Lab Two – Basic Tasks in R

- Use `R` (R Core Team 2025) to complete the tasks below. Make sure to start your solutions in on a new line that starts with "**Solution**:".
- Make sure to use the Quarto Cheatsheet. This will make completing and writing up the lab *much* easier.

1 Libraries and Packages

1.1 Part a

Install the `esquisse` package for `R`. Report the code below, ensuring to set the `eval` option to `false`, using "#| eval: false", so that we don't install the package every time you compile your Quarto document.

1.2 Part b

Load the `esquisse` package for `R`. Report the code below. It does not matter whether you set "#| eval: false" because loading the package does not add a lot of time. Note you will not evaluate any code that uses the `esquisse` package, so there is no need to load it as you compile your Quarto document.

1.3 Part c

How can you ask for help about the `esquisse` package for `R`? Report the code below, ensuring to set the `eval` option to `false`, using "#| eval: false", so that the help document isn't loaded every time you compile your Quarto document.

1.4 Part d

Is a demo or vignette available for the `esquisse` package for `R`? Report the code below, ensuring to set the `eval` option to `false`, using "#| eval: false", so that the you don't get errors when you compile your Quarto document.

1.5 Part e

Add the BibTeX citation for `esquisse` package for `R` to your `.bib` file and cite it in a sentence describing what the package does below. Note you can reference a citation named `esquisse` using "[@esquisse]".

1.6 Part f

Run `esquisser()` in your console – not in a chunk of `R` code in your Quarto document.

i. Select `palmerpenguins` from the "Select an environment in which to search:" dropdown. This should automatically select `penguins` in the "Select a data.frame:" dropdown. Click "Import Data".
ii. Drag `body_mass_g` to the X box and `species` to the fill box.
iii. Describe what you see in words. What can you conclude about Adelie, Chinstrap, and Gentoo penguins based on the resulting plot?

2 Objects and Vectors

Create the following vectors in `R`. In some cases, you may be able to use `seq()` or `rep()` while in others you cannot. Use these functions when possible, otherwise manually create the vector and explain why that was necessary.

Some Snowday Notes

There are three ways we will create a vector. Most simply, we can create one from scratch. For example, I create a vector of odds, and evens less than 10 below.

```
(odds <- c(1, 3, 5, 7, 9))
```

```
[1] 1 3 5 7 9
```

```
(evens <- c(2, 4, 6, 8))
```

```
[1] 2 4 6 8
```

There are also built-in functions like `seq(...)` for doing this:

```
(odds <- seq(from=1, to=9, by=2))
```

```
[1] 1 3 5 7 9
```

```
(evens <- seq(from=2, to=8, by=2))
```

```
[1] 2 4 6 8
```

Either approach is easy enough with a small number of elements, but what if I wanted odds less than 100? 1000? 1 million? The `rep(...)` and `seq()` approaches would be far more efficient.

We can also create repeating sequences by hand

```
(repeating.seq1 <- c(1, 2, 3, 1, 2, 3, 1, 2, 3))
```

```
[1] 1 2 3 1 2 3 1 2 3
```

```
(repeating.seq2 <- c(1, 1, 1, 2, 2, 2, 3, 3, 3))
```

```
[1] 1 1 1 2 2 2 3 3 3
```

or using a built-in function `rep(...)`

```
(repeating.seq1 <- rep(c(1,2,3), times=3))
```

```
[1] 1 2 3 1 2 3 1 2 3
```

```
(repeating.seq2 <- rep(c(1,2,3), each=3))
```

```
[1] 1 1 1 2 2 2 3 3 3
```

There are some vectors for which we can't use a `seq()` or `rep()`. For example, consider the prime numbers less than 10. The primes are not sequential (e.g., jump by a fixed amount), nor are they repeating.

```
primes <- c(2, 3, 5, 7)
```

Note: There is a `primes` package for `R` (Keyes and Egeler 2025) that contains a `generate_primes(min, max)` function for generating a vector of primes from `min` to `max`.

2.1 Part a

The Fibonacci Sequence is a recursive formula:

$$F_n = F_{n-1} + F_{n-2}$$

where $F_0 = 0$ and $F_1 = 1$.

