

```
In [1]: # Heroes of Pymoli
# Data Analysis for fantasy game Heroes of Pymoli.
```

```
In [2]: # Dependencies and file Setup
import pandas as pd

# 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()
```

Out[2]:

	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

```
In [3]: # Player Count
# Display the total number of players
total_no_player = pd.DataFrame({"Total no players": [len(purchase_data["SN"].unique())])
total_no_player
```

Out[3]:

	Total no players
0	576

```
In [4]: # 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

Unique_item = len(purchase_data["Item ID"].unique())
Total_purchases = len(purchase_data)
Total_revenue = purchase_data["Price"].sum()
Average_price = (Total_revenue / Total_purchases)

summary_df = pd.DataFrame({"Number of Unique Item": [Unique_item],
                           "Average Price": [Average_price],
                           "Number of Purchases": [Total_purchases],
                           "Total Revenue": [Total_revenue]})
summary_df["Total Revenue"] = summary_df["Total Revenue"].map("${:,.2f}".format)
summary_df["Average Price"] = summary_df["Average Price"].map("${:,.2f}".format)

summary_df
```

Out[4]:

	Number of Unique Item	Average Price	Number of Purchases	Total Revenue
0	179	\$3.05	780	\$2,379.77

```
In [5]: # Gender Demographics
# * Percentage and Count of Male Players
# * Percentage and Count of Female Players
# * Percentage and Count of Other / Non-Disclosed

Gender_demo = purchase_data.groupby(['Gender'])

Count_player = Gender_demo["SN"].nunique()
Total_player = len(purchase_data["SN"].unique())
Percent_player = (Count_player / Total_player) * 100

Gender_demo_summary = pd.DataFrame({"Total Count": Count_player,
                                    "Percentage of Players": Percent_player})

Gender_demo_summary["Percentage of Players"] = Gender_demo_summary["Percentage of Players"].map("{:.2%}".format)
Gender_demo_summary.head()
```

Out[5]:

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

```

In [6]: # Purchasing Analysis (Gender)
# * Run basic calculations to obtain purchase count, avg. purchase price, avg. pu
# * Create a summary data frame to hold the results
# * Optional: give the displayed data cleaner formatting
# * Display the summary data frame

Gender_Purchase = purchase_data.groupby(['Gender'])

Gender_count = Gender_Purchase["SN"].nunique()

Purchase_count = Gender_Purchase["Purchase ID"].count()
Avg_Purchase_Price = Gender_Purchase["Price"].mean()
Total_Purchase_value = Gender_Purchase["Price"].sum()
Total_Purchase_person = (Total_Purchase_value / Gender_count)

Gender_Purchase_summary = pd.DataFrame({"Purchase Count": Purchase_count,
                                         "Average Purchase Price": Avg_Purchase_Pr
                                         "Total Purchase Value": Total_Purchase_va
                                         "Avg Total Purchase per Person": Total_Pu
                                         })
Gender_Purchase_summary["Average Purchase Price"] = Gender_Purchase_summary["Aver
Gender_Purchase_summary["Total Purchase Value"] = Gender_Purchase_summary["Total
Gender_Purchase_summary["Avg Total Purchase per Person"] = Gender_Purchase_summar

Gender_Purchase_summary

```

Out[6]:

	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

```

In [7]: ## 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
Bins = [0, 9, 14, 19, 24, 29, 34, 39, 50]
Age_group = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]

purchase_data["Age_Band"] = pd.cut(purchase_data["Age"], Bins, labels=Age_group,
# purchase_data["Age_bin"].unique()

Age_demo_df = purchase_data.groupby(['Age_Band'])
Age_Total_count = Age_demo_df["SN"].nunique()
Age_Percent = ( Age_Total_count / Total_purchases ) * 100

Age_demo_summary = pd.DataFrame({ "Total Counts": Age_Total_count,
                                   "Percentage of Players": Age_Percent})

Age_demo_summary["Percentage of Players"] = Age_demo_summary["Percentage of Players"]
Age_demo_summary

```

Out[7]:

	Total Counts	Percentage of Players
Age_Band		
<10	17	2.18%
10-14	22	2.82%
15-19	107	13.72%
20-24	258	33.08%
25-29	77	9.87%
30-34	52	6.67%
35-39	31	3.97%
40+	12	1.54%

```

In [8]: # ## Purchasing Analysis (Age)
# Bin the purchase_data data frame by age
# Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase value
# Create a summary data frame to hold the results
# Optional: give the displayed data cleaner formatting
# Display the summary data frame

Purchase_count = Age_demo_df["Purchase ID"].count()
Avg_Purchase_Price = Age_demo_df["Price"].mean()
Total_Purchase_value = Age_demo_df["Price"].sum()
Total_Purchase_person = (Total_Purchase_value / Purchase_count)

Age_Purchase_summary = pd.DataFrame({"Purchase Count": Purchase_count,
                                     "Average Purchase Price": Avg_Purchase_Price,
                                     "Total Purchase Value": Total_Purchase_value,
                                     "Avg Total Purchase per Person": Total_Purchase_person})

Age_Purchase_summary["Average Purchase Price"] = Age_Purchase_summary["Average Purchase Price"]
Age_Purchase_summary["Total Purchase Value"] = Age_Purchase_summary["Total Purchase Value"]
Age_Purchase_summary["Avg Total Purchase per Person"] = Age_Purchase_summary["Avg Total Purchase per Person"]

Age_Purchase_summary

```

Out[8]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
Age_Band				
<10	23	\$3.35	\$77.13	\$3.35
10-14	28	\$2.96	\$82.78	\$2.96
15-19	136	\$3.04	\$412.89	\$3.04
20-24	365	\$3.05	\$1,114.06	\$3.05
25-29	101	\$2.90	\$293.00	\$2.90
30-34	73	\$2.93	\$214.00	\$2.93
35-39	41	\$3.60	\$147.67	\$3.60
40+	13	\$2.94	\$38.24	\$2.94

```

In [9]: ## 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

Top_Spender_grp = purchase_data.groupby(['SN'])

Purchase_count = Top_Spender_grp["Purchase ID"].count()
Avg_Purchase_Price = Top_Spender_grp["Price"].mean()
Total_Purchase_value = Top_Spender_grp["Price"].sum()

Top_Spender_summary = pd.DataFrame({"Purchase Count": Purchase_count,
                                     "Average Purchase Price": Avg_Purchase_Price,
                                     "Total Purchase Value": Total_Purchase_value})

Top_Spender_Sort = Top_Spender_summary.sort_values("Total Purchase Value", ascending=False)

Top_Spender_Sort["Average Purchase Price"] = Top_Spender_Sort["Average Purchase Price"].round(2)
Top_Spender_Sort["Total Purchase Value"] = Top_Spender_Sort["Total Purchase Value"].round(2)

Top_Spender_Sort.head()

```

Out[9]:

	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

```

In [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,
# 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

Most_popular_grp = purchase_data.groupby(['Item ID', 'Item Name'])

Purchase_count = Most_popular_grp["Purchase ID"].count()
Total_Purchase_value = Most_popular_grp["Price"].sum()
Item_Price = Total_Purchase_value / Purchase_count

Most_popular_summary = pd.DataFrame({"Purchase Count": Purchase_count,
                                     "Item Price": Item_Price,
                                     "Total Purchase Value": Total_Purchase_value})

Most_popular_Sort = Most_popular_summary.sort_values("Purchase Count", ascending=False)

Most_popular_Sort["Item Price"] = Most_popular_Sort["Item Price"].map("${:.2F}".format)
Most_popular_Sort["Total Purchase Value"] = Most_popular_Sort["Total Purchase Value"].map("${:.2F}".format)

Most_popular_Sort.head()

```

Out[10]:

Item ID	Item Name	Purchase Count	Item Price	Total Purchase Value
92	Final Critic	13	\$4.61	\$59.99
178	Oathbreaker, Last Hope of the Breaking Storm	12	\$4.23	\$50.76
145	Fiery Glass Crusader	9	\$4.58	\$41.22
132	Persuasion	9	\$3.22	\$28.99
108	Extraction, Quickblade Of Trembling Hands	9	\$3.53	\$31.77

```
In [11]: # 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

Most_popular_Sort = Most_popular_summary.sort_values("Total Purchase Value", asce
Most_popular_Sort["Item Price"] = Most_popular_Sort["Item Price"].map("${:.2F}").f
Most_popular_Sort["Total Purchase Value"] = Most_popular_Sort["Total Purchase Val

Most_popular_Sort.head()
```

Out[11]:

		Purchase Count	Item Price	Total Purchase Value
Item ID	Item Name			
92	Final Critic	13	\$4.61	\$59.99
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
103	Singed Scalpel	8	\$4.35	\$34.80