

Analysis

- Out of 576 active players, 84% were male and 14% were female.
- The main age demographic are players aged 20-24 (44.79%). Secondary and tertiary groups fell between 15-19 (18.60%) and 25-29 (13.4%), respectively.
- The most popular item was "Oathbreaker, Last Hope of the Breaking Storm"

Player Count

```
In [1]: # Dependencies and Setup
import pandas as pd
import numpy as np

# File to Load
pymoli_df = "Resources/purchase_data.csv"

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

```
In [3]: player_count = purchase_data["SN"].nunique()
total_df = pd.DataFrame({"Total Players": [player_count]})
total_df
```

Out[3]:

	Total Players
0	576

Purchasing Analysis (Total)

```
In [18]: item_count = purchase_data["Item ID"].nunique()
average_price = round(purchase_data["Price"].mean(),2)
purchase_count = purchase_data["Purchase ID"].nunique()
total_revenue = purchase_data["Price"].sum()

purchase_summary_df = pd.DataFrame({
    "Number of Unique Items": [item_count],
    "Average Price": [average_price],
    "Number of Purchases": [purchase_count],
    "Total Revenue": [total_revenue]
})

purchase_summary_df
```

Out[18]:

	Number of Unique Items	Average Price	Number of Purchases	Total Revenue
0	183	3.05	780	2379.77

Gender Demographics

```
In [7]: gender_count = purchase_data["Gender"].value_counts()

clean_gender_count = purchase_data[['SN', 'Gender']]

count_df = clean_gender_count.drop_duplicates()

gender_count_total = count_df["Gender"].value_counts()

percent = gender_count_total/count_df["Gender"].count() *100
round(percent,2)

gender_df = pd.DataFrame({"Total Count": gender_count_total,
    "Percentage of Players": percent.map("{:,.2f}%".format)
})

gender_df
```

Out[7]:

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

Purchasing Analysis (Gender)

```
In [26]: grouped_gender_df = purchase_data.groupby(['Gender'])

gender_purchase_count = grouped_gender_df["Price"].count()
purchase_price_avg = round(grouped_gender_df["Price"].mean(), 2)
purchase_total_avg = grouped_gender_df["Price"].sum()
avg_purchase_pp = purchase_total_avg/gender_count_total

purchase_analysis_gender = pd.DataFrame({"Purchase Count": gender_purcha
                                         "Average Purchase Price": purch
                                         "Total Purchase Value": purchas
                                         "Avg Total Purchase Per Person"

}))

purchase_analysis_gender
```

Out[26]:

	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

```

In [21]: bins = [0, 9.90, 14.90, 19.90, 24.90, 29.90,
                34.90, 39.90, 999]

group_labels = ["< 10", "10-14", "15-19", "20-24", "25-29", "30-34",
                "35-39", "40+"]

pd.cut(purchase_data["Age"], bins, labels=group_labels)
purchase_data["Age Group"] = pd.cut(purchase_data["Age"], bins, labels=g

age_df = purchase_data.groupby("Age Group")
total_age_count = age_df["Age"].count()
clean_age_count = purchase_data[["SN", "Gender", "Age", "Age Group"]].drop
clean_age_df = clean_age_count.drop_duplicates()
clean_age_df2 = clean_age_df.groupby("Age Group")
clean_total_age_count = clean_age_df2["Age"].count()
age_percentage = clean_age_df2["Age"].count()/player_count *100

age_demographic_df = pd.DataFrame({"Total Count": clean_total_age_count,
                                   "Percentage of Players": age_percent

age_demographic_df

```

Out[21]:

	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 [25]: bins = [0, 9.90, 14.90, 19.90, 24.90, 29.90,
                34.90, 39.90, 999]

group_labels = ["< 10", "10-14", "15-19", "20-24", "25-29", "30-34",
                "35-39", "40+"]

pd.cut(purchase_data["Age"], bins, labels=group_labels)
age_purchase_df = purchase_data.groupby("Age Group")
age_purchase_count = age_purchase_df["Item ID"].count()
age_avgpurchase = age_purchase_df["Price"].mean()
total_purchase_value = age_purchase_df["Price"].sum()
avg_purchase_pp = total_purchase_value/clean_total_age_count

purchase_analysis_age_df = pd.DataFrame({"Purchase Count":age_purchase_count,
                                         "Average Purchase Price":age_avgpurchase.map("${:,.2f}"),
                                         "Total Purchase Value": total_purchase_value,
                                         "Avg Total Purchase Per Person": avg_purchase_pp
                                         })

purchase_analysis_age_df
```

Out[25]:

	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	\$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

```
In [12]: grouped_spenders_df = purchase_data.groupby(['SN'])

ts_purchase_count = grouped_spenders_df["SN"].count()
ts_average_purchase = grouped_spenders_df["Price"].mean()
ts_total_purchase = grouped_spenders_df["Price"].sum()

top_spenders_df = pd.DataFrame({"Purchase Count":ts_purchase_count, "Average Purchase Price":ts_average_purchase.map("${:,.2f}"), "Total Purchase Value": ts_total_purchase})

sorted_spenders_df = top_spenders_df.sort_values(by=['Purchase Count'], ascending=False)
sorted_spenders_df.head()
```

Out[12]:

	Purchase Count	Average Purchase Price	Total Purchase Value
SN			
Lisosia93	5	\$3.79	\$18.96
Iral74	4	\$3.40	\$13.62
Idastidru52	4	\$3.86	\$15.45
Asur53	3	\$2.48	\$7.44
Inguron55	3	\$3.70	\$11.11

Most Popular Items

```
In [28]: popular_item_df = purchase_data[["Item ID", "Item Name", "Price"]]
popular_item_df = popular_item_df.rename(columns={"Price": "Item Price"})

grouped_popular = popular_item_df.groupby(['Item ID', 'Item Name'])
avg_item_price = grouped_popular.mean()['Item Price']
popular_purchase_count = grouped_popular["Item ID"].count()
popular_purchase_value = grouped_popular["Item Price"].sum()

popular_df = pd.DataFrame({"Purchase Count": popular_purchase_count, "Item
                           "Total Purchase Value": popular_purchase
                           })
sorted_popular_df = popular_df.sort_values(by=['Purchase Count'], ascend
sorted_popular_df.head()
```

Out[28]:

		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

```
In [32]: sorted_profitable_df = sorted_popular_df.sort_values(by=['Purchase Count', 'Item Price'])
sorted_profitable_df.head()
```

Out[32]:

		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
108	Extraction, Quickblade Of Trembling Hands	9	3.53	\$31.77
19	Pursuit, Cudgel of Necromancy	8	1.02	\$8.16