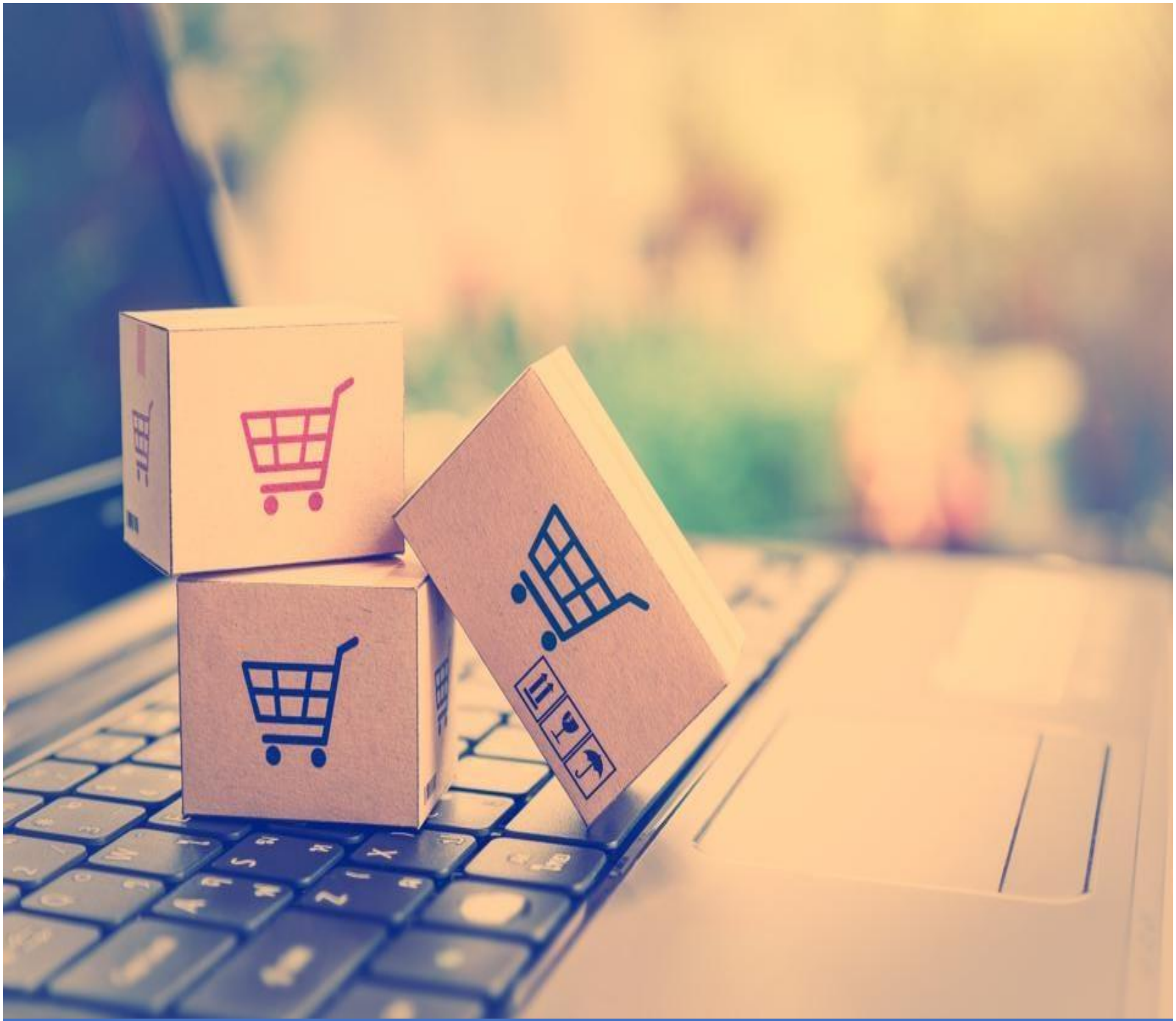


Project Topic Name - United States E-Commerce Records 2020

Team Members Name: CIN Number

Rashmi Pareek	401348464
Pooja Madhup	401977573

Introduction

This Dataset consists of 19 columns which provides us almost everything we need to know like Sales, Profit, Discounts, Quantity, State name, City name, Category, Subcategory etc. This dataset will provide the value of Ecommerce in the US market. The goal of Ecommerce is to reach maximum customers at the right time to increase sales and profitability of the business. [1]Functions of e-commerce include buying and selling goods, transmitting funds or data over the internet. Almost anything can be purchased through ecommerce today, it can be a substitute for brick-and-mortar stores, though some businesses choose to maintain both. [2]The primary objective of the Ecommerce System is to manage the details of Shopping, Internet, Payment, Bills, Customer. It manages all the information about Shopping, Products. From this dataset we will also learn How much did e-commerce grow in 2020. E-commerce is the activity of buying or selling of products on online services or over the Internet. From this dataset we will find out the sales over the city, which region of the us country could bring highest profit or lowest profit and we will also find out the total sales and revenue on the most popular product. [3]Data from the US Department of ecommerce shows that US ecommerce sales have been growing steadily for over a decade. Amazon is an excellent example of B2C ecommerce model as they sell individual goods to individual customers. [4]There are many B2C companies that have taken the market by storm, such as Expedia, Inc., IKEA, and Netflix. Here are four traditional types of ecommerce, including B2C (Business-to-Consumer), B2B (Business-to-Business), C2B (Consumer-to-Business) and C2C (Consumer-to-Consumer). There's also B2G (Business-to-Government), but it is often lumped in with B2B[5].

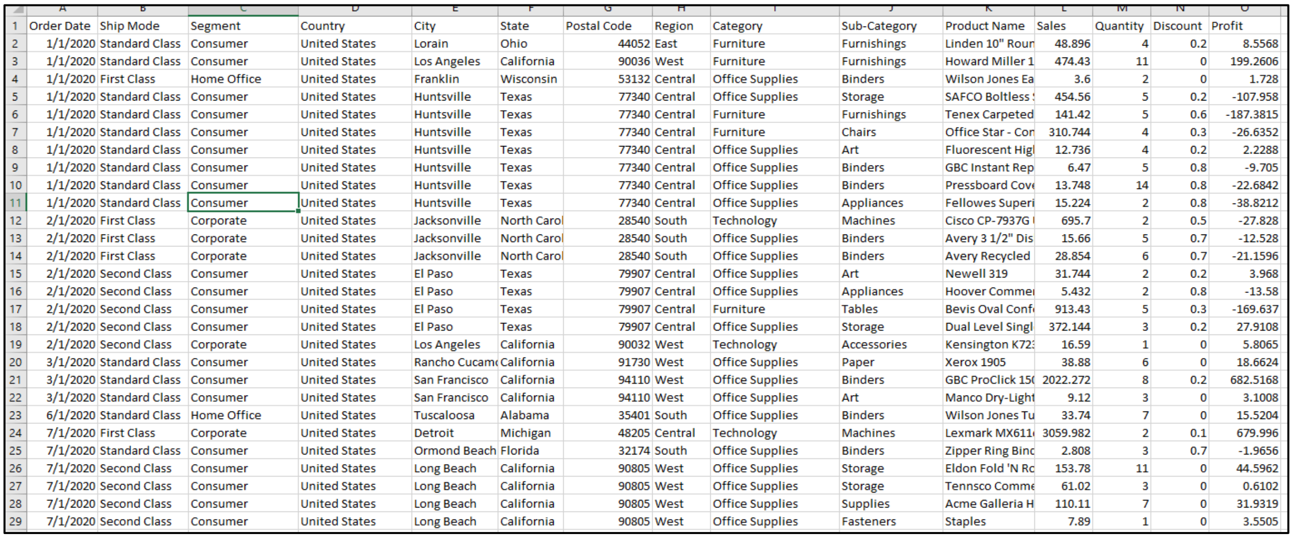
References:

US Ecommerce Sales [Updated March 2022] | Oberlo

E-commerce surged during Covid: Groceries, sporting goods top gainers (cnbc.com)

US Ecommerce by Category 2022 - Insider Intelligence Trends, Forecasts & Statistics (emarketer.com)

Below are some sample values from the dataset.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Order Date	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Product Name	Sales	Quantity	Discount	Profit
2	1/1/2020	Standard Class	Consumer	United States	Lorain	Ohio	44052	East	Furniture	Furnishings	Linden 10" Roun	48.896	4	0.2	8.5568
3	1/1/2020	Standard Class	Consumer	United States	Los Angeles	California	90036	West	Furniture	Furnishings	Howard Miller 1	474.43	11	0	199.2606
4	1/1/2020	First Class	Home Office	United States	Franklin	Wisconsin	53132	Central	Office Supplies	Binders	Wilson Jones Ea	3.6	2	0	1.728
5	1/1/2020	Standard Class	Consumer	United States	Huntsville	Texas	77340	Central	Office Supplies	Storage	SAFCO Boltless S	454.56	5	0.2	-107.958
6	1/1/2020	Standard Class	Consumer	United States	Huntsville	Texas	77340	Central	Furniture	Furnishings	Tenex Carpeted	141.42	5	0.6	-187.3815
7	1/1/2020	Standard Class	Consumer	United States	Huntsville	Texas	77340	Central	Furniture	Chairs	Office Star - Con	310.744	4	0.3	-26.6352
8	1/1/2020	Standard Class	Consumer	United States	Huntsville	Texas	77340	Central	Office Supplies	Art	Fluorescent Higl	12.736	4	0.2	2.2288
9	1/1/2020	Standard Class	Consumer	United States	Huntsville	Texas	77340	Central	Office Supplies	Binders	GBC Instant Rep	6.47	5	0.8	-9.705
10	1/1/2020	Standard Class	Consumer	United States	Huntsville	Texas	77340	Central	Office Supplies	Binders	Pressboard Cove	13.748	14	0.8	-22.6842
11	1/1/2020	Standard Class	Consumer	United States	Huntsville	Texas	77340	Central	Office Supplies	Appliances	Fellowes Superi	15.224	2	0.8	-38.8212
12	2/1/2020	First Class	Corporate	United States	Jacksonville	North Carol	28540	South	Technology	Machines	Cisco CP-7937G	695.7	2	0.5	-27.828
13	2/1/2020	First Class	Corporate	United States	Jacksonville	North Carol	28540	South	Office Supplies	Binders	Avery 3 1/2" Dis	15.66	5	0.7	-12.528
14	2/1/2020	First Class	Corporate	United States	Jacksonville	North Carol	28540	South	Office Supplies	Binders	Avery Recycled	28.854	6	0.7	-21.1596
15	2/1/2020	Second Class	Consumer	United States	El Paso	Texas	79907	Central	Office Supplies	Art	Newell 319	31.744	2	0.2	3.968
16	2/1/2020	Second Class	Consumer	United States	El Paso	Texas	79907	Central	Office Supplies	Appliances	Hoover Commer	5.432	2	0.8	-13.58
17	2/1/2020	Second Class	Consumer	United States	El Paso	Texas	79907	Central	Furniture	Tables	Bevis Oval Conf	913.43	5	0.3	-169.637
18	2/1/2020	Second Class	Consumer	United States	El Paso	Texas	79907	Central	Office Supplies	Storage	Dual Level Singl	372.144	3	0.2	27.9108
19	2/1/2020	Second Class	Corporate	United States	Los Angeles	California	90032	West	Technology	Accessories	Kensington K72	16.59	1	0	5.8065
20	3/1/2020	Standard Class	Consumer	United States	Rancho Cucam	California	91730	West	Office Supplies	Paper	Xerox 1905	38.88	6	0	18.6624
21	3/1/2020	Standard Class	Consumer	United States	San Francisco	California	94110	West	Office Supplies	Binders	GBC ProClick 15	2022.272	8	0.2	682.5168
22	3/1/2020	Standard Class	Consumer	United States	San Francisco	California	94110	West	Office Supplies	Art	Manco Dry-Light	9.12	3	0	3.1008
23	6/1/2020	Standard Class	Home Office	United States	Tuscaloosa	Alabama	35401	South	Office Supplies	Binders	Wilson Jones Tu	33.74	7	0	15.5204
24	7/1/2020	First Class	Corporate	United States	Detroit	Michigan	48205	Central	Technology	Machines	Lexmark MX611	3059.982	2	0.1	679.996
25	7/1/2020	Standard Class	Consumer	United States	Ormond Beach	Florida	32174	South	Office Supplies	Binders	Zipper Ring Binc	2.808	3	0.7	-1.9656
26	7/1/2020	Second Class	Consumer	United States	Long Beach	California	90805	West	Office Supplies	Storage	Eldon Fold 'N Rc	153.78	11	0	44.5962
27	7/1/2020	Second Class	Consumer	United States	Long Beach	California	90805	West	Office Supplies	Storage	Tennsco Comme	61.02	3	0	0.6102
28	7/1/2020	Second Class	Consumer	United States	Long Beach	California	90805	West	Office Supplies	Supplies	Acme Galleria H	110.11	7	0	31.9319
29	7/1/2020	Second Class	Consumer	United States	Long Beach	California	90805	West	Office Supplies	Fasteners	Staples	7.89	1	0	3.5505

The dataset which we selected is the US ecommerce dataset. This dataset contains the 3312 rows and 19 columns.

Data set URL’s and Data set Description

https://www.kaggle.com/datasets/ammaraahmad/us-ecommerce-record-2020

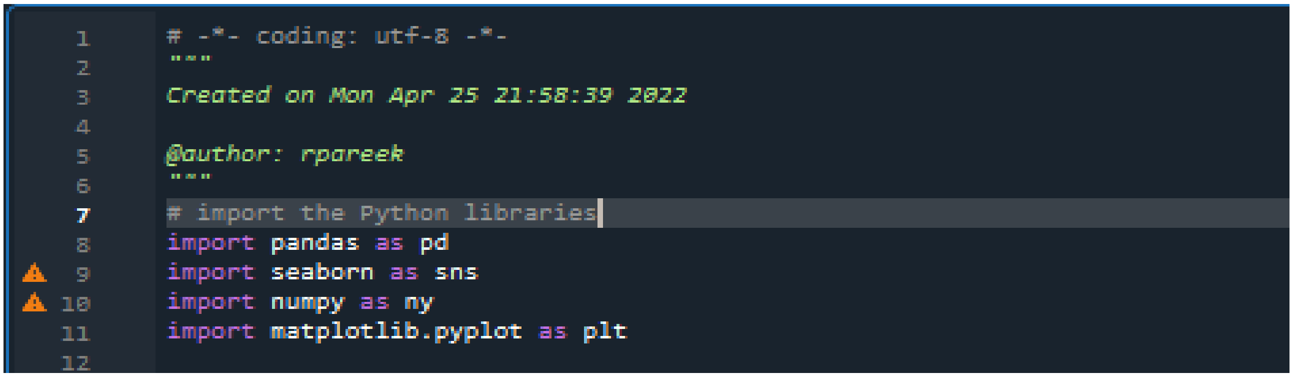
Field Name	**Description**	**Example Value**
Order Date	Order date is the date that a customer has completed the transaction and made a purchase on particular date	For example, 1/1/2020
Ship Mode	From which mode customer receive the product	Standard class, first class, Second class
Customer id	Customer ID is a unique number on your invoice that is used to reference your account	GA-14725 Unique customer id DP-13390 Unique customer id
Segment	Demographic Customer Segment	Consumer, Home Office
Country	Country is basically Showing, from which particular country buying and selling of goods	United States is basically the country name from customer purchased the product
City	City is showing the location of buying and selling of goods	Lorain, Los Angeles
State	State is showing the location of buying and selling of goods	Ohio, California
Postal Code	Postal codes are the default method used by merchants to verify a customer's information	44052 90036(Customer’s postal code)
Region	Region is basically showing the company’s branch location	East, West Central(Company’s Branch location)
Product ID	Every Product is assigned the particular product id	FUR-FU-10003878(Product Id number)
Category	Group of products is divided into different categories	Furniture, Office Supplies

Sub- Category	Subcategory is part of category	Furnishings, Binders
Product Name	Product name identifies a specific product or service and becomes a brand name	Linden 10" Round Wall Clock, Black, Howard Miller 11-1/2" Diameter Brentwood Wall Clock(Product name)
Sales	Sales is the process of convincing a consumer to purchase goods or services	48.896, (Sales) 474.43
Quantity	Quantity is defined as an amount, measure, or number	4 11
Discount	An amount or percentage deducted from the normal selling price of something	0.2(Discount on particular Product) 0
Profit	A financial gain, especially the difference between the amount earned and the amount spent in buying, operating, or producing something.	8.5568 (Profit on Particular Product) 199.2606 1.728

Data Cleaning

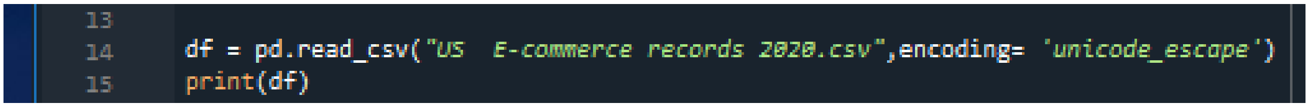
Having clean data will ultimately increase overall productivity and allow for the highest quality information in your decision making. If data is correct, outcomes and algorithms are unreliable. Before jumping into the data cleaning process. Let’s look at the data.

# Import the Python libraries

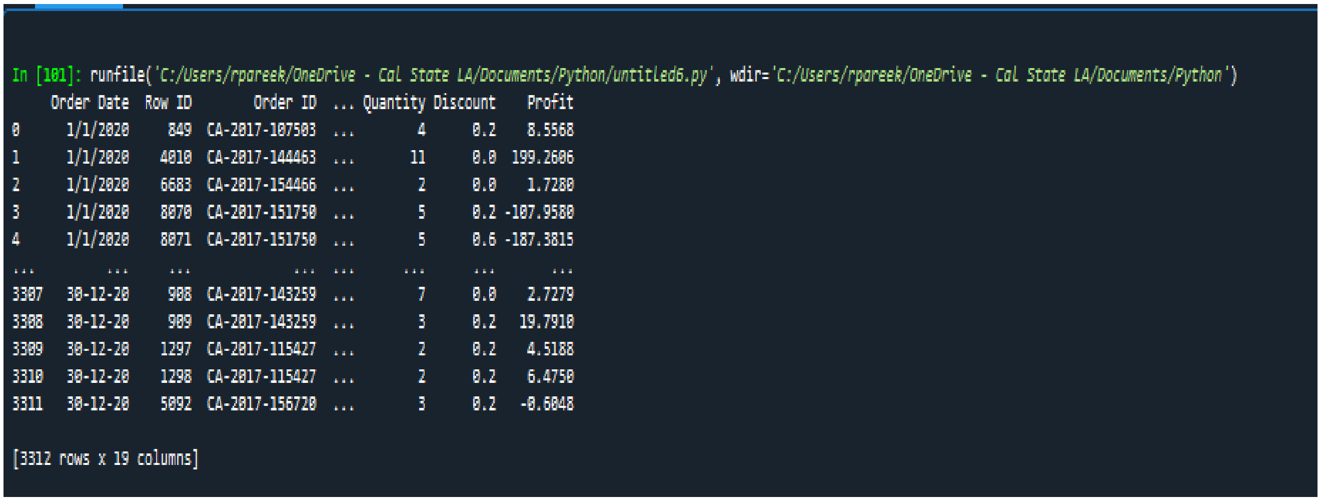


```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 25 21:58:39 2022

@author: rpareek
"""
# import the Python libraries
import pandas as pd
import seaborn as sns
import numpy as ny
import matplotlib.pyplot as plt
```

Read and show the data file:



```

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')
print(df)
```



```
In [101]: runfile('C:/Users/rpareek/OneDrive - Cal State LA/Documents/Python/untitled6.py', wdir='C:/Users/rpareek/OneDrive - Cal State LA/Documents/Python')
     Order Date  Row ID        Order ID  ... Quantity Discount    Profit
0      1/1/2020     849  CA-2017-107503  ...        4      0.2    8.5568
1      1/1/2020    4010  CA-2017-144463  ...       11      0.0  199.2606
2      1/1/2020    6683  CA-2017-154466  ...        2      0.0    1.7280
3      1/1/2020    8070  CA-2017-151750  ...        5      0.2 -107.9580
4      1/1/2020    8071  CA-2017-151750  ...        5      0.6 -187.3815
...         ...     ...             ...  ...      ...      ...       ...
3307   30-12-20     908  CA-2017-143259  ...        7      0.0    2.7279
3308   30-12-20     909  CA-2017-143259  ...        3      0.2   19.7910
3309   30-12-20    1297  CA-2017-115427  ...        2      0.2    4.5188
3310   30-12-20    1298  CA-2017-115427  ...        2      0.2    6.4750
3311   30-12-20    5092  CA-2017-156720  ...        3      0.2   -0.6048

[3312 rows x 19 columns]
```

From these results, we can see that the dataset has **3312 rows and 19 columns**. The rows show the number of US ecommerce records of 2020 year. Now I can run through the checklist of “dirty data type and fix them one by one.

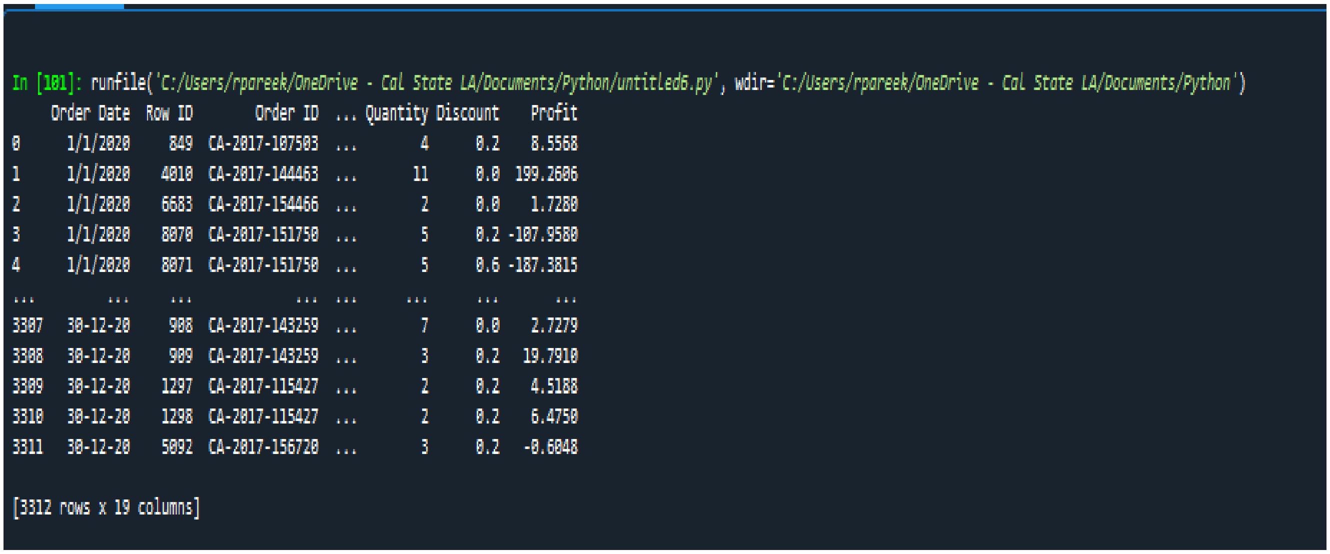
#Removed Multiple Unnecessary Column:

All the data feeding into the model should serve the purpose of the project. The unnecessary data is when the data does not add value. In our Project dataset we have multiple unnecessary Column, which are not required for the project. So, we checked and identify which columns are not required or which columns are unnecessary. Below is the code to remove unwanted columns and result of data cleaning.

Code:

```
df = pd.read_csv("US  E-commerce records 2020.csv",encoding=
'unicode_escape')
print(df)
```

Pre cleaning Screenshot



```
In [101]: runfile('C:/Users/rpareek/OneDrive - Cal State LA/Documents/Python/untitled6.py', wdir='C:/Users/rpareek/OneDrive - Cal State LA/Documents/Python')
    Order Date  Row ID         Order ID  ... Quantity Discount    Profit
0     1/1/2020     849  CA-2017-107503  ...        4      0.2    8.5568
1     1/1/2020    4010  CA-2017-144463  ...       11      0.0  199.2606
2     1/1/2020    6683  CA-2017-154466  ...        2      0.0    1.7280
3     1/1/2020    8070  CA-2017-151750  ...        5      0.2 -107.9580
4     1/1/2020    8071  CA-2017-151750  ...        5      0.6 -187.3815
...        ...     ...             ...  ...      ...      ...       ...
3307  30-12-20     908  CA-2017-143259  ...        7      0.0    2.7279
3308  30-12-20     909  CA-2017-143259  ...        3      0.2   19.7910
3309  30-12-20    1297  CA-2017-115427  ...        2      0.2    4.5188
3310  30-12-20    1298  CA-2017-115427  ...        2      0.2    6.4750
3311  30-12-20    5092  CA-2017-156720  ...        3      0.2   -0.6048

[3312 rows x 19 columns]
```

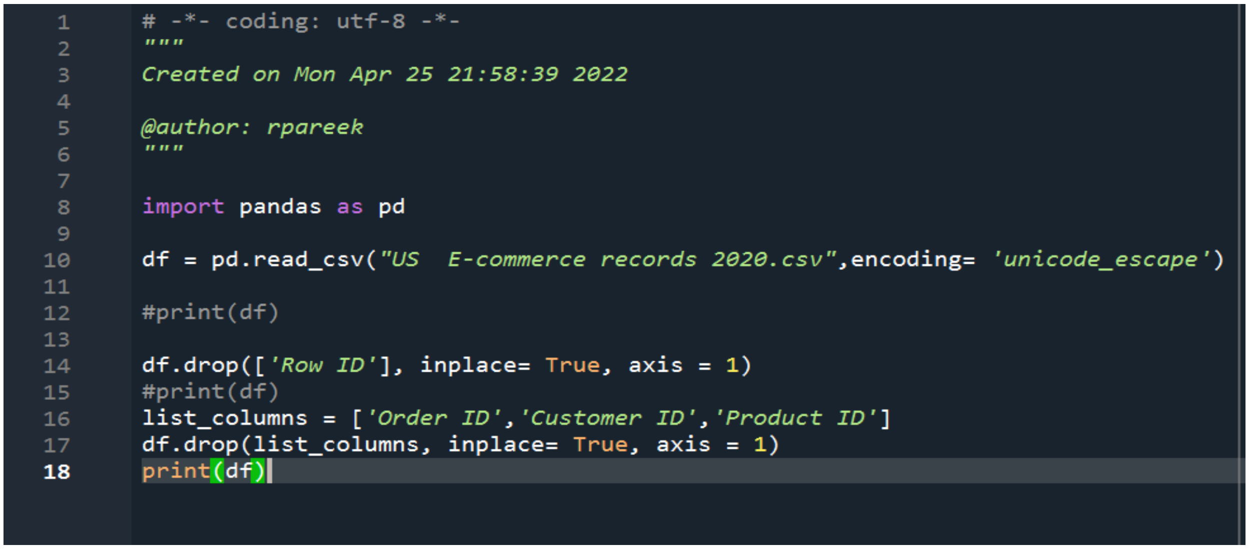
Post Cleaning Screenshot and Code:

We removed multiple unnecessary columns from our dataset and the screenshot below indicates unnecessary columns have been moved from dataset. We removed **Row id, Order id and Customer id and product id**. Below is the screenshot and code.

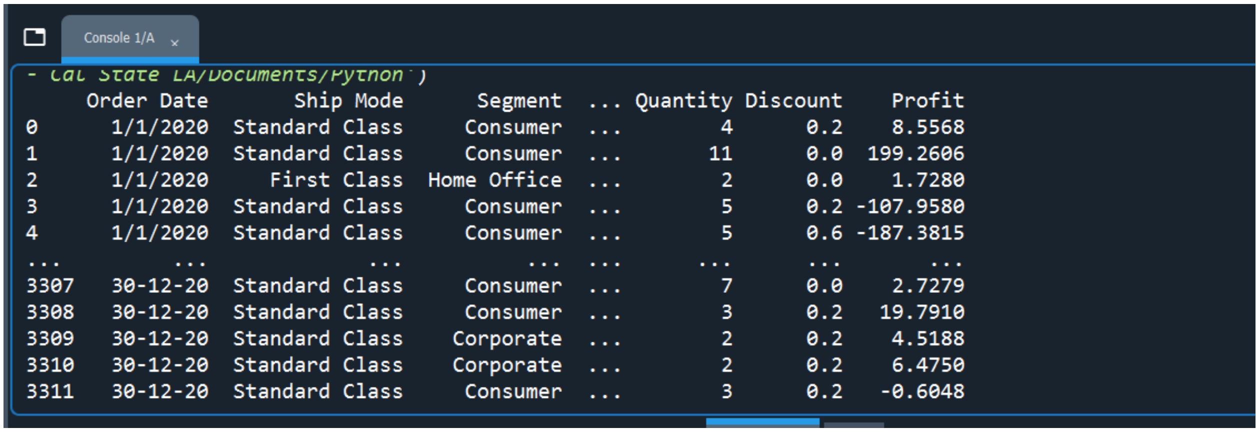
Code

```
import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np



df = pd.read_csv("./US_e-commerce_2020.csv", encoding = 'unicode_escape')

print(df)

df.drop(['Row ID'], inplace = True, axis =1)

#print(df)

list_columns = ['Order ID','Customer ID','Product ID']

df.drop(list_columns, inplace = True, axis = 1)

print(df)
```

Screenshot



```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 25 21:58:39 2022

@author: rpareek
"""

import pandas as pd

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')

#print(df)

df.drop(['Row ID'], inplace= True, axis = 1)
#print(df)
list_columns = ['Order ID','Customer ID','Product ID']
df.drop(list_columns, inplace= True, axis = 1)
print(df)
```



```
- Cal State LA/Documents/Python')
     Order Date      Ship Mode      Segment  ... Quantity Discount    Profit
0      1/1/2020 Standard Class     Consumer  ...        4      0.2    8.5568
1      1/1/2020 Standard Class     Consumer  ...       11      0.0  199.2606
2      1/1/2020    First Class  Home Office  ...        2      0.0    1.7280
3      1/1/2020 Standard Class     Consumer  ...        5      0.2 -107.9580
4      1/1/2020 Standard Class     Consumer  ...        5      0.6 -187.3815
...         ...            ...          ...  ...      ...      ...       ...
3307   30-12-20 Standard Class     Consumer  ...        7      0.0    2.7279
3308   30-12-20 Standard Class     Consumer  ...        3      0.2   19.7910
3309   30-12-20 Standard Class    Corporate  ...        2      0.2    4.5188
3310   30-12-20 Standard Class    Corporate  ...        2      0.2    6.4750
3311   30-12-20 Standard Class     Consumer  ...        3      0.2   -0.6048
```

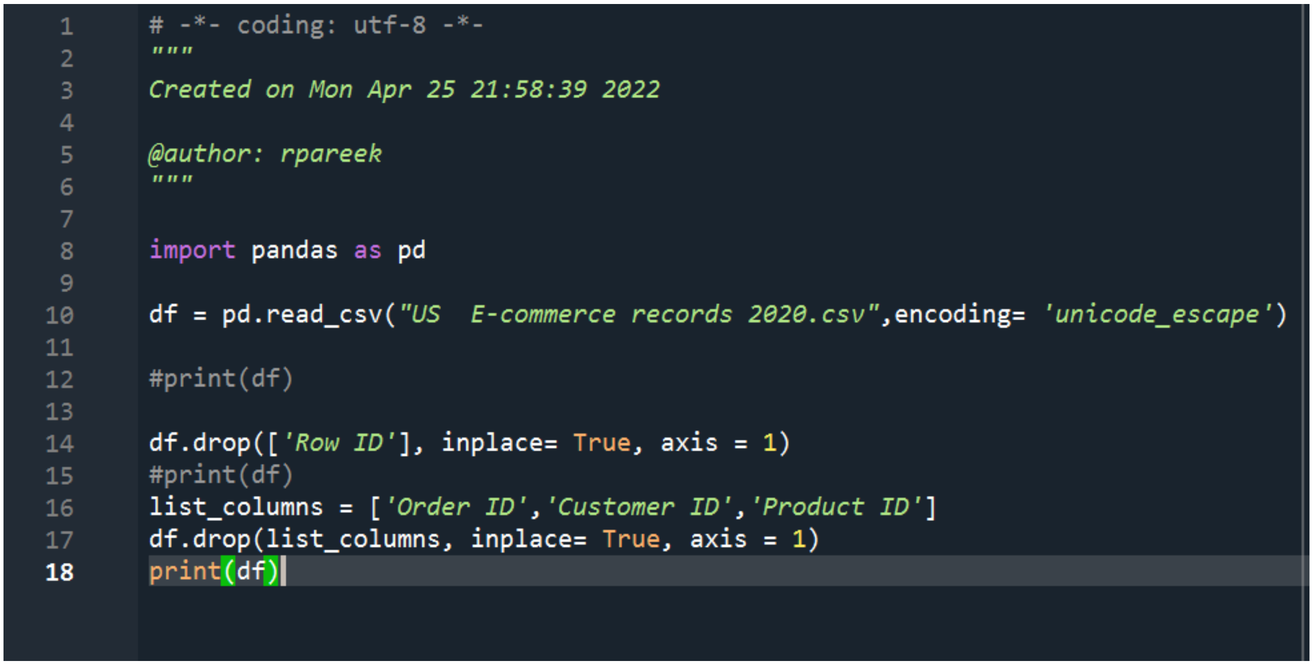
The above result shows the perfect data frame after data cleaning. So now my data frame consists of all information that I need to analyze which are the order data, ship mode, Segment, Quantity, Discount, profit, etc.

#Change Ship Mode column to lower case

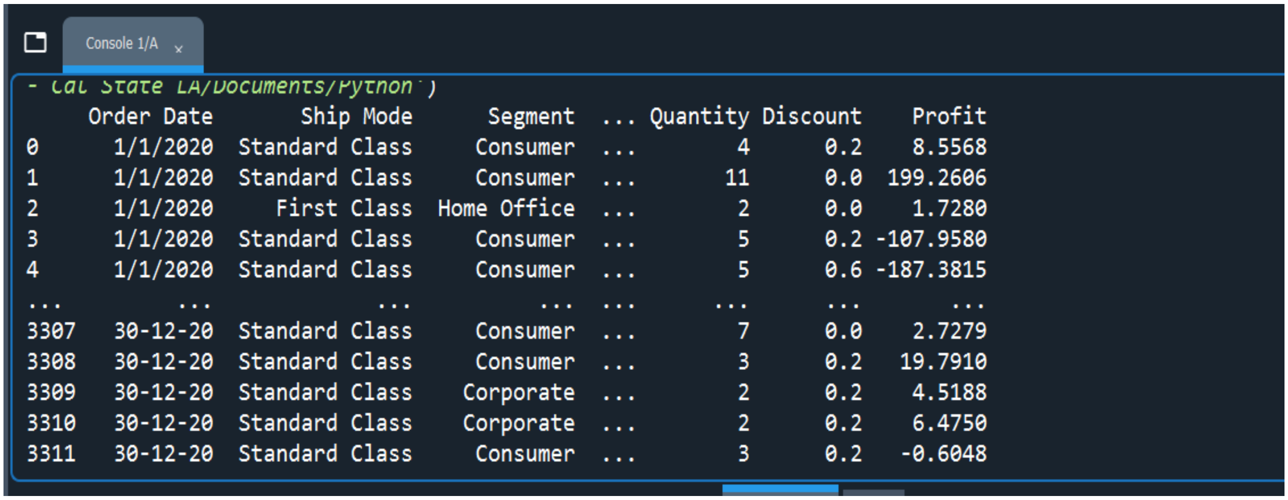
Ship mode columns data in uppercase, we have changed the uppercase data to lowercase. Below is the code for change ship mode column to lower case and result of data cleaning.

Pre-Cleaning Screenshot and code

```
import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import numpy as np



df = pd.read_csv("./US_e-commerce_2020.csv", encoding = 'unicode_escape')

print(df)

df.drop(['Row ID'], inplace = True, axis =1)

#print(df)

list_columns = ['Order ID','Customer ID','Product ID']

df.drop(list_columns, inplace = True, axis = 1)

print(df)
```



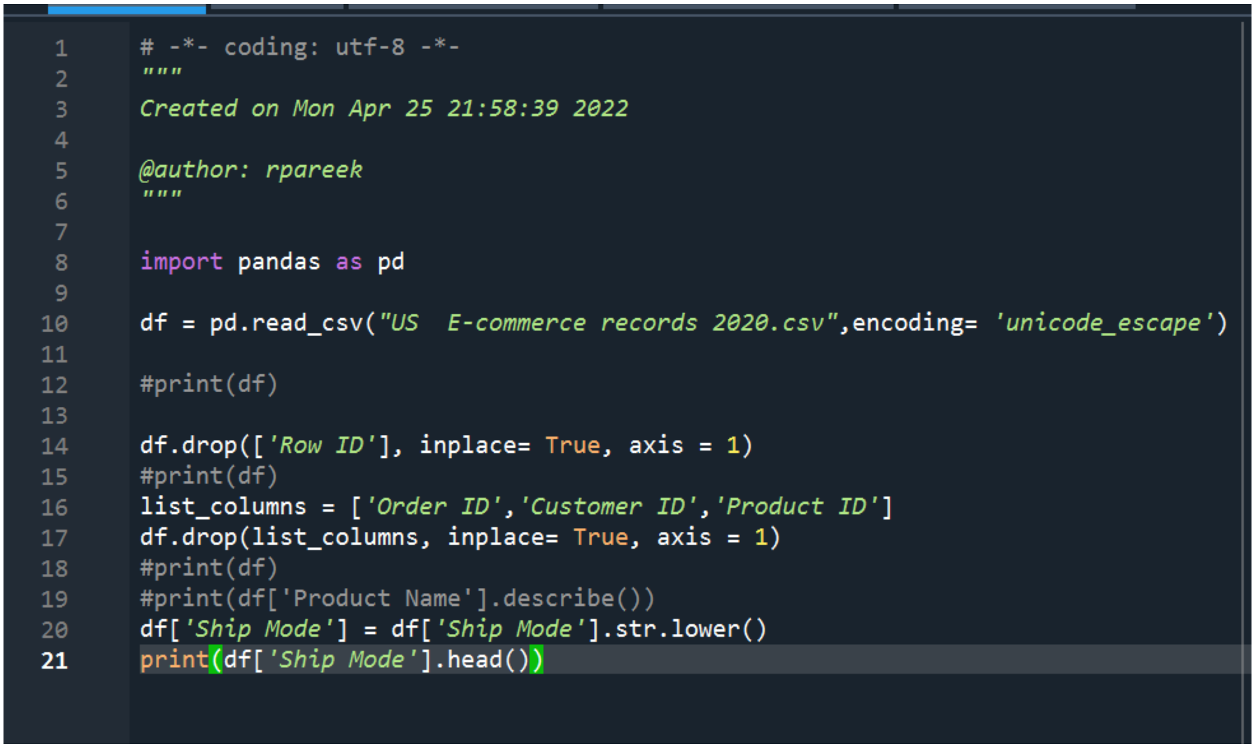
```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 25 21:58:39 2022

@author: rpareek
"""

import pandas as pd

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')

#print(df)

df.drop(['Row ID'], inplace= True, axis = 1)
#print(df)
list_columns = ['Order ID','Customer ID','Product ID']
df.drop(list_columns, inplace= True, axis = 1)
print(df)
```



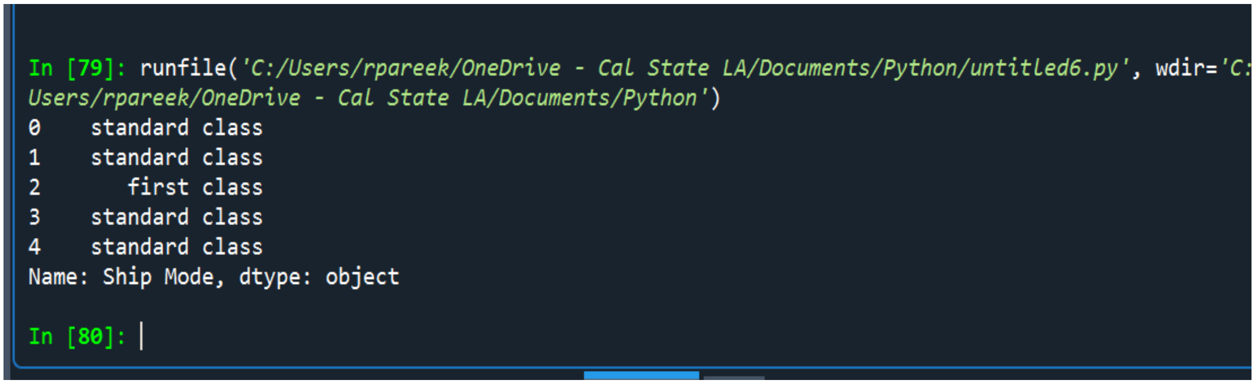
```
- Cal State LA/Documents/Python')
     Order Date       Ship Mode      Segment  ... Quantity Discount    Profit
0      1/1/2020  Standard Class     Consumer  ...        4      0.2    8.5568
1      1/1/2020  Standard Class     Consumer  ...       11      0.0  199.2606
2      1/1/2020     First Class  Home Office  ...        2      0.0    1.7280
3      1/1/2020  Standard Class     Consumer  ...        5      0.2 -107.9580
4      1/1/2020  Standard Class     Consumer  ...        5      0.6 -187.3815
...         ...             ...          ...  ...      ...      ...       ...
3307   30-12-20  Standard Class     Consumer  ...        7      0.0    2.7279
3308   30-12-20  Standard Class     Consumer  ...        3      0.2   19.7910
3309   30-12-20  Standard Class    Corporate  ...        2      0.2    4.5188
3310   30-12-20  Standard Class    Corporate  ...        2      0.2    6.4750
3311   30-12-20  Standard Class     Consumer  ...        3      0.2   -0.6048
```

Post-Cleaning Screenshot and Code

```
import pandas as pd

df = pd.read_csv("US E-commerce records 2020.csv",encoding= 'unicode_escape')

#print(df)


df.drop(['Row ID'], inplace= True, axis = 1)
#print(df)
list_columns = ['Order ID','Customer ID','Product ID']
df.drop(list_columns, inplace= True, axis = 1)
#print(df)
#print(df['Product Name'].describe())
df['Ship Mode'] = df['Ship Mode'].str.lower()
print(df['Ship Mode'].head())
```



```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 25 21:58:39 2022

@author: rpareek
"""

import pandas as pd

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')

#print(df)

df.drop(['Row ID'], inplace= True, axis = 1)
#print(df)
list_columns = ['Order ID','Customer ID','Product ID']
df.drop(list_columns, inplace= True, axis = 1)
#print(df)
#print(df['Product Name'].describe())
df['Ship Mode'] = df['Ship Mode'].str.lower()
print(df['Ship Mode'].head())
```



```
In [79]: runfile('C:/Users/rpareek/OneDrive - Cal State LA/Documents/Python/untitled6.py', wdir='C:
Users/rpareek/OneDrive - Cal State LA/Documents/Python')
0    standard class
1    standard class
2       first class
3    standard class
4    standard class
Name: Ship Mode, dtype: object

In [80]: |
```

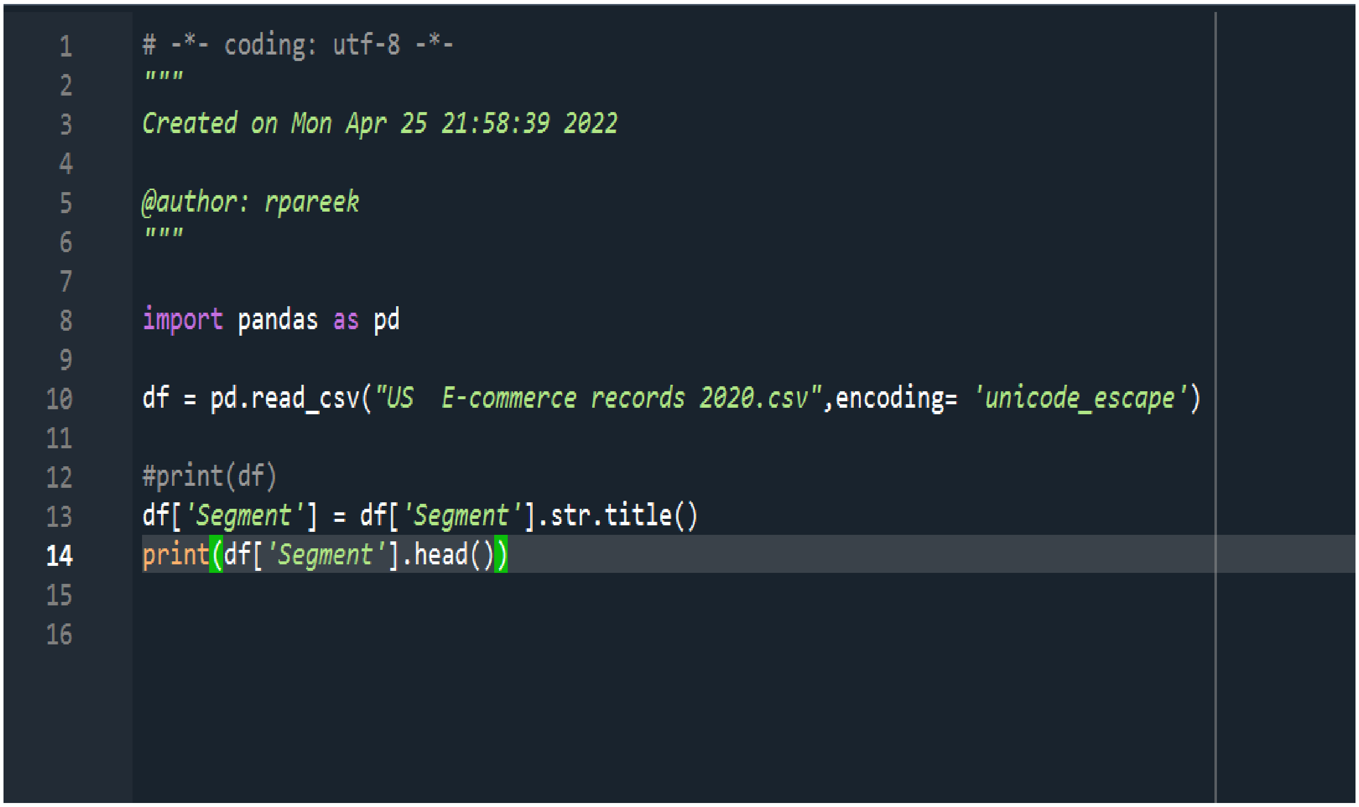
The above result shows the perfect data frame after data cleaning.

#Change Segment column to title case

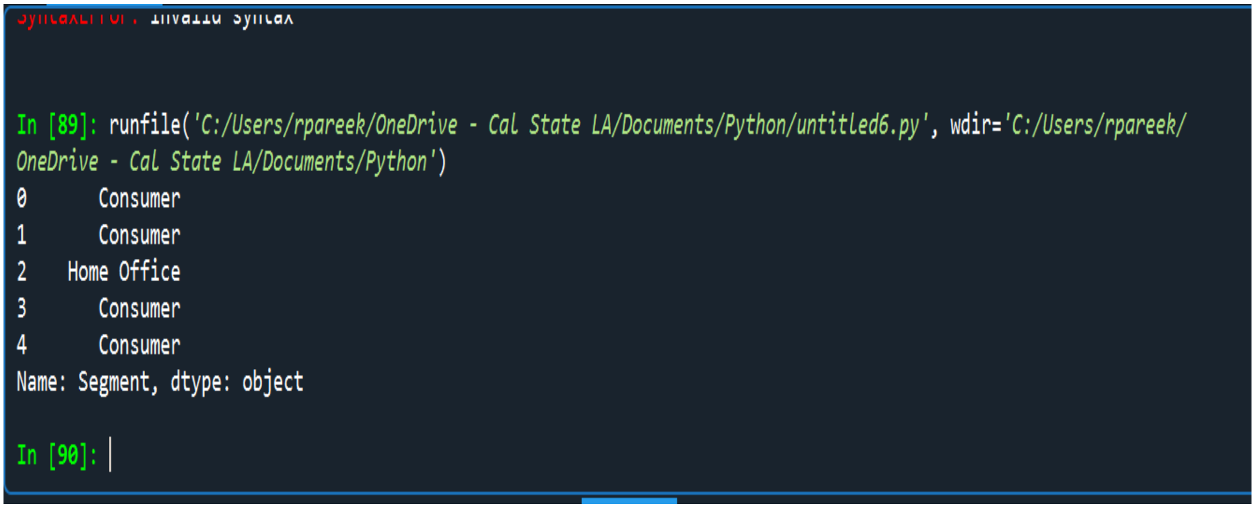
We changed the Segment column to title case because this is necessary for data cleaning part.

Screenshot and Code

```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 25 21:58:39 2022

@author: rpareek
"""

import pandas as pd

df = pd.read_csv("US E-commerce records 2020.csv",encoding= 'unicode_escape')

#print(df)
df['Segment'] = df['Segment'].str.title()
print(df['Segment'].head())
```



```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 25 21:58:39 2022

@author: rpareek
"""

import pandas as pd

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')

#print(df)
df['Segment'] = df['Segment'].str.title()
print(df['Segment'].head())


```



```
SyntaxError: invalid syntax


In [89]: runfile('C:/Users/rpareek/OneDrive - Cal State LA/Documents/Python/untitled6.py', wdir='C:/Users/rpareek/
OneDrive - Cal State LA/Documents/Python')
0       Consumer
1       Consumer
2    Home Office
3       Consumer
4       Consumer
Name: Segment, dtype: object

In [90]:
```

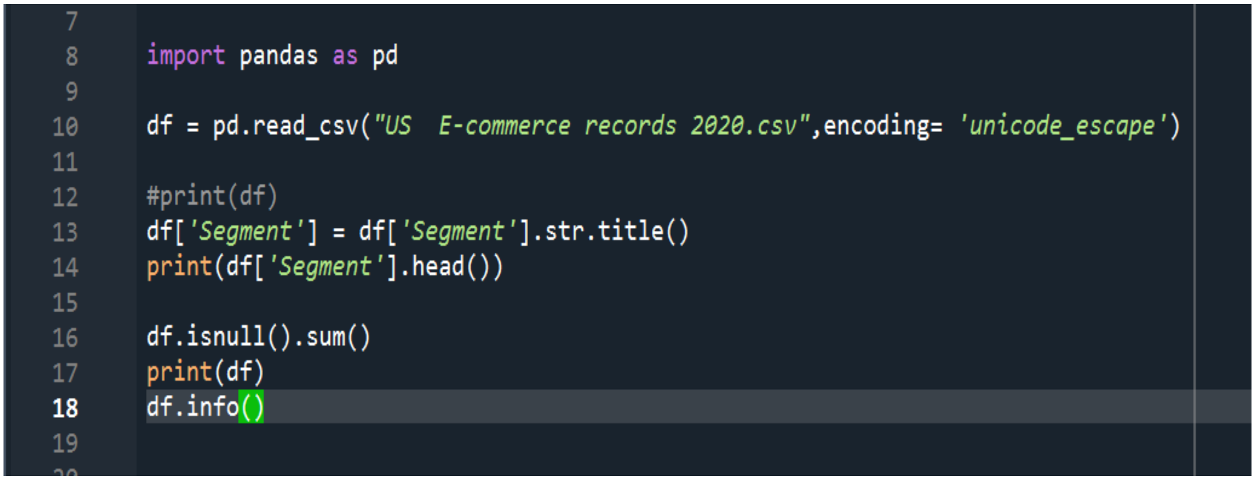
The above result shows the perfect data frame after data cleaning.

#Checking the Null Values

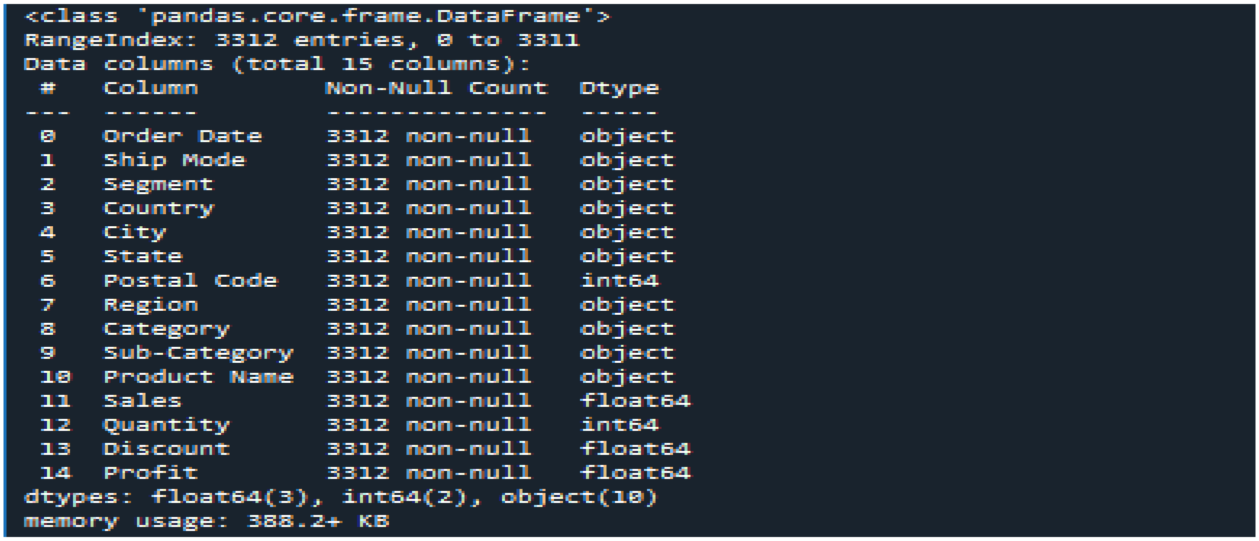
This data cleaning method we used to find the null values in our project dataset. As we can see in the screenshot below, there are no null values in our project dataset.

Screenshot and code

```
import pandas as pd

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')

#print(df)
df['Segment'] = df['Segment'].str.title()
print(df['Segment'].head())

df.isnull().sum()
print(df)
df.info()
```



```
import pandas as pd

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')

#print(df)
df['Segment'] = df['Segment'].str.title()
print(df['Segment'].head())

df.isnull().sum()
print(df)
df.info()
```



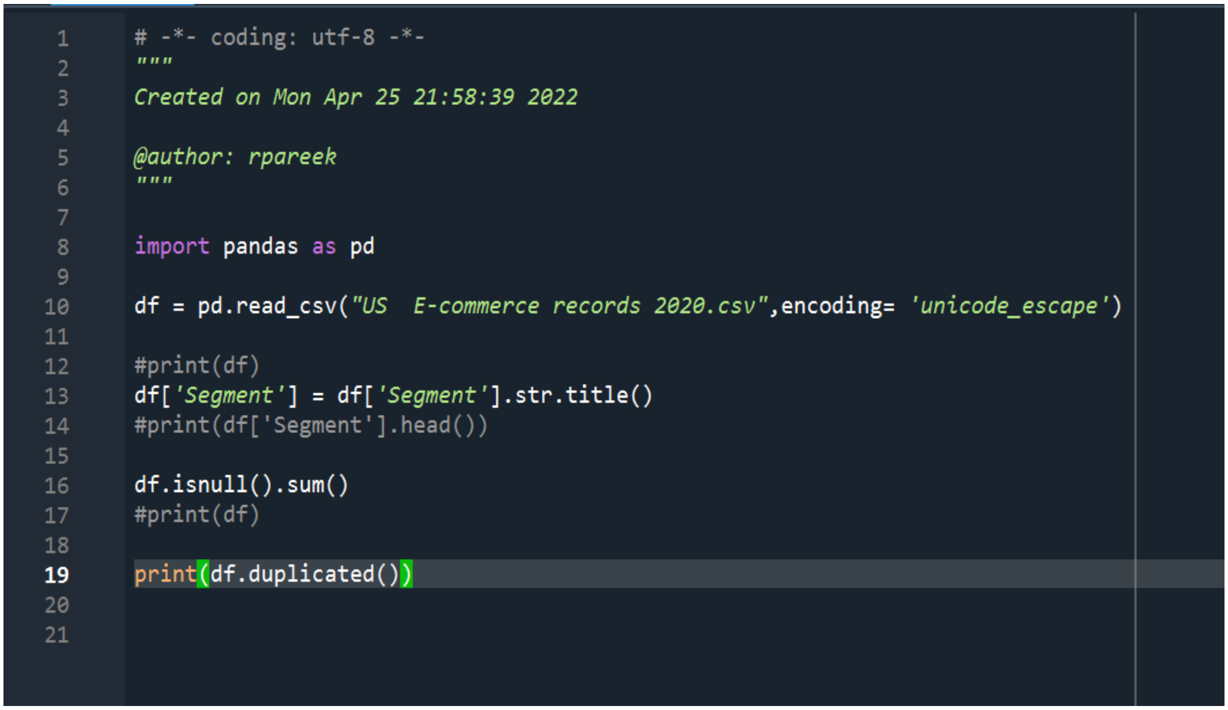
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3312 entries, 0 to 3311
Data columns (total 15 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Order Date     3312 non-null   object
 1   Ship Mode      3312 non-null   object
 2   Segment        3312 non-null   object
 3   Country        3312 non-null   object
 4   City           3312 non-null   object
 5   State          3312 non-null   object
 6   Postal Code    3312 non-null   int64
 7   Region         3312 non-null   object
 8   Category       3312 non-null   object
 9   Sub-Category   3312 non-null   object
 10  Product Name   3312 non-null   object
 11  Sales          3312 non-null   float64
 12  Quantity       3312 non-null   int64
 13  Discount       3312 non-null   float64
 14  Profit         3312 non-null   float64
dtypes: float64(3), int64(2), object(10)
memory usage: 388.2+ KB
```

The above result shows the perfect data frame after data cleaning. Because there is no null value present in our dataset.

#Check for Duplicates

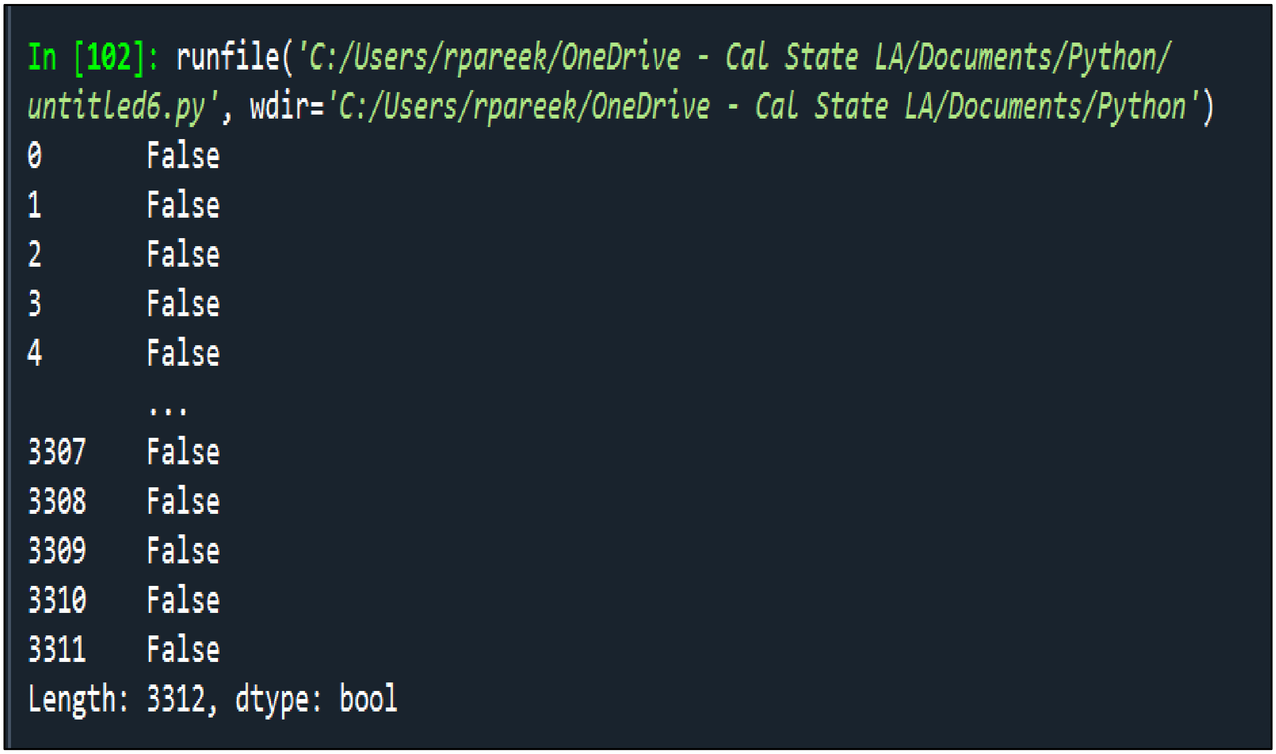
This Data cleaning method we used to check any duplicates present in our dataset or not. Below screenshot is clearly indicating there is no duplicates record or data present in dataset. So, we cleaned the complete dataset.

Code and Screenshot



```python
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 25 21:58:39 2022

@author: rpareek
"""


import pandas as pd

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')

#print(df)
df['Segment'] = df['Segment'].str.title()
#print(df['Segment'].head())

df.isnull().sum()
#print(df)

print(df.duplicated())


```

```python
import pandas as pd

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')

#print(df)
df['Segment'] = df['Segment'].str.title()
#print(df['Segment'].head())

df.isnull().sum()
#print(df)

print(df.duplicated())
```



```
In [102]: runfile('C:/Users/rpareek/OneDrive - Cal State LA/Documents/Python/
untitled6.py', wdir='C:/Users/rpareek/OneDrive - Cal State LA/Documents/Python')
0       False
1       False
2       False
3       False
4       False
        ...
3307    False
3308    False
3309    False
3310    False
3311    False
Length: 3312, dtype: bool
```

The above result shows the perfect data frame after data cleaning. So, now our data frame consists of all the information that we need to analyze. We do not have any null values in our dataset and, also no duplicate data are available in our dataset. So, from this data cleaning part 1. We are tried to remove the unnecessary columns from our dataset. 2. We convert uppercase letter to lowercase of particular column. 3. We changed the Segment column to title case 4. We checked any null values is present in our dataset or not. 5. We checked any duplicate data is present in our dataset or not. So, we cleaned the complete dataset. This is all about our data cleaning part.

US Ecommerce Statistics

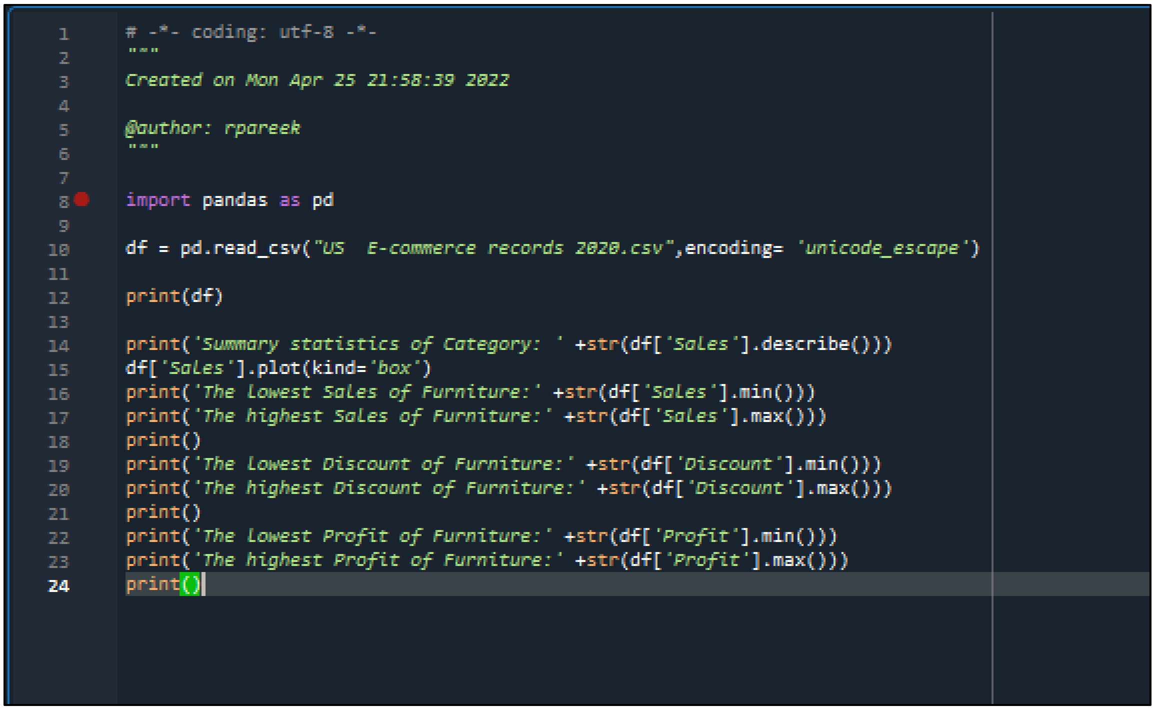
Ecommerce stands for electronic commerce and refers to a digital platform and a business model where you can buy or sell products online. Every time you purchase a product online, you're participating in the ecommerce economy. Ecommerce is a subset of e-business. It covers particularly the sales and purchases made on the internet, while e-business involves any online business activity, including sales calls, procurement of materials, signing contracts, and so on. To find the lowest and highest Sales, Profit , Discount, I used **min and max functions** in python. The code and results are shown below.

# **Show the Min/Max values of the Sales, Profit and Discount on Furniture**

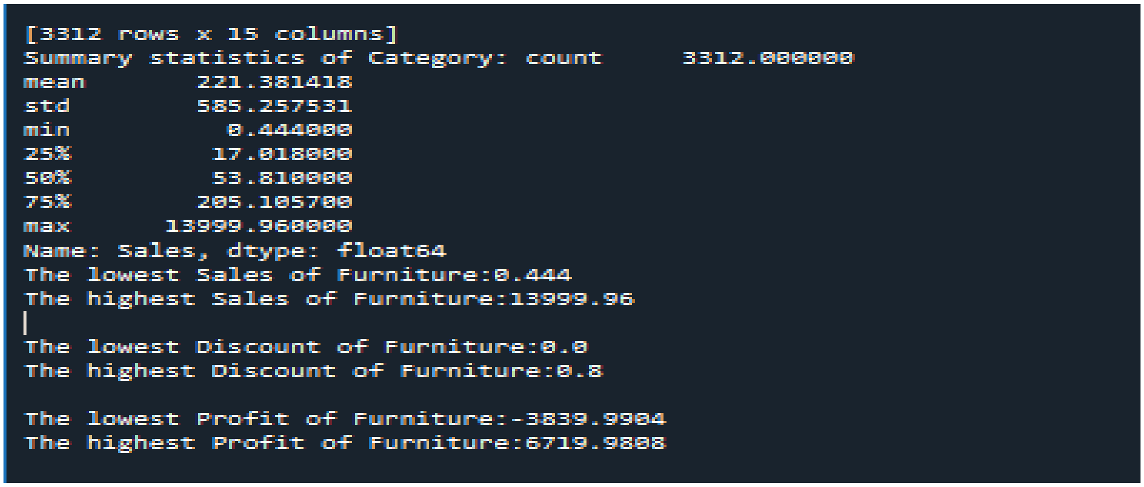
To find the summary statistics of the Sales, Discount, Profit I use '**Describe'** function in python. I also want to visualize the summary statistics to see how it looks like in a graph by using the box plot.

Code:

```
# Show the Min/Max values of the Sales, Profit and Discount on Furniture

import pandas as pd

df = pd.read_csv("./US_E-commerce_records _2020.csv", encoding = 'unicode_escape')

print(df)

print('Summary statistics of category:' +str(df['Sales'].describe()))

df['Sales'].plot(kind='box')

print('The Lowest Sales of Furniture:' +str(df['Sales'].min()))

print('The Highest Sales of Furniture:' +str(df['Sales'].max()))

print()

print('The Lowest Discount of Furniture:' +str(df['Discount'].min()))

print('The Highest Discount of Furniture:' +str(df['Discount'].max()))

print()

print('The Lowest Profit of Furniture:' +str(df['Profit'].min()))

print('The Highest Profit of Furniture:' +str(df['Profit'].max()))

print()
```



```python
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 25 21:58:39 2022

@author: rpareek
"""

import pandas as pd

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')

print(df)

print('Summary statistics of Category: ' +str(df['Sales'].describe()))
df['Sales'].plot(kind='box')
print('The lowest Sales of Furniture:' +str(df['Sales'].min()))
print('The highest Sales of Furniture:' +str(df['Sales'].max()))
print()
print('The lowest Discount of Furniture:' +str(df['Discount'].min()))
print('The highest Discount of Furniture:' +str(df['Discount'].max()))
print()
print('The lowest Profit of Furniture:' +str(df['Profit'].min()))
print('The highest Profit of Furniture:' +str(df['Profit'].max()))
print()
```



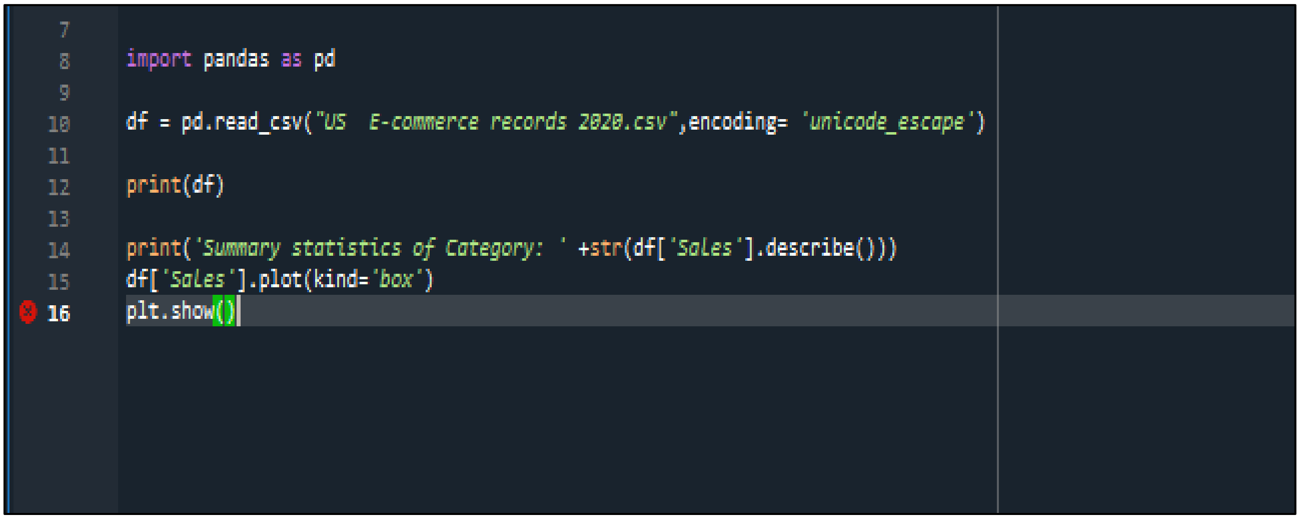
```
[3312 rows x 15 columns]
Summary statistics of Category: count     3312.000000
mean       221.381418
std        585.257531
min          0.444000
25%         17.018000
50%         53.810000
75%        205.105700
max      13999.960000
Name: Sales, dtype: float64
The lowest Sales of Furniture:0.444
The highest Sales of Furniture:13999.96

The lowest Discount of Furniture:0.0
The highest Discount of Furniture:0.8

The lowest Profit of Furniture:-3839.9904
The highest Profit of Furniture:6719.9808
```

The above results show that from **sales, discount, and Profit**. The lowest discount of furniture is 0.0 and highest discount of furniture is 0.8. The lowest sales of furniture is 0.44 and highest sales of furniture is 13999. The lowest profit of furniture is 3839 and highest profit of furniture is 6719.

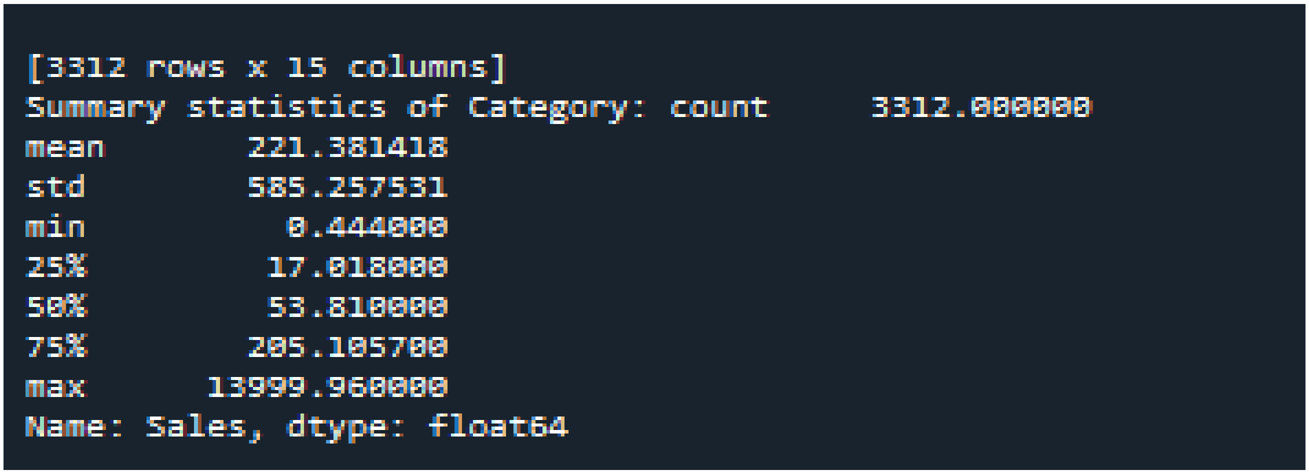
Show the Summary Statistics of the Sales

Code:

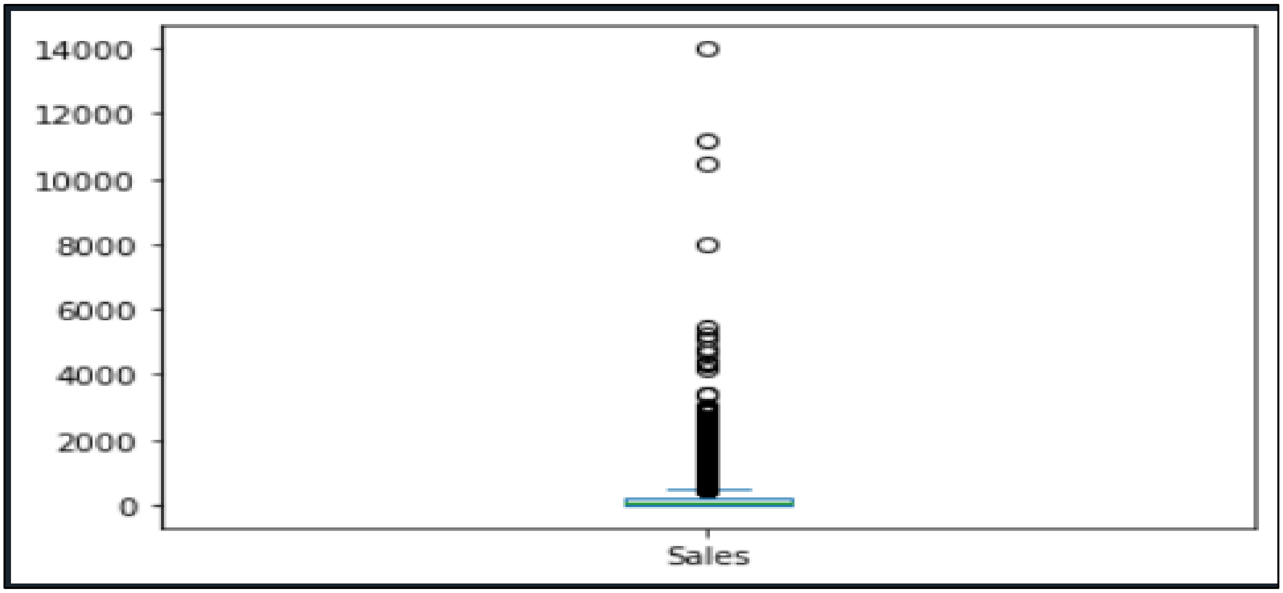
```
#Show the Summary Statistics of the Sales

import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("./US_E-commerce_records _2020.csv", encoding = 'unicode_escape')
print(df)

print('Summary statistics of category:' +str(df['Sales'].describe()))
df['Sales'].plot(kind = 'box')
plt.show()
```



```

import pandas as pd

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')

print(df)

print('Summary statistics of Category:  ' +str(df['Sales'].describe()))
df['Sales'].plot(kind='box')
plt.show()
```



```
[3312 rows x 15 columns]
Summary statistics of Category: count     3312.000000
mean        221.381418
std         585.257531
min           0.444000
25%          17.018000
50%          53.810000
75%         205.105700
max       13999.960000
Name: Sales, dtype: float64
```

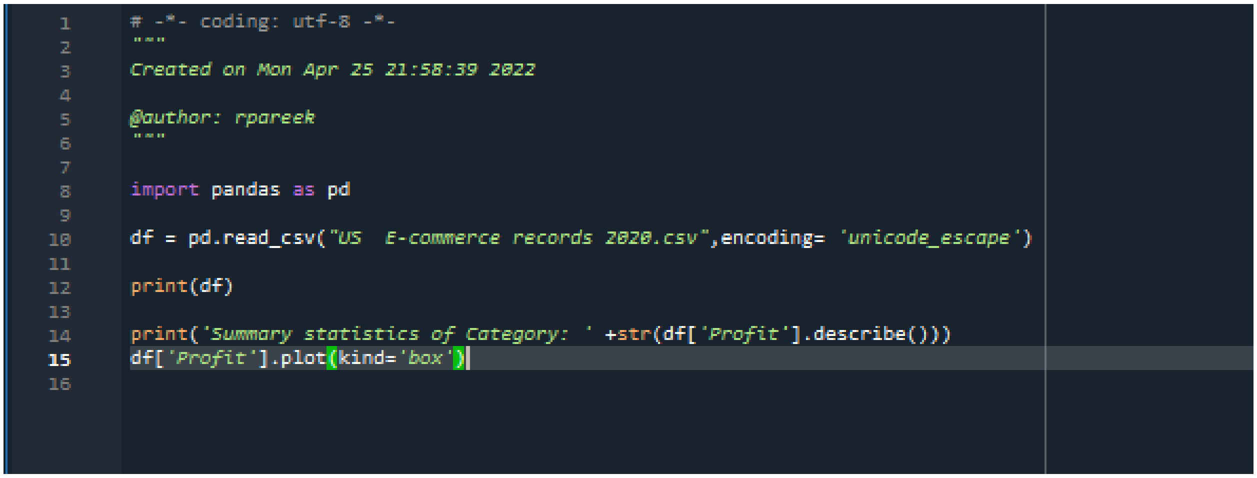


The above summary statistics shows the count of 3312. Which is number of rows in the dataset. The summary statistics also shows the **mean, standard deviation, min value, max value,** the **percentiles** of 25%,50%,75% and so on. The **box plot** shows the min and max value of sales per category by using the lowest and highest horizontal bars. The green bar inside the rectangle box shows the mean of the sales. I also use the same '**describe'** function for discount and profit.

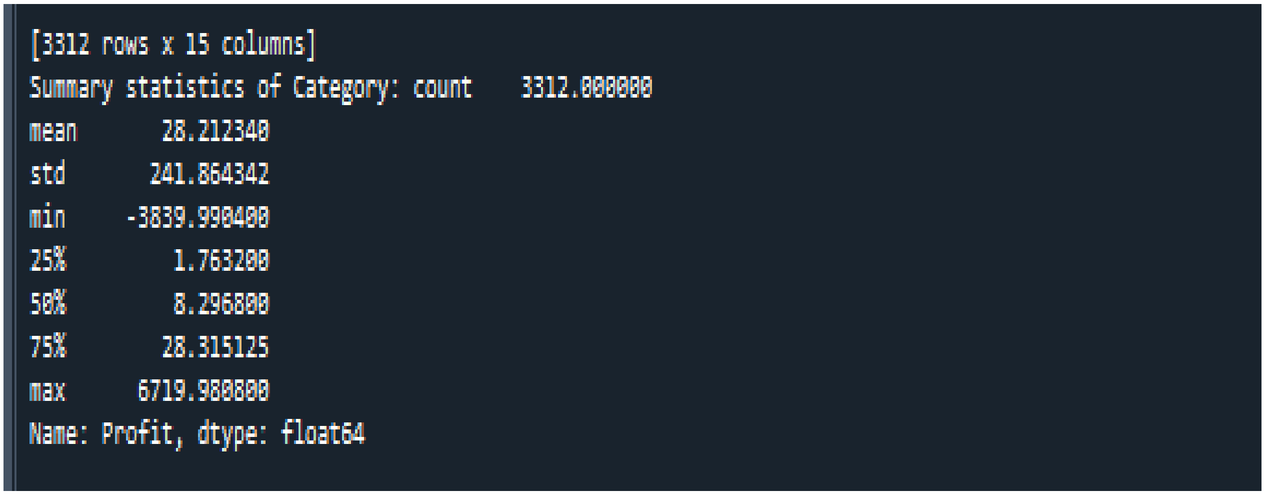
# Show the Summary Statistics of Profit

```
import pandas as pd

df = pd.read_csv("./US_E-commerce_records _2020.csv", encoding = 'unicode_escape')

print(df)

print('Summary statistics of category:' +str(df['Profit'].describe()))

df['Profit'].plot(kind = 'box')
```

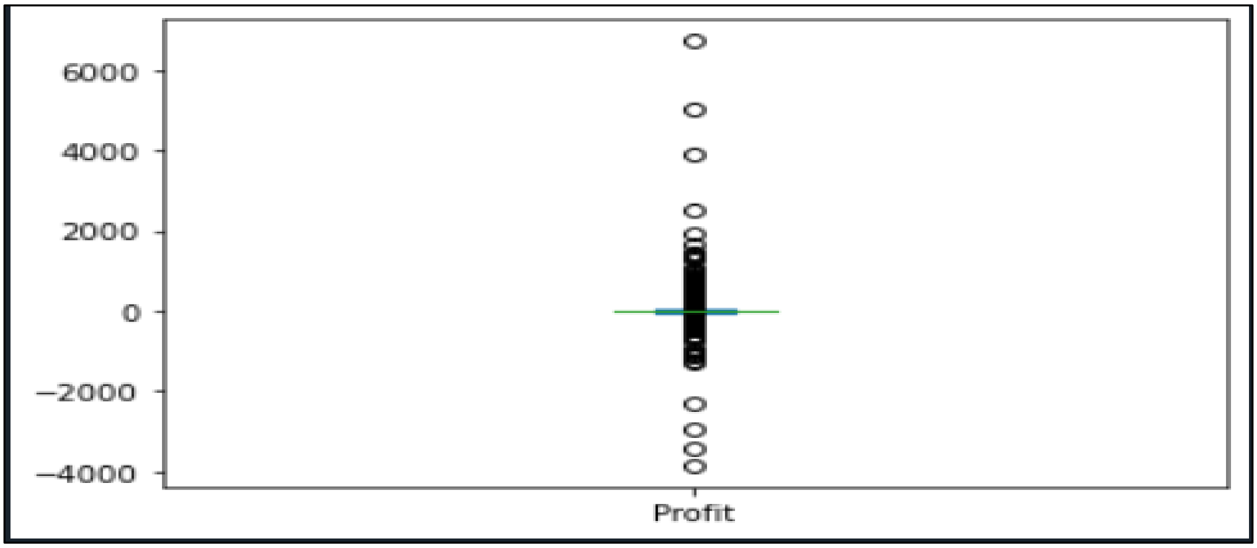
Screenshot



```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 25 21:58:39 2022

@author: rpareek
"""


import pandas as pd

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')

print(df)

print('Summary statistics of Category: ' +str(df['Profit'].describe()))
df['Profit'].plot(kind='box')

```



```
[3312 rows x 15 columns]
Summary statistics of Category: count    3312.000000
mean       28.212340
std       241.864342
min     -3839.990400
25%         1.763200
50%         8.296800
75%        28.315125
max      6719.980800
Name: Profit, dtype: float64
```

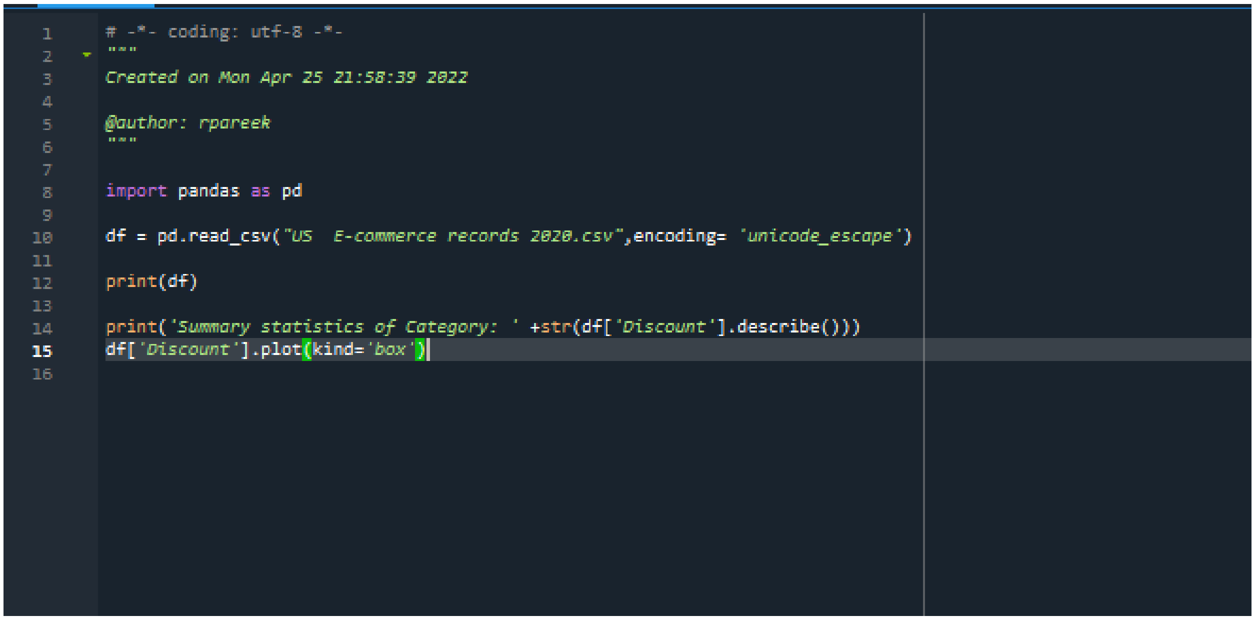


The above summary statistics shows the count of 3312. Which is number of rows in the dataset. The summary statistics also shows the **mean, standard deviation, min value, max value,** the **percentiles** of 25%,50%,75% and so on. The **box plot** shows the min and max value of profit per category by using the lowest and highest horizontal bars. The green bar inside the rectangle box shows the mean of the sales. I also use the same **'describe'** function for discount.

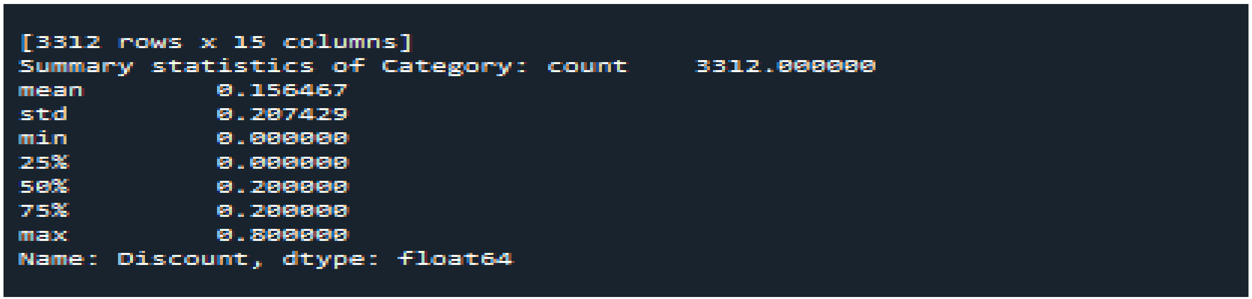
# Show the Summary Statistics of Discount

Code:

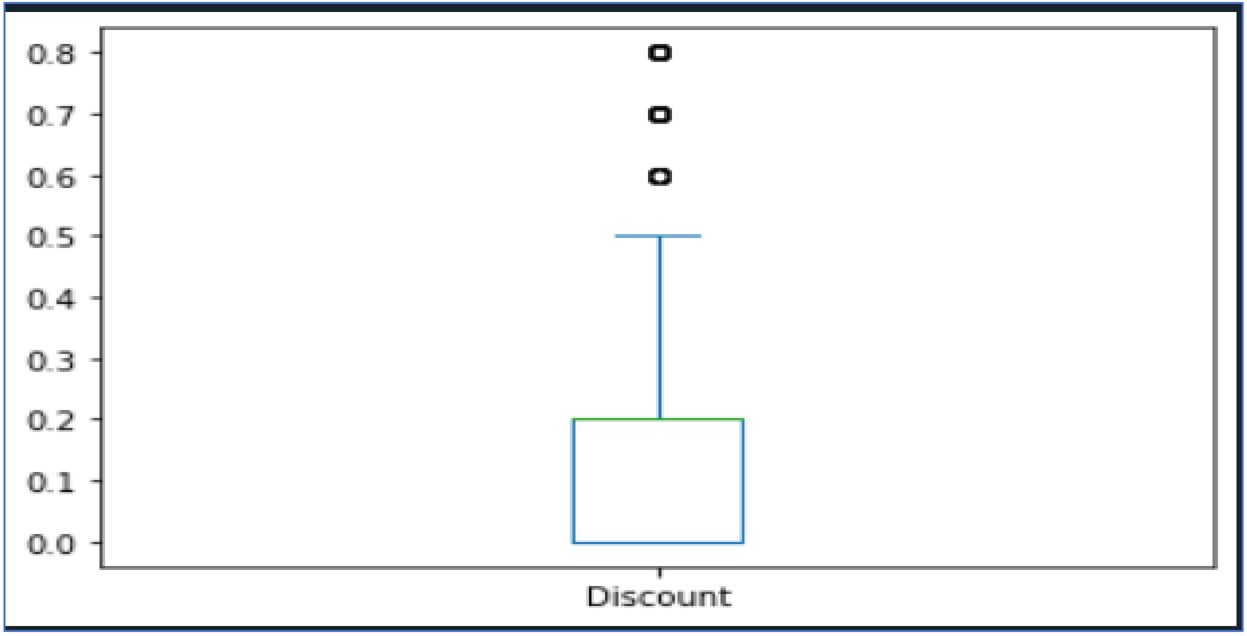
```
# Show the Summary Statistics of  Discount

import pandas as pd

df = pd.read_csv("./US_E-commerce_records _2020.csv", encoding = 'unicode_escape')
print(df)

print('Summary statistics of category:' +str(df['Discount'].describe()))
df['Discount'].plot(kind = 'box')
```



```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 25 21:58:39 2022

@author: rpareek
"""

import pandas as pd

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')

print(df)

print('Summary statistics of Category: ' +str(df['Discount'].describe()))
df['Discount'].plot(kind='box')

```



```
[3312 rows x 15 columns]
Summary statistics of Category: count     3312.000000
mean          0.156467
std           0.207429
min           0.000000
25%           0.000000
50%           0.200000
75%           0.200000
max           0.800000
Name: Discount, dtype: float64
```



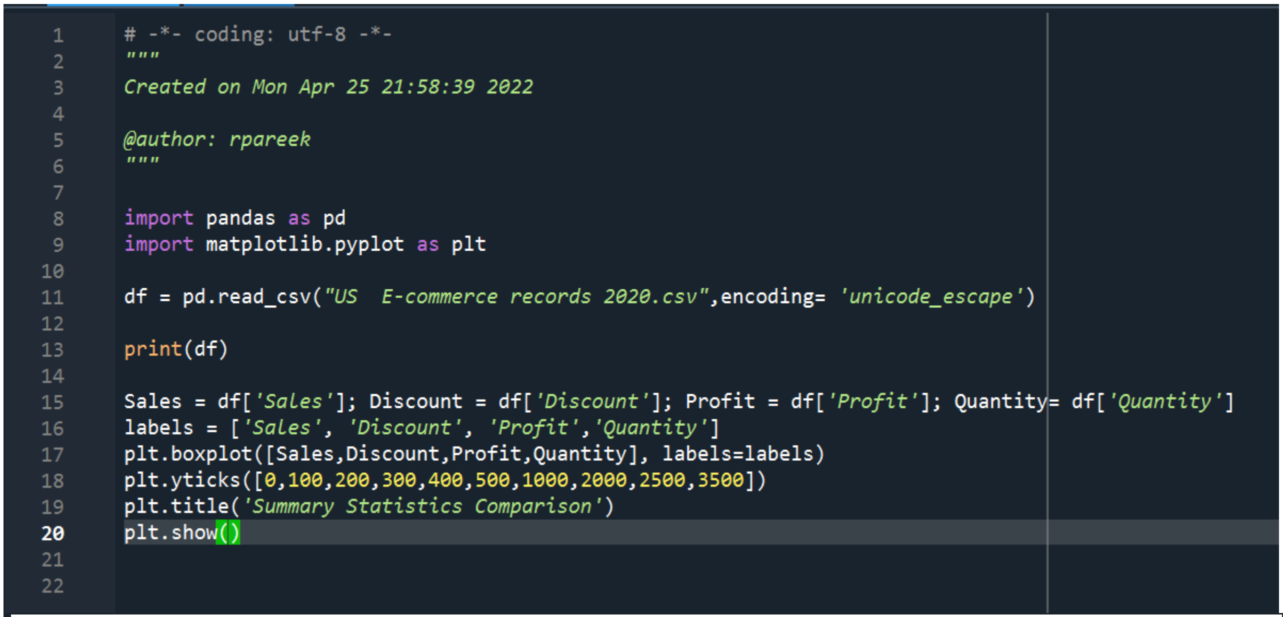
The above summary statistics shows the count of 3312. Which is number of rows in the dataset. The summary statistics also shows the **mean, standard deviation, min value, max value,** the **percentiles** of 25%,50%,75% and so on. The **box plot** shows the min and max value of **Discount** per category by using the lowest and highest horizontal bars. I use the ‘**describe’** function for discount.

<u>How summary statistics of sales, profit, discount, quantity look like in one chart</u>.

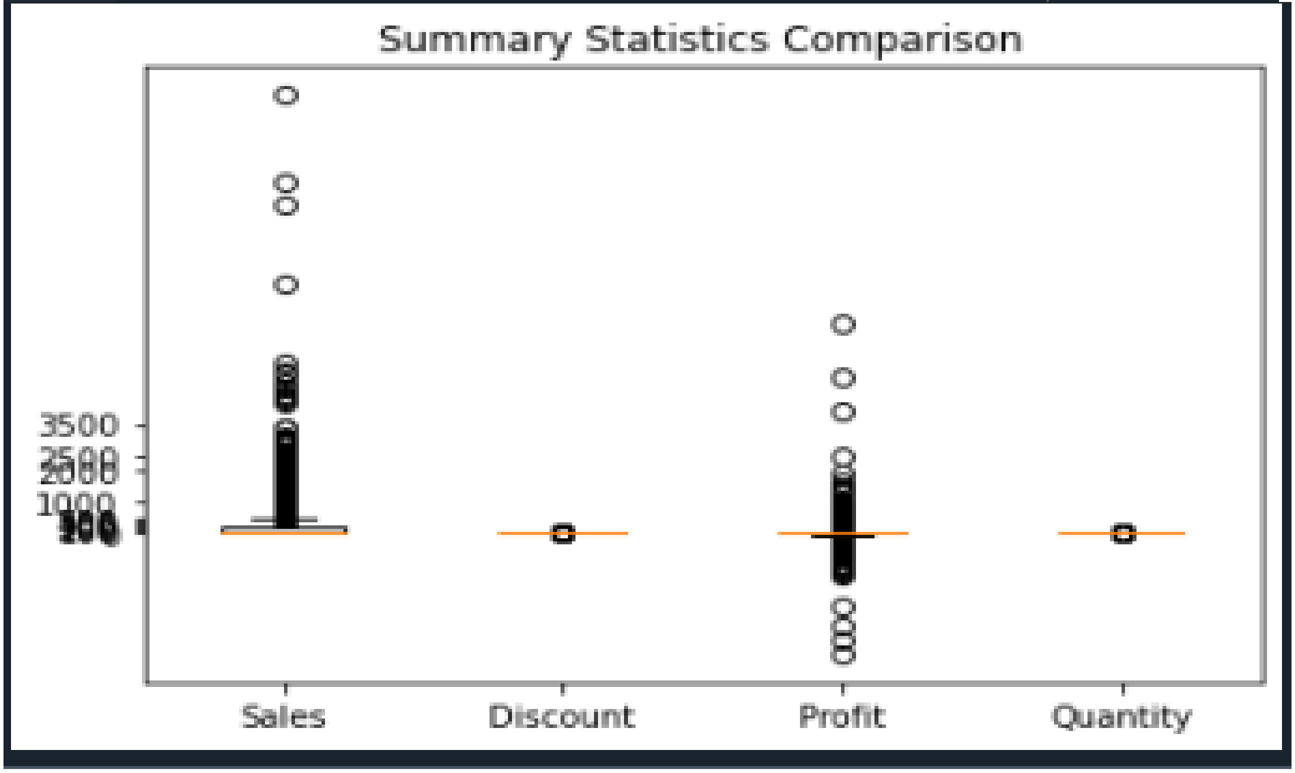
The below summary Statistics shows the Comparison between Sales, Discount, Profit and Quantity. We used **box plot** for displaying the comparison.

Code:

```
# How summary statistics of sales, profit, discount, quantity look like in one chart

import pandas as pd

import matplotlib.pyplot as plt

df = pd.read_csv("./US_E-commerce_records _2020.csv", encoding = 'unicode_escape')

print(df)

Sales = df['Sales']; Discount = df['Discount']; Profit = df['Profit']; Quantity = df['Quantity']

labels = ['Sales','Discount','Profit','Quantity']

plt.boxplot([Sales,Discount,Profit,Quantity], labels=labels)

plt.yticks([0,100,200,300,400,500,1000,2000,2500,3500])

plt.title('Summary Statistics Comparison')

plt.show()
```



```python
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 25 21:58:39 2022

@author: rpareek
"""

import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')

print(df)

Sales = df['Sales']; Discount = df['Discount']; Profit = df['Profit']; Quantity= df['Quantity']
labels = ['Sales', 'Discount', 'Profit','Quantity']
plt.boxplot([Sales,Discount,Profit,Quantity], labels=labels)
plt.yticks([0,100,200,300,400,500,1000,2000,2500,3500])
plt.title('Summary Statistics Comparison')
plt.show()


```



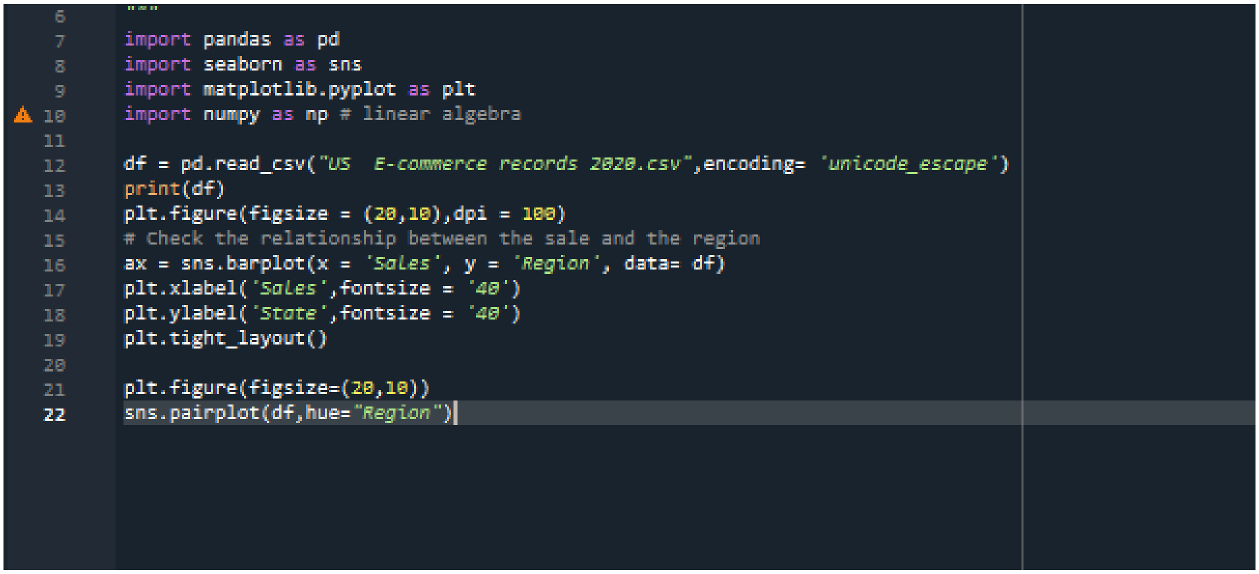
This summary statistics is showing very good comparison between the Sales, Discount, Profit and Quantity. We used **Box plot** for displaying the comparison.

Data Visualization

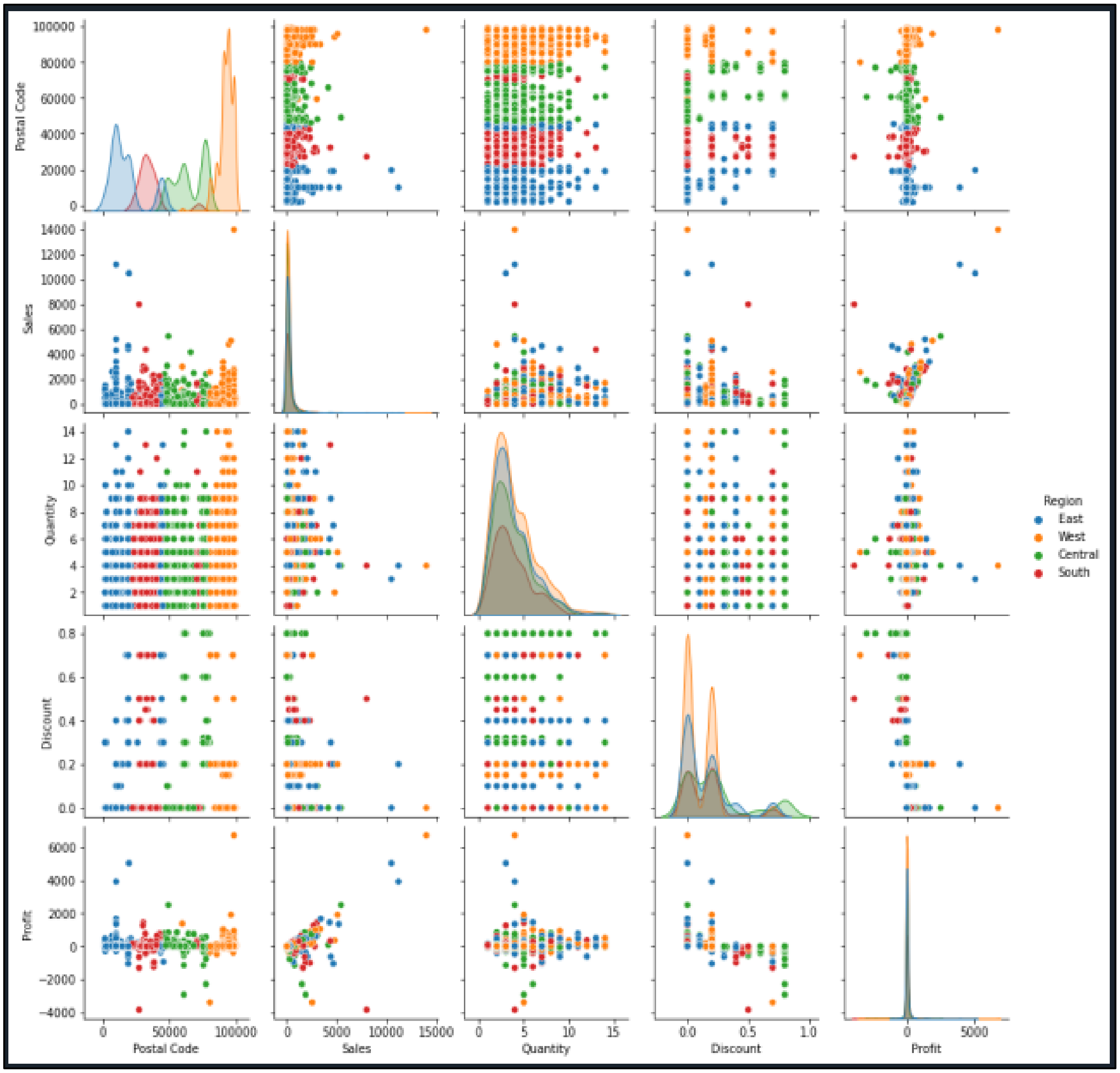
#Pair plot for a region wise see whether a bit overwhelming.

Screenshot and Code

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np # linear algebra

df = pd.read_csv("US E-commerce records 2020.csv",encoding=
'unicode_escape')
print(df)
plt.figure(figsize = (20,10),dpi = 100)
# Check the relationship between the sale and the region
ax = sns.barplot(x = 'Sales', y = 'Region', data= df)
plt.xlabel('Sales',fontsize = '40')
plt.ylabel('State',fontsize = '40')
plt.tight_layout()

plt.figure(figsize=(20,10))
sns.pairplot(df,hue="Region")
```



```
"""
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np # linear algebra

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')
print(df)
plt.figure(figsize = (20,10),dpi = 100)
# Check the relationship between the sale and the region
ax = sns.barplot(x = 'Sales', y = 'Region', data= df)
plt.xlabel('Sales',fontsize = '40')
plt.ylabel('State',fontsize = '40')
plt.tight_layout()

plt.figure(figsize=(20,10))
sns.pairplot(df,hue="Region")
```



Application Used: Pair plot

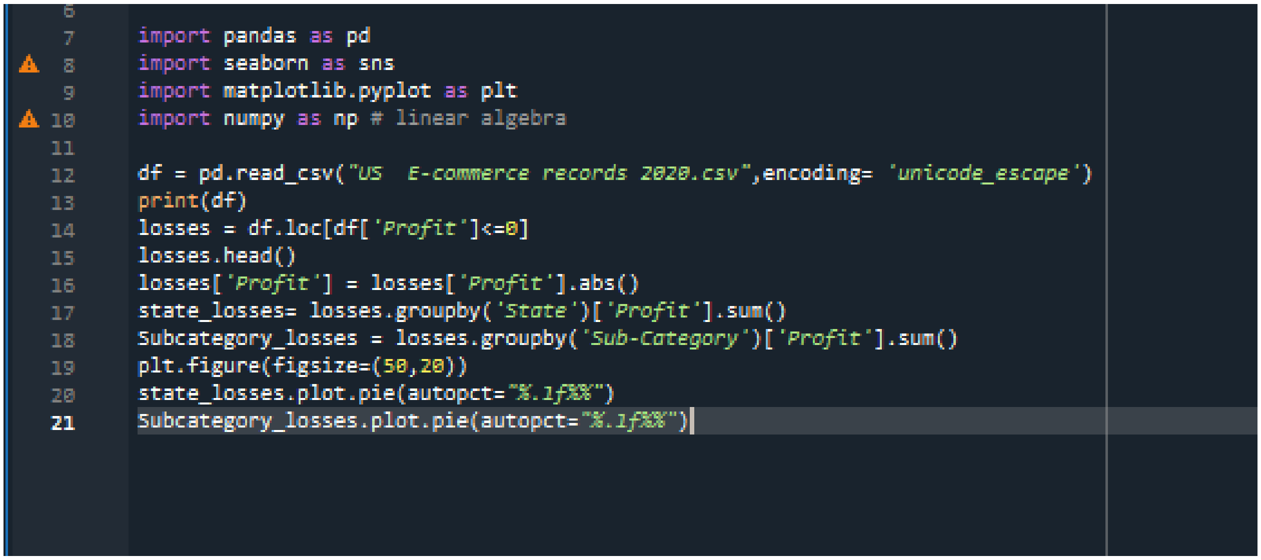
From this **Pair plot** we can see the Profit, discount, Quantity, Sales, and postal code per region. In the region we can see **East, west, central, and south** locations. From this pair chart we can see, in the east location, how much profit the company has earned based on quantity, discount and sales. So based on the different- different region we can see the US Ecommerce record. For example, west location, south location, and central location how much profit US ecommerce has earned in year 2020. So, from this pair plot we can see sales and profit of US Ecommerce market.

#Which states make for losses for category and subcategory?

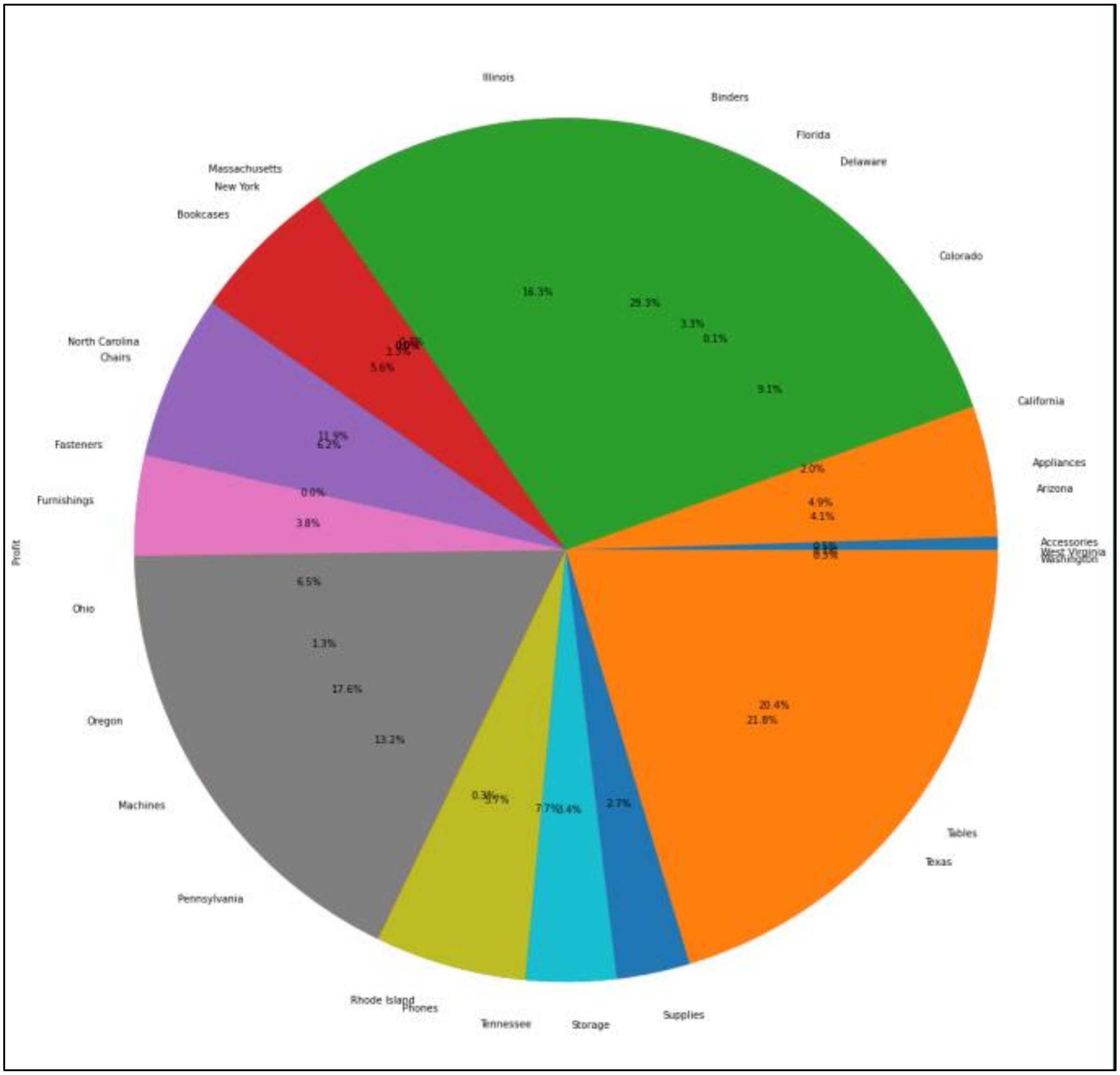
Screenshot and Code:

Checking the Relationship between Sub-category and loses

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np # linear algebra

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')
print(df)
losses = df.loc[df['Profit']<=0]
losses.head()
losses['Profit'] = losses['Profit'].abs()
state_losses= losses.groupby('State')['Profit'].sum()
Subcategory_losses = losses.groupby('Sub-Category')['Profit'].sum()
plt.figure(figsize=(50,20))
state_losses.plot.pie(autopct="%.1f%%")
Subcategory_losses.plot.pie(autopct="%.1f%%")
```



```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np # linear algebra

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')
print(df)
losses = df.loc[df['Profit']<=0]
losses.head()
losses['Profit'] = losses['Profit'].abs()
state_losses= losses.groupby('State')['Profit'].sum()
Subcategory_losses = losses.groupby('Sub-Category')['Profit'].sum()
plt.figure(figsize=(50,20))
state_losses.plot.pie(autopct="%.1f%%")
Subcategory_losses.plot.pie(autopct="%.1f%%")
```

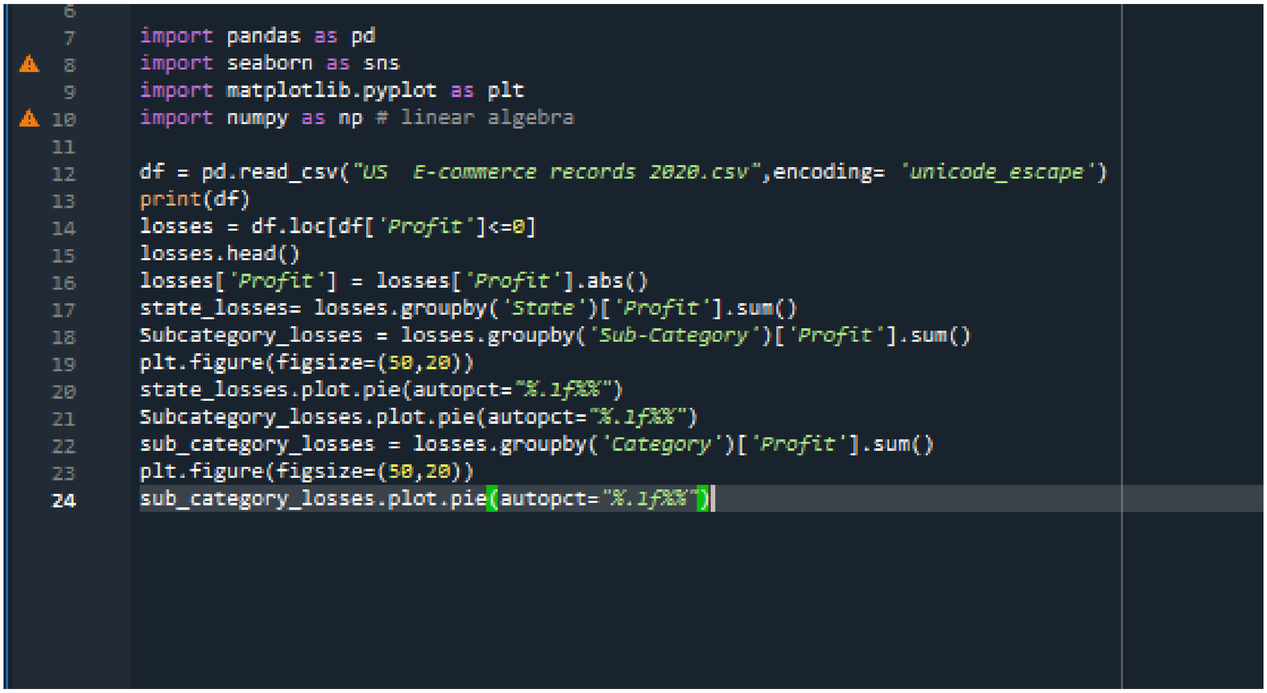


The above **Pie chart** visually shows the loss of state as per the sub- category. In above chart different colors are indicating positive and negative result. These colors are basically indicating which states make profit and which states make loss as per the sub- category. So, in above chart **green color** is showing the positive result, as per the **green color states** (Florida, Colorado, Illinois) gain the profit. And on the other side red color is indicating the negative result so, states (New York, Massachusetts) they are showing in **red** color, means these states make losses as per the sub- category.

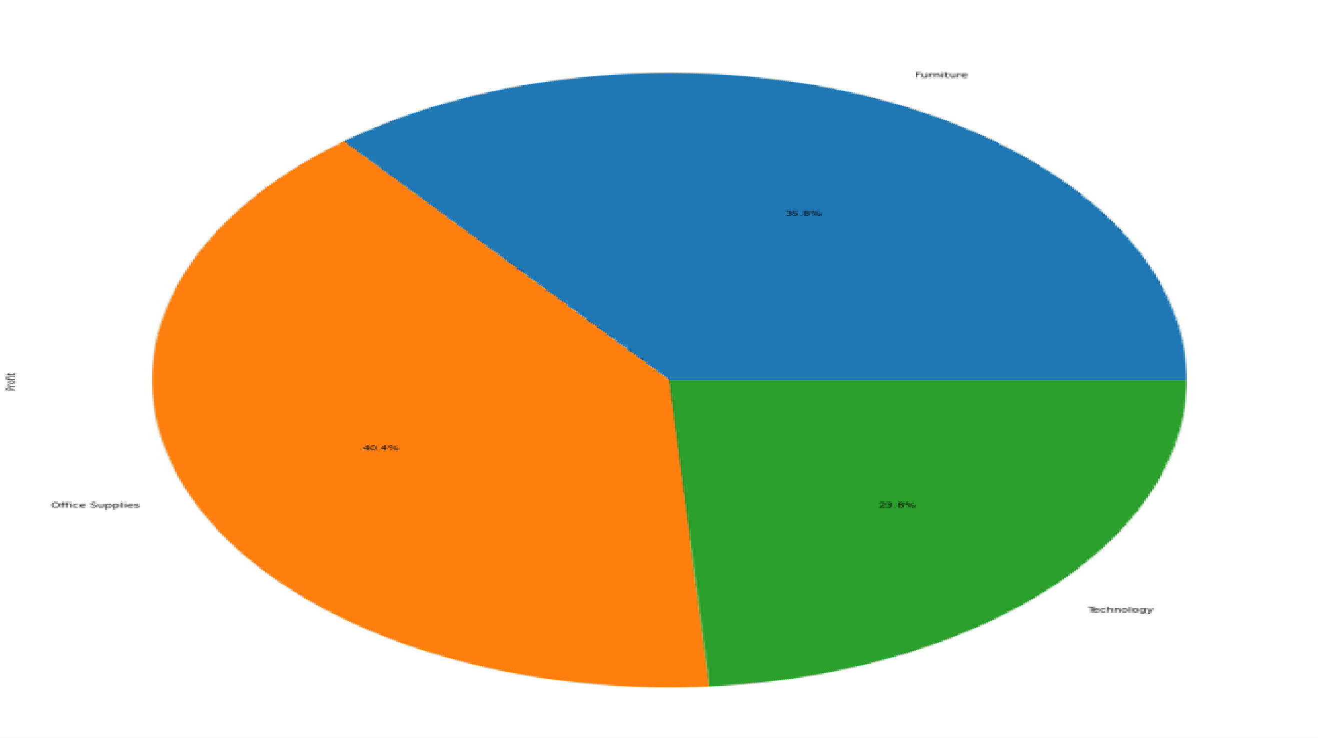
Checking the Relationship between category and loses

Code:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np # linear algebra

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')
print(df)
losses = df.loc[df['Profit']<=0]
losses.head()
losses['Profit'] = losses['Profit'].abs()
state_losses= losses.groupby('State')['Profit'].sum()
Subcategory_losses = losses.groupby('Sub-Category')['Profit'].sum()
plt.figure(figsize=(50,20))
state_losses.plot.pie(autopct="%.1f%%")
Subcategory_losses.plot.pie(autopct="%.1f%%")
sub_category_losses = losses.groupby('Category')['Profit'].sum()
plt.figure(figsize=(50,20))
sub_category_losses.plot.pie(autopct="%.1f%%")
```



```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np # linear algebra

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')
print(df)
losses = df.loc[df['Profit']<=0]
losses.head()
losses['Profit'] = losses['Profit'].abs()
state_losses= losses.groupby('State')['Profit'].sum()
Subcategory_losses = losses.groupby('Sub-Category')['Profit'].sum()
plt.figure(figsize=(50,20))
state_losses.plot.pie(autopct="%.1f%%")
Subcategory_losses.plot.pie(autopct="%.1f%%")
sub_category_losses = losses.groupby('Category')['Profit'].sum()
plt.figure(figsize=(50,20))
sub_category_losses.plot.pie(autopct="%.1f%%")
```



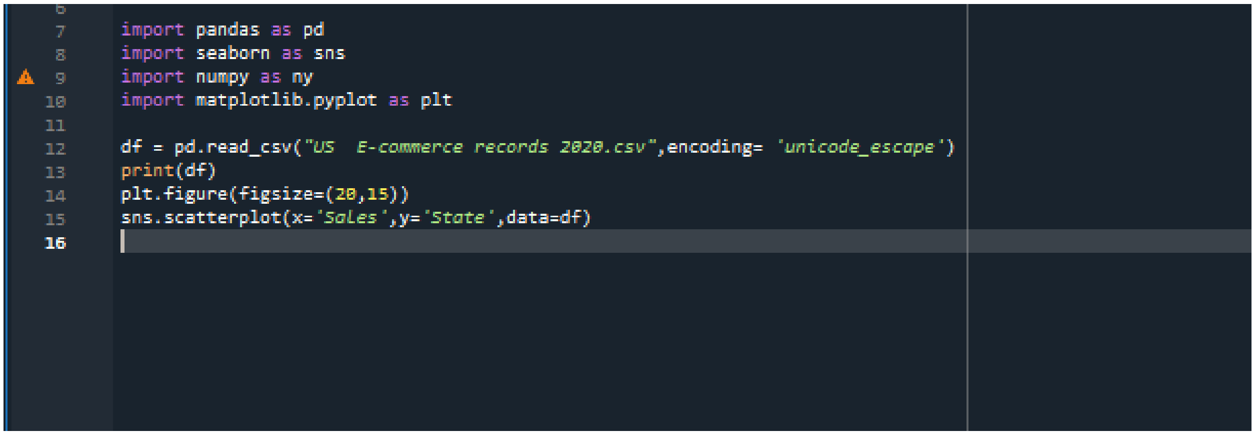
<u>Application Used</u>: **<u>Pie chart</u>**

The above **Pie chart** visually shows the loss of state as per the category. In above chart different colors are indicating positive and negative result. In above chart **green, orange and blue color** is indicating different – different result. Technology category shows the positive result in terms of profit and orange Category (office suppliers) and Furniture category indicating the negative result in terms of loss.

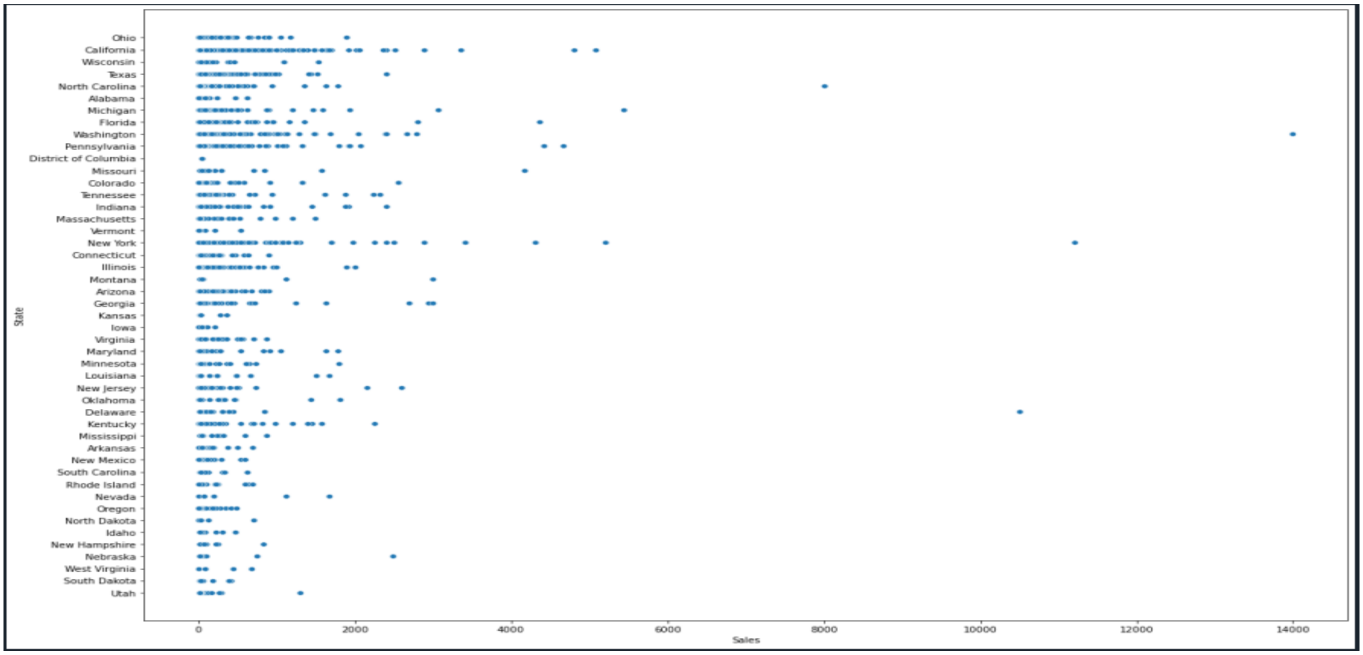
#Find out the sales, profit as per the State and discount as per the Product name

<u>Sales as per the state</u>

```
import pandas as pd
import seaborn as sns
import numpy as ny
import matplotlib.pyplot as plt

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')
print(df)
plt.figure(figsize=(20,15))
sns.scatterplot(x='Sales',y='State',data=df)
```



```
import pandas as pd
import seaborn as sns
import numpy as ny
import matplotlib.pyplot as plt

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')
print(df)
plt.figure(figsize=(20,15))
sns.scatterplot(x='Sales',y='State',data=df)
```



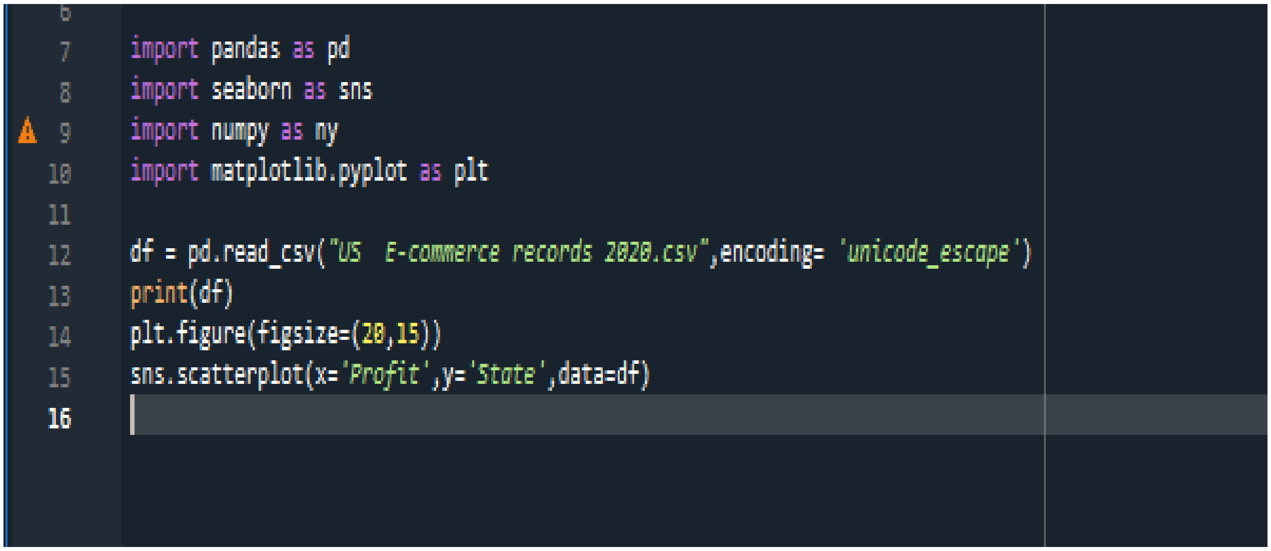
Application used: Scatter plot

In above **Scatter plot,** it is showing what is the total sales of particular state. Different- different states name and sales data are showing in the chart. The above chart also showing which states has sold maximum products.

Profit As per the State

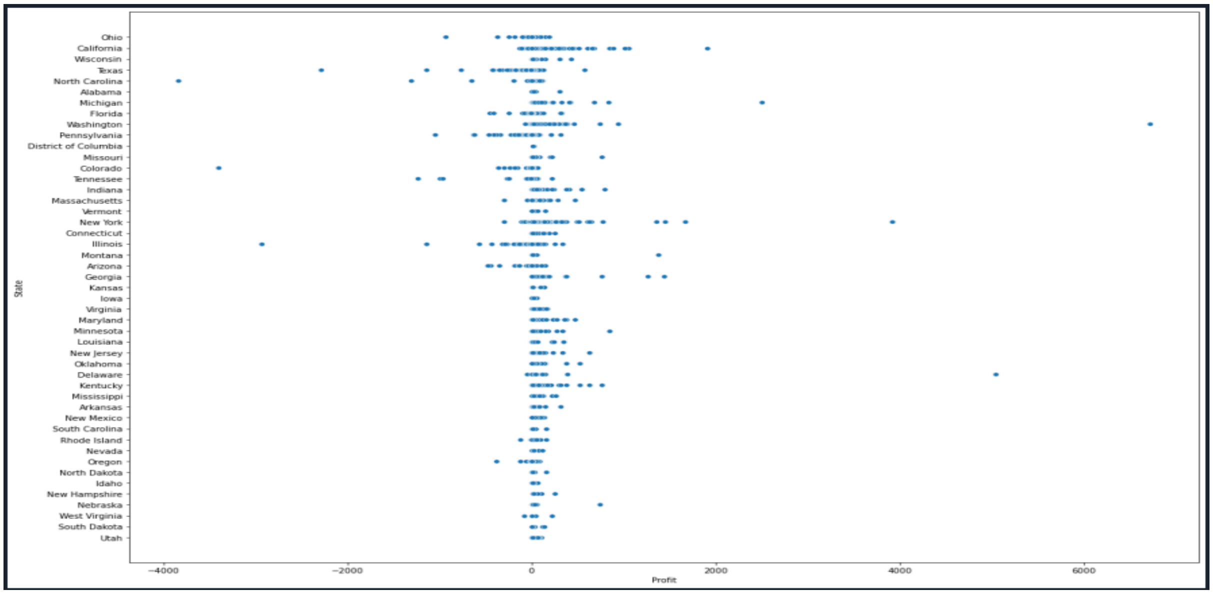
Code:

```
import pandas as pd
import seaborn as sns
import numpy as ny
import matplotlib.pyplot as plt

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')
print(df)
plt.figure(figsize=(20,15))
sns.scatterplot(x='Profit',y='State',data=df)
```



```
import pandas as pd
import seaborn as sns
import numpy as ny
import matplotlib.pyplot as plt

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')
print(df)
plt.figure(figsize=(20,15))
sns.scatterplot(x='Profit',y='State',data=df)

```

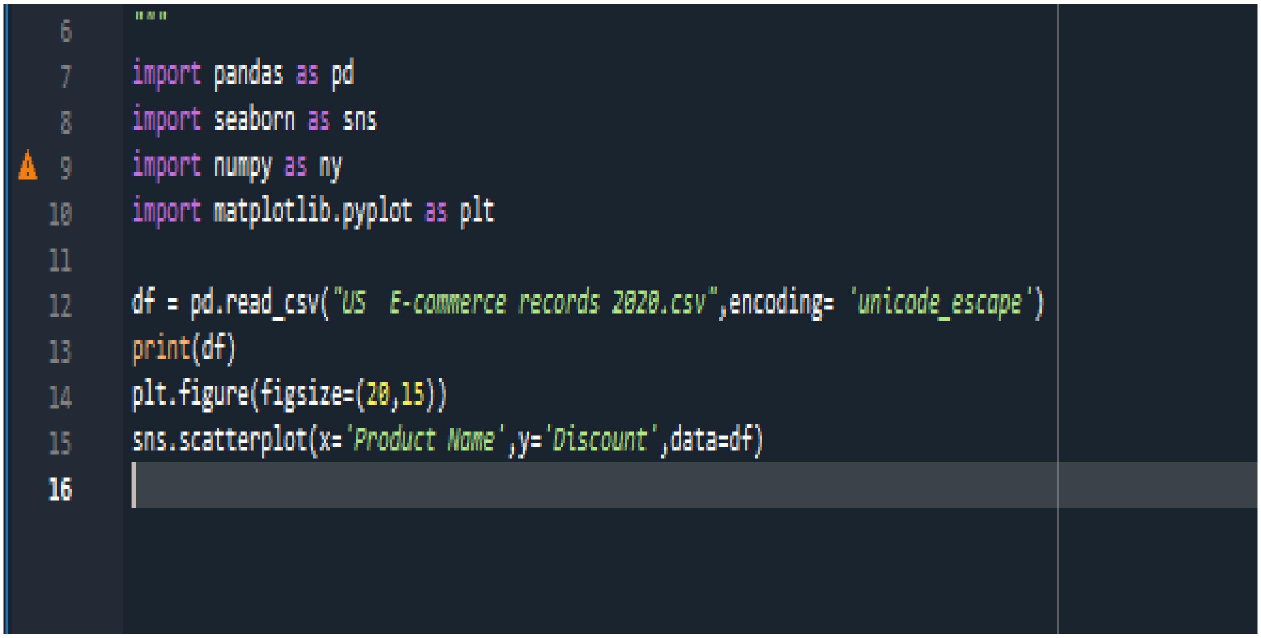
In below **scatter plot** it is showing what is the total profit of state. Different states name and profit data are showing in the chart. This chart also indicating which states make maximum profit or revenue.



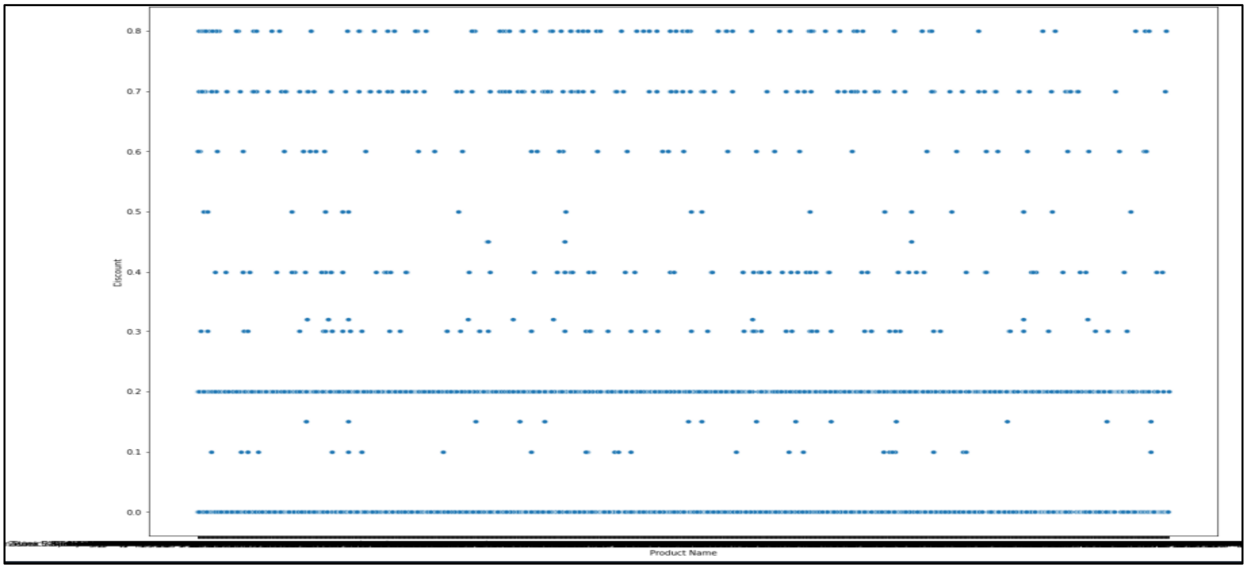
Application used: Scatter plot

Discount As per the Product

```
import pandas as pd

import seaborn as sns

import numpy as ny

import matplotlib.pyplot as plt

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')

print(df)

plt.figure(figsize=(20,15))

sns.scatterplot(x='Product Name',y='Discount',data=df)
```



```
"""
import pandas as pd
import seaborn as sns
import numpy as ny
import matplotlib.pyplot as plt

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')
print(df)
plt.figure(figsize=(20,15))
sns.scatterplot(x='Product Name',y='Discount',data=df)
```

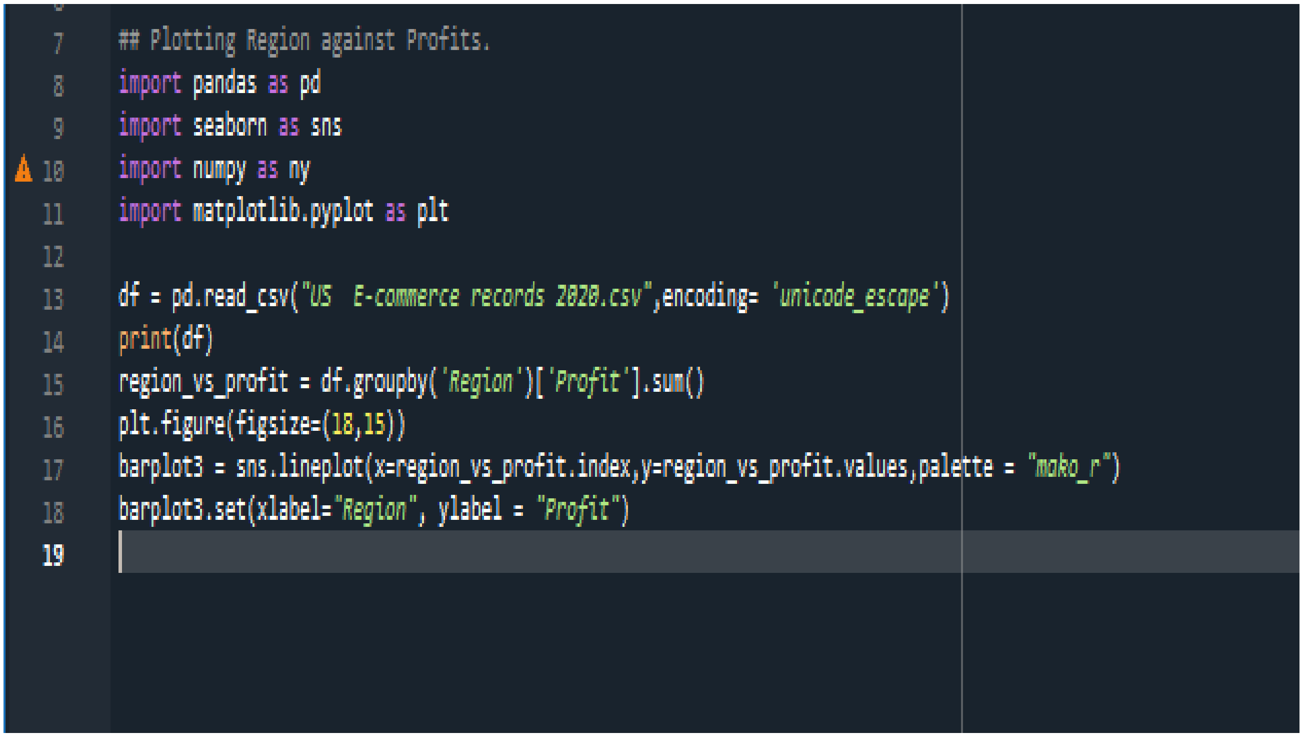


Application Used: Scatter plot

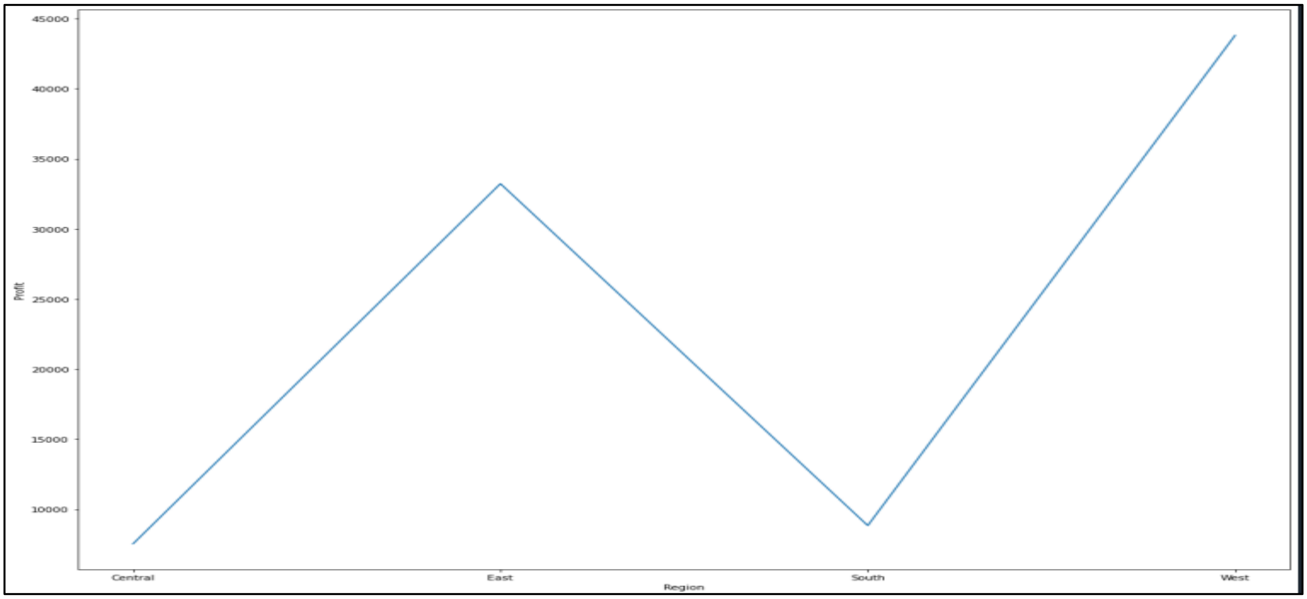
In above **scatter plot** it is showing how much discount is available on particular product. Different-different product name and discount data are showing in the chart. This chart also indicating on which particular product maximum discount and minimum discount is available.

#Plotting Region against Profits.

```
import pandas as pd
import seaborn as sns
import numpy as ny
import matplotlib.pyplot as plt

df = pd.read_csv("US  E-commerce records 2020.csv",encoding=
'unicode_escape')
print(df)
region_vs_profit = df.groupby('Region')['Profit'].sum()
plt.figure(figsize=(18,15))
barplot3 = sns.lineplot(x=region_vs_profit.index,y=region_vs_profit.values,palette
= "mako_r")
barplot3.set(xlabel="Region", ylabel = "Profit")
```



```
## Plotting Region against Profits.
import pandas as pd
import seaborn as sns
import numpy as ny
import matplotlib.pyplot as plt

df = pd.read_csv("US  E-commerce records 2020.csv",encoding= 'unicode_escape')
print(df)
region_vs_profit = df.groupby('Region')['Profit'].sum()
plt.figure(figsize=(18,15))
barplot3 = sns.lineplot(x=region_vs_profit.index,y=region_vs_profit.values,palette = "mako_r")
barplot3.set(xlabel="Region", ylabel = "Profit")
```

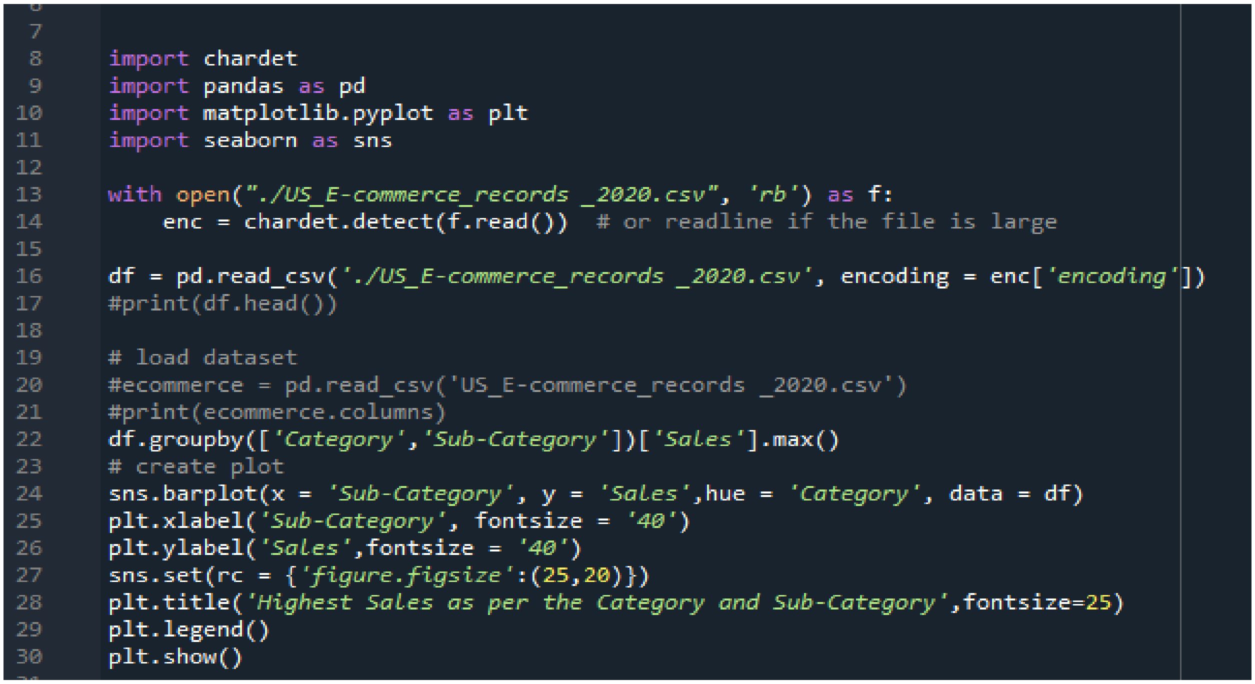


<u>Application Used: Line Plot</u>

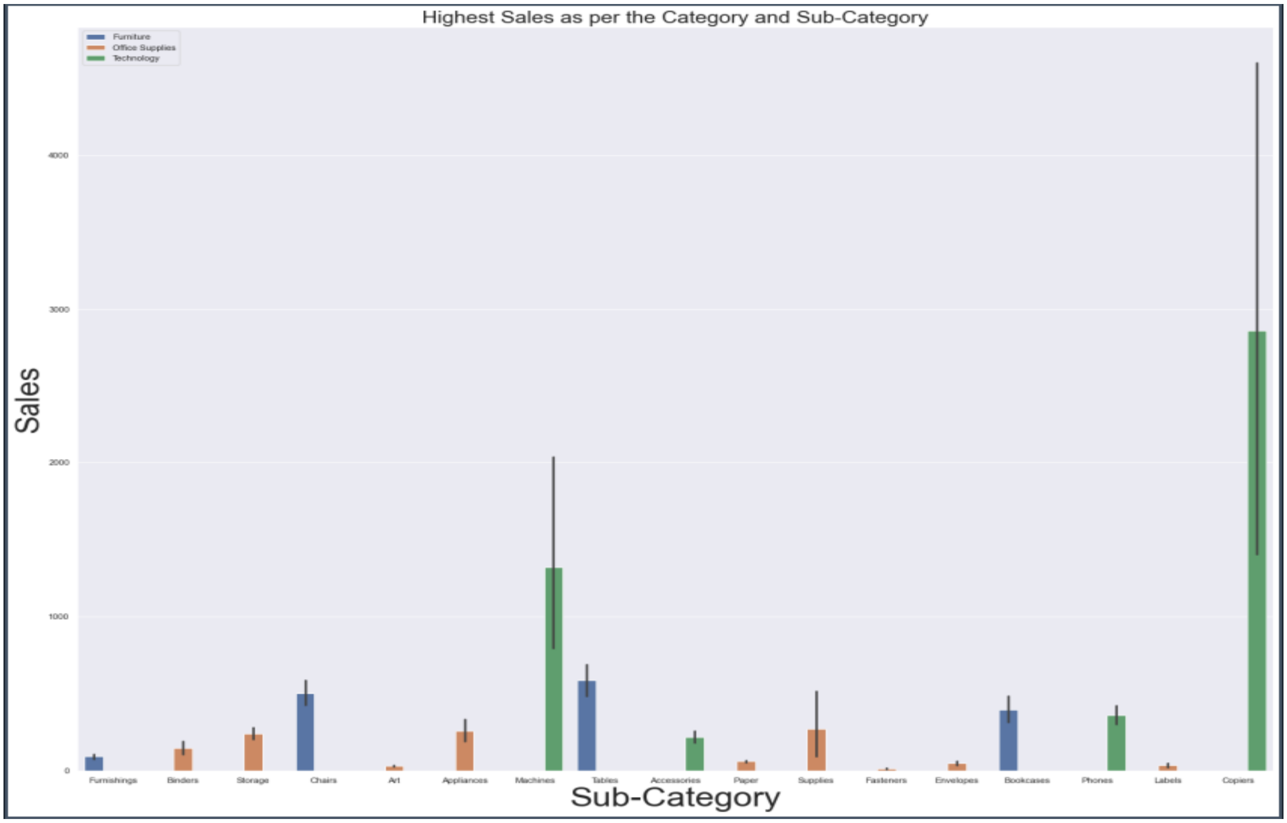
The above **line chart** visually shows the **Region against Profits**. In above chart, 4 regions are showing which are (**Central, East, South, West**) .These regions is basically indicating which region received the maximum profit and which region received the minimum profit. So, in above chart it is clearly showing **central** and **south** region gain the maximum profit and revenue, East and west region gain less profit compared to other region.

#What is the highest sales as per the category & sub-category in a given year?

```
import chardet

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

with open("./US_E-commerce_records _2020.csv", 'rb') as f:

    enc = chardet.detect(f.read())  # or readline if the file is large



df = pd.read_csv('./US_E-commerce_records _2020.csv', encoding =
enc['encoding'])

#print(df.head())



# load dataset

#ecommerce = pd.read_csv('US_E-commerce_records _2020.csv')

#print(ecommerce.columns)

df.groupby(['Category','Sub-Category'])['Sales'].max()

# create plot
```



```python

import chardet
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

with open("./US_E-commerce_records _2020.csv", 'rb') as f:
    enc = chardet.detect(f.read())  # or readline if the file is large

df = pd.read_csv('./US_E-commerce_records _2020.csv', encoding = enc['encoding'])
#print(df.head())

# load dataset
#ecommerce = pd.read_csv('US_E-commerce_records _2020.csv')
#print(ecommerce.columns)
df.groupby(['Category','Sub-Category'])['Sales'].max()
# create plot
sns.barplot(x = 'Sub-Category', y = 'Sales',hue = 'Category', data = df)
plt.xlabel('Sub-Category', fontsize = '40')
plt.ylabel('Sales',fontsize = '40')
sns.set(rc = {'figure.figsize':(25,20)})
plt.title('Highest Sales as per the Category and Sub-Category',fontsize=25)
plt.legend()
plt.show()
```



Application Used: Bar plot

The above **Bar plot** visually shows the **highest sales as per the category & sub-category in 2020**. In above chart different colors are indicating positive and negative result. These colors are basically indicating which state makes highest sales as per the category and sub- category and which state makes lowest sales as per the category and sub- category. As we can see in above bar plot different colors are indicating for category and subcategory. **Green color is for technology** category, **orange color is for office supplies**, and **blue color is for furniture category**. As we can see technology category has the highest sales so we can assign according to rank. Technology category on Number 1 Rank as per the highest sales. And 2nd rank we can assign to Furniture category and 3rd rank we can assign to office supplies category. So, from above chart we can easily get the lowest and highest sales of particular category and Sub- category.

References:

US Ecommerce Sales [Updated March 2022] | Oberlo

E-commerce surged during Covid: Groceries, sporting goods top gainers (cnbc.com)

US Ecommerce by Category 2022 - Insider Intelligence Trends, Forecasts & Statistics (emarketer.com)