Machine Learning Report - Coded Project

Table of Contents

Table of Figures

Problem Statement 1:

Clustering:

Digital Ads Data:

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

The following three features are commonly used in digital marketing:

CPM = (Total Campaign Spend / Number of Impressions) * 1,000. 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 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.
- Check if there are any outliers.
- Do you think treating outliers is necessary for K-Means clustering? Based on your judgment decide whether to treat outliers and if yes, which method to employ. (As an analyst your judgment 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.

Define the problem and perform Exploratory Data Analysis

Getting the first 10 rows of the table

Out[85]:

	Timestamp	InventoryType	Ad - Length	Ad-Width	Ad Size	Ad Type	Platform	Device Type	Format	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend
0	2020-9-2-17	Format1	300	250	75000	Inter222	Video	Desktop	Display	1806	325	323	1	0.00
1	2020-9-2-10	Format1	300	250	75000	Inter227	App	Mobile	Video	1780	285	285	1	0.00
2	2020-9-1-22	Format1	300	250	75000	Inter222	Video	Desktop	Display	2727	356	355	1	0.00
3	2020-9-3-20	Format1	300	250	75000	Inter228	Video	Mobile	Video	2430	497	495	1	0.00
4	2020-9-4-15	Format1	300	250	75000	Inter217	Web	Desktop	Video	1218	242	242	1	0.00
5	2020-9-4-5	Format1	300	250	75000	Inter219	Video	Desktop	Display	490	64	64	2	0.00
6	2020-9-4-6	Format1	300	250	75000	Inter221	App	Mobile	Video	1197	202	202	1	0.01
7	2020-9-6-7	Format1	300	250	75000	Inter228	Video	Mobile	Video	1363	198	196	1	0.00
8	2020-9-8-6	Format1	300	250	75000	Inter223	Web	Mobile	Video	1402	137	136	1	0.00
9	2020-9-11-17	Format1	300	250	75000	Inter228	Video	Mobile	Display	1816	312	311	1	0.00

Getting the last 10 rows of the table

	Timestamp	InventoryType	Ad - Length	Ad-Width	Ad Size	Ad Type	Platform	Device Type	Format	Available_Impressions	Matched_Queries	Impressions	Clicks	Sp
23056	2020-11-23-4	Format4	120	600	72000	Inter223	Web	Mobile	Video	2	2	2	1	(
23057	2020-11-20-2	Format4	120	600	72000	Inter224	Web	Desktop	Display	5	2	2	1	(
23058	2020-11-4-3	Format5	720	300	216000	Inter223	Web	Mobile	Video	1	1	1	1	(
23059	2020-11-13-4	Format5	720	300	216000	Inter228	Video	Mobile	Display	2	2	2	1	(
23060	2020-11-16-5	Format4	120	600	72000	Inter225	Video	Mobile	Display	4	4	4	1	(
23061	2020-9-13-7	Format5	720	300	216000	Inter220	Web	Mobile	Video	1	1	1	1	(
23062	2020-11-2-7	Format5	720	300	216000	Inter224	Web	Desktop	Video	3	2	2	1	(
23063	2020-9-14-22	Format5	720	300	216000	Inter218	App	Mobile	Video	2	1	1	1	(
23064	2020-11-18-2	Format4	120	600	72000	inter230	Video	Mobile	Video	7	1	1	1	(
23065	2020-9-14-0	Format5	720	300	216000	Inter221	App	Mobile	Video	2	2	2	1	(

Shape of the Data

In [118]:
```
df.shape
```
Out[118]: (23066, 19)

There are 23066 rows and 19 columns.

Getting the data type

```
<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
```

Getting the statistical summary for the numerical variables

```
df.describe().round(2).T
```

Out[88]:

	count	mean	std	min	25%	50%	75%	max
Ad - Length	23066.0	385.16	233.65	120.00	120.00	300.00	720.00	728.00
Ad- Width	23066.0	337.90	203.09	70.00	250.00	300.00	600.00	600.00
Ad Size	23066.0	96674.47	61538.33	33600.00	72000.00	72000.00	84000.00	216000.00
Available_Impressions	23066.0	2432043.67	4742887.76	1.00	33672.25	483771.00	2527711.75	27592861.00
Matched_Queries	23066.0	1295099.14	2512969.86	1.00	18282.50	258087.50	1180700.00	14702025.00
Impressions	23066.0	1241519.52	2429399.96	1.00	7990.50	225290.00	1112428.50	14194774.00
Clicks	23066.0	10678.52	17353.41	1.00	710.00	4425.00	12793.75	143049.00
Spend	23066.0	2706.63	4067.93	0.00	85.18	1425.12	3121.40	26931.87
Fee	23066.0	0.34	0.03	0.21	0.33	0.35	0.35	0.35
Revenue	23066.0	1924.25	3105.24	0.00	55.37	926.34	2091.34	21276.18
CTR	18330.0	0.07	0.08	0.00	0.00	0.08	0.13	1.00
CPM	18330.0	7.67	6.48	0.00	1.71	7.66	12.51	81.56
CPC	18330.0	0.35	0.34	0.00	0.09	0.16	0.57	7.26

checking the duplicate values in the dataset

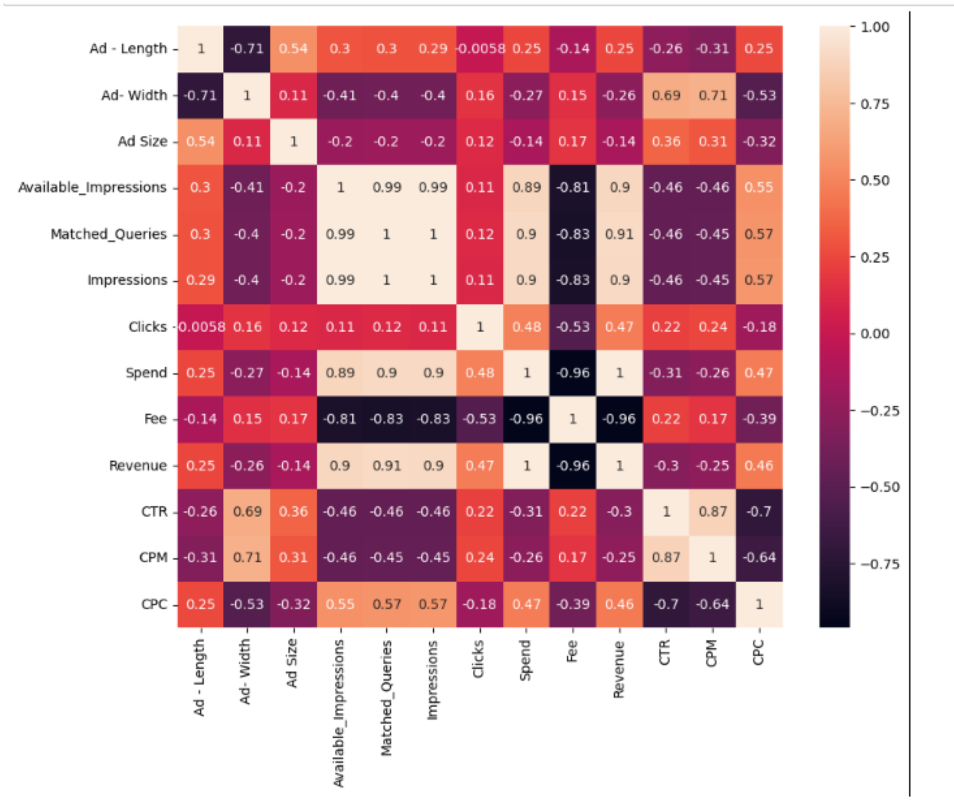
```
In [89]: #checking the duplicate values in the dataset
         df.duplicated().sum()

Out[89]: 0
```

There are no duplicate values found in the dataset.

checking the null values in the dataset

```
Out[90]: 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
```

The shape of the dataset contains the 23066 Rows and 19 columns. From the above dataset, we can find there are no duplicate values. There are 4736 null values found in the columns in CTR, CPC and CPC.



1. Heat Map

CTR and CPM have a high correlation with 0.87, and CPC and Impressions have a high correlation with 0.57.

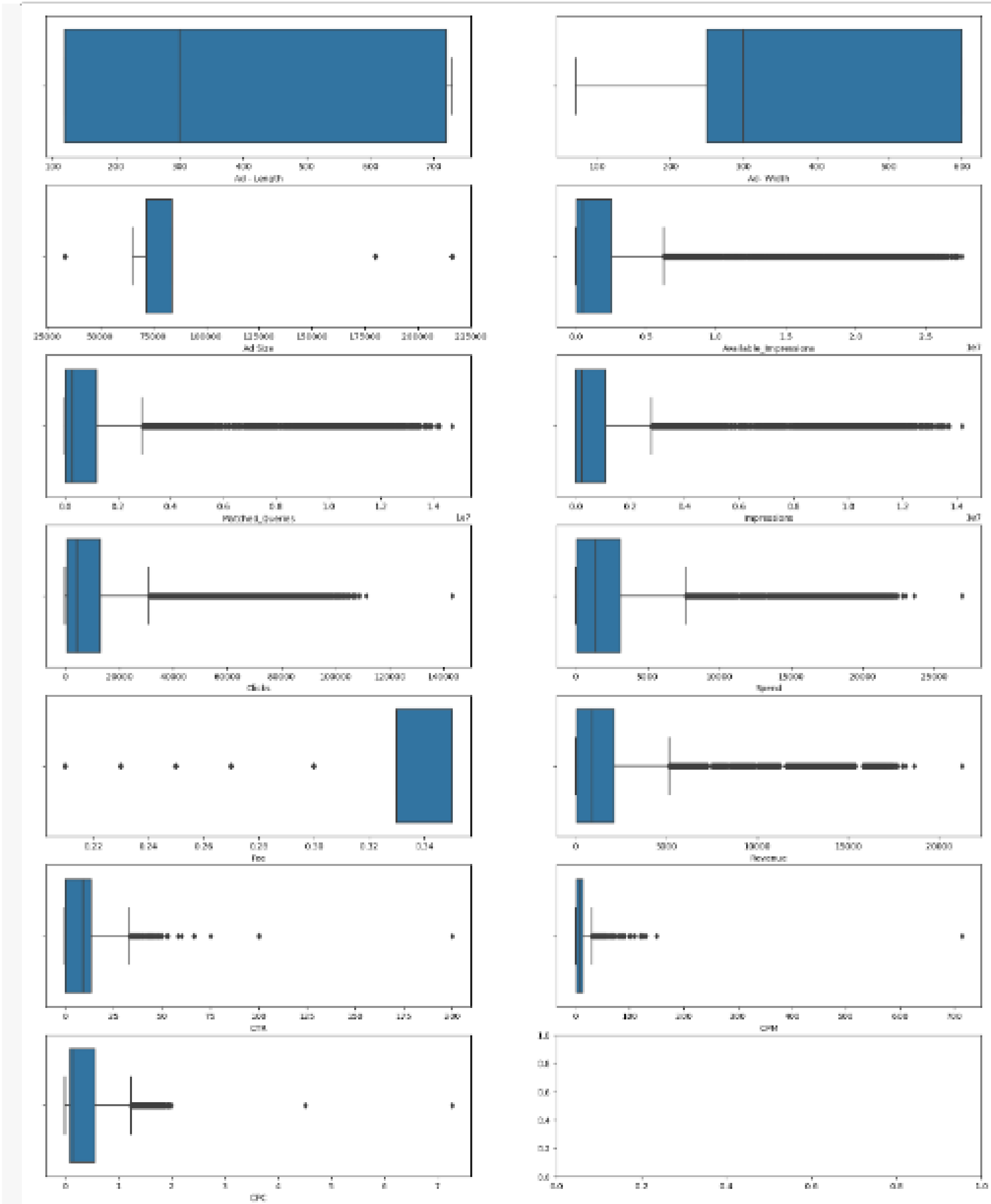
CPC and Revenue have a high correlation with 0.46.

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

Checking the null values after treating the calculated column CPC, CTR and CPM.

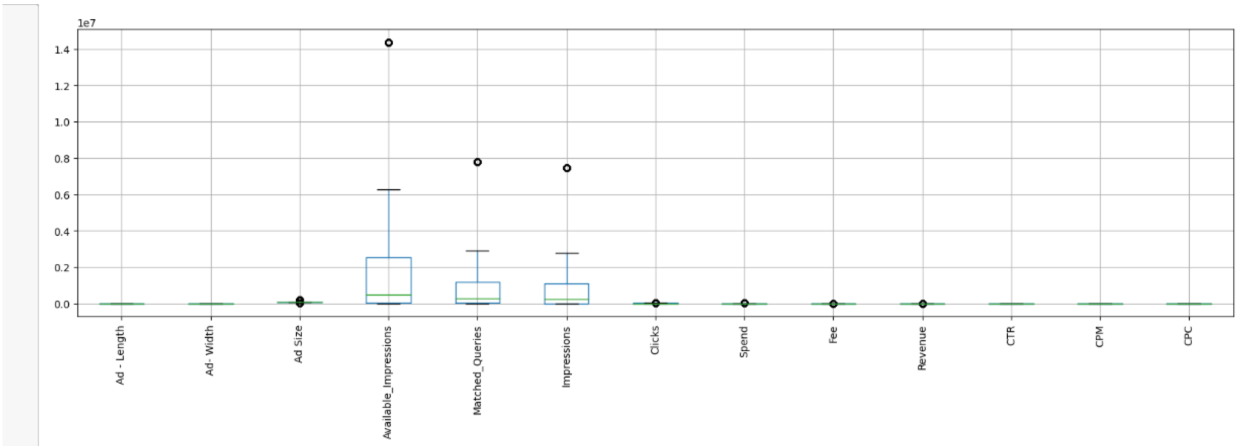
```
Out[95]: Timestamp                0
         InventoryType            0
         Ad - Length              0
         Ad- Width                0
         Ad Size                  0
         Ad Type                  0
         Platform                 0
         Device Type              0
         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
```

Before Treating Outliers



2. Boxplot before treating outliers

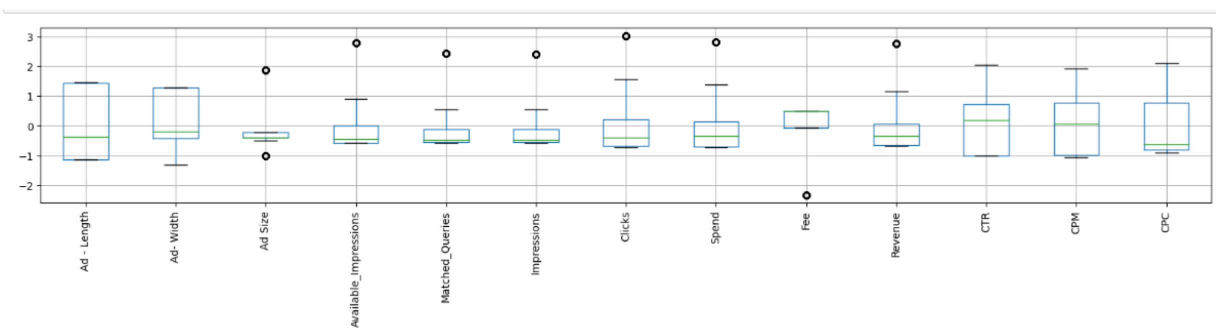
After Treating Outliers



3. Boxplot after treating outliers

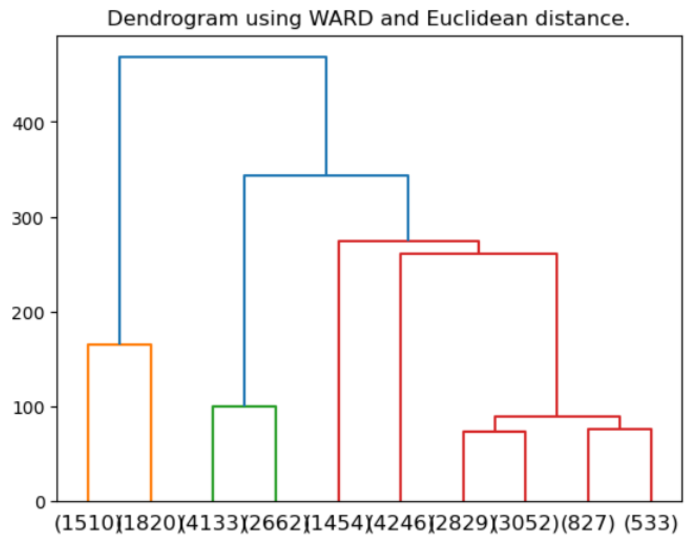
Perform z-score scaling and discuss how it affects the speed of the algorithm.

Z- Z-score scaling doesn't affect the speed of the algorithm, it only normalize the data.



4. Boxplot after scaling the data with Z-score

Perform Hierarchical by constructing a Dendrogram using WARD and Euclidean distance.



5. Dendrogram using WARD and Euclidean distance

Counting the number of clusters

```
Out[159]: clusters
          5    7241
          2    6795
          4    4246
          1    3330
          3    1454
          Name: count, dtype: int64
```

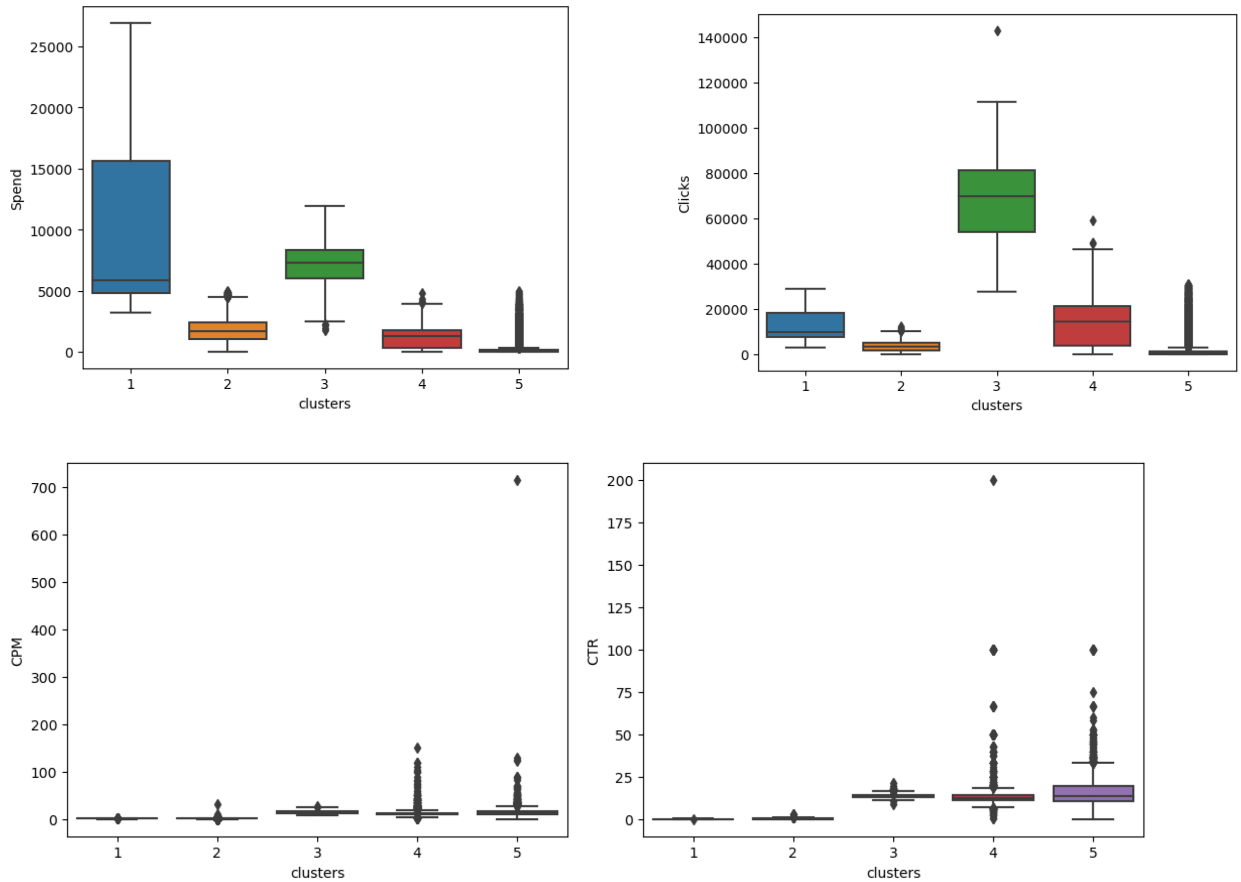
Average of cluster data

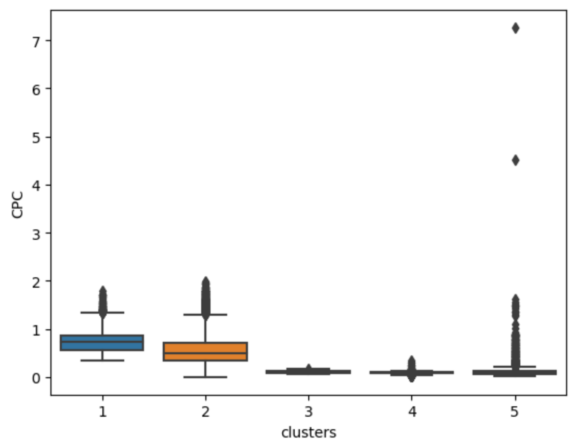
Out[165]:

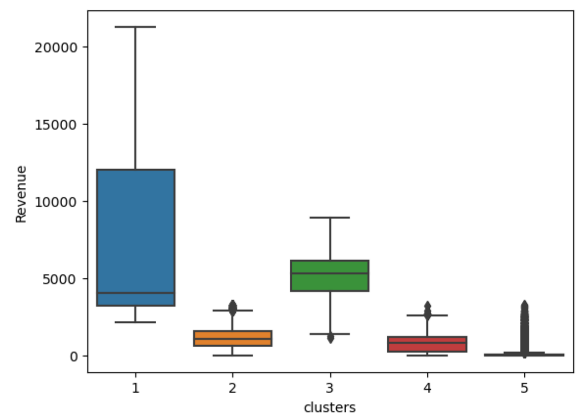
clusters	Ad - Length	Ad- Width	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
1	488.666667	189.810811	1.172432e+07	6.350334e+06	6.144069e+06	12542.445646	9692.230670	0.281796	7200.513204	0.213225	1.540055	0.755331
2	418.646652	148.232524	2.057579e+06	1.009505e+06	9.701911e+05	3465.703164	1738.297186	0.347310	1139.830653	0.387728	1.785516	0.567322
3	144.660248	568.466300	8.392876e+05	5.870326e+05	4.952795e+05	67693.137552	7168.359319	0.286169	5155.231214	13.773813	5.057546	0.109326
4	720.000000	300.000000	2.440302e+05	1.334844e+05	1.132786e+05	14031.383891	1213.779016	0.349543	790.501469	13.978391	2.149377	0.089515
5	158.093081	559.901947	1.129404e+05	6.162672e+04	5.297255e+04	3175.229941	382.222135	0.349586	249.933791	15.365237	4.216383	0.118293

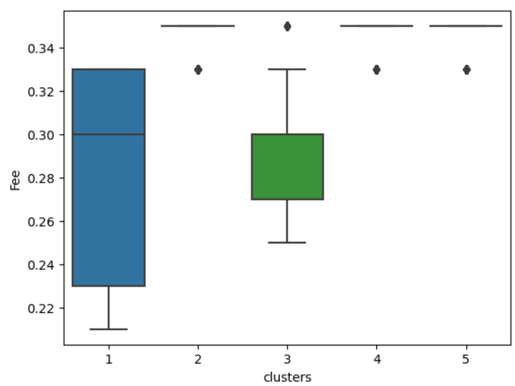
Visualize the clustering data

6. Boxplot of visualizing the clustered data







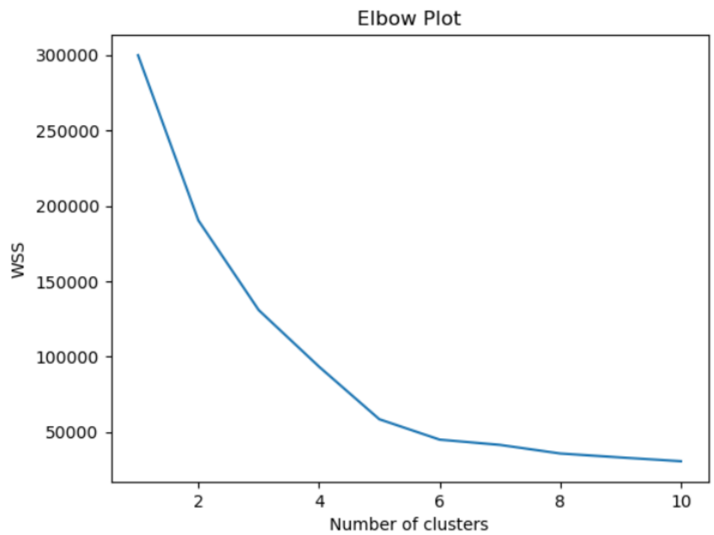


Make Elbow plot (up to n=10) and identify optimum number of clusters for k-means algorithm

K- Means Inertia values.

```
In [228]: wss

Out[228]: [299857.99999999994,
           190314.0756526771,
           130836.50109119201,
           93328.47237282965,
           58429.24504196264,
           44913.564048384476,
           41416.673056804226,
           35704.81469079199,
           33081.881861955124,
           30574.800585570243]
```

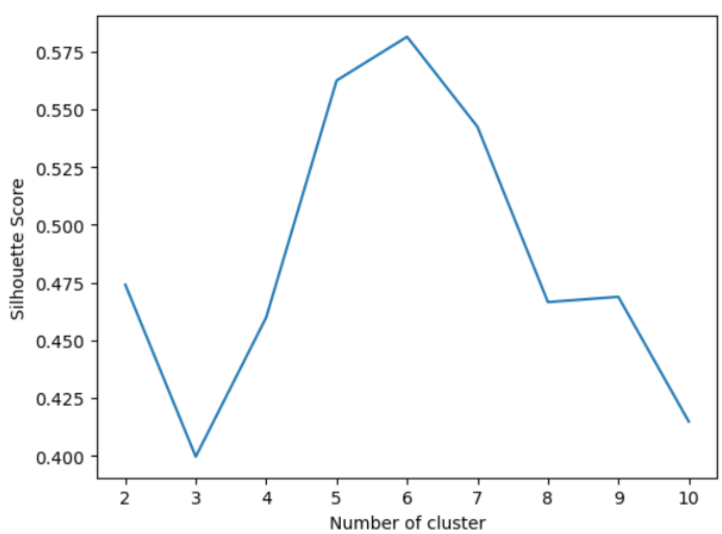
7. Elbow plot



optimum number of clusters for k-means algorithm is 5.

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

8. silhouette scores vs Number of clusters

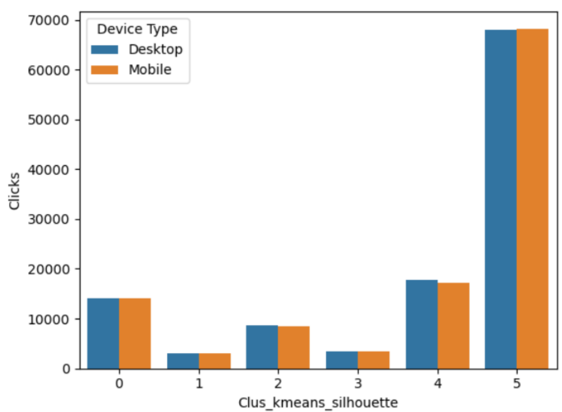


```
The silhouette scores for 2 is 0.47408614007079075
The silhouette scores for 3 is 0.3996085499898667
The silhouette scores for 4 is 0.4598885200834244
The silhouette scores for 5 is 0.5624992189264133
The silhouette scores for 6 is 0.5814351008554612
The silhouette scores for 7 is 0.5424544472408298
The silhouette scores for 8 is 0.46653628830971466
The silhouette scores for 9 is 0.4688187168141927
The silhouette scores for 10 is 0.4147804025269941
```

the optimum number of clusters based on the silhouette scores is 5.

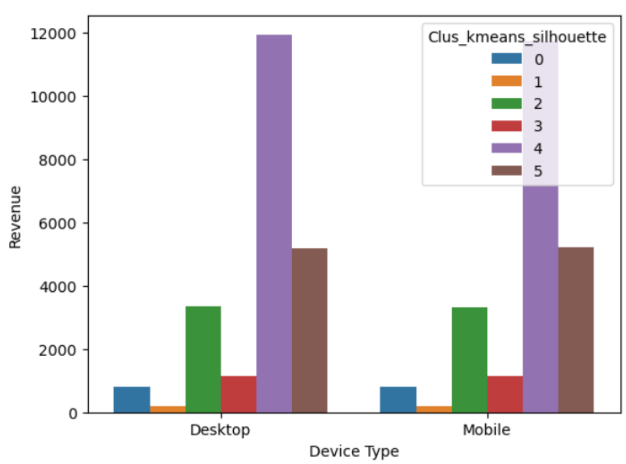
Profile the ads based on optimum number of clusters using silhouette score and your domain understanding

9. Clicks vs K_means_silhouette



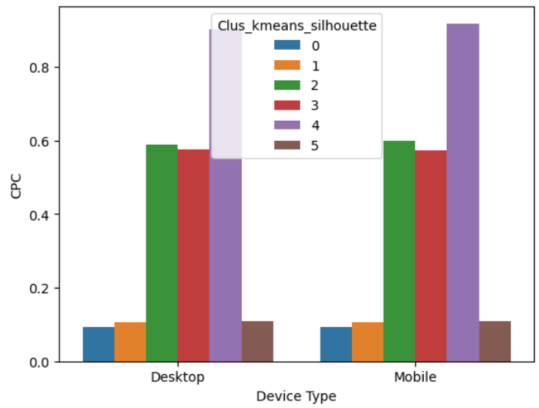
10. Revenue vs Device type

```
Out[276]: <Axes: xlabel='Device Type', ylabel='Revenue'>
```



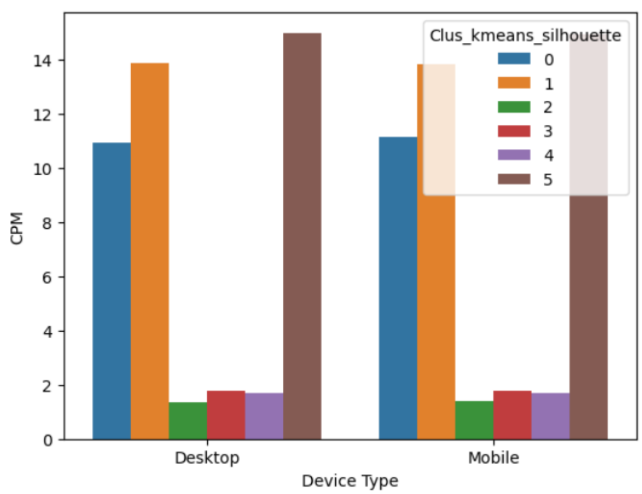
11. CPC vs Device type

Out[282]: <Axes: xlabel='Device Type', ylabel='CPC'>



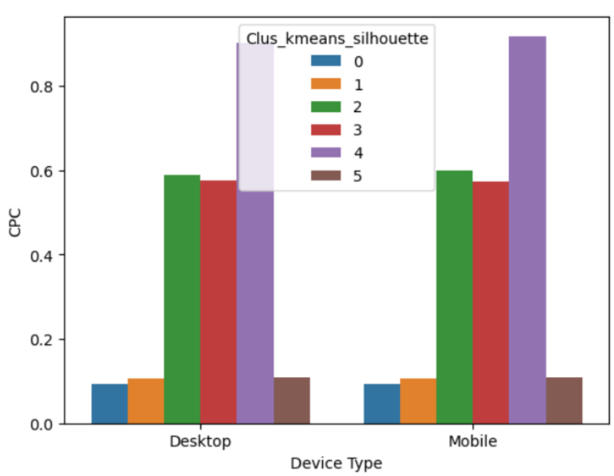
12. CPM vs Device type

Out[284]: <Axes: xlabel='Device Type', ylabel='CPM'>



13. CPC vs Device type

Out[286]: <Axes: xlabel='Device Type', ylabel='CPC'>



Observations:

Clicks It is a marketing metric that counts the number of times users have clicked on the advertisement to reach an online property.

Impressions The impression counts of the particular Advertisement out of the total available impressions.

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.

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.

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.

From the above k-means silhouette scores:

we can say that most of the advertisement cluster falls on the cluster 3 , followed by cluster 1, cluster 0 , cluster 2, cluster 4 and cluster 5.

The cost per click (CPC) for cluster 5 is high. The cost per 1000 impressions (CPM) for cluster 5 is high. The click through rate (CTR) for cluster 1 is high.

The cluster 5 has highest number of clicks that counts the number of times users have clicked on the advertisement to reach an online property done by both desktop and mobile device.

The cluster 4 has achieved the highest number of revenue using both the device type (mobile & desktop), followed by cluster 5, cluster 2, cluster 3, cluster 0, cluster 1.

Cluster 3 has a minimum ad size when compared to other clusters.

The average CPM (cost per 1000 impressions) is highest in cluster 0, cluster 1, and cluster 5 which means the ads are displayed by spending a huge amount to gain an impression on users.

The Average CPC (Cost-per-click) is high for cluster 2, cluster 3, and cluster 4, due to this it has the highest impression count of the Advertisement out of the total available impressions.

The lower the CPM higher the revenue generated.

The Digital Marketing company should focus on ads based on CPM which yields revenue by increasing the number of impressions on the ads rather than going for CPC.

Problem Statement 2:

PCA:

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

The data collected has so many variables thus making it difficult to find useful details without using Data Science Techniques. You are tasked to perform detailed EDA and identify Optimum Principal Components that explain 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

Define the problem and perform Exploratory Data Analysis

Getting the first 5 rows of the dataset

Out[6]:

click to scroll output; double click to hide

	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_
0	1	1	Jammu & Kashmir	Kupwara	7707	23388	29796	5862	6196	3	...	1150	749	180	
1	1	2	Jammu & Kashmir	Badgam	6218	19585	23102	4482	3733	7	...	525	715	123	
2	1	3	Jammu & Kashmir	Leh(Ladakh)	4452	6546	10964	1082	1018	3	...	114	188	44	
3	1	4	Jammu & Kashmir	Kargil	1320	2784	4206	563	677	0	...	194	247	61	
4	1	5	Jammu & Kashmir	Punch	11654	20591	29981	5157	4587	20	...	874	1928	465	

5 rows × 61 columns

Getting the last 5 rows of the dataset

Out[7]:

	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_
635	34	636	Puducherry	Mahe	3333	8154	11781	1146	1203	21	...	32	47	0	
636	34	637	Puducherry	Karaikal	10612	12346	21691	1544	1533	2234	...	155	337	3	
637	35	638	Andaman & Nicobar Island	Nicobars	1275	1549	2630	227	225	0	...	104	134	9	
638	35	639	Andaman & Nicobar Island	North & Middle Andaman	3762	5200	8012	723	664	0	...	136	172	24	
639	35	640	Andaman & Nicobar Island	South Andaman	7975	11977	18049	1470	1358	0	...	173	122	6	

5 rows × 61 columns

Getting the data type

```
<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
```

Out[9]:

	count	mean	std	min	25%	50%	75%	max
State Code	640.0	17.114062	9.426486	1.0	9.00	18.0	24.00	35.0
Dist.Code	640.0	320.500000	184.896367	1.0	160.75	320.5	480.25	640.0
No_HH	640.0	51222.871875	48135.405475	350.0	19484.00	35837.0	68892.00	310450.0
TOT_M	640.0	79940.576563	73384.511114	391.0	30228.00	58339.0	107918.50	485417.0
TOT_F	640.0	122372.084375	113600.717282	698.0	46517.75	87724.5	164251.75	750392.0
M_06	640.0	12309.098438	11500.906881	56.0	4733.75	9159.0	16520.25	96223.0
F_06	640.0	11942.300000	11326.294567	56.0	4672.25	8663.0	15902.25	95129.0
M_SC	640.0	13820.946875	14426.373130	0.0	3466.25	9591.5	19429.75	103307.0
F_SC	640.0	20778.392188	21727.887713	0.0	5603.25	13709.0	29180.00	156429.0
M_ST	640.0	6191.807813	9912.668948	0.0	293.75	2333.5	7658.00	96785.0
F_ST	640.0	10155.640625	15875.701488	0.0	429.50	3834.5	12480.25	130119.0
M_LIT	640.0	57967.979688	55910.282466	286.0	21298.00	42693.5	77989.50	403261.0
F_LIT	640.0	66359.565625	75037.860207	371.0	20932.00	43796.5	84799.75	571140.0
M_ILL	640.0	21972.596875	19825.605268	105.0	8590.00	15767.5	29512.50	105961.0
F_ILL	640.0	56012.518750	47116.693769	327.0	22367.00	42386.0	78471.00	254160.0
TOT_WORK_M	640.0	37992.407813	36419.537491	100.0	13753.50	27936.5	50226.75	269422.0
TOT_WORK_F	640.0	41295.760938	37192.360943	357.0	16097.75	30588.5	53234.25	257848.0
MAINWORK_M	640.0	30204.446875	31480.915680	65.0	9787.00	21250.5	40119.00	247911.0
MAINWORK_F	640.0	28198.846875	29998.262689	240.0	9502.25	18484.0	35063.25	226166.0
MAIN_CL_M	640.0	5424.342188	4739.161969	0.0	2023.50	4160.5	7695.00	29113.0
MAIN_CL_F	640.0	5486.042188	5326.362728	0.0	1920.25	3908.5	7286.25	36193.0
MAIN_AL_M	640.0	5849.109375	6399.507966	0.0	1070.25	3936.5	8067.25	40843.0
MAIN_AL_F	640.0	8925.995312	12864.287584	0.0	1408.75	3933.5	10617.50	87945.0
MAIN_HH_M	640.0	883.893750	1278.642345	0.0	187.50	498.5	1099.25	16429.0
MAIN_HH_F	640.0	1380.773438	3179.414449	0.0	248.75	540.5	1435.75	45979.0
MAIN_OT_M	640.0	18047.101562	26068.480886	36.0	3997.50	9598.0	21249.50	240855.0
MAIN_OT_F	640.0	12406.035938	18972.202369	153.0	3142.50	6380.5	14368.25	209355.0
MARGWORK_M	640.0	7787.960938	7410.791691	35.0	2937.50	5627.0	9800.25	47553.0
MARGWORK_F	640.0	13096.914062	10996.474528	117.0	5424.50	10175.0	18879.25	66915.0
MARG_CL_M	640.0	1040.737500	1311.546847	0.0	311.75	606.5	1281.00	13201.0
MARG_CL_F	640.0	2307.682813	3564.626095	0.0	630.25	1226.0	2659.25	44324.0
MARG_AL_M	640.0	3304.326562	3781.555707	0.0	873.50	2062.0	4300.75	23719.0
MARG_AL_F	640.0	6463.281250	6773.876298	0.0	1402.50	4020.5	9089.25	45301.0
MARG_HH_M	640.0	316.742188	462.661891	0.0	71.75	166.0	356.50	4298.0
MARG_HH_F	640.0	786.626562	1198.718213	0.0	171.75	429.0	962.50	15448.0
MARG_OT_M	640.0	3126.154687	3609.391821	7.0	935.50	2036.0	3985.25	24728.0
MARG_OT_F	640.0	3539.323438	4115.191314	19.0	1071.75	2349.5	4400.50	36377.0
MARGWORK_3_6_M	640.0	41948.168750	39045.316918	291.0	16208.25	30315.0	57218.75	300937.0
MARGWORK_3_6_F	640.0	81076.323438	82970.406216	341.0	26619.50	56793.0	107924.00	676450.0
MARG_CL_3_6_M	640.0	6394.987500	6019.806644	27.0	2372.00	4630.0	8167.00	39106.0
MARG_CL_3_6_F	640.0	10339.864063	8467.473429	85.0	4351.50	8295.0	15102.00	50065.0
MARG_AL_3_6_M	640.0	789.848438	905.639279	0.0	235.50	480.5	986.00	7426.0
MARG_AL_3_6_F	640.0	1749.584375	2496.541514	0.0	497.25	985.5	2059.00	27171.0
MARG_HH_3_6_M	640.0	2743.635938	3059.586387	0.0	718.75	1714.5	3702.25	19343.0

MARGWORK_3_6_F	640.0	81076.323438	82970.406216	341.0	26619.50	56793.0	107924.00	676450.0
MARG_CL_3_6_M	640.0	6394.987500	6019.806644	27.0	2372.00	4630.0	8167.00	39106.0
MARG_CL_3_6_F	640.0	10339.864063	8467.473429	85.0	4351.50	8295.0	15102.00	50065.0
MARG_AL_3_6_M	640.0	789.848438	905.639279	0.0	235.50	480.5	986.00	7426.0
MARG_AL_3_6_F	640.0	1749.584375	2496.541514	0.0	497.25	985.5	2059.00	27171.0
MARG_HH_3_6_M	640.0	2743.635938	3059.586387	0.0	718.75	1714.5	3702.25	19343.0
MARG_HH_3_6_F	640.0	5169.850000	5335.640960	0.0	1113.75	3294.0	7502.25	36253.0
MARG_OT_3_6_M	640.0	245.362500	358.728567	0.0	58.00	129.5	276.00	3535.0
MARG_OT_3_6_F	640.0	585.884375	900.025817	0.0	127.75	320.5	719.25	12094.0
MARGWORK_0_3_M	640.0	2616.140625	3036.964381	7.0	755.00	1681.5	3320.25	20648.0
MARGWORK_0_3_F	640.0	2834.545312	3327.836932	14.0	833.50	1834.5	3610.50	25844.0
MARG_CL_0_3_M	640.0	1392.973438	1489.707052	4.0	489.50	949.0	1714.00	9875.0
MARG_CL_0_3_F	640.0	2757.050000	2788.776676	30.0	957.25	1928.0	3599.75	21611.0
MARG_AL_0_3_M	640.0	250.889062	453.336594	0.0	47.00	114.5	270.75	5775.0
MARG_AL_0_3_F	640.0	558.098438	1117.642748	0.0	109.00	247.5	568.75	17153.0
MARG_HH_0_3_M	640.0	560.690625	762.578991	0.0	136.50	308.0	642.00	6116.0
MARG_HH_0_3_F	640.0	1293.431250	1585.377936	0.0	298.00	717.0	1710.75	13714.0
MARG_OT_0_3_M	640.0	71.379688	107.897627	0.0	14.00	35.0	79.00	895.0
MARG_OT_0_3_F	640.0	200.742188	309.740854	0.0	43.00	113.0	240.00	3354.0
NON_WORK_M	640.0	510.014063	610.603187	0.0	161.00	326.0	604.50	6456.0
NON_WORK_F	640.0	704.778125	910.209225	5.0	220.50	464.5	853.50	10533.0

checking the duplicate values in the dataset

```
Out[10]: 0
```

There are no duplicate values in the dataset.

checking the null values in the dataset

```
Out[11]: 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
```

There are no null values in the dataset.

(i) Which state has the highest gender ratio and which has the lowest?

```
Out[17]: State
Andhra Pradesh              1.862113
Tamil Nadu                  1.825079
Chhattisgarh                1.820831
Arunachal Pradesh           1.741054
Odisha                      1.737621
Nagaland                    1.713262
Maharashtra                 1.701224
Puducherry                  1.691728
Kerala                      1.663236
Goa                         1.608628
Mizoram                     1.603504
Tripura                     1.597749
Uttarakhand                 1.585126
Karnataka                   1.567885
Madhya Pradesh              1.563246
Manipur                     1.559626
Sikkim                      1.557081
Himachal Pradesh            1.555837
Dadara & Nagar Havelli      1.551275
West Bengal                 1.537645
Andaman & Nicobar Island    1.532148
Gujarat                     1.481824
Jharkhand                   1.466697
Assam                       1.456536
Rajasthan                   1.438257
Chandigarh                  1.428496
Daman & Diu                 1.422185
Jammu & Kashmir             1.360260
Punjab                      1.343180
Bihar                       1.343010
Meghalaya                   1.329504
Uttar Pradesh               1.329492
NCT of Delhi                1.290194
Haryana                     1.283484
Lakshadweep                 1.151993
dtype: float64
```

```
State with the highest gender ratio: Andhra Pradesh
State with the lowest gender ratio: Lakshadweep
```

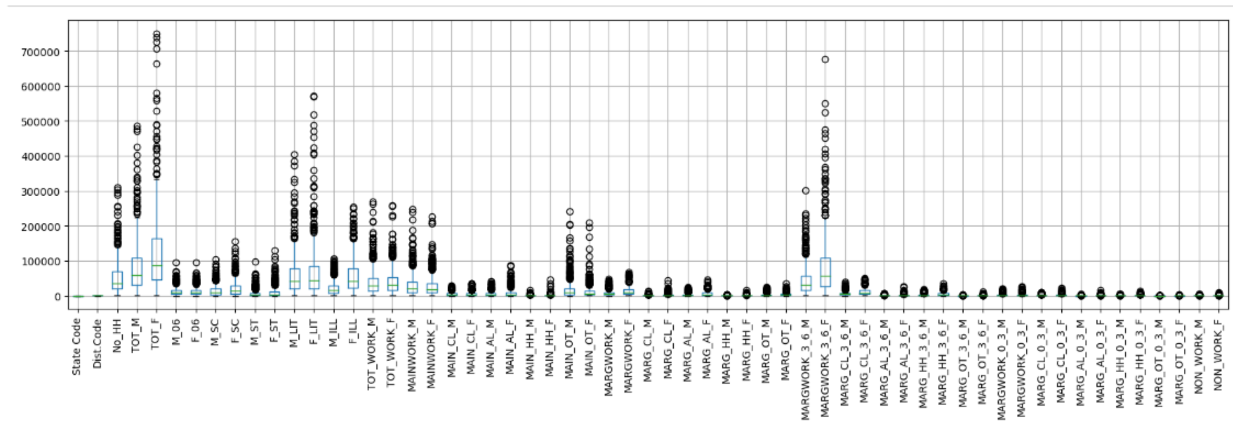
(ii) Which district has the highest & lowest gender ratio?

```
Out[20]: Area Name
Krishna            2.283250
Koraput            2.268763
Virudhunagar       2.225429
West Godavari      2.221849
Baudh              2.215060
                     ...
Baghpat            1.184830
Dhaulpur           1.180761
Mahamaya Nagar     1.180202
Badgam             1.179576
Lakshadweep        1.151993
Length: 635, dtype: float64
```

```
District with the highest gender ratio: Krishna
District with the lowest gender ratio: Lakshadweep
```

Data Preprocessing

14. Boxplot showing outliers in problem 2



True Outliers are kept without treatment for further analysis.

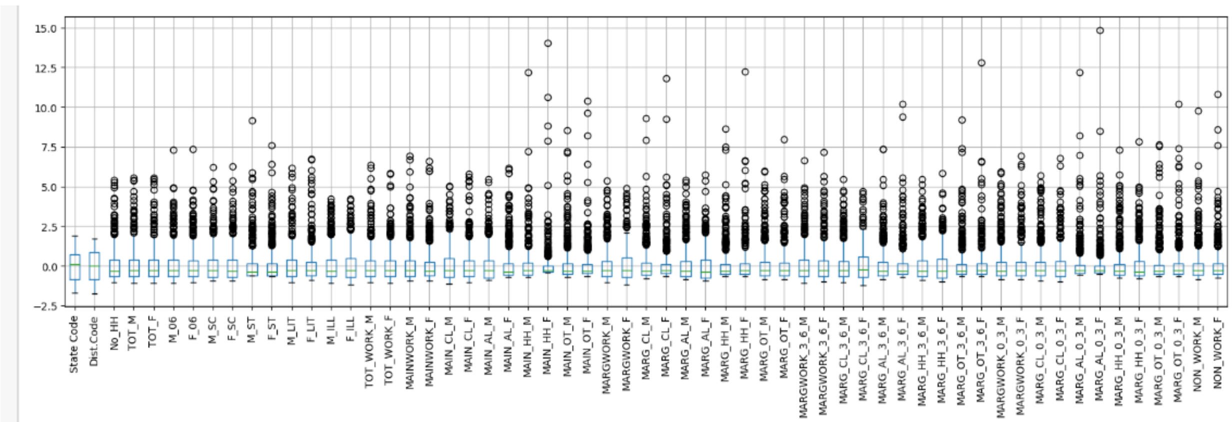
Scaling the Data using Z- score method

Out[24]:

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

5 rows × 59 columns

15. Boxplot After scaling the data using Z – score method.



KMO Test The Kaiser-Meyer-Olkin (KMO) - measure of sampling adequacy (MSA)

```
Out[35]: 0.8039889932779798
```

Generally, if MSA is less than 0.5, PCA is not recommended, since no reduction is expected. On the other hand, MSA > 0.7 is expected to provide a considerable reduction is the dimension and extraction of meaningful components.

PCA

Obtaining the Eigen Vectors

```
Eigen Vectors
 %s [[ 0.16  0.17  0.17 ...  0.13  0.15  0.13]
 [-0.13 -0.09 -0.1   ...  0.05 -0.07 -0.07]
 [-0.    0.06  0.04 ... -0.08  0.11  0.1 ]
 ...
 [ 0.   -0.21  0.04 ... -0.21 -0.04  0.05]
 [-0.08  0.08  0.05 ...  0.09 -0.32  0.22]
 [ 0.64  0.03 -0.22 ... -0.01  0.01 -0.02]]
```

Variance ratio

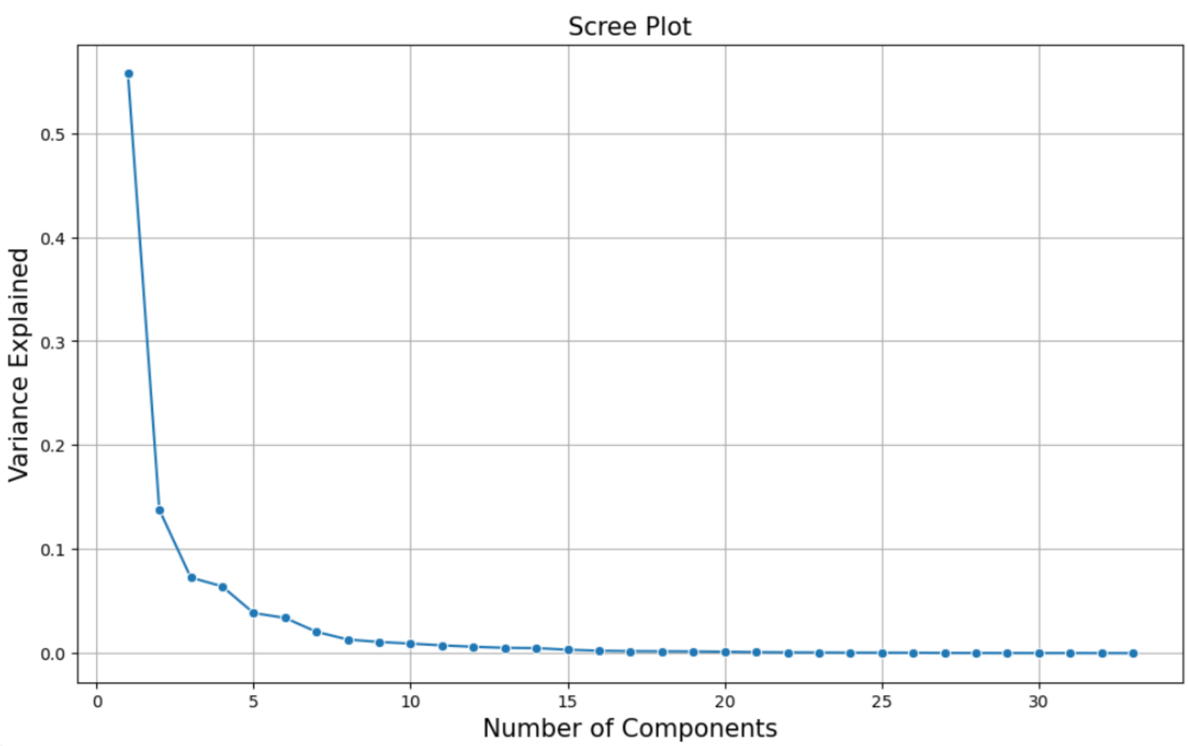
```
[0.56 0.14 0.07 0.06 0.04 0.03 0.02 0.01 0.01 0.01 0.01 0.01 0.01 0.
 0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.   0.
 0.   0.   0.   0.   0.  ]
```

Cumulative Variance ratio

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

We can see above that more than 93% of the variance is explained by 8 Principal Components. Around 98% of the variance is explained by 15 Principal Components. For the scope of this project, take at least 90% explained variance.

16. Scree plot for PCA



Dimensionality reduction from 33 to 6

Obtaining the Eigen Vectors

```
Out[57]: array([[-4.62, -4.77, -5.96, ..., -6.29, -6.22, -5.9 ],
                [ 0.14, -0.11, -0.29, ..., -0.64, -0.67, -0.94],
                [ 0.33,  0.24,  0.37, ...,  0.11,  0.27,  0.35],
                [ 1.54,  1.96,  0.62, ...,  1.37,  1.14,  1.11],
                [ 0.35, -0.15,  0.48, ...,  0.15,  0.06,  0.15],
                [-0.42,  0.42,  0.28, ...,  0.14, -0.12, -0.15]])
```

Eigen vectors for 6 PCA

```
Out[58]: array([[ 0.16,  0.17,  0.17,  0.16,  0.16,  0.15,  0.15,  0.03,  0.03,
                  0.16,  0.15,  0.16,  0.17,  0.16,  0.15,  0.15,  0.12,  0.1 ,
                  0.07,  0.11,  0.07,  0.13,  0.08,  0.12,  0.11,  0.16,  0.16,
                  0.08,  0.05,  0.13,  0.11,  0.14,  0.13,  0.16,  0.15,  0.16,
                  0.16,  0.17,  0.16,  0.09,  0.05,  0.13,  0.11,  0.14,  0.12,
                  0.15,  0.15,  0.15,  0.14,  0.05,  0.04,  0.12,  0.12,  0.14,
                  0.13,  0.15,  0.13],
                [-0.13, -0.09, -0.1 , -0.02, -0.02, -0.05, -0.05,  0.03,  0.03,
                 -0.12, -0.15, -0.01, -0.01, -0.13, -0.09, -0.18, -0.15,  0.06,
                  0.09, -0.03, -0.06, -0.08, -0.08, -0.21, -0.21,  0.09,  0.13,
                  0.27,  0.25,  0.17,  0.14,  0.07,  0.02, -0.09, -0.12, -0.04,
                 -0.11,  0.08,  0.1 ,  0.26,  0.24,  0.16,  0.13,  0.06,  0.01,
                 -0.09, -0.13,  0.15,  0.18,  0.25,  0.24,  0.19,  0.18,  0.08,
                  0.05, -0.07, -0.07],
                [-0.  ,  0.06,  0.04,  0.06,  0.05,  0.  , -0.03, -0.12, -0.14,
                  0.08,  0.12, -0.02, -0.09,  0.05, -0.06,  0.05, -0.06, -0.07,
                 -0.01, -0.25, -0.25,  0.03, -0.06,  0.14,  0.1 , -0.01, -0.05,
                  0.2 ,  0.27, -0.19, -0.27, -0.02, -0.08,  0.11,  0.1 ,  0.06,
                  0.08, -0.02, -0.07,  0.15,  0.26, -0.2 , -0.28, -0.02, -0.08,
                  0.11,  0.1 ,  0.05,  0.02,  0.27,  0.28, -0.14, -0.2 , -0.02,
                 -0.08,  0.11,  0.1 ],
                [-0.13, -0.02, -0.07,  0.01,  0.01,  0.01, -0.03, -0.22, -0.23,
                 -0.04, -0.06,  0.03, -0.08, -0.04, -0.23, -0.07, -0.25, -0.09,
                 -0.29, -0.14, -0.29,  0.15,  0.05, -0.04, -0.12,  0.09, -0.09,
                 -0.06, -0.17,  0.09, -0.11,  0.24,  0.2 ,  0.09,  0.03, -0.  ,
                  0.  ,  0.09, -0.11, -0.04, -0.18,  0.08, -0.14,  0.24,  0.19,
                  0.09,  0.03,  0.09, -0.02, -0.1 , -0.14,  0.13,  0.  ,  0.23,
                  0.21,  0.08,  0.02],
                [-0.01, -0.03, -0.01, -0.05, -0.04, -0.17, -0.16,  0.43,  0.44,
                 -0.01,  0.06, -0.1 , -0.12, -0.02, -0.04, -0.04, -0.08, -0.29,
                 -0.24, -0.21, -0.18, -0.13, -0.14,  0.06,  0.08,  0.06,  0.09,
                 -0.02, -0.06,  0.02,  0.08, -0.06, -0.03,  0.12,  0.17, -0.04,
                  0.  ,  0.05,  0.07, -0.01, -0.06,  0.01,  0.06, -0.07, -0.04,
                  0.11,  0.14,  0.08,  0.13, -0.05, -0.05,  0.06,  0.13, -0.04,
                  0.  ,  0.16,  0.24],
                [ 0.  , -0.07, -0.04, -0.16, -0.15, -0.06, -0.04,  0.22,  0.23,
                 -0.06, -0.05, -0.12, -0.03, -0.  ,  0.11,  0.02,  0.12, -0.01,
                  0.1 , -0.03,  0.02,  0.17,  0.42,  0.02,  0.08, -0.09,  0.02,
                  0.03,  0.09, -0.14, -0.09,  0.09,  0.37, -0.06,  0.  , -0.14,
                 -0.11, -0.1 ,  0.02,  0.01,  0.09, -0.14, -0.08,  0.1 ,  0.38,
                 -0.06,  0.01, -0.06, -0.  ,  0.07,  0.08, -0.12, -0.11,  0.06,
                  0.3 , -0.05, -0.02]])
```

Eigenvalues for 6 PCA

```
Out[59]: array([31.81356474,  7.86942415,  4.15340812,  3.66879058,  2.20652588,
                 1.93827502])
```

Variance Ratio for 6 PCA

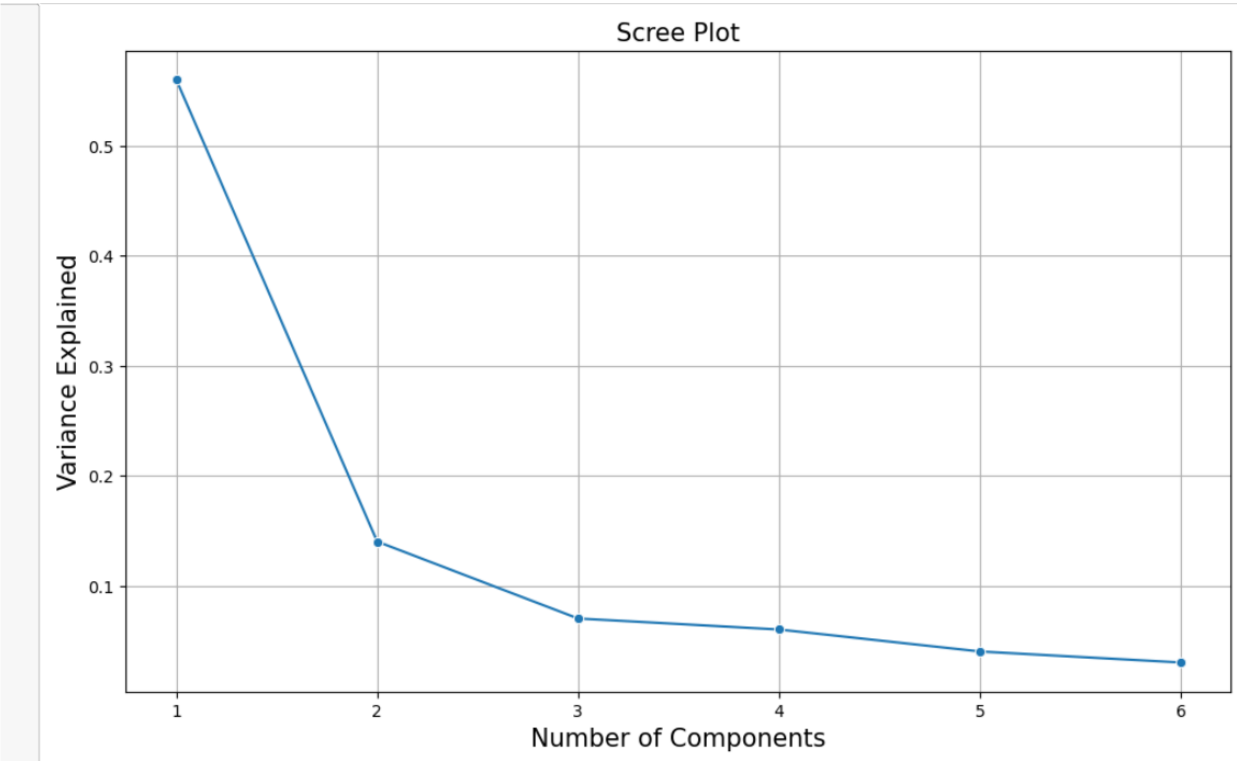
```
Out[61]: array([0.56, 0.14, 0.07, 0.06, 0.04, 0.03])
```

Cumulative Variance Ratio for 6 PCA

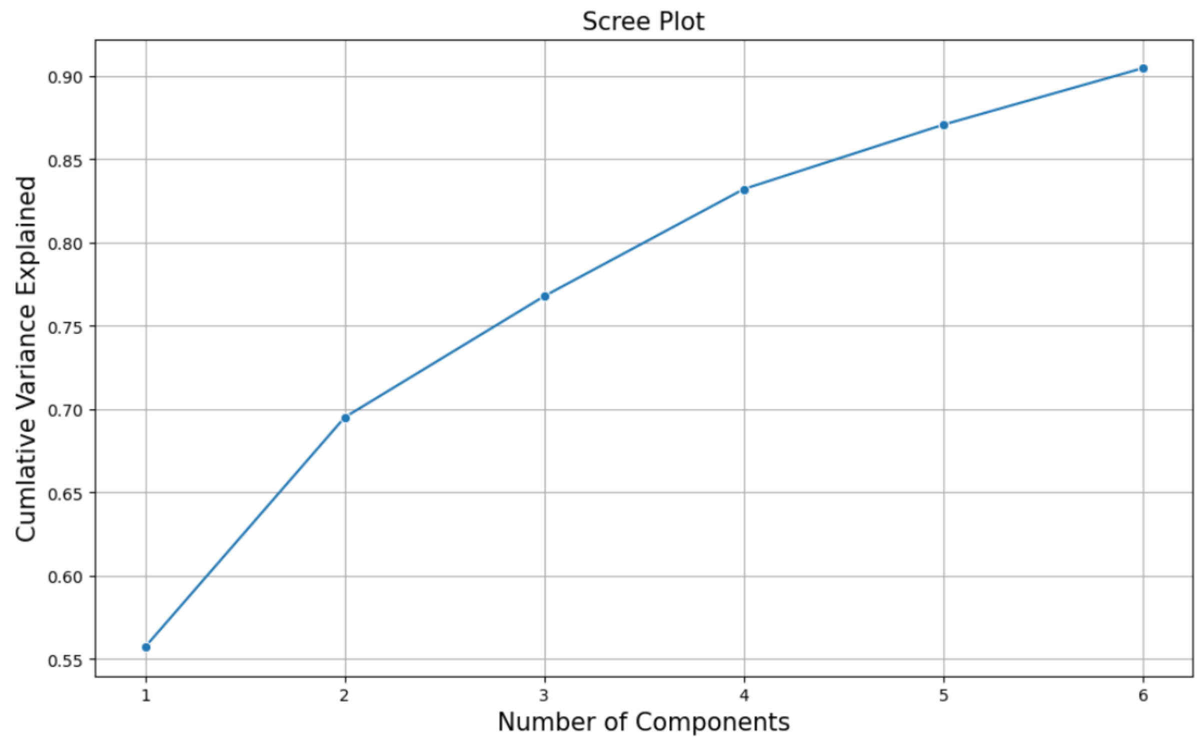
```
Out[62]: array([0.55726063, 0.69510499, 0.76785794, 0.83212212, 0.87077261,
                0.9047243 ])
```

Scree Plot for 6 PCA

17. Scree plot for 6 PCA



18. Screeplot - Cumulative variance vs number of components



create a dataframe of component loading against each field and identify the pattern

Out[66]:

	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	MARC
0	0.16	0.17	0.17	0.16	0.16	0.15	0.15	0.03	0.03	0.16	...	0.15	0.14	0.05	0.04	
1	-0.13	-0.09	-0.10	-0.02	-0.02	-0.05	-0.05	0.03	0.03	-0.12	...	0.15	0.18	0.25	0.24	
2	-0.00	0.06	0.04	0.06	0.05	0.00	-0.03	-0.12	-0.14	0.08	...	0.05	0.02	0.27	0.28	
3	-0.13	-0.02	-0.07	0.01	0.01	0.01	-0.03	-0.22	-0.23	-0.04	...	0.09	-0.02	-0.10	-0.14	
4	-0.01	-0.03	-0.01	-0.05	-0.04	-0.17	-0.16	0.43	0.44	-0.01	...	0.08	0.13	-0.05	-0.05	
5	0.00	-0.07	-0.04	-0.16	-0.15	-0.06	-0.04	0.22	0.23	-0.06	...	-0.06	-0.00	0.07	0.08	

6 rows × 57 columns

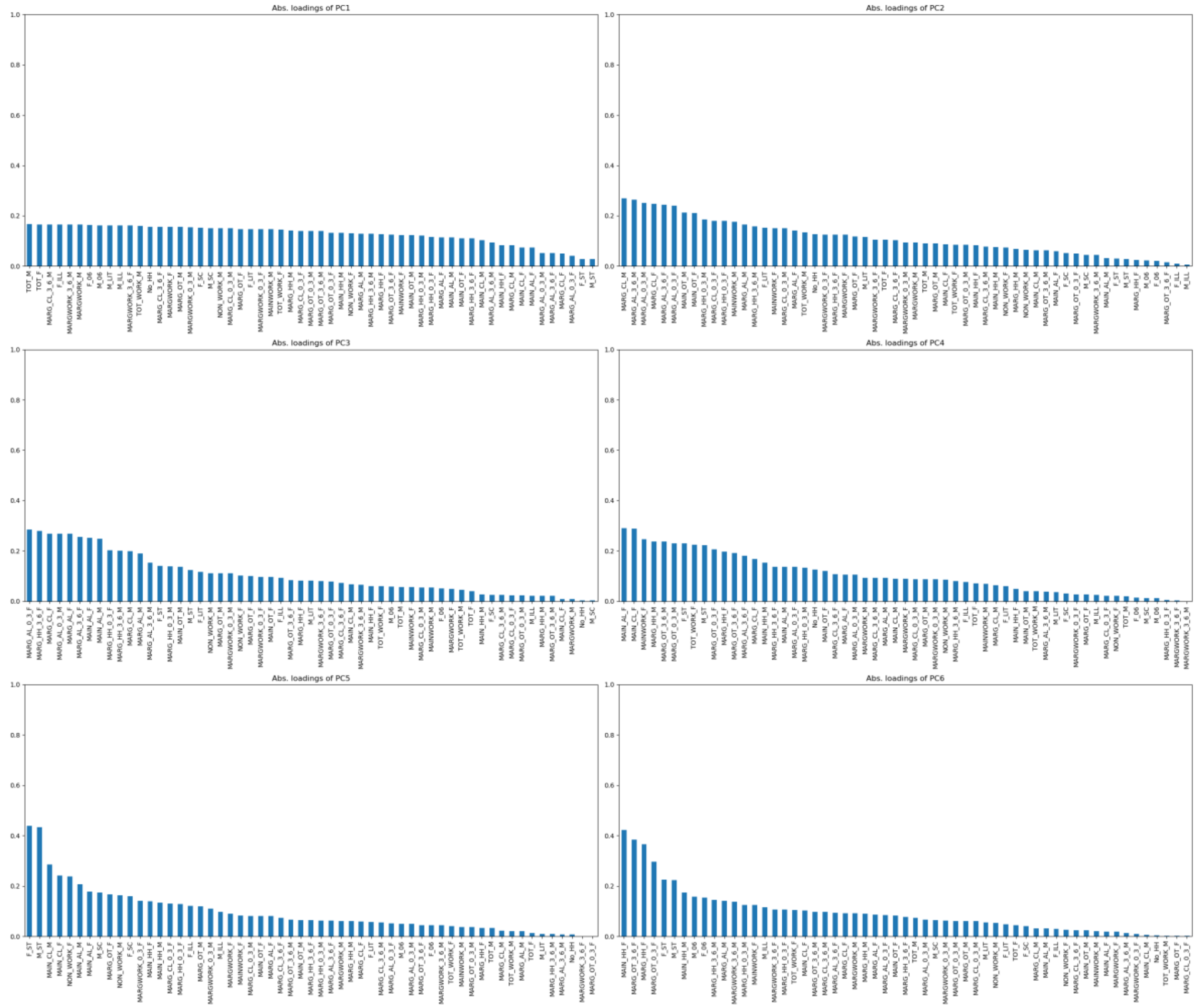
MARG_CL_0_3_F	MARG_AL_0_3_M	MARG_AL_0_3_F	MARG_HH_0_3_M	MARG_HH_0_3_F	MARG_OT_0_3_M	MARG_OT_0_3_F	NON_WORK_M	NON_WORK_F
0.14	0.05	0.04	0.12	0.12	0.14	0.13	0.15	0.13
0.18	0.25	0.24	0.19	0.18	0.08	0.05	-0.07	-0.07
0.02	0.27	0.28	-0.14	-0.20	-0.02	-0.08	0.11	0.10
-0.02	-0.10	-0.14	0.13	0.00	0.23	0.21	0.08	0.02
0.13	-0.05	-0.05	0.06	0.13	-0.04	0.00	0.16	0.24
-0.00	0.07	0.08	-0.12	-0.11	0.06	0.30	-0.05	-0.02

Out[70]:

	PC1	PC2	PC3	PC4	PC5	PC6
No_HH	0.156021	-0.126347	-0.002690	-0.125293	-0.007022	0.004083
TOT_M	0.167118	-0.089677	0.056698	-0.019942	-0.033026	-0.073389
TOT_F	0.165553	-0.104912	0.038749	-0.070873	-0.012847	-0.043647
M_06	0.162193	-0.022095	0.057788	0.011917	-0.050248	-0.157957
F_06	0.162566	-0.020271	0.050126	0.014844	-0.043848	-0.154436
M_SC	0.151358	-0.045111	0.002569	0.012485	-0.173007	-0.064295
F_SC	0.151567	-0.051924	-0.025101	-0.029893	-0.159803	-0.040518
M_ST	0.027234	0.027679	-0.123504	-0.222247	0.433163	0.222591
F_ST	0.028183	0.030223	-0.139769	-0.229754	0.438792	0.225531
M_LIT	0.161993	-0.115355	0.082168	-0.035163	-0.009101	-0.055465
F_LIT	0.146873	-0.153109	0.117098	-0.059559	0.055844	-0.048021
M_ILL	0.161749	-0.006625	-0.021855	0.025348	-0.096580	-0.115234
F_ILL	0.165248	-0.009107	-0.093062	-0.076023	-0.119911	-0.028757
TOT_WORK_M	0.159872	-0.133529	0.045176	-0.040154	-0.019553	-0.001801
TOT_WORK_F	0.145936	-0.085087	-0.059450	-0.225160	-0.040437	0.105162
MAINWORK_M	0.146201	-0.176368	0.054295	-0.068351	-0.036802	0.019283
MAINWORK_F	0.123970	-0.151413	-0.055609	-0.246640	-0.082834	0.123832
MAIN_CL_M	0.103127	0.062415	-0.067399	-0.089769	-0.286039	-0.006170
MAIN_CL_F	0.074540	0.086477	-0.009238	-0.288965	-0.241936	0.102951
MAIN_AL_M	0.113356	-0.031040	-0.247917	-0.136082	-0.205723	-0.031068
MAIN_AL_F	0.073882	-0.058688	-0.251932	-0.290042	-0.177605	0.019240
MAIN_HH_M	0.131573	-0.076021	0.026569	0.152366	-0.134089	0.174465
MAIN_HH_F	0.083383	-0.082477	-0.060523	0.048950	-0.139441	0.422309
MAIN_OT_M	0.123526	-0.212984	0.137378	-0.040289	0.064638	0.023477
MAIN_OT_F	0.111021	-0.210071	0.095634	-0.120391	0.080743	0.083079
MARGWORK_M	0.164615	0.092994	-0.008628	0.093018	0.060244	-0.090762
MARGWORK_F	0.155396	0.125270	-0.049370	-0.088707	0.089202	0.017868

MARG_CL_F	0.049195	0.246547	0.268787	-0.168402	-0.059205	0.092086
MARG_AL_M	0.128599	0.165831	-0.189868	0.091787	0.019422	-0.141605
MARG_AL_F	0.114305	0.140958	-0.267768	-0.106365	0.080527	-0.085120
MARG_HH_M	0.140853	0.068068	-0.021257	0.237985	-0.059971	0.089533
MARG_HH_F	0.127670	0.024216	-0.082504	0.196321	-0.033602	0.365112
MARG_OT_M	0.155263	-0.089442	0.111713	0.087119	0.119121	-0.061066
MARG_OT_F	0.147287	-0.117899	0.100046	0.026729	0.166882	0.001739
MARGWORK_3_6_M	0.164972	-0.043995	0.064423	-0.000026	-0.043834	-0.136253
MARGWORK_3_6_F	0.161253	-0.105502	0.079704	0.003894	0.000537	-0.106900
MARG_CL_3_6_M	0.165502	0.077193	-0.024205	0.092875	0.054073	-0.096708
MARG_CL_3_6_F	0.155647	0.103174	-0.072013	-0.107860	0.073050	0.023773
MARG_AL_3_6_M	0.093014	0.264409	0.153518	-0.038488	-0.007789	0.013477
MARG_AL_3_6_F	0.051536	0.244261	0.256213	-0.179691	-0.061303	0.093993
MARG_HH_3_6_M	0.128576	0.158783	-0.200119	0.080411	0.008457	-0.144061
MARG_HH_3_6_F	0.110646	0.125287	-0.279866	-0.136240	0.064109	-0.076709
MARG_OT_3_6_M	0.139593	0.062262	-0.020618	0.237745	-0.066400	0.097058
MARG_OT_3_6_F	0.124546	0.014766	-0.082794	0.190511	-0.044810	0.384552
MARGWORK_0_3_M	0.154294	-0.093159	0.110285	0.086479	0.108829	-0.062043
MARGWORK_0_3_F	0.146286	-0.125596	0.095667	0.027275	0.141190	0.008962
MARG_CL_0_3_M	0.150126	0.150681	0.054892	0.087433	0.081185	-0.060715
MARG_CL_0_3_F	0.140157	0.180690	0.023982	-0.022290	0.129936	-0.001727
MARG_AL_0_3_M	0.052542	0.251328	0.268330	-0.104686	-0.048849	0.065409
MARG_AL_0_3_F	0.041786	0.240720	0.284956	-0.135716	-0.051895	0.083743
MARG_HH_0_3_M	0.121840	0.185277	-0.138628	0.132544	0.062380	-0.124209
MARG_HH_0_3_F	0.116011	0.180616	-0.202198	0.004051	0.128308	-0.105530
MARG_OT_0_3_M	0.139869	0.084869	-0.022599	0.230038	-0.036390	0.061228
MARG_OT_0_3_F	0.132192	0.050813	-0.078720	0.206201	0.000165	0.295600
NON_WORK_M	0.150376	-0.065365	0.111827	0.084854	0.162862	-0.052387
NON_WORK_F	0.131066	-0.073847	0.102553	0.021124	0.238292	-0.024901

19. Analysis of PCA with the original data



PC1 has the highest magnitude in the parameter TOT_M = 0.17

PC2 has the highest magnitude in the parameter MARG_CL_M = 0.27

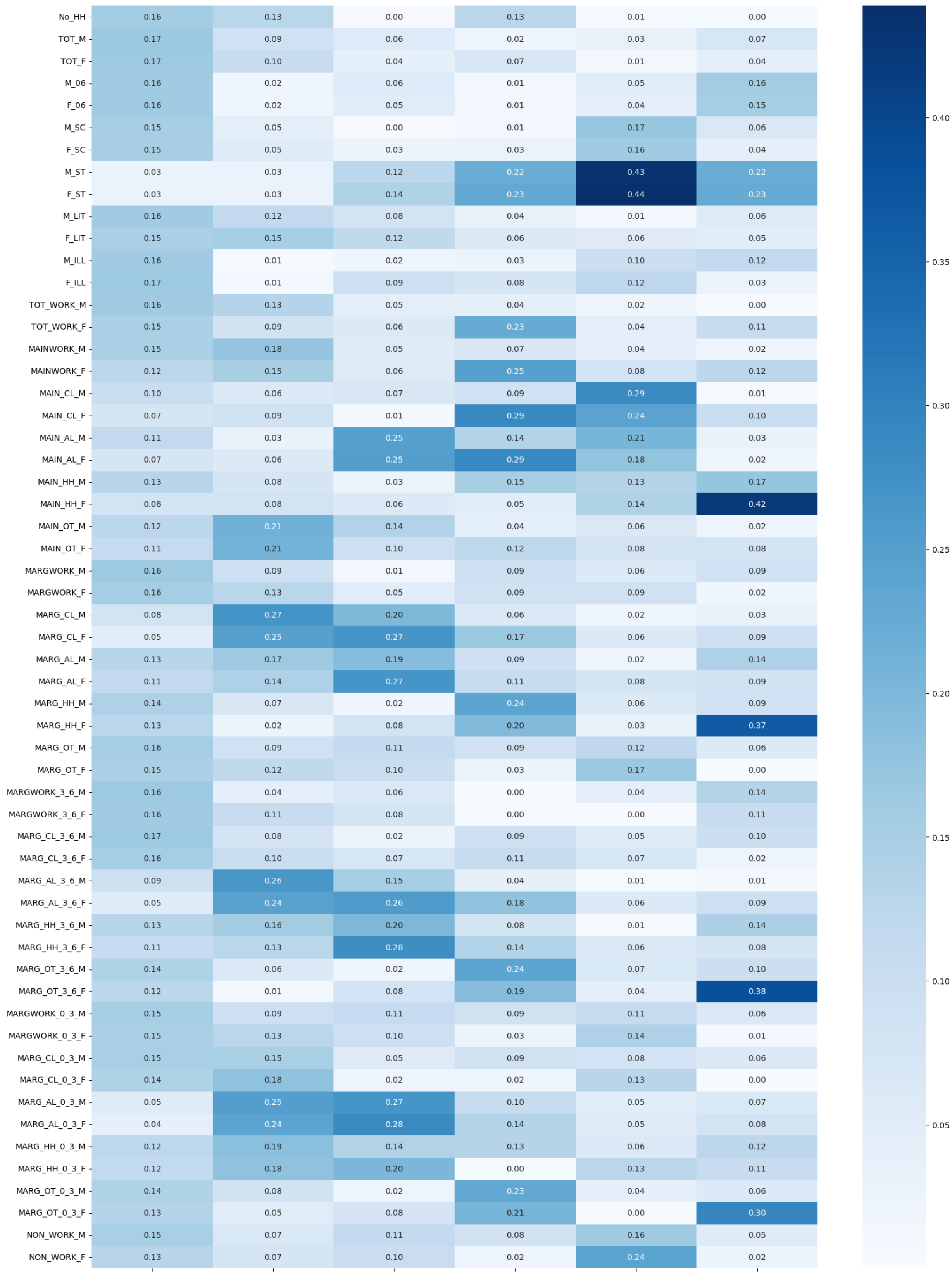
PC3 has the highest magnitude in the parameter MARG_AL_0_3_F = 0.28

PC4 has the highest magnitude in the parameter MAIN_AL_F = 0.29

PC5 has the highest magnitude in the parameter F_ST = 0.44

PC6 has the highest magnitude in the parameter MAIN_HH_F = 0.42

20. Heat Map - PCA



Write linear equation for first PC

```
The linear equation for PC1 is  {'TOT_F'  *     0.17 +  'MARG_CL_3_6_M'  *
0.17 +  'TOT_M'  *     0.17 +  'F_ILL'  *     0.17 +  'No_HH'  *     0.16
+  'M_ILL'  *     0.16 +  'MARG_CL_3_6_F'  *     0.16 +  'MARGWORK_3_6_F'
*     0.16 +  'MARGWORK_3_6_M'  *     0.16 +  'MARG_OT_M'  *     0.16 +  '
MARGWORK_F'  *     0.16 +  'TOT_WORK_M'  *     0.16 +  'M_LIT'  *     0.16
+  'F_06'  *     0.16 +  'M_06'  *     0.16 +  'MARGWORK_M'  *     0.16 +
'F_LIT'  *     0.15 +  'TOT_WORK_F'  *     0.15 +  'MAINWORK_M'  *     0.1
5 +  'MARG_OT_F'  *     0.15 +  'NON_WORK_M'  *     0.15 +  'MARG_CL_0_3_M
'  *     0.15 +  'MARGWORK_0_3_F'  *     0.15 +  'MARGWORK_0_3_M'  *     0
.15 +  'F_SC'  *     0.15 +  'M_SC'  *     0.15 +  'MARG_OT_0_3_M'  *
0.14 +  'MARG_OT_3_6_M'  *     0.14 +  'MARG_CL_0_3_F'  *     0.14 +  'MAR
G_HH_M'  *     0.14 +  'MARG_HH_3_6_M'  *     0.13 +  'MARG_OT_0_3_F'  *
0.13 +  'NON_WORK_F'  *     0.13 +  'MARG_HH_F'  *     0.13 +  'MARG_AL_M'
*     0.13 +  'MAIN_HH_M'  *     0.13 +  'MARG_HH_0_3_M'  *     0.12 +  'M
AIN_OT_M'  *     0.12 +  'MAINWORK_F'  *     0.12 +  'MARG_OT_3_6_F'  *
0.12 +  'MARG_HH_0_3_F'  *     0.12 +  'MAIN_OT_F'  *     0.11 +  'MARG_HH
_3_6_F'  *     0.11 +  'MARG_AL_F'  *     0.11 +  'MAIN_AL_M'  *     0.11
+  'MAIN_CL_M'  *     0.1 +  'MARG_AL_3_6_M'  *     0.09 +  'MARG_CL_M'  *
0.08 +  'MAIN_HH_F'  *     0.08 +  'MAIN_AL_F'  *     0.07 +  'MAIN_CL_F'
*     0.07 +  'MARG_AL_0_3_M'  *     0.05 +  'MARG_AL_3_6_F'  *     0.05 +
'MARG_CL_F'  *     0.05 +  'MARG_AL_0_3_F'  *     0.04 +  'F_ST'  *     0.
03 +  'M_ST'  *     0.03}
```