

```
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)
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	lthergue48	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 [2]: #Player Count
total_players = len(purchase_data['SN'].unique())

# Display the total number of players
players_count = pd.DataFrame({"Total Players": [total_players]})
players_count
```

Out[2]:

	Total Players
0	576

```

In [3]: #Purchasing Analysis (Total)

#Run basic calculations to obtain number of unique items, average price,
etc.
unique_items = len((purchase_data["Item ID"]).unique())
average_price = purchase_data["Price"].mean()
number_of_purchases = purchase_data["Item ID"].count()
total_revenue = purchase_data["Price"].sum()

#Number of Unique Items / Average Price /Number of Purchases / Total Revenue
summary_data = pd.DataFrame(
    {"Number of Unique Items":[unique_items],
     "Average Price":[average_price],
     "Number of Purchases":[number_of_purchases],
     "Total Revenue":[total_revenue]
    }
)

#summary_data.style.format({'Average Price': "${:,.2f}", 'Total Revenue':
"${:,.2f}"})
summary_data["Average Price"] = summary_data["Average Price"].astype(float).map("${:,.2f}".format)
summary_data["Total Revenue"] = summary_data["Total Revenue"].astype(float).map("${:,.2f}".format)

#Display the summary data frame
summary_data

```

Out[3]:

	Number of Unique Items	Average Price	Number of Purchases	Total Revenue
0	183	\$3.05	780	\$2,379.77

```
In [4]: #Gender Demographics
#Percentage and Count of Male Players
#Percentage and Count of Female Players
#Percentage and Count of Other / Non-Disclosed

purchase_data_gender= purchase_data.groupby("Gender")
total_players_gender = purchase_data_gender["SN"].nunique()

percentage_of_players = total_players_gender / total_players * 100

gender_demographics = pd.DataFrame({"Total Count": total_players_gender,
"Percentage of Players": percentage_of_players})

gender_demographics.index.name = None

gender_demographics.sort_values(["Total Count"], ascending = False).style
e.format({"Percentage of Players": "{:.2f}"})
```

Out[4]:

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

```

In [5]: #Purchasing Analysis (Gender)

#Run basic calculations to obtain: Purchase Count / Average Purchase Price / Total Purchase Value /Avg Total Purchase per Person
purchase_count_by_gender = purchase_data_gender["Purchase ID"].count()
avg_purchase_price_by_gender = purchase_data_gender["Price"].mean()
avg_purchase_total_by_gender = purchase_data_gender["Price"].sum()

avg_purchase_per_person = avg_purchase_total_by_gender / total_players_gender

#Create a summary data frame to hold the results
purchasing_analysis_gender = pd.DataFrame({
    "Purchase Count": purchase_count_by_gender,
    "Average Purchase Price": avg_purchase_price_by_gender,
    "Average Purchase Value": avg_purchase_total_by_gender,
    "Avg Purchase Total per Person": avg_purchase_per_person})

gender_demographics.index.name = "Gender"

#Format data (Optional: give the displayed data cleaner formatting)
purchasing_analysis_gender["Average Purchase Price"] = purchasing_analysis_gender["Average Purchase Price"].astype(float).map("${:,.2f}".format)
purchasing_analysis_gender["Average Purchase Value"] = purchasing_analysis_gender["Average Purchase Value"].astype(float).map("${:,.2f}".format)
purchasing_analysis_gender["Avg Purchase Total per Person"] = purchasing_analysis_gender["Avg Purchase Total per Person"].astype(float).map("${:,.2f}".format)

#Display the summary data frame
purchasing_analysis_gender

```

Out[5]:

	Purchase Count	Average Purchase Price	Average Purchase Value	Avg Purchase Total 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 [8]: #Age Demographics

#Establish bins for ages
bins = [0, 9, 14, 19, 24, 29, 34, 39, 100]
#Labels for bins
grouped_names = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]

#Categorize the existing players using the age bins. Hint: use pd.cut()
purchase_data["Age Group"] = pd.cut(purchase_data['Age'], bins, labels = grouped_names)

#Calculate the numbers and percentages by age group
age_group = purchase_data.groupby("Age Group")
total_age = age_group["SN"].nunique()
percentage_age = (total_age/total_players) * 100

#Display Age Demographics Table
age_demographics = pd.DataFrame({"Total Count":total_age,"Percentage of Players": percentage_age})
age_demographics.index.name = None

#Optional: round the percentage column to two decimal points
age_demographics.style.format({"Percentage of Players": "{:, .2f}"})

```

Out[8]:

	Total Count	Percentage of Players
<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

```

In [9]: #Purchasing Analysis (Age)

#Create a summary data frame to hold the results
#Optional: give the displayed data cleaner formatting
#Display the summary data frame

#Run basic calculations to obtain purchase count, avg. purchase price, a
vg. purchase total per person etc. in the table below
purchase_age_count = age_group["Purchase ID"].count()
Average_purchase_price_age = age_group["Price"].mean()
total_purchase_value = age_group["Price"].sum()

Average_purchase_person_age = total_purchase_value/total_age

#Create a summary data frame to hold the results
#Display the summary data frame
#Optional: give the displayed data cleaner formattingage_demographics.in
dex.name = None

age_demographics = pd.DataFrame({"Purchase Count": purchase_age_count,
                                "Average Purchase Price": Average_purch
ase_price_age,
                                "Total Purchase Value":total_purchase_v
alue,
                                "Average Total Purchase per Person": Av
erage_purchase_person_age})

age_demographics.style.format({"Average Purchase Price":"${:,.2f}", "Tota
l Purchase Value":"${:,.2f}",
                                "Average Total Purchase per Person":"$
{:,.2f}"})

```

Out[9]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Average Total Purchase per Person
<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

In [30]: *#Top Spenders*

```
#Run basic calculations to obtain the results in the table below
top_spenders = purchase_data.groupby("SN")
purchasing_spender = top_spenders["Purchase ID"].count()
Average_purchasing_price_per_spender = top_spenders["Price"].mean()
purchase_total_spender = top_spenders["Price"].sum()

#Create a summary data frame to hold the results
#Display a preview of the summary data frame
top_spenders = pd.DataFrame({"Purchase Count": purchasing_spender,
                             "Average Purchase Price": Average_purchasin
g_price_per_spender,
                             "Total Purchase Value":purchase_total_spender})

#Sort the total purchase value column in descending order
format_spenders_frame = top_spenders.sort_values(["Total Purchase Value"],
ascending=False).head()

#Optional: give the displayed data cleaner formatting
format_spenders_frame.style.format({"Average Purchase Total":"${:,.2f}",
"Average Purchase Price":"${:,.2f}", "Total Purchase Value":"${:,.2f}"})
```

Out[30]:

	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 [31]: #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

item_name = purchase_data[["Item ID", "Item Name", "Price"]]

item_group = item_name.groupby(["Item ID", "Item Name"])

purchasing_item = item_group["Price"].count()

purchasing_value = (item_group["Price"].sum())

price = purchasing_value/purchasing_item

popular_items = pd.DataFrame({"Purchase Count": purchasing_item,
                              "Item Price": price,
                              "Total Purchase Value": purchasing_val
ue})

most_popular_format = popular_items.sort_values(["Purchase Count"], asce
nding=False).head()

most_popular_format.style.format({"Item Price": "${:,.2f}", "Total Purchas
e Value": "${:,.2f}"})

```

Out[31]:

		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


```
In [32]: #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_profitable_format = popular_items.sort_values(["Total Purchase Value"], ascending=False).head()
most_profitable_format.style.format({"Item Price": "${:,.2f}", "Total Purchase Value": "${:,.2f}"})
```

Out[32]:

		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 []: