Quantitative Skills in Biosciences I

R. E. Treharne

Contents

Preface

This book accompanies the Quantitative Skills (QS) component of the **BIOS103 - Introductory Practical Skills in Biosciences** I course.

If you are participating in the weekly timetabled QS workshops associated with this course then be sure to submit your weekly summative QS assignments via Canvas by the deadline of **5pm on the day of the workshop**.

All QS workshops will be delivered online via Teams. Access the meeting link from the BIOS103 Canvas course.

1 Snails and R

If you are currently participating in a timetabled BIOS103 QS workshop, please ensure that you cover **all of this section's content** and complete the **formative and summative assessments** on Canvas before 5 PM today.

1.1 Estimating the Volume of a Snail

In this section, you will learn how to import data from a CSV file into Excel, perform basic calculations, create a scatterplot including a linear trendline to make predictions. The primary objective is to estimate the volume of a snail based on its mass using a provided dataset.

1.1.1 Download and Import the CSV File

1. **Download the CSV File:**
   - Here is an example dataset. Download it to your local machine.
2. **Import into Excel:**
   - Open Excel (Use the desktop version - you won't be able to do this using the online version!).
   - Go to `Data > From Text/CSV` and select the downloaded CSV file.
   - When the import wizard appears, click `Load`.

1.1.2 Calculate Volume V in Excel

We will estimate the volume V of our snails using the formula for the volume of a sphere:

$$V = \frac{4}{3}\pi r^3$$

where r is the radius, which we assume is equal to half the `Height L (mm)` column.

1. **Add a New Column for Volume V:**
   - In the third column, label it as `Volume V (mm^3)`.
   - In the first cell of this column, use the formula:

     ```
     = (4/3) * PI() * ((B2/2)^3)
     ```

   - Drag the formula down to apply it to all rows.

1.1.3 Add a Linear Trendline and Equation

1. **Create a Scatter Plot:**
   - Select the `Mass M (g)` and `Volume V (mm^3)` columns.
   - Go to the `Insert` tab and select `Scatter Plot`.
2. **Add Trendline:**
   - Click on the plot
   - Click the green `+` icon that appears at the top right of the plot.
   - Click the `>` symbol on the `Trendline` option and click `More options...`.

- From the trendline options menu that appears on the right, select the `Linear` trendline and check the box `Display Equation on chart`.
- Right-click on a data point in the scatter plot.

3. **Interpret the Equation:**

   - The trendline equation will appear on the chart in the form of $y = ax + b$, where:
     - y is the volume.
     - x is the mass.
     - a and b are coefficients.
   - For the Example dataset:
     - a = 1366.6 $mm^3.g^{-1}$
     - b = 1706.4 mmˆ3

1.1.4 Estimation of Volume for a Snail with Mass 10g

1. **Use the Trendline Equation:**

   - Substitute $x = 10$ into the trendline equation to calculate the estimated volume V.
   - Express the volume in cm^3 to one decimal place (Note: $1cm^3 = 1000mm^3$).

2. **For the example dataset:**

   - The estimated volume of a snail that is $10g$ is $15.4cm^3$

1.2 Getting Ready for R

Over the next couple of weeks you will continue to use Excel to load, manipulate, analyse and visualise data. Beyond this you will be using the coding language R exclusively. To prepare for this, you need to download, install and configure R and RStudio today.

1.2.1 Download and Install

`R` and `RStudio` are actually separate things, although they are often mentioned together.

`R` is a programming language and software environment specifically designed for statistical computing and data visualisation. Other examples of programming languages include Python, Java and Ruby.

`RStudio` is the integrated development environment (IDE) for R. It provides a user-friendly interface that will allow you to write all your R scripts and compile them to do stuff. I'm using it to write this handbook right now!

Despite their differences, you might hear the terms `R` and `RStudio` used interchangeably, as RStudio serves as the primary interface through which users interact with the R programming language.

You need to download and install both R and Rstudio.

- Install R first from the Comprehensive R Archive Network (CRAN)
- Then install RStudio from the RStudio website

Once both are installed, open up RStudio and get ready to create your first **R Project**.

DON'T FOGET. All University of Liverpool MWS machines already have R and RStudio installed and ready to use.

1.2.2 Creating Your First R Project in RStudio

Follow these steps to set up and manage your first R project in RStudio:

1. **Open RStudio**

- Launch RStudio from your applications menu.

2. **Start a New Project**

- Click on the `File` menu at the top of the RStudio window.
- Select `New Project...` from the dropdown menu.

3. **Choose Project Type**

You will be prompted with three options:

- **New Directory**: Create a new project in a new directory.
- **Existing Directory**: Use an existing directory as the project's folder.
- **Version Control**: Clone a project from a version control repository (e.g., GitHub).

For your first project, select `New Directory`.

4. **Select Project Template**

- Choose `Empty Project` unless you have a specific project type in mind (e.g., Shiny Web Application, R Package).
- Click `Next`.

5. **Set Up Project Directory**

- **Directory name**: Enter a name for your project folder. This will be the name of the directory created for your project.
- **Subdirectory of**: Choose the parent directory where the new project folder will be created. You can navigate to the desired location using the file browser.
- Click `Create Project`.

6. **RStudio Project Interface**

Once the project is created, you will see a new RStudio window or tab with the following components:

- **Files pane**: Displays the files and folders in your project directory.
- **Script editor**: Where you write and edit your R scripts.
- **Console**: Where you can directly enter and execute R commands.
- **Environment/History**: Shows your workspace objects and command history.
- **Plots/Packages/Help/Viewer**: Various tabs for viewing plots, managing packages, accessing help documentation, and viewing other outputs.

7. **Create and Save an R Script**

- Click `File` > `New File` > `R Script`.
- Write some R code in the script editor. For example:

```
# my first line of code. this is a comment!
print("Hello World!")
```

To run your print command, click on the line and click the `Run` button at the top right of your script editor window or press `Ctrl + ENTER` (Cmd + Enter on Mac).

You will see the following output in your console window:

```
## [1] "Hello World!"
```

And there it is! You've just successfully compiled your first line of R. Congratulations!

1.2.3 Something More Complicated

As you delve deeper into R programming, you'll find that your scripts become more sophisticated than the "Hello World" example above. In the following example, I'll walk you through a script to create a random number generator. Here's the script in its entirety:

```
# create a variable called "seed"
seed = 999

# set the seed
set.seed(seed)

# generate a random number
random_number <- runif(1)

# print the random number
print(random_number)
```

Cut and paste these lines into the script file we were working on earlier (ovewrite the "Hello World" example).

Now, you can run each line in turn as before (using `Ctrl + ENTER`) or you can run everything by clicking the `Source` button. You should see the following number appear in your console:

```
## [1] 0.3890714
```

Let's break down each line of the script to understand its purpose and functionality.

1. **Creating a Variable Called "seed"**

```
# create a variable called "seed"
seed <- 123
```

- `seed <- 123`: Here, we are using the `<-` operator to assign the value 123 to the variable named seed. In R, variables are used to store data that can be reused or manipulated later in the script. The number 123 is arbitrary in this case, but we use it to illustrate how to set a seed for random number generation.

2. **Setting the Seed**

```r
# set the seed
set.seed(seed)
```

- The set.seed() function initializes the R environment's build in random number generator with the value stored in seed. Setting a seed is essential for reproducibility, meaning that if someone else runs this code with the same seed, they will get the same random number output in their console. This is particularly useful in simulations and randomised experiments where consistent results are needed.

3. **Generating a Random Number**

```r
# generate a random number
random_number <- runif(1)
```

- random_number <- runif(1): This line generates a single random number between 0 and 1. The function runif() generates random numbers from a uniform distribution, which means that each number within the specified range has an equal probability of being selected. The 1 inside the parentheses specifies that only one random number should be generated. The resulting number is then assigned to the variable called `random_number`.

4. **Printing the Random Number**

```r
# print the random number
print(random_number)
```

- print(random_number): The print() function outputs the value stored in `random_number` to the console. This is useful for verifying the output of your code and ensuring that the operations have been executed correctly.

Summary

This example script demonstrates a more complex task than the basic "Hello World" script. It introduces key concepts like variable assignment with <-, setting a seed for reproducibility using set.seed(), generating random numbers with runif(), and printing results using print(). As you continue learning R, these foundational concepts will become increasingly important, enabling you to build more advanced and meaningful analyses. Remember, comments (#) are your friends! They help explain what each part of your code does, making it easier for you and others to understand and maintain your scripts. Be liberal with your comments. You'll thank yourself later (trust me).

1.3 Complete your Weekly Assignments

In the BIOS103 Canvas course you will find this week's **formative** and **summative** assignments. You should complete both of these before the end of the online workshop that corresponds to this section's content. The assignments are identical in all but the following details:

- You can attempt the **formative assignment** as many times as you like. It will not contribute to your overall score for this course. Make sure you practice this assignment until you're confident that you can get the correct answer on your own.
- You can attempt the **summative** assignment **only once**. It will be identical to the formative assignment but will use different values and datasets. This assignment **will** contribute to your overall score for this course.
- **Late submissions**. You have until 5pm on the day of the workshop for this week's content to submit your **summative** assignment. Late submissions will incur a 5% penalty for every part or full day beyond the deadline. Penalties will be capped at 40%.

2 Summarising Data and ANOVA in Excel

If you are currently participating in a timetabled BIOS103 QS workshop, please ensure that you cover **all of this section's content** and complete the **formative and summative assessments** on Canvas before 5 PM today.

In this section we will only be using Excel. No R today! You'll have to contain your excitement for a few more weeks yet.

Working on two important skills: + Summarising Data + Constructing and testing hypotheses

The ultimate aim is to gain insight and learn something new about the world from the data that we have painstaking measured in our well designed lab experiments. This is a cornerstone of what being a scientist is all about. This is what we are training you to become.

2.1 Summarising Data

Raw data is beautiful, but messy. Showing another person your raw data and expecting them to immediately understand it, not matter how proud you are of the toil expended to generate the data, is an unrealistic expectation. You need to boil it down into something that another person can grasp instantaneously.

Let's take a look at a Zebrafish dataset from an experiment that is uncannily similar to the one you generated in your lab practical this week.

2.1.1 Download and Import the CSV File

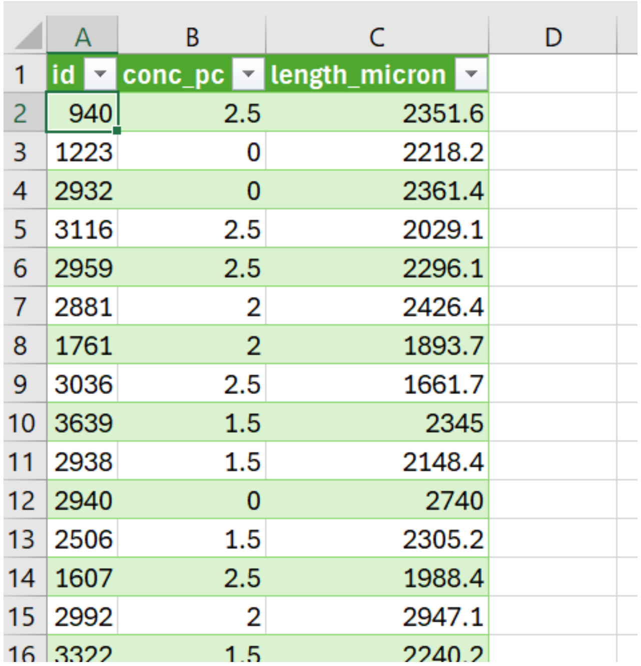
1. **Download the CSV File:**
   - Here is an example dataset. Download it to your local machine.
2. **Import into Excel:**
   - Open Excel (Use the desktop version - you won't be able to do this using the online version!).
   - Go to `Data > From Text/CSV` and select the downloaded CSV file.
   - When the import wizard appears, click `Load`.

You should now see something like that in Figure 1. There are 3 columns: - `ID` - A unique number to identify a measurement. - `conc_pc` - The ethanol concentration (%) that the each embryo was treated with. - `length_micron` - The measured lengths, in μm of the embryos.

We call this format, in which each row corresponds to a single measure mt, a **long** format.

2.1.2 Generating a Summary Table

1. **Identify your Groups**
   - Click anywhere in your table.
   - Select the `Data` menu and click the `Advanced` icon in the `Sort & Filter` section. This will bring up a window called "Advanced Filter".
   - Select the `Copy to another location` action.
   - Your list range should already be set to `$B:$B`, but if not make it so.
   - Set the `Copy to` cell to `$E:$1`
   - Make sure the `Unique records only` check box is selected and click OK.
   - You should now see a complete list of your alcohol concentration groups in a column with a header `conc_pc`. Make this into a new table by clicking on any of the concentration values, and then `Insert > New table > OK`.

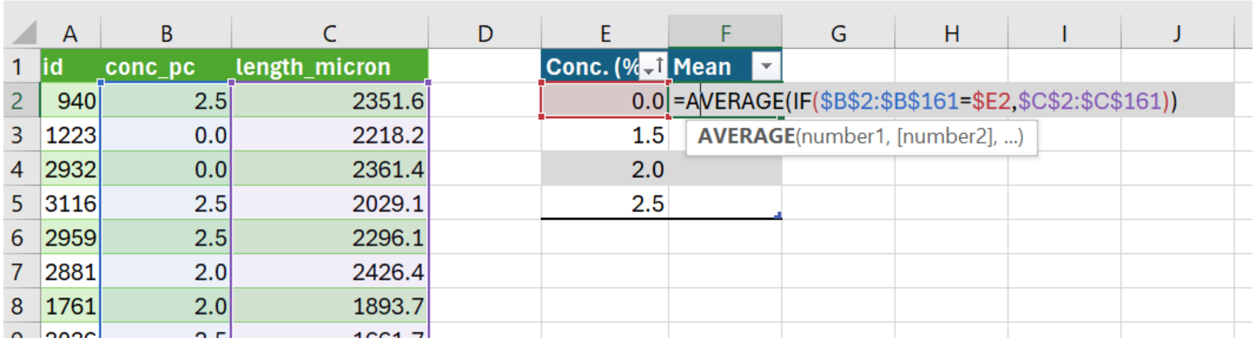


	A	B	C	D
1	id	conc_pc	length_micron	
2	940	2.5	2351.6	
3	1223	0	2218.2	
4	2932	0	2361.4	
5	3116	2.5	2029.1	
6	2959	2.5	2296.1	
7	2881	2	2426.4	
8	1761	2	1893.7	
9	3036	2.5	1661.7	
10	3639	1.5	2345	
11	2938	1.5	2148.4	
12	2940	0	2740	
13	2506	1.5	2305.2	
14	1607	2.5	1988.4	
15	2992	2	2947.1	
16	3322	1.5	2240.2	

Figure 1: Some Figure

Nice. Now you're ready to start building out your summary table horizontally. Let's start with calculating the mean Zebrafish length for each group.

Right now, your spreadsheet should look roughly the same as the screenshot in figure 2



	A	B	C	D	E	F
1	id	conc_pc	length_micron		Conc. (%	Mean
2	940	2.5	2351.6		0.0	=AVERAGE(IF($B$2:$B$161=$E2,$C$2:$C$161))
3	1223	0.0	2218.2		1.5	
4	2932	0.0	2361.4		2.0	
5	3116	2.5	2029.1		2.5	
6	2959	2.5	2296.1			
7	2881	2.0	2426.4			
8	1761	2.0	1893.7			

Figure 2: Constructing a summary table

2. **Calculating a Mean Column**

   - Create a new column in your summary table by typing the word `Mean` in cell `F1`.
   - Calculate the mean of the Zebrafish lengths for the control group (0% alcohol concentration) by entering the following formula into cell `F2`.

```
=AVERAGE(IF($B1:$B161=$E2,$C1:$C161))
```

A Deeper Explanation

The formula `=AVERAGE(IF($B$2:$B$161=$E2,$C$2:$C$161))` is an array formula that calculates the average length of Zebrafish for a specific group based on the concentration of alcohol.

- `$B$2:$B$161`: The `$` symbols before both the column letter `B` and the row numbers `2` and `161` lock the entire range. This means that when you copy the formula to other cells, this range will not change; it will always refer to cells `B2` to `B161`.
- `$E2`: The `$` before the column letter `E` locks the column, but since there's no `$` before the row number `2`, the row number can change if the formula is dragged down across rows. This cell is used to compare each value in the range `$B$2:$B$161` to the specific concentration value in the corresponding row in column `E`.

- `$C$2:$C$161`: Similar to the range for column `B`, this locks the range of cells in column `C` from which the values will be averaged, conditional on the `IF` statement.
- `AVERAGE(IF(...))`: The `IF` function checks each row in the range `$B$2:$B$161` to see if it matches the value in the corresponding row in column `E`. If it matches, the corresponding value in column `$C$2:$C$161` is included in the average calculation. The `AVERAGE` function then calculates the mean of these filtered values.

This approach is particularly useful when you want to calculate conditional averages across a dataset, ensuring that the correct cells are referenced even when copying the formula to different parts of the spreadsheet.

3. **Calculating More Columns**

   - Create four more columns with headers:
     - Std. Dev.
     - Median
     - Min
     - Max
   - Drag the cell `F2` to `G2`. Change the word `AVERAGE` in the formula in `G2` to `STDEV`. This will calculate the standard deviation for the group and the remaining cells in the column should also auto complete.
   - Do the same for the median, min and max columns. Be sure to use the corresponding function.

A Deeper Explanation

When analysing data, it's important to understand the basic statistical measures that summarise the data's distribution. Here are some key terms:

- **Mean**: The mean, often referred to as the average, is the sum of all values in a dataset divided by the number of values. It provides a central value for the data. However, the mean can be influenced by outliers (extremely high or low values).
- **Standard Deviation**: The standard deviation measures the amount of variation or dispersion in a dataset. It is calculated as the square root of the variance, where variance is the average of the squared differences between each data point and the mean. A low standard deviation indicates that the data points tend to be close to the mean, while a high standard deviation indicates more spread out data.
- **Median**: The median is the middle value in a dataset when the values are arranged in ascending or descending order. If the dataset has an odd number of values, the median is the central value. If the dataset has an even number of values, the median is the average of the two central values. The median is less affected by outliers compared to the mean.
- **Min**: The minimum (min) value is the smallest value in the dataset. It provides a measure of the lower bound of the data.
- **Max**: The maximum (max) value is the largest value in the dataset. It provides a measure of the upper bound of the data.

These measures are fundamental for understanding the distribution of data. The mean and median give you central tendencies, while the standard deviation tells you how spread out the data is. The minimum and maximum values provide the range within which all the data points fall.

2.1.3 Presenting Your Summary Table

At some point you may wish to include your Excel summary table in a Word document. There's a lot of wiggle room on how you choose to format your table but there are a few **unbreakable** rules:

- The table **MUST** have a caption.
- The caption should be placed **ABOVE** the table (not below as for a figure or graph).
- The caption should be numbered accordingly. For example, if this is the first table in your document the figure caption should start "**Figure 1: . . .** ".
- The caption should be descriptive and unambiguous. The reader should be able to quickly interpret what is going on without having to read the body of the text.
- The table headers should be sensible and unambiguous. Any symbols or variables or units should be defined in the caption.
- The data should be formatted to a sensible number of decimal places (i.e. if you're measurements are made to 1 decimal place, your summary values should not be quoted to more than this).

Follow these rules and you can't go wrong. Break them and I **will** find you. Figure 3 shows how my summary table looks when copied and pasted into Word. I like to make my tables span the entire width of my document using the `Auto-fit to window` command. I also like to center my columns. These are personal preferences, but you can't deny they look great!

