Tutorial 08: ANOVA

```
#| 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 — One-way ANOVA (manual): Happiness score by region

We start this tutorial with the city lifestyle dataset, that gives info about In the city lifestyle dataset, we want to test whether the **mean happiness score** differs across four regions: Europe, Asia, North America, and Africa.

We test:

$$H_0 : \mu_{\text{Europe}} = \mu_{\text{Asia}} = \mu_{\text{NorthAm}} = \mu_{\text{Africa}}$$

vs

$$H_1 : \text{ at least one regional mean is different.}$$

Use a **one-way ANOVA** on happiness score, computed *manually* from sums of squares.

Your task:

- Subset to those four regions.
- Remove rows with missing happiness scores.
- Compute SSR, SSE, $df1$, $df2$, MSR, MSE, F, and the p-value

$$F = \frac{\text{MSR}}{\text{MSE}}, \qquad p = P(F_{df1,df2} \geq F).$$

Return a named numeric vector: `c(F = Fstat, df1 = df1, df2 = df2, p_value = pval)`.

Photo by Arno Retief on Unsplash

Info

For a one-way ANOVA with response y_{ij} in group (i) (of size n_i), group means $bary_i$, and overall mean $bary$:

Between–group sum of squares

$$\text{SSR} = \sum_i n_i \, (\hat{y}_i - \bar{y})^2$$

Within–group sum of squares

$$\text{SSE} = \sum_i (y_i - \hat{y}_i)^2$$

Degrees of freedom

$$df_{\text{between}} = k - 1, \qquad df_{\text{within}} = N - k$$

Mean squares

$$\text{MSR} = \frac{\text{SSB}}{df_{\text{between}}}, \qquad \text{MSE} = \frac{\text{SSW}}{df_{\text{within}}}$$

Preview

```
#| echo: true
df <- read.csv("city_lifestyle_dataset.csv")

sub <- subset(
  df,
  country %in% c("Europe","Asia","North America","Africa")
  & is.finite(happiness_score)
)

table(sub$country)

tapply(sub$happiness_score, sub$country, summary)

par(mfrow = c(1, 2))
boxplot(happiness_score ~ country, data = sub,
        main = "Happiness by region", xlab = "Region",
        ylab = "Happiness score")
hist(sub$happiness_score,
     main = "Histogram of happiness", xlab = "Happiness score")
par(mfrow = c(1, 1))
```

```
#| exercise: q1_city_anova_manual
#| exercise.lines: 18
#| echo: false

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

sub <- subset(
df,
country %in% c("Europe","Asia","North America","Africa")
& is.finite(happiness_score)
)

sub$region <- factor(sub$country)

y <- sub$happiness_score
g <- sub$region

group_means <- tapply(y, g, ___)
group_ns <- tapply(y, g, ___)
```

```
overall_mean <- ___(y)

SSR <- sum(group_ns * (group_means - overall_mean)^2)

SSE <- sum(tapply(y, g, function(x) sum((x - ___(x))^2)))

k <- length(group_means)
N <- length(y)

df1 <- ___
df2 <- ___

MR <- ___
MSE <- ___

Fstat <- ___ / ___
pval <- pf(___, df1, df2, lower.tail = FALSE)

c(F = Fstat, df1 = df1, df2 = df2, p_value = pval)
```

Use mean for both group and overall means. df1 = k - 1, df2 = N - k. MSR = SSB / df1, MSE = SSW / df2, then Fstat = MSR / MSE.

Solution.

```
#| exercise: q1_city_anova_manual
#| solution: true

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

sub <- subset(
df,
country %in% c("Europe","Asia","North America","Africa")
& is.finite(happiness_score)
)

sub$region <- factor(sub$country)

y <- sub$happiness_score
g <- sub$region
```

```
group_means <- tapply(y, g, mean)
group_ns    <- tapply(y, g, length)

overall_mean <- mean(y)

SSB <- sum(group_ns * (group_means - overall_mean)^2)

SSW <- sum(tapply(y, g, function(x) sum((x - mean(x))^2)))

k  <- length(group_means)
N  <- length(y)

df1 <- k - 1
df2  <- N - k

MSR <- SSB / df1
MSE <- SSW / df2

Fstat <- MSR / MSE
pval  <- pf(Fstat, df1, df2, lower.tail = FALSE)

c(F = Fstat, df1 = df1, df2 = df2, p_value = pval)
```

```
#| exercise: q1_city_anova_manual
#| check: true

gradethis::grade_this({
df <- tryCatch(read.csv("city_lifestyle_dataset.csv"), error = function(e) NULL)
if (is.null(df)) fail("Couldn't read 'city_lifestyle_dataset.csv'.")

sub <- subset(
df,
country %in% c("Europe","Asia","North America","Africa")
& is.finite(happiness_score)
)
sub$region <- factor(sub$country)
y <- sub$happiness_score
g <- sub$region

group_means <- tapply(y, g, mean)
group_ns    <- tapply(y, g, length)
overall_mean <- mean(y)
```

```
SSR <- sum(group_ns * (group_means - overall_mean)^2)
SSE <- sum(tapply(y, g, function(x) sum((x - mean(x))^2)))

k  <- length(group_means)
N  <- length(y)

df1 <- k - 1
df2  <- N - k

MSR <- SSR / df1
MSE <- SSE / df2

Fstat <- MSR / MSE
pval  <- pf(Fstat, df1, df2, lower.tail = FALSE)

exp <- c(F = Fstat, df1 = df1, df2 = df2, p_value = pval)

res <- .result
if (!is.numeric(res) || length(res) != 4L || any(!is.finite(res))) {
fail("Return c(F = ..., df1 = ..., df2 = ..., p_value = ...).")
} else if (max(abs(res - exp)) < 1e-6) {
pass("Correct manual one-way ANOVA for happiness by region.")
} else {
fail("Something is off in SSR/SSE, degrees of freedom, or the F to p-value step.")
}
})
```

Q2 — One-way ANOVA (built-in): Happiness score by region

Now we will redo the same test using R's built-in aov() function on the same subset of the city dataset.

Return c(F = Fstat, df1 = df_region, df2 = df_resid, p_value = p) extracted from the ANOVA table.

> **Info**
>
> For a one-way ANOVA fit with fit <- aov(y ~ group, data = …), you can extract the ANOVA table with summary(fit)[[1]]. The row corresponding to the factor has columns:
> "Df": factor degrees of freedom
> "F value": F statistic

"Pr(>F)": p-value

Preview

```
#| echo: true
df <- read.csv("city_lifestyle_dataset.csv")

sub <- subset(
  df,
  country %in% c("Europe","Asia","North America","Africa")
  & is.finite(happiness_score)
)

table(sub$country)

tapply(sub$happiness_score, sub$country, summary)

par(mfrow = c(1, 2))
boxplot(happiness_score ~ country, data = sub,
        main = "Happiness by region", xlab = "Region",
        ylab = "Happiness score")
hist(sub$happiness_score,
     main = "Histogram of happiness", xlab = "Happiness score")
par(mfrow = c(1, 1))
```

```
#| exercise: q2_city_anova_builtin
#| exercise.lines: 10
#| echo: false

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

sub <- subset(
df,
country %in% c("Europe","Asia","North America","Africa")
& is.finite(happiness_score)
)

sub$region <- factor(sub$country)

fit <- aov(  ~  , data = sub)
```

```
tab <- summary(fit)[[1]]

df_region <- tab[" ", "Df"]
df_resid  <- tab["Residuals", "Df"]
Fstat     <- tab[" ", "F value"]
pval      <- tab[" ", "Pr(>F)"]

c(F = Fstat, df1 = df_region, df2 = df_resid, p_value = pval)
```

Use happiness_score ~ region in aov(), and "region" as the row name in the ANOVA table.

Solution.

```
#| exercise: q2_city_anova_builtin
#| solution: true

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

sub <- subset(
df,
country %in% c("Europe","Asia","North America","Africa")
& is.finite(happiness_score)
)

sub$region <- factor(sub$country)

fit <- aov(happiness_score ~ region, data = sub)

tab <- summary(fit)[[1]]

df_region <- tab["region", "Df"]
df_resid  <- tab["Residuals", "Df"]
Fstat     <- tab["region", "F value"]
pval      <- tab["region", "Pr(>F)"]

c(F = Fstat, df1 = df_region, df2 = df_resid, p_value = pval)
```

```
#| exercise: q2_city_anova_builtin
#| check: true

gradethis::grade_this({
df <- tryCatch(read.csv("city_lifestyle_dataset.csv"), error = function(e) NULL)
```

```
if (is.null(df)) fail("Couldn't read 'city_lifestyle_dataset.csv'.")

sub <- subset(
df,
country %in% c("Europe","Asia","North America","Africa")
& is.finite(happiness_score)
)
sub$region <- factor(sub$country)

fit <- aov(happiness_score ~ region, data = sub)
tab <- summary(fit)[[1]]

df_region <- tab["region", "Df"]
df_resid  <- tab["Residuals", "Df"]
Fstat     <- tab["region", "F value"]
pval      <- tab["region", "Pr(>F)"]

exp <- c(F = Fstat, df1 = df_region, df2 = df_resid, p_value = pval)

res <- .result
if (!is.numeric(res) || length(res) != 4L || any(!is.finite(res))) {
fail("Return c(F = ..., df1 = ..., df2 = ..., p_value = ...).")
} else if (max(abs(res - exp)) < 1e-6) {
pass("Correct one-way ANOVA using aov() for happiness by region.")
} else {
fail("Revisit the aov() call or the extraction from summary(fit)[[1]].")
}
})
```

Q3 — Fisher's LSD (no adjustment): Pairwise region comparisons

Using the same city subset as in Q1–Q2, perform pairwise t-tests on happiness scores between regions without multiple-comparison adjustment. This corresponds to Fisher's LSD after a significant global ANOVA.

Use pairwise.t.test() with p.adjust.method = "none" and return the matrix of p-values.

> **Info**
>
> Fisher's LSD (Least Significant Difference) is essentially:
> Run a global ANOVA to check that at least one mean differs.

If significant, run unadjusted pairwise t-tests between groups, using the same residual variance.
In R, pairwise.t.test(y, group, p.adjust.method = “none”)$p.value gives a matrix of unadjusted pairwise p-values.

Preview

```
#| echo: true
df <- read.csv("city_lifestyle_dataset.csv")

sub <- subset(
  df,
  country %in% c("Europe","Asia","North America","Africa")
  & is.finite(happiness_score)
)

table(sub$country)

tapply(sub$happiness_score, sub$country, summary)

par(mfrow = c(1, 2))
boxplot(happiness_score ~ country, data = sub,
        main = "Happiness by region", xlab = "Region",
        ylab = "Happiness score")
hist(sub$happiness_score,
     main = "Histogram of happiness", xlab = "Happiness score")
par(mfrow = c(1, 1))
```

```
#| exercise: q3_city_lsd
#| exercise.lines: 8
#| echo: false

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

sub <- subset(
df,
country %in% c("Europe","Asia","North America","Africa")
& is.finite(happiness_score)
)

sub$region <- factor(sub$country)
```

```
out <- pairwise.t.test( ,  , p.adjust.method = " ")

out$p.value
```

Use sub$happiness_score as the response, sub$region as the group, and “none” for p.adjust.method.

Solution.

```
#| exercise: q3_city_lsd
#| solution: true

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

sub <- subset(
df,
country %in% c("Europe","Asia","North America","Africa")
& is.finite(happiness_score)
)

sub$region <- factor(sub$country)

out <- pairwise.t.test(sub$happiness_score,
sub$region,
p.adjust.method = "none")

out$p.value
```

```
#| exercise: q3_city_lsd
#| check: true

gradethis::grade_this({
df <- tryCatch(read.csv("city_lifestyle_dataset.csv"), error = function(e) NULL)
if (is.null(df)) fail("Couldn't read 'city_lifestyle_dataset.csv'.")

sub <- subset(
df,
country %in% c("Europe","Asia","North America","Africa") &
is.finite(happiness_score)
)
sub$region <- factor(sub$country)
```

```
exp <- pairwise.t.test(sub$happiness_score,
sub$region,
p.adjust.method = "none")$p.value

res <- .result

if (!is.matrix(res)) {
fail("Return the p-value matrix from pairwise.t.test(...).")
}

# check dimensions and dimnames match

if (!identical(dim(res), dim(exp)) ||
!identical(dimnames(res), dimnames(exp))) {
fail("Row/column structure should match pairwise.t.test(...).")
}

# numeric comparison, ignoring NA entries

if (max(abs(res - exp), na.rm = TRUE) < 1e-8) {
pass("Correct Fisher's LSD pairwise p-values (no adjustment).")
} else {
fail("Check your pairwise.t.test() call and that you used p.adjust.method = 'none'.")
}
})
```

Q4 — Bonferroni-adjusted pairwise region comparisons

Now repeat the previous question but use a Bonferroni correction for multiple comparisons. Again, return the matrix of p-values.

> **Info**
>
> The Bonferroni adjustment is a simple and conservative way to control the family-wise error rate when making many comparisons. In R, use p.adjust.method = “bonferroni” in pairwise.t.test().

Preview

```
#| echo: true
df <- read.csv("city_lifestyle_dataset.csv")

sub <- subset(
  df,
  country %in% c("Europe","Asia","North America","Africa")
  & is.finite(happiness_score)
)

table(sub$country)

tapply(sub$happiness_score, sub$country, summary)

par(mfrow = c(1, 2))
boxplot(happiness_score ~ country, data = sub,
        main = "Happiness by region", xlab = "Region",
        ylab = "Happiness score")
hist(sub$happiness_score,
     main = "Histogram of happiness", xlab = "Happiness score")
par(mfrow = c(1, 1))
```

```
#| exercise: q4_city_bonferroni
#| exercise.lines: 8
#| echo: false

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

sub <- subset(
df,
country %in% c("Europe","Asia","North America","Africa")
& is.finite(happiness_score)
)

sub$region <- factor(sub$country)

out <- pairwise.t.test( ,  ,
p.adjust.method = " ")

out$p.value
```

Use the same sub$happiness_score and sub$region, but set p.adjust.method = “bonferroni”.

Solution.

```
#| exercise: q4_city_bonferroni
#| solution: true

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

sub <- subset(
df,
country %in% c("Europe","Asia","North America","Africa")
& is.finite(happiness_score)
)

sub$region <- factor(sub$country)

out <- pairwise.t.test(sub$happiness_score,
sub$region,
p.adjust.method = "bonferroni")

out$p.value
```

```
#| exercise: q4_city_bonferroni
#| check: true

gradethis::grade_this({
df <- tryCatch(read.csv("city_lifestyle_dataset.csv"), error = function(e) NULL)
if (is.null(df)) fail("Couldn't read 'city_lifestyle_dataset.csv'.")

sub <- subset(
df,
country %in% c("Europe","Asia","North America","Africa") &
is.finite(happiness_score)
)
sub$region <- factor(sub$country)

exp <- pairwise.t.test(sub$happiness_score,
sub$region,
p.adjust.method = "bonferroni")$p.value

res <- .result
```

```
# Must be a matrix

if (!is.matrix(res)) {
fail("Return the p-value matrix from pairwise.t.test(...).")
}

# Structure (dims + names) must match

if (!identical(dim(res), dim(exp)) ||
!identical(dimnames(res), dimnames(exp))) {
fail("Row/column structure should match pairwise.t.test(...).")
}

# Compare numeric entries, ignoring NAs

if (max(abs(res - exp), na.rm = TRUE) < 1e-8) {
pass("Correct Bonferroni-adjusted pairwise p-values.")
} else {
fail("Check that you used p.adjust.method = 'bonferroni' on the same subset.")
}
})
```

Q5 — One-way ANOVA (manual): Exam score by motivation level

In the student performance dataset, we want to test whether the **mean exam score** differs across **motivation levels** (Low, Medium, High).

We test

$$H_0 : \mu_{\text{Low}} = \mu_{\text{Medium}} = \mu_{\text{High}}$$

vs

$$H_1 : \text{ at least one of } \mu_{\text{Low}}, \mu_{\text{Medium}}, \mu_{\text{High}} \text{ is different.}$$

Use a **one-way ANOVA** on `Exam_Score`, computed *manually* from sums of squares.

Your task:

- Subset to rows with non-missing `Exam_Score` and `Motivation_Level`.

- Treat `Motivation_Level` as a factor with three levels (Low, Medium, High).
- Compute SSR, SSE, `df1`, `df2`, MSR, MSE, (F), and the p-value

$$F = \frac{\text{MSR}}{\text{MSE}}, \qquad p = P(F_{df_1, df_2} \geq F).$$

Return a named numeric vector:
`c(F = Fstat, df1 = df1, df2 = df2, p_value = pval)`.

Photo by Leonardo Vargas on Unsplash

Preview

```
#| echo: true
df <- read.csv("student_performance.csv")

sub <- subset(df,
!is.na(Exam_Score) &
!is.na(Motivation_Level))

table(sub$Motivation_Level)

tapply(sub$Exam_Score, sub$Motivation_Level, summary)

par(mfrow = c(1, 2))
boxplot(Exam_Score ~ Motivation_Level, data = sub,
main = "Exam score by motivation",
xlab = "Motivation level", ylab = "Exam score")
hist(sub$Exam_Score,
main = "Histogram of exam scores",
xlab = "Exam score")
par(mfrow = c(1, 1))
```

```
#| exercise: q5_stud_anova_manual
#| exercise.lines: 18
#| echo: false

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

sub <- subset(df,
!is.na(Exam_Score) &
```

```
!is.na(Motivation_Level))

sub$mot <- factor(sub$Motivation_Level)

y <- sub$Exam_Score
g <- sub$mot

group_means <- tapply(y, g, ___)
group_ns    <- tapply(y, g, ___)

overall_mean <- ___(y)

SSR <- sum(group_ns * (group_means - overall_mean)^2)

SSE <- sum(tapply(y, g, function(x) sum((x - ___(x))^2)))

k  <- length(group_means)
N  <- length(y)

df1 <- ___
df2  <- ___

MSR <- ___
MSE <- ___

Fstat <- ___ / ___
pval  <- pf(___, df1, df2, lower.tail = FALSE)

c(F = Fstat, df1 = df1, df2 = df2, p_value = pval)
```

Same pattern as Q1: use mean for group and overall means. df1 = k - 1, df2 = N - k, MSR = SSR/df1, MSE = SSE/df2.

Solution.

```
#| exercise: q5_stud_anova_manual
#| solution: true

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

sub <- subset(df,
!is.na(Exam_Score) &
```

```
!is.na(Motivation_Level))

sub$mot <- factor(sub$Motivation_Level)

y <- sub$Exam_Score
g <- sub$mot

group_means <- tapply(y, g, mean)
group_ns    <- tapply(y, g, length)

overall_mean <- mean(y)

SSB <- sum(group_ns * (group_means - overall_mean)^2)

SSW <- sum(tapply(y, g, function(x) sum((x - mean(x))^2)))

k  <- length(group_means)
N  <- length(y)

df1 <- k - 1
df2  <- N - k

MSR <- SSB / df1
MSE <- SSW / df2

Fstat <- MSR / MSE
pval  <- pf(Fstat, df1, df2, lower.tail = FALSE)

c(F = Fstat, df1 = df1, df2 = df2, p_value = pval)
```

```
#| exercise: q5_stud_anova_manual
#| check: true

gradethis::grade_this({
df <- tryCatch(read.csv("student_performance.csv"), error = function(e) NULL)
if (is.null(df)) fail("Couldn't read 'student_performance.csv'.")

sub <- subset(df,
!is.na(Exam_Score) &
!is.na(Motivation_Level))

sub$mot <- factor(sub$Motivation_Level)
```

```
y <- sub$Exam_Score
g <- sub$mot

group_means <- tapply(y, g, mean)
group_ns    <- tapply(y, g, length)

overall_mean <- mean(y)

SSR <- sum(group_ns * (group_means - overall_mean)^2)
SSE <- sum(tapply(y, g, function(x) sum((x - mean(x))^2)))

k  <- length(group_means)
N  <- length(y)

df1 <- k - 1
df2  <- N - k

MSR <- SSR / df1
MSE <- MSE <- SSE / df2

Fstat <- MSR / MSE
pval  <- pf(Fstat, df1, df2, lower.tail = FALSE)

exp <- c(F = Fstat, df1 = df1, df2 = df2, p_value = pval)

res <- .result
if (!is.numeric(res) || length(res) != 4L || any(!is.finite(res))) {
fail("Return c(F = ..., df1 = ..., df2 = ..., p_value = ...).")
} else if (max(abs(res - exp)) < 1e-6) {
pass("Correct manual one-way ANOVA for exam score by motivation.")
} else {
fail("Re-check your SSR/SSE, degrees of freedom, and F to p-value computation.")
}
})
```

Q6 — One-way ANOVA (built-in): Exam score by motivation level

Now use aov() to test for differences in mean exam score across motivation levels.

Return c(F = Fstat, df1 = df_mot, df2 = df_resid, p_value = pval).

Preview

```
#| echo: true
df <- read.csv("student_performance.csv")

sub <- subset(df,
!is.na(Exam_Score) &
!is.na(Motivation_Level))

table(sub$Motivation_Level)

tapply(sub$Exam_Score, sub$Motivation_Level, summary)

par(mfrow = c(1, 2))
boxplot(Exam_Score ~ Motivation_Level, data = sub,
main = "Exam score by motivation",
xlab = "Motivation level", ylab = "Exam score")
hist(sub$Exam_Score,
main = "Histogram of exam scores",
xlab = "Exam score")
par(mfrow = c(1, 1))
```

```
#| exercise: q6_stud_anova_builtin
#| exercise.lines: 10
#| echo: false

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

sub <- subset(df,
!is.na(Exam_Score) &
!is.na(Motivation_Level))

sub$mot <- factor(sub$Motivation_Level)

fit <- aov(  ~  , data = sub)

tab <- summary(fit)[[1]]

df_mot   <- tab[" ", "Df"]
df_resid <- tab["Residuals", "Df"]
Fstat    <- tab[" ", "F value"]
pval     <- tab[" ", "Pr(>F)"]
```

```
c(F = Fstat, df1 = df_mot, df2 = df_resid, p_value = pval)
```

Use Exam_Score ~ mot, and the row name "mot" in the ANOVA table.

Solution.

```
#| exercise: q6_stud_anova_builtin
#| solution: true

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

sub <- subset(df,
!is.na(Exam_Score) &
!is.na(Motivation_Level))

sub$mot <- factor(sub$Motivation_Level)

fit <- aov(Exam_Score ~ mot, data = sub)

tab <- summary(fit)[[1]]

df_mot   <- tab["mot", "Df"]
df_resid <- tab["Residuals", "Df"]
Fstat    <- tab["mot", "F value"]
pval     <- tab["mot", "Pr(>F)"]

c(F = Fstat, df1 = df_mot, df2 = df_resid, p_value = pval)
```

```
#| exercise: q6_stud_anova_builtin
#| check: true

gradethis::grade_this({
df <- tryCatch(read.csv("student_performance.csv"), error = function(e) NULL)
if (is.null(df)) fail("Couldn't read 'student_performance.csv'.")

sub <- subset(df,
!is.na(Exam_Score) &
!is.na(Motivation_Level))

sub$mot <- factor(sub$Motivation_Level)
```

```
fit <- aov(Exam_Score ~ mot, data = sub)
tab <- summary(fit)[[1]]

df_mot   <- tab["mot", "Df"]
df_resid <- tab["Residuals", "Df"]
Fstat    <- tab["mot", "F value"]
pval     <- tab["mot", "Pr(>F)"]

exp <- c(F = Fstat, df1 = df_mot, df2 = df_resid, p_value = pval)

res <- .result
if (!is.numeric(res) || length(res) != 4L || any(!is.finite(res))) {
fail("Return c(F = ..., df1 = ..., df2 = ..., p_value = ...).")
} else if (max(abs(res - exp)) < 1e-6) {
pass("Correct ANOVA via aov() for exam score by motivation.")
} else {
fail("Check your aov() model formula and extraction from summary().")
}
})
```

Q7 — Fisher's LSD: Exam score pairwise comparisons by motivation

Using the same subset as Q5–Q6, compute unadjusted pairwise t-tests on Exam_Score across motivation levels (Fisher's LSD).

Return the matrix of p-values.

Preview

```
#| echo: true
df <- read.csv("student_performance.csv")

sub <- subset(df,
!is.na(Exam_Score) &
!is.na(Motivation_Level))

table(sub$Motivation_Level)

tapply(sub$Exam_Score, sub$Motivation_Level, summary)

par(mfrow = c(1, 2))
boxplot(Exam_Score ~ Motivation_Level, data = sub,
main = "Exam score by motivation",
xlab = "Motivation level", ylab = "Exam score")
hist(sub$Exam_Score,
main = "Histogram of exam scores",
xlab = "Exam score")
par(mfrow = c(1, 1))
```

```
#| exercise: q7_stud_lsd
#| exercise.lines: 8
#| echo: false

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

sub <- subset(df,
!is.na(Exam_Score) &
!is.na(Motivation_Level))

sub$mot <- factor(sub$Motivation_Level)

out <- pairwise.t.test( ,  ,
p.adjust.method = " ")

out$p.value
```

Use sub$Exam_Score, sub$mot, and “none” as the adjustment method.

Solution.

```
#| exercise: q7_stud_lsd
#| solution: true

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

sub <- subset(df,
!is.na(Exam_Score) &
!is.na(Motivation_Level))

sub$mot <- factor(sub$Motivation_Level)

out <- pairwise.t.test(sub$Exam_Score,
sub$mot,
p.adjust.method = "none")

out$p.value
```

```
#| exercise: q7_stud_lsd
#| check: true

gradethis::grade_this({
df <- tryCatch(read.csv("student_performance.csv"), error = function(e) NULL)
if (is.null(df)) fail("Couldn't read 'student_performance.csv'.")

sub <- subset(
df,
!is.na(Exam_Score) &
!is.na(Motivation_Level)
)
sub$mot <- factor(sub$Motivation_Level)

exp <- pairwise.t.test(
sub$Exam_Score,
sub$mot,
p.adjust.method = "none"
)$p.value

res <- .result

# Must be a matrix

if (!is.matrix(res)) {
```

```
fail("Return the p-value matrix from pairwise.t.test(...).")
}

# Dimensions and dimnames should match

if (!identical(dim(res), dim(exp)) ||
!identical(dimnames(res), dimnames(exp))) {
fail("Row/column structure should match pairwise.t.test(...).")
}

# Compare numeric entries, ignoring NAs

if (max(abs(res - exp), na.rm = TRUE) < 1e-8) {
pass("Correct Fisher's LSD pairwise p-values for motivation levels.")
} else {
fail("Check your pairwise.t.test() call and that you used p.adjust.method = 'none'.")
}
})
```

Q8 — Bonferroni: Exam score pairwise comparisons by parental education

Finally, investigate another factor: parental education level (High School, College, Postgraduate).

Use the student dataset to:

Subset to rows with non-missing Exam_Score and Parental_Education_Level.

Treat Parental_Education_Level as a factor with three levels.

Run pairwise.t.test() on Exam_Score across parental education levels, using Bonferroni adjustment.

Return the p-value matrix.

Preview

```
#| echo: true
df <- read.csv("student_performance.csv")
sub <- subset(df,
!is.na(Exam_Score) &
!is.na(Parental_Education_Level))

table(sub$Parental_Education_Level)

tapply(sub$Exam_Score, sub$Parental_Education_Level, summary)

par(mfrow = c(1, 2))
boxplot(Exam_Score ~ Parental_Education_Level, data = sub,
main = "Exam score by parental education",
xlab = "Parental education", ylab = "Exam score")
hist(sub$Exam_Score,
main = "Histogram of exam scores",
xlab = "Exam score")
par(mfrow = c(1, 1))
```

```
#| exercise: q8_stud_bonf_edu
#| exercise.lines: 8
#| echo: false

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

sub <- subset(df,
!is.na(Exam_Score) &
!is.na(Parental_Education_Level))

sub$edu <- factor(sub$Parental_Education_Level)

out <- pairwise.t.test( ,  ,
p.adjust.method = " ")

out$p.value
```

Use sub$Exam_Score and sub$edu with p.adjust.method = “bonferroni”.

Solution.

```r
#| exercise: q8_stud_bonf_edu
#| solution: true

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

sub <- subset(df,
!is.na(Exam_Score) &
!is.na(Parental_Education_Level))

sub$edu <- factor(sub$Parental_Education_Level)

out <- pairwise.t.test(sub$Exam_Score,
sub$edu,
p.adjust.method = "bonferroni")

out$p.value
```

```
#| exercise: q8_stud_bonf_edu
#| check: true

gradethis::grade_this({
  df <- tryCatch(read.csv("student_performance.csv"), error = function(e) NULL)
  if (is.null(df)) fail("Couldn't read 'student_performance.csv'.")

  sub <- subset(df, !is.na(Exam_Score) & !is.na(Parental_Education_Level))
  sub$edu <- factor(sub$Parental_Education_Level)

  exp <- pairwise.t.test(sub$Exam_Score, sub$edu, p.adjust.method = "bonferroni")$p.value
  res <- .result

  if (!is.matrix(res)) fail("Return the p-value matrix from pairwise.t.test(...).")

  if (!identical(dim(res), dim(exp)) || !identical(dimnames(res), dimnames(exp))) {
    fail("Row/column structure should match pairwise.t.test(...).")
  }

  if (max(abs(res - exp), na.rm = TRUE) < 1e-8) {
    pass("Correct Bonferroni-adjusted pairwise p-values by parental education.")
  } else {
    fail("Re-check the subset, the grouping factor, and p.adjust.method = 'bonferroni'.")
  }
})
```