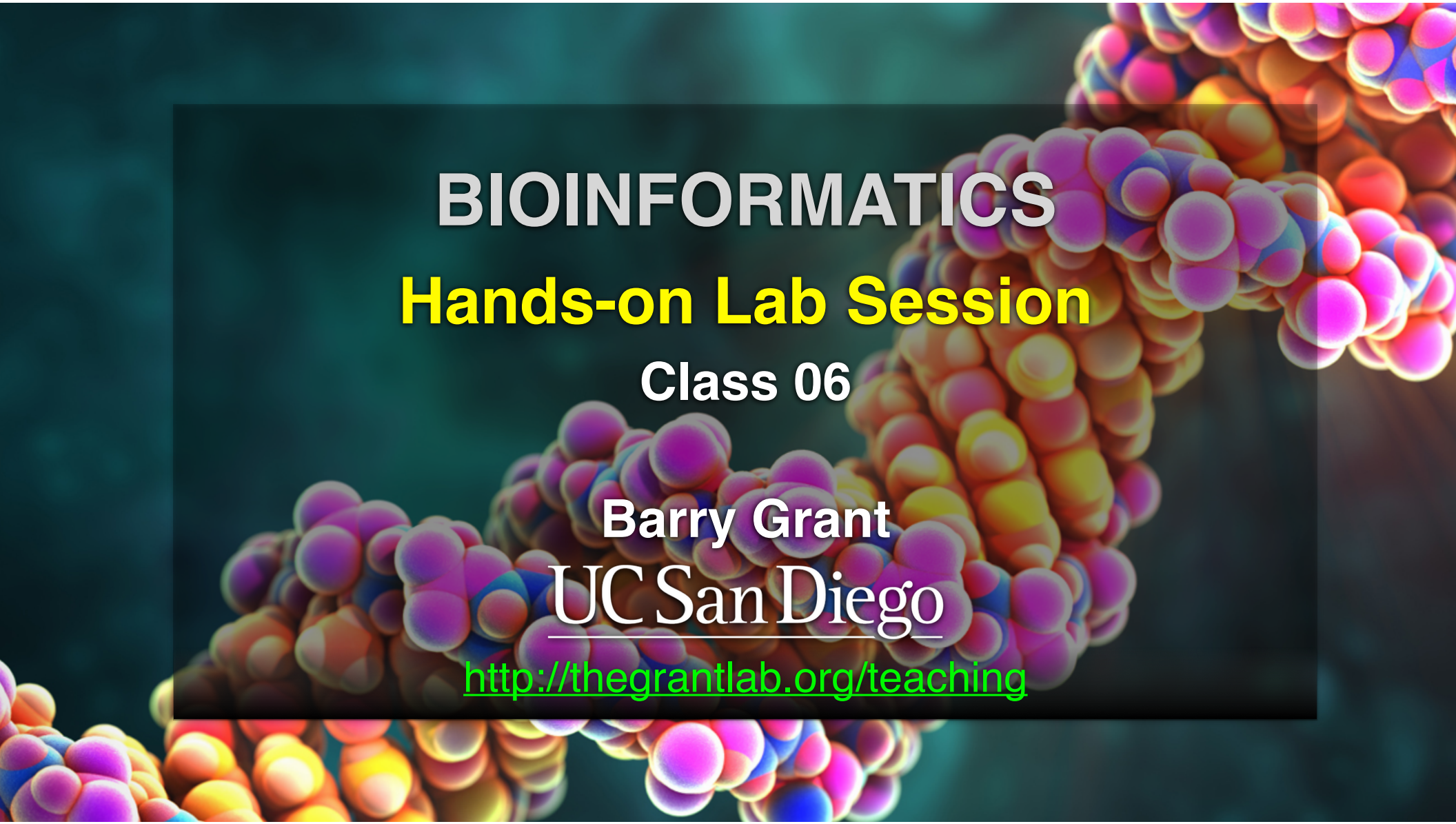

BIOINFORMATICS
Hands-on Lab Session
Class 06
Barry Grant
UC San Diego
http://thegrantlab.org/teaching

function()

# Video Recap:

- Covered the When, Why, What and How of writing your own R functions.

...

# Video Recap:

- Covered the **When**, Why, What and How of writing your own R functions.

  - **When**: When you find yourself doing the same thing 3 or more times with repetitive code consider writing a function.

...

# Video Recap:

- Covered the When, **Why**, What and How of writing your own R functions.

➡ **Why**:

1. Makes the purpose of the code more clear
2. Reduces mistakes from copy/paste
3. Makes updating your code easer
4. Reduces code duplication and facilitates re-use.

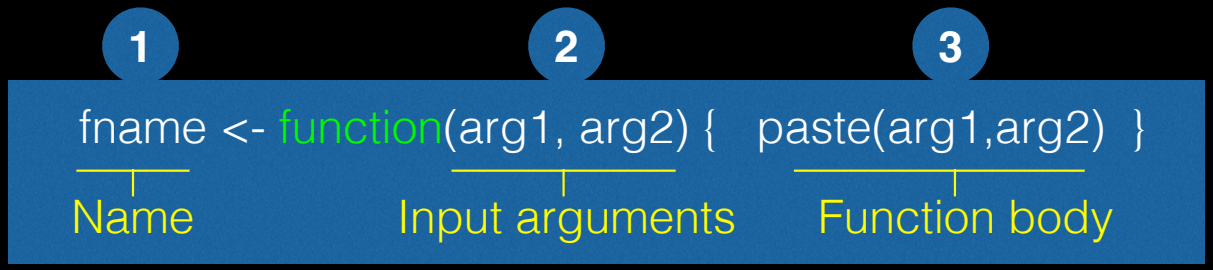
...

# Video Recap:

- Covered the When, Why, **What** and How of writing your own R functions.

  - **What**: A function is defined with:

    1. A user selected name,
    2. A comma separated set of input arguments, and
    3. Regular R code for the function body



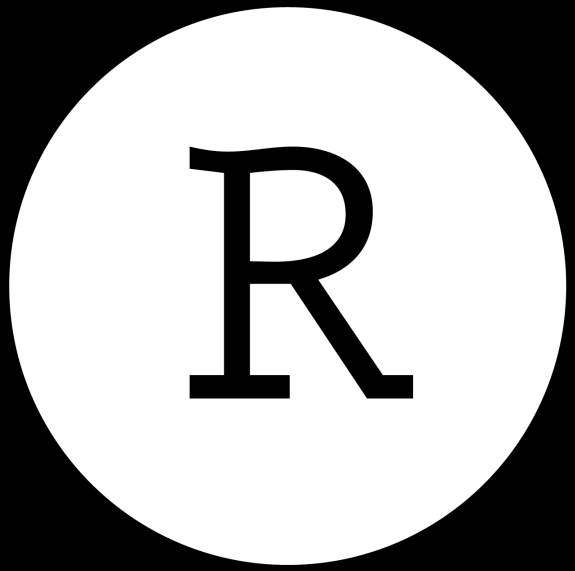

...

# Video Recap:

- **How**: Follow a step-by-step procedure to go from working code snippet to refined and tested function.

1. Start with a simple problem and write a working snippet of code.
2. Rewrite for clarity and to reduce duplication
3. Then, and only then, turn into an initial function
4. Test on small well defined input
5. Report on potential problem by failing early and loudly!

Lab session!



RStudio®

Create a new **Project** for class06

N.B. Open a new **Quarto** document
(Our goal is to make a PDF report with notes and plots)


File > **New Project** > New Directory…
File > New File > **Quarto Document**…

# Your turn...

Lab session!

- Write a function grade() to determine an overall grade from a vector of student homework assignment scores dropping the lowest single assignment score.

```
# student 1
c(100, 100, 100, 100, 100, 100, 100, 90)

# student 2
c(100, NA, 90, 90, 90, 90, 97, 80)
```

# Your turn...

Lab session!

- Write a function grade() to determine an overall grade from a vector of student homework assignment scores dropping the lowest single assignment score.

```
# student 1
c(100, 100, 100, 100, 100, 100, 100, 90)

# student 2
c(100, NA, 90, 90, 90, 90, 97, 80)

# now grade all students in an example class
url <- "https://tinyurl.com/gradeinput"
```

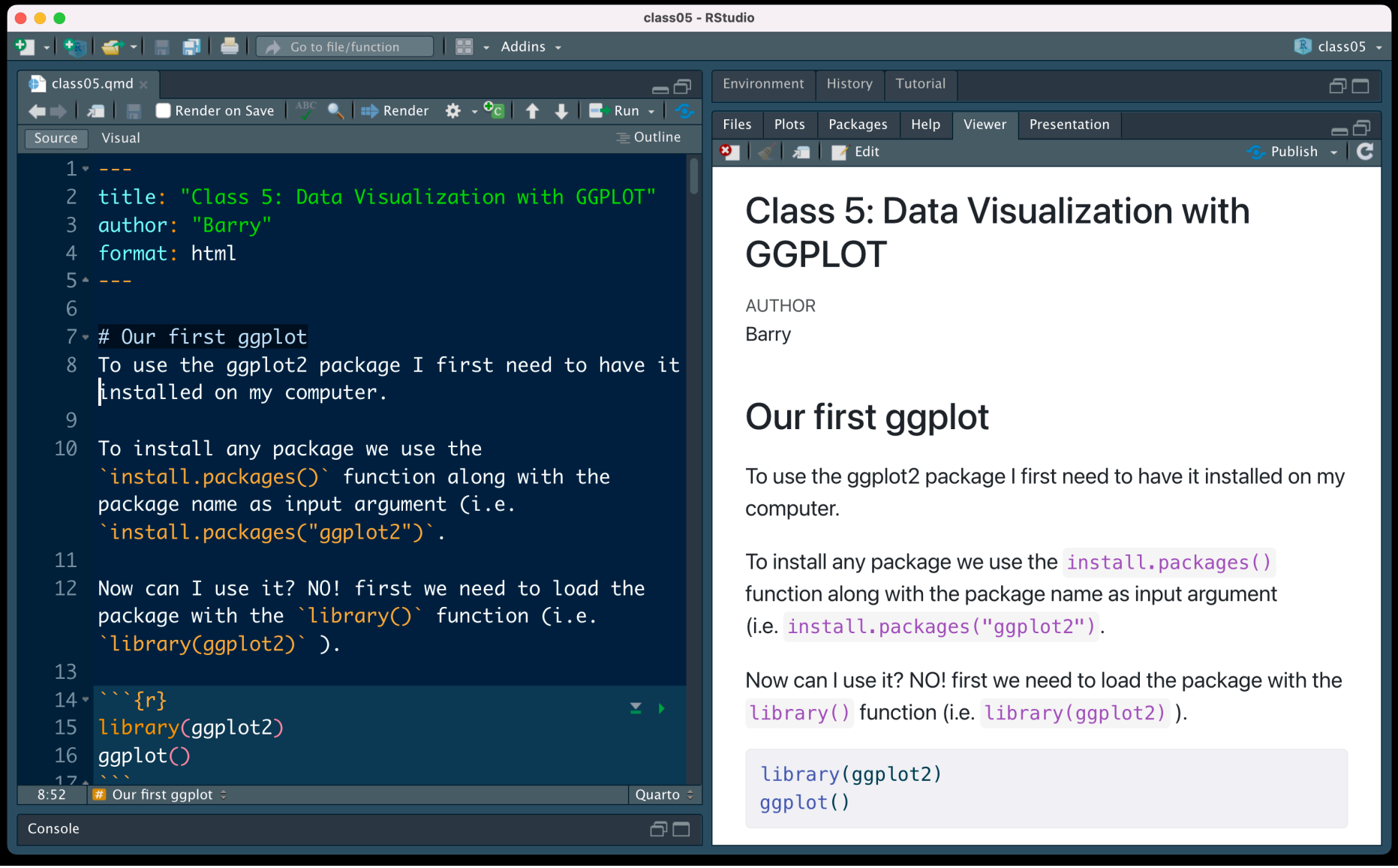
# Your turn…

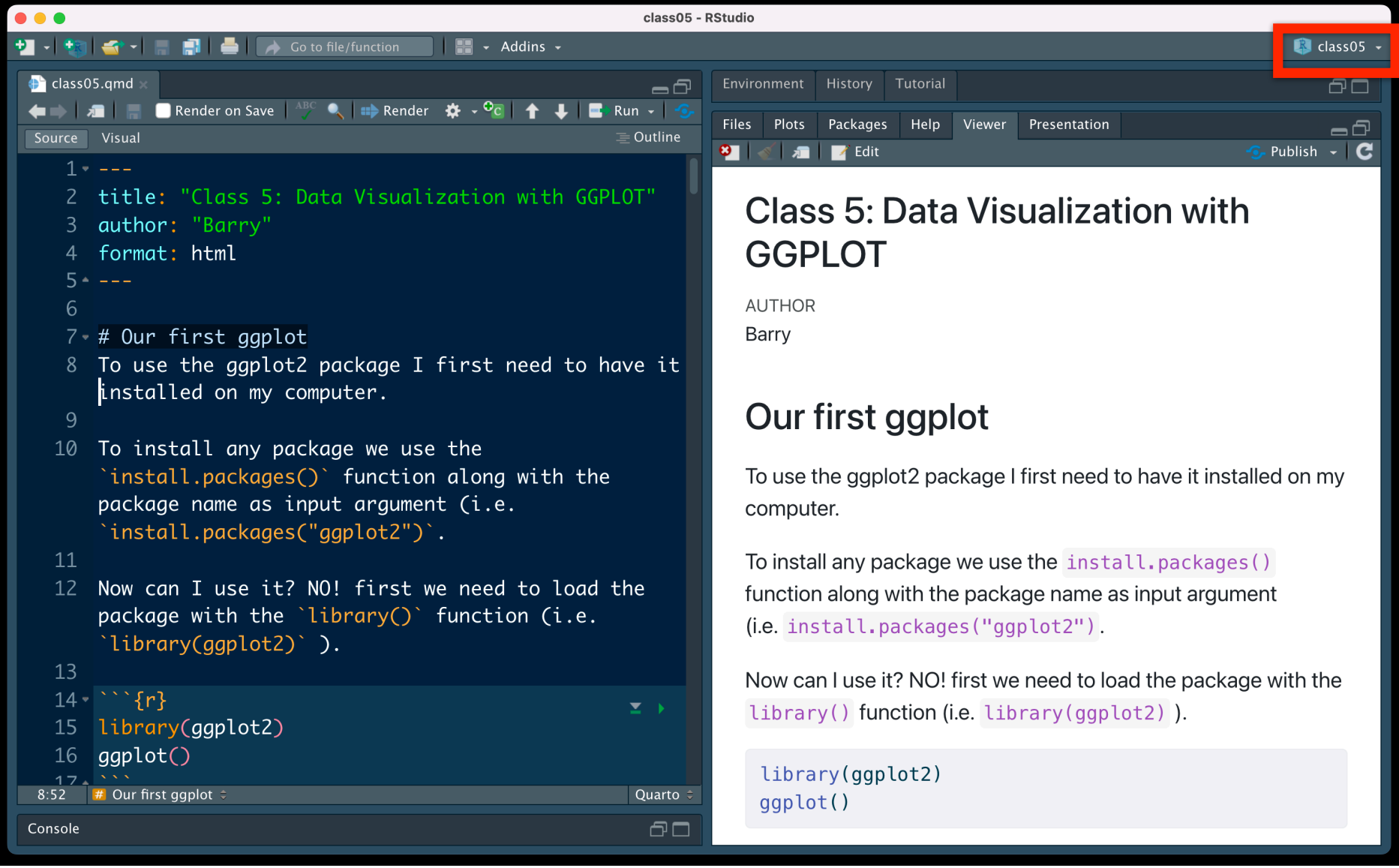
Lab session!

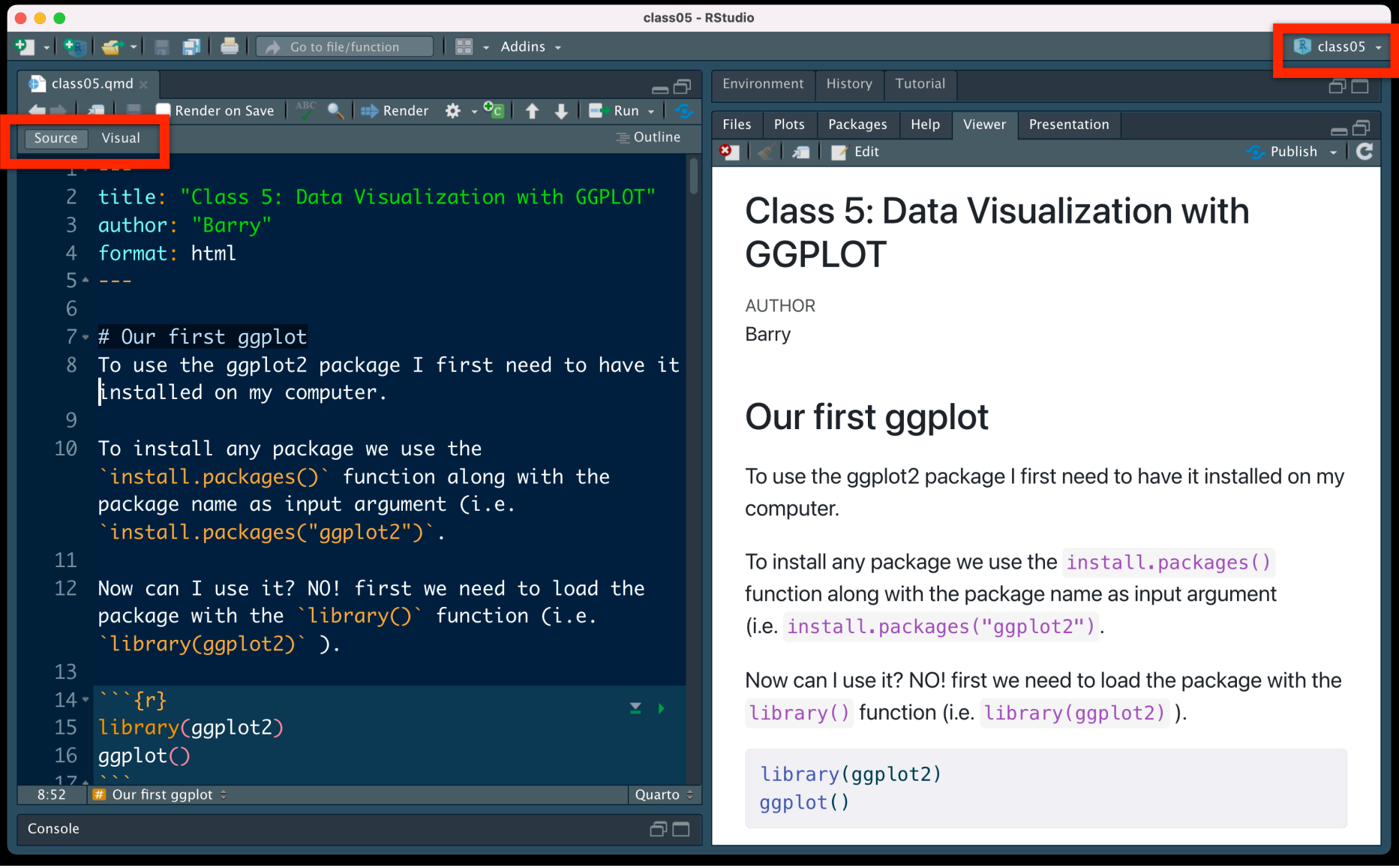
- Write your first R function, `add()` to add some numbers
- Write a second function, `generate_dna()` to return nucleotide sequences of a user specified length.
- Write a third function, `generate_protein()` to return protein sequences of different lengths and test whether these sequences are unique in nature.

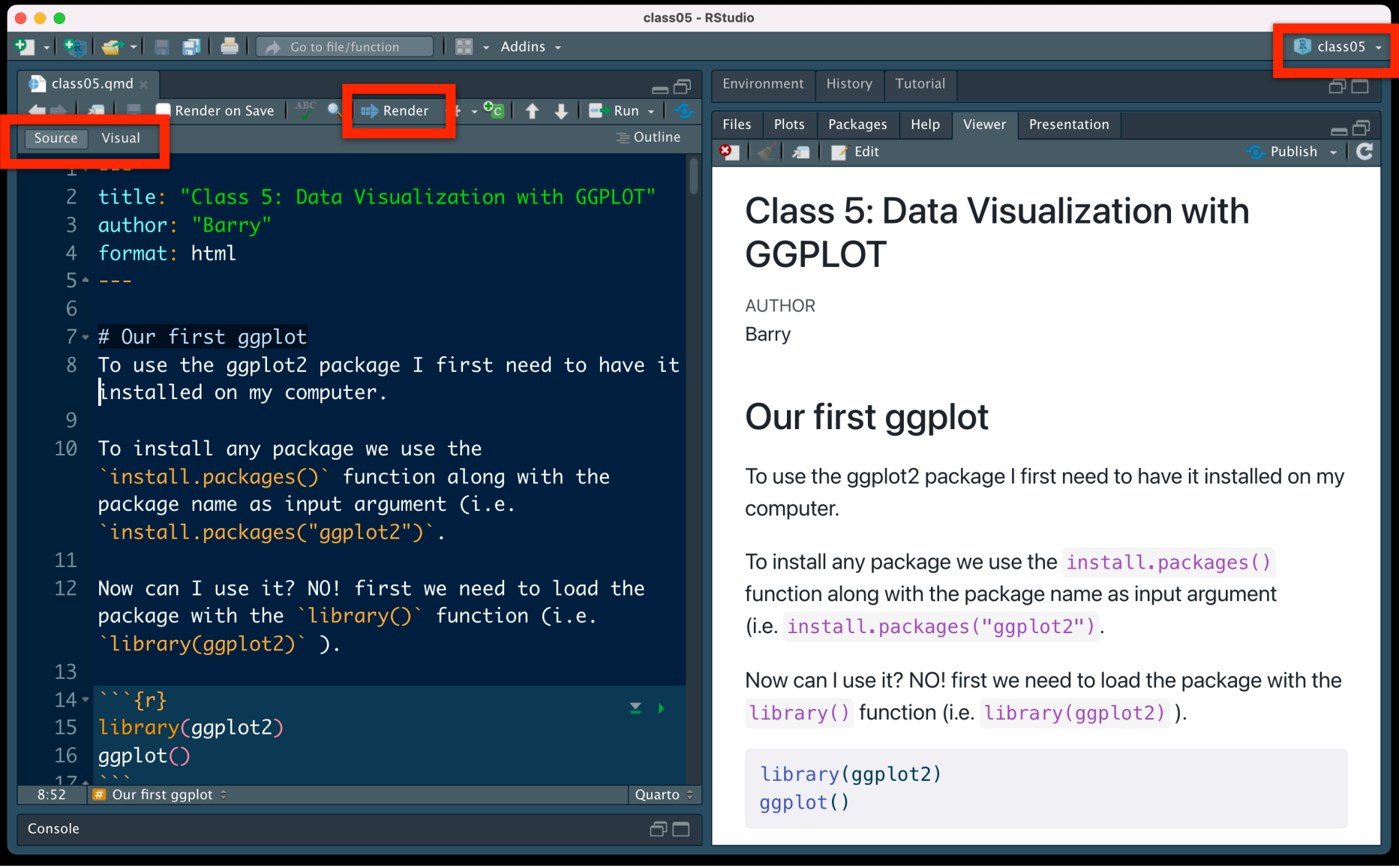

File > New File > **Quarto Document**

N.B. we will use this to save our work and make a **lab report**


class05 - RStudio
Go to file/function
Addins
class05
class05.qmd
Render on Save
Render
Run
Source
Visual
Outline
1 ---
2 title: "Class 5: Data Visualization with GGPLOT"
3 author: "Barry"
4 format: html
5 ---
6
7 # Our first ggplot
8 To use the ggplot2 package I first need to have it installed on my computer.
9
10 To install any package we use the `install.packages()` function along with the package name as input argument (i.e. `install.packages("ggplot2")`.
11
12 Now can I use it? NO! first we need to load the package with the `library()` function (i.e. `library(ggplot2)` ).
13
14 ```{r}
15 library(ggplot2)
16 ggplot()
17 ```
8:52
Our first ggplot
Quarto
Console
Environment
History
Tutorial
Files
Plots
Packages
Help
Viewer
Presentation
Edit
Publish
Class 5: Data Visualization with GGPLOT
AUTHOR
Barry
Our first ggplot
To use the ggplot2 package I first need to have it installed on my computer.
To install any package we use the install.packages() function along with the package name as input argument (i.e. install.packages("ggplot2").
Now can I use it? NO! first we need to load the package with the library() function (i.e. library(ggplot2) ).
library(ggplot2)
ggplot()


class05 - RStudio
Go to file/function
Addins
class05
class05.qmd
Render on Save
Render
Run
Source
Visual
Outline
1 ---
2 title: "Class 5: Data Visualization with GGPLOT"
3 author: "Barry"
4 format: html
5 ---
6
7 # Our first ggplot
8 To use the ggplot2 package I first need to have it installed on my computer.
9
10 To install any package we use the `install.packages()` function along with the package name as input argument (i.e. `install.packages("ggplot2")`.
11
12 Now can I use it? NO! first we need to load the package with the `library()` function (i.e. `library(ggplot2)` ).
13
14 ```{r}
15 library(ggplot2)
16 ggplot()
17 ```
8:52
Our first ggplot
Quarto
Console
Environment
History
Tutorial
Files
Plots
Packages
Help
Viewer
Presentation
Edit
Publish
Class 5: Data Visualization with GGPLOT
AUTHOR
Barry
Our first ggplot
To use the ggplot2 package I first need to have it installed on my computer.
To install any package we use the install.packages() function along with the package name as input argument (i.e. install.packages("ggplot2").
Now can I use it? NO! first we need to load the package with the library() function (i.e. library(ggplot2) ).
library(ggplot2)
ggplot()


class05 - RStudio
Go to file/function
Addins
class05
class05.qmd
Render on Save
Render
Run
Source
Visual
Outline
Environment
History
Tutorial
Files
Plots
Packages
Help
Viewer
Presentation
Edit
Publish
---
title: "Class 5: Data Visualization with GGPLOT"
author: "Barry"
format: html
---
# Our first ggplot
To use the ggplot2 package I first need to have it installed on my computer.
To install any package we use the `install.packages()` function along with the package name as input argument (i.e. `install.packages("ggplot2")`.
Now can I use it? NO! first we need to load the package with the `library()` function (i.e. `library(ggplot2)` ).
```{r}
library(ggplot2)
ggplot()
```
8:52
Our first ggplot
Quarto
Console
Class 5: Data Visualization with GGPLOT
AUTHOR
Barry
Our first ggplot
To use the ggplot2 package I first need to have it installed on my computer.
To install any package we use the install.packages() function along with the package name as input argument (i.e. install.packages("ggplot2").
Now can I use it? NO! first we need to load the package with the library() function (i.e. library(ggplot2) ).
library(ggplot2)
ggplot()


class05 - RStudio
Go to file/function
Addins
class05
class05.qmd
Render on Save
Render
Run
Source
Visual
Outline
Environment
History
Tutorial
Files
Plots
Packages
Help
Viewer
Presentation
Edit
Publish
---
title: "Class 5: Data Visualization with GGPLOT"
author: "Barry"
format: html
---
# Our first ggplot
To use the ggplot2 package I first need to have it installed on my computer.
To install any package we use the `install.packages()` function along with the package name as input argument (i.e. `install.packages("ggplot2")`.
Now can I use it? NO! first we need to load the package with the `library()` function (i.e. `library(ggplot2)` ).
```{r}
library(ggplot2)
ggplot()
```
8:52
Our first ggplot
Quarto
Console
Class 5: Data Visualization with GGPLOT
AUTHOR
Barry
Our first ggplot
To use the ggplot2 package I first need to have it installed on my computer.
To install any package we use the install.packages() function along with the package name as input argument (i.e. install.packages("ggplot2").
Now can I use it? NO! first we need to load the package with the library() function (i.e. library(ggplot2) ).
library(ggplot2)
ggplot()


Follow Along!
quarto® demo

# Lab sheet

lab_class06.pdf
Page 1 of 2

Search

## R Functions Lab (Class 06)

Barry Grant

http://thegrantlab.org/

### Background

In this session you will work through the process of developing your own function for calculating average grades for fictional students in a fictional class.

The process will involve starting slowly with small defined input vectors (where you know what the answer should be). Then building up to work with more complex input vectors (with multiple missing elements).

# And some homework….

Your Homework!

```
library(bio3d)
s1 <- read.pdb("4AKE")  # kinase with drug
s2 <- read.pdb("1AKE")  # kinase no drug
s3 <- read.pdb("1E4Y")  # kinase with drug

s1.chainA <- trim.pdb(s1, chain="A", elety="CA")
s2.chainA <- trim.pdb(s2, chain="A", elety="CA")
s3.chainA <- trim.pdb(s1, chain="A", elety="CA")

s1.b <- s1.chainA$atom$b
s2.b <- s2.chainA$atom$b
s3.b <- s3.chainA$atom$b

plotb3(s1.b, sse=s1.chainA, typ="l", ylab="Bfactor")
plotb3(s2.b, sse=s2.chainA, typ="l", ylab="Bfactor")
plotb3(s3.b, sse=s3.chainA, typ="l", ylab="Bfactor")
```

# Suggested steps for writing your functions

1. Start with a simple problem and get a working snippet of code
2. Rewrite to use temporary variables (e.g. x, y, df, m etc.)
3. Rewrite for clarity and to reduce calculation duplication
4. Turn into an initial function with clear useful names
5. Test on small well defined input and (subsets of) real input
6. Report on potential problem by failing early and loudly!
7. Refine and polish

# **Side-Note**: What makes a good function?

- Correct
- Understandable (remember that functions are for humans and computers)
- Correct + Understandable = **Obviously correct**
- Use sensible names throughout. What does this code do?

```
baz <- foo(df, v=0)
df2 < replace_missing(df, value=0)
```

- Good names make code understandable with minimal context. You should strive for self-explanatory names

# Recap From Last Time:

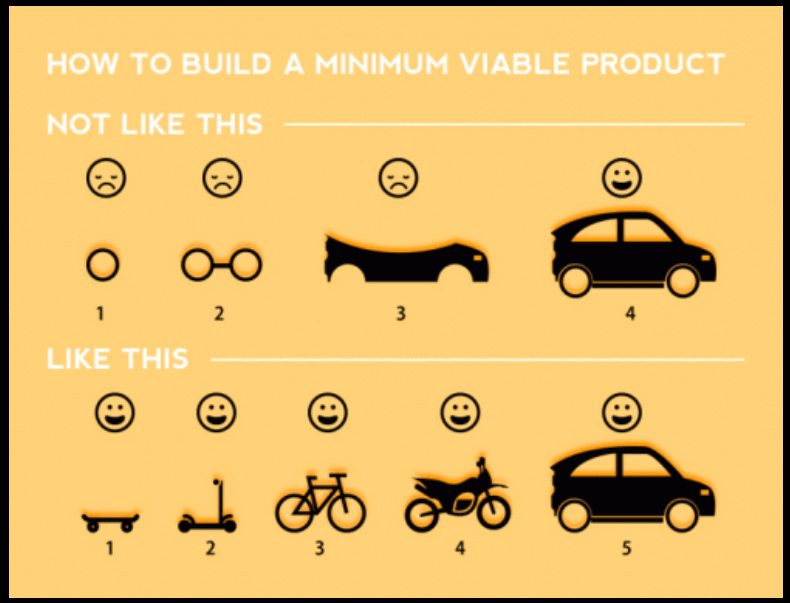
- **How**: Follow a step-by-step procedure to go from working code snippet to refined and tested function.

1. Start with a simple problem and write a working snippet of code.
2. Rewrite for clarity and to reduce duplication
3. Then, and only then, turn into an initial function
4. Test on small well defined input
5. Report on potential problem by failing early and loudly!

...

# Recap…

1. Start with a simple problem and write a working snippet of code.



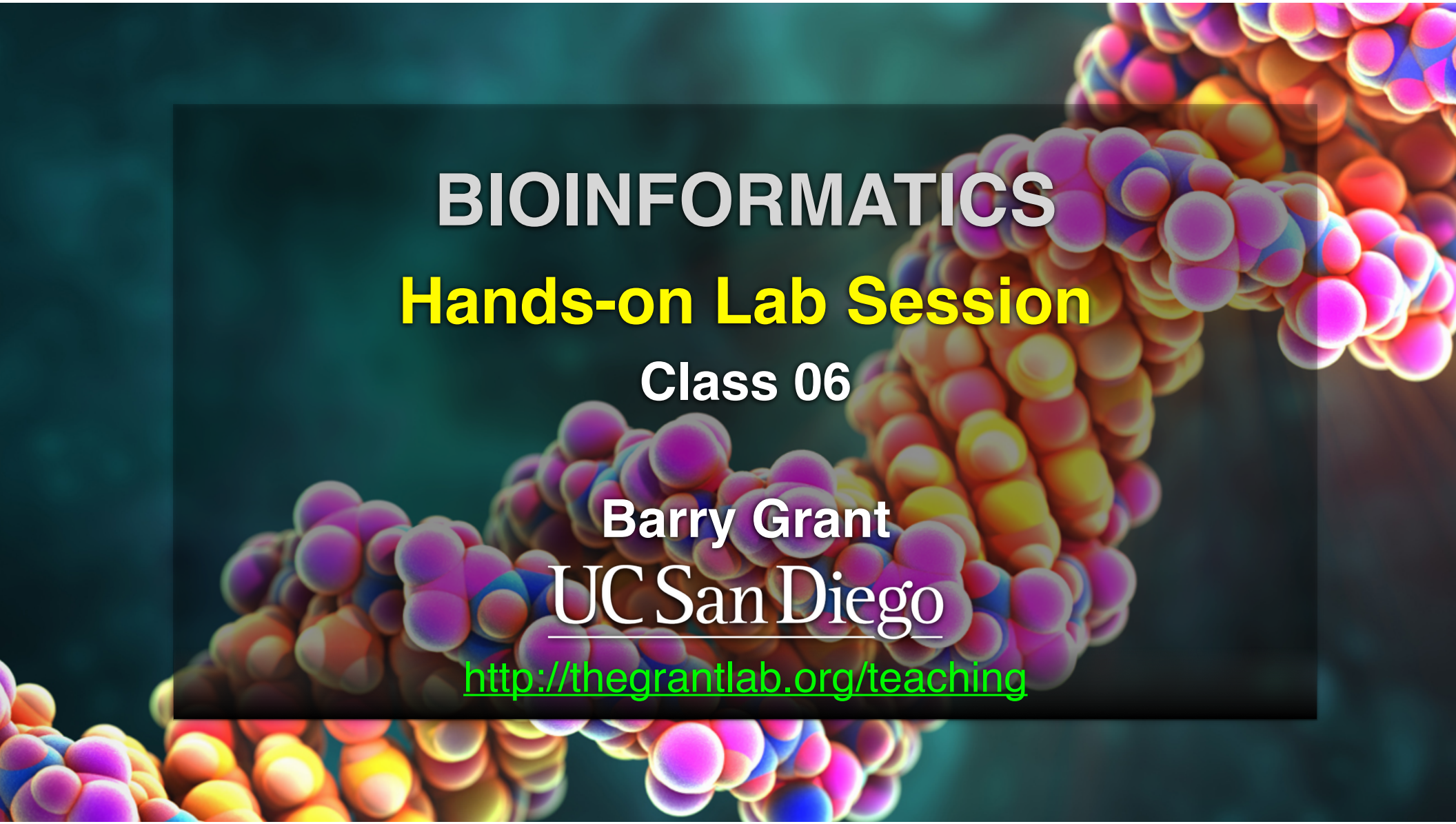

[Image credit: Spotify development team]

Build that skateboard before you build the car.

A limited but functional thing is very useful and keeps the spirits high.

# Suggested steps for writing your functions

1. Start with a simple problem and get a working snippet of code
2. Rewrite to use temporary variables (e.g. x, y, df, m etc.)
3. Rewrite for clarity and to reduce calculation duplication
4. Turn into an initial function with clear useful names
5. Test on small well defined input and (subsets of) real input
6. Report on potential problem by failing early and loudly!
7. Refine and polish


BIOINFORMATICS
Hands-on Lab Session
Class 06
Barry Grant
UC San Diego
http://thegrantlab.org/teaching