Heroes Of Pymoli Data Analysis

- Of the 1163 active players, the vast majority are male (84%). There also exists, a smaller, but notable proportion of female players (14%).
- Our peak age demographic falls between 20-24 (44.8%) with secondary groups falling between 15-19 (18.60%) and 25-29 (13.4%).

Note

- Instructions have been included for each segment. You do not have to follow them exactly, but they are included to help you think through the steps.

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

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

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

Player Count

- Display the total number of players

In [2]:
```
# Display Length for total number of players
total_players = len(purchase_data["SN"].value_counts())

# Create a data frame showing total players
player_count = pd.DataFrame({"Total Players":[total_players]})
player_count
```

Out[2]:

	Total Players
0	576

Purchasing Analysis (Total)

- Run basic calculations to obtain number of unique items, average price, etc.
- Create a summary data frame to hold the results
- Optional: give the displayed data cleaner formatting
- Display the summary data frame

In [3]:
```
# Calculate to get the number of unique items, average price, purchase count,
 and revenue
number_of_unique_items = len((purchase_data["Item ID"]).unique())
average_price = (purchase_data["Price"]).mean()
number_of_purchases = (purchase_data["Purchase ID"]).count()
total_revenue = (purchase_data["Price"]).sum()

# Data frame for number of unique items, average price, number of purchases, a
nd total revenue
summary_df = pd.DataFrame({"Number of Unique Items":[number_of_unique_items],
                           "Average Price":[average_price],
                           "Number of Purchases": [number_of_purchases],
                           "Total Revenue": [total_revenue]})

# Generate format for currency style
summary_df.style.format({'Average Price':"${:,.2f}",
                         'Total Revenue': '${:,.2f}'})
```

Out[3]:

	Number of Unique Items	Average Price	Number of Purchases	Total Revenue
0	183	$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 [4]:
```
# Groupby purchase_data by Gender
gender_stats = purchase_data.groupby("Gender")

# Total count of screen names "SN" by gender
total_count_gender = gender_stats.nunique()["SN"]

# Total count by gender divivded by total players
percentage_of_players = total_count_gender / total_players * 100

# Data frame for Percentage of Players and Total of Gender
gender_demographics = pd.DataFrame({"Percentage of Players": percentage_of_pla
yers, "Total Count": total_count_gender})

# Data frame format no index name at the corner
gender_demographics.index.name = None

# Sort the values by total count in descending order. Percentage in two decima
l places
gender_demographics.sort_values(["Total Count"], ascending = False).style.form
at({"Percentage of Players":"{:.2f}"})
```

Out[4]:

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

Purchasing Analysis (Gender)

- Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase total per person etc. by gender
- Create a summary data frame to hold the results
- Optional: give the displayed data cleaner formatting
- Display the summary data frame

In [5]:

```
# Total count of purchases by gender
purchase_count = gender_stats["Purchase ID"].count()

# Avg. Purchase Prices by gender
avg_purchase_price = gender_stats["Price"].mean()

# Avg. Purchase Total by gender
avg_purchase_total = gender_stats["Price"].sum()

# Avg. Purchase Total by gender divivded by purchase count by unique shoppers
avg_purchase_per_person = avg_purchase_total/total_count_gender

# Data frame for purchase cound, avg. purchase price, avg. purchase value, and
avg. purchase total per person
gender_demographics = pd.DataFrame({"Purchase Count": purchase_count,
                                    "Average Purchase Price": avg_purchase_pri
ce,
                                    "Average Purchase Value":avg_purchase_tota
l,
                                    "Avg Purchase Total per Person": avg_purch
ase_per_person})

# Index in top left as "Gender"
gender_demographics.index.name = "Gender"

# Generate format for currency style
gender_demographics.style.format({"Average Purchase Value":"${:,.2f}",
                                  "Average Purchase Price":"${:,.2f}",
                                  "Avg Purchase Total per Person":"${:,.2f}"})
```

Out[5]:

	Purchase Count	Average Purchase Price	Average Purchase Value	Avg Purchase Total 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

- Establish bins for ages
- Categorize the existing players using the age bins. Hint: use pd.cut()
- Calculate the numbers and percentages by age group
- Create a summary data frame to hold the results
- Optional: round the percentage column to two decimal points
- Display Age Demographics Table

In [6]:
```
# Bins for ages
age_bins = [0, 9.90, 14.90, 19.90, 24.90, 29.90, 34.90, 39.90, 99999]
group_names = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "4
0+"]

# Group and sort age values into bins noted above
purchase_data["Age Group"] = pd.cut(purchase_data["Age"],age_bins, labels=grou
p_names)
purchase_data

# Data frame for added "Age Group" and groupby
age_grouped = purchase_data.groupby("Age Group")

# Total count of players by age category
total_count_age = age_grouped["SN"].nunique()

# Percentages by age category
percentage_by_age = (total_count_age/total_players) * 100

# Data frame for Percentage of Players and Total Count of Age
age_demographics = pd.DataFrame({"Percentage of Players": percentage_by_age,
"Total Count": total_count_age})

# Data frame format no index name at the corner
age_demographics.index.name = None

# Percentage in two decimal places
age_demographics.style.format({"Percentage of Players":"{:,.2f}"})
```

