In [257]:
```
 # Import Dependencies
import pandas as pd
import numpy as np
import glob, os
```

In [258]:
```
 # Create a recursive call to read all json files
path = 'raw_data'
ls_all_json = glob.glob(os.path.join(path, "*.json"), recursive=True)

df_heroes_temp = (pd.read_json(i) for i in ls_all_json)
df_heroes = pd.concat(df_heroes_temp, ignore_index=True)

#Cleaning data, drop empty rows
df_heroes = df_heroes.dropna(how="any")
df_heroes.head()
```

Out[258]:

	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 [259]:
```
# Player Count
# Total Number of Players

#rename columns into a more understandable names
df_heroes = df_heroes.rename(columns={"SN":"Player"})

#Group by "Player" to obtain unique values
total_players = df_heroes.groupby(["Player"]).count()

#Count and store in dataframe
df_total_players = pd.DataFrame({"Total Players":[len(total_players)]})
df_total_players
```

Out[259]:

	Total Players
0	612

In [260]:
```
# Purchasing Analysis (Total)
# Number of Unique Items
# Average Purchase Price
# Total Number of Purchases
# Total Revenue

#Group by "Item ID" to obtain unique values
#**There are Item Names with the same Item ID, this could generate a confus
ion when showing results and making statistics
#**I counted Item ID instead of Item Name
total_items = df_heroes.groupby(["Item ID"]).count()
total_items=len(total_items)

#Get total average price
avg_price = df_heroes["Price"].mean()

#Get total purchases
total_purchases=len(df_heroes)

#Get total revenue
total_revenue = df_heroes["Price"].sum()

#Numbers into money format
pd.options.display.float_format = '${:,.2f}'.format
df_heroes_summary = pd.DataFrame({"Number of Unique Items":[total_items],"A
verage Purchase Price":[round(avg_price,2)]
                                  ,"Total Number of Purchases":[total_purcha
ses],"Total Revenue":[round(total_revenue,2)]})
#Showing results
df_heroes_summary[["Number of Unique Items","Average Purchase Price","Total
 Number of Purchases","Total Revenue"]]
```

Out[260]:

	Number of Unique Items	Average Purchase Price	Total Number of Purchases	Total Revenue
0	184	$2.93	858	$2,514.43

In [261]:
```
# Gender Demographics
# Percentage and Count of Male Players
# Percentage and Count of Female Players
# Percentage and Count of Other / Non-Disclosed

#Aggregating data by Gender
df_heroes_gender_data=df_heroes.groupby(["Gender"])

#Storage results into data frame
df_heroes_gender_count = pd.DataFrame(df_heroes_gender_data["Item ID"].count
())

#Rename columns for printing
df_heroes_gender_count =df_heroes_gender_count.rename(columns={"Item ID":"Tota
l Count"})









#Create a new column with percentage of players
df_heroes_gender_count["Percentage of Players"] = (df_heroes_gender_count['Tot
al Count']/total_purchases)*100

#Format percentage
pd.options.display.float_format = '%{:,.2f}'.format

#Sorting values descending
df_heroes_gender_count = df_heroes_gender_count.sort_values("Total Count",asce
nding=False)

#Show results
df_heroes_gender_count[["Percentage of Players","Total Count"]]
```

Out[261]:

	Percentage of Players	Total Count
Gender		
Male	%81.24	697
Female	%17.37	149
Other / Non-Disclosed	%1.40	12

In [262]:
```
# Purchasing Analysis (Gender)
# The below each broken by gender
# Purchase Count
# Average Purchase Price
# Total Purchase Value
# Normalized Totals

#With above dataframe (gender), aggregating by price
revenue_gender = df_heroes_gender_data["Price"].sum()

#Count values per gender
gender_counts = df_heroes["Gender"].value_counts()

#Data frame to storage values
df_purch_analysis = pd.DataFrame({"Purchase Count":gender_counts,
                                  "Total Purchase Value":revenue_gender})
#Applying format
pd.options.display.float_format = '${:,.2f}'.format

#Create a new column with average purchase price, sorting and showing resul
ts
df_purch_analysis["Average Purchase Price"] = df_purch_analysis["Total Purc
hase Value"] /  df_purch_analysis["Purchase Count"]
df_purch_analysis = df_purch_analysis.sort_values("Total Purchase Value",as
cending=False)
df_purch_analysis.head()
```

Out[262]:

	Purchase Count	Total Purchase Value	Average Purchase Price
Male	697	$2,052.28	$2.94
Female	149	$424.29	$2.85
Other / Non-Disclosed	12	$37.86	$3.15

In [263]:
```
# Age Demographics
# The below each broken into bins of 4 years (i.e. <10, 10-14, 15-19, etc.)
# Purchase Count
# Average Purchase Price
# Total Purchase Value
# Normalized Totals


# Create the bins in which Data will be held
bins = [0, 12, 20, 30,100]

# Create the names for the four bins
group_names = ['Kid (<12)', 'Teenager (12-20)', 'Young (21-30)', 'Adult (>30)'
]

#Applying bins into main data frame
df_heroes["Age Ranges"] = pd.cut(df_heroes["Age"], bins, labels=group_names)

#With above dataframe (heroes), aggregating by Age Ranges
df_heroes_ranges_data=df_heroes.groupby(["Age Ranges"])

#Aggregating by price
revenue_ranges = df_heroes_ranges_data["Price"].sum()

#Count values per age ranges
ranges_counts = df_heroes["Age Ranges"].value_counts()

#Data frame to storage values
df_age_demographics = pd.DataFrame({"Purchase Count":ranges_counts,
                                    "Total Purchase Value":revenue_ranges})

#Applying format
pd.options.display.float_format = '${:,.2f}'.format

#Create a new column with average purchase price, sorting and showing results
df_age_demographics["Average Purchase Price"] = df_age_demographics["Total Pur
chase Value"] /  df_age_demographics["Purchase Count"]
df_age_demographics = df_age_demographics.sort_values("Total Purchase Value",a
scending=False)
df_age_demographics
```

Out[263]:

	Purchase Count	Total Purchase Value	Average Purchase Price
Young (21-30)	418	$1,233.62	$2.95
Teenager (12-20)	269	$764.86	$2.84
Adult (>30)	117	$350.58	$3.00
Kid (<12)	54	$165.37	$3.06

In [264]:
```
# Top Spenders
# Identify the the top 5 spenders in the game by total purchase value, then
 list (in a table):
# SN
# Purchase Count
# Average Purchase Price
# Total Purchase Value

#Data frame by Player(SN)
df_top_spenders_data=df_heroes.groupby(["Player"])

#Aggregating by price
revenue_spenders = df_top_spenders_data["Price"].sum()


#Count values per player
spenders_counts = df_heroes["Player"].value_counts()


#Data frame to storage values
df_top_spenders = pd.DataFrame({"Purchase Count":spenders_counts,
                                "Total Purchase Value":revenue_spenders
})


#Create a new column with average purchase price, sorting and showing resul
ts
df_top_spenders["Average Purchase Price"] = df_top_spenders["Total Purchase
 Value"] /  df_top_spenders["Purchase Count"]
df_top_spenders = df_top_spenders.sort_values("Total Purchase Value",ascend
ing=False)
df_top_spenders[["Purchase Count","Average Purchase Price","Total Purchase
 Value"]].head(5)
```

Out[264]:

	Purchase Count	Average Purchase Price	Total Purchase Value
Undirrala66	5	$3.41	$17.06
Aerithllora36	4	$3.77	$15.10
Saedue76	4	$3.39	$13.56
Sondim43	4	$3.25	$13.02
Mindimnya67	4	$3.18	$12.74

In [265]:
```
# Most Popular Items
# Identify the 5 most popular items by purchase count, then list (in a table):
# Item ID
# Item Name
# Purchase Count
# Item Price
# Total Purchase Value

#Data frame by Item Name(SN) with multiple aggregating
df_popular_items_data= df_heroes.groupby(['Item Name'])['Price'].agg(['sum','count','mean'])

#Data frame to storage values
df_popular_items = pd.DataFrame({"Purchase Count":df_popular_items_data["count"],
                                 "Total Purchase Value":df_popular_items_data["sum"],
                                   "Avg Item Value":df_popular_items_data["mean"]})

#sorting and showing results (top 5)
df_popular_items.sort_values("Purchase Count",ascending=False).head(5)
```

Out[265]:

	Avg Item Value	Purchase Count	Total Purchase Value
Item Name			
Final Critic	$2.76	14	$38.60
Arcane Gem	$2.45	12	$29.34
Stormcaller	$3.35	12	$40.19
Betrayal, Whisper of Grieving Widows	$2.35	11	$25.85
Trickster	$2.32	10	$23.22

In [266]:
```
# Most Profitable Items
# Identify the 5 most profitable items by total purchase value, then list (in
 a table):
# Item ID
# Item Name
# Purchase Count
# Item Price
# Total Purchase Value

#Same dataframe above
#sorting and showing results (top 5)
df_popular_items.sort_values("Total Purchase Value",ascending=False).head(5)
```

Out[266]:

	Avg Item Value	Purchase Count	Total Purchase Value
Item Name			
Stormcaller	$3.35	12	$40.19
Final Critic	$2.76	14	$38.60
Retribution Axe	$4.14	9	$37.26
Splitter, Foe Of Subtlety	$3.67	9	$33.03
Spectral Diamond Doomblade	$4.25	7	$29.75