ML-1 GRADED PROJECT - CODED

DSBA

Krishnabhamini Sinha

Contents

Data Dictionary for Problem 1: (Clustering Clean Ads_Data.xlsx)

Sl. No	**Column Name**	**Column Description**
1	Timestamp	The Timestamp of the particular Advertisement.
2	InventoryType	The Inventory Type of the particular Advertisement. Format 1 to 7. This is a Categorical Variable.
3	Ad - Length	The Length Dimension of the particular Adverstisement.
4	Ad- Width	The Width Dimension of the particular Advertisement.
5	Ad Size	The Overall Size of the particular Advertisement. Length*Width.
6	Ad Type	The type of the particular Advertisement. This is a Categorical Variable.
7	Platform	The platform in which the particular Advertisement is displayed. Web, Video or App. This is a Categorical Variable.
8	Device Type	The type of the device which supports the partciular Advertisement. This is a Categorical Variable.
9	Format	The Format in which the Advertisement is displayed. This is a Categorical Variable.
10	Available_Impressions	How often the particular Advertisement is shown. An impression is counted each time an Advertisement is shown on a search result page or other site on a Network.
11	Matched_Queries	Matched search queries data is pulled from Advertising Platform and consists of the exact searches typed into the search Engine that generated clicks for the particular Advertisement.
12	Impressions	The impression count of the particular Advertisement out of the total available impressions.
13	Clicks	It is a marketing metric that counts the number of times users have clicked on the particular advertisement to reach an online property.
14	Spend	It is the amount of money spent on specific ad variations within a specific campaign or ad set. This metric helps regulate ad performance.
15	Fee	The percentage of the Advertising Fees payable by Franchise Entities.
16	Revenue	It is the income that has been earned from the particular advertisement.
17	CTR	CTR stands for "Click through rate". CTR is the number of clicks that your ad receives divided by the number of times your ad is shown. Formula used here is CTR = Total Measured Clicks / Total Measured Ad Impressions x 100. Note that the Total Measured Clicks refers to the 'Clicks' Column and the Total Measured Ad Impressions refers to the 'Impressions' Column.

18	CPM	CPM stands for "cost per 1000 impressions." Formula used here is CPM = (Total Campaign Spend / Number of Impressions) * 1,000. Note that the Total Campaign Spend refers to the 'Spend' Column and the Number of Impressions refers to the 'Impressions' Column.
19	CPC	CPC stands for "Cost-per-click". Cost-per-click (CPC) bidding means that you pay for each click on your ads. The Formula used here is CPC = Total Cost (spend) / Number of Clicks. Note that the Total Cost (spend) refers to the 'Spend' Column and the Number of Clicks refers to the 'Clicks' Column.

Data Dictionary for Problem 2: (PCA India Data_Census.xlsx)

Name	**Description**
State	State Code
District	District Code
Name	Name
TRU1	Area Name
No_HH	No of Household
TOT_M	Total population Male
TOT_F	Total population Female
M_06	Population in the age group 0-6 Male
F_06	Population in the age group 0-6 Female
M_SC	Scheduled Castes population Male
F_SC	Scheduled Castes population Female
M_ST	Scheduled Tribes population Male
F_ST	Scheduled Tribes population Female
M_LIT	Literates population Male
F_LIT	Literates population Female
M_ILL	Illiterate Male
F_ILL	Illiterate Female
TOT_WORK_M	Total Worker Population Male
TOT_WORK_F	Total Worker Population Female
MAINWORK_M	Main Working Population Male
MAINWORK_F	Main Working Population Female
MAIN_CL_M	Main Cultivator Population Male
MAIN_CL_F	Main Cultivator Population Female
MAIN_AL_M	Main Agricultural Labourers Population Male
MAIN_AL_F	Main Agricultural Labourers Population Female
MAIN_HH_	Main Household Industries Population Male

M	
MAIN_HH_F	Main Household Industries Population Female
MAIN_OT_M	Main Other Workers Population Male
MAIN_OT_F	Main Other Workers Population Female
MARGWORK_M	Marginal Worker Population Male
MARGWORK_F	Marginal Worker Population Female
MARG_CL_M	Marginal Cultivator Population Male
MARG_CL_F	Marginal Cultivator Population Female
MARG_AL_M	Marginal Agriculture Labourers Population Male
MARG_AL_F	Marginal Agriculture Labourers Population Female
MARG_HH_M	Marginal Household Industries Population Male
MARG_HH_F	Marginal Household Industries Population Female
MARG_OT_M	Marginal Other Workers Population Male
MARG_OT_F	Marginal Other Workers Population Female
MARGWORK_3_6_M	Marginal Worker Population 3-6 Male
MARGWORK_3_6_F	Marginal Worker Population 3-6 Female
MARG_CL_3_6_M	Marginal Cultivator Population 3-6 Male
MARG_CL_3_6_F	Marginal Cultivator Population 3-6 Female
MARG_AL_3_6_M	Marginal Agriculture Labourers Population 3-6 Male
MARG_AL_3_6_F	Marginal Agriculture Labourers Population 3-6 Female

MARG_HH_3_6_M	Marginal Household Industries Population 3-6 Male
MARG_HH_3_6_F	Marginal Household Industries Population 3-6 Female
MARG_OT_3_6_M	Marginal Other Workers Population Person 3-6 Male
MARG_OT_3_6_F	Marginal Other Workers Population Person 3-6 Female
MARGWORK_0_3_M	Marginal Worker Population 0-3 Male
MARGWORK_0_3_F	Marginal Worker Population 0-3 Female
MARG_CL_0_3_M	Marginal Cultivator Population 0-3 Male
MARG_CL_0_3_F	Marginal Cultivator Population 0-3 Female
MARG_AL_0_3_M	Marginal Agriculture Labourers Population 0-3 Male
MARG_AL_0_3_F	Marginal Agriculture Labourers Population 0-3 Female
MARG_HH_0_3_M	Marginal Household Industries Population 0-3 Male
MARG_HH_0_3_F	Marginal Household Industries Population 0-3 Female
MARG_OT_0_3_M	Marginal Other Workers Population 0-3 Male
MARG_OT_0_3_F	Marginal Other Workers Population 0-3 Female
NON_WORK_M	Non Working Population Male
NON_WORK_F	Non Working Population Female

Problem 1.1 - Define the problem and perform Exploratory Data Analysis

- Problem definition - Check shape, Data types, statistical summary - Univariate analysis - Bivariate analysis - Key meaningful observations on individual variables and the relationship between variables

Problem Statement:

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. Note that the Total Campaign Spend refers to the 'Spend' Column in the dataset and the Number of Impressions refers to the 'Impressions' Column in the dataset.

CPC = Total Cost (spend) / Number of Clicks. Note that the Total Cost (spend) refers to the 'Spend' Column in the dataset and the Number of Clicks refers to the 'Clicks' Column in the dataset.

CTR = Total Measured Clicks / Total Measured Ad Impressions x 100. Note that the Total Measured Clicks refers to the 'Clicks' Column in the dataset and the Total Measured Ad Impressions refers to the 'Impressions' Column in the dataset.

The Data Dictionary and the detailed description of the formulas for CPM, CPC and CTR are given in the sheet 2 of the Clustering Clean ads_data Excel File.

Perform the following in given order:

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. You may refer to the Bank_KMeans Solution FileView in a new window 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.

Solution:

Shape:

The given dataset has 23066 rows and 19 columns.

```
df.shape
```

```
(23066, 19)
```

Data Types:

The data type for each column is enlisted as below in the figure given below.

```
<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
```

First five rows:

	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	0.00	0.35	0.0000	0.0031	0.0	0.0
1	2020-9-2-10	Format1	300	250	75000	Inter227	App	Mobile	Video	1780	285	285	1	0.00	0.35	0.0000	0.0035	0.0	0.0
2	2020-9-1-22	Format1	300	250	75000	Inter222	Video	Desktop	Display	2727	356	355	1	0.00	0.35	0.0000	0.0028	0.0	0.0
3	2020-9-3-20	Format1	300	250	75000	Inter228	Video	Mobile	Video	2430	497	495	1	0.00	0.35	0.0000	0.0020	0.0	0.0
4	2020-9-4-15	Format1	300	250	75000	Inter217	Web	Desktop	Video	1218	242	242	1	0.00	0.35	0.0000	0.0041	0.0	0.0

Last five rows:

23061	2020-9-13-7	Format5	720	300	216000	Inter220	Web	Mobile	Video	1	1	1	1	0.07	0.35	0.0455	NaN	NaN	NaN
23062	2020-11-2-7	Format5	720	300	216000	Inter224	Web	Desktop	Video	3	2	2	1	0.04	0.35	0.0260	NaN	NaN	NaN
23063	2020-9-14-22	Format5	720	300	216000	Inter218	App	Mobile	Video	2	1	1	1	0.05	0.35	0.0325	NaN	NaN	NaN
23064	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
23065	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

23066 rows × 19 columns

Checking Null Values:

```
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
```

Treating Missing Values in CTR, CPM and CPC using the given values:

- 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.

Checking duplicate values:

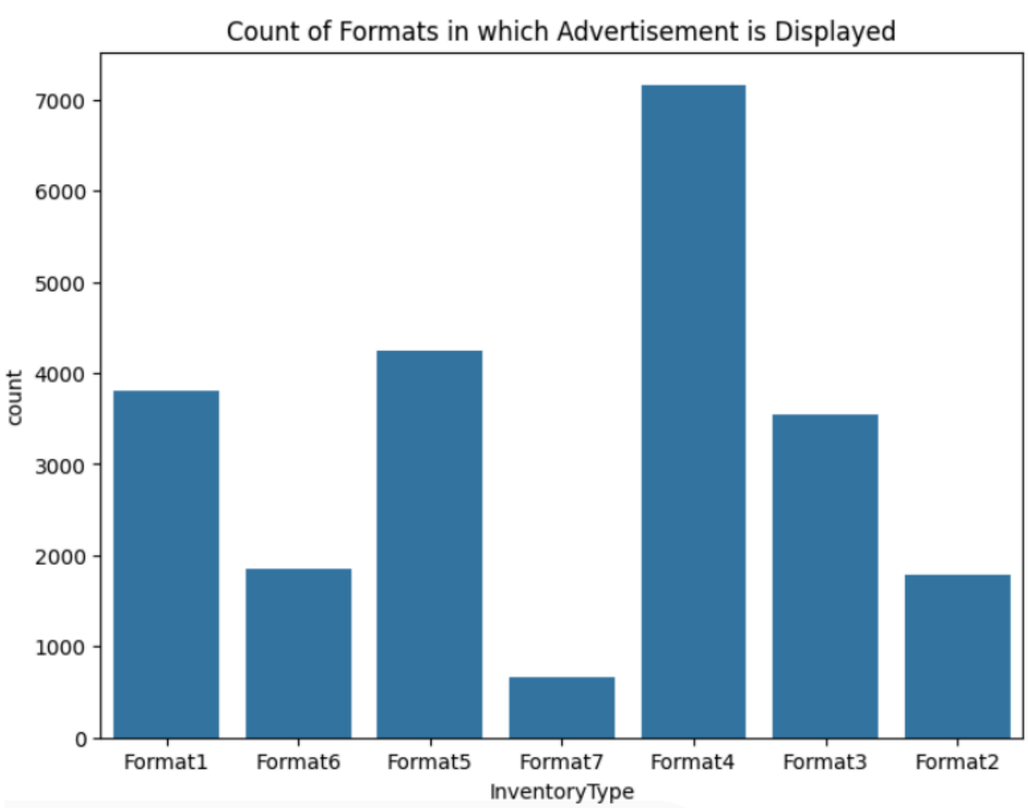
```
df.duplicated().sum()
```

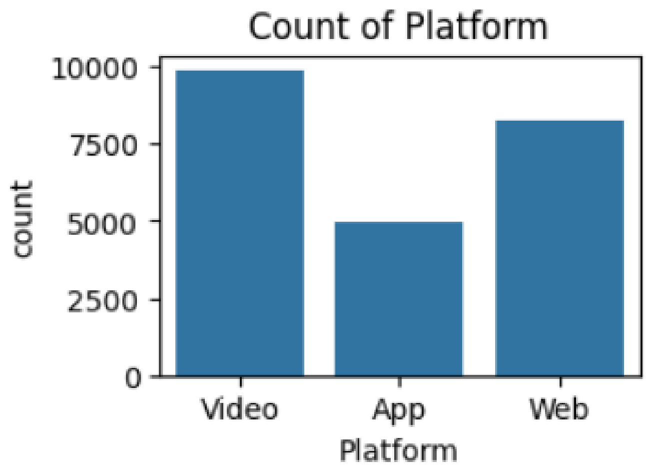
```
0
```

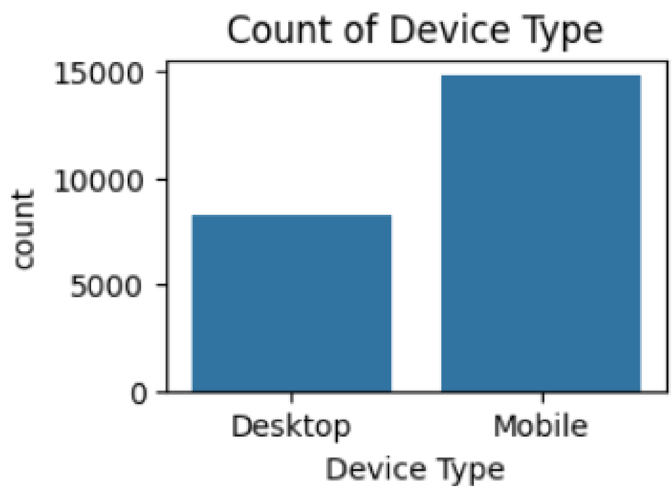
Statistical Summary:

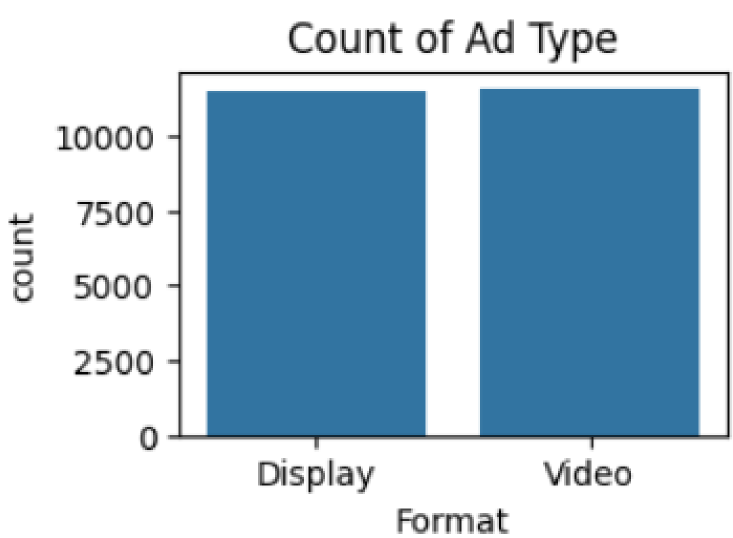
	count	mean	std	min	25%	50%	75%	max
Ad - Length	23066.0	3.851631e+02	2.336514e+02	120.0000	120.000000	300.00000	7.200000e+02	728.00
Ad- Width	23066.0	3.378960e+02	2.030929e+02	70.0000	250.000000	300.00000	6.000000e+02	600.00
Ad Size	23066.0	9.667447e+04	6.153833e+04	33600.0000	72000.000000	72000.00000	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

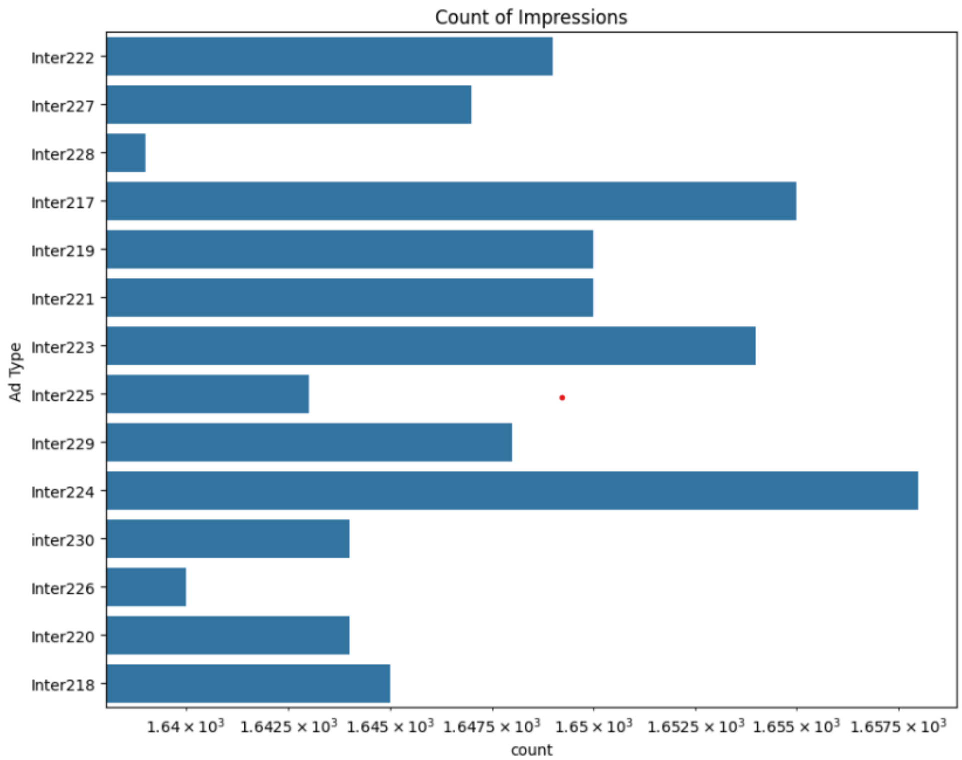
Univariate Analysis:



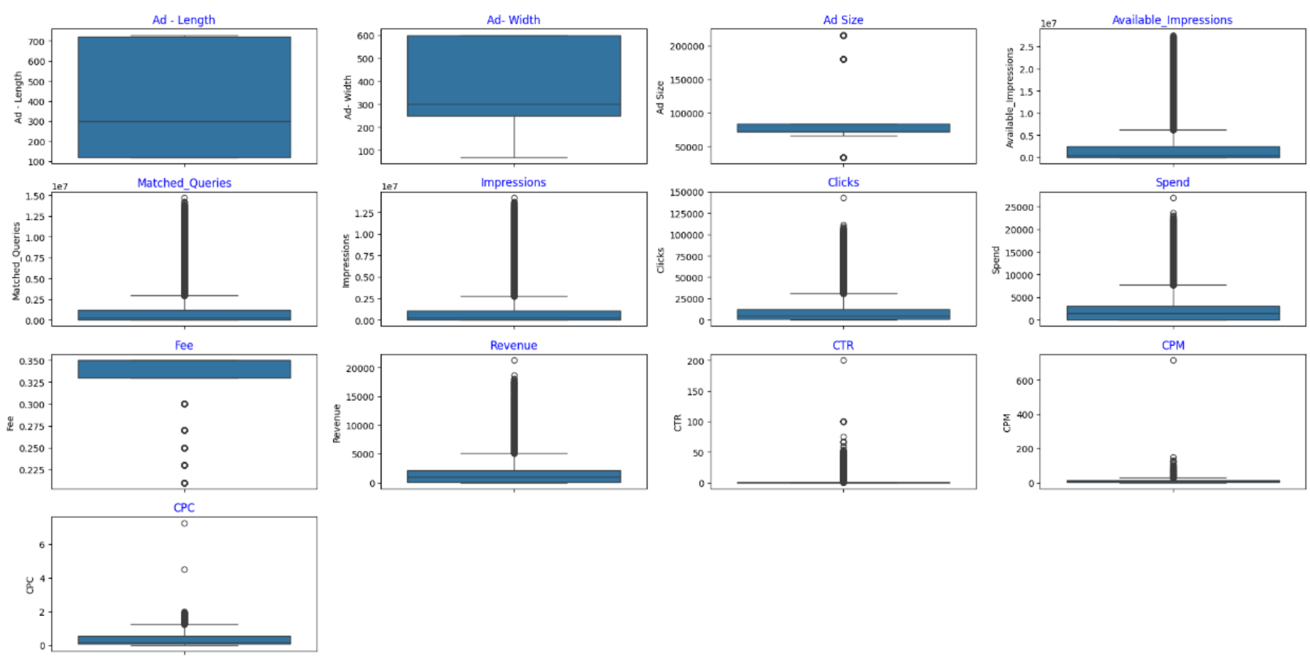

Count of Platform
count
10000
7500
5000
2500
0
Video
App
Web
Platform


Count of Device Type
count
15000
10000
5000
0
Desktop
Mobile
Device Type


Count of Ad Type
count
10000
7500
5000
2500
0
Display
Video
Format



Count of Impressions
Ad Type
Inter222
Inter227
Inter228
Inter217
Inter219
Inter221
Inter223
Inter225
Inter229
Inter224
inter230
Inter226
Inter220
Inter218
count

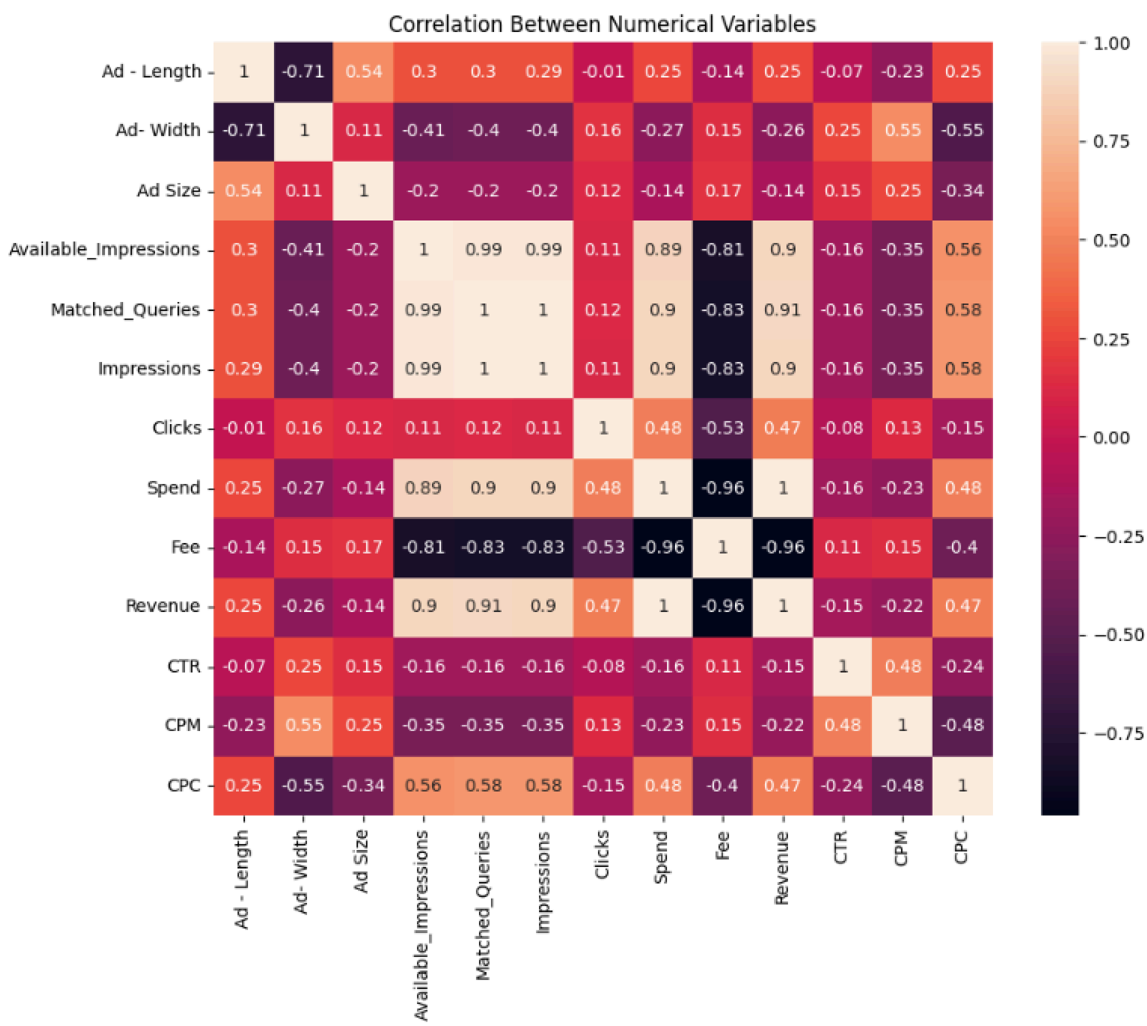


Ad - Length
Ad- Width
Ad Size
Available_Impressions
Matched_Queries
Impressions
Clicks
Spend
Fee
Revenue
CTR
CPM
CPC

Bivariate analysis

Let us look at the correlation b/w the numerical variables of the data set.

	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
Ad - Length	1.00	-0.71	0.54	0.30	0.30	0.29	-0.01	0.25	-0.14	0.25	-0.07	-0.23	0.25
Ad- Width	-0.71	1.00	0.11	-0.41	-0.40	-0.40	0.16	-0.27	0.15	-0.26	0.25	0.55	-0.55
Ad Size	0.54	0.11	1.00	-0.20	-0.20	-0.20	0.12	-0.14	0.17	-0.14	0.15	0.25	-0.34
Available_Impressions	0.30	-0.41	-0.20	1.00	0.99	0.99	0.11	0.89	-0.81	0.90	-0.16	-0.35	0.56
Matched_Queries	0.30	-0.40	-0.20	0.99	1.00	1.00	0.12	0.90	-0.83	0.91	-0.16	-0.35	0.58
Impressions	0.29	-0.40	-0.20	0.99	1.00	1.00	0.11	0.90	-0.83	0.90	-0.16	-0.35	0.58
Clicks	-0.01	0.16	0.12	0.11	0.12	0.11	1.00	0.48	-0.53	0.47	-0.08	0.13	-0.15
Spend	0.25	-0.27	-0.14	0.89	0.90	0.90	0.48	1.00	-0.96	1.00	-0.16	-0.23	0.48
Fee	-0.14	0.15	0.17	-0.81	-0.83	-0.83	-0.53	-0.96	1.00	-0.96	0.11	0.15	-0.40
Revenue	0.25	-0.26	-0.14	0.90	0.91	0.90	0.47	1.00	-0.96	1.00	-0.15	-0.22	0.47
CTR	-0.07	0.25	0.15	-0.16	-0.16	-0.16	-0.08	-0.16	0.11	-0.15	1.00	0.48	-0.24
CPM	-0.23	0.55	0.25	-0.35	-0.35	-0.35	0.13	-0.23	0.15	-0.22	0.48	1.00	-0.48
CPC	0.25	-0.55	-0.34	0.56	0.58	0.58	-0.15	0.48	-0.40	0.47	-0.24	-0.48	1.00



Key meaningful observations on individual variables and the relationship between variables:

- The format in which advertisements are displayed the most is Format 4.
- The platform in which advertisements are displayed the most is the video platform.
- The type of device which supports advertisements the most is the mobile.
- Type of advertisement which is most popular is Inter224

Problem 1.2 - Data Preprocessing

- Missing value check and treatment - Outlier Treatment - z-score scaling Note: Treat missing values in CPC, CTR and CPM using the formula given.

Solution:

```
df.isnull().sum()
```

```
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 missing values in CTC, CPM and CPC variables of the dataset. We have already treated the missing values in the previous question by use of the definitions given in the problem statement. Given below is the new display of null values in the dataset.

```
df['CTR'] = df['CTR'].fillna((df['Clicks']/df['Impressions'])*100)
df['CPM'] = df['CPM'].fillna((df['Spend']/df['Impressions'])*1000)
df['CPC'] = df['CPC'].fillna(df['Spend']/df['Clicks'])
```

```
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
```

Outlier treatment:

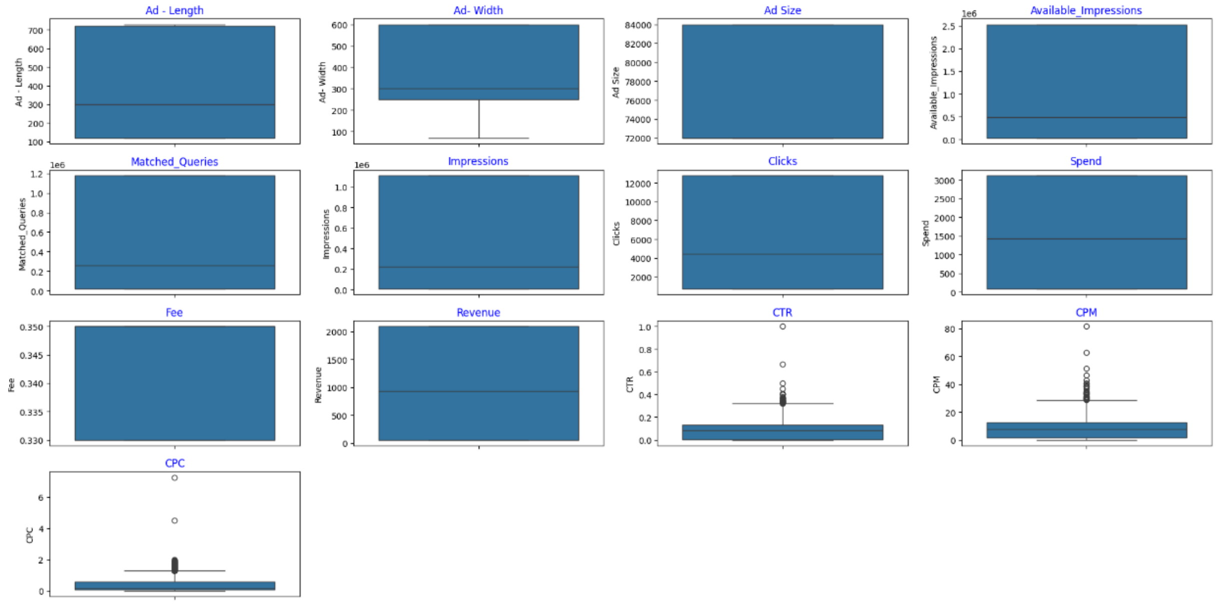
Possibilities of outlier treatment:

1. Treating outliers using IQR method.

2. Treating outliers using z-score method.

3. Using EDA results to segment data into two or more parts and then apply k-means algorithm to each part separately.

For this dataset, we will treat outliers using IQR Method, and compare results with model without outlier treatment. Outlier Detection and Treatment 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 remove_outlier' The larger values (>upper whisker) are all equated to the 95th percentile value of the distribution The smaller values (<lower whisker) are all equated to the 5th percentile value of the distribution.



We can see that even after outlier treatment, outliers are still present in CTR, CPM and CPC variables.

Z-score Scaling:

We used scikit-learn's Standard Scaler to perform z-score scaling. Below Table shows the first five rows of this scaled data.

	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
0	-0.364496	-0.432797	-0.192668	-0.934925	-0.98599	-0.978479	-1.043986	-1.156558	0.587931	-1.145389	-1.200938	-1.089678	-0.9113
1	-0.364496	-0.432797	-0.192668	-0.934925	-0.98599	-0.978479	-1.043986	-1.156558	0.587931	-1.145389	-1.199606	-1.089678	-0.9113
2	-0.364496	-0.432797	-0.192668	-0.934925	-0.98599	-0.978479	-1.043986	-1.156558	0.587931	-1.145389	-1.200938	-1.089678	-0.9113
3	-0.364496	-0.432797	-0.192668	-0.934925	-0.98599	-0.978479	-1.043986	-1.156558	0.587931	-1.145389	-1.200938	-1.089678	-0.9113
4	-0.364496	-0.432797	-0.192668	-0.934925	-0.98599	-0.978479	-1.043986	-1.156558	0.587931	-1.145389	-1.191614	-1.089678	-0.9113

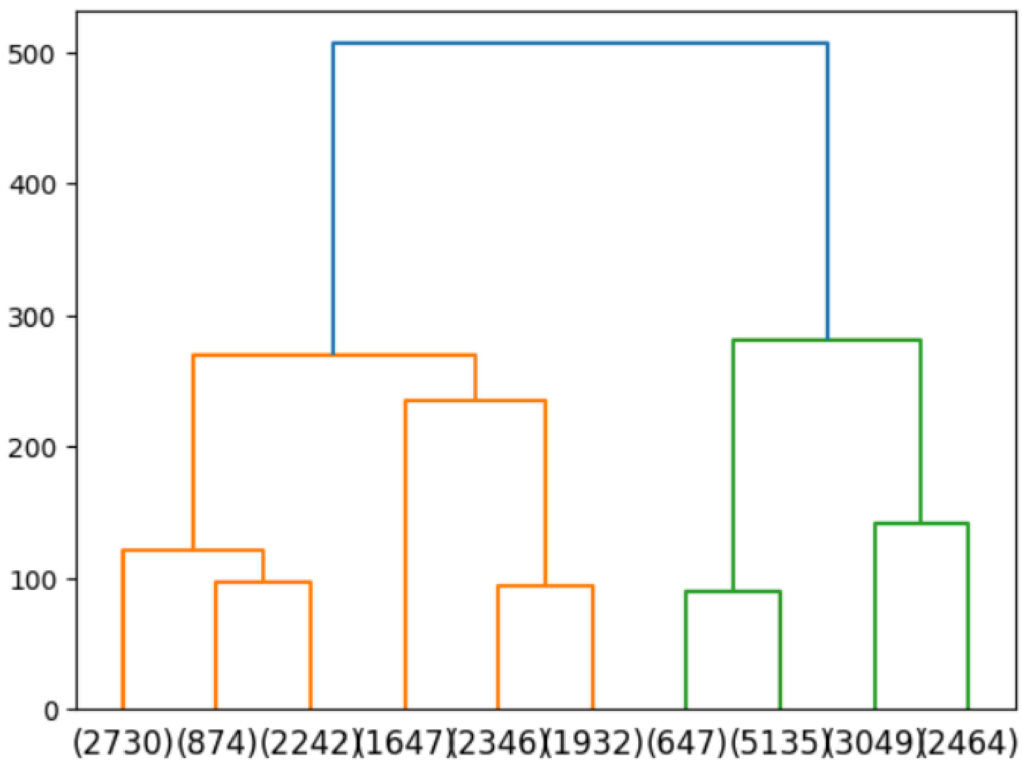
Scaling of variables is important for clustering to stabilize the weights of the different variables.

Problem 1.3 - Hierarchical Clustering

- Construct a dendrogram using Ward linkage and Euclidean distance - Identify the optimum number of Clusters

Solution:

Using SciPy's cluster hierarchy function, we created the below dendrogram Dendrogram using WARD and Euclidean distance.



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.

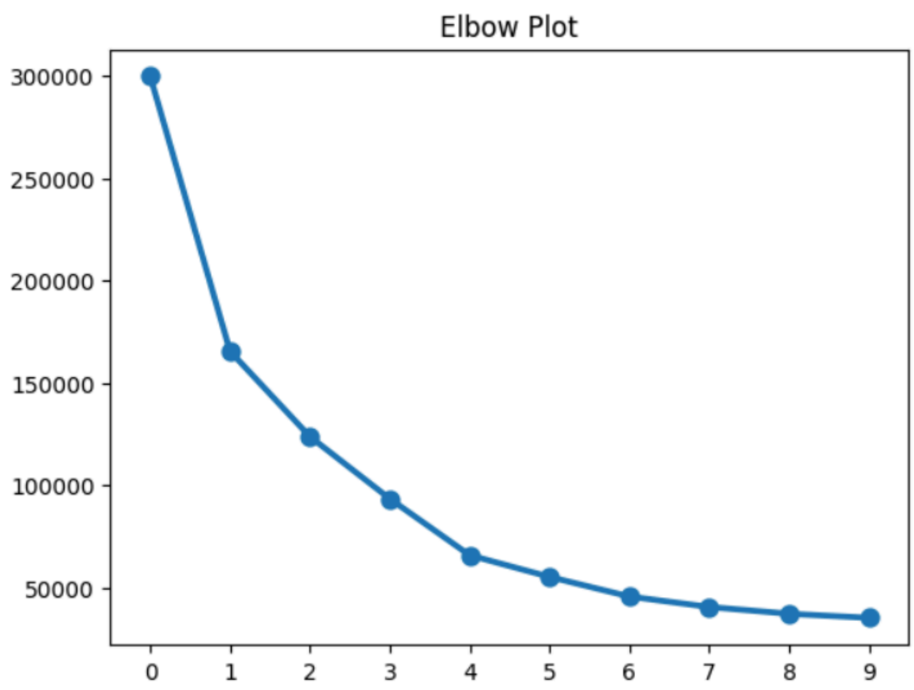
Keeping the above reference as base, we can see the longest branch (tallest branch) is in blue. If we see that only blue, it will result in only 2 clusters which is not acceptable in business. We move towards further broad divisions of branches and consider the next tallest ones - 3 red and two green branches accumulating to a total of 5 clusters.

Problem 1.4 - K-means Clustering

- Apply K-means Clustering - Plot the Elbow curve - Check Silhouette Scores - Figure out the appropriate number of clusters - Cluster Profiling.

Solution:

Elbow curve:



Using K-means algorithm we have plotted the Elbow plot and the optimal number of clusters is 5.

Silhouette Scores:

We have calculated the Silhouette Score of clusters up to 10. Silhouette Score of clusters = 5 is the highest after clusters = 4. Therefore we conclude the final number of clusters to be 5. Also, the k-means inertia seems to change almost in smaller fractions after cluster number 5.

```
For n_clusters=2, the silhouette score is 0.4174565809886918
For n_clusters=3, the silhouette score is 0.4141149320082417
For n_clusters=4, the silhouette score is 0.44561614570521746
For n_clusters=5, the silhouette score is 0.5004668128097572
For n_clusters=6, the silhouette score is 0.4964916407834132
For n_clusters=7, the silhouette score is 0.5002818809148312
For n_clusters=8, the silhouette score is 0.49531938698785
For n_clusters=9, the silhouette score is 0.5103575927327262
For n_clusters=10, the silhouette score is 0.5280527596928045
```

Let us look at the number of records per cluster and we proceed towards profiling. Proportion of records per label.

```
Clus_means
0    5807
1    5995
2    4282
3    5220
4    1762
```

Clus_means	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC	freq
0	430.698812	144.112278	74086.102979	1.681978e+06	8.046484e+05	7.615989e+05	3390.724298	1496.242025	0.350000	972.557312	0.031833	1.965797	0.439511	5807
1	132.670559	586.747289	72503.919933	4.349669e+04	2.433710e+04	1.321951e+04	1556.350626	173.170974	0.350000	112.559863	0.144654	11.225295	0.113099	5995
2	457.791686	202.230266	74968.472676	2.526526e+06	1.180612e+06	1.112347e+06	8915.523237	3119.545579	0.330019	2090.041521	0.030105	1.818765	0.537616	4282
3	647.620690	299.204981	84000.000000	2.321325e+05	1.269531e+05	1.060100e+05	8600.710297	1137.540257	0.349628	740.554309	0.138127	10.892766	0.099150	5220
4	140.124858	574.177072	73164.585698	7.292611e+05	5.147493e+05	4.340356e+05	12788.822361	3028.614103	0.332406	2023.507472	0.145453	12.609742	0.116608	1762

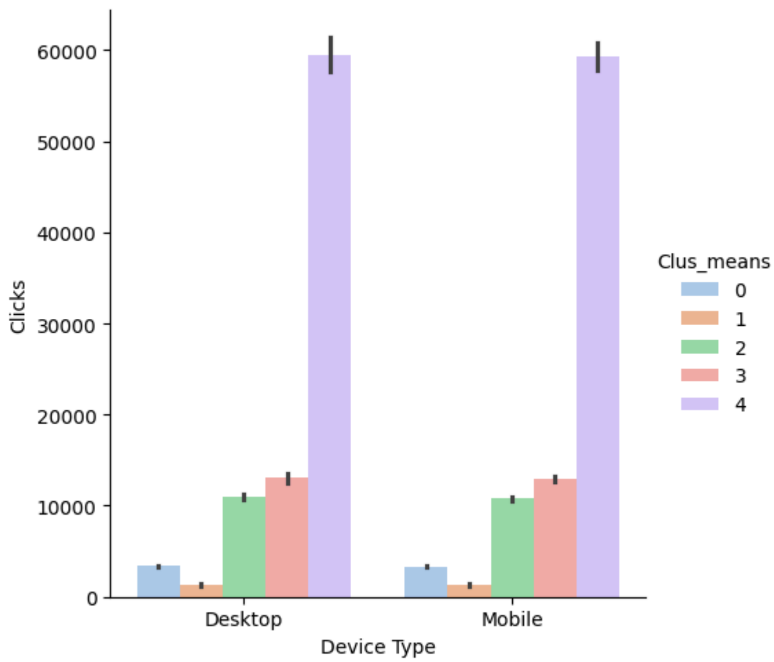
Observations:

- 0 - highest matched queries, impressions, fee, CTR and second highest CPC.
- 1 - highest ad-width, fee and second highest CPM and available impressions.
- 2 - second highest ad-length,clicks, spend, fee, CTR, highest CPC and lowest ad-width.
- 3 - highest ad-length, ad size, CPC, third highest clicks and CPM and second highest fee.
- 4 - second highest ad-width, highest available impressions, clicks and CPM.

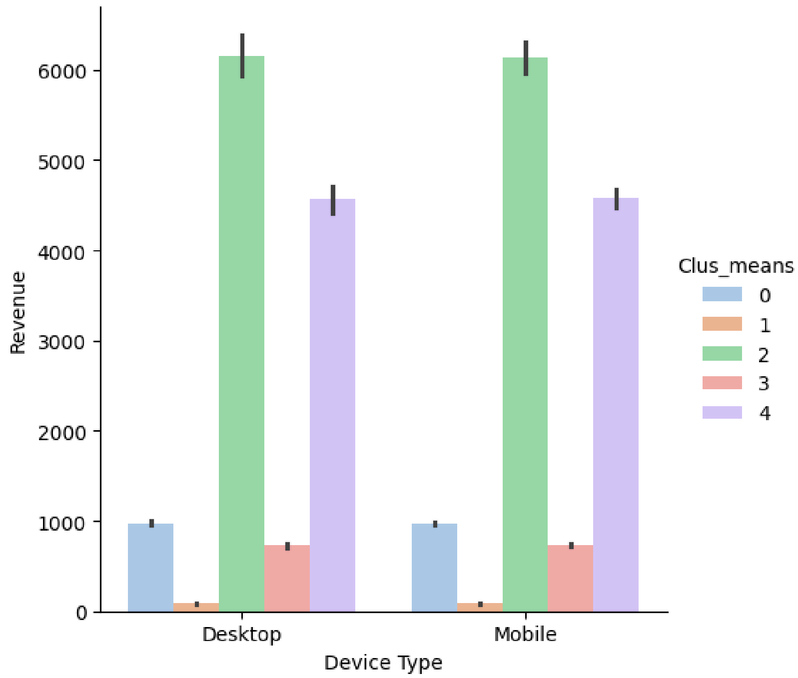
Problem 1.5 - Actionable Insights & Recommendations

- Extract meaningful insights (atleast 3) from the clusters to identify the most effective types of ads, target audiences, or marketing strategies that can be inferred from each segment. - Based on the clustering analysis and key insights, provide actionable recommendations (atleast 3) to Ads24x7 on how to optimize their digital marketing efforts, allocate budgets efficiently, and tailor ad content to specific audience segments.

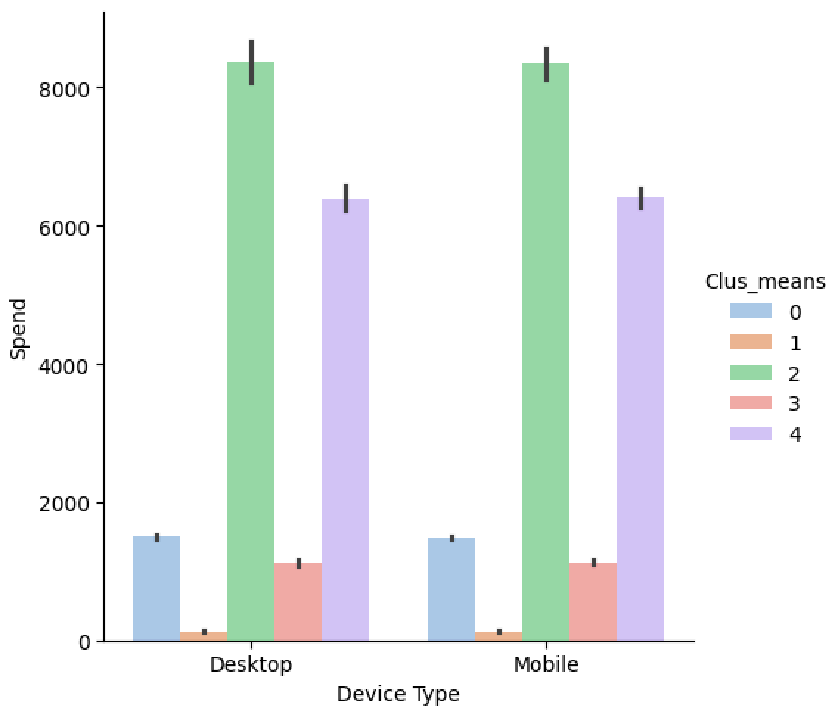
Let us look at some further charts to clarify our cluster profiling



<u>Observation 1:</u> cluster 4 has the highest number of clicks both through desktop and mobile. This statistic is followed by cluster 3 and cluster 2. The lowest count of clicks is found in cluster 1 both through mobile and desktop.



Observation 2: Revenue collected via mobile and desktop is the highest for cluster 2 followed by cluster 4 the lowest count of revenue is found in cluster 1.



Observation 3: Maximum amount of money is spent on devising ads in cluster 2 followed by cluster 4. Cluster zero has below average expenditure. Cluster 1 has the lowest amount of money spent on it..

Actionable Insights & Recommendations:

Selling ads according to CPM puts a ceiling on revenue. If we want to increase our revenue, we have to spend money on increasing our reach to create more ad opportunities or pumping out more ads to the same users before seeing a return.

But if we sell on CTR, revenue is not capped. We can increase engagement on the same number of impressions per person, or DAU (daily active user). Whereas with CPM, we stretch to reach more and more people, or degrade our user experience with more ads per user

Problem 2.1 - Define the problem and perform Exploratory Data Analysis

- Problem Definition - Check shape, Data types, statistical summary - Perform an EDA on the data to extract useful insights Note: 1. 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?

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. Use Sklearn only.

Note: The 24 variables given in the Rubric is just for performing EDA. You will have to consider the entire dataset, including all the variables for performing PCA.
Data file - PCA India Data Census.xlsx

Solution:

First five rows of the dataset:

	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

5 rows × 61 columns

Last five rows of the dataset:

	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

Shape of the dataset:

```
pc.shape
```

```
(640, 61)
```

Checking Data Types:

```
<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
```

Statistical Summary:

	count	mean	std	min	25%	50%	75%	max
State Code	640.0	17.114062	9.426486	1.000000	9.000000	18.000000	24.000000	35.000000
Dist.Code	640.0	320.500000	184.896367	1.000000	160.750000	320.500000	480.250000	640.000000
No_HH	640.0	51222.871875	48135.405475	350.000000	19484.000000	35837.000000	68892.000000	310450.000000
TOT_M	640.0	79940.576563	73384.511114	391.000000	30228.000000	58339.000000	107918.500000	485417.000000
TOT_F	640.0	122372.084375	113600.717282	698.000000	46517.750000	87724.500000	164251.750000	750392.000000
...	...	...	...	...	...	...	...	...
NON_WORK_M	640.0	510.014063	610.603187	0.000000	161.000000	326.000000	604.500000	6456.000000
NON_WORK_F	640.0	704.778125	910.209225	5.000000	220.500000	464.500000	853.500000	10533.000000
Total Male Population	640.0	92249.675000	84395.070530	447.000000	35159.250000	67967.000000	124873.500000	574013.000000
Total Female Population	640.0	134314.384375	123960.865240	754.000000	51963.250000	96462.500000	181301.500000	834570.000000
Gender Ratio	640.0	68.964855	9.107841	46.311555	62.151322	68.893181	76.416851	89.685113

Checking Null Values:

```
pc.isnull().sum()
```

```
State Code       0
Dist.Code        0
State            0
Area Name        0
No_HH            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
Length: 61, dtype: int64
```

Checking Duplicated Values:

```
pc.duplicated().sum()
```

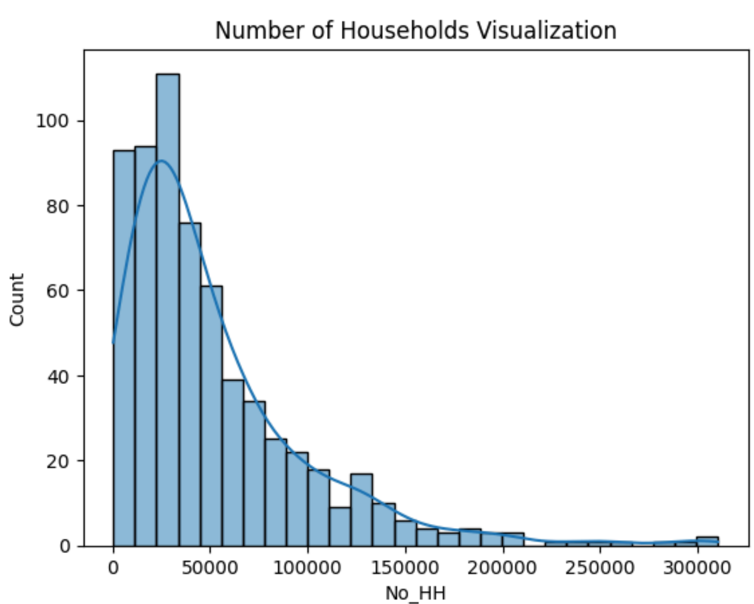
```
0
```

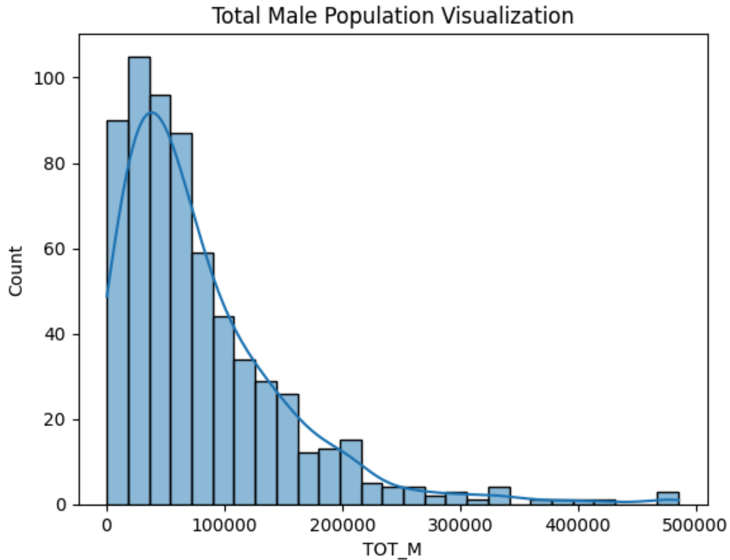
EDA

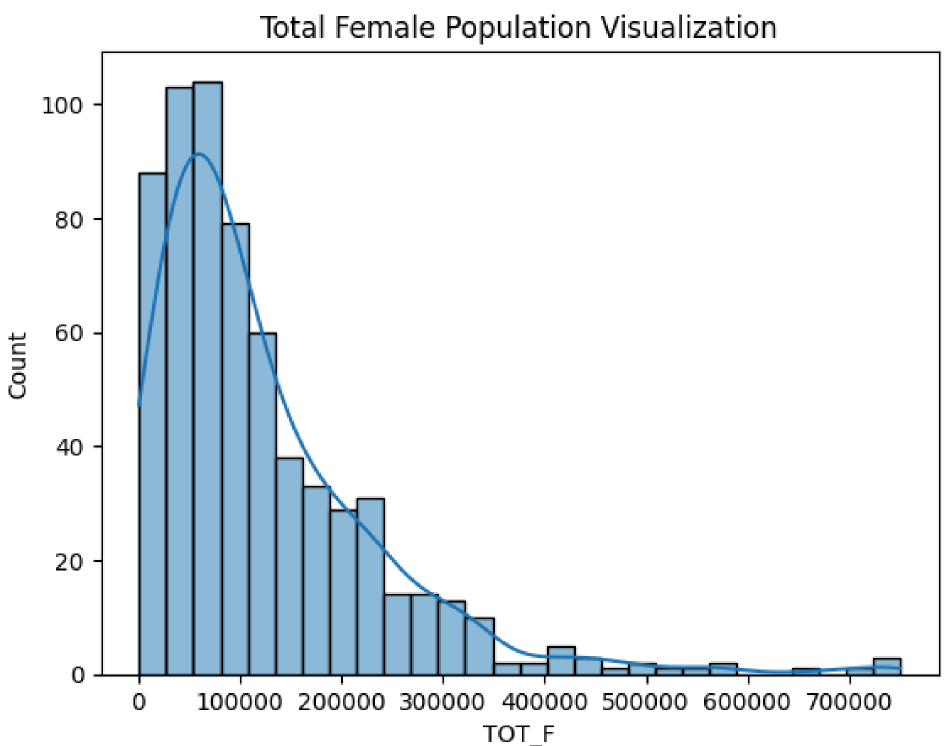
According to the given question 5 variables out of the 24 variables have been selected for EDA. these variable are: No_HH, TOT_M, TOT_F, M_06 and F_06.

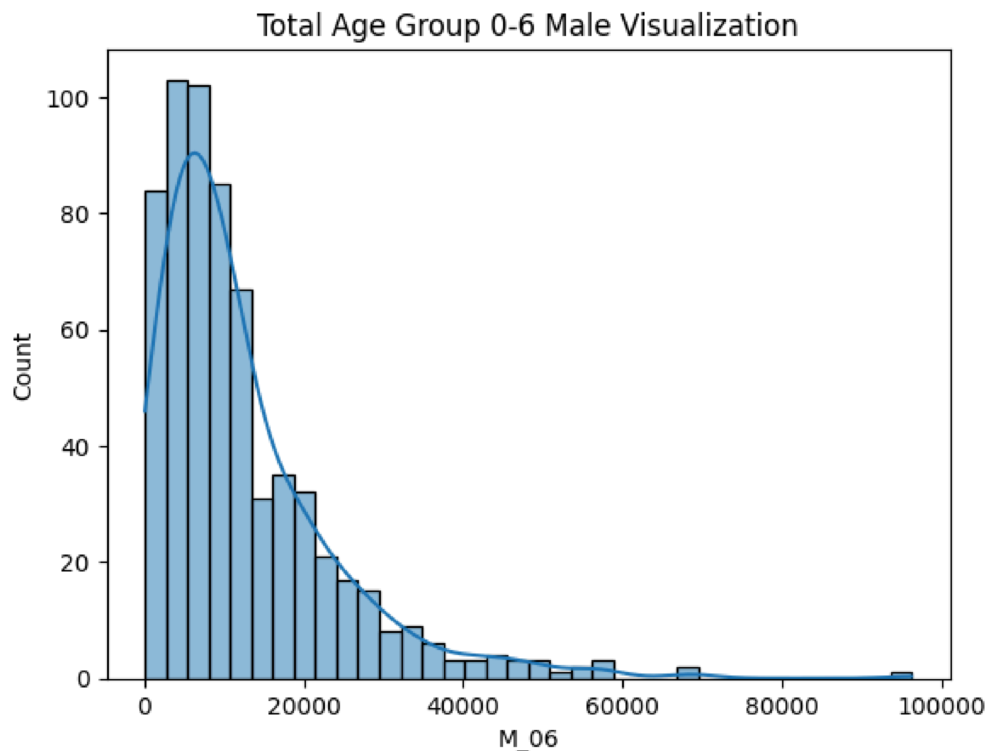
No_HH	No of Household
TOT_M	Total population Male
TOT_F	Total population Female
M_06	Population in the age group 0-6 Male
F_06	Population in the age group 0-6 Female

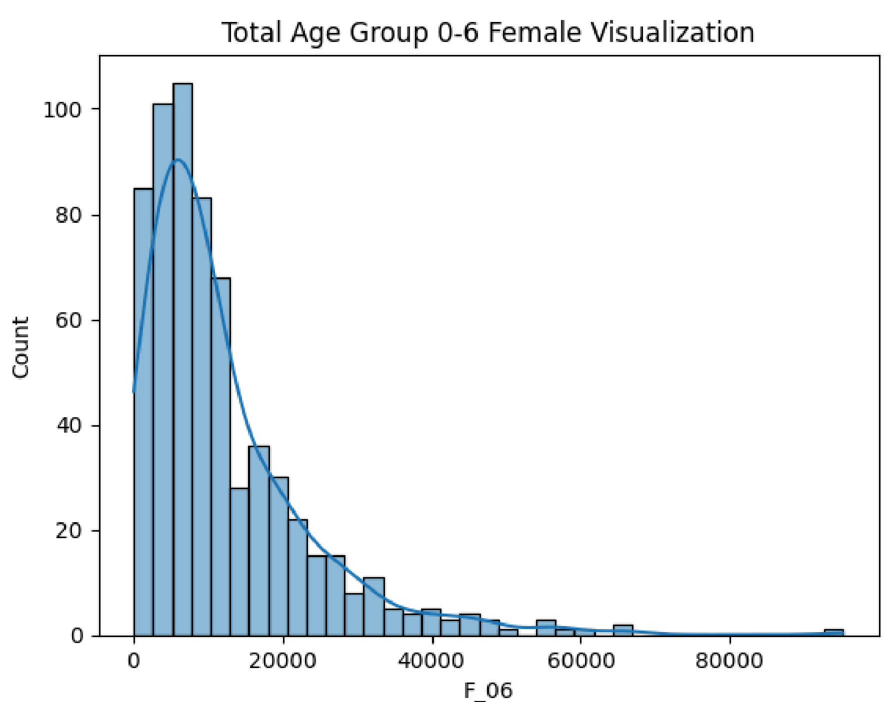
Univariate Analysis:



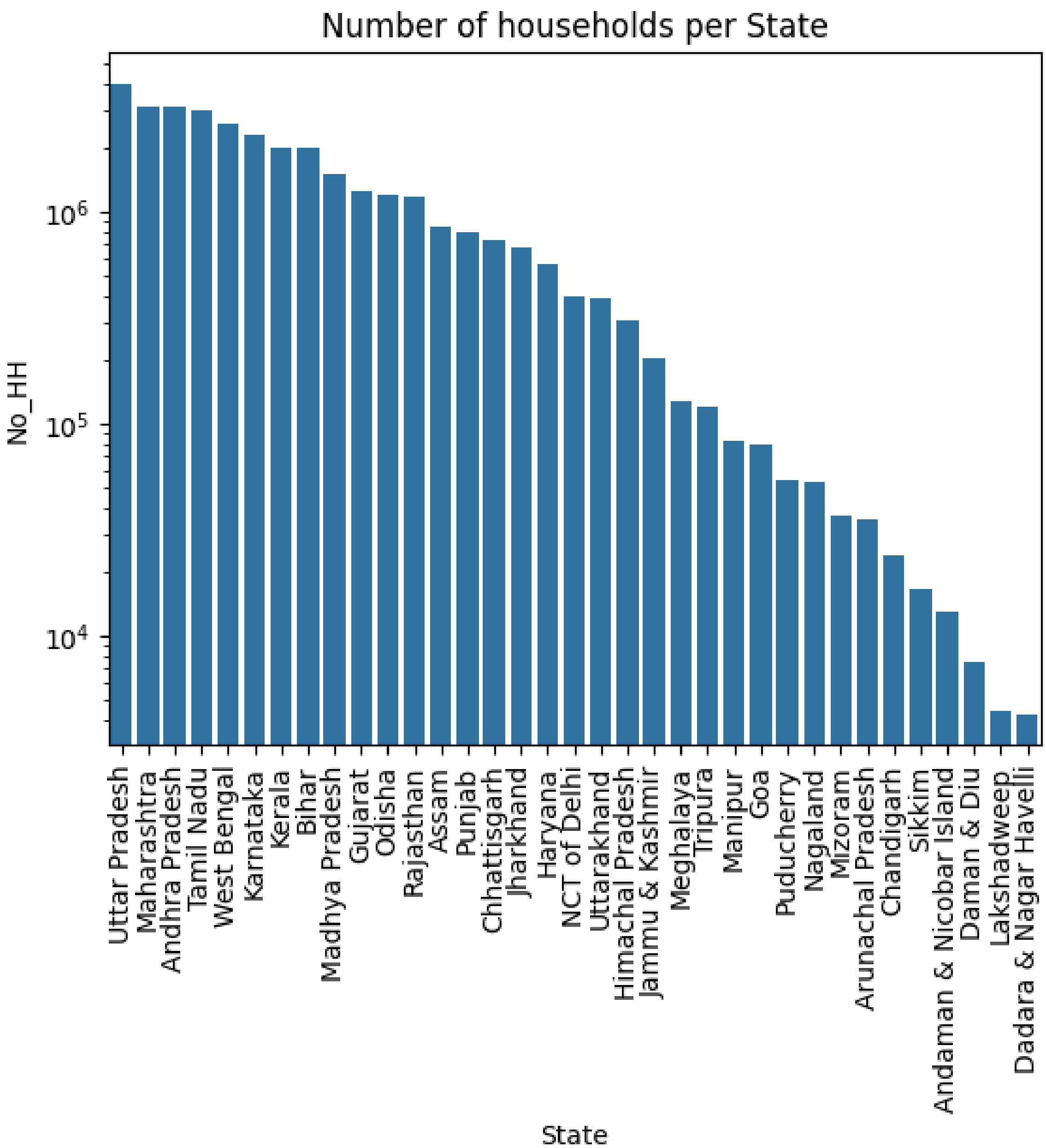

Total Male Population Visualization
Count
0
20
40
60
80
100
0
100000
200000
300000
400000
500000
TOT_M

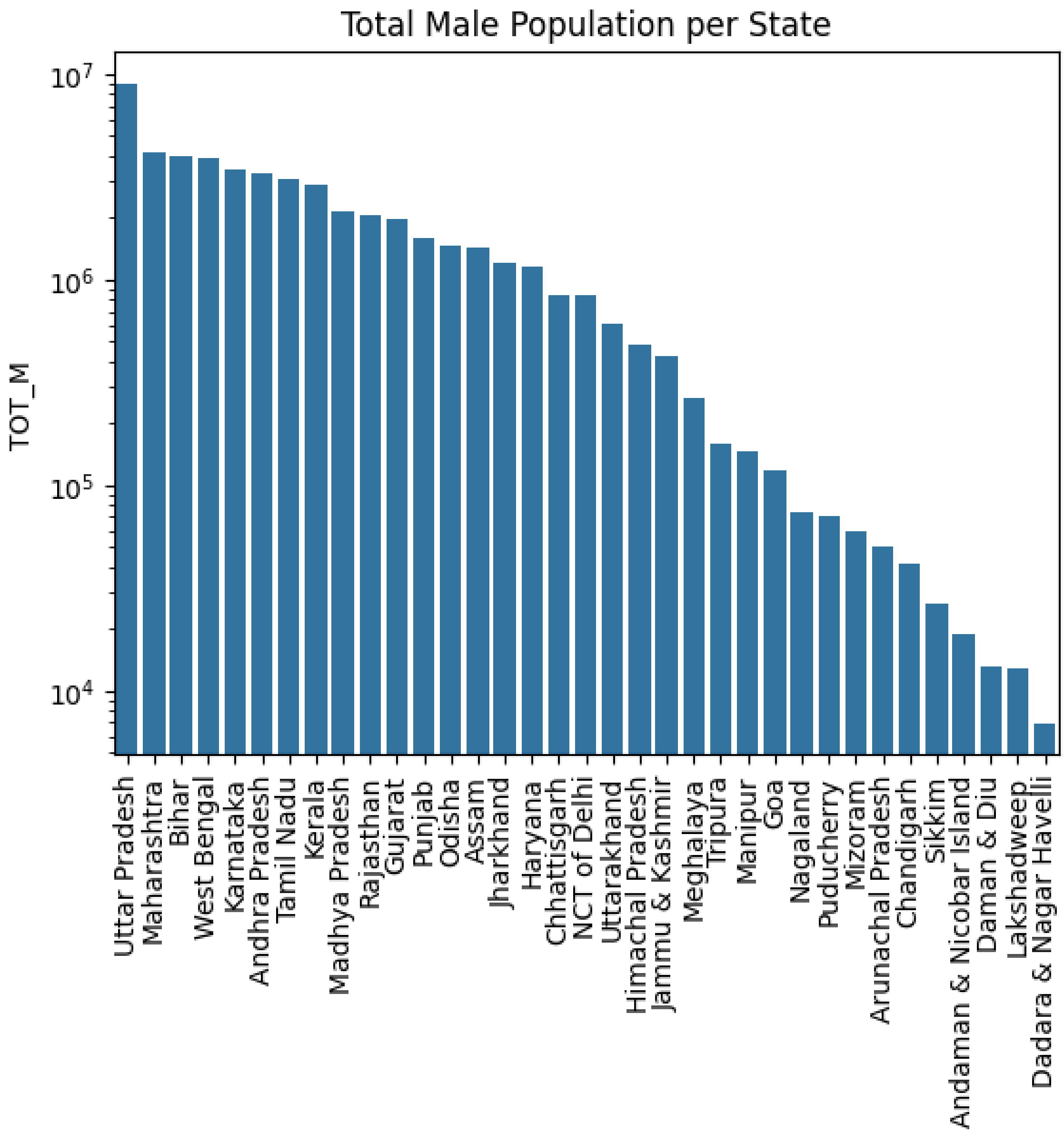

Total Female Population Visualization
Count
0
20
40
60
80
100
0
100000
200000
300000
400000
500000
600000
700000
TOT_F

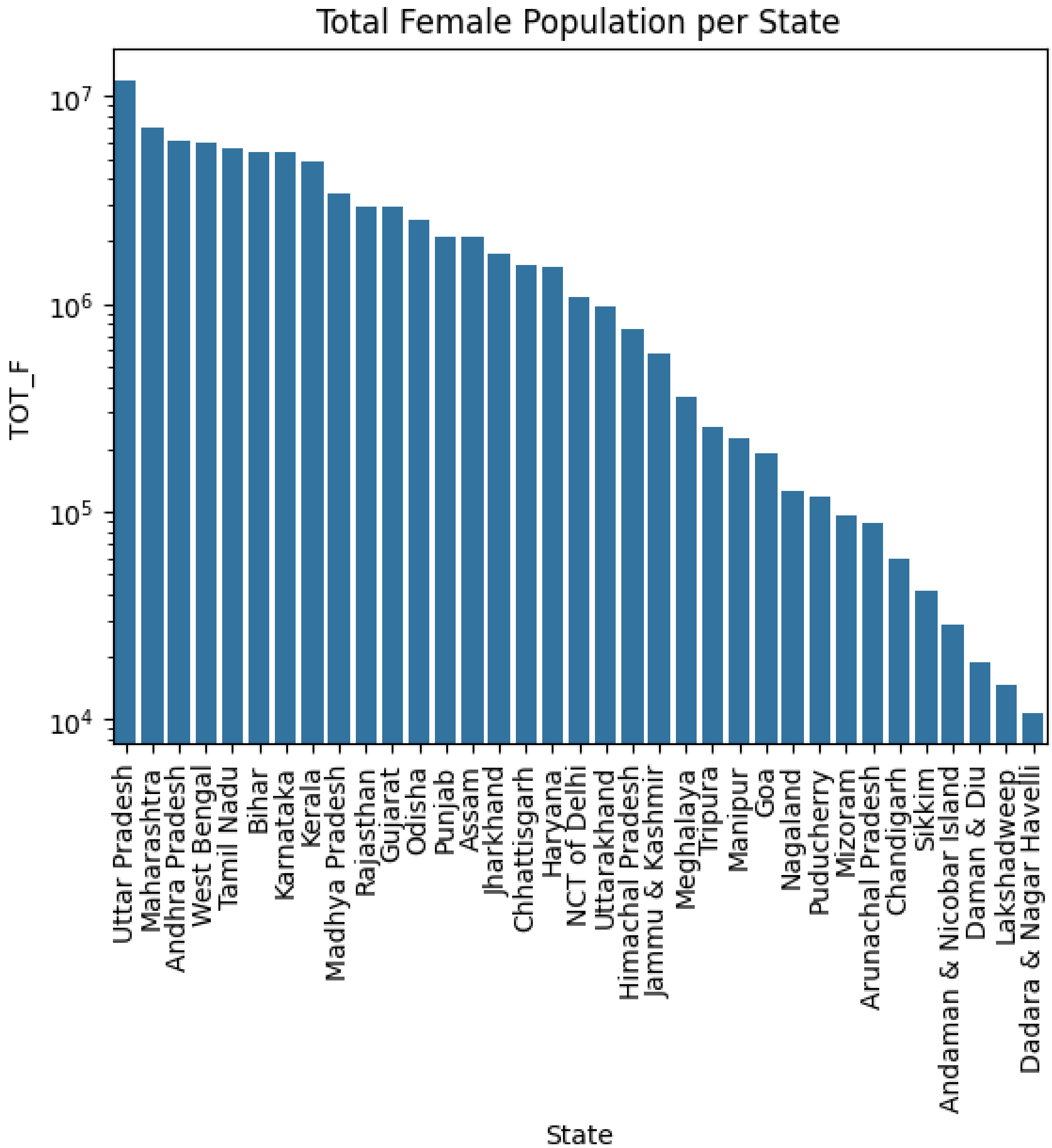

Total Age Group 0-6 Male Visualization
Count
M_06
0
20
40
60
80
100
0
20000
40000
60000
80000
100000

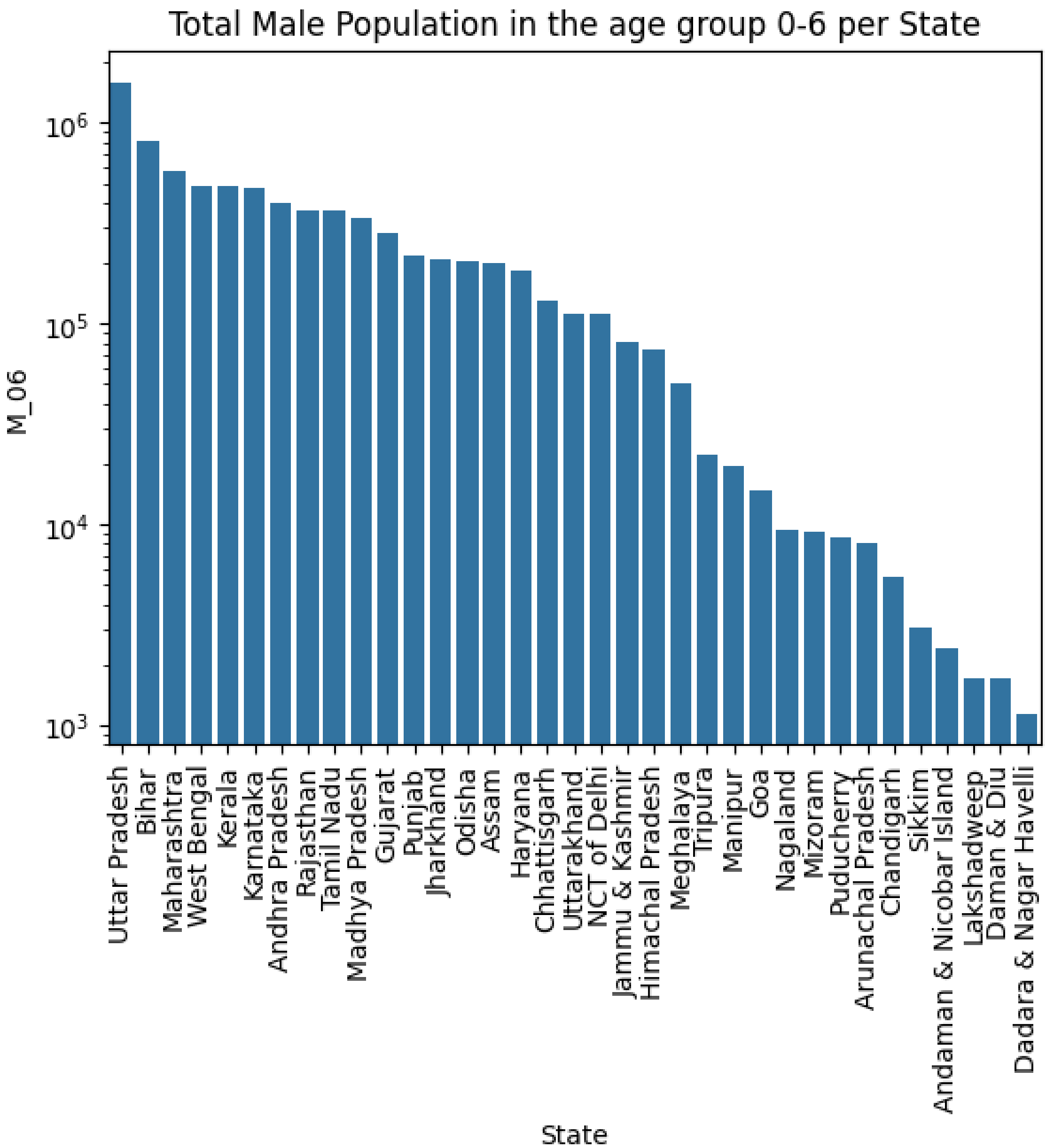

Total Age Group 0-6 Female Visualization
Count
F_06
0
20
40
60
80
100
0
20000
40000
60000
80000

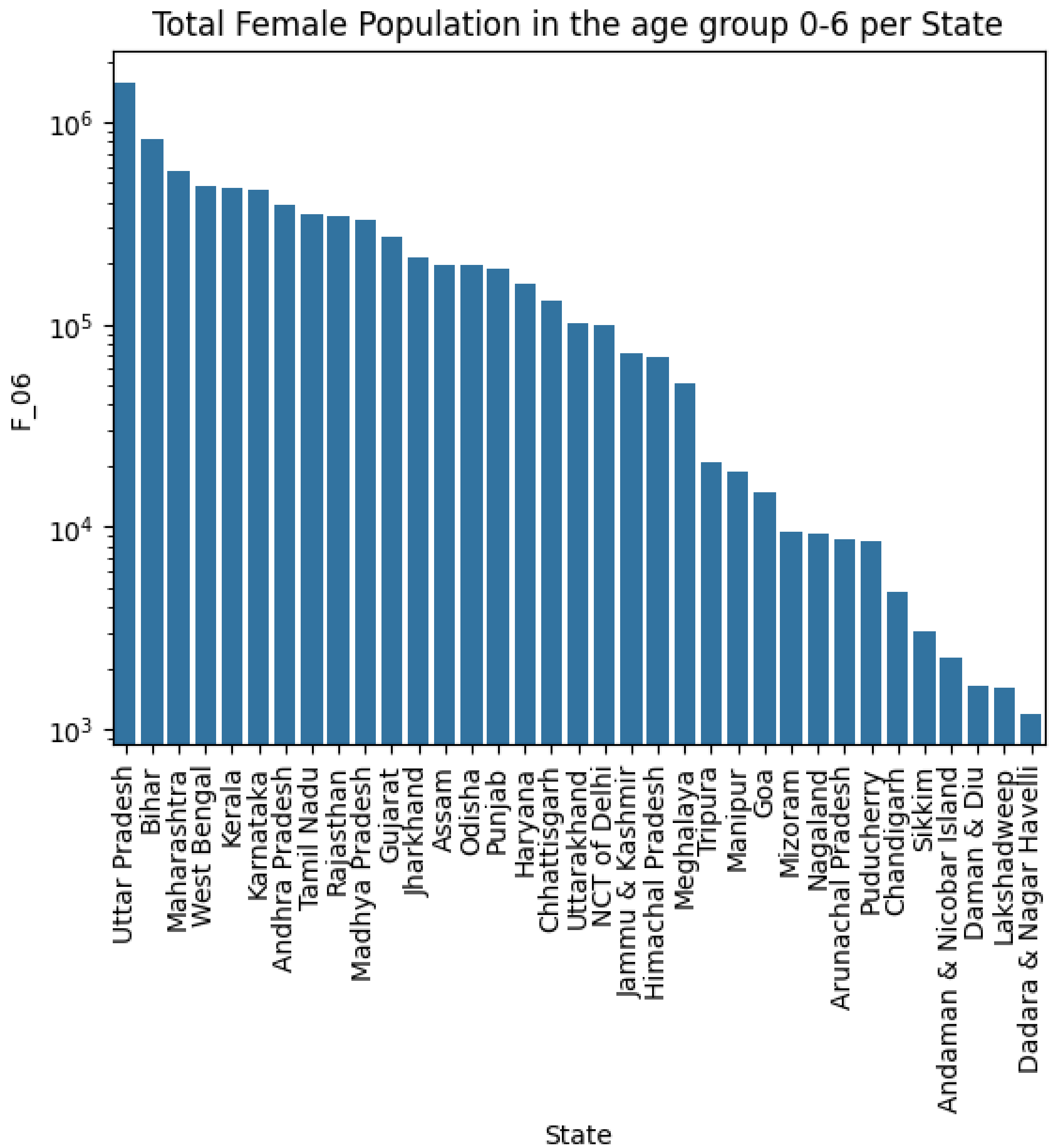
Bivariate Analysis



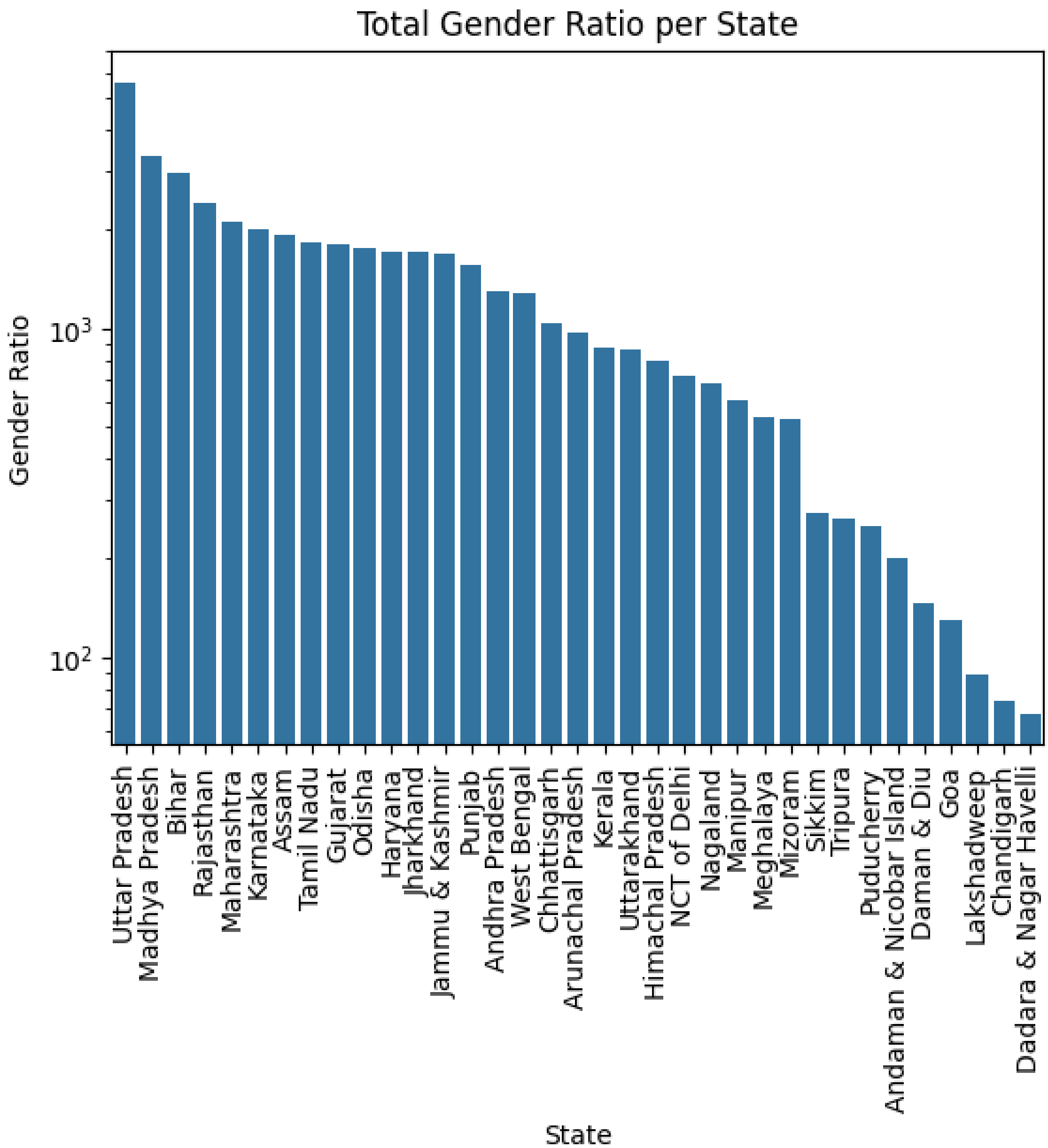

Total Male Population per State
TOT_M
10^7
10^6
10^5
10^4
State
Uttar Pradesh
Maharashtra
Bihar
West Bengal
Karnataka
Andhra Pradesh
Tamil Nadu
Kerala
Madhya Pradesh
Rajasthan
Gujarat
Punjab
Odisha
Assam
Jharkhand
Haryana
Chhattisgarh
NCT of Delhi
Uttarakhand
Himachal Pradesh
Jammu & Kashmir
Meghalaya
Tripura
Manipur
Goa
Nagaland
Puducherry
Mizoram
Arunachal Pradesh
Chandigarh
Sikkim
Andaman & Nicobar Island
Daman & Diu
Lakshadweep
Dadara & Nagar Haveli


Total Female Population per State
TOT_F
10^7
10^6
10^5
10^4
State
Uttar Pradesh
Maharashtra
Andhra Pradesh
West Bengal
Tamil Nadu
Bihar
Karnataka
Kerala
Madhya Pradesh
Rajasthan
Gujarat
Odisha
Punjab
Assam
Jharkhand
Chhattisgarh
Haryana
NCT of Delhi
Uttarakhand
Himachal Pradesh
Jammu & Kashmir
Meghalaya
Tripura
Manipur
Goa
Nagaland
Puducherry
Mizoram
Arunachal Pradesh
Chandigarh
Sikkim
Andaman & Nicobar Island
Daman & Diu
Lakshadweep
Dadara & Nagar Havelli


Total Male Population in the age group 0-6 per State
M_06
10^6
10^5
10^4
10^3
State
Uttar Pradesh
Bihar
Maharashtra
West Bengal
Kerala
Karnataka
Andhra Pradesh
Rajasthan
Tamil Nadu
Madhya Pradesh
Gujarat
Punjab
Jharkhand
Odisha
Assam
Haryana
Chhattisgarh
Uttarakhand
NCT of Delhi
Jammu & Kashmir
Himachal Pradesh
Meghalaya
Tripura
Manipur
Goa
Nagaland
Mizoram
Puducherry
Arunachal Pradesh
Chandigarh
Sikkim
Andaman & Nicobar Island
Lakshadweep
Daman & Diu
Dadara & Nagar Haveli

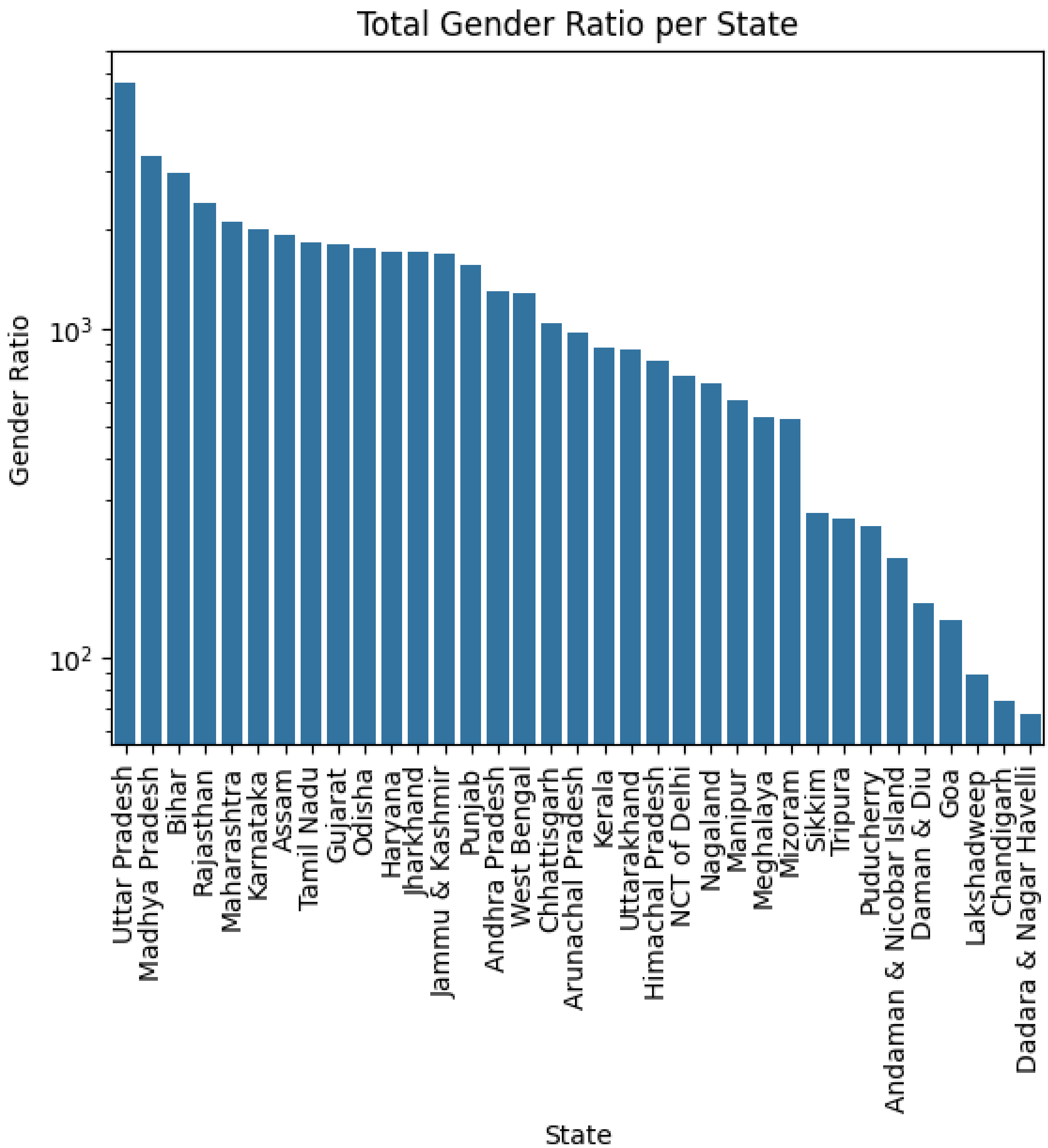


Total Female Population in the age group 0-6 per State
F_06
10^6
10^5
10^4
10^3
State
Uttar Pradesh
Bihar
Maharashtra
West Bengal
Kerala
Karnataka
Andhra Pradesh
Tamil Nadu
Rajasthan
Madhya Pradesh
Gujarat
Jharkhand
Assam
Odisha
Punjab
Haryana
Chhattisgarh
Uttarakhand
NCT of Delhi
Jammu & Kashmir
Himachal Pradesh
Meghalaya
Tripura
Manipur
Goa
Mizoram
Nagaland
Arunachal Pradesh
Puducherry
Chandigarh
Sikkim
Andaman & Nicobar Island
Daman & Diu
Lakshadweep
Dadara & Nagar Havelli



For further analysis, three more columns have been added to the dataset - total male population including total males and males from 0 to 6 years of age, total female population including total females and females from 0 to 6 years of age and gender ratio ([total male population/total female population] x 100).

```
pc['Total Male Population'] = pc['TOT_M'] + pc['M_06']
pc['Total Female Population'] = pc['TOT_F'] + pc['F_06']
pc['Gender Ratio'] = (pc['Total Male Population'] / pc['Total Female Population'])*100
```



- The state with the highest gender ratio is - Uttar Pradesh.
- The state with the lowest gender ratio is - Dadara and Nagar Havelli.

Groupby function has been used to find out the districts with highest and lowest gender ratios:

- The district with the highest gender ratio is - District code 2: Badgam in Jammu and Kashmir.
- The district with the lowest gender ratio is - District code 547: Krishna in Andhra Pradesh.

Problem 2.2 - Data Preprocessing

- Check for and treat (if needed) missing values - Check for and treat (if needed) data irregularities - Scale the Data using the z-score method - Visualize the data before and after scaling and comment on the impact on outliers.

Solution:

Checking missing values:

```
State Code                0
Dist.Code                 0
State                     0
Area Name                 0
No_HH                     0
                         ..
NON_WORK_M                0
NON_WORK_F                0
Total Male Population     0
Total Female Population   0
Gender Ratio              0
Length: 64, dtype: int64
```

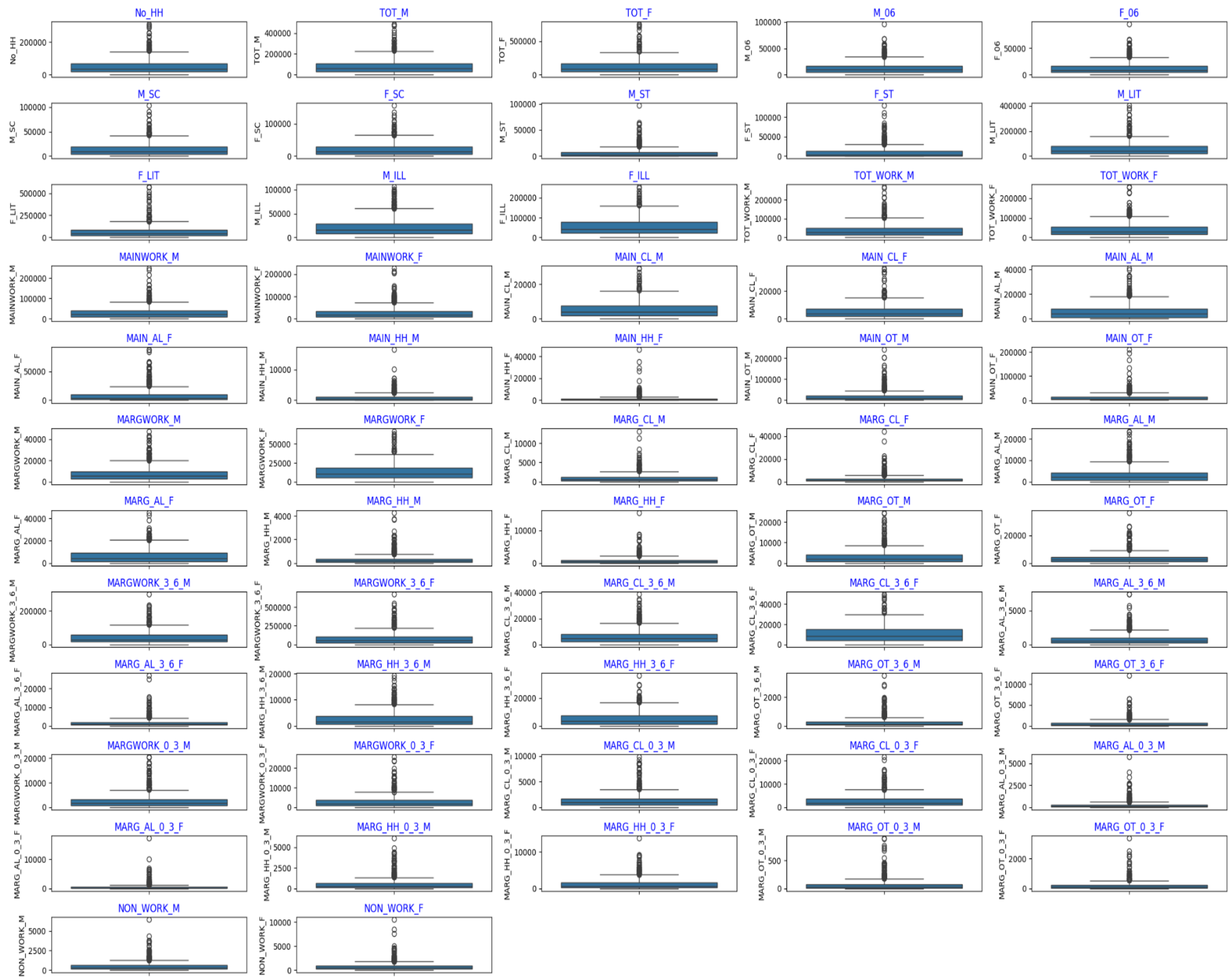
Checking duplicate values:

```
pc.duplicated().sum()
```

```
0
```

There are no data irregularities in the dataset as can be seen from the above executions.

- Visualizing data before scaling:



Outlier treatment:

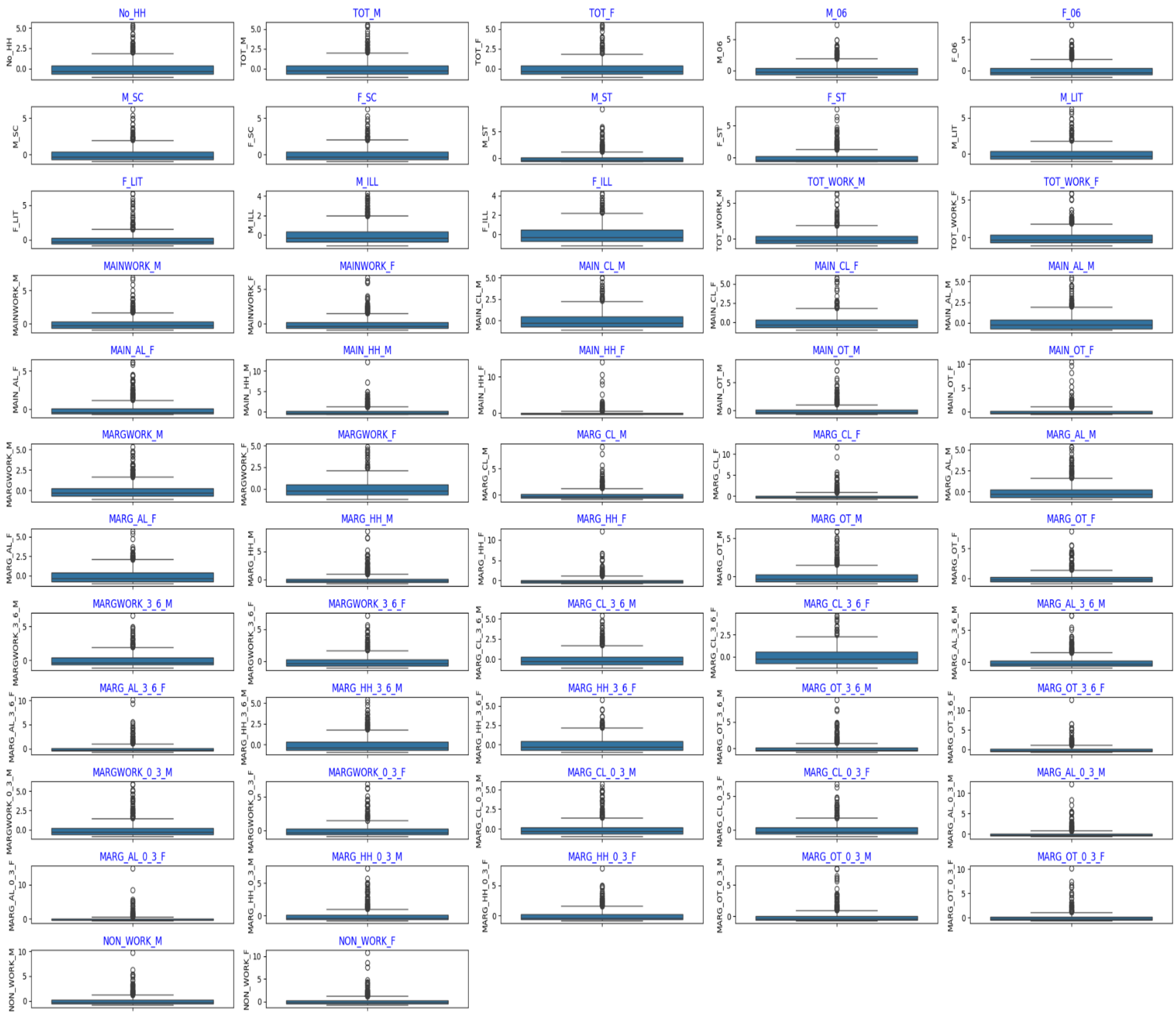
Outliers treatment is not necessary unless they are the result of a processing mistake or wrong measurement. Therefore, we will not treat outliers. True outliers must be kept in the data.

Scaling:

We scale the data by using zscore from scipy.stats on the numerical variables of the dataset. Given below are the first five rows of the scaled data:

	No_HH	TOT_M	TOT_F	M_06	F_06	M_SC	F_SC	M_ST	F_ST	M_LIT	...	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	-0.904738	-0.771236	-0.815563	-0.561012	-0.507738	-0.958575	-0.957049	-0.423306	-0.476423	-0.798097	...	-0.163229	-0.720610	-0.156494	-0.287524	0.156577	-0.657412	-0.365258	-0.499977	-0.413053	-0.539614
1	-0.935695	-0.823100	-0.874534	-0.681096	-0.725367	-0.958297	-0.956772	-0.582014	-0.607607	-0.849434	...	-0.583103	-0.732811	-0.282327	-0.294688	-0.491731	-0.723062	0.042855	-0.073481	-0.606455	-0.598988
2	-0.972412	-1.000919	-0.981466	-0.976956	-0.965262	-0.958575	-0.956772	-0.038951	-0.027273	-0.956457	...	-0.859212	-0.921931	-0.456727	-0.420050	-0.731894	-0.795026	-0.662068	-0.635680	-0.726103	-0.707839
3	-1.037530	-1.052224	-1.041001	-1.022118	-0.995393	-0.958783	-0.957049	-0.355965	-0.390060	-1.004643	...	-0.805468	-0.900758	-0.419198	-0.385127	-0.718770	-0.784926	-0.624966	-0.616294	-0.645791	-0.710038
4	-0.822676	-0.809381	-0.813933	-0.622359	-0.649908	-0.957395	-0.955529	0.149238	0.043330	-0.800568	...	-0.348645	-0.297513	0.472670	0.434200	-0.466796	-0.625849	-0.439461	-0.309346	-0.540895	-0.249344

- Visualizing data after scaling:



So, we can clearly see from above figures that scaling has no impact on outliers.

Problem 2.3 - PCA

- 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.

Solution:

For performing PCA we first test the necessary assumptions.

We conduct the `bartlett_sphericity` test. We use factor_analyzer to import the above-mentioned function.

```
from factor_analyzer.factor_analyzer import calculate_bartlett_sphericity
chi_square_value,p_value=calculate_bartlett_sphericity(scale)
p_value
```

```
0.0
```

The resultant score shows the dataset qualifies for PCA.

Next, we check the adequacy of the sample size using kmo from factor_analyzer.

```
from factor_analyzer.factor_analyzer import calculate_kmo
kmo_all,kmo_model=calculate_kmo(scale)
kmo_model
```

```
0.8039889932781807
```

The kmo score > 0.7 is considered a good sample size for PCA.

We import pca from skelarn.decomposition to conduct PCA. We keep the number of components as 12.

```
from sklearn.decomposition import PCA
pca = PCA(n_components=12, random_state=123)
pca_transformed = pca.fit_transform(scale)
```

Covariance Matrix:

Given below are 5 rows:

```
array([[ 3.18135647e+01,  0.00000000e+00,  3.55827348e-16,
        -1.06748204e-15, -6.67176278e-16, -5.33741022e-16,
        -8.89568370e-17,  0.00000000e+00, -8.89568370e-17,
         1.77913674e-16,  2.38376524e-16, -8.89568370e-17],
       [ 0.00000000e+00,  7.86942415e+00,  1.33435256e-15,
        -5.33741022e-16,  1.55674465e-16,  1.77913674e-16,
         8.89568370e-17, -8.89568370e-17,  1.11196046e-16,
        -2.22392093e-17, -3.03356714e-16, -1.11196046e-16],
       [ 3.55827348e-16,  1.33435256e-15,  4.15340812e+00,
         3.20244613e-15, -5.33741022e-16, -5.78219441e-16,
        -3.89186162e-16, -4.44784185e-17, -4.44784185e-17,
         1.33435256e-16,  1.33435256e-16,  2.77990116e-17],
       [-1.06748204e-15, -5.33741022e-16,  3.20244613e-15,
         3.66879058e+00,  4.89262604e-16, -4.89262604e-16,
         1.55674465e-16,  2.22392093e-17,  0.00000000e+00,
        -2.77990116e-17, -1.77913674e-16,  1.33435256e-16],
       [-6.67176278e-16,  1.55674465e-16, -5.33741022e-16,
         4.89262604e-16,  2.20652588e+00, -1.86809358e-15,
        -9.50726196e-16, -6.11578255e-17,  6.11578255e-17,
        -6.11578255e-17, -1.05636244e-16,  1.11196046e-17],
```

Eigen Vectors:

Given below are the initial values.

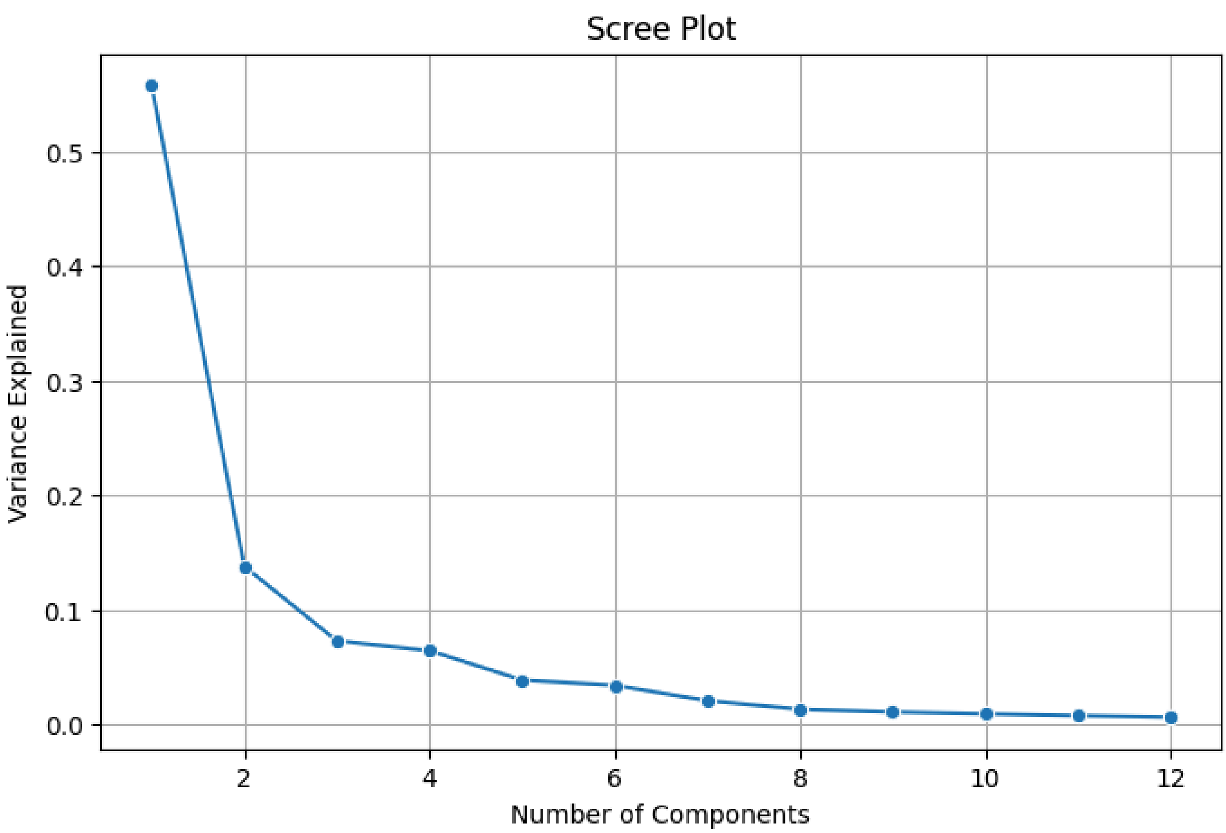
```
array([[ 1.56020579e-01,  1.67117635e-01,  1.65553179e-01,
         1.62192948e-01,  1.62566396e-01,  1.51357849e-01,
         1.51566500e-01,  2.72341946e-02,  2.81833150e-02,
         1.61992837e-01,  1.46872680e-01,  1.61749445e-01,
         1.65248187e-01,  1.59871988e-01,  1.45935804e-01,
         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],

Eigen values:

```
array([31.81356474,  7.86942415,  4.15340812,  3.66879058,  2.20652588,
        1.93827502,  1.17617374,  0.75115909,  0.61705374,  0.52830089,
        0.42983119,  0.3534402 ])
```

Scree Plot:



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

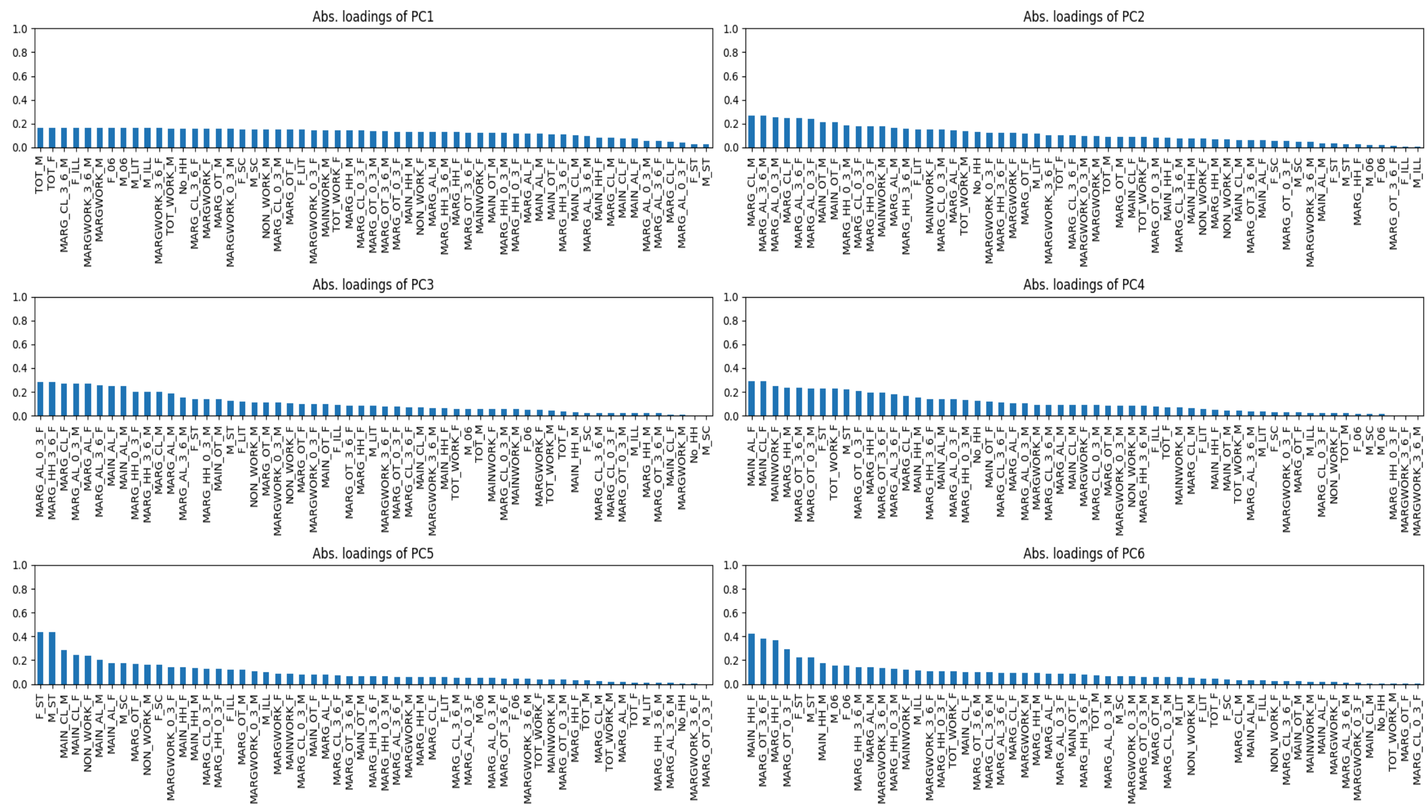
Figure be1ow Showing Cumulative explained variance:

```
np.cumsum(pca.explained_variance_ratio_)
```

```
array([0.55726063, 0.69510499, 0.76785794, 0.83212212, 0.87077261,
       0.9047243 , 0.92532669, 0.93848433, 0.94929292, 0.95854687,
       0.96607599, 0.97226701])
```

Since, we have to take at least 90% explained variance for the scope of this project, we finalize 6 principal components as the optimum number of components.

Original features that influence various PC's:



Heatmap showing Comparison of how the original features influence various Principal Components :

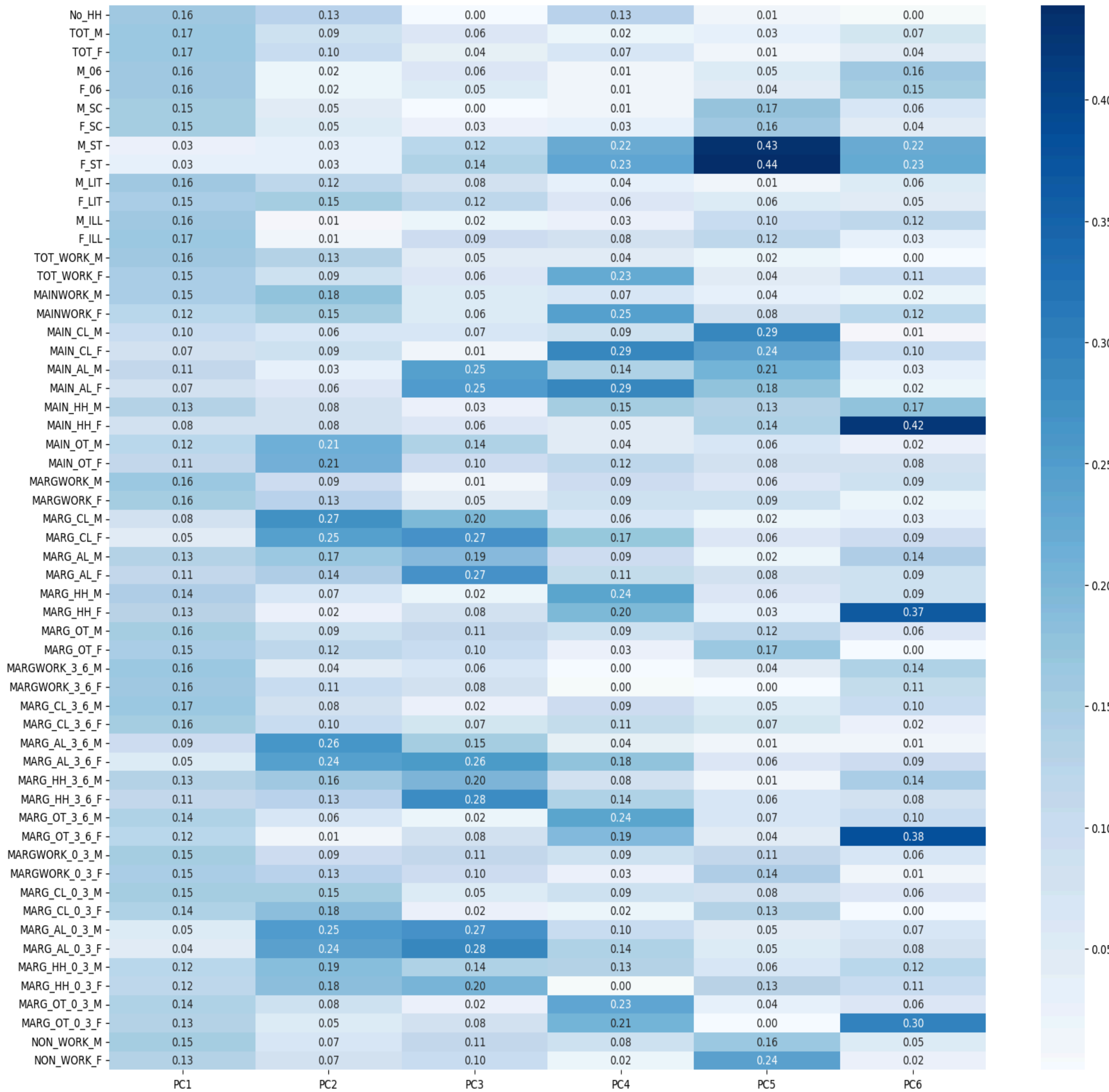
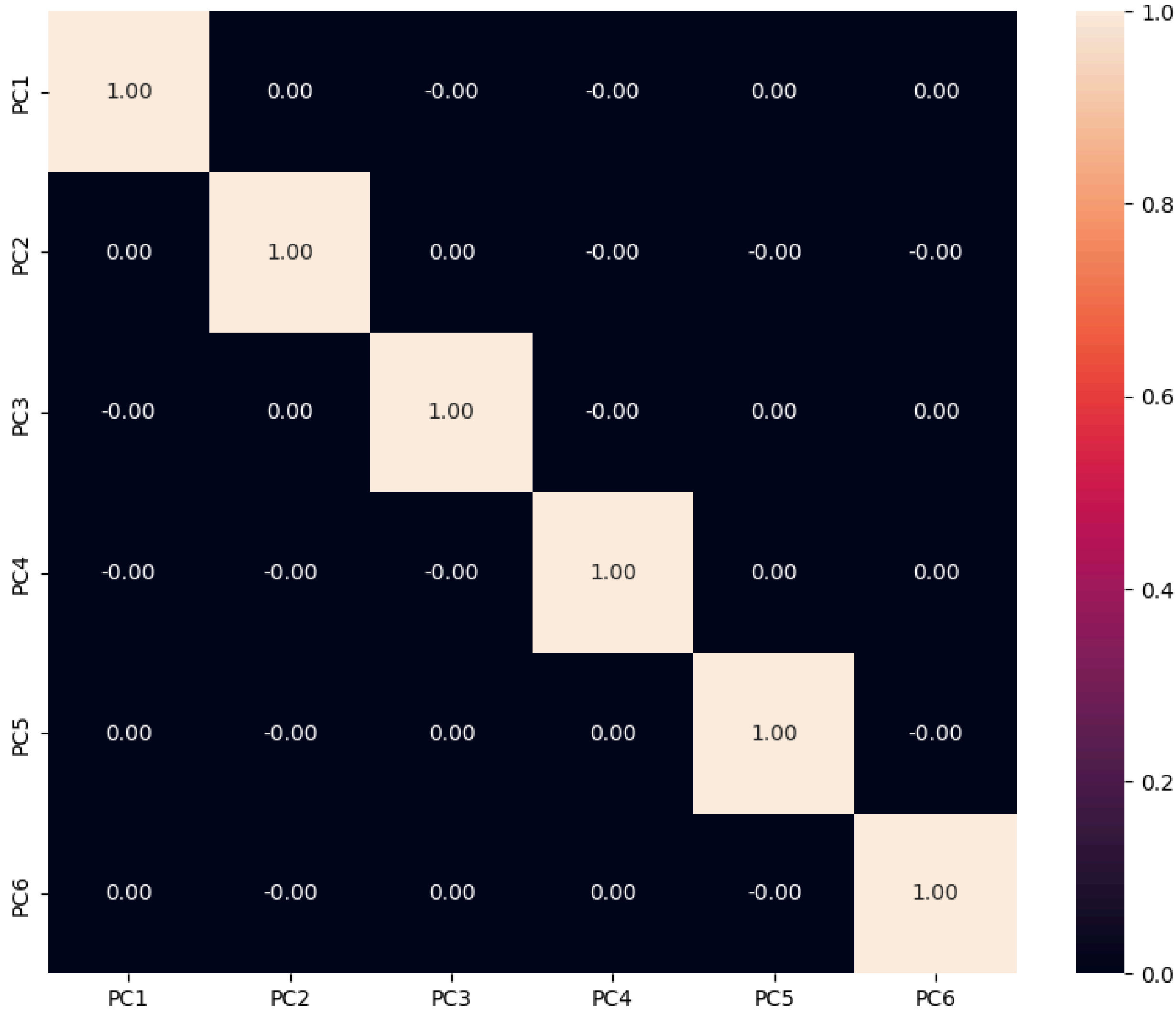


Figure showing correlation between all Principal Components:



Linear equation for first PC :

PC1 = a1x1 + a2x2 + a3X3 +a4X4 + ... + a57x57