Heroes Of Pymoli Data Analysis

- Of the 1163 active players, the vast majority are male (84%). There also exists, a smaller, but notable proportion of female players (14%).
- Our peak age demographic falls between 20-24 (44.8%) with secondary groups falling between 15-19 (18.60%) and 25-29 (13.4%).

In [1]:
```
# Import dependencies
import pandas as pd

# File to load
csvfile = "Resources/purchase_data.csv"

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

Player Count

In [2]:
```
total_players = purchase_df["SN"].nunique()
total_players_pd = pd.DataFrame({"Total Players":[total_players]})
total_players_pd
```

Out[2]:

	Total Players
0	576

Purchasing Analysis (Total)

In [3]:

```
# Run basic calculations
unique_items_counts = purchase_df["Item Name"].nunique()
average_price = purchase_df["Price"].mean()
total_purchase = purchase_df["Item ID"].count()
total_revenue = sum(purchase_df["Price"])
# Create a summary data frame to hold the results
summary_df = pd.DataFrame({"Number of Unique Items": [unique_items_counts],
                           "Average Price": [average_price],
                           "Number of Purchases": [total_purchase],
                           "Total Revenue": [total_revenue]})
# Give the displayed data cleaner formatting
summary_df["Average Price"] = summary_df["Average Price"].map("${:.2f}".format)
summary_df["Total Revenue"] = summary_df["Total Revenue"].map("${:.2f}".format)
# Display the summary data frame
summary_df
```

Out[3]:

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

Gender Demographics

In [4]:

```
# Count male total
male_count_df = purchase_df.loc[purchase_df["Gender"] == "Male", :]
male_count = male_count_df["SN"].nunique()
# Count female total
female_count_df = purchase_df.loc[purchase_df["Gender"] == "Female", :]
female_count = female_count_df["SN"].nunique()
# Count gender unknown
unknown_count = total_players - male_count - female_count

# Count percentage
male_percent = (male_count/total_players)*100
female_percent =(female_count/total_players)*100
unknown_percent =(unknown_count/total_players)*100

gender_df = pd.DataFrame({" ":["Male","Female", "Other / Non-Disclosed"],
                          "Toatl Count": [male_count,female_count,unknown_count],
                          "Percentage of Players": [male_percent,
                                                    female_percent,unknown_percent]})
gender_index_df = gender_df.set_index(" ")
# Optional: give the displayed data cleaner formatting
gender_index_df["Percentage of Players"] = gender_index_df["Percentage of Players"].map("{:.2f}%
gender_index_df
```

Out[4]:

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

Purchasing Analysis (Gender)

In [5]:

```
# Group data by gender
gender_group = purchase_df.groupby(["Gender"])

# Calculate demanded valuables:
# Purchase Count,Average Purchase Price,Total Purchase Value,Avg Total Purchase per Person
gender_PC = gender_group["SN"].count()
gender_APP = gender_group["Price"].mean()
gender_TPV = gender_group["Price"].sum()
gender_ATPP = gender_TPV /gender_group["SN"].nunique()

# Create a summary data frame to hold the results
gender_2_df = pd.DataFrame({"Purchase Count": gender_PC,
                          "Average Purchase Price": gender_APP,
                          "Total Purchase Value": gender_TPV,
                          "Avg Total Purchase per Person": gender_ATPP
                           })
# Optional: give the displayed data cleaner formatting
gender_2_df["Average Purchase Price"] = gender_2_df["Average Purchase Price"].map("${:.2f}".form
gender_2_df["Total Purchase Value"] = gender_2_df["Total Purchase Value"].map("${:.2f}".format)
gender_2_df["Avg Total Purchase per Person"] = gender_2_df["Avg Total Purchase per Person"].map(

gender_2_df
```

Out[5]:

	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	$1967.64	$4.07
Other / Non-Disclosed	15	$3.35	$50.19	$4.56

Age Demographics

In [6]:

```python
# Create the bins in which Data will be held
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+"]

# Categorize the existing players using the age bins. Hint: use pd.cut()
purchase_df["Age Group"] = pd.cut(purchase_df["Age"], bins, labels=group_names, include_lowest=T

# Calculate the numbers and percentages by age group
age_group = purchase_df.groupby("Age Group")
total_count = age_group["SN"].nunique()
percent = total_count.div(total_count.sum(axis=0))*100

# Create a summary data frame to hold the results
age_df = pd.DataFrame({"Total Count": total_count,
                       "Percentage of Players": percent })
# Optional: round the percentage column to two decimal points
age_df["Percentage of Players"] = age_df["Percentage of Players"].map("{:.2f}%".format)
# Display Age Demographics Table
age_df
```

Out[6]:

	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)

In [7]:

```
# Bin the purchase_data data frame by age(see above)
# Run basic calculations to obtain purchase count, avg. purchase price,
# avg. purchase total per person etc. in the table below
age_PC = age_group["Item ID"].count()
age_APP = age_group["Price"].mean()
age_TPV = age_group["Price"].sum()
age_ATPP = age_group["Price"].sum()/total_count

#Create a summary data frame to hold the results
age_2_df = pd.DataFrame({"Purchase Count": age_PC,
                          "Average Purchase Price": age_APP,
                          "Total Purchase Value": age_TPV,
                          "Avg Total Purchase per Person": age_ATPP
                           })
#Optional: give the displayed data cleaner formatting
age_2_df["Average Purchase Price"] = age_2_df["Average Purchase Price"].map("${:.2f}".format)
age_2_df["Total Purchase Value"] = age_2_df["Total Purchase Value"].map("${:.2f}".format)
age_2_df["Avg Total Purchase per Person"] = age_2_df["Avg Total Purchase per Person"].map("${:.2
#Display the summary data frame
age_2_df
```

Out[7]:

	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
40+	13	$2.94	$38.24	$3.19

Top Spenders

In [8]:
```
# Group data by SN
SN_group = purchase_df.groupby(["SN"])
# Calculate variables
SN_PC = SN_group["SN"].count()
SN_APP = SN_group["Price"].mean()
SN_TPV = SN_group["Price"].sum()
# Create a summary data frame to hold the results
SN_df = pd.DataFrame({"Purchase Count": SN_PC,
                      "Average Purchase Price": SN_APP,
                      "Total Purchase Value": SN_TPV })
# Sort the total purchase value column in descending order
top_spender_df = SN_df.sort_values(["Total Purchase Value"], ascending=False)
#Optional: give the displayed data cleaner formatting
top_spender_df["Average Purchase Price"] = top_spender_df["Average Purchase Price"].map("${:.2f}"
top_spender_df["Total Purchase Value"] = top_spender_df["Total Purchase Value"].map("${:.2f}".fo
# Display a preview of the summary data frame
top_spender_df.head()
```

Out[8]:

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

In [9]:

```
 Retrieve the Item ID, Item Name, and Price columns
opular_items_df = purchase_df[["Item ID", "Item Name","Price"]]
 Group by Item ID and Item Name. Perform calculations to obtain purchase count, item price, and t
I_group = popular_items_df.groupby(["Item ID","Item Name"])
I_PC = PI_group["Item ID"].count()
# Question: I don't know why I can use PI_group["Price"].mean() to get anwser.
# Is that beacuse it has two different prices?
I_price = PI_group["Price"].mean()
I_TPV = PI_group["Price"].sum()
 Create a summary data frame to hold the results
I_df = pd.DataFrame({"Purchase Count": PI_PC,
                     "Item Price": PI_price,
                     "Total Purchase Value": PI_TPV })
 Sort the purchase count column in descending order
I_2_df = PI_df.sort_values(["Purchase Count"], ascending=False)
 Optional: give the displayed data cleaner formatting
I_2_df["Item Price"] = PI_2_df["Item Price"].map("${:.2f}".format)
I_2_df["Total Purchase Value"] = PI_2_df["Total Purchase Value"].map("${:.2f}".format)
 Display a preview of the summary data frame
I_2_df.head()
```

Out[9]:

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
108	Extraction, Quickblade Of Trembling Hands	9	$3.53	$31.77

Most Profitable Items

In [10]:

```
# Sort the above table by total purchase value in descending order
PI_2_df = PI_df.sort_values(["Total Purchase Value"], ascending=False)
# Optional: give the displayed data cleaner formatting
PI_2_df["Item Price"] = PI_2_df["Item Price"].map("${:.2f}".format)
PI_2_df["Total Purchase Value"] = PI_2_df["Total Purchase Value"].map("${:.2f}".format)
# Display a preview of the data frame
PI_2_df.head()
```

Out[10]:

		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