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 [163]:

```
# Dependencies and Setup
import pandas as pd

# File to Load
file_to_load = "../Resources/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 [164]:

```
#Get Unique Players and count the total number of Players
Unique_Players = purchase_data["SN"].unique()
Total_Players = len(Unique_Players)

Total_Players_df = pd.DataFrame({"Total Players":[Total_Players]})
Total_Players_df
```

Out[164]:

```
   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 [165]:

```
#Get Number of Unique Items
Unique_Item_IDs = purchase_data["Item ID"].unique()
Unique_Items = len(Unique_Item_IDs)

#Average Price
Average_Price = purchase_data["Price"].mean()

#Number of Purchass
Purchases = purchase_data["Purchase ID"].count()

#Total Revenue
Total_Revenue = purchase_data["Price"].sum()

#Data Frame to hold the summary values
Purchases_summary_df = pd.DataFrame({"Number of Unique Items":[Unique_Items],
                                   "Average Price":[Average_Price],
                                   "Number of Purchases":[Purchases],
                                   "Total Revenue":[Total_Revenue]})

#formatting values
Purchases_summary_df["Average Price"] = Purchases_summary_df["Average Price"].astype(float).map("${:,.2f}".forma
Purchases_summary_df["Total Revenue"] = Purchases_summary_df["Total Revenue"].astype(float).map("${:,.2f}".forma

Purchases_summary_df
```

Out[165]:

	Number of Unique Items	Average Price	Number of Purchases	Total Revenue
0	183	$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 [166]:

```
#Locate  rows where Gender is Male and calculate count of unique players and percentage of male Players
Purchase_data_male = purchase_data.loc[(Purchase_data_reduced["Gender"] == "Male")]
Count_male_Players= len(Purchase_data_male["SN"].unique())
Percent_male_players = (Count_male_Players*100)/Total_Players

#Locate  rows where Gender is Female and calculate count of unique players and percentage of female Players
Purchase_data_Female = purchase_data.loc[(Purchase_data_reduced["Gender"] == "Female")]
Count_Female_Players= len(Purchase_data_Female["SN"].unique())
Percent_Female_players = (Count_Female_Players*100)/Total_Players

#Locate  rows where Gender is other and calculate count of unique players and percentage of Ohter Players
Purchase_data_Other = purchase_data.loc[(Purchase_data_reduced["Gender"] == "Other / Non-Disclosed")]
Count_other_Players = len(Purchase_data_Other["SN"].unique())
Percent_other_players = (Count_other_Players*100)/Total_Players


#Data Frame to hold the gender Demographics
Gender_Demographics_df = pd.DataFrame({"Gender": ["Male", "Female", "Other / Non-Disclosed"],
                                       "Total Count": [Count_male_Players, Count_Female_Players, Count_other_Pla
                                       "Percentage of Players":[Percent_male_players,Percent_Female_players,Perc

#Setting Gender as Index and formatting to show percetage upto 2 decimal places with a % sign
Gender_Demographics_df= Gender_Demographics_df.set_index("Gender")
Gender_Demographics_df.index.name = None
Gender_Demographics_df["Percentage of Players"] = Gender_Demographics_df["Percentage of Players"].map("{:,.2f}%"
Gender_Demographics_df
```

Out[166]:

	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 [167]:

```
#Total Purchase male
Total_Purchase_male= Purchase_data_male["Purchase ID"].count()

#Total Purchase female
Total_Purchase_female= Purchase_data_Female["Purchase ID"].count()


#Total Purchase other
Total_Purchase_other= Purchase_data_Other["Purchase ID"].count()


#Average Purchase Price male
Avg_Purchase_price_male = Purchase_data_male["Price"].mean()


#Average Purchase Price female
Avg_Purchase_price_female = Purchase_data_Female["Price"].mean()


#Average Purchase Price other
Avg_Purchase_price_other = Purchase_data_Other["Price"].mean()


#Total Purchase Value male
Total_Purchase_Value_male = Purchase_data_male["Price"].sum()


#Total Purchase Value female
Total_Purchase_Value_female = Purchase_data_Female["Price"].sum()


#Total Purchase Value other
Total_Purchase_Value_other = Purchase_data_Other["Price"].sum()


#Average Total Purchase per male
Avg_total_purchase_per_male = Total_Purchase_Value_male/Count_male_Players


#Average Total Purchase per Female
Avg_total_purchase_per_female = Total_Purchase_Value_female/Count_Female_Players
```

```
#Average Total Purchase per other
Avg_total_purchase_per_other = Total_Purchase_Value_other/Count_other_Players


#Data Frame to hold the Purchasing analysis
Purchasing_analysis_df = pd.DataFrame({"Gender": [ "Female","Male", "Other / Non-Disclosed"],
                                       "Purchase Count": [Total_Purchase_female,Total_Purchase_male, Total_Purch
                                       "Average Purchase Price":[Avg_Purchase_price_female,Avg_Purchase_price_ma
                                       "Total Purchase Value":[Total_Purchase_Value_female,Total_Purchase_Value_
                                       "Avg Total Purchase per Person":[Avg_total_purchase_per_female,Avg_total_



#Setting Gender as Index and formatting to show percetage upto 2 decimal places with a $ sign
Purchasing_analysis_df= Purchasing_analysis_df.set_index("Gender")
Purchasing_analysis_df["Average Purchase Price"] = Purchasing_analysis_df["Average Purchase Price"].map("${:,.2
Purchasing_analysis_df["Total Purchase Value"] = Purchasing_analysis_df["Total Purchase Value"].map("${:,.2f}".
Purchasing_analysis_df["Avg Total Purchase per Person"] = Purchasing_analysis_df["Avg Total Purchase per Person"
Purchasing_analysis_df
```

Out[167]:

