Fittlyf_Interview_Solution

Vaishnavi_Ravikiran_Mittha_29/11/2022

BTech Third Year

Part 0: Reading the data:

- **Print all the column names and the data types in each column.**

  Source code-

  ```
  import pandas as pd
  #making data frame
  data = pd.read_csv("test_DataScience.csv")
  # printing the column name and their datatype
  DataTypeSeries= data.dtypes
  print(DataTypeSeries)
  ```

OUTPUT:

```
Year                      int64
Month                    object
Laptop/Desktop           object
Type_of_Customers?       object
Coming from              object
Place_in_India           object
Level 1                 float64
Level 2                 float64
Level 3                   int64
Level 4                   int64
        dtype: object
```

- **Print the cities of India from which the page was accessed.**

  <u>Source code:</u>

```
import pandas as pd
dt=pd.read_csv("test_DataScience.csv")
#fetching the places in india from which the page was accessed.
print(dt[["Place_in_India"]])
```

<u>OUTPUT:</u>

```
     Place_in_India
0         Bengaluru
1         Hyderabad
2          Dehradun
3            Indore
4              Pune
...             ...
2155      Bengaluru
2156      Hyderabad
2157       Dehradun
2158         Indore
2159           Pune

[2160 rows x 1 columns]
```

- Write a brief paragraph about what you think about this dataset along the lines of :
  Q.1] Which geo-location this dataset belongs to?
  <u>Answer:</u> Flipkart is a tech company first then an e-comm firm. The dataset belongs to all famous IT hubs, metro cities in India. It belongs to Bengaluru, Hyderabad, Dehradun, Indore and Pune.
  If we run this query- **data['Place_in_India'].value_counts()**
  We get the output as,

```
Bengaluru    432
Hyderabad    432
Dehradun     432
Indore       432
Pune         432
 Name: Place_in_India,
dtype: int64
```

As we can see, number of cities from India from where the page was accessed is 432 each. E-comm is a fast developing segment in India. Hence it targets such geolocations which are metro cities, much ahead of technology. The most interesting aspect of the business is that home-grown companies like Snapdeal and Flipkart are fighting out with global majors like Amazon and eBay. It is no mean feat, considering the Indian companies are still in the nascent stage. So these companies target the tech geo locations for sales and services first.

Q.2] Given that this dataset is for a website like Flipkart, what could be the possible definitions of the columns Level 1, 2, 3, 4 in the given dataset?
Answer: Level 1,2,3,4 are the dependent attributes. They depend on Type_of_customer attribute. So here there is function dependency.
Level 1 and 2 in not applicable for the existing customers. Where as it is applicable for new customers. That means this level 1 and 2 could be steps involved to order a product. It could be subscription, premium, sign in option. Or it could be rank.
Level 3 and 4 is applicable for all. That means it is allowed for both existing customers and new customer.

Part 1: Data cleaning

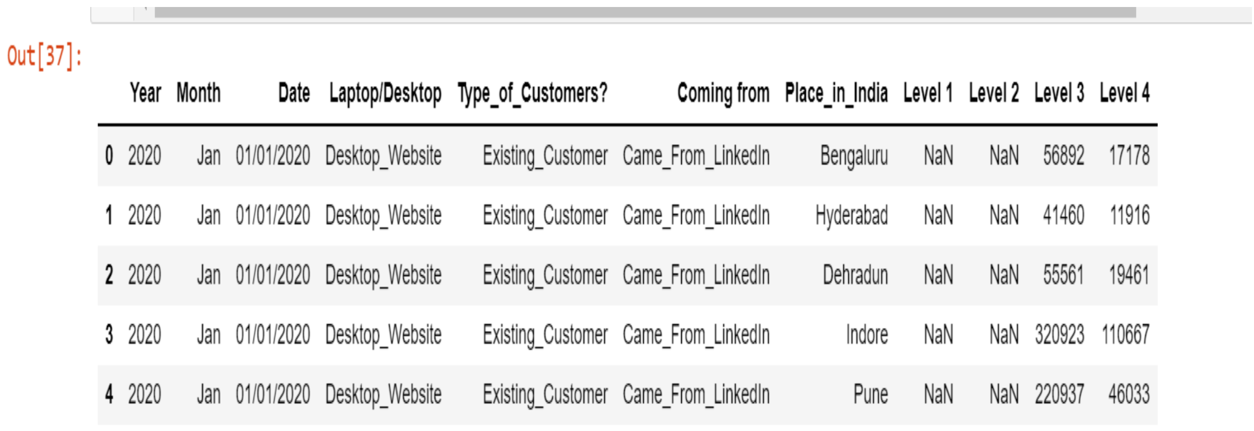
Write a function called data_cleaning() which, when called, would perform the following :

1] Create a new column, called 'Month_Year', using lambda function. The new column should be at the 3 rd position from the start in the given dataset & its values should be : '01-01-2020' for January, 2020 and '01-02-2020' for February 2020 and so on.

Source code:

```
import pandas as pd
from datetime import timedelta
df=pd.read_csv("test_DataScience.csv")
cols=["Month","Year"]
df['Date'] = df[cols].apply(lambda x: '-'.join(x.values.astype(str)), axis="columns")
df['Date']=pd.to_datetime(df['Date'])
df['Date'] = pd.to_datetime(df['Date']).dt.strftime('%d/%m/%Y')
df = df[['Year', 'Month', 'Date', 'Laptop/Desktop', 'Type_of_Customers?','Coming
from','Place_in_India','Level 1','Level 2','Level 3','Level 4']]
df.head()
```

OUTPUT:



Out[37]:

	Year	Month	Date	Laptop/Desktop	Type_of_Customers?	Coming from	Place_in_India	Level 1	Level 2	Level 3	Level 4
0	2020	Jan	01/01/2020	Desktop_Website	Existing_Customer	Came_From_LinkedIn	Bengaluru	NaN	NaN	56892	17178
1	2020	Jan	01/01/2020	Desktop_Website	Existing_Customer	Came_From_LinkedIn	Hyderabad	NaN	NaN	41460	11916
2	2020	Jan	01/01/2020	Desktop_Website	Existing_Customer	Came_From_LinkedIn	Dehradun	NaN	NaN	55561	19461
3	2020	Jan	01/01/2020	Desktop_Website	Existing_Customer	Came_From_LinkedIn	Indore	NaN	NaN	320923	110667
4	2020	Jan	01/01/2020	Desktop_Website	Existing_Customer	Came_From_LinkedIn	Pune	NaN	NaN	220937	46033

2] Replaces the null values with the average of the respective column in the data.

Source code:

```
import pandas as pd
df=pd.read_csv("test_DataScience.csv")
print(df.isnull().sum())    #no of missing values each column
print(df.isnull().sum().sum())  #no of missing values each column
df["Level 1"]=df["Level 1"].fillna(df["Level 1"].mean())
df["Level 2"]=df["Level 2"].fillna(df["Level 2"].mean())
print(df)
```

OUTPUT:

```
Place_in_India       Level 1       Level 2         Level 3  Level 4
0         Bengaluru  7.838702e+05  3.582154e+05    56892    17178
1         Hyderabad  7.838702e+05  3.582154e+05    41460    11916
2          Dehradun  7.838702e+05  3.582154e+05    55561    19461
3            Indore  7.838702e+05  3.582154e+05   320923   110667
4              Pune  7.838702e+05  3.582154e+05   220937    46033
5         Bengaluru  7.838702e+05  3.582154e+05    90241    24229
6         Hyderabad  7.838702e+05  3.582154e+05    77630    18502
7          Dehradun  7.838702e+05  3.582154e+05    91479    24363
8            Indore  7.838702e+05  3.582154e+05   436641   165036
9              Pune  7.838702e+05  3.582154e+05   531446   101317
10        Bengaluru  7.838702e+05  3.582154e+05    32119     6900
11        Hyderabad  7.838702e+05  3.582154e+05    27891     5606
12         Dehradun  7.838702e+05  3.582154e+05    34391     8459
13           Indore  7.838702e+05  3.582154e+05   142422    39296
14             Pune  1.092340e+05  9.810000e+04   120090    20223
15        Bengaluru  1.128690e+05  9.180100e+04    48979    33382
16        Hyderabad  1.103970e+05  8.742900e+04    48899    29031
17         Dehradun  1.564870e+05  1.233240e+05    59084    39804
18           Indore  1.176804e+06  9.601450e+05   604293   373155
19             Pune  3.832600e+05  3.059020e+05   172827   122285
20        Bengaluru  2.682860e+05  1.494760e+05    58622    44999
21        Hyderabad  2.921280e+05  1.318900e+05    50720    32140
22         Dehradun  4.012920e+05  1.713220e+05    51069    36016
23           Indore  1.670668e+06  9.697320e+05   477858   339970
24             Pune  1.748075e+06  8.543770e+05   314289   225823
25        Bengaluru  5.817200e+04  4.409600e+04    18169     8769
26        Hyderabad  8.038400e+04  5.809700e+04    22564     8832
27         Dehradun  1.014530e+05  7.920100e+04    17203    10064
28           Indore  4.944470e+05  3.746380e+05   170498    88331
29             Pune  2.359550e+05  1.628900e+05    54175    30731
30        Bengaluru  7.838702e+05  3.582154e+05    57469    13257
31        Hyderabad  7.838702e+05  3.582154e+05    22092     5405
32         Dehradun  7.838702e+05  3.582154e+05    40947    13345
```

………….

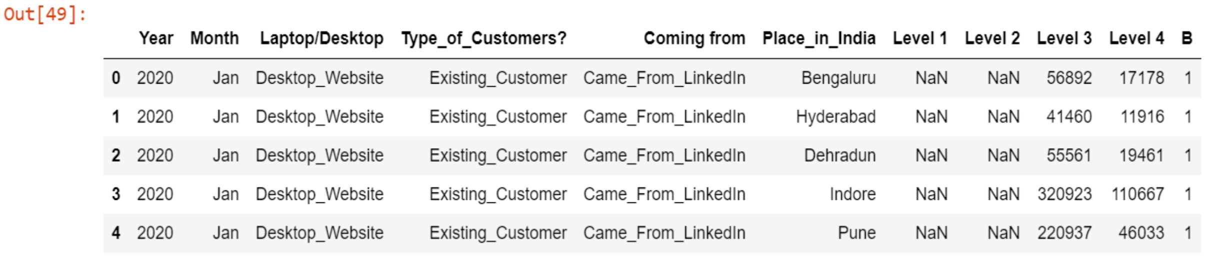
[2160 rows * 5 columns.]

3.] In column ‘B’ replace Jan with 1, feb with 2, march with 3 and so on.

Source code:

```
import pandas as pd
def GetMonthInInt(month):
MonthInInts=pd.Series([1,2,3,4,5,6,7,8,9,10,11,12],index=['jan','feb','mar','apr','may','jun','jul','aug',
'sep','oct','nov','dec'])
   return MonthInInts[month.lower()]
df=pd.read_csv("test_DataScience.csv")
df['B']= df['Month'].apply(GetMonthInInt)
#print(df)
df.head()
```

OUTPUT:



Out[49]:

	Year	Month	Laptop/Desktop	Type_of_Customers?	Coming from	Place_in_India	Level 1	Level 2	Level 3	Level 4	B
0	2020	Jan	Desktop_Website	Existing_Customer	Came_From_LinkedIn	Bengaluru	NaN	NaN	56892	17178	1
1	2020	Jan	Desktop_Website	Existing_Customer	Came_From_LinkedIn	Hyderabad	NaN	NaN	41460	11916	1
2	2020	Jan	Desktop_Website	Existing_Customer	Came_From_LinkedIn	Dehradun	NaN	NaN	55561	19461	1
3	2020	Jan	Desktop_Website	Existing_Customer	Came_From_LinkedIn	Indore	NaN	NaN	320923	110667	1
4	2020	Jan	Desktop_Website	Existing_Customer	Came_From_LinkedIn	Pune	NaN	NaN	220937	46033	1

```
   Place_in_India     Level 1    Level 2  Level 3  Level 4  B
0        Bengaluru        NaN        NaN    56892    17178  1
1        Hyderabad        NaN        NaN    41460    11916  1
2         Dehradun        NaN        NaN    55561    19461  1
3           Indore        NaN        NaN   320923   110667  1
4             Pune        NaN        NaN   220937    46033  1
5        Bengaluru        NaN        NaN    90241    24229  1
6        Hyderabad        NaN        NaN    77630    18502  1
7         Dehradun        NaN        NaN    91479    24363  1
8           Indore        NaN        NaN   436641   165036  1
9             Pune        NaN        NaN   531446   101317  1
10       Bengaluru        NaN        NaN    32119     6900  1
11       Hyderabad        NaN        NaN    27891     5606  1
12        Dehradun        NaN        NaN    34391     8459  1
13          Indore        NaN        NaN   142422    39296  1
14            Pune   109234.0    98100.0   120090    20223  1
15       Bengaluru   112869.0    91801.0    48979    33382  1
16       Hyderabad   110397.0    87429.0    48899    29031  1
17        Dehradun   156487.0   123324.0    59084    39804  1
18          Indore  1176804.0   960145.0   604293   373155  1
19            Pune   383260.0   305902.0   172827   122285  1
20       Bengaluru   268286.0   149476.0    58622    44999  1
21       Hyderabad   292128.0   131890.0    50720    32140  1
22        Dehradun   401292.0   171322.0    51069    36016  1
23          Indore  1670668.0   969732.0   477858   339970  1
24            Pune  1748075.0   854377.0   314289   225823  1
25       Bengaluru    58172.0    44096.0    18169     8769  1
26       Hyderabad    80384.0    58097.0    22564     8832  1
27        Dehradun   101453.0    79201.0    17203    10064  1
28          Indore   494447.0   374638.0   170498    88331  1
```

```
56         Hyderabad    50889.0    34069.0    10913     5024   1
57          Dehradun    58247.0    41982.0    10229     6334   1
58            Indore   507337.0   365834.0   176165    57549   1
59              Pune   221607.0   127471.0    39922    22624   1
```

....

....

.

.

.

```
60         Bengaluru        NaN        NaN    47265    14196   2
61         Hyderabad        NaN        NaN    33702     9671   2
62          Dehradun        NaN        NaN    43417    15551   2
63            Indore        NaN        NaN   245423    80299   2
64              Pune        NaN        NaN   228051    36538   2
65         Bengaluru        NaN        NaN    78378    20261   2
66         Hyderabad        NaN        NaN    64729    15062   2
67          Dehradun        NaN        NaN    75333    19025   2
68            Indore        NaN        NaN   363559   124359   2
69              Pune        NaN        NaN   441298    82153   2
70         Bengaluru        NaN        NaN    26704     5672   2
71         Hyderabad        NaN        NaN    22736     4181   2
72          Dehradun        NaN        NaN    26907     6179   2
73            Indore        NaN        NaN   115601    31589   2
74              Pune        NaN        NaN   103558    16829   2
75         Bengaluru    98574.0    80672.0    39272    26296   2
76         Hyderabad    95163.0    74427.0    39687    23332   2
77          Dehradun   131647.0   103960.0    46826    31260   2
78            Indore   942288.0   760422.0   462969   278204   2
107         Dehradun   163799.0   109105.0    50685    33708   2
108           Indore  1052420.0   733233.0   404716   232852   2
109             Pune   426286.0   304066.0   156062   108686   2
110        Bengaluru   570318.0   245665.0    56844    25016   2
111        Hyderabad   292515.0   128356.0    40287    19591   2
112         Dehradun   466905.0   193933.0    41464    16147   2
113           Indore  2505768.0  1215226.0   397087   182202   2
114             Pune  1376097.0   719141.0   304051   149451   2
115        Bengaluru    63129.0    37419.0     9099     3503   2
116        Hyderabad    43619.0    28122.0     8379     4058   2
```

....

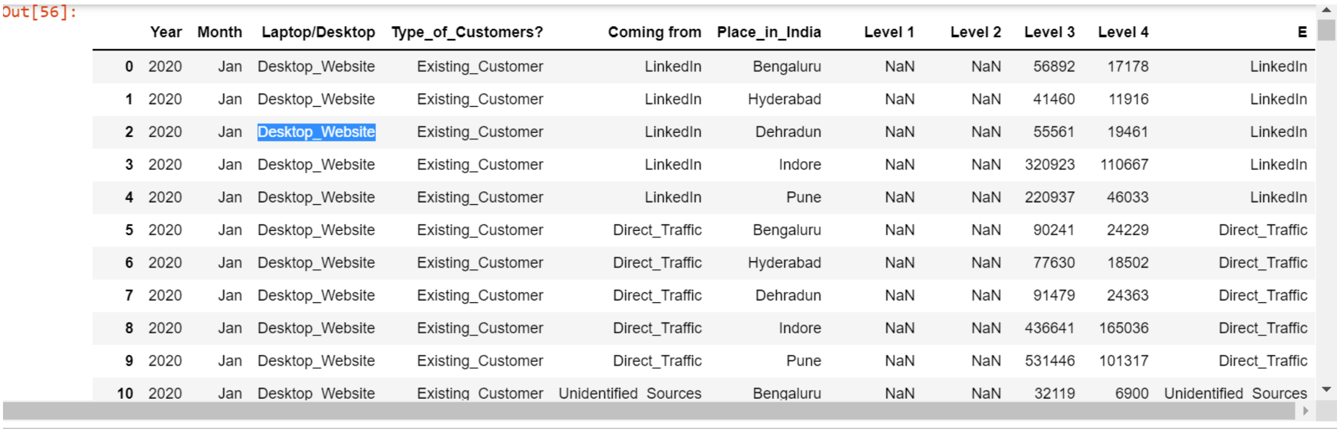
[2160 rows * 6 columns]

4.] In column ‘E’ Replace “Came_From_LinkedIn” with “LinkedIn” and “Landed_Directly” with “Direct_traffic” .

Source code:

```
import pandas as pd
df=pd.read_csv("test_DataScience.csv")
df['Coming from'] = df['Coming from'].replace(['Came_From_LinkedIn', 'Landed_Directly'],['LinkedIn',
'Direct_Traffic'])
df['E']=df['Coming from']
df
```

OUTPUT:



Out[56]:

	Year	Month	Laptop/Desktop	Type_of_Customers?	Coming from	Place_in_India	Level 1	Level 2	Level 3	Level 4	E
0	2020	Jan	Desktop_Website	Existing_Customer	LinkedIn	Bengaluru	NaN	NaN	56892	17178	LinkedIn
1	2020	Jan	Desktop_Website	Existing_Customer	LinkedIn	Hyderabad	NaN	NaN	41460	11916	LinkedIn
2	2020	Jan	Desktop_Website	Existing_Customer	LinkedIn	Dehradun	NaN	NaN	55561	19461	LinkedIn
3	2020	Jan	Desktop_Website	Existing_Customer	LinkedIn	Indore	NaN	NaN	320923	110667	LinkedIn
4	2020	Jan	Desktop_Website	Existing_Customer	LinkedIn	Pune	NaN	NaN	220937	46033	LinkedIn
5	2020	Jan	Desktop_Website	Existing_Customer	Direct_Traffic	Bengaluru	NaN	NaN	90241	24229	Direct_Traffic
6	2020	Jan	Desktop_Website	Existing_Customer	Direct_Traffic	Hyderabad	NaN	NaN	77630	18502	Direct_Traffic
7	2020	Jan	Desktop_Website	Existing_Customer	Direct_Traffic	Dehradun	NaN	NaN	91479	24363	Direct_Traffic
8	2020	Jan	Desktop_Website	Existing_Customer	Direct_Traffic	Indore	NaN	NaN	436641	165036	Direct_Traffic
9	2020	Jan	Desktop_Website	Existing_Customer	Direct_Traffic	Pune	NaN	NaN	531446	101317	Direct_Traffic
10	2020	Jan	Desktop Website	Existing Customer	Unidentified Sources	Bengaluru	NaN	NaN	32119	6900	Unidentified Sources

Part 2: Descriptive statistics

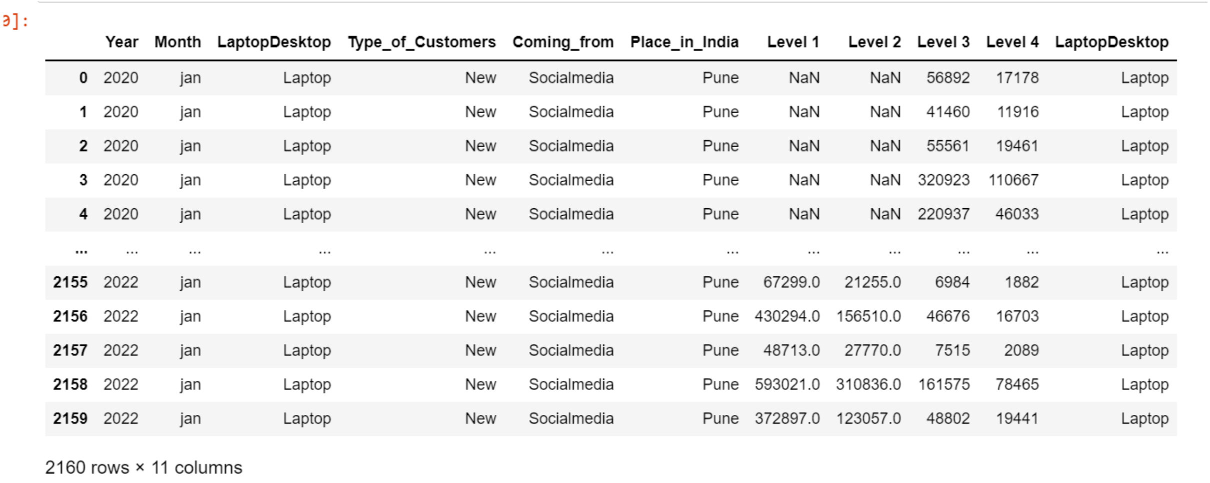
Write a function called descriptive_stats(‘Year’, ‘Month’ , ‘Laptop/Desktop’ , ‘Type_of_Customers?’ , ‘Coming from’ , ‘Place_in_India’) which, when called, would perform the following activity:

Q.1] Would filter the dataframe with the given parameters; if any parameter is missed, then consider a default value to that parameter (e.g., default: ‘year’ – 2020, ‘month’-Jan, & so on) . Let’s call this new dataframe ‘df’.

Source code:

```
import pandas as pd
df=pd.read_csv('test_DataScience.csv')
df['LaptopDesktop']=df['Laptop/Desktop']
df.rename(columns = {'Laptop/Desktop':'LaptopDesktop'}, inplace = True)
df.rename(columns = {'Type_of_Customers?':'Type_of_Customers'}, inplace = True)
df.rename(columns = {'Coming from':'Coming_from'}, inplace = True)
 def descriptive_stats(Year='2020',Month='Jan',LaptopDesktop='Laptop',Type_of_Customers='New',
Coming_from='Socialmedia',Place_in_India='Pune'):
    return df.predict_future
df.Year='2020'
df.Month='jan'
df.LaptopDesktop='Laptop'
df.Type_of_Customers='New'
df.Coming_from='Socialmedia'
df.Place_in_India='Pune'
df
```

OUTPUT:



ə]:

	Year	Month	LaptopDesktop	Type_of_Customers	Coming_from	Place_in_India	Level 1	Level 2	Level 3	Level 4	LaptopDesktop
0	2020	jan	Laptop	New	Socialmedia	Pune	NaN	NaN	56892	17178	Laptop
1	2020	jan	Laptop	New	Socialmedia	Pune	NaN	NaN	41460	11916	Laptop
2	2020	jan	Laptop	New	Socialmedia	Pune	NaN	NaN	55561	19461	Laptop
3	2020	jan	Laptop	New	Socialmedia	Pune	NaN	NaN	320923	110667	Laptop
4	2020	jan	Laptop	New	Socialmedia	Pune	NaN	NaN	220937	46033	Laptop
...	...	...	...	...	...	...	...	...	...	...	...
2155	2022	jan	Laptop	New	Socialmedia	Pune	67299.0	21255.0	6984	1882	Laptop
2156	2022	jan	Laptop	New	Socialmedia	Pune	430294.0	156510.0	46676	16703	Laptop
2157	2022	jan	Laptop	New	Socialmedia	Pune	48713.0	27770.0	7515	2089	Laptop
2158	2022	jan	Laptop	New	Socialmedia	Pune	593021.0	310836.0	161575	78465	Laptop
2159	2022	jan	Laptop	New	Socialmedia	Pune	372897.0	123057.0	48802	19441	Laptop

2160 rows × 11 columns

Q.2] Generates the summary statistics (Mean, Median, Quartile, standard deviation) of all the numerical columns of the new dataframe, df.

Source Code:

```
import pandas as pd
df=pd.read_csv('test_DataScience.csv')
df['LaptopDesktop']=df['Laptop/Desktop']
df.rename(columns = {'Laptop/Desktop':'LaptopDesktop'}, inplace = True)
df.rename(columns = {'Type_of_Customers?':'Type_of_Customers'}, inplace = True)
df.rename(columns = {'Coming from':'Coming_from'}, inplace = True)

def descriptive_stats(Year='2020',Month='Jan',LaptopDesktop='Laptop',Type_of_Customers='New',
Coming_from='Socialmedia',Place_in_India='Pune'):
    return df.predict_future
df.Year='2020'
df.Month='jan'
df.LaptopDesktop='Laptop'
df.Type_of_Customers='New'
df.Coming_from='Socialmedia'
df.Place_in_India='Pune'
df
#index_labels=['r1','r2','r3','r4','r5','r6']
print("The shape of the dataframe is: ", df.shape)
#df.describe()
dfnew = pd.DataFrame(df,index=index_labels)
df_mean = dfnew["Year"].mean()
print(df_mean) #calculating mean
print(dfnew.median()) #calculating median
dfnew.std(axis = 1, skipna = True)
# Removing the outliers
def removeOutliers(dfnew, Year):
  Q3 = np.quantile(dfnew[Year], 0.75)
  Q1 = np.quantile(dfnew[Year], 0.25)
  IQR = Q3 - Q1

print("IQR value for column %s is: %s" % (Year, IQR))
global outlier_free_list
global filtered_data
lower_range = Q1 - 1.5 * IQR
upper_range = Q3 + 1.5 * IQR
outlier_free_list = [x for x in data[Year] if (
(x > lower_range) & (x < upper_range))]
filtered_data = df.loc[data[Year].isin(outlier_free_list)]
for i in dfnew.columns:
  if i == dfnew.columns[0]:
    removeOutliers(df, i)
  else:
    removeOutliers(filtered_data, i)
```

```
# Assigning filtered data back to our original variable
dfnew = filtered_data
print("Shape of data after outlier removal is: ", dfnew.shape)
```

OUTPUT:

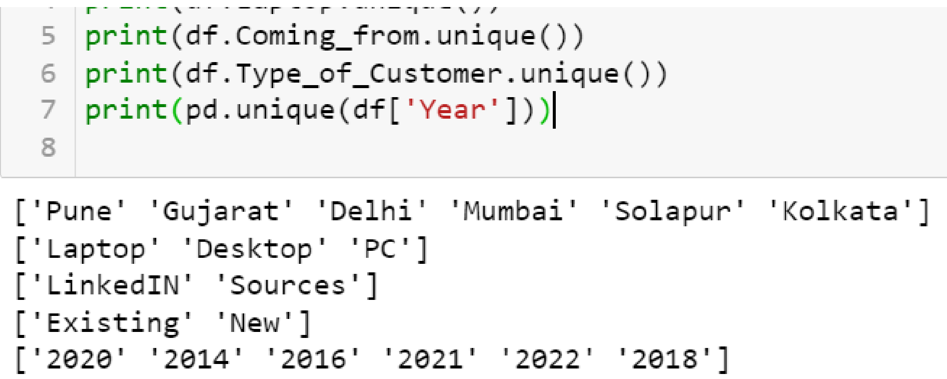
```
The shape of the dataframe is:  (6, 6)
3.367002366936702e+22
Year    2019.0
dtype: float64
```

Q.3] Produce a list of all the unique values & data types present in the non-numeric columns in df.

Source code:

```
import pandas as pd
import numpy as np
df=pd.read_csv('test_DataScience.csv')
df['LaptopDesktop']=df['Laptop/Desktop']
df.rename(columns = {'Laptop/Desktop':'LaptopDesktop'}, inplace = True)
df.rename(columns = {'Type_of_Customers?':'Type_of_Customers'}, inplace = True)
df.rename(columns = {'Coming from':'Coming_from'}, inplace = True)

def predict_future(Year='2020',Month='Jan',LaptopDesktop='Laptop',Type_of_Customers='New',
Coming_from='Socialmedia',Place_in_India='Pune'):
    return df.predict_future
df.Year='2020'
df.Month='jan'
df.LaptopDesktop='Laptop'
df.Type_of_Customers='New'
df.Coming_from='Socialmedia'
df.Place_in_India='Pune'
df
df.applymap(np.isreal).all(1)  #if all values are false then they are non-numeric.
df[~df.applymap(np.isreal).all(1)]
print(df.Place_in_India.unique())
print(df.LaptopDesktop.unique())
print(df.Coming_from.unique())
print(df.Type_of_Customers.unique())
print(pd.unique(df['Year']))
```

OUTPUT:



```
print(df.Coming_from.unique())
print(df.Type_of_Customer.unique())
print(pd.unique(df['Year']))
```

```
['Pune' 'Gujarat' 'Delhi' 'Mumbai' 'Solapur' 'Kolkata']
['Laptop' 'Desktop' 'PC']
['LinkedIN' 'Sources']
['Existing' 'New']
['2020' '2014' '2016' '2021' '2022' '2018']
```

Part 3: Prescriptive statistics

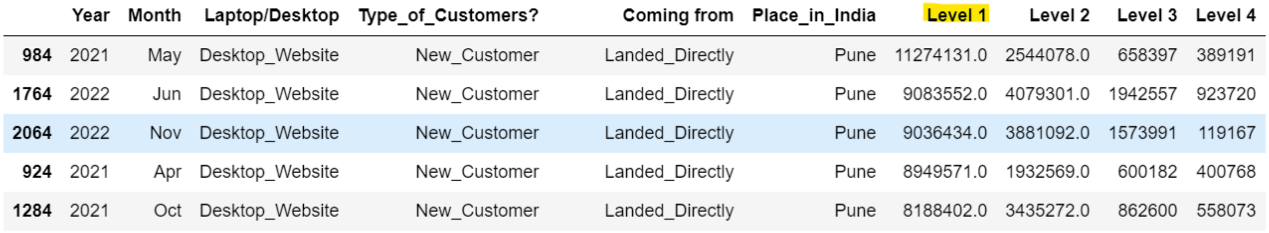
The marketing manager has asked you the following questions, please provide the answers along with summarized data supporting your answer.

1] What are the top 3 "Place_in_India" on the basis of column "Level 1" for the year 2021 and 2022 separately ?

Source code:

```
import pandas as pd
import numpy as np
df=pd.read_csv("test_DataScience.csv")
df.sort_values(['Level 1','Place_in_India'],ascending = False).groupby('Level 1').head(5)
```

OUTPUT:



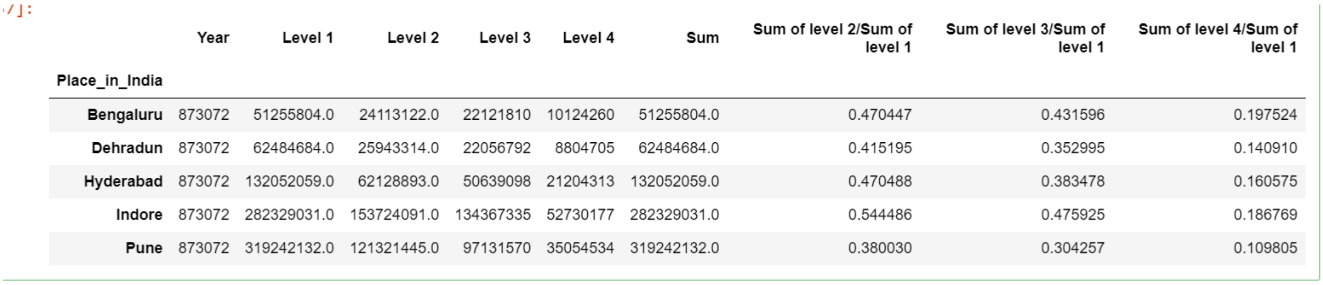
	Year	Month	Laptop/Desktop	Type_of_Customers?	Coming from	Place_in_India	Level 1	Level 2	Level 3	Level 4
984	2021	May	Desktop_Website	New_Customer	Landed_Directly	Pune	11274131.0	2544078.0	658397	389191
1764	2022	Jun	Desktop_Website	New_Customer	Landed_Directly	Pune	9083552.0	4079301.0	1942557	923720
2064	2022	Nov	Desktop_Website	New_Customer	Landed_Directly	Pune	9036434.0	3881092.0	1573991	119167
924	2021	Apr	Desktop_Website	New_Customer	Landed_Directly	Pune	8949571.0	1932569.0	600182	400768
1284	2021	Oct	Desktop_Website	New_Customer	Landed_Directly	Pune	8188402.0	3435272.0	862600	558073

Q.2 Please, provide the data for all the cities & for all the years, the following format as shown in the below snippet:

Source code:

```
df2 = df.groupby('Place_in_India').sum()
df2
df2['Sum of level 2/Sum of level 1'] = df2['Level 2']/df2['Level 1']
df2['Sum of level 3/Sum of level 1'] = df2['Level 3']/df2['Level 1']
df2['Sum of level 4/Sum of level 1'] = df2['Level 4']/df2['Level 1']
df2
```

OUTPUT:



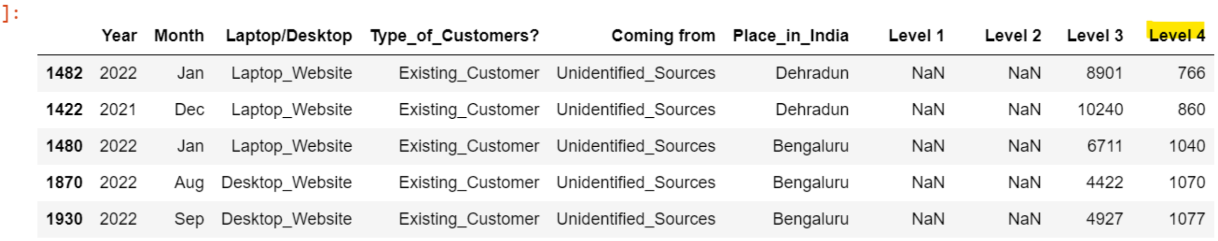
	Year	Level 1	Level 2	Level 3	Level 4	Sum	Sum of level 2/Sum of level 1	Sum of level 3/Sum of level 1	Sum of level 4/Sum of level 1
Place_in_India									
Bengaluru	873072	51255804.0	24113122.0	22121810	10124260	51255804.0	0.470447	0.431596	0.197524
Dehradun	873072	62484684.0	25943314.0	22056792	8804705	62484684.0	0.415195	0.352995	0.140910
Hyderabad	873072	132052059.0	62128893.0	50639098	21204313	132052059.0	0.470488	0.383478	0.160575
Indore	873072	282329031.0	153724091.0	134367335	52730177	282329031.0	0.544486	0.475925	0.186769
Pune	873072	319242132.0	121321445.0	97131570	35054534	319242132.0	0.380030	0.304257	0.109805

Q.3] What are the bottom 3 "Place_in_India" on the basis of column "Level 4" for the year 2021 and 2022 separately ?

Source code:

```
import pandas as pd
import numpy as np
df=pd.read_csv("test_DataScience.csv")
df.sort_values(['Level 4','Place_in_India'],ascending = True).groupby('Level 4').head(5)
```

OUTPUT:



	Year	Month	Laptop/Desktop	Type_of_Customers?	Coming from	Place_in_India	Level 1	Level 2	Level 3	Level 4
1482	2022	Jan	Laptop_Website	Existing_Customer	Unidentified_Sources	Dehradun	NaN	NaN	8901	766
1422	2021	Dec	Laptop_Website	Existing_Customer	Unidentified_Sources	Dehradun	NaN	NaN	10240	860
1480	2022	Jan	Laptop_Website	Existing_Customer	Unidentified_Sources	Bengaluru	NaN	NaN	6711	1040
1870	2022	Aug	Desktop_Website	Existing_Customer	Unidentified_Sources	Bengaluru	NaN	NaN	4422	1070
1930	2022	Sep	Desktop_Website	Existing_Customer	Unidentified_Sources	Bengaluru	NaN	NaN	4927	1077

Part 4: Simple Machine learning question:

Write a function called predict_future(‘Year’, ‘Month’ , ‘Laptop/Desktop’ , ‘Type_of_Customers?’ , ‘Coming from’ , ‘Place_in_India’) which, when called, would perform the following activity:

Q1.]Predict “Level 4” for the 12 months of 2023 given the parameters of the function. (Please make sure the parameters have default values in place)

Source code:

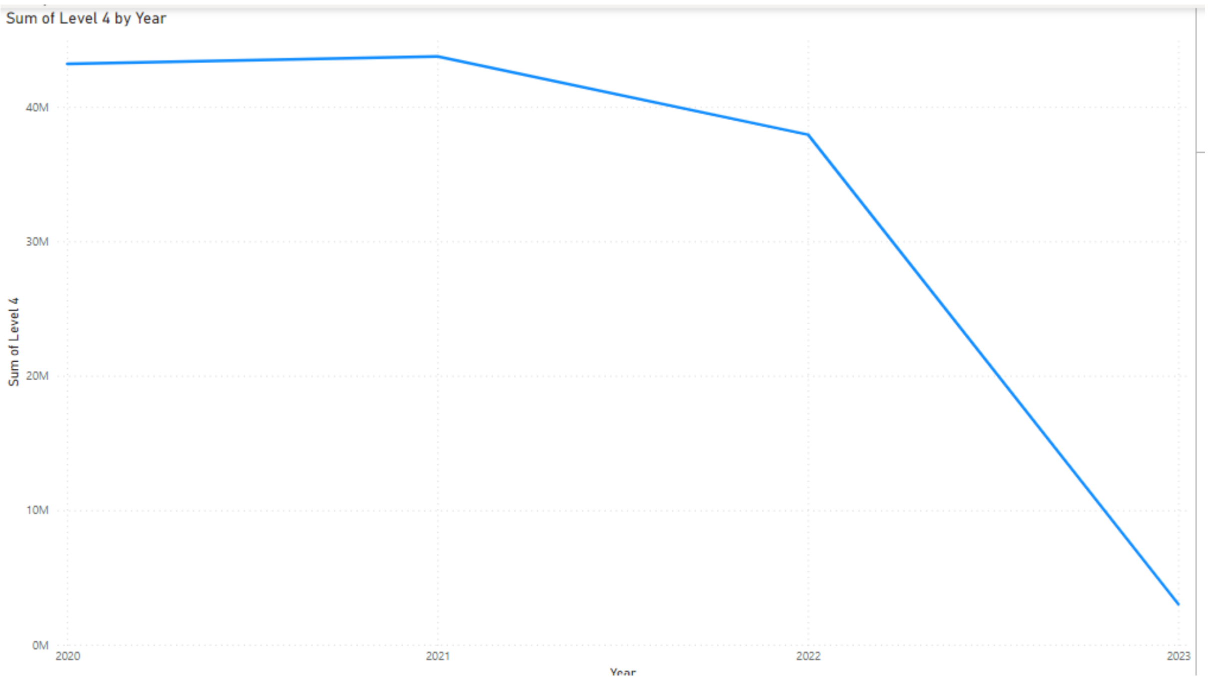
```
import pandas as pd
import numpy
df=pd.read_csv('test_DataScience.csv')
df['LaptopDesktop']=df['Laptop/Desktop']
df.rename(columns = {'Laptop/Desktop':'LaptopDesktop'}, inplace = True)
df.rename(columns = {'Type_of_Customers?':'Type_of_Customers'}, inplace = True)
df.rename(columns = {'Coming from':'Coming_from'}, inplace = True)

def predict_future(Year='2020',Month='Jan',LaptopDesktop='Laptop',Type_of_Customers='New',
Coming_from='Socialmedia',Place_in_India='Pune'):
    return df.predict_future
df.Year='2020'
df.Month='jan'
df.LaptopDesktop='Laptop'
df.Type_of_Customers='New'
df.Coming_from='Socialmedia'
df.Place_in_India='Pune'
df
#df[(df['Level 4'].dt.month == 1) & (df['Level 4'].dt.day == 1)].mean()
df = df.groupby(by=[df.index.Year, df.index.Level4]).mean()
```

Q.2]Generates the overall Forecast error, MAPE and RMSE of your prediction of the year 2022, 2021 & 2020 for the given parameters.

Source code:

```
import numpy as np
from sklearn.model_selection import train_test_split
import pandas as pd
df=pd.read_csv('test_DataScience.csv')
df['LaptopDesktop']=df['Laptop/Desktop']
df.rename(columns = {'Laptop/Desktop':'LaptopDesktop'}, inplace = True)
df.rename(columns = {'Type_of_Customers?':'Type_of_Customers'}, inplace = True)
df.rename(columns = {'Coming from':'Coming_from'}, inplace = True)
  def predict_future(Year='2020',Month='Jan',LaptopDesktop='Laptop',Type_of_Customers='New',
Coming_from='Socialmedia',Place_in_India='Pune'):
    return df.predict_future
df.Year='2020'
df.Month='jan'
df.LaptopDesktop='Laptop'
df.Type_of_Customers='New'
df.Coming_from='Socialmedia'
df.Place_in_India='Pune'
df
 #Separating the dependent and independent data variables into two data frames.
X = df.drop(['Year'],axis=1)
Y = df['Year']
 # Splitting the dataset into 80% training data and 20% testing data.
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=.20, random_state=0)
 #Defining MAPE function
def MAPE(Y_actual,Y_Predicted):
    mape = np.mean(np.abs((Y_actual - Y_Predicted)/Y_actual))*100
    return mape
#Building the Linear Regression Model
from sklearn.linear_model import LinearRegression
linear_model = LinearRegression().fit(X_train , Y_train)
 #Predictions on Testing data
LR_Test_predict = linear_model.predict(X_test)
 # Using MAPE error metrics to check for the error rate and accuracy level
LR_MAPE= MAPE(Y_test,LR_Test_predict)
print("MAPE: ",LR_MAPE)
```

Q.3] Plot a line graph of the level 4 actual numbers from 2020-2022 & in the same graph, there should be the predicted numbers for 2023. The x-axis should be the timeline from 2020 Jan to 2023 Dec and the y-axis should be the value of the level 4 column, The below graph is just an example of how your plot should look like.



Part 5: Visualization:

- Please write a code to display :

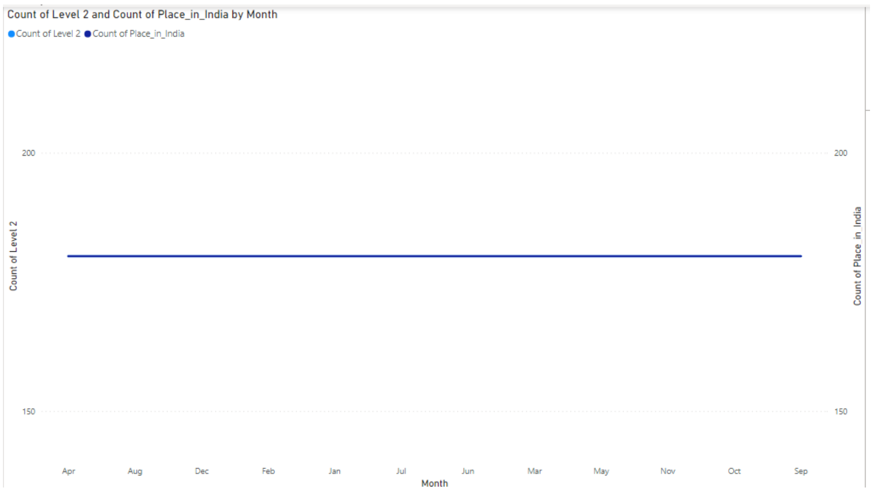
  Q1] A line graph for "Level 2" for the different "Place_in_India?" over the months of the year 2020 & 2021. (Hint: On x-axis, there should be months for 2020 & 2021 and Y axis should be "Level 2" and there should be different lines depicting different regions of "Place_in_India?") Plot a neat graph.

  Source code:

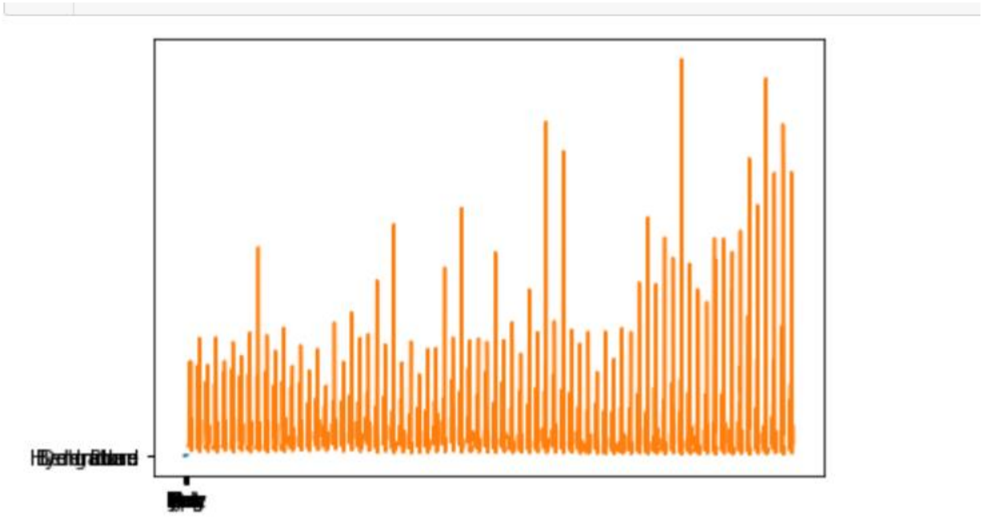
  ```
  import pandas as pd
  import matplotlib.pyplot as plt
  df=pd.read_csv('test_DataScience.csv')
  df = df.head()
  data = pd.DataFrame(df, columns=["Month", "Level 2", "Place_in_India"])
  data=data.loc["2020":"2021"]
  # plot the dataframe
  data.plot(x="Month", y=["Level 2", "Place_in_India"], kind="bar", figsize=(9, 8))
  # print bar graph
  plt.show()
  ```

  OUTPUT:

  From PowerBI



From Jupyter notebook:

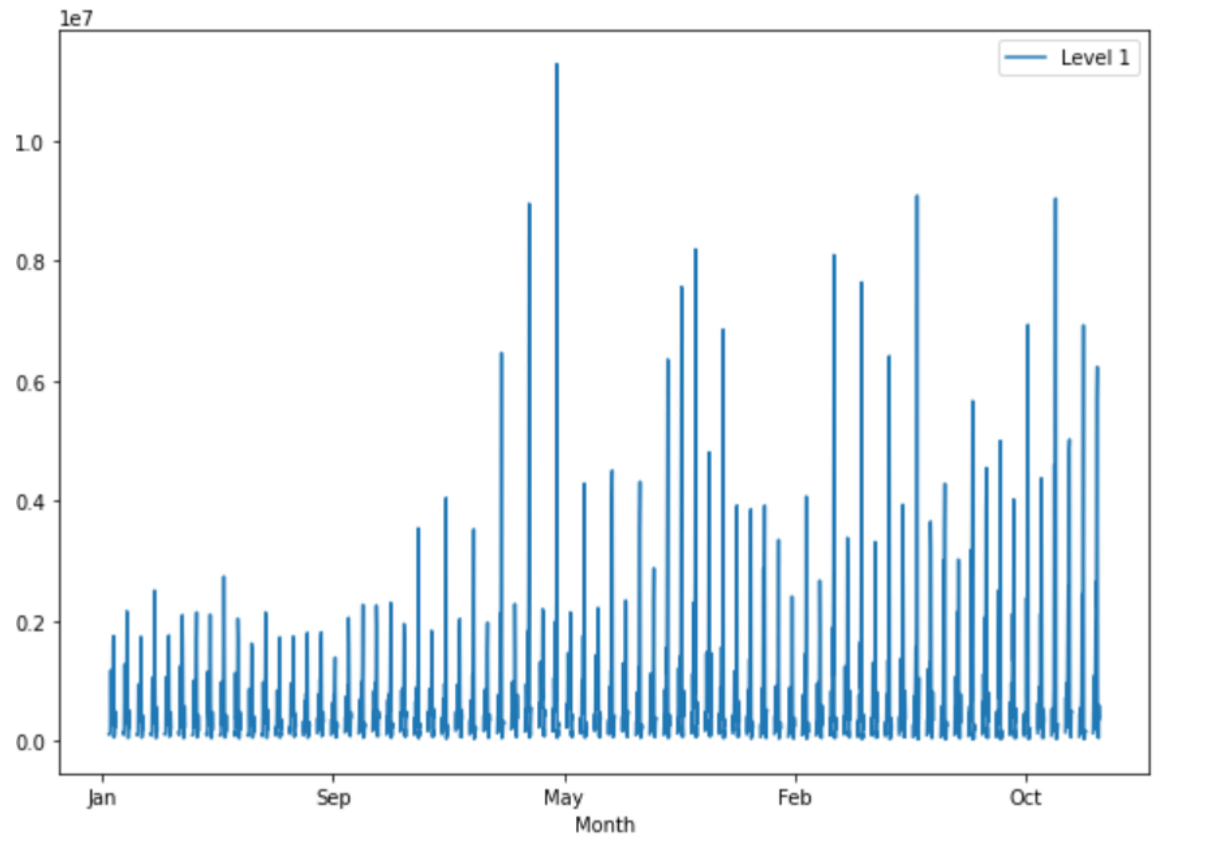


Q2] A line graph for “Level 1” for the different “Laptop/Desktop” over the months of the year 2020 & 2021. (Hint : On x axis there should be months from jan- 2020 to dec- 2021 and Y axis should be the sum of “Level 1” and there should be different lines depicting different devices used.)

Source code:

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv('test_DataScience.csv')
#df = df.head()
data = pd.DataFrame(df, columns=["Month", "Level 1", "Laptop/Desktop"])
data=data.loc["2020":"2021"]
data=data.loc[,:"Jan":"Dec"]
# plot the dataframe
data.plot(x="Month", y=["Level 1", "Laptop/Desktop"], kind="line", figsize=(10, 7))
# print bar graph
plt.show()
```

OUTPUT:

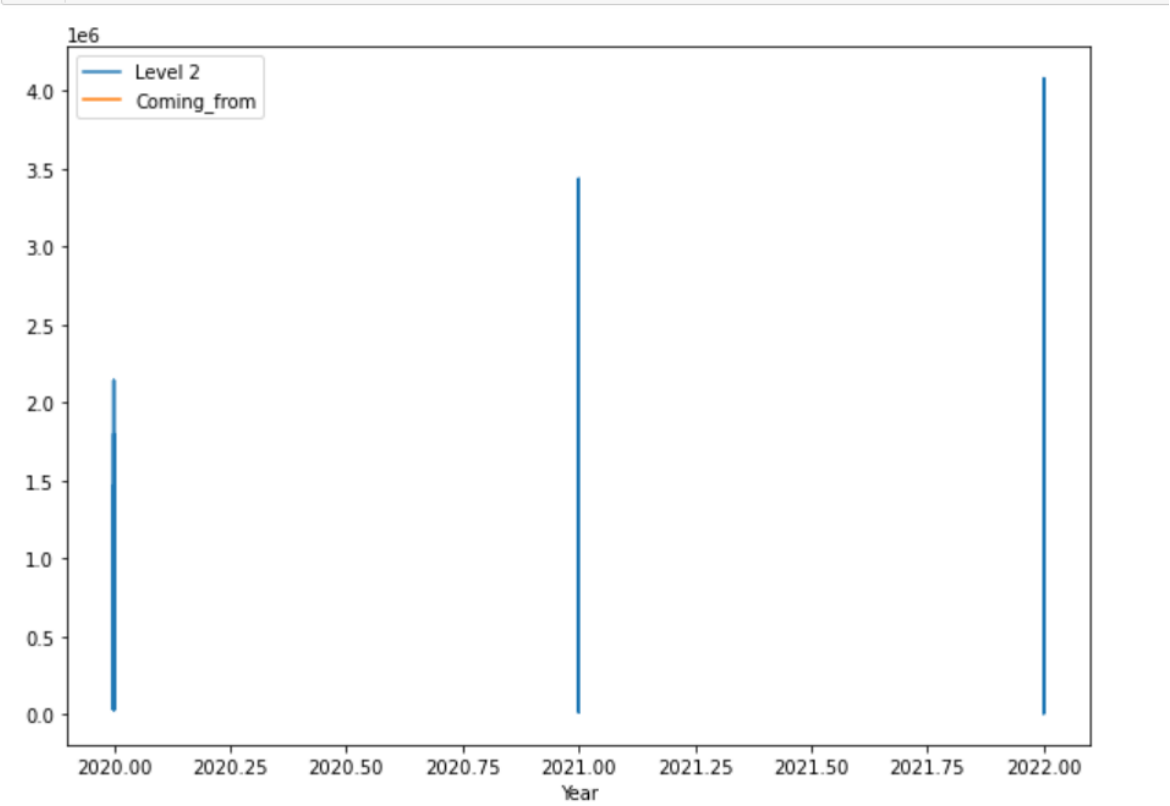


Q3] A line graph for "Level 2" for the different "Coming from" over the months of the year 2021 & 2022.

Source code:

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv('test_DataScience.csv')
#df = df.head()
data = pd.DataFrame(df, columns=["Year", "Level 2", "Coming_from"])
data=data.loc["2020":"2021"]
# plot the dataframe
data.plot(x="Year", y=["Level 2", "Coming_from"], kind="line", figsize=(10, 7))
# print bar graph
plt.show()
```

OUTPUT:

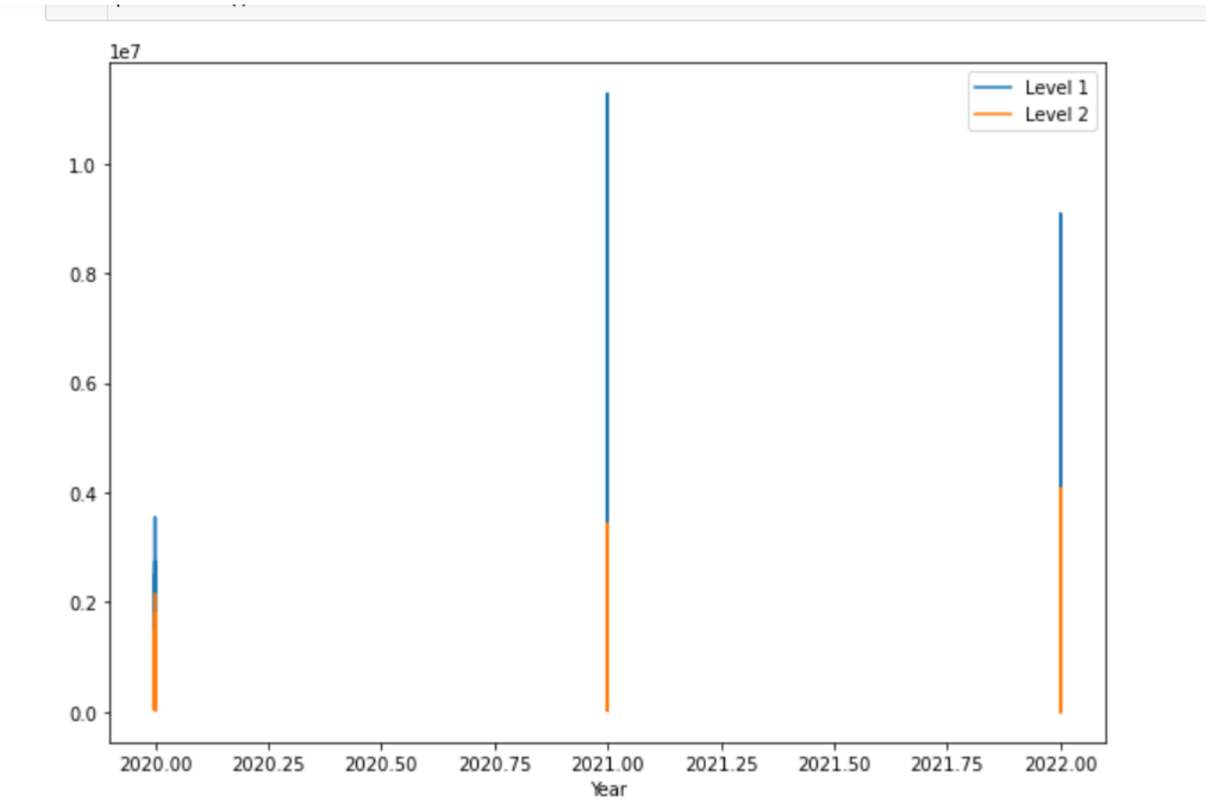


Q4] A line graph for “Level 1” and “Level 2” over the months of the year 2020, 2021 & 2022.

Source code:

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv(‘test_DataScience.csv’)
#df = df.head()
data = pd.DataFrame(df, columns=[“Year”, “Level 1”, “Level 2”])
data=data.loc["2020":"2022"]
# plot the dataframe
data.plot(x=”Year”, y=[“Level 1”, “Level 2”], kind=”line”, figsize=(10, 7))
# print bar graph
plt.show()
```

OUTPUT:

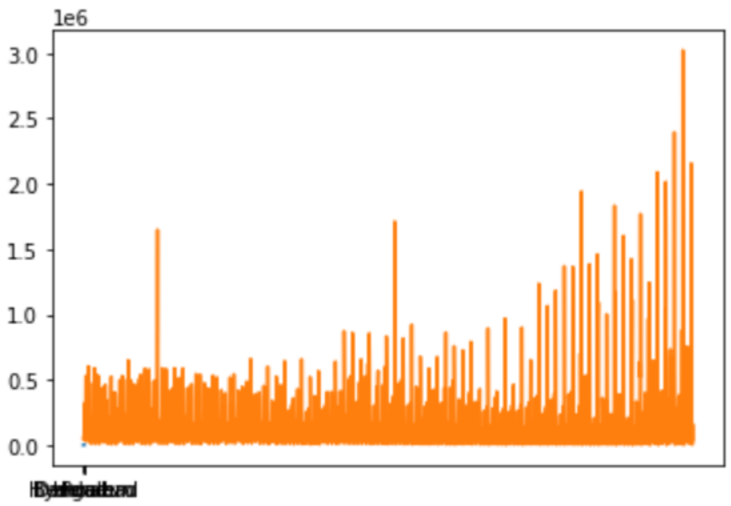


Q5] A line graph for “Level 3” foyearslace_in_India” over the months of the year 2020 and 2021.

Source code:

```
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv('test_DataScience.csv')
df=df.loc["2020":"2021"]
plt.plot( df["Place_in_India"],df["Year"], df["Level 3"])
plt.show()
```

OUTPUT:



Q6] Please add any insights you could derive from all the graphs above.

Answer: As we can see, the market was high during pandemic that is in 2021. Most of the users are from Pune and Bengaluru. And sales are high in the month of May-2021.

Part 6: About the Previous projects

● Please describe any interesting project you did in the Data Science domain in more than 250 words. Attach Github links if possible.

Answer: I have built a database in Healthcare, Banking, Ashram database using MySQL server. I'll share my drive link which includes complete database.

In June-2022, our team worked on database to build fitness freaks website. We collected data and segregated it based on different weight type. And recommended diet accordingly. We worked on bionic sensors which take input as muscle strain and through API, this data is saved on cloud and reports are generated accordingly. The report includes weekly diet, exercises according to BMI. Sport biomechanics represents an important research field aimed at analysing sport movements in order to quantitatively evaluate athlete performance, offer useful tools and guidelines for coaches to apply during athlete training and prevent or minimize the risk of injury. Recent technological innovations allow the performance of movement analysis during sporting activities thanks to the compact wearable sensors that do not influence the technical movements of athletes. The aim of this project is to present the design and development of a wearable multi-sensor system that is affordable for all types of users and can be used for a long time for the application of exercise monitoring. Wearable sensors are widely used in healthcare, due to their hardware capacity, small footprint and lower cost compared to equivalent medical instruments capable of monitoring the same vital signs.Our device includes combination of sensors that is MC sensors, Strain sensors and cloud.

I have also designed graphs, gantt chart, stacked column charts, pert chart, pie chart,etc in Power BI Desktop Visualisation tool.

There is one more project on which me and my team is working, basically we are collecting database. Processing and filtering will be done in the month of January, registered for SIH-2023.

Part 7: Time management

I managed time to solve this assignment after college hours (9:30 to 5:30-college). The same way I'll adjust and prepare new schedule for this internship. I will work from 6:00pm. I will complete my academics work and projects in college hours. And after college can work efficiently till night, at least for 6hours everyday. I will make sure that my work doesn't get affected due to academics. As there have been many incidents where I have been working on multiple thinks. I will adjust and make separate schedule for daily tasks. However we learn same concepts in college session, so this internship will more be like implementing what I have learnt throughout these 3 years of engineering. Its fun working on things which you love.