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 [3]:

```
# Dependencies and Setup
import pandas as pd

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

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

Player Count

- Display the total number of players

In [4]:

```
total_players = len(purchase_data["SN"].value_counts())
playerCT = pd.DataFrame({"Total Players":[total_players]})
playerCT
```

Out[4]:

	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 [5]:

```
N_unique_it = len((purchase_data["Item Name"]).unique())
avg_price = (purchase_data["Price"]).mean()
NofPurchases = (purchase_data["Purchase ID"]).count()
Trevenue = (purchase_data["Price"]).sum()
report_df = pd.DataFrame({"Number of Unique Items":[N_unique_it],
                        "Average Price":[avg_price],
                         "Number of Purchases":[NofPurchases],
                         "Total Revenue":[Trevenue]})
report_df["Average Price"] = report_df["Average Price"].map("${:,.2f}".format)
report_df["Total Revenue"] = report_df["Total Revenue"].map("${:,.2f}".format)
report_df
```

Out[5]:

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

In [ ]:

```

```

Gender Demographics

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

In [6]:

```
gender_report = purchase_data.groupby("Gender")
total_gender = gender_report.nunique()["SN"]
PercOfPlayers = total_gender / total_players * 100
genderNumbers_df = pd.DataFrame({"Total Count": total_gender, "Percentage of Players": PercOfPlayers})
genderNumbers_df.index.name = None
genderNumbers_df["Percentage of Players"] = genderNumbers_df["Percentage of Players"].map("{:.2f}%".format)
genderNumbers_df
```

Out[6]:

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

In [ ]:

```

```

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 [7]:

```
purchaseCt = gender_report["Purchase ID"].count()
avg_price_gender = gender_report["Price"].mean()
purchase_total = gender_report["Price"].sum()
avg_purchase_person = purchase_total/total_gender
gender_demog = pd.DataFrame({"Purchase Count": purchaseCt,
                            "Average Purchase Price": avg_price_gender,
                            "Total Purchase Value": purchase_total,
                            "Avg Total Purchase per Person": avg_purchase_person})
gender_demog.index.name = "Gender"
gender_demog["Average Purchase Price"] = gender_demog["Average Purchase Price"].map("${:,.2f}".format)
gender_demog["Total Purchase Value"] = gender_demog["Total Purchase Value"].map("${:,.2f}".format)
gender_demog["Avg Total Purchase per Person"] = gender_demog["Avg Total Purchase per Person"].map("${:,.2f}".format)

gender_demog
```

Out[7]:

	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 [10]:

```
ageBins = [0, 9.9, 14.9, 19.9, 24.9, 29.9, 34.9, 39.9, 99999]
groupNames = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]
purchase_data["Age Group"] = pd.cut(purchase_data["Age"],ageBins, labels = groupNames)

ageGroup = purchase_data.groupby("Age Group")
countAge = ageGroup["SN"].nunique()
percentAge = (countAge/total_players)*100
age_demog = pd.DataFrame({"Total Count": countAge, "Percentage of Players": percentAge})
age_demog.index.name = None
age_demog["Percentage of Players"] = age_demog["Percentage of Players"].map("{:,.2f}%".format)
age_demog
```

Out[10]:

	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 [12]:

```
purchaseCtAge = ageGroup["Purchase ID"].count()
avgPPriceAge = ageGroup["Price"].mean()
totalPValue = ageGroup["Price"].sum()
avgPPerson = totalPValue/countAge
ageDemog = pd.DataFrame({"Purchase Count": purchaseCtAge,
                        "Average Purchase Price": avgPPriceAge,
                        "Total Purchase Value": totalPValue,
                        "Avg Total Purchase per Person": avgPPerson})
ageDemog.index.name = None
ageDemog["Average Purchase Price"] = ageDemog["Average Purchase Price"].map("${:,.2f}".format)
ageDemog["Total Purchase Value"] = ageDemog["Total Purchase Value"].map("${:,.2f}".format)
ageDemog["Avg Total Purchase per Person"] = ageDemog["Avg Total Purchase per Person"].map("${:,.2f}".format)
ageDemog
```

Out[12]:

	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	$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 [13]:

```
spendersStat = purchase_data.groupby("SN")
purchaseCtSpend = spendersStat["Purchase ID"].count()
avgPPriceSpend = spendersStat["Price"].mean()
purchaseTSpend = spendersStat["Price"].sum()
topSpenders = pd.DataFrame({"Purchase Count": purchaseCtSpend,
                           "Average Purchase Price": avgPPriceSpend,
                           "Total Purchase Value": purchaseTSpend})
formatSpenders = topSpenders.sort_values(["Total Purchase Value"], ascending=False).head()

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

Out[13]:

	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, 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 [16]:

```
popularItems = purchase_data[["Item ID", "Item Name", "Price"]]
itemStats = popularItems.groupby(["Item ID", "Item Name"])
countItem = itemStats["Price"].count()
value = (itemStats["Price"].sum())
price = value/countItem
popularItems = pd.DataFrame({"Purchase Count": countItem,
                             "Item Price": price,
                             "Total Purchase Value": value})
formatPop = popularItems.sort_values(["Purchase Count"], ascending=False).head()

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

formatPop
```

Out[16]:

		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 [22]:

```
MostProfitableItems = popularItems.sort_values(["Total Purchase Value"], ascending=False).head()

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


MostProfitableItems
```

Out[22]:

		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 [ ]: