

Heroes Of Pymoli Data Analysis (Solved)

- Observed trend #1 - Of the 573 people in the dataset, the vast majority are male (81.15%). There also exists, a smaller, but notable proportion of female players (17.45%).
- Observed trend #2 - Our peak purchase over age demographics is in the age group of '20-24' which has 336 purchase count and \$978.77 of total purchase value.
- Observed trend #3 - The total purchase value of Male (1,867.68 dollars) is more than female (382.91 dollars).

```
In [167]: import pandas as pd
          json_file = "purchase_data.json"
          df = pd.read_json(json_file)
          df.head()
```

Out[167]:

	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

Player Count

```
In [168]: total_players = df['SN'].unique()
total_players_count = total_players.size
player_count_df = pd.DataFrame([{'Total Players': total_players_count}])
# df w/ duplicates removed
removed_duplicates_df = df.drop_duplicates(['SN'], keep='first')
player_count_df
```

Out[168]:

	Total Players
0	573

Purchasing Analysis (Total)

```
In [169]: # format
money_fmt = "${:,.2f}".format
fmt = '{:.2f}'.format

unique_items = df['Item ID'].unique()
unique_items_count = unique_items.size
average_price = df['Price'].sum()/df['Price'].count()
purchases_count = df['Item ID'].count()
total_revenue = df['Price'].sum()

purchase_analysis_df = pd.DataFrame([[unique_items_count,money_fmt(average_price), purchases_count, money_fmt
(total_revenue)]],
                                     columns=['Number of Unique Items','Average Price','Number of Purchases',
'Total Revenue'])
# display
purchase_analysis_df
```

Out[169]:

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

Gender Demographics

```
In [170]: # use the built-in normalize in value_counts to get the percentage
percents = removed_duplicates_df['Gender'].value_counts(normalize=True)*100
# total count
total = removed_duplicates_df['Gender'].value_counts()

gender_demographics =total.to_frame()
gender_demographics= gender_demographics.rename(columns={'Gender':'Total Count'})
gender_demographics['Percentage of Player'] = percents.map(fmt)

# change the column order and display the df
gender_demographics = gender_demographics[['Percentage of Player','Total Count']]
gender_demographics
```

Out[170]:

	Percentage of Player	Total Count
Male	81.15	465
Female	17.45	100
Other / Non-Disclosed	1.40	8

Purchasing Analysis (Gender)

```

In [171]: # group by gender
group_by_gender = df.groupby(['Gender'])
purchase_analysis_df = pd.DataFrame()
# purchase count
purchase_analysis_df['Purchase Count']=group_by_gender['Item ID'].count()
# total purchase value
purchase_analysis_df['Total Purchase Value'] = group_by_gender['Price'].sum()
# average purchase price
purchase_analysis_df['Average Purchase Price'] = purchase_analysis_df['Total Purchase Value']/purchase_analysis_df['Purchase Count']
# normalization
purchase_analysis_df['Normalized Totals'] = purchase_analysis_df['Total Purchase Value']/gender_demographics['Total Count']

# formatting
purchase_analysis_df['Total Purchase Value'] = purchase_analysis_df['Total Purchase Value'].map(money_fmt)
purchase_analysis_df['Average Purchase Price'] = purchase_analysis_df['Average Purchase Price'].map(money_fmt)
purchase_analysis_df['Normalized Totals'] = purchase_analysis_df['Normalized Totals'].map(money_fmt)

# change the column order and display
purchase_analysis_df = purchase_analysis_df[['Purchase Count', 'Average Purchase Price', \
                                              'Total Purchase Value', 'Normalized Totals']]

purchase_analysis_df

```

Out[171]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Normalized Totals
Gender				
Female	136	\$2.82	\$382.91	\$3.83
Male	633	\$2.95	\$1,867.68	\$4.02
Other / Non-Disclosed	11	\$3.25	\$35.74	\$4.47

Age Demographics

```

In [172]: max_age = removed_duplicates_df['Age'].max()
# create bins and categories
bins = [0,9,14,19,24,29,34,39,max_age]
age_categories = ['<10', '10-14', '15-19', '20-24', '25-29', '30-34', '35-39', '40+']
# cerate a df for age demographics
age_demographics_df = pd.DataFrame()
# categorize
categorized_by_age_df = pd.cut(removed_duplicates_df['Age'], bins, labels=age_categories, right=True)

# add total count that is sorted by the index to the df
age_demographics_df['Total Count'] = categorized_by_age_df.value_counts().sort_index()
# percent and formatting
age_demographics_df['Percentage of Players'] = (categorized_by_age_df.value_counts(normalize=True)*100).map(fmt)

# change the column order and display
age_demographics_df = age_demographics_df[['Percentage of Players', 'Total Count']]
age_demographics_df

```

Out[172]:

	Percentage of Players	Total Count
<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+	1.92	11

Purchasing Analysis (Age)

```
In [173]: # create a df for this portion
purchase_analysis_df = pd.DataFrame()

max_age_in_duplicates = df['Age'].max()
# create new bins for this part
bins_in_duplicates = [0,9,14,19,24,29,34,39,max_age_in_duplicates]
# reuse the age category
age_categories_in_duplicates = age_categories

# categorize
categorized_by_age_in_duplicates_df = pd.cut(df['Age'],bins=bins_in_duplicates, labels=age_categories_in_duplicates, right=True)

# purchase count
purchase_analysis_df['Purchase Count'] = categorized_by_age_in_duplicates_df.value_counts().sort_index()

# total purchase value by age categories
purchase_analysis_df['Total Purchase Value'] = df.groupby(categorized_by_age_in_duplicates_df)['Price'].sum()

# average purchase price
purchase_analysis_df['Average Purchase Price'] = purchase_analysis_df['Total Purchase Value']/purchase_analysis_df['Purchase Count']

# normalized totals by age demographics
purchase_analysis_df['Normalized Totals'] = purchase_analysis_df['Total Purchase Value']/age_demographics_df['Total Count']

# formatting
purchase_analysis_df['Total Purchase Value'] = purchase_analysis_df['Total Purchase Value'].map(money_fmt)
purchase_analysis_df['Average Purchase Price'] = purchase_analysis_df['Average Purchase Price'].map(money_fmt)
purchase_analysis_df['Normalized Totals'] = purchase_analysis_df['Normalized Totals'].map(money_fmt)

# change the column order and display
purchase_analysis_df = purchase_analysis_df[['Purchase Count','Average Purchase Price', \
                                              'Total Purchase Value','Normalized Totals']]

purchase_analysis_df
```

Out[173]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Normalized Totals
<10	28	\$2.98	\$83.46	\$4.39
10-14	35	\$2.77	\$96.95	\$4.22
15-19	133	\$2.91	\$386.42	\$3.86
20-24	336	\$2.91	\$978.77	\$3.78
25-29	125	\$2.96	\$370.33	\$4.26
30-34	64	\$3.08	\$197.25	\$4.20
35-39	42	\$2.84	\$119.40	\$4.42
40+	17	\$3.16	\$53.75	\$4.89

Top Spenders

```

In [174]: # create a new df
top_spenders_df = pd.DataFrame()
# group by SN
group_by_SN_df = df.groupby(['SN'])

# total purchase value
total_purchase_value = group_by_SN_df['Price'].sum().sort_values(ascending=False).head(5)
top_spenders_df['Total Purchase Value'] = total_purchase_value

# purchase count
top_total_purchase_sns = df[df['SN'].isin(total_purchase_value.index)]
group_by_sns = top_total_purchase_sns.groupby(['SN'])
purchase_count = group_by_sns['Item ID'].count()
top_spenders_df['Purchase Count'] = purchase_count

# average purchase price
top_spenders_df['Average Purchase Price'] = total_purchase_value/purchase_count

# formatting
top_spenders_df['Average Purchase Price'] = top_spenders_df['Average Purchase Price'].map(money_fmt)
top_spenders_df['Total Purchase Value'] = top_spenders_df['Total Purchase Value'] .map(money_fmt)

# change the column order and display
top_spenders_df = top_spenders_df[['Purchase Count', 'Average Purchase Price', 'Total Purchase Value']]
top_spenders_df

```

Out[174]:

	Purchase Count	Average Purchase Price	Total Purchase Value
SN			
Undirralla66	5	\$3.41	\$17.06
Saedue76	4	\$3.39	\$13.56
Mindimnya67	4	\$3.18	\$12.74
Haellysu29	3	\$4.24	\$12.73
Eoda93	3	\$3.86	\$11.58

Most Popular Items

```
In [175]: group_by_id_and_name = df.groupby(['Item ID', 'Item Name'])
# create a new df for this portion of code
most_popular_items = pd.DataFrame()
# purchase count
purchase_count = group_by_id_and_name.count().sort_values('Price', ascending=False)['SN']
most_popular_items['Purchase Count'] = purchase_count

# total purchase value
top_total_purchase_sns = df[df['Item ID'].isin(purchase_count.index.get_level_values(0))]
group_by_sns = top_total_purchase_sns.groupby(['Item ID', 'Item Name'])
total_purchase_value = group_by_sns['Price'].sum()

# item price
item_price = total_purchase_value/purchase_count

# formatting
most_popular_items['Item Price'] = item_price.map(money_fmt)
most_popular_items['Total Purchase Value'] = total_purchase_value.map(money_fmt)

# display top 5
most_popular_items.head(5)
```

Out[175]:

		Purchase Count	Item Price	Total Purchase Value
Item ID	Item Name			
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

Most Profitable Items

```
In [176]: # create a new df
most_profitable_items = pd.DataFrame()

# purchase count
top_total_purchase_sns = df[df['Item ID'].isin(most_profitable_total_purchase_value.index.get_level_values(0))] # 0 --> Item ID
group_by_sns = top_total_purchase_sns.groupby(['Item ID', 'Item Name'])
purchase_count = group_by_sns['Item ID'].count()

# total purchase value
most_profitable_total_purchase_value = group_by_id_and_name['Price'].sum().sort_values(ascending=False)
most_profitable_items['Total Purchase Value'] = most_profitable_total_purchase_value

# item price
most_profitable_items['Item Price'] = most_profitable_total_purchase_value/purchase_count

# formatting
most_profitable_items['Purchase Count'] = purchase_count
most_profitable_items['Item Price'] = most_profitable_items['Item Price'].map(money_fmt)
most_profitable_items['Total Purchase Value'] = most_profitable_items['Total Purchase Value'].map(money_fmt)

# change the column order and display the top 5
most_profitable_items = most_profitable_items[['Purchase Count', 'Item Price', 'Total Purchase Value']]
most_profitable_items.head(5)
```

Out[176]:

		Purchase Count	Item Price	Total Purchase Value
Item ID	Item Name			
34	Retribution Axe	9	\$4.14	\$37.26
115	Spectral Diamond Doomblade	7	\$4.25	\$29.75
32	Orenmir	6	\$4.95	\$29.70
103	Singed Scalpel	6	\$4.87	\$29.22
107	Splitter, Foe Of Subtlety	8	\$3.61	\$28.88