# Alumnos:

Stanciu Andy e Isabel Jiménez Rodríguez.

# Fecha entrega:

# Práctica 2 – GitHub y Ciencia Reproducible

Lee el siguiente artículo:

<https://www.revistaecosistemas.net/index.php/ecosistemas/article/viewFile/1178/973>

## 2.1. Responde a las siguientes preguntas:

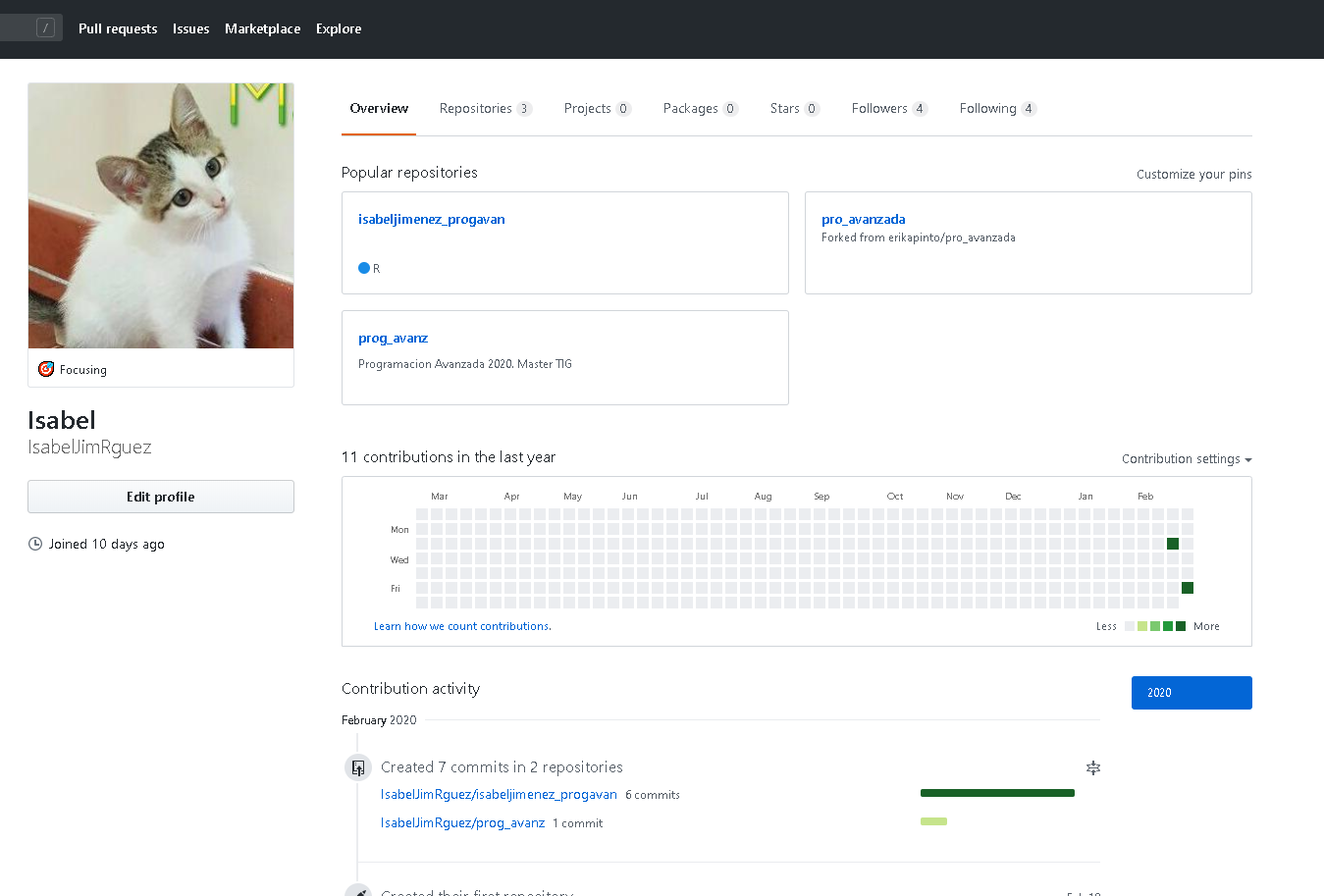
**¿Qué es la Ciencia Reproducible? ¿En qué casos se consigue un 100% de la reproducibilidad?**  
La Ciencia Reproducible la comprenden aquellos estudios acompañados de un código que permiten obtener los rsultados que se comentan en el propio artículo, permitiendo que los datos aoportados puedan ser comprobados y determinar la veracidad o no de los resultados, así como poder revisar la metodlogía de una forma más certera.  
El 100% de reproducibilidad se da en trabajos que integren códigos y datos que permitan obtener los mismo resultados que se han obtenido en el trabajo en si, Rodriguez-Sanchez, Pérez-Luque, Bartomeus, & Varela (2016)

**¿Cuáles de los beneficios que se sugieren son para ti los más importantes? Justifica tu respuesta**  
El principal beneficio que aporta la Ciencia Reproducible es el hecho de saber exactamente como se ha ejecutado la metodología de un trabajo. Como bien se menciona en el artículo, muchas veces se pueden leer metodologías muy interesantes que pueden solucionar un problema en cuestión, pero qué no se pueden reproducir porque con el aparatado de metodología no queda sufucientemente claro que pasos han seguido. No obstante, si se aporta el código puede ser reproducido con datos diferentes e incluso adaptarlos a nuevas necesidades, permitiendo un ahorro y eficiencia del trabajo y tiempo. Otra gran ventaja que tiene la Ciencia reporducible es que se puede tratar bases de datos, explotarlas y generar figuras con esos resultados, a la vez que explicarlos mediante texto solamente usando un solo programas, facilitando y economizando la creación de conocimiento.

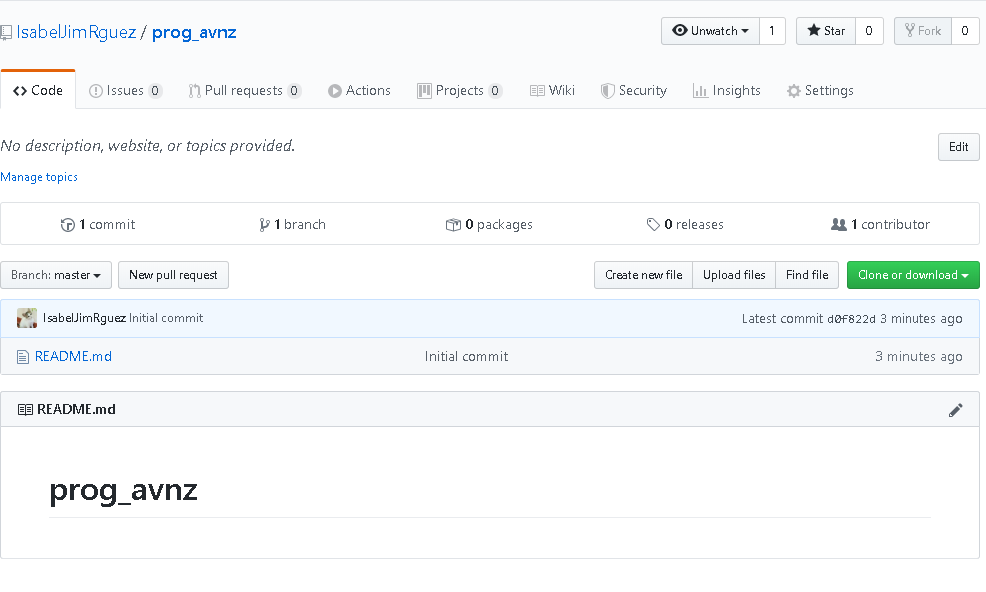
**¿Por qué no hace todo el mundo Ciencia Reproducible?**

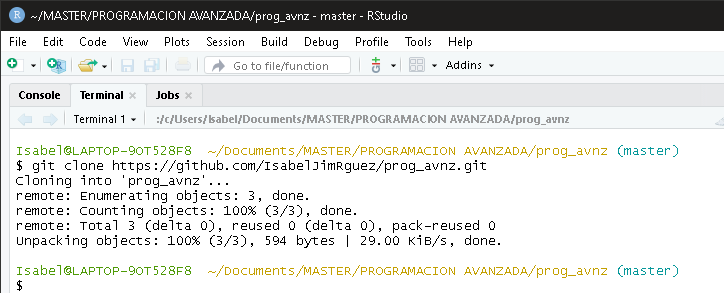
La Ciencia Reproducible a pesar de aportar muchas ventajas requiere de una formación mínima en programación, manejo de base de datos y sistemas de control de versiones, Rodriguez-Sanchez et al. (2016), lo que requiere de un tiempo y esfuerzo que del que no muchos están dispuestos a dar.

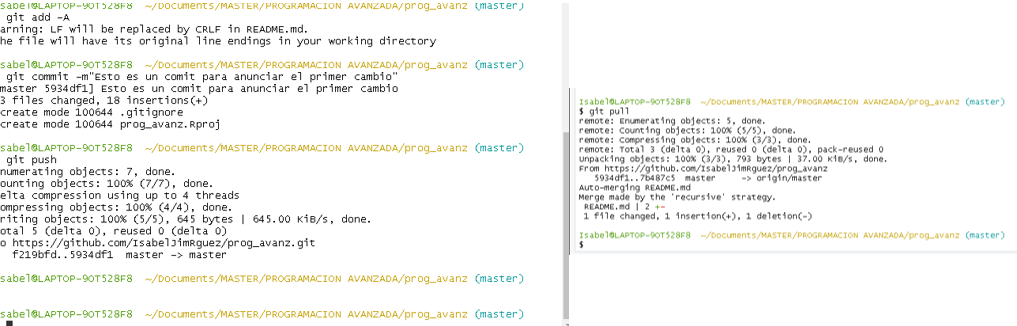
## 2.2. Lee el artículo “¿Por qué usar GitHub? Diez pasos para disfrutar de GitHub y no morir en el intento” Prieto (2018) Realiza las siguientes tareas:

* 1. Crea una cuenta en GitHub  
     
  2. Instala Git: <https://happygitwithr.com/install-git.html>  
     
  3. Preséntate: <https://happygitwithr.com/hello-git.html>  
     ###

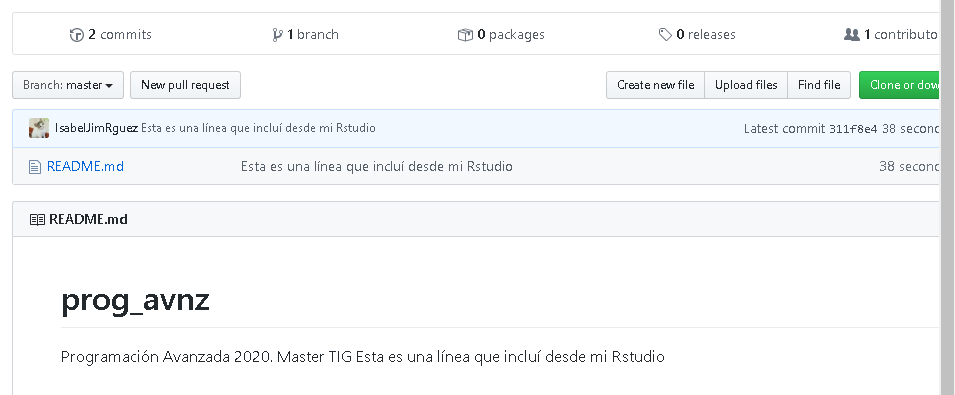
# Primero debes instalar el paquete "usethis"hh  
# install.packages("usethis")  
# Se debe invocar el paquete  
# Seguidamente usar la funcion use\_git\_config.  
# el user name puede ser disntinto al de github, per  
# pero el correo debe ser el mismo.   
  
library(usethis)  
use\_git\_config(user.name= "IsabelJimRguez",  
 user.email="isabel.jimenezr@edu.uah.es")

* 1. Crea un repositorio en GitHub, haz un clon del repositorio y confirma que puedes push/pull desde la línea de comandos: <https://happygitwithr.com/push-pull-github.html#push-pull-github>  
     





* 1. Conecta Rstudio a Git y Github: con el repositorio de GitHub clonalo a tu ordenador usando Rstudio.  
     <https://happygitwithr.com/rstudio-git-github.html#rstudio-git-github>.
  2. Introducir en el archivo README.md “Esta es una línea que incluí desde mi Rstudio” y comete los cambios (commit), posteriormente “push” a GitHub y comprueba que los cambios se realizaron correctamente en la nube.
  3. Crea un nuevo archivo para las clases del bloque 3 y sálvalo en GitHub. ¡Ahora puedes tener todos tus códigos en la nube y trabajar fácilmente en grupo!



## 2.3. Lee “¿Por qué usar GitHub? Diez pasos para disfrutar de GitHub y no morir en el intento”. Comprueba que tienes “claro” los siguientes conceptos:

**Repositorio** Los repositorios son los directorios dentro de la nube de GitHub donde se guardan los documentos que son subidos a GitHub. Se puede tener más de un repositorio, por lo que se puede trabajar en varios proyectos.

**Rama** Las ramas son versiones de un mismo script pero que no modifican al scripot principal, permitiendo hacer pruebas o versiones sobre una mismo script base. Las ramas pueden volver a unirse a la rama principal mediante el comando “compare and pull request”.

**Pull and push** Los comandos pull y push permiten descargar en el Git de escritorio un script que está en el repositorio de la nube de GitHub, mientras que push permite enviaelo.

**Fork** Este comando permite descargar repositorios ajenos desde Github a nuestro propio perfil de GitHub, los cuales pueden ser modificados.

## 2.4. RMarkdown. Genera un documento de html que contenga

* 1. Todas las opciones de sintaxis básicas de Pandoc Markdown: texto plano, cursiva, negrita, superíndice, subíndice, tachado, link, ecuación en línea de texto, ecuación en bloque, diferentes encabezados.
  + texto plano
  + *cursiva*
  + **negrita**
  + superíndice2
  + subíndice2
  + ~~tachado~~
  + [Si se clickea sobre esta frase se abrirá una pestaña en Google](https://www.google.com/webhp?hl=es&sa=X&ved=0ahUKEwjow-rs-PnnAhUF3hoKHbWKC8oQPAgH)
  + Ecuación en línea de texto:
  + Ecuación en bloque :

# Encabezado de título principal

## Encabezado del título de las preguntas de este template.

* 1. Escribe fragmentos de código con las siguientes características
  2. Peguen las palabas “Hola” y “mundo” en una línea nueva

paste("Hola", "mundo")

## [1] "Hola mundo"

* 1. Peguen las palabas “Hola” y “mundo” en la misma línea de código
  2. Peguen las palabas “Hola” y “mundo” en una línea nueva de código y que no muestren el resultado de R

## [1] "Holamundo"

* 1. Peguen las palabas “Hola” y “mundo” en una línea nueva de código y que no muestren el resultado de R y que no se muestre en el documento de Word.

## 3.Muestra una tabla y una gráfica en el archivo de salida, tomando alguno de los datos que vienen en R o alguno de sus paquetes. ¡¡Se creativo!! Usando (al menos): encabezados, explicaciones, ecuaciones, líneas de código en el texto y bloques de código.

# CREACIÓN DE TABLAS

## Tabla con el paquete flextable

# Se creará una tabla con los tamaños medios de los sépalos y pétalos en función de especies  
# datos iris  
  
library(flextable) # Necesario para generar la tabla   
library(plyr)  
  
#se preparan los datos para su posterior explotación   
datos.tabla <- ddply(iris, .(Species), summarize,  
 "Anchura Pétalo"= mean(Petal.Width),  
 "Largo Pétalo"= mean(Petal.Length),  
 "Anchura Sépalo"= mean(Sepal.Width),  
 "Largo Sépalo" = mean(Sepal.Length))  
  
## Plot de la tabla  
  
#Genera la tabla con los datos del data.frame creado  
tabla <- flextable(datos.tabla)  
  
#Pone en negrita el nombre de las columnas  
tabla <- bold(tabla, part = "header")   
  
#Añade encabezados  
tabla <- add\_header\_lines(tabla, values = c("Dimensiones medias del sépalo  
 y pétalo por especies", "BBDD Iris (R)"))   
#Estilo que se quiere aplicar a la tabla  
tabla <- theme\_box(tabla)  
  
