

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 [35]: # Dependencies and Setup
import pandas as pd
import numpy as np

# File to Load (Remember to Change These)
file_to_load = "Resources/purchase_data.csv"

# Read Purchasing File and store into Pandas data frame
df = pd.read_csv(file_to_load)

df.head()
```

Out[35]:

	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

- Display the total number of players

```
In [36]: # Total Players
total_players = len(df['SN'].unique())
total_players = pd.DataFrame({'Total Players':[total_players]})
total_players
```

Out[36]:

Total Players	
0	576

```
In [37]: # Gender Demographics
gender_df = df.groupby('Gender')
gender = gender_df['SN'].nunique()

total_gender = gender.sum()
male = gender['Male']
female = gender['Female']
other = gender['Other / Non-Disclosed']

# Percentage by gender
pct_male = (male/total_gender)*100
pct_female = (female/total_gender)*100
pct_other = (other/total_gender)*100

gender_demo = {'Total Count': [male,female,other],
               'Percentage of Players':[pct_male, pct_female, pct_other]
}
gender_demo
gender_demographics = pd.DataFrame(gender_demo, index = ['Male', 'Female', 'Other / Non-Disclosed'])

gender_demographics['Percentage of Players'] = gender_demographics['Percentage of Players'].map('{:,.2f}'.format)
gender_demographics
```

Out[37]:

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

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 [38]: # Calculations by Unique, Price, Purchase ID, Total Revenue
num_unique = df['Item ID'].nunique()
avg_price = df['Price'].mean()
num_purchase = df['Purchase ID'].count()
total_rev = df['Price'].sum()

analysis_df = pd.DataFrame({'Number of Unique Items':[num_unique], 'Average Price':[avg_price], 'Number of Purchases':[num_purchase], 'Total Revenue':[total_rev]})

analysis_df['Average Price'] = analysis_df['Average Price'].map('${:,.2f}'.format)
analysis_df['Total Revenue'] = analysis_df['Total Revenue'].map('${:,.2f}'.format)
analysis_df.head()
```

Out[38]:

	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 [39]: # Gender Count by Male, Female, Other / Non-Disclosed
gender_df = df.groupby('Gender')
gender = gender_df['SN'].nunique()

total_gender = gender.sum()
male = gender['Male']
female = gender['Female']
other = gender['Other / Non-Disclosed']

pct_male = (male/total_gender)*100
pct_female = (female/total_gender)*100
pct_other = (other/total_gender)*100

gender_demo = {'Total Count': [male,female,other],
               'Percentage of Players':[pct_male, pct_female, pct_other]
}
gender_demo
gender_demographics = pd.DataFrame(gender_demo, index = ['Male', 'Female', 'Other / Non-Disclosed'])

gender_demographics['Percentage of Players'] = gender_demographics['Percentage of Players'].map('{:,.2f}'.format)
gender_demographics
```

Out[39]:

	Total Count	Percentage of Players
Male	484	84.03
Female	81	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 [40]: # Gender Analysis
gender_df = df.groupby('Gender')
pur_count = gender_df['Purchase ID'].count()
avg_pur_price = gender_df['Price'].mean()
tot_pur_value = gender_df['Price'].sum()
avg_tot_pur = tot_pur_value / gender

purchase_analysis = pd.DataFrame({'Purchase Count' : pur_count,
                                  'Average Purchase Price': avg_pur_price,
                                  'Total Purchase Value': tot_pur_value,
                                  'Avg Total Purchase per Person' : avg_tot_pur})
purchase_analysis['Average Purchase Price'] = purchase_analysis['Average Purchase Price'].map('${:, .2f}'.format)
purchase_analysis['Total Purchase Value'] = purchase_analysis['Total Purchase Value'].map('${:, .2f}'.format)
purchase_analysis['Avg Total Purchase per Person'] = purchase_analysis['Avg Total Purchase per Person'].map('${:, .2f}'.format)

purchase_analysis.head()
```

Out[40]:

	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

- 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 [41]: # Age Demo
bins = [0, 9.99, 14.99, 19.99, 24.99, 29.99, 34.99, 39.99, 100]
group_age = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+" ]

pd.cut(df['Age'], bins, labels=group_age)
df["Age Group"] = pd.cut(df['Age'], bins, labels=group_age)

players_df= df.drop_duplicates(['SN'], keep='first')
players_total = players_df['SN'].count()

# Calculating Total Count and Percentage of Playe\\

total_count_10 = len(players_df.loc[players_df["Age Group"] == "<10"])
pct_10 = (total_count_10 / players_total)*100

total_count_14 = len(players_df.loc[players_df["Age Group"] == "10-14"
])
pct_14 = (total_count_14 / players_total)*100

total_count_19 = len(players_df.loc[players_df["Age Group"] == "15-19"
])
pct_19 = (total_count_19 / players_total)*100

total_count_24 = len(players_df.loc[players_df["Age Group"] == "20-24"
])
pct_24 = (total_count_24 / players_total)*100

total_count_29 = len(players_df.loc[players_df["Age Group"] == "25-29"
])
pct_29 = (total_count_29 / players_total)*100

total_count_34 = len(players_df.loc[players_df["Age Group"] == "30-34"
])
pct_34 = (total_count_34 / players_total)*100

total_count_39 = len(players_df.loc[players_df["Age Group"] == "35-39"
])
pct_39 = (total_count_39 / players_total)*100

total_count_40 = len(players_df.loc[players_df["Age Group"] == "40+" ])
pct_40 = (total_count_40 / players_total)*100

age_demo = ({'Total Count': [total_count_10, total_count_14, total_count_19, total_count_24,
                             total_count_29, total_count_34, total_count_39, total_count_40],
              'Percentage of Players': [pct_10,pct_14,pct_19,pct_24,
                                         pct_29,pct_34,pct_39,pct_40]})
age_demographics = pd.DataFrame(age_demo, index = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+" ])

```

```
9", "20-24", "25-29", "30-34", "35-39", "40+" ] )

age_demographics['Percentage of Players'] = age_demographics['Percentage of Players'].map('{:, .2f}'.format)

age_demographics
```

Out[41]:

	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 [45]: # Purchasing
age_df = df.groupby('Age Group')
age_df.head()
pur_count = age_df['Purchase ID'].count()
avg_pur_price = age_df['Price'].mean()
tot_pur_val= age_df['Price'].sum()

players_age_df = players_df.groupby('Age Group').SN.count()
avg_pur_per = tot_pur_val / players_age_df
avg_pur_per

pur_analysis = {"Purchase Count" : pur_count, "Average Purchase Price"
: avg_pur_price,
                "Total Purchase Value" : tot_pur_val, "Avg Total Purchase
ase per Person" : avg_pur_per}

purchase_analysis_age = pd.DataFrame(pur_analysis, index = group_age)

purchase_analysis_age['Average Purchase Price'] = purchase_analysis_age[
'Average Purchase Price'].map('${:,.2f}'.format)
purchase_analysis_age['Total Purchase Value'] = purchase_analysis_age[
'Total Purchase Value'].map('${:,.2f}'.format)
purchase_analysis_age['Avg Total Purchase per Person'] = purchase_anal
ysis_age['Avg Total Purchase per Person'].map('${:,.2f}'.format)

purchase_analysis_age

```

Out[45]:

	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	\$1,114.06	\$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 [23]: sn_df = df.groupby('SN')
sn_df.head()
pur_count = sn_df['Item ID'].count()
avg_pur_price = sn_df['Price'].mean()
tot_pur_value = pur_count * avg_pur_price

top_spender = {"Purchase Count": pur_count, "Average Purchase Price": avg_pur_price, "Total Purchase Value": tot_pur_value}
top_spenders = pd.DataFrame(top_spender, index = players_df['SN'])
top_spenders = top_spenders.sort_values(by='Total Purchase Value', ascending = False)

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

top_spenders.head()
```

Out[23]:

	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

- 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 [24]: popular_df = df.groupby(['Item ID', 'Item Name'])
popular_df.head()
pur_count = popular_df['Item ID'].count()
price = popular_df['Price'].mean()
tot_pur_value = pur_count * price
pop_items = {'Purchase Count': pur_count, 'Item Price': price, 'Total
Purchase Value': tot_pur_value}

#item_id = df_popular['Item ID']

popular_items = pd.DataFrame(pop_items)
popular_items = popular_items.sort_values(by="Purchase Count", ascending = False)

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

popular_items.head()
```

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
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 [28]: df_profitable = df.groupby(['Item ID', 'Item Name'])
purchase_count = df_profitable['Item Name'].count()
item_price = df_profitable['Price'].mean()
total_purchase_value = pur_count * item_price

item_id = df['Item ID'].unique()
item_name = df['Item Name'].unique()

profit_items = {'Purchase Count': purchase_count, 'Item Price': item_p
rice, 'Total Purchase Value': total_purchase_value}
profitable_items = pd.DataFrame(profit_items)
profitable_items = profitable_items.sort_values(by='Total Purchase Val
ue', ascending = False)

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

profitable_items.head()

```

Out[28]:

		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 []: