IDA-HW6-Group 20

Vignesh Murugan , Vivek Satya Sai Veera Venkata Talluri and Bhuvanesh So Muruganandam

2024-10-22

(a) (50 points) Preparation and modeling.

i. (10 points) Data understanding. Generate a Data Quality Report. Also, choose at least two meaningful visualizations and/or analyses and explain their relevance.

Loading and Inspecting Data

The dataset is loaded and converted into a tibble for better handling and integration with the tidyverse ecosystem. Only numeric columns are retained in a new subset for statistical summaries and modeling purposes.

Displaying Column Names:

The column names are listed. This allows a quick inspection of which numeric features were extracted, helping ensure the correct columns were selected for further analysis or modeling.

```
##  [1] "sessionId"             "custId"                "visitStartTime"
##  [4] "visitNumber"           "timeSinceLastVisit"    "isMobile"
##  [7] "isTrueDirect"          "adwordsClickInfo.page" "pageviews"
## [10] "bounces"               "newVisits"             "revenue"
```

Summary Statistics of Numeric Features

This function computes the following summary statistics for a given numeric vector: n – Total number of observations. unique – Number of unique values. missing – Count of missing values. mean – Average value (ignoring missing data). min – Minimum value. Q1 – First quartile. median – Median value. Q3 – Third quartile. max – Maximum value. sd – Standard deviation.

This produces a comprehensive summary of the numeric features, including key statistics, missing values, and uniqueness percentages. This summary provides valuable insights into the dataset's structure, completeness, and variability, helping to inform data cleaning and feature engineering decisions before modeling.

```
## # A tibble: 12 x 13
##    variable      n missing missing_pct unique unique_pct     mean    min      Q1
##    <chr>     <dbl>   <dbl>       <dbl>  <dbl>      <dbl>    <dbl>  <dbl>   <dbl>
##  1 sessionId 70071       0      0       70071  100       4.71e+12 2.00e8 2.33e12
##  2 custId    70071       0      0       47249   67.4     4.89e+ 4 1.80e3 2.51e 4
##  3 visitSta~ 70071       0      0       69951   99.8     1.49e+ 9 1.47e9 1.48e 9
##  4 visitNum~ 70071       0      0         155    0.221   3.15e+ 0 1   e0 1   e 0
##  5 timeSinc~ 70071       0      0       20970   29.9     2.56e+ 5 0      0
```

```
##  6 isMobile  70071       0      0           2    0.00285 2.29e- 1 0      0
##  7 isTrueDi~ 70071       0      0           2    0.00285 4.00e- 1 0      0
##  8 adwordsC~ 70071   68260     97.4         6    0.00856 1.01e+ 0 1   e0 1   e 0
##  9 pageviews 70071       8      0.0114    155    0.221   6.30e+ 0 1   e0 1   e 0
## 10 bounces   70071   40719     58.1         2    0.00285 1   e+ 0 1   e0 1   e 0
## 11 newVisits 70071   23944     34.2         2    0.00285 1   e+ 0 1   e0 1   e 0
## 12 revenue   70071       0      0        5850    8.35    1.02e+ 1 0      0
## # i 4 more variables: median <dbl>, Q3 <dbl>, max <dbl>, sd <dbl>
```

Columns with Excessive Missing Values:

The features adwordsClickInfo.page, bounces, and newVisits contain missing values in more than 50% of the samples. Retaining these columns could introduce bias and reduce the effectiveness of downstream models. Therefore, these features will be dropped to ensure the integrity of the analysis and maintain the reliability of insights.

Dropping sessionId:

The sessionId field contains unique values for each observation, providing no meaningful information for modeling or analysis. As it functions purely as an identifier, it does not contribute to feature engineering or prediction and will be excluded from further processing.

```
## # A tibble: 8 x 13
##   variable         n missing missing_pct unique unique_pct    mean    min     Q1
##   <chr>        <dbl>   <dbl>       <dbl>  <dbl>      <dbl>   <dbl>  <dbl>  <dbl>
## 1 custId       70071       0      0       47249   67.4     4.89e+4 1.80e3 2.51e4
## 2 visitStartT~ 70071       0      0       69951   99.8     1.49e+9 1.47e9 1.48e9
## 3 visitNumber  70071       0      0         155    0.221   3.15e+0 1   e0 1   e0
## 4 timeSinceLa~ 70071       0      0       20970   29.9     2.56e+5 0      0
## 5 isMobile     70071       0      0           2    0.00285 2.29e-1 0      0
## 6 isTrueDirect 70071       0      0           2    0.00285 4.00e-1 0      0
## 7 pageviews    70071       8      0.0114    155    0.221   6.30e+0 1   e0 1   e0
## 8 revenue      70071       0      0        5850    8.35    1.02e+1 0      0
## # i 4 more variables: median <dbl>, Q3 <dbl>, max <dbl>, sd <dbl>
```

The pageviews column contains 0.0114% missing values, a relatively small proportion of the dataset. To maintain the data quality without discarding valuable information, we will apply K-Nearest Neighbors (KNN) imputation from the VIM package.

KNN imputation is chosen as it leverages the similarity between data points, providing more accurate estimates by filling in missing values based on the closest neighbors. This approach ensures consistency and avoids introducing bias from simpler imputations like mean or median.

```
## # A tibble: 8 x 13
##   variable         n missing missing_pct unique unique_pct    mean    min     Q1
##   <chr>        <dbl>   <dbl>       <dbl>  <dbl>      <dbl>   <dbl>  <dbl>  <dbl>
## 1 custId       70071       0           0  47249   67.4     4.89e+4 1.80e3 2.51e4
## 2 visitStartT~ 70071       0           0  69951   99.8     1.49e+9 1.47e9 1.48e9
## 3 visitNumber  70071       0           0    155    0.221   3.15e+0 1   e0 1   e0
## 4 timeSinceLa~ 70071       0           0  20970   29.9     2.56e+5 0      0
## 5 isMobile     70071       0           0      2    0.00285 2.29e-1 0      0
```

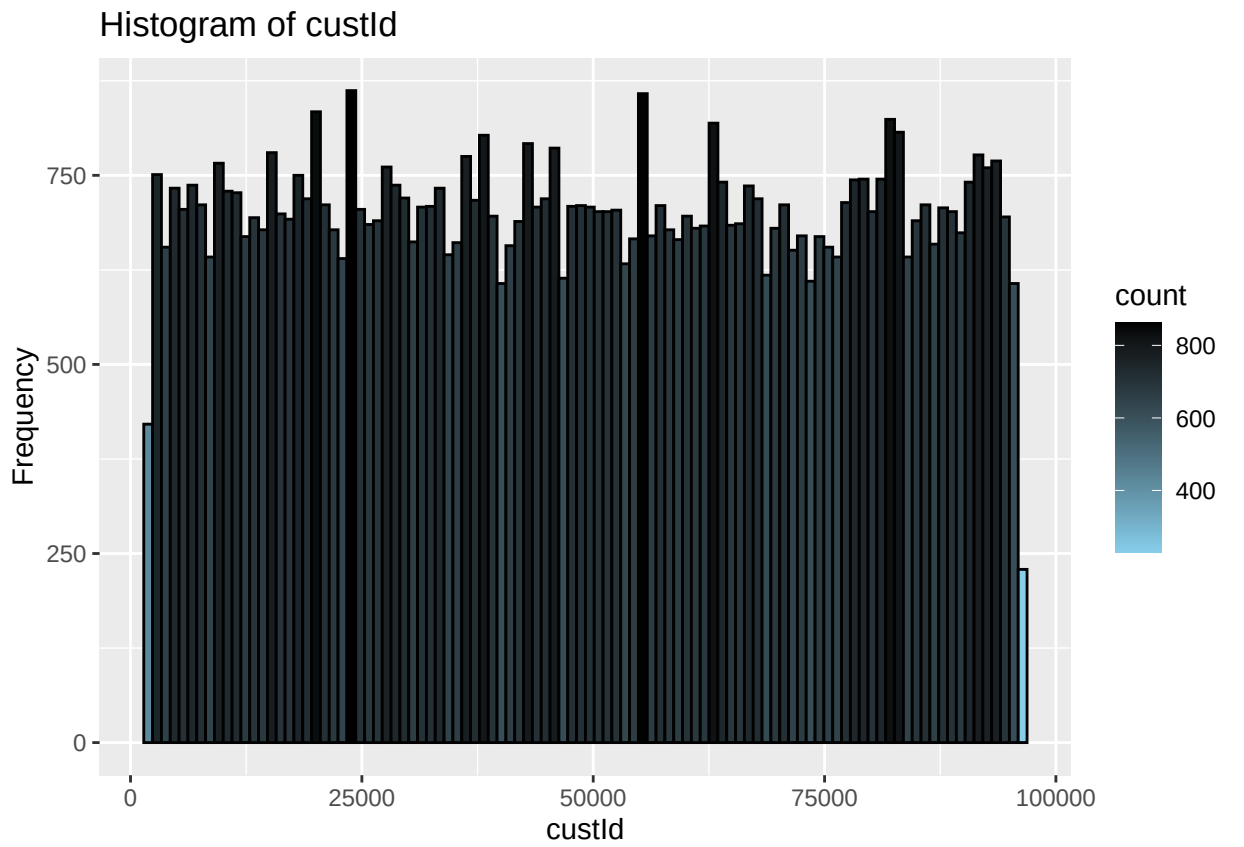
```
## 6 isTrueDirect 70071       0           0      2    0.00285 4.00e-1 0      0
## 7 pageviews    70071       0           0    154    0.220   6.30e+0 1    e0 1    e0
## 8 revenue      70071       0           0   5850    8.35    1.02e+1 0      0
## # i 4 more variables: median <dbl>, Q3 <dbl>, max <dbl>, sd <dbl>
```

We will plot histograms to examine the distribution of the variables. From the visual inspection, most variables exhibit a right-skewed distribution.

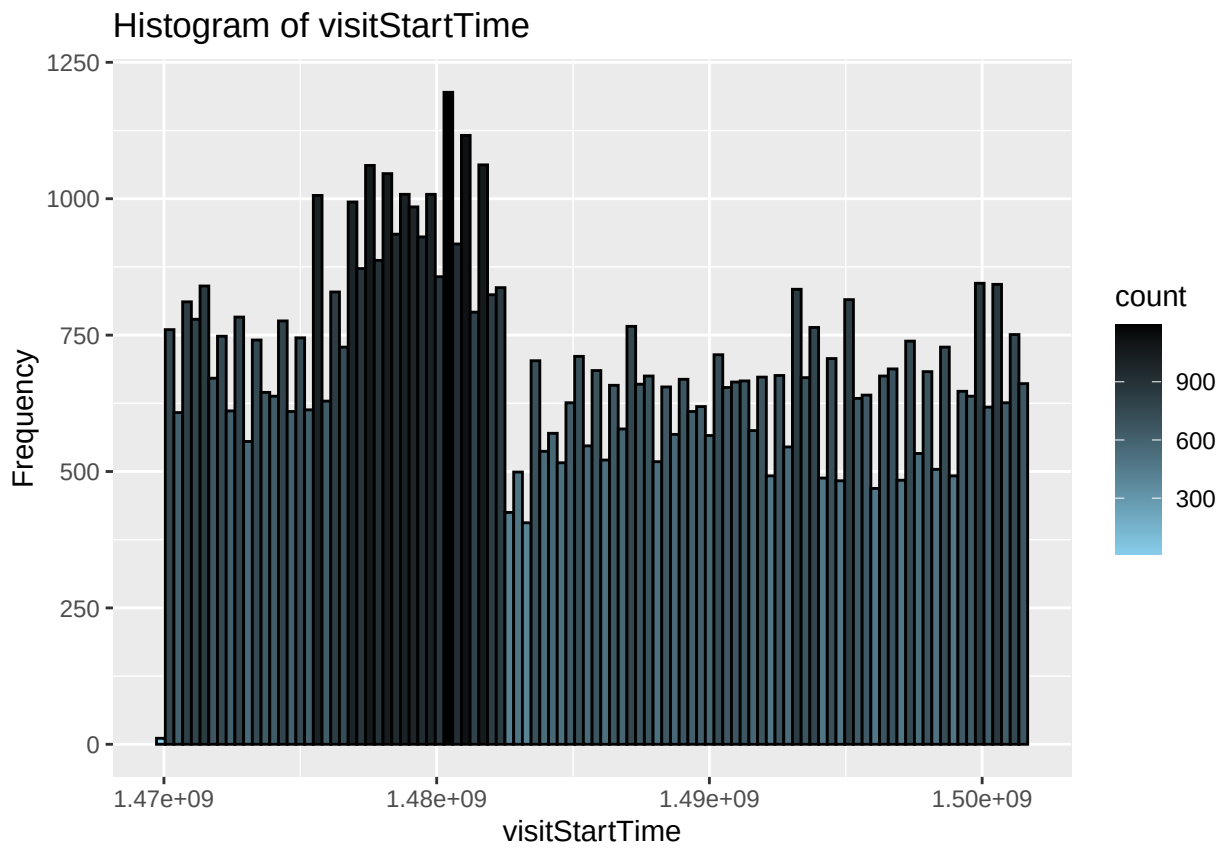
A right-skew indicates that the majority of the data points are concentrated on the lower end, with a long tail extending towards higher values. This suggests that further transformations (e.g., log or Box-Cox transformation) may be necessary to normalize the data for certain machine learning models that assume normally distributed input features.

```
## $custId
```

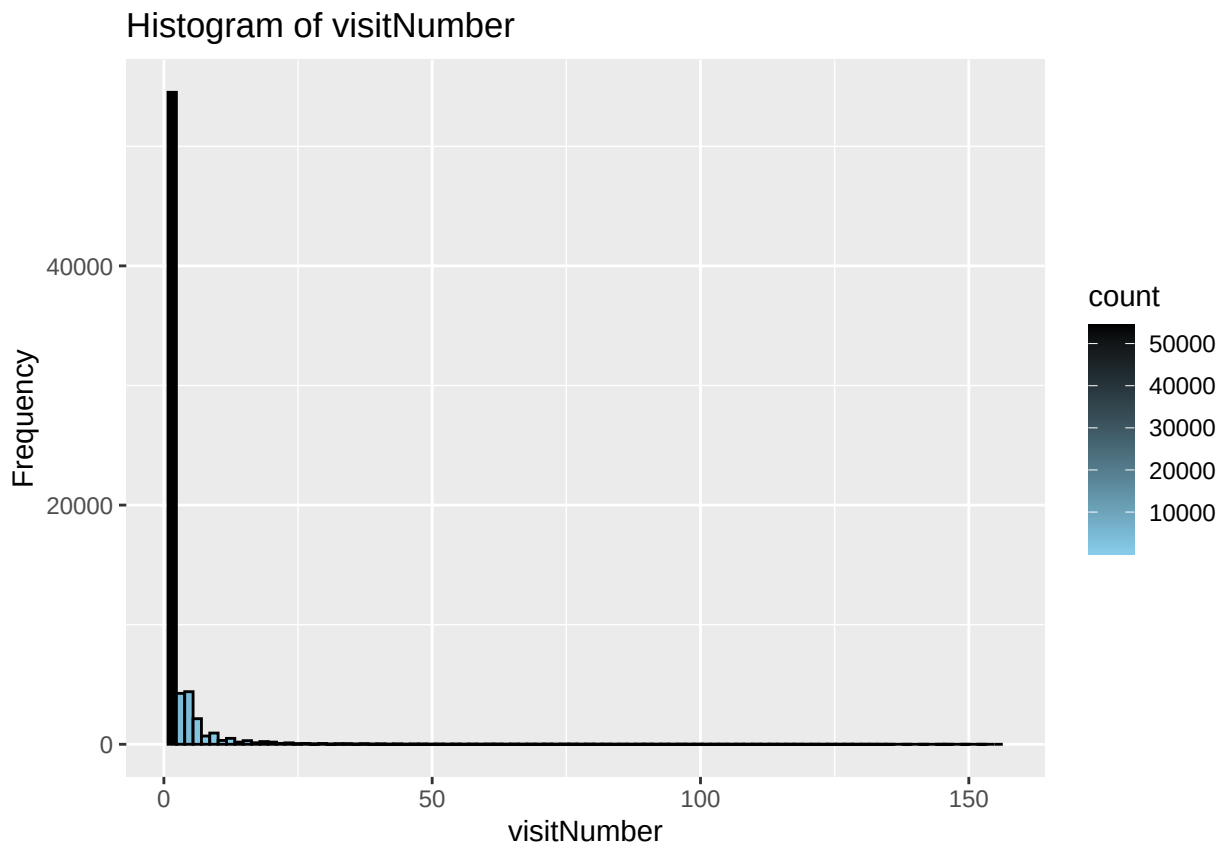
```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



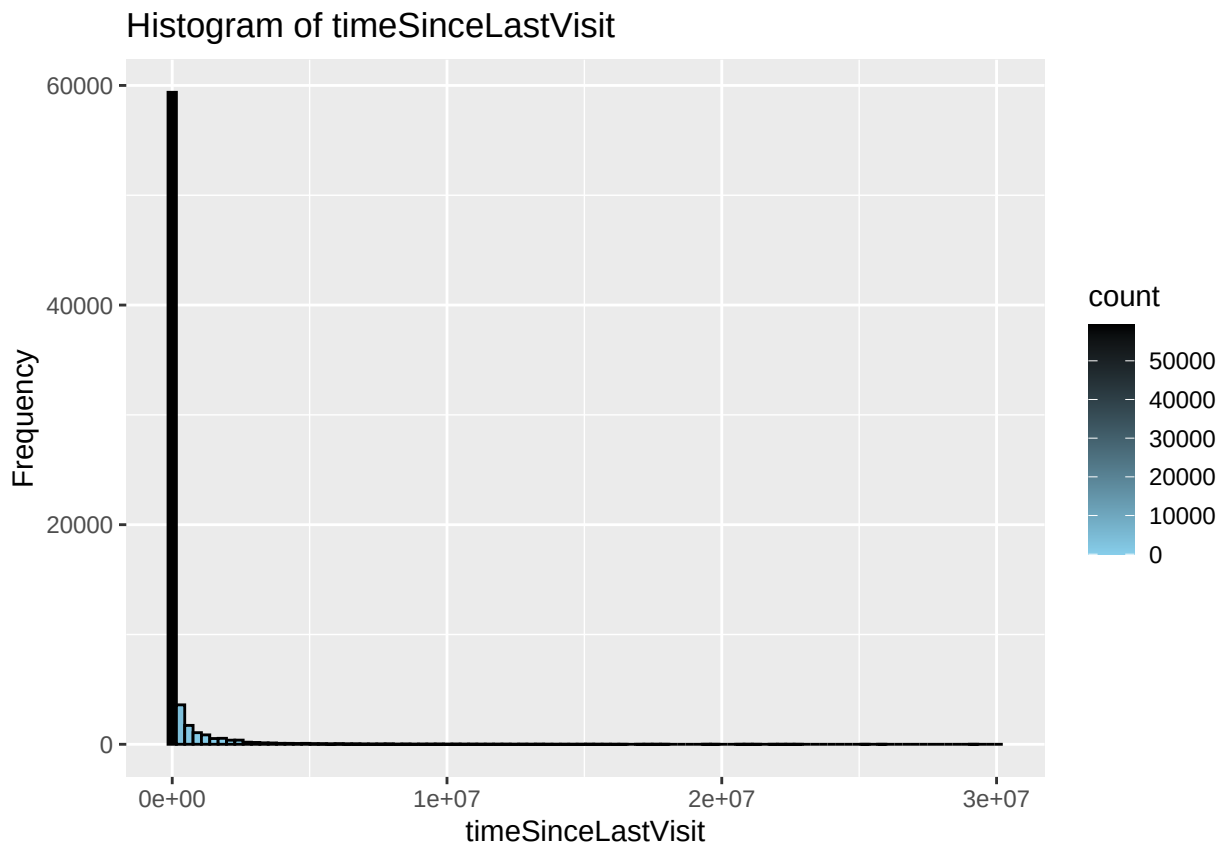

```
##
## $visitStartTime
```



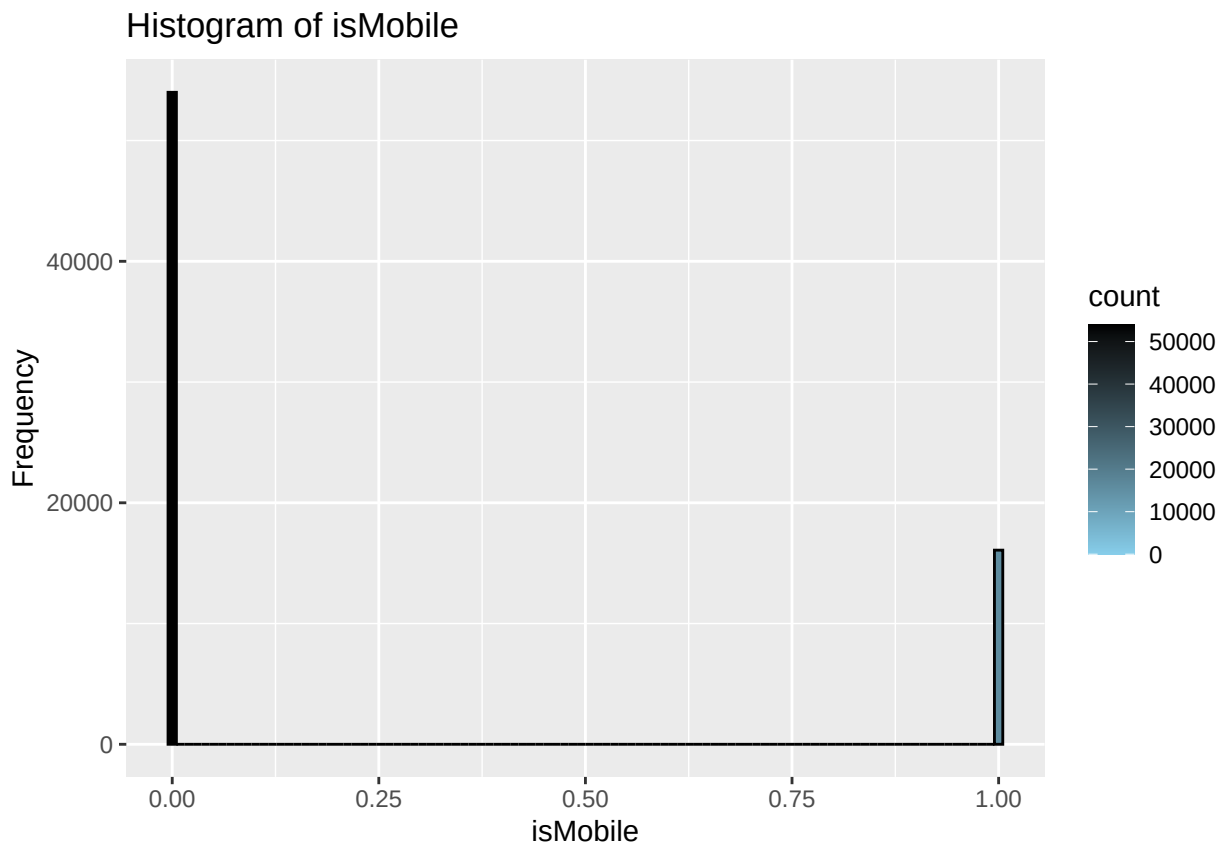

```
##
## $visitNumber
```



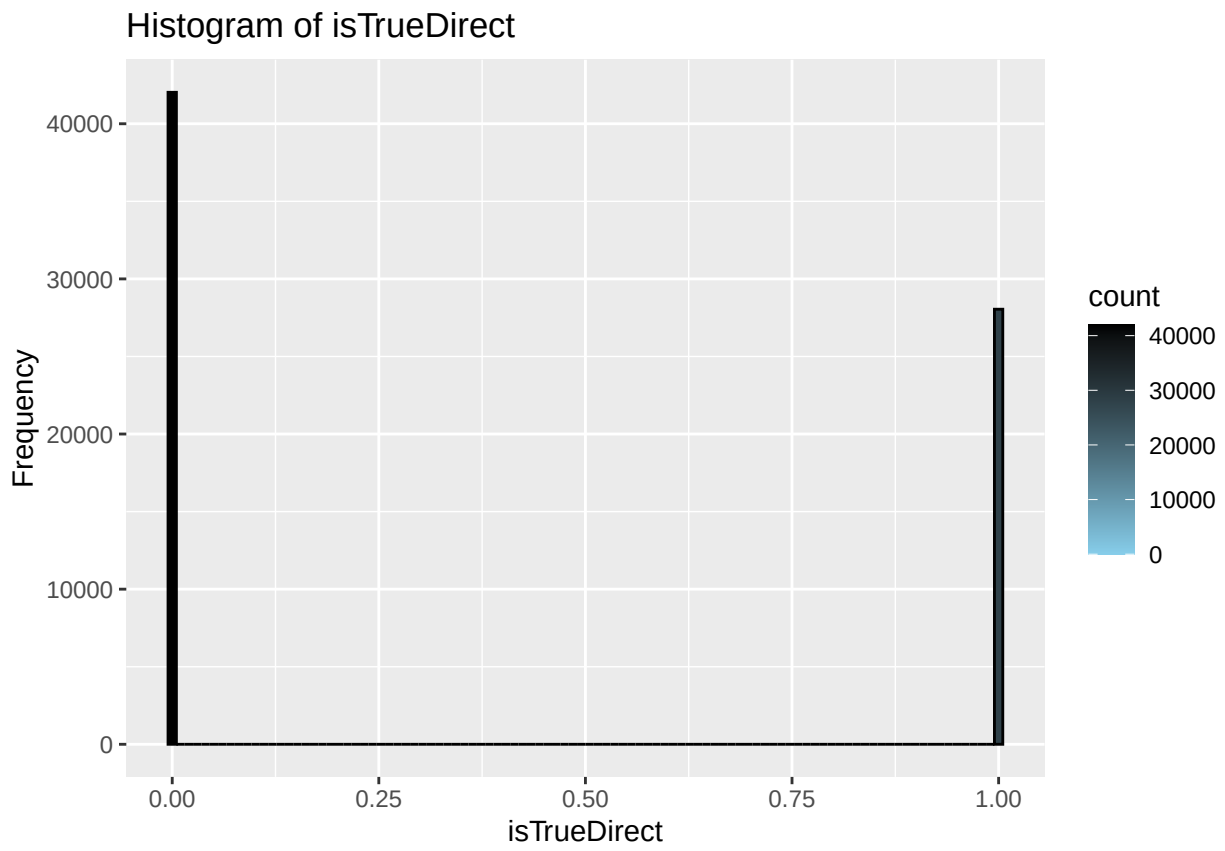

```
##
## $timeSinceLastVisit
```


Histogram of timeSinceLastVisit
60000
40000
20000
0
Frequency
0e+00
1e+07
2e+07
3e+07
timeSinceLastVisit
count
50000
40000
30000
20000
10000
0

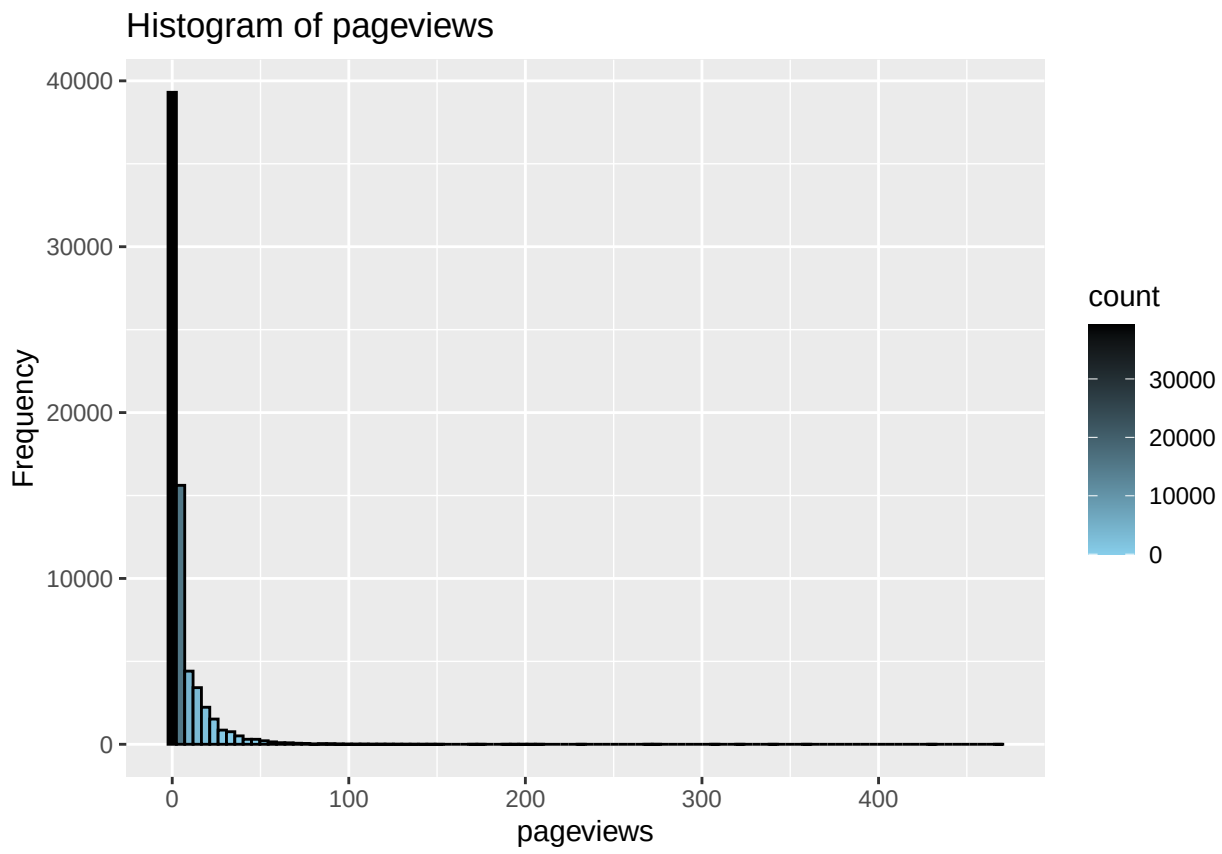
```
##
## $isMobile
```



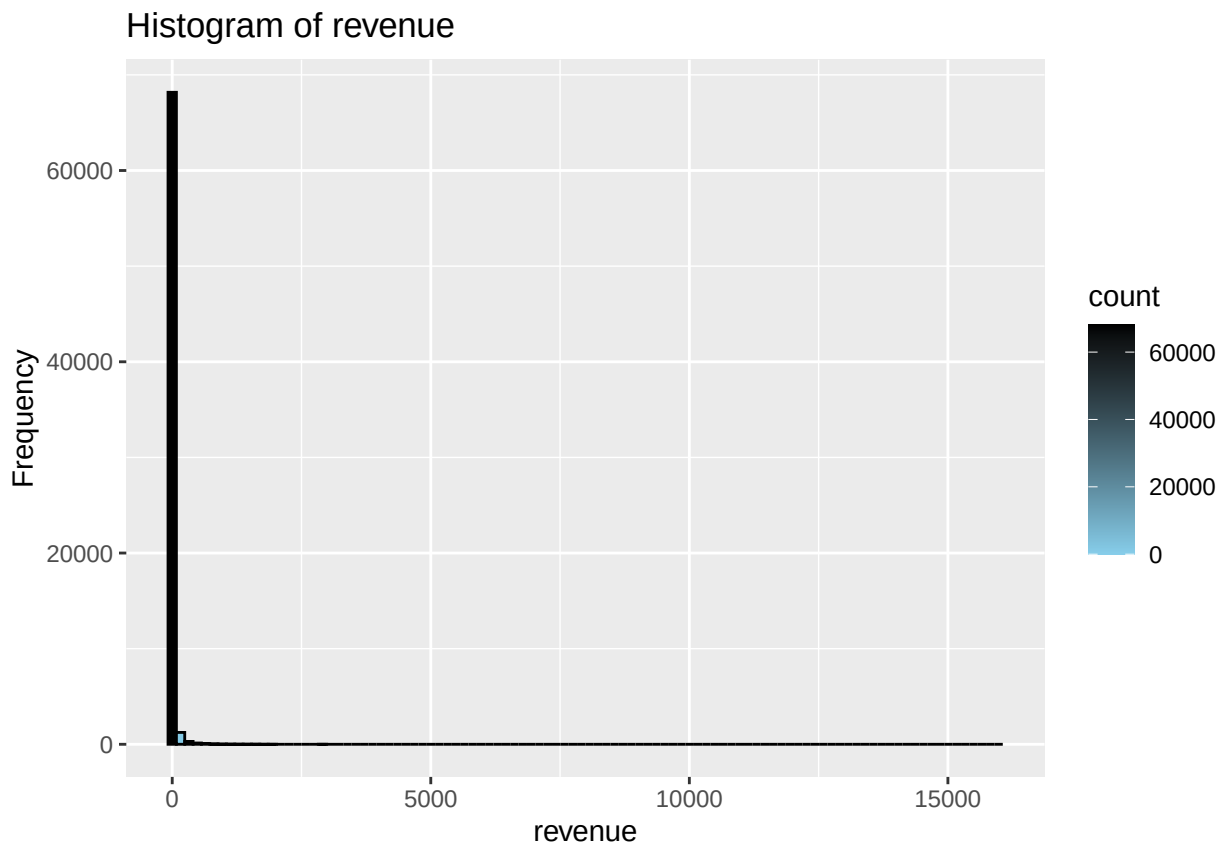

```
##
## $isTrueDirect
```


Histogram of isTrueDirect
Frequency
40000
30000
20000
10000
0
0.00
0.25
0.50
0.75
1.00
isTrueDirect
count
40000
30000
20000
10000
0

```
##
## $pageviews
```


Histogram of pageviews
40000
30000
20000
10000
0
Frequency
0
100
200
300
400
pageviews
count
30000
20000
10000
0

```
##
## $revenue
```



2. Factor Variables

Summarizes key statistics for factor variables, including: Total number of observations. Number of unique values. Count of missing values. Most frequent (1st mode) and its frequency. Second most frequent value and its frequency. Least common value and its frequency.

Displaying Column Names: The column names of Train_Factor are listed using colnames(). This allows a quick inspection of which Factor features were extracted, helping ensure the correct columns were selected for further analysis or modeling.

```
##  [1] "channelGrouping"               "browser"
##  [3] "operatingSystem"               "deviceCategory"
##  [5] "continent"                     "subContinent"
##  [7] "country"                       "region"
##  [9] "metro"                         "city"
## [11] "networkDomain"                 "topLevelDomain"
## [13] "campaign"                      "source"
## [15] "medium"                        "keyword"
## [17] "referralPath"                  "adContent"
## [19] "adwordsClickInfo.slot"         "adwordsClickInfo.gclId"
## [21] "adwordsClickInfo.adNetworkType"
```

```
## # A tibble: 21 x 2
##    variable                      missing_pct
##    <chr>                               <dbl>
```

```
##  1 adContent                          98.8
##  2 adwordsClickInfo.slot              97.4
##  3 adwordsClickInfo.adNetworkType     97.4
##  4 adwordsClickInfo.gclId             97.4
##  5 keyword                            96.2
##  6 campaign                           96.1
##  7 metro                              70.2
##  8 referralPath                       61.5
##  9 city                               55.7
## 10 region                             54.9
## # i 11 more rows
```

ii. (10 points) Data preparation. Choose two of the most critical data preparation actions you took and explain the reasoning for these actions.

The listed columns are removed from the dataset because they contain a high percentage of missing values, as outlined below: adContent: 98.8% missing adwordsClickInfo.slot: 97.4% missing adwordsClickInfo.adNetworkType: 97.4% missing keyword: 96.2% missing city: 55.7% missing These features have limited useful information because most values are missing.

Features with missing percentages over 50% (or close to it) are generally considered unreliable for analysis. Imputation becomes impractical for such features, as filling in so many missing values could introduce significant bias or noise.

Dropping these columns ensures that the dataset is cleaner and more reliable for analysis, reducing the risk of errors in downstream tasks such as model training.

The Barplot of All Factor Variables are plotted.

The Factor Levels of all Factor Varibales are printed.

ii. (10 points) Data preparation. Choose two of the most critical data preparation actions you took and explain the reasoning for these actions.

Feature Engineering with Factor Variables:

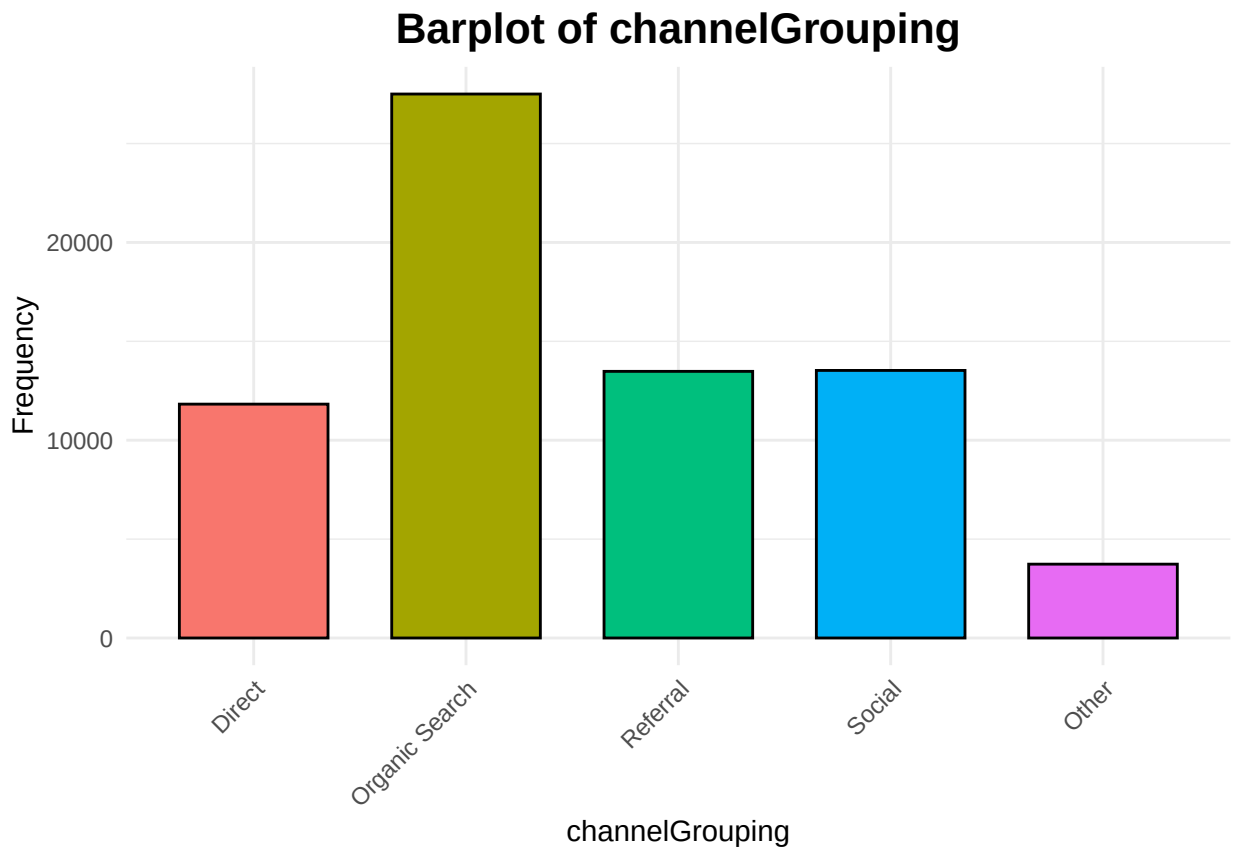
In the following steps, we are lumping factor levels using the fct_lump_n() function from the forcats package. This technique groups less frequent categories into an "Other" level to reduce the dimensionality of categorical features, making models more efficient and interpretable. Below is the summary of the transformations:

channelGrouping – Retained the 4 most frequent levels; all other levels are grouped under "Other." browser – Retained the 2 most frequent levels; the rest are grouped into "Other." operatingSystem – Retained the 6 most frequent levels, grouping the remainder as "Other." source – Retained the 4 most frequent levels, with the rest lumped into "Other." medium – Retained the 3 most frequent levels; all other levels are merged into "Other." country – Retained only the 1 most frequent country; all others are grouped as "Other." subContinent – Retained the 1 most frequent sub-continent; all others are grouped into "Other." continent – Retained the 3 most frequent continents, with the remaining grouped into "Other." This lumping process helps simplify complex categorical data by consolidating rare categories, improving the performance of downstream models while maintaining the most important information. After each transformation, fct_count() is used to display the new count of each factor level.

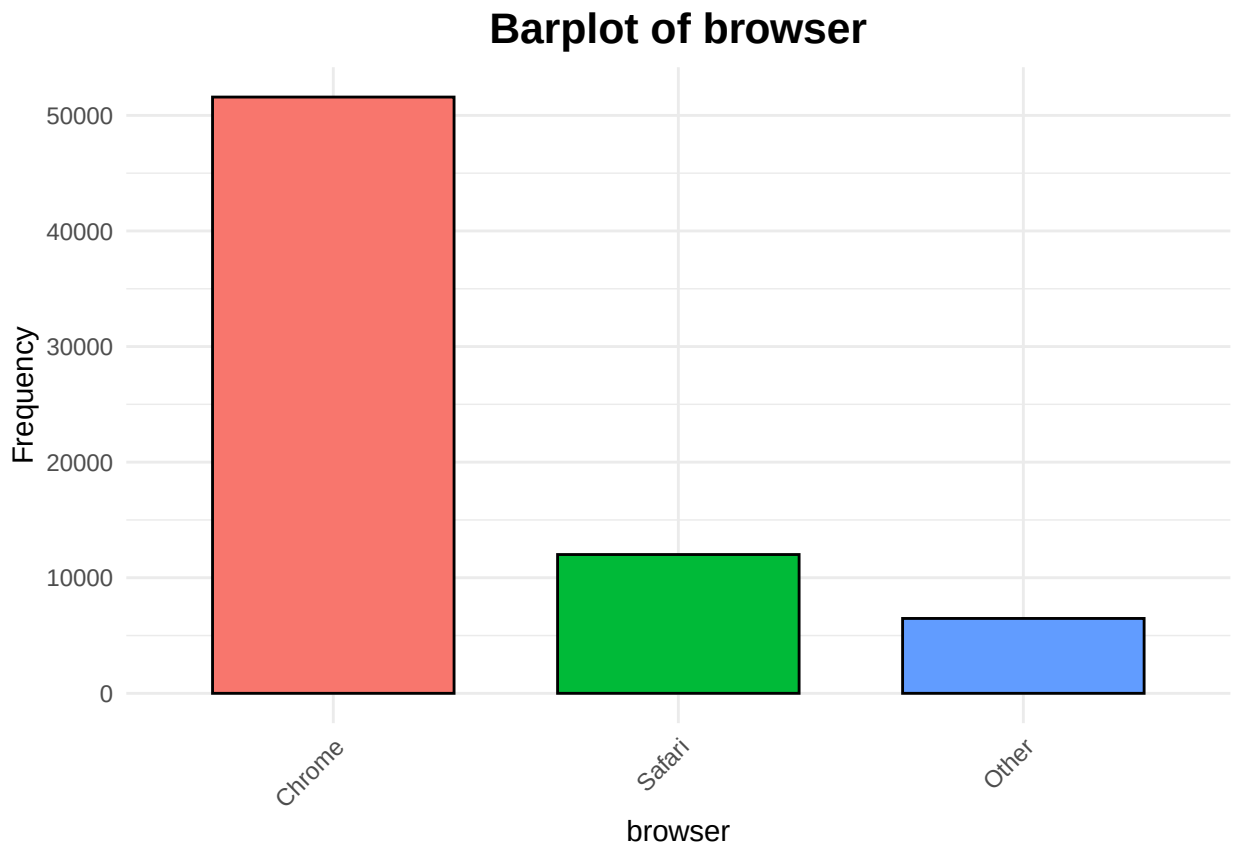
The subContinent column is being removed from the Train_Factor dataset because it is identical to the Country column, meaning it offers no additional value or new information for analysis.

The hot-deck imputation method is applied to Train_Factor using the hotdeck() function. This method replaces missing values with those from similar records within the dataset.

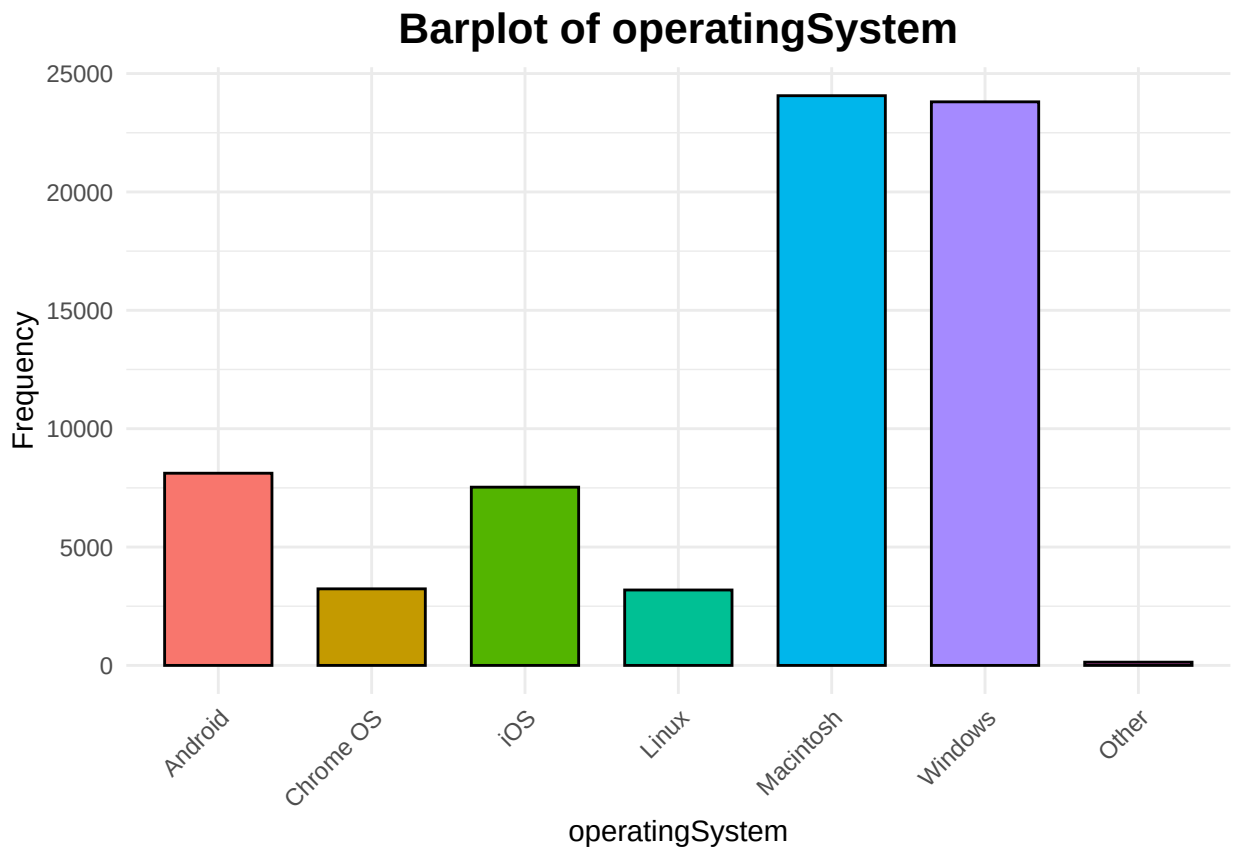
```
## $channelGrouping
```



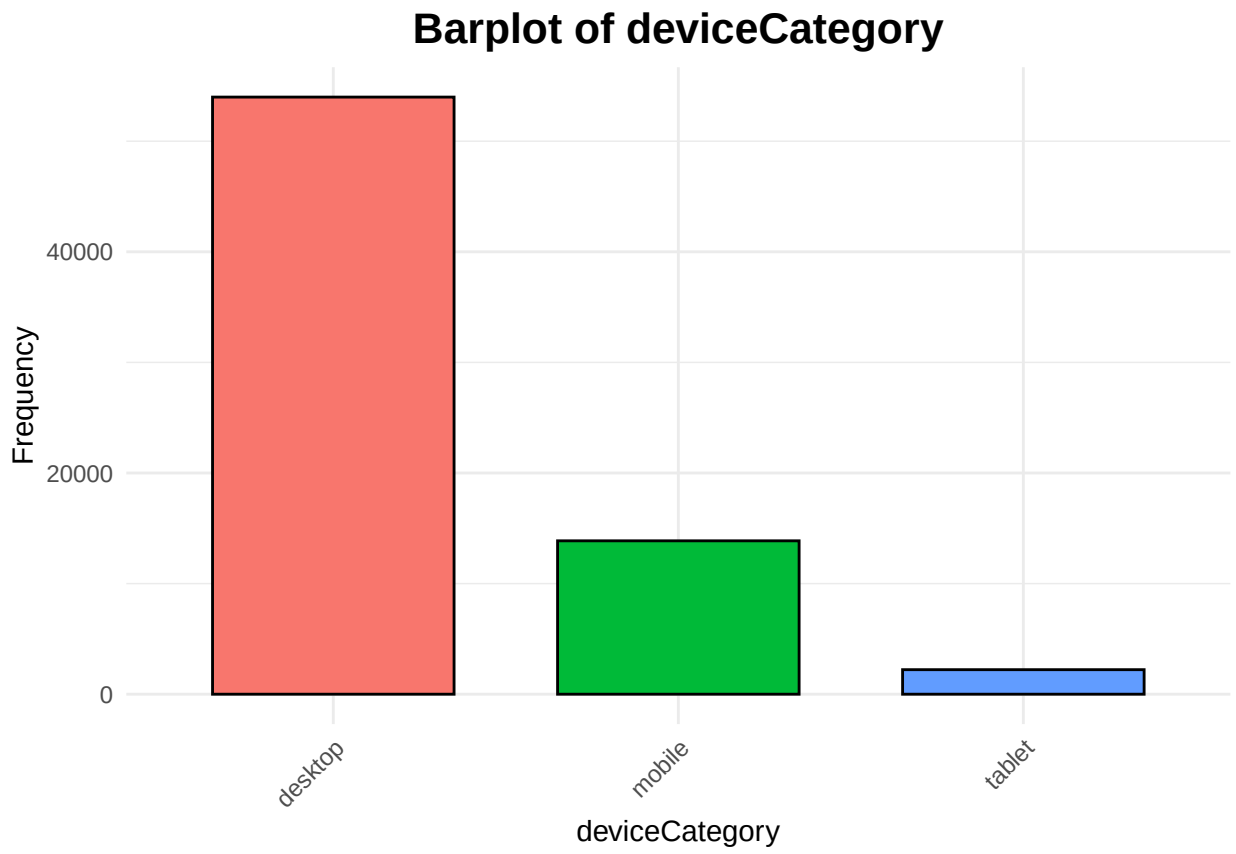

```
##
## $browser
```



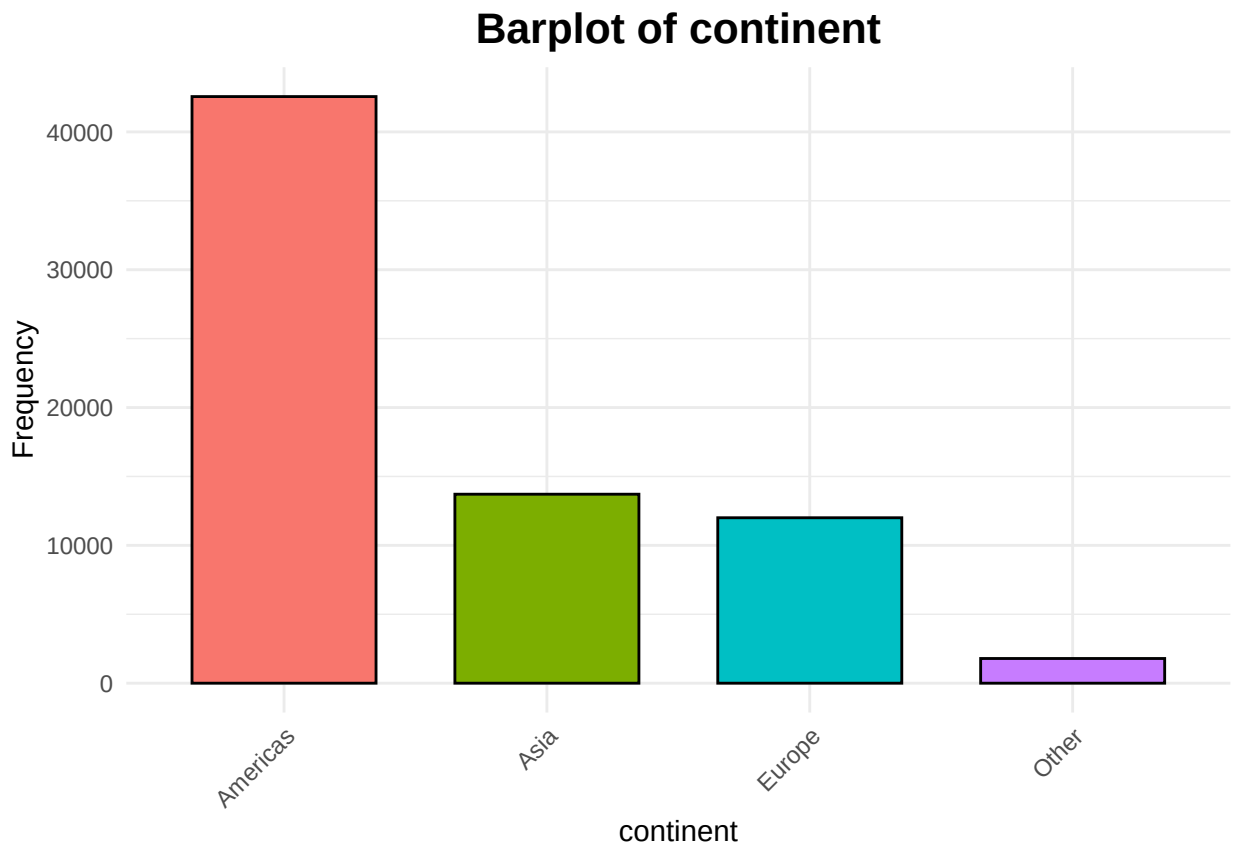

```
##
## $operatingSystem
```



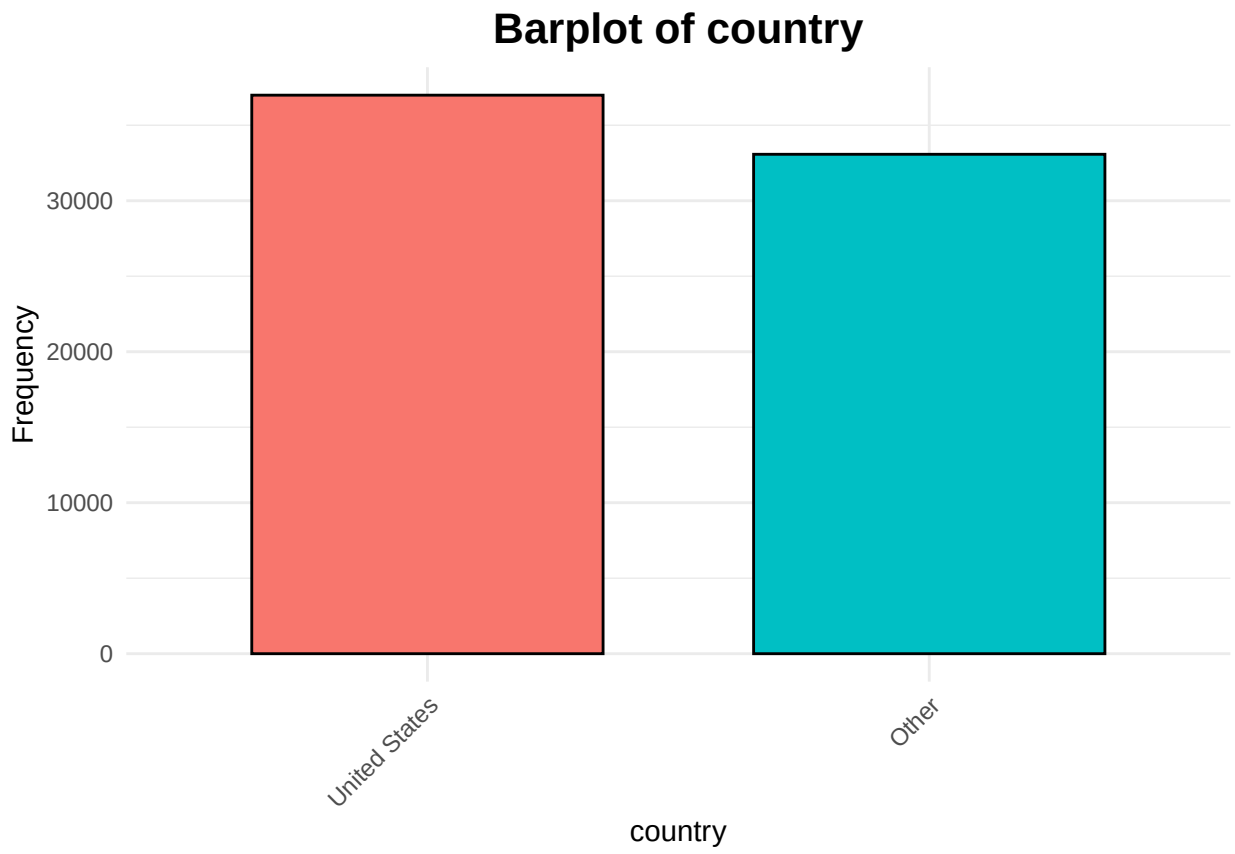

```
##
## $deviceCategory
```


Barplot of deviceCategory
Frequency
40000
20000
0
desktop
mobile
tablet
deviceCategory

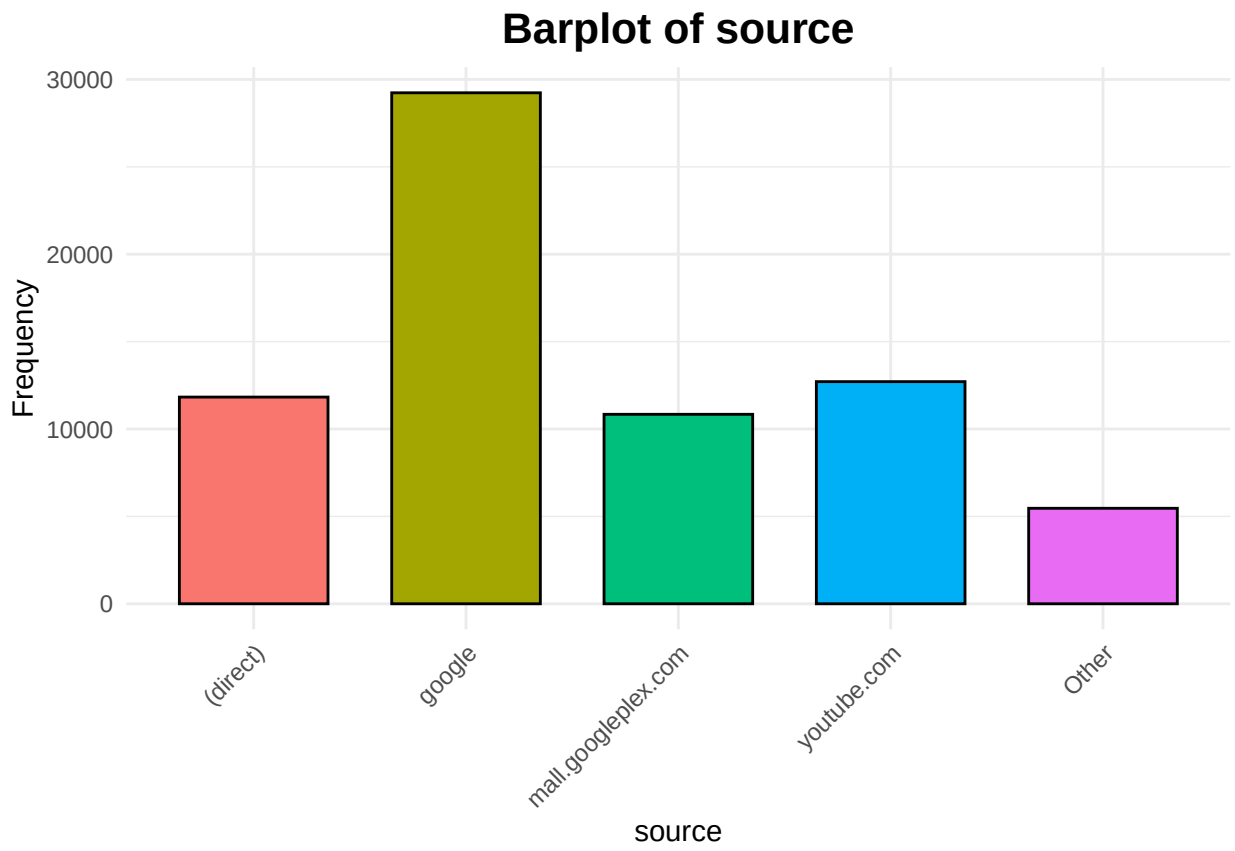
```
##
## $continent
```


Barplot of continent
40000
30000
20000
10000
0
Frequency
Americas
Asia
Europe
Other
continent

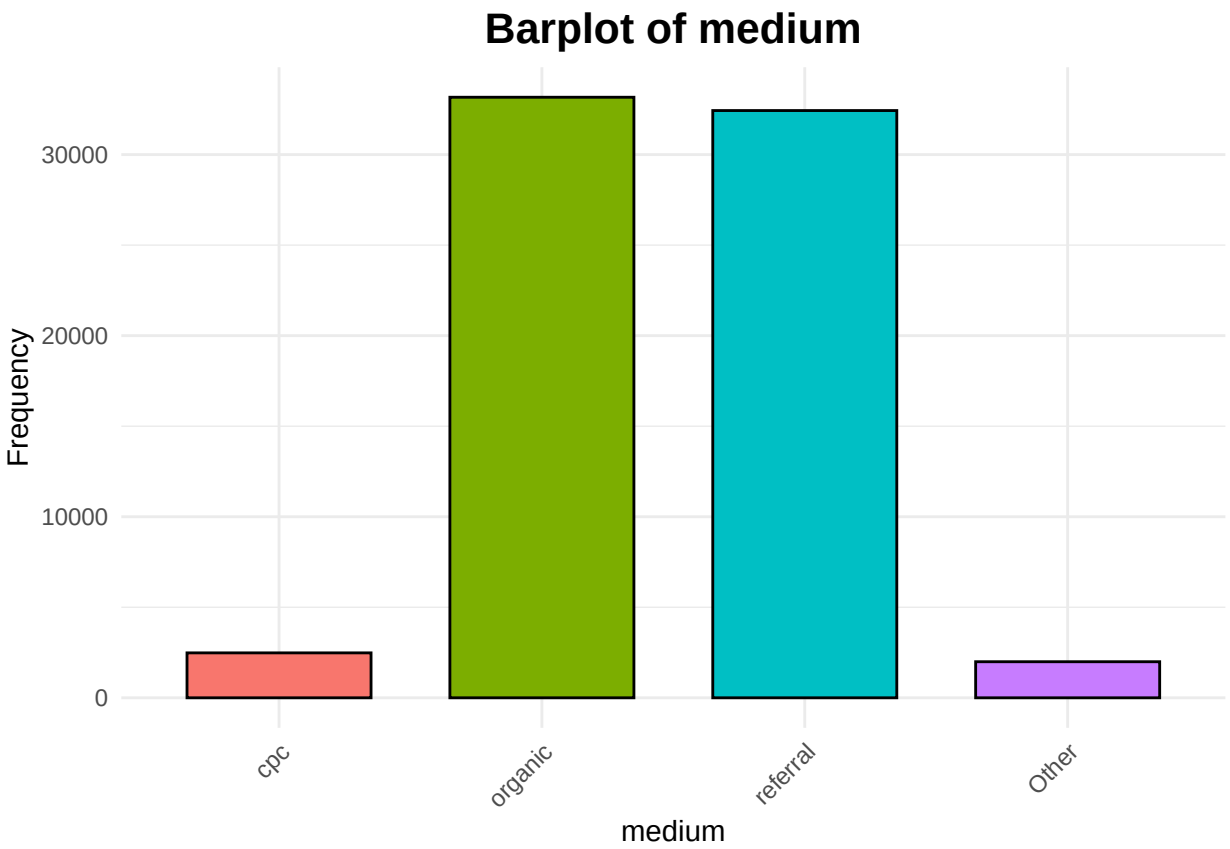
```
##
## $country
```

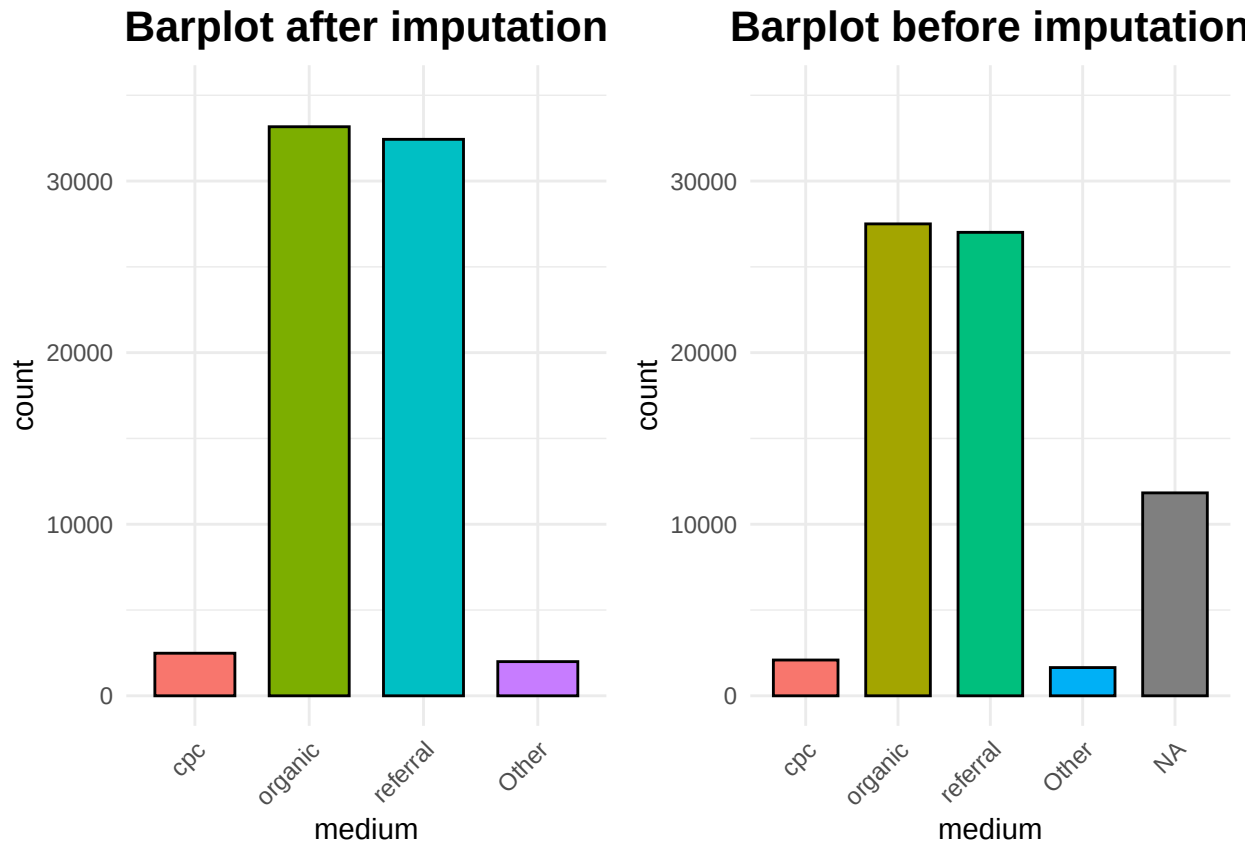

Barplot of country
Frequency
30000
20000
10000
0
United States
Other
country

```
##
## $source
```


Barplot of source
30000
20000
10000
0
Frequency
(direct)
google
mall.googleplex.com
youtube.com
Other
source

```
##
## $medium
```


Barplot of medium
Frequency
30000
20000
10000
0
cpc
organic
referral
Other
medium



The date column from the Train dataset is selected and then combined with Train_Numeric and Train_Factor_imputed using bind_cols(). This creates a consolidated dataset named Final_Data containing the date, numeric, and factor variables. This step ensures that all relevant features from different sources are merged into a single data frame for further analysis.

```
##  [1] "date"              "custId"            "visitStartTime"
##  [4] "visitNumber"       "timeSinceLastVisit" "isMobile"
##  [7] "isTrueDirect"      "pageviews"         "revenue"
## [10] "channelGrouping"   "browser"           "operatingSystem"
## [13] "deviceCategory"    "continent"         "country"
## [16] "source"            "medium"
```

```
##               date             custId     visitStartTime        visitNumber
##                  0                  0                  0                  0
## timeSinceLastVisit           isMobile      isTrueDirect          pageviews
##                  0                  0                  0                  0
##            revenue    channelGrouping            browser    operatingSystem
##                  0                  0                  0                  0
##     deviceCategory          continent            country             source
##                  0                  0                  0                  0
##             medium
##                  0
```

Aggregated customer Specific Dataset

The dataset Final_Data is grouped by the unique customer identifier custId. This allows aggregation of customer-related metrics across multiple sessions or visits.

log_total_revenue:

Computes the log-transformed total revenue for each customer using log1p(). log(1+x) to avoid log(0) issues).

total_visits:

Counts the total number of visits by each customer (n()).

avg_pageviews_per_visit:

Averages the number of page views per visit for each customer.

days_between_first_last_visit:

Calculates the time difference between the first and last visit.

different_day_visits:

Counts how many unique days a customer has visited (n_distinct()).

total_sessions:

Finds the highest recorded session number (max()) to represent total sessions.

avg_session_num:

Averages the session number to understand session frequency.

avg_time_since_last & max_time_since_last:

Calculates the average and maximum time between successive visits.

mobileUsage & directVisits:

Computes the average proportion of visits made from mobile devices and through direct access.

pageviews:

Aggregates the total number of page views by each customer.

mostCommonBrowser, mostCommonOS, and mostCommonDevice:

Finds the most frequently used browser, operating system, and device category across the customer's visits.

primaryChannel:

Identifies the most used marketing channel for each customer.

mostFrequentContinent & mostFrequentCountry:

Determines the most frequent continent and country from which the customer accessed the website.

primarySource & primaryMedium:

Extracts the most frequent traffic source and medium used by the customer.

mostCommonDay:

Finds the day of the week on which the customer most frequently visited.

```
##  [1] "custId"                       "log_total_revenue"
##  [3] "total_visits"                 "avg_pageviews_per_visit"
##  [5] "days_between_first_last_visit" "different_day_visits"
##  [7] "total_sessions"               "avg_session_num"
##  [9] "avg_time_since_last"          "max_time_since_last"
## [11] "mobileUsage"                  "directVisits"
## [13] "pageviews"                    "mostCommonBrowser"
## [15] "mostCommonOS"                 "mostCommonDevice"
## [17] "primaryChannel"               "mostFrequentContinent"
## [19] "mostFrequentCountry"          "primarySource"
## [21] "primaryMedium"                "mostCommonDay"
```

Selecting Only Numeric Features: A subset of the data is created, focusing only on numeric columns. Excluding non-numeric columns ensures that the remaining features are suitable for statistical techniques or machine learning algorithms.

Excluding Specific Columns: The code removes the log_total_revenue and custId columns because: custId is a unique identifier, not useful for analysis.

log_total_revenue is excluded to avoid data leakage as it's the target variable in a predictive model.

```
##  [1] "total_visits"                 "avg_pageviews_per_visit"
##  [3] "days_between_first_last_visit" "different_day_visits"
##  [5] "total_sessions"               "avg_session_num"
##  [7] "avg_time_since_last"          "max_time_since_last"
##  [9] "mobileUsage"                  "directVisits"
## [11] "pageviews"
```

ii. (10 points) Data preparation. Choose two of the most critical data preparation actions you

took and explain the reasoning for these actions.

Test for Skewness

Computing skewness for each variable in the customer_features_Numeric dataset.

Skewness is computed for all variables while ignoring missing values. The results are organized and sorted by skewness values to identify which variables have significant positive or negative skew. These insights can help inform further data preprocessing steps, such as transformations (e.g., log or Box-Cox) to normalize skewed distributions or detect potential outliers.

```
## # A tibble: 11 x 2
##    Variable                      Skewness
##    <chr>                            <dbl>
##  1 mobileUsage                       1.13
##  2 directVisits                      1.18
##  3 avg_pageviews_per_visit           7.23
##  4 days_between_first_last_visit     8.41
##  5 max_time_since_last               9.22
##  6 avg_time_since_last              10.6
##  7 pageviews                        13.5
##  8 different_day_visits             17.0
##  9 total_sessions                   23.4
## 10 avg_session_num                  25.2
## 11 total_visits                     26.5
```

ii. (10 points) Data preparation. Choose two of the most critical data preparation actions you took and explain the reasoning for these actions.

BoxCox Tranformations

Box-Cox transformation is applied to multiple numeric features to reduce skewness and make distributions more normal-like.

Adding 1 to certain variables (e.g., mobileUsage + 1) ensures there are no zero values, as the Box-Cox transformation only works on positive values.

Purpose: This transformation stabilizes variance and makes the data more suitable for statistical models (e.g., linear regression).

```
## Box-Cox Transformation
##
## 47249 data points used to estimate Lambda
##
## Input data summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   1.253   2.000   2.000
##
## Largest/Smallest: 2
## Sample Skewness: 1.13
##
## Estimated Lambda: -2

## Box-Cox Transformation
##
```

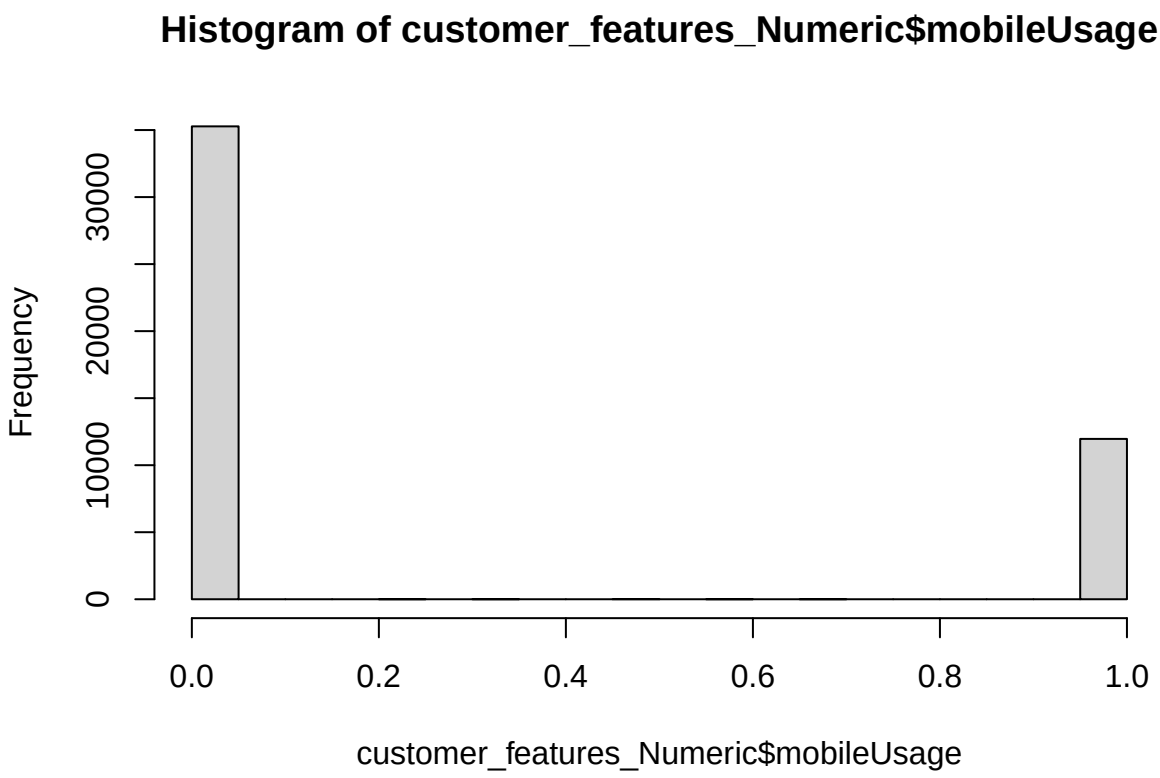
```
## 47249 data points used to estimate Lambda
##
## Input data summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   1.237   1.500   2.000
##
## Largest/Smallest: 2
## Sample Skewness: 1.18
##
## Estimated Lambda: -2

## Box-Cox Transformation
##
## 47249 data points used to estimate Lambda
##
## Input data summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   2.000   4.839   5.000 431.000
##
## Largest/Smallest: 431
## Sample Skewness: 7.23
##
## Estimated Lambda: -0.6

## Box-Cox Transformation
##
## 47249 data points used to estimate Lambda
##
## Input data summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   5.402   1.000 362.000
##
## Largest/Smallest: 362
## Sample Skewness: 8.41
##
## Estimated Lambda: -2

## Box-Cox Transformation
##
## 47249 data points used to estimate Lambda
##
## Input data summary:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##        1        1        1   268862        1 30074518
##
## Largest/Smallest: 30100000
## Sample Skewness: 9.22
##
## Estimated Lambda: -0.4

## Box-Cox Transformation
##
## 47249 data points used to estimate Lambda
```

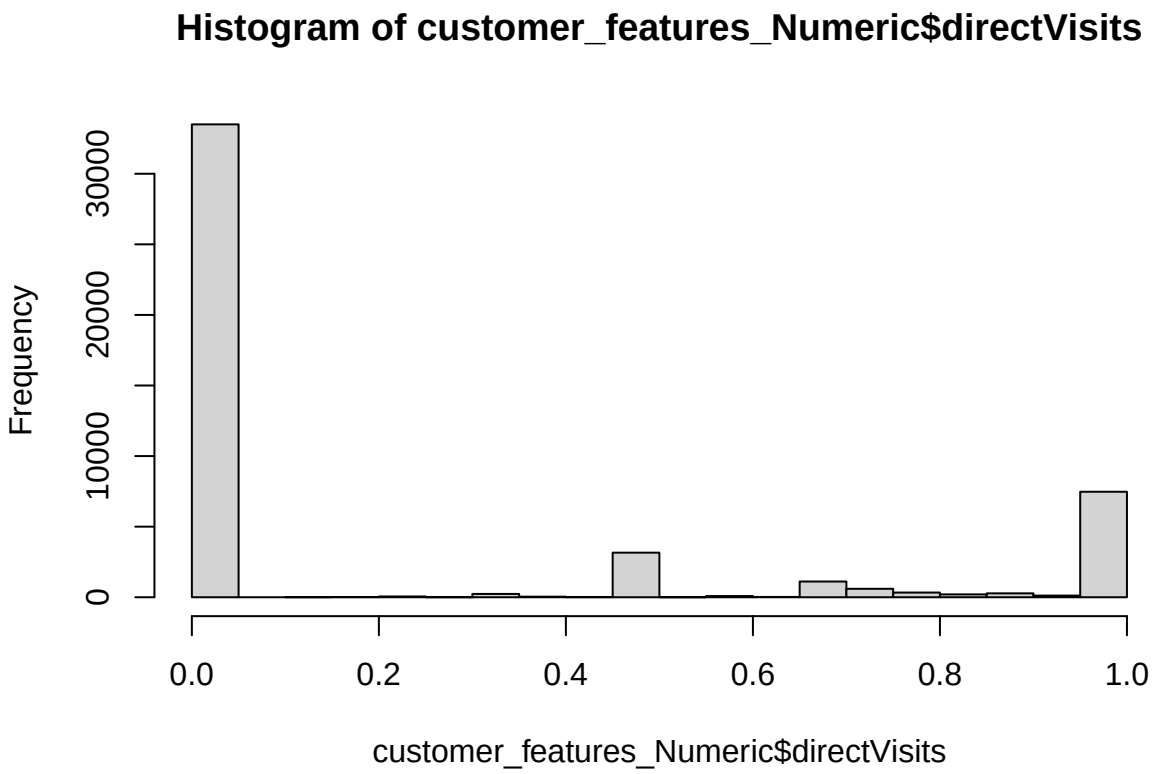
```
##
## Input data summary:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##        1        1        1   100654        1 15037260
##
## Largest/Smallest: 1.5e+07
## Sample Skewness: 10.6
##
## Estimated Lambda: -0.5

## Box-Cox Transformation
##
## 47249 data points used to estimate Lambda
##
## Input data summary:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    1.000    1.000    2.000    9.349    6.000 1496.000
##
## Largest/Smallest: 1500
## Sample Skewness: 13.5
##
## Estimated Lambda: -0.5

## Box-Cox Transformation
##
## 47249 data points used to estimate Lambda
##
## Input data summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   1.318   1.000  88.000
##
## Largest/Smallest: 88
## Sample Skewness: 17
##
## Estimated Lambda: -2

## Box-Cox Transformation
##
## 47249 data points used to estimate Lambda
##
## Input data summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   1.555   1.000 155.000
##
## Largest/Smallest: 155
## Sample Skewness: 23.4
##
## Estimated Lambda: -2

## Box-Cox Transformation
##
## 47249 data points used to estimate Lambda
##
```

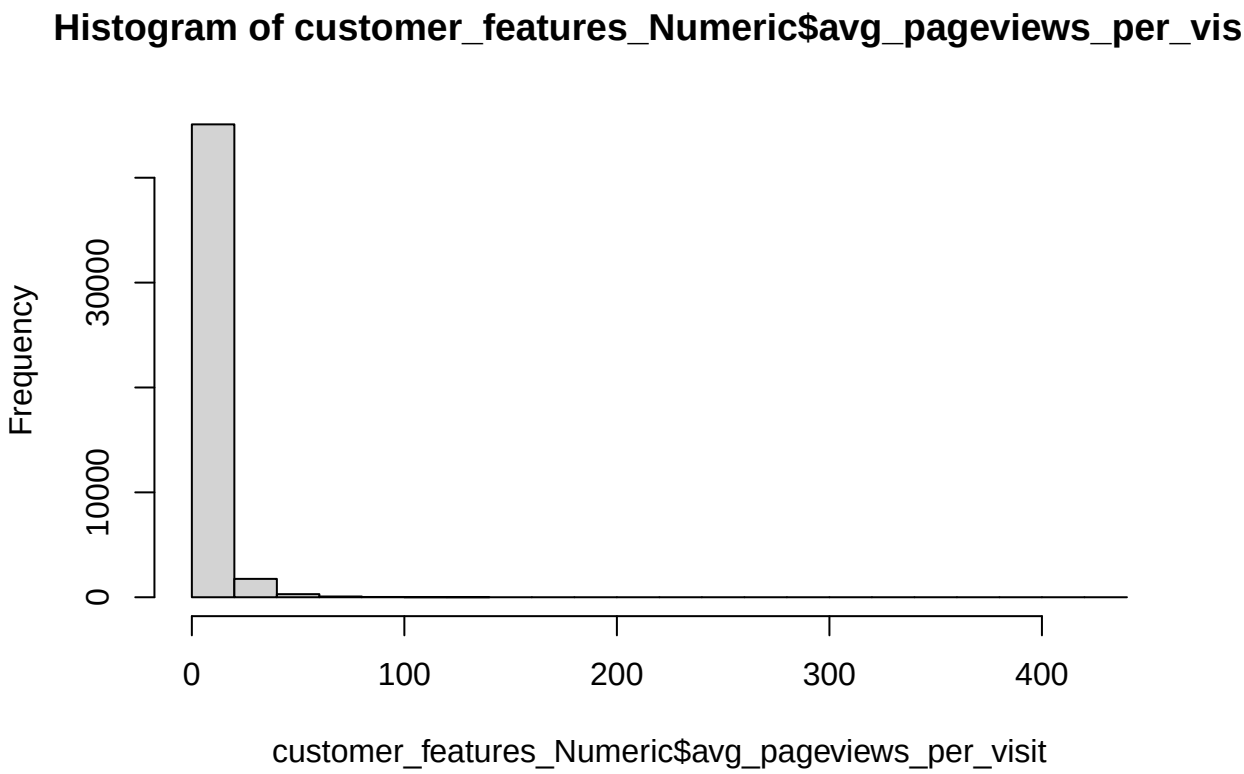
```
## Input data summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   1.314   1.000  99.000
##
## Largest/Smallest: 99
## Sample Skewness: 25.2
##
## Estimated Lambda: -2

## Box-Cox Transformation
##
## 47249 data points used to estimate Lambda
##
## Input data summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   1.483   1.000 155.000
##
## Largest/Smallest: 155
## Sample Skewness: 26.5
##
## Estimated Lambda: -2
```

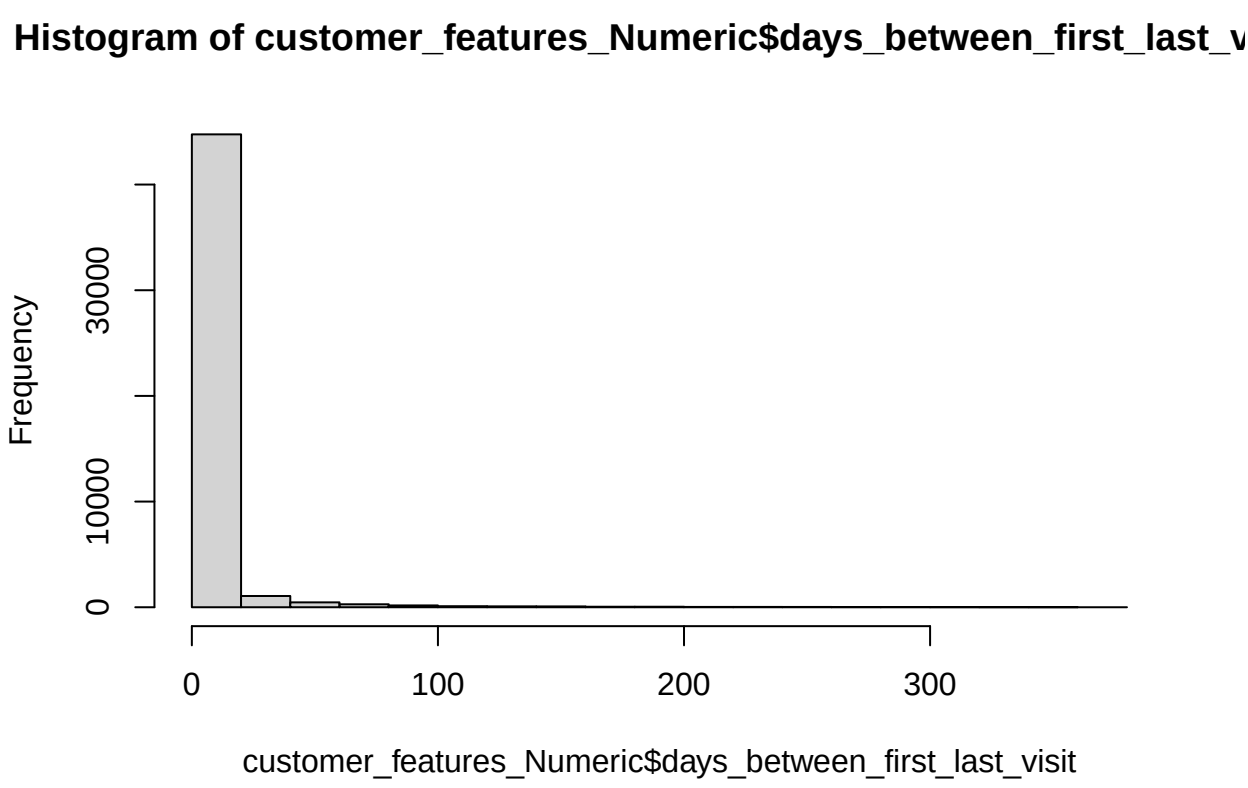
Histograms visualize the distribution of each numeric feature before transformation.

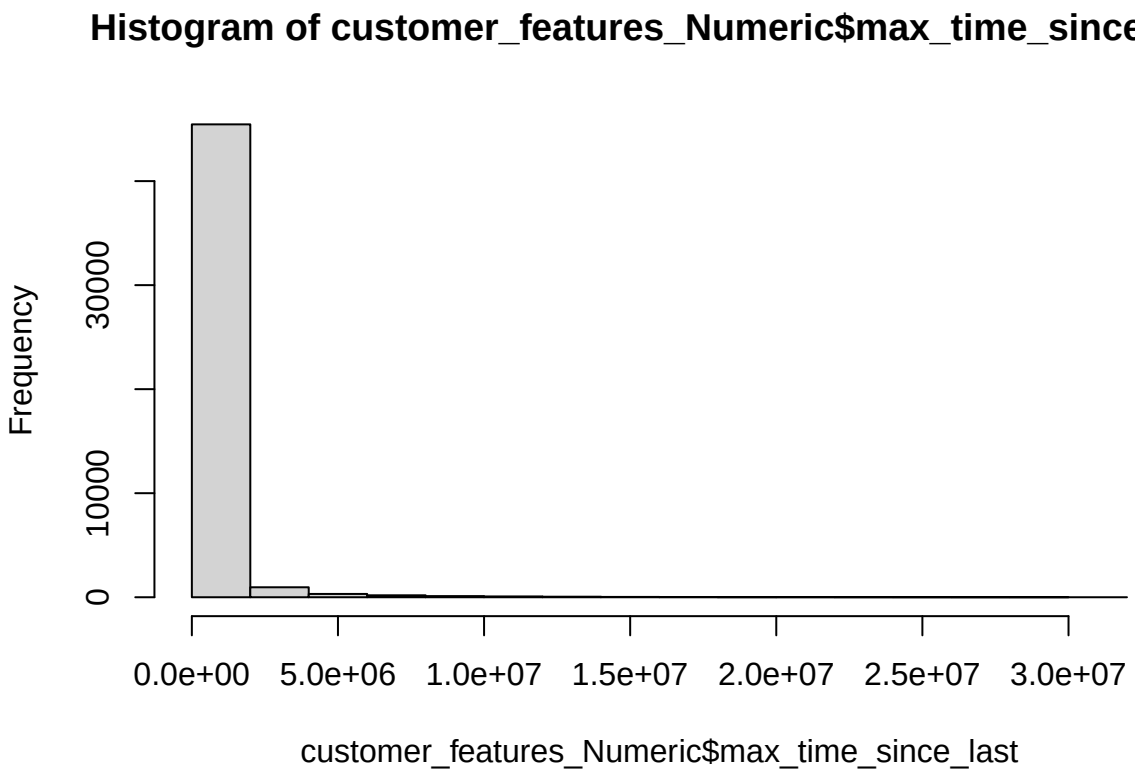
The purpose is to inspect skewness—whether the distribution is symmetric or skewed to the left/right—and assess the need for transformations.

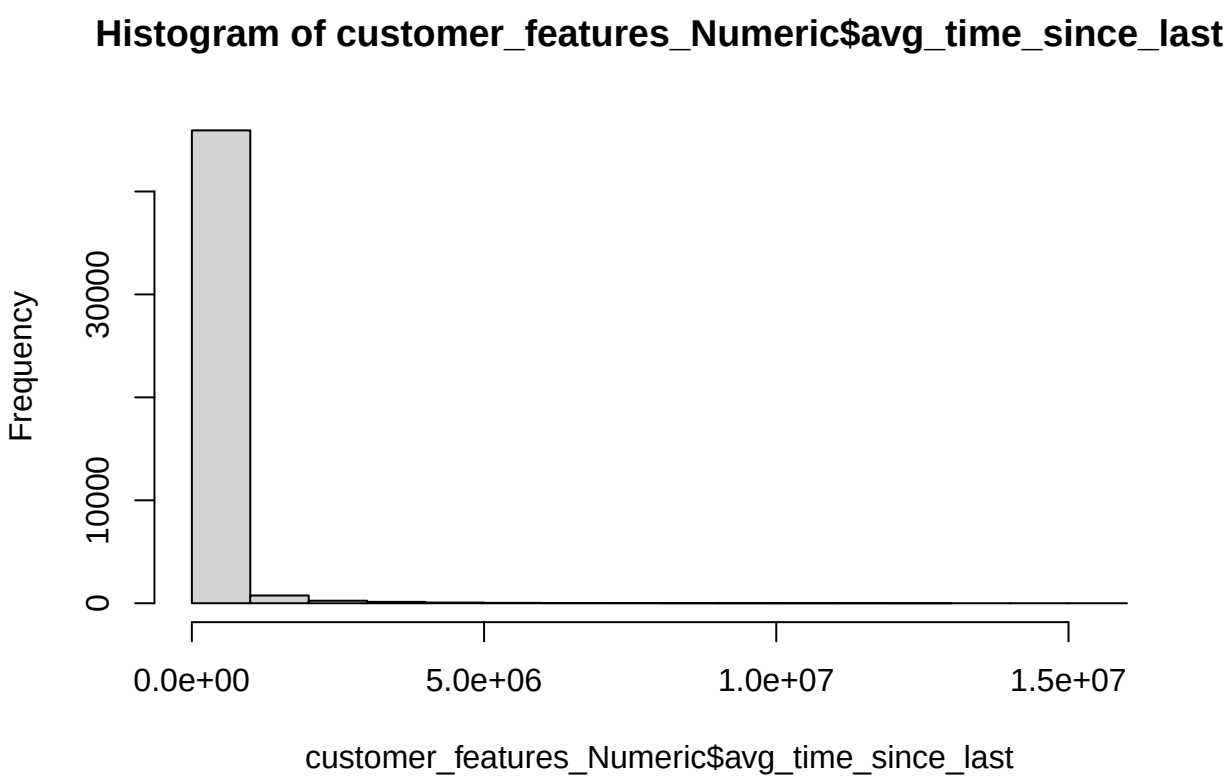

Histogram of customer_features_Numeric$mobileUsage
Frequency
0
10000
20000
30000
0.0
0.2
0.4
0.6
0.8
1.0
customer_features_Numeric$mobileUsage

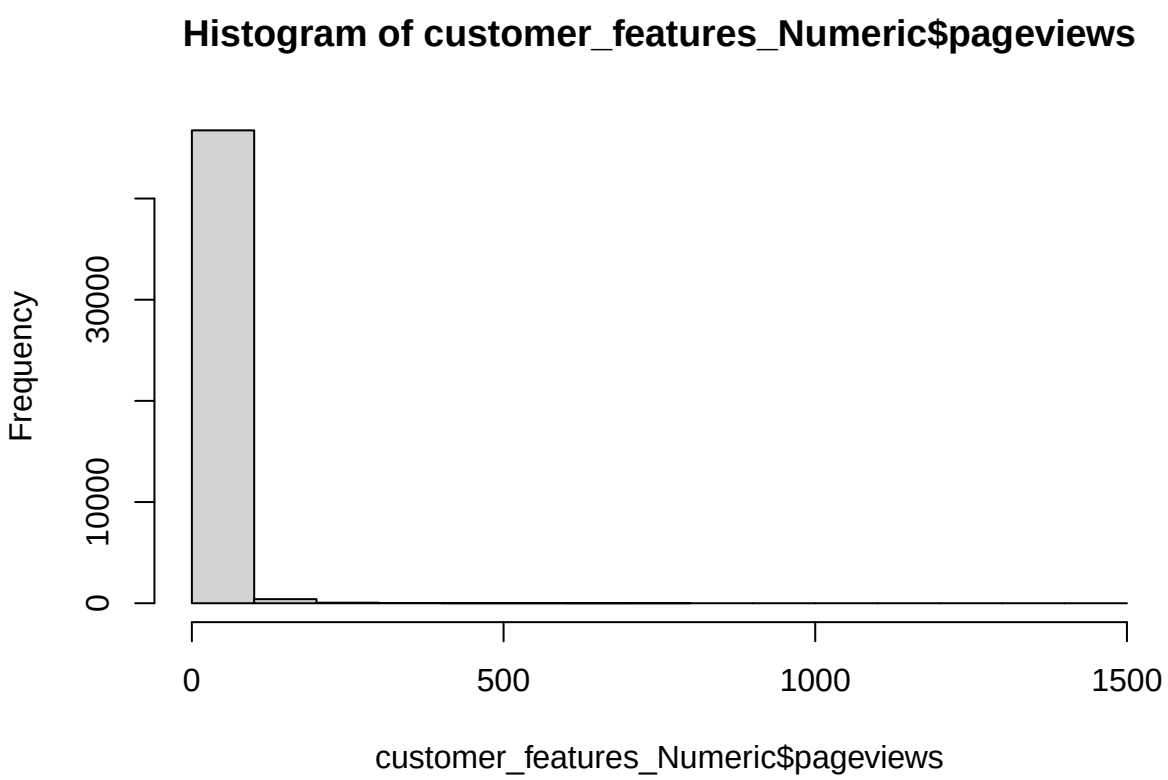

Histogram of customer_features_Numeric$directVisits
Frequency
0
10000
20000
30000
0.0
0.2
0.4
0.6
0.8
1.0
customer_features_Numeric$directVisits

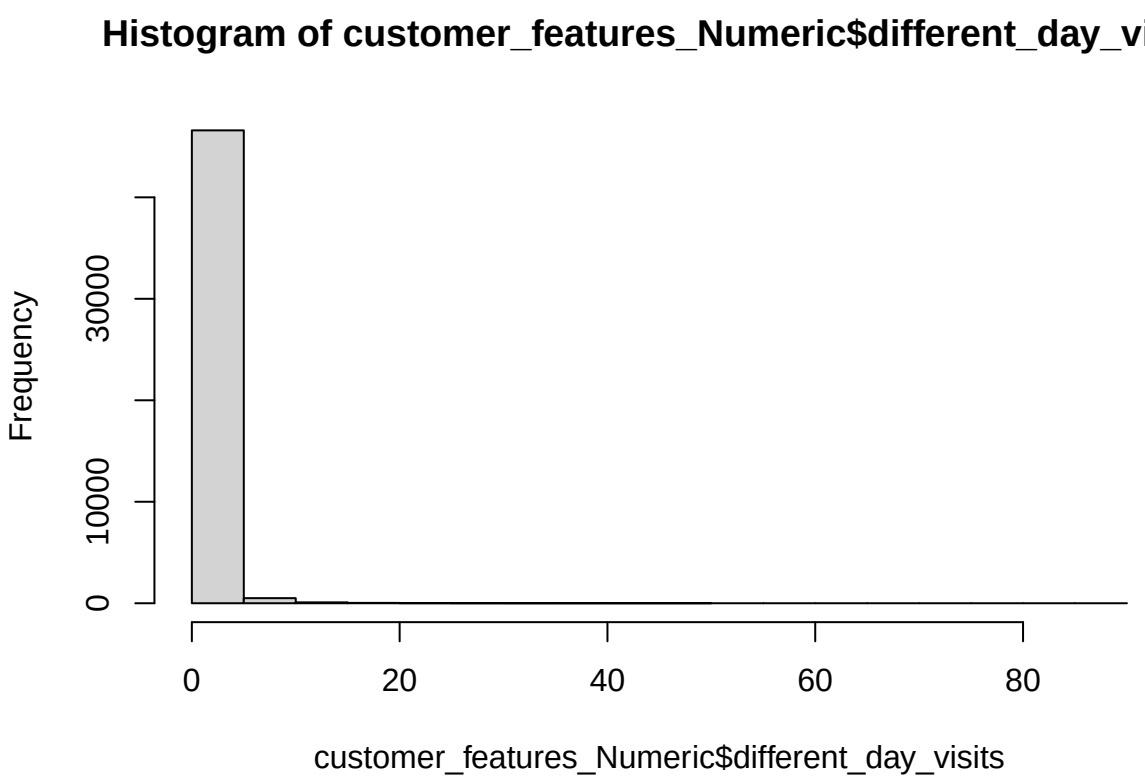

Histogram of customer_features_Numeric$avg_pageviews_per_vis
Frequency
0
10000
30000
0
100
200
300
400
customer_features_Numeric$avg_pageviews_per_visit

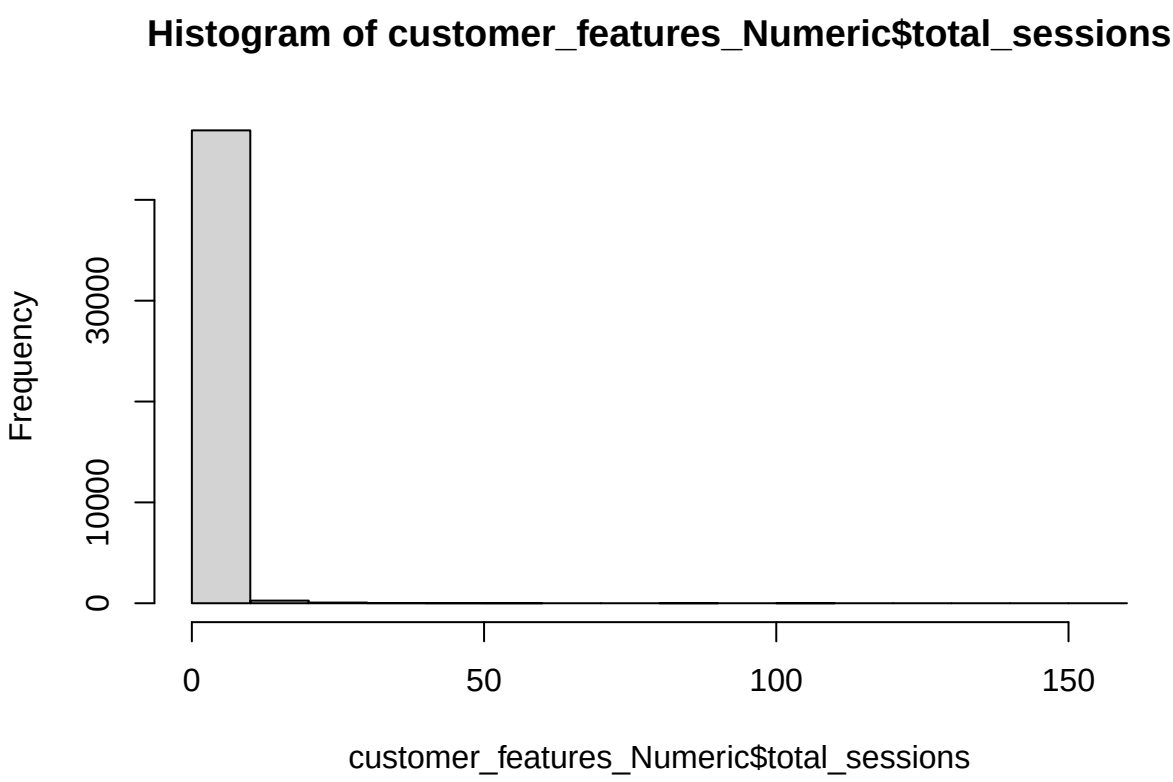

Histogram of customer_features_Numeric$days_between_first_last_v
Frequency
0
10000
30000
0
100
200
300
customer_features_Numeric$days_between_first_last_visit

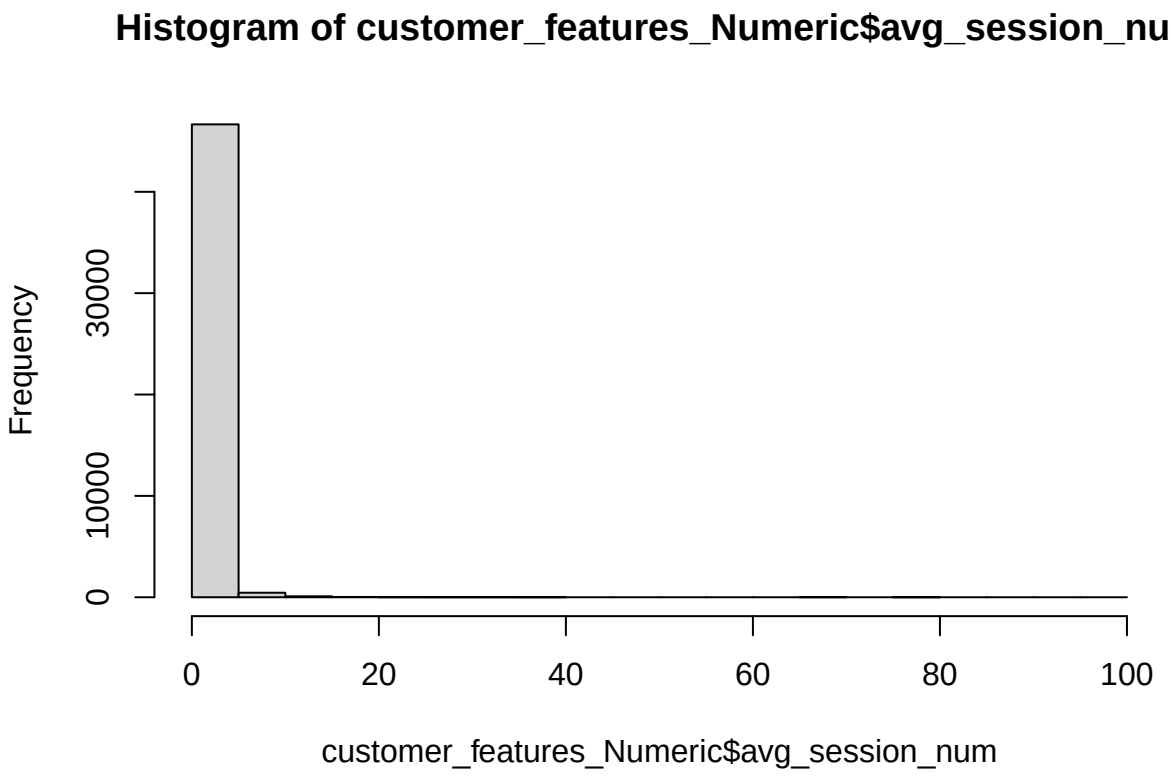

Histogram of customer_features_Numeric$max_time_since_last
Frequency
0
10000
30000
0.0e+00
5.0e+06
1.0e+07
1.5e+07
2.0e+07
2.5e+07
3.0e+07
customer_features_Numeric$max_time_since_last

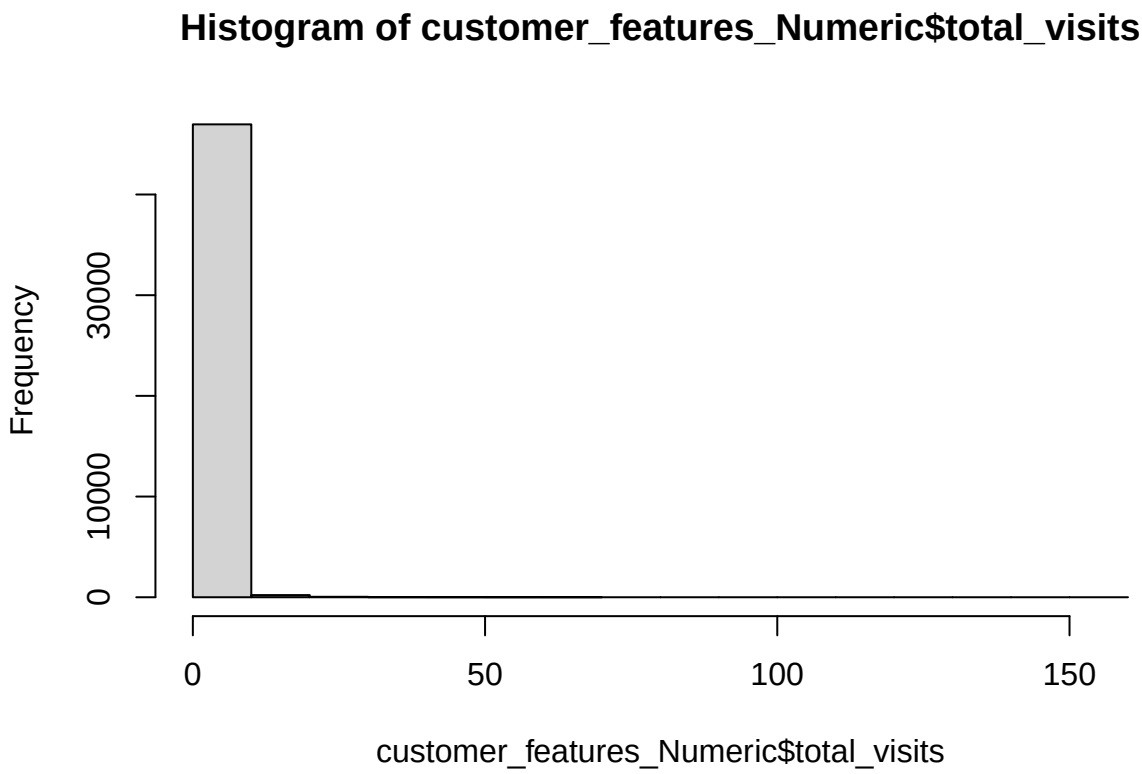

Histogram of customer_features_Numeric$avg_time_since_last
Frequency
0
10000
30000
0.0e+00
5.0e+06
1.0e+07
1.5e+07
customer_features_Numeric$avg_time_since_last


Histogram of customer_features_Numeric$pageviews
Frequency
0
10000
30000
0
500
1000
1500
customer_features_Numeric$pageviews


Histogram of customer_features_Numeric$different_day_visits
Frequency
0
10000
30000
0
20
40
60
80
customer_features_Numeric$different_day_visits


Histogram of customer_features_Numeric$total_sessions
Frequency
0
10000
30000
0
50
100
150
customer_features_Numeric$total_sessions

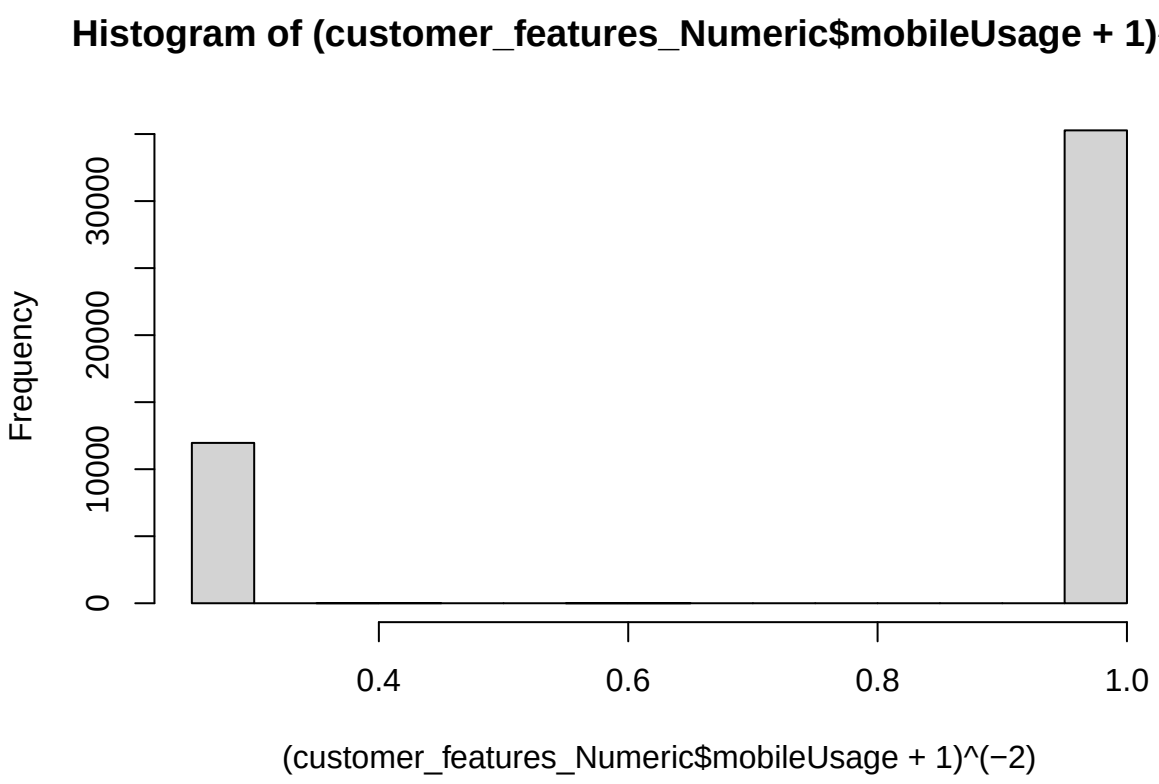

Histogram of customer_features_Numeric$avg_session_num
Frequency
0
10000
30000
0
20
40
60
80
100
customer_features_Numeric$avg_session_num

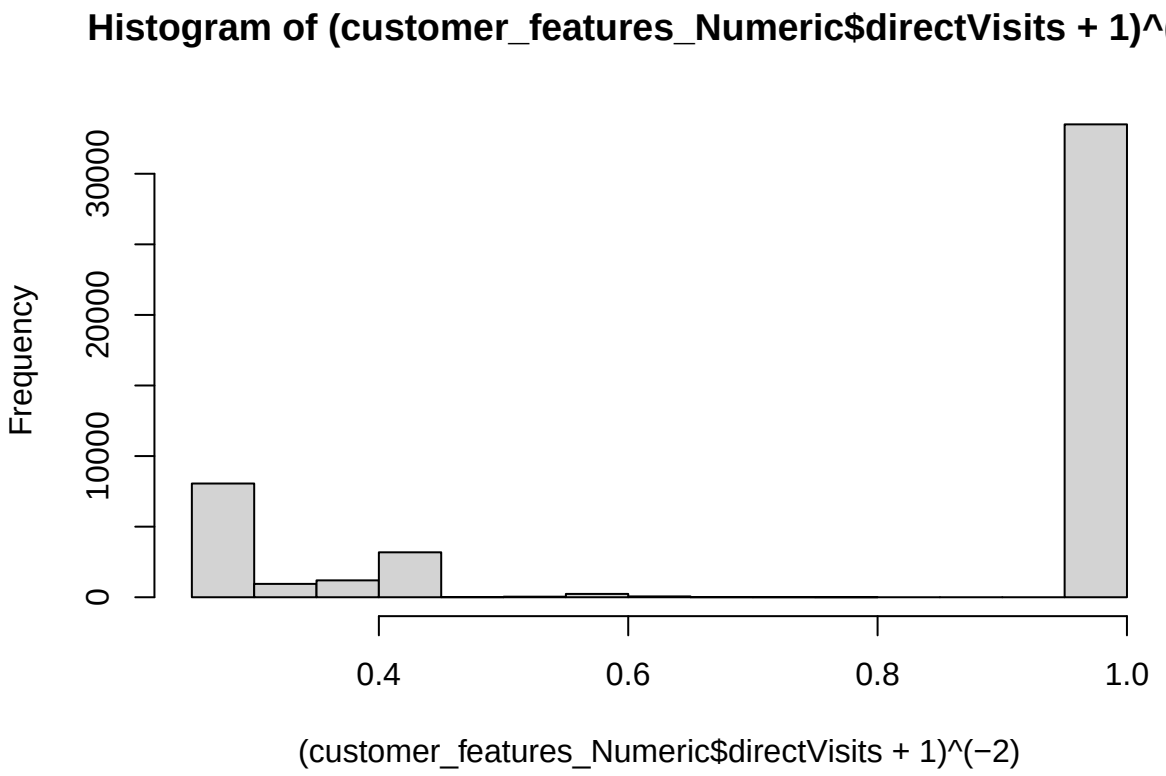


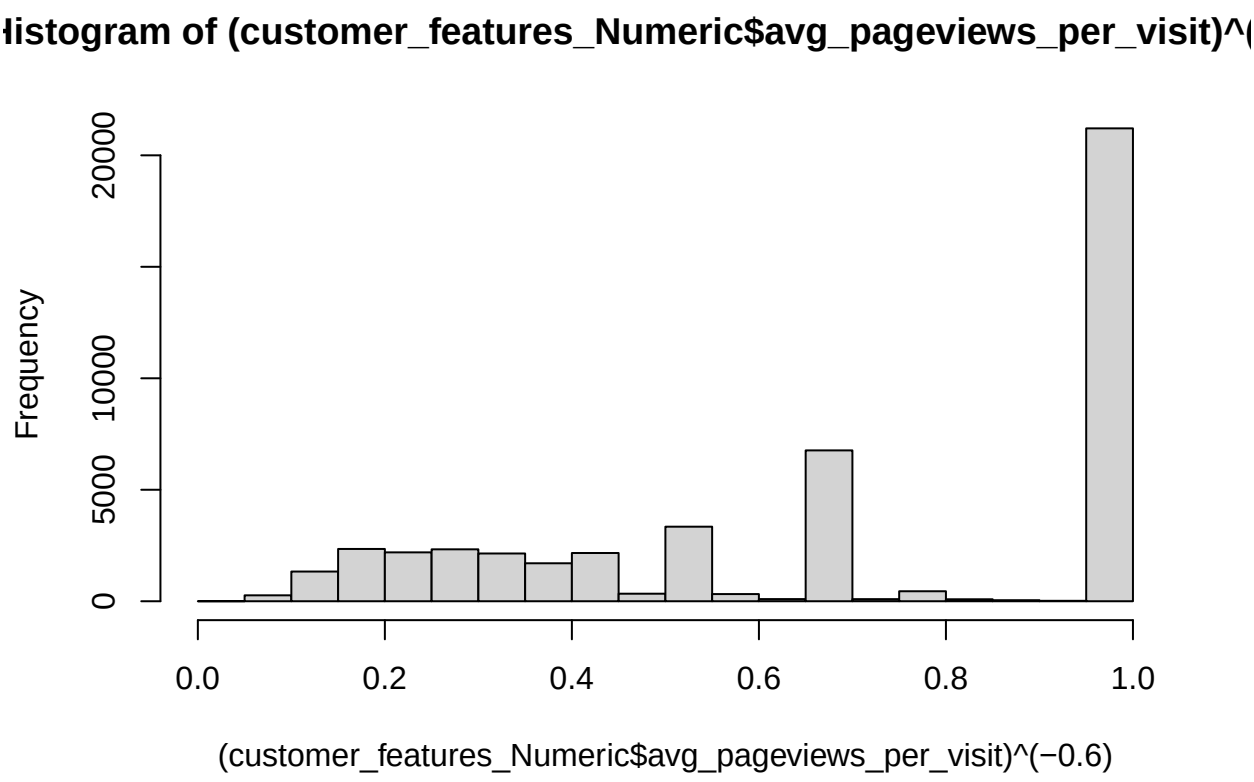
Histograms visualize the distribution of each numeric feature after transformation.

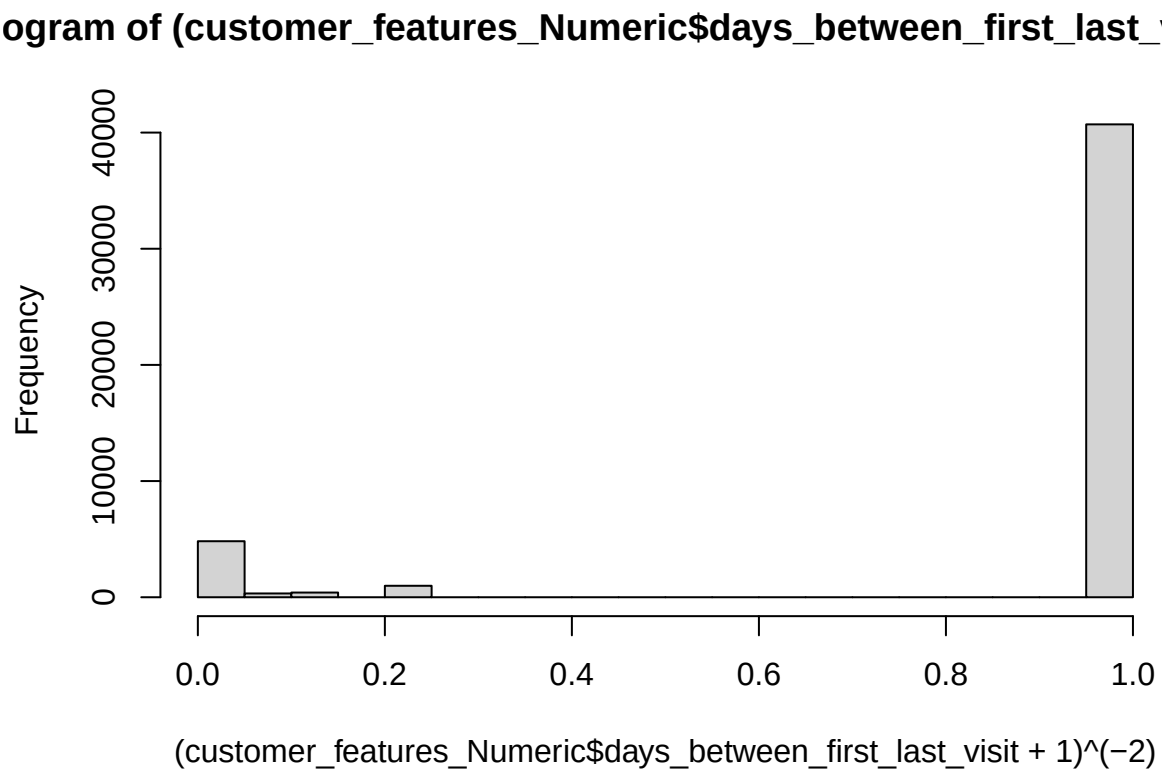
Power transformations (e.g., raising variables to negative exponents) are applied to address skewness further.

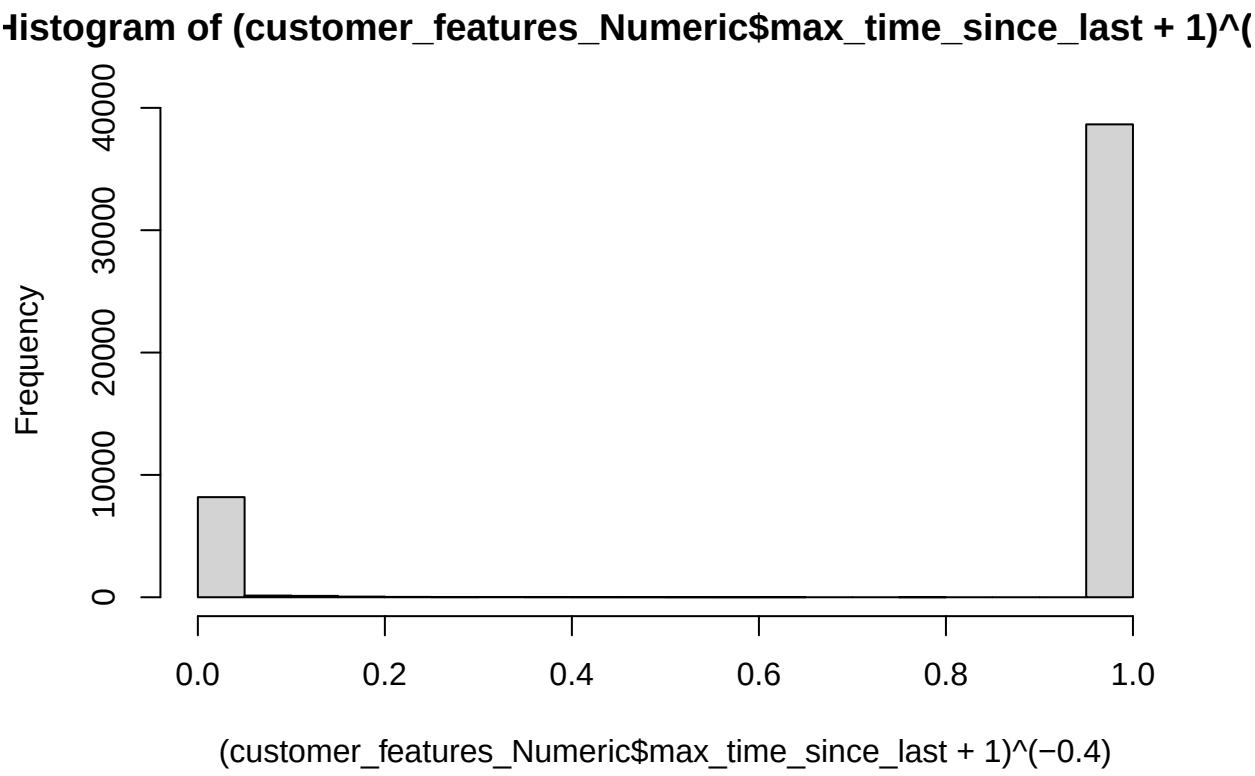
This operation reshapes distributions to be closer to normal. The transformed histograms help confirm if the transformation has successfully reduced skewness.

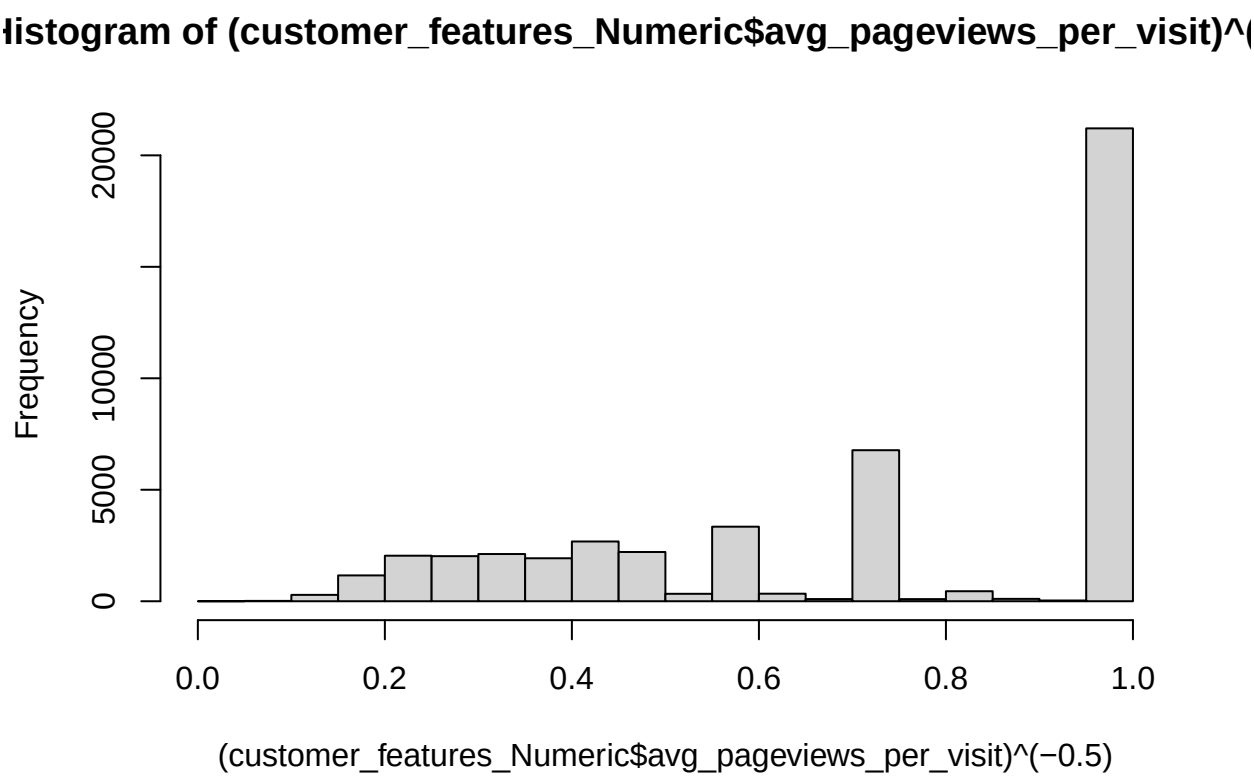

Histogram of (customer_features_Numeric$mobileUsage + 1)^(−2)
Frequency
0
10000
20000
30000
0.4
0.6
0.8
1.0
(customer_features_Numeric$mobileUsage + 1)^(−2)

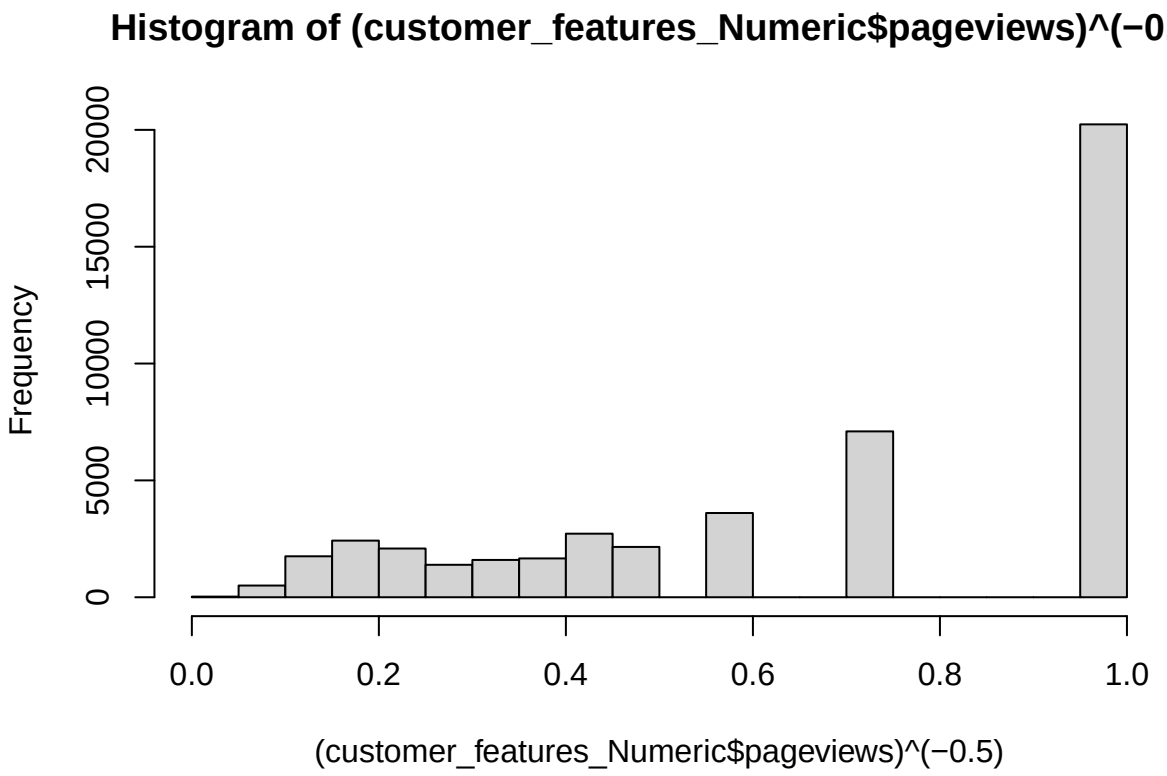

Histogram of (customer_features_Numeric$directVisits + 1)^(−2)
Frequency
0
10000
20000
30000
0.4
0.6
0.8
1.0
(customer_features_Numeric$directVisits + 1)^(−2)

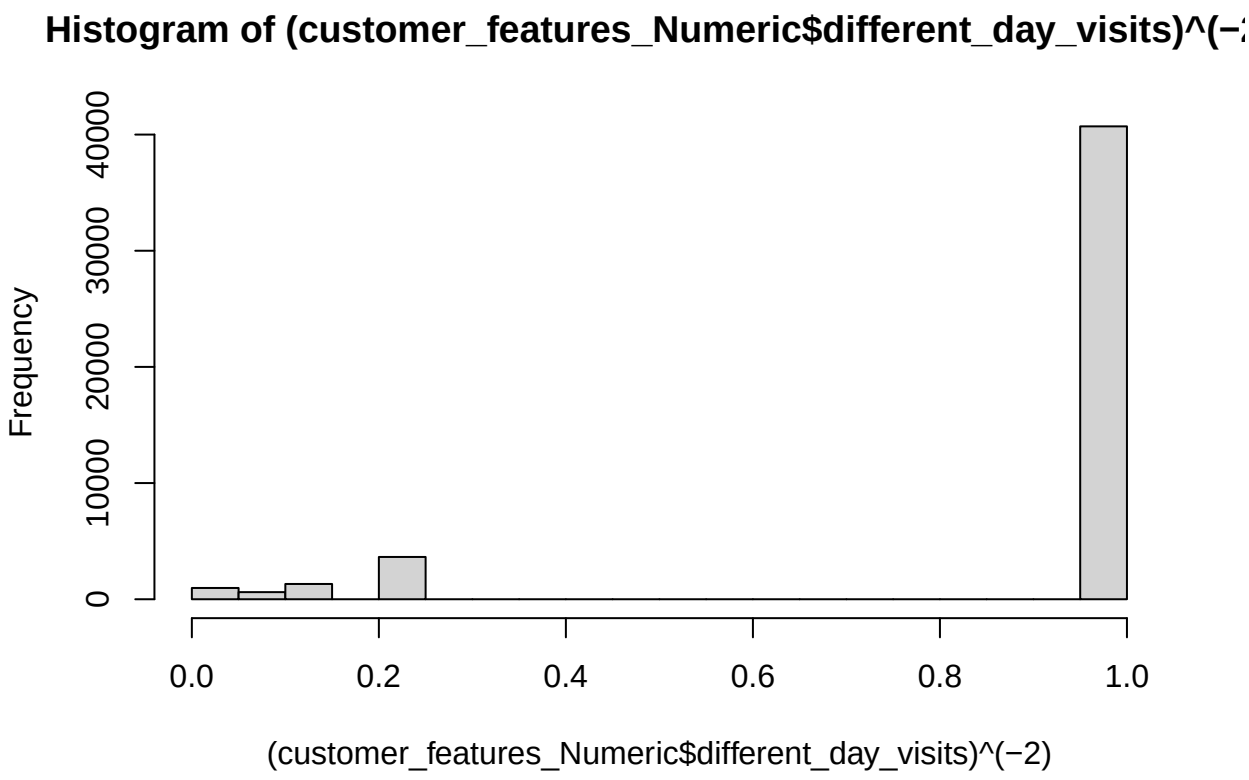

Histogram of (customer_features_Numeric$avg_pageviews_per_visit)^(
Frequency
0
5000
10000
20000
0.0
0.2
0.4
0.6
0.8
1.0
(customer_features_Numeric$avg_pageviews_per_visit)^(−0.6)

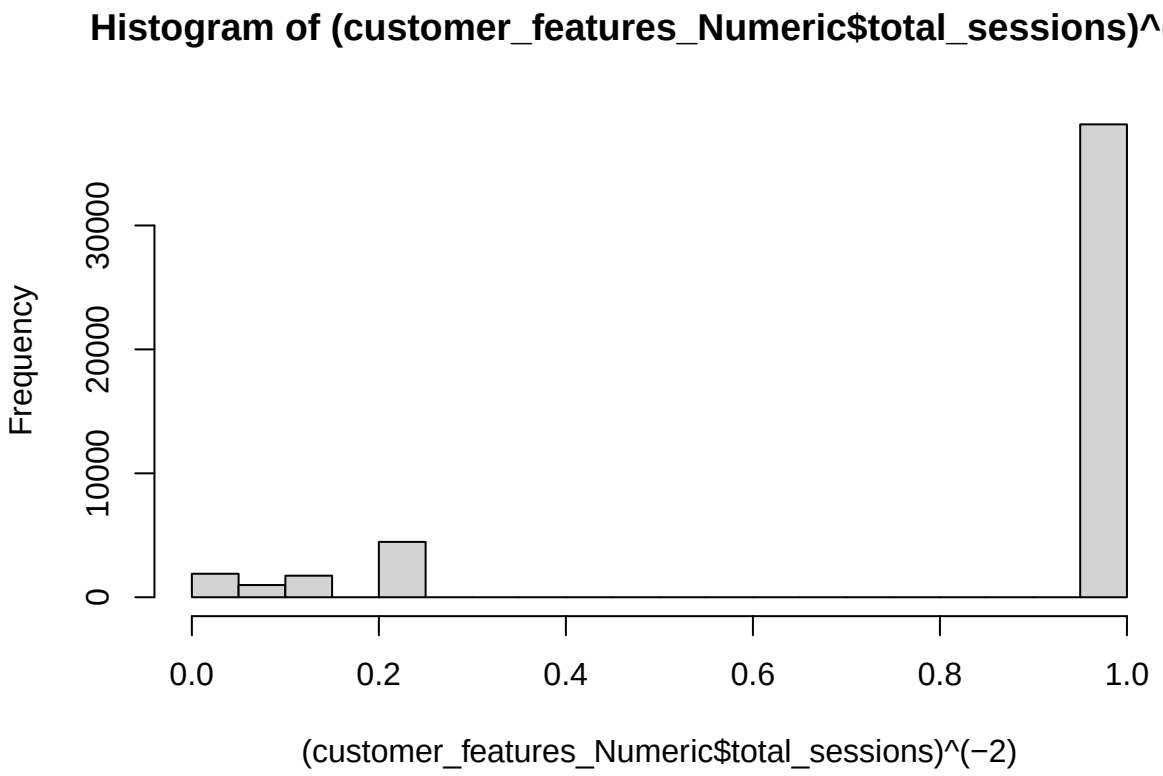

ogram of (customer_features_Numeric$days_between_first_last_visit ·
Frequency
0
10000
20000
30000
40000
0.0
0.2
0.4
0.6
0.8
1.0
(customer_features_Numeric$days_between_first_last_visit + 1)^(−2)

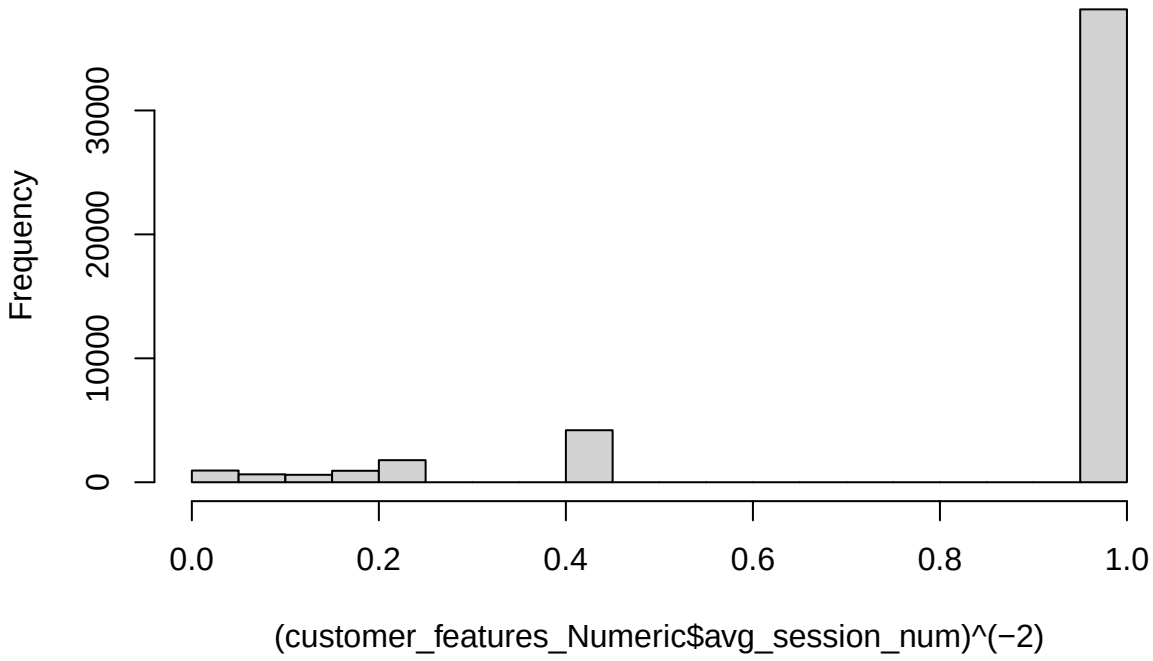

Histogram of (customer_features_Numeric$max_time_since_last + 1)^(
Frequency
0
10000
20000
30000
40000
0.0
0.2
0.4
0.6
0.8
1.0
(customer_features_Numeric$max_time_since_last + 1)^(−0.4)

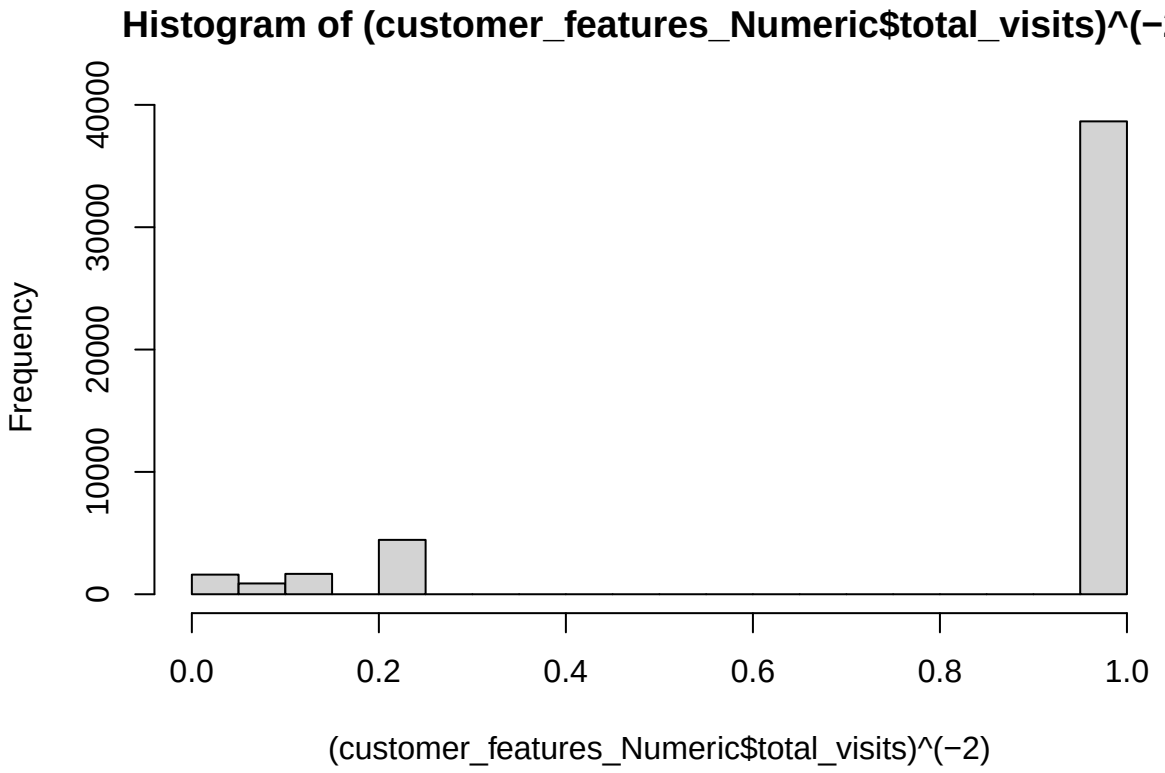

Histogram of (customer_features_Numeric$avg_pageviews_per_visit)^(
Frequency
0
5000
10000
20000
0.0
0.2
0.4
0.6
0.8
1.0
(customer_features_Numeric$avg_pageviews_per_visit)^(−0.5)


Histogram of (customer_features_Numeric$pageviews)^(−0.5)
Frequency
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
(customer_features_Numeric$pageviews)^(−0.5)


Histogram of (customer_features_Numeric$different_day_visits)^(−
Frequency
0
10000
20000
30000
40000
0.0
0.2
0.4
0.6
0.8
1.0
(customer_features_Numeric$different_day_visits)^(−2)


Histogram of (customer_features_Numeric$total_sessions)^(−2)
Frequency
0
10000
20000
30000
0.0
0.2
0.4
0.6
0.8
1.0
(customer_features_Numeric$total_sessions)^(−2)


Histogram of (customer_features_Numeric$avg_session_num)^(−2
Frequency
0
10000
20000
30000
0.0
0.2
0.4
0.6
0.8
1.0
(customer_features_Numeric$avg_session_num)^(−2)



The numeric dataset is updated in place with the transformed variables. These transformations are applied to variables like mobileUsage, pageviews, and total_visits.

Purpose: This ensures that future analyses or models will use the normalized data to improve model performance and interpretability.

Data Centering and Scaling

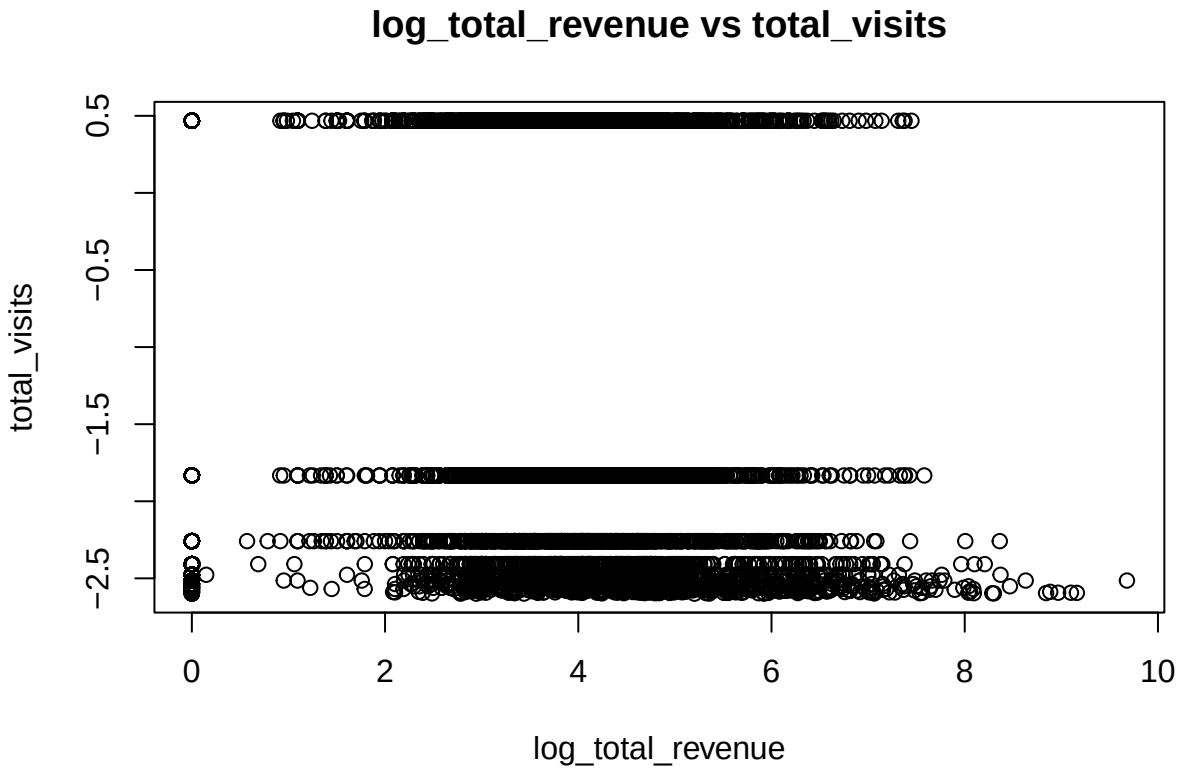
This process involves several steps to prepare and explore customer data for analysis. First, numeric data is centered and scaled to ensure that all features have zero mean and unit variance, which improves the performance of models sensitive to feature magnitude. Important features, like log_total_revenue, are added back to the transformed dataset, while some columns, such as directVisits, are removed to avoid redundancy.

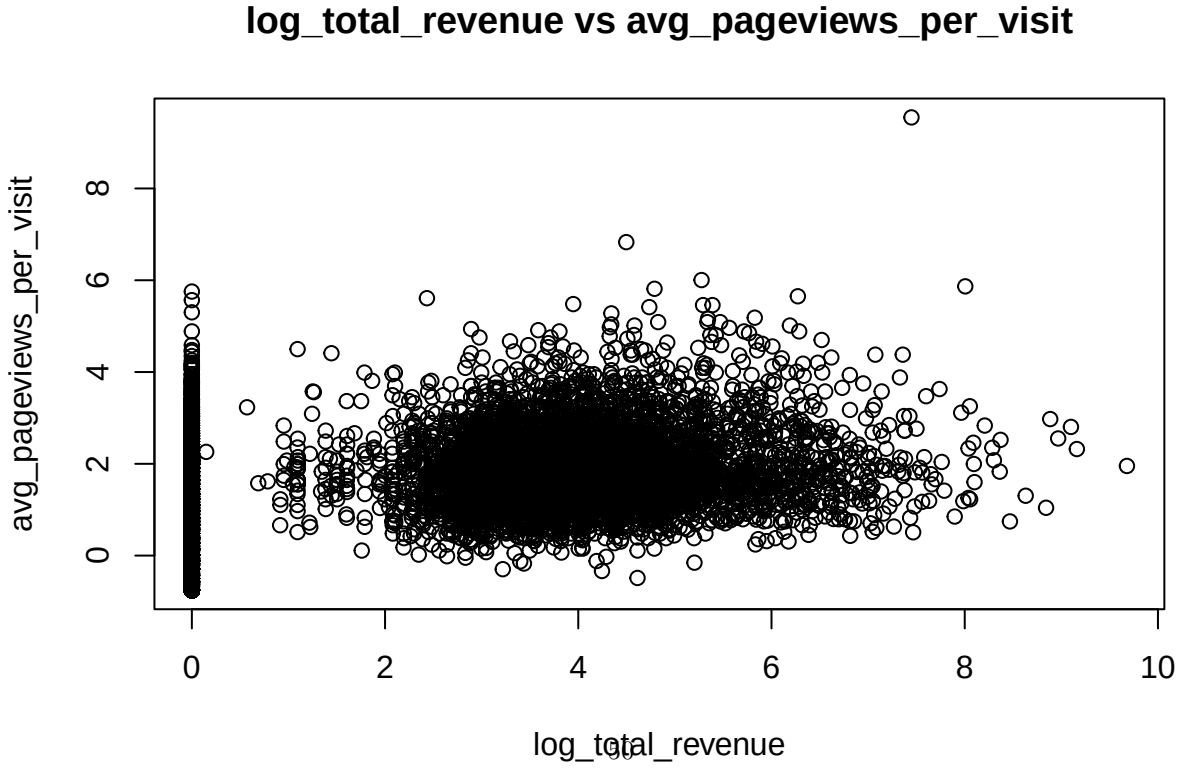
Next, character columns are converted into factors, making them compatible with models that handle categorical data. These transformed numeric and categorical datasets are combined to form a comprehensive dataset, ready for further analysis.

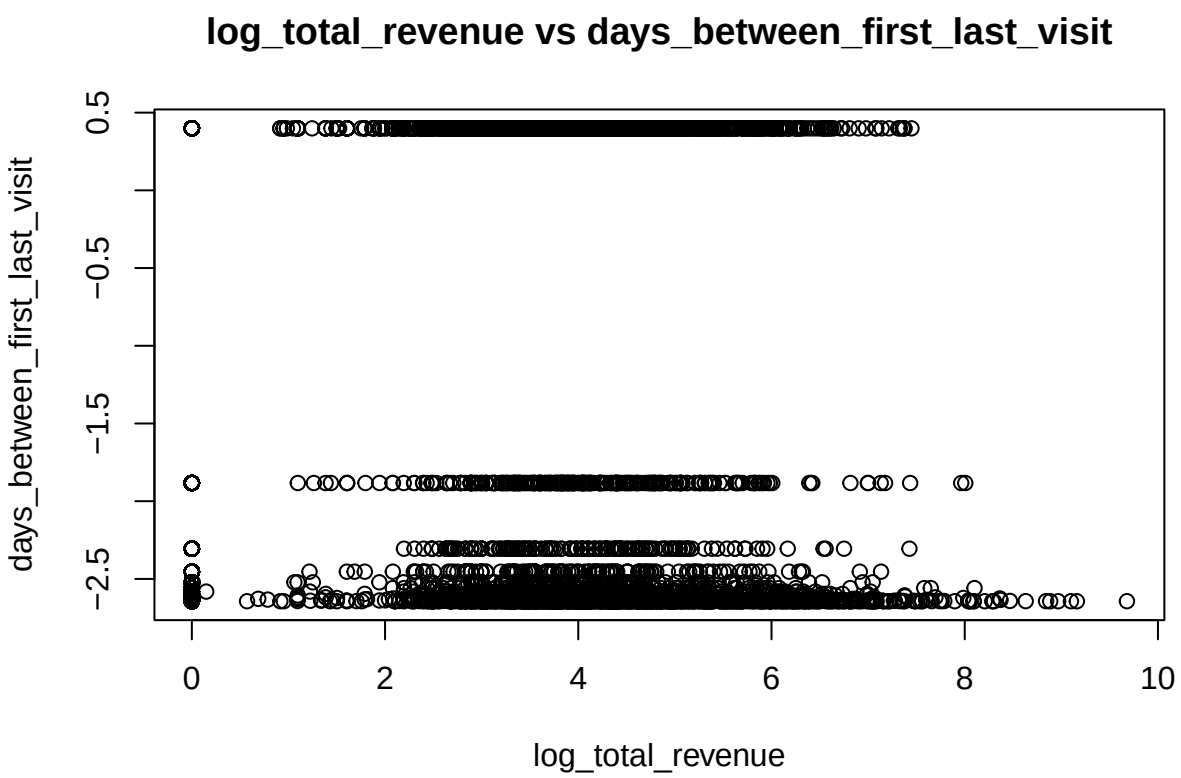
```
##  [1] "total_visits"                  "avg_pageviews_per_visit"
##  [3] "days_between_first_last_visit" "different_day_visits"
##  [5] "total_sessions"                "avg_session_num"
##  [7] "avg_time_since_last"           "max_time_since_last"
##  [9] "mobileUsage"                   "pageviews"
## [11] "log_total_revenue"             "mostCommonBrowser"
## [13] "mostCommonOS"                  "mostCommonDevice"
## [15] "primaryChannel"                "mostFrequentContinent"
```

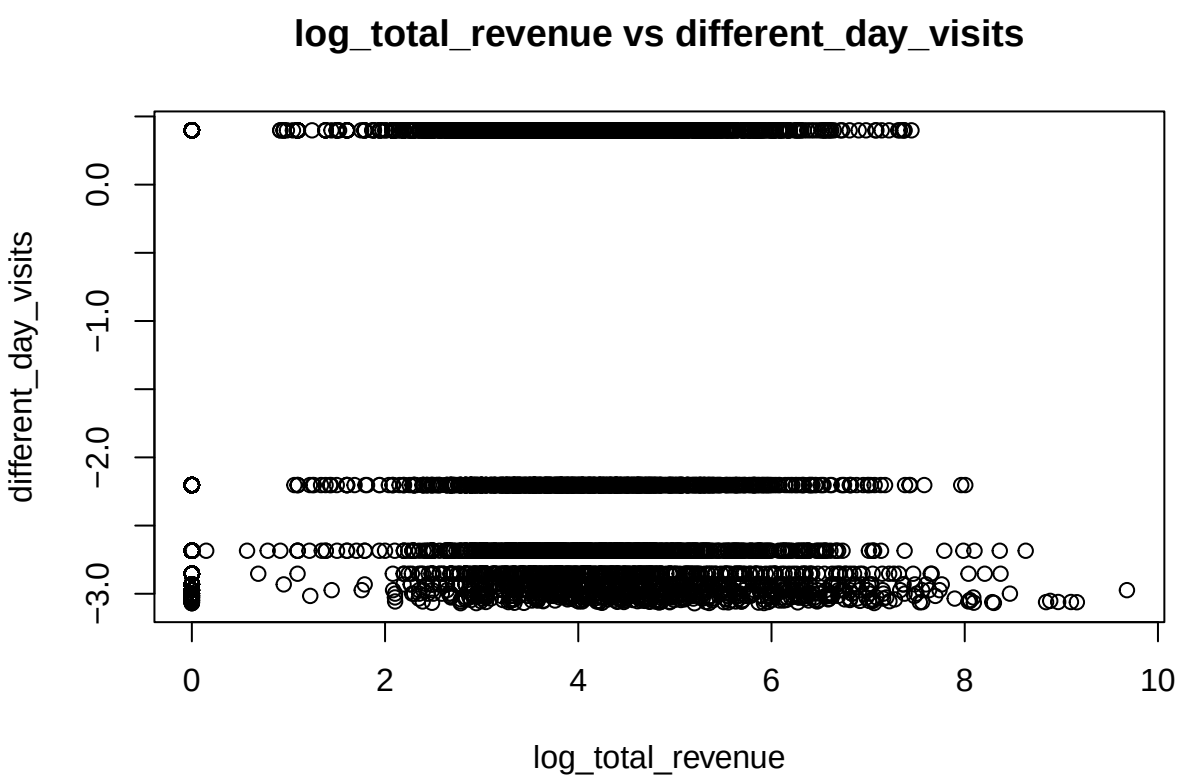
```
## [17] "mostFrequentCountry"           "primarySource"
## [19] "primaryMedium"                 "mostCommonDay"
```

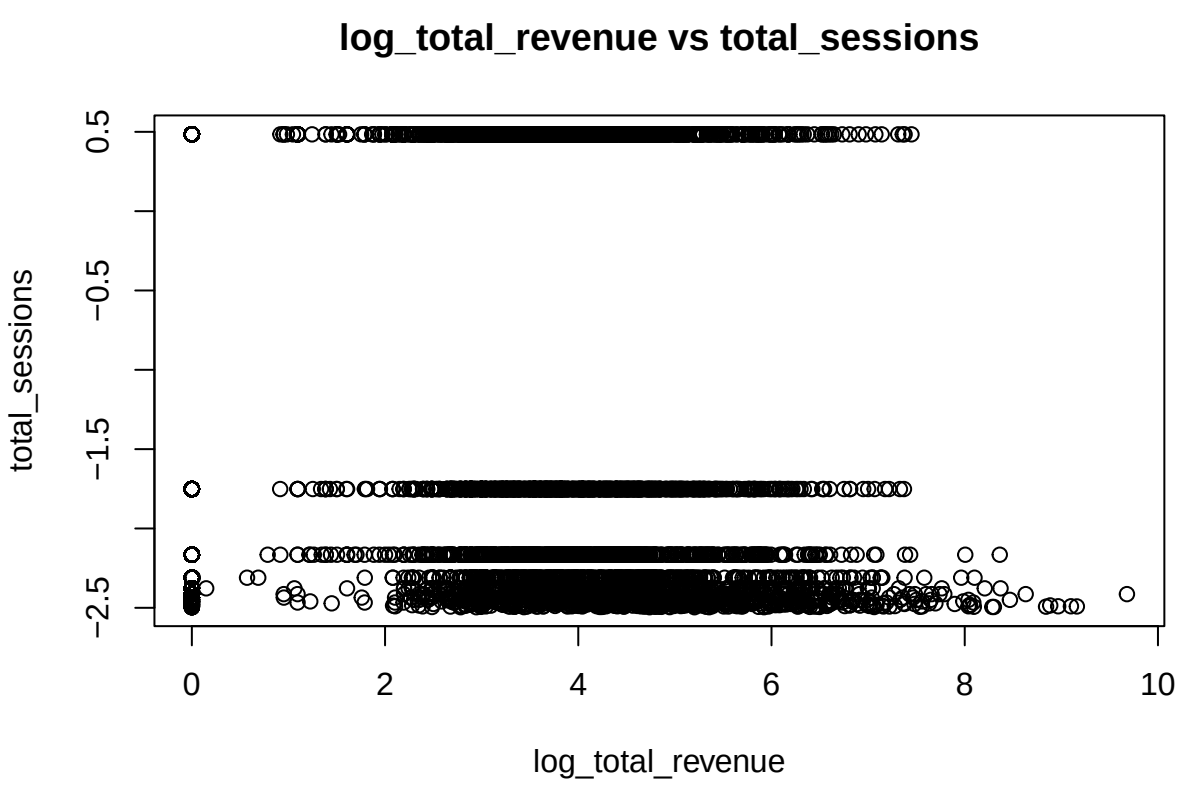
Scatter plots are used to explore relationships between log_total_revenue and various features, helping identify potential correlations or patterns between revenue and customer behavior metrics.

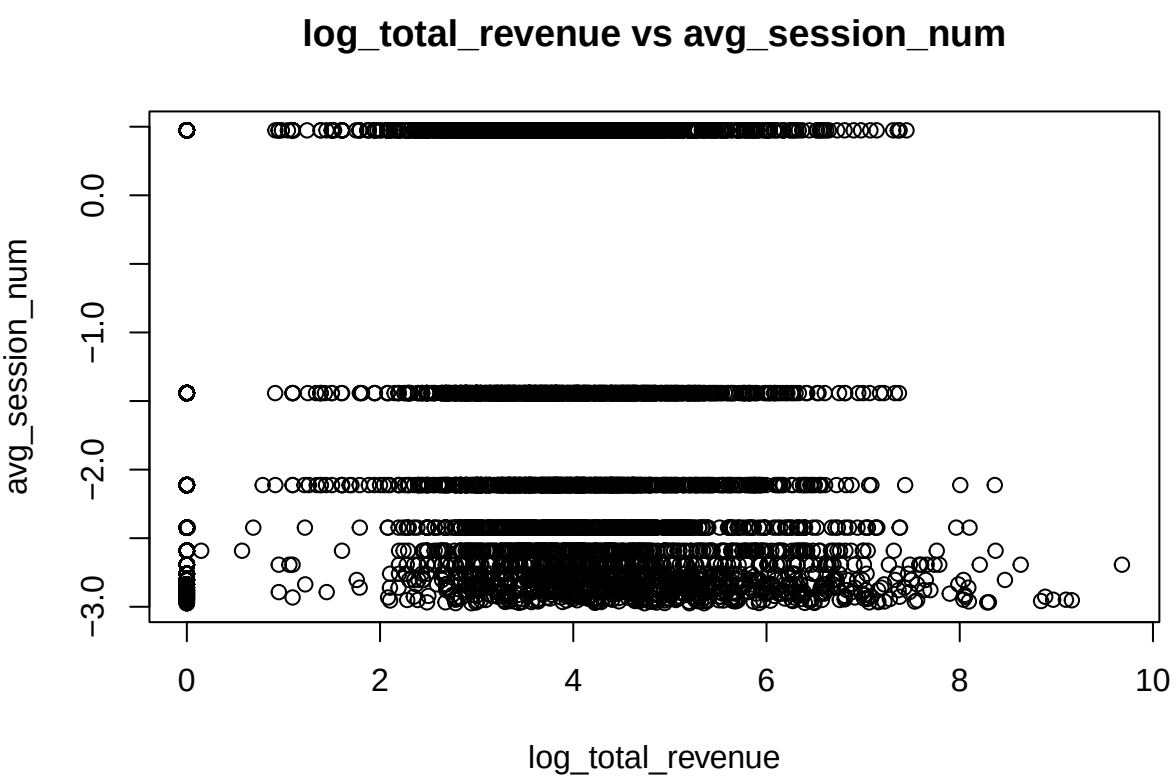


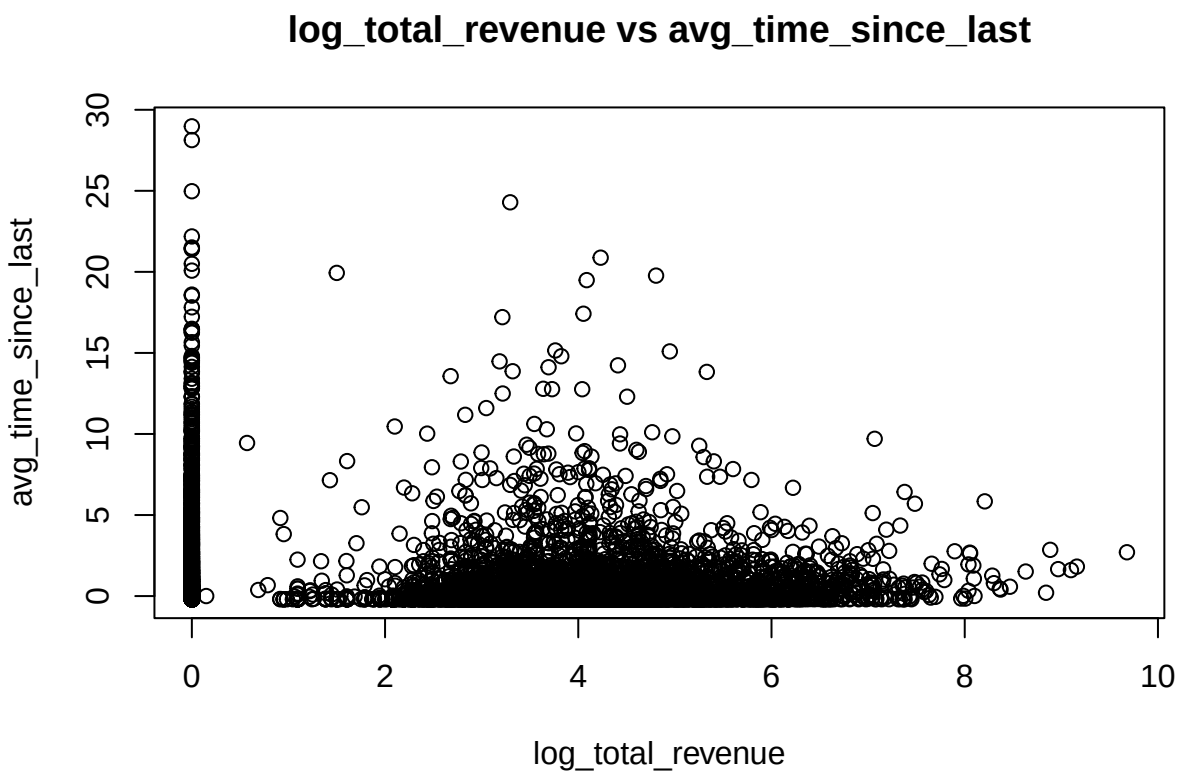


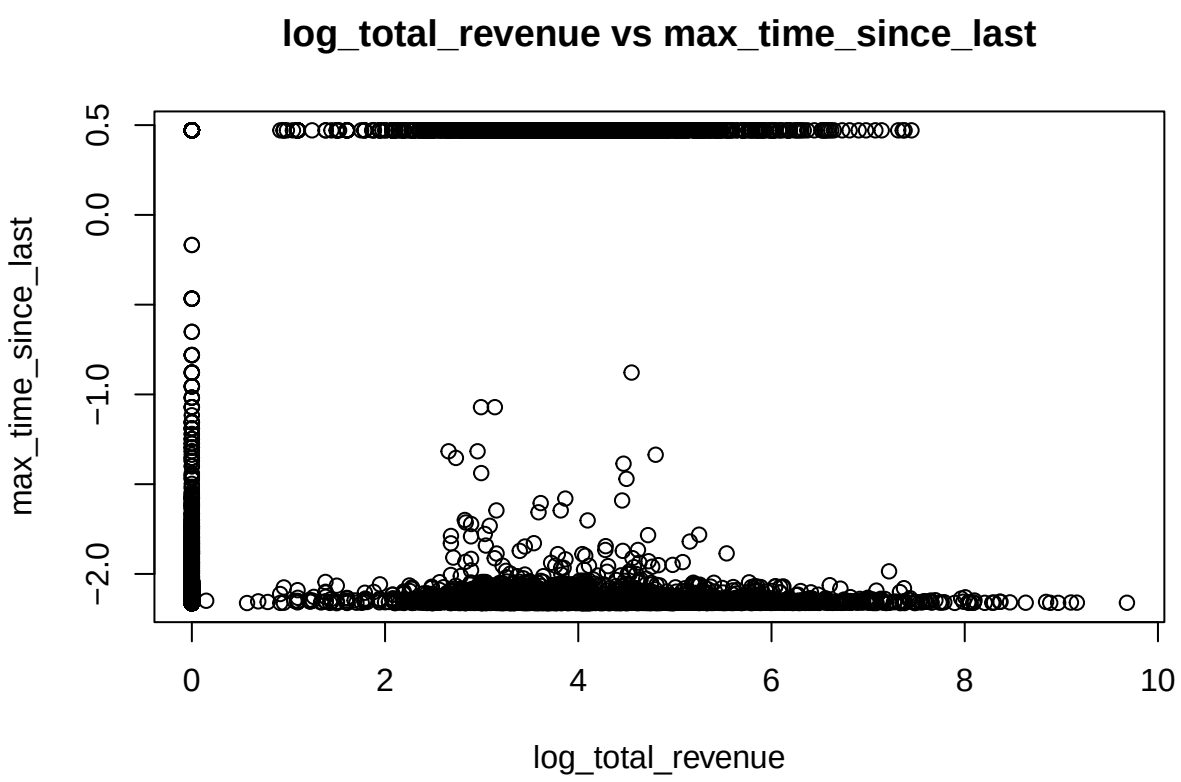

log_total_revenue vs days_between_first_last_visit
days_between_first_last_visit
0.5
−0.5
−1.5
−2.5
0
2
4
6
8
10
log_total_revenue

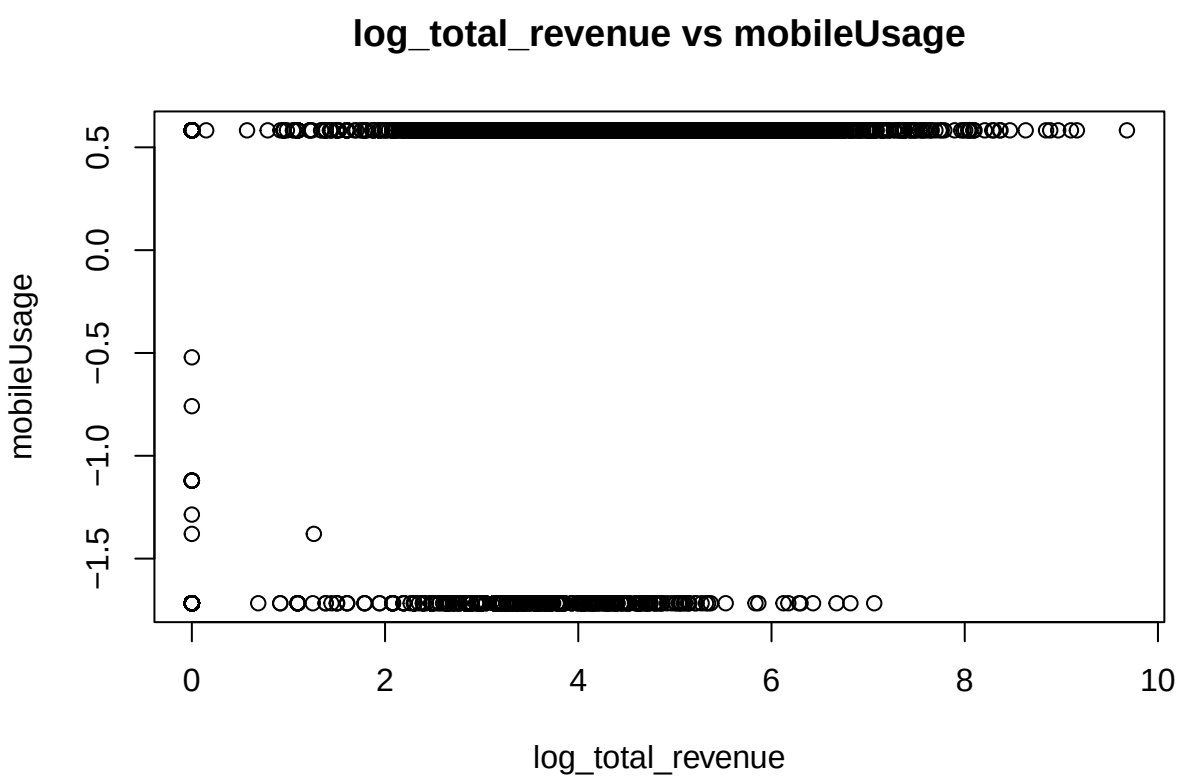

log_total_revenue vs different_day_visits
different_day_visits
0.0
−1.0
−2.0
−3.0
0
2
4
6
8
10
log_total_revenue

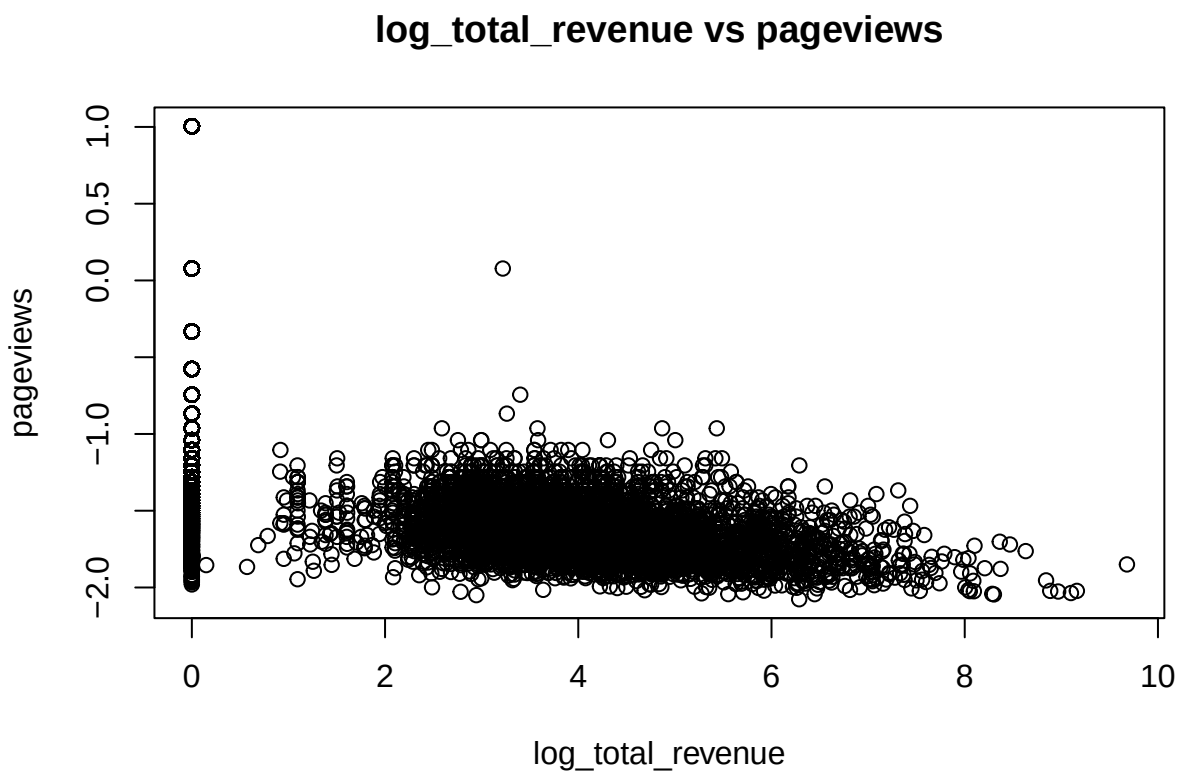

log_total_revenue vs total_sessions
total_sessions
0.5
−0.5
−1.5
−2.5
0
2
4
6
8
10
log_total_revenue


log_total_revenue vs avg_session_num
avg_session_num
0.0
−1.0
−2.0
−3.0
0
2
4
6
8
10
log_total_revenue


log_total_revenue vs avg_time_since_last
avg_time_since_last
0
5
10
15
20
25
30
0
2
4
6
8
10
log_total_revenue


log_total_revenue vs max_time_since_last
max_time_since_last
0.5
0.0
−1.0
−2.0
0
2
4
6
8
10
log_total_revenue


log_total_revenue vs mobileUsage
mobileUsage
0.5
0.0
−0.5
−1.0
−1.5
0
2
4
6
8
10
log_total_revenue



Finally, the dataset is checked for missing values to ensure it is complete and ready for modeling. This workflow ensures the data is well-prepared, clean, and properly structured, setting a solid foundation for future predictive analysis or machine learning tasks.

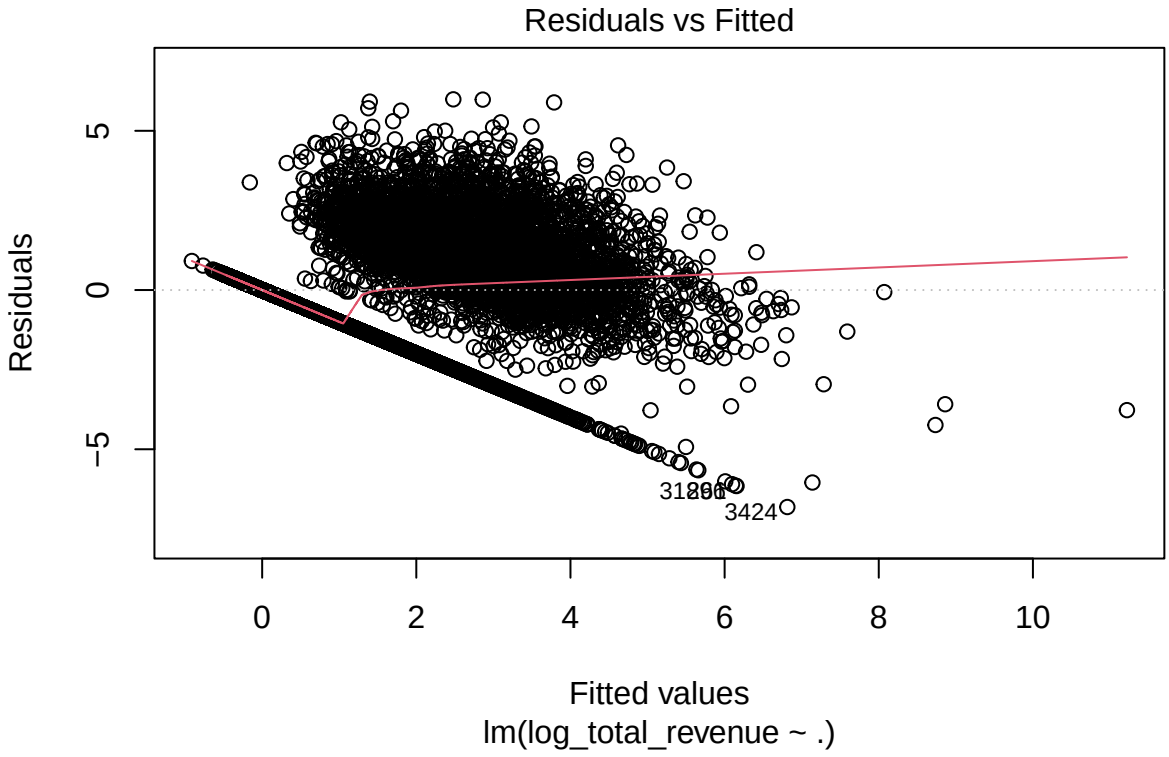
```
##                  total_visits       avg_pageviews_per_visit
##                             0                             0
## days_between_first_last_visit          different_day_visits
##                             0                             0
##                total_sessions               avg_session_num
##                             0                             0
##           avg_time_since_last           max_time_since_last
##                             0                             0
##                   mobileUsage                     pageviews
##                             0                             0
##             log_total_revenue            mostCommonBrowser
##                             0                             0
##                  mostCommonOS             mostCommonDevice
##                             0                             0
##                primaryChannel         mostFrequentContinent
##                             0                             0
##           mostFrequentCountry                 primarySource
##                             0                             0
##                 primaryMedium                 mostCommonDay
##                             0                             0
```

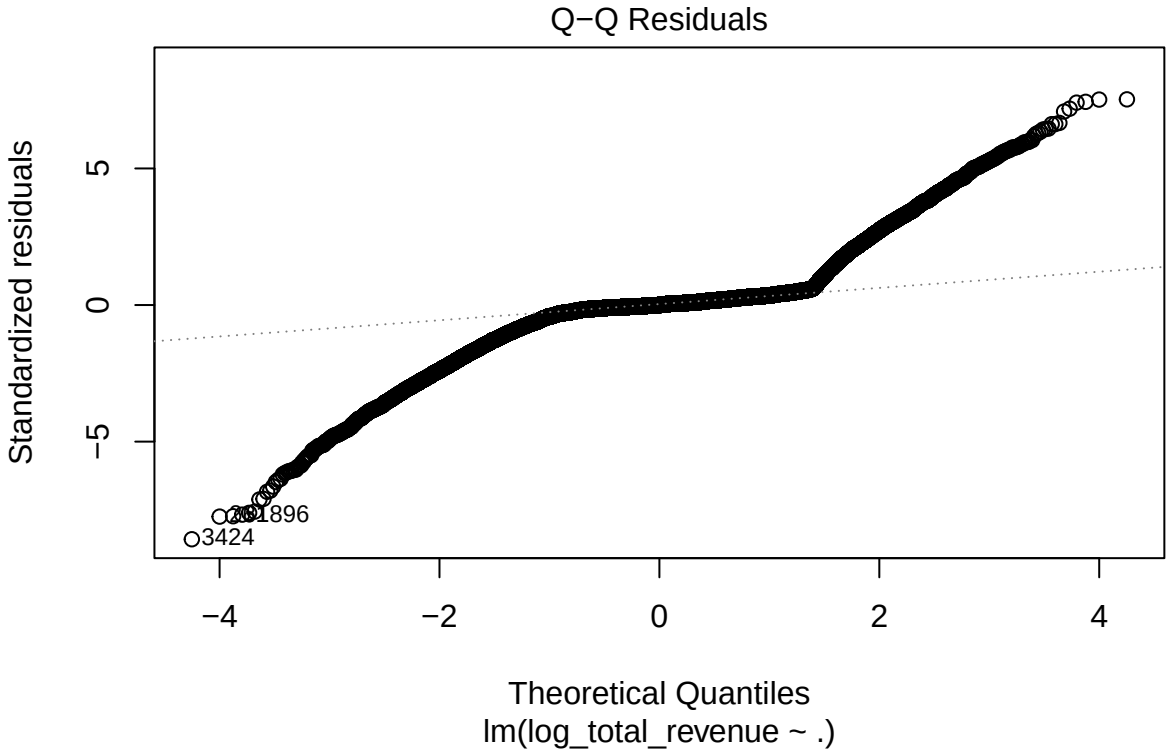
Handling multicollinearity

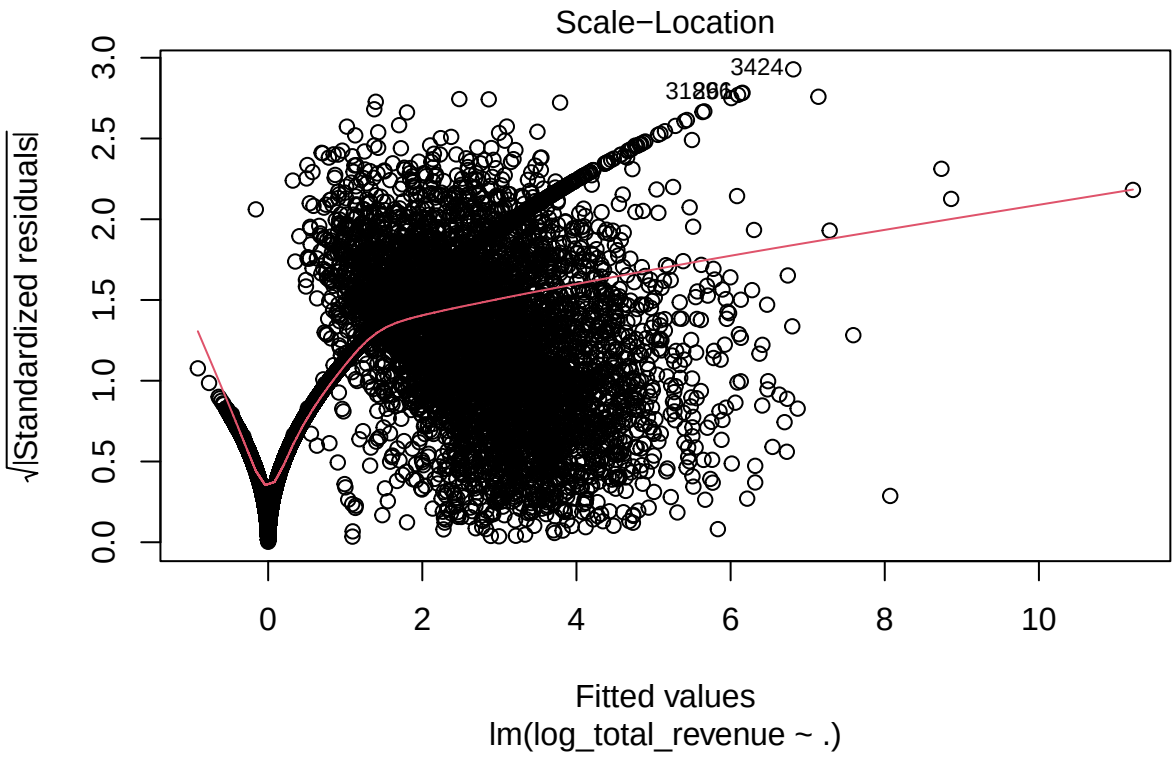
Now we focuses on **handling multicollinearity** within the dataset to ensure the quality and reliability of statistical models. Multicollinearity occurs when independent variables are highly correlated, which can inflate the variance of regression coefficients and make the model unstable.

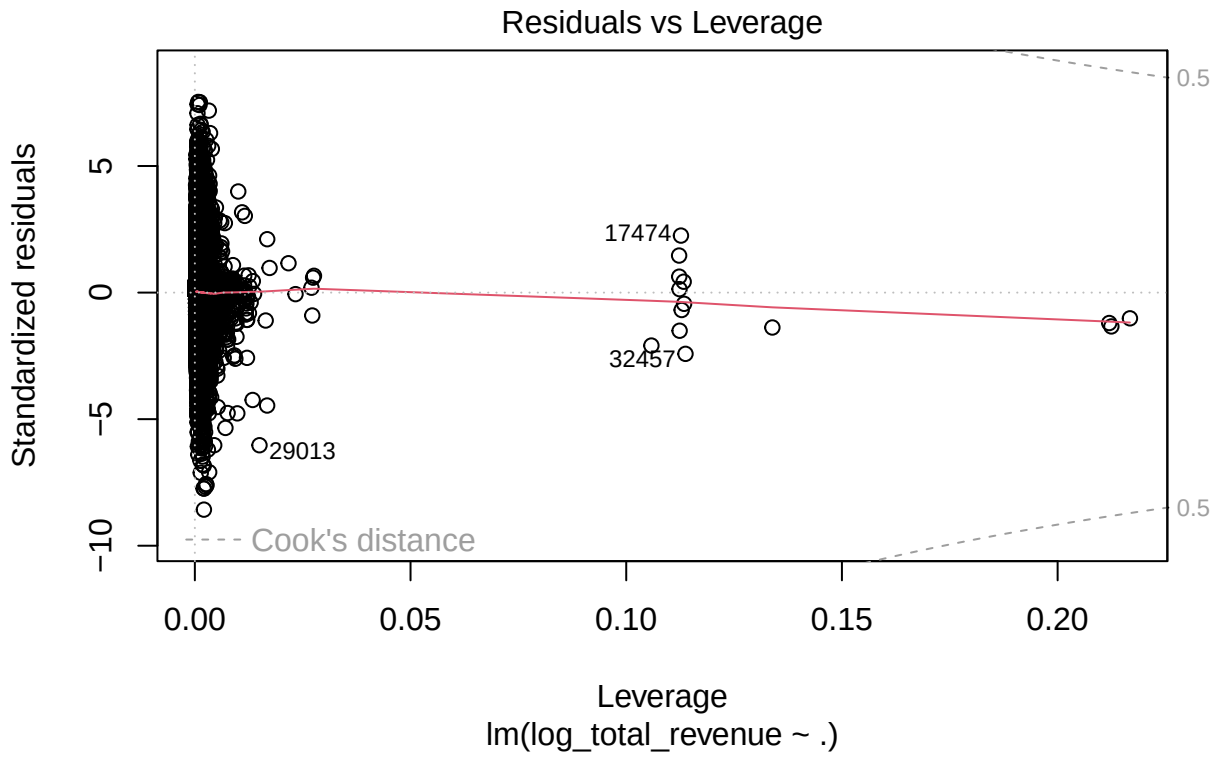
The first step involves fitting an **Ordinary Least Squares (OLS) regression model** using all the features to predict `log_total_revenue`. Diagnostic plots are generated to check for any unusual patterns or violations of regression assumptions, such as non-linearity or heteroscedasticity. Additionally, **outliers** are identified using statistical tests to see if any extreme data points are influencing the model's results.

The model exhibits non-linearity. Although there are some outliers present in the data, their removal is not critical, as they do not exert significant leverage on the model's performance or predictions.




Q−Q Residuals
Standardized residuals
5
0
−5
1896
3424
−4
−2
0
2
4
Theoretical Quantiles
lm(log_total_revenue ~ .)


Scale−Location
3424
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
2
4
6
8
10
Fitted values
lm(log_total_revenue ~ .)



Performing Outlier Tests :

There are outliers, But they dont have leverage.So We will not be removing any samples.

```
##        rstudent unadjusted p-value Bonferroni p
## 3424  -8.580483         9.7300e-18   4.5973e-13
## 261   -7.750271         9.3523e-15   4.4189e-10
## 31896 -7.736842         1.0393e-14   4.9107e-10
## 14166 -7.677546         1.6528e-14   7.8091e-10
## 18832 -7.613231         2.7229e-14   1.2865e-09
## 1591  -7.567148         3.8842e-14   1.8352e-09
## 42866  7.534648         4.9837e-14   2.3548e-09
## 20411  7.526268         5.3137e-14   2.5107e-09
## 11599  7.443267         9.9892e-14   4.7198e-09
## 26526  7.414231         1.2438e-13   5.8766e-09
```

Performing Multicollearity Tests:

Variables with vif>10 are removed.

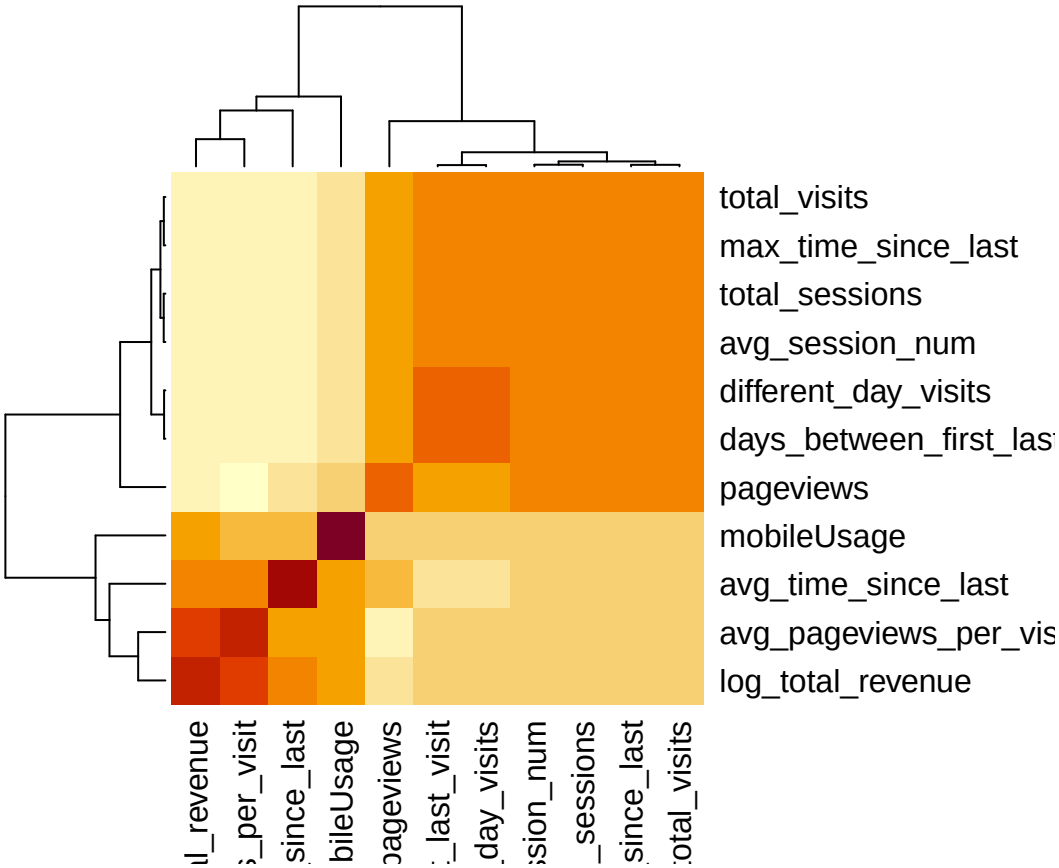
```
##                                      GVIF Df GVIF^(1/(2*Df))
## total_visits                 1.683161e+02  1       12.973669
## avg_pageviews_per_visit      5.967083e+00  1        2.442761
## days_between_first_last_visit 1.118103e+02  1       10.574037
```

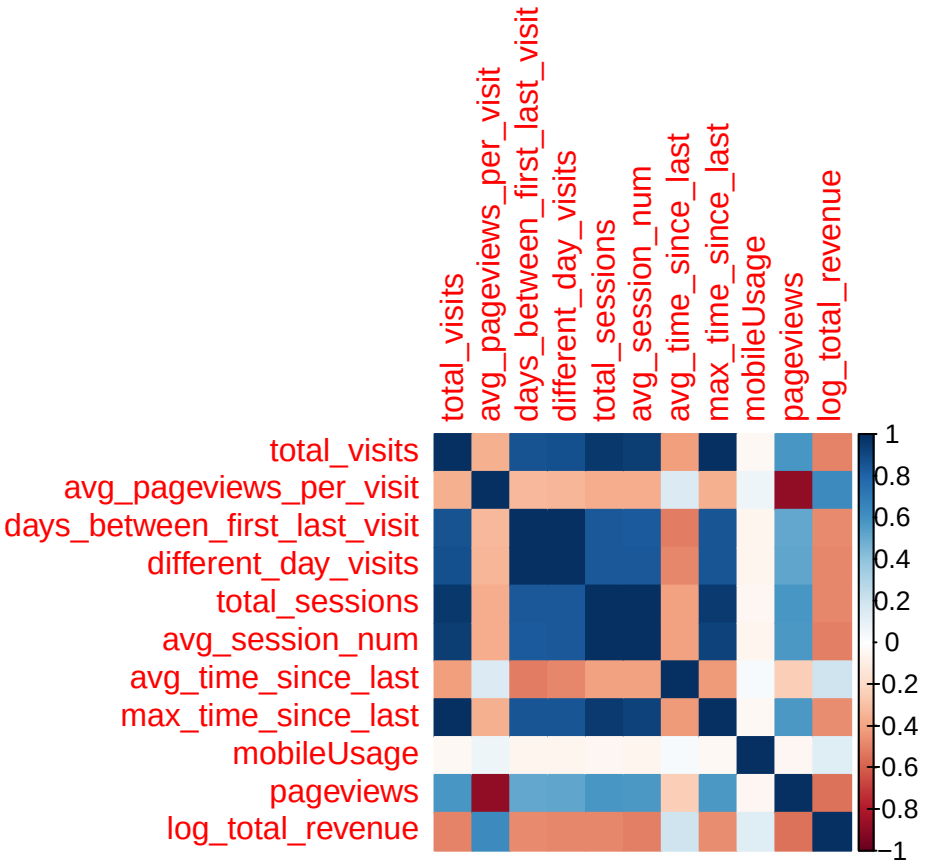
```
## different_day_visits          1.205489e+02  1       10.979474
## total_sessions                4.007805e+02  1       20.019503
## avg_session_num               2.765197e+02  1       16.628882
## avg_time_since_last           1.426044e+00  1        1.194171
## max_time_since_last           2.176524e+02  1       14.753048
## mobileUsage                   3.477885e+03  1       58.973592
## pageviews                     7.677079e+00  1        2.770754
## mostCommonBrowser             2.754942e+00  2        1.288333
## mostCommonOS                  8.694524e+01  6        1.450788
## mostCommonDevice              3.609433e+03  2        7.751036
## primaryChannel                7.221385e+05  4        5.399177
## mostFrequentContinent         3.530569e+00  3        1.233978
## mostFrequentCountry           3.694650e+00  1        1.922147
## primarySource                 2.122630e+05  4        4.632968
## primaryMedium                 4.728585e+01  3        1.901612
## mostCommonDay                 1.052724e+00  6        1.004291
```

The **Variance Inflation Factor (VIF)** is then computed to assess multicollinearity. A VIF value greater than 10 indicates that a variable is highly correlated with others in the model. In this case, variables with high VIF scores—such as `mobileUsage`, `total_sessions`, and `avg_session_num`—are removed to reduce multicollinearity and improve the stability of the model.

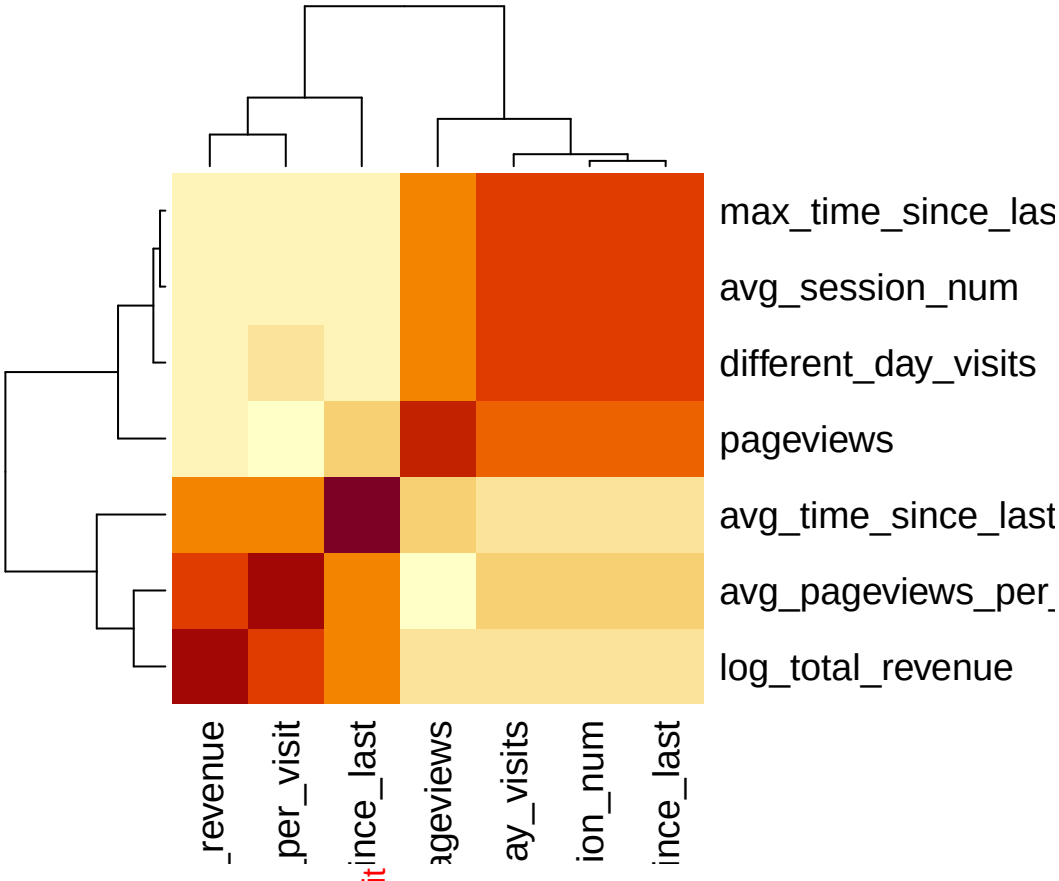
Next, a **correlation matrix** is computed to quantify the pairwise correlations between numeric variables, and both a **heatmap** and a **correlation plot** are generated for visualization. These plots help identify highly correlated pairs of variables, guiding further feature selection and dimensionality reduction.

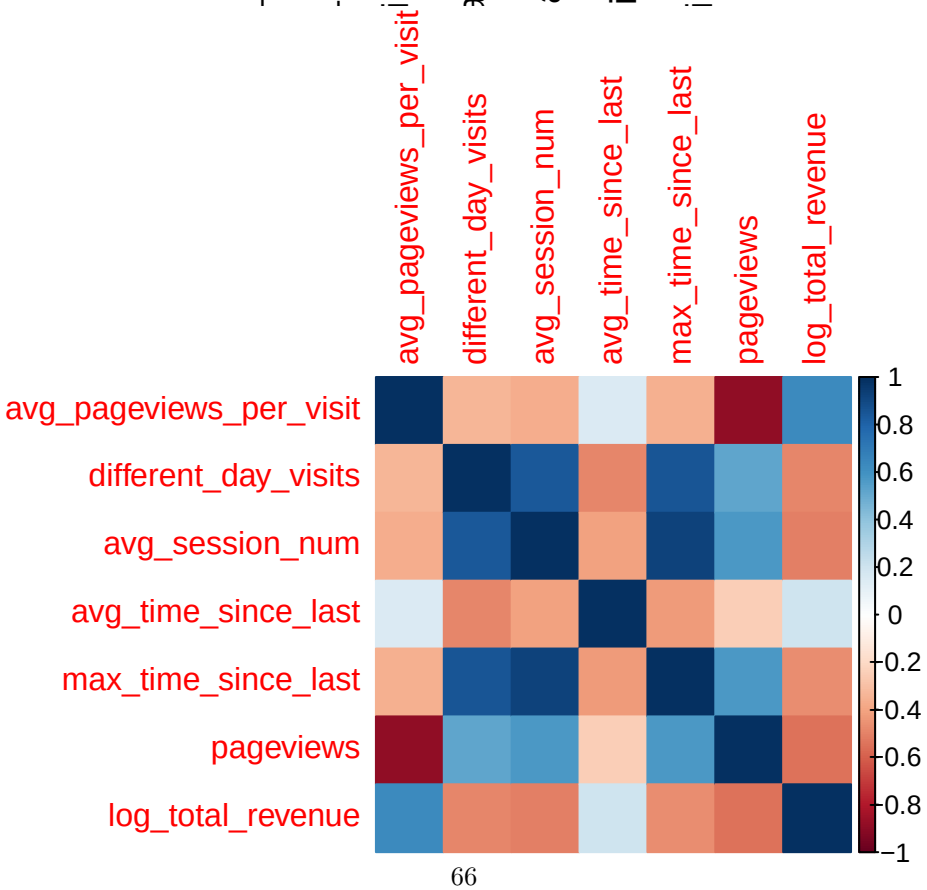
The high-VIF variables identified earlier are removed from the dataset to create a refined version of the feature set. This process ensures that the dataset is free of problematic multicollinearity, improving the interpretability and predictive performance of subsequent models.


total_visits
max_time_since_last
total_sessions
avg_session_num
different_day_visits
days_between_first_las
pageviews
mobileUsage
avg_time_since_last
avg_pageviews_per_vis
log_total_revenue
l_revenue
s_per_visit
since_last
bileUsage
pageviews
_last_visit
day_visits
ssion_num
_sessions
since_last
otal_visits


total_visits
avg_pageviews_per_visit
days_between_first_last_visit
different_day_visits
total_sessions
avg_session_num
avg_time_since_last
max_time_since_last
mobileUsage
pageviews
log_total_revenue
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1

AFter Removing varibales with VIF > 10





MODELLING

iii. (20 points) Modeling. Build an OLS model and 3 or more regression variant models (these may include robust regression, PLS, PCR, ridge regression, LASSO, elasticnet, MARS, or SVR) and summarize their performance in a table (as shown in Table 1). Clearly state your resampling approach. Note: You may combine models, techniques, etc.

10 Fold Cross Validation is used.

OLS (Ordinary Least Squares Regression) :

Trains linear models with different subsets of features. Uses 10-fold cross-validation to avoid overfitting.

```
## Linear Regression
##
## 47249 samples
##    15 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 42524, 42524, 42524, 42524, 42524, 42524, ...
## Resampling results:
##
##   RMSE       Rsquared  MAE
##   0.8266983  0.601511  0.4481246
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
## [1] 0.8266983
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1636 -0.1250  0.0255  0.1873  6.7073
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      0.241150   0.063056   3.824 0.000131 ***
## avg_pageviews_per_visit          1.329262   0.009205 144.410  < 2e-16 ***
## different_day_visits            -0.306533   0.008054 -38.061  < 2e-16 ***
## avg_session_num                 -0.355448   0.010938 -32.497  < 2e-16 ***
## avg_time_since_last             -0.058874   0.004374 -13.461  < 2e-16 ***
## max_time_since_last             -0.042672   0.011192  -3.813 0.000138 ***
```

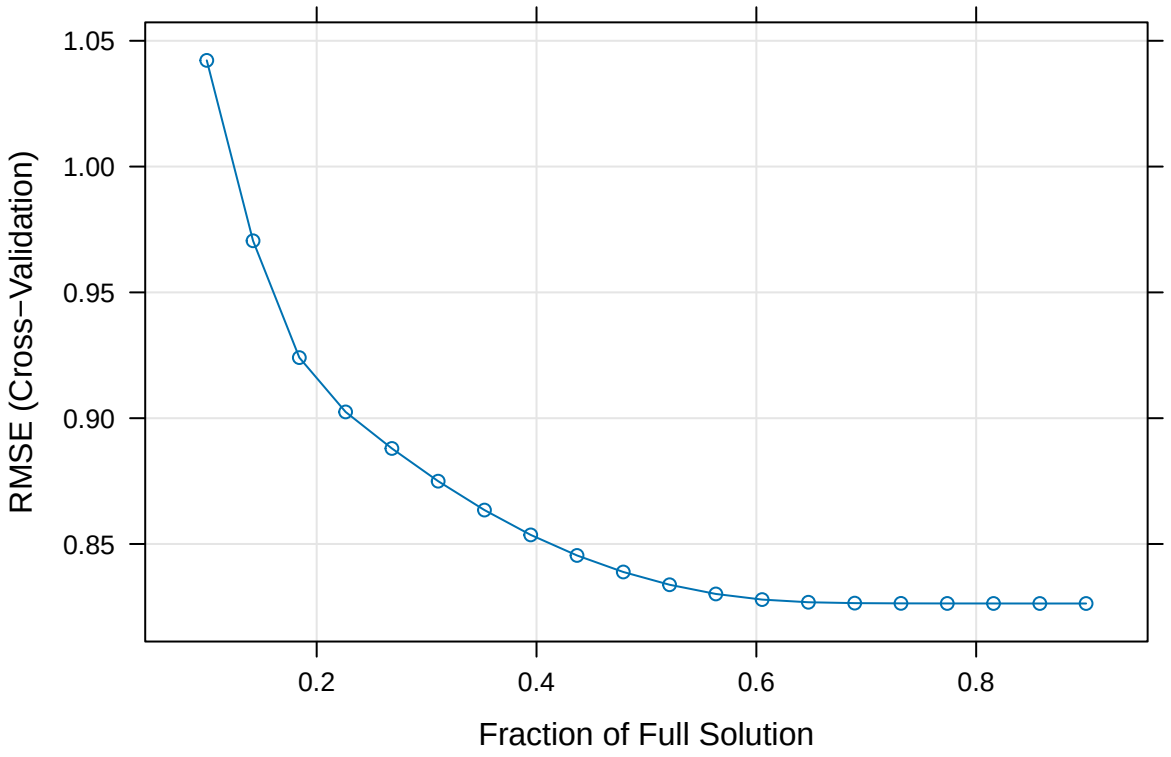
```
## pageviews                            0.934241   0.010420  89.658  < 2e-16 ***
## mostCommonBrowserOther               0.015664   0.012768   1.227 0.219877
## mostCommonBrowserSafari             -0.047199   0.014811  -3.187 0.001440 **
## `mostCommonOSChrome OS`              0.137976   0.054276   2.542 0.011021 *
## mostCommonOSiOS                      0.034218   0.020079   1.704 0.088359 .
## mostCommonOSLinux                   -0.068997   0.053308  -1.294 0.195563
## mostCommonOSMacintosh                0.150000   0.050346   2.979 0.002890 **
## mostCommonOSOther                    0.046314   0.076396   0.606 0.544357
## mostCommonOSWindows                  0.040342   0.050086   0.805 0.420563
## mostCommonDevicemobile              -0.032688   0.048914  -0.668 0.503959
## mostCommonDevicetablet              -0.053319   0.053041  -1.005 0.314787
## `primaryChannelOrganic Search`       0.674774   0.277316   2.433 0.014968 *
## primaryChannelOther                  0.708489   0.278612   2.543 0.010996 *
## primaryChannelReferral               0.570587   0.276322   2.065 0.038935 *
## primaryChannelSocial                 0.671420   0.278140   2.414 0.015784 *
## mostFrequentContinentAsia            0.031867   0.014670   2.172 0.029839 *
## mostFrequentContinentEurope          0.030764   0.015166   2.029 0.042512 *
## mostFrequentContinentOther           0.043076   0.024482   1.759 0.078503 .
## `mostFrequentCountryUnited States`   0.203551   0.014854  13.704  < 2e-16 ***
## primarySourcegoogle                 -0.738206   0.277202  -2.663 0.007746 **
## primarySourcemall.googleplex.com    -0.077298   0.276426  -0.280 0.779760
## primarySourceOther                  -0.723418   0.275874  -2.622 0.008737 **
## primarySourceyoutube.com            -0.670465   0.278197  -2.410 0.015955 *
## primaryMediumorganic                 0.054498   0.036625   1.488 0.136762
## primaryMediumOther                  -0.006423   0.044463  -0.144 0.885146
## primaryMediumreferral                0.087816   0.038547   2.278 0.022723 *
## mostCommonDayMon                    -0.018301   0.014030  -1.304 0.192090
## mostCommonDaySat                    -0.035341   0.015867  -2.227 0.025926 *
## mostCommonDaySun                    -0.078660   0.015148  -5.193 2.08e-07 ***
## mostCommonDayThu                    -0.020736   0.014335  -1.447 0.148030
## mostCommonDayTue                    -0.020016   0.014018  -1.428 0.153327
## mostCommonDayWed                    -0.034651   0.014152  -2.448 0.014352 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8261 on 47211 degrees of freedom
## Multiple R-squared:  0.6024, Adjusted R-squared:  0.602
## F-statistic:  1933 on 37 and 47211 DF,  p-value: < 2.2e-16
```

Lasso Regression:

Applies L1 regularization to encourage sparsity, reducing less impactful features. Hyperparameter tuning: fraction controls the penalty strength, with a grid search across values.

```
## The lasso
##
## 47249 samples
##    15 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 42524, 42524, 42524, 42524, 42524, 42524, ...
```

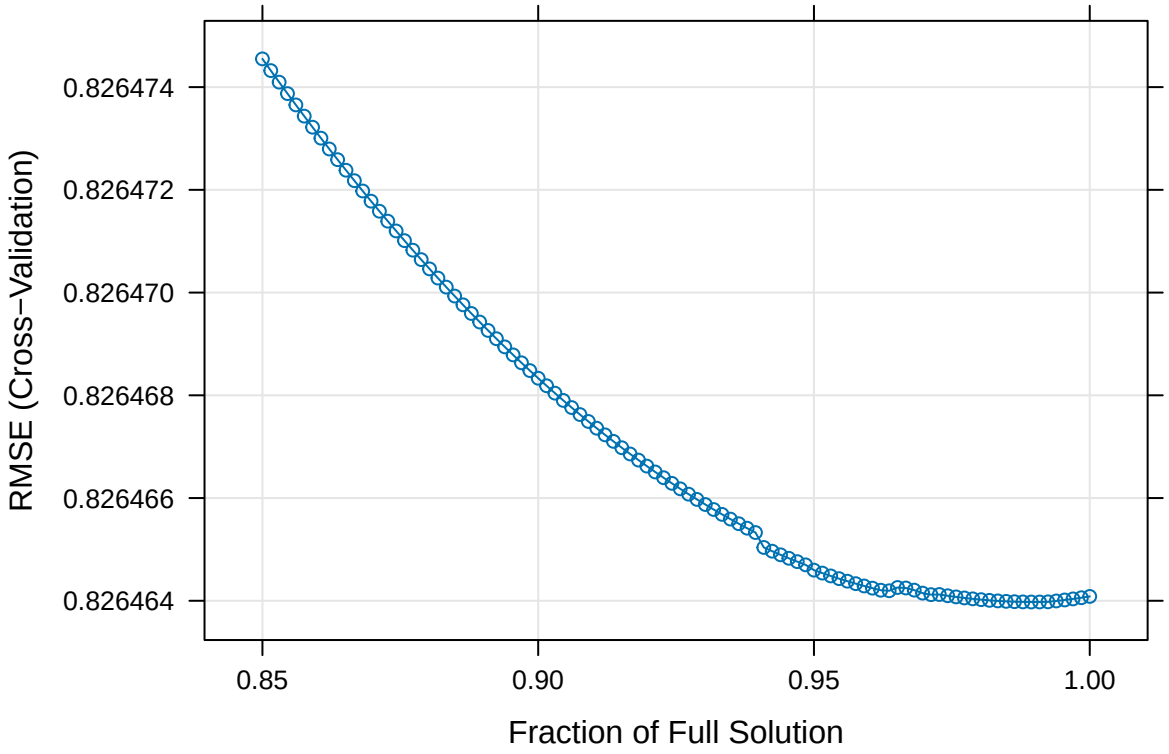
```
## Resampling results across tuning parameters:
##
##   fraction   RMSE       Rsquared   MAE
##   0.1000000  1.0422131  0.4834897  0.5731405
##   0.1421053  0.9705106  0.5103518  0.5348379
##   0.1842105  0.9240709  0.5200575  0.5560246
##   0.2263158  0.9024897  0.5294314  0.5588768
##   0.2684211  0.8879672  0.5447717  0.5341653
##   0.3105263  0.8749586  0.5581976  0.5123354
##   0.3526316  0.8634844  0.5695373  0.4924052
##   0.3947368  0.8536087  0.5788451  0.4754924
##   0.4368421  0.8454482  0.5861816  0.4619111
##   0.4789474  0.8388777  0.5918541  0.4519231
##   0.5210526  0.8337959  0.5960869  0.4464212
##   0.5631579  0.8301431  0.5990335  0.4442382
##   0.6052632  0.8279017  0.6007823  0.4441590
##   0.6473684  0.8268389  0.6015803  0.4456775
##   0.6894737  0.8264963  0.6018020  0.4471868
##   0.7315789  0.8263978  0.6018593  0.4477742
##   0.7736842  0.8263659  0.6018821  0.4479093
##   0.8157895  0.8263567  0.6018909  0.4479182
##   0.8578947  0.8263508  0.6018968  0.4479220
##   0.9000000  0.8263469  0.6019006  0.4479259
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was fraction = 0.9.
```



```
## The lasso
##
## 47249 samples
##    15 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 42525, 42524, 42524, 42524, 42524, 42524, ...
## Resampling results across tuning parameters:
##
##   fraction   RMSE       Rsquared   MAE
##   0.8500000  0.8264745  0.6017127  0.4481430
##   0.8515152  0.8264743  0.6017129  0.4481431
##   0.8530303  0.8264741  0.6017131  0.4481433
##   0.8545455  0.8264739  0.6017134  0.4481434
##   0.8560606  0.8264737  0.6017136  0.4481435
##   0.8575758  0.8264734  0.6017138  0.4481436
##   0.8590909  0.8264732  0.6017140  0.4481437
##   0.8606061  0.8264730  0.6017142  0.4481439
##   0.8621212  0.8264728  0.6017144  0.4481440
##   0.8636364  0.8264726  0.6017146  0.4481441
##   0.8651515  0.8264724  0.6017148  0.4481442
##   0.8666667  0.8264722  0.6017150  0.4481443
##   0.8681818  0.8264720  0.6017152  0.4481445
##   0.8696970  0.8264718  0.6017153  0.4481446
##   0.8712121  0.8264716  0.6017155  0.4481447
##   0.8727273  0.8264714  0.6017157  0.4481448
##   0.8742424  0.8264712  0.6017159  0.4481450
##   0.8757576  0.8264710  0.6017161  0.4481451
##   0.8772727  0.8264708  0.6017162  0.4481452
##   0.8787879  0.8264706  0.6017164  0.4481453
##   0.8803030  0.8264705  0.6017166  0.4481454
##   0.8818182  0.8264703  0.6017168  0.4481456
##   0.8833333  0.8264701  0.6017169  0.4481457
##   0.8848485  0.8264699  0.6017171  0.4481458
##   0.8863636  0.8264698  0.6017173  0.4481459
##   0.8878788  0.8264696  0.6017174  0.4481461
##   0.8893939  0.8264694  0.6017176  0.4481462
##   0.8909091  0.8264693  0.6017177  0.4481463
##   0.8924242  0.8264691  0.6017179  0.4481464
##   0.8939394  0.8264689  0.6017180  0.4481466
##   0.8954545  0.8264688  0.6017182  0.4481467
##   0.8969697  0.8264686  0.6017183  0.4481468
##   0.8984848  0.8264685  0.6017185  0.4481469
##   0.9000000  0.8264683  0.6017186  0.4481471
##   0.9015152  0.8264682  0.6017187  0.4481472
##   0.9030303  0.8264680  0.6017189  0.4481473
##   0.9045455  0.8264679  0.6017190  0.4481474
##   0.9060606  0.8264678  0.6017191  0.4481475
##   0.9075758  0.8264676  0.6017193  0.4481477
##   0.9090909  0.8264675  0.6017194  0.4481478
##   0.9106061  0.8264674  0.6017195  0.4481479
##   0.9121212  0.8264672  0.6017196  0.4481481
##   0.9136364  0.8264671  0.6017198  0.4481482
```

```
##   0.9151515  0.8264670  0.6017199  0.4481483
##   0.9166667  0.8264669  0.6017200  0.4481484
##   0.9181818  0.8264667  0.6017201  0.4481486
##   0.9196970  0.8264666  0.6017202  0.4481487
##   0.9212121  0.8264665  0.6017203  0.4481488
##   0.9227273  0.8264664  0.6017204  0.4481489
##   0.9242424  0.8264663  0.6017205  0.4481491
##   0.9257576  0.8264662  0.6017206  0.4481492
##   0.9272727  0.8264661  0.6017207  0.4481493
##   0.9287879  0.8264660  0.6017208  0.4481494
##   0.9303030  0.8264659  0.6017209  0.4481496
##   0.9318182  0.8264658  0.6017210  0.4481497
##   0.9333333  0.8264657  0.6017211  0.4481498
##   0.9348485  0.8264656  0.6017212  0.4481500
##   0.9363636  0.8264655  0.6017213  0.4481501
##   0.9378788  0.8264654  0.6017214  0.4481502
##   0.9393939  0.8264653  0.6017214  0.4481503
##   0.9409091  0.8264650  0.6017217  0.4481505
##   0.9424242  0.8264650  0.6017218  0.4481507
##   0.9439394  0.8264649  0.6017218  0.4481508
##   0.9454545  0.8264648  0.6017219  0.4481509
##   0.9469697  0.8264648  0.6017220  0.4481510
##   0.9484848  0.8264647  0.6017220  0.4481511
##   0.9500000  0.8264646  0.6017221  0.4481513
##   0.9515152  0.8264645  0.6017221  0.4481515
##   0.9530303  0.8264645  0.6017222  0.4481516
##   0.9545455  0.8264644  0.6017223  0.4481517
##   0.9560606  0.8264644  0.6017223  0.4481518
##   0.9575758  0.8264643  0.6017223  0.4481519
##   0.9590909  0.8264643  0.6017224  0.4481520
##   0.9606061  0.8264642  0.6017224  0.4481522
##   0.9621212  0.8264642  0.6017225  0.4481523
##   0.9636364  0.8264642  0.6017225  0.4481525
##   0.9651515  0.8264643  0.6017224  0.4481527
##   0.9666667  0.8264642  0.6017225  0.4481529
##   0.9681818  0.8264642  0.6017225  0.4481530
##   0.9696970  0.8264641  0.6017225  0.4481532
##   0.9712121  0.8264641  0.6017226  0.4481533
##   0.9727273  0.8264641  0.6017226  0.4481535
##   0.9742424  0.8264641  0.6017226  0.4481536
##   0.9757576  0.8264641  0.6017226  0.4481537
##   0.9772727  0.8264641  0.6017226  0.4481538
##   0.9787879  0.8264640  0.6017226  0.4481539
##   0.9803030  0.8264640  0.6017227  0.4481540
##   0.9818182  0.8264640  0.6017227  0.4481541
##   0.9833333  0.8264640  0.6017227  0.4481542
##   0.9848485  0.8264640  0.6017227  0.4481544
##   0.9863636  0.8264640  0.6017227  0.4481545
##   0.9878788  0.8264640  0.6017227  0.4481546
##   0.9893939  0.8264640  0.6017227  0.4481547
##   0.9909091  0.8264640  0.6017227  0.4481548
##   0.9924242  0.8264640  0.6017227  0.4481549
##   0.9939394  0.8264640  0.6017227  0.4481550
##   0.9954545  0.8264640  0.6017226  0.4481551
```

```
##   0.9969697  0.8264640  0.6017226  0.4481552
##   0.9984848  0.8264641  0.6017226  0.4481553
##   1.0000000  0.8264641  0.6017226  0.4481553
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was fraction = 0.9893939.
```



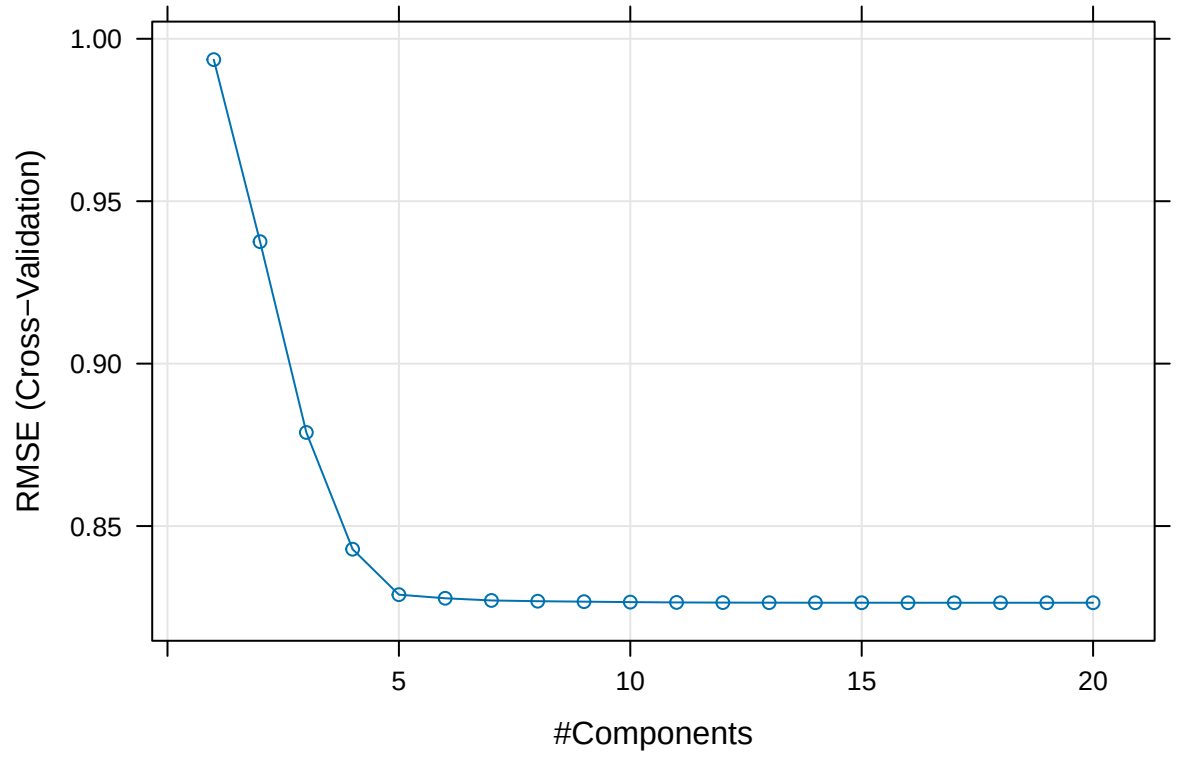

```
## [1] 0.826464
```

PLS (Partial Least Squares Regression)

Useful for handling multicollinearity and dimensionality reduction. Hyperparameter tuning: Number of components (ncomp) optimized to minimize RMSE.

```
## Partial Least Squares
##
## 47249 samples
##    15 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 42524, 42524, 42524, 42524, 42524, 42524, ...
## Resampling results across tuning parameters:
##
```

```
##    ncomp  RMSE       Rsquared   MAE
##     1     0.9935836  0.4244904  0.6098338
##     2     0.9375433  0.4874387  0.6438810
##     3     0.8788154  0.5496791  0.5501289
##     4     0.8428746  0.5858217  0.4898893
##     5     0.8288797  0.5994929  0.4483816
##     6     0.8277717  0.6005777  0.4471019
##     7     0.8271010  0.6012159  0.4481485
##     8     0.8268694  0.6014396  0.4481214
##     9     0.8267246  0.6015813  0.4496963
##    10     0.8265901  0.6017003  0.4485057
##    11     0.8264877  0.6017965  0.4479678
##    12     0.8264406  0.6018413  0.4489776
##    13     0.8264138  0.6018710  0.4482392
##    14     0.8264028  0.6018805  0.4483531
##    15     0.8264003  0.6018815  0.4483359
##    16     0.8263951  0.6018859  0.4480252
##    17     0.8263935  0.6018878  0.4481180
##    18     0.8263926  0.6018888  0.4479866
##    19     0.8263949  0.6018867  0.4479762
##    20     0.8263989  0.6018825  0.4480272
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was ncomp = 18.
```

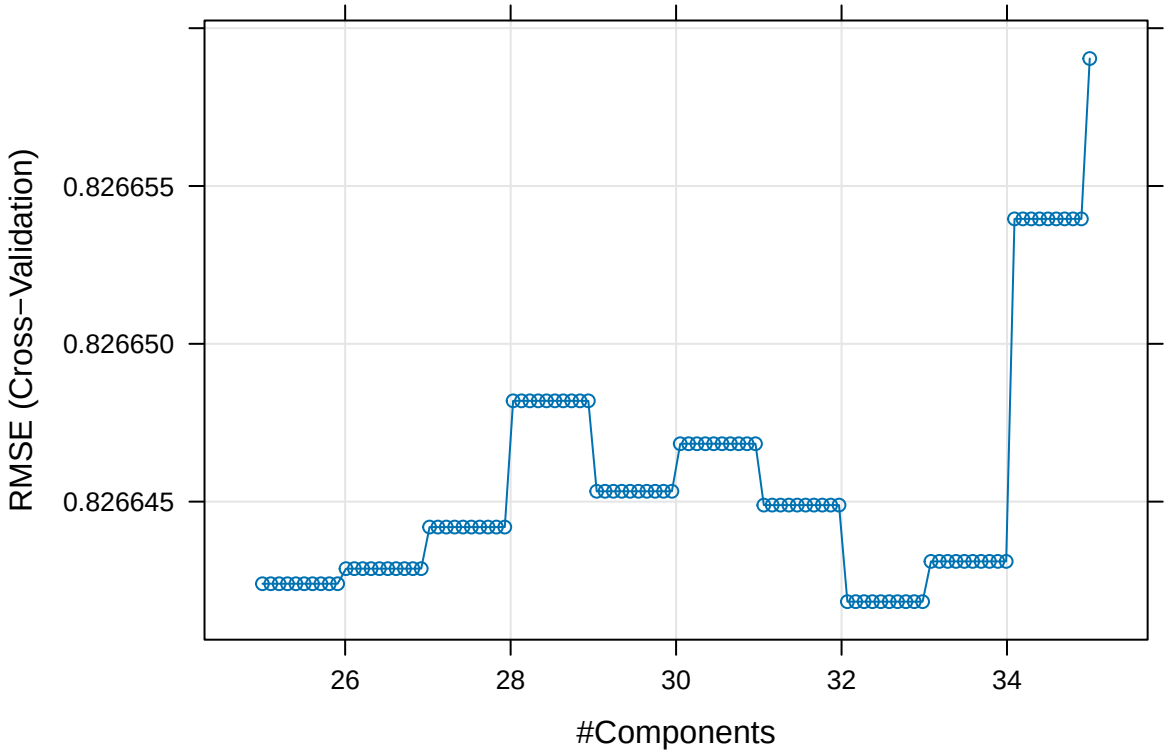



```
## Partial Least Squares
```

```
##
## 47249 samples
##    15 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 42524, 42524, 42525, 42524, 42524, 42524, ...
## Resampling results across tuning parameters:
##
##   ncomp     RMSE       Rsquared   MAE
##   25.00000  0.8266424  0.6012640  0.4480223
##   25.10101  0.8266424  0.6012640  0.4480223
##   25.20202  0.8266424  0.6012640  0.4480223
##   25.30303  0.8266424  0.6012640  0.4480223
##   25.40404  0.8266424  0.6012640  0.4480223
##   25.50505  0.8266424  0.6012640  0.4480223
##   25.60606  0.8266424  0.6012640  0.4480223
##   25.70707  0.8266424  0.6012640  0.4480223
##   25.80808  0.8266424  0.6012640  0.4480223
##   25.90909  0.8266424  0.6012640  0.4480223
##   26.01010  0.8266429  0.6012633  0.4480382
##   26.11111  0.8266429  0.6012633  0.4480382
##   26.21212  0.8266429  0.6012633  0.4480382
##   26.31313  0.8266429  0.6012633  0.4480382
##   26.41414  0.8266429  0.6012633  0.4480382
##   26.51515  0.8266429  0.6012633  0.4480382
##   26.61616  0.8266429  0.6012633  0.4480382
##   26.71717  0.8266429  0.6012633  0.4480382
##   26.81818  0.8266429  0.6012633  0.4480382
##   26.91919  0.8266429  0.6012633  0.4480382
##   27.02020  0.8266442  0.6012620  0.4480227
##   27.12121  0.8266442  0.6012620  0.4480227
##   27.22222  0.8266442  0.6012620  0.4480227
##   27.32323  0.8266442  0.6012620  0.4480227
##   27.42424  0.8266442  0.6012620  0.4480227
##   27.52525  0.8266442  0.6012620  0.4480227
##   27.62626  0.8266442  0.6012620  0.4480227
##   27.72727  0.8266442  0.6012620  0.4480227
##   27.82828  0.8266442  0.6012620  0.4480227
##   27.92929  0.8266442  0.6012620  0.4480227
##   28.03030  0.8266482  0.6012582  0.4480554
##   28.13131  0.8266482  0.6012582  0.4480554
##   28.23232  0.8266482  0.6012582  0.4480554
##   28.33333  0.8266482  0.6012582  0.4480554
##   28.43434  0.8266482  0.6012582  0.4480554
##   28.53535  0.8266482  0.6012582  0.4480554
##   28.63636  0.8266482  0.6012582  0.4480554
##   28.73737  0.8266482  0.6012582  0.4480554
##   28.83838  0.8266482  0.6012582  0.4480554
##   28.93939  0.8266482  0.6012582  0.4480554
##   29.04040  0.8266453  0.6012613  0.4480529
##   29.14141  0.8266453  0.6012613  0.4480529
##   29.24242  0.8266453  0.6012613  0.4480529
##   29.34343  0.8266453  0.6012613  0.4480529
```

```
##   29.44444  0.8266453  0.6012613  0.4480529
##   29.54545  0.8266453  0.6012613  0.4480529
##   29.64646  0.8266453  0.6012613  0.4480529
##   29.74747  0.8266453  0.6012613  0.4480529
##   29.84848  0.8266453  0.6012613  0.4480529
##   29.94949  0.8266453  0.6012613  0.4480529
##   30.05051  0.8266468  0.6012597  0.4480208
##   30.15152  0.8266468  0.6012597  0.4480208
##   30.25253  0.8266468  0.6012597  0.4480208
##   30.35354  0.8266468  0.6012597  0.4480208
##   30.45455  0.8266468  0.6012597  0.4480208
##   30.55556  0.8266468  0.6012597  0.4480208
##   30.65657  0.8266468  0.6012597  0.4480208
##   30.75758  0.8266468  0.6012597  0.4480208
##   30.85859  0.8266468  0.6012597  0.4480208
##   30.95960  0.8266468  0.6012597  0.4480208
##   31.06061  0.8266449  0.6012619  0.4480699
##   31.16162  0.8266449  0.6012619  0.4480699
##   31.26263  0.8266449  0.6012619  0.4480699
##   31.36364  0.8266449  0.6012619  0.4480699
##   31.46465  0.8266449  0.6012619  0.4480699
##   31.56566  0.8266449  0.6012619  0.4480699
##   31.66667  0.8266449  0.6012619  0.4480699
##   31.76768  0.8266449  0.6012619  0.4480699
##   31.86869  0.8266449  0.6012619  0.4480699
##   31.96970  0.8266449  0.6012619  0.4480699
##   32.07071  0.8266418  0.6012648  0.4481245
##   32.17172  0.8266418  0.6012648  0.4481245
##   32.27273  0.8266418  0.6012648  0.4481245
##   32.37374  0.8266418  0.6012648  0.4481245
##   32.47475  0.8266418  0.6012648  0.4481245
##   32.57576  0.8266418  0.6012648  0.4481245
##   32.67677  0.8266418  0.6012648  0.4481245
##   32.77778  0.8266418  0.6012648  0.4481245
##   32.87879  0.8266418  0.6012648  0.4481245
##   32.97980  0.8266418  0.6012648  0.4481245
##   33.08081  0.8266431  0.6012640  0.4480832
##   33.18182  0.8266431  0.6012640  0.4480832
##   33.28283  0.8266431  0.6012640  0.4480832
##   33.38384  0.8266431  0.6012640  0.4480832
##   33.48485  0.8266431  0.6012640  0.4480832
##   33.58586  0.8266431  0.6012640  0.4480832
##   33.68687  0.8266431  0.6012640  0.4480832
##   33.78788  0.8266431  0.6012640  0.4480832
##   33.88889  0.8266431  0.6012640  0.4480832
##   33.98990  0.8266431  0.6012640  0.4480832
##   34.09091  0.8266540  0.6012538  0.4480467
##   34.19192  0.8266540  0.6012538  0.4480467
##   34.29293  0.8266540  0.6012538  0.4480467
##   34.39394  0.8266540  0.6012538  0.4480467
##   34.49495  0.8266540  0.6012538  0.4480467
##   34.59596  0.8266540  0.6012538  0.4480467
##   34.69697  0.8266540  0.6012538  0.4480467
##   34.79798  0.8266540  0.6012538  0.4480467
```

```
##   34.89899  0.8266540  0.6012538  0.4480467
##   35.00000  0.8266590  0.6012507  0.4480336
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was ncomp = 32.07071.
```



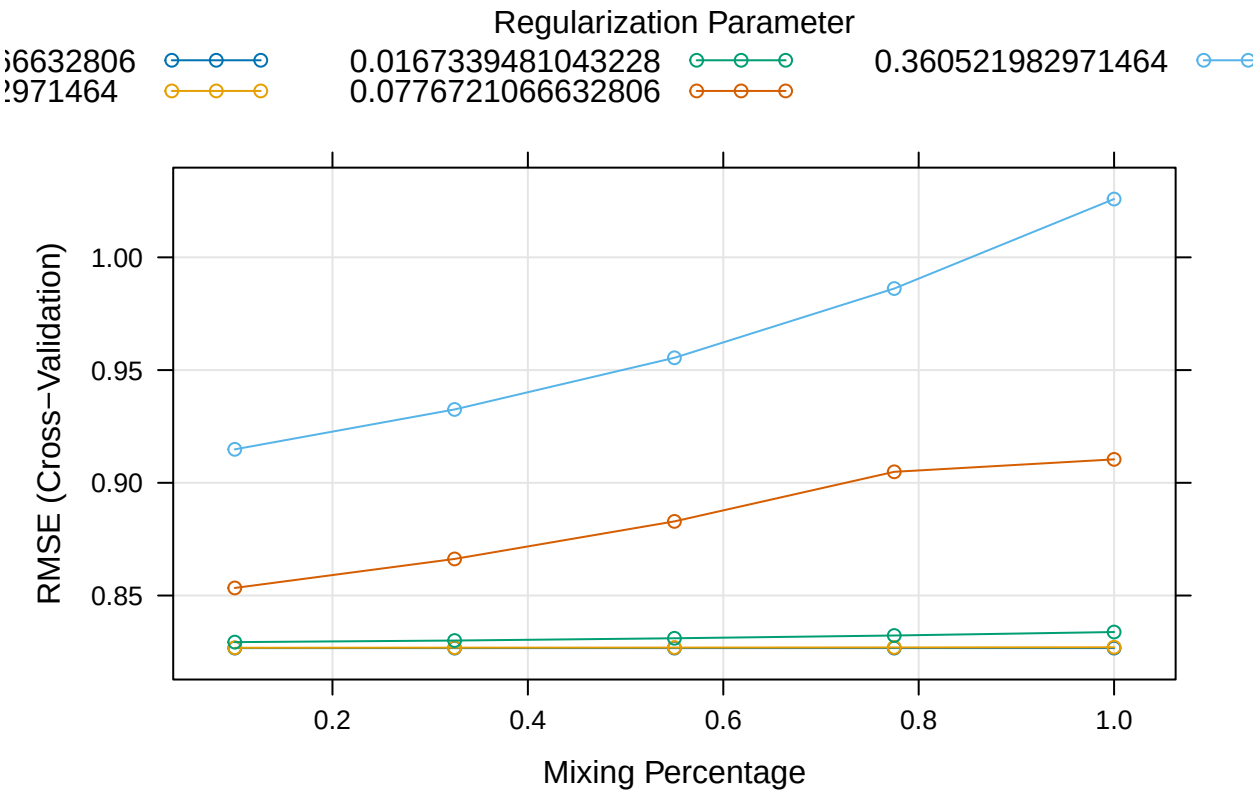

```
## [1] 0.8266418
```

ElasticNet Regression (glmnet) :

Combines Lasso (L1) and Ridge (L2) penalties, balancing feature selection and shrinkage. Hyperparameter tuning: alpha (mixing parameter) and lambda (regularization strength).

```
## glmnet
##
## 47249 samples
##    15 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 42524, 42524, 42524, 42524, 42524, 42524, ...
## Resampling results across tuning parameters:
##
```

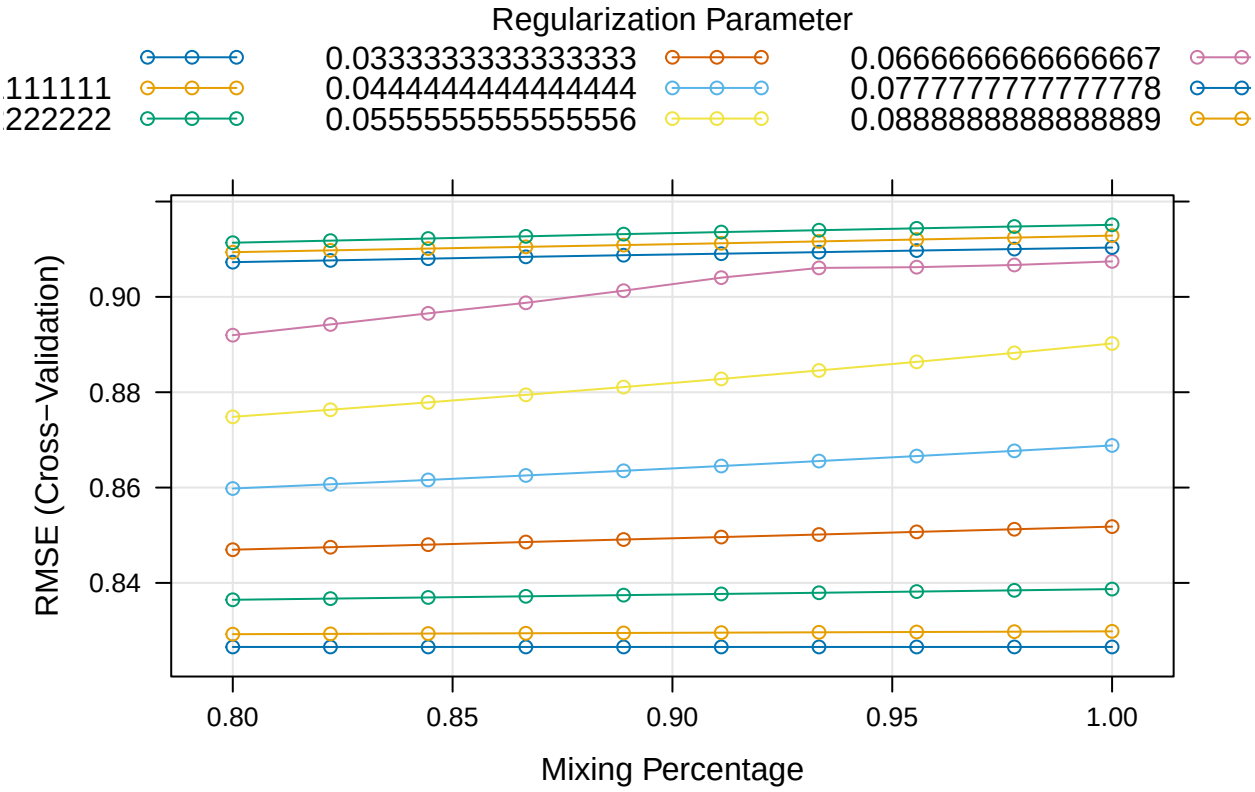
```
##   alpha  lambda        RMSE       Rsquared   MAE
##   0.100  0.0007767211  0.8266871  0.6010917  0.4476465
##   0.100  0.0036052198  0.8268198  0.6009936  0.4471469
##   0.100  0.0167339481  0.8293350  0.5990744  0.4493414
##   0.100  0.0776721067  0.8533523  0.5785180  0.4823338
##   0.100  0.3605219830  0.9148439  0.5231923  0.5474988
##   0.325  0.0007767211  0.8266867  0.6010941  0.4474106
##   0.325  0.0036052198  0.8268718  0.6009539  0.4462425
##   0.325  0.0167339481  0.8300350  0.5985802  0.4480582
##   0.325  0.0776721067  0.8662238  0.5665777  0.4980406
##   0.325  0.3605219830  0.9325298  0.5219303  0.5385146
##   0.550  0.0007767211  0.8266892  0.6010888  0.4472069
##   0.550  0.0036052198  0.8269098  0.6009255  0.4455790
##   0.550  0.0167339481  0.8310380  0.5977945  0.4470638
##   0.550  0.0776721067  0.8828844  0.5499952  0.5253157
##   0.550  0.3605219830  0.9554458  0.5196252  0.5308912
##   0.775  0.0007767211  0.8266953  0.6010818  0.4470884
##   0.775  0.0036052198  0.8269656  0.6008866  0.4449820
##   0.775  0.0167339481  0.8322733  0.5967958  0.4463827
##   0.775  0.0776721067  0.9048660  0.5266656  0.5621165
##   0.775  0.3605219830  0.9861632  0.5111896  0.5242898
##   1.000  0.0007767211  0.8267030  0.6010766  0.4469044
##   1.000  0.0036052198  0.8270420  0.6008262  0.4444691
##   1.000  0.0167339481  0.8338087  0.5955070  0.4460851
##   1.000  0.0776721067  0.9103991  0.5221413  0.5659391
##   1.000  0.3605219830  1.0258611  0.4885155  0.5575652
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 0.325 and lambda
##  = 0.0007767211.
```




```
## glmnet
##
## 47249 samples
##    15 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 42524, 42524, 42525, 42524, 42524, 42524, ...
## Resampling results across tuning parameters:
##
##   alpha      lambda      RMSE       Rsquared   MAE
##   0.8000000  0.00000000  0.8265596  0.6016240  0.4471896
##   0.8000000  0.01111111  0.8292411  0.5996118  0.4446380
##   0.8000000  0.02222222  0.8364587  0.5938231  0.4496159
##   0.8000000  0.03333333  0.8469639  0.5848174  0.4651656
##   0.8000000  0.04444444  0.8597943  0.5731136  0.4862152
##   0.8000000  0.05555556  0.8748341  0.5584834  0.5122098
##   0.8000000  0.06666667  0.8919673  0.5408184  0.5405831
##   0.8000000  0.07777778  0.9072853  0.5245456  0.5662736
##   0.8000000  0.08888889  0.9093757  0.5235860  0.5649574
##   0.8000000  0.10000000  0.9113643  0.5229061  0.5633383
##   0.8222222  0.00000000  0.8265598  0.6016226  0.4471938
##   0.8222222  0.01111111  0.8293052  0.5995598  0.4445695
##   0.8222222  0.02222222  0.8367053  0.5936145  0.4498012
##   0.8222222  0.03333333  0.8474844  0.5843518  0.4659213
##   0.8222222  0.04444444  0.8606850  0.5722704  0.4876795
```

```
##   0.8222222  0.05555556  0.8763214  0.5569768  0.5147353
##   0.8222222  0.06666667  0.8942175  0.5384088  0.5444168
##   0.8222222  0.07777778  0.9076510  0.5243032  0.5662863
##   0.8222222  0.08888889  0.9097561  0.5233761  0.5648797
##   0.8222222  0.10000000  0.9117989  0.5226858  0.5632156
##   0.8444444  0.00000000  0.8265618  0.6016212  0.4471856
##   0.8444444  0.01111111  0.8293693  0.5995081  0.4445006
##   0.8444444  0.02222222  0.8369447  0.5934128  0.4500077
##   0.8444444  0.03333333  0.8480214  0.5838700  0.4667017
##   0.8444444  0.04444444  0.8615988  0.5714005  0.4891784
##   0.8444444  0.05555556  0.8778740  0.5553940  0.5173591
##   0.8444444  0.06666667  0.8965464  0.5358998  0.5484925
##   0.8444444  0.07777778  0.9080208  0.5240575  0.5662853
##   0.8444444  0.08888889  0.9101333  0.5231720  0.5647917
##   0.8444444  0.10000000  0.9122437  0.5224560  0.5630937
##   0.8666667  0.00000000  0.8265625  0.6016203  0.4471794
##   0.8666667  0.01111111  0.8294338  0.5994559  0.4444311
##   0.8666667  0.02222222  0.8371831  0.5932123  0.4502167
##   0.8666667  0.03333333  0.8485745  0.5833710  0.4675040
##   0.8666667  0.04444444  0.8625368  0.5705028  0.4907261
##   0.8666667  0.05555556  0.8794515  0.5537767  0.5199902
##   0.8666667  0.06666667  0.8987527  0.5335162  0.5523718
##   0.8666667  0.07777778  0.9084008  0.5238016  0.5662815
##   0.8666667  0.08888889  0.9104943  0.5229882  0.5646794
##   0.8666667  0.10000000  0.9126991  0.5222160  0.5629701
##   0.8888889  0.00000000  0.8265620  0.6016209  0.4471526
##   0.8888889  0.01111111  0.8295002  0.5994023  0.4443643
##   0.8888889  0.02222222  0.8374322  0.5930019  0.4504486
##   0.8888889  0.03333333  0.8491019  0.5828964  0.4682753
##   0.8888889  0.04444444  0.8635116  0.5695653  0.4923727
##   0.8888889  0.05555556  0.8810811  0.5520964  0.5226983
##   0.8888889  0.06666667  0.9012934  0.5307425  0.5569459
##   0.8888889  0.07777778  0.9087380  0.5235988  0.5662350
##   0.8888889  0.08888889  0.9108660  0.5227950  0.5645713
##   0.8888889  0.10000000  0.9131670  0.5219648  0.5628495
##   0.9111111  0.00000000  0.8265606  0.6016209  0.4471471
##   0.9111111  0.01111111  0.8295661  0.5993485  0.4442962
##   0.9111111  0.02222222  0.8376858  0.5927869  0.4506853
##   0.9111111  0.03333333  0.8496161  0.5824382  0.4690598
##   0.9111111  0.04444444  0.8645185  0.5685920  0.4941158
##   0.9111111  0.05555556  0.8827905  0.5503232  0.5255271
##   0.9111111  0.06666667  0.9040378  0.5277280  0.5619335
##   0.9111111  0.07777778  0.9090584  0.5234178  0.5661662
##   0.9111111  0.08888889  0.9112475  0.5225928  0.5644649
##   0.9111111  0.10000000  0.9135942  0.5217651  0.5627105
##   0.9333333  0.00000000  0.8265605  0.6016209  0.4471497
##   0.9333333  0.01111111  0.8296336  0.5992932  0.4442300
##   0.9333333  0.02222222  0.8379317  0.5925786  0.4509011
##   0.9333333  0.03333333  0.8501498  0.5819605  0.4698716
##   0.9333333  0.04444444  0.8655583  0.5675812  0.4959184
##   0.9333333  0.05555556  0.8845606  0.5484756  0.5284434
##   0.9333333  0.06666667  0.9060752  0.5255190  0.5654417
##   0.9333333  0.07777778  0.9093850  0.5232315  0.5660963
##   0.9333333  0.08888889  0.9116373  0.5223830  0.5643598
```

```
##   0.9333333  0.10000000  0.9139893  0.5216079  0.5625541
##   0.9555556  0.00000000  0.8265602  0.6016203  0.4471444
##   0.9555556  0.01111111  0.8297005  0.5992386  0.4441653
##   0.9555556  0.02222222  0.8381829  0.5923649  0.4511232
##   0.9555556  0.03333333  0.8506989  0.5814664  0.4707082
##   0.9555556  0.04444444  0.8666034  0.5665600  0.4977448
##   0.9555556  0.05555556  0.8863699  0.5465764  0.5313900
##   0.9555556  0.06666667  0.9062366  0.5254773  0.5651659
##   0.9555556  0.07777778  0.9097170  0.5230407  0.5660252
##   0.9555556  0.08888889  0.9120347  0.5221665  0.5642556
##   0.9555556  0.10000000  0.9143794  0.5214604  0.5623956
##   0.9777778  0.00000000  0.8265631  0.6016176  0.4471219
##   0.9777778  0.01111111  0.8297671  0.5991848  0.4441037
##   0.9777778  0.02222222  0.8384400  0.5921455  0.4513578
##   0.9777778  0.03333333  0.8512525  0.5809669  0.4715731
##   0.9777778  0.04444444  0.8676986  0.5654843  0.4996599
##   0.9777778  0.05555556  0.8882572  0.5445838  0.5344316
##   0.9777778  0.06666667  0.9066915  0.5250955  0.5654703
##   0.9777778  0.07777778  0.9100329  0.5228701  0.5659346
##   0.9777778  0.08888889  0.9124403  0.5219420  0.5641511
##   0.9777778  0.10000000  0.9147639  0.5213223  0.5622379
##   1.0000000  0.00000000  0.8265650  0.6016166  0.4471032
##   1.0000000  0.01111111  0.8298358  0.5991289  0.4440449
##   1.0000000  0.02222222  0.8387011  0.5919222  0.4515985
##   1.0000000  0.03333333  0.8518091  0.5804637  0.4724634
##   1.0000000  0.04444444  0.8688323  0.5643649  0.5016410
##   1.0000000  0.05555556  0.8902242  0.5424950  0.5377154
##   1.0000000  0.06666667  0.9074473  0.5243641  0.5663638
##   1.0000000  0.07777778  0.9103538  0.5226954  0.5658423
##   1.0000000  0.08888889  0.9128533  0.5217102  0.5640475
##   1.0000000  0.10000000  0.9151180  0.5212287  0.5620740
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were alpha = 0.8 and lambda = 0.
```



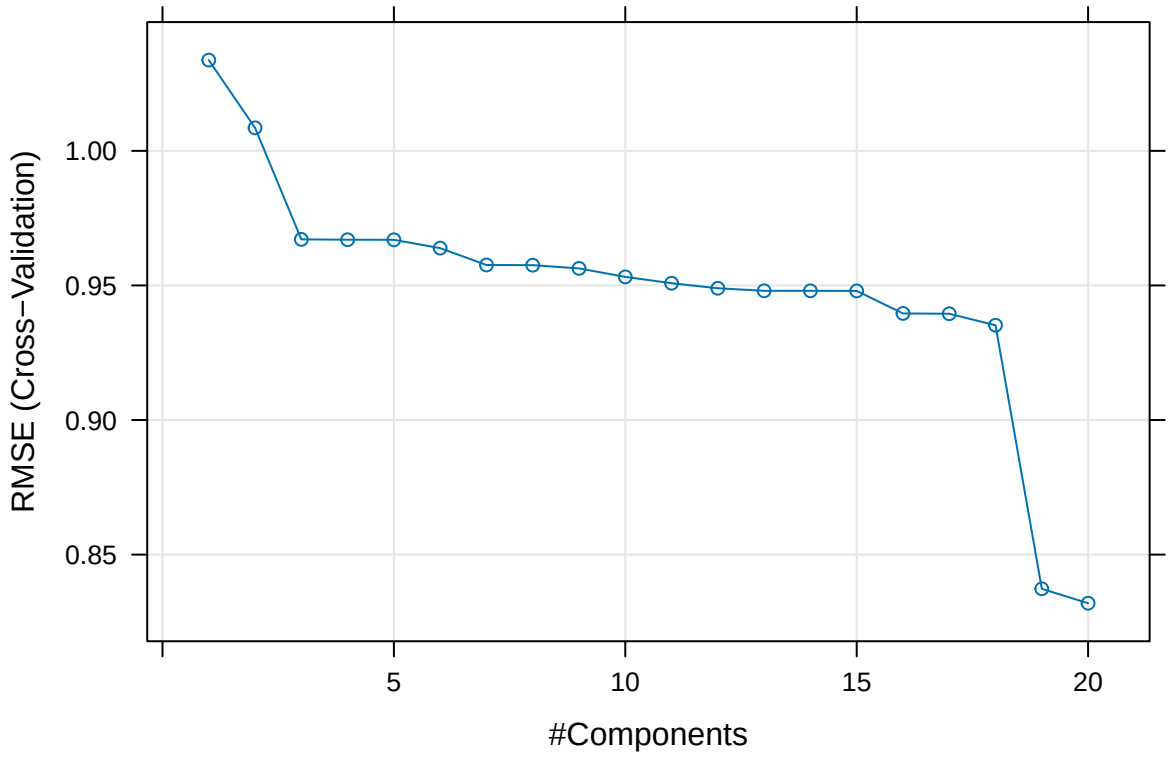

```
## [1] 0.8265596
```

PCR (Principal Component Regression):

Performs PCA before regression, handling multicollinearity. Hyperparameter tuning: Number of principal components (ncomp).

```
## Principal Component Analysis
##
## 47249 samples
##    15 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 42524, 42524, 42524, 42524, 42524, 42525, ...
## Resampling results across tuning parameters:
##
##   ncomp  RMSE       Rsquared   MAE
##    1     1.0337250  0.3769481  0.5824100
##    2     1.0085423  0.4068121  0.6571346
##    3     0.9671093  0.4545098  0.6677175
##    4     0.9669904  0.4546467  0.6670468
##    5     0.9669487  0.4546925  0.6668611
##    6     0.9638608  0.4581658  0.6605145
```

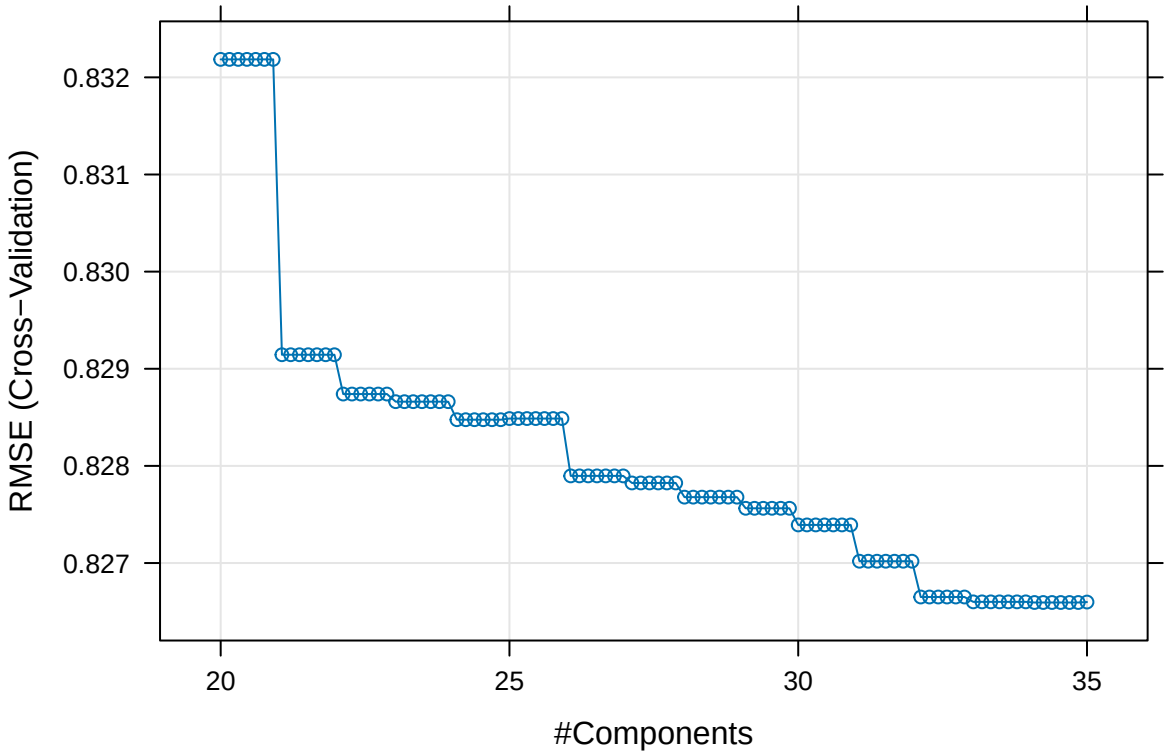
```
##     7     0.9576027  0.4651503  0.6528670
##     8     0.9575281  0.4652324  0.6529215
##     9     0.9563370  0.4665655  0.6523126
##    10     0.9531828  0.4700849  0.6465025
##    11     0.9508238  0.4727178  0.6411174
##    12     0.9489521  0.4747872  0.6361069
##    13     0.9479989  0.4758251  0.6341480
##    14     0.9479976  0.4758288  0.6341372
##    15     0.9479727  0.4758556  0.6341428
##    16     0.9395985  0.4850808  0.6177754
##    17     0.9394806  0.4852156  0.6175169
##    18     0.9352146  0.4898709  0.6107385
##    19     0.8372835  0.5911395  0.4711528
##    20     0.8319352  0.5964025  0.4529029
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was ncomp = 20.
```




```
## Principal Component Analysis
##
## 47249 samples
##    15 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
```

```
## Summary of sample sizes: 42524, 42524, 42524, 42524, 42524, 42525, ...
## Resampling results across tuning parameters:
##
##   ncomp     RMSE       Rsquared   MAE
##   20.00000  0.8321853  0.5963091  0.4528699
##   20.15152  0.8321853  0.5963091  0.4528699
##   20.30303  0.8321853  0.5963091  0.4528699
##   20.45455  0.8321853  0.5963091  0.4528699
##   20.60606  0.8321853  0.5963091  0.4528699
##   20.75758  0.8321853  0.5963091  0.4528699
##   20.90909  0.8321853  0.5963091  0.4528699
##   21.06061  0.8291443  0.5992866  0.4521484
##   21.21212  0.8291443  0.5992866  0.4521484
##   21.36364  0.8291443  0.5992866  0.4521484
##   21.51515  0.8291443  0.5992866  0.4521484
##   21.66667  0.8291443  0.5992866  0.4521484
##   21.81818  0.8291443  0.5992866  0.4521484
##   21.96970  0.8291443  0.5992866  0.4521484
##   22.12121  0.8287396  0.5996736  0.4515570
##   22.27273  0.8287396  0.5996736  0.4515570
##   22.42424  0.8287396  0.5996736  0.4515570
##   22.57576  0.8287396  0.5996736  0.4515570
##   22.72727  0.8287396  0.5996736  0.4515570
##   22.87879  0.8287396  0.5996736  0.4515570
##   23.03030  0.8286612  0.5997504  0.4513640
##   23.18182  0.8286612  0.5997504  0.4513640
##   23.33333  0.8286612  0.5997504  0.4513640
##   23.48485  0.8286612  0.5997504  0.4513640
##   23.63636  0.8286612  0.5997504  0.4513640
##   23.78788  0.8286612  0.5997504  0.4513640
##   23.93939  0.8286612  0.5997504  0.4513640
##   24.09091  0.8284755  0.5999308  0.4512624
##   24.24242  0.8284755  0.5999308  0.4512624
##   24.39394  0.8284755  0.5999308  0.4512624
##   24.54545  0.8284755  0.5999308  0.4512624
##   24.69697  0.8284755  0.5999308  0.4512624
##   24.84848  0.8284755  0.5999308  0.4512624
##   25.00000  0.8284872  0.5999194  0.4513013
##   25.15152  0.8284872  0.5999194  0.4513013
##   25.30303  0.8284872  0.5999194  0.4513013
##   25.45455  0.8284872  0.5999194  0.4513013
##   25.60606  0.8284872  0.5999194  0.4513013
##   25.75758  0.8284872  0.5999194  0.4513013
##   25.90909  0.8284872  0.5999194  0.4513013
##   26.06061  0.8278972  0.6004841  0.4490157
##   26.21212  0.8278972  0.6004841  0.4490157
##   26.36364  0.8278972  0.6004841  0.4490157
##   26.51515  0.8278972  0.6004841  0.4490157
##   26.66667  0.8278972  0.6004841  0.4490157
##   26.81818  0.8278972  0.6004841  0.4490157
##   26.96970  0.8278972  0.6004841  0.4490157
##   27.12121  0.8278243  0.6005512  0.4491758
##   27.27273  0.8278243  0.6005512  0.4491758
##   27.42424  0.8278243  0.6005512  0.4491758
```

```
##   27.57576  0.8278243  0.6005512  0.4491758
##   27.72727  0.8278243  0.6005512  0.4491758
##   27.87879  0.8278243  0.6005512  0.4491758
##   28.03030  0.8276784  0.6006908  0.4496568
##   28.18182  0.8276784  0.6006908  0.4496568
##   28.33333  0.8276784  0.6006908  0.4496568
##   28.48485  0.8276784  0.6006908  0.4496568
##   28.63636  0.8276784  0.6006908  0.4496568
##   28.78788  0.8276784  0.6006908  0.4496568
##   28.93939  0.8276784  0.6006908  0.4496568
##   29.09091  0.8275636  0.6008023  0.4498657
##   29.24242  0.8275636  0.6008023  0.4498657
##   29.39394  0.8275636  0.6008023  0.4498657
##   29.54545  0.8275636  0.6008023  0.4498657
##   29.69697  0.8275636  0.6008023  0.4498657
##   29.84848  0.8275636  0.6008023  0.4498657
##   30.00000  0.8273922  0.6009678  0.4499504
##   30.15152  0.8273922  0.6009678  0.4499504
##   30.30303  0.8273922  0.6009678  0.4499504
##   30.45455  0.8273922  0.6009678  0.4499504
##   30.60606  0.8273922  0.6009678  0.4499504
##   30.75758  0.8273922  0.6009678  0.4499504
##   30.90909  0.8273922  0.6009678  0.4499504
##   31.06061  0.8270185  0.6013281  0.4493417
##   31.21212  0.8270185  0.6013281  0.4493417
##   31.36364  0.8270185  0.6013281  0.4493417
##   31.51515  0.8270185  0.6013281  0.4493417
##   31.66667  0.8270185  0.6013281  0.4493417
##   31.81818  0.8270185  0.6013281  0.4493417
##   31.96970  0.8270185  0.6013281  0.4493417
##   32.12121  0.8266501  0.6016825  0.4480991
##   32.27273  0.8266501  0.6016825  0.4480991
##   32.42424  0.8266501  0.6016825  0.4480991
##   32.57576  0.8266501  0.6016825  0.4480991
##   32.72727  0.8266501  0.6016825  0.4480991
##   32.87879  0.8266501  0.6016825  0.4480991
##   33.03030  0.8266003  0.6017313  0.4481428
##   33.18182  0.8266003  0.6017313  0.4481428
##   33.33333  0.8266003  0.6017313  0.4481428
##   33.48485  0.8266003  0.6017313  0.4481428
##   33.63636  0.8266003  0.6017313  0.4481428
##   33.78788  0.8266003  0.6017313  0.4481428
##   33.93939  0.8266003  0.6017313  0.4481428
##   34.09091  0.8265936  0.6017394  0.4481190
##   34.24242  0.8265936  0.6017394  0.4481190
##   34.39394  0.8265936  0.6017394  0.4481190
##   34.54545  0.8265936  0.6017394  0.4481190
##   34.69697  0.8265936  0.6017394  0.4481190
##   34.84848  0.8265936  0.6017394  0.4481190
##   35.00000  0.8265983  0.6017349  0.4481142
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was ncomp = 34.09091.
```




```
## [1] 0.8265936
```

MARS (Multivariate Adaptive Regression Splines) :

MARS builds piecewise linear regressions, called splines, for different ranges of the data. It automatically selects which variables to use and where to place knots (breakpoints) in the data, making it ideal for datasets with complex, nonlinear relationships. MARS is often used when the relationship between variables is unknown or not easily captured by linear models.

```
## Loading required package: earth
```

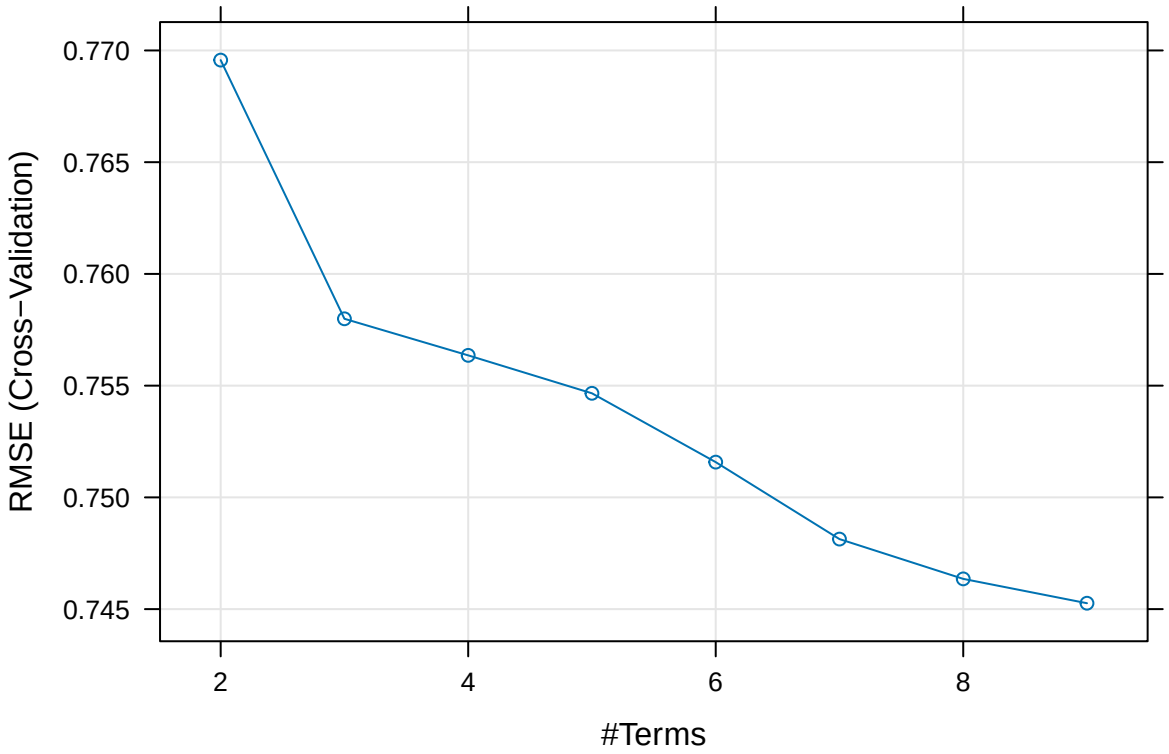
```
## Loading required package: Formula
```

```
## Loading required package: plotmo
```

```
## Loading required package: plotrix
```

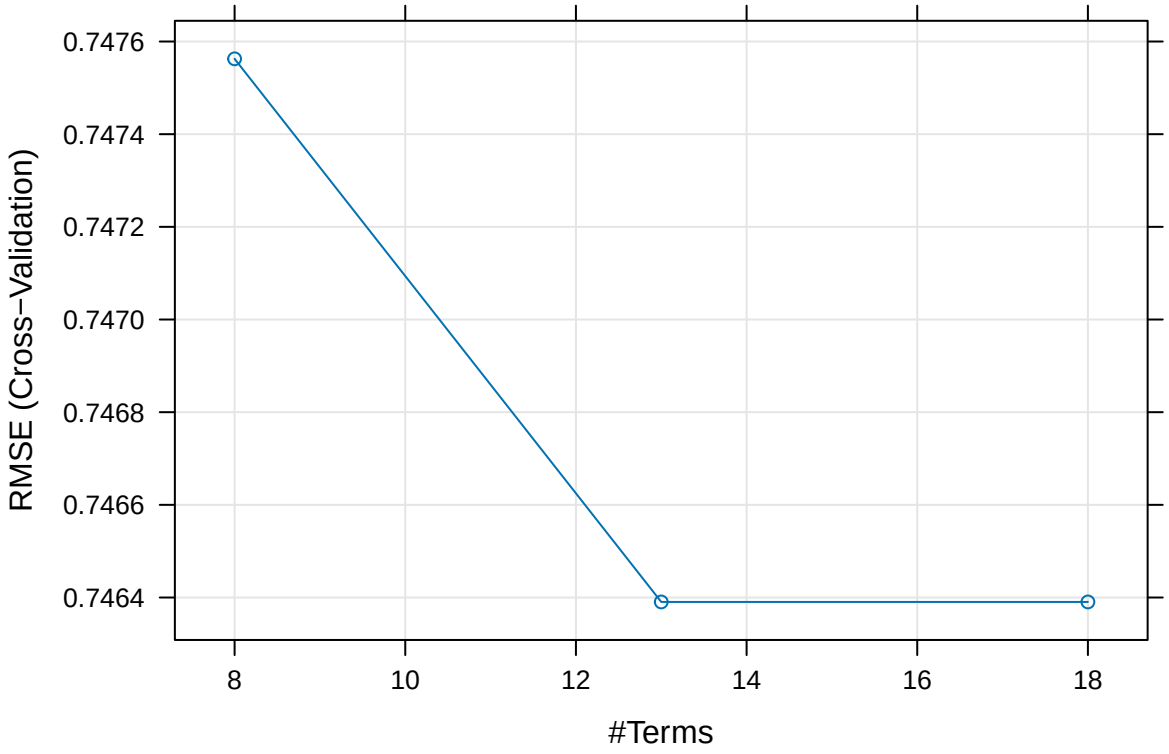
```
## Multivariate Adaptive Regression Spline
##
## 47249 samples
##    15 predictor
##
## No pre-processing
```

```
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 42524, 42524, 42524, 42524, 42524, 42524, ...
## Resampling results across tuning parameters:
##
##   nprune  RMSE       Rsquared   MAE
##   2       0.7695689  0.6544885  0.2730574
##   3       0.7579922  0.6647403  0.2850836
##   4       0.7563556  0.6661803  0.3146940
##   5       0.7546574  0.6677169  0.3085344
##   6       0.7515780  0.6704261  0.3283376
##   7       0.7481345  0.6734555  0.3071985
##   8       0.7463531  0.6750145  0.3231592
##   9       0.7452642  0.6759393  0.3295282
##
## Tuning parameter 'degree' was held constant at a value of 1
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were nprune = 9 and degree = 1.
```




```
## Multivariate Adaptive Regression Spline
##
## 47249 samples
##    15 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
```

```
## Summary of sample sizes: 42524, 42524, 42524, 42525, 42524, 42524, ...
## Resampling results across tuning parameters:
##
##   nprune  RMSE       Rsquared   MAE
##    8      0.7475627  0.6742777  0.3243684
##   13      0.7463905  0.6752750  0.3300340
##   18      0.7463905  0.6752750  0.3300340
##
## Tuning parameter 'degree' was held constant at a value of 1
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were nprune = 13 and degree = 1.
```




```
## [1] 0.7463905
```

Extracting RMSE for All Models:

The RMSE values are directly extracted and summarized for easy comparison in a data.frame.

Model	Hyperparameter	CV_RMSE
OLS	None	0.827
PLS	ncomp	0.827
LASSO	alpha=1, lambda	0.826
glmnet	alpha,Lambda	0.827

Model	Hyperparameter	CV_RMSE
PCR	ncomp	0.827
MARS	degree = 1 , nprune	0.746

iv. (10 points) Debrief. For your best predictions, describe your approach, e.g., did you examine interactions? did you use any type of model stacking? what was your secret sauce? Did you have any problems during the modeling process? If so, how did you overcome those?

The best performance in my analysis came from using MARS models. This model consistently provided the lowest RMSE during multiple runs, making them my top performers. While I did experiment with interactions and model stacking, I chose to keep the model relatively simple to maintain interpretability and avoid overfitting. Instead of adding more layers of complexity, I focused on feature engineering, which became my "secret sauce" for optimizing the model's performance.

Specifically, I crafted custom variables based on the behavioral data available. Some of the key features I introduced included:

avg_pageviews_per_visit – To capture user engagement patterns. days_between_first_last_visit – A measure of how quickly users return to the platform.

max_time_since_last – To account for dormant periods between visits. mostCommonBrowser, mostCommonOS, mostCommonDay – Indicators of user preferences across technology and time.

These new variables enhanced the predictive power of the model, providing deeper insights into user behavior. I believe that this feature engineering step was pivotal in boosting model accuracy and stability.

Challenges and Solutions: During the modeling process, I faced performance issues—with some models running for hours without completion as the complexity increased. To overcome this, I took several steps:

Dimensionality Reduction: I pruned less relevant features and focused on engineered variables that provided higher information gain.

Optimized Code: I adjusted hyperparameters and reduced batch sizes where appropriate to avoid excessive computation. Additionally, I monitored resource utilization to prevent bottlenecks during execution.

By maintaining a balance between simplicity and precision, I achieved a high-performing model without compromising speed or interpretability.