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 [145…
```
# 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[145…

	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 [146…
```
#Set Variables for Total Number of Players
total_players = len(purchase_data['SN'].unique())

#Dataframe
total_players_df = pd.DataFrame({"Total Number of Players": [len(purchase_data.SN.unique())]})

#1st Way
print(players_df)
```

```
#2nd Way
players_df
```

```
   Total Number of Players
0                      576
```

Out[146…

	Total Number of 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 [147…

```
#Run basic calculations to obtain number of unique items, average price, etc.
#Number of Unique Items
unique_items = len(purchase_data["Item ID"].unique())

#Average Price
average_price = purchase_data["Price"].mean()

#Number of Purchases
number_of_purhcases = len(purchase_data["Purchase ID"])

#Calculate the total revenue from all sales
total_revenue = sum(purchase_data["Price"])

#Display the Summary Data Frame
summarydatafr_df = pd.DataFrame({
    'Unique items': unique_items,
    'Average Price': '${:.2f}'.format(avg_price),
    'Total Purchases': total_purchases,
    'Total Revenue': '${:.2f}'.format(total_revenue)},index=[0])
summarydatafr_df
```

Out[147…

	Unique items	Average Price	Total Purchases	Total Revenue

	Unique items	Average Price	Total Purchases	Total Revenue
0	179	$3.05	780	$2379.77

Gender Demographics

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

In [148…

```
#Duplicate SN's need to be removed
player_gender = purchase_data.drop_duplicates('SN')
#https://www.datasciencemadesimple.com/delete-drop-duplicate-row-dataframe-python-pandas/#:~:text=%20Drop%20
#duplicate%20rows%20in%20pandas%20python%20drop_duplicates,argument%20.%20Keeps%20only
#%20the%20non...%20More%20

# Gender of Male/Female/Other/Non-Disclosed
gender_count = player_gender['Gender'].value_counts()

#Percentage of Gender Identification
gender_df = pd.DataFrame({
    'Total Count': gender_count,
    'Percent of Players': gender_percent.map('{:.2f}%'.format)
})
gender_df
```

Out[148…

	Total Count	Percent 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 [149…
```
#Average Purchase Price by Gender
gender_purchase_price = purchase_data.groupby('Gender')
gender_purchase_count = gender_purchase_price['Gender'].count()
gender_purchase_count

#Average Purchase Total per Gender
gender_purchase_mean = gender_purchase_price['Price'].mean()
gender_purchase_mean

#Total Purchase Value
gender_purchase_value = gender_purchase_price['Price'].sum()
gender_purchase_value

#Average Total Purchase per Person
#AveragePersonTotalPurchase = gender_purchase_price/gender_purchase_count

gender_purchase_data_df = pd.DataFrame({
    'Purchase Count': gender_purchase_count,
    'Average Purchase Price': gender_purchase_mean.map('${:,.2f}'.format),
    'Total Purchase Value': gender_purchase_value.map('${:,.2f}'.format),
})
gender_purchase_data_df
```

Out[149…

Gender	Purchase Count	Average Purchase Price	Total Purchase Value
Female	113	$3.20	$361.94
Male	652	$3.02	$1,967.64
Other / Non-Disclosed	15	$3.35	$50.19

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 [150…
```
#Establish bins for ages
bins = [0, 9, 14, 19, 24, 29, 34, 39, 400]
bin_labels = ['<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 Ranges'] =  pd.cut(purchase_data['Age'], bins, labels=bin_labels)
#Grouping Ages
players_age_df = purchase_data.groupby('Age Ranges')
```

In [151…
```
# Calculate the numbers and percentages by age group
ageCount = players_age_df["SN"].nunique()
agePercentage = ageCount / total_players * 100

#Create a summary data frame to hold the results
#Optional: round the percentage column to two decimal points
age_demo = pd.DataFrame({
    'Total Count': players_age,
    'Percentage of Players': (players_age / purchase_data.SN.nunique() * 100).map("{:.2f}%".format)
})
age_demo
```

Out[151…

player_age_range	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%

player_age_range	Total Count	Percentage of Players
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 [152…

```
#Using DataFrame above for Bins
# Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase total per person etc. in the table
#Purchase Count
purchaseCount = players_age_df["Purchase ID"].count()
#Average Purchase Price
averagePurchasePrice = players_age_df["Price"].mean()
#Total Purchase Price
totalPurchasePrice = players_age_df["Price"].sum()
#Average Total Purchase per Person
averageTotalPurchasePerPerson = totalPurchasePrice / ageCount

# Create a summary data frame to hold the results
summary_table = pd.DataFrame({"Purchase Count": purchaseCount,
                              "Average Purchase Price": averagePurchasePrice,
                              "Total Purchase Value": totalPurchasePrice,
                              "Avg Total Purchase per Person": averageTotalPurchasePerPerson})

# Optional: give the displayed data cleaner formatting
summary_table["Average Purchase Price"] = summary_table["Average Purchase Price"].map("${:,.2f}".format)
summary_table["Total Purchase Value"] = summary_table["Total Purchase Value"].map("${:,.2f}".format)
```

```
summary_table["Avg Total Purchase per Person"] = summary_table["Avg Total Purchase per Person"].map("${:,.2f}".format)

# Display the summary data frame
summary_table[["Purchase Count", "Average Purchase Price", "Total Purchase Value", "Avg Total Purchase per Person"]]
```

Out[152…

Age Ranges	Purchase Count	Average Purchase Price	Total Purchase Value	Avg 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

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 [153…

```
# Run basic calculations to obtain the results in the table below
grouped_sn_df = purchase_data.groupby(["SN"])

purchasecount = grouped_sn_df["Purchase ID"].count()
averagepurchaseprice = grouped_sn_df["Price"].mean()
totalpurchasevalue = grouped_sn_df["Price"].sum()
```

```
# Create a summary data frame to hold the results
spend_amount_df = pd.DataFrame({"Purchase Count": purchasecount,
                                "Average Purchase Price": averagepurchaseprice,
                                "Total Purchase Value": totalpurchasevalue})

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

# Optional: give the displayed data cleaner formatting
spend_amount_df["Average Purchase Price"] = spend_amount_df["Average Purchase Price"].map("${:,.2f}".format)
spend_amount_df["Total Purchase Value"] = spend_amount_df["Total Purchase Value"].map("${:,.2f}".format)

# Display a preview of the summary data frame
spend_amount_df[["Purchase Count", "Average Purchase Price", "Total Purchase Value"]].head()
```

Out[153…

SN	Purchase Count	Average Purchase Price	Total Purchase Value
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 [154…

```
# Retrieve the Item ID, Item Name, and Item Price columns
items_df = purchase_data[["Item ID", "Item Name", "Price"]]

# Group by Item ID and Item Name.
grouped_items_df = items_df.groupby(["Item ID", "Item Name"])
# Perform calculations to obtain purchase count, item price, and total purchase value
purchaseCount = grouped_items_df["Item ID"].count()
itemPrice = grouped_items_df["Price"].mean()
totalPurchaseValue = grouped_items_df["Price"].sum()

# Create a summary data frame to hold the results
items_df = pd.DataFrame({"Purchase Count": purchaseCount,
                         "Item Price": itemPrice,
                         "Total Purchase Value": totalPurchaseValue})

# Sort the purchase count column in descending order
items_df = items_df.sort_values(["Purchase Count"], ascending=False)

# Optional: give the displayed data 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)

# Display a preview of the summary data frame
items_df[["Purchase Count", "Item Price", "Total Purchase Value"]].head()
```

Out[154…

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

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 [155…
```
# Converting the "Total Purchase Value" column back to floats
items_df["Total Purchase Value"] = items_df["Total Purchase Value"].apply(lambda x: x.replace('$', '')
                                                                          .replace(',', '')).astype('float')

# Sort the above table by total purchase value in descending order
items_df = items_df.sort_values(["Total Purchase Value"], ascending=False)

# Optional: give the displayed data cleaner formatting
items_df["Total Purchase Value"] = items_df["Total Purchase Value"].map("${:,.2f}".format)

# Display a preview of the data frame
items_df[["Purchase Count", "Item Price", "Total Purchase Value"]].head()
```

Out[155…

		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 [ ]: