In [4]:
```
import pandas as pd

pymoli = 'Resources/purchase_data.csv'

data = pd.read_csv(pymoli)
```

In [5]:
```
data.head()
```

Out[5]:

	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 [6]:
```
# Player Count
total_players = data['SN'].value_counts()
players = total_players.count()
total_players = pd.DataFrame({"Total Players": [players]})
total_players
```

Out[6]:

	Total Players
0	576

In [7]:
```
# Purchasing Analysis in Total
avg_price = data['Price'].mean()
purchase_count = data['Purchase ID'].count()
unique_items = data['Item ID'].value_counts()
count_unique_item = len(unique_items)
total_revenue = data['Price'].sum()
avg_price
count_unique_item
purchase_count
total_revenue
summary = pd.DataFrame({'Number of Unique Item': [count_unique_item],
                       'Average Price': [avg_price], 'Number of Purchase
s': [purchase_count],
                       'Total Revenue': [total_revenue]})
summary
```

Out[7]:

	Number of Unique Item	Average Price	Number of Purchases	Total Revenue
0	183	3.050987	780	2379.77

In [8]:

```
# Format
#summary_format = summary.style.format({'Average Price': '$ {:.2f}', 'Total Revenue': '$ {:.2f}'})
#summary_format
summary['Average Price'] = summary['Average Price'].map("$ {:.2f}".format)
summary['Total Revenue'] = summary['Total Revenue'].map("$ {:.2f}".format)
summary
```

Out[8]:

	Number of Unique Item	Average Price	Number of Purchases	Total Revenue
0	183	$ 3.05	780	$ 2379.77

In [9]:

```
# Gender Demographic
gender = data.groupby(['Gender'])['SN'].nunique(dropna=True)
percentage_player = gender/players*100
gender_purchase = data.groupby(['Gender'])['SN'].count()
total_purchase = data.groupby(['Gender'])['Price'].sum()
avg_purchase_price = total_purchase/gender_purchase
avg_purchase_person = total_purchase/gender
demographic = pd.DataFrame({'Total Count': gender, 'Percentage of Players':percentage_player })
demographic
```

Out[9]:

	Total Count	Percentage of Players
Gender		
Female	81	14.062500
Male	484	84.027778
Other / Non-Disclosed	11	1.909722

In [10]:

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

Out[10]:

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

In [11]:

```
# Purchasing Analysis by Gender
purchasing_analysis = pd.DataFrame({'Purchase Count': gender_purchase,
'Avg Purchase Price': avg_purchase_price,
                                    'Total Purchase Price': total_purchase,
'Avg Total Purchase/Person': avg_purchase_person})
purchasing_analysis
```

Out[11]:

	Purchase Count	Avg Purchase Price	Total Purchase Price	Avg Total Purchase/Person
Gender				
Female	113	3.203009	361.94	4.468395
Male	652	3.017853	1967.64	4.065372
Other / Non-Disclosed	15	3.346000	50.19	4.562727

In [12]:

```
purchasing_analysis['Avg Purchase Price'] = purchasing_analysis['Avg Purchase Price'].map('$ {:.2f}'.format)
purchasing_analysis['Total Purchase Price'] = purchasing_analysis['Total Purchase Price'].map('$ {:.2f}'.format)
purchasing_analysis['Avg Total Purchase/Person'] = purchasing_analysis[
'Avg Total Purchase/Person'].map('$ {:.2f}'.format)
purchasing_analysis
```

Out[12]:

	Purchase Count	Avg Purchase Price	Total Purchase Price	Avg Total Purchase/Person
Gender				
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 [13]:

```python
# Creating the group ages using bins
bins = [0,9,14,19,24,29,34,39, 70]
group_names = ["< 9", "10 to 14", "15 to 19", "20 to 24", "25 to 29", "30 to 34", "35 to 39", "40 +"]
pd.cut(data['Age'], bins, labels = group_names)
data['Age Group'] = pd.cut(data['Age'], bins, labels = group_names)
age_count = data.groupby(['Age Group'])['SN'].nunique()
#print(age_group['Age'].count())
#age_group.head()
percentage_players = data.groupby(['Age Group'])['Age'].count()/data['SN'].count()*100
#percentage_players
age_count
```

Out[13]:

```
Age Group
< 9          17
10 to 14     22
15 to 19    107
20 to 24    258
25 to 29     77
30 to 34     52
35 to 39     31
40 +         12
Name: SN, dtype: int64
```

In [14]:

```python
age_demographic = pd.DataFrame({'Total Count': age_count, 'Percentage of Players':percentage_players })
age_demographic
```

Out[14]:

	Total Count	Percentage of Players
Age Group		
< 9	17	2.948718
10 to 14	22	3.589744
15 to 19	107	17.435897
20 to 24	258	46.794872
25 to 29	77	12.948718
30 to 34	52	9.358974
35 to 39	31	5.256410
40 +	12	1.666667

In [15]:

```
# Age Demographic
age_demographic['Percentage of Players'] = age_demographic['Percentage o
f Players'].map('{:.2f} %'.format)
age_demographic
```

Out[15]:

	Total Count	Percentage of Players
Age Group		
< 9	17	2.95 %
10 to 14	22	3.59 %
15 to 19	107	17.44 %
20 to 24	258	46.79 %
25 to 29	77	12.95 %
30 to 34	52	9.36 %
35 to 39	31	5.26 %
40 +	12	1.67 %

In [17]:

```
# Purchasing Analysis by Age
purchase_count = data.groupby(['Age Group'])['Age'].count()
avg_purchase_price = data.groupby(['Age Group'])['Price'].mean()
tot_purchase_value = data.groupby(['Age Group'])['Price'].sum()
avg_tot_person = tot_purchase_value/age_count
purchasing_analysis = pd.DataFrame({'Purchase Count': purchase_count, 'A
verage Purchase Price':avg_purchase_price,
                                   'Total Purchase Value': tot_purchase_
value, 'Avg Total Purchase/Person':
                                   avg_tot_person})
purchasing_analysis
```

Out[17]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase/Person
Age Group				
< 9	23	3.353478	77.13	4.537059
10 to 14	28	2.956429	82.78	3.762727
15 to 19	136	3.035956	412.89	3.858785
20 to 24	365	3.052219	1114.06	4.318062
25 to 29	101	2.900990	293.00	3.805195
30 to 34	73	2.931507	214.00	4.115385
35 to 39	41	3.601707	147.67	4.763548
40 +	13	2.941538	38.24	3.186667

In [18]:

```python
purchasing_analysis['Average Purchase Price'] = purchasing_analysis['Average Purchase Price'].map('$ {:.2f}'.format)
purchasing_analysis['Avg Total Purchase/Person'] = purchasing_analysis['Avg Total Purchase/Person'].map('$ {:.2f}'.format)
purchasing_analysis['Total Purchase Value'] = purchasing_analysis['Total Purchase Value'].map('$ {:.2f}'.format)

purchasing_analysis
```

Out[18]:

Age Group	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase/Person
< 9	23	$ 3.35	$ 77.13	$ 4.54
10 to 14	28	$ 2.96	$ 82.78	$ 3.76
15 to 19	136	$ 3.04	$ 412.89	$ 3.86
20 to 24	365	$ 3.05	$ 1114.06	$ 4.32
25 to 29	101	$ 2.90	$ 293.00	$ 3.81
30 to 34	73	$ 2.93	$ 214.00	$ 4.12
35 to 39	41	$ 3.60	$ 147.67	$ 4.76
40 +	13	$ 2.94	$ 38.24	$ 3.19

In [19]:

```python
#Grouping Players to have the posibility to find Top Spenders
purchase_count = data.groupby(['SN'])['Item ID'].count()
avg_purchase_price = data.groupby(['SN'])['Price'].mean()
tot_purchase_value = data.groupby(['SN'])['Price'].sum()
spenders = pd.DataFrame({'Purchase Count':purchase_count, 'Average Purchase Price': avg_purchase_price,
                         'Total Purchase Value':tot_purchase_value })
spenders.head()
```

Out[19]:

SN	Purchase Count	Average Purchase Price	Total Purchase Value
Adairialis76	1	2.28	2.28
Adastirin33	1	4.48	4.48
Aeda94	1	4.91	4.91
Aela59	1	4.32	4.32
Aelaria33	1	1.79	1.79

In [20]:

```python
# Top Spenders
best = spenders.sort_values('Total Purchase Value', ascending = False)
best.head()
#top_spenders = best5.head().style.format({'Average Purchase Price':'$ {:.2f}',
#                            'Total Purchase Value':'$ {:.2f}'})
#top_spenders
```

Out[20]:

SN	Purchase Count	Average Purchase Price	Total Purchase Value
Lisosia93	5	3.792000	18.96
Idastidru52	4	3.862500	15.45
Chamjask73	3	4.610000	13.83
Iral74	4	3.405000	13.62
Iskadarya95	3	4.366667	13.10

In [21]:

```python
best['Average Purchase Price'] = best['Average Purchase Price'].map('$ {:.2f}'.format)
best['Total Purchase Value'] = best['Total Purchase Value'].map('$ {:.2f}'.format)
best.head()
```

Out[21]:

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 [22]:

```
# Most Popular Items
popular = data[["Item ID", 'Item Name', 'Price']].groupby(['Item ID','Item Name'])
purchase_count = popular['Item ID'].count()
total_purchase_value = popular['Price'].sum()
item_price = total_purchase_value/purchase_count
most_popular = pd.DataFrame({'Purchase Count':purchase_count,
                             'Item Price': item_price, 'Total Purchase Value':total_purchase_value })
most_popular_item = most_popular.sort_values('Purchase Count', ascending = False)#.head().style.format({'Item Price':'$ {:.2f}',

#'Total Purchase Value':'$ {:.2f}' })
most_popular_item.head()
```

Out[22]:

		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 [23]:

```
most_popular_item['Item Price'] = most_popular_item['Item Price'].map('$ {:.2f}'.format)
most_popular_item['Total Purchase Value'] = most_popular_item['Total Purchase Value'].map('$ {:.2f}'.format)
most_popular_item.head()
```

Out[23]:

		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 [24]:

```
#Most Prifitable Items
most_profitable = most_popular.sort_values('Total Purchase Value', ascending = False)#.head().style.format({'Item Price':'$ {:.2f}',

#'Total Purchase Value':'$ {:.2f}' })
most_profitable.head()
```

Out[24]:

		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

In [25]:

```
most_profitable['Item Price'] = most_profitable['Item Price'].map('$ {:.2f}'.format)
most_profitable['Total Purchase Value'] = most_profitable['Total Purchase Value'].map('$ {:.2f}'.format)
most_profitable.head()
```

Out[25]:

		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

In [ ]:

In [ ]: