Heroes of Pymoli

February 19, 2018

In [1]:
```
#Three Observable Trends
#Players that are between ages 20-24 purchased the most amount of items.
#The Most Popular Items were Betrayal,Whisper of Grieving Widows and Arcane Gem but th
#was the Retribution Axe.
#Most players were male.
```

In [2]:
```
#state dependencies
import pandas as pd
import numpy as np
```

In [3]:
```
#declare file path
file_path="Resources/purchase_data.json"
#import file as dataframe
Heroes_df=pd.read_json(file_path)
```

In [4]:
```
#Total number of players
total=Heroes_df["SN"].count()
total_df=pd.DataFrame({"Total Number of Players":[total]})
total_df.head()
```

Out[4]:

	Total Number of Players
0	780

In [5]:
```
#Purchasing Analysis(Total)

#Number of Unique Items
unique_items=Heroes_df["Item Name"].unique()
number_of_unique_items=len(unique_items)

#Average Purchase Price
aver_purchase_price=Heroes_df["Price"].mean()

#Total Number of Purchases
number_of_purchases=Heroes_df["Price"].count()

#Total Revenue
total_revenue=Heroes_df["Price"].sum()
```

```
#Putting these aspects into a new DataFrame
purchasing_analysis_total_df=pd.DataFrame({"Number of Unique Items":[number_of_unique_
                                           "Average Purchase Price":[aver_purchase_price]
                                           "Total Number of Purchases":[number_of_purchase
                                           "Total Revenue":[total_revenue]})
purchasing_analysis_total_df.head()
```

Out[5]:

	Average Purchase Price	Number of Unique Items	Total Number of Purchases	Total Revenue
0	2.931192	179	780	2286.33

In [6]:
```
#Gender Demographics

#Percentage and Count of Male Players

male_players=Heroes_df.loc[Heroes_df["Gender"] == "Male",:]
num_of_male_players=male_players["Gender"].count()

percent_of_male_players=(num_of_male_players/total)*100


#Percentage and Count of Female Players

female_players=Heroes_df.loc[Heroes_df["Gender"] == "Female",:]
num_of_female_players=female_players["Gender"].count()

percent_of_female_players=(num_of_female_players/total)*100


#Percentage and Count of Other/Non-Disclosed

other_players=Heroes_df.loc[Heroes_df["Gender"] == "Other / Non-Disclosed",:]
num_of_other_players=other_players["Gender"].count()

percent_of_other_players=(num_of_other_players/total)*100



#organizing the dataframe
gender_demo_df=pd.DataFrame({"Gender":["Male","Female","Other/Nondisclosed"],
                             "Total Count":[num_of_male_players,num_of_femal
                             "Percentage of Players":[percent_of_male_playe
                            })

gender_demo_group=gender_demo_df.set_index('Gender')
gender_demo_group.head()
```

Out[6]:

Gender	Percentage of Players	Total Count
Male	81.153846	633
Female	17.435897	136
Other/Nondisclosed	1.410256	11

In [7]:
```
#Purchasing Analysis(Gender)
#Purchase Counts for Each Gender
#Purchase Count of Males
gender_index=Heroes_df.set_index("Gender")
male_price_df=gender_index.loc[["Male"],["Price"]]
purchase_count_males=male_price_df["Price"].count()

#Purchase Count of Females
female_price_df=gender_index.loc[["Female"],["Price"]]
purchase_count_females=female_price_df["Price"].count()

#Purchase Count of Other
other_price_df=gender_index.loc[["Other / Non-Disclosed"],["Price"]]
purchase_count_other=other_price_df["Price"].count()

#Average Purchase Price
#Average Purchase Price of Males
avg_purchase_male=male_price_df["Price"].mean()

#Average Purchase Price of Females
avg_purchase_female=female_price_df["Price"].mean()

#Average Purchase Price of Other/Non-Disclosed
avg_purchase_other=other_price_df["Price"].mean()

#Total Purchase Value
#Total Purchase Value For Males
total_purchase_male=male_price_df["Price"].sum()

#Total Purchase Value For Females
total_purchase_female=female_price_df["Price"].sum()

#Total Purchase Value For Other/Non-Disclosed
total_purchase_other=other_price_df["Price"].sum()

#Normalized Totals
#Normalized Total for Males
std_dev_male=male_price_df["Price"].std()
norm_total_male=abs(((total_purchase_male-avg_purchase_male))/(std_dev_male))

#Normalized Total for Females
std_dev_female=female_price_df["Price"].std()
```

```
norm_total_female=abs(((total_purchase_female-avg_purchase_female))/(std_dev_female))
#Normalized Total for Other/Non-Disclosed
std_dev_other=other_price_df["Price"].std()
norm_total_other=abs(((total_purchase_other-avg_purchase_other))/(std_dev_other))


#organizing a dataframe
purchasing_analysis_gender=pd.DataFrame({"Gender":["Male","Female","Other/Nondisclosed"
                                          "Purchase Count":[purchase_count_males,purc
                                          "Average Purchase Price":[avg_purchase_male
                                          "Total Purchase Value":[total_purchase_male
                                          "Normalized Totals":[norm_total_male,norm_t
                                         })
purchasing_analysis_gender_group=purchasing_analysis_gender.set_index('Gender')

purchasing_analysis_gender_group.head()
```

Out[7]:

Gender	Average Purchase Price	Normalized Totals	Purchase Count	Total Purchase Value
Male	2.950521	1679.987030	633	1867.68
Female	2.815515	330.222006	136	382.91
Other/Nondisclosed	3.249091	33.942639	11	35.74

In [8]:
```
#Age Demographics
age_bins=[0,10,14,19,24,29,34,39,100]
age_labels=['<10','10-14','15-19','20-24','25-29','30-34','35-39','40+']
Heroes_df["Age Group"]=pd.cut(Heroes_df["Age"],age_bins, labels=age_labels)
age_group_index=Heroes_df.set_index('Age Group')

#Purchase Counts for each group
binone_df=age_group_index.loc[["<10"],["Price"]]
purchase_countbinone=binone_df["Price"].count()

bintwo_df=age_group_index.loc[["10-14"],["Price"]]
purchase_countbintwo=bintwo_df["Price"].count()

binthree_df=age_group_index.loc[["15-19"],["Price"]]
purchase_countbinthree=binthree_df["Price"].count()

binfour_df=age_group_index.loc[["20-24"],["Price"]]
purchase_countbinfour=binfour_df["Price"].count()
```

```
binfive_df=age_group_index.loc[["25-29"],["Price"]]
purchase_countbinfive=binfive_df["Price"].count()

binsix_df=age_group_index.loc[["30-34"],["Price"]]
purchase_countbinsix=binsix_df["Price"].count()

binseven_df=age_group_index.loc[["35-39"],["Price"]]
purchase_countbinseven=binseven_df["Price"].count()

bineight_df=age_group_index.loc[["40+"],["Price"]]
purchase_countbineight=bineight_df["Price"].count()


#Average Purchase Price for each group
avg_purchase_binone=binone_df["Price"].mean()

avg_purchase_bintwo=bintwo_df["Price"].mean()

avg_purchase_binthree=binthree_df["Price"].mean()

avg_purchase_binfour=binfour_df["Price"].mean()

avg_purchase_binfive=binfive_df["Price"].mean()

avg_purchase_binsix=binsix_df["Price"].mean()

avg_purchase_binseven=binseven_df["Price"].mean()

avg_purchase_bineight=bineight_df["Price"].mean()


#Total Purchase Value
totalpurchasebinone=binone_df["Price"].sum()

totalpurchasebintwo=bintwo_df["Price"].sum()

totalpurchasebinthree=binthree_df["Price"].sum()

totalpurchasebinfour=binfour_df["Price"].sum()

totalpurchasebinfive=binfive_df["Price"].sum()

totalpurchasebinsix=binsix_df["Price"].sum()

totalpurchasebinseven=binseven_df["Price"].sum()

totalpurchasebineight=bineight_df["Price"].sum()
```

```
#Normalized Totals
std_devbinone=binone_df["Price"].std()
norm_totalbinone=abs(((totalpurchasebinone-avg_purchase_binone))/(std_devbinone))

std_devbintwo=bintwo_df["Price"].std()
norm_totalbintwo=abs(((totalpurchasebintwo-avg_purchase_bintwo))/(std_devbintwo))

std_devbinthree=binthree_df["Price"].std()
norm_totalbinthree=abs(((totalpurchasebinthree-avg_purchase_binthree))/(std_devbinthree

std_devbinfour=binfour_df["Price"].std()
norm_totalbinfour=abs(((totalpurchasebinfour-avg_purchase_binfour))/(std_devbinfour))

std_devbinfive=binfive_df["Price"].std()
norm_totalbinfive=abs(((totalpurchasebinfive-avg_purchase_binfive))/(std_devbinfive))

std_devbinsix=binsix_df["Price"].std()
norm_totalbinsix=abs(((totalpurchasebinsix-avg_purchase_binsix))/(std_devbinsix))

std_devbinseven=binseven_df["Price"].std()
norm_totalbinseven=abs(((totalpurchasebinseven-avg_purchase_binseven))/(std_devbinseven

std_devbineight=bineight_df["Price"].std()
norm_totalbineight=abs(((totalpurchasebineight-avg_purchase_bineight))/(std_devbineight

#finally organizing a dataframe
age_demographics_df=pd.DataFrame({"Age Groups":['<10','10-14','15-19','20-24','25-29',
                                  "Purchase Count":[purchase_countbinone,purchase_count
                                  "Average Purchase Price":[avg_purchase_binone,avg_pur
                                  "Total Purchase":[totalpurchasebinone,totalpurchaseb
                                  "Normalized Totals":[norm_totalbinone,norm_totalbintw
                                 })
age_demographics_group=age_demographics_df.set_index("Age Groups")
age_demographics_group.head()
```

Out[8]:

Age Groups	Average Purchase Price	Normalized Totals	Purchase Count	Total Purchase
<10	3.019375	80.787100	32	96.62
10-14	2.702903	76.971819	31	
15-19	2.905414	342.115295	133	
20-24	2.913006	877.080674	336	
25-29	2.962640	322.047667	125	

```
10-14                83.79
15-19               386.42
20-24               978.77
25-29               370.33
```

In [9]:
```
#Top Spenders

#group by the SN
group_by_SN=Heroes_df.groupby("SN")
#reset indexes for sorting afterwards
totalprice=group_by_SN["Price"].sum().reset_index()
average_price=group_by_SN["Price"].mean().reset_index()
purchasecount=group_by_SN["Price"].count().reset_index()
#merge purchase count data and average price data
merge=pd.merge(average_price,purchasecount,on="SN")
#merge everything together
totalpricemerge=pd.merge(totalprice,merge,on="SN")
#sort the values by total price
highest_purchases=totalpricemerge.sort_values("Price",ascending=False)
#locates the top five total purchases
topfive=highest_purchases.iloc[0:5,:]
#rename the columns of the merge
top_five=topfive.rename(index=str, columns={'SN':'SN','Price':'Total Purchase Value','
#set the SN as an index
top_spenders=top_five.set_index('SN')
top_spenders
```

Out[9]:

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

In [10]:
```
#Most Popular Items
#group by Item ID and Item Name
group_by_items=Heroes_df.groupby(["Item ID","Item Name"])
#reset the indexes for sorting
purchase_count=group_by_items["Price"].count().reset_index()
item_price=group_by_items["Price"].min().reset_index()
total_purchase_value=group_by_items["Price"].sum().reset_index()
#merge item price and total purchase value data
merge=pd.merge(item_price,total_purchase_value, on=["Item ID","Item Name"])
#merge everything together
purchase_count_merge=pd.merge(purchase_count,merge,on=["Item ID","Item Name"])
#sort the values by purchase count
highest_purchase_counts=purchase_count_merge.sort_values("Price",ascending=False)
```

```
#locates the top five purchase counts
topfive=highest_purchase_counts.iloc[0:5,:]
#rename the columns
top_five=topfive.rename(index=str,columns={"Item ID":"Item ID","Item Name":"Item Name
#set Item ID and Item Name as indices
most_popular_items=top_five.set_index(["Item ID","Item Name"])
most_popular_items
```

Out[10]:

Item ID	Item Name	Purchase Count	Item Price	Total Purchase Value
39	Betrayal, Whisper of Grieving Widows	11	2.35	25.85
84	Arcane Gem	11	2.23	24.53
31	Trickster	9	2.07	18.63
175	Woeful Adamantite Claymore	9	1.24	11.16
13	Serenity	9	1.49	13.41

In [11]:
```
#Most Profitable Items
#merge item price and purchase count together
merge=pd.merge(purchase_count,item_price,on=["Item ID","Item Name"])
#merge everything together
total_purchase_value_merge=pd.merge(total_purchase_value,merge,on=["Item ID","Item Na
#sort by highest total purchase values
highest_total=total_purchase_value_merge.sort_values("Price",ascending=False)
#locate the top five highest total purchase values
topfive=highest_total.iloc[0:5,:]
#rename columns
top_five=topfive.rename(index=str,columns={"Item ID":"Item ID","Item Name":"Item Name
#set Item ID and Item Name as indices
most_profitable_items=top_five.set_index(["Item ID","Item Name"])
most_profitable_items
```

Out[11]:

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

Item ID	Item Name	
34	Retribution Axe	4.14
115	Spectral Diamond Doomblade	4.25
32	Orenmir	4.95
103	Singed Scalpel	4.87
107	Splitter, Foe Of Subtlety	3.61