

Lab 03

In this week's lab we will work through the experimental design workflow for a one-way ANOVA.

Learning outcomes

In this lab, you will work towards achieving learning outcomes...

Lab objectives

At the end of the lab, students should be able to:

- Understand the workflow involved in conducting a one-way Analysis of Variance (ANOVA), including data exploration, model fitting, and interpretation of results.
- Analyse an experiment with a 1-way ANOVA and interpret the results.
- Explain situations when a 2-sample t-test gives the same results as a 1-way ANOVA.

Prerequisites

The following packages are used for this lab: `readxl`, `ggplot2`, `dplyr` and `tidyr`. To install these packages, run the following code **in the console**.

CODE

```
install.packages(c("readxl", "ggplot2", "dplyr", "tidyr"))
```

Exercise 1 – diatoms

Medley & Clements (1998) sampled 34 locations along streams for diversity of diatoms. Each site was classified according to the Zn concentration in the water. There were 4 classes; background, low, medium and high. Were there *differences* between each of the groupings in term of diatom diversity? Let's find out.

Practice 1 – Data import

Import the Diatoms worksheet from the `diatoms.xlsx` file into R.

If you have difficulty with this step please refer to this week's tutorial on importing MS Excel files.

```
CODE
library(readxl)

# You will need to look at the Excel file and work out
# the correct worksheet name and range:
diatoms <- read_excel(
  path = "diatoms.xlsx",
  sheet = ...,
  range = ...
)
```

Workflow

We will briefly go through a typical analytical workflow which involves data exploration, model fitting, checking of model assumptions¹ and interpretation of the results.

Data exploration

Data structure

Checking the structure of the data is the first step in any data analysis. This is done using the `str()` function in R.

```
CODE
str(diatoms)
```

Of particular interest are the variables `Stream` and `Zinc`, which have been classified as characters (`chr`).

```
OUTPUT
tibble [34 × 4] (S3: tbl_df/tbl/data.frame)
 $ Stream   : chr [1:34] "Eagle" "Blue" "Blue" "Blue" ...
 $ Zinc     : chr [1:34] "BACK" "BACK" "BACK" "BACK" ...
 $ Diversity: num [1:34] 2.27 1.7 2.05 1.98 2.2 1.53 0.76 1.89 1.4 2.18 ...
 $ Group    : num [1:34] 1 1 1 1 1 1 1 1 2 2 ...
```

These variables are most likely factors and should be converted to such. This can be done using `as.factor()`.

```
CODE
```

¹Assumptions will be covered formally next week when we look at model residuals.

```
diatoms$Zinc <- as.factor(diatoms$Zinc)
diatoms$Stream <- as.factor(diatoms$Stream)
```

💡 Tip

The tidyverse approach – We can use the `mutate()` function to convert the Zinc and Stream variables to factors.

```
CODE
library(dplyr)
diatoms <- diatoms %>%
  mutate(
    Zinc = as.factor(Zinc),
    Stream = as.factor(Stream)
  )
```

We can then check if the conversion was successful by using the `str()` function again.

```
CODE
str(diatoms)
```

```
OUTPUT
tibble [34 × 4] (S3: tbl_df/tbl/data.frame)
 $ Stream   : Factor w/ 6 levels "Arkan","Blue",..: 4 2 2 5 6 6 6 1 1 ...
 $ Zinc     : Factor w/ 4 levels "BACK","HIGH",..: 1 1 1 1 1 1 1 3 3 ...
 $ Diversity: num [1:34] 2.27 1.7 2.05 1.98 2.2 1.53 0.76 1.89 1.4 2.18 ...
 $ Group    : num [1:34] 1 1 1 1 1 1 1 1 2 2 ...
```

Summary statistics

Use `summary()` for a quick overview of common statistical measures for each variable in the data frame. However it is not informative when we are interested in differences between groups or factors as it only gives the summary statistics for the entire data set.

```
CODE
summary(diatoms)
```

```
OUTPUT
```

| Stream | Zinc | Diversity | Group |
|---------|--------|---------------|---------------|
| Arkan:7 | BACK:8 | Min. :0.630 | Min. :1.000 |
| Blue :7 | HIGH:9 | 1st Qu.:1.377 | 1st Qu.:2.000 |
| Chalk:5 | LOW :8 | Median :1.855 | Median :3.000 |
| Eagle:4 | MED :9 | Mean :1.694 | Mean :2.559 |
| Snake:5 | | 3rd Qu.:2.058 | 3rd Qu.:3.750 |
| Splat:6 | | Max. :2.830 | Max. :4.000 |

It is more useful to calculate summary statistics for each level of Zn contamination, as we are interested in differences between the mean of each group.

We can use the `tapply()` function to calculate the mean and standard deviation for each level of Zinc. This has to be done separately for each summary statistic (mean and standard deviation).

The general structure of the `tapply()` function is 3 arguments which are described below based on the code above:

- the response variable on which we wish to apply the function, `diatoms$Diversity`;
- the categorical variable which indicates the groups we wish to separately apply the function to, `diatoms$Zinc`;
- the function we are using, `mean()`.

CODE

```
tapply(
  X = diatoms$Diversity,
  INDEX = diatoms$Zinc,
  FUN = mean
)
```

OUTPUT

| BACK | HIGH | LOW | MED |
|----------|----------|----------|----------|
| 1.797500 | 1.277778 | 2.032500 | 1.717778 |

CODE

```
tapply(
  X = diatoms$Diversity,
  INDEX = diatoms$Zinc,
  FUN = sd
)
```

OUTPUT

| BACK | HIGH | LOW | MED |
|-----------|-----------|-----------|-----------|
| 0.4852613 | 0.4268717 | 0.4449960 | 0.5030104 |

Tip

The tidyverse approach – Use the `group_by()` and `summarise()` functions to calculate the mean and standard deviation for each level of Zinc. The code is longer, but readable as you can see the sequence of operations.

CODE

```
diatoms %>%
  group_by(Zinc) %>%
  summarise(
    mean = mean(Diversity),
    sd = sd(Diversity)
  )
```

Graphical summaries

Numerical summaries are nice, but visual summaries are often more informative and are the **summary of choice** for publication. They can also help us to check the assumptions of the statistical test (although sometimes this is not possible until we have fitted the model).

In the case of ANOVA model we are interested in the distribution of the response variable for each level of the factor. The table below gives heuristic rules about which graphical summary to use based on the number of observations.

| observations | graphics | command |
|--------------|---------------|-------------------------------|
| 1-5 | plot raw data | stripchart() or geom_jitter() |
| 6-20 | boxplot | boxplot() or geom_boxplot() |
| 20 or more | histogram | hist() or geom_histogram() |

We will show examples of both the histogram and boxplot for the diatom data. We recommend that you use ggplot2 for your graphical summaries as it is more flexible and has a more consistent syntax than base R graphics. However, we will show both approaches here.

Histogram

As there are less than 10 observations per group the histogram may not be very informative. However, it may still be useful for checking the normality of the data. You may be able to see the limitations of the histogram for small sample sizes.



Tip

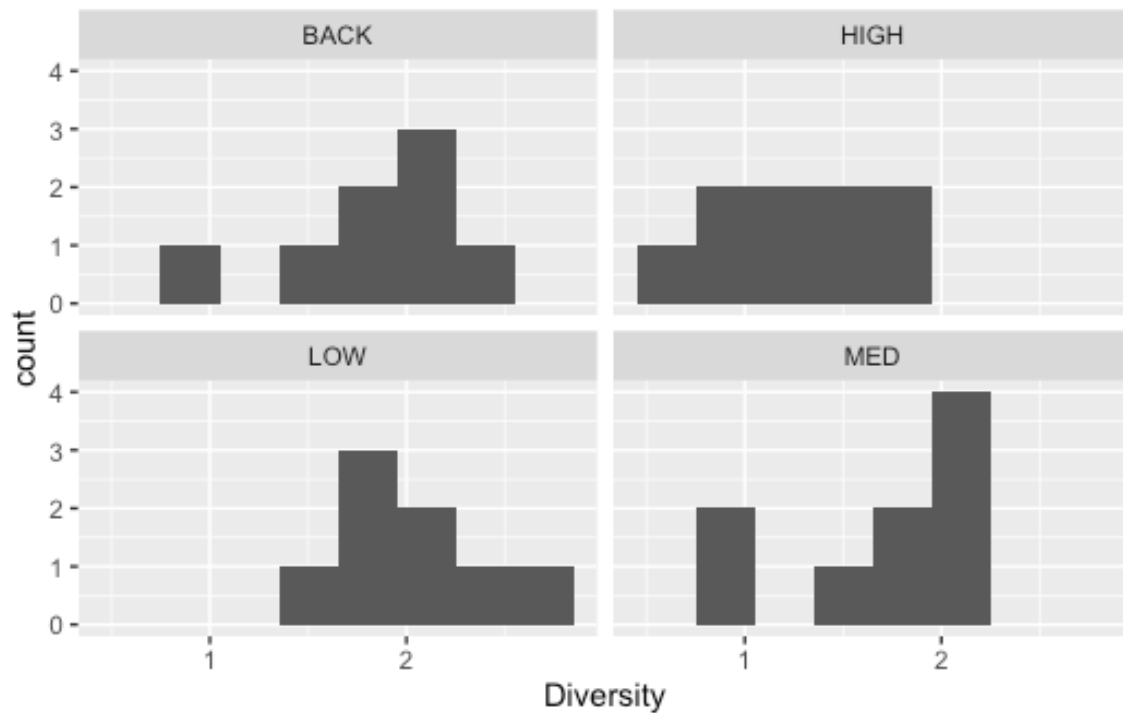
To calculate the number of observations per group:

```
CODE
tapply(
  X = diatoms$Diversity,
  INDEX = diatoms$Zinc,
  FUN = length
)
```

ggplot2

```
CODE
library(ggplot2)
ggplot(diatoms, aes(x = Diversity)) +
```

```
geom_histogram(binwidth = .3) +  
facet_wrap(~Zinc)
```



base R (no loops)

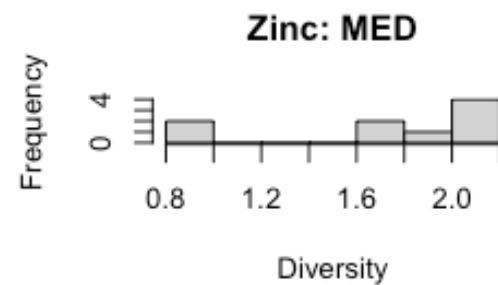
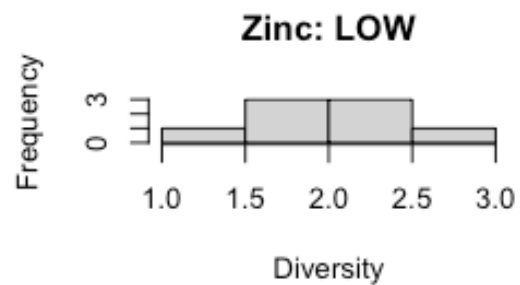
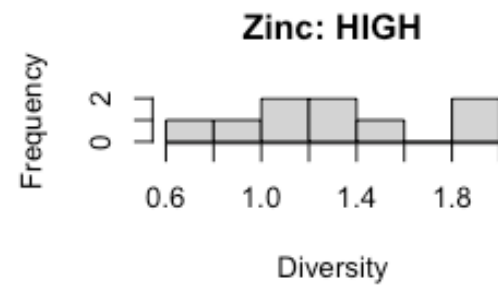
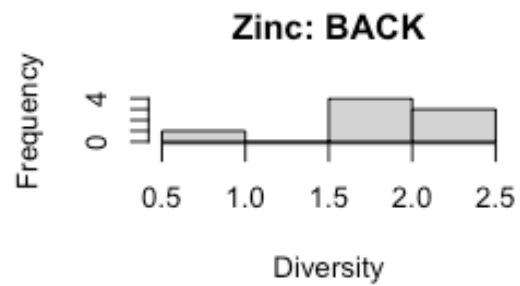
CODE

```
par(mfrow = c(2, 2))  
hist(  
  x = diatoms$Diversity[diatoms$Zinc == "BACK"],  
  main = "Zinc: BACK",  
  xlab = "Diversity"  
)  
hist(  
  x = diatoms$Diversity[diatoms$Zinc == "HIGH"],  
  main = "Zinc: HIGH",  
  xlab = "Diversity"  
)  
hist(  
  x = diatoms$Diversity[diatoms$Zinc == "LOW"],  
  main = "Zinc: LOW",  
  xlab = "Diversity"  
)  
hist(  
  x = diatoms$Diversity[diatoms$Zinc == "MED"],
```

```

    main = "Zinc: MED",
    xlab = "Diversity"
  )

```

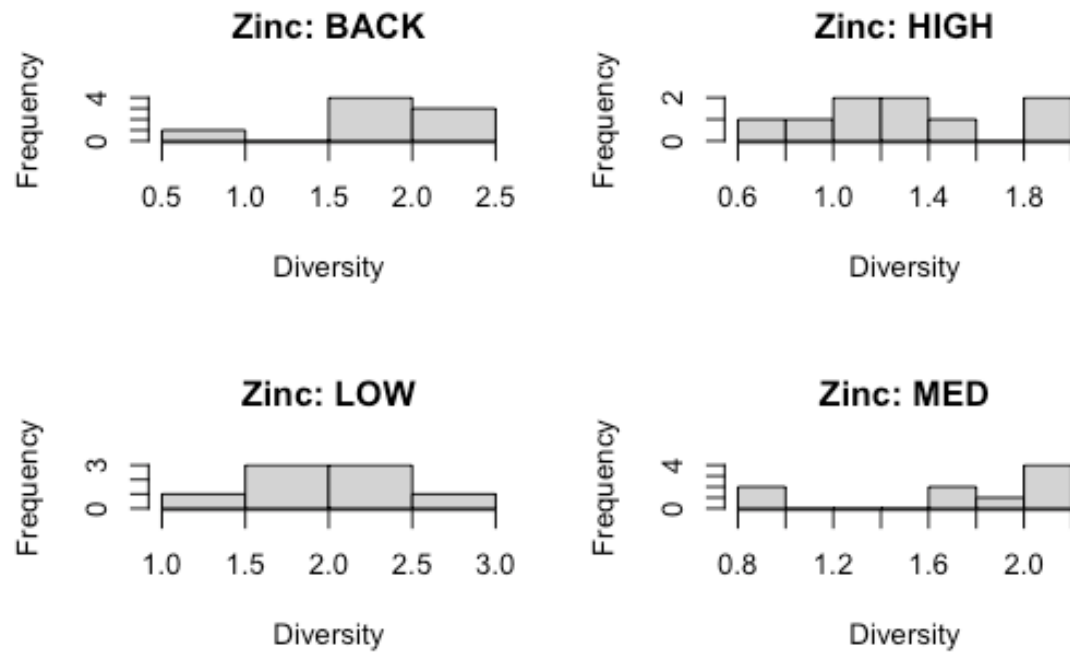


base R (loops)

```

CODE
par(mfrow = c(2, 2))
for (i in levels(diatoms$Zinc)) {
  hist(
    x = diatoms$Diversity[diatoms$Zinc == i],
    main = paste("Zinc:", i),
    xlab = "Diversity"
  )
}

```



Boxplot

A more appropriate plot for this data is the boxplot. If we want to format the plot for publication we should make the plot clear by adding axis labels and a figure caption.

base R

```
CODE
boxplot(
  Diversity ~ Zinc,
  data = diatoms,
  ylab = "Diversity",
  xlab = "Zinc concentration"
)
```

