Computer Practical 5: Data analysis and manipulation in R

Dr Alice Risely [a.risely@salford.ac.uk](mailto:a.risely@salford.ac.uk)

Table of Contents

[Learning outcomes for today 2](#_Toc182810872)

[Practical 4 learning outcomes 2](#_Toc182810873)

[Preliminary tasks 2](#_Toc182810874)

[Recap: Subsetting data 3](#_Toc182810875)

[Calculating summary statistics per group using dplyr package 3](#_Toc182810876)

[dplyr package 3](#_Toc182810877)

[Summarise by group with the earthworm dataset 4](#_Toc182810878)

[Upload a new dataset on bird counts 5](#_Toc182810879)

[Using geom\_col() instead of geom\_bar() to plot bar plots 7](#_Toc182810880)

[Chi-squared tests 7](#_Toc182810881)

[Formatting data for Chi-squared tests 9](#_Toc182810882)

[One-way chi-squared tests 10](#_Toc182810883)

[Remove zeros 10](#_Toc182810884)

[Q1) Allele frequency in reef damselfish 11](#_Toc182810885)

[Copy data into Excel and import into R 12](#_Toc182810886)

[Two-way Chi-squared tests 17](#_Toc182810887)

[Arrange data 19](#_Toc182810888)

[Make a new column in your dataset 19](#_Toc182810889)

[Remove missing data 20](#_Toc182810890)

[Get help on how a function works 21](#_Toc182810891)

[Install R and R studio software 21](#_Toc182810892)

[Additional resources to continue in R 22](#_Toc182810893)

## Learning outcomes for today

What you will learn in this practical:

* How to upload your own data
* How to summarise data per group using the package **dplyr**
* How to make a bar plot with **geom\_col()** instead of **geom\_bar()**
* How to conduct **one-way** and **two-way chi-squared tests**
* How to **arrange data**
* How to make a **new column**
* Get help on a function
* How to install R and R studio software for long term use of R

## Practical 4 learning outcomes

In last week’s practical you learned:

* How to conduct and interpret a one-way ANOVA: **summary(aov(y~x, data = your\_data))**
* How to conduct and interpret a Tukey’s posthoc test on an ANOVA: **TukeyHSD(aov(y~x, data = your\_data))**
* How to conduct and interpret a Pearson’s correlation: **cor.test(data$x ~ data$y)**
* **Optional:** You also had the option in pracs 3 and 4 to conduct **General Linear Models** using the function **lm()**.

## Preliminary tasks

1. **If you have not completed practicals 1 to 4, you need to do these first.**

2. If you are ready to start Practical 5, **DOWNLOAD the fifth practical worksheet** from Blackboard, do not just view it. To download it, click the “…” next to the file and click ‘Download original file’.

3. **Save this file in an appropriately named folder on your OneDrive** (e.g., “MODULE NAME/WEEK X”) so you can access it at home.

4. **Navigate on your web browser to** [**<https://login.rstudio.cloud/>**](https://login.rstudio.cloud/)**.** Assuming you have already registered for practical 1 to 3, then sign in. If prompted, click “Posit Cloud”.

5. **You should already have an existing R studio project from the first practical called ‘R workshop’,** click on this to enter your saved workspace.

6. **Start a new R script for this practical.** Once your project has loaded (it takes a few seconds), click ‘File’ then ‘New file’, then ‘R Script’.

7. **Press the Save button and save your script as “Practical\_5.R”.** This will save all your code you write today.

## Recap: Subsetting data

In practical 4 you learned how to subset data using the function **subset()**.

Make sure you have your worm\_data loaded from the last week to practice:

# Keep only rows where habitat equals 'Forest' as save as a new dataset  
worm\_data\_forest <- subset(worm\_data, habitat\_type == "Forest")  
  
# Keep only rows where habitat is NOT 'Unknown' and save as a new dataset  
new\_worm\_data <- subset(worm\_data, habitat\_type != "Unknown")  
  
# Keep only rows where worm abundance is under 10 (just look - don't save as anything)  
subset(worm\_data, worm\_abundance<40)

Below are the main operators you need to know when subsetting data:

|  |  |
| --- | --- |
| Operator | Meaning |
| == | Exactly equal to |
| != | NOT equal to |
| < | Less than |
| > | More than |

## Calculating summary statistics per group using dplyr package

In pracs 1-4 you learned and practiced how to do simple summaries of variables using the functions **summary()** and **table()**.

**However, often you want to to generate summaries PER GROUP**. For example, you may want to calculate the mean earthworm abundance per habitat type.

To do this, you need to another R package, called **dplyer**.

## dplyr package

dplyr is a popular R package that speciliases in data manipulation. Like ggplot2, dplyr works by using layers of instructions. **Whilst ggplot2 uses a plus sign (+) to connect layers of instructions, dplyr uses something called a ‘pipe’, which is written as %>%.**

Operators used to connect layers of instructions:

* plotting with ggplot2: **+**
* data manipulation with dplyr: **%>%**

One of the major functions of dplyer is to summarise data per group. Lets practice this in the next section. First you need to install and load the dplyr package.

install.packages("dplyr")

Once you have installed this package, you will need to load it into your environment.

library(dplyr)

## Summarise by group with the earthworm dataset

Lets practice summarising per group using the earthworm dataset you analysed in Prac 4.

The dataset consists of data from 92 x 1m squared plots across different habitat types. At each of these 92 plots, the researchers collected data how how many species of earthworms they found, as well as the total abundance.



If worm\_data is not already in your working environment, upload it again.

Lets summarise the mean earthworm abundance per habitat type. To do this, we use the coding structure:

**data %>%** # take your dataset

**group\_by(X) %>%** # then group by categorical variable X

**summarise(mean\_y = mean(Y))** # then calculate the mean of Y by X

worm\_data %>% # take worm\_data  
 group\_by(habitat\_type) %>% # group by habitat type  
 summarise(max\_abundance = max(worm\_abundance)) # return the max of worm\_abundance (per habitat type) and call new column 'max\_abundance'  
  
# max number of worms found in farmland = 130, forest = 149, and garden = 150

In the example above, I have used the max() function to calculate the maximum for each category of habitat\_type. However, you can change this to any function you like.

**You will need to refer to this table for the following exercises:**

|  |  |
| --- | --- |
| Summary statistic | Function |
| Maximum | max() |
| Minimim | min() |
| Mean | mean() |
| Median | median() |
| Standard deviation | sd() |
| Sum | sum() |
| Number of observations | n() |
| Number of unique observations | n\_distinct() |
| Keep only unique observations | unique() |

You should note that ‘**max\_abundance**’ used in the example above is **just a name I chose for the new column that contains the summary data** - in this case the maximum values per habitat type. You can change this to anything you like, for example:

worm\_data %>%   
 group\_by(habitat\_type) %>%   
 summarise(MAXIMUM\_WORMS = max(worm\_abundance)) #<-- note change here

Notice how the new column in the output now says MAXIMUM\_WORMS.

#### Exercise 1

Edit the code above to calculate the **minimum** worm abundance per habitat type.

\*\* Write your answer below.   
  
  
---

#### Exercise 2

Calculate the mean **worm diversity** per habitat type.

\*\* Write your answer below.   
  
  
---

You can calculate summaries for more than one variable by adding multiple variables to the group\_by() function.

For example, lets calculate the total sum of rainfall per habitat type AND month:

worm\_data %>%   
 group\_by(habitat\_type, month) %>%   
 summarise(total\_rainfall = sum(rainfall\_mm))

## Upload a new dataset on bird counts

We will now analyse a simple dataset on bird diversity. This dataset **contains data on the presence or absence of 10 indicators species across three habitats:** farmland, forest and gardens. If the species was seen in the habitat, it’s presence was maked as ‘1’, and if it was not seen, its presence was marked as ‘0’.

**GOAL: You want to know if there are significant differences in bird diversity seen between the three habitats.**

A bird standing on a basket

Description automatically generated

#### Exercise 3

Download the dataset (*bird\_data.csv*) from Blackboard and upload it into R.

Lets have a look at the dataset using the usual exploratory functions.

View(bird\_data) # check out dataset in pane  
  
str(bird\_data) # data types for each column  
  
head(bird\_data) # first 6 column  
  
dim(bird\_data) # 113 rows, and 5 columns (sample size = 113)  
  
unique(bird\_data$species) # what birds were observed/looked for?

#### Exercise 4

What does each row (= observation) represent in this dataset? As individual bird? A unique species? A unique species seen (or not seen) per habitat?

\*\* Write your answer below \*\*  
  
  
  
---

Always know what each row represents. In your assignment data, each row in an individual blackbird. In the worm data, each row was a plot (= quadrat) that was sampled for earthworms. In the tiger data (from prac 3) each row represented an individual tiger. Depending on how your data is formatted, you will need either n(), n\_distinct() or sum() to generate diversity data.

#### Exercise 5

Use the dplyr package to calculate the number of species seen per habitat. Which habitat has the most species, and which habitat has the least species?

Guidance: the easiest way to do this would be to use the function sum().

**data %>% group\_by(?) %>% summarise(? = sum(?))**

Alternatively, you could first subset the data to remove any zeros, then summarise the number of distinct species per habitat.

\*\* Write your code and answer here \*\*  
  
  
---

## Using geom\_col() instead of geom\_bar() to plot bar plots

So far you have learned to use geom\_bar() to make bar plots. This is because bar plots summarise how many rows you have per category.

* **geom\_bar()** = automatically counts the number of rows you have per categorical group
* **geom\_boxplot()** = compare the distributions of a numerical variable (i.e. a column in your dataset) across categorical groups

However, if you have **already summarised how many rows you have per category**, then geom\_bar will not work anymore, as you want to specify a y axis.

**geom\_col() allows you to make a bar plot but specify a y axis.**

* **geom\_bar()** = specify only the x axis; it automatically counts number of rows per category for y axis
* **geom\_col()** = specify both x axis and y axis, as you have already summarised number of rows per category

Lets plot species diversity for each habitat by summing up the number of birds present in each habitat, using the n() function (which just summarises number of rows per group).

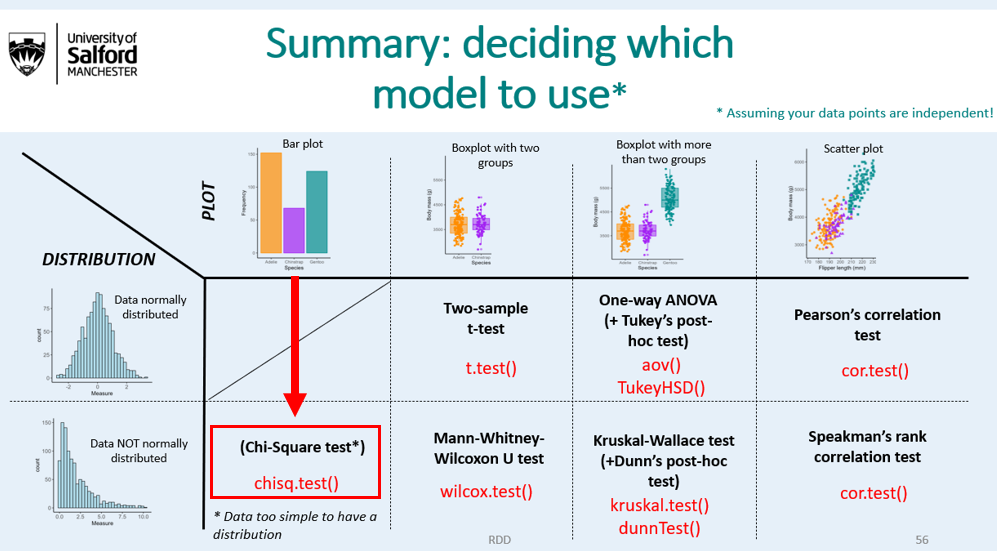
# save our results as a new object called bird\_data\_diversity  
  
bird\_data\_diversity <- bird\_data %>%   
 group\_by(habitat\_type) %>%  
 summarise(diversity = sum(presence))  
  
bird\_data\_diversity  
  
# plot with geom\_col  
  
ggplot(bird\_data\_diversity, aes(x = habitat\_type, y = diversity))+  
geom\_col(fill = "cadetblue")  
  
# note, what happens if we use geom\_bar with the original data?  
  
ggplot(bird\_data, aes(x = habitat\_type))+  
geom\_bar(fill = "pink")

## Chi-squared tests

In practicals 1-4 you have learned the basic of data analysis in R, and learned how to conduct and interpret statistical tests that look at **differences in means between categorical groups** (e.g., t-tests, ANOVAs), and those that test for **correlations between two numerical variables** (e.g., Pearson’s correlation). You also had the option to conduct more advanced multi-variate models, General Linear Models.

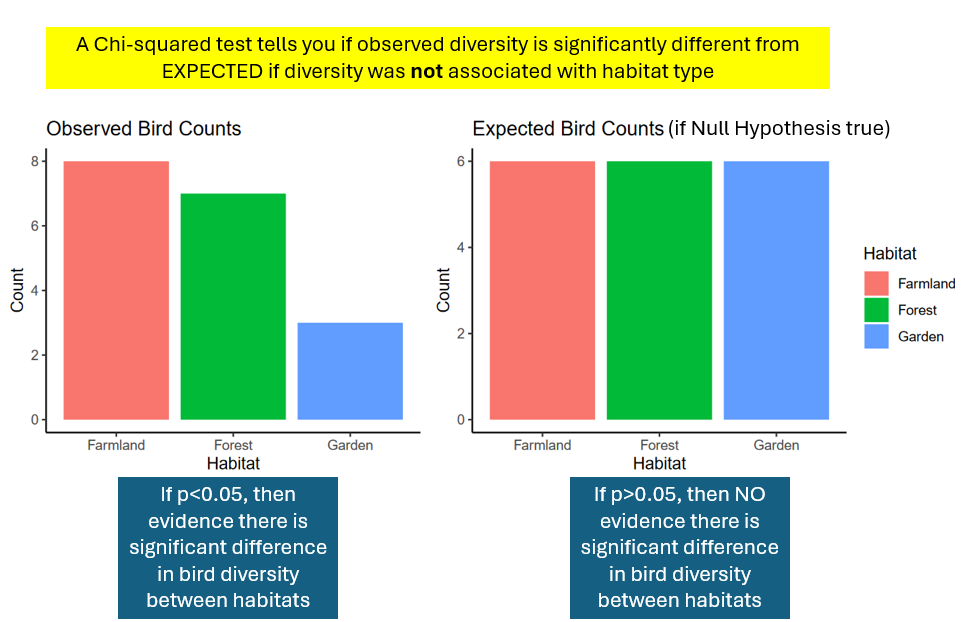
In this practical you will learn about **Chi-squarest tests**, which tests for differences in expected **frequencies** (or counts) between groups.

Above you made a barplot visualising bird diversity across the three habitat types. Since you can only make a bar plot, then the only statistical test you can do on this data is a Chi-squared test:



**Since you only have three numbers of work with (8, 7, and 3), you can’t make a histogram of your data, or do a shapiro test. You just have too few data points!**

A chi square test tests whether the observed counts across habitats (farmland = 8, woodland = 7, garden = 3) are significantly different to what would be expected due to chance (total 18 species divided by 3 habitats = 6):



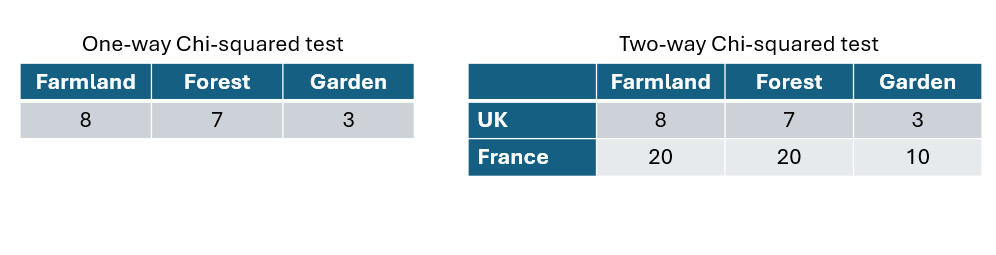
However, the exact code depends on how you have formatted the data, and whether you are performing a **one-way** or **two-way** chi squared test.

* **One way chi squared test**: if you want to know if frequencies differ across just ONE variable
* **Two-way chi-squared test**: If you want to know if frequencies differ across TWO variables.

### Formatting data for Chi-squared tests

Unlike all the other statistical tests you’ve done so far (and will do in the future), where you have one row per observation, the data format for a chi-square test is unique. This is because it is styled on what you would do in Excel, and indeed many people just do chi-squared tests in Excel.

The format generally needed for a hypothetical one-way and two-way chi-squared tests is below:



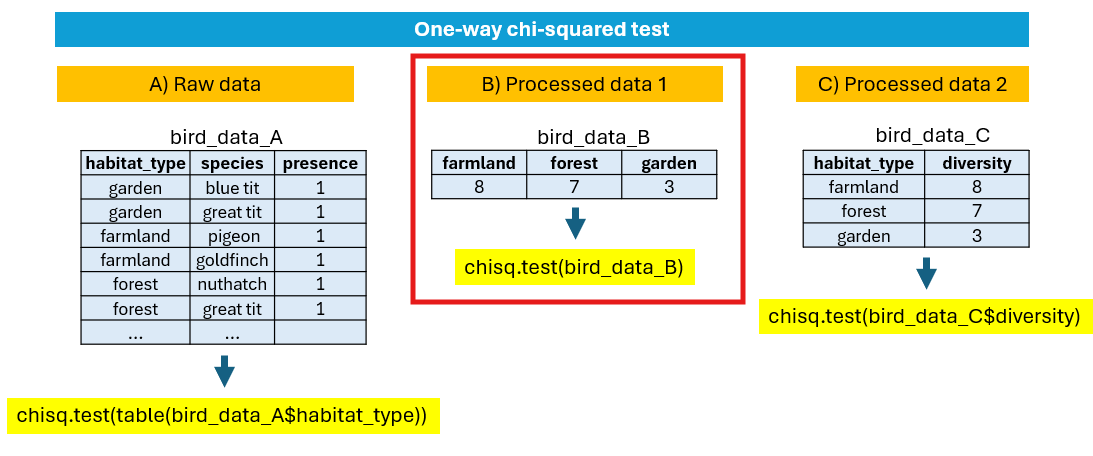
Note for the hypothetical two-way Chi-squared test you have an additional variable to consider: Country.

However, the problem with this is that in real life, you rarely are able to make a nice frequency table without first manipulating the raw data into the right format. Usually, you need to manipulate your raw data first. We are therefore going to do this in R.

### One-way chi-squared tests

In our bird example, we want to know if bird diversity differs between habitat type (just one variable). **So we need a one-way chi-squared test.** Notice the three different data formats for count data below and the code to perform a one-way chi-squared test.

**chisq.test()** would preferably like the data in the middle format in the picture below, but you can still use other data formats if you manipulate them first:



### Remove zeros

Before going further, lets remove species that were not seen from our dataset, so that each row represents a unique species seen per habitat.

bird\_data\_present <- subset(bird\_data, presence == 1)  
  
bird\_data\_present %>%   
 group\_by(habitat\_type) %>%  
 summarise(total = n\_distinct(species))

#### Exercise 6

Run the three lines of code below and work out the data format of the three outputs (A, B or C from the picture above) of each one.

bird\_data\_present  
  
table(bird\_data\_present$habitat\_type)  
  
bird\_data\_diversity

\*\* Select the correct format \*\*  
  
bird\_data\_seen = Format A/B/C?  
  
table(bird\_data$habitat\_type) = Format A/B/C?  
  
bird\_data\_diversity = Format A/B/C?  
  
---

Lets plot a barplot again and perform a chi-squared test to see if there is significant difference in diversity between three habitats. Notice how the format of the data changes how we would code the bar plot AND the chi squared test.

head(bird\_data\_present) # one row per species per habitat  
  
ggplot(bird\_data\_present, aes(x = habitat\_type)) + geom\_bar(fill = "pink")  
  
chisq.test(table(bird\_data\_present$habitat\_type))

You could have also done a chi-squared test on the summarised dataset:

bird\_data\_diversity  
  
ggplot(bird\_data\_diversity, aes(x = habitat\_type, y = diversity)) + geom\_col(fill = "pink")  
  
chisq.test(bird\_data\_diversity$diversity)

Note that the answer is exactly the same in both cases.

IMPORTANT: **chisq.test(table(bird\_data\_present$habitat\_type))** only works because the number of rows per habitat is the same as the species diversity per habitat. If this was NOT the case (e.g., if you had not removed the zeros), doing a chi-squared test on your raw data would not be appropriate.

#### Exercise 7

Based on the chi-squared test above, write a results section on whether there is a significant difference in bird diversity between the three habitat, including the relevant information in brackets (test name, effects size, and p value).

\*\* Write your results section here \*\*  
  
  
---

## Q1) Allele frequency in reef damselfish

Next we you going to learn how to do a **two-way chi squared test.**

You are a fish biologist studying gene frequencies linked to colour polymorphism in relation to coral health in tropical reef ecosystems.

You study a species of damsel fish that can either be blue, yellow, or white depending on what dominant allele is it carrying (A, B, or C).

*Theory: You think that coral bleaching due to climate change may be changing the allele frequency of fish to promote white/pale variants, because white morphs would be better camouflaged against bleached corals and would therefore have a survival advantage.*

* **Allele A = mainly blue colouration**
* **Allele B = mainly yellow colouration**
* **Allele C = mainly white/pale colouration**



**Methods:** You collect data across two sites - one site is a healthy (colourful) coral reef, and another site has a lot of coral bleaching. You collect 30 damsel fish at each of the two sites and record which allele they have.

Lets go through our normal data analysis steps.

#### Exercise 8

Step 1: Write your null and alternative hypotheses using IF/THEN statement. Note - it’s usually easier to write the alternative hypothesis first.

\*\* Write your hypotheses below \*\*  
  
  
  
---

## Copy data into Excel and import into R

#### Exercise 9

* Copy the data below and paste it into excel.
* Save the file as a .csv file, not an excel file (.csv files are more simple text files and do not have multiple tabs on the bottom).
* Save the file as **fish\_data.csv** in an appropriately named OneDrive folder.
* Import into R using either the upload and import buttons on the right, or by running:

fish\_data<- read.csv(“my\_pathway/fish\_data.csv”)

Making sure the pathway is linked to where you saved the data.

**NOTE: if you save your data as an excel file (.xlxs), then this is still ok, but make sure you only have one tab and you have to import it via the ‘Import dataset’ button and then selecting the ‘From Excel’ option on the menu.**

|  |  |  |  |
| --- | --- | --- | --- |
| fish\_ID | reef\_type | allele | Count |
| 1 | Healthy | Allele B (Yellow) | 1 |
| 2 | Healthy | Allele B (Yellow) | 1 |
| 3 | Healthy | Allele B (Yellow) | 1 |
| 4 | Healthy | Allele A (Blue) | 1 |
| 5 | Healthy | Allele A (Blue) | 1 |
| 6 | Healthy | Allele B (Yellow) | 1 |
| 7 | Healthy | Allele B (Yellow) | 1 |
| 8 | Healthy | Allele A (Blue) | 1 |
| 9 | Healthy | Allele A (Blue) | 1 |
| 10 | Healthy | Allele B (Yellow) | 1 |
| 11 | Healthy | Allele C (White) | 1 |
| 12 | Healthy | Allele B (Yellow) | 1 |
| 13 | Healthy | Allele A (Blue) | 1 |
| 14 | Healthy | Allele C (White) | 1 |
| 15 | Healthy | Allele A (Blue) | 1 |
| 16 | Healthy | Allele C (White) | 1 |
| 17 | Healthy | Allele A (Blue) | 1 |
| 18 | Healthy | Allele C (White) | 1 |
| 19 | Healthy | Allele B (Yellow) | 1 |
| 20 | Healthy | Allele B (Yellow) | 1 |
| 21 | Healthy | Allele A (Blue) | 1 |
| 22 | Healthy | Allele A (Blue) | 1 |
| 23 | Healthy | Allele B (Yellow) | 1 |
| 24 | Healthy | Allele B (Yellow) | 1 |
| 25 | Healthy | Allele C (White) | 1 |
| 26 | Healthy | Allele A (Blue) | 1 |
| 27 | Healthy | Allele B (Yellow) | 1 |
| 28 | Healthy | Allele A (Blue) | 1 |
| 29 | Healthy | Allele A (Blue) | 1 |
| 30 | Healthy | Allele A (Blue) | 1 |
| 31 | Bleached | Allele C (White) | 1 |
| 32 | Bleached | Allele B (Yellow) | 1 |
| 33 | Bleached | Allele C (White) | 1 |
| 34 | Bleached | Allele B (Yellow) | 1 |
| 35 | Bleached | Allele A (Blue) | 1 |
| 36 | Bleached | Allele C (White) | 1 |
| 37 | Bleached | Allele B (Yellow) | 1 |
| 38 | Bleached | Allele C (White) | 1 |
| 39 | Bleached | Allele C (White) | 1 |
| 40 | Bleached | Allele C (White) | 1 |
| 41 | Bleached | Allele C (White) | 1 |
| 42 | Bleached | Allele C (White) | 1 |
| 43 | Bleached | Allele C (White) | 1 |
| 44 | Bleached | Allele C (White) | 1 |
| 45 | Bleached | Allele B (Yellow) | 1 |
| 46 | Bleached | Allele C (White) | 1 |
| 47 | Bleached | Allele A (Blue) | 1 |
| 48 | Bleached | Allele B (Yellow) | 1 |
| 49 | Bleached | Allele B (Yellow) | 1 |
| 50 | Bleached | Allele C (White) | 1 |
| 51 | Bleached | Allele B (Yellow) | 1 |
| 52 | Bleached | Allele C (White) | 1 |
| 53 | Bleached | Allele A (Blue) | 1 |
| 54 | Bleached | Allele A (Blue) | 1 |
| 55 | Bleached | Allele C (White) | 1 |
| 56 | Bleached | Allele A (Blue) | 1 |
| 57 | Bleached | Allele A (Blue) | 1 |
| 58 | Bleached | Allele C (White) | 1 |
| 59 | Bleached | Allele A (Blue) | 1 |
| 60 | Bleached | Allele B (Yellow) | 1 |

You should now have fish\_data in your working environment. Run the following code and check the output looks the same.

head(fish\_data)

## fish\_ID reef\_type allele count  
## 1 1 Healthy Allele B (Yellow) 1  
## 2 2 Healthy Allele B (Yellow) 1  
## 3 3 Healthy Allele B (Yellow) 1  
## 4 4 Healthy Allele A (Blue) 1  
## 5 5 Healthy Allele A (Blue) 1  
## 6 6 Healthy Allele B (Yellow) 1

dim(fish\_data)

## [1] 60 4

str(fish\_data)

## 'data.frame': 60 obs. of 4 variables:  
## $ fish\_ID : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ reef\_type: chr "Healthy" "Healthy" "Healthy" "Healthy" ...  
## $ allele : chr "Allele B (Yellow)" "Allele B (Yellow)" "Allele B (Yellow)" "Allele A (Blue)" ...  
## $ count : int 1 1 1 1 1 1 1 1 1 1 ...

#### Exercise 10

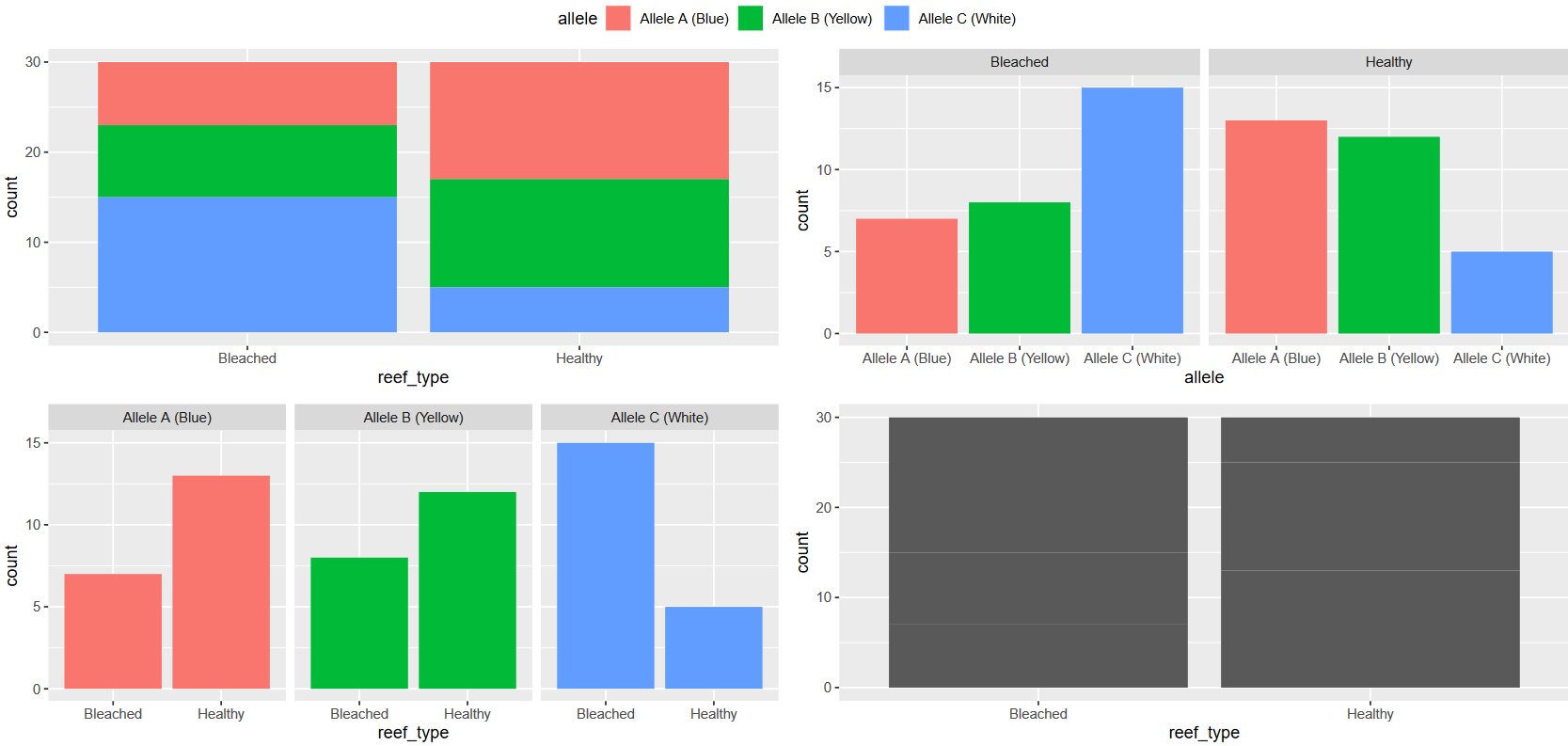
Each row in this dataset represents an individual fish that was caught and genotyped. If you wanted to summarise how many fish had each allele per habitat, which functions out of the bellow would you need? (two would work).

|  |  |
| --- | --- |
| Summary statistic | Function |
| Maximum | max() |
| Minimim | min() |
| Mean | mean() |
| Median | median() |
| Standard deviation | sd() |
| Sum | sum() |
| Number of observations | n() |
| Number of unique observations | n\_distinct() |
| Keep only unique observations | unique() |

\*\* Write the function below \*\*  
  
  
  
---

#### Exercise 11

Out of the plots below, which one do you think visualises the data the best, given your ultimate goal of understanding whether the distribution of alleles is different between bleached and healthy habitats? All plots are theoretically correct, but some are better than others at visualising the data.



\*\* Write your answer below \*\*  
  
Top left?  
Top right?  
Bottom left?  
Bottom right?  
  
---

#### Exercise 12

**Step 2: Plot the number of fish with alleles A, B and C across the two habitats. Try and make the plot you chose above.**

**Guidance:** There are a number of ways you can do this, but first you need to use dplyr package to summarise how many fish have each allele in each habitat. You want as an output the following table:

|  |  |  |
| --- | --- | --- |
| **Reef\_type** | **Allele** | **Count** |
| Healthy | A | ? |
| Healthy | B | ? |
| Healthy | C | ? |
| Bleached | A | ? |
| Bleached | B | ? |
| Bleached | C | ? |

Because each row represents one fish, you just need to count the number of rows per reef type and per allele using **n() or sum().** You should identified one of these as the correct function in Exercise 10 above.

Use the following code to summarise your allele frequencies by group(s) and make a barplot.

**allele\_freq <- fish\_data %>% group\_by(? , ?) %>% summarise(total = sum(count))**

OR

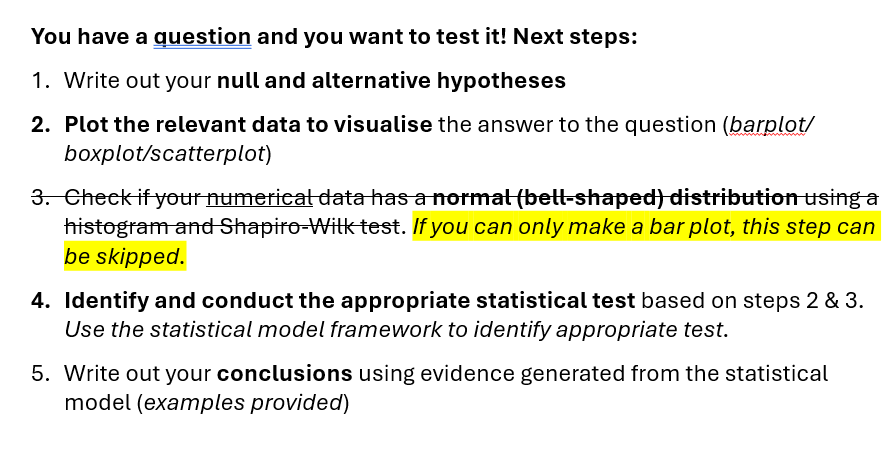
**allele\_freq <- fish\_data %>% group\_by(? , ?) %>% summarise(total = n())**

You then can use geom\_col() to make a bar plot:

**ggplot(allele\_freq, aes(x = ?, y = ?, fill = ?)) + geom\_col() + facet\_wrap(~?)**

**Step 3: Test your numerical data for normality.**

Nope - you don’t need to do this as if you have made any sort of bar plot, you don’t have enough data to test for normality:



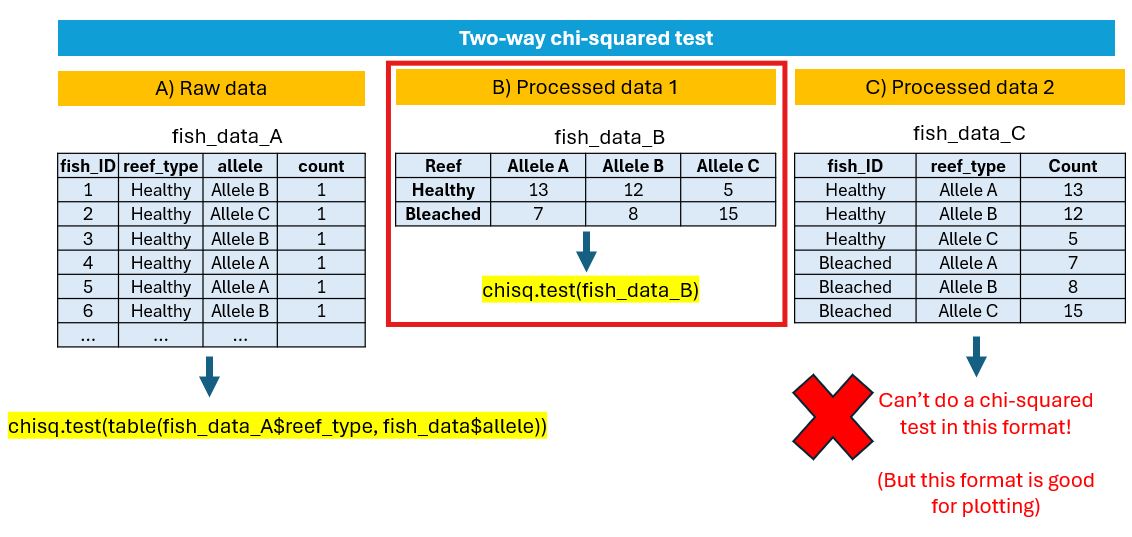
You only need to check if your numerical data has a normal distribution or not is you have made a boxplot or a scatter plot for step 2.

## Two-way Chi-squared tests

You have made a barplot, which means you will need a chi-squared test. But do you need a one-way chi-squared test of a two-way chi-squared test?

Unlike with the bird example, where you just linked species counts to habitat type, in this example you want to link allele counts to reef type (healthy/bleached) AND allele variant (A, B, or C). **So you need a two way chi-squared test.**

See the picture for how to do this. As with a one-way Chi-squared test, ideally you want the data to be in the middle format first.



#### Exercise 13

Conduct a two-way Chi-square test on your data using the following code. Note: Lets save our chi-squared model as an object called ‘Xsq’ so we can look at it in more depth.

fish\_data\_B <- table(fish\_data$reef\_type, fish\_data$allele)  
  
fish\_data\_B   
  
(Xsq <- chisq.test(fish\_data\_B)) # save model as 'Xsq'

One of the useful thing about saving models as objects is that you can look under the hood. In this case, the chi-squared model contains information on not only the observed data, but also the expected data distribution if the null hypothesis was true:

str(Xsq)  
  
Xsq$observed  
Xsq$expected

#### Exercise 14

How many blue, yellow, and white morphs would you expect in each reef type if the null hypothesis was true?

\*\* write your answer here \*\*  
  
  
  
---

#### Exercise 15

Your original goal was to identify whether pale variants (allele C) increased in bleached habitats. Write 2-3 sentences, drawing from your plots and chi-squared test, on what you conclude.

\*\* Write your results here \*\*  
  
  
  
  
---

## Arrange data

Sometimes it is helpful to arrange your data so you can easily see the smallest or largest values. To do this, you need the **arrange()** function in **dplyr.** Lets practice this with the worm dataset.

Lets arrange the worm data by both month and worm abundance.

# arrange by month and then worm abundance, both arranged smallest to largest  
  
worm\_data %>% arrange(month, worm\_abundance)   
  
# arrange by month and then worm abundance, with worm abundance arranged largest to smallest (only view first 6 rows)  
  
head(worm\_data %>% arrange(month, -worm\_abundance))

#### Exercise 16

Arrange the dataset by habitat and then worm diversity. From looking at your rearranged data, what is the highest worm diversity in Farmland habitat?

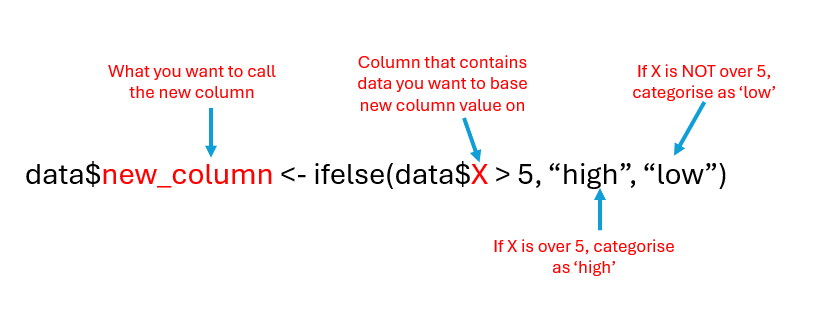
\*\*Write your answers below\*\*  
  
  
  
---

## Make a new column in your dataset

Lets make a new column that categorises whether worms are present or not at the plot. Since all plots found at least some worms, we can just put a ‘1’ in all rows of our new column.

worm\_data$presence <- 1 # Make a new column called 'Presence' and make all values 1  
  
head(worm\_data)

However, in most cases we want to make a new column that is dependent on the value of another column. In this case it is often useful to use the function ifelse().



Lets make a new column that categorises whether the number of birds is above or below average.

head(worm\_data)  
  
mean(worm\_data$worm\_abundance) # return the mean count  
  
  
worm\_data$High\_or\_low <- ifelse(worm\_data$worm\_abundance > 95, "High", "Low")  
  
  
head(worm\_data)# notice new column on the right

Lets make another new column that categorises whether the plot is in farmland or not:

worm\_data$farmland <- ifelse(worm\_data$habitat\_type=="Farmland", "yes", "no")  
  
head(worm\_data, 20)

#### Exercise 17

Make a new column in your dataset called ‘high\_diversity\_site’, and this column represents whether the plot had higher species diversity than average. If the site has higher diversity than average, the new column should have a ‘yes’, and if lower than average, it should have a ‘no’. Check your column is correctly formatted by looking at the new column.

\*\* Write your code here \*\*  
  
  
---

## Remove missing data

Datasets often have missing data, which can interfere with analyses. If you want to remove missing data, I have added how to do this below using either **na.omit()** or the **subset()** function.

First, lets upload the penguin dataset from the palmerpenguin package, as this has missing data.

library(palmerpenguins)  
  
data(penguins)  
  
head(penguins) # notice missing data in row 4

# remove all rows with any missing data anywhere  
penguins1 <-na.omit(penguins)  
  
# remove rows with missing data just in body\_mass\_g column  
  
penguins2 <- subset(penguins, !is.na(body\_mass\_g))

## Get help on how a function works

Not sure how a function works? Use the help function, or ?function().

?t.test() # get a help page for the function t.test

A help page will come up which will tell you what arguments you need to put inside the brackets.

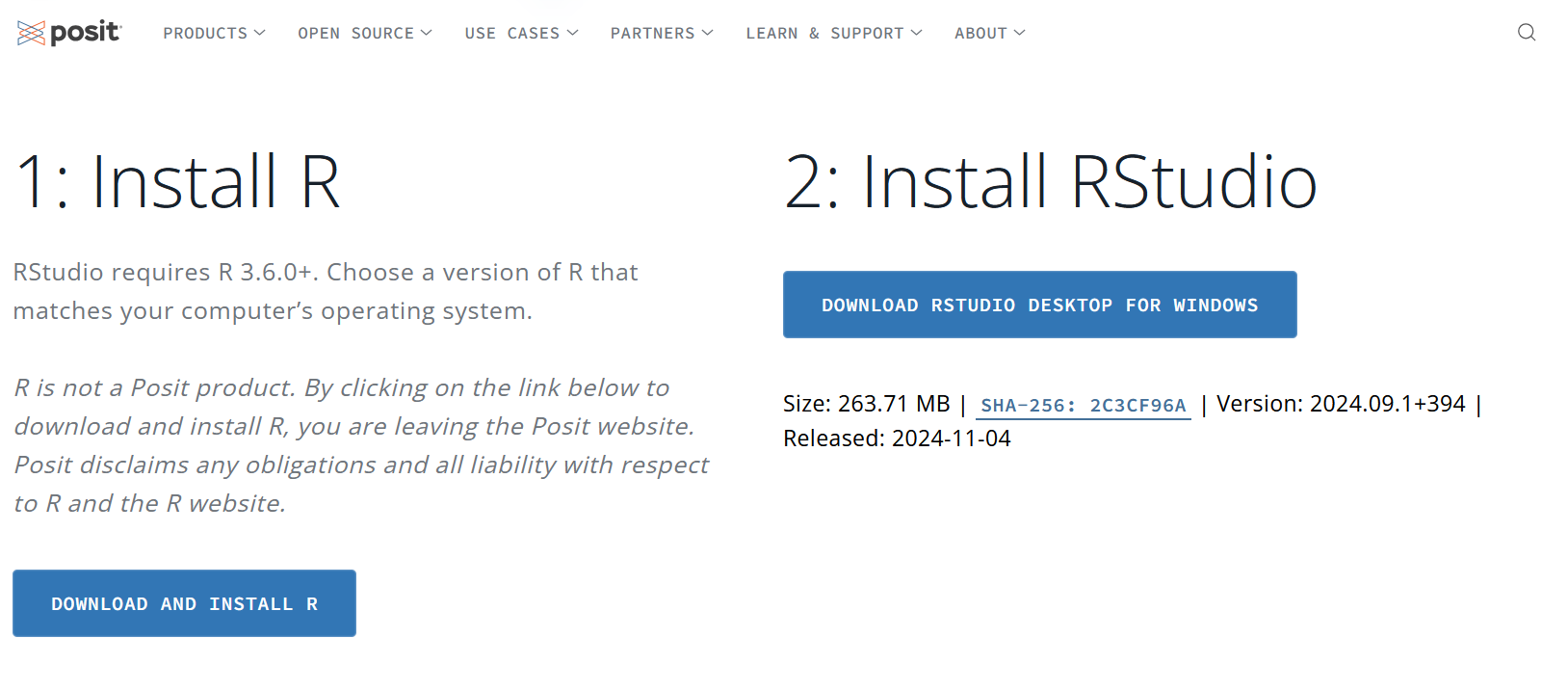
**Scroll down to find “Examples” and copy and paste the code into a new R script.** This will provide working examples of how the function works.

## Install R and R studio software

Important: The free version of R studio cloud only provides so many free hours per month. If you think you will exceed this, it is worth downloading R and R studio software. The software looks almost identical to the cloud version you have been using.

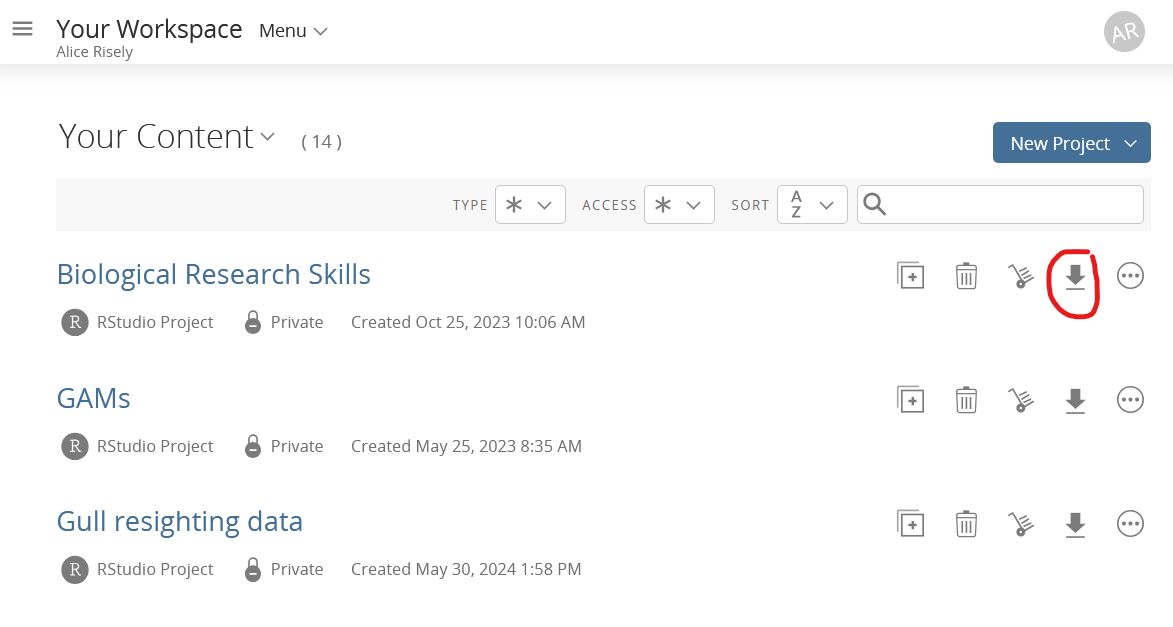
To install R Studio, you need to install both:

* R <https://cran.r-project.org/>
* R Studio <https://posit.co/download/rstudio-desktop/> (you can do both steps from this link).



R is the R language software, whilst R Studio is the user interface. Once you have downloaded both, you only ever need to open **R Studio**, not R.

**If you want to keep all the work you’ve done so far, you can download your whole R Studio cloud project on to your computer by pressing the ‘Export’ button next to the project you want to download (which should be called “R workshop”), and then open it with your R Studio software.**



This will download all your scripts and R data in one file, and you can open it by clicking on the downloaded R studio project icon, which should be called “**R workshop.Rproj**”.



**NOTE: You can just code in R, without using the user interface R Studio.** If you are doing Joe Jackson’s Data Science module next semester, you may use R without R Studio. It is your choice - if you feel more comfortable using R studio, please use it.

## Additional resources to continue in R

There are countless resources out there to learn R. Everyone needs to proactice in their own time if they are to get confident. This course is just a taster. Here are a few links to other resources:

R Studio basics: <https://posit.cloud/learn/recipes>

R Cookbook: <https://rc2e.com/>

R for Data Science: <https://r4ds.hadley.nz/>

Plotting with ggplot2: <https://r-graph-gallery.com/ggplot2-package.html>