

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%).

```
In [153]: # Dependencies and Setup
import pandas as pd
import numpy as np
import html

# File to Load (Remember to Change These)
input = "c:/Users/jhhaskin/Desktop/SMU/Homework/Pandas/purchase_data.csv"

# Read Purchasing File and store into Pandas data frame
purchase_data_df = pd.read_csv(input)
purchase_data_df.head(3)
```

Out[153]:

	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

```
In [109]: # purchase_data_df.columns
```

Out[109]: Index(['Purchase ID', 'SN', 'Age', 'Gender', 'Item ID', 'Item Name', 'Price'], dtype='object')

```
In [110]: # players_df = purchase_data_df.groupby("SN")["SN"].nunique()
players_df.count()
```

Out[110]: 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 [111]: #Number of unique items.

items_df = purchase_data_df.groupby("Item Name")["Item Name"].nunique().count()
items_df
```

Out[111]: 179

```
In [112]: avg_price_df = np.round(purchase_data_df["Price"].mean())
avg_price_df
```

Out[112]: 3.0

```
In [113]: total_purchases_df = purchase_data_df["Purchase ID"].count()
total_purchases_df
```

Out[113]: 780

```
In [114]: total_revenue_df = purchase_data_df["Price"].sum()
total_revenue_df
```

Out[114]: 2379.77

```
In [115]: summary_df = pd.DataFrame({"Number of Unique Items":[items_df],
                                     "Average Purchase Price":[avg_price_df],
                                     "Number of Purchases":[total_purchases_df],
                                     "Total Revenue":[total_revenue_df]})
summary_df
```

Out[115]:

	Number of Unique Items	Average Purchase Price	Number of Purchases	Total Revenue
0	179	3.0	780	2379.77

Gender Demographics

Percentage and Count of Male Players

Percentage and Count of Female Players

Percentage and Count of Other / Non-Disclosed

```
In [116]: # Gender Count
gender_count_df = purchase_data_df.groupby("Gender")["SN"].nunique()
gender_count_df.head()
```

Out[116]:

Gender	
Female	81
Male	484
Other / Non-Disclosed	11

Name: SN, dtype: int64

```
In [117]: # Gender Percentage
gender_percentage_df = np.round(100*gender_count_df/gender_count_df.sum())
gender_percentage_df
```

```
Out[117]: Gender
Female                14.0
Male                 84.0
Other / Non-Disclosed  2.0
Name: SN, dtype: float64
```

```
In [118]: #Gender demographics DataFrame.

gender_demographics_df = pd.DataFrame({"Count": gender_count_df,
                                       "Percentage":gender_percentage_df})

gender_demographics_df
```

```
Out[118]:
```

	Count	Percentage
Gender		
Female	81	14.0
Male	484	84.0
Other / Non-Disclosed	11	2.0

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 [158]: #Total number of purchases by gender.

gender_purchases_df = purchase_data_df.groupby("Gender")["Item Name"]
gender_purchases_df.count()
```

```
Out[158]: Gender
Female                113
Male                 652
Other / Non-Disclosed  15
Name: Item Name, dtype: int64
```

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 [120]: # Create bins in which data will be held. Bins are <10, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, >=40
bins = [0,10,15,20,25,30,35,40, 45]
age_ranges = [<"<10", "10-14","15-19", "20-24", "25-29", "30-34", "35-39", ">=40"]
```

```
In [131]: # Cut purchase data and place the ages into bins
pd.cut(purchase_data_df["Age"], bins, labels=age_ranges).head(3)
```

```
Out[131]: 0    15-19
1    35-39
2    20-24
Name: Age, dtype: category
Categories (8, object): [<10 < 10-14 < 15-19 < 20-24 < 25-29 < 30-34 < 35-39 < >=40]
```

```
In [124]: purchase_data_df["Age Range"] = pd.cut(purchase_data_df["Age"], bins, labels=age_ranges)
purchase_data_df.head()
```

Out[124]:

	Purchase ID	SN	Age	Gender	Item ID	Item Name	Price	Age Range
0	0	Lisim78	20	Male	108	Extraction, Quickblade Of Trembling Hands	3.53	15-19
1	1	Lisovynya38	40	Male	143	Frenzied Scimitar	1.56	35-39
2	2	lthergue48	24	Male	92	Final Critic	4.88	20-24
3	3	Chamassasya86	24	Male	100	Blindscythe	3.27	20-24
4	4	Iskosia90	23	Male	131	Fury	1.44	20-24

```
In [129]: #Players percentage by age range.

age_group_percentage_df = np.round(100*purchase_data_df["Age Range"].value_counts()/purchase_data_df["Age Range"].count(), 1)
age_group_percentage_df
```

```
Out[129]: 20-24      42.0
          15-19      26.0
          25-29      10.0
          10-14       7.0
          30-34       7.0
          35-39       4.0
          <10        4.0
          >=40        1.0
          Name: Age Range, dtype: float64
```

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 b

Display the summary data frame

```
In [132]: # Purchase count by age range.

age_group_count_df = purchase_data_df.groupby("Age Range")["Item Name"].count()
age_group_count_df
```

```
Out[132]: Age Range
          <10      32
          10-14     54
          15-19    200
          20-24    325
          25-29     77
          30-34     52
          35-39     33
          >=40       7
          Name: Item Name, dtype: int64
```

In [139]:  *# Average purchase price by age range.*

```
age_group_average_df = purchase_data_df.groupby("Age Range")["Price"].mean()  
age_group_average_df
```

Out[139]:

Age Range	
<10	3.40
10-14	2.90
15-19	3.11
20-24	3.02
25-29	2.88
30-34	2.99
35-39	3.40
>=40	3.08

Name: Price, dtype: float64

In [140]:  *#Total purchase value by age range.*

```
age_group_total_df = purchase_data_df.groupby("Age Range")["Price"].sum()  
age_group_total_df
```

Out[140]:

Age Range	
<10	108.96
10-14	156.60
15-19	621.56
20-24	981.64
25-29	221.42
30-34	155.71
35-39	112.35
>=40	21.53

Name: Price, dtype: float64

In [142]:  *# Convert to DataFrame.*

```
age_range_df = pd.DataFrame({"Purchase Count":age_group_count_df,
                             "Average Purchase Price":age_group_average_df,
                             "Total Purchase Value": age_group_total_df
                             })
age_range_df
```

Out[142]:

		Purchase Count	Average Purchase Price	Total Purchase Value
Age Range				
<10	(<10, [Blood-Forged Skeletal Spine, Faith's Sc...		3.40	108.96
10-14	(10-14, [Demise, Extraction, Quickblade Of Tre...		2.90	156.60
15-19	(15-19, [Extraction, Quickblade Of Trembling H...		3.11	621.56
20-24	(20-24, [Final Critic, Blindscythe, Fury, Drea...		3.02	981.64
25-29	(25-29, [Blazefury, Protector of Delusions, Pe...		2.88	221.42
30-34	(30-34, [Ghastly Adamantite Protector, Bone Cr...		2.99	155.71
35-39	(35-39, [Frenzied Scimitar, Interrogator, Bloo...		3.40	112.35
>=40	(>=40, [Aetherius, Boon of the Blessed, Demise...		3.08	21.53

In [145]:  *#Overall spending analysis*

```
players_purchase_count_df = purchase_data_df.groupby("SN").count()["Price"].re
players_average_price_df = purchase_data_df.groupby("SN").mean()["Price"].rer
players_total_df = purchase_data_df.groupby("SN").sum()["Price"].rename("Total")

#Convert to DataFrame.

total_user_data_df = pd.DataFrame({"Purchase Count":players_purchase_count_df,
                                    "Average Purchase Price": players_average_
                                    "Total Purchase Value": players_total_df})

total_user_data_df.head(3)
```

Out[145]:

	Purchase Count	Average Purchase Price	Total Purchase Value
SN			
Adairialis76	1	2.28	2.28
Adastirin33	1	4.48	4.48
Aeda94	1	4.91	4.91

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 [146]: # Sort table to show the top five spenders.

top_five_spenders = total_user_data_df.sort_values("Total Purchase Value", as
top_five_spenders.head()
```

Out[146]:

	Purchase Count	Average Purchase Price	Total Purchase Value
SN			
Lisosia93	5	3.792000	18.96
Idastidru52	4	3.862500	15.45
Chamjask73	3	4.610000	13.83
Iral74	4	3.405000	13.62
Iskadarya95	3	4.366667	13.10

```
In [148]: # Total items purchases analysis.

items_purchase_count_df = purchase_data_df.groupby(["Item ID", "Item Name"]).
items_average_price_df = purchase_data_df.groupby(["Item ID", "Item Name"]).n
items_value_total_df = purchase_data_df.groupby(["Item ID", "Item Name"]).sum

# Convert to DataFrame

items_purchased_df = pd.DataFrame({"Purchase Count":items_purchase_count_df,
                                   "Item Price":items_average_price_df,
                                   "Total Purchase Value":items_value_total_c

items_purchased_df.head(3)
```

Out[148]:

		Purchase Count	Item Price	Total Purchase Value
Item ID	Item Name			
0	Splinter	4	1.28	5.12
1	Crucifer	3	3.26	9.78
2	Verdict	6	2.48	14.88

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 [150]: #Sort table to show the five the most popular items.  
  
most_popular_items_df = items_purchased_df.sort_values("Purchase Count", asce  
most_popular_items_df.head()
```

Out[150]:

		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 [152]: # Sort table to show the five the most profitable items.  
  
most_profitable_items_df = items_purchased_df.sort_values("Total Purchase Value", ascending=False)  
most_profitable_items_df.head()
```

Out[152]:

		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

In []: