Introduction to coding

with R and GitHub

Becky Heath

IPB University

30th August 2024

Selamat Pagi

Postdoctoral research associate working in the Museum of Zoology at the University of Cambridge in collaboration with IPB university

I work on the RERTA project with industrial and smallholder palm oil plantations to develop better sustainability strategies along waterways

@Becky_Heath

rh862@cam.ac.uk

Overview

- Introduction to coding and R
- Introduction to GitHub
- Plan for the rest of the session
- Where to find help

Introduction to coding and R

- Introduction to coding and R
- Introduction to GitHub
- Plan for the rest of the session
- Where to find help

What do we mean by coding?

Writing code is essentially writing a list of instructions for a computer to follow.

Computers operate using 1s and 0s so programming languages are used as a bridge between the user and the system.

Algorithms/scripts are just chunks of code that have been written to tell a computer how to accomplish certain tasks.

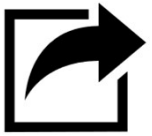
Uses for coding in data science?

Clean, extract, or organise data

Automate tasks and do things more quickly

Create models and perform statistical analysis

Create reproducable analysis pipelines



Create professional data visualisations

Machine learning, big data, predictive modelling

What about R?

R is the most commonly used programming language for data analysis in ecology and biosciences.

R is specifically designed for statistical analysis, so it's particularly powerful with **data visualisation, data manipulation, modelling and visualisation**

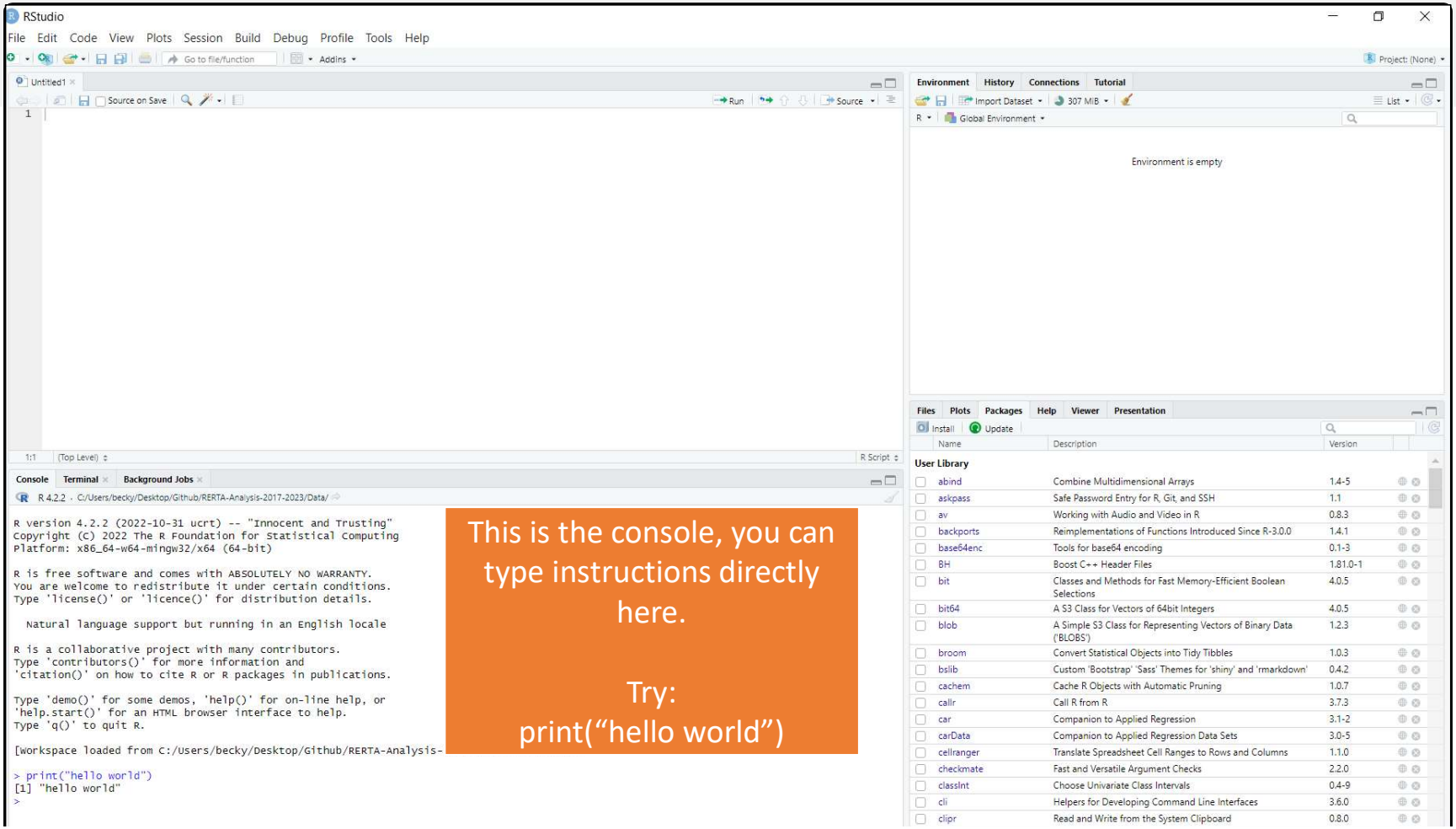
It's completely open source (free!)

R vs. R studio?

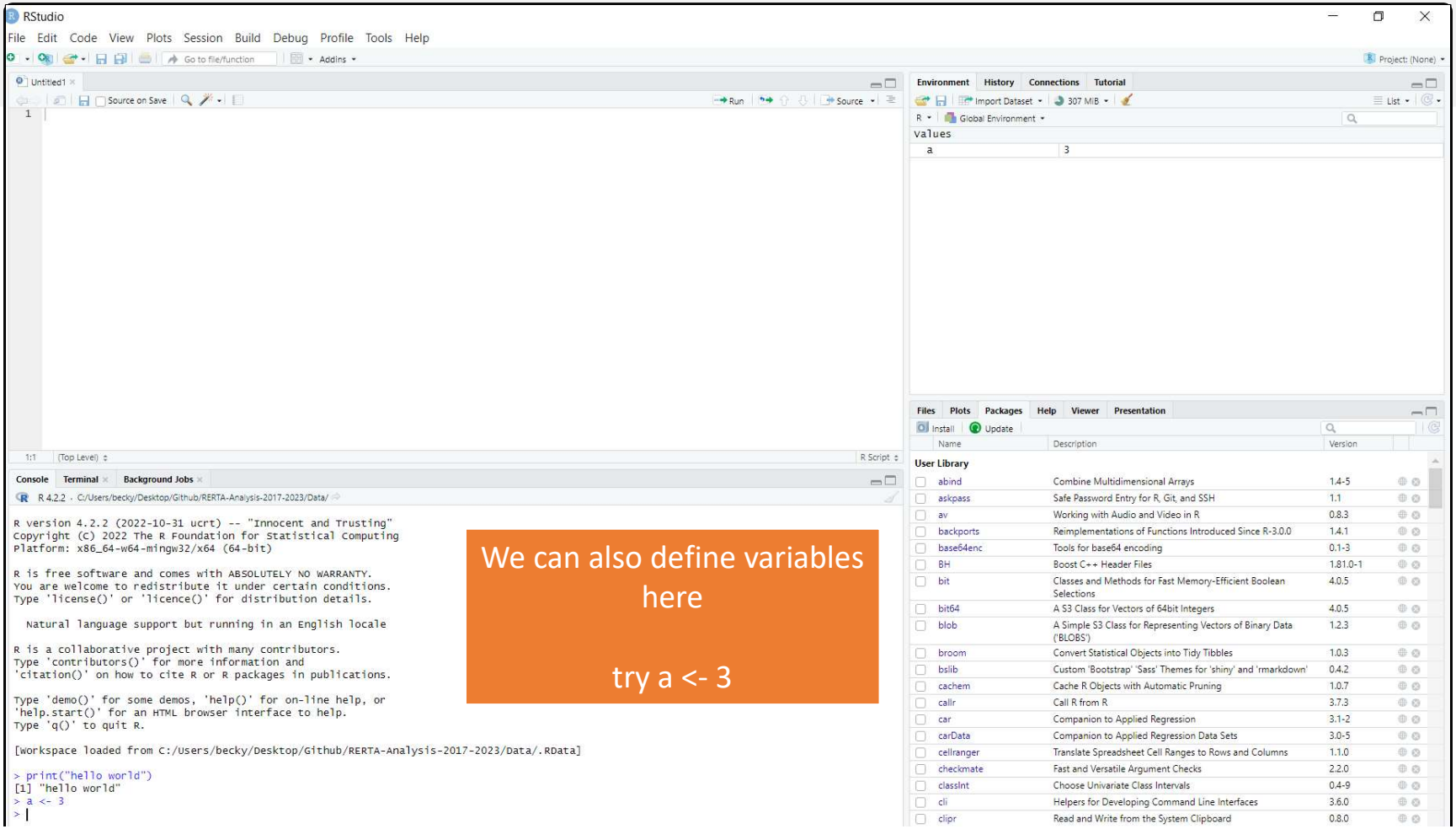
Language

A software platform for writing and using R scripts

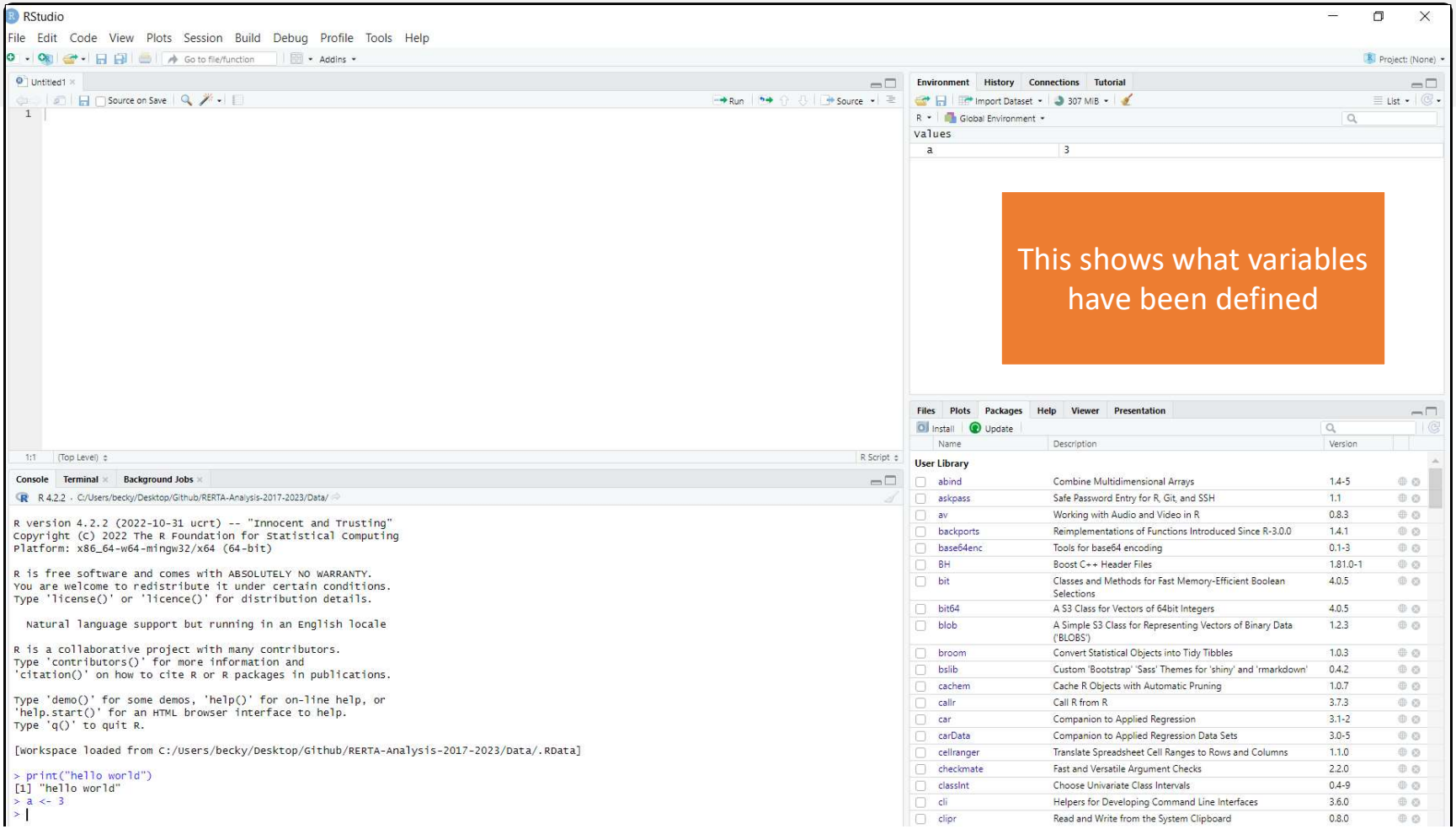
The R studio interface



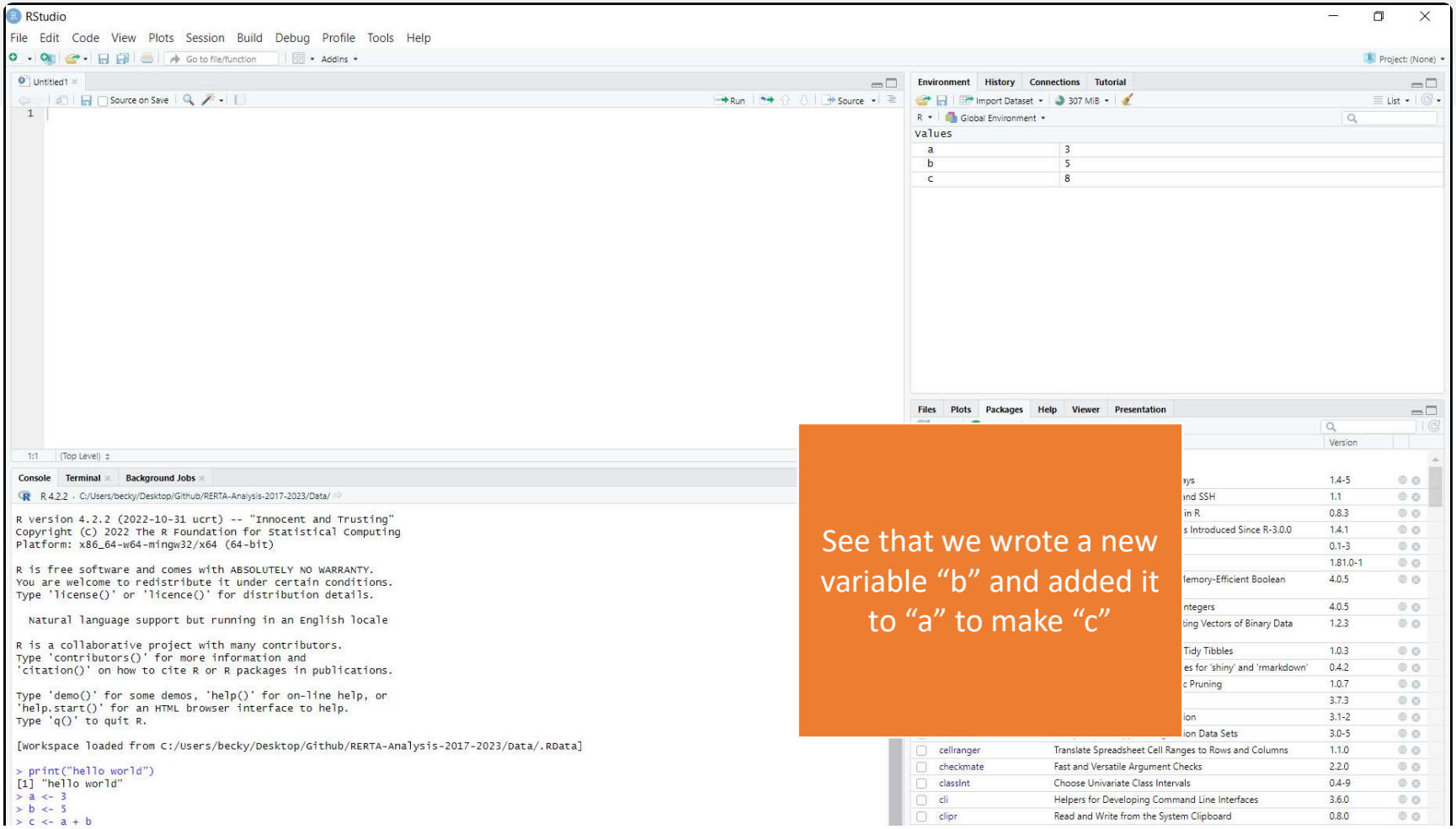
The R studio interface



The R studio interface

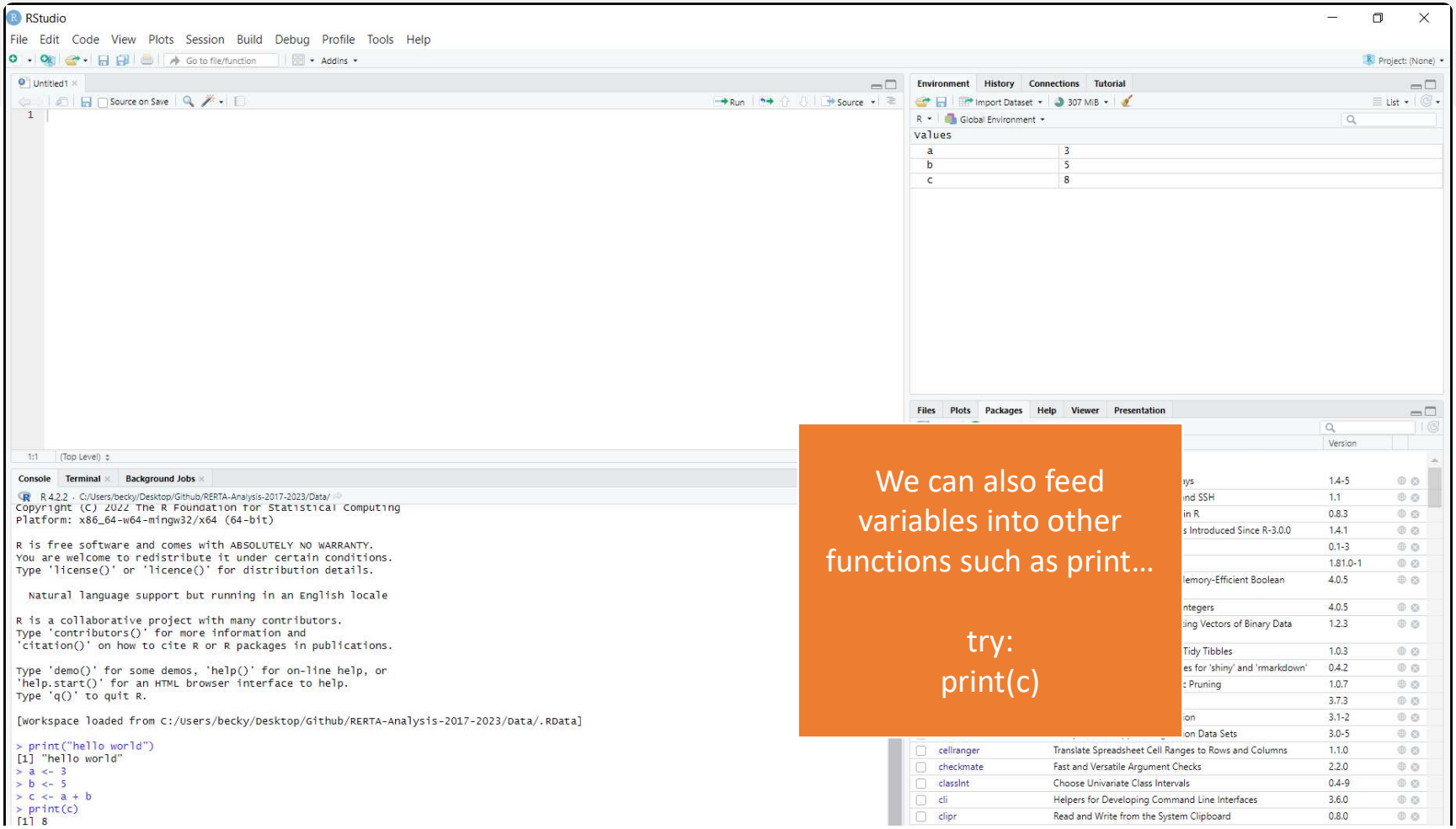


The R studio interface

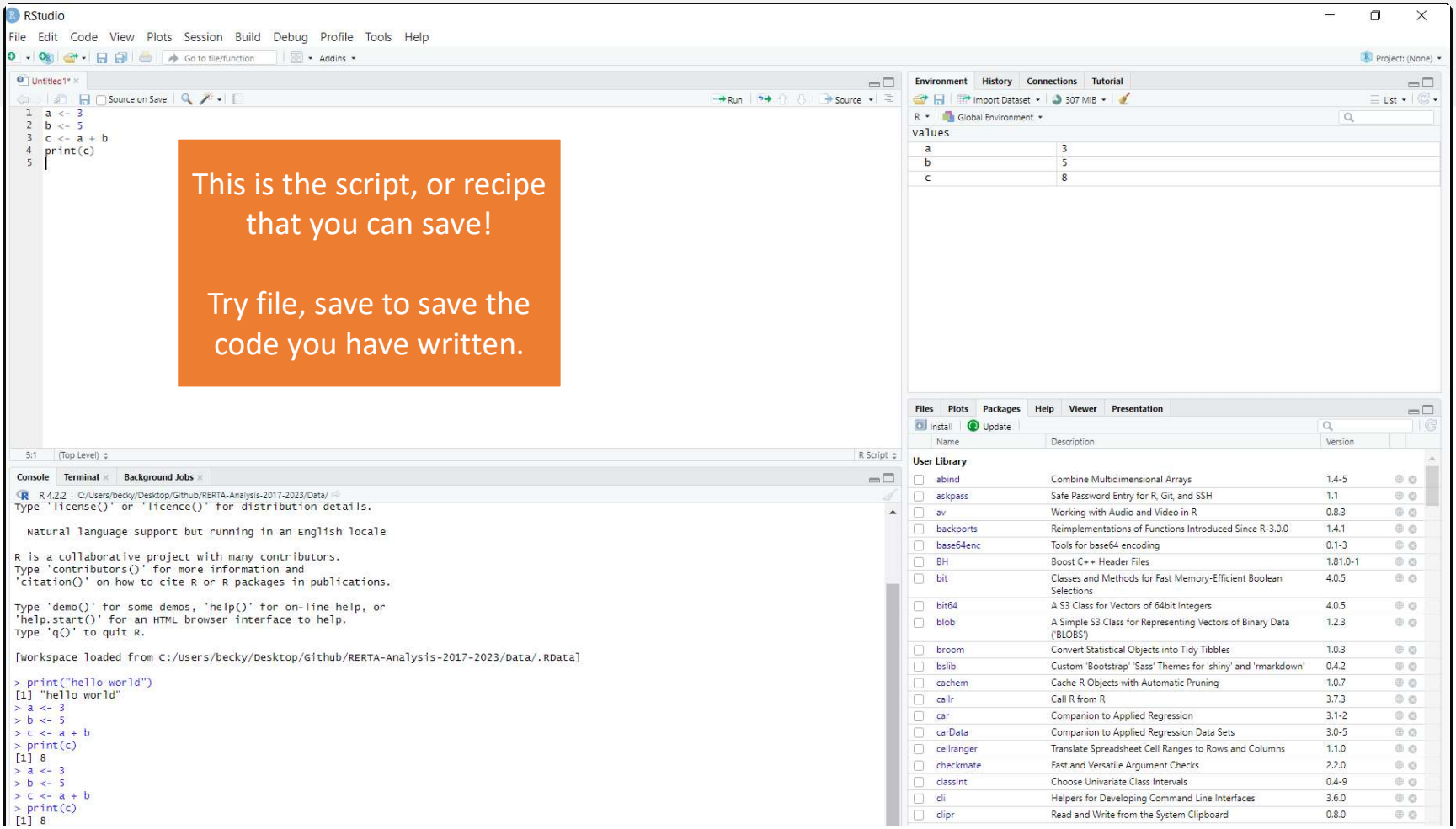


See that we wrote a new variable “b” and added it to “a” to make “c”

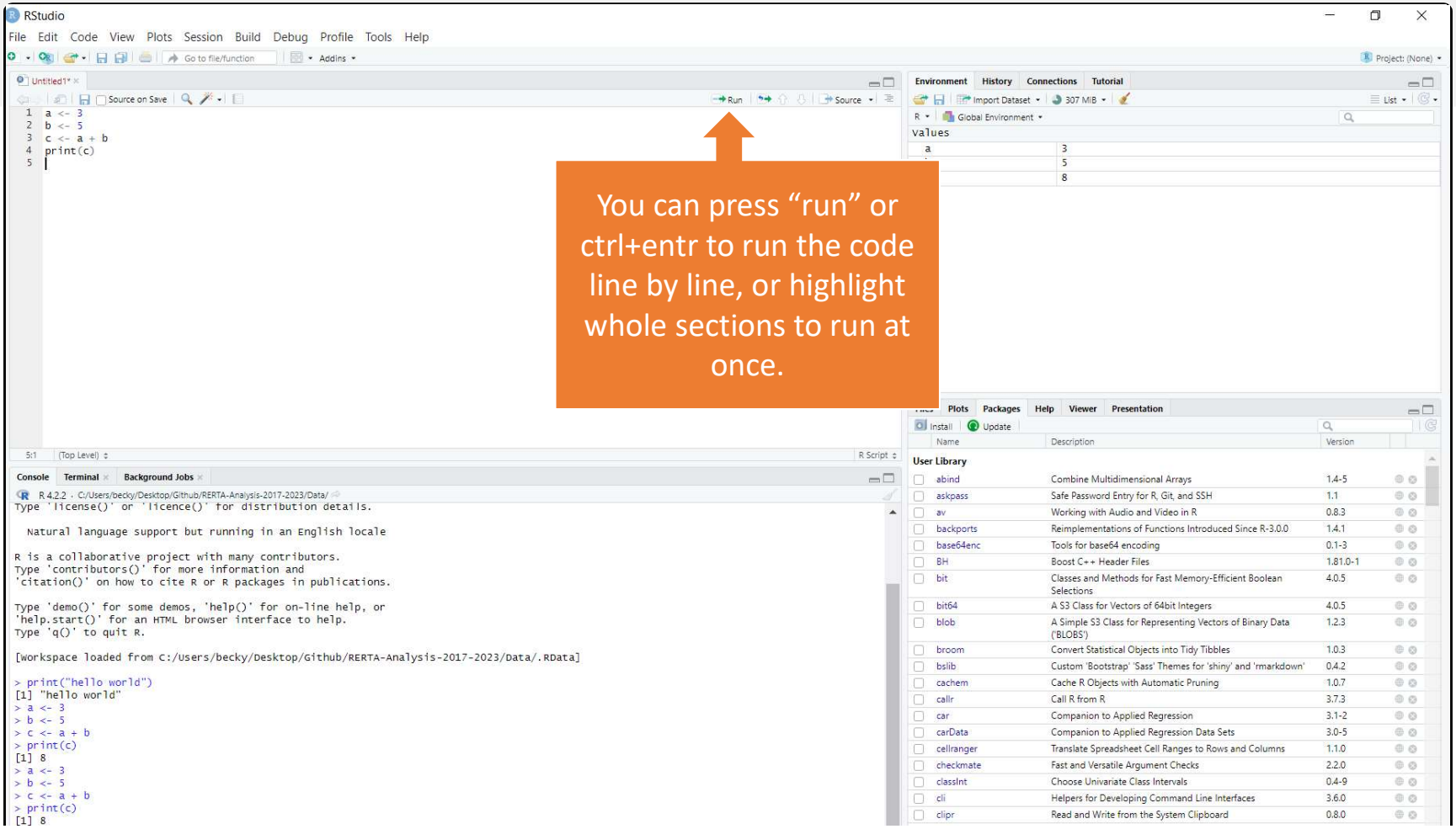
The R studio interface



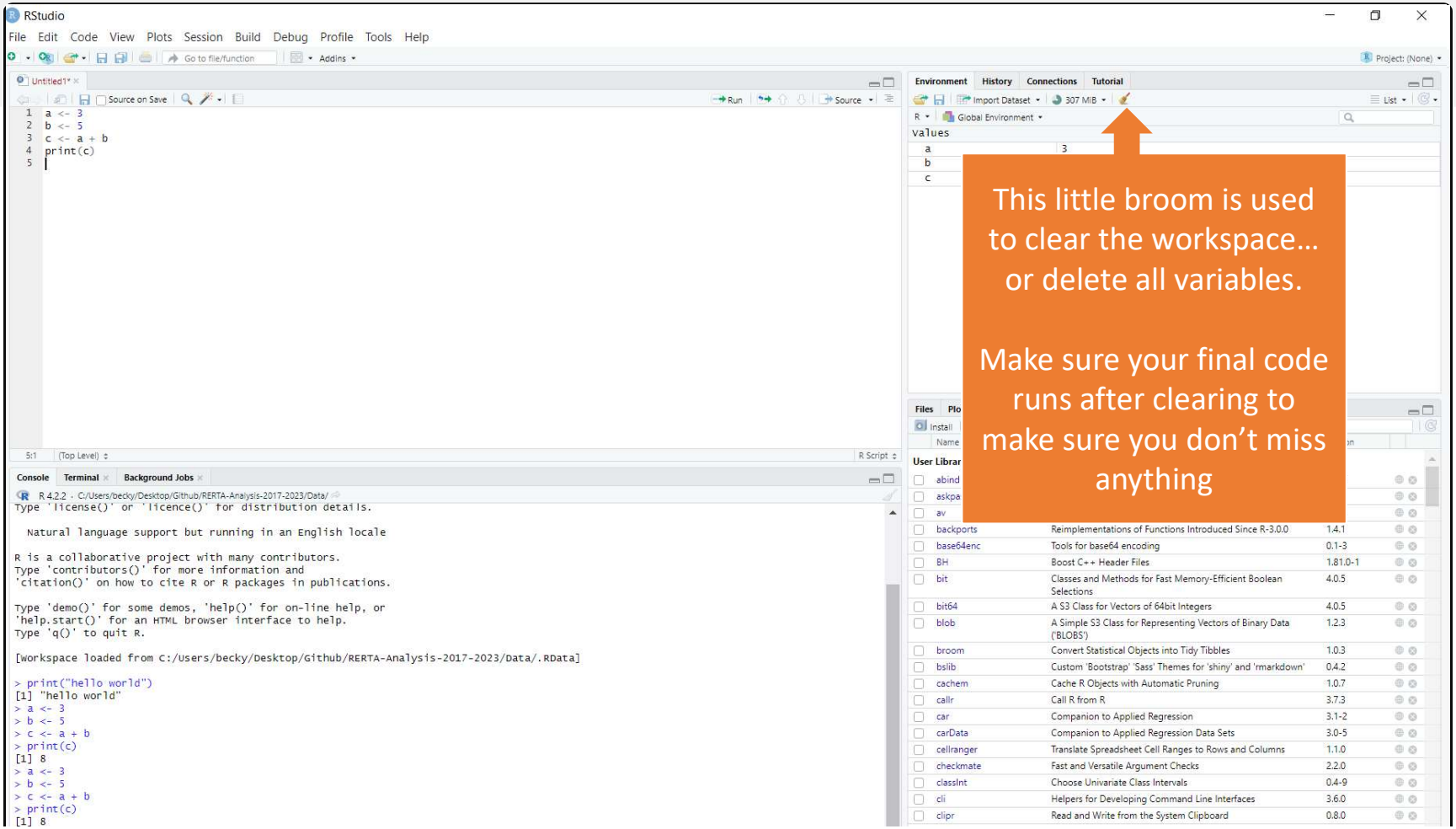
The R studio interface



The R studio interface



The R studio interface



Data Types
using "`typeof()`"

Numeric

```
var1 <- 42
var2 <- 9.81

print(typeof(var1))
print(typeof(var2))
```

Integer

```
var1 <- 5L
var2 <- 15L

print(typeof(var1))
print(typeof(var2))
```

Data frame

```
var1 <- data.frame(
    ID = 1:3,
    Name = c("Alice", "Bob", "Charlie"),
    Age = c(25, 30, 35))

print(typeof(var1))
```

Logical

```
var1 <- TRUE
var2 <- FALSE

print(typeof(var1))
print(typeof(var2))
```

Character

```
var1 <- “Hello, world!”
var2 <- “405”

print(typeof(var1))
print(typeof(var2))
```

Vector

```
var1 <- c(1,2,3,4,5)
var2 <- c(“1”, “hi”, “IPB”)

print(typeof(var1))
print(typeof(var2))
```

Common operators:
What do they mean?

```
a <- 5
b <- 3
result <- (a + b) * b
print(result)
```

```
p <- TRUE
q <- FALSE
print(p & q)
```

```
vec1 <- c(1, 2, 3)
vec2 <- c(4, 5, 6)
result <- vec1 + vec2
print(result)
```

```
a <- 7
b <- 4
print(a > b)
```

```
x <- 10
x <- x + 5
print(x)
```

Common operators:
Answers

```
a <- 5
b <- 3
result <- (a + b) * b
print(result)
```
24

```
p <- TRUE
q <- FALSE
print(p & q)
```
False

```
vec1 <- c(1, 2, 3)
vec2 <- c(4, 5, 6)
result <- vec1 + vec2
print(result)
```
c(5, 7, 9)

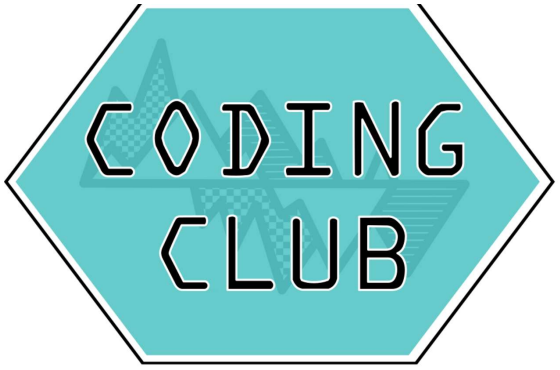
```
a <- 7
b <- 4
print(a > b)
```
TRUE

```
x <- 10
x <- x + 5
print(x)
```
15

Now try:
Loading in a dataset

Go to:

https://ourcodingclub.github.io/tutorials/intro-to-r/



Introduction to GitHub

- Introduction to R
- Introduction to GitHub
- Plan for the rest of the session
- Where to find help

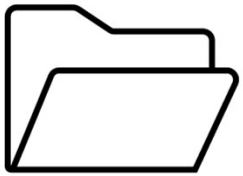
@Becky_Heath rh862@cam.ac.uk

GitHub

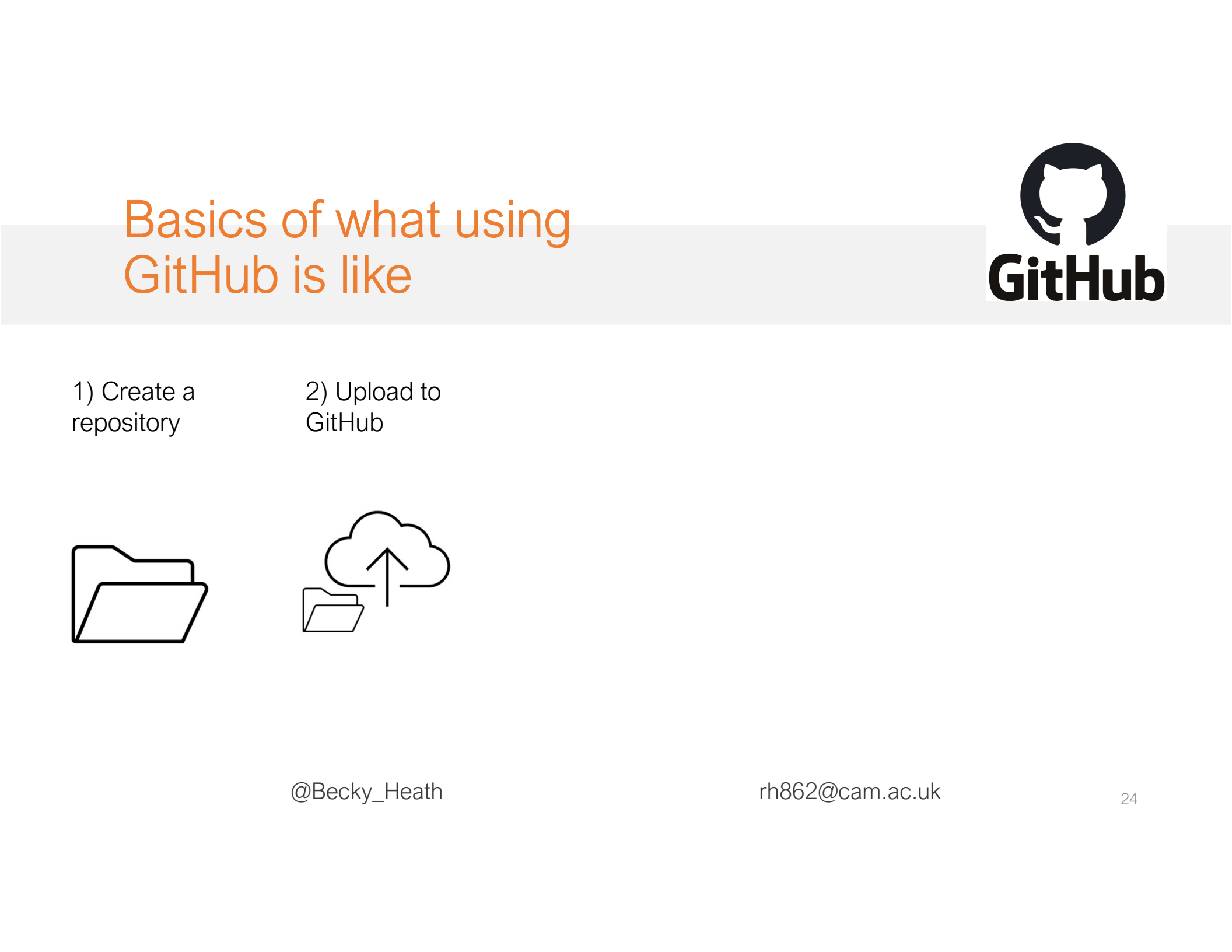
A platform for sharing, storing, and collaborating on code projects.

Basics of what using GitHub is like

1) Create a repository



Basics of what using GitHub is like

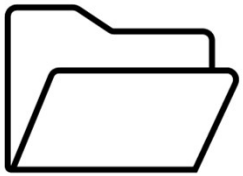


1) Create a repository

2) Upload to GitHub

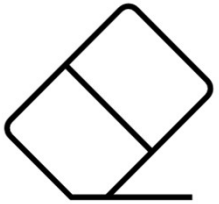
Basics of what using GitHub is like

1) Create a repository



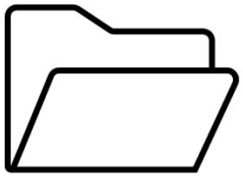
2) Upload to GitHub

3) Make changes locally



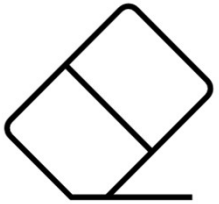
Basics of what using GitHub is like

1) Create a repository

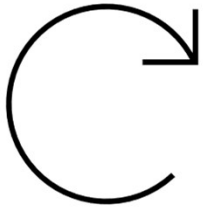


2) Upload to GitHub

3) Make changes locally



4) Send updates to GitHub

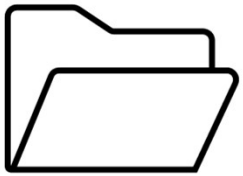


@Becky_Heath

rh862@cam.ac.uk

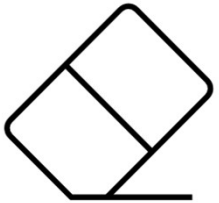
Basics of what using GitHub is like

1) Create a repository

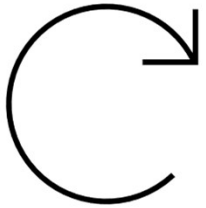


2) Upload to GitHub

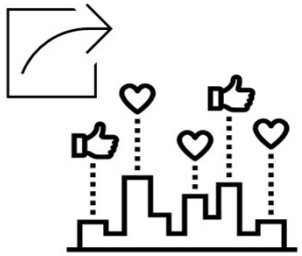
3) Make changes locally



4) Send updates to GitHub



5) Share/ collaborate



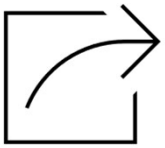
@Becky_Heath

rh862@cam.ac.uk

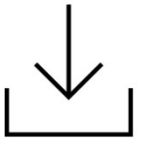
Why bother using GitHub?

Keep a record of your work, revisit old versions.

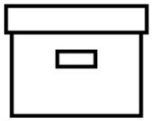
Share code and collaborate more easily.



Publish code in journals.

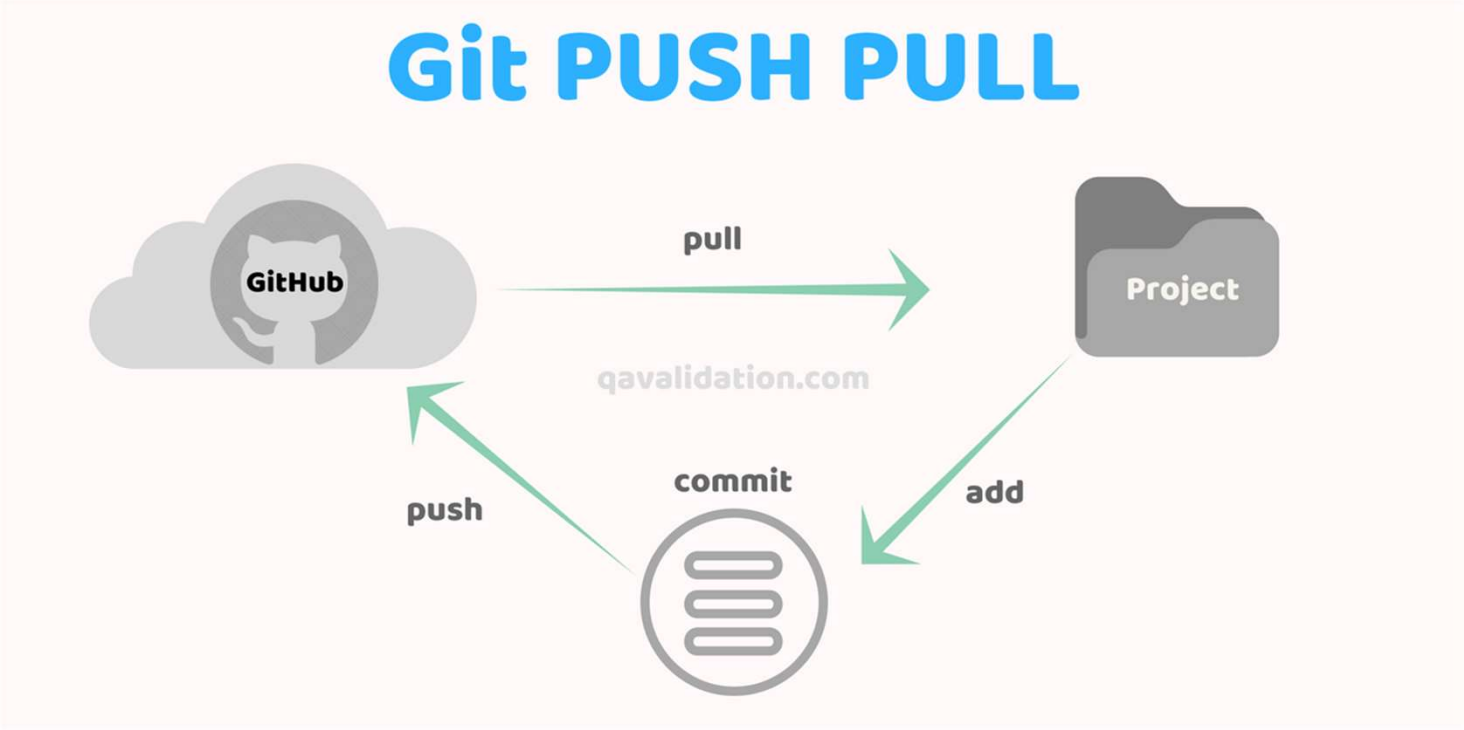


Use other's work for your own projects

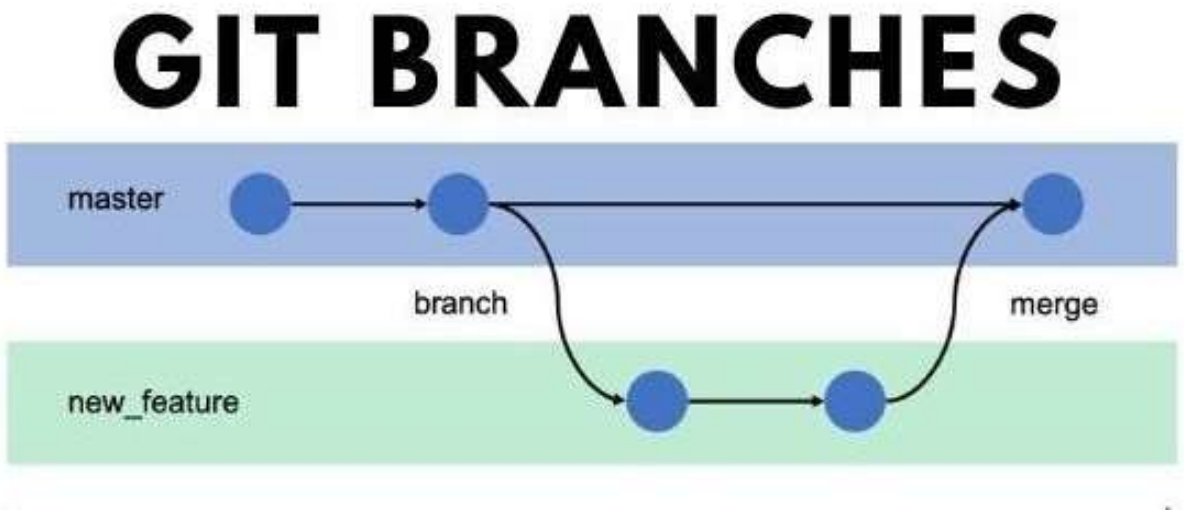


Online backup/ archive

GitHub Core Concepts



GitHub Core Concepts



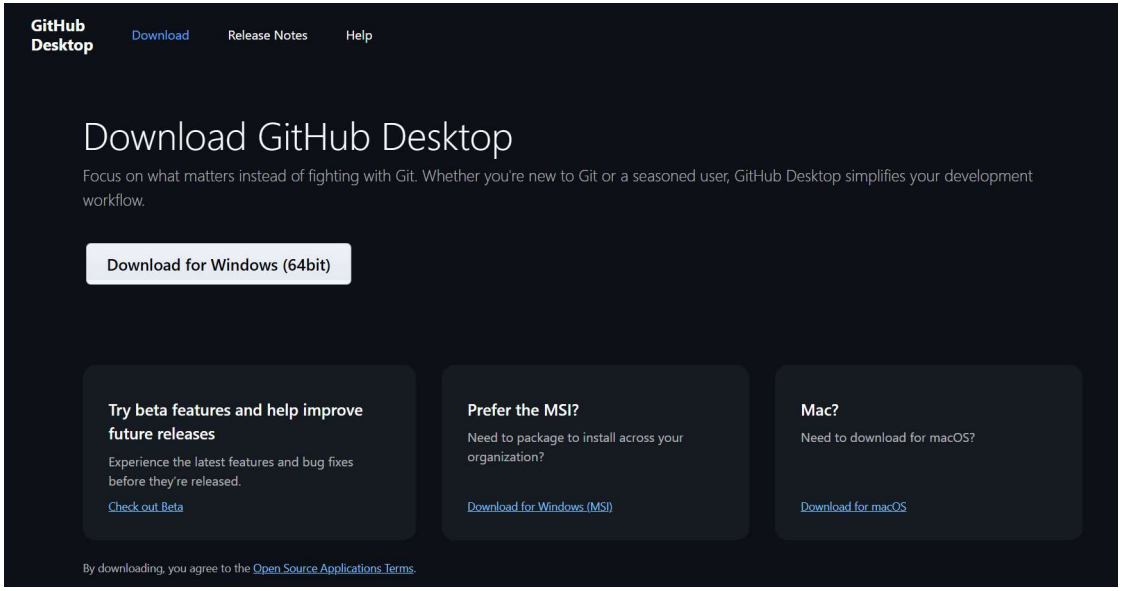
GitHub Desktop

GitHub Desktop is a friendly interface for working with GitHub repos

Here you can create a new repo locally, clone repos, and push to your GitHub account online

Getting started with GitHub

1) Download GitHub Desktop (this will also download Git)
https://docs.github.com/en/desktop/installing-and-authenticating-to-github-desktop/setting-up-github-desktop#



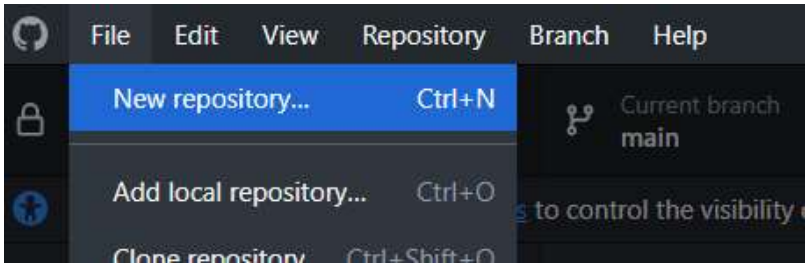
rh862@cam.ac.uk

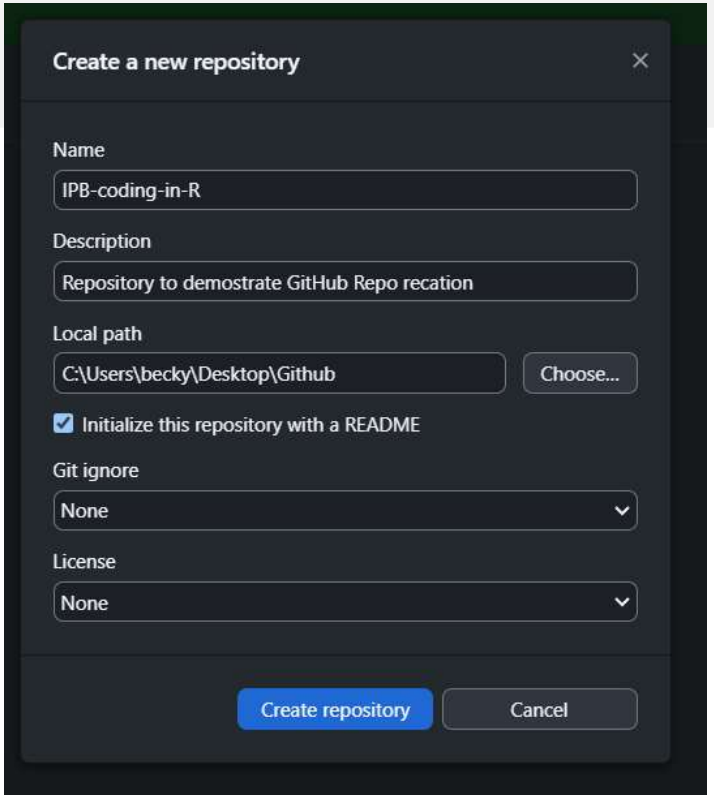
Getting started with GitHub

2) Authenticate by signing in (you'll need to create an account if you don't have one https://docs.github.com/en/desktop/installing-and-authenticating-to-github-desktop/authenticating-to-github-in-github-desktop

Getting started with GitHub

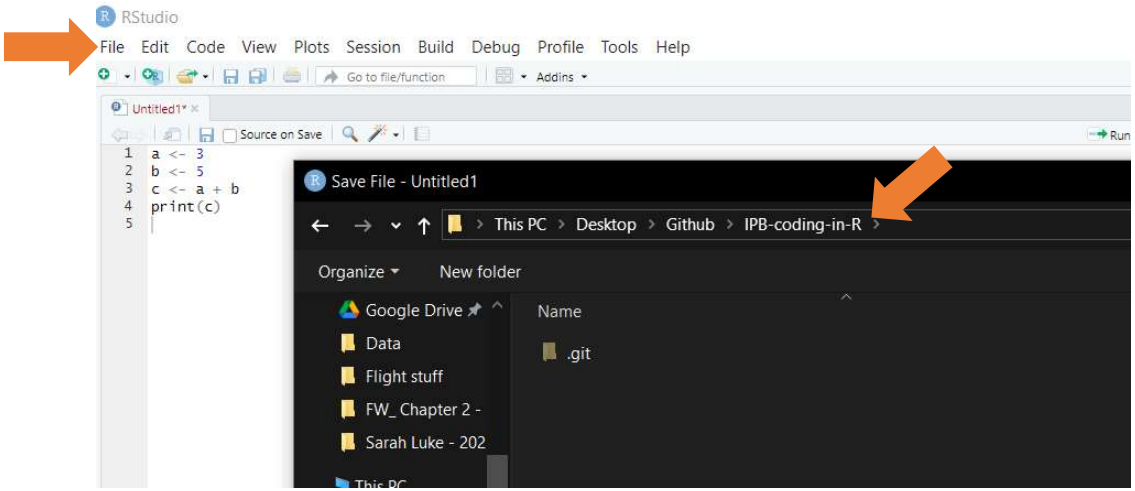
3) Create a new repository





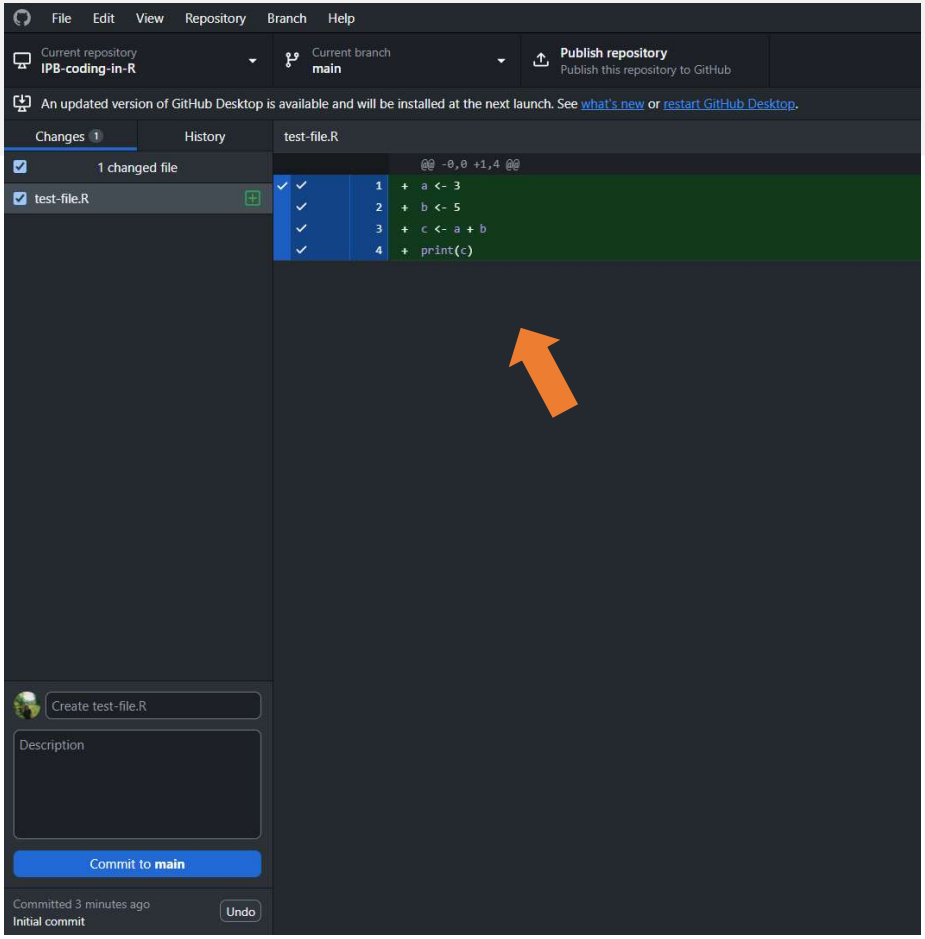
Getting started with GitHub

4) Save R file to new repo



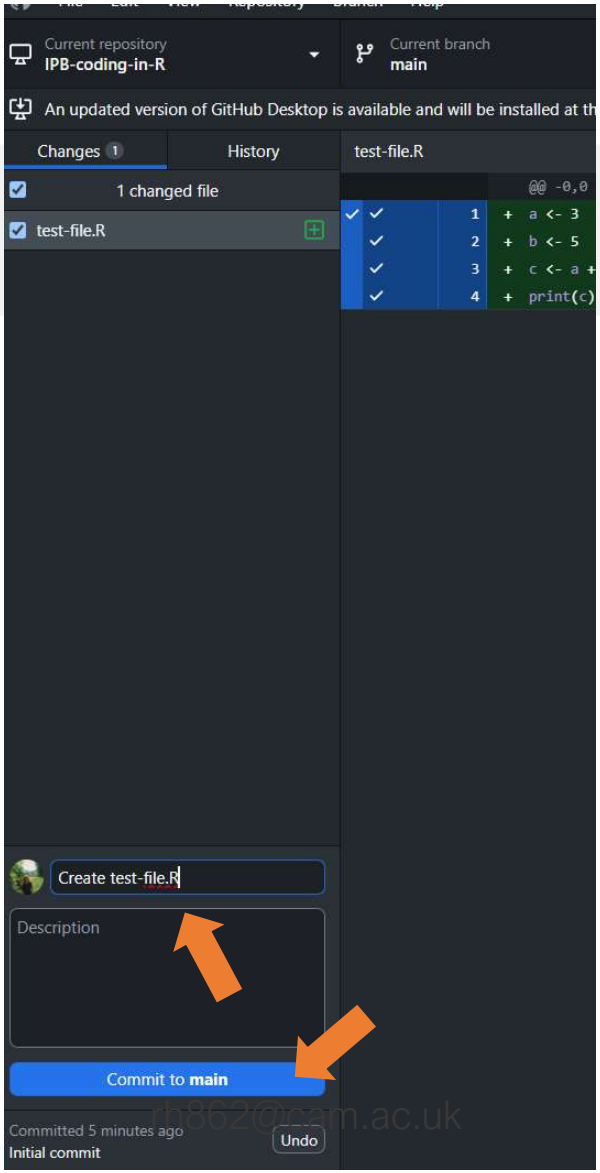
Getting started with GitHub

5) Check that the new file/ edits have been picked up by github desktop

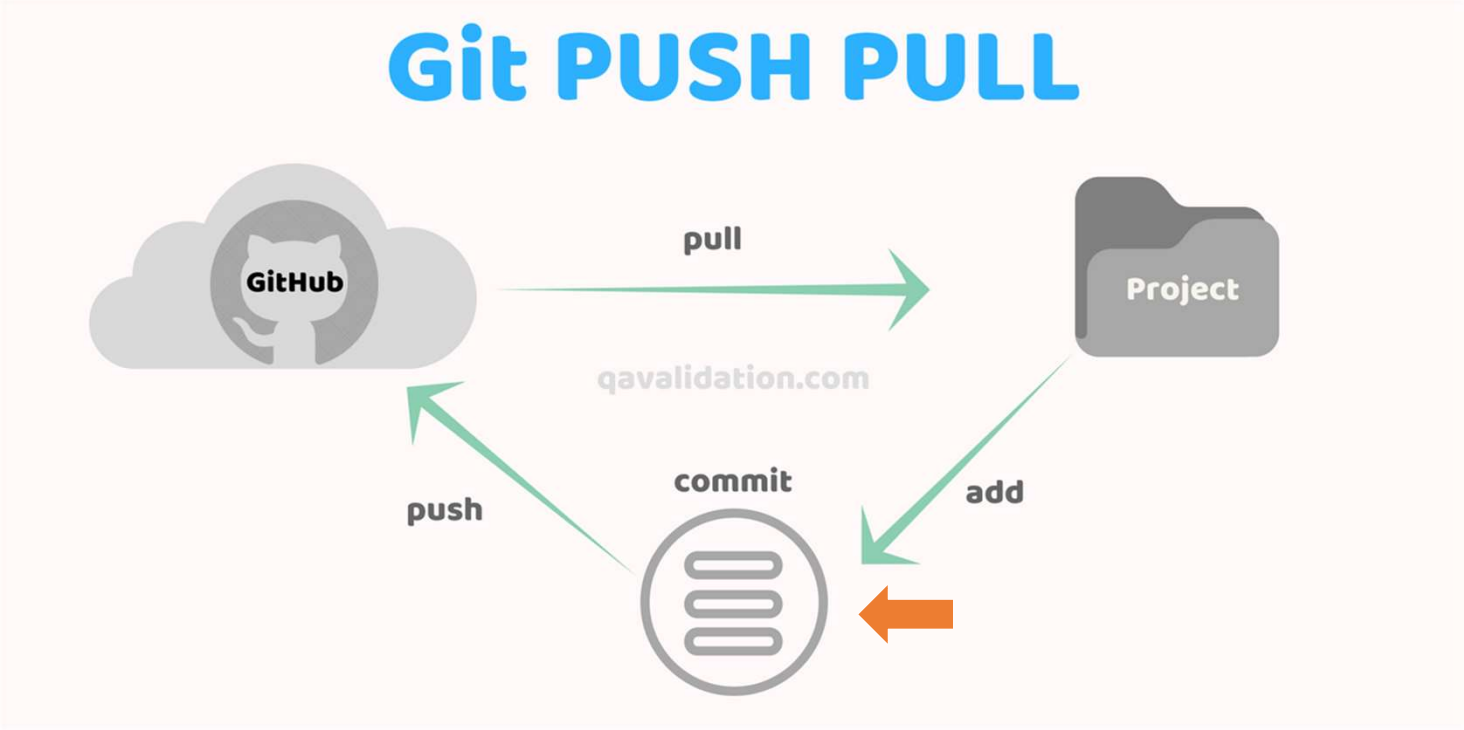


Getting started with GitHub

6) Write a description for the commit (try to be informative!)



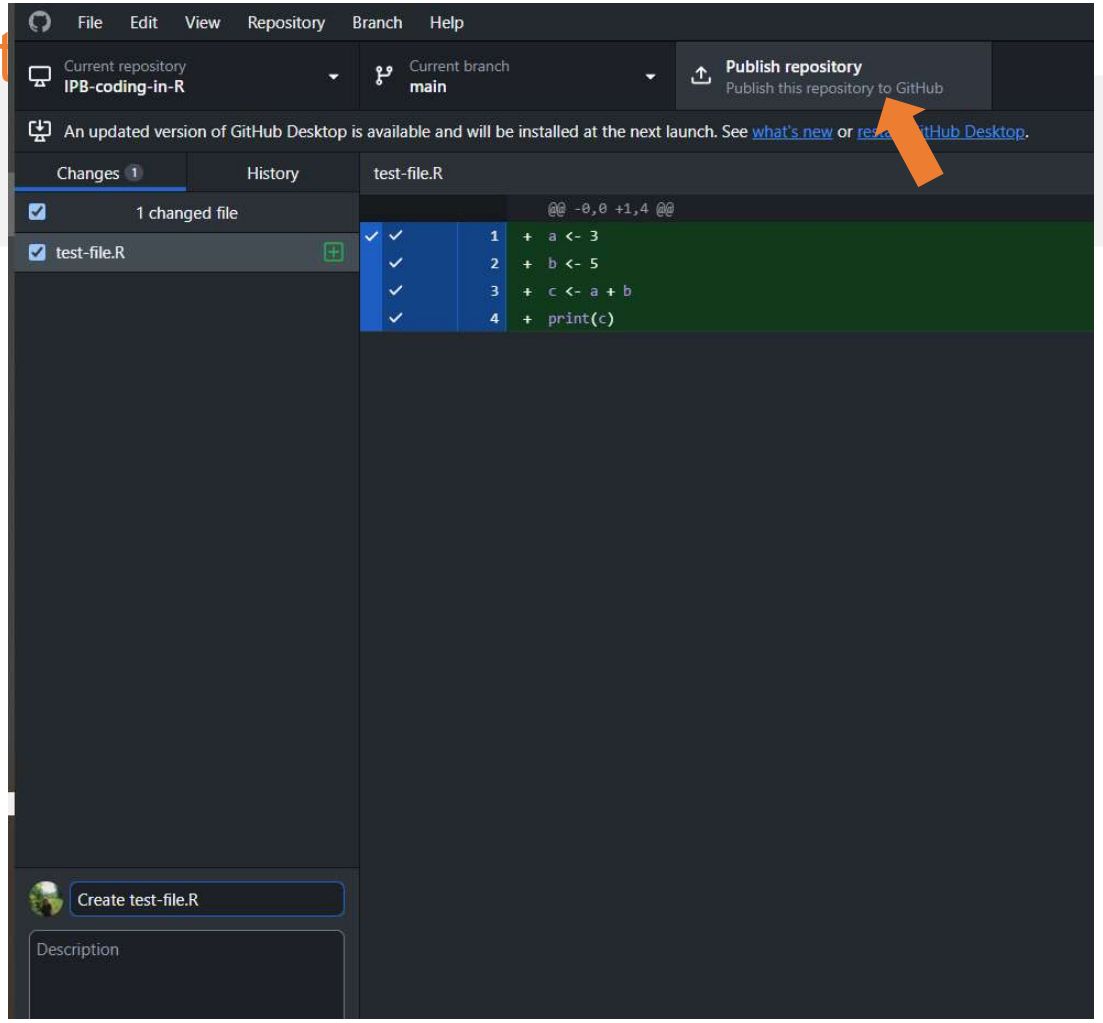
GitHub Core Concepts



Getting started with Gi

7) Push the commit to GitHub ☺

NB: it says publish the first time but it will say push after

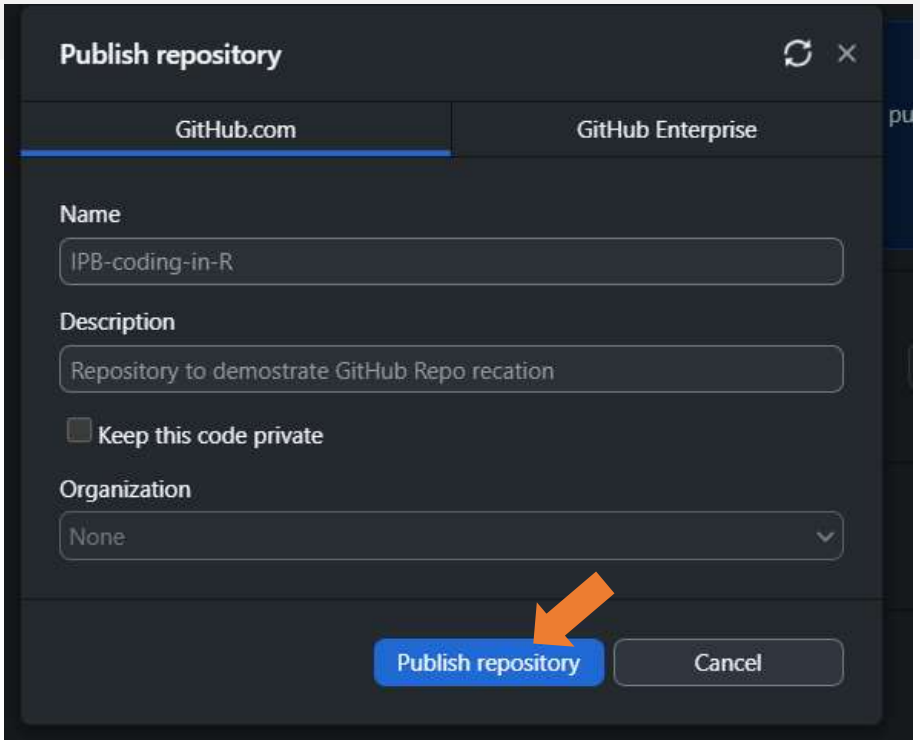


Getting started with GitHub

6) only the first time:

check details are correct and decide if you want the repo to be public or not

…. then publish

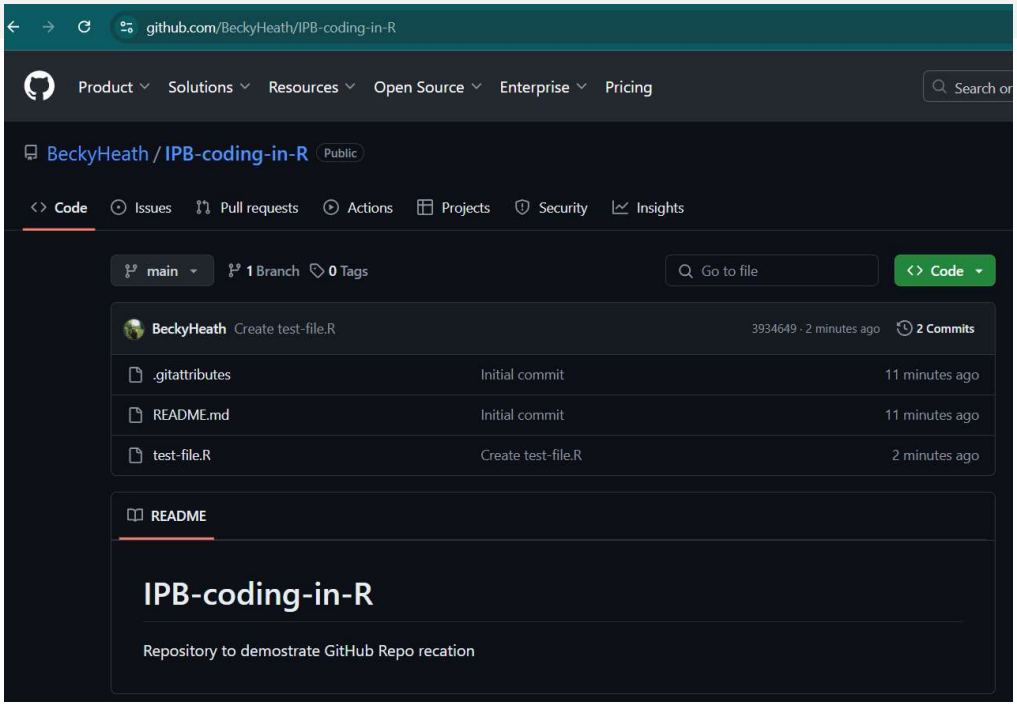


Getting started with GitHub

6) Check the repo is online!!

To check, go to

github.com/[USERNAME]/[REPO-NAME]



This lesson's repo

The repo created just now containing slides and links to tutorials:

https://github.com/BeckyHeath/IPB-coding-in-R

Rest of the session

- Introduction to coding and R
- Introduction to GitHub
- Plan for the rest of the session
- Where to find help

Plan for the rest of the session

Work through tutorials online at your own pace

Try to find solutions online (next section).

Ask for help if you can't find a solution

@Becky_Heath

rh862@cam.ac.uk

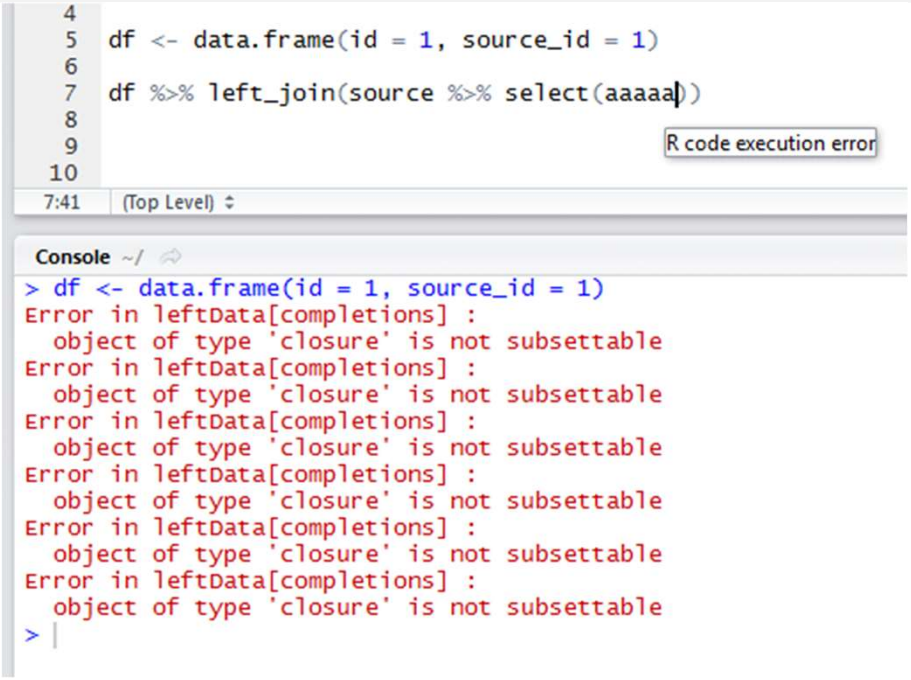
Where to find help

- Introduction to coding and R
- Introduction to GitHub
- Plan for the rest of the session
- Where to find help

Using error messages?

The fastest and easiest way to debug code in R is to type the error message into google and test out solutions

Best to use public forums such as “cross check”, “stack overflow” and “stack exchange”.

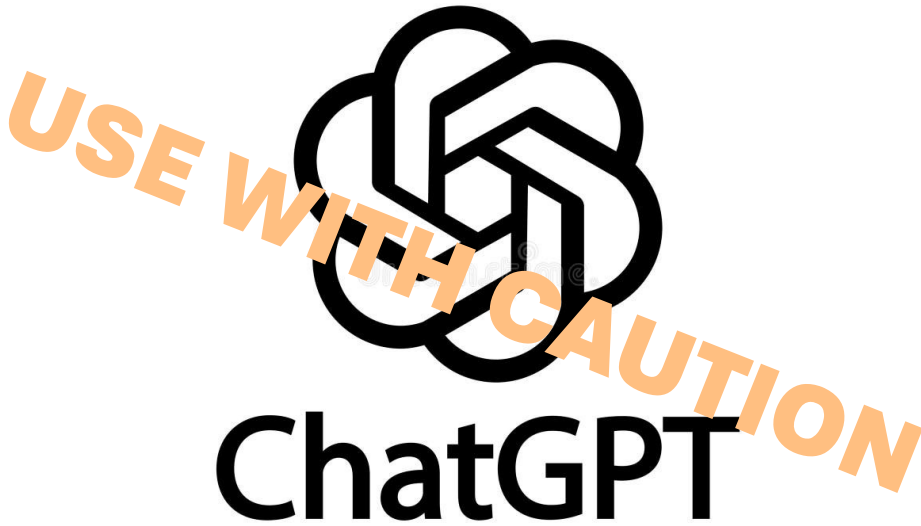


Using AI for debugging?

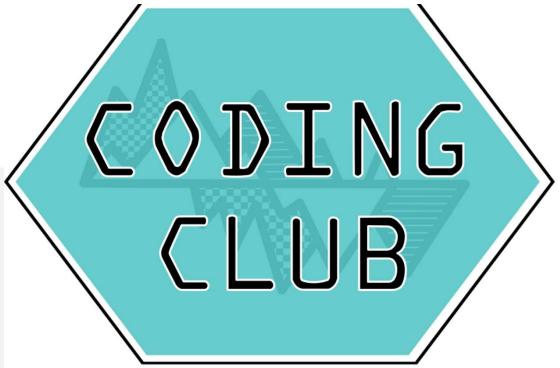
Chat GPT is an algorithm that is really good at picking out patterns in human language.

It is very good at sounding correct, and is very often good at finding simple solutions, but not complex issues.

Chat GPT gets stuff wrong a *lot*.



Selected Tutorials



UNIVERSITY OF CAMBRIDGE

Code Club (University of Edinburgh)

1- R Basics : All

2 – Data Manipulation: Basic Data Manipulation

3 – Data Manipulation: Efficient Data Manipulation

4– Data Visualisation: Basic Part1 and Part2

5 – Modelling: From distributions to Linear Models

6 – Modelling: Introduction to model design