Sampling distribution

Katie Schuler

Acknowledgement

These notes are inspired by a MATLAB course by Kendrick Kay

Exploring a simple dataset

Suppose we measure a single quantity of interest (the simplest possible dataset!): the brain volume of human adults. How do we explore these data?

We can create a visual summary of our dataset with a **histogram**. A histogram plots the distribution of a set of data, which allows us to get a quick visual of the data: formally we have plotted the the frequency distribution (count) of the data, but this also gives a sense of the central tendency and variability in our dataset.

We can summarize (or describe) a set of data with **summary statistics** (aka descriptive statistics). There are three summary stats we typically use:

- **central tendency** describes a central or typical value (mean, median, mode)
- **variability** describes dispersion or spread of values (variance, standard deviation, range, IQR)
- **frequency distribution** describes how frequently different values occur (count)

R has built-in functions to handle descriptive statistics (we saw these in lecture 1):

```
data %>%
    summarise(
        n = n(),
        mean = mean(volume),
        median = median(volume),
        sd = sd(volume),
        iqr_lower = quantile(volume, 0.25),
```

```
        iqr_upper = quantile(volume, 0.75)
    )
```

Some statistics are considered **parametric** because they make assumptions about the the distribution of the data (can therefore be computed theoretically from parameters). The mean and standard deviation assume the distribution is **Gaussian** (aka "normal") and can therefore be computed via the following equations:

- $mean(x) = \overline{x} = \frac{\sum_{i=i}^{n} x_i}{n}$
- $sd(x) = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \overline{x})^2}{n-1}}$

Other statistics are **non-parametric** because they make minimal assumptions about the distribution of the data:

- **median** is the 50th percentile, the value below which 50% of the data points fall.
- **inter-quartile range (IQR)** is the difference between the 25th and 75th percentiles (sometimes called the **50% coverage interval** because 50% of the data fall in this range).

- Note that we can calculate any arbitrary coverage interval. The 95% coverage interval — widely used in the sciences — is the difference between the 2.5 percentile and the 97.5 percentile, including all but 5% of the data.

Probability distributions

A **probability distribution** (aka "probability density function (PDF)") is a mathematical function that describes the probability of observing the different possible values of a variable (or variables!)

One of the simplest probability distributions is the **uniform distribution**, where all possible values of a variable are equally likely. The probability density function for the uniform distribution is given by the following equation with two parameters (the boundaries, `min` and `max`):

- $p(x) = \frac{1}{max - min}$

One of the most useful probability distributions for our purposes is the **Gaussian (or Normal) distribution**. The probability density function for the Gaussian distribution is given by the following equation, with the parameters μ (mean) and σ (standard deviation):

- $p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$

- The Gaussian distribution assumes that the distribution of a set of data takes a certain form (is unimodal, symmetric, etc).
- When values are sampled from a Gaussian distribution, 68% of the values will be within one standard deviation from the mean and 95% within two standard deviations from the mean.
- When computing the mean and standard deviation of a set of data, we are fitting a Gaussian distribution to the data.

The probability distributions we've discussed so far are considered "parametric" because they are given by one or more parameters. Base R has four useful functions we will use to work with probability distribution.

- `dnorm(n, mean=5, sd=1)` returns the height of the **probability density function** at the given values
- `pnorm(5, mean=5, sd=1)` returns the **cumulative density function** (the probability that a random number from the distribution will be less than the given values)
- `qnorm(0.8, mean=5, sd=1)` returns the value whose cumulative distribution matches the probability (**inverse of `p`**)
- `rnorm(1000, mean=5, sd=1)` returns n **random numbers** generated from the distribution

To use another distribution, change the function's suffix to the name of the distribution and the parameters to those that define the distribution. For example, to generate n random numbers from a uniform distribution with a min of 1 and a max of 5, run `runif(n, min=0, max=1)`

What if the data does not meet the assumptions of the Gaussian distribution? One option is to choose another parametric probability distribution (run `help(Distributions)` for a full list of available distributions). Another is to use a nonparametric approach, where the probability distribution is not determined by parameters but is instead determined by the data.

Sampling distribution

When measuring some quantity we are usually interested in knowing something about the **population** (the height of human adults, for example). But in practice we can only observe a small **sample** of the entire population.

- Any statistic we compute from a random sample we've collected (technically known as the **parameter estimate**) will be subject to sampling variability and will differ from that statistic computed on the entire population (technically known as the **parameter**). In other words, our measurements are noisy, and we need a way to express our uncertainty on the statistic we've computed. Quantifying this sampling variability is an important component of statistical inference.

- The **sampling distribution** is the probability distribution of the values our parameter estimate can take on. We can construct the sampling distribution by taking a random sample, computing the statistic of interest, and repeating this process many times. The spread of these results indicates how the parameter estimate will vary from different random samples.
- We can quantify the spread of our results (AKA express our uncertainty on our parameter estimate) using a parametric approach, by computing the standard deviation of our sampling distribution (called standard error!), or using a nonparametric approach, by constructing a confidence interval.

The standard deviation of the sampling distribution is known as the **standard error**. When the statistic of interest is the mean, the standard error is given by the following equation, where σ is the standard deviation of the population and n is the sample size: $\frac{\sigma}{\sqrt{n}}$

- In practice, the standard deviation of the population is unknown, so we use the standard deviation of the *sample* as an estimate. *That* is why we use $n-1$ in our mean square calculation. We assume our sample standard deviation is probably underestimating the population, so we "correct" this by dividing by $n-1$ instead of n.
- Standard error is considered parametric because we assume a parametric probability distribution (Gaussian) and compute the standard error based on what happens theoretically when we sample that distribution.
- clt? sample size relationship.

We can also quantify the sampling variability with a **confidence interval**, which expresses our uncertainty on our parameter estimate via a coverage interval. We can construct any confidence interval, but in science the convention is to choose the 95% coverage interval.

- Recall from last lecture that a coverage interval is a nonparametric statistic. The 95% coverage interval are the values between which 95% of the data points fall (the difference between the 2.5 percentile and the 97.5 percentile in our sampling distribution).
- Confidence intervals are closely related to standard error: assuming the sampling distribution is Gaussian (the parametric approach), the 68% confidence interval is +/- 1 standard error and the 95% confidence interval is +/- 2 standard error.

Bootstrapping

Ideally, we would construct the sampling distribution by repeating our experiment many times, drawing new random samples from the population each time. But in practice, this is impossible. We are usually constrained — by time, money, access, etc. — such that we can only take *one* sample.

- This is no problem if we can assume the underlying population distribution is Gaussian: we can just compute the standard error, which relies on the sample standard deviation,

to approximate what would happen if we *had* sampled from a Gaussian probability distribution (see above!).
- What if the underlying distribution is not Gaussian, or we want to drop these parametric assumptions (i.e., the assumption that our distribution is well-characterized by mean and standard deviation)? We can use a technique called bootstrapping.

With **bootstrapping**, instead of assuming a parametric probability distribution, we can use the data themselves to approximate the underlying probability distribution. In other words, instead of sampling from the population, we can sample our sample! We're "pulling ourselves up by our bootstraps": constructing the sampling distribution from our own data.

- The procedure is very simple. To illustrate, suppose we have a set a data with 100 data points. We generate the bootstrap sampling distribution by drawing the same number of data points (100) *with replacement* from our data set and compute the parameter estimate — mean, median, whatever — on those points, then we repeat the process many times.
- However, there is "no free lunch". Bootstrapping still relies on the (weaker) assumption that your sample is a *representative sample* of the population. If your sample was not representative, then bootstrapping will not help you estimate the parameters any better. Garbage in, garbage out!

There are many ways to generate a bootstrap sampling distribution in R. We will use the `infer` package in this class, which was developed by Hadley Wickham (the `tidyverse` guy!) and others to simplify aspects of statistical inference in R.

- `specify(response=x)`: choose which variable is the focus of our inference
- `generate(reps=n, type="bootstrap")`: generate n replicates of the data
- `calculate(stat="mean")`: statistic to calculate on each sample; what parameter are you trying to estimate?

We can further use `infer` to visualize the bootstrap sampling distribution and get a confidence interval around the parameter we estimated.

- `visualize()`: quick visualization of the distribution
- `get_confidence_interval(level=0.95, type="percentile")`: computes the confidence interval
- `shade_ci(endpoints=c(min, max))`: shades the visualization with the computed confidence interval

Further Reading

Suggested further reading:

- Basics of descriptive statistics in Statistics for linguists

- Appendix A: Statistical Background in Modern Dive
- Ch 7 Sampling in Modern Dive
- Ch 8 Bootstrapping and Confidence Intervals in Modern Dive
- Infer package introductory vignette
- Ch 11: Modeling Randomness in Statistical Modeling