



Clase 6

Proyectos en R

Miriam Lerma

Marzo 2021

Intro

- Trabajar por proyectos.
- Ordenar mis carpetas y crear mi proyecto.
- Crear un proyecto.
- Guardarlo y compartirlo.

Ustedes

- Conocimientos básicos de R (saben abrirlo, cargar paquetes y datos, saben hacer algunas operaciones y gráficos, saben exportar csvs y gráficos).
- Quieren tener ordenado su trabajo para poder re-hacer sus análisis en el futuro.
- Quieren trabajar con alguien mas y mandarle materiales.

Créditos

Si quieren profundizar en el tema:

-  Proyectos en R4DS
-  Nombrar cosas por Stephanie Orellana
-  Naming things by Jenny Bryan
- Presentaciones de reproducibilidad
 - RLadiesJohannesburg
- Imágenes adicionales
 -  Unsplash
 -  Portada por Hannah Busing



Empecemos!

1. Teoria

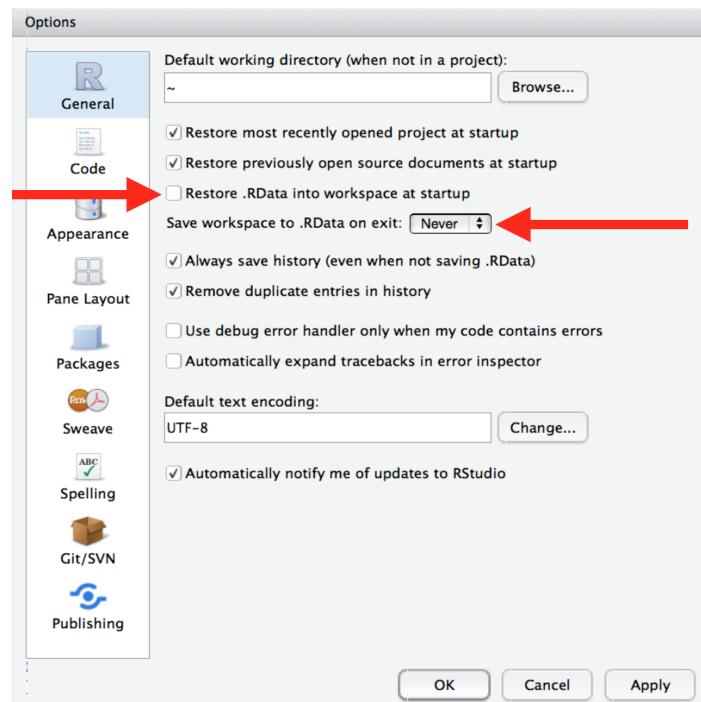
Llegara el día en que cerraran R y harán otra cosa y tendrán que volver al análisis que estaban haciendo.

También llegara el día en que trabajen en múltiples proyectos a la vez y quieran mantenerlos separados.

1.1. Donde vive tu análisis?

Tus scripts te permiten recrear el environment (ambiente) del trabajo donde estabas trabajando.

Por esto R4DS recomienda, que **no guardes tu espacio de trabajo**.



1.2. Rutas

Su directorio de trabajo (working directory en inglés) es donde R busca los archivos que le pides que lea y donde colocará todos los archivos que le pidas que guarde.

RStudio muestra tu directorio de trabajo actual en la parte superior de la consola o puedes escribir:

```
getwd()
```

Existen dos estilos básicos de rutas:

- Mac y Linux usan barras (slash).
- Windows usa barras invertidas (backslash) o dos barras.

1.2. Rutas absolutas

Las **rutas absolutas** te llevan a la misma ubicación sin importar tu directorio de trabajo.

Tu directorio de trabajo se puede definir manualmente, pero usar rutas absolutas dificulta compartir tus scripts.

⚠ Nadie más va a tener exactamente la misma configuración de directorios que tú.

Por ejemplo, esta es la ruta a un archivo que solo aplica para mi computadora.

```
Pingus<-read_csv("C://Users//Lerma//Documents//4Cursos//1CIAD//01Clase :
```

1.2. Rutas relativas

Las **rutas relativas**, siempre que hagas referencia a un archivo con una ruta relativa, RStudio lo buscará en este directorio.

También librerías como `here`, se adaptan a la ubicación de tus proyectos.

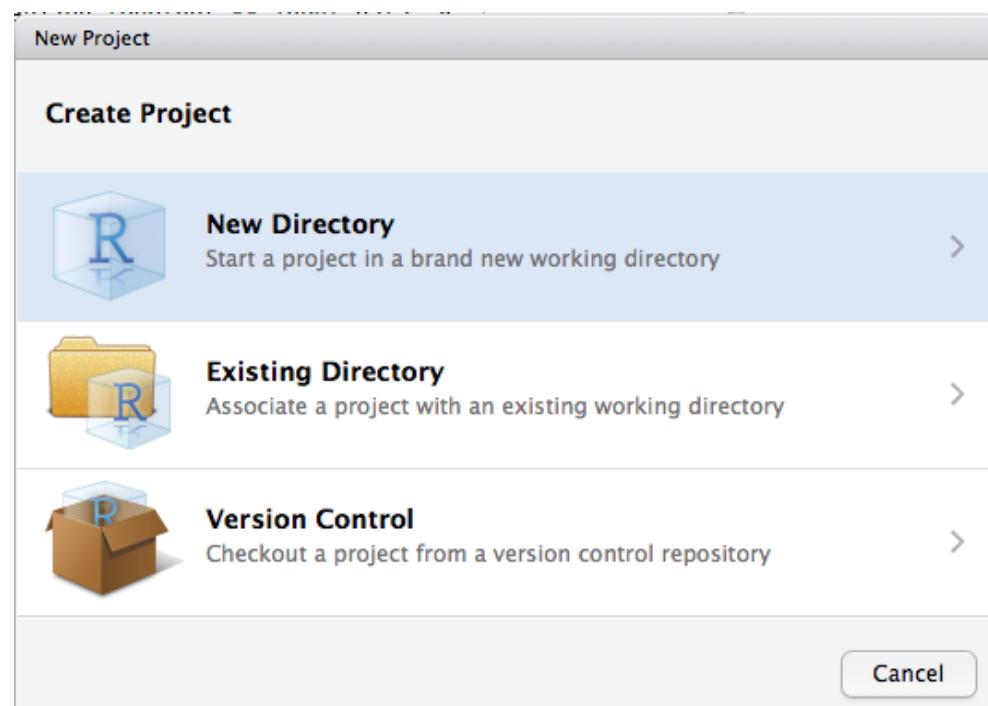
```
here()
```

```
Pingus<-read_csv(paste0(here(),"//Pingus_1.csv"))
```

1.3. Proyectos en RStudio

Lo ideal es tener todos los archivos asociados a un proyecto en un mismo lugar.

Para crear un proyecto en R File > New Project, y después elegir la adecuada. Hoy usaremos New Directory.



1.4. Proyectos

Es **bueno** que se limpie el environment, te obliga a reportar todo lo que hiciste...
... y que puedas replicarlo en el futuro.



Cuidado

Cuando creas tu nuevo proyecto, no guarda los objetos que tengas en el environment.

Pero lo que si estas siguiendo estos pasos a la par y tienes objetos importantes en tu environment, guardalos antes de crear tu proyecto

Para guardar tus objetos puedes usar `write_csv`, `saveRSD` o lo que prefieras.

1.4. Ventajas

- Un proyecto te permite tener todo un 'proyecto' como su tesis, artículos, etc, dentro de una sola carpeta.
- La carpeta se puede cambiar de lugar sin perder todos las direcciones (usando direcciones relativas).
- También te permite definir que vas a usar y enfocarte en eso, no tener todo en diferentes lados.
- Te puede ayudar a colaborar con otros.

En resumen, los proyectos en RStudio te ofrecen la oportunidad de mantener un flujo de trabajo consistente que te será de mucha utilidad en el futuro.



1.4. Recomendaciones

- Crea un proyecto RStudio para cada proyecto de análisis de datos.
- Mantén los archivos ahí.
- Conserva los scripts ahí también; edítalos, ejecútalos por partes o en su totalidad.
- Guarda todos los resultados de tu trabajo (gráficos y sets de datos limpios) ahí.
- Siempre usa rutas relativas, nunca rutas absolutas.

Todo lo que necesites siempre estará concentrado en una sola ubicación y claramente separado de los otros proyectos en los que estés trabajando.

Fuente R4DS

1.5. Nombrar las cosas.

Cuales son nombres de proyecto o de objetos significativos?

Malos nombres:

- Introduccion final diciembre.pdf
- modelo.xlsx
- figural.png
- Introduccion final diciembre-2.pdf
- Datos contacto@ gatos.cl.txt
- W.docx
- Introduccion final diciembreFINAL.pdf
- Introduccion final diciembreFINAL-final-revisado.pdf

Fuente: Stephanie Orellana

Buenos nombres

 Leibles para la computadora

 Leibles para los humanos

 Funcionen bien con orden por defecto

1.7. Para la computadora

- Usar palabras clave.
- Sin caracteres especiales.
- Sin signos de puntuación.

Usar " - " para separar nombres con mas de una palabra.

Evita

- Usar espacios en los nombres de los archivos.
- Signos de puntuación.
- Caracteres acentuados.
- Diferentes archivos con nombres similares y con diferente contenido (ej. gato y Gato).

1.7. Para humanos

Fácil de saber que contiene el archivo a partir de su nombre.

- 01_introduccion.docx
- 02_metodologia.docx
- 03_discucion.docx
- 04_tablas.docx

1.7. Orden por defecto

Usar algo numérico al inicio permite ordenarlos por defecto, y lo mejor es anteponer ceros.

Si no anteponemos el cero, 1... se desalinea con 10.

Mis carpetas son:

01Datos
02Analisis
03Resultados
04Discussion
05Referencias

Otras opciones:

01_Datos
01_datos

1.8. Comentar su código

Pongan la mayor cantidad de información que puedan para recordar porque hicieron que cosa. Tener tus datos ordenados y tener un solo archivo ayuda a compartir tu trabajo y te permite replicar el experimento en el futuro.

```
#Este código es para cargar mis datos
Pingus_csv<-read_csv(here("Pingus_1.csv"))
#Para cargar otros datos
#Tengo que nombrar un objeto
#Usar la función read_csv del paquete tidyverse
#Recordar cargar la librería here
#Cambiar el nombre de lo que está entre comillas
```

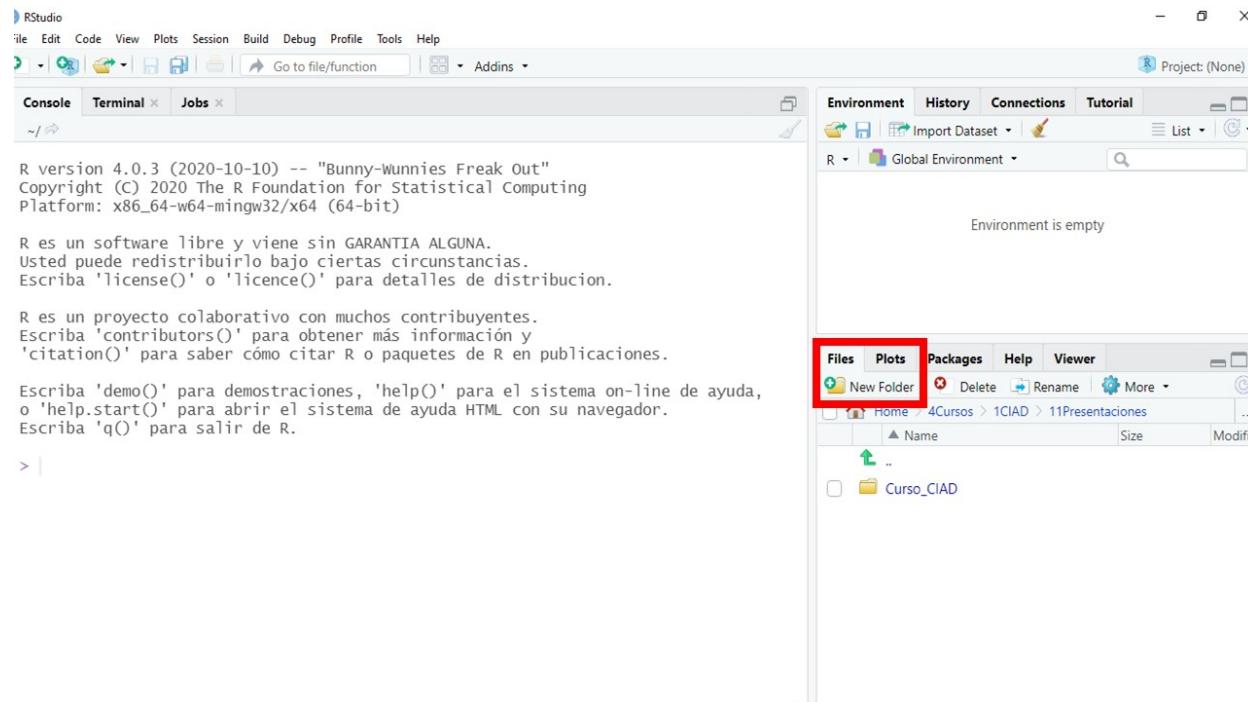
Ejercicios



- Vamos a descargar datos de library(datos) y los van a guardar en su computadora usando el argumento write_csv
- Vamos a crear un script con código que debe incluir: número de muestras y el promedio de alguna variable del data.frame
- Vamos a exportar un gráfico en la carpeta resultados
- Vamos a cambiar su proyecto de carpeta y lo van a volver a abrir y volver a correr el script

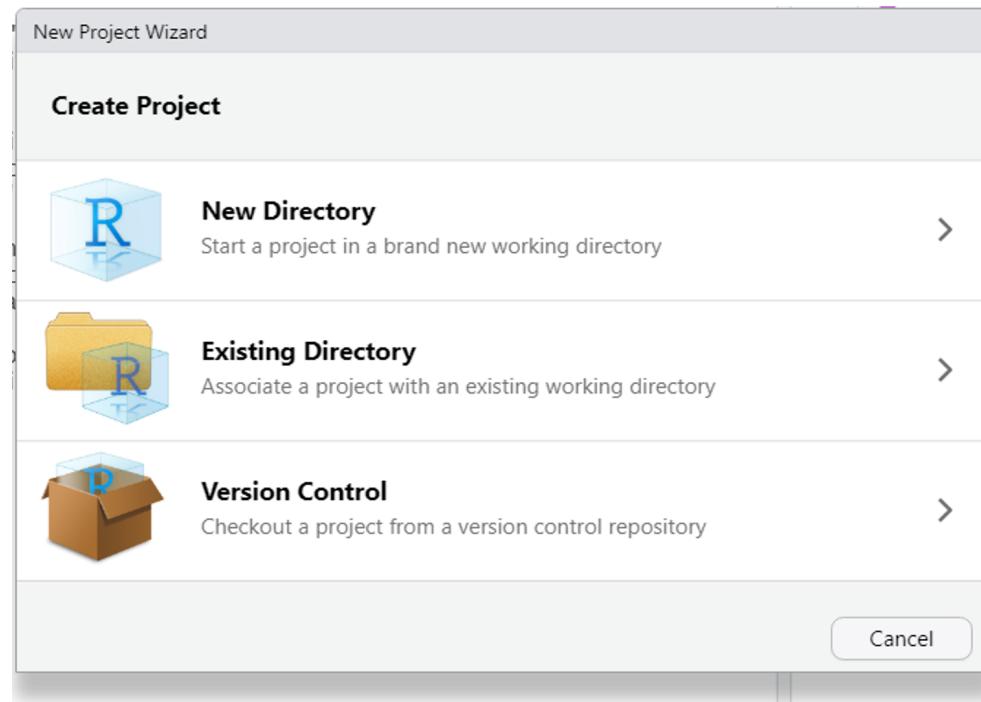
2.1. Directorio

- Crea un folder para tu proyecto
- Crea un subfolder que se llame datos (01Datos)
- Crea un subfolder que se llame código (02Scripts)
- Crea un subfolder que se llama resultados (03Resultados)



2.2. Crear un proyecto

Haz clic en File > New Project, y después:



Traten de usar un nombre significativo (p.e. 'ProyectoPingus')

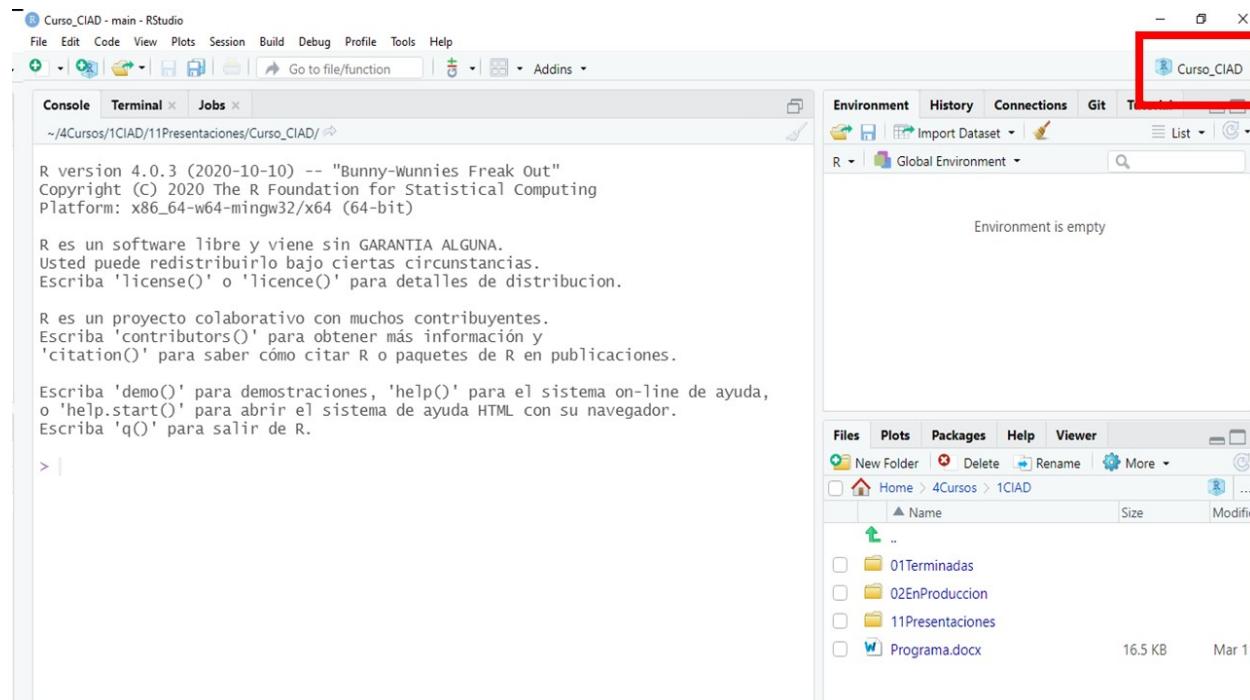
2.2. Crear un proyecto

Que cambio?

En la esquina superior derecha sale el nombre de su proyecto.

En donde estas trabajando? Revisa con getwd.

Revisa con here.



2.3. Cargar datos

- Van a descargar datos de library(datos) y los van a guardar en su compu usando el argumento write_csv en su subcarpeta **01Datos**.

```
library(tidyverse)
library(datos)
library(here)
Pingus<-pinguinos
```

1- Define la carpeta.

```
DatosFolder<-here("01Datos")
```

2- Exporta el objeto a csv.

```
write_csv(Pingus, file =paste0(DatosFolder, '/DatosPinguinos.csv'))
```

Ve en files 01Datos si aparece tu nuevo csv.

2.4. Script

- Van a crear un **script** con codigo debe incluir:

1. Como cargar mis datos
2. Cual es mi numero de muestras por sexo
3. Cual es el promedio de peso por sexo

2.4. Script

File>NewFile>NewScript

- Escribir y cargar los paquetes

```
library(here)
library(tidyverse)
```

- Cargar sus datos

```
here()
read_csv('..//01Datos//Pinguinos.csv')
```

2.4. Script

- Calcular numero de muestras por sexo

```
Pingus %>%
  group_by(sexo) %>%
  tally()
```

```
## # A tibble: 3 x 2
##   sexo      n
## * <fct>  <int>
## 1 hembra    165
## 2 macho     168
## 3 <NA>      11
```

- Calcular el promedio de peso por sexo

```
Pingus %>%
  group_by (sexo) %>%
  summarise(promedio=mean(masa_corporal_g))
```

```
## # A tibble: 3 x 2
##   sexo    promedio
## * <fct>    <dbl>
```

2.5. Figuras

- Van a exportar un gráfico en su carpeta resultados.
Un boxplot de pesos de hembras y machos separados por especie.

```
MiGrafico<-ggplot(Pingus,
                     aes(x=especie,
                         y=masa_corporal_g,
                         color=sexo,
                         fill=sexo))+  
geom_boxplot()+
theme_bw()
```

2.5. Figuras

Elige el directorio donde quieras guardar tu gráfico.

```
ResultsFolder<-here::here("03Resultados")
```

Si no usan here, escriban la dirección completa del Folder.

```
MiGrafico  
ggsave("MiGrafico.jpg",  
       path = ResultsFolder, #tambien la pueden escribir aqui  
       width = 16, height = 8,  
       units = "in") #puede ser centimetros
```

2.5. Guardar script

Revisar tener:

- DatosPinguinos.csv en carpeta de 01Datos
- MiScript.R en la carpeta de 02Script
- MiGrafico.jpg en la carpeta de 03Resultados

2.6. Limpiar

- Limpiar su environment
- Limpiar su consola

2.7. Reusar script

Abrir su script (MiScript.R).

Hacer segundo gráfico.

File>Save as>02_grafico_longitudpico_pingus.

Modifiquemos **todas** las partes donde dice masa_corporal_g por largo_pico_mm

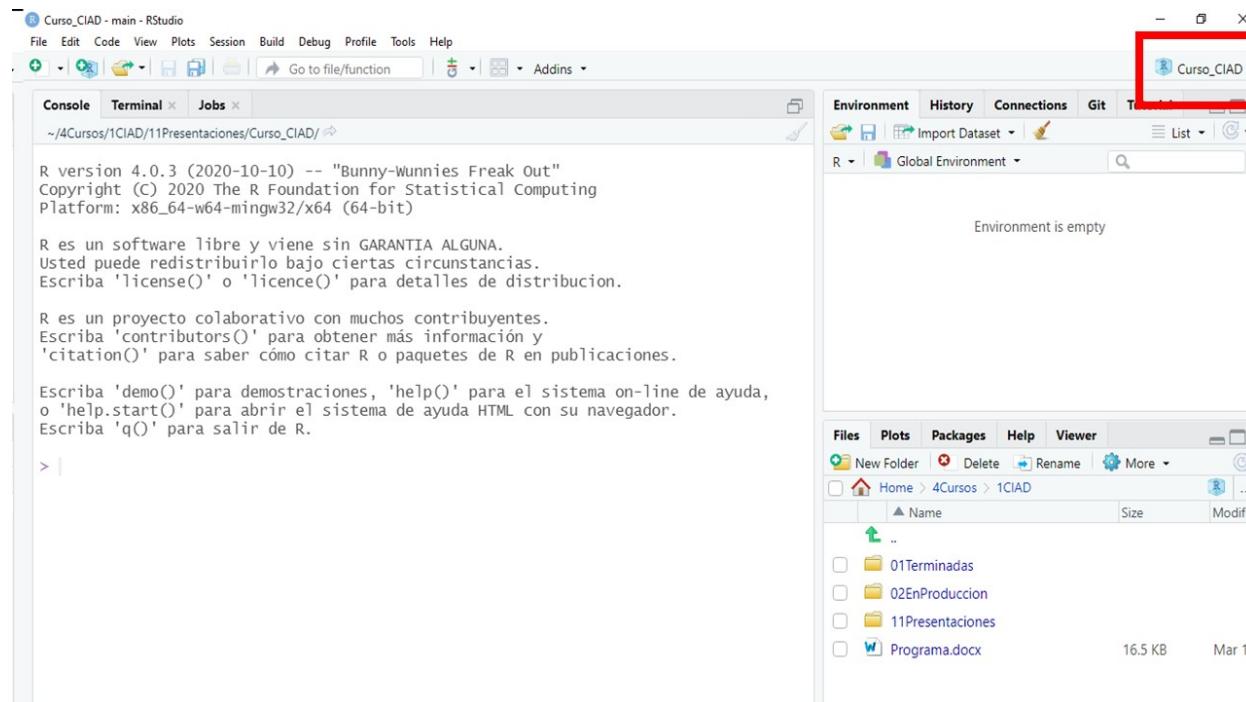
Cambien en ggsave

```
ggsave("02_LargoPico.jpg",
       path = ResultsFolder,
       width = 16, height = 8,
       units = "in")
```

2.8. Mover carpeta

Cierren su RStudio.

- Muevan la carpeta a otra carpeta.
- Abran de nuevo su proyecto picandole al Rproj o abriendo R y moviendose al proyecto.
- Rehagan sus analisis abriendo: O2Script>Script



A close-up photograph showing a group of people's hands clasped together in a circle. The hands belong to individuals wearing various colored knitted sweaters, including red, orange, green, blue, brown, yellow, and white. Some hands have rings on them. The hands are positioned in the center of the frame, with the background being a soft-focus view of the same or similar hands.

Compartir el proyecto

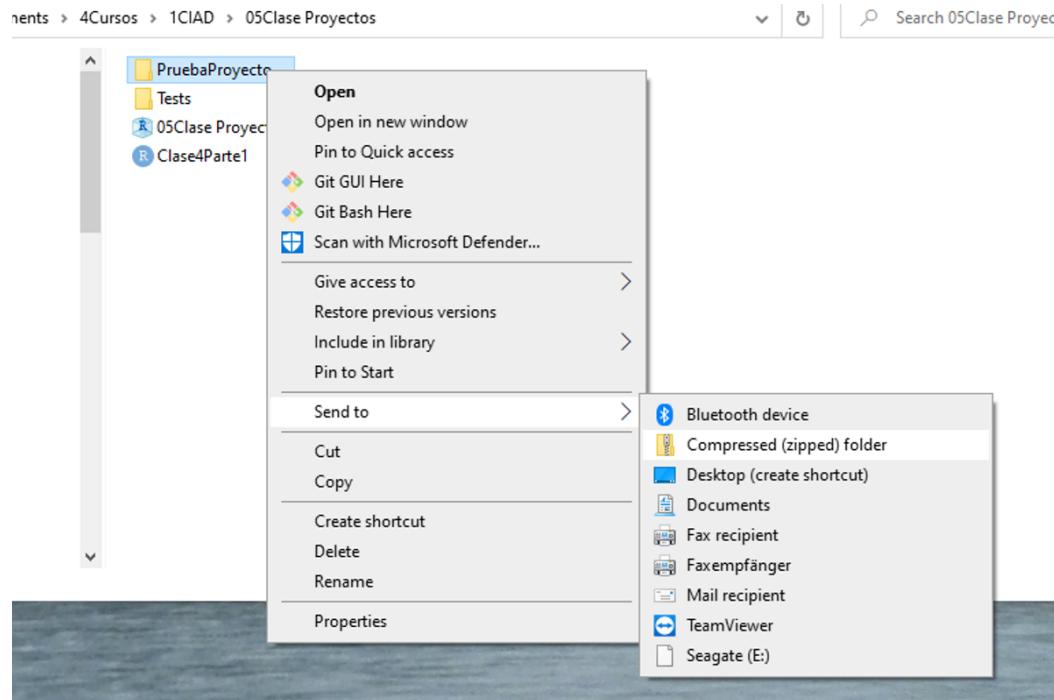
3. Zip

Ahora vamos a:

- Crear un archivo zip
- Bajar un archivo zip y replicar el trabajo

3.1. Crear un zip del proyecto y enviarlo.

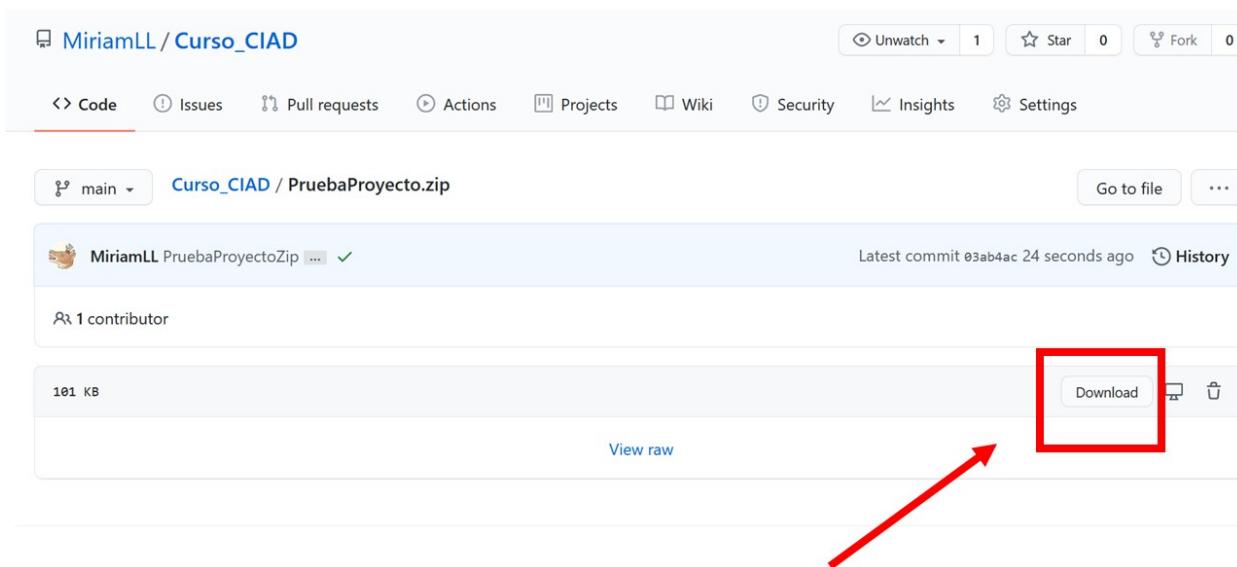
- Crea tu archivo zip.



3.2. Bajar un zip

Enlace github bajar el zip

↗ Enlace Proyecto.



Unzip>Correr el script>Replicar analisis.

Recapitulando

Esta clase:

- Trabajar por proyecto.
- Compartir proyectos.

Siguiente clase:

- Sintaxis de modelos lineares.
- Línea de ajuste en nuestros gráficos.

Contacto

Para dudas, comentarios y sugerencias:
Escríbeme a miriamjlerma@gmail.com

Este material está accesible y se encuentra en
mi [github](#) y mi [página](#)

