Statistical Analysis and Visualization with Python

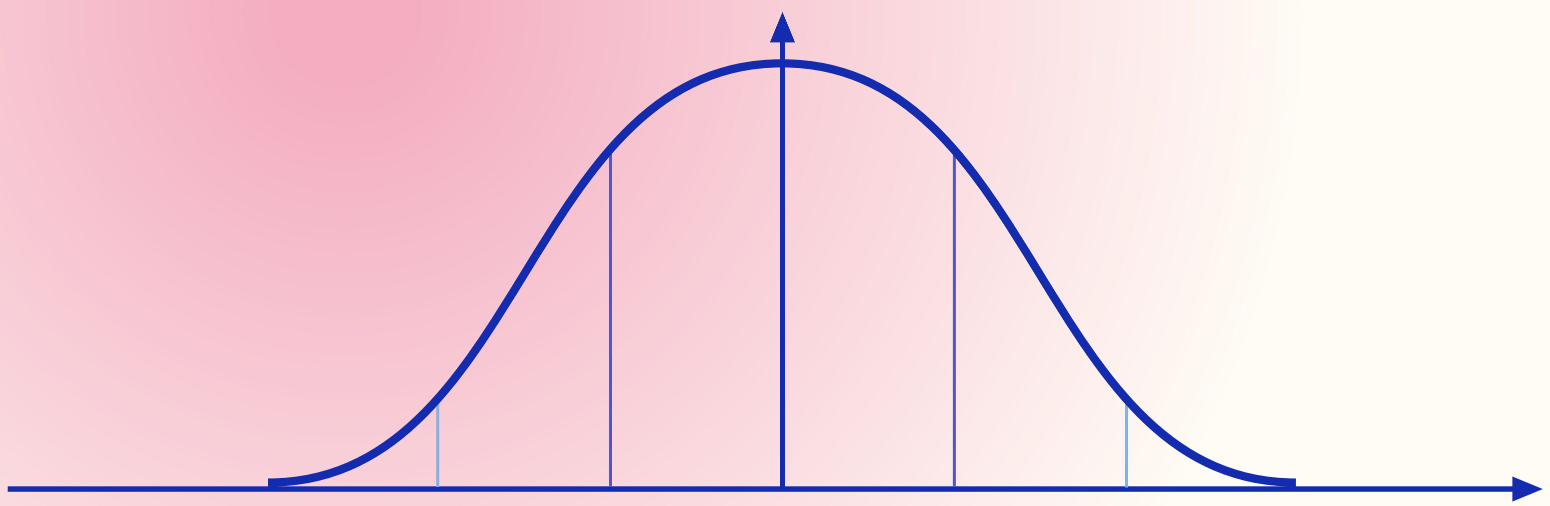
16/06/2024

Yossi
TAPIERO

01 - Introduction

- **Objective**: Perform statistical analysis and data visualization using a synthetic dataset.
- **Overview**: Generate dataset, perform analyses, create visualizations, and draw insights.

02 - Methodology



- Used NumPy to create a dataset with 1000 samples.
- Columns: Age, Height, Weight, Gender, and Income.
- Normal distributions for Age, Height, Weight, and Income.
- Random assignment for Gender.

How Did You Generate the Synthetic Dataset?

02 - Methodology

Why Normal Distributions?

- Real-world phenomena often follow normal distributions (Central Limit Theorem).
- Simplifies data generation with controllable mean and standard deviation.

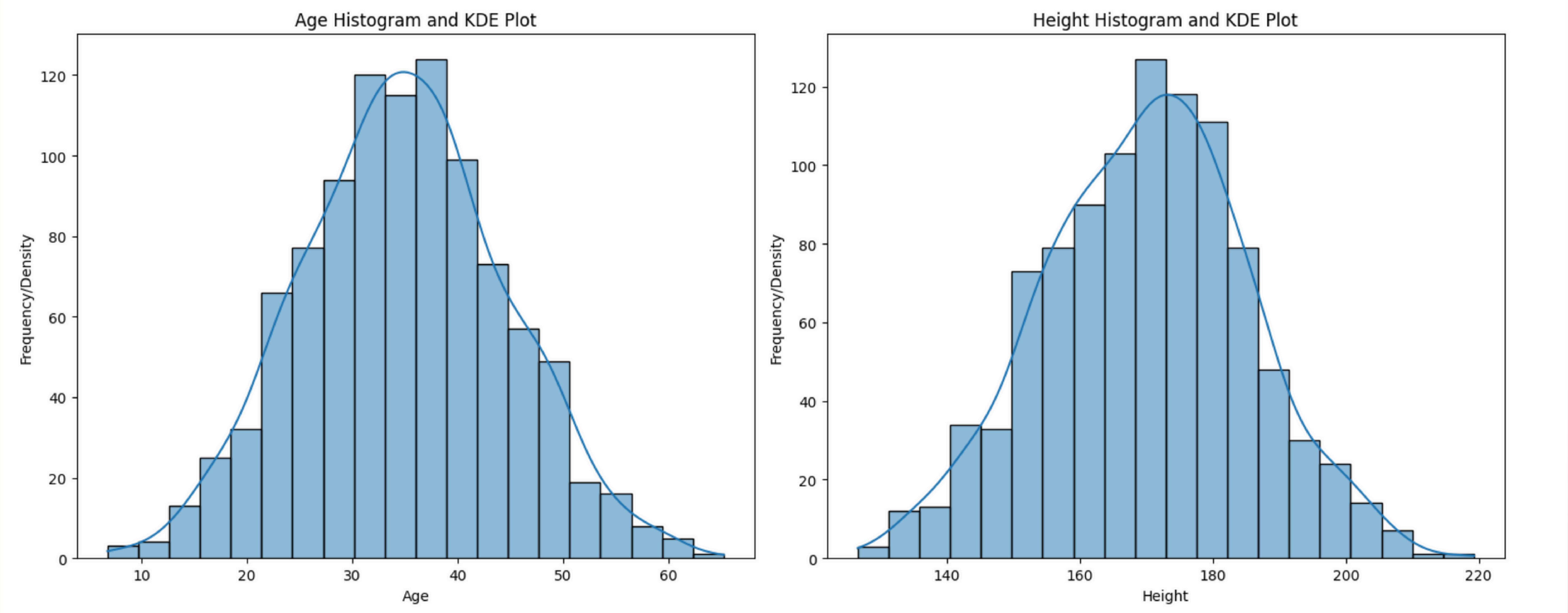
03 - Descriptive Statistics

	Age	Height	Weight	Gender	Income
0	28.984305	179.136700	69.103686	Female	43927.448994
1	30.575528	182.439736	69.708583	Male	69184.526443
2	38.992909	176.255734	61.596764	Female	46650.332896
3	39.543098	157.991884	67.202764	Female	50693.207819
4	59.308231	170.636188	57.098455	Female	33694.459229

