1. **# NAME : Kalyani Pardeshi**
2. **# ROLL NO : 750**

**3 # PRN NO : 202201040114**

**4 # BATCH : G3**

7 import numpy as np 8 import pandas as pd

9 all\_data=pd.read\_csv("/content/1686715083343\_all\_data (7).csv")

10 all\_data.head()

**Order**

**ID**

**Product**

**Quan tity Orde red**

**Pri ce Ea ch**

**Order Date Purchase Address**

**0** 176559.0 Bose

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| So  **1** | undSport  176560.0 | Headphones  Google Phone | 2019  1.0 | 600.00 |
| **2** | 176560.0 | Wired Headphones | 1.0 | 11.99 |
| **3** | 176561.0 | Wired Headphones | 1.0 | 11.99 |

1.0 99.99 04-07-

22:30

04-12-2019

682 Chestnut St, Boston,

MA 02215

14:38 669 Spruce St, Los Angeles,

04-12-2019

14:38

CA 90001

669 Spruce St, Los Angeles,

CA 90001

05/30/19 9:27 333 8th St, Los Angeles, CA 90001

381 Wilson St San Francisco CA

1

1 #clean up the data 2 all\_data.shape

(69, 6)

1 # drop rows of nana

2 nan\_df=all\_data[all\_data.isna().any(axis=1)]

3 display(nan\_df.head())

**Order Product**

**Pri**

**Ord**

**Purchase**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **ID** | **QOuradenrted** | **cEeach** | **eDrate** | **Address** |
| **36** | NaN **ity**NaN | NaN | NaN | NaN | NaN |
| **51** | NaN NaN | NaN | NaN | NaN | NaN |

1

all\_data.sh ape

(69, 6)

1 all\_data=all\_data.dropna(how='all')

2 all\_data.head()

**Order**

**ID**

**Product**

**Quan tity Orde red**

**Pri ce Ea ch**

**Ord er Da te**

**Purchase Address**

**0** 176559.0 Bose

SoundSport

Headphones

1.0 99.99 04-

07-2019 22:30

682 Chestnut

St, Boston, MA

02215

**1** 176560.0 Google Phone 1.0 600.00 04-

12-2019

669 Spruce

1

**2** 176560.0 Wired 1.0 11.99 04-12-2019 669 Spruce St, Los

Headphones 14:38 Angeles, CA 90001

all\_data.sh ape

14:38

St, Los Angeles, CA 90001

(67, 6)

**3** 176561.0 Wired 1.0 11.99 05/30/19 333 8th St, Los

Headphones 9:27 Angeles, CA 90001

1 #get rid of text order date column

2 all\_data=all\_data[all\_data['Order Date'].str[0:2]!='Or']

3 print(all\_data)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Order  176559  .0 | Product  Bose SoundSport  Headphones | | Quantity  1.0 | Price  99.99 | \ |
| 0 |  |
| 1 | 176560 |  | Google | 1.0 | 600.00 |  |
| 2 | 176560 |  | Wired | 1.0 | 11.99 |  |
| 3 | 176561 |  | Wired | 1.0 | 11.99 |  |
| 4  .. | 176562  ... | USB-C Charging  ... | | 1.0  ... | 11.95  ... |  |
| 6 | 259329 | Lightning | | 1.0 | 14.95 |  |
| 4 | .0 | Charging Cable | |
| 6 | 259330 | AA Batteries | | 2.0 | 3.84 |  |
| 5 | .0 | (4-pack) | |
| 6 | 259331 | Apple Airpods | | 1.0 | 150.00 |  |
| 6 | .0 | Headphones | |
| 6 | 259332 | Apple Airpods | | 1.0 | 150.00 |  |
| 7 | .0 | Headphones | |
| 6 | 259333 | Bose SoundSport | | 1.0 | 99.99 |  |
|  | Order | |  | Purchase Address | |  |
| 0 | 04-07-2019 | | 682 Chestnut St, Boston, MA | | |  |
| 22:30 | | 02215 | | |
| 1 | 04-12-2019 | | 669 Spruce St, Los Angeles, CA | | |  |

|  |  |  |
| --- | --- | --- |
|  | 14:38 | 90001 |
| 2 | 04-12-2019  14:38 | 669 Spruce St, Los Angeles, CA  90001 |
| 3 | 05/30/19  9:27 | 333 8th St, Los Angeles, CA  90001 |
| 4 | 04/29/19  13:03 | 381 Wilson St, San Francisco, CA  94016 |
| .  . | ... | ... |
| 6  4 | 09-05-2019  19:00 | 480 Lincoln St, Atlanta, GA  30301 |
| 6  5 | 09/25/19  22:01 | 763 Washington St, Seattle, WA  98101 |
| 6  6 | 09/29/19  7:00 | 770 4th St, New York City, NY  10001 |
| 6  7 | 09/16/19  19:21 | 782 Lake St, Atlanta, GA 30301 |
| 6  8 | 09/19/19  18:03 | 347 Ridge St, San Francisco, CA 94016 |

[67 rows x 6 columns]

1 #make column correct type

2 all\_data['Quantity Ordered']=pd.to\_numeric(all\_data['Quantity Ordered']) 3 all\_data['Price

Each']=pd.to\_numeric(all\_data['Price Each']) 4 all\_data.head()

**Order**

**ID**

**Product**

**Quan tity**

**0** 176559.0 Bose SoundSport 1.0 99.99 04-07-2019 682 Chestnut St,

Headphones 22:30 Boston, MA 02215

**1** 176560.0 Google Phone 1.0 600.00 04-12-2019 669 Spruce St, Los

14:38 Angeles, CA 90001

**2** 176560.0 Wired 1.0 11.99 04-12-2019 669 Spruce St, Los

Headphones 14:38 Angeles, CA 90001

**3** 176561.0 Wired 1.0 11.99 05/30/19 333 8th St, Los

Headphones 9:27 Angeles, CA 90001

**Pri ce Ea**

**Ord er Da**

**Purchase Address**

1. all\_data['Month']= all\_data['Order Dat**O**e'**r**]**d**.s**e**tr[0:2]
2. all\_data['Month']= all\_data['Month'].ast**r**y**e**pe**d**('int3**c**2**h**') **te**
3. all\_data.head()

**Order Product Quantity Price Order Purchase Month**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Ordere**  **d** | **Each** | **Date** | **Address** |  |
| Bose |  |  | 04-07- | Chestnu 682 |  |
| **20** 1716756605.059.0 Wired | 11..00 | 991.19.999 | 2200191  194:38  22:3  9:27 | Los ABngoesletso, nCA,  MA 90001 | 44 |
| port |  |  | 333 8th S0t,2L2os15 |  |
| Headphones | 1.0 | 11.99 | Angeles, CA | 5 |
|  |  | 90001 |  |
| ones |
|  |  |  | 04-12- | 636891 WSiplsroun cSet |  |
| **1** 176560.0 Google | 1.0 | 600.0 | 2019 | Los | 4 |
|  |  |  | 14:38 | 90001 |  |

1 #Add city column

2 def get\_city(address):

3 return address.split(",")[1].strip(" ")

4 def get\_state(address):

5 return address.split(",")[2].strip(" ")[1] 6

7 all\_data['city']=all\_data['Purchase Address'].apply(lambda x:f"{get\_city(x)} ({get\_state(x)}))") 8 all\_data.head()

9

**Order**

**ID**

**Product Quantity**

**Ordered**

**Pri ce**

**Ord er**

**Purch ase**

**Month city**

Go ogle

Pho ne

**2** 176560.0 Wir

ed

Headphones

1.0

600.00

1.0

11.

99

04-

12-

20

19

14:

38

04-

12-

20

19

14:

38

02215

669

Spruce St,

Los 4

Angel es, CA 90001

669

Spruce St,

Los Angel es, CA

90001 4

333 8th St,

(A

Los

|  |  |  |  |
| --- | --- | --- | --- |
| Beos |  | **Ea**  **ch** | **Da Addre**  **te ss** |
|  |  |  | 07- 682 |
| **0** 176559.0 SoundSport | 1.0 |  | 20 Chest 4 |
| Headphones | 99 | 99. | 19 nut St, B  22: Boston, os  to |
|  |  |  | 30 MA n |
| **1** 176560.0 |  |  | )) |

A

ng el es (A

))

Los

A

ng el es (A

))

**3** 176561.0 Wired 1.0 11.99

05/30/19

Los

Los

5 Angeles

1 #waht was the best month for sales?how much was earned that month?

2 all\_data['Sales']=all\_data['Quantity Ordered'].astype('int')\*all\_data['Price Each'].astype('float')

3 all\_data.groupby(['Month']).sum()

4

<ipython-input-11-8fec2581ce34>:3: FutureWarning: The



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Month** | **Order ID** | **Quantity Ordered** | **Price Each** | **Sales** |
| **4** | 7335546.0 | 123.0 | 885.80 | 1210.76 |
| **5** | 353124.0 | 2.0 | 111.98 | 111.98 |
| **6** | 184076.0 | 1.0 | 14.95 | 14.95 |
| **8** | 726962.0 | 9.0 | 23.92 | 50.83 |
| **9** | 2378802.0 | 17.0 | 591.44 | 616.62 |
| **10** | 550924.0 | 11.0 | 10.67 | 39.69 |
| **11** | 740314.0 | 19.0 | 13.66 | 65.31 |
| **12** | 550635.0 | 17.0 | 8.97 | 50.83 |

default value of numeric\_onl all\_data.groupby(['Month']).sum()

1 #2)WHICH CITY SOLD THE MOST PRODUCT?

2 Dummycity=all\_data.groupby(['city'])

3 print(Dummycity)

4 #city\_max=all\_data.groupby(['city']).sum()

5 #print(max(city\_max))

<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7f62dbe6fd00>

1 #waht products are most often sold together

2 df=all\_data[all\_data['Order ID'].duplicated(keep=False)]

3 df['Grouped']=df.groupby('Order

ID')['Product'].transform(lambda x:','.join(x)) 4 df2=df[['Order ID','Grouped']].drop\_duplicates() 5 print(df['Grouped'])

1 Google Phone,Wired Headphones

2 Google Phone,Wired Headphones Name:

Grouped, dtype: object

<ipython-input-18-1970be6762a6>:3: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame. Try using

.loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) [docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy) df['Grouped']=df.groupby('Order ID')['Product'].transform(lambda x:','.join(x))

1 from itertools import combinations 2 from collections import Counter

3

4 count=Counter()

5

6 for row in df2['Grouped']:

7 row\_list=row.split(',')

8 count.update(Counter(combinations (row\_list,2))) 9

10 for key,value in count.most\_common(10):

11 print(key,value)

12

13

('Google Phone', 'Wired Headphones') 1

1 product\_group=all\_data.groupby('Product')

2 quantity\_ordered=product\_group.sum()['Quantity Ordered']

<ipython-input-20-11142b314e0e>:2: FutureWarning: The default value of numeric\_only in DataFrameGroupBy.sum is deprecated. In a

future version, numeric\_only will default to False. Ei quantity\_ordered=product\_group.sum()['Quantity Ordered'] 1 print(quantity\_ordered)

|  |  |
| --- | --- |
| Product |  |
| AA Batteries (4-  pack) | 64.  0 |
| AAA Batteries (4-  pack) | 109  .0 |
| Apple Airpods  Headphones | 3.  0 |
| Bose SoundSport  Headphones | 3.  0 |
| Google Phone | 1.  0 |
| Lightning Charging  Cable | 4.  0 |
| USB-C Charging  Cable | 8.  0 |
| Wired Headphones | 7.  0 |

Name: Quantity Ordered, dtype: float64

1 prices=all\_data.groupby('Product').mean()['Price Each']

<ipython-input-22-1f4f73bca841>:1: FutureWarning: The default value of numeric\_only in DataFrameGroupBy.mean is deprecated. In

|  |  |
| --- | --- |
| Product  AA Batteries (4-  pack) | 3.84 |
| AAA Batteries (4-  pack) | 2.99 |
| Apple Airpods  Headphones | 150.0  0 |
| Bose SoundSport  Headphones | 99.99 |
| Google Phone | 600.0  0 |

a future version, numeric\_only will default to False. E prices=all\_data.groupby('Product').mean()['Price Each'] 1 print(prices)

|  |  |
| --- | --- |
| Lightning Charging  Cable | 14.95 |
| USB-C Charging  Cable | 11.95 |
| Wired Headphones | 11.99 |

Name: Price Each, dtype: float64

