Tutorial 07: Two Sample Hypothesis Tests

```
#| 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 two-sample t-test (Audience score: Avengers vs Spider-Man)

For this question, we use marvel.csv. Compare the audience % score between Avengers and Spider-Man movies.

We test

$$H_0 : \mu_{Avengers} - \mu_{Spider-Man} = 0$$

vs

$$H_1 : \mu_{Avengers} - \mu_{Spider-Man} \neq 0$$

assuming equal population variances and using a pooled two-sample t-test. Compute the p-value manually from the sample statistics and the t distribution. Your final output should be a single numeric p-value.

Photo by Erik Mclean on Unsplash

Info

In a pooled two-sample t-test for two groups with sample sizes n_1, n_2, sample means $\bar{x}_1, \bar{x}_2$, and sample standard deviations s_1, s_2, we first compute the pooled variance

$$s_p^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}.$$

The test statistic for testing $H_0 : \mu_1 - \mu_2 = 0$ is

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{s_p^2 \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}}.$$

Under the null hypothesis, this statistic follows a t distribution with

$$\mathrm{df} = n_1 + n_2 - 2$$

degrees of freedom. For a two-sided test, the p-value is

$$p\text{-value} = 2\,P(T_{\mathrm{df}} \geq |t|).$$

Preview

Run this code chunk to get a glimpse of the dataset and visualize the audience scores.

```
#| echo: true
df <- read.csv("marvel.csv", check.names = FALSE)
sub <- subset(df, category %in% c("Avengers","Spider-Man"))
sub$aud <- suppressWarnings(as.numeric(sub[["audience % score"]]))
sub <- subset(sub, is.finite(aud))

head(sub[, c("film","category","aud")], 10)

par(mfrow = c(1, 2))
hist(sub$aud,
main = "Audience % score (Avengers + Spider-Man)",
xlab = "Audience % score")
boxplot(aud ~ category, data = sub,
main = "Audience % score by category",
xlab = "Category", ylab = "Audience % score")
par(mfrow = c(1, 1))
```

```
#| exercise: q1_pooled_audience
#| exercise.lines: 10
#| echo: false

df <- read.csv("marvel.csv", check.names = FALSE)

df <- subset(df, category %in% c("Avengers","Spider-Man"))
df$aud <- suppressWarnings(as.numeric(sub("%", "", df[["audience % score"]], fixed = TRUE)))
df <- subset(df, is.finite(aud))

g1 <- df$aud[df$category == "Avengers"]
g2 <- df$aud[df$category == "Spider-Man"]

n1 <- length(g1); n2 <- length(g2)
m1 <- mean(g1);   m2 <- mean(g2)
s1 <- sd(g1);     s2 <- sd(g2)

sp2  <- _
tval <- _
df_t <- _
```

```
pval <- _
pval
```

Separate the audience scores for each franchise, then use their sample sizes, means, and standard deviations to construct the pooled variance, the standardized test statistic, and the two-sided p-value from the appropriate t distribution.

Solution.

```
#| exercise: q1_pooled_audience
#| solution: true
df <- read.csv("marvel.csv", check.names = FALSE)

df <- subset(df, category %in% c("Avengers","Spider-Man"))
df$aud <- suppressWarnings(as.numeric(sub("%", "", df[["audience % score"]], fixed = TRUE)))
df <- subset(df, is.finite(aud))

g1 <- df$aud[df$category == "Avengers"]
g2 <- df$aud[df$category == "Spider-Man"]

n1 <- length(g1); n2 <- length(g2)
m1 <- mean(g1);   m2 <- mean(g2)
s1 <- sd(g1);     s2 <- sd(g2)

sp2  <- ((n1-1)*s1^2 + (n2-1)*s2^2)/(n1+n2-2)
tval <- (m1-m2)/sqrt(sp2*(1/n1 + 1/n2))
df_t <- n1+n2-2
pval <- 2*pt(-abs(tval), df_t)
pval
```

```
#| exercise: q1_pooled_audience
#| check: true
gradethis::grade_this({
  df <- read.csv("marvel.csv", check.names = FALSE)

  # robust column lookup (works whether names were preserved or not)
  aud_col <- if ("audience % score" %in% names(df)) {
    "audience % score"
  } else {
    grep("^audience", names(df), value = TRUE)[1]
  }
  df$aud <- suppressWarnings(as.numeric(sub("%", "", df[[aud_col]], fixed = TRUE)))

  df <- subset(df, category %in% c("Avengers","Spider-Man") & is.finite(aud))
  g1 <- df$aud[df$category == "Avengers"]
  g2 <- df$aud[df$category == "Spider-Man"]

  n1 <- length(g1); n2 <- length(g2)
  m1 <- mean(g1);   m2 <- mean(g2)
  s1 <- sd(g1);     s2 <- sd(g2)

  sp2  <- ((n1 - 1) * s1^2 + (n2 - 1) * s2^2) / (n1 + n2 - 2)
  tval <- (m1 - m2) / sqrt(sp2 * (1 / n1 + 1 / n2))
  df_t <- n1 + n2 - 2
  exp  <- 2 * pt(-abs(tval), df_t)

  x <- .result
  ok <- is.numeric(x) && length(x) == 1L && is.finite(x)

  if (!ok) {
    fail("Return a single numeric p-value.")
  } else if (abs(x - exp) < 1e-10) {
    pass("Pooled two-sample p-value computed correctly.")
  } else {
    fail("Your p-value does not match the pooled two-sample calculation.")
  }
})
```

Q2 — Welch two-sample t-test (Opening weekend: Avengers vs Spider-Man)

Now use marvel.csv to compare the opening weekend gross (in $m) between Avengers and Spider-Man movies.

We test

$$H_0 : \mu_{Avengers} - \mu_{Spider-Man} = 0$$

vs

$$H_1 : \mu_{Avengers} - \mu_{Spider-Man} \neq 0$$

without assuming equal variances. Use the Welch two-sample t-test formula and compute the p-value manually from the t distribution. Do not call the built-in t.test function in this question. Return a single numeric p-value.

Photo by Erik Mclean on Unsplash

Info

In the Welch two-sample t-test, we keep the sample variances separate. For two groups with sample sizes n_1, n_2, means $\bar{x}_1, \bar{x}_2$, and standard deviations s_1, s_2, the test statistic is

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}.$$

The approximate Welch degrees of freedom are

$$\mathrm{df}_W = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(\frac{s_1^2}{n_1}\right)^2}{n_1 - 1} + \frac{\left(\frac{s_2^2}{n_2}\right)^2}{n_2 - 1}}.$$

For a two-sided test of $H_0 : \mu_1 - \mu_2 = 0$, the p-value is

$$p\text{-value} = 2\,P(T_{\mathrm{df}_W} \geq |t|).$$

Preview

```r
#| echo: true
df <- read.csv("marvel.csv", check.names=FALSE)
sub <- subset(df, category %in% c("Avengers","Spider-Man"))
sub$open <- sub[["opening weekend ($m)"]]
sub <- subset(sub, is.finite(open))

head(sub[, c("film","category","open")], 10)

par(mfrow = c(1, 2))
hist(sub$open,
main = "Opening weekend gross (Avengers + Spider-Man)",
xlab = "Opening weekend ($m)")
boxplot(open ~ category, data = sub,
main = "Opening weekend by category",
xlab = "Category", ylab = "Opening weekend ($m)")
par(mfrow = c(1, 1))
```

```r
#| exercise: q2_mcu_welch_open
#| exercise.lines: 8
#| echo: false
df <- read.csv("marvel.csv", check.names=FALSE)
df <- subset(df, category %in% c(___, ___))
df$open <- df[["opening weekend ($m)"]]
df <- subset(df, is.finite(open))

g1 <- df$open[df$category == ___]
g2 <- df$open[df$category == ___]

n1 <- ___
n2 <- ___

m1 <- ___
m2 <- ___

s1 <- ___
s2 <- ___

se_diff <- ___
tval <- ___

df_w <- ___
pval <- ___
pval
```

Separate the opening weekend values by franchise, compute the sample means and variances, standardize the difference using the Welch standard error, then use the Welch degrees-of-freedom formula and a two-sided t distribution to obtain the p-value.

Solution.

```
#| exercise: q2_mcu_welch_open
#| solution: true
df <- read.csv("marvel.csv", check.names = FALSE)
df <- subset(df, category %in% c("Avengers","Spider-Man"))
df$open <- df[["opening weekend ($m)"]]
df <- df[is.finite(df$open), ]

g1 <- df$open[df$category == "Avengers"]
g2 <- df$open[df$category == "Spider-Man"]
```

```
n1 <- length(g1); n2 <- length(g2)
m1 <- mean(g1);   m2 <- mean(g2)
s1 <- sd(g1);     s2 <- sd(g2)

se_diff <- sqrt(s1^2 / n1 + s2^2 / n2)
tval <- (m1 - m2) / se_diff

df_w <- (s1^2 / n1 + s2^2 / n2)^2 /
  ((s1^2 / n1)^2 / (n1 - 1) + (s2^2 / n2)^2 / (n2 - 1))

pval <- 2 * pt(-abs(tval), df_w)
pval
```

```
#| exercise: q2_mcu_welch_open
#| check: true
gradethis::grade_this({
  df <- read.csv("marvel.csv", check.names = FALSE)
  df$open <- df[["opening weekend ($m)"]]
  df <- df[df$category %in% c("Avengers","Spider-Man") & is.finite(df$open), ]

  g1 <- df$open[df$category == "Avengers"]
  g2 <- df$open[df$category == "Spider-Man"]

  n1 <- length(g1); n2 <- length(g2)
  m1 <- mean(g1);   m2 <- mean(g2)
  s1 <- sd(g1);     s2 <- sd(g2)

  se_diff <- sqrt(s1^2 / n1 + s2^2 / n2)
  tval <- (m1 - m2) / se_diff

  df_w <- (s1^2 / n1 + s2^2 / n2)^2 /
    ((s1^2 / n1)^2 / (n1 - 1) + (s2^2 / n2)^2 / (n2 - 1))

  exp <- 2 * pt(-abs(tval), df_w)

  x <- .result
  ok <- is.numeric(x) && length(x) == 1L && is.finite(x)

  if (!ok) fail("Return a single numeric p-value.")
  else if (abs(x - exp) < 1e-10) pass("Welch two-sample p-value computed correctly.")
  else fail("Your p-value does not match the Welch two-sample calculation.")
})
```

Q3 — Pooled two-sample t-test (Critics score: Captain America vs Iron Man)

Now compare critics % score between Captain America and Iron Man movies in marvel.csv.

We test

$$H_0 : \mu_{Cap} - \mu_{IronMan} = 0$$

vs

$$H_1 : \mu_{Cap} - \mu_{IronMan} \neq 0$$

and we treat the population variances as equal, using a pooled two-sample t-test. Your answer should be a single numeric p-value.

Photo by Mateusz Wacławek on Unsplash

Info

This is another two-sample mean comparison where we assume the two sets of critics scores arise from populations with the same variance. Under that assumption, we combine the information from both groups into one pooled estimate of the variance to build the test statistic.

Preview

```
#| echo: true
df <- read.csv("marvel.csv", check.names=FALSE)
sub <- subset(df, category %in% c("Captain America","Iron Man"))
sub$crit <- suppressWarnings(as.numeric(sub[["critics % score"]]))
sub <- subset(sub, is.finite(crit))

head(sub[, c("film","category","crit")], 10)

par(mfrow = c(1, 2))
hist(sub$crit,
main = "Critics % score (Cap + Iron Man)",
xlab = "Critics % score")
boxplot(crit ~ category, data = sub,
main = "Critics % score by category",
xlab = "Category", ylab = "Critics % score")
par(mfrow = c(1, 1))
```

```
#| exercise: q3_mcu_pooled_critics
#| exercise.lines: 8
#| echo: false
df <- read.csv("marvel.csv", check.names = FALSE)
df <- subset(df, category %in% c("___","___"))
df$crit <- suppressWarnings(as.numeric(sub("%","", df[["critics % score"]], fixed=TRUE)))


df <- subset(df, is.finite(crit))
df$grp <- ___(df$category, levels = c("Captain America","Iron Man"))
t.test(crit ~ grp, data = df, var.equal = ___)$___
```

Limit the data to the two franchises named in the question, use the critics scores as the response, and form a two-level grouping factor. Then carry out a pooled two-sample test and extract the single p-value it produces.

Solution.

```
#| exercise: q3_mcu_pooled_critics
#| solution: true
df <- read.csv("marvel.csv", check.names = FALSE)
df <- subset(df, category %in% c("Captain America","Iron Man"))

df$crit <- suppressWarnings(as.numeric(sub("%","", df[["critics % score"]], fixed = TRUE)))
df <- subset(df, is.finite(crit))

df$grp <- factor(df$category, levels = c("Captain America","Iron Man"))
t.test(crit ~ grp, data = df, var.equal = TRUE)$p.value
```

```
#| exercise: q3_mcu_pooled_critics
#| check: true
gradethis::grade_this({
  df <- read.csv("marvel.csv", check.names = FALSE)
  df$crit <- suppressWarnings(as.numeric(sub("%","", df[["critics % score"]], fixed = TRUE))
  df <- subset(df, category %in% c("Captain America","Iron Man") & is.finite(crit))
  df$grp <- factor(df$category, levels = c("Captain America","Iron Man"))
  exp <- t.test(crit ~ grp, data = df, var.equal = TRUE)$p.value

  x <- .result
  ok <- is.numeric(x) && length(x) == 1L && is.finite(x)
  if (!ok) fail("Your answer should be a single numeric p-value.")
  else if (abs(x - exp) < 1e-12) pass("Correct pooled two-sample p-value for the critics sco
  else fail("Revisit your subset, grouping, and the assumption of equal variances.")
})
```

Q4 — Welch two-sample t-test (Domestic gross: Black Panther vs Thor)

Finally, compare the domestic gross (in $m) between Black Panther movies and Thor movies in marvel.csv.

We test

$$H_0 : \mu_{BlackPanther} - \mu_{Thor} = 0$$

vs

$$H_1 : \mu_{BlackPanther} - \mu_{Thor} \neq 0$$

Here we do not assume the variances are equal and instead use a Welch two-sample t-test. Your answer should be a single numeric p-value.

Photo by Mateusz Wacławek on Unsplash

> ℹ Info
>
> This is another two-sample test where the two groups may have quite different variability. The Welch approach adjusts both the test statistic and degrees of freedom to account for unequal variances.

ℹ Preview

```
#| echo: true
df <- read.csv("marvel.csv", check.names=FALSE)
sub <- subset(df, category %in% c("Black Panther","Thor"))
sub$dom <- sub[["domestic gross ($m)"]]
sub <- subset(sub, is.finite(dom))

head(sub[, c("film","category","dom")], 10)

par(mfrow = c(1, 2))
hist(sub$dom,
main = "Domestic gross (Black Panther + Thor)",
xlab = "Domestic gross ($m)")
boxplot(dom ~ category, data = sub,
main = "Domestic gross by category",
xlab = "Category", ylab = "Domestic gross ($m)")
par(mfrow = c(1, 1))
```

```
#| exercise: q4_mcu_welch_domestic
#| exercise.lines: 8
#| echo: false
df <- read.csv("marvel.csv", check.names=FALSE)
df <- subset(df, category %in% c("___","___"))
df$dom <- suppressWarnings(as.numeric(df[["domestic gross ($m)"]]))
df <- df[is.finite(df$dom), ]
df$grp <- ___(df$category, ___ = c("___","___"))
p-val <- t.test(dom ~ grp, data = df, var.equal = ___)$___
unname(as.numeric(pval))
```

Work only with the two franchises specified, treat domestic gross as the numeric response, and compare the two groups while allowing their variances to differ. Extract just the p-value from the resulting test object.

Solution.

```
#| exercise: q4_mcu_welch_domestic
#| solution: true
df <- read.csv("marvel.csv", check.names = FALSE)
df <- subset(df, category %in% c("Black Panther","Thor"))
```

```
df$dom <- suppressWarnings(as.numeric(df[["domestic gross ($m)"]]))
df <- df[is.finite(df$dom), ]

df$grp <- factor(df$category, levels = c("Black Panther","Thor"))

pval <- t.test(dom ~ grp, data = df, var.equal = FALSE)$p.value
unname(as.numeric(pval))
```

```
#| exercise: q4_mcu_welch_domestic
#| check: true
gradethis::grade_this({
  df <- read.csv("marvel.csv", check.names = FALSE)
  df <- subset(df, category %in% c("Black Panther","Thor"))

  df$dom <- suppressWarnings(as.numeric(df[["domestic gross ($m)"]]))
  df <- df[is.finite(df$dom), ]

  df$grp <- factor(df$category, levels = c("Black Panther","Thor"))

  exp <- t.test(dom ~ grp, data = df, var.equal = FALSE)$p.value
  exp <- unname(as.numeric(exp))

  x <- .result
  ok <- is.numeric(x) && length(x) == 1L && is.finite(x)

  if (!ok) {
    fail("Your answer should be a single numeric p-value.")
  } else if (abs(unname(as.numeric(x)) - exp) < 1e-12) {
    pass("Correct Welch two-sample p-value for the domestic grosses.")
  } else {
    fail("Check that you have the right subset, response variable, and that you are not impos
  }
})
```

Q5 — Welch two-sample t-test using built-in method (HP: Legendary vs non-Legendary)

Use `Pokemon.csv` to test whether the **proportion of Legendary Pokémon** is the same in **Generation 1** and **Generation 2**.

Let “success” be **Legendary = TRUE**. Define:

- Group 1: Generation 1
- Group 2: Generation 2

We test

$$H_0: \; p_1 - p_2 = 0$$

versus

$$H_1: \; p_1 - p_2 \neq 0,$$

where (p__1) and (p__2) are the true proportions of Legendary Pokémon in Generation 1 and 2.

Compute the p-value **manually** using the normal approximation formulas. Your final output must be a single numeric p-value.

Photo by Thimo Pedersen on Unsplash

ℹ Info

Let (x__1, x__2) be the number of successes in groups 1 and 2, and (n__1, n__2) the corresponding sample sizes. The sample proportions are

$$\hat{p}_1 = \frac{x_1}{n_1}, \qquad \hat{p}_2 = \frac{x_2}{n_2}.$$

Under (H__0: p__1 = p__2), the **pooled proportion** is

$$\hat{p} = \frac{x_1 + x_2}{n_1 + n_2}.$$

The **standard error** for the difference in proportions under the null is

$$\mathrm{SE}(\hat{p}_1 - \hat{p}_2) = \sqrt{\hat{p}(1-\hat{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}.$$

The **test statistic** ::: {.callout-note title="Preview"}

```
#| echo: true
df <- read.csv("Pokemon.csv")
df$Legendary <- df$Legendary == "True"

sub <- df[, c("Name", "Legendary", "HP")]

head(sub, 10)

par(mfrow = c(1, 2))
hist(sub$HP,
main = "HP (all Pokémon)",
xlab = "HP")
boxplot(HP ~ Legendary, data = sub,
main = "HP by Legendary status",
xlab = "Legendary", ylab = "HP")
par(mfrow = c(1, 1))
```

```
#| exercise: q6_poke_ttest
#| exercise.lines: 10
#| echo: false
df <- read.csv("Pokemon.csv")
df$Legendary <- df$Legendary == "True"
df$grp <- factor(df$Legendary, levels = c(___, ___))
res <- t.test(___ ~ ___, data = df, var.equal = ___)
res$___
```

Form a two-level factor for Legendary status, use HP as the response in a two-sample procedure that allows unequal variances, and then pull out the single p-value component from the test result.

Solution.

```
#| exercise: q6_poke_ttest
#| solution: true
#| echo: false
df <- read.csv("Pokemon.csv")
df$Legendary <- df$Legendary == "True"
df$grp <- factor(df$Legendary, levels = c(FALSE, TRUE))
res <- t.test(HP ~ grp, data = df, var.equal = FALSE)
res$p.value
```

```
#| exercise: q6_poke_ttest
#| check: true
gradethis::grade_this({
df <- read.csv("Pokemon.csv")
df$Legendary <- df$Legendary == "True"
df$grp <- factor(df$Legendary, levels = c(FALSE, TRUE))
exp <- t.test(HP ~ grp, data = df, var.equal = FALSE)$p.value

x <- .result
ok <- is.numeric(x) && length(x) == 1L && is.finite(x)

if (!ok) {
fail("Return a single numeric p-value.")
} else if (abs(x - exp) < 1e-12) {
pass("Correct Welch two-sample p-value for HP using the built-in test.")
} else {
fail("Check your response variable, grouping factor, and unequal-variance setting.")
}
})
```

Q6 — Two-sample test for difference of proportions using built-in method

Again compare the **proportion of Legendary Pokémon** in **Generation 1** and **Generation 2**, but now using a **built-in** two-sample proportion test.

Let “success” be **Legendary = TRUE**.

We test

$$H_0: \; p_1 - p_2 = 0$$

versus

$$H_1: \; p_1 - p_2 \neq 0,$$

where (p__1) and (p__2) are the true proportions of Legendary Pokémon in Generation 1 and Generation 2.
Use a built-in two-sample test for proportions and return a **single numeric p-value**.

Photo by Thimo Pedersen on Unsplash

Info

A built-in two-sample test for proportions needs: - the vector of successes in each group, ((x_1, x_2))
- the vector of sample sizes, ((n_1, n_2))
It then constructs the appropriate test statistic and p-value under the null hypothesis that the two population proportions are equal.

Preview

```
#| echo: true
df <- read.csv("Pokemon.csv")
df$Legendary <- df$Legendary == "True"

sub <- subset(df, Generation %in% c(1, 2))
tab <- table(sub$Generation, sub$Legendary)
tab

prop.table(tab, 1)
```

```
#| exercise: q8_poke_prop_test
#| exercise.lines: 12
#| echo: false
df <- read.csv("Pokemon.csv")
df$Legendary <- df$Legendary == "True"
df <- subset(df, Generation %in% c(___, ___))

x1 <- sum(df$Legendary[df$Generation == ___])
x2 <- sum(df$Legendary[df$Generation == ___])

n1 <- sum(df$Generation == ___)
n2 <- sum(df$Generation == ___)

res <- suppressWarnings(prop.test(c(___, ___), c(___, ___), correct = TRUE))
res$___
```

Count the Legendary Pokémon and total Pokémon in Generations 1 and 2, pass those counts and sample sizes to a two-sample proportion test, and extract the p-value from the result.

Solution.

```
#| exercise: q8_poke_prop_test
#| solution: true
#| echo: false
df <- read.csv("Pokemon.csv")
df$Legendary <- df$Legendary == "True"
df <- subset(df, Generation %in% c(1, 2))

x1 <- sum(df$Legendary[df$Generation == 1])
x2 <- sum(df$Legendary[df$Generation == 2])

n1 <- sum(df$Generation == 1)
n2 <- sum(df$Generation == 2)

res <- suppressWarnings(prop.test(c(x1, x2), c(n1, n2), correct = TRUE))
res$p.value
```

```
#| exercise: q8_poke_prop_test
#| check: true
gradethis::grade_this({
df <- read.csv("Pokemon.csv")
df$Legendary <- df$Legendary == "True"
df <- subset(df, Generation %in% c(1, 2))
x1 <- sum(df$Legendary[df$Generation == 1])
x2 <- sum(df$Legendary[df$Generation == 2])
n1 <- sum(df$Generation == 1)
n2 <- sum(df$Generation == 2)
exp <- prop.test(c(x1, x2), c(n1, n2), correct = TRUE)$p.value

x <- .result
ok <- is.numeric(x) && length(x) == 1L && is.finite(x)

if (!ok) {
fail("Return a single numeric p-value.")
} else if (abs(x - exp) < 1e-12) {
pass("Correct two-proportion test p-value using the built-in method.")
} else {
fail("Check your counts, sample sizes, and how you called the built-in proportion test.")
}
})
```

Q7 — F-test for ratio of variances using built-in method (Attack: Fire vs Water)

Use `Pokemon.csv` to compare the **variance of Attack** between **Fire-type** and **Water-type** Pokémon (using the primary type `Type 1`).

Let

- Group 1: Type 1 = Fire
- Group 2: Type 1 = Water

We test

$$H_0 : \ \sigma^2_{\text{Fire}} = \sigma^2_{\text{Water}}$$

versus

$$H_1 : \ \sigma^2_{\text{Fire}} \neq \sigma^2_{\text{Water}}.$$

Use an appropriate built-in two-sample test for equality of variances and return a **single numeric p-value**.

Photo by Life Time Values on Unsplash

ℹ Info

A built-in two-sample variance test compares the sample variances of two groups using an F statistic and the F distribution. It uses the group sample sizes to set the numerator and denominator degrees of freedom, and then computes a p-value for testing whether the two population variances are equal.

Preview

```
#| echo: true
df <- read.csv("Pokemon.csv")
sub <- subset(df, Type.1 %in% c("Fire", "Water"))

head(sub[, c("Name", "Type.1", "Attack")], 10)

boxplot(Attack ~ Type.1, data = sub,
        main = "Attack by Type (Fire vs Water)",
        xlab = "Type 1", ylab = "Attack")

tapply(sub$Attack, sub$Type.1, var)
```

```
#| exercise: q10_poke_var_test
#| exercise.lines: 10
#| echo: false
df <- read.csv("Pokemon.csv")
df <- subset(df, Type.1 %in% c(___, ___))
df$grp <- factor(df$Type.1, levels = c(___, ___))

res <- var.test(___ ~ ___, data = df)
res$___
```

Restrict the data to Fire and Water types, build a two-level factor for the type, apply a built-in two-sample variance test using Attack as the response, and extract the single p-value from the test output.

Solution.

```
#| exercise: q10_poke_var_test
#| solution: true
#| echo: false
df <- read.csv("Pokemon.csv")
df <- subset(df, Type.1 %in% c("Fire", "Water"))
df$grp <- factor(df$Type.1, levels = c("Fire", "Water"))

res <- var.test(Attack ~ grp, data = df)
res$p.value
```

```
#| exercise: q10_poke_var_test
#| check: true
gradethis::grade_this({
df <- read.csv("Pokemon.csv")
df <- subset(df, Type.1 %in% c("Fire", "Water"))
df$grp <- factor(df$Type.1, levels = c("Fire", "Water"))
exp <- var.test(Attack ~ grp, data = df)$p.value

x <- .result
ok <- is.numeric(x) && length(x) == 1L && is.finite(x)

if (!ok) {
fail("Return a single numeric p-value.")
} else if (abs(x - exp) < 1e-12) {
pass("Correct variance-test p-value using the built-in F-test.")
} else {
fail("Check that you are using Attack as the response, Fire and Water as the groups, and ext
}
})
```