

House of Pymoli

Ellen White

```
In [1]: 1 import pandas as pd
```

```
In [4]: 1 # Reading purchasing data
2 file = "Resources/purchase_data.csv"
3 purchase_data = pd.read_csv(file)
```

```
In [44]: 1 purchase_data
```

Out[44]:

	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	92	Final Critic	4.19

```
In [6]: 1 purchase_data.head()
```

Out[6]:

	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

Player Count

Total Number of Players

```
In [15]: 1 # Count total number of unique usernames
          2 totalPlayers = purchase_data.nunique()[1]
          3 print(f'The Total Number of Players is: {totalPlayers}')
```

The Total Number of Players is: 576

Purchasing Analysis (Total)

Number of Unique Items

```
In [17]: 1 uniqueItems = purchase_data.nunique()
          2 print(uniqueItems)
```

```
Purchase ID    780
SN             576
Age            39
Gender         3
Item ID        179
Item Name      179
Price          145
dtype: int64
```

Average Purchase Price

```
In [36]: 1 avgP = round(purchase_data["Price"].mean(), 2)
          2 print(f'The Average Purchase Price is: ${avgP}')
```

The Average Purchase Price is: \$3.05

Total Number of Purchases

```
In [37]: 1 totalP = purchase_data.count()["Purchase ID"]
          2 print(f'The Total Amount of Purchases is: {totalP}')
```

The Total Amount of Purchases is: 780

Total Revenue

```
In [38]: 1 totalR = purchase_data.sum()["Price"]
          2 print(f'The Total Revenue is: ${totalR}')
```

The Total Revenue is: \$2379.77

```
In [48]: 1 data = {'Average Purchase Price': ['$' + str(avgP)], 'Total Number of P
2 Total = pd.DataFrame(data)
3 Total
```

```
Out[48]:
```

	Average Purchase Price	Total Number of Purchases	Total Revenue
0	\$3.05	780	\$2379.77

Gender Demographics

Percentage and Count of Male Players

```
In [72]: 1 noDuplicatePlayers = purchase_data.drop_duplicates(subset = ['SN'])
2 maleCount = noDuplicatePlayers[noDuplicatePlayers.Gender == 'Male'].sha
3 malePerc = round((maleCount/totalPlayers)*100, 2)
4 print(f' The Total Number of male players is {maleCount} which is {male
```

The Total Number of male players is 484 which is 84.03% of the total

```
In [74]: 1 femaleCount = noDuplicatePlayers[noDuplicatePlayers.Gender == 'Female']
2 femalePerc = round((femaleCount/totalPlayers)*100, 2)
3 print(f' The Total Number of female players is {femaleCount} which is {
```

The Total Number of female players is 81 which is 14.06% of the total

```
In [79]: 1 other = totalPlayers - (maleCount+femaleCount)
2 otherPerc = round(100 - (malePerc+femalePerc),2)
3 print(f' The Total Number of other players is {other} which is {otherPe
```

The Total Number of other players is 11 which is 1.91% of the total

```
In [88]: 1 datal = {'Gender': ['Male', 'Female', 'other/non-disclosed'], 'Total Cou
2 gendersummary = pd.DataFrame(datal)
3 gendersummary
```

```
Out[88]:
```

	Gender	Total Count	Percent of Players
0	Male	484	84.03%
1	Female	81	14.06%
2	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

```
In [141]: 1 MC = purchase_data[purchase_data.Gender == 'Male'].sum()
2 print(MC)
3 malePurchases = MC[6]
4 FC = purchase_data[purchase_data.Gender == 'Female'].sum()
5 femalePurchases = FC[6]
6 otherPurchases = totalR - malePurchases - femalePurchases
7 avgpricemale = round(malePurchases / totalR, 2)
8 avgpricefemale = round(femalePurchases / totalR, 2)
9 avgpriceother = round(otherPurchases / totalR, 2)
10 permale = round(malePurchases / maleCount, 2)
11 perfemale = round(femalePurchases / femaleCount, 2)
12 perother = round(otherPurchases / other, 2)
```

```
Purchase ID                255921
SN          Lisim78Lisovynya38Ithergue48Chamassasya86Iskos...
Age                14942
Gender          MaleMaleMaleMaleMaleMaleMaleMaleMaleMaleMa...
Item ID                60698
Item Name          Extraction, Quickblade Of Trembling HandsFrenz...
Price                1967.64
dtype: object
```

```
In [142]: 1 purchasecount = [malePurchases, femalePurchases, otherPurchases]
2 avgprice = [avgpricemale, avgpricefemale, avgpriceother]
3 perperson = [permale, perfemale, perother]
```

```
In [149]: 1 gendersummary['Purchase Count'] = purchasecount
2 gendersummary['Average Price'] = avgprice
3 gendersummary['Average Purchase Total Per Person'] = perperson
```

```
In [156]: 1 gendersummary.drop(['Percent of Players'], axis = 1)
```

```
Out[156]:
```

