Beyond the bench R tutorial

The following is a Quarto document that illustrates how to do a few basic data analyses in R.

Quarto is an open-source scientific publishing system that allows users to create documents, presentations, and websites that contain written text, code, and visualizations.

This Quarto document covers the following topics:

- R basics (which we will like go through in an R script)
- Data visualization using ggplot
- Basic hypothesis tests

Please follow along as we go through the material!

Getting started

Before we start, please run the following code once to install the packages we will need by pressing the green play buttong

```r
install.packages("palmerpenguins")
install.packages("ggplot2")
install.packages("tidyr")
install.packages("dplyr")
```

Part 1: R Basics

Let's start by going through some of the basic syntax of R.

1.1 Mathematical operations

As you might expect, R can be used to do basic mathematical operations.

```r
2 + 3
```

```
[1] 5
```

```r
7 * 5
```

```
[1] 35
```

1.2 Storing results in objects

We can store results in "objects" using the assignment operator `<-`.

```r
a <- 4

b <- 7

z  <- a + b

z
```

```
[1] 11
```

1.3 Numeric, string and Boolean data types

R has a few basic data types. The most common ones are numeric, string and Boolean values. We can check what type of value an object is holding using the `class()` function.

```r
a <- 7

s <- "s is a terrible name for an object"

b <- TRUE


class(a)
```

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

```r
class(s)
```

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

1.4 Functions

We can also apply functions to values (and values stored in objects).

To get help on a function we can use the syntax `? function_name`.

We can also add comments (which is notes that are not executed) using the `#` symbol.

```r
sqrt(49)
```

```
[1] 7
```

```r
tolower("DATA is AWESOME!")
```

```
[1] "data is awesome!"
```

```r
# To get help information about a function or data set
# ? sqrt


# One can add comments to your code to describe what you are doing
sqrt(49)    # this takes the square root of 49
```

```
[1] 7
```

1.5 Vectors

Vectors are ordered sequences of numbers or letter.

We use the `c()` function to create vectors.

We can access elements of vectors using `[]` brackets.

Functions can also be applied to vectors.

```r
v  <-  c(5, 232, 5, 543)

s  <-  c("statistics", "data", "science", "fun")


# One can access elements of a vector using square brackets []
s[4]         # what will the answer be?
```

```
[1] "fun"
```

```r
# One can test which elements are greater than a value
z > 3
```

```
[1] TRUE
```

```r
# One can also apply functions to vectors
z <- 2:10

sqrt(z)
```

```
[1] 1.414214 1.732051 2.000000 2.236068 2.449490 2.645751 2.828427 3.000000
[9] 3.162278
```

```r
mean(z)
```

```
[1] 6
```

1.6 Data frames

Data frames are a type of object in R that are used to store tabular data.

We can extract columns a column from a data frame as a vector using the `$` operator.

As a motivating example, let's look at data on penguins from the Palmer Penguin dataset.

"PAY THE TARIFFS!"
"WE'LL BE TAKING OUR BUSINESS ELSEWHERE"
www.NaturalWorldHeritageSites.org/flickr/laikolosse
Image posted on X by @GabeFin
imgflip.com

Figure 1: Penguins

```r
#install.packages("palmerpenguins")

library(palmerpenguins)
```

```
Warning: package 'palmerpenguins' was built under R version 4.4.3
```

```r
penguins
```

```
# A tibble: 344 x 8
   species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
   <fct>   <fct>              <dbl>         <dbl>             <int>       <int>
 1 Adelie  Torgersen           39.1          18.7               181        3750
 2 Adelie  Torgersen           39.5          17.4               186        3800
 3 Adelie  Torgersen           40.3          18                 195        3250
 4 Adelie  Torgersen           NA            NA                  NA          NA
 5 Adelie  Torgersen           36.7          19.3               193        3450
 6 Adelie  Torgersen           39.3          20.6               190        3650
 7 Adelie  Torgersen           38.9          17.8               181        3625
 8 Adelie  Torgersen           39.2          19.6                 195         4675
 9 Adelie  Torgersen           34.1          18.1                 193         3475
10 Adelie  Torgersen           42            20.2                 190         4250
# i 334 more rows
# i 2 more variables: sex <fct>, year <int>
```

```
# Get the mean flipper length
mean(penguins$flipper_length_mm, na.rm = TRUE)
```

```
[1] 200.9152
```

```
# Get the number of penguins of different species
table(penguins$species)
```

```

   Adelie Chinstrap    Gentoo 
      152        68       124 
```

Part 2: Data visualization using ggplot

The ggplot2 library was created by Hadley Wickham. It is based on Leland Wilkinson's "grammar of graphics" where graphics are created from a combination of basic visual elements.

A few resources to learn more about ggplot are:

- R for Data Science
- The ggplot cheatsheet

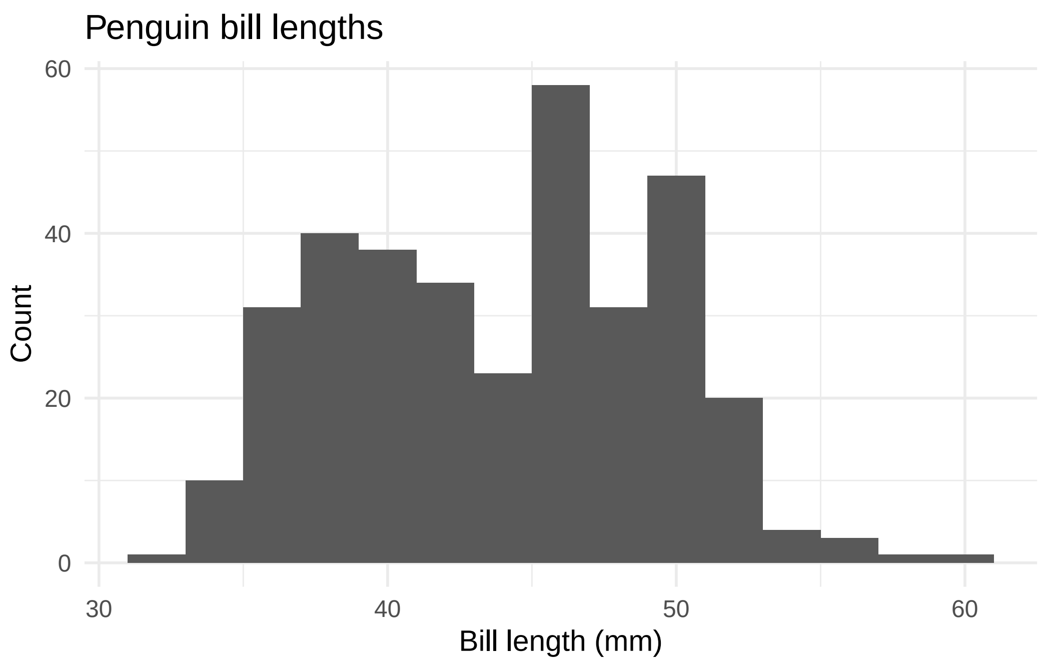
Let's now visualize some of the penguins data using the ggplot2 package!

Part 2.1 Visualzing a single quatitative variables

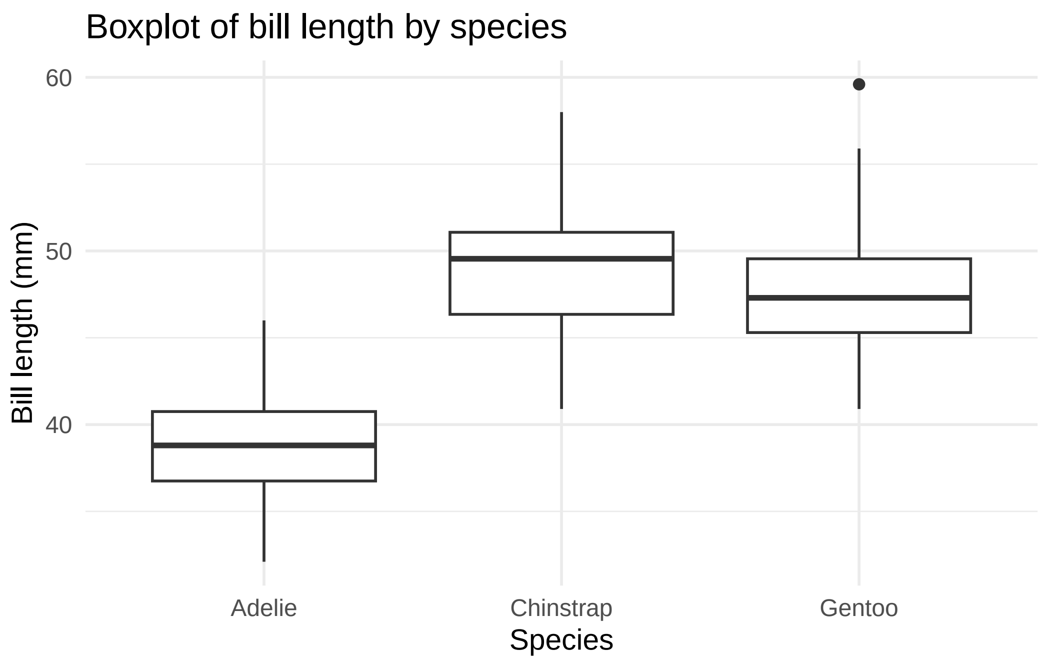
As you recall from last week's talk, we can visualize a single quantitative variable using histograms, boxplots and violin plots (among other types of plots).

Let's explore this by visualizing the bill length (`bill_length_mm`) of the penguins in the Palmer dataset.

```r
# install.packages(ggplot2)

library(ggplot2)


# Histogram, density plot, boxplot and violin plot
ggplot(penguins, aes(x = bill_length_mm)) +
  geom_histogram(binwidth = 2) +
  # geom_density() +
  # geom_boxplot()
  # geom_violin()
  labs(title = "Penguin bill lengths",
       x = "Bill length (mm)",
       y = "Count") +
  theme_minimal()
```



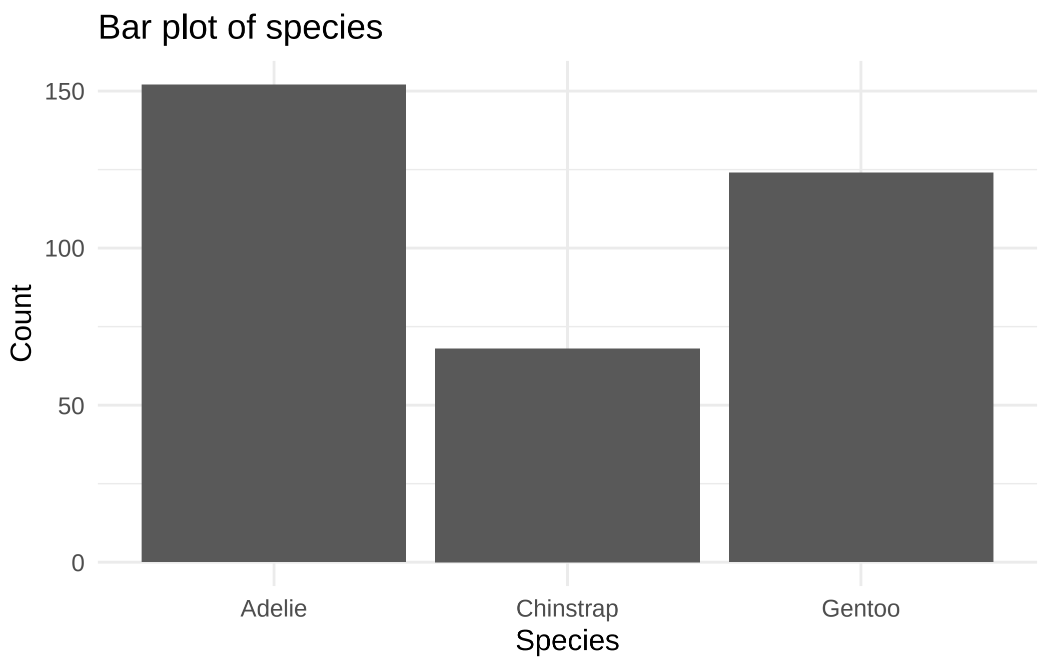
```r
# Side-by-side box/violin plots
ggplot(penguins, aes(x = species, y = bill_length_mm)) +
  geom_boxplot() +
  # geom_violin()
  labs(title = "Boxplot of bill length by species",
       x = "Species",
         y = "Bill length (mm)") +
  theme_minimal()
```



Part 2.2 Visualizing a single categorical variable

As you recall, we can visualize a single categorical variable using a bar plot.

```
ggplot(penguins, aes(x = species)) +
  geom_bar() +
  labs(title = "Bar plot of species",
       x = "Species",
       y = "Count") +
  theme_minimal()
```

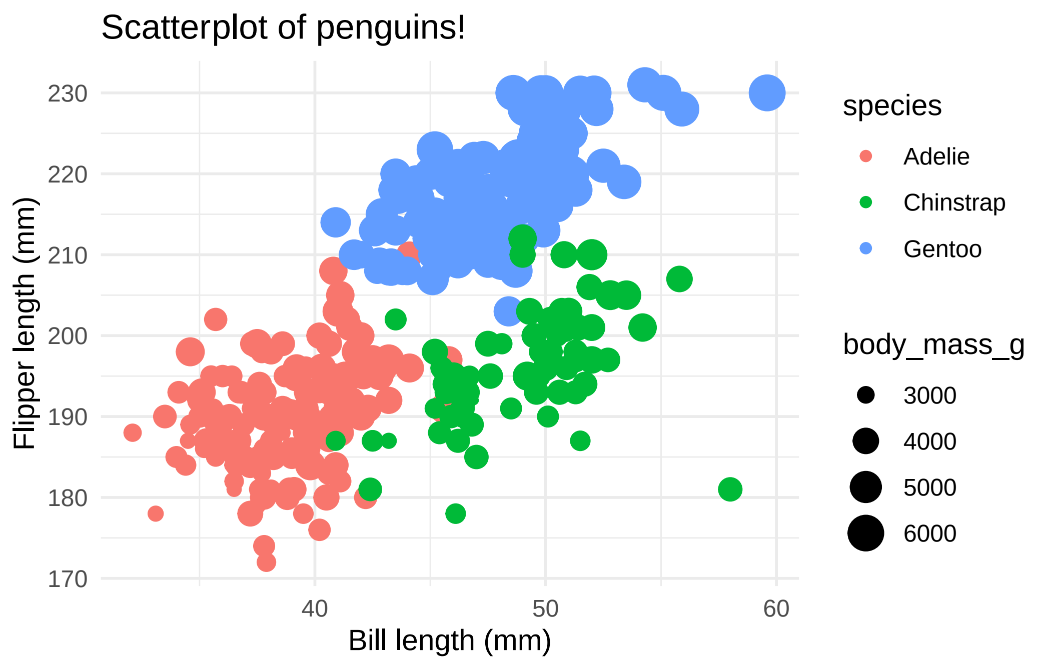


Part 2.3 Visualizing two quantitative variables

As I'm sure you know, we can visualize two quantitative variables using a scatter plot. We can also use ggplot to map on other visualize properties to the points in the scatter plot, such as color, size and shape.

```
ggplot(penguins, aes(x = bill_length_mm,
                     y = flipper_length_mm,
                     color = species,
                     size = body_mass_g)) +
  geom_point() +
  labs(title = "Scatterplot of penguins!",
       x = "Bill length (mm)",
       y = "Flipper length (mm)") +
  theme_minimal()
```

```
Warning: Removed 2 rows containing missing values or values outside the scale range
(`geom_point()`).
```



Part 3: Basic hypothesis tests

In this section we will cover a few basic hypothesis tests. As motivation, we will focus on figures 6D, and 6F from a paper by Smarduch et al, (2025) which were analyses that Ece suggested we cover. The original paper can be found at: https://www.embopress.org/doi/full/10.1038/s44318-025-00370-y

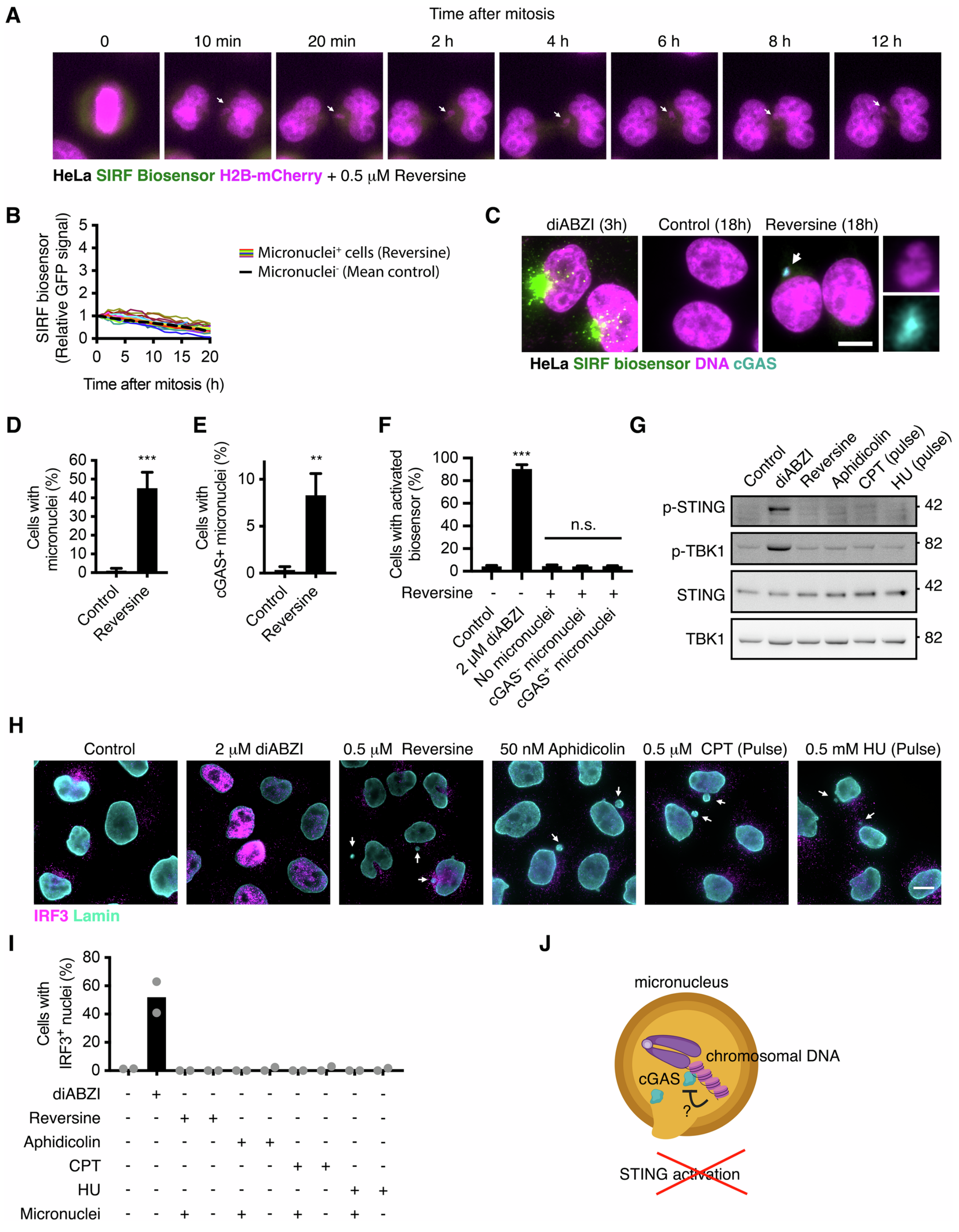


Figure 2: Figure 6 from Smarduch et al. In (D, E), P-values from a t-test between the two groups with N=3 independent experiments are indicated as P=0.0009 an P=0.0041, respectively. In (F), P-values from one-way ANOVA between the indicated groups with N=3 independent experiments are indicated as P<0.0001, or not significant (n.s.) from left to right as P=0.9998, P=0.9999, P=0.9999.

Part 3.1: The data

To analyze the data, I started by doing a little manual data cleaning in excel to make the data a little more "tidy". I then saved this data in a .csv file.

Let's view the data in Figure 6D.

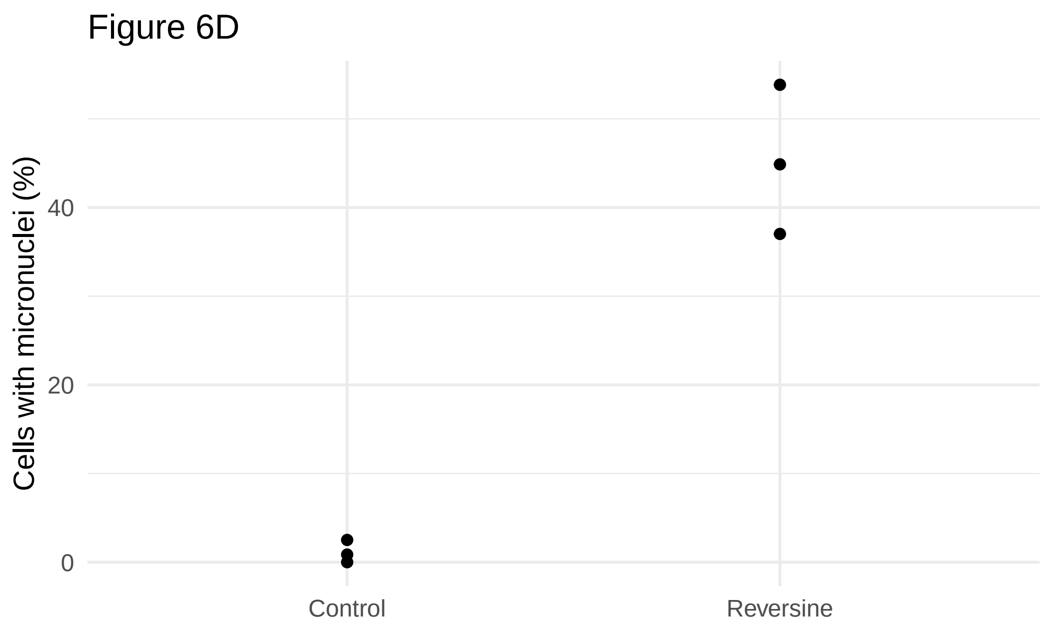
```
fig6d <- read.csv("fig6d.csv")

fig6d
```

```
  Experiment   Control Reversine
1          1 0.0000000  37.02422
2          2 0.8571429  44.87179
3          3 2.5089606  53.84615
```

Part 3:2 Visualzing the data

We can visualize the data by convering it from "wide" format to a "long" format using the `tidyr` `pivot_longer()` function. Once it is it long format, we can visualize the data using ggplot.

```
# install.packages("tidyr")

library(tidyr)


# converting the data to "long" format for plotting purposes
fig6d_long <- tidyr::pivot_longer(fig6d,
                                  cols = -Experiment,
                                  names_to = "Condition",
                                  values_to = "Value")



ggplot(fig6d_long, aes(x = Condition, y = Value)) +
  geom_point() +
  #geom_jitter(width = 0.2) +
  theme_minimal() +
  labs(title = "Figure 6D",
       y = "Cells with micronuclei (%)",
       x = "")
```



Part 3:3 Replicating the t-test

Let's now try to replicate the t-test that was used in the paper to compare the means of the two groups compare the Control and Reversine groups.

The paper reported the p-value as 0.0009. Let's see if we replicate this...

```r
t.test(fig6d$Control, fig6d$Reversine)
```

```
	Welch Two Sample t-test

data:  fig6d$Control and fig6d$Reversine
t = -8.9774, df = 2.0918, p-value = 0.01057
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -64.40889 -23.84182
sample estimates:
mean of x mean of y
 1.122034 45.247390
```

```r
# Are the measurements paired?
t.test(fig6d$Control, fig6d$Reversine, paired = TRUE)
```

```
    Paired t-test

data:  fig6d$Control and fig6d$Reversine
t = -10.678, df = 2, p-value = 0.008656
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 -61.90465 -26.34606
sample estimates:
mean difference
      -44.12536
```

Using an independent samples t-test yields a p-value of 0.01057, which is "statistically significant" by the conventional threshold of 0.05, but is much higher than the value reported in the paper. Unfortunately the paper does not report the t-statistic value so it's hard to tell exactly why we are getting different results.

More likely, the paper used a paired t-test. When we run this analysis, get a p-value of 0.009, which seems similar to the p-value of 0.0009 reported in the paper (same number, but off by an order of magnitude).

Part 3:4 Figure 6F analysis

Let's now look at the data in Figure 6F. The code below loads the data and converts it to a long format which will be useful for visualization and analysis.

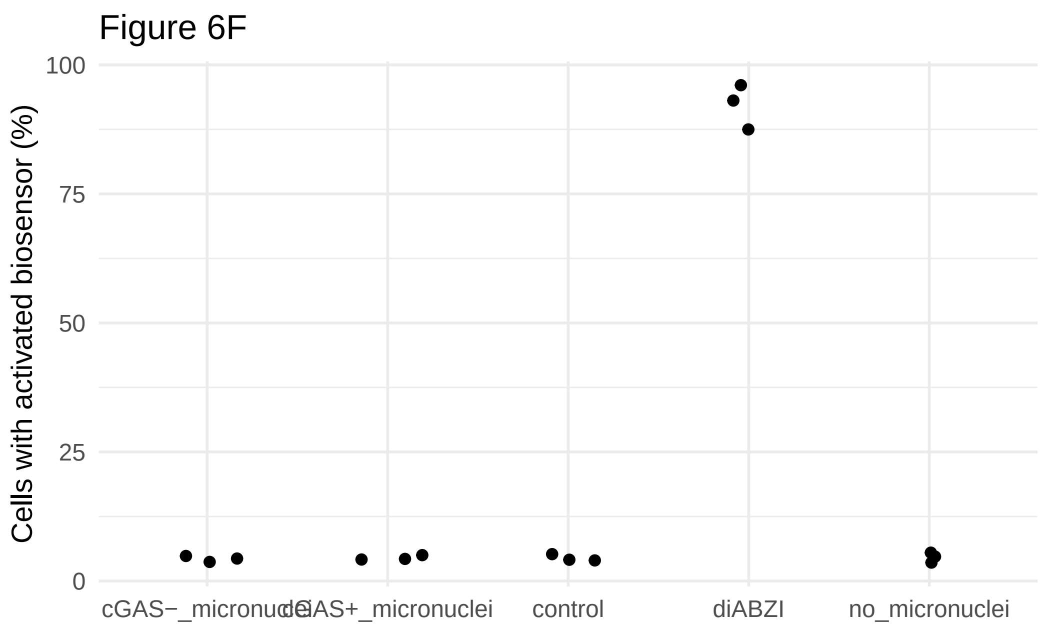
```
fig6f <- read.csv("fig6f.csv")

names(fig6f) <- c("experiment", "control", "diABZI", "no_micronuclei", "cGAS-_micronuclei", "

fig5f_long <- fig6f |>
  pivot_longer(cols = -experiment, names_to = "Condition", values_to = "Value")

fig5f_long
```

```
# A tibble: 15 x 3
   experiment Condition         Value
   <chr>      <chr>             <dbl>
 1 EXP1       control            5.19
 2 EXP1       diABZI            87.5
 3 EXP1       no_micronuclei     3.55
 4 EXP1       cGAS-_micronuclei  3.70
 5 EXP1       cGAS+_micronuclei  4.29
 6 EXP2       control            3.98
 7 EXP2       diABZI            93.1
 8 EXP2       no_micronuclei     4.72
 9 EXP2       cGAS-_micronuclei  4.84
10 EXP2       cGAS+_micronuclei  5
11 EXP3       control            4.12
12 EXP3       diABZI            96.1
13 EXP3       no_micronuclei     5.48
14 EXP3       cGAS-_micronuclei  4.35
15 EXP3       cGAS+_micronuclei  4.17
```

Part 3:5 Visualizing the data in figure 6F

To start the analysis of figure 6F, let's visualize the data.

```r
ggplot(fig5f_long, aes(x = Condition, y = Value)) +
geom_jitter(width = 0.2) +
theme_minimal() +
labs(title = "Figure 6F", y = "Cells with activated biosensor (%)", x = "")
```



Part 3:6 Replicating the ANOVA

Let's now try to replicate the ANOVA that was used in the paper to compare the means of the four groups (we will leave aside whether this is the most appropriate analysis to do for this data).

We can also use the `TukeyHSD()` function to do a post-hoc test to see which groups are significantly different from each other which is what appears to have been done in the paper.

From figure caption: P-values from one-way ANOVA between the indicated groups with N=3 independent experiments are indicated as p < 0.0001, or not significant (n.s.) from left to right as P=0.9998, P=0.9999, P=0.9999. Let's see if we can replicate this...

```r
anova(lm(Value ~ Condition, data = fig5f_long))
```

```
Analysis of Variance Table

Response: Value
          Df  Sum Sq Mean Sq F value    Pr(>F)
Condition  4 18490.2  4622.6  1107.4 3.472e-13 ***
Residuals 10    41.7     4.2
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
TukeyHSD(aov(Value ~ Condition, data = fig5f_long))
```

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Value ~ Condition, data = fig5f_long)

$Condition
                                          diff        lwr       upr
cGAS+_micronuclei-cGAS-_micronuclei   0.18738050  -5.302689  5.677450
control-cGAS-_micronuclei             0.13518326  -5.354886  5.625253
diABZI-cGAS-_micronuclei             87.92587427  82.435805 93.415944
no_micronuclei-cGAS-_micronuclei      0.28649774  -5.203572  5.776567
control-cGAS+_micronuclei            -0.05219723  -5.542267  5.437872
diABZI-cGAS+_micronuclei             87.73849378  82.248424 93.228563
no_micronuclei-cGAS+_micronuclei      0.09911724  -5.390952  5.589187
diABZI-control                       87.79069101  82.300621 93.280761
no_micronuclei-control                0.15131447  -5.338755  5.641384
```

```
no_micronuclei-diABZI              -87.63937654 -93.129446 -82.149307
                                       p adj
cGAS+_micronuclei-cGAS-_micronuclei 0.9999571
control-cGAS-_micronuclei           0.9999883
diABZI-cGAS-_micronuclei            0.0000000
no_micronuclei-cGAS-_micronuclei    0.9997680
control-cGAS+_micronuclei           0.9999997
diABZI-cGAS+_micronuclei            0.0000000
no_micronuclei-cGAS+_micronuclei    0.9999966
diABZI-control                      0.0000000
no_micronuclei-control              0.9999817
no_micronuclei-diABZI               0.0000000
```