

Talleres de Análisis Político I

Sesión 1

5-6/11/2023

Pau Vall-Prat

pau.vall@uc3m.es

Preliminares

Presentación

Pau Vall-Prat

- Grado y Máster en Ciencia Política (UPF)
- Máster en Ciencia Política y Economía Política (LSE)
- Doctor en Ciencia Política (UB)
 - Estancias en Yale y Göteborg
 - Tesis: *The Political Economy of Regional Elite Splits*
- Investigador posdoctoral (UC3M–IC3JM)

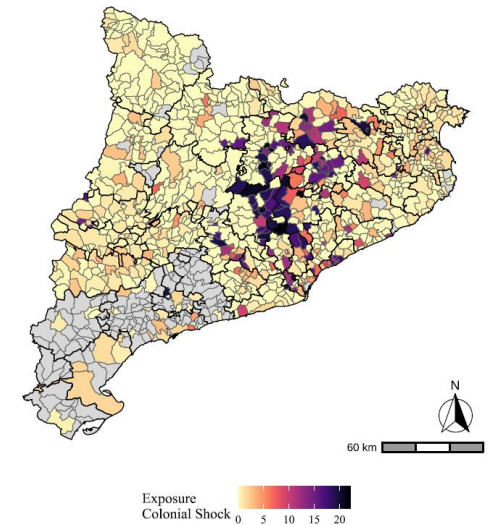
Agenda de investigación

- Economía política histórica y política comparada
 - Transiciones de régimen político
 - Elecciones en autocracias
 - Competición entre elites
 - Orígenes de partidos regionales
 - Patrones de uso de métodos de fraude electoral
 - Crisis bancarias y apoyo democrático
- También: formación de gobiernos y política multinivel

Para más información: www.pauvallprat.com

Metodología

- Eminentemente cuantitativa
- Datos a nivel micro
 - Municipal o sección censal →
 - Individual
- Recogida y sistematización de datos a partir de nuevas fuentes (archivos) o combinación de datos existentes
- Aplicación de técnicas estadísticas a partir de métodos de inferencia causal



Exposición al choque económico causado por el *Desastre* colonial de 1898. Datos a nivel municipal

Presentaciones

Vuestro turno

- Nombre
- ¿De dónde sois?
 - Ciudad
 - Universidad
- Estudios previos
- ¿Experiencia previa en softwares estadísticos?
- Principales intereses de investigación/análisis
- Figura política que admiréis
- Politólogo de cabecera
- Base de datos preferida

Aspectos prácticos

- Dos bloques de 1.15h
- En cada sesión:
 - Introducción “teórica”
 - Sobre todo práctica en R
- Evaluación
 - 10% Participación
 - 40% Ejercicios en clase → 2 entregas
 - 50% Trabajo final

Dudas y contacto

1. Plan docente colgado en el aula virtual y en GitHub
2. Presencial: 30' antes y después del horario de clase
 - Mandadme un correo
3. Correo electrónico: pau.vall@uc3m.es
 - Si no contesto en 24/48h (laborables), insistid

(Re-)Introducción a R

¿Qué es?

- Es un entorno de programación útil para
 - Manipulación de datos
 - Cálculo y análisis de datos
 - Representación gráfica
 - Producción de documentos
- Proyecto colaborativo
 - Comunidad entusiasta
 - Miles de paquetes
 - Se puede usar en cualquier ordenador
 - ¡Gratuito!

Valorando R

Ventajas

- Software libre
- Múltiples funcionalidades
- Potencia elevada para hacer casi cualquier cosa
- Obliga a pensar de forma diferente en los datos
- Permite trabajar con múltiples datos a la vez
- Facilita reproducibilidad

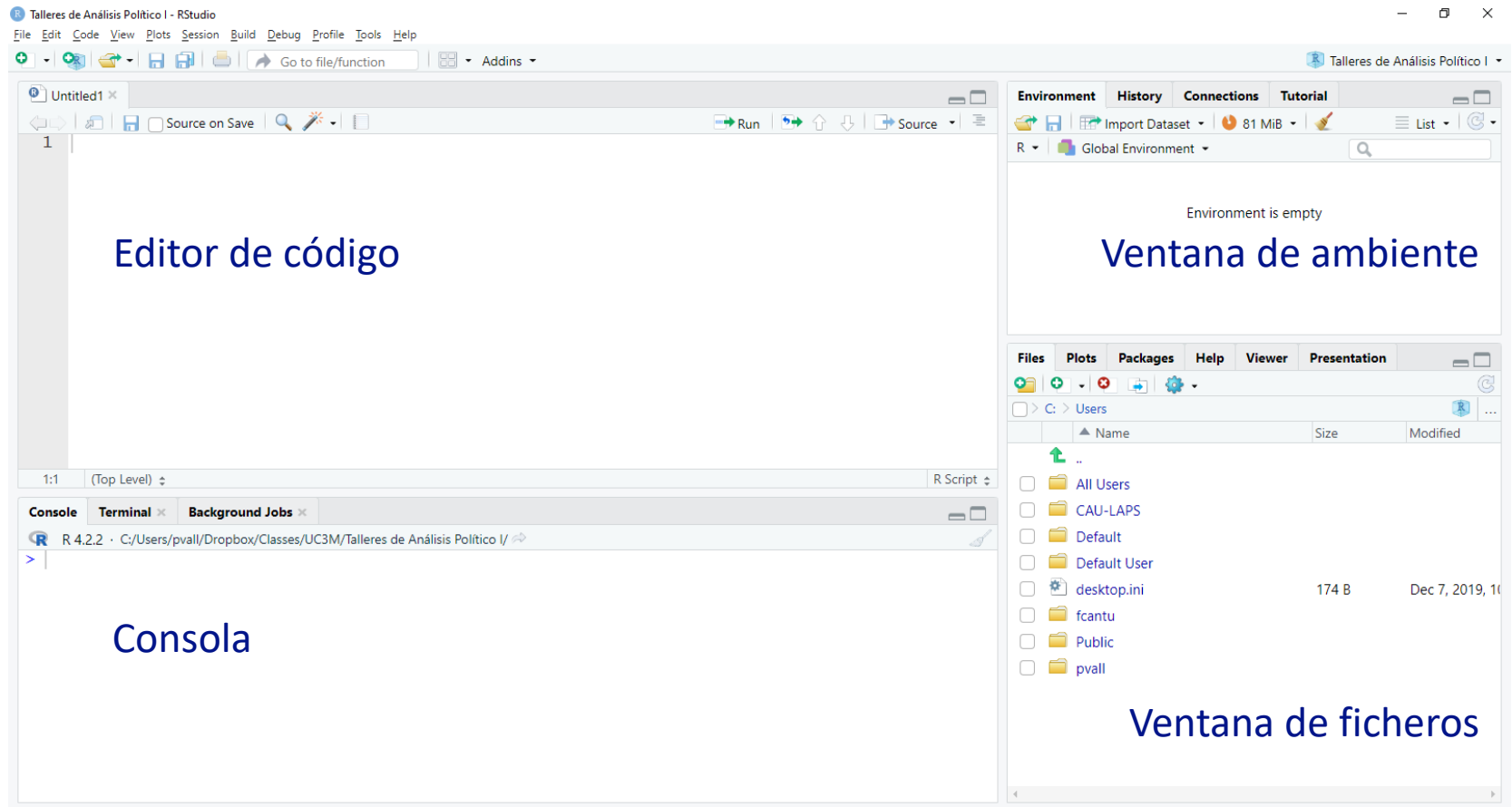
Desventajas

- Curva de aprendizaje pronunciada
- No es muy visual, dificultad de identificar problemas
- Limitaciones para trabajar con grandes bases de datos

Ante la duda

- Stackoverflow
- Tutoriales y manuales
 - An introduction to R <https://cran.r-project.org/doc/manuals/r-release/R-intro.html>
 - R for Data Science <https://r4ds.had.co.nz/>
 - Quantitative Social Science:
<https://press.princeton.edu/books/hardcover/9780691167039/quantitative-social-science>
 - R en Ciencias Sociales: <https://jmas.quarto.pub/r-en-ciencias-sociales-ii/>
 - DataCamp, Coursera, edX...

RStudio



¿Para qué sirve cada ventana?

- Editor de código

- Ficheros donde se escribe el código. Elemento clave para la reproducibilidad

- Consola

- Muestra los resultados del código ejecutado
- También permite ejecutar código de forma rápida
¡no recomendable!

- Ventana de ambiente

- Muestra los objetos almacenados

- Ventana de ficheros/gráficos...

- Permite visualizar resultados del trabajo: gráficos, html...
- Encontrar ficheros
- Ver detalles de los paquetes empleados, etc.

Paquetes

- Sin paquetes, R no haría nada
- El paquete es un conjunto de código, datos y documentación que permiten implementar funciones predefinidas.
- Por defecto, vienen algunos ya cargados
 - Descubrid cuales con la función `search()`
- Se pueden instalar
 - CRAN: `install.packages("nombrepaquete")`
 - GitHub: `devtools::install_github("nombrepaquete")`
- Instalar una vez, cargar en cada sesión que se quiera emplear el paquete

Un paquete para gobernarlos a todos



- El paquete pacman
- Permite cargar directamente distintos paquetes e instalarlos si no lo están
- Una única función permite
 - Instalar +
 - Cargar

<https://www.rdocumentation.org/packages/pacman/versions/0.5.1>

Empezar a trabajar con R

1. Crear un proyecto de R para cada trabajo o proyecto en el que se quiera trabajar
 - Hay distintos modos de hacerlo
2. Mantener un orden dentro del proyecto
 - Carpetas para cada tipo de documentación
 - Nombres de ficheros lógicos y ordenados
3. Guardar todos los códigos de forma estructurada

Una opción es usar el paquete r4np que crea un proyecto y carpetas para mantener el orden [[link](#)]

```
devtools::install_github("ddauber/r4np")
```

Buenas prácticas

Carpetas

Separadas por usos

- Datos de lectura (originales)
- Datos procesados
- Gráficos
- Tablas
- Archivos suplementarios

Archivos

- Indicar claramente qué hacen
- Nombre secuencial
 - 0_crear_base_datos
 - 1_analisis_descriptivo
 - 2_analisis_inferencial
 - 3_graficos
 - 4_apéndice

Código en R

- Pensar antes de escribir código
- El código como un medio de comunicación
 1. Entre uno y el ordenador
 2. Entre uno y otra gente
 3. Entre uno y su yo del futuro
- Crucial comentar el código, es importante describir todo lo que se está haciendo
- El código debe funcionar de inicio a fin
- Se pueden hacer pruebas en la consola, en el código sólo lo que sea realmente necesario



¿Cómo funciona R?

- Todo lo que existe es un objeto
- Todo lo que se ejecuta es una función
- Los objetos son como los sustantivos, las funciones como los verbos.
- Con “<-” asignamos objetos
- Las funciones son una palabra seguida de paréntesis
 - Comprenden argumentos, especificaciones sobre lo que se quiere hacer

Contenidos en R: objetos

- R se basa completamente en objetos, cualquier dato deberá formar parte de un objeto
- Todo elemento en R tiene un nombre
 - No hay un estándar sobre nombres, recomendaciones
 - Consistencia
 - Reconoscibilidad
 - Brevedad
 - R es sensible a mayúsculas/minúsculas
- Todo objeto tiene una clase
 - Character
 - Number
 - Integer
 - Logical
 - Vector
 - Matrix
 - Dataframe
 - List
 - Function...

Marco de datos (*dataframe*)

- Es el tipo de objeto más habitual para trabajar en R
 - Consiste de múltiples vectores del mismo tamaño
 - Cada fila corresponde a una observación
 - Cada columna corresponde a una variable
-
- Importantísimo tener claro los datos en qué unidad de observación están y que se ajuste a la unidad de análisis

Funciones: creación

- Una función siempre implica la creación de un objeto a partir de un argumento.
- Por ejemplo, si se quiere una función que devuelva la mitad del valor de un número se podría crear:

```
funcionmitad <- function(x) {  
  y <- x/2  
  return(y)  
}
```

Cread una función que permita hacer algún otro cálculo

Funciones: argumentos

- Una función puede tener más de un argumento

- Se pueden especificar distintos objetos dentro de la función

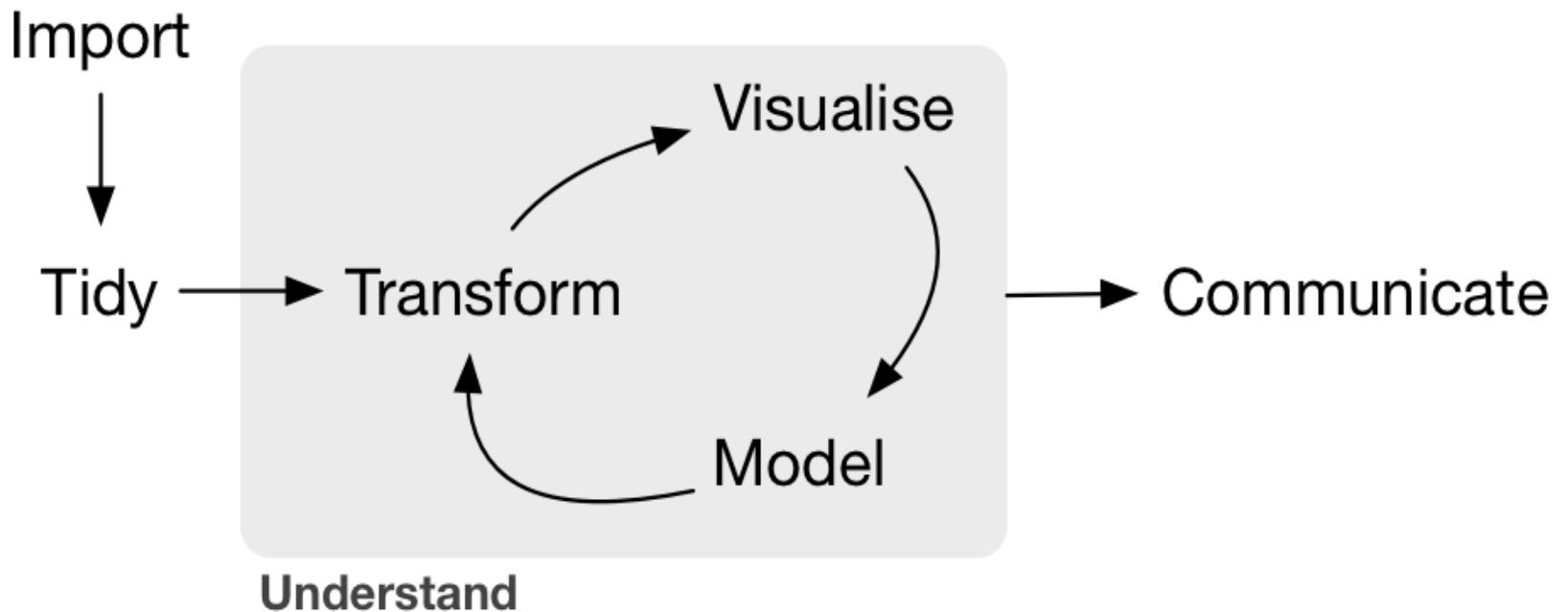
```
funciontexto <- function(x,t) {  
y <- x/2  
z <- paste0(t, " ", y)  
return(z)  
}
```

- Se podría aplicar de dos modos

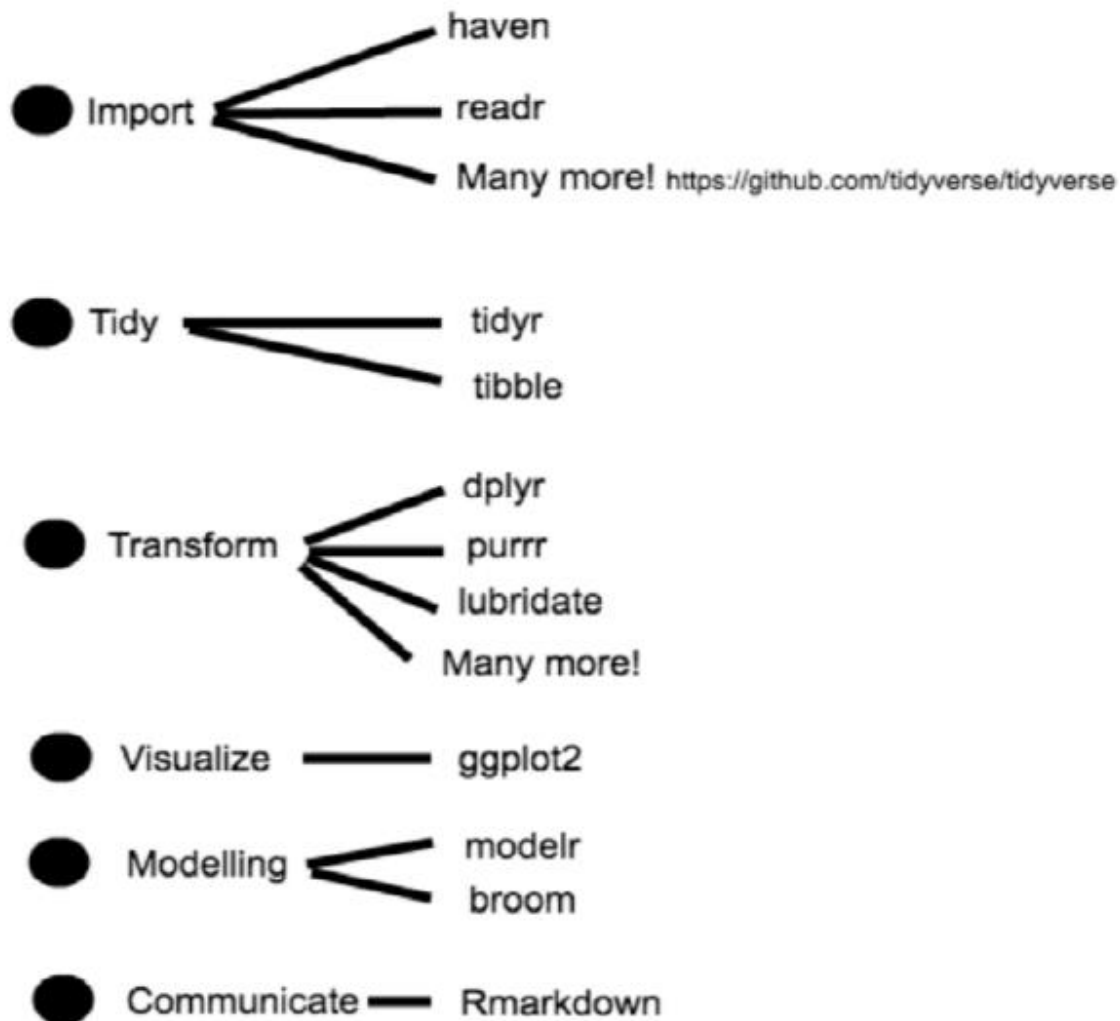
- `funciontexto(4, "Valor")`
 - `funciontexto(x = 4, t = "Valor")`

El proceso de análisis de datos

Un resumen gráfico



¿Cómo hacerlo?



Un poco de motivación, o no...

Hadley Wickham

The bad news is that when ever you learn a new skill you're going to suck. It's going to be frustrating. The good news is that is typical and happens to everyone and it is only temporary. You can't go from knowing nothing to becoming an expert without going through a period of great frustration and great suckiness.

Kosuke Imai

One can learn data analysis only by doing, not by reading.



El mundo tidyverse

¿Qué es tidyverse?



- Conjunto de paquetes creados por Hadley Wickham
- Permiten implementar el proceso de análisis de datos presentado antes de forma efectiva
- Basado en la lógica de *tidy data* o dato limpio
 - Fila: unidad observada
 - Columna: variable
 - Tabla: una por tipo de unidad de observación

Tidyverse: paquetes



- dplyr
 - modificar datos
- forcats
 - factores
- ggplot2
 - gráficos
- haven
 - importar
- lubridate
 - fechas
- magrittr
 - pipes
- purrr
 - funciones
- readr
 - importar tablas
- readxl
 - importar excel
- stringr
 - texto
- tibble
 - tablas
- tidyr
 - reestructurar datos

Clave de tidyverse

- Encadenar funciones con la *pipe*, tubería o pipa
- Hay dos tipos de pipa
 - *magrittr pipe*: %>%
 - *base pipe*: |> [creación reciente, [más info](#)]
- Se puede añadir rápidamente: Ctrl/Cmd+Shift+M
- Elegir el tipo de pipa
 - Tools > Global Options > Code > Editing > Native pipe operator

Uso de la pipa (1)

- Permite ejecutar código de izquierda a derecha
- Por defecto, las pipas hacen que el primer argumento en una función después de una pipa sea el objeto que queda a la izquierda de la pipa
- Por ejemplo: como obtener la media de los valores 1, 5 y 5 redondeados a 2 decimales?
 - `round(mean(c(1,5,10)), digits = 2)`
 - `mean(c(1,5,10)) |> round(digits = 2)`
- Permite ejecutar diferentes funciones de forma secuencial
 - No requiere insertar funciones dentro de otras funciones
 - Simplifica el código empleado

Uso de la pipa (2)

- Permite leer el código de forma más fácil
- Evita tener que guardar objetos para pasos intermedios
- En el ejemplo anterior se podría haber simplificado aún más:

```
c(1,5,10) |> mean() |> round(digits = 2)
```

- Definimos un vector, del que calculamos la media, y luego redondeamos el valor a 2 dígitos tras el decimal

Importar datos en R

- Hay tres paquetes de tidyverse

1. Readr

- Importar csv y tablas simples

2. Readxl

- Importar excel

3. Haven

- Importar de otros formatos de software estadístico
 - .sav – SPSS
 - .dta – STATA



Ejercicio

- Vamos a importar la base de datos del QoG en distintos formatos

- csv
- excel
- Stata

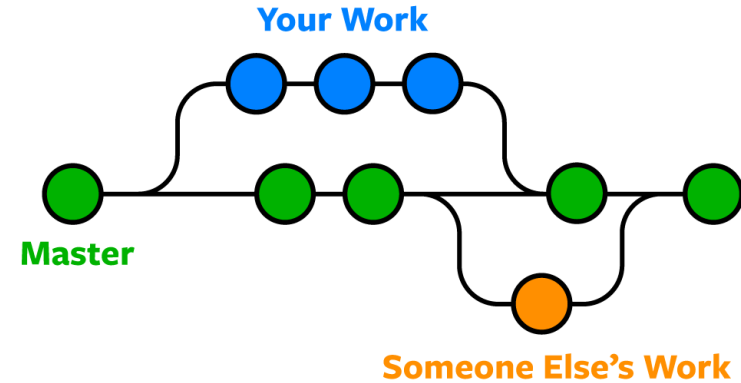
- Fuente

<https://www.qogdata.pol.gu.se/data/>

- Cread un objeto distinto en R para cada fichero
 - Recordad usar los paquetes adecuados

GitHub

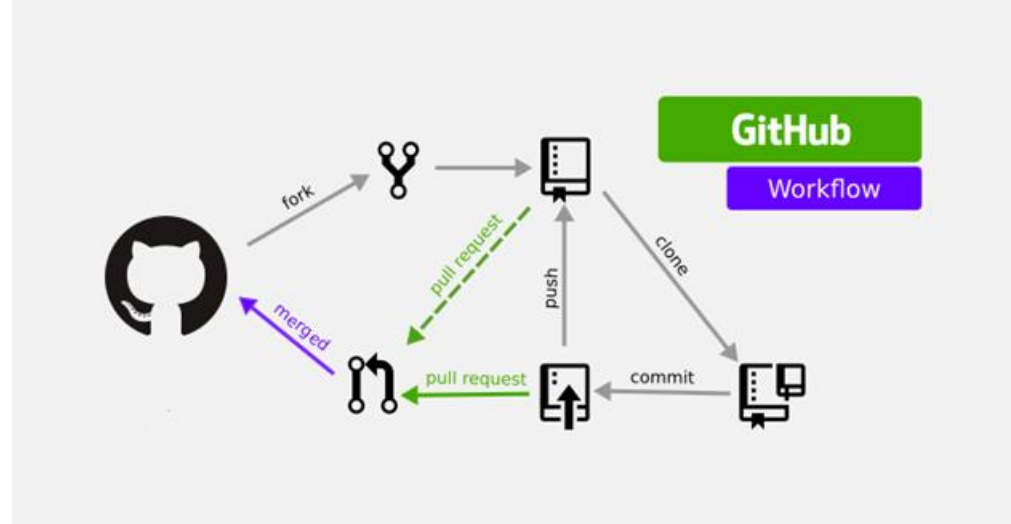
Github



- Es una de las formas de compartir código y proyectos más habitual
 - Git: Protocolo de control de versiones +
 - Hub: Centro
- Github es la plataforma para
 - Albergar
 - Compartir
 - Descargar
 - Trabajar con proyectos que siguen los protocolos git
- Creados para el trabajo colaborativo en entornos de software libre

<https://github.com/skills/introduction-to-github>


Diccionario Git










- Repositorio: conjunto de archivos (puedes enviar nuevos archivos o modificaciones de los mismos o descargarlos)
- Branch: rama del trabajo en la que los usuario está trabajando y que puede enviar al “master”
- Master: rama establecida del repositorio
- Commit: cambio propuesto para ser añadido a la rama master
- Fork: desviación de la rama principal
- Pull: enviar tu repositorio al servidor
- Push: descargar un repositorio del servidor


Ejemplo


https://github.com/pauvallprat/24-25_mape_talleres1

 **24-25_mape_talleres1** Public

 Pin  Unwatch **1**  Fork **0**  Star **0**


 main  1 Branch  Tags


 Add file


 Code


About


Asignatura Talleres I del Máster en Análisis Político y Electoral (UC3M) del curso 2024-2025

 Readme

 Activity

 0 stars

 1 watching











 0 forks



Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

 pauvallprat	structure of folder	5daa032 · 11 minutes ago	 3 Commits
	0_presentaciones	structure of folder	11 minutes ago
	1_datos	structure of folder	11 minutes ago
	2_ejercicios_clase	structure of folder	11 minutes ago
	3_entregas	structure of folder	11 minutes ago
	.Rhistory	Create .Rhistory	33 minutes ago
	.gitattributes	Initial commit	2 hours ago
	PD_2024_mape-talleres1.pdf	structure of folder	11 minutes ago
	README.md	Initial commit	2 hours ago

 **README** 

24-25_mape_talleres1

Bienvenidos a la página GitHub de la asignatura Talleres I del Máster en Análisis Político y Electoral (UC3M) del curso 2024-2025. En este entorno encontraréis toda la documentación relativa al curso que necesitaréis. Encontraréis carpetas con las presentaciones de clase, con datos con los que vamos a trabajar y carpetas para colgar los ejercicios de la asignatura.

¿Qué podemos hacer?

- Clonar el proyecto, descargar, modificar, etc.
- Crear un *fork* y mandar una actualización
 1. Editamos los documentos
 2. Explicamos los cambios hechos en el apartado de *commit*
 3. Enviar pull request
 4. Esperar que el propietario del proyecto acepte o no
 5. Si acepta el cambio, hay que actualizar la fork en nuestra cuenta

Ejercicio

- Cread vuestra propia cuenta de GitHub
- Descargaos GitHub Desktop
- Subid el script que habéis trabajado hoy en la carpeta 24-25_mape_talleres1\3_entregas\0_entrega_prueba
 - Recordad que deberéis
 1. Crear un fork en vuestra cuenta
 2. Subir el fichero
 - Importante que se titule:
apellido1_apellido2_nombre_scriptprueba.r
 3. Pull request

La semana que viene

Modificar bases de datos: repaso a dplyr y tidyr, stringr, summarise y el mundo markdown/Quarto