	Gender	Total Count	Purchase Count	Average Price	Average Purchase Total Per Person
0	Male	484	1967.64	0.83	4.07
1	Female	81	361.94	0.15	4.47
2	other/non-disclosed	11	50.19	0.02	4.56

Age Demographics

Establish bins for ages

```
In [282]: 1 # I did this is a really strange way I apologize, the next dataframe wa
```

```
In [283]: 1 bins = [0, 9.9, 14.9, 19.9, 24.9, 29.9, 34.9, 39.9, 100]
2 groupnames = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]
3 noDuplicatePlayers['Age Group'] = pd.cut(noDuplicatePlayers['Age'], bins=bins, labels=groupnames)
4 agedemo = noDuplicatePlayers.groupby(['Age Group']).count()
5 new = agedemo.drop(['Purchase ID', 'SN', 'Gender', 'Item ID', 'Item Name', 'Price'])
6 perc = [round(x/totalPlayers*100, 2) for x in agedemo['Age Group']]
7 new['Percentage of Players'] = perc
8 new.rename(inplace=True, columns={'Age Group': 'Count of Age'})
9 new
10
```

/var/folders/ks/35kq94394nv3t6m7t5_66w9c0000gn/T/ipykernel_14922/1336948708.py:3: SettingWithCopyWarning:
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: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
noDuplicatePlayers['Age Group'] = pd.cut(noDuplicatePlayers['Age'], bins=bins, labels=groupnames)
```

Out[283]:

	Count of Age	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

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)

```
In [293]: 1 x = purchase_data.groupby(['Age Group']).count()['Price']
2 y = purchase_data.groupby(['Age Group']).sum()['Price']
3 z = round(purchase_data.groupby(['Age Group']).mean()['Price'], 2)
4 t = round(y/(noDuplicatePlayers.groupby(['Age Group']).count()['Price']
```

```
In [294]: 1 pd.DataFrame({'Purchase Count': x, 'Total Purchase Value': y, 'Average
```

Out[294]:

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

Top Spenders

```
In [306]: 1 purchase_data.groupby(['SN']).count()
```

Out[306]:

	Purchase ID	Age	Gender	Item ID	Item Name	Price	Age Group
SN							
Adairialis76	1	1	1	1	1	1	1
Adastirin33	1	1	1	1	1	1	1
Aeda94	1	1	1	1	1	1	1
Aela59	1	1	1	1	1	1	1
Aelaria33	1	1	1	1	1	1	1
...
Yathecal82	3	3	3	3	3	3	3
Yathedeu43	2	2	2	2	2	2	2
Yoishirralla98	1	1	1	1	1	1	1
Zhisrisu83	2	2	2	2	2	2	2
Zontibe81	3	3	3	3	3	3	3

576 rows × 7 columns

```
In [308]: 1 x = purchase_data.groupby(['SN']).count()['Purchase ID']
          2 y = round(purchase_data.groupby(['SN']).mean()['Price'],2)
          3 z = purchase_data.groupby(['SN']).sum()['Price']
          4 df = pd.DataFrame({'Purchase Count':x, 'Average Purchase Price':y, 'Total Purchase Value':z})
          5 df
```

Out[308]:

	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	3.40	13.62
Iskadarya95	3	4.37	13.10
...
Ililsasya43	1	1.02	1.02
Irilis75	1	1.02	1.02
Aidai61	1	1.01	1.01
Chanirra79	1	1.01	1.01
Alo38	1	1.00	1.00

576 rows × 3 columns

Most Popular Items

```
In [421]: 1 df = purchase_data.groupby(['Item ID', 'Item Name', 'Price']).size().so
2 df['Total Purchase Value'] = df['Price'] * df['Purchase Count']
3 df
```

Out[421]:

	Item ID	Item Name	Price	Purchase Count	Total Purchase Value
0	178	Oathbreaker, Last Hope of the Breaking Storm	4.23	12	50.76
1	108	Extraction, Quickblade Of Trembling Hands	3.53	9	31.77
2	145	Fiery Glass Crusader	4.58	9	41.22
3	82	Nirvana	4.90	9	44.10
4	37	Shadow Strike, Glory of Ending Hope	3.16	8	25.28
...
178	27	Riddle, Tribute of Ended Dreams	3.30	1	3.30
179	30	Stormcaller	3.36	1	3.36
180	126	Exiled Mithril Longsword	2.00	1	2.00
181	51	Endbringer	4.66	1	4.66
182	91	Celeste	4.17	1	4.17

183 rows × 5 columns

Most Profitable Items

```
In [422]: 1 df2 = df.sort_values(['Total Purchase Value'], ascending = False).head(
2 df2
```

Out[422]:

	Item ID	Item Name	Price	Purchase Count	Total Purchase Value
0	178	Oathbreaker, Last Hope of the Breaking Storm	4.23	12	50.76
3	82	Nirvana	4.90	9	44.10
2	145	Fiery Glass Crusader	4.58	9	41.22
7	92	Final Critic	4.88	8	39.04
6	103	Singed Scalpel	4.35	8	34.80

In []:

1

