

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 [8]: # Dependencies and Setup
import pandas as pd

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

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

Out[8]:

	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 [9]: total_players = len(purchase_data["SN"].unique())
total_players_pd = pd.DataFrame({"Total Players": [total_players]})
total_players_pd
```

Out[9]:

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 [10]: total_items = len(purchase_data["Item Name"].unique())
average_price = purchase_data["Price"].mean()
total_number_purchases = purchase_data["Purchase ID"].count()
total_revenue = purchase_data["Price"].sum()
purchase_summary_table = pd.DataFrame({"Total Items": total_items,
                                       "Average Price": average_price,
                                       "Total Number Purchases": total_number_purchases,
                                       "Total Revenue": total_revenue,
                                       }, index=[0])

purchase_summary_table["Average Price"] = purchase_summary_table["Average Price"]
purchase_summary_table["Total Revenue"] = purchase_summary_table["Total Revenue"]

purchase_summary_table.head()
```

Out[10]:

	Total Items	Average Price	Total Number Purchases	Total Revenue
0	179	\$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 [11]: total_gender = purchase_data["Gender"].count()
male = purchase_data["Gender"].value_counts()['Male']
female = purchase_data["Gender"].value_counts()['Female']
non_gender = total_gender - male - female
male_percent = male/total_gender *100
female_percent = female/total_gender *100
non_gender_percent = non_gender/total_gender * 100
gender_breakdown = pd.DataFrame({"Total Gender":[total_gender],
                                "Male":[male],
                                "Female":[female],
                                "Non Gender Spe":[non_gender],
                                "% Male":[male_percent],
                                "% Female":[female_percent],
                                "% Non Gender Spe":[non_gender_percent],
                                })
gender_breakdown = gender_breakdown[["Total Gender", "Male", "Female", "Non Gender Spe"]]
gender_breakdown = gender_breakdown.round(2)
gender_breakdown

```

Out[11]:

	Total Gender	Male	Female	Non Gender Spe	% Male	% Female	% Non Gender Spe
0	780	652	113	15	83.59	14.49	1.92

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 [12]: # male count
male_purchase_data = purchase_data.loc[purchase_data["Gender"] == "Male", :]
male_purchase_count = len(male_purchase_data)

# male average purchase price
avg_male_purchase_price = purchase_data.loc[purchase_data["Gender"] == "Male", :]

# total male purchase
total_male_purchase_value = purchase_data.loc[purchase_data["Gender"] == "Male", :]

# female count
female_purchase_data = purchase_data.loc[purchase_data["Gender"] == "Female", :]
female_purchase_count = len(female_purchase_data)

# female average purchase price
avg_female_purchase_price = purchase_data.loc[purchase_data["Gender"] == "Female", :]

# total female purchase
total_female_purchase_value = purchase_data.loc[purchase_data["Gender"] == "Female", :]

# other count
other_purchase_data = purchase_data.loc[purchase_data["Gender"] == "Other / Non-Disclosed", :]
other_purchase_count = len(other_purchase_data)

# other average purchase price
avg_other_purchase_price = purchase_data.loc[purchase_data["Gender"] == "Other / Non-Disclosed", :]

# total other purchase
total_other_purchase_value = purchase_data.loc[purchase_data["Gender"] == "Other / Non-Disclosed", :]

# average purchase per person
avg_male_purchase_total_person = total_male_purchase_value / male_purchase_count
avg_female_purchase_total_person = total_female_purchase_value / female_purchase_count
avg_other_purchase_total_person = total_other_purchase_value / other_purchase_count

# create data table
gender_purchase_ana_table = pd.DataFrame([
    {
        "Gender": "Female", "Purchase Count": female_purchase_count,
        "Average Purchase Price": "${:.2f}".format(avg_female_purchase_price[0]),
        "Total Purchase Value": "${:.2f}".format(total_female_purchase_value[0]),
        "Avg Total Purchase per Person": "${:.2f}".format(avg_female_purchase_total_person[0]),
    },
    {
        "Gender": "Male", "Purchase Count": male_purchase_count,
        "Average Purchase Price": "${:.2f}".format(avg_male_purchase_price[0]),
        "Total Purchase Value": "${:.2f}".format(total_male_purchase_value[0]),
        "Avg Total Purchase per Person": "${:.2f}".format(avg_male_purchase_total_person[0]),
    },
    {
        "Gender": "Other / Non-Disclosed", "Purchase Count": other_purchase_count,
        "Average Purchase Price": "${:.2f}".format(avg_other_purchase_price[0]),
        "Total Purchase Value": "${:.2f}".format(total_other_purchase_value[0]),
        "Avg Total Purchase per Person": "${:.2f}".format(avg_other_purchase_total_person[0]),
    },
], columns=["Gender", "Purchase Count", "Average Purchase Price", "Total Purchase Value", "Avg Total Purchase per Person"])

gender_purchase_ana_table = gender_purchase_ana_table.set_index("Gender")
gender_purchase_ana_table.index.name = None
gender_purchase_ana_table

```

Out[12]:

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

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 [13]: # Establish Bins for Ages & Create Corresponding Names For Bins
age_bins = [0, 9, 14, 19, 24, 29, 34, 39, 46]
groups_names = [<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40-46"]

# Place Data Series Into New Column Inside DataFrame
purchase_data["Age Group"] = pd.cut(purchase_data["Age"], bins=age_bins, labels=groups_names)

# Create a GroupBy Object Based Upon "Age Group"
age_group = purchase_data.groupby("Age Group")

# Count Total Players by Age Category
total_age_count = age_group["SN"].nunique()

# Calculate Percentages by Age Category
percentage_by_age = round(total_age_count / total_players * 100,2)

# Create Summary DataFrame
age_demographics_table = pd.DataFrame({
    "Total Count": total_age_count,
    "Percentage of Players": percentage_by_age
})

age_demographics_table["Percentage of Players"] = age_demographics_table["Percentage of Players"] / 100
age_demographics_table.index.name = None
age_demographics_table

```

Out[13]:

	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 [14]: # Number of Unique Items
unique_items = len(purchase_data["Item ID"].unique())

# Average Purchase Price
avg_purchase_price = purchase_data["Price"].mean()

# Total Number of Purchases
total_purchases = len(purchase_data["Purchase ID"].unique())

# Total Revenue
total_revenue = purchase_data["Price"].sum()

# Create Summary DataFrame
purchasing_analysis_table = pd.DataFrame([
    "Number of Unique Items": unique_items,
    "Average Price": avg_purchase_price,
    "Number of Purchases": total_purchases,
    "Total Revenue": total_revenue,
]), columns=["Number of Unique Items", "Average Price", "Number of Purchases", "Total Revenue"]

purchasing_analysis_table["Average Price"] = purchasing_analysis_table["Average Price"].round(2)
purchasing_analysis_table["Total Revenue"] = purchasing_analysis_table["Total Revenue"].round(2)
purchasing_analysis_table

```

Out[14]:

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

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 [15]: top_spenders = purchase_data.groupby("SN")

# Calculate "Purchase Count"
spender_purchase_count = top_spenders["Purchase ID"].count()

# Calculate "Average Purchase Price"
average_spender_purchase_price = round(top_spenders["Price"].mean(),2)

# Calculate "Total Purchase Value"
total_spender_purchase_value = top_spenders["Price"].sum()

# Create Summary DataFrame
top_spenders_table = pd.DataFrame({
    "Purchase Count": spender_purchase_count,
    "Average Purchase Price": average_spender_purchase_price,
    "Total Purchase Value": total_spender_purchase_value
})

sort_top_spenders = top_spenders_table.sort_values(["Total Purchase Value"], ascending=False)
sort_top_spenders["Average Purchase Price"] = sort_top_spenders["Average Purchase Price"]
sort_top_spenders["Total Purchase Value"] = sort_top_spenders["Total Purchase Value"]
sort_top_spenders
```

Out[15]:

	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 [16]: popular_items_list = purchase_data[["Item ID", "Item Name", "Price"]]

popular_items = popular_items_list.groupby(["Item ID", "Item Name"])

item_purchase_count = popular_items["Price"].count()

item_price = popular_items["Price"].sum()

item_purchase_value = item_price / item_purchase_count

# create data frame
most_popular_items = pd.DataFrame({
    "Purchase Count": item_purchase_count,
    "Item Price": item_purchase_value,
    "Total Purchase Value": item_price
})

popular_items_formatted = most_popular_items.sort_values(["Purchase Count"], ascending=False)
popular_items_formatted["Item Price"] = popular_items_formatted["Item Price"].astype(float)
popular_items_formatted["Total Purchase Value"] = popular_items_formatted["Total Purchase Value"].astype(float)
popular_items_formatted

```

Out[16]:

		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 [17]: popular_items_formatted = most_popular_items.sort_values(["Total Purchase Value"]
popular_items_formatted["Item Price"] = popular_items_formatted["Item Price"].as
popular_items_formatted["Total Purchase Value"] = popular_items_formatted["Total
popular_items_formatted
```

Out[17]:

		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 [18]: # Export File As CSV
purchase_data.to_csv("Output/purchase_data_revised.csv", index=False, header=True)
```

```
In [ ]: # written description of three observable trends based on the data
#1. 83.59% buyers are males.
#2. 44.79% of total purchases were made by age group 20-24;
#3. The most popular and profitable games have no significant profit/sales comparison
```

In []:

In []:

In []: