Tutorial 05: Two 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 — Pooled (Equal Variances): Reviews

Using the local file recipe_reviews.csv, compute a 95% two-sided confidence interval for

$$\mu_{FiveStar} - \mu_{NotFive}$$

where the outcome is best_score and the groups are stars == 5 (FiveStar) vs stars != 5 (NotFive). Read the CSV with read.csv("recipe_reviews.csv"), build the two groups, and print the 2-element vector c(lower, upper).

Photo by Maximus Mazar on Unsplash

Info

For a pooled two-sample CI with equal variances:

$$\bar{x}_1 - \bar{x}_2 \quad \pm \quad t^*_{df=n_1+n_2-2} s_p \sqrt{\frac{1}{n1} + \frac{1}{n2}}, \quad s_p^2 = \frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}$$

Use $\alpha = 0.05 \implies t^* = t_{1-\frac{\alpha}{2},df}$
Group 1 = FiveStar (rows with stars == 5), Group 2 = NotFive (rows with stars != 5).

Preview

The "Recipe Reviews and User Feedback Dataset" is a comprehensive repository of data encompassing various aspects of recipe reviews and user interactions. It includes essential information such as the recipe name, its ranking on the top 100 recipes list, a unique recipe code, and user details like user ID, user name, and an internal user reputation score. Each review comment is uniquely identified with a comment ID and comes with additional attributes, including the creation timestamp, reply count, and the number of up-votes and down-votes received. Users' sentiment towards recipes is quantified on a 1 to 5 star rating scale, with a score of 0 denoting an absence of rating. This dataset is a valuable resource for researchers and data scientists, facilitating endeavors in sentiment analysis, user behavior analysis, recipe recommendation systems, and more. It offers a window into the dynamics of recipe reviews and user feedback within the culinary website domain.

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
df <- read.csv("recipe_reviews.csv")
head(df[, c("recipe_name","stars","best_score","thumbs_up","thumbs_down","user_reputation")
df2 <- subset(df, is.finite(best_score) & is.finite(stars))
df2$group <- ifelse(df2$stars == 5, "FiveStar", "NotFive")
set.seed(258)
grp_idx <- split(seq_len(nrow(df2)), df2$group)
idx <- unlist(lapply(grp_idx, function(i) sample(i, min(200, length(i)))))
df3 <- df2[idx, ]
ylim <- quantile(df3$best_score, c(.01,.99), na.rm = TRUE)
par(mar = c(4,4,1,1))
boxplot(best_score ~ group, data = df3, notch = FALSE, outline = FALSE, ylim = ylim,
ylab = "best_score", xlab = "Group")
```

```
#| exercise: q1_pooled_reviews
#| exercise.lines: 10
#| echo: false
df <- read.csv("recipe_reviews.csv")

x1 <- subset(df, stars == 5)$  #segregate the dataset based on best_scores
x2 <- subset(df, stars != 5)$

n1 <- sum(!is.na(x1)); n2 <- sum(!is.na(x2))
m1 <-  (x1, na.rm = TRUE) #compute mean of x1,x2
m2 <-  (x2, na.rm = TRUE)
s1 <-  (x1, na.rm = TRUE)
s2 <-  (x2, na.rm = TRUE) #compute standard deviation of x1,x2

dfree <-   #insert degrees of freedom
sp2 <-   #calculate $s^2_p$
sp <- sqrt(sp2)

tstar <-   #find critical value
se <-   #compute the standard error using formula given above
```

Confirm you're comparing the means of two groups formed by a boolean condition on stars. Use the pooled-variance standard error and the appropriate two-sided t critical value for a 95% interval. Print just the lower and upper bounds as a numeric vector.

Solution.

```
#| exercise: q1_pooled_reviews
#| solution: true
df <- read.csv("recipe_reviews.csv")

x1 <- subset(df, stars == 5)$best_score
x2 <- subset(df, stars != 5)$best_score

n1 <- sum(!is.na(x1)); n2 <- sum(!is.na(x2))
m1 <- mean(x1, na.rm = TRUE); m2 <- mean(x2, na.rm = TRUE)
s1 <- stats::sd(x1, na.rm = TRUE); s2 <- stats::sd(x2, na.rm = TRUE)

dfree <- n1 + n2 - 2
sp2 <- (((n1 - 1) * s1^2) + ((n2 - 1) * s2^2)) / dfree
sp <- sqrt(sp2)

tstar <- qt(0.975, dfree)
se <- sp * sqrt(1/n1 + 1/n2)
est <- m1 - m2

c(est - tstar*se, est + tstar*se)
```

```
#| exercise: q1_pooled_reviews
#| check: true
gradethis::grade_this({
df <- tryCatch(read.csv("recipe_reviews.csv"), error = function(e) NULL)
if (is.null(df)) fail("Couldn't read 'recipe_reviews.csv'.")

need <- c("best_score","stars")
if (!all(need %in% names(df))) fail("CSV must contain 'best_score' and 'stars' columns.")

# Build groups (work even if stars is character or numeric)

x1 <- df$best_score[df$stars == 5]
x2 <- df$best_score[df$stars != 5]

if (sum(is.finite(x1)) < 2 || sum(is.finite(x2)) < 2)
fail("Both groups need at least 2 non-missing observations.")

n1 <- sum(!is.na(x1)); n2 <- sum(!is.na(x2))
m1 <- mean(x1, na.rm = TRUE); m2 <- mean(x2, na.rm = TRUE)
s1 <- stats::sd(x1, na.rm = TRUE); s2 <- stats::sd(x2, na.rm = TRUE)
```

```
dfree <- n1 + n2 - 2
sp2 <- (((n1 - 1) * (s1^2)) + ((n2 - 1) * (s2^2))) / dfree
sp  <- sqrt(sp2)

tstar <- qt(0.975, dfree)
se <- sp * sqrt(1/n1 + 1/n2)
exp_ci <- c((m1 - m2) - tstar*se, (m1 - m2) + tstar*se)

x <- .result
if (!is.numeric(x) || length(x) != 2L || any(!is.finite(x)))
fail("Print a length-2 numeric vector: c(lower, upper).")

if (max(abs(x - exp_ci)) < 1e-6)
pass("  Correct pooled 95% CI for FiveStar - NotFive (best_score).")
else
fail("Not quite - recheck pooled s_p, df, and t* with stars==5 vs stars!=5.")
})
```

Q2 — Pooled (Equal Variances): Using t.test()

Compute a 90% two-sided confidence interval for

$$\mu_{Adelie} - \mu_{Gentoo}$$

for bill length (mm) using pooled variances. Use the local file penguins.csv and only the two species Adelie and Gentoo.

Photo by Adam Tarshis on Unsplash

Info

For equal-variance two-sample CIs, t.test(yield ~ method, var.equal = TRUE, conf.level = 0.90) uses the pooled standard error and df = $n_1 + n_2 - 2$. To get the CI for Adelie − Gentoo, set the factor level order to c("Adelie","Gentoo").

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
df <- read.csv("penguins.csv")
head(df[, c("species","bill_length_mm","island","sex")], 8)
df2 <- subset(df, species %in% c("Adelie","Gentoo") & is.finite(bill_length_mm))
set.seed(258)
grp_idx <- split(seq_len(nrow(df2)), df2$species)
idx <- unlist(lapply(grp_idx, function(i) sample(i, min(120, length(i)))))
df3 <- df2[idx, ]
par(mar = c(4,4,1,1))
boxplot(bill_length_mm ~ species, data = df3, notch = FALSE, outline = FALSE,
ylab = "bill_length_mm", xlab = "Species")
```

Info

When is known,

$$\mathrm{CI}_{1-\alpha}: \ \bar{x} \pm x_{1-\alpha/2} \frac{\sigma}{\sqrt{n}}.$$

```
#| exercise: q2_pooled_penguins
#| exercise.lines: 8
#| echo: false

df <- read.csv("penguins.csv")
keep <- df$species %in% c(" "," ") #filter based on species needed
df <- subset(df, keep & !is.na( ))


df$method <- factor(df$ , levels = c(" "," ")) # order matters for the contrast
df$yield <- df$bill_length_mm


tt <- t.test(  ~  , data = df, var.equal = TRUE, conf.level =  )
tt$conf.int #gives confidence interval
```

Use a two-group factor for the species and the pooled-variance option in t.test. Ensure you're returning just the interval bounds, not the whole test object.

Solution.

```
#| exercise: q2_pooled_penguins
#| solution: true
```

```
df <- read.csv("penguins.csv")
keep <- df$species %in% c("Adelie","Gentoo")
df <- subset(df, keep & !is.na(bill_length_mm))

df$method <- factor(df$species, levels = c("Adelie","Gentoo")) # order matters for the contra
df$yield <- df$bill_length_mm

tt <- t.test(yield ~ method, data = df, var.equal = TRUE, conf.level = 0.90)
tt$conf.int
```

```
#| exercise: q2_pooled_penguins
#| check: true
gradethis::grade_this({
df <- tryCatch(read.csv("penguins.csv"), error = function(e) NULL)
if (is.null(df)) fail("Couldn't read 'penguins.csv'.")




need <- c("species","bill_length_mm")
if (!all(need %in% names(df))) fail("CSV must have 'species' and 'bill_length_mm' columns.")




keep <- df$species %in% c("Adelie","Gentoo")
df <- subset(df, keep & !is.na(bill_length_mm))
if (nrow(df) < 4) fail("Need data for both Adelie and Gentoo with non-missing bill_length_mm




df$method <- factor(df$species, levels = c("Adelie","Gentoo"))
df$yield <- df$bill_length_mm




exp_ci <- t.test(yield ~ method, data = df, var.equal = TRUE, conf.level = 0.90)$conf.int
```

```
x <- .result
if (!is.numeric(x) || length(x) != 2L || any(!is.finite(x)))
fail("Print a 2-number numeric vector: c(lower, upper).")




if (max(abs(x - exp_ci)) < 1e-6)
pass("  Correct 90% pooled CI for Adelie - Gentoo (bill length).")
else
fail("Not quite - check species filtering, factor level order, var.equal=TRUE, and conf.level
})
```

Q3 — Welch (Unequal Variances): Using t.test()

Compute an 80% two-sided confidence interval for

$$\mu_{Chinstrap} - \mu_{Gentoo}$$

for flipper_length_mm using Welch's unequal-variance method (the default in t.test). Use the local file penguins.csv and only the species Chinstrap and Gentoo. Print only the 2-number CI vector.

Info

t.test(y ~ g, conf.level = 0.80) uses Welch by default (var.equal = FALSE). To obtain Chinstrap − Gentoo, set the factor level order to c("Chinstrap","Gentoo").

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
df <- read.csv("penguins.csv")
head(df[, c("species","bill_length_mm","island","sex")], 8)
df2 <- subset(df, species %in% c("Chinstrap","Gentoo") & is.finite(flipper_length_mm))
set.seed(258)
grp_idx <- split(seq_len(nrow(df2)), df2$species)
idx <- unlist(lapply(grp_idx, function(i) sample(i, min(120, length(i)))))
df3 <- df2[idx, ]
par(mar = c(4,4,1,1))
boxplot(flipper_length_mm ~ species, data = df3, notch = FALSE, outline = FALSE,
ylab = "flipper_length_mm", xlab = "Species")
```

```
#| exercise: q4_welch_ttest_penguins
#| exercise.lines: 8
#| echo: false
df <- read.csv("penguins.csv")

keep <- #filter based on species needed here
df <- subset(df, keep & !is.na( ))

df$method <- factor(df$ , levels = c(" "," "))
df$yield <-

tt <- t.test(  ~  , data = df, conf.level =  ) # Welch by default
```

Ensure you keep exactly two species and set their order to match the contrast. Return just the two CI bounds from the test result.

Solution.

```
#| exercise: q4_welch_ttest_penguins
#| solution: true

df <- read.csv("penguins.csv")

keep <- df$species %in% c("Chinstrap","Gentoo")
df <- subset(df, keep & !is.na(flipper_length_mm))

df$method <- factor(df$species, levels = c("Chinstrap","Gentoo"))
df$y <- df$flipper_length_mm
```

```
tt <- t.test(y ~ method, data = df, conf.level = 0.80) # Welch by default
tt$conf.int
```

```
#| exercise: q4_welch_ttest_penguins
#| check: true
gradethis::grade_this({
df <- tryCatch(read.csv("penguins.csv"), error = function(e) NULL)
if (is.null(df)) fail("Couldn't read 'penguins.csv'.")
need <- c("species","flipper_length_mm")
if (!all(need %in% names(df))) fail("CSV must have 'species' and 'flipper_length_mm' columns

keep <- df$species %in% c("Chinstrap","Gentoo")
df <- subset(df, keep & !is.na(flipper_length_mm))
if (nrow(df) < 4) fail("Need data for both Chinstrap and Gentoo with non-missing flipper_leng

df$method <- factor(df$species, levels = c("Chinstrap","Gentoo"))
df$y <- df$flipper_length_mm

exp_ci <- t.test(y ~ method, data = df, conf.level = 0.80)$conf.int

x <- .result
if (!is.numeric(x) || length(x) != 2L || any(!is.finite(x)))
fail("Print a 2-number numeric vector: c(lower, upper).")

if (max(abs(x - exp_ci)) < 1e-6)
pass("  Correct 80% Welch CI for Chinstrap - Gentoo (flipper length).")
else
fail("Not quite - check species filter, factor order, and conf.level = 0.80.")
})
```

Q4 — Two-Sample CI for a Difference of Proportions

In recipe_reviews.csv, compare the proportion of reviews where thumbs_up > thumbs_down between FiveStar (stars == 5) and NotFive (stars != 5). Compute a 95% two-sided CI for

$$p_{FiveStar} - p_{NotFive}$$

. Print c(lower, uppper).

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
df <- read.csv("recipe_reviews.csv")
head(df[, c("stars","thumbs_up","thumbs_down","best_score")], 10)
df2 <- subset(df, is.finite(thumbs_up) & is.finite(thumbs_down) & is.finite(stars))
df2$group <- ifelse(df2$stars == 5, "FiveStar", "NotFive")
df2$success <- as.integer(df2$thumbs_up > df2$thumbs_down)
set.seed(258)
grp_idx <- split(seq_len(nrow(df2)), df2$group)
idx <- unlist(lapply(grp_idx, function(i) sample(i, min(400, length(i)))))
df3 <- df2[idx, ]
p <- tapply(df3$success, df3$group, function(x) mean(x, na.rm = TRUE))
par(mar = c(4,4,1,1))
barplot(p, ylim = c(0,1), ylab = "Proportion (thumbs_up > thumbs_down)", xlab = "Group")
```

Info

CI for difference of proportions:

$$\text{CI}_{1-\alpha}: \ \hat{p_1} - \hat{p_2} \pm z_{1-\alpha/2} * \sqrt{\frac{\hat{p_1}(1-\hat{p_1})}{n_1} + \frac{\hat{p_2}(1-\hat{p_2})}{n_2}}.$$

```
#| exercise: q11_wald_2prop_reviews
#| exercise.lines: 10
#| echo: false
df <- read.csv(" ")
g1 <- subset(df, stars == 5) #segregating the dataset
g2 <- subset(df, stars != 5)

count1 <- as.integer(g1$thumbs_up > g1$thumbs_down) #helps in making 0/1 success.
#Makes all thumbs_up values 1 and thumbs_down values 0.
count2 <- as.integer(g2$thumbs_up > g2$thumbs_down)

n1 <- sum( is.finite (   ) ); n2 <- sum( is.finite (   ) ) #is.finite helps to
#filter only usable rows
p1 <- mean(   , na.rm = TRUE)
p2 <- mean(   , na.rm = TRUE)
```

```
se <-
z <-
```

Compute

$$\hat{p}$$

per group, then compute their SE and z* at 0.975.

Solution.

```
#| exercise: q11_wald_2prop_reviews
#| solution: true
df <- read.csv("recipe_reviews.csv")
g1 <- subset(df, stars == 5) #segregating the dataset
g2 <- subset(df, stars != 5)
count1 <- as.integer(g1$thumbs_up > g1$thumbs_down)
count2 <- as.integer(g2$thumbs_up > g2$thumbs_down)
n1 <- sum(is.finite(count1)); n2 <- sum(is.finite(count2))
p1 <- mean(count1, na.rm = TRUE); p2 <- mean(count2, na.rm = TRUE)
se <- sqrt(p1*(1-p1)/n1 + p2*(1-p2)/n2)
z <- qnorm(0.975);
(p1 - p2) + c(-1,1)*z*se
```

```
#| exercise: q11_wald_2prop_reviews
#| check: true
gradethis::grade_this({
df <- tryCatch(read.csv("recipe_reviews.csv"), error = function(e) NULL)
if (is.null(df)) fail("Couldn't read 'recipe_reviews.csv'.")
g1 <- subset(df, stars == 5)
g2 <- subset(df, stars != 5)
y1 <- as.integer(g1$thumbs_up > g1$thumbs_down)
y2 <- as.integer(g2$thumbs_up > g2$thumbs_down)
n1 <- sum(is.finite(y1)); n2 <- sum(is.finite(y2))
if (min(n1, n2) < 2) fail("Both groups need at least 2 finite rows.")
p1 <- mean(y1, na.rm = TRUE); p2 <- mean(y2, na.rm = TRUE)
se <- sqrt(p1*(1-p1)/n1 + p2*(1-p2)/n2)
z <- qnorm(0.975); est <- p1 - p2
exp <- est + c(-1,1)*z*se
x <- .result
ok <- is.numeric(x) && length(x)==2L && all(is.finite(x))
if (!ok) fail("Print c(lower, upper).")
if (max(abs(x - exp)) < 1e-6) pass("  Correct 95% Wald CI for p_FiveStar - p_NotFive.")
```

```
else fail("Recheck successes, n1/n2, and z*.")
})
```

Q5 — Two-Sample CI for a Difference of Proportions using prop.test

In penguins.csv, compare the male proportion between Chinstrap and Gentoo. Compute an evidence-only CI (no hypothesis interpretation needed) for $p_{Chinstrap} - p_{Gentoo}$ at 90% and print c(lower upper).

Preview

```
#| echo: true
df <- read.csv("penguins.csv")
head(df[, c("species","sex","island")], 10)
df2 <- subset(df, species %in% c("Chinstrap","Gentoo") & !is.na(sex))
set.seed(258)
grp_idx <- split(seq_len(nrow(df2)), df2$species)
idx <- unlist(lapply(grp_idx, function(i) sample(i, min(200, length(i)))))
df3 <- df2[idx, ]
p <- with(df3, tapply(sex == "male", species, function(x) mean(x, na.rm = TRUE)))
par(mar = c(4,4,1,1))
barplot(p, ylim = c(0,1), ylab = "Proportion male", xlab = "Species")
```

Info

The prop.test() function in R performs tests of proportions, allowing for the comparison of proportions across groups or against specific values. Here is an example according to this question. here's a super simple one-sample prop.test using penguins.csv. It asks: among Gentoo penguins, what fraction are male? It prints the estimate and a 95% CI. Run the code to see!

```
#| echo: true

# One-sample prop.test example: proportion of males within Gentoo

df <- read.csv("penguins.csv")

g <- subset(df, species == "Gentoo" & !is.na(sex))
x <- sum(g$sex == "male")   # number of "successes"
n <- nrow(g)                # total trials

prop.test(x = x, n = n, conf.level = 0.95, correct = FALSE)
```

```
#| exercise: q12_prop_test_penguins
#| exercise.lines: 10
#| echo: false
df <- read.csv("penguins.csv")
df <- subset(df, species %in% c(" "," ") & !is.na(sex)) #filtering based
#on the species we need and removing NA rows
x1 <- sum(df$species=="Chinstrap" & df$sex==" ")
n1 <-
x2 <- sum(df$species=="Gentoo" & df$sex==" ")
n2 <-
prop.test(x = c(   ,   ), n = c(   ,   ), conf.level =   , correct = FALSE)$conf.int
```

Filter to two species, count "male" in each, and feed (x1,n1),(x2,n2) to prop.test.

Solution.

```
#| exercise: q12_prop_test_penguins
#| solution: true
df <- read.csv("penguins.csv")
df <- subset(df, species %in% c("Chinstrap","Gentoo") & !is.na(sex))
x1 <- sum(df$species=="Chinstrap" & df$sex=="male")
n1 <- sum(df$species=="Chinstrap")
x2 <- sum(df$species=="Gentoo" & df$sex=="male")
n2 <- sum(df$species=="Gentoo")
prop.test(x = c(x1, x2), n = c(n1, n2), conf.level = 0.90, correct = FALSE)$conf.int
```

```
#| exercise: q12_prop_test_penguins
#| check: true
gradethis::grade_this({
```

```
df <- tryCatch(read.csv("penguins.csv"), error=function(e) NULL)
if (is.null(df)) fail("Couldn't read 'penguins.csv'.")
df <- subset(df, species %in% c("Chinstrap","Gentoo") & !is.na(sex))
x1 <- sum(df$species=="Chinstrap" & df$sex=="male"); n1 <- sum(df$species=="Chinstrap")
x2 <- sum(df$species=="Gentoo" & df$sex=="male");    n2 <- sum(df$species=="Gentoo")
exp <- prop.test(x=c(x1,x2), n=c(n1,n2), conf.level=0.90, correct=FALSE)$conf.int
x <- .result
ok <- is.numeric(x) && length(x)==2L && all(is.finite(x))
if (!ok) fail("Print c(lower, upper).")
if (max(abs(x - exp)) < 1e-6) pass("  Correct 90% CI for p_Chinstrap - p_Gentoo.")
else fail("Check species filter, male counts, and conf.level=0.90.")
})
```

Q6 — Two-Sample CI for a Difference of Proportions (island=Biscoe)

Using penguins.csv, compare the proportion of penguins on Biscoe island between Adelie and Gentoo. Compute a 95% two-sided CI for $p_{Adelie} - p_{Gentoo}$ and print c(lower upper).

Preview

```
#| echo: true
df <- read.csv("penguins.csv")
head(df[, c("species","sex","island")], 10)
df2 <- subset(df, species %in% c("Adelie","Gentoo") & !is.na(island))
set.seed(258)
grp_idx <- split(seq_len(nrow(df2)), df2$species)
idx <- unlist(lapply(grp_idx, function(i) sample(i, min(200, length(i)))))
df3 <- df2[idx, ]
p <- with(df3, tapply(island == "Biscoe", species, function(x) mean(x, na.rm = TRUE)))
par(mar = c(4,4,1,1))
barplot(p, ylim = c(0,1), ylab = 'Proportion on "Biscoe"', xlab = "Species")
```

```
#| exercise: q5_diff_prop_biscoe
#| exercise.lines: 6
#| echo: false
df <- read.csv("penguins.csv")
sub <- subset(df, species %in% c(" "," ") & !is.na( )) #filter based on species

x1 <- sum(sub$species==" " & sub$ ==" ") #add additional filter for island
n1 <- sum(sub$species=="Adelie")
x2 <- sum(sub$species==" " & sub$ ==" ")
```

```
n2 <- sum(sub$species=="Gentoo")

prop.test(x=c( ,  ), n=c( , ), conf.level= , correct=FALSE)$conf.int
```

Count "Biscoe" within each species to get x1, x2. Use totals for n1, n2, then prop.test(…)$conf.int. Order your counts as (Adelie, Gentoo) to get Adelie − Gentoo.

Solution.

```
#| exercise: q5_diff_prop_biscoe
#| solution: true
df <- read.csv("penguins.csv")
sub <- subset(df, species %in% c("Adelie","Gentoo") & !is.na(island))
x1 <- sum(sub$species=="Adelie" & sub$island=="Biscoe"); n1 <- sum(sub$species=="Adelie")
x2 <- sum(sub$species=="Gentoo" & sub$island=="Biscoe"); n2 <- sum(sub$species=="Gentoo")
prop.test(x=c(x1,x2), n=c(n1,n2), conf.level=0.95, correct=FALSE)$conf.int
```

```
#| exercise: q5_diff_prop_biscoe
#| check: true
gradethis::grade_this({
df <- tryCatch(read.csv("penguins.csv"), error=function(e) NULL)
if (is.null(df)) fail("Couldn't read 'penguins.csv'.")
sub <- subset(df, species %in% c("Adelie","Gentoo") & !is.na(island))
x1 <- sum(sub$species=="Adelie" & sub$island=="Biscoe"); n1 <- sum(sub$species=="Adelie")
x2 <- sum(sub$species=="Gentoo" & sub$island=="Biscoe"); n2 <- sum(sub$species=="Gentoo")
exp <- prop.test(x=c(x1,x2), n=c(n1,n2), conf.level=0.95, correct=FALSE)$conf.int
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 p_Adelie - p_Gentoo (Biscoe).")
else fail("Recheck counts, totals, order (Adelie first), and conf.level=0.95.")
})
```

Q7 — Two-Sample CI for a Ratio of Variances (manual F-based)

Using recipe_reviews.csv, compare the variance of best_score for FiveStar vs NotFive. Build a 95% CI for $\sigma^2_{FiveStar}/\sigma^2_{NotFive}$ and print c(lower, upper).

Preview

Here is a glimpse of the dataset:

```
#| echo: true
df <- read.csv("recipe_reviews.csv")
head(df[, c("stars","best_score")], 10)
df2 <- subset(df, is.finite(best_score) & is.finite(stars))
df2$group <- ifelse(df2$stars == 5, "FiveStar", "NotFive")
set.seed(258)
grp_idx <- split(seq_len(nrow(df2)), df2$group)
idx <- unlist(lapply(grp_idx, function(i) sample(i, min(200, length(i)))))
df3 <- df2[idx, ]
ylim <- quantile(df3$best_score, c(.01, .99), na.rm = TRUE)
par(mar = c(4,4,1,1))
boxplot(best_score ~ group, data = df3, notch = FALSE, outline = FALSE, ylim = ylim,
ylab = "best_score", xlab = "Group")
```

Info

Taking group 1 as FiveStar here, with s_1^2 and s_2^2 as sample variances, here is the CI:

$$\mathrm{CI}_{1-\alpha}\left(\frac{\sigma_1^2}{\sigma_2^2}\right) = \left(\frac{s_1^2/s_2^2}{F_{1-\alpha/2,\,df_1,\,df_2}},\ \frac{s_1^2/s_2^2}{F_{\alpha/2,\,df_1,\,df_2}}\right), \qquad df_1 = n_1 - 1,\ df_2 = n_2 - 1.$$

```
#| exercise: q7_ratio_var_reviews
#| exercise.lines: 12
#| echo: false
df <- read.csv("recipe_reviews.csv")
x1 <- subset(df, stars == 5)$
x2 <- subset(df, stars != 5)$
x1 <- x1[is.finite(x1)] #only keep usable rows
x2 <- x2[is.finite(x2)]
n1 <- length( ); n2 <- length( )
s1 <- stats:: (x1); s2 <- stats:: (x2) #computing variance of both vectors x1,x2
df1 <-  ; df2 <-
ratio <- s1/s2
F_lo <-
F_hi <-
```

Use FiveStar as numerator. Compute s_1^2 and s_2^2 and divide by the F cutoffs (0.975 and 0.025).

Solution.

```
#| exercise: q7_ratio_var_reviews
#| solution: true
df <- read.csv("recipe_reviews.csv")
x1 <- subset(df, stars == 5)$best_score
x2 <- subset(df, stars != 5)$best_score
x1 <- x1[is.finite(x1)]
x2 <- x2[is.finite(x2)]
n1 <- length(x1)
n2 <- length(x2)
s1 <- stats::var(x1)
s2 <- stats::var(x2)
df1 <- n1 - 1
df2 <- n2 - 1
ratio <- s1/s2
F_lo <- qf(0.975, df1, df2)
F_hi <- qf(0.025, df1, df2)
c(ratio/F_lo, ratio/F_hi)
```

```
#| exercise: q7_ratio_var_reviews
#| check: true
gradethis::grade_this({
df <- tryCatch(read.csv("recipe_reviews.csv"), error=function(e) NULL)
if (is.null(df)) fail("Couldn't read 'recipe_reviews.csv'.")
x1 <- subset(df, stars == 5)$best_score
x2 <- subset(df, stars != 5)$best_score
x1 <- x1[is.finite(x1)]; x2 <- x2[is.finite(x2)]
if (min(length(x1), length(x2)) < 3) fail("Each group needs at least 3 finite rows.")
s1 <- stats::var(x1); s2 <- stats::var(x2)
df1 <- length(x1)-1; df2 <- length(x2)-1
ratio <- s1/s2
F_lo <- qf(0.975, df1, df2); F_hi <- qf(0.025, df1, df2)
exp <- c(ratio/F_lo, ratio/F_hi)
x <- .result
ok <- is.numeric(x) && length(x)==2L && all(is.finite(x))
if (!ok) fail("Print c(lower, upper).")
if (max(abs(x - exp)) < 1e-6) pass("  Correct 95% CI for  ²_FiveStar /  ²_NotFive.")
else fail("Check df and F quantiles (0.975, 0.025).")
})
```

Q8 - Two-Sample CI for a Ratio of Variances using var.test

In penguins.csv, compare variance of bill_length_mm between Adelie (group 1) and Gentoo (group 2). Compute an 80% CI for $\sigma^2_{Adelie}/\sigma^2_{Gentoo}$ using var.test and print the two bounds.

Info

Here is an example to demonstrate how var.test works. We are taking the recipe reviews dataset to showcase the usage. Run the code to view results.

```
#thumbs_up variance: FiveStar vs NotFive (95% CI)
df <- read.csv("recipe_reviews.csv")

x1 <- df$thumbs_up[df$stars == 5]
x2 <- df$thumbs_up[df$stars != 5]
x1 <- x1[is.finite(x1)]; x2 <- x2[is.finite(x2)]

vt <- var.test(x1, x2, conf.level = 0.95)

vt$estimate   # s_FiveStar^2 / s_NotFive^2
vt$conf.int   # 95% CI for  _FiveStar^2 /  _NotFive^2
```

Preview

```
#| echo: true
df <- read.csv("penguins.csv")
head(df[, c("species","bill_length_mm")], 10)
df2 <- subset(df, species %in% c("Adelie","Gentoo") & is.finite(bill_length_mm))
set.seed(258)
grp_idx <- split(seq_len(nrow(df2)), df2$species)
idx <- unlist(lapply(grp_idx, function(i) sample(i, min(120, length(i)))))
df3 <- df2[idx, ]
par(mar = c(4,4,1,1))
boxplot(bill_length_mm ~ species, data = df3, notch = FALSE, outline = FALSE,
ylab = "bill_length_mm", xlab = "Species")
```

```
#| exercise: q8_var_test_penguins
#| exercise.lines: 8
#| echo: false
df <- read.csv("penguins.csv")
df <- subset(df, species %in% c( " " , " " ) & !is.na( ))
x <- df$ [df$ =="Adelie"]
```

```
y <- df$ [df$ =="Gentoo"]
var.test( )$conf.int
```

Filter to Adelie and Gentoo, remove NAs, then feed the two numeric vectors to var.test.

Solution.

```
#| exercise: q8_var_test_penguins
#| solution: true
df <- read.csv("penguins.csv")
df <- subset(df, species %in% c("Adelie","Gentoo") & !is.na(bill_length_mm))
x <- df$bill_length_mm[df$species=="Adelie"]
y <- df$bill_length_mm[df$species=="Gentoo"]
var.test(x, y, conf.level = 0.80)$conf.int
```

```
#| exercise: q8_var_test_penguins
#| check: true
gradethis::grade_this({
df <- tryCatch(read.csv("penguins.csv"), error=function(e) NULL)
if (is.null(df)) fail("Couldn't read 'penguins.csv'.")
df <- subset(df, species %in% c("Adelie","Gentoo") & !is.na(bill_length_mm))
x <- df$bill_length_mm[df$species=="Adelie"]
y <- df$bill_length_mm[df$species=="Gentoo"]
exp <- var.test(x, y, conf.level = 0.80)$conf.int
z <- .result
ok <- is.numeric(z) && length(z)==2L && all(is.finite(z))
if (!ok) fail("Print c(lower, upper).")
if (max(abs(z - exp)) < 1e-6) pass("  Correct 80% CI for  ²_Adelie /  ²_Gentoo.")
else fail("Confirm vector order (Adelie first) and conf.level=0.80.")
})
```