In [1]:
```
# #task 2 - Superstore_USA
# 1 . load this data in sql and in pandas with a relation in sql
# 2 . while loading this data you dont have to create a table manually you can use any automated
# 3 . Find out how return that we ahve recived and with a product id
# 4 . try  to join order and return data both in sql and pandas
# 5 . Try to find out how many unique customer that we have
# 6 . try to find out in how many regions we are selling a product and who is a manager for a res
# 7 . find out how many different differnet shipement mode that we have and what is a percentage
# 8 . Create a new coulmn and try to find our a diffrence between order date and shipment date
# 9 . base on question number 8 find out for which order id we have shipment duration more than 1
# 10 . Try to find out a list of a returned order which sihpment duration was more then 15 days a
# 11 . Gorup by region and find out which region is more profitable
# 12 . Try to find out overalll in which country we are giving more didscount
# 13 . Give me a list of unique postal code
# 14 . which customer segement is more profitalble find it out .
# 15 . try to find out the 10th most loss making product catagory .
# 16 . Try to find out 10 top  product with highest margins

# submit your git link for the same

# you have to submit this task by next Friday 5th 11PM IST  to respective mail id
# sudhanshu@ineuron.ai , sunny.savita@ineuron.ai
```

In [2]:
```
# 1 . load this data in sql and in pandas with a relation in sql
```

In [31]:
```
import mysql.connector as conn
from mysql.connector import Error
```

In [45]:
```
try:
    mydb = conn.connect(host='localhost',user='root',password='123456')
    print(mydb)

    if mydb.is_connected():
        db_Info = mydb.get_server_info()
        print("Connected to MySQL Server version ", db_Info)
        cursor = mydb.cursor()

except Error as e:
    print("Error while connecting to MySQL", e)
```

```
<mysql.connector.connection.MySQLConnection object at 0x000002E4D9A4C730>
Connected to MySQL Server version  8.0.26
```

In [5]:
```
import csvkit
```

In [6]:
```
!csvsql -i mysql Users.csv
```

```
CREATE TABLE `Users` (
        `Region` VARCHAR(7) NOT NULL,
        `Manager` VARCHAR(7) NOT NULL
);

C:\Users\user\anaconda3\lib\threading.py:874: ResourceWarning: unclosed file <_io.BufferedWriter nam
e=4>
C:\Users\user\anaconda3\lib\threading.py:874: ResourceWarning: unclosed file <_io.BufferedReader nam
e=5>
C:\Users\user\anaconda3\lib\threading.py:874: ResourceWarning: unclosed file <_io.BufferedReader nam
e=6>
sys:1: ResourceWarning: unclosed socket <zmq.Socket(zmq.PUSH) at 0x2e4d9927880>
```

In [18]:
```
!csvsql -i mysql Orders.csv
```

```
CREATE TABLE `Orders` (

C:\Users\user\anaconda3\lib\site-packages\agate\table\from_csv.py:74: RuntimeWarning: Error sniffing
CSV dialect: Could not determine delimiter
sys:1: ResourceWarning: unclosed socket <zmq.Socket(zmq.PUSH) at 0x2e4d993b040>


        `Row ID` DECIMAL(38, 0) NOT NULL,
        `Order Priority` VARCHAR(13) NOT NULL,
        `Discount` DECIMAL(38, 2) NOT NULL,
        `Unit Price` DECIMAL(38, 2) NOT NULL,
        `Shipping Cost` DECIMAL(38, 2) NOT NULL,
        `Customer ID` DECIMAL(38, 0) NOT NULL,
        `Customer Name` VARCHAR(28) NOT NULL,
        `Ship Mode` VARCHAR(14) NOT NULL,
        `Customer Segment` VARCHAR(14) NOT NULL,
        `Product Category` VARCHAR(15) NOT NULL,
        `Product Sub-Category` VARCHAR(30) NOT NULL,
        `Product Container` VARCHAR(10) NOT NULL,
        `Product Name` VARCHAR(98) NOT NULL,
        `Product Base Margin` DECIMAL(38, 2),
        `Region` VARCHAR(7) NOT NULL,
        `State or Province` VARCHAR(20) NOT NULL,
        `City` VARCHAR(19) NOT NULL,
        `Postal Code` DECIMAL(38, 0) NOT NULL,
        `Order Date` DATE NOT NULL,
        `Ship Date` DATE NOT NULL,
        `Profit` DECIMAL(38, 7) NOT NULL,
        `Quantity ordered new` DECIMAL(38, 0) NOT NULL,
        `Sales` DECIMAL(38, 2) NOT NULL,
        `Order ID` DECIMAL(38, 0) NOT NULL
);

C:\Users\user\anaconda3\lib\threading.py:874: ResourceWarning: unclosed file <_io.BufferedWriter nam
e=4>
C:\Users\user\anaconda3\lib\threading.py:874: ResourceWarning: unclosed file <_io.BufferedReader nam
e=5>
C:\Users\user\anaconda3\lib\threading.py:874: ResourceWarning: unclosed file <_io.BufferedReader nam
e=6>
sys:1: ResourceWarning: unclosed socket <zmq.Socket(zmq.PUSH) at 0x2e4d993b100>
```

In [13]:
```
!csvsql -i mysql Returns.csv
```

```
CREATE TABLE `Returns` (

C:\Users\user\anaconda3\lib\threading.py:874: ResourceWarning: unclosed file <_io.BufferedWriter nam
e=4>
C:\Users\user\anaconda3\lib\threading.py:874: ResourceWarning: unclosed file <_io.BufferedReader nam
e=5>


        `Order ID` DECIMAL(38, 0) NOT NULL,
        `Status` VARCHAR(8) NOT NULL
);

C:\Users\user\anaconda3\lib\threading.py:874: ResourceWarning: unclosed file <_io.BufferedReader nam
e=6>
sys:1: ResourceWarning: unclosed socket <zmq.Socket(zmq.PUSH) at 0x2e4d9927e80>
```

In [41]:
```
try:
    cursor.execute("Create database Customers_USA")
    print("database created successfully")
except conn.Error as error:
    print("Failed to create Database in MySQL: {}".format(error))
finally:
    cursor.execute("Use Customers_USA")
```

```
Failed to create Database in MySQL: 1007 (HY000): Can't create database 'customers_usa'; database ex
ists
```

In [42]:
```
t = """CREATE TABLE `Orders` (
`Row ID` DECIMAL(38, 0) NOT NULL,
`Order Priority` VARCHAR(13) NOT NULL,
`Discount` DECIMAL(38, 2) NOT NULL,
`Unit Price` DECIMAL(38, 2) NOT NULL,
`Shipping Cost` DECIMAL(38, 2) NOT NULL,
`Customer ID` DECIMAL(38, 0) NOT NULL,
`Customer Name` VARCHAR(28) NOT NULL,
`Ship Mode` VARCHAR(14) NOT NULL,
`Customer Segment` VARCHAR(14) NOT NULL,
`Product Category` VARCHAR(15) NOT NULL,
`Product Sub-Category` VARCHAR(30) NOT NULL,
`Product Container` VARCHAR(10) NOT NULL,
`Product Name` VARCHAR(98) NOT NULL,
`Product Base Margin` INT,
`Region` VARCHAR(7) NOT NULL,
`State or Province` VARCHAR(20) NOT NULL,
`City` VARCHAR(19) NOT NULL,
`Postal Code` DECIMAL(38, 0) NOT NULL,
`Order Date` DATE NOT NULL,
`Ship Date` DATE NOT NULL,
`Profit` DECIMAL(38, 7) NOT NULL,
`Quantity ordered new` DECIMAL(38, 0) NOT NULL,
`Sales` DECIMAL(38, 2) NOT NULL,
`Order ID` DECIMAL(38, 0) NOT NULL
);"""
```

In [43]:
```
try:
    cursor.execute(t)
    print("Table created successfully")
except conn.Error as error:
    print("Failed to create table in MySQL: {}".format(error))
finally:
    if mydb.is_connected():
        cursor.close()
        mydb.close()
        print("MySQL connection is closed")
```

```
Table created successfully
MySQL connection is closed
```

In [49]:
```
try:
    load =  """LOAD DATA INFILE "D:/18_iNeuron/00_S_Assignments/task/Orders.csv"
            INTO table Customers_USA.Orders
            FIELDS TERMINATED BY ','
            ENCLOSED BY '"'
            LINES TERMINATED BY '\r\n'
            IGNORE 1 LINES;"""
    cursor.execute(load)
except conn.Error as error:
    print("Failed to create table in MySQL: {}".format(error))
```

In [1]:
```
import pandas as pd
```

In [260]:
```
order_df = pd.read_csv("Orders.csv")
returns_df = pd.read_csv("Returns.csv")
users_df = pd.read_csv("Users.csv")
```

In [9]:
```
pd.set_option('display.max_columns', None)
```

In [4]:
```
#3. Find out how return that we ahve recived and with a product id
```

In [6]:
```
1 returns_df.head()
```

Out[6]:

	Order ID	Status
0	65	Returned
1	612	Returned
2	614	Returned
3	678	Returned
4	710	Returned

In [78]:
```
1 order_df.head()
```

Out[78]:

ıer ıne	Ship Mode	Customer Segment	Product Category	Product Sub-Category	Product Container	Product Name	Product Base Margin	Region	State or Province	City	Postal Code	Or D
ice ıer	Regular Air	Corporate	Office Supplies	Labels	Small Box	Avery 49	0.36	Central	Illinois	Addison	60101	20 05
nie ıter	Express Air	Corporate	Office Supplies	Pens & Art Supplies	Wrap Bag	SANFORD Liquid Accent™ Tank-Style Highlighters	0.54	West	Washington	Anacortes	98221	20 07
nie ıter	Express Air	Corporate	Office Supplies	Paper	Small Box	Xerox 1968	0.37	West	Washington	Anacortes	98221	20 07
nie ıter	Regular Air	Corporate	Office Supplies	Scissors, Rulers and Trimmers	Small Pack	Acme® Preferred Stainless Steel Scissors	0.56	West	Washington	Anacortes	98221	20 07
nie ıter	Express Air	Corporate	Technology	Telephones and Communication	Small Box	V70	0.59	West	Washington	Anacortes	98221	20 07

In [11]:
```
1 order_df.merge(returns_df).head()
```

Out[11]:

ct y	Product Sub-Category	Product Container	Product Name	Product Base Margin	Region	State or Province	City	Postal Code	Order Date	Ship Date	Profit	Quantity ordered new	Sales
e s	Pens & Art Supplies	Wrap Bag	Dixon My First Ticonderoga Pencil, #2	0.56	East	New York	New York City	10012	2011-04-20	2011-04-24	-6.8200	9	54.79
e s	Labels	Small Box	Avery 493	0.36	West	Washington	Seattle	98103	2010-04-04	2010-04-06	112.0600	47	228.46
e s	Paper	Wrap Bag	EcoTones® Memo Sheets	0.37	West	Washington	Seattle	98103	2010-04-04	2010-04-06	16.7900	19	77.61
e s	Pens & Art Supplies	Wrap Bag	Newell 35	0.56	West	Washington	Seattle	98103	2013-08-16	2013-08-18	-89.0600	52	190.52
e s	Pens & Art Supplies	Small Pack	Staples SlimLine Pencil Sharpener	0.60	South	North Carolina	Charlotte	28204	2013-08-12	2013-08-12	-81.9413	76	912.06

In [14]:
```
1 #3. Find out how return that we ahve recived and with a product id
```

In [12]:
```
1 df1 = order_df.merge(returns_df)[['Product Name','Order ID','Status']]
```

In [57]:
```
1 df1
```

Out[57]:

	Product Name	Order ID	Status
0	Dixon My First Ticonderoga Pencil, #2	9895	Returned
1	Avery 493	13959	Returned
2	EcoTones® Memo Sheets	13959	Returned
3	Newell 35	36038	Returned
4	Staples SlimLine Pencil Sharpener	39490	Returned
...	...	...	...
93	Snap-A-Way® Black Print Carbonless Ruled Speed...	7107	Returned
94	Recycled Steel Personal File for Standard File...	7107	Returned
95	Xerox 1980	42823	Returned
96	TDK 4.7GB DVD+RW	13638	Returned
97	SAFCO Folding Chair Trolley	47109	Returned

98 rows × 3 columns

In [33]:
```
1 df1.groupby(['Product Name'])['Status'].count()
```

Out[33]:
```
Product Name
#10 White Business Envelopes,4 1/8 x 9 1/2            1
12 Colored Short Pencils                              1
232                                                   1
600 Series Flip                                       1
6160                                                  1
                                                     ..
Xerox 197                                             2
Xerox 1980                                            1
Xerox 1983                                            2
Xerox 210                                             1
Zoom V.92 V.44 PCI Internal Controllerless FaxModem   1
Name: Status, Length: 94, dtype: int64
```

In [25]:
```
1 df1[df1['Status'] == "Returned"]["Product Name"]
```

Out[25]:
```
0                Dixon My First Ticonderoga Pencil, #2
1                                         Avery 493
2                                EcoTones® Memo Sheets
3                                         Newell 35
4                    Staples SlimLine Pencil Sharpener
                            ...
93    Snap-A-Way® Black Print Carbonless Ruled Speed...
94    Recycled Steel Personal File for Standard File...
95                                        Xerox 1980
96                                    TDK 4.7GB DVD+RW
97                          SAFCO Folding Chair Trolley
Name: Product Name, Length: 98, dtype: object
```

In [38]:
```
1 df1.groupby(['Product Name'])['Status'].value_counts()
```

Out[38]:
```
Product Name                                          Status
#10 White Business Envelopes,4 1/8 x 9 1/2            Returned    1
12 Colored Short Pencils                              Returned    1
232                                                   Returned    1
600 Series Flip                                       Returned    1
6160                                                  Returned    1
                                                                 ..
Xerox 197                                             Returned    2
Xerox 1980                                            Returned    1
Xerox 1983                                            Returned    2
Xerox 210                                             Returned    1
Zoom V.92 V.44 PCI Internal Controllerless FaxModem   Returned    1
Name: Status, Length: 94, dtype: int64
```

In [39]:
```
type(df1.groupby(['Product Name'])['Status'].value_counts())
```

Out[39]:
```
pandas.core.series.Series
```

In [40]:
```
df2 = pd.DataFrame(df1.groupby(['Product Name'])['Status'].value_counts())
```

In [51]:
```
df2
```

Out[51]:

		Status
Product Name	Status	
#10 White Business Envelopes,4 1/8 x 9 1/2	Returned	1
12 Colored Short Pencils	Returned	1
232	Returned	1
600 Series Flip	Returned	1
6160	Returned	1
...	...	...
Xerox 197	Returned	2
Xerox 1980	Returned	1
Xerox 1983	Returned	2
Xerox 210	Returned	1
Zoom V.92 V.44 PCI Internal Controllerless FaxModem	Returned	1

94 rows × 1 columns

In [58]:
```
type(df2)
```

Out[58]:
```
pandas.core.frame.DataFrame
```

In [59]:
```
# 4 . try  to join order and return data both in sql and pandas
```

In [ ]:
```
# order_df = pd.read_csv("Orders.csv")
# returns_df = pd.read_csv("Returns.csv")
# users_df = pd.read_csv("Users.csv")
```

In [64]:
```
order_df.join(returns_df, on = "Order ID", how="left",lsuffix='_left', rsuffix='_right').head()
```

Out[64]:

Product Container	Product Name	Product Base Margin	Region	State or Province	City	Postal Code	Order Date	Ship Date	Profit	Quantity ordered new	Sales	Order ID_left	O ID_
Small Box	Avery 49	0.36	Central	Illinois	Addison	60101	2012-05-28	2012-05-30	1.3200	2	5.90	88525	
Wrap Bag	SANFORD Liquid Accent™ Tank-Style Highlighters	0.54	West	Washington	Anacortes	98221	2010-07-07	2010-07-08	4.5600	4	13.01	88522	
Small Box	Xerox 1968	0.37	West	Washington	Anacortes	98221	2011-07-27	2011-07-28	-47.6400	7	49.92	88523	
Small Pack	Acme® Preferred Stainless Steel Scissors	0.56	West	Washington	Anacortes	98221	2011-07-27	2011-07-28	-30.5100	7	41.64	88523	
Small Box	V70	0.59	West	Washington	Anacortes	98221	2011-07-27	2011-07-27	998.2023	8	1446.67	88523	

In [115]:
```
df_C= order_df.merge(returns_df)
```

In [116]:
```
df_C.head(2)
```

Out[116]:

	Row ID	Order Priority	Discount	Unit Price	Shipping Cost	Customer ID	Customer Name	Ship Mode	Customer Segment	Product Category	Product Sub-Category	Product Container	Pro N
)	1359	Low	0.05	5.85	2.27	21	Tony Wilkins Winters	Regular Air	Small Business	Office Supplies	Pens & Art Supplies	Wrap Bag	Dixo Ticonde Penci
I	1950	Medium	0.01	4.91	0.50	117	Linda Weiss	Regular Air	Home Office	Office Supplies	Labels	Small Box	Avery

In [119]:
```
returns_df.info
```

Out[119]:
```
<bound method DataFrame.info of       Order ID    Status
0           65  Returned
1          612  Returned
2          614  Returned
3          678  Returned
4          710  Returned
...        ...       ...
1629    182681  Returned
1630    182683  Returned
1631    182750  Returned
1632    182781  Returned
1633    182906  Returned

[1634 rows x 2 columns]>
```

In [ ]:
```
# 5 . Try to find out how many unique customer that we have
```

In [76]:
```
df_C['Customer ID'].value_counts()
```

Out[76]:
```
1193    27
699     26
2107    22
2491    22
2882    21
        ..
2536     1
2538     1
1172     1
1168     1
2        1
Name: Customer ID, Length: 2703, dtype: int64
```

In [77]:
```
pd.DataFrame(df_C['Customer ID'].value_counts())
```

Out[77]:

	Customer ID
1193	27
699	26
2107	22
2491	22
2882	21
...	...
2536	1
2538	1
1172	1
1168	1
2	1

2703 rows × 1 columns

In [ ]:
```
# 6 . try to find out in how many regions we are selling a product and who is a manager for a res
```

In [79]:
```
users_df
```

Out[79]:

	Region	Manager
0	Central	Chris
1	East	Erin
2	South	Sam
3	West	William

In [122]:
```
df_C1 = order_df.merge(users_df, on='Region')
```

In [123]:
```
df_C1.head(2)
```

Out[123]:

	Row ID	Order Priority	Discount	Unit Price	Shipping Cost	Customer ID	Customer Name	Ship Mode	Customer Segment	Product Category	Product Sub-Category	Product Container
0	18606	Not Specified	0.01	2.88	0.50	2	Janice Fletcher	Regular Air	Corporate	Office Supplies	Labels	Small Box
1	24844	Medium	0.09	78.69	19.99	14	Gwendolyn F Tyson	Regular Air	Small Business	Furniture	Office Furnishings	Small Box

In [124]:
```
df_C1.groupby('Region')['Manager'].value_counts()
```

Out[124]:
```
Region   Manager
Central  Chris      2899
East     Erin       2289
South    Sam        1954
West     William    2284
Name: Manager, dtype: int64
```

In [125]:
```
df_C1.groupby('Region')['Manager']
```

Out[125]:
```
<pandas.core.groupby.generic.SeriesGroupBy object at 0x000001F4F1EFAC70>
```

In [127]:
```
df_C1.groupby('Region')['Product Name'].value_counts()
```

Out[127]:
```
Region   Product Name
Central  Fiskars® Softgrip Scissors
11
         80 Minute CD-R Spindle, 100/Pack - Staples
10
         Memorex 4.7GB DVD+RW, 3/Pack
10
         Office Star - Mid Back Dual function Ergonomic High Back Chair with 2-Way Adjustable Arms
10
         Verbatim DVD-RAM, 5.2GB, Rewritable, Type 1, DS
10

..
West     Zoom V.92 USB External Faxmodem
1
         g520
1
         i270
1
         iDEN i550
1
         iDENi80s
1
Name: Product Name, Length: 3976, dtype: int64
```

In [ ]:
```
# 7 . find out how many different differnet shipement mode that we have and what is a percentage
```

In [132]:
```
order_df.head(3)
```

Out[132]:

Customer ID	Customer Name	Ship Mode	Customer Segment	Product Category	Product Sub-Category	Product Container	Product Name	Product Base Margin	Region	State or Province	City
2	Janice Fletcher	Regular Air	Corporate	Office Supplies	Labels	Small Box	Avery 49	0.36	Central	Illinois	Addison
3	Bonnie Potter	Express Air	Corporate	Office Supplies	Pens & Art Supplies	Wrap Bag	SANFORD Liquid Accent™ Tank-Style Highlighters	0.54	West	Washington	Anacortes
3	Bonnie Potter	Express Air	Corporate	Office Supplies	Paper	Small Box	Xerox 1968	0.37	West	Washington	Anacortes

In [134]:
```
order_df['Ship Mode'].value_counts()
```

Out[134]:
```
Regular Air       7036
Delivery Truck    1283
Express Air       1107
Name: Ship Mode, dtype: int64
```

In [135]:
```
type(order_df['Ship Mode'].value_counts())
```

Out[135]:
```
pandas.core.series.Series
```

In [139]:
```
order_df['Ship Mode'].count()
```

Out[139]:
```
9426
```

In [141]:
```
df_P = pd.DataFrame(order_df['Ship Mode'].value_counts())
```

In [142]:
```
df_P
```

Out[142]:

	Ship Mode
Regular Air	7036
Delivery Truck	1283
Express Air	1107

In [160]:
```
m = order_df['Ship Mode'].count()
```

In [161]:
```
m
```

Out[161]:
```
9426
```

In [157]:
```
l = df_P['Ship Mode'].tolist()
```

In [158]:
```
l
```

Out[158]:
```
[7036, 1283, 1107]
```

In [162]:
```
for i in l:
    v = (l/m)*100
```

In [163]:
```
v
```

Out[163]:
```
array([74.64460004, 13.61128793, 11.74411203])
```

In [165]:
```
df_P['Percentage'] = v
```

In [166]:
```
1 df_P.head()
```

Out[166]:

	Ship Mode	Percentage
Regular Air	7036	74.644600
Delivery Truck	1283	13.611288
Express Air	1107	11.744112

In [ ]:
```
1 # 8 . Create a new coulmn and try to find our a diffrence between order date and shipment date
```

In [167]:
```
1 order_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9426 entries, 0 to 9425
Data columns (total 24 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Row ID                9426 non-null   int64
 1   Order Priority        9426 non-null   object
 2   Discount              9426 non-null   float64
 3   Unit Price            9426 non-null   float64
 4   Shipping Cost         9426 non-null   float64
 5   Customer ID           9426 non-null   int64
 6   Customer Name         9426 non-null   object
 7   Ship Mode             9426 non-null   object
 8   Customer Segment      9426 non-null   object
 9   Product Category      9426 non-null   object
 10  Product Sub-Category  9426 non-null   object
 11  Product Container     9426 non-null   object
 12  Product Name          9426 non-null   object
 13  Product Base Margin   9354 non-null   float64
 14  Region                9426 non-null   object
 15  State or Province     9426 non-null   object
 16  City                  9426 non-null   object
 17  Postal Code           9426 non-null   int64
 18  Order Date            9426 non-null   object
 19  Ship Date             9426 non-null   object
 20  Profit                9426 non-null   float64
 21  Quantity ordered new  9426 non-null   int64
 22  Sales                 9426 non-null   float64
 23  Order ID              9426 non-null   int64
dtypes: float64(6), int64(5), object(13)
memory usage: 1.7+ MB
```

In [264]:
```
1 order_df["Order Date_For"] = pd.to_datetime(order_df['Order Date'])
2 order_df["Ship Date_For"] = pd.to_datetime(order_df['Ship Date'])
```

In [265]:
```
order_df.head()
```

Out[265]:

	Row ID	Order Priority	Discount	Unit Price	Shipping Cost	Customer ID	Customer Name	Ship Mode	Customer Segment	Product Category	Product Sub-Category	Proc Conta
0	18606	Not Specified	0.01	2.88	0.50	2	Janice Fletcher	Regular Air	Corporate	Office Supplies	Labels	Small
1	20847	High	0.01	2.84	0.93	3	Bonnie Potter	Express Air	Corporate	Office Supplies	Pens & Art Supplies	Wrap
2	23086	Not Specified	0.03	6.68	6.15	3	Bonnie Potter	Express Air	Corporate	Office Supplies	Paper	Small
3	23087	Not Specified	0.01	5.68	3.60	3	Bonnie Potter	Regular Air	Corporate	Office Supplies	Scissors, Rulers and Trimmers	S F
4	23088	Not Specified	0.00	205.99	2.50	3	Bonnie Potter	Express Air	Corporate	Technology	Telephones and Communication	Small

In [266]:
```
order_df['Date_Difference'] = order_df['Ship Date_For'].sub(order_df['Order Date_For'], axis=0)
```

In [267]:
```
order_df.head()
```

Out[267]:

roduct Base Margin	Region	State or Province	City	Postal Code	Order Date	Ship Date	Profit	Quantity ordered new	Sales	Order ID	Order Date_For	Ship Date_For	Date_Dif
0.36	Central	Illinois	Addison	60101	2012-05-28	2012-05-30	1.3200	2	5.90	88525	2012-05-28	2012-05-30	
0.54	West	Washington	Anacortes	98221	2010-07-07	2010-07-08	4.5600	4	13.01	88522	2010-07-07	2010-07-08	
0.37	West	Washington	Anacortes	98221	2011-07-27	2011-07-28	-47.6400	7	49.92	88523	2011-07-27	2011-07-28	
0.56	West	Washington	Anacortes	98221	2011-07-27	2011-07-28	-30.5100	7	41.64	88523	2011-07-27	2011-07-28	
0.59	West	Washington	Anacortes	98221	2011-07-27	2011-07-27	998.2023	8	1446.67	88523	2011-07-27	2011-07-27	

In [ ]:
```
# 9 . base on question number 8 find out for which order id we have shipment duration more than 1
```

In [268]:
```
order_df['Date_Difference_int'] = pd.DataFrame(order_df['Date_Difference'].dt.days)
```

In [269]:
```
order_df.head(2)
```

Out[269]:

State or Province	City	Postal Code	Order Date	Ship Date	Profit	Quantity ordered new	Sales	Order ID	Order Date_For	Ship Date_For	Date_Difference	Date_Differen
Illinois	Addison	60101	2012-05-28	2012-05-30	1.32	2	5.90	88525	2012-05-28	2012-05-30	2 days	
Washington	Anacortes	98221	2010-07-07	2010-07-08	4.56	4	13.01	88522	2010-07-07	2010-07-08	1 days	

In [270]:
```
1 order_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9426 entries, 0 to 9425
Data columns (total 28 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Row ID                9426 non-null   int64
 1   Order Priority        9426 non-null   object
 2   Discount              9426 non-null   float64
 3   Unit Price            9426 non-null   float64
 4   Shipping Cost         9426 non-null   float64
 5   Customer ID           9426 non-null   int64
 6   Customer Name         9426 non-null   object
 7   Ship Mode             9426 non-null   object
 8   Customer Segment      9426 non-null   object
 9   Product Category      9426 non-null   object
 10  Product Sub-Category  9426 non-null   object
 11  Product Container     9426 non-null   object
 12  Product Name          9426 non-null   object
 13  Product Base Margin   9354 non-null   float64
 14  Region                9426 non-null   object
 15  State or Province     9426 non-null   object
 16  City                  9426 non-null   object
 17  Postal Code           9426 non-null   int64
 18  Order Date            9426 non-null   object
 19  Ship Date             9426 non-null   object
 20  Profit                9426 non-null   float64
 21  Quantity ordered new  9426 non-null   int64
 22  Sales                 9426 non-null   float64
 23  Order ID              9426 non-null   int64
 24  Order Date_For        9426 non-null   datetime64[ns]
 25  Ship Date_For         9426 non-null   datetime64[ns]
 26  Date_Difference       9426 non-null   timedelta64[ns]
 27  Date_Difference_int   9426 non-null   int64
dtypes: datetime64[ns](2), float64(6), int64(6), object(13), timedelta64[ns](1)
memory usage: 2.0+ MB
```

In [272]:
```
order_df.loc[order_df['Date_Difference_int'] > 10][['Order ID','Date_Difference_int']]
```

Out[272]:

	Order ID	Date_Difference_int
643	87215	84
1548	86318	11
1549	86318	15
1678	87957	17
1679	87957	11
1680	87957	28
1697	19556	17
1698	19556	11
1699	19556	28
2515	86177	92
5548	88223	14
5673	88352	17
5859	87572	22
5881	91294	24
8607	86434	18
8609	86436	17
8610	86436	19
8973	87300	19
8982	19841	27
8983	19841	31
8993	19841	19
8996	87300	27
8997	87300	31

In [280]:
```
S10 = order_df[order_df['Date_Difference_int'] > 10][['Order ID','Date_Difference_int']]
```

In [281]:
```
df_S10 = pd.DataFrame(S10)
```

In [276]:
```
pd.set_option('display.max_rows', 25)
```

In [282]:
```
df_S10.head(25)
```

Out[282]:

	Order ID	Date_Difference_int
643	87215	84
1548	86318	11
1549	86318	15
1678	87957	17
1679	87957	11
1680	87957	28
1697	19556	17
1698	19556	11
1699	19556	28
2515	86177	92
5548	88223	14
5673	88352	17
5859	87572	22
5881	91294	24
8607	86434	18
8609	86436	17
8610	86436	19
8973	87300	19
8982	19841	27
8983	19841	31
8993	19841	19
8996	87300	27
8997	87300	31

In [197]:
```
# 10 . Try to find out a list of a returned order which sihpment duration was more then 15 days a
```

In [283]:
```
order_df = returns_df.merge(order_df).head()
```

In [284]:
```
1 order_df.head()
```

Out[284]:

duct iner	Product Name	Product Base Margin	Region	State or Province	City	Postal Code	Order Date	Ship Date	Profit	Quantity ordered new	Sales	Order Date_For	Date
Box	Xerox 1928	0.40	East	Massachusetts	Boston	2113	2013-10-20	2013-10-22	-166.290	105	552.03	2013-10-20	201
Box	6190	0.55	East	Massachusetts	Boston	2113	2013-10-20	2013-10-22	881.676	90	5297.63	2013-10-20	201
mall ?ack	80 Minute Slim Jewel Case CD-R , 10/Pack - Sta...	0.52	West	California	Los Angeles	90068	2011-06-17	2011-06-19	-17.010	12	100.32	2011-06-17	201
Box	#10 White Business Envelopes,4 1/8 x 9 1/2	0.38	West	California	Los Angeles	90068	2011-06-17	2011-06-21	203.010	44	678.72	2011-06-17	201
mbo)rum	Hon 2090 “Pillow Soft” Series Mid Back Swivel/...	0.78	East	New York	New York City	10177	2011-02-08	2011-02-09	-416.700	44	12296.49	2011-02-08	201

In [285]:
```
1 order_df[order_df['Date_Difference_int'] > 15]["Status"]
```

Out[285]:
```
Series([], Name: Status, dtype: object)
```

In [288]:
```
1 order_df.merge(returns_df)[['Date_Difference']].drop_duplicates()
```

Out[288]:

	Date_Difference
0	2 days
3	4 days
4	1 days

In [ ]:
```
1 # 11 . Gorup by region and find out which region is more profitable
```

In [292]:
```
1 order_df = pd.read_csv("Orders.csv")
```

In [293]:
```
1 order_df.groupby('Region')['Profit'].mean()
```

Out[293]:
```
Region
Central    179.312027
East       164.948094
South       53.327120
West       136.098710
Name: Profit, dtype: float64
```

In [ ]:
```
1 # 12 . Try to find out overalll in which country we are giving more didscount
```

In [296]:
```
1 order_df.groupby('State or Province')['Discount'].mean()
```

Out[296]:
```
State or Province
Alabama           0.046960
Arizona           0.051194
Arkansas          0.047154
California        0.051205
Colorado          0.045480
                    ...
Virginia          0.049495
Washington        0.047339
West Virginia     0.050233
Wisconsin         0.047692
Wyoming           0.054762
Name: Discount, Length: 49, dtype: float64
```

In [ ]:
```
# 13 . Give me a list of unique postal code
```

In [297]:
```
order_df['Postal Code'].unique().tolist()
```

```
8109,
7203,
2907,
55372,
11787,
13210,
59405,
59601,
59801,
68005,
10012,
92653,
92677,
92530,
92630,
90712,
93534,
90260,
97405,
97526.
```

In [ ]:
```
# 14 . which customer segement is more profitalble find it out .
```

In [303]:
```
order_df.groupby('Customer Segment')['Profit'].mean().sort_values(ascending=False).nlargest(n=1)
```

Out[303]:
```
Customer Segment
Small Business    171.903635
Name: Profit, dtype: float64
```

In [ ]:
```
# 15 . try to find out the 10th most loss making product catagory .
```

In [307]:
```
order_df.groupby('Product Sub-Category')['Profit'].mean().sort_values().nsmallest(n=10)
```

Out[307]:
```
Product Sub-Category
Tables                        -179.443222
Bookcases                      -37.421110
Rubber Bands                   -14.572936
Scissors, Rulers and Trimmers  -12.495801
Pens & Art Supplies              1.658672
Storage & Organization          13.243942
Paper                           25.642945
Labels                          54.028330
Computer Peripherals           103.921800
Office Furnishings             104.427209
Name: Profit, dtype: float64
```

In [308]:
```
order_df.groupby('Product Name')['Profit'].mean().sort_values().nsmallest(n=10)
```

Out[308]:
```
Product Name
Okidata Pacemark 4410N Wide Format Dot Matrix Printer   -6623.862973
Polycom ViewStation™ ISDN Videoconferencing Unit        -5206.631933
Epson DFX-8500 Dot Matrix Printer                       -4222.523685
Contemporary Wood/Metal Frame                           -2665.512920
Newell 332                                              -1914.808000
Situations Contoured Folding Chairs, 4/Set              -1333.594500
Epson DFX5000+ Dot Matrix Printer                       -1326.753978
Balt Solid Wood Rectangular Table                       -1296.642765
High Speed Automatic Electric Letter Opener             -1263.545000
Hewlett Packard 610 Color Digital Copier / Printer      -1255.248000
Name: Profit, dtype: float64
```

In [ ]:
```
# 16 . Try to find out 10 top  product with highest margins
```

In [323]:
```
order_df[['Product Base Margin','Product Name']].drop_duplicates().sort_values(by='Product Base M
```

Out[323]:

	Product Base Margin	Product Name
984	0.35	Accessory12
1384	0.35	Brown Kraft Recycled Envelopes
4072	0.35	Accessory32
4057	0.35	Xerox 1916
1989	0.35	Avery Binding System Hidden Tab™ Executive Sty...
245	0.35	"While you Were Out" Message Book, One Form pe...
1116	0.35	Mead 1st Gear 2" Zipper Binder, Asst. Colors
1118	0.35	Surelock™ Post Binders
1655	0.35	Presstex Flexible Ring Binders
741	0.35	Unpadded Memo Slips

In [ ]:
```
# submit your git link for the same
 # you have to submit this task by next Friday 5th 11PM IST  to respective mail id
# sudhanshu@ineuron.ai , sunny.savita@ineuron.ai
```

In [ ]:
```

```