

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 []: `# Trend analysis`

In []: `# 1.Over 84% of the 576 total players are males.
#The total revenue is 2,379.77 USD, where 1,967.64 USD is from purchases made by male p
with the overwhelming mjprity of playership and total purchase from male, , female pl`

In []: `# 2. the players between 20-24 make up almost half of the player grou(45%) and the seco
#the least palyer are aged over 40 years old.`

In []: `#3. The most popular items sold were Final Critic , Oathbreaker, Last Hope of the Break
these two items above also contribute the most to the sales`

In [1]: `# 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)`

Player Count

- Display the total number of players

In [2]: `players=purchase_data["SN"].nunique()
players_Total= pd.DataFrame({'Total Players': [players]})
players_Total.head()`

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]: Uniqueitem=purchase_data["Item ID"].nunique()
Meanprice=purchase_data["Price"].mean()
Meanprice2=round(Meanprice,2)
Purchase=purchase_data["Purchase ID"].count()
Revenues=purchase_data["Price"].sum()
Purchaseanalysis=pd.DataFrame({'Number of Unique Items': [Uniqueitem], 'Average Price':
pd.options.display.float_format='{:,}'.format
Purchaseanalysis.head()
```

```
Out[3]:
```

	Number of Unique Items	Average Price	Number of 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 [4]: Onlymale=purchase_data.loc[purchase_data["Gender"]=="Male"]
Onlymalecount=Onlymale["SN"].nunique()
Onlymalepr=Onlymalecount/players*100

Onlyfemale=purchase_data.loc[purchase_data["Gender"]=="Female"]
Onlyfemalecount=Onlyfemale["SN"].nunique()
Onlyfemalepr=Onlyfemalecount/players*100

Other=purchase_data.loc[purchase_data["Gender"]=="Other / Non-Disclosed"]
Othercount=Other["SN"].nunique()
Othercountpr=Othercount/players*100

Demographics_analysis=pd.DataFrame({ "Gender": ["Male", "Female", "Other/ND"],
    "Counts of players": [Onlymalecount, Onlyfemalecount, Othercount],
    "Percentage of players": [Onlymalepr, Onlyfemalepr, Othercountpr]})
pd.options.display.float_format='{:.2f}%'.format
Demographics_analysis.head()
```

```
Out[4]:
```

	Gender	Counts of players	Percentage of players
0	Male	484	84.03%
1	Female	81	14.06%
2	Other/ND	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 [5]:

```

malepurchasec=Onlymale["Item ID"].count()
femalepurchasec=Onlyfemale["Item ID"].count()
otherpurchasec=Other["Item ID"].count()

maleavgprice=Onlymale["Price"].mean()
femaleavgprice=Onlyfemale["Price"].mean()
otheravgpric=Other["Price"].mean()
#maleavgprice
#femaleavgprice
#otheravgpric

malepurchase=Onlymale["Price"].sum()
femalepurchase=Onlyfemale["Price"].sum()
otherpurchase=Other["Price"].sum()

maleavgpurchase=Onlymale["Price"].sum()/Onlymalecount
femaleavgpurchase=Onlyfemale["Price"].sum()/Onlyfemalecount
otheravgpurchase=Other["Price"].sum()/Othercount
#maleavgpurchase
#femaleavgpurchase
#otheravgpurchase

Gender_analysis=pd.DataFrame({ "Gender": ["Male", "Female", "Other/ND"], "Purchase Count":
pd.options.display.float_format='${:,.2f}'.format
Gender_analysis.head()

```

Out[5]:

	Gender	Purchase Count	Average Purchase Price	Total Purchase	Purchase per Person
0	Male	652	\$3.02	\$1967.64	\$4.07
1	Female	113	\$3.20	\$361.94	\$4.47
2	Other/ND	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 [6]:

```
bins=[0,9.9,14.9,19.9,24.9,29.9,34.9,39.9,100]
```

```

Age_Labels=["<10","10-14","15-19","20-24","25-29","30-34","35-39","40+"]
#Add a column:AGE GROUP into the DF
purchase_data["Age Group"]=pd.cut(purchase_data["Age"], bins, labels=Age_Labels)
purchase_data
#!!Slice the date into groups
Purchase_agegroup=purchase_data.groupby("Age Group")
# check date type after slicing the date, and this is a dataframe -> Purchase_agegroup.
# Aggregate the grouped by
Totalplayers= Purchase_agegroup.nunique()['SN']
Totalplayers
# aftergrouped by age the list is a series

pcplayers=Totalplayers/players*100

#state_office.div(state, level='state') * 100
AgeDemographicssum_df= pd.DataFrame({"Total Players" : Totalplayers,"Percentage of Play

pd.options.display.float_format='{:.2f}%'.format
AgeDemographicssum_df

```

Out[6]:

	Total Players	Percentage of Players
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)

- 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 [7]:

```

purchasecountage=Purchase_agegroup["Purchase ID"].count()
purchasecountage
avgpurchaseage=Purchase_agegroup["Price"].mean()

pd.options.display.float_format='${:.2f}'.format

```

```

totalpurchaseage=Purchase_agegroup["Price"].sum()
totalpurchaseagepp=totalpurchaseage/Totalplayers
totalpurchaseagepp

Agepurchasingsum_df= pd.DataFrame({"Purchase Count" : purchasecountage,"Average Purchas
Agepurchasingsum_df

```

Out[7]:

	Purchase Count	Average Purchase Price	Total Purchase Value	Avg Total Purchase per Person
Age Group				
<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	\$1114.06	\$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 [11]:

```

Playerdata=purchase_data.groupby("SN")

playerpurchasecount=Playerdata['Purchase ID'].count()
#purchase_count #*****

# Total purchase value by player:
Playerpurchasevalue=Playerdata['Price'].sum()

# Average purchase price by player:
Playeravgpurchaseprice=Playerpurchasevalue/playerpurchasecount

# Top spenders summary data frame:
topspenders_df=pd.DataFrame({'Purchase Count': playerpurchasecount,'Average Purchase Pr

topspenders_df.sort_values('Total Purchase Value', ascending=False).head()

```

Out[11]:

	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, average 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 [17]:

```
items_df=purchase_data[['Item ID', 'Item Name', 'Price']]
itemsgroup_df=items_df.groupby(['Item ID', 'Item Name'])
itempurchasecount=itemsgroup_df['Price'].count()
itempurchasetotal=itemsgroup_df['Price'].sum()
itemavgpurchase=itempurchasetotal/itempurchasecount
popularitems_df=pd.DataFrame({'Purchase Count': itempurchasecount,'Item Price': itemavg
popularitems_df.sort_values('Purchase Count', ascending=False).head()
```

Out[17]:

		Purchase Count	Item Price	Total Purchase Value
Item ID	Item Name			
92	Final Critic	13	\$4.61	\$59.99
178	Oathbreaker, Last Hope of the Breaking Storm	12	\$4.23	\$50.76
145	Fiery Glass Crusader	9	\$4.58	\$41.22
132	Persuasion	9	\$3.22	\$28.99
108	Extraction, Quickblade Of Trembling Hands	9	\$3.53	\$31.77

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 [18]: mostvalueitems_df=popularitems_df.sort_values('Total Purchase Value', ascending=False)
mostvalueitems_df.head()
```

```
Out[18]:
```

		Purchase Count	Item Price	Total Purchase Value
Item ID	Item Name			
92	Final Critic	13	\$4.61	\$59.99
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
103	Singed Scalpel	8	\$4.35	\$34.80

```
In [ ]:
```