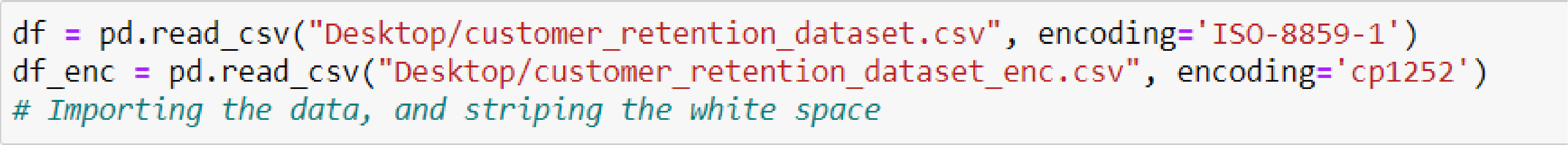
Customer retention case study

Study of online retailers

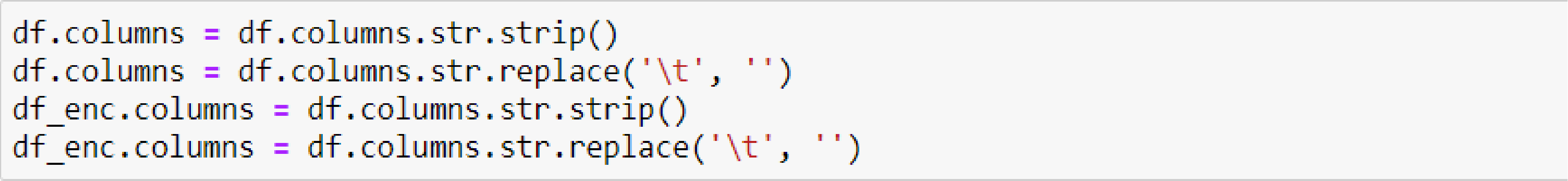
Problem statement

- We have many online retailors, the most important aspect for the retailors is to retain the customers.
- As it is more expensive to gain new customers than it is retain the current ones, retailors spend a lot of money to keep their customer loyal to their own platform
- This survey will help us dive deeper into what causes people to recommend a website/ use it again and again.

Data cleaning



```
df = pd.read_csv("Desktop/customer_retention_dataset.csv", encoding='ISO-8859-1')
df_enc = pd.read_csv("Desktop/customer_retention_dataset_enc.csv", encoding='cp1252')
# Importing the data, and striping the white space
```

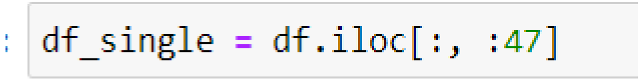


```
df.columns = df.columns.str.strip()
df.columns = df.columns.str.replace('\t', '')
df_enc.columns = df.columns.str.strip()
df_enc.columns = df.columns.str.replace('\t', '')
```

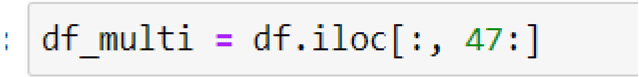
Here I’ve imported the encoded as well as the non encoded data, and cleaned the data of any White spaces as well as striping the column names of ‘\t’

Dividing the data into two parts

- df_single contains the questions which have a single answer
- df_multi contains the questions where multiple answers are allowed.



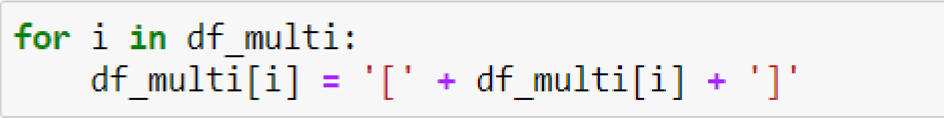
```
df_single = df.iloc[:, :47]
```



```
df_multi = df.iloc[:, 47:]
```

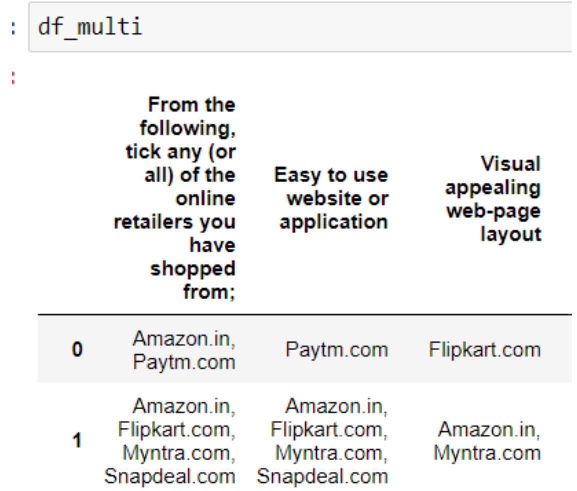
Converting df_multi responses into a list

- Now we must convert the df_multi responses into a list format before we can perform any analysis on it.



```
for i in df_multi:
    df_multi[i] = '[' + df_multi[i] + ']'
```

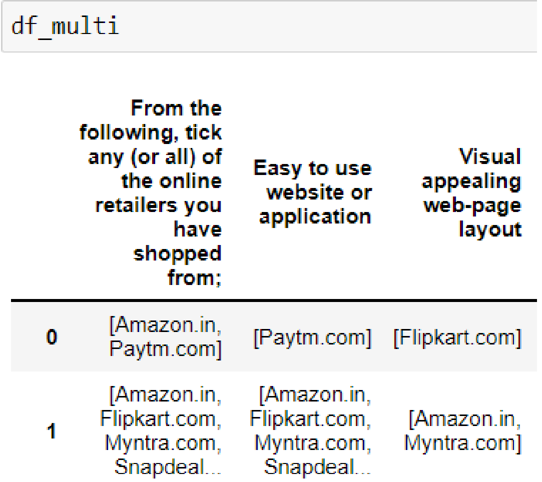
- Before



```
df_multi
```

	From the following, tick any (or all) of the online retailers you have shopped from;	Easy to use website or application	Visual appealing web-page layout
0	Amazon.in, Paytm.com	Paytm.com	Flipkart.com
1	Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com	Amazon.in, Flipkart.com, Myntra.com, Snapdeal.com	Amazon.in, Myntra.com

After

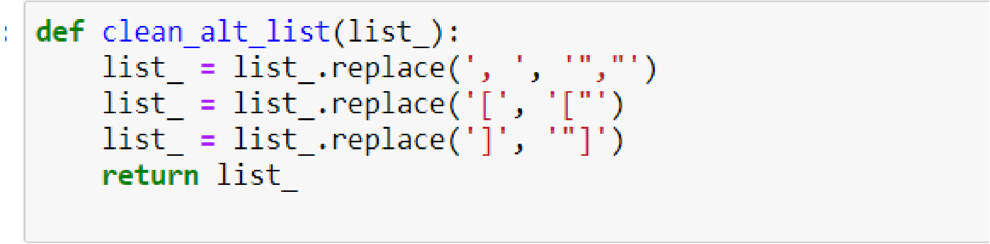


```
df_multi
```

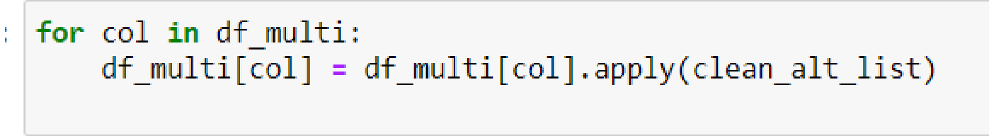
	From the following, tick any (or all) of the online retailers you have shopped from;	Easy to use website or application	Visual appealing web-page layout
0	[Amazon.in, Paytm.com]	[Paytm.com]	[Flipkart.com]
1	[Amazon.in, Flipkart.com, Myntra.com, Snapdeal...	[Amazon.in, Flipkart.com, Myntra.com, Snapdeal...	[Amazon.in, Myntra.com]

Running a script through all the coulmns

- Script

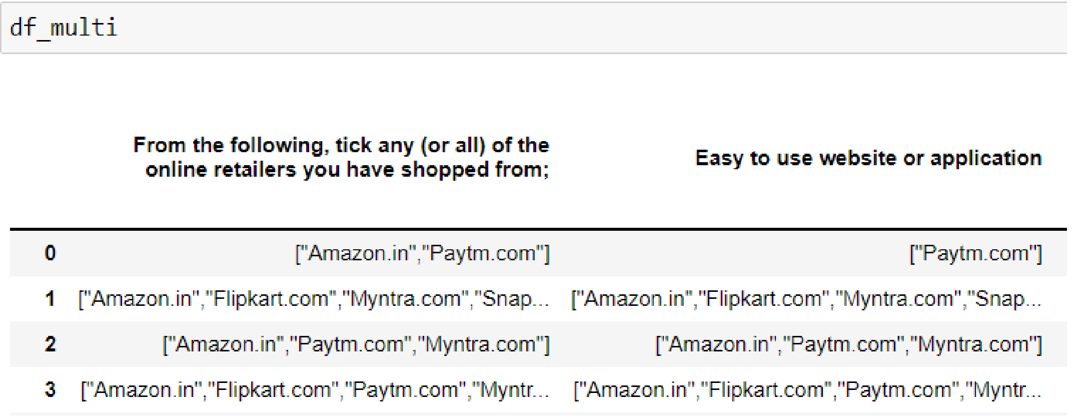


```
def clean_alt_list(list_):
    list_ = list_.replace(', ', '","')
    list_ = list_.replace('[', '["')
    list_ = list_.replace(']', '"]')
    return list_
```



```
for col in df_multi:
    df_multi[col] = df_multi[col].apply(clean_alt_list)
```

- Final result (list)



```
df_multi
```

	From the following, tick any (or all) of the online retailers you have shopped from;	Easy to use website or application
0	["Amazon.in","Paytm.com"]	["Paytm.com"]
1	["Amazon.in","Flipkart.com","Myntra.com","Snap...	["Amazon.in","Flipkart.com","Myntra.com","Snap...
2	["Amazon.in","Paytm.com","Myntra.com"]	["Amazon.in","Paytm.com","Myntra.com"]
3	["Amazon.in","Flipkart.com","Paytm.com","Myntr...	["Amazon.in","Flipkart.com","Paytm.com","Myntr...

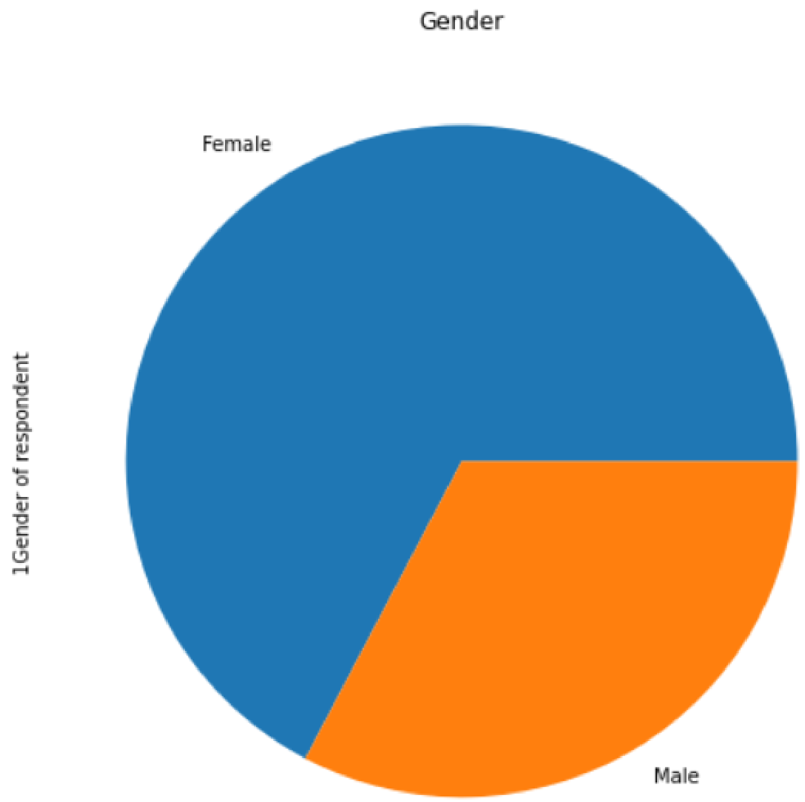
Assumptions:

- Before proceeding with EDA, we have to make some assumptions
- As the dataset available to us is only a fraction of the customers shopping online:
- We have to assume that this data set is not biased towards any field

That is, we have to assume that the ratios of say the gender are similar between our data and the actual data.

We might have, outliers, overfitting and most important of all smapling bias; which we will ignore for this study

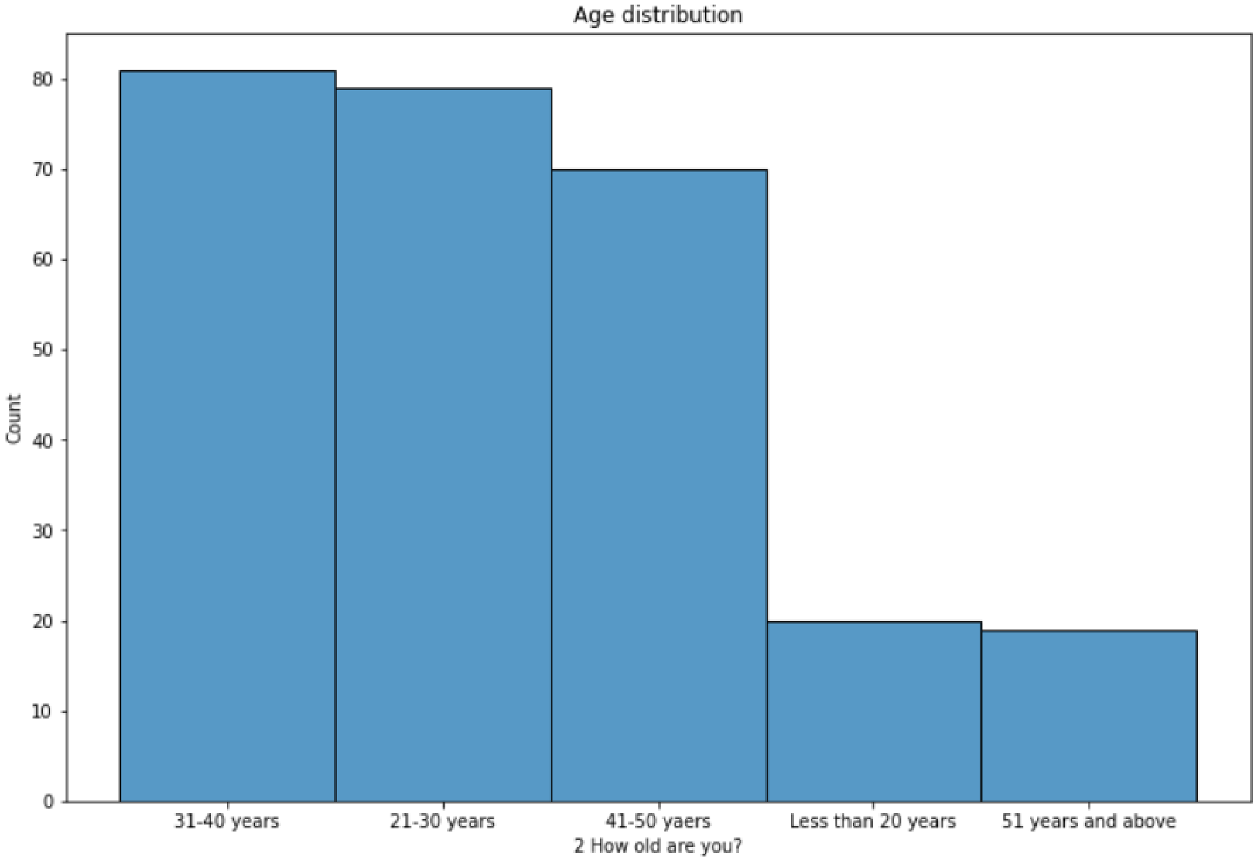
Gender



We can see that,
 the majority of online shoppers
 are women

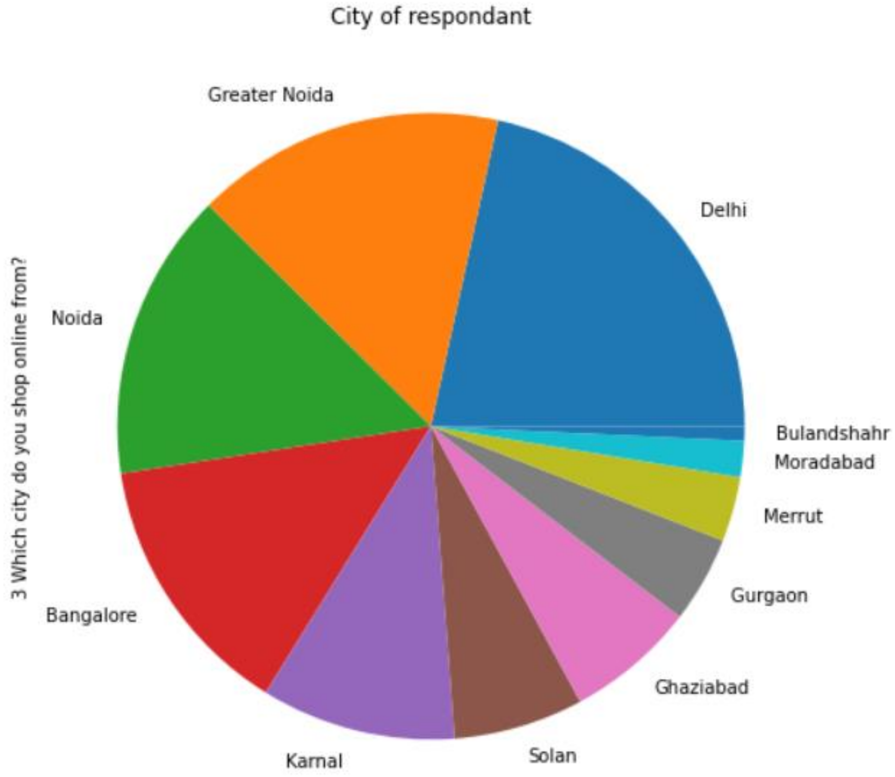
Age distribution

- We can see that that majority of respondents are between 21 and 50 years old



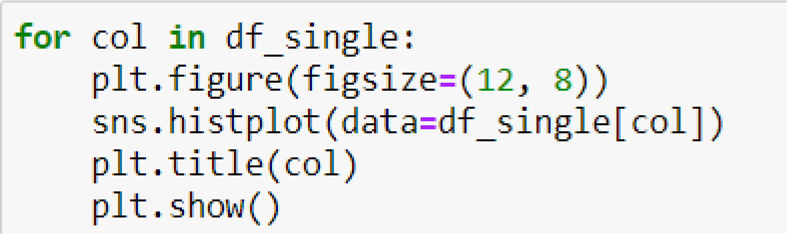
City of respondants

- Here we can see where the people are located



Similarly, using a script each and every column was converted into a histogram

- All the plots are available in the jupyter notebook.
- Script used to generate the plots:

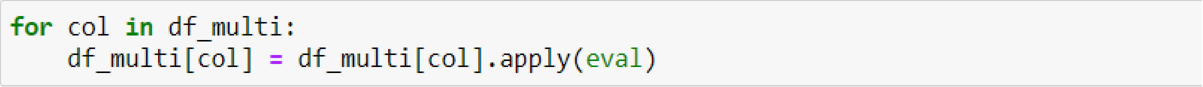


```
for col in df_single:
    plt.figure(figsize=(12, 8))
    sns.histplot(data=df_single[col])
    plt.title(col)
    plt.show()
```

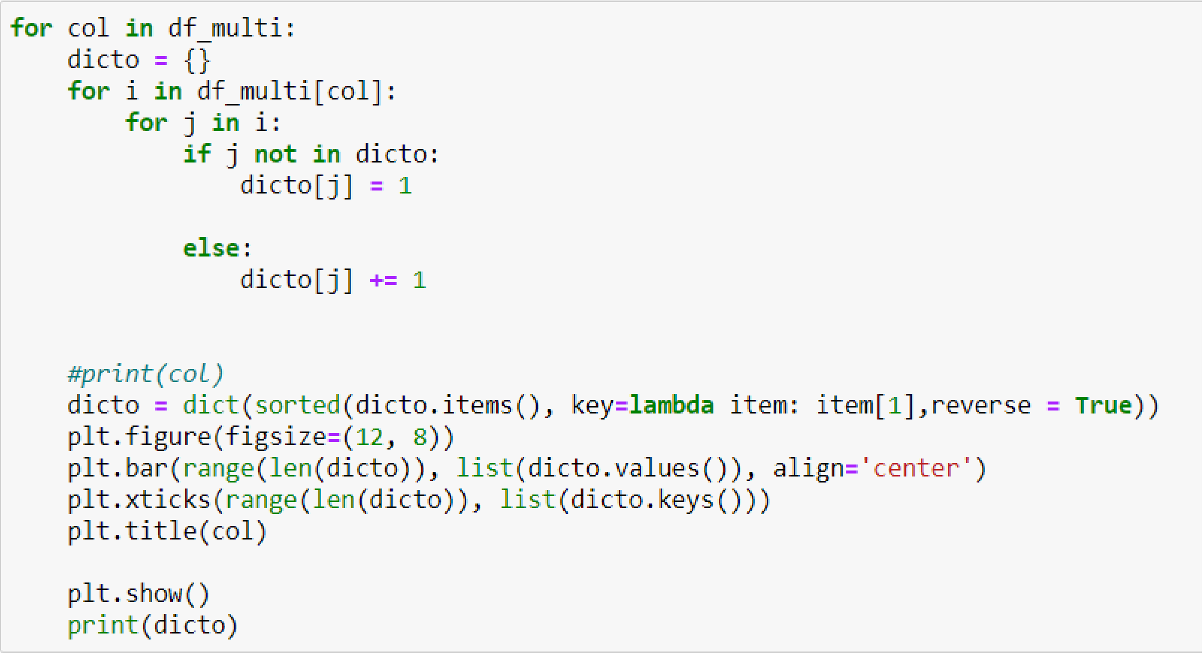
- This script iterates through all the columns and generates an individual plot for all of them.

Another script for multiple answer questions

- This script plots the values of each website in every question, along with a dictionary printed below each graph of the results.



```
for col in df_multi:
    df_multi[col] = df_multi[col].apply(eval)
```



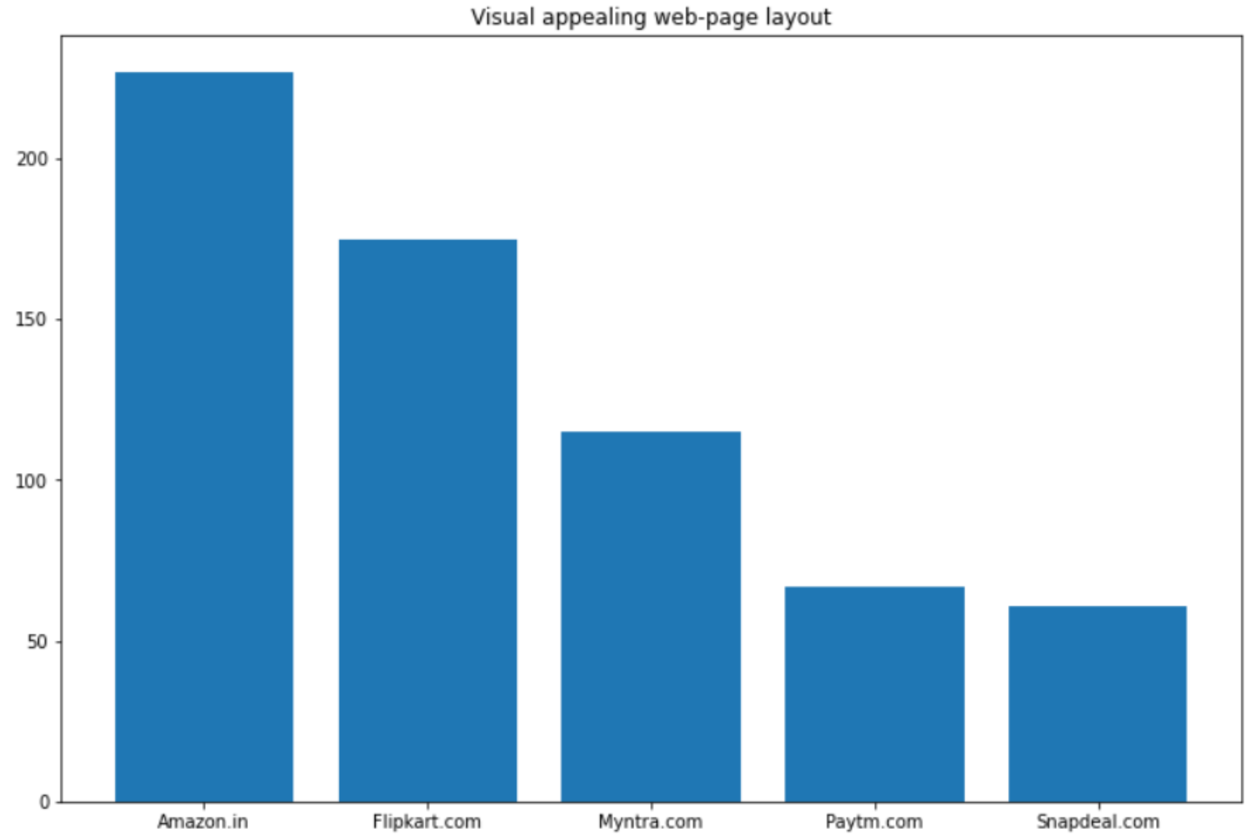
```
for col in df_multi:
    dicto = {}
    for i in df_multi[col]:
        for j in i:
            if j not in dicto:
                dicto[j] = 1

            else:
                dicto[j] += 1


    #print(col)
    dicto = dict(sorted(dicto.items(), key=lambda item: item[1],reverse = True))
    plt.figure(figsize=(12, 8))
    plt.bar(range(len(dicto)), list(dicto.values()), align='center')
    plt.xticks(range(len(dicto)), list(dicto.keys()))
    plt.title(col)

    plt.show()
    print(dicto)
```

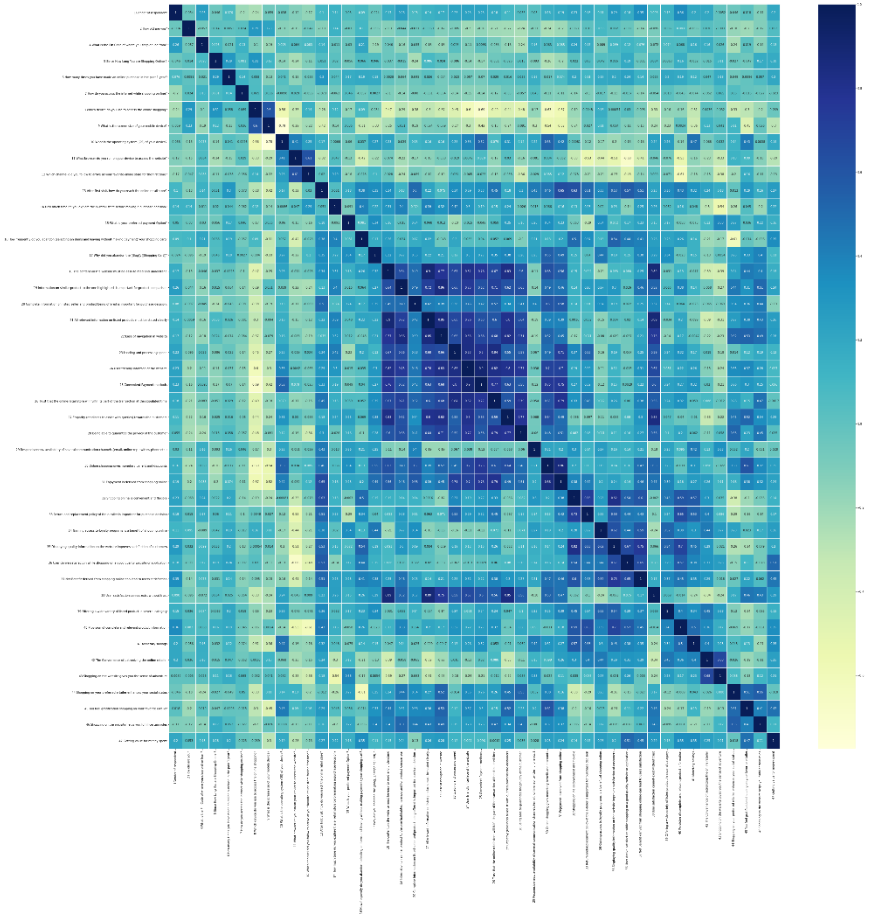
One of the responses of the above script

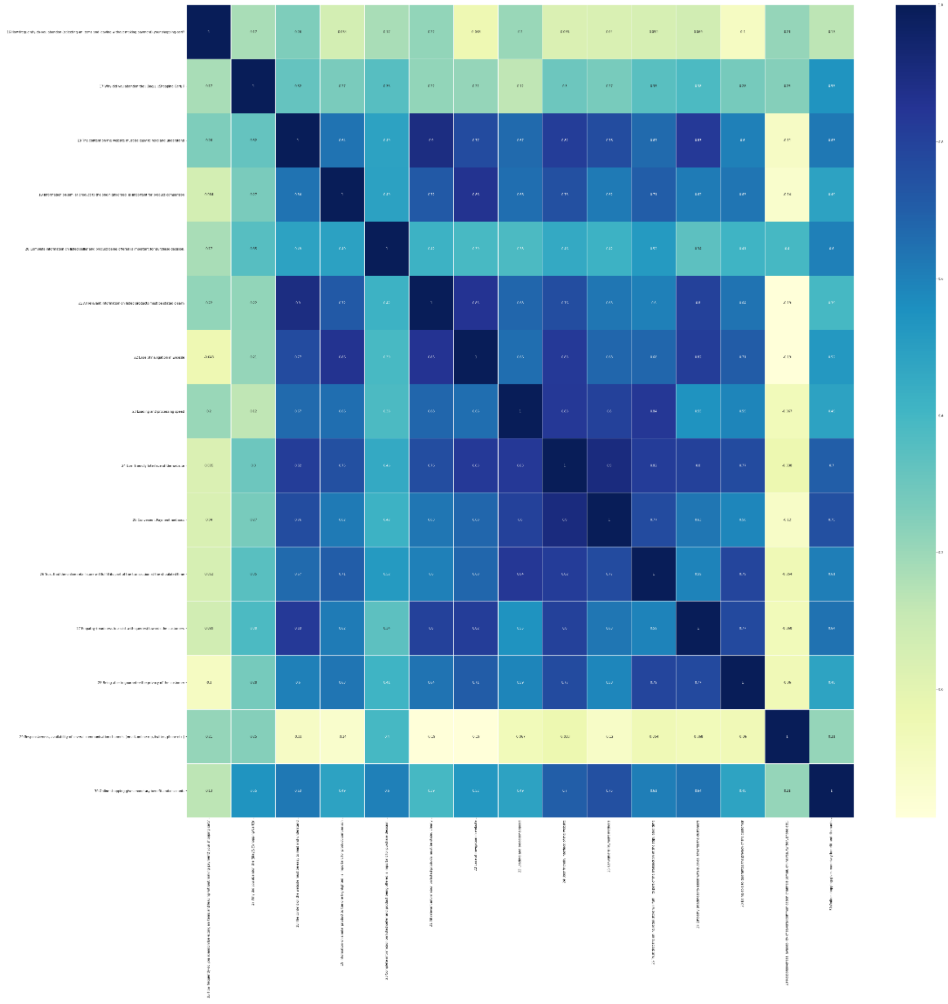


```
{'Amazon.in': 227, 'Flipkart.com': 175, 'Myntra.com': 115, 'Paytm.com': 67, 'Snapdeal.com': 61}
```

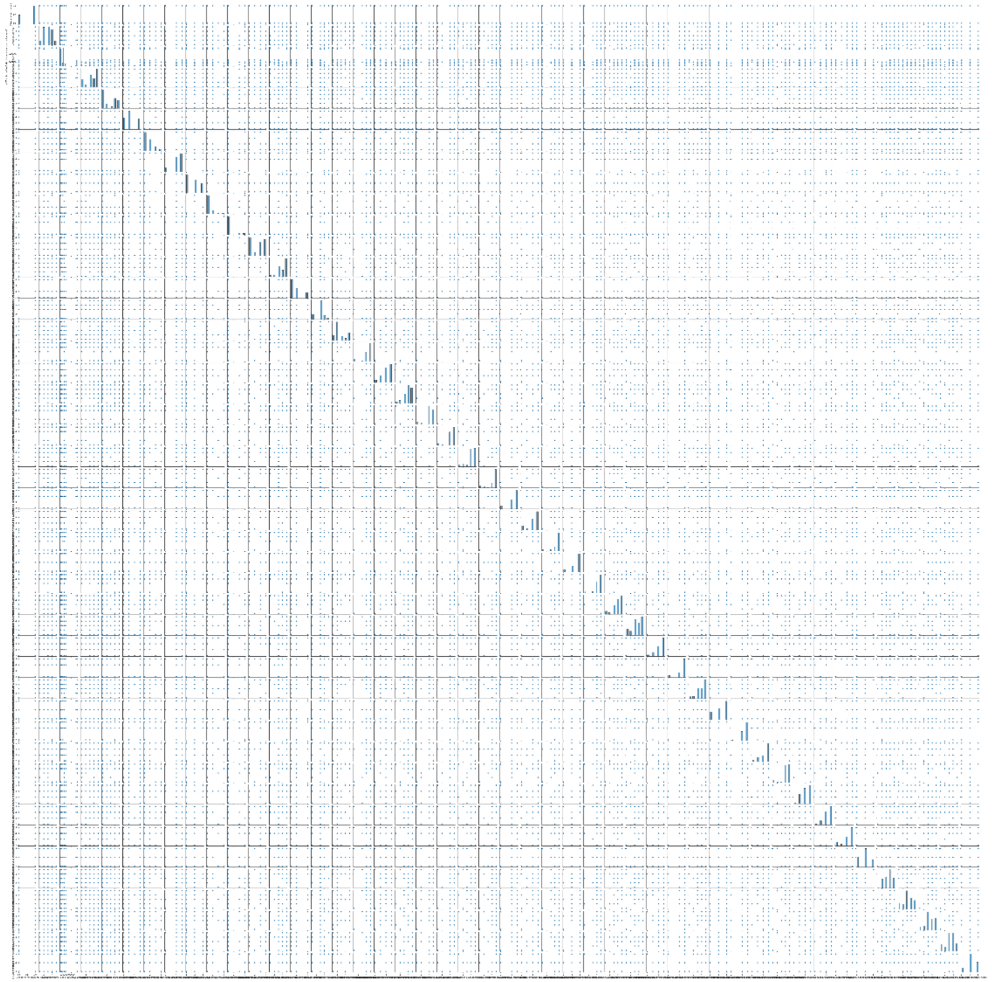
- A similar graph is plotted for each and every column

Co-relation matrix (from df_enc)





Pairplot of the columns



Conclusion

- Online retailors should value the values of it's customers, the best way to do that is providing a fast website which does not track users; providing:
- high service quality (good customer support)
- high system quality (fast website which loads quickly),
- a good information quality (by having the best product deals, alternative choices as recommended products)
- trust and net benefit (fast deliveries, best offers)