DATA MINING PROJECT

JAYA PREETHI R M

TABLE OF CONTENT - DIAGRAM & TABLE

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.

Perform the following in given order:

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.
2. Treat missing values in CPC, CTR and CPM using the formula given. You may refer to the Bank_KMeans Solution File 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.
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. (As an analyst your judgement may be different from another analyst).

4. Perform z-score scaling and discuss how it affects the speed of the algorithm.
5. Perform clustering and do the following: Perform Hierarchical by constructing a Dendrogram using WARD and Euclidean distance.
6. Make Elbow plot (up to n=10) and identify optimum number of clusters for k-means algorithm.
7. Print silhouette scores for up to 10 clusters and identify optimum number of clusters.
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.]
9. Conclude the project by providing summary of your learnings.

ANSWER

1. Preliminary analysis gives us several important information about the given dataset. They are as follows:

- The shape of the dataset shows us that there are 23,066 entries and 19 variables in total, i.e. 23,066 rows and 19 columns in total.

```
df.shape
```

```
(23066, 19)
```

- The head of the dataset shows us that there are several variables recorded to measure the nature and the dimensions of the advertisement (like ad length and ad size), the response variables (like number of clicks and revenue from the ad) and KPIs (like CPC and CPM.

- Below given table shows the statistical summary of the variables:

	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
count	23066.000000	23066.000000	23066.000000	2.306600e+04	2.306600e+04	2.306600e+04	23066.000000	23066.000000	23066.000000	23066.000000	18330.000000	18330.000000	18330.000000
mean	385.163097	337.896037	96674.468048	2.432044e+06	1.295099e+06	1.241520e+06	10678.518816	2706.625689	0.335123	1924.252331	0.073661	7.672045	0.351061
std	233.651434	203.092885	61538.329557	4.742888e+06	2.512970e+06	2.429400e+06	17353.409363	4067.927273	0.031963	3105.238410	0.075160	6.481391	0.343334
min	120.000000	70.000000	33600.000000	1.000000e+00	1.000000e+00	1.000000e+00	1.000000	0.000000	0.210000	0.000000	0.000100	0.000000	0.000000
25%	120.000000	250.000000	72000.000000	3.367225e+04	1.828250e+04	7.990500e+03	710.000000	85.180000	0.330000	55.365375	0.002600	1.710000	0.090000
50%	300.000000	300.000000	72000.000000	4.837710e+05	2.580875e+05	2.252900e+05	4425.000000	1425.125000	0.350000	926.335000	0.082550	7.660000	0.160000
75%	720.000000	600.000000	84000.000000	2.527712e+06	1.180700e+06	1.112428e+06	12793.750000	3121.400000	0.350000	2091.338150	0.130000	12.510000	0.570000
max	728.000000	600.000000	216000.000000	2.759286e+07	1.470202e+07	1.419477e+07	143049.000000	26931.870000	0.350000	21276.180000	1.000000	81.560000	7.260000

- Below given table shows us the datatype and total number of entries of the variables.

```
df.info()

<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
```

We can see that there are some categorical variables under object data type and continuous variables under integer data type variables.

- From the below given output , we can see that there are certain null variables in CPC, CTR and CPM.

```
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 no duplicated entries in this particular dataset

```
[ ]  df.duplicated().sum()

     0
```

2. To treat the null variables in the KPI variables I have created a user defined function for each variable based on the formula for each variable.

For CTR, i.e., click through rate, I created a user defined function based on the formula:

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

For CPM, i.e., cost per 1000 impressions, I created a user defined function based on the formula:

CPC = Total Cost (spend) / Number of Clicks

For CPC, i.e., cost per click, I created a user defined function based on the formula:

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

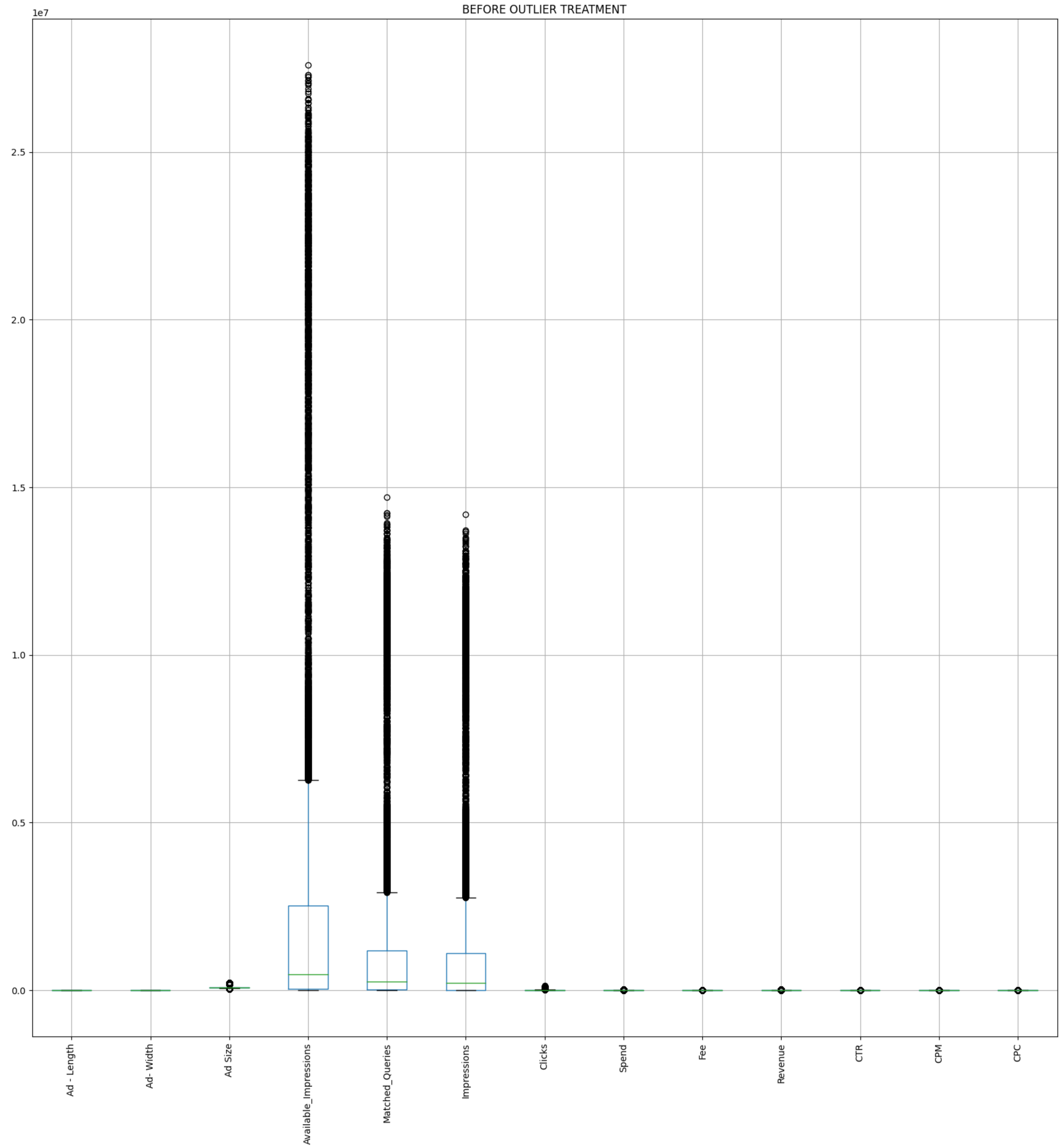
Consequently, I was able to impute the null values by applying these functions in place of null values to calculate the KPIs respectively.

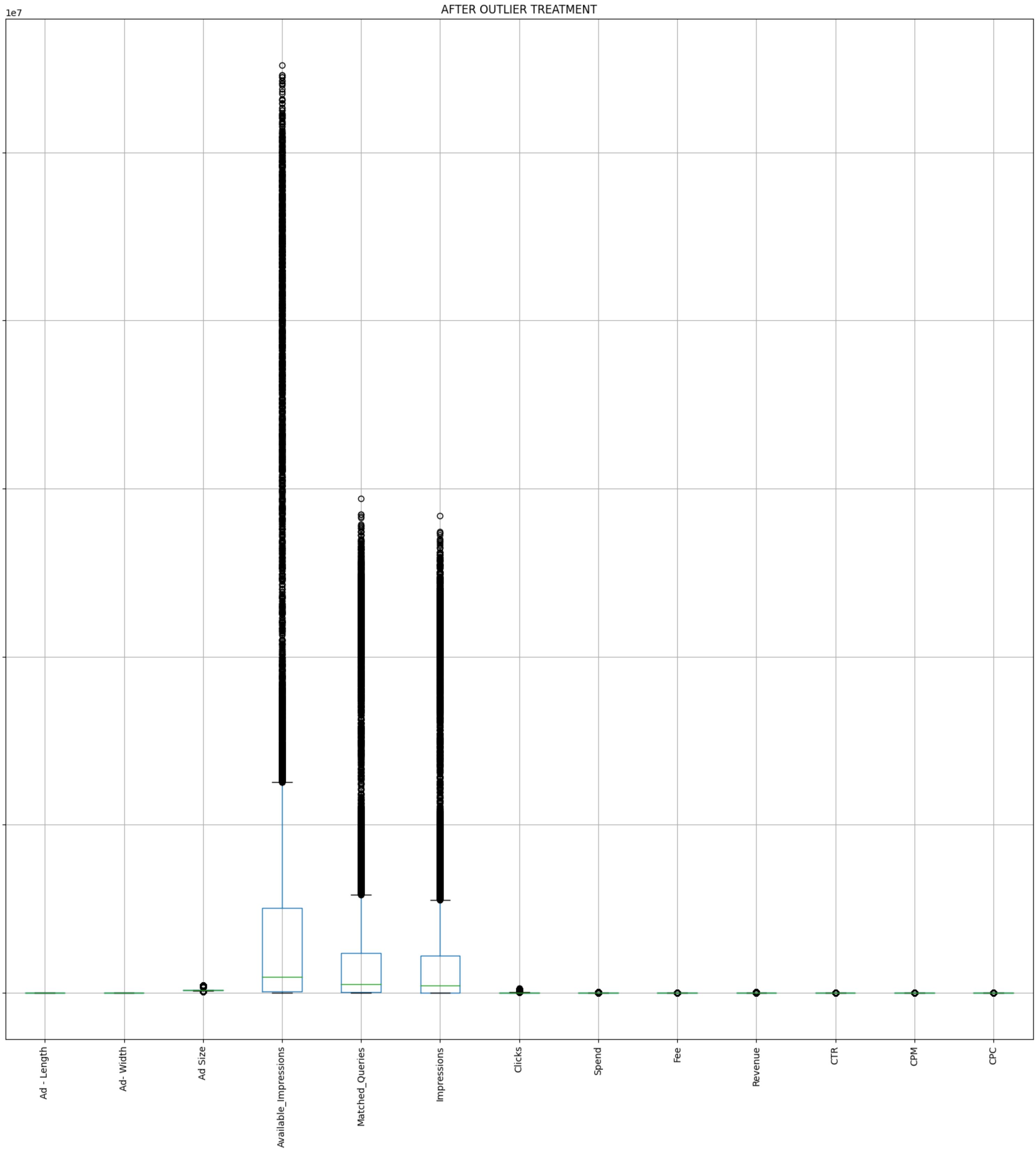
```
[ ]  df.isnull().sum()

