

Python Documentation

#important library

import pandas as pd

from datetime import date

#Original Data exploration

Original_data=pd.read_csv('Data.csv')

print(original_data)

```

----- ***Start Of Ahmed Matter Part*** -----
   Row ID   Order ID   Order Date   Sub-Category   Product Name   Sales
0         1   CA-2017-152156   08/11/2017   Bookcases   Bush Somerset Collection Bookcase   261.9600
1         2   CA-2017-152156   08/11/2017   Chairs   Hon Deluxe Fabric Upholstered Stacking Chairs,...   731.9400
2         3   CA-2017-138688   12/06/2017   Labels   Self-Adhesive Address Labels for Typewriters b...   14.6200
3         4   US-2016-108966   11/10/2016   Tables   Bretford CR4500 Series Slim Rectangular Table   957.5775
4         5   US-2016-108966   11/10/2016   Storage   Eldon Fold 'N Roll Cart System   22.3680
...     ...     ...     ...     ...     ...
9795     9796   CA-2017-125920   21/05/2017   Binders   Cardinal HOLDit! Binder Insert Strips,Extra St...   3.7980
9796     9797   CA-2016-128608   12/01/2016   Art   BIC Brite Liner Highlighters, Chisel Tip   10.3680
9797     9798   CA-2016-128608   12/01/2016   Phones   GE 30524EE4   235.1880
9798     9799   CA-2016-128608   12/01/2016   Phones   Anker 24W Portable Micro USB Car Charger   26.3760
9799     9800   CA-2016-128608   12/01/2016   Accessories   SanDisk Cruzer 4 GB USB Flash Drive   10.3840

[9800 rows x 18 columns]

```

Display the dataset information

df_infoo=originaldata.info()

print(df_infoo)

```

#   Column      Non-Null Count  Dtype
---  -
0   Row ID      9800 non-null     int64
1   Order ID     9800 non-null     object
2   Order Date   9800 non-null     object
3   Ship Date    9800 non-null     object
4   Ship Mode    9800 non-null     object
5   Customer ID  9800 non-null     object
6   Customer Name 9800 non-null     object
7   Segment      9800 non-null     object
8   Country      9800 non-null     object
9   City         9800 non-null     object
10  State        9800 non-null     object
11  Postal Code   9789 non-null     float64
12  Region       9800 non-null     object
13  Product ID   9800 non-null     object
14  Category     9800 non-null     object
15  Sub-Category 9800 non-null     object
16  Product Name 9800 non-null     object
17  Sales        9800 non-null     float64
dtypes: float64(2), int64(1), object(15)
memory usage: 1.3+ MB
None

```

Display the number of missing values for each column .

df_null=originaldata.isnull().sum()

print(df_null)

```

Row ID      0
Order ID    0
Order Date  0
Ship Date   0
Ship Mode   0
Customer ID 0
Customer Name 0
Segment     0
Country     0
City        0
State       0
Postal Code 11
Region      0
Product ID  0
Category    0
Sub-Category 0
Product Name 0
Sales       0
dtype: int64

```

#display duplicate values

```
df_dubli=originaldata[originaldata.duplicated()]
print('The num of duplicated value is : \n')
print(df_dubli)
```

The num of duplicated value is :

```
Empty DataFrame
Columns: [Row ID, Order ID, Order Date, Ship Date, Ship Mode, Customer ID, Customer Name, Segment, Country, City, State, Postal Code, Region, Product ID, Category, Sub-Category, Product Name, Sales]
Index: []
```

#display Some statistical operations

```
df_summary=originaldata.describe(include="all")
print(df_summary)
```

	Row ID	Order ID	Order Date	Ship Date	...	Category	Sub-Category	Product Name	Sales
count	9800.000000	9800	9800	9800	...	9800	9800	9800	9800.000000
unique	NaN	4922	1230	1326	...	3	17	1849	NaN
top	NaN	CA-2018-100111	05/09/2017	26/09/2018	...	Office Supplies	Binders	Staple envelope	NaN
freq	NaN	14	38	34	...	5909	1492	47	NaN
mean	4900.500000	NaN	NaN	NaN	...	NaN	NaN	NaN	230.769059
std	2829.160653	NaN	NaN	NaN	...	NaN	NaN	NaN	626.651875
min	1.000000	NaN	NaN	NaN	...	NaN	NaN	NaN	0.444000
25%	2450.750000	NaN	NaN	NaN	...	NaN	NaN	NaN	17.248000
50%	4900.500000	NaN	NaN	NaN	...	NaN	NaN	NaN	54.490000
75%	7350.250000	NaN	NaN	NaN	...	NaN	NaN	NaN	210.605000
max	9800.000000	NaN	NaN	NaN	...	NaN	NaN	NaN	22638.480000

[11 rows x 18 columns]

#count of unique value for all col

```
print('The Unique Values For Each Col Is :>> \n',originaldata.nunique())
```

The Unique Values For Each Col Is :>>

```
Row ID      9800
Order ID    4922
Order Date  1230
Ship Date   1326
Ship Mode    4
Customer ID  793
Customer Name 793
Segment      3
Country      1
City         529
State        49
Postal Code  626
Region       4
Product ID   1861
Category     3
Sub-Category 17
Product Name 1849
Sales       5757
dtype: int64
```

#formula to Split date col

```
def seprate_values(col):
    new_cols=(originaldata[col].str.split('/', expand=True).rename(columns=lambda x: f"{col}_{x+1}"))
    return new_cols
#split order date to get year col
df_ordrdate=seprate_values('Order Date')
print(df_ordrdate)
#split Ship date to get year col
df_shipdate=seprate_values('Ship Date')
print(df_shipdate)
```

```

-----
      Order Date_1 Order Date_2 Order Date_3
0           08           11           2017
1           08           11           2017
2           12           06           2017
3           11           10           2016
4           11           10           2016
...
9795        21           05           2017
9796        12           01           2016
9797        12           01           2016
9798        12           01           2016
9799        12           01           2016

```

[9800 rows x 3 columns]

```

-----
      Ship Date_1 Ship Date_2 Ship Date_3
0           11           11           2017
1           11           11           2017
2           16           06           2017
3           18           10           2016
4           18           10           2016
...
9795        28           05           2017
9796        17           01           2016
9797        17           01           2016
9798        17           01           2016
9799        17           01           2016

```

[9800 rows x 3 columns]

#unique ordr years

```

uni_years=df_ordrdte['Order Date_3'].unique()
print('our data talks about ',df_ordrdte['Order Date_3'].nunique(),' order years : >> ',uni_years)

```

```

-----
our data talks about 4 order years : >> ['2017' '2016' '2015' '2018']
-----

```

#unique Ship years

```

uni_shpyears=df_shipdte['Ship Date_3'].unique()
print('our data talks about ',df_shipdte['Ship Date_3'].nunique(),' ship years : >> ',uni_shpyears)

```

```

-----
our data talks about 5 ship years : >> ['2017' '2016' '2015' '2018' '2019']
-----

```

#unique customers

```

uni_cust=originaldata['Customer ID'].nunique()
print('we have ',uni_cust,'Customers')

```

```

-----
we have 793 Customers
-----

```

#unique Orders(Invoices)

```

uni_invoice=originaldata['Order ID'].nunique()
print('we have ',uni_invoice,'Invoice(Order ID)')

```

```

-----
we have 4922 Invoice(Order ID)
-----

```

#unique ship methods

```

uni_shipmode=originaldata['Ship Mode'].unique()
print('we have ',originaldata['Ship Mode'].nunique(),'ship modes : >> ',uni_shipmode)

```

```

-----
we have 4 ship modes : >> ['Second Class' 'Standard Class' 'First Class' 'Same Day']
-----

```

#unique Customer segmentation

```

uni_custseg=originaldata['Segment'].unique()
print('we have ',originaldata['Segment'].nunique(),' Customer segmentations : >> ',uni_custseg)

```

```

-----
we have 3 Customer segmentations : >> ['Consumer' 'Corporate' 'Home Office']
-----

```

#unique City

```
uni_city=originaldata['City'].nunique()  
print('we have ',uni_city,' City')
```

```
we have 529 City
```

#unique Country

```
uni_country=originaldata['Country'].unique()  
print('we have ',originaldata['Country'].nunique(), ' Country : >> ',uni_country)
```

```
we have 1 Country : >> ['United States']
```

#unique state

```
uni_state=originaldata['State'].nunique()  
print('we have ',uni_state,' state')
```

```
we have 49 state
```

#unique Region

```
uni_region=originaldata['Region'].unique()  
print('we have ',originaldata['Region'].nunique(), ' region : >> ',uni_region)
```

```
we have 4 region : >> ['South' 'West' 'Central' 'East']
```

#unique category

```
uni_category=originaldata['Category'].unique()  
print('we have ',originaldata['Category'].nunique(), ' Category : >> ',uni_category)
```

```
we have 3 Category : >> ['Furniture' 'Office Supplies' 'Technology']
```

#unique subcategory

```
uni_subcat=originaldata['Sub-Category'].nunique()  
print('we have ',uni_subcat,' Sub Category')
```

```
we have 17 Sub Category
```

#Common Category

```
print('The most common category is : >>',originaldata['Category'].mode())
```

#Common Sub Category

```
print('The most common sub category is : >>',originaldata['Sub-Category'].mode())
```

#Most Valuable Customer Segmentation

```
print('The most common Customer Segment is : >>',originaldata['Segment'].mode())
```

#Most Demand Product

```
print('The Highest Demand Product is : >>',originaldata['Product Name'].mode())
```

#Most Common City

```
print('The most common City is : >>',originaldata['City'].mode())
```

#Most Common Ship mode

```
print('The most common Ship Mode is : >>',originaldata['Ship Mode'].mode())
```

#Highest Value Customer

```
print('The Highest Value Customer is : >>',originaldata['Customer Name'].mode())
```

#Total Sales

```
print('The Total Sales equal >>','$',f"{{(originaldata['Sales'].sum()).round():.2f}}")
```

```
-----
The most common category is : >> 0    Office Supplies
Name: Category, dtype: object
-----

The most common sub category is : >> 0    Binders
Name: Sub-Category, dtype: object
-----

The most common Customer Segment is : >> 0    Consumer
Name: Segment, dtype: object
-----

The Highest Demand Product is : >> 0    Staple envelope
Name: Product Name, dtype: object
-----

The most common City is : >> 0    New York City
Name: City, dtype: object
-----

The most common Ship Mode is : >> 0    Standard Class
Name: Ship Mode, dtype: object
-----

The Highest Value Customer is : >> 0    William Brown
Name: Customer Name, dtype: object
-----

The Total Sales equal >> $ 2,261,537.00
-----
```

#Uni Prod Id & Prod Name

```
uni_prodid=originaldata['Product ID'].unique()
uni_prodnam=originaldata['Product Name'].unique()
print('>>>Note<<< :\n','We Have', uni_prodid, 'Product Id With', uni_prodnam,'Product Name' '--
Seems To Be Proplem--\n')
print("--*20,'***End Of Ahmed Matter Part***' "--*20)
```

```
-----
>>>Note<<< :
We Have 1861 Product Id With 1849 Product Name --Seems To Be Proplem--
-----
***End Of Ahmed Matter Part*** -----
```

```
----- ****APPLIED STEPS**** -----
--
1-Explore Data Frame Using Pandas
2-Display the dataset information
3-Display the number of missing values for each column(Check Null Values)--->Postal Code Only 11 Null (Unused Col)
4-Display duplicate values(Check For Duplicates)----> No Duplicates Found
5-Display Some statistical operations
6-Display count of unique value for all Used columns & Show The Unique Value
7-Add formula to Split date col To Display Unique Years
8-Display The most common category
9-Display The most common Sub category
10-Display The most Common City
11-Display The Highest Demand Product Name
12-Display The Total Sales
13-Display The most common Customer Segment
14-Display The most common Ship Mode
```

Excel Documentation

-Converting CSV To XLSX

-Split Data with Text to Column Method

-Adding 6 Calculated Column:

Ship Time

=[@[Ship Date]]-[@[Order Date]]

Avg Ship Time by Ship Mode

=ROUND(AVERAGEIF([Ship Mode];[@[Ship Mode]];[Ship Time]);0)

Ship Status

=IF([@[Ship Time]]>[@[Avg Ship Time Per Ship Mode]];"Above Average";"Normal")

Num Of product Per 1 Order

=COUNTIF([Order ID];[@[Order ID]])

Weekday Name

=TEXT([@[Order Date]];"dddd")

Days above Avg

=IF([@[Ship Time]]>[@[Avg Ship Time Per Ship Mode]];[@[Ship Time]]-[@[Avg Ship Time Per Ship Mode]];0)

Ship Time	Num of Prod per order	Avg Ship Time Per Ship Mode	Status	WeekDay Name	Days After Average
3	2	3	Normal	الأربعاء	0
3	2	3	Normal	الأربعاء	0
4	1	3	Above Average	الاثنين	1
7	2	5	Above Average	الثلاثاء	2
7	2	5	Above Average	الثلاثاء	2

-Adding Unique Orders Table & It's related Data.

Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Region	Ship Time	Num of Prod per order	Avg Ship Time Per Ship Mode	Status	WeekDay Name	Days After Average	Sales
CA-2015-100006	07/09/2015	13/09/2015	Standard Class	DK-13375	Dennis Kane	Consumer	United States	New York City	New York	East	6	1	5	Above Average	الاثنين	1	378
CA-2015-100090	08/07/2015	12/07/2015	Standard Class	EB-13705	Ed Braxton	Corporate	United States	San Francisco	California	West	4	2	5	Normal	الأربعاء	0	699.2
CA-2015-100293	14/03/2015	18/03/2015	Standard Class	NF-18475	Neil Franz	Home Office	United States	Jacksonville	Florida	South	4	1	5	Normal	الجمعة	0	91.06
CA-2015-100328	28/01/2015	03/02/2015	Standard Class	JC-15340	Jasper Cacioppo	Consumer	United States	New York City	New York	East	6	1	5	Above Average	الأربعاء	1	3.928
CA-2015-100363	08/04/2015	15/04/2015	Standard Class	JM-15655	Jim Mitthum	Corporate	United States	Glendale	Arizona	West	7	2	5	Above Average	الأربعاء	2	21.38

-Start Exploring & Analyzing Data Using Pivot tables.

-Make Presentation.