# HeroesOfPymoli

February 13, 2018

In [1]:
```
# Import Dependencies
import pandas as pd
import os
```

In [2]:
```
# Create a reference the JSON file desired
json_path = os.path.join('Resources', 'purchase_data.json')

# Read the CSV into a Pandas DataFrame
purchase_data_df = pd.read_json(json_path)
```

In [3]:
```
# Print the first five rows of the first data frame to the screen
purchase_data_df.head()
```

Out[3]:

| | 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 [4]:
```
#Counting the total number of unique players
purchase_data_df["SN"].nunique()
```

Out[4]: 573

In [5]:
```
#Creating the values for the purchasing analysis summary table
unique_items = purchase_data_df["Item Name"].nunique()
avg_pur_price = purchase_data_df["Price"].mean()
total_purcahse = purchase_data_df["Price"].count()
total_rev = purchase_data_df["Price"].sum()
```

```
#Create summary table for purchasing analysis
summary_df = pd.DataFrame ({"Unique Items": [unique_items],
                            "Average Purchase Price": [avg_pur_price],
                            "Number of Purchases": [total_purcahse],
                            "Total Revenue": [total_rev]})

#Arrange columns in desired order and print summary table
summary_df_2 = summary_df[["Unique Items", "Average Purchase Price", "Number of Purchas
summary_df_2

summary_df_2.style.format({"Average Purchase Price": "${:.2f}", "Total Revenue": "${:.
```

Out[5]: <pandas.io.formats.style.Styler at 0x106285828>

```
In [6]: #Gender Analytics
total_gender = purchase_data_df["Gender"].count()
male = purchase_data_df["Gender"].value_counts()['Male']
female = purchase_data_df["Gender"].value_counts()['Female']
other = purchase_data_df["Gender"].value_counts()['Other / Non-Disclosed']
male_percent = (male/total_gender) * 100
female_percent = (female/total_gender) * 100
other_percent = (other/total_gender) * 100

#Create gender analytics summary table
summary_gender_df = pd.DataFrame ({"Percentage of Players": [male_percent, female_perc
                                   "Total Count": [male, female, other]}, index = ["Mal

summary_gender_df.style.format({"Percentage of Players": "%{:.2f}"})

summary_gender_df
```

Out[6]:

| | Percentage of Players | Total Count |
|---|---|---|
| Male | 81.153846 | 633 |
| Female | 17.435897 | 136 |
| Non-Specific | 1.410256 | 11 |

```
In [7]: #Purchasing Analysis by Gender

pur_gender = purchase_data_df.groupby(["Gender"])

pur_count_gender = pur_gender["SN"].count()

avg_pur_price = pur_gender["Price"].mean()

tot_pur_value = pur_gender["Price"].sum()

duplicate_data = purchase_data_df.drop_duplicates(subset='SN', keep="first")
grouped_duplicate_data = duplicate_data.groupby(["Gender"])
```

```
        norm_total = (pur_gender["Price"].sum() / grouped_duplicate_data["SN"].count())

        summary_analysis_gender_df = pd.DataFrame({ "Purchase Count":pur_count_gender,
                                                    "Average Purchase Price": avg_pur_price,
                                                    "Total Purchase Value": tot_pur_value,
                                                    "Normalized Total": norm_total })

        summary_analysis_gender_df

        summary_analysis_gender_df.style.format({"Total Purchase Value": '${:.2f}', "Average Pu
```

Out[7]: <pandas.io.formats.style.Styler at 0x106c6c390>

```
In [8]: #Age Demographics by Binning

        bins = [0,10,15,20,25,30,35,40,200]
        bins_list = ["< 10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]

        #Bin dataframe
        bin_df = purchase_data_df.copy()
        bin_df["Age Groups"] = pd.cut(bin_df["Age"], bins, labels =bins_list)
        bin_cut = pd.cut(bin_df["Age"], bins, labels = bins_list)
        grouped_bin_df = bin_df.groupby(["Age Groups"])

        pur_count_age = grouped_bin_df["Age"].count()
        avg_price_age = grouped_bin_df["Price"].mean()
        tot_pur_age = grouped_bin_df["Price"].sum()

        duplicate = purchase_data_df.drop_duplicates(subset = 'SN', keep = "first")
        duplicate["Age Groups"] = pd.cut(duplicate["Age"], bins, labels = bins_list)
        duplicate = duplicate.groupby(["Age Groups"])

        norm_total_bin_df = (grouped_bin_df["Price"].sum()/duplicate["SN"].count())
        norm_total_bin_df

        age_bin_df = pd.DataFrame({ "Purchase Count": pur_count_age,
                                    "Average Purchase Price": avg_price_age,
                                    "Total Purchase Value": tot_pur_age,
                                    "Normalized Total": norm_total_bin_df})

        age_bin_df

        age_bin_df.style.format({"Total Purchase Value": '${:.2f}', "Average Purchase Price":
```

```
/anaconda3/envs/PythonData/lib/python3.6/site-packages/ipykernel_launcher.py:17: SettingWithCo
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html
```

Out[8]: <pandas.io.formats.style.Styler at 0x106d0f588>

In [9]:
```
#Top Spenders

SN_group = purchase_data_df.groupby(["SN"])

SN_group_count = SN_group["Item ID"].count()
SN_group_total = SN_group["Price"].sum()
SN_group_Average = (SN_group_total/SN_group_count)

spender_df = pd.DataFrame({"Purchase Count": SN_group_count,
                           "Average Purchase Price": SN_group_Average,
                           "Total Purchase Value":SN_group_total})

spender_df = spender_df.sort_values("Total Purchase Value", ascending = False)

spender_df.style.format({"Total Purchase Value": '${:.2f}', "Average Purchase Price":

spender_df.head(5)
```

Out[9]:

| SN | Average Purchase Price | Purchase Count | Total Purchase Value |
|---|---|---|---|
| Undirrala66 | 3.412000 | 5 | 17.06 |
| Saedue76 | 3.390000 | 4 | 13.56 |
| Mindimnya67 | 3.185000 | 4 | 12.74 |
| Haellysu29 | 4.243333 | 3 | 12.73 |
| Eoda93 | 3.860000 | 3 | 11.58 |

In [10]:
```
#Most Popular Items

top5_items_ID = pd.DataFrame(purchase_data_df.groupby("Item ID")["Item ID"].count())

top5_items_ID.sort_values("Item ID", ascending = False, inplace = True)

top5_items_ID = top5_items_ID.iloc[0:5][:]

top5_items_total = pd.DataFrame(purchase_data_df.groupby("Item ID")["Price"].sum())

top5_items = pd.merge(top5_items_ID, top5_items_total, left_index = True, right_index

no_dup_items = purchase_data_df.drop_duplicates(["Item ID"], keep = 'last')

top5_merge_ID = pd.merge(top5_items, no_dup_items, left_index = True, right_on = "Iten

top5_merge_ID = top5_merge_ID[["Item ID", "Item Name", "Item ID_x", "Price_y", "Price_
```

```
top5_merge_ID.set_index(["Item ID"], inplace = True)

top5_merge_ID.rename(columns =  {"Item ID_x": "Purchase Count",
                                 "Price_y": "Item Price",
                                 "Price_x": "Total Purchase Value"}, inplace=True)

top5_merge_ID.style.format({"Item Price": '${:.2f}', "Total Purchase Value": '${:.2f}
```

```
Out[10]: <pandas.io.formats.style.Styler at 0x106281e48>
```

```
In [11]: #Most Profitable

top5_profit = pd.DataFrame(purchase_data_df.groupby("Item ID")["Price"].sum())
top5_profit.sort_values("Price", ascending = False, inplace = True)

top5_profit = top5_profit.iloc[0:5][:]

pur_count_profit = pd.DataFrame(purchase_data_df.groupby("Item ID")["Item ID"].count()

top5_profit = pd.merge(top5_profit, pur_count_profit, left_index = True, right_index =
top5_merge_profit = pd.merge(top5_profit, no_dup_items, left_index = True, right_on =
top5_merge_profit = top5_merge_profit[["Item ID", "Item Name", "Item ID_x", "Price_y"
top5_merge_profit.set_index(["Item ID"], inplace=True)
top5_merge_profit.rename(columns = {"Item ID_x": "Purchase Count",
                                    "Price_y": "Item Price",
                                    "Price_x": "Total Purchase Value"}, inplace = Tru
top5_merge_profit.style.format({"Item Price": '${:.2f}', "Total Purchase Value": '${:
```

```
Out[11]: <pandas.io.formats.style.Styler at 0x106d0fa58>
```