In [340]:

```
import os
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

In [341]:

```
raw_data_path = "purchase_data.json"

raw_data = pd.read_json(raw_data_path)
```

In [342]:

```
raw_data_df = pd.DataFrame(raw_data)
```

In [343]:

```
raw_data_df.head()
```

Out[343]:

	Age	Gender	Item ID	Item Name	Price	SN
0	38	Male	165	Bone Crushing Silver Skewer	3.37	Aelalis34
1	21	Male	119	Stormbringer, Dark Blade of Ending Misery	2.32	Eolo46
2	34	Male	174	Primitive Blade	2.46	Assastnya25
3	21	Male	92	Final Critic	1.36	Pheusrical25
4	23	Male	63	Stormfury Mace	1.27	Aela59

In [344]:

```
deduped_df = raw_data_df.drop_duplicates(["SN"])
no_of_players_df = pd.DataFrame({"Total Players":[len(deduped_df)]})
no_of_players_df
```

Out[344]:

	Total Players
0	573

In [345]:

```
total_players = 573
unique_items = raw_data_df["Item ID"].unique()
no_of_items = len(unique_items)
average_price = raw_data_df["Price"].mean()
total_purchases = raw_data_df["Price"].count()
total_revenue = raw_data_df["Price"].sum()
topline_items_df = pd.DataFrame([{"Number of Unique Items":no_of_item
s,
                                  "Average Item Price":average_price,
                                  "Total Purchases":total_purchases,
                                  "Total Revenue":total_revenue,}])

topline_items_df.style.format({"Average Item Price":"${:,.2f}", "Tota
l Revenue":"${:,.2f}"})
```

Out[345]:

	Average Item Price	Number of Unique Items	Total Purchases	Total Revenue
0	$2.93	183	780	$2,286.33

In [346]:

```python
#Create dataframes broken down by gender
male_data_df = raw_data_df.loc[raw_data_df["Gender"]=="Male"]
female_data_df = raw_data_df.loc[raw_data_df["Gender"]=="Female"]
nd_data_df = raw_data_df.loc[(raw_data_df["Gender"]!="Male") & (raw_d
ata_df["Gender"]!="Female")]

#Find number of unique players by gender
unique_male_players = male_data_df["SN"].unique()
total_male_players = len(unique_male_players)
unique_female_players = female_data_df["SN"].unique()
total_female_players = len(unique_female_players)
unique_nd_players = nd_data_df["SN"].unique()
total_nd_players = len(unique_nd_players)

gender_breakdown_df = pd.DataFrame({"Gender":["Male","Female","Other
 / Non-Disclosed"],
                                   "Total Players":[total_male_player
s,total_female_players,total_nd_players],
                                   "Percentage of Players":[(total_ma
le_players/total_players)*100,(total_female_players/total_players)*10
0,(total_nd_players/total_players)*100]})

gender_breakdown_df.style.format({"Percentage of Players":"{:,.2f}%"
})
```

Out[346]:

	Gender	Percentage of Players	Total Players
0	Male	81.15%	465
1	Female	17.45%	100
2	Other / Non-Disclosed	1.40%	8

In [347]:

```python
#Find averages per gender
m_avg_price = male_data_df["Price"].mean()
f_avg_price = female_data_df["Price"].mean()
nd_avg_price = nd_data_df["Price"].mean()

m_total_revenue = male_data_df["Price"].sum()
f_total_revenue = female_data_df["Price"].sum()
nd_total_revenue = nd_data_df["Price"].sum()
m_total_norm = m_total_revenue / total_male_players
f_total_norm = f_total_revenue / total_female_players
nd_total_norm = nd_total_revenue / total_nd_players

purchases_gender_df = pd.DataFrame({"Gender":["Male","Female","Other
 / Non-Disclosed"],
                                    "Total Players":[total_male_playe
rs,total_female_players,total_nd_players],
                                    "Average Purchase Price":[m_avg_p
rice,f_avg_price,nd_avg_price],
                                    "Total Purchase Value":[m_total_r
evenue,f_total_revenue,nd_total_revenue],
                                    "Normalized Total":[m_total_norm,
f_total_norm,nd_total_norm]})

purchases_gender_df.style.format({"Average Purchase Price":"${:,.2f}"
,"Total Purchase Value":"${:,.2f}","Normalized Total":"${:,.2f}"})
```

Out[347]:

	Average Purchase Price	Gender	Normalized Total	Total Players	Total Purchase Value
0	$2.95	Male	$4.02	465	$1,867.68
1	$2.82	Female	$3.83	100	$382.91
2	$3.25	Other / Non-Disclosed	$4.47	8	$35.74

In [348]:

```python
age_bins = [0,10,15,20,25,30,35,40,45,50,1000]
age_labels=["<10","10-14","15-19","20-24","25-29","30-34","35-39","40
-44","45-49",">50"]

deduped_df["Age Range"] = pd.cut(deduped_df["Age"], age_bins, labels
= age_labels, right = False)
raw_data_df["Age Range"] = pd.cut(raw_data_df["Age"], age_bins, label
s = age_labels, right = False)

group_by_ages = raw_data_df.groupby("Age Range")
unique_age_range_members = deduped_df.groupby("Age Range").size()
age_avg_price = group_by_ages["Price"].mean()
age_total_revenue = group_by_ages["Price"].sum()
age_total_purchases = group_by_ages.size()
age_revenue_norm = age_total_revenue / unique_age_range_members

population_breakdown_by_age = pd.DataFrame({"Percentage of Players":
(unique_age_range_members / total_players) * 100,
                                            "Total Individuals": uniqu
e_age_range_members})
population_breakdown_by_age.style.format({"Percentage of Players":"
{:,.2f}%"})
```

Out[348]:

Age Range	Percentage of Players	Total Individuals
<10	3.32%	19
10-14	4.01%	23
15-19	17.45%	100
20-24	45.20%	259
25-29	15.18%	87
30-34	8.20%	47
35-39	4.71%	27
40-44	1.75%	10
45-49	0.17%	1
>50	0.00%	0

In [349]:

```
age_grouped_df = pd.DataFrame({"Average Price":age_avg_price,
                              "Total Purchases":age_total_purchases,
                              "Normalized Total":age_revenue_norm})
age_grouped_df = age_grouped_df.dropna()
age_grouped_df.style.format({"Average Price":"${:,.2f}","Normalized T
otal":"${:,.2f}"})
```

Out[349]:

	Average Price	Normalized Total	Total Purchases
Age Range			
<10	$2.98	$4.39	28
10-14	$2.77	$4.22	35
15-19	$2.91	$3.86	133
20-24	$2.91	$3.78	336
25-29	$2.96	$4.26	125
30-34	$3.08	$4.20	64
35-39	$2.84	$4.42	42
40-44	$3.19	$5.10	16
45-49	$2.72	$2.72	1

In [350]:

```python
per_person_group = raw_data_df.groupby("SN")
total_spent_per_sn = per_person_group["Price"].sum()
number_of_purchases_per_sn = raw_data_df.groupby("SN").size()
avg_purchase_price_per_sn = total_spent_per_sn / number_of_purchases_
per_sn
total_spent_per_sn_df = pd.DataFrame({"Number of Purchases":number_of
_purchases_per_sn,
                                      "Total Purchase Value":total_sp
ent_per_sn,
                                      "Average Purchase Price": avg_p
urchase_price_per_sn})
total_spent_per_sn_df = total_spent_per_sn_df.sort_values(by="Total P
urchase Value", ascending=False)

#total_spent_per_sn_df.style.format({"Average Purchase Price":"${:,.2
f}",
#                                    "Total Purchase Value":"${:,.2
f}",})

total_spent_per_sn_df["Total Purchase Value"] = total_spent_per_sn_df
["Total Purchase Value"].map('${:,.2f}'.format)
total_spent_per_sn_df["Average Purchase Price"] = total_spent_per_sn_
df["Average Purchase Price"].map('${:,.2f}'.format)
total_spent_per_sn_df.head()
```

Out[350]:

	Average Purchase Price	Number of Purchases	Total Purchase Value
SN			
Undirrala66	$3.41	5	$17.06
Saedue76	$3.39	4	$13.56
Mindimnya67	$3.18	4	$12.74
Haellysu29	$4.24	3	$12.73
Eoda93	$3.86	3	$11.58

In [351]:

```
by_item_group = raw_data_df.groupby("Item ID")
total_revenue_per_item = by_item_group["Price"].sum()
number_of_item_sold = by_item_group["Item ID"].size()
item_name = by_item_group["Item Name"].first()
price_of_item = by_item_group["Price"]

popular_items_df = pd.DataFrame({"Item Name":item_name,
                                 "Total Purchase Value": total_rev
enue_per_item,
                                 "Number Sold":number_of_item_sold
})
popular_items_df = popular_items_df.sort_values(by="Number Sold", asc
ending=False)
profitable_items_df = popular_items_df.sort_values(by="Total Purchase
 Value", ascending=False)
popular_items_df["Total Purchase Value"] = popular_items_df["Total Pu
rchase Value"].map('${:,.2f}'.format)

popular_items_df.head()
```

Out[351]:

	Item Name	Number Sold	Total Purchase Value
Item ID			
39	Betrayal, Whisper of Grieving Widows	11	$25.85
84	Arcane Gem	11	$24.53
31	Trickster	9	$18.63
175	Woeful Adamantite Claymore	9	$11.16
13	Serenity	9	$13.41

In [352]:

```
profitable_items_df["Total Purchase Value"] = profitable_items_df["To
tal Purchase Value"].map('${:,.2f}'.format)
profitable_items_df.head()
```

Out[352]:

Item ID	Item Name	Number Sold	Total Purchase Value
34	Retribution Axe	9	$37.26
115	Spectral Diamond Doomblade	7	$29.75
32	Orenmir	6	$29.70
103	Singed Scalpel	6	$29.22
107	Splitter, Foe Of Subtlety	8	$28.88