

# Introducción al flujo de investigación reproducible

## Introducción al flujo de investigación reproducible

- La presentación estará disponible en <https://lisa-coes.github.io/presentaciones/escuela-elsoc-julio2024/escuela-elsoc-julio2024.html#1>
- Este taller estará disponible en <https://lisa-coes.github.io/presentaciones/escuela-elsoc-julio2024/taller-elsoc-julio2024.html>

## Taller práctico

### Prerequisitos

- Crear cuenta en [www.github.com](https://www.github.com)
- Descargar Github Desktop
- Pronto actualización de este documento

## Github

### Descripción

Github es una plataforma de desarrollo colaborativo que permite alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas (software).

### Note

El 4 de junio de 2018 Microsoft compró GitHub por la cantidad de 7500 millones de dólares. Al inicio, el cambio de propietario generó preocupaciones y la salida de algunos proyectos de este sitio; sin embargo, no fueron representativos. GitHub continúa siendo la plataforma más importante de colaboración para proyectos de código abierto.

## Repositorios

Un repositorio contiene todo el código, tus archivos y el historial de revisiones y cambios de cada uno de ellos. Es el elemento más básico de Github.

Los repositorios pueden contar con múltiples colaboradores y pueden ser públicos o privados.

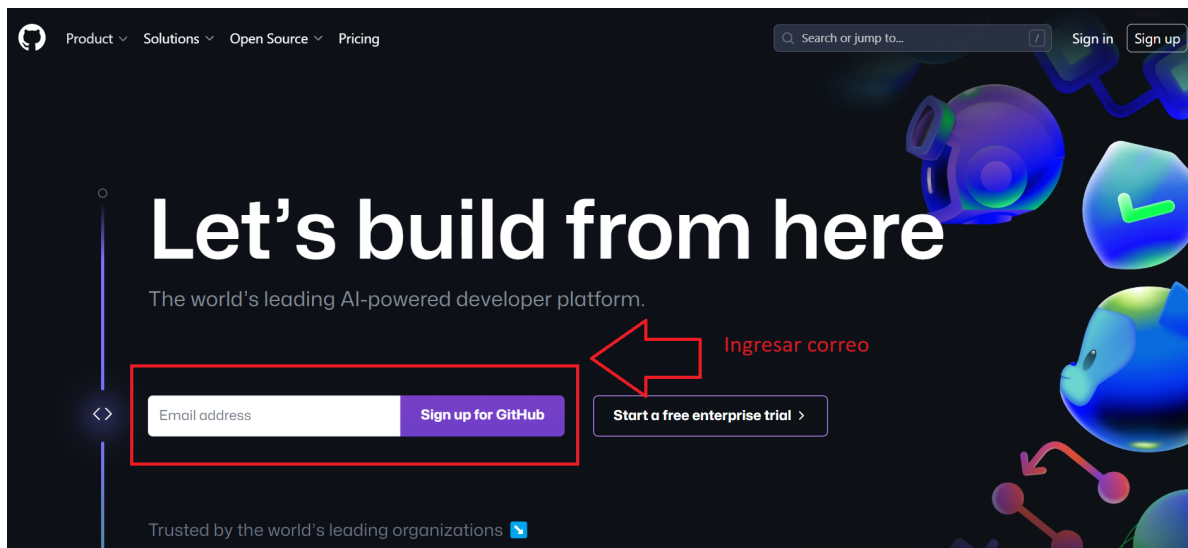
## Principales términos

Término	Definición
Branch	Una versión paralela del código contenido en el repositorio, pero que no afecta a la rama principal.
Clonar	Para descargar una copia completa de los datos de un repositorio de GitHub.com, incluidas todas las versiones de cada archivo y carpeta.
Fork	Un nuevo repositorio que comparte la configuración de visibilidad y código con el repositorio «ascendente» original.
Merge	Para aplicar los cambios de una rama y en otra.
Pull request	Una solicitud para combinar los cambios de una branch en otra.
Remote	Un repositorio almacenado en GitHub, no en el equipo.
Upstream	La branch de un repositorio original que se ha <i>forkeado</i> o clonado. La branch correspondiente de la branch clonada o <i>forkeada</i> se denomina «descendente».

## Crear cuenta e instalación

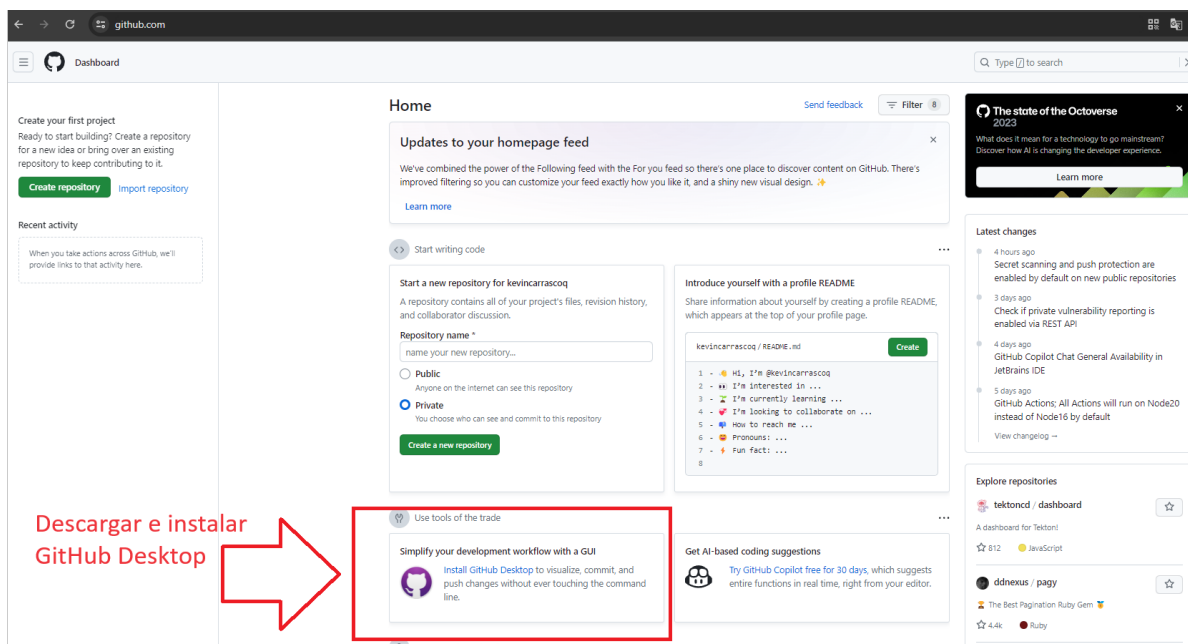
1. Acceder a la página de [github](#)

Registrarse ingresando correo electrónico y siguiendo los pasos siguientes (crear contraseña y nombre de usuario)



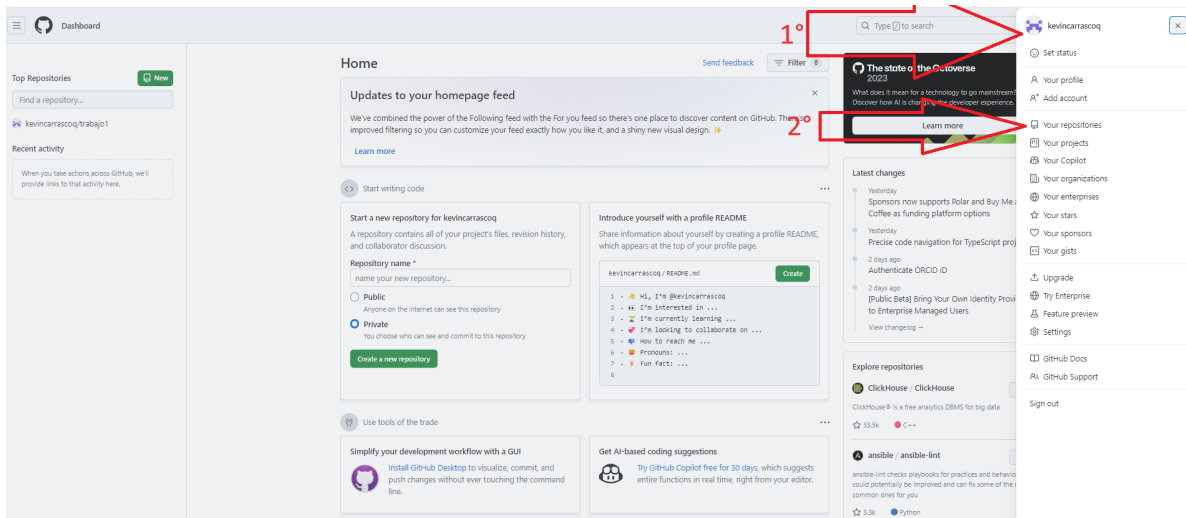
La personalización de la cuenta se puede saltar haciendo click en **skip** abajo de la selección de opciones

## 2. Descargar e instalar Github Desktop

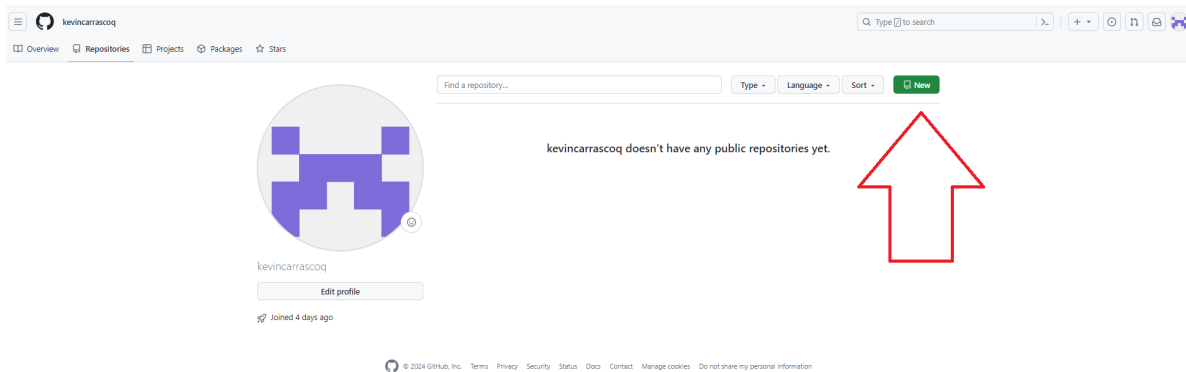


## Crear repositorio

En la página principal de [github](#) hacer click en el ícono de usuario de la esquina superior derecha y luego ir a Tus repositorios



Una vez accedemos a Tus repositorios hacemos click en New/Nuevo



Luego le ponemos un nombre a nuestro repositorio, evitando siempre espacios, ñ y tildes, y apretamos Crear repositorio

**Create a new repository**

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \* kevincarrasco / Repository name \* R-data-analysis 1°

R-data-analysis is available.

Great repository names are short and memorable. Need inspiration? How about [psychology-system-7](#)

Description (optional)

☐ Public  
Anyone on the internet can see this repository. You choose who can commit.

☐ Private  
You choose who can see and commit to this repository.

Initialize this repository with:

☐ Add a README file  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

☐ You are creating a public repository in your personal account.

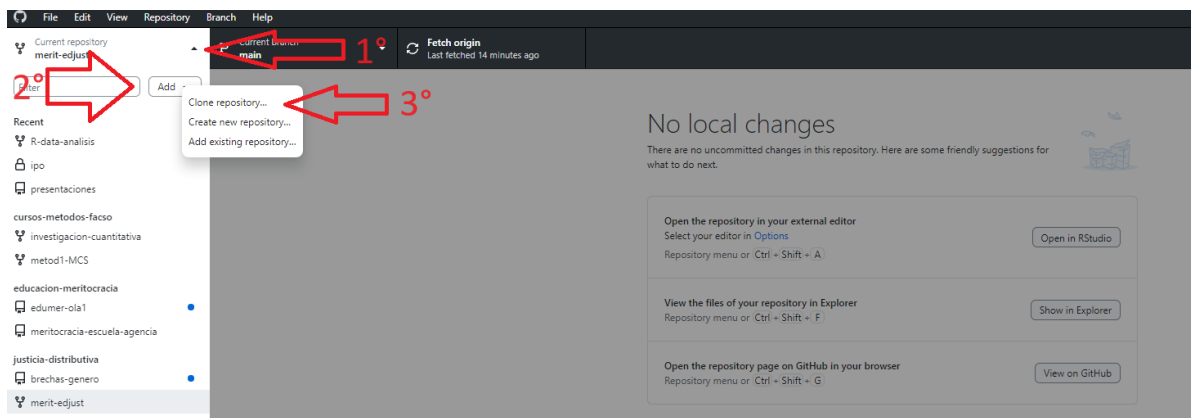
Create repository 2°

## Github desktop

Una vez creado un repositorio, lo que nos interesa es descargarlo. Al abrir la aplicación de Github desktop por primera vez (descargada anteriormente), nos debería aparecer la opción de clonar nuestro repositorio R-data-analysis en la pantalla de inicio. Lo clonamos y seleccionamos una carpeta de nuestro computador para almacenarlo.

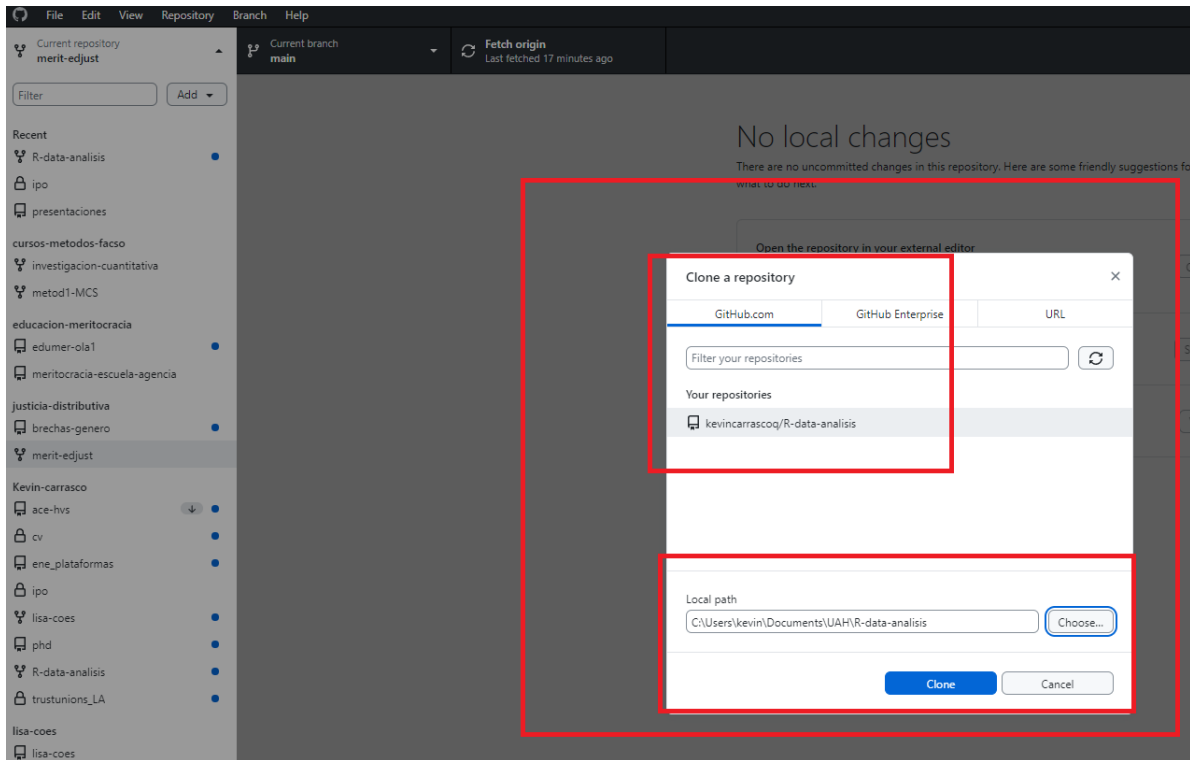
Para todas las siguientes veces, las instrucciones son estas:

- 1- Apretamos Repositorio actual en la esquina superior izquierda
- 2- Apretamos añadir
- 3- Apretamos clonar repositorio...



4- Seleccionamos nuestro repositorio

5- seleccionamos la carpeta donde se almacenará. Siempre evitando tener tildes, ñ y espacios en la dirección de almacenamiento y apretamos 'clone'.



6- Vamos al repositorio de LISA y descargamos el template de ipo. Link directo acá: [https://github.com/lisa-coes/ipo/tree/master/IPO\\_template](https://github.com/lisa-coes/ipo/tree/master/IPO_template)

7- Lo guardamos en la carpeta que creamos recién desde github desktop

## Quarto

La escritura en Quarto tiene algunos códigos o funciones, aquí un resumen de su mayoría:

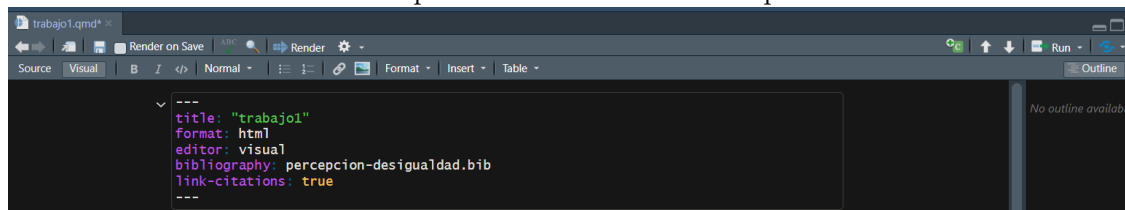
Código	Así se ve
Algo de texto en el párrafo.	Algo de texto.
Más texto espacio entre lineas.	Algo de texto en el párrafo. Siempre utilizando espacios para dividir párrafos
*Cursivas*	<i>Cursivas</i>
**Negrita**	<b>Negrita</b>

Código	Así se ve
# Título 1	<b>Título 1</b>
## Título 2	<b>Título 2</b>
### Título 3	<b>Título 3</b>
(puedes llegar hasta un título N° 6 con #####)	
[Texto enlace] ( <a href="https://quarto.org/">https://quarto.org/</a> )	<a href="#">Texto enlace</a>
> Citas	Citas
1. Una	1. Una
2. lista	2. lista
3 ordenada	3. ordenada
- Otro	• Otro
- tipo	• tipo
- de lista	• de lista

1. Abrimos nuestro Rproject y creamos un nuevo documento de Quarto file -> new file -> Quarto document
2. Editamos el yaml, agregando bibliography: percepcion-desigualdad.bib y link-citations: yes

#### Note

YAML: Lenguaje de programación. Es un formato de serialización de datos que proporcionan un mecanismo de intercambio de datos legible por humanos. Dan formato a los datos de manera estandarizada para su intercambio entre aplicaciones de software.



```

---
title: "trabajo1"
format: html
editor: visual
bibliography: percepcion-desigualdad.bib
link-citations: true
---

```

```

---
title: "Mi Documento"
format:

```

```
html:
  toc: true
  number-sections: true
---
```

Luego, podemos escribir en el documento, separando por títulos (#) cada sección. La jerarquía de los títulos se establece según la cantidad de '#’.

A continuación, en esta guía combinaremos el paso-a-paso de crear un documento dinámico con quarto, a la vez que vamos viendo distintas funciones de este proceso.

Por ejemplo, como hacer una nota al pie<sup>1</sup>. Para hacerlo, solo debemos escribir [ ^2] pero sin el espacio entre los corchetes. Luego, en otra línea escribimos [^2]: Esta es la nota al pie

## Código de análisis de ejemplo

Para poder escribir código de análisis en un documento Quarto debemos generar trozo de código llamado ‘Chunk’, que se puede crear con ctrl+alt+i o directamente en el menú de arriba en ‘Code -> Insert Chunk’.

### Cargar paquetes

```
pacman::p_load(sjlabelled,
               dplyr, #Manipulacion de datos
               stargazer, #Tablas
               sjmisc, # Tablas
               summarytools, # Tablas
               kableExtra, #Tablas
               sjPlot, #Tablas y gráficos
               corrplot, # Correlaciones
               sessioninfo, # Información de la sesión de trabajo
               ggplot2) # Para la mayoría de los gráficos
```

### Cargar bases de datos

Cargamos ambas bases de datos desde internet

---

<sup>1</sup>Esta es la nota al pie



```
load(url("https://github.com/Kevin-carrasco/R-data-analisis/raw/main/files/data/latinobarometro"))
load(url("https://github.com/Kevin-carrasco/R-data-analisis/raw/main/files/data/data_wvs.RData"))
```

Para trabajar con ambas bases, agruparemos las variables de interés por país, por lo que ya no trabajaremos directamente con individuos.

```
context_data <- wvs %>% group_by(B_COUNTRY) %>% # Agrupar por país
  summarise(gdp = mean(GDPpercap1, na.rm = TRUE), # Promedio de GDP per capita
            life_exp = mean(lifeexpect, na.rm = TRUE), # Promedio esperanza de vida
            gini = mean(giniWB, na.rm = TRUE)) %>% # Promedio gini
  rename(idenpa=B_COUNTRY) # Para poder vincular ambas bases, es necesario que la variable de
context_data$idenpa <- as.numeric(context_data$idenpa) # Como era categórica, la dejamos numérica

proc_data <- proc_data %>% group_by(idenpa) %>% # agrupamos por país
  summarise(promedio = mean(conf_inst, na.rm = TRUE)) # promedio de confianza en instituciones
```

## Unir bases de datos

Para vincular nuestras bases de datos existen múltiples opciones, la primera es ‘merge’ de R base y las siguientes tres vienen desde dplyr: ‘right\_join’, ‘full\_join’ y ‘left\_join’. Cada una tiene sus propias potencialidades y limitaciones y dependerá de cada caso cuál usemos

### Probemos merge

```
data <- merge(proc_data, context_data, by="idenpa")
```

```
data <- data %>%
  mutate(idenpa = as.character(idenpa)) %>%
  mutate(idenpa = case_when(
    idenpa == "32" ~ "Argentina",
    idenpa == "68" ~ "Bolivia",
    idenpa == "76" ~ "Brasil",
    idenpa == "152" ~ "Chile",
    idenpa == "170" ~ "Colombia",
    idenpa == "188" ~ "Costa Rica",
    idenpa == "214" ~ "Cuba",
    idenpa == "218" ~ "República Dominicana",
    idenpa == "222" ~ "Ecuador",
    idenpa == "320" ~ "El Salvador",
```

```

idenpa == "340" ~ "Guatemala",
idenpa == "484" ~ "Honduras",
idenpa == "558" ~ "México",
idenpa == "591" ~ "Nicaragua",
idenpa == "600" ~ "Panamá",
idenpa == "604" ~ "Paraguay",
idenpa == "858" ~ "Uruguay",
idenpa == "862" ~ "Venezuela"))

```

```

data$gdp <- as.numeric(data$gdp)
data$gdp[data$gdp==0] <- NA
data <- na.omit(data)

```

**Guardamos esta nueva base en nuestra carpeta input**

```

save(data, file="input/data/proc/data.RData")

```

## Visualizaciones

Podemos establecer referencias cruzadas para las tablas y gráficos dentro del texto, para poder automatizarlo, como ejemplo así, pero dentro del chunk:

```

#| label: tbl-sjmisc
#| tbl-cap: "Descriptivos con sjmisc"

```

## Descriptivos

El Chunk se debería ver así:

```

#| label: tbl-sjmisc
#| tbl-cap: "Descriptivos con sjmisc"
sjmisc::descr(data,

  show = c("label", "range", "mean", "sd", "NA.prc", "n")) %>% # Selecciona estadísticos

  kable(., "markdown") # Esto es para que se vea bien en quarto

```

```
sjmisc::descr(data,
  show = c("label", "range", "mean", "sd", "NA.prc", "n"))%>% # Selecciona estadísticos
  kable(., "markdown") # Esto es para que se vea bien en quarto
```

Table 3: Descriptivos con sjmisc

	var	label	n	NA.prc	mean	sd	range
4	promedio	promedio	11	0	3.40077	1.016976	3.59 (2.3-5.9)
1	gdp	gdp	11	0	15528.18364	6480.045512	19523.79 (5631.2-25154.99)
3	life_exp	life_exp	11	0	75.90909	2.286593	8.8 (71.24-80.04)
2	gini	gini	11	0	45.46364	4.156266	14.2 (39.7-53.9)

Luego de establecer el link y el nombre de la tabla, podemos referenciar acá con un @, así: @tbl-sjmisc (pero junto), y que se vería así [Table 3](#)

## Gráficos

Y para los gráficos se hace de la misma forma:

```
#| label: fig-gdp
```

```
#| fig-cap: "Plots"
```

```
graph1<-ggplot(data, aes(x = idenpa, y = gdp)) +
  geom_point() +
  labs(x = "País", y = "Gdp") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

graph1
```

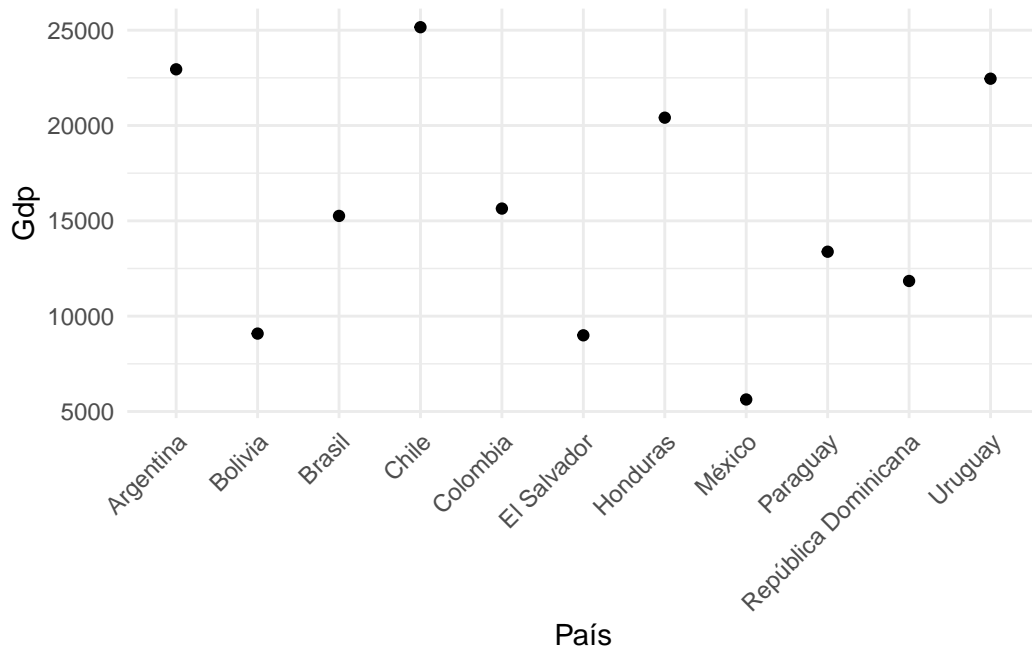


Figure 1: Producto interno bruto por país

Sin embargo la Figure 1 entrega información desordenada. Mejor ordenar por tamaño de PIB que por orden alfabético de los países. Para eso

```
data_sorted <- data %>% arrange(desc(gdp))
graph2<-ggplot(data_sorted, aes(x = factor(idenpa, levels = idenpa), y = gdp)) +
  geom_point() +
  labs(x = "País", y = "GDP") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

graph2
```

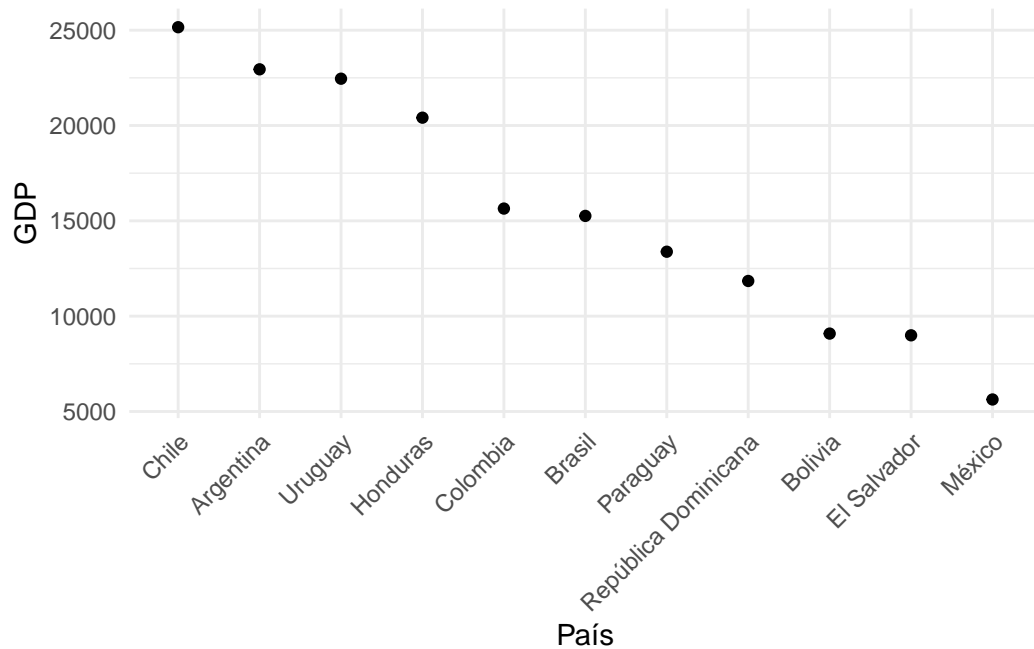


Figure 2: Producto interno bruto por país ordenado

Ahora sí la Figure 2 muestra un gráfico más ordenado.

### Guardamos este nuevo gráfico en la carpeta output

```
ggsave(graph2, file="output/graphs/graph2.png")
```

Y comparar el promedio de confianza en instituciones según producto interno bruto por país?

```
data %>%
  ggplot(aes(x = gdp, y = promedio, label = idenpa)) +
  geom_point() +
  geom_text(vjust = -0.5) +
  labs(x = "GDP", y = "Promedio") +
  theme_bw()
```

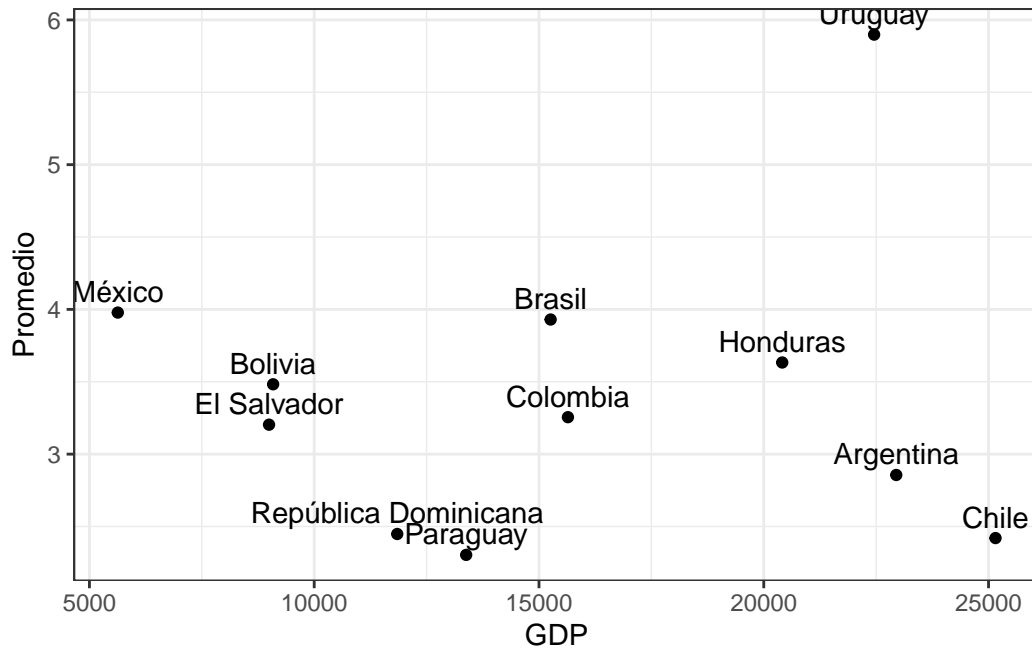
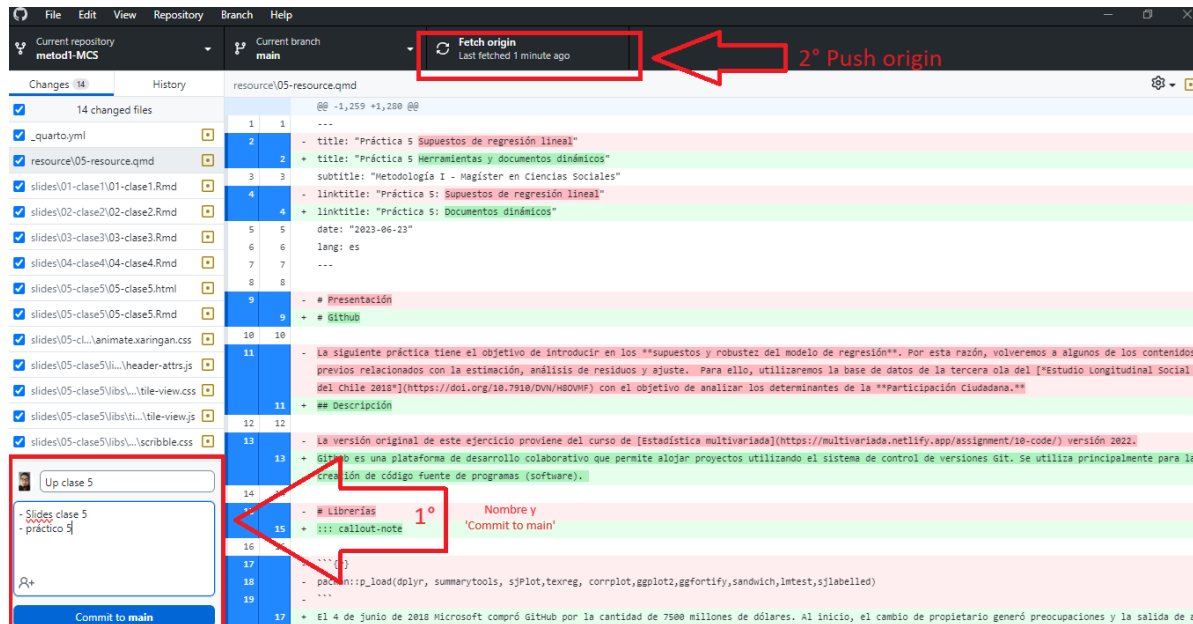


Figure 3: Confianza en instituciones según el producto interno bruto por país

La Figure 3 muestra la relación que existe entre el producto interno bruto y la confianza en instituciones para los 18 países analizados. Es interesante comparar los casos de Chile y Uruguay, que al tener similar GDP, tienen un nivel de confianza en instituciones muy diferente.

5. Luego renderizamos y se debería ver así:
6. Ahora que tenemos nuestra investigación podemos subirla a Github Pages a través de Github Desktop.

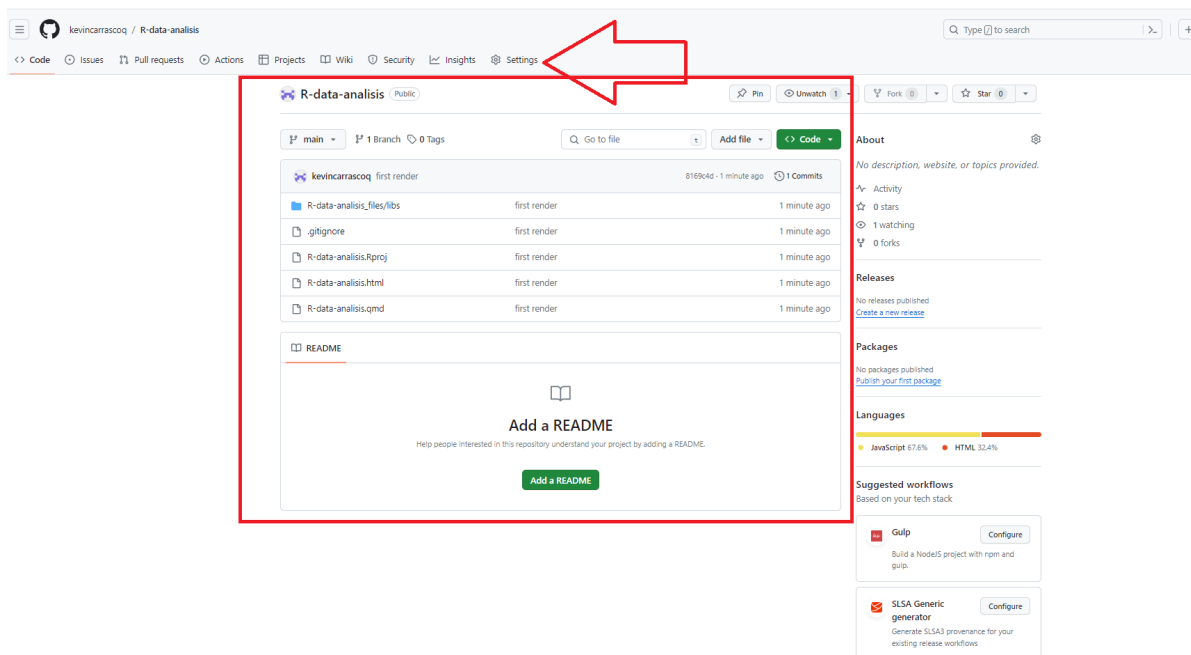
## Github desktop



## Github pages

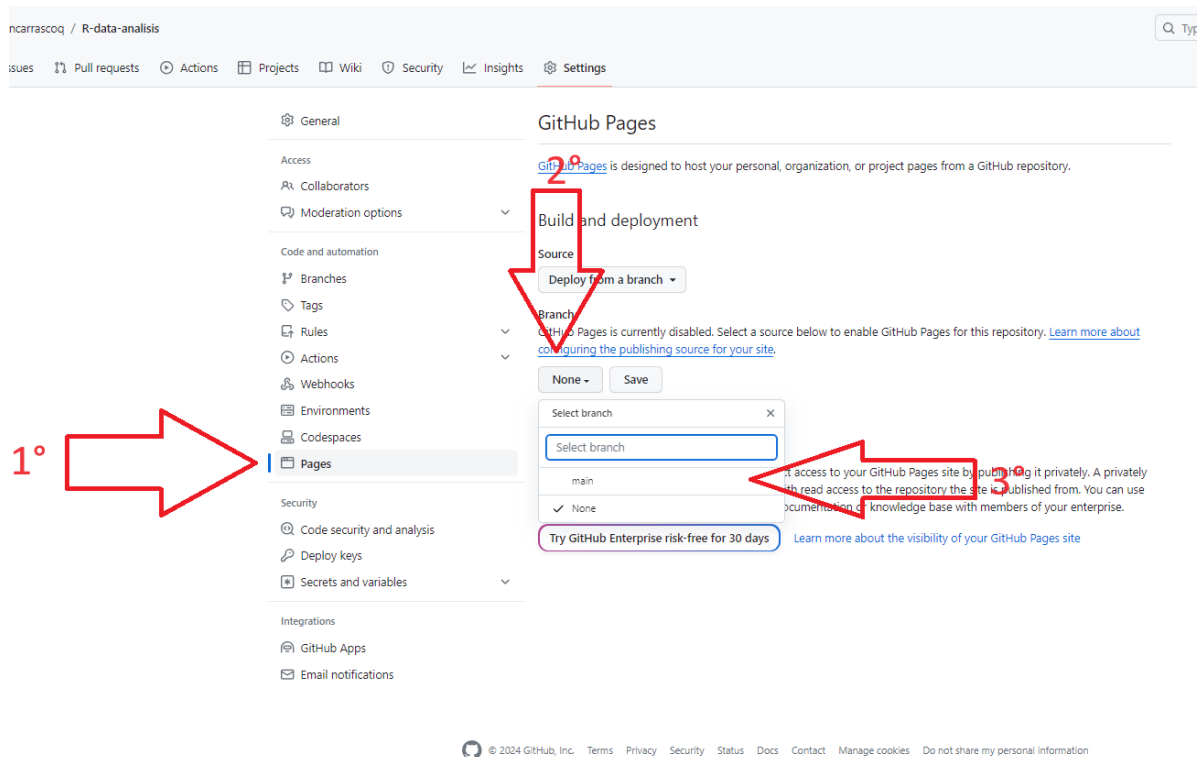
Ahora podemos ver los documentos modificados en nuestro repositorio online de github.

### 7. Vamos a settings



8. Dentro de Settings vamos a Pages, luego 'none' y seleccionamos 'main'. Luego apretamos Save

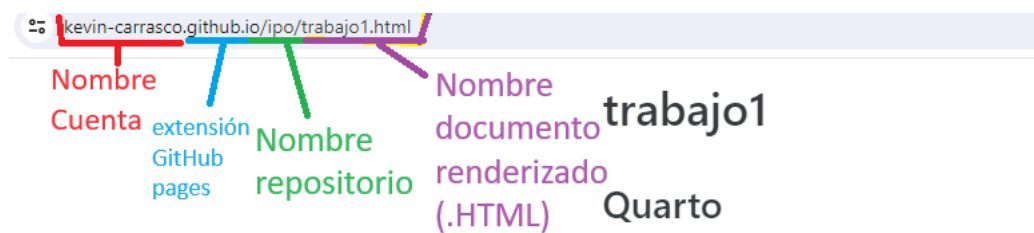




Luego de aproximadamente un minuto se actualiza la página y aparecerá un link en la parte superior, algo así como [kevin-carrasco.github.io/ipo](https://kevin-carrasco.github.io/ipo) que es nuestra página principal de nuestro sitio web de github.

El link para llegar a nuestro documento renderizado de quarto sigue la estructura del repositorio:

[kevin-carrasco.github.io/ipo/trabajo.html](https://kevin-carrasco.github.io/ipo/trabajo.html)



Quarto enables you to weave together content from various sources. Learn more about Quarto see <https://quarto.org>.