

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 [44]: # 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[44]:
```

	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 [45]: players = len(purchase_data["SN"].value_counts())
player_count = pd.DataFrame([players], columns = ["Total Players"])
player_count
```

```
Out[45]:
```

	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 [67]: #basic calculations for values
unique= len(purchase_data["Item Name"].unique())
AveragePrice = purchase_data["Price"].mean()
NumberPurchase = len(purchase_data["Item Name"])
revenue = purchase_data["Price"].sum()

#new df
Purchase_Analysis_df = pd.DataFrame({"Number of Unique Items": [unique],
                                     "Average Price": [AveragePrice],
                                     "Number of Purchases": [NumberPurchase],
                                     "Total Revenue": [revenue]})

#cleaner formatting
Purchase_Analysis_df["Average Price"] = Purchase_Analysis_df["Average Price"]
Purchase_Analysis_df["Total Revenue"] = Purchase_Analysis_df["Total Revenue"]

Purchase_Analysis_df.head()
```

```
Out[67]:
```

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

```
In [ ]:
```

Gender Demographics

- Percentage and Count of Male Players
- Percentage and Count of Female Players
- Percentage and Count of Other / Non-Disclosed

```
In [69]: gender = purchase_data[["SN", "Gender"]]
gender = gender.drop_duplicates()
counts = gender["Gender"].value_counts()

#lis of values
total_counts = [counts[0],counts[1],counts[2]]
percents = [round((counts[0]/players)*100,2),round((counts[1]/players)*100,2)
#gender df
gender_df = pd.DataFrame({"Total Count": total_counts,
                          "Percentage of Players": percents})

#cleaner formatting
gender_df["Percentage of Players"] = gender_df["Percentage of Players"].map("
gender_df.index = (["Male", "Female", "Other / Non-Disclosed"])

gender_df
```

```
Out[69]:
```

	Total Count	Percentage of Players
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 [70]: gender = purchase_data.groupby(["Gender"])

#values
count = gender["SN"].count()
price = gender["Price"].mean()
value = gender["Price"].sum()

#delete duplicates
duplicates = purchase_data.drop_duplicates(subset='SN', keep="first")
duplicate = duplicates.groupby(["Gender"])
avg_total = (gender["Price"].sum() / duplicate["SN"].count())

#gender purchases df
purchase_gender = pd.DataFrame({"Purchase Count": count,
                                "Average Purchase Price": price,
                                "Total Purchase Value": value,
                                "Avg Total Purchase per Person": avg_total})

#cleaner formatting
purchase_gender["Average Purchase Price"] = purchase_gender["Average Purchase Price"].round(2)
purchase_gender["Total Purchase Value"] = purchase_gender["Total Purchase Value"].round(2)
purchase_gender["Avg Total Purchase per Person"] = purchase_gender["Avg Total Purchase per Person"].round(2)

purchase_gender

```

```

Out[70]:

```

	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 [157...

```
#set prameters & create bins
age_bins = [0, 9, 14, 19, 24, 29, 34, 39, 9999]
group_names = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]

#sort players into bins
purchase_data["Age Demographics"] = pd.cut(purchase_data["Age"], age_bins, labels=group_names)
age_demo = purchase_data.groupby("Age Demographics")

#calculate the numbers and percentages
count = age_demo["SN"].nunique()
percentage = count / players * 100

#summary data frame
age_demo_df = pd.DataFrame({"Total Count": count,
                             "Percentage of Players": percentage})

#cleaning the percentage column
age_demo_df["Percentage of Players"] = age_demo_df["Percentage of Players"].map(lambda x: f"{x:.2%}")

age_demo_df
```

Out[157...

	Total Count	Percentage of Players
Age Demographics		
<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 [162...

```
#set prameters & create bins
age_bins = [0, 9, 14, 19, 24, 29, 34, 39, 9999]
group_names = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40-44", "45-49", "50-54", "55-59", "60-64", "65-69", "70-74", "75-79", "80-84", "85-89", "90-94", "95-99"]

#create the bins
purchasing_df = purchase_data
purchasing_df['Age Groups'] = pd.cut(purchase_data['Age'], age_bins, labels = group_names)

#calculations for values
purchase_count = purchasing_df[['Price', 'Age Groups']].groupby(['Age Groups']).count().reset_index()
avg_price = purchasing_df[['Price', 'Age Groups']].groupby(['Age Groups']).mean().reset_index()
total_purchase = purchasing_df[['Price', 'Age Groups']].groupby(['Age Groups']).sum().reset_index()
avg_total = (purchase_data['Price'].sum() / count)

#summary data frame
purchase_age = pd.DataFrame({"Purchase Count": [purchase_count],
                             "Average Purchase Price": [avg_price],
                             "Total Purchase Value": [total_purchase],
                             "Avg Total Purchase per Person": [avg_total]})

purchase_age.head()
```

Out[162...

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
0	Price Age Groups <10 ...	Price Age Groups <10 ...	Price Age Groups <10 ...	Age Demographics <10 139.986471 10-14 ...

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 [118... top_spenders = purchase_data.groupby(["SN"])

purchase_count = top_spenders["Item Name"].count()
avg_price = top_spenders["Price"].mean()
total_value = top_spenders["Price"].sum()

#summary table
top_spenders_df = pd.DataFrame({'Purchase Count':purchase_count,
                                'Average Purchase Price':avg_price,
                                'Total Purchase Value': total_value})

#purchase value in descending order
top_spenders_df = top_spenders_df.sort_values(["Total Purchase Value"], ascen

#cleaner formatting
top_spenders_df["Average Purchase Price"] = top_spenders_df["Average Purchase
top_spenders_df["Total Purchase Value"] = top_spenders_df["Total Purchase Val

top_spenders_df.head()

```

```

Out[118...

```

	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, average 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 [119...

```
#retrieve the Item ID, Item Name, and Item Price columns
pop_items = purchase_data[["Item ID", "Item Name", "Price"]]

#group by items
items = pop_items.groupby(["Item ID", "Item Name"])
purchase_count = items["Item ID"].count()
item_price = items["Price"].mean()
tpv = items["Price"].sum()

#summary data frame
items_df = pd.DataFrame({"Purchase Count":purchase_count,
                        "Item Price":item_price,
                        "Total Purchase Value":tpv})

#sort by descending order
items_df = items_df.sort_values(["Purchase Count"], ascending=False)

#cleaner formatting
items_df["Item Price"] = items_df["Item Price"].map("${:,.2f}".format)
items_df["Total Purchase Value"] = items_df["Total Purchase Value"].map("${:,.2f}".format)

items_df.head()
```

Out[119...

		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 [134... #sort by total purchase value
items_df = items_df.sort_values(["Total Purchase Value"], ascending=False)

#cleaner formatting
items_df["Total Purchase Value"] = items_df["Total Purchase Value"].map("${:,.2f}")
items_df.head()
```

Out[134...

		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 [ ]:
In [ ]:
In [ ]:
```