#Muestra la tabla  
tabla

| **BBDD Iris (R)** | | | | |
| --- | --- | --- | --- | --- |
| **Dimensiones medias del sépalo  y pétalo por especies** | | | | |
| **Species** | **Anchura Pétalo** | **Largo Pétalo** | **Anchura Sépalo** | **Largo Sépalo** |
| setosa | 0.246 | 1.462 | 3.428 | 5.006 |
| versicolor | 1.326 | 4.260 | 2.770 | 5.936 |
| virginica | 2.026 | 5.552 | 2.974 | 6.588 |

## Creación de tablas de manera manual

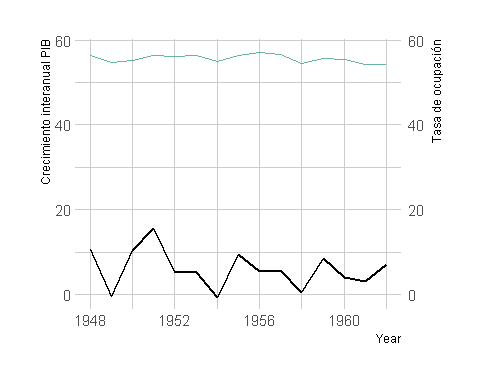
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Especies | Anchura del pétalo | Largo del pétalo | Anchura del sépalo | Largo del sépalo |
| setosa | 0.246 | 1.462 | 3.428 | 5.006 |
| versicolor | 1.326 | 4.26 | 2.77 | 5.936 |
| virginica | 2.026 | 5.552 | 2.974 | 6.588 |

# CREACIÓN DE GRÁFICAS

* En este caso emplearemos el dataset incluiodo por defecto en r llamado “longley”. Este data set aporta datos macroeconomicos desde el año 1947 hasta 1962 de un país en regresión. Entre sus columnas dispone de datos acerca del producto interior bruto (PIB), el PIB con base 1954, datos brutos de desempleo y empleo, datos brutos de personas inscritas a las fuerzas armadas y por último, el total de población del país en edad de trabajo.

Para este caso querremos calcular la tasa de ocupación y la tasa de crecimiento interanual del PIB.

library(ggplot2)  
library(dplyr)  
library(patchwork)  
library(hrbrthemes)  
  
datos <- longley # Toma los datos del dataset de R   
  
# se añade y calcula la tasa de ocupacion al dataframe  
datos$Tasa.ocupacion <- (datos$Employed/datos$Population)\*100  
  
#Para poder aplicar la formula de variacion del PIB  
#Se necesita obtner una columna con el PIB referido al año anterio  
  
for (i in 1:nrow(datos)){  
 pib.anterior.15<- datos[1:15,2] #selecciona los datos del pib del año anterior  
 # y los coloca en la fila del año de interes  
}  
  
#Se crea un valor 0 para unirlo al anterior vector y que den   
# 16 observaciones y asi poder unirlo como una nueva columna del dataframe  
  
base <- c(0)  
pib.anterior.16 <- c(base, pib.anterior.15)  
  
#Añade nueva columna al dataframe  
datos$pib.anterior <- pib.anterior.16  
  
#Se hace un bucle que permita aplicar la fórmula del crecimiento del pib a todo las filas  
for(i in 1:nrow(datos)){  
 datos$Crecimiento.pib <- ((datos$GNP-datos$pib.anterior)/datos$pib.anterior)\*100  
}  
  
datos.2 <- datos[-1,]  
#plot  
x <- 1  
ggplot(datos.2, aes(x=Year, name= "año")) +   
   
 geom\_line( aes(y=Tasa.ocupacion), stat = "identity", size=.1, color= "#69b3a2") +  
 geom\_line( aes(y=Crecimiento.pib), size=1) +  
   
 scale\_y\_continuous(  
 name= "Crecimiento interanual PIB",   
 sec.axis = sec\_axis(~.x, name="Tasa de ocupación")  
 ) +   
   
 theme\_ipsum()



# Bibliografía

Prieto, J. G. (2018). ?‘ Por qué usar github? Diez pasos para disfrutar de github y no morir en el intento. *Revista Ecosistemas*, *27*, 140–141.

Rodriguez-Sanchez, F., Pérez-Luque, A. J., Bartomeus, I., & Varela, S. (2016). Ciencia reproducible: Qué, por qué, cómo. *Revista Ecosistemas*, *25*, 83–92.