

Introducción al manejo y análisis de datos con R

Unidad 1

- Software libre y de código abierto vs. Software privativo



Documental sobre GNU-Linux

https://www.youtube.com/watch?v=9ip3UA_04LM



Libre software (free software)

- Busca asegurar que los usuarios tengan la libertad de usar el software.

Libertad 0: Libertad de correr el programa como quieras y con cualquier propósito

Libertad 1: Libertad de estudiar como funciona el programa y modificarlo con el fin de Usarlo como quieras. Esto implica la pre-condición de acceso directo al código

Libertad 2: Libertad de redistribuir copias con el fin de ayudar a otros.

Libertad 3: Libertad de distribuir copias de las versiones que modificaste a otros. De Vuelta, esto implica la pre-condición de tener acceso libre al código fuente.

Copyleft: Regla que obliga a si redistribuyes un programa no puedes negar las 4 libertades del software libre a los usuarios

<https://www.gnu.org/philosophy/free-software-even-more-important.html>





GNU/LINUX

<https://www.youtube.com/watch?v=jUibaPTXSHk>



GNU/LINUX

<https://www.youtube.com/watch?v=jUibaPTXSHk>



Richard Stallman

Software abierto (open source software, OSS)



- Es parecido a libre software, pero difiere en la ideología. Permite inspeccionar, modificar y mejorar el código fuente, pero debe cumplir ciertos criterios.
- Trata de resolver la confusión de free software conceptualizado como software gratis y no como software libre.

...

Criterio 5: Licencias que no discriminación contra personas o grupos

Criterio 6: Licencias que no discriminación contra campos de conocimiento o empresas

Criterio 8: La licencias no debe ser específico de un producto. La licencia es independiente de en que otro producto se use el programa.

...

Software privativo



- Propiedad de particular asociado a restricciones de uso y su código fuente en general esta oculto.

El software privativo pone en posición de poder al propietario de la licencia
Del software a los usuarios y a la comunidad

Este software, en general, trata mal al usuario (malware)



Ejemplos de software



Software privativo, o con restricciones	Libre software
Adobe ilustrator	Inkscape
Explorer, Chrome	Firefox, Chromium
Unix	Linux + GNU ("GNU is not Unix")
Wix	Wordpress
Photoshop	Gimp
???	ImageJ, Fiji
Windows	Ubuntu, Debian
GIS, Arcview	QGIS, Grass, UDig
Matlab, Mathematica	Python, Pearl
SAS, Stata, Jump, Systat, SPSS, etc	R

Pros y contras

Concepto	Software privativo, o con restricciones	Libre software
Costo	Caro	Gratis o barato
Soporte	Restringido al desarrollador + comunidad	Desarrollador + comunidad
Transferencias a otras áreas	Complicado	Fácil y rápida
Habilidad para adaptarse	Lenta y dependiente del mercado	Rápida
Corrección de errores	Lenta	Rápida
Aplicaciones	Se venden por separado	En paquetes descargables
Mantenimiento	Caro	Gratis o barato
Seguridad	Depende del vendedor y usuario	Depende de usuario
Amigabilidad	Suele ser alta	Baja, media, alta

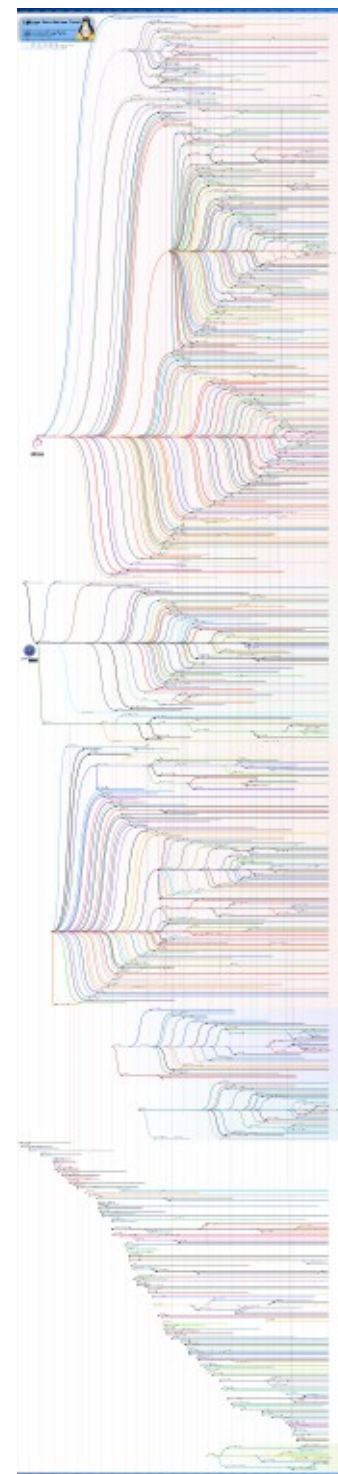
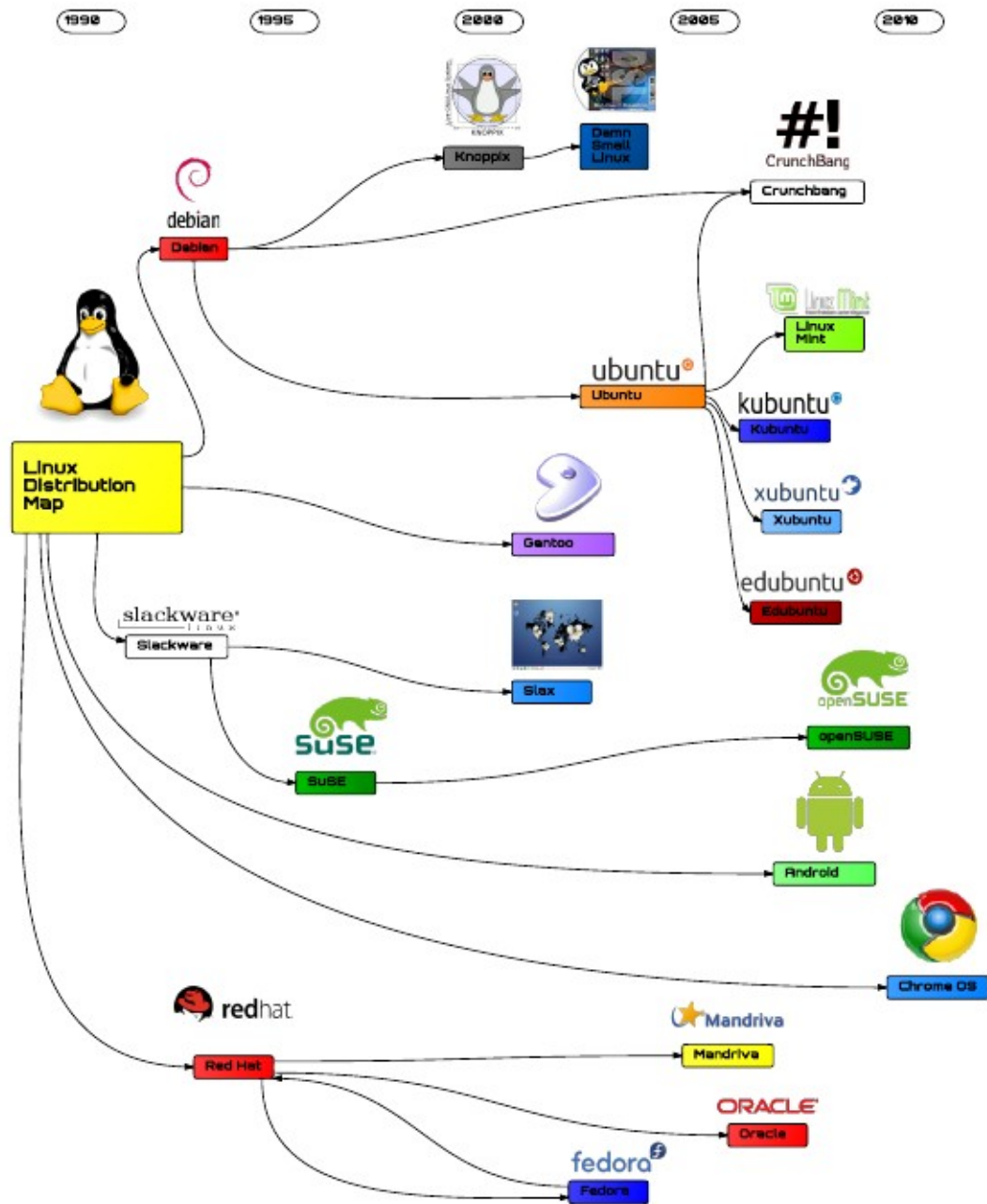
Ejemplos interesantes de free software

LuMux: Gobierno de Munich (2005-2013), ahorro 11.7 millones de euros.

Kylin: China busca incrementar acceso y alta seguridad en gobierno y sistemas Militares.

GendBuntu: Francia (2017) Gendarmería migra a Linux con ahorro de 2 millones de Euros.

Canaima: Venezuela (2016). En búsqueda de usar tecnologías libres en Administración publica 2.5 millones de computadoras a estudiantes



- ¿Te acuerdas cuando **PIRATEÁBAMOS SPSS, MINITAB, STATA, SAS, MATLAB...** para nuestros análisis de datos?
- Pero finalmente estamos usando R o Python





Generales de R



- Desarrollado por **R**obert Gentleman y **R**oss Ihaka en la Universidad de Auckland en Nueva Zelanda.
- Basado en el lenguaje de programación S desarrollado por Bell Company



Robert Gentleman



Ross Ihaka



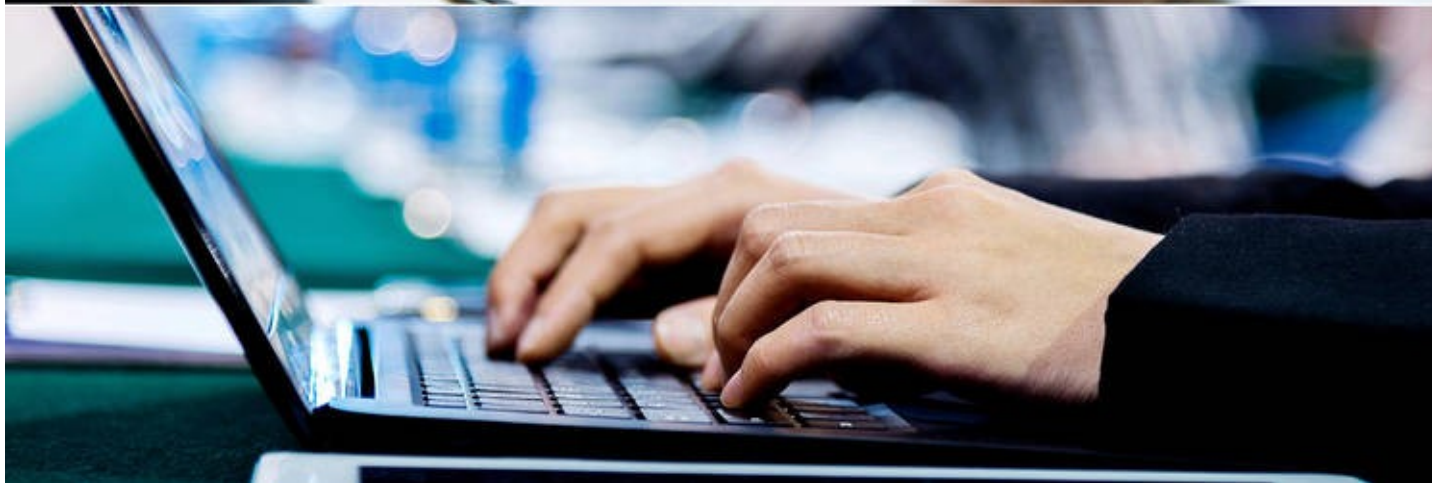
Disponibilidad de información libre y soporte:

- Libros gratis: ej. R for data Science
- Páginas de apoyo: Quick-R by DataCamp, <https://bit.ly/2QcoEgB>
- Comunidades: StackOverflow
- Bloggs: r-bloggers.com, Rweekly.org
- Cursos en línea: DataCamp, udemy.com

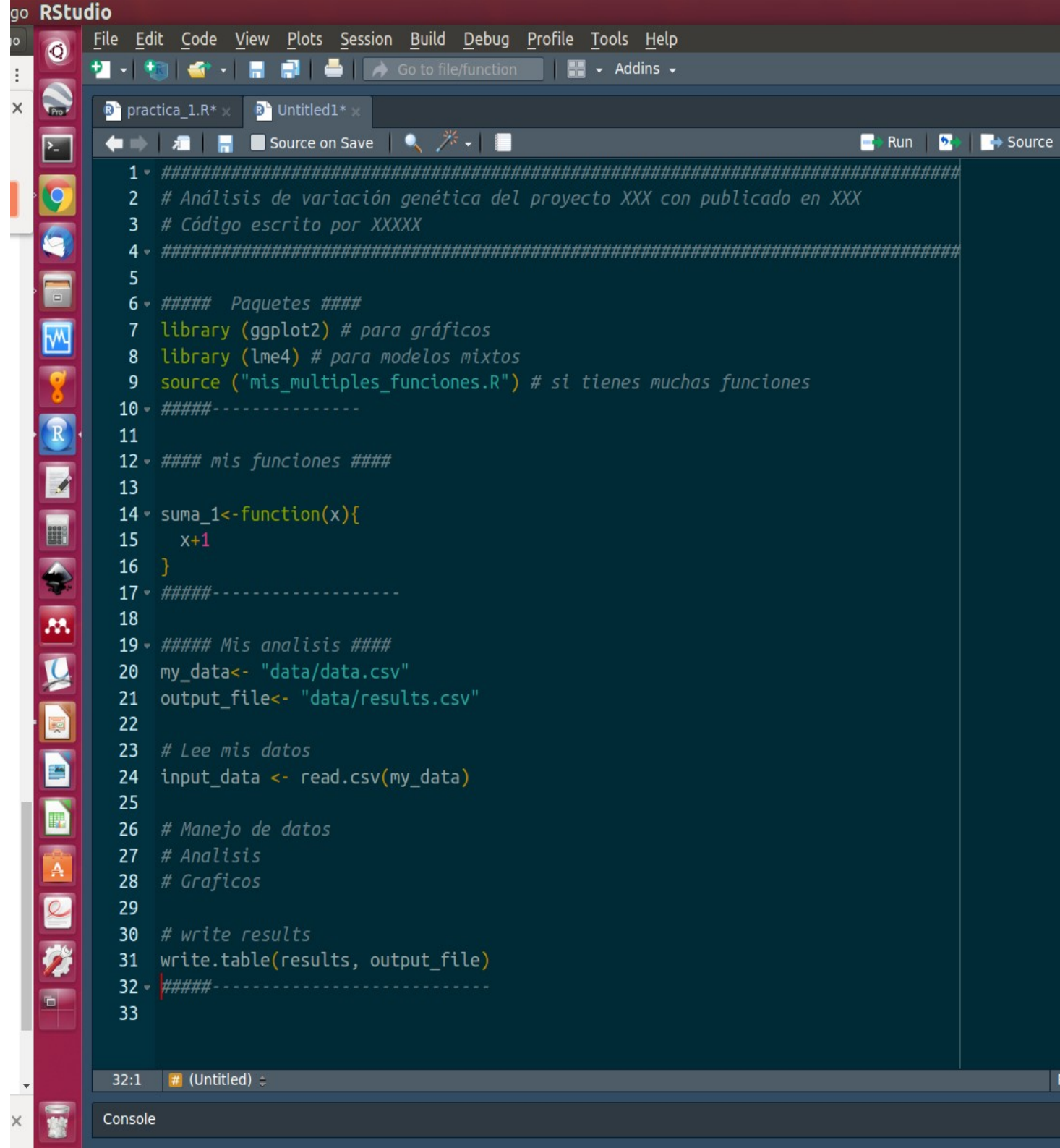
Herramientas para R:

- Para manejo de código: Rstudio, Jupyter notebooks, Sublime
- Para reproducibilidad: Github
- Para comunicar resultados: Markdown, Shiny

¿Cómo tomar apuntes?



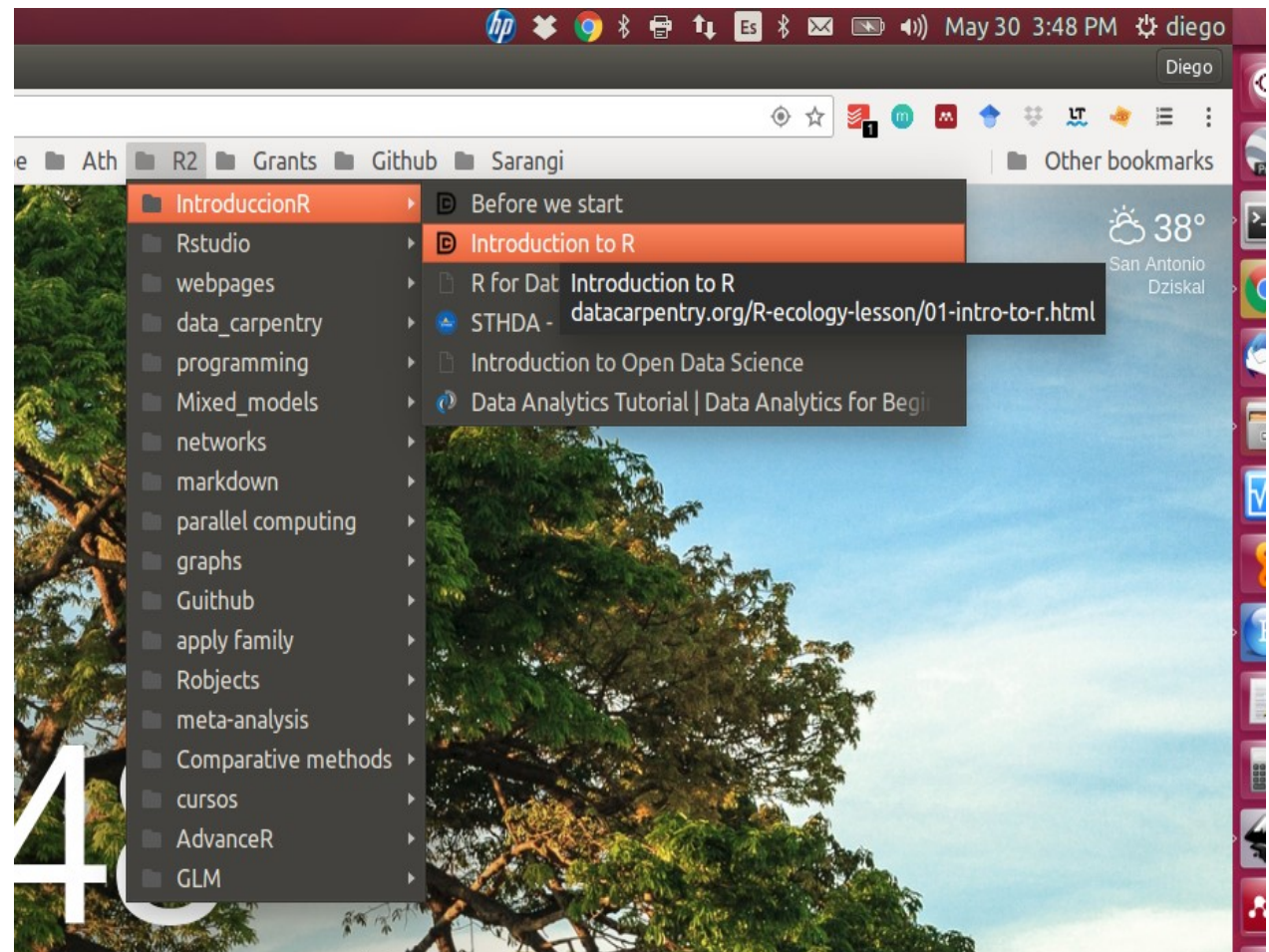
Estructura de trabajo con código



```
1 #####
2 # Análisis de variación genética del proyecto XXX con publicado en XXX
3 # Código escrito por XXXXX
4 #####
5
6 ##### Paquetes ####
7 library (ggplot2) # para gráficos
8 library (lme4) # para modelos mixtos
9 source ("mis_multiples_funciones.R") # si tienes muchas funciones
10 #####-----
11
12 ##### mis funciones ####
13
14 suma_1<-function(x){
15   x+1
16 }
17 #####-----
18
19 ##### Mis analisis ####
20 my_data<- "data/data.csv"
21 output_file<- "data/results.csv"
22
23 # Lee mis datos
24 input_data <- read.csv(my_data)
25
26 # Manejo de datos
27 # Analisis
28 # Graficos
29
30 # write results
31 write.table(results, output_file)
32 #####-----
33
```

Crea un bookmark exclusivo para salvar recursos y crea subcarpetas:

- Básico
- Gráficos
- Programación
- Estadística
 - Modelos lineales
 - Modelos mixtos
 - Meta-análisis
- Páginas y bloggs
- Rstudio
- Markdown
- Github



Actividad

- Instala R (<https://www.r-project.org/>)
- Instala Rstudio (<https://rstudio.com/products/rstudio/download/>)
- Abre Rstudio
- Abre archivo .Rproj. Esto activará Rstudio
- Dentro del navegador de Rstudio ve a la carpeta src y abre el archivo llamado Practico_1_lectura_base_datos.R
- Corre el código.
- Crea una base en excel, salvala como .csv en el directorio de bases de datos y trata cargar la base en R usando el código que usaste con la base de datos llamada iris.csv