Calgary Citizen Satisfaction Survey Analysis Report

Deepika Gollamandala

Riya Chevli

Romith Bondada

DATA 606
Statistical Methods in Data Science

Contents

1. Introduction

Citizen satisfaction is a crucial metric for any municipal government seeking to deliver high-quality public services and maintain community trust. The City of Calgary conducts a biennial Citizen Satisfaction Survey to capture residents' attitudes toward a wide range of city operations—including public transit, road maintenance, waste management, and overall quality of life. This project uses the City of Calgary's 2018–2021 Citizen Satisfaction Survey dataset to explore the factors that influence residents' satisfaction levels and to uncover demographic patterns that may highlight areas for improvement.

The primary goal of this analysis is to identify how socio-demographic characteristics (such as income, homeownership status, age, and education) and perceptions of specific municipal services impact overall satisfaction. By understanding these relationships, City of Calgary decision-makers can prioritize resources, target service enhancements, and tailor communication strategies to address the most significant drivers of satisfaction.

While raw satisfaction scores provide a snapshot of resident sentiment, they do not reveal the underlying factors that drive these perceptions. For example, higher-income households may report higher satisfaction overall, but their expectations for service quality may differ from those of lower-income groups. Similarly, homeowners and renters might experience city services in distinct ways that affect their ratings. Without a deeper exploration of these dimensions, municipal leaders risk making decisions that do not serve all communities equitably. I selected this topic for several reasons. First, the City of Calgary is a rapidly growing urban center with diverse neighbourhoods and socio-economic profiles; understanding citizen perceptions is vital for sustainable development. Second, the 2018–2021 dataset offers a rich blend of demographic and service-specific variables, enabling a multifaceted analysis. Finally, improving public service delivery through data-driven insights is a pressing challenge in municipal governance, one that directly impacts quality of life and civic engagement.

The survey includes approximately 3,500 respondents per year, representing various wards and quadrants of Calgary. Variables encompass overall satisfaction ratings (q11a), categorical measures of household income (q39), tenure type (q34), education level (q38), age (q30), and ratings of specific services (questions q19_1 to q19_8). Calgary's municipal budget is spread across numerous services—public transit, road repair, leisure facilities, and environmental programs. Resource constraints require that city administrators understand which services most strongly influence resident satisfaction.

This report will employ exploratory data analysis to characterize distributions, inferential statistics to test group differences, and predictive modelling to determine key drivers of satisfaction. By integrating demographic insights with service perceptions, this analysis aims to provide actionable

recommendations to enhance citizen satisfaction and inform strategic planning within the City of Calgary.

1.1 Objective

One of the objectives of this study is to identify the demographic factors that influence an individual's satisfaction with life in Calgary and to develop a model for estimating satisfaction based on these demographic characteristics. A secondary objective is to analyze which municipal services, including public transit, infrastructure, safety, and recreational programs, significantly impact overall life satisfaction.By pinpointing the driving factors of satisfaction and their relative importance, the study aims to deliver actionable insights to enhance the City of Calgary's strategic planning, optimize resource allocation, and inform evidence-based policy development. These findings will support improvements in municipal service delivery, foster greater citizen well-being, and contribute to a more resilient and inclusive urban environment.

1.2 Dataset

We are using the data from **Fall Survey of Calgarians (2021) - Dataset** that is available at data.calgary.ca. It is open data published by the city of Calgary and is free to use under "Open Government License - City of Calgary". A dataset preview can be found at the following link: Fall Survey of Calgarians (2021). More information about the variable can be found at: Variable information

The original dataset consists of approximately 10,000 responses and 139 variables, each representing a question asked in the survey. For the purpose of this analysis, we focused exclusively on responses from the year 2021, resulting in a filtered dataset of around 2,500 observations. Out of the available variables, we selected 39 that are most relevant to our evaluation objectives. These selected variables include responses to questions related to city services, public perceptions, and demographic characteristics, which serve as key inputs in our predictive modeling and analysis.

The questions that were considered for our analysis in Part -1

1. In which quadrant of the city does the survey participant reside in?
2. What is the annual income of the survey participant?
3. Do they own the current place of residence
4. How many years did they live in Calgary?
5. What is their highest level of schooling/ education that they attended?
6. Number of children under the age of 18 living with them

7. Do they consider themselves a member of visual minority
8. Gender
9. Year of birth
10. Overall rating for quality of life in Calgary (OUR RESPONSE VARIABLE)

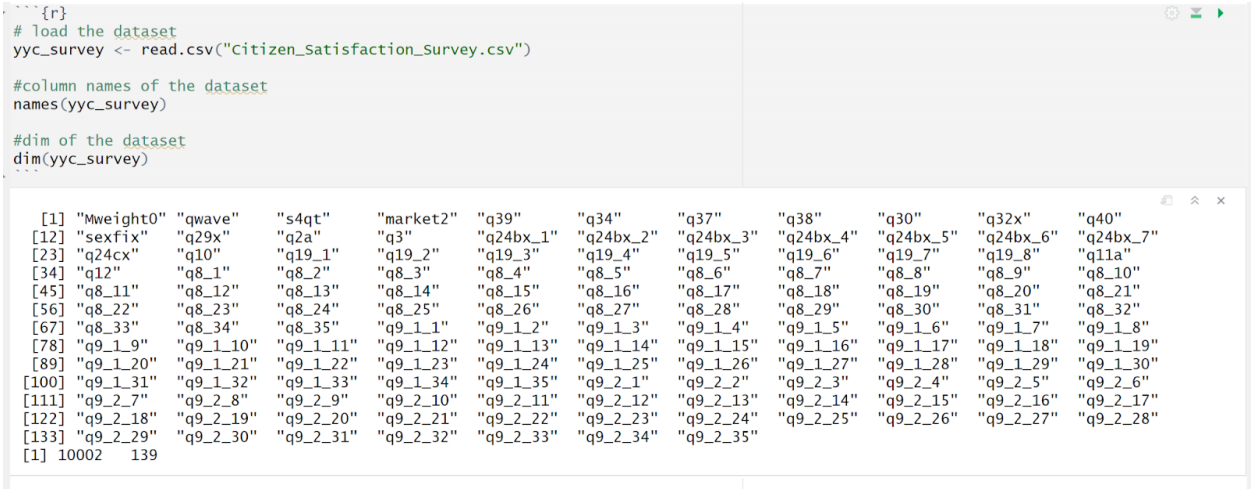
Another subset of questions was regarding their satisfaction ratings about the services provided by the city of Calgary. These are the exploratory variables considered for the analysis in Part 2. The services provided are listed below.

1. Calgary Police Service
2. Calgary Fire Department
3. 911
4. Calgary Fire Department
5. Protection from river flooding
6. Disaster planning and response
7. Residential garbage collection service
8. Residential Blue Cart recycling
9. Residential Green Cart service
10. The quality of drinking water
11. Transportation planning
12. Calgary Transit including bus and CTrain service
13. City operated roads and infrastructure
14. Road maintenance including pothole repairs
15. Spring road cleaning
16. Snow removal
17. Traffic flow management
18. Bylaw services for things such as noise complaints, fire pits and weeds
19. Animal control services for stray animals and pet licensing
20. Calgary's parks, playgrounds and other open spaces
21. Community services such as support for community associations and not for profit groups
22. Social services for individuals such as seniors or youth

23. Affordable housing for low
24. City operated recreation PROGRAMS such as swimming lessons
25. City operated recreation FACILITIES such as pools, leisure centres, and golf courses
26. Support for arts and culture including festivals
27. On-street bikeways
28. Calgary's pathway system
29. City land use planning
30. Development and building inspections and permits
31. Business licenses and inspections
32. City growth management
33. Downtown revitalization
34. Property tax assessment
35. City of Calgary website
36. 311 service

2. Data Pre-Processing

First, we will import the dataset. We check the column names, dimensions.



```{r}
# load the dataset
yyc_survey <- read.csv("Citizen_Satisfaction_Survey.csv")

#column names of the dataset
names(yyc_survey)

#dim of the dataset
dim(yyc_survey)
```

```
  [1] "Mweight0" "qwave"    "s4qt"     "market2"  "q39"      "q34"      "q37"      "q38"      "q30"      "q32x"     "q40"
 [12] "sexfix"   "q29x"     "q2a"      "q3"       "q24bx_1"  "q24bx_2"  "q24bx_3"  "q24bx_4"  "q24bx_5"  "q24bx_6"  "q24bx_7"
 [23] "q24cx"    "q10"      "q19_1"    "q19_2"    "q19_3"    "q19_4"    "q19_5"    "q19_6"    "q19_7"    "q19_8"    "q11a"
 [34] "q12"      "q8_1"     "q8_2"     "q8_3"     "q8_4"     "q8_5"     "q8_6"     "q8_7"     "q8_8"     "q8_9"     "q8_10"
 [45] "q8_11"    "q8_12"    "q8_13"    "q8_14"    "q8_15"    "q8_16"    "q8_17"    "q8_18"    "q8_19"    "q8_20"    "q8_21"
 [56] "q8_22"    "q8_23"    "q8_24"    "q8_25"    "q8_26"    "q8_27"    "q8_28"    "q8_29"    "q8_30"    "q8_31"    "q8_32"
 [67] "q8_33"    "q8_34"    "q8_35"    "q9_1_1"   "q9_1_2"   "q9_1_3"   "q9_1_4"   "q9_1_5"   "q9_1_6"   "q9_1_7"   "q9_1_8"
 [78] "q9_1_9"   "q9_1_10"  "q9_1_11"  "q9_1_12"  "q9_1_13"  "q9_1_14"  "q9_1_15"  "q9_1_16"  "q9_1_17"  "q9_1_18"  "q9_1_19"
 [89] "q9_1_20"  "q9_1_21"  "q9_1_22"  "q9_1_23"  "q9_1_24"  "q9_1_25"  "q9_1_26"  "q9_1_27"  "q9_1_28"  "q9_1_29"  "q9_1_30"
[100] "q9_1_31"  "q9_1_32"  "q9_1_33"  "q9_1_34"  "q9_1_35"  "q9_2_1"   "q9_2_2"   "q9_2_3"   "q9_2_4"   "q9_2_5"   "q9_2_6"
[111] "q9_2_7"   "q9_2_8"   "q9_2_9"   "q9_2_10"  "q9_2_11"  "q9_2_12"  "q9_2_13"  "q9_2_14"  "q9_2_15"  "q9_2_16"  "q9_2_17"
[122] "q9_2_18"  "q9_2_19"  "q9_2_20"  "q9_2_21"  "q9_2_22"  "q9_2_23"  "q9_2_24"  "q9_2_25"  "q9_2_26"  "q9_2_27"  "q9_2_28"
[133] "q9_2_29"  "q9_2_30"  "q9_2_31"  "q9_2_32"  "q9_2_33"  "q9_2_34"  "q9_2_35"
[1] 10002   139
```

There are 10002 records and 139 columns in the original dataset. But we only use a subset of the records and columns for our study here.

Let us inspect the dataset further.

```{r}
head(yyc_survey)
```

Description: df [6 × 139]

	Mweight0 <dbl>	qwave <chr>	s4qt <int>	market2 <int>	q39 <int>	q34 <int>	q37 <int>	q38 <int>	q30 <int>
1	0.51	Year-2021	4	10	4	1	9	3	6
2	0.60	Year-2021	1	8	1	1	9	2	6
3	0.70	Year-2021	2	12	7	1	8	2	6
4	0.49	Year-2021	3	6	9	1	8	3	6
5	0.77	Year-2021	1	11	4	1	10	2	5
6	0.67	Year-2021	1	6	1	2	1	3	6

6 rows | 1-10 of 139 columns

The dataset consists of data from the surveys conducted in the years 2018, 2019, 2020 and 2021.

```{r}
table(yyc_survey$qwave)
```

```
Year-2018 Year-2019 Year-2020 Year-2021
     2500      2502      2500      2500
```

For the current study we restricted ourselves to the 2021 survey.

```{r}
library(dplyr)


# filter to include only the survey responses from year 2021
filtered_df <- yyc_survey %>% filter(qwave == 'Year-2021')

# checking to ensure we only have 2021 survey data
dim(filtered_df)
unique(filtered_df$qwave)
```

```
[1] 2500  139
[1] "Year-2021"
```

Select the columns to include only the demographic features and our response variable

```{r}
filtered_df <- filtered_df %>% select(s4qt, q39, q34, q37, q38, q32x, q40, q29x, q30, q2a)

```

```{r}
dim(filtered_df)
```

```
[1] 2500   10
```

Rename the column names for ease

```{r}
renamed_df <- filtered_df %>% rename("Quadrant" = s4qt, "Income" = q39, "Tenancy" = q34, "Years_in_yyc" = q37, "Education" = q38, "Children" = q32x, "Minority" = q40, "Gender" = q29x, "Age" = q30, "Satisfaction_level" = q2a )

```

```{r}
names(renamed_df)
```

```
 [1] "Quadrant"           "Income"             "Tenancy"
 [4] "Years_in_yyc"       "Education"          "Children"
 [7] "Minority"           "Gender"             "Age"
[10] "Satisfaction_level"
```

Check the levels in our response variable

```{r}
table(renamed_df$Satisfaction_level)
```

```
  1   2   3   4   5   6   7   8   9  10  11
 18  18  45  69 169 225 625 907 278 145   1
```

We excluded the single record where the Satisfaction Rating was 11, as this value indicated that the respondent was unsure how to rate the quality of life in Calgary. This clarification was provided in the metadata file, confirming that a score of 11 does not represent an actual satisfaction level. Removing this entry ensures the integrity and consistency of our dataset.

```{r}
#remove the row with satisfaction Rating of "11"
renamed_df <- renamed_df %>% filter(Satisfaction_level != 11)

# inspect the levels
table(renamed_df$Satisfaction_level)
```

```
  1   2   3   4   5   6   7   8   9  10
 18  18  45  69 169 225 625 907 278 145
```

Next, we transform the Satisfaction_level column into a binary column with levels "Yes" and "No". The new column's name is "Satisfaction". When the value of Satisfaction_level is <= 5,value

is "Not Satisfied" or "No" and when Satisfaction_level is >5, it is "Satisfied" or "Yes". We also rename our data frame to a more intuitive name ‘demographic_df’ for ease.

```{r}
demographic_df <- renamed_df %>% mutate(Satisfaction = ifelse(Satisfaction_level <= 5, "No", "Yes"))

# check the column names
names(demographic_df)

# check the levels of our new column.
table(demographic_df$Satisfaction)
```

```
 [1] "Quadrant"           "Income"             "Tenancy"
 [4] "Years_in_yyc"       "Education"          "Children"
 [7] "Minority"           "Gender"             "Age"
[10] "Satisfaction_level" "Satisfaction"

  No  Yes
 319 2180
```

Let us check the data types of the variables

```
str(demographic_df)
```

```
 [1] "Quadrant"           "Income"             "Tenancy"
 [4] "Years_in_yyc"       "Education"          "Children"
 [7] "Minority"           "Gender"             "Age"
[10] "Satisfaction_level" "Satisfaction"

  No  Yes
 319 2180
'data.frame':   2499 obs. of  11 variables:
 $ Quadrant          : int  4 1 2 3 1 1 1 2 1 1 ...
 $ Income            : int  4 1 7 9 4 1 4 7 9 2 ...
 $ Tenancy           : int  1 1 1 1 1 2 1 1 1 1 ...
 $ Years_in_yyc      : int  9 9 8 8 10 1 11 8 3 7 ...
 $ Education         : int  3 2 2 3 2 3 3 2 2 3 ...
 $ Children          : int  2 2 2 2 2 2 2 2 1 2 ...
 $ Minority          : int  2 2 2 3 2 1 2 2 2 2 ...
 $ Gender            : int  1 2 1 1 1 2 1 1 1 1 ...
 $ Age               : int  6 6 6 6 5 6 6 5 4 6 ...
 $ Satisfaction_level: int  6 6 8 7 8 7 7 10 8 5 ...
 $ Satisfaction      : chr  "Yes" "Yes" "Yes" "Yes" ...
```

All our exploratory variables are categorical but are in wrong datatype int. We need to first convert them to be categorical.

```{r}
demographic_df <- demographic_df %>%
  mutate(across(where(is.integer), as.factor))

str(demographic_df)
```

```
'data.frame':   2499 obs. of  11 variables:
 $ Quadrant          : Factor w/ 4 levels "1","2","3","4": 4 1 2 3 1 1 1 2 1 1 ...
 $ Income            : Factor w/ 9 levels "1","2","3","4",..: 4 1 7 9 4 1 4 7 9 2 ...
 $ Tenancy           : Factor w/ 5 levels "1","2","3","4",..: 1 1 1 1 1 2 1 1 1 1 ...
 $ Years_in_yyc      : Factor w/ 12 levels "1","2","3","4",..: 9 9 8 8 10 1 11 8 3 7 ...
 $ Education         : Factor w/ 4 levels "1","2","3","4": 3 2 2 3 2 3 3 2 2 3 ...
 $ Children          : Factor w/ 3 levels "1","2","3": 2 2 2 2 2 2 2 2 1 2 ...
 $ Minority          : Factor w/ 3 levels "1","2","3": 2 2 2 3 2 1 2 2 2 2 ...
 $ Gender            : Factor w/ 4 levels "1","2","3","7": 1 2 1 1 1 2 1 1 1 1 ...
 $ Age               : Factor w/ 7 levels "1","2","3","4",..: 6 6 6 6 5 6 6 5 4 6 ...
 $ Satisfaction_level: Factor w/ 10 levels "1","2","3","4",..: 6 6 8 7 8 7 7 10 8 5 ...
 $ Satisfaction      : chr  "Yes" "Yes" "Yes" "Yes" ...
```

We convert our response variable into a factor

```{r}
demographic_df <- demographic_df %>% mutate(Satisfaction = factor(Satisfaction, levels =
c("No", "Yes")))

str(demographic_df$Satisfaction)
```

```
 Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 1 ...
```

We check the proportion of classes in our dataset

```{r}
table(demographic_df$Satisfaction)
```

```
  No  Yes
 319 2180
```

3. Exploratory Data Analysis:

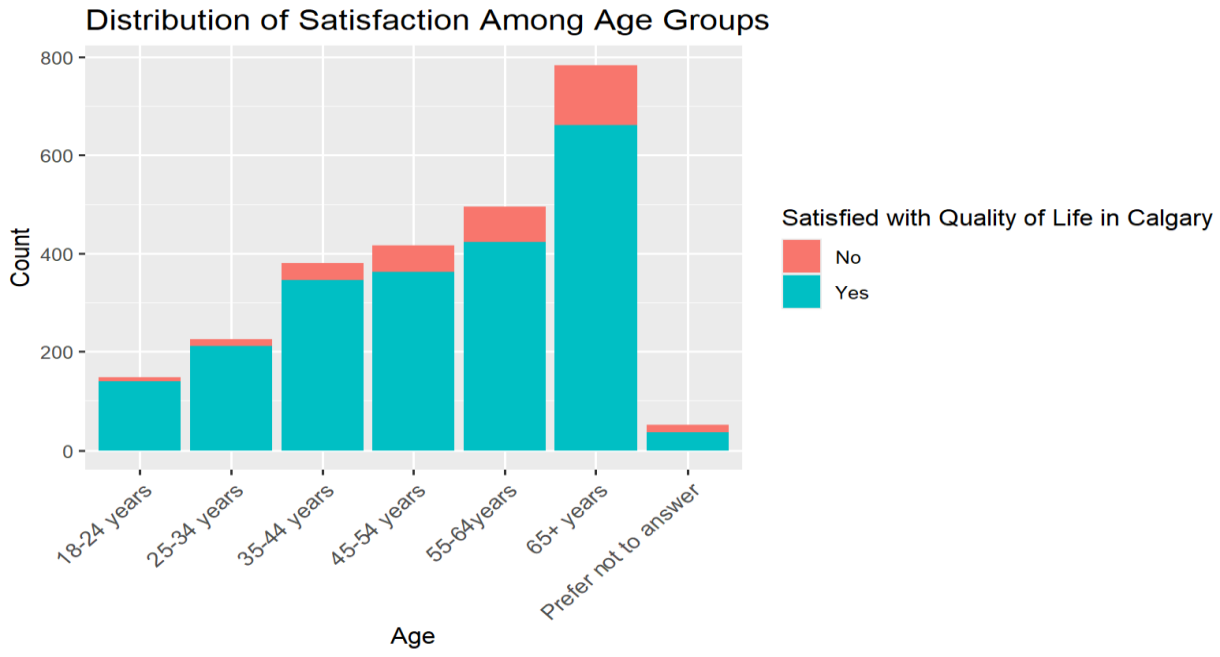
In this section, we focus on exploring the demographic variables of our dataset in detail. These variables play a crucial role in our analysis, as they provide valuable insights into how different population groups perceive their quality of life in Calgary. By examining factors such as age, gender, education level, and income, we aim to understand how these characteristics influence overall life satisfaction.

3.1 Distribution and proportion of satisfaction across age groups:

The distribution analysis aims to understand how satisfaction levels are spread within each age group, identifying central tendencies (e.g., most common satisfaction score) and variability (e.g., spread of scores).

In the original dataset, age was coded numerically from 1 to 7. For analysis, we recorded these values into categorical age groups (e.g., "<18", "18–24", "25–34", etc.) to improve interpretability and support meaningful comparisons across age demographics.

In our first graph, the distribution of satisfaction among the age group x-axis represents Age groups categorized as "18-24 years", "25-34 years" etc and the Y-axis represents the number of respondents, ranging from 0 to 800. A legend indicating "Satisfied with Quality of Life in Calgary" (teal = Yes, coral = No).

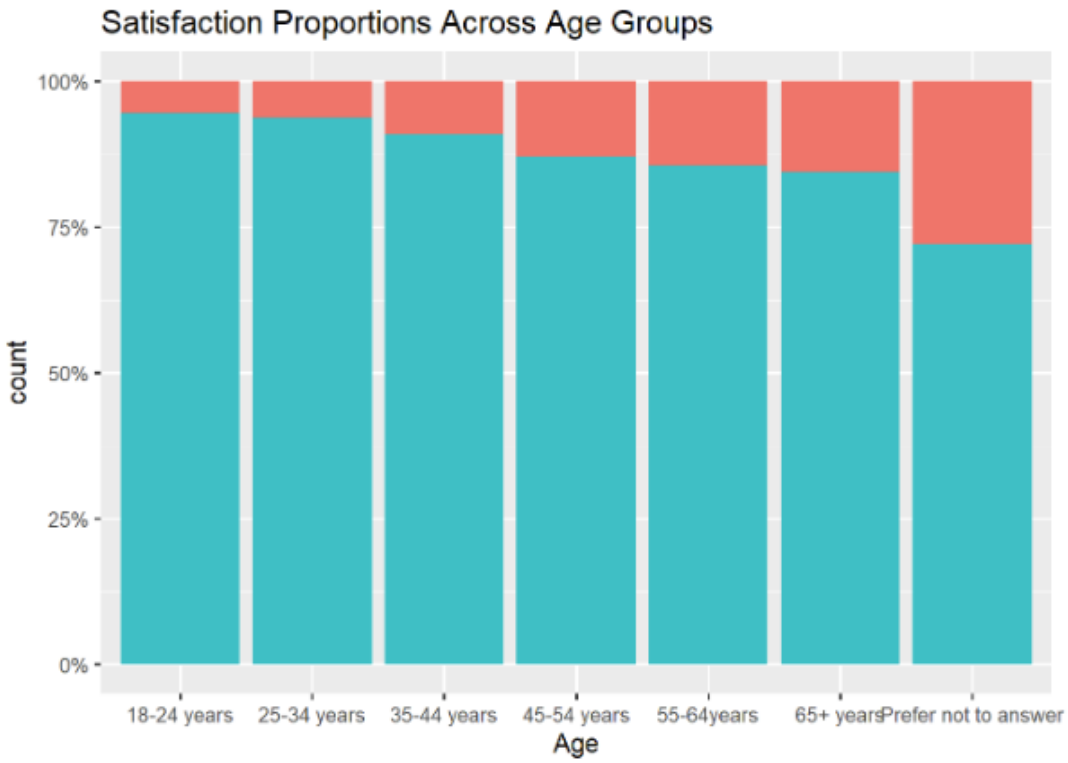


Interpretation of First Graph:
This graph shows the original data, and it's clear that the "65+ years" and "55–64 years" age groups have the most people in the survey. They also have the highest number of people who are not satisfied, even though many in these groups are satisfied too.

Younger age groups (18–34 years) have fewer people, but most of them report being satisfied. The "Prefer not to answer" group is small and has a mix of satisfied and unsatisfied responses.

In this second graph X-axis defines different categories of age group and Y-axis defines Proportion (0% to 100%) of that.

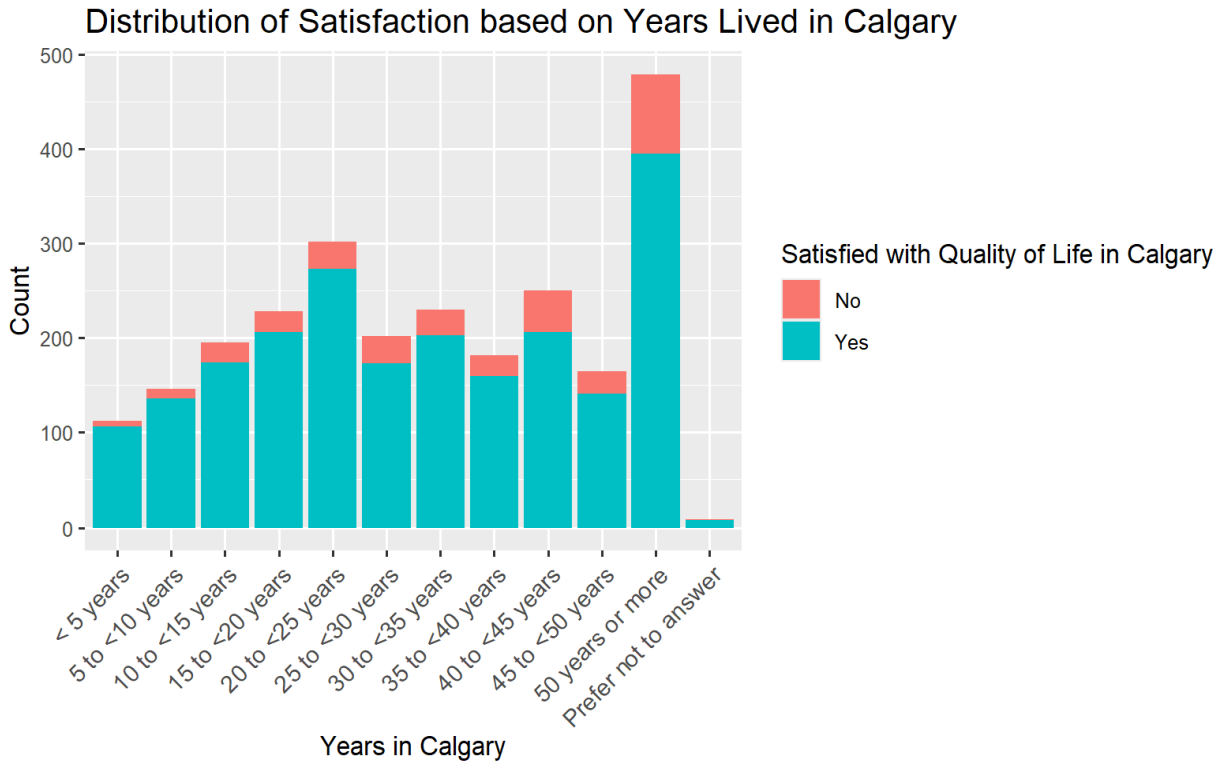
This graph highlights that satisfaction tends to be highest among younger adults (18-34 years) and decreases with age, with the oldest group (65+) and those unwilling to disclose age showing the highest dissatisfaction proportions. This suggests potential age-related factors influencing satisfaction.



3.2 Distribution and proportion of satisfaction across Years live in Calgary :

In the original dataset, years_in_yyc was coded numerically from 1 to 12. For analysis, we recorded these values into categorical groups (e.g., "<5 years", "5 to <10 years" etc.) to improve interpretability and support meaningful comparisons across this demographic.

In our first graph, the distribution of satisfaction based on years lived in Calgary x-axis represents Categories of years lived in Calgary: "<5 years", "5 to <10 years", "10 to <15 years" etc and the Y-axis represents the number of respondents, ranging from 0 to 500. A legend indicating "Satisfied with Quality of Life in Calgary" (teal = Yes, coral = No).

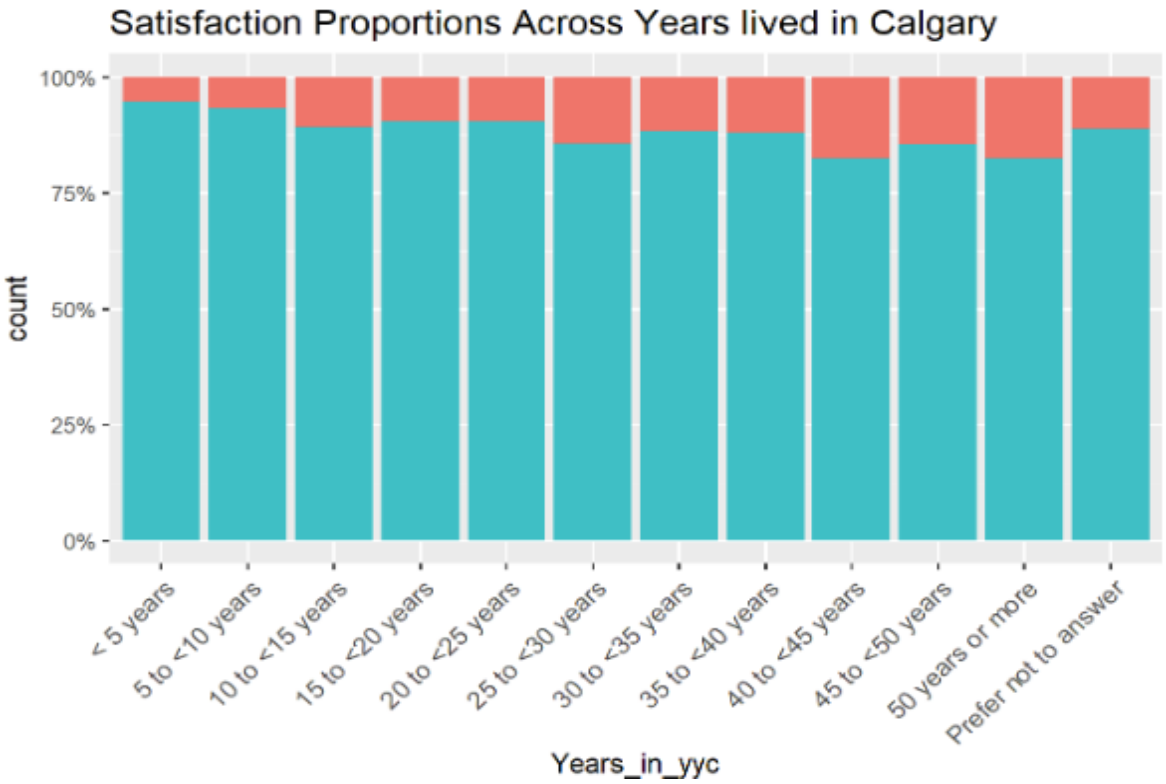


Interpretation of First graph:

The "50 years or more" group has the highest total count, with a large, satisfied portion and a large, dissatisfied portion.
Overall, most people across all groups are satisfied, but satisfaction seems to peak among short-term residents, while long-term residents show more dissatisfaction.

In this second graph X-axis defines different categories of years lived in Calgary and Y-axis defines Proportion (0% to 100%) of that.

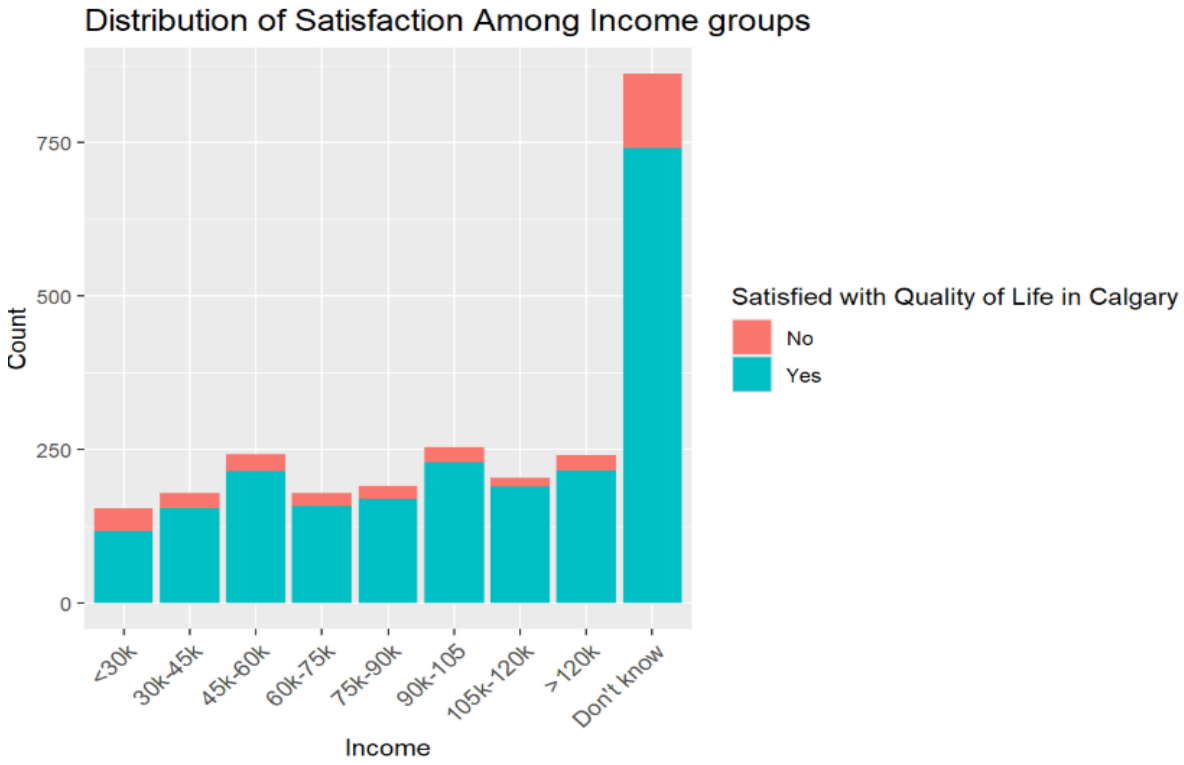
The proportional view shows that newer residents (less than 10 years) and some mid-term residents report the highest satisfaction levels, while long-term residents lived in Calgary show the highest levels of dissatisfaction.



3.3 Distribution and proportion of satisfaction among income groups:

In the original dataset, income was coded numerically from 1 to 9. For analysis, we recorded these values into categorical groups (e.g., "<30k", "30k to 45k" etc.) to improve interpretability and support meaningful comparisons across these demographics.

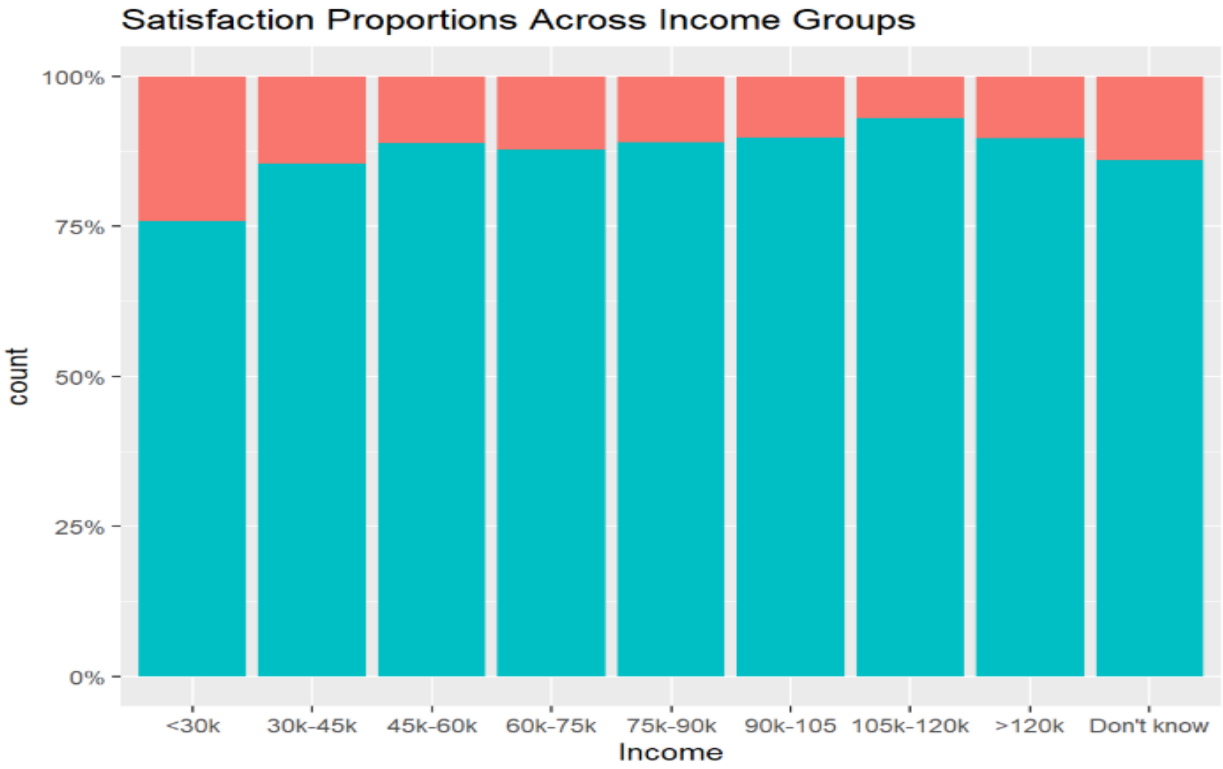
In our first graph, the distribution of satisfaction based among the income group x-axis represents Income groups categorized as "<30k", "30k-45k", "45k-60k" etc and the Y-axis represents the number of respondents, ranging from 0 to 750. A legend indicating "Satisfied with Quality of Life in Calgary" (teal = Yes, coral = No).



Interpretation

The "Don't know" group has the highest total count with a large satisfied portion and a larger dissatisfied portion.
The "<30k" group has a moderate count, with a significant dissatisfied portion and satisfied portion.
Other income groups have counts ranging from 150-300, with a majority satisfied and varying dissatisfied portions.

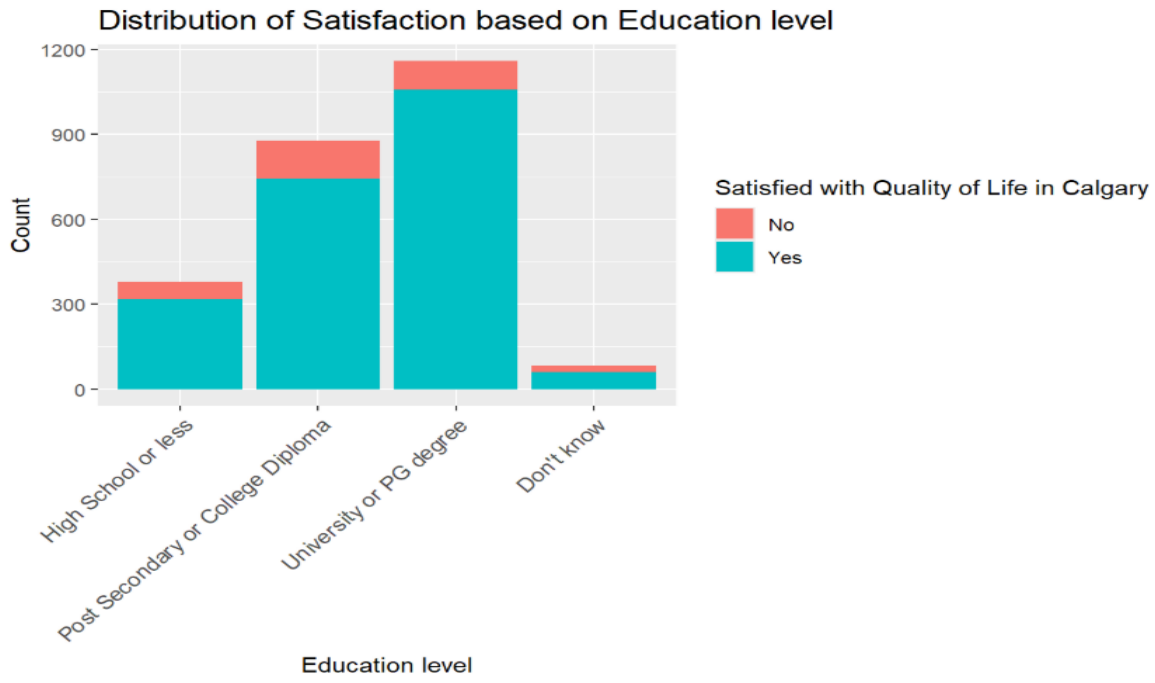
In this second graph, the x-axis represents Income groups categorized, and Y-axis defines the Proportion (0% to 100%) of that. The proportional view highlights that satisfaction increases with income up to a high level (e.g., 75k-90k and above), with the lowest income group (<30k) showing the highest dissatisfaction. The "Don't know" group maintains a high satisfaction rate despite its large size.



3.4 Distribution and proportion of satisfaction based on Education level:

In the original dataset, income was coded numerically from 1 to 4. For analysis, we recorded these values into categorical groups ("High School or less", "Post Secondary or College Diploma", "University or PG degree", and "Don't know") to improve interpretability and support meaningful comparisons across these demographics.

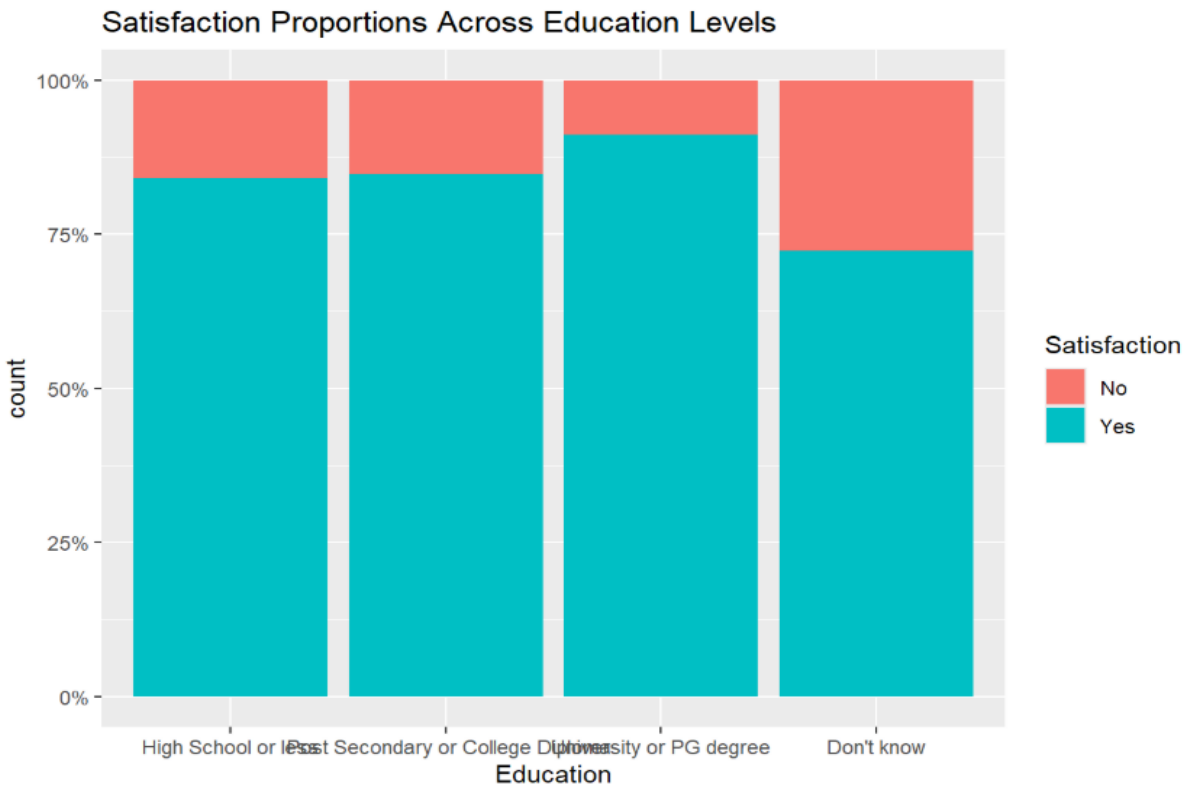
In our first graph, the distribution of satisfaction based on education level x-axis represents Education levels categorized groups "High School or less", "Post Secondary or College Diploma", "University or PG degree", and "Don't know" and the Y-axis represents the number of respondents, ranging from 0 to 1200. A legend indicating "Satisfied with Quality of Life in Calgary" (teal = Yes, coral = No).



Interpretation

The "University or PG degree" group has the highest total count, with a large, satisfied portion and a smaller dissatisfied portion.
The "High School or less" group has a moderate count, with a significantly dissatisfied portion and a satisfied portion.
The "Post Secondary or College Diploma" group has a count of around 600-700, with a majority satisfied and a smaller dissatisfied portion.
The "Don't know" group has the lowest count, with a balanced split between satisfied and dissatisfied.

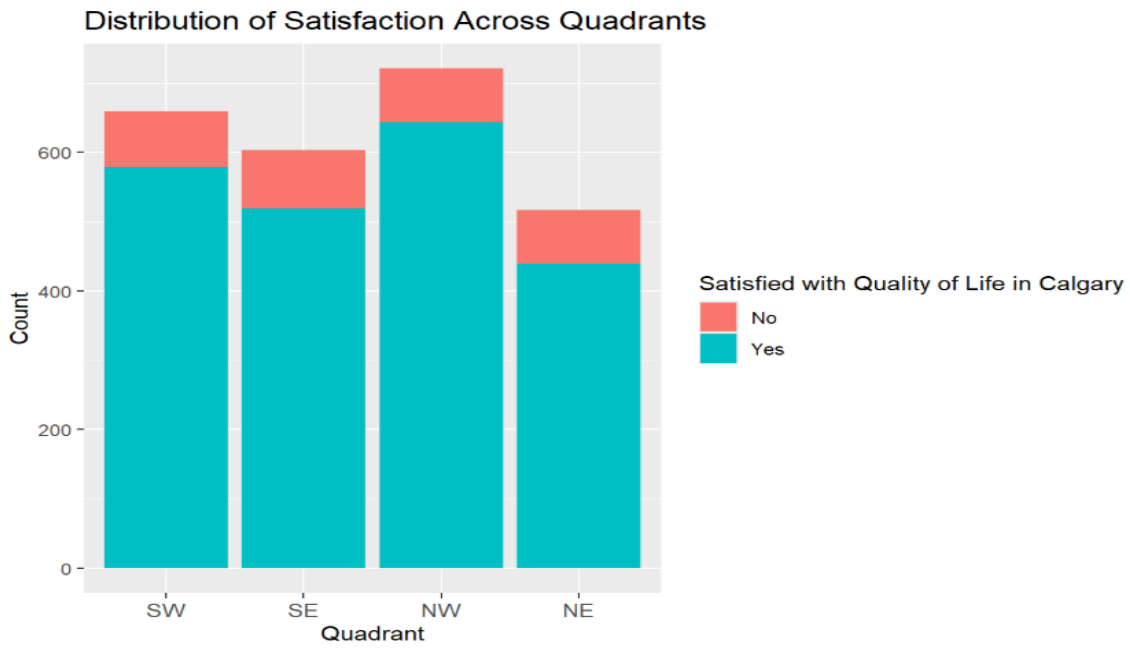
In this second graph, the x-axis represents Education levels categorized, and Y-axis defines the Proportion (0% to 100%) of that. The proportional view highlights that satisfaction is highest among those with post-secondary education (college diploma or university degree) while those with only high school or less education show a higher dissatisfaction rate. The "Don't know" group stands out with the highest dissatisfaction proportion, possibly indicating uncertainty or non-response bias.



3.5 Distribution of satisfaction across Quadrant:

In the original dataset, quadrants were coded numerically from 1 to 4. For analysis, we recorded these values into categorical groups (SW, SE, NW, NE) to improve interpretability and support meaningful comparisons across these demographics.

In our graph, the distribution of satisfaction across quadrant x-axis represents quadrant categorized groups SW,NW,SE,NE and the Y-axis represents the number of respondents, ranging from 0 to 700. A legend indicating "Satisfied with Quality of Life in Calgary" (teal = Yes, coral = No).



Interpretation of graph:

The "NW" quadrant has the highest total count with a large satisfied portion and a smaller dissatisfied portion .
The "SW" quadrant has a moderate count with a significant dissatisfied portion and a satisfied portion.
The "SE" quadrant has a count of around 500-600, with a notable dissatisfied portion and a satisfied portion.
The "NE" quadrant has the lowest total count with a smaller dissatisfied portion and a larger satisfied portion.

4. Analysis - 1

In this part of the analysis, we examine whether satisfaction levels can be estimated based on key demographic characteristics of survey respondents. The following factors are considered as exploratory variables in our analysis for this part of the study:

- **Quadrant:** The geographic quadrant of the city where the respondent resides.
- **Income:** The respondent's annual household income group.
- **Tenancy:** Whether the respondent owns or rents their home.
- **Years in YYC:** Duration of residence in Calgary.

- **Education:** The highest level of education attained by the respondent.
- **Children:** Presence of children living in the same household.
- **Minority Status:** Whether the respondent identifies as part of a visible minority group.
- **Gender:** The respondent's gender identity.
- **Age:** The respondent's age group.

The response variable in this study is Satisfaction_level which is a rating given by the survey respondents for quality of life in Calgary, measured on a scale of 1 to 10, were

- 1 = Very Poor Satisfaction
- 10 = Well Satisfied

To simplify analysis, **Satisfaction** is converted into a binary variable and takes the values:

- "Yes" – If Rating > 5 (indicating overall satisfaction)
- "No" – If Rating ≤ 5 (indicating dissatisfaction)

This transformation allows for classification-based modeling to predict satisfaction levels based on demographic features.

We will first discuss the Chi-Square statistical tests for Dependence between our response variable, Satisfaction and all the other exploratory variables. Then we will move on to discuss the various models we build

We will be primarily focusing on two models

- Logistic Regression
    - Full Logit model
    - Reduced Logit model - our base model
    - Logit model with increased weight for" No" and increased prob. threshold
    - Logit model with up-sampling
- Tree Classification
    - Tree model using train data
    - Tree model using up-sampled train data

4.1 Data Preparation for Analysis

First, we will import the dataset. We check the column names and dimensions.

```{r}
# load the dataset
yyc_survey <- read.csv("Citizen_Satisfaction_Survey.csv")

#column names of the dataset
names(yyc_survey)

#dim of the dataset
dim(yyc_survey)
```

```
  [1] "Mweight0" "qwave"    "s4qt"     "market2"  "q39"      "q34"      "q37"      "q38"      "q30"      "q32x"     "q40"
 [12] "sexfix"   "q29x"     "q2a"      "q3"       "q24bx_1"  "q24bx_2"  "q24bx_3"  "q24bx_4"  "q24bx_5"  "q24bx_6"  "q24bx_7"
 [23] "q24cx"    "q10"      "q19_1"    "q19_2"    "q19_3"    "q19_4"    "q19_5"    "q19_6"    "q19_7"    "q19_8"    "q11a"
 [34] "q12"      "q8_1"     "q8_2"     "q8_3"     "q8_4"     "q8_5"     "q8_6"     "q8_7"     "q8_8"     "q8_9"     "q8_10"
 [45] "q8_11"    "q8_12"    "q8_13"    "q8_14"    "q8_15"    "q8_16"    "q8_17"    "q8_18"    "q8_19"    "q8_20"    "q8_21"
 [56] "q8_22"    "q8_23"    "q8_24"    "q8_25"    "q8_26"    "q8_27"    "q8_28"    "q8_29"    "q8_30"    "q8_31"    "q8_32"
 [67] "q8_33"    "q8_34"    "q8_35"    "q9_1_1"   "q9_1_2"   "q9_1_3"   "q9_1_4"   "q9_1_5"   "q9_1_6"   "q9_1_7"   "q9_1_8"
 [78] "q9_1_9"   "q9_1_10"  "q9_1_11"  "q9_1_12"  "q9_1_13"  "q9_1_14"  "q9_1_15"  "q9_1_16"  "q9_1_17"  "q9_1_18"  "q9_1_19"
 [89] "q9_1_20"  "q9_1_21"  "q9_1_22"  "q9_1_23"  "q9_1_24"  "q9_1_25"  "q9_1_26"  "q9_1_27"  "q9_1_28"  "q9_1_29"  "q9_1_30"
[100] "q9_1_31"  "q9_1_32"  "q9_1_33"  "q9_1_34"  "q9_1_35"  "q9_2_1"   "q9_2_2"   "q9_2_3"   "q9_2_4"   "q9_2_5"   "q9_2_6"
[111] "q9_2_7"   "q9_2_8"   "q9_2_9"   "q9_2_10"  "q9_2_11"  "q9_2_12"  "q9_2_13"  "q9_2_14"  "q9_2_15"  "q9_2_16"  "q9_2_17"
[122] "q9_2_18"  "q9_2_19"  "q9_2_20"  "q9_2_21"  "q9_2_22"  "q9_2_23"  "q9_2_24"  "q9_2_25"  "q9_2_26"  "q9_2_27"  "q9_2_28"
[133] "q9_2_29"  "q9_2_30"  "q9_2_31"  "q9_2_32"  "q9_2_33"  "q9_2_34"  "q9_2_35"
[1] 10002   139
```

There are 10002 records and 139 columns in the original dataset. But we only use a subset of the records and columns for our study here. Let us inspect the dataset further.

```{r}
head(yyc_survey)
```

Description: df [6 × 139]

	Mweight0 <dbl>	qwave <chr>	s4qt <int>	market2 <int>	q39 <int>	q34 <int>	q37 <int>	q38 <int>	q30 <int>
1	0.51	Year-2021	4	10	4	1	9	3	6
2	0.60	Year-2021	1	8	1	1	9	2	6
3	0.70	Year-2021	2	12	7	1	8	2	6
4	0.49	Year-2021	3	6	9	1	8	3	6
5	0.77	Year-2021	1	11	4	1	10	2	5
6	0.67	Year-2021	1	6	1	2	1	3	6

6 rows | 1-10 of 139 columns

The dataset consists of data from the surveys conducted in the years 2018, 2019, 2020 and 2021.

```{r}
table(yyc_survey$qwave)
```

```
Year-2018 Year-2019 Year-2020 Year-2021
     2500      2502      2500      2500
```

For the current study we restricted ourselves to the 2021 survey.

```r
library(dplyr)


# filter to include only the survey responses from year 2021
filtered_df <- yyc_survey %>% filter(qwave == 'Year-2021')

# checking to ensure we only have 2021 survey data
dim(filtered_df)
unique(filtered_df$qwave)
```

```
[1] 2500  139
[1] "Year-2021"
```

Select the columns to include only the demographic features and our response variable

```r
filtered_df <- filtered_df %>% select(s4qt, q39, q34, q37, q38, q32x, q40, q29x, q30, q2a)
```

```r
dim(filtered_df)
```

```
[1] 2500   10
```

Rename the column names for ease

```r
renamed_df <- filtered_df %>% rename("Quadrant" = s4qt, "Income" = q39, "Tenancy" = q34,
"Years_in_yyc" = q37, "Education" = q38, "Children" = q32x, "Minority" = q40, "Gender" =
q29x, "Age" = q30, "Satisfaction_level" = q2a )
```

```r
names(renamed_df)
```

```
 [1] "Quadrant"           "Income"             "Tenancy"
 [4] "Years_in_yyc"       "Education"          "Children"
 [7] "Minority"           "Gender"             "Age"
[10] "Satisfaction_level"
```

Check the levels in our response variable

```r
table(renamed_df$Satisfaction_level)
```

```
  1   2   3   4   5   6   7   8   9  10  11
 18  18  45  69 169 225 625 907 278 145   1
```

We excluded the single record where the Satisfaction Rating was 11, as this value indicated that the respondent was unsure how to rate the quality of life in Calgary. This clarification was provided in the metadata file, confirming that a score of 11 does not represent an actual satisfaction level. Removing this entry ensures the integrity and consistency of our dataset.

```r
#remove the row with satisfaction Rating of "11"
renamed_df <- renamed_df %>% filter(Satisfaction_level != 11)

# inspect the levels
table(renamed_df$Satisfaction_level)
```

```
  1   2   3   4   5   6   7   8   9  10
 18  18  45  69 169 225 625 907 278 145
```

Next, we transform the Satisfaction_level column into a binary column with levels "Yes" and "No". The new column's name is "Satisfaction". When the value of Satisfaction_level is <= 5, value is "Not Satisfied" or "No" and when Satisfaction_level is >5, it is "Satisfied" or "Yes". We also rename our dataframe to a more intuitive name 'demographic_df' for ease.

```r
demographic_df <- renamed_df %>% mutate(Satisfaction = ifelse(Satisfaction_level <= 5, "No",
"Yes"))

# check the column names
names(demographic_df)

# check the levels of our new column.
table(demographic_df$Satisfaction)
```

```
 [1] "Quadrant"           "Income"             "Tenancy"
 [4] "Years_in_yyc"       "Education"          "Children"
 [7] "Minority"           "Gender"             "Age"
[10] "Satisfaction_level" "Satisfaction"

  No  Yes
 319 2180
```

Let us check the data types of the variables

```
str(demographic_df)
```

```
 [1] "Quadrant"           "Income"             "Tenancy"
 [4] "Years_in_yyc"       "Education"          "Children"
 [7] "Minority"           "Gender"             "Age"
[10] "Satisfaction_level" "Satisfaction"

  No  Yes
 319 2180
'data.frame':   2499 obs. of  11 variables:
 $ Quadrant          : int  4 1 2 3 1 1 1 2 1 1 ...
 $ Income            : int  4 1 7 9 4 1 4 7 9 2 ...
 $ Tenancy           : int  1 1 1 1 1 2 1 1 1 1 ...
 $ Years_in_yyc      : int  9 9 8 8 10 1 11 8 3 7 ...
 $ Education         : int  3 2 2 3 2 3 3 2 2 3 ...
 $ Children          : int  2 2 2 2 2 2 2 2 1 2 ...
 $ Minority          : int  2 2 2 3 2 1 2 2 2 2 ...
 $ Gender            : int  1 2 1 1 1 2 1 1 1 1 ...
 $ Age               : int  6 6 6 6 5 6 6 5 4 6 ...
 $ Satisfaction_level: int  6 6 8 7 8 7 7 10 8 5 ...
 $ Satisfaction      : chr  "Yes" "Yes" "Yes" "Yes" ...
```

All our exploratory variables are categorical but are in wrong datatype int. We need to first convert them to be categorical.

```
```{r}
demographic_df <- demographic_df %>%
  mutate(across(where(is.integer), as.factor))

str(demographic_df)
```
```

```
'data.frame':   2499 obs. of  11 variables:
 $ Quadrant          : Factor w/ 4 levels "1","2","3","4": 4 1 2 3 1 1 1 2 1 1 ...
 $ Income            : Factor w/ 9 levels "1","2","3","4",..: 4 1 7 9 4 1 4 7 9 2 ...
 $ Tenancy           : Factor w/ 5 levels "1","2","3","4",..: 1 1 1 1 1 2 1 1 1 1 ...
 $ Years_in_yyc      : Factor w/ 12 levels "1","2","3","4",..: 9 9 8 8 10 1 11 8 3 7 ...
 $ Education         : Factor w/ 4 levels "1","2","3","4": 3 2 2 3 2 3 3 2 2 3 ...
 $ Children          : Factor w/ 3 levels "1","2","3": 2 2 2 2 2 2 2 2 1 2 ...
 $ Minority          : Factor w/ 3 levels "1","2","3": 2 2 2 3 2 1 2 2 2 2 ...
 $ Gender            : Factor w/ 4 levels "1","2","3","7": 1 2 1 1 1 2 1 1 1 1 ...
 $ Age               : Factor w/ 7 levels "1","2","3","4",..: 6 6 6 6 5 6 6 5 4 6 ...
 $ Satisfaction_level: Factor w/ 10 levels "1","2","3","4",..: 6 6 8 7 8 7 7 10 8 5 ...
 $ Satisfaction      : chr  "Yes" "Yes" "Yes" "Yes" ...
```

We convert our response variable into a factor

```r
demographic_df <- demographic_df %>% mutate(Satisfaction = factor(Satisfaction, levels = c("No", "Yes")))

str(demographic_df$Satisfaction)
```

```
 Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 1 ...
```

4.2 Independence Discussion (Pearson's Chi Square test)

We ran a number of independence tests between our response variable, Satisfaction and the other exploratory variables to find out if there is any relationship between them.

Chi-Square test of Satisfaction and Age:
To test the dependence of Satisfaction and Age, the hypothesis is

Ho: Satisfaction and Age are independent

Ha: Satisfaction and Age are dependent.

Using a significance level of 0.05, we apply the Pearson's Chi-square test to check the dependence between the two variables.

```r
Age_Satis <- table(df$Age, df$Satisfaction)
Age_Satis
chisq.test(Age_Satis)
```

```
                       No Yes
  18-24 years           8 140
  25-34 years          14 211
  35-44 years          35 346
  45-54 years          54 362
  55-64years           72 424
  65+ years           122 662
  Prefer not to answer  14  36

        Pearson's Chi-squared test

data:  Age_Satis
X-squared = 37.553, df = 6, p-value = 1.374e-06
```

As the p-value is less than our significance level, we reject the null hypothesis and conclude that there is a statistically significant dependence between Satisfaction and Age. We ran the Pearson's Chi-Square tests for each of the other exploratory variables and Satisfaction separately:

Independence Test for Satisfaction and Years_in_yyc

```{r warning=FALSE}
Years_Satis <- table(df$Years_in_yyc, df$Satisfaction)
Years_Satis
chisq.test(Years_Satis)
```

```
     No Yes
1     6 106
2    10 136
3    21 174
4    22 206
5    29 273
6    29 173
7    27 203
8    22 160
9    44 206
10   24 141
11   84 395
12    1   8

        Pearson's Chi-squared test

data:  Years_Satis
X-squared = 31.801, df = 11, p-value = 0.0008208
```

Independence test for Satisfaction and Education

```{r}
Education_Satis <- table(df$Education, df$Satisfaction)
Education_Satis
chisq.test(Education_Satis)
```

```
                                    No  Yes
  High School or less               60  318
   Post Secondary or College Diploma 134  745
  University or PG degree           102 1058
  Don't know                         23   60

        Pearson's Chi-squared test

data:  Education_Satis
X-squared = 41.23, df = 3, p-value = 5.845e-09
```

Independence test for Satisfaction and Income

```{r}
Income_Satis <- table(df$Income, df$Satisfaction)
Income_Satis
chisq.test(Income_Satis)
```

```
             No Yes
  <30k       37 116
  30k-45k    26 153
  45k-60k    27 214
  60k-75k    22 157
  75k-90k    21 169
  90k-105    26 228
  105k-120k  14 189
  >120k      25 215
  Don't know 121 740

        Pearson's Chi-squared test

data:  Income_Satis
X-squared = 29.694, df = 8, p-value = 0.0002394
```

Independence test for Satisfaction and Children

```r
Children_Satis <- table(df$Children, df$Satisfaction)
Children_Satis
chisq.test(Children_Satis)
```

```
              No  Yes
  Yes         66  606
  No         248 1573
  Don't know   5    2
Warning: Chi-squared approximation may be incorrect
        Pearson's Chi-squared test

data:  Children_Satis
X-squared = 28.064, df = 2, p-value = 8.054e-07
```

Independence test for Satisfaction and Minority

```r
Minority_Satis <- table(df$Minority, df$Satisfaction)
Minority_Satis
chisq.test(Minority_Satis)
```

```
              No  Yes
  Yes         47  430
  No         253 1694
  Don't know  19   57

        Pearson's Chi-squared test

data:  Minority_Satis
X-squared = 13.945, df = 2, p-value = 0.0009373
```

From the Chi-square Independence tests, all p-values except for “Satisfaction and Quadrant” are small. Thus, we can conclude that there is a statistically significant dependence between Satisfaction and the variables: Age, Income, Years lived in Calgary, Minority, Education, Children

Now let’s move on to applying different models to our ‘demographic’ dataset.

4.3 Logistic Regression

Checking for Multicollinearity

Before applying statistical methods to train our models, we would typically check for multicollinearity. However, since all our predictors and the response variable are categorical, this step is omitted. Unlike continuous variables, categorical predictors do not exhibit traditional multicollinearity, making this check unnecessary for our dataset.

Train test split

Before conducting our analysis, we split the dataset into training and test sets. Given the imbalance in class proportions for the response variable, we applied stratified sampling to ensure that both the training and test datasets maintain similar distributions of “No” and “Yes” in our response variable, Satisfaction. This approach helps preserve representativeness and prevents biases in model evaluation.

We check the proportion of classes in our dataset

```{r}
table(demographic_df$Satisfaction)
```

```
  No  Yes
 319 2180
```

Let us split the dataset - 70% for training and 30% for testing

```{r warning = FALSE}
library(caret)
set.seed(2024)

# Create stratified split (70% train, 30% test)
train_index <- createDataPartition(demographic_df$Satisfaction, p = 0.7, list = FALSE)

# Split data
train_data <- demographic_df[train_index, ]
test_data <- demographic_df[-train_index, ]

```

Let us check to ensure that the proportion of classes in our train and test are approximately the same.

```{r}
# To ensure we have the same proportion of classes of our response variable in test and train

prop.table(table(train_data$Satisfaction))
prop.table(table(test_data$Satisfaction))

```

```
   No   Yes
0.128 0.872

       No       Yes
0.1268358 0.8731642
```

Let us check the dimensions of the train and test

```{r}
# Check the dimensions of train and test
dim(train_data)
dim(test_data)
```

```
[1] 1750   11
[1] 749  11
```

4.3.1 Full Logit Model

Let's create and train the logistic model over the training data and check the summary. We use all our exploratory variables to build this full logit model. We remove the column Satisfaction_level from the model to avoid bias.

```{r}
model1 <- glm(Satisfaction ~ .-Satisfaction_level, data = train_data, family = binomial)
summary(model1)
```

Summary of our model

```
Call:
glm(formula = Satisfaction ~ . - Satisfaction_level, family = binomial,
    data = train_data)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    2.52050    0.76186   3.308 0.000939 ***
Quadrant2      0.06788    0.21589   0.314 0.753195
Quadrant3      0.14933    0.21104   0.708 0.479216
Quadrant4     -0.14617    0.22092  -0.662 0.508207
Income2        0.52717    0.33315   1.582 0.113556
Income3        0.86924    0.33966   2.559 0.010494 *
Income4        0.43191    0.35031   1.233 0.217603
Income5        1.24178    0.41903   2.963 0.003042 **
Income6        0.60286    0.33635   1.792 0.073073 .
Income7        1.44069    0.43677   3.299 0.000972 ***
Income8        0.47289    0.34262   1.380 0.167522
Income9        0.52285    0.27135   1.927 0.053995 .
Tenancy2      -0.38002    0.21812  -1.742 0.081465 .
Tenancy3      -0.47813    1.13937  -0.420 0.674746
Tenancy4       0.21037    1.07345   0.196 0.844631
Tenancy5      -2.89334    1.26101  -2.294 0.021764 *
Years_in_yyc2 -0.05367    0.60171  -0.089 0.928930
Years_in_yyc3 -0.52990    0.53429  -0.992 0.321309
Years_in_yyc4 -0.01534    0.55909  -0.027 0.978107
Years_in_yyc5 -0.16108    0.51938  -0.310 0.756457
Years_in_yyc6 -0.90641    0.51686  -1.754 0.079487 .
Years_in_yyc7 -0.19583    0.54069  -0.362 0.717217
Years_in_yyc8 -0.44497    0.54254  -0.820 0.412131
Years_in_yyc9 -0.78118    0.50613  -1.543 0.122719
Years_in_yyc10 -0.32748   0.54930  -0.596 0.551049
Years_in_yyc11 -0.52142   0.50024  -1.042 0.297258
Years_in_yyc12  2.61160   2.09669   1.246 0.212917
Education2     0.05059    0.22327   0.227 0.820751
Education3     0.59574    0.24071   2.475 0.013327 *
Education4    -0.50862    0.36145  -1.407 0.159378
Children2     -0.15843    0.23346  -0.679 0.497395
Children3     -2.09788    1.33503  -1.571 0.116086
Minority2     -0.22286    0.23062  -0.966 0.333859
Minority3     -0.62670    0.41262  -1.519 0.128805
Gender2        0.33290    0.15561   2.139 0.032406 *
Gender3       -0.79694    1.23378  -0.646 0.518326
Gender7        0.52050    1.08908   0.478 0.632701
Age2          -0.12872    0.57298  -0.225 0.822253
Age3          -0.58061    0.51908  -1.119 0.263340
Age4          -0.86301    0.50341  -1.714 0.086470 .
Age5          -0.85944    0.50929  -1.688 0.091503 .
Age6          -0.87254    0.51061  -1.709 0.087482 .
Age7          -1.61144    0.67929  -2.372 0.017681 *
---
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1339.0  on 1749  degrees of freedom
Residual deviance: 1216.5  on 1707  degrees of freedom
AIC: 1302.5

Number of Fisher Scoring iterations: 5
```

In our analysis, we initially aimed to use a significance level of 0.05 for coefficient evaluation. However, given very few variables met this threshold, we adjusted the significance level to 0.1. This approach allows us to explore potential relationships that may not be strongly significant but could still provide meaningful insights.

Based on the significance level of 0.1, the following variables are insignificant:

- Quadrant
- Children
- Minority

To be consistent, we will consider these variables to be insignificant for other models as well.

Let’s test this model on the test dataset and evaluate its classification performance.

```{r}
# making predictions on the test set
pred1 <- predict(model1, test_data, type = "response" )
```

```{r}
# setting the probability threshold.
predicted_class <- ifelse(pred1 > 0.5, "Yes", "No")
head(predicted_class)
```

```
    4    13    17    18    20    25
"Yes" "Yes" "Yes" "Yes" "Yes" "Yes"
```

```{r}
# Accuracy and classification performance
Actuals <- test_data$Satisfaction
table(predicted_class, Actuals)

Accuracy <- mean(predicted_class == Actuals)
Accuracy
```

```
               Actuals
predicted_class  No Yes
            No    1   4
            Yes  94 650
[1] 0.8691589
```

Although our model achieved a high accuracy of 86.9% we observe that it is unable to capture the "No" class as seen in the confusion matrix. This issue is likely due to class imbalance in the dataset, where "Yes" class dominates leading to fewer correct predictions of "No"

4.3.2 Reduced Logit Model

Let's fit our model using only the significant variables and take a look at the summary.

```r
model_sgx <- glm(Satisfaction ~ . - Satisfaction_level -Quadrant -Children -Minority , data =
train_data, family = binomial)

summary(model_sgx)
```

Summary of the reduced logit model:

```
Call:
glm(formula = Satisfaction ~ . - Satisfaction_level - Quadrant -
    Children - Minority, family = binomial, data = train_data)

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)     2.210960   0.707339   3.126 0.001774 **
Income2         0.572869   0.331026   1.731 0.083526 .
Income3         0.886185   0.337670   2.624 0.008680 **
Income4         0.451944   0.347922   1.299 0.193950
Income5         1.309238   0.416953   3.140 0.001689 **
Income6         0.630035   0.334710   1.882 0.059791 .
Income7         1.495527   0.435017   3.438 0.000586 ***
Income8         0.532533   0.340661   1.563 0.117997
Income9         0.577337   0.268637   2.149 0.031624 *
Tenancy2       -0.383710   0.217197  -1.767 0.077287 .
Tenancy3       -0.504114   1.131512  -0.446 0.655942
Tenancy4        0.178856   1.072986   0.167 0.867614
Tenancy5       -3.324825   1.274229  -2.609 0.009073 **
Years_in_yyc2   0.005245   0.600047   0.009 0.993026
Years_in_yyc3  -0.537442   0.530750  -1.013 0.311248
Years_in_yyc4   0.006891   0.556560   0.012 0.990121
Years_in_yyc5  -0.185872   0.513784  -0.362 0.717524
Years_in_yyc6  -0.931310   0.511855  -1.819 0.068838 .
Years_in_yyc7  -0.178599   0.536558  -0.333 0.739240
Years_in_yyc8  -0.487477   0.536465  -0.909 0.363517
Years_in_yyc9  -0.804087   0.501396  -1.604 0.108781
Years_in_yyc10 -0.324223   0.544210  -0.596 0.551330
Years_in_yyc11 -0.545293   0.493745  -1.104 0.269419
Years_in_yyc12  1.753934   1.644680   1.066 0.286230
Education2      0.089710   0.220473   0.407 0.684082
Education3      0.651308   0.233405   2.790 0.005263 **
Education4     -0.495047   0.357289  -1.386 0.165880
Gender2         0.334213   0.154764   2.159 0.030811 *
Gender3        -0.497230   1.262028  -0.394 0.693586
Gender7         0.503624   1.047517   0.481 0.630674
Age2           -0.148246   0.569139  -0.260 0.794498
Age3           -0.560053   0.509258  -1.100 0.271445
Age4           -0.872502   0.496078  -1.759 0.078612 .
Age5           -0.961762   0.501422  -1.918 0.055102 .
Age6           -0.967896   0.501129  -1.931 0.053430 .
Age7           -2.012373   0.639158  -3.148 0.001641 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1339.0  on 1749  degrees of freedom
Residual deviance: 1224.1  on 1714  degrees of freedom
AIC: 1296.1

Number of Fisher Scoring iterations: 5
```

Let's evaluate the performance of our reduced logit model on test data

```r
# predicting from the model with only significant variables
pred_sgx <- predict(model_sgx, test_data, type = "response" )
predicted_class_sgx <- ifelse(pred_sgx > 0.5, "Yes", "No")
Actual <- test_data$Satisfaction
table(predicted_class_sgx, Actual)

Accuracy_sgx <- mean(predicted_class_sgx == test_data$Satisfaction)
Accuracy_sgx
```

```
                   Actual
predicted_class_sgx  No Yes
                 No   0   6
                 Yes 95 648
[1] 0.8651535
```

Our Reduced Logit model too has a high accuracy of 86.51% but is struggling to capture the "No" class. This clearly indicates the effect of class imbalance inherent in the dataset. We want to improve upon these models by adjusting class weights and optimizing probability thresholds to capture more "No"s. So let's build our next model with these improvements.

4.3.3 Logit model with increased weight for" No" and increased prob. threshold

To mitigate the class imbalance issue observed in our previous models, we implemented two key adjustments:

(i) Increased weight for the "No" class, ensuring the model gives more importance to the minority category.

(ii) Raised the probability threshold to 0.75, making the model more selective in classifying "Yes" cases and improving identification of "No" instances.

Let us fit our model on the train set:

```r
model2 <- glm(Satisfaction ~ . - Satisfaction_level-Minority -Children - Quadrant, data =
train_data, family = binomial, weights = ifelse(train_data$Satisfaction == "No", 2, 1))

```

Let’s make predictions on the test set and get the accuracy and classification performance of this model.

```{r}
pred2 <- predict(model2, test_data, type = "response" )
predicted_incNO <- ifelse(pred2 > 0.75, "Yes", "No")

Actuals <- test_data$Satisfaction
table(predicted_incNO, Actuals)

Accuracy2 <- mean(predicted_incNO == Actuals)
Accuracy2
```

```
               Actuals
predicted_incNO  No Yes
            No   40 207
            Yes  55 447
[1] 0.6502003
```

Our model, which incorporates increased weighting for the "No" class and a higher probability threshold, successfully improves classification for the minority class. However, this comes at the expense of overall accuracy and reduces the model’s ability to correctly identify "Yes" cases. This trade-off highlights the challenge of balancing precision and recall in imbalanced datasets.

4.3.4 Logit model with up-sampling

We next try up-sampling the train data as we feel that the reason why our models are struggling to identify the “No” class is that there simply are not enough “No” classes that our algorithms can get trained on. After we split our data into train and test, we up-sample the train set - to include 2500 samples from “Yes” class and 2000 samples from “No” class. We then train our logit regression model on this up-sampled data and test our model on the test data (which has the class proportions of the original dataset).

Up sampling the train data

```{r}
set.seed(2024)
library(sampling)
library(survey)
idx <- sampling:::strata(train_data, stratanames = c("Satisfaction"), size = c(2500,2000), method = "srswr")
```

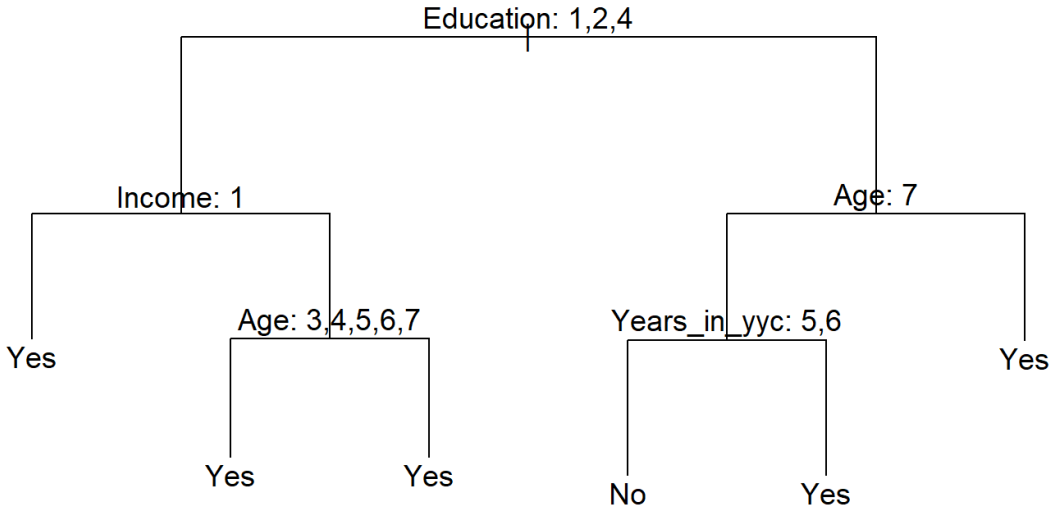
Let us check the dimensions of original train set and up sampled train set

```{r}
train_data_upsampled <- train_data[idx$ID_unit, ]
testing_data <- test_data

dim(train_data)     # original train set
dim(train_data_upsampled)  # upsampled train set.
```

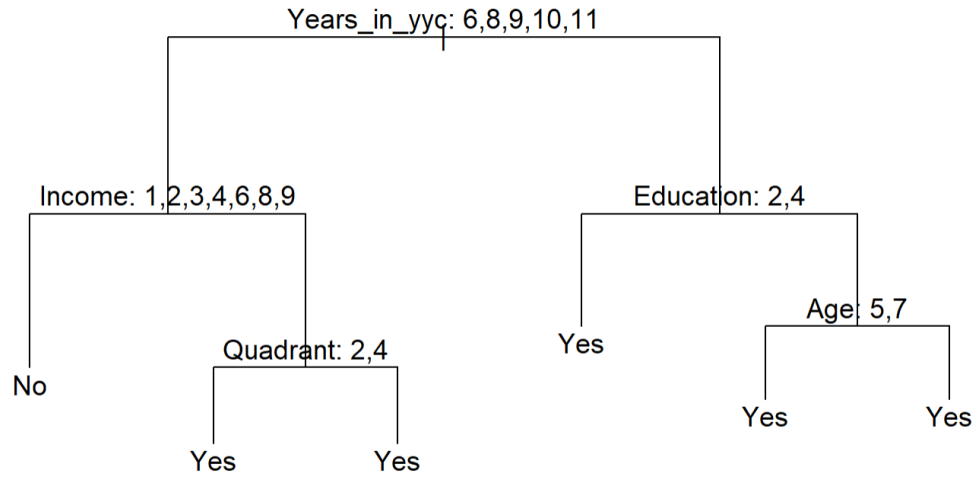
```
[1] 1750   11
[1] 4500   11
```

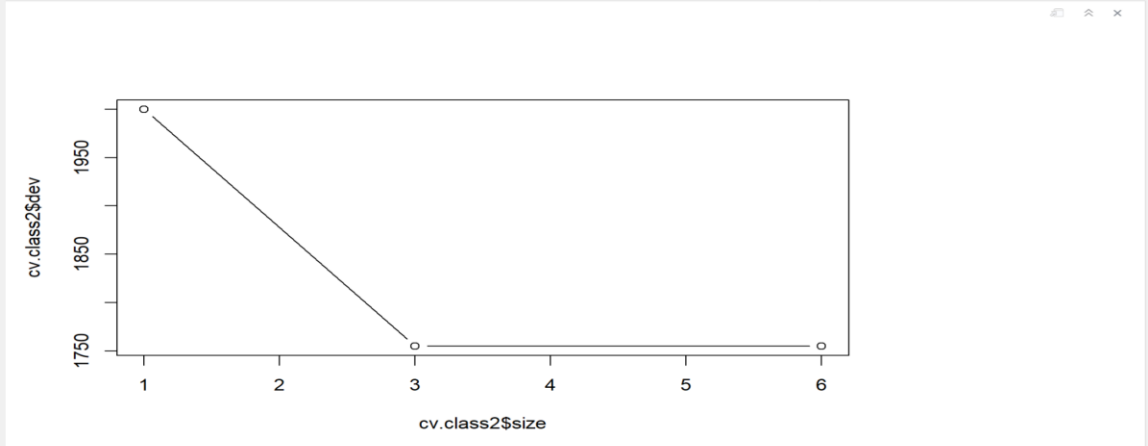
Let us fit our model on the up-sampled train data.

```
# Fitting the model on upsampled data
```{r}
model3 <- glm(Satisfaction ~ . - Satisfaction_level-Quadrant -Children -Minority , data = train_data_upsampled,
family = binomial)

```
```

Let's make predictions on our test set and evaluate its accuracy and classification performance

```
# Making predictions on test data that has the class proportions of original dataset.
```{r}
pred3 <- predict(model3, testing_data, type = "response" )
predicted_upsamp <- ifelse(pred3 > 0.5, "Yes", "No")
Actual <- testing_data$Satisfaction
table(predicted_upsamp, Actual)

Accuracy3 <- mean(predicted_upsamp == testing_data$Satisfaction)
Accuracy3
```
```

```
                Actual
predicted_upsamp  No Yes
             No   33 166
             Yes  62 488
[1] 0.6955941
```

We observe that the accuracy and the overall classification performance (including both positive and negative class) have slightly improved compared to our model with increased weightage for negative class and increased probability threshold. The accuracy is now 69.56%, which is the best so far.

To ensure that our model's improved performance is not simply due to a 'lucky' split of the training and test data, we employ K-fold cross-validation. This technique provides a more robust estimate of the misclassification error, allowing us to validate the model's reliability and efficiency across different data partitions. By evaluating its performance on multiple subsets, we confirm that our model generalizes well and is not overly dependent on a specific train-test split.

4.3.5 K- fold cross validation of our model with up sampled train data.

We implemented 10-fold cross-validation to ensure a balanced evaluation of our model. The dataset was first split into K folds, maintaining approximately equal class proportions across all folds. For each iteration, one-fold was set aside for testing, while the remaining K-1 folds were used for training. To address class imbalance, the training set was up sampled to include 2,500 samples from the 'Yes' class and 2,000 samples from the 'No' class, ensuring sufficient representation of both categories. The trained model was then tested on the fold set aside as the test set. This process was repeated across all K folds, and the error rate was averaged to obtain a robust estimate of the model's accuracy. By following this approach, we ensured that our performance metrics were reliable and reflective of the model's ability to generalize to new data.

Let us first create 10 folds

```
## K-fold cross validation
```{r}
set.seed(2024)
folds<-createFolds(demographic_df$Satisfaction, k=10)
```
```

Let us fit our model on our 10 folds and then predict and finally compute the CV error for misclassification

```
```{r}
library(MASS)
log_misclassification <- c()
for (i in 1:10) {
  trainIndex <- unlist(folds[-i])
  testIndex <- unlist(folds[i])

  trainData <- demographic_df[trainIndex, ]
  testData <- demographic_df[testIndex, ]

  idx2 <- sampling:::strata(trainData, stratanames = c("Satisfaction"), size = c(2500,2000), method = "srswr")
  trainData_upsampled <- trainData[idx2$ID_unit, ]

  log_cv_upsampled <- glm(Satisfaction ~ . - Satisfaction_level-Quadrant -Children -Minority , data =
trainData_upsampled, family = binomial)

  pred_log_cv <- predict(log_cv_upsampled, testData, type = "response" )
  predicted_satis <- ifelse(pred_log_cv > 0.5, "Yes", "No")

  log_misclassification[i] <- mean(predicted_satis != testData$Satisfaction)
}

log_cv_error <- mean(log_misclassification)
cat("The cross validation error for log model is: ", log_cv_error)
log_misclassification
```
```

```
The cross validation error for log model is:  0.3221173 [1] 0.4760000 0.3000000 0.2931727 0.2760000 0.2920000
0.3200000 0.3000000 0.3440000 0.3200000 0.3000000
```

This gives us an average cv misclassification error for this model as 0.322 or 32.2% We may deduce that average accuracy of model across K folds is ~ (1-0.322) = ~ 0.675 or 67.5%

Comparing our models:

Model	Test Accuracy	Classification performance	Observations/Remarks
Full Logit Model	86.92%	Actuals predicted_class No Yes No 1 4 Yes 94 650	High accuracy as a result of Class imbalance. Poor classification performance. Unable to capture minority class.
Reduced Logit Model *used only significant variables* *Removed Quadrant, Minority, Children*	86.56%	Actual predicted_class_sgx No Yes No 0 5 Yes 95 646	High accuracy as a result of Class imbalance. Poor classification performance. Unable to capture minority class
Logit Model with *-increased weight for Minor Class* *-increased prob threshold to 0.75*	65.02%	Actuals predicted_incNO No Yes No 40 207 Yes 55 447	Reduced accuracy but improved classification performance.

Model	Test Accuracy	Classification performance	Remarks/ Observations
Logit model with Upsampling	69.55%	Actual predicted_upsamp No Yes No 33 166 Yes 62 488	Improved Accuracy and classification performance. Best model so far.
Logit model with Upsampling and K-fold cross validation *Cross validation ensures that we have reliable metrics to evaluate our model.*	67.08% cv_misclassification error = 32.3 %	***worst performing fold:*** predicted_satis No Yes No 22 119 Yes 10 99 [1] 0.484 ***best performing fold:*** predicted_satis No Yes No 17 53 Yes 15 165 [1] 0.728	Our model has average accuracy of **67.08 %** And misclassification error of **32.3 %** ***Our model is not at peak performance but is a good improvement from our initial model in classification performance.***

4.4 Tree Models

In this section we see how classification trees perform on our data. We want to apply decision trees to our data. First, we split the data into two parts - train and test using stratified sampling in order to maintain the same proportions of the classes train and test. We will then apply a classification tree algorithm to the training set to establish a relationship between "Satisfaction" and our exploratory features like income, gender, age, years lived in Calgary, tenancy etc.

4.4.1 Tree model using train data

Let us use train data and fit the classification tree model and plot the full tree

```
## TREE MODEL
```{r}
# Tree model
library(tree)

# fit the model on train data
class_tree_model <- tree(Satisfaction ~.-Satisfaction_level, data = train_data)


# plot the tree
summary(class_tree_model)
plot(class_tree_model)
text(class_tree_model, pretty = 0)
```
```



We next prune the tree by restricting our classification tree to the best number of terminal nodes obtained from cross validation selection. We then evaluate our model.

```
{r}
# Predict Satisfaction on test set
tree_pred <- predict(class_tree_model, test_data, type = "class")

table(tree_pred, test_data$Satisfaction)

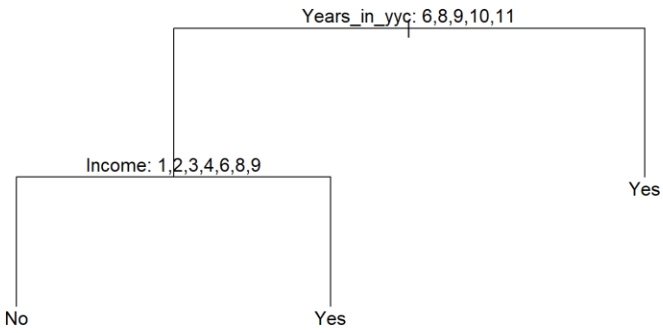
# Compute accuracy
conf_matrix <- confusionMatrix(tree_pred, test_data$Satisfaction)

# Print accuracy
print(conf_matrix$overall["Accuracy"])
```

```
tree_pred  No Yes
      No    0   2
      Yes  95 652
Accuracy
0.870494
```

We observe that our tree classification model too is suffering from the class imbalance problem inherent in our dataset. While our accuracy is pretty high at 87.0494 %, our model is failing to identify any samples belonging to the minority class (“No” class). Therefore we use the upsampled train set so that the algorithm can get trained on enough number of “No” class samples to identify them.

4.4.2 Tree model using up sampled train data

We followed the same method we utilized earlier while building logistic models to up-sample our train data. We fit our classification tree on our up-sampled train data and plot our full tree.

```
```{r}
# fit the model on upsampled train data
tree_model2 <- tree(Satisfaction ~.-Satisfaction_level, data = train_data_upsampled)


# plot the tree
summary(tree_model2)
plot(tree_model2)
text(tree_model2, pretty = 0)
```
```

Let's look at our full tree



We prune our tree to restrict it to the best number of terminal nodes obtained through cross validation selection.

```
```{r}
cv.class2 = cv.tree(tree_model2, FUN=prune.misclass)
plot(cv.class2$size,cv.class2$dev, type="b")
```
```



We decided to prune our classification tree to 3 terminal nodes and we plot the pruned tree

```r
prune.satisfaction = prune.tree(tree_model2, best = 3)
plot(prune.satisfaction)
text(prune.satisfaction, pretty = 0)
```



Let us evaluate the performance of our tree model trained on up-sampled data by testing it on the test set.

```r
satisfaction_pruned = predict(prune.satisfaction, test_data, type = "class")

table(satisfaction_pruned, test_data$Satisfaction)
accuracy <- mean(satisfaction_pruned == test_data$Satisfaction)
accuracy
```

```
satisfaction_pruned  No Yes
                 No  46 281
                 Yes 49 373
[1] 0.5594126
```

We observe that our model is now able to capture the minority class much better. Although this improved ability to capture the minority class has come at the cost of decreased accuracy and a decrease in the ability to correctly identify the positive class.

Model	Test Accuracy	Classification performance	Remarks/Observations
Classification tree	87.04%	tree_pred No Yes No 0 2 Yes 95 652	Sufferers from Same class imbalance problem as seen in logit model
Classification tree with upsampling	55.94%	tree_upsamp No Yes No 46 281 Yes 49 373	Better at identifying "No" class but decreased overall accuracy.

4.4.3 Conclusions:

Our objective for this part of the analysis was to build a model to estimate whether an individual is satisfied with quality of life in Calgary when given their demographic features, including income, gender, age, years lived in the city, and tenancy etc. Our findings highlight key strengths and limitations in our selected models' predictive performance.

Among all tested approaches, **logistic regression with upsampling** emerged as the best-performing model, achieving a **K-fold accuracy of 67.08%**. It proved particularly effective in identifying individuals who reported satisfaction with their quality of life.

For the "Yes" category, the model demonstrated high **recall 88.7%**, meaning it successfully captured most satisfied individuals. Additionally, **precision of 74.6%** ensured that a strong majority of those predicted as satisfied were correctly classified, leading to an **F1 score of 81.1%**.

For the "No" category, the model struggled more, with **precision at 34.7%** and **recall at 16.6%**, resulting in an F1 score of 22.5%. This indicates challenges in accurately identifying individuals who reported dissatisfaction, suggesting potential refinements are needed.

While the model significantly improves classification performance compared to previous iterations, there is room to enhance its ability to distinguish between satisfied and dissatisfied individuals more effectively. Future work could focus on addressing imbalances in classification performance, particularly for those reporting lower satisfaction, to provide a more comprehensive understanding of quality-of-life factors in Calgary.

4.4.4 Future Work

To improve our models, we aim to address uncertainty introduced when respondents select "Prefer not to answer" for income, gender, and education qualifications. This added uncertainty into the system. We believe we can build better models when we handle this uncertainty. Developing strategies like imputation techniques, separate classification handling, or refined survey designs to

minimize ambiguity will be part of future work. Additionally, improving predictions for minority classes by exploring strategies such as SMOTE, cost-sensitive learning, and ensemble models will strengthen balance in the data. Finally, optimizing feature selection and fine-tuning model parameters will enhance both accuracy and interpretability, leading to more reliable outcomes.

5. Analysis - 2

In this part of the analysis, we identify which of the services provided by the city of Calgary impacts the resident's satisfaction ratings. The following factors are considered as exploratory variables in our analysis for this part of the study:

1. Calgary Fire Department
2. 911
3. Protection from river flooding
4. Disaster planning and response
5. Residential garbage collection service
6. Residential Blue Cart recycling
7. Residential Green Cart service
8. The quality of drinking water
9. Transportation planning
10. Calgary Transit including bus and CTrain service
11. City operated roads and infrastructure
12. Road maintenance including pothole repairs
13. Spring road cleaning
14. Snow removal
15. Traffic flow management
16. Bylaw services for things such as noise complaints, fire pits and weeds
17. Animal control services for stray animals and pet licensing
18. Calgary's parks, playgrounds and other open spaces
19. Community services such as support for community associations and not for profit groups
20. Social services for individuals such as seniors or youth
21. Affordable housing for low
22. City operated recreation PROGRAMS such as swimming lessons
23. City operated recreation FACILITIES such as pools, leisure centers, and golf courses
24. Support for arts and culture including festivals
25. On-street bikeways
26. Calgary's pathway system

27. City land use planning
28. Development and building inspections and permits
29. Business licenses and inspections
30. City growth management
31. Downtown revitalization
32. Property tax assessment
33. City of Calgary website
34. 311 service

The response variable in this study is Satisfaction_level which is a rating given by the survey respondents for quality of life in Calgary, measured on a scale of 1 to 10, where

- 1 = Very Poor Satisfaction
- 10 = Well Satisfied

To simplify analysis, **Satisfaction** is converted into a binary variable and takes the values:

- "Yes" – If Rating > 5 (indicating overall satisfaction)
- "No" – If Rating ≤ 5 (indicating dissatisfaction)

Checking for Multicollinearity

Before applying statistical methods to train our models, we would typically check for multicollinearity. However, since all our predictors and the response variable are categorical, this step is omitted. Unlike continuous variables, categorical predictors do not exhibit traditional multicollinearity, making this check unnecessary for our dataset.

Train test split

Before conducting our analysis, we split the dataset into training and test sets. Given the imbalance in class proportions for the response variable, we applied stratified sampling to ensure that both the training and test datasets maintain similar distributions of "No" and "Yes" in Satisfaction. This approach helps preserve representativeness and prevents biases in model evaluation.

5.1 Logistic Regression

5.1.1 Full Logit Model

Let's create and train the logistic model over the training data and check the summary

```
```{r}
logit_model <- glm(satisfaction ~ ., data = train_balanced, family = binomial)

summary(logit_model)

```
```

Summary of our model

```
Call:
glm(formula = satisfaction ~ ., family = binomial, data = train_balanced)

Coefficients:
                                                                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                                                                     -26.94298    2.40413 -11.207  < 2e-16 ***
`Calgary Fire Department`                                                        -0.31519    0.29381  -1.073 0.283378
`911`                                                                             0.86110    0.17281   4.983 6.26e-07 ***
`Protection from river flooding`                                                  0.86299    0.19862   4.345 1.39e-05 ***
`Disaster planning and response`                                                 -0.87963    0.22079  -3.984 6.78e-05 ***
`Residential garbage collection service`                                          0.64079    0.17798   3.600 0.000318 ***
`Residential Blue Cart recycling`                                                 0.78940    0.18933   4.169 3.05e-05 ***
`Residential Green Cart service`                                                  1.75099    0.23195   7.549 4.38e-14 ***
`The quality of drinking water`                                                   0.14847    0.18007   0.825 0.409651
`Transportation planning`                                                        -1.10117    0.17186  -6.407 1.48e-10 ***
`Calgary Transit including bus and CTrain service`                               -0.34057    0.15021  -2.267 0.023368 *
`City operated roads and infrastructure`                                          0.96749    0.22433   4.313 1.61e-05 ***
`Road maintenance including pothole repairs`                                     -0.29923    0.18786  -1.593 0.111187
`Spring road cleaning`                                                           -0.09701    0.21030  -0.461 0.644605
`Snow removal`                                                                    1.10010    0.14252   7.719 1.17e-14 ***
`Traffic flow management`                                                        -0.13829    0.18642  -0.742 0.458189
`Bylaw services for things such as noise complaints, fire pits and weeds`         0.65287    0.19658   3.321 0.000896 ***
`Animal control services for stray animals and pet licensing`                    -1.20367    0.19005  -6.333 2.40e-10 ***
`Calgary's parks, playgrounds and other open spaces`                              1.43814    0.14928   9.634  < 2e-16 ***
`Community services such as support for community associations and not for profit groups`  1.05893    0.17315   6.116 9.62e-10 ***
`Social services for individuals such as seniors or youth`                       -0.27896    0.13501  -2.066 0.038812 *
`Affordable housing for low`                                                      0.23114    0.12514   1.847 0.064747 .
`City operated recreation PROGRAMS such as swimming lessons`                     -0.21848    0.13953  -1.566 0.117386
`City operated recreation FACILITIES such as pools, leisure centres, and golf courses`  0.43224    0.18353   2.355 0.018515 *
`Support for arts and culture including festivals`                                0.14851    0.14767   1.006 0.314546
`On-street bikeways`                                                             -0.02277    0.11886  -0.192 0.848089
`Calgary's pathway system`                                                        0.17848    0.14876   1.200 0.230204
`City land use planning`                                                         -0.31715    0.15546  -2.040 0.041343 *
`Development and building inspections and permits`                                0.25030    0.15987   1.566 0.117440
`Business licenses and inspections`                                              0.14547    0.12125   1.200 0.230238
`City growth management`                                                         0.80436    0.13610   5.910 3.42e-09 ***
`Downtown revitalization`                                                        -0.26179    0.12757  -2.052 0.040157 *
`Property tax assessment`                                                         0.21567    0.12782   1.687 0.091555 .
`City of Calgary website`                                                         0.63582    0.12003   5.297 1.18e-07 ***
`311 service`                                                                    -0.95363    0.14491  -6.581 4.68e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2098.8  on 1513  degrees of freedom
Residual deviance: 1289.4  on 1479  degrees of freedom
AIC: 1359.4

Number of Fisher Scoring iterations: 6
```

Performance:

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 54 26
         1 46 74
```

- Accuracy : 0.64
- 95% CI : (0.5693, 0.7065)
- No Information Rate : 0.5
- P-Value [Acc > NIR] : 4.565e-05
- Sensitivity : 0.5400
- Specificity : 0.7400
- Pos Pred Value : 0.6750

- Neg Pred Value : 0.6167
- Prevalence : 0.5000
- Detection Rate : 0.2700
- Detection Prevalence : 0.4000
- Balanced Accuracy : 0.6400

Based on the significance level of 0.05, the following variables are insignificant:

- Residential garbage collection service
- Residential Green Cart service
- City operated roads and infrastructure
- Road maintenance including pothole repairs
- Snow removal
- Affordable housing for low
- Development and building inspections and permits

Now, A reduced logit model is built using only the significant variables. Let's see how that model is performing.

5.1.2 Reduced Logit Model

```
formula_significant <- as.formula(paste("satisfaction ~", paste(significant_vars, collapse = " + ")))
model_sig_2 <- glm(formula_significant, data = train_balanced, family = binomial)
```

Summary:

```
Call:
glm(formula = formula_significant, family = binomial, data = train_balanced)

Coefficients:
                                                                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                                                                      -26.4583     1.7101 -15.471  < 2e-16 ***
`911`                                                                              0.6655     0.1452   4.582 4.60e-06 ***
`Protection from river flooding`                                                   0.7406     0.1830   4.046 5.21e-05 ***
`Disaster planning and response`                                                  -0.7620     0.1942  -3.924 8.70e-05 ***
`Residential garbage collection service`                                           0.4428     0.1618   2.736 0.006214 **
`Residential Blue Cart recycling`                                                  0.7428     0.1738   4.274 1.92e-05 ***
`Residential Green Cart service`                                                   1.5897     0.2042   7.785 6.97e-15 ***
`Transportation planning`                                                         -1.2670     0.1448  -8.753  < 2e-16 ***
`City operated roads and infrastructure`                                           0.6951     0.1721   4.039 5.37e-05 ***
`Snow removal`                                                                     0.9444     0.1307   7.226 4.99e-13 ***
`Bylaw services for things such as noise complaints, fire pits and weeds`          0.5996     0.1708   3.511 0.000446 ***
`Animal control services for stray animals and pet licensing`                     -1.0808     0.1714  -6.304 2.91e-10 ***
`Calgary's parks, playgrounds and other open spaces`                               1.5418     0.1377  11.193  < 2e-16 ***
`Community services such as support for community associations and not for profit groups`  0.9993     0.1253   7.975 1.52e-15 ***
`City operated recreation FACILITIES such as pools, leisure centres, and golf courses`    0.5335     0.1430   3.731 0.000190 ***
`City land use planning`                                                          -0.3099     0.1197  -2.589 0.009618 **
`City growth management`                                                           0.7927     0.1180   6.719 1.83e-11 ***
`City of Calgary website`                                                          0.6265     0.1021   6.136 8.44e-10 ***
`311 service`                                                                     -0.7732     0.1211  -6.384 1.73e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 2098.8  on 1513  degrees of freedom
Residual deviance: 1328.0  on 1495  degrees of freedom
AIC: 1366

Number of Fisher Scoring iterations: 6
```

Performance:

```
Confusion Matrix and Statistics

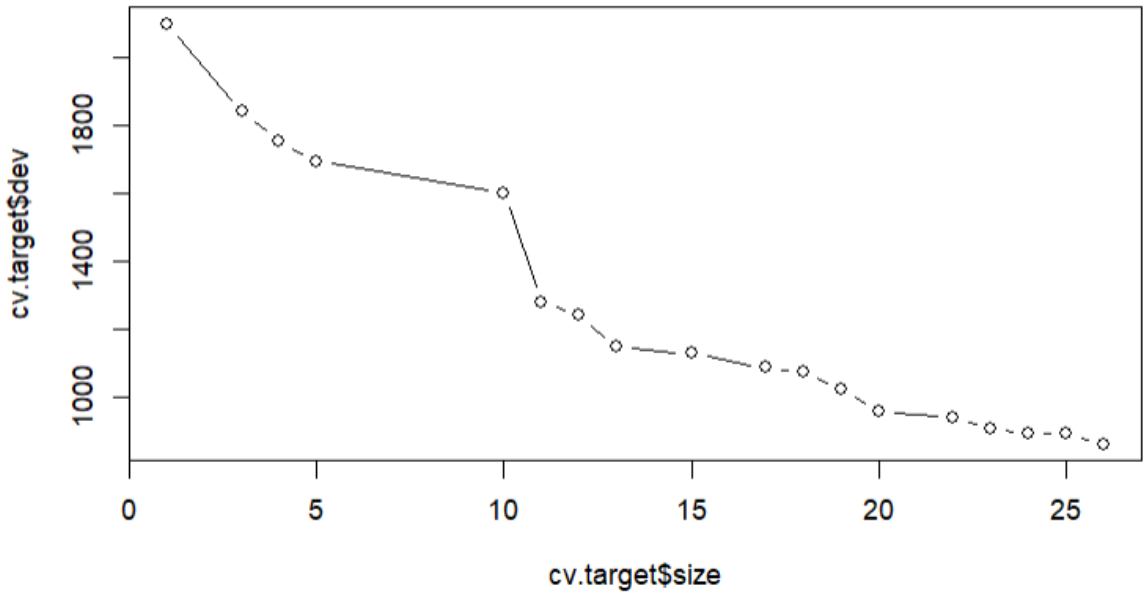
          Reference
Prediction  0  1
         0 52 26
         1 48 74
```

- Accuracy : 0.63
- 95% CI : (0.5591, 0.697)
- No Information Rate : 0.5
- P-Value [Acc > NIR] : 0.0001446
- Kappa : 0.26
- Mcnemar's Test P-Value : 0.0146385
- Sensitivity : 0.5200
- Specificity : 0.7400
- Pos Pred Value : 0.6667
- Neg Pred Value : 0.6066
- Prevalence : 0.5000
- Detection Rate : 0.2600
- Detection Prevalence : 0.3900
- Balanced Accuracy : 0.6300

We have used this model to get the most influential services among all the services discussed above earlier. We have used the **varImp()** function to get the importance of these variables in our model. It's a univariate measure (each variable evaluated independently).The function computes the absolute value of the t-statistic (or z-value) for each predictor. The higher the absolute value, the more "important" the variable is considered to be. It does not consider interaction effects or multicollinearity — it's based on the individual contribution of each predictor. To answer the question “Can we really depend on this measure?”, the reasons such as

- They have statistically significant p-values (< 0.05).
- They show high variable importance (varImp()).
- They have acceptable VIFs.
- They make logical sense.

For example, more satisfaction with **parks** and **snow removal** would logically connect to higher **quality of life ratings**. Poor **transportation planning** or **garbage collection** would likely cause dissatisfaction.

5.2 Classification Tree

Classification trees are a type of decision tree used for predicting categorical outcomes. They work by recursively splitting the data into subsets based on the values of predictor variables, creating a tree-like structure where each branch represents a decision rule, and each leaf represents a class label. The goal is to create simple, interpretable rules that classify observations as accurately as possible. Classification trees are especially useful because they can handle both numerical and categorical predictors, capture complex interactions between variables, and provide visual insight into how decisions are made. However, they can overfit the data if not pruned or tuned properly.

```
```{r}
tree.satisfaction<-tree(satisfaction ~ ., train_data)
summary(tree.satisfaction)

```
```

Summary:

```
Classification tree:
tree(formula = satisfaction ~ ., data = train_data)
Variables actually used in tree construction:
 [1] "Residential.Green.Cart.service"
 [2] "Road.maintenance.including.pothole.repairs"
 [3] "X911"
 [4] "City.of.Calgary.website"
 [5] "Calgary.s.pathway.system"
 [6] "Community.services.such.as.support.for.community.associations.and.not.for.profit.groups"
 [7] "On.street.bikeways"
 [8] "Calgary.s.parks..playgrounds.and.other.open.spaces"
 [9] "Property.tax.assessment"
[10] "City.operated.recreation.PROGRAMS.such.as.swimming.lessons"
[11] "Affordable.housing.for.low"
[12] "Disaster.planning.and.response"
[13] "Spring.road.cleaning"
[14] "Calgary.Fire.Department"
[15] "Residential.Blue.Cart.recycling"
[16] "Animal.control.services.for.stray.animals.and.pet.licensing"
[17] "Calgary.Transit.including.bus.and.CTrain.service"
[18] "City.operated.roads.and.infrastructure"
[19] "Snow.removal"
[20] "Downtown.revitalization"
Number of terminal nodes:  26
Residual mean deviance:  0.4749 = 706.6 / 1488
Misclassification error rate: 0.09313 = 141 / 1514
```

Since the model has 20 variables which would mean the tree would be having 40 branches, so we decided to not plot the tree as it would be incomprehensible. We have done a cross validation pruning and the graph of the CV error and the number of primary nodes is given below.



The graph indicates that we can achieve the lowest error with having all the leaf nodes. So we decided not to prune the tree anymore.
Performance:

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 32 19
         1 68 81
```

- Accuracy : 0.565
- 95% CI : (0.4933, 0.6348)
- No Information Rate : 0.5
- P-Value [Acc > NIR] : 0.03842
- Mcnemar's Test P-Value : 2.659e-07
- Sensitivity : 0.3200
- Specificity : 0.8100
- Pos Pred Value : 0.6275
- Neg Pred Value : 0.5436
- Prevalence : 0.5000
- Detection Rate : 0.1600
- Detection Prevalence : 0.2550
- Balanced Accuracy : 0.5650
- Misclassification Rate on Test Set: 0.435

5.3 Model Comparison

Model	Accuracy
Full Logit Model	64%
Reduced Logit Model	63%
Classification Tree	54%

5.4 Conclusions

- The reduced logistic regression model focused only on predictors with significant p-values ($p < 0.05$). This model identified several city services that had a statistically significant impact on overall satisfaction ratings. Some of the key services that were significant include:
    - 911 services
    - Protection from river flooding
    - Disaster planning and response
    - Residential garbage collection
    - Blue and Green Cart recycling
    - Transportation planning
    - City operated roads and infrastructure
    - Snow removal
    - Bylaw services
    - Parks, recreation, and community services
    - City website and 311 service.
- These variables had coefficients suggesting the direction of their influence (positive or negative) on satisfaction. For example, services like Green Cart recycling and parks had positive coefficients, indicating higher satisfaction when these services were rated better.

The model achieved a notable reduction in deviance and a reasonable AIC score, suggesting a better fit compared to the null model.

- The tree had 26 terminal nodes, with a **misclassification rate of ~42.5%** and an accuracy of **~57.5%** on the test data. While it included many relevant city services as important decision points, the overall accuracy was modest, and the balanced accuracy indicated the model had challenges, especially in correctly classifying both satisfaction classes equally.
- Both models highlighted similar city services as important for predicting satisfaction. However, the logistic regression provided clearer statistical evidence through significance levels and coefficients, whereas the classification tree gave a more visual, rule-based structure but slightly lower predictive performance.

5.5 Future Work

- Apply ensemble methods such as random forests, gradient boosting (e.g., xgboost), or bagging to capture complex patterns and interactions missed by simpler models.
- If satisfaction data spans multiple years, perform a **longitudinal analysis** to see how the impact of services changes over time.

- Use mixed-effects models or time-series approaches.

- Instead of binning into binary classes, use **ordinal models** that respect this natural order for more nuanced results.
- Map satisfaction ratings geographically to see how service impact varies by location (e.g., wards or neighborhoods).

- Apply **geographically weighted regression (GWR)** or spatial trees.

- Explore how combinations of services (e.g., parks + transportation) jointly affect satisfaction.

- Use models that can automatically detect interactions (e.g., GAMs, boosted trees).

- Build models to **simulate the impact of hypothetical improvements** in specific services (e.g., “If snow removal satisfaction increased by 1 point, what would be the expected change in overall satisfaction?”)

6. References

1. *City of Calgary. (2023). Fall Survey of Calgarians 2023. [Dataset] City of Calgary Open Data Portal. Available at:* https://data.calgary.ca/Help-and-Information/Fall-Survey-of-Calgarians-2023-/kt4w-gkth [Accessed 30 May 2025].
2. Lohr, S.L. (2019). *Sampling: Design and Analysis*. 2nd ed. Boca Raton: CRC Press.
3. R Core Team. (2024). Logistic regression using glm() function. In: *R: A language and environment for statistical computing*. R Foundation for Statistical Computing.
4. Therneau, T. and Atkinson, B. (2024) *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1.23. Available at: https://CRAN.R-project.org/package=rpart (Accessed: 15 June 2025).

Contributions:

Riya Chevli – EDA

Deepika Gollamandala – Analysis-1

Romith Bondada – Analysis -2

R-code

https://github.com/romith05/Calgary-Citizen-satisfaction-survey-Analysis-