Revised edition

Teammate: Chen Liyu 2330024014
Chen Yunzhi 2330006022
Yuan Heng 2430025072
Luo JunYao 2230033028

Part 1: Data Analysis (50 Points)

Run a preliminary analysis on the dataset to answer the following questions:

1. **How many records are in the dataset?**

We add some possible exception to catch exceptions that may occur when reading a CSV file, such as if the file does not exist or if the file is empty.

```
import pandas as pd
try:
    data = pd.read_csv("FitTrackData.csv")
    a1 = len(data)
    print(a1)
except FileNotFoundError:
    print("Error: The file 'FitTrackData.csv' does not exist.")

except pd.errors.EmptyDataError:
    print("Error: The file 'FitTrackData.csv' is empty.")

except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

2. **How many unique device models are in the dataset?**

We add some possible exceptions and plan to fill the missing value selection with "unkown" as the DeviceModel column does not contain numeric data.

```
import pandas as pd
try:
    data = pd.read_csv("FitTrackData.csv")
    data.head()
    if 'DeviceModel' not in data.columns:
        raise KeyError("The 'DeviceModel' column does not exist in the data.")

    data['DeviceModel'] = data['DeviceModel'].fillna('Unknown')
    a2 = len(data['DeviceModel'].unique())
    print(a2)

except FileNotFoundError:
    print("Error: The file 'FitTrackData.csv' does not exist.")

except pd.errors.EmptyDataError:
    print("Error: The file 'FitTrackData.csv' is empty.")

except KeyError as e:
    print(f"Error: {e}")

except Exception as e:
    print(f"An unexpected error occurred: {e}") import pandas as pd
```

3. What is the largest age difference in the dataset (maximum age - minimum age)? We add some possible exceptions and if there is a missing value in Age, we intend to use the average of the Age columns to fill in the missing value.

```
import pandas as pd
try:
    data = pd.read_csv("FitTrackData.csv")
    data.head()
    if 'DeviceModel' not in data.columns:
        raise KeyError("The 'DeviceModel' column does not exist in the data.")
    if 'Age' in data.columns:
        data['Age'] = data['Age'].fillna(data['Age'].mean())
    else:
        raise KeyError("The 'Age' column does not exist in the data.")
    a2 = len(data['DeviceModel'].unique())
    print(a2)

except FileNotFoundError:
    print("Error: The file 'FitTrackData.csv' does not exist.")

except pd.errors.EmptyDataError:
    print("Error: The file 'FitTrackData.csv' is empty.")

except KeyError as e:
    print(f"Error: {e}")

except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

4. What is the ratio between 'Female' and 'Male' in the column Gender (Female/Male)? We add some possible exceptions and if there is a missing value in Gender, we plan to fill it with "Unknown".

```
    import pandas as pd
    try:
        data = pd.read_csv("FitTrackData.csv")
        data.head()
        if 'Gender' not in data.columns:
            raise KeyError("The 'Gender' column does not exist in the data.")
        if data['Gender'].isnull().any():
            data['Gender'].fillna('Unknown', inplace=True)
        female_count = len(data[data['Gender'] == 'Female'])
        male_count = len(data[data['Gender'] == 'Male'])
        if male_count == 0:
            raise ZeroDivisionError("All people are female or unknown.")
        a4 = female_count / male_count
        print(f"Female to Male Ratio: {a4}")

    except FileNotFoundError:
        print("Error: The file 'FitTrackData.csv' does not exist.")

    except pd.errors.EmptyDataError:
        print("Error: The file 'FitTrackData.csv' is empty.")

    except KeyError as e:
        print(f"Error: {e}")

    except ZeroDivisionError as e:
        print(f"Error: {e}")

    except Exception as e:
        print(f"An unexpected error occurred: {e}")
```

5. **What is the median number of years of education completed by customers?**

We add some possible exceptions and if there is a missing value in EducationYears, we plan to fill it with the median of EducationYears.

```
import pandas as pd
try:
    data = pd.read_csv("FitTrackData.csv")
    if 'EducationYears' not in data.columns:
        raise KeyError("The 'EducationYears' column does not exist in the data.")

    if data['EducationYears'].isnull().any():
        data['EducationYears'].fillna(data['EducationYears'].median(), inplace=True)
    education_median = data['EducationYears'].median()
    print(education_median)
```

```
except FileNotFoundError:
    print("Error: The file 'FitTrackData.csv' does not exist.")

except pd.errors.EmptyDataError:
    print("Error: The file 'FitTrackData.csv' is empty.")

except KeyError as e:
    print(f"Error: {e}")

except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

6. How many customers are not single?

We add some possible exceptions and if there is a missing value in MaritalStatus, we plan to fill it with "Unknown".

```
import pandas as pd
try:
    data = pd.read_csv("FitTrackData.csv")
    if 'MaritalStatus' not in data.columns:
        raise KeyError("The 'MaritalStatus' column does not exist in the data.")
    if data['MaritalStatus'].isnull().any():
        data['MaritalStatus'].fillna('Unknown', inplace=True)
    not_single_count = (data['MaritalStatus'] != 'Single').sum()
    print(not_single_count)

except FileNotFoundError:
    print("Error: The file 'FitTrackData.csv' does not exist.")

except pd.errors.EmptyDataError:
    print("Error: The file 'FitTrackData.csv' is empty.")

except KeyError as e:
    print(f"Error: {e}")

except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

7. On average, how many times per week do customers use their fitness trackers?

We add some possible exceptions and if there is a missing value in UsageFrequency, we plan to fill it with the mean of UsageFrequency.

```
import pandas as pd
try:
     data = pd.read_csv("FitTrackData.csv")
     if 'UsageFrequency' not in data.columns:
         raise KeyError("The 'UsageFrequency' column does not exist in the data.")
     if data['UsageFrequency'].isnull().any():
         data['UsageFrequency'].fillna(data['UsageFrequency'].mean(), inplace=True)
     avg_usage_frequency = data['UsageFrequency'].mean()
     print(avg_usage_frequency)

 except FileNotFoundError:
     print("Error: The file 'FitTrackData.csv' does not exist.")

 except pd.errors.EmptyDataError:
     print("Error: The file 'FitTrackData.csv' is empty.")

 except KeyError as e:
     print(f"Error: {e}")

 except Exception as e:
     print(f"An unexpected error occurred: {e}")
```

8. What percentage of customers rated their health as excellent (HealthScore = 5)?

We add some possible exceptions and if there is a missing value in HealthScore, we plan to fill it with the mean of HealthScore, which is 3.

```
import pandas as pd
try:
    data = pd.read_csv("FitTrackData.csv")
    if 'HealthScore' not in data.columns:
        raise KeyError("The 'HealthScore' column does not exist in the data.")
    if data['HealthScore'].isnull().any():
        data['HealthScore'].fillna(3, inplace=True)
    health_score_five_count = (data['HealthScore'] == 5).sum()
    if len(data) == 0:
        raise ZeroDivisionError("The dataset is empty, cannot perform division."
    health_score_five_percentage = health_score_five_count / len(data) * 100
    print(f"Percentage of users with HealthScore 5: {health_score_five_percentage:.2f}%")

except FileNotFoundError:
    print("Error: The file 'FitTrackData.csv' does not exist.")

except pd.errors.EmptyDataError:
    print("Error: The file 'FitTrackData.csv' is empty.")

except KeyError as e:
    print(f"Error: {e}")

except ZeroDivisionError as e:
    print(f"Error: {e}")

except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

9. **What is the highest annual income in the dataset?**

We add some possible exceptions and if there is a missing value in AnnualIncome, we plan to fill it with the median of AnnualIncome.

```
import pandas as pd
try:
    data = pd.read_csv("FitTrackData.csv")
    if 'AnnualIncome' not in data.columns:
        raise KeyError("The 'AnnualIncome' column does not exist in the data.")
    if data['AnnualIncome'].isnull().any():
        data['AnnualIncome'].fillna(data['AnnualIncome'].median(), inplace=True)
    max_annual_income = data['AnnualIncome'].max()
    print(max_annual_income)

except FileNotFoundError:
    print("Error: The file 'FitTrackData.csv' does not exist.")

except pd.errors.EmptyDataError:
    print("Error: The file 'FitTrackData.csv' is empty.")

except KeyError as e:
    print(f"Error: {e}")

except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

10. **How many customers walk more than 50,000 steps per week (50,000 not included)?**

We add some possible exceptions and if there is a missing value in StepsPerWeek, we plan to fill it with the median of StepsPerWeek.

```
import pandas as pd
try:
    data = pd.read_csv("FitTrackData.csv")
    if 'StepsPerWeek' not in data.columns:
        raise KeyError("The 'StepsPerWeek' column does not exist in the data.")
    if data['StepsPerWeek'].isnull().any():
        data['StepsPerWeek'].fillna(data['StepsPerWeek'].median(), inplace=True)
    steps_over_50000_count = len(data[data['StepsPerWeek'] > 50000])
    print(steps_over_50000_count)

except FileNotFoundError:
    print("Error: The file 'FitTrackData.csv' does not exist.")

except pd.errors.EmptyDataError:
    print("Error: The file 'FitTrackData.csv' is empty.")

except KeyError as e:
    print(f"Error: {e}")

except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

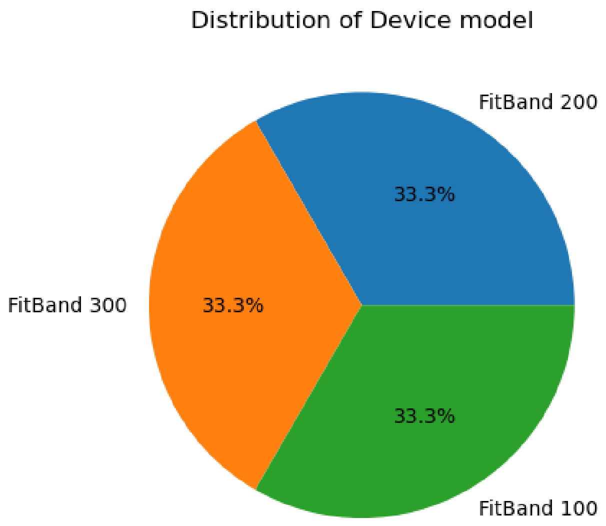
Part 2: Data Visualization (50 Points)

Create the following visualizations to better understand the data:

1. Create a pie chart showing the distribution of device models in the dataset.

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv("FitTrackData.csv")
data.head()
data['DeviceModel'].value_counts().plot(kind='pie', autopct='%1.1f%%')
plt.title("Distribution of Device model")
plt.ylabel("")
plt.show()
```

Output:



Significance tests to test whether there is a significant difference in the frequency of use by users of different device models

```
import pandas as pd
import numpy as np
from scipy.stats import ttest_ind, f_oneway
data = pd.read_csv("FitTrackData.csv")
fitband100_usage_frequency = data[data['DeviceModel'] == 'FitBand 100']['UsageFrequency']
fitband200_usage_frequency = data[data['DeviceModel'] == 'FitBand 200']['UsageFrequency']
fitband300_usage_frequency = data[data['DeviceModel'] == 'FitBand 300']['UsageFrequency']
f_statistic, p_value = f_oneway(fitband100_usage_frequency, fitband200_usage_frequency, fitband300_usage_frequency)
print("F statistic of the effect of the device model on the frequency of use:", f_statistic)
print("The P-value of the effect of the device model on the frequency of use:", p_value)
```