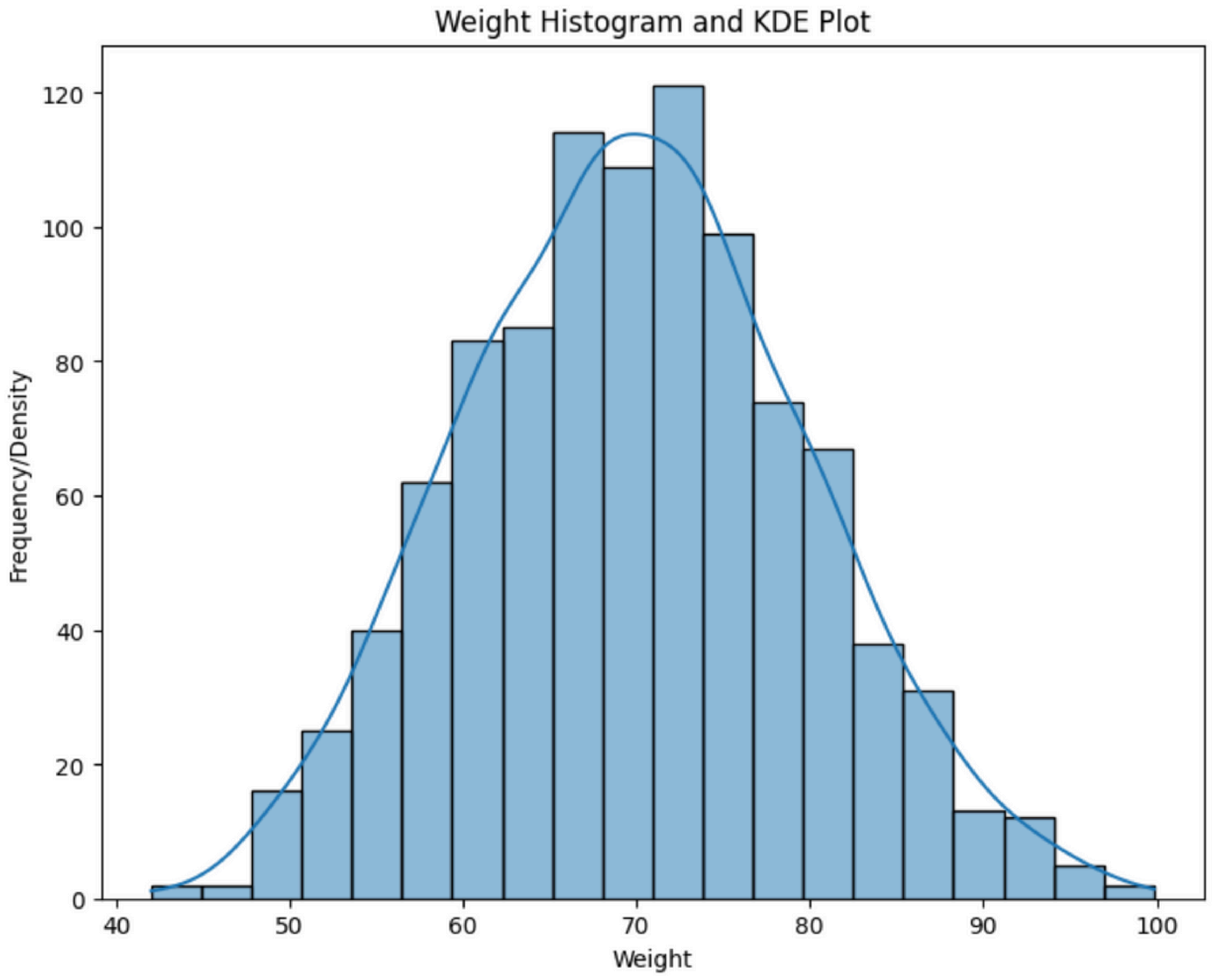
```
data.head()
```

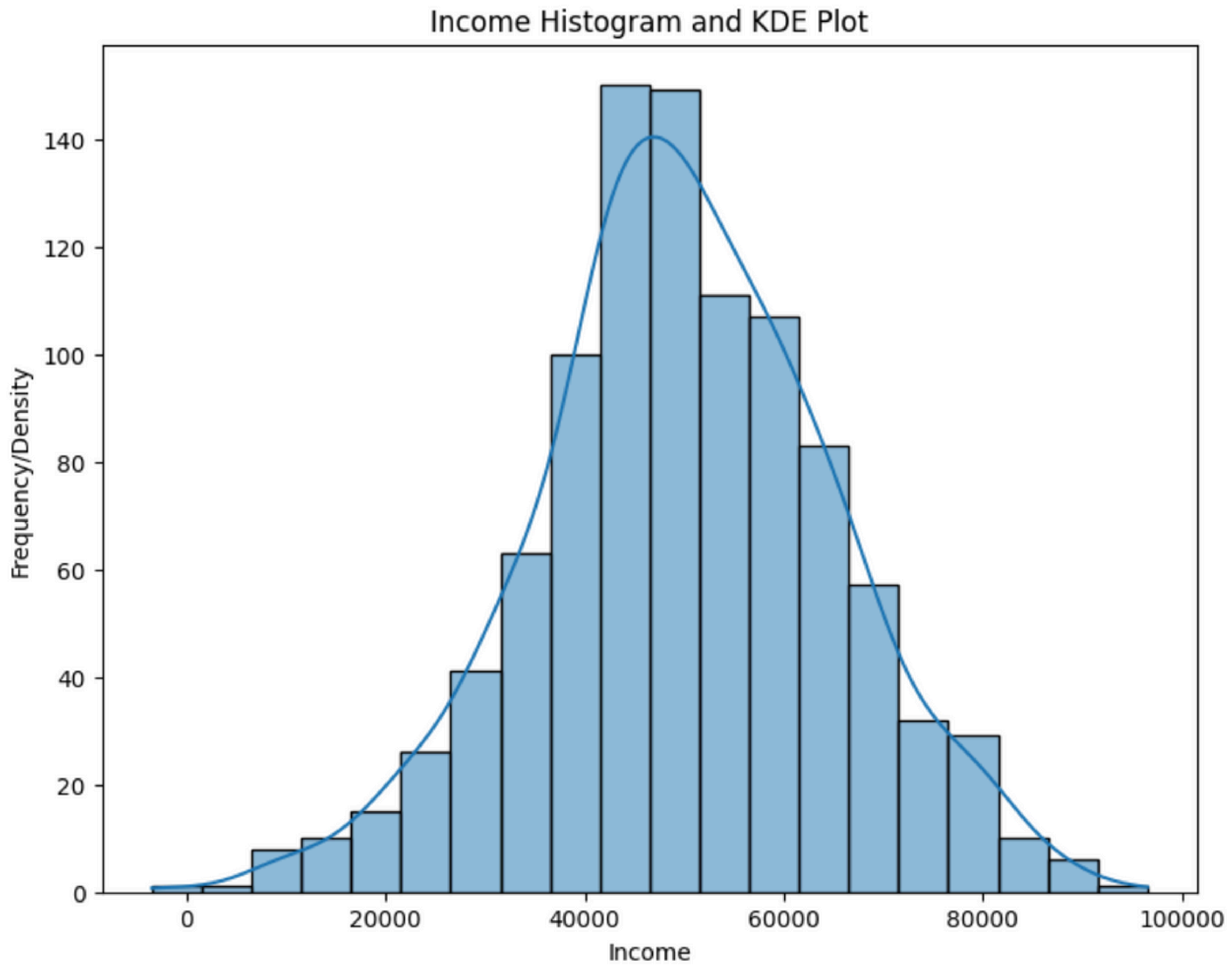
```
Mean :
Age            34.895440
Height        169.949639
Weight         69.913384
Income      49932.559831
dtype: float64

Median :
Age            34.856857
Height        170.615221
Weight         69.557417
Income      49601.915809
dtype: float64

Standard Deviation :
Age             9.618064
Height         15.370160
Weight          9.779957
Income      15024.479725
dtype: float64

Variance :
Age         9.250715e+01
Height      2.362418e+02
Weight      9.564756e+01
Income      2.257350e+08
dtype: float64

Mode :
0    Female
Name: Gender, dtype: object
```

