

Heroes Of Pymoli Data Analysis by Alex Koynoff

1. 84% of the unique users is made up of male.
2. Females and Other/non-disclosed spend more on average per person compared to male.
3. More than 44% of the unique players are in the 20-24 age group, with the next group being the 15-19 age group at 18.58%.
4. Age group 35-39 has the highest average purchase price compared to the rest of the age groups, as well as the highest average total per person.

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 [220]: # 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)
purchase_data.head(5)
```

Out[220]:

	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 [221]: #Unique players by SN. Display the total unique SNs in a new column named Total Players  
unique_players = purchase_data['SN'].nunique()  
total_players = pd.DataFrame({'Total Players':[unique_players]})  
total_players
```

Out[221]:

	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 [222]: #Create various metrics based on csv file info
unique_items = purchase_data['Item ID'].nunique()
average_price = purchase_data['Price'].mean()
count_purchases = purchase_data['Item ID'].count()
total_revenue = purchase_data['Price'].sum()

#Summary table of above metrics
summary = pd.DataFrame(
{'Number of Unique Items':[unique_items],
'Average Price': [average_price],
'Number of Purchases':[count_purchases],
'Total Revenue': [total_revenue]})

#Formatting Table
pd.options.display.float_format = '${:,.2f}'.format
summary
```

Out[222]:

	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 [223]: #Create variables for each gender and perform calculations
unique_players = purchase_data['SN'].nunique()
male = purchase_data[purchase_data["Gender"] == "Male"]["SN"].nunique()
female = purchase_data[purchase_data["Gender"] == "Female"]["SN"].nunique()
other = purchase_data[purchase_data["Gender"] == "Other / Non-Disclosed"]["SN"].nunique()
male_percentage = ((male/unique_players)*100)
female_percentage = ((female/unique_players)*100)
other_percentage = ((other/unique_players)*100)

#Set up data frame and column names
gender_demographics = pd.DataFrame(
    {"Gender":["Male", "Female", "Other / Non-Disclosed"], "Total Count":[male, female, other],
     "Percentage of Players":[male_percentage, female_percentage, other_percentage]})

#Formatting and displaying of data frame. Setting up the index by Gender
pd.options.display.float_format = '{:,.2f}%'.format
gender_demographics=gender_demographics.set_index("Gender")
gender_demographics
```

Out[223]:

	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 [224]: #Calculate values by setting up seperate variables
male_purch_count = purchase_data[purchase_data["Gender"] == "Male"]["Price"].count()
female_purch_count = purchase_data[purchase_data["Gender"] == "Female"]["Price"].count()
other_purch_count = purchase_data[purchase_data["Gender"] == "Other / Non-Disclosed"]["Price"].count()
male_purch_avg = purchase_data[purchase_data["Gender"] == "Male"]["Price"].mean()
female_purch_avg = purchase_data[purchase_data["Gender"] == "Female"]["Price"].mean()
other_purch_avg = purchase_data[purchase_data["Gender"] == "Other / Non-Disclosed"]["Price"].mean()
male_purch_total = purchase_data[purchase_data["Gender"] == "Male"]["Price"].sum()
female_purch_total = purchase_data[purchase_data["Gender"] == "Female"]["Price"].sum()
other_purch_total = purchase_data[purchase_data["Gender"] == "Other / Non-Disclosed"]["Price"].sum()
male_purch_per = male_purch_total/male
female_purch_per = female_purch_total/female
other_purch_per = other_purch_total/other

#Set up data frame with column names and the values from above
purchase_analysis_gender = pd.DataFrame(
    {"Gender":["Male","Female","Other / Non-Disclosed"],"Purchase Count":[male_purch_count,female_purch_count,other_purch_count],
    "Average Purchase Price":[male_purch_avg,female_purch_avg,other_purch_avg],
    "Total Purchase Value":[male_purch_total,female_purch_total,other_purch_total],
    "Avg Total Purchase per Person":[male_purch_per,female_purch_per,other_purch_per]})

#Formatting and displaying of the data frame. Setting up the index by Gender and sorting the Gender column
pd.options.display.float_format = '${:,.2f}'.format
purchase_analysis_gender=purchase_analysis_gender.set_index("Gender")
purchase_analysis_gender=purchase_analysis_gender.sort_values("Gender")
purchase_analysis_gender
```

Out[224]:

	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 [225]: *#Set bins for the age groups*

```
bins = [0,9.99,14.99,19.99,24.99,29.99,34.99,39.99,100]
group_labels = [<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]
purchase_data["Age Group"]=pd.cut(purchase_data["Age"],bins, labels=group_labels)
```

#Set the variables for each age group by pulling the total unique values using the SN

