sales_data_analysis

August 28, 2020

[2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import os as os
import seaborn as sns
%matplotlib inline
```

0.1 Upload data after cleaning it to start analysis.

[3]:
```
df=pd.read_csv('C:/Users/ENTER/Desktop/my work now/all data before merge/
 ↪all_data_cleaning.csv')
df.head(1)
```

[3]:
```
   Order ID               Product  Quantity Ordered  Price Each  \
0    176558  USB-C Charging Cable                 2       11.95

       Order Date              Purchase Address Purchase city  Month  \
0  04/19/19 08:46  917 1st St, Dallas, TX 75001        Dallas      4

   Total Price
0         23.9
```

Analyis and plotting

0.1.1 analysis for purchasing time [each month]

(top month , chart for all months according to purchasing)

0.1.2 analysis for purchaser

(top ten order id ,in which city)

0.1.3 analysis for the products

(top priceful products , the most order products)

0.1.4 place

(top three city for purchasing)

(FIRST):Analysis for months.

top month with it's orders amount,and all other month

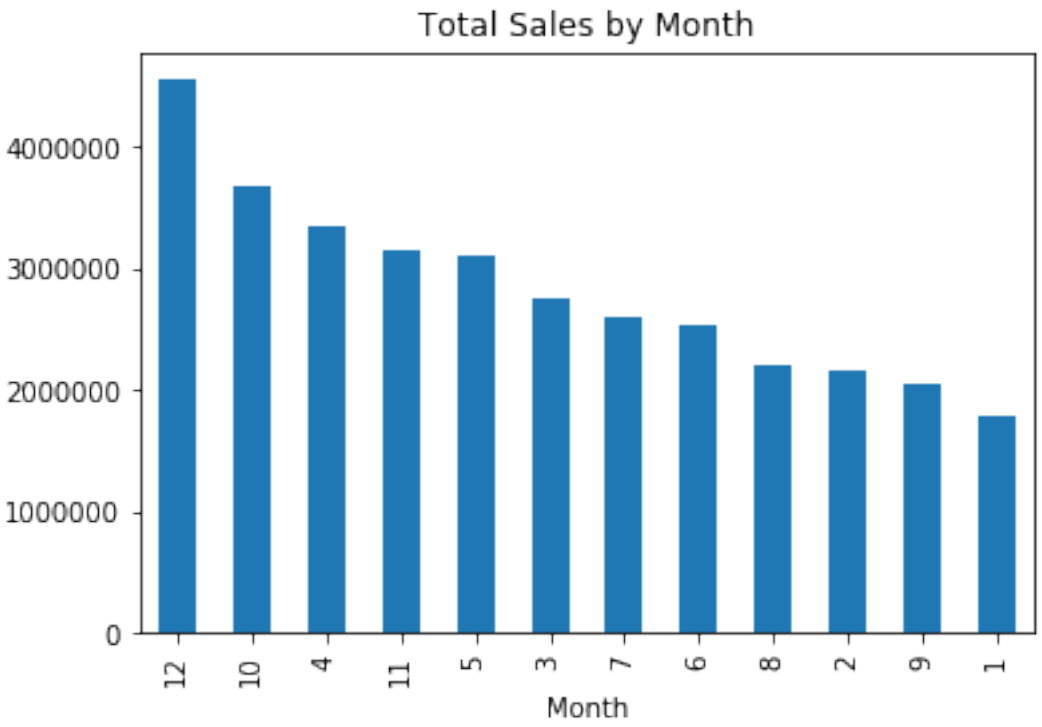
[4]:
```
#top orders in month
table=df[['Month','Quantity Ordered']]
top_month_ordered=table.groupby('Month').sum()
top_month_ordered.sort_values('Quantity Ordered',ascending=False).
 ↪plot(kind='bar',legend=None,title='top month\'s orders')
```

[4]: <matplotlib.axes._subplots.AxesSubplot at 0x3c1dc10>

top month with it's total price,and all other month

[6]:
```
table=df[['Month','Total Price']]
top_pricful_month=table.groupby('Month').sum()
top_pricful_month.sort_values('Total Price',ascending=False).
 ↪plot(kind='bar',legend=None,title="Total Sales by Month")
#top_pricful_month.set_xlabel("Months")
#top_pricful_month.set_ylabel("Amounts")
```

[6]: <matplotlib.axes._subplots.AxesSubplot at 0x8b405d0>



Second : Analysis for purchaser.

top ten purchasers ,and their cityes

[21]:
```
purchaser=df[['Order ID','Purchase city','Total Price']]
purchaser.groupby(['Order ID','Total Price','Purchase city']).sum()
purchaser=purchaser.sort_values('Total Price',ascending=False)
puchaser_df=pd.DataFrame(purchaser)
puchaser_df.head(10)
```

[21]:

	Order ID	Purchase city	Total Price
90015	210292	San Francisco	3400.00
5074	181544	New York City	3400.00
4587	181069	San Francisco	3400.00
124411	200528	Boston	3400.00
74697	149611	New York City	1999.98
171188	278637	Boston	1999.98
174701	251453	Dallas	1700.00
31080	296997	San Francisco	1700.00
146699	291141	San Francisco	1700.00
31050	296969	San Francisco	1700.00

1 Thrird : Analysis for the products.

1.0.1 top priceful products

[30]:
```
products=df[['Product','Price Each']]
#products.drop_duplicates()
#products['Product'].unique()
products_df=pd.DataFrame(products)
products_df=products_df.sort_values('Price Each',ascending=False)
products_df.drop_duplicates()
```

[30]:

	Product	Price Each
152201	Macbook Pro Laptop	1700.00
59237	ThinkPad Laptop	999.99
18902	iPhone	700.00
66716	Google Phone	600.00
139638	LG Dryer	600.00
37272	LG Washing Machine	600.00
49709	Vareebadd Phone	400.00
147456	27in 4K Gaming Monitor	389.99
46746	34in Ultrawide Monitor	379.99
47023	Flatscreen TV	300.00
133894	Apple Airpods Headphones	150.00
62757	27in FHD Monitor	149.99
117753	20in Monitor	109.99
122869	Bose SoundSport Headphones	99.99
104619	Lightning Charging Cable	14.95
22241	Wired Headphones	11.99
161587	USB-C Charging Cable	11.95
45017	AA Batteries (4-pack)	3.84
87828	AAA Batteries (4-pack)	2.99

1.0.2 top ordered products

[32]:
```
top_ordered=df[['Product','Quantity Ordered','Total Price']]
top_ordered=top_ordered.sort_values('Quantity Ordered',ascending=False)
top_ordered_df=pd.DataFrame(top_ordered)
top_ordered_df=top_ordered_df.drop_duplicates()
top_ordered_df.head(10)
```

[32]:

	Product	Quantity Ordered	Total Price
90844	AAA Batteries (4-pack)	9	26.91
171116	AAA Batteries (4-pack)	8	23.92
619	AAA Batteries (4-pack)	7	20.93
33845	AA Batteries (4-pack)	7	26.88
85932	AAA Batteries (4-pack)	6	17.94
77076	AA Batteries (4-pack)	6	23.04

```
10656     USB-C Charging Cable                6        71.70
95013   AAA Batteries (4-pack)                5        14.95
3114     AA Batteries (4-pack)                5        19.20
79610     USB-C Charging Cable                5        59.75
```

2 NUMBER four: places Analysis

2.0.1 top 3 citeyes in purchasing

[55]:
```
citeyes=df[['Purchase city','Quantity Ordered','Total Price']]
citeyes=citeyes.groupby(['Purchase city'])['Quantity Ordered','Total Price'].sum()
citeyes=citeyes.sort_values('Total Price',ascending=False)
citeyes_df=pd.DataFrame(citeyes)
citeyes_df
citeyes_df=citeyes_df.drop_duplicates()
citeyes_df.head(3)
```

```
C:\Users\ENTER\anaconda3\lib\site-packages\ipykernel_launcher.py:3:
FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of
keys) will be deprecated, use a list instead.
  This is separate from the ipykernel package so we can avoid doing imports
until
```

[55]:
```
                Quantity Ordered   Total Price
Purchase city
 San Francisco             49363  8.124121e+06
 Los Angeles               32722  5.354040e+06
 New York City             27470  4.581659e+06
```

2.1 top products in those citeyes with Quantity ordered

[75]:
```
citeyes_products=df[['Product','Purchase city','Quantity Ordered']]
citeyes_products=citeyes_products.groupby(['Purchase city','Product'])['Quantity␣
↪Ordered'].sum()
citeyes_products=pd.DataFrame(citeyes_products)
citeyes_products
```

[75]:
```
                                      Quantity Ordered
Purchase city Product
 Atlanta      20in Monitor                         340
              27in 4K Gaming Monitor               485
              27in FHD Monitor                     576
              34in Ultrawide Monitor               468
              AA Batteries (4-pack)               2169
…                                                  …
 Seattle      ThinkPad Laptop                      326
```

USB-C Charging Cable	1834
Vareebadd Phone	175
Wired Headphones	1617
iPhone	529

[171 rows x 1 columns]

3