Introduction to probability

Clayton Baker

2-22-21

Upcoming Deadline

- Exam due by 11:59 PM on February 22nd.
- Lab in groups on Thursday.

Main ideas

- Use formulas to compute probabilities from tabular data
- Compute empirical probabilities in R via simulation

Packages

```
library(tidyverse)
library(usethis)
use_git_config(user.name= "claybaker99", user.email="crb75@duke.edu")
# install.packages("vcd")
library(vcd) # used for Arthritis data
```

Computing probabilities

```
data(Arthritis)
glimpse(Arthritis)
```

```
#> Rows: 84
#> Columns: 5
#> $ ID        <int> 57, 46, 77, 17, 36, 23, 75, 39, 33, 55, 30, 5, 63, 83, 66...
#> $ Treatment <fct> Treated, Treated, Treated, Treated, Treated, Treated, Tre...
#> $ Sex       <fct> Male, Male, Male, Male, Male, Male, Male, Male, Male, Mal...
#> $ Age       <int> 27, 29, 30, 32, 46, 58, 59, 59, 63, 63, 64, 64, 69, 70, 2...
#> $ Improved  <ord> Some, None, None, Marked, Marked, Marked, None, Marked, N...
```

Take a look at the help for `Arthritis` to understand where this data comes from and the variable meanings.

Let's look at the data in a tabular view. Don't worry about understanding these functions, we're only using it to better visualize our data via a table.

```
xtabs(~ Treatment + Improved, data = Arthritis) %>%
  addmargins()
```

```
#>           Improved
#> Treatment None Some Marked Sum
#>   Placebo   29    7      7  43
#>   Treated   13    7     21  41
#>   Sum       42   14     28  84
```

- How many patients were enrolled in the clinical trial? There are 84 patients in the study.
- What is the probability a randomly selected patient received the placebo? The probability would be 43/84. (marginal prob)
- What is the probability a randomly selected patient received the placebo and had a marked improvement? The probability would be 7/84.
- What is the probability a randomly selected patient received the placebo and the treatment? The probability would be 0.
- What is the probability a randomly selected patient had some improvement or was on the treatment? The probability would (14+41-7)/84.

Using computer simulations to calculate probabilities

Example Recall that a **vector** is the basic building block in R. Let's create a vector called `marbles`.

```
marbles <- c("red", "red", "white", "red", "blue", "blue", "red", "blue")
```

Suppose we draw a single marble from our imaginary box, where all the marbles are equally likely to be selected. What is the probability the marble is blue? How about white? It is 3/8.

We can simulate this "drawing" with the `sample()` function.

```
sample(marbles, size = 1)
```

```
#> [1] "blue"
```

We produced one random outcome from this experiment. To estimate the probability of say getting a white marble, we need to repeat this experiment many many times.

In the `sample()` function we can change the `size` argument and set `replace = TRUE`. Setting `replace = TRUE` allows to draw from our population of eight marbles each time. This way we can easily simulate our marble-drawing experiment.

```
draw_results <- sample(marbles, size = 10000000, replace = TRUE)
```

```
counts <- table(draw_results)
prop.table(counts)
```

```
#> draw_results
#>      blue       red     white
#> 0.3748301 0.5000609 0.1251090
```

How close is this value to the "true" probability? This is very close to the expected probability.

To summarize our process:

1. We defined the sample space for our experiment - `marbles`
2. We simulated this experiment many many times and recorded the outcomes from each of the simulations.
3. We computed the relative frequency of the observed outcomes from our many simulations.

Another example What if we want to compute the probability of getting two marbles of the same color if we make two draws with replacement? We haven't discussed how to compute this theoretically yet, but this is what computers are good at.

Before we do this, what is your guess as to what the probability will be? No, it will be less likely.

We'll still use `sample()` to run our simulation many times, but we'll use `dplyr` functions to compute the relative frequencies.

```
two_draw_results <- tibble(
  draw_1 = sample(marbles, size = 10000, replace = TRUE),
  draw_2 = sample(marbles, size = 10000, replace = TRUE)
)
two_draw_results
```

```
#> # A tibble: 10,000 x 2
#>    draw_1 draw_2
#>    <chr>  <chr>
#>  1 blue   red
#>  2 red    blue
#>  3 red    red
#>  4 blue   white
#>  5 blue   white
#>  6 white  red
#>  7 blue   red
#>  8 red    blue
#>  9 white  red
#> 10 red    red
#> # ... with 9,990 more rows
```

How can we add a variable to `two_draw_results` to see if `draw_1` and `draw_2` match? We can add a boolean to see if the colors are the same.

```
two_draw_results <- two_draw_results %>%
  mutate(color_match = draw_1 == draw_2)
two_draw_results
```

```
#> # A tibble: 10,000 x 3
#>    draw_1 draw_2 color_match
#>    <chr>  <chr>  <lgl>
#>  1 blue   red    FALSE
#>  2 red    blue   FALSE
#>  3 red    red    TRUE
#>  4 blue   white  FALSE
#>  5 blue   white  FALSE
#>  6 white  red    FALSE
#>  7 blue   red    FALSE
#>  8 red    blue   FALSE
#>  9 white  red    FALSE
#> 10 red    red    TRUE
#> # ... with 9,990 more rows
```

All that remains is to compute the relative frequency of the observed outcomes from our many simulations.

```
two_draw_results %>%
  count(color_match) %>%
  mutate(proportion = n / sum(n))
```

```
#> # A tibble: 2 x 3
#>   color_match     n proportion
#> * <lgl>       <int>      <dbl>
#> 1 FALSE        5887      0.589
#> 2 TRUE         4113      0.411
```

Practice

Suppose you roll two fair six-sided dice. Which has a higher probability: the square of dice roll 1 is equal to dice roll 2; or the absolute value of the difference between dice roll 1 and dice roll 2 is equal to 4.

Perform a simulation to compute this empirical probability.

Write down your guess to the answer before you calculate it. I think the difference, since only 1 pair of square and other is value.

```
dice <- c(1,2,3,4,5,6)

two_roll_results <- tibble(
  roll_1 = sample(dice, size = 100000, replace = TRUE),
  roll_2 = sample(dice, size = 100000, replace = TRUE)
)
two_roll_results
```

```
#> # A tibble: 100,000 x 2
#>    roll_1 roll_2
#>     <dbl>  <dbl>
#>  1      2      6
#>  2      1      2
#>  3      3      5
#>  4      5      4
#>  5      3      4
#>  6      4      6
#>  7      1      3
#>  8      2      5
#>  9      4      2
#> 10      4      6
#> # ... with 99,990 more rows
```

```
roll_results <- tibble(
  die_1 = replicate(n=100000, expr = sample(1:6, size=1)),
  die_2 = replicate(n=100000, expr = sample(1:6, size=1))
)
roll_results
```

```
#> # A tibble: 100,000 x 2
#>    die_1 die_2
#>    <int> <int>
#>  1     3     6
#>  2     2     4
#>  3     4     4
#>  4     1     5
#>  5     3     5
#>  6     5     6
#>  7     5     4
#>  8     6     6
#>  9     5     2
#> 10     1     3
#> # ... with 99,990 more rows
```

```
roll_results <- roll_results %>%
  mutate(sq_match = (die_1)^2 == die_2) %>%
  mutate(abs_diff = abs(die_1 - die_2)==4)
```

```
roll_results
```

```
#> # A tibble: 100,000 x 4
#>    die_1 die_2 sq_match abs_diff
#>    <int> <int> <lgl>    <lgl>
#>  1     3     6 FALSE    FALSE
#>  2     2     4 TRUE     FALSE
#>  3     4     4 FALSE    FALSE
#>  4     1     5 FALSE    TRUE
#>  5     3     5 FALSE    FALSE
#>  6     5     6 FALSE    FALSE
#>  7     5     4 FALSE    FALSE
#>  8     6     6 FALSE    FALSE
#>  9     5     2 FALSE    FALSE
#> 10     1     3 FALSE    FALSE
#> # ... with 99,990 more rows
```

```
roll_results %>%
  summarise(sq_match_prob = mean(sq_match),
            abs_diff_prob = mean(abs_diff))
```

```
#> # A tibble: 1 x 2
#>   sq_match_prob abs_diff_prob
#>           <dbl>         <dbl>
#> 1        0.0553         0.111
```

Additional Resources-please look at before Weds.

- Open Intro Stats Sections 3.1 and 3.2