miniessay 10

YANING JIN

```
install.packages("readxl")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)
```

```
install.packages("dplyr")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)
```

```
install.packages("tidyr")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)
```

```
install.packages("tidyverse")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)
```

```
install.packages("broom")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)
```

```
install.packages("ggplot2")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)
```

```
install.packages("MASS")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
(as 'lib' is unspecified)
```

```
library("ggplot2")
library("MASS")
library("tidyverse")
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v lubridate 1.9.3     v tibble    3.2.1
v purrr     1.0.2     v tidyr     1.3.1

-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
x dplyr::select() masks MASS::select()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library("broom")
library("readxl")
library("dplyr")
library("tidyr")
```

```
library("readxl")
library("dplyr")


data <- read_excel("Vote Totals Report From Official Tabulation - Hamilton Centre 036.xlsx",
```

```
New names:
* `` -> `...1`
* `` -> `...2`
* `` -> `...3`
```

```
* `` -> `...4`
* `` -> `...5`
* `` -> `...6`
* `` -> `...7`
* `` -> `...8`
* `` -> `...9`
* `` -> `...10`
* `` -> `...11`
* `` -> `...12`
* `` -> `...13`
* `` -> `...14`
* `` -> `...15`
* `` -> `...16`
* `` -> `...17`
* `` -> `...18`
* `` -> `...21`
```

```
head(data)
```

```
# A tibble: 6 x 23
  ...1  ...2  ...3  ...4  ...5  ...6  ...7  ...8  ...9  ...10 ...11 ...12 ...13
  <chr> <chr> <chr> <lgl> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
1 001   U     3     NA    61    246   0     245   0     1     4     92    0
2 002   U     5     NA    64    390   0     138   1     1     6     91    2
3 003   U     5     NA    61    427   2     171   7     2     11    101   3
4 004   U     1     NA    57    335   1     142   1     1     0     72    2
5 005   U     4     NA    50    374   6     198   5     1     3     115   0
6 006   U     0     NA    2     37    0     8     0     0     0     5     0
# i 10 more variables: ...14 <lgl>, ...15 <lgl>, ...16 <lgl>, ...17 <lgl>,
#   ...18 <lgl>, `REJETÉS EN RAISON DU MARQUAGE` <chr>,
#   `NON\r\nMARQUÉS PAR L'ÉLECTEUR` <chr>, ...21 <lgl>,
#   `BULLETINS REFUSÉS PAR LES ÉLECTEURS` <chr>,
#   `VOTING PLACE ADDRESS OR LOCATION / ADRESSE OU EMPLACEMENT DU BUREAU DE VOTE` <chr>
```

```
print(colnames(data))
```

```
 [1] "...1"
 [2] "...2"
 [3] "...3"
 [4] "...4"
 [5] "...5"
```

```
 [6] "...6"
 [7] "...7"
 [8] "...8"
 [9] "...9"
[10] "...10"
[11] "...11"
[12] "...12"
[13] "...13"
[14] "...14"
[15] "...15"
[16] "...16"
[17] "...17"
[18] "...18"
[19] "REJETÉS EN RAISON DU MARQUAGE"
[20] "NON\r\nMARQUÉS PAR L'ÉLECTEUR"
[21] "...21"
[22] "BULLETINS REFUSÉS PAR LES ÉLECTEURS"
[23] "VOTING PLACE ADDRESS OR LOCATION / ADRESSE OU EMPLACEMENT DU BUREAU DE VOTE"
```

```r
col_names <- c("Poll_Number", "Some_Other_Column", "Accepted_Ballots", "Column4", "Candidate1
               "Total_Ballots", "Ballots_Declined", "Column8", "Column9", "Column10", "Colum
               "Column12", "Column13", "Column14", "Column15", "Column16", "Column17", "Colu
               "Column19", "Column20", "Column21", "Column22", "Voting_Location")

names(data) <- col_names

data_cleaned <- data %>%
  select(Poll_Number, Accepted_Ballots, Candidate1_Votes, Total_Ballots, Ballots_Declined) %>
  mutate(
    Accepted_Ballots = as.integer(gsub(",", "", Accepted_Ballots)),
    Candidate1_Votes = as.integer(gsub(",", "", Candidate1_Votes)),
    Total_Ballots = as.integer(gsub(",", "", Total_Ballots)),
    Ballots_Declined = as.numeric(gsub(",", "", Ballots_Declined))
  )
```

```
Warning: There was 1 warning in `mutate()`.
i In argument: `Accepted_Ballots = as.integer(gsub(",", "",
  Accepted_Ballots))`.
Caused by warning:
! NAs introduced by coercion
```

```r
head(data_cleaned)
```

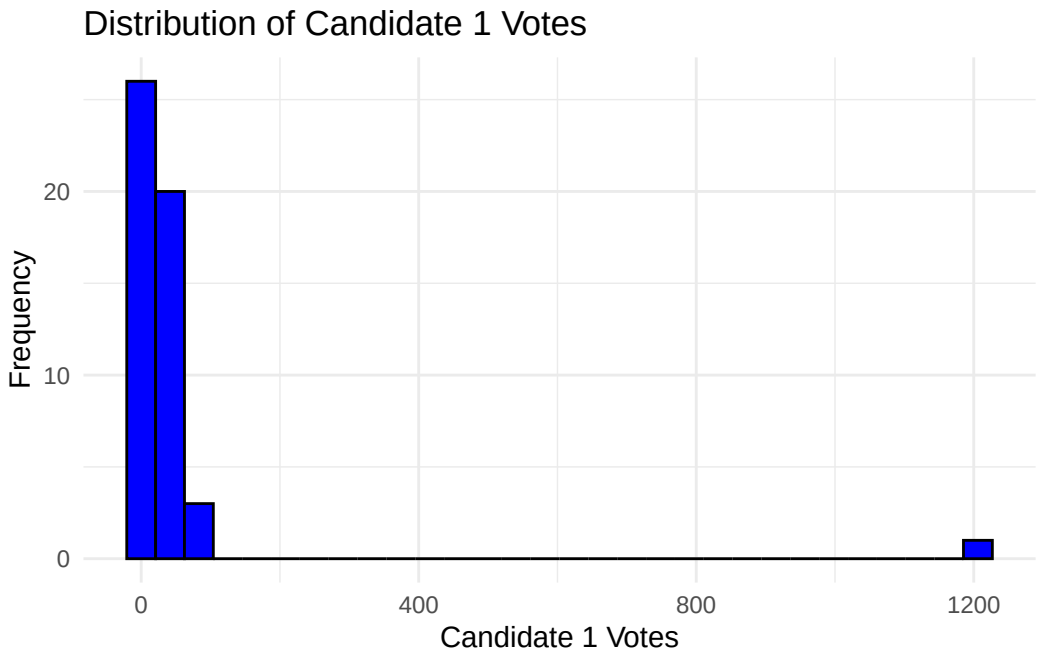
```
# A tibble: 6 x 5
  Poll_Number Accepted_Ballots Candidate1_Votes Total_Ballots Ballots_Declined
  <chr>                  <int>            <int>         <int>            <dbl>
1 001                        3               61           246                0
2 002                        5               64           390                0
3 003                        5               61           427                2
4 004                        1               57           335                1
5 005                        4               50           374                6
6 006                        0                2            37                0
```

Exploratory Data Analysis (EDA)

Before deciding on the model, perform EDA to understand the distribution of your outcome variable.

```r
library("ggplot2")
ggplot(data_cleaned, aes(x=Candidate1_Votes)) +
  geom_histogram(bins=30, fill="blue", color="black") +
  theme_minimal() +
  labs(title="Distribution of Candidate 1 Votes", x="Candidate 1 Votes", y="Frequency")
```

```
Warning: Removed 4 rows containing non-finite outside the scale range
(`stat_bin()`).
```



Model Estimation

Option A: Logistic Regression

If your outcome is binary, whether a candidate won a majority in each polling station, we might convert Candidate1_Votes into a binary variable and use logistic regression.

```
library("dplyr")

data_cleaned <- data_cleaned %>%
  mutate(Candidate1_Win = ifelse(Candidate1_Votes > Total_Ballots/2, 1, 0))

model_logistic <- glm(Candidate1_Win ~ Total_Ballots + Ballots_Declined, data=data_cleaned, f
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(model_logistic)
```

```
Call:
```

```
glm(formula = Candidate1_Win ~ Total_Ballots + Ballots_Declined,
    family = binomial, data = data_cleaned)

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)         7.0105     6.5661   1.068    0.286
Total_Ballots      -1.0905     0.9747  -1.119    0.263
Ballots_Declined  -21.4045  6793.8896  -0.003    0.997

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 16.7944  on 49  degrees of freedom
Residual deviance:  4.8439  on 47  degrees of freedom
  (4 observations deleted due to missingness)
AIC: 10.844

Number of Fisher Scoring iterations: 25
```

Option B: Poisson Regression

Poisson regression is appropriate for count data that meet the assumption that the mean is equal to the variance.

```
model_poisson <- glm(Candidate1_Votes ~ Total_Ballots + Ballots_Declined, data=data_cleaned,
summary(model_poisson)
```

```
Call:
glm(formula = Candidate1_Votes ~ Total_Ballots + Ballots_Declined,
    family = poisson, data = data_cleaned)

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)       3.0426381  0.0314108   96.87   <2e-16 ***
Total_Ballots     0.0021004  0.0001526   13.76   <2e-16 ***
Ballots_Declined -0.1616481  0.0146562  -11.03   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)
```

```
    Null deviance: 7267.09  on 49  degrees of freedom
Residual deviance:  851.94  on 47  degrees of freedom
  (4 observations deleted due to missingness)
AIC: 1073.4

Number of Fisher Scoring iterations: 5
```

Option C: Negative Binomial Regression

If the count data are overdispersed which is the variance exceeds the mean, negative binomial regression is more appropriate.

```
model_negbin <- MASS::glm.nb(Candidate1_Votes ~ Total_Ballots + Ballots_Declined, data=data_
summary(model_negbin)
```

```
Call:
MASS::glm.nb(formula = Candidate1_Votes ~ Total_Ballots + Ballots_Declined,
    data = data_cleaned, init.theta = 0.9269733256, link = log)

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)       2.9760859  0.1561980  19.053   <2e-16 ***
Total_Ballots     0.0017253  0.0008147   2.118   0.0342 *
Ballots_Declined -0.1150320  0.0786546  -1.462   0.1436
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(0.927) family taken to be 1)

    Null deviance: 122.463  on 49  degrees of freedom
Residual deviance:  58.863  on 47  degrees of freedom
  (4 observations deleted due to missingness)
AIC: 428

Number of Fisher Scoring iterations: 1

              Theta:  0.927
          Std. Err.:  0.191

 2 x log-likelihood:  -419.998
```

Exploratory Data Analysis (EDA)

The EDA step involves visualizing the distribution of `Candidate1_Votes` using a histogram. This visualization is crucial as it provides insights into the distribution of the count data, including its skewness, presence of outliers, and whether it resembles the Poisson distribution (where the mean roughly equals the variance) or exhibits overdispersion (variance significantly exceeds the mean).

Logistic Regression

The logistic regression model is used to predict a binary outcome based on predictor variables. In this context, created a binary variable `Candidate1_Win` to indicate whether a candidate won a majority of votes in each polling station. Logistic regression is suitable for questions about the probability of an event (e.g., winning) occurring given a set of predictors. Applicability: If the research question is specifically about the likelihood of a candidate winning or losing based on other factors, logistic regression is the appropriate model.

Poisson Regression

Poisson regression is used for modeling count data, particularly when the data distribution is expected to follow the Poisson distribution, where the mean count is approximately equal to its variance.

Applicability: This model is suitable if `Candidate1_Votes` across polling stations follows a Poisson distribution. If the EDA indicates that the mean and variance of votes are roughly equal and there's no overdispersion, Poisson regression can effectively model the vote count based on predictors like `Total_Ballots` and `Ballots_Declined`.

Negative Binomial Regression

Negative binomial regression extends the Poisson regression to account for overdispersion in the count data, where the variance exceeds the mean. This is a more flexible model that includes an additional parameter to model the variance independently of the mean.

Applicability: If the EDA reveals overdispersion in `Candidate1_Votes`, which is common in real-world count data, negative binomial regression is more appropriate than Poisson regression. It can more accurately estimate the effects of predictors on the vote count while properly accounting for the extra variability in the data.

Decision Criteria and Discussion

-Based on EDA: The choice between these models should first consider the EDA findings. If the vote counts are overdispersed, negative binomial regression is preferred. If the data appear to follow a Poisson distribution (mean variance), Poisson regression is suitable.

Research Question: The choice also depends on the research question. If the interest lies in predicting binary outcomes (e.g., win or lose), logistic regression is the way to go. For modeling the count of votes as influenced by other variables, depending on the presence or absence of overdispersion, choose between Poisson and negative binomial regression.

Model Diagnostics: After fitting the models, it's crucial to perform diagnostics. For logistic regression, check the model's goodness-of-fit (e.g., Hosmer-Lemeshow test) and the area under the ROC curve (AUC). For Poisson and negative binomial regressions, assess the dispersion and the fit to ensure that the chosen model appropriately addresses the data's characteristics.