Note

- Instructions have been included for each segment. You do not have to follow them exactly, but they are included to help you think through the steps.

In [170]:

```
# Dependencies and Setup
import pandas as pd

# File to Load (Remember to Change These)
file_to_load = "Resources/purchase_data.csv"

# Read Purchasing File and store into Pandas data frame
purchase_data = pd.read_csv(file_to_load)
```

In [171]:

```
purchase_data.head()
purchase_data["Age"].max()
```

Out[171]:

```
45
```

Player Count

- Display the total number of players

In [172]:

```
# Get count of unique players

unique = purchase_data["SN"].unique()
unique_count = len(unique)
unique_count

# Display as df

unique_df = pd.DataFrame({"Total Players": [unique_count]})
unique_df
```

Out[172]:

	Total Players
0	576

Purchasing Analysis (Total)

- Run basic calculations to obtain number of unique items, average price, etc.
- Create a summary data frame to hold the results
- Optional: give the displayed data cleaner formatting
- Display the summary data frame

In [173]:

```
# Run basic calculations

unique_items = len(purchase_data["Item ID"].unique())
average_price = purchase_data["Price"].mean()
number_purchases = purchase_data["Purchase ID"].count()
total_revenue = purchase_data["Price"].sum()

# Create summary df to hold results

purchasing_analysis_df = pd.DataFrame({"Number of Unique Items": [unique_it
                                       "Average Price": [average_price],
                                       "Number of Purchases": [number_purch
                                       "Total Revenue": [total_revenue]
                                      }
                                     )

# Clean formatting of data and display df

purchasing_analysis_df["Average Price"] = purchasing_analysis_df["Average P
purchasing_analysis_df["Total Revenue"] = purchasing_analysis_df["Total Rev
purchasing_analysis_df
```

Out[173]:

	Number of Unique Items	Average Price	Number of Purchases	Total Revenue
0	179	$3.05	780	$2379.77

Gender Demographics

- Percentage and Count of Male Players
- Percentage and Count of Female Players
- Percentage and Count of Other / Non-Disclosed

In [174]:

```
# Group data by gender and count unique SN values

gender_group = purchase_data.groupby("Gender")["SN"].nunique()

# Hold this data in a dataframe and sort from greatest to least

gender_df = pd.DataFrame(gender_group)
gender_df = gender_df.sort_values("SN", ascending = False)

# Rename column

gender_df = gender_df.rename(columns={"SN":"Total Count"})

# Calculate percentages based off of total count data

gender_df["Percentage of Players"] = ((gender_df["Total Count"])/(gender_df

# Clean formatting of data and display df

gender_df["Percentage of Players"] = gender_df["Percentage of Players"].map
gender_df
```

Out[174]:

	Total Count	Percentage of Players
Gender		
Male	484	84.03%
Female	81	14.06%
Other / Non-Disclosed	11	1.91%

Purchasing Analysis (Gender)

- Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase total per person etc. by gender
- Create a summary data frame to hold the results
- Optional: give the displayed data cleaner formatting
- Display the summary data frame

In [175]:

```
# Group data by gender

gender_group2 = purchase_data.groupby("Gender")

# Get count of purchases by gender and make df

gender_purchase = gender_group2["SN"].count()
gender_group2_df = pd.DataFrame(gender_purchase)

# Rename column

gender_group2_df = gender_group2_df.rename(columns={"SN":"Purchase Count"})

# Calculate other required values and add them as columns in the df

gender_group2_df["Average Purchase Price"] = gender_group2["Price"].mean()
gender_group2_df["Total Purchase Value"] = gender_group2_df["Average Purcha
gender_group2_df["Avg Total Purchase per Person"] = gender_group2_df["Total

# Clean the formatting of the data and display summary df

gender_group2_df["Average Purchase Price"] = gender_group2_df["Average Purc
gender_group2_df["Total Purchase Value"] = gender_group2_df["Total Purchase
gender_group2_df["Avg Total Purchase per Person"] = gender_group2_df["Avg T
gender_group2_df
```

Out[175]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
Gender				
Female	113	$3.20	$361.94	$4.47
Male	652	$3.02	$1,967.64	$4.07
Other / Non-Disclosed	15	$3.35	$50.19	$4.56

Age Demographics

- Establish bins for ages
- Categorize the existing players using the age bins. Hint: use pd.cut()
- Calculate the numbers and percentages by age group
- Create a summary data frame to hold the results
- Optional: round the percentage column to two decimal points

- Display Age Demographics Table

In [176]:

```
# Establish bins/group labels

bins = [0, 9, 14, 19, 24, 29,
        34, 39, 45]
group_labels = ["<10", "10-14","15-19","20-24","25-29","30-34","35-39","40+

# Add binned data to df

purchase_data["Age Group"] = pd.cut(purchase_data["Age"], bins, labels=grou
purchase_data

# Group by Age Group and unique SN and create a new df

age_group = purchase_data.groupby("Age Group")["SN"].nunique()
age_df = pd.DataFrame(age_group)

# Rename df column and index

age_df = age_df.rename(columns={"SN":"Total Count"})
age_df.index.names = ["Age Ranges"]

# Calculate percentages based off of total count data

age_df["Percentage of Players"] = ((age_df["Total Count"]) / (gender_df["To

# Clean the formatting of the data and display summary df

age_df["Percentage of Players"] = age_df["Percentage of Players"].map("{:,.
age_df
```

Out[176]:

	Total Count	Percentage of Players
Age Ranges		
<10	17	2.95%
10-14	22	3.82%
15-19	107	18.58%
20-24	258	44.79%
25-29	77	13.37%
30-34	52	9.03%
35-39	31	5.38%
40+	12	2.08%

Purchasing Analysis (Age)

- Bin the purchase_data data frame by age
- Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase total per person etc. in the table below
- Create a summary data frame to hold the results
- Optional: give the displayed data cleaner formatting
- Display the summary data frame

In [177]:

```
# Using df with binned data, group by Age Group

age_group2 = purchase_data.groupby("Age Group")

# Retrieve purchase counts by age

age2_purchase = age_group2["SN"].count()
age2_df = pd.DataFrame(age2_purchase)

# Rename column and index

age2_df = age2_df.rename(columns={"SN":"Purchase Count"})
age2_df.index.names = ["Age Ranges"]
age2_df

# Calculate other required values and add them as columns in the df

age2_df["Average Purchase Price"] = age_group2["Price"].mean()
age2_df["Total Purchase Value"] = age2_df["Average Purchase Price"] * age2_
age2_df["Avg Total Purchase per Person"] = age2_df["Total Purchase Value"]

# Clean the formatting of the data and display summary df

age2_df["Average Purchase Price"] = age2_df["Average Purchase Price"].map("
age2_df["Total Purchase Value"] = age2_df["Total Purchase Value"].map("${:,
age2_df["Avg Total Purchase per Person"] = age2_df["Avg Total Purchase per
age2_df
```

Out[177]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
Age Ranges				
<10	23	$3.35	$77.13	$4.54
10-14	28	$2.96	$82.78	$3.76
15-19	136	$3.04	$412.89	$3.86
20-24	365	$3.05	$1,114.06	$4.32
25-29	101	$2.90	$293.00	$3.81
30-34	73	$2.93	$214.00	$4.12
35-39	41	$3.60	$147.67	$4.76
40+	13	$2.94	$38.24	$3.19

Top Spenders

- Run basic calculations to obtain the results in the table below
- Create a summary data frame to hold the results

- Sort the total purchase value column in descending order
- Optional: give the displayed data cleaner formatting
- Display a preview of the summary data frame

In [209]:

```
# Group by SN

sn_group = purchase_data.groupby("SN")

# Retrieve Purchase Count and convert to df

sn_purchase_count = sn_group["Purchase ID"].count()
sn_df = pd.DataFrame(sn_purchase_count)

# Rename column

sn_df = sn_df.rename(columns={"Purchase ID":"Purchase Count"})

# Calculate other required values and add them as columns in the df

sn_df["Average Purchase Price"] = sn_group["Price"].mean()
sn_df["Total Purchase Value"] = sn_df["Average Purchase Price"] * sn_df["Pu

# Sort by Total Purchase Value

sn_df = sn_df.sort_values("Total Purchase Value", ascending = False)

# Clean the formatting of the data and display summary df

sn_df["Average Purchase Price"] = sn_df["Average Purchase Price"].map("${:.
sn_df["Total Purchase Value"] = sn_df["Total Purchase Value"].map("${:.2f}"
sn_df.head()
```

Out[209]:

	Purchase Count	Average Purchase Price	Total Purchase Value
SN			
Lisosia93	5	$3.79	$18.96
Idastidru52	4	$3.86	$15.45
Chamjask73	3	$4.61	$13.83
Iral74	4	$3.40	$13.62
Iskadarya95	3	$4.37	$13.10

Most Popular Items

- Retrieve the Item ID, Item Name, and Item Price columns

- Group by Item ID and Item Name. Perform calculations to obtain purchase count, average item price, and total purchase value
- Create a summary data frame to hold the results
- Sort the purchase count column in descending order
- Optional: give the displayed data cleaner formatting
- Display a preview of the summary data frame

In [239]:

```
# Retrieve Item ID, Item Name, Item Price Columns

popular = purchase_data[["Item ID","Item Name","Price"]]

# Group by Item ID and Item Name

popular_group = popular.groupby(["Item ID","Item Name"])

# Obtain purchase count and store in df

popular_purchase_count = popular_group["Price"].count()
popular_df = pd.DataFrame(popular_purchase_count)

# Rename column

popular_df = popular_df.rename(columns={"Price":"Purchase Count"})

# Calculate other required values and add them as columns in the df

popular_df["Item Price"] = popular_group["Price"].mean()
popular_df["Total Purchase Value"] = popular_df["Purchase Count"] * popular

# Sort by Purchase Count

popular_df = popular_df.sort_values("Purchase Count", ascending = False)

# Clean the formatting of the data and display summary df

popular_df["Item Price"] = popular_df["Item Price"].map("${:.2f}".format)
popular_df["Total Purchase Value"] = popular_df["Total Purchase Value"].map
popular_df.head()
```

Out[239]:

		Purchase Count	Item Price	Total Purchase Value
Item ID	**Item Name**			
92	**Final Critic**	13	$4.61	$59.99
178	**Oathbreaker, Last Hope of the Breaking Storm**	12	$4.23	$50.76
145	**Fiery Glass Crusader**	9	$4.58	$41.22
132	**Persuasion**	9	$3.22	$28.99
108	**Extraction, Quickblade Of Trembling Hands**	9	$3.53	$31.77

Most Profitable Items

- Sort the above table by total purchase value in descending order
- Optional: give the displayed data cleaner formatting

- Display a preview of the data frame

In [242]:

```
# Re-running the entire code from the previous part to clear formatting for

# Data is sorted by Total Purchase Value.

popular = purchase_data[["Item ID","Item Name","Price"]]

popular_group = popular.groupby(["Item ID","Item Name"])
popular_purchase_count = popular_group["Price"].count()
popular_df = pd.DataFrame(popular_purchase_count)

popular_df = popular_df.rename(columns={"Price":"Purchase Count"})
popular_df["Item Price"] = popular_group["Price"].mean()
popular_df["Total Purchase Value"] = popular_df["Purchase Count"] * popular

popular_df = popular_df.sort_values("Total Purchase Value", ascending = Fal

popular_df["Item Price"] = popular_df["Item Price"].map("${:.2f}".format)
popular_df["Total Purchase Value"] = popular_df["Total Purchase Value"].map
popular_df.head()
```

Out[242]:

		Purchase Count	Item Price	Total Purchase Value
Item ID	**Item Name**			
92	**Final Critic**	13	$4.61	$59.99
178	**Oathbreaker, Last Hope of the Breaking Storm**	12	$4.23	$50.76
82	**Nirvana**	9	$4.90	$44.10
145	**Fiery Glass Crusader**	9	$4.58	$41.22
103	**Singed Scalpel**	8	$4.35	$34.80

In [ ]: