

Introduction to Probability

Alexander B. McMullen

February 22, 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 were enrolled in the trial.

- What is the probability a randomly selected patient received the placebo? Probability of receiving placebo = Patients receiving placebo/total patients. $43/84 = 0.51$
- What is the probability a randomly selected patient received the placebo and had a marked improvement? $7/84 = 0.083$
- What is the probability a randomly selected patient received the placebo and the treatment? **This is disjoint probability. These cannot occur at the same time.**
- 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?

There are 8 marbles total, and each is equally as likely to be selected. The probability of any color being picked is **# of color / 8**.

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

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

```
#> [1] "red"
```

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
#> 0.3773 0.5001 0.1226
```

How close is this value to the “true” probability?

These values are extremely close to the true theoretical probability. The probabilities of each color calculated in Rstudio are each less than 0.005 off from the real value.

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  red  
#> 2 blue  blue  
#> 3 red   blue  
#> 4 blue  red  
#> 5 red   white  
#> 6 blue  white  
#> 7 white red  
#> 8 red   red  
#> 9 blue  blue  
#> 10 red  blue  
#> # ... 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  red    FALSE  
#> 2 blue  blue   TRUE  
#> 3 red   blue   FALSE  
#> 4 blue  red    FALSE  
#> 5 red   white  FALSE  
#> 6 blue  white  FALSE  
#> 7 white red    FALSE  
#> 8 red   red    TRUE  
#> 9 blue  blue   TRUE  
#> 10 red  blue   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         5957      0.596  
#> 2 TRUE          4043      0.404
```

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.

```
two_dice_roll <- tibble(  
  roll1 = replicate(n = 10000, expr = sample(1:6, size = 1)),  
  roll2 = replicate(n = 10000, expr = sample(1:6, size = 1))
```

#My guess is that the absolute value will be more common.

```
two_dice_roll <- two_dice_roll %>%  
  mutate(square_roll = roll1^2 == roll2, abs_roll = abs(roll1-roll2) == 4)
```

```
two_dice_roll %>%  
  summarize(square_roll_prob = mean(square_roll),  
            abs_roll_prob = mean(abs_roll))
```

```
#> # A tibble: 1 x 2  
#>   square_roll_prob abs_roll_prob  
#>       <dbl>         <dbl>  
#> 1         0.0559         0.112
```

Additional Resources-please look at before Weds.

- Open Intro Stats Sections 3.1 and 3.2