     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                     0
     CPM                     0
     CPC                     0
     dtype: int64
```

3. The boxplot diagram given below shows us that there are outliers in the variables: Available_impressions, Matched_queries, and Impressions. To answer the question whether treating outliers is important for K-means clustering, Although treating outliers can transform the true nature of the data, I believe that treating outliers is important for the clustering process. This is because K-means clustering will be influenced by the presence of outliers thus affecting the final clustering. To provide actionable insights from our analysis, I believe that treating outliers is important in our case.

   Resultantly, I created a user_defined function called treat_outliers to utilize the IQR method of outlier treatment. I have attached the boxplot of the variables before and after outlier treatment for your reference.


BEFORE OUTLIER TREATMENT
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Ad - Length
Ad- Width
Ad Size
Available_Impressions
Matched_Queries
Impressions
Clicks
Spend
Fee
Revenue
CTR
CPM
CPC

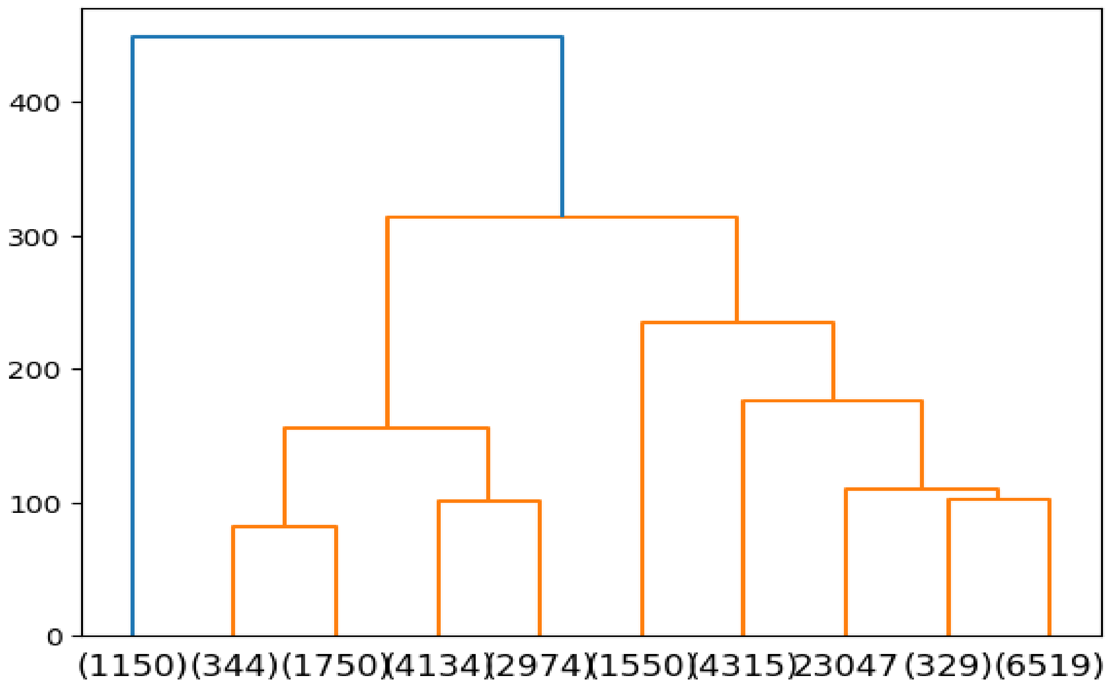

AFTER OUTLIER TREATMENT
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Ad - Length
Ad- Width
Ad Size
Available_Impressions
Matched_Queries
Impressions
Clicks
Spend
Fee
Revenue
CTR
CPM
CPC

4. Using the Standardscaler function from Sklearn, I was able to z-score scale the dataset.

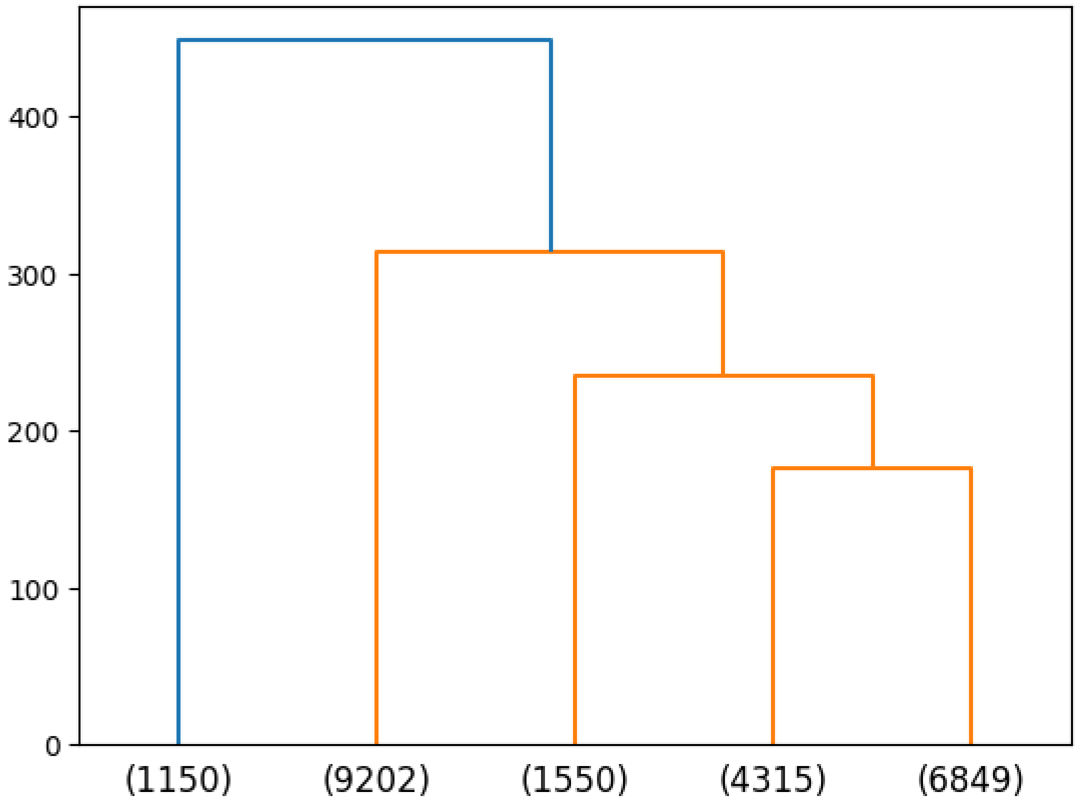
	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
0	-0.432797	-0.359227	-0.569484	-0.567061	-0.563943	-0.719779	-0.722776	0.487214	-0.676118	-0.978783	-1.220485	-1.081139
1	-0.432797	-0.359227	-0.569490	-0.567076	-0.563958	-0.719779	-0.722776	0.487214	-0.676118	-0.973764	-1.220485	-1.081139
2	-0.432797	-0.359227	-0.569269	-0.567049	-0.563931	-0.719779	-0.722776	0.487214	-0.676118	-0.982548	-1.220485	-1.081139
3	-0.432797	-0.359227	-0.569339	-0.566994	-0.563875	-0.719779	-0.722776	0.487214	-0.676118	-0.992587	-1.220485	-1.081139
4	-0.432797	-0.359227	-0.569622	-0.567093	-0.563975	-0.719779	-0.722776	0.487214	-0.676118	-0.966235	-1.220485	-1.081139
...	...	...	...	...	...	...	...	...	...	...	...	...
23061	-0.186599	1.871803	-0.569906	-0.567185	-0.564071	-0.719779	-0.722756	0.487214	-0.676102	1.966364	1.837274	-0.846532
23062	-0.186599	1.871803	-0.569905	-0.567185	-0.564071	-0.719779	-0.722765	0.487214	-0.676109	1.966364	1.780997	-0.947078
23063	-0.186599	1.871803	-0.569905	-0.567185	-0.564071	-0.719779	-0.722762	0.487214	-0.676107	1.966364	1.837274	-0.913563
23064	1.290590	-0.406696	-0.569904	-0.567185	-0.564071	-0.719779	-0.722756	0.487214	-0.676102	1.966364	1.837274	-0.846532
23065	-0.186599	1.871803	-0.569905	-0.567185	-0.564071	-0.719779	-0.722751	0.487214	-0.676098	1.966364	1.837274	-0.779501

As for the speed of the algorithm, the coding process was much smoother and more efficient. The speed of the algorithm was much faster than before due to closer proximity of the variable's values.

5. Initially, I used the Ward method in dendrogram to construct a dendrogram diagram with 10 clusters. I specifically created 10 clusters using truncate_mode function due to the sheer size of the dataset and for effective analysis.

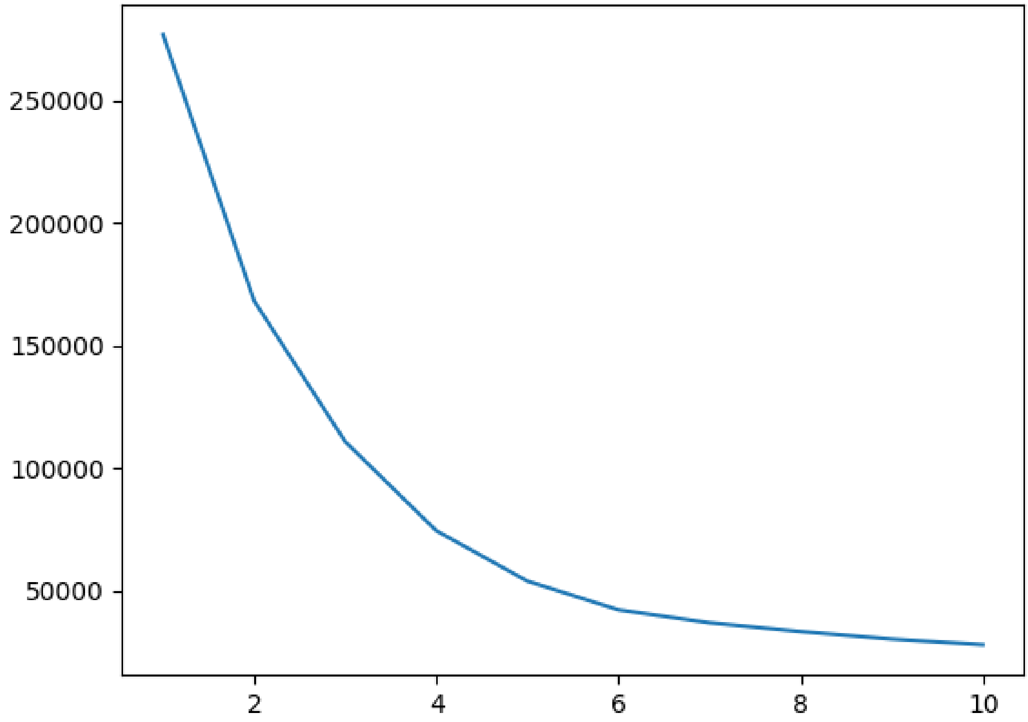


From the diagram, I decided to form a hierarchical cluster using 4 numbers of clusters, since I believed that 4 numbers of clusters would be the most efficient number of clusters. Given the size of the dataset anything less than 4 will not be representative of the dataset and anything above 4 will make it too convoluted to analyse. Thus, by using an agglomerative clustering method with euclidean distance, I clustered the dataset into 4 clusters. I have added the dendrogram representation of the clustering process and the profile of agglomerative clustering below.



Agglo_CLusters	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC	Freq
0	385.14925	337.876594	96661.57662	2.432460e+06	1.295320e+06	1.241732e+06	10680.370436	2707.092715	0.335121	1924.584507	8.40056	8.360727	0.33613	23062
1	510.00000	450.000000	198000.00000	6.201050e+04	4.036650e+04	3.558200e+04	4.500000	27.185000	0.350000	17.670000	0.01500	0.755000	5.88500	2
2	720.00000	300.000000	216000.00000	8.000000e+00	2.000000e+00	1.000000e+00	2.000000	0.150000	0.350000	0.097500	200.00000	150.000000	0.08000	1
3	120.00000	600.000000	72000.00000	2.000000e+00	2.000000e+00	2.000000e+00	1.000000	1.430000	0.350000	0.929500	50.00000	715.000000	1.43000	1

6. The below given diagram shows us the elbow plot for the model with 10 clusters.



The elbow plot shows us the within-cluster sum of squares, with each cluster. We shall see that from the 6 cluster model the WSS tends to stay the same, as in, there is no change in the graph. Since, there are no significant changes after 6 th level, clustering of the dataset beyond point 6 will unnecessarily hamper the representation of the model. Thus, the 6 cluster model is the ideal number of clusters for the given dataset.

7. The silhouette score shows us the measure of similarity within one's own cluster and other clusters. It essentially shows us the goodness of the clustering model based on the number of clusters. A good clustering model must have the highest silhouette score. Below given are the silhouette scores for clustering models from 3 clusters upto 10 clusters.

NUMBER OF CLUSTERS	SILHOUETTE SCORE
3	0.4265
4	0.4897
5	0.5287

6	0.5436
7	0.5038
8	0.4518
9	0.4522
10	0.4539

From the above given table, we can see that the silhouette score for the clustering model with 6 numbers of clusters has the highest silhouette score with 0.5436. We must also notice that the silhouette score gradually increases to the 6 cluster point and starts to decline after that point. Therefore, we can conclude that the optimum number of clusters for the given dataset is 6.

8. I have decided to cluster the dataset into 6 clusters as suggested by the WSS plot and the silhouette score as the optimum number of clusters for the model. After the clustering of the data, I have profiled the data accordingly. Following are the results:

clusters	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC	Freq
1	141.835595	572.067039	75715.881883	8.787287e+05	6.207391e+05	5.238120e+05	71556.263368	7645.819425	0.278819	5532.487799	13.774270	15.215491	0.110383	1253
2	679.560265	118.556291	70230.834437	1.797533e+07	9.595386e+06	9.239259e+06	17537.707285	15474.630285	0.238715	11843.556259	0.188252	1.708675	0.916934	1510
3	320.673516	252.905251	78264.794521	6.613046e+06	3.697269e+06	3.616010e+06	8535.450342	4875.173664	0.318219	3331.590484	0.235656	1.371872	0.598196	1752
4	420.155594	148.331876	53891.870629	2.082522e+06	1.025575e+06	9.856740e+05	3478.984848	1774.309592	0.346841	1165.813151	0.385980	1.785280	0.573515	6864
5	146.660504	556.791412	73480.095423	4.711609e+04	2.900135e+04	2.151616e+04	2981.553004	322.788071	0.349663	211.021298	15.874635	14.630951	0.101810	6707
6	652.790361	341.857430	206648.192771	3.327757e+05	1.795705e+05	1.574390e+05	14324.827711	1325.104373	0.348735	866.349248	13.435120	11.897394	0.115408	4980

9. As my concluding remarks, I would like to state my findings from the above given profile of the clustered model.
    - The 1st cluster and the 5th cluster have the lowest ad length but highest ad width. The 5th cluster has the highest click through rate followed by the 1st cluster. Thus, we can say that low ad dimensions can result in high click through rates.
    - The second cluster has the highest revenue turnover from the advertisement and also has the highest matched_queries. . However, we must also notice that the second cluster has the highest cost per click.

- The first cluster has the highest cost per impression and Available impressions.
- Interestingly, 5th cluster has the lowest Spend on ad set but highest click through rates.
- There is a wide gap in ad performance between the clusters. Where cluster 1, 5 and 6 have done significantly better than cluster 2, 3 and 4.

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.

1. Part 2 - PCA: Read the data and perform basic checks like checking head, info, summary, nulls, and duplicates, etc.
2. Part 2 - PCA: Perform detailed Exploratory analysis by creating certain questions like (i) Which state has highest gender ratio and which has the lowest? (ii) Which district has the highest & lowest gender ratio? (Example Questions). 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
3. Part 2 - PCA: We choose not to treat outliers for this case. Do you think that treating outliers for this case is necessary?
4. Part 2 - PCA: Scale the Data using z-score method. Does scaling have any impact on outliers? Compare box plots before and after scaling and comment
5. Part 2 - PCA: Perform all the required steps for PCA (use sklearn only) Create the covariance Matrix Get eigen values and eigen vector.
6. Part 2 - PCA: Identify the optimum number of PCs (for this project, take at least 90% explained variance). Show Scree plot.
7. Part 2 - PCA: Compare PCs with Actual Columns and identify which is explaining most variance. Write inferences about all the Principal components in terms of actual variables.
8. Part 2 - PCA: Write linear equation for first PC.

ANSWER

1. Below, I have attached the result of the preliminary checks on the given data.

- The head of the dataset shows us the census data; the female and male counts are subsections based on the nature of the population. There are totally 640 entries and 61 variables, i.e, there are 61 columns and 640 rows.

	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_
0	1	1	Jammu & Kashmir	Kupwara	7707	23388	29796	5862	6196	3	...	1150	749	180	237	680	252	32	
1	1	2	Jammu & Kashmir	Badgam	6218	19585	23102	4482	3733	7	...	525	715	123	229	186	148	76	
2	1	3	Jammu & Kashmir	Leh(Ladakh)	4452	6546	10964	1082	1018	3	...	114	188	44	89	3	34	0	
3	1	4	Jammu & Kashmir	Kargil	1320	2784	4206	563	677	0	...	194	247	61	128	13	50	4	
4	1	5	Jammu & Kashmir	Punch	11654	20591	29981	5157	4587	20	...	874	1928	465	1043	205	302	24	

- Below given table shows us the datatypes of the variables.

```
<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
```

```
 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
```

Almost all the variables are continuous variables with integer type except for state and area name variables, which are categorical, object type variables.

- Below given is the statistical summary of the given dataset

	count	mean	std	min	25%	50%	75%	max
State Code	640.0	17.11	9.43	1.0	9.00	18.0	24.00	35.0
Dist.Code	640.0	320.50	184.90	1.0	160.75	320.5	480.25	640.0
No_HH	640.0	51222.87	48135.41	350.0	19484.00	35837.0	68892.00	310450.0
TOT_M	640.0	79940.58	73384.51	391.0	30228.00	58339.0	107918.50	485417.0
TOT_F	640.0	122372.08	113600.72	698.0	46517.75	87724.5	164251.75	750392.0
M_06	640.0	12309.10	11500.91	56.0	4733.75	9159.0	16520.25	96223.0
F_06	640.0	11942.30	11326.29	56.0	4672.25	8663.0	15902.25	95129.0
M_SC	640.0	13820.95	14426.37	0.0	3466.25	9591.5	19429.75	103307.0
F_SC	640.0	20778.39	21727.89	0.0	5603.25	13709.0	29180.00	156429.0
M_ST	640.0	6191.81	9912.67	0.0	293.75	2333.5	7658.00	96785.0
F_ST	640.0	10155.64	15875.70	0.0	429.50	3834.5	12480.25	130119.0
M_LIT	640.0	57967.98	55910.28	286.0	21298.00	42693.5	77989.50	403261.0
F_LIT	640.0	66359.57	75037.86	371.0	20932.00	43796.5	84799.75	571140.0
M_ILL	640.0	21972.60	19825.61	105.0	8590.00	15767.5	29512.50	105961.0
F_ILL	640.0	56012.52	47116.69	327.0	22367.00	42386.0	78471.00	254160.0
TOT_WORK_M	640.0	37992.41	36419.54	100.0	13753.50	27936.5	50226.75	269422.0
TOT_WORK_F	640.0	41295.76	37192.36	357.0	16097.75	30588.5	53234.25	257848.0
MAINWORK_M	640.0	30204.45	31480.92	65.0	9787.00	21250.5	40119.00	247911.0
MAINWORK_F	640.0	28198.85	29998.26	240.0	9502.25	18484.0	35063.25	226166.0
MAIN_CL_M	640.0	5424.34	4739.16	0.0	2023.50	4160.5	7695.00	29113.0
MAIN_CL_F	640.0	5486.04	5326.36	0.0	1920.25	3908.5	7286.25	36193.0
MAIN_AL_M	640.0	5849.11	6399.51	0.0	1070.25	3936.5	8067.25	40843.0
MAIN_AL_F	640.0	8926.00	12864.29	0.0	1408.75	3933.5	10617.50	87945.0
MAIN_HH_M	640.0	883.89	1278.64	0.0	187.50	498.5	1099.25	16429.0
MAIN_HH_F	640.0	1380.77	3179.41	0.0	248.75	540.5	1435.75	45979.0
MAIN_OT_M	640.0	18047.10	26068.48	36.0	3997.50	9598.0	21249.50	240855.0

MARG_AL_F	640.0	6463.28	6773.88	0.0	1402.50	4020.5	9089.25	45301.0
MARG_HH_M	640.0	316.74	462.66	0.0	71.75	166.0	356.50	4298.0
MARG_HH_F	640.0	786.63	1198.72	0.0	171.75	429.0	962.50	15448.0
MARG_OT_M	640.0	3126.15	3609.39	7.0	935.50	2036.0	3985.25	24728.0
MARG_OT_F	640.0	3539.32	4115.19	19.0	1071.75	2349.5	4400.50	36377.0
MARGWORK_3_6_M	640.0	41948.17	39045.32	291.0	16208.25	30315.0	57218.75	300937.0
MARGWORK_3_6_F	640.0	81076.32	82970.41	341.0	26619.50	56793.0	107924.00	676450.0
MARG_CL_3_6_M	640.0	6394.99	6019.81	27.0	2372.00	4630.0	8167.00	39106.0
MARG_CL_3_6_F	640.0	10339.86	8467.47	85.0	4351.50	8295.0	15102.00	50065.0
MARG_AL_3_6_M	640.0	789.85	905.64	0.0	235.50	480.5	986.00	7426.0
MARG_AL_3_6_F	640.0	1749.58	2496.54	0.0	497.25	985.5	2059.00	27171.0
MARG_HH_3_6_M	640.0	2743.64	3059.59	0.0	718.75	1714.5	3702.25	19343.0
MARG_HH_3_6_F	640.0	5169.85	5335.64	0.0	1113.75	3294.0	7502.25	36253.0
MARG_OT_3_6_M	640.0	245.36	358.73	0.0	58.00	129.5	276.00	3535.0
MARG_OT_3_6_F	640.0	585.88	900.03	0.0	127.75	320.5	719.25	12094.0
MARGWORK_0_3_M	640.0	2616.14	3036.96	7.0	755.00	1681.5	3320.25	20648.0
MARGWORK_0_3_F	640.0	2834.55	3327.84	14.0	833.50	1834.5	3610.50	25844.0
MARG_CL_0_3_M	640.0	1392.97	1489.71	4.0	489.50	949.0	1714.00	9875.0
MARG_CL_0_3_F	640.0	2757.05	2788.78	30.0	957.25	1928.0	3599.75	21611.0
MARG_AL_0_3_M	640.0	250.89	453.34	0.0	47.00	114.5	270.75	5775.0
MARG_AL_0_3_F	640.0	558.10	1117.64	0.0	109.00	247.5	568.75	17153.0
MARG_HH_0_3_M	640.0	560.69	762.58	0.0	136.50	308.0	642.00	6116.0
MARG_HH_0_3_F	640.0	1293.43	1585.38	0.0	298.00	717.0	1710.75	13714.0
MARG_OT_0_3_M	640.0	71.38	107.90	0.0	14.00	35.0	79.00	895.0
MARG_OT_0_3_F	640.0	200.74	309.74	0.0	43.00	113.0	240.00	3354.0
NON_WORK_M	640.0	510.01	610.60	0.0	161.00	326.0	604.50	6456.0
NON_WORK_F	640.0	704.78	910.21	5.0	220.50	464.5	853.50	10533.0

- Lastly, we have several initial checks in a table.

```
[ ]  data.shape

     (640, 61)

[ ]  data.duplicated().value_counts()

     False    640
     dtype: int64

[ ]  data.isnull().sum().sum()

     0
```

2. From the given output, we can see that there are no duplicate entries or null variables in the dataset. However, we must consider that there are minimum values of zeroes in our statistical summary for several variables. This might be due to no representation in the given variable at the selected area of the census. Thus, treating this null variable might affect the results of our analysis.

   For EDA, I have created a separate column for each subset of population to study the gender ratio across the states.

- Total Gender ratio

A separate column for gender ratio among the total population was created where TOT_F / TOT_M = GenderRatio. From the below given table, we can see that Arunachal Pradesh has the highest gender ratio with 1.077. Whereas, Haryana has the lowest gender ratio with 0.86.

```
State
Arunachal Pradesh           1.077129
Dadara & Nagar Havelli      1.041812
Mizoram                     1.029533
Meghalaya                   1.023831
Jharkhand                   1.018963
Bihar                       1.004937
Chhattisgarh                1.004087
Goa                         1.002041
Maharashtra                 0.996049
Puducherry                  0.995456
West Bengal                 0.989665
Assam                       0.989520
Uttar Pradesh               0.984971
Karnataka                   0.977602
Kerala                      0.977270
Daman & Diu                 0.976702
Nagaland                    0.976432
Manipur                     0.974591
Odisha                      0.974122
Andhra Pradesh              0.969294
Sikkim                      0.969226
Madhya Pradesh              0.962490
Gujarat                     0.952710
Tamil Nadu                  0.951862
Tripura                     0.947326
Andaman & Nicobar Island    0.944465
Rajasthan                   0.942213
Jammu & Kashmir             0.937727
Himachal Pradesh            0.931490
Lakshadweep                 0.923211
Uttarakhand                 0.900583
NCT of Delhi                0.886346
Chandigarh                  0.874544
Punjab                      0.874173
Haryana                     0.860116
Name: GR_06, dtype: float64
```

- Gender ratio for literate population

A separate column for gender ratio among the literate population was created where F_LIT / M_LIT = GR_LIT. From the below given table, we can see that Kerala has the highest gender ratio in literate population with 1.665. Whereas, Rajasthan has the lowest gender ratio in literate population with 0.876.

```
State
Kerala                        1.665331
Mizoram                       1.565487
Nagaland                      1.465217
Goa                           1.413201
Tripura                       1.406433
Puducherry                    1.385163
Maharashtra                   1.365704
Andaman & Nicobar Island      1.308977
Arunachal Pradesh             1.307364
Meghalaya                     1.303851
Uttarakhand                   1.302127
Chandigarh                    1.294647
Himachal Pradesh              1.282816
Tamil Nadu                    1.279773
Sikkim                        1.238620
Manipur                       1.220003
Odisha                        1.177696
Daman & Diu                   1.153941
West Bengal                   1.146578
Chhattisgarh                  1.144733
NCT of Delhi                  1.140560
Andhra Pradesh                1.120612
Gujarat                       1.119928
Assam                         1.117179
Karnataka                     1.094448
Punjab                        1.077167
Lakshadweep                   1.069144
Madhya Pradesh                1.050261
Dadara & Nagar Havelli        1.036921
Jharkhand                     0.955202
Jammu & Kashmir               0.954500
Haryana                       0.939747
Uttar Pradesh                 0.898215
Bihar                         0.896294
Rajasthan                     0.876478
Name: GR_LIT, dtype: float64
```

- Gender Ratio for Illiterate population

A separate column for gender ratio among the literate population was created where F_ILL / M_ILL = GR_ILL. From the below given table, we can see that Tamil nadu has a significant gender ratio among the illiterate population with 4.27. Whereas, Meghalaya has the lowest gender ratio among illiterate population with 1.52.

```
State
Tamil Nadu                    4.271965
Andhra Pradesh                4.051520
Chhattisgarh                  3.705274
Odisha                        3.573074
Maharashtra                   3.102779
Karnataka                     3.000637
Dadara & Nagar Havelli        2.964573
Puducherry                    2.957248
Himachal Pradesh              2.880461
Madhya Pradesh                2.852555
Rajasthan                     2.812395
Daman & Diu                   2.767135
Uttarakhand                   2.716193
Gujarat                       2.698594
Andaman & Nicobar Island      2.695249
Arunachal Pradesh             2.643311
Goa                           2.633756
West Bengal                   2.627779
Manipur                       2.594775
Sikkim                        2.572014
Nagaland                      2.508156
Jharkhand                     2.488781
Tripura                       2.464976
Assam                         2.307773
Haryana                       2.301337
Jammu & Kashmir               2.150122
Uttar Pradesh                 2.135951
Punjab                        2.060683
Bihar                         1.989239
Chandigarh                    1.976100
NCT of Delhi                  1.965874
Kerala                        1.831228
Mizoram                       1.572103
Lakshadweep                   1.547255
Meghalaya                     1.523958
Name: GR_ILL, dtype: float64
```

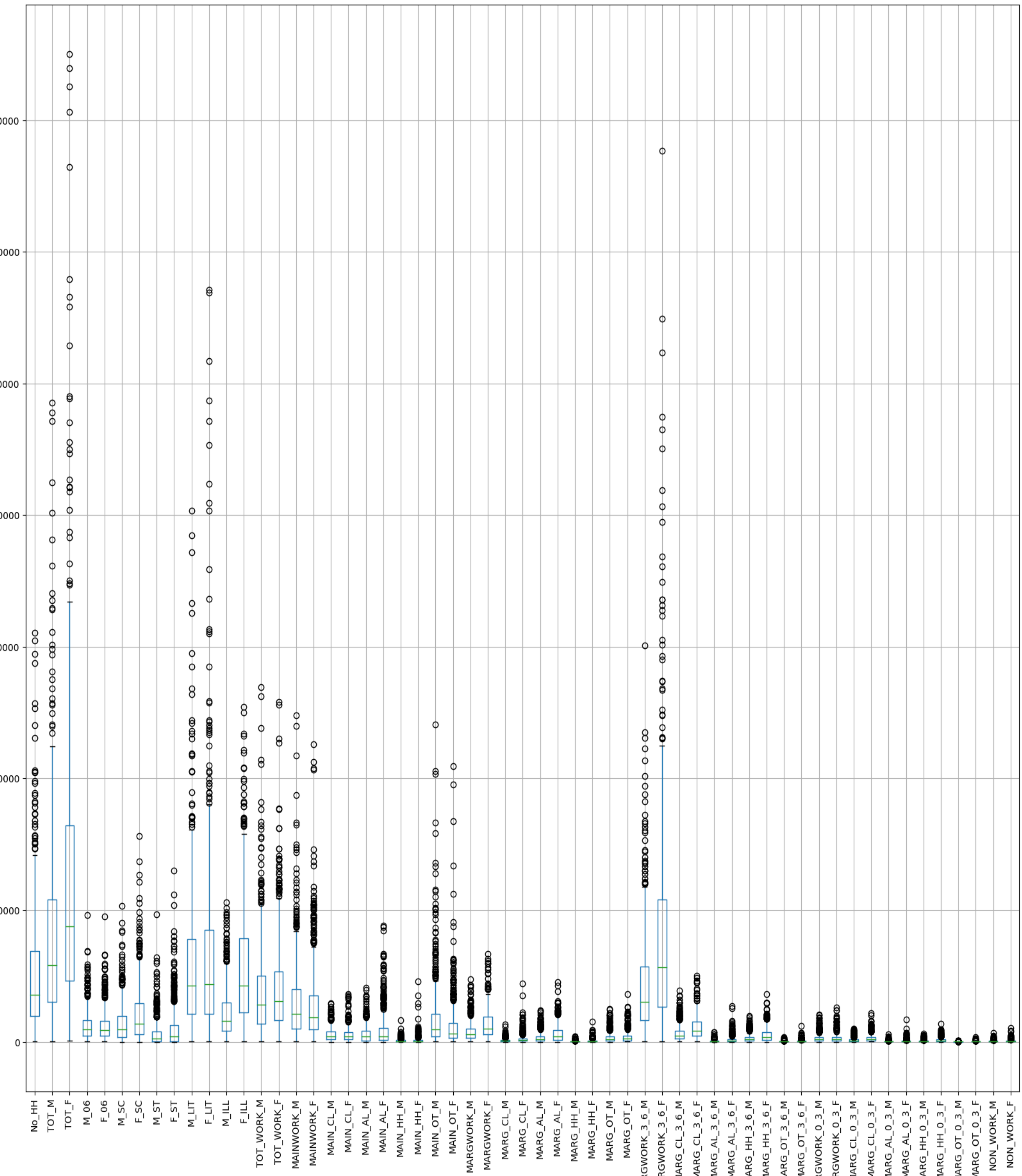
- Gender Ratio for Total worker population

A separate column for gender ratio among the literate population was created where TOT_WORK_F / TOT_WORK_M = TOT_WORK_GR. From the below given table, we can see that Arunachal Pradesh has the highest gender ratio in the total working population with 2.8. Whereas, the lowest gender ratio among the total working population is at Lakshadweep with 0.348.

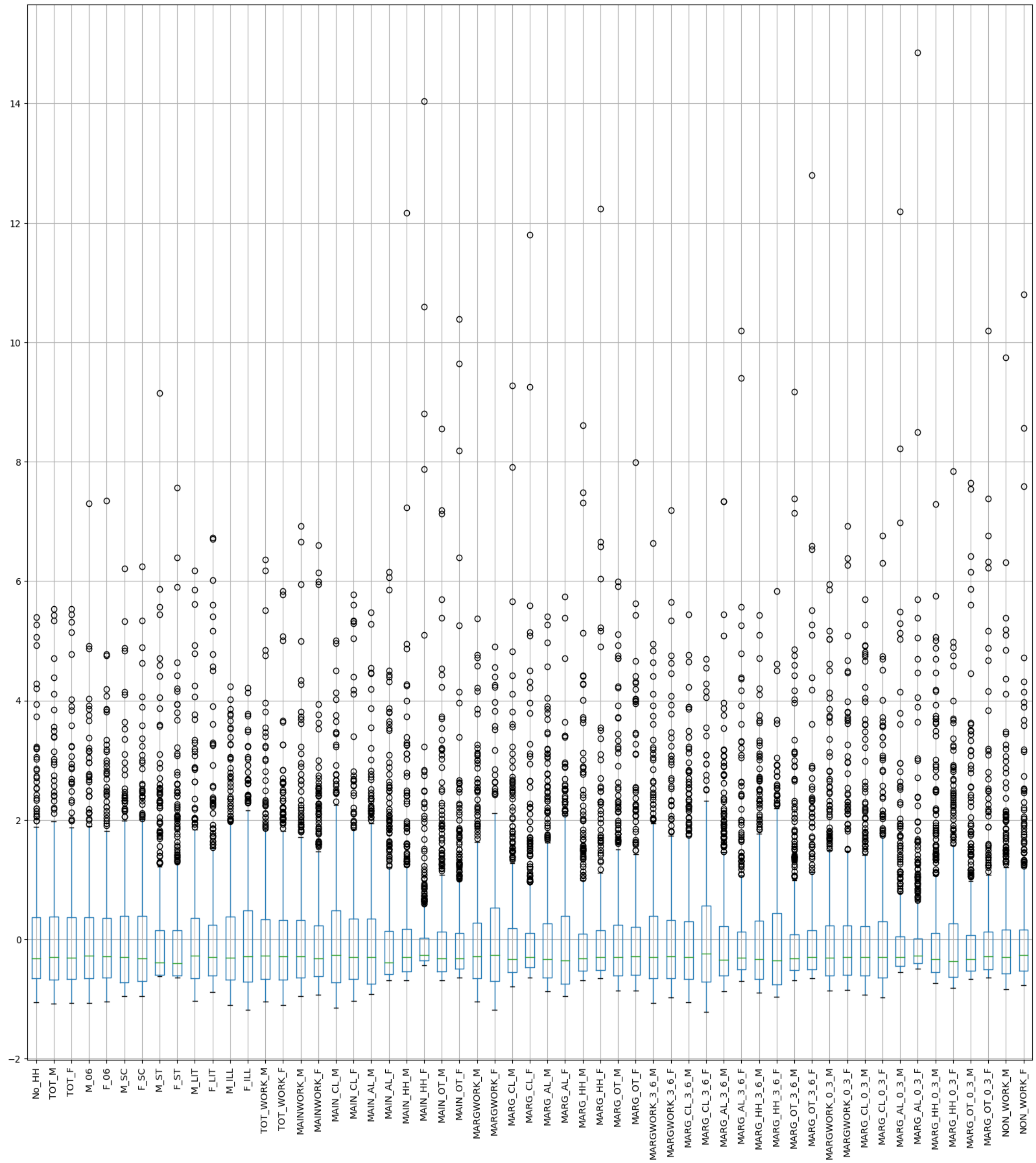
```
State
Arunachal Pradesh             2.800141
Nagaland                      2.447661
Uttarakhand                   2.041131
Chhattisgarh                  2.002111
Manipur                       1.758061
Andhra Pradesh                1.750836
Mizoram                       1.743010
Maharashtra                   1.671265
Himachal Pradesh              1.616604
Odisha                        1.538105
Madhya Pradesh                1.527034
Tamil Nadu                    1.513244
Sikkim                        1.487855
Meghalaya                     1.470992
Jharkhand                     1.443190
Rajasthan                     1.407223
Dadara & Nagar Havelli        1.404079
Tripura                       1.185948
Karnataka                     1.179587
Andaman & Nicobar Island      1.025632
Bihar                         1.011342
Assam                         1.008075
Gujarat                       0.997892
Jammu & Kashmir               0.922714
West Bengal                   0.864583
Uttar Pradesh                 0.822479
Kerala                        0.812395
Goa                           0.803559
Puducherry                    0.782222
Chandigarh                    0.760037
Haryana                       0.656364
Daman & Diu                   0.589655
NCT of Delhi                  0.543726
Punjab                        0.488921
Lakshadweep                   0.347996
Name: TOT_WORK_GR, dtype: float64
```

3. I believe that treating outliers is an essential step in data analysis and unsupervised learning. However, for this specific project, since I will be z-score scaling the data for PCA, I believe that treating outliers might not be necessary. Since, by simply scaling the data we can bring the data points to a proximity. Thus, yielding actionable insights.
4. I have attached the boxplot representation of the variables before and after z-score scaling the dataset.

BEFORE Z-SCORE SCALING



AFTER Z-SCORE SCALING



The boxplots before scaling are scattered, i.e, all the variables have varying boxplots at varying levels. Similarly, some box plots have outliers while some boxplots do not. Whereas , after z-score scaling we can see that the boxplots for all the variables are at a similar level with 0 as their mid-level. The outlier patterns for all boxplots are also similar. Thus, we can say that transforming the data made the variables much more scalable.

5. Prior to performing PCA, we must do certain necessary hypothesis testing.

- Bartlett's test of sphericity tells us whether the variables are correlated or not. Therefore, our hypothesis is:

Ho: All variables in the data are uncorrelated

Ha: At least one pair of variables in the data are correlated

If null hypothesis cannot be rejected, performing PCA on the given dataset is not advisable.

For our dataset, the Bartlett's sphericity test gives us an output where p-value is 0. Since, P-value is less than the significance level of 0.05, We fail to reject the null hypothesis at 5% level of significance. At Least one pair of variables in the data are correlated.

- The Kaiser-Meyer-Olkin test gives us the measure of sampling adequacy (MSA) for PCA to be done. Generally, if MSA is less than 0.5, PCA is not recommended. If the MSA is above 0.7 the PCA is meant to provide meaningful components.

The KMO test on the given dataset, gave us 0.8 MSA. Thus, PCA on the given dataset can be done.

```
[ ]  from factor_analyzer.factor_analyzer import calculate_bartlett_sphericity
     chi_square_value, p_value=calculate_bartlett_sphericity(data_scaled)
     p_value

     0.0

[ ]  from factor_analyzer.factor_analyzer import calculate_kmo
     kmo_all,kmo_model=calculate_kmo(data_scaled)
     kmo_model

     0.8039889932781807
```

From the tests done prior to PCA, we shall learn that our PCA testing is possible. Therefore, I was able to get the covariance matrix, eigenvalues and eigenvector.

```
[[1.00156495 0.91760364 0.97210871 ... 0.53769433 0.76357722 0.73684378]
 [0.91760364 1.00156495 0.98417823 ... 0.5891007  0.84621844 0.71718181]
 [0.97210871 0.98417823 1.00156495 ... 0.572748   0.82894851 0.74775097]
 ...
 [0.53769433 0.5891007  0.572748   ... 1.00156495 0.61052325 0.52191235]
 [0.76357722 0.84621844 0.82894851 ... 0.61052325 1.00156495 0.88228018]
 [0.73684378 0.71718181 0.74775097 ... 0.52191235 0.88228018 1.00156495]]
```

Above given output is the covariance matrix for all 57 variables.

```
array([ 3.18135647e+01+0.00000000e+00j,  7.86942415e+00+0.00000000e+00j,
        4.15340812e+00+0.00000000e+00j,  3.66879058e+00+0.00000000e+00j,
        2.20652588e+00+0.00000000e+00j,  1.93827502e+00+0.00000000e+00j,
        1.17617374e+00+0.00000000e+00j,  7.51159086e-01+0.00000000e+00j,
        6.17053743e-01+0.00000000e+00j,  5.28300887e-01+0.00000000e+00j,
        4.29831189e-01+0.00000000e+00j,  3.53440201e-01+0.00000000e+00j,
        2.96163013e-01+0.00000000e+00j,  2.81275560e-01+0.00000000e+00j,
        1.92158325e-01+0.00000000e+00j,  1.36267920e-01+0.00000000e+00j,
        1.13389199e-01+0.00000000e+00j,  1.06303946e-01+0.00000000e+00j,
        9.72885376e-02+0.00000000e+00j,  8.01062194e-02+0.00000000e+00j,
        5.76089954e-02+0.00000000e+00j,  4.43955966e-02+0.00000000e+00j,
        3.78910846e-02+0.00000000e+00j,  2.96360194e-02+0.00000000e+00j,
        2.70797618e-02+0.00000000e+00j,  2.34458139e-02+0.00000000e+00j,
        1.45111511e-02+0.00000000e+00j,  7.13559124e-04+0.00000000e+00j,
        1.06789820e-03+0.00000000e+00j,  2.59771182e-03+0.00000000e+00j,
        5.02601514e-03+0.00000000e+00j,  1.09852268e-02+0.00000000e+00j,
        9.31507853e-03+0.00000000e+00j,  8.13540203e-03+0.00000000e+00j,
        7.89250253e-03+0.00000000e+00j, -1.62639278e-15+0.00000000e+00j,
        1.73838880e-15+0.00000000e+00j, -1.23163836e-15+0.00000000e+00j,
       -1.09693403e-15+0.00000000e+00j,  1.22980914e-15+2.39724559e-16j,
        1.22980914e-15-2.39724559e-16j,  1.17710893e-15+0.00000000e+00j,
       -8.30353209e-16+0.00000000e+00j,  1.00394338e-15+0.00000000e+00j,
        9.54474887e-16+0.00000000e+00j, -6.07659961e-16+7.92737922e-17j,
       -6.07659961e-16-7.92737922e-17j,  7.62061776e-16+0.00000000e+00j,
       -3.84222466e-16+0.00000000e+00j, -3.62660144e-16+0.00000000e+00j,
       -1.65061771e-16+0.00000000e+00j,  1.29780694e-17+2.35304366e-17j,
        1.29780694e-17-2.35304366e-17j,  1.54490058e-16+0.00000000e+00j,
        3.05572930e-16+0.00000000e+00j,  4.57999896e-16+0.00000000e+00j,
        4.31159259e-16+0.00000000e+00j])
```

The above given array gives us the eigenvalue of the dataset.

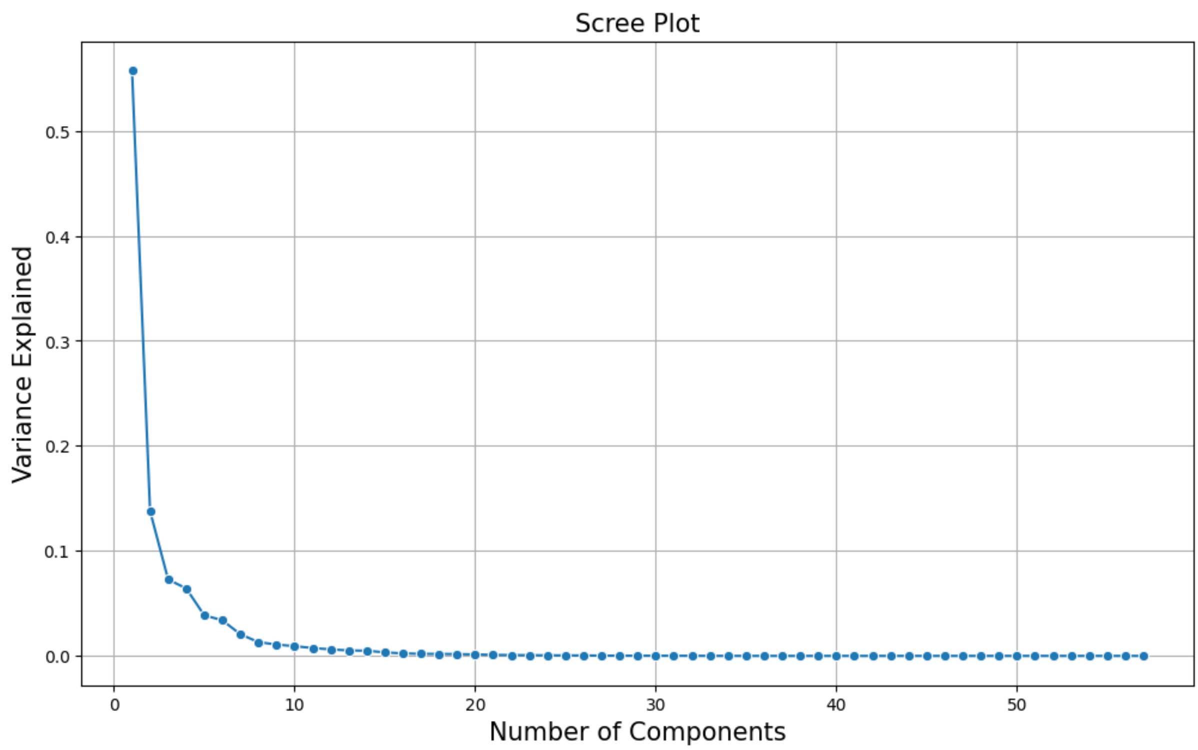
```
array([[-1.56020579e-01+0.j,  1.26346525e-01+0.j, -2.69025037e-03+0.j,
        ...,  8.59813362e-14+0.j, -1.76746604e-13+0.j,
         1.08099785e-13+0.j],
       [-1.67117635e-01+0.j,  8.96765481e-02+0.j,  5.66976191e-02+0.j,
        ...,  5.28021877e-02+0.j, -1.10854618e-01+0.j,
         2.72794503e-02+0.j],
       [-1.65553179e-01+0.j,  1.04912371e-01+0.j,  3.87494746e-02+0.j,
        ...,  2.04945480e-01+0.j, -2.01549789e-01+0.j,
         1.06371673e-01+0.j],
       ...,
       [-1.32192245e-01+0.j, -5.08133220e-02+0.j, -7.87198691e-02+0.j,
        ...,  5.02777797e-03+0.j,  2.99863496e-03+0.j,
         5.44284135e-05+0.j],
       [-1.50375578e-01+0.j,  6.53645529e-02+0.j,  1.11827318e-01+0.j,
        ..., -3.00935627e-02+0.j, -1.48422322e-02+0.j,
         9.62462558e-02+0.j],
       [-1.31066203e-01+0.j,  7.38474208e-02+0.j,  1.02552501e-01+0.j,
        ...,  2.84776742e-02+0.j, -5.59381919e-02+0.j,
         1.00243536e-01+0.j]])
```

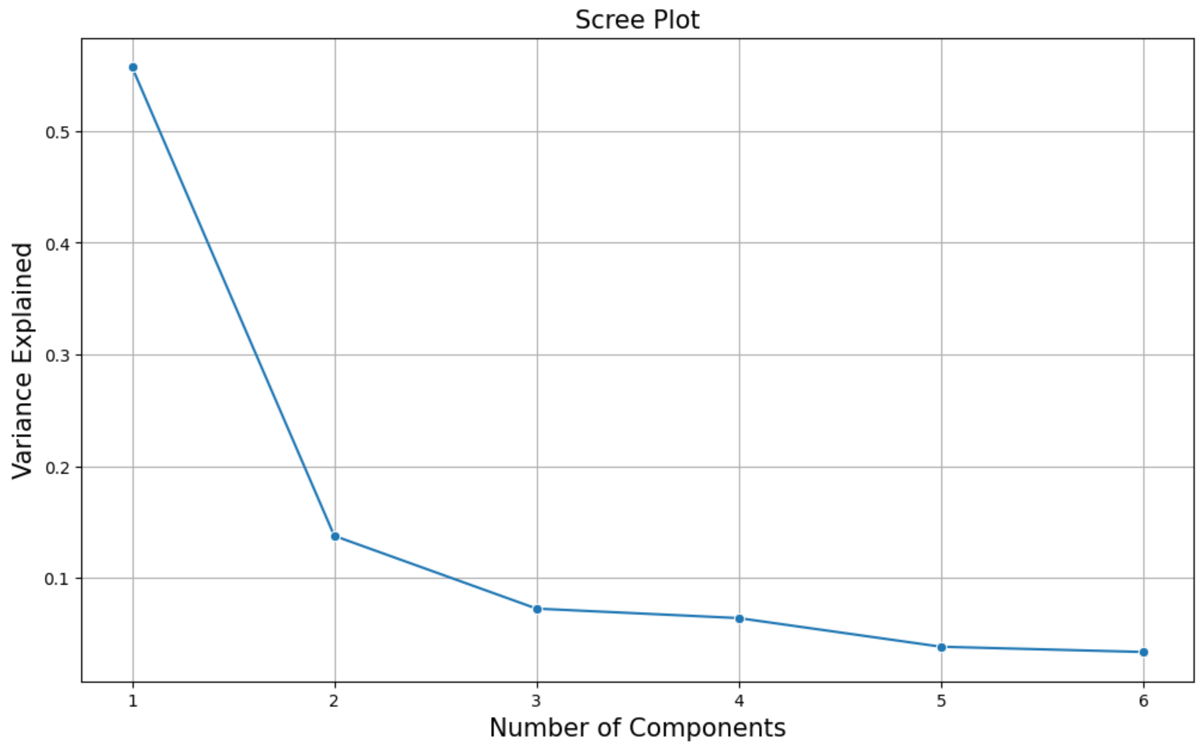
The above given array shows the eigenvectors of the dataset.

6. Next, By fitting the scaled dataset into the PCA model with n_components at 57, I was able to find the cumulative summation of the explained variance for the above mentioned model. I decided to set the n_component at 57 to find the optimum n_components for the PCA. I got the following output.

```
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, 0.97745473, 0.98238168, 0.98574761,
       0.98813454, 0.99012071, 0.99198278, 0.99368693, 0.99509011,
       0.99609921, 0.99687687, 0.99754058, 0.9980597 , 0.99853404,
       0.99894473, 0.99919891, 0.99939134, 0.9995545 , 0.99969701,
       0.99983525, 0.99992329, 0.9999688 , 0.9999875 , 1.        ,
       1.        , 1.        , 1.        , 1.        , 1.        ,
       1.        , 1.        , 1.        , 1.        , 1.        ,
       1.        , 1.        , 1.        , 1.        , 1.        ,
       1.        , 1.        , 1.        , 1.        , 1.        ,
       1.        , 1.        ])
```

As we can see, at the 6th component the value is 0.905. Thus, 90% of the model is explained by the 6th component, the optimum number of PCs for the model is 6.

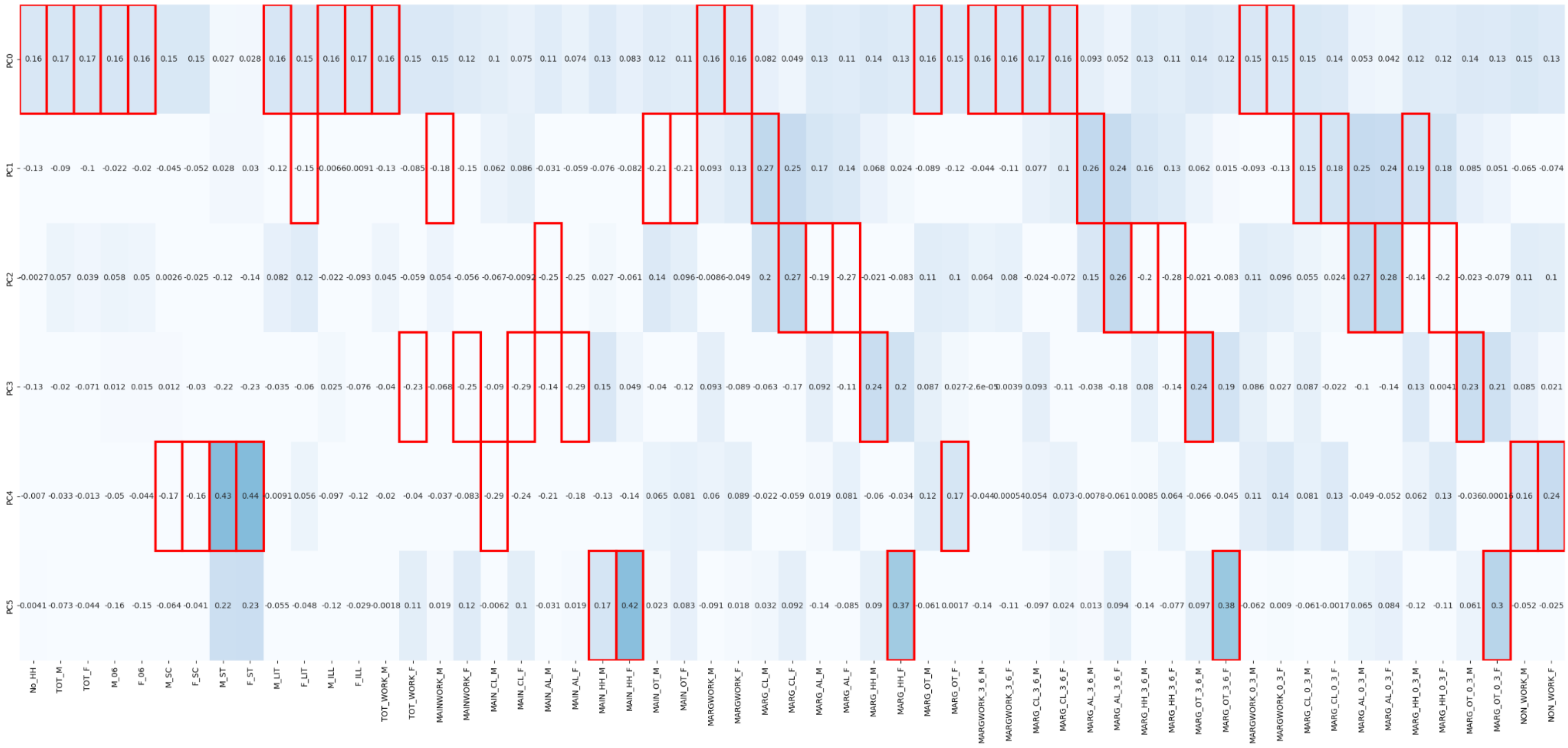




From the first given scree plot, we can notice that from the 6th component the variance explained by the model tends to stagnate. This is further proven in the second scree plot where the line tends to have no significant change after point 6.

Thus, the optimum number of PCs for the given dataset for the PCA is 6.

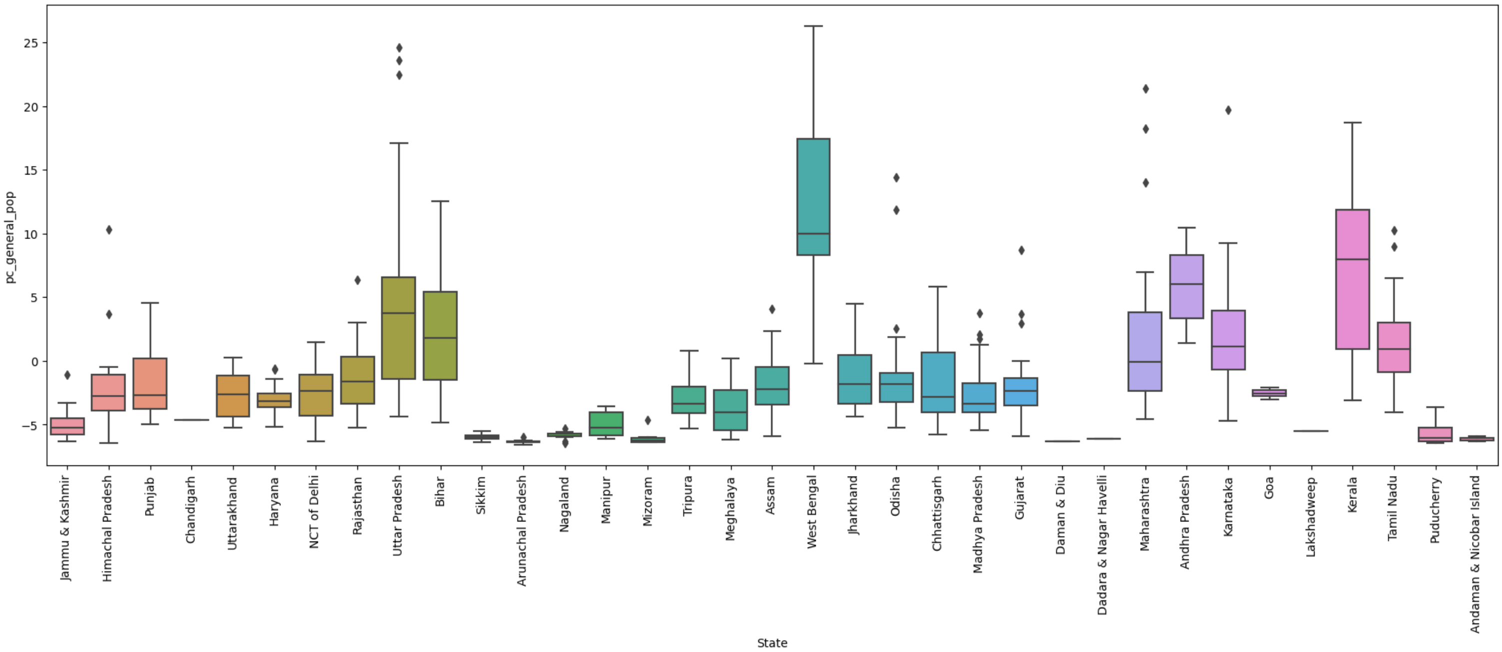
7. The diagram gives us the explained variance of the 6 components. (Note: The diagram is not clear in the word document due to constrained borders. Please refer to the jupyter notebook for clearer view.)



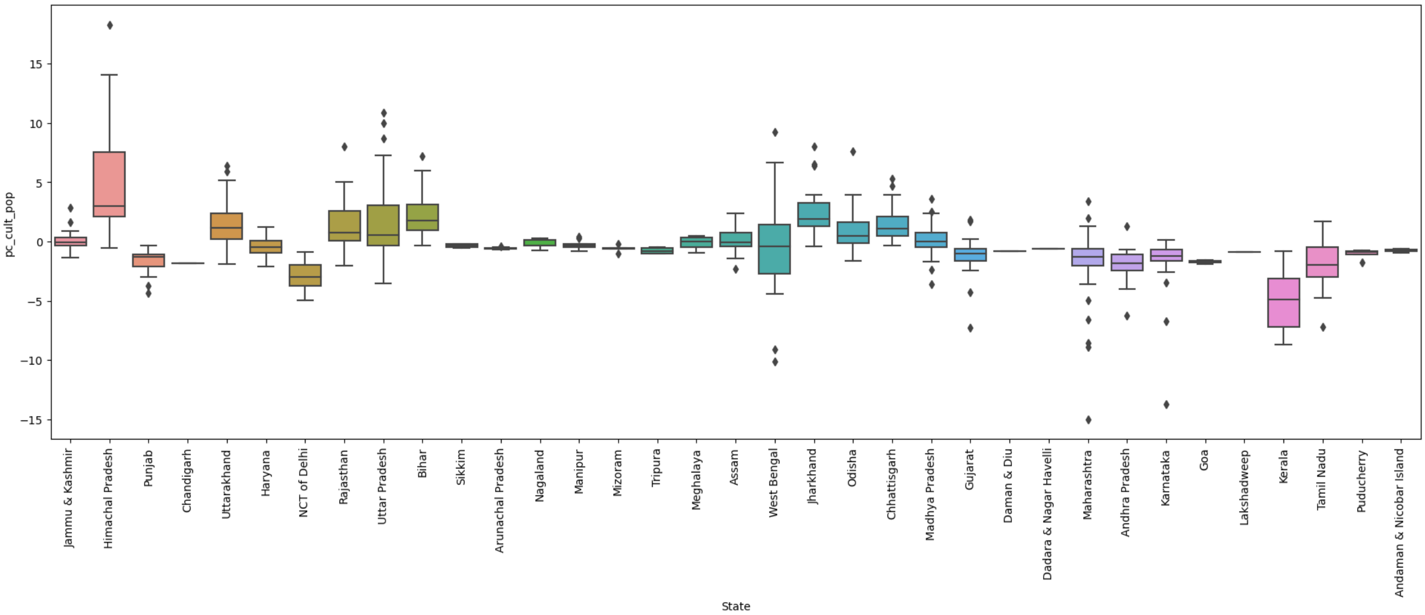
We can learn from the given diagram that

- The first principal component explains the variables like TOT_M, TOT_F, M_LIT, F_ILL. Thus, we can say the first principal component mostly explains the general population.
- The second principal component explains the variables like MARG_CL_M and MARG_CL_0_3_F. Thus, we can say that the second principal component explains the cultivator population.
- The third principal component explains the variables like MARG_AL_3_6_M, MARG_AL_0_3_M, and MARG_AL_0_3_F, Thus we can say that the third principal component explains the agriculturing population.

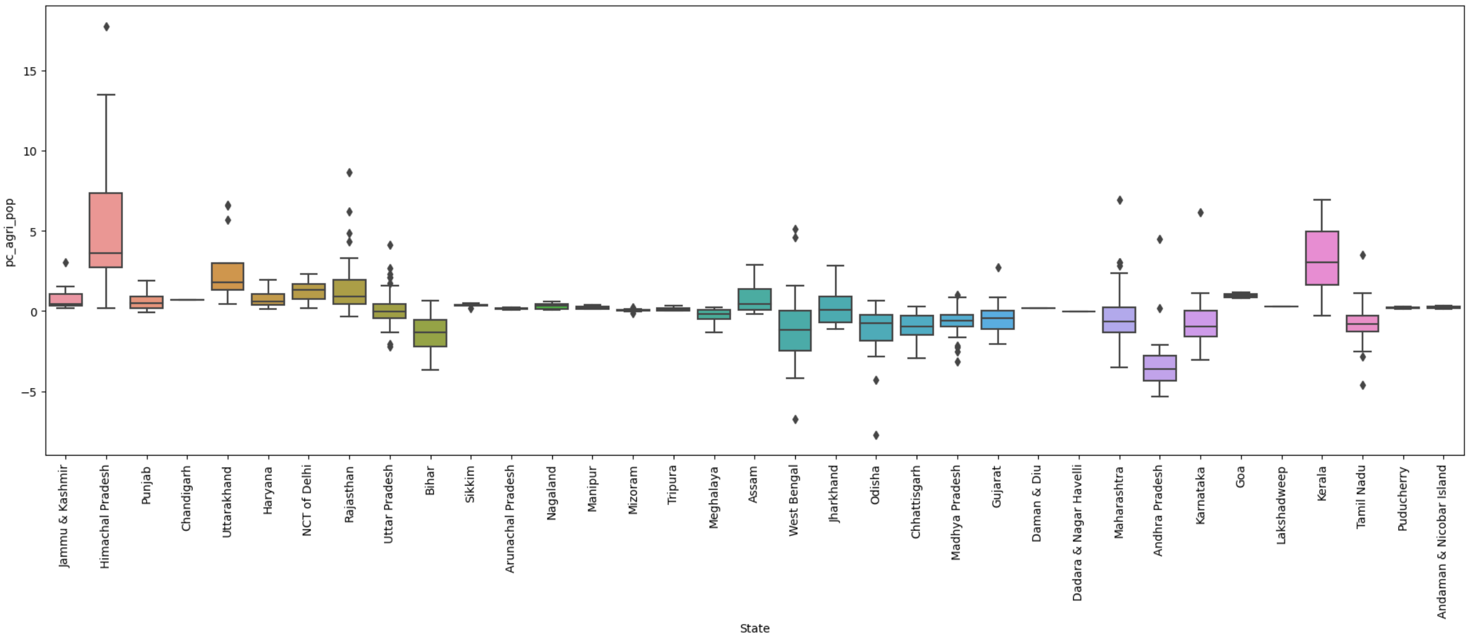
- The fourth principal component explains the variables like MARG_OT_3_6_M and MARG_OT_0_3_F. Thus, we can say that the fourth principal component explains the other worker population
- The fifth principal component explains the variables like M_ST and F_ST. Thus we can say that the fifth principal component explains the scheduled tribe population.
- The sixth principal component explains the variables like MAIN_HH_F and MARG_HH_0_3_F. Thus we can say tha the sixth principal component explains the household industries employed population.



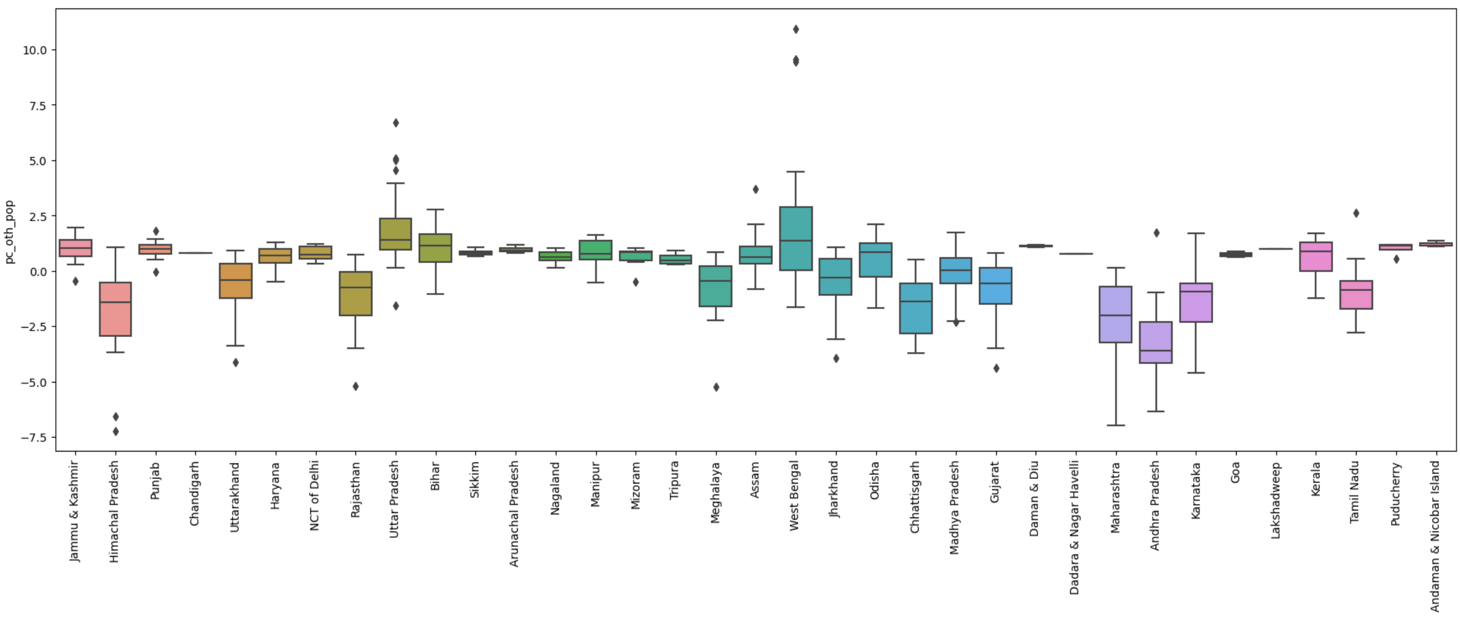
The above given diagram shows tha picturisation of the first principal component with the states. We can say that the general population in west bengal is the highest.



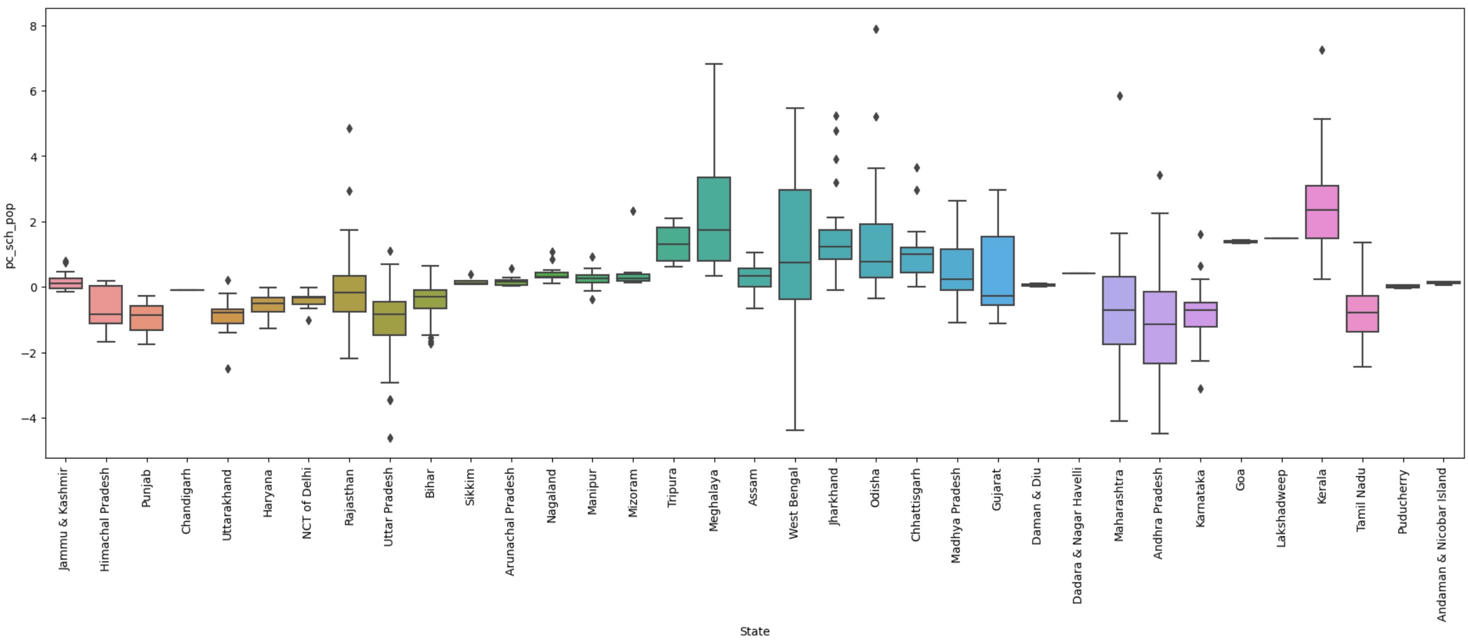
The above given plot shows us the picturisation of the cultivator population across the states of India. We can see from the plot that Himachal Pradesh has the highest number of cultivator population.



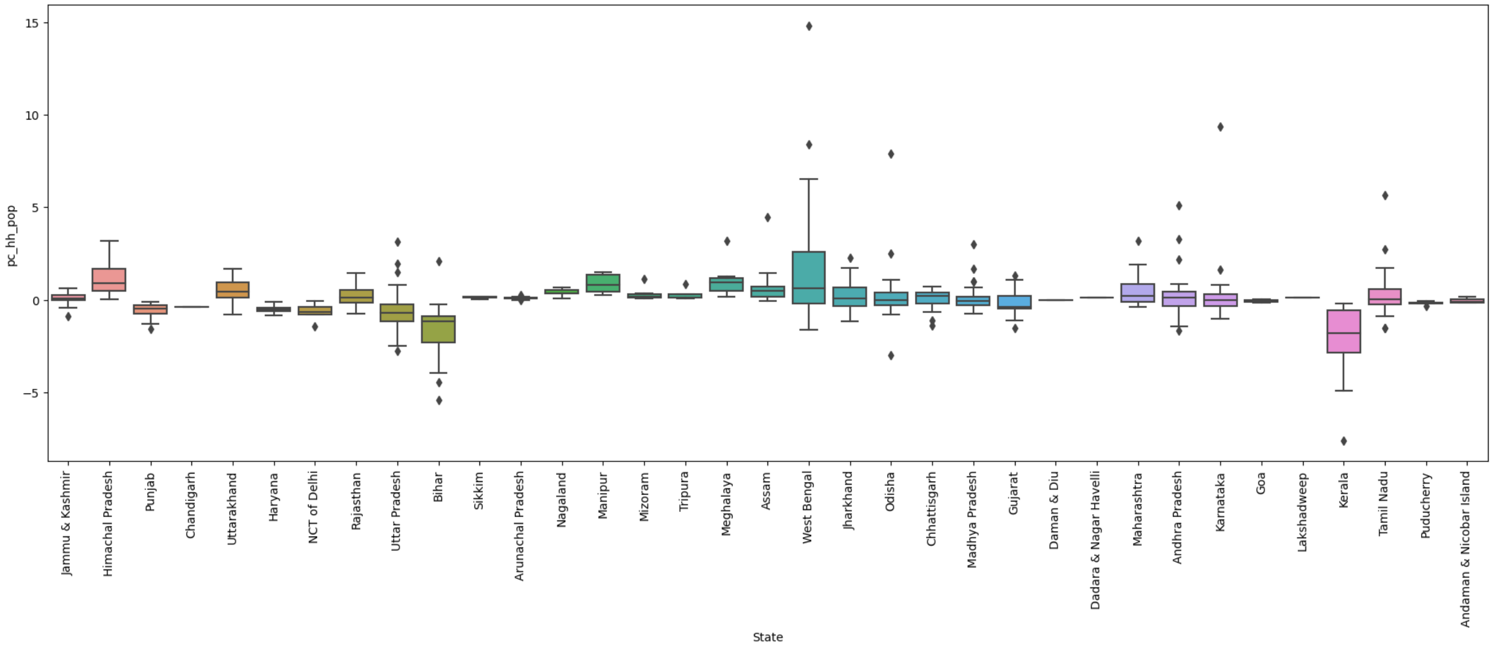
From the above given plot, we can see the picturisation of the agriculturalist population among the states. Himachal Pradesh has the highest agriculturally employed population.



The picturisation of the population employed in other jobs is given in terms of the states in the above given diagram. West Bengal has the highest number of population employed in other work.



The above given diagram shows the scheduled tribe population across the states. Meghalaya tends to have the highest amount of scheduled tribe population.



The above given diagram gives us the Household employed population across the states.

West Bengal has the highest population with household employment.

8. The linear equation for the first PC is as follows:

Y = (0.16) * No_HH + (0.17) * TOT_M + (0.17) * TOT_F + (0.16) * M_06 + (0.16) * F_06 + (0.15) * M_SC +