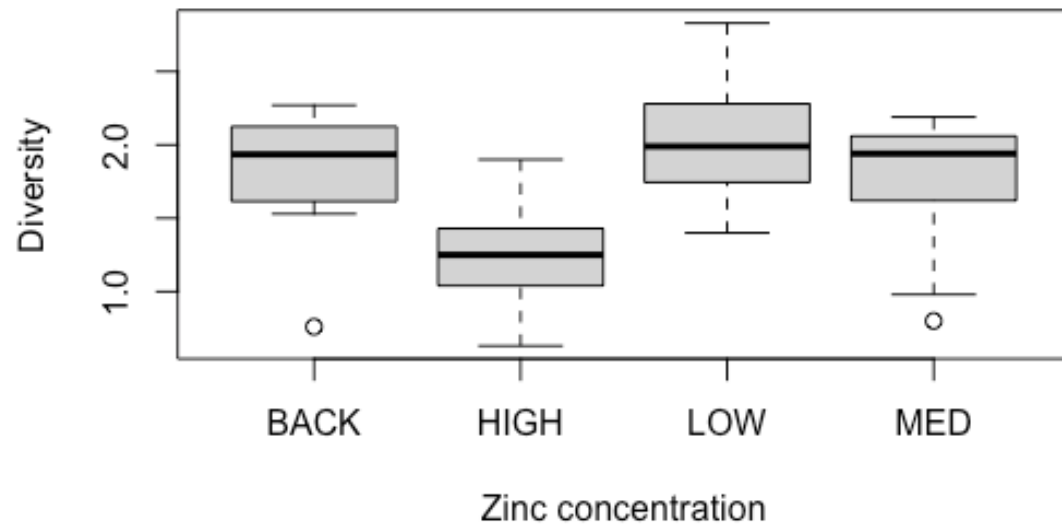



Figure 1: Boxplot of diatom diversity by Zinc concentration.

ggplot2

```
CODE
ggplot(diatoms, aes(x = Zinc, y = Diversity)) +
  geom_boxplot() +
  ylab("Diversity") +
  theme_minimal(base_size = 12)
```

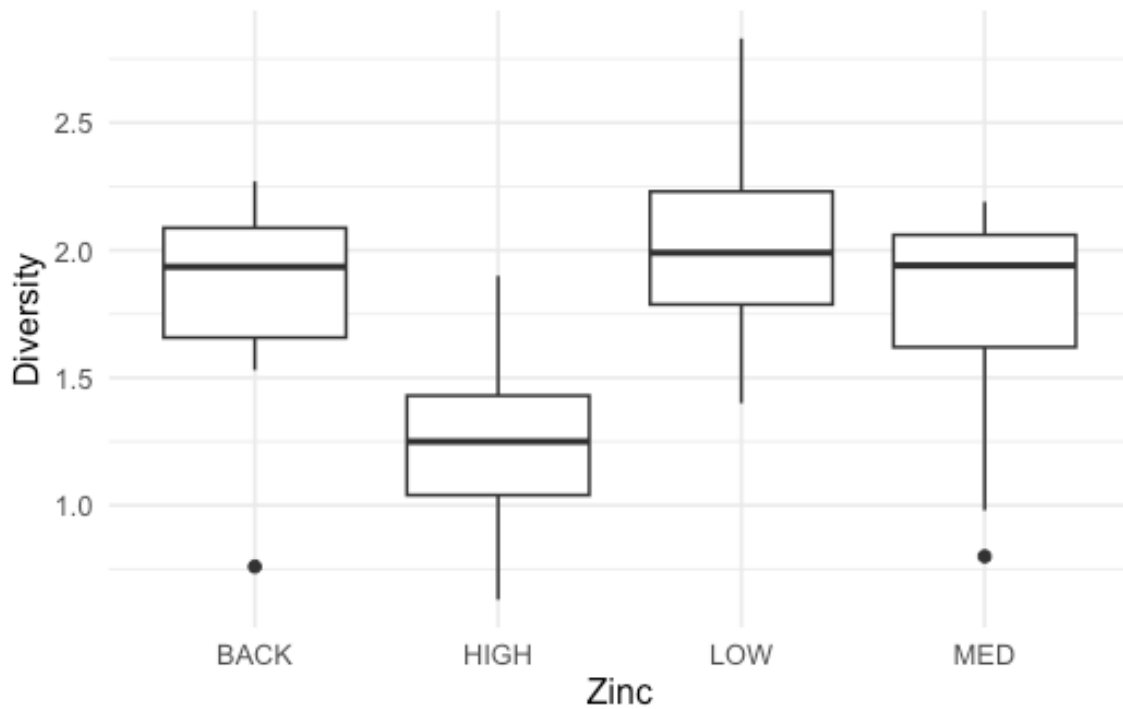


Figure 2: Boxplot of diatom diversity by Zinc concentration.

Question 1

What can you say about the the different levels of Zinc from looking at the mean and standard deviation values when running the following?

```
CODE
tapply(
  X = diatoms$Diversity,
  INDEX = diatoms$Zinc,
  FUN = mean
)

tapply(
  X = diatoms$Diversity,
  INDEX = diatoms$Zinc,
  FUN = sd
)
```

Question 2

Can you interpret and describe the boxplot above?

Model fitting

A 1-way ANOVA involves one treatment (or grouping) factor. The model we are fitting is:

$$y_{i,j} = \mu_i + \epsilon_{i,j}$$

where:

1. $y_{i,j}$ is the response for observation j in treatment (or group) i ,
2. μ_j is the mean of treatment (or group) i ,
3. $\epsilon_{i,j}$ is the residual term which is an independent random variable that has a mean of 0, constant variance and is normally distributed. This can be expressed shorthand as $\sim N(0, \sigma_2)$. The residual MS estimates σ_2 .

The statistical hypotheses we are testing are:

$$H_0 : \mu_1 = \mu_2 = \dots \mu_t$$

$$H_1 : \text{not all } \mu_j \text{ are equal}$$

where μ_j is the mean diatom diversity for each level of Zn concentration.

The code below fits the ANOVA model using the `aov()` function and saves the it to an object called `anova.diatoms`. We can then extract the ANOVA table using the `summary` function.

CODE

```
anova.diatoms <- aov(formula = Diversity ~ Zinc, data = diatoms)
summary(anova.diatoms)
```

OUTPUT

```
      Df Sum Sq Mean Sq F value Pr(>F)
Zinc      3   2.567    0.8555   3.939 0.0176 *
Residuals 30   6.516    0.2172
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Assumptions and interpretation of results

The assumptions of the ANOVA model are:

- The residuals are independent,
- The residuals are normally distributed,
- The residuals have constant variance.

Based on the boxplots and histograms during data exploration, the assumptions of normality and equal variances are met. We will discuss assumptions again in greater detail next week when we start to look at residuals.

We can only interpret the results of the ANOVA model if the assumptions are met. We can report the results the following way:

Reporting: The results indicate that there are significant differences between the levels of Zn concentration in terms of diatom diversity ($F = 3.9$, $df = 3, 30$, $P = 0.02$).

Post-hoc testing

The ANOVA test only tells us that there are differences between the groups, but it does not tell us which groups are different. We can use the `emmeans` package to perform post-hoc testing.

CODE

```
library(emmeans)
```

OUTPUT

```
Welcome to emmeans.  
Caution: You lose important information if you filter this package's results.  
See '? untidy'
```

CODE

```
posthoc ← emmeans(anova.diatoms, "Zinc")  
posthoc
```

OUTPUT

| Zinc | emmean | SE | df | lower.CL | upper.CL |
|------|--------|-------|----|----------|----------|
| BACK | 1.80 | 0.165 | 30 | 1.461 | 2.13 |
| HIGH | 1.28 | 0.155 | 30 | 0.961 | 1.60 |
| LOW | 2.03 | 0.165 | 30 | 1.696 | 2.37 |
| MED | 1.72 | 0.155 | 30 | 1.401 | 2.04 |

Confidence level used: 0.95

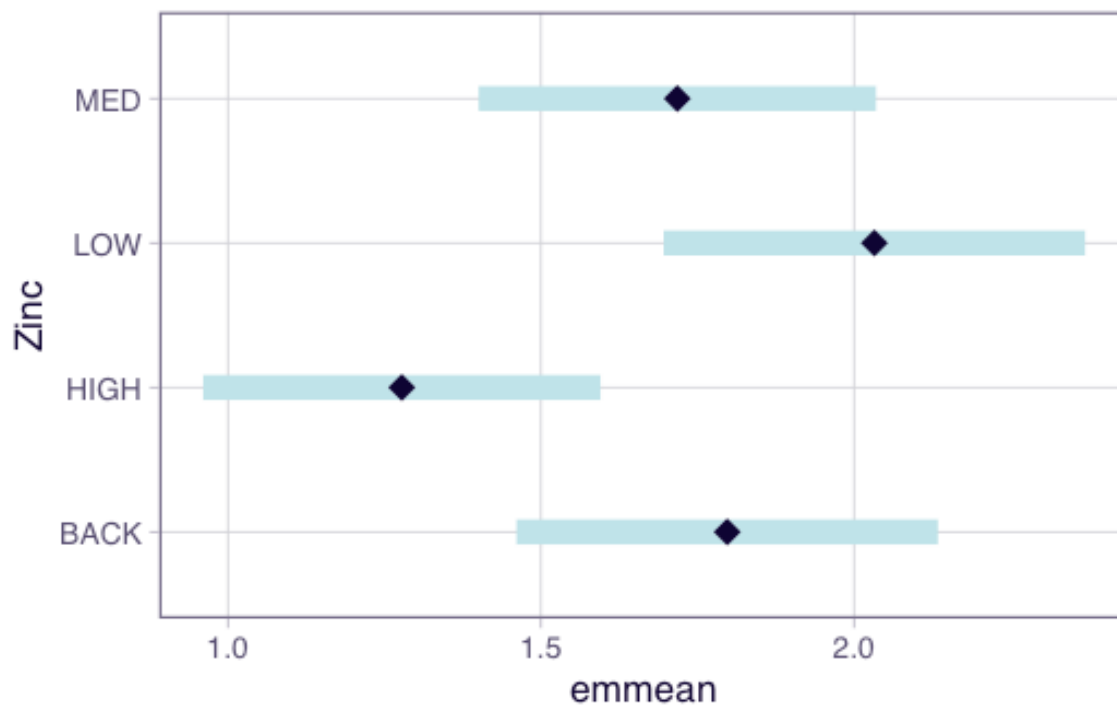
The output of the `emmeans` function gives us the estimated marginal means for each level of Zinc and the 95% confidence intervals. The confidence intervals for the HIGH level of Zn concentration do not overlap with the other levels of Zn concentration, indicating that the HIGH level of Zn concentration is significantly different from the other levels.

Reporting: The post-hoc test indicates that the HIGH level of Zn concentration is significantly different from the other levels of Zn concentration. There are no other significant differences.

We can visualise the results of the post-hoc testing using the `plot` function, if you prefer a visual representation.

CODE

```
plot(posthoc)
```



Exercise 2 – chicks

An experiment was designed to compare 15-day mean comb weights (g) of two lots of male chicks, one receiving sex hormone A (testosterone), the other C (dehydroandrosterone). While we usually analyse these data by using a (pooled) two-sample t-test, a single factor analysis of variance approach could be used (with two levels of the treatment factor). We will compare the results from both analyses.

The data is found in the *Comb* worksheet of the *chick_marigold.xlsx* file. Download it below:

Read the data and save it as *comb*. It should look like below:

CODE

```
comb
```

OUTPUT

```
# A tibble: 22 × 2
  CombWt Hormone
  <dbl> <chr>
1    57 A
2   120 A
3   101 A
```

```
4 137 A
5 119 A
6 117 A
7 104 A
8 73 A
9 53 A
10 68 A
# i 12 more rows
```

Question 3

Perform some checks to verify the two-sample t-test (or one-way ANOVA) is appropriate, i.e. investigate the shape of the distributions, and the standard deviations.

Question 4

In R, perform a 2-sample t-test using the `t.test()` function, and use the `var.equal = TRUE` argument to ensure a standard 2-sample t-test is performed. Interpret the output.

Question 5

Next, perform a one-way ANOVA using the `aov` function, followed by `summary`, and interpret the results.

Question 6

Compare the t-test and ANOVA outputs. What do you notice about:

- i. the degrees of freedom;
- ii. the P-value?

Question 7

When there are only two treatment groups, the observed F and t values are related by $F = t^2$. Demonstrate this for the observed values in this exercise.

Exercise 3 – lambs

Work on this exercise in your own time.

The levels of immunoglobulin (Ig) in blood serum (g/100 ml) in 3 breeds of newborn lambs have been investigated. A total of 44 lambs were sampled, with approximately equal numbers per breed. The researcher wants to know whether or not there are significant differences in immunoglobulin levels between the breeds.

The data is found in `lambs.csv`. Download the data if you have not done so:

Read the data into R and save it as `lambs`. Use the `read_csv()` function which should be available when you load the `tidyverse` package.

It should look like below:

```
CODE
lambs

OUTPUT
# A tibble: 44 × 2
  Ig Breed
<dbl> <dbl>
1  1.1     1
2  2.2     1
3  1.7     1
4  1.4     1
5  1.6     1
6  2.3     1
7  1.4     1
8  1.9     1
9  0.8     1
10 1.6     1
# i 34 more rows
```

Based on the demonstrator walkthrough, analyse the `lambs` dataset and test the hypothesis that:

$$H_0 : \mu_1 = \mu_2 = \mu_3$$

$$H_1 : \text{not all } \mu_j \text{ are equal}$$

where μ_j is the mean Ig level for breed j .

Remember to:

- ☐ State your null and alternate hypotheses.
- ☐ Plot an appropriate summary graph for the data.
- ☐ Demonstrate the model fit using the `aov()` function.
- ☐ Test assumptions, by checking sd values, or by looking at your exploratory plots (checking residuals is not necessary at this point).
- ☐ Report your test statistic, degrees of freedom and p-value.
- ☐ Report the statistical conclusion by addressing the null hypothesis.
- ☐ Explain the results within a biological context to the data.

Thanks!

Did you know you can also knit to PDF? Check the documentation for [R Markdown](#) or [Quarto](#) for more information.

Attribution

This lab was developed using resources that are available under a [Creative Commons Attribution 4.0 International license](#), made available on the [SOLES Open Educational Resources repository](#).

[Click here for session information](#)

CODE

```
sessionInfo()
```

OUTPUT

```
R version 4.5.2 (2025-10-31)
Platform: aarch64-apple-darwin20
Running under: macOS Tahoe 26.2

Matrix products: default
BLAS:   /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versions/A/LibBLAS.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/4.5-arm64/Resources/lib/libRlapack.dylib; LAPACK
version 3.12.1

locale:
[1] en_AU.UTF-8/en_AU.UTF-8/en_AU.UTF-8/C/en_AU.UTF-8/en_AU.UTF-8

time zone: Australia/Sydney
tzcode source: internal

attached base packages:
[1] stats      graphics  grDevices  utils      datasets  methods   base

other attached packages:
[1] emmeans_2.0.0 ggplot2_4.0.2 readxl_1.4.5

loaded via a namespace (and not attached):
 [1] bit_4.6.0      rematch_2.0.0  gtable_0.3.6   jsonlite_2.0.0
 [5] crayon_1.5.3   dplyr_1.2.0    compiler_4.5.2 tidyselect_1.2.1
 [9] parallel_4.5.2 scales_1.4.0   yaml_2.3.12    fastmap_1.2.0
[13] lattice_0.22-7 coda_0.19-4.1  readr_2.1.6    R6_2.6.1
[17] labeling_0.4.3 generics_0.1.4 knitr_1.51     tibble_3.3.1
[21] tzdb_0.5.0     pillar_1.11.1 RColorBrewer_1.1-3 rlang_1.1.7
[25] utf8_1.2.6     xfun_0.56      S7_0.2.1       bit64_4.6.0-1
[29] otl_0.2.0      estimability_1.5.1 cli_3.6.5      withr_3.0.2
[33] magrittr_2.0.4 digest_0.6.39  grid_4.5.2     vroom_1.6.7
[37] xtable_1.8-4   mvtnorm_1.3-3  hms_1.1.4      lifecycle_1.0.5
[41] vctrs_0.7.1    evaluate_1.0.5 glue_1.8.0     farver_2.1.2
[45] cellranger_1.1.0 rmarkdown_2.30 tools_4.5.2    pkgconfig_2.0.3
[49] htmltools_0.5.9
```