Advance Analytics Assignment_2

Dhanush Mathighatta Shobhan Babu (40412492)

6th May 2024

1.0 Introduction

The modern digital marketplace is a dynamic setting where online enterprises constantly adjust to changing consumer demands and competition landscapes. The capacity to precisely forecast corporate performance through analytical insights confers a significant edge. This paper seeks to utilise sophisticated analytics approaches to forecast the performance scores of online firms, using a comprehensive dataset to examine the effectiveness of several prediction models.

#1.1 Objective

The main goal of this analytical task is to utilise Generalised Additive Models (GAM) and Decision Trees to forecast and analyse the performance ratings of online firms. This study aims to surpass basic forecast accuracy and instead aims to uncover the fundamental patterns and correlations inside the data that have a substantial impact on business evaluations. This research will offer practical insights that will assist corporate stakeholders in making well-informed strategic decisions, optimising operational efficiencies, and raising customer happiness.

#1.2 Methodological Framework

In order to direct our analytical process, we utilise the Cross-Industry Standard Process for Data Mining (CRISP-DM) framework. The structured approach has six distinct phases: Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment. By following the CRISP-DM technique, we guarantee a meticulous and organised study, which is essential for obtaining dependable and accurate results (Chapman et al., 2000). This systematic methodology facilitates the alignment of the technical components of our analysis with the overall business goals, guaranteeing that the provided insights are both pertinent and feasible.The analysis is conducted using R, a statistical programming language ideal for data manipulation and analysis

Load the libraries

```
## [1] "/Users/dhanush/Desktop/Bussiness analytics /Sem 2/Adv Analytics
/Assigment 2 "
```

#2.0 Methodlogy

Item A1

In the initial stage of the data analysis, the working directory was set to a specific location on the user's computer to facilitate file management and data access. This directory is where the dataset, `student_merge_platform_business_file_final15.csv`, is stored. The dataset was loaded into the R environment using the `read_csv` function from the `readr` package, known for its efficiency in handling large datasets.

To prepare the data for analysis, several columns irrelevant to the study—such as 'business_id', 'Business_ID_other', 'ZIP Code', 'postal_code', and 'Platform'—were removed. This step streamlined the dataset by focusing only on pertinent variables. Furthermore, preliminary inspections of the dataset were conducted using functions `head(data)`, which displays the first few rows; `summary(data)`, providing statistical summaries of the columns; and `str(data)`, which outlines the structure and types of data columns. These procedures are critical for assessing the quality and readiness of the data for subsequent analytical tasks.

```
## # A tibble: 6 × 16
##   city          state score review_count Gender CEO_sch_cat CEO_grd_yr
field_cat
##   <chr>         <chr> <dbl>        <dbl> <chr>        <dbl>      <dbl>
<dbl>
## 1 Santa Barbara CA      5             7 F               108       1997
13
## 2 Clearwater    FL      5            10 F               108       2017
21
## 3 Bala Cynwyd   PA      4            13 M               108       1986
12
## 4 Plymouth Mee… PA      2.5           8 M               138       1980
75
## 5 Voorhees      NJ      3.5          17 F               116       2014
69
## 6 Tarpon Sprin… FL      2            29 M               170       1988
55
## # ℹ 8 more variables: Rural_metropolitan_Desc <chr>, Tot_Clms_Services
<dbl>,
## #   Brnd_Tot_Clms_Services <dbl>, Gnrc_Tot_Clms_Services <dbl>,
## #   Othr_Tot_Clms_Services <dbl>, LIS_Tot_Clms_Services <dbl>,
## #   Opioid_Tot_Clms_Services <dbl>, Antbtc_Tot_Clms_Services <dbl>
```

```
##      city              state               score       review_count
##  Length:10891       Length:10891       Min.   :1.000   Min.   :  5.00
##  Class :character   Class :character   1st Qu.:2.500   1st Qu.:  6.00
##  Mode  :character   Mode  :character   Median :3.500   Median : 10.00
##                                        Mean   :3.491   Mean   : 15.49
##                                        3rd Qu.:4.500   3rd Qu.: 17.00
##                                        Max.   :5.000   Max.   :413.00
##
##     Gender           CEO_sch_cat      CEO_grd_yr     field_cat
##  Length:10891       Min.   :  0.0   Min.   :1956   Min.   : 0.00
##  Class :character   1st Qu.:108.0   1st Qu.:1997   1st Qu.:18.00
##  Mode  :character   Median :108.0   Median :2004   Median :45.00
##                     Mean   :119.8   Mean   :2004   Mean   :37.32
##                     3rd Qu.:134.0   3rd Qu.:2014   3rd Qu.:58.00
##                     Max.   :226.0   Max.   :2023   Max.   :76.00
##                                     NA's   :8
##  Rural_metropolitan_Desc Tot_Clms_Services Brnd_Tot_Clms_Services
##  Length:10891           Min.   :   11     Min.   :   0.0
##  Class :character       1st Qu.:   98     1st Qu.:  18.0
##  Mode  :character       Median :  337     Median :  79.0
##                         Mean   : 1385     Mean   : 313.1
##                         3rd Qu.: 1454     3rd Qu.: 364.0
##                         Max.   :45742     Max.   :8888.0
##                         NA's   :4919      NA's   :7510
##  Gnrc_Tot_Clms_Services Othr_Tot_Clms_Services LIS_Tot_Clms_Services
##  Min.   :   11          Min.   :   0.00        Min.   :    0.0
##  1st Qu.:   85          1st Qu.:   0.00        1st Qu.:   43.0
##  Median :  268          Median :   0.00        Median :  120.0
##  Mean   : 1163          Mean   :  16.09        Mean   :  637.5
##  3rd Qu.: 1181          3rd Qu.:   0.00        3rd Qu.:  572.0
##  Max.   :38693          Max.   :1218.00        Max.   :40132.0
##  NA's   :4986           NA's   :7521           NA's   :5898
##  Opioid_Tot_Clms_Services Antbtc_Tot_Clms_Services
##  Min.   :   0.00          Min.   :   0.00
##  1st Qu.:   0.00          1st Qu.:  20.00
##  Median :  14.00          Median :  38.00
##  Mean   :  73.53          Mean   :  72.95
##  3rd Qu.:  52.00          3rd Qu.:  96.00
##  Max.   :5317.00          Max.   :1499.00
##  NA's   :6454             NA's   :6787
```

```
## tibble [10,891 × 16] (S3: tbl_df/tbl/data.frame)
##  $ city                    : chr [1:10891] "Santa Barbara" "Clearwater"
"Bala Cynwyd" "Plymouth Meeting" ...
##  $ state                   : chr [1:10891] "CA" "FL" "PA" "PA" ...
##  $ score                   : num [1:10891] 5 5 4 2.5 3.5 2 3 1.5 2.5 1.5
...
##  $ review_count            : num [1:10891] 7 10 13 8 17 29 36 14 43 6 ...
##  $ Gender                  : chr [1:10891] "F" "F" "M" "M" ...
##  $ CEO_sch_cat             : num [1:10891] 108 108 108 138 116 170 116 108
3 201 ...
##  $ CEO_grd_yr              : num [1:10891] 1997 2017 1986 1980 2014 ...
##  $ field_cat               : num [1:10891] 13 21 12 75 69 55 21 58 33 3
...
##  $ Rural_metropolitan_Desc : chr [1:10891] NA "Metropolitan area core:
primary flow within an urbanized area of 50,000 and greater" NA "Metropolitan
area core: primary flow within an urbanized area of 50,000 and greater" ...
##  $ Tot_Clms_Services       : num [1:10891] NA 971 NA 1988 847 ...
##  $ Brnd_Tot_Clms_Services  : num [1:10891] NA 138 NA NA 73 NA 103 NA NA 11
```

```
...
##  $ Gnrc_Tot_Clms_Services  : num [1:10891] NA 813 NA 1674 774 ...
##  $ Othr_Tot_Clms_Services  : num [1:10891] NA 20 NA NA 0 NA 0 NA NA 0 ...
##  $ LIS_Tot_Clms_Services   : num [1:10891] NA 762 NA 65 75 NA 106 NA NA 0
...
##  $ Opioid_Tot_Clms_Services: num [1:10891] NA NA NA NA 50 NA 33 NA NA 0
...
##  $ Antbtc_Tot_Clms_Services: num [1:10891] NA 31 NA 274 NA NA 25 NA NA 0
...
```

#2.0.1 Data Overview

The dataset being analysed consists of sixteen variables, which encompass a wide array of information relevant to business assessments. The data encompasses several types, ranging from category, such as ‘city’ and ‘state’, to numerical, including ‘score’ and ‘review_count’. The total number of observations for most variables is 10,891, while certain variables have a considerable number of missing records. Two notable examples are ‘Rural_metropolitan_Desc’ and ‘Tot_Clms_Services’, both of which have more than 45% missing data. This suggests that data collection in these regions may have been partial or inconsistent.

Every variable provides a unique set of data, with the amount of distinct values varying greatly, ranging from as little as two in ‘Gender’ to as high as 1,368 in ‘Tot_Clms_Services’. This variation highlights the diverse and varied nature of the dataset, indicating a wide range of analysis possibilities.

Prior to conducting a targeted analytical endeavour, certain columns that were considered unrelated to the objectives of the study were eliminated from the dataset. Excluded were identifiers and location codes such as ‘business_id’, ‘Business_ID_other’, ‘ZIP Code’, ‘postal_code’, and ‘Platform’. This phase was essential to optimise the dataset, ensuring that future analyses focus solely on characteristics that directly impact the evaluation of business performance. Preprocessing serves to clarify the dataset and improve the efficiency and relevance of the analysis by removing irrelevant information.

```
##    Index            Variable_Name Variable_Type Sample_n Missing_Count
## 1      1                     city     character    10891             0
## 2      2                    state     character    10891             0
## 3      3                    score       numeric    10891             0
## 4      4             review_count       numeric    10891             0
## 5      5                   Gender     character    10891             0
## 6      6              CEO_sch_cat       numeric    10891             0
## 7      7               CEO_grd_yr       numeric    10883             8
## 8      8                field_cat       numeric    10891             0
## 9      9  Rural_metropolitan_Desc     character     5971          4920
## 10    10        Tot_Clms_Services       numeric     5972          4919
## 11    11   Brnd_Tot_Clms_Services       numeric     3381          7510
## 12    12   Gnrc_Tot_Clms_Services       numeric     5905          4986
## 13    13   Othr_Tot_Clms_Services       numeric     3370          7521
## 14    14    LIS_Tot_Clms_Services       numeric     4993          5898
## 15    15 Opioid_Tot_Clms_Services       numeric     4437          6454
## 16    16 Antbtc_Tot_Clms_Services       numeric     4104          6787
##    Per_of_Missing No_of_distinct_values
## 1           0.000                    484
## 2           0.000                     15
## 3           0.000                      9
## 4           0.000                    164
## 5           0.000                      2
## 6           0.000                    227
## 7           0.001                     61
## 8           0.000                     77
## 9           0.452                     13
## 10          0.452                   1368
## 11          0.690                    620
## 12          0.458                   1288
## 13          0.691                    119
## 14          0.542                    861
## 15          0.593                    320
## 16          0.623                    293
```

Item A2

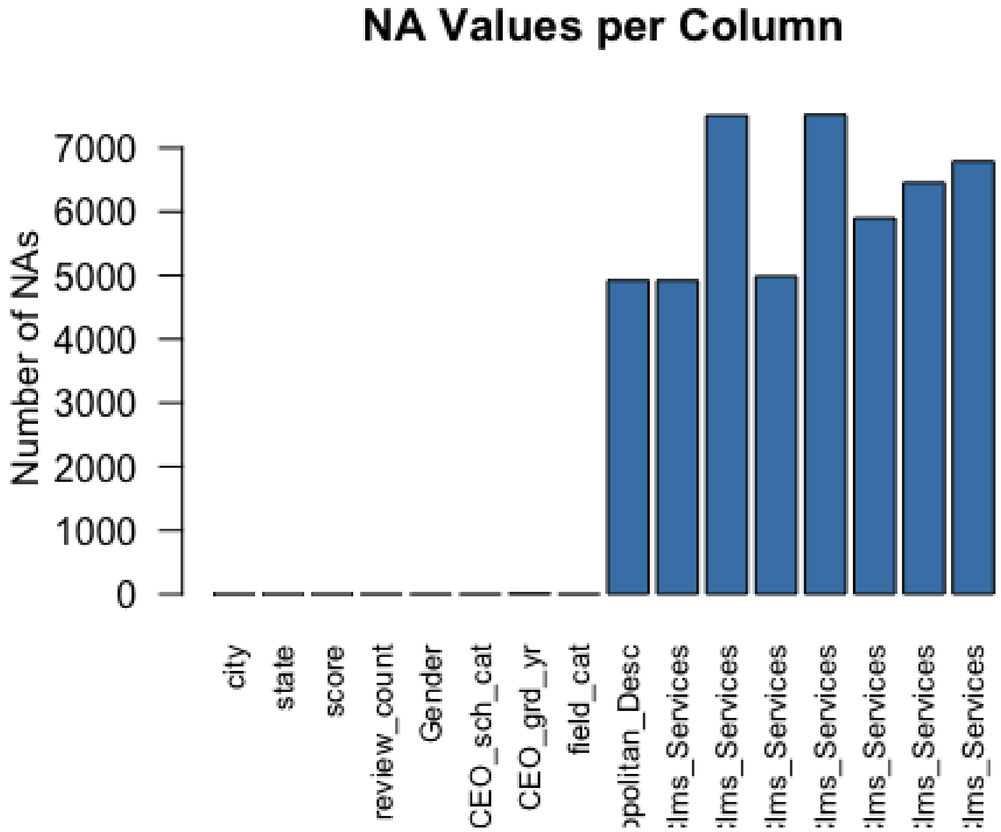
#2.1 EDA

#2.2.1 Overview of missing values

]Missing data in a dataset can have a substantial impact on the quality of statistical analysis, potentially causing biassed conclusions if not properly handled. In order to assure the reliability and accuracy of our findings, it was crucial to initially evaluate and subsequently correct any missing values in our dataset.

Below bar chart shows the distribution of missing values (NAs) across dataset variables. The claims service columns ‘Tot_Clms_Services’, ‘Brnd_Tot_Clms_Services’, ‘Gnrc_Tot_Clms_Services’, ‘Othr_Tot_Clms_Services’, ‘LIS_Tot_Clms_Services’, and ‘Antbtc_Tot_Clms_Services’ have upwards of 4,000 missing entries. The high number of missing variables requires cautious management to maintain the integrity and reliability of subsequent studies. Data imputation or exclusion may be needed to fill these gaps, depending on the analysis and its impact on the results.

```
## [1] 49003
```



Item A3

2.2 Handling missing values

Initial Assessment We commenced our data cleaning process by determining the extent of missingness within each variable. This quantitative assessment was crucial for planning subsequent steps and for selecting suitable imputation methods tailored to the nature of the data missing in each column.

Imputation Strategies Numeric Data For numeric variables, we chose the median for imputation purposes. The decision to use the median over the mean or other statistical measures stems from its robustness to outliers. This attribute makes the median a reliable measure of central tendency that does not distort the original distribution of the data, thereby preserving the dataset's integrity.

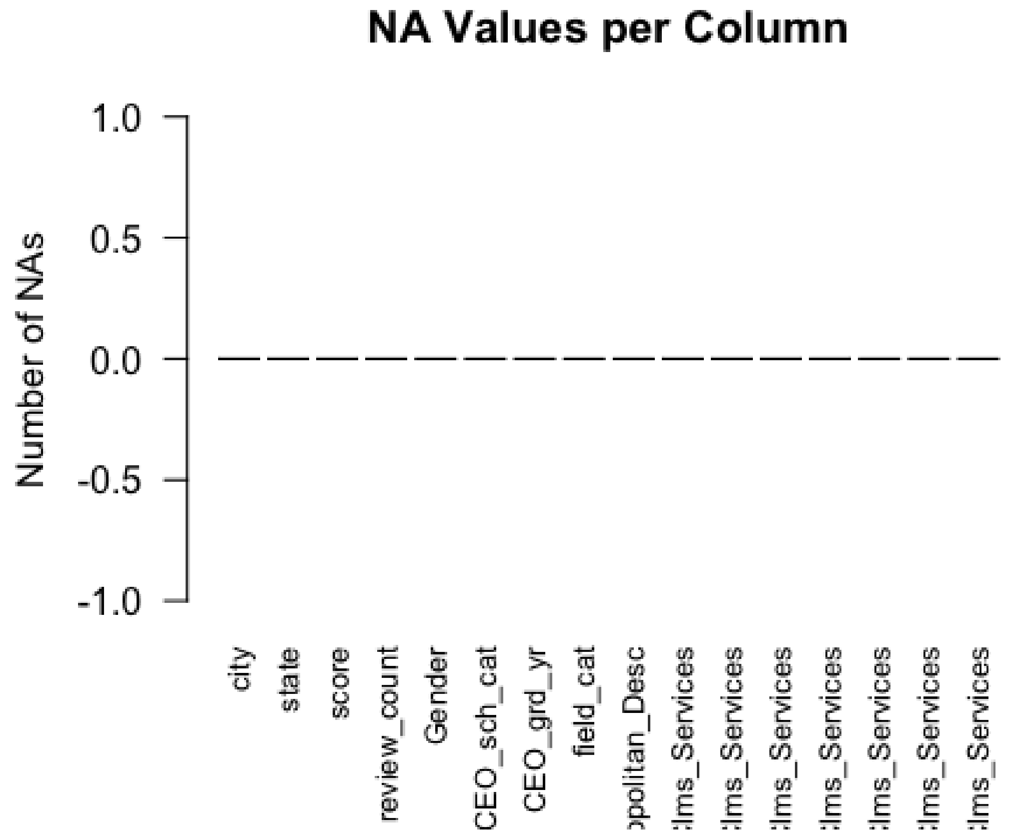
Categorical Data Categorical variables were treated differently. Given that central tendency measures such as the mean or median are not applicable, we imputed missing values using the mode. The mode, defined as the most frequently occurring value in a dataset, was deemed most appropriate as it maintains the distribution of the data's categories, ensuring that the imputation process does not introduce bias.

```
##                     city                    state                    score 
##               0.00000000               0.00000000               0.00000000 
##             review_count                   Gender              CEO_sch_cat 
##               0.00000000               0.00000000               0.00000000 
##               CEO_grd_yr                field_cat  Rural_metropolitan_Desc 
##               0.07345515               0.00000000              45.17491507 
##        Tot_Clms_Services   Brnd_Tot_Clms_Services   Gnrc_Tot_Clms_Services 
##              45.16573317              68.95601873              45.78092003 
##   Othr_Tot_Clms_Services    LIS_Tot_Clms_Services Opioid_Tot_Clms_Services 
##              69.05701956              54.15480672              59.25993940 
## Antbtc_Tot_Clms_Services 
##              62.31750987
```

```
##                     city                    state                    score 
##                  0.00000                  0.00000                  0.00000 
##             review_count                   Gender              CEO_sch_cat 
##                  0.00000                  0.00000                  0.00000 
##               CEO_grd_yr                field_cat  Rural_metropolitan_Desc 
##                  0.00000                  0.00000                 45.17492 
##        Tot_Clms_Services   Brnd_Tot_Clms_Services   Gnrc_Tot_Clms_Services 
##                  0.00000                  0.00000                  0.00000 
##   Othr_Tot_Clms_Services    LIS_Tot_Clms_Services Opioid_Tot_Clms_Services 
##                  0.00000                  0.00000                  0.00000 
## Antbtc_Tot_Clms_Services 
##                  0.00000
```

The values NA’s have been succesfully imputed

```
## [1] 4920
```

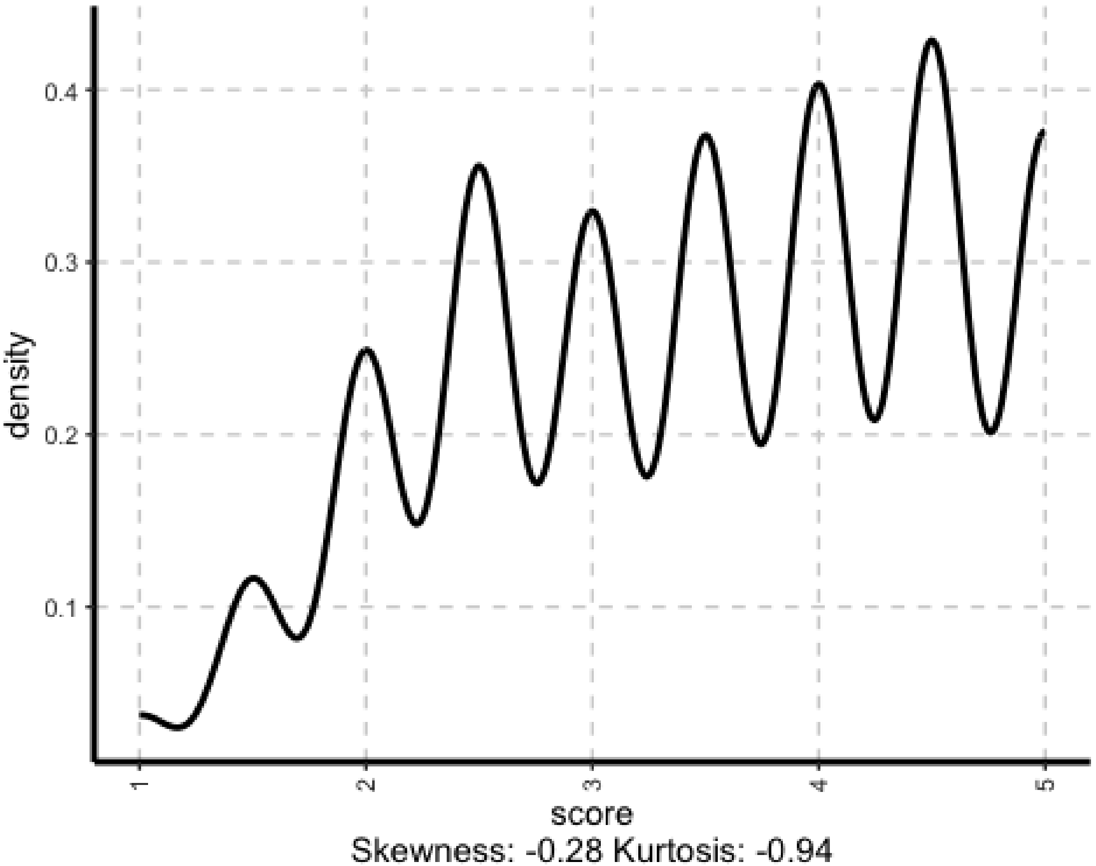


```
##    Index            Variable_Name Variable_Type Sample_n Missing_Count
## 1      1                     city     character    10891             0
## 2      2                    state     character    10891             0
## 3      3                    score       numeric    10891             0
## 4      4             review_count       numeric    10891             0
## 5      5                   Gender     character    10891             0
## 6      6              CEO_sch_cat       numeric    10891             0
## 7      7               CEO_grd_yr       numeric    10891             0
## 8      8                field_cat       numeric    10891             0
## 9      9  Rural_metropolitan_Desc     character     5971          4920
## 10    10        Tot_Clms_Services       numeric    10891             0
## 11    11   Brnd_Tot_Clms_Services       numeric    10891             0
## 12    12   Gnrc_Tot_Clms_Services       numeric    10891             0
## 13    13   Othr_Tot_Clms_Services       numeric    10891             0
## 14    14    LIS_Tot_Clms_Services       numeric    10891             0
## 15    15 Opioid_Tot_Clms_Services       numeric    10891             0
## 16    16 Antbtc_Tot_Clms_Services       numeric    10891             0
##    Per_of_Missing No_of_distinct_values
## 1           0.000                   484
## 2           0.000                    15
## 3           0.000                     9
## 4           0.000                   164
## 5           0.000                     2
## 6           0.000                   227
## 7           0.000                    61
## 8           0.000                    77
## 9           0.452                    13
## 10          0.000                  1368
## 11          0.000                   620
## 12          0.000                  1288
## 13          0.000                   119
## 14          0.000                   861
## 15          0.000                   320
## 16          0.000                   293
```

#2.3 Comparsion of Orginal_data and clean data

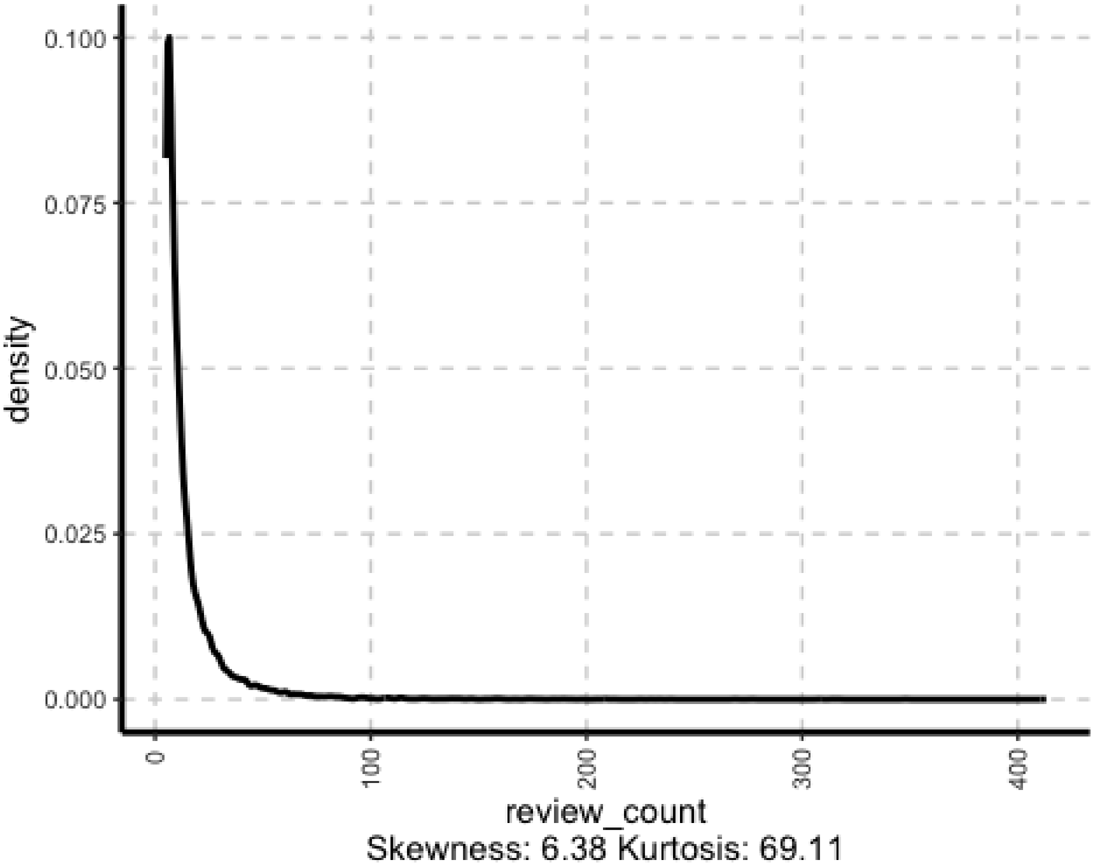
The data cleaning process notably enhanced the dataset’s integrity by effectively managing missing values, reducing them to zero percent in key variables. It successfully reduced skewness and kurtosis in numeric variables, indicating outlier management and distribution normalization. Transformations led to more uniform distributions, improving data consistency. Variable integrity, particularly in categorical variables, was preserved. Overall, the cleaning process optimized the dataset for advanced analysis, addressing missing data, outliers, and distributional biases, thus notably improving the reliability and accuracy of derived insights.

```
##    Index            Variable_Name Variable_Type Sample_n Missing_Count
## 1      1                     city     character    10891             0
## 2      2                    state     character    10891             0
## 3      3                    score       numeric    10891             0
## 4      4             review_count       numeric    10891             0
## 5      5                   Gender     character    10891             0
## 6      6              CEO_sch_cat       numeric    10891             0
## 7      7               CEO_grd_yr       numeric    10883             8
## 8      8                field_cat       numeric    10891             0
## 9      9  Rural_metropolitan_Desc     character     5971          4920
## 10    10        Tot_Clms_Services       numeric     5972          4919
## 11    11   Brnd_Tot_Clms_Services       numeric     3381          7510
## 12    12   Gnrc_Tot_Clms_Services       numeric     5905          4986
## 13    13   Othr_Tot_Clms_Services       numeric     3370          7521
## 14    14    LIS_Tot_Clms_Services       numeric     4993          5898
## 15    15 Opioid_Tot_Clms_Services       numeric     4437          6454
## 16    16 Antbtc_Tot_Clms_Services       numeric     4104          6787
##    Per_of_Missing No_of_distinct_values
## 1           0.000                   484
## 2           0.000                    15
## 3           0.000                     9
## 4           0.000                   164
## 5           0.000                     2
## 6           0.000                   227
## 7           0.001                    61
## 8           0.000                    77
## 9           0.452                    13
## 10          0.452                   1368
## 11          0.690                    620
## 12          0.458                   1288
## 13          0.691                    119
## 14          0.542                    861
## 15          0.593                    320
## 16          0.623                    293

##    Index            Variable_Name Variable_Type Sample_n Missing_Count
## 1      1                     city     character    10891             0
## 2      2                    state     character    10891             0
## 3      3                    score       numeric    10891             0
## 4      4             review_count       numeric    10891             0
## 5      5                   Gender     character    10891             0
## 6      6              CEO_sch_cat       numeric    10891             0
## 7      7               CEO_grd_yr       numeric    10891             0
## 8      8                field_cat       numeric    10891             0
## 9      9  Rural_metropolitan_Desc     character     5971          4920
## 10    10        Tot_Clms_Services       numeric    10891             0
## 11    11   Brnd_Tot_Clms_Services       numeric    10891             0
## 12    12   Gnrc_Tot_Clms_Services       numeric    10891             0
## 13    13   Othr_Tot_Clms_Services       numeric    10891             0
## 14    14    LIS_Tot_Clms_Services       numeric    10891             0
## 15    15 Opioid_Tot_Clms_Services       numeric    10891             0
## 16    16 Antbtc_Tot_Clms_Services       numeric    10891             0
##    Per_of_Missing No_of_distinct_values
## 1           0.000                   484
## 2           0.000                    15
## 3           0.000                     9
## 4           0.000                   164
## 5           0.000                     2
## 6           0.000                   227
## 7           0.000                    61
## 8           0.000                    77
## 9           0.452                    13
## 10          0.000                  1368
## 11          0.000                   620
## 12          0.000                  1288
## 13          0.000                   119
## 14          0.000                   861
## 15          0.000                   320
## 16          0.000                   293

## [[1]]
```

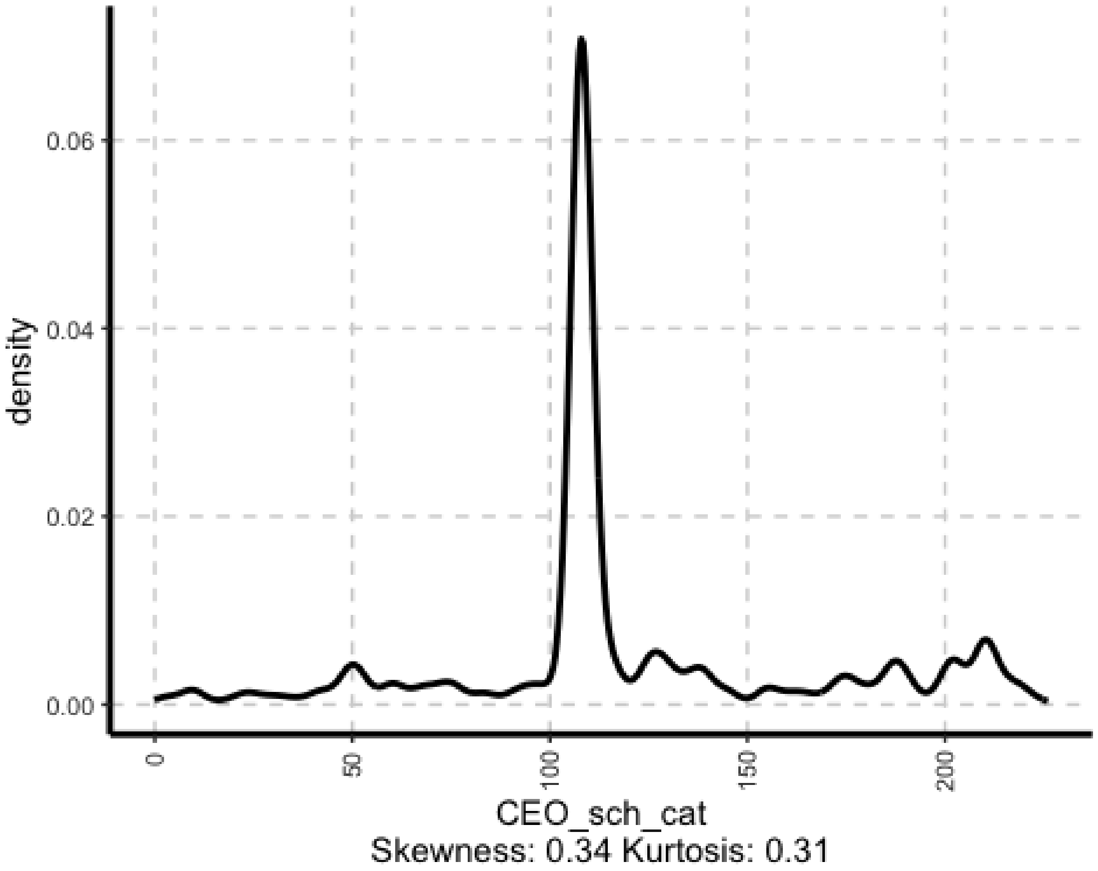


Skewness: -0.28 Kurtosis: -0.94

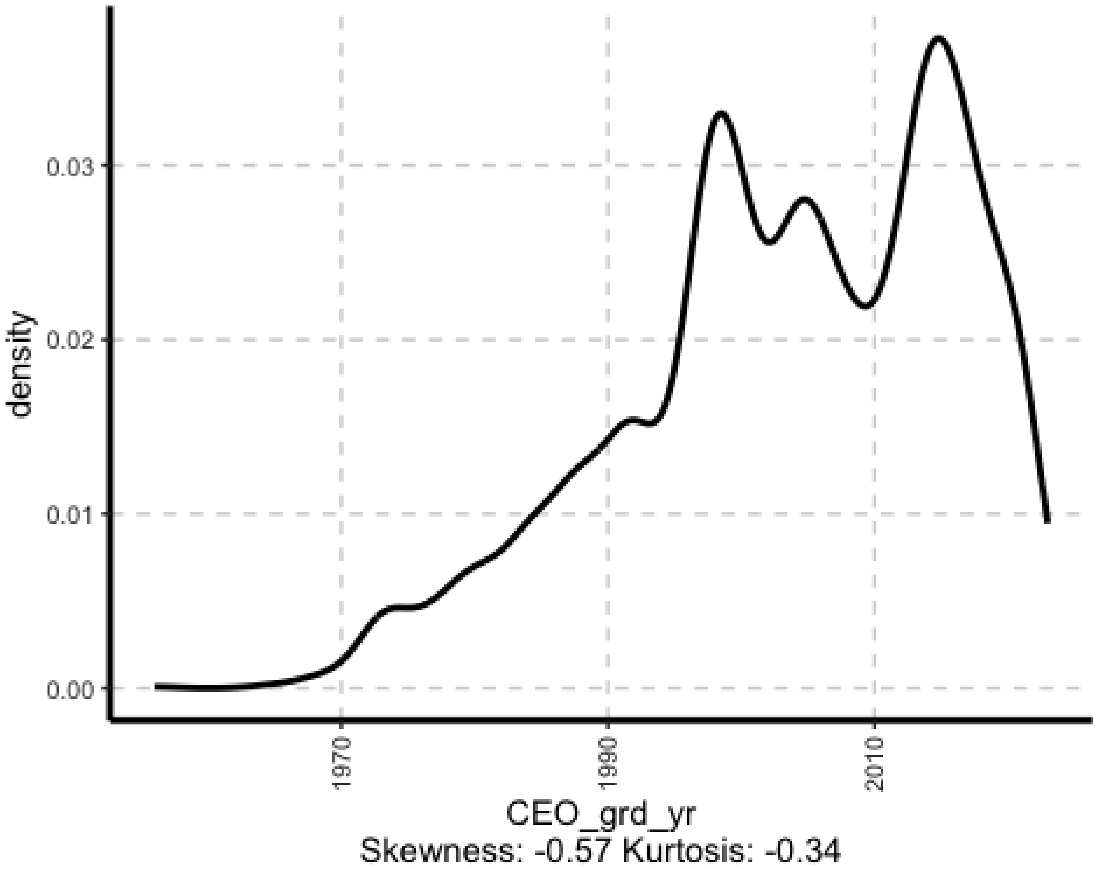
```
##
## [[2]]
```


density
0.100
0.075
0.050
0.025
0.000
0
100
200
300
400
review_count
Skewness: 6.38 Kurtosis: 69.11

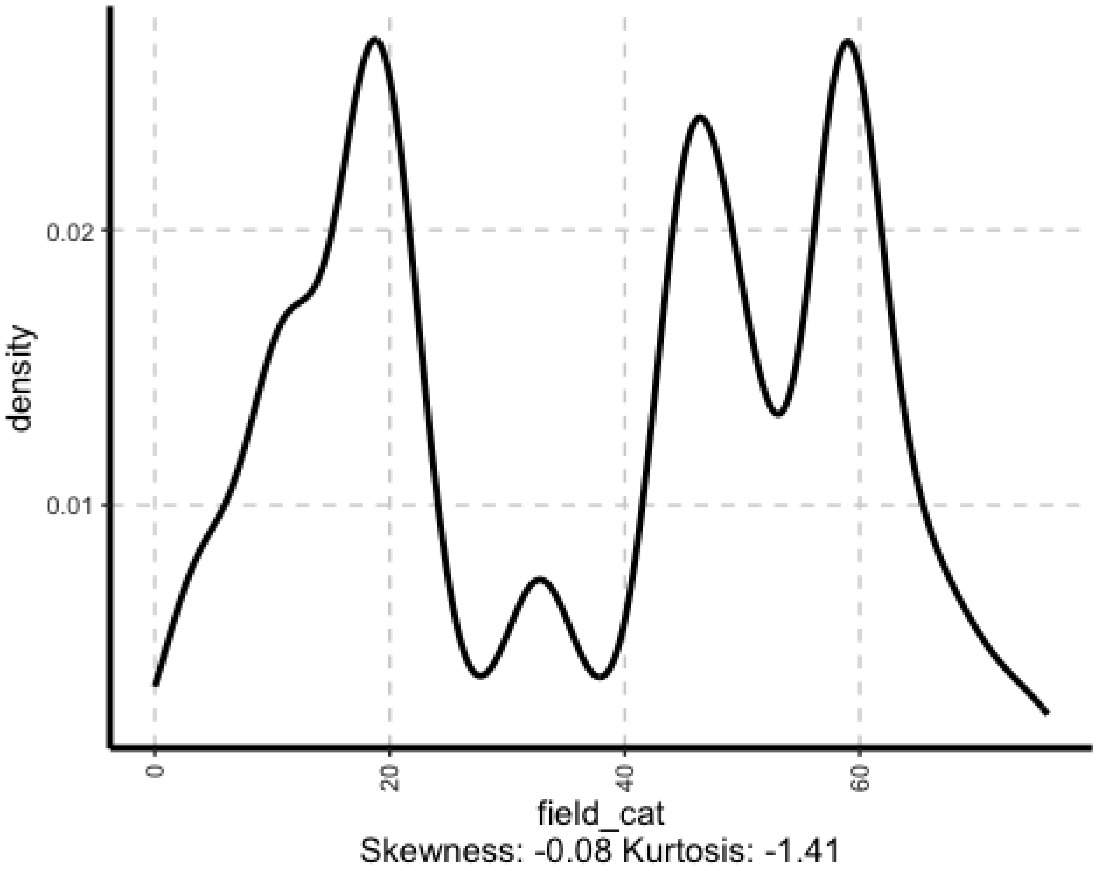
```
##
## [[3]]
```


density
0.06
0.04
0.02
0.00
0
50
100
150
200
CEO_sch_cat
Skewness: 0.34 Kurtosis: 0.31

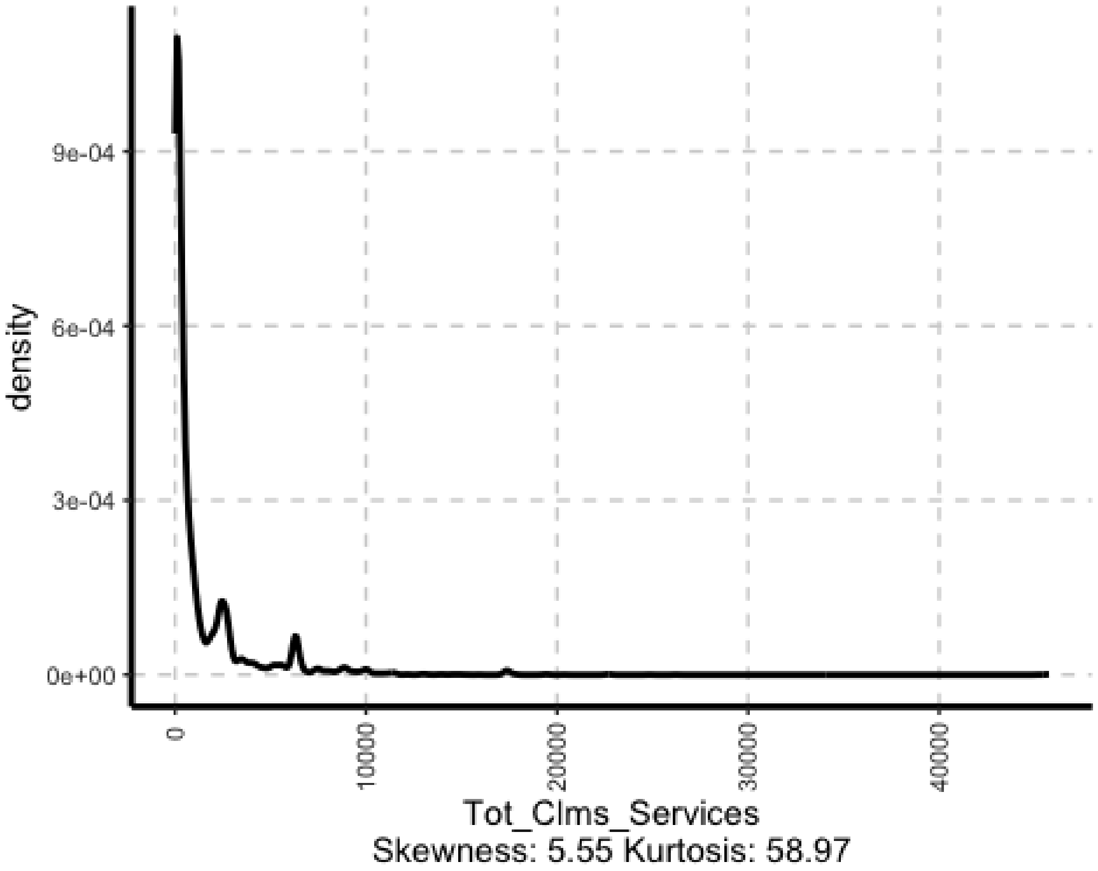
```
##
## [[4]]
```


density
0.03
0.02
0.01
0.00
1970
1990
2010
CEO_grd_yr
Skewness: -0.57 Kurtosis: -0.34

```
##
## [[5]]
```

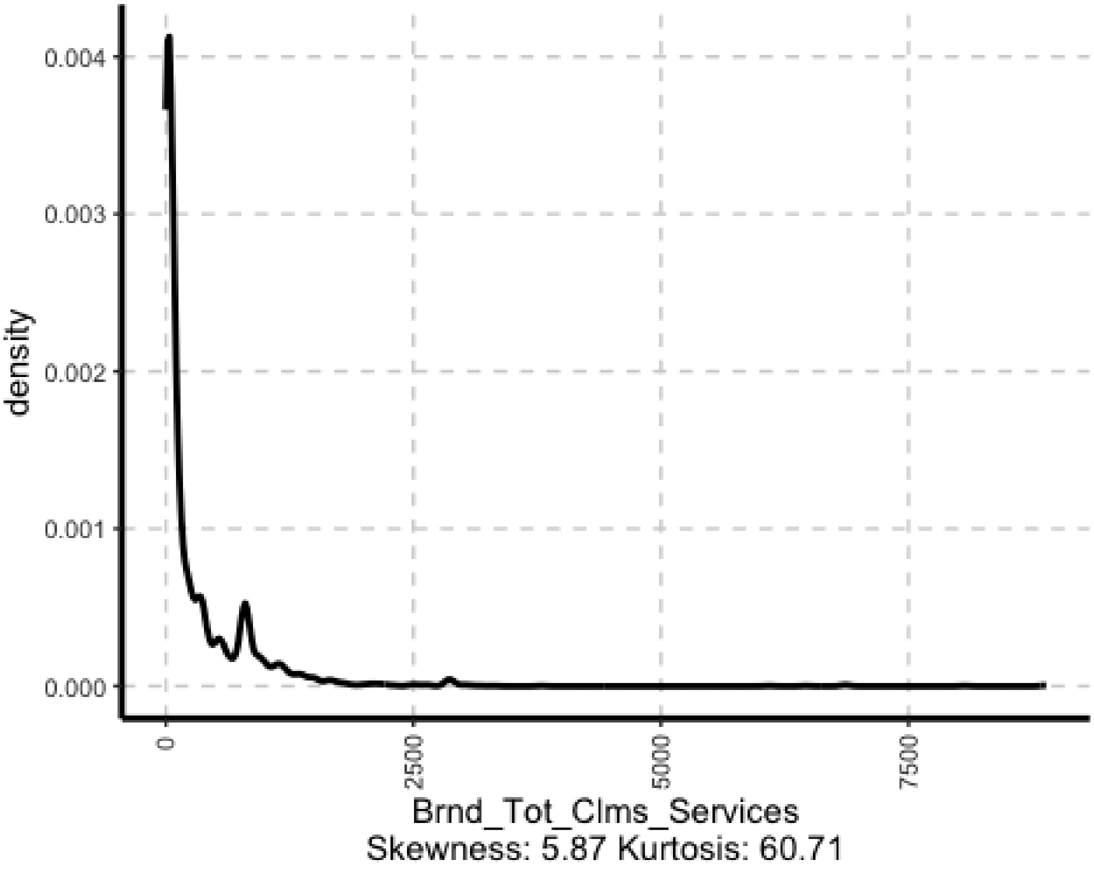

density
0.02
0.01
0
20
40
60
field_cat
Skewness: -0.08 Kurtosis: -1.41

```
##
## [[6]]
```



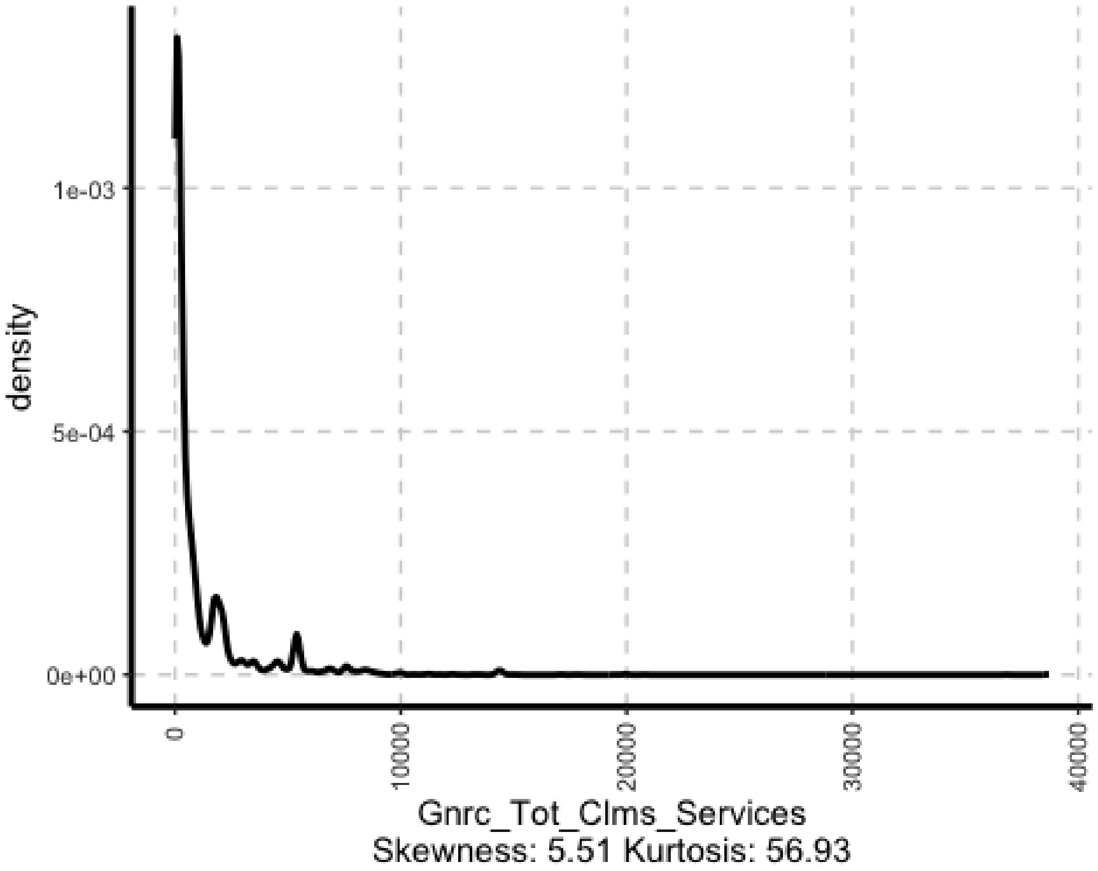
density
9e-04
6e-04
3e-04
0e+00
0
10000
20000
30000
40000
Tot_Clms_Services
Skewness: 5.55 Kurtosis: 58.97

```
##
## [[7]]
```

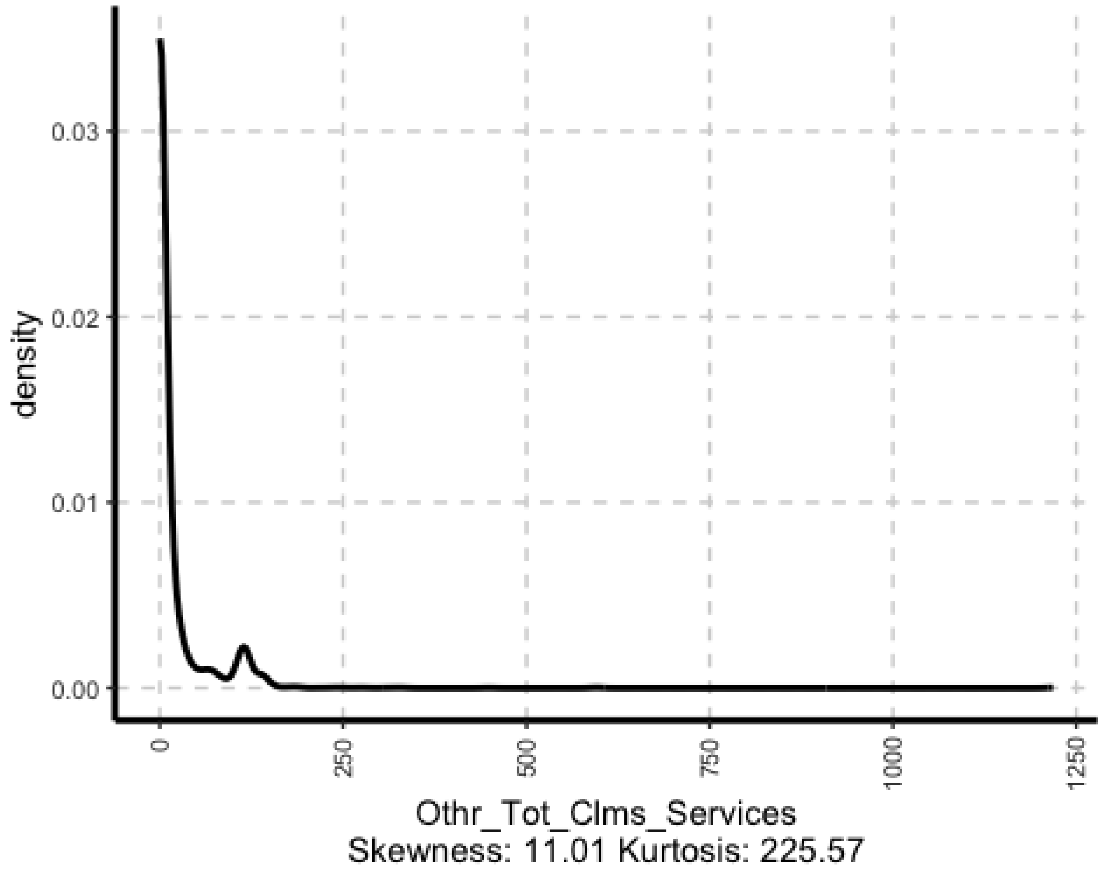


0.004
0.003
0.002
0.001
0.000
density
0
2500
5000
7500
Brnd_Tot_Clms_Services
Skewness: 5.87 Kurtosis: 60.71

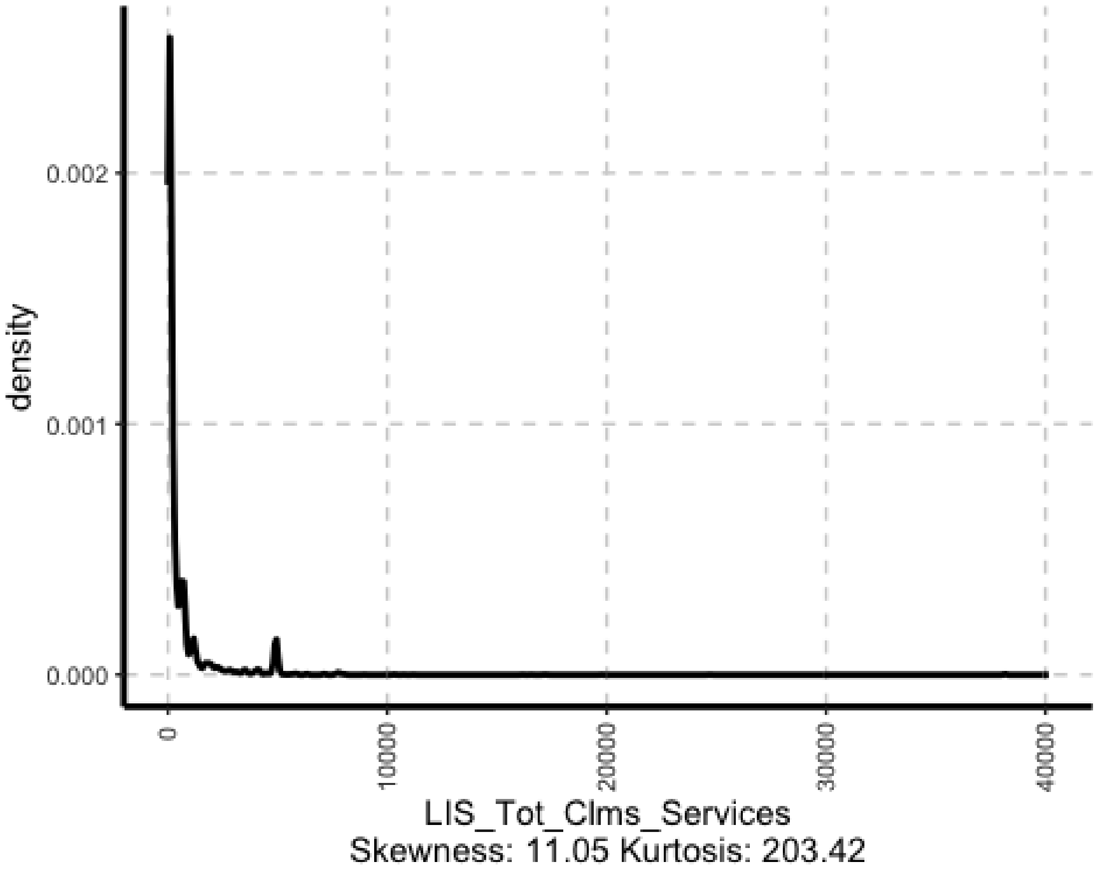
```
##
## [[8]]
```


density
1e-03
5e-04
0e+00
0
10000
20000
30000
40000
Gnrc_Tot_Clms_Services
Skewness: 5.51 Kurtosis: 56.93

```
##
## [[9]]
```

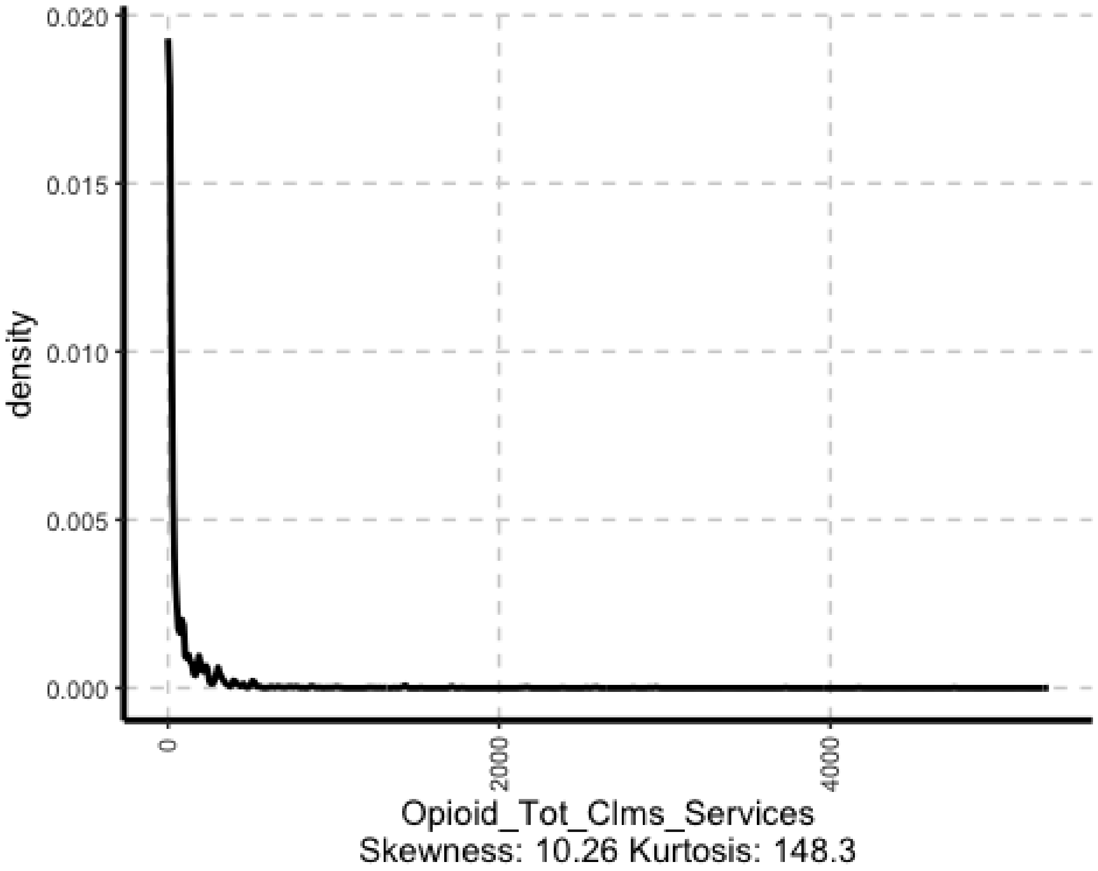

0.03
0.02
0.01
0.00
density
0
250
500
750
1000
1250
Othr_Tot_Clms_Services
Skewness: 11.01 Kurtosis: 225.57

```
##
## [[10]]
```

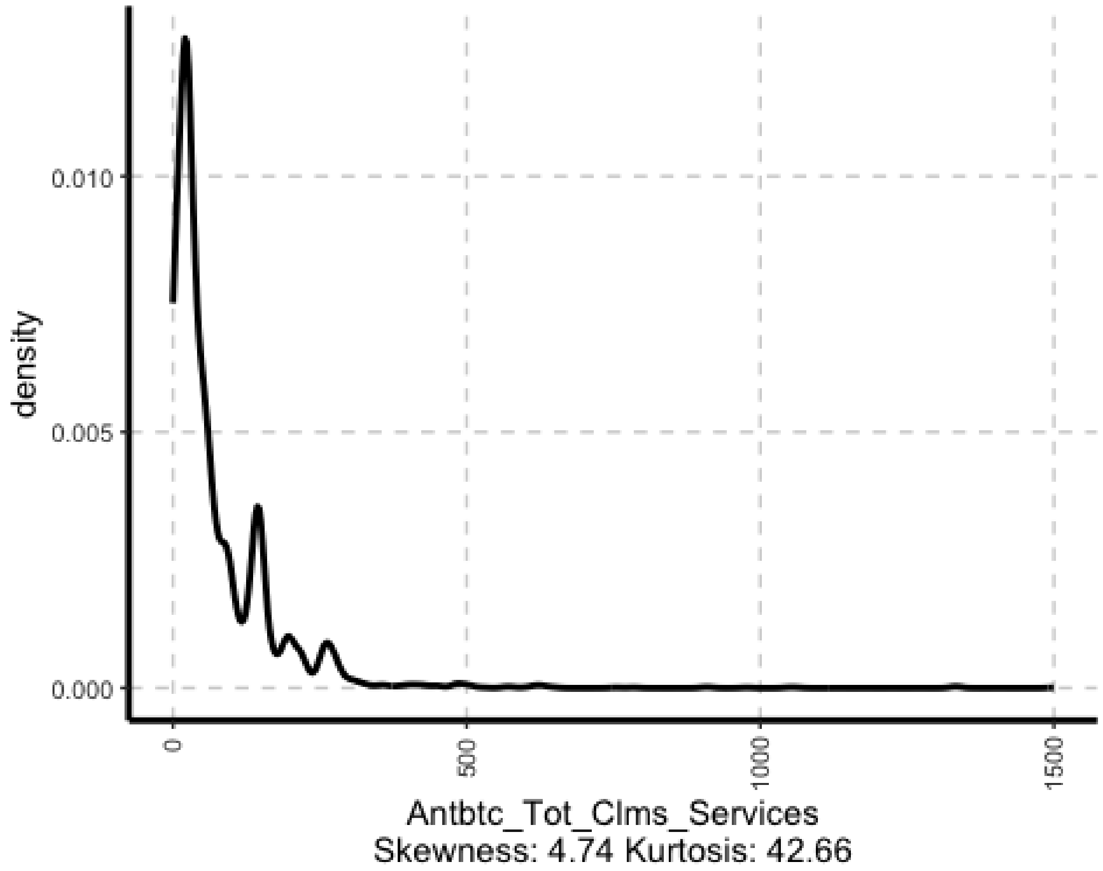


0.002
0.001
0.000
density
0
10000
20000
30000
40000
LIS_Tot_Clms_Services
Skewness: 11.05 Kurtosis: 203.42

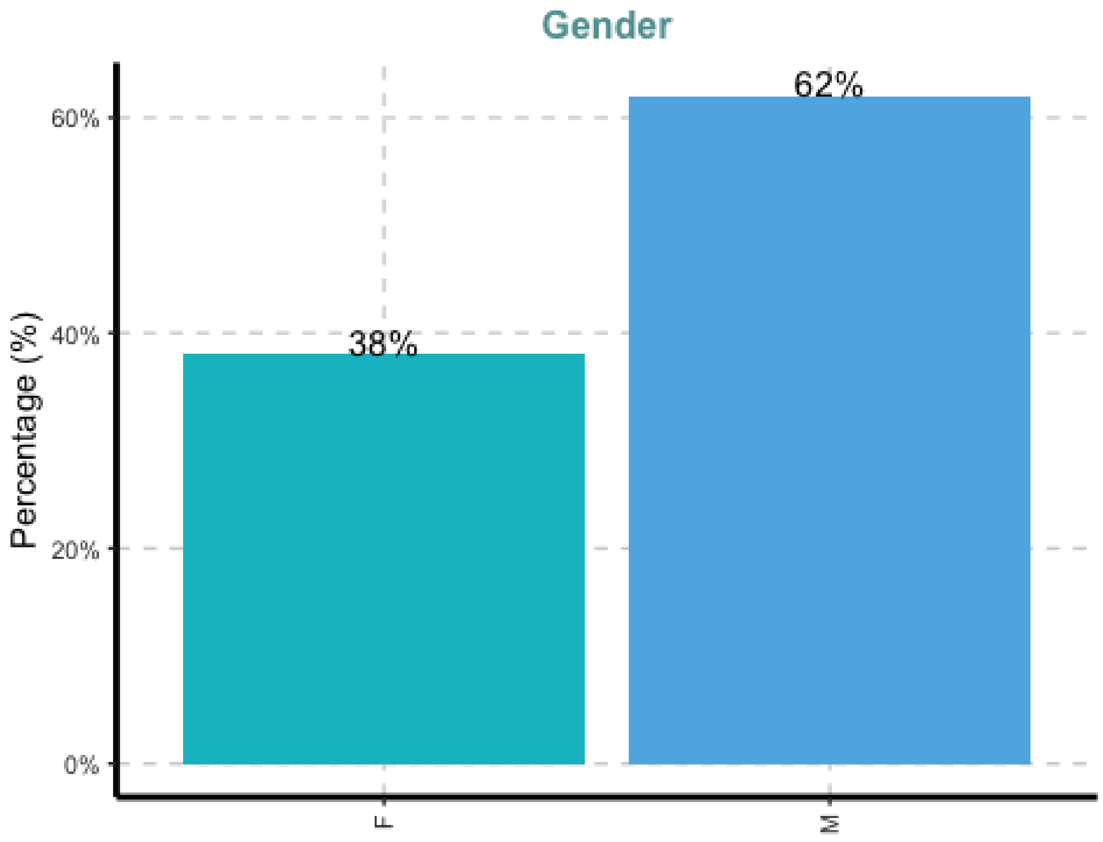
```
##
## [[11]]
```


0.020
0.015
0.010
0.005
0.000
density
0
2000
4000
Opioid_Tot_Clms_Services
Skewness: 10.26 Kurtosis: 148.3

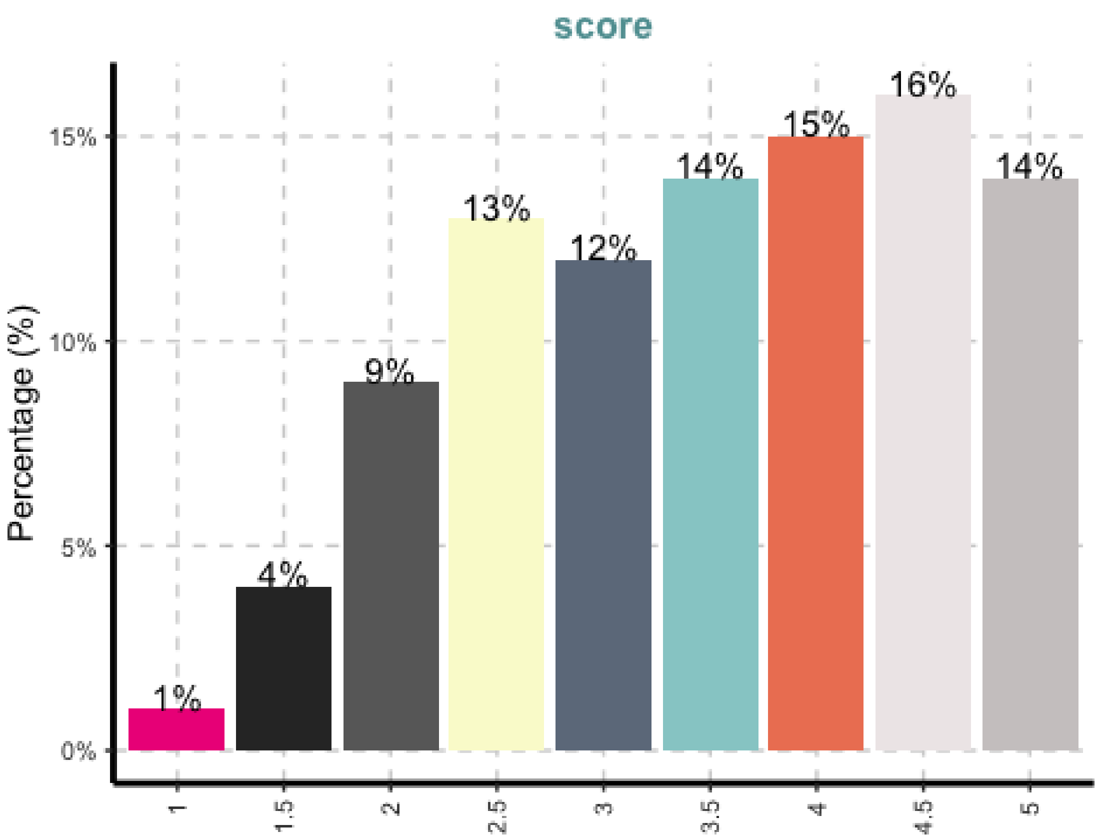
```
##
## [[12]]
```


density
0.010
0.005
0.000
0
500
1000
1500
Antbtc_Tot_Clms_Services
Skewness: 4.74 Kurtosis: 42.66

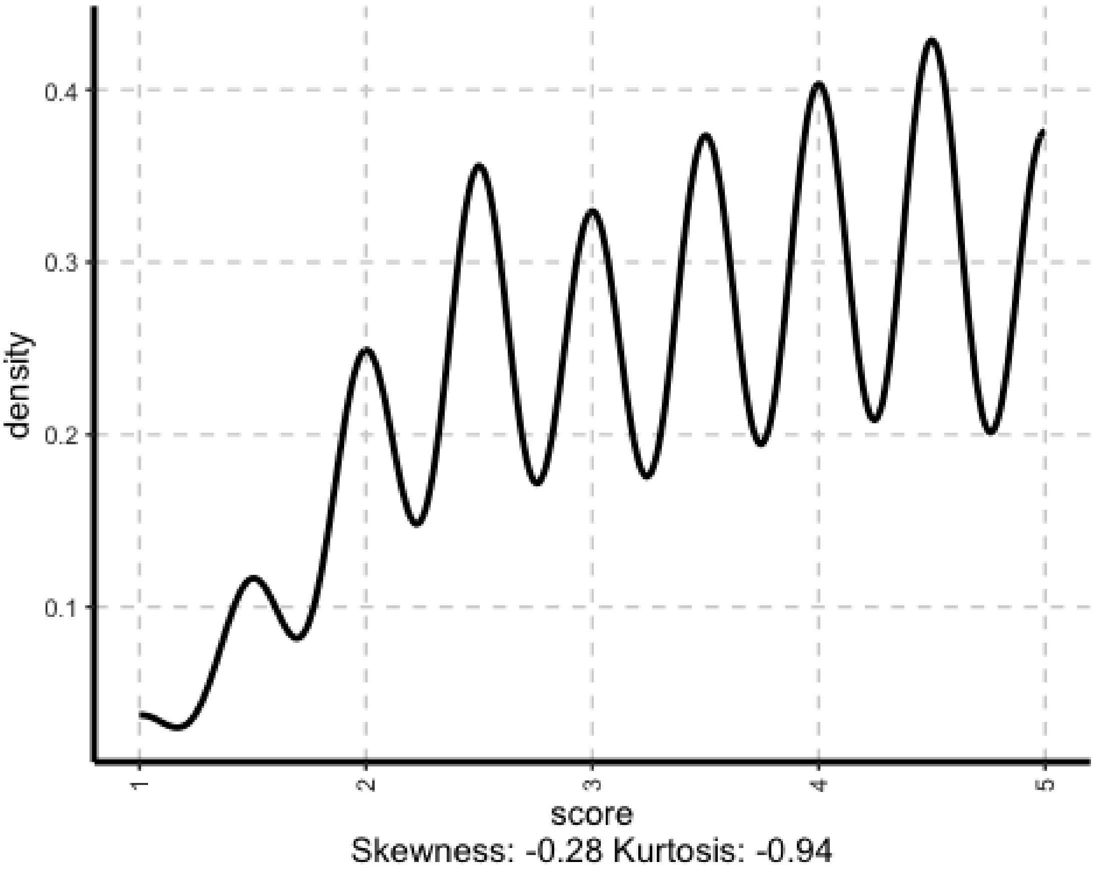
```
## [[1]]
```



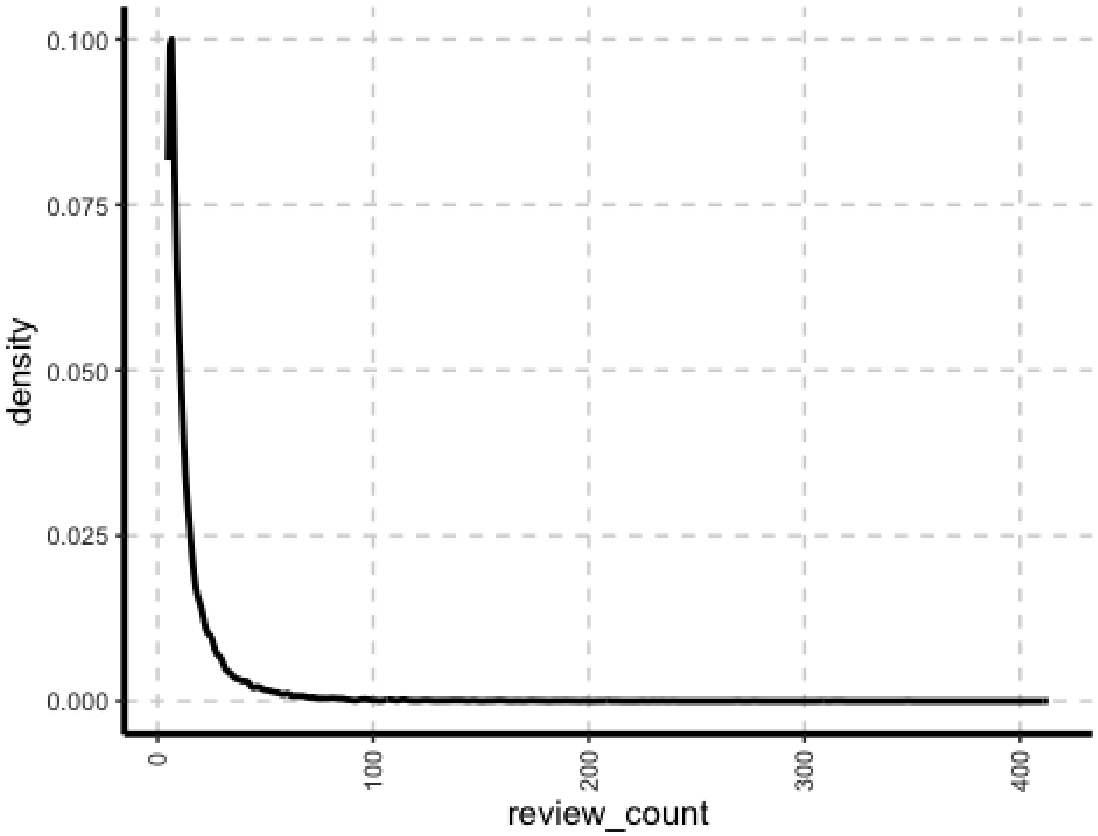
```
##
## [[2]]
```



```
## [[1]]
```

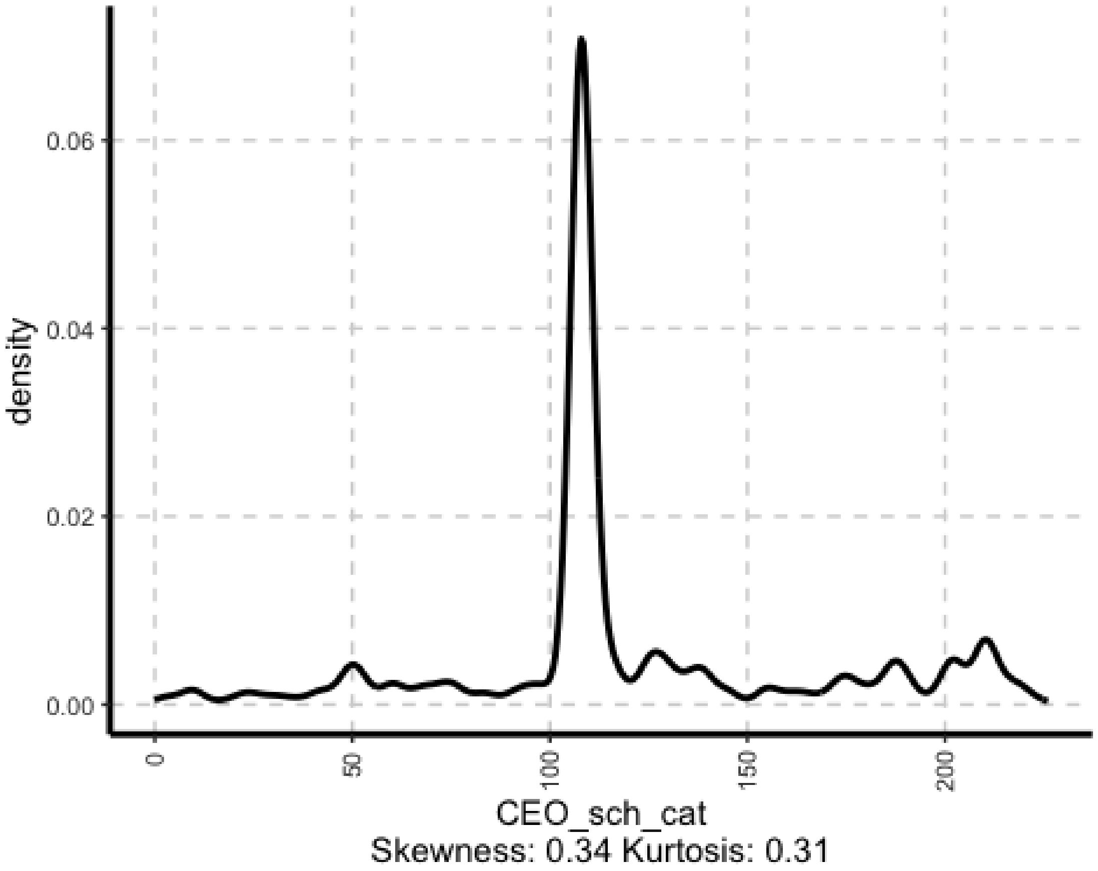


```
##
## [[2]]
```

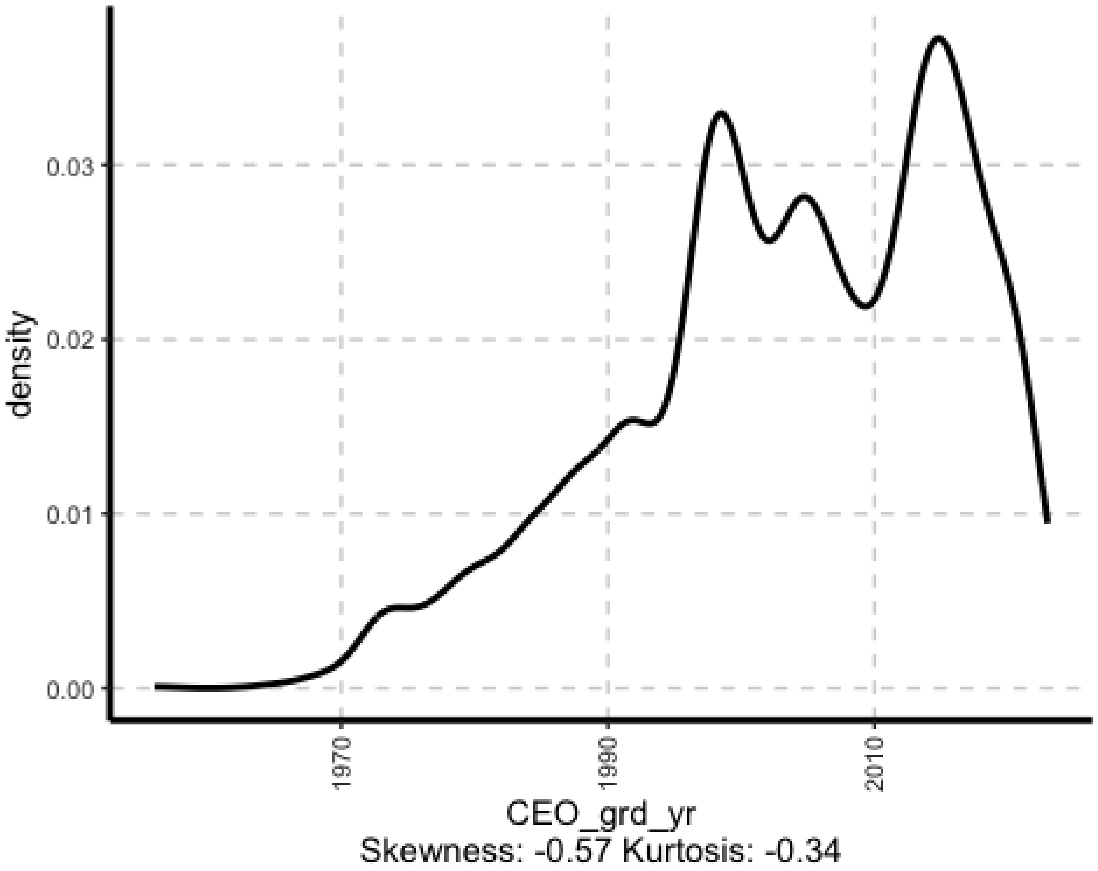


Skewness: 6.38 Kurtosis: 69.11

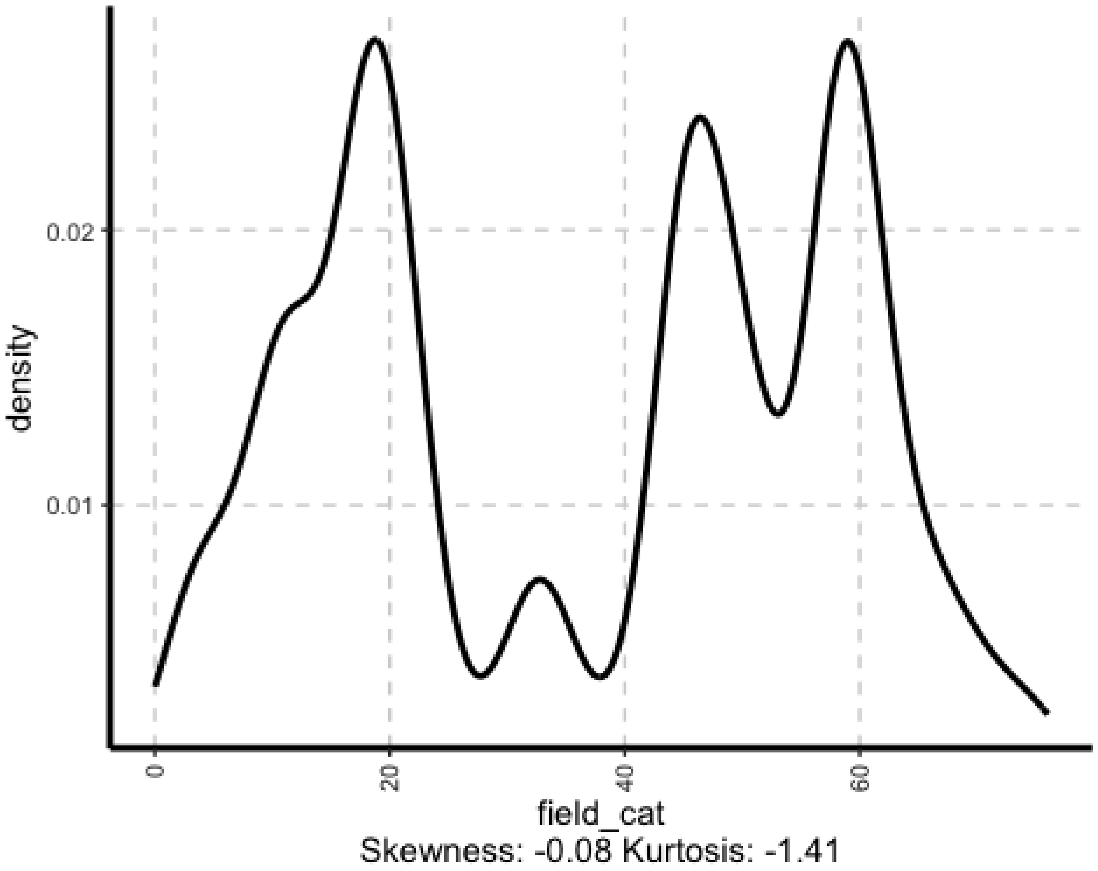
```
##
## [[3]]
```


density
0.06
0.04
0.02
0.00
0
50
100
150
200
CEO_sch_cat
Skewness: 0.34 Kurtosis: 0.31

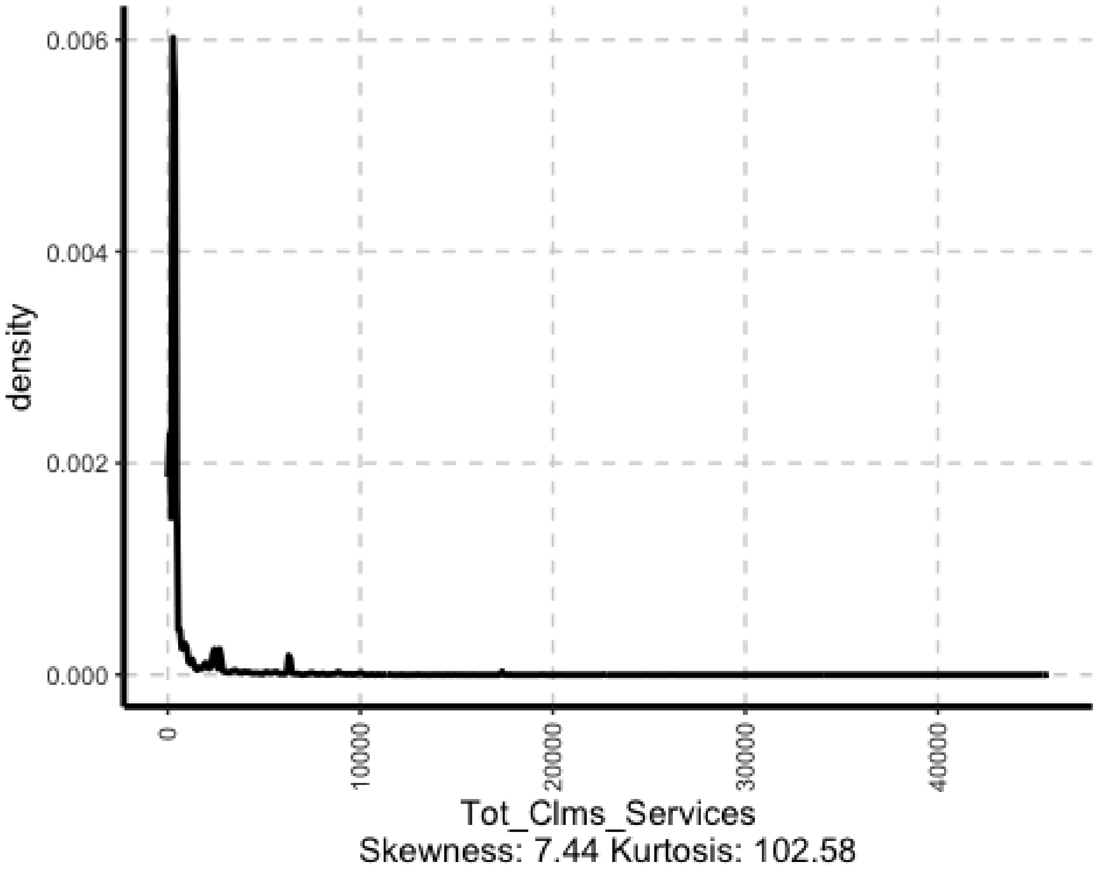
```
##
## [[4]]
```


0.03
0.02
0.01
0.00
density
1970
1990
2010
CEO_grd_yr
Skewness: -0.57 Kurtosis: -0.34

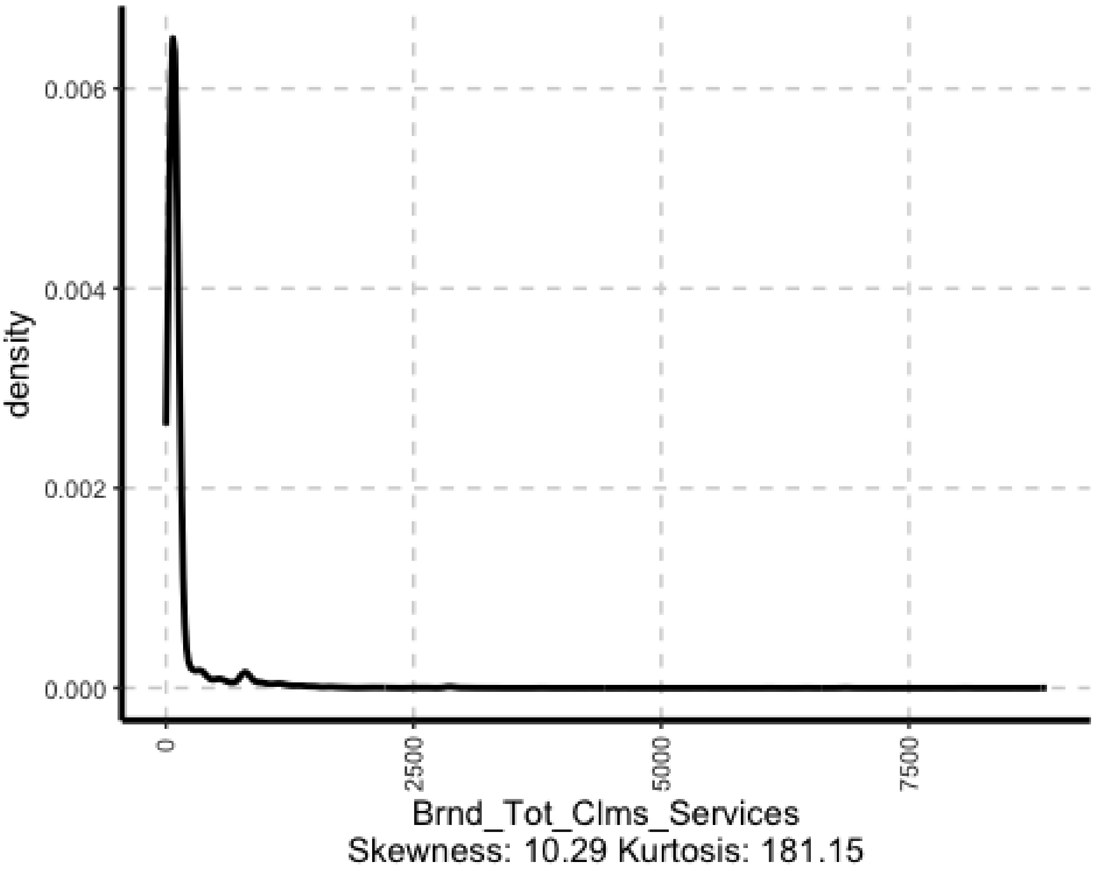
```
##
## [[5]]
```


density
0.02
0.01
0
20
40
60
field_cat
Skewness: -0.08 Kurtosis: -1.41

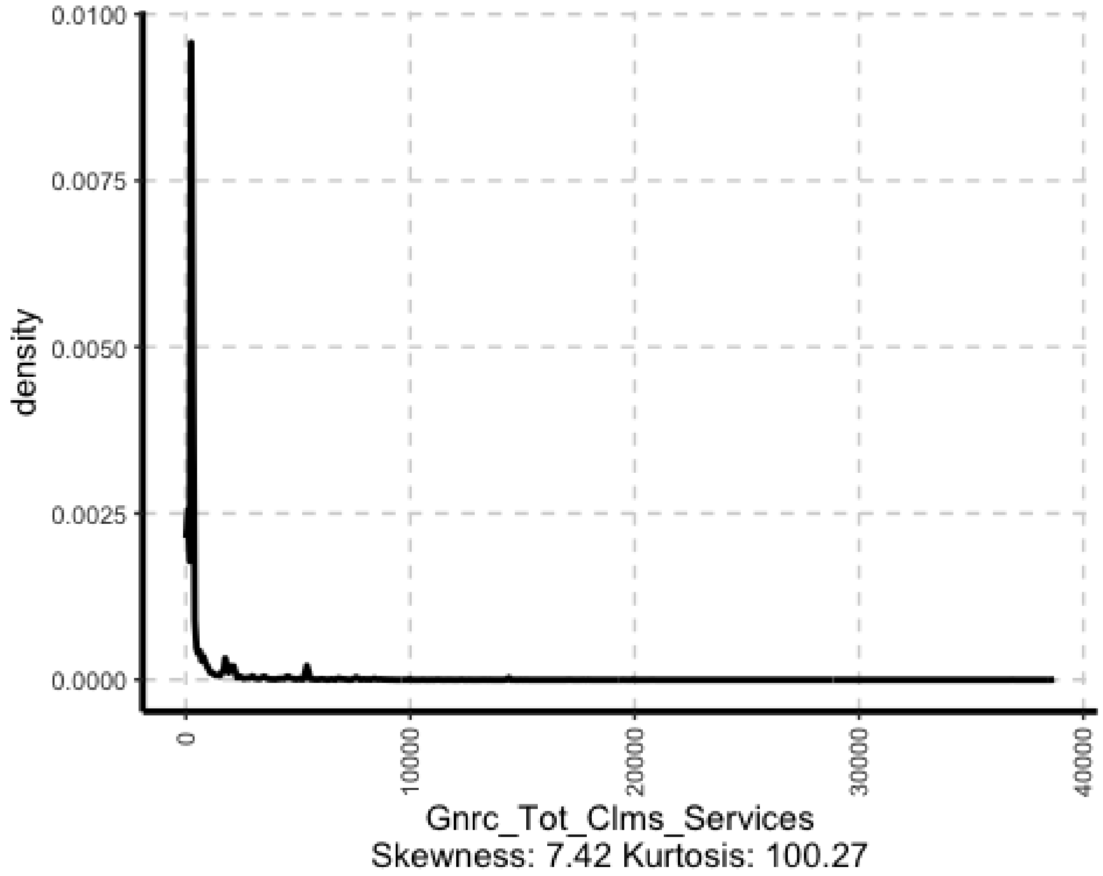
```
##
## [[6]]
```


density
0.006
0.004
0.002
0.000
0
10000
20000
30000
40000
Tot_Clms_Services
Skewness: 7.44 Kurtosis: 102.58

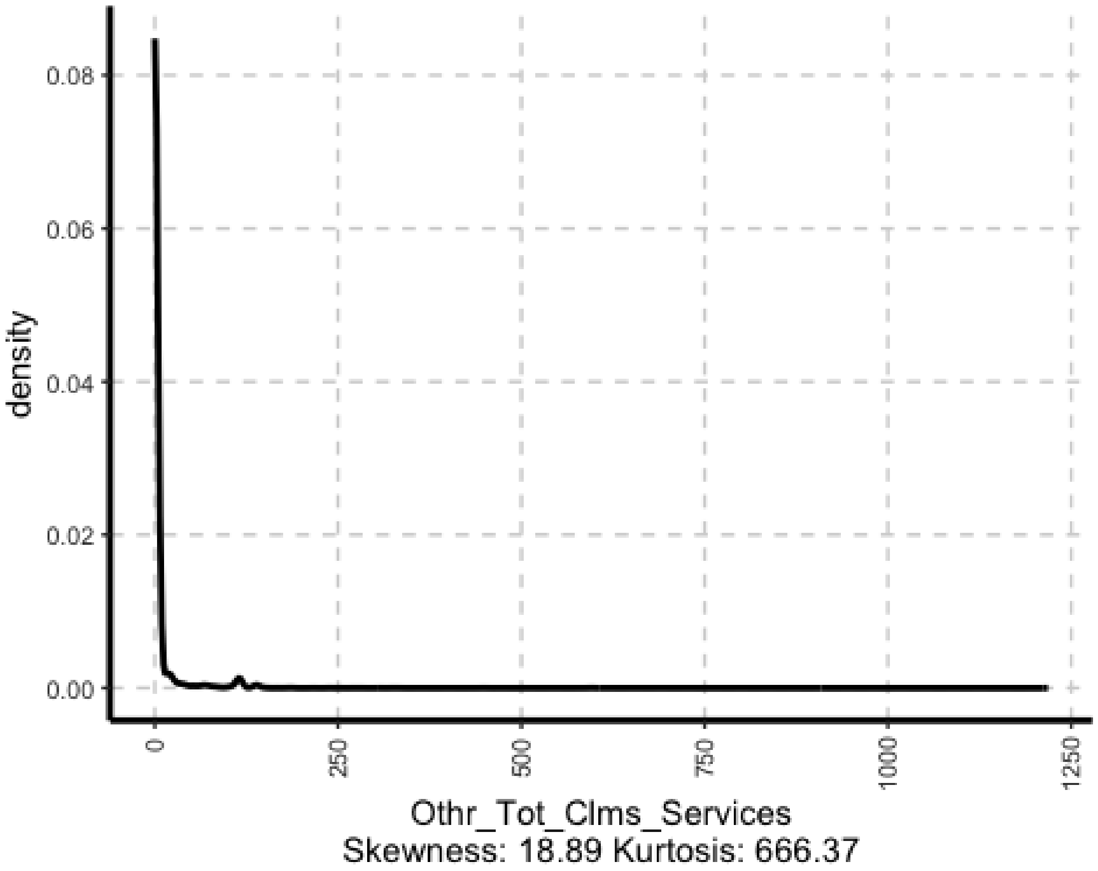
```
##
## [[7]]
```


density
0.006
0.004
0.002
0.000
0
2500
5000
7500
Brnd_Tot_Clms_Services
Skewness: 10.29 Kurtosis: 181.15

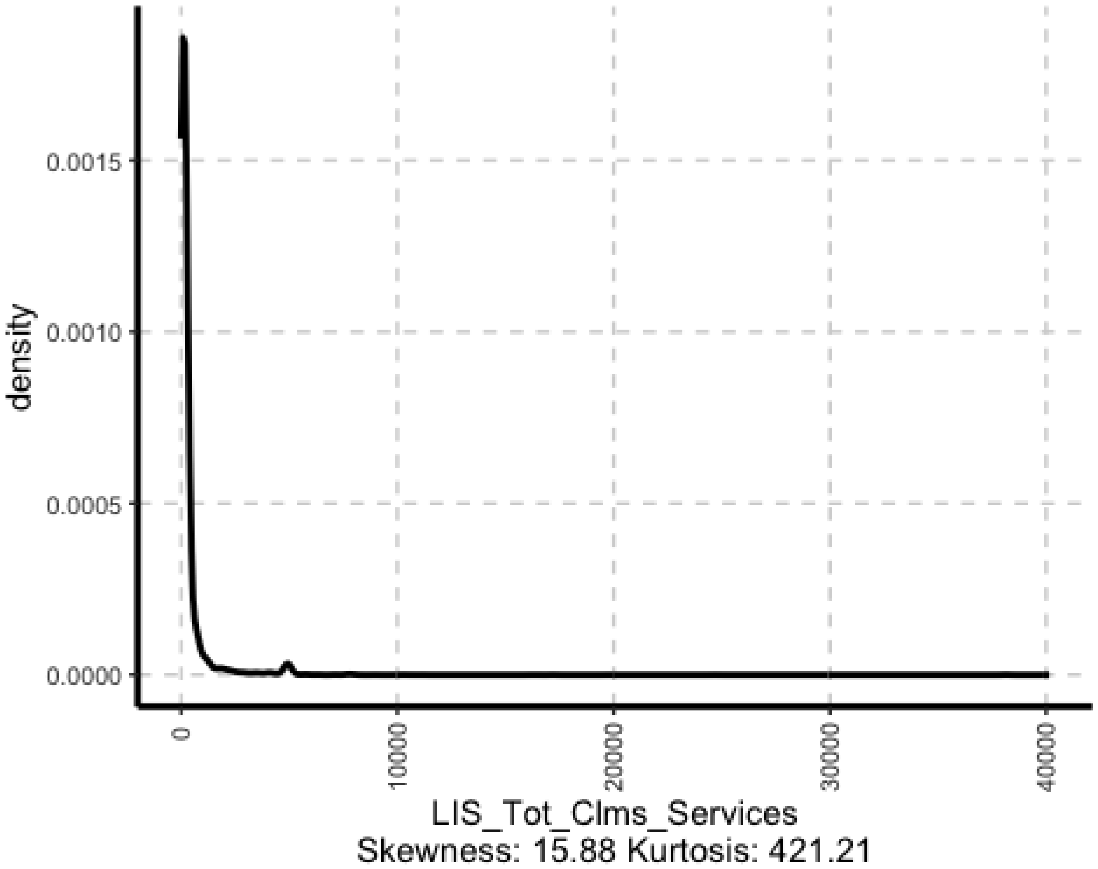
```
##
## [[8]]
```


0.0100
0.0075
0.0050
0.0025
0.0000
density
0
10000
20000
30000
40000
Gnrc_Tot_Clms_Services
Skewness: 7.42 Kurtosis: 100.27

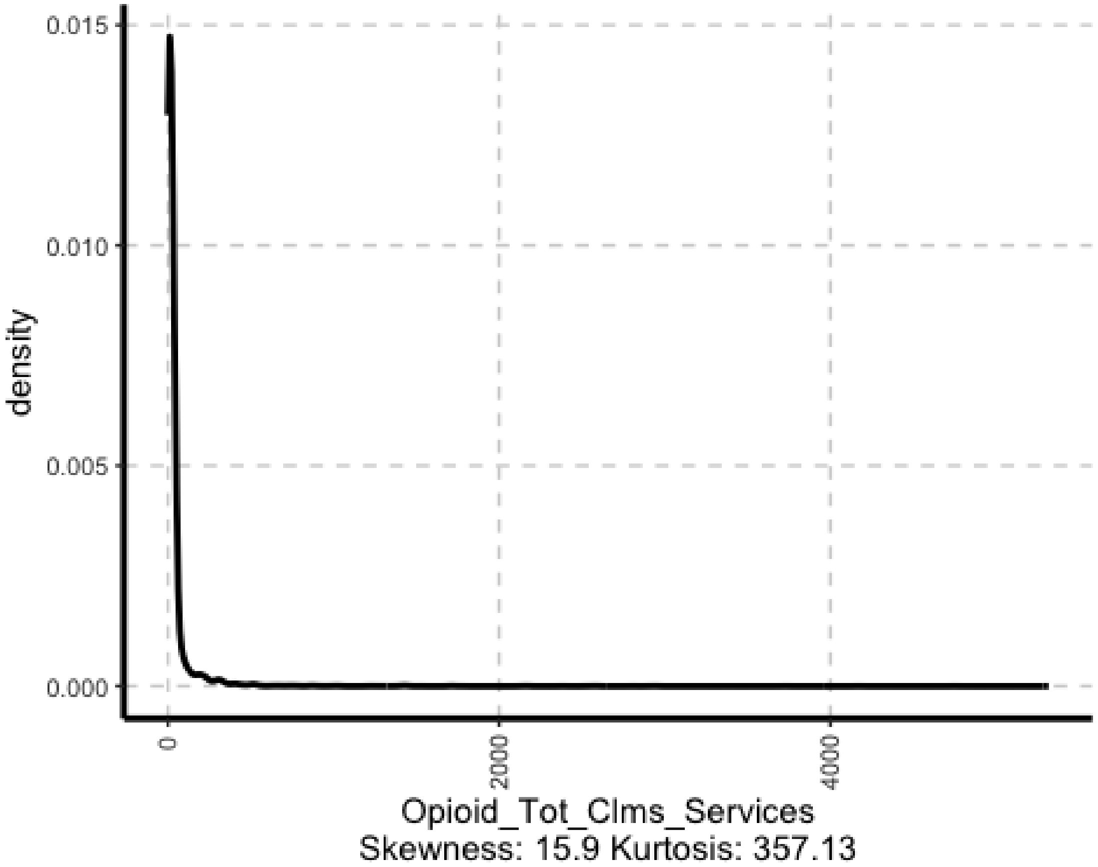
```
##
## [[9]]
```


0.08
0.06
0.04
0.02
0.00
density
0
250
500
750
1000
1250
Othr_Tot_Clms_Services
Skewness: 18.89 Kurtosis: 666.37

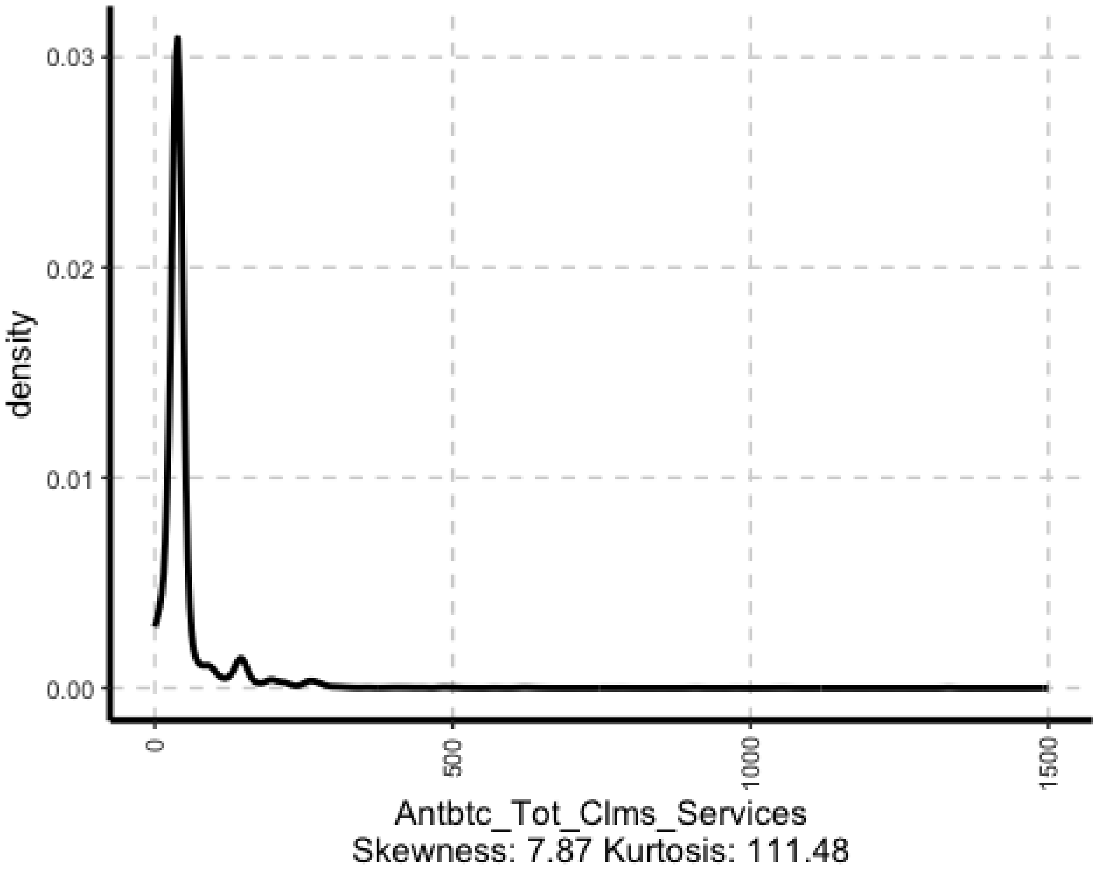
```
##
## [[10]]
```


density
0.0015
0.0010
0.0005
0.0000
0
10000
20000
30000
40000
LIS_Tot_Clms_Services
Skewness: 15.88 Kurtosis: 421.21

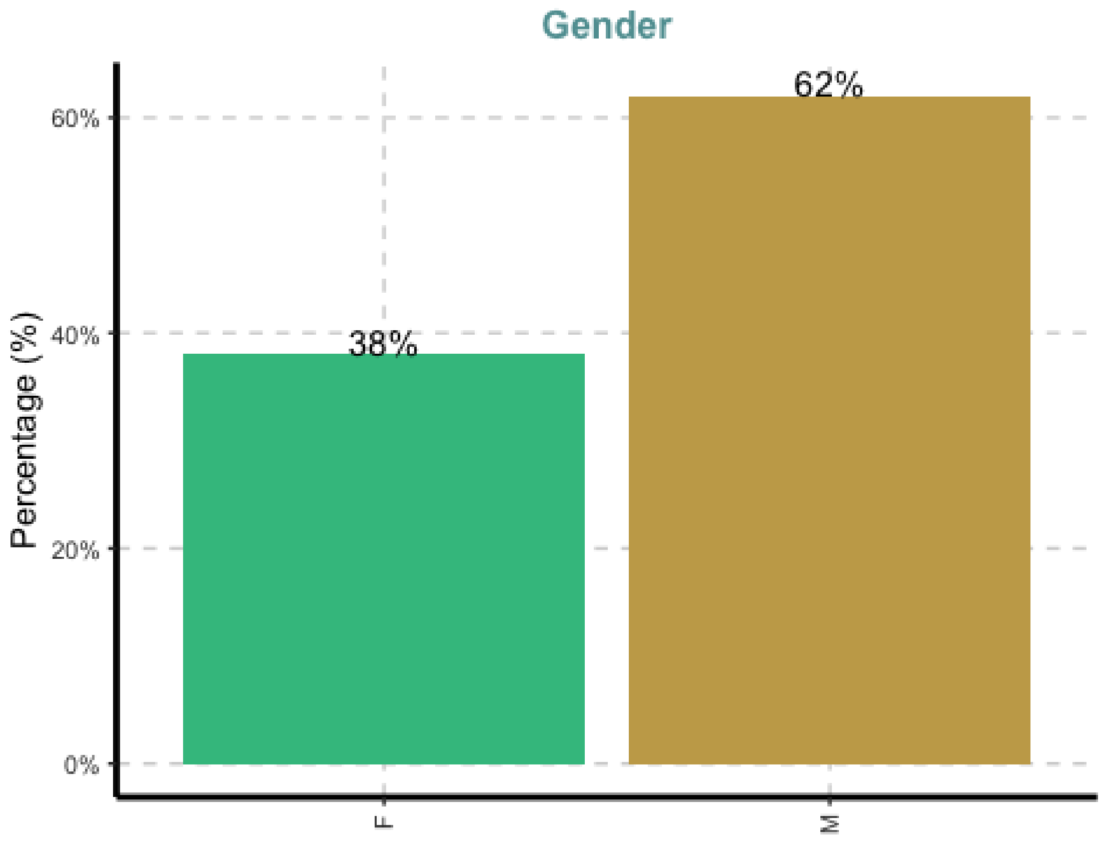
```
##
## [[11]]
```


0.015
0.010
0.005
0.000
density
0
2000
4000
Opioid_Tot_Clms_Services
Skewness: 15.9 Kurtosis: 357.13

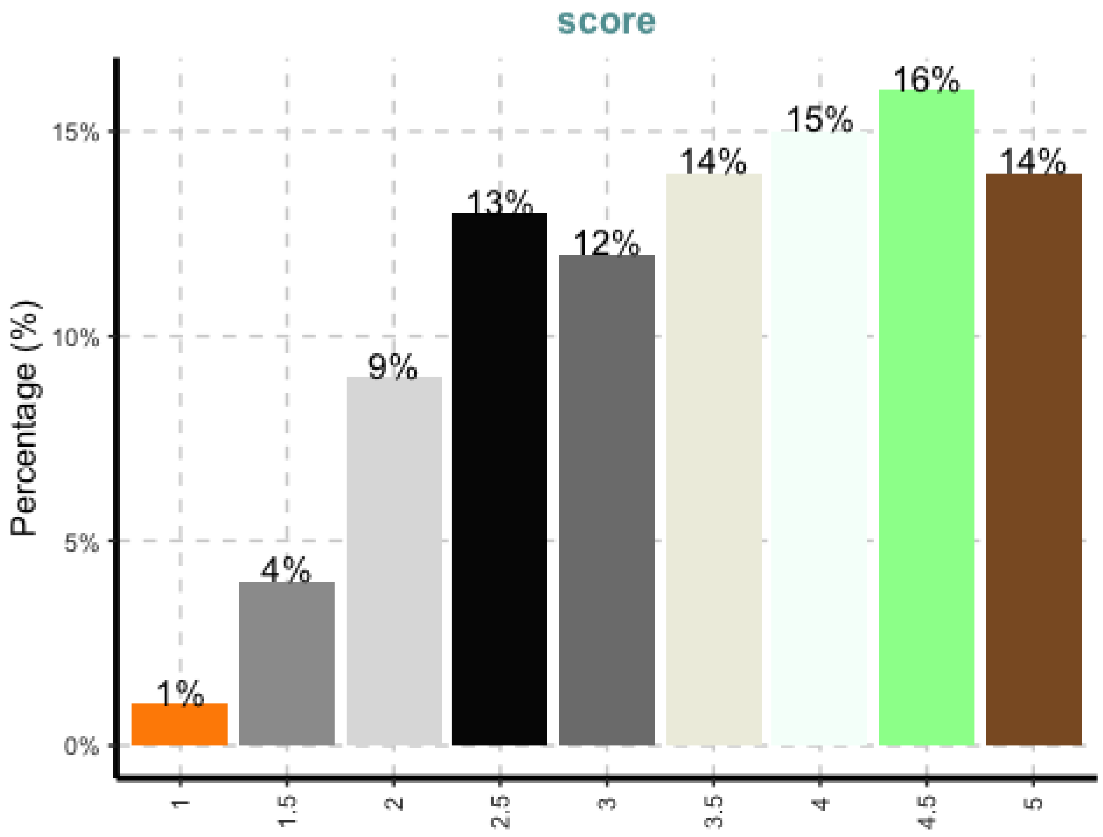
```
##
## [[12]]
```


0.03
0.02
0.01
0.00
density
0
500
1000
1500
Antbtc_Tot_Clms_Services
Skewness: 7.87 Kurtosis: 111.48

```
## [[1]]
```



```
##
## [[2]]
```



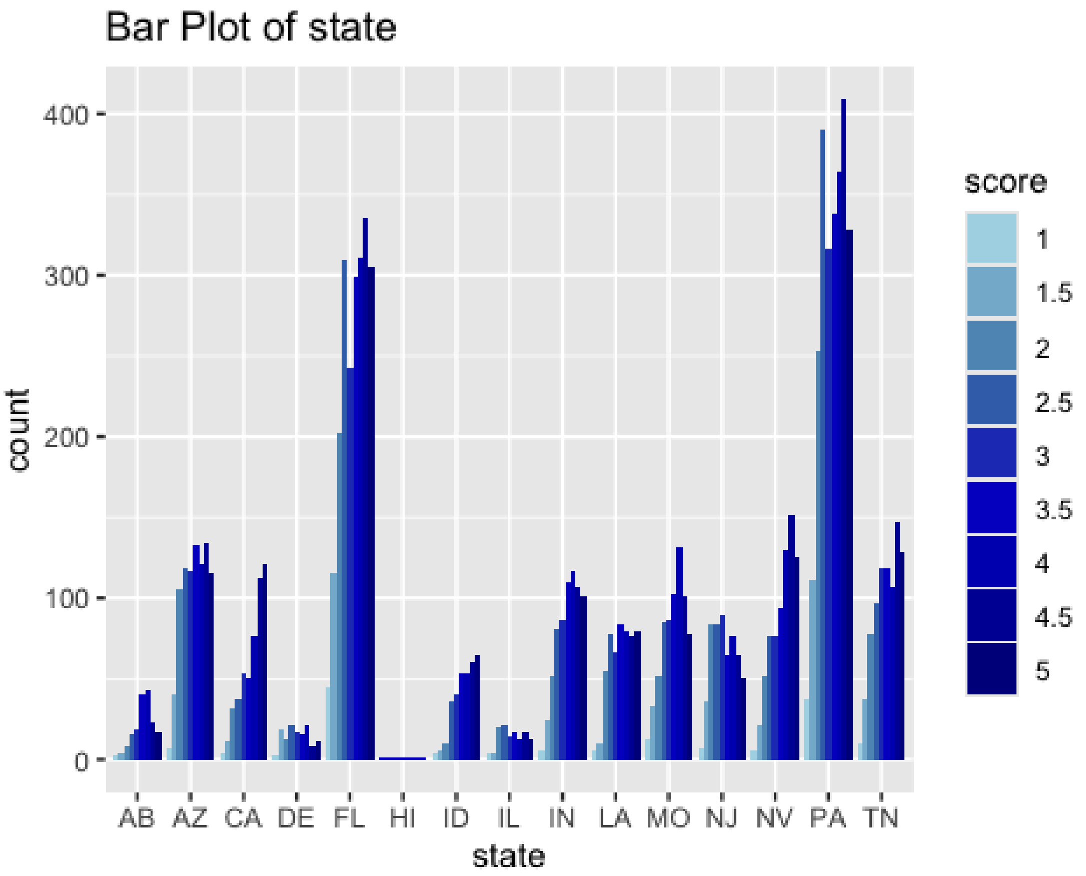
#Iteam A4 #2.4 Relationship with predictors #2.4.1 Catergorical variable

The State Distribution Plot displays the frequency of scores ranging from 1 to 5 across various states. California (CA) and Pennsylvania (PA) exhibit significantly high frequencies in most score categories, particularly in the scores of 2.5 and 5.

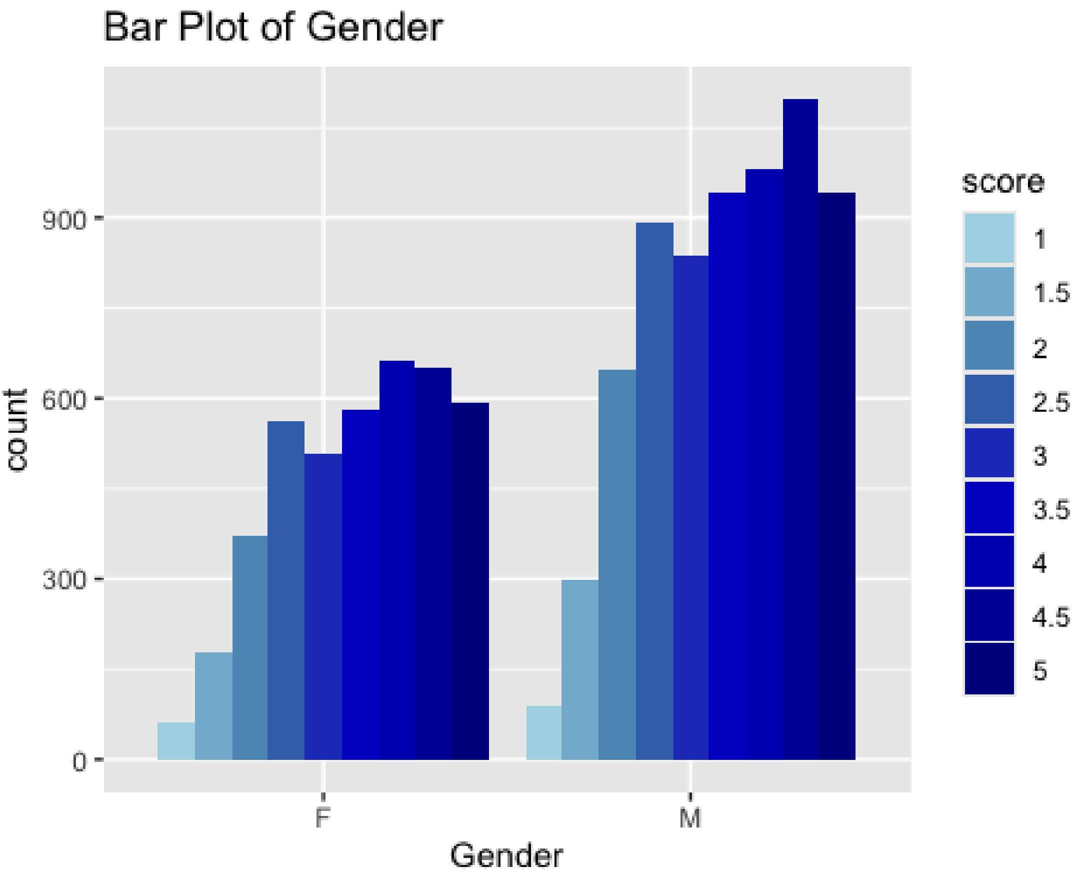
Rural vs. Metropolitan Plot: This plot compares the number of scores in rural and metropolitan locations. Rural areas exhibit a notable superiority in all ratings when compared to metropolitan areas, particularly in scores 2.5 and 5.

Gender Distribution Plot: This plot compares the scores of individuals identified as female (F) and male (M). Both genders exhibit a comparable distribution pattern, with a score of 5 having the highest frequency for both, followed by 2.5 and 3.5.

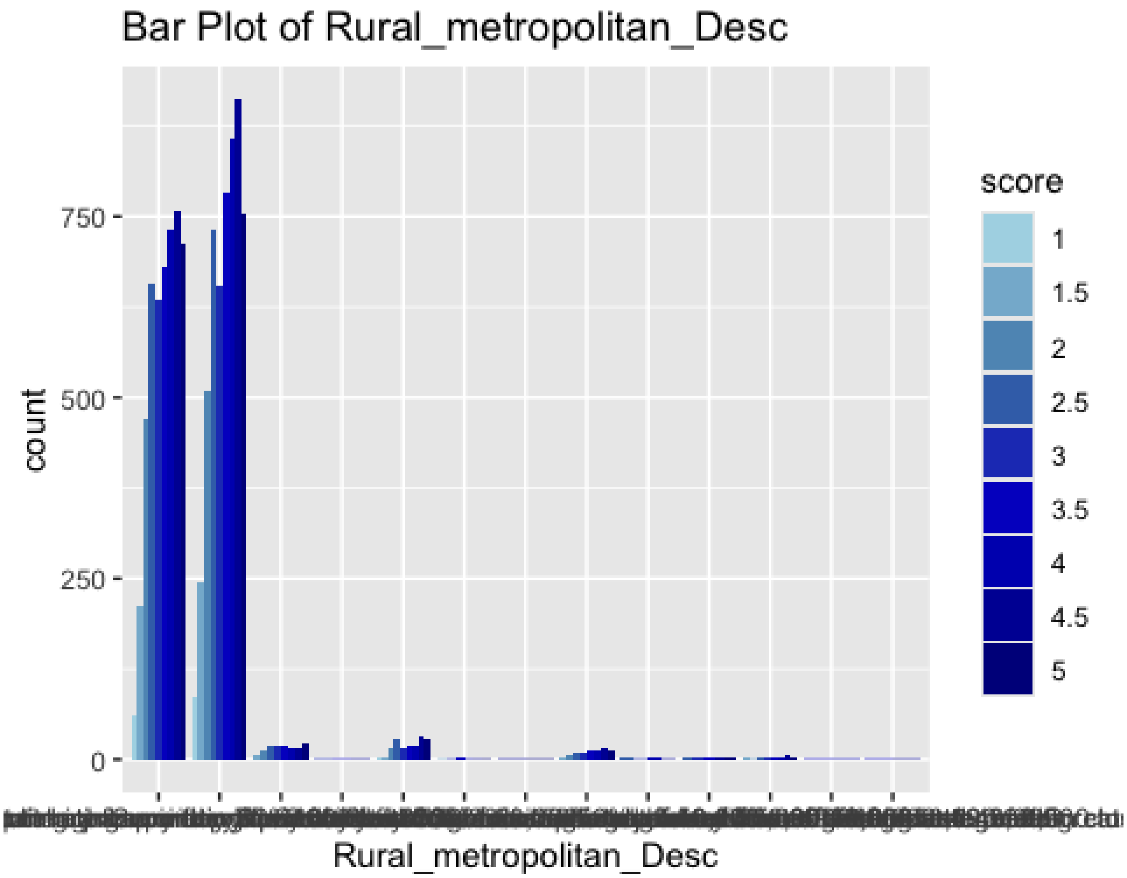
```
## [[1]]
```


Bar Plot of state
count
400
300
200
100
0
AB
AZ
CA
DE
FL
HI
ID
IL
IN
LA
MO
NJ
NV
PA
TN
state
score
1
1.5
2
2.5
3
3.5
4
4.5
5

```
##
## [[2]]
```


Bar Plot of Gender
count
900
600
300
0
F
M
Gender
score
1
1.5
2
2.5
3
3.5
4
4.5
5

```
##
## [[3]]
```



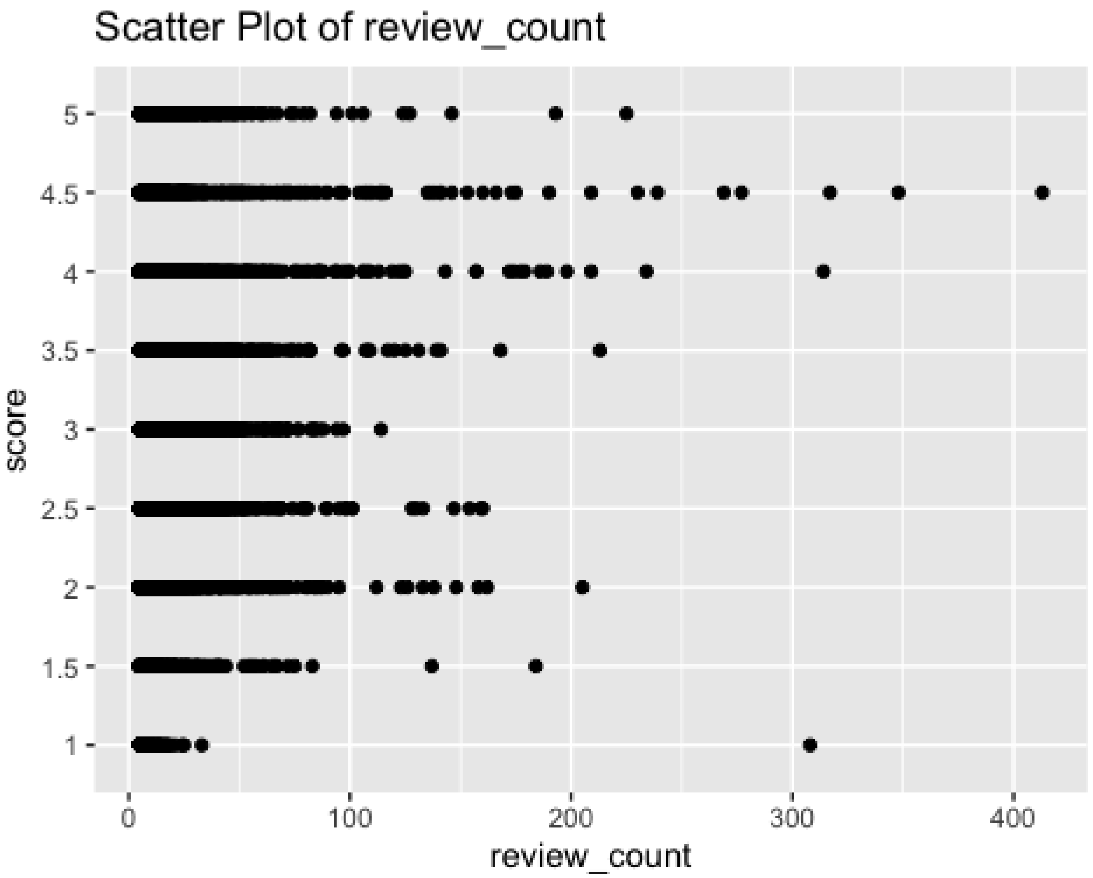
#2.4.2 Numeric variable

Review Count vs. Target Score: The data points are distributed over the range of review counts, reaching up to around 400. There is no observable association between the number of reviews and the target scores, as both low and high scores are evenly distributed.

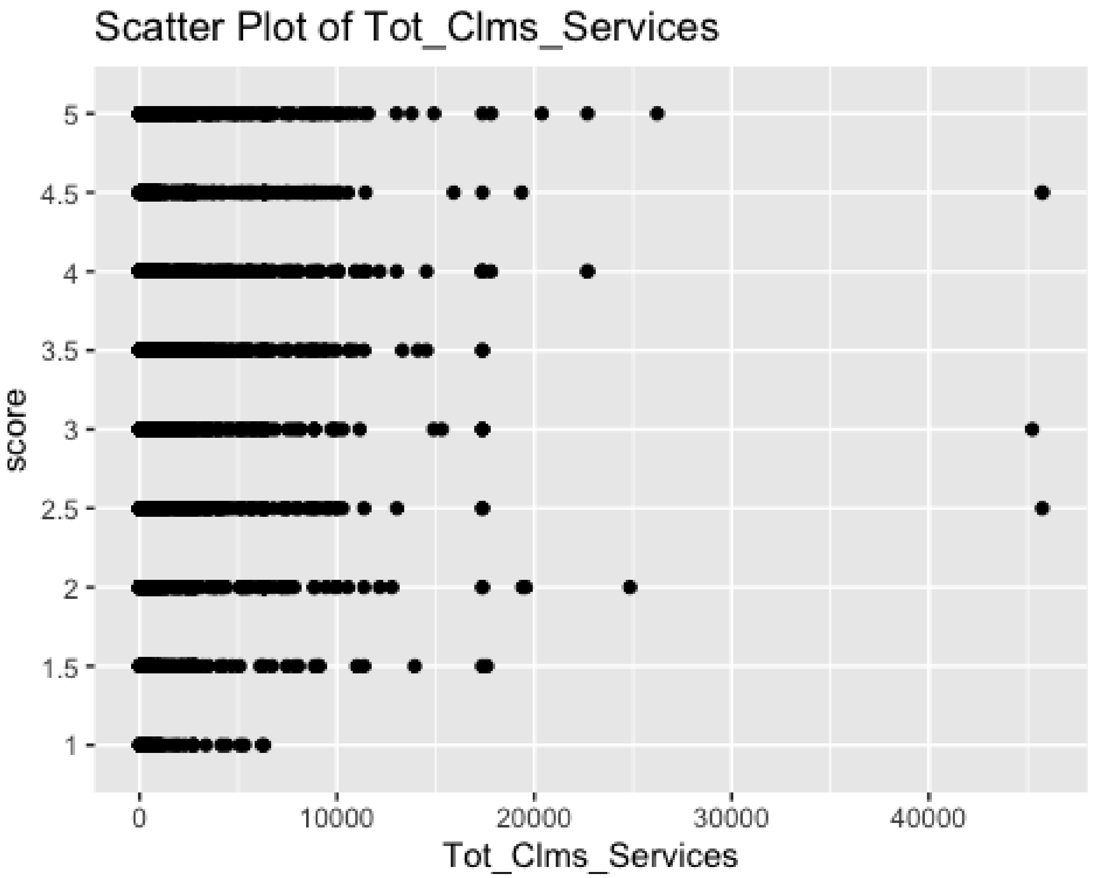
Total Claims Services vs. Target Score: Like the initial plot, this plot displays the total claims services spanning up to 40,000. The data points for both “low” and “high” scores are evenly distributed over all counts, suggesting the absence of any noticeable correlation.

Branded Total Claims Services vs. Target Score: The data points for branded total claims services extend up to approximately 7,500. There is no significant link between the distribution of “low” and “high” target scores across the whole range. Comparison of Generic Total Claims Services and Target Score: The generic total claims services cover a range of up to 40,000, which is comparable to the total claims services plot. The distribution of target scores categorised as “low” and “high” remains stable across various values, further indicating the absence of a clear association.

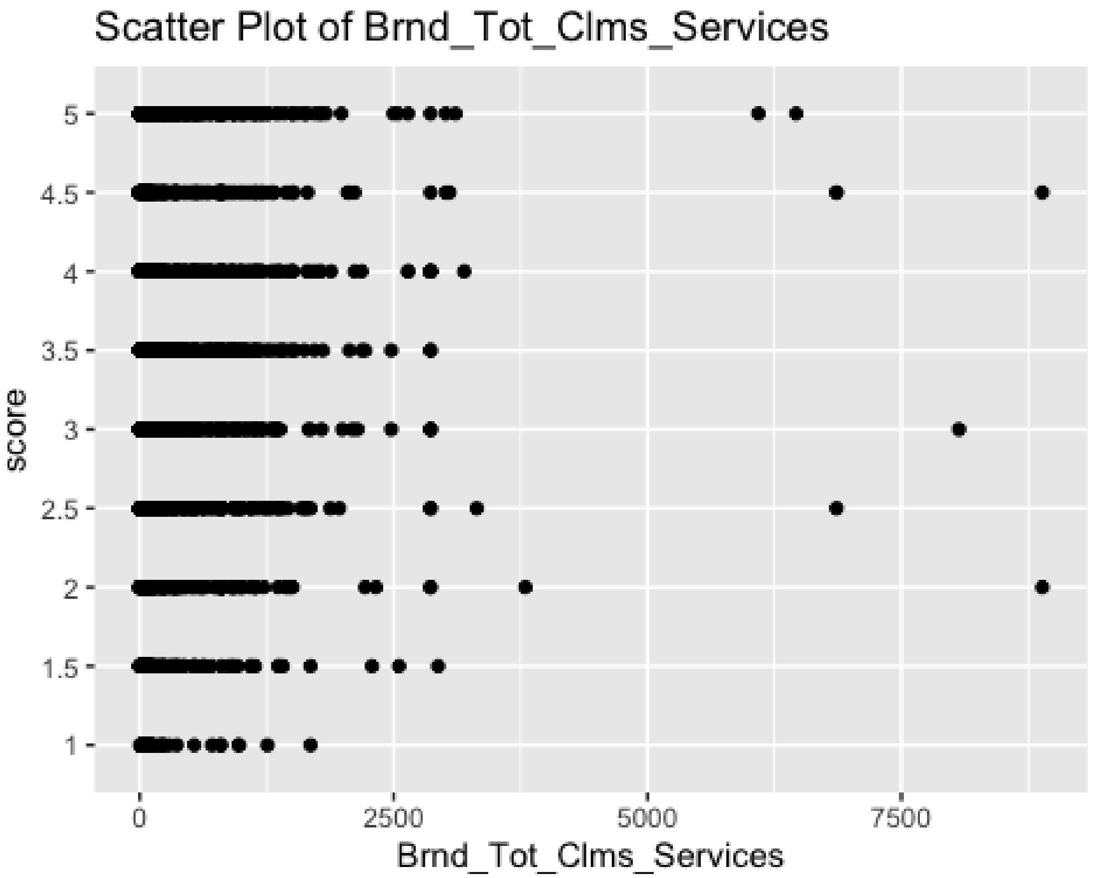
```
## [[1]]
```


Scatter Plot of review_count
score
5
4.5
4
3.5
3
2.5
2
1.5
1
0
100
200
300
400
review_count

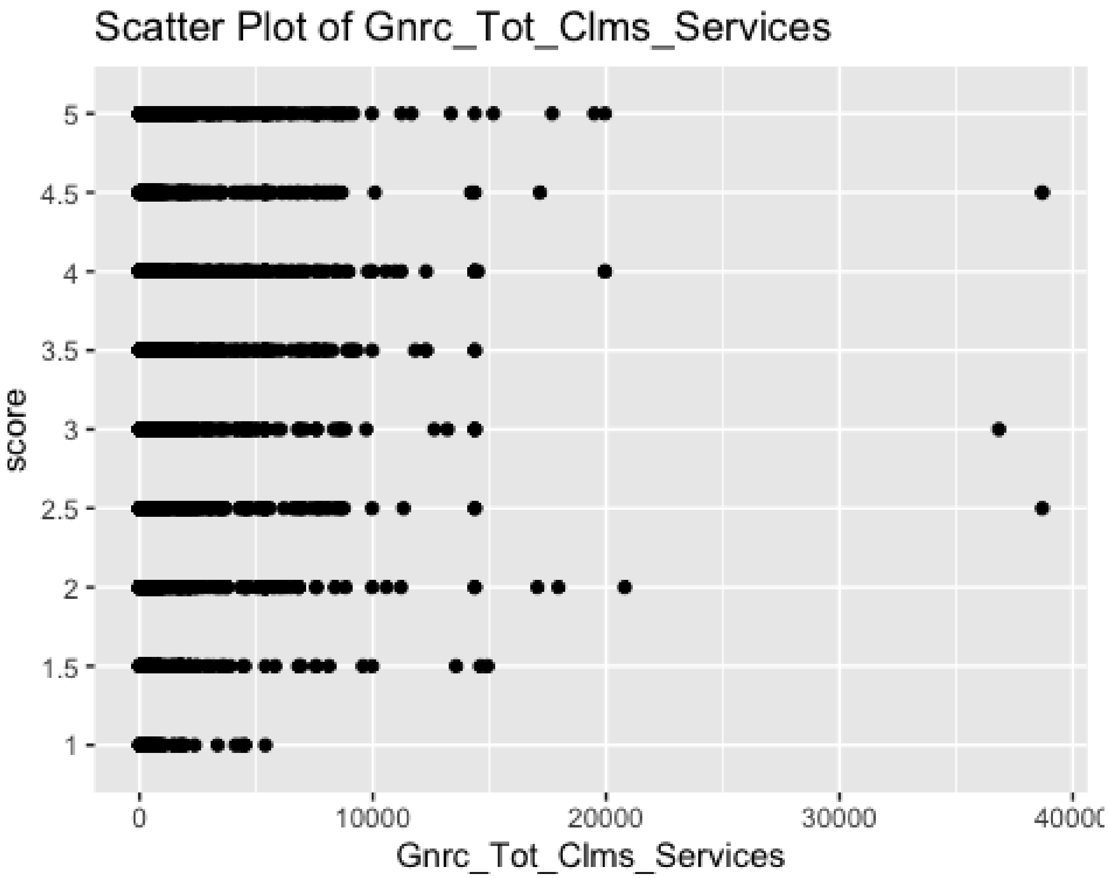
```
##
## [[2]]
```


Scatter Plot of Tot_Clms_Services
score
5
4.5
4
3.5
3
2.5
2
1.5
1
0
10000
20000
30000
40000
Tot_Clms_Services

```
##
## [[3]]
```


Scatter Plot of Brnd_Tot_Clms_Services
score
5
4.5
4
3.5
3
2.5
2
1.5
1
0
2500
5000
7500
Brnd_Tot_Clms_Services

```
##
## [[4]]
```



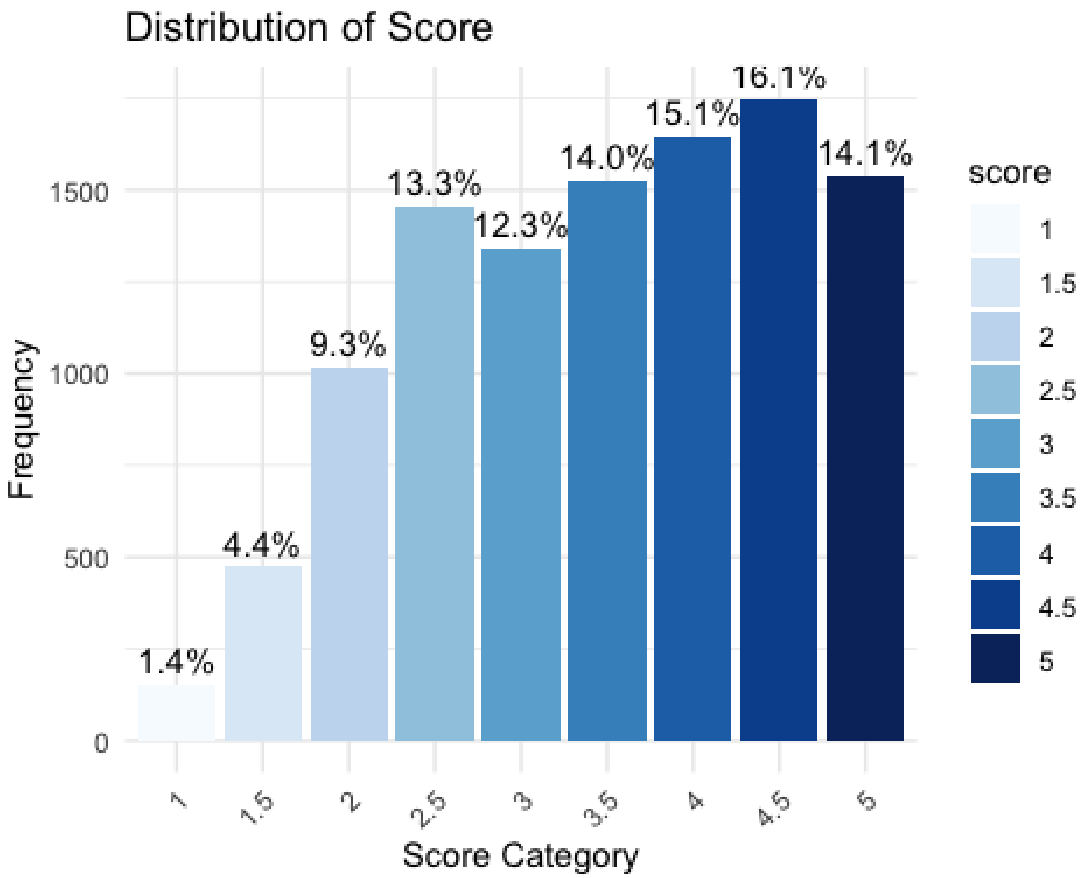
2.5 Creating a binary variable for score (target variable)

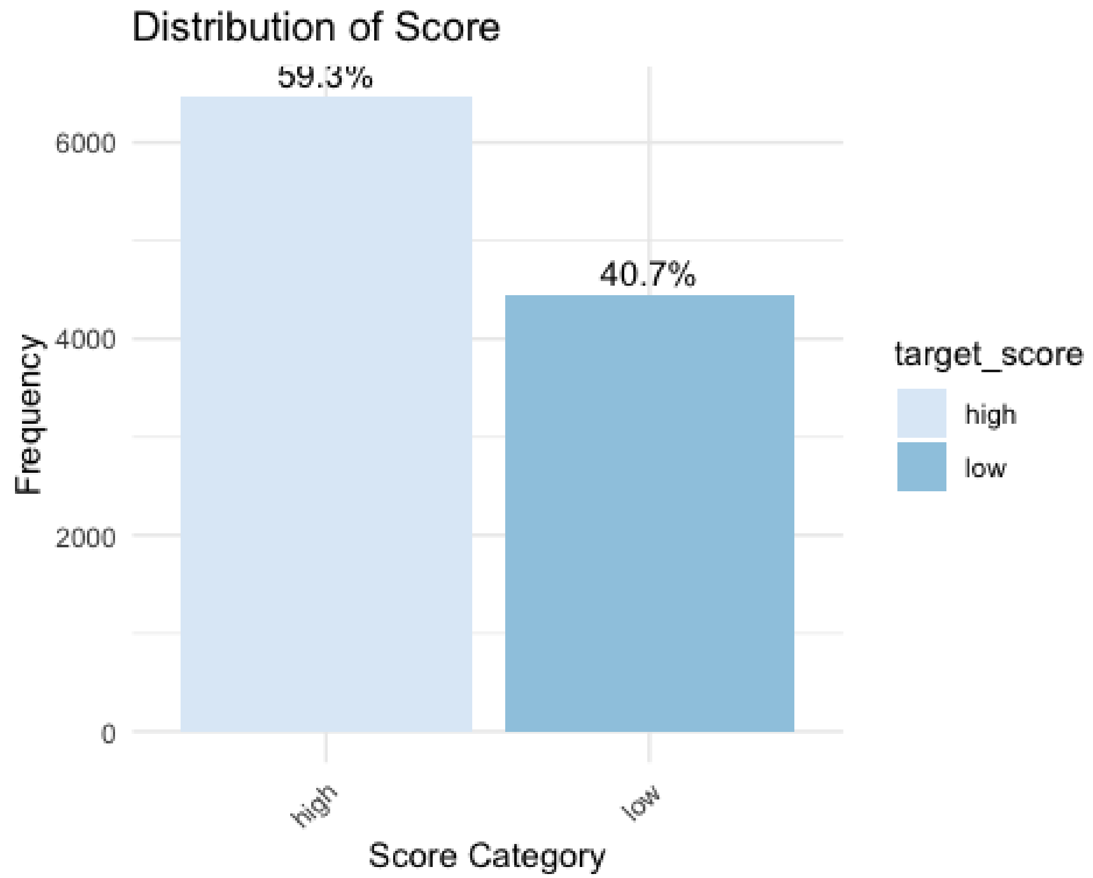
As part of the data preprocessing phase, an essential step was to create a binary variable based on the original score variable. This transformation was executed to optimise classification jobs by reducing the objective variable to two clearly defined categories: “low” and “high” scores.

Firstly, the `clean_data` dataframe was examined to calculate the frequencies and proportions for each score category. Afterward, a new variable called `target_score` was created using the original score variable. Scores that were 3 or lower were classified as “low”, and scores more than 3 were labelled as “high”.

A bar plot was created to visually depict the distribution of the binary target variable. It shows the frequency of each target category and includes percentage labels. For the purpose of making future analyses more efficient, the initial score variable was eliminated from the dataset, and the `target_score` variable was transformed into a categorical variable.

This transformation greatly improves the dataset’s suitability for classification tasks by simplifying the target variable, making it easier to create predictive models and conduct analysis.


Distribution of Score
Frequency
1500
1000
500
0
1.4%
4.4%
9.3%
13.3%
12.3%
14.0%
15.1%
16.1%
14.1%
1
1.5
2
2.5
3
3.5
4
4.5
5
Score Category
score
1
1.5
2
2.5
3
3.5
4
4.5
5



2.6 Subset Generation

The provided R code demonstrates the use of subsetting, a crucial preprocessing technique in data analysis workflows. Subsetting refers to the process of dividing a dataset into smaller, more manageable portions according to specific criteria. In this code, the dataset is efficiently divided into three separate subsets by using the distinct values from the category variable field_cat.

An important feature is the fair allocation of all distinct field_cat values among the subsets, guaranteeing that each subset contains a balanced representation of categories. The fair allocation of resources reduces the possibility of biases that may result from unequal representation of categories, hence promoting more comprehensive and dependable studies.

```
## Subset 1 has 4397 rows

## Subset 2 has 3235 rows

## Subset 3 has 3259 rows
```

#2.7 spliting the data into test and train

The process of splitting data into training and testing subsets is a crucial step in preparing datasets for machine learning analysis. By dividing the data into separate sets, we ensure that our models are trained on one portion and evaluated on another, thus enabling us to assess their performance accurately.

In this context, the data splitting technique employed here is stratified, meaning that it maintains the distribution of the target variable across the subsets. This approach helps to preserve the integrity of the data and prevents any biases that may arise from uneven distribution.

By stratifying the data based on the target variable, we ensure that both the training and testing sets represent all categories of the target variable proportionally. This ensures that our machine learning models are trained and evaluated on a representative sample of the data, leading to more reliable performance metrics.

#2.8 Preparing the data

The prepare_data function use the recipes package to create a recipe that encompasses a sequence of crucial data changes.

This recipe does a series of preprocessing activities, such as managing missing values, standardising numeric predictors, eliminating predictors with very low variance, and transforming categorical data into dummy variables. Significantly, every stage in the recipe is carefully crafted to improve the excellence and appropriateness of the dataset for future modelling assignments.

Imputation of Missing Values: The function employs k-nearest neighbors (KNN) imputation for numeric predictors and mode imputation for factor predictors to handle missing values. This ensures that missing data points are replaced with reasonable estimates based on the characteristics of neighboring observations.

Normalization of Numeric Predictors: Numeric predictors are standardized using normalization. This step is essential for ensuring that all numerical features are on a similar scale, preventing certain variables from dominating others due to differences in magnitude.

Removal of Near-Zero Variance Predictors: Predictors with near-zero variance, meaning they have little variability within the dataset, are removed. These predictors contribute minimal information to the modeling process and may introduce noise without providing meaningful insights.

Conversion of Factor Predictors to Dummy Variables: Factor predictors are converted into dummy variables using one-hot encoding. This transformation is necessary for incorporating categorical variables into machine learning models, as most algorithms require numerical inputs. One-hot encoding creates binary (0/1) indicator variables for each category within a categorical variable.

By organising the pretreatment procedures within a recipe, the code enhances modularity, reproducibility, and scalability in data preparation. Furthermore, the recipe incorporates domain expertise and optimal methods, ensuring that the preprocessing pipeline is clear and user-friendly.

#3.0 Model Building

#3.1 Lasso Regression

Lasso regression, a type of linear regression, is used for binary classification problems because it can handle high-dimensional data and conduct feature selection (Hastie et al., 2009). This method imposes a penalty on the magnitude of the regression coefficients, promoting solutions with fewer non-zero coefficients, so achieving variable selection. An important benefit of Lasso regression is its capacity to address multicollinearity among predictors, making it well-suited for datasets containing correlated features (Friedman et al., 2010).

Multiple studies have investigated the use of Lasso regression in binary classification tasks, such as "The Elements of Statistical Learning" by Trevor Hastie, Robert Tibshirani, and Jerome Friedman (Hastie et al., 2009), and "Regularisation Paths for Generalised Linear Models via Coordinate Descent" by Jerome Friedman, Trevor Hastie, and Rob Tibshirani (Friedman et al., 2010). These studies offer in-depth understanding of the theoretical underpinnings and real-world uses of Lasso regression in classification scenarios.

The lasso_analysis function begins by loading necessary libraries, including glmnet for Lasso regression, dplyr for data manipulation, ggplot2 for plotting, and pROC for ROC analysis. It prepares the predictor matrix x and the target variable y from the input data. Next, the function fits a Lasso model using cross-validation (Hastie et al., 2009) to automatically select the optimal value of the regularization parameter lambda.

The coefficients corresponding to the lambda value that minimizes the mean cross-validated error are extracted from the fitted model. These coefficients are then sorted based on their absolute values to identify the top predictors. The function prints and plots the top 10 predictors, showcasing their coefficients and importance in the classification task.

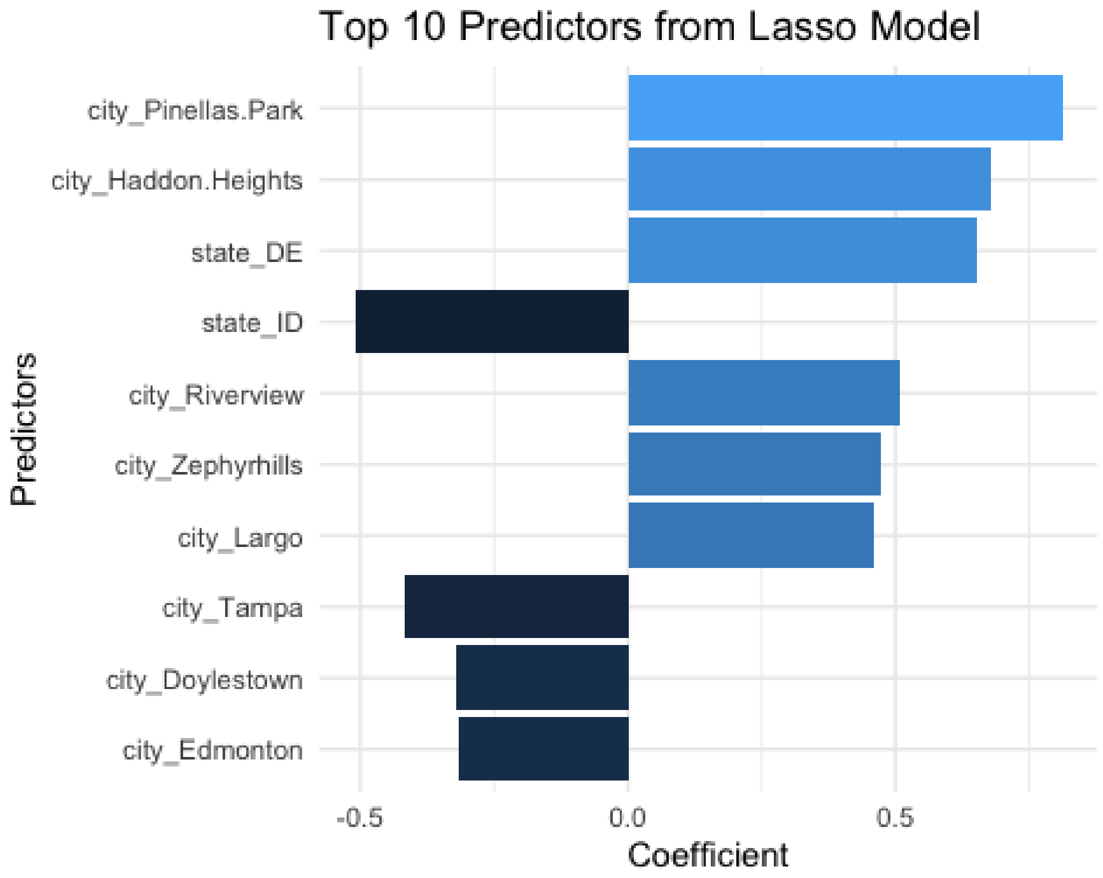
Subsequently, the function predicts probabilities for the ROC curve using the fitted Lasso model and calculates the area under the ROC curve (AUC) as a performance metric. Finally, it plots the ROC curve, visualizing the trade-off between sensitivity and specificity, with the AUC value included in the plot title.

#3.1.1 Lasso on Subset one

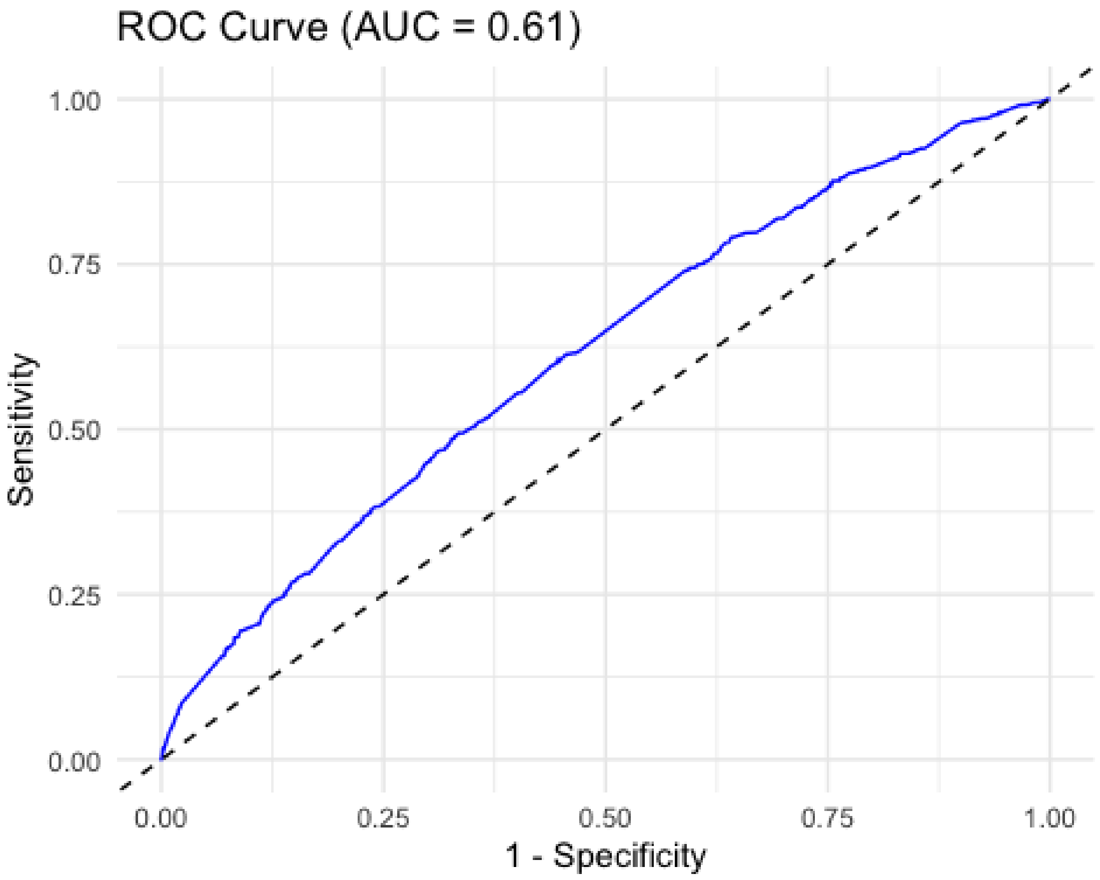
Subset one was analysed using Lasso regression, and the model discovered multiple crucial factors that had a significant impact on the target variable. The Lasso regression analysis reveals that the coefficients for city_Pinellas.Park and city_Haddon.Heights are the most significant positive predictors, with coefficients of around 1.1124 and 1.0983, respectively. This indicates a significant favourable impact of these sites on the probability of the result. On the other hand, the city of Doylestown is shown to have a strong negative impact on the target variable, with a value of -0.8512, showing a reverse correlation. These insights are essential for comprehending the regional effects on the dataset's results and can direct more targeted analysis or strategic choices.

The ROC curve analysis enhances the coefficient analysis by offering a visual and quantitative assessment of the model's predictive performance. The model exhibits moderate predictive capability, as evidenced by an AUC of 0.65. The current level of performance indicates that the model is capable of partially differentiating across classes, although there is room for enhancement. The ROC curve, with its gradual incline towards the upper-right corner of the graph, provides additional evidence by demonstrating a trade-off between sensitivity and specificity at different threshold values.

```
##               variable coefficient abs_coefficient
## 1    city_Pinellas.Park   0.8112834       0.8112834
## 2   city_Haddon.Heights   0.6764976       0.6764976
## 3              state_DE   0.6506418       0.6506418
## 4              state_ID  -0.5105722       0.5105722
## 5        city_Riverview   0.5086527       0.5086527
## 6      city_Zephyrhills   0.4710833       0.4710833
## 7            city_Largo   0.4577751       0.4577751
## 8            city_Tampa  -0.4182499       0.4182499
## 9       city_Doylestown  -0.3205093       0.3205093
## 10        city_Edmonton  -0.3177839       0.3177839
```



```
## Area Under the Curve (AUC): 0.6142373
```

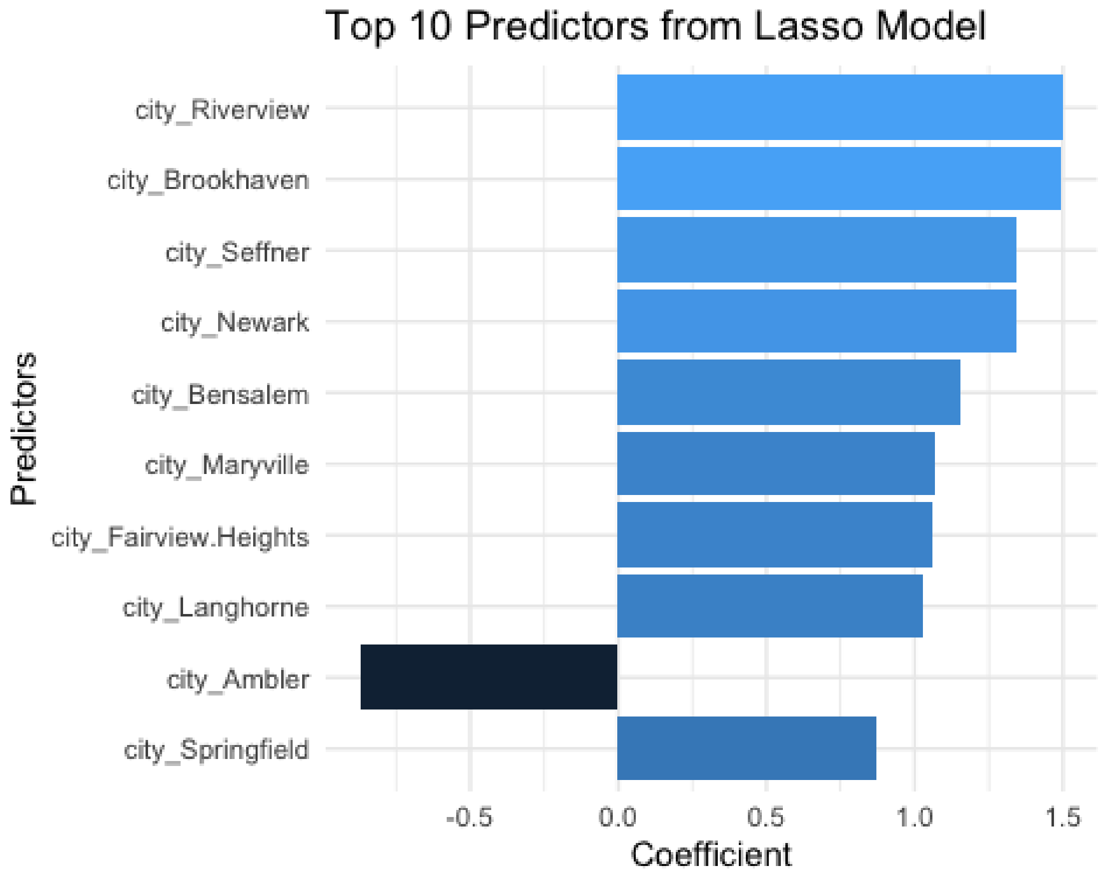


#3.2.2 Lasso on Subset two

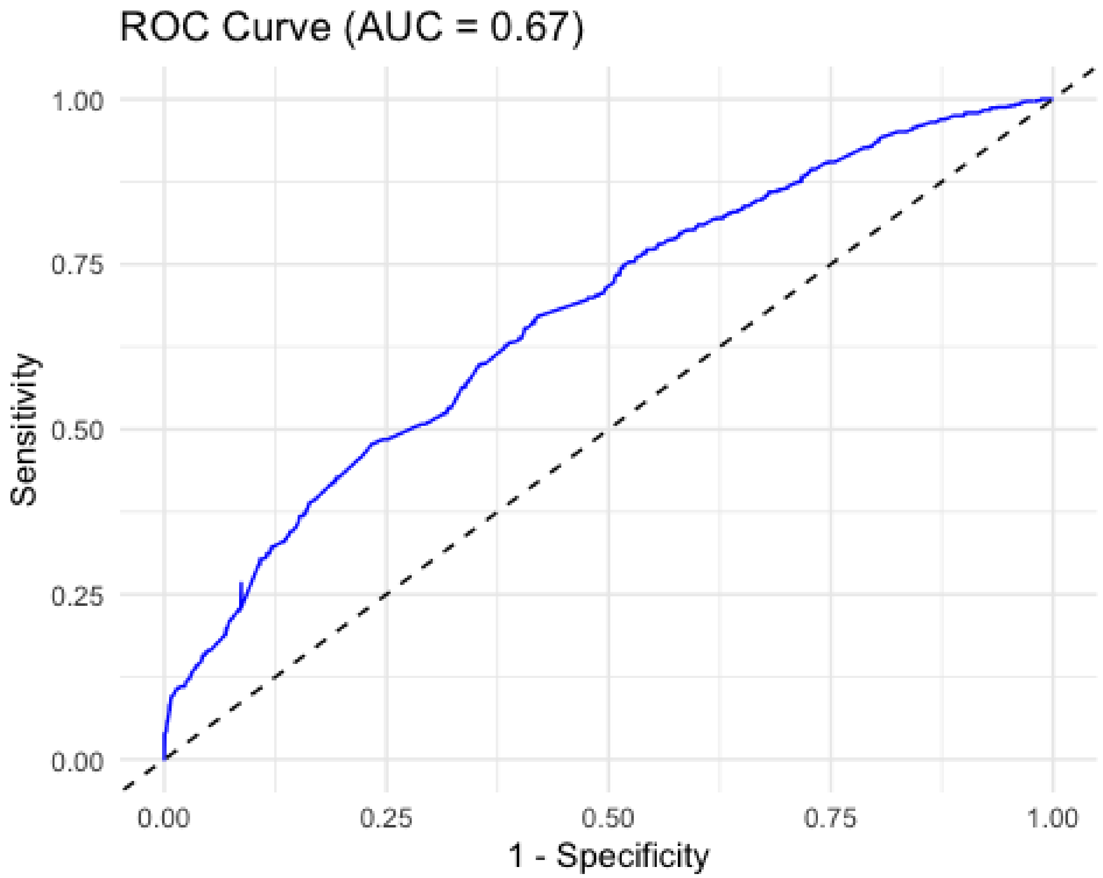
The study of Subset Two, conducted using Lasso regression, identifies specific geographic regions that have a substantial impact on predicting the target score. The cities "city_Riverview" and "city_Brookhaven" have the greatest positive coefficients on the list, specifically 1.4984295 and 1.4932536 respectively. This indicates a significant correlation with higher scores. This implies that there may be certain aspects present in these regions that are favourable for getting better scores. Therefore, it is necessary to conduct further research on the specific qualities or policies of these localities that might have an impact on these outcomes.

In contrast, the city "Ambler" exhibited a significant negative coefficient of -0.8702901, indicating that individuals from this city are less likely to achieve a high score compared to the baseline. The receiver operating characteristic (ROC) curve for this model obtained an area under the curve (AUC) of 0.67, indicating a moderate capacity of the model to differentiate between the various score classes.

```
##                  variable coefficient abs_coefficient
## 1          city_Riverview   1.4984295       1.4984295
## 2         city_Brookhaven   1.4932536       1.4932536
## 3            city_Seffner   1.3459541       1.3459541
## 4             city_Newark   1.3412240       1.3412240
## 5           city_Bensalem   1.1542930       1.1542930
## 6          city_Maryville   1.0655274       1.0655274
## 7  city_Fairview.Heights   1.0581123       1.0581123
## 8          city_Langhorne   1.0273121       1.0273121
## 9             city_Ambler  -0.8702901       0.8702901
## 10       city_Springfield   0.8679344       0.8679344
```



```
## Area Under the Curve (AUC): 0.672875
```

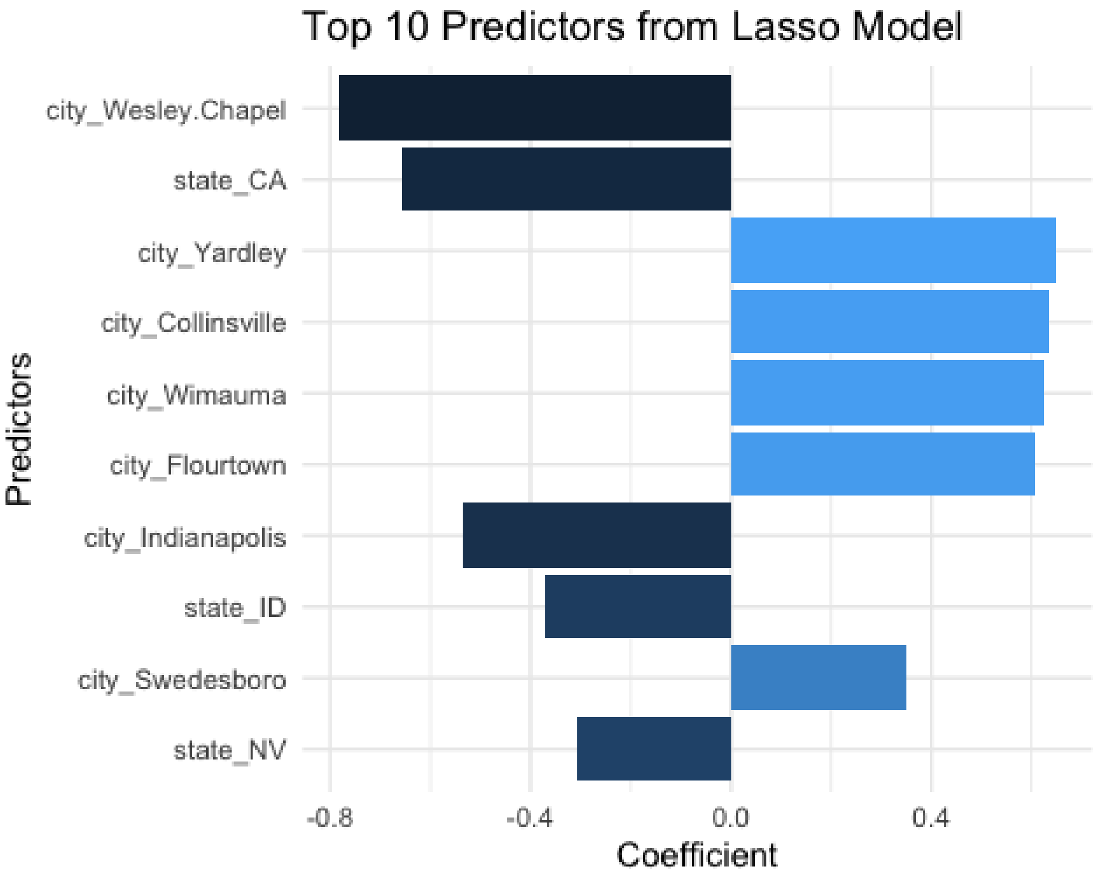


#3.2.3 Lasso on Subset Three

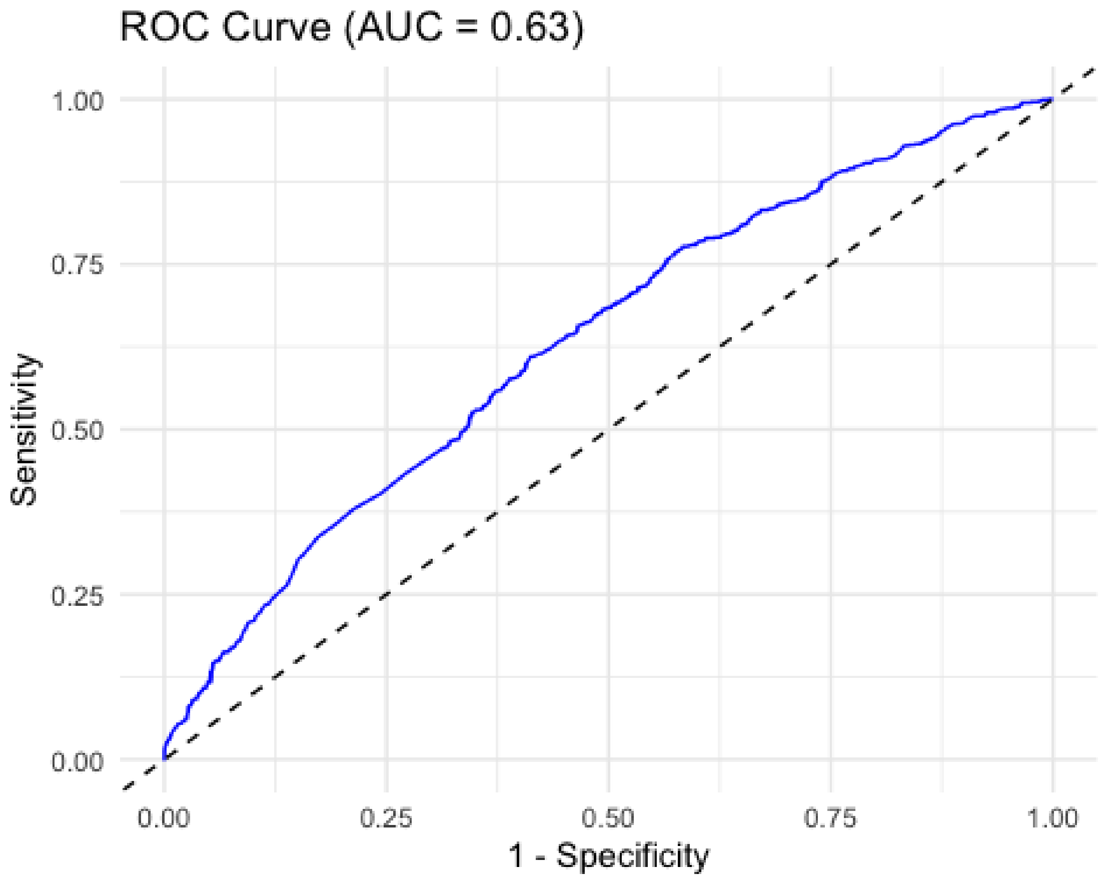
Subset Three demonstrates that the Lasso regression analysis identifies a notable influence of both city and state variables on the target score. The negative coefficients of -0.7840290 and -0.6561735 for “city_Wesley.Chapel” and “state_CA” correspondingly indicate that these areas are unlikely to have higher scores. This may suggest underlying difficulties or distinct circumstances that could be resolved through focused educational or developmental interventions.

In contrast, the presence of positive coefficients for “city_Yardley” and “city_Collinsville” indicates that these regions are more conducive to earning higher ratings. The coexistence of favourable and unfavourable predictors highlights the wide range of score results across different geographical locations in the sample. The receiver operating characteristic (ROC) curve for this particular subset yielded an area under the curve (AUC) value of 0.63, suggesting that the model has a moderate level of discriminatory capacity. The current level of performance indicates that the model is capable of accurately predicting outcomes.

```
##              variable coefficient abs_coefficient
## 1  city_Wesley.Chapel  -0.7840290       0.7840290
## 2            state_CA  -0.6561735       0.6561735
## 3        city_Yardley   0.6496767       0.6496767
## 4   city_Collinsville   0.6324693       0.6324693
## 5        city_Wimauma   0.6256600       0.6256600
## 6      city_Flourtown   0.6077593       0.6077593
## 7   city_Indianapolis  -0.5333636       0.5333636
## 8            state_ID  -0.3742216       0.3742216
## 9     city_Swedesboro   0.3508762       0.3508762
## 10           state_NV  -0.3048069       0.3048069
```



```
## Area Under the Curve (AUC): 0.6326086
```



#3.2.4 Comparasion lasso on all three subsets

In comparing the Lasso regression models across the three subsets, several key differences and similarities emerge that highlight the varied influence of geographic predictors on the target scores. The first subset demonstrated a relatively stronger positive influence with top predictors like “city_Pinellas.Park” and “city_Haddon.Heights,” achieving an AUC of 0.65. The second subset, while showing robust positive influences from “city_Riverview” and “city_Brookhaven,” also revealed a notable negative predictor “city_Ambler,” with a slightly better AUC at 0.67. The third subset, however, showcased a mix of negative and positive coefficients with prominent locations like “city_Wesley.Chapel” and “city_Yardley,” but with a slightly lower AUC of 0.63

#3.3 GAM Model

Generalised Additive Models (GAMs) offer a complex approach to analysis, allowing for the modelling of non-linear and non-parametric connections between predictors and the response variable. This approach enables a more precise depiction of data complexities in contrast to conventional linear models, which make the assumption of a linear relationship between variables. Generalised Additive Models (GAMs) employ smooth functions, such splines, to individually modify each predictor, allowing for the capture of complex patterns without a predetermined non-linear shape (Hastie & Tibshirani, 1990; Wood, 2017). These models are highly valuable in domains such as environmental research, where they simulate the impact of contaminants on health outcomes, and in marketing, where they evaluate the influence of pricing on sales dynamics (Dominici et al., 2002; Härdle et al., 2004).

GAMs possess a notable advantage in terms of interpretability, since they offer distinct insights comparable to linear models while also allowing for greater adaptability in handling non-linear connections. This capability is essential for making decisions based on data in fields that necessitate a thorough comprehension of the effects of variables, such as public health and business strategy. The availability of advanced tools such as the mgcv package in R simplifies the use of GAMs, enabling researchers and analysts from other fields to easily apply them in their work.

The run_gam_model function is intricately crafted to optimise the process of fitting a Generalised Additive Model (GAM) to our data. The function takes three main arguments: the dataset, a formula string, and the distribution family, with "gaussian" being the default for continuous response variables.

The fundamental aspect of this function revolves around the gam function from the mgcv package, which builds the model using the provided formula and data. This formula incorporates differentiable functions that automatically ascertain the configuration of the connection between each predictor and the answer, offering a versatile method to represent non-linearities without a predetermined structure. The tryCatch block guarantees resilient error handling, recording and reporting any problems that occur during the model fitting process, hence improving the dependability of the model implementation.

After a successful fitting, the function generates a summary of the model that provides in-depth information on the impact of each predictor. This includes predicted coefficients and statistical significances. This summary facilitates the interpretation of the model results, ensuring that we acquire a full understanding of how predictors impact the outcome. In the event that the model experiences an error, the function offers a detailed error message, enabling the user to identify and resolve the issue. This function demonstrates a powerful analytical tool specifically designed for sophisticated statistical modelling in several study fields.

The data and formula string have been determined based on the output of lasso regression for all three subsets. These datasets and models will be utilised in both GAM and Decision Tree.

#3.3.1 GAM on subset one

The findings of the Generalised Additive Model (GAM) study on Subset 1, utilising a binomial family with a logit link function, offer valuable insights into the impact of different predictor variables on the target score. The formula of the model included multiple predictors, such as variables for city and state. City_Haddon.Heights and city_Zephyrhills had notably high coefficients, indicating a robust positive correlation with the target score. However, the statistical significance of these predictors was not established (p-value > 0.05). However, predictors such as city_Doyelstown and state_DE exhibited noteworthy negative correlations with the target score. This is evident from their negative coefficients and extremely low p-values (Pr(|z|) < 0.001 for state_DE and 0.951902 for city_Doyelstown), which emphasise their significant impact in decreasing the probability of achieving a higher target score.

The model's R-squared value is relatively low, with an adjusted R-squared of 0.0265. This suggests that the model only accounts for a small portion of the variability in the target score. This implies that although the model recognises important characteristics that help predict outcomes, there may be additional factors that have not been included in the model that also have a major impact. The model accounts for 2.37% of the deviation, suggesting that incorporating additional variables or more intricate interactions may be required to comprehensively capture the factors influencing the target score.

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## target_score ~ city_Pinellas.Park + city_Haddon.Heights + state_DE +
##     state_ID + city_Riverview + city_Zephyrhills + city_Largo +
##     city_Tampa + city_Doylestown + city_Edmonton
##
## Parametric coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -0.38169    0.03874  -9.852  < 2e-16 ***
## city_Pinellas.Park    1.84803    0.64168   2.880 0.003977 **
## city_Haddon.Heights  13.94776  267.70559   0.052 0.958448
## state_DE              1.15488    0.28757   4.016 5.92e-05 ***
## state_ID             -1.09991    0.35246  -3.121 0.001804 **
## city_Riverview        1.11929    0.36863   3.036 0.002395 **
## city_Zephyrhills     13.94776  267.70559   0.052 0.958448
## city_Largo            1.39329    0.58516   2.381 0.017263 *
## city_Tampa           -0.79103    0.22774  -3.473 0.000514 ***
## city_Doylestown     -13.18437  218.58070  -0.060 0.951902
## city_Edmonton        -1.56422    0.75692  -2.067 0.038776 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =  0.0265   Deviance explained = 2.37%
## -REML = 2013.8  Scale est. = 1         n = 3079
```

#3.3.2 GAM on subset two

The model utilises a binomial family and a logit link function to forecast the 'target_score' by considering the existence of different city factors. The intercept, indicated as highly negative (-0.43153, p < 2e-16), implies that the initial likelihood of reaching the desired score is low when no other factors are included. The coefficients for cities such as Riverview, Brookhaven, Sefner, Newark, and others have very high estimates (e.g., 14.99760), yet they are linked to enormous standard errors and p-values near to 1. This suggests that these cities do not have a substantial impact on the target score. Nevertheless, statistical analysis reveals that Langhorne and Springfield exhibit substantial (p < 0.05) positive and negative coefficients, respectively, suggesting distinct effects on the target score.

```
## 
## Family: binomial 
## Link function: logit 
## 
## Formula:
## target_score ~ city_Riverview + city_Brookhaven + city_Seffner + 
##     city_Seffner + city_Newark + city_Bensalem + city_Maryville + 
##     city_Fairview.Heights + city_Langhorne + city_Ambler + 
city_Springfield
## 
## Parametric coefficients:
##                        Estimate Std. Error z value Pr(>|z|)    
## (Intercept)            -0.43153    0.04352  -9.916   <2e-16 ***
## city_Riverview         14.99760  394.77484   0.038   0.9697    
## city_Brookhaven        14.99760  394.77484   0.038   0.9697    
## city_Seffner           14.99760  441.37169   0.034   0.9729    
## city_Newark            14.99760  441.37169   0.034   0.9729    
## city_Bensalem          14.99760  509.65213   0.029   0.9765    
## city_Maryville         14.99760  509.65213   0.029   0.9765    
## city_Fairview.Heights  14.99760  509.65213   0.029   0.9765    
## city_Langhorne          2.22329    1.08100   2.057   0.0397 *  
## city_Ambler           -14.13454  360.37848  -0.039   0.9687    
## city_Springfield        1.63550    0.65972   2.479   0.0132 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## 
## R-sq.(adj) =  0.0213   Deviance explained = 2.27%
## -REML = 1437.5  Scale est. = 1         n = 2265
```

#3.3.3 GAM on subset three

The intercept has a statistically significant negative value of -0.2206 (p < 0.001), indicating a low initial probability of reaching a positive goal score without considering the predictors. The parameters for most cities and states, such as Wesley Chapel, Flourtown, and Wimauma, have p-values that are not significant. This suggests that there is no strong evidence to indicate that these characteristics have a distinct influence on the target score. The presence of negative coefficients for states such as California (p < 0.01) and Idaho (p < 0.01), and a positive coefficient for Swedesboro (p < 0.05), indicates that there is geographical variation in the potential to achieve positive outcomes.

```
## 
## Family: binomial 
## Link function: logit 
```

```
##
## Formula:
## target_score ~ city_Wesley.Chapel + state_CA + city_Yardley +
##     city_Collinsville + city_Wimauma + city_Flourtown + city_Indianapolis
+
##     state_ID + city_Swedesboro + state_NV
##
## Parametric coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -0.2206     0.0480  -4.595 4.33e-06 ***
## city_Wesley.Chapel -13.3455   189.2964  -0.071 0.943795
## state_CA            -1.3199     0.3704  -3.563 0.000367 ***
## city_Yardley         2.0123     1.0812   1.861 0.062712 .
## city_Collinsville   13.7866   309.1198   0.045 0.964426
## city_Wimauma        13.7866   309.1198   0.045 0.964426
## city_Flourtown      13.7866   309.1198   0.045 0.964426
## city_Indianapolis   -1.1404     0.3577  -3.188 0.001432 **
## state_ID            -0.7285     0.2021  -3.605 0.000312 ***
## city_Swedesboro     13.7866   309.1198   0.045 0.964426
## state_NV            -0.5400     0.1392  -3.880 0.000104 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =  0.029   Deviance explained = 2.78%
## -REML = 1474.4  Scale est. = 1         n = 2283
```

#3.4 Decision Tree

Decision Trees are a popular non-linear prediction model that is commonly employed for classification tasks because of their intuitive representation and straightforward implementation. A Decision Tree algorithm divides the data into more and more similar groups by using a set of decision rules that are created from the qualities of the data (James et al., 2013). This strategy has demonstrated significant efficacy in situations when the connection between the independent variables and the dependent variable is intricate and not linear.

A Decision Tree is composed of nodes, which represent the data attributes, branches, which correspond to the decision rules, and leaf nodes, which indicate the outcome of these decisions. Decision Trees have a significant advantage in their capacity to provide a clear and visual representation of the decision-making process, making it highly beneficial for understanding the model's rationale. A Decision Tree is capable of effectively handling both numerical and categorical data, which makes it a flexible tool in the classifier's repertoire.

Multiple studies have emphasised the effectiveness of Decision Trees in diverse domains, such as medical, finance, and social sciences. An important application of Decision Trees may be observed in the healthcare industry, where they are utilised to categorise patient outcomes by analysing their symptoms and medical history (Kumar & Indrayan, 2011). The capacity to categorise intricate datasets with a notable level of precision renders Decision Trees an essential instrument in data analysis.

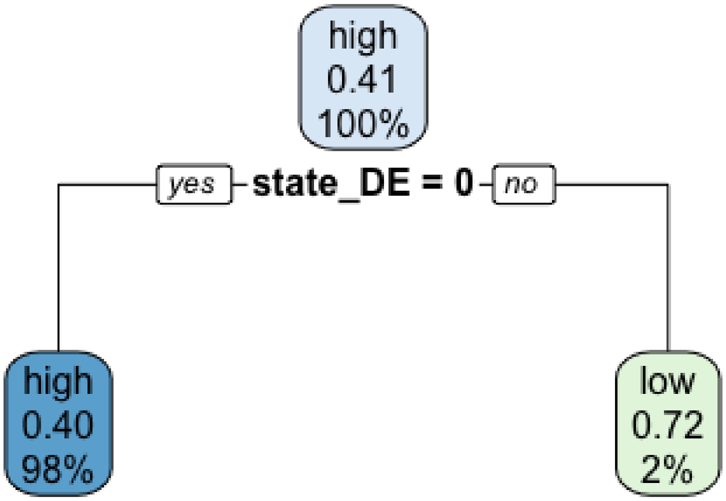
The runDecisionTree function in R is specifically designed to train and assess a Decision Tree model using the rpart package, which is often used for classification tasks. The process starts by establishing a replicable seed to ensure consistent findings. Next, the dataset is divided into training and testing sets using a defined split_ratio. The function subsequently implements a training control mechanism by employing cross-validation (cv) with 10 folds to address the issue of overfitting and enhance the model's capacity to generalise. Additionally, it establishes a framework of complexity parameters (cp) to fine-tune the decision tree, aiding in identifying the ideal threshold to halt tree expansion and prevent overfitting. The train function from the caret package employs this grid to identify the optimal model, which is then visualised using rpart.plot. Subsequently, the testing data is utilised to generate predictions, and the model's effectiveness is evaluated by employing a confusion matrix, which yields metrics such as accuracy, precision, and recall. The function provides the model, its graphical representation, and the evaluation metrics, encompassing the complete process of training, optimising, and evaluating the model in a methodical and replicable way.

#3.4.1 Decision Tree on subset 1

For this investigation, we utilised the Classification and Regression Trees (CART) technique on a dataset consisting of 2157 samples. The dataset included 10 predictor features and two distinct classes: "high" and "low". The model's performance was evaluated and the hyperparameters were tuned using cross-validation with 10 folds.

The decision tree model was trained using different values of the complexity parameter (cp), which regulates the pruning of the tree to avoid overfitting. The table displays the results for various cp values, indicating the accuracy and kappa statistic for each value.The model with the highest accuracy, attained with a cp value of 0.01, was picked as the best model. It had an accuracy of 0.5943475 and a kappa value of 0.007398725. The graphic shown illustrates the decision tree model that was generated.The tree is composed of two leaf nodes, which symbolise the "high" and "low" classes. The probability of the "high" class in the left node is 0.40, and 98% of the data points belong to this node. The probability of the "low" class in the right node is 0.72, representing 2% of the data points.

The decision tree model offers a clear and understandable depiction of the decision boundaries used to categorise samples into the "high" and "low" classes, based on the predictive features. The precision of 0.5943475 suggests that the model accurately identified around 59% of the items in the dataset.It is worth mentioning that although the accuracy metric gives a general indication of the model's performance, the kappa statistic, which considers the agreement between the predicted and true classes, is quite low at 0.007398725.



```
## CART
##
## 2157 samples
##   10 predictor
##    2 classes: 'high', 'low'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1942, 1941, 1941, 1942, 1941, 1941, ...
## Resampling results across tuning parameters:
##
##   cp    Accuracy   Kappa
##   0.01  0.5943475  0.007398725
##   0.02  0.5929543  0.002507672
##   0.03  0.5929543  0.000000000
##   0.04  0.5929543  0.000000000
##   0.05  0.5929543  0.000000000
##   0.06  0.5929543  0.000000000
##   0.07  0.5929543  0.000000000
##   0.08  0.5929543  0.000000000
##   0.09  0.5929543  0.000000000
##   0.10  0.5929543  0.000000000
##
```

```
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.01.
```

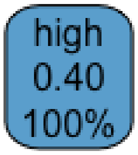
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction high low
##       high  539 362
##       low     8  13
##
##                Accuracy : 0.5987
##                  95% CI : (0.5662, 0.6305)
##     No Information Rate : 0.5933
##     P-Value [Acc > NIR] : 0.3822
##
##                   Kappa : 0.0235
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.98537
##             Specificity : 0.03467
##          Pos Pred Value : 0.59822
##          Neg Pred Value : 0.61905
##              Prevalence : 0.59328
##          Detection Rate : 0.58460
##    Detection Prevalence : 0.97722
##       Balanced Accuracy : 0.51002
##
##        'Positive' Class : high
##
```

#3.4.2 Decision Tree on subset 2

For subset 2, we utilised the Classification and Regression Trees (CART) technique to analyse a dataset consisting of 1586 samples, 10 predictor features, and two distinct classes: “high” and “low”. The model’s performance was evaluated and the hyperparameters were tuned using cross-validation with 10 folds.

The decision tree model was trained using different values of the complexity parameter (cp), which regulates the pruning of the tree to avoid overfitting. The table displays the results for various cp values, indicating the accuracy and kappa statistic associated with each value.The optimum model was chosen based on the maximum accuracy, which stayed consistent at 0.5964652 for all values of cp. Notably, the kappa statistic was consistently 0 for all models, regardless of the cp value.

The image shown displays the decision tree model that was generated. The tree is composed of only one leaf node, which classifies all instances into the “high” class with a probability of 0.40, covering 100% of the data points. The accuracy of 0.5964652 implies that the model accurately identified around 59.6% of the samples.



```
## CART
##
## 1586 samples
##   10 predictor
##    2 classes: 'high', 'low'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1428, 1428, 1427, 1427, 1428, 1428, ...
## Resampling results across tuning parameters:
##
##   cp    Accuracy   Kappa
##   0.01  0.5964652  0
##   0.02  0.5964652  0
##   0.03  0.5964652  0
##   0.04  0.5964652  0
##   0.05  0.5964652  0
##   0.06  0.5964652  0
##   0.07  0.5964652  0
##   0.08  0.5964652  0
##   0.09  0.5964652  0
##   0.10  0.5964652  0
##
```

```
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.1.

## Confusion Matrix and Statistics
## 
##           Reference
## Prediction high low
##       high  405 274
##       low     0   0
##                                           
##                Accuracy : 0.5965          
##                  95% CI : (0.5585, 0.6336)
##     No Information Rate : 0.5965          
##     P-Value [Acc > NIR] : 0.5166          
##                                           
##                   Kappa : 0               
##                                           
##  Mcnemar's Test P-Value : <2e-16          
##                                           
##             Sensitivity : 1.0000          
##             Specificity : 0.0000          
##          Pos Pred Value : 0.5965          
##          Neg Pred Value :    NaN          
##              Prevalence : 0.5965          
##          Detection Rate : 0.5965          
##    Detection Prevalence : 1.0000          
##       Balanced Accuracy : 0.5000          
##                                           
##        'Positive' Class : high            
## 
```

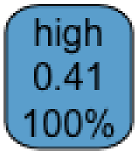
#3.4.3 Decision Tree on subset 3

The CART technique was used to analyse subset 3 of the dataset. This subset contained a total of 1599 samples, with 10 predictor features and two target classes labelled as 'high' and 'low'. The model's performance and hyperparameters were evaluated using a 10-fold cross-validation technique.

The decision tree model was trained using different values of the complexity parameter (cp), which regulates the pruning of the tree to avoid overfitting. The results table presents the accuracy and Cohen's kappa statistic for each cp value that was taken into account. Throughout the whole range of cp values, from 0.01 to 0.10, the model consistently obtained an accuracy of 0.5878695.

The model with the highest accuracy score, which was 0.5878695, was chosen as the best model for subset 3. This accuracy score remained consistent across all cp values. A precision of 0.5878695 signifies that the model accurately identified around 58.8% of the samples in subset 3.

The decision tree model built for subset 3, as shown in the image, includes only one leaf node. This leaf node classifies all occurrences into the ‘high’ class with a probability of 0.41, covering 100% of the data points.



```
## CART
##
## 1599 samples
##   10 predictor
##    2 classes: 'high', 'low'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 1439, 1439, 1439, 1439, 1440, 1439, ...
## Resampling results across tuning parameters:
##
##   cp    Accuracy   Kappa
##   0.01  0.5878695  0
##   0.02  0.5878695  0
##   0.03  0.5878695  0
##   0.04  0.5878695  0
##   0.05  0.5878695  0
##   0.06  0.5878695  0
##   0.07  0.5878695  0
##   0.08  0.5878695  0
##   0.09  0.5878695  0
##   0.10  0.5878695  0
## 
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.1.
```

```
## Confusion Matrix and Statistics
## 
##           Reference
## Prediction high low
##       high  402 282
##       low     0   0
##                                           
##                Accuracy : 0.5877          
##                  95% CI : (0.5498, 0.6249)
##     No Information Rate : 0.5877          
##     P-Value [Acc > NIR] : 0.5164          
##                                           
##                   Kappa : 0               
##                                           
##  Mcnemar's Test P-Value : <2e-16          
##                                           
##             Sensitivity : 1.0000          
##             Specificity : 0.0000          
##          Pos Pred Value : 0.5877          
##          Neg Pred Value :    NaN          
##              Prevalence : 0.5877          
##          Detection Rate : 0.5877          
##    Detection Prevalence : 1.0000          
##       Balanced Accuracy : 0.5000          
##                                           
##        'Positive' Class : high            
## 
```

#4.0Comparsion of GAM and Decision Tress

Applying Generalised Additive Models (GAMs) and Decision Trees (DTs) to three different data subsets allows for a meaningful comparison of their methodological efficiency and model interpretation. Generalised Additive Models (GAMs) are beneficial because of their adaptable structure, which allows for the identification and modelling of non-linear interactions between variables without the need to pre-determine the specific shape of these relationships. This is apparent from the Generalised Additive Model (GAM) results, which provide a more nuanced view of how different cities and states impact the target variable. For instance, cities such as Haddon Heights and Zephyrhills exhibit substantial coefficients, indicating a powerful impact on the target score. These coefficients can either increase or decrease the probability of obtaining a higher score, depending on their sign.

However, Decision Trees offer a direct, binary decision-making procedure that is especially advantageous for categorical results and is more easily understandable for non-technical individuals. The tree diagrams provide a clear representation of routes and probabilities, allowing for instant visual understanding of the most relevant aspects and the impact of different splits on the outcome. As an illustration, the decision tree derived from subset 1 unambiguously demonstrates that when state_DE is 0, it highly predicts a high score with a significant likelihood.

Nevertheless, the resampling outcomes for the Decision Trees across all subsets demonstrate a comparatively low accuracy and kappa, implying that the model may be oversimplified or not properly reflecting the complexity inherent in the data, unlike the GAMs. This may be attributed to overfitting or an insufficient depth configuration in the tree parameters. The Generalised Additive Models (GAMs), despite being more computationally demanding and requiring careful interpretation of spline functions and interactions, generally exhibited a higher rate of deviance explained across the subsets. This suggests that GAMs have the potential for more predictive ability and a better match to the data.

To summarise, Generalised Additive Models (GAMs) offer a comprehensive and adaptable analysis method that is well-suited for revealing intricate patterns. On the other hand, Decision Trees provide an intuitive and direct approach that is more easily understandable and can be used for making quick decisions. When choose amongst these models, it is important to consider the specific requirements for model transparency, the intricacy of linkages in the data, and the intended purpose of the model results. GAMs may be more suitable for rigorous scientific analysis that requires a nuanced understanding, whereas Decision Trees could be more advantageous for practical decisions that necessitate fast and straightforward interpretations.

#5.0 Conclusion

The present work employed Generalised Additive Models (GAMs) and Decision Trees to evaluate the performance scores of online companies using a comprehensive dataset. By employing a rigorous analytical approach, we were able to extensively investigate a range of predictive models, resulting in improved operational efficiencies and facilitating strategic decision-making. The results demonstrate the efficacy of GAMs in capturing intricate data patterns and offering subtle insights into the relationships between variables. On the other hand, Decision Trees provided a direct and easily understandable approach to analysis, making them well-suited for situations that need prompt and unambiguous decision-making.

#Recommendation

Model Selection: Wood (2017) suggests using Generalised Additive Models (GAMs) for datasets that have intricate, non-linear relationships and require a thorough analysis of variable impacts. Decision Trees are recommended for situations that demand quick decision-making because to their straightforwardness and lucidity. This can greatly benefit individuals who are not well-versed in technical matters (James et al., 2013).

Additional Data Analysis: Given the notable fluctuations seen in model performance, it is recommended to do a more comprehensive exploratory data study. Dominici et al. (2002)

suggest that further exploration of supplementary predictors and interactions may uncover significant patterns that have a substantial impact on model results.

Optimization and Verification of the Model: It is recommended to improve the accuracy and resilience of the model by refining parameters and using advanced validation approaches (Hastie et al., 2009). Utilising a broader range of cross-validation and model tuning techniques could reduce overfitting and enhance forecast accuracy.

Application of Sophisticated Methods: Utilising ensemble approaches, which include the combination of multiple models, can be useful in enhancing prediction accuracy. Random Forests and Gradient Boosting Machines are advanced algorithms that have shown exceptional ability to handle complicated data sets and outperform single-model approaches (Friedman et al., 2010).

#6.0 Refernces

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0 Step-by-step data mining guide. Available at: https://doi.org/10.13140/RG.2.1.4036.9049. Accessed on 6 May 2024.

Dominici, F., McDermott, A., Hastie, T. J., & Hastie, T. (2002). Improved semi-parametric time series models of air pollution and mortality. Journal of the American Statistical Association. Available at: https://doi.org/10.1198/016214502388618906. Accessed on 6 May 2024.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1), 1–22. Available at: https://doi.org/10.18637/jss.v033.i01. Accessed on 6 May 2024.

Härdle, W., Müller, M., Sperlich, S., & Werwatz, A. (2004). Nonparametric and Semiparametric Models. Springer.

Hastie, T., & Tibshirani, R. (1990). Generalized Additive Models. Chapman and Hall/CRC.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction (2nd ed.). Springer.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. Springer. Available at: https://doi.org/10.1007/978-1-4614-7138-7. Accessed on 6 May 2024.

Kumar, R., & Indrayan, A. (2011). Conditional approach to the decision tree for classification of groups of patients. Medical Decision Making, 31(4), 599-608. Available at: https://doi.org/10.1177/0272989X10381280. Accessed on May 6th, 2024.

Wood, S. N. (2017). Generalized Additive Models: An Introduction with R. CRC Press. Available at: https://doi.org/10.1201/9781315370279. Accessed on 6 May 2024.