Biofilm Regulatory Toolbox
September 8, 2026

1. **Download and Install R on your laptop**

   (a) Visit http://cran.r-project.org. This is the website for The Comprehensive R Archive Network (CRAN) from which you can download R and R packages.

   (b) The first box on this page is labeled *Download and Install R*. In that box, click on the appropriate link. For example, MAC users will click on *Download R for (Mac) OS X* and Microsoft Windows users will click on the link *Download R for Windows*. The rest of these instructions are specific to Windows users.

   (c) On the new page, click on the link named *base*.

   (d) Click on the link *Download R 4.6.1 for Windows* to download the executable file R43.6.1-win.exe to the hard drive on your computer. This version of R requires Windows 10 or newer.

   (e) In Windows Explorer, go to the folder where you saved R-4.6.1-win.exe and run the program.

   (f) You will be guided through the installation by a Setup Wizard.

2. **Download and Install RStudio on your laptop**: It is highly recommended that you download RStudio from:

   https://www.rstudio.com/products/rstudio/download/

   You will be taken to the posit website wher you can download `RStudio` and then install it.

3. There are many **excellent resources** for using R. One helpful resource is, *An Introduction to R* is available at https://cran.r-project.org/doc/manuals/r-release/R-intro.html.

4. **Help within R**: Type `help(name of function)` if you want additional help using a function. For example, type `help(mean)` or `?mean` for additional information about the function `mean()`. Examples are usually given at the very end of the help window.

5. RStudio is composed of 4 panels:

   - **Editor** for your `R`-scripts (top-left). Simply type your `R` commands here and then save them into a file. Put your cursor onto a line with an R-command and type Ctrl-Enter to run the command.
   - **R Console** (bottom-left). You can type commands directly here without using the Editor. Or you can copy and paste commands from the Editor to here.
   - **Environment** (Objects/Variables)/History (top-right)
   - **Files/Plots/Packages/Help/Viewer** (bottom-right).

6. **Packages:** Special-purpose software routines are bundled as separate "packages" in R. Some packages are automatically downloaded when R is downloaded. To download additional packages, run RStudio on your PC and then click on the *Packages* tab from the

bottom right pane of RStudio. From the drop down menu, click on the *Install* option and then choose the package(s) that you want to download. One packages that you will need to download for this workshop is *lme4*.

7. **Entering Data into R by Hand**: For small data sets, one option is to enter the data by hand. Consider when counts from 16 agar plates were obtained and recorded. We can enter the number of cfus for each plate:

```
cfus = c(5,8,7,5,3,4,3,9,5,8,5,6,5,6,4,7)
```

   When entering data by hand note that the name of the variable is whatever is to the left of the `=` which is `cfus` in our case. You can choose any name you wish (including any combination of letters, numbers, and symbols). Often it is best to name your variable something short that describes the variable. If you wish to view the data simply type in `cfus`. Note that if you capitalize a letter `R` considers this a different name. In other words `R` is case sensitive meaning that if you type in `Cfus` instead of `cfus` you will an error message telling you that the object `Cfus` does not exist.

8. **Importing a csv data file into R**: Consider three experiments performed to determine whether a treatment is effective against *Pseudomonas aeruginosa* biofilms grown in the high shear environment of the CDC reactor. The results were stored as a "comma separated value" file, or CSV file, `biofilmdata.csv`, from spreadsheet software, like Microsoft Excel. The contents of the CSV file are shown below:

```
experiment,treatment,cfu
1,control,5476335
1,control,6615459
1,control,8672813
1,disinfected,1205
1,disinfected,1965
1,disinfected,1076
2,control,3653928
2,control,4360838
2,control,4373811
2,disinfected,838
2,disinfected,1084
2,disinfected,1627
3,control,8234552
3,control,5566343
3,control,7899143
3,disinfected,546
3,disinfected,457
3,disinfected,533
```

   CSV data files can easily be imported into R. To import biofilmdata.csv into R, execute the following command:

```
D = read.csv("biofilmdata.csv",stringsAsFactors = TRUE)
```

read.csv is a *function*, and the *parameter* **stringsAsFactors = TRUE** tells R to read in columns with strings as a categorical factor. You could end up with an error like:

```
Error in file(file, "r") : unable to open connection
In addition:
Warning message: cannot open file `biofilmdata.csv'
```

The above error occurred because `biofilmdata.csv` was not in RStudio's **working directory**. To change the working directory to the one where you saved `biofilmdata.csv`, in RStudio click on the tab from the top of the screen **Session** → **Set Working Directory** → **Choose Directory**. A **Choose Directory** window appears where you can directly enter the directory that contains `biofilmdata.csv` on your computer, or browse to find the directory. Now we can try to read the data into R again.

```
D = read.csv("biofilmdata.csv",stringsAsFactors = TRUE)
```

9. **Simple statistical calculations**: The R-variable **D** that contains the data is called a **data frame**. We could have used any variable name like `CFUData` or `Experiments1-3`, but I don't like to type much, so I used the variable name `D`. Type the variable name at the R prompt to see what the imported data look like:

```
> D
   experiment   treatment     cfu
1           1     control 5476335
2           1     control 6615459
3           1     control 8672813
4           1 disinfected    1205
5           1 disinfected    1965
6           1 disinfected    1076
7           2     control 3653928
8           2     control 4360838
9           2     control 4373811
10          2 disinfected     838
11          2 disinfected    1084
12          2 disinfected    1627
13          3     control 8234552
14          3     control 5566343
15          3     control 7899143
16          3 disinfected     546
17          3 disinfected     457
18          3 disinfected     533
```

To access the individual columns of the data in D, type

```
> D$experiment
 [1] 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3
> D$treatment
 [1] control     control     control     disinfected disinfected disinfected control     c
[13] control     control     control     disinfected disinfected disinfected
Levels: control disinfected
> D$cfu
 [1] 5476335 6615459 8672813    1205    1965    1076 3653928 4360838
 [9]   4373811     838    1084    1627 8234552 5566343 7899143     546     457     533
```

R is case-sensitive! The upper and lower-case letters in the variable name must be EXACTLY as given in the data file or R will not find it. For example,

```
> D$CFU
NULL
```

Notice that R recognizes that **treatment** is a categorical variable within the data frame **D** and gives the *levels* or categories associated with each. The variables **experiment** and **cfu** are recognized as quantitative variables.

In addition to **read.csv()**, we will be using many other functions that R has available. For example, **log10()** performs a $\log_{10}$ transform, **mean()** calculates the mean and **median()** calculates the median. The functions **sd()** and **var()** calculate the standard deviation and variance respectively. The next line calculates the mean of the $\log_{10}$-transformed CFUs across all experiments:

```
> mean(log10(D$cfu))
[1] 4.866603
```

Many times after entering data into R, I will use **summary()** to calculate the 5-number summary for quantitative variables in the data frame and to calculate frequencies of categories for categorical variables. For example:

```
   experiment       treatment      cfu
 Min.   :1    control    :9   Min.   :    457
 1st Qu.:1    disinfected:9   1st Qu.:   1078
 Median :2                    Median :1827946
 Mean   :2                    Mean   :3047920
 3rd Qu.:3                    3rd Qu.:5543841
 Max.   :3                    Max.   :8672813
```

To graph the $\log_{10}$(CFU) using boxplots:

```
boxplot(log10(D$cfu) ~ D$treatment)
```

To compute the mean $\log_{10}$(CFU) for each treatment separately, execute

```
> tapply(log10(D$cfu),D$treatment,mean)
    control disinfected
   6.766577    2.966629
```

The mean of the $\log_{10}$(CFU) of the untreated controls for each experiment, and the repeatability SD are:

```
> i=D$treatment=="control"
> C.mean=tapply(log10(D$cfu[i]),D$experiment[i],mean)
> C.mean
       1        2        3
6.832403 6.614397 6.852930
> sd(C)   # repeatability SD of the untreated controls
[1] 0.1321908
```

The last block of commands stores the location of the control data into the indexing variable `i` and stores the means of the untreated controls into a variable called `C.mean` so that you can take the SD of the 3 mean values to get the repeatability SD of the untreated controls. The hash # is used to add comments, and tells `R` to ignore everything after the #.

Similarly, the mean of the treated $\log_{10}$(CFU)s by experiment is:

```
> i=D$treatment=="disinfected"
> T.mean=tapply(log10(D$cfu[i]),D$experiment[i],mean)
> T.mean
       1        2        3
3.135387 3.056554 2.707945
```

The 3 log reductions by experiment:

```
> LR = C.mean-T.mean
> LR
       1        2        3
3.697016 3.557843 4.144985
```

The last command shows that `R`-variables can be used with the mathematical operators +, -, * and /.

Now we can calulcate the overall mean LR and the repeatability SD of the LRs

```
> LR
       1        2        3
3.697016 3.557843 4.144985
> mean(LR)
[1] 3.799948
> sd(LR)    # repeatability SD of the LRs
[1] 0.3068062
```

And finally, a two-sided t-test and a two-sided 95% t-CI for the true mean LR:

```
> t.test(LR)

 One Sample t-test

data:  LR
t = 21.452, df = 2, p-value = 0.002166
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 3.037799 4.562097
sample estimates:
mean of x
 3.799948
```