Tutorial 04: One Sample Confidence Intervals

```
#| edit: false
#| output: false
webr::install("gradethis", quiet = TRUE)
library(gradethis)
options(webr.exercise.checker = function(
  label, user_code, solution_code, check_code, envir_result, evaluate_result,
  envir_prep, last_value, engine, stage, ...
) {
  if (is.null(check_code)) {
    # No grading code, so just skip grading
    invisible(NULL)
  } else if (is.null(label)) {
    list(
      correct = FALSE,
      type = "warning",
      message = "All exercises must have a label."
    )
  } else if (is.null(solution_code)) {
    list(
      correct = FALSE,
      type = "warning",
      message = htmltools::tags$div(
        htmltools::tags$p("A problem occurred grading this exercise."),
        htmltools::tags$p(
          "No solution code was found. Note that grading exercises using the ",
          htmltools::tags$code("gradethis"),
          "package requires a model solution to be included in the document."
        )
      )
    )
  } else {
    gradethis::gradethis_exercise_checker(
```

```
      label = label, solution_code = solution_code, user_code = user_code,
      check_code = check_code, envir_result = envir_result,
      evaluate_result = evaluate_result, envir_prep = envir_prep,
      last_value = last_value, stage = stage, engine = engine)
  }
})
```

Q1 — Z Critical Value

Print the 95% 2-sided Z^* critical value when $\alpha = 0.05$.

> **Info**
>
> Remember: Critical values represent points on a distribution which play an important role in both confidence intervals and hypothesis testing. They are the cutoffs on a reference distribution that set the width of confidence intervals.

```
#| exercise: q1_zstar_95
#| exercise.lines: 1
#| echo: false
```

Use qnorm.

Solution.

```
#| exercise: q1_zstar_95
#| solution: true
qnorm(0.975)
```

```
#| exercise: q1_zstar_95
#| check: true
gradethis::grade_this({
exp <- qnorm(0.975)
x <- .result
if (!is.numeric(x) || length(x)!=1L || !is.finite(x)) fail("Print a single numeric value.")
if (abs(x - exp) < 1e-6) pass("  Correct Z* for 95%.")
else fail("Not quite - check the percentile for a two-sided 95% interval.")
})
```

Q2 — One Sample Z Confidence Interval

The penguins dataset contains measurements for 3 penguin species from the Palmer Archipelago. Use body_mass_g (grams). Assume the population SD is known: = 450 g. Construct a 90% Z-interval for the true mean body mass. Print c(lower, upper).

> **Info**
>
> When is known, the margin of error uses the standard normal critical value.

> **Preview**
>
> Run this code chunk to get a glimpse of the dataset. Feel free to change the values to visualize more/less number of rows.
>
> ```
> #| echo: true
> if (!requireNamespace("palmerpenguins", quietly = TRUE)) webr::install("palmerpenguins")
> library(palmerpenguins)
> df <- na.omit(palmerpenguins::penguins[, c("species","island","sex","body_mass_g")])
> head(df, 10)
> summary(df$body_mass_g)
> ```

> **Info**
>
> When is known,
>
> $$\mathrm{CI}_{1-\alpha}: \ \bar{x} \pm x_{1-\alpha/2} \frac{\sigma}{\sqrt{n}}.$$

```
#| exercise: q2_z_ci_penguins
#| exercise.lines: 4
#| echo: false
if (!requireNamespace("palmerpenguins", quietly = TRUE)) webr::install("palmerpenguins")
library(palmerpenguins)
x <- na.omit(palmerpenguins::penguins$body_mass_g)
xbar <-
n <-
z <-
sigma <- 450
```

Compute the sample mean, pair with the known and your Z critical value. Output c(lower, upper).

Solution.

```
#| exercise: q2_z_ci_penguins
#| solution: true
if (!requireNamespace("palmerpenguins", quietly = TRUE)) webr::install("palmerpenguins")
library(palmerpenguins)
x <- na.omit(palmerpenguins::penguins$body_mass_g)
xbar <- mean(x)
n <- length(x)
z <- qnorm(0.95) # 90% two-sided
sigma <- 450
xbar + c(-1,1) * z * sigma / sqrt(n)
```

```
#| exercise: q2_z_ci_penguins
#| check: true
gradethis::grade_this({
if (!requireNamespace("palmerpenguins", quietly = TRUE)) webr::install("palmerpenguins")
library(palmerpenguins)
x <- na.omit(palmerpenguins::penguins$body_mass_g)
xbar <- mean(x); n <- length(x); z <- qnorm(0.95); sigma <- 450
exp <- xbar + c(-1,1) * z * sigma / sqrt(n)

xres <- .result
ok <- is.numeric(xres) && length(xres)==2L && all(is.finite(xres))
if (!ok) fail("Print a numeric vector c(lower, upper).")
if (max(abs(xres - exp)) < 1e-6) pass("  Correct 90% Z-interval for penguin body mass.")
else fail("Recheck mean, n, z*, and the known  .")
})
```

Q3 — T Critical Value

PlantGrowth is a dataset that contains results from an experiment to compare yields (as measured by dried weight of plants) obtained under a control and two different treatment conditions. Print the 95% two-sided t* using PlantGrowth's weight column when $\alpha = 0.05$.

Photo by Nagy Arnold on Unsplash

> **Preview**
>
> Run this code chunk to get a glimpse of the dataset. Feel free to change the values to visualize more/less number of rows.

```
#| echo: true
plantgrowth_data <- datasets::PlantGrowth
head(plantgrowth_data, 10)
```

```
#| exercise: q3_tstar_95
#| exercise.lines: 1
#| echo: false
data("PlantGrowth")
n <-
```

Find n from the weights vector and print the two-sided 95% t critical value using df = n − 1. Refer to the previous tutorial for a hint about the R command used to print t quantiles.

Solution.

```
#| exercise: q3_tstar_95
#| solution: true
data("PlantGrowth")
n <- length(PlantGrowth$weight)
qt(0.975, df = n - 1)
```

```
#| exercise: q3_tstar_95
#| check: true
gradethis::grade_this({
n <- length(datasets::PlantGrowth$weight)
exp <- qt(0.975, df = n - 1)
x <- .result
if (!is.numeric(x) || length(x)!=1L || !is.finite(x)) fail("Print a single numeric value.")
if (abs(x - exp) < 1e-6) pass("  Correct t* for 95%.")
else fail("Check df and make sure it's two-sided 95%.")
})
```

Q4 — One-Sample Confidence Interval for (Unknown)

Using PlantGrowth$weight (unknown), construct a 95% CI for the true mean plant weight. Give your output as a vector of the form c(lower, upper).

Preview

Run this code chunk to get a glimpse of the dataset. Feel free to change the values to visualize more/less number of rows.

```r
#| echo: true
data("PlantGrowth")
head(PlantGrowth, 10)
```

Info

When is unknown, replace with the sample SD s and use the t distribution:

$$\mathrm{CI}_{1-\alpha}: \ \bar{x} \pm t_{1-\alpha/2,\, n-1} \frac{s}{\sqrt{n}}, \ \mathrm{df} = n - 1.$$

```r
#| exercise: q4_t_ci_weight
#| exercise.lines: 1
#| echo: false
data("PlantGrowth")
xbar <-
sd <-
n <-
critical_value <-
```

Form "center ± margin of error": center is the sample mean; margin uses the t critical value with df = n−1, the sample SD, and √n. Output c(lower, upper).

Solution.

```r
#| exercise: q4_t_ci_weight
#| solution: true
data("PlantGrowth")
xbar <- mean(PlantGrowth$weight);
sd <- sd(PlantGrowth$weight);
n <- length(PlantGrowth$weight)
critical_value <- qt(0.975, df = n - 1)
xbar + c(-1,1)*critical_value*sd/sqrt(n)
```

```r
#| exercise: q4_t_ci_weight
#| check: true
```

```
gradethis::grade_this({

x <- datasets::PlantGrowth$weight
xbar <- mean(x); s <- sd(x); n <- length(x)
tstar <- qt(0.975, df = n - 1)
exp <- xbar + c(-1,1) * tstar * s / sqrt(n)

r <- .result
ok <- is.numeric(r) && length(r)==2L && all(is.finite(r))
if (!ok) fail("Print a numeric vector c(lower, upper).")
if (max(abs(r - exp)) < 1e-6) pass("  Correct 95% t-interval for PlantGrowth$weight.")
else fail("Recheck x̄, s, n, and t* with df = n-1.")
})
```

Q5 — One sample CI for

The dataset used here is river lengths, which gives the lengths (in miles) of 141 "major" rivers in North America, as compiled by the US Geological Survey. Assume river lengths (miles) have known population SD $= 300$. Build a 95% CI for the true mean river length using rivers whe $\alpha = 0.05$.

Photo by kazuend on Unsplash

> **Preview**
>
> Run this code chunk to get a glimpse of the dataset. Feel free to change the values to visualize more/less number of rows.
>
> ```
> #| echo: true
> rivers_data <- data.frame(length_miles = as.numeric(datasets::rivers))
> head(rivers_data, 10)
> ```

> **Info**
>
> When is unknown, use the t distribution with degrees of freedom $n - 1$.

```
#| exercise: q5_zstar_95
#| exercise.lines: 1
#| echo: false
rivers_data <- data.frame(length_miles = as.numeric(datasets::rivers))
```

Find n from the length_miles vector and compute the two-sided 95% z critical value. Then combine it using the formula you have already used in previous questions.

Solution.

```
#| exercise: q5_zstar_95
#| solution: true
rivers_data <- data.frame(length_miles = as.numeric(datasets::rivers))
x <- rivers_data$length_miles
xbar <- mean(x); n <- length(x); z <- qnorm(0.975); sigma <- 300
xbar + c(-1,1) * z * sigma / sqrt(n)
```

```
#| exercise: q5_zstar_95
#| check: true
gradethis::grade_this({
x <- as.numeric(datasets::rivers)
xbar <- mean(x); n <- length(x); z <- qnorm(0.975); sigma <- 300
exp <- xbar + c(-1,1)*z*sigma/sqrt(n)
xres <- .result
ok <- is.numeric(xres) && length(xres)==2L && all(is.finite(xres))
if (!ok) fail("Print a numeric vector c(lower, upper).")
if (max(abs(xres - exp)) < 1e-6) pass(" Correct 95% Z-interval with  known.")
else fail("Recheck x̄, n, z*, and  = 300.")
})
```

Q6 — One-Sample CI for

Using airquality$Ozone (ignore missing), construct and print 90% CI for the true mean ozone level (ppb). Assume normality is reasonable.

Photo by Tim Witzdam on Unsplash

> **Preview**
>
> ```
> #| echo: true
> aq <- datasets::airquality
> oz <- na.omit(aq$Ozone)
> head(oz, 10)
> ```

```
#| exercise: q6_t_ci_ozone
w <- na.omit(datasets::airquality$Ozone)
```

```r
xbar <- mean(w)
s <-
n <- length(w)
tstar <-
xbar + c(-1,1)*tstar * s/sqrt(n)
```

Two-sided 90% → t^* at 0.95 with df = n−1; use s for the SD.

Solution.

```r
#| exercise: q6_t_ci_ozone
#| solution: true
w <- na.omit(datasets::airquality$Ozone)
xbar <- mean(w); s <- sd(w); n <- length(w)
tstar <- qt(0.95, df = n - 1) # two-sided 90%
xbar + c(-1,1) * tstar * s / sqrt(n)
```

```r
#| exercise: q6_t_ci_ozone
#| check: true
gradethis::grade_this({
w <- na.omit(datasets::airquality$Ozone)
xbar <- mean(w); s <- sd(w); n <- length(w)
tstar <- qt(0.95, df = n - 1)
exp <- xbar + c(-1,1)*tstar*s/sqrt(n)
r <- .result
ok <- is.numeric(r) && length(r)==2L && all(is.finite(r))
if (!ok) fail("Print a numeric vector c(lower, upper).")
if (max(abs(r - exp)) < 1e-6) pass("  Correct 90% t-interval.")
else fail("Check t* at 0.95 and your s, n.")
})
```

Q7 — One-Sample CI for a Proportion

In the UCBAdmissions dataset, Admit status for a student is "Admitted" or "Rejected". Build and print c(lower, upper) for the 95% CI on the overall admission proportion.

Photo by Matt Ragland on Unsplash

ℹ Info

For a single proportion:

$$\mathrm{CI}_{1-\alpha}: \ \hat{p} \pm z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}.$$

Rule: Ensure $n\hat{p}$ and $n(1-\hat{p})$ are not too small

ℹ Preview

```
#| echo: true
df <- as.data.frame(datasets::UCBAdmissions)  # factors Admit, Gender, Dept + Freq
head(df, 10)
aggregate(Freq ~ Admit, data = df, sum)       # totals by Admit
sum(df$Freq)                                  # total n
```

```
#| exercise: q7_p_ci_ucb
df <- as.data.frame(datasets::UCBAdmissions)
n <- sum(df$Freq)
phat <-    # proportion admitted
z <-
se <-   #compute standard error
```

phat <- sum(df$Freq[df$Admit$ == “Admitted”]) / n and z <- qnorm(0.975) for 95%.

Solution.

```
#| exercise: q7_p_ci_ucb
#| solution: true
df <- as.data.frame(datasets::UCBAdmissions)
n <- sum(df$Freq)
phat <- sum(df$Freq[df$Admit == "Admitted"]) / n
z <- qnorm(0.975)
se <- sqrt(phat*(1 - phat)/n)
phat + c(-1,1) * z * se
```

```
#| exercise: q7_p_ci_ucb
#| check: true
gradethis::grade_this({
df <- as.data.frame(datasets::UCBAdmissions)
n <- sum(df$Freq)
```

```
phat <- sum(df$Freq[df$Admit == "Admitted"]) / n
z <- qnorm(0.975)
se <- sqrt(phat*(1 - phat)/n)
exp <- phat + c(-1,1) * z * se

r <- .result
ok <- is.numeric(r) && length(r) == 2L && all(is.finite(r))
if (!ok) fail("Print a numeric vector c(lower, upper).")
if (max(abs(r - exp)) < 1e-6) pass(" Correct 95% Wald CI for the admission proportion.")
else fail("Recompute p̂, n, z*, and SE = sqrt(p̂(1-p̂)/n).")
})
```

Q8 - One-Sample CI for a Variance (²)

Using PlantGrowth$weight, print c(lower, upper) for 95% CI on ^2. (Assume normality.)

> **Info**
>
> For normal data, the variance CI uses the chi-square distribution with df=n−1:
>
> $$\left(\frac{(n-1)s^2}{\chi^2_{1-\alpha/2,\, n-1}}, \frac{(n-1)s^2}{\chi^2_{\alpha/2,\, n-1}} \right), \qquad s^2 = \text{sample variance.}$$
>
> For 95%, the cutoffs are at 0.025 and 0.975.

> **Preview**
>
> ```
> #| echo: true
> data("PlantGrowth")
> head(PlantGrowth, 10)
> ```

```
#| exercise: q8_var_ci_pg
x <- datasets::PlantGrowth$weight
n <- length(x); df <-   ; s2 <- var(x)
chi_lo <-   ( , df = df)
chi_hi <-   (0.025,  )
c(df * s2/chi_lo, df*s2/chi_hi)
```

Use s^2, df = n−1, and chi-square quantiles at 0.975 and 0.025 (note the order).

Solution.

```
#| exercise: q8_var_ci_pg
#| solution: true
x <- datasets::PlantGrowth$weight
n <- length(x); df <- n - 1; s2 <- var(x)
chi_lo <- qchisq(0.975, df = df) # upper-tail cutoff
chi_hi <- qchisq(0.025, df = df) # lower-tail cutoff
c(df * s2/chi_lo, df*s2/chi_hi)
```

```
#| exercise: q8_var_ci_pg
#| check: true
gradethis::grade_this({
x <- datasets::PlantGrowth$weight
n <- length(x); df <- n - 1; s2 <- var(x)
chi_lo <- qchisq(0.975, df = df)
chi_hi <- qchisq(0.025, df = df)
exp <- c(df*s2/chi_lo, df*s2/chi_hi)
r <- .result
ok <- is.numeric(r) && length(r)==2L && all(is.finite(r))
if (!ok) fail("Print a numeric vector c(lower, upper).")
if (max(abs(r - exp)) < 1e-6) pass("  Correct 95% CI for  ².")
else fail("Check df=n-1 and chi-square cutoffs (0.025 & 0.975).")
})
```

Q9 - One-Sample CI for a Proportion (99% Practice)

HairEyeColor is a built-in contingency table of Hair × Eye × Gender of Statistics students with counts. Treat each person as a trial and estimate the true proportion with Brown eyes. Construct a 95% Z-interval for the population proportion p. Print c(lower, upper).

> **Info**
>
> Two-sided 1− CI for a proportion:
>
> $$\mathrm{CI}_{1-\alpha}: \ \hat{p} \pm z_{1-\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}.$$
>
> For 99%, use $z^* = \phi^{-1}(0.995)$.

> **Preview**
>
> Here is a glimpse of the UCBAdmissions dataset.
>
> ```
> #| echo: true
> df <- as.data.frame(datasets::HairEyeColor)
> eye_counts <- aggregate(Freq ~ Eye, data = df, sum)
> eye_counts
> sum(eye_counts$Freq)  # total n
> ```

```
#| exercise: q10_z_ci_browneyes
#| exercise.lines: 3
#| echo: false
df <- as.data.frame(datasets::HairEyeColor)
n <- sum(df$ )
phat <- sum(df$Freq[df$  == "Brown"]) /
z <-
```

Use z <- qnorm(0.975) for 95%.

Solution.

```
#| exercise: q10_z_ci_browneyes
#| solution: true
df <- as.data.frame(datasets::HairEyeColor)
n <- sum(df$Freq)
phat <- sum(df$Freq[df$Eye == "Brown"]) / n
z <- qnorm(0.975)
phat + c(-1,1) * z * sqrt(phat*(1-phat)/n)
```

```
#| exercise: q10_z_ci_browneyes
#| check: true
gradethis::grade_this({
df <- as.data.frame(datasets::HairEyeColor)
n <- sum(df$Freq)
phat <- sum(df$Freq[df$Eye=="Brown"]) / n
z <- qnorm(0.975)
exp <- phat + c(-1,1)*z*sqrt(phat*(1-phat)/n)

x <- .result
ok <- is.numeric(x) && length(x)==2L && all(is.finite(x))
if (!ok) fail("Print a numeric vector c(lower, upper).")
if (max(abs(x - exp)) < 1e-6) pass(" Correct 95% Z-interval for the Brown-eye proportion.")
else fail("Recheck n, phat, and the Z margin for a proportion.")
})
```