that Adam has inadvertently revealed that he makes over \$75,000 a year, we can further estimate his salary. The dataset can now become smaller by looking at only the values that are > \$75,000 and a new probability distribution can be created. This graph and estimate are again made with the rounding set to the nearest 1,000.



Based on this filtered probability distribution, Adam's salary can be estimated to be approximately \$78,000. The probability that this estimate is correct is 0.6 or 60%. Adam inadvertently revealing that he makes above \$75,000, it has helped to further refine and make this estimate's probability percent much higher than the original estimation.

Conclusion:

The frequency plot provided a clear representation of how often each income level appeared in our dataset. For example, the top 3 frequent salary values were \$68000, \$69000, \$70000. The probability distribution mirrored the frequency plot in shape, showing the likelihood of each salary value. From this we made an initial estimate that Adam's salary was \$70,000, with a probability of 17%. After incorporating the additional information that Adam earns more than \$75,000, we refined our estimate based on the filtered dataset. This resulted in a new estimated salary of \$78,000 with a probability of 60%.

The inference attack provided valuable insights into Adam's probable income bracket. Using a combination of frequency and probability analyses, we were able to make informed estimates about Adam's salary, further refined by auxiliary information. Some of my key takeaways include:

Visualization tools, like frequency plots and PMFs, are instrumental in making sense of data distributions. Auxiliary or additional information can significantly narrow down estimations, highlighting the risks of seemingly harmless disclosures on platforms like social media. I think that many people don't quite understand the magnitude of the information that they share daily on social media whether it is sensitive personal information they are sharing or information that can be used to gain access through attacks such as the inference attack to their sensitive personal information. One last takeaway that I had from this lab is that using Python to read and analyze the dataset is more straight forward than I was anticipating, I can really see how powerful it would be especially when analyzing a larger dataset. This just furthers the fact that it doesn't take a super genius to carry out certain attacks on sensitive personal information, especially when you consider all the information/tutorials that are available to follow. The matplotlib for plotting is also a lot easier to use than having to construct a graph yourself, it seems like it is an incredibly useful library to take advantage of.