output

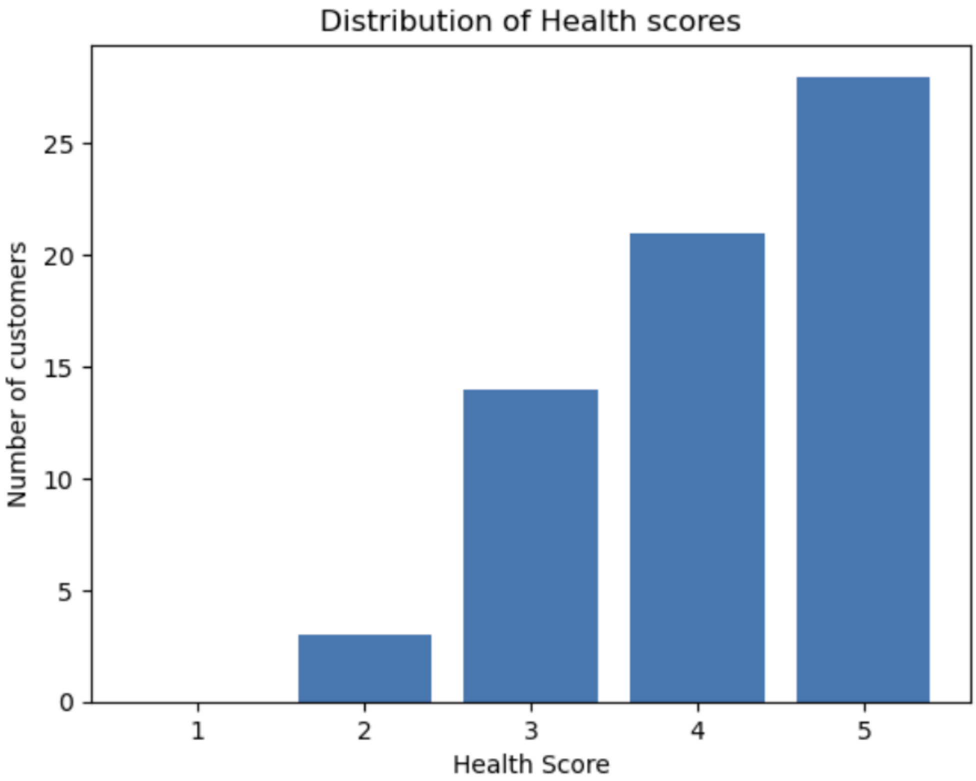
```
F statistic of the effect of the device model on the frequency of use: 1.3892889288928902
The P-value of the effect of the device model on the frequency of use: 0.2567829571737283
```

Because the F statistic is 1.3892889288928902 and the p-value is 0.2567829571737283, we usually set the significance level at 0.05. Since the p-value (0.2567829571737283) is greater than 0.05, we cannot reject the null hypothesis. This indicates that there is not enough evidence to suggest that there is a significant difference in the usage frequency of different device models (such as Fitband 100, Fitband 200, Fitband 300).

2. Plot the distribution of customer health scores (HealthScore).

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
data = pd.read_csv("FitTrackData.csv")
health_score = data['HealthScore']
bins = [0.5, 1.5, 2.5, 3.5, 4.5, 5.5]
grouped_data, _ = np.histogram(health_score, bins=bins)
plt.bar(range(1, 6), grouped_data)
plt.xlabel('Health Score')
plt.ylabel('Number of customers')
plt.title('Distribution of Health scores')
plt.show()
```

Output:



Significance test for Whether the population average health score is greater than 3.

We use the sample in the FitTrackData.csv and ttest_1samp function in the SciPy library. This function performs a one-sample t test to determine whether the sample mean is significantly different from 3.

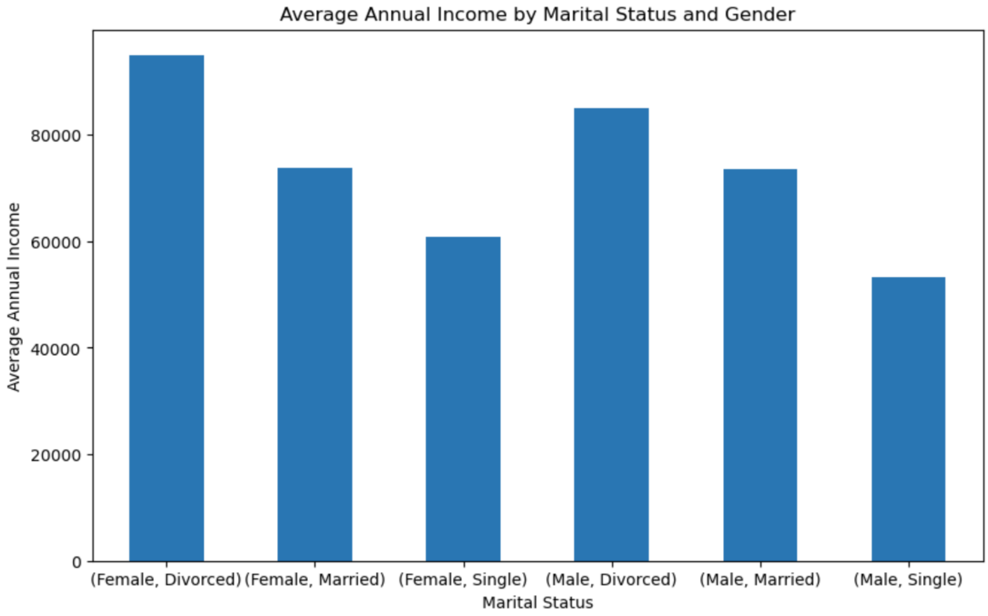
```
import pandas as pd
from scipy import stats
data = pd.read_csv("FitTrackData.csv")
health_score = data['HealthScore']
t_stat, p_value = stats.ttest_1samp(health_score, 3)
print(f"t-statistic: {t_stat:.4f}")
print(f"P-value: {p_value:.4f}")
alpha = 0.05    # Significance level
if p_value < alpha:
    print("The mean health score is significantly greater than 3.")
else:
    print("The mean health score is not significantly greater than 3.")
```

Output:

```
t-statistic: 10.0845
P-value: 0.0000
The mean health score is significantly greater than 3.
```

3. Draw horizontal bar plots of MaritalStatus (on the y-axis) and AnnualIncome (on the x-axis), with gender distinction.

```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(10, 6))
data.groupby(['Gender', 'MaritalStatus'])['AnnualIncome'].mean().plot(kind='bar', ax=ax, rot=0)
ax.set_title('Average Annual Income by Marital Status and Gender')
ax.set_xlabel('Marital Status')
ax.set_ylabel('Average Annual Income')
plt.show()
```



Significance tests to test whether annual income is related to marital status

```
import pandas as pd
import numpy as np
from scipy.stats import f_oneway
data = pd.read_csv("FitTrackData.csv")
single_income = data[data['MaritalStatus'] == 'Single']['AnnualIncome']
married_income = data[data['MaritalStatus'] == 'Married']['AnnualIncome']
divorced_income = data[data['MaritalStatus'] == 'Divorced']['AnnualIncome']
f_statistic, p_value = f_oneway(single_income, married_income, divorced_income)
print("F statistic of the effect of marital status on annual income:", f_statistic)
print("The P-value of the effect of marital status on annual income:", p_value)
```

output

```
F statistic of the effect of marital status on annual income: 30.452594593966047
The P-value of the effect of marital status on annual income: 5.583521730596691e-10
```

The P-value is 5.583521730596691e-10, which is a very small value, far less than the commonly set significance level of 0.05. When the P-value is less than the significance level, the null hypothesis can be rejected and the annual income of different marital status is significantly different.