2020 Presidential Race Benford's Law Analysis UPDATED

After some serious deliberation on which article to focus on for this paper, I decided to focus on the Coleman et al.’s paper, “Potential for Concordance between Plurality and Instant-Runoff Election Algorithms as a Function of Ballot Dispersion.” Out of each article I read, Coleman et al.’s paper was easily the most interesting. It was a beautiful example of how mathematics can appear in aspects of our lives that we would never even consider due to the underlying use of applied mathematics.

***Required Prerequisite Knowledge***

A robust knowledge of statistics and some understanding of information theory are both needed to fully comprehend the concepts communicated in Coleman et al.—such as Shannon entropy and Herfindahl-Hirschman Index (HHI). Regardless, if a reader wishes to understand the process of gathering and interpreting data results in any research article which does a statistical experiment, they must have some background in statistics. As discussed previously in Assignment 1, Shannon entropy, *H(x)*, and HHI, *HHI(x),* are given by

and

where *x* is a discrete random variable and *P(x)* is the probability mass function of *x*. Both Shannon entropy and HHI were used to measure ballot concentration, but what are these concepts and why were they of particular interest?

In statistics, Shannon entropy, or simply “entropy,” is a measure of a process’s uncertainty (Glen, 2020). More specifically, Shannon entropy measures the amount of information or “surprise” in a variable’s theorized outcomes. HHI is a bit different in the sense that it measures “market concentration” which is used to determine how competitive a select market is (Hayes, 2021). However, HHI is used outside of this context to measure the concentration of any variable of interest, not just of a particular market or business. Altogether, Coleman et al. uses Shannon entropy and HHI to complement each other: the former measures the amount of uncertainty within a ballot’s structure, and the latter the dispersion (i.e. the lack of concentration) of the ballot.

While a prerequisite knowledge of statistical software such as R is not required to comprehend this article, it is a must if one desires to reproduce this experiment. Specifically, R is needed to reproduce models such as the Monte Carlo simulation done in Coleman et al. A Monte Carlo simulation is a way of gathering data in situations where there is “significant uncertainty” due to the appearance of random variables by repeating the same experiment numerously with different random samples used in each iteration (Kenton, 2021). A proficiency in R would prove very useful in reproducing Monte Carlo simulations since there already exists a *MonteCarlo* package that has the functions *MonteCarlo()* and *MakeTable()* which runs a Monte Carlo simulation and displays the results, respectively (Leschinski, 2019). Without the use of R or software of the like, I am unsure if this Monte Carlo simulation could be replicated without a significant loss of time.

***Analysis of Election Results in Washoe County***

The results produced by the two different election algorithms discussed within Coleman et al. were most intriguing to me. It was concluded that a more evenly distributed voter preference results in a lower concordance, or agreement, between Plurality and Instant-Runoff Voting (IRV) election algorithms, while a more concentrated voter preference yields a high concordance. In simulated/hypothetical elections, these results are drastic; however, Coleman et al. concluded that in real-world elections, situations where voter preference is evenly distributed are “extremely uncommon” due to the inherent nature of the U.S. two-party system. In other words, there is “very little difference” realistically in voting algorithms for the U.S. two-party system.

In order to see just how different Coleman et al.'s conclusion would be in the U.S. two-party system, I chose to analyze the 2020 presidential election results in Washoe County by precinct. The 2020 presidential election was chosen due to its high rate of voter engagement and because it is a prime example of how the two-party system works. Washoe County, Nevada was chosen as the area of focus since it is a familiar area to both me and Prof. Waddell; in fact, Nevada was of particular interest since it is known to be a “swing state” and I desired to know whether this factor could potentially change the results of elections under different voting algorithms. It should be noted that Nevada uses the Plurality election algorithm exclusively regardless of the election. As supported by Coleman et al., most elections governed under the Plurality algorithm have very limited information available to any interested observers, so attempting to find voters’ 2nd or 3rd choice candidates for the 2020 presidential election yielded no results.

The 2020 presidential election results in Washoe County by precinct data were collected via the Nevada Secretary of State website through a provided Excel spreadsheet. This Excel spreadsheet originally contained the 2020 statewide general election results of Nevada and had data categorized into five columns: county jurisdiction, precinct, contest, selection, and votes tallied. I edited the data to only include the precinct data of Nevada in the “President and Vice President of the United States” contest, which ultimately left me with 182,843 rows of data. In order to protect the secrecy of the voter’s ballot in accordance with Nevada law, cumulative voter turnout that was greater than 0 but less than 10 were assigned an asterisk on this spreadsheet; I will omit the 640 rows with no election data provided, leaving me with 182,203 rows of election data to analyze. The data was then transferred to R for further analysis.

In R, the data was categorized such that each presidential candidate was grouped with their total number of votes accrued and was then filtered to only show the results of those in Washoe county. This candidate data was then transformed into a bar chart for easy visualization (Figure 1).

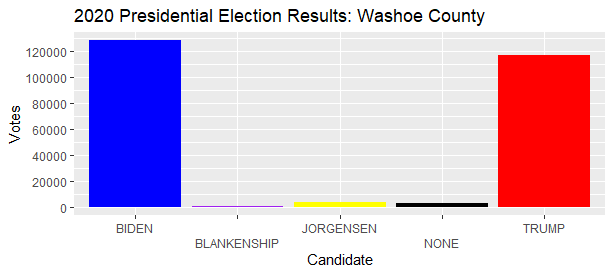


Figure 1

As highlighted by Figure 1, candidate Trump, the Republican party nominee, and candidate Biden, the Democratic party nominee, both hold the vast majority of total votes in Washoe County. In fact, proportionally, Biden holds approximately 51% of total recorded votes in Washoe County while Trump holds just over 46% of the votes. Observe how both of the third-party candidates have little support: Jorgensen of the Libertarian party held only 1% of the votes, while Blankenship of the Constitution party garnered a mere 0.2%. These results are a shining example of Coleman et al.’s conclusion about the U.S. two-party system, wherein that for election algorithms to disagree “all… candidates must be quite similar in the size of their support” (p. 37); as shown by Figure 1, this is hardly the case. With the race only being between candidates Trump and Biden, the results would likely not differ under a different election algorithm such as IRV. This is not only because the total votes between the two candidates differ by 5% (a fairly large difference in an election) but also because voters in either the Republican or Democratic parties would very unlikely select their opposing party’s candidate as their 2nd choice in an IRV election and would rather select a third-party candidate. Thus, the results in the Washoe county 2020 presidential election would likely not differ under the IRV election algorithm.

If we were to expand outward to the state level, perhaps a change in election algorithm could change the outcome. Using the same information inputted into R, Figure 2 was created to present the total number of votes per candidate statewide.

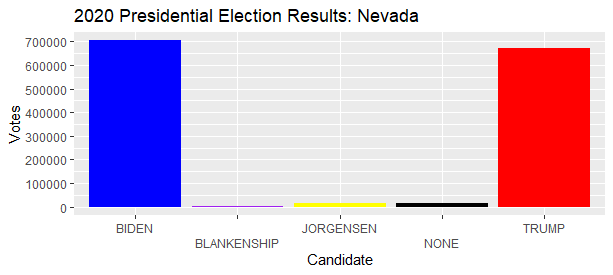


Figure 2

In Figure 2, the election in Nevada is revealed to be a much closer race between candidates Biden and Trump. Statewide, Biden just barely holds the majority with 50.06% of the total votes whereas Trump holds a little over 47% of all votes in Nevada. These results are exactly the reason why Nevada was a crucial swing state to win in the 2020 presidential race since the difference in votes between candidates Biden and Trump were within 3% of each other. However, Coleman et al. states that election algorithm concordance approaches 100% when a single candidate holds “close to a majority of first-choice preferences'' (p. 36). Coupled with the earlier hypothesis regarding how the IRV election algorithm would prove futile due to Democratic and Republican party voter ties, a change in election algorithm even at this close of a race would likely not change the outcome of the election. However, in the hypothetical scenario where voters who had chosen either Blankenship, Jorgensen, or none had instead voted for candidate Trump, then Trump would have 49.93% of total votes. In this scenario, the race would have been incredibly close to call between candidates Biden and Trump. Now since this hypothetical candidate pool is much more uniform, the concordance between the IRV and Plurality algorithms would be low, however we will not be focusing on this hypothetical scenario in our paper and we will instead note that this would have made for a very exciting race.

***Benford Analysis of Election Results in Washoe County***

A typical analysis done in certain situations to test for fraud or “funny business” is the Benford analysis. Benford analysis was introduced to me in MATH 305, Functions and Modeling, by Prof. Waddell and Prof. Keppelmann and I recognized it to be a powerful tool in mathematics, business, statistics, and generally any area where fraud detection is needed. While usually Benford analyses are conducted in legal situations involving businesses, we will be using Benford to analyze the election results in Washoe County.

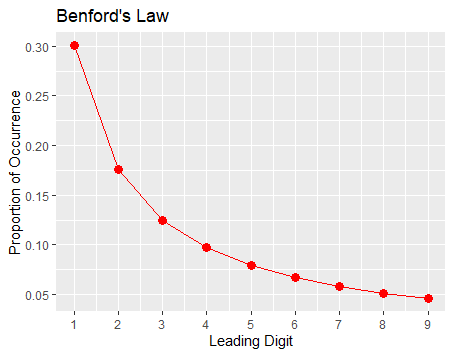
Using the “2020 Presidential Election Results in Washoe County” spreadsheet in the previous section, we calculated the leading digit of the number of votes each candidate received by precinct by using the *LEFT* function in Google Sheets. In some cases, candidates did not receive any votes in certain precincts (these were usually third-party candidates in smaller precincts); regardless, these cases were still uploaded to R and were simply omitted from the Benford analysis.

Plotting the original Benford graph was simple. I defined a function, *B(d)*, based on the original formula of Benford’s Law as follows:

,

where *d* are the leading digits 1 through 9. This graph of Benford’s Law (Figure 3) will be used as reference for the upcoming Benford analyses.

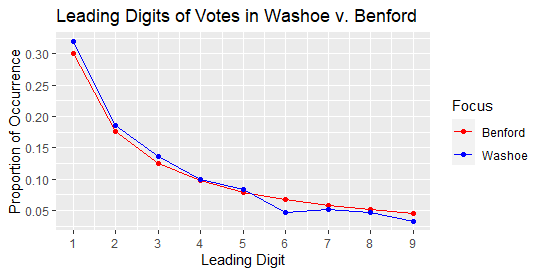
Figure 3



We will first start our Benford analysis at the county level by once again focusing on Washoe county. Since we have already calculated the leading digits of the votes that each candidate received by precinct before importing our data into R, all that is left to do is to calculate the percentage of each digit’s occurrence based on the total occurrences of all the digits. These percentages were first put into a dataset and then transferred to a graph (Figure 4) along with the previous plot of Benford’s Law for ease of visualization.

At this point in our investigation of fraud, it is important to note that the efficacy of the Benford analysis is largely based on the size of the data that is being analyzed. Studies have shown that Benford’s Law holds for data sets as small as 50 to 100 numbers, though typically data sets containing 500 or more numbers are best suited for the Benford analysis (Collins, 2017). Our data set at the county level is 1732 numbers large after the omission of any leading digit 0, so the Washoe county 2020 presidential election is of good size for the Benford analysis.

Figure 4



As exemplified by Figure 4, the leading digits of the votes in Washoe county seem to line up fairly nicely with that of *B(d)*. With the exception of digit 6, every leading digit of the collected votes in the Washoe county election were within 2% of the leading digit percentage associated with Benford. The biggest variation was digit 6 which occurred 4.68% of the time, whereas according to Benford’s Law, digit 6 should be apparent 6.70% of the time. Since the biggest variation to *B(d)* is 2.02%, it is highly unlikely that any sort of ballot manipulation has taken place.

Expanding outward to the state level, we find the Benford analysis of the 2020 presidential race in Nevada to be a bit more interesting. We show the results of the Nevada Benford analysis in Figure 5 along with a combined Benford analysis featuring both Washoe county, Nevada, and *B(d)* in Figure 6 for comparison.

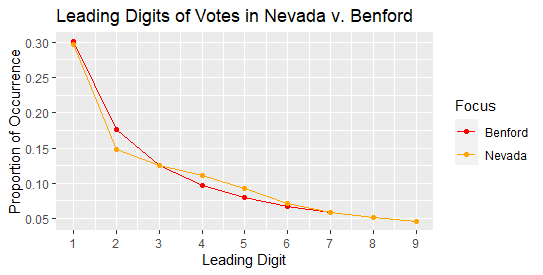


Figure 5

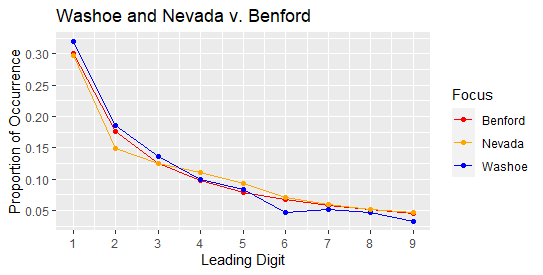


Figure 6

After inspection of Figure 5, we can see that the leading digits of the votes at the Nevada level follow the Benford trend slightly worse than Washoe county, although the concordance between Nevada’s votes and *B(d)* is still strong. The largest difference in leading digit occurrence between the Benford and Nevada graphs are at leading digit 2 with an approximate 2.8% difference (where *B(2) = 17.6%* and Nevada at leading digit 2 with 14.8%). It is safe to assume that even at the state level there was likely no outside manipulation of the ballot.

***Discussion***

Analysis of the 2020 presidential election at both the county and state levels has led to two conclusions. First, if the elections at either the state or county level were conducted via the IRV voting algorithm instead of the Plurality algorithm, the outcome would likely remain the same. This conclusion only further substantiates the conclusion proposed by Coleman et al. where there is “very little difference” in voting algorithms for the two-party U.S. electoral system (p. 37). Secondly, based solely on the results of the Benford analyses, there was likely no outside manipulation in either the Washoe county nor Nevada election ballots. Although a Benford analysis alone is likely not enough to prove that there was no ballot manipulation at all, it does give us a good reason to suspect that there was no manipulation.

Since this paper only conducted surface-level analysis on the 2020 presidential election at the state and county levels, there is ample opportunity for future research. Perhaps the most obvious opportunity is a deeper ballot manipulation analysis of Nevada’s election results. As stated previously, Nevada was a crucial swing state to win during the 2020 election, and thus there is a good reason to believe that the voter ballot may have been expertly tampered with to bypass even a Benford analysis. In fact, there has been some skepticism as to whether or not Benford’s Law can suggest election fraud, thus other statistical methods for fraud detection should be performed to support the Benford analysis. Other research opportunities consist of conducting analyses of election algorithm concordance in other swing states during the 2020 presidential race. In this paper, we found that the race between Biden and Trump in Nevada was very close, but ultimately concrete in favor of Biden regardless of the election algorithm used. Perhaps in an even closer race where the two-party candidates are within less than 3% of each other, concordance in election algorithms would decrease.

***R Code***

# Data Analysis: 2020 Presidential Election Fraud Analysis via Benford's Law

# Author: Dillan Marroquin

# Date: 05/01/2024 (Originally conducted on 08/11/2021)

# Load libraries

library(tidyverse)

library(scales)

# Set working directory

setwd("C:/Users/di11m/Downloads/Coursera/Data Analysis/RStudio")

# Load election data for Washoe and NV

washoe <- read.csv("Election\_Results\_Washoe\_CLEANED.csv")

nv <- read.csv("Election\_Results\_NV\_CLEANED.csv")

# Define candidate names

candidates <- c("BIDEN, JOSEPH R.", "BLANKENSHIP, DON", "JORGENSEN, JO", "None Of These Candidates", "TRUMP, DONALD J.")

# Function to filter and sum votes per candidate

sum\_votes <- function(data, candidate\_name){

  candidate\_data <- filter(data, Candidate == candidate\_name)

  return(sum(candidate\_data$Votes))

}

# Calculate total votes per candidate: Washoe and NV

biden\_votes\_w <- sum\_votes(washoe, candidates[1])

biden\_votes\_n <- sum\_votes(nv, candidates[1])

blank\_votes\_w <- sum\_votes(washoe, candidates[2])

blank\_votes\_n <- sum\_votes(nv, candidates[2])

jorg\_votes\_w <- sum\_votes(washoe, candidates[3])

jorg\_votes\_n <- sum\_votes(nv, candidates[3])

none\_votes\_w <- sum\_votes(washoe, candidates[4])

none\_votes\_n <- sum\_votes(nv, candidates[4])

trump\_votes\_w <- sum\_votes(washoe, candidates[5])

trump\_votes\_n <- sum\_votes(nv, candidates[5])

# Total votes per candidate: Washoe and NV

total\_votes\_w <- sum(biden\_votes\_w, blank\_votes\_w, jorg\_votes\_w, none\_votes\_w, trump\_votes\_w)

total\_votes\_n <- sum(biden\_votes\_n, blank\_votes\_n, jorg\_votes\_n, none\_votes\_n, trump\_votes\_n)

# Total votes by candidate data frame: Washoe and NV

votes\_by\_candidate\_w <- data.frame(

  Candidate = candidates,

  Votes = c(biden\_votes\_w, blank\_votes\_w, jorg\_votes\_w, none\_votes\_w, trump\_votes\_w)

)

votes\_by\_candidate\_n <- data.frame(

  Candidate = candidates,

  Votes = c(biden\_votes\_n, blank\_votes\_n, jorg\_votes\_n, none\_votes\_n, trump\_votes\_n)

)

# Percentage of votes per candidate: Washoe and NV

percents\_w <- votes\_by\_candidate\_w$Votes / total\_votes\_w

percents\_n <- votes\_by\_candidate\_n$Votes / total\_votes\_n

# Percentage of votes per candidate data frame: Washoe and NV

percent\_total\_w <- data.frame(

  Candidate = candidates,

  Percent = percents\_w \* 100

)

percent\_total\_n <- data.frame(

  Candidate = candidates,

  Percent = percents\_n \* 100

)

# Graph generator function

generate\_graph <- function(candidate\_names, votes, title){

  # Data frame for graph

  graph\_data <- data.frame(

    Candidate = candidate\_names,

    Votes = votes

  )

  # Generate graph

  graph <- ggplot(graph\_data, aes(x = Candidate, y = Votes, fill = Candidate)) +

    geom\_col() +

    theme(legend.position = "none") +

    scale\_y\_continuous(breaks = scales::pretty\_breaks(), labels = scales::comma\_format()) +

    scale\_x\_discrete(guide = guide\_axis(n.dodge = 2)) +

    scale\_fill\_manual(values = c(

      "BIDEN, JOSEPH R." = "blue",

      "BLANKENSHIP, DON" = "purple",

      "JORGENSEN, JO" = "yellow",

      "None Of These Candidates" = "black",

      "TRUMP, DONALD J." = "red")) +

    labs(

      title = title,

      x = "Candidate",

      y = "Votes"

    )

  return(graph)

}

# Graph of election results: Washoe and NV

election\_graph\_w <- generate\_graph(

  candidate\_names = candidates,

  votes = c(biden\_votes\_w, blank\_votes\_w, jorg\_votes\_w, none\_votes\_w, trump\_votes\_w),

  title = "2020 Presidential Election Results: Washoe County"

  )

election\_graph\_n <- generate\_graph(

  candidate\_names = candidates,

  votes = c(biden\_votes\_n, blank\_votes\_n, jorg\_votes\_n, none\_votes\_n, trump\_votes\_n),

  title = "2020 Presidential Election Results: Nevada"

)

# Benford's Law distribution

bens\_law <- function(d) log10(1 + 1 / d)  #Formula

digits <- 1:9

bens\_data <- data.frame(Digits = digits, Percentage = bens\_law(digits))

bens\_graph <-

  ggplot(bens\_data,

         aes(x = Digits, y = Percentage)) +

  geom\_line(color = "red") +

  geom\_point(color = "red", size = 3) +

  scale\_x\_continuous(n.breaks = 9) +

  scale\_y\_continuous(n.breaks = 8) +

  labs(title = "Benford's Law",

       x = "Leading Digit",

       y = "Proportion of Occurrence")

# Leading Digits: Washoe

ld\_occur\_w <- sapply(1:9, function(x) sum(washoe$LeadDigit == x))

ld\_percent\_w <- ld\_occur\_w / sum(ld\_occur\_w)

ld\_data\_w <- data.frame(Digits = digits, Percentage = ld\_percent\_w)

ld\_graph\_w <-

  ggplot(ld\_data\_w,

         aes(x = Digits, y = Percentage)) +

  geom\_line(color = "blue") +

  geom\_point(color = "blue", size = 3) +

  scale\_x\_continuous(n.breaks = 9) +

  scale\_y\_continuous(n.breaks = 8) +

  labs(title = "Leading Digit of Total Votes\nin Washoe Presidential Election",

       x = "Leading Digit",

       y = "Proportion of Occurrence")

# Leading Digits: NV

ld\_occur\_n <- sapply(1:9, function(x) sum(washoe$LeadDigit == x))

ld\_percent\_n <- ld\_occur\_n / sum(ld\_occur\_n)

ld\_data\_n <- data.frame(Digits = digits, Percentage = ld\_percent\_w)

ld\_graph\_n <-

  ggplot(ld\_data\_n,

         aes(x = Digits, y = Percentage)) +

  geom\_line(color = "orange") +

  geom\_point(color = "orange", size = 3) +

  scale\_x\_continuous(n.breaks = 9) +

  scale\_y\_continuous(n.breaks = 8) +

  labs(title = "Leading Digit of Total Votes\nin Nevada Presidential Election",

       x = "Leading Digit",

       y = "Proportion of Occurrence")

# Benford's Law comparison: Washoe

table\_wb <- data.frame(Digits = digits,

                      Percentage = c(bens\_law(digits), ld\_percent\_w),

                      Focus = c(rep("Benford", 9), rep("Washoe", 9)))

graph\_wb <-

  ggplot(table\_wb,

         aes(x = Digits, y = Percentage, group = Focus)) +

  geom\_line(aes(color = Focus)) +

  geom\_point(aes(color = Focus)) +

  scale\_x\_continuous(n.breaks = 9) +

  scale\_y\_continuous(n.breaks = 8) +

  scale\_color\_manual(values = c("red", "blue")) +

  labs(title = "Leading Digits of Votes in Washoe v. Benford",

       x = "Leading Digit",

       y = "Proportion of Occurrence")

# Benford's Law comparison: NV

table\_nb <- data.frame(Digits = digits,

                      Percentage = c(bens\_law(digits), ld\_percent),

                      Focus = c(rep("Benford", 9), rep("Nevada", 9)))

graph\_nb <-

  ggplot(table\_nb,

         aes(x = Digits, y = Percentage, group = Focus)) +

  geom\_line(aes(color = Focus)) +

  geom\_point(aes(color = Focus)) +

  scale\_x\_continuous(n.breaks = 9) +

  scale\_y\_continuous(n.breaks = 8) +

  scale\_color\_manual(values=c("red", "orange")) +

  labs(title = "Leading Digits of Votes in Nevada v. Benford",

       x = "Leading Digit",

       y = "Proportion of Occurrence")

# Benford's Law comparison: NV, Washoe, Benford

table\_all <- data.frame(Digits = digits,

                       Percentage = c(bens\_law(digits), ld\_percent\_w, ld\_percent\_n),

                       Focus = c(rep("Benford", 9), rep("Washoe", 9), rep("Nevada", 9)))

graph\_all <-

  ggplot(table\_all,

         aes(x = Digits, y = Percentage, group = Focus)) +

  geom\_line(aes(color = Focus)) +

  geom\_point(aes(color = Focus)) +

  scale\_x\_continuous(n.breaks = 9) +

  scale\_y\_continuous(n.breaks = 8) +

  scale\_color\_manual(values = c("red", "orange", "blue")) +

  labs(title = "Washoe and Nevada v. Benford",

       x = "Leading Digit",

       y = "Proportion of Occurrence")

References

Coleman, M., Higinbotham, S., & Arroyo, A. (2021). Potential for Concordance between Plurality and Instant-Runoff Election Algorithms as a Function of Ballot Dispersion. *Journal of Young Investigators*, *39*(3). https://doi.org/10.22186/jyi.39.3.32-37

Collins, C. J. (2017, April 1). *Using Excel and Benford's Law to detect fraud*. Journal of Accountancy. https://www.journalofaccountancy.com/issues/2017/apr/excel-and-benfords-law-to-detect-fraud.html.

Glen, S. (2020, August 26). Shannon Entropy. Statistics How To. https://www.statisticshowto.com/shannon-entropy/.

Hayes, A. (2021, May 19). Herfindahl-Hirschman Index (HHI). Investopedia. https://www.investopedia.com/terms/h/hhi.asp.

Kenton, W. (2021, July 7). *Monte Carlo Simulation*. Investopedia. <https://www.investopedia.com/terms/m/montecarlosimulation.asp>.

Leschinski, C. (2019, January 31). Vignette: The MonteCarlo Package. <https://cran.r-project.org/web/packages/MonteCarlo/vignettes/MonteCarlo-Vignette.html>.

Nevada Secretary of State. (n.d.). *Election Reporting by Precinct*. nvsos.gov. <https://www.nvsos.gov/electionresults/>

Nevada Secretary of State. (n.d.). *Precinct Level Results*. nvsos.gov. <https://www.nvsos.gov/sos/elections/election-information/precinct-level-results>.