Create a vector of the first 10 Fibonacci numbers.

2.2 Part b

Triangular Numbers are the cumulative sums of natural numbers:

$$F_n = \frac{n(n+1)}{2}.$$

Create a vector of the first 10 Triangular Numbers.

2.3 Part c

Suppose I were designing a repeated meausres experiment with three treatment conditions. Each of $n = 10$ participants (with `ID` 1 to 10) will receive *all* experimental conditions, call them "Control", "Treatment A", and "Treatment B".

Consider setting up data entry for this experiment.

i. Create a vector containing each `ID` repeated three times, once for each treatment.
ii. Create a vector containing each `Treatment` repeated for each participant ID.

2.4 Part d

Create a vector containing the `character` "MATH" and the `numeric` 240. What is the resulting class? Explain why in a sentence.

3 Solution:

3.1 Install package

```
install.packages("esquisse")
```

3.2 Load package

```
library(esquisse)
```

3.3 Asking for help

```
help(esquisse)
```

3.4 Checking for demo or vignette

```
demo("esquisse")
vignette("esquisse")
```

3.5 Citation

The `esquisse` package has drag-and-drop interface for mapping variables and creating ggplot2 figures without writing a lot of code (Meyer and Perrier 2025).

3.6 Observation and Conclusion

After using `esquisser()` with the `palmerpenguins` dataset, I plotted `body_mass_g` on the x-axis and filled by `species`. Based on the plot, Gentoo penguins are larger and heavier than the other two species. Adelie and Chinstrap penguins are more similar in body mass, though Chinstrap penguins are marginally heavier.

Body mass is strong characteristic that distinguishes Gentoo penguins from Adelie and Chinstrap penguins, while it's less effective at separating Adelie and Chinstrap penguins from each other.

3.7 Fibonacci

```
(fib <- c(0, 1, 1, 2, 3, 5, 8, 13, 21, 34))
```

```
 [1]  0  1  1  2  3  5  8 13 21 34
```

Manually created a vector for 1st 10 fibonacci numbers. Both `seq()` and `rep()` cannot be used here since the numbers are neither sequential nor repeating.

3.8 Triangular numbers

1st 10 triangular numbers.

```
n<-1:10
(nums<-n * (n+1)/2)
```

```
 [1]  1  3  6 10 15 21 28 36 45 55
```

Used the colon operator to find the sequence of numbers from 1 to 10, both inclusive, and used the formula

$$F_n = \frac{n(n+1)}{2}.$$

to find the 1st 10 triangular numbers

3.9 Measures experiment

```
(ID<-rep(1:10,times=3))
```

```
 [1]  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5  6  7  8  9 10  1  2  3  4  5
[26]  6  7  8  9 10
```

```
(Treatment <- factor(
  rep(c("Control", "Treatment A", "Treatment B"), times = 10)
))
```

```
 [1] Control     Treatment A Treatment B Control     Treatment A Treatment B
 [7] Control     Treatment A Treatment B Control     Treatment A Treatment B
[13] Control     Treatment A Treatment B Control     Treatment A Treatment B
[19] Control     Treatment A Treatment B Control     Treatment A Treatment B
[25] Control     Treatment A Treatment B Control     Treatment A Treatment B
Levels: Control Treatment A Treatment B
```

I used `rep()` for both ID and Treatment to repeat the given values the set number of times and used `factor()` for the treatment variables to show that they are categorical.

3.10 Resulting class

```
v<-c("MATH",240)
class(v)
```

```
[1] "character"
```

The resulting class is `character` because `R` convertes the values into the highest level type and since "MATH" is a `character` and is of a higher level than `numeric` 240, it results in class `character`.

References

Keyes, Os, and Paul Egeler. 2025. *Primes: Fast Functions for Prime Numbers*. https://doi.org/10.32614/CRAN.package.primes.

Meyer, Fanny, and Victor Perrier. 2025. *Esquisse: Explore and Visualize Your Data Interactively*. https://doi.org/10.32614/CRAN.package.esquisse.

R Core Team. 2025. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.