Phyllotaxis: draw flowers using mathematics

"The scientist does not study nature because it is useful; he studies it because he delights in it, and he delights in it because it is beautiful." (Henri Poincaré)

There are many examples of *natural facts* that can be described in mathematical terms. Nice examples are the shape of snowflakes, the *fractal geometry* of romanesco broccoli or how self-similarity rules the growth of plants.

R is a tool for doing serious analysis, but not everything in life is serious. Life is also funny, and R can be used to have fun and to do beautiful things. Its graphical power can be used to produce artistic images like the one that illustrates this section, which is inspired by how plants arrange their leaves. This fact is called *phyllotaxis* and will serve as the basis of this project.

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**.

In this Workspace project, we are using the `ggplot2` package. Apart from having fun, we will learn many important features of it that will be useful not only to do art but also to represent data in real-life problems.

```
# Set plot images to a nice size
options(repr.plot.width = 6, repr.plot.height = 6)

# Load the ggplot2 package
library(ggplot2)
```

Warming up: drawing points on a circle

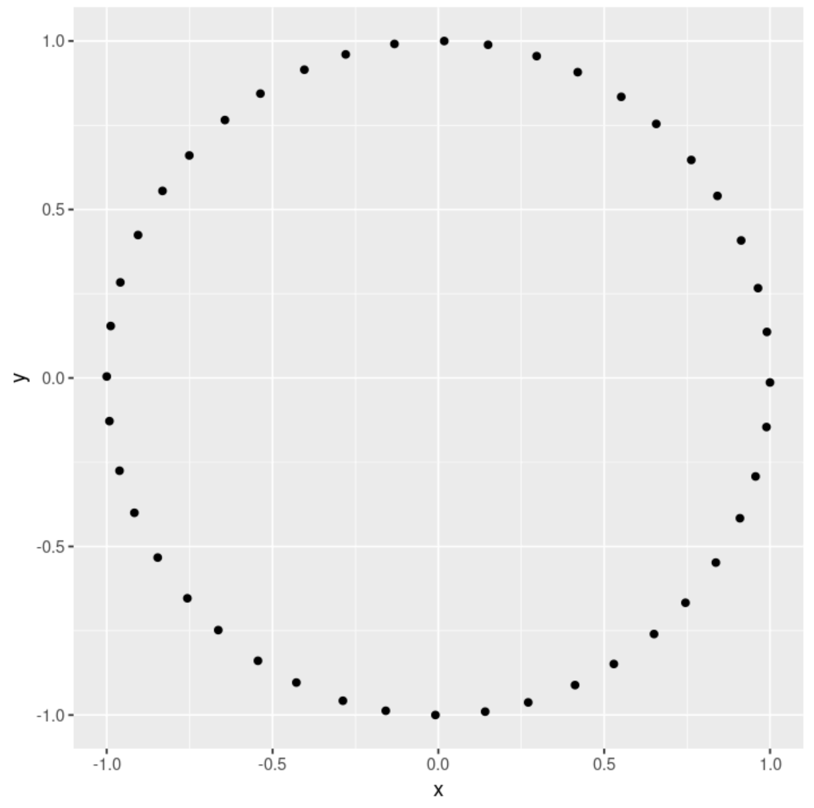
In this project, we will only work with `geom_point()` which plots points in two dimensions. We'll need a dataset with two variables; let's call them `x` and `y`.

To get you started, we've drawn `44` points on a circle of radius `1`. As every `(x, y)` point should be in the unit circle, it follows that `x² + y² = 1`. We can get this using the *super famous* Pythagorean trigonometric identity which states that `sin²(θ) + cos²(θ) = 1` for any real number `θ`.

For this project, all plots will be stored in separate variables so they can be validated when you submit. Storing a `ggplot()` object in a variable (`p`), will not display an image on the screen, but it allows you to modify the object in a subsequent step.

```r
# Create circle data to plot
t <- seq(1:44)
x <- sin(t)
y <- cos(t)
df <- data.frame(t, x, y)

# Make a scatter plot of points in a circle
p <- ggplot(df, aes(x, y))
circle <- p + geom_point()
circle
```



Plants can arrange their leaves in spirals. A spiral is a curve which starts from the origin and *moves away* from the origin as it revolves around it. In the plot above all our points are the same distance from the origin. A simple way to arrange them in a spiral is to multiply `x` and `y` by a factor which increases for each point. In mathematical terms, this corresponds for example to `x = t * sin(t)` and `y = t * sin(t)`.

We *could* use our sequence `t` as that factor, as it meets these conditions, but we will do something more *harmonious*. We will use the Golden Angle:

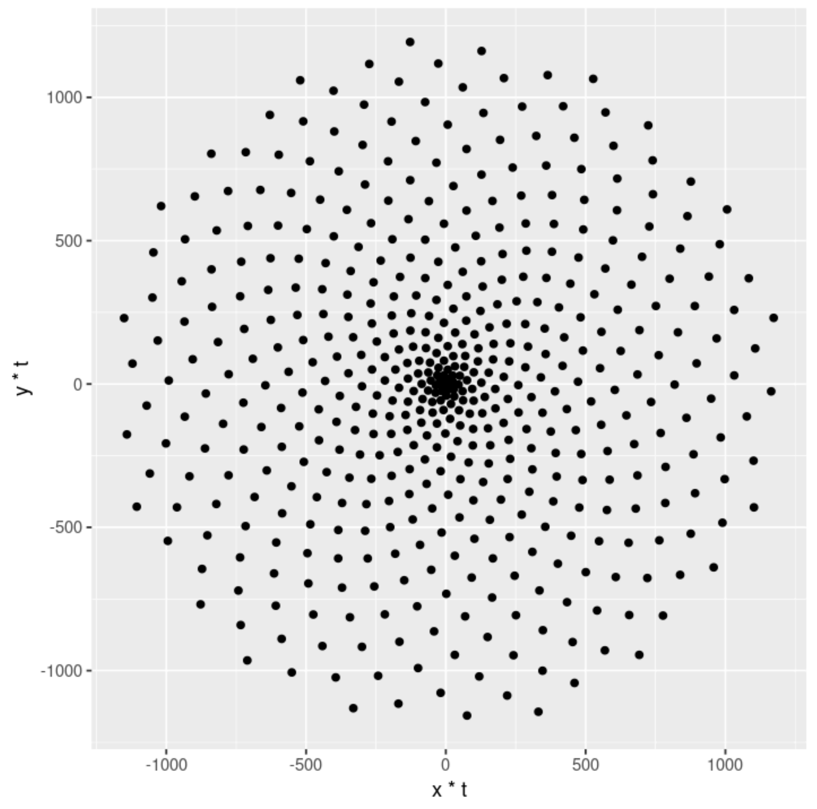
`Golden Angle = π(3 - √5)`

Both the Golden Ratio and the Golden Angle appear in unexpected places in nature. Apart of flower petals and plant leaves, you'll find them in seed heads, pine cones, sunflower seeds, shells, spiral galaxies, hurricanes, etc.

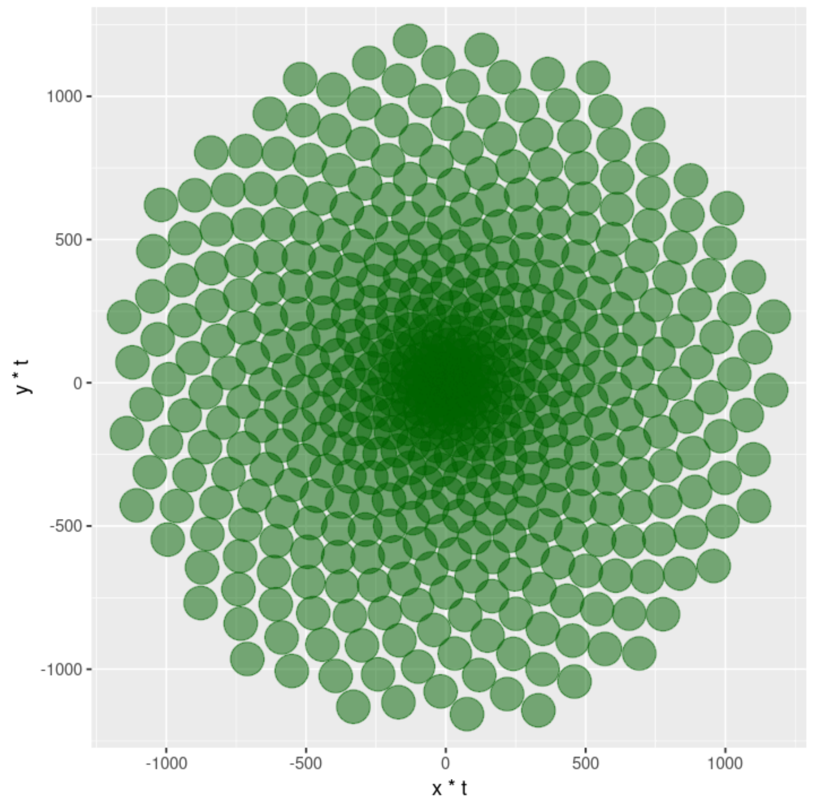
It's time to *spiralize*!

```
points <- 500
angle <- pi * (3 - sqrt(5))
t <- (1:points) * angle
x <- sin(t)
y <- cos(t)
df <- data.frame(t, x, y)
```

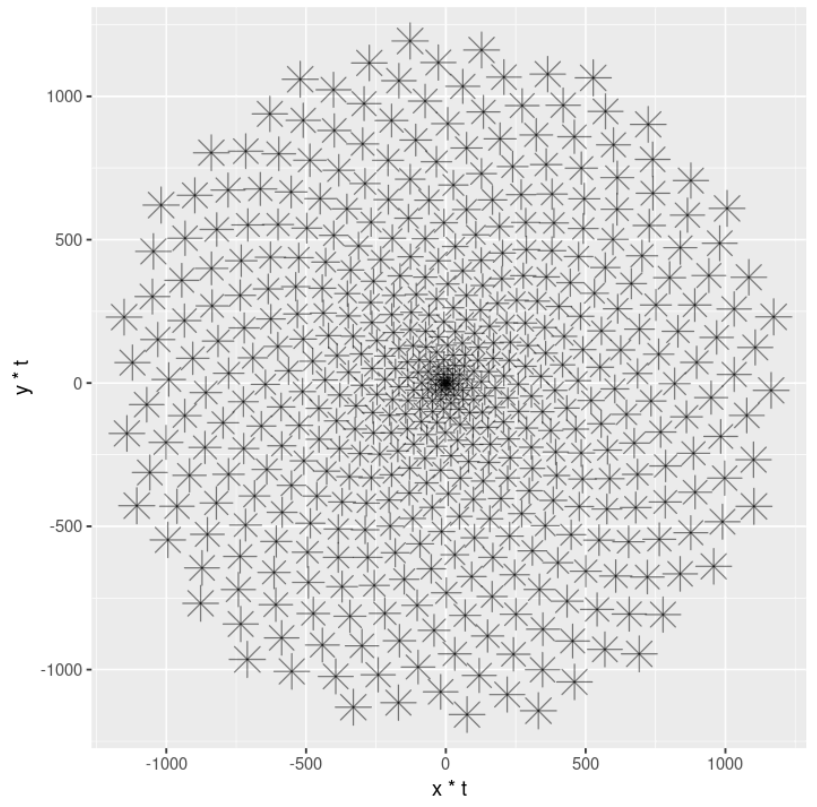
```
spiral_flower <- ggplot(df,aes(x*t,y*t)) + geom_point()
spiral_flower
```



```r
tidy_flower <- ggplot(df,aes(x*t,y*t)) + geom_point(size=8, alpha = 0.5, color="darkgreen")
tidy_flower
```



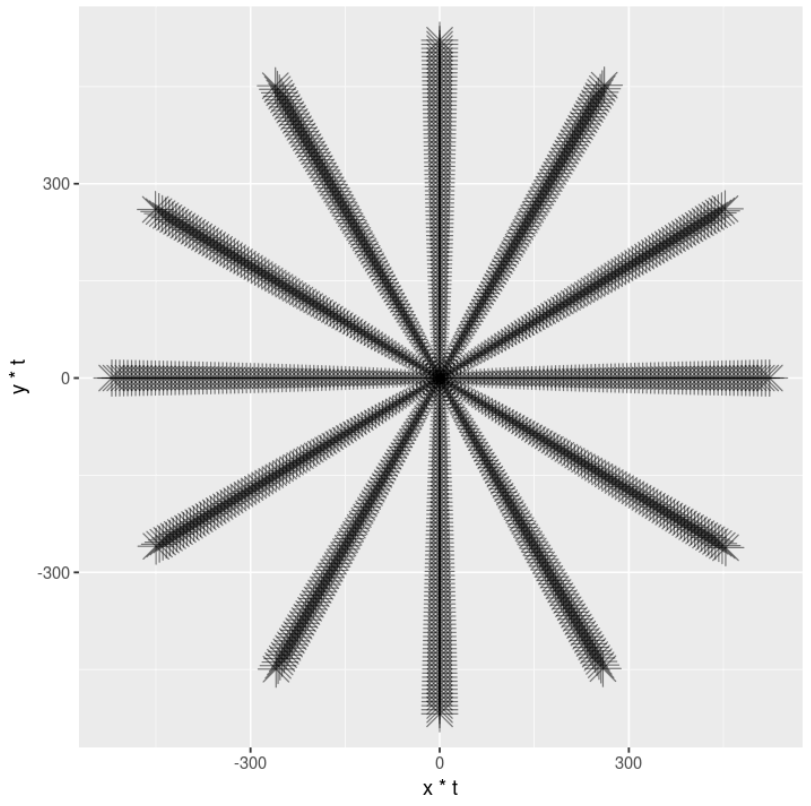
```r
dandelion_flower <- ggplot(df,aes(x*t,y*t)) + geom_point(aes(size=t), alpha = 0.5, color="black",shape = 8) + theme(legend.position = "none")
dandelion_flower
```



```r
points <- 1000
angle <- pi/6
t <- (1:points) * angle
x <- sin(t)
y <- cos(t)
df <- data.frame(t, x, y)

imaginary_flower  <-ggplot(df,aes(x*t,y*t)) + geom_point(aes(size=t), alpha = 0.5, color="black",shape = 8) + theme(legend.position = "none")

imaginary_flower
```


300
0
-300
-300
0
300
x * t
y * t