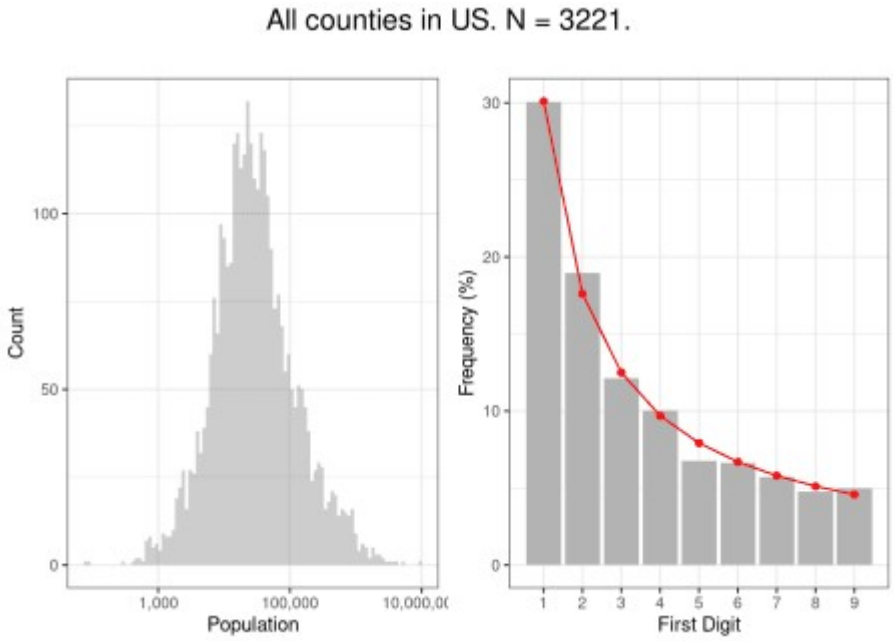
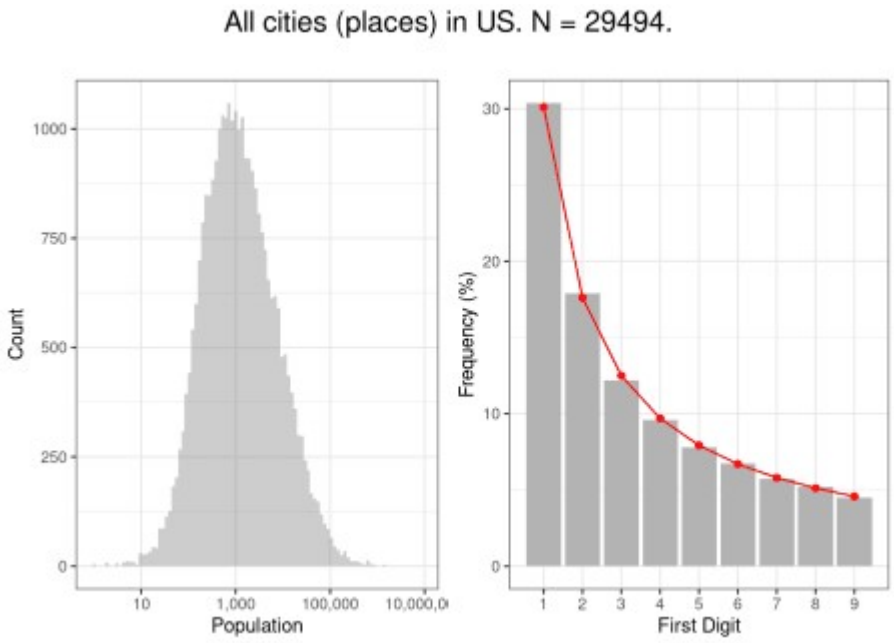
Benford's law attracted lots of attention after the 2020 election, as some people attempted to use this law to question the integrity of the election. Benford's law states that the frequency of the first digit of a large set of numerical data follows this rule: 30.1% are 1s, 17.6% are 2s, 12.5% are 3s, …, 4.6% are 9s. Therefore it can be used to detect data manipulation that leads to the violaton of the law.

I am not going to repudiate those "findings" in the election data but rather provide some examples using census data to show when this law holds and when you should not expect it to work. It is well known that this law works best when the data spread several orders of magnitude and not so well when the numbers are in a narrow range. We will verify this using the population data from the 2010 decennial census, which contains the voting district data.

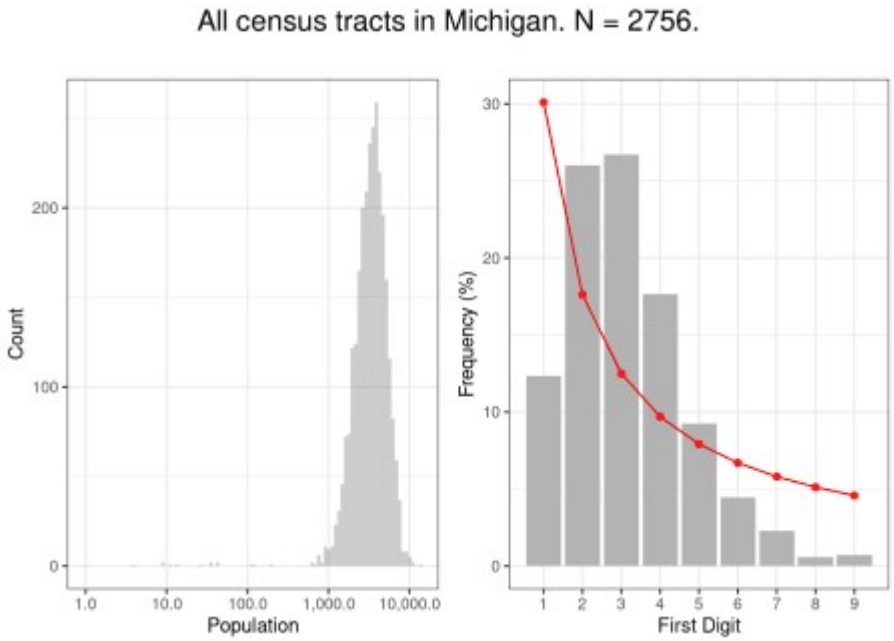
Counties: The population of the 3221 counties varies from a few hundreds to several millions so we expect the distribution of the first digits is a good match to the Benford's law. It is true as shown in the figure below (red points and line show the values for Benford's law).



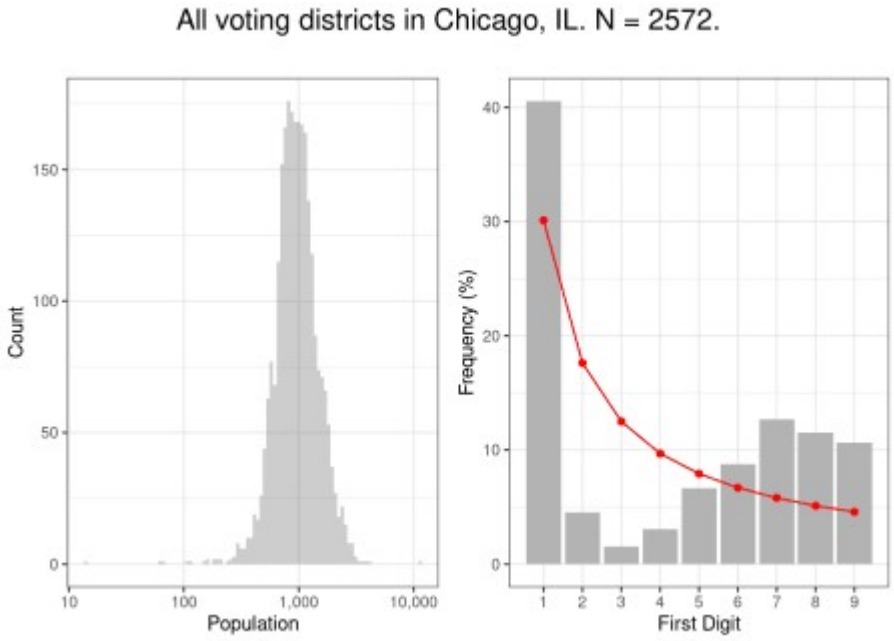
Cities: We have the population of 29494 cities (and places) and the distribution of the first digit (almost) perfectly follows the Benford's law.

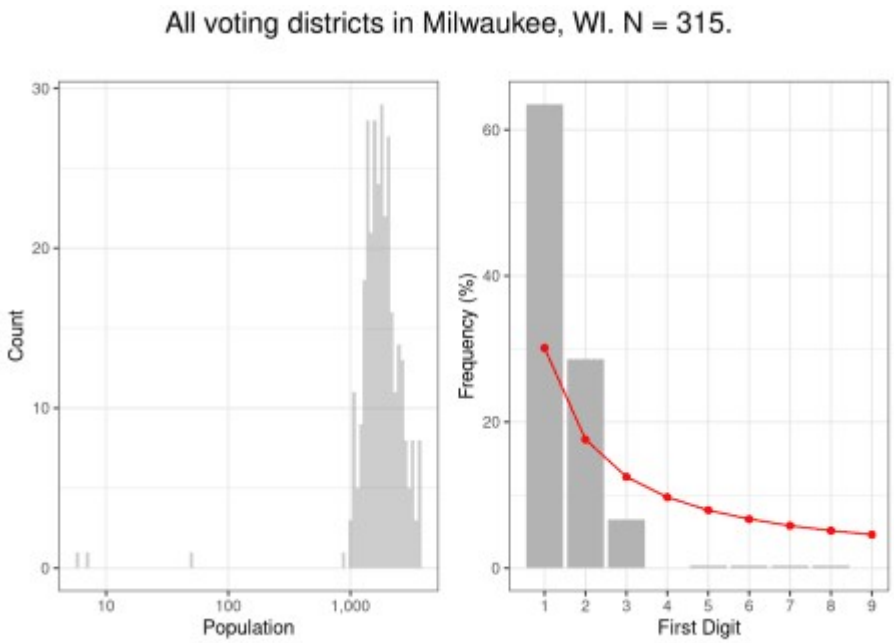


Census tracts: We have seen two good examples. Here we will see examples that do not follow the Benford's law. The first one is the population in census tracts. Most census tracts have a few thousands people. For the 2756 census tract in Michigan, the most frequent leading digit is 3 instead of 1.



Voting districts population does not obey the Benford's law either. Here we show the data for two cities, Chicago IL and Milwaukee WI. What can you tell from the distribution of the first digit? Do not expect the Benford's law to bring you any excitement.





Code:

```
library(totalcensus)
library(ggplot2)
library(scales)
library(grid)
library(gridExtra)
library(data.table)
library(magrittr)
library(stringr)

df_benford <- data.table(
    first_digit = 1:9,
    Freq = log10(1 + 1 / (1:9))
)

plot_benford <- function(x, title = ''){
```

```
    first_digit <- x %>%
        abs() %>%
        as.character() %>%
        str_sub(1, 1) %>%
        as.integer()
    count <- (table(first_digit) / length(first_digit)) %>%
        as.data.frame()
    count$first_digit <- as.integer(as.character(count$first_digit))
    count %>% ggplot(aes(first_digit, 100 * Freq)) +
        geom_col(fill = 'grey70') +
        geom_point(data = df_benford, color = 'red') +
        geom_line(data = df_benford, color = 'red') +
        scale_x_continuous(breaks = 1:9, minor_breaks = NULL) +
        labs(title = title,
             x = 'First Digit',
             y = 'Frequency (%)') +
        theme_bw()
}

plot_hist <- function(x, title = ''){
    ggplot() +
        geom_histogram(aes(x), alpha = 0.3, bins = 100) +
        scale_x_continuous(trans = 'log10', labels = comma,
minor_breaks = NULL) +
        labs(title = title,
             x = 'Population',
             y = 'Count') +
        theme_bw()
}


plot_together <- function(x, title = ''){
    g1 <- plot_hist(x)
    g2 <- plot_benford(x)
    grid.arrange(g1, g2, nrow = 1,
                 top = textGrob(title,
                                gp = gpar(fontsize = 16)))
}


# counties
county <- read_decennial(2010, "US", summary_level = 'county') %>%
    .[population != 0, population]
plot_together(county, 'All counties in US. N = 3221.')


# cities and places
place <- read_decennial(2010, "US", summary_level = 'place') %>%
    .[population != 0, population]
plot_together(place, 'All cities (places) in US. N = 29494.')
```

```
# centus tracts
tract <- read_decennial(2010, "MI", summary_level = 'tract') %>%
    .[population != 0, population]
plot_together(tract, 'All census tracts in Michigan. N = 2756.')


# voting districts
chicago_vtd <- read_decennial(2010, 'IL',
                        geo_headers = c('PLACE', 'VTD'),
                        summary_level = 'block') %>%
    .[PLACE == '14000'] %>% # select Chicago
    .[population != 0] %>%
    .[, .(popul = sum(population)), VTD] %>%
    .[, popul]
plot_together(chicago_vtd, 'All voting districts in Chicago, IL. N =
2572.')


milwaukee_vtd <- read_decennial(2010, 'WI',
                             geo_headers = c('PLACE', 'VTD'),
                             summary_level = 'block') %>%
    .[PLACE == '53000'] %>% # select Milwaukee
    .[population != 0] %>%
    .[, .(popul = sum(population)), VTD] %>%
    .[, popul]
plot_together(milwaukee_vtd, 'All voting districts in Milwaukee, WI. N
= 315.')
```