

```
In [14]: # Dependencies and Setup
# ALL FUNCTIONS USED IN THIS EXERCISE ARE BUILT IN PANDAS

import pandas as pd
import csv

# 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[14]:

	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

## Player Count

- Display the total number of players

```
In [2]: #Count number of rows in SN column and place it in a data frame
total_players = len(purchase_data['SN'].value_counts())

df = pd.DataFrame([
    {"Total Players": total_players}
])

df
```

Out[2]:

	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 [3]: #-----  
# Number of Unique Items  
#-----  
unique_items = len(purchase_data["Item ID"].unique())  
  
#-----  
#Average Price  
#-----  
avg_price = purchase_data["Price"].mean()  
avg_price_format = "${:,.2f}".format(avg_price)  
  
#-----  
# Number of purchase  
#-----  
purchase_number = purchase_data["Purchase ID"].count()  
  
#-----  
#Total Revenue  
#-----  
total_revenue = purchase_data["Price"].sum()  
total_revenue_format = "${:,.2f}".format(total_revenue)  
  
#-----  
# Summary Table 1  
#-----  
summary_1 = pd.DataFrame([  
    {"Unique items":unique_items, "Average Price": avg_price_format, "Number of purchase": purchase_number, '  
    ])  
summary_1
```

Out[3]:

	Unique items	Average Price	Number of purchase	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 [10]: unique_players_list = []

# Create a list of unique players and count

with open(file_to_load, 'r', encoding='utf-8') as csvfile:
    csvreader = csv.reader(csvfile, delimiter=",")

    #Skip header
    header = next(csvreader)

    for row in csvreader:

        if row[1] not in unique_players_list:
            unique_players_list.append(row[1])

unique_players = pd.DataFrame([
    {"Unique Players": len(unique_players_list)}
])
```

```
In [13]: unique_players_count = []
unique_gender_list = []
age_list = []

#-----
# Summary table 2: Unique players and Count number of their purchases
#-----

for i in range(len(unique_players_list)):

    player_i = unique_players_list[i]
    gender_i = " "
    age_i = 0

    player_counter = 0

    with open(file_to_load, 'r', encoding='utf-8') as csvfile:
        csvreader = csv.reader(csvfile, delimiter=",")
```

Command Prompt - jupyter notebook

```

#Skip header
header = next(csvreader)

for row in csvreader:

    if str(row[1]) == player_i:
        player_counter += 1
        gender_i = str(row[3])
        age_i = row[2]

unique_players_count.append(player_counter)
unique_gender_list.append(gender_i)
age_list.append(age_i)

summary_2 = pd.DataFrame({
    "Player": unique_players_list,
    "Count": unique_players_count,
    "Gender": unique_gender_list,
    "Age": age_list
})

summary_2.head()

```

Out[13]:

	Player	Count	Gender	Age
0	Lisim78	3	Male	20
1	Lisovynya38	1	Male	40
2	lthergue48	1	Male	24
3	Chamassasya86	1	Male	24
4	Iskoscia90	1	Male	23

In [22]:

```

#-----
# Summary table 3: Total unique players by gender and the %
#-----
#Get number of unique male players
males_count = len(summary_2.loc[summary_2["Gender"] == "Male"])

#Get number of unique female players
females_count = len(summary_2.loc[summary_2["Gender"] == "Female"])

#Get number of unique other genders players
othergenders_count = len(summary_2.loc[summary_2["Gender"] == "Other / Non-Disclosed"])

```

```

#Calculate percentage of each gender
males_percentage = (males_count) / len(unique_players_count) * 100
males_percentage_formated = "{:,.2f}%".format(males_percentage)

females_percentage = (females_count) / len(unique_players_count) * 100
females_percentage_formated = "{:,.2f}%".format(females_percentage)

othergenders_percentaget = (othergenders_count) / len(unique_players_count) * 100
othergenders_percentaget_formated = "{:,.2f}%".format(othergenders_percentaget)

#Create a summary table
summary_3 = pd.DataFrame([
    {"Gender": "Male", "Total Count": males_count, "Percentage of Players": males_percentage_formated},
    {"Gender": "Female", "Total Count": females_count, "Percentage of Players": females_percentage_formated},
    {"Gender": "Other / Non-Disclosed", "Total Count": othergenders_count, "Percentage of Players": othergenc
    ])

#summary_3.set_index('Gender', drop=True, inplace=True)
summary_3

```

Out[22]:

	Gender	Total Count	Percentage of Players
0	Male	484	84.03%
1	Female	81	14.06%
2	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 [24]: #-----
# Summary table 4: Purchase Analysis
#-----

# Create series for each gender
number_of_females = summary_3.loc[ summary_3["Gender"] == "Female", "Total Count" ]
number_of_males = summary_3.loc[ summary_3["Gender"] == "Male", "Total Count" ]
number_of_others = summary_3.loc[ summary_3["Gender"] == "Other / Non-Disclosed", "Total Count" ]

purchase_count = purchase_data.groupby("Gender")["Price"].count()

average_bygender = round(purchase_data.groupby("Gender")["Price"].mean(),2)

purchase_total = purchase_data.groupby("Gender")["Price"].sum()

players_count = summary_2.groupby("Gender")["Player"].count()

summary_4 = pd.DataFrame({
    "Purchase Count": purchase_count,
    "Average Purchase Price": average_bygender,
    "Total Purchase Value": purchase_total,
    #"Players Count": players_count,
    "Avg Total Purchase per Person": purchase_total / players_count
})

summary_4['Average Purchase Price'] = summary_4['Average Purchase Price'].map('${:,.2f}'.format)
summary_4['Total Purchase Value'] = summary_4['Total Purchase Value'].map('${:,.2f}'.format)
summary_4['Avg Total Purchase per Person'] = summary_4['Avg Total Purchase per Person'].map('${:,.2f}'.format)

summary_4
```

Out[24]:

	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 [26]: # Change data type to float
summary_2.loc[:, "Age"] = summary_2["Age"].astype("float")
```

```
In [27]: #-----
# Summary table 5: Age Demographics
#-----

# Create bins for "Age column"
bins = [0, 9, 14, 19, 24, 29, 34, 39, 100]

group_names = [ "<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+" ]

summary_2["Age Group"] = pd.cut(summary_2["Age"], bins, labels=group_names)

# Group by "Age group"
summary_2_group = summary_2.groupby(["Age Group"])

# Calculate the numbers and percentages by age group
age_group_count = summary_2_group["Age Group"].count()
percentage_by_age = age_group_count / total_players * 100

summary_5 = pd.DataFrame({
    "Total Count": age_group_count,
    "Percentage of Players": percentage_by_age,
})

summary_5['Percentage of Players'] = summary_5['Percentage of Players'].map('{:,.2f}%'.format)

summary_5
```



Out[27]:

	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%



```

In [28]: #-----
# Summary table 6: Purchasing Analysis (Age)
#-----
# Have to perform this on a clean df that show unique players

bins_2 = [0, 9, 14, 19, 24, 29, 34, 39, 100]

group_names_2 = [ "<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+" ]

purchase_data["Age Ranges"] = pd.cut(purchase_data["Age"], bins_2, labels=group_names_2)

purchase_data_grouped = purchase_data.groupby(["Age Ranges"])["Age Ranges"].count()
sum_purchase_price = purchase_data.groupby(["Age Ranges"])["Price"].sum()
avg_purchase_price = sum_purchase_price/purchase_data_grouped
avg_perperson = sum_purchase_price/ age_group_count

summary_6 = pd.DataFrame({

    "Purchase Count": purchase_data_grouped,
    "Average Purchase Price": avg_purchase_price,
    "Total Purchase Value": sum_purchase_price,
    "Avg Total Purchase Per Person": avg_perperson

})

summary_6['Average Purchase Price'] = summary_6['Average Purchase Price'].map('${:,.2f}'.format)
summary_6['Total Purchase Value'] = summary_6['Total Purchase Value'].map('${:,.2f}'.format)
summary_6['Avg Total Purchase Per Person'] = summary_6['Avg Total Purchase Per Person'].map('${:,.2f}'.format)

summary_6

```

Out[28]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase Per Person
Age Ranges				
<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 [31]: #-----
# Summary table 7: Top Sepnders
#-----

purchase_count_bySN = purchase_data.groupby(["SN"])["Price"].count()
total_purchase_bySN = purchase_data.groupby(["SN"])["Price"].sum()
avg_purchase_bySN = total_purchase_bySN/ purchase_count_bySN

summary_7 = pd.DataFrame({

    "Purchase Count": purchase_count_bySN,
    "Average Purchase Price": avg_purchase_bySN,
    "Total Purchase Price": total_purchase_bySN,

})

summary_7_sorted = summary_7.sort_values(by=['Total Purchase Price'], ascending=False)

summary_7_sorted['Average Purchase Price'] = summary_7_sorted['Average Purchase Price'].map('${:,.2f}'.format)
summary_7_sorted['Total Purchase Price'] = summary_7_sorted['Total Purchase Price'].map('${:,.2f}'.format)

summary_7_sorted.head()

```

Out[31]:

	Purchase Count	Average Purchase Price	Total Purchase Price
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 [33]: #-----
# Summary table 8: Most popular items
#-----

#Retrieve Item ID, Item Name, and Item Price columns
summary_8 = purchase_data[["Item ID", "Item Name", "Price"]]

# Group by Item ID and Item Name
summary_8_group = summary_8.groupby(["Item ID", "Item Name"])

#Obtain purchase count, average item price, and total purchase value
purchase_count = summary_8_group["Item ID"].count()
total_value = summary_8_group["Price"].sum()
average_price = summary_8_group["Price"].mean()

summary_8 = pd.DataFrame({

    "Purchase Count": purchase_count,
    "Total Purchase Value": total_value,
    "Average Price": average_price

})

summary_8_sorted = summary_8.sort_values(by=['Purchase Count'], ascending=False)

summary_8_sorted['Purchase Count'] = summary_8_sorted['Purchase Count'].map('{:,.0f}'.format)
summary_8_sorted['Total Purchase Value'] = summary_8_sorted['Total Purchase Value'].map('${:,.2f}'.format)
summary_8_sorted['Average Price'] = summary_8_sorted['Average Price'].map('${:,.2f}'.format)

summary_8_sorted.head()

```

Out[33]:

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

## 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 [34]: #-----  
# Sort table above by "Total Purchase"  
#-----  
  
summary_8_sorted.sort_values(by=['Total Purchase Value'], ascending=False)  
summary_8_sorted.head()
```

Out[34]:

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