Data Mining Project

Clustering Clean Ads

Submitted by: Taniya Dubey

INDEX

List of figures

List of tables

Part 1: Clustering: Digital Ads Data:

The ads24x7 is a Digital Marketing company which has now got seed funding of $10 Million. They are expanding their wings in Marketing Analytics. They collected data from their Marketing Intelligence team and now wants you (their newly appointed data analyst) to segment type of ads based on the features provided. Use Clustering procedure to segment ads into homogeneous groups.

The following three features are commonly used in digital marketing:

CPM = (Total Campaign Spend / Number of Impressions) * 1,000

CPC = Total Cost (spend) / Number of Clicks

CTR = Total Measured Clicks / Total Measured Ad Impressions x 100

1.1 Clustering: Read the data and perform basic analysis such as printing a few rows (head and tail), info, data summary, null values duplicate values, etc.

- Dataset has 25857 rows and 19 columns.

Table 1: Top five rows of the dataset

	Timestamp	InventoryType	Ad - Length	Ad- Width	Ad Size	Ad Type	Platform	Device Type	Format	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
0	2020-9-2-17	Format1	300	250	75000	Inter222	Video	Desktop	Display	1806	325	323	1	0.0	0.35	0.0	0.0031	0.0	0.0
1	2020-9-2-18	Format1	300	250	75000	Inter223	Web	Mobile	Display	1979	384	380	0	0.0	0.35	0.0	0.0000	0.0	NaN
2	2020-9-3-16	Format6	336	250	84000	Inter217	Web	Desktop	Video	1566	298	297	0	0.0	0.35	0.0	0.0000	0.0	NaN
3	2020-9-3-2	Format1	300	250	75000	Inter224	Web	Desktop	Display	643	103	102	0	0.0	0.35	0.0	0.0000	0.0	NaN
4	2020-9-3-13	Format1	300	250	75000	Inter225	Video	Mobile	Display	1550	347	345	0	0.0	0.35	0.0	0.0000	0.0	NaN

Table 2: Top last rows of the dataset

	Timestamp	InventoryType	Ad - Length	Ad- Width	Ad Size	Ad Type	Platform	Device Type	Format	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
25852	2020-10-1-5	Format5	720	300	216000	Inter222	Video	Desktop	Video	1	1	1	0	0.01	0.35	0.0065	NaN	NaN	NaN
25853	2020-11-18-2	Format4	120	600	72000	inter230	Video	Mobile	Video	7	1	1	1	0.07	0.35	0.0455	NaN	NaN	NaN
25854	2020-9-14-0	Format5	720	300	216000	Inter221	App	Mobile	Video	2	2	2	1	0.09	0.35	0.0585	NaN	NaN	NaN
25855	2020-9-30-4	Format7	300	600	180000	Inter228	Video	Mobile	Display	1	1	1	0	0.01	0.35	0.0065	NaN	NaN	NaN
25856	2020-10-17-3	Format5	720	300	216000	Inter225	Video	Mobile	Display	1	1	1	0	0.01	0.35	0.0065	NaN	NaN	NaN

Table 3: Basic Information of the dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25857 entries, 0 to 25856
Data columns (total 19 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Timestamp              25857 non-null  object
 1   InventoryType          25857 non-null  object
 2   Ad - Length            25857 non-null  int64
 3   Ad- Width              25857 non-null  int64
 4   Ad Size                25857 non-null  int64
 5   Ad Type                25857 non-null  object
 6   Platform               25857 non-null  object
 7   Device Type            25857 non-null  object
 8   Format                 25857 non-null  object
 9   Available_Impressions  25857 non-null  int64
 10  Matched_Queries        25857 non-null  int64
 11  Impressions            25857 non-null  int64
 12  Clicks                 25857 non-null  int64
 13  Spend                  25857 non-null  float64
 14  Fee                    25857 non-null  float64
 15  Revenue                25857 non-null  float64
 16  CTR                    19392 non-null  float64
 17  CPM                    19392 non-null  float64
 18  CPC                    18330 non-null  float64
dtypes: float64(6), int64(7), object(6)
memory usage: 3.7+ MB
```

Table 4: Value Counts of the Categorical variables

```
Inventory type
 Format4     7218
Format1     5432
Format5     5151
Format3     3542
Format6     1968
Format2     1789
Format7      757
Name: InventoryType, dtype: int64

AD type
 Inter217     1849
Inter227     1848
Inter218     1848
Inter219     1848
Inter220     1848
Inter221     1848
Inter222     1847
Inter223     1847
Inter224     1847
inter230     1847
Inter225     1845
Inter226     1845
Inter228     1845
Inter229     1845
Name: Ad Type, dtype: int64
```

```
Platform
 Video     11077
Web        9236
App        5544
Name: Platform, dtype: int64