Out[6]:

	Percentage of Players	Total Count
<10	2.95	17
10-14	3.82	22
15-19	18.58	107
20-24	44.79	258
25-29	13.37	77
30-34	9.03	52
35-39	5.38	31
40+	2.08	12

Purchasing Analysis (Age)

- Bin the purchase_data data frame by age
- Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase total per person etc. in the table below
- Create a summary data frame to hold the results
- Optional: give the displayed data cleaner formatting
- Display the summary data frame

In [7]:
```
# Count of purchases by age group
purchase_count_age = age_grouped["Purchase ID"].count()

# Avg. purchase price by age group
avg_purchase_price_age = age_grouped["Price"].mean()

# Total purchase value by age group
total_purchase_value = age_grouped["Price"].sum()

# Avg. purchase per person in the age group
avg_purchase_per_person_age = total_purchase_value/total_count_age

# Data frame for purchase count, avg. purchase price, total purchase value, an
d avg. purchase total per person.
age_demographics = pd.DataFrame({"Purchase Count": purchase_count_age,
                                 "Average Purchase Price": avg_purchase_price_
age,
                                 "Total Purchase Value":total_purchase_value,
                                 "Average Purchase Total per Person": avg_purc
hase_per_person_age})

# Data frame format no index name at the corner
age_demographics.index.name = None

# Generate format for currency style
age_demographics.style.format({"Average Purchase Price":"${:,.2f}",
                               "Total Purchase Value":"${:,.2f}",
                               "Average Purchase Total per Person":"${:,.2f}"
})
```

Out[7]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Average Purchase Total per Person
<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

- Run basic calculations to obtain the results in the table below
- Create a summary data frame to hold the results
- Sort the total purchase value column in descending order
- Optional: give the displayed data cleaner formatting
- Display a preview of the summary data frame

In [8]:
```
# Groupby purchase data by screen names
spender_stats = purchase_data.groupby("SN")

# Total count of purchases by name
purchase_count_spender = spender_stats["Purchase ID"].count()

# Avg. purchase by name
avg_purchase_price_spender = spender_stats["Price"].mean()

# Purchase total
purchase_total_spender = spender_stats["Price"].sum()

# Data frame for purchase count, avg. purchase price, and total purchase valu
e.
top_spenders = pd.DataFrame({"Purchase Count": purchase_count_spender,
                             "Average Purchase Price": avg_purchase_price_spen
der,
                             "Total Purchase Value":purchase_total_spender})

# Sort by descending order to generate the top 5 spender names
formatted_spenders = top_spenders.sort_values(["Total Purchase Value"], ascend
ing=False).head()

# Generate format for currency style
formatted_spenders.style.format({"Average Purchase Total":"${:,.2f}",
                                 "Average Purchase Price":"${:,.2f}",
                                 "Total Purchase Value":"${:,.2f}"})
```

Out[8]:

	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

Most Popular Items

- Retrieve the Item ID, Item Name, and Item Price columns
- Group by Item ID and Item Name. Perform calculations to obtain purchase count, item price, and total purchase value
- Create a summary data frame to hold the results
- Sort the purchase count column in descending order
- Optional: give the displayed data cleaner formatting
- Display a preview of the summary data frame

In [9]:
```
# Data frame for item id, item name, and price
items = purchase_data[["Item ID", "Item Name", "Price"]]

# Groupby item id and item name
item_stats = items.groupby(["Item ID","Item Name"])

# Number of times an item has been purchased
purchase_count_item = item_stats["Price"].count()

# Purchase value per item
purchase_value = (item_stats["Price"].sum())

# Identify item price
item_price = purchase_value/purchase_count_item

# Data frame for purchase count, item price, and total purchase value
most_popular_items = pd.DataFrame({"Purchase Count": purchase_count_item,
                                   "Item Price": item_price,
                                   "Total Purchase Value":purchase_value})

# Sort by descending order to generate top spender names and top 5 item names
popular_formatted = most_popular_items.sort_values(["Purchase Count"], ascendi
ng=False).head()

# Generate format for currency style
popular_formatted.style.format({"Item Price":"${:,.2f}",
                                "Total Purchase Value":"${:,.2f}"})
```

Out[9]:

		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

- Sort the above table by total purchase value in descending order
- Optional: give the displayed data cleaner formatting
- Display a preview of the data frame

```
In [10]: # Get the most_popular items data frame then change the sorting to identify th
         e highest total purchase value
         popular_formatted = most_popular_items.sort_values(["Total Purchase Value"],
                                                            ascending=False).head()
         # Generate format for currency style
         popular_formatted.style.format({"Item Price":"${:,.2f}",
                                         "Total Purchase Value":"${:,.2f}"})
```

Out[10]:

		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
92	**Final Critic**	8	$4.88	$39.04
103	**Singed Scalpel**	8	$4.35	$34.80