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
purchase_data = pd.read_csv(file_to_load)
```

In [2]:
```
# Analyse the Data Set

purchase_data.head()

# View Data Types for each columns

purchase_data.dtypes
#purchase_data.describe
```

Out[2]:
```
Purchase ID      int64
SN              object
Age              int64
Gender          object
Item ID          int64
Item Name       object
Price          float64
dtype: object
```

Player Count

- Display the total number of players

In [3]:
```
# count the total number of unique players by SN
total_players=purchase_data["SN"].nunique()

# Dataframe using dictorary to show the Player Count
player_count_df=pd.DataFrame({
                              "Total Players":[total_players]
                              })
player_count_df
```

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]:
```
# Defining Variable for Calculation
unique_items = purchase_data["Item ID"].nunique()
avg_price = round(purchase_data["Price"].mean(),2)
num_purchases = purchase_data["Purchase ID"].nunique()
total_revenue =purchase_data["Price"].sum()

# Creating Dataframe
summary_df=pd.DataFrame({"Number of Unique Items":[unique_items],
              "Average Price":["$"+str(avg_price)],
              "Number of Purchases":[num_purchases],
              "Total Revenue":["$"+'{0:,.2f}'.format(total_revenue)]})
summary_df
```

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]:
```
# New Dataframe with unique values
purchase_data_unique = purchase_data.drop_duplicates(["SN"])

# Caluculating Percentage and Count
gender_count= purchase_data_unique["Gender"].value_counts()
gender_percentage = round((gender_count/total_players)*100,2).astype(str) + '%'

# Creating Dataframe with count and percentage
demographics = gender_count.to_frame("Total Count")
demographics["Percentage of Players"] = gender_percentage
demographics
```

Out[5]:

	Total Count	Percentage of Players
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 [6]:
```
# Defining Variable with  Calculation
#groupby_gender= purchase_data.groupby(["Gender"])
purchase_count = purchase_data.groupby("Gender")["Purchase ID"].count()
purchase_count_per_person =purchase_data_unique.groupby("Gender")["Purchase ID"].count(
avg_purchase_value = round(purchase_data.groupby("Gender")["Price"].mean(),2).map("${:.
total_purchase_value = round(purchase_data.groupby("Gender")["Price"].sum(),2)#.map("${
avg_total_purchase_per_person = round(total_purchase_value /purchase_count_per_person,2


#Summary Table for Purchasing Analysis by Gender

summary_purchasing_analysis=pd.concat([purchase_count,avg_purchase_value,
                                       total_purchase_value,avg_total_purchase_per_pers
summary_purchasing_analysis.columns = ["Purchase Count","Average Purchase Price",
                                       "Total Purchase Value","Avg Total Purchase per P
summary_purchasing_analysis["Total Purchase Value"] = summary_purchasing_analysis["Tota
summary_purchasing_analysis
```

Out[6]:

Gender	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
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]:
```
#Creating age bins
age_bin = [0, 9, 14, 19, 24, 29, 34, 39, 100]

#creating bin labels
age_ranges_label =["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]

#adding bin to the original dataframe
purchase_data["Age Ranges"] =pd.cut(purchase_data["Age"], age_bin, labels=age_ranges_la

#calculating count and percentage
total_count = purchase_data.groupby(["Age Ranges"]).nunique()["SN"]
percentage = round(total_count/total_players *100,2).map("{:.2f}%".format)

#creating new summaryDataFrame

age_demographics_summary = pd.DataFrame({"Total Count": total_count,
                                         "Percentage of Players":percentage})
age_demographics_summary.index.name = None
age_demographics_summary
```

Out[7]:

	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]:
```
# variables and calculations for purchase analysis
purchase_count = purchase_data.groupby(["Age Ranges"]).count()["Price"]
avg_purchase = purchase_data.groupby(["Age Ranges"]).mean()["Price"]
total_purchase = purchase_data.groupby(["Age Ranges"]).sum()["Price"]
avg_purchase_per_person = total_purchase / purchase_data.groupby(["Age Ranges"]).uniqu
```

```
# create new dataframe using dictonary to view the summary
purchasing_analysis_summary = pd.DataFrame({"Purchase Count": purchase_count,
                                "Average Purchase Price": avg_purchase.map("${:.2f}".for
                               "Total Purchase Value": total_purchase.map("${:.2f}".form
                               "Avg Total Purchase per Person": avg_purchase_per_person.

purchasing_analysis_summary
```

Out[8]:

Age Ranges	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase 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	$1114.06	$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]:

```
# variables and calculations for total spender analysis
purchase_count = purchase_data.groupby(["SN"]).count()["Price"]
avg_purchase_price = purchase_data.groupby(["SN"]).mean()["Price"]
total_purchase_value = purchase_data.groupby(["SN"]).sum()["Price"]

# create new dataframe using dictonary to view the summary
total_spender_summary = pd.DataFrame({"Purchase Count": purchase_count,
                                "Average Purchase Price": avg_purchase_price.map("
                                "Total Purchase Value": total_purchase_value})

total_spender_summary.sort_values("Total Purchase Value", ascending=False).head()
```

Out[9]:

	Purchase Count	Average Purchase Price	Total Purchase Value

SN	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 [10]:
```
# variables and calculations for most popular item
item_count = purchase_data.groupby(["Item ID","Item Name"]).count()["Price"]
item_total = purchase_data.groupby(["Item ID","Item Name"]).sum()["Price"]
item_average = purchase_data.groupby(["Item ID","Item Name"]).mean()["Price"]

# create new dataframe using dictonary to view the summary
item_data = pd.DataFrame({"Purchase Count": item_count,
                          "Item Price": item_average.map("${:.2f}".format),
                          "Total Purchase Value": item_total.map("${:.2f}".format)})
 # Sort by most popular item in descending order
item_data.sort_values("Purchase Count", ascending=False).head()
```

Out[10]:

Item ID	Item Name	Purchase Count	Item Price	Total Purchase Value
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

Item ID	Item Name	Purchase Count	Item Price	Total Purchase Value
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]:
```
 # Sort by most profitable item by total purchase value in descending order
item_data.sort_values("Total Purchase Value", ascending=False).head()
```

Out[11]:

Item ID	Item Name	Purchase Count	Item Price	Total Purchase Value
63	**Stormfury Mace**	2	$4.99	$9.98
29	**Chaos, Ender of the End**	5	$1.98	$9.90
173	**Stormfury Longsword**	2	$4.93	$9.86
38	**The Void, Vengeance of Dark Magic**	4	$2.37	$9.48
143	**Frenzied Scimitar**	6	$1.56	$9.36