Device Type
 Mobile     16621
Desktop     9236
Name: Device Type, dtype: int64

Format
 Display    12929
Video      12928
Name: Format, dtype: int64
```

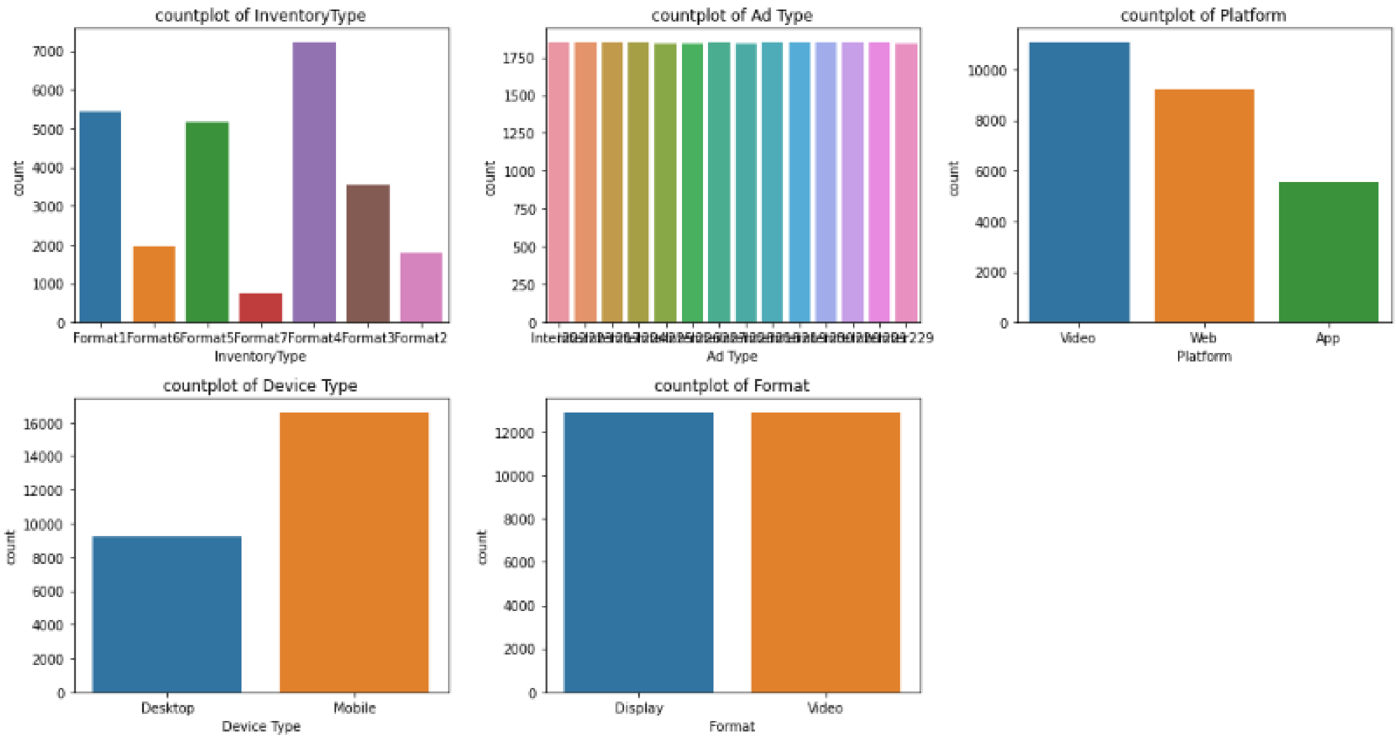


Fig1: count plot of variables

- The info table of dataset tells the following details:
  - Numerical variables: 13
  - Categorical variables: 6
  - Null values: CTR, CPM, CPC
  - Duplicate: 0
- From the count plot of device type we can see that most of the devices are mobile.
- There is no visible difference in count plot of format and ad type.

Table 5: statistical summary

	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
count	25857.000000	25857.000000	25857.000000	2.585700e+04	2.585700e+04	2.585700e+04	25857.000000	25857.000000	25857.000000	25857.000000	25638.000000	25638.000000	2.327100e+04
mean	390.431218	332.182774	99683.276482	2.169621e+06	1.155322e+06	1.107525e+06	9525.881388	2414.473115	0.336729	1716.548955	0.075883	7.588959	inf
std	230.696051	194.260924	62640.685612	4.542680e+06	2.407244e+06	2.326648e+06	16721.686071	3932.835240	0.030540	2993.025498	0.091413	8.938999	NaN
min	120.000000	70.000000	33600.000000	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	0.000000	0.210000	0.000000	0.000000	0.000000	0.000000e+00
25%	120.000000	250.000000	72000.000000	9.133000e+03	5.451000e+03	2.558000e+03	305.000000	36.030000	0.350000	23.420000	0.002400	1.589608	8.996397e-02
50%	300.000000	300.000000	75000.000000	3.309680e+05	1.894490e+05	1.621620e+05	3457.000000	1173.660000	0.350000	762.880000	0.007700	3.333333	1.408485e-01
75%	720.000000	600.000000	84000.000000	2.208484e+06	1.008171e+06	9.496930e+05	10681.000000	2692.280000	0.350000	1749.982000	0.130900	12.491089	5.584425e-01
max	728.000000	600.000000	216000.000000	2.759286e+07	1.470202e+07	1.419477e+07	143049.000000	26931.870000	0.350000	21276.180000	2.000000	715.000000	inf

1.2 Clustering: Treat missing values in CPC, CTR and CPM using the formula given.

- **Null values**

CTR 6465
CPM 6465
CPC 7527

- **Handling Null values**

Step 1: With the help of formulas given in the question we will make a user define function then make new columns so that most of the null values be replaced with right values in CTR, CPM & CPC.

Step 2: after 1 step we will drop 219 null values from CPC, CPM, CTR as it will impact further analysis

Step 3: after 2 step we will replace null values (Nan) of CPC with 0 and inf values with 0.

Step 4: check nulls in the data

Table 6: statistical summary after null value treatment

	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
count	25638.000000	25638.000000	25638.000000	2.563800e+04	2.563800e+04	2.563800e+04	25638.000000	25638.000000	25638.000000	25638.000000	25638.000000	25638.000000	25638.000000
mean	388.127155	332.155394	98773.479991	2.188154e+06	1.165191e+06	1.116986e+06	9607.251541	2435.097563	0.336616	1731.211730	0.075663	7.588959	0.302902
std	229.832116	194.881782	62055.513861	4.557595e+06	2.415124e+06	2.334302e+06	16769.662367	3943.234023	0.030645	3001.556311	0.091413	8.938999	0.339119
min	120.000000	70.000000	33600.000000	1.000000e+00	1.000000e+00	1.000000e+00	0.000000	0.000000	0.210000	0.000000	0.000000	0.000000	0.000000
25%	120.000000	250.000000	72000.000000	1.053350e+04	6.328500e+03	2.873250e+03	340.000000	40.012500	0.350000	26.007500	0.002400	1.589608	0.078988
50%	300.000000	300.000000	75000.000000	3.469430e+05	1.953880e+05	1.664925e+05	3536.000000	1194.930000	0.350000	776.710000	0.007700	3.333333	0.119688
75%	720.000000	600.000000	84000.000000	2.232121e+06	1.018126e+06	9.616298e+05	10856.250000	2718.142500	0.350000	1766.792875	0.130900	12.491089	0.503213
max	728.000000	600.000000	216000.000000	2.759286e+07	1.470202e+07	1.419477e+07	143049.000000	26931.870000	0.350000	21276.180000	2.000000	715.000000	7.264000

We can see from the statistical summary that now shows data at place of inf and nan values.

Table 7: few rows of cleaned data

	Timestamp	InventoryType	Ad - Length	Ad- Width	Ad Size	Ad Type	Platform	Device Type	Format	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
0	2020-9-2-17	Format1	300	250	75000	Inter222	Video	Desktop	Display	1806	325	323	1	0.0	0.35	0.0	0.0031	0.0	0.0
1	2020-9-2-18	Format1	300	250	75000	Inter223	Web	Mobile	Display	1979	384	380	0	0.0	0.35	0.0	0.0000	0.0	0.0
2	2020-9-3-16	Format6	336	250	84000	Inter217	Web	Desktop	Video	1566	298	297	0	0.0	0.35	0.0	0.0000	0.0	0.0
3	2020-9-3-2	Format1	300	250	75000	Inter224	Web	Desktop	Display	643	103	102	0	0.0	0.35	0.0	0.0000	0.0	0.0
4	2020-9-3-13	Format1	300	250	75000	Inter225	Video	Mobile	Display	1550	347	345	0	0.0	0.35	0.0	0.0000	0.0	0.0

1.3 - Clustering: Check if there are any outliers. Do you think treating outliers is necessary for K-Means clustering? Based on your judgement decide whether to treat outliers and if yes, which method to employ.

As we know that there are 6 categorical variables which we will drop because for further analysis we will only need numerical variables. The name of new data frame will be df3 which will contain the following columns:

Table 8: datatype of df3 data frame

Ad - Length	int64
Ad- Width	int64
Ad Size	int64
Available_Impressions	int64
Matched_Queries	int64
Impressions	int64
Clicks	int64
Spend	float64
Revenue	float64
CTR	float64
CPM	float64
CPC	float64

Now let's see boxplot for outliers

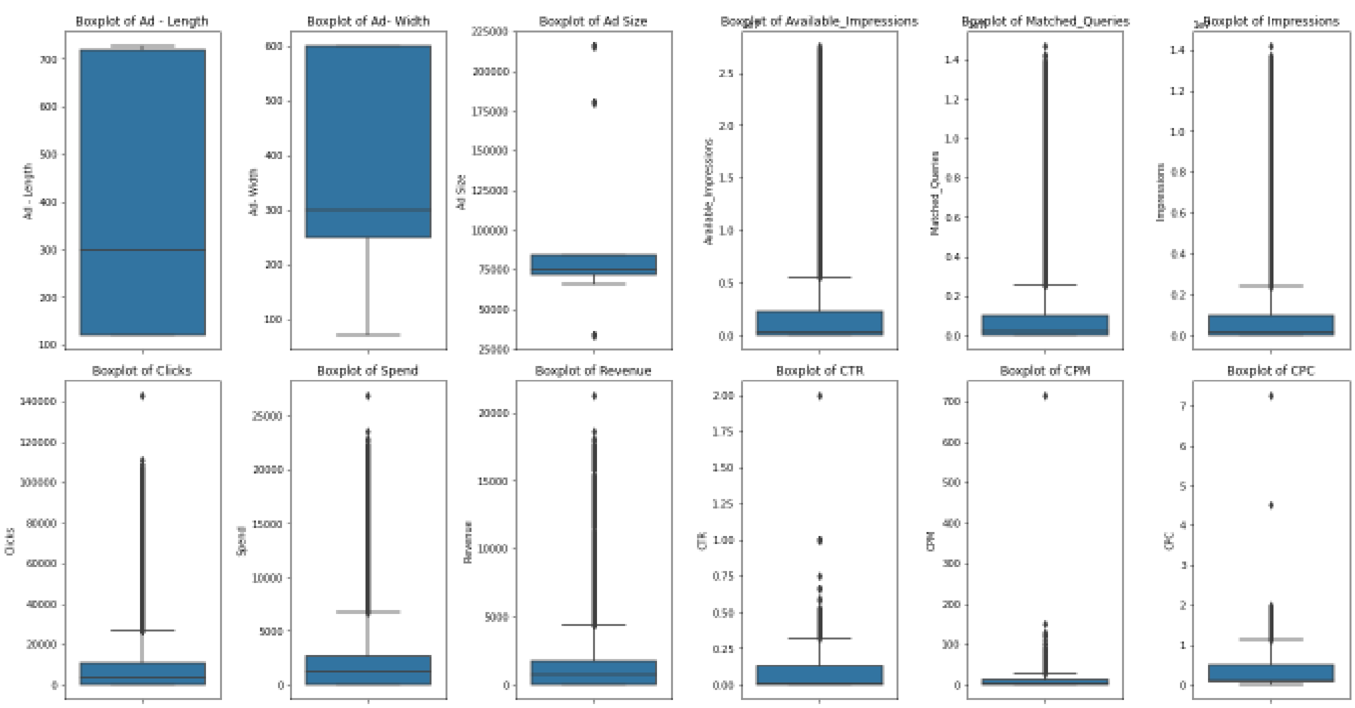


Fig 2: box plot with outliers' part 1

As we can see all the variables except first 2 have outliers

And we know that clustering is very sensitive to outliers and if we don't treat them then it can give rise to errors and then we will not get the most accurate analysis of the data present so we have treated the outliers.

Outliers were treated by using Winsorization, i.e., bringing the larger outliers (Data points above the Q3 + 1.5 * IQR value) to the upper whisker value and bringing the smaller outliers (Data points below the Q1 - 1.5 * IQR value) to the lower whisker.

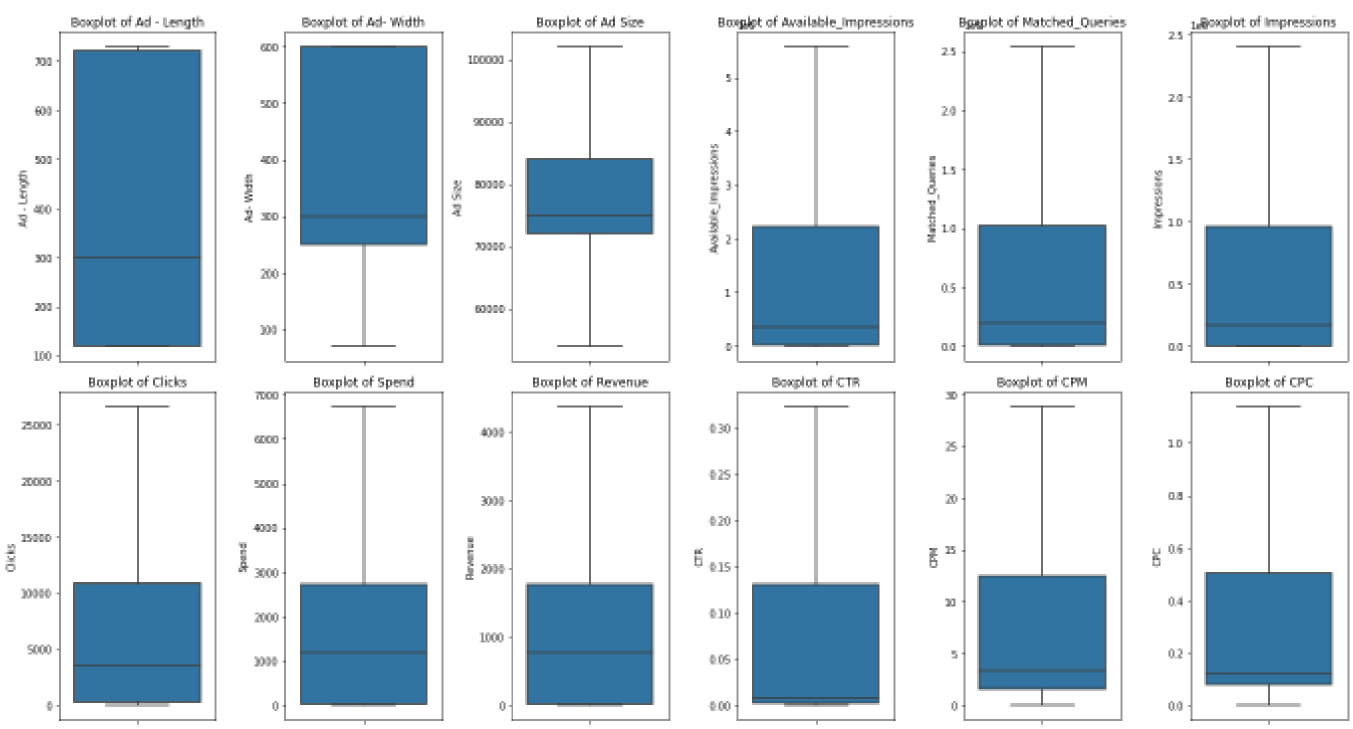


Fig 3: box plot without outliers' part 1

1.4 - Clustering: Perform z-score scaling and discuss how it affects the speed of the algorithm.

We will scale the data with the help of Sklenar preprocessing (standard scaler).

Z-Score scaling consists in transforming each feature subtracting the mean value and dividing by the standard deviation

Z-Score scaling adjusts the numbers to make it easy to compare the values that are out of each other's scope. This **helps increase the accuracy of the models & analysis of the data.** Z-score also increase the speed of further analysis.

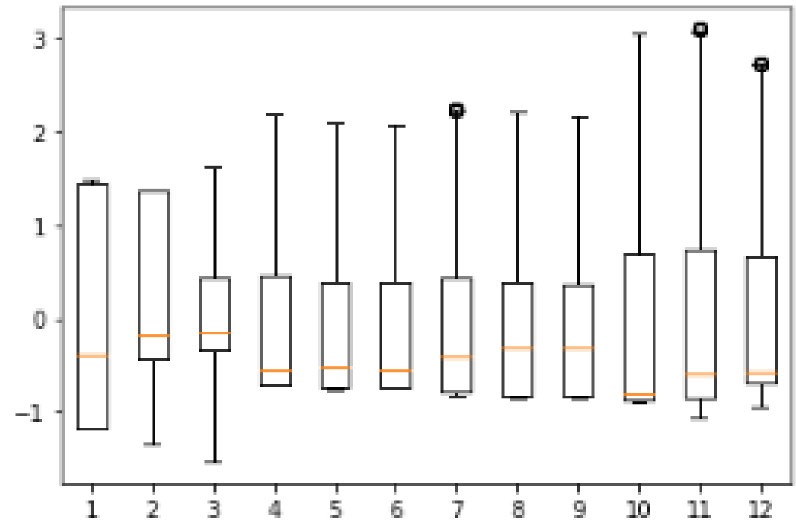


Fig 4: box plot of scaled data

1.5 - Clustering: Perform Hierarchical by constructing a Dendrogram using WARD and Euclidean distance.

We will perform Hierarchical by constructing a Dendrogram using ward with the help of scipy.cluster.hierarchy

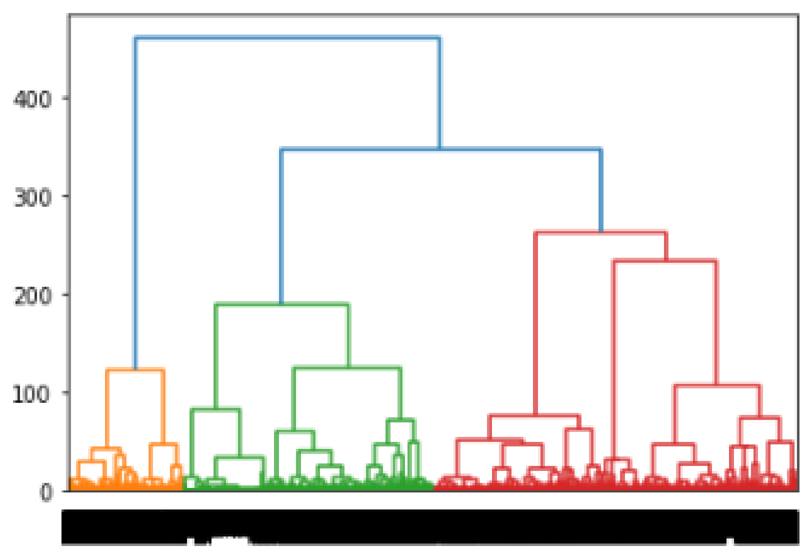


Fig 5: dendrogram

1.6 - Clustering: Make Elbow plot (up to n=10) and identify optimum number of clusters for k-means algorithm.

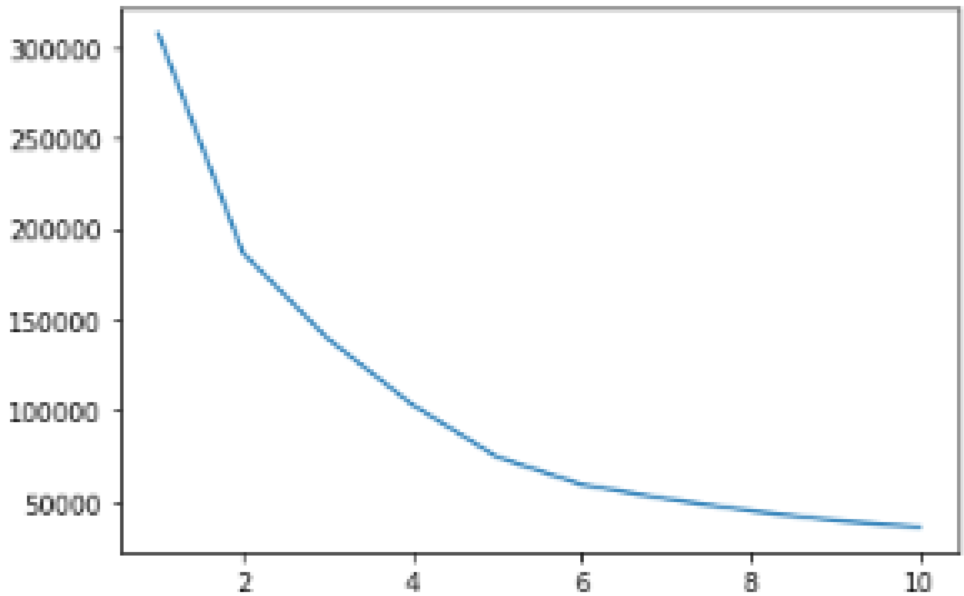


Fig 6: scree plot part 1

As we can see that after 4 the difference is less and less so we will take the 4 of the clusters for k means.

	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Revenue	CTR	CPM	CPC	kmeans
0	300.0	250.0	75000.0	1806.0	325.0	323.0	1.0	0.0	0.0	0.0031	0.0	0.0	2
1	300.0	250.0	75000.0	1979.0	384.0	380.0	0.0	0.0	0.0	0.0000	0.0	0.0	2
2	336.0	250.0	84000.0	1566.0	298.0	297.0	0.0	0.0	0.0	0.0000	0.0	0.0	2
3	300.0	250.0	75000.0	643.0	103.0	102.0	0.0	0.0	0.0	0.0000	0.0	0.0	2
4	300.0	250.0	75000.0	1550.0	347.0	345.0	0.0	0.0	0.0	0.0000	0.0	0.0	2

Table 9: of data with k means

1.7 - Clustering: Print silhouette scores for up to 10 clusters and identify optimum number of clusters.

```
For n_clusters = 2 The average silhouette_score is : 0.3790731628519866
For n_clusters = 3 The average silhouette_score is : 0.3497350804755862
For n_clusters = 4 The average silhouette_score is : 0.425875327632047
For n_clusters = 5 The average silhouette_score is : 0.4839513671813748
For n_clusters = 6 The average silhouette_score is : 0.4879521917720897
For n_clusters = 7 The average silhouette_score is : 0.4908768188248661
For n_clusters = 8 The average silhouette_score is : 0.5127144457099674
For n_clusters = 9 The average silhouette_score is : 0.5064391516899717
For n_clusters = 10 The average silhouette_score is : 0.47033918891999843
```

Table 10: table of silhouette score

4 is the optimum number of clusters which we require.

	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Revenue	CTR	CPM	CPC	kmeans	sil_width
0	300.0	250.0	75000.0	1806.0	325.0	323.0	1.0	0.0	0.0	0.0031	0.0	0.0	0	0.889424
1	300.0	250.0	75000.0	1979.0	384.0	380.0	0.0	0.0	0.0	0.0000	0.0	0.0	0	0.897774

Table 11: data with silhouette width

Minimum silhouette sample is –0.2566

1.8 - Clustering: Profile the ads based on optimum number of clusters using silhouette score and your domain understanding

Table 12: cluster and kmeans table

Clus_kmeans8	Device Type						
0	Desktop	2739.499379	1878.141205	1225.868135	0.002800	1.888647	0.814333
	Mobile	2696.674403	1881.736233	1228.186592	0.002742	1.890719	0.817669
1	Desktop	1621.475632	166.433312	108.181600	0.163459	14.629112	0.101136
	Mobile	1593.275967	169.120551	109.928303	0.163285	14.947250	0.101228
2	Desktop	8490.731092	5242.815011	3654.079817	0.002404	1.469917	0.620312
	Mobile	8347.800242	5182.984304	3603.685739	0.002403	1.478311	0.624004
3	Desktop	14560.822610	1251.564690	814.951886	0.138678	12.147766	0.089129
	Mobile	14457.597914	1261.891205	821.717384	0.139819	12.224552	0.090142
4	Desktop	62108.580699	6658.438802	4764.444965	0.138379	15.597476	0.113041
	Mobile	62297.969582	6706.956778	4803.094412	0.137512	15.547503	0.113266
5	Desktop	95.441361	21.279311	13.831562	0.007076	0.996617	0.028351
	Mobile	144.570525	28.578127	18.575772	0.008241	1.080855	0.029425
6	Desktop	3848.759684	1348.308055	876.502439	0.004877	1.851673	0.403109
	Mobile	3840.537549	1332.892653	866.537555	0.004853	1.832937	0.398911
7	Desktop	15815.355839	14167.534069	10794.622053	0.001796	1.711820	0.975508
	Mobile	15582.460381	14056.087944	10706.635538	0.001810	1.740815	0.985900

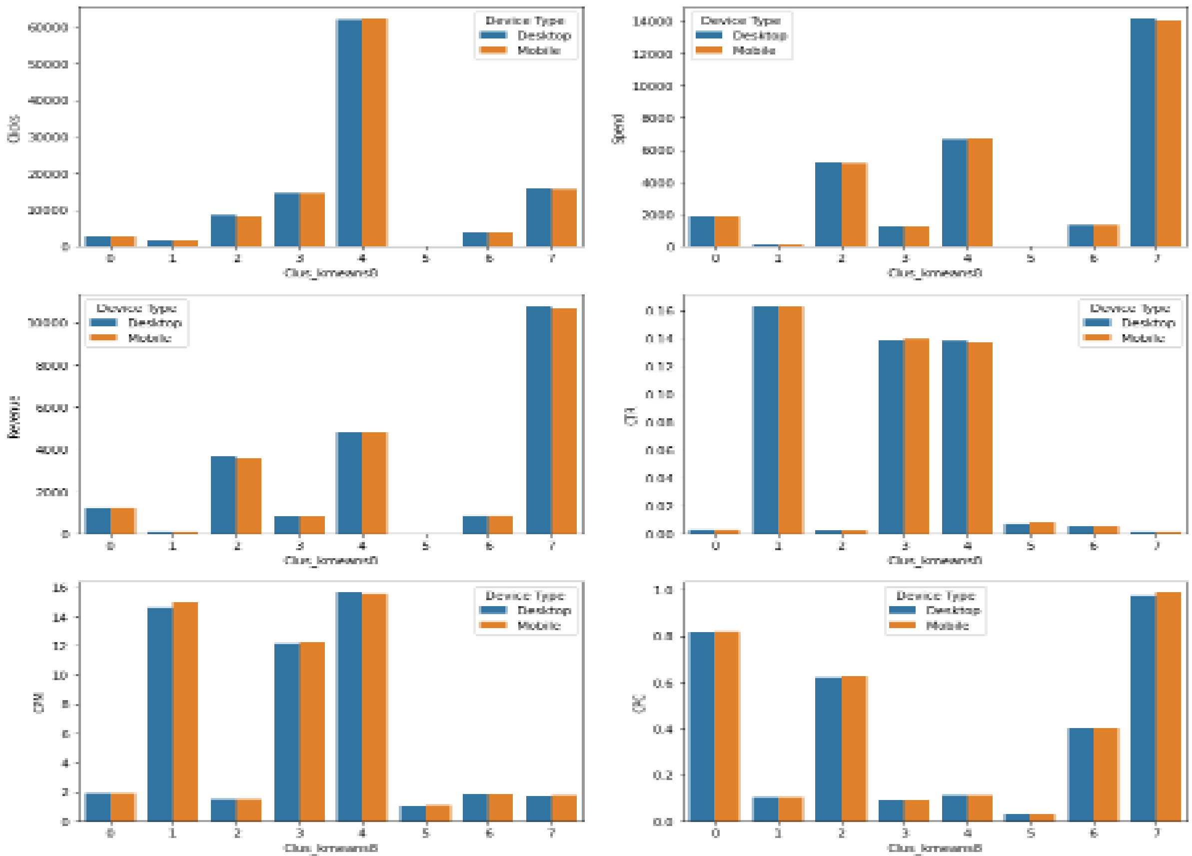


Fig 25: plot of clusters.

1.9 - Clustering: Conclude the project by providing summary of your learnings.

Some recommendation which can be followed by the ads24x7 company are as follows

We can make some conclusion based on above figures and tables.

- **Cluster 0:**

  This type of cluster has highest average **CTR (**click through rate) among other clusters. It shows that a greater number of users sees and click these types of ads. The average **CPM** (cost per impression) is more as compare to others which means that the relevance and quality of these are more. The average **CPC (**cost per clicks) is not so high which shows that the user of this type does not purchase the product that much. this type of ads can provide some offers to increase CPC.

Like this every cluster is providing meaning full information which will help the business to grow and expand.

After the whole process we can conclude that the ads having low CTR and CPM value have to improve their quality and relevance of content by spend some amount of money while the ads having low CPC value have to provide some great offers so more number of customers are buying the product. This also results in larger revenue.

Part 2 : PCA

PCA FH (FT): Primary census abstract for female headed households excluding institutional households (India & States/UTs - District Level), Scheduled tribes - 2011 PCA for Female Headed Household Excluding Institutional Household. The Indian Census has the reputation of being one of the best in the world. The first Census in India was conducted in the year 1872. This was conducted at different points of time in different parts of the country. In 1881 a Census was taken for the entire country simultaneously. Since then, Census has been conducted every ten years, without a break. Thus, the Census of India 2011 was the fifteenth in this unbroken series since 1872, the seventh after independence and the second census of the third millennium and twenty first century. The census has been uninterruptedly continued despite of several adversities like wars, epidemics, natural calamities, political unrest, etc. The Census of India is conducted under the provisions of the Census Act 1948 and the Census Rules, 1990. The Primary Census Abstract which is important publication of 2011 Census gives basic information on Area, Total Number of Households, Total Population, Scheduled Castes, Scheduled Tribes Population, Population in the age group 0-6, Literates, Main Workers and Marginal Workers classified by the four broad industrial categories, namely, (I) Cultivators, (ii) Agricultural Laborers, (iii) Household Industry Workers, and (iv) Other Workers and also non-Workers. The characteristics of the Total Population include Scheduled Castes, Scheduled Tribes, Institutional and Houseless Population and are presented by sex and rural-urban residence. Census 2011 covered 35 States/Union Territories, 640 districts, 5,924 sub-districts, 7,935 Towns and 6,40,867 Villages. The data collected has so many variables thus making it difficult to find useful details without using Data Science Techniques. You are tasked to perform detailed EDA and identify Optimum Principal Components that explains the most variance in data.

2.1 - PCA: Read the data and perform basic checks like checking head, info, summary, nulls, and duplicates, etc.

Table 13: few top rows of data set

	State Code	Dist.Code	State	Area Name	No_HH	TOT_M	TOT_F	M_06	F_06	M_SC	...	MARG_CL_0_3_M	MARG_CL_0_3_F	MARG_AL_0_3_M	MARG_AL_0_3_F	MARG_HH_0_3_M	MARG_HH_0_3_F	MARG_OT_0_3_M	MARG_OT_0_3_F	NON_WORK_M	NON_WORK_F
0	1	1	Jammu & Kashmir	Kupwara	7707	23388	29796	5862	6196	3	...	1150	749	180	237	680	252	32	46	258	214
1	1	2	Jammu & Kashmir	Badgam	6218	19585	23102	4482	3733	7	...	525	715	123	229	186	148	76	178	140	160
2	1	3	Jammu & Kashmir	Leh(Ladakh)	4452	6546	10964	1082	1018	3	...	114	188	44	89	3	34	0	4	67	61
3	1	4	Jammu & Kashmir	Kargil	1320	2784	4206	563	677	0	...	194	247	61	128	13	50	4	10	116	59
4	1	5	Jammu & Kashmir	Punch	11654	20591	29981	5157	4587	20	...	874	1928	465	1043	205	302	24	105	180	478

Table 14: few last rows of data set

	State Code	Dist.Code	State	Area Name	No_HH	TOT_M	TOT_F	M_06	F_06	M_SC	...	MARG_CL_0_3_M	MARG_CL_0_3_F	MARG_AL_0_3_M	MARG_AL_0_3_F	MARG_HH_0_3_M	MARG_HH_0_3_F	MARG_OT_0_3_M	MARG_OT_0_3_F	NON_WORK_M	NON_WORK_F
635	34	636	Puducherry	Mahe	3333	8154	11781	1146	1203	21	...	32	47	0	0	0	0	0	0	32	47
636	34	637	Puducherry	Karaikal	10612	12346	21691	1544	1533	2234	...	155	337	3	14	38	130	4	23	110	170
637	35	638	Andaman & Nicobar Island	Nicobars	1275	1549	2630	227	225	0	...	104	134	9	4	2	6	17	47	76	77
638	35	639	Andaman & Nicobar Island	North & Middle Andaman	3762	5200	8012	723	664	0	...	136	172	24	44	11	21	1	4	100	103
639	35	640	Andaman & Nicobar Island	South Andaman	7975	11977	18049	1470	1358	0	...	173	122	6	2	17	17	2	4	148	99

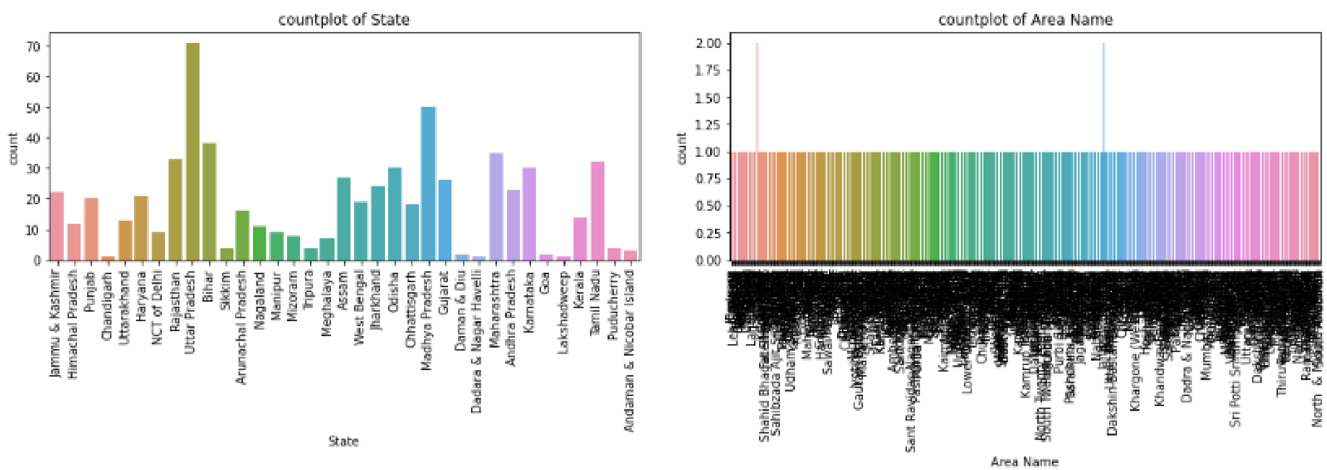


fig 7: count plot of state and area

Table 15: info of data set

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 640 entries, 0 to 639
Data columns (total 61 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   State Code        640 non-null    int64 
 1   Dist.Code         640 non-null    int64 
 2   State             640 non-null    object
 3   Area Name         640 non-null    object
 4   No_HH             640 non-null    int64 
 5   TOT_M             640 non-null    int64 
 6   TOT_F             640 non-null    int64 
 7   M_06              640 non-null    int64 
 8   F_06              640 non-null    int64 
 9   M_SC              640 non-null    int64 
 10  F_SC              640 non-null    int64 
 11  M_ST              640 non-null    int64 
 12  F_ST              640 non-null    int64 
 13  M_LIT             640 non-null    int64 
 14  F_LIT             640 non-null    int64 
 15  M_ILL             640 non-null    int64 
 16  F_ILL             640 non-null    int64 
 17  TOT_WORK_M        640 non-null    int64 
 18  TOT_WORK_F        640 non-null    int64 
 19  MAINWORK_M        640 non-null    int64 
 20  MAINWORK_F        640 non-null    int64 
 21  MAIN_CL_M         640 non-null    int64 
 22  MAIN_CL_F         640 non-null    int64 
 23  MAIN_AL_M         640 non-null    int64 
 24  MAIN_AL_F         640 non-null    int64 
 25  MAIN_HH_M         640 non-null    int64 
 26  MAIN_HH_F         640 non-null    int64 
 27  MAIN_OT_M         640 non-null    int64 
 28  MAIN_OT_F         640 non-null    int64 
 29  MARGWORK_M        640 non-null    int64 
 30  MARGWORK_F        640 non-null    int64 
 31  MARG_CL_M         640 non-null    int64 
 32  MARG_CL_F         640 non-null    int64 
 33  MARG_AL_M         640 non-null    int64 
 34  MARG_AL_F         640 non-null    int64 
 35  MARG_HH_M         640 non-null    int64 
 36  MARG_HH_F         640 non-null    int64 
 37  MARG_OT_M         640 non-null    int64 
 38  MARG_OT_F         640 non-null    int64 
 39  MARGWORK_3_6_M    640 non-null    int64 
 40  MARGWORK_3_6_F    640 non-null    int64 
 41  MARG_CL_3_6_M     640 non-null    int64 
 42  MARG_CL_3_6_F     640 non-null    int64 
 43  MARG_AL_3_6_M     640 non-null    int64 
 44  MARG_AL_3_6_F     640 non-null    int64 
 45  MARG_HH_3_6_M     640 non-null    int64 
 46  MARG_HH_3_6_F     640 non-null    int64 
 47  MARG_OT_3_6_M     640 non-null    int64 
 48  MARG_OT_3_6_F     640 non-null    int64 
 49  MARGWORK_0_3_M    640 non-null    int64 
 50  MARGWORK_0_3_F    640 non-null    int64 
 51  MARG_CL_0_3_M     640 non-null    int64 
 52  MARG_CL_0_3_F     640 non-null    int64 
 53  MARG_AL_0_3_M     640 non-null    int64 
 54  MARG_AL_0_3_F     640 non-null    int64 
 55  MARG_HH_0_3_M     640 non-null    int64 
 56  MARG_HH_0_3_F     640 non-null    int64 
 57  MARG_OT_0_3_M     640 non-null    int64 
 58  MARG_OT_0_3_F     640 non-null    int64 
 59  NON_WORK_M        640 non-null    int64 
 60  NON_WORK_F        640 non-null    int64 
dtypes: int64(59), object(2)
memory usage: 305.1+ KB
```

