

Problem 1

Q1. What steps will you take upon receiving this dataset before commencing data analysis?

First, I would take a quick look at the dataset in a spreadsheet app like Excel to understand the structure of the dataset and the type of data it contains. By looking at the columns and the numerical values under the columns, I can get a great idea of what the data is supposed to represent which may help my coding later on.

I would then begin cleaning the dataset. I'd likely either remove missing/corrupted values or replace them with column averages. Then I might look at outliers and process them similarly as well.

Q2. If you are to inquire about Q1 from ChatGPT or Bing, what responses will you receive? Do you find them reasonable? If not, how will you rectify it?

Here is the response I received from ChatGPT:

Before commencing data analysis on the provided dataset, it's essential to perform several preliminary steps to ensure that the data is clean, structured, and ready for analysis. Here are the typical steps you might take:

1. ****Data Loading****: Load the dataset into your Python environment using a suitable library like Pandas.

```
```python
import pandas as pd

Load the dataset
df = pd.read_csv("your_dataset.csv")
```
```

2. ****Exploratory Data Analysis (EDA)****:

- Inspect the first few rows of the dataset using ``head()`` to understand its structure.
- Check for the presence of missing values using ``isnull().sum()``.
- Get a summary of numerical columns using ``describe()``.
- Check for data types of columns using ``info()``.

```

```python
Display the first few rows
print(df.head())

Check for missing values
print(df.isnull().sum())

Summary statistics
print(df.describe())

Data types of columns
print(df.info())
```

```

3. ****Data Cleaning****:

- Handle missing values: You may choose to drop rows with missing values or impute them with suitable values.
- Remove duplicate rows if any.
- Convert data types if necessary (e.g., converting object type to categorical or datetime).

```

```python
Drop rows with missing values
df.dropna(inplace=True)

Remove duplicate rows
df.drop_duplicates(inplace=True)

Convert data types
df['Location'] = df['Location'].astype('category')
Convert other columns as necessary
```

```

4. ****Feature Engineering****:

- Create new features from existing ones if it adds value to the analysis.

```

```python
Example of feature engineering: calculating price per square foot
df['Price_per_sqft'] = df['Sell Price'] / df['Area']
```

```

5. ****Data Visualization****:

- Visualize the distribution of numerical features using histograms, box plots, etc.
- Explore relationships between variables using scatter plots, pair plots, etc.

```

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

Example: Histogram of rent price per month
sns.histplot(df['Rent Price per Month'], bins=20)
plt.title('Histogram of Rent Price per Month')
plt.show()

Example: Scatter plot of area vs. sell price
sns.scatterplot(x='Area', y='Sell Price', data=df)
plt.title('Area vs. Sell Price')
plt.show()
```

```

6. **Handling Outliers**:

- Identify and handle outliers if they exist in the dataset. This could involve removing outliers or applying transformations.

```

```python
Example: Removing outliers using z-score
from scipy.stats import zscore

z_scores = zscore(df['Sell Price'])
abs_z_scores = np.abs(z_scores)
filtered_entries = (abs_z_scores < 3)
df = df[filtered_entries]
```

```

Once these steps are completed, you'll have a clean and structured dataset ready for analysis. You can then proceed with your specific analysis or modeling tasks based on the goals of your project.

The response as a whole looks quite reasonable. I believe the response adequately details how to properly process and clean the dataset before full data analysis. There were three steps ChatGPT suggested that I thought were useful, but not necessary. The first step was feature engineering. Although new features created from the dataset could be helpful down the line, if you don't have specifics on what needs to be analyzed the feature engineering could potentially be unused — therefore rendering it redundant. The other step I thought was potentially redundant was data visualization. Plotting the relationship between variables seems unnecessary unless the relationship has relevance, or the question demands it. The last step I

thought was redundant was the handling of outliers. Although outliers could potentially skew the dataset, I learned from other classes that outliers in large datasets such as this should actually be preserved as they may represent natural variations in the dataset, and should still be considered in order to more accurately assess the dataset.

Q3. If you are restricted to renting a house, which one or ones will you select, and why?

The house I rent has to meet several criteria for it to be eligible for me. 1) The number of rooms has to be more than 3, as I would like to have multiple roommates 2) The house has to be located in a suburb or a city center 3) The rent must be less than or equal to \$16,000 (I'm assuming the data is in NTD) 4) The distance from school has to be less than 50

After meeting these criteria, I sort them by rent price, and choose the house with the lowest rent price to rent. The details of the selected house are displayed in the code.

Q4. Assuming you have enough funds to purchase a house, will you opt to continue renting or proceed with a purchase? If renting, which one will you choose? If buying, which one will you select? Why?

If purchasing a home, there are several criteria the house needs to meet 1) The house has to be located in a suburb or city center 2) The distance between the school and house needs to be less than 50 3) The sell price needs to be less than 50,000,000.

After meeting these criteria, I sort them by sell price, and choose the house with the lowest sell price to buy. The details of the selected house are displayed in the code

In terms of whether I choose to buy or rent the house, it all depends on how long I plan to stay. I read somewhere that if the total sum of rent you pay equals $\leq \frac{1}{4}$ of the purchase price of the property, you may as well buy the property outright as it could become an investment. I did the calculation in python and I determined that I would choose to buy the house if I am planning on staying more than 215.245 months, or about 17.937 years.

Q5. Are there any properties with rent or selling prices that seem unusually high or low? Why?

No, at least not *unusually* high or low.

To calculate properties with "unusually" high/low rent/selling prices, I calculated the mean and standard deviation for rent and selling prices. I then calculated floor and ceiling thresholds for rent and sell by calculating three standard deviations from the mean. After computations were done, it turned out there were no properties which were *statistical* outliers.

I then decided to plot the properties' rent and selling prices to see if any dots (representing the rent and selling prices) looked particularly . After viewing the plots, it seems like the data is almost perfectly distributed and because of this, I believe it was likely generated randomly. As a result of the uniform distribution of data, I would say there are no properties with rent/sell prices that are unusually high/low.

Problem 2

Q1. Which family boasts the highest annual income, and which has the lowest? How do you ascertain this?

We can determine which family has the highest and lowest annual income by first grouping total Spend and Income by their respective families. I can then sort this object by Income, and the first index is the lowest income family while the last income is the highest income family.

Q2. Which families do not possess adequate annual income to cover all members' spending?

What is the maximum shortfall? How do you determine this?

It appears there are no families where there is inadequate income to support spending! I determined this by creating a new dataframe which displays all families where total spending exceeds income, however the dataframe was empty so I guess there were none!

Q3. Are there any single-parent families, where only one Adult is present? Are there any childless families? How do you discern this?

Yes to both. I discerned this by counting the amount of adults in each family. Families with only one adult were filtered, and these were printed as the single-parent families. For childless families, I counted families where there was no "child" string present in their members.

Q4. Do you suspect any errors within this dataset? Examples may include negative figures, missing or duplicate data, etc. Why?

Not that I can think of. I could not find any negative figures or null values. Duplicate data is ok for this dataset as for a family can have the exact same Members, Income, and Spend data and still be feasible. I thought that maybe a logical error could be families with only children, however looking in the dataset there were none. Another potential fallacy was families where children earned income, but none of these existed, so there are no errors as far as I can tell.

Q5. Can ChatGPT or Bing assist with the aforementioned four questions? If so, to what extent? How do you issue commands to the AI tool? If not, why not?

Yes, ChatGPT helped a TON with the previous questions. First I feed the chatbot a small portion of the dataset, as the whole thing would be too long of a message to process. You can usually then straight up ask the given questions to the chatbot, although you need to pay close attention to the outputs as sometimes it gives the wrong answer. If this happens you can either issue more detailed commands explaining the error/misinterpretation, or fix the issue yourself. For example, a logical error I encountered was that there were apparently no childless families according to my code. I discovered that the code was looking for families with members that

started with “child”, however if the family was childless it would contain no strings that started with child. I fixed this issue by explaining it to the chatbot and I was able to find the correct solution.

RAW CODE

