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

In [2]:
```
# Get total number of unique players
pd.DataFrame([{"Total Players" : purchase_data["SN"].nunique()}])
```

Out[2]:

	Total Players
0	576

In [3]:
```
# Create a summary data frame
summary_df = pd.DataFrame([{
    # Get total number of unique items
    "Number of Unique Items" : purchase_data["Item Name"].nunique(),
    # Get average of items
    "Average Price" : purchase_data["Price"].mean(),
    # Count number of purchases
    "Number of Purchases" : purchase_data["Purchase ID"].count(),
    # Get total revenue
    "Total Revenue" : purchase_data["Price"].sum()
}])

# Convert formats
summary_df["Average Price"] = summary_df["Average Price"].map("${:,.2f}".format)
summary_df["Total Revenue"] = summary_df["Total Revenue"].map("${:,.2f}".format)

summary_df
```

Out[3]:

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

In [4]:
```
# Get total number of each gender, Duplicates removed
gender_df = pd.DataFrame({"Total Count" : purchase_data.drop_duplicates("SN")["Gender"].value_counts()})

# Get percentage of players by gender
gender_df["Percentage of Players"] = gender_df["Total Count"] / gender_df["Total Count"].sum() * 100

# Convert formats
gender_df["Percentage of Players"] = gender_df["Percentage of Players"].map("{:,.2f}%".format)

gender_df
```

Out[4]:

	Total Count	Percentage of Players
Male	484	84.03%
Female	81	14.06%
Other / Non-Disclosed	11	1.91%

In [5]:
```
# Group by gender to get values
gen_group = purchase_data["Price"].groupby(purchase_data["Gender"])

summary_gen_df = pd.DataFrame({
    # Get purchase count by gender
    "Purchase Count" : gen_group.count(),
    # Get average purchase price per case by gender
    "Average Purchase Price" : gen_group.mean(),
    # Get total purchase value by gender
    "Total Purchase Value" : gen_group.sum(),
    # Get average total purchase per person by gender
    "Avg Total Purchase per Person" : gen_group.sum() / gender_df["Total Count"]
})

# Convert formats
summary_gen_df["Average Purchase Price"] = summary_gen_df["Average Purchase Price"].map("${:,.2f}".format)
summary_gen_df["Avg Total Purchase per Person"] = summary_gen_df["Avg Total Purchase per Person"].map("${:,.2f}".format)

summary_gen_df
```

Out[5]:

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	1967.64	$4.07
Other / Non-Disclosed	15	$3.35	50.19	$4.56

In [6]:
```
# Set binning values and labels
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+"]

# Cut dataframe and count by bin
age_demo_df = pd.DataFrame({"Total Count" : pd.cut(purchase_data.drop_duplicates("SN")["Age"], bins, labels = group_names).value_counts()},

# Get percentage of players by bin
age_demo_df["Percentage of Players"] = age_demo_df["Total Count"] / age_demo_df["Total Count"].sum() * 100

# Convert format
age_demo_df["Percentage of Players"] = age_demo_df["Percentage of Players"].map("{:.2f}%".format)

age_demo_df
```

Out[6]:

	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%
30-34	52	9.03%
35-39	31	5.38%
40+	12	2.08%

In [7]:
```
# Cut dataframe by bin
purchase_data["Age Ranges"] = pd.DataFrame({"Purchase Count" : pd.cut(purchase_data["Age"], bins, labels = group_names)})
# Group data by age range
age_group = purchase_data["Price"].groupby(purchase_data["Age Ranges"])

# Make dataframe to store result
summary_age_df = pd.DataFrame({
    # Get purchase count by age range
    "Purchase Count" : age_group.count(),
    # Get average purchase value by age range
    "Average Purchase Price" : age_group.mean(),
    # Get total purchase value by age range
    "Total Purchase Value" : age_group.sum(),
    # Get average total purchase value per person by age range
    "Avg Total Purchase per Person" : age_group.sum() / age_demo_df["Total Count"]
}, index = group_names)

# Convert formats
summary_age_df["Average Purchase Price"] = summary_age_df["Average Purchase Price"].map("${:.2f}".format)
summary_age_df["Total Purchase Value"] = summary_age_df["Total Purchase Value"].map("${:.2f}".format)
summary_age_df["Avg Total Purchase per Person"] = summary_age_df["Avg Total Purchase per Person"].map("${:.2f}".format)

summary_age_df
```

Out[7]:

	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	$1114.06	$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 [8]:
```
# Group by SN to get top spenders
sn_group = purchase_data["Price"].groupby(purchase_data["SN"])

# Create dataframe to hold result
top_spenders_df = pd.DataFrame({
    # Get purchase count by SN
    "Purchase Count" : sn_group.count(),
    # Get average purchase value by SN
    "Average Purchase Price" : sn_group.mean(),
    # Get total purchase value by SN
    "Total Purchase Value" : sn_group.sum()
}).sort_values(by="Total Purchase Value", ascending = False)

# Convert formats
top_spenders_df["Average Purchase Price"] = top_spenders_df["Average Purchase Price"].map("${:.2f}".format)
top_spenders_df["Total Purchase Value"] = top_spenders_df["Total Purchase Value"].map("${:.2f}".format)

top_spenders_df.head()
```

Out[8]:

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

In [9]:
```
# Group by Item ID, Item Name
id_group = purchase_data[["Item ID", "Item Name", "Price"]].groupby(["Item ID", "Item Name"])

# Create dataframe to hole result
popular_items_df = pd.DataFrame({
    # Get purchase count by item
    "Purchase Count" : id_group["Price"].count(),
    # Get item price
    "Item Price" : id_group["Price"].mean(),
    # Get total purchase value by item
    "Total Purchase Value" : id_group["Price"].sum()
# Sort by purchase count
}).sort_values(["Purchase Count"], ascending = False)

# Copy dataframe before converting format to use later
profitable_items_df = popular_items_df.copy()

# Convert formats
popular_items_df["Item Price"] = popular_items_df["Item Price"].map("${:.2f}".format)
popular_items_df["Total Purchase Value"] = popular_items_df["Total Purchase Value"].map("${:.2f}".format)

''' As a side note, this result is different from the result of example you provided because the data were grouped the data
by Item ID and Item Name at the same time as the above instruction says. There are two kinds of Final Critic in the provided CSV file
seperated by Item ID. One's Item ID is 92, and the other one is 101. '''

popular_items_df.head()
```

Out[9]:

		Purchase Count	Item Price	Total Purchase Value
Item ID	**Item Name**			
178	**Oathbreaker, Last Hope of the Breaking Storm**	12	$4.23	$50.76
145	**Fiery Glass Crusader**	9	$4.58	$41.22
108	**Extraction, Quickblade Of Trembling Hands**	9	$3.53	$31.77
82	**Nirvana**	9	$4.90	$44.10
19	**Pursuit, Cudgel of Necromancy**	8	$1.02	$8.16

In [10]:
```
# Sort by total purchase value
profitable_items_df = profitable_items_df.sort_values(["Total Purchase Value"], ascending = False)

# Convert formats
profitable_items_df["Item Price"] = profitable_items_df["Item Price"].map("${:.2f}".format)
profitable_items_df["Total Purchase Value"] = profitable_items_df["Total Purchase Value"].map("${:.2f}".format)

profitable_items_df.head()

# this result is also different from the result of example because of the same issue.
```

Out[10]:

		Purchase Count	Item Price	Total Purchase Value
Item ID	**Item Name**			
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
92	**Final Critic**	8	$4.88	$39.04
103	**Singed Scalpel**	8	$4.35	$34.80

Result

- Male gamers have played this game much more than the other gender gamers, and they account for 84% of the total sales.
- 44% of users are in 20 - 24 age group and they've purchased items the most.
- There's not a correlation between item prices and the number of purchases.