BMR 617

Confidence Interval of a Mean

Confidence Intervals

- Remember our framework for confidence intervals:
- We gather a sample of data and compute some statistic from it
  - In this presentation, this statistic will be the mean
- The idea is that this statistic for the sample approximates the population value
- We choose a confidence level
  - Typically 95%
- We compute a range of values that we are 95% confident contains the population value

Using the central limit theorem to calculate the confidence interval of a mean

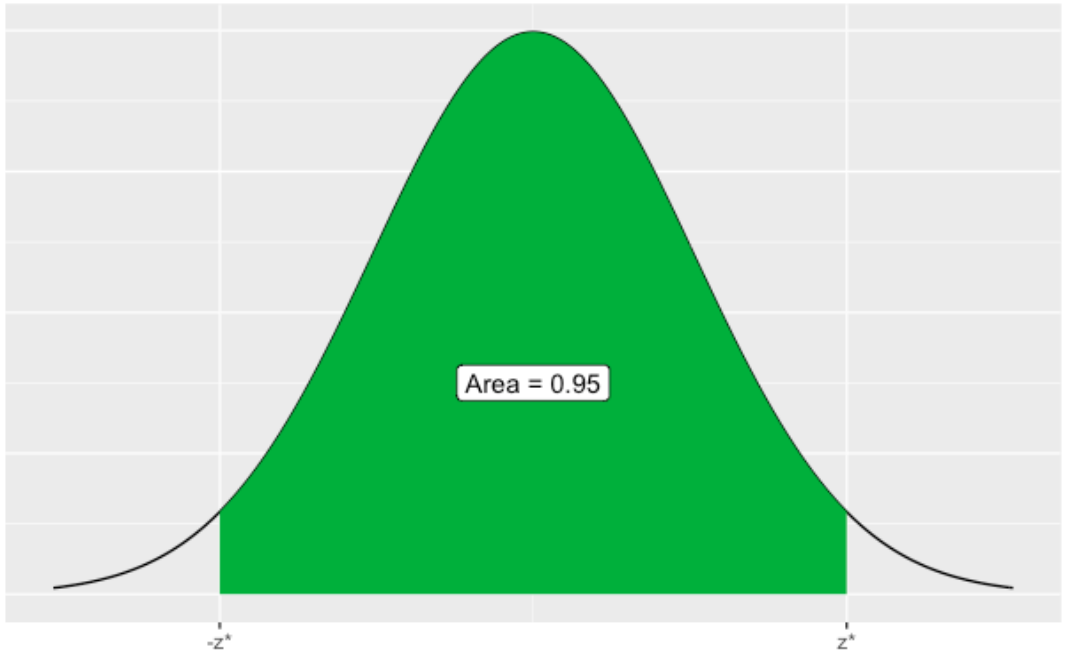
- Suppose we take a sample of size *n*, and compute the mean of the sample, which we call *m*
- For concreteness, you can think of the Cholesterol measure for one of our mouse groups (e.g. TH mice fed a Chow diet)
- The Central Limit Theorem tells us that the set of all possible sample means is approximately normally distributed with mean μ and standard deviation $\sigma/\sqrt{n}$, where μ and σ are the mean and standard deviation of the population, respectively.

Using the central limit theorem to calculate the confidence interval of a mean

- Properties of normal distributions tell us that the quantity

$$\frac{m-\mu}{\sigma/\sqrt{n}}$$

  is normally distributed with mean 0 and standard deviation 1
- This means we can find a value z^* with the property that the probability that $\frac{m-\mu}{\sigma/\sqrt{n}}$ lies between $-z^*$ and $z^*$is, for example, 0.95.



Using the central limit theorem to calculate the confidence interval of a mean

- Saying that $\frac{m-\mu}{\sigma/\sqrt{n}}$ lies between $-z^*$ and z^* is equivalent to

$$-z^* < \frac{m-\mu}{\sigma/\sqrt{n}} < z^*$$

which in turn is equivalent to

$$-\frac{\sigma z^*}{\sqrt{n}} < m-\mu < \frac{\sigma z^*}{\sqrt{n}}$$

which means the distance between m and μ is less than $\frac{\sigma z^*}{\sqrt{n}}$, which we can state as

$$m-\frac{\sigma z^*}{\sqrt{n}} < \mu < m+\frac{\sigma z^*}{\sqrt{n}}$$

i.e. there is a 95% probability the range $\left[m-\frac{\sigma z^*}{\sqrt{n}}, m+\frac{\sigma z^*}{\sqrt{n}}\right]$ contains the true mean μ.

We don't usually know σ

- The previous two slides produced a formula for the confidence interval for the mean
- However, this formula relied on σ, the *population* standard deviation
- We almost never know this

The t-distribution

- The previous work relied on the fact that

$$\frac{m-\mu}{\sigma/\sqrt{n}}$$

  was normally distributed with mean 0 and standard deviation 1
- Or more generally, that it had a known distribution
- In the 1920s, William Sealy Gosset computed the distribution for the quantity

$$\frac{m-\mu}{s/\sqrt{n}}$$

  under the additional assumption that the population is normally distributed
- This distribution is called the t-distribution (sometimes "Student's t-distribution)

Computing the confidence interval of the mean when the population is normally distributed

- We can do the same math we did before:
- Find t^* so that the probability a value in the t-distribution lies between $-t^*$ and t^* is 0.95 (I'll show how to do this soon)
- This means

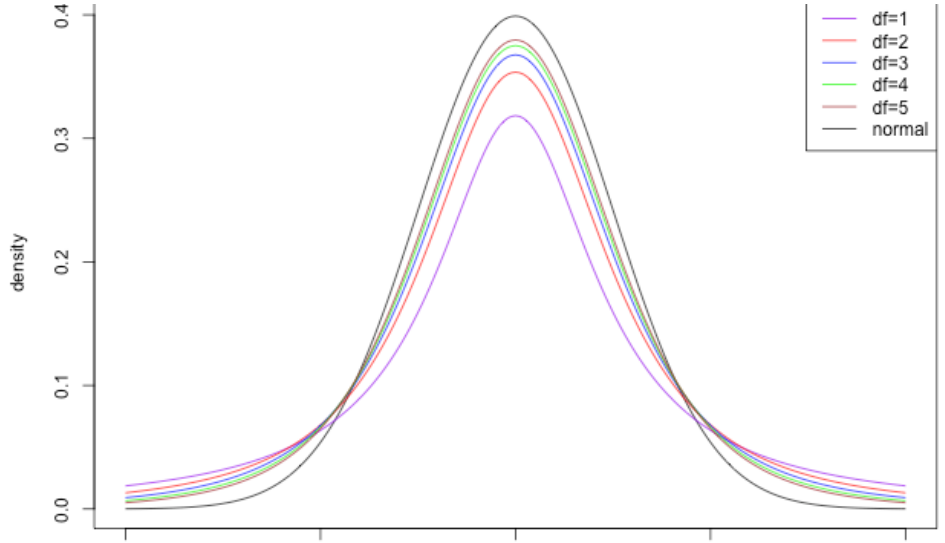
$$P\left(-t^* < \frac{m-\mu}{s/\sqrt{n}} < t^*\right) = 0.95$$

$$P\left(m - \frac{t^* s}{\sqrt{n}} < \mu < m + \frac{t^* s}{\sqrt{n}}\right) = 0.95$$

i.e. $\left[m - {t^* s}/{\sqrt{n}}, m + {t^* s}/{\sqrt{n}}\right]$ is the 95% confidence interval for the mean

More about the t-distribution

- The t-distribution is really a family of distributions, which depend on a value called the number of degrees of freedom
- If the number of degrees of freedom is larger (say, larger than about 15), the t-distribution is almost identical to the normal distribution.
- For small degrees of freedom (2, 3, or 4), the distributions are quite different.
- For the previous discussion, the number of degrees of freedom is $n - 1$.

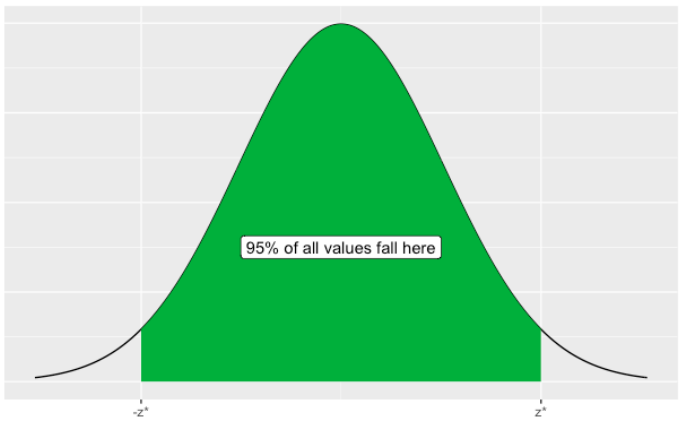


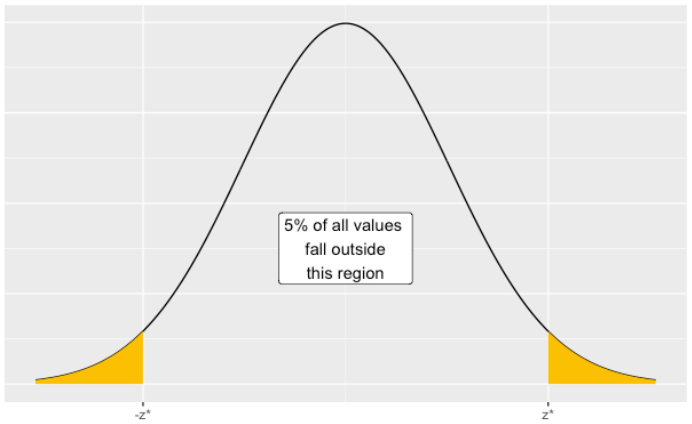
Finding t^* (or z^*) in R

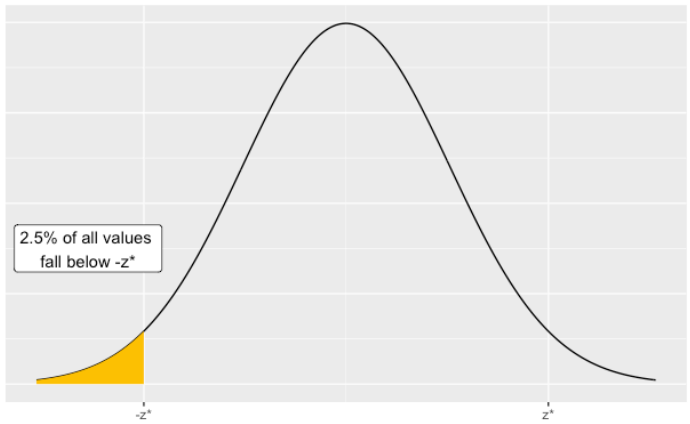
- For each well-known distribution, R has a number of functions
- The quantile functions give a value x for which $P(X < x) = p$, assuming X follows the distribution (p is passed to the function)
- So, for example
  `qnorm(0.7)`
  gives 0.5244, which means $P(X < 0.5244) = 0.7$ if X is normally distributed (with mean 0 and sd 1)
- Similarly,
  `qt(0.7, df=3)`
  gives 0.5844, which means $P(X < 0.5844) = 0.7$ if X is distributed according to the t-distribution with 3 degrees of freedom

Care in finding t^* (or z^*)

- To find z^* for a 95% confidence interval, we want
$$P(-z^* < X < z) = 0.95$$
- In pictures:



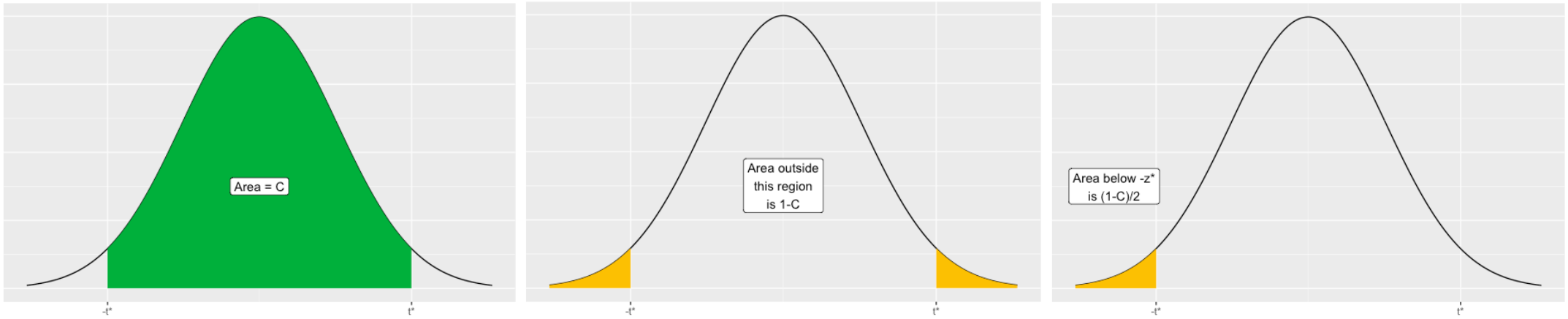




- So 95%+2.5%=97.5% fall below z^*

Care in finding t^* (or z^*)

- Generally, to find t^* for a confidence level C:



- So the value to pass to `qt(…)` is $C + \frac{1-C}{2} = \frac{1+C}{2}$

Example

- Find the 95% confidence interval for the Cholesterol level for TH mice fed the Chow diet

```
library(tidyverse)
met <- read_csv("https://denvirlab.marshall.edu/BMR617-2021/data/TH-B6-metabolic.csv")
met <- separate(met, MouseID, sep="-", into=c("Strain", "Diet", "ID"))
# Filter for TH Chow:
th_chow <- filter(met, Strain=="TH" & Diet == "Chow")
# Get the cholesterol values from the table:
chol_th_chow <- pull(th_chow, Cholesterol)
m <- mean(chol_th_chow)
s <- sd(chol_th_chow)
n <- length(chol_th_chow)
conf_level <- 0.95
t_star <- qt((1+conf_level)/2, df=n-1)
conf_int_lower <- m - t_star*s/sqrt(n)
conf_int_upper <- m + t_star*s/sqrt(n)
```

Writing functions in R

- To get the confidence interval took seven lines of R code!
- Each time we want the confidence interval, we could repeat these lines of code
- A better way is to write our own function, so we can just call the function
- The general syntax is

```
conf_int <- function(x) {
  # ...
  return(result_of_calculation)
}
```

- We can allow optional parameters to the function by giving a default value:

```
conf_int <- function(x, conf_level=0.95) {
  # ...
  return(result_of_calculation)
}
```

A function to compute confidence intervals

```
conf_int <- function(x, conf_level=0.95) {
  m <- mean(x)
  s <- sd(x)
  n <- length(x)
  t_star <- qt((1+conf_level)/2, df=n-1)
  conf_int_lower <- m - t_star*s/sqrt(n)
  conf_int_upper <- m + t_star*s/sqrt(n)
  return(c(conf_int_lower, conf_int_upper))
}
```

Using our function

- We can call our function to get the 95% confidence interval using

```
conf_int(chol_th_chow)
```

- To get a 99% confidence interval:

```
conf_int(chol_th_chow, conf_level=0.99)
```

- Note we can get individual ends using

```
conf_int(chol_th_chow)[[1]]
conf_int(chol_th_chow)[[2]]
```

Bar Charts with confidence intervals as error bars

- Last time we discussed error bars on bar charts:
- Standard deviation is descriptive of the sample
- Standard Error of the Mean is a measure of how accurately the sample mean approximates the population mean
- Confidence intervals arguably give a better depiction of this
  - We are 95% (for example) confident the range shown in the error bar captures the population mean
- Confidence intervals are not commonly used as error bars
  - But worth considering

Creating bar charts with confidence intervals as error bars

- We can leverage our function for computing confidence intervals in our graphing code:

```
met_grouped <- group_by(met, Strain, Diet)
met_summary <- summarize(met_grouped,
      MeanCholesterol = mean(Cholesterol),
      lowerCI = conf_int(Cholesterol, conf_level=0.95)[[1]],
      upperCI = conf_int(Cholesterol, conf_level=0.95)[[2]])
barplot <- ggplot(met_summary, aes(x=Diet, y=MeanCholesterol,
fill=Strain)) +
  geom_bar(stat='identity', position=position_dodge())
ciError <- geom_errorbar(aes(ymin=lowerCI, ymax=upperCI),
                         position=position_dodge(width=0.9), width=0.2)
barplot + ciError
```

Practice!

- Create some bar charts:
- Use SD, SEM, and Confidence Intervals for the error bars
- Experiment with different confidence levels for the confidence interval error bars
- Create bar charts for Cholesterol, Triglycerides, Fat Mass, Body Weight, Insulin
- For each, ask whether the strain appears to affect the variable, the diet appears to affect the variable, and whether the diet effect is the same in both strains. Which group has the least variation? The most?
- Modify the x- and y-axis labels, and add a title
- Experiment with the colors: see http://www.sthda.com/english/wiki/ggplot2-colors-how-to-change-colors-automatically-and-manually for some examples