```
"""
Assignment: Python Programming Homework 1 Housing
Author: Jason Zhao
Student ID: T12611201
Date created: April 1, 2024
Description: This script demonstrates the usage of a Python file header.
"""

import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv("housing_data.csv")

#%% Question 1 and 2: Data Pre-processing

#%% Exploratory Data Analysis

# Display the first few rows
print(df.head())

# Check for missing values
print(df.isnull().sum())

# Summary statistics
print(df.describe())

# Data types of columns
print(df.info())

print() # new line

#%% Data Cleaning
# Drop rows with missing values
df.dropna(inplace=True)

# Remove duplicate rows
df.drop_duplicates(inplace=True)

print()
```


Question 3: Renting a house

Filtering based on # rooms, location, rent price, and distance to school

```
filtered_rent_df = df[(df['No. of Rooms'] >= 3) &
    ((df['Location'] == 'Suburb') |
    (df['Location'] == 'City Center') ) &
    (df['Rent Price per Month'] <= 16000) &
    (df['Miles (dist. between school and house)'] <= 50)]
```

Ranking the filtered options (for example, based on lowest rent price)

```
ranked_rent_options = filtered_rent_df.sort_values(by='Rent Price per Month')
```

Final selection based on personal preferences

```
selected_rent_house = ranked_rent_options.iloc[0] # Selecting the top-ranked option
```

```
print("Selected House for Renting:")
```

```
print(selected_rent_house)
```

```
print()
```

Question 4: Renting vs Buying

Buying

Filtering based on location, distance to school, and sell price

```
filtered_buy_df = df[((df['Location'] == 'Suburb') |
    (df['Location'] == 'City Center') ) &
    (df['Miles (dist. between school and house)'] <= 50) &
    (df['Sell Price'] <= 50000000)]
```

Ranking the filtered options (for example, based on lowest rent price)

```
ranked_buy_options = filtered_buy_df.sort_values(by='Sell Price')
```

Final selection based on personal preferences

```
selected_buy_house = ranked_buy_options.iloc[0] # Selecting the top-ranked option
```

```
print("Selected House for Purchase:")
```

```
print(selected_buy_house)
```

```
print()
```

When to buy vs rent a house

```

# Rent price of the selected renting house
selected_rent_price = selected_rent_house.at["Rent Price per Month"]

# Sell price of the selected buying house
selected_sell_price = selected_buy_house.at["Sell Price"]

# Time until it becomes more worth it to buy the house instead of renting
time_till_buy = 0.25*(selected_sell_price) / selected_rent_price

print("You should buy the house if the you decide to rent for more than ",
      time_till_buy.round(3), "months, or ",
      (time_till_buy/12).round(3), "years")

### Question 5: Outlier rent/sell prices

### Statistical Outliers

# Calculate summary statistics
mean_rent_price = df["Rent Price per Month"].mean()
median_rent_price = df["Rent Price per Month"].median()
std_rent_price = df["Rent Price per Month"].std()

mean_sell_price = df["Sell Price"].mean()
median_sell_price = df["Sell Price"].median()
std_sell_price = df["Sell Price"].std()

# Define thresholds for identifying outliers (e.g., 3 standard deviations from the mean)
rent_price_ceiling = mean_rent_price + (3 * std_rent_price)
rent_price_floor = mean_rent_price - (3 * std_rent_price)

sell_price_ceiling = mean_sell_price + (3 * std_sell_price)
sell_price_floor = mean_sell_price - (3 * std_sell_price)

# Identify properties with unusually high or low rent or selling prices
high_rent_properties = df[df["Rent Price per Month"] > rent_price_ceiling]
low_rent_properties = df[df["Rent Price per Month"] < rent_price_floor]

high_sell_properties = df[df["Sell Price"] > sell_price_ceiling]
low_sell_properties = df[df["Sell Price"] < sell_price_floor]

# Print the properties with unusually high or low prices
print("Properties with unusually high rent prices:")

```

```

print(high_rent_properties)

print("Properties with unusually low rent prices:")
print(low_rent_properties)

print("Properties with unusually high selling prices:")
print(high_sell_properties)

print("Properties with unusually low selling prices:")
print(low_sell_properties)

print() # New Line

### Plotting rent price

# Plot rent prices as individual points
plt.figure(figsize=(10, 6))
plt.scatter(range(len(df)), df['Rent Price per Month'], color='skyblue', alpha=0.6)
plt.title('Rent Prices')
plt.xlabel('Index')
plt.ylabel('Rent Price per Month')
plt.show()

### Plotting sell price

# Plot sell prices as individual points
plt.figure(figsize=(10, 6))
plt.scatter(range(len(df)), df['Sell Price'], color='lightcoral', alpha=0.6)
plt.title('Purchasing prices')
plt.xlabel('Index')
plt.ylabel('Sell Price per Month')
plt.show()

```

```

"""

```

Assignment: Python Programming Homework 1 Family

Author: Jason Zhao

Student ID: T12611201

Date created: April 1, 2024

Description: This script demonstrates the usage of a Python file header.

```

"""

```

```

import pandas as pd

# Load the dataset
df = pd.read_csv("family_data.csv")

### Question 1: Family with highest/lowest annual income

# Group the data by family and aggregate income and spend
families = df.groupby('Family').agg({'Income': 'sum', 'Spend': 'sum'})

# Reset index to make 'Family' a regular column
families.reset_index(inplace=True)

# Sorting families by income
ranked_incomes = families.sort_values(by='Income')

# Lowest and highest incomes
lowest_income = ranked_incomes.iloc[0]
highest_income = ranked_incomes.iloc[-1]

print('The family with the lowest income is: ')
print(lowest_income, "\n")

print('The family with the highest income is: ')
print(highest_income)

### Question 2: Family with inadequate income to cover spending

broke_families = families[families['Spend'] > families['Income']]

# Check if there are any families where spend exceeds income
if broke_families.empty:
    print("From the given dataset, it appears there are no families" ,
          "where spending exceeds income.")
else:
    print("There are families where spending exceeds income.")

print()

### Question 3: Presence of single-parent and childless families

```

```
##### Single-Parent
```

```
# Group the data by family and count the number of unique adult members
family_adult_counts = df[df['Member'].str.startswith('Adult')].groupby('Family').size()
```

```
# Filter families with only one adult
single_parent_families = family_adult_counts[family_adult_counts == 1]
```

```
if single_parent_families.empty:
    print("There are no single-parent families in the dataset.")
else:
    print("Single-parent families:")
    print(single_parent_families)
```

```
print()
```

```
##### Childless
```

```
# Group the data by family and check if any member starts with 'Child'
childless_families = df.groupby('Family').filter(lambda x: not
any(x['Member'].str.startswith('Child')))
```

```
if childless_families.empty:
    print("There are no childless families in the dataset.")
else:
    print("Childless families:")
    print(childless_families['Family'].unique())
```

```
print()
```

```
##### Question 4: Errors in the dataset
```

```
##### Only child families
child_only_families = df.groupby('Family').filter(lambda x: not
any(x['Member'].str.startswith('Adult')))
```

```
if child_only_families.empty:
    print("There are no families with only children in the dataset.")
else:
    print("Families with only children:")
    print(child_only_families)
```

```
print()
```

```
##### Children earn income
```

```
# Filter the dataset to include only child members
```

```
child_df = df[df['Member'].str.startswith('Child')]
```

```
# Check if any child member has non-zero income
```

```
child_income_families = child_df[child_df['Income'] > 0]['Family'].unique()
```

```
if len(child_income_families) == 0:
```

```
    print("There are no families where children earn income in the dataset.")
```

```
else:
```

```
    print("Families where children earn income:")
```

```
    print(child_income_families)
```