Table 16: statical summary of data set

	count	mean	std	min	25%	50%	75%	max
State Code	640.0	17.114062	9.426486	1.0	9.00	18.0	24.00	35.0
Dist.Code	640.0	320.500000	184.896367	1.0	160.75	320.5	480.25	640.0
No HH	640.0	51222.871875	48135.405475	350.0	19484.00	35837.0	68892.00	310450.0
TOT M	640.0	79940.576563	73384.511114	391.0	30228.00	58339.0	107918.50	485417.0
TOT F	640.0	122372.084375	113600.717282	698.0	46517.75	87724.5	164251.75	750392.0
M 06	640.0	12309.098438	11500.906881	56.0	4733.75	9159.0	16520.25	96223.0
F 06	640.0	11942.300000	11326.294567	56.0	4672.25	8663.0	15902.25	95129.0
M SC	640.0	13820.946875	14426.373130	0.0	3466.25	9591.5	19429.75	103307.0
F SC	640.0	20778.392188	21727.887713	0.0	5603.25	13709.0	29180.00	156429.0
M ST	640.0	6191.807813	9912.668948	0.0	293.75	2333.5	7658.00	96785.0
F ST	640.0	10155.640625	15875.701488	0.0	429.50	3834.5	12480.25	130119.0
M LIT	640.0	57967.979688	55910.282466	286.0	21298.00	42693.5	77989.50	403261.0
F LIT	640.0	66359.565625	75037.860207	371.0	20932.00	43796.5	84799.75	571140.0
M ILL	640.0	21972.596875	19825.605268	105.0	8590.00	15767.5	29512.50	105961.0
F ILL	640.0	56012.518750	47116.693769	327.0	22367.00	42386.0	78471.00	254160.0
TOT WORK M	640.0	37992.407813	36419.537491	100.0	13753.50	27936.5	50226.75	269422.0
TOT WORK F	640.0	41295.760938	37192.360943	357.0	16097.75	30588.5	53234.25	257848.0
MAINWORK M	640.0	30204.446875	31480.915680	65.0	9787.00	21250.5	40119.00	247911.0
MAINWORK F	640.0	28198.846875	29998.262689	240.0	9502.25	18484.0	35063.25	226166.0
MAIN CL M	640.0	5424.342188	4739.161969	0.0	2023.50	4160.5	7695.00	29113.0
MAIN CL F	640.0	5486.042188	5326.362728	0.0	1920.25	3908.5	7286.25	36193.0
MAIN AL M	640.0	5849.109375	6399.507966	0.0	1070.25	3936.5	8067.25	40843.0
MAIN AL F	640.0	8925.995312	12864.287584	0.0	1408.75	3933.5	10617.50	87945.0
MAIN HH M	640.0	883.893750	1278.842345	0.0	187.50	498.5	1099.25	16429.0
MAIN HH F	640.0	1380.773438	3179.414449	0.0	248.75	540.5	1435.75	45979.0
MAIN OT M	640.0	18047.101562	26068.480886	36.0	3997.50	9598.0	21249.50	240855.0
MAIN OT F	640.0	12406.035938	18972.202369	153.0	3142.50	6380.5	14368.25	209355.0
MARGWORK M	640.0	7787.960938	7410.791691	35.0	2937.50	5627.0	9800.25	47553.0
MARGWORK F	640.0	13096.914062	10996.474528	117.0	5424.50	10175.0	18879.25	66915.0
MARG CL M	640.0	1040.737500	1311.546847	0.0	311.75	606.5	1281.00	13201.0
MARG CL F	640.0	2307.682813	3564.626095	0.0	630.25	1226.0	2659.25	44324.0
MARG AL M	640.0	3304.326562	3781.555707	0.0	873.50	2062.0	4300.75	23719.0
MARG AL F	640.0	6463.281250	6773.876298	0.0	1402.50	4020.5	9089.25	45301.0
MARG HH M	640.0	316.742188	462.661891	0.0	71.75	166.0	356.50	4298.0
MARG HH F	640.0	786.626562	1198.718213	0.0	171.75	429.0	962.50	15448.0
MARG OT M	640.0	3126.154687	3609.391821	7.0	935.50	2036.0	3985.25	24728.0
MARG OT F	640.0	3539.323438	4115.191314	19.0	1071.75	2349.5	4400.50	36377.0
MARGWORK 3 6 M	640.0	41948.168750	39045.316918	291.0	16208.25	30315.0	57218.75	300937.0
MARGWORK 3 6 F	640.0	81076.323438	82970.406216	341.0	26619.50	56793.0	107924.00	676450.0
MARG CL 3 6 M	640.0	6394.987500	6019.806844	27.0	2372.00	4630.0	6167.00	39106.0

MARG CL 3 6 F	640.0	10339.884083	8467.473429	85.0	4351.50	8295.0	15102.00	50065.0
MARG AL 3 6 M	640.0	789.848438	905.639279	0.0	235.50	480.5	986.00	7426.0
MARG AL 3 6 F	640.0	1749.584375	2496.541514	0.0	497.25	985.5	2059.00	27171.0
MARG HH 3 6 M	640.0	2743.635938	3059.586387	0.0	718.75	1714.5	3702.25	19343.0
MARG HH 3 6 F	640.0	5169.850000	5335.640960	0.0	1113.75	3294.0	7502.25	36253.0
MARG OT 3 6 M	640.0	245.362500	358.728567	0.0	58.00	129.5	276.00	3535.0
MARG OT 3 6 F	640.0	585.884375	900.025817	0.0	127.75	320.5	719.25	12094.0
MARGWORK 0 3 M	640.0	2616.140625	3036.964381	7.0	755.00	1681.5	3320.25	20648.0
MARGWORK 0 3 F	640.0	2834.545312	3327.836932	14.0	833.50	1834.5	3810.50	25844.0
MARG CL 0 3 M	640.0	1392.973438	1489.707052	4.0	489.50	949.0	1714.00	9875.0
MARG CL 0 3 F	640.0	2757.050000	2788.776676	30.0	957.25	1928.0	3599.75	21611.0
MARG AL 0 3 M	640.0	250.889062	453.336594	0.0	47.00	114.5	270.75	5775.0
MARG AL 0 3 F	640.0	558.098438	1117.642748	0.0	109.00	247.5	568.75	17153.0
MARG HH 0 3 M	640.0	560.690625	762.578991	0.0	136.50	308.0	642.00	6116.0
MARG HH 0 3 F	640.0	1293.431250	1585.377936	0.0	298.00	717.0	1710.75	13714.0
MARG OT 0 3 M	640.0	71.379688	107.897627	0.0	14.00	35.0	79.00	895.0
MARG OT 0 3 F	640.0	200.742188	309.740854	0.0	43.00	113.0	240.00	3354.0
NON WORK M	640.0	510.014063	610.603187	0.0	161.00	326.0	604.50	6456.0
NON WORK F	640.0	704.778125	910.209225	5.0	220.50	464.5	853.50	10533.0

- Dataset has 640 rows and 61 columns.
- There are no duplicates in the data.
- There are no null values in the data.
- There are two objective type data and others are integers.

2.2 - PCA: Perform detailed Exploratory analysis by creating certain questions

The 5 variables which I have choose are following:

1)F_ILL

2)M_ILL

3)F_LIT

4)M_LIT

5)TOT_WORK_F

Table 17: statical summary of data set

	count	mean	std	min	25%	50%	75%	max
F_ILL	640.0	56012.518750	47116.693769	327.0	22367.00	42386.0	78471.00	254160.0
M_ILL	640.0	21972.596875	19825.605268	105.0	8590.00	15767.5	29512.50	105961.0
F_LIT	640.0	66359.565625	75037.860207	371.0	20932.00	43796.5	84799.75	571140.0
M_LIT	640.0	57967.979688	55910.282466	286.0	21298.00	42693.5	77989.50	403261.0
TOT_WORK_F	640.0	41295.760938	37192.360943	357.0	16097.75	30588.5	53234.25	257848.0

Fig 8: heatmap of these variables

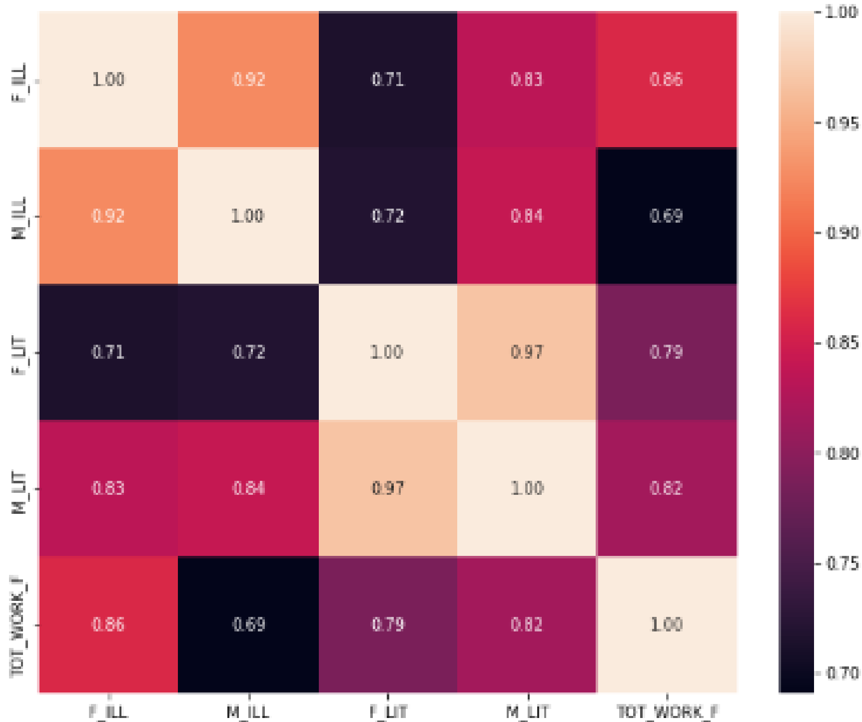
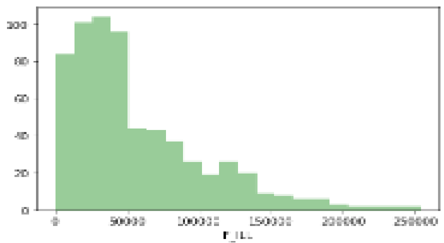
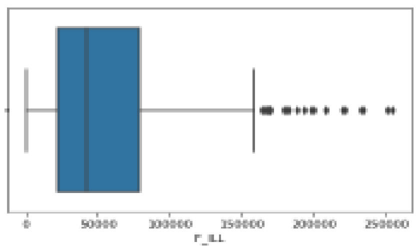


Fig 9: EDA of variables

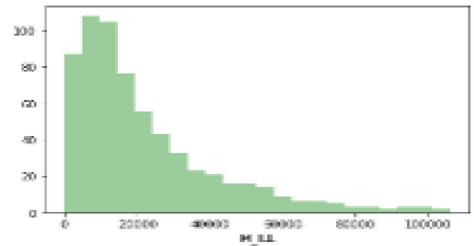
```
Description of F_ILL
--------------------------------------------------------------------------
count        640.000000
mean       56012.518750
std        47116.693769
min          327.000000
25%        22367.000000
50%        42386.000000
75%        78471.000000
max       254160.000000
Name: F_ILL, dtype: float64 Distribution of F_ILL
--------------------------------------------------------------------------
```



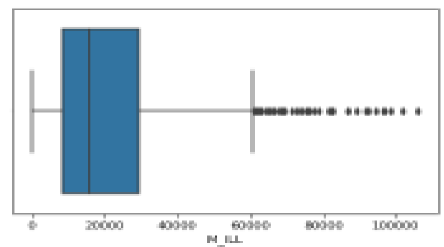

```
BoxPlot of F_ILL
--------------------------------------------------------------------------
```



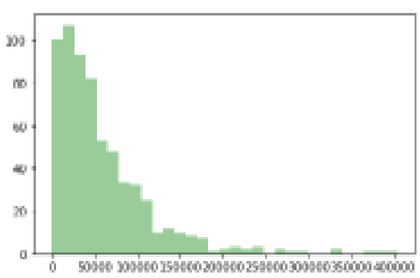

```
Description of M_ILL
--------------------------------------------------------------------------
count        640.000000
mean       21972.596875
std        19825.605268
min          105.000000
25%         8590.000000
50%        15767.500000
75%        29512.500000
max       105961.000000
Name: M_ILL, dtype: float64 Distribution of M_ILL
--------------------------------------------------------------------------
```



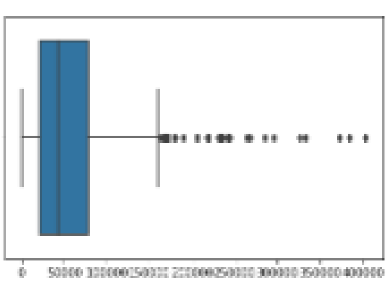

```
BoxPlot of M_ILL
--------------------------------------------------------------------------
```



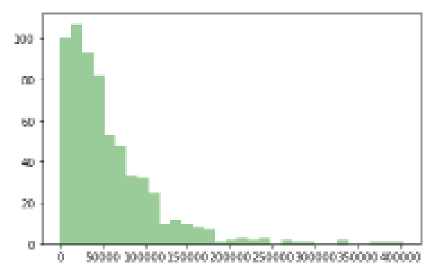

```
Description of M_LIT
--------------------------------------------------------------------------
count        640.000000
mean       57967.979688
std        55910.282466
min          286.000000
25%        21298.000000
50%        42693.500000
75%        77989.500000
max       403261.000000
Name: M_LIT, dtype: float64 Distribution of M_LIT
--------------------------------------------------------------------------
```



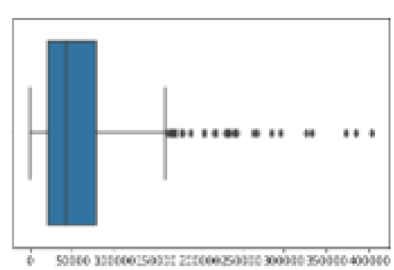

```
BoxPlot of M_LIT
--------------------------------------------------------------------------
```



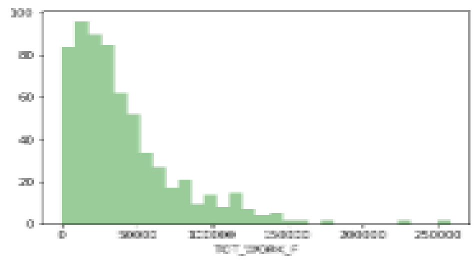

```
Description of M_LIT
--------------------------------------------------------------------------
count        640.000000
mean       57967.979688
std        55910.282466
min          286.000000
25%        21298.000000
50%        42693.500000
75%        77989.500000
max       403261.000000
Name: M_LIT, dtype: float64 Distribution of M_LIT
--------------------------------------------------------------------------
```


```
BoxPlot of M_LIT
--------------------------------------------------------------------------
```

```
Description of TOT_WORK_F
----------------------------------------------------------------------------------
count        640.000000
mean       41295.760938
std        37192.360943
min          357.000000
25%        16097.750000
50%        30588.500000
75%        53234.250000
max       257848.000000
Name: TOT_WORK_F, dtype: float64 Distribution of TOT_WORK_F
----------------------------------------------------------------------------------
```




```
BoxPlot of TOT_WORK_F
----------------------------------------------------------------------------------
```

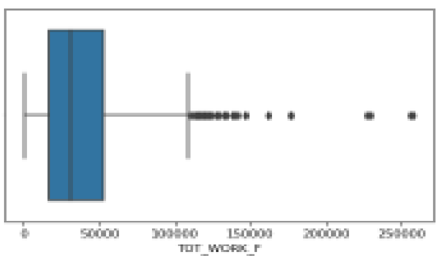
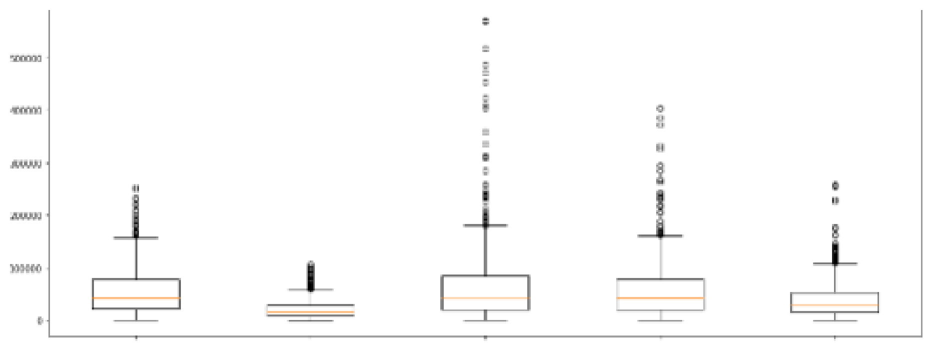


Fig 10: boxplot of five variables



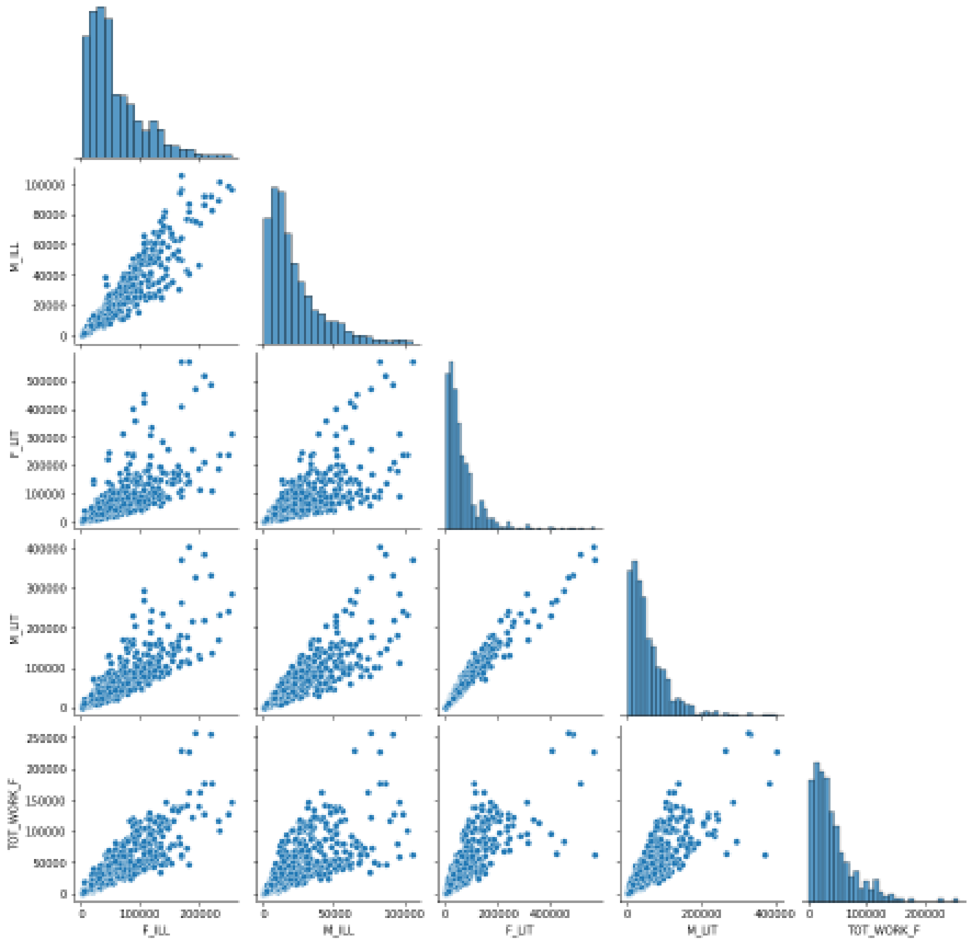


Fig 11: pair plot of variables

Q1 Is most of the male population is literate or illiterate ?

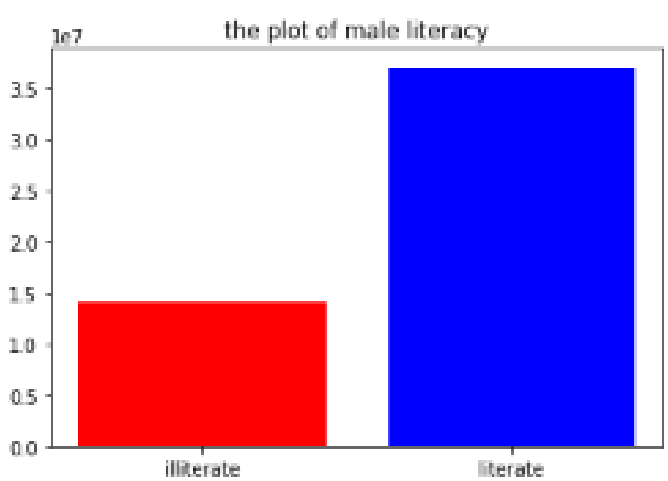


Fig 12: plot of male literacy

We can clearly see from the plot that most of the male population is literate.

Q2 Is most of the female population is literate or illiterate?

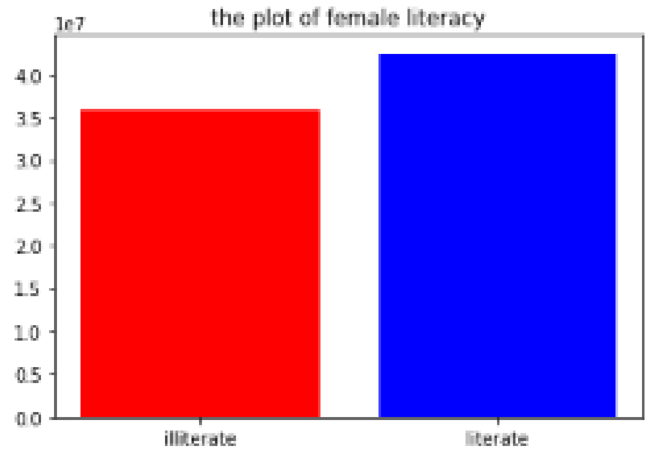


Fig 13: plot of female literacy

We can clearly see from the plot that most of the female population is literate

Q3 Which gender has more number of literate population ?

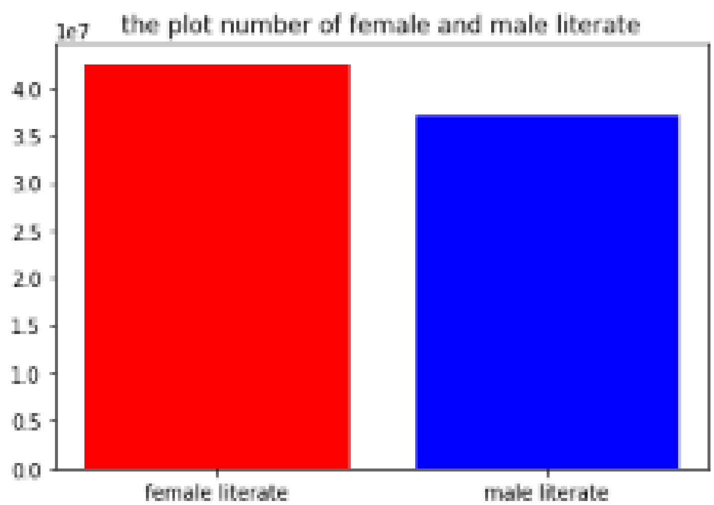


Fig 14: plot of literacy

We can clearly see from the plot that most of the female population is literate

Q4 which gender has more number of illiterate population ?

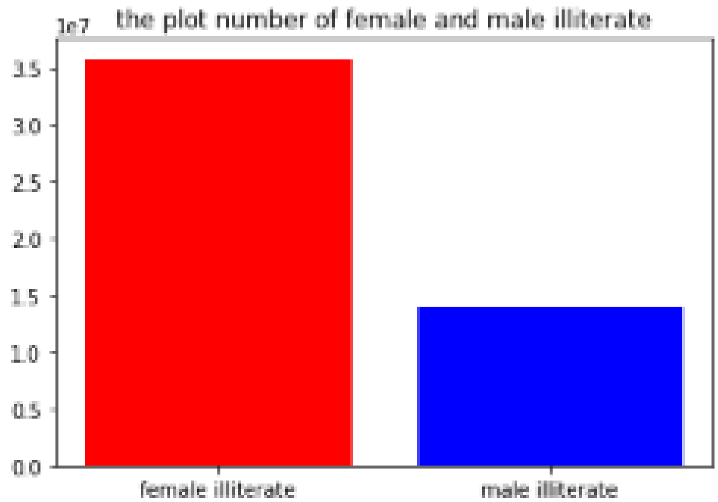


Fig 15: plot of illiteracy

We can clearly see from the plot that most of the female population is literate

Q5 which gender has most literate and illiterate population

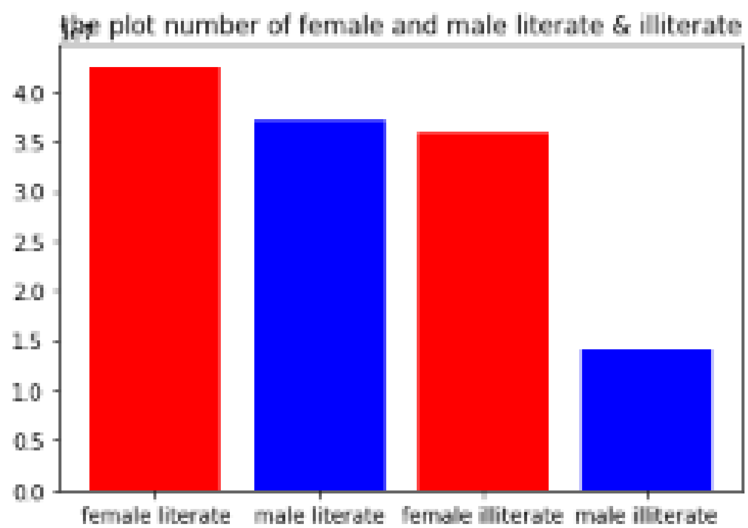


Fig 16: plot of illiteracy and literacy

We can clearly see from the plot that most of the female population is literate

And males have the least illiterate population

Q6 how much total working population of females is literate

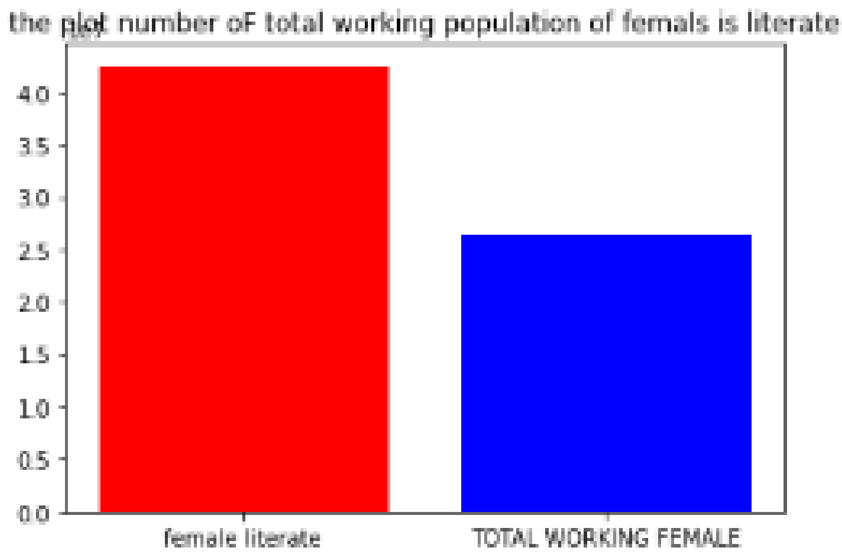


Fig 17: plot of total working population and female literate

We can clearly see from the plot that most of the female population is literate but only 50-60% of them are working.

2.3 - PCA: We choose not to treat outliers for this case. Do you think that treating outliers for this case is necessary?

Yes, treating outliers are necessary to treat outliers as it helps in getting the accurate data analysis results and helps in reducing errors.

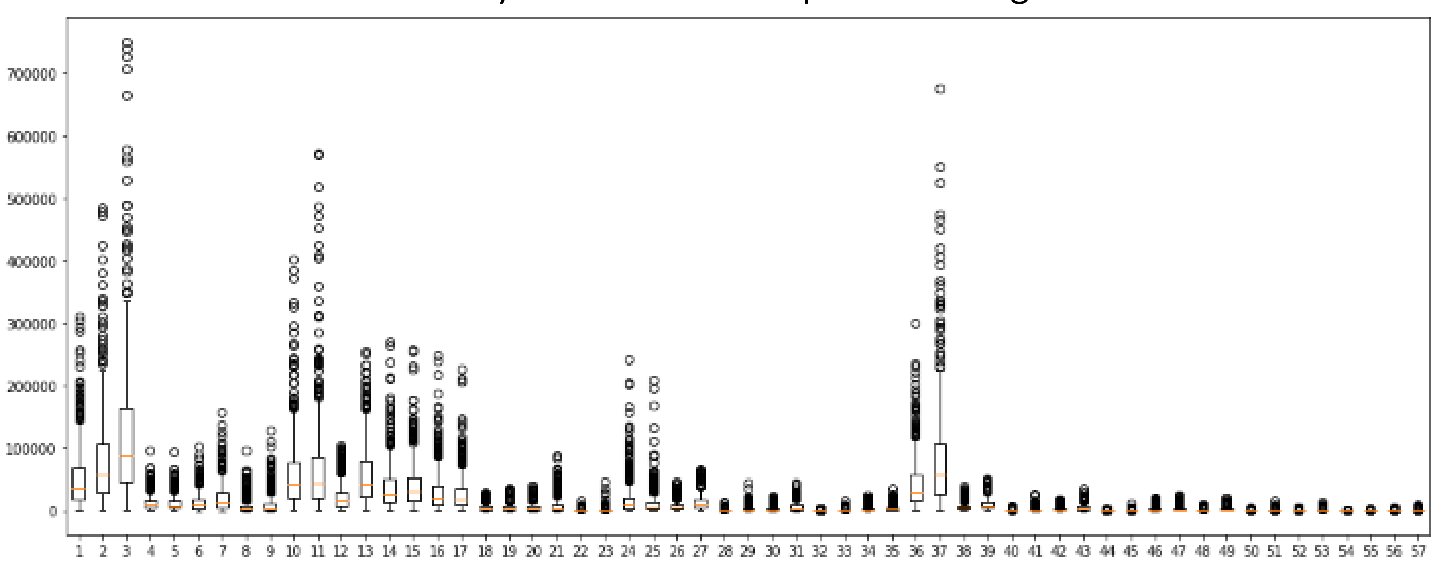


Fig 18: box plot of all variables

2.4 - PCA: Scale the Data using z-score method. Does scaling have any impact on outliers?

I have scaled the data with SciPy stat but we can also do scaling of the with sklearn.preprocessing.

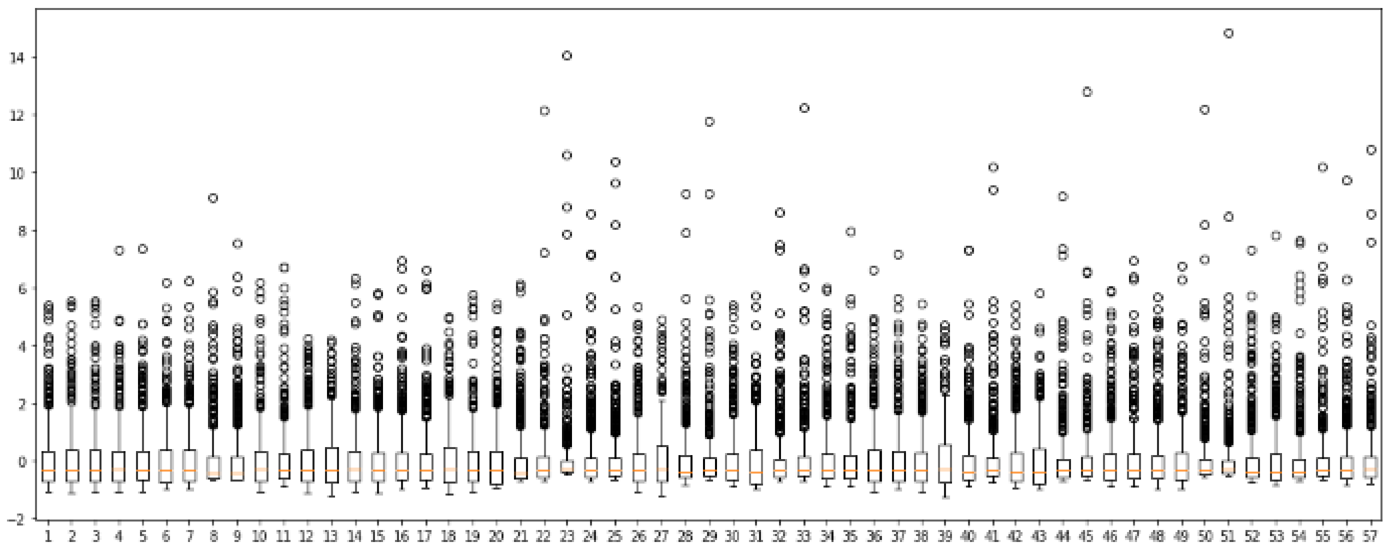


Fig 19: box plot of all variables

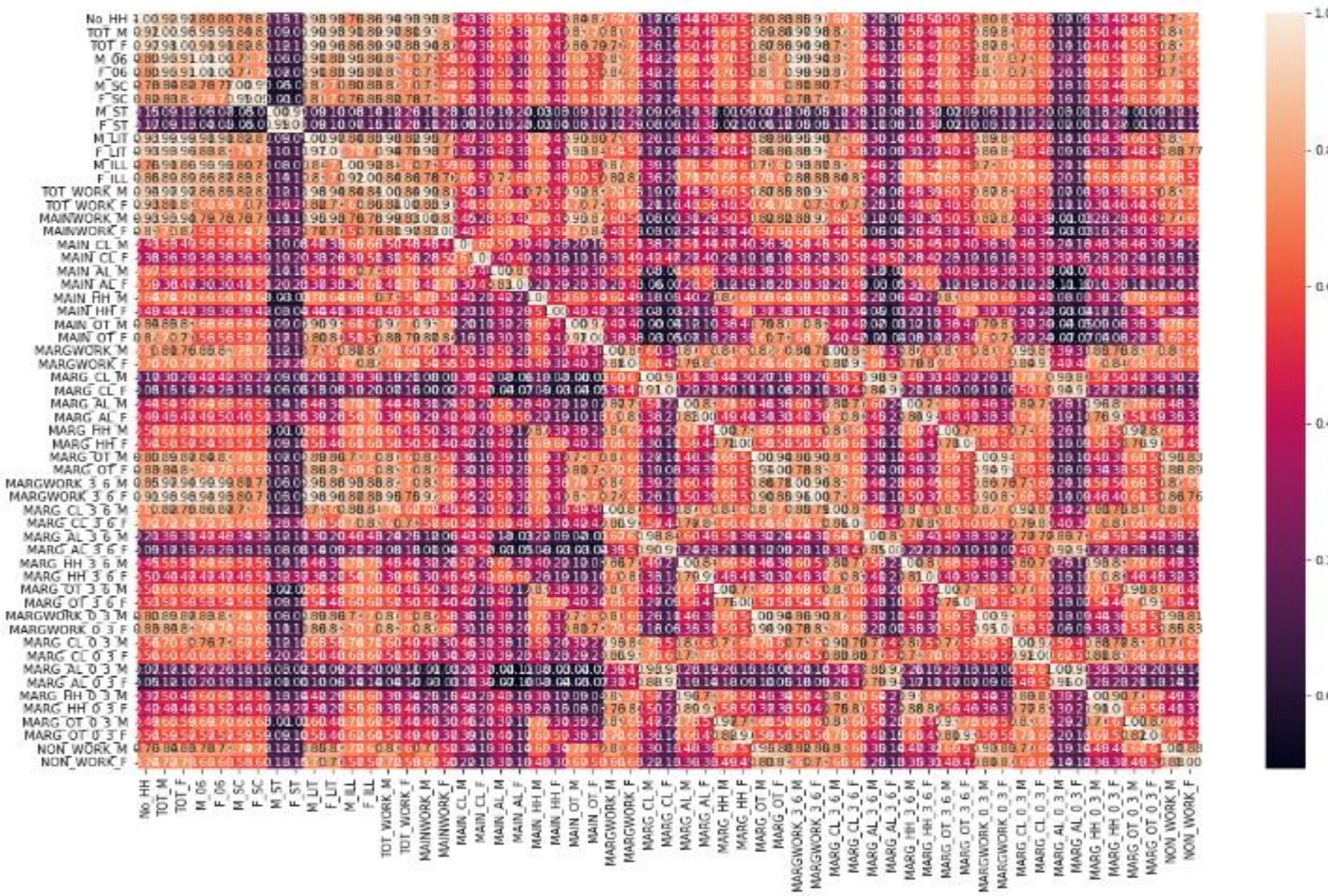


Fig 20: heatmap of all variables

2.5 - PCA: Perform all the required steps for PCA (use Sklenar only) Create the covariance Matrix Get eigen values and eigen vector.

We will check if the data is right to perform the PCA or not if it is not then we will not perform the PCA.

Firstly, we will Confirm the statistical significance of correlations for it we will calculate **bartlett sphericity** (H0: Correlations are not significant, H1: There are significant correlations, Reject H0 if p-value < 0.05). In the present data set the p values is 0.0

Second, we will Confirm the adequacy of sample size for it we will **calculate kmo**(Above 0.7 is good, below 0.5 is not acceptable) in our dataset we will get 0.8.

Now we know that the data is right so we will apply PCA from Sklenar.

With help of NumPy we get covariance matric:

```
array ([[2.65384958, 2.71559682, 3.0810424, ..., 3.32705346, 3.26012156,
        3.09943089],
       [2.71559682, 2.95818195, 3.21927718, ..., 3.52898196, 3.42192001,
        3.25382599],
       [3.0810424, 3.21927718, 3.85909673, ..., 4.05405594, 3.96445766,
        3.77161732],
       ...,
       [3.32705346, 3.52898196, 4.05405594, ..., 4.36443144, 4.25265838,
        4.05275828],
       [3.26012156, 3.42192001, 3.96445766, ..., 4.25265838, 4.15880459,
        3.96567682],
       [3.09943089, 3.25382599, 3.77161732, ..., 4.05275828, 3.96567682,
        3.79502824]])
```

With the help of Sklenar function we got eigen values: `array ([31.81356474, 7.86942415, 4.15340812, 3.66879058, 2.20652588, 1.93827502, 1.17617374, 0.75115909, 0.61705374, 0.52830088])`

And eigen vectors are

```
array ([[ 1.56020579e-01, 1.67117635e-01, 1.65553179e-01,
         1.62192948e-01, 1.62566396e-01, 1.51357849e-01,
         1.51566500e-01, 2.72341946e-02, 2.81833150e-02,
         1.61992837e-01, 1.46872680e-01, 1.61749445e-01,
         1.65248187e-01, 1.59871988e-01, 1.45935804e-01,
         1.46200730e-01, 1.23970284e-01, 1.03127159e-01,
         7.45397856e-02, 1.13355712e-01, 7.38821590e-02,
         1.31572584e-01, 8.33826397e-02, 1.23526242e-01,
         1.11021264e-01, 1.64615479e-01, 1.55395618e-01,
         8.23885414e-02, 4.91953957e-02, 1.28598563e-01,
         1.14305073e-01, 1.40853227e-01, 1.27669598e-01,
         1.55262872e-01, 1.47286584e-01, 1.64971950e-01,
         1.61253433e-01, 1.65501611e-01, 1.55647049e-01,
         9.30142064e-02, 5.15358640e-02, 1.28576116e-01,
         1.10645843e-01, 1.39592763e-01, 1.24545909e-01,
         1.54293786e-01, 1.46285654e-01, 1.50125706e-01,
         1.40157047e-01, 5.25417829e-02, 4.17859530e-02,
         1.21840354e-01, 1.16011410e-01, 1.39868774e-01,
         1.32192245e-01, 1.50375578e-01, 1.31066203e-01],
       [-1.26346525e-01, -8.96765481e-02, -1.04912371e-01,
        -2.20945086e-02, -2.02705495e-02, -4.51109032e-02,
        -5.19237543e-02, 2.76790387e-02, 3.02225550e-02,
        -1.15354767e-01, -1.53109487e-01, -6.62537318e-03,
        -9.10743682e-03, -1.33529221e-01, -8.50869689e-02,
        -1.76368057e-01, -1.51412544e-01, 6.24149874e-02,
```

```
 8.64767269e-02, -3.10403498e-02, -5.86880214e-02,
-7.60210677e-02, -8.24766375e-02, -2.12984254e-01,
-2.10071166e-01, 9.29935012e-02, 1.25269967e-01,
 2.69449716e-01, 2.46546811e-01, 1.65830750e-01,
 1.40957749e-01, 6.80679428e-02, 2.42164125e-02,
-8.94419720e-02, -1.17899307e-01, -4.39949601e-02,
-1.05501898e-01, 7.71926975e-02, 1.03173976e-01,
 2.64409408e-01, 2.44261317e-01, 1.58782773e-01,
 1.25286970e-01, 6.22623250e-02, 1.47659019e-02,
-9.31585894e-02, -1.25595577e-01, 1.50680869e-01,
 1.80690375e-01, 2.51328442e-01, 2.40719745e-01,
 1.85277342e-01, 1.80615650e-01, 8.48690452e-02,
 5.08133220e-02, -6.53645529e-02, -7.38474208e-02],
[-2.69025027e-03, 5.66976190e-02, 3.87494746e-02,
 5.77881518e-02, 5.01255677e-02, 2.56890397e-03,
-2.51008795e-02, -1.23504453e-01, -1.39768833e-01,
 8.21676677e-02, 1.17097683e-01, -2.18550935e-02,
-9.30623763e-02, 4.51763691e-02, -5.94495457e-02,
 5.42945289e-02, -5.56090961e-02, -6.73992944e-02,
-9.23808946e-03, -2.47917055e-01, -2.51932296e-01,
 2.65689386e-02, -6.05232993e-02, 1.37377881e-01,
 9.56339841e-02, -8.62782527e-03, -4.93697036e-02,
 1.98754143e-01, 2.68786906e-01, -1.89867566e-01,
-2.67767729e-01, -2.12567389e-02, -8.25040484e-02,
 1.11712747e-01, 1.00045670e-01, 6.44232083e-02,
 7.97035639e-02, -2.42054166e-02, -7.20134423e-02,
 1.53517557e-01, 2.56212918e-01, -2.00118572e-01,
-2.79866018e-01, -2.06182531e-02, -8.27935649e-02,
 1.10285441e-01, 9.56665986e-02, 5.48919412e-02,
 2.39815623e-02, 2.68330072e-01, 2.84955665e-01,
-1.38627894e-01, -2.02198401e-01, -2.25985196e-02,
-7.87198692e-02, 1.11827318e-01, 1.02552501e-01],
[-1.25293371e-01, -1.99415702e-02, -7.08726202e-02,
 1.19171726e-02, 1.48442005e-02, 1.24850957e-02,
-2.98925083e-02, -2.22247412e-01, -2.29754420e-01,
-3.51625572e-02, -5.95594178e-02, 2.53483370e-02,
-7.60233569e-02, -4.01544117e-02, -2.25160033e-01,
-6.83507465e-02, -2.46639865e-01, -8.97686819e-02,
-2.88964883e-01, -1.36082339e-01, -2.90042169e-01,
 1.52366335e-01, 4.89504702e-02, -4.02891831e-02,
-1.20391064e-01, 9.30182650e-02, -8.87071351e-02,
-6.27609058e-02, -1.68401590e-01, 9.17874509e-02,
-1.06365430e-01, 2.37984720e-01, 1.96320743e-01,
 8.71191186e-02, 2.67292472e-02, -2.55415177e-05,
 3.89358966e-03, 9.28748901e-02, -1.07860188e-01,
-3.84875762e-02, -1.79691340e-01, 8.04108514e-02,
-1.36240262e-01, 2.37744957e-01, 1.90510604e-01,
 8.64794098e-02, 2.72754576e-02, 8.74333682e-02,
```

```
 -2.22902303e-02,  -1.04686028e-01,  -1.35715829e-01,
  1.32544187e-01,   4.05131037e-03,   2.30037988e-01,
  2.06200724e-01,   8.48540392e-02,   2.11244736e-02],
[-7.02208326e-03,  -3.30261797e-02,  -1.28467033e-02,
 -5.02475101e-02,  -4.38479670e-02,  -1.73006734e-01,
 -1.59803417e-01,   4.33163419e-01,   4.38791921e-01,
 -9.10133039e-03,   5.58437008e-02,  -9.65797550e-02,
 -1.19910504e-01,  -1.95528839e-02,  -4.04373671e-02,
 -3.68019626e-02,  -8.28338587e-02,  -2.86039079e-01,
 -2.41936366e-01,  -2.05723504e-01,  -1.77604766e-01,
 -1.34088831e-01,  -1.39440884e-01,   6.46377099e-02,
  8.07427680e-02,   6.02435623e-02,   8.92022890e-02,
 -2.22632349e-02,  -5.92051969e-02,   1.94217749e-02,
  8.05270982e-02,  -5.99705215e-02,  -3.36016572e-02,
  1.19120710e-01,   1.66882498e-01,  -4.38337603e-02,
  5.37136660e-04,   5.40731673e-02,   7.30497551e-02,
 -7.78921571e-03,  -6.13026163e-02,   8.45700429e-03,
  6.41091701e-02,  -6.64002513e-02,  -4.48097937e-02,
  1.08828726e-01,   1.41190165e-01,   8.11854109e-02,
  1.29936485e-01,  -4.88490365e-02,  -5.18948148e-02,
  6.23798835e-02,   1.28308262e-01,  -3.63901233e-02,
  1.64823759e-04,   1.62862488e-01,   2.38291984e-01],
[ 4.08281236e-03,  -7.33892582e-02,  -4.36468776e-02,
 -1.57956741e-01,  -1.54435838e-01,  -6.42950849e-02,
 -4.05178242e-02,   2.22590756e-01,   2.25530984e-01,
 -5.54647180e-02,  -4.80207140e-02,  -1.15234150e-01,
 -2.87572169e-02,  -1.80069366e-03,   1.05162304e-01,
  1.92826056e-02,   1.23832265e-01,  -6.16981242e-03,
  1.02951208e-01,  -3.10678194e-02,   1.92397157e-02,
  1.74465381e-01,   4.22309325e-01,   2.34771546e-02,
  8.30792178e-02,  -9.07614920e-02,   1.78676850e-02,
  3.19145578e-02,   9.20857153e-02,  -1.41604662e-01,
 -8.51201684e-02,   8.95332953e-02,   3.65111795e-01,
 -6.10655989e-02,   1.73934355e-03,  -1.36253329e-01,
 -1.06900175e-01,  -9.67084917e-02,   2.37730742e-02,
  1.34769857e-02,   9.39927460e-02,  -1.44060823e-01,
 -7.67084401e-02,   9.70574753e-02,   3.84552143e-01,
 -6.20430271e-02,   8.96165312e-03,  -6.07153533e-02,
 -1.72704069e-03,   6.54086839e-02,   8.37426375e-02,
 -1.24209277e-01,  -1.05529913e-01,   6.12279896e-02,
  2.95599696e-01,  -5.23862474e-02,  -2.49010756e-02],
[-1.18110420e-01,   8.95543279e-02,  -2.12425804e-03,
  1.65066693e-01,   1.69082313e-01,  -1.56616120e-03,
 -8.46576325e-02,   4.05505171e-01,   3.57799733e-01,
  4.59340904e-02,  -2.10641811e-02,   2.01946551e-01,
  2.84250386e-02,   4.50530285e-02,  -1.19423828e-01,
  4.73668824e-02,  -9.04307518e-02,   3.85792299e-01,
  2.07881596e-01,  -1.30773481e-02,  -1.58333973e-01,
```

```
  1.19824865e-01,  -1.39294074e-01,  -1.56013662e-02,
 -7.06449807e-02,   2.01945537e-02,  -1.57222087e-01,
  2.90721618e-02,  -4.58837977e-02,   2.02976379e-02,
 -1.50711850e-01,   1.08603621e-01,  -4.94715251e-02,
 -4.28750247e-03,  -1.17886196e-01,   1.26291461e-01,
  5.06245188e-02,   2.66713701e-02,  -1.38020671e-01,
  6.32740099e-02,  -1.92212141e-02,   2.14514915e-02,
 -1.46120967e-01,   1.15067653e-01,  -4.22393179e-02,
 -1.20438787e-03,  -9.10604465e-02,  -7.31610967e-03,
 -2.00877437e-01,  -4.22951220e-02,  -1.03406947e-01,
  1.45872850e-02,  -1.52174700e-01,   8.31223341e-02,
 -6.87217760e-02,  -1.93539659e-02,  -2.00052834e-01],
[ 5.72389403e-02,   1.11431209e-01,   8.83553113e-02,
  1.69594884e-01,   1.69458172e-01,  -1.29301290e-01,
 -1.44352081e-01,   2.19818496e-02,   1.48736254e-02,
  9.94228857e-02,   1.10359861e-01,   1.32079860e-01,
  3.72704185e-02,   7.68689545e-02,  -4.12544522e-02,
  8.79616816e-02,  -1.80367069e-02,  -2.31343705e-01,
 -2.99574045e-01,   5.17576329e-02,  -1.13997037e-01,
 -1.35092829e-01,   3.77112324e-01,   1.42202440e-01,
  6.96842900e-02,   4.10448554e-03,  -9.03271864e-02,
  7.38189176e-02,   4.89339728e-03,   1.41822624e-01,
 -2.51860324e-02,  -2.06463038e-01,   7.49567994e-02,
 -1.40518702e-01,  -2.25984329e-01,   1.37732088e-01,
  1.39466320e-01,  -2.69871622e-03,  -1.17627847e-01,
  6.80031882e-02,  -1.94446250e-02,   1.40602227e-01,
 -5.20077614e-02,  -2.13660011e-01,   8.44478387e-02,
 -1.42040131e-01,  -2.24162681e-01,   3.13237674e-02,
  9.78947498e-04,   7.77140227e-02,   5.90416261e-02,
  1.39166557e-01,   6.74209419e-02,  -1.74949447e-01,
  4.47046152e-02,  -1.24166132e-01,  -2.02142424e-01],
[ 4.26426297e-03,   1.88718994e-02,   1.49110483e-02,
 -5.67729534e-02,  -5.93231092e-02,   3.74800933e-02,
  4.12320246e-02,   1.86322807e-02,   4.38659393e-02,
  4.51935138e-02,   2.19962970e-02,  -5.75960731e-02,
  9.20071457e-04,   4.52565826e-02,   1.14253939e-01,
  6.20674638e-02,   1.43678886e-01,  -3.64574761e-01,
 -1.13373697e-01,  -1.79033216e-01,   2.49005270e-02,
  3.83824269e-01,  -2.14204450e-01,   1.66356835e-01,
  2.78022734e-01,  -4.12529185e-02,  -5.52388166e-03,
  1.56551464e-02,   2.37227547e-02,  -1.13496935e-02,
  9.19244295e-02,   2.38752017e-01,  -1.47744801e-01,
 -1.09101817e-01,  -1.43586594e-01,  -6.74392520e-03,
 -3.07997521e-02,  -4.78759050e-02,   4.34856707e-03,
  6.04167971e-03,   1.23446283e-02,  -2.77157875e-02,
  7.99587767e-02,   2.42022448e-01,  -1.69232875e-01,
 -9.73659075e-02,  -8.06274087e-02,  -1.17560663e-02,
 -3.49847304e-02,   3.32222805e-02,   4.80868090e-02,
```

```
 5.49180458e-02,             1.23663504e-01,             2.19106709e-01,
-8.00360275e-02,            -1.60651654e-01,            -3.54392625e-01],
[           1.99858786e-02,          -2.45018137e-02,   -3.80403962e-02,
-1.53572957e-01,         -1.69566102e-01,               4.48517063e-01,
 4.46968631e-01,             1.60418089e-01,             1.34862457e-01,
-5.75150260e-03,            -4.06669110e-02,            -7.44736649e-02,
-2.69513459e-02,                 8.01538493e-02,        -7.17307919e-02,
 9.20972776e-02,         -5.01329263e-02,               6.54575202e-02,
-3.58864622e-01,             2.63960172e-01,             8.99292496e-02,
 1.25604473e-02,         -1.45091255e-01,               3.39036748e-02,
-1.51815819e-02,                 2.67980655e-03,        -1.05846361e-01,
 4.39105156e-02,         -1.68268728e-03,               2.37382369e-02,
-1.11565248e-01,         -6.56107508e-02,               7.44503857e-03,
-2.69139751e-02,            -9.99064061e-02,            -1.20813970e-01,
-1.99297420e-02,            -3.26079861e-02,            -1.55205351e-01,
-9.78232111e-03,         -6.03894553e-02,               1.21088009e-02,
-1.29665982e-01,            -7.33733362e-02,            -1.58073738e-02,
-6.52498031e-02,         -1.37432885e-01,               1.45097829e-01,
 5.38803891e-02,             1.46579723e-01,             1.29528538e-01,
 6.91332224e-02,            -4.02920095e-02,            -3.73917620e-02,
 7.47448958e-02,  1.65440091e-01,  5.07798151e-02]])
```

Explained variance for each pc is `array ([0.55726063, 0.13784435, 0.07275295, 0.06426418, 0.03865049, 0.03395169, 0.02060239, 0.01315764, 0.01080859, 0.00925395])`

After checking all these things, we will create a data frame of PCs named df_extracted_loadings

2.6 - PCA: Identify the optimum number of PCs (for this project, take at least 90% explained variance). Show Scree plot.

Table18: table of PCs

	No_HH	TOT_M	TOT_F	M_06	F_06	M_SC	F_SC	M_ST	F_ST	M_LIT	...
PC1	0.156021	0.167118	0.165553	0.162193	0.162566	0.151358	0.151567	0.027234	0.028183	0.161993	...
PC2	-0.126347	-0.089677	-0.104912	-0.022095	-0.020271	-0.045111	-0.051924	0.027879	0.030223	-0.115355	...
PC3	-0.002690	0.056898	0.038749	0.057788	0.050126	0.002569	-0.025101	-0.123504	-0.139769	0.082168	...
PC4	-0.125293	-0.019942	-0.070873	0.011917	0.014844	0.012485	-0.029893	-0.222247	-0.229754	-0.035183	...
PC5	-0.007022	-0.033026	-0.012847	-0.050248	-0.043848	-0.173007	-0.159803	0.433183	0.438792	-0.009101	...
PC6	0.004083	-0.073389	-0.043847	-0.157957	-0.154436	-0.064295	-0.040518	0.222591	0.225531	-0.055465	...
PC7	-0.118110	0.089554	-0.002124	0.165067	0.169082	-0.001586	-0.084658	0.405505	0.357800	0.045934	...
PC8	0.057239	0.111431	0.088355	0.169595	0.169458	-0.129301	-0.144352	0.021982	0.014874	0.099423	...
PC9	0.004284	0.018872	0.014911	-0.056773	-0.059323	0.037480	0.041232	0.018832	0.043866	0.045194	...
PC10	0.019986	-0.024502	-0.038040	-0.153573	-0.169566	0.448517	0.446989	0.160418	0.134862	-0.005752	...

	MARG_CL_0_3_M	MARG_CL_0_3_F	MARG_AL_0_3_M	MARG_AL_0_3_F	MARG_HH_0_3_M	MARG_HH_0_3_F	MARG_OT_0_3_M	MARG_OT_0_3_F	NON_WORK_M	NON_WORK_F
PC1	0.150128	0.140157	0.052542	0.041786	0.121840	0.116011	0.139869	0.132192	0.150376	0.131066
PC2	0.150681	0.180690	0.251326	0.240720	0.185277	0.180616	0.084869	0.050813	-0.085385	-0.073847
PC3	0.054892	0.023982	0.288330	0.284956	-0.138826	-0.202198	-0.022569	-0.078720	0.111827	0.102563
PC4	0.087433	-0.022290	-0.104686	-0.135716	0.132544	0.004051	0.230038	0.206201	0.084854	0.021124
PC5	0.081185	0.129938	-0.048849	-0.051895	0.062380	0.128308	-0.038390	0.000165	0.162662	0.238292
PC6	-0.080715	-0.001727	0.085409	0.083743	-0.124209	-0.105530	0.061228	0.295800	-0.052386	-0.024901
PC7	-0.007316	-0.200877	-0.042295	-0.103407	0.014587	-0.152175	0.083122	-0.068722	-0.019354	-0.200063
PC8	0.031324	0.000979	0.077714	0.059042	0.139167	0.067421	-0.174949	0.044705	-0.124166	-0.202142
PC9	-0.011756	-0.034985	0.033222	0.048087	0.054916	0.123884	0.219107	-0.080036	-0.160652	-0.354393
PC10	0.145068	0.053880	0.146580	0.129529	0.069133	-0.040292	-0.037392	0.074745	0.165440	0.050780

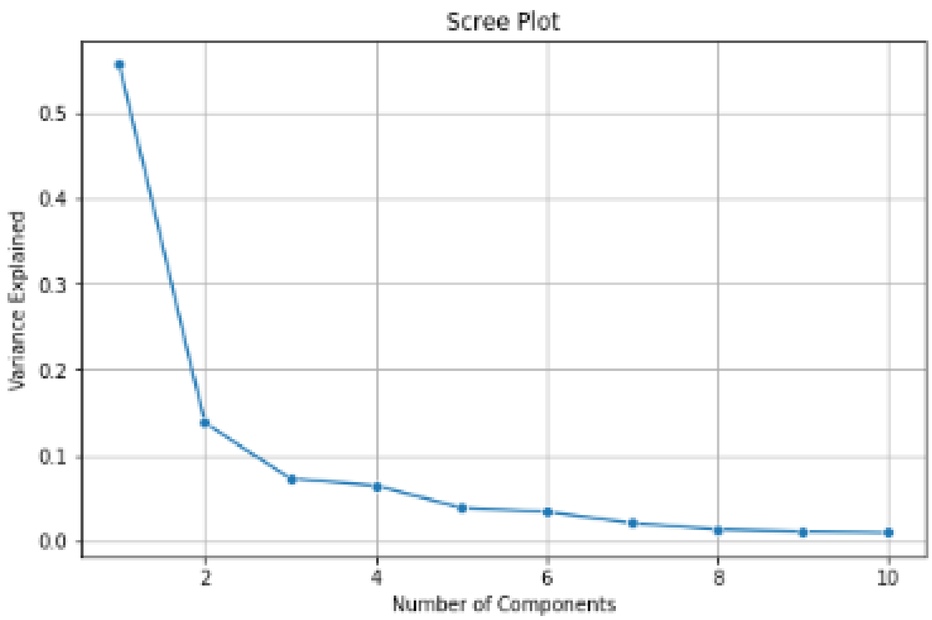


Fig21: scree plot

We will check the cumulative explained variance ratio to find a cut off for selecting the number of PCs (cut off is 90%)

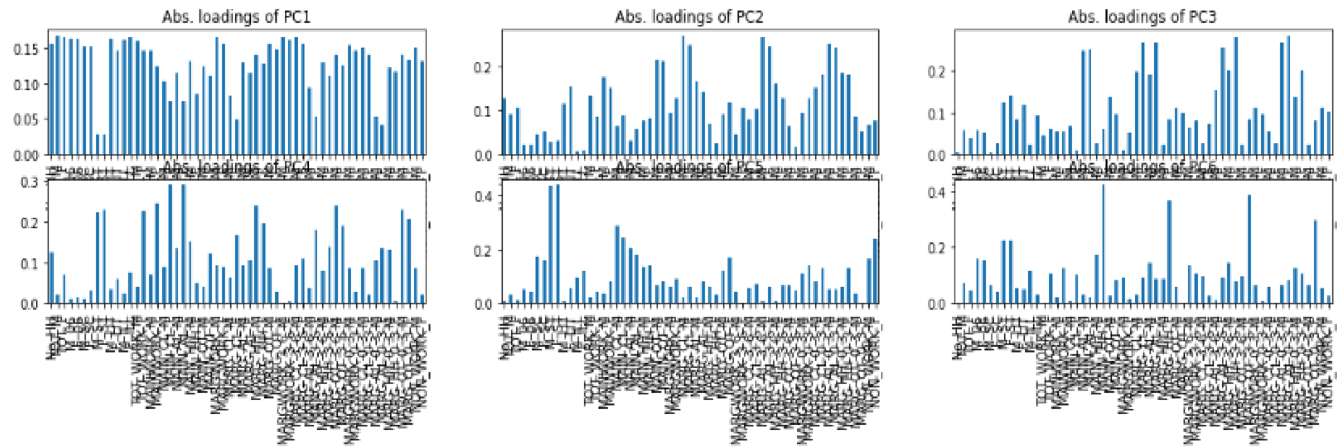
```
array ([0.55726063, 0.69510499, 0.76785794, 0.83212212, 0.87077261,
       0.9047243, 0.92532669, 0.93848433, 0.94929292, 0.95854687])
```

From the scree plot and from cumulative explained variance we can see that for 90% cutoff we will require 6 PCs as after which there is not much difference in the data.

Df_selected is the data frame with selected 6 PCs.

2.7 - PCA: Compare PCs with Actual Columns and identify which is explaining most variance. Write inferences about all the principal components in terms of actual variables.

Fig 22: compare pcs



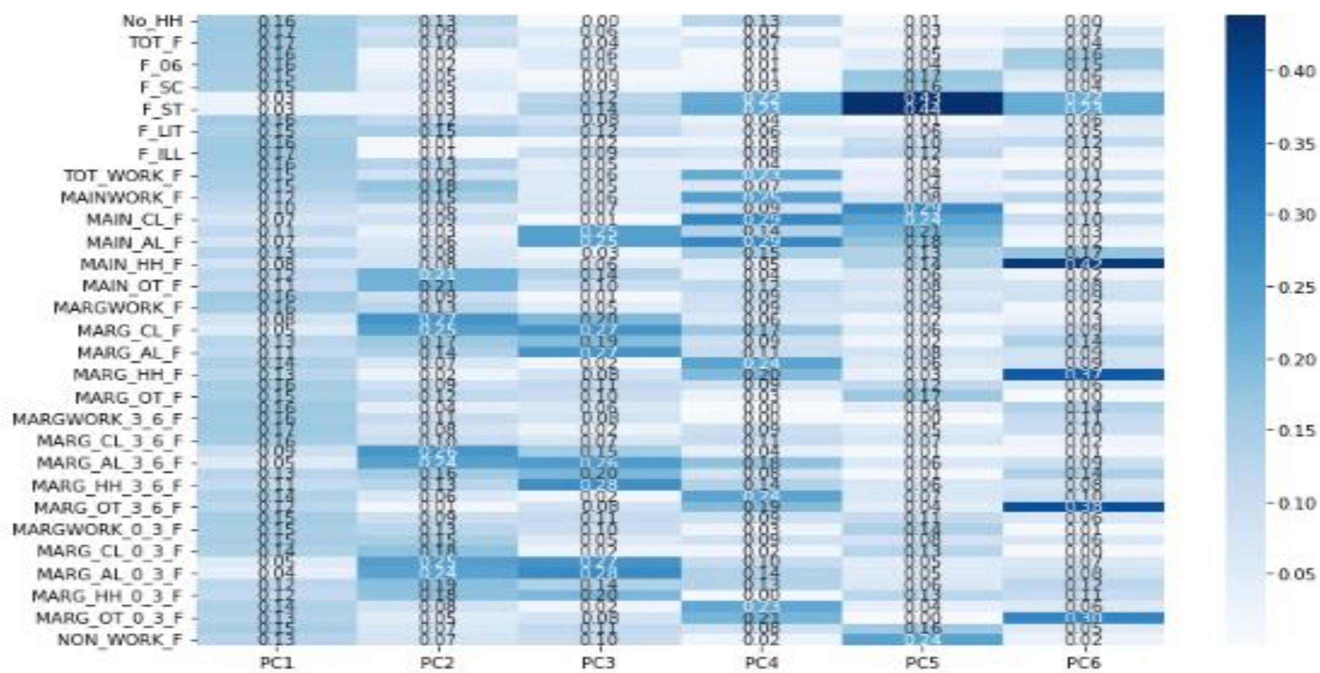


Fig 23: heatmap of pcs

we will extract the required number of PCs which is 6.

we will create a data frame out of fit_transformed scaled data above.

PC 1 explain most of the variance as we can see .

PC1 has captures most of the data.

Table 19: table of pcs

	PC1	PC2	PC3	PC4	PC5	PC6
0	-4.617263	0.138116	0.328545	1.543697	0.353736	-0.420948
1	-4.771662	-0.105865	0.244449	1.963215	-0.153884	0.417308
2	-5.964836	-0.294347	0.367394	0.619543	0.478199	0.278581
3	-6.280796	-0.500384	0.212701	1.074515	0.300799	0.051157
4	-4.478566	0.894154	1.078277	0.535557	0.804065	0.341678
5	-3.319963	2.823865	3.058460	-0.447904	0.742445	0.634676
6	-5.021393	-0.348359	0.650378	0.981072	-0.059778	-0.248957
7	-4.608709	0.022370	0.398755	1.576995	0.171316	-0.139444
8	-5.186703	-0.059097	0.184397	1.735440	0.169174	0.455039
9	-4.226190	-1.335080	0.697838	1.470509	0.269146	-0.002576

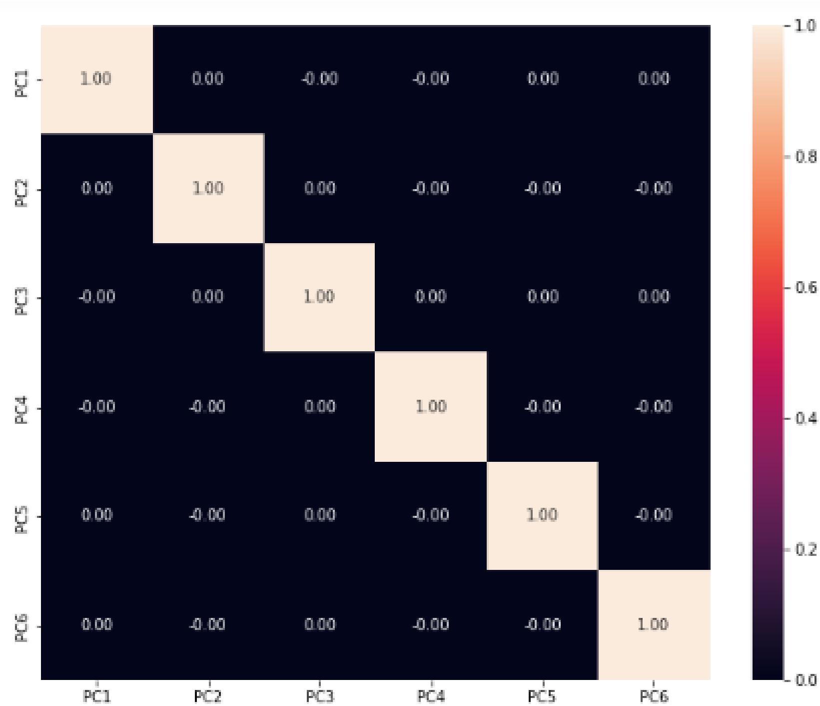


Fig 24: correlation matrix

2.8 - PCA: Write linear equation for first PC.

linear equation formula: PC1 = a1x1+a2x2+a3x3+a4x4+....a57x57

a1,a2,a2..= Coefficients of PC1

```
array([0.15602058,     0.16711763,     0.16555318,     0.16219295,     0.1625664     ,
       0.15135785,     0.1515665     ,     0.02723419,     0.02818332,     0.16199284,
```

```
0.14687268,     0.16174944,     0.16524819,     0.15987199,     0.1459358     ,
0.14620073,      0.12397028,      0.10312716,      0.07453979,      0.11335571,
0.07388216,      0.13157258,      0.08338264,      0.12352624,      0.11102126,
0.16461548,     0.15539562,     0.08238854,     0.0491954      ,     0.12859856,
0.11430507,     0.14085323,     0.1276696      ,     0.15526287,     0.14728658,
0.16497195,      0.16125343,      0.16550161,      0.15564705,      0.09301421,
0.05153586,      0.12857612,      0.11064584,      0.13959276,      0.12454591,
0.15429379,      0.14628565,      0.15012571,      0.14015705,      0.05254178,
0.04178595,      0.12184035,      0.11601141,      0.13986877,      0.13219224,
0.15037558, 0.1310662 ])
```

x1,x2,x3..= Variables of PC1

No_HH	-0.904738
TOT_M	-0.771236
TOT_F	-0.815563
M_06	-0.561012
F_06	-0.507738
M_SC	-0.958575
F_SC	-0.957049
M_ST	-0.423306
F_ST	-0.476423
M_LIT	-0.798097
F_LIT	-0.733477
M_ILL	-0.604015
F_ILL	-0.798229
TOT_WORK_M	-0.859260
TOT_WORK_F	-1.010238
MAINWORK_M	-0.872367
MAINWORK_F	-0.898216
MAIN_CL_M	-1.042844
MAIN_CL_F	-0.986630
MAIN_AL_M	-0.851060
MAIN_AL_F	-0.683276
MAIN_HH_M	-0.630766
MAIN_HH_F	-0.407555
MAIN_OT_M	-0.624042
MAIN_OT_F	-0.611637
MARGWORK_M	-0.516943
MARGWORK_F	-0.966512
MARG_CL_M	-0.321809
MARG_CL_F	-0.485053
MARG_AL_M	-0.331426
MARG_AL_F	-0.860192
MARG_HH_M	-0.377984
MARG_HH_F	-0.453026
MARG_OT_M	-0.548764
MARG_OT_F	-0.614625

```
MARGWORK_3_6_M        -0.648040
MARGWORK_3_6_F        -0.663795
MARG_CL_3_6_M         -0.595998
MARG_CL_3_6_F         -1.017848
MARG_AL_3_6_M         -0.387707
MARG_AL_3_6_F         -0.563854
MARG_HH_3_6_M         -0.448658
MARG_HH_3_6_F         -0.896723
MARG_OT_3_6_M         -0.377635
MARG_OT_3_6_F         -0.431307
MARGWORK_0_3_M        -0.569151
MARGWORK_0_3_F        -0.612451
MARG_CL_0_3_M         -0.163229
MARG_CL_0_3_F         -0.720610
MARG_AL_0_3_M         -0.156494
MARG_AL_0_3_F         -0.287524
MARG_HH_0_3_M          0.156577
MARG_HH_0_3_F         -0.657412
MARG_OT_0_3_M         -0.365258
MARG_OT_0_3_F         -0.499977
NON_WORK_M            -0.413053
NON_WORK_F            -0.539614
```

Linear equation of pc1= Coefficients PC1 * Variables

So, the sum after Coefficients of PC1 * Variables is -4.617263481554375.

Now if we compare our PC1 score with original value i.e in our pca_final_df data frame both values are same (-4.617263481554375).