# SML 201 Mini Project 3: 2016 U.S. Presidential Election

Solutions

*Bill Haarlow & Weston Carpenter*

*2019-12-03*

**Project 3 is due by 11:59pm on Monday November 25.** Please submit both a .Rmd and a .pdf file on Blackboard by the deadline **and** drop off a hard copy of the pdf file at 26 Prospect Avenue by 5pm of the **next weekday** of the due date. To look for the drop-off cabinet, after you enter the building turn to the left to enter the lounge area and the file cabinet is to your right with an open slot with the label "SML 201 Homework"; note that the building might be locked after 6pm and on the weekends. You are also welcome to bring your PDF copy to any lecture **before** the deadline and I will drop off the copy for you.

Late **projects** will be penalized at intervals rounded up to multiples of 24 hours. For example, if you are 3 hours late, 10% off or if you are 30 hours late, 20% off.

This project can be completed in groups of up to 3 students. It is okay to work by yourself, if this is preferable. You are welcome to get help (you can either ask questions on Piazza or talk to instructors in person during office hours) from instructors; however, please do not post code/solutions on Piazza on a public post.

You are encouraged to get help from the instructors (either through Piazza or in person) if you need help to understand the definitions of the variables of the dataset or the procedure of the experiment.

When working in a group it is your responsibility to make sure that you are satisfied with all parts of the report and the submission is on time (e.g., we will not entertain arguments that deficiencies are the responsibility of other group members). We expect that you each work independently first and then compare your answers with each other once you all finish, or you all work together on your own laptops. Failing to make contributions and then putting your name on a report will be considered a violation of the honor code. **Please do not divide work among group mates.** Everyone in your group is responsible for being able to explain the solutions in the report.

For all parts of this problem set, you **MUST** use R commands to print the output as part of your R Markdown file. You are not permitted to find the answer in the R console and then copy and paste the answer into this document.

**If you are completing this project in a group**, please have only **one** person in your group turn in the .Rmd and .pdf files; the other person in your group should turn in the list of the people in your group in the *Text Submission* field on the submission page. This means that **everyone should make a submission**–either a file-upload or a text submission–regardless of whether you are working in a group or not.

**The physical pdf report that you drop off and the .pdf file that you submit on Blackboard should be identical. Modifying your report after the deadline could result in a penalty as much as getting a zero score for the assignment.**

---

Please type your name(s) after "Digitally signed:" below the honor pledge to serve as digital signature(s). Put the pledge and your signature(s) at the beginning of each document that you turn in.

I pledge my honor that I have not violated the honor code when completing this assignment.

Digitally signed: Bill Haarlow & Weston Carpenter

---

**In order to receive full credits, please have sensible titles and axis labels for all your graphs and adjust values for all the relevant graphical parameters so that your plots are informative. Also, all answers must be written in complete sentences.**

**(-3 pts each if any of these are not satisfied: code runs, code has annotations, answers are in complete sentences)**

**Before you start: loops are not allowed for this project. Please report all numerical answers to 2 digits after the decimal. Remember not to round intermediate calculations and please avoid hard-coding.**

**Please submit a color copy for the physical pdf file.**

---

**Please read and work through all the parts in *Project 3 Hints*. Doing so will help you avoid making some common mistakes for this project. Also, you will need to have the `maps` package installed for this project.**

## Objective of this project

The outcome of the 2016 presidential election was a big surprise to many of us; all 6 probabilistic and 3 non-probabilistic models published on New York Times missed the election result (click here for reference). Even on November 8, 2016 (the election day) before the voting results were out Clinton was still estimated to have a winning chance of 71.4% (compared to 28.6% for Trump) by Nate Silver, a statistician who had a track record of predicting the presidential election outcomes correctly until 2016 (click here for reference).

Many believe that the main factor contributes toward the failure of these models was the bias in the polling data (click here for reference). In this project we will explore the problems in the polling data that could potentially affect the accuracy of the presidential election prediction.

## Background info and datasets used in this project

We will use two datasets: `oct_poll.Rdata` and `2016_election_result_csv.csv` in this project.

The dataset `oct_poll.Rdata` is a subset of the dataset from this website (https://www.r-bloggers.com/fivethirtyeights-polling-data-for-the-us-presidential-election/). `oct_poll.Rdata` contains poll data collected from polls that were taken between Oct. 1, 2016 and Oct. 27, 2016. The original dataset was first posted on fivethirtyeight.com, a website created by Nate Silver. (Note: Even though Silver's prediction also missed the truth, Nate Silver's model was the most robust one among all the models published on *New York Times* for predicting the 2016 presidential election result; a robust model is a model that performs well even when some of the assumptions of the model were slightly violated.)

## How did all 6 probabilistic models published on New York Times missed the election result?

In this section we will study the patterns of the errors in the poll data. We would like to answer these two questions:

- Do the errors in the poll dataset seem random? Are they distributed symmetrically around zero?
- Among the states with CI estimates that do not cover zero (i.e., among the states that we were confident about who the winning candidates for the states would be), how many of their CIs actually predicted the correct state winner?

## (18 pts) Question 1

In 2016 the election day was Nov. 8. `oct_poll.Rdata` contains poll data collected from polls that ended between Oct. 1, 2016 and Oct. 27, 2016; the dataset has 752 rows and 6 columns. If you forget how to read in this dataset try running `?load` to get help. You do not need to supply the values for the `envir` and `verbose` arguments in `load()`.

The descriptions of the column names are:

- **state**: The state which was polled;
- **grade**: The rating of the company taking the poll;
- **samplesize**: Sample size of the poll;
- **rawpoll_clinton**: The percentage of people favoring Clinton in the sample;
- **rawpoll_trump**: The percentage of people favoring Trump in the sample;
- **end.date**: The closing date of the poll.

Note that there are multiple entries for each state since there are multiple poll agencies for each state. Also note that on each row the percentages for Clinton and Trump do not add up to 100% since there are other candidates.

### (1 pt to load data set)

```
# Loads data set
load("/Users/billhaarlow/Desktop/SML201/Projects/Project3/oct_poll.rdata")
```

### (17 pts) Part a

Make a data frame `poll.oct2016`(see examples on how to create a data frame in Chapter 2.3) ; this data frame should have 4 columns:

- `state`
- `samplesize`
- `rawpoll_clinton`
- `rawpoll_trump`.

`poll.oct2016` should aggregate the data in `oct_poll` to the state level (we want to combine multiple polls within the same state into one sample so that we have a sample that is as large as possible.) and express `rawpoll_clinton` and `rawpoll_trump` as proportions instead of percentages. You should keep `District of Columbia` even though technically it is not a state. `poll.oct2016` should have one row for each state. You need to be careful about how to aggregate the percentages; e.g., suppose state 1 has two polls: poll 1 has a sample of 100 and 30% of the people in the sample voted for Clinton and poll 2 has a sample of 400 and 60% of the people in the sample voted for Clinton; then, the aggregated proportion of people voted for Clinton in the polls for state 1 should be (100 x .3 + 400x.6)/(100+400) = .54.

Hint: If you do not remember how to do this, it is time to review what object type the `tapply()` outputs are of; also remember that `as.vector()` removes element names of the input object by default; thus, you will need to assign the element names back to the vector after you convert a R object into a vector by using `as.vector()`; see Chapter 2.3 for a review on both of these two concepts.

```
# Creates `poll.oct2016` and aggregates the appropriate data
poll.oct2016 = data.frame(state = unique(oct_poll$state), samplesize = as.vector(tapply(X = oct_poll$sa
    INDEX = oct_poll$state, FUN = sum)), rawpoll_clinton = as.vector(tapply(X = (oct_poll$rawpoll_clint
    oct_poll$samplesize), INDEX = oct_poll$state, FUN = sum))/as.vector(tapply(X = oct_poll$samplesize,
    INDEX = oct_poll$state, FUN = sum)), rawpoll_trump = as.vector(tapply(X = (oct_poll$rawpoll_trump/1
    oct_poll$samplesize), INDEX = oct_poll$state, FUN = sum))/as.vector(tapply(X = oct_poll$samplesize,
    INDEX = oct_poll$state, FUN = sum)))
str(poll.oct2016)
'data.frame':   51 obs. of  4 variables:
 $ state          : Factor w/ 51 levels "Alabama","Alaska",..: 10 5 39 26 3 30 34 16 32 44 ...
 $ samplesize     : int  4021 2958 14609 3400 14515 14341 3981 1957 1376 40411 ...
 $ rawpoll_clinton: num  0.296 0.328 0.423 0.338 0.525 ...
 $ rawpoll_trump  : num  0.541 0.391 0.414 0.502 0.281 ...
```

## (16 pts) Question 2

### (10 pts) Part a

'2016_election_result_csv.csv' contains the presidential election results for 2016. Read in the dataset `2016_election_result_csv.csv` and name this dataset `true.perc`. Use the `merge()` function in R to merge `true.perc` with `poll.oct2016` by using the state names (you can review how to use the `merge()` function in Chapter 2.3). In addition, add the following columns to the data frame after the merging (see Chapter 2.3 again for how to use the `mutate()` function):

- **poll.diff** = rawpoll_trump - rawpoll_clinton
- **vote.diff** = trump.vote - clinton.vote
- **deviation** = poll.diff - vote.diff

Name the resulting data frame `election`. `election` should have 51 rows and 9 columns.

```
library(dplyr)
# Reads the .csv file and creates `true.perc`
true.perc = read.csv("/Users/billhaarlow/Desktop/SML201/Projects/Project3/2016_election_result_csv.csv"
# Merges the data frames into one
merged.data = merge(x = poll.oct2016, y = true.perc, by.x = "state",
    by.y = "state")
# Mutates the data frame
election = mutate(merged.data, poll.diff = rawpoll_trump - rawpoll_clinton,
    vote.diff = trump.vote - clinton.vote, deviation = poll.diff -
        vote.diff)
str(election)
'data.frame':   51 obs. of  9 variables:
 $ state          : Factor w/ 51 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ samplesize     : int  2391 6793 14515 1911 2958 6591 27295 3088 17664 4021 ...
 $ rawpoll_clinton: num  0.304 0.4 0.525 0.499 0.328 ...
 $ rawpoll_trump  : num  0.564 0.451 0.281 0.339 0.391 ...
 $ clinton.vote   : num  0.344 0.366 0.451 0.337 0.617 ...
 $ trump.vote     : num  0.621 0.513 0.487 0.606 0.316 ...
 $ poll.diff      : num  0.2598 0.0516 -0.2442 -0.1602 0.063 ...
 $ vote.diff      : num  0.2773 0.1473 0.0355 0.2692 -0.3011 ...
 $ deviation      : num  -0.0174 -0.0957 -0.2797 -0.4294 0.3641 ...
dim(election)
[1] 51  9
```

### (6 pts) Part b

Choose the correct answer for each of the following questions:

(2 pt) (i.) For a particular state if `poll.diff` is positive this means

(A.) for the people in the *combined poll* for the state there are more people supporting Trump than supporting Clinton;
(B.) for the people in the *combined poll* for the state there are less people supporting Trump than supporting Clinton;
(C.) for the people who casted their votes in the *actual election* there are more voters supporting Trump than

supporting Clinton within the state;
(D.) for the people who casted their votes in the *actual election* there are less voters supporting Trump than supporting Clinton within the state;
(E.) Trump won the actual election at the state level;
(F.) Clinton won the actual election at the state level.

Answer: A.

(4 pt) (ii.) For a particular state if `deviation` is positive this means

(A.) Trump won the actual election at the state level;
(B.) Trump lost the actual election at the state level;
(C.) at the state level the poll over-estimated how well Trump would do in the actual election compared to Clinton;
(D.) at the state level the poll under-estimated how well Trump would do in the actual election compared to Clinton.

Answer: C.

## (66 pts) Question 3

### (23 pts) Part a

Create the following vectors and add them as the columns to `election`:

- **std.unit**: For the $i^{th}$ state `std.unit[i]` is `poll.diff[i]` in standard units assuming `vote.diff[i]` is the mean for `poll.diff[i]`; e.g., if `poll.diff[1]` is 3 SE(poll.diff for state i) above `vote.diff[i]`, `std.unit[i]` should be 3; if `poll.diff[1]` is 2.5 SE(poll.diff for state i) below `vote.diff[1]`, `std.unit[1]` should be -2.5. (See exercise 1.h in Precept 9.)
- **same.sign**: A logical vector whose $i^{th}$ element should be `TRUE` if both `poll.diff[i]` and `vote.diff[i]` have the same sign (i.e., either both positive or both negative), and `FALSE` otherwise.
- **significant**: A logical vector that shows `TRUE` if the 95% CI for estimating (proportion of Trump supporters - proportion of Clinton supporters) does not cover 0 and `FALSE` otherwise.

```
# Creates sample using `mapply`
clinton = with(election, mapply(rep, -1, as.vector(tapply(X = rawpoll_clinton *
    samplesize, INDEX = state, FUN = mean))))
trump = with(election, mapply(rep, 1, as.vector(tapply(X = rawpoll_trump *
    samplesize, INDEX = state, FUN = mean))))
other = with(election, mapply(rep, 0, as.vector(tapply(X = (1 -
    rawpoll_clinton - rawpoll_trump) * samplesize, INDEX = state,
    FUN = mean))))
# Outputs the individual SE for each state
sd.func = function(x) {
    z = sd(c(clinton[[x]], trump[[x]], other[[x]]))/sqrt(election[x,
        2])
    return(z)
}
se = as.numeric(lapply(X = c(1:51), FUN = sd.func))
# Creates std.unit
std.unit = (election$poll.diff - election$vote.diff)/se
std.unit
```

```
# Creates same.sign
pos = as.numeric(election$poll.diff > 0) + as.numeric(election$vote.diff >
    0)
both.pos = as.numeric(pos == 2)
```

```
neg = as.numeric(election$poll.diff < 0) + as.numeric(election$vote.diff <
    0)
both.neg = as.numeric(neg == 2)
same.sign = (both.pos == 1) | (both.neg == 1)
same.sign
```

```
# Creates significant
m = matrix(c(election$poll.diff - qnorm(p = 0.975) * se, election$poll.diff +
    qnorm(p = 0.975) * se), nrow = 51, ncol = 2)
significant = as.logical(m[, 1] > 0 | m[, 2] < 0)
significant
```

```
# Mutates `election` to add the new data
election = mutate(election, std.unit = std.unit, same.sign = same.sign,
    significant = significant)
str(election)
'data.frame':   51 obs. of  12 variables:
 $ state          : Factor w/ 51 levels "Alabama","Alaska",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ samplesize     : int  2391 6793 14515 1911 2958 6591 27295 3088 17664 4021 ...
 $ rawpoll_clinton: num  0.304 0.4 0.525 0.499 0.328 ...
 $ rawpoll_trump  : num  0.564 0.451 0.281 0.339 0.391 ...
 $ clinton.vote   : num  0.344 0.366 0.451 0.337 0.617 ...
 $ trump.vote     : num  0.621 0.513 0.487 0.606 0.316 ...
 $ poll.diff      : num  0.2598 0.0516 -0.2442 -0.1602 0.063 ...
 $ vote.diff      : num  0.2773 0.1473 0.0355 0.2692 -0.3011 ...
 $ deviation      : num  -0.0174 -0.0957 -0.2797 -0.4294 0.3641 ...
 $ std.unit       : num  -0.953 -8.56 -38.991 -20.824 23.424 ...
 $ same.sign      : logi  TRUE TRUE FALSE FALSE FALSE FALSE ...
 $ significant    : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
```

### (8 pts) Part b

Use the information that you calculated in 3.a to answer the following questions:

i. How many of the 95% CIs for estimating (proportion of Trump supporters - proportion of Clinton supporters) mentioned in Part (a) are away from 0?

```
# Outputs the relevant number
sum(election$significant == TRUE)
[1] 48
```

Answer: Using the information that we calculated in 3.a, 48 out of the 51 95% CIs for estimating (proportion of Trump supporters - proportion of Clinton supporters) are away from 0.

ii. Among the CIs that are away from 0 how many of them predict the winning candidate of the state correctly?

```
# Outputs the relevant number
sum(election$significant == TRUE & election$same.sign == TRUE)
[1] 22
```

Answer: 22 out of the 48 95% CIs that are away from 0 predicts the winning candidate of the sate correctly.

iii. How many of the 95% CIs cover 0?

```
# Outputs the relevant number
sum(election$significant == FALSE)
[1] 3
```

Answer: Using the information that we calculated in 3.a, 3 out of the 51 95% CIs for estimating (proportion of Trump supporters - proportion of Clinton supporters) cover 0.

iv. Among the states whose CIs cover 0 how many of them predict the winning candidate of the state correctly with their `poll.diff` value?

```
# Outputs the relevant number
sum(election$significant == FALSE & election$same.sign == TRUE)
[1] 1
```

Answer: Out of the states whose CIs include 0, 1 out of the 3 states predicts the winning candidate of the state correctly with their `poll.diff` value.

### (12 pts) Part c

The error for a prediction is defined as

Prediction error := Predicted value - Actual value

The errors in the polls are actually not much bigger than that in previous elections on average (See NY Times article here). However, the problem is that the polls that over estimated Clinton's votes are polls for the battleground states.

**(6 pts) i.** In Part b.ii you reported the number of CIs that correctly predict the winning candidate of the state among the CIs that are away from 0. Among the CIs that are away from 0, how many incorrectly predicts the winner of the state? List the names of the states that associate with these CIs; for these states did the polls predict that Clinton was going to win or lose the election for the state?

Answer: Among the CIs that are away from 0 that incorrectly predicted the winner of the state, the corresponding states are listed below:

```
# Outputs the appropriate information
election[election$significant == TRUE & election$same.sign == FALSE,
    ]$state
 [1] Arizona              Arkansas
 [3] California           Colorado
 [5] Delaware             District of Columbia
 [7] Illinois             Indiana
 [9] Iowa                 Kentucky
[11] Maine                Michigan
[13] Montana              Nebraska
[15] New York             North Carolina
[17] Oklahoma             South Carolina
[19] South Dakota         Tennessee
[21] Texas                Utah
[23] Vermont              Virginia
[25] Washington           Wisconsin
51 Levels: Alabama Alaska Arizona Arkansas ... Wyoming
```

Among these incorrect significant CIs, these are the states that predicted Clinton was going to lose the election for that state:

```
# Outputs the appropriate information
election[election$significant == TRUE & election$same.sign == FALSE &
    election$poll.diff > 0, ]$state
 [1] California           Colorado
 [3] Delaware             District of Columbia
 [5] Illinois             Maine
 [7] New York             Vermont
 [9] Virginia             Washington
51 Levels: Alabama Alaska Arizona Arkansas ... Wyoming
```

Among these incorrect significant CIs, these are the states that predicted Clinton was going to win the election for that state:

```
# Outputs the appropriate information
election[election$significant == TRUE & election$same.sign == FALSE &
    election$poll.diff < 0, ]$state
 [1] Arizona        Arkansas       Indiana
 [4] Iowa           Kentucky       Michigan
 [7] Montana        Nebraska       North Carolina
[10] Oklahoma       South Carolina South Dakota
[13] Tennessee      Texas          Utah
[16] Wisconsin
51 Levels: Alabama Alaska Arizona Arkansas ... Wyoming
```

**(3 pts) ii.** According to Wikipedia (click here)

> Election analytics website FiveThirtyEight identifies the states of Colorado, Florida, Iowa, Michigan, Minnesota, Ohio, Nevada, New Hampshire, North Carolina, Pennsylvania, Virginia, and Wisconsin as 'perennial' swing states that have regularly seen close contests over the last few presidential campaigns.

Which of the states that you identified in part i are among the swing states?

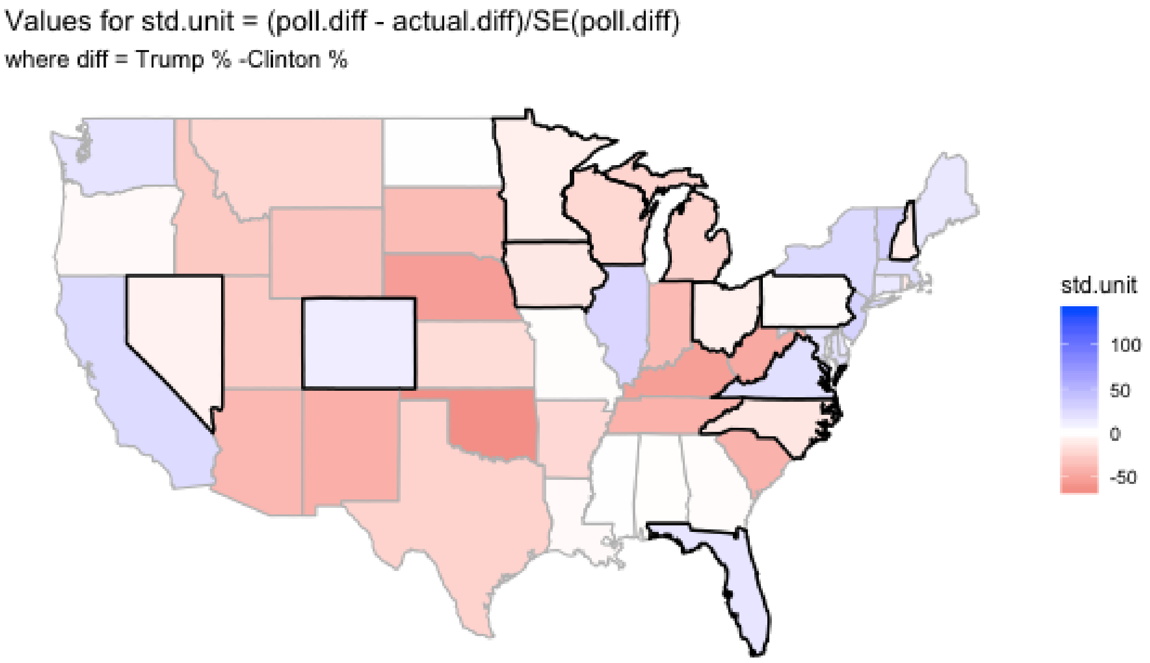
Answer: The states that we identified in part i that are also swing states are Colorado, Iowa, Michigan, North Carolina, Virginia, & Wisconsin.

**(3 pts) iii**. We are going to plot the values of `std.unit` on the map to investigate the geographic pattern of the `std.unit` values. Please make sure that you have installed two packages: `maps` and `dplyr`.

Run the code in the following code chunk will create the function `create.map()`. `create.map()` will make a map for the states defined by the variable `input.state`, color the states on the map according to the values in `map.var`, and then outlines the states specified by the variable `outlined.state`. (You do not need to understand line by line how the function is being constructed; you just need to know how to use the function.)

Here is an example of how to use the function (you will need to remove `eval = F` if you would like to show this in your report). The following code draws the states Michigan and North Carolina, colors Michigan with the color that correspond to the value 1 and colors North Carolina with the color that correspond to the value 2. The code also outlines Michigan.

```
# Creates the map with the appropriate data Note: We edited the
# map creation code to have states predicted to be Democrat to
# be blue, rather than green.
create.map(input.state = election$state, map.var = election$std.unit,
    outlined.state = c("Colorado", "Florida", "Iowa", "Michigan",
        "Minnesota", "Ohio", "Nevada", "New Hampshire", "North Carolina",
        "Pennsylvania", "Virginia", "Wisconsin"))
```


Values for std.unit = (poll.diff - actual.diff)/SE(poll.diff)
where diff = Trump % -Clinton %
std.unit
100
50
0
-50

Use `create.map()` to plot the values of `std.unit` for each state and then outline the swing states on the map. (Note that Alaska, Washington D.C. and Hawaii will not show up on the map because the package used to draw the map excludes these places.)

Hint: `input.state` should be a factor or a character vector of the state names, `map.var` should be a numeric vector and `outlined.state` should be a character vector of the names of the states you would like to outline. Please make sure the `input.state` and `map.var` have the same length and the elements in both vectors are arranged in the same order.

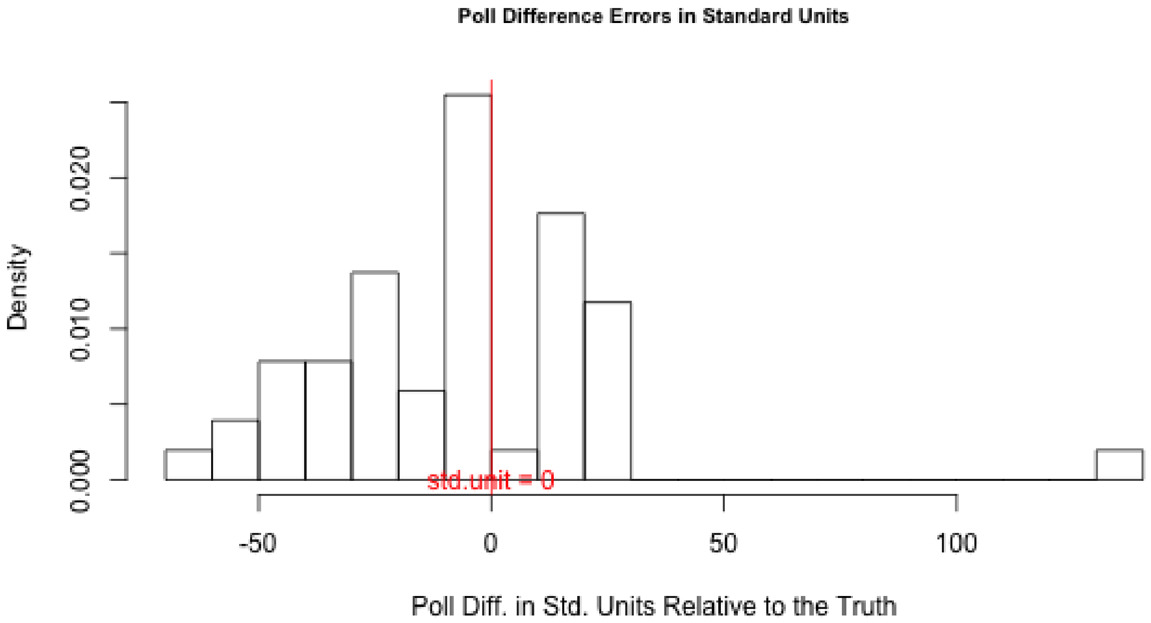
Based on the map do the polls under- or over-estimated Clinton's chance of winning for most of the swing states?

Answer: Based on the map, the polls over-estimated Clinton's chance of winning for most pf the swing states.

### (5 pts) Part d

Fill in the value for the x argument below to make a histogram for the values of `std.unit` and mark the location where `std.unit = 0` with a red vertical line. Do the errors in the estimates look like they have mean zero?

```r
# Generate histogram
hist(x = election$std.unit, breaks = 25, freq = F, main = "Poll Difference Errors in Standard Units",
    xlab = "Poll Diff. in Std. Units Relative to the Truth", cex.main = 0.8)
abline(v = 0, col = "red")
text(x = 0, y = 0, label = "std.unit = 0", col = "red")
```



Answer: The errors in the estimates look like they have mean zero, since the distribution of the errors in the estimates appears to be a normal distribution.

### (8 pts) Part e

How many states have polls underestimated (i.e., the value in `std.unit` is less than zero) how well Trump would do? Let's call this number $m$.

```
# Creates `m`
m = sum(election$std.unit < 0)
m
[1] 34
```

Answer: The polls underestimated how well Trump would do in $m = 34$ states.

If the errors in the poll data were random, there should be equally likely chance for a state to underestimate or overestimate Trump. Then, the values in `std.unit` should be like 51 values drawn from a common distribution independently.

What is this common distribution? Report the numerical value(s) of the parameters of this distribution. According to this common distribution what is the chance $g$ that the `std.unit` value for a particular state would be positive?

This common distribution is a standard normal distribution with parameters (0,1). According to this common distribution, the chance $g$ that the `std.unit` value for a particular state would be positive is 0.50.

### (10 pts) Part f

If the errors in the poll data were random and independent among the states, each state should have $g$ probability to over estimate how well Trump would do. Calculate the chance that we would have $m$ or more states that over estimate how well Trump would do in this case.

Hint: If the errors in the poll data were random and independent among the states, each state should have $g$ probability of overestimating how well Trump would do. You can think of the model as having a large population with $g$ fraction of 1's and $1 - g$ fraction of 0's. Here the question is asking that if you randomly select 51 numbers with replacement from this population, what is the chance that you would observe $m$ or more 1's?

```
# Creates `g` and a sample for `m`
g = 0.5
m = rep(c(1, 0), c(34, 17))
# Performs the appropriate t-test
t.test(m, mu = g)

    One Sample t-test

data:  m
t = 2.5, df = 50, p-value = 0.01574
alternative hypothesis: true mean is not equal to 0.5
95 percent confidence interval:
 0.5327627 0.8005706
sample estimates:
mean of x
0.6666667
```

Answer: The chance that we would observer $m$ or more 1's is 0.01574.

Based on the data is it reasonable to assume that the errors in the poll data were random and independent among the states? Explain why.

Answer: Based on the data, it is not reasonable to assume that the errors in the poll data were random and independent among the states. Assuming an $\alpha$ value of 0.05 and our t-test returned a p-value of 0.01574, the chance that we would observe $m$ or more 1's is statistically significant. This means that the sample mean and the population mean are probably not the same, and thus not independent.