Introduction to probability

Robert Haywood

2/27/2021

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(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?

84 patients.

- What is the probability a randomly selected patient received the placebo?

$43/84 = 0.512$

- What is the probability a randomly selected patient received the placebo and had a marked improvement?

$7/84 = 0.0833$

- What is the probability a randomly selected patient received the placebo and the treatment?

0

- What is the probability a randomly selected patient had some improvement or was on the treatment?

$48/84 = 0.571$

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?

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 = 10000, replace = TRUE)
```

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

```
#> draw_results
#>   blue    red  white
#> 0.3823 0.4990 0.1187
```

How close is this value to the "true" probability?

0.125 is the "true" value, 0.1216 is given by the simulation.

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?

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   blue
#>  2 white  red
#>  3 white  red
#>  4 blue   blue
#>  5 red    red
#>  6 red    red
#>  7 blue   red
#>  8 white  blue
#>  9 red    white
#> 10 white  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?

```
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   blue   TRUE
#>  2 white  red    FALSE
#>  3 white  red    FALSE
#>  4 blue   blue   TRUE
#>  5 red    red    TRUE
#>  6 red    red    TRUE
#>  7 blue   red    FALSE
#>  8 white  blue   FALSE
#>  9 red    white  FALSE
#> 10 white  red    FALSE
#> # ... 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        5943      0.594
#> 2 TRUE         4057      0.406
```

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 first situation would occur $1/15 = 0.067$. I think the second would be $2/15$.

```
dice <- c(1, 2, 3, 4, 5, 6)
two_rolls <- tibble(
  roll1 = sample(dice, size = 10000, replace = TRUE),
  roll2 = sample(dice, size = 10000, replace = TRUE))
two_rolls
```

```
#> # A tibble: 10,000 x 2
#>    roll1 roll2
#>    <dbl> <dbl>
#>  1     4     4
#>  2     6     3
#>  3     6     5
#>  4     5     5
#>  5     5     5
#>  6     1     5
#>  7     2     3
#>  8     5     2
#>  9     2     5
#> 10     4     2
#> # ... with 9,990 more rows
```

```
two_rolls <- two_rolls %>%
  mutate(square = roll1 * roll1 == roll2) %>%
  mutate(diff = abs(roll2 - roll1) == 4)
```

```
two_rolls %>%
  count(square) %>%
  mutate(prop = n / sum(n))
```

```
#> # A tibble: 2 x 3
#>   square     n   prop
#> * <lgl>  <int>  <dbl>
#> 1 FALSE   9463 0.946
#> 2 TRUE     537 0.0537
```

```
two_rolls %>%
  count(diff) %>%
  mutate(prop = n / sum(n))
```

```
#> # A tibble: 2 x 3
#>   diff      n  prop
#> * <lgl> <int> <dbl>
#> 1 FALSE  8847 0.885
#> 2 TRUE   1153 0.115
```

Based on these results I think I was right.

Additional Resources-please look at before Weds.

- Open Intro Stats Sections 3.1 and 3.2