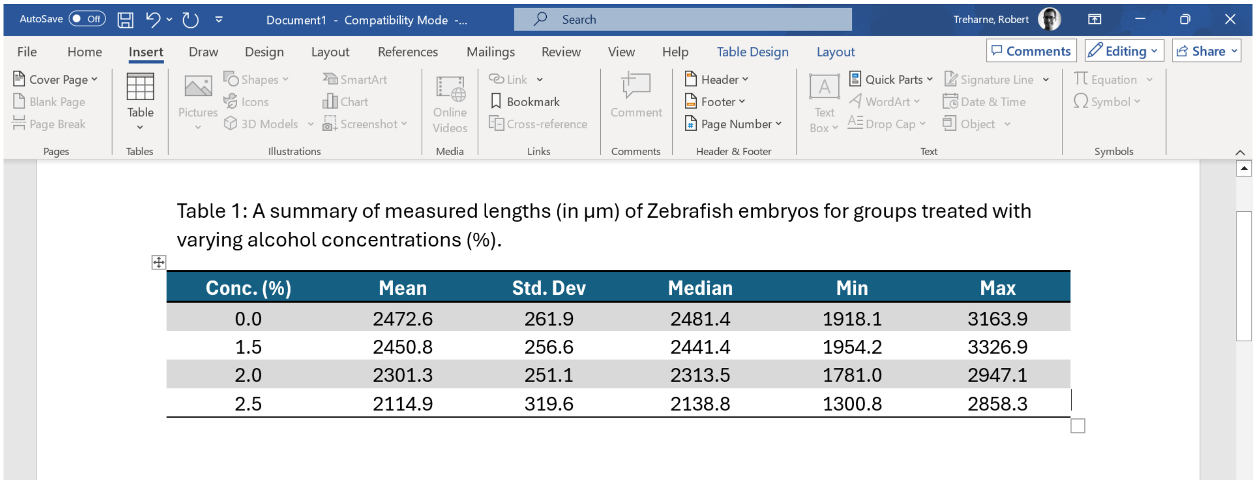


Table 1: A summary of measured lengths (in μm) of Zebrafish embryos for groups treated with varying alcohol concentrations (%).

Conc. (%)	Mean	Std. Dev	Median	Min	Max
0.0	2472.6	261.9	2481.4	1918.1	3163.9
1.5	2450.8	256.6	2441.4	1954.2	3326.9
2.0	2301.3	251.1	2313.5	1781.0	2947.1
2.5	2114.9	319.6	2138.8	1300.8	2858.3

Figure 3: Formatting a summary table in Microsoft Word.

2.2 Analysis of Variance (ANOVA)

2.3 Complete your Weekly Assignments

In the BIOS103 Canvas course you will find this week's **formative** and **summative** assignments. You should complete both of these before the end of the online workshop that corresponds to this section's content. The assignments are identical in all but the following details:

- You can attempt the **formative assignment** as many times as you like. It will not contribute to your overall score for this course. Make sure you practice this assignment until you're confident that you can get the correct answer on your own.
- You can attempt the **summative** assignment **only once**. It will be identical to the formative assignment but will use different values and datasets. This assignment **will** contribute to your overall score for this course.
- **Late submissions**. You have until 5pm on the day of the workshop for this week's content to submit your **summative** assignment. Late submissions will incur a 5% penalty for every part or full day beyond the deadline. Penalties will be capped at 40%.

3 Calibration Curves and Linear Regression in Excel

Hello world

3.1 Complete your Weekly Assignments

In the BIOS103 Canvas course you will find this week's **formative** and **summative** assignments. You should complete both of these before the end of the online workshop that corresponds to this section's content. The assignments are identical in all but the following details:

- You can attempt the **formative assignment** as many times as you like. It will not contribute to your overall score for this course. Make sure you practice this assignment until you're confident that you can get the correct answer on your own.
- You can attempt the **summative** assignment **only once**. It will be identical to the formative assignment but will use different values and datasets. This assignment **will** contribute to your overall score for this course.
- **Late submissions**. You have until 5pm on the day of the workshop for this week's content to submit your **summative** assignment. Late submissions will incur a 5% penalty for every part or full day beyond the deadline. Penalties will be capped at 40%.