Monte Carlo Presidency

Grant Corkren

2025-01-12

The goal of this method is very simple. Given polling data, we can create a probability mass function of electoral college votes. To create this I am using simulated data from the 2024 presidential election and took a random n=1 sample for each state from standard normal distribution.That is then transformed into a simulated proportion for Republican votes. The simulated sample sizes will be calculated by dividing the total vote count of each state by ten thousand. Polling data where sample sizes of $n > 30$ for each state would be the appropriate methodology. In that case the CLT would apply to the mean of proportions for each state. That would give it a normal distribution for each state where a Monte Carlo simulation could be applied. Both Independence among states and correlation among states will be applied.

The Monte Carlo simulation in the state independence case is done by taking a sample from each normal distribution of the states. One round of a simulation samples the states once and adds the number of electoral college votes based if a state's sampled proportion is above .5. The simulation is done many times and in this case 10,000 times.

The Monte Carlo simulation in the state correlation case is done by taking a sample from a multivariate normal distribution of the states. The covariance matrix is determined by recent previous years elections results as that serves a more accurate measure to correlation among states. One round of a simulation samples the multivariate normal distribution once and translates it back into state's proportions. Then the number the of electoral college votes are determined the same as the independence case.

```
##         State Elec_votes   Trump  Harris REPUBLICANPERCENTAGE
## 1    Alabama          9 1462616  772412            0.6544061
## 2     Alaska          3  184458  140026            0.5684656
## 3    Arizona         11 1770242 1582860            0.5279416
## 4   Arkansas          6  759241  396905            0.6566999
## 5 California         54 6081697 9276179            0.3959986
## 6   Colorado         10 1377441 1728159            0.4435346
```

```
## # A tibble: 6 x 5
##   State      Elec_votes REPUBLICANPERCENTAGE simulated_sample_size
##   <chr>           <dbl>                <dbl>                 <dbl>
## 1 Alabama             9                0.654                   224
## 2 Alaska              3                0.568                    32
## 3 Arizona            11                0.528                   335
## 4 Arkansas            6                0.657                   116
## 5 California         54                0.396                  1536
## 6 Colorado           10                0.444                   311
##   simulated_proportion
##                  <dbl>
## 1                0.635
## 2                0.585
## 3                0.505
```

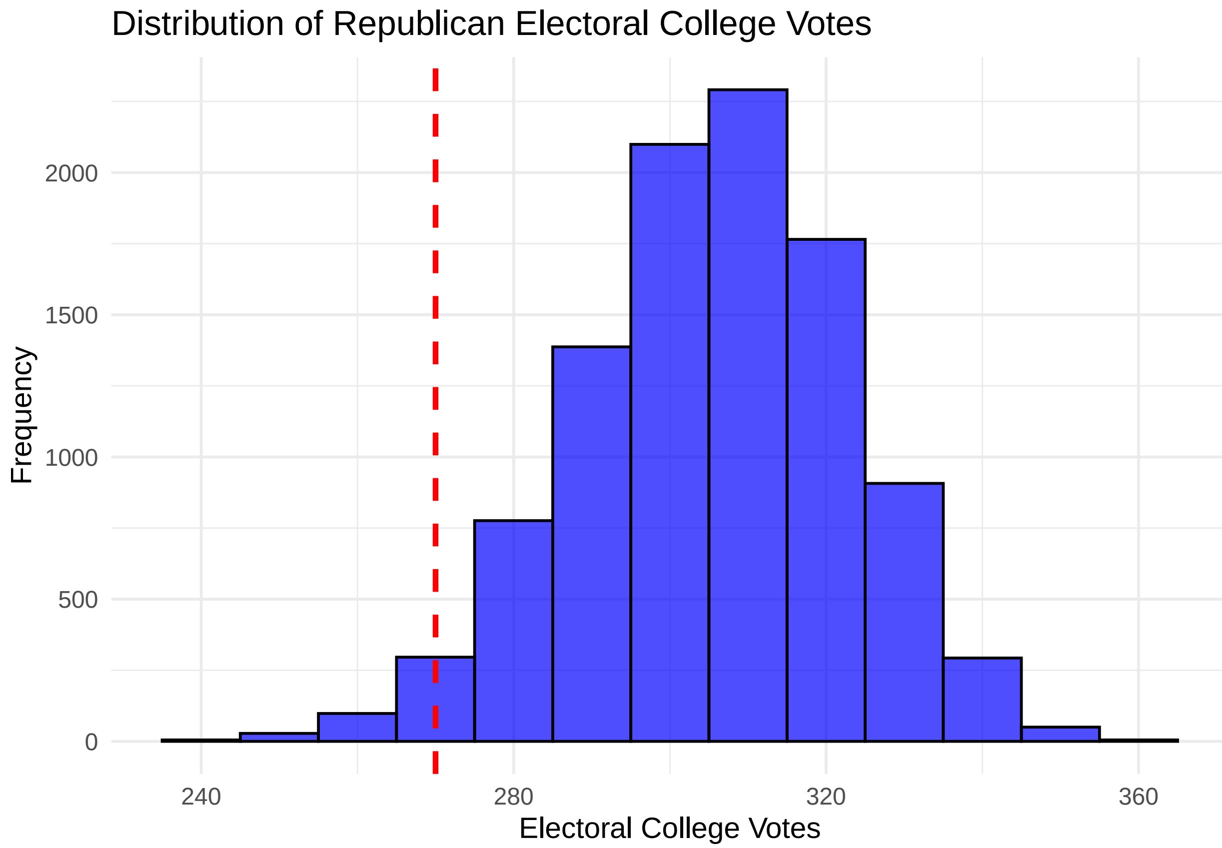
```
## 4                0.727
## 5                0.400
## 6                0.420
```

This is the simulated sample proportions from the results of the 2024 election and is statistical accurate in theory of what could happen in polling. In reality you would need to account for "dropout", providing false information, unaccounted variable, etc.

Assumed Independence among States

This method creates a univariate normal distribution for all states with the simulated means and their variances.

```
# Monte Carlo simulation
set.seed(1)
n_simulations <- 10000


simulation_results <- replicate(n_simulations, {
  simulated_values <- pmin(pmax(rnorm(nrow(sim_data), mean = sim_data$simulated_proportion,
                                      sd = sqrt(sim_data$variance)), 0), 1)

  total_electoral_votes <- sum(sim_data$Elec_votes[simulated_values > 0.5])

  return(total_electoral_votes)
})

simulation_df <- data.frame(total_electoral_votes = simulation_results)



ggplot(simulation_df, aes(x = total_electoral_votes)) +
  geom_histogram(binwidth = 10, fill = "blue", color = "black", alpha = 0.7) +
   geom_vline(xintercept = 270, color = "red", linetype = "dashed", size = 1) +
  labs(
    title = "Distribution of Republican Electoral College Votes",
    x = "Electoral College Votes",
    y = "Frequency"
  ) +
  theme_minimal()
```



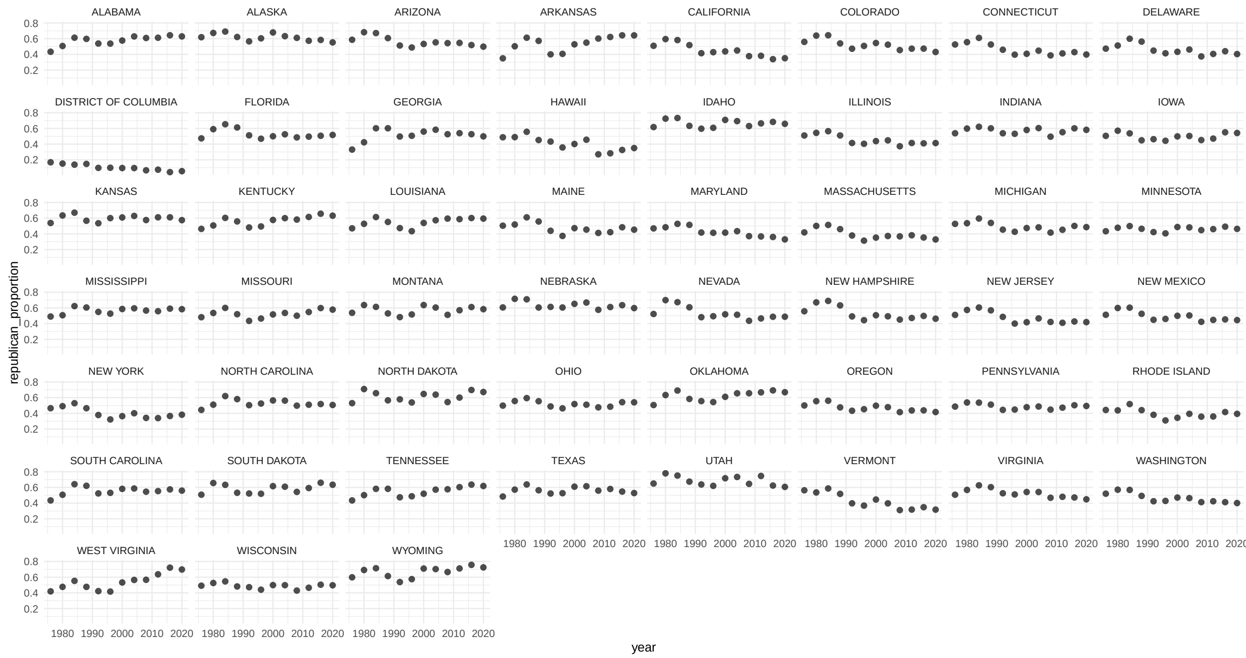

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   235.0   295.0   307.0   306.1   318.0   364.0
```

```
## [1] 0.0235
```

This simulation shows Trump winning Harris winning the presidency 2.35%.

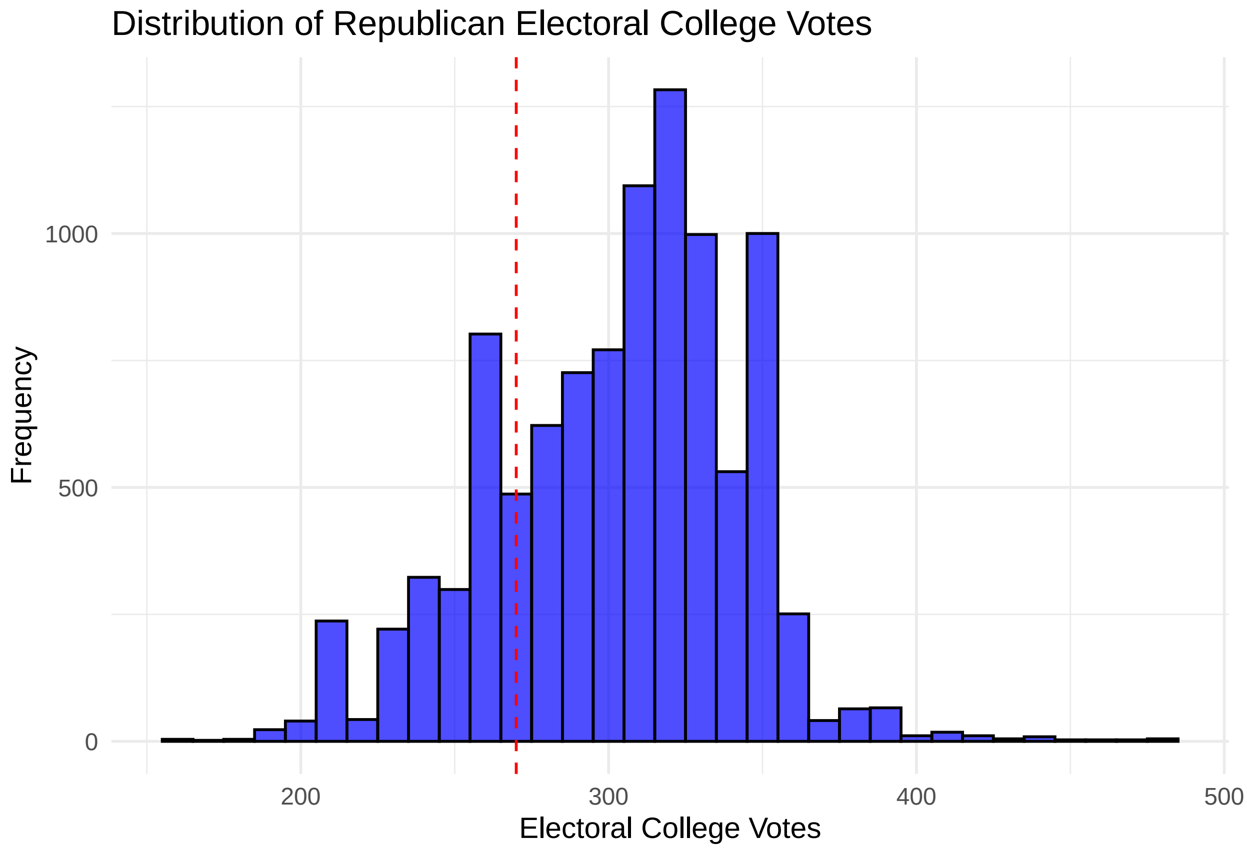
Assumed Dependence Among States

This method creates a multivariate normal distribution with the means of simulation proportion and a covariance matrix of past elections proportions. A Monte Carlo simulation is then conducted with this distribution.



Lets choose the past four elections for our covariance matrix

```
# Monte Carlo simulation
library(MASS)

set.seed(1)
n_simulations <- 10000

simulation_results <- replicate(n_simulations, {
  # multivariate normal distribution
  simulated_proportions <- mvrnorm(1, mu = sim_data$simulated_proportion, Sigma = cov_matrix)

  simulated_proportions <- pmin(pmax(simulated_proportions, 0), 1)

  total_electoral_votes <- sum(sim_data$Elec_votes[simulated_proportions > 0.5])

  return(total_electoral_votes)
})

simulation_df <- data.frame(total_electoral_votes = simulation_results)


library(ggplot2)
ggplot(simulation_df, aes(x = total_electoral_votes)) +
  geom_histogram(binwidth = 10, fill = "blue", color = "black", alpha = 0.7) +
   geom_vline(xintercept = 270, color = "red", linetype = "dashed") +
  labs(
    title = "Distribution of Republican Electoral College Votes",
    x = "Electoral College Votes",
    y = "Frequency"
  ) +
  theme_minimal()
```




```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   155.0   277.0   309.0   303.3   329.0   483.0
```

```
## [1] 0.2183
```

This simulation show Democrats winning 21.83%.

Next Steps

This was done with simulated data and proves as an option to predicting the presidency. Testing and evaluation will need to be done to determine if it is a viable option.

Testing and Evaluation

This method of prediction would require evaluation towards the polling estimates. This could be done by finding biases and adjusting biases, aggregating different polling sources with bias adjusted. Adjusting for bias would include finding biases among polling sources and potentially adjusting an underestimate of variance.

Underestimate of Variance: A common problem with political polling is incorrect information (aka lying to pollsters). While adjusting for bias of mean may seem to correct this, an adjustment of variance may be required.

Conclusion

This method of prediction could prove useful but could also improved upon in many ways. Improvements include accounting for other political parties and increasing the number of simulations.