Precept 6

R functions

This week's precept assignment will focus on the construction of confidence intervals, in addition to providing further practice with loops. You will find the following functions helpful.

`matrix`:

The `matrix` function will create a matrix. It takes the following form: `matrix(data, nrow, ncol)` where `data` is what we want to populate our matrix with, `nrow` is the number of desired rows, and `ncol` is the number of desired columns. The `data` argument is optional (hint: we can populate a matrix with missing values). You can read more about matrices in QSS on page 108.

```
# create an empty matrix with ten rows and three columns
my.matrix <- matrix(NA, nrow = 10, ncol = 3)
```

`qnorm`:

`qnorm` returns quantiles from a normal distribution. `qnorm` takes the following form: `qnorm(p, mean, sd)` where `p` is a probability, `mean` is the mean of the distribution, and `sd` is the standard deviation of the distribution. If the mean and standard deviation are not specified, `qnorm` will set the mean to 0 and the standard deviation to 1 by default. You can read further about `qnorm` in QSS on page 328.

```
z.score <- qnorm(p = .5, mean = 0, sd = 1)
```

Revisiting the 2016 election

In the 2016 US presidential election, the Republican candidate Donald Trump surprised many by defeating the Democratic candidate Hillary Clinton. In particular, even right before the election, polls were predicting that Hillary Clinton would win the election by a comfortable margin. Why did preelection polls fail to predict the election outcome? We analyze the polling data, taken from Huffington post, that include the most recent polls leading up to the election. The dataset we will be analyzing (`data/polls16.csv`) has 1395 observations, each representing a different poll, and includes the following variables:

Name	Description
`id`	Poll ID
`state`	U.S. state where poll was fielded
`Clinton`	The poll's estimated level of support for Hillary Clinton
`Trump`	The poll's estimated level of support for Donald Trump
`Undecided`	The poll's estimated percentage of undecided voters
`days_to_election`	Number of days before November 4, 2016.
`electoral_votes`	Number of electoral votes allocated to the state where the poll was fielded (a state-level variable)
`sample_size`	The number of people surveyed in the poll

We will also analyze a dataset (`results16.csv`) which contains the state-by-state voteshare for each candidate collected from CNN. This data set has the following variables:

Name	Description
`State`	U.S. state where poll was fielded
`Clinton`	The percent of votes Clinton received
`Trump`	The percent of votes Trump received

Question 1

We will begin by calculating the predicted vote share for Hillary Clinton by using the average support rate of the most recent (based on the `days_to_election` variable) polls for each state. Also, if there are multiple polls on the same day, find the average sample size. What is the bias of prediction across states? What is the root mean squared error? Create a histogram of prediction error. Briefly interpret these results.

Answer 1

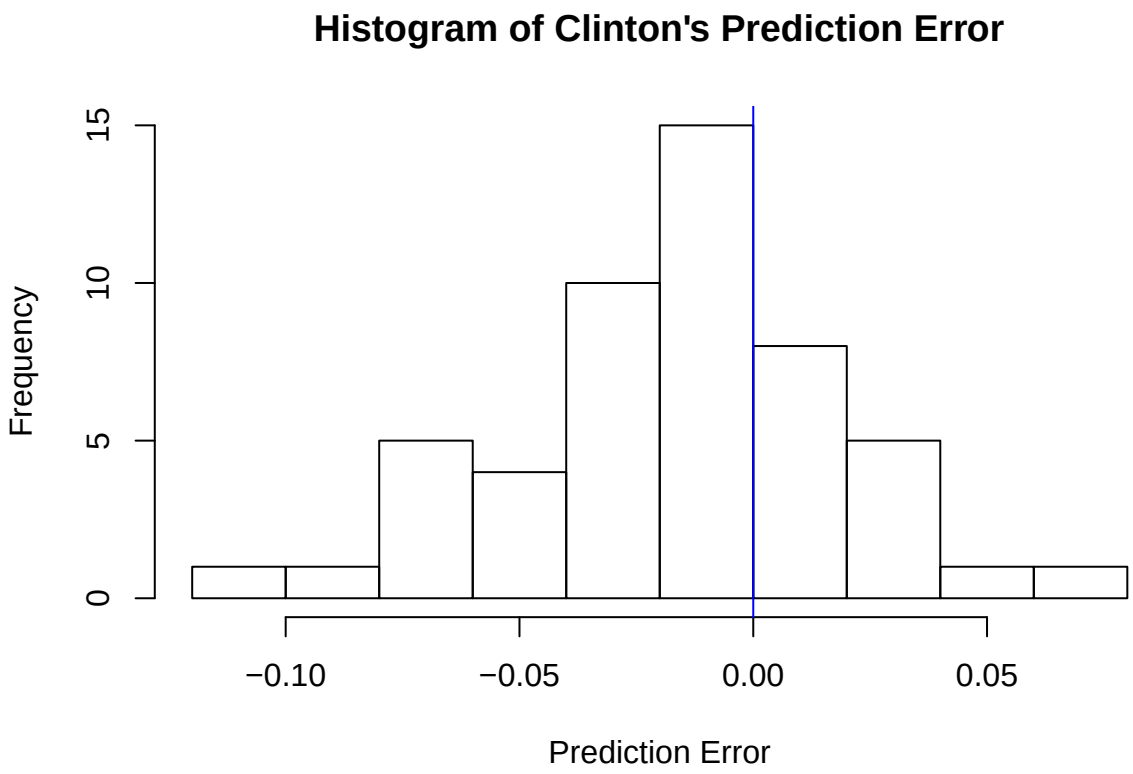
```
# load data
results <- read.csv("data/results16.csv")
polls <- read.csv("data/polls16.csv")
state.names <- unique(polls$state)

## Predictions for Clinton
n <- rep(NA, 51)
poll.pred.C <- matrix(NA, nrow = 51, ncol = 3)
row.names(poll.pred.C) <- as.character(state.names)
for (i in 1:51) {
  ## subset the ith state
  state.data <- subset(polls, subset = (state == state.names[i]))
  ## subset the latest polls within the state
  latest <- state.data$days_to_election == min(state.data$days_to_election)
  ## compute the mean of latest polls and store it
  poll.pred.C[i, 1] <- mean(state.data$Clinton[latest])
  n[i] <- mean(state.data$sample_size[latest])
}

## Calculate Bias
Clinton.bias <- poll.pred.C[,1] - results$Clinton
mean(Clinton.bias)
```

```
[1] -0.01672549
```

```
## Root Mean Squared Error
sqrt(mean((Clinton.bias)^2))
```

```
[1] 0.03820633
```

```
## Histogram of Bias
hist(Clinton.bias, xlab = "Prediction Error",
     main = "Histogram of Clinton's Prediction Error")
abline(v = 0, col = "blue")
```



The polls under-predicted her voteshare by only 1.67 percentage points. The RMSE is around 3.82 percentage points which tells us there is a substantial amount of variation in the prediction error. Additionally the histogram demonstrates that the prediction error is pretty evenly distributed around 0. In other words, the bias is relatively small.

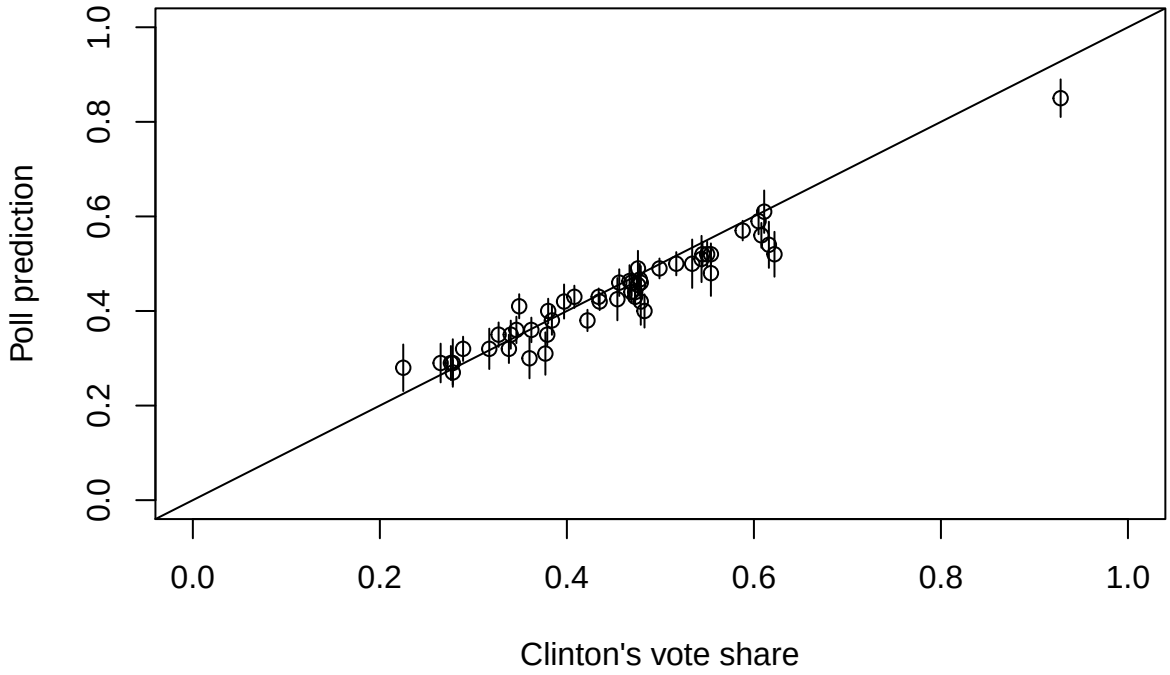
Question 2

Construct 95% confidence intervals for each of the state-level predictions obtained in the previous question. Plot the prediction against the true result with a 45-degree line to indicate whether the polls under or over predicted Clinton's voteshare. What proportion of the actual election results are contained within these confidence intervals? Does the coverage improve if we correct for the bias of prediction obtained in the previous question? Briefly interpret your results.

Answer 2

```
## 95% Confidence intervals
alpha <- 0.05

## CI for Clinton
s.e. <- sqrt(poll.pred.C[, 1] * (1 - poll.pred.C[, 1]) / n)
poll.pred.C[, 2] <- poll.pred.C[, 1] - qnorm(1 - alpha / 2) * s.e.
poll.pred.C[, 3] <- poll.pred.C[, 1] + qnorm(1 - alpha / 2) * s.e.

## Plot results for Clinton
plot(results$Clinton, poll.pred.C[, 1], xlim = c(0, 1), ylim = c(0, 1),
     xlab = "Clinton's vote share", ylab = "Poll prediction")
abline(0, 1)
```

```
for (i in 1:51) {
    lines(rep(results$Clinton[i], 2), c(poll.pred.C[i, 2], poll.pred.C[i, 3]))
}
```




```
## proportion of confidence intervals that contain the election day outcome
mean((poll.pred.C[, 2] <= results$Clinton) &
         (poll.pred.C[, 3] >= results$Clinton))
```

```
[1] 0.627451
```

```
## bias corrected estimate for Clinton
poll.bias.C <- poll.pred.C[, 1] - mean(Clinton.bias)

## bias corrected standard error
s.e.bias.C <- sqrt(poll.bias.C * (1 - poll.bias.C) / n)

## bias-corrected 95% confidence interval
ci.bias.C.lower <- poll.bias.C - qnorm(1 - alpha / 2) * s.e.bias.C
ci.bias.C.upper <- poll.bias.C + qnorm(1 - alpha / 2) * s.e.bias.C

## proportion of bias-corrected CIs that contain the election day outcome
mean((ci.bias.C.lower <= results$Clinton) &
         (ci.bias.C.upper >= results$Clinton))
```

```
[1] 0.627451
```

Before correcting for bias, our confidence intervals contained approximately 63% of the true results, after correcting for bias the confidence intervals covered 63% of the true results. In other words, correcting for bias does not improve our predictions for Clinton. This is because the bias is small. The result suggests that these confidence intervals are too narrow, leading to over-confidence.

Question 3

Repeat the analysis from Questions 1 and 2 for Donald Trump. Compare and interpret your results.

Answer 3

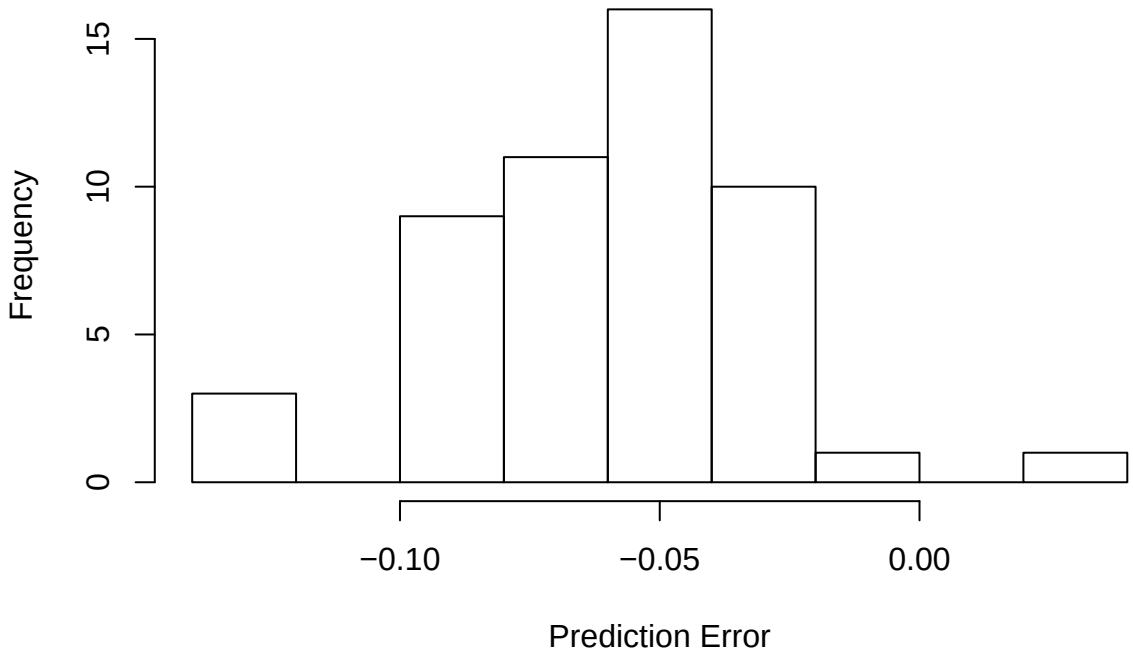
```
# First, the analysis from Question 1:

## Predictions for Trump
poll.pred.T <- matrix(NA, nrow = 51, ncol = 3)
row.names(poll.pred.T) <- as.character(state.names)
for (i in 1:51) {
  ## subset the ith state
  state.data <- subset(polls, subset = (state == state.names[i]))
  ## subset the latest polls within the state
  latest <- state.data$days_to_election == min(state.data$days_to_election)
  ## compute the mean of latest polls and store it
  poll.pred.T[i, 1] <- mean(state.data$Trump[latest])
}

## Bias
Trump.bias <- poll.pred.T[,1] - results$Trump
mean(Trump.bias)
```

```
[1] -0.05840523
```

```
## Root Mean Squared Error
sqrt(mean((Trump.bias)^2))
```

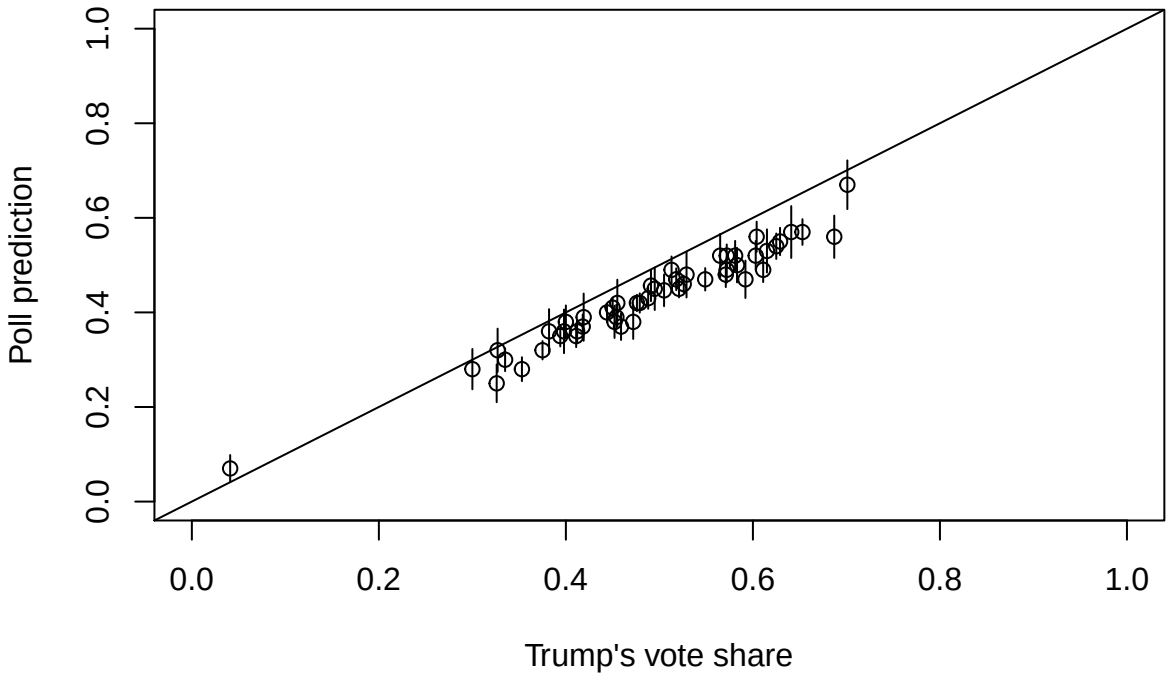
```
[1] 0.06519496
```

```
## Histogram of Bias
hist(Trump.bias, xlab = "Prediction Error",
     main = "Histogram of Trump's Prediction Error")
```



We can see that in contrast with the errors in Clinton's predictions, the error for Trump is consistently underestimating his true vote share. Both the average bias) and the RMSE) are larger for Trump than for Clinton, indicating the polls performed consistently poorly for him, both in terms of magnitude and direction.

```
# Now the analysis from Question 2:

## CI for Trump
s.e.2 <- sqrt(poll.pred.T[, 1] * (1 - poll.pred.T[, 1]) / n)
poll.pred.T[, 2] <- poll.pred.T[, 1] - qnorm(1 - alpha / 2) * s.e.2
poll.pred.T[, 3] <- poll.pred.T[, 1] + qnorm(1 - alpha / 2) * s.e.2

## Plot results for Trump
plot(results$Trump, poll.pred.T[, 1], xlim = c(0, 1), ylim = c(0, 1),
     xlab = "Trump's vote share", ylab = "Poll prediction")
abline(0, 1)
for (i in 1:51) {
    lines(rep(results$Trump[i], 2), c(poll.pred.T[i, 2], poll.pred.T[i, 3]))
}
```




```
## Confidence intervals containing true result
mean((poll.pred.T[, 2] <= results$Trump) &
         (poll.pred.T[, 3] >= results$Trump))
```

```
[1] 0.1960784
```

```
## bias corrected estimate for Trump
poll.bias.T <- poll.pred.T[, 1] - mean(Trump.bias)

## bias corrected standard error
s.e.bias.T <- sqrt(poll.bias.T * (1 - poll.bias.T) / n)

## bias-corrected 95% confidence interval
ci.bias.T.lower <- poll.bias.T - qnorm(1 - alpha / 2) * s.e.bias.T
ci.bias.T.upper <- poll.bias.T + qnorm(1 - alpha / 2) * s.e.bias.T
## proportion of bias-corrected CIs that contain the election day outcome
mean((ci.bias.T.lower <= results$Trump) &
         (ci.bias.T.upper >= results$Trump))
```

```
[1] 0.8039216
```

In contrast to Question 1, the bias for Trump is highly systematic. Since all of the polls under-predicted, correcting for this bias greatly improved the coverage of the true results.

Question 4

We will now explore one hypothesis for Trump's surprising victory in the election: a large proportion of voters whom polls classified as "undecided" cast ballots for Trump on the election day. These voters may not have wanted to admit they supported Trump when answering surveys. It is also possible that they made up their minds right before the election following the FBI announcements. Although we do not have individual data necessary for directly testing this hypothesis, we will predict Trump's electoral college votes under the assumption that all undecided voters voted for Trump. Specifically, run 1000 Monte Carlo simulations under

this assumption by computing the probability of winning each state j for Trump as follows:

$$P(\text{Trump wins state } j) = P(Z_j > 0.5)$$

where Z_j is a Normal random variable with mean $\hat{p}_j$ and standard deviation $\sqrt{\hat{p}_j(1-\hat{p}_j)/n_j}$ with n_j being the sample size of the latest poll for that state and

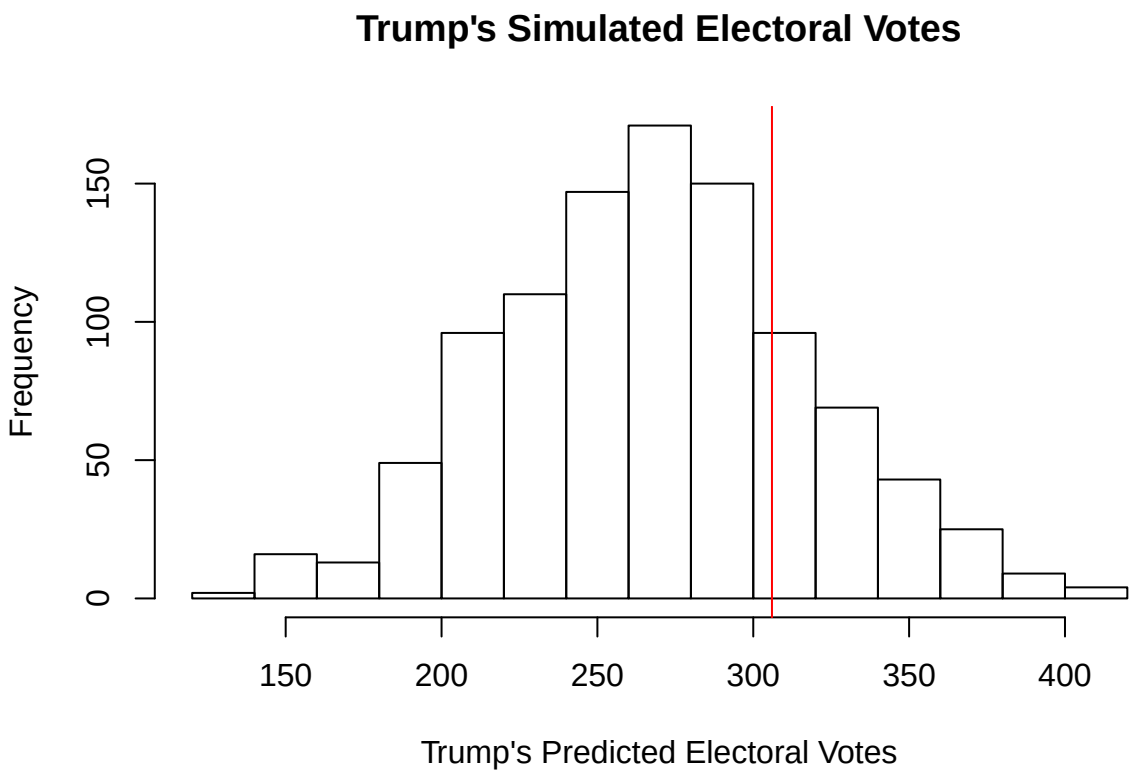
$$\hat{p}_j = \frac{\text{Trump supporters + undecided respondents}}{\text{Trump supporters + Clinton supporters + undecided respondents}}.$$

Simulate Trump's electoral vote outcomes by sampling its winner using the above probability. In other words, first calculate $\hat{p}_j$ for each state j, then run a simulation where you sample whether Trump wins that state using a draw from a Bernoulli Distribution with the probability of success equal to the above probability $P(\text{Trump wins state } j)$. Present the results using a histogram with a red vertical line representing the actual outcome (Trump = 306). Additionally report the point estimate, standard error, and its 95% confidence interval for the total number of electoral votes for Trump.

Answer 4

```
ev <- tapply(polls$electoral_votes, polls$state, mean)

## Calculate the average undecided voter proportion according to latest poll(s)
poll.pred.UD <- rep(NA, 51)
for (i in 1:51) {
  state.data <- subset(polls, subset = (state == state.names[i]))
  latest <- state.data$days_to_election == min(state.data$days_to_election)
  poll.pred.UD[i] <- mean(state.data$Undecided[latest])
}


## Create new predictions for Trump and Clinton
new.pred.T <- (poll.pred.T[,1] + poll.pred.UD) /
  (poll.pred.T[,1] + poll.pred.C[,1] + poll.pred.UD)

## Calculate probability Trump wins the state based on normal distribution
Trump.prob <- qnorm(.5,
                    mean = new.pred.T,
                    sd = sqrt((1 - new.pred.T) * new.pred.T/n),
                    lower.tail = FALSE)

## Simulations
sims <- 1000
Trump.ev <- rep(0, sims)

for (i in 1:sims) {
  ## Draw whether Trump wins the state (0,1)
  Trump.draws <- rbinom(51, size = 1, prob = Trump.prob)
  ## sums state's electoral college votes if Trump wins the state
  Trump.ev[i] <- sum(ev*Trump.draws)
}

## Histogram of Simulation results
hist(Trump.ev, main = "Trump's Simulated Electoral Votes",
```

```
     xlab = "Trump's Predicted Electoral Votes")
abline(v = 306, col = "red")
```




```
## point estimate
mean.t <- mean(Trump.ev)
mean.t
```

```
[1] 268.718
```

```
## Standard error and 95% confidence intervals
se.t <- sqrt(var(Trump.ev))
se.t
```

```
[1] 49.50231
```

```
Trump.sim.upper <- mean.t + qnorm(1 - alpha / 2) * se.t
Trump.sim.lower <- mean.t - qnorm(1 - alpha / 2) * se.t

c(round(Trump.sim.lower), round(Trump.sim.upper))
```

```
[1] 172 366
```

As we can see, the simulation in which Trump gets all the undecided votes slightly under-predicts for Trump electoral votes. This simulation may indicate that a large number of "undecided" voters either decided at the last minute to vote for Trump or did not disclose their true voting intentions to the pollsters. Additionally, we see that the true results (Trump = 306) is within the confidence interval. While this is not an exhaustive test of the various hypotheses that may explain this surprising election outcome, it is an interesting result and could explain why the polls drastically underpredicted Trump's performance.