

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]: 1 # Dependencies and Setup
2 import pandas as pd
3 import numpy as np
4
5 # File to Load (Remember to Change These)
6 file_to_load = "Resources/purchase_data.csv"
7
8 # Read Purchasing File and store into Pandas data frame
9 purchase_data = pd.read_csv(file_to_load)
10 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	Ithergue48	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]: 1 players = purchase_data['SN'].unique()
2 player_no = len(players)
3 player_df = pd.DataFrame({"Total Players": [player_no]})
4 player_df
```

```
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]: 1 items = purchase_data['Item ID'].unique()
2 items_no = len(items)
3 avg_price = purchase_data['Price'].mean()
4 purchase_no = purchase_data['Purchase ID'].count()
5 total_rev = purchase_data['Price'].sum()
6 analysis_df = pd.DataFrame({"Number of Unique Items":items_no,
7                             "Average Price":avg_price,
8                             "Number of Purchases":purchase_no,
9                             "Total Revenue":total_rev
10                            })
11 analysis_df["Average Price"] = analysis_df["Average Price"].map("${:,.2f}".format)
12 analysis_df["Total Revenue"] = analysis_df["Total Revenue"].map("${:,.2f}".format)
13 analysis_df
14
```

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]: 1 gendercounts = purchase_data.drop_duplicates('SN')['Gender'].value_counts()
2 percentage = gendercounts.div(player_no)
3
4 gender_df = pd.DataFrame({"Total Count":gendercounts,
5                           "Percentage of Players":percentage
6                           })
7 gender_df["Percentage of Players"] = gender_df["Percentage of Players"].map("{:.2%}".format)
8 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]: 1 p_grouped = purchase_data.groupby("Gender")
2 p_count = p_grouped['Purchase ID'].count()
3 p_mean = p_grouped['Price'].mean()
4 p_total = p_grouped['Price'].sum()
5 p_avgsum = p_total/gendercounts
6
7 gender_analysis_df = pd.DataFrame({"Purchase Count":p_count,
8                                   "Average Purchase Price":p_mean,
9                                   "Total Purchase Value":p_total,
10                                  "Avg Total Purchase per Person":p_avgsum
11                                  })
12 gender_analysis_df["Average Purchase Price"]=gender_analysis_df["Average Purchase Price"].map("{:,.2f}".format)
13 gender_analysis_df["Total Purchase Value"]=gender_analysis_df["Total Purchase Value"].map("{:,.2f}".format)
14 gender_analysis_df["Avg Total Purchase per Person"]=gender_analysis_df["Avg Total Purchase per Person"].map("{:,.2f}".format)
15 gender_analysis_df

```

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]: 1 bins = [0, 9, 14, 19, 24, 29, 34, 39, 200]
2 group_labels = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]
3
4 purchase_data["Age Group"] = pd.cut(purchase_data["Age"], bins, labels=group_labels)
5 unique_players_df = purchase_data.drop_duplicates("SN")
6
7 agecount = unique_players_df.groupby("Age Group")["SN"].count()
8 agepercentage = agecount/player_no
9
10 age_df = pd.DataFrame({"Total Count":agecount, "Percentage of Players":agepercentage})
11
12 age_df["Percentage of Players"]=age_df["Percentage of Players"].map("{:.2%}".format)
13 age_df
14
15

```

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]:

```

1 p_age_df = purchase_data.groupby("Age Group")
2 p_age_count = p_age_df['Purchase ID'].count()
3 p_age_mean = p_age_df['Price'].mean()
4 p_age_total = p_age_df['Price'].sum()
5 p_age_avgsum = p_age_total/agecount
6
7 age_analysis_df = pd.DataFrame({"Purchase Count":p_age_count,
8                                "Average Purchase Price":p_age_mean,
9                                "Total Purchase Value":p_age_total,
10                               "Avg Total Purchase per Person":p_age_avgsum
11                               })
12
13 age_analysis_df["Average Purchase Price"]=age_analysis_df["Average Purchase Price"].map("${:,.2f}".format)
14 age_analysis_df["Total Purchase Value"]=age_analysis_df["Total Purchase Value"].map("${:,.2f}".format)
15 age_analysis_df["Avg Total Purchase per Person"]=age_analysis_df["Avg Total Purchase per Person"].map("${:,.2f}".format)
16 age_analysis_df

```

Out[7]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
Age Group				
<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 [27]: 1 p_spend_df = purchase_data.groupby("SN")
2
3 p_spend_count = p_spend_df['Purchase ID'].count()
4 p_spend_mean = p_spend_df['Price'].mean()
5 p_spend_total = p_spend_df['Price'].sum()
6
7 topspenders_df = pd.DataFrame({"Purchase Count":p_spend_count,
8                               "Average Purchase Price":p_spend_mean,
9                               "Total Purchase Value":p_spend_total
10                              })
11
12 topspenders_df = topspenders_df.sort_values(by="Total Purchase Value", ascending=False)
13 topspenders_df["Average Purchase Price"] = topspenders_df["Average Purchase Price"].map("${:,.2f}".format)
14 topspenders_df["Total Purchase Value"] = topspenders_df["Total Purchase Value"].map("${:,.2f}".format)
15
16 topspenders_df.head()
```

Out[27]:

	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 [25]: 1 short_df = purchase_data.loc[:,['Item ID', 'Item Name', 'Price']]
2 short_df = short_df.groupby(['Item ID', 'Item Name'])
3
4 p_popular_count = short_df['Item ID'].count()
5 p_popular_price = short_df['Price'].mean()
6 p_popular_total = short_df['Price'].sum()
7
8 df = pd.DataFrame({"Purchase Count":p_popular_count,
9                   "Item Price":p_popular_price,
10                  "Total Purchase Value":p_popular_total
11                 })
12
13 popular_df = df.sort_values(by="Purchase Count", ascending=False)
14
15 popular_df['Item Price'] = popular_df['Item Price'].map("${:,.2f}".format)
16 popular_df['Total Purchase Value'] = popular_df['Total Purchase Value'].map("${:,.2f}".format)
17 popular_df.head()
```

Out[25]:

		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 [13]: 1 profitable_df = df.sort_values(by="Total Purchase Value", ascending=False)
2
3 profitable_df['Item Price'] = profitable_df['Item Price'].map("${:,.2f}".format)
4 profitable_df['Total Purchase Value'] = profitable_df['Total Purchase Value'].map("${:,.2f}".format)
5
6 profitable_df.head()
```

Out[13]:

		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

Observable Trends

- The volume is highest in the 20-14 group, however the total spent per person is lower than the <10 and 35-39 groups.
- The top spenders are so mainly because of larger number of purchases and not because of higher average price of the items purchased.
- The most profitable items have both a higher than average price, and have sold more times.

In []:

1