Data 8 Spring 2016 lab07, Due: *Not collected*

Problem 1 Votes

In this problem, we are a polling company trying to predict the vote in the 2016 Republican primary election in California. (In a party's primary election, people vote to determine the party's candidate for the actual election, which happens later.) Millions of people will vote in the election. We select a simple random sample of 200 people likely to vote in the Republican primary, and we ask who they will vote for.

We're interested in predicting the proportion of people who will vote for Donald Trump. Let's say that that proportion is *actually* .2. Our polling company doesn't know that, though.

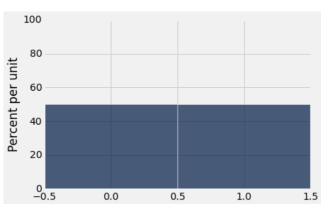
Suppose there is a table of all the electoral votes called `all_votes` and a table of our sample of 200 called `sample_votes`:

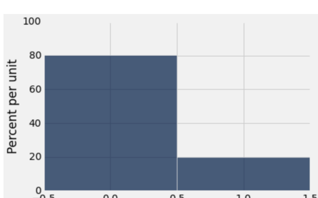
```
N = 200

all_votes = Table.read_table("all_votes.csv")
sample_votes = all_votes.sample(N)
sample_votes.show(10)
```

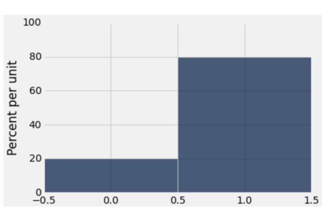
Name
Ted Cruz
Ted Cruz
Donald Trump
John Kasich
Marco Rubio
Ted Cruz
Marco Rubio
John Kasich
Ben Carson
Donald Trump

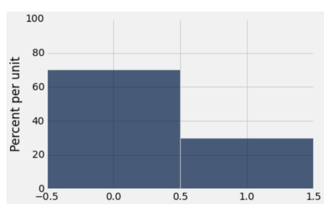
... (190 rows omitted)

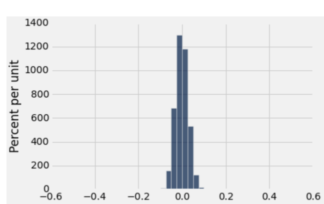
For each of the scenarios below, we've described a histogram made from this data, or written some code that manipulates the data and makes a histogram. For each one, find in the list below *all* the histograms that would not be surprising results for the computations described.

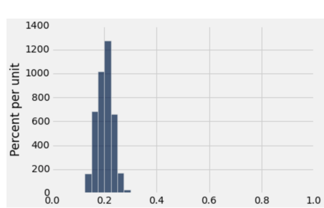
(i)


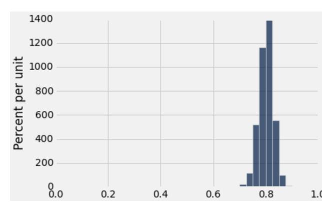
(ii)


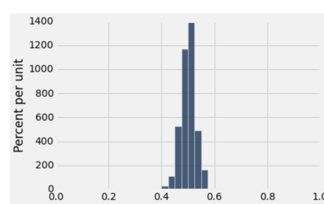
(iii)


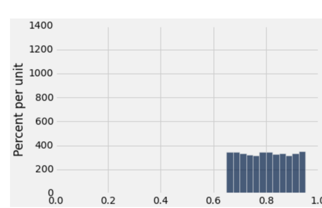
(iv)


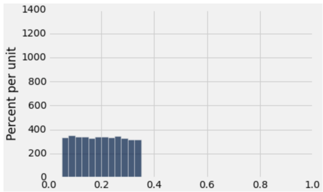
(v)


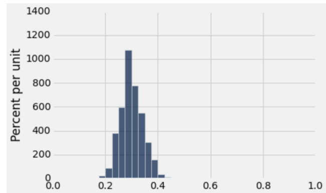
(vi)


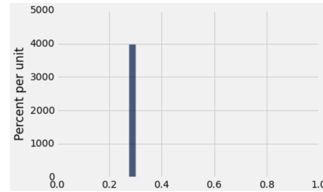
(vii)


(viii)


(ix)


(x) 

(xi) 

(xii) 

Hint: Pay attention to the *mean* and *shape* of each histogram. The *spread* of each histogram might be hard to compute, but you can compare the spreads of similarly-shaped histograms to each other.

(a) Suppose we mark each vote as a 1 if it's a vote for Trump and a 0 otherwise, and then make a histogram of those numbers from `all_votes`, using this code:

```
def one_if_trump(candidate):
    if candidate == "Donald Trump":
        return 1
    else:
        return 0

votes_with_trump_indicator = all_votes.with_column(
        "Trump",
        all_votes.apply(one_if_trump, "Name"))
votes_with_trump_indicator.hist("Trump", bins=[-0.5, 0.5, 1.5])
```

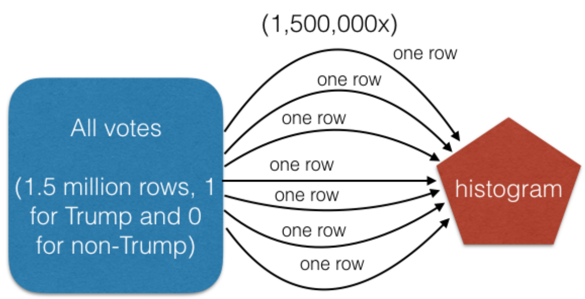


Figure 1: A visual depiction of (a).

(b) We do the same thing, but instead of `all_votes`, we use the smaller sample our polling company actually sees.

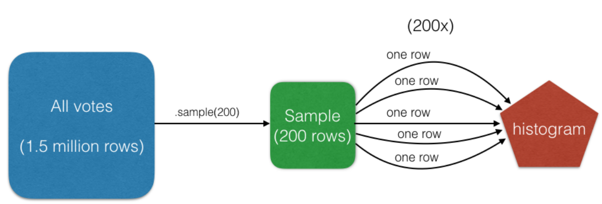


Figure 2: A visual depiction of (b).

(c) Now we'd like to understand how our sample-based estimate of the proportion of Trump votes would "typically" (that is, across many re-runs of the sampling process) behave. We produce an empirical histogram of the proportions of Trump votes in 1000 different random samples of size 200. (If you're not sure how to write the code to do that, it would be a very useful exercise to try.)

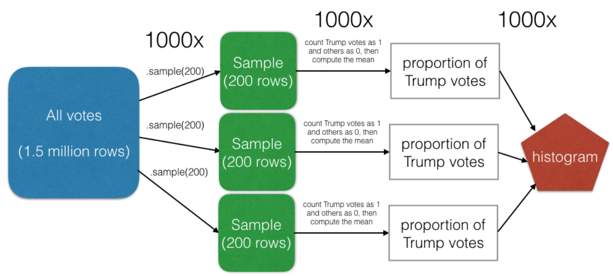


Figure 3: A visual depiction of (c).

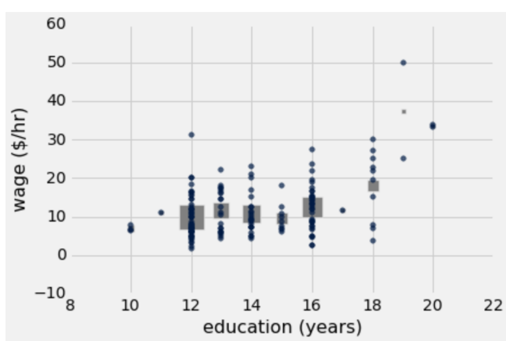
(d) Now we'd like to understand how the sample-based estimate of the proportion of *non-Trump* votes would typically behave. We produce an empirical histogram of the proportions of *non-Trump* votes in 1000 different random samples of size 200.

(e) A political scientist on our staff believes that polls underestimate Trump's support because some Trump voters are reluctant to tell pollsters they're voting for Trump. (This is a real idea.) To correct for this, we estimate the *true proportion of Trump voters in California* by multiplying the proportion of Trump voters in our poll by 1.5. We produce an empirical histogram of these new estimated proportions of Trump votes (that is, the sample proportion of Trump votes times 1.5) from 1000 different random samples of size 200.

Problem 2 Education

In a 1994 study, economists Ashenfelter and Krueger analyzed the *financial returns to education* – that is, how much getting more school increases income. They were interested in comparing twins, so they collected their data at the 16th Annual Twins Days Festival in Twinsburg, Ohio. We'll do a simplified version of their analysis.

Here's a graph of education (measured in years spent in school) and hourly wages for all the people in the study. The light squares denote the *mean* of each vertical slice. (The area of each square is proportional to the number of people in that slice.)



(a) We'd like to model the relationship between education and hourly wages with a smooth curve. Suppose we don't restrict ourselves to fitting lines, and we allow any curve. Draw a curve that fits the data as well as possible. (If you'd like the goal to be stated more precisely: Draw a curve so that the squared vertical distance between the curve and each point is minimized.)

(b) True or false: Your curve passes through the average of each vertical strip.

(c) Now draw the *line* that fits the data as well as possible.

(d) True or false: Your line passes through the average of each vertical strip.

(e) Is the correlation between education and wages positive, negative, or zero? What's your best guess at the correlation?

(f) Would you say there's a roughly linear relationship between education and wages in these data? If not, how would you characterize the relationship?

(g) *[Optional]* It's important to note that these data only show an *association* between education and wages. Showing causation is much harder. For example, maybe we see this relationship because people with the perseverence to stay in school apply the same perseverence to get higher-paying jobs. That's an example of a *confounding factor*. Describe how knowing about the *association by itself* could be useful.

(h) *[Optional]* Describe how knowing whether more education *causes* higher wages could be useful.

(i) *[Optional]* Can you guess how Ashenfelter and Krueger used the fact that their dataset actually contained many pairs of *genetically-identical twins* to eliminate confounding factors like that?