

## 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]:

```
import pandas as pd

pd.df = "Resources/purchase_data.csv"

purchase_df = pd.read_csv(pd.df)
```

## Player Count

- Display the total number of players

In [2]:

```
players = purchase_df['SN'].nunique()
tot_players = pd.DataFrame({"Total Players" : [players]})
tot_players
```

Out[2]:

Total 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 [3]: unique_items = purchase_df["Item ID"].nunique()
avg_price = (purchase_df["Price"].sum() /
             purchase_df["Price"].count()).round(2) #round 2 places
tot_purchases = purchase_df["Price"].count()
tot_revenue = purchase_df["Price"].sum()

purchase_analysis_df = pd.DataFrame({"Number of Unique Items" : [unique_items],
                                   "Average Price" : [avg_price],
                                   "Number of Purchases" : [tot_purchases],
                                   "Total Revenue" : [tot_revenue]})
purchase_analysis_df["Average Price"] = purchase_analysis_df["Average Price"]
purchase_analysis_df["Total Revenue"] = purchase_analysis_df["Total Revenue"]

purchase_analysis_df

```

Out[3]:

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

## Gender Demographics

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

```
In [6]: total_players = purchase_df["SN"].nunique()
male_players = purchase_df[purchase_df["Gender"] == "Male"]["SN"].nunique()
female_players = purchase_df[purchase_df["Gender"] == "Female"]["SN"].nunique()
other_players = (total_players - male_players) - female_players

male_per = ((male_players / total_players) * 100)
female_per = ((female_players / total_players) * 100)
other_per = ((other_players / total_players) * 100)

gender_df = pd.DataFrame({"Gender" : [ "Male" , "Female", "Other/Non-Disclosed" ],
                          "Total Count" : [male_players, female_players, other_players],
                          "Percentage of Players" : [male_per, female_per, other_per]})

gender_df["Percentage of Players"] = gender_df["Percentage of Players"].astype(float)

gender_df
```

Out[6]:

	Gender	Total Count	Percentage 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
- Create a summary data frame to hold the results
- Optional: give the displayed data cleaner formatting
- Display the summary data frame

```

In [8]: tot_purchase = purchase_df["Price"].count()
male_purchase = purchase_df[purchase_df["Gender"] == "Male"]["Price"].count()
female_purchase = purchase_df[purchase_df["Gender"] == "Female"]["Price"].count()
other_purchase = (tot_purchase - male_purchase) - female_purchase

tot_price_avg = purchase_df["Price"].mean()
male_price_avg = purchase_df[purchase_df["Gender"] == "Male"]["Price"].mean()
female_price_avg = purchase_df[purchase_df["Gender"] == "Female"]["Price"].mean()
other_price_avg = purchase_df[purchase_df["Gender"] == "Other / Non-Disclosed"]["Price"].mean()

male_price_tot = purchase_df[purchase_df["Gender"] == "Male"]["Price"].sum()
female_price_tot = purchase_df[purchase_df["Gender"] == "Female"]["Price"].sum()
other_price_tot = purchase_df[purchase_df["Gender"] == "Other / Non-Disclosed"]["Price"].sum()

total_purchase_count = purchase_df["SN"].count()
male_purchase_count = purchase_df[purchase_df["Gender"] == "Male"]["SN"].count()
female_purchase_count = purchase_df[purchase_df["Gender"] == "Female"]["SN"].count()
other_purchase_count = (total_purchase_count - male_purchase_count) - female_purchase_count

male_avg_tot = male_price_tot / male_players
female_avg_tot = female_price_tot / female_players
other_avg_tot = other_price_tot / other_players

gender_purchase_df = pd.DataFrame({"Gender" : [ "Male" , "Female", "Other/Non-Disclosed" ],
                                   "Purchase Count" : [male_purchase, female_purchase, other_purchase],
                                   "Avg Purchase Price" : [male_price_avg, female_price_avg, other_price_avg],
                                   "Total Purchase Value" : [male_price_tot, female_price_tot, other_price_tot],
                                   "Avg Total Purchase per Person" : [male_avg_tot, female_avg_tot, other_avg_tot]})

gender_purchase_df["Avg Purchase Price"] = gender_purchase_df["Avg Purchase Price"] * 100
gender_purchase_df["Total Purchase Value"] = gender_purchase_df["Total Purchase Value"] * 100
gender_purchase_df["Avg Total Purchase per Person"] = gender_purchase_df["Avg Total Purchase per Person"] * 100

gender_purchase_df

```

Out[8]:

	Gender	Purchase Count	Avg Purchase Price	Total Purchase Value	Avg Total Purchase per Person
0	Male	652	\$3.02	\$1967.64	\$4.07
1	Female	113	\$3.20	\$361.94	\$4.47
2	Other/Non-Disclosed	15	\$3.35	\$50.19	\$4.56

## 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 [9]: bins = [0, 9.90, 14.90, 19.90, 24.90, 29.90, 34.90, 39.90, 99999]

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

age_purchase_df = purchase_df

age_purchase_df["Age Summary"] = pd.cut(age_purchase_df["Age"], bins, labels=group)

age_purchase_df = age_purchase_df.groupby("Age Summary")
age_purchase_df.count()

summary_age_df = pd.DataFrame(age_purchase_df.count())
summary_age_df

summary_age_df["Purchase ID"] = (summary_age_df["Purchase ID"] / tot_purchase) * 100
summary_age_df

summary_age_df["Purchase ID"] = summary_age_df["Purchase ID"].map("{:,.2f}%")
summary_age_df

sum_age_df = summary_age_df[["Purchase ID", "SN"]]
sum_age_df

sum_fin_df = sum_age_df.rename(columns={"Purchase ID": "Percentage of Players",
                                         "SN": "Total Count"})
sum_fin_df
```

Out[9]:

	Percentage of Players	Total Count
Age Summary		
<10	2.95%	23
10-14	3.59%	28
15-19	17.44%	136
20-24	46.79%	365
25-29	12.95%	101
30-34	9.36%	73
35-39	5.26%	41
40+	1.67%	13

## 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 [10]: by_age_df = pd.DataFrame(age_purchase_df["Purchase ID"].count())
by_age_df

tot_pur_val_df = age_purchase_df["Price"].sum()
tot_pur_val_df

tot_purchase_value_df = tot_pur_val_df.map("${:.2f}".format)
tot_purchase_value_df

avg_price_age_df = age_purchase_df["Price"].mean()
avg_price_age_df

avg_price_byage_df = avg_price_age_df.map("${:.2f}".format)
avg_price_byage_df

avg_tot_pur_age = tot_pur_val_df/age_purchase_df["Purchase ID"].count()
avg_tot_pur_age_df = avg_tot_pur_age.map("${:.2f}".format)
avg_tot_pur_age_df

by_age_df["Average Purchase Price"] = avg_price_byage_df
by_age_df["Total Purchase Value"] = tot_purchase_value_df
by_age_df["Avg Total Purchase per Person"] = avg_tot_pur_age_df
by_age_df

summary_age_total = by_age_df.rename(columns={"Purchase ID" : "Purchase Count"})
summary_age_total

```

Out[10]:

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

## 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 [11]: top_purchase_df = pd.DataFrame(purchase_df)
top_purchase_df.head()

top_spenders_df = top_purchase_df.groupby("SN")
top_spenders_df.count()

by_spender_df = pd.DataFrame(top_spenders_df["Purchase ID"].count())
by_spender_df

sn_tot_pur_value = top_spenders_df["Price"].sum()

sn_tot_pur_value

avg_pur_sn = top_spenders_df["Price"].mean()

by_spender_df["Average Purchase Price"] = avg_pur_sn
by_spender_df["Total Purchase Value"] = sn_tot_pur_value
by_spender_df["Purchase ID"] = by_spender_df
by_spender_df

sum_sn = by_spender_df.rename(columns={"Purchase ID" : "Purchase Count"})
sum_sn

top_five = sum_sn.sort_values("Total Purchase Value", ascending=False)

sn_tot_pur_value = sn_tot_pur_value.map("${:.2f}".format)
avg_pur_sn = avg_pur_sn.map("${:.2f}".format)

top_five.head()
```

Out[11]:

	Purchase Count	Average Purchase Price	Total Purchase Value
SN			
<b>Lisosia93</b>	5.0	3.792000	18.96
<b>Idastidru52</b>	4.0	3.862500	15.45
<b>Chamjask73</b>	3.0	4.610000	13.83
<b>Iral74</b>	4.0	3.405000	13.62
<b>Iskadarya95</b>	3.0	4.366667	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 [12]: top_item_df = top_purchase_df.groupby(["Item ID", "Item Name"])
top_item_df

by_item_df = pd.DataFrame(top_item_df["Purchase ID"].count())
by_item_df

tot_item_vale = top_item_df["Price"].sum()
tot_item_vale
tot_item_value = tot_item_vale.map("${:.2f}".format)
tot_item_value

pur_price = top_item_df["Price"].mean()
pur_price
pur_price_tot = pur_price.map("${:.2f}".format)
pur_price_tot

by_item_df["Item Price"] = pur_price_tot
by_item_df["Total Purchase Price"] = tot_item_value
by_item_df

sum_by_item_df = by_item_df.rename(columns={"Purchase ID" : "Purchase Count",
                                             "Item Price" : "Item Price",
                                             "Total Purchase Price" : "Total Purchase Price"})

top_five=sum_by_item_df.sort_values("Purchase Count", ascending=False)
top_five.head()
```

Out[12]:

Item ID	Item Name	Purchase Count	Item Price	Total Purchase Price
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 [14]: sum_by_item_df["Total Purchase Value"] = top_item_df["Price"].sum()
sum_by_item_df

top_five=sum_by_item_df.sort_values("Total Purchase Value", ascending=0)
top_five.head()
```

Out[14]:

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

### SUMMARY

Based on the data provided we can make three conclusions.

1. The top five items are: Oathbreaker, Last Hope of the Breaking Storm, it has 12 purchases with a total purchase value of 50.76. *Followed closely by Nirvana with 44.10 purchase value.* The least selling objects are: The Decapitator and The Gladiator's Glaive.
2. The top spender is: Lisosia93 with almost \$19.00 in spending on five items. The spending trend is a bit low, maybe some types of advertising on the most popular items could help the overall spending.
3. The majority of players are between the age of 20-24 with almost half of all players, 46.79%, being in this age range. They also provided the most in spending at \$1,114.06. I would suggest to advertise heavily on social networking sites that reach this age demographic.

In [ ]: