**John Hopkins Panda-Challenge**

**6/14/2021**

**Note**[**¶**](http://localhost:8888/notebooks/Instructions/HeroesOfPymoli/HeroesOfPymoli_starter.ipynb#Note)

* Instructions have been included for each segment. You do not have to follow them exactly, but they are included to help you think through the steps.

In [1]:

*# Dependencies and Setup*

**import** pandas **as** pd

​

*# Load data file*

datafile **=** "Resources/purchase\_data.csv"

​

*# Read Data File and Store into Pandas data frame*

purchases\_df **=** pd.read\_csv(datafile)

purchases\_df.head(10)

Out[1]:

|  | **Purchase ID** | **SN** | **Age** | **Gender** | **Item ID** | **Item Name** | **Price** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 0 | Lisim78 | 20 | Male | 108 | Extraction, Quickblade Of Trembling Hands | 3.53 |
| **1** | 1 | Lisovynya38 | 40 | Male | 143 | Frenzied Scimitar | 1.56 |
| **2** | 2 | Ithergue48 | 24 | Male | 92 | Final Critic | 4.88 |
| **3** | 3 | Chamassasya86 | 24 | Male | 100 | Blindscythe | 3.27 |
| **4** | 4 | Iskosia90 | 23 | Male | 131 | Fury | 1.44 |
| **5** | 5 | Yalae81 | 22 | Male | 81 | Dreamkiss | 3.61 |
| **6** | 6 | Itheria73 | 36 | Male | 169 | Interrogator, Blood Blade of the Queen | 2.18 |
| **7** | 7 | Iskjaskst81 | 20 | Male | 162 | Abyssal Shard | 2.67 |
| **8** | 8 | Undjask33 | 22 | Male | 21 | Souleater | 1.10 |
| **9** | 9 | Chanosian48 | 35 | Other / Non-Disclosed | 136 | Ghastly Adamantite Protector | 3.58 |

**Player Count**

* Display the total number of players

In [2]:

total\_players **=** len(purchases\_df.SN.unique())

print(f'Total # of players that have made purchases: {total\_players}')

Total # of players that have made purchases: 576

**Purchasing Analysis (Total)**

* Run basic calculations to obtain number of unique items, average price, etc.
* Create a summary data frame to hold the results
* Optional: give the displayed data cleaner formatting
* Display the summary data frame

In [3]:

unique\_items\_df **=** purchases\_df.loc[purchases\_df["Item ID"].unique()].copy()

num\_products **=** unique\_items\_df['Item ID'].count()

avg\_price **=** purchases\_df['Price'].mean()

num\_purchases **=** purchases\_df['Purchase ID'].count()

total\_revenue **=** purchases\_df['Price'].sum()

​

summary\_df **=** pd.DataFrame({"Number of Unique Items": [num\_products],

"Average Price": [avg\_price],

"Number of Purchases": [num\_purchases],

"Total Revenue": [total\_revenue]

})

​

summary\_df["Average Price"] **=** summary\_df["Average Price"].map("${:.2f}".format)

summary\_df["Total Revenue"] **=** summary\_df["Total Revenue"].map("${:.2f}".format)

​

summary\_df

Out[3]:

|  | **Number of Unique Items** | **Average Price** | **Number of Purchases** | **Total Revenue** |
| --- | --- | --- | --- | --- |
| **0** | 179 | $3.05 | 780 | $2379.77 |

**Gender Demographics**

* Percentage and Count of Male Players
* Percentage and Count of Female Players
* Percentage and Count of Other / Non-Disclosed

In [4]:

*# Groupby gender and count unique players*

groupedby **=** purchases\_df.groupby(['Gender'])

gender\_counts **=** groupedby['SN'].nunique()

sum\_gender\_counts **=** gender\_counts.sum()

​

male\_count **=** gender\_counts['Male']

male\_percent **=** (male\_count**/**sum\_gender\_counts) **\*** 100

​

female\_count **=** gender\_counts['Female']

female\_percent **=** (female\_count**/**sum\_gender\_counts) **\*** 100

​

other\_count **=** gender\_counts['Other / Non-Disclosed']

other\_percent **=** (other\_count**/**sum\_gender\_counts) **\*** 100

​

gender\_df **=** pd.DataFrame(gender\_counts)

gender\_df['Percents'] **=** [male\_percent, female\_percent, other\_percent]

gender\_df["Percents"] **=** gender\_df["Percents"].map("{:.2f}%".format)

x **=** gender\_df["Percents"]

gender\_df.columns **=** ["Total Count", "Percentage"]

​

gender\_df

Out[4]:

|  | **Total Count** | **Percentage** |
| --- | --- | --- |
| **Gender** |  |  |
| **Female** | 81 | 84.03% |
| **Male** | 484 | 14.06% |
| **Other / Non-Disclosed** | 11 | 1.91% |

**Purchasing Analysis (Gender)**

* Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase total per person etc. by gender
* Create a summary data frame to hold the results
* Optional: give the displayed data cleaner formatting
* Display the summary data frame

In [5]:

total\_item\_count\_by\_gender **=** purchases\_df.groupby('Gender')['Item ID'].count()

total\_spent\_by\_gender **=** purchases\_df.groupby('Gender')['Price'].sum()

max\_cost\_by\_gender **=** purchases\_df.groupby('Gender')['Price'].max()

min\_cost\_by\_gender **=** purchases\_df.groupby('Gender')['Price'].min()

avg\_cost\_by\_gender **=** purchases\_df.groupby('Gender')['Price'].mean()

avg\_spend\_by\_gender **=** purchases\_df.groupby(["Gender"]).mean()["Price"].rename("Average Purchase Price by Gender")

​

norm\_spend\_by\_gender **=** x

​

avg\_spent\_by\_female **=** total\_spent\_by\_gender.Female **/** gender\_counts.Female

avg\_spent\_by\_male **=** total\_spent\_by\_gender.Male **/** gender\_counts.Male

avg\_spent\_by\_other **=** total\_spent\_by\_gender["Other / Non-Disclosed"] **/** gender\_counts["Other / Non-Disclosed"]

​

gender\_table\_df **=** pd.DataFrame(

{

"Purchase Count": total\_item\_count\_by\_gender,

"Average Purchase Price": avg\_spend\_by\_gender,

"Total Purchase Value": total\_spent\_by\_gender,

"Avg Total Purchase Per Person": norm\_spend\_by\_gender

}

)

​

gender\_table\_df

Out[5]:

|  | **Purchase Count** | **Average Purchase Price** | **Total Purchase Value** | **Avg Total Purchase Per Person** |
| --- | --- | --- | --- | --- |
| **Gender** |  |  |  |  |
| **Female** | 113 | 3.203009 | 361.94 | 84.03% |
| **Male** | 652 | 3.017853 | 1967.64 | 14.06% |
| **Other / Non-Disclosed** | 15 | 3.346000 | 50.19 | 1.91% |

**Age Demographics**

* Establish bins for ages
* Categorize the existing players using the age bins. Hint: use pd.cut()
* Calculate the numbers and percentages by age group
* Create a summary data frame to hold the results
* Optional: round the percentage column to two decimal points
* Display Age Demographics Table

In [6]:

*# Establish bins for ages*

age\_bins **=** [0, 9.90, 14.90, 19.90, 24.90, 29.90, 34.90, 39.90, 99999]

group\_names **=** ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]

​

purchases\_df["Age Ranges"] **=** pd.cut(purchases\_df["Age"], age\_bins, labels**=**group\_names)

​

groupedby\_ages **=** purchases\_df.groupby('Age Ranges')

age\_counts **=** groupedby\_ages['SN'].nunique()

total\_unique\_players **=** age\_counts.sum()

​

age\_list **=** [{"Unique Players": age\_counts["<10"], "Percentage": (age\_counts["<10"] **\*** 100) **/** total\_unique\_players},

{"Unique Players": age\_counts["10-14"], "Percentage": (age\_counts["10-14"] **\*** 100) **/** total\_unique\_players},

{"Unique Players": age\_counts["15-19"], "Percentage": (age\_counts["15-19"] **\*** 100) **/** total\_unique\_players},

{"Unique Players": age\_counts["20-24"], "Percentage": (age\_counts["20-24"] **\*** 100) **/** total\_unique\_players},

{"Unique Players": age\_counts["25-29"], "Percentage": (age\_counts["25-29"] **\*** 100) **/** total\_unique\_players},

{"Unique Players": age\_counts["30-34"], "Percentage": (age\_counts["30-34"] **\*** 100) **/** total\_unique\_players},

{"Unique Players": age\_counts["35-39"], "Percentage": (age\_counts["35-39"] **\*** 100) **/** total\_unique\_players},

{"Unique Players": age\_counts["40+"], "Percentage": (age\_counts["40+"] **\*** 100) **/** total\_unique\_players}

]

​

summary\_ages\_df **=** pd.DataFrame(age\_list)

summary\_ages\_df.rename(index**=**{0:'<10',

1:'10-14',

2:'15-19',

3:'20-24',

4:'25-29',

5:'30-34',

6:'35-39',

7:'40+'},

inplace **=** **True**

)

​

summary\_ages\_df["Percentage"] **=** summary\_ages\_df["Percentage"].map("{:.2f}%".format)

summary\_ages\_df **=** summary\_ages\_df[["Unique Players", "Percentage"]]

​

summary\_ages\_df **=** summary\_ages\_df.rename(columns **=** {'Unique Players': 'Total Count', 'Percentage': 'Percentage of Players'}, inplace **=** **False**)

summary\_ages\_df

Out[6]:

|  | **Total Count** | **Percentage of Players** |
| --- | --- | --- |
| **<10** | 17 | 2.95% |
| **10-14** | 22 | 3.82% |
| **15-19** | 107 | 18.58% |
| **20-24** | 258 | 44.79% |
| **25-29** | 77 | 13.37% |
| **30-34** | 52 | 9.03% |
| **35-39** | 31 | 5.38% |
| **40+** | 12 | 2.08% |

**Purchasing Analysis (Age)**

* Bin the purchase\_data data frame by age
* Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase total per person etc. in the table below
* Create a summary data frame to hold the results
* Optional: give the displayed data cleaner formatting
* Display the summary data frame

In [7]:

gender\_table\_df

Out[7]:

|  | **Purchase Count** | **Average Purchase Price** | **Total Purchase Value** | **Avg Total Purchase Per Person** |
| --- | --- | --- | --- | --- |
| **Gender** |  |  |  |  |
| **Female** | 113 | 3.203009 | 361.94 | 84.03% |
| **Male** | 652 | 3.017853 | 1967.64 | 14.06% |
| **Other / Non-Disclosed** | 15 | 3.346000 | 50.19 | 1.91% |

In [8]:

*# Reuse Previous DataFrame*

groupedby\_age\_range **=** purchases\_df.groupby("Age Ranges")

total\_item\_count\_by\_age **=** groupedby\_age\_range['Item ID'].count()

total\_spent\_by\_age **=** groupedby\_age\_range['Price'].sum()

avg\_purchase\_price **=** groupedby\_age\_range['Price'].mean().round(2)

*# .groupby("Age Ranges")*

​

age\_purchase\_list **=** [

{"Purchase Count": total\_item\_count\_by\_age["<10"], "Average Purchase Price": avg\_purchase\_price["<10"], "Total Purchases": total\_spent\_by\_age["<10"], "Avg Spent per Player": total\_spent\_by\_age["<10"] **/** age\_counts["<10"]},

{"Purchase Count": total\_item\_count\_by\_age["10-14"], "Average Purchase Price": avg\_purchase\_price["10-14"], "Total Purchases": total\_spent\_by\_age["10-14"], "Avg Spent per Player": total\_spent\_by\_age["10-14"] **/** age\_counts["10-14"]},

{"Purchase Count": total\_item\_count\_by\_age["15-19"], "Average Purchase Price": avg\_purchase\_price["15-19"], "Total Purchases": total\_spent\_by\_age["15-19"], "Avg Spent per Player": total\_spent\_by\_age["15-19"] **/** age\_counts["15-19"]},

{"Purchase Count": total\_item\_count\_by\_age["20-24"], "Average Purchase Price": avg\_purchase\_price["20-24"], "Total Purchases": total\_spent\_by\_age["20-24"], "Avg Spent per Player": total\_spent\_by\_age["20-24"] **/** age\_counts["20-24"]},

{"Purchase Count": total\_item\_count\_by\_age["25-29"], "Average Purchase Price": avg\_purchase\_price["25-29"], "Total Purchases": total\_spent\_by\_age["25-29"], "Avg Spent per Player": total\_spent\_by\_age["25-29"] **/** age\_counts["25-29"]},

{"Purchase Count": total\_item\_count\_by\_age["30-34"], "Average Purchase Price": avg\_purchase\_price["30-34"], "Total Purchases": total\_spent\_by\_age["30-34"], "Avg Spent per Player": total\_spent\_by\_age["30-34"] **/** age\_counts["30-34"]},

{"Purchase Count": total\_item\_count\_by\_age["35-39"], "Average Purchase Price": avg\_purchase\_price["35-39"], "Total Purchases": total\_spent\_by\_age["35-39"], "Avg Spent per Player": total\_spent\_by\_age["35-39"] **/** age\_counts["35-39"]},

{"Purchase Count": total\_item\_count\_by\_age["40+"], "Average Purchase Price": avg\_purchase\_price["40+"], "Total Purchases": total\_spent\_by\_age["40+"], "Avg Spent per Player": total\_spent\_by\_age["40+"] **/** age\_counts["40+"]}]

​

summary\_purchases\_by\_age\_df **=** pd.DataFrame(age\_purchase\_list)

​

summary\_purchases\_by\_age\_df **=** summary\_purchases\_by\_age\_df.rename(columns **=** {'Total Purchases': 'Total Purchase Value', 'Avg Spent per Player': 'Avg Total Purchase per Person'}, inplace **=** **False**)

​

summary\_purchases\_by\_age\_df

Out[8]:

|  | **Purchase Count** | **Average Purchase Price** | **Total Purchase Value** | **Avg Total Purchase per Person** |
| --- | --- | --- | --- | --- |
| **0** | 23 | 3.35 | 77.13 | 4.537059 |
| **1** | 28 | 2.96 | 82.78 | 3.762727 |
| **2** | 136 | 3.04 | 412.89 | 3.858785 |
| **3** | 365 | 3.05 | 1114.06 | 4.318062 |
| **4** | 101 | 2.90 | 293.00 | 3.805195 |
| **5** | 73 | 2.93 | 214.00 | 4.115385 |
| **6** | 41 | 3.60 | 147.67 | 4.763548 |
| **7** | 13 | 2.94 | 38.24 | 3.186667 |

In [9]:

*# Formatting*

summary\_purchases\_by\_age\_df["Avg Total Purchase per Person"] **=** summary\_purchases\_by\_age\_df["Avg Total Purchase per Person"].map("${:.2f}".format)

​

summary\_purchases\_by\_age\_df

Out[9]:

|  | **Purchase Count** | **Average Purchase Price** | **Total Purchase Value** | **Avg Total Purchase per Person** |
| --- | --- | --- | --- | --- |
| **0** | 23 | 3.35 | 77.13 | $4.54 |
| **1** | 28 | 2.96 | 82.78 | $3.76 |
| **2** | 136 | 3.04 | 412.89 | $3.86 |
| **3** | 365 | 3.05 | 1114.06 | $4.32 |
| **4** | 101 | 2.90 | 293.00 | $3.81 |
| **5** | 73 | 2.93 | 214.00 | $4.12 |
| **6** | 41 | 3.60 | 147.67 | $4.76 |
| **7** | 13 | 2.94 | 38.24 | $3.19 |

In [10]:

*# Formatting*

summary\_purchases\_by\_age\_df["Average Purchase Price"] **=** summary\_purchases\_by\_age\_df["Average Purchase Price"].map("${:.2f}".format)

​

summary\_purchases\_by\_age\_df

Out[10]:

|  | **Purchase Count** | **Average Purchase Price** | **Total Purchase Value** | **Avg Total Purchase per Person** |
| --- | --- | --- | --- | --- |
| **0** | 23 | $3.35 | 77.13 | $4.54 |
| **1** | 28 | $2.96 | 82.78 | $3.76 |
| **2** | 136 | $3.04 | 412.89 | $3.86 |
| **3** | 365 | $3.05 | 1114.06 | $4.32 |
| **4** | 101 | $2.90 | 293.00 | $3.81 |
| **5** | 73 | $2.93 | 214.00 | $4.12 |
| **6** | 41 | $3.60 | 147.67 | $4.76 |
| **7** | 13 | $2.94 | 38.24 | $3.19 |

In [11]:

*# Formatting*

summary\_purchases\_by\_age\_df["Total Purchase Value"] **=** summary\_purchases\_by\_age\_df["Total Purchase Value"].map("${:.2f}".format)

​

summary\_purchases\_by\_age\_df

Out[11]:

|  | **Purchase Count** | **Average Purchase Price** | **Total Purchase Value** | **Avg Total Purchase per Person** |
| --- | --- | --- | --- | --- |
| **0** | 23 | $3.35 | $77.13 | $4.54 |
| **1** | 28 | $2.96 | $82.78 | $3.76 |
| **2** | 136 | $3.04 | $412.89 | $3.86 |
| **3** | 365 | $3.05 | $1114.06 | $4.32 |
| **4** | 101 | $2.90 | $293.00 | $3.81 |
| **5** | 73 | $2.93 | $214.00 | $4.12 |
| **6** | 41 | $3.60 | $147.67 | $4.76 |
| **7** | 13 | $2.94 | $38.24 | $3.19 |

In [12]:

*# Rename indexes*

summary\_purchases\_by\_age\_df.rename(index**=**{0:'<10', 1:'10-14', 2:'15-19', 3:'20-24', 4:'25-29', 5:'30-34',6:'35-39', 7:'40+'}, inplace**=True**)

​

summary\_purchases\_by\_age\_df

Out[12]:

|  | **Purchase Count** | **Average Purchase Price** | **Total Purchase Value** | **Avg Total Purchase per Person** |
| --- | --- | --- | --- | --- |
| **<10** | 23 | $3.35 | $77.13 | $4.54 |
| **10-14** | 28 | $2.96 | $82.78 | $3.76 |
| **15-19** | 136 | $3.04 | $412.89 | $3.86 |
| **20-24** | 365 | $3.05 | $1114.06 | $4.32 |
| **25-29** | 101 | $2.90 | $293.00 | $3.81 |
| **30-34** | 73 | $2.93 | $214.00 | $4.12 |
| **35-39** | 41 | $3.60 | $147.67 | $4.76 |
| **40+** | 13 | $2.94 | $38.24 | $3.19 |

**Top Spenders**

* Run basic calculations to obtain the results in the table below
* Create a summary data frame to hold the results
* Sort the total purchase value column in descending order
* Optional: give the displayed data cleaner formatting
* Display a preview of the summary data frame

In [13]:

groupedby\_player **=** purchases\_df.groupby(['SN'])

total\_spent\_player **=** groupedby\_player['Price'].sum()

items\_per\_player **=** groupedby\_player['Price'].count()

​

player\_list\_df **=** pd.DataFrame(total\_spent\_player)

player\_list\_df['Item Count'] **=** items\_per\_player

player\_list\_df['Avg Spent'] **=** total\_spent\_player **/** items\_per\_player

player\_list\_df.columns **=** ["Total Purchases", "Item Count", "Avg Spent"]

player\_list\_df

*# Formatting*

player\_list\_df["Avg Spent"] **=** player\_list\_df["Avg Spent"].map("${:.2f}".format)

player\_list\_df.sort\_values(by**=**'Total Purchases', ascending**=False**, inplace**=True**)

player\_list\_df["Total Purchases"] **=** player\_list\_df["Total Purchases"].map("${:.2f}".format)

​

player\_list\_df.head(5)

Out[13]:

|  | **Total Purchases** | **Item Count** | **Avg Spent** |
| --- | --- | --- | --- |
| **SN** |  |  |  |
| **Lisosia93** | $18.96 | 5 | $3.79 |
| **Idastidru52** | $15.45 | 4 | $3.86 |
| **Chamjask73** | $13.83 | 3 | $4.61 |
| **Iral74** | $13.62 | 4 | $3.40 |
| **Iskadarya95** | $13.10 | 3 | $4.37 |

**Most Popular Items**

* Retrieve the Item ID, Item Name, and Item Price columns
* Group by Item ID and Item Name. Perform calculations to obtain purchase count, average item price, and total purchase value
* Create a summary data frame to hold the results
* Sort the purchase count column in descending order
* Optional: give the displayed data cleaner formatting
* Display a preview of the summary data frame

In [14]:

popular\_item\_df **=** purchases\_df.loc[:, 'Item ID':'Price'].copy()

groupedby\_popular **=** popular\_item\_df.groupby(['Item ID', 'Item Name'])

​

total\_purchases **=** groupedby\_popular['Price'].sum()

num\_items\_purchased **=** groupedby\_popular['Item ID'].count()

​

*# Create dataframe from groupby values*

popular\_df **=** pd.DataFrame(num\_items\_purchased)

popular\_df['Total Purchase Value'] **=** total\_purchases

popular\_df.columns **=** ["Purchase Count", "Total Purchase Value"]

popular\_df **=** popular\_df.sort\_values(by**=**'Item ID')

​

*# Create prices dataframe*

prices\_df **=** purchases\_df[['Item ID', 'Item Name', 'Price']]

prices\_nodup\_df **=** prices\_df.drop\_duplicates(['Item ID'], keep **=** 'last')

prices\_post\_df **=** prices\_nodup\_df.sort\_values(by**=**['Item ID'])

prices\_post\_df.set\_index(['Item ID', 'Item Name'], inplace**=True**)

*# Join the 2 dataframes*

popular\_df **=** popular\_df.join(prices\_post\_df)

​

*# Sort by Purchase Count and format*

popular\_df **=** popular\_df.sort\_values(by**=**'Purchase Count', ascending**=False**)

popular\_df["Price"] **=** popular\_df["Price"].map("${:.2f}".format)

popular\_df["Total Purchase Value"] **=** popular\_df["Total Purchase Value"].map("${:.2f}".format)

​

popular\_df.head()

Out[14]:

|  |  | **Purchase Count** | **Total Purchase Value** | **Price** |
| --- | --- | --- | --- | --- |
| **Item ID** | **Item Name** |  |  |  |
| **92** | **Final Critic** | 13 | $59.99 | $4.19 |
| **178** | **Oathbreaker, Last Hope of the Breaking Storm** | 12 | $50.76 | $4.23 |
| **145** | **Fiery Glass Crusader** | 9 | $41.22 | $4.58 |
| **132** | **Persuasion** | 9 | $28.99 | $3.33 |
| **108** | **Extraction, Quickblade Of Trembling Hands** | 9 | $31.77 | $3.53 |

**Most Profitable Items**

* Sort the above table by total purchase value in descending order
* Optional: give the displayed data cleaner formatting
* Display a preview of the data frame

In [21]:

*# Working copy for most profitable items list*

profit\_df **=** popular\_df

profit\_df.head()

Out[21]:

|  |  | **Purchase Count** | **Total Purchase Value** | **Price** |
| --- | --- | --- | --- | --- |
| **Item ID** | **Item Name** |  |  |  |
| **92** | **Final Critic** | 13 | $59.99 | $4.19 |
| **178** | **Oathbreaker, Last Hope of the Breaking Storm** | 12 | $50.76 | $4.23 |
| **145** | **Fiery Glass Crusader** | 9 | $41.22 | $4.58 |
| **132** | **Persuasion** | 9 | $28.99 | $3.33 |
| **108** | **Extraction, Quickblade Of Trembling Hands** | 9 | $31.77 | $3.53 |

In [33]:

*# Sort by Total Purchase Value and format*

profit\_df **=** profit\_df.sort\_values(by**=**'Total Purchase Value', ascending**=False**)

profit\_df["Price"] **=** profit\_df["Price"]

​

profit\_df.head()

Out[33]:

|  |  | **Purchase Count** | **Total Purchase Value** | **Price** |
| --- | --- | --- | --- | --- |
| **Item ID** | **Item Name** |  |  |  |
| **63** | **Stormfury Mace** | 2 | $9.98 | $4.99 |
| **29** | **Chaos, Ender of the End** | 5 | $9.90 | $1.98 |
| **173** | **Stormfury Longsword** | 2 | $9.86 | $4.93 |
| **38** | **The Void, Vengeance of Dark Magic** | 4 | $9.48 | $2.37 |
| **143** | **Frenzied Scimitar** | 6 | $9.36 | $1.56 |