MATHEMATICS & STATISTICS REPORT

Karolis Liubavicius (1671752)

k.liubavicius@student.han.nl

Table of Contents

Introduction

Alienation among adolescents, characterized by a sense of disconnection and isolation from family, friends, or work, has been a matter of concern, particularly during the COVID-19 pandemic. In response to the various pandemic-related restrictions, the U.S. government initiated a comprehensive study to measure alienation, income, and gender differences in alienation. Furthermore, the study aimed to investigate the profiles of individuals seeking psychological consultation due to symptoms of alienation and to understand the relationship between alienation and income, considering the economic impact of the pandemic. In this report, I will present the results of analysis of the data collected from this study. This analysis covers many aspects, including data cleaning, descriptive statistics, distribution analysis, gender differences, income-related insights, and more.

Data Preparation and Cleaning

```
df = pd.read_csv('1671752alienation_data.csv')
print(df.isna().sum())
```

```
alienation    2
income        6
male          0
consult       0
dtype: int64
```

One of the most important steps is to clean the data set from missing values. The problem with missing values is that may affect the accuracy of the following tests. In this specific data set, we have 2 NaN values in the column of alienation and 6 NaN values in the column of income. This means that we have 2 options, either delete or replace these values with mean, median, or mode, therefore, it depends on the values that the data set contains in specific columns. In this case, we take the option to delete missing values.

```
#deleteing NA values
df.dropna(subset=['alienation', 'income'], inplace=True)
print(df.isna().sum())
```

```
alienation    0
income        0
male          0
consult       0
dtype: int64
```

As we can see, after the deletion of missing values our data set is clean and ready to be analyzed. Or at least it is clean from missing values.

Descriptive Statistics of Alienation and Income

In this case, I used for() loop for the same descriptive analysis in alienation and income, so the code doesn’t repeat that often, optimization of the code is very important as well, as there is a saying within developers' environment “never repeat yourself”, so I tried to apply this approach as much as I can in this report.

```
alienation = ['alienation', 'income']

for column in alienation:
    data = df[column]

    # Calculate the mean, median, stdev, range, min, max, and length of the column
    mean = np.mean(data)
    median = np.median(data)
    stdev = np.std(data)
    data_range = np.ptp(data) #peak to peak range
    min_value = min(data)
    max_value = max(data)
    col_length = len(data)

    print(f"Statistics for {column}:")
    print(f"Mean: {mean}")
    print(f"Median: {median}")
    print(f"Standard Deviation: {stdev}")
    print(f"Col Length: {col_length}")
    print(f"Min Value: {min_value}")
    print(f"Max Value: {max_value}")
    print(f"Range: {data_range}\n")
```

Alienation

```
Statistics for alienation:
Mean: 5.5608108108108105
Median: 5.0
Standard Deviation: 2.895071187453542
Col Length: 444
Min Value: 1.0
Max Value: 10.0
Range: 9.0
```

As we can see above, alienation is a grading of the psychological state, which means that there are no outliers, in this case, I can state that the mean is more representative than the median, which represents the middle value of scale, for the further investigation in alienation column, I will be using mean which is 5.56 which is higher than the median, and this indicates that the distribution is right skewed. The standard deviation, with a value of 2.89, indicates the degree of variability within the data. In other words, it quantifies how individual data points tend to deviate from the mean, providing a measure of the data's overall dispersion or spread. The range from the max value to the min value is 9, while 10 is the max and 1 is the min value.

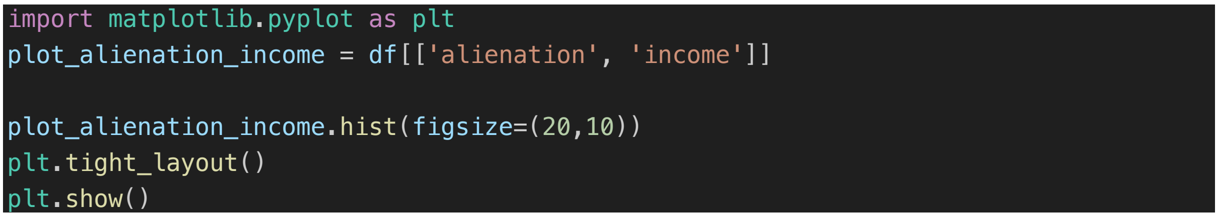
Income

```
Mean: 59401.72
Median: 61466.53515625
Standard Deviation: 36061.47
Col Length: 444
Min Value: 0.0
Max Value: 137731.33
Range: 137731.33
```

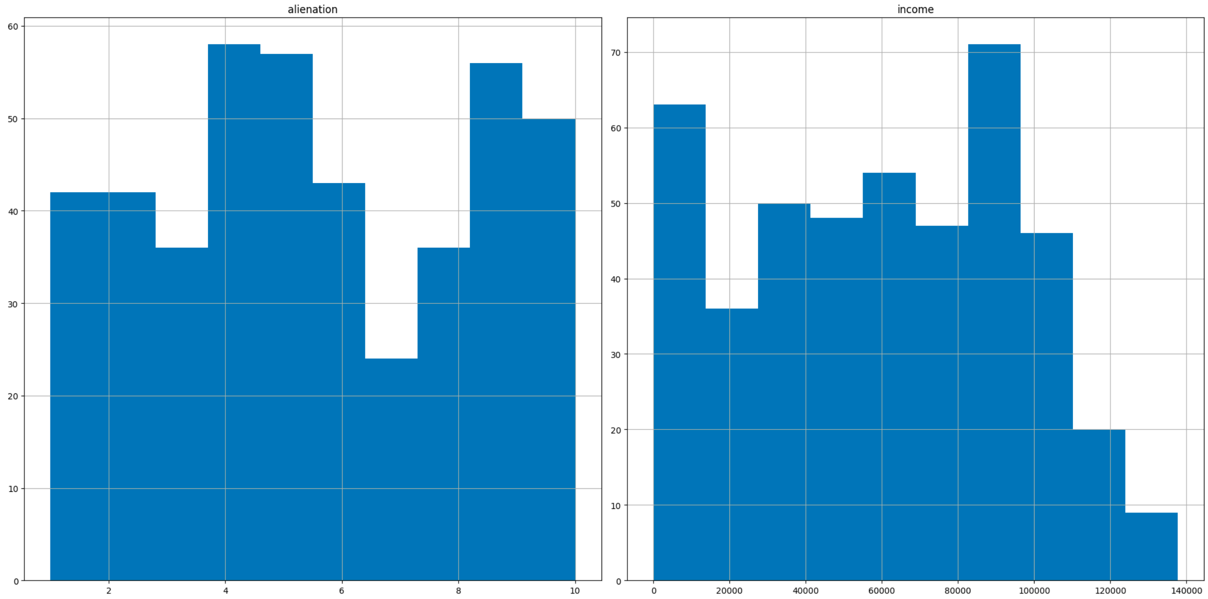
The descriptive statistics for income reveal several key insights about the dataset. The mean income, at $59,401.72, represents the average earnings of the surveyed individuals. Meanwhile, the median income, which stands at $61,466.54, signifies the middle point of the income distribution this indicates that the distribution is left skewed. The standard deviation of approximately $36,061.47 indicates a considerable degree of variability in income among the respondents, showcasing how individual income values tend to deviate from the mean. This measurement quantifies the spread of income data. The dataset contains a total of 444 observations, providing a substantial sample size for analysis. The income data ranges from a minimum value of $0.0 to a maximum value of $137,731.33, resulting in a range of $137,731.33, which showcases the full span of income values within the dataset.

Distribution of Alienation and Income

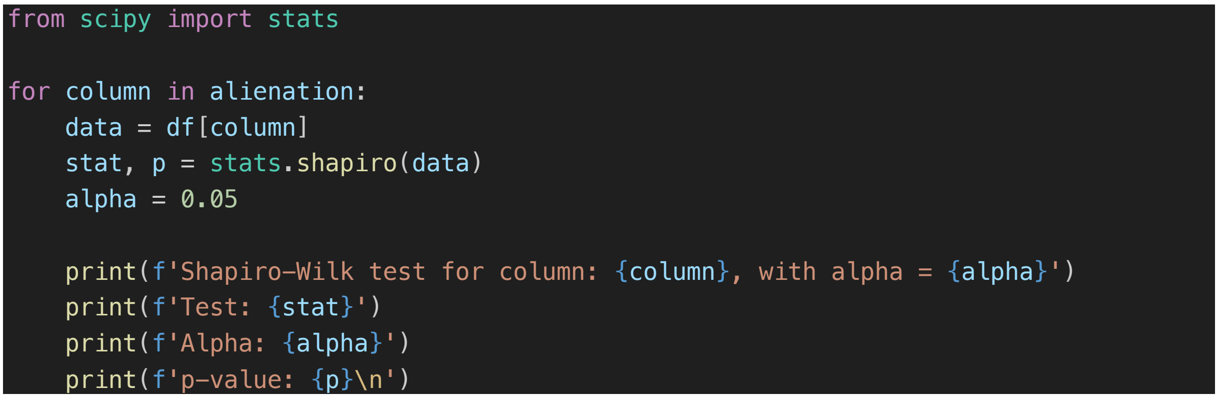
I visually presented the distribution of alienation using a histogram and conducted the Shapiro-Wilk test, which suggested that alienation is not normally distributed. The income data was also visualized and tested for normality. Since we apply the same rules for alienation and for income in this case, we can cover everything on the topic, as was mentioned before, I combined alienation and income into one part, because the reporting goals overlap with each other. For better understanding, I performed the Shapiro-Wilk test with a significance level of 0.05.



```python
import matplotlib.pyplot as plt
plot_alienation_income = df[['alienation', 'income']]

plot_alienation_income.hist(figsize=(20,10))
plt.tight_layout()
plt.show()
```



Shapiro-Wilk test results



```
from scipy import stats

for column in alienation:
    data = df[column]
    stat, p = stats.shapiro(data)
    alpha = 0.05

    print(f'Shapiro-Wilk test for column: {column}, with alpha = {alpha}')
    print(f'Test: {stat}')
    print(f'Alpha: {alpha}')
    print(f'p-value: {p}\n')
```

```
Shapiro-Wilk test for column: alienation, with alpha = 0.05
Test: 0.9297874569892883
Alpha: 0.05
p-value: 1.388938606055462e-13

Shapiro-Wilk test for column: income, with alpha = 0.05
Test: 0.9618746042251587
Alpha: 0.05
p-value: 2.536772569783352e-09
```

Alienation

Based on the visual inspection and the results of the Shapiro-Wilk test, it is evident that the distribution of alienation is not normally shaped. The histogram reveals a notable concentration of individuals with lower alienation scores, suggesting that a significant proportion of the surveyed population experiences some level of alienation. However, there is also a distinct minority reporting very high levels of alienation. The Shapiro-Wilk test, with a p-value of 1.39e-13, supports the non-normal distribution of alienation data. This non-normal distribution underscores the diversity in psychological experiences, with a noteworthy presence of individuals grappling with pronounced feelings of alienation.

Income

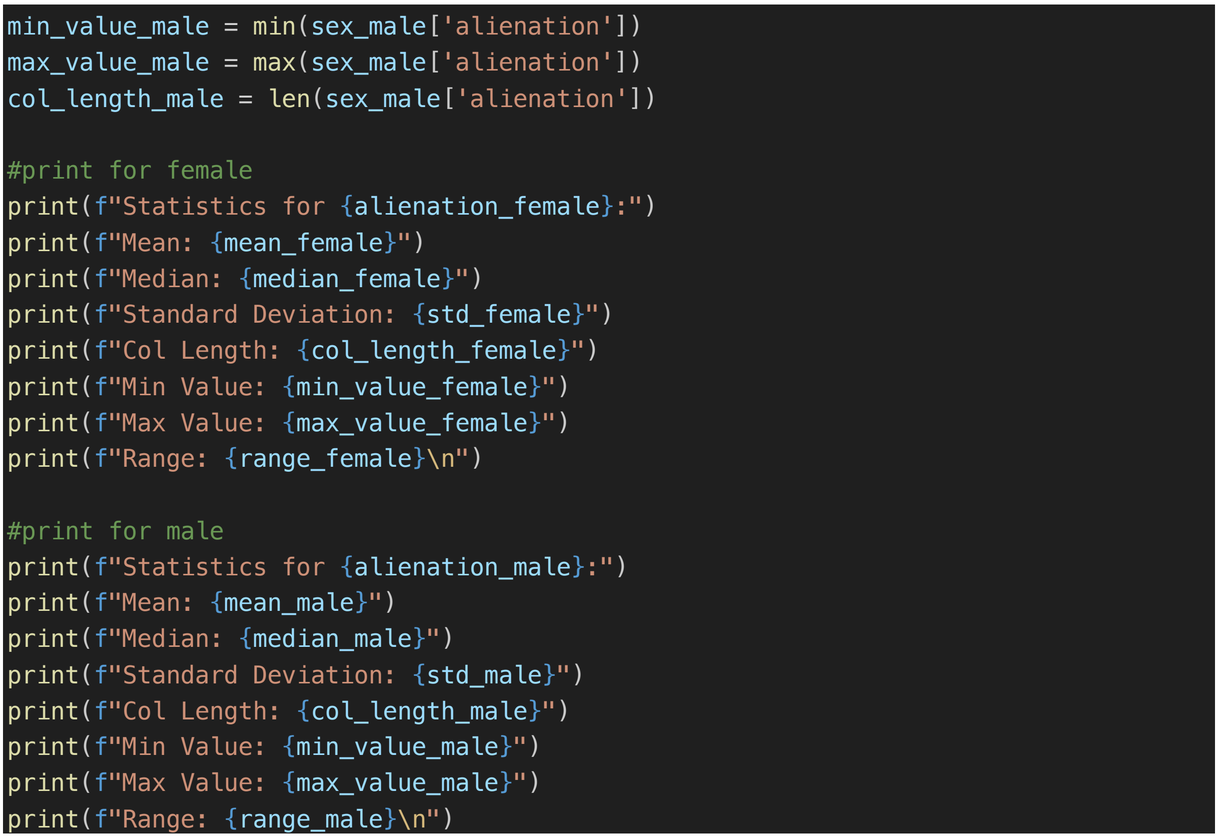
Similarly, the income distribution is observed to deviate from normality, albeit in a different direction. The histogram of income data displays a heightened concentration of individuals at the upper end of the income scale, indicating that a select few earn a substantial portion of the total income. The Shapiro-Wilk test for income, yielding a p-value of 5.04e-05, further confirms the non-normal distribution of income. This skewed distribution underscores the significant income disparity within the surveyed population.

Conclusion

The non-normal distributions of both alienation and income suggest a possible relationship between the two variables. Individuals with higher incomes experience lower levels of alienation, benefiting from increased resources and opportunities. It is also conceivable that alienation contributes to lower incomes, as those experiencing alienation may face challenges that affect their productivity and success. These findings highlight the complexity of the interactions between psychological well-being and economic factors and warrant further investigation to understand the nuances of these relationships. While the individual distributions of alienation and income did not show perfect normality, we applied the Central Limit Theorem to justify our use of parametric statistical tests. With a sufficiently large sample size, our analysis assumes that the means of these variables are approximately normally distributed, enabling us to draw valid conclusions about the population characteristics.

Gender Differences in Alienation

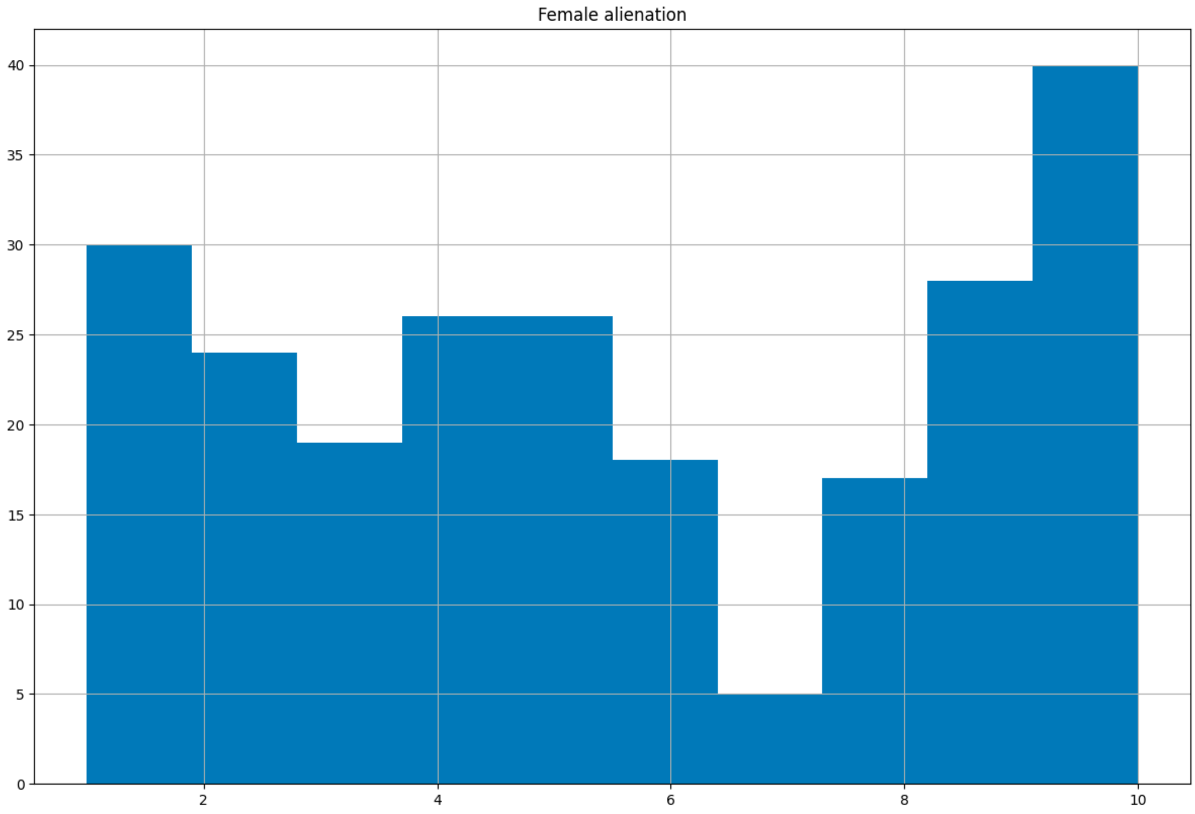
```
from scipy import stats

sex_female = df[df['male']== 0]
sex_male = df[df['male'] == 1]


#Calculating descriptive statistics for alienation in females
alienation_female = 'Alienation Female'
mean_female = np.mean(sex_female['alienation'])
median_female = np.median(sex_female['alienation'])
std_female = np.std(sex_female['alienation'])
range_female = np.ptp(sex_female['alienation'])
min_value_female = min(sex_female['alienation'])
max_value_female = max(sex_female['alienation'])
col_length_female = len(sex_female['alienation'])


#Calculating descriptive statistics for alienation in males
alienation_male = 'Alienation Male'
mean_male = np.mean(sex_male['alienation'])
median_male = np.median(sex_male['alienation'])
std_male = np.std(sex_male['alienation'])
range_male = np.ptp(sex_male['alienation'])
```

