Chan Shuk Yee_3035995096

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.

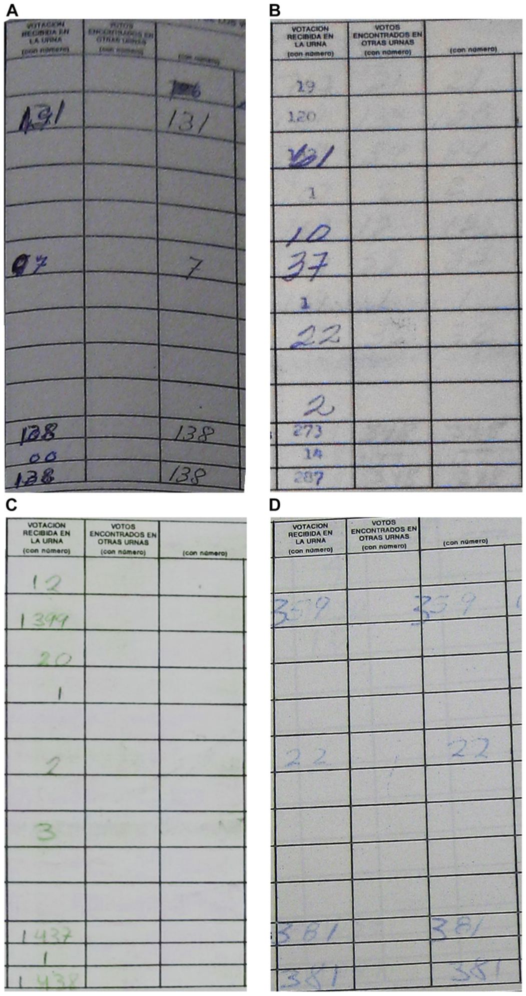

A
B
C
D
VOTACION RECIBIDA EN LA URNA (con número)
VOTOS ENCONTRADOS EN OTRAS URNAS (con número)
(con número)

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.

```r
library(tidyverse)
library(dplyr)
library(sf)
library(gridExtra)
```

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.

```r
# load the data set and name it d_tally
d_tally <- read_csv("data/classification.txt",
                    col_names = c("name_image", "label", "probability"))

print(d_tally)
```

```
## # A tibble: 55,334 x 3
##    name_image                              label probability
##    <chr>                                   <chr> <chr>
##  1 Aguascalientes_I_2014-05-26 00.00.10.jpg [[0]] [[ 0.99919599]]
##  2 Aguascalientes_I_2014-05-26 00.00.17.jpg [[0]] [[ 0.95722806]]
##  3 Aguascalientes_I_2014-05-26 00.00.25.jpg [[0]] [[ 0.57690716]]
##  4 Aguascalientes_I_2014-05-26 00.00.31.jpg [[0]] [[ 0.96505082]]
##  5 Aguascalientes_I_2014-05-26 00.00.38.jpg [[0]] [[ 0.86975688]]
##  6 Aguascalientes_I_2014-05-26 00.00.45.jpg [[0]] [[ 0.78825063]]
##  7 Aguascalientes_I_2014-05-26 00.00.52.jpg [[0]] [[ 0.96493018]]
##  8 Aguascalientes_I_2014-05-26 00.00.59.jpg [[0]] [[ 0.68087846]]
##  9 Aguascalientes_I_2014-05-26 00.01.06.jpg [[0]] [[ 0.99999994]]
## 10 Aguascalientes_I_2014-05-26 00.01.15.jpg [[0]] [[ 0.64047635]]
## # ... with 55,324 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.

```
# clean these two variables
d_tally <- d_tally |>
  mutate(
    label = as.numeric(str_extract(label, "\\d")),
    probability = as.numeric(str_extract(probability, "\\d+\\.\\d+")))

# summary statistics of label and probability
summary(d_tally$label)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.3623  1.0000  1.0000
```

```
summary(d_tally$probability)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.5000  0.8053  0.9619  0.8865  0.9988  1.0000    2960
```

```
print(d_tally)
```

```
## # A tibble: 55,334 x 3
##    name_image                              label probability
##    <chr>                                   <dbl>       <dbl>
##  1 Aguascalientes_I_2014-05-26 00.00.10.jpg     0       0.999
##  2 Aguascalientes_I_2014-05-26 00.00.17.jpg     0       0.957
##  3 Aguascalientes_I_2014-05-26 00.00.25.jpg     0       0.577
##  4 Aguascalientes_I_2014-05-26 00.00.31.jpg     0       0.965
##  5 Aguascalientes_I_2014-05-26 00.00.38.jpg     0       0.870
##  6 Aguascalientes_I_2014-05-26 00.00.45.jpg     0       0.788
##  7 Aguascalientes_I_2014-05-26 00.00.52.jpg     0       0.965
##  8 Aguascalientes_I_2014-05-26 00.00.59.jpg     0       0.681
##  9 Aguascalientes_I_2014-05-26 00.01.06.jpg     0       1.00
## 10 Aguascalientes_I_2014-05-26 00.01.15.jpg     0       0.640
## # ... with 55,324 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.

```r
# Extract state and district information
d_tally <- d_tally |>
  separate(name_image, into = c("state", "district"), sep = "_", remove = FALSE) |>
  mutate(
    state = str_remove(state, ".jpg"),
    state = as.factor(state),
    district = as.factor(district))

print(d_tally)
```

```
## # A tibble: 55,334 x 5
##    name_image                              state          distr~1 label proba~2
##    <chr>                                   <fct>          <fct>   <dbl>   <dbl>
##  1 Aguascalientes_I_2014-05-26 00.00.10.jpg Aguascalientes I           0   0.999
##  2 Aguascalientes_I_2014-05-26 00.00.17.jpg Aguascalientes I           0   0.957
##  3 Aguascalientes_I_2014-05-26 00.00.25.jpg Aguascalientes I           0   0.577
##  4 Aguascalientes_I_2014-05-26 00.00.31.jpg Aguascalientes I           0   0.965
##  5 Aguascalientes_I_2014-05-26 00.00.38.jpg Aguascalientes I           0   0.870
##  6 Aguascalientes_I_2014-05-26 00.00.45.jpg Aguascalientes I           0   0.788
##  7 Aguascalientes_I_2014-05-26 00.00.52.jpg Aguascalientes I           0   0.965
##  8 Aguascalientes_I_2014-05-26 00.00.59.jpg Aguascalientes I           0   0.681
##  9 Aguascalientes_I_2014-05-26 00.01.06.jpg Aguascalientes I           0   1.00
## 10 Aguascalientes_I_2014-05-26 00.01.15.jpg Aguascalientes I           0   0.640
## # ... with 55,324 more rows, and abbreviated variable names 1: district,
## #   2: probability
```

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`.

```
#Re-code a state's name
d_tally<- d_tally |>
  mutate(
    state = case_when(
      state == "Estado de Mexico" ~ "Edomex",
      TRUE ~ state ))

print(d_tally)
```

```
## # A tibble: 55,334 x 5
##    name_image                               state          distr~1 label proba~2
##    <chr>                                    <chr>          <fct>   <dbl>   <dbl>
##  1 Aguascalientes_I_2014-05-26 00.00.10.jpg Aguascalientes I           0   0.999
##  2 Aguascalientes_I_2014-05-26 00.00.17.jpg Aguascalientes I           0   0.957
##  3 Aguascalientes_I_2014-05-26 00.00.25.jpg Aguascalientes I           0   0.577
##  4 Aguascalientes_I_2014-05-26 00.00.31.jpg Aguascalientes I           0   0.965
##  5 Aguascalientes_I_2014-05-26 00.00.38.jpg Aguascalientes I           0   0.870
##  6 Aguascalientes_I_2014-05-26 00.00.45.jpg Aguascalientes I           0   0.788
##  7 Aguascalientes_I_2014-05-26 00.00.52.jpg Aguascalientes I           0   0.965
##  8 Aguascalientes_I_2014-05-26 00.00.59.jpg Aguascalientes I           0   0.681
##  9 Aguascalientes_I_2014-05-26 00.01.06.jpg Aguascalientes I           0   1.00
## 10 Aguascalientes_I_2014-05-26 00.01.15.jpg Aguascalientes I           0   0.640
## # ... with 55,324 more rows, and abbreviated variable names 1: district,
## #   2: probability
```

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).

```
#probability of fraud indicator
d_tally <- d_tally |>
  mutate(fraud_proba = ifelse(label == 1, probability, 1 - probability)) |>
  select(-label, -probability)

print(d_tally)
```

```
## # A tibble: 55,334 x 4
##    name_image                               state          district  fraud_proba
##    <chr>                                    <chr>          <fct>           <dbl>
##  1 Aguascalientes_I_2014-05-26 00.00.10.jpg Aguascalientes I        0.000804
##  2 Aguascalientes_I_2014-05-26 00.00.17.jpg Aguascalientes I        0.0428
##  3 Aguascalientes_I_2014-05-26 00.00.25.jpg Aguascalientes I        0.423
##  4 Aguascalientes_I_2014-05-26 00.00.31.jpg Aguascalientes I        0.0349
##  5 Aguascalientes_I_2014-05-26 00.00.38.jpg Aguascalientes I        0.130
##  6 Aguascalientes_I_2014-05-26 00.00.45.jpg Aguascalientes I        0.212
##  7 Aguascalientes_I_2014-05-26 00.00.52.jpg Aguascalientes I        0.0351
##  8 Aguascalientes_I_2014-05-26 00.00.59.jpg Aguascalientes I        0.319
##  9 Aguascalientes_I_2014-05-26 00.01.06.jpg Aguascalientes I        0.0000000600
## 10 Aguascalientes_I_2014-05-26 00.01.15.jpg Aguascalientes I        0.360
## # ... with 55,324 more rows
```

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.

```
#binary fraud indicator
d_tally<- d_tally |>
  mutate(fraud_bin = fraud_proba >= 2/3)

print(d_tally)
```

```
## # A tibble: 55,334 x 5
##    name_image                              state        distr~1 fraud~2 fraud~3
##    <chr>                                   <chr>        <fct>     <dbl> <lgl>
##  1 Aguascalientes_I_2014-05-26 00.00.10.jpg Aguascalien~ I       8.04e-4 FALSE
##  2 Aguascalientes_I_2014-05-26 00.00.17.jpg Aguascalien~ I       4.28e-2 FALSE
##  3 Aguascalientes_I_2014-05-26 00.00.25.jpg Aguascalien~ I       4.23e-1 FALSE
##  4 Aguascalientes_I_2014-05-26 00.00.31.jpg Aguascalien~ I       3.49e-2 FALSE
##  5 Aguascalientes_I_2014-05-26 00.00.38.jpg Aguascalien~ I       1.30e-1 FALSE
##  6 Aguascalientes_I_2014-05-26 00.00.45.jpg Aguascalien~ I       2.12e-1 FALSE
##  7 Aguascalientes_I_2014-05-26 00.00.52.jpg Aguascalien~ I       3.51e-2 FALSE
##  8 Aguascalientes_I_2014-05-26 00.00.59.jpg Aguascalien~ I       3.19e-1 FALSE
##  9 Aguascalientes_I_2014-05-26 00.01.06.jpg Aguascalien~ I       6.00e-8 FALSE
## 10 Aguascalientes_I_2014-05-26 00.01.15.jpg Aguascalien~ I       3.60e-1 FALSE
## # ... with 55,324 more rows, and abbreviated variable names 1: district,
## #   2: fraud_proba, 3: fraud_bin
```

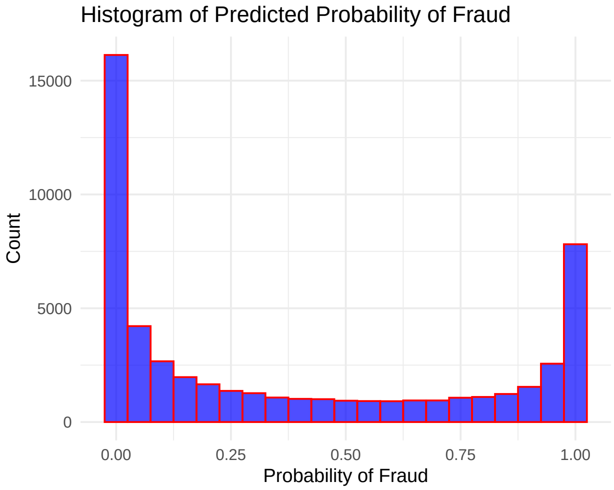
Task 2. Visualize machine classification results (3pt)

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.

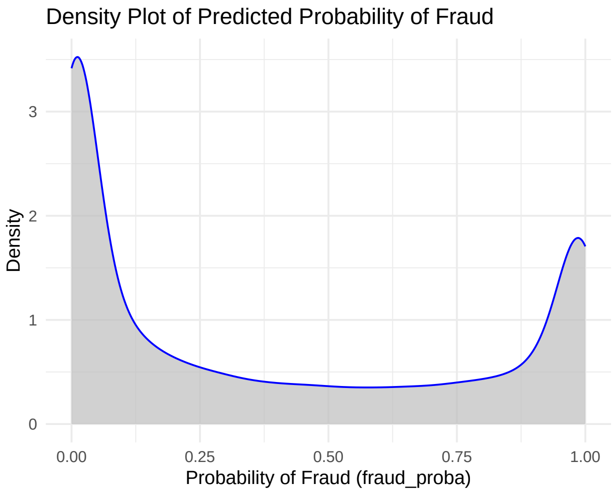
Task 2.1. Visualize distribution of `fraud_proba`

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.

```r
# Use histogram to visualize the distribution
d_tally |>
  ggplot(aes(x = fraud_proba)) +
  geom_histogram(binwidth = 0.05, fill = "blue", color = "red", alpha = 0.7) +
  labs(title = "Histogram of Predicted Probability of Fraud",
       x = "Probability of Fraud", y = "Count") +
  theme_minimal()
```



```r
# Use Density Plot to visualize the distribution
d_tally |>
  ggplot(aes(x = fraud_proba)) +
  geom_density(fill = "gray", color = "blue", alpha = 0.7) +
  labs(title = "Density Plot of Predicted Probability of Fraud",
       x = "Probability of Fraud (fraud_proba)",
       y = "Density") +
  theme_minimal()
```



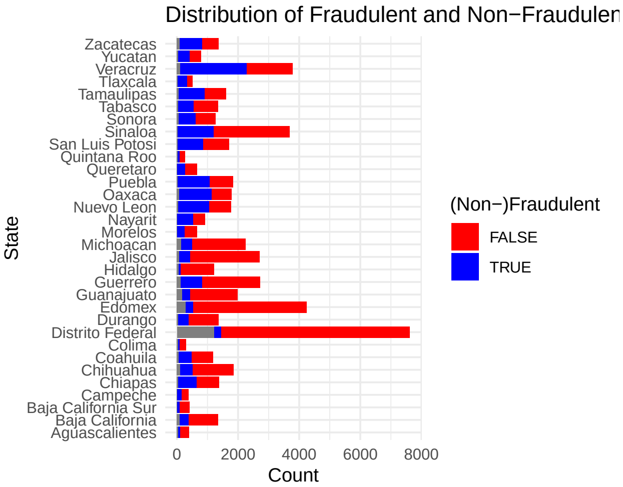
Describe the plot with a few sentences:

These two plots, the histogram and the density plot, are related to the Predicted Probability of Fraud in the 1998 Mexico election, the x-axis refers to the probability of fraud while the y-axis refers to its count and density. The distribution of the histogram and density plot are non-symmetrical bimodal, the first peak is around the probability of 0.00 and the second peak is around the probability of 1.00.

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.

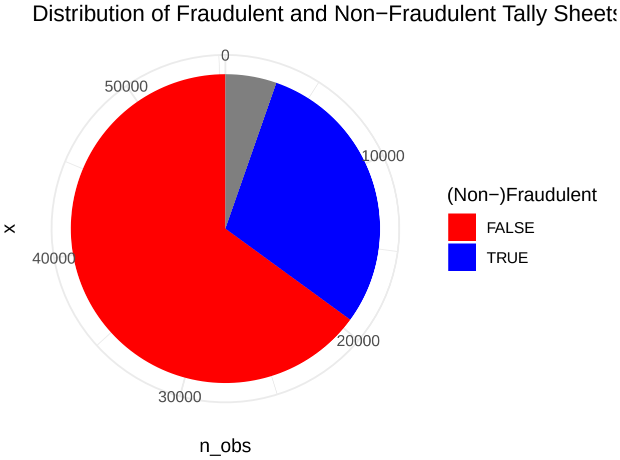
```r
## Use Stacked Bar Chart to visualize the distribution
d_tally |>
  group_by(state, district, fraud_bin) |>
  summarise(n_obs = n()) |>
  ggplot(aes(x = n_obs, y = state, fill = factor(fraud_bin))) +
  geom_bar(stat = "identity", position = "stack") +
  labs(y = "State", x = "Count", title = "Distribution of Fraudulent and Non-Fraudulent Tally Sheets by
    scale_fill_manual(values = c("TRUE" = "blue", "FALSE" = "red", "NA" = "gray"))
```



Describe your plots with a few sentences:

The title of the stacked bar chart is Distribution of fraudulent and non-fraudulent tally sheets by state, it shows that there is a special case in Distrito Federal. In the case of Distrito Federal, the situation of fraudulent tally sheets is relatively more serious than in other states and far away from the second most fraudulent tally sheets.

```r
d_tally |>
  group_by(fraud_bin) |>
  summarise(n_obs = n()) |>
  ggplot(aes(x = "", y = n_obs, fill = factor(fraud_bin))) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar(theta = "y") +
  labs(title = "Distribution of Fraudulent and Non-Fraudulent Tally Sheets", fill = "(Non-)Fraudulent")
  scale_fill_manual(values = c("TRUE" = "blue", "FALSE" = "red", "NA" = "gray"))
```



Describe your plots with a few sentences:

The title of the pie bar chart is Distribution of fraudulent and non-fraudulent tally sheets by state, it shows that more than half of the data is fraudulent. As seen from the stacked bar chart and the pie chart, it is concentrated at the "false."

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.

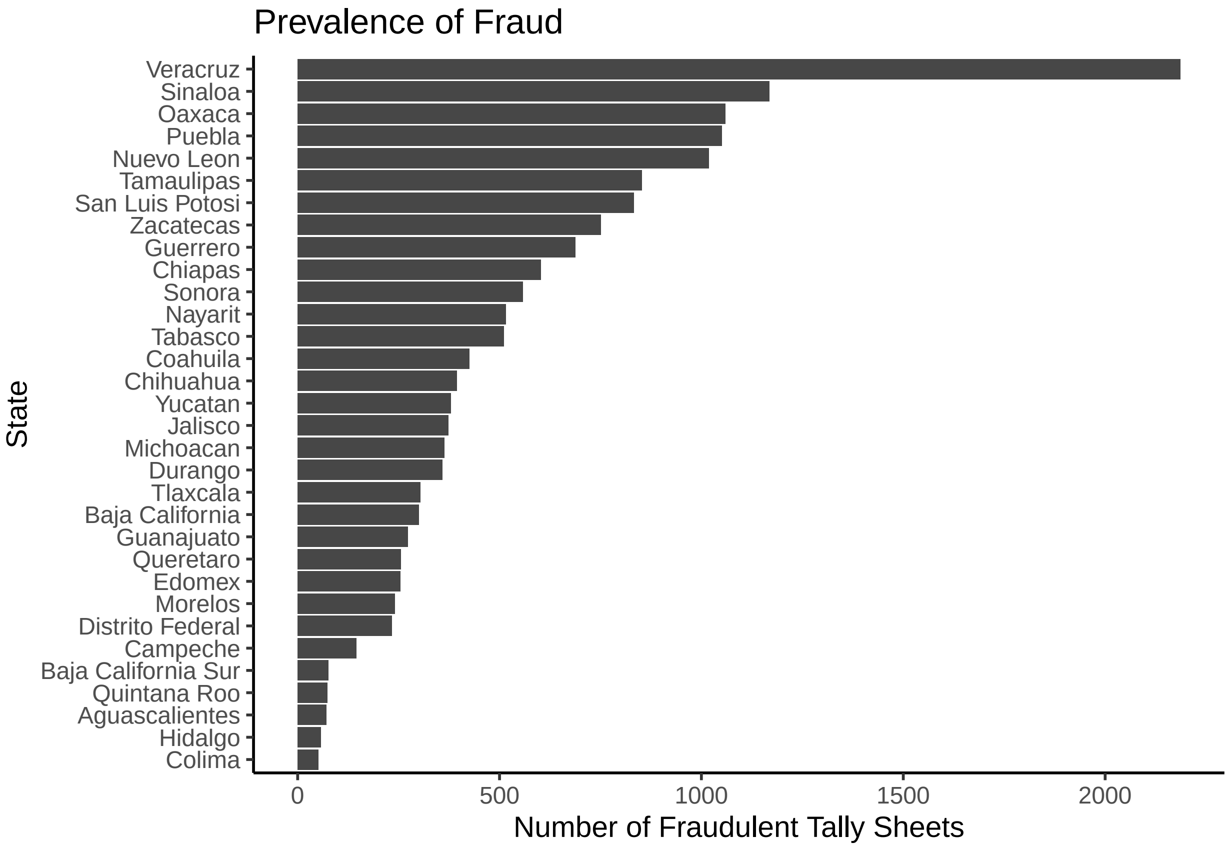
```r
# Summarize prevalence of fraud by state
d_state <- d_tally |>
  group_by(state) |>
  summarise(
    n_fraud = sum(fraud_bin, na.rm = TRUE),
    prop_fraud = mean(fraud_bin, na.rm = TRUE) * 100 )


summary(d_state)
```

```
##     state              n_fraud         prop_fraud
##  Length:32          Min.   :  51.0   Min.   : 3.633
##  Class :character   1st Qu.: 250.8   1st Qu.:18.703
##  Mode  :character   Median : 376.0   Median :37.398
##                     Mean   : 513.3   Mean   :35.330
##                     3rd Qu.: 703.8   3rd Qu.:51.696
##                     Max.   :2186.0   Max.   :61.736
```

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.



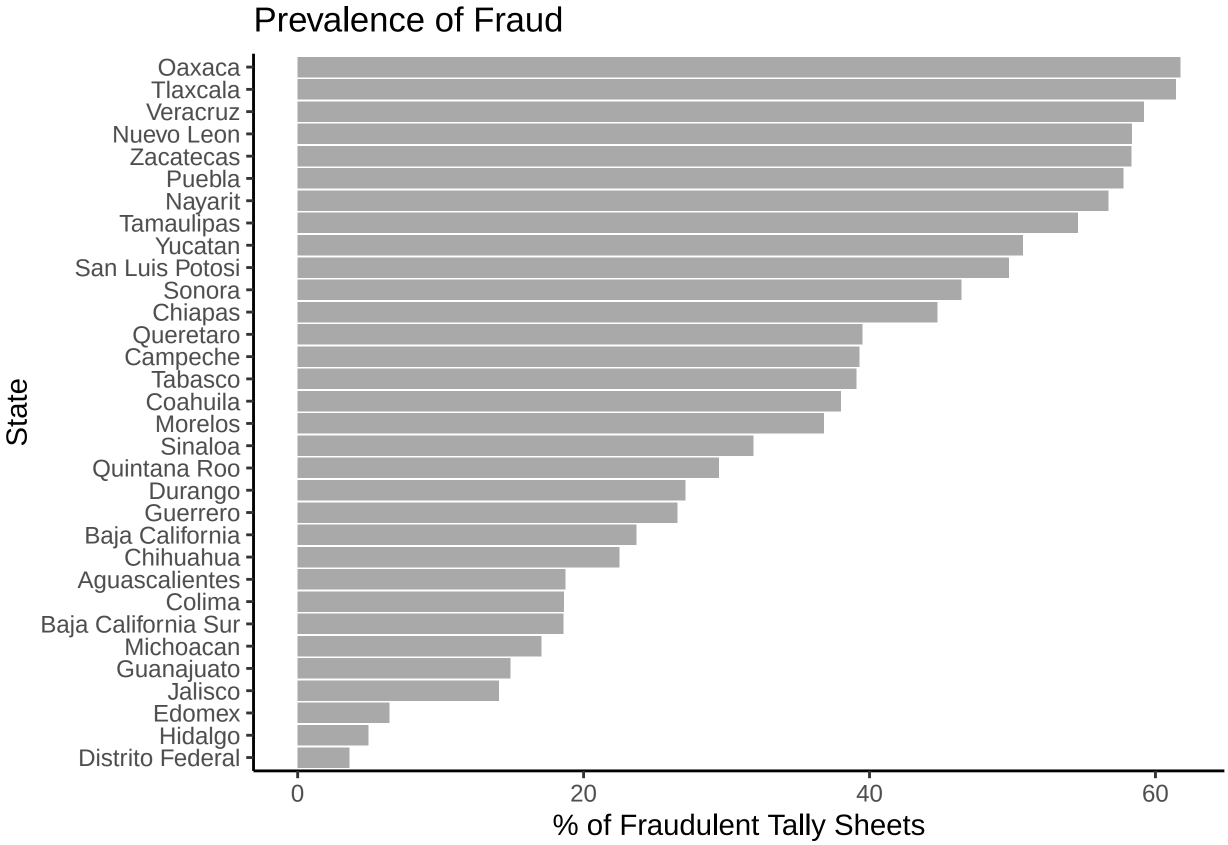
Describe the key takeaway from the visualization with a few sentences:

The title of the bar chart is the Number of Fraudulent Tally Sheets which is a plot clearly compares the number of fraudulent tallies in each state. It provides a clear comparison of the number of fraudulent tallies in each state, in which Veracruz has the highest prevalence of fraudulent tallies whereas Colima has the lowest prevalence of fraudulent tallies.

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.

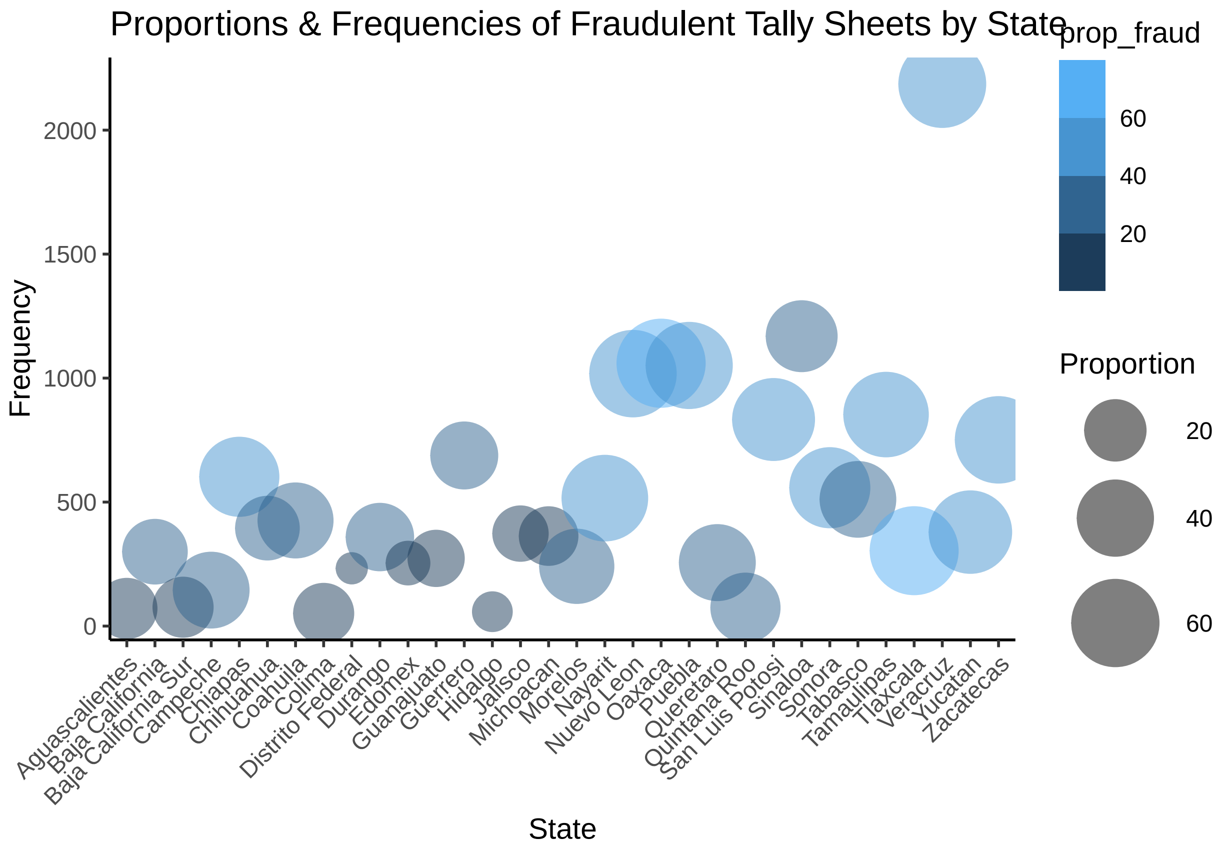


The title of the bar chart is the prevalence of fraud which it visualizes the variation in the proportion of fraudulent tallies across different states. The plot shows that Oaxaca has the highest proportion of fraudulent tallies and Distrito Federal has the lowest proportion of fraudulent tallies.

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.

```r
# creat a bubble plot
d_state |>
  ggplot(aes(x = state, y = n_fraud, size = prop_fraud)) +
  geom_point(aes(color = prop_fraud), alpha = 0.5) +
  scale_size_continuous(range = c(5, 15)) +
  labs(title = "Proportions & Frequencies of Fraudulent Tally Sheets by State",
       x = "State",
       y = "Frequency",
       size = "Proportion") +
  scale_color_binned() +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Describe the takeaways from the figure with a few sentences:

The title of the bubble plot is proportions & frequencies of fraudulent tally sheets by sate, Veracruz has a relatively high frequency and a large proportion of fraudulent tally sheets. The proportion of states having fraudulent tally sheets is relatively large as well.

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.

```
# load d_return
d_return <- read_csv("data/VoteReturns.csv")

print(d_return)
```

```
## # A tibble: 53,499 x 91
##    foto     seccion casilla dtto    dto munic~1 edo   entidad pagina    p1    p2
##    <chr>    <chr>   <chr>   <chr> <dbl> <chr>   <chr> <chr>    <dbl> <dbl> <dbl>
##  1 2014-05~ 83      83      I         1 AGUASC~ Agua~ AGS        127   108   333
##  2 2014-05~ 1       84      <NA>      1 AGUASC~ Agua~ AGUASC~    128   919   453
##  3 2014-05~ 85      85      1         1 AGUASC~ Agua~ AGUASC~    129   795   264
##  4 2014-05~ 45      45-A    1         1 AGUASC~ Agua~ AGUA       130   767   450
##  5 2014-05~ 86      86      1         1 AGUASC~ Agua~ AGUAS      131  1243   578
##  6 2014-05~ 87      87      1         1 <NA>    Agua~ 1          132   718   333
##  7 2014-05~ 1       87-A    7         1 AGUASC~ Agua~ AGUAS      133   710   299
##  8 2014-05~ 88      88      1         1 AGUAS   Agua~ AGUAS      134     0     0
##  9 2014-05~ 89      89      1         1 AGUASC~ Agua~ AGUAS      135   764     8
## 10 2014-05~ 89      89-A    7         1 AGUSCA~ Agua~ 1          136   759   256
## # ... with 53,489 more rows, 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>, ...
```

```
# create summary statistics
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
##
```

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.

```
#re-code the names
d_return <- d_return |>
  mutate(edo = ifelse(edo == "Chihuhua", "Chihuahua", edo),
         edo = ifelse(edo == "Edomex", "Estado de Mexico", edo))

print(d_return)
```

```
## # A tibble: 53,499 x 91
##    foto     seccion casilla dtto    dto munic~1 edo   entidad pagina    p1    p2
##    <chr>    <chr>   <chr>   <chr> <dbl> <chr>   <chr> <chr>    <dbl> <dbl> <dbl>
##  1 2014-05~ 83      83      I         1 AGUASC~ Agua~ AGS        127   108   333
##  2 2014-05~ 1       84      <NA>      1 AGUASC~ Agua~ AGUASC~    128   919   453
##  3 2014-05~ 85      85      1         1 AGUASC~ Agua~ AGUASC~    129   795   264
##  4 2014-05~ 45      45-A    1         1 AGUASC~ Agua~ AGUA       130   767   450
##  5 2014-05~ 86      86      1         1 AGUASC~ Agua~ AGUAS      131  1243   578
##  6 2014-05~ 87      87      1         1 <NA>    Agua~ 1          132   718   333
##  7 2014-05~ 1       87-A    7         1 AGUASC~ Agua~ AGUAS      133   710   299
##  8 2014-05~ 88      88      1         1 AGUAS   Agua~ AGUAS      134     0     0
##  9 2014-05~ 89      89      1         1 AGUASC~ Agua~ AGUAS      135   764     8
## 10 2014-05~ 89      89-A    7         1 AGUSCA~ Agua~ 1          136   759   256
## # ... with 53,489 more rows, 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>, ...
```

\clearpage

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./

```
# summarize d_tally
summary(d_tally$district)
```

```
##        I      II     III      IV      IX       V      VI     VII    VIII       X
##     6218    6251    5065    4513    2490    5101    4246    3262    2956    1904
##       XI     XII    XIII     XIV     XIX      XL      XV     XVI    XVII   XVIII
##     1016    1014    1004     630     590     366     592     570     673     491
##       XX     XXI    XXII   XXIII    XXIV    XXIX     XXV    XXVI   XXVII  XXVIII
##      603     587     433     447     307     246     287     319     346     295
##      XXX     XXXI    XXXII   XXXIII    XXXIV    XXXIX     XXXV    XXXVI  XXXVII XXXVIII
##      274      343      302      248      354      202      125      193      210      261
```

```
unique(d_tally$district)
```

```
##  [1] I        II       III      IV       V        VI       IX       VII      VIII
## [10] X        XI       XII      XIII     XIV      XIX      XL       XV       XVI
## [19] XVII     XVIII    XX       XXI      XXII     XXIII    XXIV     XXIX     XXV
## [28] XXVI     XXVII    XXVIII   XXX      XXXI     XXXII    XXXIII   XXXIV    XXXIX
## [37] XXXV     XXXVI    XXXVII   XXXVIII
## 40 Levels: I II III IV IX V VI VII VIII X XI XII XIII XIV XIX XL XV ... XXXVIII
```

```
# summarize d_return
summary(d_return$dto)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA’s
##   1.000   3.000   6.000   8.704  10.000 341.000       4
```

```
unique(d_return$dto)
```

```
##  [1]   1   2   5   6   3   4   7   8   9  10  11  12  13  14  15  16  17  18  19
## [20]  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38
## [39]  39  40  NA 341
```

```
# preparation of conversion
roman_convertor <- function(x) {
  roman_result <- ifelse(x >= 1000, "M", "")
  roman_result <- paste0(roman_result, strrep("C", x %% 1000 %/% 100))
  roman_result <- gsub("CCCC", "CD", roman_result)
  roman_result <- paste0(roman_result, strrep("X", x %% 100 %/% 10))
  roman_result <- gsub("XXXX", "XL", roman_result)
  roman_result <- paste0(roman_result, strrep("I", x %% 10))
  roman_result <- gsub("IIII", "IV", roman_result)
  roman_result}

# to convert
d_return <- d_return |>
  mutate(dto = ifelse(!is.na(dto), roman_convertor(as.integer(dto)), as.character(dto)))


print(d_return)
```

```
## # A tibble: 53,499 x 91
##    foto     seccion casilla dtto  dto   munic~1 edo   entidad pagina    p1    p2
##    <chr>    <chr>   <chr>   <chr> <chr> <chr>   <chr> <chr>    <dbl> <dbl> <dbl>
##  1 2014-05~ 83      83      I     I     AGUASC~ Agua~ AGS        127   108   333
##  2 2014-05~ 1       84      <NA>  I     AGUASC~ Agua~ AGUASC~    128   919   453
##  3 2014-05~ 85      85      1     I     AGUASC~ Agua~ AGUASC~    129   795   264
##  4 2014-05~ 45      45-A    1     I     AGUASC~ Agua~ AGUA       130   767   450
##  5 2014-05~ 86      86      1     I     AGUASC~ Agua~ AGUAS      131  1243   578
##  6 2014-05~ 87      87      1     I     <NA>    Agua~ 1          132   718   333
##  7 2014-05~ 1       87-A    7     I     AGUASC~ Agua~ AGUAS      133   710   299
##  8 2014-05~ 88      88      1     I     AGUAS   Agua~ AGUAS      134     0     0
##  9 2014-05~ 89      89      1     I     AGUASC~ Agua~ AGUAS      135   764     8
## 10 2014-05~ 89      89-A    7     I     AGUSCA~ Agua~ 1          136   759   256
## # ... with 53,489 more rows, 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>, ...
```

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.

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

#debug
d_return <- d_return |>
  select(name_image, everything())

print(d_return)
```

```
## # A tibble: 53,499 x 92
##    name_i~1 foto  seccion casilla dtto  dto   munic~2 edo   entidad pagina    p1
##    <chr>    <chr> <chr>   <chr>   <chr> <chr> <chr>   <chr> <chr>    <dbl> <dbl>
##  1 Aguasca~ 2014~ 83      83      I     I     AGUASC~ Agua~ AGS        127   108
##  2 Aguasca~ 2014~ 1       84      <NA>  I     AGUASC~ Agua~ AGUASC~    128   919
##  3 Aguasca~ 2014~ 85      85      1     I     AGUASC~ Agua~ AGUASC~    129   795
##  4 Aguasca~ 2014~ 45      45-A    1     I     AGUASC~ Agua~ AGUA       130   767
##  5 Aguasca~ 2014~ 86      86      1     I     AGUASC~ Agua~ AGUAS      131  1243
##  6 Aguasca~ 2014~ 87      87      1     I     <NA>    Agua~ 1          132   718
##  7 Aguasca~ 2014~ 1       87-A    7     I     AGUASC~ Agua~ AGUAS      133   710
##  8 Aguasca~ 2014~ 88      88      1     I     AGUAS   Agua~ AGUAS      134     0
##  9 Aguasca~ 2014~ 89      89      1     I     AGUASC~ Agua~ AGUAS      135   764
## 10 Aguasca~ 2014~ 89      89-A    7     I     AGUSCA~ Agua~ 1          136   759
## # ... with 53,489 more rows, 81 more variables: p2 <dbl>, 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>, ...
```

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.

```
## # A tibble: 53,499 x 92
##    name_i~1 foto  seccion casilla dtto  dto   munic~2 edo   entidad pagina    p1
##    <chr>    <chr> <chr>   <chr>   <chr> <chr> <chr>   <chr> <chr>    <dbl> <dbl>
##  1 aguasca~ 2014~ 83      83      I     I     AGUASC~ Agua~ AGS        127   108
##  2 aguasca~ 2014~ 1       84      <NA>  I     AGUASC~ Agua~ AGUASC~    128   919
##  3 aguasca~ 2014~ 85      85      1     I     AGUASC~ Agua~ AGUASC~    129   795
##  4 aguasca~ 2014~ 45      45-A    1     I     AGUASC~ Agua~ AGUA       130   767
##  5 aguasca~ 2014~ 86      86      1     I     AGUASC~ Agua~ AGUAS      131  1243
##  6 aguasca~ 2014~ 87      87      1     I     <NA>    Agua~ 1          132   718
##  7 aguasca~ 2014~ 1       87-A    7     I     AGUASC~ Agua~ AGUAS      133   710
##  8 aguasca~ 2014~ 88      88      1     I     AGUAS   Agua~ AGUAS      134     0
##  9 aguasca~ 2014~ 89      89      1     I     AGUASC~ Agua~ AGUAS      135   764
## 10 aguasca~ 2014~ 89      89-A    7     I     AGUSCA~ Agua~ 1          136   759
## # ... with 53,489 more rows, 81 more variables: p2 <dbl>, 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>, ...
```

```
## # A tibble: 55,334 x 5
##    name_image                           state          district fraud_~1 fraud~2
##    <chr>                                <chr>          <fct>       <dbl> <lgl>
##  1 aguascalientes_i_2014-05-26 00.00.10 Aguascalientes I         8.04e-4 FALSE
##  2 aguascalientes_i_2014-05-26 00.00.17 Aguascalientes I         4.28e-2 FALSE
##  3 aguascalientes_i_2014-05-26 00.00.25 Aguascalientes I         4.23e-1 FALSE
##  4 aguascalientes_i_2014-05-26 00.00.31 Aguascalientes I         3.49e-2 FALSE
##  5 aguascalientes_i_2014-05-26 00.00.38 Aguascalientes I         1.30e-1 FALSE
##  6 aguascalientes_i_2014-05-26 00.00.45 Aguascalientes I         2.12e-1 FALSE
##  7 aguascalientes_i_2014-05-26 00.00.52 Aguascalientes I         3.51e-2 FALSE
##  8 aguascalientes_i_2014-05-26 00.00.59 Aguascalientes I         3.19e-1 FALSE
##  9 aguascalientes_i_2014-05-26 00.01.06 Aguascalientes I         6.00e-8 FALSE
## 10 aguascalientes_i_2014-05-26 00.01.15 Aguascalientes I         3.60e-1 FALSE
## # ... with 55,324 more rows, and abbreviated variable names 1: fraud_proba,
## #   2: fraud_bin
```

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?

```
## [1] "Rows in d_return but not in d_tally: 42166"
```

```
##  [1] "Aguascalientes"      "Baja California"     "Baja California Sur"
##  [4] "Campeche"            "Chiapas"             "Chihuahua"
##  [7] "Coahuila"            "Colima"              "Distrito Federal"
## [10] "Durango"             "Estado de Mexico"    "Guanajuato"
## [13] "Guerrero"            "Hidalgo"             "Jalisco"
## [16] "Michoacan"           "Morelos"             "Nayarit"
## [19] "Nuevo Leon"          "Oaxaca"              "Puebla"
## [22] "Queretaro"           "Quintana Roo"        "San Luis Potosi"
## [25] "Sinaloa"             "Sonora"              "Tabasco"
## [28] "Tamaulipas"          "Tlaxcala"            "Veracruz"
## [31] "Yucatan"             "Zacatecas"
```

```
##  [1] "I"        "II"       "IVI"      "IVII"     "III"      "IV"
##  [7] "IVIII"    "IVIV"     "IVIVI"    "X"        "XI"       "XII"
## [13] "XIII"     "XIV"      "XIVI"     "XIVII"    "XIVIII"   "XIVIV"
## [19] "XIVIVI"   "XX"       "XXI"      "XXII"     "XXIII"    "XXIVI"
## [25] "XXIVII"   "XXIVIII"  "XXIVIV"   "XXIVIVI"  "XXX"      "XXXI"
## [31] "XXXII"    "XXXIII"   "XXXIV"    "XXXIVI"   "XXXIVII"  "XXXIVIII"
## [37] "XXXIVIV"  "XXXIVIVI" "XL"       NA         "XXIV"     "CCCXLI"
```

```
## [1] "Rows in d_return but not in d_return: 44016"
```

```
##  [1] "Aguascalientes"      "Baja California Sur" "Baja California"
##  [4] "Campeche"            "Chiapas"             "Chihuahua"
##  [7] "Coahuila"            "Colima"              "Distrito Federal"
## [10] "Durango"             "Edomex"              "Guanajuato"
## [13] "Guerrero"            "Hidalgo"             "Jalisco"
## [16] "Michoacan"           "Morelos"             "Nayarit"
## [19] "Nuevo Leon"          "Oaxaca"              "Puebla"
## [22] "Queretaro"           "Quintana Roo"        "San Luis Potosi"
## [25] "Sinaloa"             "Sonora"              "Tabasco"
## [28] "Tamaulipas"          "Tlaxcala"            "Veracruz"
## [31] "Yucatan"             "Zacatecas"
```

```
##  [1] I       II      III     IV      V       VI      IX      VII     VIII
## [10] X       XI      XII     XIII    XIX     XL      XV      XVI     XVII
## [19] XVIII   XX      XXI     XXII    XXIII   XXIX    XXV     XXVI    XXVII
## [28] XXVIII  XXX     XXXI    XXXII   XXXIII  XXXIV   XXXIX   XXXV    XXXVI
## [37] XXXVII  XXXVIII XIV     XXIV
## 40 Levels: I II III IV IX V VI VII VIII X XI XII XIII XIV XIX XL XV ... XXXVIII
```

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.

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

print(d)
```

```
## # A tibble: 11,333 x 96
##    name_i~1 foto  seccion casilla dtto  dto   munic~2 edo   entidad pagina    p1
##    <chr>    <chr> <chr>   <chr>   <chr> <chr> <chr>   <chr> <chr>    <dbl> <dbl>
##  1 aguasca~ 2014~ 1       84      <NA>  I     AGUASC~ Agua~ AGUASC~    128   919
##  2 aguasca~ 2014~ 85      85      1     I     AGUASC~ Agua~ AGUASC~    129   795
##  3 aguasca~ 2014~ 45      45-A    1     I     AGUASC~ Agua~ AGUA       130   767
##  4 aguasca~ 2014~ 86      86      1     I     AGUASC~ Agua~ AGUAS      131  1243
##  5 aguasca~ 2014~ 87      87      1     I     <NA>    Agua~ 1          132   718
##  6 aguasca~ 2014~ 1       87-A    7     I     AGUASC~ Agua~ AGUAS      133   710
##  7 aguasca~ 2014~ 88      88      1     I     AGUAS   Agua~ AGUAS      134     0
##  8 aguasca~ 2014~ 89      89      1     I     AGUASC~ Agua~ AGUAS      135   764
##  9 aguasca~ 2014~ 89      89-A    7     I     AGUSCA~ Agua~ 1          136   759
## 10 aguasca~ 2014~ 89      89-B    7     I     AGS     Agua~ AGS        137   757
## # ... with 11,323 more rows, 85 more variables: p2 <dbl>, 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>, ...
```

\clearpage

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.

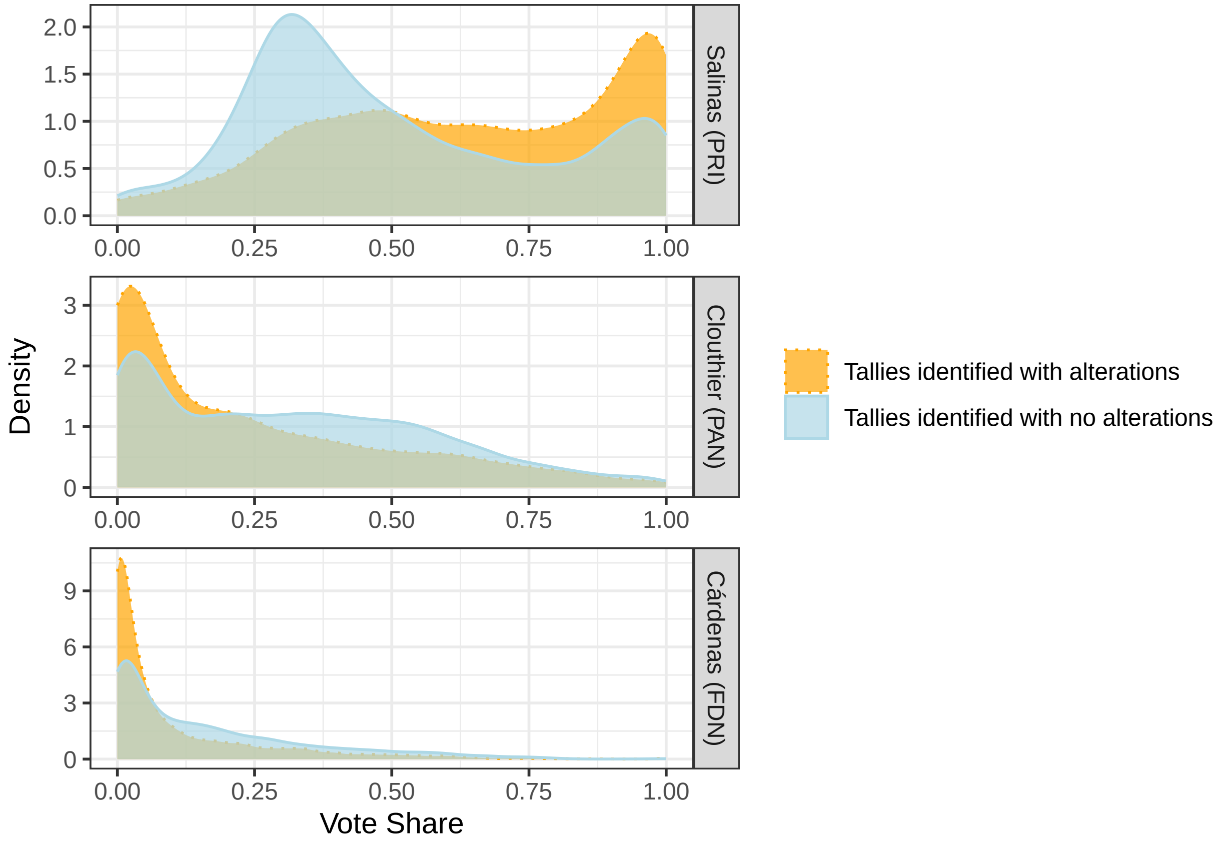
```r
#load the file and need to to scad
d <- d|>
  mutate(total_president = salinas + cardenas + clouthier + ibarra + castillo)

d <- d |>
  mutate(salinas_prop = salinas / total_president,
         cardenas_prop = cardenas / total_president,
         clouthier_prop = clouthier / total_president)

print(d)
```

```
## # A tibble: 11,333 x 100
##    name_i~1 foto  seccion casilla dtto  dto   munic~2 edo   entidad pagina    p1
##    <chr>    <chr> <chr>   <chr>   <chr> <chr> <chr>   <chr> <chr>    <dbl> <dbl>
##  1 aguasca~ 2014~ 1       84      <NA>  I     AGUASC~ Agua~ AGUASC~    128   919
##  2 aguasca~ 2014~ 85      85      1     I     AGUASC~ Agua~ AGUASC~    129   795
##  3 aguasca~ 2014~ 45      45-A    1     I     AGUASC~ Agua~ AGUA       130   767
##  4 aguasca~ 2014~ 86      86      1     I     AGUASC~ Agua~ AGUAS      131  1243
##  5 aguasca~ 2014~ 87      87      1     I     <NA>    Agua~ 1          132   718
##  6 aguasca~ 2014~ 1       87-A    7     I     AGUASC~ Agua~ AGUAS      133   710
##  7 aguasca~ 2014~ 88      88      1     I     AGUAS   Agua~ AGUAS      134     0
##  8 aguasca~ 2014~ 89      89      1     I     AGUASC~ Agua~ AGUAS      135   764
##  9 aguasca~ 2014~ 89      89-A    7     I     AGUSCA~ Agua~ 1          136   759
## 10 aguasca~ 2014~ 89      89-B    7     I     AGS     Agua~ AGS        137   757
## # ... with 11,323 more rows, 89 more variables: p2 <dbl>, 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>, ...
```

Task 4.2. Replicate Figure 4

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

```
## [1] 3
```



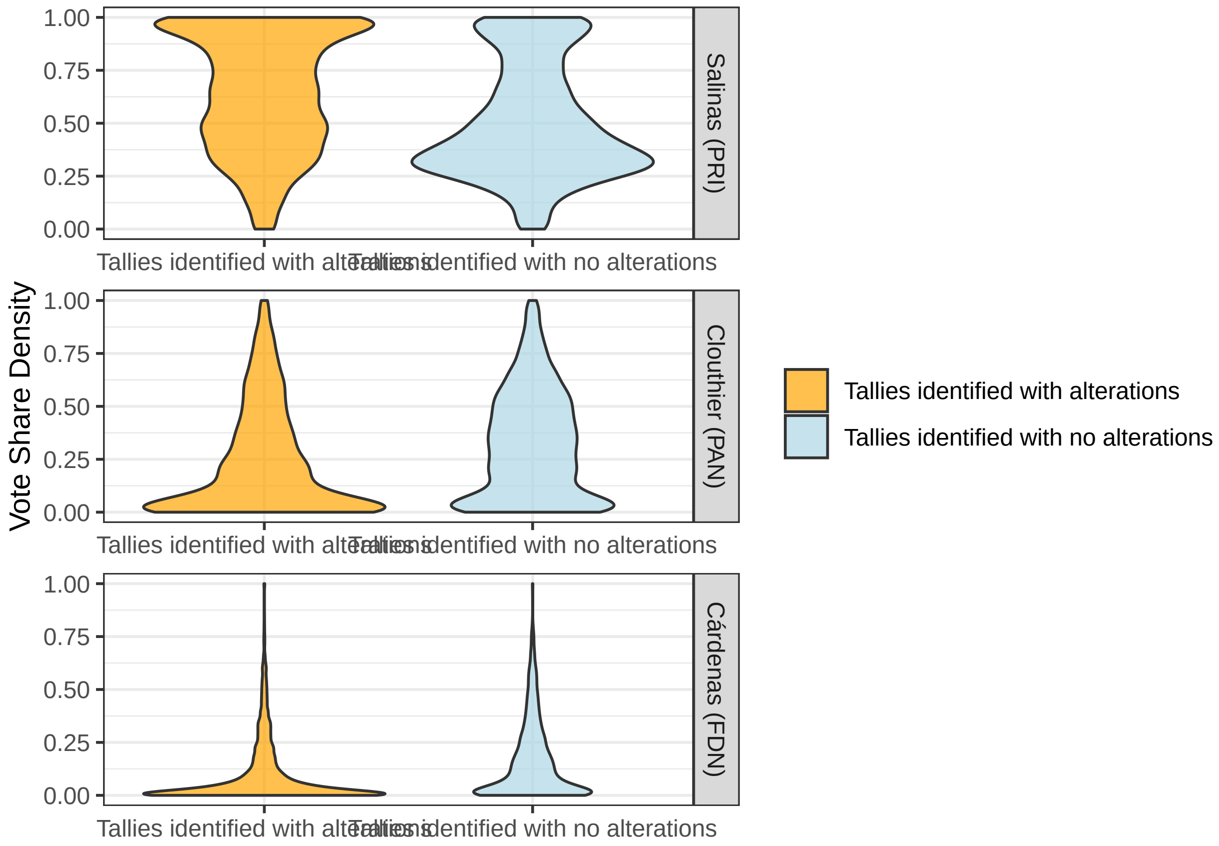
Note: Your performance in this task will be mainly evaluated based on your output's similarity with the original figure. Pay attention to the details. For your reference, below is a version created by the instructor.

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.

```r
# Alternative graph
d_figure_2 |>
  ggplot(aes(x = fraud_bin, y = prop, fill = fraud_bin)) +
  geom_violin(alpha = 0.7) +
  scale_fill_manual(name = NULL, values = c("orange", "lightblue")) +
  facet_wrap(~ president, ncol = 1, strip.position = "right", scales = "free") +
  labs(x = NULL, y = "Vote Share Density") +
  theme_bw()
```



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.

Discussion:

Figure 4 backs the researcher's claim showing clear differences in vote shares between "clean" and "fraudulent" counts for the top three contenders in Mexico's 1988 election. Salinas's vote shares in clean counts stand out against the modified ones. It signifies strong support and points towards possible voting fraud. Cárdenas's vote shares drop in fraudulent counts, this supports existing thoughts about fraud methods during one-party rule. The figure effectively shows changes in vote behaviors; enhancing the researcher's ability to visually detect potential fraud, and backing the researcher's statements about the kind of voting irregularities.

The alternative design swaps the clustering graph for violin plots, showing a fresh perspective on how vote shares are distributed. It holds the line between dishonest and honest counts, but lays more stress on the pattern and spread of distributions. The violin plots reveal how thick the data is, making it simpler to spot peaks, skew, and shifts in the vote share distributions. The new design is successful because it presents a novel view of data distribution, possibly improving the watcher's comprehension of the trends and changes in the vote shares among the candidates running for president.

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`)

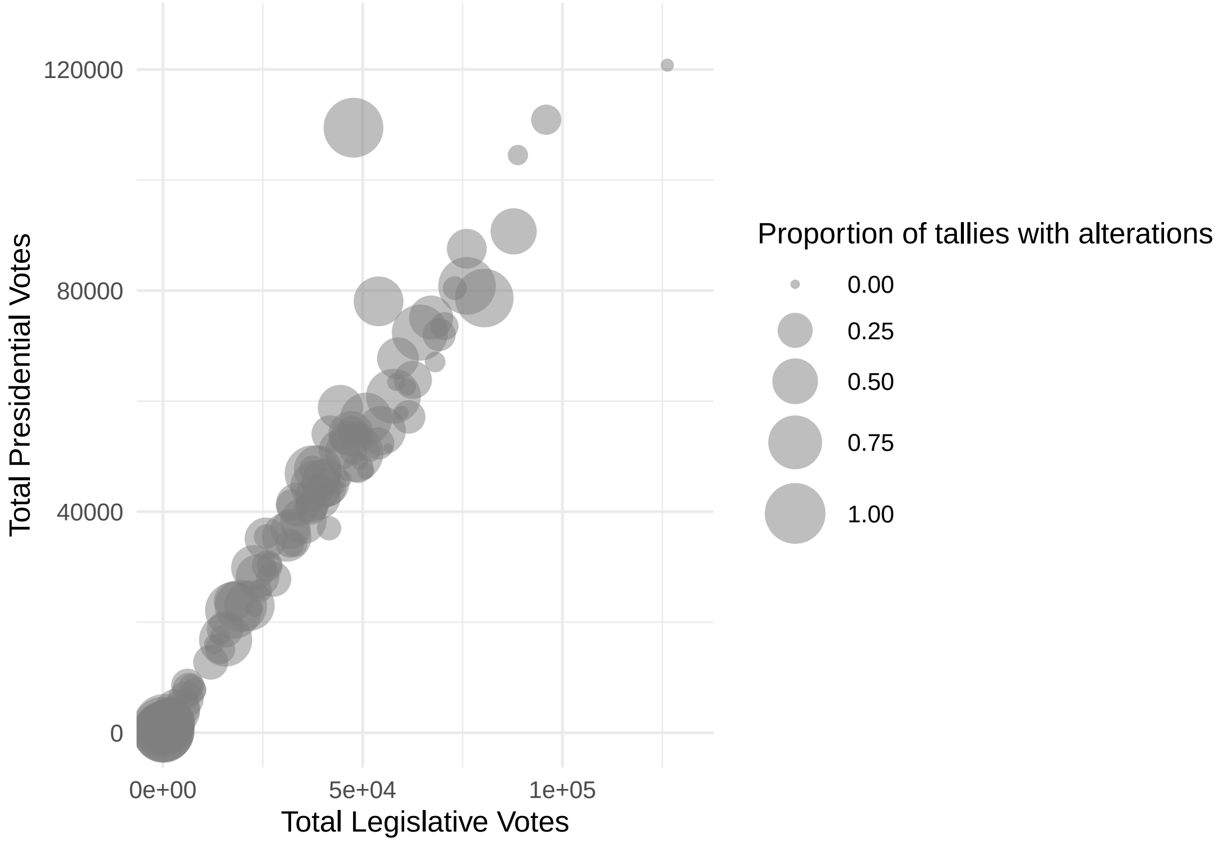
```r
sum_fraud_by_district <- d |>
  group_by(state, district) |>
  summarise(
    vote_president = sum(total_president),
    vote_legislature = sum(total),
    vote_diff = sum(total_president) - sum(total),
    prop_fraud = mean(fraud_bin, na.rm = TRUE))


print(sum_fraud_by_district)
```

```
## # A tibble: 109 x 6
## # Groups:   state [28]
##    state               district vote_president vote_legislature vote_d~1 prop_~2
##    <chr>               <fct>             <dbl>            <dbl>    <dbl>   <dbl>
##  1 Aguascalientes      I                110913            95933    14980  0.165
##  2 Aguascalientes      II                 1576             1290      286  0.6
##  3 Baja California     I                 41239            32322     8917  0.184
##  4 Baja California     II                35484            25840     9644  0.0886
##  5 Baja California     III               73577            70490     3087  0.132
##  6 Baja California     IV                 3261             2006     1255  0.125
##  7 Baja California Sur I                   625              783     -158  0.5
##  8 Baja California Sur II                30405            26641     3764  0.0940
##  9 Campeche            I                    73               73        0  1
## 10 Campeche            II                  101              101        0  1
## # ... with 99 more rows, and abbreviated variable names 1: vote_diff,
## #   2: prop_fraud
```

Task 5.2. Replicate Figure 5

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

```r
sum_fraud_by_district %>%
  ggplot(aes(x = vote_legislature, y = vote_president)) +
  geom_count(aes(size = prop_fraud), color = "grey50", alpha = 0.5) +
  labs(x = "Total Legislative Votes", y = "Total Presidential Votes",
       size = "Proportion of tallies with alterations") +
  scale_size_continuous(range = c(1, 10)) +
  theme_minimal() +
  scale_x_continuous(limits = c(0, max(sum_fraud_by_district$vote_legislature) + 5000)) +
  scale_y_continuous(limits = c(0, max(sum_fraud_by_district$vote_president) + 5000))
```



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.

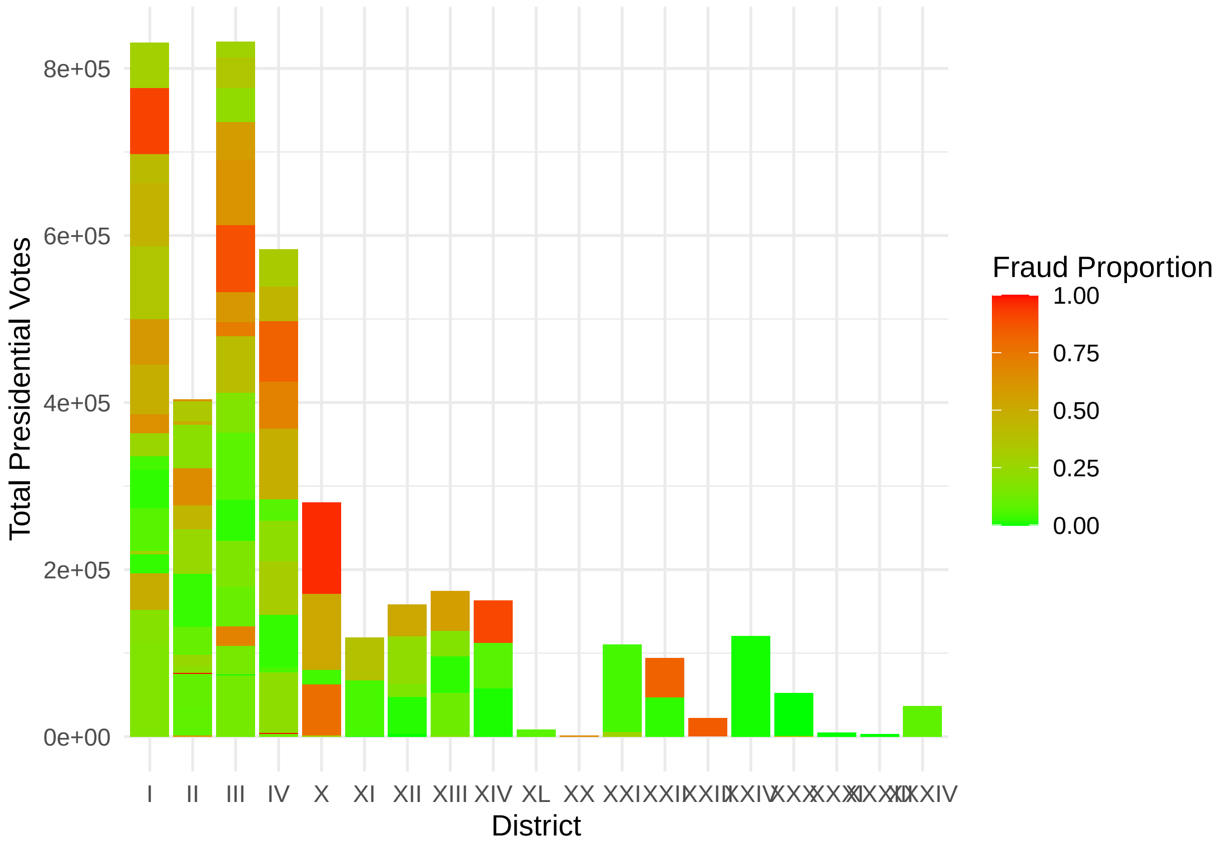
Note 2: The instructor has detected some differences between the above figure with Figure 5 on the published article. Please use the instructor's version as your main benchmark.

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.

```r
sum_fraud_by_district %>%
  ggplot(aes(x = district, y = vote_president, fill = prop_fraud)) +
  geom_bar(stat = "identity") +
  labs(x = "District", y = "Total Presidential Votes",
       fill = "Proportion of tallies with alterations") +
  scale_fill_gradient(low = "green", high = "red", name = "Fraud Proportion") +
  theme_minimal()
```



Discussion:

Figure 5 gives a clear image that backs the researcher's claim. It shows a mismatch in the vote count of both the presidential and legislative elections in 1988. Bigger dots in the figure mean more altered results in that area.

The alternative bar chart can provide visual aid as effectively as the original chart because it presents the number of votes received in the presidential election for each district in a long bar chart and uses colour coding to represent the proportion of fraudulent counts, effectively showing the variation in fraudulent proportions across districts. The colour gradient emphasises the importance of the fraudulent proportions and provides a more intuitive visual interpretation.

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.

```r
# 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: 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.

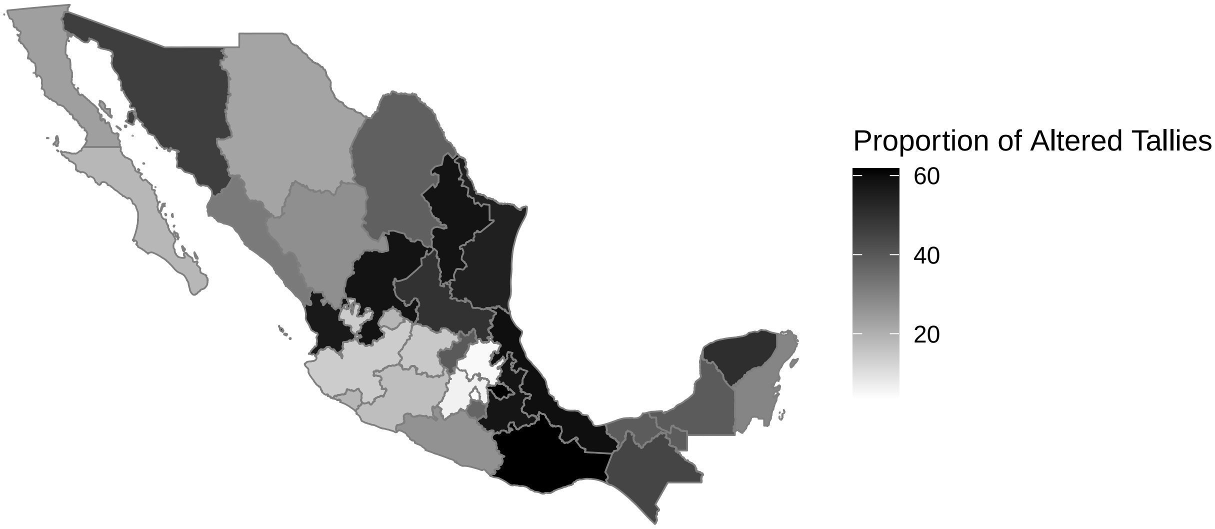
```r
# Identify items that require recoding
setdiff(unique(map_mex$state_name), unique(d_state$state))
```

```
## [1] "Ciudad de México" "México"           "Michoacán"        "Nuevo León"
## [5] "Querétaro"        "San Luis Potosí"  "Yucatán"
```

```r
setdiff(unique(d_state$state), unique(map_mex$state_name))
```

```
## [1] "Distrito Federal" "Edomex"           "Michoacan"        "Nuevo Leon"
## [5] "Queretaro"        "San Luis Potosi"  "Yucatan"
```

```r
# Recode items
d_state_count <- d_state %>%
  mutate(state = ifelse(state == "Distrito Federal", "Ciudad de México",
                        ifelse(state == "Edomex", "México",
                               ifelse(state == "Michoacan", "Michoacán",
                                      ifelse(state == "Nuevo Leon", "Nuevo León",
                                             ifelse(state == "Queretaro", "Querétaro",
                                                    ifelse(state == "San Luis Potosi",
                                                           "San Luis Potosí",
                                                           ifelse(state == "Yucatan",
                                                                  "Yucatán", state))))))))

# Join the count data with the map data
tallies_by_state <- left_join(map_mex, d_state_count, by = c("state_name" = "state"))

# Plot the map
ggplot() +
  geom_polygon(data = tallies_by_state, aes(x = long, y = lat, group = group,
                                            fill = prop_fraud), color = "gray50", linewidth = 0.3) +
  scale_fill_gradient(low = "white", high = "black") +
  theme(legend.position = "right") +
  coord_map() +
  theme_void() +
  labs(fill = "Proportion of Altered Tallies")
```


Proportion of Altered Tallies
60
40
20

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.

```r
# Identify items that require recoding
setdiff(unique(map_mex_sf$state_name), unique(d_state$state))
```

```
## NULL
```

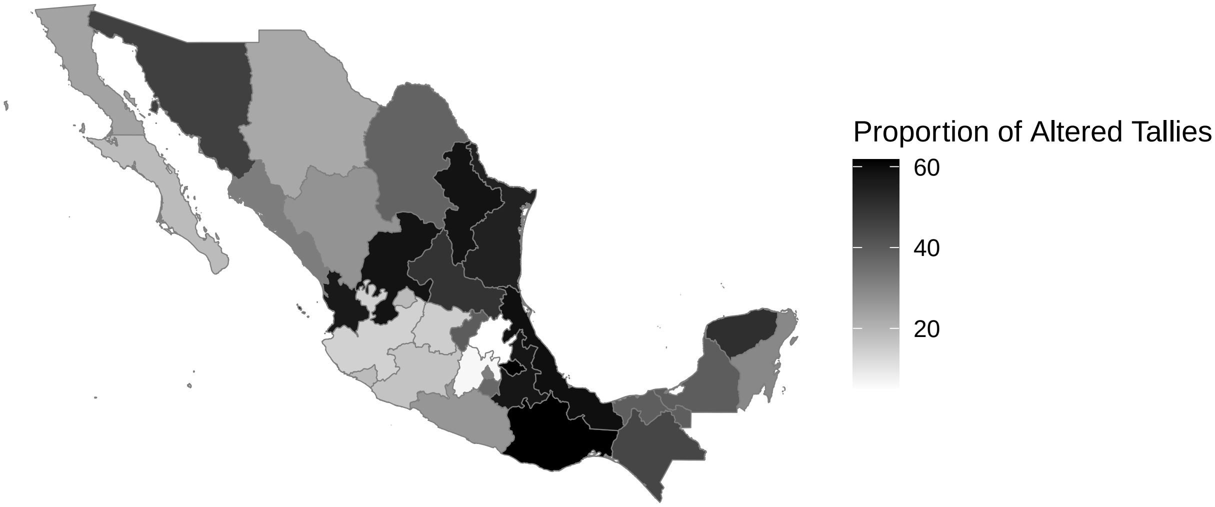
```r
setdiff(unique(d_state$state), unique(map_mex_sf$state_name))
```

```
##  [1] "Aguascalientes"      "Baja California"     "Baja California Sur"
##  [4] "Campeche"            "Chiapas"             "Chihuahua"
##  [7] "Coahuila"            "Colima"              "Distrito Federal"
## [10] "Durango"             "Edomex"              "Guanajuato"
## [13] "Guerrero"            "Hidalgo"             "Jalisco"
## [16] "Michoacan"           "Morelos"             "Nayarit"
## [19] "Nuevo Leon"          "Oaxaca"              "Puebla"
## [22] "Queretaro"           "Quintana Roo"        "San Luis Potosi"
## [25] "Sinaloa"             "Sonora"              "Tabasco"
## [28] "Tamaulipas"          "Tlaxcala"            "Veracruz"
## [31] "Yucatan"             "Zacatecas"
```

```r
# Recode items
d_state_count_sf <- d_state %>%
  mutate(state = ifelse(state == "Distrito Federal", "Ciudad de México",
                        ifelse(state == "Edomex", "México",
                               ifelse(state == "Michoacan", "Michoacán",
                                      ifelse(state == "Nuevo Leon", "Nuevo León",
                                             ifelse(state == "Queretaro", "Querétaro",
                                                    ifelse(state == "San Luis Potosi",
                                                           "San Luis Potosí",
                                                           ifelse(state == "Yucatan",
                                                                  "Yucatán", state))))))))

# Join the count data with the map data
tallies_by_state_sf <- left_join(map_mex_sf, d_state_count_sf, by = c("NAME_1" = "state"))

names(tallies_by_state_sf)
```

```
##  [1] "GID_0"      "NAME_0"     "GID_1"      "NAME_1"     "VARNAME_1"
##  [6] "NL_NAME_1"  "TYPE_1"     "ENGTYPE_1"  "CC_1"       "HASC_1"
## [11] "n_fraud"    "prop_fraud" "geometry"
```

```r
# Plot the map
ggplot() +
  geom_sf(data = tallies_by_state_sf, aes(fill = prop_fraud),
          color = "gray50", size = 0.3) +
  scale_fill_gradient(low = "white", high = "black") +
  theme(legend.position = "right") +
  theme_void() +
  labs(fill = "Proportion of Altered Tallies")
```

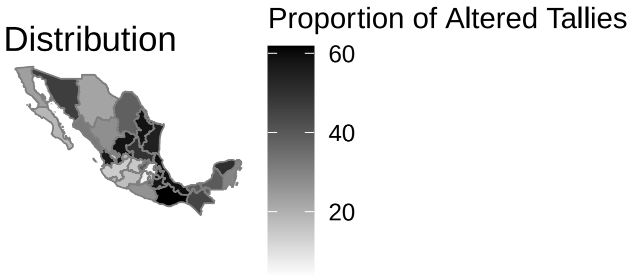


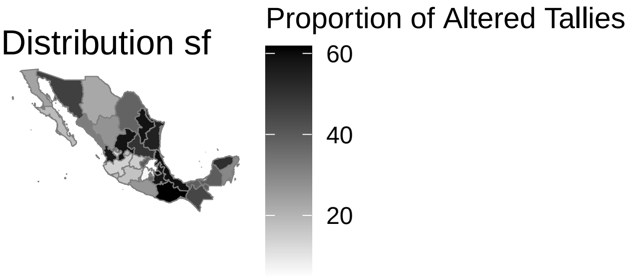
Task 6.3. Discuss and extend the reproduced figures

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.

```r
figure_3 <- ggplot() +
  geom_polygon(data = tallies_by_state, aes(x = long, y = lat, group = group,
                                            fill = prop_fraud), color = "gray50",
               linewidth = 0.3) +
  scale_fill_gradient(low = "white", high = "black") +
  theme(legend.position = "right") +
  coord_map() +
  theme_void() +
  labs(fill = "Proportion of Altered Tallies")+
   ggtitle("Distribution")

figure_4 <- ggplot() +
  geom_sf(data = tallies_by_state_sf, aes(fill = prop_fraud), color = "gray50",
          size = 0.3) +
  scale_fill_gradient(low = "white", high = "black") +
  theme(legend.position = "right") +
  theme_void() +
  labs(fill = "Proportion of Altered Tallies") +
  ggtitle("Distribution sf")

# Arrange the plots side by side
grid.arrange(figure_3, figure_4, ncol = 2)
```





Discussion:
The visual evidence provided in Figure 3 supports researchers' argument which reveals the spatial distribution of election fraud across Mexican states. The paper claims that, at state level, the map shows different rates of statistical change, with less than 3% in Mexico City and 66% in Tlaxcala. Another design compares the original map side-by-side with its equivalent in sf format providing a more detailed look of how changes are spread geographically as argued by the researcher. Thus, this visualization clearly indicates that southern part has a comparatively higher concentration of statistical changes that is consistent with research article's stance.

The alternative design shows the original map (figure_3) and a map that uses the sf format (figure_4) side by side. The major role of this design is to show any differences between the two maps. Alternatively, the alternative design helps in providing more information about electoral fraud's distribution across space as it gives a chance for an analytical approach to such data through both approaches.