DATA MINING
BUSINESS REPORT

THANUSRI A

03/12/2023

PROBELEM 1

- Read the data and perform basic analysis such as printing a few rows (head and tail), info, data summary, null values duplicate values, etc.
- Treat missing values in CPC, CTR and CPM using the formula given. to understand the coding behind treating the missing values using a specific formula. You have to basically create an user defined function and then call the function for imputing.
- 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 an analyst your judgement may be different from another analyst).
- Perform z-score scaling and discuss how it affects the speed of the algorithm.
- Perform clustering and do the following:
- Perform Hierarchical by constructing a Dendrogram using WARD and Euclidean distance.
- Make Elbow plot (up to n=10) and identify optimum number of clusters for k-means algorithm.
- Print silhouette scores for up to 10 clusters and identify optimum number of clusters.
- Profile the ads based on optimum number of clusters using silhouette score and your domain understanding
  [Hint: Group the data by clusters and take sum or mean to identify trends in clicks, spend, revenue, CPM, CTR, & CPC based on Device Type. Make bar plots.]
- Conclude the project by providing summary of your learnings.

Problem 1

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

Top 5 rows of the data set:

	0	1	2	3	4
Timestamp	2020-9-2-17	2020-9-2-10	2020-9-1-22	2020-9-3-20	2020-9-4-15
InventoryType	Format1	Format1	Format1	Format1	Format1
Ad - Length	300	300	300	300	300
Ad- Width	250	250	250	250	250
Ad Size	75000	75000	75000	75000	75000
Ad Type	Inter222	Inter227	Inter222	Inter228	Inter217
Platform	Video	App	Video	Video	Web
Device Type	Desktop	Mobile	Desktop	Mobile	Desktop
Format	Display	Video	Display	Video	Video
Available_Impressions	1806	1780	2727	2430	1218
Matched_Queries	325	285	356	497	242
Impressions	323	285	355	495	242
Clicks	1	1	1	1	1
Spend	0.0	0.0	0.0	0.0	0.0
Fee	0.35	0.35	0.35	0.35	0.35
Revenue	0.0	0.0	0.0	0.0	0.0
CTR	0.0031	0.0035	0.0028	0.002	0.0041
CPM	0.0	0.0	0.0	0.0	0.0
CPC	0.0	0.0	0.0	0.0	0.0

Bottom 5 rows of the data set:

	23061	23062	23063	23064	23065
Timestamp	2020-9-13-7	2020-11-2-7	2020-9-14-22	2020-11-18-2	2020-9-14-0
InventoryType	Format5	Format5	Format5	Format4	Format5
Ad - Length	720	720	720	120	720
Ad- Width	300	300	300	600	300
Ad Size	216000	216000	216000	72000	216000
Ad Type	Inter220	Inter224	Inter218	inter230	Inter221
Platform	Web	Web	App	Video	App
Device Type	Mobile	Desktop	Mobile	Mobile	Mobile
Format	Video	Video	Video	Video	Video
Available_Impressions	1	3	2	7	2
Matched_Queries	1	2	1	1	2
Impressions	1	2	1	1	2
Clicks	1	1	1	1	1
Spend	0.07	0.04	0.05	0.07	0.09
Fee	0.35	0.35	0.35	0.35	0.35
Revenue	0.0455	0.026	0.0325	0.0455	0.0585
CTR	NaN	NaN	NaN	NaN	NaN
CPM	NaN	NaN	NaN	NaN	NaN
CPC	NaN	NaN	NaN	NaN	NaN

Basic info about the data set:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23066 entries, 0 to 23065
Data columns (total 19 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Timestamp              23066 non-null  object
 1   InventoryType          23066 non-null  object
 2   Ad - Length            23066 non-null  int64
 3   Ad- Width              23066 non-null  int64
 4   Ad Size                23066 non-null  int64
 5   Ad Type                23066 non-null  object
 6   Platform               23066 non-null  object
 7   Device Type            23066 non-null  object
 8   Format                 23066 non-null  object
 9   Available_Impressions  23066 non-null  int64
 10  Matched_Queries        23066 non-null  int64
 11  Impressions            23066 non-null  int64
 12  Clicks                 23066 non-null  int64
 13  Spend                  23066 non-null  float64
 14  Fee                    23066 non-null  float64
 15  Revenue                23066 non-null  float64
 16  CTR                    18330 non-null  float64
 17  CPM                    18330 non-null  float64
 18  CPC                    18330 non-null  float64
dtypes: float64(6), int64(7), object(6)
memory usage: 3.3+ MB
```

Shape of the data set:

```
(23066, 19)
```

- There are 23066 rows and 19 rows.

Data set summary:

	count	mean	std	min	25%	50%	75%	max
Ad - Length	23066.0	3.851631e+02	2.336514e+02	120.0000	120.000000	300.00000	7.200000e+02	728.00
Ad- Width	23066.0	3.378960e+02	2.030929e+02	70.0000	250.000000	300.00000	6.000000e+02	600.00
Ad Size	23066.0	9.667447e+04	6.153833e+04	33600.0000	72000.000000	72000.00000	8.400000e+04	216000.00
Available_Impressions	23066.0	2.432044e+06	4.742888e+06	1.0000	33672.250000	483771.00000	2.527712e+06	27592861.00
Matched_Queries	23066.0	1.295099e+06	2.512970e+06	1.0000	18282.500000	258087.50000	1.180700e+06	14702025.00
Impressions	23066.0	1.241520e+06	2.429400e+06	1.0000	7990.500000	225290.00000	1.112428e+06	14194774.00
Clicks	23066.0	1.067852e+04	1.735341e+04	1.0000	710.000000	4425.00000	1.279375e+04	143049.00
Spend	23066.0	2.706626e+03	4.067927e+03	0.0000	85.180000	1425.12500	3.121400e+03	26931.87
Fee	23066.0	3.351231e-01	3.196322e-02	0.2100	0.330000	0.35000	3.500000e-01	0.35
Revenue	23066.0	1.924252e+03	3.105238e+03	0.0000	55.365375	926.33500	2.091338e+03	21276.18
CTR	18330.0	7.366054e-02	7.515992e-02	0.0001	0.002600	0.08255	1.300000e-01	1.00
CPM	18330.0	7.672045e+00	6.481391e+00	0.0000	1.710000	7.66000	1.251000e+01	81.56
CPC	18330.0	3.510606e-01	3.433338e-01	0.0000	0.090000	0.16000	5.700000e-01	7.26

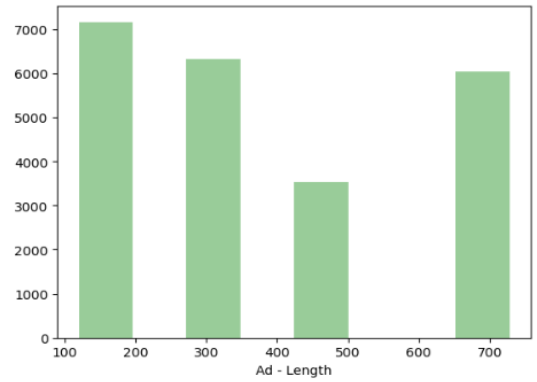
Data set null value check:

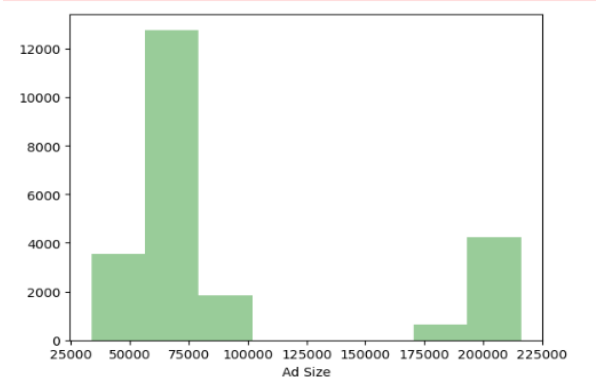
```
Timestamp                  0
InventoryType              0
Ad - Length                0
Ad- Width                  0
Ad Size                    0
Ad Type                    0
Platform                   0
Device Type                0
Format                     0
Available_Impressions      0
Matched_Queries            0
Impressions                0
Clicks                     0
Spend                      0
Fee                        0
Revenue                    0
CTR                     4736
CPM                     4736
CPC                     4736
dtype: int64
```

- There are no duplicate values in the data set.

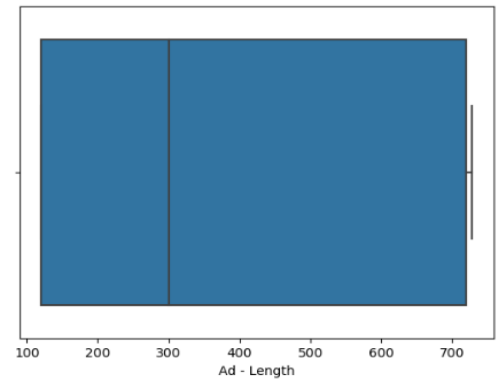
Univariate analysis:

Univariate analysis for few of the variables is given below:

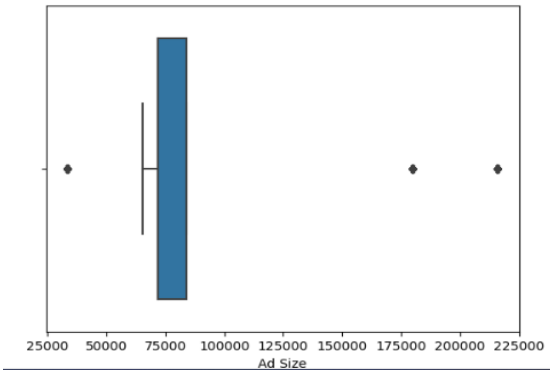


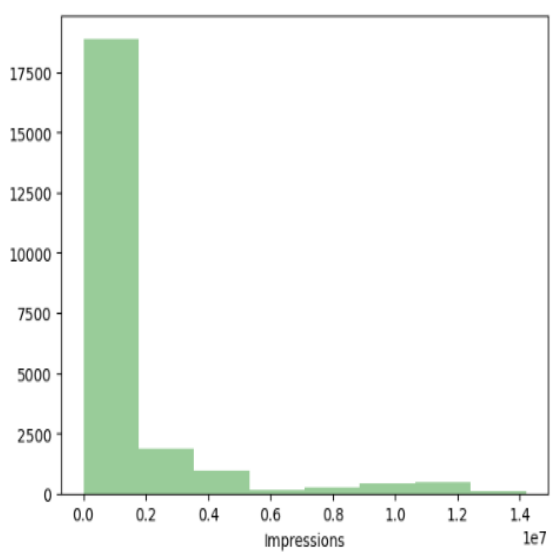


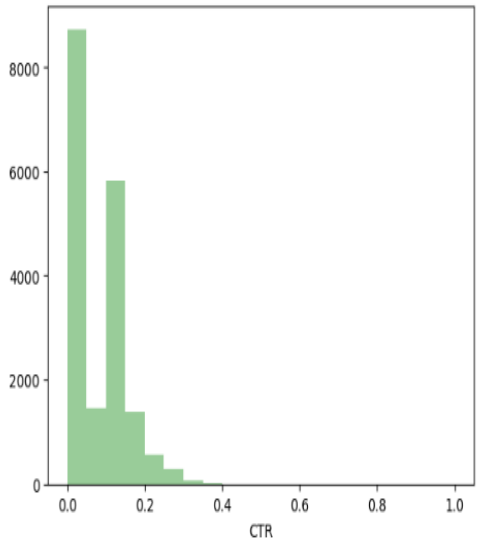
BoxPlot of Ad - Length



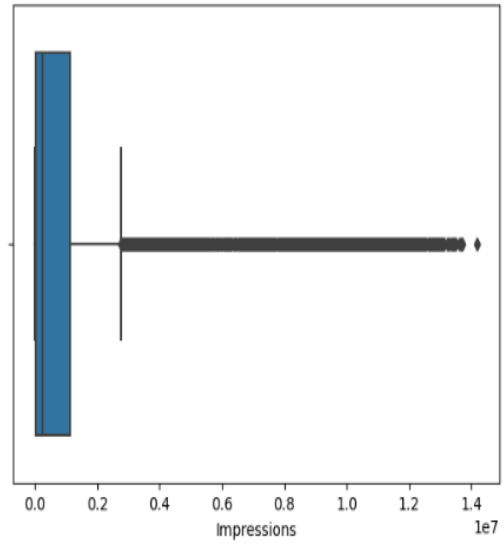
BoxPlot of Ad Size



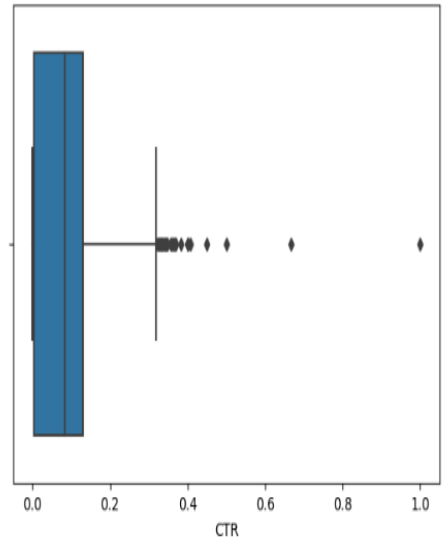




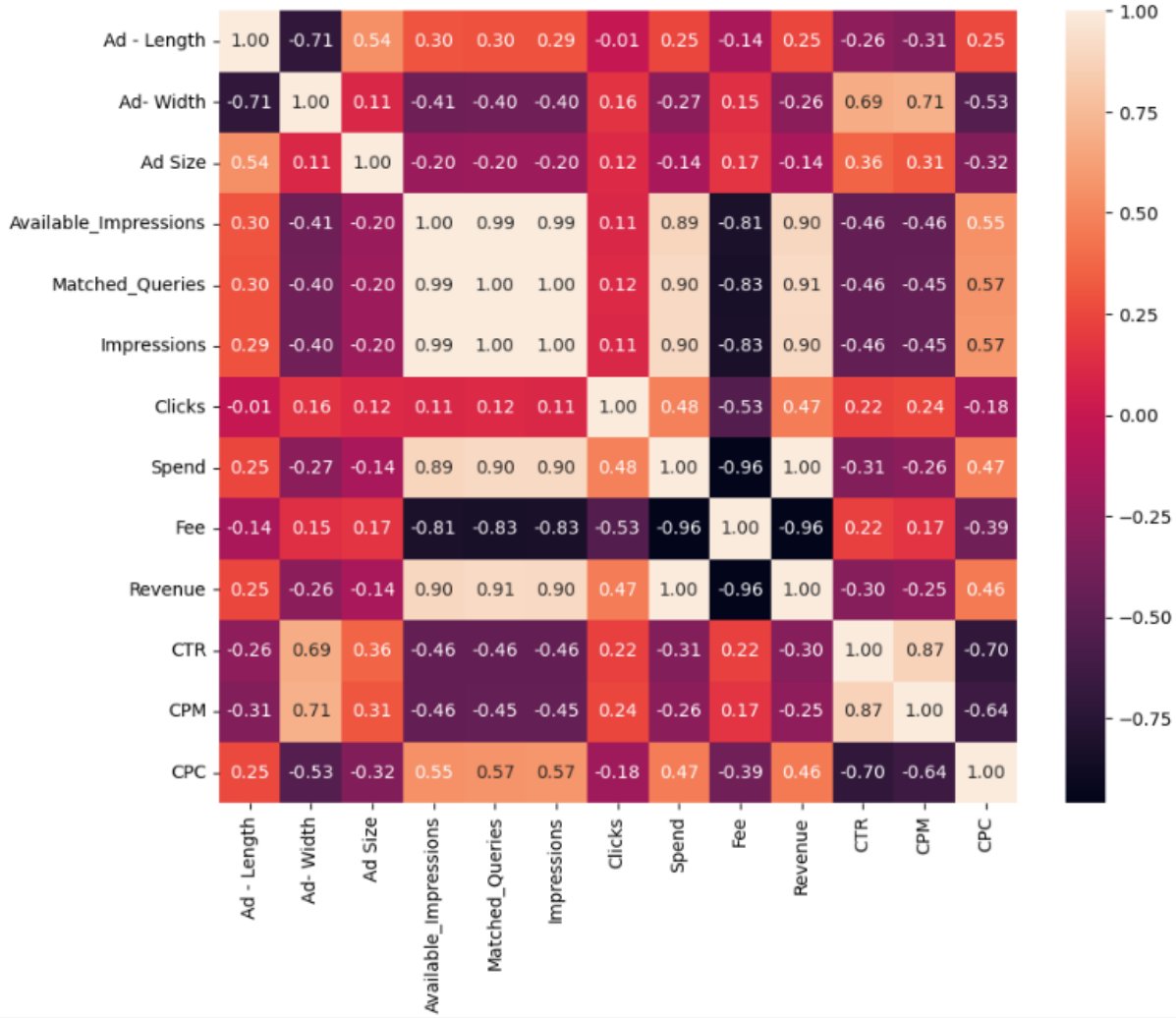
BoxPlot of Impressions



BoxPlot of CTR



Bivariate analysis :

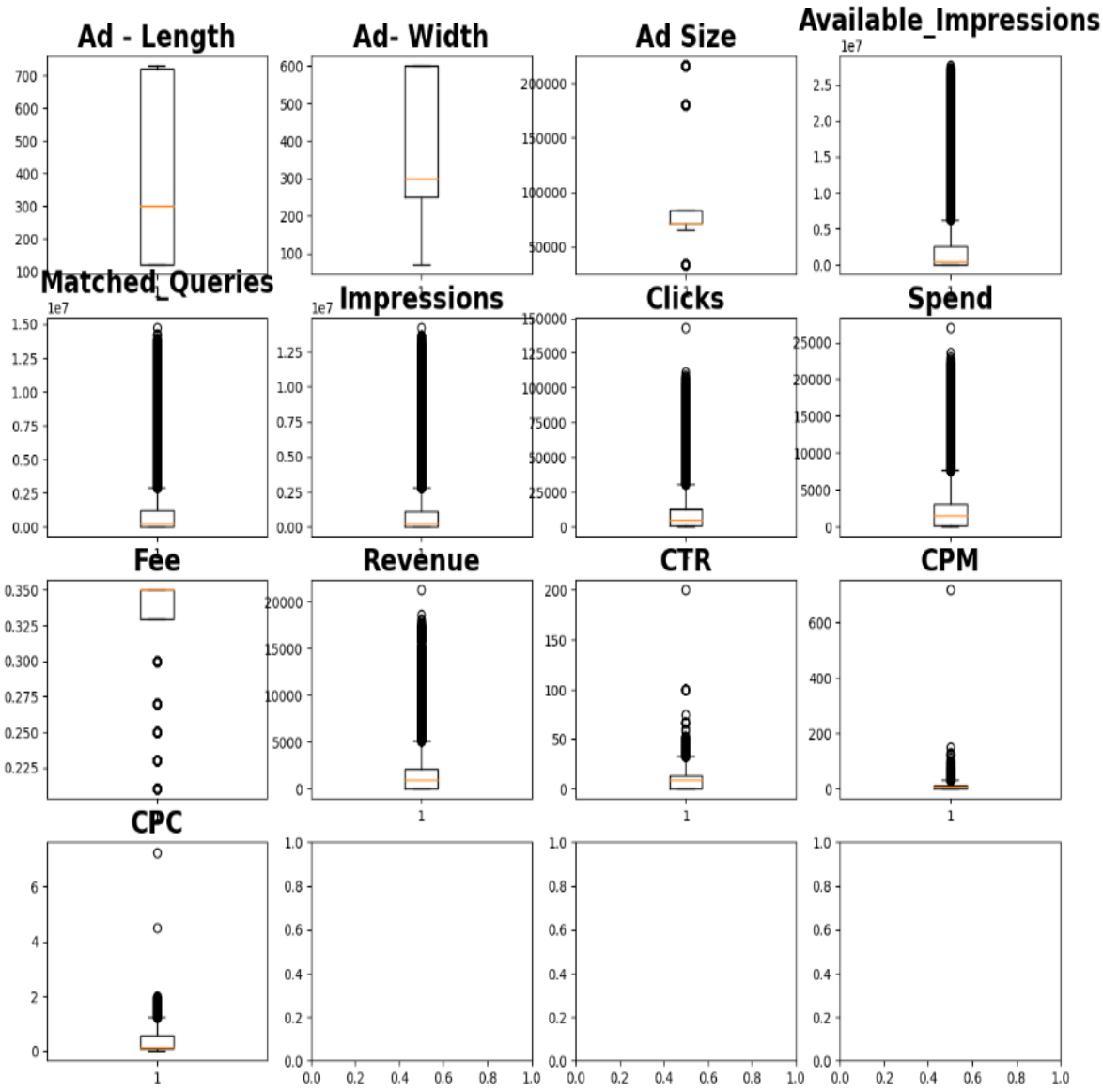


1.2 Treat missing values in CPC, CTR and CPM using the formula given.

```
Timestamp                0
InventoryType            0
Ad - Length              0
Ad- Width                0
Ad Size                  0
Ad Type                  0
Platform                 0
Device Type              0
Format                   0
Available_Impressions    0
Matched_Queries          0
Impressions              0
Clicks                   0
Spend                    0
Fee                      0
Revenue                  0
CTR                      0
CPM                      0
CPC                      0
dtype: int64
```

- We have no missing values now after the treatment.

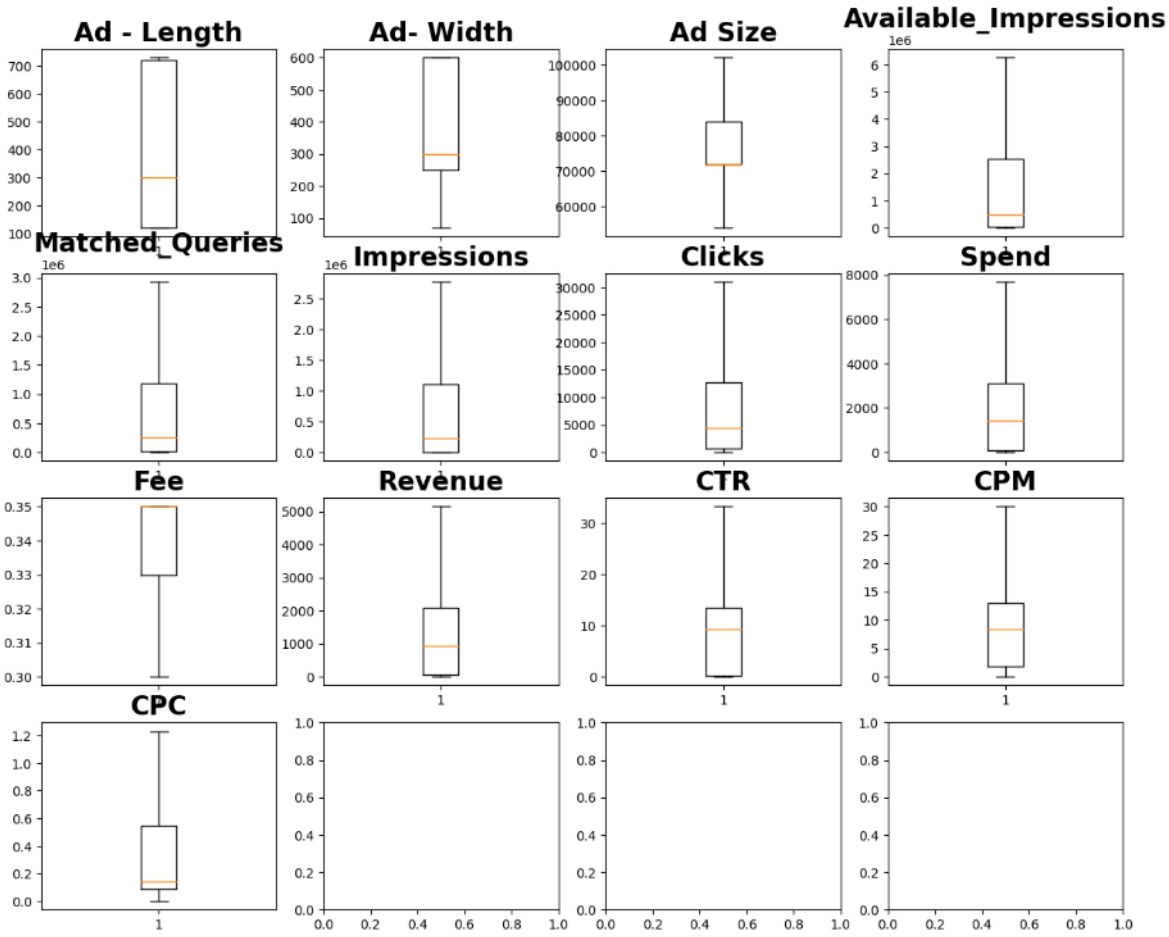
1.3 Check if there are any outliers. Do you think treating outliers is necessary for K-Means clustering? Based on your judgement decide whether to treat outliers and if yes, which method to employ.



Yes, kmeans clustering is sensitive towards outliers.

So, we treat them using IQR method.

In this method, any observation that is less than Q1- 1.5 IQR or more than Q3 + 1.5 IQR is considered an outlier.
To treat outliers, we defined a function 'treat_outlier' where
The larger values (>upper whisker) are all equated to the 95th percentile value of thedistribution
The smaller values (<lower whisker) are all equated to the 5th percentile value of thedistribution

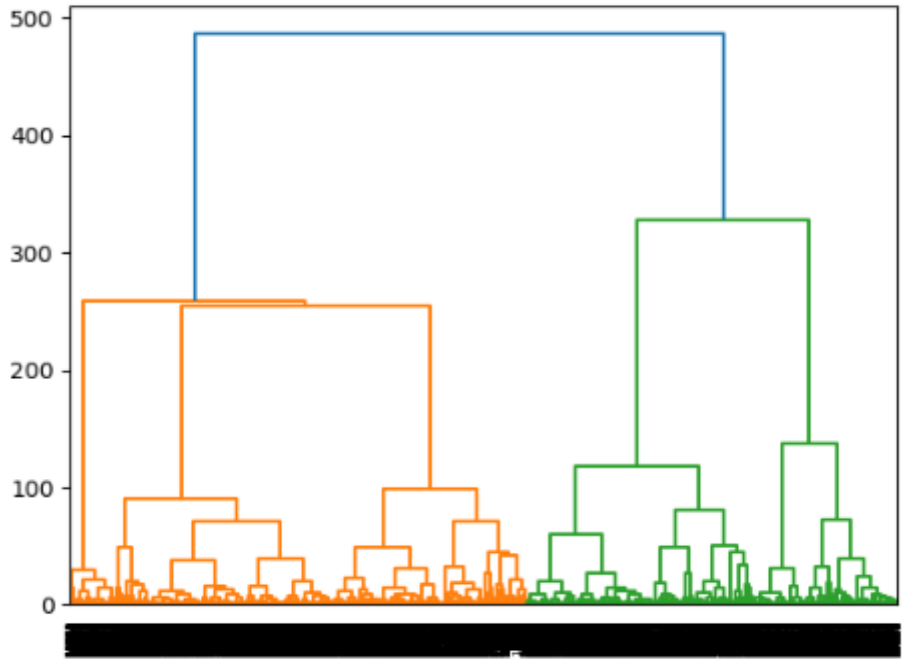


From the above plot, we observe that the outliers are successfully treated and the features are printed.

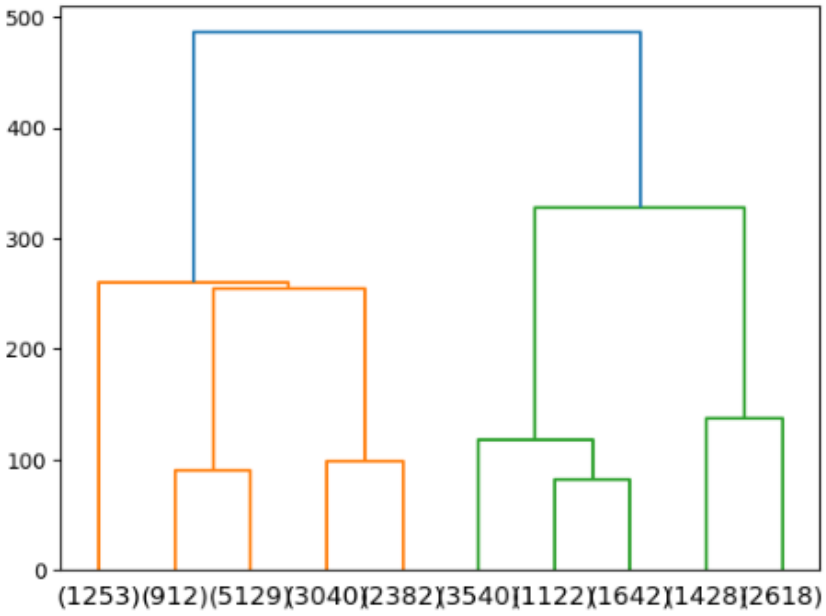
1.4 Perform z-score scaling and discuss how it affects the speed of the algorithm.

	0	1	2	3	4
Ad - Length	-0.364496	-0.364496	-0.364496	-0.364496	-0.364496
Ad- Width	-0.432797	-0.432797	-0.432797	-0.432797	-0.432797
Ad Size	-0.102518	-0.102518	-0.102518	-0.102518	-0.102518
Available_Impressions	-0.755347	-0.755359	-0.754914	-0.755053	-0.755624
Matched_Queries	-0.778958	-0.778997	-0.778928	-0.778791	-0.779039
Impressions	-0.768487	-0.768525	-0.768454	-0.768311	-0.768569
Clicks	-0.866322	-0.866322	-0.866322	-0.866322	-0.866322
Spend	-0.892774	-0.892774	-0.892774	-0.892774	-0.892774
Fee	0.535724	0.535724	0.535724	0.535724	0.535724
Revenue	-0.879713	-0.879713	-0.879713	-0.879713	-0.879713
CTR	-0.999543	-0.994171	-1.003175	-1.013543	-0.986058
CPM	-1.067315	-1.067315	-1.067315	-1.067315	-1.067315
CPC	-0.908581	-0.908581	-0.908581	-0.908581	-0.908581

1.5 Clustering: Perform Hierarchical by constructing a Dendrogram using WARD and Euclideandistance.



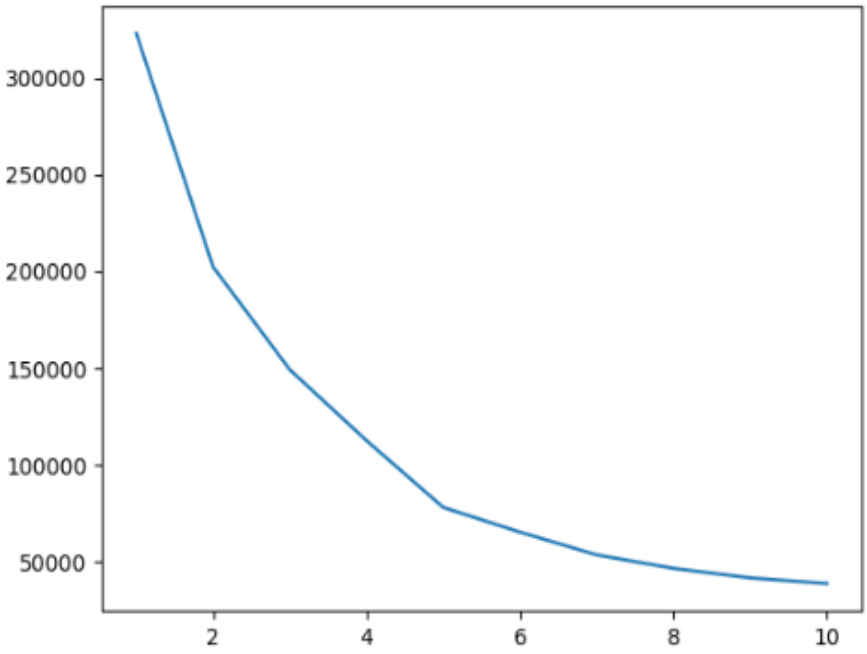
Truncating the dendrogram into 10 clusters.



In a Dendrogram, each branch is called a clade. The terminal end of each clade is called a leaf. The arrangement of the clades tells us which leaves are most similar to each other. The height of the branching points indicates how similar or different they are from each other: the greater the height, the greater the difference.

Keeping the above reference as base, we can see the longest branch (tallest branch) is in blue. If we seethat only blue, it will result in only 2 clusters which is not acceptable in business. If however thesegmentation is at the tallest red branches, separated by the yellow horizontal line, 5 clusters areidentified. Alternatively, there may be 3 clusters as well, designated by the yellow horizontal line. But wechoose 5 Clusters using Dendrogram for this project.

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



The wss value of cluster 1 is `322924.0`,
The wss value of cluster 2 is `202179.4351898643`,
The wss value of cluster 3 is `148996.75114258018`,
The wss value of cluster 4 is `112314.15876227134`,
The wss value of cluster 5 is `77747.23923187575`,
The wss value of cluster 6 is `64973.61874332653`,
The wss value of cluster 7 is `53190.16827930424`,
The wss value of cluster 8 is `46233.055962827115`,
The wss value of cluster 9 is `41317.55350736825`,
The wss value of cluster 10 is `38394.427948061966`

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

```
Average silhouette_score for 2 clusters is 0.39779
Average silhouette_score for 3 clusters is 0.39663
Average silhouette_score for 4 clusters is 0.45847
Average silhouette_score for 5 clusters is 0.53414
Average silhouette_score for 6 clusters is 0.51436
Average silhouette_score for 7 clusters is 0.52355
Average silhouette_score for 8 clusters is 0.48622
Average silhouette_score for 9 clusters is 0.48691
Average silhouette_score for 10 clusters is 0.44516
```

1.8 Profile the ads based on optimum number of clusters using silhouette score and your domain understanding
[Hint: Group the data by clusters and take sum or mean to identify trends in clicks, spend, revenue, CPM, CTR, & CPC based on Device Type. Make bar plots.]

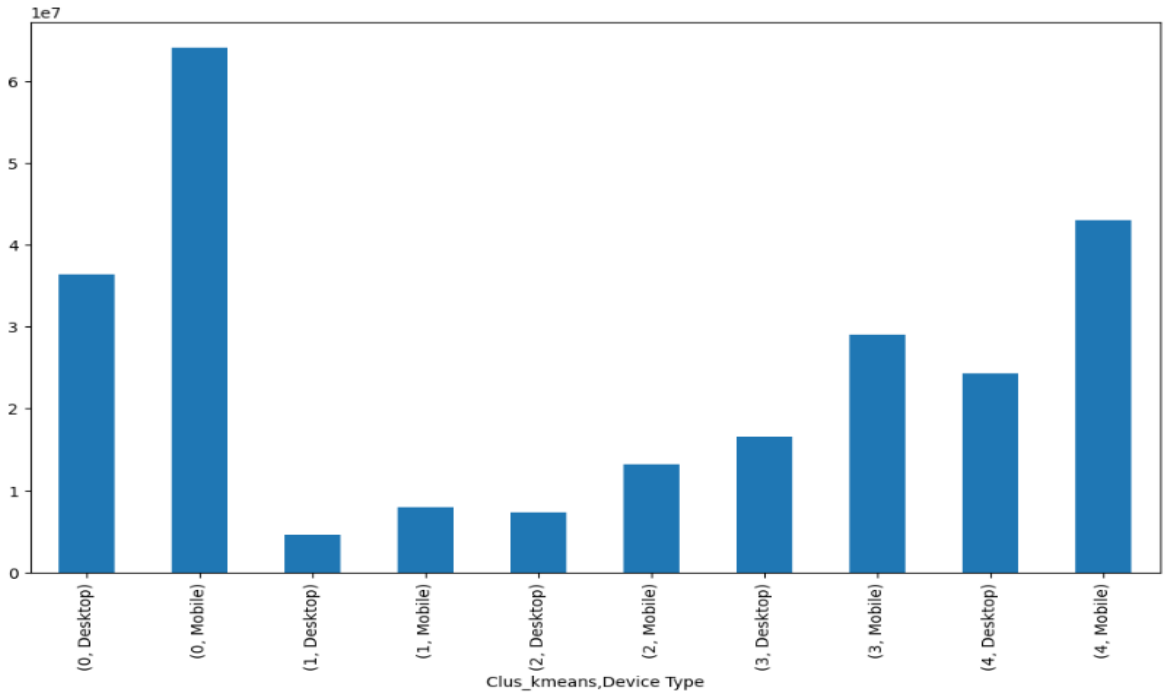
The different clusters present are

```
array([1, 3, 0, 2, 4])
```

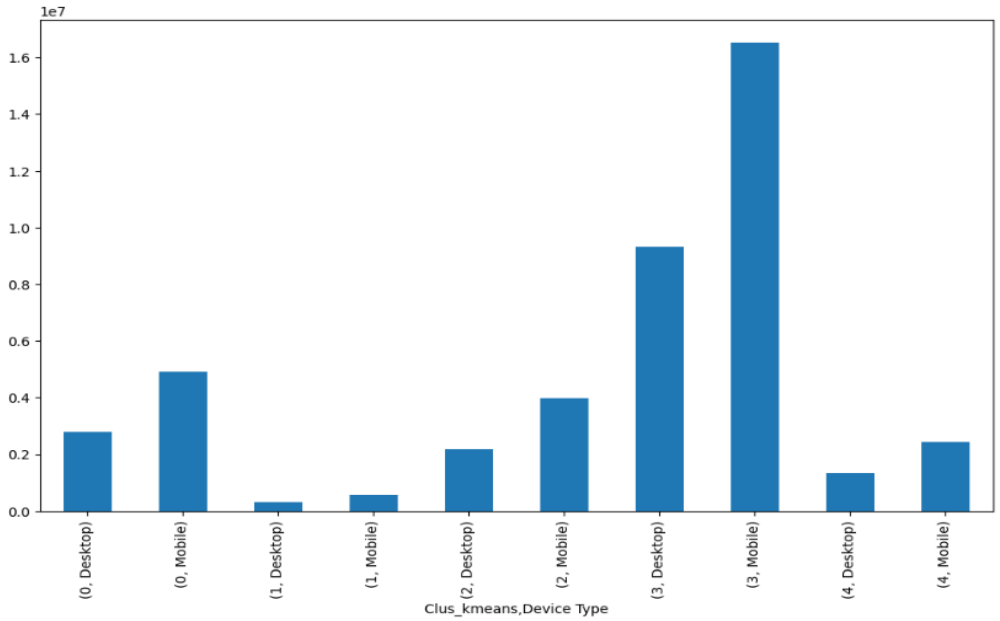
	0	1	2	3	4
Timestamp	2020-9-2-17	2020-9-2-10	2020-9-1-22	2020-9-3-20	2020-9-4-15
InventoryType	Format1	Format1	Format1	Format1	Format1
Ad - Length	300	300	300	300	300
Ad- Width	250	250	250	250	250
Ad Size	75000	75000	75000	75000	75000
Ad Type	Inter222	Inter227	Inter222	Inter228	Inter217
Platform	Video	App	Video	Video	Web
Device Type	Desktop	Mobile	Desktop	Mobile	Desktop
Format	Display	Video	Display	Video	Video
Available_Impressions	1806	1780	2727	2430	1218
Matched_Queries	325	285	356	497	242
Impressions	323	285	355	495	242
Clicks	1	1	1	1	1
Spend	0.0	0.0	0.0	0.0	0.0
Fee	0.35	0.35	0.35	0.35	0.35
Revenue	0.0	0.0	0.0	0.0	0.0
CTR	0.309598	0.350877	0.28169	0.20202	0.413223
CPM	0.0	0.0	0.0	0.0	0.0
CPC	0.0	0.0	0.0	0.0	0.0
Clus_kmeans	1	1	1	1	1
sil_width	0.152046	0.151469	0.152456	0.153516	0.150554

The frequency of the clusters is as follows:

Clus_kmeans	0	1	2	3	4
Ad - Length	120.71	422.55	466.08	631.74	138.86
Ad- Width	600.00	148.66	199.08	312.14	576.29
Ad Size	72426.29	54074.34	75184.26	192851.66	75348.09
Available_Impressions	32365.93	1815118.65	10413336.84	237633.11	818473.93
Matched_Queries	20312.82	867909.95	5639488.80	128166.47	575666.51
Impressions	14046.91	829888.91	5460509.01	108991.71	485592.06
Clicks	2061.01	3260.07	11271.57	12717.77	66329.94
Spend	228.45	1506.24	8666.32	1118.23	7096.74
Fee	0.35	0.35	0.29	0.35	0.29
Revenue	148.76	981.58	6389.03	728.51	5099.19
CTR	16.30	0.40	0.22	13.74	13.76
CPM	14.91	1.79	1.57	12.14	15.35
CPC	0.10	0.55	0.76	0.10	0.11
freq	5827.00	6242.00	4039.00	5467.00	1491.00

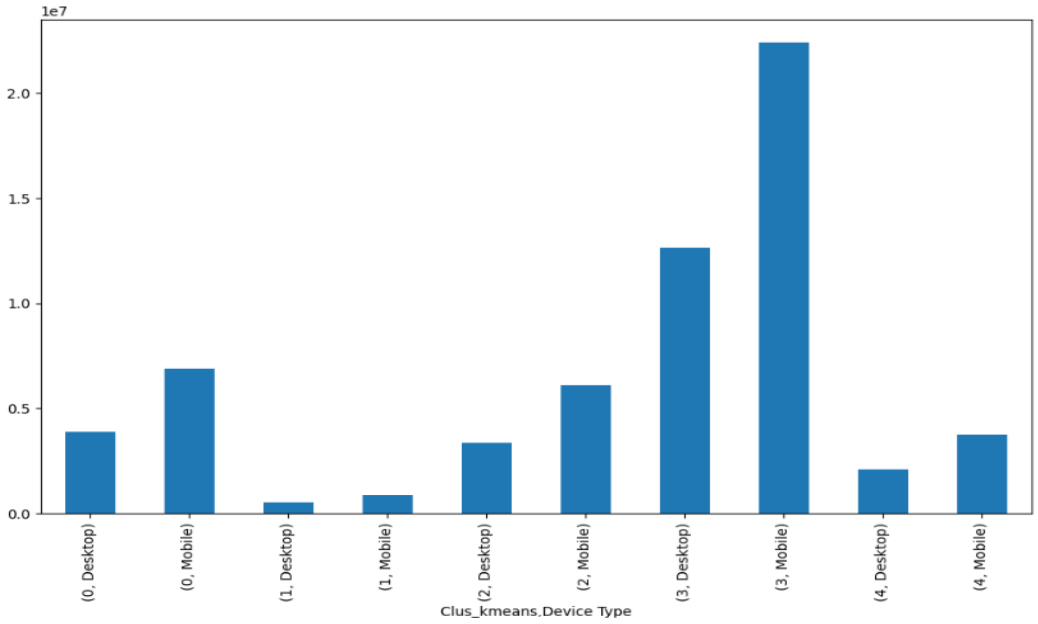


Observation:
The Mobile segment within Cluster 3 has the maximum number of clicks followed by Mobile segment within Cluster 2. Only for Cluster 3, desktop segment shows considerable number of clicks.



Observation:

The mobile segment within Cluster 5 have most revenue generated and may be considered the best ads.Similarly, the desktop segment cluster label has highest revenue generated for Desktop Ads.



Observation :

The mobile segment within cluster 5 show the highest total spending may be considered premium ads.Similarly, the desktop segment cluster label has highest spending done for Desktop Ads.For Mobile segments clusters 3 and 4 show the most spending after cluster label 5 respectively.

Conclusion drawn by the above analysis:

- The dataset has 25857 rows and 19 columns.
- The missing values in CPC, CTR and CPM are treated by using the formulae given and writing auser-defined function, and calling it.
- We check for outliers, we can see there are outliers in the variables.
- Dendogram is the visualization and linkage is for computing the distances and merging theclusters from n to 1.
- The output of Linkage is visualized by Dendogram.
- We will create linkage using Ward's method and run linkage function on the usable columns ofthe data.
- The linkage now stores the various distance at which the n clusters are sequentially merged intoa single cluster.
- using fit – transform function and viewing the output - The dataframe is now stored in an array.
- Using this array we can now perform k-means
- The one requirement before we run the k-means algorithm, is to know how many clusters werequire as output
- We map the elbow plot using wss values
- From the plot we have following observations:
- When we move from k=1 to k=2 , we see that there is a significant drop in the value , also whenwe move from k=2 to k=3,k=3 to k=4 there is a significant drop aswell.
- But from k=4 to k=5 , k=5 to k=6 , the drop in values reduces significantly.
- In otherwords, the wss is not significantly dropping beyond 5,
- So 5 is optimal number of clusters.

Summary of my learnings :

- We learned to impute missing values using a different approach i.e. using custom formulae.
- We discussed about outlier's effect on quality of clustering profiles.
- We discussed about the scaling and its effect on performance of the algorithm.
- We discussed that clusters need to be revisited if there is too much similarity, or overlap, among them.
- We learned about certain digital marketing terms and their significance.

PROBLEM 2

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

top 5 rows of the dataset

	0	1	2	3	4
State Code	1	1	1	1	1
Dist.Code	1	2	3	4	5
State	Jammu & Kashmir	Jammu & Kashmir	Jammu & Kashmir	Jammu & Kashmir	Jammu & Kashmir
Area Name	Kupwara	Badgam	Leh(Ladakh)	Kargil	Punch
No_HH	7707	6218	4452	1320	11654
TOT_M	23388	19585	6546	2784	20591
TOT_F	29796	23102	10964	4206	29981
M_06	5862	4482	1082	563	5157
F_06	6196	3733	1018	677	4587
M_SC	3	7	3	0	20
F_SC	0	6	6	0	33
M_ST	1999	427	5806	2666	7670
F_ST	2598	517	9723	3968	10843
M_LIT	13381	10513	4534	1842	13243
F_LIT	11364	7891	5840	1962	13477
M_ILL	10007	9072	2012	942	7348
F_ILL	18432	15211	5124	2244	16504
TOT_WORK_M	6723	6982	2775	1002	5717
TOT_WORK_F	3752	4200	4800	1118	7692
MAINWORK_M	2763	4628	1940	491	2523
MAINWORK_F	1275	1733	2923	408	2267
MAIN_CL_M	486	1098	519	35	743
MAIN_CL_F	235	357	1205	102	766
MAIN_AL_M	407	442	36	8	254
MAIN_AL_F	143	108	71	24	237
MAIN_HH_M	78	538	19	9	35
MAIN_HH_F	86	343	55	6	64
MAIN_OT_M	1792	2550	1366	439	1491
MAIN_OT_F	811	925	1592	276	1200
MARGWORK_M	3960	2354	835	511	3194
MARGWORK_F	2477	2467	1877	710	5425
MARG_CL_M	619	384	360	135	1327
MARG_CL_F	580	661	1250	286	2462
MARG_AL_M	2052	915	44	63	1037
MARG_AL_F	641	547	157	176	1069

MARG_HH_M	142	369	15	10	62
MARG_HH_F	244	627	32	43	319
MARG_OT_M	1147	686	416	303	768
MARG_OT_F	1012	632	438	205	1575
MARGWORK_3_6_M	16665	12603	3771	1782	14874
MARGWORK_3_6_F	26044	18902	6164	3088	22289
MARG_CL_3_6_M	2810	1829	721	317	2320
MARG_CL_3_6_F	1728	1752	1689	463	3497
MARG_AL_3_6_M	439	261	316	74	862
MARG_AL_3_6_F	343	432	1161	158	1419
MARG_HH_3_6_M	1372	729	41	50	832
MARG_HH_3_6_F	389	399	123	126	767
MARG_OT_3_6_M	110	293	15	6	38
MARG_OT_3_6_F	198	449	28	33	214
MARGWORK_0_3_M	889	546	349	187	588
MARGWORK_0_3_F	798	472	377	146	1097
MARG_CL_0_3_M	1150	525	114	194	874
MARG_CL_0_3_F	749	715	188	247	1928
MARG_AL_0_3_M	180	123	44	61	465
MARG_AL_0_3_F	237	229	89	128	1043
MARG_HH_0_3_M	680	186	3	13	205
MARG_HH_0_3_F	252	148	34	50	302
MARG_OT_0_3_M	32	76	0	4	24
MARG_OT_0_3_F	46	178	4	10	105
NON_WORK_M	258	140	67	116	180
NON_WORK_F	214	160	61	59	478

Checking the shape of the data:

```
(640, 61)
```

Info of the dataset:

```
<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
```

Data set summary:

	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	1278.642345	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	6019.806644	27.0	2372.00	4630.0	8167.00	39106.0

MARG_CL_3_6_F	640.0	10339.864063	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	3610.50	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

Data set null value check:

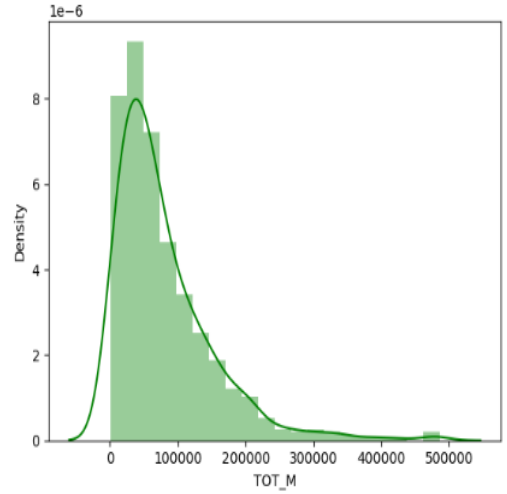
```
State Code          0
Dist.Code           0
State               0
Area Name           0
No_HH               0
TOT_M               0
TOT_F               0
M_06                0
F_06                0
M_SC                0
F_SC                0
M_ST                0
F_ST                0
M_LIT               0
F_LIT               0
M_ILL               0
F_ILL               0
TOT_WORK_M          0
TOT_WORK_F          0
MAINWORK_M          0
MAINWORK_F          0
MAIN_CL_M           0
MAIN_CL_F           0
MAIN_AL_M           0
MAIN_AL_F           0
MAIN_HH_M           0
MAIN_HH_F           0
MAIN_OT_M           0
MAIN_OT_F           0
MARGWORK_M          0
MARGWORK_F          0
MARG_CL_M           0
MARG_CL_F           0
MARG_AL_M           0
MARG_AL_F           0
MARG_HH_M           0
MARG_HH_F           0
MARG_OT_M           0
MARG_OT_F           0
MARGWORK_3_6_M      0
MARGWORK_3_6_F      0
MARG_CL_3_6_M       0
MARG_CL_3_6_F       0
MARG_AL_3_6_M       0
MARG_AL_3_6_F       0
MARG_HH_3_6_M       0
MARG_HH_3_6_F       0
MARG_OT_3_6_M       0
MARG_OT_3_6_F       0
MARGWORK_0_3_M      0
MARGWORK_0_3_F      0
MARG_CL_0_3_M       0
MARG_CL_0_3_F       0
MARG_AL_0_3_M       0
MARG_AL_0_3_F       0
MARG_HH_0_3_M       0
MARG_HH_0_3_F       0
MARG_OT_0_3_M       0
MARG_OT_0_3_F       0
NON_WORK_M          0
NON_WORK_F          0
dtype: int64
```

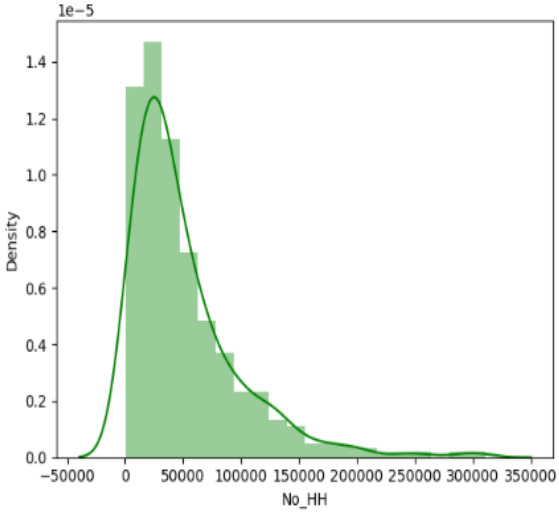
- Total duplicate values is equal to 0.

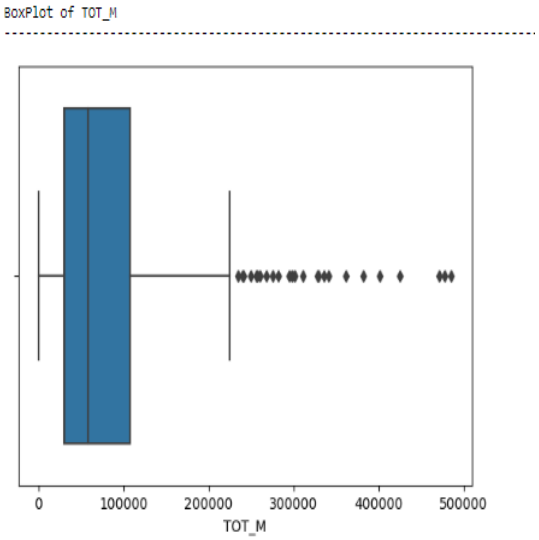
2.2 Pick 5 variables out of the given 24 variables below for EDA: No_HH, TOT_M, TOT_F, M_06, F_06, M_SC, F_SC, M_ST, F_ST, M_LIT, F_LIT, M_ILL, F_ILL, TOT_WORK_M, TOT_WORK_F, MAINWORK_M, MAINWORK_F, MAIN_CL_M, MAIN_CL_F, MAIN_AL_M, MAIN_AL_F, MAIN_HH_M, MAIN_HH_F, MAIN_OT_M, MAIN_OT_F 2. Example questions to answer from EDA - (i) Which state has highest gender ratio and which has the lowest? (ii) Which district has the highest & lowest gender ratio?

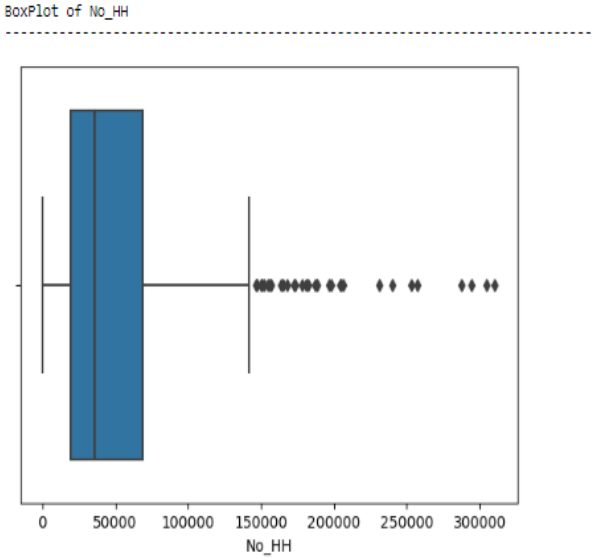
Sol :

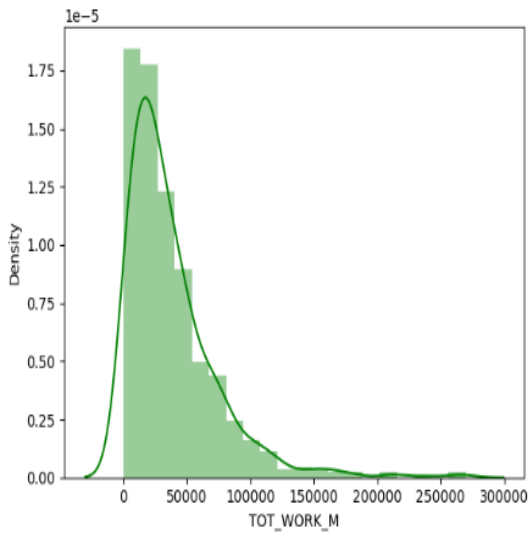
Choose these 5 variables for EDA: 'No_HH', 'TOT_M', 'TOT_F', 'TOT_WORK_M', 'TOT_WORK_F'.

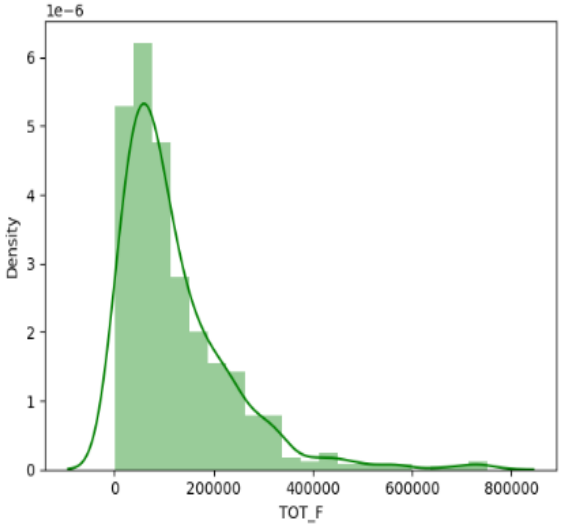




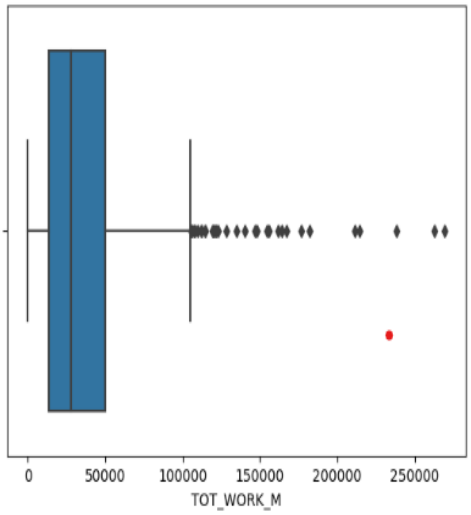




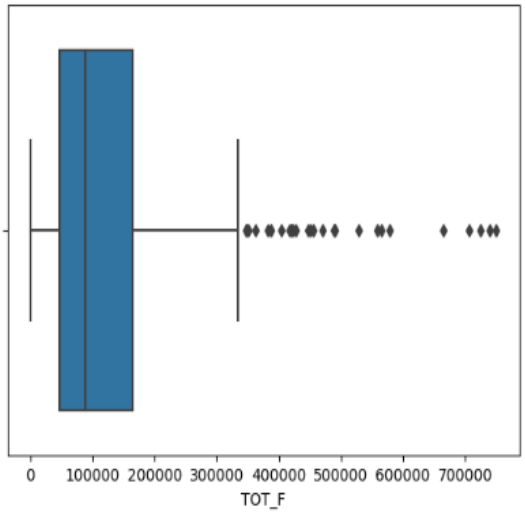


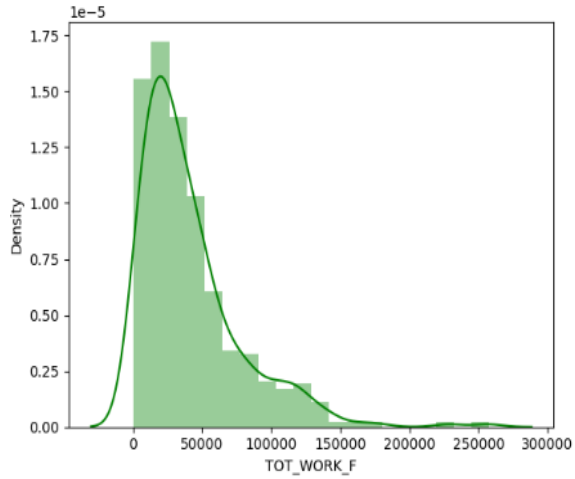


BoxPlot of TOT_WORK_M

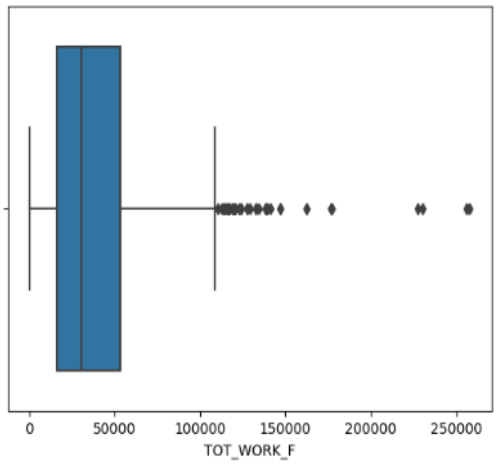


BoxPlot of TOT_F



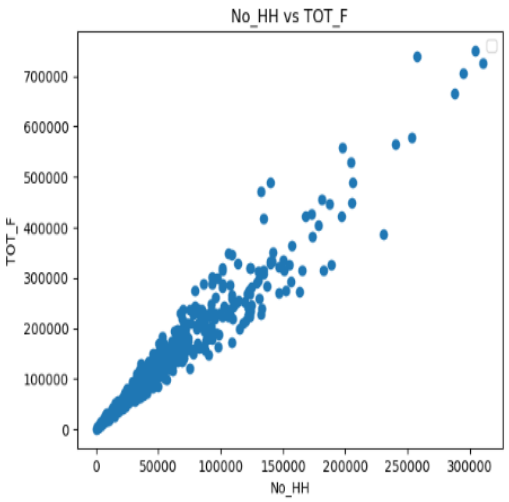


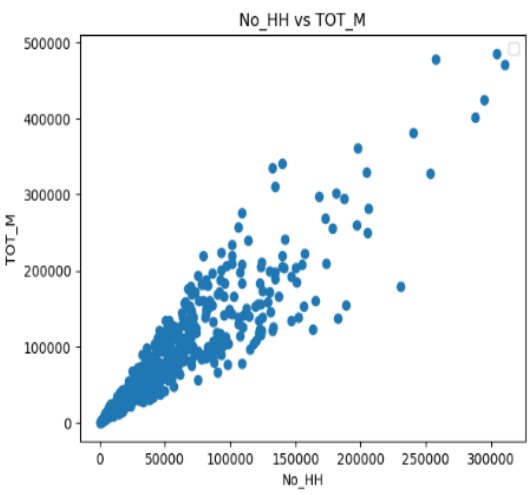
BoxPlot of TOT_WORK_F

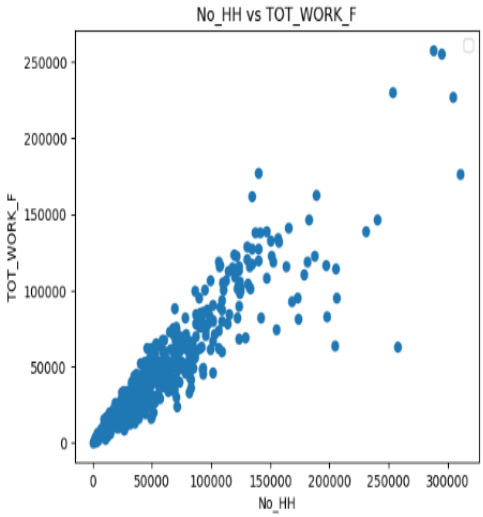


- From the Univariate Analysis we can say all variables are Left Skewed here and all are having Outliers

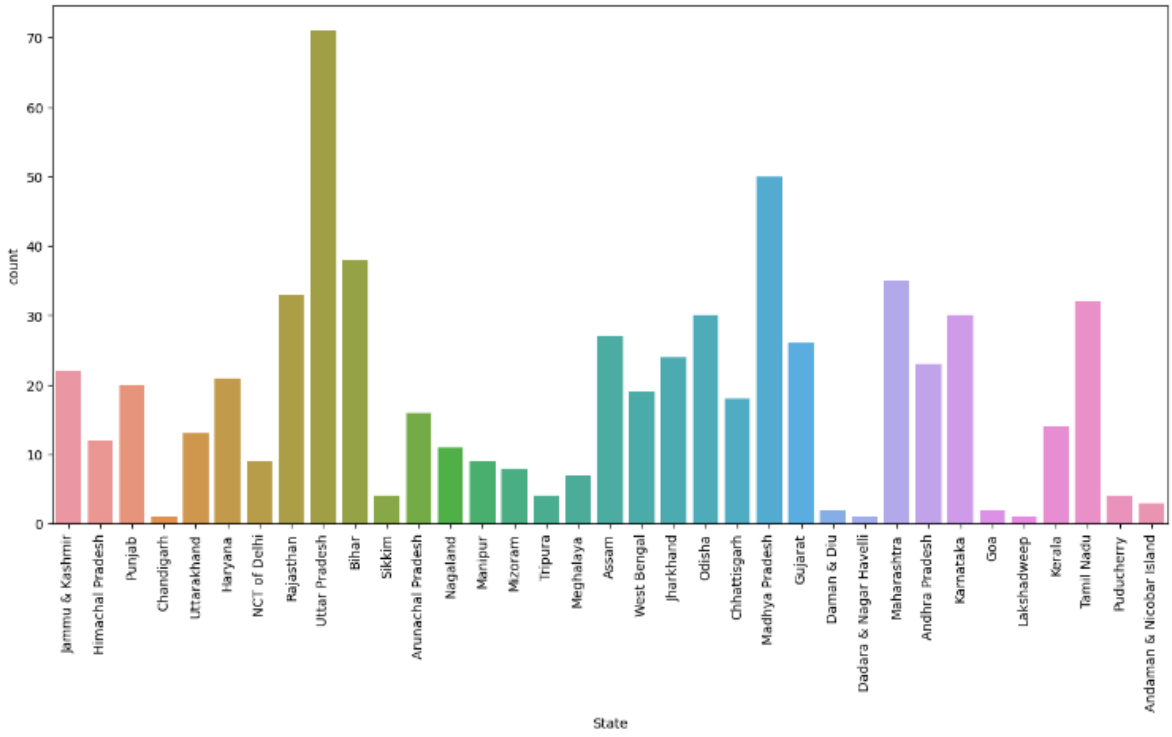
Bivariate analysis:



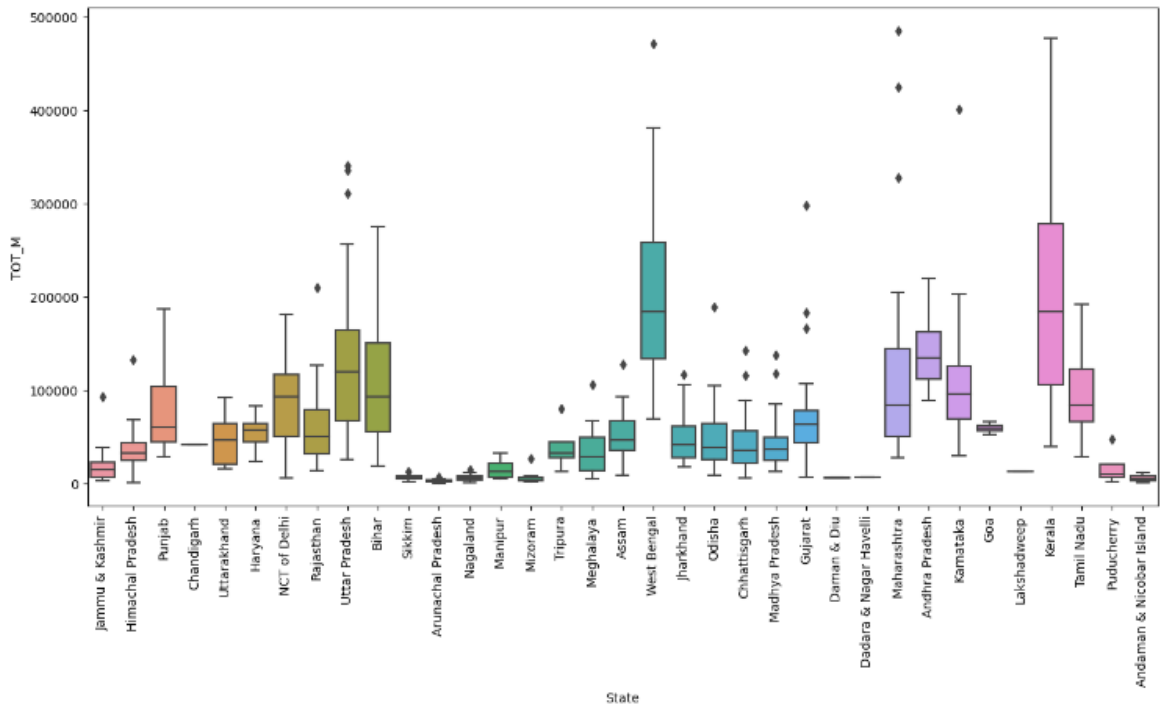


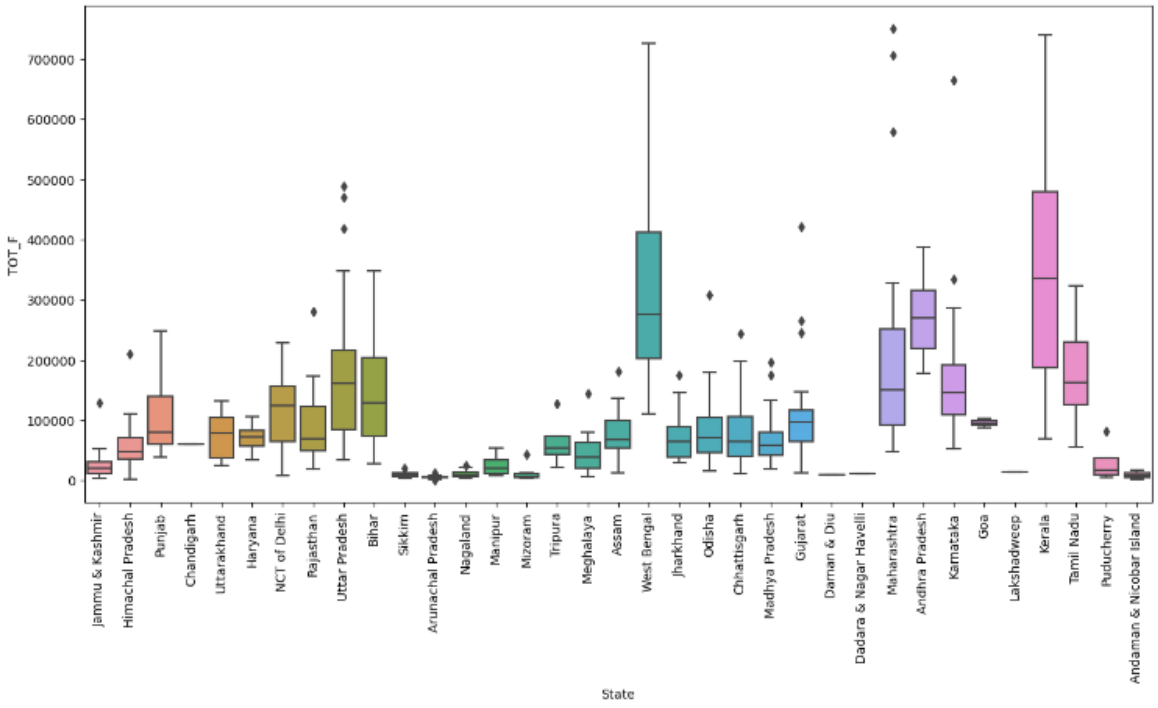


From the Bivariate Analysis we can say all variables are Positively Co-related to each other.



From the above plot, we conclude that Uttarpradesh has the highest population. And Lakshadweep , chandigarh, Dadara and nagar havelli has the lowest population.


TOT_M
500000
400000
300000
200000
100000
0
Jammu & Kashmir
Himachal Pradesh
Punjab
Chandigarh
Uttarakhand
Haryana
NCT of Delhi
Rajasthan
Uttar Pradesh
Bihar
Sikkim
Arunachal Pradesh
Nagaland
Manipur
Mizoram
Tripura
Meghalaya
Assam
West Bengal
Jharkhand
Odisha
Chhattisgarh
Madhya Pradesh
Gujarat
Daman & Diu
Dadara & Nagar Havelli
Maharashtra
Andhra Pradesh
Karnataka
Goa
Lakshadweep
Kerala
Tamil Nadu
Puducherry
Andaman & Nicobar Island
State


TOT_F
700000
600000
500000
400000
300000
200000
100000
0
Jammu & Kashmir
Himachal Pradesh
Punjab
Chandigarh
Uttarakhand
Haryana
NCT of Delhi
Rajasthan
Uttar Pradesh
Bihar
Sikkim
Arunachal Pradesh
Nagaland
Manipur
Mizoram
Tripura
Meghalaya
Assam
West Bengal
Jharkhand
Odisha
Chhattisgarh
Madhya Pradesh
Gujarat
Daman & Diu
Dadara & Nagar Havelli
Maharashtra
Andhra Pradesh
Karnataka
Goa
Lakshadweep
Kerala
Tamil Nadu
Puducherry
Andaman & Nicobar Island
State

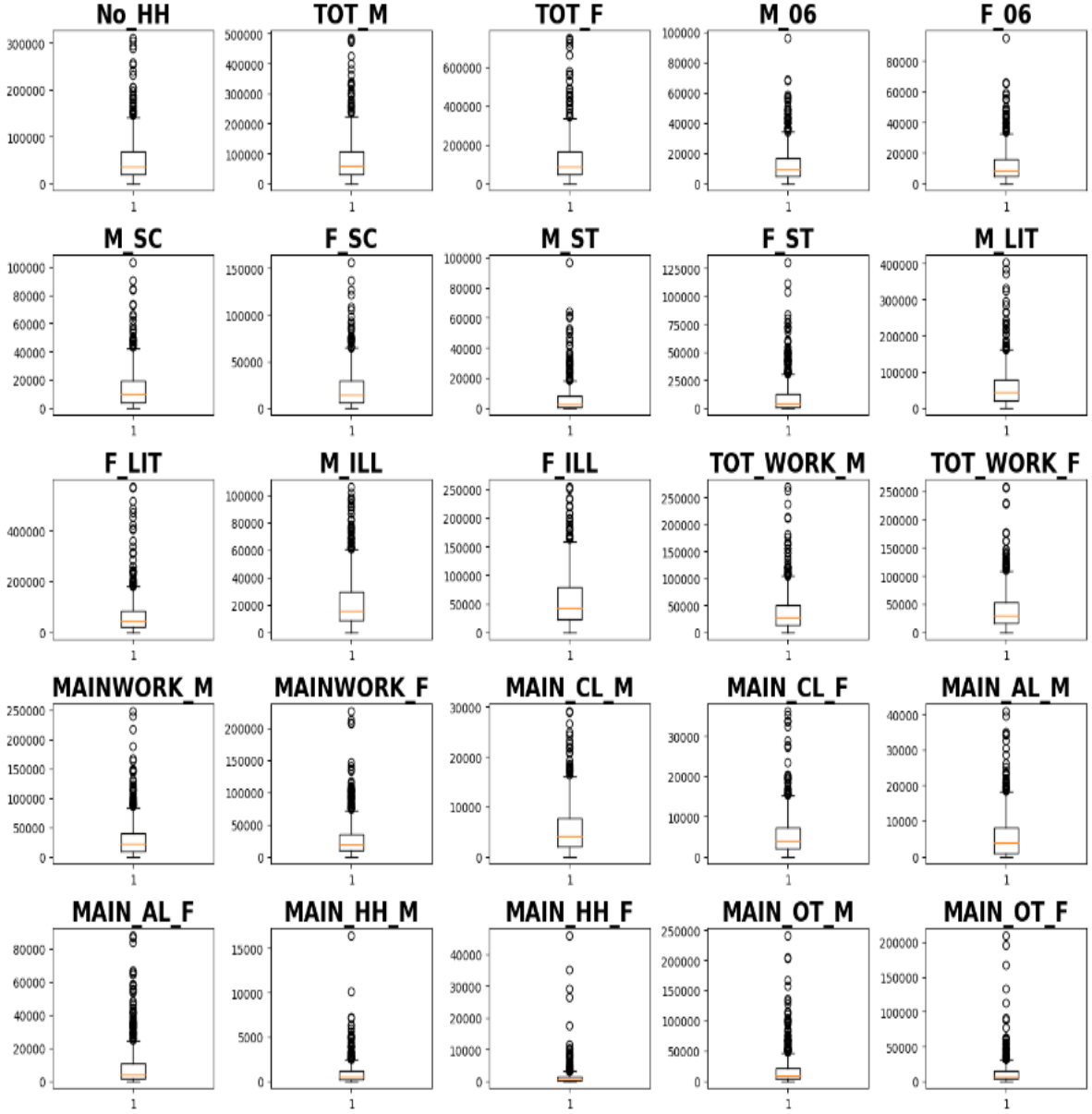
Check for and treat (if needed) missing values

We choose not to treat outliers for this case. Outliers treatment is not necessary unless the they are the result from a processing mistake or wrong measurement. True outliers must be kept in the data.

Scale the Data using the z-score method Visualize the data before and after scaling and comment on the impact on outliers

We drop the features state code and district code as they make no sense while dealing with the data.

Presence of outliers before scaling:

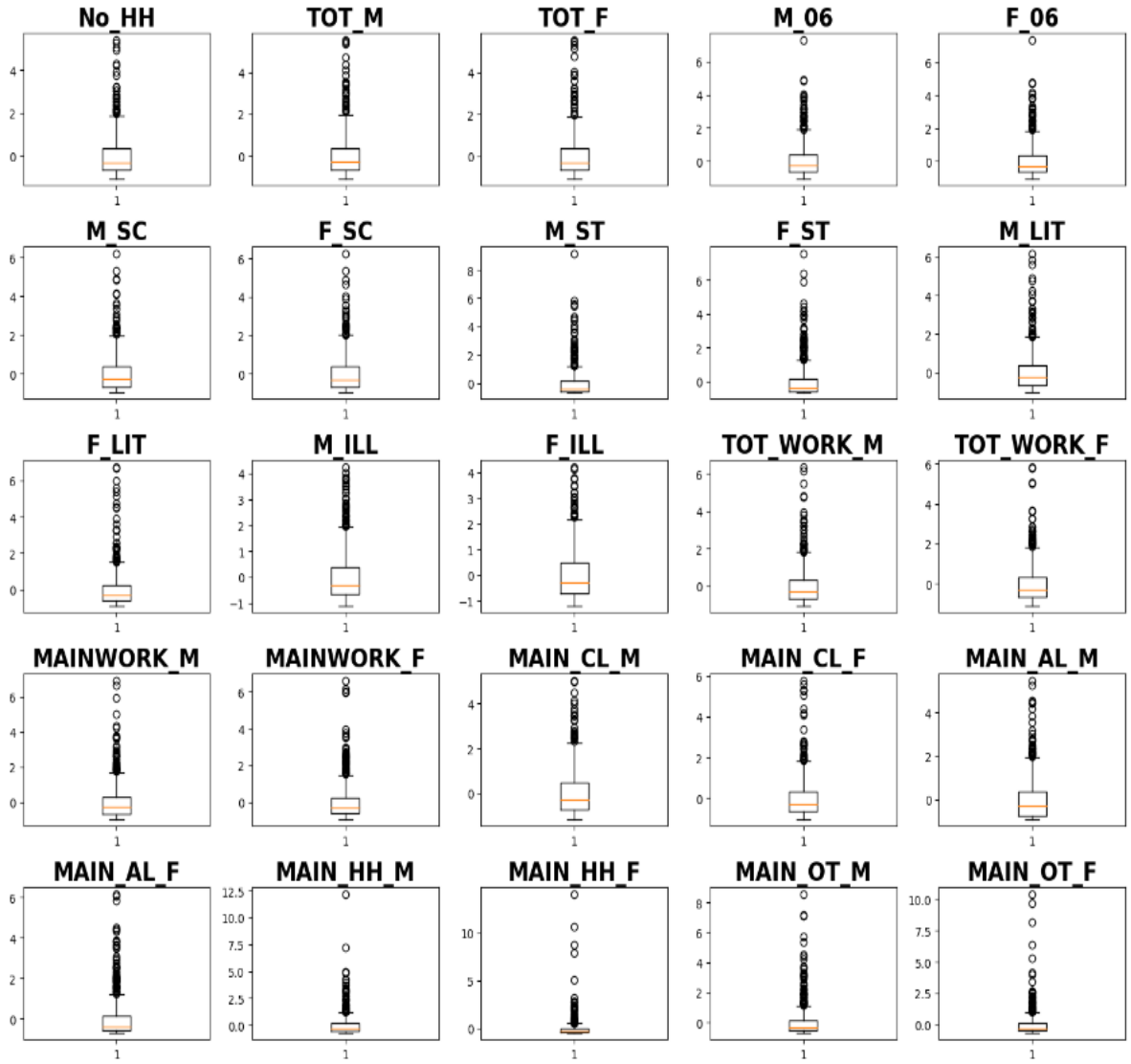


Scaled data frame after applying z score:

	State Code	Dist.Code	No_HH	TOT_M	TOT_F	M_06	F_06	M_SC	F_SC	M_ST	...	MARG_CL_0_3_M	MARG_CL_0_3_F	MARG_A
0	-1.710782	-1.729347	-0.904738	-0.771236	-0.815563	-0.561012	-0.507738	-0.958575	-0.957049	-0.423306	...	-0.163229	-0.720610	-
1	-1.710782	-1.723934	-0.935695	-0.823100	-0.874534	-0.681096	-0.725367	-0.958297	-0.956772	-0.582014	...	-0.583103	-0.732811	-
2	-1.710782	-1.718521	-0.972412	-1.000919	-0.981466	-0.976956	-0.965262	-0.958575	-0.956772	-0.038951	...	-0.859212	-0.921931	-
3	-1.710782	-1.713109	-1.037530	-1.052224	-1.041001	-1.022118	-0.995393	-0.958783	-0.957049	-0.355965	...	-0.805468	-0.900758	-
4	-1.710782	-1.707696	-0.822676	-0.809381	-0.813933	-0.622359	-0.649908	-0.957395	-0.955529	0.149238	...	-0.348645	-0.297513	

5 rows x 59 columns

Presence of outliers after z score method:



We can clearly see that scaling has no impact on outliers.

Create the covariance matrix - Get eigen values and eigen vectors - Identify the optimum number of PCs - Show Scree plot - Compare PCs with Actual Columns and identify which is explaining most variance - Write inferences about all the PCs in terms of actual variables - Write linear equation for first PC Note: For the scope of this project, take at least 90% explained variance.

Covariance matrix

```
array([[-4.62, -4.77, -5.96, ..., -6.29, -6.22, -5.9 ],
       [ 0.14, -0.11, -0.29, ..., -0.64, -0.67, -0.94],
       [ 0.33,  0.24,  0.37, ...,  0.11,  0.27,  0.35],
       ...,
       [ 0.  ,  0.  , -0.  , ..., -0.  , -0.  ,  0.  ],
       [ 0.  , -0.  ,  0.  , ..., -0.  ,  0.  , -0.  ],
       [ 0.  , -0.  ,  0.  , ...,  0.  , -0.  ,  0.  ]])
```

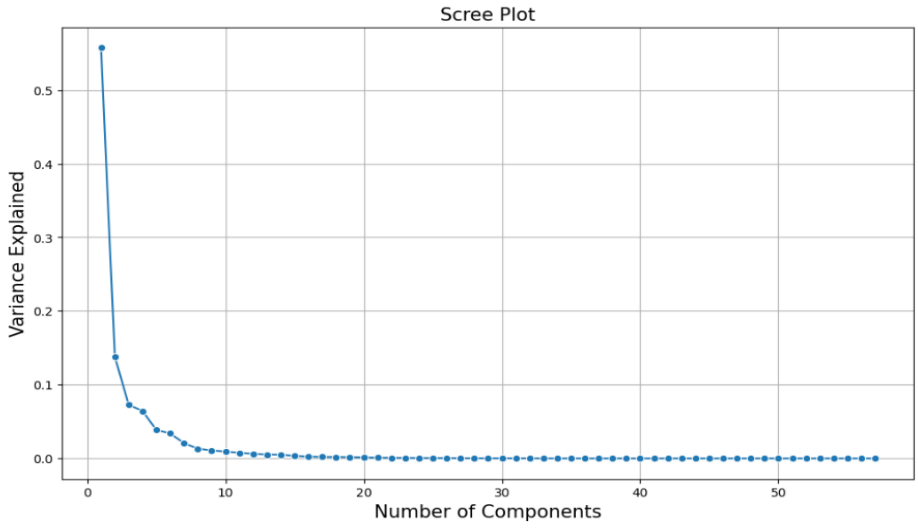
Eigen vectors:

```
Eigen Vectors
 %s [[ 0.16  0.17  0.17 ...  0.13  0.15  0.13]
 [-0.13 -0.09 -0.1  ...  0.05 -0.07 -0.07]
 [-0.    0.06  0.04 ... -0.08  0.11  0.1 ]
 ...
 [-0.   -0.17 -0.1  ...  0.01 -0.04 -0.02]
 [-0.   -0.01 -0.08 ...  0.05 -0.17  0.04]
 [ 0.    0.18  0.04 ... -0.   -0.06 -0.04]]
```

Eigen values:

```
[0.56 0.14 0.07 0.06 0.04 0.03 0.02 0.01 0.01 0.01 0.01 0.01 0.01 0.
 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.  ]
```

Scree plot is as follows:

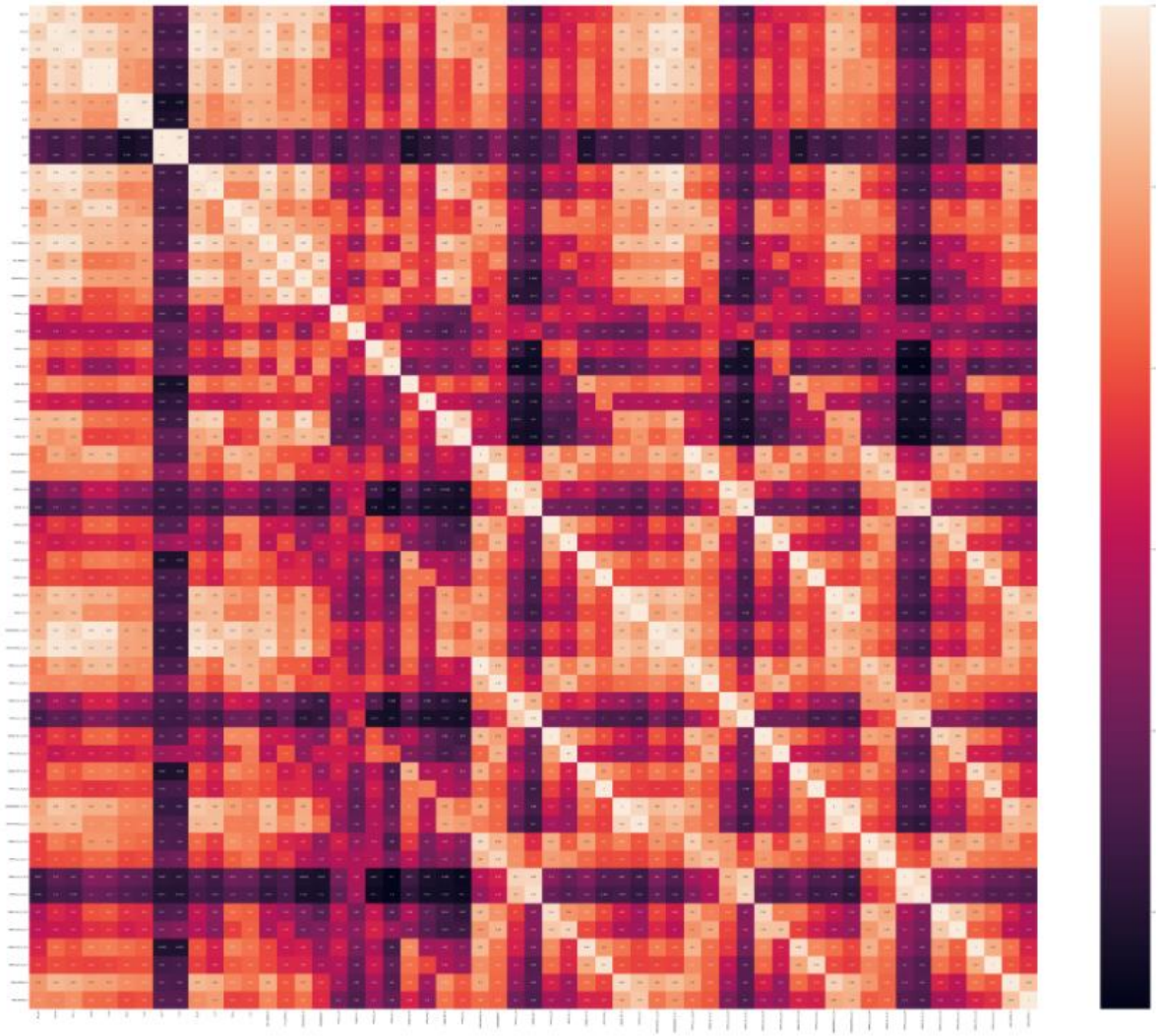


Cumulative explained variance ratio to find a cut off for selecting the number of PCs:

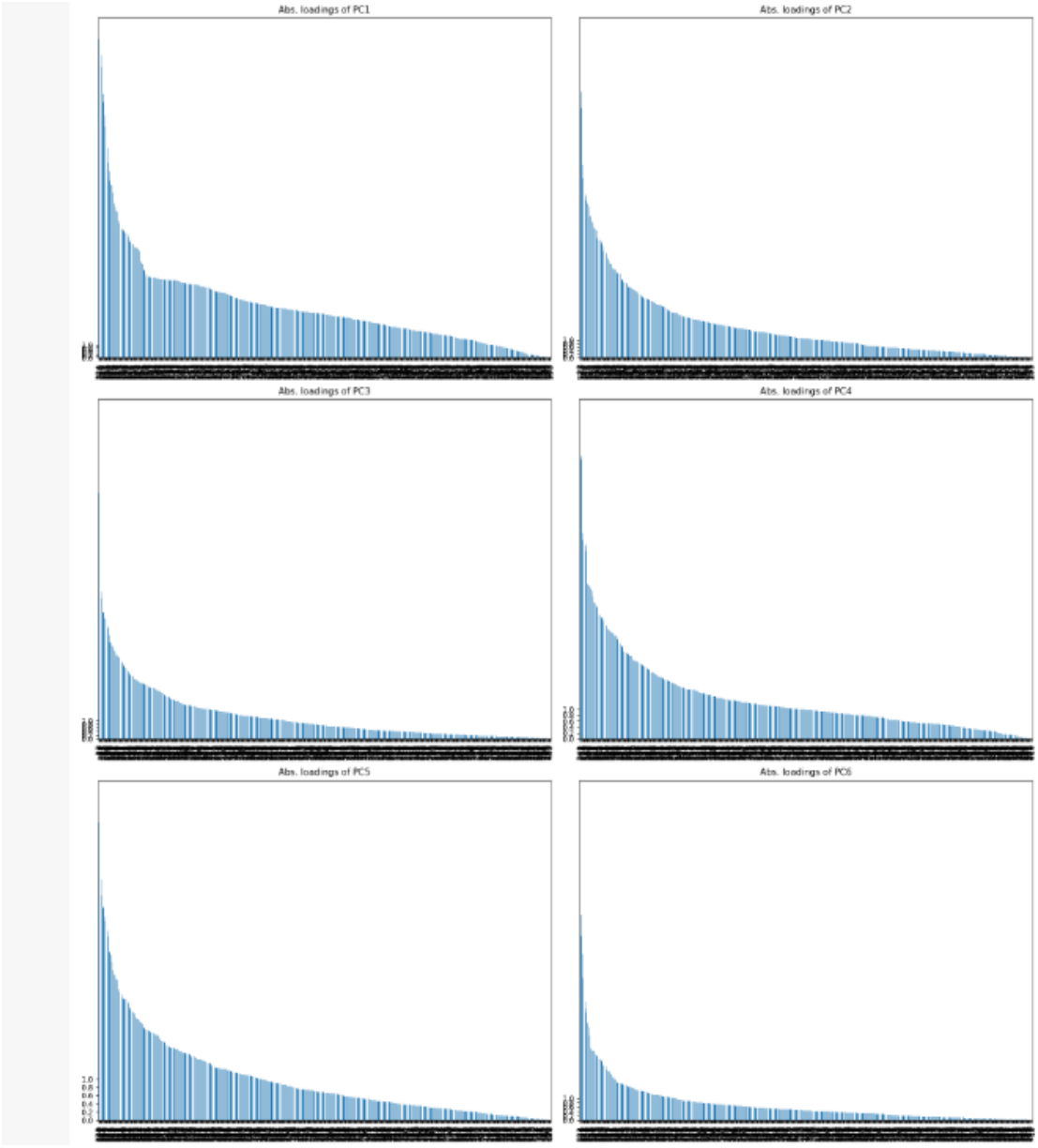
```
Cumulative Variance Explained in Percentage: [ 55.73  69.51  76.79  83.21  87.08  90.47  92.53  93.85  94.93  95.85
  96.61  97.23  97.75  98.24  98.57  98.81  99.01  99.2   99.37  99.51
  99.61  99.69  99.75  99.81  99.85  99.89  99.92  99.94  99.96  99.97
  99.98  99.99 100.   100.   100.   100.   100.   100.   100.   100.
 100.   100.   100.   100.   100.   100.   100.   100.   100.   100.
 100.   100.   100.   100.   100.   100.   100.  ]
```

For this project, we need to consider at least 90% explained variance, so cut off for selecting thenumber of PCs is: '6'

Heatmap:



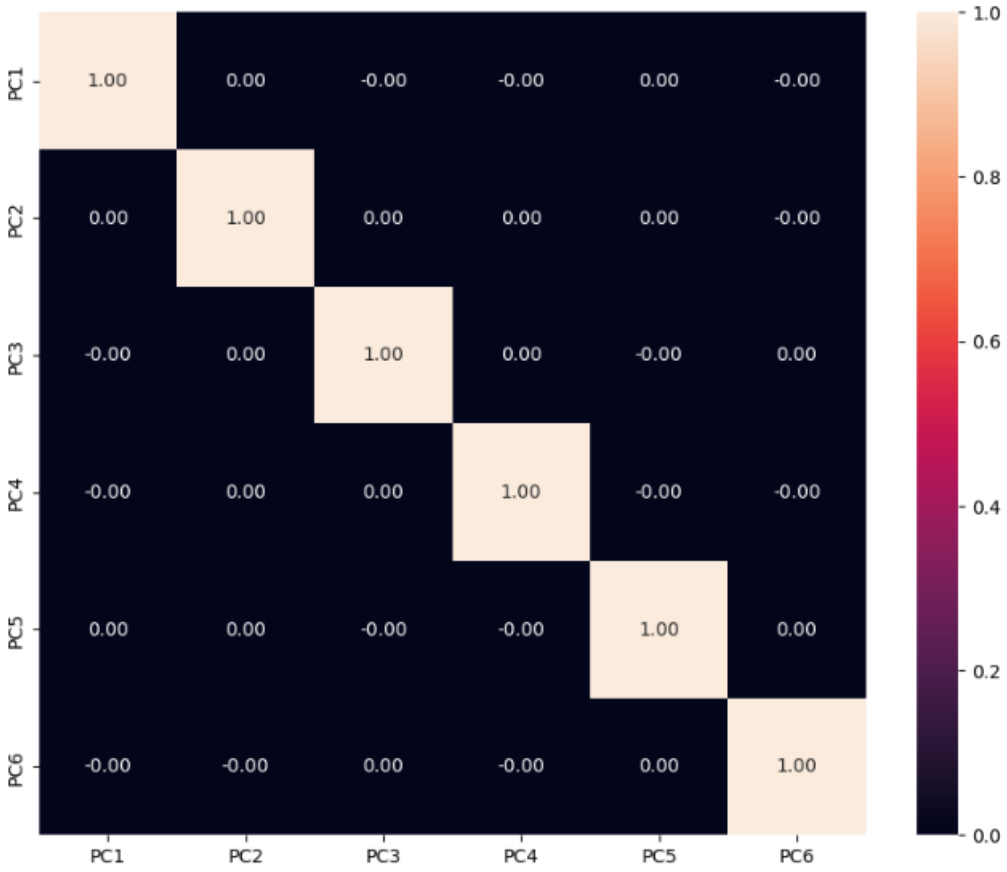
Graph showing how origianl features influences PC'S



Extracting the number of PC's we require:

	PC1	PC2	PC3	PC4	PC5	PC6
0	-4.62	0.14	0.33	1.54	0.35	-0.42
1	-4.77	-0.11	0.24	1.96	-0.15	0.42
2	-5.96	-0.29	0.37	0.62	0.48	0.28
3	-6.28	-0.50	0.21	1.07	0.30	0.05
4	-4.48	0.89	1.08	0.54	0.80	0.34

Check for the correlations among the pc's:



Linear equation for the first PC:

PC1= a1x1 + a2x2 + a3x3 + a4x4 ……...+ a57x57

Where a1 , a2 …... are the coefficients extracted through the process

And x1 , x2 , x3 …… are the observed data.