Business Report

ML - 1

By Gagandeep Singh

Problem Statement:

Clustering:

Digital Ads Data:
The ads24x7 is a Digital Marketing company which has now got seed funding of $10 Million. They are expanding their wings in Marketing Analytics. They collected data from their Marketing Intelligence team and now wants you (their newly appointed data analyst) to segment the 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.

Part 1: Clustering: Define the problem and perform Exploratory Data Analysis

Data Dictionary

1. Timestamp - The Timestamp of the particular Advertisement.
2. InventoryType - The Inventory Type of the particular Advertisement. Format 1 to 7. This is a Categorical Variable.
3. Ad - Length - The Length Dimension of the particular Advertisement.
4. Ad- Width - The Width Dimension of the particular Advertisement.
5. Ad Size - The Overall Size of the particular Advertisement. Length*Width.
6. Ad Type - The type of the particular Advertisement. This is a Categorical Variable.
7. Platform - The platform in which the particular Advertisement is displayed. Web, Video or App. This is a Categorical Variable.

8. Device Type - The type of the device which supports the particular Advertisement. This is a Categorical Variable.
9. Format - The Format in which the Advertisement is displayed. This is a Categorical Variable.
10. Available_Impressions - How often the particular Advertisement is shown. An impression is counted each time an Advertisement is shown on a search result page or other site on a Network.
11. Matched_Queries - Matched search queries data is pulled from Advertising Platform and consists of the exact searches typed into the search Engine that generated clicks for the particular Advertisement.
12. Impressions - The impression count of the particular Advertisement out of the total available impressions.
13. Clicks - It is a marketing metric that counts the number of times users have clicked on the particular advertisement to reach an online property.
14. Spend - It is the amount of money spent on specific ad variations within a specific campaign or ad set. This metric helps regulate ad performance.
15. Fee - The percentage of the Advertising Fees payable by Franchise Entities.
16. Revenue - It is the income that has been earned from the particular advertisement.
17. CTR - CTR stands for "Click through rate". CTR is the number of clicks that your ad receives divided by the number of times your ad is shown. Formula used here is CTR = Total Measured Clicks / Total Measured Ad Impressions x 100. Note that the Total Measured Clicks refers to the 'Clicks' Column and the Total Measured Ad Impressions refers to the 'Impressions' Column.
18. CPM - CPM stands for "cost per 1000 impressions." Formula used here is CPM = (Total Campaign Spend / Number of Impressions) * 1,000. Note that the Total Campaign Spend refers to the 'Spend' Column and the Number of Impressions refers to the 'Impressions' Column.
19. CPC - CPC stands for "Cost-per-click". Cost-per-click (CPC) bidding means that you pay for each click on your ads. The Formula used here is CPC = Total Cost (spend) / Number of Clicks. Note that the Total Cost (spend) refers to the 'Spend' Column and the Number of Clicks refers to the 'Clicks' Column.

Data Overview

Structure of the Data:

- Number of Rows: 23066
- Number of Columns: 19
- Memory Usage: 3.3+ MB
- Range Index: 0 to 23065
- Data Types: Float, Int, and Object

Data Type:

The different datatypes in the dataset are as follows

a.There are 7 columns in the with int64 data type

b.There are 6 columns in the with object data type

c.There are 6 columns in the with float64 data type

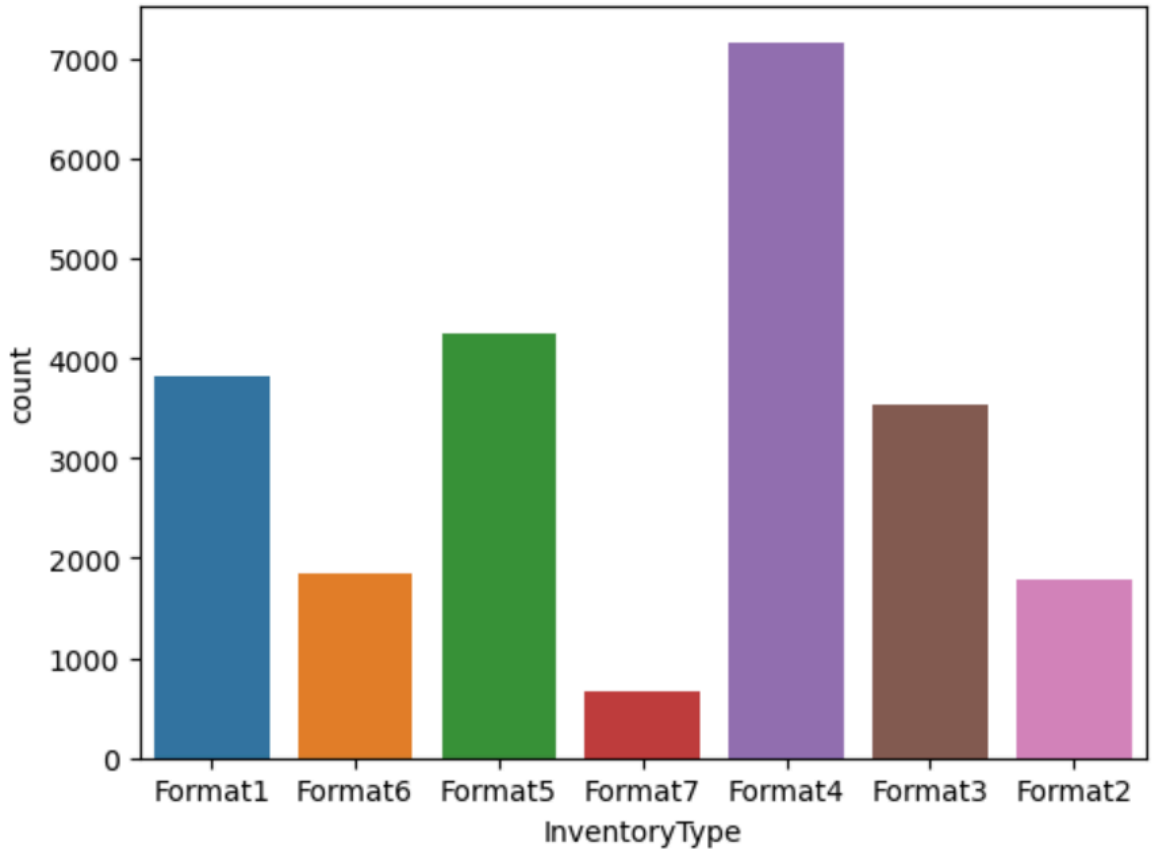
Statistical Summary

	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Rev
count	23066.000000	23066.000000	23066.000000	2.306600e+04	2.306600e+04	2.306600e+04	23066.000000	23066.000000	23066.000000	23066.00
mean	385.163097	337.896037	96674.468048	2.432044e+06	1.295099e+06	1.241520e+06	10678.518816	2706.625689	0.335123	1924.25
std	233.651434	203.092885	61538.329557	4.742888e+06	2.512970e+06	2.429400e+06	17353.409363	4067.927273	0.031963	3105.23
min	120.000000	70.000000	33600.000000	1.000000e+00	1.000000e+00	1.000000e+00	1.000000	0.000000	0.210000	0.00
25%	120.000000	250.000000	72000.000000	3.367225e+04	1.828250e+04	7.990500e+03	710.000000	85.180000	0.330000	55.36
50%	300.000000	300.000000	72000.000000	4.837710e+05	2.580875e+05	2.252900e+05	4425.000000	1425.125000	0.350000	926.33
75%	720.000000	600.000000	84000.000000	2.527712e+06	1.180700e+06	1.112428e+06	12793.750000	3121.400000	0.350000	2091.33
max	728.000000	600.000000	216000.000000	2.759286e+07	1.470202e+07	1.419477e+07	143049.000000	26931.870000	0.350000	21276.18

- In the "Ad Size" column, the mean is greater than the median, indicating a positive skewed distribution.
- Similarly, the "spend" column also displays a positive skewed distribution, with the mean exceeding the median.
- Moreover, the "clicks" column exhibits a highly positive skewed distribution, as evidenced by the mean being substantially greater than the median.

Univariate Analysis

Inventory Type




```
InventoryType
Format4    7165
Format5    4249
Format1    3814
Format3    3540
Format6    1850
Format2    1789
Format7     659
Name: count, dtype: int64
```

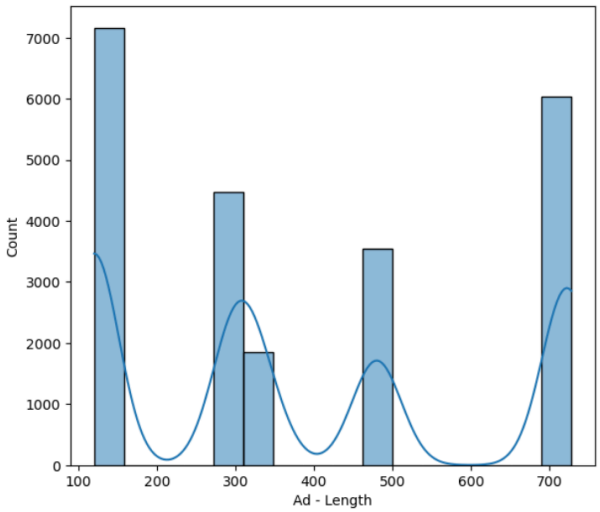
Observations -

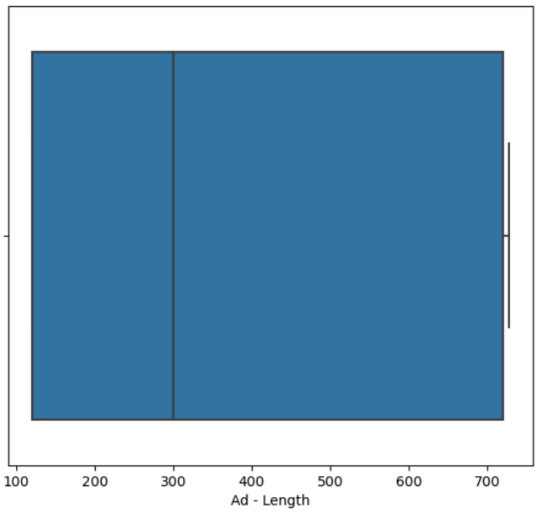
* The most frequently used type of inventory is 'Format4', with a count of 7165.

* 'Format7' represents the least utilized inventory type, with only 659 instances.

* Following 'Format4' and 'Format5', the next commonly employed inventory types are 'Format1' and 'Format3'.

Ad - Length



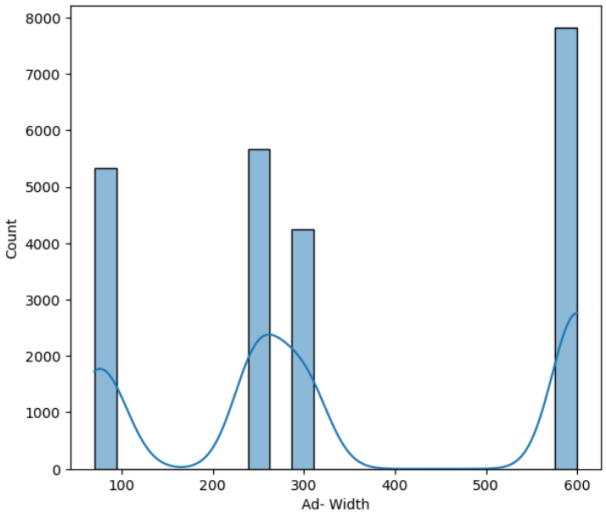


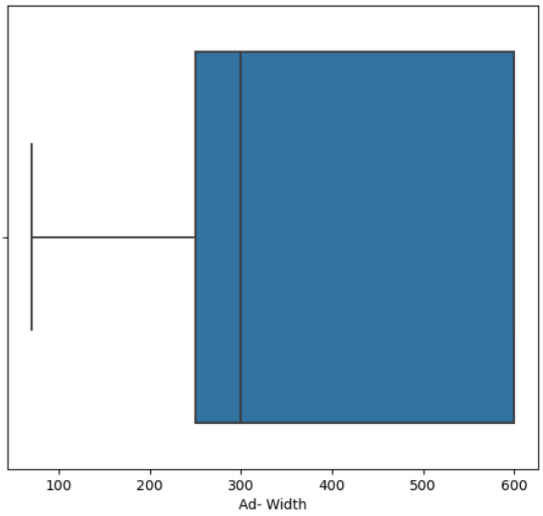

```
Ad - Length
120    7165
300    4473
720    4249
480    3540
336    1850
728    1789
Name: count, dtype: int64
```

Observations -

* The most common ad length dimension is 120, occurring 7165 times.
* The ad length with the least occurrence is of dimension 728.
* Common ad length dimensions include 300 and 720.

Ad- Width



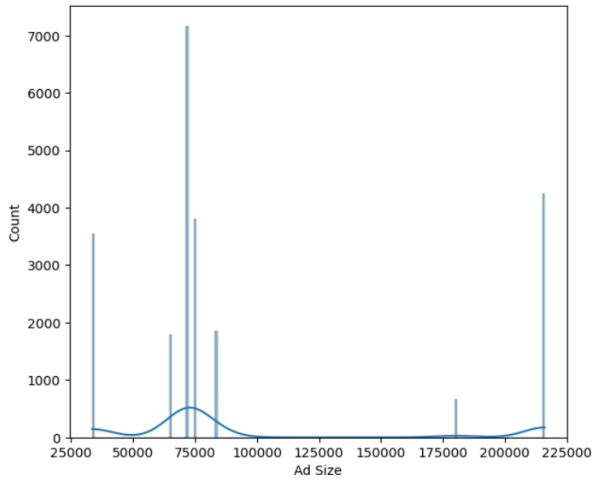


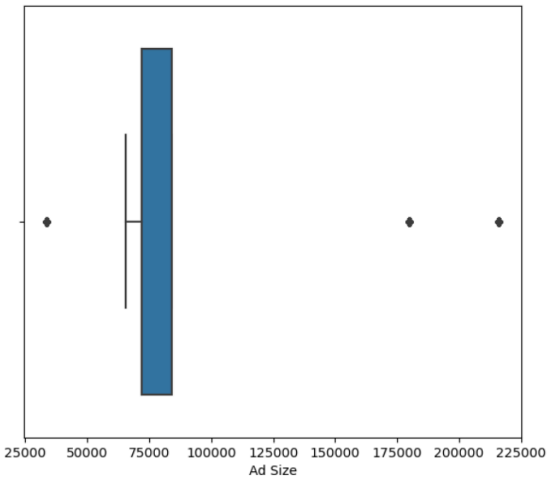

```
Ad- Width
600    7824
250    5664
300    4249
70     3540
90     1789
Name: count, dtype: int64
```

Observations -

* The most common ad width dimension is 600, occurring 7824 times.
* The ad width with the least occurrence is of dimension 1789.
* Common ad width dimensions include 300 and 250.
* Ad- Width distribution is right-skewed.

Ad Size



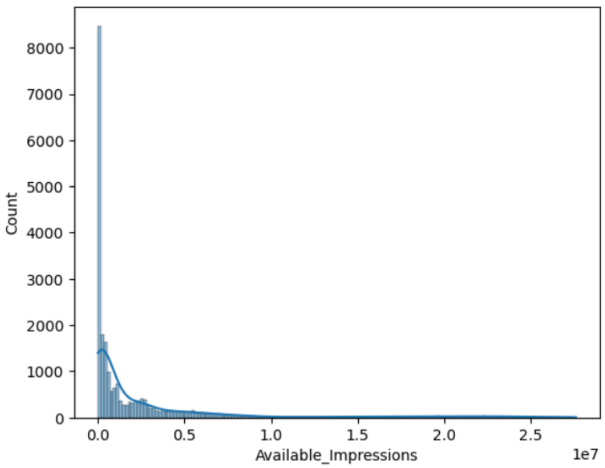


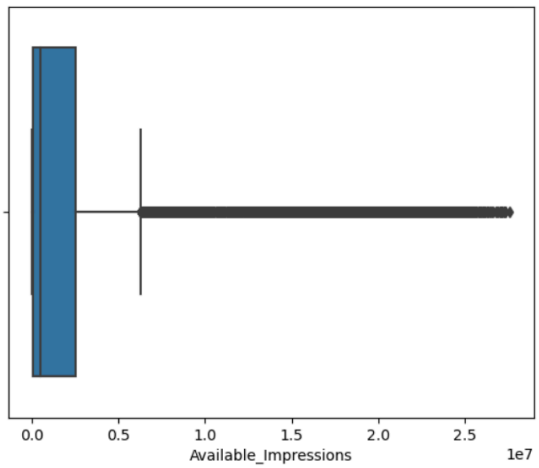

```
Ad Size
72000     7165
216000    4249
75000     3814
33600     3540
84000     1850
65520     1789
180000     659
Name: count, dtype: int64
```

Observations -

* The most frequent ad size is 72,000.
* The least occurring ad size is 180,000, observed 659 times.
* Common ad sizes include 216,000.
* There are outliers present in the data.

Available_Impressions



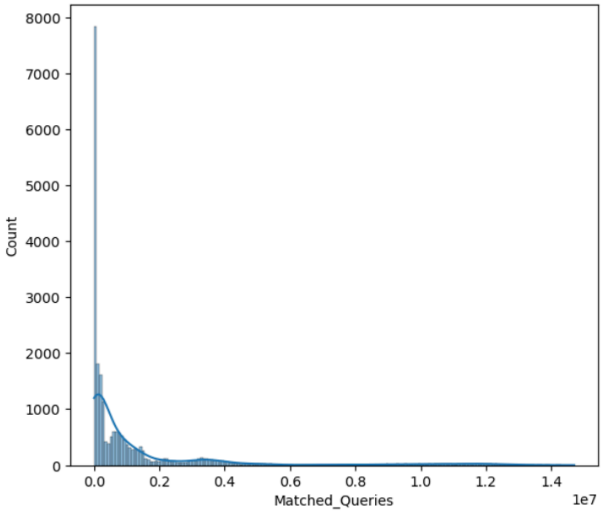


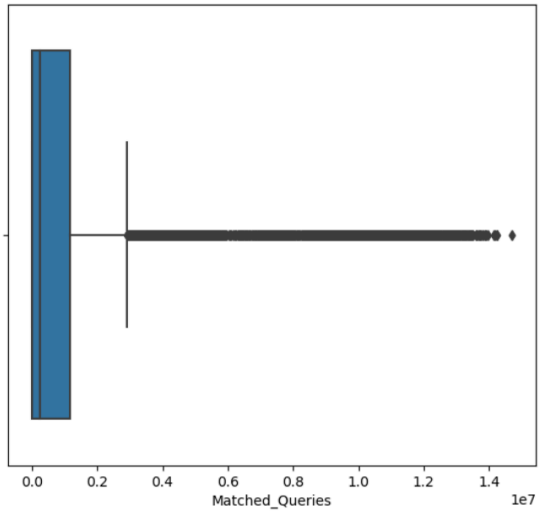

```
Available_Impressions
7           33
9           25
5           24
3           23
11          23
            ..
1950296      1
3990532      1
483612       1
1034014      1
114          1
Name: count, Length: 21560, dtype: int64
```

Observations -

* The most frequent observation is 7, occurring 33 times.
* There are several unique observations that occur only once, such as 1,950,296; 3,990,532; 483,612; 1,034,014, etc.

Matched_Queries



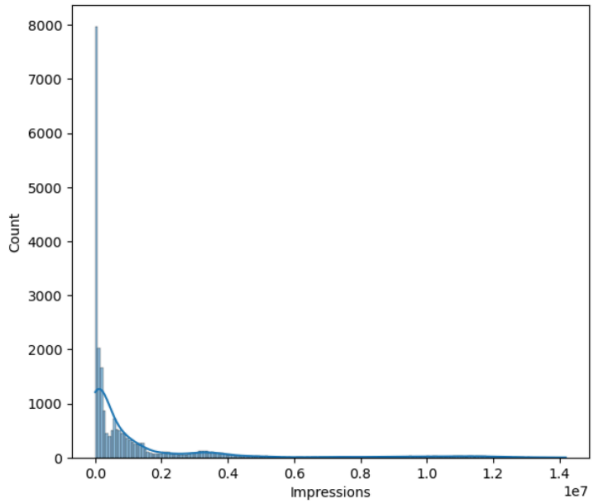


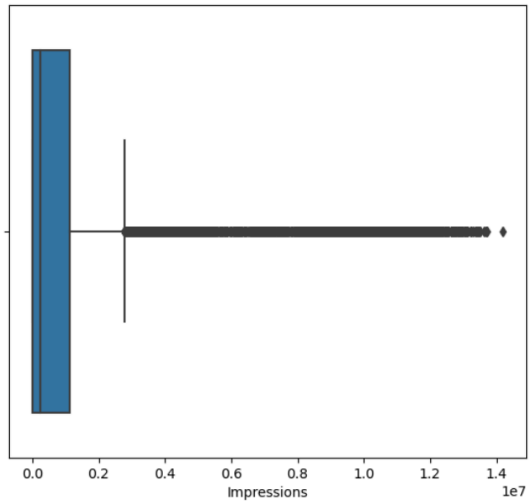

```
Matched_Queries
5          50
4          49
3          41
2          40
6          33
           ..
537979      1
613290      1
1335760     1
213348      1
197         1
Name: count, Length: 20919, dtype: int64
```

Observations -

* Outliers are present in the "Matched_Queries" column.
* The most common value in the "Matched_Queries" column is 5.

Impressions



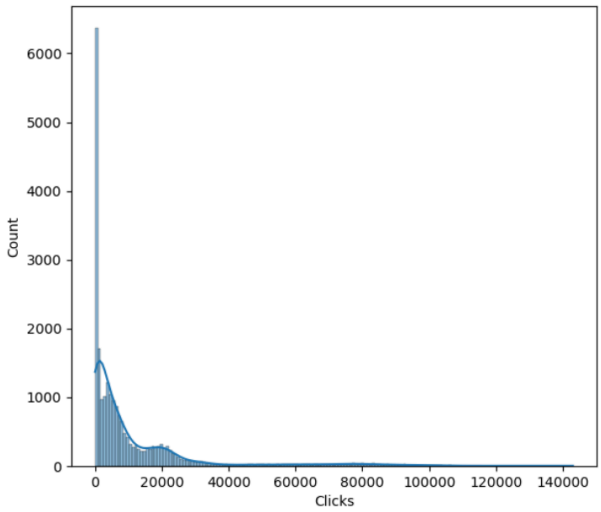


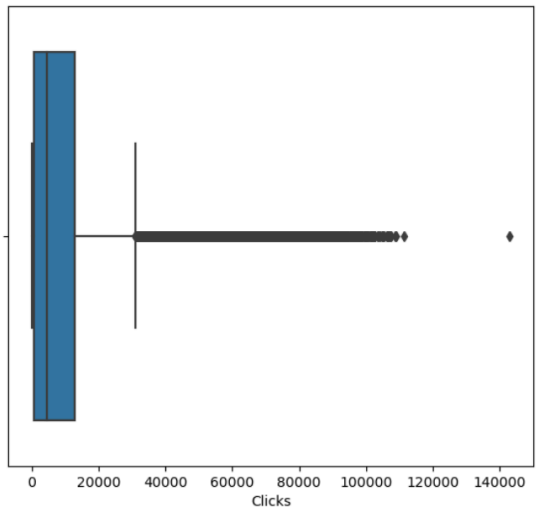

```
Impressions
2          57
4          53
5          49
3          38
7          37
           ..
185375      1
702352      1
126782      1
151283      1
143         1
Name: count, Length: 20405, dtype: int64
```

Observations -

* Outliers are present in the "Imressions" column.
* The most common value in the "Impressions" column is 2.

Clicks



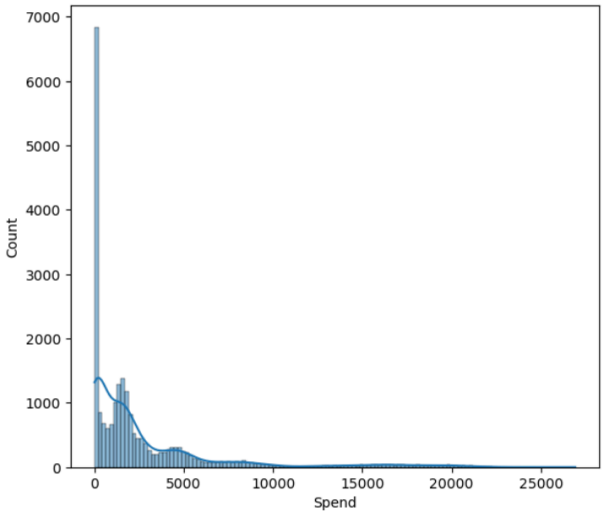


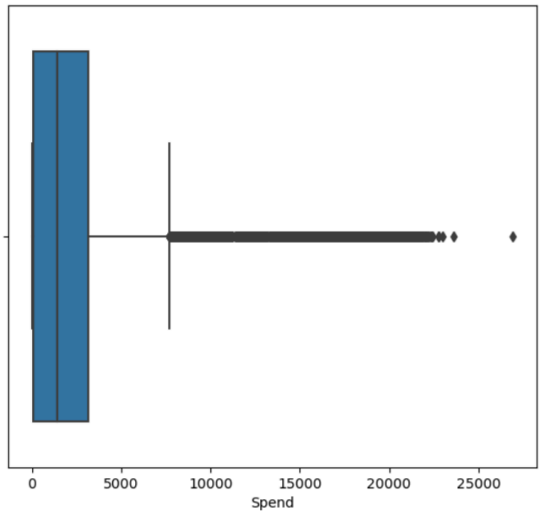

```
Clicks
1         540
2         162
3          76
4          58
6          50
          ...
5430        1
5222        1
27306       1
22512       1
1201        1
Name: count, Length: 12752, dtype: int64
```

Observations -

* Advertisements with a click count of one are the most frequently occurring, with 540 instances.
* The majority of advertisements have click counts of 1, 2, 3, 4, and 6.

Spend



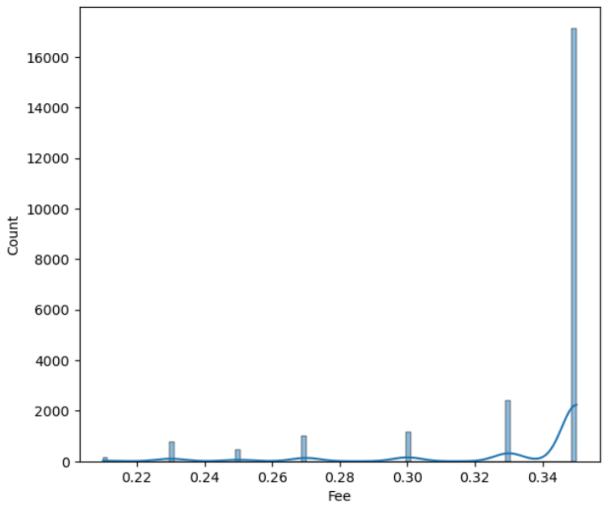


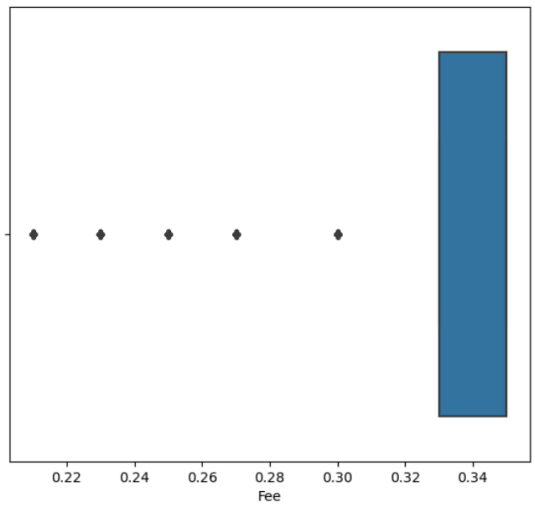

```
Spend
0.00       97
0.04       56
0.03       46
0.05       43
0.07       39
           ..
1394.29     1
1394.25     1
1394.17     1
1393.84     1
1.43        1
Name: count, Length: 20467, dtype: int64
```

Observations -

* There are 97 advertisements with no amount of money spent.
* The maximum amount spent on an advertisement is 26931.

Fee



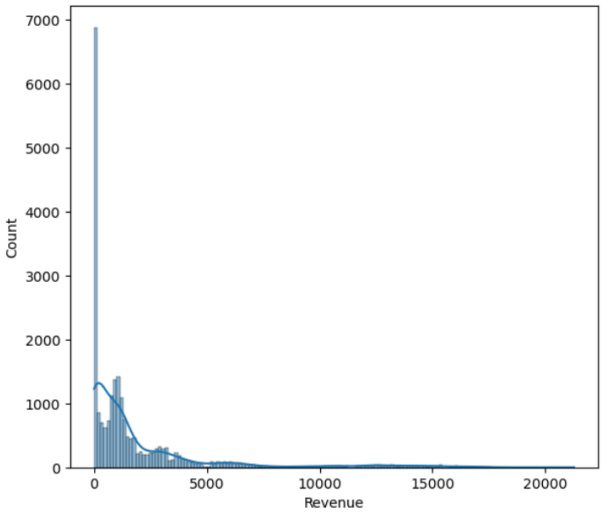


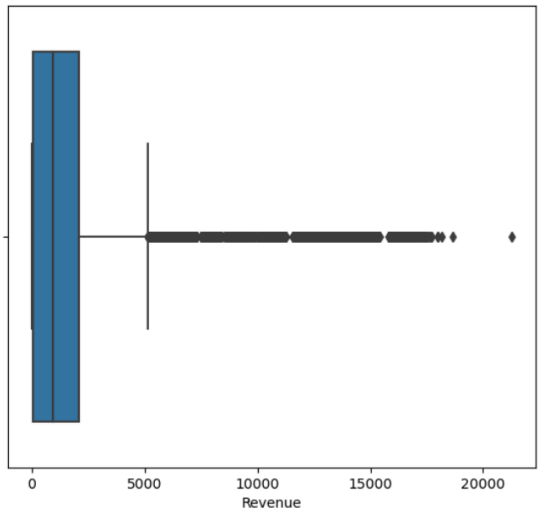

```
Fee
0.35    17141
0.33     2408
0.30     1169
0.27      989
0.23      752
0.25      445
0.21      162
Name: count, dtype: int64
```

Observations-

* The most frequent and maximum percentage of advertising fees is 0.35, with 17141 instances.
* The least occurring percentage is 0.21, with 162 instances.

Revenue

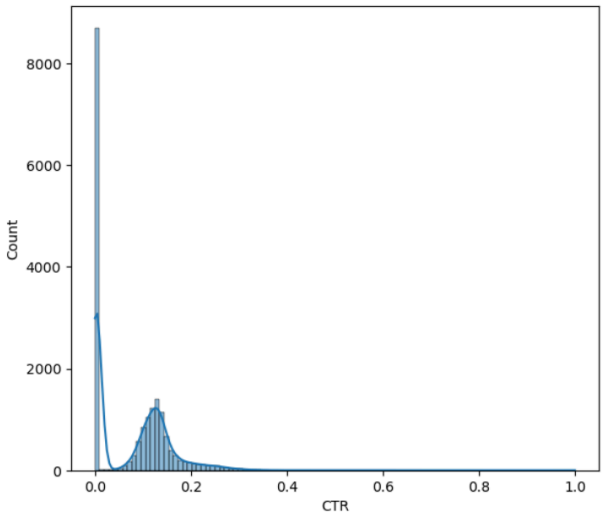


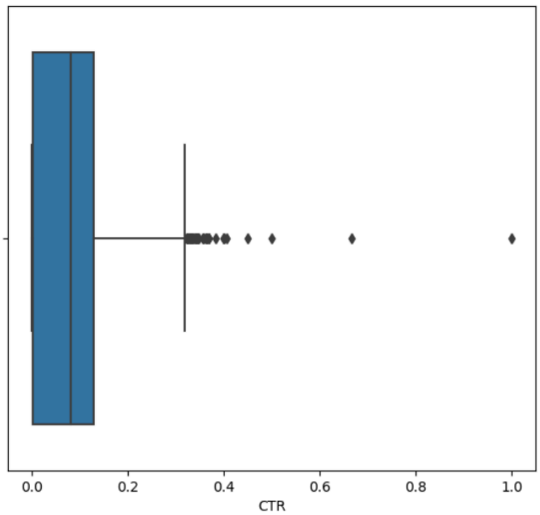


Observations-

* Approximately 100 ads have generated zero revenue.
* There are 49 ads that generated 0.0260 in revenue.
* The maximum revenue generated by an ad is 21276.18.
* There are few outliers present.

CTR



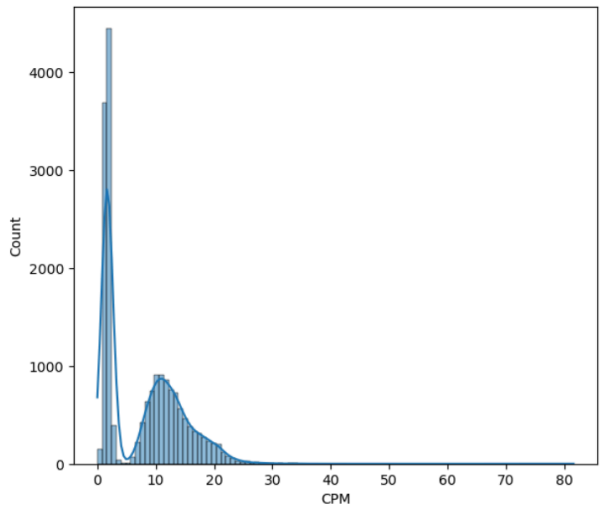


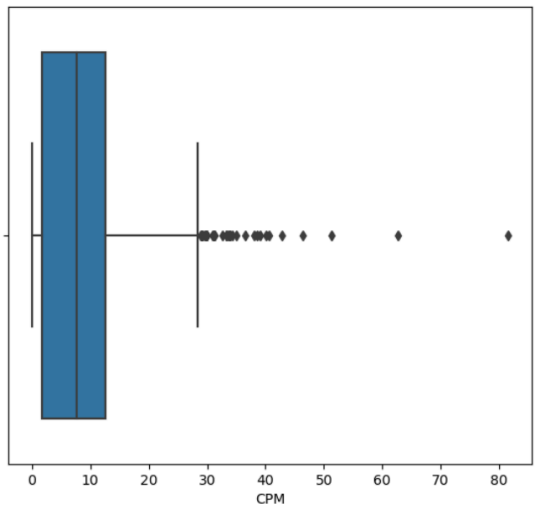
```
CTR
0.0024    630
0.0025    598
0.0023    588
0.0022    507
0.0026    492
          ...
0.2693      1
0.3230      1
0.0682      1
0.0539      1
0.1741      1
Name: count, Length: 2066, dtype: int64
```

Observations-

* Ads with a click-through rate (CTR) of 0.0024 have the highest count, with 630 instances.
* The least common CTR values are 0.2693, 0.3230, 0.0682, 0.0539, and 0.1741, each occurring only once.

CPM



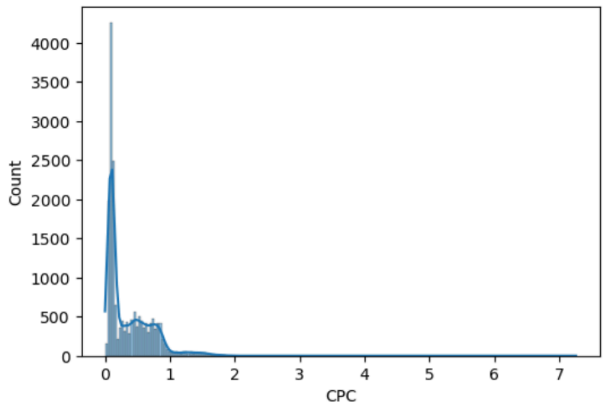


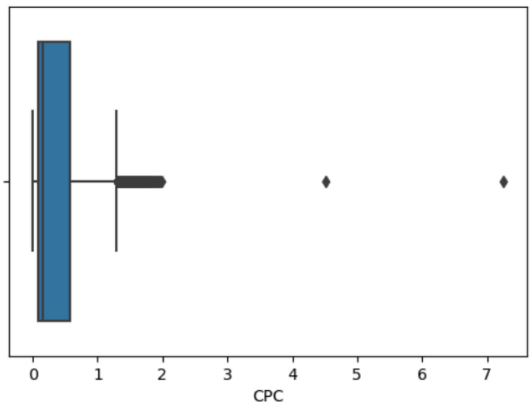
```
CPM
1.66     123
1.62     103
1.69     102
1.64     101
1.74      99
        ...
15.28      1
0.63       1
25.39      1
5.30       1
15.95      1
Name: count, Length: 2084, dtype: int64
```

Observations -

* Ads with a CPM of 1.66 have the highest count, with 123 instances.
* The least common CPM values are 15.28, 0.63, 25.39, 5.30, and 15.95, each occurring only once.

CPC



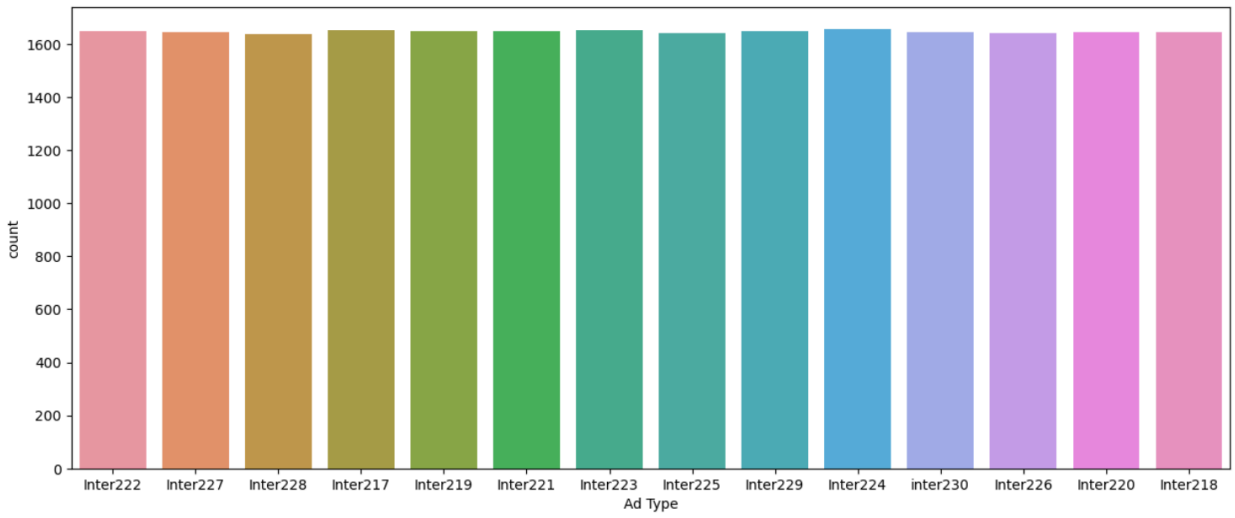


```
CPC
0.09    1577
0.10    1431
0.08    1247
0.07     963
0.11     872
        ...
1.87       1
1.71       1
1.94       1
4.51       1
1.96       1
Name: count, Length: 194, dtype: int64
```

Observations -

* Ads with a CPC of 1.66 have the highest count, with 123 instances.
* The least common CPM values are 15.28, 0.63, 25.39, 5.30, and 15.95, each occurring only once.

