Project

Exploratory Data Analysis in Python for Retail

In [129…
```
#Python version
from platform import python_version
print("Python version is: ", python_version())
```

```
Python version is:  3.9.13
```

In [130…
```
#Import packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
```

In [131…
```
#import dataset
dataset = pd.read_csv("dataset.csv")
```

In [132…
```
#Checking the first 5 columns of the dataset
dataset.head()
```

Out[132]:

	ID_Pedido	Data_Pedido	ID_Cliente	Segmento	Pais	Cidade	Estado	ID_Produto	Cate
0	CA-2017-152156	08/11/2017	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-BO-10001798	Furr
1	CA-2017-152156	08/11/2017	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-CH-10000454	Furr
2	CA-2017-138688	12/06/2017	DV-13045	Corporate	United States	Los Angeles	California	OFF-LA-10000240	Sup
3	US-2016-108966	11/10/2016	SO-20335	Consumer	United States	Fort Lauderdale	Florida	FUR-TA-10000577	Furr
4	US-2016-108966	11/10/2016	SO-20335	Consumer	United States	Fort Lauderdale	Florida	OFF-ST-10000760	Sup

In [133…
```
#Checking the last 5 columns of the dataset
dataset.tail()
```

Out[133]:

	ID_Pedido	Data_Pedido	ID_Cliente	Segmento	Pais	Cidade	Estado	ID_Produto
9695	CA-2018-154116	15/12/2018	KM-16660	Consumer	United States	Inglewood	California	OFF-PA-10004569
9696	CA-2018-154116	15/12/2018	KM-16660	Consumer	United States	Inglewood	California	OFF-AP-10000027
9697	CA-2018-154116	15/12/2018	KM-16660	Consumer	United States	Inglewood	California	TEC-PH-10000675
9698	CA-2017-105291	30/10/2017	SP-20920	Consumer	United States	San Luis Obispo	California	OFF-FA-10003059
9699	CA-2018-147032	31/07/2018	LB-16795	Home Office	United States	Wilmington	Delaware	OFF-PA-10003256

In [134…
```
#Cheacking shape of dataset
dataset.shape
```

Out[134]:
```
(9700, 11)
```

Exploratory Analysis

In [135…
```
#checking coloumns of dataset
dataset.columns
```

Out[135]:
```
Index(['ID_Pedido', 'Data_Pedido', 'ID_Cliente', 'Segmento', 'Pais', 'Cidade',
       'Estado', 'ID_Produto', 'Categoria', 'SubCategoria', 'Valor_Venda'],
      dtype='object')
```

In [136…
```
#Checking types of data
dataset.dtypes
```

Out[136]:
```
ID_Pedido        object
Data_Pedido      object
ID_Cliente       object
Segmento         object
Pais             object
Cidade           object
Estado           object
ID_Produto       object
Categoria        object
SubCategoria     object
Valor_Venda     float64
dtype: object
```

In [137…
```
#Statistical summary of the valor venda column
dataset["Valor_Venda"].describe()
```

Out[137]:
```
count     9700.000000
mean       230.469892
std        627.504252
min          0.444000
25%         17.248000
50%         54.272000
75%        209.932500
max      22638.480000
Name: Valor_Venda, dtype: float64
```

In [138…
```
#Checking duplicate data
dataset[dataset.duplicated()]
```

Out[138]:

ID_Pedido	Data_Pedido	ID_Cliente	Segmento	Pais	Cidade	Estado	ID_Produto	Categoria	Su

In [139…
```
#Null value
dataset.isnull().sum()
```

Out[139]:
```
ID_Pedido       0
Data_Pedido     0
ID_Cliente      0
Segmento        0
Pais            0
Cidade          0
Estado          0
ID_Produto      0
Categoria       0
SubCategoria    0
Valor_Venda     0
dtype: int64
```

In [140…
```
dataset.head()
```

Out[140]:

	ID_Pedido	Data_Pedido	ID_Cliente	Segmento	Pais	Cidade	Estado	ID_Produto	Cate
0	CA-2017-152156	08/11/2017	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-BO-10001798	Furr
1	CA-2017-152156	08/11/2017	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-CH-10000454	Furr
2	CA-2017-138688	12/06/2017	DV-13045	Corporate	United States	Los Angeles	California	OFF-LA-10000240	Sup
3	US-2016-108966	11/10/2016	SO-20335	Consumer	United States	Fort Lauderdale	Florida	FUR-TA-10000577	Furr
4	US-2016-108966	11/10/2016	SO-20335	Consumer	United States	Fort Lauderdale	Florida	OFF-ST-10000760	Sup

1- City with the highest sales volume in the "office supplies" category

In [141…
```
#Filter the dataframe by categorizing "Office Supplies"
categorize = dataset[dataset["Categoria"] == "Office Supplies"]
```

In [142…
```
categorize.head()
```

Out[142]:

	ID_Pedido	Data_Pedido	ID_Cliente	Segmento	Pais	Cidade	Estado	ID_Produto	Cate
2	CA-2017-138688	12/06/2017	DV-13045	Corporate	United States	Los Angeles	California	OFF-LA-10000240	Sup
4	US-2016-108966	11/10/2016	SO-20335	Consumer	United States	Fort Lauderdale	Florida	OFF-ST-10000760	Sup
6	CA-2015-115812	09/06/2015	BH-11710	Consumer	United States	Los Angeles	California	OFF-AR-10002833	Sup
8	CA-2015-115812	09/06/2015	BH-11710	Consumer	United States	Los Angeles	California	OFF-BI-10003910	Sup
9	CA-2015-115812	09/06/2015	BH-11710	Consumer	United States	Los Angeles	California	OFF-AP-10002892	Sup

In [143…
```
#Group by city and calculate total of Valor_venda
total_sales_volume = categorize.groupby("Cidade")["Valor_Venda"].sum()
```

In [144…
```
#City with the highest sales volume
City_Highest_Sales_Volume = total_sales_volume.idxmax()
print("The city with the highest sales volume in the office supplies category: ", C
```

```
The city with the highest sales volume in the office supplies category:  New York
City
```

In [145…
```
#Checking the result
checking_result = total_sales_volume.sort_values(ascending = False)
print(checking_result)
```

```
Cidade
New York City    68362.814
Los Angeles      47098.100
San Francisco    41771.198
Seattle          34856.878
Philadelphia     29313.687
                    ...
Ormond Beach         2.808
Pensacola            2.214
Jupiter              2.064
Elyria               1.824
Abilene              1.392
Name: Valor_Venda, Length: 480, dtype: float64
```

2 Total sale by order date

In [146…
```
#Group by Data_Pedido of Valor_venda
total_sale_by_Data_Pedido = dataset.groupby("Data_Pedido")["Valor_Venda"].sum()
print(total_sale_by_Data_Pedido)
```

```
Data_Pedido
01/01/2018    1481.8280
01/02/2015     468.9000
01/02/2017     161.9700
01/03/2015    2203.1510
01/03/2016    1642.1744
                 ...
31/10/2017    2346.5790
31/10/2018     523.9280
31/12/2015    5253.2700
31/12/2016    1381.3440
31/12/2017     731.7680
Name: Valor_Venda, Length: 1226, dtype: float64
```

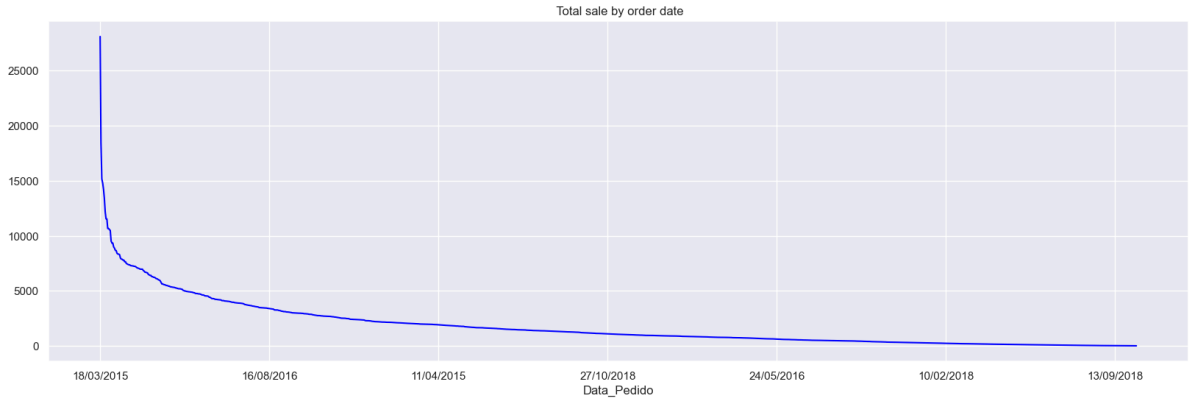
In [147…
```
#Data_Pedido with highest sale volume
Data_Pedido_highest_sale = total_sale_by_Data_Pedido.idxmax()
print(Data_Pedido_highest_sale)
```

```
18/03/2015
```

In [148…
```
#Checking the result
result = total_sale_by_Data_Pedido.sort_values(ascending = False)
print(result)
```

```
Data_Pedido
18/03/2015    28106.716
02/10/2017    18452.972
22/10/2018    15158.877
23/03/2018    14816.068
08/09/2015    14228.428
                 ...
01/10/2015        4.710
24/06/2015        4.272
28/01/2015        3.928
12/07/2018        3.816
19/07/2016        2.025
Name: Valor_Venda, Length: 1226, dtype: float64
```

In [149…
```
#Plot
plt.figure(figsize = (20,6))
result.plot(x = "Data_Pedido", y = "Valor_Venda", color = "blue")
plt.title(" Total sale by order date")
plt.show()
```



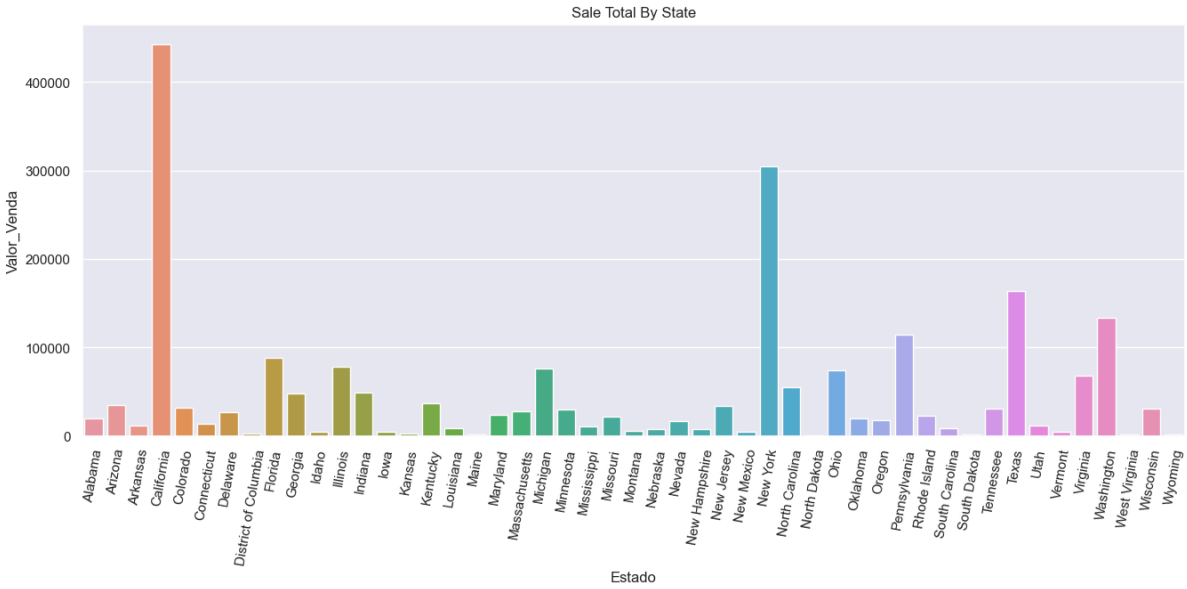
3 What is the total sales by state?

In [150…
```
#Categorizing by state
Categorizing_by_state = dataset.groupby("Estado")["Valor_Venda"].sum().reset_index(
print(Categorizing_by_state)
```

	Estado	Valor_Venda
0	Alabama	19510.6400
1	Arizona	35272.6570
2	Arkansas	11673.8300
3	California	442927.0975
4	Colorado	31841.5980
5	Connecticut	13366.7370
6	Delaware	26452.5890
7	District of Columbia	2865.0200
8	Florida	88043.7000
9	Georgia	48083.1600
10	Idaho	4292.5160
11	Illinois	78109.9270
12	Indiana	48718.4000
13	Iowa	4443.5600
14	Kansas	2914.3100
15	Kentucky	36409.5800
16	Louisiana	9131.0500
17	Maine	1270.5300
18	Maryland	23705.5230
19	Massachusetts	27363.2640
20	Michigan	76081.1740
21	Minnesota	29863.1500
22	Mississippi	10771.3400
23	Missouri	22205.1500
24	Montana	5589.3520
25	Nebraska	7194.9500
26	Nevada	16729.1020
27	New Hampshire	7132.5440
28	New Jersey	34265.7120
29	New Mexico	4783.5220
30	New York	304536.4010
31	North Carolina	55165.9640
32	North Dakota	919.9100
33	Ohio	74277.8020
34	Oklahoma	19683.3900
35	Oregon	17284.4620
36	Pennsylvania	114411.6800
37	Rhode Island	22525.0260
38	South Carolina	8481.7100
39	South Dakota	1315.5600
40	Tennessee	30661.8730
41	Texas	163549.8602
42	Utah	11220.0560
43	Vermont	4524.4700
44	Virginia	68194.6700
45	Washington	133826.0060
46	West Virginia	1209.8240
47	Wisconsin	31154.4700
48	Wyoming	1603.1360

In [151…

```
#Plot
plt.figure(figsize = (16,6))
sns.barplot(data = Categorizing_by_state, x = "Estado", y = "Valor_Venda")
plt.title("Sale Total By State")
plt.xticks(rotation = 80)
plt.show()
```



4 Which 10 cities have the highest total sales?

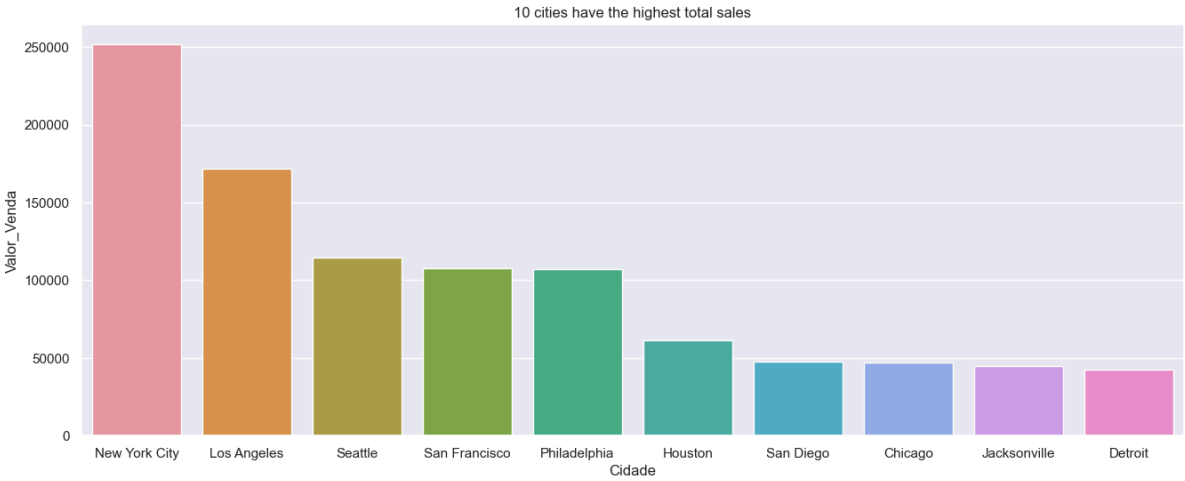
In [152…
```
#Categorizing by cities
cities_with_highest_total_sales = dataset.groupby("Cidade")["Valor_Venda"].sum().re
result = cities_with_highest_total_sales.sort_values(by = 'Valor_Venda', ascending
```

In [153…
```
result.head(10)
```

Out[153]:

	Cidade	Valor_Venda
327	New York City	251749.2190
265	Los Angeles	171654.6330
450	Seattle	114725.4780
436	San Francisco	107489.9520
372	Philadelphia	107197.8030
207	Houston	61590.1868
435	San Diego	47458.3790
80	Chicago	46974.3430
216	Jacksonville	44713.1830
123	Detroit	42446.9440

In [154…
```
plt.figure(figsize = (16, 6))
sns.set_palette('coolwarm')
sns.barplot(data = result.head(10), x = 'Cidade', y = 'Valor_Venda')
plt.title("10 cities have the highest total sales")
plt.show()
```



5 Which segment had the highest total sales

In [155...
```
#Categorizing
categorizing_by_segment = dataset.groupby('Segmento')['Valor_Venda'].sum().reset_ir
```

In [156...
```
result_categorizing_by_segment = categorizing_by_segment.sort_values(by = 'Valor_Ve
```

In [157...
```
result_categorizing_by_segment
```

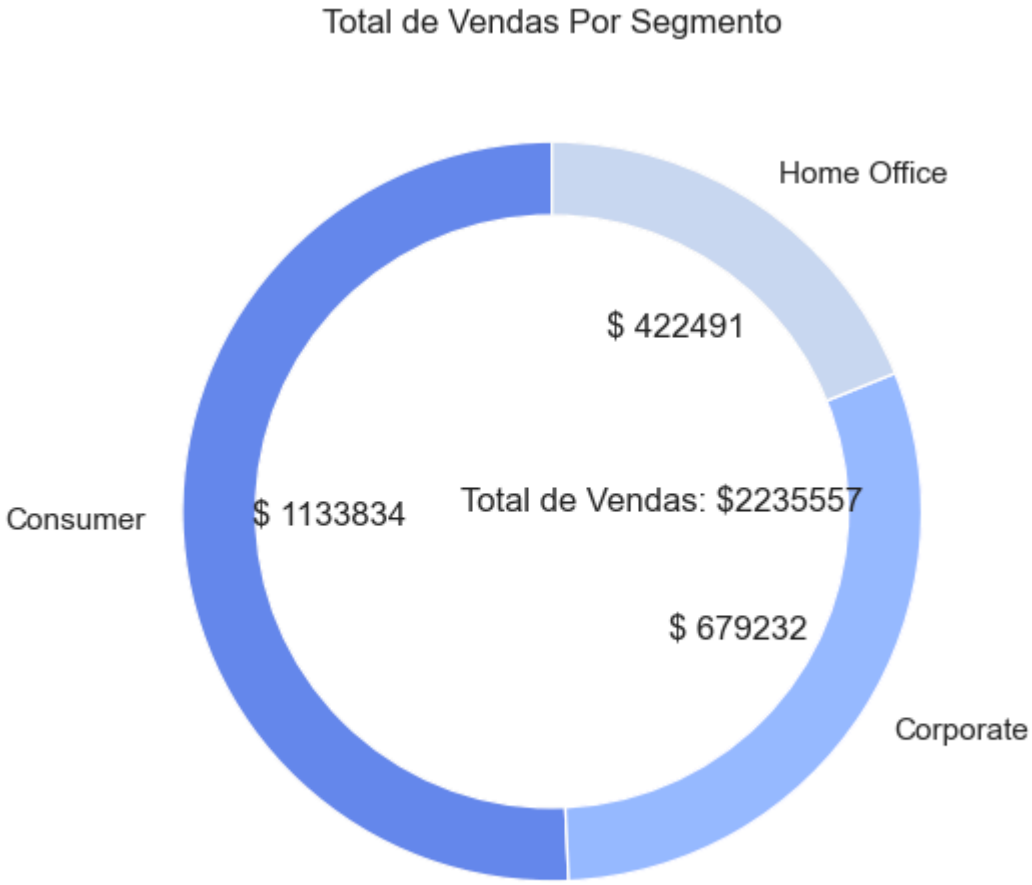
Out[157]:

	Segmento	Valor_Venda
0	Consumer	1.133834e+06
1	Corporate	6.792322e+05
2	Home Office	4.224914e+05

In [158...
```
#Covert datas to absolute value
#Function to convert data to absolute value
def autopct_format(values):
    def my_format(pct):
        total = sum(values)
        val = int(round(pct*total/100.0))
        return '$ {v:d}'.format(v=val)
    return my_format
```

In [159...
```
#Plot using Pizza Graphic

#Figure size
plt.figure(figsize = (16,6))

#Pie Chart
plt.pie(result_categorizing_by_segment["Valor_Venda"],
        labels = result_categorizing_by_segment["Segmento"],
        autopct = autopct_format(result_categorizing_by_segment["Valor_Venda"])
       ,startangle = 90)

#Clean inside of the circle
centre_circle = plt.Circle((0,0), 0.80, fc = 'white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

#Labels and anotations
plt.annotate(text = 'Total de Vendas: ' + '$' + str(int(sum(result_categorizing_by_
             xy =(-0.25, 0))
```

```
plt.title("Total de Vendas Por Segmento")
plt.show()
```



6 Total sales by segment and by year

In [160…
```
dataset.head()
```

Out[160]:

	ID_Pedido	Data_Pedido	ID_Cliente	Segmento	Pais	Cidade	Estado	ID_Produto	Cate
0	CA-2017-152156	08/11/2017	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-BO-10001798	Furr
1	CA-2017-152156	08/11/2017	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-CH-10000454	Furr
2	CA-2017-138688	12/06/2017	DV-13045	Corporate	United States	Los Angeles	California	OFF-LA-10000240	Sup
3	US-2016-108966	11/10/2016	SO-20335	Consumer	United States	Fort Lauderdale	Florida	FUR-TA-10000577	Furr
4	US-2016-108966	11/10/2016	SO-20335	Consumer	United States	Fort Lauderdale	Florida	OFF-ST-10000760	Sup

In [161…
```
#Total sales by segment
total_sales_by_segment = dataset.groupby("Segmento")["Valor_Venda"].sum()
total_sales_by_segment
```

Out[161]:
```
Segmento
Consumer       1.133834e+06
Corporate      6.792322e+05
Home Office    4.224914e+05
Name: Valor_Venda, dtype: float64
```

In [162…
```
#Convert date column to datetime
dataset['Data_Pedido'] = pd.to_datetime(dataset["Data_Pedido"], dayfirst = True)
```

In [163…
```
dataset.dtypes
```

Out[163]:
```
ID_Pedido               object
Data_Pedido     datetime64[ns]
ID_Cliente              object
Segmento                object
Pais                    object
Cidade                  object
Estado                  object
ID_Produto              object
Categoria               object
SubCategoria            object
Valor_Venda            float64
dtype: object
```

In [164…
```
#Extract the year by creating a new variable
dataset["Year"] = dataset["Data_Pedido"].dt.year
```

In [165…
```
dataset.head()
```

Out[165]:

	ID_Pedido	Data_Pedido	ID_Cliente	Segmento	Pais	Cidade	Estado	ID_Produto	Cate
0	CA-2017-152156	2017-11-08	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-BO-10001798	Furr
1	CA-2017-152156	2017-11-08	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-CH-10000454	Furr
2	CA-2017-138688	2017-06-12	DV-13045	Corporate	United States	Los Angeles	California	OFF-LA-10000240	Sup
3	US-2016-108966	2016-10-11	SO-20335	Consumer	United States	Fort Lauderdale	Florida	FUR-TA-10000577	Furr
4	US-2016-108966	2016-10-11	SO-20335	Consumer	United States	Fort Lauderdale	Florida	OFF-ST-10000760	Sup

In [166…
```
#Total sales by year
Total_sales_by_year = dataset.groupby("Year")["Valor_Venda"].sum()
Total_sales_by_year
```

Out[166]:
```
Year
2015    470768.6001
2016    454072.5154
2017    595365.9240
2018    715350.9152
Name: Valor_Venda, dtype: float64
```

In [167…
```
#Total sales by segment and year
Total_sales_by_segment_and_year = dataset.groupby(["Segmento", "Year"])["Valor_Venc
Total_sales_by_segment_and_year
```

Out[167]:
```
Segmento     Year
Consumer     2015    256719.9166
             2016    265295.2593
             2017    288459.5572
             2018    323359.6019
Corporate    2015    125819.5957
             2016    114643.1229
             2017    203266.7398
             2018    235502.7284
Home Office  2015     88229.0878
             2016     74134.1332
             2017    103639.6270
             2018    156488.5849
Name: Valor_Venda, dtype: float64
```

The company manager wishes to offer a discount following the rules below:

1) If Sales Value is over 1000, get a 15% discount

2) If Sales Value is under 1000, get a 10% discount

7 How many Sales will receive a 15% discount

In [168…
```
dataset["Discount"] = np.where(dataset["Valor_Venda"] > 1000, 0.15, 0.10)
```

In [169…
```
dataset.head()
```

Out[169]:

	ID_Pedido	Data_Pedido	ID_Cliente	Segmento	Pais	Cidade	Estado	ID_Produto	Cate
0	CA-2017-152156	2017-11-08	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-BO-10001798	Furr
1	CA-2017-152156	2017-11-08	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-CH-10000454	Furr
2	CA-2017-138688	2017-06-12	DV-13045	Corporate	United States	Los Angeles	California	OFF-LA-10000240	Sup
3	US-2016-108966	2016-10-11	SO-20335	Consumer	United States	Fort Lauderdale	Florida	FUR-TA-10000577	Furr
4	US-2016-108966	2016-10-11	SO-20335	Consumer	United States	Fort Lauderdale	Florida	OFF-ST-10000760	Sup

In [170…
```
dataset["Discount"].value_counts()
```

Out[170]:
```
0.10    9243
0.15     457
Name: Discount, dtype: int64
```

In [171…
```
print("So 457 values will receive a 15% discount")
```

```
So 457 values will receive a 15% discount
```

Consider that the company decides to grant a 15% discount on the previous item.

8 What would be the average sales value before and after the discount?

In [172…
```
#All values affected by 15%
before_discount15 = dataset[dataset['Discount'] == 0.15]
before_discount15.head()
```

Out[172]:

	ID_Pedido	Data_Pedido	ID_Cliente	Segmento	Pais	Cidade	Estado	ID_Produto
10	CA-2015-115812	2015-06-09	BH-11710	Consumer	United States	Los Angeles	California	FUR-TA-10001539
24	CA-2016-106320	2016-09-25	EB-13870	Consumer	United States	Orem	Utah	FUR-TA-10000577
27	US-2016-150630	2016-09-17	TB-21520	Consumer	United States	Philadelphia	Pennsylvania	FUR-BO-10004834
35	CA-2017-117590	2017-12-08	GH-14485	Corporate	United States	Richardson	Texas	TEC-PH-10004977
54	CA-2017-105816	2017-12-11	JM-15265	Corporate	United States	New York City	New York	TEC-PH-10002447

In [173…
```
#Average of all values before the 15% discount
average_before_discount15 = before_discount15.groupby("Discount")['Valor_Venda'].me
```

In [174…
```
print(average_before_discount15)
```

```
Discount
0.15    2116.807815
Name: Valor_Venda, dtype: float64
```

In [175…
```
#All value after 15% discount
dataset["Valor_Venda_With_Discount"] = dataset["Valor_Venda"] - (dataset["Valor_Ver
dataset.head()
```

Out[175]:

	ID_Pedido	Data_Pedido	ID_Cliente	Segmento	Pais	Cidade	Estado	ID_Produto	Cate
0	CA-2017-152156	2017-11-08	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-BO-10001798	Furr
1	CA-2017-152156	2017-11-08	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-CH-10000454	Furr
2	CA-2017-138688	2017-06-12	DV-13045	Corporate	United States	Los Angeles	California	OFF-LA-10000240	Sup
3	US-2016-108966	2016-10-11	SO-20335	Consumer	United States	Fort Lauderdale	Florida	FUR-TA-10000577	Furr
4	US-2016-108966	2016-10-11	SO-20335	Consumer	United States	Fort Lauderdale	Florida	OFF-ST-10000760	Sup

In [176…
```
discount15 = dataset[dataset["Discount"] == 0.15]
```

In [177…
```
groupby_discount15 = discount15.groupby("Discount")["Valor_Venda_With_Discount"]
```

In [178…
```
#Average of all values affected by 15
average_after_discount15 = groupby_discount15.mean()
```

In [179…
```
print(average_after_discount15)
```

```
Discount
0.15    1799.286643
Name: Valor_Venda_With_Discount, dtype: float64
```

9 What is the average sales per segment, per year and per month?

In [180…
```
#Create the column of the month
dataset["Month"] = dataset["Data_Pedido"].dt.month
```

In [181…
```
dataset.head()
```

Out[181]:

	ID_Pedido	Data_Pedido	ID_Cliente	Segmento	Pais	Cidade	Estado	ID_Produto	Cate
0	CA-2017-152156	2017-11-08	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-BO-10001798	Furr
1	CA-2017-152156	2017-11-08	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-CH-10000454	Furr
2	CA-2017-138688	2017-06-12	DV-13045	Corporate	United States	Los Angeles	California	OFF-LA-10000240	Sup
3	US-2016-108966	2016-10-11	SO-20335	Consumer	United States	Fort Lauderdale	Florida	FUR-TA-10000577	Furr
4	US-2016-108966	2016-10-11	SO-20335	Consumer	United States	Fort Lauderdale	Florida	OFF-ST-10000760	Sup

In [182…
```
#Group by year, by month and segment,and calculate aggregation statistics
dataset1 = dataset.groupby(['Year', "Month", 'Segmento'])["Valor_Venda"].agg([np.su
```

In [183…
```
dataset1
```

Out[183]:

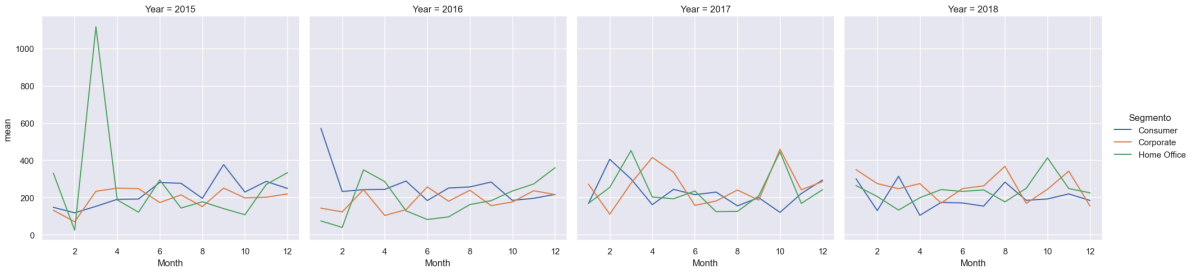
Year	Month	Segmento	sum	mean	median
2015	1	Consumer	6896.6290	146.736787	36.440
		Corporate	1701.5280	130.886769	51.940
		Home Office	5607.5500	329.855882	62.820
	2	Consumer	3167.8540	117.327926	22.776
		Corporate	623.1180	69.235333	62.310
...	...	...	...	...	...
2018	11	Corporate	44357.9862	341.215278	79.530
		Home Office	24009.3840	247.519423	56.560
	12	Consumer	48107.4558	183.616244	52.533
		Corporate	20524.4320	153.167403	46.460
		Home Office	13022.3910	224.523983	49.260

144 rows × 3 columns

In [184...
```
#Extract the levels
year = dataset1.index.get_level_values(0)
month = dataset1.index.get_level_values(1)
segment = dataset1.index.get_level_values(2)
```

In [185...
```
#Plot
plt.figure(figsize = (12,6))
sns.set()
fig1 = sns.relplot(kind = 'line',
                   data = dataset1,
                   y = 'mean',
                   x = month,
                   hue = segment,
                   col = year,
                   col_wrap =4)
plt.show()
```

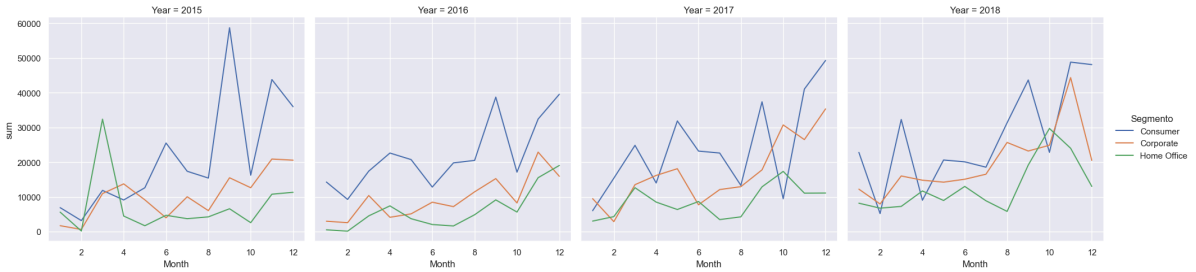
```
<Figure size 1200x600 with 0 Axes>
```



In [186...
```
#Plot
plt.figure(figsize = (12,6))
sns.set()
fig1 = sns.relplot(kind = 'line',
                   data = dataset1,
                   y = 'sum',
                   x = month,
                   hue = segment,
                   col = year,
```

```
                    col_wrap =4)
plt.show()
```

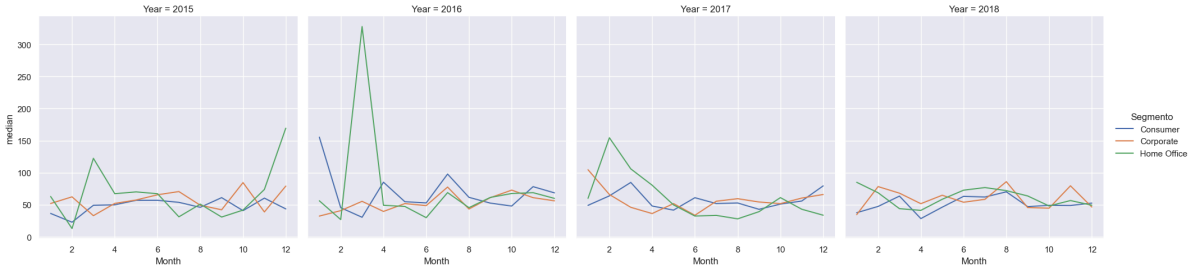
```
<Figure size 1200x600 with 0 Axes>
```



In [187…

```
#Plot
plt.figure(figsize = (12,6))
sns.set()
fig1 = sns.relplot(kind = 'line',
                   data = dataset1,
                   y = 'median',
                   x = month,
                   hue = segment,
                   col = year,
                   col_wrap =4)
plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```



10 What The Total Sales By Categories e Subcategories For only Top 12 Subcategories

In [188…

```
dataset.head()
```

Out[188]:

	ID_Pedido	Data_Pedido	ID_Cliente	Segmento	Pais	Cidade	Estado	ID_Produto	Cate
0	CA-2017-152156	2017-11-08	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-BO-10001798	Furr
1	CA-2017-152156	2017-11-08	CG-12520	Consumer	United States	Henderson	Kentucky	FUR-CH-10000454	Furr
2	CA-2017-138688	2017-06-12	DV-13045	Corporate	United States	Los Angeles	California	OFF-LA-10000240	Sup
3	US-2016-108966	2016-10-11	SO-20335	Consumer	United States	Fort Lauderdale	Florida	FUR-TA-10000577	Furr
4	US-2016-108966	2016-10-11	SO-20335	Consumer	United States	Fort Lauderdale	Florida	OFF-ST-10000760	Sup

In [189…

```
#Group by Categorie and SubCategorie, and calculate only numeric sum
dataset2 = dataset.groupby(["Categoria", "SubCategoria"]).sum(numeric_only = True).
```

In [190…
```
# Convert the Sales Value column to an integer and sort by category
dataset3 =  dataset2[['Valor_Venda']].astype(int).sort_values(by = 'Categoria').res
```

In [192…
```
# Dataframe with categories and subcategories
dataset3
```

Out[192]:

	Categoria	SubCategoria	Valor_Venda
0	Furniture	Chairs	317919
1	Furniture	Tables	202083
2	Furniture	Bookcases	108045
3	Furniture	Furnishings	88862
4	Office Supplies	Storage	216188
5	Office Supplies	Binders	194723
6	Office Supplies	Appliances	104061
7	Office Supplies	Paper	76312
8	Technology	Phones	325271
9	Technology	Machines	189238
10	Technology	Accessories	162791
11	Technology	Copiers	146248

In [195…
```
#Create another dataframe with only the totals by category
dataset3_cat = dataset3.groupby('Categoria').sum(numeric_only = True).reset_index()
```

In [196…
```
# Dataframe with categories
dataset3_cat
```

Out[196]:

	Categoria	Valor_Venda
0	Furniture	716909
1	Office Supplies	591284
2	Technology	823548

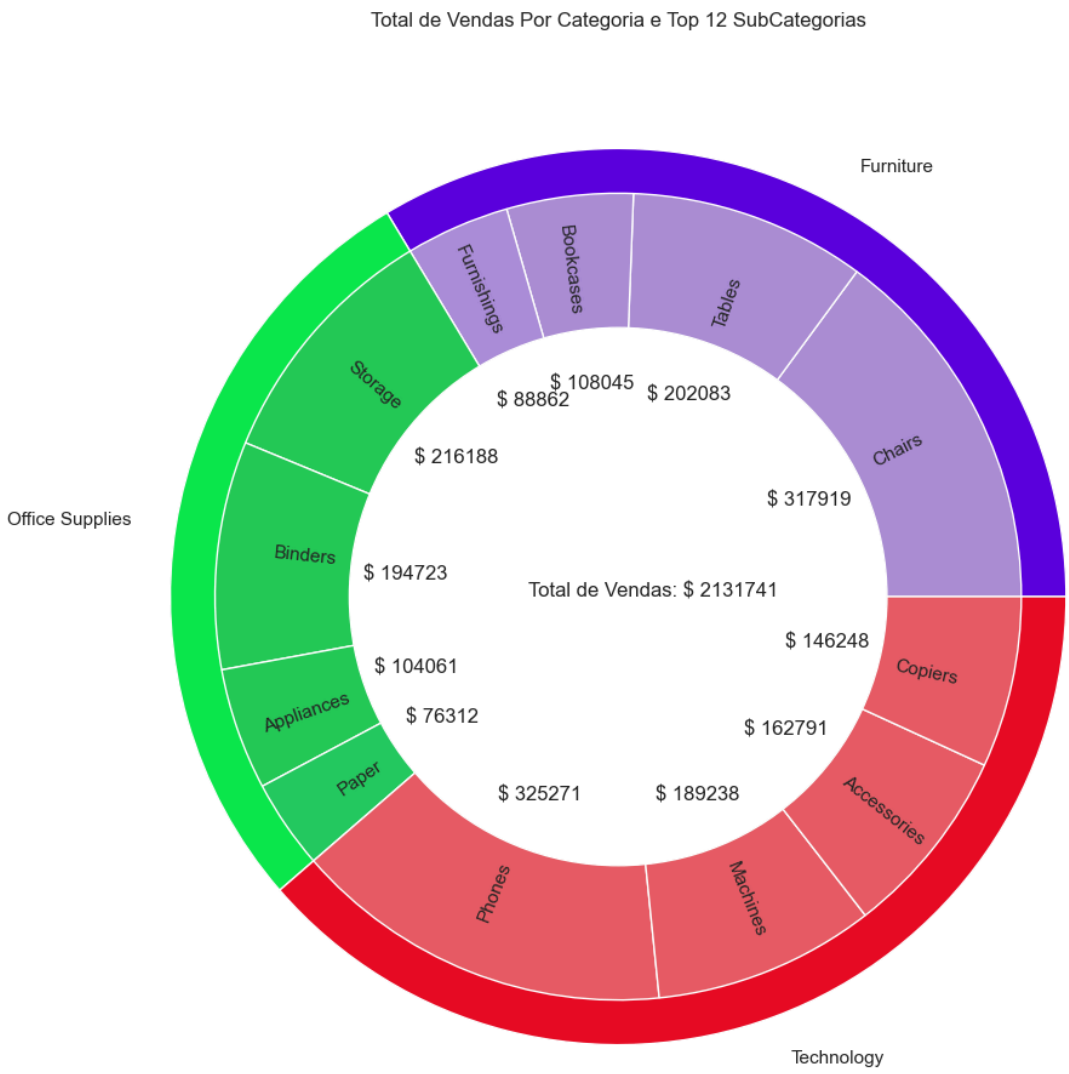
In [197…
```
# Color lists for categories
colors_categories = ['#5d00de',
                     '#0ee84f',
                     '#e80e27']
```

In [198…

```
# Color lists for subcategories
colors_subcategories = ['#aa8cd4',
                        '#aa8cd5',
                        '#aa8cd6',
                        '#aa8cd7',
                        '#26c957',
                        '#26c958',
                        '#26c959',
                        '#26c960',
                        '#e65e65',
                        '#e65e66',
                        '#e65e67',
                        '#e65e68']
```

In [201…

```
# Plot

# Size of the figure
fig, ax = plt.subplots(figsize = (18,12))

# Graphics of the categories
p1 = ax.pie(dataset3_cat['Valor_Venda'],
            radius = 1,
            labels = dataset3_cat['Categoria'],
            wedgeprops = dict(edgecolor = 'white'),
            colors = colors_categories)

# Graphics of the subcategories
p2 = ax.pie(dataset3['Valor_Venda'],
            radius = 0.9,
            labels = dataset3['SubCategoria'],
            autopct = autopct_format(dataset3['Valor_Venda']),
            colors = colors_subcategories,
            labeldistance = 0.7,
            wedgeprops = dict(edgecolor = 'white'),
            pctdistance = 0.53,
            rotatelabels = True)

# Clean the center of the circle
centre_circle = plt.Circle((0, 0), 0.6, fc = 'white')

# Labels and notes
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.annotate(text = 'Total de Vendas: ' + '$ ' + str(int(sum(dataset3['Valor_Venda'
plt.title('Total de Vendas Por Categoria e Top 12 SubCategorias')
plt.show()
```


Total de Vendas Por Categoria e Top 12 SubCategorias
Furniture
Office Supplies
Technology
Furnishings
Bookcases
Tables
Chairs
Storage
Binders
Appliances
Paper
Phones
Machines
Accessories
Copiers
$ 88862
$ 108045
$ 202083
$ 317919
$ 216188
$ 194723
$ 104061
$ 76312
$ 325271
$ 189238
$ 162791
$ 146248
Total de Vendas: $ 2131741

In [ ]: