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 [1]:
```
# 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
df = pd.read_csv(file_to_load)
```

Player Count

- Display the total number of players

In [2]:
```
df.columns
```

Out[2]:
```
Index(['Purchase ID', 'SN', 'Age', 'Gender', 'Item ID', 'Item Name',
'Price'], dtype='object')
```

In [3]:
```
total_players = len(df['SN'].value_counts())
total_players

Summary_players = pd.DataFrame({"Total Players": [total_players]})
Summary_players
```

Out[3]:

	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 [4]:
```
unique_items = len(df['Item ID'].value_counts())
unique_items

average_price = df['Price'].mean()
average_price =  round(average_price,2)
average_price

number_of_purchases = len(df)
number_of_purchases

Total_revenue = df['Price'].sum()
Total_revenue

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

Summary_purchases["Average Price"] = Summary_purchases["Average Price"
Summary_purchases["Total Revenue"] =Summary_purchases["Total Revenue"]

Summary_purchases
```

Out[4]:

	Number of Unique Items	Average Price	Number of Purchases	Total Revenue
0	179	$3.05	780	$2,379.77

Gender Demographics

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

In [5]:
```
# Group by gender
gender_groups = df.groupby(['Gender'])
genders = gender_groups.nunique()

#Total Gender
total_gender = genders["SN"].sum()

# Count and percentage of players
Count_players = genders["SN"].unique()
Percentage_players= genders["SN"]/ total_gender

#Summary Table
Gender_demographics = pd.DataFrame({"Percentage of Players": Percentag
                                    "Players Count":Count_players})

# Formatting the percentage of players
Gender_demographics["Percentage of Players"] = Gender_demographics["Pe
Gender_demographics
```

Out[5]:

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

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 [6]:
```
# Purchase Analysis

purchase_count = gender_groups['Purchase ID'].count()


Average_purchase = gender_groups['Price'].mean()

Total_purchase = gender_groups['Price'].sum()

Avg_Total = Total_purchase / Count_players


Purchase_Analysis = pd.DataFrame({"Purhcase Count": purchase_count,
                                  "Average Purchase Price": Average_pu
                                  "Total Purchase Price":Total_purchas
                                  "Average Total Purchase": Avg_Total}

# Formatting the columns

Purchase_Analysis["Average Purchase Price"] = Purchase_Analysis["Avera
Purchase_Analysis["Total Purchase Price"] = Purchase_Analysis["Total P
Purchase_Analysis["Average Total Purchase"] = Purchase_Analysis["Avera

# Naming the first column
Purchase_Analysis.index.name = "Gender"


Purchase_Analysis.head()
```

Out[6]:

Gender	Purhcase Count	Average Purchase Price	Total Purchase Price	Average Total Purchase
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 [7]:
```
# Age Demographics

# Establishing bins for ages and creating labels
age_bins = [0,9,14,19,24,29,34,39,100]
age_labels = ["<10","10-14", "15-19","20-24","25-29","30-34","35-39","

#categorized the existing players by age group
df["Age Demographics"]= pd.cut(df["Age"], age_bins, labels=age_labels)

#New dataframe
age_demographics = df.groupby("Age Demographics")

#Total age count of players by age
total_age = age_demographics["SN"].nunique()

# calculating the age percentage
age_percent = total_age/total_players


# Summary data frame to hold the values
age_demographics_analysis = pd.DataFrame({"Total Count": total_age,
                                "Percentage of Players":age_percent})

age_demographics_analysis["Percentage of Players"] = age_demographics_

age_demographics_analysis
```

Out[7]:

Age Demographics	Total Count	Percentage of Players
<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 [8]:
```
# Purchase Analysis on the basis of Age



#Count purchase by age
Purchase_count =age_demographics['Purchase ID'].count()

# Average purchase price by age demographics
avg_purchase_age = age_demographics['Price'].mean()

# Total Purchase value by age_demographics
total_purchase_value = age_demographics['Price'].sum()

# Average purchase per person by age
avg_purchase_per_person = total_purchase_value/total_age

Age_Purchase_Analysis = pd.DataFrame({"Purchase Count": Purchase_count
                                      "Average Purchase Price": avg_pur
                                      "Total Purchase Value":total_purc
                                       "Average Purchase Total per Pers

Age_Purchase_Analysis["Average Purchase Price"] = Age_Purchase_Analysi
Age_Purchase_Analysis["Total Purchase Value"] = Age_Purchase_Analysis[
Age_Purchase_Analysis[ "Average Purchase Total per Person"] = Age_Purc


Age_Purchase_Analysis
```

Out[8]:

Age Demographics	Purchase Count	Average Purchase Price	Total Purchase Value	Average Purchase Total per Person

<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 [9]:
```
# Top Spenders Analyis

Spender_data = df.groupby('SN')

Count_spender_data = Spender_data['Purchase ID'].count()

# average purchase by SN
average_purchase_price = Spender_data['Price'].mean()

# Sum of purchase total
purchase_total_spender = Spender_data['Price'].sum()

# Summary Data
Top_Spender = pd.DataFrame({"Purchase Count": Count_spender_data,
                            "Average Purchase Price": average_purchase_
                             "Total Purchase Value": purchase_total_spe
# Formatting the columnns & Sort the table
Top_Spender_Formatted = Top_Spender.sort_values(["Total Purchase Value
Top_Spender_Formatted["Average Purchase Price"] = Top_Spender_Formatte
Top_Spender_Formatted["Total Purchase Value"] = Top_Spender_Formatted[


Top_Spender_Formatted.head()
```

Out[9]:

SN	Purchase Count	Average Purchase Price	Total Purchase Value
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 [10]:
```
# Most Popular Item

items_id = df[['Item ID', 'Item Name', 'Price']]

# Grouped by Item Name & ID
items_data = items_id.groupby(['Item ID','Item Name'])

#count the items
purchase_items = items_data['Price'].count()

# Counting items per item
purchase_item_count = items_data['Price'].sum()

#Total items
total_purchase_items = purchase_item_count/purchase_items

# Creating the summary table
Popular_item = pd.DataFrame({"Purchase Count": purchase_items,
                             "Item Price":total_purchase_items,
                             "Total Purchase Items": purchase_item_coun

Popular_item_formatted = Popular_item.sort_values(["Purchase Count"],a

Popular_item_formatted["Item Price"] = Popular_item_formatted["Item Pr
Popular_item_formatted["Total Purchase Items"] = Popular_item_formatte



Popular_item_formatted
```

Out[10]:

Item ID	Item Name	Purchase Count	Item Price	Total Purchase Items
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 [11]:
```
# Most Profitable Items

Popular_item_formatted = Popular_item.sort_values(["Purchase Count"],a

Popular_item_formatted["Item Price"] = Popular_item_formatted["Item Pr
Popular_item_formatted["Total Purchase Items"] = Popular_item_formatte



Popular_item_formatted.head()
```

Out[11]:

Item ID	Item Name	Purchase Count	Item Price	Total Purchase Items
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

In [13]:
```
# Observations For Heroes Of Pymoli
# 1. The gender demographics reveals that the number of male players i
# 2. The purchases analysis reveals that the male players play an inte
# 3. The age-group between 20-24 have the highest purchase count compa
# 4. 'Final Critic' is the popular game in the list
```