Ad Type

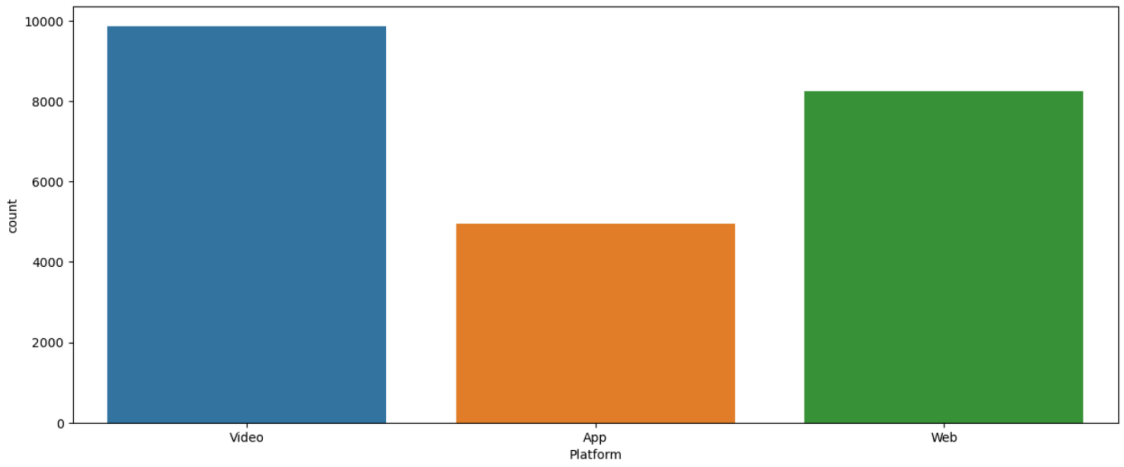


```
Ad Type
Inter224    1658
Inter217    1655
Inter223    1654
Inter219    1650
Inter221    1650
Inter222    1649
Inter229    1648
Inter227    1647
Inter218    1645
inter230    1644
Inter220    1644
Inter225    1643
Inter226    1640
Inter228    1639
Name: count, dtype: int64
```

Observations -

* The Inter224 ad type has the highest count, with 1,658 occurrences.
* The Inter228 ad type has the lowest count, with 1,639 occurrences.
* The counts of the ad types do not significantly differ, as they all fall within the range of 1,639 to 1,658 occurrences.

Platform

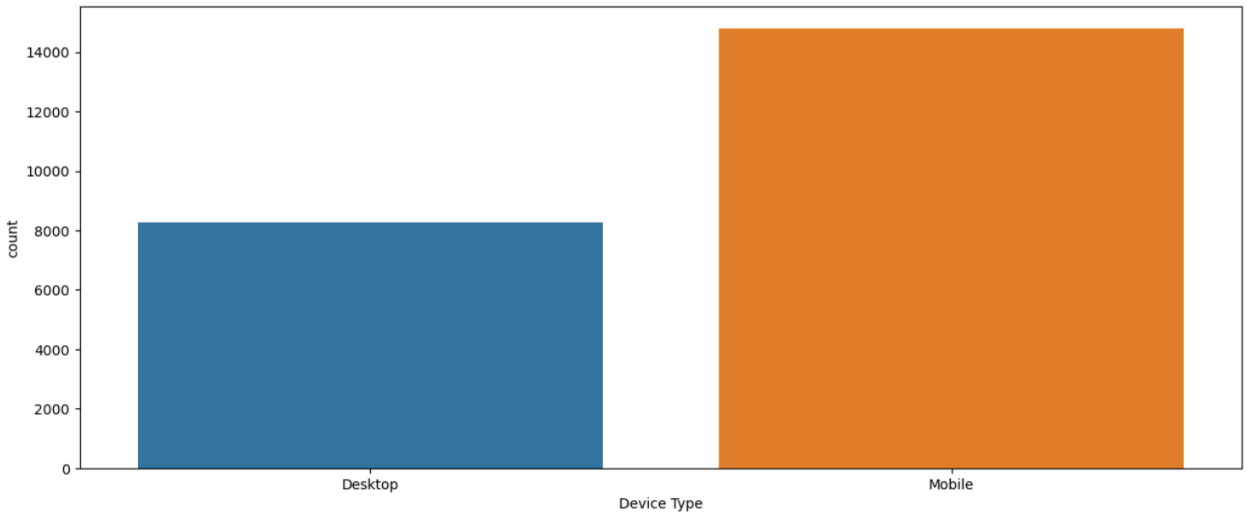


```
Platform
Video    9873
Web      8251
App      4942
Name: count, dtype: int64
```

Observations -

* Video advertisements have the highest count, with 9,873 occurrences.
* The app type has the lowest count of advertisement plays, with 4,942 occurrences.

Device Type

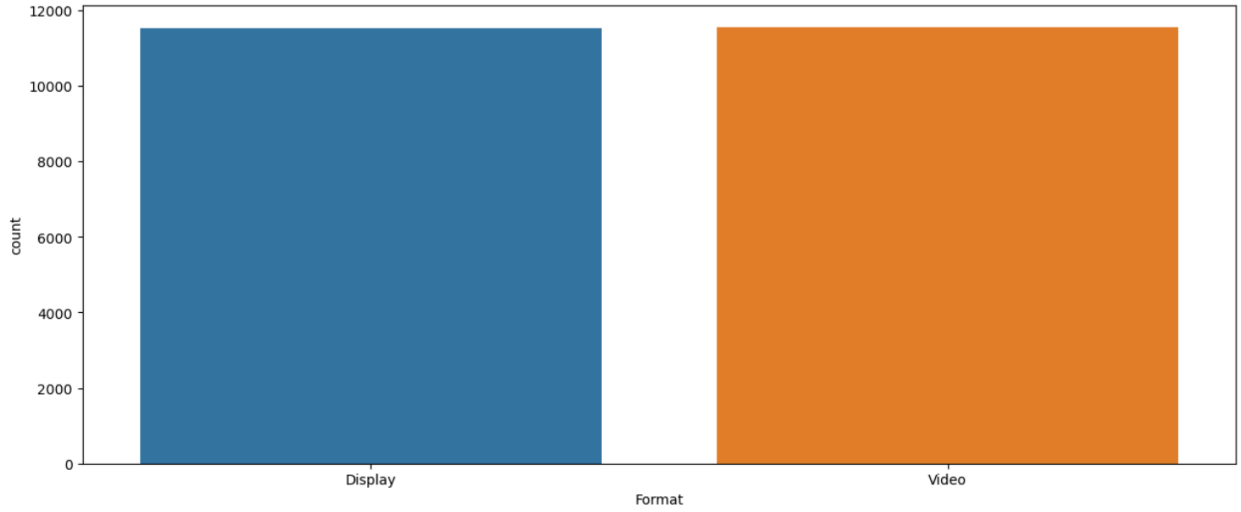



```
Device Type
Mobile     14806
Desktop     8260
Name: count, dtype: int64
```

Observations -

* Mobile platforms support the majority of the advertisements, with 14,806 instances.
* Desktop platforms support a comparatively lower number of advertisements, with only 8,260 instances.

Format

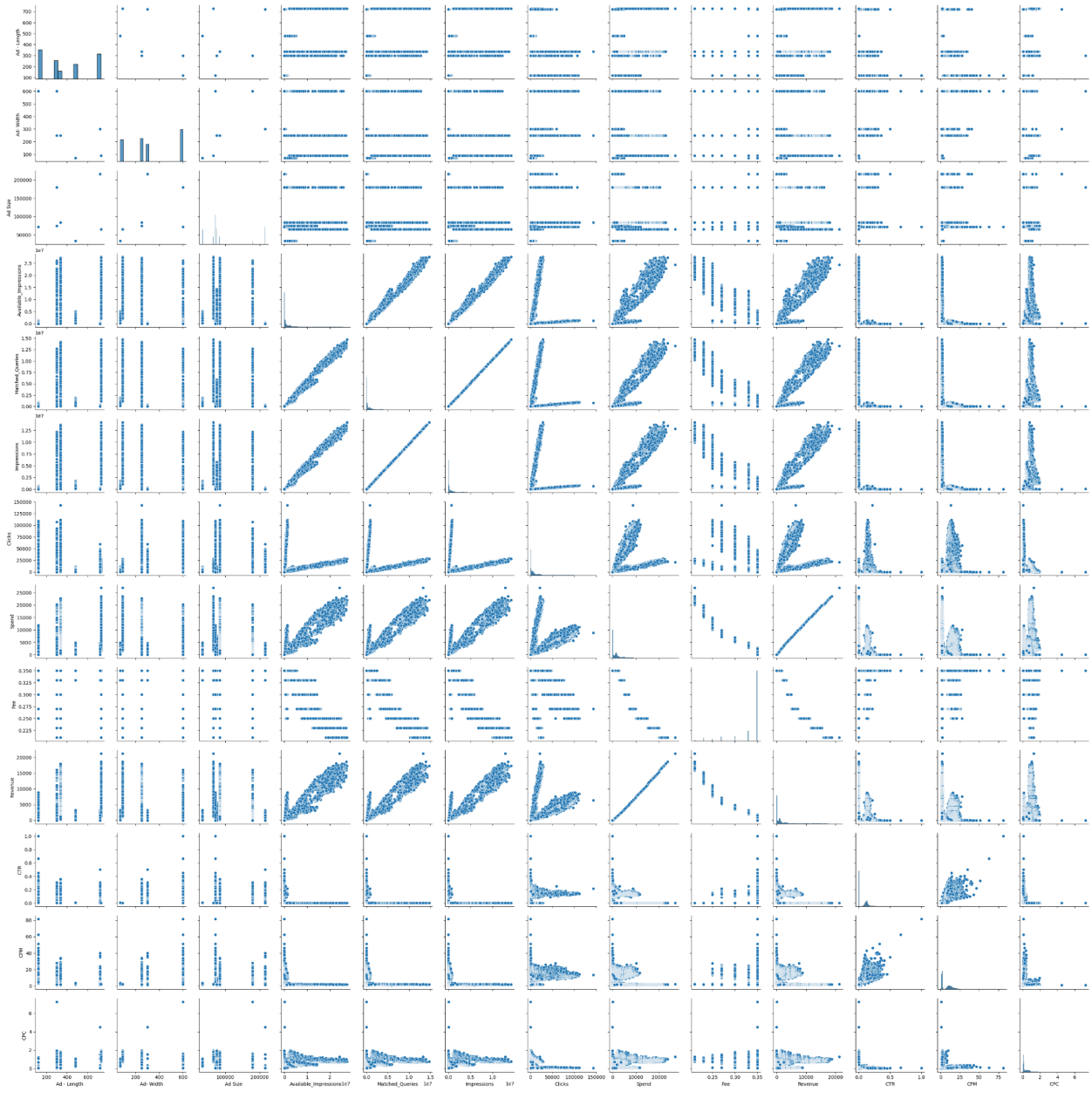


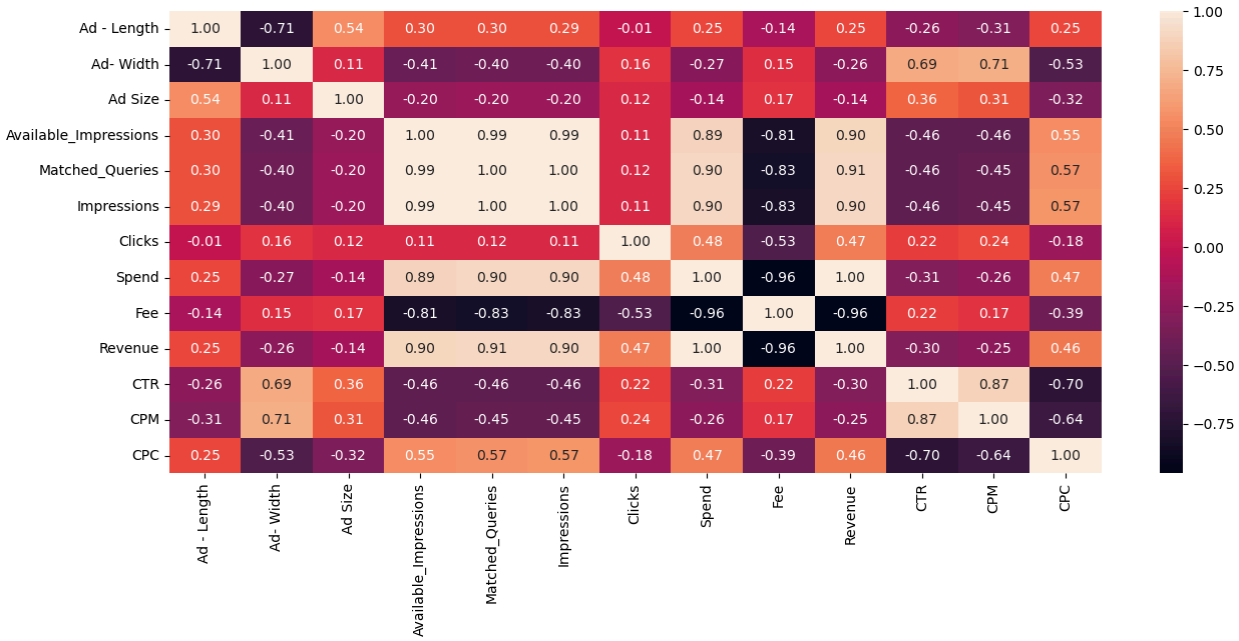

```
Format
Video      11552
Display    11514
Name: count, dtype: int64
```

Observations -

* The majority of advertisements are displayed in video format, with 11,552 instances.
* There is a balanced distribution between the video format and the display format, with a ratio of 11,552 to 11,514 instances respectively.

Bivariate analysis


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



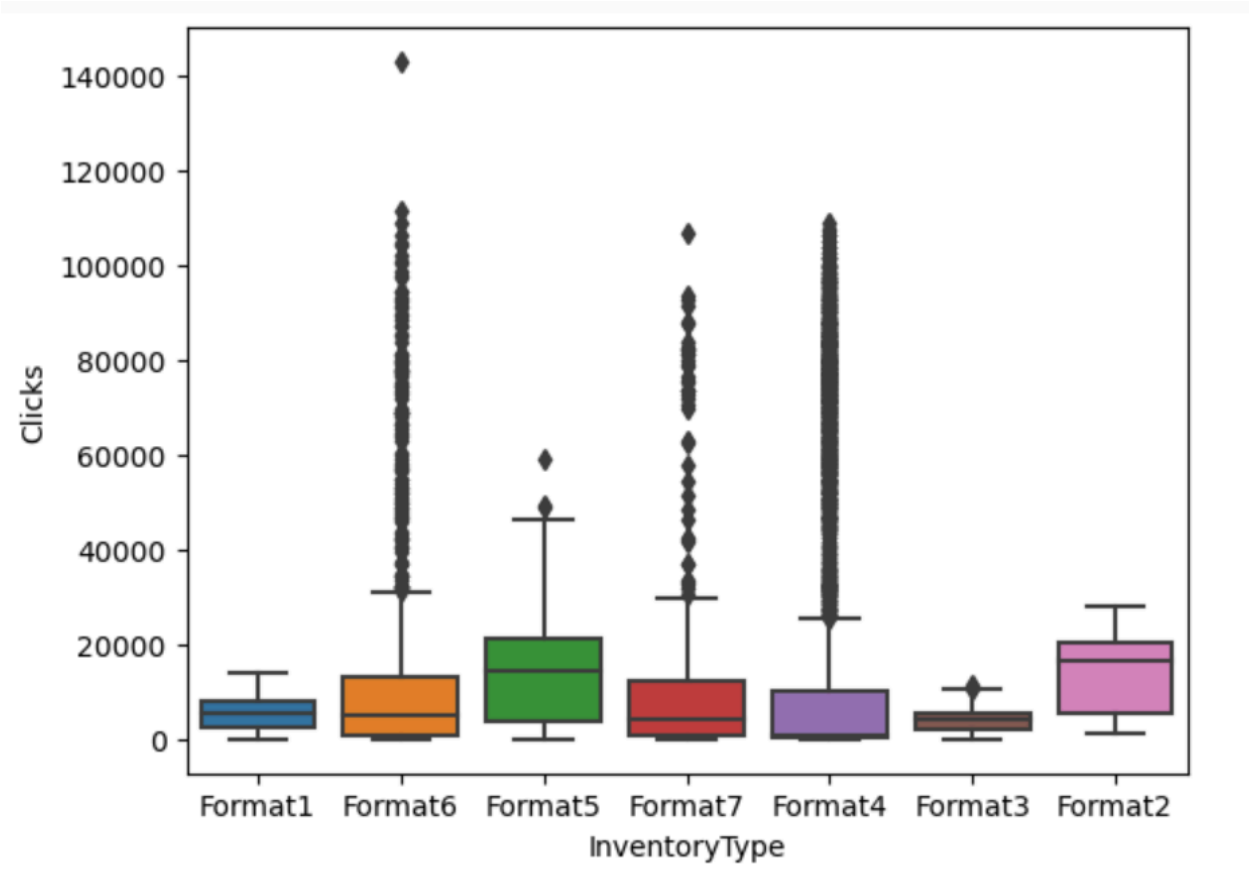
Observations -

- Positive correlation observed between CTR and CPM, implying higher CTR tends to accompany higher CPM.
- Only two ads recorded a CTR above 6.
- Majority of ads fall within CTR 0.0 to 0.5 and CPM 0 to 50.
- Positive correlation noticed between Spend and Revenue, indicating an increase in Spend tends to lead to higher Revenue.
- Two ads had a spend exceeding 25000.
- Most ads generate revenue below 20000.
- Positive correlation found between Impressions and Revenue, suggesting higher Impressions tend to result in increased Revenue.
- Only one ad had revenue exceeding 20000.
- Majority of ads generate revenue below 20000.
- Positive correlation observed between Available Impressions and Matched Queries, indicating an increase in Available Impressions tends to result in higher Matched Queries.
- Majority of ads fall within Matched Queries 0 to 0.8 and Available Impressions 0 to 1.5.
- Positive correlation noted between Available Impressions and Impressions, implying an increase in Available Impressions tends to lead to higher Impressions.

- Majority of ads fall within Impressions 0 to 0.8 and Available Impressions 0 to 1.5.
- Positive correlation observed between Clicks and Spend, indicating an increase in Clicks tends to result in higher Spend.
- Only one ad had a spend above 25000, and one ad had clicks exceeding 120000.

Relationship Between Categorical and Numerical variable

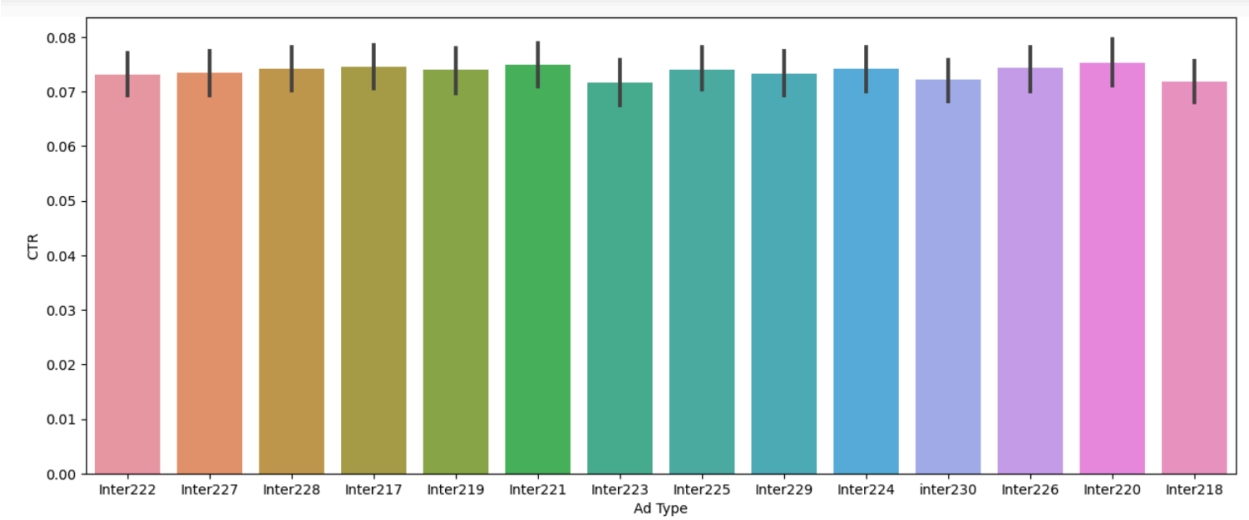
Inventory type and Clicks



Observations -

* Outliers are observed in Format 3, 4, 5, 6, and 7.
* Excluding outliers, the highest number of clicks is observed in Format 5 and Format 2.
* Format 3 has the lowest number of clicks.

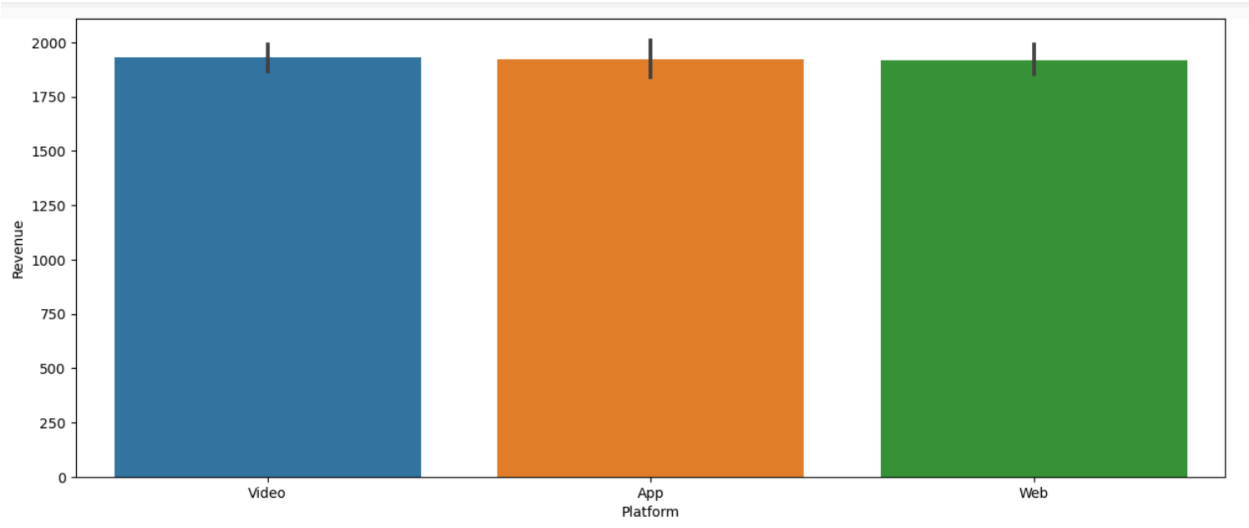
Ad Type and CTR



Observations-

* Most of the ad type have the same median in relation to CTR.
* Inter223 ad type have the lowest CTR.

Platform and Revenue

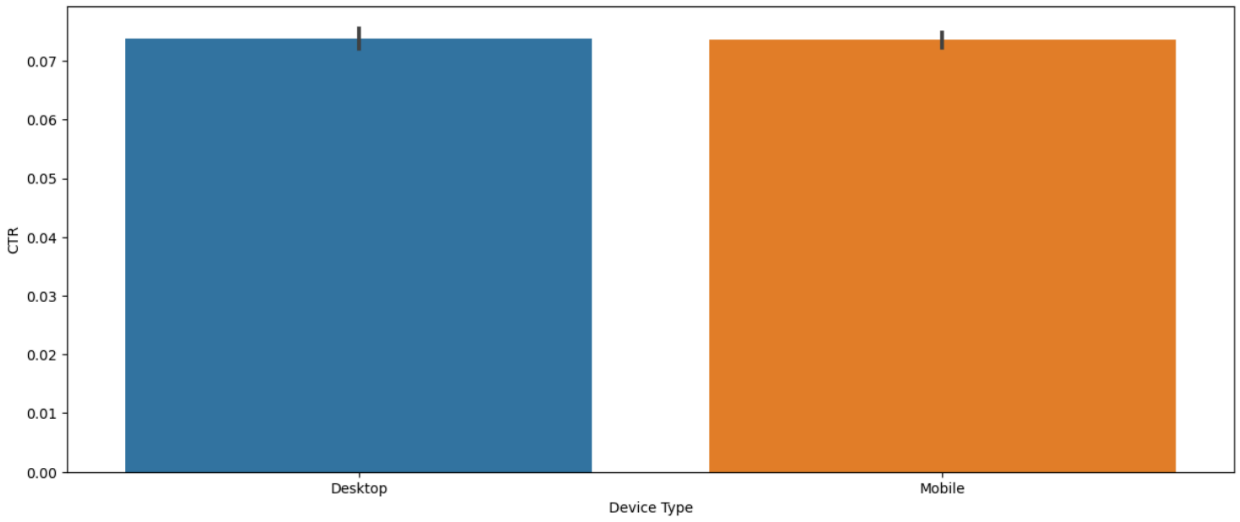


Observations -

* The median revenue across different platforms does not show significant variation.

* There is a slightly lower revenue observed in the Web platform compared to the App and Video platforms.

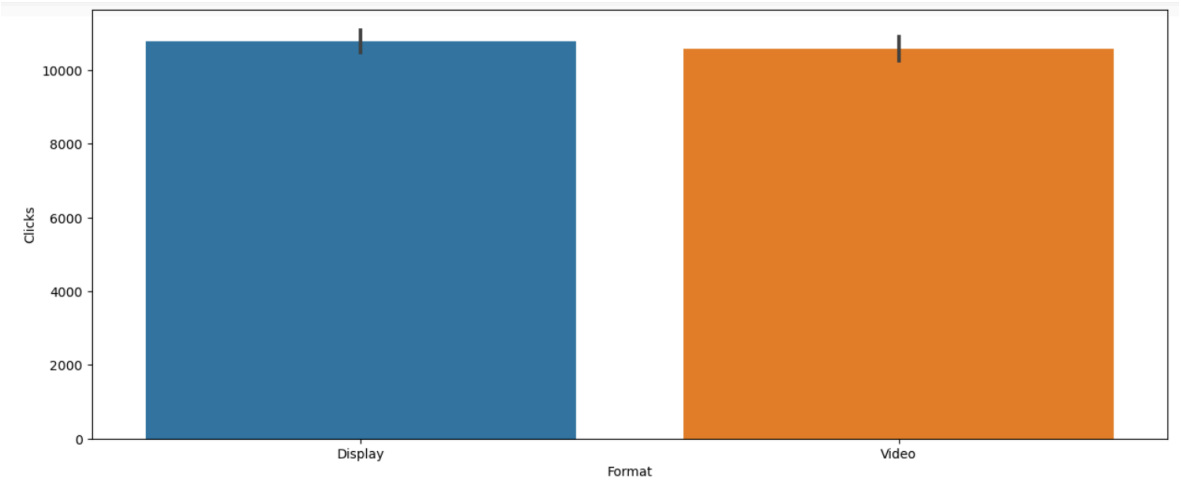
Device Type and CTR



Observations -

* There is not a significant difference in the median of the Click-Through Rate (CTR) across different devices.
* Mobile devices exhibit a slightly lower CTR compared to Desktop devices.

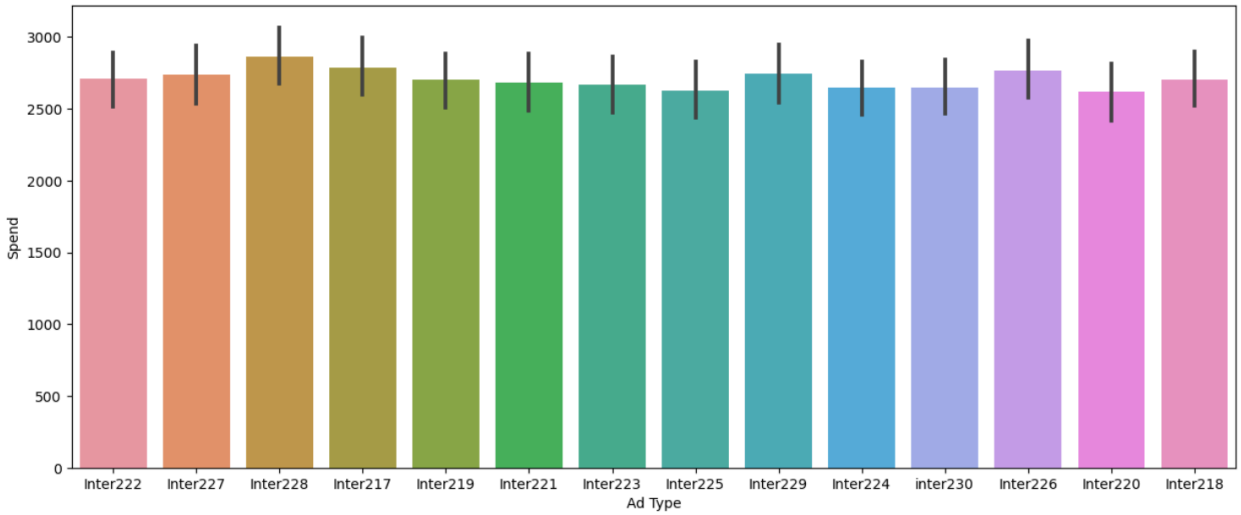
Format and Clicks



Observations -

* There is not a significant difference in the median of the Clicks across different formats.
* Video formats exhibit a slightly lower CTR compared to Display.

Ad Type and Spend



Observations -

* The highest amount spent is on the ad type Inter228, while the lowest is on Inter225.
* There is not a significant difference in the median spending across different ad types.

Part 1: Clustering: Data Preprocessing

Missing value Check and Treatment

```
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
```

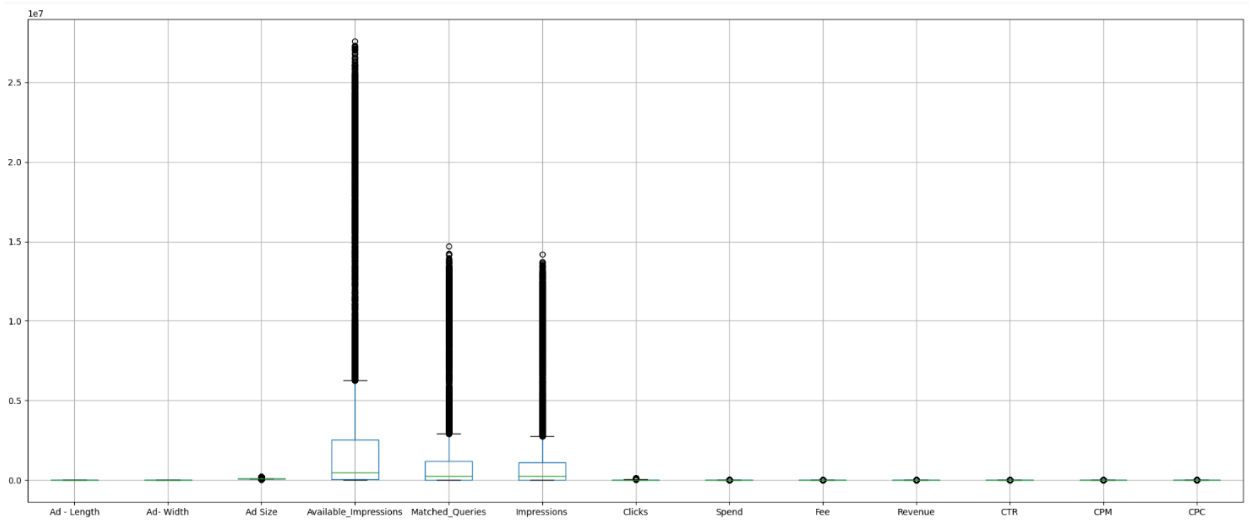
There are 4736 missing values in each of the columns: CTR, CPM, and CPC.

```
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
```

The missing values have been treated as follows:

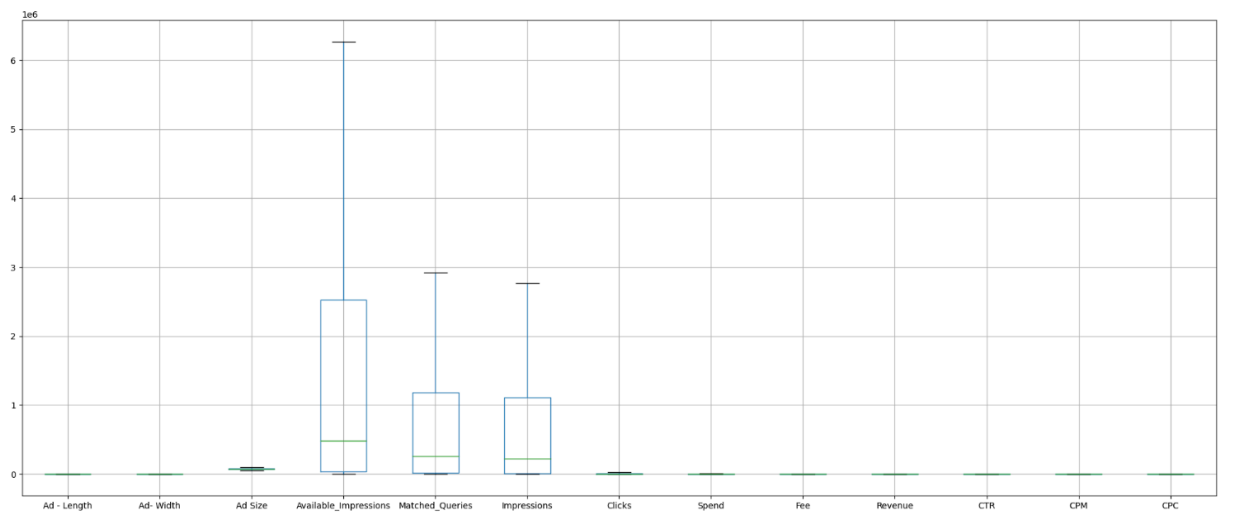
* For CPC: The missing values were imputed using the formula: Spend divided by Clicks.
* For CTR: The missing values were imputed using the formula: (Clicks divided by Impressions) multiplied by 100.
* For CPM: The missing values were imputed using the formula: (Spend divided by Impressions) multiplied by 1000.

Outlier Treatment



As can be seen, outliers are present in most variables except for Ad-Length and Ad-Width.

```
The number of outliers in Ad - Length is 0
The number of outliers in Ad- Width is 0
The number of outliers in Ad Size is 8448
The number of outliers in Available_Impressions is 2378
The number of outliers in Matched_Queries is 3192
The number of outliers in Impressions is 3269
The number of outliers in Clicks is 1691
The number of outliers in Spend is 2081
The number of outliers in Fee is 3517
The number of outliers in Revenue is 2325
The number of outliers in CTR is 275
The number of outliers in CPM is 207
The number of outliers in CPC is 585
```



The outliers have been addressed for all the variables.

Z-Score Scaling

We dropped the following categorical variables for scaling: 'Timestamp', 'InventoryType', 'Ad Type', 'Platform', 'Device Type', and 'Format'.

	Ad - Length	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
0	300.0	250.0	75000.0	1806.0	325.0	323.0	1.0	0.00	0.35	0.0000	0.309598	0.000000	0.00
1	300.0	250.0	75000.0	1780.0	285.0	285.0	1.0	0.00	0.35	0.0000	0.350877	0.000000	0.00
2	300.0	250.0	75000.0	2727.0	356.0	355.0	1.0	0.00	0.35	0.0000	0.281690	0.000000	0.00
3	300.0	250.0	75000.0	2430.0	497.0	495.0	1.0	0.00	0.35	0.0000	0.202020	0.000000	0.00
4	300.0	250.0	75000.0	1218.0	242.0	242.0	1.0	0.00	0.35	0.0000	0.413223	0.000000	0.00
...	...	...	...	...	...	...	...	...	...	...	...	...	...
23061	720.0	300.0	102000.0	1.0	1.0	1.0	1.0	0.07	0.35	0.0455	33.278766	29.981418	0.07
23062	720.0	300.0	102000.0	3.0	2.0	2.0	1.0	0.04	0.35	0.0260	33.278766	20.000000	0.04
23063	720.0	300.0	102000.0	2.0	1.0	1.0	1.0	0.05	0.35	0.0325	33.278766	29.981418	0.05
23064	120.0	600.0	72000.0	7.0	1.0	1.0	1.0	0.07	0.35	0.0455	33.278766	29.981418	0.07
23065	720.0	300.0	102000.0	2.0	2.0	2.0	1.0	0.09	0.35	0.0585	33.278766	29.981418	0.09

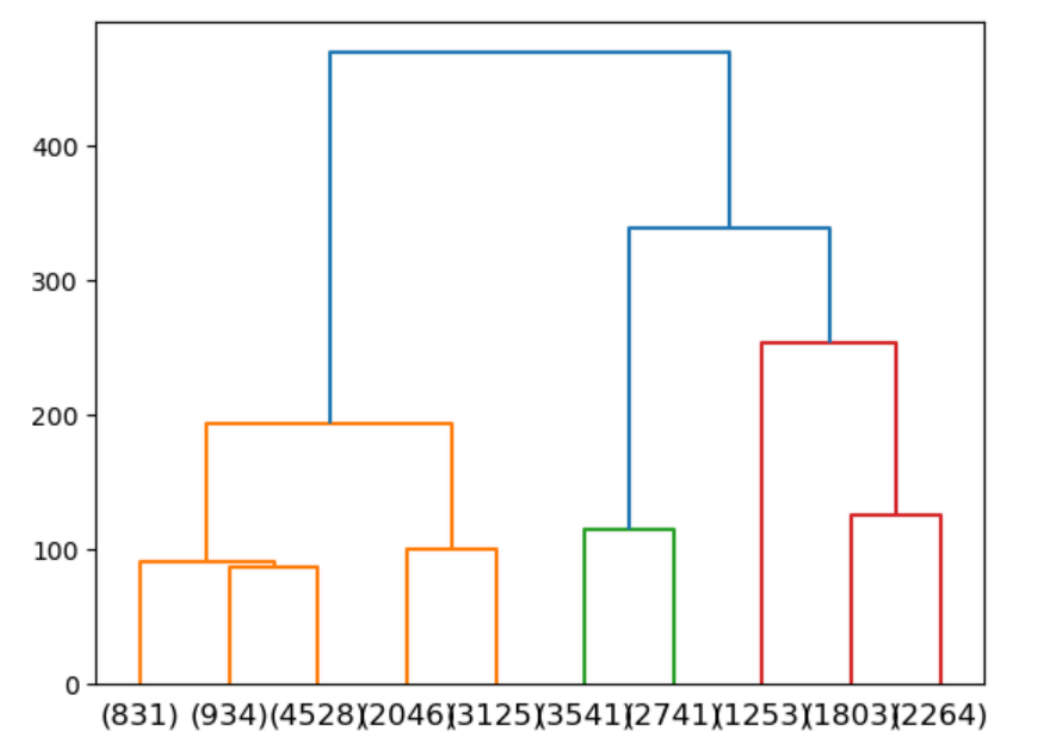
Here are the values we get after scaling -

	Ad- Width	Ad Size	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC
0	-0.432797	-0.102518	-0.755333	-0.778949	-0.768478	-0.867488	-0.893170	0.535724	-0.880093	-0.958836	-1.194498	-1.042561
1	-0.432797	-0.102518	-0.755345	-0.778988	-0.768516	-0.867488	-0.893170	0.535724	-0.880093	-0.953835	-1.194498	-1.042561
2	-0.432797	-0.102518	-0.754900	-0.778919	-0.768445	-0.867488	-0.893170	0.535724	-0.880093	-0.962218	-1.194498	-1.042561
3	-0.432797	-0.102518	-0.755040	-0.778781	-0.768302	-0.867488	-0.893170	0.535724	-0.880093	-0.971871	-1.194498	-1.042561
4	-0.432797	-0.102518	-0.755610	-0.779030	-0.768560	-0.867488	-0.893170	0.535724	-0.880093	-0.946281	-1.194498	-1.042561
...	...	...	...	...	...	...	...	...	...	...	...	...
23061	-0.186599	1.652896	-0.756182	-0.779265	-0.768806	-0.867488	-0.893141	0.535724	-0.880066	3.035808	3.162718	-0.821435
23062	-0.186599	1.652896	-0.756181	-0.779264	-0.768805	-0.867488	-0.893154	0.535724	-0.880078	3.035808	1.712113	-0.916204
23063	-0.186599	1.652896	-0.756182	-0.779265	-0.768806	-0.867488	-0.893150	0.535724	-0.880074	3.035808	3.162718	-0.884614
23064	1.290590	-0.297564	-0.756179	-0.779265	-0.768806	-0.867488	-0.893141	0.535724	-0.880066	3.035808	3.162718	-0.821435
23065	-0.186599	1.652896	-0.756182	-0.779264	-0.768805	-0.867488	-0.893133	0.535724	-0.880058	3.035808	3.162718	-0.758256

Z-score scaling, or standardization, can speed up the algorithm by reducing numerical instability, and improving regularization. It ensures consistent feature scales, leading to quicker optimization. Overall, z-score scaling helps algorithms converge faster and perform more efficiently.

Part 1: Clustering: Hierarchical Clustering

Construct a dendrogram using Ward linkage and Euclidean distance



A dendrogram is a branching diagram that represents the relationships of similarity among a group of
entities.

Identify the optimum number of Clusters

When interpreting a dendrogram, we can identify clusters by cutting the tree diagram at a certain height or distance. This cutting point determines the number of clusters formed. In this case, cutting the dendrogram at a height of 200 results in five distinct clusters.

The decision to cut the dendrogram at a height of 200 is typically based on visual inspection or statistical methods. By choosing this cutoff point, we ensure that the resulting clusters are sufficiently separated and meaningful. These clusters represent groups of data points that are more similar to each other than to data points in other clusters.

Part 1: Clustering: K-means Clustering

Forming 2 Clusters with K=2

We have successfully created two clusters using the k_means.fit function, as demonstrated earlier.

Cluster Output for all the observations

Once the clusters are formed, we can examine the labels using the k_means.labels_ function. Here is a representation of the labels:

```
array([1, 1, 1, ..., 1, 1, 1])
```

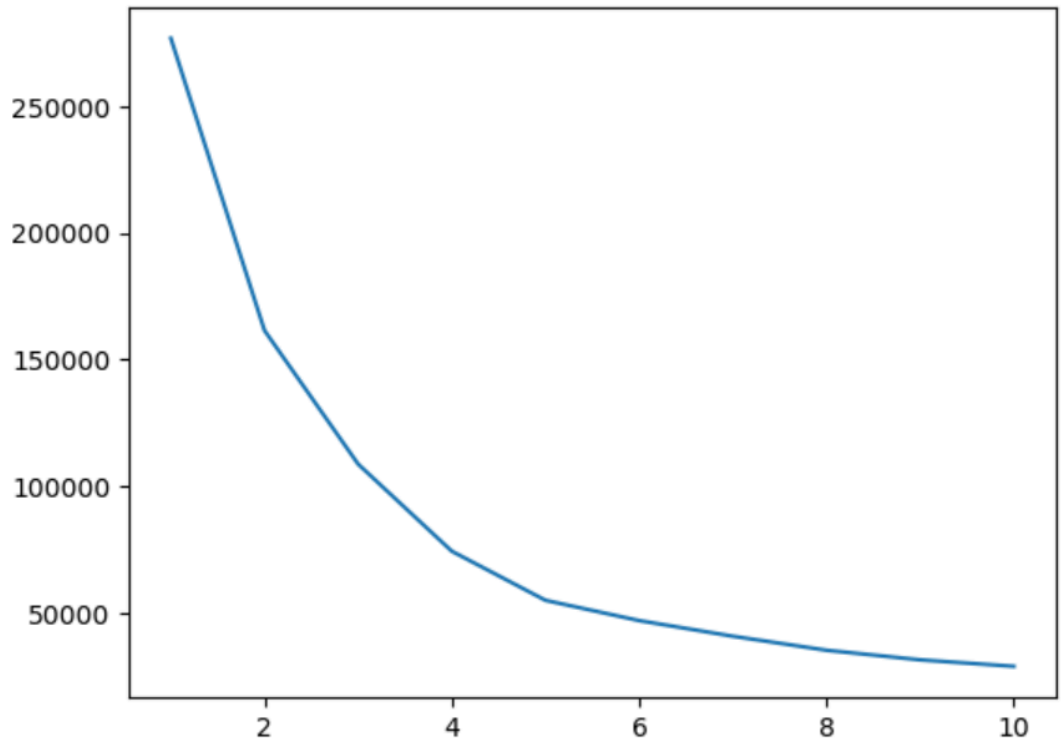
Within Cluster Sum of Squares

Let's calculate the within sum of squares using the k_means.inertia_ function, which results in 161421.56.

Forming clusters with K = 1,3,4,5,6 and comparing the WSS

By forming clusters with k = 1, 3, 4, 5, and 6 and comparing the within sum of squares (WSS), we obtain the following values: 276792.0, 108643.09, 74262.29, 54880.69, 46875.62. It's evident that as the value of k increases, the WSS decreases.

Calculating WSS for other values of K - Elbow Method



To create an elbow plot, we first calculate the within sum of squares (WSS) for the values of k ranging from 1 to 11. Here are the corresponding WSS values: 276792.0, 161421.56, 108643.09, 74262.29, 54880.69, 46875.62, 40700.97, 35166.96, 31376.51, 28827.12. It's evident that as the value of k increases, the WSS decreases.

Silhouette Analysis

Let us now find the Silhouette Score for the values of K from 2 to 10

```
For n_clusters=2, the silhouette score is 0.41132031730541524
For n_clusters=3, the silhouette score is 0.4205087831535238
For n_clusters=4, the silhouette score is 0.4810514153256946
For n_clusters=5, the silhouette score is 0.48715956183236386
For n_clusters=6, the silhouette score is 0.46872835080621006
For n_clusters=7, the silhouette score is 0.4677281384209657
For n_clusters=8, the silhouette score is 0.4505568016703387
For n_clusters=9, the silhouette score is 0.4377449858378993
For n_clusters=10, the silhouette score is 0.4429996956689956
```

Silhouette score is highest for k = 5, among all values of k considered.

Figure out the appropriate number of clusters

Given that the silhouette score is better for 5 clusters than for 4 clusters, we conclude that the optimum number of clusters is 5.

This indicates that the data points within each cluster are relatively closer to each other compared to points in other clusters.

Appending Clusters to the original dataset

Let's proceed with recalculating the labels for the 5 clusters. We'll assign each data point in our dataset to one of these clusters based on the clustering algorithm. This will allow us to append the cluster labels to our original dataframe for further analysis.

Here is how the data frame looks after inserting the column of cluster labels -

Ad Type	Platform	Device Type	Format	Available_Impressions	Matched_Queries	Impressions	Clicks	Spend	Fee	Revenue	CTR	CPM	CPC	Clus_kmeans5
Inter222	Video	Desktop	Display	1806.0	325.0	323.0	1.0	0.0	0.35	0.0	0.309598	0.0	0.0	2
Inter227	App	Mobile	Video	1780.0	285.0	285.0	1.0	0.0	0.35	0.0	0.350877	0.0	0.0	2
Inter222	Video	Desktop	Display	2727.0	356.0	355.0	1.0	0.0	0.35	0.0	0.281690	0.0	0.0	2
Inter228	Video	Mobile	Video	2430.0	497.0	495.0	1.0	0.0	0.35	0.0	0.202020	0.0	0.0	2
Inter217	Web	Desktop	Video	1218.0	242.0	242.0	1.0	0.0	0.35	0.0	0.413223	0.0	0.0	2

Cluster Profiling

Here is the count of rows of different clusters in the dataset:

```
Clus_kmeans5
0    5013
1    4072
2    6134
3    1524
4    6323
Name: count, dtype: int64
```

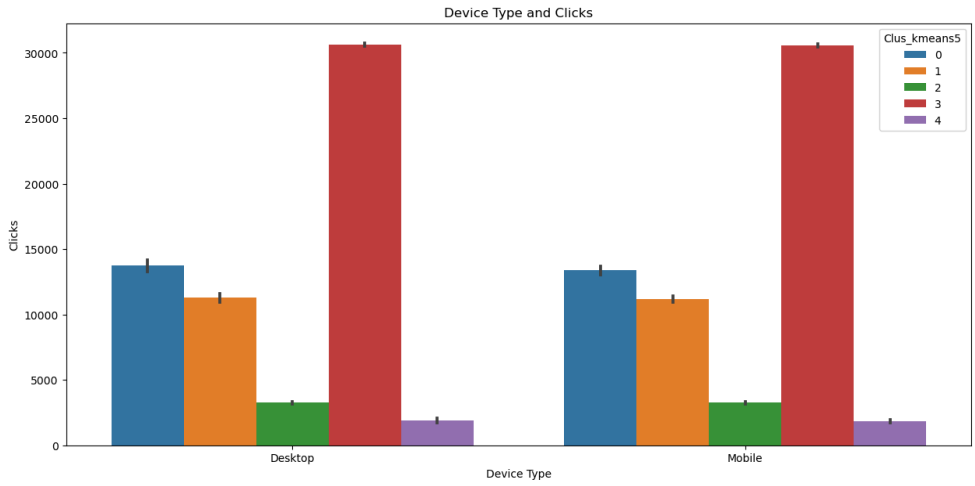
By performing a group by operation and calculating the mean of the Clus_kmeans5 column, we obtained the following results:

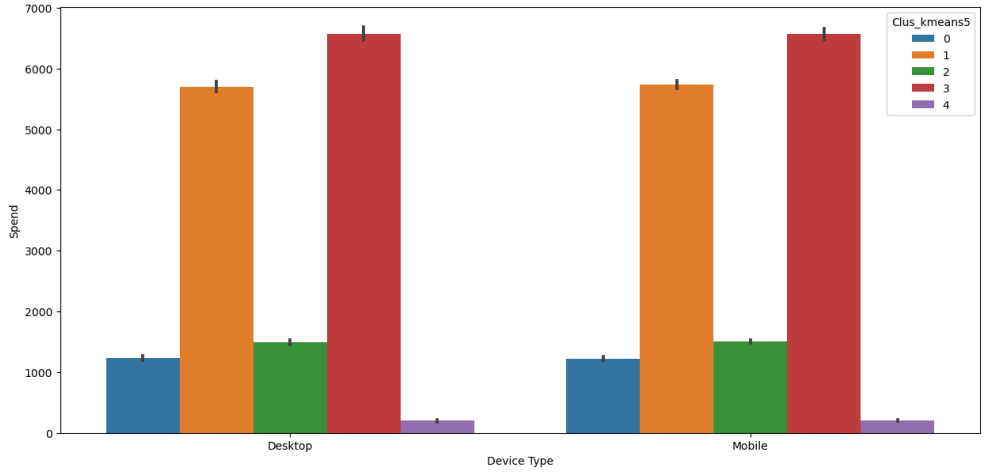
Clus_kmeans5	0	1	2	3	4
Ad - Length	647.576302	4.649528e+02	4.245517e+02	140.401575	146.514629
Ad- Width	314.083383	1.996709e+02	1.425905e+02	574.081365	578.333070
Ad Size	99887.492519	7.299737e+04	6.349935e+04	73590.551181	73807.211767
Available_Impressions	283349.118492	5.686035e+06	1.806281e+06	808339.110236	29620.666930
Matched_Queries	150450.231797	2.802398e+06	8.630091e+05	568685.078740	18377.310296
Impressions	129252.545183	2.667921e+06	8.250218e+05	479669.387139	12556.560177
Clicks	13503.117819	1.121794e+04	3.248226e+03	30577.718176	1851.916021
Spend	1226.823000	5.728236e+03	1.497807e+03	6573.668780	202.606424
Fee	0.349521	3.133694e-01	3.492990e-01	0.305335	0.349987
Revenue	799.050362	3.871128e+03	9.758217e+02	4490.869272	131.733063
CTR	12.248938	2.174335e-01	4.030643e-01	13.757394	16.439732
CPM	10.959629	1.573890e+00	1.786708e+00	15.416594	14.831498
CPC	0.100835	7.482696e-01	5.296129e-01	0.112095	0.101323
freq	5013.000000	4.072000e+03	6.134000e+03	1524.000000	6323.000000

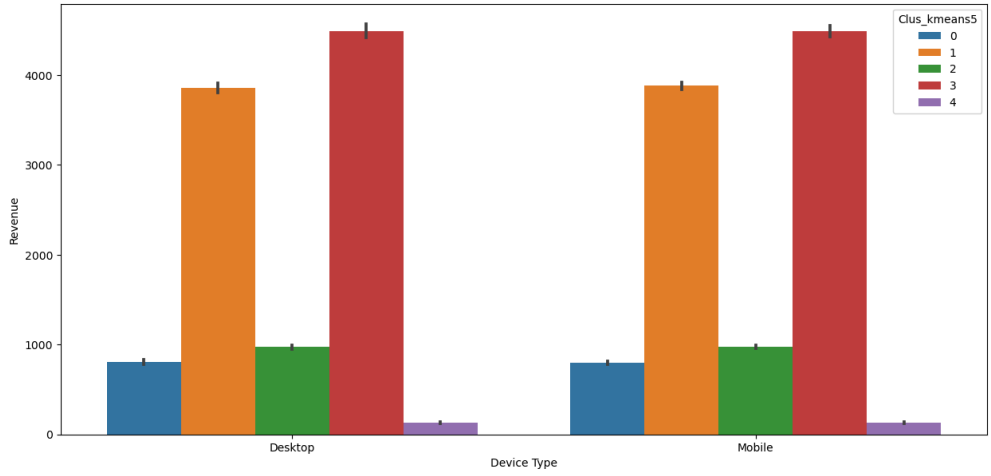
Part 1: Clustering: Actionable Insights & Recommendations

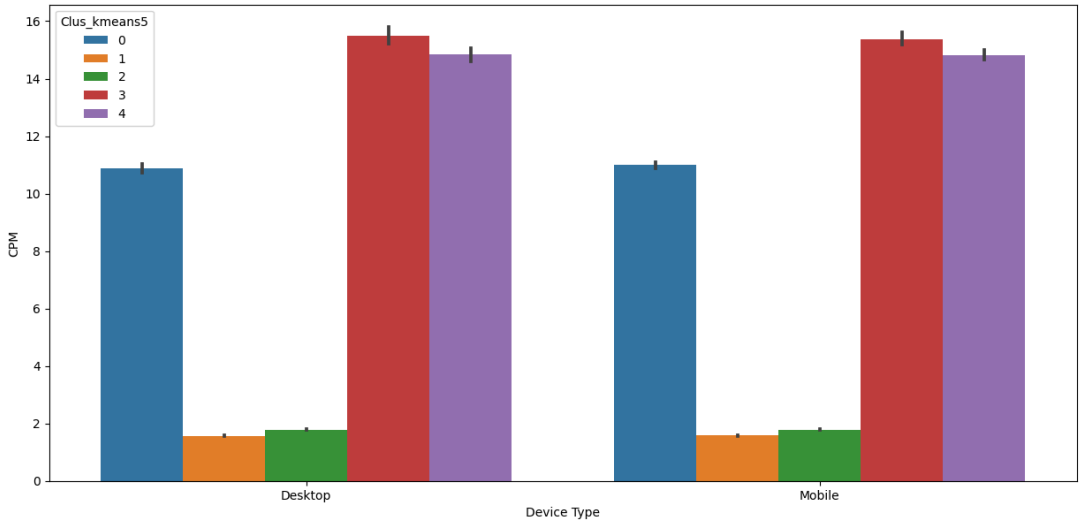
Cluster Visualization Analysis

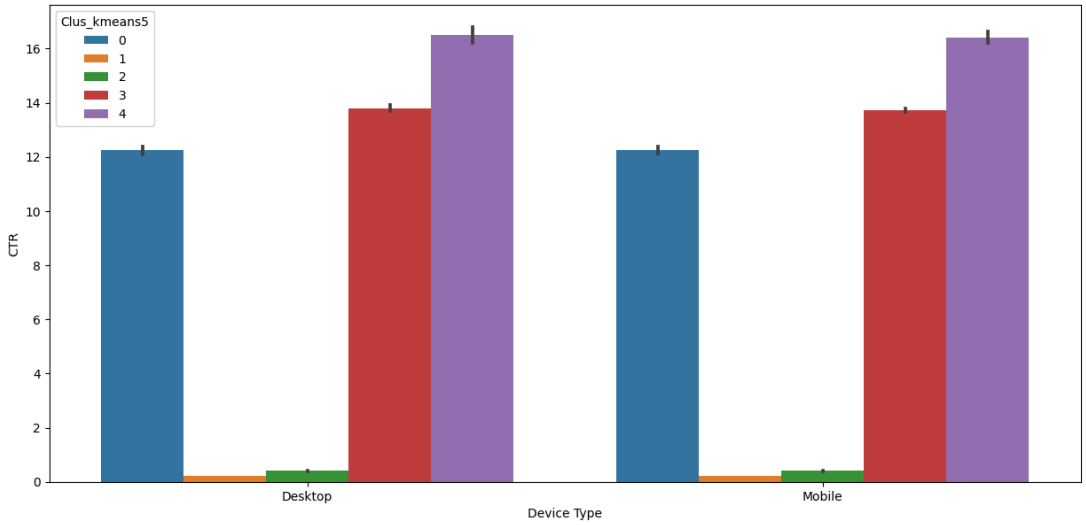
In this section, we will visualize the clusters obtained from the analysis to gain deeper insights into the distribution and characteristics of each cluster.











Insights -

* Cluster 3 has the highest click count on both platforms, while Cluster 4 has the lowest.
* The top spend is observed in Cluster 1 and 2, with the lowest spend in Cluster 4.
* CPM is lowest in Cluster 1 and highest in Clusters 3 and 4.
* There is minimal variation in performance and other specifics among different device types.
* CTR is highest in Cluster 4 and lowest in Cluster 1.
* CPC is highest in Cluster 1 and lowest in Cluster 4 and 0.

Recommendations -

* **Target High-Performing Clusters:** Focus on targeting audiences within Cluster 3, which shows the highest click count on both platforms. Allocate a significant portion of the marketing budget towards campaigns aimed at engaging users in this cluster.

* **Budget Allocation:** Allocate more budget towards campaigns targeting Clusters 1 and 2, which exhibit the highest spend. However, ensure that a portion of the budget is also reserved for targeting Cluster 4 to improve engagement and drive conversions.

* **Monitor CPC:** Keep a close eye on CPC trends across clusters and adjust bidding strategies accordingly. Since Cluster 1 exhibits the highest CPC, explore opportunities to improve ad relevance and quality score to lower costs.

Summary -

* **Missing Values:** There are 4736 missing values in each of the columns: CTR, CPM, and CPC.

* **Hierarchical Clustering:** We perform hierarchical clustering and cut the dendrogram at a height of 200. This cutoff point ensures that resulting clusters are sufficiently separated and meaningful, representing groups of data points that are more similar to each other than to data points in other clusters.

* **K-Means Clustering:** We also performed k-means clustering and formed clusters with K = 1, 3, 4, 5, and 6. By comparing the within sum of squares (WSS), we observed that as the value of k increases, the WSS decreases.

* **Silhouette Score:** We calculated the Silhouette score for different values of k and found that the highest score was for k = 5, indicating that this is the optimal number of clusters among those considered.

* **Label Assignment:** We calculated the labels for 5 clusters using k-means clustering and appended them to our dataframe as new columns.

* **Cluster Profiling:** After assigning labels, we conducted cluster profiling by analyzing mean values of clusters. This analysis provided insights and recommendations for optimizing digital marketing strategies and budget allocation based on the characteristics of each cluster.

Problem Statement -

Part 2: PCA: Define the problem

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.

Data Dictionary

* Name - Description
* State - State Code
* District - District Code
* Name - Name
* TRU1 - Area Name
* No_HH - No of Household
* TOT_M - Total population Male

* TOT_F - Total population Female
* M_06 - Population in the age group 0-6 Male
* F_06 - Population in the age group 0-6 Female
* M_SC - Scheduled Castes population Male
* F_SC - Scheduled Castes population Female
* M_ST - Scheduled Tribes population Male
* F_ST - Scheduled Tribes population Female
* M_LIT - Literates population Male
* F_LIT - Literates population Female
* M_ILL - Illiterate Male
* F_ILL - Illiterate Female
* TOT_WORK_M - Total Worker Population Male
* TOT_WORK_F - Total Worker Population Female
* MAINWORK_M - Main Working Population Male
* MAINWORK_F - Main Working Population Female
* MAIN_CL_M - Main Cultivator Population Male
* MAIN_CL_F - Main Cultivator Population Female
* MAIN_AL_M - Main Agricultural Labourers Population Male
* MAIN_AL_F - Main Agricultural Labourers Population Female
* MAIN_HH_M - Main Household Industries Population Male
* MAIN_HH_F - Main Household Industries Population Female
* MAIN_OT_M - Main Other Workers Population Male
* MAIN_OT_F - Main Other Workers Population Female
* MARGWORK_M - Marginal Worker Population Male
* MARGWORK_F - Marginal Worker Population Female
* MARG_CL_M - Marginal Cultivator Population Male
* MARG_CL_F - Marginal Cultivator Population Female
* MARG_AL_M - Marginal Agriculture Labourers Population Male
* MARG_AL_F - Marginal Agriculture Labourers Population Female
* MARG_HH_M - Marginal Household Industries Population Male
* MARG_HH_F - Marginal Household Industries Population Female
* MARG_OT_M - Marginal Other Workers Population Male
* MARG_OT_F- Marginal Other Workers Population Female
* MARGWORK_3_6_M - Marginal Worker Population 3-6 Male
* MARGWORK_3_6_F - Marginal Worker Population 3-6 Female
* MARG_CL_3_6_M - Marginal Cultivator Population 3-6 Male
* MARG_CL_3_6_F - Marginal Cultivator Population 3-6 Female
* MARG_AL_3_6_M - Marginal Agriculture Labourers Population 3-6 Male
* MARG_AL_3_6_F - Marginal Agriculture Labourers Population 3-6 Female

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

Data Overview

Structure of the Data:

● Number of Rows: 640

● Number of Columns: 60

● Memory Usage: 305.1+ KB

● Range Index: 0 to 639

● Data Types: Integer, Object

Data Type:

The different datatypes in the dataset are as follows

a). There are 59 columns in the with int64 data type

b). There are 2 columns in the with object data type

Statistical Summary

	count	mean	std	min	25%	50%	75%	max
State Code	640.0	17.114062	9.426486	1.0	9.00	18.0	24.00	35.0
Dist.Code	640.0	320.500000	184.896367	1.0	160.75	320.5	480.25	640.0
No_HH	640.0	51222.871875	48135.405475	350.0	19484.00	35837.0	68892.00	310450.0
TOT_M	640.0	79940.576563	73384.511114	391.0	30228.00	58339.0	107918.50	485417.0
TOT_F	640.0	122372.084375	113600.717282	698.0	46517.75	87724.5	164251.75	750392.0
M_06	640.0	12309.098438	11500.906881	56.0	4733.75	9159.0	16520.25	96223.0
F_06	640.0	11942.300000	11326.294567	56.0	4672.25	8663.0	15902.25	95129.0
M_SC	640.0	13820.946875	14426.373130	0.0	3466.25	9591.5	19429.75	103307.0
F_SC	640.0	20778.392188	21727.887713	0.0	5603.25	13709.0	29180.00	156429.0
M_ST	640.0	6191.807813	9912.668948	0.0	293.75	2333.5	7658.00	96785.0
F_ST	640.0	10155.640625	15875.701488	0.0	429.50	3834.5	12480.25	130119.0
M_LIT	640.0	57967.979688	55910.282466	286.0	21298.00	42693.5	77989.50	403261.0
F_LIT	640.0	66359.565625	75037.860207	371.0	20932.00	43796.5	84799.75	571140.0
M_ILL	640.0	21972.596875	19825.605268	105.0	8590.00	15767.5	29512.50	105961.0

Observations:

* **Gender Disparity in Workforce:** The average count of male workers (MAINWORK_M) is higher than that of female workers (MAINWORK_F), indicating a gender disparity in the workforce.

* **Education Disparity:** The average count of illiterate females (F_ILL) is significantly higher than that of illiterate males (M_ILL), suggesting a disparity in education levels between genders.

* **Non-Working Population:** The average count of non-working females (NON_WORK_F) is higher than that of non-working males (NON_WORK_M), indicating a higher proportion of females not engaged in work.

* **Total Working Population:** Despite the higher count of non-working females, the total average female workers (TOT_WORK_F) outnumber male workers (TOT_WORK_M), suggesting that a larger proportion of females are engaged in work compared to males.

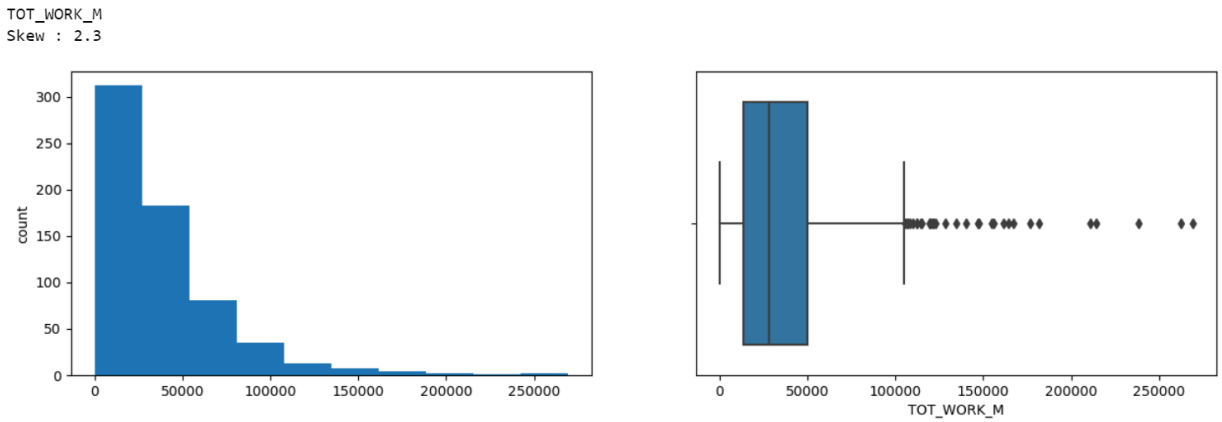
* **Marginal Employment:** In categories such as Marginal Agricultural Labourer and Marginal household industries, females have a higher average count than males in both the 0-3 and 3-6 brackets, indicating a higher participation of females in marginal employment sectors.

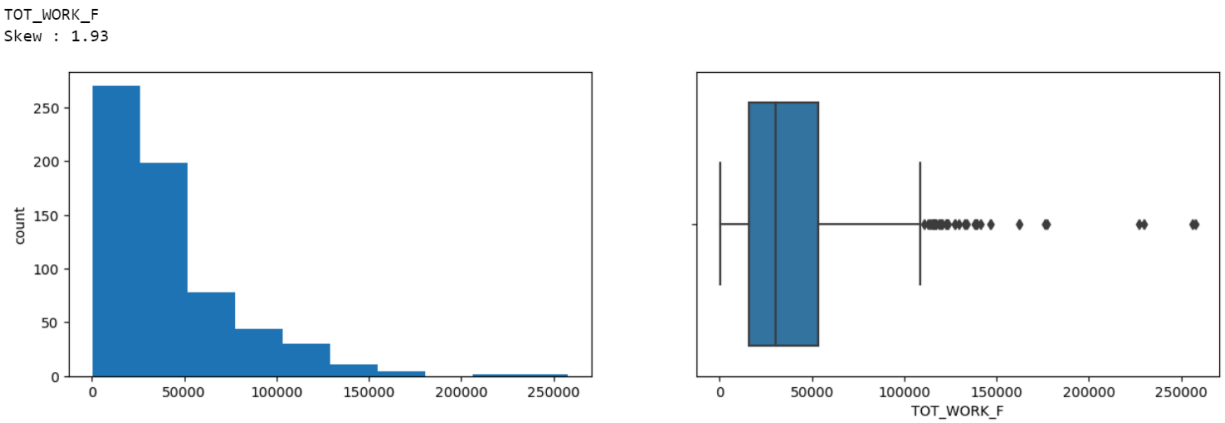
Exploratory Data Analysis

For exploratory Data analysis, We are informed to pick out 5 variables, Here are the ones I am taking as follows -

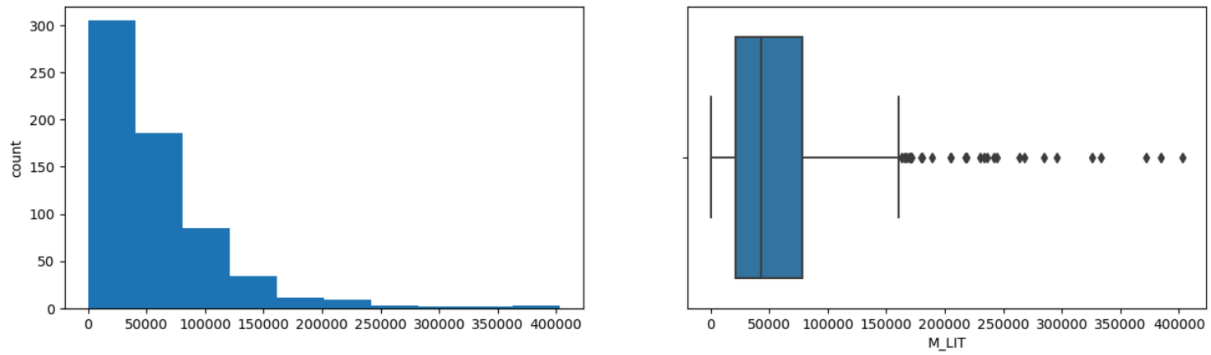
TOT_WORK_M, TOT_WORK_F, M_LIT, F_LIT, No_HH

Univariate Analysis

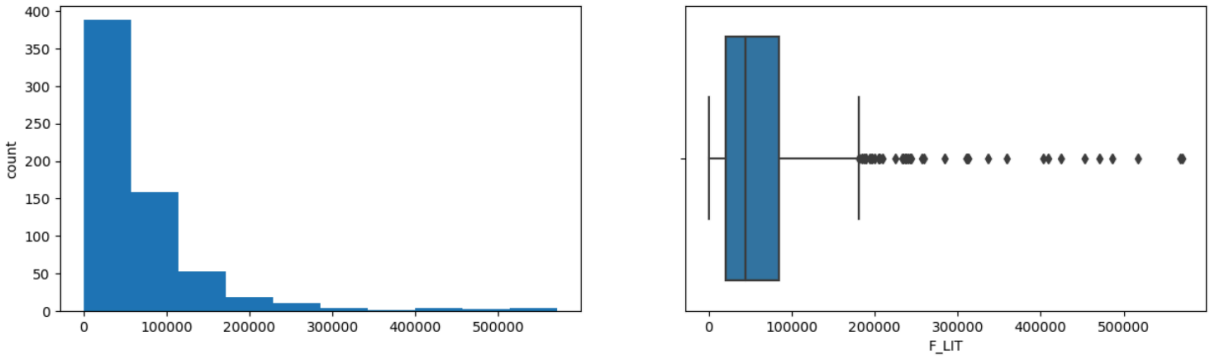




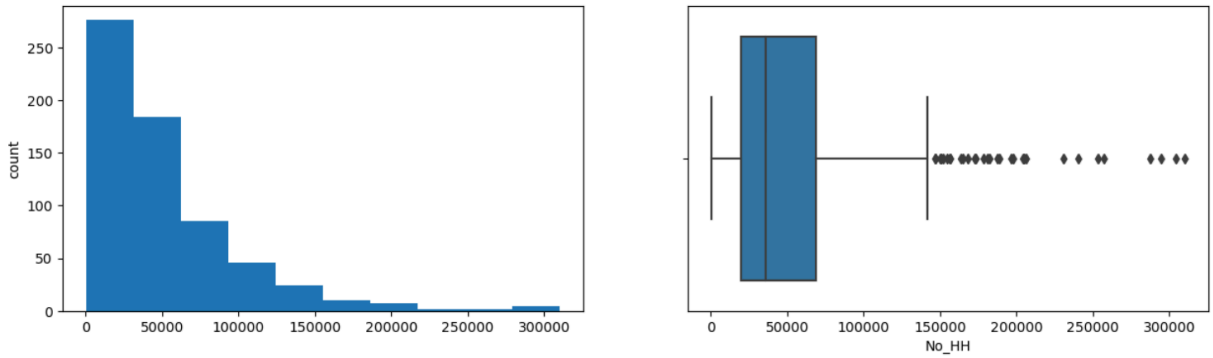
M_LIT
Skew : 2.34



F_LIT
Skew : 3.15



No_HH
Skew : 2.02



Observations -

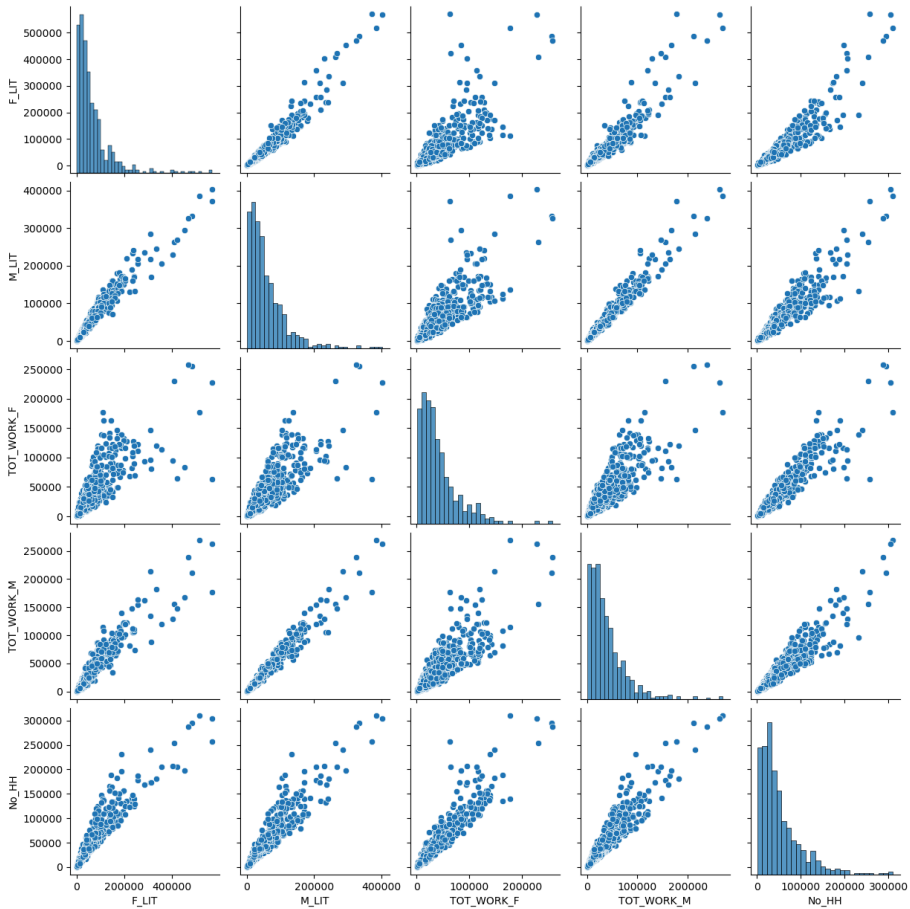
* **Maximum Skewness in Female Literacy:** Female literacy exhibits the highest skewness among all the five variables, indicating that the distribution of female literacy rates may be more positively or negatively skewed compared to the other variables.

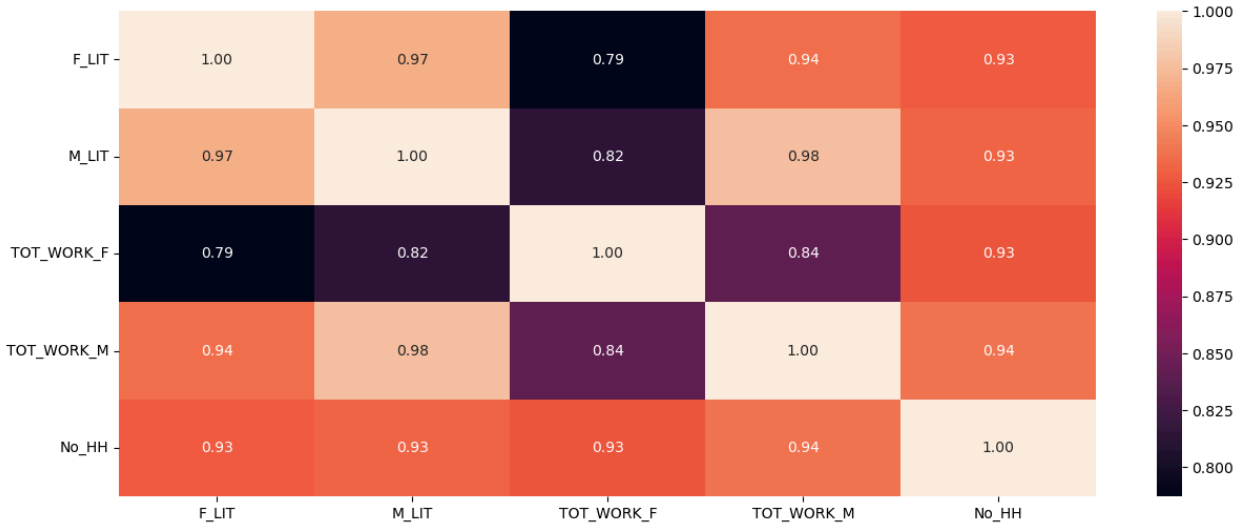
* **Presence of Outliers:** All five variables show the presence of outliers, suggesting that there are data points that significantly deviate from the typical range of values for each variable. These outliers may require further investigation to understand their impact on the analysis.

* **Household Distribution:** Approximately 75% of the total number of households fall below 80000, indicating that a significant proportion of households have relatively low counts.

* **Literacy Levels:** The 75th percentile of female literacy lies below 90000, while for male literacy, it lies below 80000. This suggests that a larger proportion of females are literate compared to males, as indicated by the higher literacy rate threshold for females.

Bivariate Analysis





Observations -

* **High Correlation with Household Count:** There is a notable correlation between the total number of working males and females with the number of households. This suggests that areas with higher household counts tend to have more individuals in the workforce, both male and female.

* **Density of Points:** The density plot indicates a concentration of data points below a threshold of 200000 for both total working males and females. This implies that a significant portion of observations falls within this range, reflecting the distribution of the workforce across different regions or demographic groups.

* **Correlation between Female Literacy and Workforce:** There is a strong positive correlation (0.79) between the count of literate females and the total number of working females. This indicates that areas with higher female literacy rates tend to have more women participating in the workforce.

* **Highest Correlation between Literacy Counts:** The highest correlation is observed between the counts of literate females and literate males. This suggests a strong relationship between male and female literacy levels within the dataset.

* **Distribution of Female Literacy and Workforce:** The majority of observations in the plot of female literacy count versus total working females lie below the threshold of 150000 total working females. This highlights that a significant proportion of regions or

demographic groups have a relatively lower number of working females despite varying levels of female literacy.

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

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

Part 2: PCA: Data Preprocessing

Missing values

```
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
```

No missing values are present in the dataset, as indicated below.

Data Irregularities

Upon reviewing the statistical summary and examining the minimum and maximum values of the variables, no bad data or irregularities were observed. Therefore, no treatment is required for these variables.

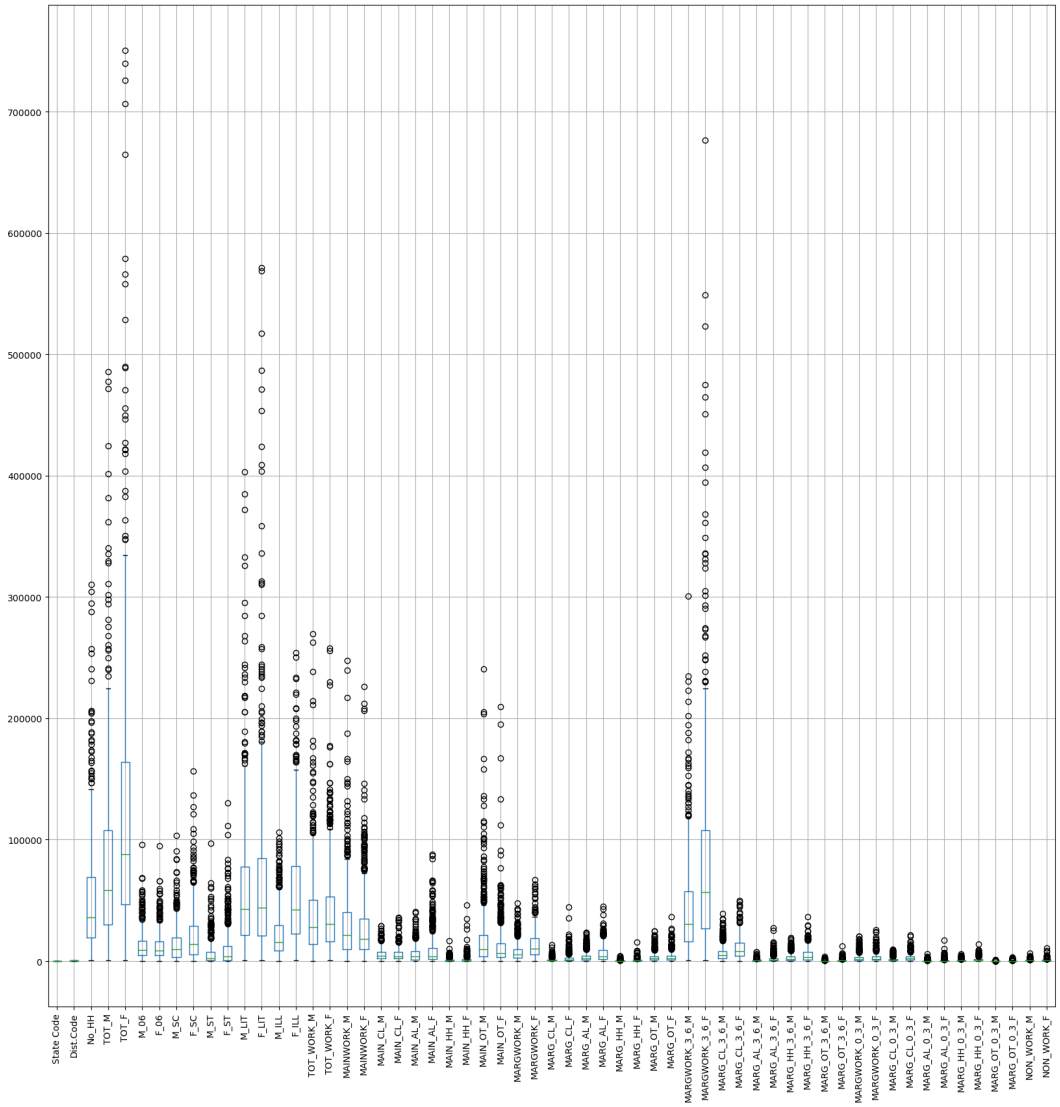
Data Preprocessing before Scaling

	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_AL_0_3_M	MARG_AL_0_3_F	M
0	1	1	7707	23388	29796	5862	6196	3	0	1999	...	1150	749	180	237	
1	1	2	6218	19585	23102	4482	3733	7	6	427	...	525	715	123	229	
2	1	3	4452	6546	10964	1082	1018	3	6	5806	...	114	188	44	89	
3	1	4	1320	2784	4206	563	677	0	0	2666	...	194	247	61	128	
4	1	5	11654	20591	29981	5157	4587	20	33	7670	...	874	1928	465	1043	

5 rows × 59 columns

Prior to scaling, it's necessary to drop the categorical variables from the dataset because scaling operations can only be applied to numerical columns. So, we are dropping 'Area Name' and 'State'.

Visualization (Before Scaling)



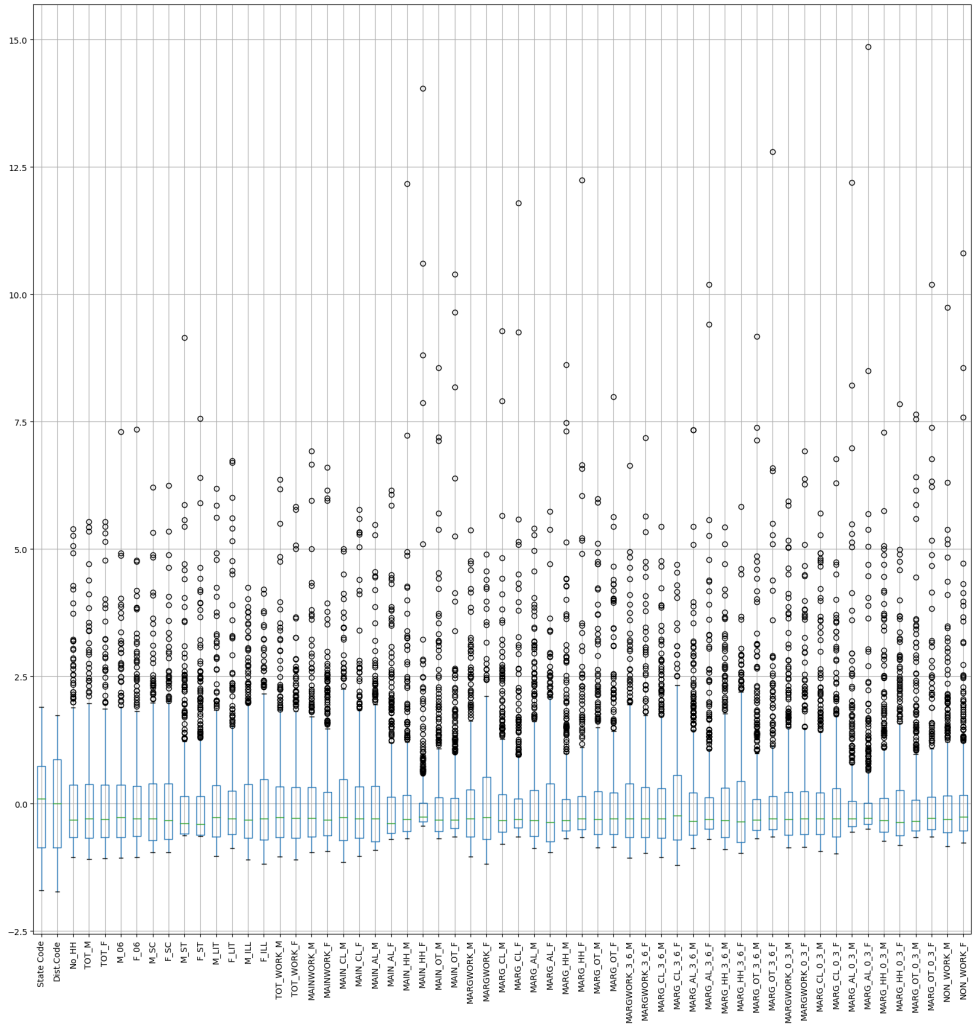
Scaling the data

This is how the data looks after z-score scaling -

	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_AL
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

Visualization (After Scaling)



Observations -

After scaling, the skewness remains the same among all columns but now the data and outliers are distributed among the same magnitude.

Part 2; PCA: PCA

Create the covariance matrix

```
Covariance Matrix
 [[1.00156495 0.99457535 0.38502614 ... 0.03409773 0.12572474 0.23208471]
 [0.99457535 1.00156495 0.37756089 ... 0.03334295 0.11226784 0.21313518]
 [0.38502614 0.37756089 1.00156495 ... 0.53769433 0.76357722 0.73684378]
 ...
 [0.03409773 0.03334295 0.53769433 ... 1.00156495 0.61052325 0.52191235]
 [0.12572474 0.11226784 0.76357722 ... 0.61052325 1.00156495 0.88228018]
 [0.23208471 0.21313518 0.73684378 ... 0.52191235 0.88228018 1.00156495]]
```

The covariance matrix is generated using the np.cov() function, which computes the covariance between pairs of variables in a dataset. This matrix provides valuable information about the relationships and dependencies among the different features or variables in the dataset.

Comparing Correlation and Covariance Matrix -

	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_
State Code	1.000000	0.993021	0.384425	0.166238	0.273080	0.052300	0.063743	0.049352	0.149796	0.240812	...	-0.095886	0.00
Dist.Code	0.993021	1.000000	0.376971	0.164378	0.269653	0.058868	0.071644	0.042254	0.142388	0.234392	...	-0.092615	0.00
No_HH	0.384425	0.376971	1.000000	0.916170	0.970590	0.797559	0.796373	0.775309	0.823847	0.149627	...	0.556941	0.55
TOT_M	0.166238	0.164378	0.916170	1.000000	0.982640	0.950825	0.947792	0.839925	0.826299	0.091421	...	0.698310	0.59
TOT_F	0.273080	0.269653	0.970590	0.982640	1.000000	0.907975	0.906557	0.816959	0.832756	0.123626	...	0.655347	0.59
M_06	0.052300	0.058868	0.797559	0.950825	0.907975	1.000000	0.998151	0.781120	0.747530	0.055274	...	0.760610	0.64
F_06	0.063743	0.071644	0.796373	0.947792	0.906557	0.998151	1.000000	0.773135	0.741686	0.065138	...	0.763614	0.64
M_SC	0.049352	0.042254	0.775309	0.839925	0.816959	0.781120	0.773135	1.000000	0.985071	-0.045666	...	0.673633	0.56
F_SC	0.149796	0.142388	0.823847	0.826299	0.832756	0.747530	0.741686	0.985071	1.000000	-0.014122	...	0.650455	0.58
M_ST	0.240812	0.234392	0.149627	0.091421	0.123626	0.055274	0.065138	-0.045666	-0.014122	1.000000	...	0.122967	0.19
F_ST	0.262816	0.256950	0.165102	0.086180	0.128646	0.043948	0.054662	-0.047825	-0.009190	0.988047	...	0.121411	0.21
M_LIT	0.214163	0.208501	0.931938	0.989312	0.985441	0.912757	0.907641	0.818484	0.814150	0.090541	...	0.652507	0.56
F_LIT	0.293141	0.284111	0.928087	0.931708	0.957012	0.832509	0.829128	0.713939	0.728755	0.100488	...	0.547296	0.48
M_ILL	0.011367	0.020452	0.763041	0.911539	0.858199	0.945409	0.948609	0.800775	0.762560	0.083063	...	0.744658	0.62
F_ILL	0.191555	0.197672	0.862074	0.885361	0.886917	0.863324	0.865289	0.832714	0.847203	0.138031	...	0.708454	0.67
TOT_WORK_M	0.245114	0.237493	0.938199	0.970417	0.968955	0.855773	0.852793	0.824773	0.823689	0.122643	...	0.600872	0.51

'Covariance` indicates the direction of the linear relationship between variables.

`Correlation` on the other hand measures both the strength and direction of the linear relationship between two variables. Correlation is a function of the covariance.

You can obtain the correlation coefficient of two variables by dividing the covariance of these variables by the product of the standard deviations of the same values.

We can state that above three approaches yield the same eigenvectors and eigenvalue pairs:

- Eigen decomposition of the covariance matrix after standardizing the data.
- Eigen decomposition of the correlation matrix.
- Eigen decomposition of the correlation matrix after standardizing the data.

Finally we can say that after scaling - the covariance and the correlation have the same values.

Identify eigenvalues and eigenvectors

```
Eigen Values
%s [ 3.18176335e+01+0.00000000e+00j  8.17627518e+00+0.00000000e+00j
 4.53565320e+00+0.00000000e+00j  3.83736258e+00+0.00000000e+00j
 2.26750940e+00+0.00000000e+00j  1.95686350e+00+0.00000000e+00j
 1.37333087e+00+0.00000000e+00j  8.85956201e-01+0.00000000e+00j
 7.18773122e-01+0.00000000e+00j  6.13100086e-01+0.00000000e+00j
 4.93627186e-01+0.00000000e+00j  4.23485259e-01+0.00000000e+00j
 3.43394965e-01+0.00000000e+00j  2.75530570e-01+0.00000000e+00j
 2.95655943e-01+0.00000000e+00j  1.84706213e-01+0.00000000e+00j
 1.28645538e-01+0.00000000e+00j  1.11362686e-01+0.00000000e+00j
 1.03432923e-01+0.00000000e+00j  9.71908362e-02+0.00000000e+00j
 7.80910464e-02+0.00000000e+00j  5.58740146e-02+0.00000000e+00j
 4.43520192e-02+0.00000000e+00j  3.78063225e-02+0.00000000e+00j
 2.96241834e-02+0.00000000e+00j  2.70149631e-02+0.00000000e+00j
 2.34051410e-02+0.00000000e+00j  1.43387165e-02+0.00000000e+00j
 1.10791547e-02+0.00000000e+00j  9.27324621e-03+0.00000000e+00j
 8.25884163e-03+0.00000000e+00j  7.60154888e-03+0.00000000e+00j
 5.01515304e-03+0.00000000e+00j  4.49240577e-03+0.00000000e+00j
 2.51180436e-03+0.00000000e+00j  1.06091149e-03+0.00000000e+00j
 7.10770360e-04+0.00000000e+00j  5.19236511e-15+0.00000000e+00j
 3.33765707e-15+0.00000000e+00j -3.65215676e-15+0.00000000e+00j
 2.43794710e-15+6.90916960e-17j  2.43794710e-15-6.90916960e-17j
-2.64961285e-15+0.00000000e+00j -2.52266201e-15+0.00000000e+00j
-2.16658620e-15+0.00000000e+00j -2.10879909e-15+0.00000000e+00j
```

```
Eigen Vectors
 %s [[-3.00700521e-02+0.00000000e+00j -1.62782525e-01+0.00000000e+00j
  -2.50129023e-01+0.00000000e+00j ...  1.08726437e-14+0.00000000e+00j
   2.15388551e-15+1.68448756e-15j  2.15388551e-15-1.68448756e-15j]
 [-3.00751392e-02+0.00000000e+00j -1.58821825e-01+0.00000000e+00j
  -2.59359844e-01+0.00000000e+00j ... -1.22761569e-14+0.00000000e+00j
  -3.82931258e-15-1.54369620e-15j -3.82931258e-15+1.54369620e-15j]
 [-1.56432451e-01+0.00000000e+00j -1.28322211e-01+0.00000000e+00j
  -3.34978669e-02+0.00000000e+00j ...  8.94213361e-14+0.00000000e+00j
   1.04292485e-13+3.68078801e-14j  1.04292485e-13-3.68078801e-14j]
 ...
 [-1.31868671e-01+0.00000000e+00j  5.40694563e-02+0.00000000e+00j
  -1.83333910e-03+0.00000000e+00j ...  6.41925391e-02+0.00000000e+00j
  -8.92203212e-02-3.29754361e-05j -8.92203212e-02+3.29754361e-05j]
 [-1.50219557e-01+0.00000000e+00j -5.44095594e-02+0.00000000e+00j
   1.28955424e-01+0.00000000e+00j ...  2.60010031e-02+0.00000000e+00j
  -3.20232917e-02+1.90823576e-02j -3.20232917e-02-1.90823576e-02j]
 [-1.31179136e-01+0.00000000e+00j -6.94741471e-02+0.00000000e+00j
   8.67015734e-02+0.00000000e+00j ...  5.48015576e-02+0.00000000e+00j
  -6.90366616e-03-6.89016310e-03j -6.90366616e-03+6.89016310e-03j]]
```

	State Code	Dist.Code	No_HH	TOT_M	TOT_F	M_06	F_06	
0	-0.030070+0.000000j	-0.162783+0.000000j	-0.250129+0.000000j	-0.120049+0.000000j	-0.145753+0.000000j	-0.090244+0.000000j	0.352205+0.000000j	-0.38629
1	-0.030075+0.000000j	-0.158822+0.000000j	-0.259360+0.000000j	-0.110852+0.000000j	-0.136167+0.000000j	-0.079450+0.000000j	0.351971+0.000000j	-0.39244
2	-0.156432+0.000000j	-0.128322+0.000000j	-0.033498+0.000000j	-0.101335+0.000000j	0.022504+0.000000j	0.000996+0.000000j	0.054283+0.000000j	0.11666
3	-0.167038+0.000000j	-0.080861+0.000000j	0.063630+0.000000j	-0.033299+0.000000j	0.049227+0.000000j	0.074100+0.000000j	-0.069576+0.000000j	-0.00788
4	-0.165702+0.000000j	-0.101111+0.000000j	0.024403+0.000000j	-0.071948+0.000000j	0.027928+0.000000j	0.046350+0.000000j	-0.008746+0.000000j	0.03315
5	-0.161871+0.000000j	-0.012753+0.000000j	0.070453+0.000000j	-0.007703+0.000000j	0.069415+0.000000j	0.152284+0.000000j	-0.088987+0.000000j	-0.08832

The eigenvalues and eigenvectors have been computed at this stage using the np.linalg.eig() function. These eigenvalues and eigenvectors provide crucial insights into the variance and directions of the principal components within the dataset. Additionally, the results have been presented in a dataframe format for further analysis and interpretation.

Principal Component Analysis

To start, we'll define the number of principal components to generate. Then, we'll compute the principal components of the data using the pca.fit_transform method. Finally, we'll determine the percentage of variance explained by each principal component. The variance explained is as follows:

```
array([5.39281923e-01, 1.38580935e-01, 7.68754779e-02, 6.50400438e-02,
       3.84323628e-02, 3.31671780e-02, 2.32767944e-02, 1.50162068e-02,
       1.21825953e-02, 1.03915269e-02, 8.36656248e-03, 7.17771626e-03,
       5.82025365e-03, 5.01111768e-03, 4.67000966e-03, 3.13061378e-03,
       2.18043284e-03, 1.88750315e-03, 1.75310038e-03, 1.64730231e-03,
       1.32357706e-03, 9.47017197e-04, 7.51729139e-04, 6.40785128e-04,
       5.02104803e-04, 4.57880730e-04, 3.96697306e-04, 2.43029093e-04,
       1.87782283e-04, 1.57173665e-04, 1.39980367e-04, 1.28839812e-04,
       8.50025939e-05, 7.61424707e-05, 4.25729552e-05, 1.79815507e-05,
       1.20469553e-05, 1.06354607e-31, 4.88011278e-33, 2.78264111e-33,
       2.78264111e-33, 2.78264111e-33, 2.78264111e-33, 2.78264111e-33,
       2.78264111e-33, 2.78264111e-33, 2.78264111e-33, 2.78264111e-33,
       2.78264111e-33, 2.78264111e-33, 2.78264111e-33, 2.78264111e-33,
       2.78264111e-33, 2.78264111e-33, 2.78264111e-33, 2.78264111e-33,
       2.78264111e-33, 2.78264111e-33, 5.57761502e-34])
```

Identify the optimum number of PCs

Now, we'll check the minimum number of components required to explain more than 90% of the variance. We'll run a loop to sum the explained variances and stop when the cumulative sum exceeds 90%. After the loop, we find that the number of principal components required to explain at least 90% of the variance is 7.

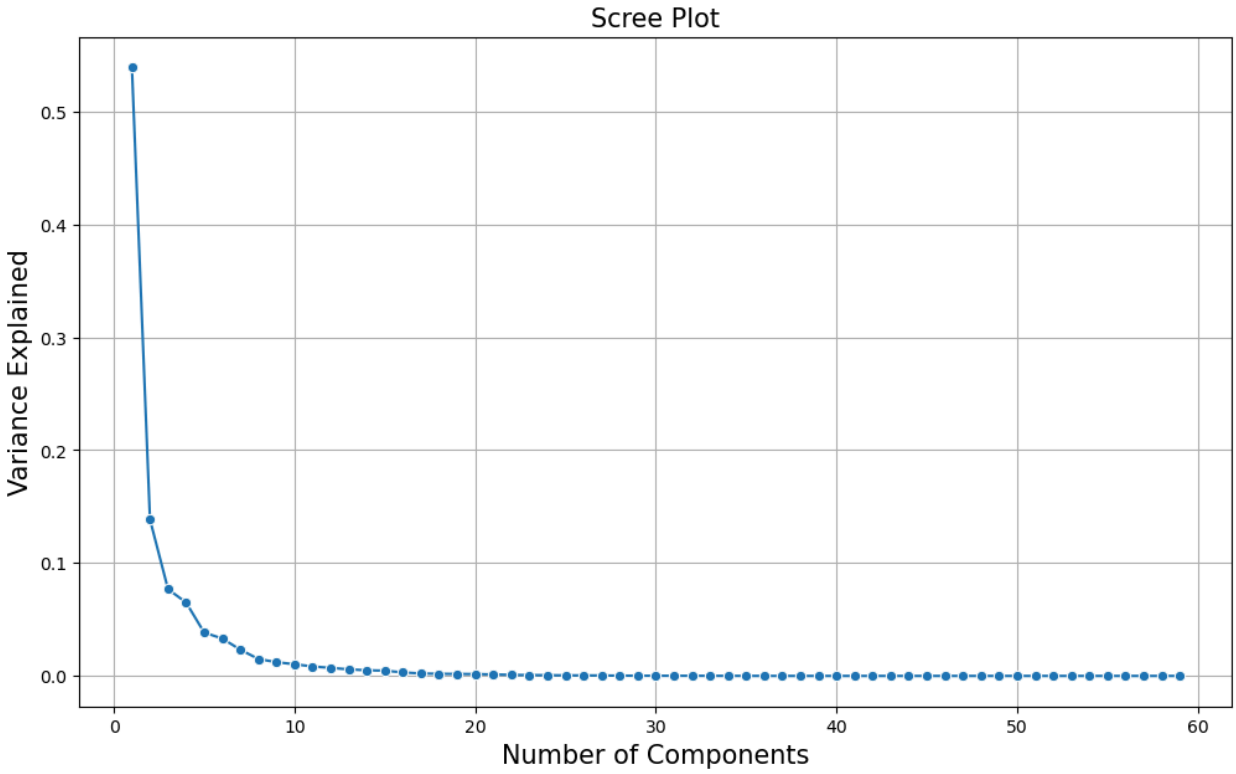
```
Number of PCs that explain at least 90% variance:  7
```

Observations:

* We observe that out of the 59 original features, we reduced the dimensionality to 7 principal components. These components explain more than 90% of the original variance.
* Therefore, there is approximately an 88% reduction in dimensionality with a loss of only 10% in variance.

* Now, let's examine these principal components as linear combinations of the original features.

Scree Plot



Visually we can observe that there is steep drop in variance explained with increase in number of PC's.

Computing Principal components and data reduction

Based on the determination that the optimum number of clusters is 7, we proceed to generate 7 principal components using principal component analysis (PCA). Subsequently, we compute the data reduced to these 7 principal components, resulting

in a dataframe representation of the reduced dataset. This process facilitates dimensionality reduction while preserving the essential information present in the original data.

```
array([[-4.71938093,  0.71750418,  1.63226573, ...,  0.09025696,
        -0.61257314,  0.74128854],
       [-4.87329665,  0.49200093,  1.75212655, ..., -0.26297192,
         0.30521661,  0.67768498],
       [-6.06294775,  0.23375092,  1.33306818, ...,  0.15217157,
        -0.0165205 ,  1.12038163],
       ...,
       [-6.18034109, -1.2162661 , -0.34610853, ...,  0.90705497,
         0.54839523, -1.85319893],
       [-6.10874064, -1.24897987, -0.27949751, ...,  0.7765009 ,
         0.30553866, -1.91221533],
       [-5.78130461, -1.50149132, -0.1861305 , ...,  0.84695498,
         0.24827686, -1.88048749]])
```

	0	1	2	3	4	5	6
0	-4.719381	0.717504	1.632266	-1.524984	0.090257	-0.612573	0.741289
1	-4.873297	0.492001	1.752127	-1.938533	-0.262972	0.305217	0.677685
2	-6.062948	0.233751	1.333068	-0.710272	0.152172	-0.016521	1.120382
3	-6.378387	0.042766	1.404373	-1.187672	0.013924	-0.177346	0.759801
4	-4.581259	1.431602	1.722496	-0.231724	0.579574	0.058364	0.894611
...	...	...	...	...	...	...	...
635	-6.150873	-1.405780	-0.232601	-0.486829	0.748188	0.245564	-1.842009
636	-5.656448	-1.453038	-0.310560	-0.465031	0.723287	0.242883	-1.898197
637	-6.180341	-1.216266	-0.346109	-0.684361	0.907055	0.548395	-1.853199
638	-6.108741	-1.248980	-0.279498	-0.421100	0.776501	0.305539	-1.912215
639	-5.781305	-1.501491	-0.186131	-0.373330	0.846955	0.248277	-1.880487

Finding PC with most variance

```
array([0.53928192, 0.13858094, 0.07687548, 0.06504004, 0.03843236,
       0.03316718, 0.02327679])
```

After applying PCA, we utilize the principal components to identify the component that explains the highest proportion of variance. Through computation, we determine that

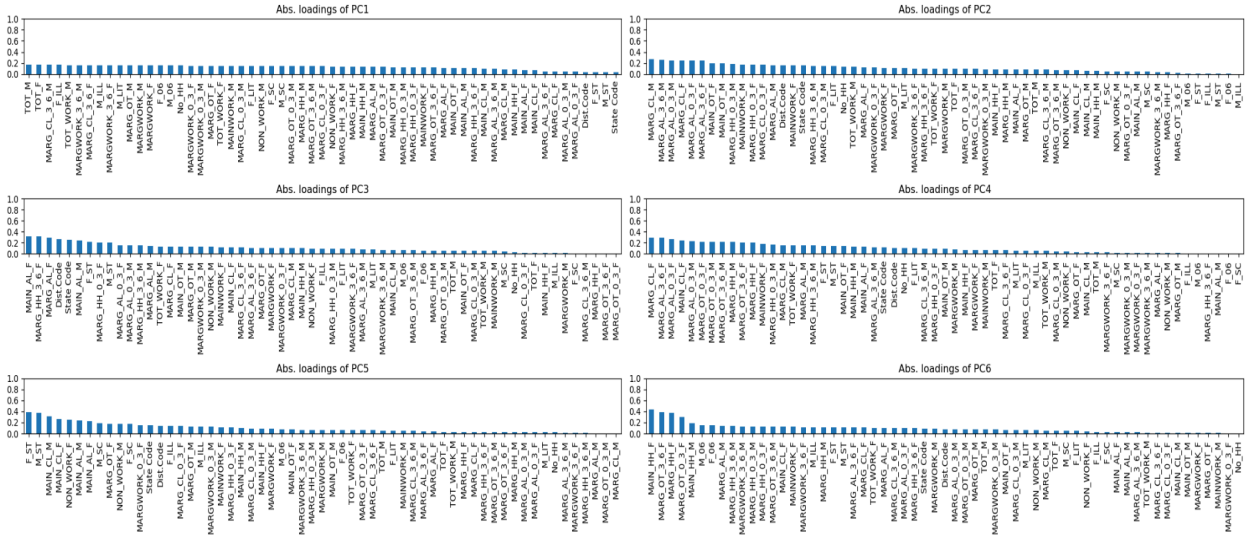
the first principal component (PC1) exhibits the most explained variance ratio, accounting for 53.92% of the total variance captured by the PCA transformation. This implies that PC1 encapsulates a significant portion of the original data's variability, making it a crucial component for dimensionality reduction and data interpretation.

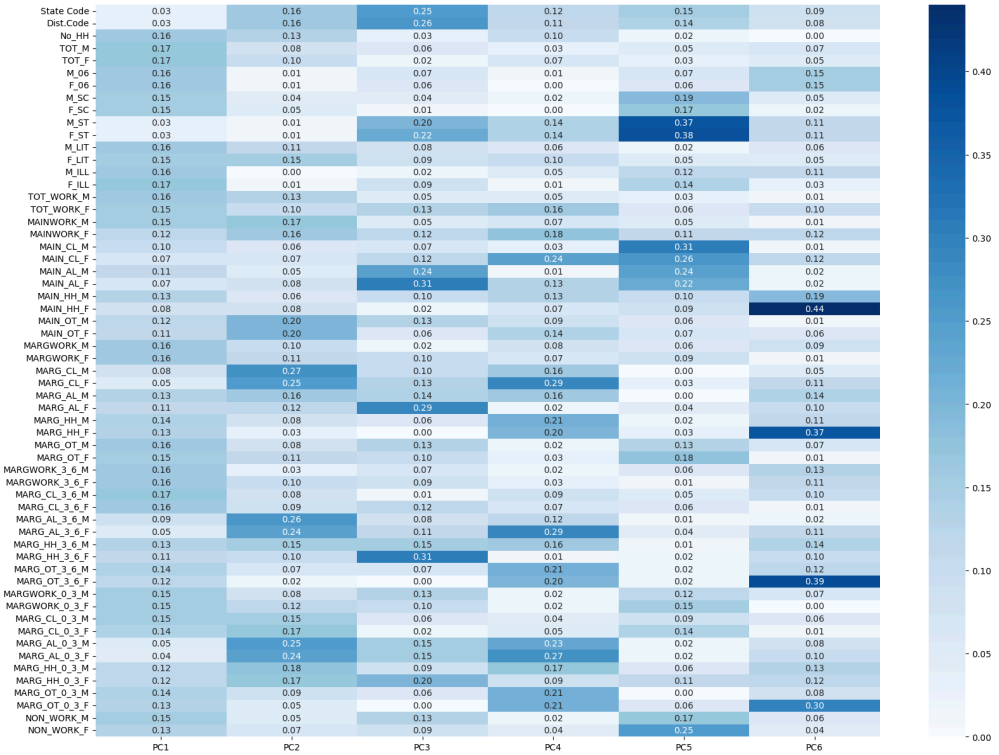
Comparing PCs with Actual Columns

We will now compare the principal components (PCs) obtained from the PCA transformation with the actual columns present in the dataframe to understand how each PC corresponds to the original features.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
State Code	0.03	-0.16	-0.25	0.12	0.15	0.09	-0.35
Dist.Code	0.03	-0.16	-0.26	0.11	0.14	0.08	-0.35
No_HH	0.16	-0.13	-0.03	0.10	-0.02	-0.00	-0.05
TOT_M	0.17	-0.08	0.06	0.03	-0.05	-0.07	0.07
TOT_F	0.17	-0.10	0.02	0.07	-0.03	-0.05	0.01
M_06	0.16	-0.01	0.07	0.01	-0.07	-0.15	0.09
F_06	0.16	-0.01	0.06	0.00	-0.06	-0.15	0.09
M_SC	0.15	-0.04	0.04	-0.02	-0.19	-0.05	0.02
F_SC	0.15	-0.05	-0.01	0.00	-0.17	-0.02	-0.04
M_ST	0.03	0.01	-0.20	0.14	0.37	0.11	0.50
F_ST	0.03	0.01	-0.22	0.14	0.38	0.11	0.47
M_LIT	0.16	-0.11	0.08	0.06	-0.02	-0.06	0.04
F_LIT	0.15	-0.15	0.09	0.10	0.05	-0.05	-0.02
M_ILL	0.16	0.00	0.02	-0.05	-0.12	-0.11	0.16
F_ILL	0.17	-0.01	-0.09	0.01	-0.14	-0.03	0.05
TOT_WORK_M	0.16	-0.13	0.05	0.05	-0.03	-0.01	0.06

We will also visualize the principal components (PCs) alongside the actual columns to gain insights into their relationships and understand how each PC corresponds to the original features.





	PC1	PC2	PC3	PC4	PC5	PC6	PC7
State Code	0.030000	-0.160000	-0.250000	0.120000	0.150000	0.090000	-0.350000
Dist.Code	0.030000	-0.160000	-0.260000	0.110000	0.140000	0.080000	-0.350000
No_HH	0.160000	-0.130000	-0.030000	0.100000	-0.020000	-0.000000	-0.050000
TOT_M	0.170000	-0.080000	0.060000	0.030000	-0.050000	-0.070000	0.070000
TOT_F	0.170000	-0.100000	0.020000	0.070000	-0.030000	-0.050000	0.010000
M_06	0.160000	-0.010000	0.070000	0.010000	-0.070000	-0.150000	0.090000
F_06	0.160000	-0.010000	0.060000	0.000000	-0.060000	-0.150000	0.090000
M_SC	0.150000	-0.040000	0.040000	-0.020000	-0.190000	-0.050000	0.020000
F_SC	0.150000	-0.050000	-0.010000	0.000000	-0.170000	-0.020000	-0.040000
M_ST	0.030000	0.010000	-0.200000	0.140000	0.370000	0.110000	0.500000
F_ST	0.030000	0.010000	-0.220000	0.140000	0.380000	0.110000	0.470000
M_LIT	0.160000	-0.110000	0.080000	0.060000	-0.020000	-0.060000	0.040000
F_LIT	0.150000	-0.150000	0.090000	0.100000	0.050000	-0.050000	-0.020000
M_ILL	0.160000	0.000000	0.020000	-0.050000	-0.120000	-0.110000	0.160000
F_ILL	0.170000	-0.010000	-0.090000	0.010000	-0.140000	-0.030000	0.050000
TOT_WORK_M	0.160000	-0.130000	0.050000	0.050000	-0.030000	-0.010000	0.060000
TOT_WORK_F	0.150000	-0.100000	-0.130000	0.160000	-0.060000	0.100000	-0.030000

Here we are plotting cells in pink where the PC value is less than or equal to -20 and in blue where it is greater than or equal to 30.

Inferences about all the PCs in terms of actual variables -

* **PC2** is characterized by a strong presence of MAIN_OT_M and MAIN_OT_F variables, indicating a significant influence from the main other workers in male and female populations.

* **PC3** exhibits notable contributions from MAIN_AL_M, MAIN_AL_F, MARG_AL_F, and MARG_HH_3_6_F variables, suggesting a combination of main agricultural laborers and marginal household industries in the 3-6 age group for both male and female populations.

* **PC4** is primarily composed of MARG_OT_0_3_M, MARG_OT_3_6_M, and MARG_HH_M variables, indicating a relationship between marginal other workers in the age group 0-3 and 3-6, and marginal household industries for the male population.
* **PC5** showcases significant associations with M_ST, F_ST, MAIN_CL_M, and MAIN_CL_F variables, suggesting correlations between scheduled tribes, main cultivators in male and female populations.
* **PC6** predominantly includes MAIN_HH_F, MARG_HH_F, MARG_OT_3_6_F, and MARG_OT_0_3_F variables, indicating a combination of main household industries and marginal other workers in the age groups 3-6 and 0-3 for the female population.
* **PC7** is mainly characterized by State Code, Dist.Code, M_ST, and F_ST variables, highlighting correlations between state and district codes and scheduled tribes.

Write linear equation for first PC

Each principal component is a linear combination of original features. For example, we can write the equation for PC1 in the following manner:

'State Code'*0.3 + 'Dist.Code'*0.3 + 'No_HH'*0.16 + 'TOT_M'*0.17 + 'TOT_F'*0.17 + 'M_06'*0.16 + 'F_06'*0.16 + 'M_SC'*0.15 + 'F_SC'*0.15 + 'M_ST'*0.3 + 'F_ST'*0.3 + 'M_LIT'*0.16 + 'F_LIT'*0.15 + 'M_ILL'*0.16 + 'F_ILL'*0.17 + 'TOT_WORK_M'*0.16 + 'TOT_WORK_F'*0.15 + 'MAINWORK_M'*0.15 + 'MAINWORK_F'*0.12 + 'MAIN_CL_M'*0.10 + 'MAIN_CL_F'*0.07 + 'MAIN_AL_M'*0.11 + 'MAIN_AL_F'*0.07 + 'MAIN_HH_M'*0.13 + 'MAIN_HH_F'*0.08 + 'MAIN_OT_M'*0.12 + 'MAIN_OT_F'*0.11 + 'MARGWORK_M'*0.16 + 'MARGWORK_F'*0.16 + 'MARG_CL_M'*0.08 + 'MARG_CL_F'*0.05 + 'MARG_AL_M'*0.13 + 'MARG_AL_F'*0.11 + 'MARG_HH_M'*0.14 +

'MARG_HH_F'*0.13 + 'MARG_OT_M'*0.16 + 'MARG_OT_F'*0.15 +

'MARGWORK_3_6_M'*0.16 + 'MARGWORK_3_6_F'*0.16 +

'MARG_CL_3_6_M'*0.17 + 'MARG_CL_3_6_F'*0.16 + 'MARG_AL_3_6_M'*0.09 +

'MARG_AL_3_6_F'*0.05 + 'MARG_HH_3_6_M'*0.13 + 'MARG_HH_3_6_F'*0.11 +

'MARG_OT_3_6_M'*0.14 + 'MARG_OT_3_6_F'*0.12 + 'MARGWORK_0_3_M'0.15 +

'MARGWORK_0_3_F0.15' + 'MARG_CL_0_3_M'*0.15 + 'MARG_CL_0_3_F'*0.14 +

'MARG_AL_0_3_M'*0.05 + 'MARG_AL_0_3_F'*0.04 + 'MARG_HH_0_3_M'*0.12 +

'MARG_HH_0_3_F'*0.12 + 'MARG_OT_0_3_M'*0.14 + 'MARG_OT_0_3_F'*0.13 +

'NON_WORK_M'*0.15 + 'NON_WORK_F'*0.13