P I N K
{programming}

LET'S GET ARTSY

A pink programming workshop
Juulia Suvilehto
2023-05-07

INSTALL ALL THE THINGS

- Copy the code file from slack
  - if you're not yet on slack, raise a hand and a code mentor will help
- Go to the link on slack
- Upload the code file
- Run (click on the little ▸ mark with mouse OR press shift + enter repeatedly) until you reach "End of set-up"

P I N K
{programming}

PLAN FOR THE WORKSHOP

- Get installations started
- A very quick intro to some central programming + plotting concepts
- Testing things together
- Time to test on your own
- Ask for help if you get stuck!

Alternative plan for those feeling brave (+having a bit of coding background):

- Run installations
- Hop to notebook section “Advanced module” and read info there
- Start trying out things on your own
- Ask for help if you get stuck!

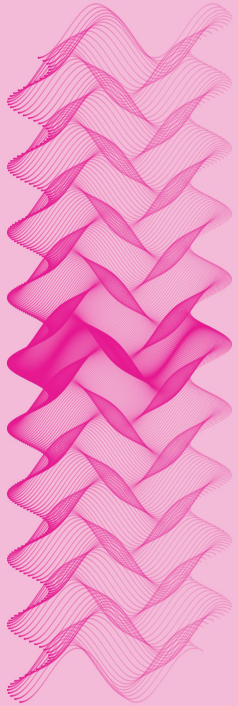
P I N K
{programming}

WHAT IS HAPPENING HERE?

R:
Programming language
Beginner friendly
Pretty plots!

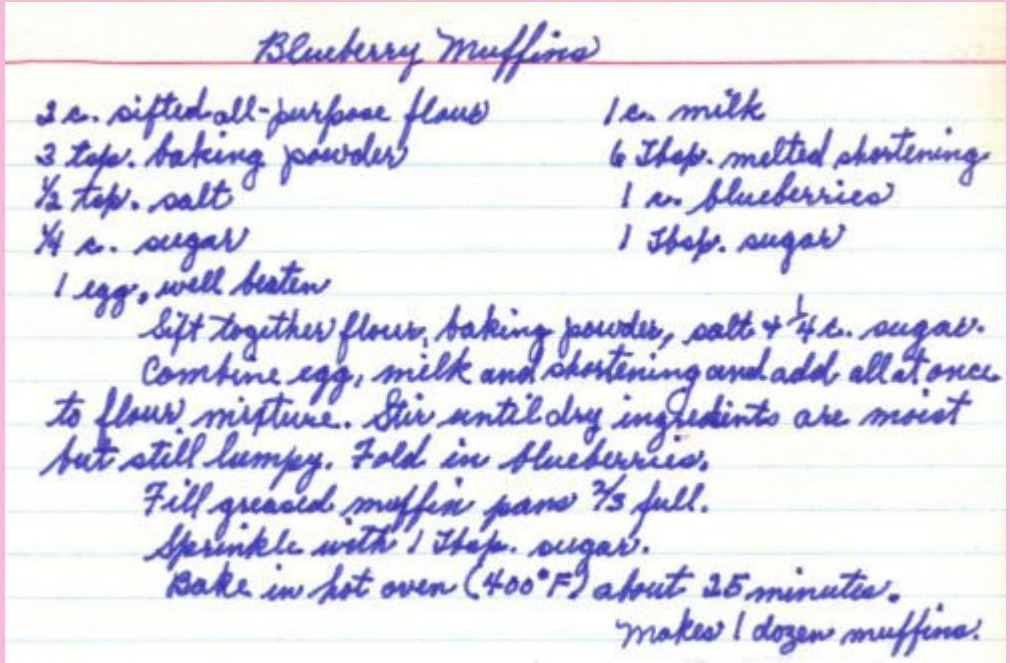
Google Colab:
Online “computer”
Standardised environment

Jupyter notebooks:
Our development environment
How we get google colab to run our R code



P I N K
{programming}

FUNCTIONS ARE LIKE RECIPES



Blueberry Muffins

2 c. sifted all-purpose flour
3 tsp. baking powder
½ tsp. salt
¼ c. sugar
1 egg, well beaten
1 c. milk
6 Tbsp. melted shortening
1 c. blueberries
1 Tbsp. sugar

Sift together flour, baking powder, salt + ¼ c. sugar. Combine egg, milk and shortening and add all at once to flour mixture. Stir until dry ingredients are moist but still lumpy. Fold in blueberries.

Fill greased muffin pans ⅔ full. Sprinkle with 1 Tbsp. sugar. Bake in hot oven (400°F) about 25 minutes.

Makes 1 dozen muffins.

P I N K
{programming}

FUNCTIONS ARE LIKE RECIPES

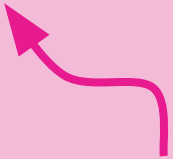
```
bake_blueberry_muffins(servings = 8)
```

P I N K
{programming}

FUNCTIONS ARE LIKE RECIPES

```
bake_blueberry_muffins(servings = 8)
```

This is called a **function**
A function call always has () after it



Inside the () you place **arguments**
Arguments are extra instructions to the function

P I N K
{programming}

FUNCTIONS ARE LIKE RECIPES

```
bake_blueberry_muffins(servings = 8,
                       frosting_color = 'blue')
```

This is called a **function**
A function call always has () after it

Inside the () you place **arguments**
Arguments are extra instructions to the function

P I N K
{programming}

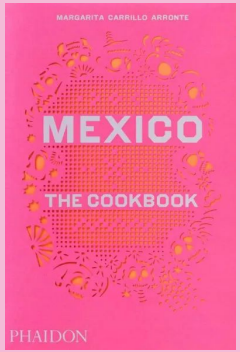
FUNCTIONS ARE LIKE RECIPES

```
bake_blueberry_muffins(servings = 8,
                       frosting_color = 'blue')
```

This is called the function "output"

P I N K
{programming}

PACKAGES ARE LIKE COOKBOOKS



A package
"mexico_recipes"

install.packages(mexico_recipes)

library(mexico_recipes)

P I N K
{programming}

P I N K
{programming}

HANDS-ON PRACTICE TIME!

VARIABLES ARE HANDY HANDLES

- Variables are like names you give things you may want to use later
- They can take many different values and you don't necessarily need to know which value they will eventually get when you write your code
- Variables get assigned a value in R like so

my_favorite_color <- "pink"

Variable name assign operator value

P I N K
{programming}

VARIABLES ARE HANDY HANDLES

```
number_of_guests <- 12
my_favorite_color <- “pink”

bake_blueberry_muffins(servings = number_of_guests,
                       frosting_color  = my_favorite_color)
```

P I N K
{programming}

VARIABLES ARE HANDY HANDLES

```
number_of_guests <- 12
my_favorite_color <- “pink”
party_muffins <- bake_blueberry_muffins(servings = number_of_guests,
                                        frosting_color  = my_favorite_color)
add_sprinkles(party_muffins)
```

EVERYTHING HAS A TYPE

```
number_of_guests <- 12
my_favorite_color <- “pink”
want_sprinkles <- TRUE
frosting_color <- my_favorite_color
```

What type	

P I N K
{programming}

QUOTES ARE IMPORTANT FOR TYPES

```
number_of_guests <- 12
my_favorite_color <- “pink”
want_sprinkles <- TRUE
frosting_color <- my_favorite_color
```

What type	
Numeric	
String (text)	
boolean	
variable	

P I N K
{programming}

P I N K
{programming}

TRY RUNNING THE SECTION

“Let's work with variables”

COMPUTERS ARE STUPID -> ERRORS!

- They happen
- A lot
- It's not dangerous to get an error; it just means the computer didn't understand what you want
- Often the error message will try to help you fix whatever went wrong
- If you don't understand what it says, try googling

  R error <insert your error message here>
- You can always ask a code mentor to take a look at your errors with you!

P I N K
{programming}

P I N K
{programming}

ART FROM RANDOMNESS AND EQUATIONS

COMPUTERS ARE REALLY BAD AT RANDOM

This is good news; we can use something called a ‘random seed’ to repeat same pseudorandom process multiple times!

P I N K
{programming}

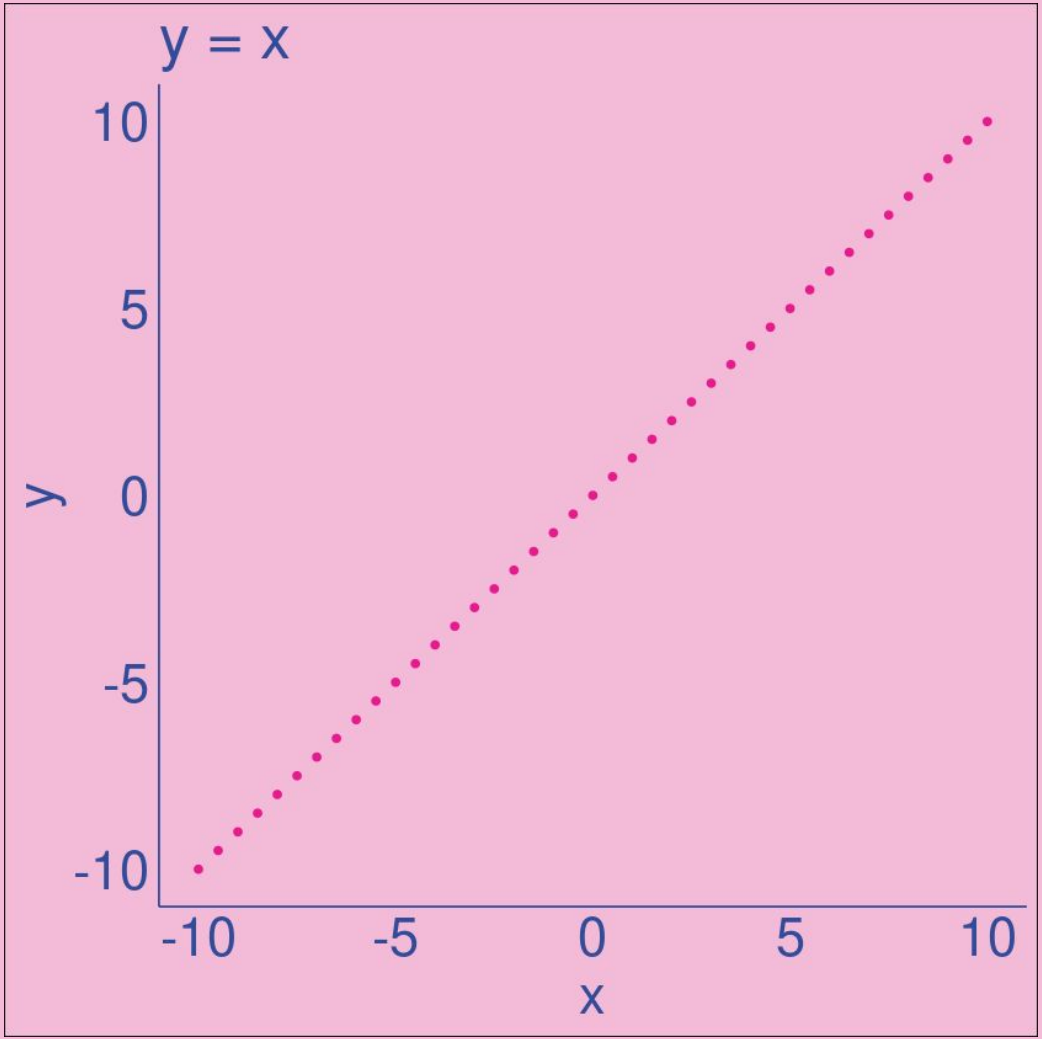
COMPUTERS ARE REALLY BAD AT RANDOM

…instead, they do an internal time warp dance.

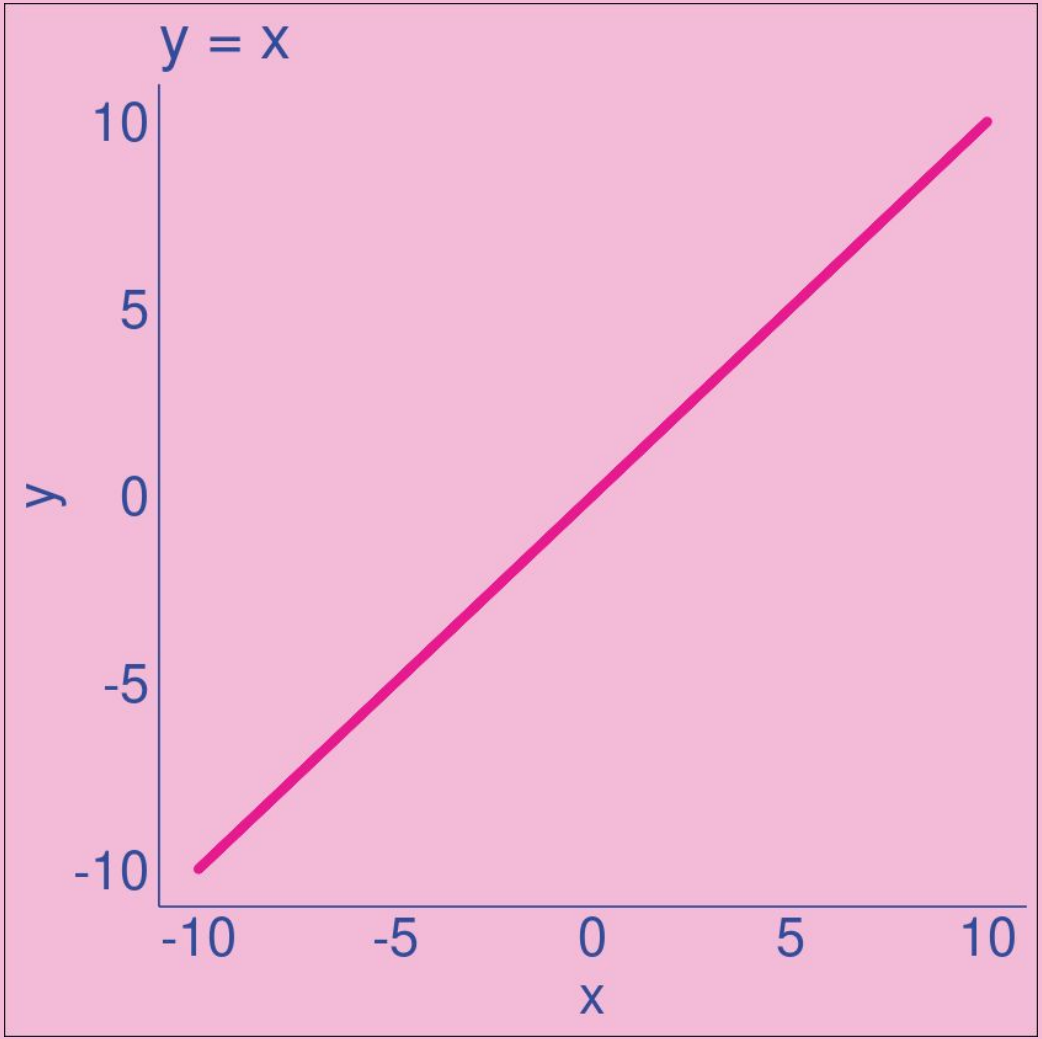
By setting a “random seed” we can tell the computer where to start the dance

P I N K
{programming}

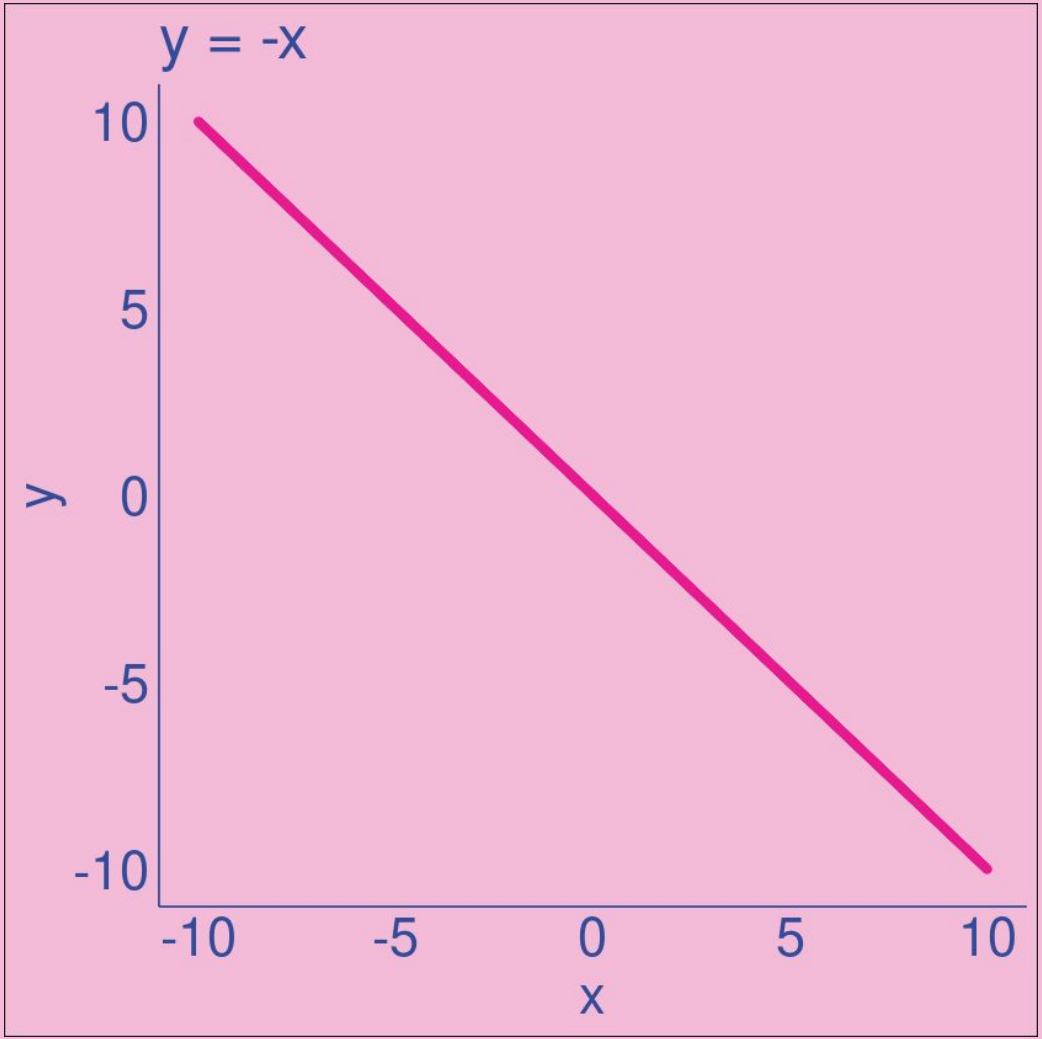
EQUATIONS, THAT SOUNDS REALLY MATHS-Y



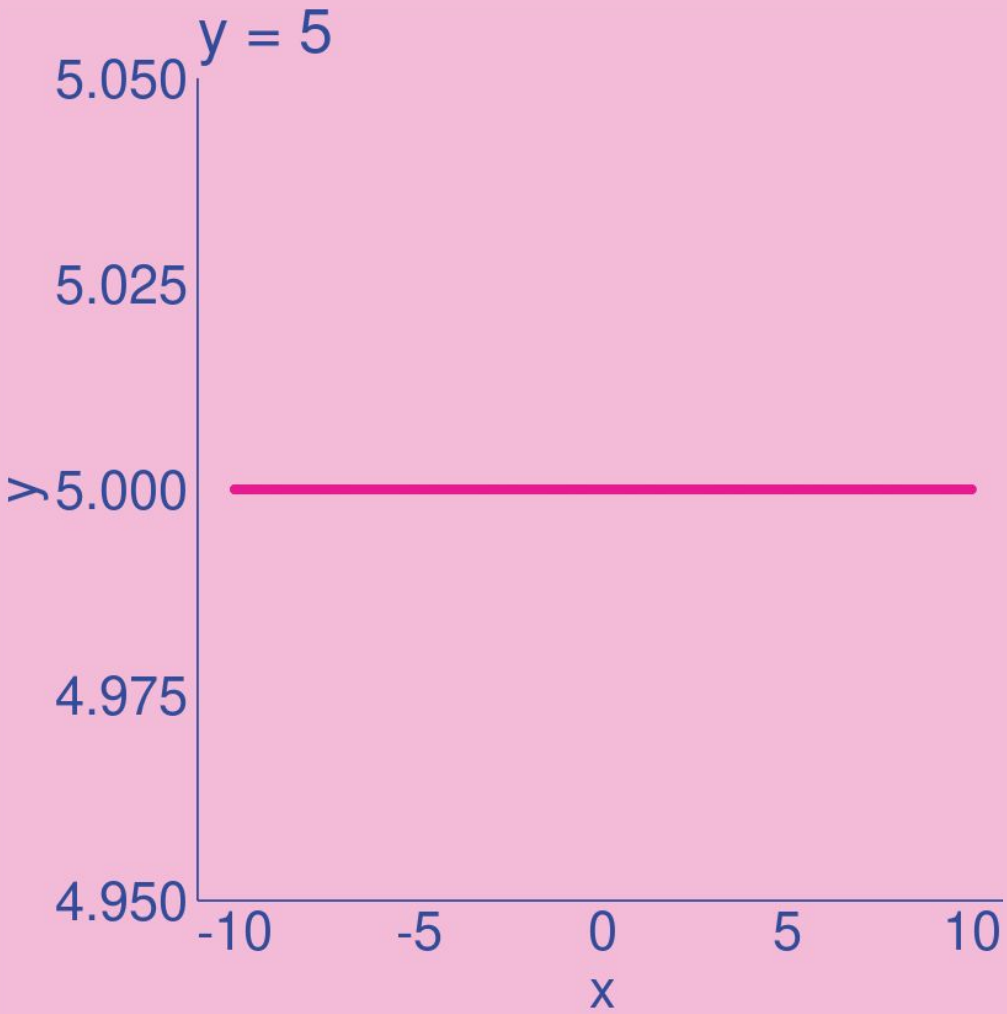
P I N K
{programming}



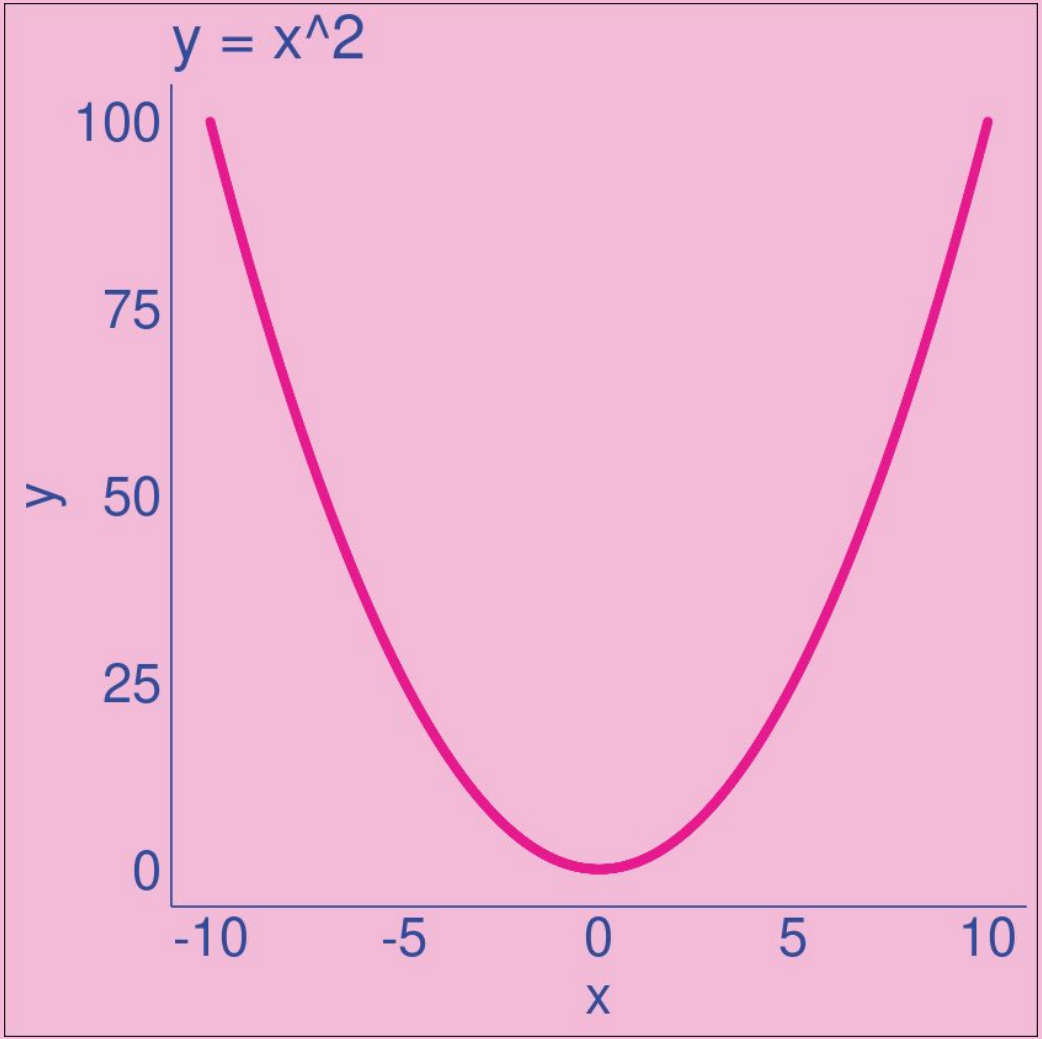
P I N K
{programming}



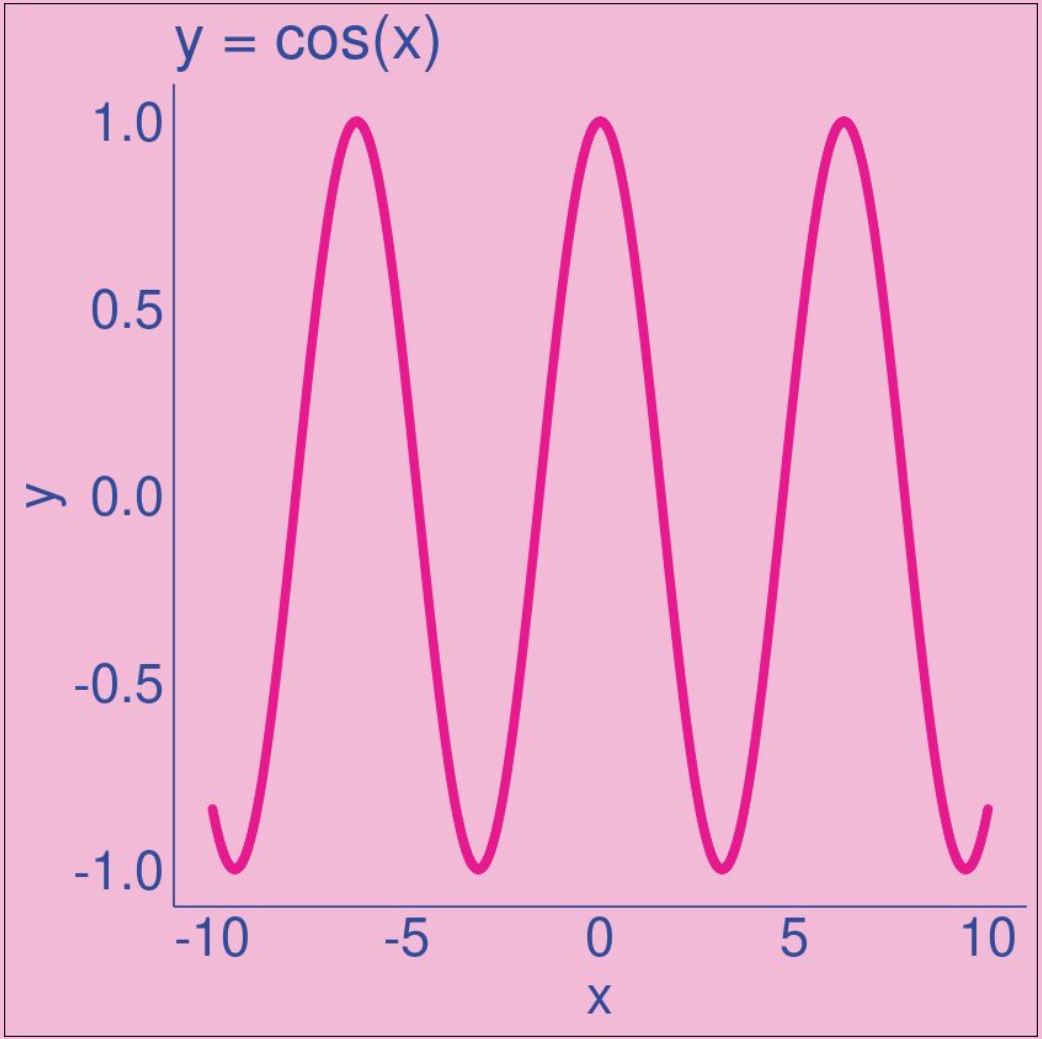
P I N K
{programming}



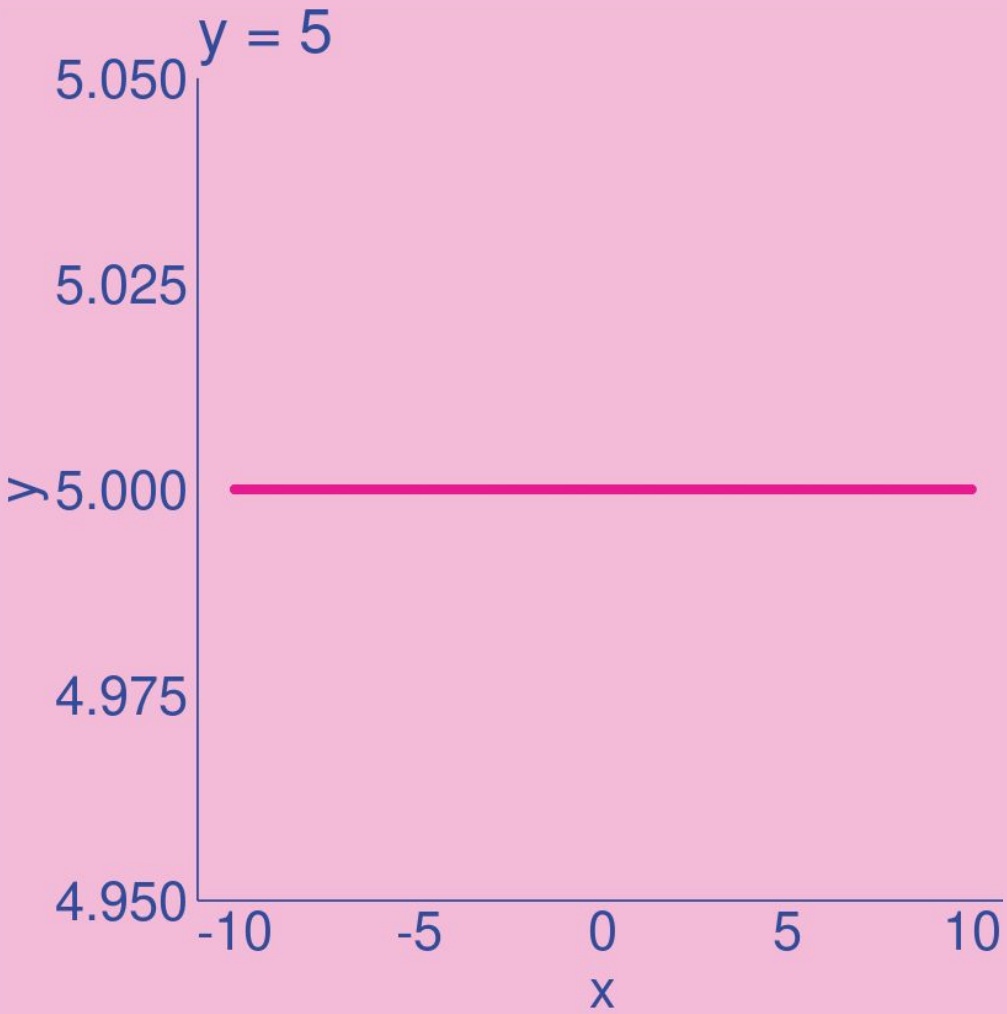
P I N K
{programming}



P I N K
{programming}



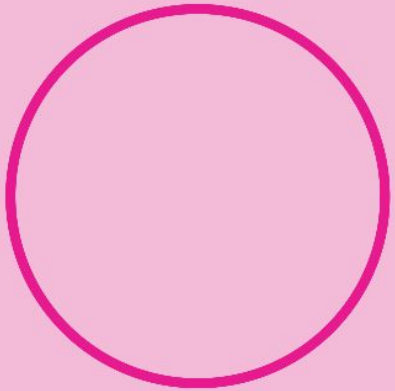
P I N K
{programming}



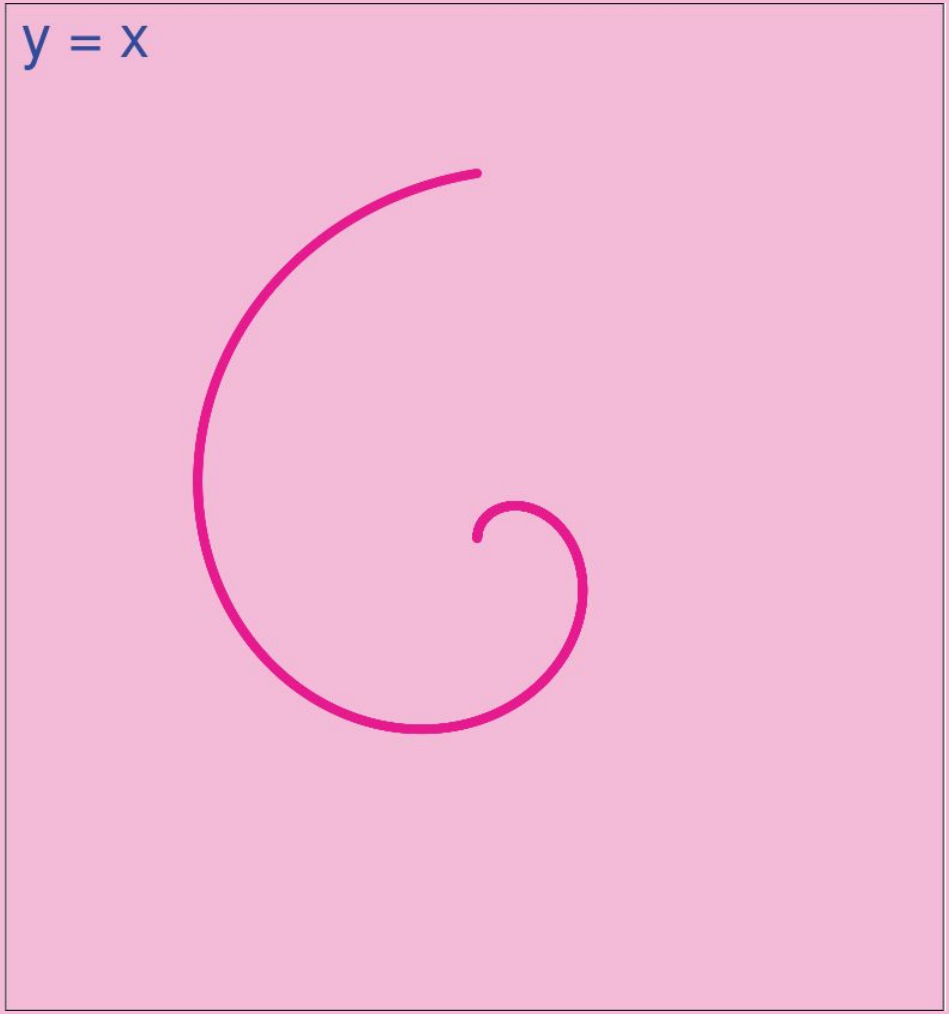
P I N K
{programming}

$y = 5$

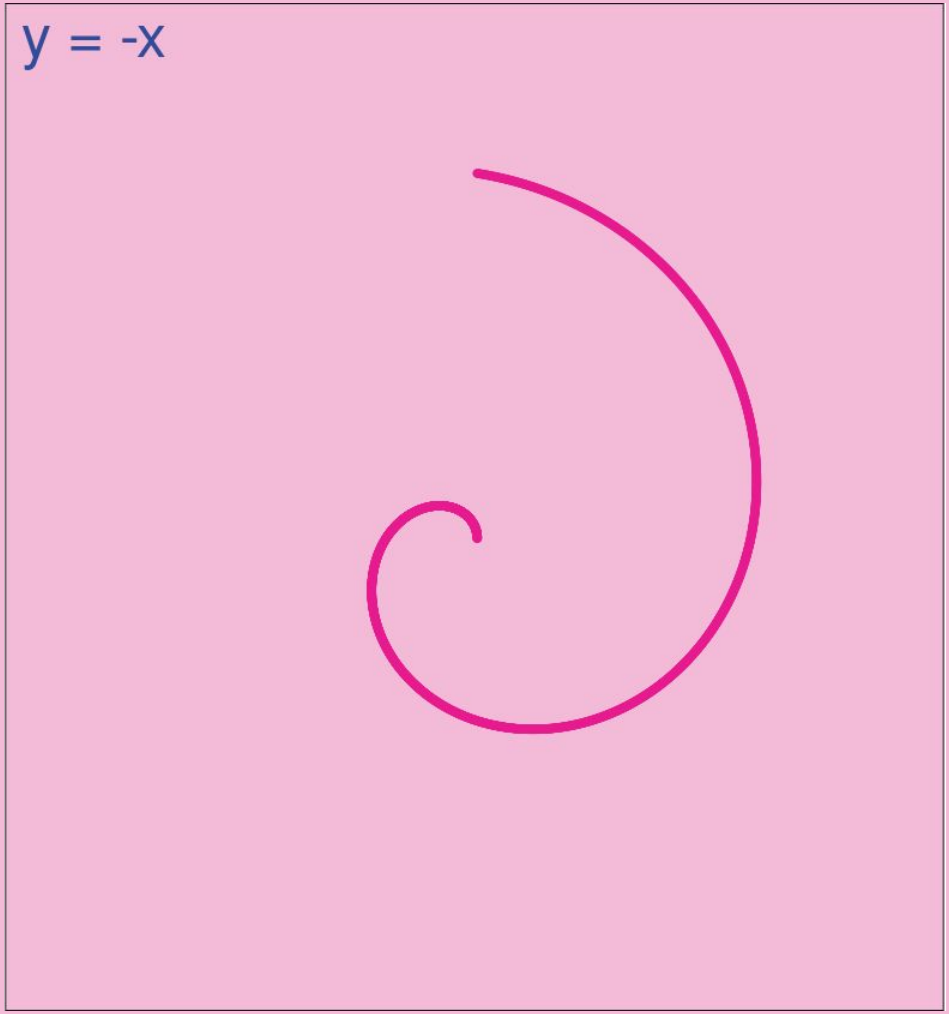
Want something nice and round instead of rectangular?
The polar coordinate system is your friend!



P I N K
{programming}

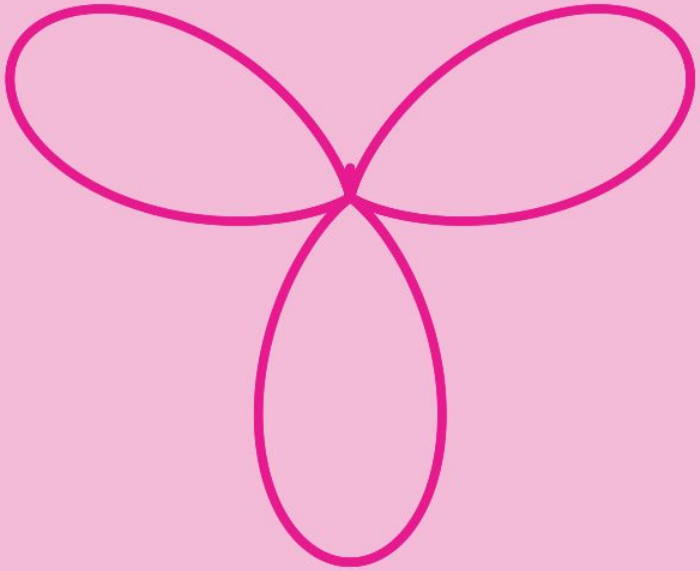


P I N K
{programming}

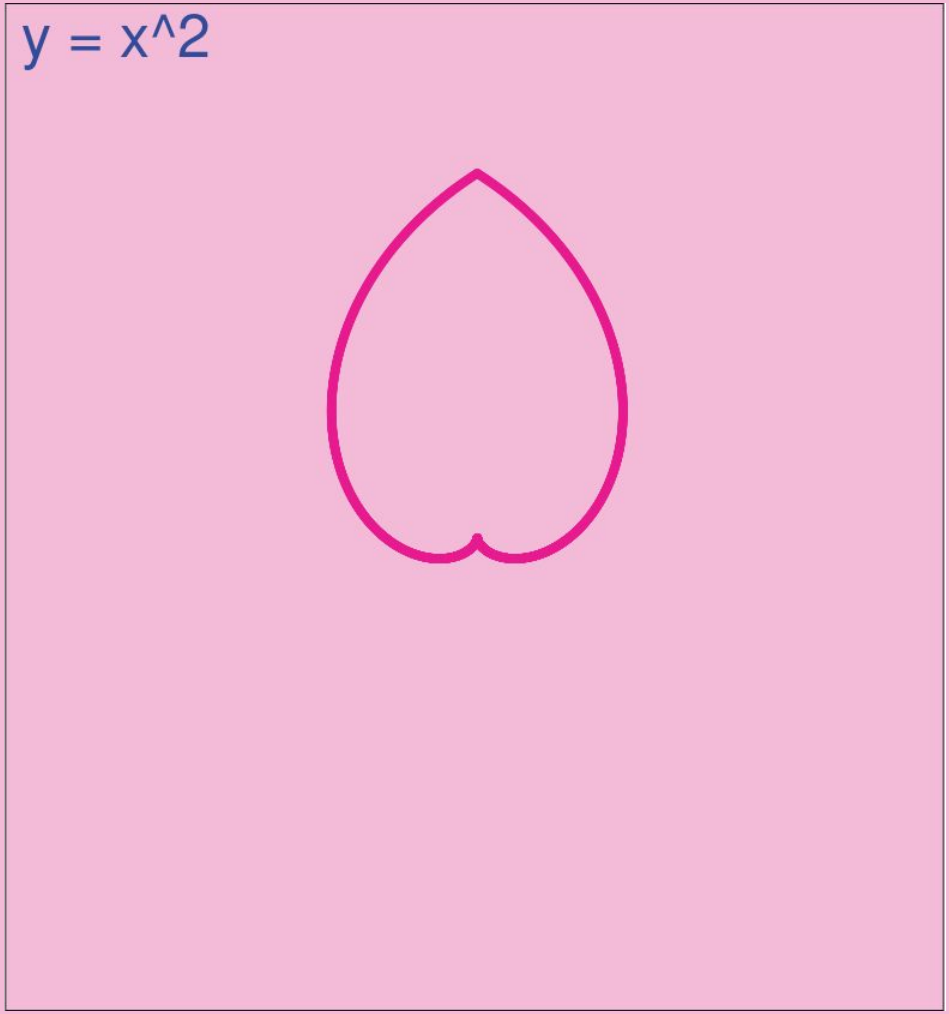


P I N K
{programming}

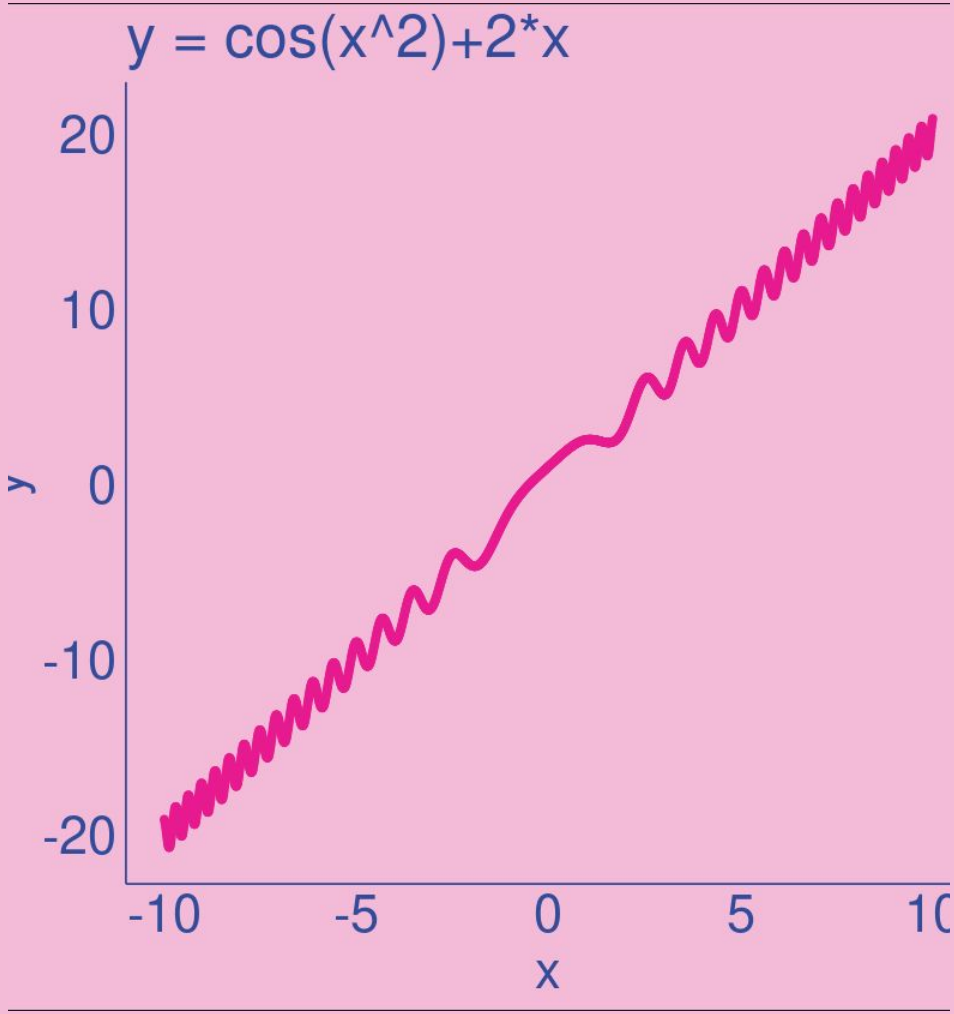
y = cos(x)



P I N K
{programming}

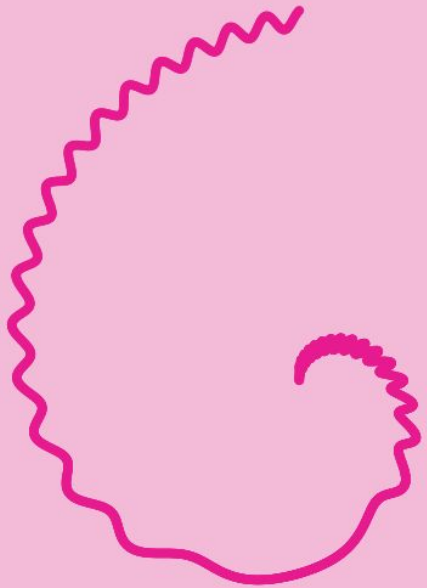


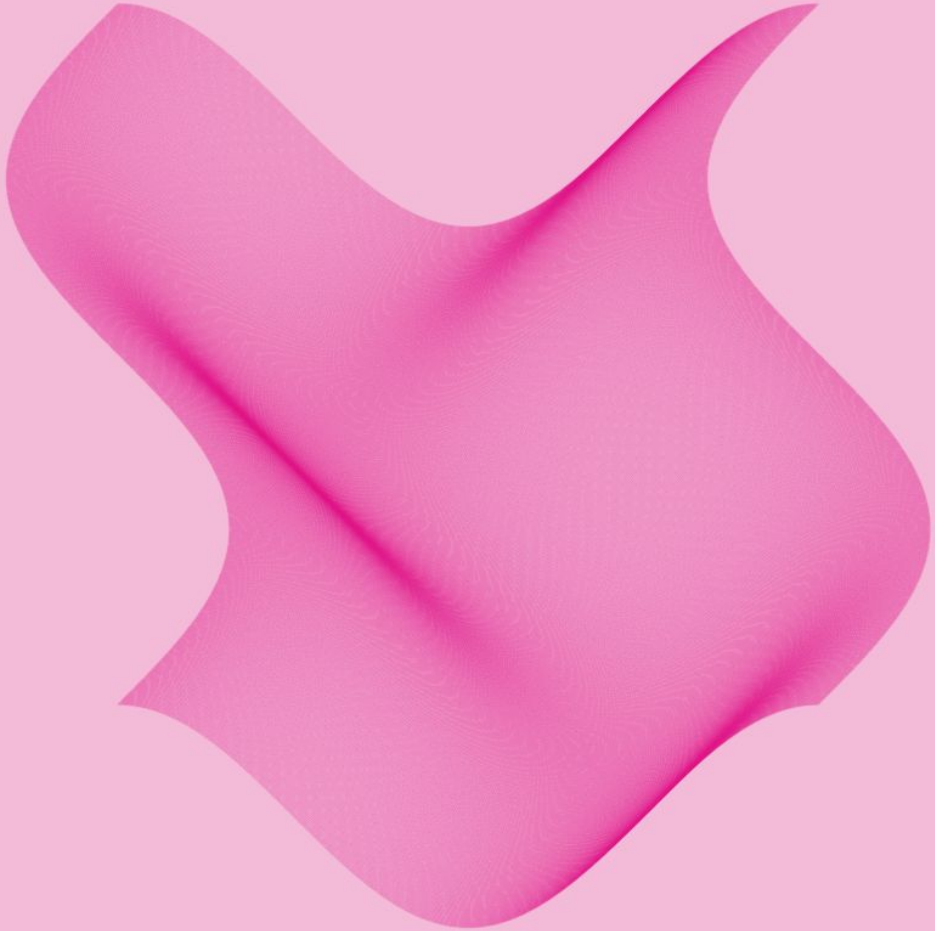
P I N K
{programming}



P I N K
{programming}

$$y = \cos(x^2)+2*x$$





P I N K
{programming}

P I N K

{programming}

P I N K
{programming}

HANDS-ON PRACTICE TIME!

Made something you think looks cool? Save it & share it to slack. We all want to see it!

THIS WAS COOL! WHAT DO I DO NEXT?

- There are links to lots of additional resources at the end of the notebook
- If you are new to programming, maybe take an online course on R first
- Try different functions in the package(s) we worked with today or try out some of the other packages made for generative art
- Maybe start developing your own package if no-one has done one which does what you want it to do?
- Doesn't have to be R: at least Python & Javascript have generative art packages

P I N K
{programming}


That's all