HU Wenxiao 3036174950

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.

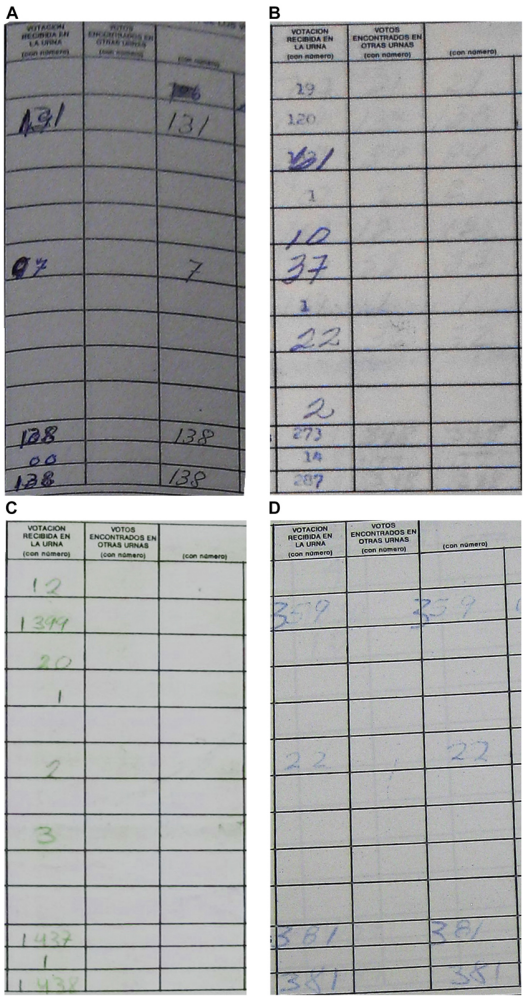


Figure 1: Examples of altered tally sheets (reproducing Figure 1 of Cantú 2018)

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.

```
library(tidyverse)
library(stringr)

library(ggplot2)
library(ggpie)
library(patchwork)
library(GGally)

library(sf)
library(ggthemes)
theme_set(theme_map())
library(cartogram)
```

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.d <- read.csv("data/classification.txt",header=FALSE) colnames(d)<-c("name_image","label","probability")

```
d_tally <- read.csv("data/classification.txt",header=FALSE)
```

```
d_tally <-d_tally |> rename("name_image"="V1","label"="V2","probability"="V3")
tibble(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
```

another way:

```
colnames(d_tally)<-c("name_image","label","probability")
```

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$label <-
  as.numeric(str_replace_all(d_tally$label, "\\[\\[(.*?)\\]\\]", "\\1"))
d_tally$probability <-
  as.numeric(str_replace_all(d_tally$probability,"\\[\\[(.*?)\\]\\]", "\\1"))
tibble(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
## # i 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.

```
d_tally<-d_tally |>
  mutate(state=str_extract(name_image,"^[^_]+"),
         district=str_match(name_image, "_(.+?)_")[,2])|>
  select(name_image,state,district,label,probability)
tibble(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
```

or using separate function:

```
d_separate <-d_tally |>
  separate(name_image,into=c("state","district"),sep="_")
d_tally <-
  bind_cols(select(d_tally,name_image),d_separate)
tibble(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`.

```
d_tally<-d_tally |>
  mutate(state=case_when(
    str_detect(state,"Estado de Mexico")~"Edomex",TRUE~state))|>
  mutate(state=ifelse(is.na(state),"Unknown",state))
tibble(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=ifelse(label==0,1-probability,probability))|>
  mutate(fraud_proba=signif(fraud_proba,digits=3))|>
  select(name_image,state,district,fraud_proba)
tibble(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         0.13
##  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         0.00000006
## 10 Aguascalientes_I_2014-05-26 00.01.15.jpg Aguascalientes I         0.36
## # 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.

```
d_tally<-d_tally |>
  mutate(fraud_bin=ifelse(fraud_proba>2/3,TRUE,FALSE))
tibble(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          0.000804   FALSE
##  2 Aguascalientes_I_2014-05-26 00.00.17.jpg Agua~ I          0.0428     FALSE
##  3 Aguascalientes_I_2014-05-26 00.00.25.jpg Agua~ I          0.423      FALSE
##  4 Aguascalientes_I_2014-05-26 00.00.31.jpg Agua~ I          0.0349     FALSE
##  5 Aguascalientes_I_2014-05-26 00.00.38.jpg Agua~ I          0.13       FALSE
##  6 Aguascalientes_I_2014-05-26 00.00.45.jpg Agua~ I          0.212      FALSE
##  7 Aguascalientes_I_2014-05-26 00.00.52.jpg Agua~ I          0.0351     FALSE
##  8 Aguascalientes_I_2014-05-26 00.00.59.jpg Agua~ I          0.319      FALSE
##  9 Aguascalientes_I_2014-05-26 00.01.06.jpg Agua~ I          0.00000006 FALSE
## 10 Aguascalientes_I_2014-05-26 00.01.15.jpg Agua~ I          0.36       FALSE
## # i 55,324 more rows
```

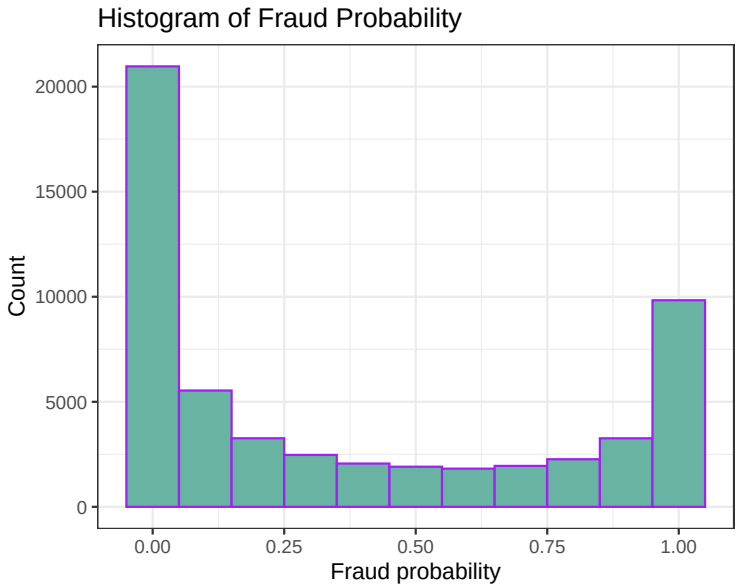
Task 2. Visualize machine classification results (3pt)

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.

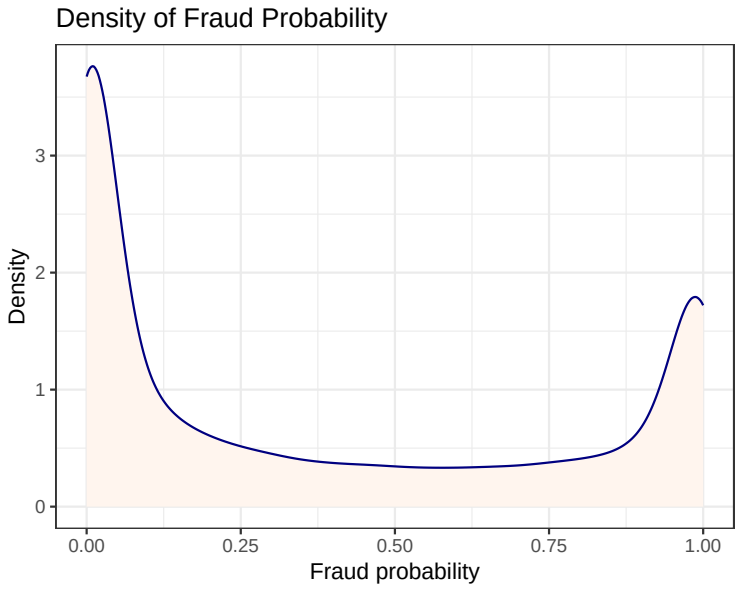
Task 2.1. Visualize distribution of `fraud_proba`

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.

```
d_tally |>
  ggplot(aes(x=fraud_proba))+
  geom_histogram(binwidth=0.1,fill="#69b3a2",color="purple")+
  theme_bw()+
  labs(x="Fraud probability",y="Count",title="Histogram of Fraud Probability")
```




```
d_tally |>
  ggplot(aes(x=fraud_proba))+
  geom_density(fill="seashell",color="navy")+
  theme_bw()+
  labs(x="Fraud probability",y="Density",title="Density of Fraud Probability")
```



Although both histograms and density plots can represent the distribution of fraud probability, I believe density plots provide a more intuitive view of the probability distribution due to the small differences between each probability.

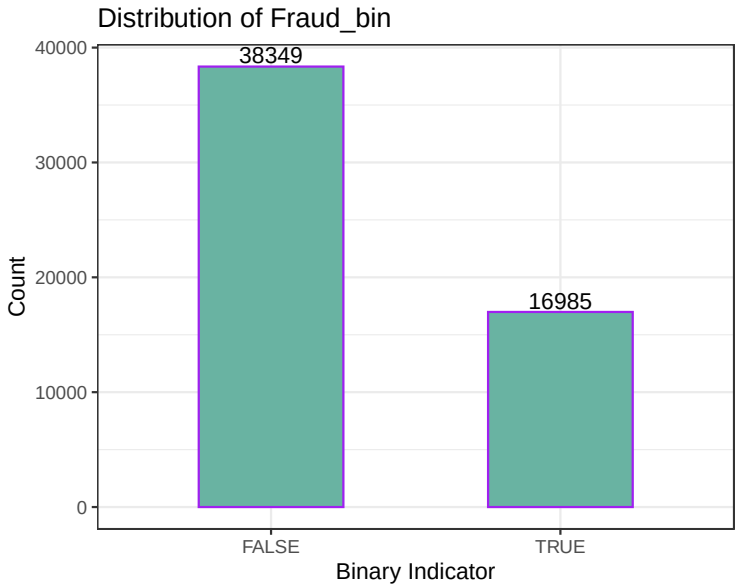
Firstly, the fraud probability between 0-0.25 has the highest frequency, with a slight increase in the graph followed by a sharp decrease at the peak. This indicates that the majority of tally sheets have a low fraud probability. The density between 0.25-0.875 is relatively small, showing a relatively flat curve. However, from 0.875-1, there is a steep increase in density, followed by a slight decrease near 1.

Task 2.2. Visualize distribution of `fraud_bin`

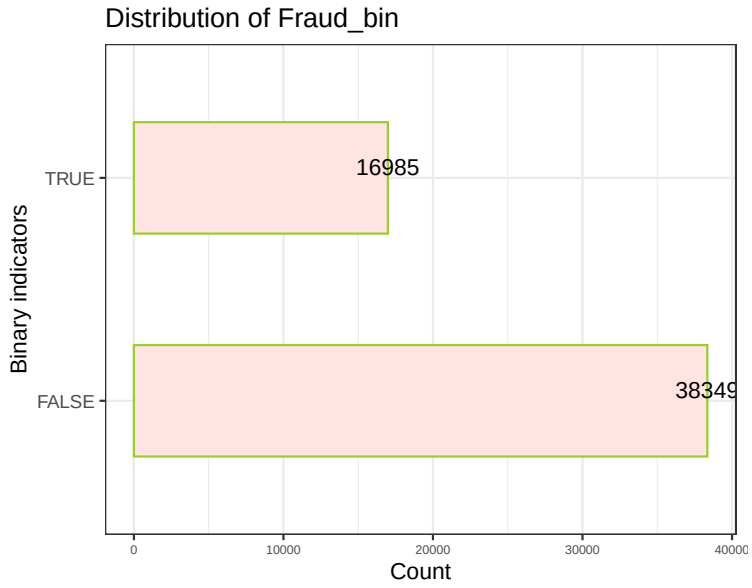
How many tally sheets are fraudulent and how many are not? We may answer this question by visualizing the binary indicator of tally-level states of fraud. Use at least two methods to visualize the distribution of `fraud_bin`. Remember to add informative labels to the figure. Describe your plots with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

```
fraud_summary <-d_tally |>
  group_by(fraud_bin)|>
  summarise(count=n())
```

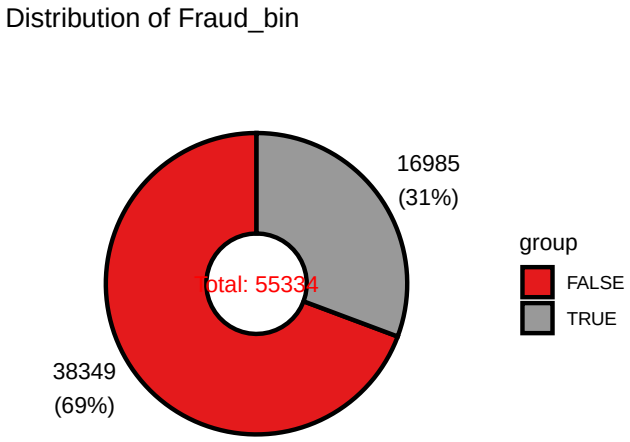
```
d_tally |>
  ggplot(aes(x=fraud_bin))+
  geom_bar(width=0.5,fill="#69b3a2",color="purple")+
  theme_bw()+
  geom_text(data=fraud_summary,aes(label=count,y=count),vjust=-0.2)+
  labs(x="Binary Indicator",y="Count",title = "Distribution of Fraud_bin")
```




```
d_tally |>
  ggplot(aes(y=fraud_bin))+
  geom_bar(width=0.5,fill="mistyrose",color="olivedrab3")+
  theme_bw()+
  geom_text(data=fraud_summary,aes(label=count,x=count),vjust=-0.2)+
  labs(y="Binary indicators", x="Count",title = "Distribution of Fraud_bin")+
  theme(axis.text.x = element_text(size = 6))
```




```
d_tally |>
 ggdonut(group_key = "fraud_bin",count_type = "full",
        label_info = "all", label_type = "horizon",
        label_size = 4, label_pos = "out")+
  labs(title = "Distribution of Fraud_bin")
```



From the above three charts, it can be seen that 16,985 tally sheets were identified as fraudulent, accounting for 31% of the total. There were 38,349 (69%) tally sheets determined to be non-fraudulent (with a probability less than or equal to 2/3).

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.

```
fraud_by_state <-d_tally |>
  group_by(state)|>
  summarize(n_fraud=sum(fraud_bin),
            prop_fraud=(n_fraud/n()* 100))
tibble(fraud_by_state)
```

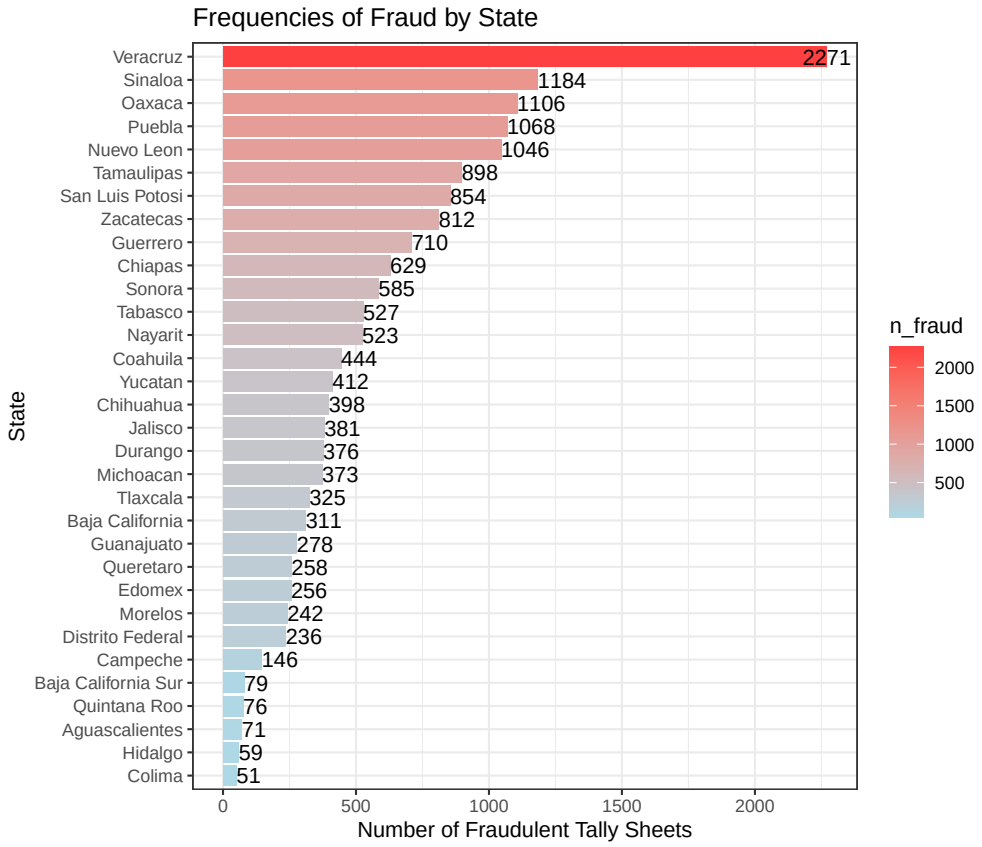
```
## # A tibble: 32 x 3
##    state               n_fraud prop_fraud
##    <chr>                 <int>      <dbl>
##  1 Aguascalientes           71      17.6
##  2 Baja California         311      23.1
##  3 Baja California Sur      79      19.1
##  4 Campeche                146      38.6
##  5 Chiapas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

```
fraud_by_state |>
  ggplot(aes(y=reorder(state,n_fraud),x=n_fraud,fill=n_fraud))+
  geom_bar(stat="identity")+
  geom_text(data=fraud_by_state,aes(label=n_fraud,y=state),hjust=ifelse(fraud_by_state$state=="Veracruz
  scale_fill_gradient(low="lightblue",high="brown1")+
  theme_bw()+
  labs(y="State",x="Number of Fraudulent Tally Sheets", title="Frequencies of Fraud by State")
```



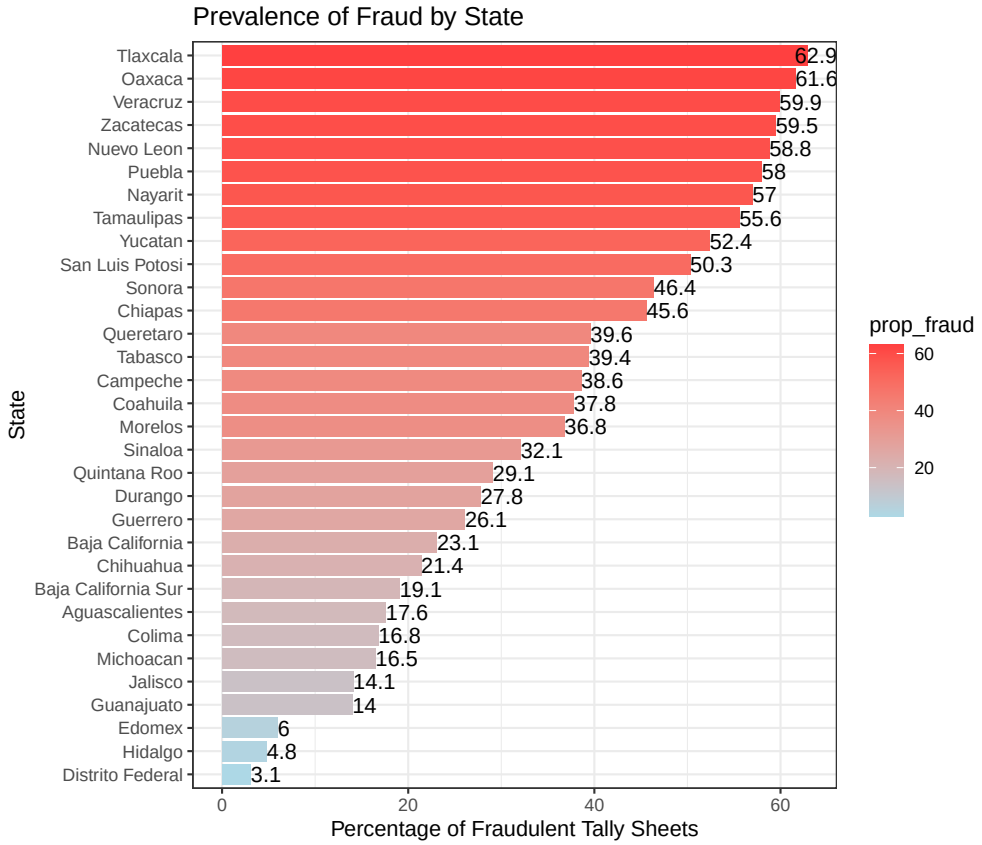
Seeing from the frequencies of tally sheets, Veracruz has the highest number of fraudulent tally sheets, which is more than twice as much as the second highest. Except for Veracruz, the difference in the number of fraudulent sheets is relatively small. State with lowest number of fraudulent is Colima, with only 51.

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.

```
fraud_by_state_round <- fraud_by_state |>
  mutate(prop_fraud=round(prop_fraud,digits=1))
fraud_by_state_round |>
  ggplot(aes(y=reorder(state,prop_fraud),x=prop_fraud,fill=prop_fraud))+
  geom_bar(stat="identity")+
  geom_text(data=fraud_by_state_round,aes(label=prop_fraud,y=state),hjust=ifelse(fraud_by_state$state==
  scale_fill_gradient(low="lightblue",high="brown1")+
  theme_bw()+
  labs(y="State",x="Percentage of Fraudulent Tally Sheets", title="Prevalence of Fraud by State")
```

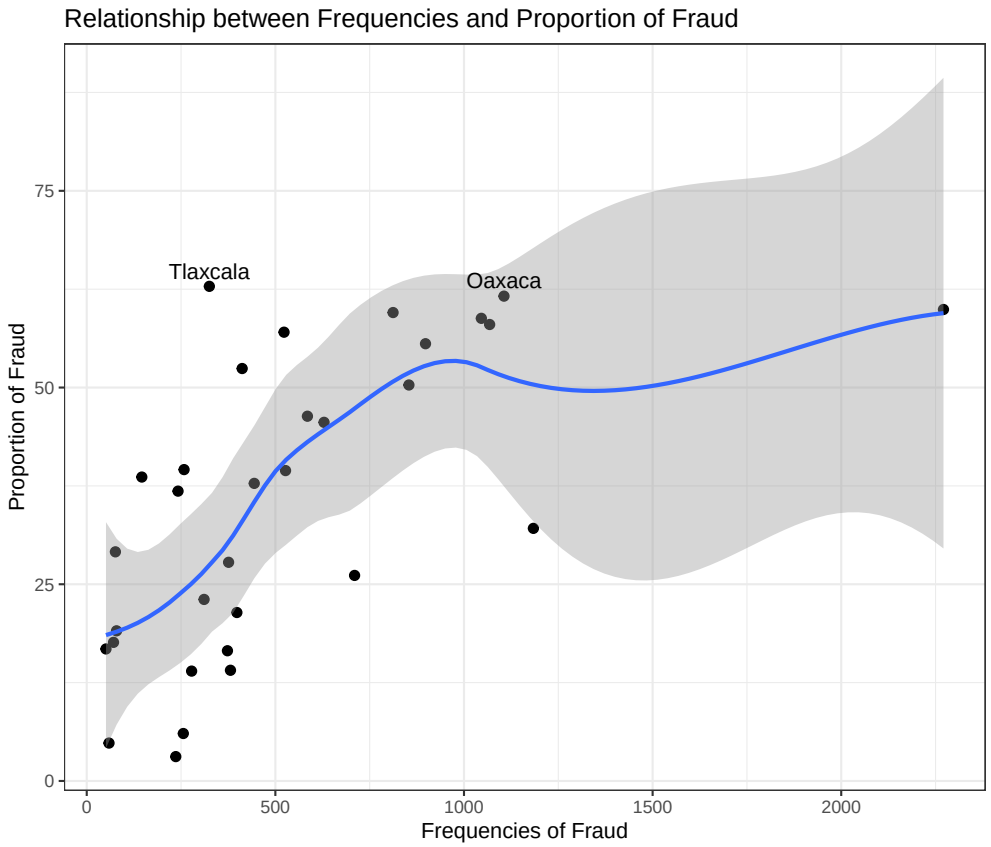


Seeing from the percentage of fraudulent tally sheets, there are 10 states with fraud rates greater than 50%, and the differences in their fraud rates are relatively small. The highest fraud rate is in Tlaxcala, and the lowest is in Distrito Federal. There are 9 states with fraud rates less than 20%

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.

```
fraud_by_state |>
  ggplot(aes(x = n_fraud, y = prop_fraud)) +
  geom_point(size=2) +
  geom_smooth()+
  geom_text(data = fraud_by_state[fraud_by_state$prop_fraud>60, ], aes(label = state), vjust = -0.5)+
  theme_bw()+
  labs(x = "Frequencies of Fraud", y = "Proportion of Fraud", title = "Relationship between Frequencies
```



As can be seen from the above figure, the curve has ups and downs but mainly shows an upward trend, indicating that the proportion of fraud and the frequency of fraud show a certain degree of correlation. As the frequency increases, the proportion has an overall upward trend. The states labelled in the plot are states with more than 60% proportion of fraud. Tlaxcala is the state with the highest proportion of fraud. I think it has the highest level of fraud because proportion is more representative than 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",na.strings = c("", "NA"))
tibble(d_return)
```

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

This is a 53499*91 data frame with blank data in it, which has been replaced with NA.

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

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

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.

```
d_tally_sum <-d_tally |>
  summarize(roman_min=min(district),
            roman_max=max(district))
d_tally_sum
```

```
##   roman_min roman_max
## 1         I   XXXVIII
```

```
sum(is.na(d_return$dto))
```

```
d_return_sum <-d_return |>
  summarize(arabic_min=min(dto,na.rm=TRUE),
            arabic_max=max(dto,na.rm=TRUE))
d_return_sum
```

```
##   arabic_min arabic_max
## 1          1        341
```

```
d_return<-d_return |>
  mutate(dto= as.roman(dto))
tibble(d_return)
```

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

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_name <-d_return |>
  mutate(name_image=paste(edo,dto,foto,sep="_"))
d_return <-
  bind_cols(select(d_return_name,name_image),d_return)
tibble(d_return)
```

```
## # A tibble: 53,499 x 92
##    name_image   foto  seccion casilla dtto  dto   municipio edo   entidad pagina
##    <chr>        <chr> <chr>   <chr>   <chr> <rom> <chr>     <chr> <chr>   <chr>
##  1 Aguascalien~ 2014~ 83      83      I     I     AGUASCAL~ Agua~ AGS     127
##  2 Aguascalien~ 2014~ 1       84      <NA>  I     AGUASCAL~ Agua~ AGUASC~ 128
##  3 Aguascalien~ 2014~ 85      85      1     I     AGUASCAL~ Agua~ AGUASC~ 129
##  4 Aguascalien~ 2014~ 45      45-A    1     I     AGUASCAL~ Agua~ AGUA    130
##  5 Aguascalien~ 2014~ 86      86      1     I     AGUASCAL~ Agua~ AGUAS   131
##  6 Aguascalien~ 2014~ 87      87      1     I     <NA>      Agua~ 1       132
##  7 Aguascalien~ 2014~ 1       87-A    7     I     AGUASCAL~ Agua~ AGUAS   133
##  8 Aguascalien~ 2014~ 88      88      1     I     AGUAS     Agua~ AGUAS   134
##  9 Aguascalien~ 2014~ 89      89      1     I     AGUASCAL~ Agua~ AGUAS   135
## 10 Aguascalien~ 2014~ 89      89-A    7     I     AGUSCALI~ Agua~ 1       136
## # i 53,489 more rows
## # i 82 more variables: p1 <int>, p2 <int>, p3 <int>, p4 <int>, p5 <int>,
## #   pan <int>, pri <int>, pps <int>, psm <int>, pms <int>, pfcrn <int>,
## #   prt <int>, parm <int>, noregis <int>, nombrenore <chr>, otros <int>,
## #   otroscan <chr>, pan2 <int>, pri2 <int>, pps2 <int>, psm2 <int>, pms2 <int>,
## #   pfcrn2 <int>, prt2 <int>, parm2 <int>, noregis2 <int>, otro2 <int>,
## #   pan3 <int>, pri3 <int>, pps3 <int>, psm3 <int>, pms3 <int>, ...
```

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_tally <-d_tally |>
  mutate(name_image=tolower(name_image))|>
  mutate(name_image=str_replace(name_image,"\\.jpg$",""))
tibble(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 Aguascal~ I          0.000804   FALSE
##  2 aguascalientes_i_2014-05-26 00.00.17 Aguascal~ I          0.0428     FALSE
##  3 aguascalientes_i_2014-05-26 00.00.25 Aguascal~ I          0.423      FALSE
##  4 aguascalientes_i_2014-05-26 00.00.31 Aguascal~ I          0.0349     FALSE
##  5 aguascalientes_i_2014-05-26 00.00.38 Aguascal~ I          0.13       FALSE
##  6 aguascalientes_i_2014-05-26 00.00.45 Aguascal~ I          0.212      FALSE
##  7 aguascalientes_i_2014-05-26 00.00.52 Aguascal~ I          0.0351     FALSE
##  8 aguascalientes_i_2014-05-26 00.00.59 Aguascal~ I          0.319      FALSE
##  9 aguascalientes_i_2014-05-26 00.01.06 Aguascal~ I          0.00000006 FALSE
## 10 aguascalientes_i_2014-05-26 00.01.15 Aguascal~ I          0.36       FALSE
## # i 55,324 more rows
```

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

```
## # A tibble: 53,499 x 92
##    name_image   foto  seccion casilla dtto  dto   municipio edo   entidad pagina
##    <chr>        <chr> <chr>   <chr>   <chr> <rom> <chr>     <chr> <chr>   <chr>
##  1 aguascalien~ 2014~ 83      83      I     I     AGUASCAL~ Agua~ AGS     127
##  2 aguascalien~ 2014~ 1       84      <NA>  I     AGUASCAL~ Agua~ AGUASC~ 128
##  3 aguascalien~ 2014~ 85      85      1     I     AGUASCAL~ Agua~ AGUASC~ 129
##  4 aguascalien~ 2014~ 45      45-A    1     I     AGUASCAL~ Agua~ AGUA    130
##  5 aguascalien~ 2014~ 86      86      1     I     AGUASCAL~ Agua~ AGUAS   131
##  6 aguascalien~ 2014~ 87      87      1     I     <NA>      Agua~ 1       132
##  7 aguascalien~ 2014~ 1       87-A    7     I     AGUASCAL~ Agua~ AGUAS   133
##  8 aguascalien~ 2014~ 88      88      1     I     AGUAS     Agua~ AGUAS   134
##  9 aguascalien~ 2014~ 89      89      1     I     AGUASCAL~ Agua~ AGUAS   135
## 10 aguascalien~ 2014~ 89      89-A    7     I     AGUSCALI~ Agua~ 1       136
## # i 53,489 more rows
## # i 82 more variables: p1 <int>, p2 <int>, p3 <int>, p4 <int>, p5 <int>,
## #   pan <int>, pri <int>, pps <int>, psm <int>, pms <int>, pfcrn <int>,
## #   prt <int>, parm <int>, noregis <int>, nombrenore <chr>, otros <int>,
## #   otroscan <chr>, pan2 <int>, pri2 <int>, pps2 <int>, psm2 <int>, pms2 <int>,
## #   pfcrn2 <int>, prt2 <int>, parm2 <int>, noregis2 <int>, otro2 <int>,
## #   pan3 <int>, pri3 <int>, pps3 <int>, psm3 <int>, pms3 <int>, ...
```

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?

```
result_return <-anti_join(d_return,d_tally,by="name_image")
tibble(state=result_return$edo,district=result_return$dto)
```

```
## # A tibble: 210 x 2
##    state               district
##    <chr>               <roman>
##  1 Aguascalientes      I
##  2 Aguascalientes      I
##  3 Aguascalientes      V
##  4 Aguascalientes      VI
##  5 Baja California Sur II
##  6 Campeche            I
##  7 Chiapas             I
##  8 Chiapas             I
##  9 Chiapas             II
## 10 Chiapas             III
## # i 200 more rows
```

There are 210 rows in `d_return` but not in `d_tally`. They are listed above.

```
result_tally <-anti_join(d_tally,d_return,by="name_image")
tibble(state=result_tally$state,district=result_tally$district)
```

```
## # A tibble: 2,368 x 2
##    state               district
##    <chr>               <chr>
##  1 Aguascalientes      I
##  2 Aguascalientes      I
##  3 Aguascalientes      I
##  4 Aguascalientes      I
##  5 Aguascalientes      II
##  6 Aguascalientes      II
##  7 Baja California Sur I
##  8 Baja California Sur I
##  9 Baja California Sur I
## 10 Baja California Sur I
## # i 2,358 more rows
```

There are 2368 rows are in `d_tally` but not in `d_return`.They are listed above

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 <- inner_join(d_tally,d_return,by="name_image")
tibble(d)
```

```
## # A tibble: 53,289 x 96
##    name_image   state district fraud_proba fraud_bin foto  seccion casilla dtto
##    <chr>        <chr> <chr>          <dbl> <lgl>     <chr> <chr>   <chr>   <chr>
##  1 aguascalien~ Agua~ I          0.000804   FALSE     2014~ 1       84      <NA>
##  2 aguascalien~ Agua~ I          0.0428     FALSE     2014~ 85      85      1
##  3 aguascalien~ Agua~ I          0.423      FALSE     2014~ 45      45-A    1
##  4 aguascalien~ Agua~ I          0.0349     FALSE     2014~ 86      86      1
##  5 aguascalien~ Agua~ I          0.13       FALSE     2014~ 87      87      1
##  6 aguascalien~ Agua~ I          0.212      FALSE     2014~ 1       87-A    7
##  7 aguascalien~ Agua~ I          0.0351     FALSE     2014~ 88      88      1
##  8 aguascalien~ Agua~ I          0.319      FALSE     2014~ 89      89      1
##  9 aguascalien~ Agua~ I          0.00000006 FALSE     2014~ 89      89-A    7
## 10 aguascalien~ Agua~ I          0.36       FALSE     2014~ 89      89-B    7
## # i 53,279 more rows
## # i 87 more variables: dto <roman>, municipio <chr>, edo <chr>, entidad <chr>,
## #   pagina <chr>, p1 <int>, p2 <int>, p3 <int>, p4 <int>, p5 <int>, pan <int>,
## #   pri <int>, pps <int>, psm <int>, pms <int>, pfcrn <int>, prt <int>,
## #   parm <int>, noregis <int>, nombrenore <chr>, otros <int>, otroscan <chr>,
## #   pan2 <int>, pri2 <int>, pps2 <int>, psm2 <int>, pms2 <int>, pfcrn2 <int>,
## #   prt2 <int>, parm2 <int>, noregis2 <int>, otro2 <int>, pan3 <int>, ...
```

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.

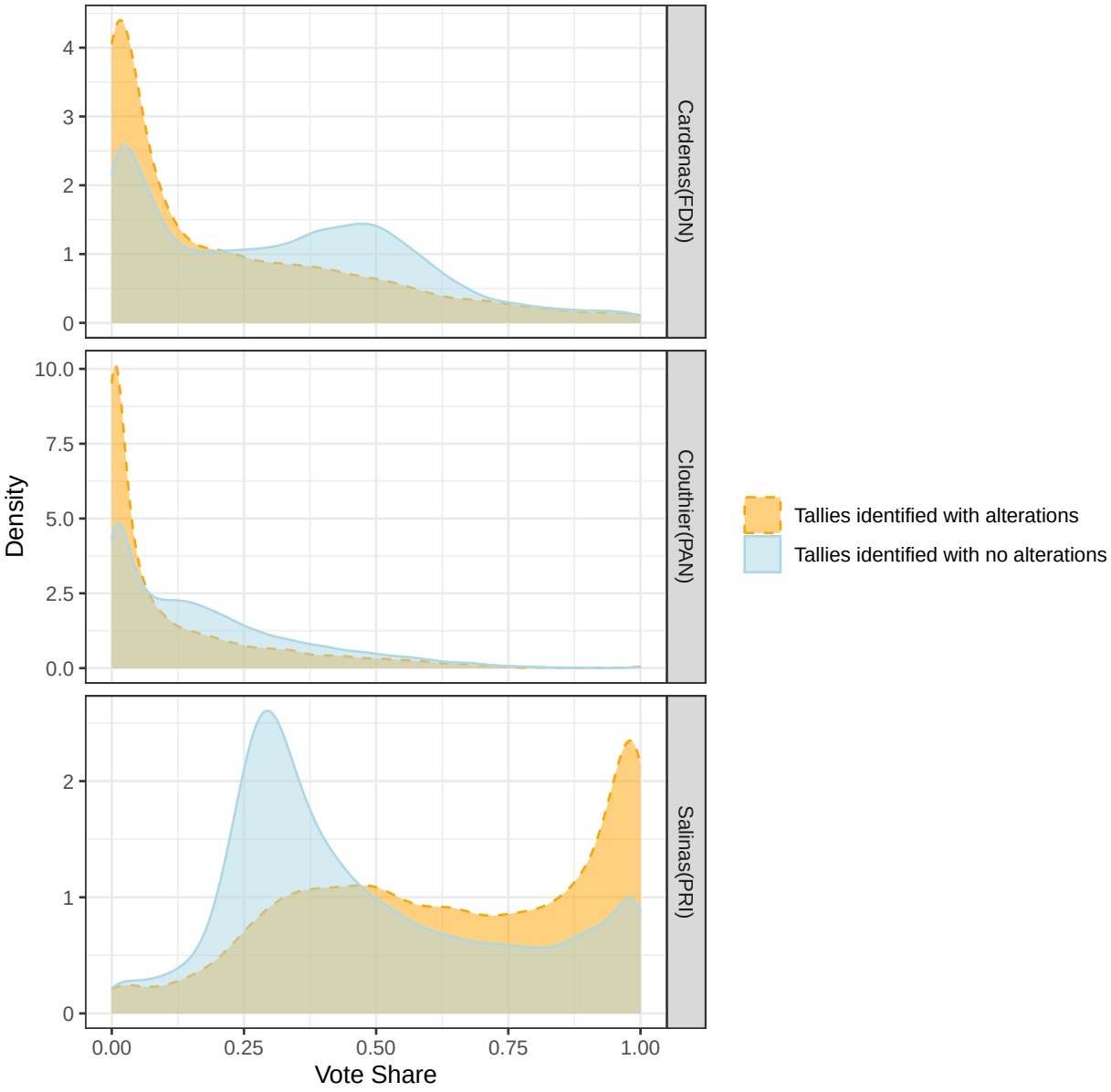
```
d <- d|>
  mutate(total_president =ibarra+castillo+salinas+cardenas+clouthier)|>
  mutate(
    salinas_prop=salinas/total_president,
    cardenas_prop=cardenas/total_president,
    clouthier_prop=clouthier/total_president
  )
tibble(d)
```

```
## # A tibble: 53,289 x 100
##    name_image   state district fraud_proba fraud_bin foto  seccion casilla dtto
##    <chr>        <chr> <chr>          <dbl> <lgl>     <chr> <chr>   <chr>   <chr>
##  1 aguascalien~ Agua~ I         0.000804   FALSE     2014~ 1       84      <NA>
##  2 aguascalien~ Agua~ I         0.0428     FALSE     2014~ 85      85      1
##  3 aguascalien~ Agua~ I         0.423      FALSE     2014~ 45      45-A    1
##  4 aguascalien~ Agua~ I         0.0349     FALSE     2014~ 86      86      1
##  5 aguascalien~ Agua~ I         0.13       FALSE     2014~ 87      87      1
##  6 aguascalien~ Agua~ I         0.212      FALSE     2014~ 1       87-A    7
##  7 aguascalien~ Agua~ I         0.0351     FALSE     2014~ 88      88      1
##  8 aguascalien~ Agua~ I         0.319      FALSE     2014~ 89      89      1
##  9 aguascalien~ Agua~ I         0.00000006 FALSE     2014~ 89      89-A    7
## 10 aguascalien~ Agua~ I         0.36       FALSE     2014~ 89      89-B    7
## # i 53,279 more rows
## # i 91 more variables: dto <roman>, municipio <chr>, edo <chr>, entidad <chr>,
## #   pagina <chr>, p1 <int>, p2 <int>, p3 <int>, p4 <int>, p5 <int>, pan <int>,
## #   pri <int>, pps <int>, psm <int>, pms <int>, pfcrn <int>, prt <int>,
## #   parm <int>, noregis <int>, nombrenore <chr>, otros <int>, otroscan <chr>,
## #   pan2 <int>, pri2 <int>, pps2 <int>, psm2 <int>, pms2 <int>, pfcrn2 <int>,
## #   prt2 <int>, parm2 <int>, noregis2 <int>, otro2 <int>, pan3 <int>, ...
```

Task 4.2. Replicate Figure 4

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

```
ggplot(d_long, aes(x = prop, linetype = fraud_bin)) +
  geom_density(alpha = 0.5, aes(fill = fraud_bin, color = fraud_bin)) +
  scale_fill_manual(
    values = c("Tallies identified with no alterations" = "lightblue",
               "Tallies identified with alterations" = "orange")) +
  scale_color_manual(
    values = c("Tallies identified with no alterations" = "lightblue",
               "Tallies identified with alterations" = "orange")) +
  scale_linetype_manual(
    values = c("Tallies identified with no alterations" = "solid",
               "Tallies identified with alterations" = "dashed")) +
  theme(legend.position = "right",
        plot.caption = element_text(hjust=0))+
  theme_bw()+
  facet_grid(rows = vars(d_long$candidate),scales = "free") +
  labs(x = "Vote Share",
       y = "Density",
       fill = "",
       color = "",
       linetype = "",
       legend.title = "",
       caption = "Figure 2: Distribution of Vote Share for Ecah of the Candidates. Mexico,1988")
```



igure 2: Distribution of Vote Share for Ecah of the Candidates. Mexico,1988

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.

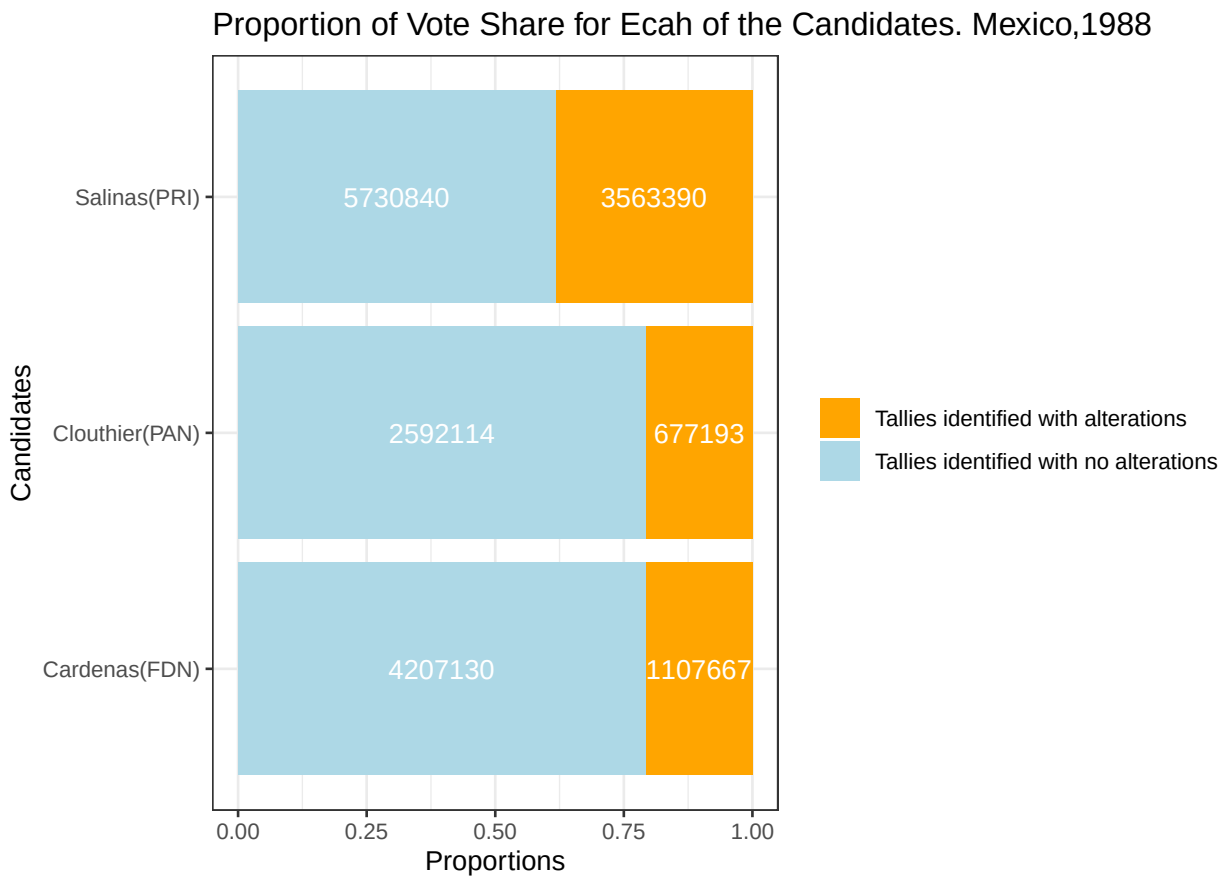
Task 4.3. Discuss and extend the reproduced figure

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.

```
d_long_summary <- d_long |>
  mutate(count=case_when(
    candidate=="Salinas(PRI)" ~ salinas,
    candidate=="Clouthier(PAN)" ~ clouthier,
    candidate=="Cardenas(FDN)"~cardenas,
    TRUE~NA_real_ ))|>
  group_by(candidate, fraud_bin) |>
  summarize(total = sum(count)) |>
  mutate(prop = total / sum(total, na.rm = TRUE))
```

```
ggplot(d_long_summary,aes(x=prop,y=candidate,fill=fraud_bin))+
  geom_bar(stat="identity",position="stack")+
  scale_fill_manual(values = c("Tallies identified with no alterations" = "lightblue", "Tallies identifi
  theme_bw()+
  geom_text(aes(label = total), position = position_stack(vjust = 0.5), size = 4,color="white") +
  labs(y="Candidates",x="Proportions",title="Proportion of Vote Share for Ecah of the Candidates. Mexic
  guides(fill = guide_legend(title = NULL))
```



I designed a stacked bar chart based on the fraudulent status of three candidates' votes. In the graph, the color represents the proportion of fraudulent votes for each candidate, providing a visual representation of the fraudulent percentage.

Additionally, the chart also includes labels indicating the corresponding count for each segment. Since the received vote counts vary, the graph is plotted based on proportions. By including the actual count, it allows for further analysis of the proportion of fraudulent votes in different count scenarios.

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) |>
  summarize(vote_president = sum(total_president),
            vote_legislature = sum(total),
            vote_diff = vote_president - vote_legislature,
            prop_fraud = sum(fraud_bin)/n())
```

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))+
geom_point(aes(size=prop_fraud),alpha=0.1)+
theme_bw()+
labs(x="Total Legislative Votes",
     y="Total Presidential Vote",
     caption="Figure 3: Toal Number of District Votes for Presidential and Legislative Election. Mexi
     size=str_wrap("Proportion of tallies identified with alterations", width=25)
     )+
  theme(plot.caption=element_text(size=10,hjust=0.3))
```

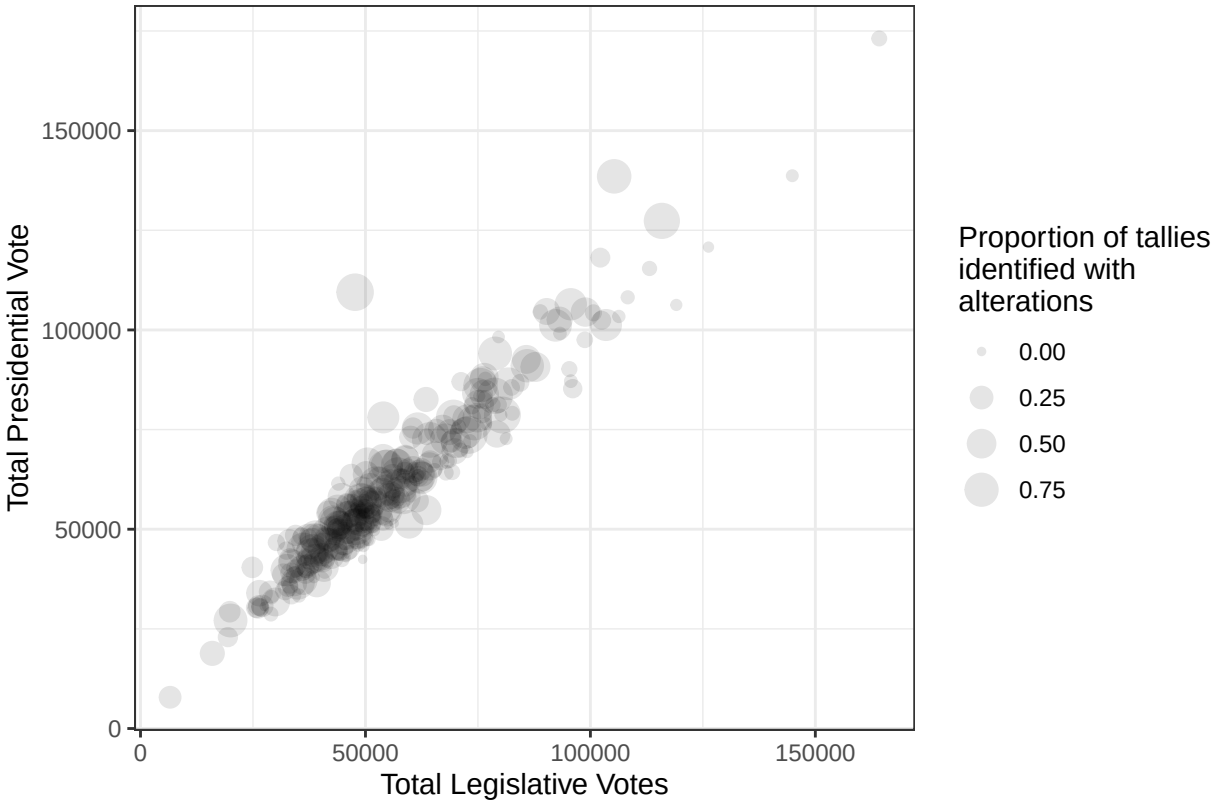


Figure 3: Toal Number of District Votes for Presidential and Legislative Election. Mexico, 1988

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.

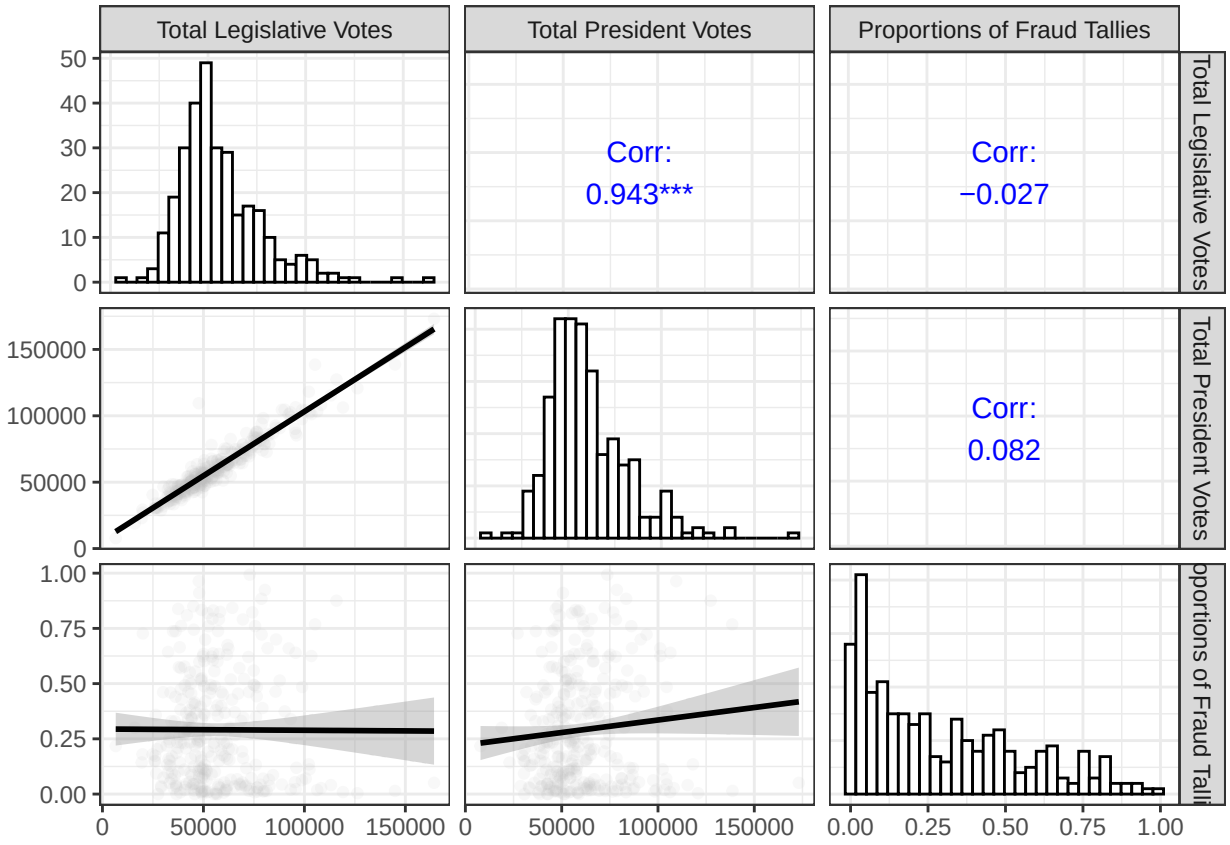
Note 2: The instructor has detected some differences between the above figure with Figure 5 on the published article. Please use the instructor's version as your main benchmark.

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.

```
sum_fraud_by_district |>
  select(vote_legislature, vote_president, prop_fraud) |>
  ggpairs(
    columns = c("vote_legislature", "vote_president", "prop_fraud"),
    columnLabels = c("Total Legislative Votes", "Total President Votes", "Proportions of Fraud Tallies"
    upper = list(continuous = wrap("cor", method = "spearman", color = "blue")),
    diag = list(continuous = wrap("barDiag", bins = 30, fill = "white", color = "black")),
    lower = list(continuous = wrap("smooth", alpha = 0.1, color = "gray")))+
  theme_bw()
```



The matrix scatter plot provides a clearer visualization of the relationships among the three variables: "Total legislative vote," "Total president vote," and "Proportions of fraud tallies." The diagonal bar charts display the distribution and value ranges of each variable. The remaining correlation plots depict the relationships between pairs of variables.

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

Answer:

In the first line of the code, the instructor used st_read() function to load a Shapefile data which contains the information about a state-level Mexico map.

For the second line, the instructor used st_simplify() function to reduce the complexity of the map. dTolerance =100 is the simplification tolerance parameter that represents the level of simplification.

For the result below the code, it shows a basic description of the map, specifically "Dimension: XY" indicates the map is two-dimensional. "xmin""ymax" in the bounding box mean the minimum and maximum of their longitude and latitude. "Geodetic CRS: WGS 84" specifies the geodetic coordinate reference system (CRS) used for the layer is WGS 84.

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.

```
statename_map <- distinct(map_mex,state_name,state_name_official)
statename_fraud <-distinct(sum_fraud_by_district,state)

map_mex <- map_mex |>
  mutate(state=state_name)|>
  mutate(state=case_when(state=="Ciudad de México"~"Distrito Federal",
                         state=="México"~"Edomex",
                         TRUE~iconv(state,"UTF-8","ASCII//TRANSLIT")))

summary(map_mex)
```

```
##       long              lat             order          hole
##  Min.   :-117.25   Min.   :14.53   Min.   :    1   Mode :logical
##  1st Qu.:-103.06   1st Qu.:19.40   1st Qu.:16296   FALSE:65171
##  Median : -99.74   Median :20.71   Median :32592   TRUE :11
##  Mean   :-100.65   Mean   :21.40   Mean   :32592
##  3rd Qu.: -98.46   3rd Qu.:22.26   3rd Qu.:48887
##  Max.   : -86.72   Max.   :32.72   Max.   :65182
##      piece             id               group              region
##  Min.   : 1.000   Length:65182       Length:65182       Length:65182
##  1st Qu.: 1.000   Class :character   Class :character   Class :character
##  Median : 1.000   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 1.076
##  3rd Qu.: 1.000
##  Max.   :17.000
##   state_name        state_name_official  state_abbr        state_abbr_official
##  Length:65182       Length:65182        Length:65182       Length:65182
##  Class :character   Class :character    Class :character   Class :character
##  Mode  :character   Mode  :character    Mode  :character   Mode  :character
##
##
##
##     state
##  Length:65182
##  Class :character
##  Mode  :character
##
##
##
```

```
fraudprop_by_state <- fraud_by_state |>
  mutate('Proportion of altered tallies'=as.numeric(paste0(format(prop_fraud/100))))

merged_map_mex=map_mex|>
  left_join(fraudprop_by_state, by = "state")
```

```
merged_map_mex |>
  ggplot(aes(x = long, y = lat)) +
  geom_map(
    map = merged_map_mex,
    aes(map_id = region, fill = `Proportion of altered tallies`),
    color = "black", size = 0.1
  ) +
  coord_map() +
  scale_fill_gradient(low = "white", high = "black")+
  labs(fill="Proportion \nof altered \ntallies",caption="Figure 4:Rates of Tallies Classified as Altered
    theme(plot.caption=element_text(size=15,hjust=0.5))
```

Figure 4:Rates of Tallies Classified as Altered by State

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.

```
map_mex_sf <- map_mex_sf |>
  mutate(NAME_1=case_when(NAME_1=="México"~"Edomex",
                          TRUE~iconv(NAME_1,"UTF-8","ASCII//TRANSLIT")))|>
  rename(state=NAME_1)
```

```
ggplot()+
  geom_sf(data=merged_map_mex_sf, aes(fill=prop_fraud))+
  scale_fill_gradient(low = "white", high = "black")+
  labs(fill="Proportion \nof altered \ntallies",caption="Figure 5:Rates of Tallies Classified as Altered
    theme(plot.caption=element_text(size=15,hjust=0.5))
```

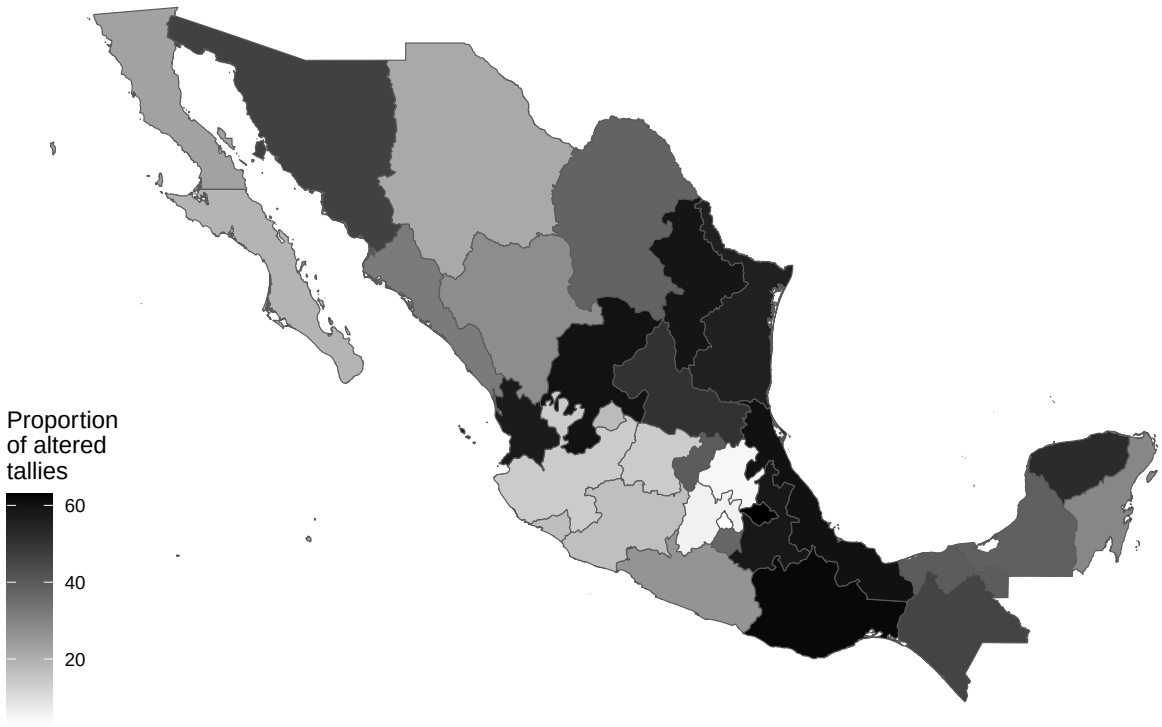


Figure 5:Rates of Tallies Classified as Altered by State

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.

Point map:

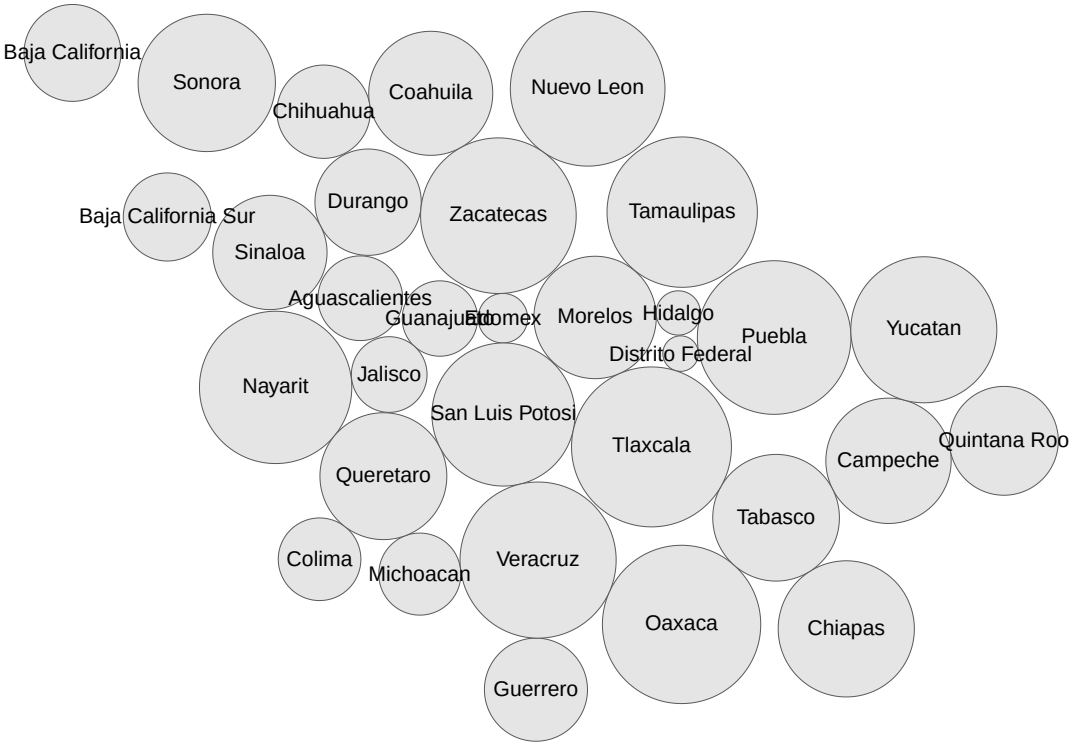
```
merged_map_mex_sf=merged_map_mex_sf|>
  mutate(geometry=st_transform(geometry,3857))

summary(merged_map_mex_sf$prop_fraud)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.096  18.716  37.327  35.373  53.205  62.863
```

```
merged_map_mex_sf_dorling =merged_map_mex_sf |>
  cartogram_dorling(weight="prop_fraud")
```

```
ggplot(data = merged_map_mex_sf) +
  geom_sf(aes(fill = prop_fraud)) +
  geom_sf_label(aes(label = state),size=4)+
  labs(fill = "Proportion \nof altered \ntallies") +
  scale_fill_viridis_c(option = "B", direction = -1, trans = "log")
```


```
ggplot(data = merged_map_mex_sf_dorling) +
  geom_sf() +
  geom_sf_text(aes(label = state))
```



The following cartogram is based on the weight of "proportion of altered tallies" and distorts the map of Mexico, transforming each state into a circle. The larger the circle, the higher the proportion of fraud. Compared to reading a figure without state names, having the state names on the cartogram provides a more intuitive view of the fraud situation in different states.