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
```

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
...	...	...	...	...	...	...	...
775	775	Aethedru70	21	Female	60	Wolf	3.54
776	776	Iral74	21	Male	164	Exiled Doomblade	1.63
777	777	Yathecal72	20	Male	67	Celeste, Incarnation of the Corrupted	3.46
778	778	Sisur91	7	Male	92	Final Critic	4.19
779	779	Ennrian78	24	Male	50	Dawn	4.60

780 rows × 7 columns

Player Count

- Display the total number of players

In [2]:
```
purchase_data.describe()
```

Out[2]:

	Purchase ID	Age	Item ID	Price
count	780.000000	780.000000	780.000000	780.000000
mean	389.500000	22.714103	91.755128	3.050987
std	225.310896	6.659444	52.697702	1.169549
min	0.000000	7.000000	0.000000	1.000000
25%	194.750000	20.000000	47.750000	1.980000
50%	389.500000	22.000000	92.000000	3.150000
75%	584.250000	25.000000	138.000000	4.080000
max	779.000000	45.000000	183.000000	4.990000

In [3]:
```
total_players = len(purchase_data['SN'].value_counts())
players_count = pd.DataFrame({'Total Players':[total_players]})
players_count
```

Out[3]:

	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 [4]:
```
Number_of_Unique_Items = len(purchase_data['Item Name'].unique())
Average_Price = purchase_data['Price'].mean()
Number_of_Purchases = purchase_data['Purchase ID'].count()
Total_Revenue = purchase_data['Price'].sum()
```

In [5]:
```
#create a new dataframe
purchase_summary = pd.DataFrame([{'Number of Unique Items':Number_of_Unique_It
ems,
                                  'Average Price':Average_Price,
                                  'Number of Purchases':Number_of_Purchases,
                                  'Total Revenue':Total_Revenue}])
```

In [6]:
```
columns = ["Number_of_Unique_Items", "Average_Price", "Number_of_Purchases",
"Total_Revenue"]

print(columns)
```

```
['Number_of_Unique_Items', 'Average_Price', 'Number_of_Purchases', 'Total_Rev
enue']
```

In [7]:
```
#purchase_summary.style.format({'Average Price':"${:,.2f}",
#                                'Total Revenue':"${:,.2f}"})

purchase_summary['Average Price']=purchase_summary['Average Price'].astype(flo
at).map("${:,.2f}".format)
purchase_summary['Total Revenue']=purchase_summary['Total Revenue'].astype(flo
at).map("${:,.2f}".format)
```

In [8]:
```
purchase_summary
```

Out[8]:

	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 [9]:
```
# Identify Unique Records using column 'SN'
purchase_data_unique = purchase_data.drop_duplicates('SN')
purchase_data_unique.head()
```

Out[9]:

	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

In [10]:
```
# Identify gender counts

total_gender = purchase_data_unique['Gender'].value_counts()
total_gender
```

Out[10]:
```
Male                     484
Female                    81
Other / Non-Disclosed     11
Name: Gender, dtype: int64
```

In [11]:
```
# calculate percentage

Percentage_of_Players = total_gender/total_players * 100
Percentage_of_Players
```

Out[11]:
```
Male                     84.027778
Female                   14.062500
Other / Non-Disclosed     1.909722
Name: Gender, dtype: float64
```

In [12]:
```
# create a summary data frame
gender_demographics = pd.DataFrame({'Total Count':total_gender,'Percentage of
 Players':Percentage_of_Players})
```

In [13]:
```
# format the Percetage field

gender_demographics['Percentage of Players'] = gender_demographics['Percentage
of Players'].astype(float).map("{:,.2f}%".format)
gender_demographics
```

Out[13]:

	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 [14]:
```
#group by gender
group_gender_df = purchase_data.groupby(['Gender'])
print(group_gender_df)
group_gender_df.count().head()
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001ED8A1BD048>
```

Out[14]:

Gender	Purchase ID	SN	Age	Item ID	Item Name	Price
Female	113	113	113	113	113	113
Male	652	652	652	652	652	652
Other / Non-Disclosed	15	15	15	15	15	15

In [15]:
```
# calculate purchase count by 'Gender'
Purchase_Count = group_gender_df['Purchase ID'].count().head()
Purchase_Count
```

Out[15]:
```
Gender
Female                   113
Male                     652
Other / Non-Disclosed     15
Name: Purchase ID, dtype: int64
```

In [16]:
```
# calculate Average Purchase Price by gender
Average_Purchase_Price = group_gender_df['Price'].mean().head()
Average_Purchase_Price
```

Out[16]:
```
Gender
Female                   3.203009
Male                     3.017853
Other / Non-Disclosed    3.346000
Name: Price, dtype: float64
```

In [17]:
```
# Total purchase value per gender
Total_Purchase_Value = group_gender_df['Price'].sum().head()
Total_Purchase_Value
```

Out[17]:
```
Gender
Female                    361.94
Male                     1967.64
Other / Non-Disclosed      50.19
Name: Price, dtype: float64
```

In [18]:
```
# Calculate Avg total purchase per person
Avg_Total_Purcahse_per_Person = Total_Purchase_Value/total_gender
Avg_Total_Purcahse_per_Person
```

Out[18]:
```
Female                   4.468395
Male                     4.065372
Other / Non-Disclosed    4.562727
dtype: float64
```

In [20]:
```
# create a summary dataframe
purchasing_analysis_summary = pd.DataFrame({
    'Purchase Count':Purchase_Count,
    'Average Purchase Price':Average_Purchase_Price,
    'Total Purchase Value': Total_Purchase_Value,
    'Avg Total Purchase per Person': Avg_Total_Purcahse_per_Person
})
```

In [21]:
```
# Format the price columns
purchasing_analysis_summary['Average Purchase Price'] = purchasing_analysis_summary['Average Purchase Price'].astype(float).map("${:,.2f}".format)
purchasing_analysis_summary['Total Purchase Value'] = purchasing_analysis_summary['Total Purchase Value'].astype(float).map("${:,.2f}".format)
purchasing_analysis_summary['Avg Total Purchase per Person'] = purchasing_analysis_summary['Avg Total Purchase per Person'].astype(float).map("${:,.2f}".format)
purchasing_analysis_summary
```

Out[21]:

Gender	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
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 [24]:
```
purchase_data_unique.describe()
```

Out[24]:

	Purchase ID	Age	Item ID	Price
count	576.000000	576.000000	576.000000	576.000000
mean	350.331597	22.741319	92.527778	3.070573
std	222.226127	6.838568	53.923997	1.164585
min	0.000000	7.000000	0.000000	1.000000
25%	158.750000	19.000000	46.000000	1.980000
50%	336.500000	22.000000	93.000000	3.160000
75%	529.250000	25.000000	142.000000	4.102500
max	778.000000	45.000000	183.000000	4.990000

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 [33]:
```
bins = [0,9.99,14.99, 19.99, 24.99, 29.99, 34.99,39.99,199.99]
Group_names = ['<10','10-14','15-19','20-24','25-29','30-34','35-39','40+']

purchase_data["Age Group"] = pd.cut(purchase_data["Age"], bins, labels=Group_n
ames, include_lowest=True)
purchase_data
```

Out[33]:

	Purchase ID	SN	Age	Gender	Item ID	Item Name	Price	Age Group
0	0	Lisim78	20	Male	108	Extraction, Quickblade Of Trembling Hands	3.53	20-24
1	1	Lisovynya38	40	Male	143	Frenzied Scimitar	1.56	40+
2	2	Ithergue48	24	Male	92	Final Critic	4.88	20-24
3	3	Chamassasya86	24	Male	100	Blindscythe	3.27	20-24
4	4	Iskosia90	23	Male	131	Fury	1.44	20-24
...	...	...	...	...	...	...	...	...
775	775	Aethedru70	21	Female	60	Wolf	3.54	20-24
776	776	Iral74	21	Male	164	Exiled Doomblade	1.63	20-24
777	777	Yathecal72	20	Male	67	Celeste, Incarnation of the Corrupted	3.46	20-24
778	778	Sisur91	7	Male	92	Final Critic	4.19	<10
779	779	Ennrian78	24	Male	50	Dawn	4.60	20-24

780 rows × 8 columns

In [39]:
```
# Identify Unique Records using column 'SN'
purchase_data_unique = purchase_data.drop_duplicates('SN')
purchase_data_unique.head()
```

Out[39]:

	Purchase ID	SN	Age	Gender	Item ID	Item Name	Price	Age Group
0	0	Lisim78	20	Male	108	Extraction, Quickblade Of Trembling Hands	3.53	20-24
1	1	Lisovynya38	40	Male	143	Frenzied Scimitar	1.56	40+
2	2	Ithergue48	24	Male	92	Final Critic	4.88	20-24
3	3	Chamassasya86	24	Male	100	Blindscythe	3.27	20-24
4	4	Iskosia90	23	Male	131	Fury	1.44	20-24

In [43]:
```
# Group by Age Group

group_age_df = purchase_data_unique.groupby(['Age Group'])
print(group_age_df)
group_age_df.count()
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001ED8B306A90>
```

Out[43]:

Age Group	Purchase ID	SN	Age	Gender	Item ID	Item Name	Price
<10	17	17	17	17	17	17	17
10-14	22	22	22	22	22	22	22
15-19	107	107	107	107	107	107	107
20-24	258	258	258	258	258	258	258
25-29	77	77	77	77	77	77	77
30-34	52	52	52	52	52	52	52
35-39	31	31	31	31	31	31	31
40+	12	12	12	12	12	12	12

In [49]:
```
# calculate purchase count by 'Age Group'
unique_age_Count = group_age_df['Purchase ID'].count().head()
unique_age_Count
```

Out[49]:
```
Age Group
<10       17
10-14     22
15-19    107
20-24    258
25-29     77
Name: Purchase ID, dtype: int64
```

In [57]:
```
# Calculate purchase_count_by_age_group

purchase_count_by_age_group = purchase_data['Age Group'].value_counts()
purchase_count_by_age_group
```

Out[57]:
```
20-24    365
15-19    136
25-29    101
30-34     73
35-39     41
10-14     28
<10       23
40+       13
Name: Age Group, dtype: int64
```

In [60]:
```
# calculate Average Purchase Price by Age Group
Average_Purchase_Price = purchase_data.groupby('Age Group')['Price'].mean().he
ad()
Average_Purchase_Price
```

Out[60]:
```
Age Group
<10      3.353478
10-14    2.956429
15-19    3.035956
20-24    3.052219
25-29    2.900990
Name: Price, dtype: float64
```

In [61]:
```
# Total purchase value per Age Group
Total_Purchase_Value = purchase_data.groupby('Age Group')['Price'].sum().head
()
Total_Purchase_Value
```

Out[61]:
```
Age Group
<10        77.13
10-14      82.78
15-19     412.89
20-24    1114.06
25-29     293.00
Name: Price, dtype: float64
```

In [47]:
```
# Identify age group counts

total_age_group = purchase_data_unique['Age Group'].value_counts()
total_age_group
```

Out[47]:
```
20-24    258
15-19    107
25-29     77
30-34     52
35-39     31
10-14     22
<10       17
40+       12
Name: Age Group, dtype: int64
```

In [62]:
```
# Calculate Avg total purchase per
Avg_Total_Purcahse_per_Person = Total_Purchase_Value/total_age_group
Avg_Total_Purcahse_per_Person
```

Out[62]:
```
<10      4.537059
10-14    3.762727
15-19    3.858785
20-24    4.318062
25-29    3.805195
30-34         NaN
35-39         NaN
40+           NaN
dtype: float64
```

In [51]:
```
# calculate Percentage of Players

Percentage_of_Players = unique_age_Count/total_players * 100
```

In [53]:
```
# create a new DF with summary

age_summary = pd.DataFrame({
    'Total Count': unique_age_Count ,
    'Percentage of Players': Percentage_of_Players
})
age_summary
```

Out[53]:

Age Group	Total Count	Percentage of Players
<10	17	2.951389
10-14	22	3.819444
15-19	107	18.576389
20-24	258	44.791667
25-29	77	13.368056

In [54]:
```
# Format Percentqage column
age_summary['Percentage of Players'] = age_summary['Percentage of Players'].as
type(float).map("{:,.2f}%".format)
age_summary
```

Out[54]:

Age Group	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%

In [55]:
```
# Remove Index name
age_summary.index.name=None
age_summary
```

Out[55]:

	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%

In [63]:
```
Purchasing_Analysis_Summary = pd.DataFrame({
    'Purchase Count': purchase_count_by_age_group,
    'Average Purchase Price': Average_Purchase_Price,
    'Total Purchase Value': Total_Purchase_Value,
    'Avg Total Purchase per Person':Avg_Total_Purcahse_per_Person
})
Purchasing_Analysis_Summary
```

Out[63]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
<10	23	3.353478	77.13	4.537059
10-14	28	2.956429	82.78	3.762727
15-19	136	3.035956	412.89	3.858785
20-24	365	3.052219	1114.06	4.318062
25-29	101	2.900990	293.00	3.805195
30-34	73	NaN	NaN	NaN
35-39	41	NaN	NaN	NaN
40+	13	NaN	NaN	NaN

In [65]:
```
Purchasing_Analysis_Summary['Average Purchase Price'] = Purchasing_Analysis_Summary['Average Purchase Price'].astype(float).map("${:,.2f}".format)
Purchasing_Analysis_Summary['Total Purchase Value'] = Purchasing_Analysis_Summary['Total Purchase Value'].astype(float).map("${:,.2f}".format)
Purchasing_Analysis_Summary['Avg Total Purchase per Person'] = Purchasing_Analysis_Summary['Avg Total Purchase per Person'].astype(float).map("${:,.2f}".format)
Purchasing_Analysis_Summary
```

Out[65]:

	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	$nan	$nan	$nan
35-39	41	$nan	$nan	$nan
40+	13	$nan	$nan	$nan

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 [73]:
```
# Group purchase data by screen names
spender_stats = purchase_data.groupby("SN")

# Count the total purchases by name
purchase_count_spender = spender_stats["Purchase ID"].count()

# Calculate the average purchase by name
avg_purchase_price_spender = spender_stats["Price"].mean()

# Calculate purchase total
purchase_total_spender = spender_stats["Price"].sum()

# Create data frame with obtained values
top_spenders = pd.DataFrame({"Purchase Count": purchase_count_spender,
                             "Average Purchase Price": avg_purchase_price_spen
der,
                             "Total Purchase Value":purchase_total_spender})

# Sort in descending order to obtain top 5 spender names
formatted_spenders = top_spenders.sort_values(["Total Purchase Value"], ascend
ing=False).head()

# Format with currency style
formatted_spenders.style.format({"Average Purchase Price":"${:,.2f}",
                                 "Total Purchase Value":"${:,.2f}"})
```

Out[73]:

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 [75]:
```
# create a df with above 3 cols
item_df = purchase_data[['Item ID','Item Name','Price']]

#group by Item ID & Item Name
item_stats = item_df.groupby(['Item ID','Item Name'])

# purchase count
Item_Purchase_Count = item_stats['Price'].count()

# Total Purchase count
Item_Total_Purchase_Value = (item_stats['Price'].sum())

# Average Item price
Item_average_item_price = (item_stats['Price'].mean())

# Create summary DF
item_summary = pd.DataFrame({"Purchase Count":Item_Purchase_Count,
                             "Item Price": Item_average_item_price,
                              "Total Purchase Value":Item_Total_Purchase_Value
                             })

item_summary_formatted =item_summary.sort_values(['Purchase Count'],ascending=
False).head()

# Format with currency style
item_summary_formatted.style.format({"Item Price":"${:,.2f}",
                                     "Total Purchase Value":"${:,.2f}"})
```

Out[75]:

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

In [76]:
```
sorted_total_purchase =item_summary.sort_values(['Total Purchase Value'],ascending=False).head()
# Format with currency style
sorted_total_purchase.style.format({"Item Price":"${:,.2f}",
                                    "Total Purchase Value":"${:,.2f}"})
```

Out[76]:

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
82	**Nirvana**	9	$4.90	$44.10
145	**Fiery Glass Crusader**	9	$4.58	$41.22
103	**Singed Scalpel**	8	$4.35	$34.80

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 [ ]:
```
```