04 – Visualization

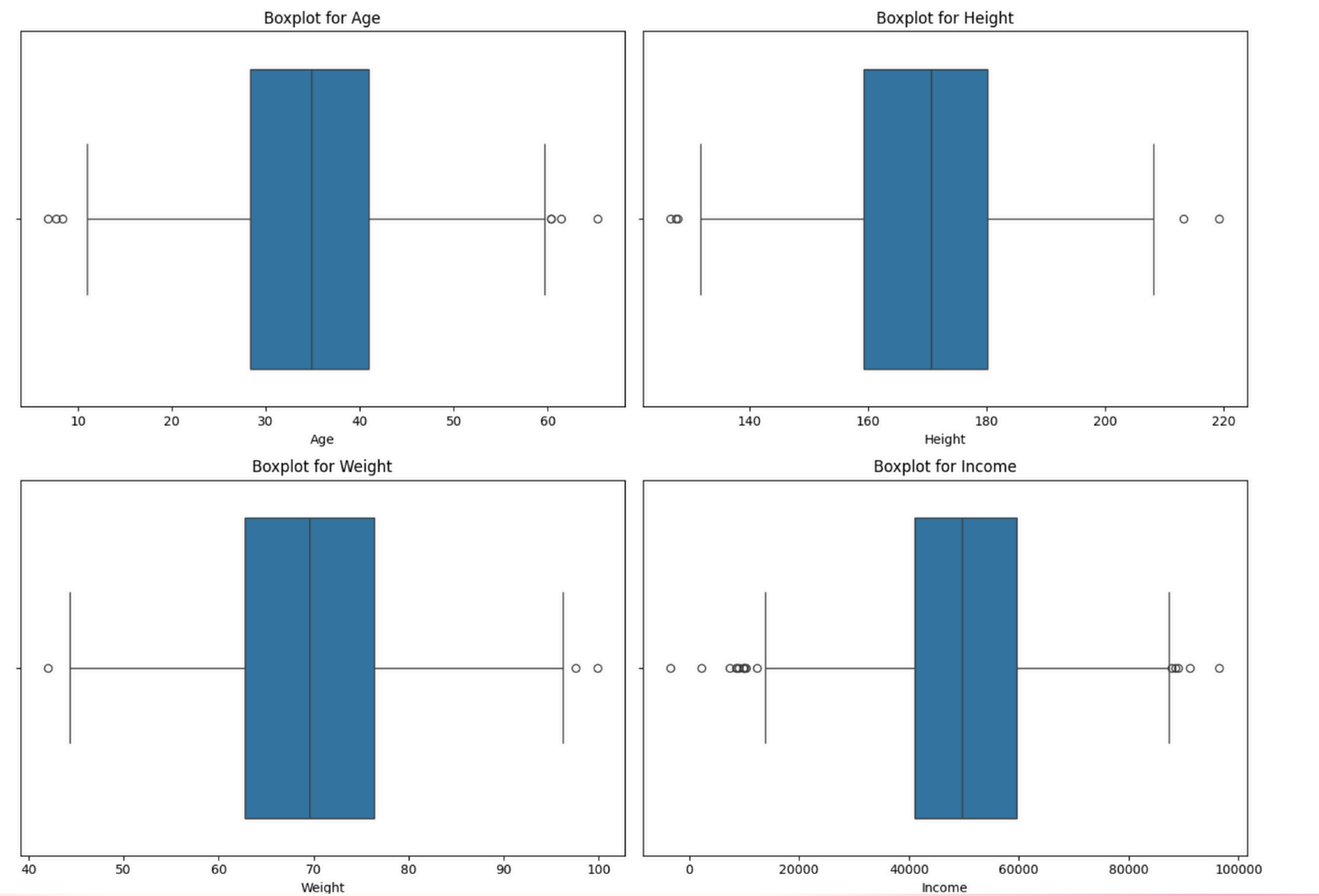


04 – Visualization





04 - Visualization



05 - Technical questions

How did you generate the synthetic dataset, and why did you choose normal distributions for the variables?

I used a function that generates random numbers from a normal (Gaussian) distribution - np.random.normal() - , available in the Numpy library. I chose normal distributions because many real-world phenomenas tend to follow normal distribution - cf. Central Limit Theorem (CLT) - and variables like Age, Height, Weight, and income often hightlight characteristics that are approximately normally distributed in real life.

05 - Technical questions

What insights can you draw from the descriptive statistics calculated for Age, Height, Weight, and Income?

- **Insight 1 : Symetry in Age, Height, Weight, and income.**
- **Insight 2 : Income has a significantly higher standard deviation than Age, Weight or Height, indicating substantial variability in income among the individuals in the dataset.**
- **Insight 3 : The mode is 'Female', indicating that the most frequently occurring gender in the dataset is Female.**

05 - Technical questions

What do the KDE plots and histograms tell you about the distribution of the data?

The four distribution appears to be approximately normally distributed, centered around the mean.

05 - Technical questions

How can you interpret the boxplots, and what do they reveal about potential outliers in the dataset?

- **Age:**
  - **Median: ~34 years.**
  - **IQR: 25 to 45 years.**
  - **Outliers: Below 10 and above 60 years.**
- **Height:**
  - **Median: ~170 cm.**
  - **IQR: 155 to 185 cm.**
  - **Outliers: Below 125 cm and above 205 cm.**
- **Weight:**
  - **Median: ~70 kg.**
  - **IQR: 60 to 80 kg.**
  - **Outliers: Below 45 kg and above 95 kg.**
- **Income:**
  - **Median: ~$50,000.**
  - **IQR: 35,000**to**65,000.**
  - **Outliers: Below 5,000**$andabove$**95,000.**