	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 [168]:

```
age_bins = [0, 9, 14, 19, 24, 29, 34, 39, 100]
group_names = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+" ]

#Add a new column Age Range by binning
purchase_data["Age Range"] = pd.cut(purchase_data["Age"], age_bins, labels=group_names)

#calculation for players and percentage whose age is <10
purchase_data_less_than10 = purchase_data.loc[(purchase_data["Age Range"] == "<10")]
Count_Players_less_than10= len(purchase_data_less_than10["SN"].unique())
Perent_Players_less_than10 = (Count_Players_less_than10*100)/Total_Players

Purchase_Count_less_than10 = len(purchase_data_less_than10["Purchase ID"])
Average_Price_less_than10 = purchase_data_less_than10["Price"].mean()
Total_purchase_less_than10 = purchase_data_less_than10["Price"].sum()
Avg_purchase_per_person_less_than10 = Total_purchase_less_than10/Count_Players_less_than10

#calculation for  players and percentage whose age is 10-14
purchase_data_10_14 = purchase_data.loc[(purchase_data["Age Range"] == "10-14")]
Count_Players_10_14= len(purchase_data_10_14["SN"].unique())
Perent_Players_10_14 = (Count_Players_10_14*100)/Total_Players

Purchase_Count_10_14 = len(purchase_data_10_14["Purchase ID"])
Average_Price_10_14 = purchase_data_10_14["Price"].mean()
Total_purchase_10_14 = purchase_data_10_14["Price"].sum()
Avg_purchase_per_person_10_14 = Total_purchase_10_14/Count_Players_10_14

#calculation for players and percentage whose age is 15-19
purchase_data_15_19 = purchase_data.loc[(purchase_data["Age Range"] == "15-19")]
Count_Players_15_19= len(purchase_data_15_19["SN"].unique())
Perent_Players_15_19 = (Count_Players_15_19*100)/Total_Players

Purchase_Count_15_19 = len(purchase_data_15_19["Purchase ID"])
Average_Price_15_19 = purchase_data_15_19["Price"].mean()
Total_purchase_15_19 = purchase_data_15_19["Price"].sum()
Avg_purchase_per_person_15_19 = Total_purchase_15_19/Count_Players_15_19

#calculation for  players and percentage whose age is 20-24
purchase_data_20_24 = purchase_data.loc[(purchase_data["Age Range"] == "20-24")]
Count_Players_20_24= len(purchase_data_20_24["SN"].unique())
Perent_Players_20_24 = (Count_Players_20_24*100)/Total_Players

Purchase_Count_20_24 = len(purchase_data_20_24["Purchase ID"])
```

```
Average_Price_20_24 = purchase_data_20_24["Price"].mean()
Total_purchase_20_24 = purchase_data_20_24["Price"].sum()
Avg_purchase_per_person_20_24 = Total_purchase_20_24/Count_Players_20_24

#calculation for  players and percentage whose age is 25-29
purchase_data_25_29 = purchase_data.loc[(purchase_data["Age Range"] == "25-29")]
Count_Players_25_29= len(purchase_data_25_29["SN"].unique())
Perent_Players_25_29 = (Count_Players_25_29*100)/Total_Players

Purchase_Count_25_29 = len(purchase_data_25_29["Purchase ID"])
Average_Price_25_29 = purchase_data_25_29["Price"].mean()
Total_purchase_25_29 = purchase_data_25_29["Price"].sum()
Avg_purchase_per_person_25_29 = Total_purchase_25_29/Count_Players_25_29

#calculation for  players and percentage whose age is 30-34
purchase_data_30_34 = purchase_data.loc[(purchase_data["Age Range"] == "30-34")]
Count_Players_30_34= len(purchase_data_30_34["SN"].unique())
Perent_Players_30_34 = (Count_Players_30_34*100)/Total_Players

Purchase_Count_30_34 = len(purchase_data_30_34["Purchase ID"])
Average_Price_30_34 = purchase_data_30_34["Price"].mean()
Total_purchase_30_34 = purchase_data_30_34["Price"].sum()
Avg_purchase_per_person_30_34 = Total_purchase_30_34/Count_Players_30_34

#calculation for  players and percentage whose age is 35-39
purchase_data_35_39 = purchase_data.loc[(purchase_data["Age Range"] == "35-39")]
Count_Players_35_39= len(purchase_data_35_39["SN"].unique())
Perent_Players_35_39 = (Count_Players_35_39*100)/Total_Players

Purchase_Count_35_39 = len(purchase_data_35_39["Purchase ID"])
Average_Price_35_39 = purchase_data_35_39["Price"].mean()
Total_purchase_35_39 = purchase_data_35_39["Price"].sum()
Avg_purchase_per_person_35_39 = Total_purchase_35_39/Count_Players_35_39

#calculation for  players and percentage whose age is 40+
purchase_data_40 = purchase_data.loc[(purchase_data["Age Range"] == "40+")]
Count_Players_40= len(purchase_data_40["SN"].unique())
Perent_Players_40 = (Count_Players_40*100)/Total_Players

Purchase_Count_40  = len(purchase_data_40["Purchase ID"])
Average_Price_40  = purchase_data_40["Price"].mean()
Total_purchase_40  = purchase_data_40["Price"].sum()
Avg_purchase_per_person_40  = Total_purchase_40/Count_Players_40
```

```
#Data Frame to hold the Age Demographics
Age_demographics_df = pd.DataFrame({"Age Group": ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", '
                                    "Total Count": [Count_Players_less_than10,Count_Players_10_14,
                                                    Count_Players_15_19,Count_Players_20_24,Count_Players_25_29,
                                                    Count_Players_30_34,Count_Players_35_39,Count_Players_40],
                                    "Percentage of Players":[Perent_Players_less_than10,Perent_Players_10_14,
                                                             Perent_Players_15_19,Perent_Players_20_24,Perent_Pl
                                                             Perent_Players_30_34,Perent_Players_35_39,Perent_Pl

#Setting Age Group as Index and formatting to show percetage upto 2 decimal places with a
Age_demographics_df= Age_demographics_df.set_index("Age Group")
Age_demographics_df["Percentage of Players"] = Age_demographics_df["Percentage of Players"].map("{:.2f}%".format

Age_demographics_df
```

Out[168]:

	Total Count	Percentage of Players
Age Group		
<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 [173]:

```
#Data Frame to hold Purchasing Analysis by Age
Purchase_by_age_df = pd.DataFrame({"Age Group": ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "4
                                    "Purchase Count": [Purchase_Count_less_than10,Purchase_Count_10_14,
                                                       Purchase_Count_15_19,Purchase_Count_20_24,Purchase_Count_
                                                       Purchase_Count_30_34,Purchase_Count_35_39,Purchase_Count_
                                    "Average Purchase Price":[Average_Price_less_than10,Average_Price_10_14,
                                                             Average_Price_15_19,Average_Price_20_24,Average_Pri
                                                             Average_Price_30_34,Average_Price_35_39,Average_Pri
                                    "Total Purchase Value":[Total_purchase_less_than10,Total_purchase_10_14,
                                                            Total_purchase_15_19,Total_purchase_20_24,
                                                            Total_purchase_25_29,Total_purchase_30_34,Total_purc
                                                            Total_purchase_40],
                                    "Avg Total Purchase per Person":[Avg_purchase_per_person_less_than10,
                                                                     Avg_purchase_per_person_10_14,Avg_purchase_p
                                                                     Avg_purchase_per_person_20_24,Avg_purchase_p
                                                                     Avg_purchase_per_person_30_34,Avg_purchase_p
                                                                     Avg_purchase_per_person_40]})

#Setting Age Group as Index and formatting
Purchase_by_age_df= Purchase_by_age_df.set_index("Age Group")
Purchase_by_age_df["Average Purchase Price"] = Purchase_by_age_df["Average Purchase Price"].map("${:.2f}".format
Purchase_by_age_df["Total Purchase Value"] = Purchase_by_age_df["Total Purchase Value"].map("${:.2f}".format)
Purchase_by_age_df["Avg Total Purchase per Person"] = Purchase_by_age_df["Avg Total Purchase per Person"].map("$
Purchase_by_age_df
```

Out[173]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
Age Group				
<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

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
Age Group				
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 [182]:

```
#Purchase data grouped by SN
Purchase_data_grouped = purchase_data.groupby("SN")

#Purchase Count
Purchase_count = Purchase_data_grouped["Purchase ID"].count()


#Average Purchase Price
Avg_purchase_price = Purchase_data_grouped["Price"].mean()


#Total Purchase Value
Total_purchase_value = Purchase_data_grouped["Price"].sum()


#Data Frame to hold the summary values
Spenders_df = pd.DataFrame({"Purchase Count":Purchase_count,
                            "Average Purchase Price":Avg_purchase_price,
                            "Total Purchase value":Total_purchase_value})




#Sorting
Top_Spenders = Spenders_df.sort_values("Total Purchase value", ascending=False)

#formatting
Top_Spenders["Average Purchase Price"] = Top_Spenders["Average Purchase Price"].astype(float).map("${:,.2f}".for
Top_Spenders["Total Purchase value"] = Top_Spenders["Total Purchase value"].astype(float).map("${:,.2f}".format)

Top_Spenders.head()
```

Out[182]:

```
            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
```

	Purchase Count	Average Purchase Price	Total Purchase value
SN			
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 [191]:

```
#Purchase data Frame Reduced
Reduced_purchase_data = purchase_data[["Item ID","Item Name","Price"]]

Reduced_purchase_data

#Purchase data grouped by Item ID
Purchase_data_grouped = purchase_data.groupby(["Item ID","Item Name"])

#Purchase Count
purchase_count_by_Item = Purchase_data_grouped["Purchase ID"].count()


#Item Price
Item_price = Purchase_data_grouped["Price"].first()


#Total Purchase Value
Total_purchase_value = Purchase_data_grouped["Price"].sum()


#Data Frame to hold summary values
Items_sale_summary_df = pd.DataFrame({"Purchase Count":purchase_count_by_Item,
                                  "Item Price":Item_price,
                                  "Total Purchase Value":Total_purchase_value})


#Sorting
Popular_Items = Items_sale_summary_df.sort_values("Purchase Count", ascending=False)


#formatting
Popular_Items["Item Price"] = Popular_Items["Item Price"].astype(float).map("${:,.2f}".format)
Popular_Items["Total Purchase Value"] = Popular_Items["Total Purchase Value"].astype(float).map("${:,.2f}".forma

Popular_Items.head()
```

Out[191]:

		Purchase Count	Item Price	Total Purchase Value
Item ID	Item Name			

		Purchase Count	Item Price	Total Purchase Value
Item ID	**Item Name**			
178	Oathbreaker, Last Hope of the Breaking Storm	12	$4.23	$50.76
145	Fiery Glass Crusader	9	$4.58	$41.22
108	Extraction, Quickblade Of Trembling Hands	9	$3.53	$31.77
82	Nirvana	9	$4.90	$44.10
19	Pursuit, Cudgel of Necromancy	8	$1.02	$8.16

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 [192]:

```
#Sorting
Profitable_Items = Items_sale_summary_df.sort_values("Total Purchase Value", ascending=False)

#formatting
Profitable_Items["Item Price"] = Profitable_Items["Item Price"].astype(float).map("${:,.2f}".format)
Profitable_Items["Total Purchase Value"] = Profitable_Items["Total Purchase Value"].astype(float).map("${:,.2f}"

Profitable_Items.head()
```

Out[192]:

```
                                                      Purchase Count  Item Price  Total Purchase Value
Item ID                                  Item Name
    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
     92                                Final Critic                8       $4.88                $39.04
    103                              Singed Scalpel                8       $4.35                $34.80
```

In [ ]: