## [Part 1] Task 03. Import your analysis libraries, as well as your new customer data set as a dataframe

In [3]:
```
# Import libraries
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt
import seaborn as sns
import scipy
```

In [7]:
```
# Set path
path = r'C:\Users\Asus\Music\Instacart Basket Analysis'
```

In [24]:
```
# Import new customer data set from the 'Original Data' folder as a dataframe
customer  = pd.read_csv(os.path.join(path, 'Data', 'Original Data', 'customers.csv'))
customer_data.head()
```

Out[24]:

| | user_id | First Name | Surnam | Gender | STATE | Age | date_joined | n_dependants | fam_status | income |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 26711 | Deborah | Esquivel | Female | Missouri | 48 | 1/1/2017 | 3 | married | 165665 |
| 1 | 33890 | Patricia | Hart | Female | New Mexico | 36 | 1/1/2017 | 0 | single | 59285 |
| 2 | 65803 | Kenneth | Farley | Male | Idaho | 35 | 1/1/2017 | 2 | married | 99568 |
| 3 | 125935 | Michelle | Hicks | Female | Iowa | 40 | 1/1/2017 | 0 | single | 42049 |
| 4 | 130797 | Ann | Gilmore | Female | Maryland | 26 | 1/1/2017 | 1 | married | 40374 |

In [26]:
```
# Preview the newly imported dataframe
customers.head(10)
```

Out[26]:

| | user_id | First Name | Surnam | Gender | STATE | Age | date_joined | n_dependants | fam_status | income |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 26711 | Deborah | Esquivel | Female | Missouri | 48 | 1/1/2017 | 3 | married | 165665 |
| 1 | 33890 | Patricia | Hart | Female | New Mexico | 36 | 1/1/2017 | 0 | single | 59285 |
| 2 | 65803 | Kenneth | Farley | Male | Idaho | 35 | 1/1/2017 | 2 | married | 99568 |
| 3 | 125935 | Michelle | Hicks | Female | Iowa | 40 | 1/1/2017 | 0 | single | 42049 |
| 4 | 130797 | Ann | Gilmore | Female | Maryland | 26 | 1/1/2017 | 1 | married | 40374 |
| 5 | 133128 | Cynthia | Noble | Female | Kentucky | 43 | 1/1/2017 | 2 | married | 49643 |
| 6 | 152052 | Chris | Walton | Male | Montana | 20 | 1/1/2017 | 0 | single | 61746 |
| 7 | 168851 | Joseph | Hickman | Male | South Carolina | 30 | 1/1/2017 | 0 | single | 63712 |
| 8 | 69965 | Jeremy | Vang | Male | Texas | 47 | 1/1/2017 | 1 | married | 162432 |
| 9 | 82820 | Shawn | Chung | Male | Virginia | 26 | 1/1/2017 | 2 | married | 32072 |

In [28]:
```
# View descriptive statistics
customers.describe()
```

Out[28]:

| | user_id | Age | n_dependants | income |
|---|---|---|---|---|
| count | 206209.000000 | 206209.000000 | 206209.000000 | 206209.000000 |
| mean | 103105.000000 | 49.501646 | 1.499823 | 94632.852548 |
| std | 59527.555167 | 18.480962 | 1.118433 | 42473.786988 |
| min | 1.000000 | 18.000000 | 0.000000 | 25903.000000 |
| 25% | 51553.000000 | 33.000000 | 0.000000 | 59874.000000 |
| 50% | 103105.000000 | 49.000000 | 1.000000 | 93547.000000 |
| 75% | 154657.000000 | 66.000000 | 3.000000 | 124244.000000 |
| max | 206209.000000 | 81.000000 | 3.000000 | 593901.000000 |

In [30]:
```
# Check data types
customers.dtypes
```

Out[30]:
```
user_id          int64
First Name      object
Surnam          object
Gender          object
STATE           object
Age              int64
date_joined     object
n_dependants     int64
fam_status      object
income           int64
dtype: object
```

## [Part 1] Task 04. Wrangle the data so that it follows consistent logic

Ensure the column names are consistent and descriptive. For example, change Surnam to Surname, fam_status to family_status, n_dependants to Number_Of_Dependants, STATE to State, Date_joined, Date Joined, income to Income etc.

In [33]:
```
# Renaming columns to be consistent and descriptive
customers.rename(columns={
    'First Name': 'First_Name',
    'Surnam': 'Surname',
    'STATE': 'State',
    'date_joined': 'Date_Joined',
    'n_dependants': 'Number_Of_Dependants',
    'fam_status': 'Family_Status',
    'income': 'Income'
}, inplace=True)
# Check the new column names to ensure changes are applied
print(customers.columns)
```

```
Index(['user_id', 'First_Name', 'Surname', 'Gender', 'State', 'Age',
       'Date_Joined', 'Number_Of_Dependants', 'Family_Status', 'Income'],
      dtype='object')
```

## Dropping Unnecessary Columns

In [40]:
```
customers.drop(columns=['Date_Joined'], inplace=True)
```

## [Part 1] Task 05. Complete the fundamental data quality and consistency checks

## Checking for Missing Values:

Identify and handle missing values.

In [44]:
```
customers.isnull().sum()
# If there are missing values, decide how to handle them, e.g., drop or fill with a placeholder.
```

Out[44]:
```
user_id                 0
First_Name          11259
Surname                 0
Gender                  0
State                   0
Age                     0
Number_Of_Dependants    0
Family_Status           0
Income                  0
dtype: int64
```

## Checking for Duplicates:

Identify and remove duplicate rows if any.

In [47]:
```
customers.duplicated().sum()
# Remove duplicates if any
customers.drop_duplicates(inplace=True)
```

## Converting Data Types:

In [54]:
```
# The output revealed that the 'first_name' column has mixed data-types that need to be addressed
customers['First_Name'] = customers['First_Name'].astype('str')
```

In [56]:
```
# The 'user_id' column, while not containing mixed data-types, is still improperly listed as an integer
customers['user_id'] = customers['user_id'].astype('str')
```

In [62]:
```
customers['Income'] = customers['Income'].astype(int)
```

In [64]:
```
customers.head(10)
```

Out[64]:

| | user_id | First_Name | Surname | Gender | State | Age | Number_Of_Dependants | Family_Status | Income |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 26711 | Deborah | Esquivel | Female | Missouri | 48 | 3 | married | 165665 |
| 1 | 33890 | Patricia | Hart | Female | New Mexico | 36 | 0 | single | 59285 |
| 2 | 65803 | Kenneth | Farley | Male | Idaho | 35 | 2 | married | 99568 |
| 3 | 125935 | Michelle | Hicks | Female | Iowa | 40 | 0 | single | 42049 |
| 4 | 130797 | Ann | Gilmore | Female | Maryland | 26 | 1 | married | 40374 |
| 5 | 133128 | Cynthia | Noble | Female | Kentucky | 43 | 2 | married | 49643 |
| 6 | 152052 | Chris | Walton | Male | Montana | 20 | 0 | single | 61746 |
| 7 | 168851 | Joseph | Hickman | Male | South Carolina | 30 | 0 | single | 63712 |
| 8 | 69965 | Jeremy | Vang | Male | Texas | 47 | 1 | married | 162432 |
| 9 | 82820 | Shawn | Chung | Male | Virginia | 26 | 2 | married | 32072 |

## [Part 1] Task 06. Combine your customer data with the rest of your prepared Instacart data

In [69]:
```
# Import the rest of the prepared Instacart data
orders_products = pd.read_pickle(os.path.join(path, 'Data', 'Prepared Data', 'ords_prods_merge_final.pkl'))
```

In [71]:
```
# Check the shape of the newly imported dataframe
orders_products.shape
```

Out[71]:
```
(32434212, 26)
```

In [73]:
```
# Check the shape of the customers dataframe
customers.shape
```

Out[73]:
```
(206209, 9)
```

In [75]:
```
# Identify a shared column
orders_products.head(5)
```

Out[75]:

| | Unnamed: 0_x | order_id | user_id | eval_set | order_number | order_dow | order_hour_of_day | days_since_prior_order | product_id | add_to_cart_order | ... | busiest_period_of_day | price_label | busiest_day | busiest_days | loyalty_flag | order_count | spending_flag | spending_category | order_frequency_flag |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 2539329 | 1 | prior | 1 | 2 | 8 | NaN | 196 | 1 | ... | Average orders | Medium | Other day | Regular days | New Customer | 10 | 6.367797 | Low spender | 20.5 |
| 1 | 0 | 2539329 | 1 | prior | 1 | 2 | 8 | NaN | 14084 | 2 | ... | Average orders | High | Other day | Regular days | New Customer | 10 | 6.367797 | Low spender | 20.5 |
| 2 | 0 | 2539329 | 1 | prior | 1 | 2 | 8 | NaN | 12427 | 3 | ... | Average orders | Low | Other day | Regular days | New Customer | 10 | 6.367797 | Low spender | 20.5 |
| 3 | 0 | 2539329 | 1 | prior | 1 | 2 | 8 | NaN | 26088 | 4 | ... | Average orders | Low | Other day | Regular days | New Customer | 10 | 6.367797 | Low spender | 20.5 |
| 4 | 0 | 2539329 | 1 | prior | 1 | 2 | 8 | NaN | 26405 | 5 | ... | Average orders | Low | Other day | Regular days | New Customer | 10 | 6.367797 | Low spender | 20.5 |

5 rows × 26 columns

In [77]:
```
customers.head(5)
```

Out[77]:

| | user_id | First_Name | Surname | Gender | State | Age | Number_Of_Dependants | Family_Status | Income |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 26711 | Deborah | Esquivel | Female | Missouri | 48 | 3 | married | 165665 |
| 1 | 33890 | Patricia | Hart | Female | New Mexico | 36 | 0 | single | 59285 |
| 2 | 65803 | Kenneth | Farley | Male | Idaho | 35 | 2 | married | 99568 |
| 3 | 125935 | Michelle | Hicks | Female | Iowa | 40 | 0 | single | 42049 |
| 4 | 130797 | Ann | Gilmore | Female | Maryland | 26 | 1 | married | 40374 |

In [ ]:
```
# Ensure the data types of key columns match
customers['user_id'] = customers['user_id'].astype(int)
orders_products['user_id'] = orders_products['user_id'].astype(int)

# Merge the datasets on 'user_id'
df_combined = pd.merge(customers, orders_products, on='user_id', how='left')
```

## Step 7: Notebook Structure and Comments

Customer Data Wrangling and Quality Checks

In [86]:
```
# Checking for missing values in the customer dataset
customers.isnull().sum()
```

Out[86]:
```
user_id                 0
First_Name              0
Surname                 0
Gender                  0
State                   0
Age                     0
Number_Of_Dependants    0
Family_Status           0
Income                  0
dtype: int64
```

## [Part 1] Task 08. Export this new dataframe as a pickle file so you can continue to use it in the second part of this task

In [93]:
```
customers.to_pickle(os.path.join(path, 'Data','Prepared Data', 'combined_data.pkl'))
```