05 - Technical questions

Explain the results of the t-test. What does the p-value indicate about the difference in Income between Male and Female?

```
TtestResult(statistic=0.7462866580138902, pvalue=0.4556700222716189, df=998.0)
```

T- Tests

- **Ho: There is no significant difference in income between Male and Female.**
- **H1: There is a significant difference between Male and Female.**

On basis on T-Test , we have pvalue > 0.05 --> Not enough evidence to reject the null hypothesis, so we accept the Null hypothesis H0

Thanks

Yossi
TAPIERO

06 - Annex

```
# Import the necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sample = 1000

age = np.random.normal(35, 10, sample)
height = np.random.normal(170, 15, sample)
weight = np.random.normal(70, 10, sample)
gender = np.random.choice(['Male','Female'], sample, p=[0.5,0.5])
income = np.random.normal(50000, 15000, sample)

data = pd.DataFrame({'Age': age, 'Height': height, 'Weight': weight, 'Gender': gender, 'Income': income})
print(data.head())
```

06 - Annex

```
data_num = data.select_dtypes(include=['int64', 'float64'])
# Other way : data_num = data.drop(columns = ['Gender'])

print("\nMean : ")
print(data_num.mean())

print("\nMedian : ")
print(data_num.median())

print("\nStandard Deviation : ")
print(data_num.std())

print("\nVariance : ")
print(data_num.var())

print("\nMode : ")
print(data['Gender'].mode())
```

06 - Annex

```
import seaborn as sns
import matplotlib.pyplot as plt

# Create a 2x2 grid for the histograms and KDE plots
fig, axes = plt.subplots(2, 2, figsize=(15, 12))

# Age Histogram and KDE plot
sns.histplot(data['Age'], kde=True, bins=20, ax=axes[0, 0])
axes[0, 0].set_title('Age Histogram and KDE Plot')
axes[0, 0].set_xlabel('Age')
axes[0, 0].set_ylabel('Frequency/Density')

# Height Histogram and KDE plot
sns.histplot(data['Height'], kde=True, bins=20, ax=axes[0, 1])
axes[0, 1].set_title('Height Histogram and KDE Plot')
axes[0, 1].set_xlabel('Height')
axes[0, 1].set_ylabel('Frequency/Density')

# Weight Histogram and KDE plot
sns.histplot(data['Weight'], kde=True, bins=20, ax=axes[1, 0])
axes[1, 0].set_title('Weight Histogram and KDE Plot')
axes[1, 0].set_xlabel('Weight')
axes[1, 0].set_ylabel('Frequency/Density')

# Income Histogram and KDE plot
sns.histplot(data['Income'], kde=True, bins=20, ax=axes[1, 1])
axes[1, 1].set_title('Income Histogram and KDE Plot')
axes[1, 1].set_xlabel('Income')
axes[1, 1].set_ylabel('Frequency/Density')

# Adjust layout
plt.tight_layout()
plt.show()
```

06 - Annex

```
# Creating a grid of boxplots
fig, axes = plt.subplots(2, 2, figsize=(14, 10))

# Boxplot for Age
sns.boxplot(ax=axes[0, 0], x=data['Age'])
axes[0, 0].set_title('Boxplot for Age')
axes[0, 0].set_xlabel('Age')

# Boxplot for Height
sns.boxplot(ax=axes[0, 1], x=data['Height'])
axes[0, 1].set_title('Boxplot for Height')
axes[0, 1].set_xlabel('Height')

# Boxplot for Weight
sns.boxplot(ax=axes[1, 0], x=data['Weight'])
axes[1, 0].set_title('Boxplot for Weight')
axes[1, 0].set_xlabel('Weight')

# Boxplot for Income
sns.boxplot(ax=axes[1, 1], x=data['Income'])
axes[1, 1].set_title('Boxplot for Income')
axes[1, 1].set_xlabel('Income')

# Adjust layout
plt.tight_layout()
plt.show()
```

06 - Annex

```python
from scipy.stats import ttest_ind

# Income of Male
male_income= data[data['Gender'] == 'Male' ]['Income']

# Income of Female
female_income= data[data['Gender'] == 'Female']['Income']

t_stats = ttest_ind(male_income, female_income)
print(t_stats)
```