

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 = "purchase_data.csv"

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

Out[1]:

	Purchase ID	SN	Age	Gender	Item ID	Item Name	Price
0	0	Lisim78	20	Male	108	Extraction, Quickblade Of Trembling Hands	3.53
1	1	Lisovynya38	40	Male	143	Frenzied Scimitar	1.56
2	2	lthergue48	24	Male	92	Final Critic	4.88
3	3	Chamassasya86	24	Male	100	Blindscythe	3.27
4	4	Iskosia90	23	Male	131	Fury	1.44

Player Count

- Display the total number of players

```
In [2]: Total_Players = pd.DataFrame({"Total Players": [purchase_data["SN"].unique().size,
Total_Players
```

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]: unique_items = purchase_data["Item ID"].unique().size
average_price = "${:,.2f}".format(purchase_data["Price"].mean())
purchases = purchase_data["Purchase ID"].size
revenue = "${:,.2f}".format(purchase_data["Price"].sum())

Summary = pd.DataFrame({"Number of Unique Items": [unique_items],
                        "Average Price": [average_price],
                        "Number of Purchases": [purchases],
                        "Total Revenue": [revenue]})

Summary
```

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]: new_index = ["Male", "Female", "Other / Non-Disclosed"]

total_count = purchase_data.nunique()["SN"]
male_count = purchase_data[purchase_data["Gender"] == "Male"]["SN"].nunique()
female_count = purchase_data[purchase_data["Gender"] == "Female"]["SN"].nunique()
others_count = purchase_data[purchase_data["Gender"] == "Other / Non-Disclosed"]

gender_df = pd.DataFrame({
    "": new_index,
    "Total Count": [male_count, female_count, others_count],
    "Percentage of Players": [
        "{:,.2f}".format((male_count / total_count * 100)),
        "{:,.2f}".format((female_count / total_count * 100)),
        "{:,.2f}".format((others_count / total_count * 100))]})

gender_df = gender_df.set_index("")
gender_df

```

Out[4]:

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

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]: grouped_gender = purchase_data.groupby("Gender")

purchase_count = grouped_gender["Purchase ID"].count()
ave_purchase_price = grouped_gender["Price"].mean()
total_purchase_value = grouped_gender["Price"].sum()
ave_per_person = total_purchase_value / grouped_gender.nunique()["SN"]

gender_demo = pd.DataFrame({
    "Purchase Count": purchase_count,
    "Average Purchase Price": ave_purchase_price,
    "Total Purchase Value" : total_purchase_value,
    "Avg Total Purchase per Person": ave_per_person})

gender_demo.style.format({
    "Average Purchase Price": "${:,.2f}",
    "Total Purchase Value" : "${:,.2f}",
    "Avg Total Purchase per Person": "${:,.2f}"})
```

Out[5]:

	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

- 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 = [0, 9, 14, 19, 24, 29, 34, 39, 50]
group_labels = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40-44"]
purchase_data[""] = pd.cut(purchase_data["Age"], bins, labels=group_labels)

index = purchase_data.groupby("")
age_count = index['SN'].nunique()

age_demo = pd.DataFrame({
    "Total Count": age_count,
    "Percentage of Players": (age_count / total_count) * 100})

age_demo.style.format({"Percentage of Players": "{:,.2f}"})
```

Out[6]:

	Total Count	Percentage of Players
<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)

- 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]: age_demo2 = pd.DataFrame({
    "Purchase Count": index["Purchase ID"].count(),
    "Average Purchase Price": index["Price"].mean(),
    "Total Purchase Value" : index["Price"].sum(),
    "Avg Total Purchase per Person": index["Price"].sum() / age_count})

age_demo2.style.format({
    "Average Purchase Price": "${:,.2f}",
    "Total Purchase Value" : "${:,.2f}",
    "Avg Total Purchase per Person": "${:,.2f}"})
```

Out[7]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase 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]: grouped_SN = purchase_data.groupby("SN")

top_spenders = pd.DataFrame({
    "Purchase Count": grouped_SN["Purchase ID"].count(),
    "Average Purchase Price": grouped_SN["Price"].mean(),
    "Total Purchase Value" : grouped_SN["Price"].sum()})

descending = top_spenders.sort_values(["Total Purchase Value"], ascending=False)

descending.style.format({
    "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 [31]: retrieve_data = purchase_data[["Item ID", "Item Name", "Price"]]

most_popular = retrieve_data.groupby(["Item ID", "Item Name"])

most_popular = pd.DataFrame({
    "Purchase Count": most_popular["Price"].count(),
    "Item Price": most_popular["Price"].unique(),
    "Total Purchase Value": most_popular["Price"].sum()})

most_popular = most_popular.sort_values(["Purchase Count"], ascending=False).head(5)

most_popular.style.format({"Total Purchase Value": "${:,.2f}"})
```

Out[31]:

		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	[4.9]	\$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 [32]: most_popular_value = most_popular.sort_values(["Total Purchase Value"], ascending=False)
most_popular_value.style.format({"Total Purchase Value": "${:,.2f}"})
```

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	[4.9]	\$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

In []: