# PROJECT: PHYLLOTAXIS: DRAW FLOWERS USING MATHEMATICS

**Task 1: Instructions**

* Read the project introduction.
* Load the ggplot2 package.

**Good to know**

To complete this project, you will need to know your way around the ggplot2 package, one of the most useful and downloaded packages of R which should be in the toolbox of every R user. If you're not familiar with ggplot2, we recommend that you complete the [**Data Visualization with ggplot2 (Part 1)**](https://www.datacamp.com/courses/data-visualization-with-ggplot2-1) course first.

The options() function used in this notebook is to set global options related to the way that R works. Here, we use it to define the size of the images in the notebook.

You don't need any mathematical background to complete this project, but if you want to know more about phyllotaxis, you can check out [**this article on Wikipedia**](https://en.wikipedia.org/wiki/Phyllotaxis)(<https://en.wikipedia.org/wiki/Phyllotaxis>).

Take Hint

**Task 2: Instructions**

Create a scatter plot of a circle.

* Complete the statement to make a scatter plot using geom\_point().

After completing this task, you will see a scatter plot of 50 points arranged in a circle.

Storing a ggplot() object in a variable (p in our case), will not display an image on the screen, but it allows you to modify the object in a subsequent step. Writing aes(x, y) will map the first parameter (x) to the x-axis and the second (y) to the y-axis. This is the same as writing aes(x=x, y=y); writing aes(x=y, y=x) would swap the axes.

The function seq() is quite useful. It allows you to create a sequence of n numbers between a lower and upper bound or by defining a lower bound and a step size.

Helpful links:

* [**ggplot2 and its geometries**](http://ggplot2.tidyverse.org/reference/)(<https://ggplot2.tidyverse.org/reference/>)

### HINT

Here you should add a geom\_point() layer to p using + without storing the result in a variable.

**Task 3: Instructions**

* Create the variable points which defines the number of points to draw and set its value to 500.
* Create another variable called angle and set its value to π(3 − √5) .
* Make a scatter plot.

To correctly set angle you need to translate the mathematical formula above into R code. If you can't figure it out, check out the hint below.

Take Hint

**Task 4: Instructions**

* Make the background white and remove the grid, ticks, axes titles, and axes text from the spiral plot.

To make these changes to the plot appearance, you need to use the theme() function.

There are some predetermined themes in ggplot2 that would help with the plot's appearance, but we will use the theme() function instead to have move control over every detail of the plot.

Many colors in R can be specified by name: search for *colors in R* in Google and you will find the complete list.

Helpful links:

* theme() [**documentation**](http://ggplot2.tidyverse.org/reference/theme.html) (<https://ggplot2.tidyverse.org/reference/theme.html>)
* [**predetermined themes in ggplot2**](https://ggplot2.tidyverse.org/reference/ggtheme.html)(<https://ggplot2.tidyverse.org/reference/ggtheme.html>)

### HINT

All elements of a plot can be removed with element\_blank(). The background could also be removed with element\_blank, but you should color it white with element\_rect(fill="white"). If you want to remove ticks and color the background white, use the following:

theme(panel.background = element\_rect(fill="white"),

axis.ticks=element\_blank())

**Task 5: Instructions**

* Copy and paste the last lines of code that created the plot from Task 4.
* Change the call to geom\_point() so that the size of points equals 8, the alpha (transparency) equals 0.5, and the color is darkgreen.

After doing this, you should see a plot where all points will have the same size, alpha, and color. The alpha parameter will produce a darker color where points overlap.

Colors can be directly specified by name in R (i.e. "darkgreen") or by its *hexadecimal code* (i.e. "#006400"). alpha values go from 0 (totally transparent) to 1 (totally opaque), and size can take any value, but if it is negative no points will be displayed.

Helpful links:

* **Changing the appearance of points in (**<https://ggplot2.tidyverse.org/reference/geom_point.html>**)geom\_point()**

Take Hint

**Task 6: Instructions**

* Copy and paste the solution from Task 5.
* Within geom\_point(), map the size aesthetic to the variable t, remove the color, and set the shape (outside the aesthetics) to an asterisk (\*).
* Remove the legend from the plot.

As size now depends on the variable t, a legend will appear. You can add an argument to theme() to remove it. The Cookbook for R has more information about how to do this.

Every shape is defined by a number: search for *R plot symbols* on Google, and you'll find many lists of shapes and codes available for geom\_point().

Helpful links:

* [**Cookbook for R**](http://www.cookbook-r.com/Graphs/Legends_(ggplot2)/#removing-the-legend)(<http://www.cookbook-r.com/Graphs/Legends_(ggplot2)/#removing-the-legend>)

**HINT**

To map size to t you have to use aes(size=t) as the first argument to geom\_point(). To remove legend you should add legend.position="none" to the theme() function. The shape of points can be changed with the shape argument of geom\_point() (asterisks have shape=8).

**Task 7: Instructions**

* Copy and paste the solution from Task 6.
* Change the shape of all points to *filled triangles*, and change the color of the points to yellow.
* Change the color of the background to "darkmagenta".

The code for filled triangles is the 7th prime number. If you're not that into prime numbers, remember that you can always search for R plot symbols on Google.

**HINT**

The beginning of your code should look like this:

p <- ggplot(df, aes(x\*t, y\*t))

p + geom\_point(aes(size=t), alpha=0.5, shape=17, color="yellow") +

theme(legend.position="none",

panel.background = element\_rect(fill="darkmagenta"),

....

**Task 8: Instructions**

* Change the value of angle from the Golden Angle (which is about 2.4) to 2.0 .
* Copy and paste the code from Task 7 and create the plot.

Take Hint

**Task 9: Instructions**

* Set angle to 13\*pi/180, and double the amount of points.
* Within geom\_point(), set size to 80, alpha to 0.1, shape to open circles, and color to "magenta4".
* Set the background fill to "white".

Congratulations! You have finished the project!

There are so many possible images you can create here - play around with different values of angle, points, and the ggplot() parameters to see what you can come up with.

Helpful links:

* [**Table of different shape values**](http://www.cookbook-r.com/Graphs/Shapes_and_line_types/)(<http://www.cookbook-r.com/Graphs/Shapes_and_line_types/>)
* [**Color names**](http://research.stowers.org/mcm/efg/R/Color/Chart/ColorChart.pdf)(<https://research.stowers.org/mcm/efg/R/Color/Chart/ColorChart.pdf>)

### HINT

To recreate the flower, use:

angle <- 13\*pi/180

points <- 2000

t <- (1:points)\*angle

x <- sin(t)

y <- cos(t)

df <- data.frame(t, x, y)

p <- ggplot(df, aes(x\*t, y\*t))

p + geom\_point(size=80, alpha=0.1, shape = 1, color="magenta4")+

theme(legend.position="none",

panel.background = element\_rect(fill="white"),

panel.grid=element\_blank(),

axis.ticks=element\_blank(),

axis.title=element\_blank(),

axis.text=element\_blank())