**Heroes Of Pymoli Data Analysis**

Task: Include a written description of three observable trends based on the data

OBSERVED TREND 1: *Gender Demographics and Repeat User*s: 84% of players are male as compared to female and other users. We can improve the interface or graphics so as to provide user experience which is appealing to all genders. While the total purchase count is 780, the number of unique players is only 573, suggesting there is a sizeable number of repeat purchasers for the game.

OBSERVED TREND 2: *Age Demographics*: Players from 20-24 ages are the top users accounting for 45% of the entire sample. There is a huge drop in the number for players from ages 25-29. We can look at improving the stages in the game in order to attract players from the age group 25-29.

Unsurprisingly, the 20-24 age bracket makes up the largest portion of Total Purchase Value, but in Average Purchase Price it is behind two other categories, suggesting this group spends less than two other groups on a per purchase basis. Under 10 years olds and 35-39 year olds tend to spend a little more per person, but they are also substantially smaller groups.

OBSERVED TREND 3: *Purchasing Analysis:* Highest total purchase value is achieved by male users with $1,968 revenue. Despite the much higher volume of male purchasers, the average total purchase per person at $4.47 is 40 cents higher with women. Again, total revenue could be enhanced with an increased number of female and other users by including new features or improving the existing ones.

## Dependencies and starter code

In [2]:

**import** **pandas** **as** **pd**

*# # CSV File to Load*

file\_to\_load = "purchase\_data.csv"

*# Read Purchasing File and store into Pandas data frame*

purchase\_data = pd.read\_csv(file\_to\_load)

**Player Count**

In [3]:

Player\_Totals = data\_file.loc[:,["Gender", "SN", "Age"]]

Player\_Totals = Player\_Totals.drop\_duplicates()

Number\_of\_Players = Player\_Totals.count()[0]

pd.DataFrame({"Total Players" : [Number\_of\_Players]})

total\_players = purchase\_data ['SN'].nunique()

total\_players\_df = pd.DataFrame({'Total Players': [total\_players]})

total\_players\_df

Out[3]:

|  | **Total Players** |
| --- | --- |
| **0** | 576 |

**Purchasing Analysis (Total)**

# Number of Unique Items

# Average Purchase Price

# Total Number of Purchases

# Total Revenue

In [4]:

unique\_items = len(purchase\_data['Item ID'].unique())

average\_price = purchase\_data ['Price'].mean()

num\_purchases = purchase\_data ['Purchase ID'].count()

revenue = purchase\_data ['Price'].sum()

summary\_table = pd.DataFrame({'Number of Unique Items': [unique\_items],

'Average Price': ['${:,.2f}'.format(average\_price)],

'Number of Purchases': [num\_purchases],

'Total Revenue': ['${:,.2f}'.format(revenue)]})

summary\_table

Out[4)]:

|  | **Number of Unique Items** | **Average Price** | **Number of Purchases** | **Total Revenue** |
| --- | --- | --- | --- | --- |
|  | 179 | $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 [5]:

male\_players = purchase\_data.loc[purchase\_data['Gender']== 'Male','SN'].nunique()

female\_players = purchase\_data.loc[purchase\_data['Gender']== 'Female','SN'].nunique()

other\_players = purchase\_data.loc[purchase\_data['Gender']== 'Other / Non-Disclosed','SN'].nunique()

male\_p = "{:.2%}".format(male\_players/purchase\_data['SN'].nunique())

female\_p = "{:.2%}".format(female\_players/purchase\_data['SN'].nunique())

other\_p = "{:.2%}".format(other\_players/purchase\_data['SN'].nunique())

gender\_summary\_table = pd.DataFrame({'':['Male', 'Female', 'Other / Non-Disclosed'],

'Total Count':[male\_players,female\_players,other\_players],

'Percentage of Players':[male\_p, female\_p, other\_p]})

gender\_summary\_table = gender\_summary\_table.set\_index('')

gender\_summary\_table

Out[5]:

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

**Purchasing Analysis (Gender)**

# Purchase Count

# Average Purchase Price

# Total Purchase Value

# Average Purchase Total per Person by Gender

In [6]:

g\_p = purchase\_data.groupby(["Gender"]).sum()["Price"]

g\_avg = purchase\_data.groupby(["Gender"]).mean()["Price"]

g\_counts = purchase\_data.groupby(["Gender"]).count()["Price"]

Avg\_p = g\_p / purchase\_data.groupby(["Gender"]).nunique()["SN"]

gender\_data = pd.DataFrame({"Purchase Count": g\_counts,

"Average Purchase Price": g\_avg.map("${:.2f}".format),

"Total Purchase Value": g\_p.map("${:.2f}".format),

"Avg Total Purchase per Person": Avg\_p.map("${:.2f}".format)})

gender\_data

Out[6]:

|  | **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 [7]:

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+"]

purchase\_data["Age Ranges"] = pd.cut(purchase\_data["Age"], age\_bins, labels=group\_names)

age\_demographics\_totals = purchase\_data.groupby(["Age Ranges"]).nunique()["SN"]

age\_demographics\_percents = age\_demographics\_totals / purchase\_data['SN'].nunique() \* 100

age\_demographics = pd.DataFrame({"Total Count": age\_demographics\_totals, "Percent of Players": age\_demographics\_percents})

age\_demographics = age\_demographics.sort\_index()

age\_demographics.round(2)

Out[7]:

|  | **Total Count** | **Percent of Players** |
| --- | --- | --- |
| **Age Ranges** |  |  |
| **<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 [8]:

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+"]

purchase\_data["Age Ranges"] = pd.cut(purchase\_data["Age"], age\_bins, labels=group\_names)

purchase\_count = purchase\_data.groupby(["Age Ranges"]).count()["Price"]

avg\_purchase = purchase\_data.groupby(["Age Ranges"]).mean()["Price"]

total\_purchase = purchase\_data.groupby(["Age Ranges"]).sum()["Price"]

avg\_purchase\_p = total\_purchase / purchase\_data.groupby(["Age Ranges"]).nunique()["SN"]

age\_p\_analysis = pd.DataFrame({"Purchase Count": purchase\_count,

"Average Purchase Price": avg\_purchase.map("${:.2f}".format),

"Total Purchase Value": total\_purchase.map("${:.2f}".format),

"Avg Total Purchase per Person": avg\_purchase\_p.map("${:.2f}".format)})

age\_p\_analysis = age\_p\_analysis.sort\_index()

age\_p\_analysis

Out[8]:

|  | **Purchase Count** | **Average Purchase Price** | **Total Purchase Value** | **Avg Total Purchase per Person** |
| --- | --- | --- | --- | --- |
| **Age Ranges** |  |  |  |  |
| **<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 [9]:

# Identify top 5 spenders in the game by total purchase value

# SN

# Purchase Count

# Average Purchase Price

# Total Purchase Value/Amount

user\_count = purchase\_data.groupby(["SN"]).count()["Price"]

user\_total = purchase\_data.groupby(["SN"]).sum()["Price"]

user\_average = purchase\_data.groupby(["SN"]).mean()["Price"]

user\_data = pd.DataFrame({"Purchase Count": user\_count,

"Total Purchase Amount": user\_total,

"Average Purchase Price": user\_average}).round(2)

user\_data.sort\_values("Total Purchase Amount", ascending=False).head()

Out[9]:

|  | **Purchase Count** | **Total Purchase Amount** | **Average Purchase Price** |
| --- | --- | --- | --- |
| **SN** |  |  |  |
| **Lisosia93** | 5 | 18.96 | 3.79 |
| **Idastidru52** | 4 | 15.45 | 3.86 |
| **Chamjask73** | 3 | 13.83 | 4.61 |
| **Iral74** | 4 | 13.62 | 3.40 |
| **Iskadarya95** | 3 | 13.10 | 4.37 |

**Most Popular Items**

In [10]:

Identify the 5 most popular items by purchase count

# Item ID

# Item Name

# Purchase Count

# Item Price

# Total Purchase Value

item\_count = purchase\_data.groupby(["Item ID","Item Name"]).count()["Price"]

item\_total = purchase\_data.groupby(["Item ID","Item Name"]).sum()["Price"]

item\_average = purchase\_data.groupby(["Item ID","Item Name"]).mean()["Price"]

item\_data = pd.DataFrame({"Purchase Count": item\_count,

"Item Price": item\_average.map("${:.2f}".format),

"Total Purchase Value": item\_total.map("${:.2f}".format)})

item\_data.sort\_values("Purchase Count", ascending=False).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 |
| **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 [11]:

# Identify the 5 most popular items by purchase count

# Item ID

# Item Name

# Purchase Count

# Item Price

# Total Purchase Value

item\_count = purchase\_data.groupby(["Item ID","Item Name"]).count()["Price"]

item\_total = purchase\_data.groupby(["Item ID","Item Name"]).sum()["Price"]

item\_average = purchase\_data.groupby(["Item ID","Item Name"]).mean()["Price"]

item\_data = pd.DataFrame({"Purchase Count": item\_count,

"Item Price": item\_average.map("${:.2f}".format),

"Total Purchase Value": item\_total.map("${:.2f}".format)})

item\_data.sort\_values("Purchase Count", ascending=False).head()

Out[11]:

|  |  | **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 |
| **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 ###

# Item Price

In [12]:

item\_data.sort\_values("Item Price", ascending=False).head()

|  |  | **Purchase Count** | **Item Price** | **Total Purchase Value** |
| --- | --- | --- | --- | --- |
| **Item ID** | **Item Name** |  |  |  |
| **63** | **Stormfury Mace** | 2 | $4.99 | $9.98 |
| **139** | **Mercy, Katana of Dismay** | 5 | $4.94 | $24.70 |
| **173** | **Stormfury Longsword** | 2 | $4.93 | $9.86 |
| **147** | **Hellreaver, Heirloom of Inception** | 3 | $4.93 | $14.79 |
| **128** | **Blazeguard, Reach of Eternity** | 5 | $4.91 | $24.55 |