Fanglin Liu, 3035770795

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.

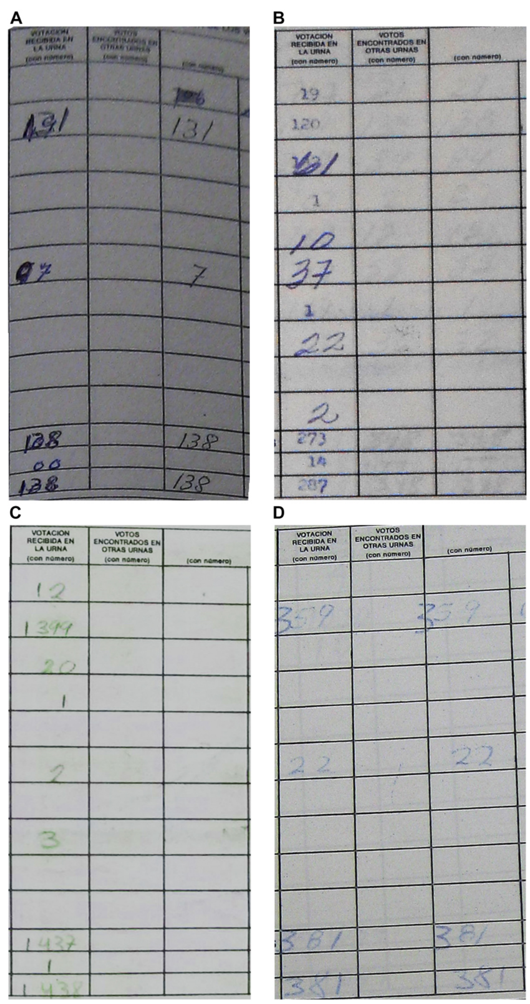


Figure 1: Examples of altered tally sheets (reproducing Figure 1 of Cantú 2018)

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.

```r
library(tidyverse)
library(ggplot2)
library(stringr)
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.

```r
d_tally <- read_csv("data/classification.txt",
  col_names = FALSE)
colnames(d_tally) <- 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]]
## # i 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.

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

d_tally$label <- as.numeric(d_tally$label)
d_tally$probability <- as.numeric(d_tally$probability)

summary(d_tally)
```

```
##   name_image            label         probability
##  Length:55334       Min.   :0.0000   Min.   :0.5000
##  Class :character   1st Qu.:0.0000   1st Qu.:0.8185
##  Mode  :character   Median :0.0000   Median :0.9710
##                     Mean   :0.3623   Mean   :0.8926
##                     3rd Qu.:1.0000   3rd Qu.:0.9996
##                     Max.   :1.0000   Max.   :1.0000
```

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.

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

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

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`.

```r
d_tally <- d_tally |>
  mutate(state = if_else(state == "Estado de Mexico", "Edomex", state))
print(d_tally)
```

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

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).

```
d_tally <- d_tally |>
  mutate(fraud_proba = if_else(label == 0, 1 - probability, probability))

d_tally <- d_tally |> select(-label, -probability)

print(d_tally)
```

```
## # A tibble: 55,334 x 4
##    name_image                               state          district  fraud_proba
##    <chr>                                    <chr>          <chr>           <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
## # i 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.

```r
d_tally <- d_tally |>
  mutate(fraud_bin = if_else(fraud_proba > 2/3, TRUE, FALSE))
print(d_tally)
```

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

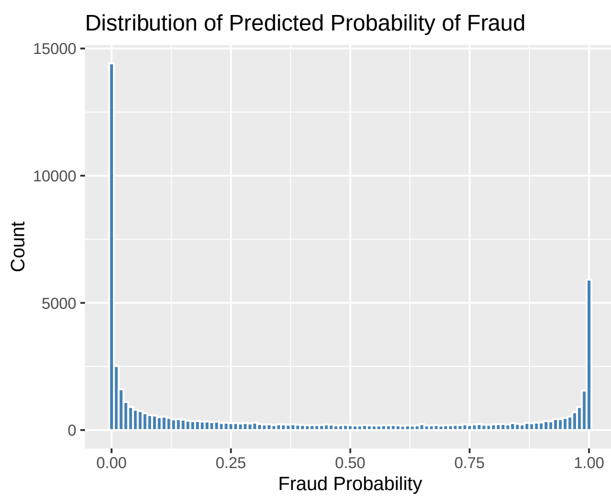
Task 2. Visualize machine classification results (3pt)

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.

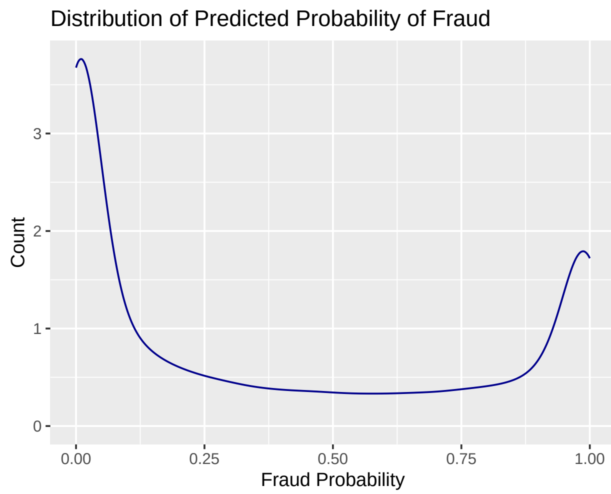
Task 2.1. Visualize distribution of `fraud_proba`

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.

```r
ggplot(d_tally, aes(fraud_proba)) +
  geom_histogram(binwidth = 0.01, fill = "steelblue", color = "white") +
  labs(title = "Distribution of Predicted Probability of Fraud",
       x = "Fraud Probability",
       y = "Count")
```




```r
ggplot(d_tally, aes(fraud_proba)) +
  geom_density(color = "blue4") +
  labs(title = "Distribution of Predicted Probability of Fraud",
       x = "Fraud Probability",
       y = "Count")
```

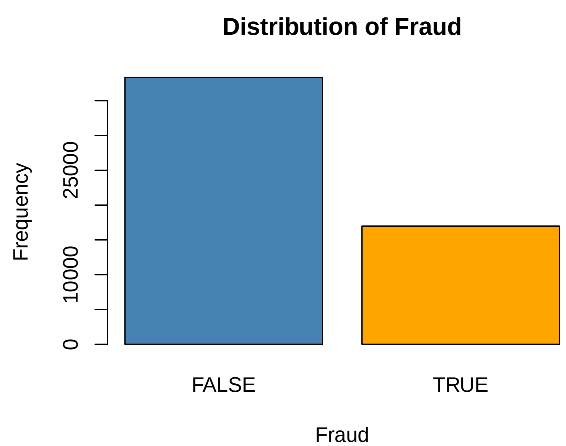



```
# The two plot shows that the machine learning is generally confident in its result,
#as most of the fraud probability is close to 0 and 1.
#Plus, votes identified as fraud is around half of those identified as real.
```

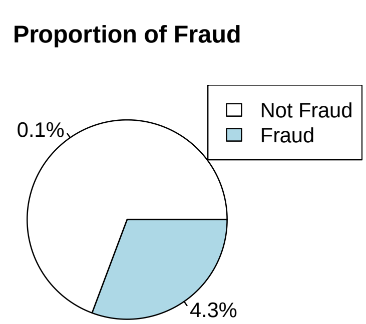
Task 2.2. Visualize distribution of `fraud_bin`

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.

```
barplot(table(d_tally$fraud_bin),
        main = "Distribution of Fraud",
        xlab = "Fraud",
        ylab = "Frequency",
        col = c("steelblue", "orange"))
```




```
# The bar plot shows the number of fraud or not with a visualized comparison on the number.


pie(table(d_tally$fraud_bin),
        main = "Proportion of Fraud",
    labels = paste0(round(d_tally$fraud_proba * 100, 1), "%"))
legend("topright",
       c("Not Fraud", "Fraud"),
       fill = c("white", "lightblue"))
```



```
# By contrast, the pie chart emphasize the proportion of the two types,
#while not showing the real number
```

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.

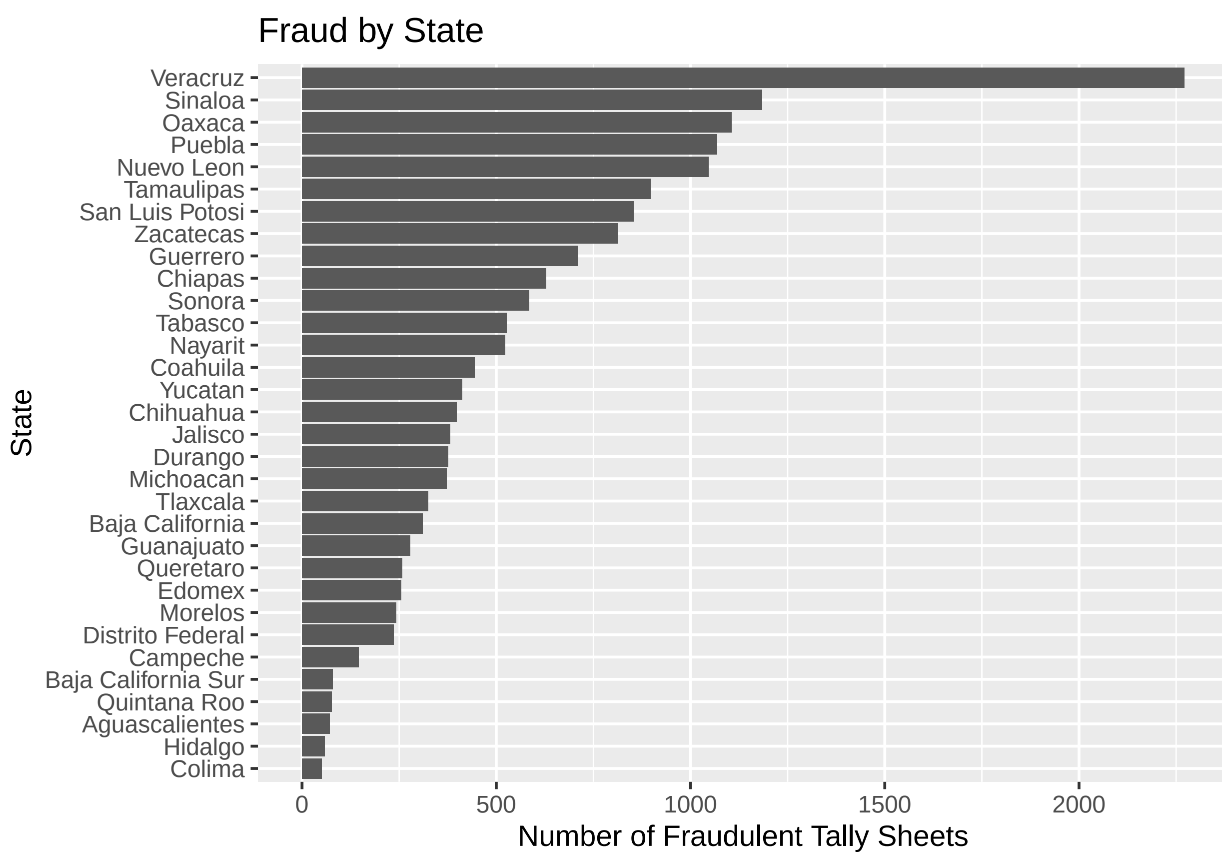
```
## # A tibble: 32 x 3
##    state               n_fraud prop_fraud
##    <chr>                 <int>      <dbl>
##  1 Aguascalientes           71      17.6
##  2 Baja California         311      23.1
##  3 Baja California Sur      79      19.1
##  4 Campeche                146      38.6
##  5 Chiapas                 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.

```r
ggplot(d_tally_state, aes(x = reorder(state, n_fraud), y = n_fraud)) +
  geom_col() +
  coord_flip()  +
  ggtitle("Fraud by State") +
  xlab("State") +
  ylab("Number of Fraudulent Tally Sheets")
```



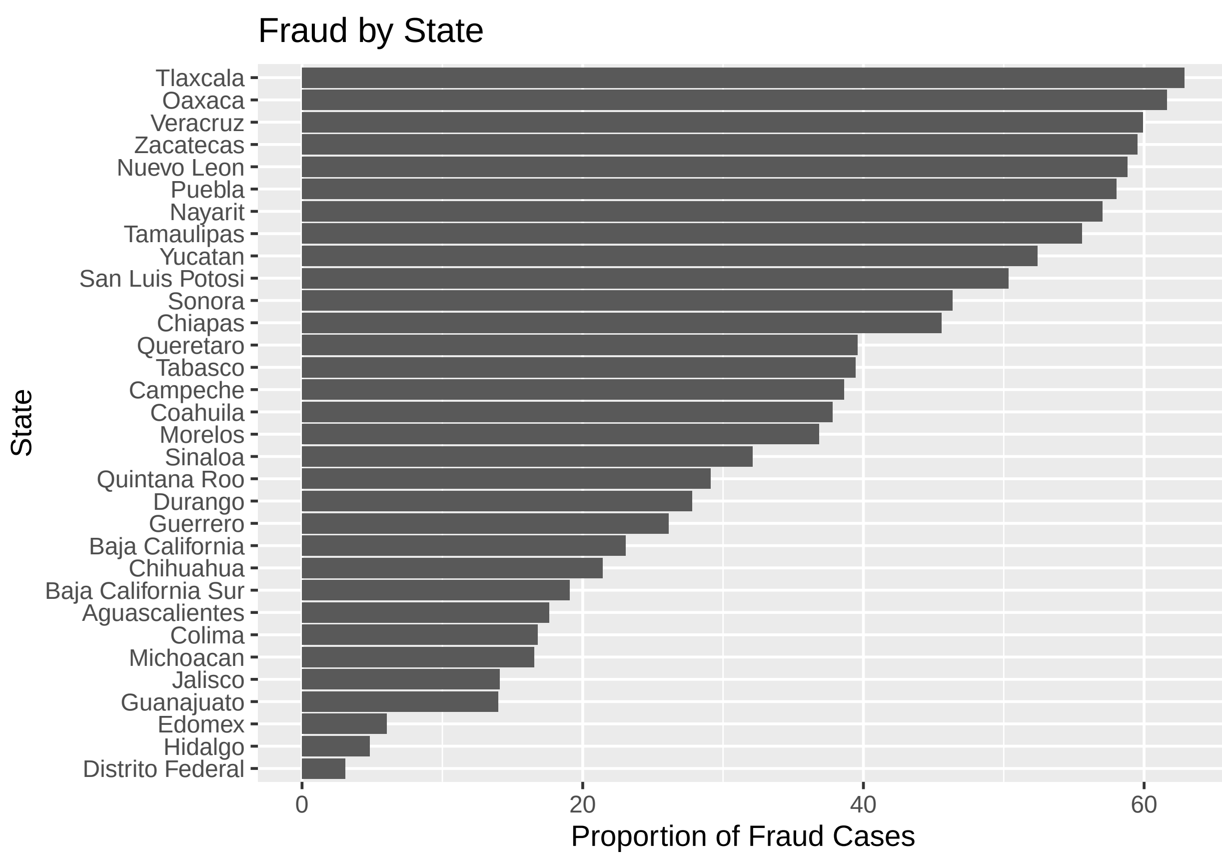

```r
# It could be found that Veracruz is the worst in fraud tally sheets, and
# significantly exceeds other states.
# Baja California Sur, Quintana Roo, Aguascalientes, Hidalgo, and
# Colima, the five states has almost no fraud sheets
```

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.

```
ggplot(d_tally_state, aes(x = reorder(state, prop_fraud), y = prop_fraud)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  ggtitle("Fraud by State") +
  xlab("State") +
  ylab("Proportion of Fraud Cases")
```

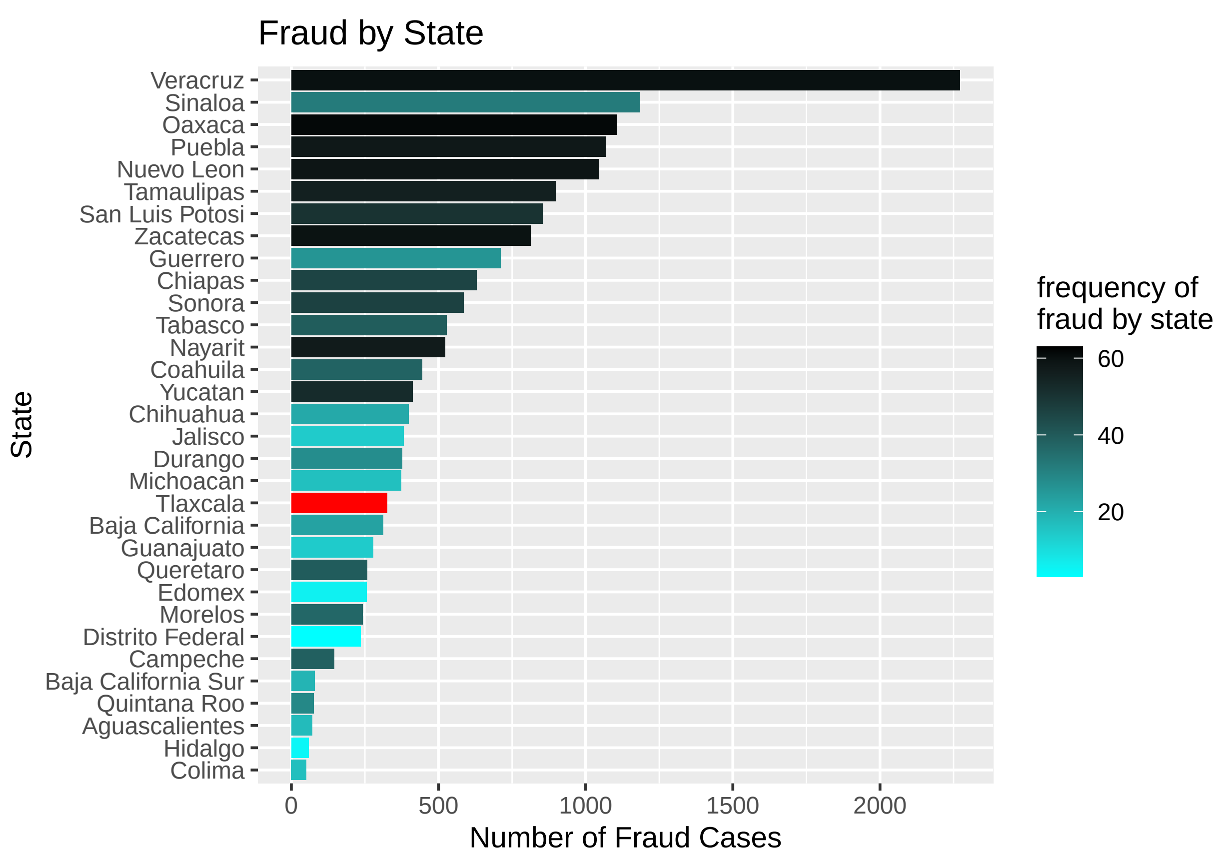



```
# Almost all states has a proportion of fraud sheets less than 60%
# And roughly 2/3 of them falls under 40%
```

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.

```
d_tally_state <- d_tally_state[order(-d_tally_state$prop_fraud), ]

plot_2 <- ggplot(d_tally_state, aes(x = reorder(state, n_fraud), y = n_fraud, fill = d_tally_state$prop_
  geom_bar(stat = "identity") +
  coord_flip() +
  scale_fill_gradient(low = "cyan", high = "black") +
  ggtitle("Fraud by State") +
  xlab("State") +
  ylab("Number of Fraud Cases") +
  labs(fill = "frequency of
fraud by state")

plot_2 + geom_bar(data = d_tally_state[1, ], stat = "identity", fill = "red")
```




```
# In my graph, the length of the bar indicates the number of fraud sheets.
# Therfore, the state with highest number, Veracruze, is put on top.
# The state with highest frequency, Tlaxala, is highlighted in red.
# Surprisingly, Tlacala hsas a relatively low number in fraud sheets.
# Veracruz is the worst state both in number and frequency.
```

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.

```
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 92 variables and 53499 entries, combing character and
#numerical data
```

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.

```
d_return$edo <- gsub("Chihuhua", "Chihuahua", d_return$edo)
d_return$edo <- gsub("Edomex", "Estado de Mexico", d_return$edo)
```

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
table(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
```

```
table(d_return$dto)
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 5976 6095 4865 4217 4942 4127 3008 2782 2524 1875  992  991  989  622  578  554
##   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
##  668  485  586  605  550  428  438  307  279  304  339  295  245  272  342  301
##   33   34   35   36   37   38   39   40  341
##  248  353  124  187  206  259  202  334    1
```

```
# convert
d_return$dto <- (as.roman(d_return$dto))
d_return$dto <- as.character(d_return$dto)
```

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.

```
d_return <- d_return %>%
  mutate(name_image = paste(d_return$edo, d_return$dto, d_return$foto, sep = "_"),
         .before = foto)
```

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`.

```
d_return$name_image <- tolower(d_return$name_image)
d_return$name_image <- sub("\\.jpg$", "", d_return$name_image)

d_tally$name_image <- tolower(d_tally$name_image)
d_tally$name_image <- sub("\\.jpg$", "", d_tally$name_image)
```

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?

```
# in d_return not in d_tally
atj_d_return <- anti_join(d_return, d_tally, by = "name_image")
table(atj_d_return$dto)
```

```
##
##  CCCXLI       I      II     III      IV      IX       V      VI     VII    VIII
##       1      39      24      16      24       8      16      12       8       7
##       X      XI     XII    XIII     XIX      XV     XVI    XVII   XVIII      XX
##       2       3       3       3       2       3       2       2       2      13
##     XXI    XXII   XXIII    XXVI   XXVII  XXVIII   XXXII  XXXIII   XXXIV   XXXIX
##       2       1       3       2       1       1       1       1       1       1
##  XXXVII XXXVIII
##       1       1
```

```
table(atj_d_return$edo)
```

```
##
##     Aguascalientes Baja California Sur           Campeche            Chiapas
##                  4                   1                  1                  9
##          Chihuahua            Coahuila             Colima   Distrito Federal
##                  7                   1                  2                 27
##            Durango    Estado de Mexico         Guanajuato           Guerrero
##                  1                  22                  6                  7
##            Hidalgo             Jalisco          Michoacan            Morelos
##                  2                   3                  5                  4
##            Nayarit          Nuevo Leon             Oaxaca             Puebla
##                  4                   5                  7                  9
##          Queretaro        Quintana Roo    San Luis Potosi            Sinaloa
##                 16                   4                  7                 18
##             Sonora             Tabasco         Tamaulipas           Tlaxcala
##                  1                   7                  3                  3
##           Veracruz             Yucatan         Zacatecas
##                 20                   1                  3
```

```
# in d_tally not in d_return
atj_d_tally <- anti_join(d_tally, d_return, by = "name_image")
table(atj_d_tally$district)
```

```
##
##       I      II     III      IV      IX       V      VI     VII    VIII       X
##     292     326     221     332      96     179     137     266     182      31
##      XI     XII    XIII     XIV     XIX      XL      XV     XVI    XVII   XVIII
```

```
##       27       27       19        9        6       32       19       18        7        8
##       XX      XXI     XXII    XXIII     XXIX      XXV     XXVI    XXVII   XXVIII      XXX
##       11       40        6       12        1        8       17        8        1        2
##     XXXI    XXXII   XXXIII    XXXIV    XXXIX     XXXV    XXXVI   XXXVII  XXXVIII
##        1        2        1        3        1        1       11        5        3
```

```r
table(atj_d_tally$state)
```

```
##
##      Aguascalientes     Baja California Baja California Sur            Campeche
##                   6                  17                  25                   5
##             Chiapas           Chihuahua            Coahuila              Colima
##                  88                  82                   9                   8
##    Distrito Federal             Durango              Edomex          Guanajuato
##                 193                   7                  32                  28
##            Guerrero             Hidalgo             Jalisco           Michoacan
##                  84                 191                  34                  35
##             Morelos             Nayarit          Nuevo Leon              Oaxaca
##                  16                  87                 184                  45
##              Puebla           Queretaro        Quintana Roo     San Luis Potosi
##                  73                  26                   2                  36
##             Sinaloa              Sonora             Tabasco          Tamaulipas
##                 252                  79                 276                  61
##            Tlaxcala            Veracruz             Yucatan           Zacatecas
##                 164                 191                   7                  25
```

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.

```r
d <- merge(d_tally, d_return, by = "name_image")
```

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:

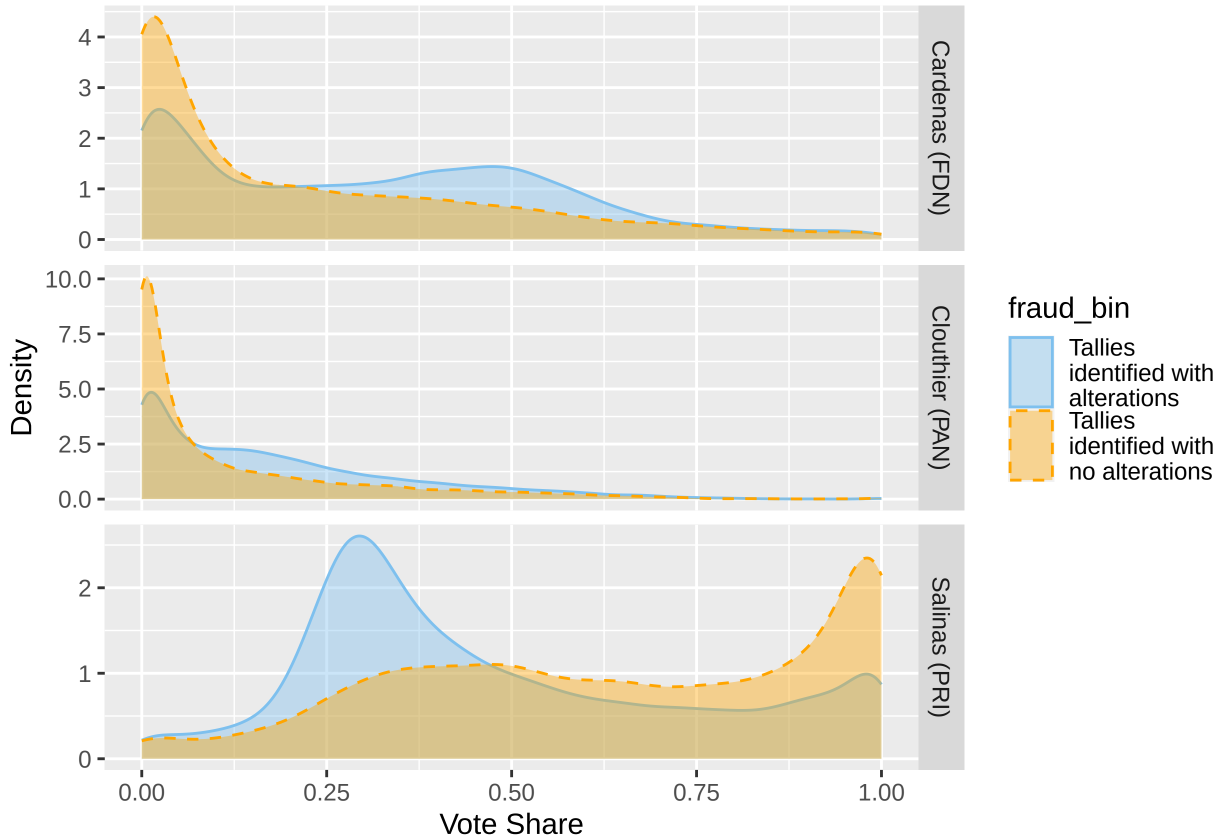
- Create a new column named `total_president` as an indicator of the total number of votes of the 5 presidential candidates.
- Create three columns `salinas_prop`, `cardenas_prop`, and `clouthier_prop` that indicate the proportions of the votes these three candidates receive respectively.

```
d$total_president <- d$ibarra + d$castillo + d$salinas + d$cardenas + d$clouthier

d$salinas_prop <- d$salinas / d$total_president
d$cardenas_prop <- d$cardenas / d$total_president
d$clouthier_prop <- d$clouthier / d$total_president
```

Task 4.2. Replicate Figure 4

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

```
d_gathered <- d %>%
  gather(key = "candidates", value = "candidate_prop", salinas_prop, cardenas_prop, clouthier_prop)  %>%
  mutate(candidate = case_when(
    candidates == "salinas_prop" ~ "Salinas (PRI)",
    candidates == "cardenas_prop" ~ "Cardenas (FDN)",
    candidates == "clouthier_prop" ~ "Clouthier (PAN)",
    TRUE ~ candidates
  ))

ggplot(d_gathered, aes(x = candidate_prop, fill = fraud_bin, linetype = fraud_bin, linecolor = fraud_bin
    geom_density(alpha = 0.4) +
  scale_fill_manual(values = c("skyblue2","orange"), labels = c("Tallies
identified with
alterations", "Tallies
identified with
no alterations")) +
  scale_color_manual(values = c("skyblue2", "orange"), labels = c("Tallies
identified with
alterations", "Tallies
identified with
no alterations")) +
  scale_linetype_manual(values = c("solid", "dashed"), labels = c("Tallies
identified with
alterations", "Tallies
identified with
no alterations")) +
  facet_wrap(~ candidate, ncol = 1,
             strip.position = "right",
             scales = "free_y") +
  xlab("Vote Share") +
  ylab("Density")
```




```r
labs(fill = "", color = "", linetype = "", linecolor = "")
```

```
## $fill
## [1] ""
##
## $colour
## [1] ""
##
## $linetype
## [1] ""
##
## $linecolour
## [1] ""
##
## attr(,"class")
## [1] "labels"
```

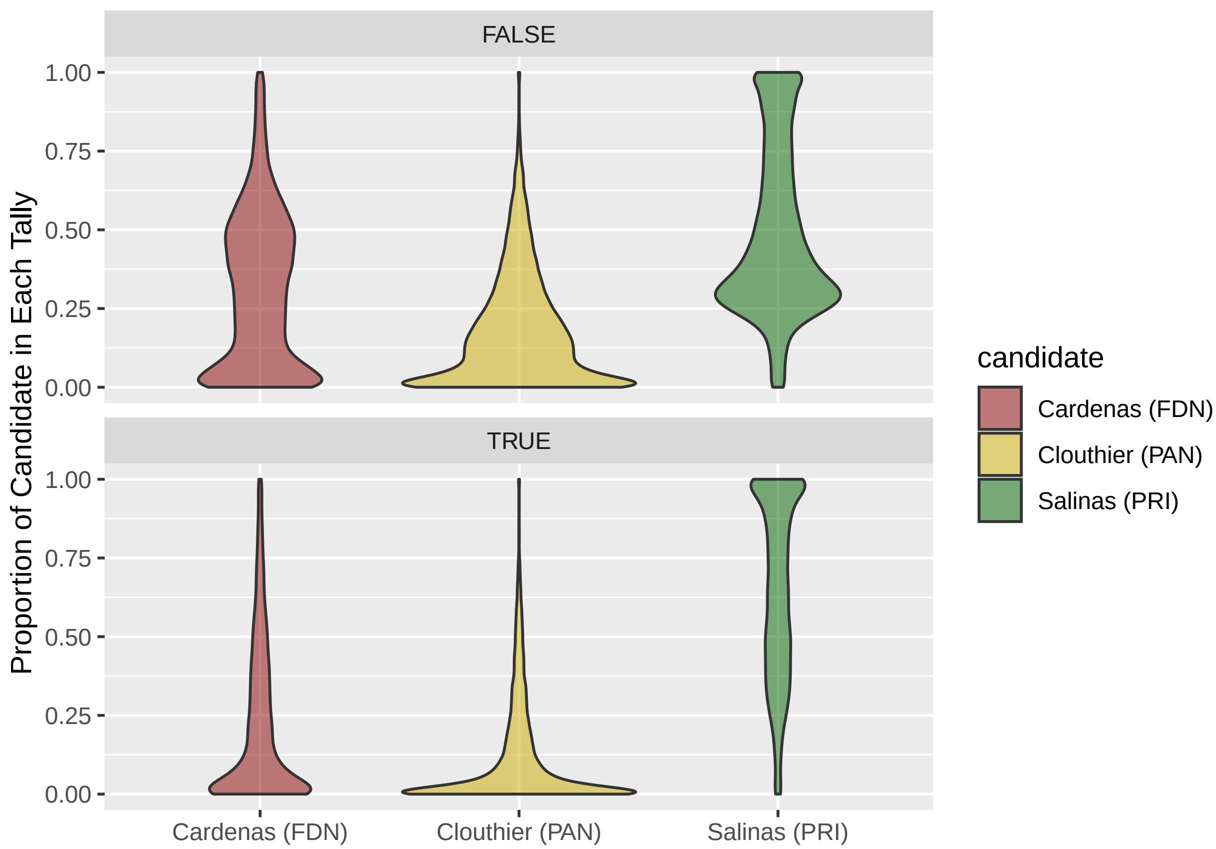
Note: Your performance in this task will be mainly evaluated based on your output's similarity with the original figure. Pay attention to the details. For your reference, below is a version created by the instructor.

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
# The image indicates that for tallies where the Salina,
# leader of PRI has a close to 100% vote share, there is
# significantly higher frequencies of altered tallies
# (shown by the blue line) compared to the otehr two candidates,
# indicating a suspicious situation. This observation could be
# overlooked if only focus on the clean tallies, where Salinas
# did has high vote share.

ggplot(d_gathered, aes(x = candidate, y = candidate_prop, fill = candidate)) +
    geom_violin(alpha = 0.5) +
  scale_fill_manual(values = c("darkred","gold3","darkgreen")) +
  facet_wrap(~ fraud_bin, ncol = 1) +
  xlab(NULL) +
  ylab("Proportion of Candidate in Each Tally")
```



```
# To better illustrate the conclusion emphasized by the author, I facet the
# graph by fraud_bin instead of candidate. This would allow us to directly
# compare the porportion of each candidate in tallies identified with false.
# In the upper graph, it is more evident that Salinas has significantly more
# false tally for those supports her in a large proportion.
```

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`)

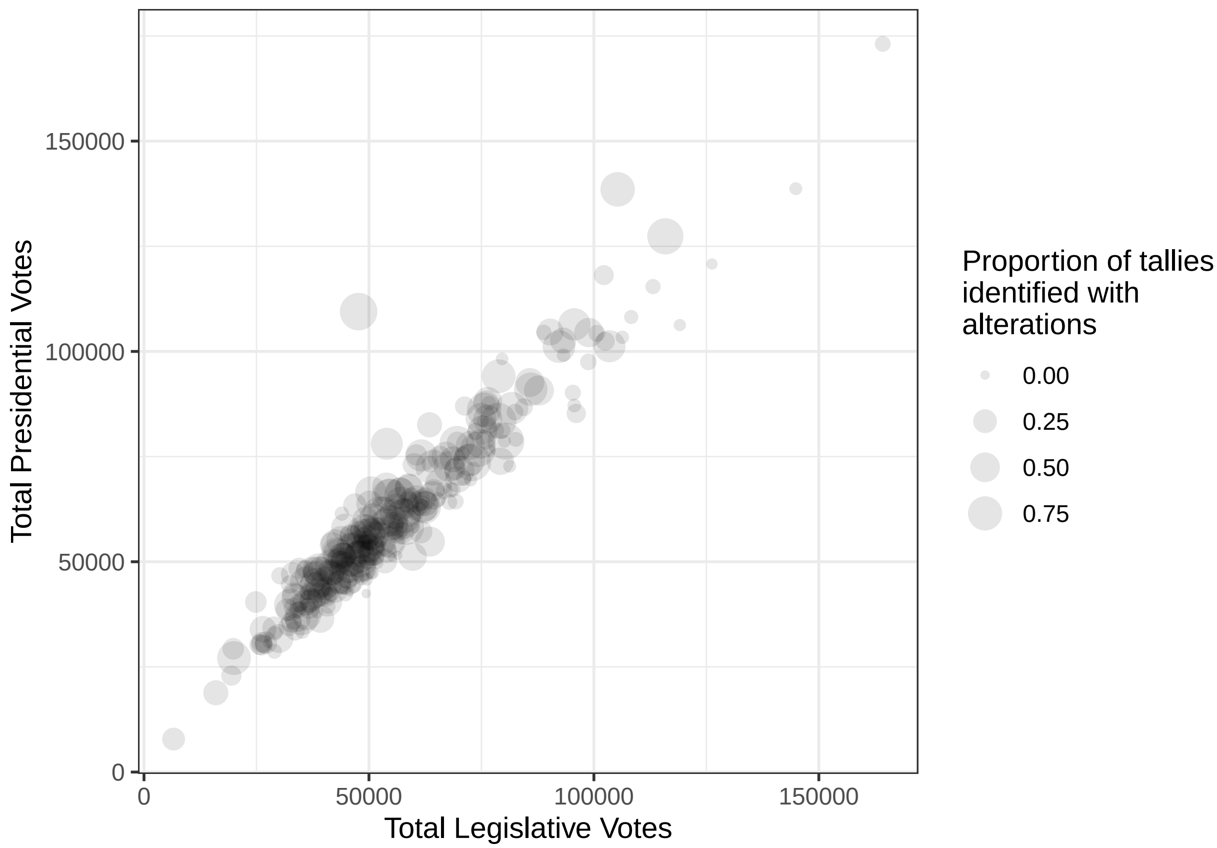
```
sum_fraud_by_district <- d |>
  group_by(state, district) |>
  summarise(vote_president = sum(total_president),
            vote_legislature = sum(total),
            prop_fraud = mean(fraud_bin == "TRUE", na.rm = TRUE)) |>
  mutate(vote_diff = vote_president - vote_legislature)

print(sum_fraud_by_district)
```

```
## # A tibble: 300 x 6
## # Groups:   state [32]
##    state           district vote_president vote_legislature prop_fraud vote_diff
##    <chr>           <chr>             <dbl>            <dbl>      <dbl>     <dbl>
##  1 Aguascalientes  I                118139           102213     0.135      15926
##  2 Aguascalientes  II                58722            55271     0.215       3451
##  3 Baja California I                 75385            60550     0.171      14835
##  4 Baja California II                44630            32429     0.0960     12201
##  5 Baja California III               79072            75940     0.132       3132
##  6 Baja California IV               104627            90270     0.375      14357
##  7 Baja California V                 55792            48971     0.152       6821
##  8 Baja California VI                64986            60596     0.368       4390
##  9 Baja Californi~ I                 52226            47569     0.259       4657
## 10 Baja Californi~ II                30405            26641     0.0933      3764
## # i 290 more rows
```

Task 5.2. Replicate Figure 5

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

```
ggplot(sum_fraud_by_district, aes(x = vote_legislature, y = vote_president, size = prop_fraud)) +
  geom_point(alpha = 0.1) +
  labs(x = "Total Legislative Votes", y = "Total Presidential Votes",
  size = "Proportion of tallies
identified with
alterations") +
  theme_bw()
```



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.

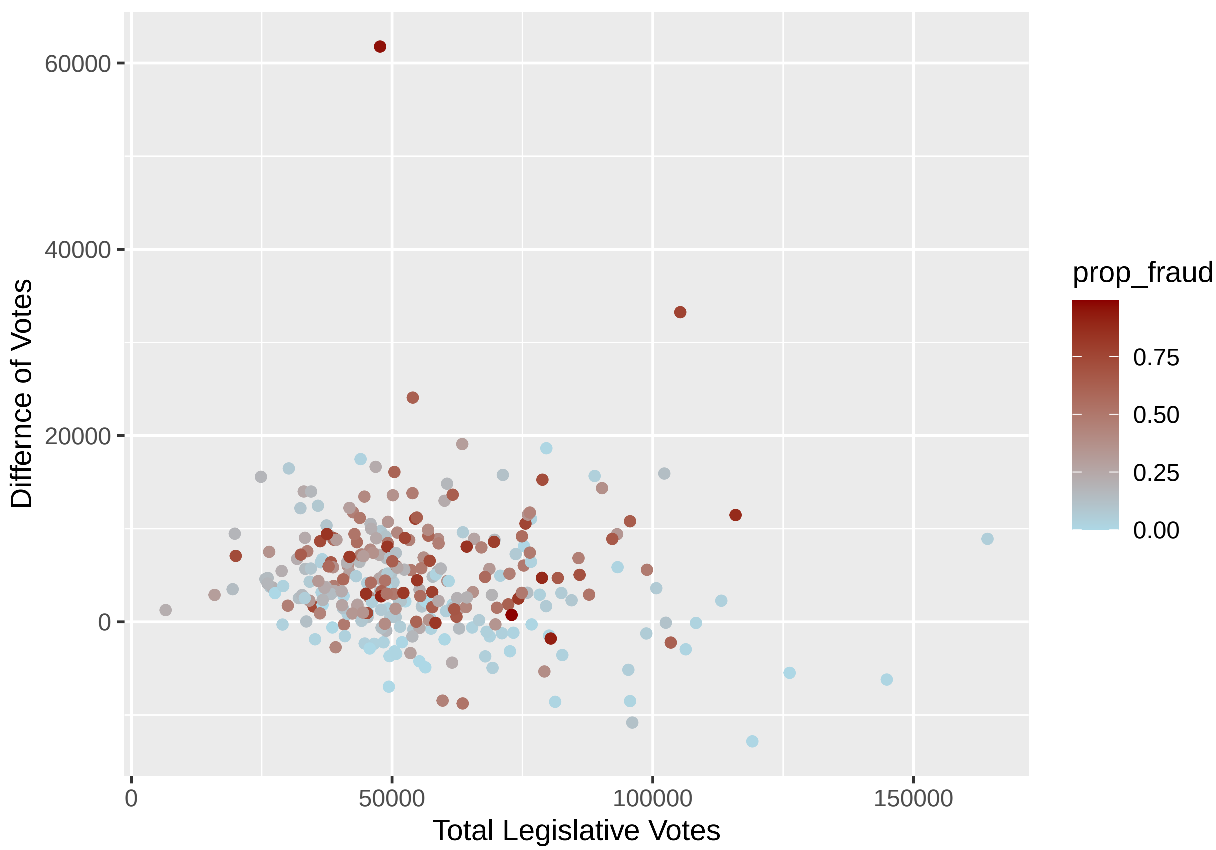
Note 2: The instructor has detected some differences between the above figure with Figure 5 on the published article. Please use the instructor's version as your main benchmark.

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.

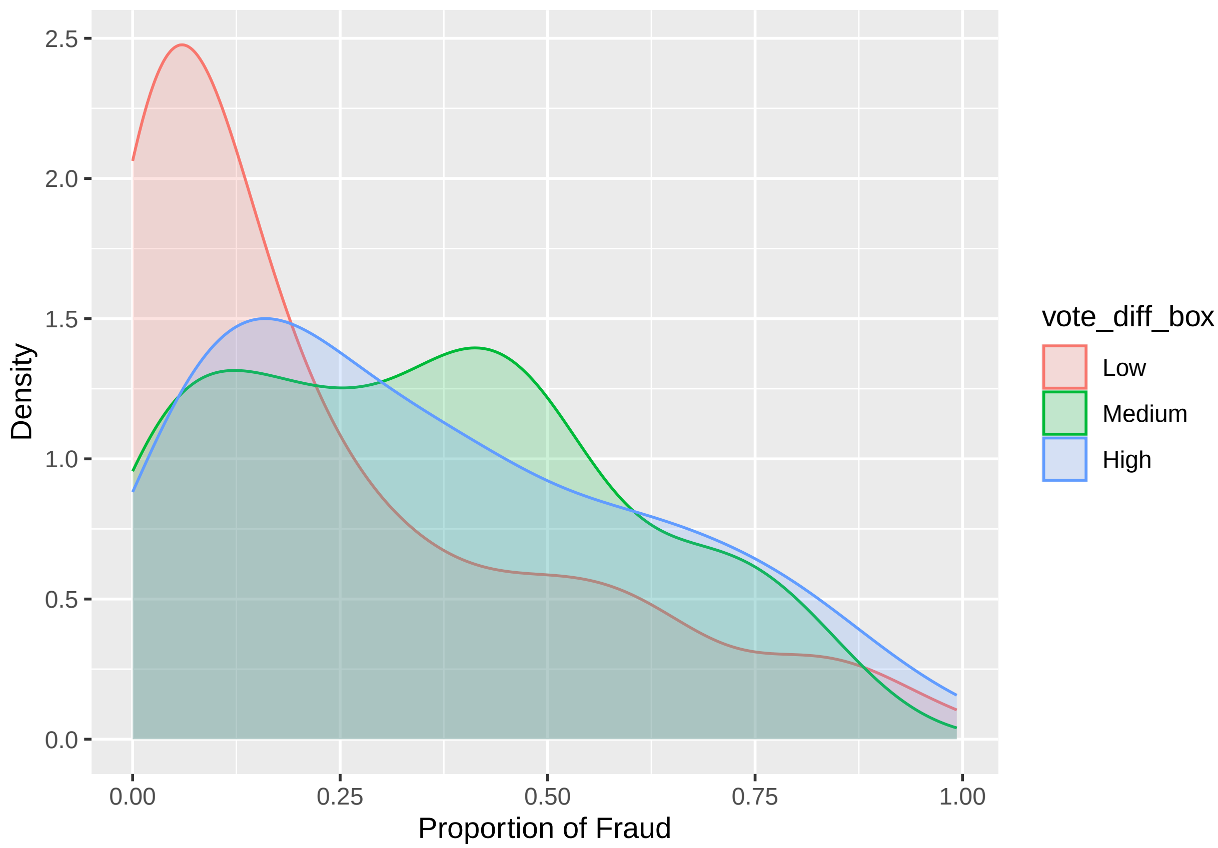
```
# The author observe and argues that for dots offset the y = x angle bisector
# line, namely a largre difference between legislative and presidential vote,
# suspiciously high proportion of altered tallies was observed.

# My plot

ggplot(sum_fraud_by_district, aes(x = vote_legislature, y = vote_diff,
                                  color = prop_fraud, na.rm = TRUE)) +
  geom_point(size = 1.5) +
  scale_color_gradient(low = "lightblue", high = "darkred") +
  labs(x = "Total Legislative Votes", y = "Differnce of Votes",
  size = "Proportion of tallies
identified with
alterations")
```



```r
# In this graph, I change the y-axis to be difference of votes. So that the
# outlier, as emphasized by the author, is more apparent.
# Plus, I change the proportion of fraud to be indicated by the color instead
# of the size of the point. Compared to the original graph,
#we could easier conclude from the graph that the proportion of fraud is mixing
# and shows no apparent pattern when the difference of vote is not exceptionally
# high.

# Plot 2

sum_fraud_by_district$vote_diff_abs <- abs(sum_fraud_by_district$vote_diff)
print(quantile(sum_fraud_by_district$vote_diff_abs, probs = seq(0, 1, 0.1), na.rm = TRUE))
```

```
##      0%     10%     20%     30%     40%     50%     60%     70%     80%     90%
##    16.0   936.8  1753.2  2647.8  3332.0  4592.0  5726.4  7135.8  8908.0 11480.4
##    100%
## 61767.0
```

```r
#Divide them into three categories by vote_diff_abs, namely low, medium, high

sum_fraud_by_district$vote_diff_box <- cut(sum_fraud_by_district$vote_diff_abs, breaks = c(0,5000,9000,6

sum_fraud_by_district_plot <- sum_fraud_by_district[!is.na(sum_fraud_by_district$vote_diff_box), ]

# My Plot 2
ggplot(sum_fraud_by_district_plot, aes(prop_fraud,
                                  color = vote_diff_box,
                                  fill = vote_diff_box)) +
  geom_density(alpha = 0.2) +
  labs(x = "Proportion of Fraud", y = "Density",
  size = "Total Presidential Votes")
```




```
#In the second plot, I categorize them based on their vote of difference after
# checking the 10% quantile. By ploting the density of proportion of fraud,
# we could conclude that lower difference in vote suggest the proportion of
# fraud is more likely to be low.
```

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 simplifies the map data to a certian degree, measured by d
```

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.

```
ggplot() +
  geom_polygon(data = map_mex, aes(x = long, y = lat, group = group))  +
  theme_void() +
  labs(title = "Rates of Tallies Classified as Altered by State",
       fill = "Proportion
of altered
tallies")
```


```
table(d_tally_state$state)
```

```
##
##      Aguascalientes     Baja California Baja California Sur            Campeche
##                   1                   1                   1                   1
```

```
##            Chiapas            Chihuahua             Coahuila               Colima
##                  1                    1                    1                    1
##   Distrito Federal              Durango               Edomex           Guanajuato
##                  1                    1                    1                    1
##           Guerrero              Hidalgo              Jalisco            Michoacan
##                  1                    1                    1                    1
##            Morelos              Nayarit           Nuevo Leon               Oaxaca
##                  1                    1                    1                    1
##             Puebla            Queretaro         Quintana Roo     San Luis Potosi
##                  1                    1                    1                    1
##            Sinaloa               Sonora              Tabasco           Tamaulipas
##                  1                    1                    1                    1
##           Tlaxcala             Veracruz              Yucatan            Zacatecas
##                  1                    1                    1                    1
```

```
table(map_mex$state_name)
```

```
##
##     Aguascalientes      Baja California Baja California Sur             Campeche
##                361                 1459                1544                  773
##            Chiapas            Chihuahua    Ciudad de México             Coahuila
##               1141                 1654                 596                 1246
##             Colima              Durango          Guanajuato             Guerrero
##                391                 1286                2127                 2218
##            Hidalgo              Jalisco              México            Michoacán
##               5721                 4544                5258                 2667
##            Morelos              Nayarit          Nuevo León               Oaxaca
##                750                 2228                 698                 1386
##             Puebla            Querétaro        Quintana Roo     San Luis Potosí
##               3851                 2172                1102                 5090
##            Sinaloa               Sonora             Tabasco           Tamaulipas
##               1329                 1546                1415                 1445
##           Tlaxcala             Veracruz             Yucatán            Zacatecas
##               1514                 4687                 315                 2668
```

```
map_mex$state_name <- gsub("Yucatán", "Yucatan", map_mex$state_name)
map_mex$state_name <- gsub("San Luis Potosí", "San Luis Potosi", map_mex$state_name)
map_mex$state_name <- gsub("Querétaro", "Queretaro", map_mex$state_name)
map_mex$state_name <- gsub("Nuevo León", "Nuevo Leon", map_mex$state_name)
map_mex$state_name <- gsub("Michoacán", "Michoacan", map_mex$state_name)
map_mex$state_name <-gsub("México", "Edomex", map_mex$state_name)
map_mex$state_name <-gsub("México", "Edomex", map_mex$state_name)
table(map_mex$state_name)
```

```
##
##     Aguascalientes      Baja California Baja California Sur             Campeche
##                361                 1459                1544                  773
##            Chiapas            Chihuahua     Ciudad de Edomex             Coahuila
##               1141                 1654                 596                 1246
##             Colima              Durango              Edomex           Guanajuato
##                391                 1286                5258                 2127
##           Guerrero              Hidalgo             Jalisco            Michoacan
##               2218                 5721                4544                 2667
```

```
##         Morelos         Nayarit      Nuevo Leon          Oaxaca
##             750            2228             698            1386
##          Puebla       Queretaro    Quintana Roo San Luis Potosi
##            3851            2172            1102            5090
##         Sinaloa          Sonora         Tabasco      Tamaulipas
##            1329            1546            1415            1445
##        Tlaxcala        Veracruz         Yucatan       Zacatecas
##            1514            4687             315            2668
```

```
map_mex$state <- map_mex$state_name

d_map_mex <- left_join(map_mex, d_tally_state, by = "state")

ggplot() +
  geom_polygon(data = d_map_mex, aes(x = long, y = lat, group = group,
                                     fill = prop_fraud))  +
  theme_void() +
   scale_fill_gradient(low = "white", high = "black") +
  labs(title = "Rates of Tallies Classified as Altered by State",
       fill = "Proportion
of altered
tallies")
```

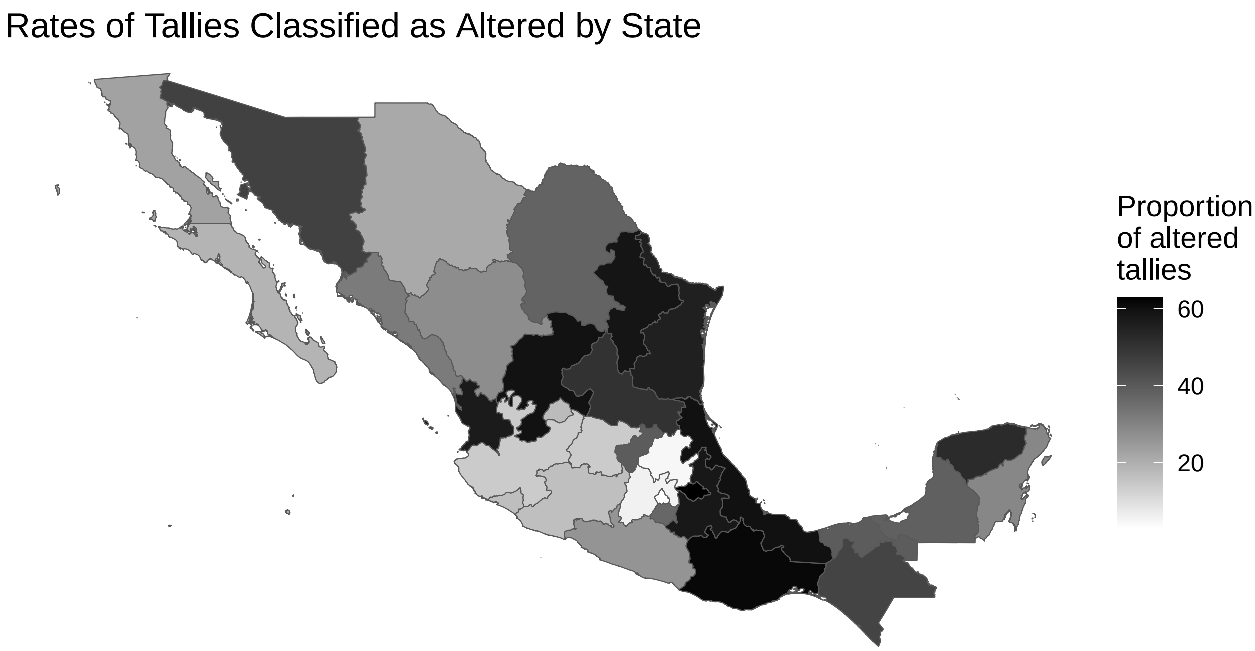
Task 6.2. Reproduce Figure 3 with `map_mex_sf`

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
map_mex_sf$NAME_1 <- gsub("Yucatán", "Yucatan", map_mex_sf$NAME_1)
map_mex_sf$NAME_1 <- gsub("San Luis Potosí", "San Luis Potosi", map_mex_sf$NAME_1)
map_mex_sf$NAME_1 <- gsub("Querétaro", "Queretaro", map_mex_sf$NAME_1)
map_mex_sf$NAME_1 <- gsub("Nuevo León", "Nuevo Leon", map_mex_sf$NAME_1)
map_mex_sf$NAME_1 <- gsub("Michoacán", "Michoacan", map_mex_sf$NAME_1)
map_mex_sf$NAME_1 <- gsub("México", "Edomex", map_mex_sf$NAME_1)

d_map_sf <- merge(d_tally_state, map_mex_sf, by.x = "state", by.y = "NAME_1")

ggplot() +
  geom_sf(data = d_map_sf, aes(geometry = geometry, fill = prop_fraud)) +
  scale_fill_gradient(low = "white", high = "black") +
  theme_void() +
  labs(title = "Rates of Tallies Classified as Altered by State",
       fill = "Proportion
of altered
tallies")
```

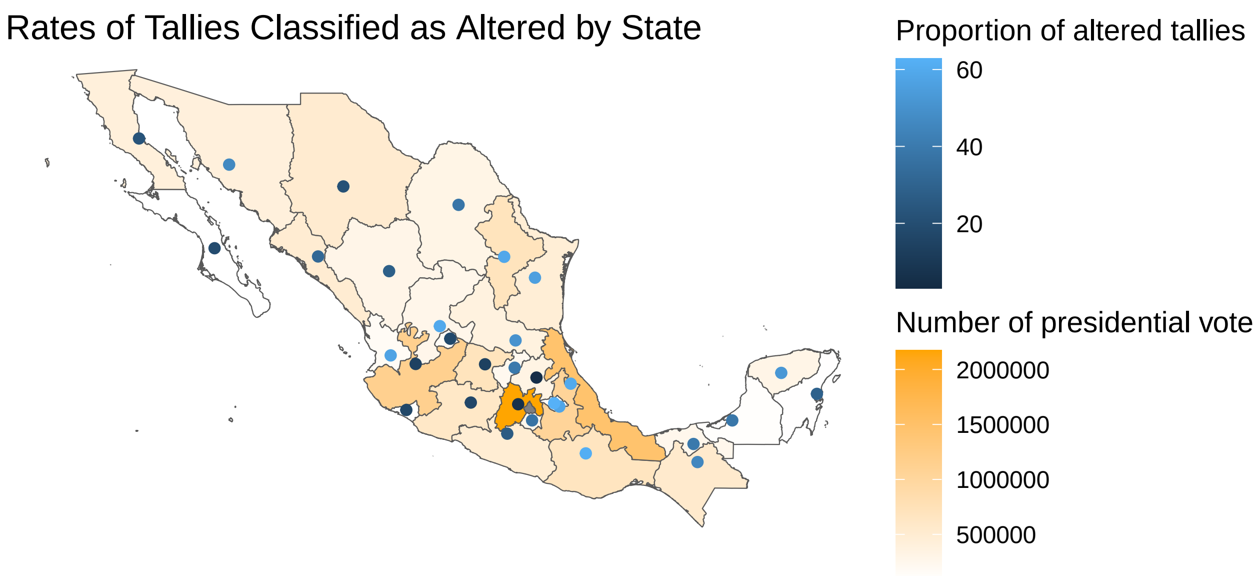

Rates of Tallies Classified as Altered by State
Proportion of altered tallies
60
40
20

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.f

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.

```
# Aligned with the author's argument, most of the tallies with alterations, illustrated by darker shade

#data preparation
sum_fraud_by_state <- sum_fraud_by_district %>%
  group_by(state) %>%
  summarise(vote_president = sum(vote_president))

state_coordinate <- map_mex |> group_by(state_name) |> summarise(lat_mean = mean(lat), long_mean = mean

sum_fraud_by_state <- merge(sum_fraud_by_state, state_coordinate, by.x = "state", by.y = "state_name")

d_map_extend <- left_join(d_map_sf, sum_fraud_by_state, by = "state")


ggplot() +
  geom_sf(data = d_map_extend, aes(geometry = geometry, fill = vote_president)) +
  geom_point(data = d_map_extend, aes(x = long_mean, y = lat_mean, color = prop_fraud)) +
    coord_sf() +
  scale_fill_gradient(low = "white", high = "orange") +
  theme_void() +
  labs(title = "Rates of Tallies Classified as Altered by State",
       fill = "Number of presidential vote",
       color = "Proportion of altered tallies")
```




```
# This plot combine the information of altered tallies and presidential vote.
# The number of votes is indicated by the fill color and the proportion of
# altered tallies is indicated by the color of point at the middle.
# It can ne seen from the map that while the northen state has relatively low
# presidential votes, accompanied by the low proportion of altered tallies.
# For the southern part, however, the proportion could be high regardless
# the number of presidential vote.
```