```
unique_players = purchase_data['SN'].nunique()
ten = purchase_data[purchase_data["Age Group"] == "<10"]["SN"].nunique()
lowteens= purchase_data[purchase_data["Age Group"] == "10-14"]["SN"].nunique()
highteens= purchase_data[purchase_data["Age Group"] == "15-19"]["SN"].nunique()
lowtwenty= purchase_data[purchase_data["Age Group"] == "20-24"]["SN"].nunique()
hightwenty= purchase_data[purchase_data["Age Group"] == "25-29"]["SN"].nunique()
lowthirty= purchase_data[purchase_data["Age Group"] == "30-34"]["SN"].nunique()
highthirty= purchase_data[purchase_data["Age Group"] == "35-39"]["SN"].nunique()
forty= purchase_data[purchase_data["Age Group"] == "40+"]["SN"].nunique()
```

#Calculate the percentage of total count by age group

```
ten_percent = ten/unique_players*100
lowteens_percent= lowteens/unique_players*100
highteens_percent= highteens/unique_players*100
lowtwenty_percent= lowtwenty/unique_players*100
hightwenty_percent= hightwenty/unique_players*100
lowthirty_percent= lowthirty/unique_players*100
highthirty_percent= highthirty/unique_players*100
forty_percent= forty/unique_players*100
```

#Set up Table

```
age_demographics=pd.DataFrame(
    {"Age Group":["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"],
     "Total Count":[ten,lowteens,highteens,lowtwenty,hightwenty, lowthirty,highthirty,forty],
     "Percentage of Players":[ten_percent,lowteens_percent,highteens_percent,lowtwenty_percent,hightwenty_per
cent,lowthirty_percent,highthirty_percent,forty_percent]})
```

#Formatting and display

```
pd.options.display.float_format = '{:,.2f}%'.format
age_demographics=age_demographics.set_index("Age Group")
age_demographics
```


Out[225]:

	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 [226]: *#Set bins for the age groups*

```
bins = [0,9.99,14.99,19.99,24.99,29.99,34.99,39.99,100]
group_labels = [<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]
purchase_data["Age Group"] = pd.cut(purchase_data["Age"], bins, labels=group_labels)

#Calculate values by setting up separate variables
ten_purch_count = purchase_data[purchase_data["Age Group"]=="<10"]["Price"].count()
lowteens_purch_count = purchase_data[purchase_data["Age Group"]=="10-14"]["Price"].count()
highteens_purch_count = purchase_data[purchase_data["Age Group"]=="15-19"]["Price"].count()
lowtwenty_purch_count = purchase_data[purchase_data["Age Group"]=="20-24"]["Price"].count()
hightwenty_purch_count = purchase_data[purchase_data["Age Group"]=="25-29"]["Price"].count()
lowthirty_purch_count = purchase_data[purchase_data["Age Group"]=="30-34"]["Price"].count()
highthirty_purch_count = purchase_data[purchase_data["Age Group"]=="35-39"]["Price"].count()
forty_purch_count = purchase_data[purchase_data["Age Group"]=="40+"]["Price"].count()

ten_purch_avg = purchase_data[purchase_data["Age Group"]=="<10"]["Price"].mean()
lowteens_purch_avg = purchase_data[purchase_data["Age Group"]=="10-14"]["Price"].mean()
highteens_purch_avg = purchase_data[purchase_data["Age Group"]=="15-19"]["Price"].mean()
lowtwenty_purch_avg = purchase_data[purchase_data["Age Group"]=="20-24"]["Price"].mean()
hightwenty_purch_avg = purchase_data[purchase_data["Age Group"]=="25-29"]["Price"].mean()
lowthirty_purch_avg = purchase_data[purchase_data["Age Group"]=="30-34"]["Price"].mean()
highthirty_purch_avg = purchase_data[purchase_data["Age Group"]=="35-39"]["Price"].mean()
forty_purch_avg = purchase_data[purchase_data["Age Group"]=="40+"]["Price"].mean()

ten_purch_sum = purchase_data[purchase_data["Age Group"]=="<10"]["Price"].sum()
lowteens_purch_sum = purchase_data[purchase_data["Age Group"]=="10-14"]["Price"].sum()
highteens_purch_sum = purchase_data[purchase_data["Age Group"]=="15-19"]["Price"].sum()
lowtwenty_purch_sum = purchase_data[purchase_data["Age Group"]=="20-24"]["Price"].sum()
hightwenty_purch_sum = purchase_data[purchase_data["Age Group"]=="25-29"]["Price"].sum()
lowthirty_purch_sum = purchase_data[purchase_data["Age Group"]=="30-34"]["Price"].sum()
highthirty_purch_sum = purchase_data[purchase_data["Age Group"]=="35-39"]["Price"].sum()
forty_purch_sum = purchase_data[purchase_data["Age Group"]=="40+"]["Price"].sum()

ten_purch_avg_person = ten_purch_sum/ten
lowteens_purch_avg_person = lowteens_purch_sum/lowteens
highteens_purch_avg_person = highteens_purch_sum/highteens
lowtwenty_purch_avg_person = lowtwenty_purch_sum/lowtwenty
hightwenty_purch_avg_person = hightwenty_purch_sum/hightwenty
lowthirty_purch_avg_person = lowthirty_purch_sum/lowthirty
highthirty_purch_avg_person = highthirty_purch_sum/highthirty
forty_purch_avg_person = forty_purch_sum/forty
```

#Set up Table

```

purchase_analysis_age=pd.DataFrame(
    {"Age Group":["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"],
     "Purchase Count":[ten_purch_count,lowteens_purch_count,highteens_purch_count,lowtwenty_purch_count,hightwenty_purch_count,lowthirty_purch_count,highthirty_purch_count,forty_purch_count,
    ],
     "Average Purchase Price":[ten_purch_avg,lowteens_purch_avg,highteens_purch_avg,lowtwenty_purch_avg,hightwenty_purch_avg,lowthirty_purch_avg,highthirty_purch_avg,forty_purch_avg,
    ],
     "Total Purcahse Value":[ten_purch_sum,lowteens_purch_sum,highteens_purch_sum,lowtwenty_purch_sum,hightwenty_purch_sum,lowthirty_purch_sum,highthirty_purch_sum,forty_purch_sum,
    ],
     "Avg Total Purchase Per Person":[ten_purch_avg_person,lowteens_purch_avg_person,highteens_purch_avg_person,lowtwenty_purch_avg_person,hightwenty_purch_avg_person,lowthirty_purch_avg_person,highthirty_purch_avg_person,forty_purch_avg_person,
    ]})

```

#Formatting and display

```

pd.options.display.float_format = '{:,.2f}'.format
purchase_analysis_age=purchase_analysis_age.set_index("Age Group")
purchase_analysis_age

```

Out[226]:

	Purchase Count	Average Purchase Price	Total Purcahse 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 [227]: #Reorganize the file data by assigning SN as the first column, then group by SN and create variables to calculate various information
purchase_data_sn = purchase_data[["SN", "Purchase ID", "Age", "Age Group", "Gender", "Item ID", "Item Name", "Price"]]
purchase_data_sn_group = purchase_data_sn.groupby(["SN"])
sn_count = purchase_data_sn_group["Price"].count()
sn_avg = purchase_data_sn_group["Price"].mean()
sn_total = purchase_data_sn_group["Price"].sum()

#Set up the Data Frame using the variables above
purchase_data_sn_group = pd.DataFrame({"Purchase Count":sn_count,
                                       "Average Purchase Price":sn_avg,
                                       "Total Purchase Value":sn_total})

#Sort values, format and display
purchase_data_sn_group=purchase_data_sn_group.sort_values("Total Purchase Value", ascending=False)
pd.options.display.float_format = '${:,.2f}'.format
purchase_data_sn_group.head()

```

Out[227]:

	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 [228]: ##Reorganize the file data by assigning SN as the first column, then group by Item ID and Item Name and create variables to calculate various information
purchase_data_popular = purchase_data[["Item ID", "Item Name", "SN", "Purchase ID", "Age", "Age Group", "Gender", "Price"]]
purchase_data_popular_renamed=purchase_data_popular.rename(columns={"Price":"Item Price"})
purchase_data_popular_renamed_grouped=purchase_data_popular_renamed.groupby(["Item ID", "Item Name"])
popular_count = purchase_data_popular_renamed_grouped["Item Price"].count()
popular_avg = purchase_data_popular_renamed_grouped["Item Price"].mean()
popular_total = purchase_data_popular_renamed_grouped["Item Price"].sum()

#Set up the Data Frame using the variables above
purchase_data_popular_renamed_grouped=pd.DataFrame({"Purchase Count":popular_count,
                                                    "Item Price":popular_avg,
                                                    "Total Purchase Value": popular_total})

#Sort values, format and display
purchase_data_popular_renamed_grouped=purchase_data_popular_renamed_grouped.sort_values("Purchase Count", ascending=False)
pd.options.display.float_format = '${:,.2f}'.format
purchase_data_popular_renamed_grouped.head()
```

Out[228]:

		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 [229]: #Resort by Total Purchase Value in descending order, format and display
purchase_data_popular_renamed_grouped=purchase_data_popular_renamed_grouped.sort_values("Total Purchase Value", ascending=False)
pd.options.display.float_format = '{:,.2f}'.format
purchase_data_popular_renamed_grouped.head()
```

Out[229]:

		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 []: