

```
In [1]: # import modules
import pandas as pd
import os
import warnings

# Only show warning once in the notebook
warnings.filterwarnings(action = 'ignore')

# Adding options to display all of the rows of DataFrame w/o truncation
pd.set_option('display.max_rows',None)
pd.set_option('display.max_columns',None)
```

```
In [2]: # Create DataFrame from csv file
path_to_csv = os.path.join('Resources', 'purchase_data.csv')
purchase_df = pd.read_csv(path_to_csv)
purchase_df.head()
```

Out[2]:

	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

```
In [3]: # Player Count
total_players = purchase_df["SN"].nunique()
pd.DataFrame({"Total Players":[total_players]})
```

Out[3]:

	Total Players
0	576

## Purchasing Analysis (Total)

```
In [4]: total_number_of_unique_items = purchase_df["Item ID"].nunique()
average_purchase_price = purchase_df["Price"].mean()
total_number_of_purchases = purchase_df["Item ID"].count()
total_revenue = purchase_df["Price"].sum()

# Collect analysis in a summary data frame
purchasing_analysis_df = pd.DataFrame({"Number of Unique Items":[total_number_of_unique_items],\
                                     "Average Price":[average_purchase_price],\
                                     "Number of Purchases":[total_number_of_purchases],\
                                     "Total Revenue":total_revenue})

# Change formatting for cleaner display
purchasing_analysis_df.style.format({'Average Price': "${:,.2f}", \
                                     'Total Revenue': '${:,.2f}'})
```

Out[4]:

	Number of Unique Items	Average Price	Number of Purchases	Total Revenue
0	183	\$3.05	780	\$2,379.77

## Gender Demographics

```
In [5]: # Performing value_counts() on Gender column after
# dropping duplicate players from the data set
# Reasoning: A player may have bought items more than once and therefore
# may appear multiple times in transactions.
gender_df = pd.DataFrame(purchase_df.drop_duplicates("SN")["Gender"].value_counts())
gender_df = gender_df.rename(columns = {"Gender":"Total Counts"})
gender_df["Percentage of Players"] = gender_df["Total Counts"]/total_players*100

# Change formatting for cleaner display
gender_df.style.format({"Percentage of Players": "{:,.2f}%"})
```

Out[5]:

	Total Counts	Percentage of Players
Male	484	84.03%
Female	81	14.06%
Other / Non-Disclosed	11	1.91%

## Purchasing Analysis (Gender)

```

In [19]: # Groupby Gender to calculate gender demographics
purchase_count = (purchase_df.groupby(["Gender"])[ "Purchase ID"].count())
average_price = (purchase_df.groupby(["Gender"])[ "Price"].mean())
total_revenue = (purchase_df.groupby(["Gender"])[ "Price"].sum())
avg_price_per_person = (purchase_df.groupby(["Gender", "SN"])[ 'Price' ].sum())

# Summary Table for purchasing analysis by Gender
summary_purchase_by_gender = pd.DataFrame({"Purchase Count": purchase_count,\
                                           "Average Purchase Price": average_price,\
                                           "Total Purchase Value": total_revenue})

# Calculate total price per ID per gender by adding all the prices per ID.
total_purchase_per_person = purchase_df.groupby(['Gender', 'SN'])[ 'Price' ].sum().reset_index()
avg_total_purchase_per_person = total_purchase_per_person.groupby('Gender').mean()

# using reset_index() to bring Gender as common column in both dfs
avg_total_purchase_per_person = avg_total_purchase_per_person.reset_index()

# Merge summary_purchase_by_gender with newly calculated avg_total_purchase_per_person
summary_purchase_by_gender_merged = pd.merge(summary_purchase_by_gender.reset_index(),\
                                              avg_total_purchase_per_person)

summary_purchase_by_gender_merged = summary_purchase_by_gender_merged.set_index('Gender')
summary_purchase_by_gender_merged = summary_purchase_by_gender_merged.rename(columns={'Price': 'Avg Total Purchase per Person'})

#summary_purchase_by_gender
# Change formatting for cleaner display
summary_purchase_by_gender_merged.style.format({'Average Purchase Price': "${:,.2f}",\
                                              'Total Purchase Value': '${:,.2f}', 'Avg Total Purchase per Person': '${:,.2f}'})

```

Out[19]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
Gender				
Female	113	\$3.20	\$361.94	\$4.47
Male	652	\$3.02	\$1,967.64	\$4.07
Other / Non-Disclosed	15	\$3.35	\$50.19	\$4.56

## Age Demographics

```
In [20]: # Drop duplicates for analysis on players
age_df = purchase_df.drop_duplicates(['SN'])

# Define age bins and labels
age_bins = [0,9,14,19,24,29,34,39,50]
age_labels = ['<10', '10-14', '15-19', '20-24', '25-29', '30-34', '35-39', '40+']

age_df['Age Ranges'] = pd.cut(age_df['Age'], bins = age_bins, labels = age_labels)

# Calculate total counts per bin and percentage of players per bin
counts_per_age_bin = age_df.groupby(['Age Ranges'])['Age Ranges'].count()
percentage_per_age_bin = counts_per_age_bin/total_players*100

age_demo_df = pd.DataFrame({'Total Counts': counts_per_age_bin,\
                           'Percentage of Players': percentage_per_age_bin})

# Change formatting for cleaner display
age_demo_df.style.format({"Percentage of Players": "{:,.2f}%"})
```

Out[20]:

	Total Counts	Percentage of Players
Age Ranges		
<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 [21]: # Not Dropping duplicates for analysis on item purchases

# Define age bins and labels
age_bins = [0,9,14,19,24,29,34,39,50]
age_labels = ['<10', '10-14', '15-19', '20-24', '25-29', '30-34', '35-39', '40+']

purchase_df['Age Ranges'] = pd.cut(purchase_df['Age'], bins = age_bins, labels = age_labels)

# Calculate total purchase count,
purchase_counts_per_age_bin = purchase_df.groupby(['Age Ranges'])['Purchase ID'].count()
avg_purchase_price_per_age_bin = purchase_df.groupby(['Age Ranges'])['Price'].mean()
total_purchase_value_per_age_bin = purchase_df.groupby(['Age Ranges'])['Price'].sum()
#avg_total_purchase_per_person = ???

purchase_by_age_df = pd.DataFrame({'Purchase Count': purchase_counts_per_age_bin,\
                                   'Average Purchase Price': avg_purchase_price_per_age_bin,\
                                   'Total Purchase Value': total_purchase_value_per_age_bin,\
                                   'Avg Total Purchase Per Person': ['?', '?', '?', '?', '?', '?', '?', '?']})

# Change formatting for cleaner display
purchase_by_age_df.style.format({'Average Purchase Price': "${:,.2f}",\
                                 'Total Purchase Value': "${:,.2f}"})

```

Out[21]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase Per Person
<b>Age Ranges</b>				
<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	\$1,114.06	?
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

```
In [22]: # Group by SN to study player statistics
purchase_count = (purchase_df.groupby(["SN"])["Item ID"].count())
average_price = (purchase_df.groupby(["SN"])["Price"].mean())
total_revenue = (purchase_df.groupby(["SN"])["Price"].sum())

# Summary Table for purchasing analysis by SN
summary_purchase_by_SN = pd.DataFrame({"Purchase Count": purchase_count,\
                                       "Average Purchase Price": average_price,\
                                       "Total Purchase Value": total_revenue})

# Sort summary DataFrame by Total purchase value first before performing formatting
summary_purchase_by_SN = summary_purchase_by_SN.sort_values(["Total Purchase Value"],ascending = False)

# Change formatting for cleaner display
summary_purchase_by_SN.head().style.format({'Average Purchase Price': "${:,.2f}", 'Total Purchase Value': '${:,.2f}'})
```

Out[22]:

	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

## Most Popular Items



```
In [23]: purchase_count = purchase_df.groupby(['Item ID', 'Item Name', 'Price'])['Item ID'].count()
total_purchase_value = purchase_df.groupby(['Item ID', 'Item Name', 'Price'])['Price'].sum()

popular_df = pd.DataFrame({'Purchase Count': purchase_count,\
                           'Total Purchase Value': total_purchase_value}).reset_index()
popular_df = popular_df[['Item ID', 'Item Name', 'Purchase Count', 'Price', 'Total Purchase Value']]
popular_df = popular_df.rename(columns = {'Price': 'Item Price'})
popular_df = popular_df.set_index(['Item ID', 'Item Name'])
popular_df = popular_df.sort_values(['Purchase Count'], ascending = False)

popular_df.head().style.format({'Item Price': "${:,.2f}", 'Total Purchase Value': '${:,.2f}'})
```

Out[23]:

		Purchase Count	Item Price	Total Purchase Value
Item 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

```
In [24]: profitable_df = popular_df.sort_values('Total Purchase Value', ascending = False)
profitable_df.head().style.format({'Item Price': "${:,.2f}", 'Total Purchase Value': '${:,.2f}'})
```

Out[24]:

		Purchase Count	Item Price	Total Purchase Value
Item 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 [ ]: