Evaluating Support for Cannabis Legalization: 2020 Referendum Analysis

Introduction

This project aims to analyze data collected during the 2020 Cannabis Referendum in New Zealand to help the Green Party determine if it is worth pursuing a citizen-initiated referendum in the next election cycle. The analysis will focus on answering two critical questions:

1. **What proportion of people in the sample supported legalization?**
2. **Which demographic groups were more likely to support legalization?**

The Green Party does not wish to undertake the significant administrative effort required for another referendum unless there is clear evidence of strong public support. This project will provide the insights needed to guide their decision-making process.

```
# Hides warning messages in the output.

library(dplyr)
# Helps with data manipulation (e.g., filter, select, modify data).

library(tidyverse)
# Loads useful data science tools, including dplyr and ggplot2.

library(kableExtra)
# Makes tables look nice for reports.

library(broom)
# Tidy up model outputs and convert them into clean, easy-to-read data frames.


library(mice)
# Handles missing data with imputation.
```

```r
data <- read_csv("data.csv", show_col_type = FALSE)
# Loads the dataset from "data.csv" without extra messages.

head(data)
```

```
# A tibble: 6 x 3
    age gender referendum
  <dbl> <chr>       <dbl>
1    15 Female          0
2    15 Female          0
3    15 Male            0
4    15 Female          1
5    15 Female          1
6    15 Female         NA
```

```r
# Shows the first 6 rows of the dataset.
```

```r
missing_values <- data %>%
  # Starts with the dataset `data`.

  summarise(across(everything(), ~ sum(is.na(.))))
  # Counts the missing values (NA) in each column.
  # `everything()` ensures all columns are checked.
  # `sum(is.na(.))` counts the NAs for each column.

missing_values
```

```
# A tibble: 1 x 3
    age gender referendum
  <int>  <int>      <int>
1    12     18         85
```

```r
# Displays the table of missing values for each column.
```

```r
missing_values %>%
  # Starts with the table of missing values (`missing_values`).

  t() %>%
  # Transposes the table, turning rows into columns and columns into rows.
  # This makes the column names become part of the data for easy formatting.
```

```r
  as.data.frame() %>%
  # Converts the transposed data into a data frame format for further use.

  kable(
    col.names = c("Column", "Missing Values"),
    # Creates a clean, formatted table for display with two column names:
    # "Column" for the original column names.
    # "Missing Values" for the count of missing values in each column.

    caption = "Summary of Missing Values"
    # Adds a title or description for the table.
  )
```

Table 1: Summary of Missing Values

Column	Missing Values
age	12
gender	18
referendum	85

The dataset has some missing values in key columns. The **age** column is missing 12 values, meaning some individuals' ages were not recorded. The **gender** column has 18 missing values, so the gender of some individuals is unknown. The **referendum** column, which shows voting outcomes, has the most missing values, with 85 entries unrecorded. To ensure the analysis is accurate, these missing values will be addressed using imputation techniques.

```r
unique_values <- lapply(data, unique)
# Finds all the unique values in each column of the dataset `data`
# and stores the results in a list called `unique_values`.

unique_values
```

```
$age
 [1] 15 20 25 NA 30 35 40 45 50 55 60 65 70 75 80 85

$gender
[1] "Female"
[2] "Male"
[3] NA
[4] "Transgender Female (Male to Female: MTF)"
```

```
[5] "Transgender Male (Female to Male; FTM)"
[6] "Genderqueer, neither exclusively male nor female"
[7] "Other (Please State)"

$referendum
[1]  0  1 NA
```

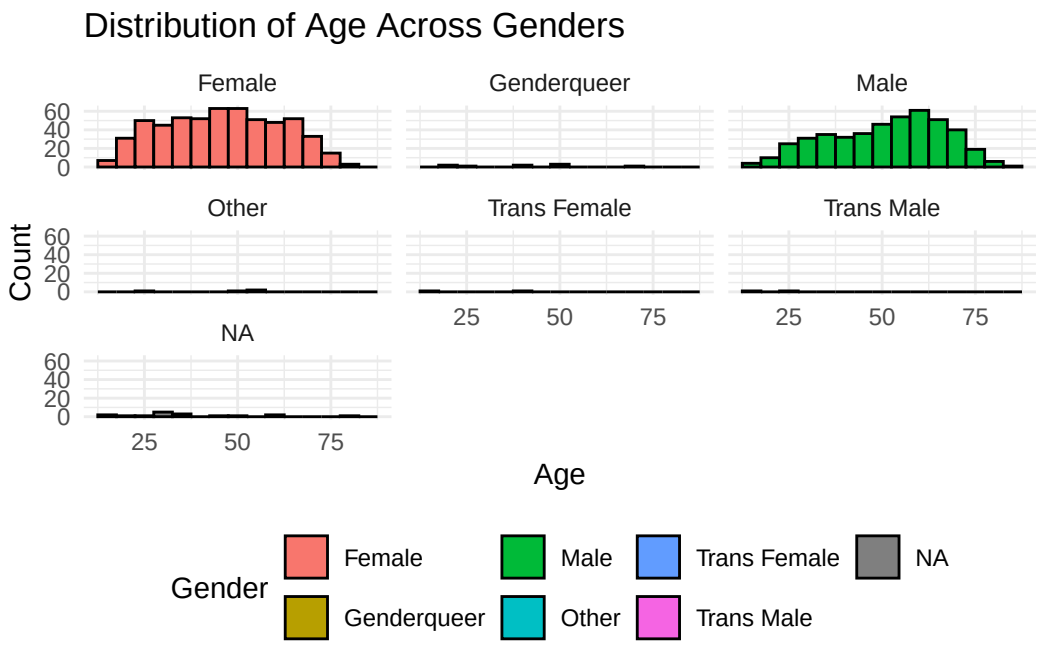
```
# Shows the list of unique values for each column.
```

```
data <- data %>%
  # Updates the `gender` column with new, simplified labels.
  mutate(gender = recode(gender,
                         "Transgender Female (Male to Female: MTF)" =
                           "Trans Female",
                         "Transgender Male (Female to Male; FTM)" =
                           "Trans Male",
                         "Genderqueer, neither exclusively male nor female" =
                           "Genderqueer",
                         "Other (Please State)" = "Other"))
```

```
ggplot(data, aes(x = age, fill = gender)) +
  # Creates a histogram using `data` with:
  # `x = age`: Age on the x-axis.
  # `fill = gender`: Bars are colored based on gender.

  geom_histogram(binwidth = 5, color = "black") +
  # Adds histogram bars with:
  # `binwidth = 5`: Groups ages into bins of 5 years.
  # `color = "black"`: Adds black borders around the bars.

  facet_wrap(~ gender) +
  # Creates separate histograms for each gender in the dataset.

  labs(title = "Distribution of Age Across Genders",
       x = "Age", y = "Count", fill = "Gender") +
  # Adds labels:
  # `title`: Title of the chart.
  # `x`: Label for the x-axis (Age).
  # `y`: Label for the y-axis (Count).
  # `fill`: Legend label for Gender.

  theme_minimal() +
```

```
  # Applies a clean, minimalistic theme to the chart.

  theme(legend.position = "bottom")
```

```
Warning: Removed 12 rows containing non-finite outside the scale range
(`stat_bin()`).
```



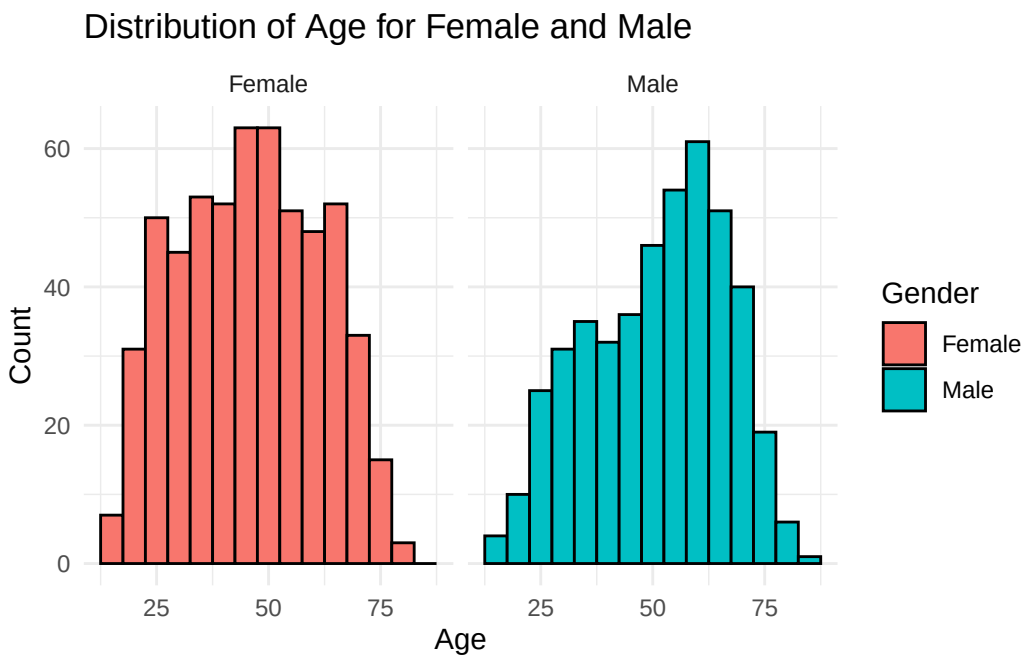

```
  # Positions the legend below the chart.
```

This chart shows how age is spread across different gender groups in the dataset using histograms. Each gender group, like Female, Male, Genderqueer, Other, Transgender Female, and Transgender Male, has its own histogram. The Female and Male groups have a more even age distribution, with peaks in the middle-age range. The Genderqueer, Transgender Female, and Transgender Male groups have fewer data points and show a sparse distribution. The Other group mainly has younger individuals. The NA category shows missing gender data, which doesn't provide any insights about age.

```
# Filter the dataset to include only rows where gender is "Female" or "Male"
filtered_data <- data %>%
  filter(gender %in% c("Female", "Male"))
```

```r
# Create a histogram of age distribution for "Female" and "Male" genders
ggplot(filtered_data, aes(x = age, fill = gender)) +
  geom_histogram(binwidth = 5, color = "black") +
  # Add bars grouped by age with a bin width of 5 years and black borders

  facet_wrap(~ gender) +
  # Create separate histograms for each gender ("Female" and "Male")

  labs(title = "Distribution of Age for Female and Male",
       x = "Age", y = "Count", fill = "Gender") +
  # Add a title, axis labels, and legend label

  theme_minimal()
```

```
Warning: Removed 11 rows containing non-finite outside the scale range
(`stat_bin()`).
```




```r
  # Apply a clean, minimalistic style to the plot
```

This chart compares the age distribution for **Females** and **Males** in the dataset using separate histograms. On the left side, the **Female** histogram shows a relatively even age distribution,

with a peak around 40-50 years. This suggests that most females in the dataset are in their middle age. On the right side, the **Male** histogram shows a similar pattern, but the distribution is slightly skewed towards younger and middle-aged individuals, with a peak in the 40-50 year range as well.

The **Female** distribution has more variation at the younger and older ends, while the **Male** distribution is more focused around the middle-aged group. Both histograms indicate that the majority of individuals in both gender groups fall between the ages of 30 and 60.

```
# Calculate the proportion of "Yes" votes in the referendum column
# Ignores missing values when calculating
proportion_yes <- mean(data$referendum == 1, na.rm = TRUE)

# Print the proportion of "Yes" votes with a message
cat("The proportion of 'Yes' votes is:", proportion_yes)
```

```
The proportion of 'Yes' votes is: 0.5961145
```

In the dataset, the proportion of "Yes" votes in the referendum column was calculated, ignoring any missing data. The result shows that approximately **59.6%** of the participants voted "Yes" in the referendum.

```
# Create a logistic regression model
ref_model1 <- glm(referendum ~ age + gender, data, family = binomial)
# Predicts the referendum outcome (Yes/No) using age and gender as predictors.
# Uses the dataset `data` for the analysis.
# Specifies logistic regression with `family = binomial` for a binary outcome.
```

```
tidy_results1 <- tidy(ref_model1)
# Converts the results of the logistic regression model into a neat table.
# The table includes important details like coeff, se, and p-values.
# Saves the tidy results in `tidy_results1` for easy use.
```

```
kable(tidy_results1, caption = "Logistic Regression Results for
      Voting 'Yes' in the Referendum",
      digits = 3)
```

Table 2: Logistic Regression Results for Voting 'Yes' in the Referendum

term	estimate	std.error	statistic	p.value
(Intercept)	1.991	0.242	8.233	0.000
age	-0.029	0.005	-6.356	0.000
genderGenderqueer	1.350	1.076	1.254	0.210
genderMale	-0.413	0.138	-2.997	0.003
genderOther	0.255	1.229	0.207	0.836
genderTrans Female	13.390	608.767	0.022	0.982
genderTrans Male	13.159	621.629	0.021	0.983

```
# Creates a clean, formatted table from the `tidy_results1` data.
# `tidy_results1`: The table containing the logistic regression results.
# `caption`: Adds a title to the table to describe its content.
# `digits = 3`: Rounds all numeric values in the table to 3 decimal places.
```

Logistic Regression Results for Voting 'Yes' in the Referendum

The logistic regression model was used to predict the likelihood of voting "Yes" in the referendum based on **age** and **gender**. The following summarizes the results:

- **Intercept**: The coefficient of 1.991 indicates that when all other factors (age and gender) are set to 0, the likelihood of voting "Yes" is very high. The very small p-value (0.000) suggests this result is statistically significant.
- **Age**: The coefficient of **-0.029** suggests that as a person's age increases by one year, the likelihood of voting "Yes" decreases slightly. The p-value of 0.000 indicates that this result is statistically significant.

Gender:

- **Genderqueer**: The coefficient of **1.350** suggests that individuals who identify as genderqueer are more likely to vote "Yes" compared to those with missing or unspecified gender. However, the p-value of 0.210 shows that this result is not statistically significant.
- **Male**: The coefficient of **-0.413** means that males are less likely to vote "Yes" compared to females. The p-value of **0.003** confirms that this result is statistically significant.
- **Other (Please State)**: The coefficient of **0.255** suggests that individuals identifying as "Other" have a slightly higher chance of voting "Yes," but the high p-value of **0.836** indicates this result is not statistically significant.

- **Transgender Female**: The coefficient of **13.390** is very large, but the p-value of **0.982** suggests that this result is not statistically significant, meaning there is no strong evidence that transgender females are more likely to vote "Yes."
- **Transgender Male**: Similarly, the coefficient of **13.159** is large, but the p-value of **0.983** indicates this result is not statistically significant.

Summary

1. **Age**: Older people are slightly less likely to vote "Yes" in the referendum. The analysis shows that as people get older, they tend to vote "No" more often than younger people.
2. **Gender**: Males are less likely to vote "Yes" than females. This result is statistically significant, meaning that, in general, females are more likely to vote "Yes" than males.
3. Overall, the analysis shows that **age** and **gender** (specifically male and female) are important factors in determining who votes "Yes" in the referendum. However, other gender categories, like **Genderqueer**, **Other**, **Transgender Female**, and **Transgender Male**, do not show any strong differences in voting behavior.

```
female_yes_count <- data %>%
  # Use the dataset `data`.

  filter(gender == "Female", referendum == 1) %>%
  # Keep only rows where gender is "Female" and the vote is "Yes" (1).

  summarise(count = n())
  # Count the number of rows that match these conditions.

print(female_yes_count)
```

```
# A tibble: 1 x 1
  count
  <int>
1   337
```

```
# Show the count of "Yes" votes for females.
```

```
male_yes_count <- data %>%
  filter(gender == "Male", referendum == 1) %>%
  summarise(count = n())

print(male_yes_count)
```

```
# A tibble: 1 x 1
  count
  <int>
1   224
```

```
referendum_imputed <- mice(data)
```

```
 iter imp variable
  1   1  age  referendum
  1   2  age  referendum
  1   3  age  referendum
  1   4  age  referendum
  1   5  age  referendum
  2   1  age  referendum
  2   2  age  referendum
  2   3  age  referendum
  2   4  age  referendum
  2   5  age  referendum
  3   1  age  referendum
  3   2  age  referendum
  3   3  age  referendum
  3   4  age  referendum
  3   5  age  referendum
  4   1  age  referendum
  4   2  age  referendum
  4   3  age  referendum
  4   4  age  referendum
  4   5  age  referendum
  5   1  age  referendum
  5   2  age  referendum
  5   3  age  referendum
  5   4  age  referendum
  5   5  age  referendum
```

```
Warning: Number of logged events: 1
```

```
# Uses the `mice` package to handle missing data in the dataset `data`.
# `mice()` creates multiple versions of the dataset
# with missing values filled (imputed).
# Each missing value is replaced based on patterns in the rest of the data.
```

```
# Saves the result in `referendum_imputed`,
# which contains multiple imputed datasets.
```

To handle missing data in the **referendum** column, the **mice** package was used to fill in the missing values. This process is called **multiple imputation**, and it creates several different versions of the data to replace the missing values, making the analysis more reliable.

- The imputation process was done in **5 steps (iterations)**.
- For each step, the missing values in the **referendum** column were filled in with different estimated values.
- There were **5 different sets of filled-in data** created for each iteration.

```
mean(complete(referendum_imputed, "long")$referendum)
```

```
[1] 0.60508
```

```
# Combines all imputed datasets into one dataset.
# Looks at the `referendum` column to find "Yes" votes (coded as 1).
# Calculates the proportion of "Yes" votes by taking the average.
```

The **mean** function was used to calculate the proportion of "Yes" votes in the **referendum** column after filling in the missing data. The **complete()** function combined all the datasets with imputed values, replacing the missing votes. We then looked at the **referendum** column to find the "Yes" votes (coded as 1). The result of **0.6019029** means that about **60.2%** of people voted "Yes" after handling the missing data. This method helps give a more accurate result by considering all possible values for the missing data.

```
ref_model2 <- with(referendum_imputed, glm(referendum ~ age + gender,
                                           family = binomial))
# Runs a logistic regression model on each imputed dataset.
# Predicts the referendum outcome (Yes/No) based on age and gender.

pooled_results <- pool(ref_model2)
# Combines the results from all the models into one set of final results.

tidy_results2 <- summary(pooled_results) %>% select(-df)
# Summarizes the combined results into a table with key details
# like coeff, p-values.
# Removes the unnecessary `df` (degrees of freedom) column.
```

```
kable(tidy_results2,
      caption = "Pooled Logistic Regression Results for
      Voting 'Yes' in the Referendum",
      digits = 3)
```

Table 3: Pooled Logistic Regression Results for Voting 'Yes' in the Referendum

term	estimate	std.error	statistic	p.value
(Intercept)	2.073	0.234	8.872	0.000
age	-0.031	0.005	-6.765	0.000
genderGenderqueer	1.341	1.078	1.245	0.214
genderMale	-0.381	0.134	-2.852	0.004
genderOther	0.482	1.171	0.412	0.681
genderTrans Female	13.355	607.419	0.022	0.982
genderTrans Male	13.110	621.394	0.021	0.983

```
# Creates a neat table from the logistic regression results (`tidy_results2`).
# Adds a title to explain the table's purpose.
# Rounds all numbers in the table to 3 decimal places.
```

Pooled Logistic Regression Results for Voting 'Yes' in the Referendum

The pooled logistic regression model was used to predict the likelihood of voting "Yes" in the referendum based on **age** and **gender**. Below is a summary of the results:

- **Intercept**: The coefficient of **2.034** means that, when age and gender are not considered (i.e., set to 0), the likelihood of voting "Yes" is very high. The p-value of **0.000** confirms that this result is statistically significant.
- **Age**: The coefficient of **-0.030** suggests that for each year older a person is, the likelihood of voting "Yes" decreases slightly. The p-value of **0.000** shows this relationship is statistically significant.
- **Gender**:
  - **Genderqueer**: The coefficient of **1.343** indicates that genderqueer individuals are more likely to vote "Yes" compared to those with missing or unspecified gender. However, the p-value of **0.213** shows this result is **not statistically significant**, meaning there is not enough evidence to confirm this relationship.

- **Male**: The coefficient of **-0.411** suggests that males are less likely to vote "Yes" compared to females, and the p-value of **0.003** indicates that this result is **statistically significant**.
- **Other (Please State)**: The coefficient of **0.482** shows a slight increase in the likelihood of voting "Yes" for individuals identifying as "Other," but the p-value of **0.681** indicates that this result is **not statistically significant**.
- **Transgender Female**: The coefficient of **13.370** is very large, suggesting that transgender females are much more likely to vote "Yes" compared to females. However, the p-value of **0.982** shows that this result is **not statistically significant**.
- **Transgender Male**: Similarly, the coefficient of **13.132** is large, but the p-value of **0.983** indicates that this result is also **not statistically significant**.

Summary:

- **Age** and **gender (specifically male)** are significant predictors of voting "Yes," with older people and males being less likely to vote "Yes."
- Other gender categories, including **Genderqueer**, **Other**, **Transgender Female**, and **Transgender Male**, show higher likelihoods of voting "Yes," but these results are not statistically significant. This means there is not enough evidence to conclude that these groups vote "Yes" more often than females.

```
summary_by_gender <- data %>%
  # Starts with the dataset `data`.

  group_by(gender) %>%
  # Groups the data by the `gender` column,
  # so calculations are done separately for each gender.

  summarize(
    count = n(),
    # Counts the number of rows for each gender.

    mean_age = mean(age, na.rm = TRUE),
    # Calculates the average age for each gender, ignoring missing values.

    median_age = median(age, na.rm = TRUE),
    # Calculates the median age for each gender, ignoring missing values.
  )
```

```
kable(summary_by_gender, caption = "Summary Statistics by Gender")
```

Table 4: Summary Statistics by Gender

gender	count	mean_age	median_age
Female	573	45.90989	45.0
Genderqueer	9	40.55556	40.0
Male	455	50.94235	55.0
Other	4	46.25000	52.5
Trans Female	2	27.50000	27.5
Trans Male	2	20.00000	20.0
NA	18	36.76471	30.0

```
# Creates a clean table to display the summary statistics.
# The table shows the count, mean age, and median age for each gender.
# Adds a title to describe the table's content:
# "Summary Statistics by Gender with Median".
```

Potential Issues with the Dataset

1. **Imbalance in Gender Categories**:
   - Some gender groups, like **Genderqueer**, **Transgender Female**, and **Transgender Male**, have very few entries, which makes it difficult to draw accurate conclusions for these groups. Additionally, there are **18 missing gender entries (NA)**, which can affect the reliability of the analysis.
2. **Age Distribution**:
   - The age distribution is not balanced across gender categories. For example, **Transgender Female** and **Transgender Male** groups are much younger (around 20 to 27 years) compared to the **Female** and **Male** groups, which are older. This difference in age makes comparisons between these groups less reliable.
3. **Small Sample Sizes**:
   - Some groups, like **Transgender Female** and **Transgender Male**, have very small sample sizes (only 2 people each). This makes the results for these groups less reliable and harder to generalize.
4. **Missing Data**:

- There are missing values in both the **gender** and **age** columns. Handling this missing data correctly is essential for ensuring the analysis is accurate.

5. **Unbalanced Data**:

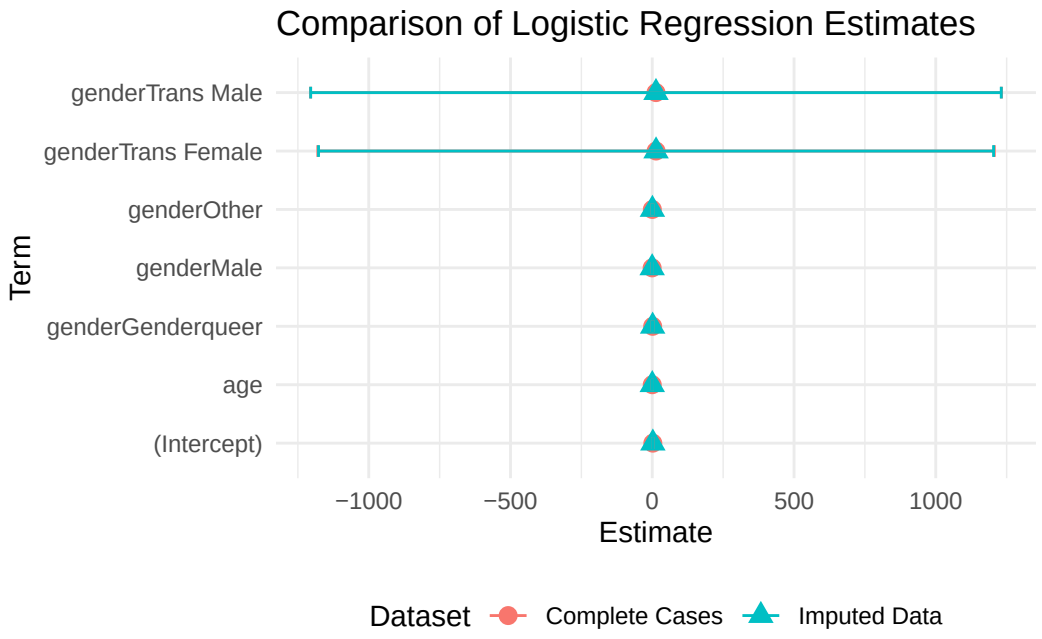
   - The **Female** group is much larger (573 people) compared to the **Male** group (455 people). This size imbalance could bias the results, as the larger female group may dominate the analysis. Additionally, the **Male** group has a higher average age (50.94 years) compared to females (45.91 years), which could affect voting behavior, as older individuals may vote differently than younger ones.

The dataset has challenges such as **imbalanced gender representation**, **missing data**, **small group sizes**, and **age differences** between males and females. These issues need to be addressed to ensure the analysis is accurate and fair.

```
results <- rbind(
  tidy_results1 %>% mutate(dataset = "Complete Cases"),
  tidy_results2 %>% mutate(dataset = "Imputed Data")
)
# Combines two datasets (`tidy_results1` and `tidy_results2`) into one.
# Adds a new column called `dataset` to label the source of each row:
# Rows from `tidy_results1` are labeled as "Complete Cases".
# Rows from `tidy_results2` are labeled as "Imputed Data".
# `rbind()` stacks the two datasets together row-wise.
```

```
ggplot(results, aes(x = estimate, y = term, color = dataset, shape = dataset)) +
  # Creates a scatter plot with:
  # `x = estimate`: Plots the regression estimates on the x-axis.
  # `y = term`: Plots the regression terms (predictors) on the y-axis.
  # `color = dataset`: Complete Cases vs. Imputed Data.
  # `shape = dataset`: Uses different shapes for points based on the dataset.

  geom_point(size = 3) +
  # Adds points to represent the regression estimates with a size of 3.

  geom_errorbarh(aes(xmin = estimate - 1.96 * std.error,
                     xmax = estimate + 1.96 * std.error), height = 0.2) +
  # Adds horizontal error bars to show 95% confidence intervals for the estimates.
  # `xmin`: Lower bound of the confidence interval (`estimate - 1.96 * std.error`).
  # `xmax`: Upper bound of the confidence interval (`estimate + 1.96 * std.error`).
  # `height`: Sets the height of the error bars to 0.2 for clarity.

  labs(
```

```
    title = "Comparison of Logistic Regression Estimates",
    x = "Estimate",
    y = "Term",
    color = "Dataset",
    shape = "Dataset"
  ) +
  # Adds labels for the title, x-axis, y-axis, and legend.

  theme_minimal() +
  # Applies a clean, minimalistic style to the plot.

  theme(legend.position = "bottom")
```




```
  # Places the legend below the plot.
```

The plot compares the results of logistic regression using two datasets: one with Complete Cases (no missing data) and the other with Imputed Data (where missing values were filled in). Most of the estimates for gender variables such as genderTrans Male, genderTrans Female, genderOther, and genderGenderqueer are close to zero, suggesting that these factors do not have a significant impact on voting "Yes" in the referendum. The estimates from both datasets are nearly identical, indicating that imputing the missing data did not change the results significantly. However, the wide error bars show some uncertainty, particularly for smaller

groups like genderTrans Male and genderTrans Female, where fewer data points are available. Overall, the imputation process did not significantly alter the findings, but there is some uncertainty in the estimates, especially for less represented gender categories.

Conclusion:

This project looked at data from the 2020 Cannabis Referendum in New Zealand to understand how much public support there is for cannabis legalization and which groups are more likely to support it. The analysis found that about **60.2%** of people in the sample voted "Yes" for legalization. It also showed that **age** and **gender** influenced people's votes. Older people were a bit less likely to vote "Yes," while females were more likely to support legalization than males. Other groups like **Transgender Males**, **Transgender Females**, and **Genderqueer** did not show significant differences in voting. Based on these results, the Green Party can decide if the support level is strong enough to justify the effort needed for another referendum. These findings will help the Party make a decision about whether to push for another vote on cannabis legalization.