Comprehensive data analaysis with pandas

In [2]:
```
# Importing the pandas
```

In [3]:
```
import pandas

import pandas as pd
```

In [4]:
```
pd.__version__
```

Out[4]:
```
'2.2.2'
```

In [5]:
```
# Importing the numpy
```

In [6]:
```
import numpy

import numpy as np
```

In [7]:
```
# Data import with pandas
```

In [8]:
```
data = pd.read_csv(r'C:\Users\SAIF SHAIK\Downloads\test.csv.zip')
```

In [9]:
```
data
```

Out[9]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_Cit
0	1000004	P00128942	M	46-50	7	B	
1	1000009	P00113442	M	26-35	17	C	
2	1000010	P00288442	F	36-45	1	B	
3	1000010	P00145342	F	36-45	1	B	
4	1000011	P00053842	F	26-35	1	C	
...	...	...	...	...	...	...	
233594	1006036	P00118942	F	26-35	15	B	
233595	1006036	P00254642	F	26-35	15	B	
233596	1006036	P00031842	F	26-35	15	B	
233597	1006037	P00124742	F	46-50	1	C	
233598	1006039	P00316642	F	46-50	0	B	

233599 rows × 11 columns

In [10]:
```
type(data)
```

Out[10]:
```
pandas.core.frame.DataFrame
```

In [11]:
```
data.shape
```

Out[11]:
```
(233599, 11)
```

In [12]:
```
data.head()  # default its prints the first 5 coloumns
```

Out[12]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Year
0	1000004	P00128942	M	46-50	7	B	
1	1000009	P00113442	M	26-35	17	C	
2	1000010	P00288442	F	36-45	1	B	4
3	1000010	P00145342	F	36-45	1	B	4
4	1000011	P00053842	F	26-35	1	C	

In [13]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 233599 entries, 0 to 233598
Data columns (total 11 columns):
 #   Column                      Non-Null Count   Dtype  
---  ------                      --------------   -----  
 0   User_ID                     233599 non-null  int64  
 1   Product_ID                  233599 non-null  object 
 2   Gender                      233599 non-null  object 
 3   Age                         233599 non-null  object 
 4   Occupation                  233599 non-null  int64  
 5   City_Category               233599 non-null  object 
 6   Stay_In_Current_City_Years  233599 non-null  object 
 7   Marital_Status              233599 non-null  int64  
 8   Product_Category_1          233599 non-null  int64  
 9   Product_Category_2          161255 non-null  float64
 10  Product_Category_3          71037 non-null   float64
dtypes: float64(2), int64(4), object(5)
memory usage: 19.6+ MB
```

In [14]:

```
data.isnull().sum()          # We can check the total number of missing values in e
```

Out[14]:

```
User_ID                            0
Product_ID                         0
Gender                             0
Age                                0
Occupation                         0
City_Category                      0
Stay_In_Current_City_Years         0
Marital_Status                     0
Product_Category_1                 0
Product_Category_2             72344
Product_Category_3            162562
dtype: int64
```

In [15]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [16]:

```
data.fillna(method = 'pad')
```

Out[16]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_Cit
0	1000004	P00128942	M	46-50	7	B	
1	1000009	P00113442	M	26-35	17	C	
2	1000010	P00288442	F	36-45	1	B	
3	1000010	P00145342	F	36-45	1	B	
4	1000011	P00053842	F	26-35	1	C	
...	...	...	...	...	...	...	
233594	1006036	P00118942	F	26-35	15	B	
233595	1006036	P00254642	F	26-35	15	B	
233596	1006036	P00031842	F	26-35	15	B	
233597	1006037	P00124742	F	46-50	1	C	
233598	1006039	P00316642	F	46-50	0	B	

233599 rows × 11 columns

In [18]:

```
data[['Product_Category_3']].head()
```

Out[18]:

	Product_Category_3
0	NaN
1	NaN
2	NaN
3	NaN
4	12.0

In [19]:

```
data.fillna(method = 'backfill')
```

Out[19]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_Cit
0	1000004	P00128942	M	46-50	7	B	
1	1000009	P00113442	M	26-35	17	C	
2	1000010	P00288442	F	36-45	1	B	
3	1000010	P00145342	F	36-45	1	B	
4	1000011	P00053842	F	26-35	1	C	
...	...	...	...	...	...	...	
233594	1006036	P00118942	F	26-35	15	B	
233595	1006036	P00254642	F	26-35	15	B	
233596	1006036	P00031842	F	26-35	15	B	
233597	1006037	P00124742	F	46-50	1	C	
233598	1006039	P00316642	F	46-50	0	B	

233599 rows × 11 columns

In [44]:

```
data.fillna(method = 'backfill')
```

Out[44]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_Cit
0	1000004	P00128942	M	46-50	7	B	
1	1000009	P00113442	M	26-35	17	C	
2	1000010	P00288442	F	36-45	1	B	
3	1000010	P00145342	F	36-45	1	B	
4	1000011	P00053842	F	26-35	1	C	
...	...	...	...	...	...	...	
233594	1006036	P00118942	F	26-35	15	B	
233595	1006036	P00254642	F	26-35	15	B	
233596	1006036	P00031842	F	26-35	15	B	
233597	1006037	P00124742	F	46-50	1	C	
233598	1006039	P00316642	F	46-50	0	B	

233599 rows × 11 columns

In [48]:

```
data = data.fillna(method = 'pad')
```

In [50]:

```
data
```

Out[50]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_Cit
0	1000004	P00128942	M	46-50	7	B	
1	1000009	P00113442	M	26-35	17	C	
2	1000010	P00288442	F	36-45	1	B	
3	1000010	P00145342	F	36-45	1	B	
4	1000011	P00053842	F	26-35	1	C	
...	...	...	...	...	...	...	
233594	1006036	P00118942	F	26-35	15	B	
233595	1006036	P00254642	F	26-35	15	B	
233596	1006036	P00031842	F	26-35	15	B	
233597	1006037	P00124742	F	46-50	1	C	
233598	1006039	P00316642	F	46-50	0	B	

233599 rows × 11 columns

In [52]:

```
data.isnull().sum()
```

Out[52]:

```
User_ID                       0
Product_ID                    0
Gender                        0
Age                           0
Occupation                    0
City_Category                 0
Stay_In_Current_City_Years    0
Marital_Status                0
Product_Category_1            0
Product_Category_2            0
Product_Category_3            4
dtype: int64
```

In [54]:

```
data[[ 'Product_Category_3']].head()
```

Out[54]:

	Product_Category_3
0	NaN
1	NaN
2	NaN
3	NaN
4	12.0

In [56]:
```
data = data.fillna(method = 'backfill')
```

In [58]:
```
data
```

Out[58]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_Cit
0	1000004	P00128942	M	46-50	7	B	
1	1000009	P00113442	M	26-35	17	C	
2	1000010	P00288442	F	36-45	1	B	
3	1000010	P00145342	F	36-45	1	B	
4	1000011	P00053842	F	26-35	1	C	
...	...	...	...	...	...	...	
233594	1006036	P00118942	F	26-35	15	B	
233595	1006036	P00254642	F	26-35	15	B	
233596	1006036	P00031842	F	26-35	15	B	
233597	1006037	P00124742	F	46-50	1	C	
233598	1006039	P00316642	F	46-50	0	B	

233599 rows × 11 columns

In [60]:
```
data.isnull().sum()
```

Out[60]:
```
User_ID                       0
Product_ID                    0
Gender                        0
Age                           0
Occupation                    0
City_Category                 0
Stay_In_Current_City_Years    0
Marital_Status                0
Product_Category_1            0
Product_Category_2            0
Product_Category_3            0
dtype: int64
```

In [62]:
```
assert pd.notnull(data).all().all()        # assert that there are no missing value
```

In [64]:
```
# make a copy of dataframe
data1 = data.copy()
```

In [66]:
```
# select first row of dataframe

data1.loc[0]
```

Out[66]:
```
User_ID                         1000004
Product_ID                    P00128942
Gender                                M
Age                               46-50
Occupation                            7
City_Category                         B
Stay_In_Current_City_Years            2
Marital_Status                        1
Product_Category_1                    1
Product_Category_2                 11.0
Product_Category_3                 12.0
Name: 0, dtype: object
```

In [70]:
```
#select first five rows for a specific column

data1.loc[:,'Product_Category_3'].head()
```

Out[70]:
```
0    12.0
1    12.0
2    12.0
3    12.0
4    12.0
Name: Product_Category_3, dtype: float64
```

In [72]:
```
#select first row of dataframe

data1.iloc[0]
```

```
Out[72]: User_ID                         1000004
         Product_ID                    P00128942
         Gender                                M
         Age                               46-50
         Occupation                            7
         City_Category                         B
         Stay_In_Current_City_Years            2
         Marital_Status                        1
         Product_Category_1                    1
         Product_Category_2                 11.0
         Product_Category_3                 12.0
         Name: 0, dtype: object
```

```
In [74]: #select last row of dataframe

         data1.iloc[-1]
```

```
Out[74]: User_ID                         1006039
         Product_ID                    P00316642
         Gender                                F
         Age                               46-50
         Occupation                            0
         City_Category                         B
         Stay_In_Current_City_Years           4+
         Marital_Status                        1
         Product_Category_1                    4
         Product_Category_2                  5.0
         Product_Category_3                 12.0
         Name: 233598, dtype: object
```

```
In [76]: data['Product_Category_3'].idxmax()
```

```
Out[76]: 213
```

```
In [78]: data1.loc[data1['Product_Category_3'].idxmax()]
```

```
Out[78]: User_ID                         1000348
         Product_ID                    P00281742
         Gender                                M
         Age                               51-55
         Occupation                            7
         City_Category                         B
         Stay_In_Current_City_Years            2
         Marital_Status                        1
         Product_Category_1                    5
         Product_Category_2                  8.0
         Product_Category_3                 18.0
         Name: 213, dtype: object
```

```
In [80]: data1.at[1, 'Product_Category_3']
```

```
Out[80]: 12.0
```

```
In [84]: data1.iat[1, 10]
```

```
Out[84]: 12.0
```

```
In [88]: data2= data.copy()
```

```
In [90]: data2.head()
```

Out[90]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Year
0	1000004	P00128942	M	46-50	7	B	
1	1000009	P00113442	M	26-35	17	C	
2	1000010	P00288442	F	36-45	1	B	4
3	1000010	P00145342	F	36-45	1	B	4
4	1000011	P00053842	F	26-35	1	C	

```
In [92]: data2.loc[((data2['User_ID'] == 1000004) & (data2['Product_ID'] == 'P00128942')), '
```

```
Out[92]: 0    12.0
         Name: Product_Category_3, dtype: float64
```

```
In [98]: values = [1000004,'P00128942','M',46-50,7,'B',2,1,1,6,11.0,12.0]

         data2_indexed = data2.isin(values)


         data2_indexed.head(10)
```

Out[98]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Yea
0	True	True	True	False	True	True	Fals
1	False	False	True	False	False	False	Fals
2	False	False	False	False	True	True	Fals
3	False	False	False	False	True	True	Fals
4	False	False	False	False	True	False	Fals
5	False	False	True	False	True	False	Fals
6	False	False	True	False	True	False	Fals
7	False	False	True	False	True	False	Fals
8	False	False	True	False	True	False	Fals
9	False	False	True	False	False	False	Fals

In [173…

```
row_mask = data2.isin(values).all(1)

data[row_mask]
```

Out[173…

User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years

In [175…

```
data2_where=data2.where(data2 == 0)


(data2_where).head(10)
```

Out[175…

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Year
0	NaN	NaN	NaN	NaN	NaN	NaN	Nal
1	NaN	NaN	NaN	NaN	NaN	NaN	Nal
2	NaN	NaN	NaN	NaN	NaN	NaN	Nal
3	NaN	NaN	NaN	NaN	NaN	NaN	Nal
4	NaN	NaN	NaN	NaN	NaN	NaN	Nal
5	NaN	NaN	NaN	NaN	NaN	NaN	Nal
6	NaN	NaN	NaN	NaN	NaN	NaN	Nal
7	NaN	NaN	NaN	NaN	NaN	NaN	Nal
8	NaN	NaN	NaN	NaN	NaN	NaN	Nal
9	NaN	NaN	NaN	NaN	NaN	NaN	Nal

In [177…

```
data2.query('(Product_Category_1 > Product_Category_2) & (Product_Category_2 > Prod
```

Out[177…

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_Cit
46	1000090	P00117542	M	55+	13	C	
446	1000767	P00261542	M	26-35	12	C	
1026	1001667	P00020542	M	51-55	16	B	
1076	1001733	P00117542	M	18-25	14	B	
1152	1001837	P00185442	M	26-35	2	B	
...	...	...	...	...	...	...	
232384	1004087	P00255842	M	0-17	4	C	
232442	1004204	P00326742	M	36-45	7	C	
232495	1004277	P00020542	M	36-45	16	A	
232805	1004795	P00255842	M	46-50	16	C	
233193	1005442	P00117542	M	26-35	7	C	

1089 rows × 11 columns

In [179…

```
# let's create a new dataframe

food = pd.DataFrame({'Place':['Home', 'Home', 'Hotel', 'Hotel'],
                    'Time': ['Lunch', 'Dinner', 'Lunch', 'Dinner'],
                    'Food':['Soup', 'Rice', 'Soup', 'Chapati'],
                    'Price($)':[10, 20, 30, 40]})

food
```

Out[179…

	Place	Time	Food	Price($)
0	Home	Lunch	Soup	10
1	Home	Dinner	Rice	20
2	Hotel	Lunch	Soup	30
3	Hotel	Dinner	Chapati	40

In [181…

```
food_indexed1=food.set_index('Place')
```

```
food_indexed1
```

Out[181…

	Time	Food	Price($)
Place			
Home	Lunch	Soup	10
Home	Dinner	Rice	20
Hotel	Lunch	Soup	30
Hotel	Dinner	Chapati	40

In [183…

```
food_indexed2=food.set_index(['Place', 'Time'])

food_indexed2
```

Out[183…

		Food	Price($)
Place	**Time**		
Home	**Lunch**	Soup	10
	Dinner	Rice	20
Hotel	**Lunch**	Soup	30
	Dinner	Chapati	40

In [185…

```
food_indexed2.reset_index()
```

Out[185…

	Place	Time	Food	Price($)
0	Home	Lunch	Soup	10
1	Home	Dinner	Rice	20
2	Hotel	Lunch	Soup	30
3	Hotel	Dinner	Chapati	40

In [187…

```
sales=pd.DataFrame([['books','online', 200, 50],['books','retail', 250, 75],
                    ['toys','online', 100, 20],['toys','retail', 140, 30],
                    ['watches','online', 500, 100],['watches','retail', 600, 150],
                    ['computers','online', 1000, 200],['computers','retail', 1200,
                    ['laptops','online', 1100, 400],['laptops','retail', 1400, 500]
                    ['smartphones','online', 600, 200],['smartphones','retail', 800
                    columns=['Items', 'Mode', 'Price', 'Profit'])


sales
```

Out[187…

	Items	Mode	Price	Profit
0	books	online	200	50
1	books	retail	250	75
2	toys	online	100	20
3	toys	retail	140	30
4	watches	online	500	100
5	watches	retail	600	150
6	computers	online	1000	200
7	computers	retail	1200	300
8	laptops	online	1100	400
9	laptops	retail	1400	500
10	smartphones	online	600	200
11	smartphones	retail	800	250

In [189…

```
sales1=sales.set_index(['Items', 'Mode'])

sales1
```

Out[189…

		Price	Profit
Items	**Mode**		
books	**online**	200	50
	retail	250	75
toys	**online**	100	20
	retail	140	30
watches	**online**	500	100
	retail	600	150
computers	**online**	1000	200
	retail	1200	300
laptops	**online**	1100	400
	retail	1400	500
smartphones	**online**	600	200
	retail	800	250

In [191…

```
# View index

sales1.index
```

Out[191…

```
MultiIndex([(      'books', 'online'),
            (      'books', 'retail'),
            (       'toys', 'online'),
            (       'toys', 'retail'),
            (    'watches', 'online'),
            (    'watches', 'retail'),
            (  'computers', 'online'),
            (  'computers', 'retail'),
            (    'laptops', 'online'),
            (    'laptops', 'retail'),
            ('smartphones', 'online'),
            ('smartphones', 'retail')],
           names=['Items', 'Mode'])
```

In [193…

```
# Swap the column  in multiple index

sales2=sales1.swaplevel('Mode', 'Items')

sales2
```

Out[193…

		Price	Profit
Mode	**Items**		
online	**books**	200	50
retail	**books**	250	75
online	**toys**	100	20
retail	**toys**	140	30
online	**watches**	500	100
retail	**watches**	600	150
online	**computers**	1000	200
retail	**computers**	1200	300
online	**laptops**	1100	400
retail	**laptops**	1400	500
online	**smartphones**	600	200
retail	**smartphones**	800	250

In [197…

```
# sort the dataframe df2 by label

data2.sort_index()
```

Out[197…

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_Cit
0	1000004	P00128942	M	46-50	7	B	
1	1000009	P00113442	M	26-35	17	C	
2	1000010	P00288442	F	36-45	1	B	
3	1000010	P00145342	F	36-45	1	B	
4	1000011	P00053842	F	26-35	1	C	
...	...	...	...	...	...	...	
233594	1006036	P00118942	F	26-35	15	B	
233595	1006036	P00254642	F	26-35	15	B	
233596	1006036	P00031842	F	26-35	15	B	
233597	1006037	P00124742	F	46-50	1	C	
233598	1006039	P00316642	F	46-50	0	B	

233599 rows × 11 columns

In [199…

```
data2.sort_values(by=['Product_Category_1'])
```

Out[199…

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_Cit
0	1000004	P00128942	M	46-50	7	B	
149548	1001968	P00016042	M	26-35	11	B	
149540	1001958	P00243942	F	26-35	1	B	
45672	1004318	P00016042	M	26-35	5	B	
149539	1001958	P00244242	F	26-35	1	B	
...	...	...	...	...	...	...	
195953	1001920	P00271442	F	36-45	7	B	
133275	1001196	P00117542	F	18-25	14	C	
105784	1000977	P00117542	M	26-35	2	C	
105922	1001211	P00037442	M	18-25	4	A	
135012	1003823	P00286042	M	55+	7	B	

233599 rows × 11 columns

In [201…

```
data3 = data.copy()

data3.dtypes
```

Out[201…

```
User_ID                         int64
Product_ID                     object
Gender                         object
Age                            object
Occupation                      int64
City_Category                  object
Stay_In_Current_City_Years     object
Marital_Status                  int64
Product_Category_1              int64
Product_Category_2            float64
Product_Category_3            float64
dtype: object
```

In [203…

```
data3['Gender'].describe()
```

Out[203…

```
count     233599
unique         2
top            M
freq      175772
Name: Gender, dtype: object
```

In [205…

```
data3['Age'].describe()
```

Out[205…

```
count     233599
unique         7
top        26-35
freq       93428
Name: Age, dtype: object
```

In [207…

```
data3['City_Category'].describe()
```

Out[207…

```
count     233599
unique         3
top            B
freq       98566
Name: City_Category, dtype: object
```

In [215…

```
data3['Gender'].unique()
```

Out[215…

```
array(['M', 'F'], dtype=object)
```

In [213…

```
data3['Age'].unique()
```

Out[213…

```
array(['46-50', '26-35', '36-45', '18-25', '51-55', '55+', '0-17'],
      dtype=object)
```

In [217…

```
data3['City_Category'].unique()
```

Out[217…

```
array(['B', 'C', 'A'], dtype=object)
```

In [219…

```
data3['Gender'].value_counts()
```

Out[219…

```
Gender
M    175772
F     57827
Name: count, dtype: int64
```

In [223…

```
data3['City_Category'].value_counts()
```

Out[223…

```
City_Category
B    98566
C    72509
A    62524
Name: count, dtype: int64
```

In [225…

```
data3['Gender'].value_counts(ascending=True)
```

Out[225…

```
Gender
F     57827
M    175772
Name: count, dtype: int64
```

In [227…

```
data3['City_Category'].value_counts(ascending=True)
```

Out[227…

```
City_Category
A    62524
C    72509
B    98566
Name: count, dtype: int64
```

In [229…

```
data4 = data.copy()
data4.max(0)
```

Out[229…

```
User_ID                       1006040
Product_ID                   P0099942
Gender                              M
Age                               55+
Occupation                         20
City_Category                       C
Stay_In_Current_City_Years         4+
Marital_Status                      1
Product_Category_1                 18
Product_Category_2               18.0
Product_Category_3               18.0
dtype: object
```

In [231…

```
data4.describe()
```

Out[231…

	User_ID	Occupation	Marital_Status	Product_Category_1	Product_Category_
count	2.335990e+05	233599.000000	233599.000000	233599.000000	233599.00000
mean	1.003029e+06	8.085407	0.410070	5.276542	9.86828
std	1.726505e+03	6.521146	0.491847	3.736380	5.07544
min	1.000001e+06	0.000000	0.000000	1.000000	2.00000
25%	1.001527e+06	2.000000	0.000000	1.000000	5.00000
50%	1.003070e+06	7.000000	0.000000	5.000000	9.00000
75%	1.004477e+06	14.000000	1.000000	8.000000	15.00000
max	1.006040e+06	20.000000	1.000000	18.000000	18.00000

In [247…

```
data5 = data.copy()
data5
```

Out[247…

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_Cit
0	1000004	P00128942	M	46-50	7	B	
1	1000009	P00113442	M	26-35	17	C	
2	1000010	P00288442	F	36-45	1	B	
3	1000010	P00145342	F	36-45	1	B	
4	1000011	P00053842	F	26-35	1	C	
...	...	...	...	...	...	...	
233594	1006036	P00118942	F	26-35	15	B	
233595	1006036	P00254642	F	26-35	15	B	
233596	1006036	P00031842	F	26-35	15	B	
233597	1006037	P00124742	F	46-50	1	C	
233598	1006039	P00316642	F	46-50	0	B	

233599 rows × 11 columns

In [ ]: