Sampling Distributions

ML

10/20/2021

Transitioning to Probabilty

Today we'll move from multivariate descriptions to probability, leading to confidence intervals. Load the `cupid` dataset and the usual packages to get started.

BACK TO SLIDES; WE'LL RETURN TO R SOON

Introducing Probability And Z-Scores

A z-score or standardized value is a value's distance from the mean in standard deviations. It is calculated as: $z = \frac{x-\mu}{\sigma}$. In words, the z-score is the difference between the observed value and the sample mean divided by the standard deviation.

After confirming that `height` is approximately normally distributed, let's make a new variable with the standardized values of `height`:

```
cupid <- mutate(cupid,
     height_z = (height - mean(height)) /
                          sd(height))
```

Z-scores should be normally distributed with a mean of 0 and a standard deviation of 1. Were we successful?

REPLACE THIS LINE WITH YOUR CODE

```
summary(cupid$height_z)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -2.92468 -0.86381 -0.09099  0.00000  0.68184  3.00031
```

```
sd(cupid$height_z)
```

```
## [1] 1
```

What is the z-score for a height of 71 inches?

REPLACE THIS LINE WITH YOUR CODE

```
mean(cupid$height_z[cupid$height==71])
```

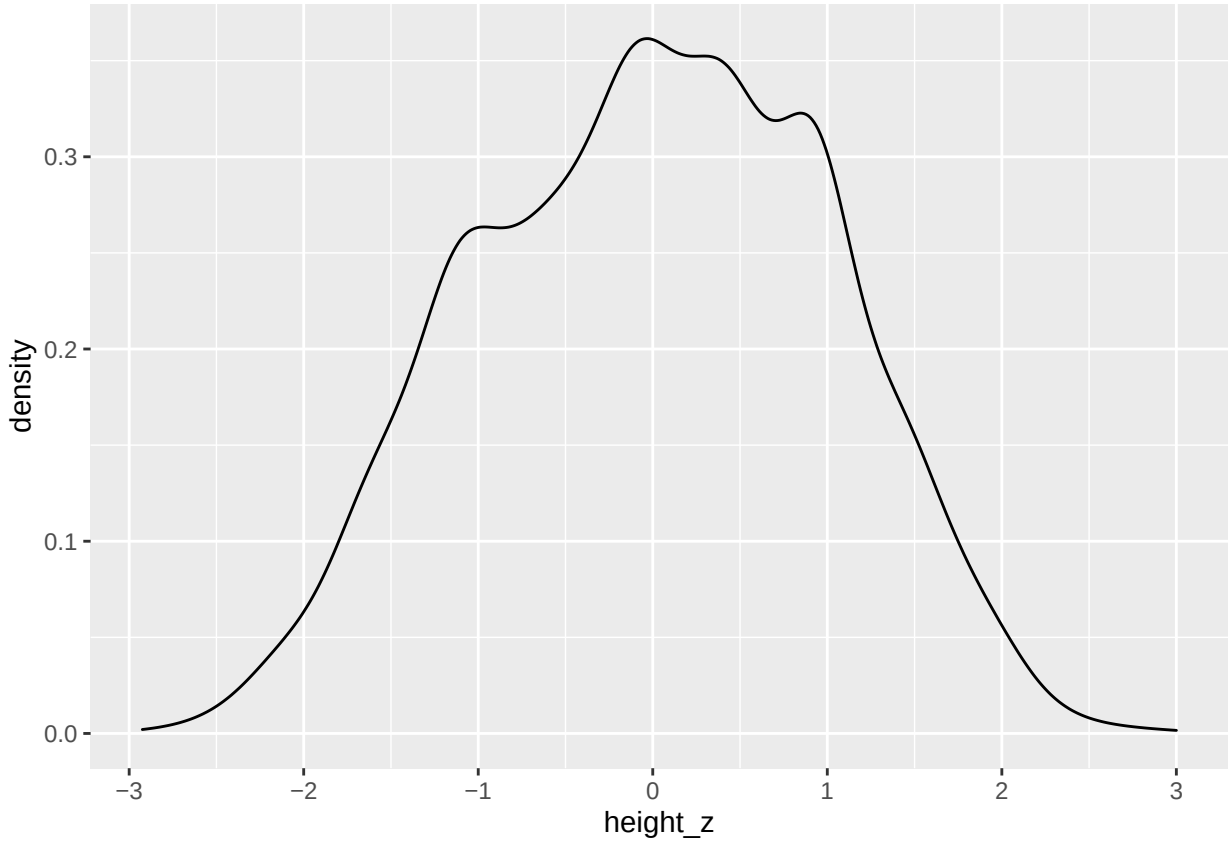
```
## [1] 0.6818372
```

When we plot standardized values that are approximately normal, we now know a lot about the proportion of observations falling along different points of the distribution. To see how, make a density plot showing the distribution of the standardized heights.

REPLACE THIS LINE WITH YOUR CODE

```
height_z_plot <- ggplot(cupid, aes(x = height_z))

height_z_plot + geom_density()
```



To find the probability of getting any z-score, use `dnorm()`. Think about this value as the y axis intersection with the density curve for any specific value on the x axis. For example, the probability that a randomly pulled observation in our sample would have a height of 71 inches is:

```
dnorm(0.682)
```

```
## [1] 0.316162
```

Probabilities of specific values are more helpful for descriptives than for inference. Moving forward, what is more helpful is knowing the probability of randomly pulling a value that is greater than or less than an observed value. In other words, we want to add up the probabilities of pulling any value less than the value for 71 inches.

We get that summed probability by thinking not of the density but of the *cumulative density*. The cumulative density is also the percentile.

If you have the z-value and want the percentile associated with it, use `pnorm()`. For a height of 71 inches:

```
pnorm(.682)
```

```
## [1] 0.7523805
```

The pnorm() function will give you the proportion of the distribution to the left of the z score. So about 75% of respondents in our sample are shorter than 71 inches.

Use what you know to find out what you don't know!

What is the probability of another respondent being taller than 71 inches?

REPLACE THIS LINE WITH YOUR CODE

```
1 - pnorm(.682)
```

```
## [1] 0.2476195
```

What's the point?

The key bridge to inference is thinking of the x-axis not as observed values of height in our sample but as possible values of the true mean of height in the population. We want to know how close the mean in our observed sample is to the true (unobserved) population mean. Knowing where it falls in the distribution of all the possible sample means is how we infer how similar the sample mean and the population are. Remember our new language: what is the probability of another randomly drawn sample mean being more extreme than our sample mean *simply by chance*.

To find out, we need to use our knowledge of sampling distributions. We won't use bootstrapping to pull repeated samples. But we'll use the *standard deviation* of our sample to calculate the *standard error* of the sampling distribution.

Standard Errors

The standard deviation of the sampling distribution is called the *standard error*. It is calculated as:

$\sigma_{\bar{y}} = \frac{\sigma}{\sqrt{n}} = \frac{sd}{\sqrt{samplesize}}$

Let's find the standard error of the `age` variable. We'll save this as an object, not as a new variable (since it is the same for the entire sample):

```
age_se <- sd(cupid$age) / sqrt(length(cupid$age))

age_se
```

```
## [1] 0.1847327
```

From Standard Errors to Confidence Intervals

For a 95% confidence interval, we need the z-scores that are associated with .025 and .975. To find them, use `qnorm()`:

```
qnorm(.025)
```

```
## [1] -1.959964
```

```
qnorm(.975)
```

```
## [1] 1.959964
```

In common practice, we round this to 1.96. We'll use this number a lot; remember it so you don't have to use `qnorm()` every time you need it.

Margin of Error

The margin of error is the z-score associated with the confidence interval we are constructing multipled by the standard error:

```
1.96*age_se
```

```
## [1] 0.3620762
```

Building Confidence Intervals

The sample mean plus and minus the margin of error is the confidence interval.

For the *lower limit* of the confidence interval:

```
age_ll <- mean(cupid$age) - 1.96*age_se
```

For the *upper limit* of the confidence interval:

```
age_ul <- mean(cupid$age) + 1.96*age_se
```

Save the 95% confidence interval in a vector:

```
age_ci <- c(age_ll, mean(cupid$age), age_ul)

age_ci
```

```
## [1] 32.02192 32.38400 32.74608
```

How do you interpret this confidence interval? 95% of the repeated samples we might imagine pulling would be expected to have means within this range. There is only a 5% chance that the true population mean falls outside this range.

Exercise

What is the 95% confidence interval for height?

REPLACE THIS LINE WITH YOUR CODE

```
height_se <- sd(cupid$height) / sqrt(length(cupid$height))

height_se
```

```
## [1] 0.0776373
```

```
height_ll <- mean(cupid$height) - 1.96*height_se
height_ul <- mean(cupid$height) + 1.96*height_se
height_ci <- c(height_ll, mean(cupid$height), height_ul)
height_ci
```

```
## [1] 68.20103 68.35320 68.50537
```