Week 1 - Hands-On Examples

Create an R project

1. Create a new project in your folder.

Some rules for naming your project:

- be descriptive and keep it short
- use snake case (only lowercase letters and underscores allowed), avoid special characters (such as `!`, `#`,) and spaces
- the name cannot starts with numbers

2. Open your R project, create three new folders, *i.e.*, `data`, `scripts`, `outputs`.

Import Dataset

Data Description

We'll be working with a gene expression dataset as an example, sourced from this link.

The specific file we'll use is named "**read-counts.csv**", which you can download from this zipped folder.

The data comes from an experiment using PCR to study 44 genes. The results were measured to see which genes are active at different stages in Yeast cell cycling. Several strains were tested, including wildtype and some with specific genes knock-downs. Samples were taken at nine time points over two cell cycles (two hours).

Importing into Rstudio

1. Download the zipped file to your computer and extract its contents.
2. If you are using the RStudio server, upload the `read-counts.csv` to the folder `data` of your R project.
3. Click on the file to "View" it and identify the column separator.
4. Import the file into R and call the imported data "counts".

Play with Basic R Commands

1. What is the absolute file path of the count data? What is its absolute path? Verify your answer using the function `file.exists()`.

```
file.exists("relative_path/to/your/file") # replace the path by the yours
file.exists("absolute_path/to/your/file") # replace the path by the yours
```

2. Check the "Environment" panel or use the function `dim()`. What is the dimension of the data frame?

```
dim(counts)
```

```
[1] 45 41
```

In the "Environment" panel, click on the tabular icon next to the dataset to visualize the it.

We can extract all gene expressions for the sample named "WT:2" using `counts[["WT:2"]]`.

3. Try `mode()` on the expression data for "WT:2", what does it return?

```
mode(counts[["WT:2"]])
```

```
[1] "numeric"
```

4. Calculate the average expression (`mean()`) and standard deviation (`sd()`) of genes from the sample "WT:2".

```
mean(counts[["WT:2"]])
```

```
[1] 148
```

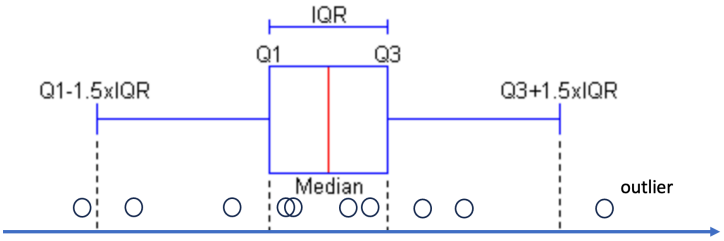
```r
sd(counts[["WT:2"]])
```

```
[1] 392.7854
```

5. Generate descriptive statistics for all genes from the sample "WT:2" using `summary()`.

```r
summary(counts[["WT:2"]])
```

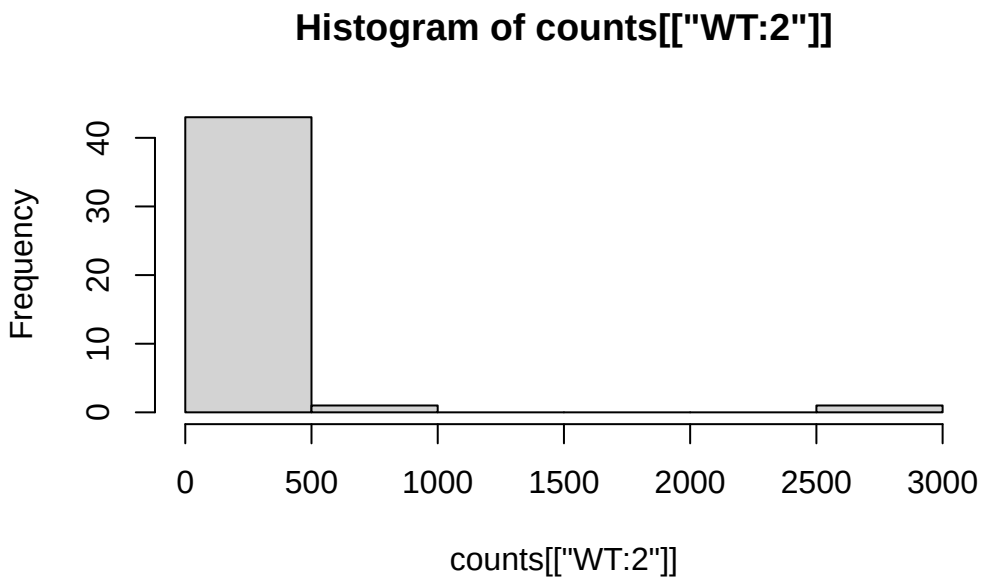
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0       6      27     148     110    2527
```

> **Stats Time!**
>
> What are quartiles?
> Quartiles are three values that split sorted data into four equal parts.
>
> 
>
>
> IQR (Interquartile range) = Q3 - Q1

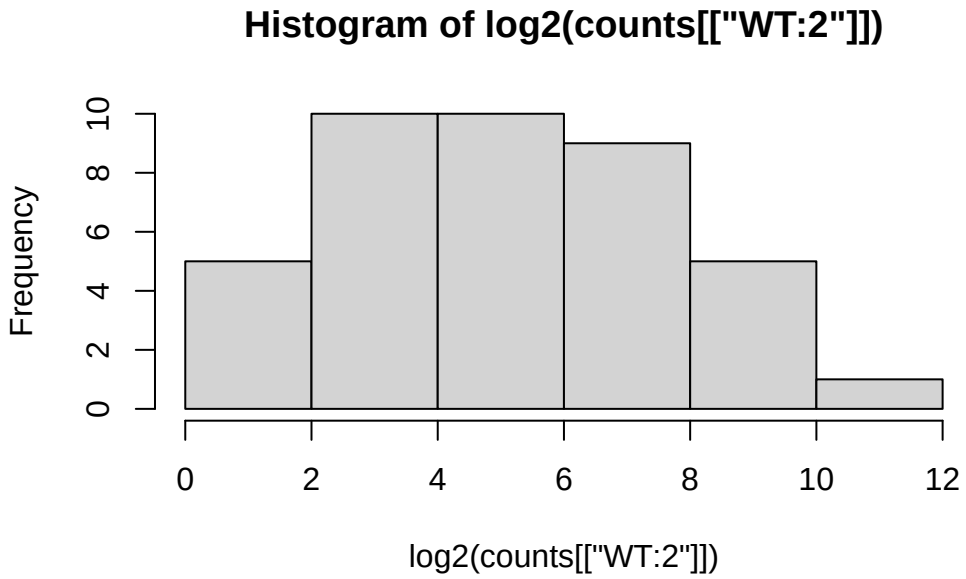
6. Generate a histogram for the "WT:2" sample using `hist()`. What does the distribution look like?

> **Stats Time!**
>
> **Histograms** help us see how data is spread out. They show how many data points fall into different ranges, or *bins*. By looking at a histogram, we can quickly understand the shape of the data, like if it's skewed or has outliers. It's a simple way to get an overview of your data.

```r
hist(counts[["WT:2"]])
```




```
hist(log2(counts[["WT:2"]]))
```



Get Your First Quarto Report

Click the "Render" to generate the report!

- Change something in this script and re-render it, is the report up-to-date?
- Where is your report stored?
- What should you do if you want the report be stored in a specified folder? => use a configuration file for Quarto.

Open a new text file and copy paste following code, save it as `_quarto.yml` in your project folder.

```
project:
  output-dir: outputs/
```

Try "Render" again, now where is your report?

Good job! You've taken your first big steps into R, and you're off to a great start, keep it up!