[INSERT YOUR NAME and UID]

Problem Set 1+2 (15% + 15%)

Due: 2023-12-3 23:59 (HKT)

General Introduction

In this Problem Set, you will apply data science skills to wrangle and visualize the replication data of the following research article:

Cantú, F. (2019). The fingerprints of fraud: Evidence from Mexico's 1988 presidential election. *American Political Science Review*, *113*(3), 710-726.

Requirements and Reminders

- You are required to use **RMarkdown** to compile your answer to this Problem Set.
- Two submissions are required (via Moodle)
  - A `.pdf` file rendered by `Rmarkdown` that contains all your answer.
  - A compressed (in `.zip` format) R project repo. The expectation is that the instructor can unzip, open the project file, knitr your `.Rmd` file, and obtain the exact same output as the submitted `.pdf` document.
- The Problem Set is worth 30 points in total, allocated across 7 tasks. The point distribution across tasks is specified in the title line of each task. Within each task, the points are evenly distributed across sub-tasks. Bonus points (+5% max.) will be awarded to recognize exceptional performance.
- Grading rubrics: Overall, your answer will be evaluated based on its quality in three dimensions
  - Correctness and beauty of your outputs
  - Style of your code
  - Insightfulness of your interpretation or discussion
- Unless otherwise specified, you are required to use functions from the `tidyverse` package to complete this assignments.
- Fo some tasks, they may be multiple ways to achieve the same desired outcomes. You are encouraged to explore multiple methods. If you perform a task using multiple methods, do show it in your submission. You may earn bonus points for it.
- You are encouraged to use Generative AI such as ChatGPT to assist with your work. However, you will need to acknowledge it properly and validate AI's outputs. You may attach selected chat history with the AI you use and describe how it helps you get the work done. Extra credit may be rewarded to recognize creative use of Generative AI.
- This Problem Set is an individual assignment. You are expected to complete it independently. Clarification questions are welcome. Discussions on concepts and techniques related to the Problem Set among peers is encouraged. However, without the instructor's consent, sharing (sending and requesting) code and text that complete the entirety of a task is prohibited. You are strongly encouraged to use *CampusWire* for clarification questions and discussions.

Background

In 1998, Mexico had a close presidential election. Irregularities were detected around the country during the voting process. For example, when 2% of the vote tallies had been counted, the preliminary results showed the PRI's imminent defeat in Mexico City metropolitan area and a very narrow vote margin between PRI and FDN. A few minutes later, the screens at the Ministry of Interior went blank, an event that electoral authorities justified as a technical problem caused by an overload on telephone lines. The vote count was therefore suspended for three days, despite the fact that opposition representatives found a computer in the basement that continued to receive electoral results. Three days later, the vote count resumed, and soon the official announced PRI's winning with 50.4% of the vote.

What happened on that night and the following days? Were there electoral fraud during the election? A political scientist, Francisco Cantú, unearths a promising dataset that could provide some clues. At the National Archive in Mexico City, Cantú discovered about 53,000 vote tally sheets. Using machine learning methods, he detected that a significant number of tally sheets were *altered*! In addition, he found evidence that the altered tally sheets were biased in favor of the incumbent party. In this Problem Set, you will use Cantú's replication dossier to replicate and extend his data work.

Please read Cantú (2019) for the full story. And see Figure 1 for a few examples of altered (fraudulent) tallies.

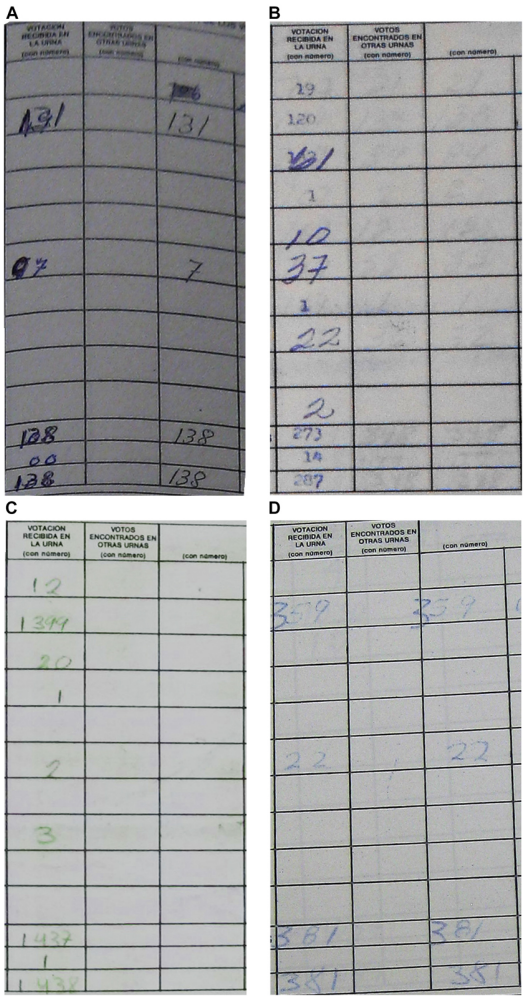


Figure 1: Examples of altered tally sheets (reproducing Figure 1 of Cantú 2018)

Task 0. Loading required packages (3pt)

For Better organization, it is a good habit to load all required packages up front at the start of your document. Please load the all packages you use throughout the whole Problem Set here.

```
# YOUR CODE HERE
library(tidyverse)
library(dplyr)
library(stringr)
library(ggplot2)
library(gridExtra)
library(cowplot)
library(sf)
```

Task 1. Clean machine classification results (3pt)

Cantú applys machine learning models to 55,334 images of tally sheets to detect signs of fraud (i.e., alteration). The machine learning model returns results recorded in a table. The information in this table is messy and requires data wrangling before we can use them.

Task 1.1. Load classified images of tally sheets

The path of the classified images of tally sheets is `data/classification.txt`. Your first task is loading these data onto R using a `tidyverse` function. Name it `d_tally`.

Note:

- Although the file extension of this dataset is `.txt`, you are recommended to use the `tidyverse` function we use for `.csv` files to read it.
- Unlike the data files we have read in class, this table has *no column names*. Look up the documentation and find a way to handle it.
- There will be three columns in this dataset, name them `name_image`, `label`, and `probability`.

Print your table to show your output.

```
# YOUR CODE HERE
d_tally<-read_csv("data/classification.txt")
colnames(d_tally) <- c("name_image", "label", "probability")
print(d_tally)
```

```
## # A tibble: 55,333 x 3
##    name_image                                label probability
##    <chr>                                     <chr> <chr>
##  1 Aguascalientes_I_2014-05-26 00.00.17.jpg [[0]] [[ 0.95722806]]
##  2 Aguascalientes_I_2014-05-26 00.00.25.jpg [[0]] [[ 0.57690716]]
##  3 Aguascalientes_I_2014-05-26 00.00.31.jpg [[0]] [[ 0.96505082]]
##  4 Aguascalientes_I_2014-05-26 00.00.38.jpg [[0]] [[ 0.86975688]]
##  5 Aguascalientes_I_2014-05-26 00.00.45.jpg [[0]] [[ 0.78825063]]
##  6 Aguascalientes_I_2014-05-26 00.00.52.jpg [[0]] [[ 0.96493018]]
##  7 Aguascalientes_I_2014-05-26 00.00.59.jpg [[0]] [[ 0.68087846]]
##  8 Aguascalientes_I_2014-05-26 00.01.06.jpg [[0]] [[ 0.99999994]]
##  9 Aguascalientes_I_2014-05-26 00.01.15.jpg [[0]] [[ 0.64047635]]
## 10 Aguascalientes_I_2014-05-26 00.01.50.jpg [[1]] [[ 0.93540865]]
## # i 55,323 more rows
```

Note 1. What are in this dataset?

Before you proceed, let me explain the meaning of the three variables.

- `name_image` contains the names of of the tallies' image files (as you may infer from the `.jpg` file extensions. They contain information about the locations where each of the tally sheets are produced.
- `label` is a machine-predicted label indicating whether a tally is fraudulent or not. `label = 1` means the machine learning model has detected signs of fraud in the tally sheet. `label = 0` means the machine detects no sign of fraud in the tally sheet. In short, `label = 1` means fraud; `label = 0` means no fraud.
- `probability` indicates the machine's certainty about its predicted `label` (explained above). It ranges from 0 to 1, where higher values mean higher level of certainty.

Interpret `label` and `probability` carefully. Two examples can hopefully give you clues about their correct interpretation. In the first row, `label = 0` and `probability = 0.9991`. That means the machine thinks this tally sheet is NOT FRAUDULENT with a probability of 0.9991. Then, the probability that this tally sheet is fraudulent is `1 - 0.9991 = 0.0009`. Take another example, in the 11th row, `label = 1` and `probability = 0.935`. This means the machine thinks this tally sheet IS FRAUDULENT with a probability of 0.935. Then, the probability that it is NOT FRAUDULENT is `1 - 0.9354 = 0.0646`.

Task 1.2. Clean columns `label` and `probability`

As you have seen in the printed outputs, columns `label` and `probability` are read as `chr` variables when they are actually numbers. A close look at the data may tell you why — they are "wrapped" by some non-numeric characters. In this task, you will clean these two variables and make them valid numeric variables. You are required to use `tidyverse` operations to for this task. Show appropriate summary statistics of `label` and `probability` respectively after you have transformed them into numeric variables.

```
# YOUR CODE HERE

d_tally$label <- as.numeric(str_remove_all(d_tally$label, "\\[\\[|\\]\\]"))
d_tally$probability <- as.numeric(str_remove_all(d_tally$probability, "\\[\\[|\\]\\]"))
print(d_tally)
```

```
## # A tibble: 55,333 x 3
##    name_image                                label probability
##    <chr>                                     <dbl>       <dbl>
##  1 Aguascalientes_I_2014-05-26 00.00.17.jpg     0       0.957
##  2 Aguascalientes_I_2014-05-26 00.00.25.jpg     0       0.577
##  3 Aguascalientes_I_2014-05-26 00.00.31.jpg     0       0.965
##  4 Aguascalientes_I_2014-05-26 00.00.38.jpg     0       0.870
##  5 Aguascalientes_I_2014-05-26 00.00.45.jpg     0       0.788
##  6 Aguascalientes_I_2014-05-26 00.00.52.jpg     0       0.965
##  7 Aguascalientes_I_2014-05-26 00.00.59.jpg     0       0.681
##  8 Aguascalientes_I_2014-05-26 00.01.06.jpg     0       1.00
##  9 Aguascalientes_I_2014-05-26 00.01.15.jpg     0       0.640
## 10 Aguascalientes_I_2014-05-26 00.01.50.jpg     1       0.935
## # i 55,323 more rows
```

Task 1.3. Extract state and district information from `name_image`

As explained in the note, the column `name_image`, which has the names of tally sheets' images, contains information about locations where the tally sheets are produced. Specifically, the first two elements of these file names indicates the **states'** and districts' identifiers respectively, for example, `name_image = "Aguascalientes_I_2014-05-26 00.00.10.jpg"`. It means this tally sheet is produced in state **`Aguascalientes`**, district `I`. In this task, you are required to obtain this information. Specifically, create two columns named `state` and `district` as state and district identifiers respectively. You are required to use `tidyverse` functions to perform the task.

```
# YOUR CODE HERE

# Method 1
d_tally <- d_tally |>
  mutate(
    state = str_extract(name_image, "^[^_]+"),
    district = str_extract(name_image, "(?<=_)[^_]+(?=_)")
  )
print(d_tally)
```

```
## # A tibble: 55,333 x 5
##    name_image                                label probability state     district
##    <chr>                                     <dbl>       <dbl> <chr>     <chr>
##  1 Aguascalientes_I_2014-05-26 00.00.17.jpg     0       0.957 Aguascal~ I
##  2 Aguascalientes_I_2014-05-26 00.00.25.jpg     0       0.577 Aguascal~ I
##  3 Aguascalientes_I_2014-05-26 00.00.31.jpg     0       0.965 Aguascal~ I
##  4 Aguascalientes_I_2014-05-26 00.00.38.jpg     0       0.870 Aguascal~ I
##  5 Aguascalientes_I_2014-05-26 00.00.45.jpg     0       0.788 Aguascal~ I
##  6 Aguascalientes_I_2014-05-26 00.00.52.jpg     0       0.965 Aguascal~ I
##  7 Aguascalientes_I_2014-05-26 00.00.59.jpg     0       0.681 Aguascal~ I
##  8 Aguascalientes_I_2014-05-26 00.01.06.jpg     0       1.00  Aguascal~ I
##  9 Aguascalientes_I_2014-05-26 00.01.15.jpg     0       0.640 Aguascal~ I
## 10 Aguascalientes_I_2014-05-26 00.01.50.jpg     1       0.935 Aguascal~ I
## # i 55,323 more rows
```

```
# Method 2
#d_tally <- d_tally |>
#  separate(name_image, into = c("state", "district"), sep = "_", remove = FALSE)
#print(d_tally)
```

Task 1.4. Re-code a state's name

One of the states (in the newly created column `state`) is coded as "`Estado de Mexico`." The researchers decide that it should instead re-coded as "**`Edomex`**." Please use a `tidyverse` function to perform this task.

Hint: Look up functions `ifelse` and `case_match`.

```
# YOUR CODE HERE

# Method 1
d_tally <- d_tally |>
  mutate(
    state = case_when(
      state == "Estado de Mexico" ~ "Edomex",
      TRUE ~ state))

# Method 2
#d_tally <- d_tally |>
#          mutate(state = ifelse(state == "Estado de Mexico", "Edomex", state))
#print(d_tally)
```

Task 1.5. Create a *probability of fraud* indicator

As explained in Note 1, we need to interpret `label` and `probability` with caution, as the meaning of `probability` is conditional on the value of `label`. To avoid confusion in the analysis, your next task is to create a column named `fraud_proba` which indicates the probability that a tally sheet is is fraudulent. After you have created the column, drop the `label` and `probability` columns.

Hint: Look up the `ifelse` function and the `case_when` function (but you just need either one of them).

```
# YOUR CODE HERE

# Method 1
d_tally <- d_tally |>
  mutate(
    fraud_proba = ifelse(label == 0, 1 - probability, probability),
    fraud_proba = round(fraud_proba, 4)
    ) |>
  select(-label, -probability)
print(d_tally)
```

```
## # A tibble: 55,333 x 4
##    name_image                                state          district fraud_proba
##    <chr>                                     <chr>          <chr>          <dbl>
##  1 Aguascalientes_I_2014-05-26 00.00.17.jpg Aguascalientes I             0.0428
##  2 Aguascalientes_I_2014-05-26 00.00.25.jpg Aguascalientes I             0.423
##  3 Aguascalientes_I_2014-05-26 00.00.31.jpg Aguascalientes I             0.0349
##  4 Aguascalientes_I_2014-05-26 00.00.38.jpg Aguascalientes I             0.130
##  5 Aguascalientes_I_2014-05-26 00.00.45.jpg Aguascalientes I             0.212
##  6 Aguascalientes_I_2014-05-26 00.00.52.jpg Aguascalientes I             0.0351
##  7 Aguascalientes_I_2014-05-26 00.00.59.jpg Aguascalientes I             0.319
##  8 Aguascalientes_I_2014-05-26 00.01.06.jpg Aguascalientes I             0
##  9 Aguascalientes_I_2014-05-26 00.01.15.jpg Aguascalientes I             0.360
## 10 Aguascalientes_I_2014-05-26 00.01.50.jpg Aguascalientes I             0.935
## # i 55,323 more rows
```

```
# Method 2
# d_tally <- d_tally |>
#   mutate(
#     fraud_proba = case_when(
#       label == 0 ~ 1 - probability,
#       TRUE ~ probability),
#     fraud_proba = round(fraud_proba, 4)) |>
#   select(-label, -probability)
#print(d_tally)
```

Task 1.6. Create a binary *fraud* indicator

In this task, you will create a binary indicator called `fraud_bin` in indicating whether a tally sheet is fraudulent. Following the researcher's rule, we consider a tally sheet fraudulent only when the machine thinks it is at least 2/3 likely to be fraudulent. That is, `fraud_bin` is set to TRUE when `fraud_proba` is greater to `2/3` and is FALSE otherwise.

```
# YOUR CODE HERE
d_tally <- d_tally |>
  mutate(
    fraud_bin = fraud_proba > 2/3)
print(d_tally)
```

```
## # A tibble: 55,333 x 5
##    name_image                                state district fraud_proba fraud_bin
##    <chr>                                     <chr> <chr>          <dbl> <lgl>
##  1 Aguascalientes_I_2014-05-26 00.00.17.jpg Agua~ I              0.0428 FALSE
##  2 Aguascalientes_I_2014-05-26 00.00.25.jpg Agua~ I              0.423  FALSE
##  3 Aguascalientes_I_2014-05-26 00.00.31.jpg Agua~ I              0.0349 FALSE
##  4 Aguascalientes_I_2014-05-26 00.00.38.jpg Agua~ I              0.130  FALSE
##  5 Aguascalientes_I_2014-05-26 00.00.45.jpg Agua~ I              0.212  FALSE
##  6 Aguascalientes_I_2014-05-26 00.00.52.jpg Agua~ I              0.0351 FALSE
##  7 Aguascalientes_I_2014-05-26 00.00.59.jpg Agua~ I              0.319  FALSE
##  8 Aguascalientes_I_2014-05-26 00.01.06.jpg Agua~ I              0      FALSE
##  9 Aguascalientes_I_2014-05-26 00.01.15.jpg Agua~ I              0.360  FALSE
## 10 Aguascalientes_I_2014-05-26 00.01.50.jpg Agua~ I              0.935  TRUE
## # i 55,323 more rows
```

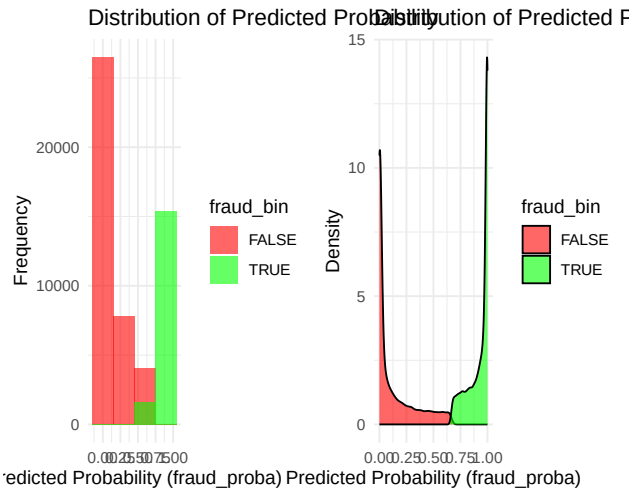
Task 2. Visualize machine classification results (3pt)

In this section, you will visualize the `tally` dataset that you have cleaned in Task 1. Unless otherwise specified, you are required to use the `ggplot` packages to perform all the tasks.

Task 2.1. Visualize distribution of `fraud_proba`

How is the predicted probability of fraud (`fraud_proba`) distributed? Use two methods to visualize the distribution. Remember to add informative labels to the figure. Describe the plot with a few sentences.

```
# YOUR CODE HERE

# Method 1: Histogram
hist_plot <- ggplot(d_tally, aes(x = fraud_proba, fill = fraud_bin)) +
  geom_histogram(binwidth = 0.3, position = "identity", alpha = 0.6) +
  labs(title = "Distribution of Predicted Probability",
       x = "Predicted Probability (fraud_proba)",
       y = "Frequency") +
  scale_fill_manual(values = c("FALSE" = "red", "TRUE" = "green")) +
  theme_minimal()

# Method 2: Density Plot
density_plot <- ggplot(d_tally, aes(x = fraud_proba, fill = fraud_bin)) +
  geom_density(alpha = 0.6) +
  labs(title = "Distribution of Predicted Probability",
       x = "Predicted Probability (fraud_proba)",
       y = "Density") +
  scale_fill_manual(values = c("FALSE" = "red", "TRUE" = "green")) +
  theme_minimal()

# Placing the graphs
grid.arrange(hist_plot, density_plot, ncol = 2)
```

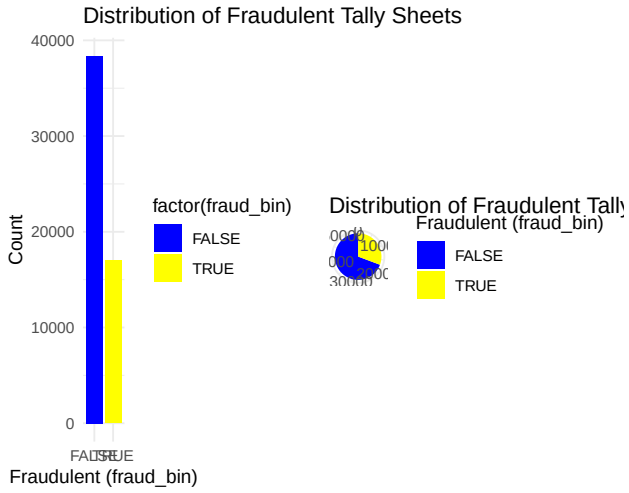



```
#The histogram: a binned representation of probability values.
#The density plot: a smoother view of the distribution, emphasizing the overall shape.
#The green areas: the portion of tallies predicted as fraudulent.
```

Task 2.2. Visualize distribution of `fraud_bin`

How many tally sheets are fraudulent and how many are not? We may answer this question by visualizing the binary indicator of tally-level states of fraud. Use at least two methods to visualize the distribution of `fraud_bin`. Remember to add informative labels to the figure. Describe your plots with a few sentences.

```
# YOUR CODE HERE

# Method 1: Bar Plot
bar_plot <- ggplot(d_tally, aes(x = factor(fraud_bin), fill = factor(fraud_bin))) +
  geom_bar() +
  labs(title = "Distribution of Fraudulent Tally Sheets",
       x = "Fraudulent (fraud_bin)",
       y = "Count") +
  scale_fill_manual(values = c("FALSE" = "blue", "TRUE" = "yellow")) +
  theme_minimal()

# Method 2: Pie Chart
pie_chart <- ggplot(d_tally, aes(x = "", fill = factor(fraud_bin))) +
  geom_bar(width = 1, stat = "count") +
  coord_polar("y") +
  labs(title = "Distribution of Fraudulent Tally Sheets",
       fill = "Fraudulent (fraud_bin)",
       x = NULL,
       y = NULL) +
  scale_fill_manual(values = c("FALSE" = "blue", "TRUE" = "yellow")) +
  theme_minimal()

# Placing the graphs
grid.arrange(bar_plot, pie_chart, ncol = 2)
```




```
#Bar plot: the blue and yellow bars represent non-fraudulent and fraudulent tally sheets.
#Pie chart: a circular visualization, the yellow slice indicates proportion of fraudulent sheets relati
```

The figure below serve as a reference. Feel free to try alternative approach(es) to make your visualization nicer and more informative.

Task 2.3. Summarize prevalence of fraud by state

Next, we will examine the between-state variation with regards to the prevalence of election fraud. In this task, you will create a new object that contains two state-level indicators regarding the prevalence of election fraud: The count of fraudulent tallies and the proportion of fraudulent tallies.

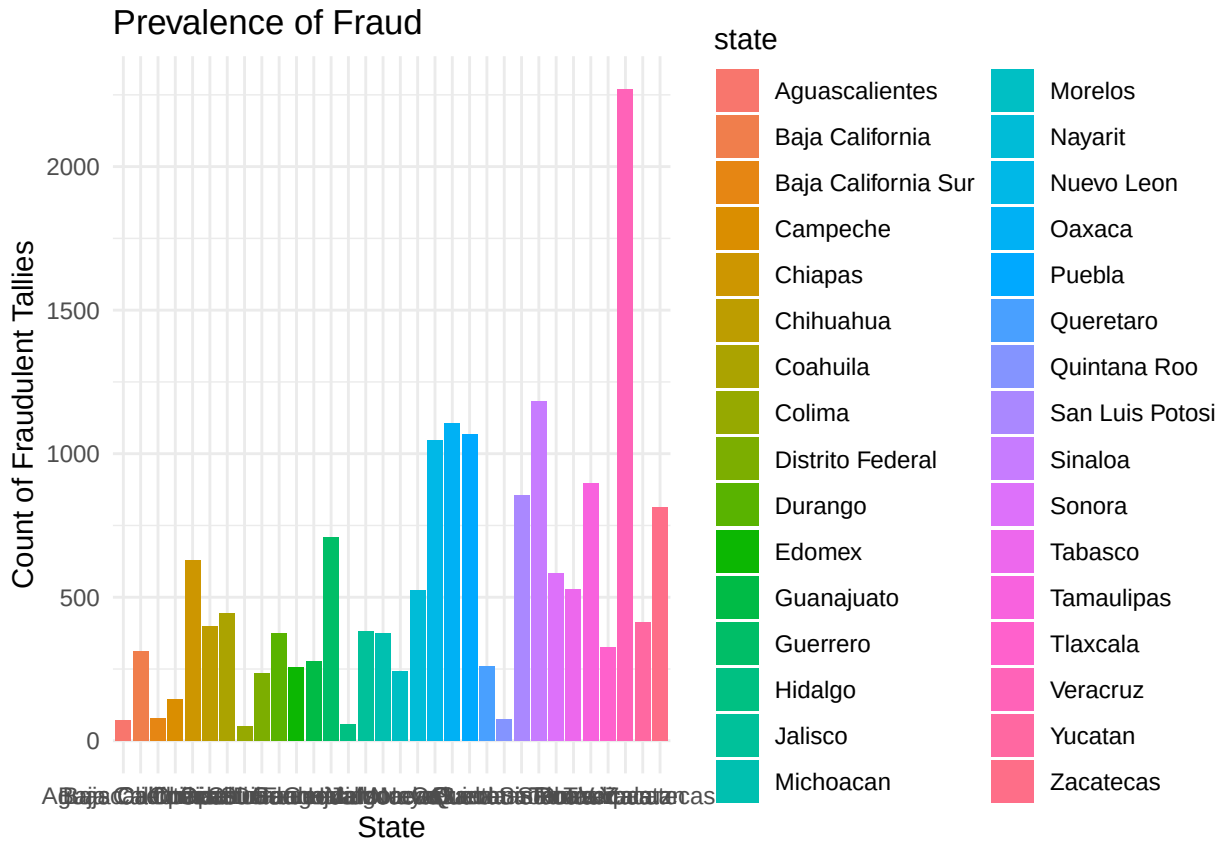
```
# YOUR CODE HERE
state_level_indicators_summary <- d_tally |>
  group_by(state) |>
  summarize(
    count_fraudulent = sum(fraud_bin),
    proportion_fraudulent = mean(fraud_bin) * 100  # Multiply by 100 for percentage
  )
print(state_level_indicators_summary)
```

```
## # A tibble: 32 x 3
##    state               count_fraudulent proportion_fraudulent
##    <chr>                          <int>                 <dbl>
##  1 Aguascalientes                    71                 17.7
##  2 Baja California                  311                 23.1
##  3 Baja California Sur               79                 19.1
##  4 Campeche                         146                 38.6
##  5 Chiapas                          629                 45.6
##  6 Chihuahua                        398                 21.4
##  7 Coahuila                         444                 37.8
##  8 Colima                            51                 16.8
##  9 Distrito Federal                 236                  3.10
## 10 Durango                          376                 27.8
## # i 22 more rows
```

Task 2.4. Visualize frequencies of fraud by state

Using the new data frame created in Task 2.3, please visualize the *frequencies* of fraudulent tallies of every state. Describe the key takeaway from the visualization with a few sentences.

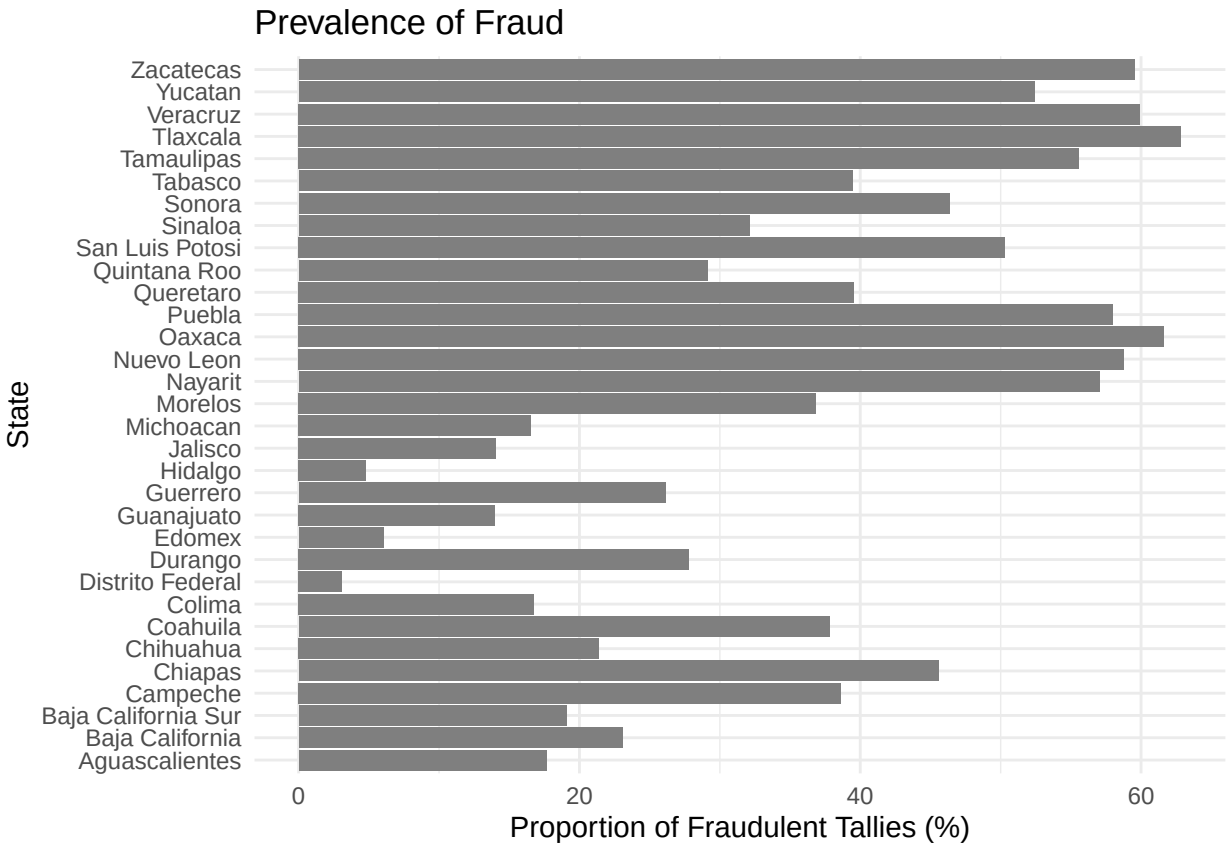
Feel free to try alternative approach(es) to make your visualization nicer and more informative.



Task 2.5. Visualize proportions of fraud by state

Using the new data frame created in Task 2.3, please visualize the *proportion of* of fraudulent tallies of every state. Describe the key takeaway from the visualization with a few sentences.

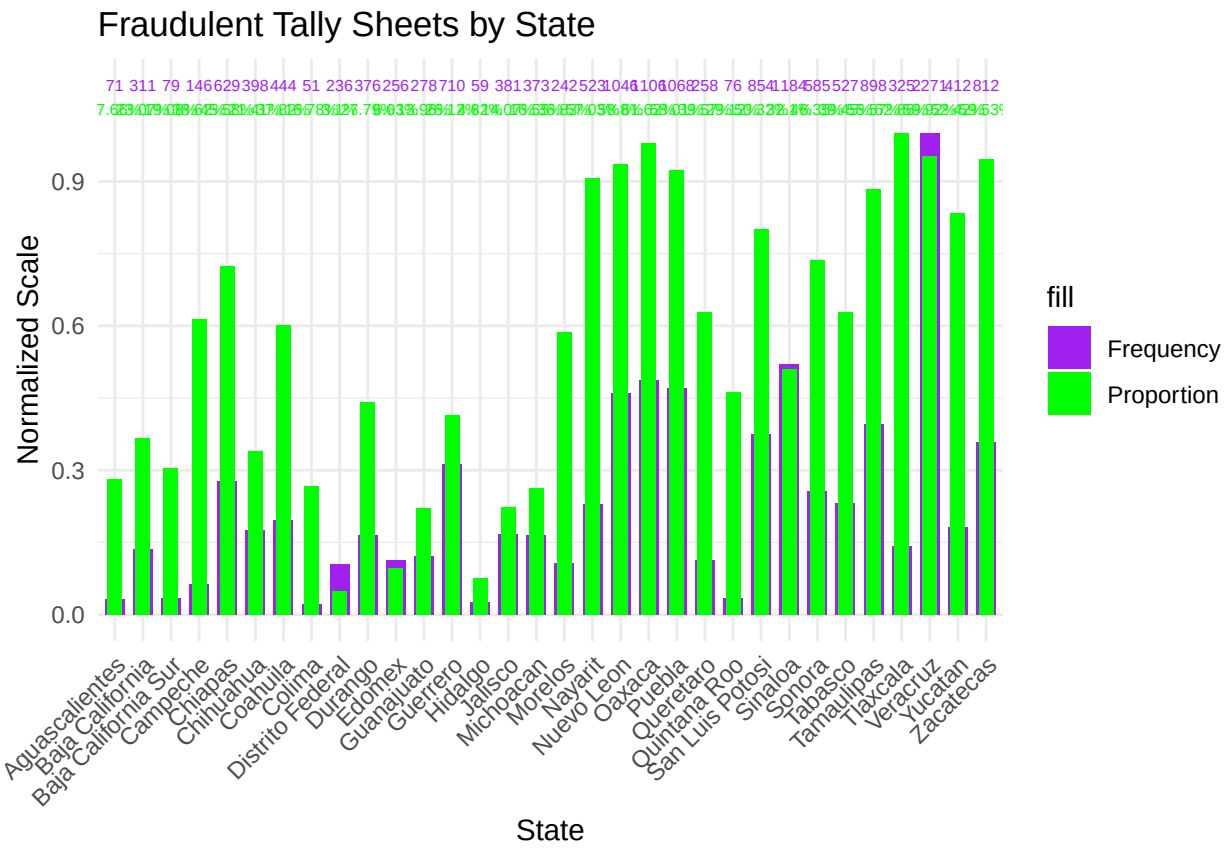
Feel free to try alternative approach(es) to make your visualization nicer and more informative.



Task 2.6. Visualize both proportions & frequencies of fraud by state

Create data visualization to show BOTH the *proportions* and *frequencies* of fraudulent tally sheets by state in one figure. Include annotations to highlight states with the highest level of fraud. Add informative labels to the figure. Describe the takeaways from the figure with a few sentences.

```
# YOUR CODE HERE

# Combined bar plot
combined_plot <- ggplot(state_level_indicators_summary, aes(x = state)) +
  geom_bar(aes(y = count_fraudulent / max(count_fraudulent), fill = "Frequency"), stat = "identity", pos
  geom_bar(aes(y = proportion_fraudulent / max(proportion_fraudulent), fill = "Proportion"), stat = "ide
  scale_fill_manual(values = c("Frequency" = "purple", "Proportion" = "green")) +
  labs(title = "Fraudulent Tally Sheets by State",
       x = "State",
       y = "Normalized Scale") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))

# Annotate states with the highest fraud
combined_plot +
  annotate("text", x = state_level_indicators_summary$state, y = 1.1,
           label = state_level_indicators_summary$count_fraudulent, color = "purple", size = 2) +
  annotate("text", x = state_level_indicators_summary$state, y = 1.05,
           label = paste0(round(state_level_indicators_summary$proportion_fraudulent, 2), "%"), color =
```



```
#Purple bars: the raw count of fraudulent tallies.
#Green bars: the proportion of fraud relative to the maximum value for each metric.
#Emphasize: states with the highest fraud levels.
```

Task 3. Clean vote return data (3pt)

Your next task is to clean a different dataset from the researchers' replication dossier. Its path is `data/Mexican_Election_Fraud/dataverse/VoteReturns.csv`. This dataset contains information about vote returns recorded in every tally sheet. This dataset is essential for the replication of Figure 4 in the research article.

Task 3.1. Load vote return data

Load the dataset onto your R environment. Name this dataset `d_return`. Show summary statistics of this dataset and describe the takeaways using a few sentences.

```
# YOUR CODE
d_return<-read_csv("data/VoteReturns.csv")
summary(d_return)
```

```
##      foto              seccion            casilla              dtto
##  Length:53499       Length:53499       Length:53499       Length:53499
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##       dto            municipio            edo              entidad
##  Min.   :  1.000   Length:53499       Length:53499       Length:53499
##  1st Qu.:  3.000   Class :character   Class :character   Class :character
##  Median :  6.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :  8.704
##  3rd Qu.: 10.000
##  Max.   :341.000
##  NA's   :4
##      pagina            p1                 p2                p3
##  Min.   :   1   Min.   :     0.0   Min.   :    0.0   Min.   :   0.0
##  1st Qu.:  45   1st Qu.:   250.0   1st Qu.:   67.0   1st Qu.:  98.0
##  Median :  92   Median :   530.0   Median :  245.0   Median : 233.0
##  Mean   : 104   Mean   :   671.9   Mean   :  343.3   Mean   : 319.3
##  3rd Qu.: 146   3rd Qu.:   941.5   3rd Qu.:  482.0   3rd Qu.: 442.0
##  Max.   :2020   Max.   :364105.0   Max.   :48225.0   Max.   :9127.0
##  NA's   :39                                          NA's   :1
##        p4                p5                pan               pri
##  Min.   :    0.0   Min.   :   0.00   Min.   :   0.00   Min.   :   0.0
##  1st Qu.:   73.0   1st Qu.:   0.00   1st Qu.:   2.00   1st Qu.:  52.0
##  Median :  222.0   Median :  13.00   Median :  18.00   Median : 107.0
##  Mean   :  369.7   Mean   :  29.36   Mean   :  56.88   Mean   : 162.7
##  3rd Qu.:  464.0   3rd Qu.:  36.00   3rd Qu.:  72.00   3rd Qu.: 195.0
##  Max.   :21265.0   Max.   :6650.00   Max.   :4436.00   Max.   :6080.0
##
##       pps               psm               pms              pfcrn
##  Min.   :   0.00   Min.   :   0.000   Min.   :   0.00   Min.   :   0.00
##  1st Qu.:   0.00   1st Qu.:   0.000   1st Qu.:   0.00   1st Qu.:   0.00
##  Median :   9.00   Median :   1.000   Median :   2.00   Median :  11.00
##  Mean   :  35.04   Mean   :   3.637   Mean   :  12.19   Mean   :  34.17
```

```
##  3rd Qu.:  47.00   3rd Qu.:   3.000   3rd Qu.:  13.00   3rd Qu.:  45.00
##  Max.   :1056.00   Max.   :1802.000   Max.   :5511.00   Max.   :1011.00
##
##       prt              parm           noregis           nombrenore
##  Min.   :  0.000   Min.   :   0.00   Min.   :   0.0000   Length:53499
##  1st Qu.:  0.000   1st Qu.:   0.00   1st Qu.:   0.0000   Class :character
##  Median :  0.000   Median :   5.00   Median :   0.0000   Mode  :character
##  Mean   :  1.912   Mean   :  20.44   Mean   :   0.8175
##  3rd Qu.:  1.000   3rd Qu.:  23.00   3rd Qu.:   0.0000
##  Max.   :592.000   Max.   :1170.00   Max.   :1604.0000
##                                      NA's   :1
##      otros          otroscan             pan2              pri2
##  Min.   :   0.00   Length:53499       Min.   :   0.000   Min.   :   0.00
##  1st Qu.:   0.00   Class :character   1st Qu.:   0.000   1st Qu.:   0.00
##  Median :   0.00   Mode  :character   Median :   0.000   Median :   0.00
##  Mean   :   3.17                      Mean   :   1.475   Mean   :   3.94
##  3rd Qu.:   0.00                      3rd Qu.:   0.000   3rd Qu.:   0.00
##  Max.   :1734.00                      Max.   :1239.000   Max.   :2651.00
##  NA's   :4
##       pps2              psm2              pms2              pfcrn2
##  Min.   :  0.0000   Min.   :  0.000   Min.   :  0.0000   Min.   :   0.0000
##  1st Qu.:  0.0000   1st Qu.:  0.000   1st Qu.:  0.0000   1st Qu.:   0.0000
##  Median :  0.0000   Median :  0.000   Median :  0.0000   Median :   0.0000
##  Mean   :  0.7557   Mean   :  0.116   Mean   :  0.3039   Mean   :   0.7968
##  3rd Qu.:  0.0000   3rd Qu.:  0.000   3rd Qu.:  0.0000   3rd Qu.:   0.0000
##  Max.   :680.0000   Max.   :429.000   Max.   :427.0000   Max.   :1319.0000
##
##       prt2             parm2             noregis2            otro2
##  Min.   :  0.000   Min.   :  0.0000   Min.   :  0.00000   Min.   : 0.000000
##  1st Qu.:  0.000   1st Qu.:  0.0000   1st Qu.:  0.00000   1st Qu.: 0.000000
##  Median :  0.000   Median :  0.0000   Median :  0.00000   Median : 0.000000
##  Mean   :  0.073   Mean   :  0.5122   Mean   :  0.01837   Mean   : 0.002935
##  3rd Qu.:  0.000   3rd Qu.:  0.0000   3rd Qu.:  0.00000   3rd Qu.: 0.000000
##  Max.   :429.000   Max.   :429.0000   Max.   :259.00000   Max.   :26.000000
##
##       pan3              pri3             pps3              psm3
##  Min.   :   0.00   Min.   :   0.0   Min.   :  0.00   Min.   :  0.000
##  1st Qu.:   0.00   1st Qu.:   0.0   1st Qu.:  0.00   1st Qu.:  0.000
##  Median :   0.00   Median :  32.0   Median :  0.00   Median :  0.000
##  Mean   :  39.36   Mean   :  93.5   Mean   : 22.08   Mean   :  2.094
##  3rd Qu.:  45.00   3rd Qu.: 127.0   3rd Qu.: 21.00   3rd Qu.:  1.000
##  Max.   :2194.00   Max.   :6080.0   Max.   :921.00   Max.   :856.000
##                    NA's   :1                         NA's   :2
##       pms3              pfcrn3            prt3              parm3
##  Min.   :   0.000   Min.   :  0.00   Min.   :  0.000   Min.   :   0.00
##  1st Qu.:   0.000   1st Qu.:  0.00   1st Qu.:  0.000   1st Qu.:   0.00
##  Median :   0.000   Median :  0.00   Median :  0.000   Median :   0.00
##  Mean   :   7.803   Mean   : 21.63   Mean   :  1.077   Mean   :  12.68
##  3rd Qu.:   5.000   3rd Qu.: 23.00   3rd Qu.:  1.000   3rd Qu.:  11.00
##  Max.   :8932.000   Max.   :992.00   Max.   :413.000   Max.   :1170.00
##  NA's   :1          NA's   :1
##     noregis3           otro3               suma             nulos
##  Min.   :  0.0000   Min.   :   0.0000   Min.   :   0.0   Min.   :   0.00
##  1st Qu.:  0.0000   1st Qu.:   0.0000   1st Qu.:  82.0   1st Qu.:   0.00
```

```
##  Median :  0.0000   Median :   0.0000   Median : 217.0   Median :   3.00
##  Mean   :  0.3498   Mean   :   0.3016   Mean   : 296.4   Mean   :  21.93
##  3rd Qu.:  0.0000   3rd Qu.:   0.0000   3rd Qu.: 420.0   3rd Qu.:  11.00
##  Max.   :747.0000   Max.   :1353.0000   Max.   :9962.0   Max.   :8770.00
##                     NA's   :1           NA's   :1        NA's   :1
##      total             suma1              nulos1            total1
##  Min.   :    0.0   Min.   :   0.000   Min.   :   0.000   Min.   :   0.000
##  1st Qu.:   90.0   1st Qu.:   0.000   1st Qu.:   0.000   1st Qu.:   0.000
##  Median :  229.0   Median :   0.000   Median :   0.000   Median :   0.000
##  Mean   :  315.7   Mean   :   4.865   Mean   :   0.635   Mean   :   7.175
##  3rd Qu.:  440.0   3rd Qu.:   0.000   3rd Qu.:   0.000   3rd Qu.:   0.000
##  Max.   :16811.0   Max.   :3333.000   Max.   :1600.000   Max.   :2787.000
##  NA's   :1         NA's   :2          NA's   :2          NA's   :2
##      suma2             nulos2             total2         inciden
##  Min.   :   0.0   Min.   :   0.00   Min.   :   0.0   Length:53499
##  1st Qu.:   0.0   1st Qu.:   0.00   1st Qu.:   0.0   Class :character
##  Median :   0.0   Median :   0.00   Median :   0.0   Mode  :character
##  Mean   : 176.9   Mean   :  11.38   Mean   : 192.6
##  3rd Qu.: 280.0   3rd Qu.:   5.00   3rd Qu.: 299.0
##  Max.   :7633.0   Max.   :7734.00   Max.   :9855.0
##  NA's   :2        NA's   :2         NA's   :2
##  representante_pan  representante_pri  representante_pps  representante_pms
##  Length:53499       Length:53499       Length:53499       Length:53499
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  representante_psm  representante_pfcrn representante_prt  representante_parm
##  Length:53499       Length:53499        Length:53499       Length:53499
##  Class :character   Class :character    Class :character   Class :character
##  Mode  :character   Mode  :character    Mode  :character   Mode  :character
##
##
##
##
##  protesta_pan       protesta_pri       protesta_pps       protesta_pms
##  Length:53499       Length:53499       Length:53499       Length:53499
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  protesta_psm       protesta_pfcrn     protesta_prt       protesta_parm
##  Length:53499       Length:53499       Length:53499       Length:53499
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  protesta_otro       presidente         secretario           primer
```

```
##  Length:53499       Length:53499       Length:53499       Length:53499
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    segundo            observa              var79           salinas
##  Length:53499       Length:53499       Min.   :   1.0   Min.   :   0.0
##  Class :character   Class :character   1st Qu.:   1.0   1st Qu.:  63.0
##  Mode  :character   Mode  :character   Median :   1.0   Median : 115.0
##                                        Mean   : 131.2   Mean   : 174.4
##                                        3rd Qu.:   2.0   3rd Qu.: 206.0
##                                        Max.   :9999.0   Max.   :6080.0
##                                        NA's   :53422
##    clouthier           ibarra           castillo        ppsccs
##  Min.   :   0.00   Min.   :  0.000   Min.   :   0   Min.   :   0.00
##  1st Qu.:   3.00   1st Qu.:  0.000   1st Qu.:   0   1st Qu.:   1.00
##  Median :  23.00   Median :  0.000   Median :   1   Median :  12.00
##  Mean   :  61.37   Mean   :  2.185   Mean   :   4   Mean   :  37.67
##  3rd Qu.:  78.00   3rd Qu.:  2.000   3rd Qu.:   3   3rd Qu.:  51.00
##  Max.   :4436.00   Max.   :592.000   Max.   :1802   Max.   :1056.00
##
##     pfcrnccs          parmccs            nrccs            noregccs
##  Min.   :   0.00   Min.   :   0.00   Min.   :0.000000   Min.   :   0.0000
##  1st Qu.:   1.00   1st Qu.:   0.00   1st Qu.:0.000000   1st Qu.:   0.0000
##  Median :  14.00   Median :   6.00   Median :0.000000   Median :   0.0000
##  Mean   :  36.85   Mean   :  21.98   Mean   :0.006654   Mean   :   0.1439
##  3rd Qu.:  48.00   3rd Qu.:  25.00   3rd Qu.:0.000000   3rd Qu.:   0.0000
##  Max.   :1319.00   Max.   :1170.00   Max.   :1.000000   Max.   :1125.0000
##
##       occs           otrosccs           cardenas
##  Min.   :0.0000   Min.   :   0.000   Min.   :   0.00
##  1st Qu.:1.0000   1st Qu.:   0.000   1st Qu.:  10.00
##  Median :1.0000   Median :   0.000   Median :  53.00
##  Mean   :0.9942   Mean   :   3.106   Mean   :  99.75
##  3rd Qu.:1.0000   3rd Qu.:   0.000   3rd Qu.: 141.00
##  Max.   :1.0000   Max.   :1734.000   Max.   :2280.00
##
```

```
#The dataset has a total of 53,499 entries
#The columns 'p1' to 'pfcrn3' represent the votes for different political parties and candidates.
#Some columns, such as 'ppsm2', 'ppm2', and 'presidente', have missing values
```

Note 2. What are in this dataset?

This table contains a lot of different variables. The researcher offers no comprehensive documentation to tell us what every column means. For the sake of this problem set, you only need to know the meanings of the following columns:

- `foto` is an identifier of the images of tally sheets in this dataset. We will need it to merge this dataset with the `d_tally` data.
- `edo` contains the names of states.
- `dto` contains the names of districts (in Arabic numbers).
- `salinas`, `clouthier`, and `ibarra` contain the counts of votes (as recorded in the tally sheets) for presidential candidates Salinas (PRI), Cardenas (FDN), and Clouthier (PAN). In addition, the summation of all three makes the total number of **presidential votes**.
- `total` contains the total number of **legislative votes**.

Task 3.2. Recode names of states

A state whose name is `Chihuahua` is mislabelled as `Chihuhua`. A state whose name is currently `Edomex` needs to be recoded to `Estado de Mexico`. Please re-code the names of these two states accordingly.

```
# YOUR CODE

d_return <- d_return |>
  mutate(edo = case_when(
    edo == "Chihuhua" ~ "Chihuahua",
    edo == "Edomex" ~ "Estado de Mexico",
    TRUE ~ edo
  ))
print(d_return)
```

```
## # A tibble: 53,499 x 91
##    foto   seccion casilla dtto    dto municipio edo   entidad pagina    p1    p2
##    <chr>  <chr>   <chr>   <chr> <dbl> <chr>     <chr> <chr>    <dbl> <dbl> <dbl>
##  1 2014-~ 83      83      I         1 AGUASCAL~ Agua~ AGS        127   108   333
##  2 2014-~ 1       84      <NA>      1 AGUASCAL~ Agua~ AGUASC~    128   919   453
##  3 2014-~ 85      85      1         1 AGUASCAL~ Agua~ AGUASC~    129   795   264
##  4 2014-~ 45      45-A    1         1 AGUASCAL~ Agua~ AGUA       130   767   450
##  5 2014-~ 86      86      1         1 AGUASCAL~ Agua~ AGUAS      131  1243   578
##  6 2014-~ 87      87      1         1 <NA>      Agua~ 1          132   718   333
##  7 2014-~ 1       87-A    7         1 AGUASCAL~ Agua~ AGUAS      133   710   299
##  8 2014-~ 88      88      1         1 AGUAS     Agua~ AGUAS      134     0     0
##  9 2014-~ 89      89      1         1 AGUASCAL~ Agua~ AGUAS      135   764     8
## 10 2014-~ 89      89-A    7         1 AGUSCALI~ Agua~ 1          136   759   256
## # i 53,489 more rows
## # i 80 more variables: p3 <dbl>, p4 <dbl>, p5 <dbl>, pan <dbl>, pri <dbl>,
## #   pps <dbl>, psm <dbl>, pms <dbl>, pfcrn <dbl>, prt <dbl>, parm <dbl>,
## #   noregis <dbl>, nombrenore <chr>, otros <dbl>, otroscan <chr>, pan2 <dbl>,
## #   pri2 <dbl>, pps2 <dbl>, psm2 <dbl>, pms2 <dbl>, pfcrn2 <dbl>, prt2 <dbl>,
## #   parm2 <dbl>, noregis2 <dbl>, otro2 <dbl>, pan3 <dbl>, pri3 <dbl>,
## #   pps3 <dbl>, psm3 <dbl>, pms3 <dbl>, pfcrn3 <dbl>, prt3 <dbl>, ...
```

Task 3.3. Recode districts' identifiers

Compare how districts' identifiers are recorded differently in the tally (`d_tally`) from vote return (`d_return`) datasets. Specifically, in the `d_tally` dataset, `district` contains Roman numbers while in the `d_return` dataset, `dto` contains Arabic numbers. Recode districts' identifiers in the `d_return` dataset to match those in the `d_tally` dataset. To complete this task, first summarize the values of the two district identifier columns in the two datasets respectively to verify the above claim. Then do the requested conversion.

```
# YOUR CODE HERE

# Function to convert Arabic numbers to Roman numbers
arabic_to_roman <- function(number) {
  if (is.na(number)) {
    return(NA)
  }

  roman_data <- data.frame(
    arabic = c(1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1),
    roman = c("M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I")
  )

  result <- ""
  for (i in 1:nrow(roman_data)) {
    while (number >= roman_data$arabic[i]) {
      result <- paste0(result, roman_data$roman[i])
      number <- number - roman_data$arabic[i]
    }
  }

  return(result)
}


# Recode dto column and rename it to dto
d_return <- d_return |>
  mutate(dto = map_chr(dto, arabic_to_roman))

head(d_return)
```

```
## # A tibble: 6 x 91
##   foto    seccion casilla dtto  dto   municipio edo   entidad pagina    p1    p2
##   <chr>   <chr>   <chr>   <chr> <chr> <chr>     <chr> <chr>    <dbl> <dbl> <dbl>
## 1 2014-0~ 83      83      I     I     AGUASCAL~ Agua~ AGS        127   108   333
## 2 2014-0~ 1       84      <NA>  I     AGUASCAL~ Agua~ AGUASC~    128   919   453
## 3 2014-0~ 85      85      1     I     AGUASCAL~ Agua~ AGUASC~    129   795   264
## 4 2014-0~ 45      45-A    1     I     AGUASCAL~ Agua~ AGUA       130   767   450
## 5 2014-0~ 86      86      1     I     AGUASCAL~ Agua~ AGUAS      131  1243   578
## 6 2014-0~ 87      87      1     I     <NA>      Agua~ 1          132   718   333
## # i 80 more variables: p3 <dbl>, p4 <dbl>, p5 <dbl>, pan <dbl>, pri <dbl>,
## #   pps <dbl>, psm <dbl>, pms <dbl>, pfcrn <dbl>, prt <dbl>, parm <dbl>,
## #   noregis <dbl>, nombrenore <chr>, otros <dbl>, otroscan <chr>, pan2 <dbl>,
## #   pri2 <dbl>, pps2 <dbl>, psm2 <dbl>, pms2 <dbl>, pfcrn2 <dbl>, prt2 <dbl>,
## #   parm2 <dbl>, noregis2 <dbl>, otro2 <dbl>, pan3 <dbl>, pri3 <dbl>,
## #   pps3 <dbl>, psm3 <dbl>, pms3 <dbl>, pfcrn3 <dbl>, prt3 <dbl>, parm3 <dbl>,
```

```
## #   noregis3 <dbl>, otro3 <dbl>, suma <dbl>, nulos <dbl>, total <dbl>, ...
```

Task 3.4. Create a `name_image` identifier for the `d_return` dataset

In the `d_return` dataset, create a column named `name_image` as the first column. The column concatenate values in the three columns: `edo`, `dto`, and `foto` with an underscore `_` as separators.

```
# YOUR CODE HERE

# Create name_image column for d_return
d_return <- d_return |>
  mutate(name_image = paste(edo, dto, foto, sep = "_"))
```

Task 3.5. Wrangle the `name_image` column in two datasets

As a final step before merging `d_return` and `d_tally`, you are required to perform the following data wrangling. For the `name_image` column in BOTH `d_return` and `d_tally`:

- Convert all characters to lower case.
- Remove ending substring `.jpg`.

```
# YOUR CODE HERE

# Convert name_image column to lower case and remove ".jpg" ending substring
d_return <- d_return |>
  mutate(name_image = tolower(name_image),
         name_image = str_replace(name_image, "\\.jpg$", ""))

d_tally <- d_tally |>
  mutate(name_image = tolower(name_image),
         name_image = str_replace(name_image, "\\.jpg$", ""))
```

Task 3.6 Join classification results and vote returns

After you have successfully completed all the previous steps, join `d_return` and `d_tally` by column `name_image`. This task contains two part. First, use appropriate `tidyverse` functions to answer the following questions:

- How many rows are in `d_return` but not in `d_tally`? Which states and districts are they from?
- How many rows are in `d_tally` but not in `d_return`? Which states and districts are they from?

```
# YOUR CODE HERE

# Join d_return and d_tally by name_image
joined_data <- left_join(d_return, d_tally, by = "name_image")

# Identify rows in d_return but not in d_tally
rows_in_return_not_in_tally <- anti_join(d_return, d_tally, by = "name_image")

# Identify rows in d_tally but not in d_return
rows_in_tally_not_in_return <- anti_join(d_tally, d_return, by = "name_image")

# Count the number of rows in each set
num_rows_in_return_not_in_tally <- nrow(rows_in_return_not_in_tally)
num_rows_in_tally_not_in_return <- nrow(rows_in_tally_not_in_return)

# Display the results
cat("Number of rows in d_return but not in d_tally:", num_rows_in_return_not_in_tally, "\n")
```

```
## Number of rows in d_return but not in d_tally: 211
```

```
cat("Number of rows in d_tally but not in d_return:", num_rows_in_tally_not_in_return, "\n")
```

```
## Number of rows in d_tally but not in d_return: 2368
```

```
# States and districts in d_return but not in d_tally
cat("States and districts in d_return but not in d_tally:\n")
```

```
## States and districts in d_return but not in d_tally:
```

```
print(rows_in_return_not_in_tally)
```

```
## # A tibble: 211 x 92
##    foto   seccion casilla dtto  dto   municipio edo   entidad pagina    p1    p2
##    <chr>  <chr>   <chr>   <chr> <chr> <chr>     <chr> <chr>    <dbl> <dbl> <dbl>
##  1 2014-~ 83      83      I     I     AGUASCAL~ Agua~ AGS        127   108   333
##  2 2014-~ 1       84      <NA>  I     AGUASCAL~ Agua~ AGUASC~    128   919   453
##  3 DSC_0~ 70      70- C   1     I     AGUASCAL~ Agua~ AGUASC~    103     0     0
##  4 DSC_0~ 67      67      5     V     AGUASCAL~ Agua~ AGUASC~     94   606   231
##  5 DSC_0~ 63      <NA>    6     VI    <NA>      Agua~ 2           88   710   369
##  6 20140~ <NA>    <NA>    0     II    <NA>      Baja~ 0            2     0     0
##  7 DSC_0~ <NA>    1-A     1     I     CHAMPOTON Camp~ CAMPEC~    132   448   238
```

```
##  8 20140~ 7A      51A     1     I     BERRIOZA~ Chia~ CHIAPAS     12   173     0
##  9 DSC_0~ 37      37C     1     I     TUXTLA    Chia~ CHIAPAS    224     0     0
## 10 20140~ 1       1       2     II    SAN CRIS~ Chia~ CHIAPAS     52   165   102
## # i 201 more rows
## # i 81 more variables: p3 <dbl>, p4 <dbl>, p5 <dbl>, pan <dbl>, pri <dbl>,
## #   pps <dbl>, psm <dbl>, pms <dbl>, pfcrn <dbl>, prt <dbl>, parm <dbl>,
## #   noregis <dbl>, nombrenore <chr>, otros <dbl>, otroscan <chr>, pan2 <dbl>,
## #   pri2 <dbl>, pps2 <dbl>, psm2 <dbl>, pms2 <dbl>, pfcrn2 <dbl>, prt2 <dbl>,
## #   parm2 <dbl>, noregis2 <dbl>, otro2 <dbl>, pan3 <dbl>, pri3 <dbl>,
## #   pps3 <dbl>, psm3 <dbl>, pms3 <dbl>, pfcrn3 <dbl>, prt3 <dbl>, ...
```

```
# States and districts in d_tally but not in d_return
cat("States and districts in d_tally but not in d_return:\n")
```

```
## States and districts in d_tally but not in d_return:
```

```
print(rows_in_tally_not_in_return)
```

```
## # A tibble: 2,368 x 5
##    name_image                              state    district fraud_proba fraud_bin
##    <chr>                                   <chr>    <chr>          <dbl> <lgl>
##  1 aguascalientes_i_dsc_0064               Aguasca~ I              0      FALSE
##  2 aguascalientes_i_dsc_0090               Aguasca~ I              0.0028 FALSE
##  3 aguascalientes_i_dsc_0096               Aguasca~ I              0.002  FALSE
##  4 aguascalientes_i_img_7897               Aguasca~ I              0.999  TRUE
##  5 aguascalientes_ii_2014-05-26 00.17.24   Aguasca~ II             0.156  FALSE
##  6 aguascalientes_ii_2014-05-26 00.24.27   Aguasca~ II             0.551  FALSE
##  7 baja california sur_i_img_6671          Baja Ca~ I              0.124  FALSE
##  8 baja california sur_i_img_6716          Baja Ca~ I              0      FALSE
##  9 baja california sur_i_img_6717          Baja Ca~ I              0.482  FALSE
## 10 baja california sur_i_img_6718          Baja Ca~ I              0.731  TRUE
## # i 2,358 more rows
```

Second, create a dataset call `d` by joining `d_return` and `d_tally` by column `name_image`. `d` contains rows whose identifiers appear in *both* datasets and columns from *both* datasets.

```
# YOUR CODE HERE

# Join d_return and d_tally by name_image
d <- inner_join(d_return, d_tally, by = "name_image")

# Display the structure of d
str(d)
```

```
## tibble [53,288 x 96] (S3: tbl_df/tbl/data.frame)
##  $ foto              : chr [1:53288] "2014-05-26 00.00.17" "2014-05-26 00.00.25" "2014-05-26 00.00.3
##  $ seccion           : chr [1:53288] "85" "45" "86" "87" ...
##  $ casilla           : chr [1:53288] "85" "45-A" "86" "87" ...
##  $ dtto              : chr [1:53288] "1" "1" "1" "1" ...
##  $ dto               : chr [1:53288] "I" "I" "I" "I" ...
##  $ municipio         : chr [1:53288] "AGUASCALIENTES" "AGUASCALIENTES" "AGUASCALIENTES" NA ...
##  $ edo               : chr [1:53288] "Aguascalientes" "Aguascalientes" "Aguascalientes" "Aguascalie
```

```
##  $ entidad           : chr [1:53288] "AGUASCALIE" "AGUA" "AGUAS" "1" ...
##  $ pagina            : num [1:53288] 129 130 131 132 133 134 135 136 137 138 ...
##  $ p1                : num [1:53288] 795 767 1243 718 710 ...
##  $ p2                : num [1:53288] 264 450 578 333 299 0 8 256 347 551 ...
##  $ p3                : num [1:53288] 545 316 666 384 411 0 757 238 381 243 ...
##  $ p4                : num [1:53288] 483 316 614 349 411 0 694 220 368 234 ...
##  $ p5                : num [1:53288] 61 0 60 35 31 0 60 14 30 27 ...
##  $ pan               : num [1:53288] 306 192 432 181 0 0 429 86 188 59 ...
##  $ pri               : num [1:53288] 165 88 173 145 0 0 216 110 140 140 ...
##  $ pps               : num [1:53288] 23 10 19 15 0 0 35 6 19 18 ...
##  $ psm               : num [1:53288] 11 1 2 6 0 0 9 2 5 6 ...
##  $ pms               : num [1:53288] 12 1 4 4 0 0 13 5 5 7 ...
##  $ pfcrn             : num [1:53288] 8 8 10 12 0 0 21 6 10 9 ...
##  $ prt               : num [1:53288] 2 1 1 1 0 0 3 1 3 0 ...
##  $ parm              : num [1:53288] 5 10 14 7 0 0 6 0 3 4 ...
##  $ noregis           : num [1:53288] 0 0 0 0 0 0 0 1 1 0 ...
##  $ nombrenore        : chr [1:53288] NA NA NA NA ...
##  $ otros             : num [1:53288] 0 0 0 0 0 0 6 0 2 0 ...
##  $ otroscan          : chr [1:53288] NA NA NA NA ...
##  $ pan2              : num [1:53288] 0 0 0 0 0 324 0 5 12 0 ...
##  $ pri2              : num [1:53288] 0 0 0 0 0 347 0 7 10 0 ...
##  $ pps2              : num [1:53288] 0 0 0 0 0 56 0 3 2 0 ...
##  $ psm2              : num [1:53288] 0 0 0 0 0 13 0 0 0 0 ...
##  $ pms2              : num [1:53288] 0 0 0 0 0 31 0 1 2 0 ...
##  $ pfcrn2            : num [1:53288] 0 0 0 0 0 41 0 0 0 0 ...
##  $ prt2              : num [1:53288] 0 0 0 0 0 1 0 0 0 0 ...
##  $ parm2             : num [1:53288] 0 0 0 0 0 21 0 0 1 0 ...
##  $ noregis2          : num [1:53288] 0 0 0 0 0 0 0 0 0 0 ...
##  $ otro2             : num [1:53288] 0 0 0 0 0 0 0 0 0 0 ...
##  $ pan3              : num [1:53288] 306 192 0 181 170 0 429 91 200 59 ...
##  $ pri3              : num [1:53288] 165 88 0 145 170 0 216 117 150 140 ...
##  $ pps3              : num [1:53288] 23 10 0 15 21 0 35 9 21 18 ...
##  $ psm3              : num [1:53288] 11 1 0 6 4 0 9 2 8 6 ...
##  $ pms3              : num [1:53288] 12 1 0 4 15 0 13 6 7 7 ...
##  $ pfcrn3            : num [1:53288] 8 8 0 12 14 0 21 6 10 9 ...
##  $ prt3              : num [1:53288] 2 1 0 1 1 0 3 1 3 0 ...
##  $ parm3             : num [1:53288] 5 10 0 7 7 0 6 0 5 4 ...
##  $ noregis3          : num [1:53288] 0 0 0 0 0 0 0 0 0 0 ...
##  $ otro3             : num [1:53288] 0 0 0 0 0 0 0 0 0 0 ...
##  $ suma              : num [1:53288] 532 311 655 371 0 0 738 216 383 243 ...
##  $ nulos             : num [1:53288] 13 5 11 13 0 0 19 7 6 551 ...
##  $ total             : num [1:53288] 545 316 666 184 0 0 765 223 177 794 ...
##  $ suma1             : num [1:53288] 0 0 0 0 0 457 0 16 25 0 ...
##  $ nulos1            : num [1:53288] 0 0 0 0 0 23 0 0 2 0 ...
##  $ total1            : num [1:53288] 0 0 0 0 0 510 0 16 23 0 ...
##  $ suma2             : num [1:53288] 532 311 0 371 402 0 738 232 408 243 ...
##  $ nulos2            : num [1:53288] 13 5 0 13 9 0 19 0 8 551 ...
##  $ total2            : num [1:53288] 545 316 0 184 411 0 765 232 400 794 ...
##  $ inciden           : chr [1:53288] "NINGUNO" NA NA "NINGUNO" ...
##  $ representante_pan : chr [1:53288] "Si" "Si" "Si" "Si" ...
##  $ representante_pri : chr [1:53288] "No" "Si" "Si" "Si" ...
##  $ representante_pps : chr [1:53288] "No" "Si" "No" "Si" ...
##  $ representante_pms : chr [1:53288] "No" "No" "No" "No" ...
##  $ representante_psm : chr [1:53288] "No" "Si" "No" "Si" ...
```

```
##  $ representante_pfcrn: chr [1:53288] "No" "No" "Si" "Si" ...
##  $ representante_prt  : chr [1:53288] "No" "No" "No" "No" ...
##  $ representante_parm : chr [1:53288] "No" "No" "No" "No" ...
##  $ protesta_pan       : chr [1:53288] "No" "No" "No" "No" ...
##  $ protesta_pri       : chr [1:53288] "No" "No" "No" "No" ...
##  $ protesta_pps       : chr [1:53288] "No" "No" "No" "No" ...
##  $ protesta_pms       : chr [1:53288] "No" "No" "No" "No" ...
##  $ protesta_psm       : chr [1:53288] "No" "No" "No" "No" ...
##  $ protesta_pfcrn     : chr [1:53288] "No" "No" "No" "No" ...
##  $ protesta_prt       : chr [1:53288] "No" "No" "No" "No" ...
##  $ protesta_parm      : chr [1:53288] "No" "No" "No" "No" ...
##  $ protesta_otro      : chr [1:53288] "No" "No" "No" "No" ...
##  $ presidente         : chr [1:53288] "Si" "Si" "Si" "Si" ...
##  $ secretario         : chr [1:53288] "Si" "Si" "Si" "Si" ...
##  $ primer             : chr [1:53288] "Si" "Si" "Si" "Si" ...
##  $ segundo            : chr [1:53288] "No" "Si" "Si" "Si" ...
##  $ observa            : chr [1:53288] "1" NA NA NA ...
##  $ var79              : num [1:53288] NA NA NA NA NA NA NA NA NA NA ...
##  $ salinas            : num [1:53288] 165 88 173 145 170 347 216 117 150 140 ...
##  $ clouthier          : num [1:53288] 306 192 432 181 170 324 429 91 200 59 ...
##  $ ibarra             : num [1:53288] 2 1 1 1 1 1 3 1 3 0 ...
##  $ castillo           : num [1:53288] 11 1 2 6 4 13 9 2 8 6 ...
##  $ ppsccs             : num [1:53288] 23 10 19 15 21 56 35 9 21 18 ...
##  $ pfcrnccs           : num [1:53288] 8 8 10 12 14 41 21 6 10 9 ...
##  $ parmccs            : num [1:53288] 5 10 14 7 7 21 6 0 5 4 ...
##  $ nrccs              : num [1:53288] 0 0 0 0 0 0 0 0 0 0 ...
##  $ noregccs           : num [1:53288] 0 0 0 0 0 0 0 0 0 0 ...
##  $ occs               : num [1:53288] 1 1 1 1 1 1 1 1 1 1 ...
##  $ otrosccs           : num [1:53288] 0 0 0 0 0 0 6 0 2 0 ...
##  $ cardenas           : num [1:53288] 36 28 43 34 42 118 68 15 38 31 ...
##  $ name_image         : chr [1:53288] "aguascalientes_i_2014-05-26 00.00.17" "aguascalientes_i_2014-0
##  $ state              : chr [1:53288] "Aguascalientes" "Aguascalientes" "Aguascalientes" "Aguascalien
##  $ district           : chr [1:53288] "I" "I" "I" "I" ...
##  $ fraud_proba        : num [1:53288] 0.0428 0.4231 0.0349 0.1302 0.2117 ...
##  $ fraud_bin          : logi [1:53288] FALSE FALSE FALSE FALSE FALSE FALSE ...
```

Task 4. Visualize distributions of fraudulent tallies across candidates (6pt)

In this task, you will visualize the distributions of fraudulent tally sheets across three presidential candidates: **Sarinas (PRI)**, **Cardenas (FDN)**, and **Clouthier (PAN)**. The desired output of is reproducing and extending Figure 4 in the research article (Cantu 2019, pp. 720).

Task 4.1. Calculate vote proportions of Salinas, Clouthier, and Cardenas

Before getting to the visualization, you should first calculate the proportion of votes (among all) received by the three candidates of interest. As additional background information, there are two more presidential candidates in this election, whose votes received are recorded in `ibarra` and `castillo` respectively. Please perform the tasks in the following two steps on the `d` dataset:

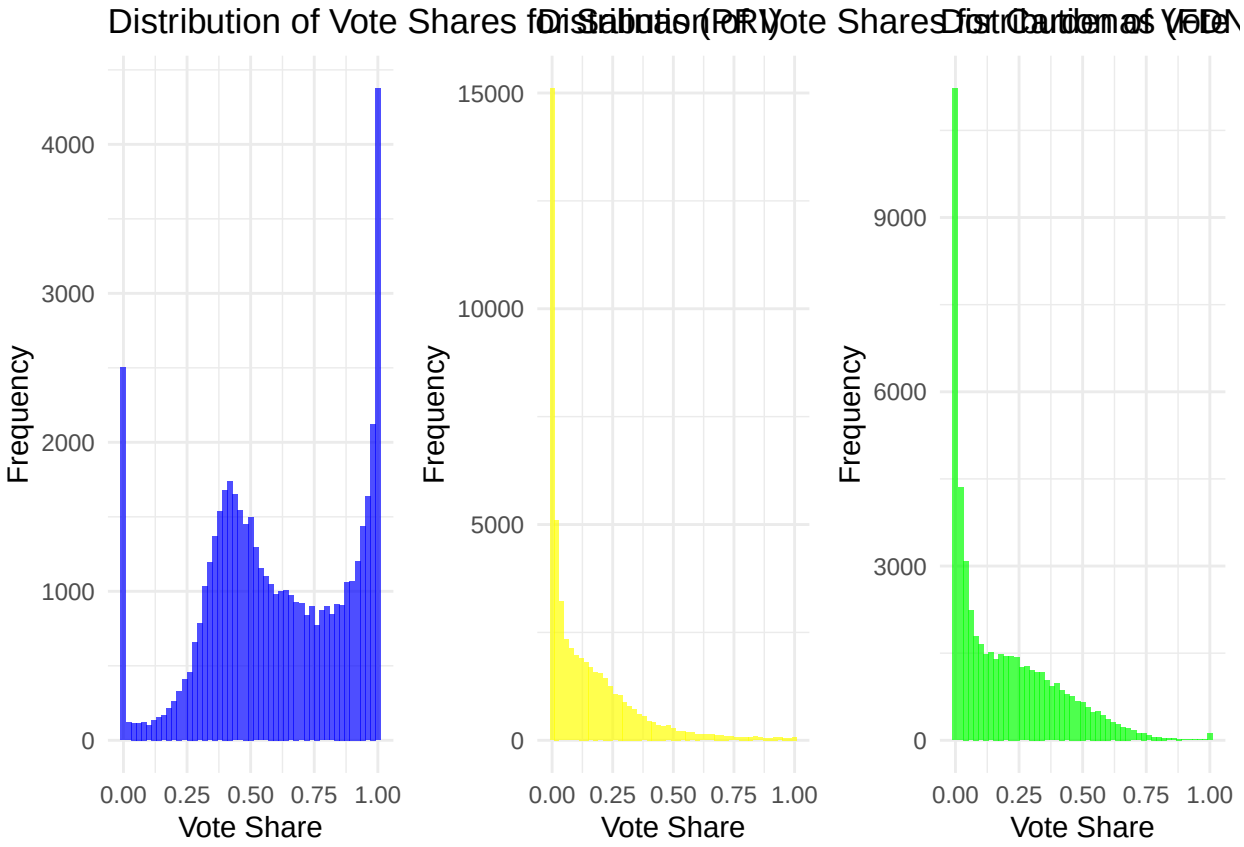
- Create a new column named `total_president` as an indicator of the total number of votes of the 5 presidential candidates.
- Create three columns `salinas_prop`, `cardenas_prop`, and `clouthier_prop` that indicate the proportions of the votes these three candidates receive respectively.

```
# YOUR CODE HERE

# Step 1: Create a new column named total_president
d$total_president <- rowSums(d[, c("pri", "pfcrn", "pan", "ibarra", "castillo")], na.rm = TRUE)

# Step 2: Create three columns salinas_prop, cardenas_prop, and clouthier_prop
d$salinas_prop <- d$pri / d$total_president
d$cardenas_prop <- d$pfcrn / d$total_president
d$clouthier_prop <- d$pan / d$total_president
```

Task 4.2. Replicate Figure 4

Based on all the previous step, reproduce Figure 4 in Cantu (2019, pp. 720).

```
# YOUR CODE HERE

# histogram Salinas
salinas_plot <- ggplot(d, aes(x = salinas_prop)) +
  geom_histogram(binwidth = 0.02, fill = "blue", alpha = 0.7) +
  labs(title = "Distribution of Vote Shares for Salinas (PRI)",
       x = "Vote Share",
       y = "Frequency") +
  theme_minimal()

# histogram  Cardenas
cardenas_plot <- ggplot(d, aes(x = cardenas_prop)) +
  geom_histogram(binwidth = 0.02, fill = "yellow", alpha = 0.7) +
  labs(title = "Distribution of Vote Shares for Cardenas (FDN)",
       x = "Vote Share",
       y = "Frequency") +
  theme_minimal()

# histogram Clouthier
clouthier_plot <- ggplot(d, aes(x = clouthier_prop)) +
  geom_histogram(binwidth = 0.02, fill = "green", alpha = 0.7) +
  labs(title = "Distribution of Vote Shares for Clouthier (PAN)",
       x = "Vote Share",
       y = "Frequency") +
  theme_minimal()

multiplot <- cowplot::plot_grid(salinas_plot, cardenas_plot, clouthier_plot, ncol = 3)

print(multiplot)
```



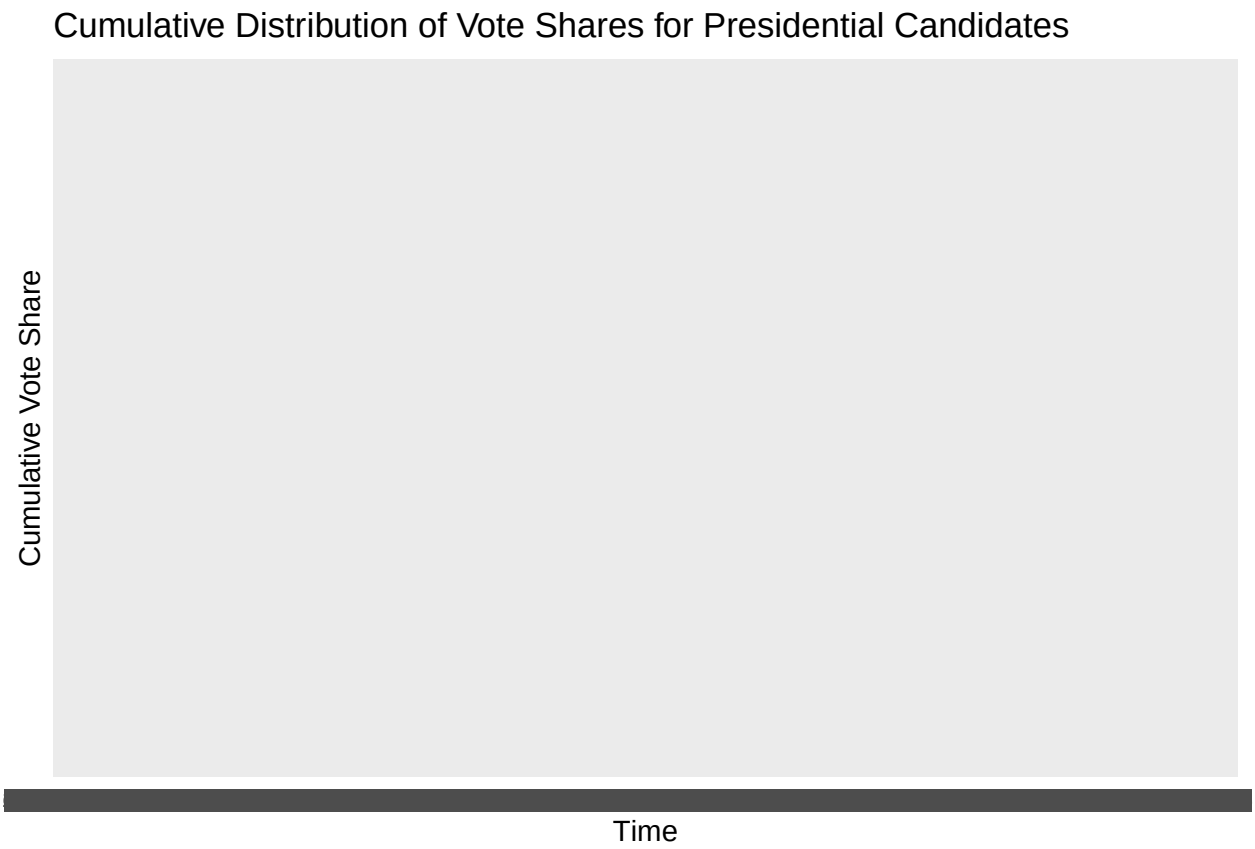
Note: Your performance in this task will be mainly evaluated based on your output's similarity with the original figure. Pay attention to the details. For your reference, below is a version created by the instructor.

Task 4.3. Discuss and extend the reproduced figure

Referring to your reproduced figures and the research articles, in what way is the researcher's argument supported by this figure? Make an alternative visualization design that can substantiate and even augment the current argument. After you have shown your alternative design, in a few sentences, describe how your design provides visual aid as effectively as or more effectively than the original figure.

Note: Feel free to make *multiple* alternative designs to earn bonus credits. However, please be selective. Only a design with major differences from the existing ones can be counted as an alternative design.

```
# YOUR CODE HERE

#The initial reproduction: histograms illustrating the distribution of vote shares for each candidate.
#Histograms provide a visual representation of the spread and concentration of vote shares for each can

#Alternative Design: Stacked Area Chart
#A stacked area chart can illustrate the cumulative distribution of vote shares over time, providing a
#A stacked area chart can be effective in showing how vote shares evolve over time, allowing for a temp
#The stacked area chart introduces a temporal dimension to candidate support.

# Create a stacked area chart
area_chart <- ggplot(d, aes(x = reorder(name_image, desc(salinas_prop)), y = salinas_prop, fill = "Sali
  geom_area() +
  geom_area(aes(y = cardenas_prop, fill = "Cardenas")) +
  geom_area(aes(y = clouthier_prop, fill = "Clouthier")) +
  labs(title = "Cumulative Distribution of Vote Shares for Presidential Candidates",
       x = "Time",
       y = "Cumulative Vote Share") +
  scale_fill_manual(values = c(Salinas = "blue", Cardenas = "red", Clouthier = "green"), name = "Candida
  theme_minimal() +
  theme(legend.position = "top")

print(area_chart)
```



Note: Feel free to suggest *multiple* alternative designs to earn bonus credits. However, please be selective. Only a design with major differences from the existing ones can be counted as an alternative design.

Task 5. Visualize the discrepancies between presidential and legislative Votes (6pt)

In this task, you will visualize the differences between the number of presidential votes across tallies. The desired output of is reproducing and extending Figure 5 in the research article (Cantu 2019, pp. 720).

Task 5.1. Get district-level discrepancies and fraud data

As you might have noticed in the caption of Figure 5 in Cantu (2019, pp. 720), the visualized data are aggregated to the *district* level. In contrast, the unit of analysis in the dataset we are working with, `d`, is *tally*. As a result, the first step of this task is to aggregate the data. Specifically, please aggregate `d` into a new data frame named `sum_fraud_by_district`, which contains the following columns:

- `state`: Names of states
- `district`: Names of districts
- `vote_president`: Total numbers of presidential votes
- `vote_legislature`: Total numbers of legislative votes
- `vote_diff`: Total number of presidential votes minus total number of legislative votes
- `prop_fraud`: Proportions of fraudulent tallies (hint: using `fraud_bin`)

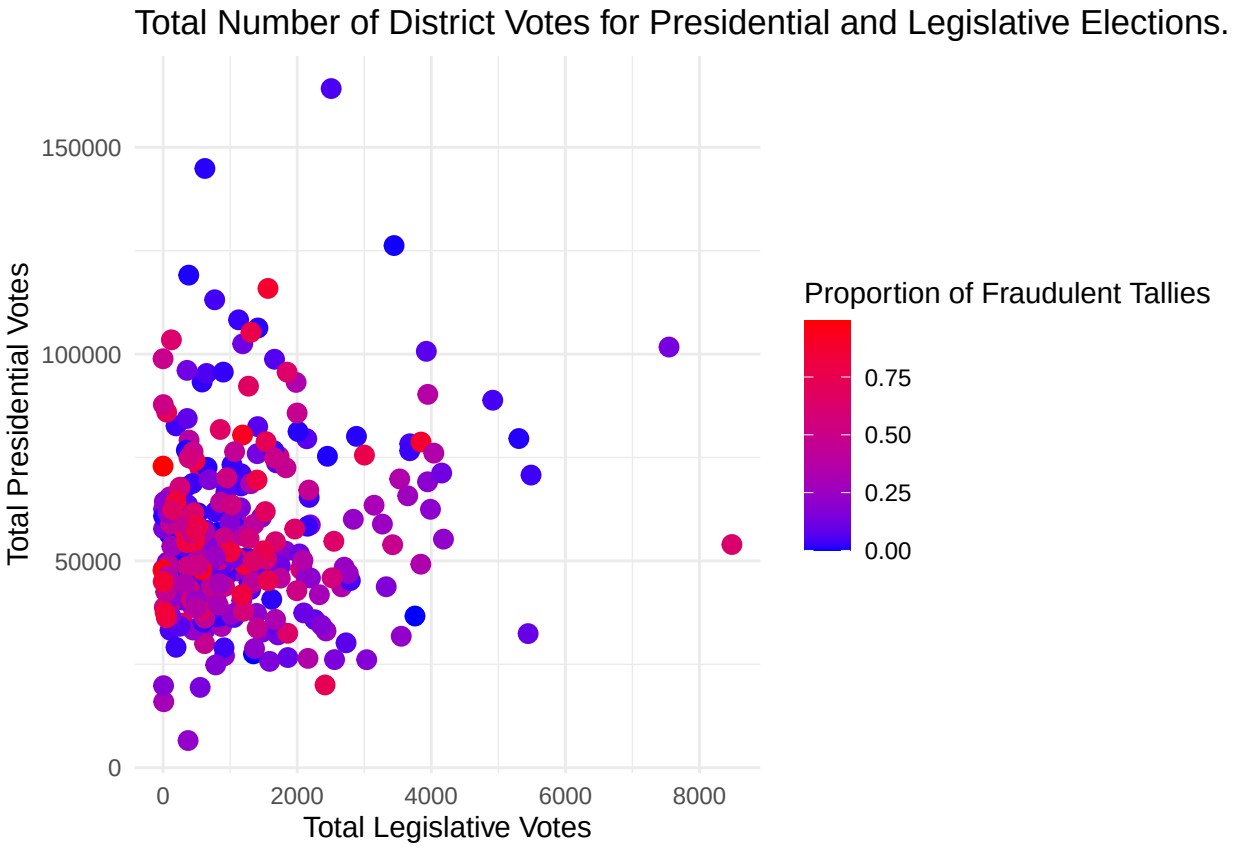
```
# YOUR CODE HERE

# Aggregate data to get district-level discrepancies and fraud data
sum_fraud_by_district <- d |>
  group_by(state, district) |>
  summarize(
    vote_president = sum(total),          # Total presidential votes
    vote_legislature = sum(total1),        # Total legislative votes
    vote_diff = sum(total) - sum(total1),  # Presidential votes minus legislative votes
    prop_fraud = mean(fraud_bin)           # Proportion of fraudulent tallies
  ) |>
  ungroup()

print(sum_fraud_by_district)
```

```
## # A tibble: 300 x 6
##    state           district vote_president vote_legislature vote_diff prop_fraud
##    <chr>           <chr>             <dbl>            <dbl>     <dbl>      <dbl>
##  1 Aguascalientes  I                101716             7546     94170     0.135
##  2 Aguascalientes  II                55271             4182     51089     0.215
##  3 Baja California I                 60550             1467     59083     0.171
##  4 Baja California II                32429             5444     26985     0.0960
##  5 Baja California III               75940             1404     74536     0.132
##  6 Baja California IV                90270             3948     86322     0.375
##  7 Baja California V                 48971              581     48390     0.152
##  8 Baja California VI                60596             1082     59514     0.368
##  9 Baja Californi~ I                 47569             1110     46459     0.259
## 10 Baja Californi~ II                26641             1860     24781     0.0933
## # i 290 more rows
```

Task 5.2. Replicate Figure 5

Based on all the previous step, reproduce Figure 5 in Cantu (2019, pp. 720).

```
# YOUR CODE HERE

# Replicate Figure 5
ggplot(sum_fraud_by_district, aes(x = vote_legislature, y = vote_president, color = prop_fraud)) +
  geom_point(size = 3) +
  scale_color_gradient(low = "blue", high = "red") +
  labs(
    title = "Total Number of District Votes for Presidential and Legislative Elections. Mexico, 1988",
    x = "Total Legislative Votes",
    y = "Total Presidential Votes",
    color = "Proportion of Fraudulent Tallies"
  ) +
  theme_minimal()
```



Note 1: Your performance in this task will be mainly evaluated based on your output's similarity with the original figure. Pay attention to the details.

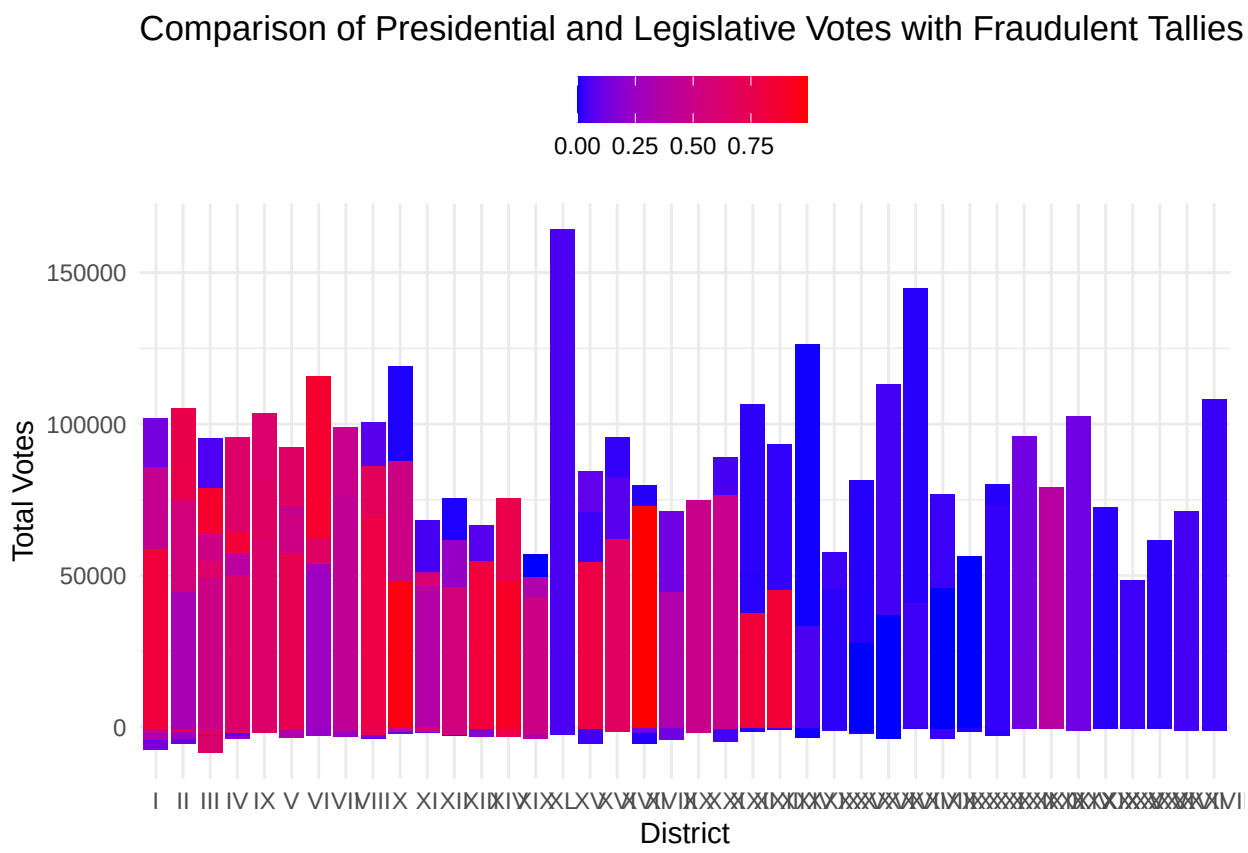
Note 2: The instructor has detected some differences between the above figure with Figure 5 on the published article. Please use the instructor's version as your main benchmark.

Task 5.3. Discuss and extend the reproduced figure

Referring to your reproduced figures and the research articles, in what way is the researcher's argument supported by this figure? Make an alternative visualization design that can substantiate and even augment the current argument. After you have shown your alternative design, in a few sentences, describe how your design provides visual aid as effectively as or more effectively than the original figure.

Note: Feel free to make *multiple* alternative designs to earn bonus credits. However, please be selective. Only a design with major differences from the existing ones can be counted as an alternative design.

```
# YOUR CODE HERE

# Alternative Design: Grouped Bar Chart
ggplot(sum_fraud_by_district, aes(x = district)) +
  geom_bar(aes(y = vote_president, fill = prop_fraud), position = "dodge", stat = "identity") +
  geom_bar(aes(y = -vote_legislature, fill = prop_fraud), position = "dodge", stat = "identity") +
  scale_fill_gradient(low = "blue", high = "red") +
  labs(
    title = "Comparison of Presidential and Legislative Votes with Fraudulent Tallies by District",
    x = "District",
    y = "Total Votes",
    fill = "Proportion of Fraudulent Tallies"
  ) +
  theme_minimal() +
  theme(legend.position = "top", legend.title = element_blank())
```



```
#The scatter plot effectively highlights the relationship between these variables, emphasizing regions
#An alternative design could be a grouped bar chart, where each district is represented by a pair of ba
#The height of the bar: represents the total number of votes in that category
#Colors within each bar segments: the proportion of fraudulent tallies.
#This alternative design provides a more direct comparison between presidential and legislative votes i
#The alternative design is effective as it complements the scatter plot by offering a different perspec
```

Task 6. Visualize the spatial distribution of fraud (6pt)

In this final task, you will visualize the spatial distribution of electoral fraud in Mexico. The desired output of is reproducing and extending Figure 3 in the research article (Cantu 2019, pp. 720).

Note 3. Load map data

As you may recall, map data can be stored and shared in **two** ways. The simpler format is a table where each row has information of a point that "carves" the boundary of a geographic unit (a Mexican state in our case). In this type of map data, a geographic unit is is represented by multiple rows. Alternatively, a map can be represented by a more complicated and more powerful format, where each geographic unit (a Mexican state in our case) is represented by an element of a `geometry` column. For this task, I provide you with a state-level map of Mexico represented by both formats respectively.

Below the instructor provide you with the code to load the maps stored under the two formats respectively. Please run them before starting to work on your task.

```
# IMPORTANT: Remove eval=FALSE above when you start this part!

# Load map (simple)
map_mex <- read_csv("data/map_mexico/map_mexico.csv")
# Load map (sf): You need to install and load library "sf" in advance
map_mex_sf <- st_read("data/map_mexico/shapefile/gadm36_MEX_1.shp")
map_mex_sf <- st_simplify(map_mex_sf, dTolerance = 100)

# Bonus question: the st_simplify function is applied to map_mex_sf with a tolerance of 100. This funct
# Lower values of the dTolerance parameter retain more detail but may result in larger file sizes, whil
```

Bonus question: Explain the operations on `map_mex_sf` in the instructor's code above.

Note: The map (sf) data we use are from https://gadm.org/download_country_v3.html.

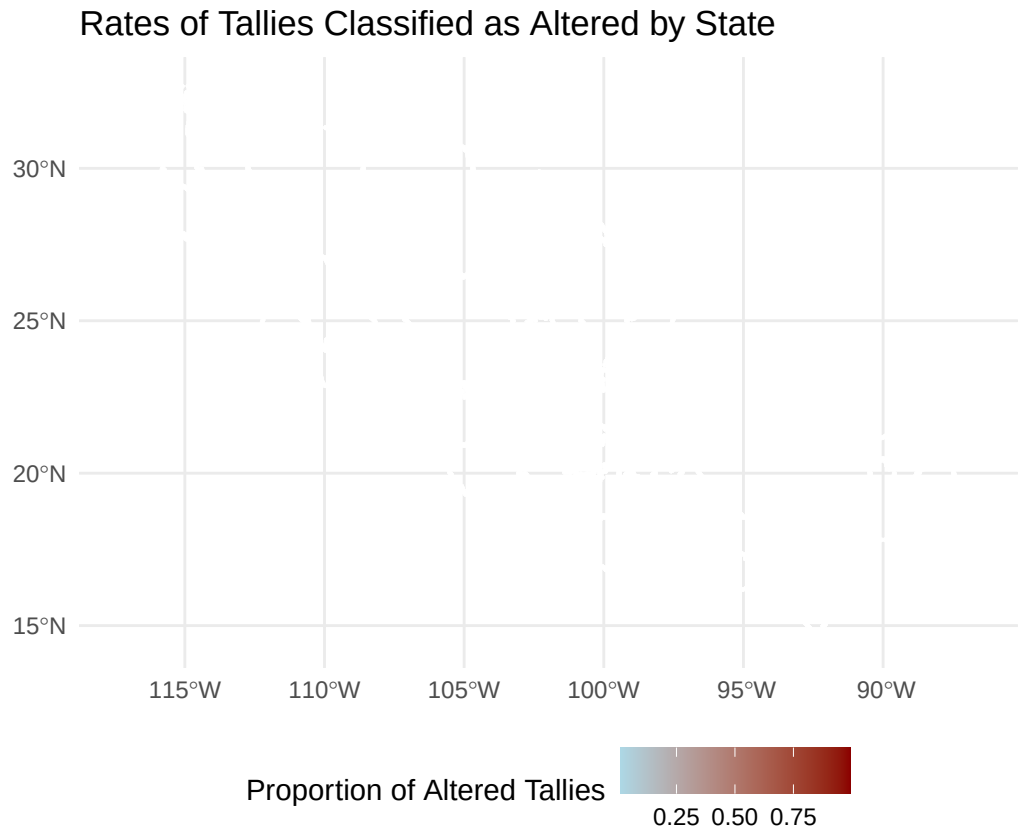
Task 6.1. Reproduce Figure 3 with `map_mex`

In this task, you are required to reproduce Figure 3 with the `map_mex` data.

Note:

- Your performance in this task will be mainly evaluated based on your output's similarity with the original figure. Pay attention to the details. For your reference, below is a version created by the instructor.
- Hint: Check the states' names in the map data and the electoral fraud data. Recode them if necessary.

```
# YOUR CODE HERE

# Convert map_mex to an sf object
map_mex_sf <- st_as_sf(map_mex, coords = c("long", "lat"), crs = st_crs(4326))

# Merge electoral fraud data with map data
map_fraud <- left_join(map_mex_sf, sum_fraud_by_district, by = c("state_name" = "state"))

# Check for missing data in the merged dataset
missing_data <- sum(is.na(map_fraud$prop_fraud))
if (missing_data > 0) {
  warning(paste("There are", missing_data, "states with missing data. Check for discrepancies in state n
}

# Plot the map
ggplot() +
  geom_sf(data = map_fraud, aes(fill = prop_fraud), color = "white", size = 0.2) +
  scale_fill_gradient(name = "Proportion of Altered Tallies", low = "lightblue", high = "darkred") +
  theme_minimal() +
  theme(legend.position = "bottom") +
  ggtitle("Rates of Tallies Classified as Altered by State")
```


Rates of Tallies Classified as Altered by State
30°N
25°N
20°N
15°N
115°W
110°W
105°W
100°W
95°W
90°W
Proportion of Altered Tallies
0.25
0.50
0.75

Task 6.2. Reproduce Figure 3 with `map_mex_sf`

In this task, you are required to reproduce Figure 3 with the `map_mex` data.

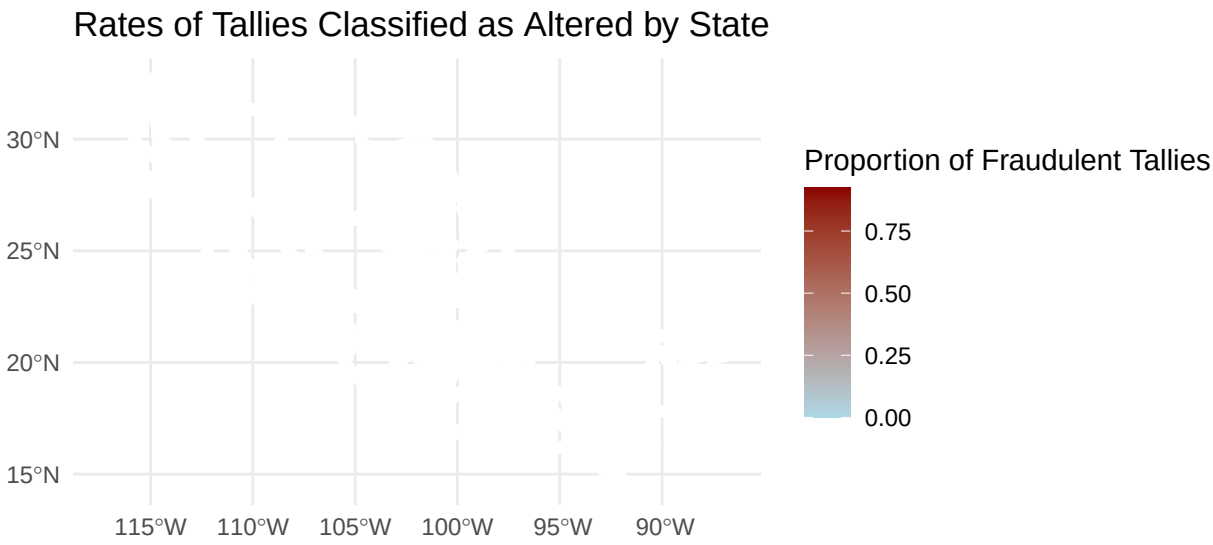
Note:

- Your performance in this task will be mainly evaluated based on your output's similarity with the original figure. Pay attention to the details. For your reference, below is a version created by the instructor.
- Hint: Check the states' names in the map data and the electoral fraud data. Recode them if necessary.

```
# YOUR CODE HERE

# Check the structure of map_mex_sf
str(map_mex_sf)
```

```
## sf [65,182 x 11] (S3: sf/tbl_df/tbl/data.frame)
##  $ order              : num [1:65182] 1 2 3 4 5 6 7 8 9 10 ...
##  $ hole               : logi [1:65182] FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ piece              : num [1:65182] 1 1 1 1 1 1 1 1 1 1 ...
##  $ id                 : chr [1:65182] "01" "01" "01" "01" ...
##  $ group              : chr [1:65182] "01.1" "01.1" "01.1" "01.1" ...
##  $ region             : chr [1:65182] "01" "01" "01" "01" ...
##  $ state_name         : chr [1:65182] "Aguascalientes" "Aguascalientes" "Aguascalientes" "Aguascalien
##  $ state_name_official: chr [1:65182] "Aguascalientes" "Aguascalientes" "Aguascalientes" "Aguascalien
##  $ state_abbr         : chr [1:65182] "AGS" "AGS" "AGS" "AGS" ...
##  $ state_abbr_official: chr [1:65182] "Ags." "Ags." "Ags." "Ags." ...
##  $ geometry           :sfc_POINT of length 65182; first list element:  'XY' num [1:2] -102.3 22.4
##  - attr(*, "sf_column")= chr "geometry"
##  - attr(*, "agr")= Factor w/ 3 levels "constant","aggregate",..: NA NA NA NA NA NA NA NA NA NA
##   ..- attr(*, "names")= chr [1:10] "order" "hole" "piece" "id" ...
```

```
# Merge electoral fraud data with map data
map_fraud <- left_join(map_mex_sf, sum_fraud_by_district, by = c("state_name_official" = "state"))

# Check for missing values in the merged data
missing_values <- sum(is.na(map_fraud$prop_fraud))

if (missing_values > 0) {
  cat()
}

# Plot the spatial distribution of fraud
ggplot(map_fraud) +
  geom_sf(aes(fill = prop_fraud), color = "white", lwd = 0.2) +
  scale_fill_gradient(name = "Proportion of Fraudulent Tallies", low = "lightblue", high = "darkred") +
  theme_minimal() +
  labs(title = "Rates of Tallies Classified as Altered by State")
```

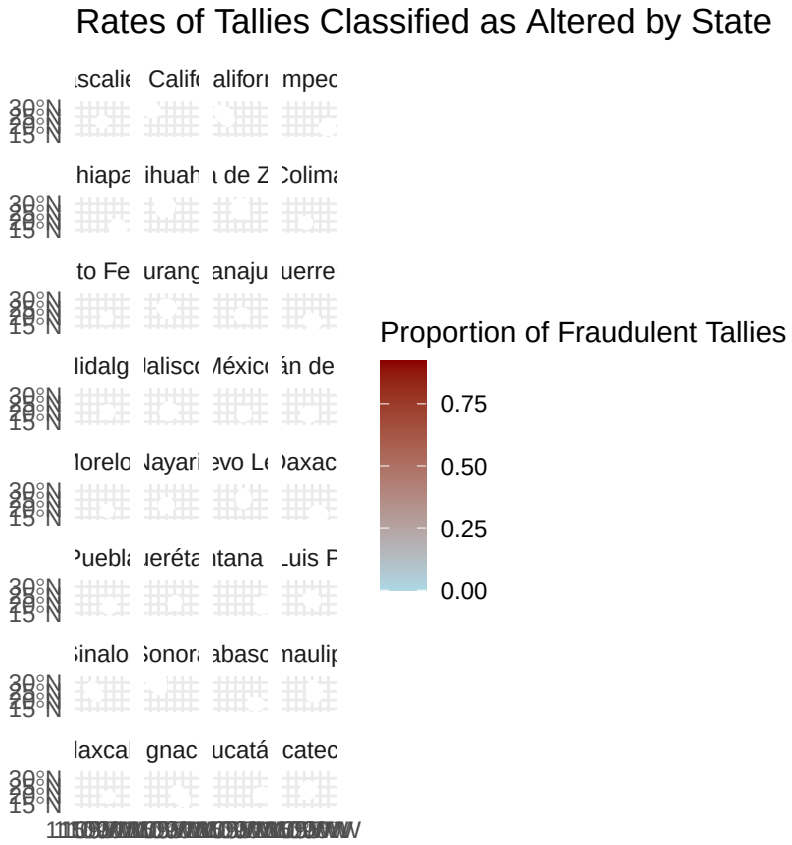

Rates of Tallies Classified as Altered by State
30°N
25°N
20°N
15°N
115°W
110°W
105°W
100°W
95°W
90°W
Proportion of Fraudulent Tallies
0.75
0.50
0.25
0.00

Task 6.3. Discuss and extend the reproduced figures

Referring to your reproduced figures and the research articles, in what way is the researcher's argument supported by this figure? Make an alternative visualization design that can substantiate and even augment the current argument. After you have shown your alternative design, in a few sentences, describe how your design provides visual aid as effectively as or more effectively than the original figure.

Note: Feel free to make *multiple* alternative designs to earn bonus credits. However, please be selective. Only a design with major differences from the existing ones can be counted as an alternative design.

```
# YOUR CODE HERE

# Alternative Design: Facetted Choropleth Map with Mini-Histograms
ggplot(map_fraud) +
  geom_sf(aes(fill = prop_fraud), color = "white", lwd = 0.2) +
  scale_fill_gradient(name = "Proportion of Fraudulent Tallies", low = "lightblue", high = "darkred") +
  facet_wrap(~state_name_official, scales = "fixed", ncol = 4) +
  theme_minimal() +
  labs(title = "Rates of Tallies Classified as Altered by State")
```




```
#The researchers' map: Visual representation of the spatial distribution of fraudulent tallies

#Alternative Design:

#Choropleth map but with an additional facetted approach: each state has its own mini-histogram or bar
```

```
#This provides a more nuanced view of fraud distribution within each state, shows the variation in frau

#This alternative design provides a state-level breakdown, allowing viewers to know patterns and variat
```