Introduction to simulation, Part 2

Put your name here

Put the date here

Introduction

In this module, we'll learn more about simulation and randomization. Instead of flipping coins, in this assignment we'll randomly shuffle our data around in order to explore the effects of randomizing an explanatory variable.

Instructions

Presumably, you have already created a new project and downloaded this file into it. From the `Run` menu above, select `Run All` to run all existing code chunks.

When prompted to complete an exercise or demonstrate skills, you will see the following lines in the document:

ANSWER

These lines demarcate the region of the R Markdown document in which you are to show your work.

Sometimes you will be asked to add your own R code. That will appear in this document as a code chunk with a request for you to add your own code, like so:

```
# Add code here
```

Be sure to remove the line `# Add code here` when you have added your own code. You should run each new code chunk you create by clicking on the dark green arrow in the upper-right corner of the code chunk.

Sometimes you will be asked to type up your thoughts. That will appear in the document with the words, "Please write up your answer here." Be sure to remove the line "Please write up your answer here" when you have written up your answer. In these areas of the assignment, please use contextually meaningful full sentences/paragraphs (unless otherwise indicated) and proper spelling, grammar, punctuation, etc. This is not R code, but rather a free response section where you talk about your analysis and conclusions. You may need to use inline R code in these sections.

When you are finished with the assignment, knit to PDF and proofread the PDF file **carefully**. Do not download the PDF file from the PDF viewer; rather, you should export the PDF file to your computer by selecting the check box next to the PDF file in the Files pane, clicking the `More` menu, and then clicking `Export`. Submit your assignment according to your professor's instructions.

Load Packages

We load the `MASS` package to access the `birthwt` data on risk factors associated with low birth weight. We also load the `mosaic` package for simulation tools.

```
library(MASS)
library(mosaic)
```

As explained in an earlier module, we will set the seed so that our results are reproducible.

```
set.seed(3141593)
```

Our research question

We are interested in finding out if there is an association between low birth weight and smoking during pregnancy. We'll use the birth weight data `birthwt` from the `MASS` package.

Exploratory data analysis

Rather than using the actual birth weight of the baby, let's use the categorical variable `low` that is simply an indicator (yes/no) of whether the birth weight is less than 2.5 kg.

We can see below that neither `low` nor `smoke` are factor variables as we need them to be:

```
str(birthwt)
```

```
## 'data.frame':    189 obs. of  10 variables:
##  $ low  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ age  : int  19 33 20 21 18 21 22 17 29 26 ...
##  $ lwt  : int  182 155 105 108 107 124 118 103 123 113 ...
##  $ race : int  2 3 1 1 1 3 1 3 1 1 ...
##  $ smoke: int  0 0 1 1 1 0 0 0 1 1 ...
##  $ ptl  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ht   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ui   : int  1 0 0 1 1 0 0 0 0 0 ...
##  $ ftv  : int  0 3 1 2 0 0 1 1 1 0 ...
##  $ bwt  : int  2523 2551 2557 2594 2600 2622 2637 2637 2663 2665 ...
```

We'll fix that by creating new factor variables called `low` and `smoke` and putting them in their own data frame.

In a categorical variable, the order of the categories usually does not matter. However, in statistics, we often designate one category as the "success" category, or the category of interest to us. All other categories are considered "failures". For example, it is the low birth weight babies who are of interest to us; therefore a "Yes" in this variable will be considered a "success". The terminology is unfortunate, but we're stuck with it.

In the `factor` command in R, be sure to list the success category first in both the `levels` and `labels` parts of the command. We are interested in whether mothers have babies with low birth weight, so (unfortunately) having low birth weight is considered the "success" condition. The order of the categories of `smoke` is less important because that is the explanatory variable. Observe the ordering of the levels and labels in the following:

```
low <- factor(birthwt$low, levels = c(1, 0), labels = c("Yes", "No"))
smoke <- factor(birthwt$smoke, levels = c(1, 0), labels = c("Yes", "No"))
low_smoke <- data.frame(low, smoke)
```

Use `View` at the Console to look at a spreadsheet view of `low_smoke` and make sure that everything looks okay. We'll also use `str` and `head`.

```
str(low_smoke)
```

```
## 'data.frame':    189 obs. of  2 variables:
##  $ low  : Factor w/ 2 levels "Yes","No": 2 2 2 2 2 2 2 2 2 2 ...
##  $ smoke: Factor w/ 2 levels "Yes","No": 2 2 1 1 1 2 2 2 1 1 ...
```

```
head(low_smoke)
```

```
##   low smoke
## 1  No    No
## 2  No    No
## 3  No   Yes
## 4  No   Yes
## 5  No   Yes
## 6  No    No
```

Exercise

Create two contingency tables with `low` as the row variable and `smoke` as the column variable, one with counts and one with percentages displayed. Be sure to include the marginal distribution (the column sums) as well. Go back to the R module `Tables.Rmd` if you need to review the `tally` command.

ANSWER

```
# Add code here to create a contingency tables of counts with
# low as the row variable and smoke as the column variable.
```

```
# Add code here to create a contingency tables of percents with
# low as the row variable and smoke as the column variable.
```

By placing `low` as the row variable and `smoke` as the column variable, and then looking at percentages, we are implying that one variable is response and one is explanatory. Which variable are we treating as response and which are we treating as explanatory? Do you agree with that choice? Why or why not?

ANSWER

Please write up your answer here.

Although we can read off the percentages in the contingency table, we need a command to extract the proportions for use in further calculations. One way is via the `prop` command. This command uses the "tilde" notation. The variable on the left of the tilde (`low`) is the variable of interest (the response variable), and we're dividing up the data into groups based on the variable on the right of the tilde (`smoke`). You can remember this by thinking, "Calculate low birth weight **by** smoking status."

```
prop(low ~ smoke, data = low_smoke)
```

```
##   Yes.Yes    Yes.No
## 0.4054054 0.2521739
```

The column headers in the resulting output are a bit opaque. What we have here are the two percentages of interest that we're trying to compare.

Exercise

Interpret these percentages in the context of the data. In other words, what do these percentages say about the women who smoke during pregnancy versus the women who do not? (Hint: look back at the contingency table you made earlier. Where do these percentages come from?)

ANSWER

Please write up your answer here.

The real statistic of interest to us, though, is the difference between these percentages. We can use the `diffprop` function from the `mosaic` package.

```
diffprop(low ~ smoke, data = low_smoke)
```

```
##   diffprop
## -0.1532315
```

Let's store this value for future use. We can use any name we want, but I've chosen `obs_diff` here for "observed difference".

```
obs_diff <- diffprop(low ~ smoke, data = low_smoke)
obs_diff
```

```
##   diffprop
## -0.1532315
```

Exercise

In which order are the groups being subtracted? In other words, why is the observed difference negative?

ANSWER

Please write up your answer here.

Note that if we had used

```
smoke <- factor(birthwt$smoke, levels = c(0, 1), labels = c("No", "Yes"))
```

instead of

```
smoke <- factor(birthwt$smoke, levels = c(1, 0), labels = c("Yes", "No"))
```

the order of the groups would be the opposite, and the results of `diffprop` would have been positive instead of negative. It makes very little difference conceptually (there are two groups and they have to be subtracted in *some* order), but there are several places in the R code in which you have to be very aware of that choice. Just remember that `diff`, `diffprop`, and other related differencing operations always subtract the first entry *from* the second entry.

Shuffling

One way to see if there is evidence of an association between low birth weight and smoking is to assume, temporarily, that there is no association. If there were truly no association, then the difference between the smoking group and the nonsmoking group should be 0%.

Now, we saw a difference of -15.3% between the two groups in the data. Then again, non-zero differences can just come about by pure chance alone. We may have accidentally sampled more smokers who also just happened to have babies with low birth weight, even though there may be no association in the general population.

So how do we test the range of values that could arise from just chance alone? In other words, how do we explore sampling variability?

One way to force the variables to be independent is to "shuffle" the values of `smoke`. If instead of measuring whether women actually smoke or not, we just randomly and arbitrarily label them as "smokers" or "nonsmokers" (independent of their *actual* smoking status), we know for sure that such an assignment is random and not due to any actual evidence of smoking. In that case, low birth weight babies are equally likely to occur in both groups.

Let's see how shuffling works in R. To begin with, look at the first 20 actual values of `smoke` in our data:

```
head(low_smoke$smoke, n = 20)
```

```
##  [1] No  No  Yes Yes Yes No  No  No  Yes Yes No  No  No  No  Yes Yes No 
## [18] Yes No  Yes
## Levels: Yes No
```

Now we "shuffle" all the values around and look at the first 20 again:

```
head(shuffle(low_smoke$smoke), n = 20)
```

```
##  [1] Yes No  No  Yes Yes No  No  No  Yes Yes Yes No  No  Yes No  Yes No 
## [18] Yes Yes Yes
## Levels: Yes No
```

Do it again, just to make sure it's random:

```
head(shuffle(low_smoke$smoke), n = 20)
```

```
##  [1] No  No  Yes No  No  No  Yes Yes No  Yes No  Yes Yes Yes No  Yes No 
## [18] No  Yes No
## Levels: Yes No
```

Simulation

The idea here is to keep the low birth weight status the same for each woman, but randomly shuffle the smoking labels. There will still be the same number of women who "smoke", but now they will be randomly assigned such a designation. Since this new grouping into "smoking" and "nonsmoking" is completely random and arbitrary, we expect the likelihood of having a low birth weight baby to be equal for both groups.

A more precise way of saying this is that the expected difference under the assumption of independent variables is 0%. If there were truly no association, then the percentage of women having low birth weight babies would be independent of smoking. However, sampling variability means that we are not likely to see an exact difference of 0%. In fact, due to the sample sizes in each group, it is impossible to get a difference of exactly 0%. The real question, then, is how different could the difference be from 0% and still be reasonably possible due to random chance.

Here are a few random simulations. (The randomness is built into the `shuffle` command.)

```
diffprop(low ~ shuffle(smoke), data = low_smoke)
```

```
##   diffprop
## 0.09106933
```

```
diffprop(low ~ shuffle(smoke), data = low_smoke)
```

```
##    diffprop
## -0.04218566
```

```
diffprop(low ~ shuffle(smoke), data = low_smoke)
```

```
##   diffprop
## 0.02444183
```

The `do` command from the `mosaic` package allows us to repeat this simulation process any number of times. From an earlier module, we learned that the number of simulations needs to be sufficiently high to make sure we have a good representative set of outcomes. Here, we'll use 5000 simulations. The results will be gathered together in a data frame that we will call `sims`.

```
sims <- do(5000) * diffprop(low ~ shuffle(smoke), data = low_smoke)
head(sims, n = 20)
```

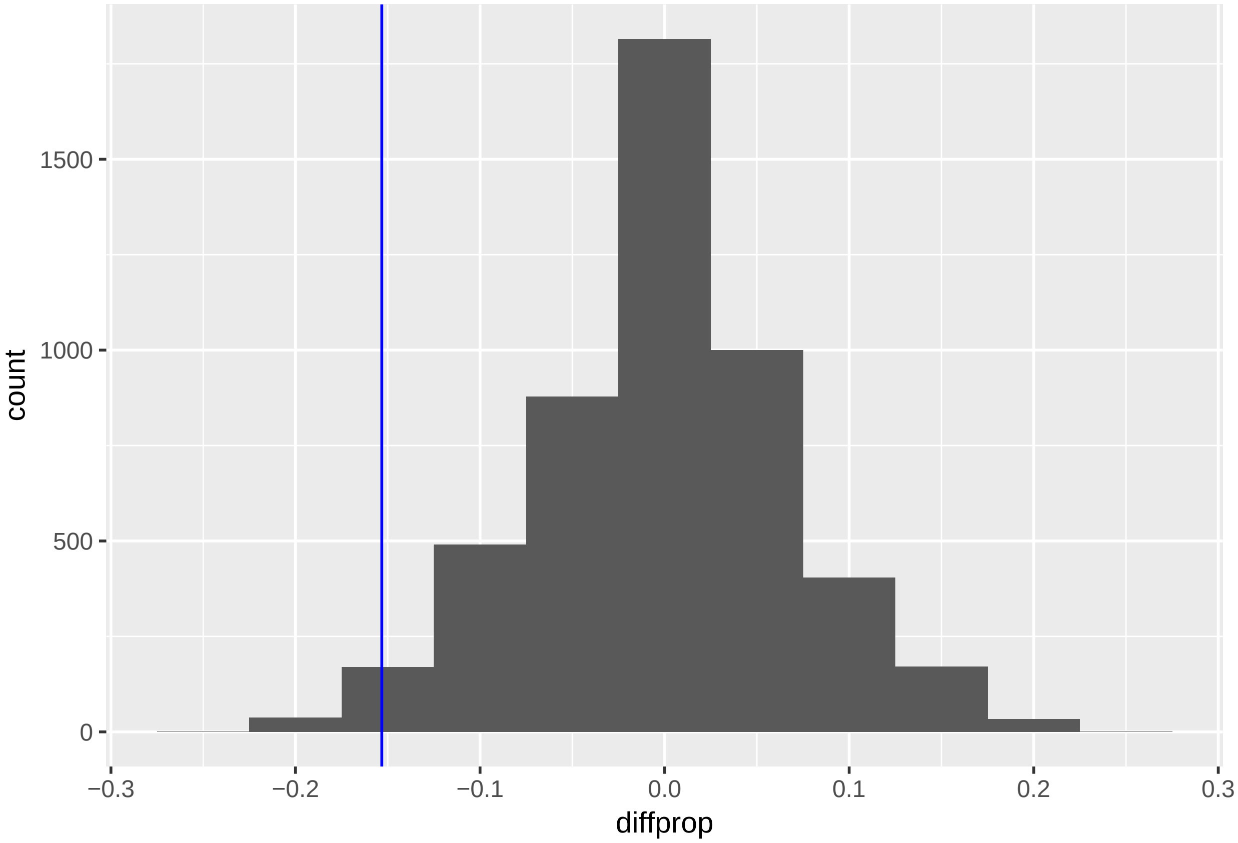
```
##        diffprop
## 1  -0.153231492
## 2   0.024441833
## 3   0.068860165
## 4   0.002232667
## 5  -0.019976498
## 6  -0.019976498
## 7   0.046650999
## 8  -0.042185664
## 9   0.046650999
## 10 -0.108813161
## 11  0.046650999
## 12  0.046650999
## 13 -0.019976498
## 14 -0.108813161
## 15 -0.086603995
## 16  0.068860165
## 17  0.002232667
## 18 -0.019976498
## 19 -0.108813161
## 20  0.002232667
```

Think carefully about what this pile of numbers means. Each row of the `sims` data frame represents a simulated difference in the proportion of low birth weight babies between groups of women who smoke or who don't smoke. The `shuffle` part of the simulation ensures that there is no actual relationship between smoking and low birth weight among these simulated values. We expect each simulated difference to be close to zero, but we also expect deviations from zero due to randomness and chance.

Plot results

A histogram will show us the range of possible values under the assumption of independence of the two variables. On the same plot, we graph a line at the value of the actual observed difference in proportions to see if that true value from our data could have reasonably occurred by chance alone.

```r
ggplot(sims, aes(x = diffprop)) +
    geom_histogram(binwidth = 0.05) +
    geom_vline(xintercept = obs_diff, color = "blue")
```



Exercise

Why is the mode of the graph above at 0? This has been explained several different times in this module, but put it into your own words to make sure you understand the logic behind the shuffling.

ANSWER

Please write up your answer here.

By chance?

How likely is it that the observed difference (or a difference even more extreme) could have resulted from chance alone? Because `sims` contains simulated results after shuffling, the variable `diffprop` contains values that assume that smoking is independent of birth weight. In order to assess how plausible our observed difference is under that assumption, we want to find out how many of the simulated values are at least as small, if not smaller, than the observed difference, -0.1532315.

Look at the 100 smallest entries of the values of `diffprop`:

```
head(sort(sims$diffprop), n = 100)
```

```
##   [1] -0.2420682 -0.2198590 -0.2198590 -0.2198590 -0.2198590 -0.2198590
##   [7] -0.2198590 -0.1976498 -0.1976498 -0.1976498 -0.1976498 -0.1976498
##  [13] -0.1754407 -0.1754407 -0.1754407 -0.1754407 -0.1754407 -0.1754407
##  [19] -0.1754407 -0.1754407 -0.1754407 -0.1754407 -0.1754407 -0.1754407
##  [25] -0.1754407 -0.1754407 -0.1754407 -0.1754407 -0.1754407 -0.1754407
##  [31] -0.1754407 -0.1754407 -0.1754407 -0.1754407 -0.1754407 -0.1754407
##  [37] -0.1754407 -0.1754407 -0.1532315 -0.1532315 -0.1532315 -0.1532315
##  [43] -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315
##  [49] -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315
##  [55] -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315
##  [61] -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315
##  [67] -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315
##  [73] -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315
##  [79] -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315
##  [85] -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315
##  [91] -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315 -0.1532315
##  [97] -0.1532315 -0.1532315 -0.1310223 -0.1310223
```

Of the 5000 simulations, the most extreme difference of about 24% occurred once, just by chance. That seems like a pretty extreme value when expecting a value of 0%, but the laws of probability tell us that extreme values will be observed from time to time, even if rarely. Also recall that the observed difference in the actual data was -0.1532315. This specific value came up quite a bit in our simulated data. In fact, the 98th entry of the sorted data above is the last occurrence of the value -0.1532315. After that, the next higher value is -0.1310223.

So let's return to the original question. How many simulated values are as small—if not smaller—than the observed difference? Apparently, 98 out of 5000, which is 0.0196. In other words 1.96% of the simulated data is as extreme or more extreme than the actual difference in low birth weight prevalence between smoking and nonsmoking women in the real data. That's not very large. In other words, a difference like -0.1532315 could occur just by chance—like flipping 10 out of 10 heads or something like that. But it doesn't happen very often.

We can automate this using the same `prop` command we saw before, but used in a slightly different way. In the command below, we use the condition

`sims$diffprop <= obs_diff`.

In other words this is asking, "When is the value of `diffprop` less than or equal to the difference that was actually observed in our data?"

```
prop(sims$diffprop <= obs_diff)
```

```
##   TRUE
## 0.0196
```

COPY/PASTE WARNING: If the observed difference were positive, then extreme values of interest would be *greater* than 0.1532315, not less than -0.1532315. You must note if the observed difference is positive or negative and then use `<=` or `>=` as appropriate!

Again, 0.0196 is a small number. This shows us that if there were truly no association between low birth weight and smoking, then our data is a rare event. (An observed difference this extreme or more extreme would only occur about 2% of the time by chance.)

Because the probability above is so small, it seems unlikely that our variables are independent. Therefore, it seems more likely that there is an association between low birth weight and smoking. We have evidence of a statistically significant difference between the chance of having a baby with low birth weight among women who smoke versus women who don't smoke.

Keep in mind that this data is from an observational study, so we cannot conclude that smoking *causes* low birth weight. All we can say is that a difference of -15.3% is evidence that mothers who smoke are more likely to have babies with low birth weight for *some* reason (maybe because they smoke, but maybe for some other reason).

Exercise

When we see an association between two variables, it may be that both factors are related to some third variable that we haven't measured. When that happens, that third variable is called a *lurking variable*. Get creative for a minute and see if you can come up with a reasonably plausible lurking variable for the scenario above. In other words, if smoking itself doesn't cause low birth weight babies, can you imagine another factor that would cause low birth weight babies and also be associated with mothers smoking?

ANSWER

Please write up your answer here.

The point of the above exercise is not to convince people that smoking during pregnancy is not actually the cause of low birth weight babies. In fact, there is quite a bit of evidence that smoking during pregnancy causes all sorts of problems, including low birth weight.[1] However, **association is not causation**. From observational data like this, it is generally impossible to prove a causal relationship. I'm not saying there is never a causal relationship, only that you can't *prove* it from the analysis we did. Nevertheless, there is a clear association between low birth weight and smoking during pregnancy. Make sure you understand the difference.

Your turn

Walk through the following sequence of steps to explore whether low birth weight is associated with the presence of uterine irritability. You should carefully copy and paste commands from earlier in the module, making the necessary changes to the variable names.

Uterine irritability is recorded in the variable `ui` in the `birthwt` data set.

1. Convert `ui` to a factor variable. Then create a new data frame from the factor variables `low` and `ui`. (As `low` is already a factor variable, you don't need to re-do the work for that one.) Call this data frame `low_ui`.

ANSWER

```
# Add code here to convert ui to a factor variable.
# Combine low and ui into a single data frame called low_ui.
```

[1] See this for example: http://www.ncbi.nlm.nih.gov/pubmed/16323070

2. Exploratory data analysis: make two contingency tables with `low` as the response variable and `ui` as the explanatory variable. One table should have counts and the other table should have percentages. (Both tables should include the marginal distribution.)

ANSWER

```
# Add code here to make a contingency table with counts.
```

```
# Add code here to make a contingency table with percentages.
```

3. Use the `diffprop` function to calculate and store the observed difference in the proportion of low birth weight babies between women with and without uterine irritability. Call this `obs_diff2` so that it doesn't conflict with the earlier `obs_diff`.

ANSWER

```
# Add code here to calculate the observed difference.
# Store this as obs_diff2.
```

4. Simulate 2000 outcomes under the assumption that low birth weight is independent of uterine irritability. Use the `do` command in conjunction with `diffprop` and `shuffle` in a single line of code. Call the simulated data frame `sims2` so that it doesn't conflict with the earlier `sims`.

ANSWER

```
# Add code here to simulate 2000 outcomes under the independence assumption
# and store the simulations in a data frame called sims2.
```

5. Plot the simulated values in a histogram. Be sure to include a vertical line at the value of the observed difference.

ANSWER

```
# Add code here to plot the results.
```

6. Calculate how likely it is to see our observed difference or something even more extreme among the randomly simulated values. Pay close attention to whether the observed difference is positive or negative and choose `>=` or `<=` accordingly.

ANSWER

```
# Add code here to calculate how likely it is to see
# our observed difference or something even more extreme
# among the randomly simulated values.
```

Finally, comment on what you see. Based on the number you get in step 6 above, is the observed difference rare? In other words, under the assumption that low birth weight and uterine irritability are independent, are we likely to see an observed difference as far away from zero as we actually see in the data? So what is your conclusion then? Do you believe there is an association between low birth weight and uterine irritability?

ANSWER

Please write up your answer here.

Conclusion

Here we used simulation to explore the idea of two variables being independent or associated. When we assume they are independent, we can explore the sampling variability of the differences that could occur by pure chance alone. We expect the difference to be zero, but we know that randomness will cause the simulated differences to have a range of values. Is the difference in the observed data far away from zero? In that case, we can say we have evidence that the variables are not independent; in other words, it is more likely that our variables are associated.