Why Do Biden's Votes Not Follow Benford's Law?

A Verification Experiment

Corrina M. Del Greco

Embry-Riddle Aeronautical University

INTRODUCTION

For my final project, I chose to create a Jupyter Notebook based on a YouTube video by *Stand-up Math* called "Why do Biden's votes not follow Benford's Law?" in which I verify the claims from the video in a series of individually run blocks of Python code, leading the user through the process in a lesson style. The problem: Benford's law and election data from Chicago have been used for claims of (and defenses against) election fraud. There are a lot of "backed up" claims that we are expected to trust because data is presented. The video disproves one of those claims, but in doing so, it itself is one. Our solution is to verify, double check, and see for ourselves. That is an integral part of data science.

THE CODE

Before any further analysis can be done, the data must be loaded. Because the notebook was created for others to use and learn from, this is an opportunity for user interaction. Pictured below are the code cells that the user is invited to change, as well as the output instructions.

```
# --set path to project files--
print('\nChange the path in this cell to where the project files were downloaded on your machine.')
my_folder = "c:\\Users\\Corrina Del Greco\\Documents\\ERAU\\MA305\\My Work\\Project\\"
print('You\'re telling me that the files are at: ' + my_folder + '\n')
```

```
# --check if benford djstribution file exists--
bd_local_csv_filename = my_folder + "benforddist.csv"
my_file = Path(bd_local_csv_filename)
if my_file.exists() == False:
    print('\nFile does not exist: ' + bd_local_csv_filename)
    print('Check and rerun the cell that sets that path before continuing.\n')
    exit()
```

```
Change the path in this cell to where the project files were downloaded on your machine.
You're telling me that the files are at: c:\Users\Corrina Del Greco\Documents\ERAU\MA305\My Work\Project\
```

```
# --set variables based on data file--
print('\nElection data for Chicago\'s 2069 precincts is publicly available online.')
# https://chicagoelections.gov/en/election-results-specifics.asp
print('I\'ve downloaded the data and converted it to a csv containing only columns relevant to our experiment:')
print('   Total votes, votes for Biden, and votes for Trump')
print('If you want to do this yourself, change the values in this cell to match your file.\n')

local_csv_filename = my_folder + "mydataexport.csv" # csv file name
v_column = 'Votes'                                  # column header for total votes
d_column = 'Joseph R. Biden & Kamala D. Harris'     # column header for democrat votes
r_column = 'Donald J. Trump & Michael R. Pence'     # column header for republican votes
```

```
# --check if data file exists--
my_file = Path(local_csv_filename)
if my_file.exists() == False:
    print('\nFile does not exist: ' + local_csv_filename)
    print('Check and rerun cells that set that path before continuing.\n')
    exit()
```

```
Election data for Chicago's 2069 precincts is publicly available online.
I've downloaded the data and converted it to a csv containing only columns relevant to our experiment:
   Total votes, votes for Biden, and votes for Trump
If you want to do this yourself, change the values in this cell to match your file.
```

Figure 1: Modifiable cells and their output

Benford's Law (the values for which are called Benford's Distribution) is captured in a comma separated values (csv) file I created. Once loaded, it is displayed and explained to the user, and the premise for our experiment is set up.

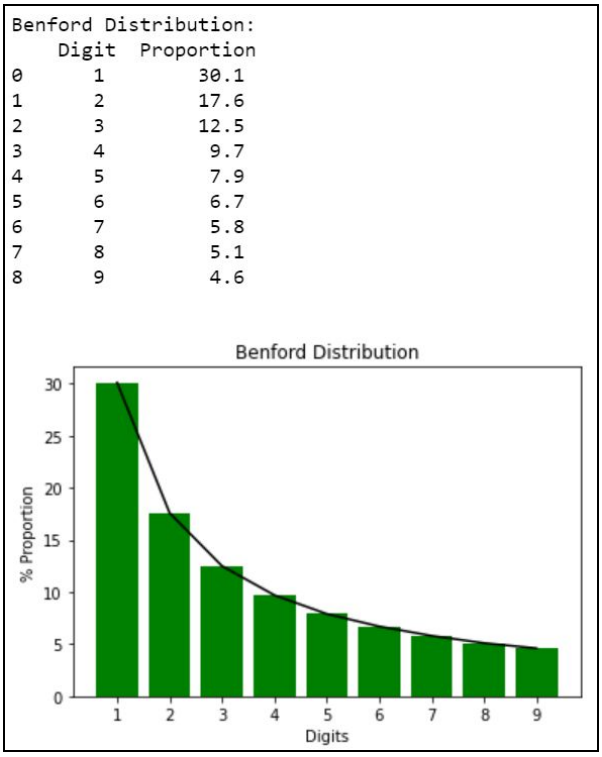

Figure 2: Plot of Benford Distribution

So what is Benford's law? Benford's law is a statement that if you get a large range of data from the "real world" and you look at the lead digit of each of the values, if you count up how many are ones, how many are twos, and so on, you get way more ones than anything else. It is important to note: Benford's Law applies only when the data spans multiple magnitudes. More on this later. The discussion is about a) Benford's law being used for detecting election fraud, and b) Joe Biden's vote totals not matching Benford's Law (where Donald Trump's vote totals do).

Before starting to explore that, we define some functions that will come in handy:

```
# --define helper functions--
def first_digit(n):
    while (n >= 10):
        n /= 10
    return int(n)

def last_digit(n):
    return n % 10

def last_two_digits(n):
    return np.abs(n) % 100
```

Figure 3: Helper functions

We can use the functions when adding columns to our loaded dataframe. This takes the form of the first line in Figure 4. The lambda operation applies our helper function to each of the values in a chosen column (in this case, whichever column has a header string equal to the string stored in `d_column`) and the result is stored in a new column named to the left of the equals sign (in this case, `Biden First Digit`).

```
# --plot Biden's first digits--
df['Biden First Digit'] = df[d_column].apply(lambda x: first_digit(x)) # add a column to the dataframe using first_digit
d_count = df['Biden First Digit'].value_counts().to_dict()            # count frequency of each digit
plt.bar(d_count.keys(),d_count.values(),color='b')                    # plot
plt.title("Biden Lead Digit Frequency")
plt.xlabel("Digit")
plt.ylabel("Frequency")
plt.xticks(df_bd['Digit'])
plt.plot(df_bd['Digit'],df_bd['Proportion']/100*max(df[v_column]), color='black') # overlay Benford's distribution
print()
plt.show()
print('\nIf you look at Biden\'s vote totals across the Chicago area, they do not match Benford\'s Law.\n')
```

Figure 4: Code for Biden's Benford Distribution

We often have to graph the number of times a number occurs, or the frequency. To do so, I use the `value_counts()` dictionary, and use the `values()` and `keys()` accessors for the x and y values of my plot (see the second and third lines in Figure 4). The result (with the Benford Distribution overlaid on top) looks like this:

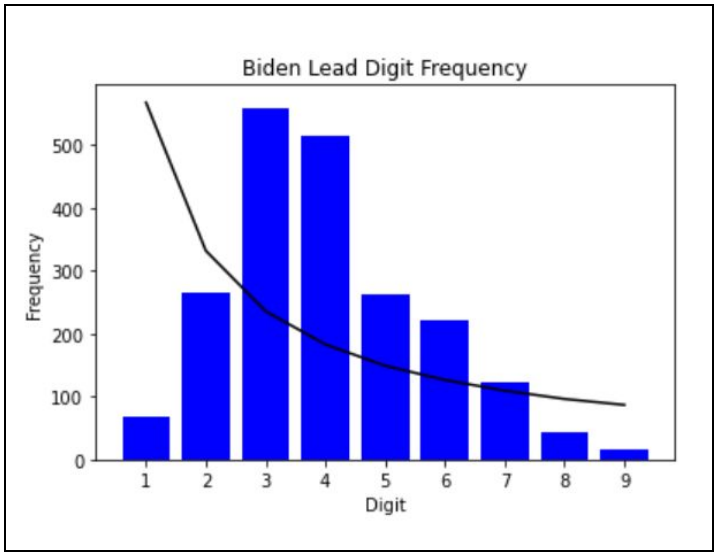

Figure 5: Plot of Biden's Benford Distribution

With this we can confirm that Biden's vote totals across the Chicago area do not match Benford's Law.

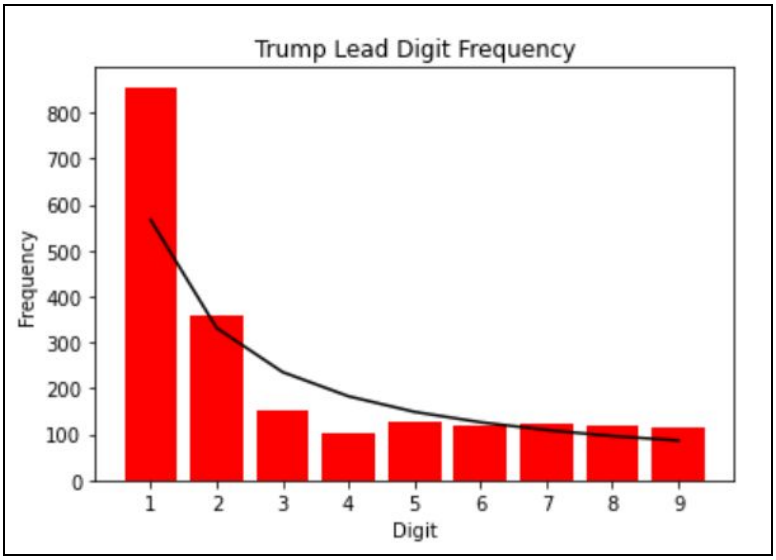

Figure 6: Plot of Trump's Benford Distribution

On the other hand, Trump's data looks like it does match Benford's Law. And with that, we've successfully verified the initial premise of the video (and this specific claim of fraud to which it

responds to). Let's continue, and explore why Biden's data does not follow Benford's Law when Trump's seemingly does. Trump's data offers some interesting results, as does Biden's.

We make some calculations to evaluate our dataset more closely.

```
There are 2069 precincts.
They're picked to have roughly the same size population each, and we can see that.
According to the data we've loaded:
The smallest one, according to our data, had 39 votes.
The biggest one, 1884.
The average however, 550.
The standard deviation, very tight! 195.
There are only 6 precincts with less than 100 votes,
and only 46 precincts that had 1000 or more votes,
which means 2023 or a massive 97% of precincts had three digit totals.
```

Figure 7: Evaluating the dataset

Recall that we said that Benford's Law only applies when the data spans multiple magnitudes. The above shows that we have the opposite of that. So it's actually not expected to get a Benford Distribution.

Those calculations are used in the test cases for this project. How do our results compare to that of the YouTube video? We take numbers cited in "Why do Biden's votes not follow Benford's Law?" as "expected" values and compare our own calculated values.

	Value	Expected	Actual	Pass
0	Precincts	2069.0	2069.000000	True
1	Min votes	39.0	39.000000	True
2	Max votes	1655.0	1884.000000	False
3	Average votes	516.0	550.614790	False
4	Standard deviation in votes	173.0	195.816018	False
5	Precincts with < 100 votes	7.0	6.000000	False
6	Precincts with >= 1000 votes	20.0	46.000000	False
7	Precincts with three-digit vote totals	2042.0	2023.000000	False
8	% of precincts with three-digit vote totals	98.7	97.776704	False

Figure 8: Test cases

Many of the numbers do not match, so they failed the equals test. We can redownload the data and examine it, but the results are the same. This does not necessarily mean the experiment is failed, some observations can be made: The passed test for number of precincts is a good sign; we certainly wouldn't expect that to change. The differences are not huge. Where they do differ, the value from our data is always higher than the "expected", with the exception of the number of precincts below 100 votes and the number/percentage of precincts with three-digit vote totals. As for those, a lower value actually represents the vote totals being higher anyway. The conclusion: More votes have been counted. The video which our experiment is verifying was published on November 10th, just days after the election. Our data was downloaded again nearly a month later.

At this point, we've verified the claim that Biden's votes do not match Benford's law but Trump's do, we have verified the response that this is not a good test for fraud because of the magnitude issue, but there is more exploration done in the video, so we continue to verify.

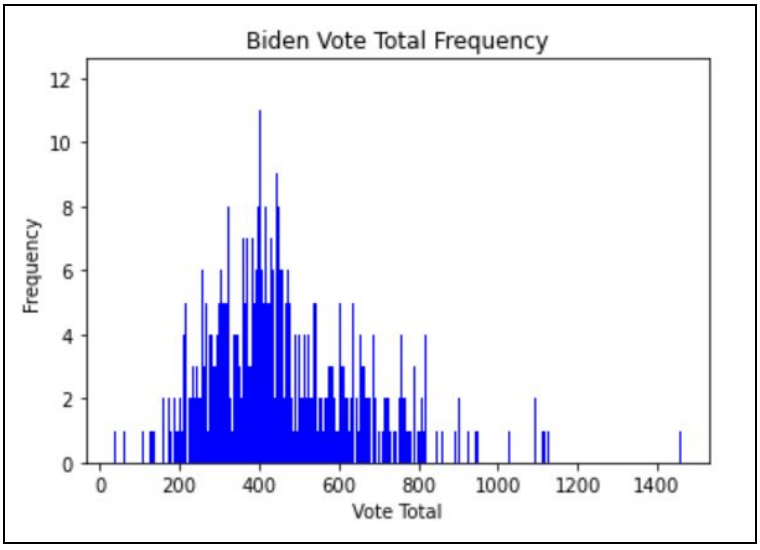

Figure 9: Plot of Biden's total votes

Pictured in Figure 9 is a distribution of Biden's vote totals. As you can see, it's basically a normal distribution. That's because Biden was the favorite in the Chicago area, so he got a consistent percentage of those three-digit precincts. Take a look again at the Benford Distribution for Biden in Figure 5 and compare. What you're looking at in Figure 9 is a simplified version of Figure 5. The "Benford Distribution" is reflecting the data itself.

There is another method. You can look at the last digits and expect them to be roughly random. If not, that may be something to look into. Humans are bad at picking random numbers. With the following screenshots, you can compare the last two digits of Biden's vote totals with completely random two digits, and see that they're very similar. You can also compare the same with Trump's, and see something completely different!

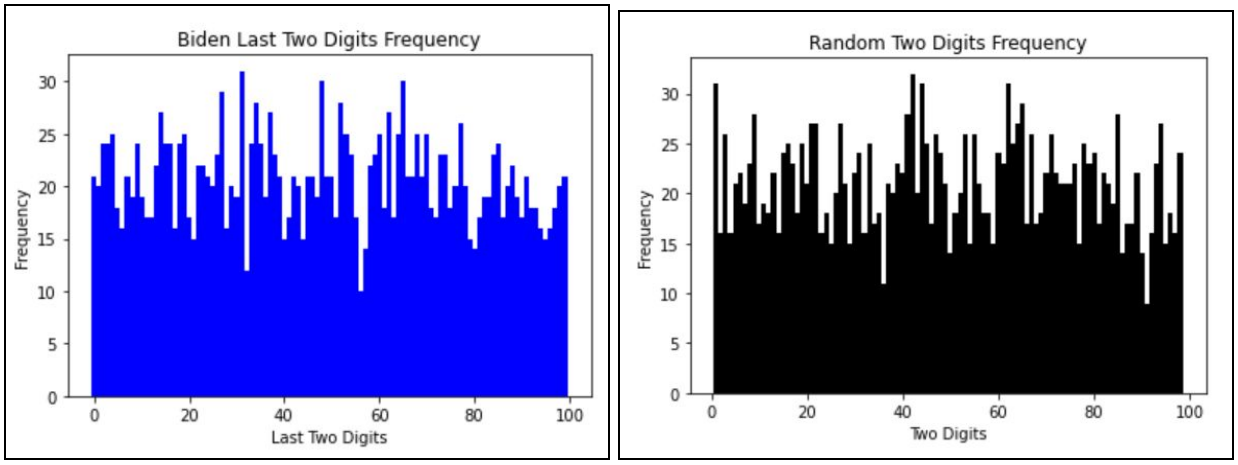

Figure 10: Plot of Biden's last two digits compared to random

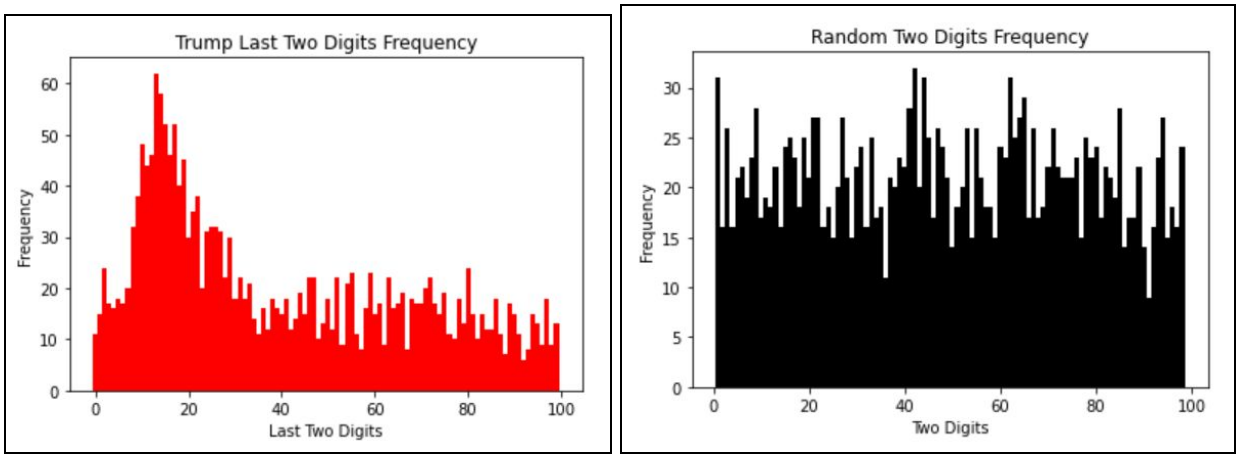

Figure 10: Plot of Trump's last two digits compared to random

Trump's data is a major deviation from what we would expect and therefore a cause for concern, possibly fraud. Or is it? Earlier we looked at the distribution of vote totals for Biden (see figure 9). If we do the same for Trump, we can compare.

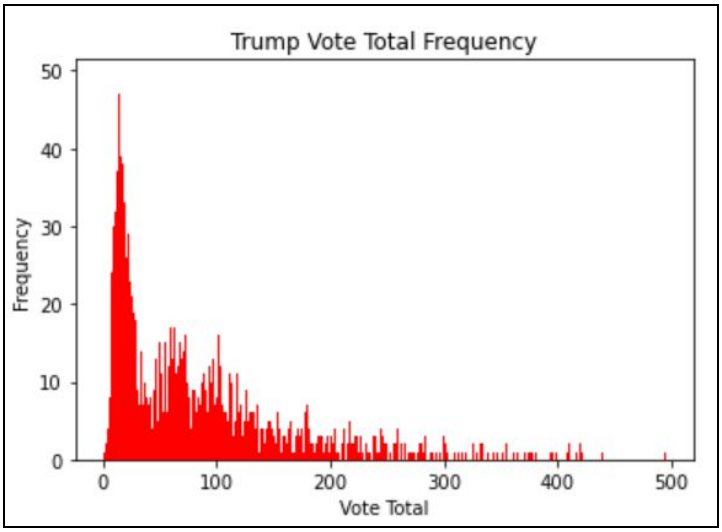

Figure 11: Plot of Trump's vote total frequency

Whereas Biden's graph resembled a normal distribution, the majority of Trump's vote totals were only a two digit number. So when we're looking at the "last two digits", in most cases, it's just

the digits. So we should not expect a random distribution. Once again, though at first something seems off, we are actually just looking at the data itself.

CONCLUSION

That is the moral of the story: these tests can be an interesting way to spot that something might be wrong, but they do not guarantee that there is something wrong. You should take a closer look at the data, don't take someone's word for it, verify for yourself, double check.

I did not repeatedly affirm in this report that the plots and data from my experiment did in fact match those from the video, because it would have become redundant. All of the plots "passed the test," and the test cases that I discussed earlier passed as well, with the observations that were made being taken into account.