

## ▼ PRACTICAL NO.04

Practice Lab Assignment:

1. Perform all the pandas operations in python. Lab Assignment: Read any real life dataset. Store the data into Data Frames. Identify 20 grains for the given dataset. Implement all 20 grains using Pandas methods. The Sample Grains for Sales Dataset as:

- Which was the best month for sales? How much was earned that month?
- Which product sold the most? Why do you think it did?
- Which city sold the most products?
- What Products are most often sold together? Self Study Assignment: Perform advanced Data Manipulation operations

```
import numpy as np
import pandas as pd
```

```
all_data=pd.read_csv("/content/all_data.csv")
```

```
all_data.head()
```

1 to 5 of 5 entries Filter 📄 ?

index	Order ID	Product	Quantity Ordered	Price Each	Order Date	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	600.0	04-12-2019 14:38	669 Spruce St, Los Angeles, CA 90001
2	176560.0	Wired Headphones	1.0	11.99	04-12-2019 14:38	669 Spruce St, Los Angeles, CA 90001
3	176561.0	Wired Headphones	1.0	11.99	05/30/19 08:27	333 8th St, Los Angeles, CA 90004

## Clean up the data!

## ▼ Drop rows of NAN

```
# Find NAN
nan_df = all_data[all_data.isna().any(axis=1)]
display(nan_df.head())
```

```
all_data = all_data.dropna(how='all')
all_data.head()
```

None  
Like what you see? Visit the [data table notebook](#) to learn more about interactive tables.

	Order ID	Product	Quantity Ordered	Price Each	Order Date	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 14:38	669 Spruce St, Los Angeles, CA 90001
2	176560.0	Wired Headphones	1.0	11.99	04-12-2019 14:38	669 Spruce St, Los Angeles, CA 90001

## ▼ Get rid of text in order date column

```
all_data = all_data[all_data['Order Date'].str[0:2]!='0r']
```

▼ Make columns correct type

```
all_data['Quantity Ordered'] = pd.to_numeric(all_data['Quantity Ordered'])
all_data['Price Each'] = pd.to_numeric(all_data['Price Each'])
```

▼ Augment data with additional columns

Add month column

```
all_data['Month'] = all_data['Order Date'].str[0:2]
all_data['Month'] = all_data['Month'].astype('int32')
all_data.head()
```

1 to 5 of 5 entries 

Filter

?

index	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month
0	176559.0	Bose SoundSport Headphones	1.0	99.99	04-07-2019 22:30	682 Chestnut St, Boston, MA 02215	4
1	176560.0	Google Phone	1.0	600.0	04-12-2019 14:38	669 Spruce St, Los Angeles, CA 90001	4
2	176560.0	Wired Headphones	1.0	11.99	04-12-2019 14:38	669 Spruce St, Los Angeles, CA 90001	4
3	176561.0	Wired Headphones	1.0	11.99	05/30/19 9:27	333 8th St, Los Angeles, CA 90001	5

▼ Add month column (alternative method)

```
all_data['Month 2'] = pd.to_datetime(all_data['Order Date']).dt.month
all_data.head()
```

1 to 5 of 5 entries 

Filter

?

index	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	Month 2
0	176559.0	Bose SoundSport Headphones	1.0	99.99	04-07-2019 22:30	682 Chestnut St, Boston, MA 02215	4	4
1	176560.0	Google Phone	1.0	600.0	04-12-2019 14:38	669 Spruce St, Los Angeles, CA 90001	4	4
2	176560.0	Wired Headphones	1.0	11.99	04-12-2019 14:38	669 Spruce St, Los Angeles, CA 90001	4	4
3	176561.0	Wired Headphones	1.0	11.99	05/30/19 9:27	333 8th St, Los Angeles, CA 90001	5	5
4	176562.0	USB-C Charging Cable	1.0	11.95	04/29/19 13:03	381 Wilson St, San Francisco, CA 94016	4	4

Show 

25

 per page  
Like what you see? Visit the [data table notebook](#) to learn more about interactive tables.

▼ Add city column

```
def get_city(address):
    return address.split(",")[1].strip(" ")

def get_state(address):
    return address.split(",")[2].split(" ")[1]

all_data['City'] = all_data['Purchase Address'].apply(lambda x: f"{get_city(x)} ({get_state(x)})")
all_data.head()
```

1 to 5 of 5 entries 

Filter

?

index	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address	Month	City
0	176559.0	Bose SoundSport Headphones	1.0	99.99	04-07-2019 22:30	682 Chestnut St, Boston, MA 02215	4	Boston, MA
1	176560.0	Google Phone	1.0	600.0	04-12-2019 14:38	669 Spruce St, Los Angeles, CA 90001	4	Los Angeles, CA
2	176560.0	Wired Headphones	1.0	11.99	04-12-2019 14:38	669 Spruce St, Los Angeles, CA 90001	4	Los Angeles, CA

▼ Data Exploration!



Question 1: What was the best month for sales? How much was earned that month?

```
all_data['Sales'] = all_data['Quantity Ordered'].astype('int') * all_data['Price Each'].astype('float')
```

```
all_data.groupby(['Month']).sum()
```

<ipython-input-11-dce0a735c05d>:1: FutureWarning: The default value of numeric\_only in DataFrameGroupBy.sum is deprecated. In a future version, only numerical data will be allowed.

```
all_data.groupby(['Month']).sum()
```

1 to 8 of 8 entries   

Month	Order ID	Quantity Ordered	Price Each	Month 2	Sales
4	7335546.0	123.0	885.8	160	1210.76
5	353124.0	2.0	111.97999999999999	10	111.97999999999999
6	184076.0	1.0	14.95	6	14.95
8	726962.0	9.0	23.92	32	50.83
9	2378802.0	17.0	591.4399999999999	90	616.62
10	550924.0	11.0	10.67	30	39.69
11	740314.0	19.0	13.66	44	65.31
12	550635.0	17.0	8.97	36	50.830000000000005

Show 25 per page

Like what you see? Visit the [data table notebook](#) to learn more about interactive tables.

## Question 2: What city sold the most product?

```
city_max=all_data.groupby(['City']).sum()
print(max(city_max))
```

Sales

<ipython-input-12-79b556d70b46>:1: FutureWarning: The default value of numeric\_only in DataFrameGroupBy.sum is deprecated. In a future version, only numerical data will be allowed.

```
city_max=all_data.groupby(['City']).sum()
```

## Question 4: What products are most often sold together?

```
df = all_data[all_data['Order ID'].duplicated(keep=False)]
```

# Referenced: <https://stackoverflow.com/questions/27298178/concatenate-strings-from-several-rows-using-pandas-groupby>

```
df['Grouped'] = df.groupby('Order ID')['Product'].transform(lambda x: ', '.join(x))
df2 = df[['Order ID', 'Grouped']].drop_duplicates()
print(df['Grouped'])
```

```
1    Google Phone,Wired Headphones
2    Google Phone,Wired Headphones
```

Name: Grouped, dtype: object

<ipython-input-13-9a93a24e3a06>:4: 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-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))
```

```
from itertools import combinations
from collections import Counter
```

```
count = Counter()
```

```
for row in df2['Grouped']:
    row_list = row.split(',')
    count.update(Counter(combinations(row_list, 2)))
```

```
for key,value in count.most_common(10):
    print(key, value)
```

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

## ▼ What product sold the most? Why do you think it sold the most?

```
product_group = all_data.groupby('Product')
quantity_ordered = product_group.sum()['Quantity Ordered']
```

<ipython-input-15-4815a60ac30b>:2: FutureWarning: The default value of numeric\_only in DataFrameGroupBy.sum is deprecated. In a future \n quantity\_ordered = product\_group.sum()['Quantity Ordered']

```
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
```

```
prices = all_data.groupby('Product').mean()['Price Each']
```

<ipython-input-17-225049d1ed32>:1: FutureWarning: The default value of numeric\_only in DataFrameGroupBy.mean is deprecated. In a future \n prices = all\_data.groupby('Product').mean()['Price Each']

```
print(prices)
```

```
Product
AA Batteries (4-pack)      3.84
AAA Batteries (4-pack)     2.99
Apple AirPods Headphones  150.00
Bose SoundSport Headphones 99.99
Google Phone              600.00
Lightning Charging Cable  14.95
USB-C Charging Cable      11.95
Wired Headphones          11.99
Name: Price Each, dtype: float64
```