

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]: # Dependencies and 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[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]: #Copy Original DataFrame
Playercount_df = purchase_data.copy()

#Counting total player by counting unique screen names
Total_Player = len(Playercount_df["SN"].unique())

#Creating dataframe with total player value
TotalPlayer_df = pd.DataFrame([{"Total Player": Total_Player}])

TotalPlayer_df.head()
```

Out[2]:

	Total Player
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]: #Copy Original DataFarme
Purchasing_df = purchase_data.copy()

#Count unique number of items
Unique_Item = len(Purchasing_df["Item Name"].unique())

#Average price paid for the items
Avg_price = Purchasing_df["Price"].mean()

#Total number of purchases
Num_purchase = len(Purchasing_df["Purchase ID"])

#Total revenue generated
Total_rev = Purchasing_df["Price"].sum()

#Creating dataframe for the informatino reuqested
Total_df = pd.DataFrame([{"Number of Unique Items": Unique_Item, "Average Price": Avg_price,
                           "Number of Purchases": Num_purchase, "Total Revenue": Total_rev}])

#Format price valudes to two decimal and show $ symbole
Total_df["Average Price"] = Total_df["Average Price"].apply('${:.2f}'.format)
Total_df["Total Revenue"] = Total_df["Total Revenue"].apply('${:,.2f}'.format)

Total_df.head()
```

Out[3]:

	Number of Unique Items	Average Price	Number of Purchases	Total Revenue
0	179	\$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]: #Copy Original DataFarme
Gender_Demo_df = Purchasing_df.copy()

#Groupby Gender and count unique user names per gender
Gender_count = Gender_Demo_df.groupby("Gender").nunique()["SN"]

#Calcdulate percent per Gender catagory
Gender_Percent = (Gender_count / Total_Player) * 100

#Create dataframe for the Gender Demographics
Gender_df = pd.DataFrame({"Total Count": Gender_count, "Percentage of Players"
: Gender_Percent})

#Format percentage of Players colume to two decimal and show % symbole
Gender_df["Percentage of Players"] = Gender_df["Percentage of Players"].apply(
'{:.2f}%'.format)

#Display Gender Demographics table with sorted high to low total count
Gender_df.sort_values(["Total Count"], ascending = False)
```

Out[4]:

	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 [5]: #Copy Original DataFarme
data_df= Purchasing_df.copy()

#Group by Gender perfrom calucations using aggration function on Price
P_Data = data_df.groupby(["Gender"]).Price.agg(["count", "mean", "sum"])

#Calculate total purchase per person by total value with unique Gender count
Total_Purchase_Per_Person = P_Data["sum"]/Gender_count

#Add the avg total purchase per person column to currnect p_data daraframe
P_Data["Avg Total Purchase per Person"] = Total_Purchase_Per_Person

#Rename the columns from count, mean & sum to rquired names
P_Data = P_Data.rename(columns={"count":"Purchase Count", "sum":"Total Purchase Value",
                                "mean":"Average Purchase Price"})

#Format price valudes to two decimal and show $ symbole
P_Data["Average Purchase Price"]= P_Data["Average Purchase Price"].apply('${:,.2f}'.format)
P_Data["Total Purchase Value"]= P_Data["Total Purchase Value"].apply('${:,.2f}'.format)
P_Data["Avg Total Purchase per Person"]= P_Data["Avg Total Purchase per Person"].apply('${:,.2f}'.format)

P_Data.head()

```

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]: #Copy Original DataFarme
Age_df= Purchasing_df.copy()

#Creating Binns and group names
bins = [0, 9.90, 14.90, 19.90, 24.90, 29.90, 34.90, 39.90, 100]
group_name = [<"<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40
+"]

#Matching bin values with Age column
Age_df["Age Range"] = pd.cut(Age_df["Age"], bins, labels=group_name, include_l
owest=True)

#grouping by Age range for unique screen name
Age_group = Age_df.groupby(["Age Range"]).nunique()["SN"]

#creating dataframe for the age range and number of players within in each age
range
Age_demo_df = pd.DataFrame({"Total Count": Age_group})

#Performing calculation to obtain percent value for each age range
Age_percent = (Age_demo_df["Total Count"]/Total_Player) * 100

#adding percent of players column to datafram
Age_demo_df["Percentage of Players"] = Age_percent

#Formating the percentage of players columns with two decimals and show % symb
ole
Age_demo_df["Percentage of Players"] = Age_demo_df["Percentage of Players"].ap
ply('{:.2f}%'.format)

Age_demo_df.head(10)

```

Out[6]:

	Total Count	Percentage of Players
Age Range		
<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]: #Copy Original DataFarme
Purchase_df= Purchasing_df.copy()

#bining the dataframe by age
Purchase_df["Age Range"] = pd.cut(Age_df["Age"], bins, labels=group_name, include_lowest=True)

#Group by Age Range perfrom calucations using aggration function on Price
Purchase_df = Purchase_df.groupby(["Age Range"]).Price.agg(["count", "mean", "sum"])

#Calculate avg purchase per person by total value with unique user count per age range
Avg_Purchase_Per_Person = Purchase_df["sum"]/Age_group

#Add the avg total purchase per person column to currnect Purchase_df darafram e
Purchase_df["Avg Total Purchase per Person"] = Avg_Purchase_Per_Person

#Rename the columns from count, mean & sum to rquired names
Purchase_df = Purchase_df.rename(columns={"count":"Purchase Count", "sum":"Total Purchase Value",
                                         "mean":"Average Purchase Price"})

#Format price valudes to two decimal and show $ symbole
Purchase_df["Average Purchase Price"]= Purchase_df["Average Purchase Price"].apply('${:,.2f}'.format)
Purchase_df["Total Purchase Value"]= Purchase_df["Total Purchase Value"].apply('${:,.2f}'.format)
Purchase_df["Avg Total Purchase per Person"]= Purchase_df["Avg Total Purchase per Person"].apply('${:,.2f}'.format)

Purchase_df.head(10)

```

Out[7]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
Age Range				
<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 [8]: #Copy Original DataFarme
Top_df= Purchasing_df.copy()

#Group by Screen Name (SN) perfrom calucations using aggration function on Pri
ce
Top_df = Top_df.groupby(["SN"]).Price.agg(["count", "mean", "sum"])

#Rename the columns from count, mean & sum to rquired names
Top_df = Top_df.rename(columns={"count":"Purchase Count", "sum":"Total Purchas
e Value",
                               "mean":"Average Purchase Price"})

#Sort the Purchase Count column in desending
Top_Spend_df = Top_df.sort_values(["Total Purchase Value"], ascending = False)

#Format price valudes to two decimal and show $ symbole
Top_Spend_df["Average Purchase Price"]= Top_Spend_df["Average Purchase Price"]
.apply('${:,.2f}'.format)
Top_Spend_df["Total Purchase Value"]= Top_Spend_df["Total Purchase Value"].app
ly('${:,.2f}'.format)

Top_Spend_df.head()
```

Out[8]:

	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 [9]: #Copy Original DataFarme
Most_df= Purchasing_df.copy()

#grouping by Age range for unique screen name
Most_df = Most_df.groupby(["Item ID", "Item Name"]).Price.agg(["count", "sum"
])

#Calculating Item price column & adding to dataframe
Most_df["Item Price"] = Most_df["sum"]/Most_df["count"]

#Rename the columns from count, mean & sum to rquired names
Most_df = Most_df.rename(columns={"count":"Purchase Count", "sum":"Total Purch
ase Value"})

#Reagrande the column Location
Most_df = Most_df[["Purchase Count", "Item Price", "Total Purchase Value"]]

#sorting in decending for Purchase Count
Most_Item_df = Most_df.sort_values(["Purchase Count"], ascending = False)

#Format price valudes to two decimal and show $ symbole
Most_Item_df["Item Price"]= Most_Item_df["Item Price"].apply('${:,.2f}'.format
)
Most_Item_df["Total Purchase Value"]= Most_Item_df["Total Purchase Value"].app
ly('${:,.2f}'.format)

Most_Item_df.head()

```

Out[9]:

		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
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

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 [10]: Most_Profit_df = Most_df.sort_values(["Total Purchase Value"], ascending = False)

#Format price valudes to two decimal and show $ symbole
Most_Profit_df["Item Price"] = Most_Profit_df["Item Price"].apply('${:,.2f}'.format)
Most_Profit_df["Total Purchase Value"] = Most_Profit_df["Total Purchase Value"].apply('${:,.2f}'.format)

Most_Profit_df.head()
```

Out[10]:

		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

In []: