Heroes of Pymoli Data Analysis

Target Market: Customer base consists of males and those that are between the ages of 20-24.

- Recommendation:Use Facebook Audience Insights and Google Analytics to learn more about this target audience. Develop and implement a marketing campaign based on those insights.

The most profitable items are the most expensive: Aside from item 52, which lists as number 1 in "Top Sellers" and "Most Profitable", the higher priced items are generating the most revenue for the company.

- Recommendation: Gather additional information through in-game data such such as item utilization and chat text, as well as surveying customers who made these purchases, to identify attributes/features that drove them to make these purchases. Ensure we are incorporating these attributes/features into our product design strategy.

About 25% of total purchases were made by repeat customers.

- Recommendation:Conduct an analysis on those repeat customers. Identify what drove them to make those repeat purchases through surveys and segmentation(gender/age).

In [1]:

```
#import
import pandas as pd
import numpy as np
rawpurchasedata = "purchase_data_2.json"
sales_data = pd.read_json(rawpurchasedata)


```

Player Count

In [2]:

```
#Create Dataframe with a unique count of players using SN as the unique identif
pc_df = pd.DataFrame(
    {"Players Count": [sales_data["SN"].nunique()]
    })

pc_df



```

Out[2]:

	Players Count
0	581

Purchasing Analysis (Total)

In [3]:

```
#create lists with relevant fields

uniqueitems = sales_data["Item Name"].nunique()
average_price = [np.round(sales_data["Price"].mean(), 2)]
number_of_purchases = [sales_data["Price"].count()]
total_revenue =  [sales_data["Price"].sum()]
number_of_people= sales_data["SN"].nunique()

#create Dataframe with those lists
pa_df = pd.DataFrame(
    {"Number of Unique Items": uniqueitems ,"Average Price": average_price ,
     "Number of Purchases": number_of_purchases,"Total Revenue": total_revenue

    })
#change formating
pa_df["Average Price"] = pa_df["Average Price"].map("${:.2f}".format)
pa_df["Total Revenue"] = pa_df["Total Revenue"].map("${:.2f}".format)

#change column order
pa_df[['Number of Unique Items','Average Price','Number of Purchases','Total Re
```

Out[3]:

	Number of Unique Items	Average Price	Number of Purchases	Total Revenue
0	178	$3.03	780	$2365.17

Gender Demographics

In [4]:

```
#find total number of unique players and assign to "total"
total=sales_data['SN'].nunique()
#groupby gender based on unique counts of those corresponding genders(based on
ga1_uniquecount = sales_data.groupby(["Gender"])["SN"].nunique()

#create percent
ga1_percent= (ga1_uniquecount/total)*100
ga1_percent= ga1_percent.map("{:.2f}%".format)

#create new data frame based on unique count and percent
ga1 = pd.DataFrame(
    {"Percentage of Players": ga1_percent,
     "Total Count": ga1_uniquecount})

ga1
```

Out[4]:

	Percentage of Players	Total Count
Gender		
Female	17.04%	99
Male	81.76%	475
Other / Non-Disclosed	1.20%	7

Purchasing Analysis (Gender)

In [5]:

```
#create multiple groupby objects
countoft = sales_data
ga2_uniquecount = sales_data.groupby(["Gender"])["SN"].nunique()
ga2_purchase_count = sales_data.groupby(["Gender"])["SN"].count()
ga2_avg_puchase_price = sales_data.groupby(["Gender"])["Price"].mean()
ga2_ttl_purchase_price = sales_data.groupby(["Gender"])["Price"].sum()
ga2_nmlzd_totals = ga2_ttl_purchase_price/ga2_uniquecount
#change formatting
ga2_avg_puchase_price = ga2_avg_puchase_price.map("${:,.2f}".format)
ga2_ttl_purchase_price = ga2_ttl_purchase_price.map("${:,.2f}".format)
ga2_nmlzd_totals= ga2_nmlzd_totals.map("${:,.2f}".format)
#create new dataframe with groupby ibjects
ga2 = pd.DataFrame({"Purchase Count": ga2_purchase_count,
                    "Average Purchase Price":ga2_avg_puchase_price,
                    "Total Purchase Value":ga2_ttl_purchase_price,
                    "Normalized Totals":ga2_nmlzd_totals})
ga2 = ga2.round(2)
#change order
ga2[['Purchase Count','Average Purchase Price','Total Purchase Value','Normalize

```

Out[5]:

Gender	Purchase Count	Average Purchase Price	Total Purchase Value	Normalized Totals
Female	130	$3.04	$395.80	$4.00
Male	642	$3.03	$1,945.51	$4.10
Other / Non-Disclosed	8	$2.98	$23.86	$3.41

Age Demographics

In [6]:

```
#create unique total count based on SN; assign to total2
total2=sales_data['SN'].nunique()

#create dataframe
age_demo=pd.DataFrame(
    {"Age": sales_data['Age'],"Percentage of Players": sales_data['Age'],
     "Players Count": sales_data['Age'],})

#create a function that replaces age within the age_demo column with the corres
def agerange_func(row):
    if row['Age'] < [10]:
        val = "<10"
    elif(row['Age'] >=[10]) & (row['Age'] <=[14]):
        val = "10-14"
    elif(row['Age'] >=[15]) & (row['Age'] <=[19]):
        val = "15-19"
    elif(row['Age'] >=[20]) & (row['Age'] <=[24]):
        val = "20-24"
    elif(row['Age'] >=[25]) & (row['Age'] <=[29]):
        val = "25-29"
    elif(row['Age'] >=[30]) & (row['Age'] <=[34]):
        val = "30-34"
    elif(row['Age'] >=[35]) & (row['Age'] <=[39]):
        val = "35-39"
    else:
        val = "40+"
    return val
age_demo['Age'] = age_demo.apply(agerange_func, axis=1)

#create new dataframe
age_demoaba=pd.DataFrame(
    {"Age": age_demo['Age'],"SN":sales_data['SN']})

#calculate totals and change format
age_demoaba1 = age_demoaba.groupby(["Age"])["SN"].nunique()
age_demoaba2 = (age_demoaba.groupby(["Age"])["SN"].nunique()/total2*100)
age_demoaba2 = age_demoaba2.map("{:.2f}%".format)
age_demo2_f=pd.DataFrame(
    {"Percentage of Players": age_demoaba2,"Total Count": age_demoaba1})
del age_demo2_f.index.name

age_demo2_f
```

Out[6]:

	Percentage of Players	Total Count
10-14	3.61%	21
15-19	17.73%	103
20-24	39.59%	230
25-29	18.07%	105
30-34	8.09%	47
35-39	7.06%	41
40+	1.38%	8
<10	4.48%	26

Purchasing Analysis (Age)

In [7]:

```python
#create dataframe
age_demo2 = pd.DataFrame(
    {"Age": sales_data['Age'],"Purchase Count": sales_data['Age'],
    "Average Purchase Price": sales_data['Price'],
    "Total Purchase Value": sales_data['Price'],
    "Normalized Totals": sales_data['Price']})


#create a function that replaces age within the age_demo2 column with the corre
def agerange_func(row):
    if row['Age'] < [10]:
        val = "<10"
    elif(row['Age'] >=[10]) & (row['Age'] <=[14]):
        val = "10-14"
    elif(row['Age'] >=[15]) & (row['Age'] <=[19]):
        val = "15-19"
    elif(row['Age'] >=[20]) & (row['Age'] <=[24]):
        val = "20-24"
    elif(row['Age'] >=[25]) & (row['Age'] <=[29]):
        val = "25-29"
    elif(row['Age'] >=[30]) & (row['Age'] <=[34]):
        val = "30-34"
    elif(row['Age'] >=[35]) & (row['Age'] <=[39]):
        val = "35-39"
    else:
        val = "40+"
    return val
age_demo2['Age'] = age_demo2.apply(agerange_func, axis=1)

age_demo2a=pd.DataFrame(
    {"Age": age_demo2['Age'],"Average Purchase Price": age_demo2['Average Purcha


#calculate totals and change format
ad2_pc = age_demo2a.groupby(["Age"])["Average Purchase Price"].count()
ad2_avpp = age_demo2a.groupby(["Age"])["Average Purchase Price"].mean()
ad2_ttl = age_demo2a.groupby(["Age"])["Average Purchase Price"].sum()
ad2_nmzdttl11 =age_demo2a.groupby(["Age"])["Average Purchase Price"].nunique()
ad2_nmzdttl = ad2_ttl /ad2_nmzdttl11

ad2_avpp= ad2_avpp.map("${:,.2f}".format)
ad2_ttl = ad2_ttl.map("${:,.2f}".format)
ad2_nmzdttl= ad2_nmzdttl.map("${:,.2f}".format)

age_demo3_f=pd.DataFrame(
    {"Purchase Count": ad2_pc,
    "Average Purchase Price": ad2_avpp,
    "Total Purchase Value": ad2_ttl,
    "Normalized Totals": ad2_nmzdttl})
del age_demo3_f.index.name

#reorder
age_demo3_f[['Purchase Count','Average Purchase Price','Total Purchase Value','
```

Out[7]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Normalized Totals
10-14	31	$2.98	$92.42	$3.19
15-19	132	$3.08	$406.62	$4.73
20-24	316	$3.01	$950.59	$7.67
25-29	137	$3.05	$417.21	$5.03
30-34	62	$2.93	$181.40	$3.78
35-39	54	$3.18	$171.50	$4.08
40+	11	$3.41	$37.47	$3.75
<10	37	$2.92	$107.96	$3.37

Top Spenders

In [8]:

```
#create groupby objects and with relevant calculations
ts_c_df=pd.DataFrame(sales_data.groupby("SN")["Price"].count())
ts_avg_df=pd.DataFrame(sales_data.groupby("SN")["Price"].mean())
ts_ttl_df=pd.DataFrame(sales_data.groupby("SN")["Price"].sum())

#create dataframe with groupby objects
ts=pd.DataFrame(
    {"Purchase Count": ts_c_df['Price'],
     "Average Purchase Price": (ts_avg_df['Price']).map("${:,.2f}".format),
     "Total Purchase Value": (ts_ttl_df['Price'])})

#sort, format, and rename
ts=ts.sort_values(["Total Purchase Value"], ascending=False)
ts["Total Purchase Value"] = ts["Total Purchase Value"].map("${:,.2f}".format)
ts[['Purchase Count','Average Purchase Price','Total Purchase Value']].head(5)
```

Out[8]:

	Purchase Count	Average Purchase Price	Total Purchase Value
SN			
Chamalo71	4	$3.36	$13.45
Strithenu87	4	$3.21	$12.83
Mindosia50	3	$4.00	$12.01
Aeralria27	3	$3.79	$11.38
Eudai71	3	$3.79	$11.37

Most Popular Items

In [9]:

```
#create groupby objects and with relevant calculations
mp1_c_df=pd.DataFrame(sales_data.groupby(['Item ID','Item Name']) ["Price"].coun
mp1_avg_df=pd.DataFrame(sales_data.groupby(["Item ID", "Item Name"])["Price"].me
mp1_ttl_df=pd.DataFrame(sales_data.groupby(["Item ID", "Item Name"])["Price"].su

#create dataframe with groupby objects
mp1=pd.DataFrame(
    {"Purchase Count": mp1_c_df['Price'],
     "Item Price": (mp1_avg_df['Price']).map("${:,.2f}".format),
     "Total Purchase Value": (mp1_ttl_df['Price']).map("${:,.2f}".format)})

#sort, format, and rename
mp1=mp1.sort_values(["Purchase Count"], ascending=False)
mp1[['Purchase Count','Item Price','Total Purchase Value']].head(5)
```

Out[9]:

```
                          Purchase Count  Item Price  Total Purchase Value
Item ID  Item Name
     52            Hatred             11       $4.59                $50.49
    174   Primitive Blade              9       $1.39                $12.51
    111      Misery's End              9       $4.90                $44.10
     91           Celeste              8       $2.07                $16.56
    143 Frenzied Scimitar              8       $3.35                $26.80
```

Most Profitable Items

In [10]:

```
#create groupby objects and with relevant calculations
mp2_c_df=pd.DataFrame(sales_data.groupby(['Item ID','Item Name']) ["Price"].cou
mp2_avg_df=pd.DataFrame(sales_data.groupby(["Item ID", "Item Name"])["Price"].me
mp2_ttl_df=pd.DataFrame(sales_data.groupby(["Item ID", "Item Name"])["Price"].su

#create dataframe with groupby objects
mp2=pd.DataFrame(
    {"Purchase Count": mp2_c_df['Price'],
     "Item Price": (mp2_avg_df['Price']).map("${:,.2f}".format),
     "Total Purchase Value": (mp2_ttl_df['Price'])})

#sort, format, and rename
mp2=mp2.sort_values(["Total Purchase Value"], ascending=False).head(30)
mp2["Total Purchase Value"] = mp2["Total Purchase Value"].map("${:,.2f}".format
mp2[['Purchase Count','Item Price','Total Purchase Value']].head(5)
```

Out[10]:

```
                                       Purchase Count  Item Price  Total Purchase Value
Item ID  Item Name
     52                         Hatred             11       $4.59                $50.49
    111                   Misery's End              9       $4.90                $44.10
    120                         Agatha              8       $4.93                $39.44
     93       Apocalyptic Battlescythe              8       $4.85                $38.80
     49  The Oculus, Token of Lost Worlds           8       $4.61                $36.88
```

In [ ]:

```

```