Heroes Of Pymoli Data Analysis:

Below you will find a brief report that analyzes Heroes Of Pymoli purchasing data into relevant insights. Based on the purchase data collected, we have been able to find the following three observable trends:

1. The vast majority of the purchasers were men, accounting for 84% of the 576 players.
2. Almost two thirds of the players (63%) came from the 15-19 and 20-24 age groups, with the latter accounting for the largest share of the age group dempgraphics with approximately 45%.
3. Although men dominated purchases of the game, it is worthwhile highlighting the fact that the avgerage purchase price was higher for women ($3.20) than for men ($3.02), but that fact may regress to the mean with more purchases by women.

In [1]:

```
#Import dependencies
import pandas as pd
import numpy as np
```

In [2]:

```
# File to Load (Remember to Change These)
csv_path = "Resources/purchase_data.csv"

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

In [3]:

```
#Get a sense of the file
purchase_data_df.set_index('Purchase ID').head()
```

Out[3]:

	SN	Age	Gender	Item ID	Item Name	Price
Purchase ID						
0	Lisim78	20	Male	108	Extraction, Quickblade Of Trembling Hands	3.53
1	Lisovynya38	40	Male	143	Frenzied Scimitar	1.56
2	Ithergue48	24	Male	92	Final Critic	4.88
3	Chamassasya86	24	Male	100	Blindscythe	3.27
4	Iskosia90	23	Male	131	Fury	1.44

In [4]:

```
# A.PLAYER COUNT
#a1. Total Number of Players
player_count = len(purchase_data_df['SN'].unique())
player_count
```

Out[4]:

```
576
```

In [5]:

```
#B. PURCHASING ANALYSIS(TOTAL)
```

In [6]:

```
# B1. Number of Unique Items
number_items = len(purchase_data_df["Item ID"].unique())
number_items
```

Out[6]:

```
183
```

In [7]:

```
#b2. Average Purchase Price
avg_price = round(purchase_data_df["Price"].mean(), 2)
avg_price
```

Out[7]:

```
3.05
```

In [8]:

```
#b3. Total Number of Purchase
total_purchases = purchase_data_df["Purchase ID"].count()
total_purchases
```

Out[8]:

```
780
```

In [9]:

```
#b4. Total Revenue
total_revenue = round(total_purchases * avg_price, 2)
total_revenue
```

Out[9]:

```
2379.0
```

In [10]:

```
purchasing_analysis_total =  pd.DataFrame({"Total Unique Items": [number_items],
                                  "AVG. Purchase Price ": [avg_price],
                                  "Total Purchases": [total_purchases],
                                  "Total Revenue": [total_revenue]})

print('Purchasing Analysis')
print('________________________________________________________________________________')

print(purchasing_analysis_total)
```

```
Purchasing Analysis
________________________________________________________________________________
   Total Unique Items  AVG. Purchase Price   Total Purchases  Total Revenue
0                 183                 3.05               780         2379.0
```

In [11]:

```
#C. Gender Demographics
#c1.Male
avg_total_male_df = purchase_data_df.loc[purchase_data_df["Gender"] == 'Male']
total_male_purchasers = len(avg_total_male_df.SN.unique())
male_per = total_male_purchasers/player_count
print('TOTAL PLAYERS BY GENDER')
print('_______________________')
print('Total Male / Percentage Male(%)')
print('------------------------------')
print(total_male_purchasers)
print(round(male_per*100,2))
#c2. Female
avg_total_female_df = purchase_data_df.loc[purchase_data_df["Gender"] == 'Female']
total_female_purchasers = len(avg_total_female_df.SN.unique())
female_per = total_female_purchasers/player_count
print('------------------------------')
print('Total Female / Percentage Female(%)')
print(total_female_purchasers)
print(total_female_purchasers)
print(round(female_per *100,2))
#c3. Other
avg_total_other_df = purchase_data_df.loc[purchase_data_df["Gender"] == 'Other / Non-Disclosed']
total_other_purchasers = len(avg_total_other_df.SN.unique())
other_per = total_other_purchasers/player_count
print('------------------------------')
print('Total Other/Non_Disclosed / Percentage Other/Non-Disclosed(%)')
print(total_other_purchasers)
print(round(other_per *100,2))
```

```
TOTAL PLAYERS BY GENDER
_______________________
Total Male / Percentage Male(%)
-----------------------------
484
84.03
-----------------------------
Total Female / Percentage Female(%)
81
14.06
------------------------------
Total Other/Non_Disclosed / Percentage Other/Non-Disclosed(%)
11
1.91
```

In [12]:

```
#D.Purchasing Analysis(Gender)
#d1. Purchase Count per Gender
purchase_count = purchase_data_df['Gender'].value_counts()
purchase_count
```

Out[12]:

```
Male                     652
Female                   113
Other / Non-Disclosed     15
Name: Gender, dtype: int64
```

In [13]:

```
#d2. Avg purchase price
grouped_gender_df = purchase_data_df.groupby("Gender")
grouped_gender_df.Price.mean()
```

Out[13]:

```
Gender
Female                   3.203009
Male                     3.017853
Other / Non-Disclosed    3.346000
Name: Price, dtype: float64
```

In [14]:

```
#d3. Total Purchase Value
grouped_gender_df.Price.sum()
```

Out[14]:

```
Gender
Female                    361.94
Male                     1967.64
Other / Non-Disclosed      50.19
Name: Price, dtype: float64
```

In [15]:

```
#d4. Average Purchase Total per Person by gender
#a. Male
print(round(1967.64/total_male_purchasers,2))
```

```
#b. Female
print(round(361.94/total_female_purchasers,2))
#c. Other
print(round(50.19/total_other_purchasers, 2))
```

```
4.07
4.47
4.56
```

In [16]:

```
#E.Age Demographics
```

In [17]:

```
# Find the range of Age for binnung
purchase_data_df.Age.max()
```

Out[17]:

```
45
```

In [18]:

```
purchase_data_df.Age.min()
```

Out[18]:

```
7
```

In [19]:

```
bins=(0,9,14,19,24,29,34,39,50)
group_labels = ["0 > 10", "10 to 14", "15 to 19", "20 to 24", "25 to 29", "30 to 34",
               "35 to 39", "40 to 50"]
pd.cut(purchase_data_df["Age"], bins, labels=group_labels).head()

#purchase_data['Age Range'] = pd.cut(purchase_data['Age'], bins, labels=group_names)
```

Out[19]:

```
0    20 to 24
1    40 to 50
2    20 to 24
3    20 to 24
4    20 to 24
Name: Age, dtype: category
Categories (8, object): [0 > 10 < 10 to 14 < 15 to 19 < 20 to 24 < 25 to 29 < 30 to 34 < 35 to 39
< 40 to 50]
```

In [20]:

```
purchase_data_df["Age_Group"] = pd.cut(purchase_data_df["Age"], bins, labels=group_labels)
purchase_data_df
```

Out[20]:

	Purchase ID	SN	Age	Gender	Item ID	Item Name	Price	Age_Group
0	0	Lisim78	20	Male	108	Extraction, Quickblade Of Trembling Hands	3.53	20 to 24
1	1	Lisovynya38	40	Male	143	Frenzied Scimitar	1.56	40 to 50
2	2	Ithergue48	24	Male	92	Final Critic	4.88	20 to 24
3	3	Chamassasya86	24	Male	100	Blindscythe	3.27	20 to 24
4	4	Iskosia90	23	Male	131	Fury	1.44	20 to 24
...	...	...	...	...	...	...	...	...
775	775	Aethedru70	21	Female	60	Wolf	3.54	20 to 24

	Purchase ID	SN	Age	Gender	Item ID	Item Name	Price	Age_Group
776	776	Iral74	21	Male	164	Exiled Doomblade	1.63	20 to 24
777	777	Yathecal72	20	Male	67	Celeste, Incarnation of the Corrupted	3.46	20 to 24
778	778	Sisur91	7	Male	101	Final Critic	4.19	0 > 10
779	779	Ennrian78	24	Male	50	Dawn	4.60	20 to 24

780 rows × 8 columns

In [36]:

```
#1. Purchase_count
age_purchase_count = purchase_data_df.groupby(["Age_Group"])["SN"].count()
age_purchase_count
```

Out[36]:

```
Age_Group
0 > 10       23
10 to 14     28
15 to 19    136
20 to 24    365
25 to 29    101
30 to 34     73
35 to 39     41
40 to 50     13
Name: SN, dtype: int64
```

In [37]:

```
#2. Avg purchase price
age_purchase_price = purchase_data_df.groupby(["Age_Group"])["Price"].mean()
age_purchase_price
```

Out[37]:

```
Age_Group
0 > 10      3.353478
10 to 14    2.956429
15 to 19    3.035956
20 to 24    3.052219
25 to 29    2.900990
30 to 34    2.931507
35 to 39    3.601707
40 to 50    2.941538
Name: Price, dtype: float64
```

In [38]:

```
#3. Total Purchase Value
age_purchase_value = age_purchase_count * age_purchase_price
age_purchase_value
```

Out[38]:

```
Age_Group
0 > 10        77.13
10 to 14      82.78
15 to 19     412.89
20 to 24    1114.06
25 to 29     293.00
30 to 34     214.00
35 to 39     147.67
40 to 50      38.24
dtype: float64
```

In [39]:

```
#4. Avg Purchase total per person
age_pp = purchase_data_df.groupby(["Age_Group"])["SN"].nunique()
age_purchase_value/age_pp
```

Out[39]:

```
Age_Group
0 > 10      4.537059
10 to 14    3.762727
15 to 19    3.858785
20 to 24    4.318062
25 to 29    3.805195
30 to 34    4.115385
35 to 39    4.763548
40 to 50    3.186667
dtype: float64
```

In [41]:

```
#Summary Table (Age Purchasing Analysis)
age_table = pd.concat([age_purchase_count, round(age_purchase_price,2), round(age_purchase_value,2)
, round(age_purchase_value/age_pp,2)], axis=1)
age_table.columns = ['Purchase Count', '($) AVG Purchase Price',  '($) Total Purchase Value',  '($
) AVG Total Purchase Per Person']
age_table
```

Out[41]:

	Purchase Count	($) AVG Purchase Price	($) Total Purchase Value	($) AVG Total Purchase Per Person
Age_Group				
0 > 10	23	3.35	77.13	4.54
10 to 14	28	2.96	82.78	3.76
15 to 19	136	3.04	412.89	3.86
20 to 24	365	3.05	1114.06	4.32
25 to 29	101	2.90	293.00	3.81
30 to 34	73	2.93	214.00	4.12
35 to 39	41	3.60	147.67	4.76
40 to 50	13	2.94	38.24	3.19

In [43]:

```
age_total_count = purchase_data_df.groupby(["Age_Group"])["SN"].nunique()
age_total_count
```

Out[43]:

```
Age_Group
0 > 10       17
10 to 14     22
15 to 19    107
20 to 24    258
25 to 29     77
30 to 34     52
35 to 39     31
40 to 50     12
Name: SN, dtype: int64
```

In [54]:

```
#Summary Table (Age Demograhics)
new_table = pd.concat([age_total_count, round(age_total_count/player_count*100,2)], axis=1)
new_table.columns = ['Total Count', '(%) Percentage of Players']
new_table
```

Out[54]:

	Total Count	(%) Percentage of Players

Age_Group	Total Count	(%) Percentage of Players
Age_Group		
0 > 10	17	2.95
10 to 14	22	3.82
15 to 19	107	18.58
20 to 24	258	44.79
25 to 29	77	13.37
30 to 34	52	9.03
35 to 39	31	5.38
40 to 50	12	2.08

In [26]:

```
#F. Top Spenders
purchase_data_df['Item ID'].nunique()
```

Out[26]:

```
183
```

In [27]:

```
#Groupby SN and break down relevant data
ts_spenders = purchase_data_df.groupby('SN')
ts_purchase_count = ts_spenders['Purchase ID'].nunique()
ts_average_purchase_price = ts_spenders['Price'].mean()
ts_total_purchase_value = ts_spenders['Price'].sum()
```

In [28]:

```
# create columns
ts_spender_table = pd.concat([ts_purchase_count, ts_average_purchase_price,
ts_total_purchase_value], axis=1)
ts_spender_table.columns = ['Purchase Count', 'Average Purchase Price', 'Total Purchase Value']
ts_spender_table.sort_values(['Total Purchase Value'], ascending = False, inplace=True)
```

In [29]:

```
#format table
ts_spender_table["Average Purchase Price"] = ts_spender_table["Average Purchase
Price"].map("${:.2f}".format)
ts_spender_table["Total Purchase Value"] = ts_spender_table["Total Purchase Value"].map("${:.2f}".f
ormat)
ts_spender_table.head()
```

Out[29]:

	Purchase Count	Average Purchase Price	Total Purchase Value
SN			
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

In [30]:

```
#H. MOST PROFITABLE ITEMS(note: changed order with G.)
#Groupby Item Name and break down
profitable_items = purchase_data_df.groupby('Item Name')
item_id = profitable_items['Item ID'].unique()
item_name = profitable_items['Item Name'].nunique()
item_purchase_count = profitable_items['Purchase ID'].count()
```

```
item_price = profitable_items['Price'].mean()
item_purchase_value = profitable_items['Price'].sum()
```

In [31]:

```
#Create the table
item_table = pd.concat([item_id, item_purchase_count, item_price, item_purchase_value], axis=1)
item_table.columns = ['Item ID', 'Purchase Count', 'Item Price', 'Total Purchase Value']
item_table.sort_values(['Total Purchase Value'], ascending = False, inplace=True)
#item_table = item_table.drop_duplicates(subset='Item ID', keep='first')
```

In [32]:

```
#Format the table
item_table["Item Price"] = item_table["Item Price"].map("${:.2f}".format)
item_table["Total Purchase Value"] = item_table["Total Purchase Value"].map("${:.2f}".format)
item_table.head()
```

Out[32]:

	Item ID	Purchase Count	Item Price	Total Purchase Value
Item Name				
Final Critic	[92, 101]	13	$4.61	$59.99
Oathbreaker, Last Hope of the Breaking Storm	[178]	12	$4.23	$50.76
Nirvana	[82]	9	$4.90	$44.10
Fiery Glass Crusader	[145]	9	$4.58	$41.22
Singed Scalpel	[103]	8	$4.35	$34.80

In [33]:

```
#G. Most Popular Items
item_table.sort_values(['Purchase Count'], ascending = False, inplace=True)
item_table.head()
```

Out[33]:

	Item ID	Purchase Count	Item Price	Total Purchase Value
Item Name				
Final Critic	[92, 101]	13	$4.61	$59.99
Oathbreaker, Last Hope of the Breaking Storm	[178]	12	$4.23	$50.76
Nirvana	[82]	9	$4.90	$44.10
Fiery Glass Crusader	[145]	9	$4.58	$41.22
Extraction, Quickblade Of Trembling Hands	[108]	9	$3.53	$31.77