Data Analytics Problems on Sales_Data

1. Import necessary libraries for sales Data Analysis

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

In [ ]:
```
# 2.Concatenate each month sale Data
(https://github.com/svkarthik86/Assignment/tree/main/Sales_Data) into one dataframe and save the dataframe to annual_sale.csv
```

In [ ]:
```
# hint:
# step1:download the Sale_date folder contain 12 month file from github link to local current woring directory
# setp2: import os module , use os.listdir() get all file
#step3: create empty DataFrame as annual_sale
#step4: using for loop to read file locotain contain file and use pd.concat(ignore_index=True) function
#       to concatenate all the file into one dataFrame
# as annual_sale
# at last store DataFrame annual_sale to annual_sale.csv
#annual_sale.to_csv("annual_sale.csv", index=False)
```

In [4]:
```
annual_sale=pd.read_csv("https://raw.githubusercontent.com/svkarthik86/Assignment/main/Sales_Data/Sales_April_2019.csv")
annual_sale.head(5)
```

Out[4]:

	Order ID	Product	Quantity Ordered	Price Each	Order Date	Purchase Address
0	176558	USB-C Charging Cable	2	11.95	04/19/19 08:46	917 1st St, Dallas, TX 75001
1	NaN	NaN	NaN	NaN	NaN	NaN
2	176559	Bose SoundSport Headphones	1	99.99	04/07/19 22:30	682 Chestnut St, Boston, MA 02215
3	176560	Google Phone	1	600	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001
4	176560	Wired Headphones	1	11.99	04/12/19 14:38	669 Spruce St, Los Angeles, CA 90001

In [ ]:
```
4.show the metadata information of the annual_sale data frame and check if data is missing or not,
if yes How many data are missing.
```

In [5]:
```
print(annual_sale.info())
annual_sale.isna().sum().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18383 entries, 0 to 18382
Data columns (total 6 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   Order ID          18324 non-null  object
 1   Product           18324 non-null  object
 2   Quantity Ordered  18324 non-null  object
 3   Price Each        18324 non-null  object
 4   Order Date        18324 non-null  object
 5   Purchase Address  18324 non-null  object
dtypes: object(6)
memory usage: 861.8+ KB
None
```

Out[5]:
```
354
```

In [6]:
```
annual_sale.Product.memory_usage()
```

Out[6]:
```
147192
```

In [7]:
```
annual_sale.Product.value_counts()[:1]
```

Out[7]:
```
Lightning Charging Cable    2201
Name: Product, dtype: int64
```

In [ ]:
```
5.Clean up the data!
5.1 Verify all the column names are in a valid format, if any space between the column name then rename the column names
example: Product ID as Product_ID
```

In [ ]:
```
annual_sale.columns=[i.replace(" ","_") for i in annual_sale.columns]
```

In [ ]:
```
5.2 check the isnan is present in dataframe, if there is nan is present remove the nan using dropna() function
```

In [ ]:
```
annual_sale.dropna(inplace=True)
```

In [ ]:
```
5.3 Find The duplicated data present in the data frame, and remove the duplicated data from the dataframe
```

In [ ]:
```
annual_sale=annual_sale[~(annual_sale.Price_Each=="Price Each")]
```

In [ ]:
```
annual_sale.drop_duplicates(inplace=True)
```

In [ ]:
```
6.memory_usage
check memory_usage of Product column , type cast the Product column as "category" type and then check memmory_usage compare
the memory utialization and how much percentage effectively reduce the storage space?
```

In [8]:
```
annual_sale.Product.memory_usage()
```

Out[8]:
```
147192
```

In [9]:
```
annual_sale.Product=annual_sale.Product.astype("category")
annual_sale.Product.memory_usage()
```

Out[9]:
```
19227
```

In [ ]:
```
7.Create and add a new column
Add month column to annual_sale DataFrame object from 'Order Date' column using Series.str method
annual_sale["month"]
```

In [ ]:
```
annual_sale['month']=annual_sale.Order_Date.str[:2]
```

In [ ]:
```
7.1 converts the datatype of month column as int using astype('int32')
```

In [ ]:
```
annual_sale.month=annual_sale.month.astype('int32')
```

In [ ]:
```
7.2 Add sale column to annual_sale DataFrame object using following calculation
sales = Quantity_Ordered * Price_Each
```

In [ ]:
```
annual_sale['sale']=annual_sale.Quantity_Ordered.astype(float)*annual_sale.Price_Each.astype(float)
```

In [ ]:
```
8. Find out the day,in which sales is high? using gruop by agg function
```

In [ ]:
```
annual_sale.groupby(annual_sale.Order_Date.str[:8])["sale"].sum().sort_values(ascending=False)[:1]
```

In [ ]:
```
9. What Product is most frequently purchased over the all period?
```

In [14]:
```
annual_sale.Product.value_counts()[:1]
```

Out[14]:
```
Lightning Charging Cable    2201
Name: Product, dtype: int64
```

In [ ]:
```
10. List out the Product price above 200$
```

In [22]:
```
annual_sale[annual_sale.Price_Each.astype(float)>200].loc[:,["Product","Price_Each"]]
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [22], in <cell line: 1>()
----> 1 annual_sale[annual_sale.Price_Each.astype(float)>200].loc[:,["Product","Price_Each"]]

File ~\anaconda3\lib\site-packages\pandas\core\generic.py:5575, in NDFrame.__getattr__(self, name)
   5568 if (
   5569     name not in self._internal_names_set
   5570     and name not in self._metadata
   5571     and name not in self._accessors
   5572     and self._info_axis._can_hold_identifiers_and_holds_name(name)
   5573 ):
   5574     return self[name]
-> 5575 return object.__getattribute__(self, name)

AttributeError: 'DataFrame' object has no attribute 'Price_Each'
```

In [ ]:
```
```