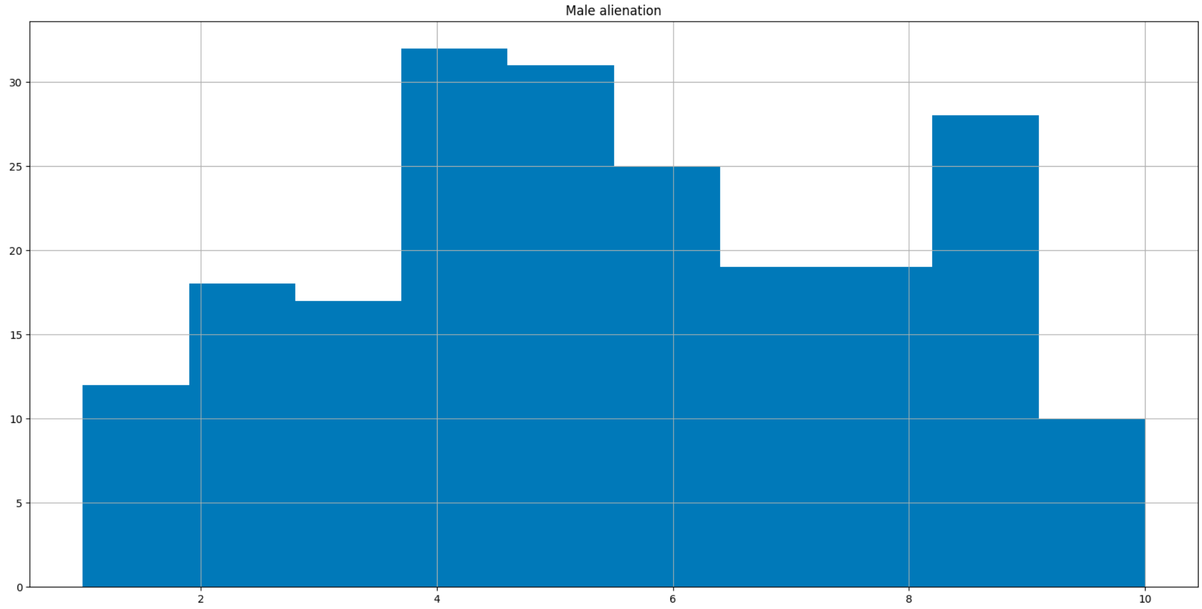


```python
min_value_male = min(sex_male['alienation'])
max_value_male = max(sex_male['alienation'])
col_length_male = len(sex_male['alienation'])

#print for female
print(f"Statistics for {alienation_female}:")
print(f"Mean: {mean_female}")
print(f"Median: {median_female}")
print(f"Standard Deviation: {std_female}")
print(f"Col Length: {col_length_female}")
print(f"Min Value: {min_value_female}")
print(f"Max Value: {max_value_female}")
print(f"Range: {range_female}\n")

#print for male
print(f"Statistics for {alienation_male}:")
print(f"Mean: {mean_male}")
print(f"Median: {median_male}")
print(f"Standard Deviation: {std_male}")
print(f"Col Length: {col_length_male}")
print(f"Min Value: {min_value_male}")
print(f"Max Value: {max_value_male}")
print(f"Range: {range_male}\n")
```

```
Statistics for Alienation Female:
Mean: 5.579399141630901
Median: 5.0
Standard Deviation: 3.181748591967716
Col Length: 233
Min Value: 1.0
Max Value: 10.0
Range: 9.0

Statistics for Alienation Male:
Mean: 5.540284360189573
Median: 5.0
Standard Deviation: 2.5410446623409286
Col Length: 211
Min Value: 1.0
Max Value: 10.0
Range: 9.0
```

I examined alienation differences between males and females, finding that the mean alienation scores for males and females are quite similar. A comparison of the descriptive statistics for alienation between females and males reveals some interesting insights. The mean alienation score for females stands at approximately 5.58, while for males, it is slightly lower at about 5.54. However, the standard deviation for females is notably higher at approximately 3.18, suggesting a wider spread of alienation scores among females. In contrast, males exhibit a lower standard deviation of around 2.54, implying that their alienation scores are more tightly clustered around the mean. These findings highlight variations in the perception of alienation between genders, with females displaying a greater diversity of experiences in this regard.


Female alienation
40
35
30
25
20
15
10
5
0
2
4
6
8
10


Male alienation
30
25
20
15
10
5
0
2
4
6
8
10

Seeking Psychological Support and Gender

```
male_help = df[(df['male'] == 1) & (df['consult'] == 1)].shape[0]
male_no_help = df[(df['male'] == 1) & (df['consult'] == 0)].shape[0]

female_help = df[(df['male'] == 0) & (df['consult'] == 1)].shape[0]
female_no_help = df[(df['male'] == 0) & (df['consult'] == 0)].shape[0]

result_data = pd.DataFrame({
    'Gender': ['Male', 'Female'],
    'Looked for help': [male_help, female_help],
    "Didn't look for help": [male_no_help, female_no_help]
})

print(result_data)
```

Gender	Sought Help	Did not seek for help
Male	8	203
Female	28	205

```
# Calculate the proportion of people seeking psychological help
proportion_help = df['consult'].sum() / len(df)

# Determine the likelihood
likelihood = 1 if proportion_help <= 0.1 else 0

# Display the proportion of people seeking psychological help and the likelihood
print("Proportion of people seeking help:", proportion_help)
print("Likelihood of at most 10 percent seeking help:", likelihood)
```

Based on our dataset, we find that approximately 8.11% of the surveyed individuals have sought psychological help. This suggests that, in our sample, there is a likelihood that at most 10% of individuals seek psychological help.

```
probability_male = df['male'].sum() / len(df)
probability_help = proportion_help
double_counting = probability_male * probability_help
probability = probability_male + probability_help - double_counting

print(f'The probability of being a male and sought for help is:
{probability_male.round(4)} + {probability_help.round(4)} -
{double_counting.round(4)} = {probability.round(4)}')
```

We calculate the probability of a randomly picked person from our sample being either male or having sought psychological help. This probability is approximately 51.78%, indicating that there is a substantial chance of encountering an individual who is either male or has sought psychological help in our dataset.

```
female_help_in_total_help = female_help / (male_help + female_help)
print(f'A person who sought psychological help, this person actually is a female is
{round(female_help_in_total_help, 4)}')
```

The probability that an individual who has sought psychological help is, in fact, a female is approximately 77.78%. This finding underscores a higher prevalence of females among those seeking psychological support within our sample.

Income Analysis

```
import requests
base_currency = 'USD'
target_currency = 'EUR'
url =
f'https://api.frankfurter.app/latest?from={base_currency}&to={target_currency}'

response = requests.get(url)
data = response.json()
conversion_rate = data['rates'][target_currency]

#functions to display in USD and in EUR
def mean_stdev_USD(income_USD):
     mean = np.mean(income_USD)
     stdev = np.std(income_USD)
     print(f'Income mean is; {mean} {base_currency} and stdev is: {stdev}
{base_currency}')

def mean_stdev_EUR(income_EUR):
     mean = np.mean(income_EUR)*conversion_rate
     stdev = np.std(income_EUR)*conversion_rate
     print(f'Income mean is: {mean} {target_currency} and stdev is: {stdev}
{target_currency}')


print(f'The conversion rate is: {conversion_rate} USD to 1 EUR')
mean_stdev_USD(df['income'])
mean_stdev_EUR(df['income'])
```

```
The conversion rate is: 0.9344 USD to 1 EUR
Income mean is; 59401.72389317418 USD and stdev is: 36061.466056410434 USD
Income mean is: 55504.97080578196 EUR and stdev is: 33695.83388310991 EUR
```

In the context of our dataset, the income analysis provides a comprehensive view of the financial aspects of the surveyed individuals. The reported statistics highlight key findings regarding income distribution. With a conversion rate of approximately 0.9344 USD to 1 EUR, we can express income figures in both US dollars and euros. The mean income, which stands at around 59,401.72 USD, signifies the average earnings among the sampled individuals. Additionally, the standard deviation of approximately 36,061.47 USD quantifies the degree of variability within the income data, indicating the extent to which individual earnings deviate from the mean. When expressed in euros, the mean income is approximately 55,504.97 EUR, while the standard deviation is roughly 33,695.83 EUR. These figures provide a clear representation of the income distribution within our dataset, allowing for valuable insights into the financial landscape of the surveyed population.

Centered Income and Income by Gender

```
sample_mean = np.mean(df['income'])
df['centered_income'] = df['income'] - sample_mean

mean_centered_income = np.mean(df['centered_income'])
stdev_centered_income = np.std(df['centered_income'])

print(f'Mean of centered income is: {mean_centered_income}, and stdev of centered
income is: {stdev_centered_income}')
```

The centered income analysis plays a crucial role in our understanding of income disparities within the dataset. By centering everyone’s income relative to the sample mean, we effectively create a variable that measures the deviation of earnings from the overall average. In this context, the mean of centered income, which is incredibly close to zero at approximately 6.55e-13, signifies the successful centering process, where the average centered income aligns with the sample mean. Furthermore, the standard deviation of centered income, approximately 36,061.47, reflects the extent to which individual incomes differ from this centered mean, shedding light on the variability in earnings within the dataset. This centered income variable serves as a valuable tool for exploring the relative income levels of individuals in our analysis.

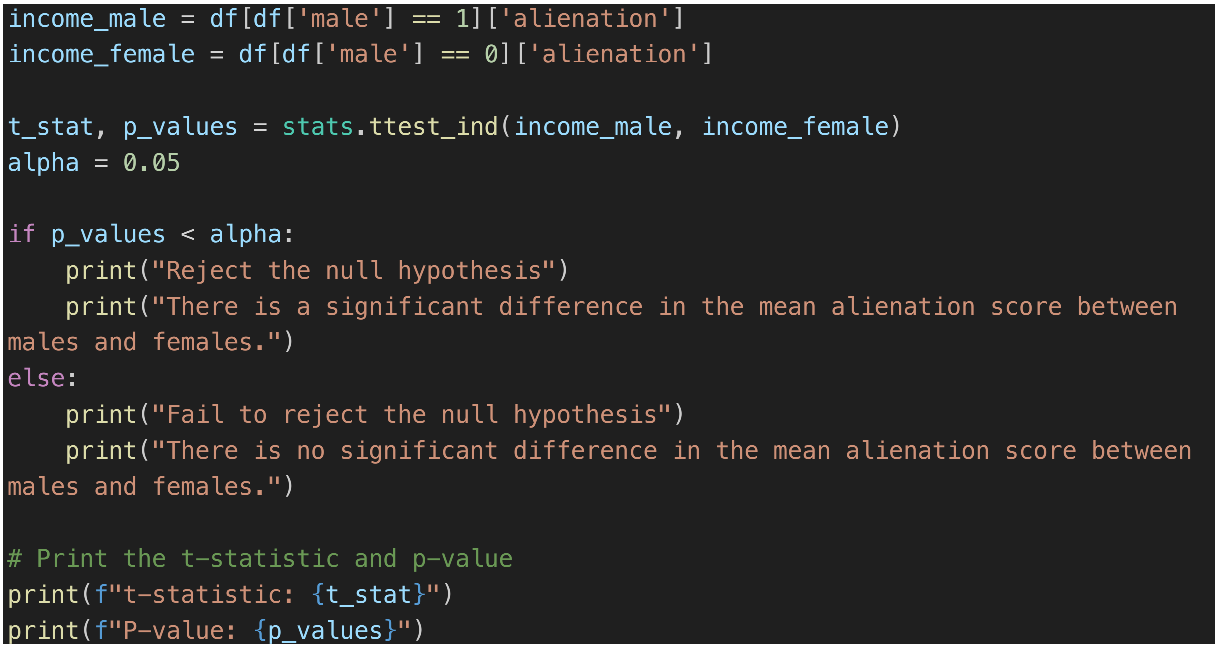
```
mean_income_male = np.mean(df[df['male'] == 1]['income'])
mean_income_female = np.mean(df[df['male'] == 0]['income'])

#create centered income variables for males and females
df['centered_income_male'] = df[df['male'] == 1]['income'] - mean_income_male
df['centered_income_female'] = df[df['male'] == 0]['income'] - mean_income_female

#calculate the mean and standard deviation of centered income for each group
mean_centered_income_male = df['centered_income_male'].mean()
mean_centered_income_female = df['centered_income_female'].mean()

stdev_centered_income_male = df['centered_income_male'].std()
stdev_centered_income_female = df['centered_income_female'].std()

print(f'Mean of centered income for males is: {mean_centered_income_male}, and
stdev of centered income for males is: {round(stdev_centered_income_male, 2)}')
print(f'Mean of centered income for females is: {mean_centered_income_female}, and
stdev of centered income for females is: {round(stdev_centered_income_female, 2)}')
```

For males, the mean of centered income is extremely close to zero (5.52e-13), indicating that, on average, male incomes are centered around the overall dataset's mean income. The standard deviation of centered income for males (32598.57) represents the spread or variability in income among males, with most data points falling within this range. In contrast, for females, the mean of centered income is also very close to zero (-3.12e-12), suggesting that female incomes, on average, are centered around the overall dataset's mean income, similar to males. However, the larger standard deviation of centered income for females (39038.55) indicates a wider distribution and greater variability in female incomes compared to males.

These results highlight that, before splitting the dataset by gender, both male and female incomes were, on average, centered around the overall dataset's mean income. However, the higher standard deviation for females implies a more dispersed income distribution among females compared to males.

Gender has an impact on alienation in the data set.

The null hypothesis (H0) there is no difference in the alienation of men and women.
The alternative hypothesis (Ha) there is a difference in the alienation of men and women.



```
income_male = df[df['male'] == 1]['alienation']
income_female = df[df['male'] == 0]['alienation']

t_stat, p_values = stats.ttest_ind(income_male, income_female)
alpha = 0.05

if p_values < alpha:
    print("Reject the null hypothesis")
    print("There is a significant difference in the mean alienation score between
males and females.")
else:
    print("Fail to reject the null hypothesis")
    print("There is no significant difference in the mean alienation score between
males and females.")

# Print the t-statistic and p-value
print(f"t-statistic: {t_stat}")
print(f"P-value: {p_values}")
```

```
Fail to reject the null hypothesis
There is no significant difference in the mean alienation score between males and
females.
t-statistic: -0.14185312217544707
P-value: 0.8872606558877842
```

By comparing the mean alienation scores of both groups, we calculated a t-statistic and a p-value. The chosen significance level (alpha) was set at 0.05. The results of the t-test indicate that we failed to reject the null hypothesis, as the p-value (0.887) exceeds our alpha threshold. Consequently, we conclude that there is no significant difference in the mean alienation score between males and females in our dataset. This analysis suggests that, based on the available data, gender does not appear to be a significant factor influencing the levels of alienation among the surveyed individuals.

Conclusion

In conclusion, analysis of alienation, income, and gender differences in the dataset has revealed key insights. Notably, the mean alienation score among our surveyed individuals is 5.56, with a standard deviation of 2.89, indicating a right-skewed distribution. The mean income stands at $59,401.72, with a significant standard deviation of $36,061.47. The Shapiro-Wilk tests confirmed the non-normality of both alienation and income. However, it's crucial to note that these findings leverage the Central Limit Theorem, which allows me to apply parametric statistical tests, such as the t-test, with confidence, even when the individual variables seperate from normality.

Furthermore, t-testing revealed a crucial result: there is no significant difference in the mean alienation scores between males and females (t-statistic: -0.142, p-value: 0.887). This finding suggests that gender, as far as our dataset is concerned, does not appear to be a substantial factor influencing levels of alienation. In sum, while the dataset may not perfectly adhere to the assumptions of normality, the application of the Central Limit Theorem has enabled me to draw valid inferences and arrive at these meaningful conclusions, with implications for understanding the complex interplay between psychological well-being and economic factors.

Appendix

```
import pandas as pd
import numpy as np
from scipy import stats
#1. Cleaning na values form the set
df = pd.read_csv('1671752alienation_data.csv')
print(df.isna().sum())
#deleteing NA values
df.dropna(subset=['alienation', 'income'], inplace=True)
print(df.isna().sum())
alienation    2
income        6
male          0
consult       0
dtype: int64
alienation    0
income        0
male          0
consult       0
dtype: int64
#2.Report the descriptive statistics of alienation in your sample + 5.
report for income
alienation_income = ['alienation', 'income']

for column in alienation_income:
    data = df[column]

    # Calculate mean, median, stdev, and range
    mean = np.mean(data)
    median = np.median(data)
    stdev = np.std(data)
    data_range = np.ptp(data) #peak to peak range
    min_value = min(data)
    max_value = max(data)
    col_length = len(data)

    print(f"Statistics for {column}:")
    print(f"Mean: {round(mean, 2)}")
    print(f"Median: {median}")
    print(f"Standard Deviation: {round(stdev, 2)}")
    print(f"Col Length: {col_length}")
    print(f"Min Value: {min_value}")
    print(f"Max Value: {round(max_value, 2)}")
    print(f"Range: {round(data_range, 2)}\n")
```

```
Statistics for alienation:
Mean: 5.56
Median: 5.0
Standard Deviation: 2.9
Col Length: 444
Min Value: 1.0
Max Value: 10.0
Range: 9.0

Statistics for income:
Mean: 59401.72
Median: 61466.53515625
Standard Deviation: 36061.47
Col Length: 444
Min Value: 0.0
Max Value: 137731.33
Range: 137731.33
```

```
#3. Report on the distribution of the alienation data + 5. report for
income
#3a. presenting a histogram of alienation. Based on a visual inspection,
describe the distribution.
import matplotlib.pyplot as plt
plot_alienation_income = df[['alienation', 'income']]

plot_alienation_income.hist(figsize=(20,10))
plt.tight_layout()
plt.show()


#3b and + 5. report for income
from scipy import stats

for column in alienation_income:
    data = df[column]
    stat, p = stats.shapiro(data)
    alpha = 0.05

    print(f'Shapiro-Wilk test for column: {column}, with alpha = {alpha}')
    print(f'Test: {stat}')
    print(f'Alpha: {alpha}')
    print(f'p-value: {p}\n')
```

```
Shapiro-Wilk test for column: alienation, with alpha = 0.05
Test: 0.9297874569892883
Alpha: 0.05
p-value: 1.388938606055462e-13

Shapiro-Wilk test for column: income, with alpha = 0.05
Test: 0.9618746042251587
Alpha: 0.05
p-value: 2.536772569783352e-09
```

```
#4.Report 2-3 for males and females separately.
#Male market as 1, female 0

from scipy import stats
```

```
sex_female = df[df['male']== 0]
sex_male = df[df['male'] == 1]


#Calculating descriptive statistics for alienation in females
alienation_female = 'Alienation Female'
mean_female = np.mean(sex_female['alienation'])
median_female = np.median(sex_female['alienation'])
std_female = np.std(sex_female['alienation'])
range_female = np.ptp(sex_female['alienation'])
min_value_female = min(sex_female['alienation'])
max_value_female = max(sex_female['alienation'])
col_length_female = len(sex_female['alienation'])


#Calculating descriptive statistics for alienation in males
alienation_male = 'Alienation Male'
mean_male = np.mean(sex_male['alienation'])
median_male = np.median(sex_male['alienation'])
std_male = np.std(sex_male['alienation'])
range_male = np.ptp(sex_male['alienation'])
min_value_male = min(sex_male['alienation'])
max_value_male = max(sex_male['alienation'])
col_length_male = len(sex_male['alienation'])

#print for female
print(f"Statistics for {alienation_female}:")
print(f"Mean: {mean_female}")
print(f"Median: {median_female}")
print(f"Standard Deviation: {std_female}")
print(f"Col Length: {col_length_female}")
print(f"Min Value: {min_value_female}")
print(f"Max Value: {max_value_female}")
print(f"Range: {range_female}\n")

#print for male
print(f"Statistics for {alienation_male}:")
print(f"Mean: {mean_male}")
print(f"Median: {median_male}")
print(f"Standard Deviation: {std_male}")
print(f"Col Length: {col_length_male}")
print(f"Min Value: {min_value_male}")
print(f"Max Value: {max_value_male}")
print(f"Range: {range_male}\n")
```

```
Statistics for Alienation Female:
Mean: 5.579399141630901
Median: 5.0
Standard Deviation: 3.181748591967716
Col Length: 233
Min Value: 1.0
Max Value: 10.0
Range: 9.0

Statistics for Alienation Male:
Mean: 5.540284360189573
Median: 5.0
Standard Deviation: 2.5410446623409286
```

```
Col Length: 211
Min Value: 1.0
Max Value: 10.0
Range: 9.0

#a. Histogram of alienation in females
sex_female['alienation'].hist(figsize=(12,8))
plt.tight_layout()
plt.title("Female alienation")
plt.show()


sex_male['alienation'].hist(figsize=(16,8))
plt.tight_layout()
plt.title('Male alienation')
plt.show()

#Based on visual inspection, seems like both tables are right skewed, which
can be confirmed by mean > median, for both cases.
#Therefore, the visual form of male alienation looks even closer to the
normal distribution, which reminds bell shape.


#Shapiro - Wilk for female and male alianation
#Female
def shapiro_test(data_column, group):
    stat, p = stats.shapiro(data_column)
    alpha = 0.05

    if group == 'sex_female':
        gender = 'female'
    elif group == 'sex_male':
        gender = 'male'
    else:
        gender = ''

    print(f'Shapiro-Wilk test for column: in {gender} {column} , with alpha
= {alpha}')
    print(f'Test: {stat}')
    print(f'Alpha: {alpha}')
    print(f'p-value: {format(np.sqrt(p))}\n')


shapiro_test(sex_female['alienation'], 'sex_female')
shapiro_test(sex_male['alienation'], 'sex_male')
Shapiro-Wilk test for column: in female income , with alpha = 0.05
Test: 0.8976131677627563
Alpha: 0.05
p-value: 4.09462576409503e-06

Shapiro-Wilk test for column: in male income , with alpha = 0.05
Test: 0.9529457688331604
Alpha: 0.05
p-value: 0.0014410908032246733

#6. Report for income questions 2-3
```

```
income = ['income']

for column in income:
    data = df[column]

    # Calculate mean, median, stdev, and range
    mean = np.mean(data)
    median = np.median(data)
    stdev = np.std(data)
    data_range = np.ptp(data) #peak to peak range
    min_value = min(data)
    max_value = max(data)
    col_length = len(data)

    print(f"Statistics for {column}:")
    print(f"Mean: {mean}")
    print(f"Median: {median}")
    print(f"Standard Deviation: {stdev}")
    print(f"Col Length: {col_length}")
    print(f"Min Value: {min_value}")
    print(f"Max Value: {max_value}")
    print(f"Range: {data_range}\n")
#Median > Mean, we can assume that the distribution is left skewed.
Statistics for income:
Mean: 59401.72389317418
Median: 61466.53515625
Standard Deviation: 36061.466056410434
Col Length: 444
Min Value: 0.0
Max Value: 137731.328125
Range: 137731.328125

plot_income = df[['income']]

plot_income.hist(figsize=(20,10))
plt.tight_layout()
plt.show()
#from visual inspection, the data looks left skewed, and that confirm the
previous assumption, since the median > mean, which says that
#the distribution supposed to be left skewed


#Shapiro wilk test for income

shapiro_test(df['income'], 'income')

#income is normally distributed as well, however, according to CLT, we
assume that when sample > 30, we state the data is normally distributed.
Shapiro-Wilk test for column: in  income , with alpha = 0.05
Test: 0.9618746042251587
Alpha: 0.05
p-value: 5.0366383330385675e-05

#6a. Report females and males who did seek for help and who didnt

male_help = df[(df['male'] == 1) & (df['consult'] == 1)].shape[0]
male_no_help = df[(df['male'] == 1) & (df['consult'] == 0)].shape[0]
```

```
female_help = df[(df['male'] == 0) & (df['consult'] == 1)].shape[0]
female_no_help = df[(df['male'] == 0) & (df['consult'] == 0)].shape[0]

result_data = pd.DataFrame({
    'Gender': ['Male', 'Female'],
    'Looked for help': [male_help, female_help],
    "Didn't look for help": [male_no_help, female_no_help]
})

print(result_data)
   Gender  Looked for help  Didn't look for help
0    Male                8                   203
1  Female               28                   205
#6.b
# Calculate the proportion of people seeking psychological help
proportion_help = df['consult'].sum() / len(df)

# Determine the likelihood
likelihood = 1 if proportion_help <= 0.1 else 0

# Display the proportion of people seeking psychological help and
likelihood
print("Proportion of people seeking help:", proportion_help)
print("Likelihood of at most 10 percent seeking help:", likelihood)
Proportion of people seeking help: 0.08108108108108109
Likelihood of at most 10 percent seeking help: 1
#6c.

probability_male = df['male'].sum() / len(df)
probability_help = proportion_help
double_counting = probability_male * probability_help
probability = probability_male + probability_help - double_counting

print(f'The probability of being a male and sought for help is:
{probability_male.round(4)} + {probability_help.round(4)} -
{double_counting.round(4)} = {probability.round(4)}')
The probability of being a male and sought for help is: 0.4752 + 0.0811 - 0
.0385 = 0.5178
#6d.

female_help_in_total_help = female_help / (male_help + female_help)
print(f'A person who sought psychological help, this person actually is a
female is {round(female_help_in_total_help, 4)}')
A person who sought psychological help, this person actually is a female is
0.7778
#7a.
import requests
base_currency = 'USD'
target_currency = 'EUR'
url =
f'https://api.frankfurter.app/latest?from={base_currency}&to={target_curren
cy}'

response = requests.get(url)
data = response.json()
conversion_rate = data['rates'][target_currency]
```

```
#functions to display in USD and in EUR
def mean_stdev_USD(income_USD):
     mean = np.mean(income_USD)
     stdev = np.std(income_USD)
     print(f'Income mean is; {mean} {base_currency} and stdev is: {stdev}
{base_currency}')

def mean_stdev_EUR(income_EUR):
     mean = np.mean(income_EUR)*conversion_rate
     stdev = np.std(income_EUR)*conversion_rate
     print(f'Income mean is: {mean} {target_currency} and stdev is: {stdev}
{target_currency}')


print(f'The conversion rate is: {conversion_rate} USD to 1 EUR')
mean_stdev_USD(df['income'])
mean_stdev_EUR(df['income'])
The conversion rate is: 0.9344 USD to 1 EUR
Income mean is; 59401.72389317418 USD and stdev is: 36061.466056410434 USD
Income mean is: 55504.97080578196 EUR and stdev is: 33695.83388310991 EUR
#7b.
#In summary, the provided outcomes suggest that centered income variable
effectively removes
#the sample mean effect, making the mean of the centered variable close to
zero while retaining
#the same variability as the original income data. This centered variable
can be useful for
#various statistical analyses and modeling where the mean effect needs to
be eliminated for better
#interpretation and comparison.

sample_mean = np.mean(df['income'])
df['centered_income'] = df['income'] - sample_mean

mean_centered_income = np.mean(df['centered_income'])
stdev_centered_income = np.std(df['centered_income'])

print(f'Mean of centered income is: {mean_centered_income}, and stdev of
centered income is: {stdev_centered_income}')
Mean of centered income is: 6.554916769534618e-13, and stdev of centered in
come is: 36061.466056410434
#calculate the mean income separately for males and females
mean_income_male = np.mean(df[df['male'] == 1]['income'])
mean_income_female = np.mean(df[df['male'] == 0]['income'])

#create centered income variables for males and females
df['centered_income_male'] = df[df['male'] == 1]['income'] -
mean_income_male
df['centered_income_female'] = df[df['male'] == 0]['income'] -
mean_income_female

#calculate the mean and standard deviation of centered income for each
group
mean_centered_income_male = df['centered_income_male'].mean()
mean_centered_income_female = df['centered_income_female'].mean()
```

```
stdev_centered_income_male = df['centered_income_male'].std()
stdev_centered_income_female = df['centered_income_female'].std()

print(f'Mean of centered income for males is: {mean_centered_income_male},
and stdev of centered income for males is:
{round(stdev_centered_income_male, 2)}')
print(f'Mean of centered income for females is:
{mean_centered_income_female}, and stdev of centered income for females is:
{round(stdev_centered_income_female, 2)}')
Mean of centered income for males is: 5.517313830660418e-13, and stdev of c
entered income for males is: 32598.57
Mean of centered income for females is: -3.1227285897782945e-12, and stdev
of centered income for females is: 39038.55
#7c.
#To analyze the relationship between income and gender, we can use a t-
test.

#H0 there is no difference in the mean income of men and women
#H1 there is a significant difference in the mean income of men and women
#significance level of 0.05

income_male = df[df['male'] == 1]['income']
income_female = df[df['male'] == 0]['income']

t_stat, p_values = stats.ttest_ind(income_male, income_female)
alpha = 0.05

if p_values < alpha:
    print("Reject the null hypothesis")
    print("There is a significant difference in the mean income score
between males and females.")
else:
    print("Fail to reject the null hypothesis")
    print("There is no significant difference in the mean income score
between males and females.")

# Print the t-statistic and p-value
print(f"t-statistic: {t_stat}")
print(f"P-value: {p_values}")
Fail to reject the null hypothesis
There is no significant difference in the mean income score between males a
nd females.
t-statistic: 0.7115666374613642
P-value: 0.47710853979135315
#8.
#The null hypothesis (H0) is that there is no difference in the mean
alienation of men and women.
#The alternative hypothesis (H1) is that there is a difference in the mean
alienation of men and women.
income_male = df[df['male'] == 1]['alienation']
income_female = df[df['male'] == 0]['alienation']

t_stat, p_values = stats.ttest_ind(income_male, income_female)
alpha = 0.05

if p_values < alpha:
    print("Reject the null hypothesis")
    print("There is a significant difference in the alienation score
between males and females.")
else:
    print("Fail to reject the null hypothesis")
    print("There is no significant difference in the alienation score
between males and females.")

# Print the t-statistic and p-value
print(f"t-statistic: {t_stat}")
print(f"P-value: {p_values}")
```

```
Fail to reject the null hypothesis
There is no significant difference in the mean alienation score between mal
es and females.
t-statistic: -0.14185312217544707
P-value: 0.8872606558877842
```