Heroes Of Pymoli Data Analysis

- Observed Trend 1: There are significantly more male players than females.
- Observed Trend 2: More than half of the players are under age 24.
- Observed Trend 3: Players above age 35 spent more per person as indicated in normalized average.

In [1]:

```
import pandas as pd
import os
hop_data = os.path.join('purchase_data2.json')
hop_data_df = pd.read_json(hop_data, orient=None, typ='frame', dtype=True, convert_axes=True, convert_dates=True, keep_default_dates=True, numpy=False, precise_float=False, date_unit=None, encoding=None, lines=False)
hop_data_df.head()
```

Out[1]:

	Age	Gender	Item ID	Item Name	Price	SN
0	20	Male	93	Apocalyptic Battlescythe	4.49	Iloni35
1	21	Male	12	Dawne	3.36	Aidaira26
2	17	Male	5	Putrid Fan	2.63	Irim47
3	17	Male	123	Twilight's Carver	2.55	Irith83
4	22	Male	154	Feral Katana	4.11	Philodil43

Player Count

In [2]:

```
total_players = hop_data_df["SN"].nunique()
# print(unique_player)
# print(f" Total Players: {total_players}")
totalplayers = pd.DataFrame({"Total Players":[total_players]})
totalplayers
```

Out[2]:

	Total Players
0	74

Purchasing Analysis (Total)

In [3]:

```
unique_item = hop_data_df["Item ID"].nunique()
# print(unique_item)
average_price = round(hop_data_df["Price"].mean(), 2)
# print(average_price)
number_purchase = hop_data_df["Item ID"].count()
# print(number_purchase)
total_revenue = round(hop_data_df["Price"].sum(), 2)
# print(total_revenue)
purchasing_analysis = pd.DataFrame({"Number of Unique Items":[unique_item],
                                    "Average Price":[average_price],
                                    "Number of Purchases": [number_purchase],
                                    "Total Revenue":[total_revenue]})
purchasing_analysis["Average Price"] = purchasing_analysis["Average Price"].map(
'${:,.2f}'.format)
purchasing_analysis["Total Revenue"] = purchasing_analysis["Total Revenue"].map(
'${:,.2f}'.format)
purchasing_analysis
```

Out[3]:

	Average Price	Number of Purchases	Number of Unique Items	Total Revenue
0	$2.92	78	64	$228.10

Gender Demographics

In [4]:

```
gender_df = hop_data_df.iloc[:,[1,5]]
unique_df = gender_df.drop_duplicates()
total_gender = unique_df.groupby("Gender")
count = total_gender["Gender"].count()
percentage = round(count/count.sum()*100, 2)
gender_demo = pd.concat([percentage, count], axis=1)
gender_demo.columns = ["Percentage of Players", "Total Count"]
gender_demo
```

Out[4]:

	Percentage of Players	**Total Count**
Gender		
Female	17.57	13
Male	81.08	60
Other / Non-Disclosed	1.35	1

Purchasing Analysis (Gender)

In [5]:

```python
total_purchase = hop_data_df.groupby("Gender")
gender_purchase = total_purchase["Gender"].count()
gender_average = round(total_purchase["Price"].mean(), 2)
gender_revenue = round(total_purchase["Price"].sum(), 2)
normalized = round((gender_revenue / count), 2)
gender_buy_analysis = pd.concat([gender_purchase, gender_average, gender_revenue, normalized], axis=1)
gender_buy_analysis.columns = ["Purchase Count", "Average Purchase Price", "Total Revenue", "Normalized Average"]
gender_buy_analysis["Average Purchase Price"] = gender_buy_analysis["Average Purchase Price"].map('${:,.2f}'.format)
gender_buy_analysis["Total Revenue"] = gender_buy_analysis["Total Revenue"].map('${:,.2f}'.format)
gender_buy_analysis["Normalized Average"] = gender_buy_analysis["Normalized Average"].map('${:,.2f}'.format)
gender_buy_analysis
```

Out[5]:

	Purchase Count	Average Purchase Price	Total Revenue	Normalized Average
Gender				
Female	13	$3.18	$41.38	$3.18
Male	64	$2.88	$184.60	$3.08
Other / Non-Disclosed	1	$2.12	$2.12	$2.12

Age Demographics

In [6]:

```
# print(hop_data_df["Age"].max())
# print(hop_data_df["Age"].min())
bins = [0,9,14,19,24,29,34,39,50]
age_label = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]
pd.cut(hop_data_df["Age"], bins, labels = age_label)
hop_data_df["Age Group"] = pd.cut(hop_data_df["Age"], bins, labels = age_label)
age_df = hop_data_df.iloc[:,[5,6]]
unique_player_df = age_df.drop_duplicates()
age_group = unique_player_df.groupby("Age Group")
# HOW TO sort to put age < 10 atop?
age_count = age_group["Age Group"].count()
age_percentage = round(age_count/age_count.sum()*100, 2)
age_demo = pd.concat([age_percentage, age_count], axis=1)
age_demo.columns = ["Percentage of Players", "Total Count"]
age_demo
```

Out[6]:

	Percentage of Players	Total Count
Age Group		
10-14	4.05	3
15-19	14.86	11
20-24	45.95	34
25-29	10.81	8
30-34	8.11	6
35-39	8.11	6
40+	1.35	1
<10	6.76	5

Purchasing Analysis (Age)

In [7]:

```
age_analysis_df = hop_data_df.groupby("Age Group")
age_purchase = age_analysis_df["Age Group"].count()
age_average = round(age_analysis_df["Price"].mean(), 2)
age_revenue = round(age_analysis_df["Price"].sum(), 2)
age_normalized = round(age_revenue/age_count, 2)
age_buy_analysis = pd.concat([age_purchase, age_average, age_revenue, age_normalized], axis=1)
age_buy_analysis.columns = ["Purchase Count", "Average Purchase Price","Total Revenue","Normalized Average"]
age_buy_analysis["Average Purchase Price"] = age_buy_analysis["Average Purchase Price"].map('${:,.2f}'.format)
age_buy_analysis["Total Revenue"] = age_buy_analysis["Total Revenue"].map('${:,.2f}'.format)
age_buy_analysis["Normalized Average"] = age_buy_analysis["Normalized Average"].map('${:,.2f}'.format)
age_buy_analysis
```

Out[7]:

	Purchase Count	Average Purchase Price	Total Revenue	Normalized Average
Age Group				
10-14	3	$2.99	$8.96	$2.99
15-19	11	$2.76	$30.41	$2.76
20-24	36	$3.02	$108.89	$3.20
25-29	9	$2.90	$26.11	$3.26
30-34	7	$1.98	$13.89	$2.32
35-39	6	$3.56	$21.37	$3.56
40+	1	$4.65	$4.65	$4.65
<10	5	$2.76	$13.82	$2.76

Top Spenders

In [8]:

```
top_spenders = hop_data_df.groupby("SN")
top_spender_total = top_spenders["Price"].sum()
top_spender_count = top_spenders["Item ID"].count()
top_spender_average = round(top_spender_total/top_spender_count, 2)
top5spenders = pd.concat([top_spender_count, top_spender_average,top_spender_total], axis=1).nlargest(5,"Price")
top5spenders.columns = ["Purchase Count", "Average Purchase Price", "Total Purchase Value"]
top5spenders["Average Purchase Price"] = top5spenders["Average Purchase Price"].map('${:,.2f}'.format)
top5spenders["Total Purchase Value"] = top5spenders["Total Purchase Value"].map('${:,.2f}'.format)
top5spenders
```

Out[8]:

	Purchase Count	Average Purchase Price	Total Purchase Value
SN			
Sundaky74	2	$3.70	$7.41
Aidaira26	2	$2.56	$5.13
Eusty71	1	$4.81	$4.81
Chanirra64	1	$4.78	$4.78
Alarap40	1	$4.71	$4.71

Most Popular Items

In [9]:

```
popular_df = hop_data_df.loc[:,["Item ID", "Item Name","Price"]]
most_popular = popular_df.groupby("Item Name")
most_popular_count = most_popular["Item ID"].count()
most_popular_total = most_popular["Price"].sum()
most_popular_average = round(most_popular_total/most_popular_count, 2)
most_popular_table = pd.DataFrame({"Purchase Count": most_popular_count,
                                   "Item Price": most_popular_average,
                                    "Total Purchase Value": most_popular_total})
.nlargest(5, "Purchase Count")
most_popular_table["Total Purchase Value"]= most_popular_table["Total Purchase V
alue"].map('${:,.2f}'.format)
most_popular_table["Item Price"]= most_popular_table["Item Price"].map('${:,.2f}
'.format)
most_popular_table
```

Out[9]:

	Item Price	Purchase Count	Total Purchase Value
Item Name			
Mourning Blade	$3.64	3	$10.92
Apocalyptic Battlescythe	$4.49	2	$8.98
Betrayer	$4.12	2	$8.24
Crucifer	$2.64	2	$5.29
Deadline, Voice Of Subtlety	$1.29	2	$2.58

Most Profitable Items

In [10]:

```
most_profitable = popular_df.groupby("Item Name")
most_profitable_count = most_profitable["Item Name"].count()
most_profitable_total = most_profitable["Price"].sum()
most_profitable_average = round(most_profitable_total/most_profitable_count, 2)
most_profitable_table = pd.DataFrame({"Purchase Count": most_profitable_count,
                                   "Item Price": most_profitable_average,
                                    "Total Purchase Value": most_profitable_tota
l}).nlargest(5, "Total Purchase Value")
most_profitable_table["Total Purchase Value"]= most_profitable_table["Total Purc
hase Value"].map('${:,.2f}'.format)
most_profitable_table["Item Price"]= most_profitable_table["Item Price"].map('${
:,.2f}'.format)
most_profitable_table
```

Out[10]:

	Item Price	Purchase Count	Total Purchase Value
Item Name			
Mourning Blade	$3.64	3	$10.92
Heartstriker, Legacy of the Light	$4.71	2	$9.42
Apocalyptic Battlescythe	$4.49	2	$8.98
Betrayer	$4.12	2	$8.24
Feral Katana	$4.11	2	$8.22

In [ ]: