

Danica Rios

Heroes Of Pymoli Data Analysis

- The majority of purchasers are actually repeat purchasers for the Heroes of Pymoli game. The total purchase count is 780 but our unique players is 576.
- Most purchases are made by individuals aged 15-24, which is 44.79% of the overall Total Count. This age group also has the highest total purchase value of 1,114.06 USD and their average purchase price is 3.05 USD.
- Lastly, the majority of players are male, representing 84.03%, but their average purchase per person is lower than females and the other bucket. Males have an average purchase per person of 4.07 USD and females have an average purchase per person 4.47 USD. Therefore, the purchase price for females is roughly 9.82% more than the purchase price of males.

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]: ▶ # Use the Length of the "SN" List for total unique players
total_players = len(purchase_data["SN"].unique())

# Create a data frame with total players named players count
players_count = pd.DataFrame({"Total Players": [total_players]})
players_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]: ► number_items = len(purchase_data["Item ID"].unique())
average_price = (purchase_data["Price"]).mean()
purchases = purchase_data["Purchase ID"].count()
revenue = purchase_data["Price"].sum()

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

purchasing_analysis["Average Price"] = purchasing_analysis["Average Price"].map("${0:,.
purchasing_analysis["Total Revenue"] = purchasing_analysis["Total Revenue"].map("${0:,.

final_purchase_analysis = pd.DataFrame(purchasing_analysis)
final_purchase_analysis
```

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]: # Counts all players but does not consider duplicates  
gender_count = purchase_data.groupby("Gender")  
  
# Finds gender of unique players  
gender_unique_count = gender_count.nunique()["SN"]  
  
# Percentage of Players  
gender_percent = (gender_unique_count/total_players)*100  
  
# Creates data frame  
gender_summary = pd.DataFrame({"Total Count": gender_unique_count, "Percentage of Playe"  
  
# Format Results  
gender_summary.sort_values(["Total Count"], ascending=False).style.format({"Percentage of
```

Out[4]:

	Total Count	Percentage of Players
Gender		
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]: # Purchase Count by Gender
purchase_count = gender_count["Purchase ID"].count()

# Average Purchase Price by Gender
mean_purchase_price = gender_count["Price"].mean()

# Total Purchase Value by Gender
total_purchase_value = gender_count["Price"].sum()

# Average Total Purchase per Person
mean_total_purchase = total_purchase_value/gender_unique_count

# Data Frame
purchasing_analysis = pd.DataFrame({"Purchase Count": purchase_count,
                                     "Average Purchase Price": mean_purchase_price,
                                     "Total Purchase Value": total_purchase_value,
                                     "Avg Total Purchase per Person": mean_total_purchase})

# Format Results
purchasing_analysis.style.format({"Average Purchase Price": "${:.2f}", "Total Purchase Value": "${:.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]: ► age_bins = [0, 9, 14, 19, 24, 29, 34, 39, 100]
bin_groups = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]

purchase_data["Age Group"] = pd.cut(purchase_data["Age"], age_bins, labels=bin_groups)
purchase_data

group_by_age = purchase_data.groupby("Age Group")
count_age = group_by_age["SN"].nunique()

age_demo_percents = (count_age/total_players) * 100

age_demo_summary = pd.DataFrame({"Total Count": count_age, "Percentage of Players": age_demo_summary.style.format({"Percentage of Players": "{:.2f}%"})})
```

Out[6]:

	Total Count	Percentage of Players
Age Group		
<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]: # Bin the purchase_data data frame by age
purchase_data["Age Ranges"] = pd.cut(purchase_data["Age"], age_bins, labels=bin_groups)

# Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase total
age_purchase_total = purchase_data.groupby(["Age Ranges"]).sum()["Price"].rename("Total Purchase Value")
age_average = purchase_data.groupby(["Age Ranges"]).mean()["Price"].rename("Average Purchase Price")
age_count = purchase_data.groupby(["Age Ranges"]).count()["Price"].rename("Purchase Count")

# ***This calculation is incorrect!
avg_purchase_person = age_purchase_total/count_age

# Data Frame
age_purchase_analysis = pd.DataFrame({"Purchase Count": age_count, "Average Purchase Price": age_average,
                                     "Total Purchase Value": age_purchase_total,
                                     "Avg Total Purchase per Person": avg_purchase_person})

# Optional: Data Frame with optional formatting
age_purchase_analysis.style.format({"Average Purchase Price": "${:,.2f}", "Total Purchase Value": "${:,.2f}"})
```

Out[7]:

	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	\$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]: # Basic Calculations
sn_purchase_count = purchase_data.groupby(["SN"]).count()["Price"].rename("Purchase Count")
sn_avg_purchase_price = purchase_data.groupby(["SN"]).mean()["Price"].rename("Average Purchase Price")
sn_total_purchase_value = purchase_data.groupby(["SN"]).sum()["Price"].rename("Total Purchase Value")

# Create Data Frame
top_spenders_summary = pd.DataFrame({"Purchase Count": sn_purchase_count, "Average Purchase Price": sn_avg_purchase_price, "Total Purchase Value": sn_total_purchase_value})

# Sorts data by Total Purchase Value in descending order, adds cleaner formatting, and
top_spenders_summary.sort_values(["Total Purchase Value"], ascending=False).head(5).style
```

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]: # Retrieve the Item ID, Item Name, and Item Price columns
popular_items = purchase_data.loc[:,["Item ID", "Item Name", "Price"]]

# Group by Item ID and Item Name. Perform calculations to obtain purchase count, item price, and total purchase value
sum_items_purchase = popular_items.groupby(["Item ID", "Item Name"]).sum()["Price"].reset_index()
avg_item_price = popular_items.groupby(["Item ID", "Item Name"]).mean()["Price"].reset_index()
count_item_purchase = popular_items.groupby(["Item ID", "Item Name"]).count()["Price"].reset_index()

# Create Data Frame
most_popular_items = pd.DataFrame({"Purchase Count": count_item_purchase["Price"],
                                   "Item Price": avg_item_price["Price"],
                                   "Total Purchase Value": sum_items_purchase["Price"]})

# Sorts data by Total Purchase Value in descending order, adds cleaner formatting, and displays the top 5 items
most_popular_items.sort_values(["Purchase Count"], ascending=False).head(5).style.format
```

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]: most_popular_items.sort_values(["Total Purchase Value"], ascending=False).head(5).style
```

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

