

## PD assignment Data Analysis

- Of the 1163 active players, the vast majority are male (84%). There also exists, a smaller, but notable proportion of female players (14%).
  - Our peak age demographic falls between 20-24 (44.8%) with secondary groups falling between 15-19 (18.60%) and 25-29 (13.4%).
- 

## 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 pandas, numpy
import pandas as pd
import numpy as np

# Read Purchasing File and store into Pandas data frame

data=pd.read_csv('/Users/roatny/Desktop/ICT Master Application/3.Pr
data
```

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
...	...	...	...	...	...	...	...
775	775	Aethedru70	21	Female	60	Wolf	3.54
776	776	Iral74	21	Male	164	Exiled Doomblade	1.63
777	777	Yathecal72	20	Male	67	Celeste, Incarnation of the Corrupted	3.46
778	778	Sisur91	7	Male	101	Final Critic	4.19
779	779	Ennrian78	24	Male	50	Dawn	4.60

780 rows × 7 columns

```
In [2]: #check for missing/na
```

```
data.isna().sum()
```

```
Out[2]: Purchase ID    0
        SN            0
        Age           0
        Gender        0
        Item ID       0
        Item Name     0
        Price         0
        dtype: int64
```

## Player Count

- Display the total number of players

```
In [14]: #display/set to variable
total_number_players=data['SN'].nunique()

#set in DF
pd.DataFrame(
    {'total_number_players':[total_number_players]}
)
```

Out [14]:

	total_number_players
0	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 [17]: #Count unique items
uni_items=data['Item ID'].nunique()
#Get the average price
aver_price=data['Price'].mean()

#Count the number of purchases
num_pur=data['Purchase ID'].count()

#Calculate the total revenue from all sales
total_sales=data['Price'].sum()

#create dataframe to hold above results

dataframe=pd.DataFrame({'uni_items':[uni_items],
                        'aver_price':[aver_price],
                        'num_pur':[num_pur],
                        'total_sales':[total_sales]})

dataframe[['uni_items','aver_price','num_pur','total_sales']]
```

Out[17]:

	uni_items	aver_price	num_pur	total_sales
0	183	3.050987	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 [33]: #find unique SN within df to make sure the same player is not recor
uniq_purchase=data.groupby(['Gender'])

#count each gender
gender_c=uniq_purchase['SN'].nunique()

#get percentage
gender_cPercentage=gender_c/total_number_players*100

#present data in dataframe
gender_dataframe=pd.DataFrame({"Total Count": gender_c,
                              "Percentage Gender":gender_cPercentag

gender_dataframe[["Total Count","Percentage Gender"]]
```

Out [33]:

	Total Count	Percentage Gender
Gender		
Female	81	14.062500
Male	484	84.027778
Other / Non-Disclosed	11	1.909722

## 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 [39]: #purchases by each gender
Purchase_count=uniq_purchase['Purchase ID'].count()

#average purchase price
Avg_Purchase_Price=uniq_purchase['Price'].mean()

#total revenue
Total_Revenue=uniq_purchase['Price'].sum()

#Create DataFrame

Purchasing=pd.DataFrame({'Total Count':gender_c,
                          'Purchase Count':Purchase_count,
                          'Avg Purchase Price':Avg_Purchase_Price,
                          'Total Revenue':Total_Revenue})
Purchasing[['Total Count', 'Purchase Count', 'Avg Purchase Price', 'To
```

Out [39]:

	Total Count	Purchase Count	Avg Purchase Price	Total Revenue
Gender				
Female	81	113	3.203009	361.94
Male	484	652	3.017853	1967.64
Other / Non-Disclosed	11	15	3.346000	50.19

In [27]:

Out [27]:

	Number of Players	Purchase Count	Avg Purchase Price	Total Revenue
Female	81	113	\$3.20	\$361.94
Male	484	652	\$3.02	\$1,967.64
Other / Non-Disclosed	11	15	\$3.35	\$50.19

## Age Demographics

- ```
In [41]: #establish bins
bi=[0,9,14,19,24,29,34,39,150]

age_labels=['<10', '10-14', '15-19', '20-24', '25-29', '30-34', '35-39', '
```

```
In [44]: #groupby
age_player=data.groupby(['Age Labels'])

#create dataframe to show total count and percentage of total playe
Agecount=age_player['SN'].nunique()
AgePercentage=Agecount/total_number_players*100

dataframe=pd.DataFrame({'Total Count':Agecount,
                        'Percentage of Players':AgePercentage})

dataframe[["Total Count", 'Percentage of Players']]
```

|            | Total Count | Percentage of Players |
|------------|-------------|-----------------------|
| Age Labels |             |                       |
| <10        | 17          | 2.951389              |
| 10-14      | 22          | 3.819444              |
| 15-19      | 107         | 18.576389             |
| 20-24      | 258         | 44.791667             |
| 25-29      | 77          | 13.368056             |
| 30-34      | 52          | 9.027778              |
| 35-39      | 31          | 5.381944              |
| 40+        | 12          | 2.083333              |

```
In [30]: #groupby  
  
#create dataframe to show total count and percentage of total playe
```

```
Out[30]:
```

|           | Total Count | Percentage of Players |
|-----------|-------------|-----------------------|
| age_group |             |                       |
| <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 [49]: #isolate columns needed for chunk in two frames
purchase_count=age_player['Purchase ID'].count()
Avg_purchase=age_player['Price'].mean()

#totals purchase by age bin
totalpurchasebyAge=age_player['Price'].sum()

#Dollar values of purchase by age

#display summary df
Purchase_Age=pd.DataFrame({'Purchase Count':purchase_count,
                           'Average Purchase Price':Avg_purchase,
                           'Total Purchase Value':totalpurchasebyAge}
Purchase_Age[['Purchase Count', 'Average Purchase Price', 'Total Purc

```

Out [49]:

|            | Purchase Count | Average Purchase Price | Total Purchase Value |
|------------|----------------|------------------------|----------------------|
| Age Labels |                |                        |                      |
| <10        | 23             | 3.353478               | 77.13                |
| 10-14      | 28             | 2.956429               | 82.78                |
| 15-19      | 136            | 3.035956               | 412.89               |
| 20-24      | 365            | 3.052219               | 1114.06              |
| 25-29      | 101            | 2.900990               | 293.00               |
| 30-34      | 73             | 2.931507               | 214.00               |
| 35-39      | 41             | 3.601707               | 147.67               |
| 40+        | 13             | 2.941538               | 38.24                |

In [31]: *#isolate columns needed for chunk in two frames*

*#totals purchase by age bin*

*#Dollar values of purchase by age*

*#display summary df*

Out [31]:

|           | Purchase Count | Average Purchase Price | Total Purchase Value |
|-----------|----------------|------------------------|----------------------|
| age_group |                |                        |                      |
| <10       | 23             | \$3.35                 | \$77.13              |
| 10-14     | 28             | \$2.96                 | \$82.78              |
| 15-19     | 136            | \$3.04                 | \$412.89             |
| 20-24     | 365            | \$3.05                 | \$1114.06            |
| 25-29     | 101            | \$2.90                 | \$293.00             |
| 30-34     | 73             | \$2.93                 | \$214.00             |
| 35-39     | 41             | \$3.60                 | \$147.67             |
| 40+       | 13             | \$2.94                 | \$38.24              |

## 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 [52]: #Count purchases made by unique SN
purchase_sn=data.groupby(['SN'])
count_purchase_sn=purchase_sn['Purchase ID'].count()

#Calculate the average purchase by unique SN
Avg_purchase_sn=purchase_sn['Price'].mean()

#Calculate the total revenue from each unique SN
total_revenue_sn=purchase_sn['Price'].sum()

#display summary df
Top_Spender=pd.DataFrame({'Purchase Count':count_purchase_sn,
                          'Average Purchase Price':Avg_purchase_sn,
                          'Total Purchase Value':total_revenue_sn})

#create df, sort and format
Top_Spender=Top_Spender.sort_values(['Total Purchase Value'],ascend)
Top_Spender["Average Purchase Price"]=Top_Spender['Average Purchase Price']
Top_Spender["Total Purchase Value"]=Top_Spender['Total Purchase Value']

Top_Spender[['Purchase Count','Average Purchase Price','Total Purchase Value']]
```

Out [52]:

|             | Purchase Count | Average Purchase Price | Total Purchase Value |
|-------------|----------------|------------------------|----------------------|
| SN          |                |                        |                      |
| Lisosia93   | 5              | \$3.79                 | \$18.96              |
| Idastidru52 | 4              | \$3.86                 | \$15.45              |
| Chamjask73  | 3              | \$4.61                 | \$13.83              |
| Iral74      | 4              | \$3.41                 | \$13.62              |
| Iskadarya95 | 3              | \$4.37                 | \$13.10              |

## 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, 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 [74]: *#isolate columned needed*

```

item_isolate=data[['Item ID','Item Name','Price']]

#count transactions for each item
item_count=item_isolate.groupby(['Item ID','Item Name'])
purchase_item_count=item_count['Item ID'].count()

#Price of each item
Price_item=item_count['Price'].mean()

#Calculate total revenue from each item
Total_revenue_item=item_count['Price'].sum()

#display summary df
item_data=pd.DataFrame({'Purchase Count':purchase_item_count,
                        'Item Price':Price_item,
                        'Total Purchase Value':Total_revenue_item})

#create new df, sort and format
item_data=item_data.sort_values(['Purchase Count'],ascending=False)

item_data['Item Price']=item_data['Item Price'].map('${:,.2f}'.format)
item_data['Total Purchase Value']=item_data['Total Purchase Value'].h

#print the DF
item_data[['Purchase Count','Item Price','Total Purchase Value']].h

```

Out [74]:

|            |                                              | Purchase<br>Count | Item<br>Price | Total Purchase<br>Value |
|------------|----------------------------------------------|-------------------|---------------|-------------------------|
| Item<br>ID | Item Name                                    |                   |               |                         |
| 178        | Oathbreaker, Last Hope of the Breaking Storm | 12                | \$4.23        | \$50.76                 |
| 145        | Fiery Glass Crusader                         | 9                 | \$4.58        | \$41.22                 |
| 108        | Extraction, Quickblade Of Trembling Hands    | 9                 | \$3.53        | \$31.77                 |
| 82         | Nirvana                                      | 9                 | \$4.90        | \$44.10                 |
| 19         | Pursuit, Cudgel of Necromancy                | 8                 | \$1.02        | \$8.16                  |

## 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 [77]: #re-sort above table by total purchase
item_data['Total Purchase Value']=item_data['Total Purchase Value']
item_data=item_data.sort_values(['Total Purchase Value'],ascending=
#print the DF
```

```
In [78]: item_data['Total Purchase Value']=item_data['Total Purchase Value']
```

```
In [79]: item_data[['Purchase Count','Item Price','Total Purchase Value']].h
```

Out[79]:

|            |                                              | Purchase<br>Count | Item<br>Price | Total Purchase<br>Value |
|------------|----------------------------------------------|-------------------|---------------|-------------------------|
| Item<br>ID | Item Name                                    |                   |               |                         |
| 178        | Oathbreaker, Last Hope of the Breaking Storm | 12                | \$4.23        | \$50.76                 |
| 82         | Nirvana                                      | 9                 | \$4.90        | \$44.10                 |
| 145        | Fiery Glass Crusader                         | 9                 | \$4.58        | \$41.22                 |
| 92         | Final Critic                                 | 8                 | \$4.88        | \$39.04                 |
| 103        | Singed Scalpel                               | 8                 | \$4.35        | \$34.80                 |

In [ ]: