

Introducción a R



COMPASS

Community Platform for Agricultural Sciences

Paulo Izquierdo
Viviana Ortiz



Paulo



Viviana

Quienes somos?

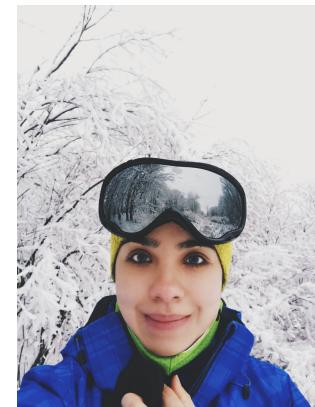
Viviana Ortiz PhDc

Michigan State University



- ortizviv@msu.edu | @VivianaOrtizL
- Intereses de investigación: Genética de poblaciones, evolución y ecología en patógenos de plantas, monitoreo y predicción de enfermedades

An evolutionary genomics perspective of adaptation in plant pathogens



How plant pathogens adapt to plant hosts,
pesticides and environmental factors?



Genomic data



Demographic histories
Signatures of selection

Phenotypic data



Pathogenicity
Fungal growth rates
Fungicide susceptibility

Ecological data

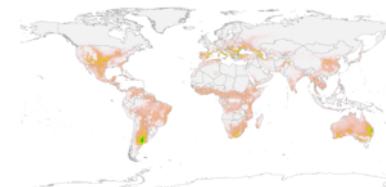


Environmental data
Geographic distributions

Adaptive potential and mechanisms of adaptation



Improve disease management
strategies



Improve disease prediction

Paulo Izquierdo PhDc

Michigan State University

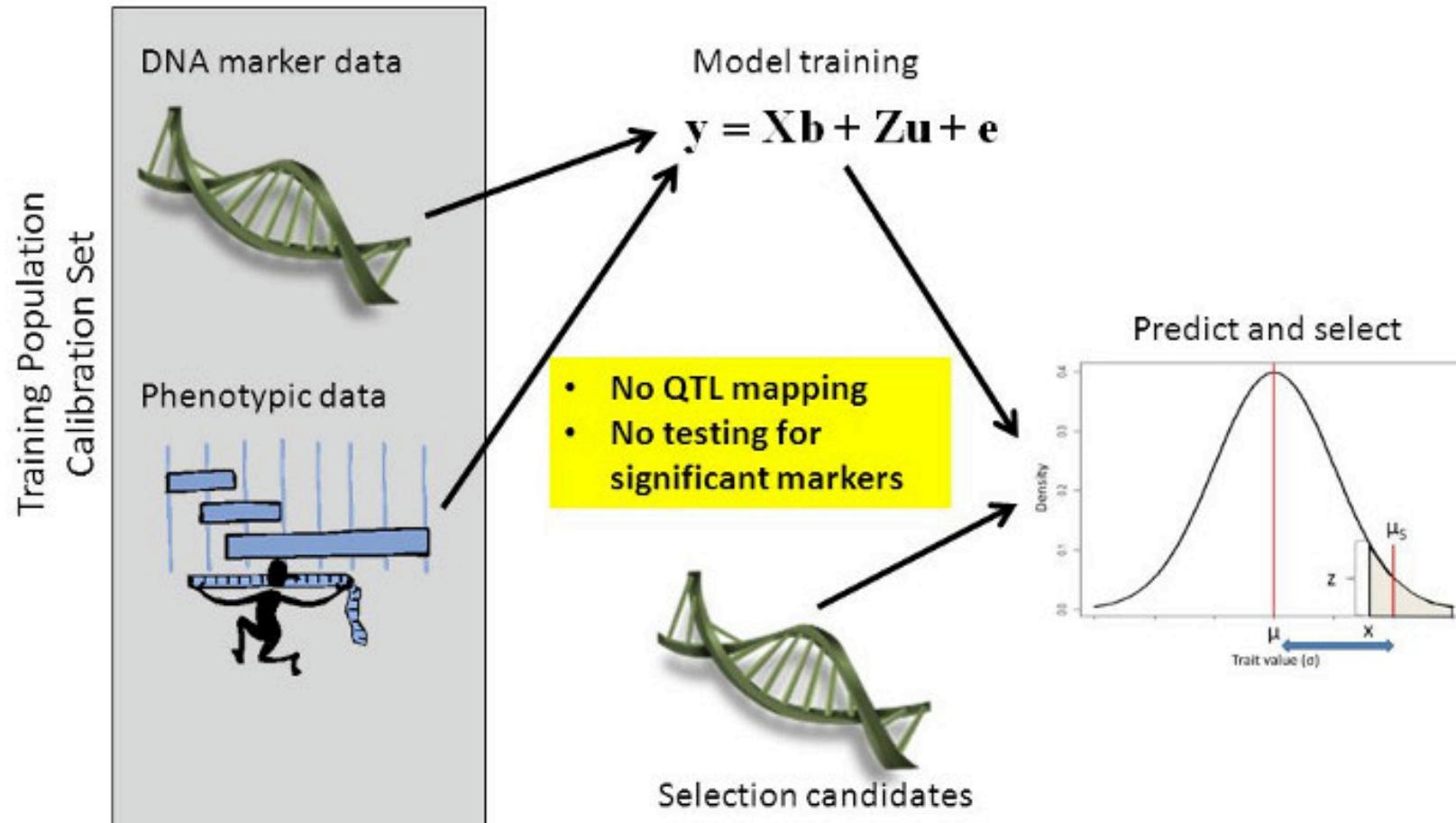


- izquier7@msu.edu | @paulocizquierdo
- Intereses de investigación: Genética cuantitativa, fitomejoramiento, predicción genómica y bioinformática.

Quienes somos?



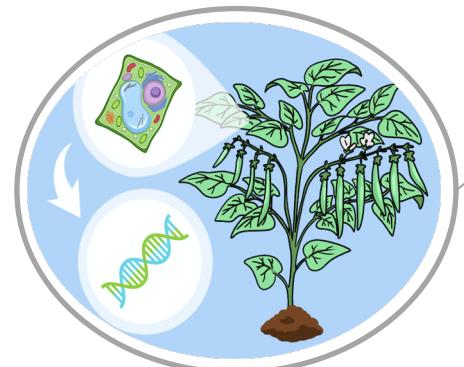
PhDc - MSU





¿Cuál es el objetivo de COMPASS?

Inspirar, compartir, transformar



COMPASS
@COMPASS_Col

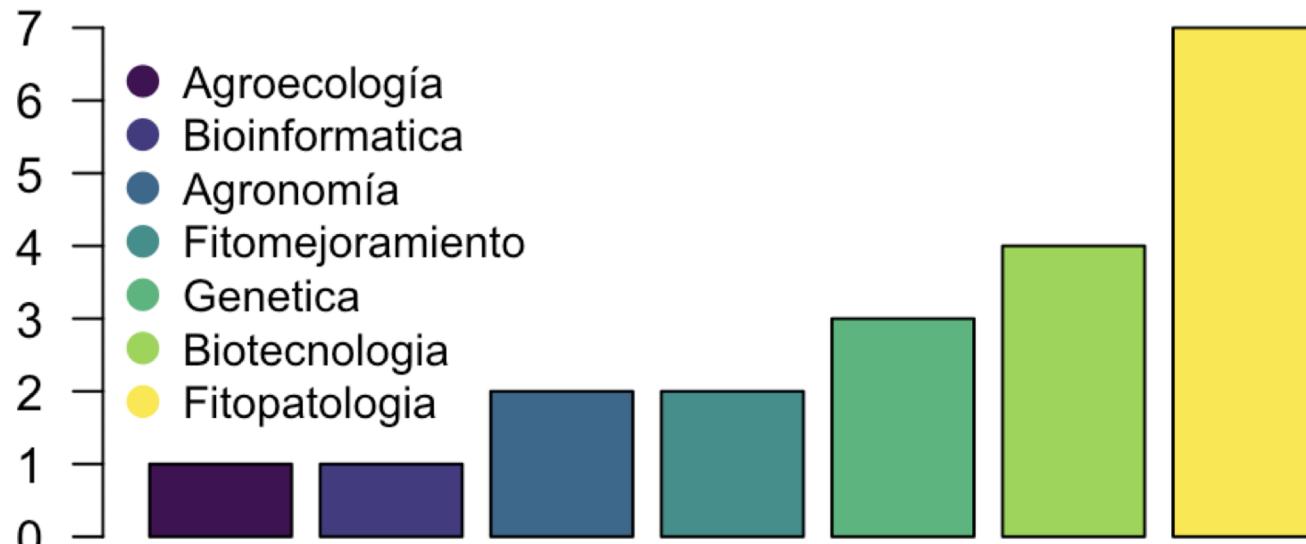
¿ Dónde están nuestros aliados?



> 40 aliados internacionales y nacionales en diferentes áreas

¿Cómo estamos logrando el objetivo mediante COMPASS?

Aliados Nacionales



¿Cómo estamos logrando el objetivo mediante COMPASS?

UNIVERSITÀ
DEGLI STUDI
DI UDINE

USDA
ARS



PURDUE
UNIVERSITY®

BTI
BOYCE
THOMPSON
INSTITUTE



Colorado State
University

WASHINGTON STATE
UNIVERSITY



MICHIGAN STATE
UNIVERSITY

The James
Hutton
Institute

CIP
INTERNATIONAL
POTATO CENTER

UNIVERSITY OF
BIRMINGHAM

McGILL
UNIVERSITY

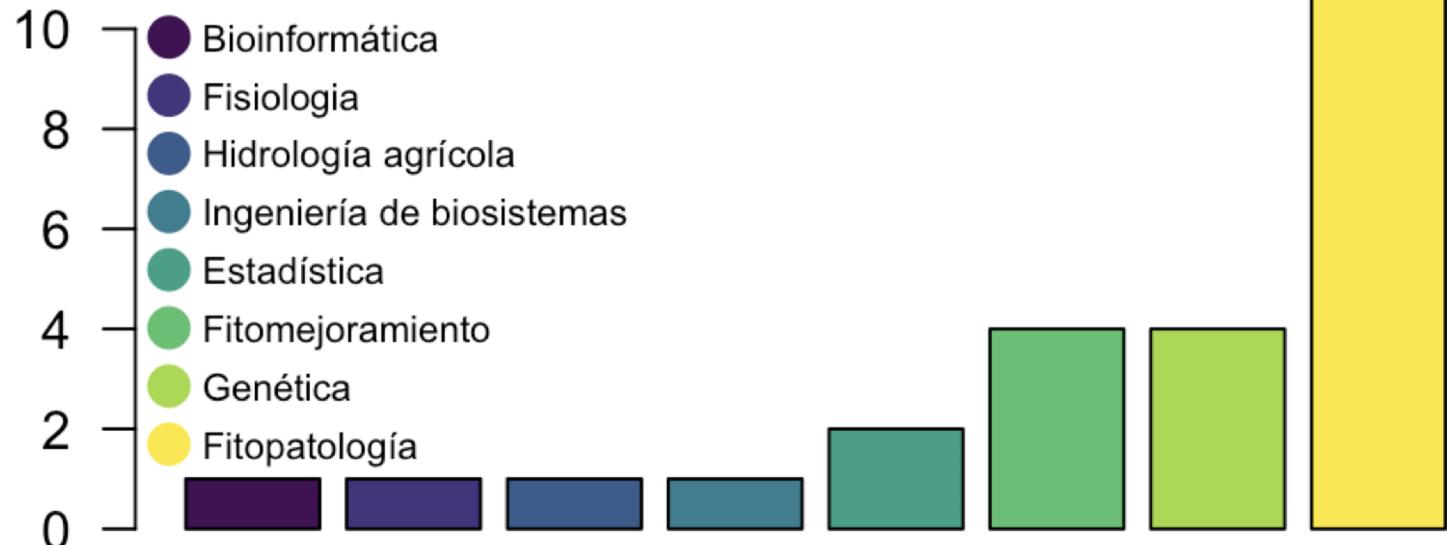
BAYER

UCDAVIS
UNIVERSITY OF CALIFORNIA

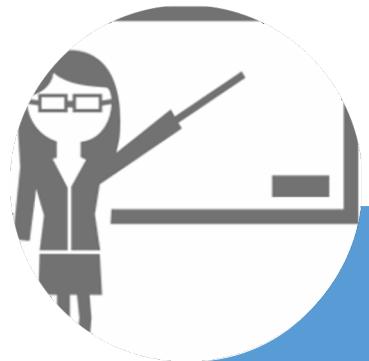
BASF
The Chemical Company

UNIVERSITY OF
ILLINOIS
URBANA-CHAMPAIGN

Aliados Internacionales



Extensión académica



- Educación técnica y superior
- Promoción de talleres y clases magistrales enfocadas a áreas de estudio en el campo de las ciencias agrícolas

Alianzas internacionales



- Creación de convenios con entidades académicas internacionales
- Promoción de oportunidades como becas, pasantías y ofertas laborales

Eventos científicos



- Intercambio y transferencia de conocimiento entre investigadores internacionales y colombianos con presencia en semilleros y grupos de investigación.
- Participación en eventos de divulgación científica

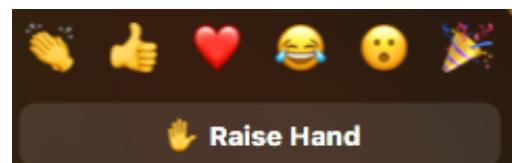
Introducción a R



Tengo una pregunta
o necesito ayuda!



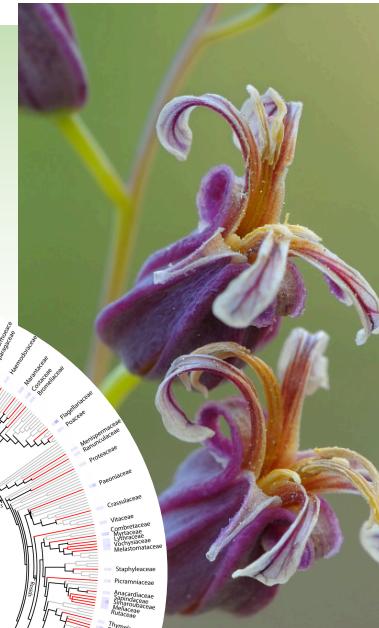
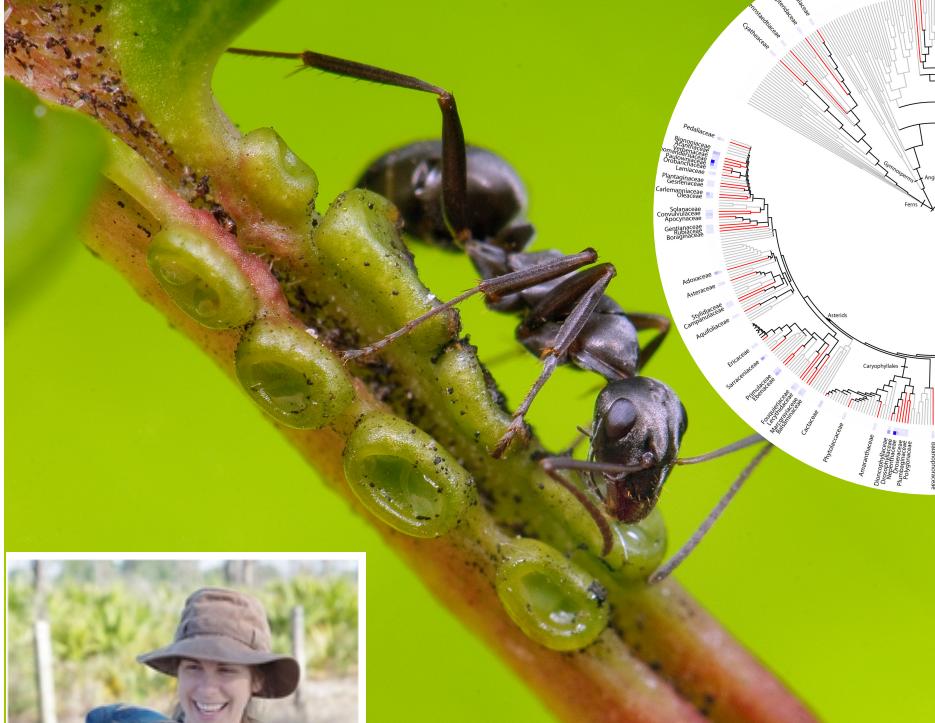
Estoy trabajando
en ello



Lo hice!



Dr. Marjorie Weber
Michigan State University
Assistant Professor of Plant Biology
<http://www.theweberlab.com/>



Join us!

The Weber Lab is always looking for motivated undergraduate students, graduate students, and postdocs to join the lab.



Ya tienen instalado R y RStudio?



Introducción a R

- Todos estamos en un nivel diferente!



Por que usar R?



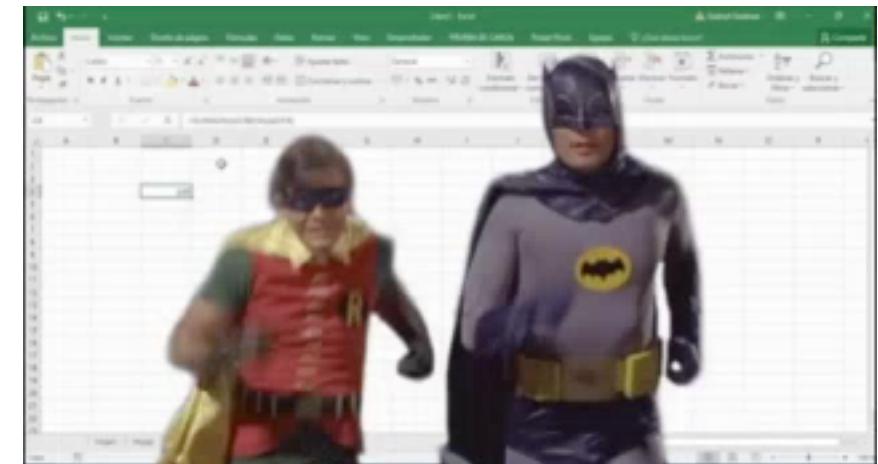
- Excel puede ser bueno!

- ✓ Excel es intuitivo
- ✓ La manipulación de datos es fácil
- ✓ Puede ver lo que está sucediendo
- ✓ Ingresar los datos



- Pero ... En otras cosas....

- Los gráficos son deficientes
- Paquetes estadísticos limitados
- Hay algunas cosas que Excel no puede hacer

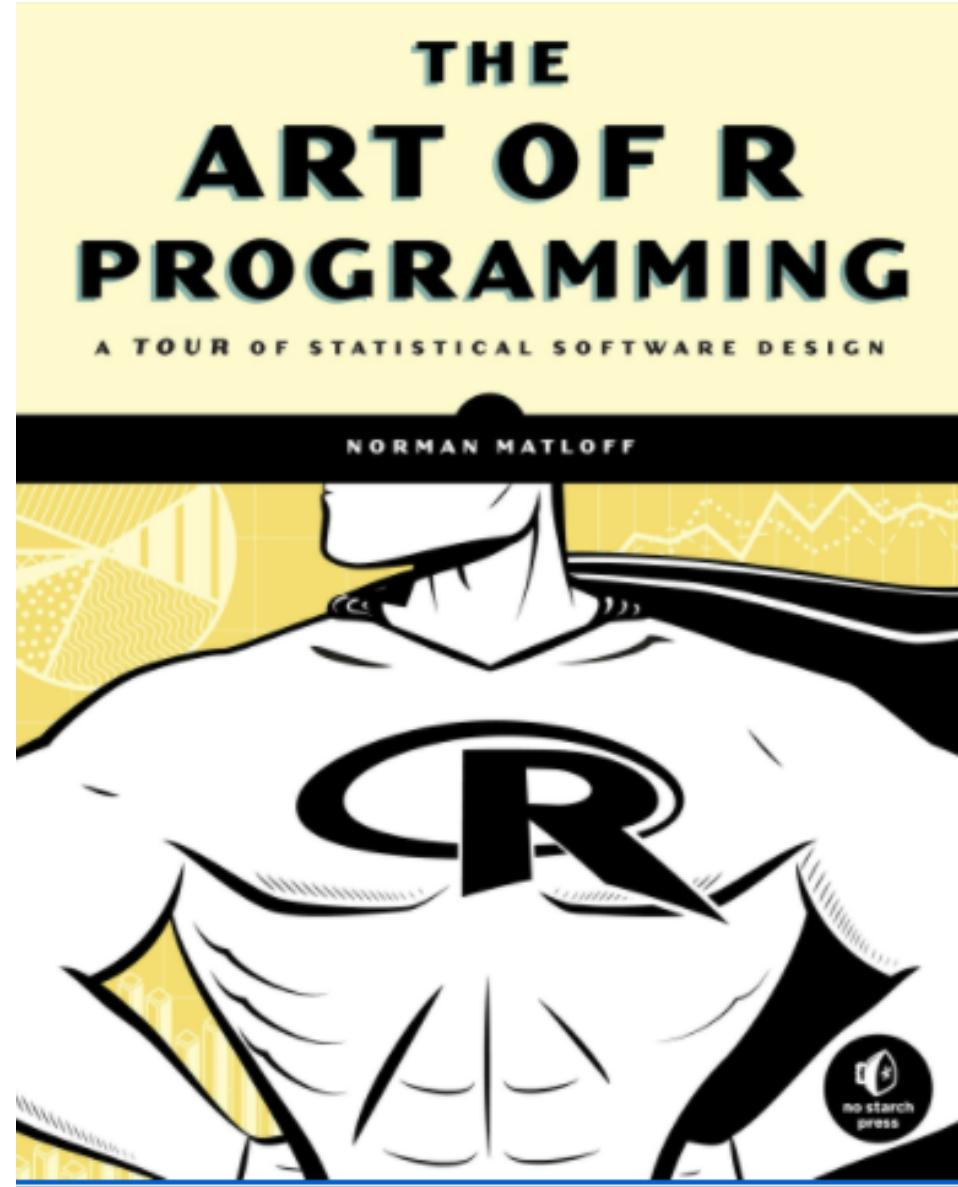


Excel no es bueno para...

- Manejar grandes bases de datos
- Análisis de datos
- Limitado para manipular y graficar datos



La solución : R

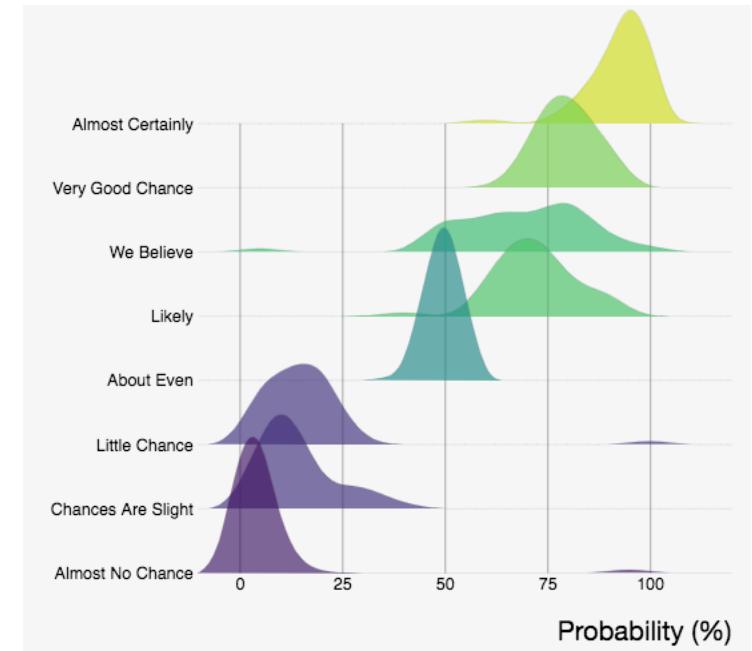


Que es R?

- R es un lenguaje y entorno informático para gráficos y análisis estadísticos.
- Está basado en script (código)

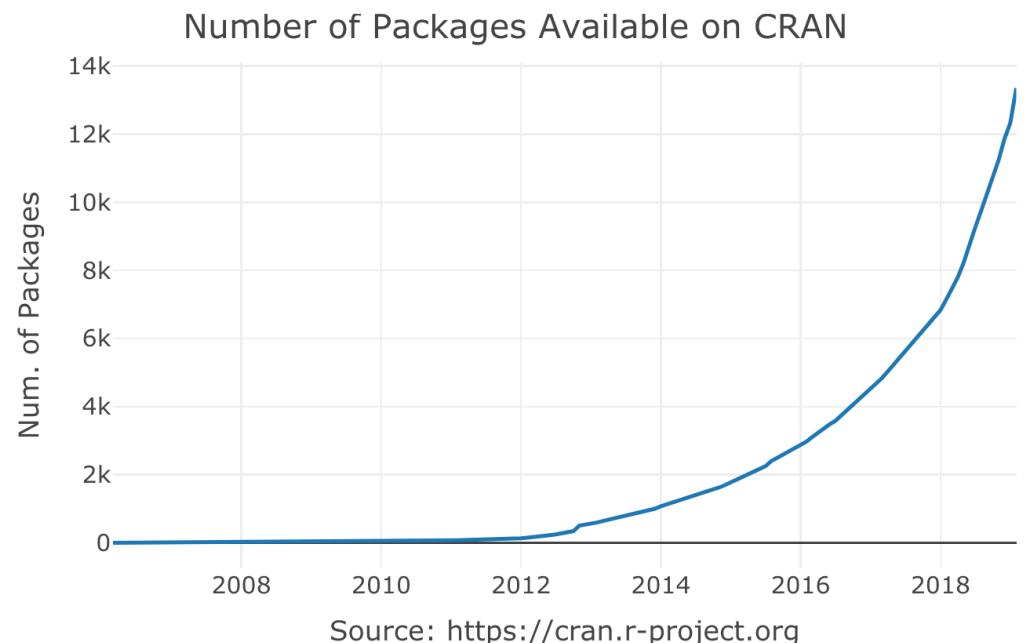
Fortalezas de R

- Completamente de código abierto
 - Los usuarios contribuyen y crean nuevas funciones
 - Las funciones R existentes se pueden editar y expandir - Gratis
 - Gran comunidad de científicos que utilizan R
- Gráficos con calidad de publicación:
 - Muchos gráficos predeterminados
 - Control total de gráficos

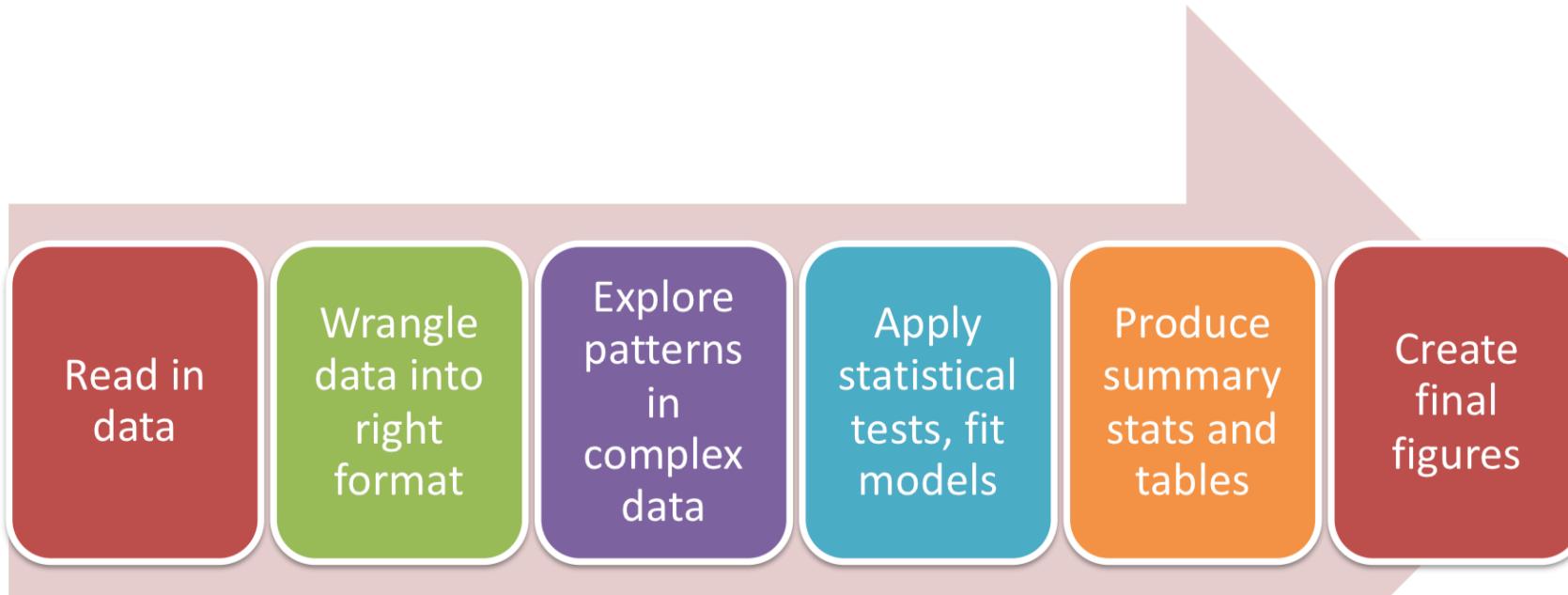


Fortalezas de R

- Prácticamente todas las técnicas estadísticas comunes están disponible como paquete gratuito
 - Modelos lineales
 - Clasificación y análisis de conglomerados
 - Análisis espaciales
 - Estadísticas bayesianas
 - ...



Fortalezas de R



ALL IN A SINGLE R SCRIPT!

Reproducibilidad!

- R se considera uno de los principales herramientas para análisis de datos
- El uso de R está aumentando exponencialmente
- Un número creciente de trabajos requiere experiencia con R
- Muchas empresas utilizan R



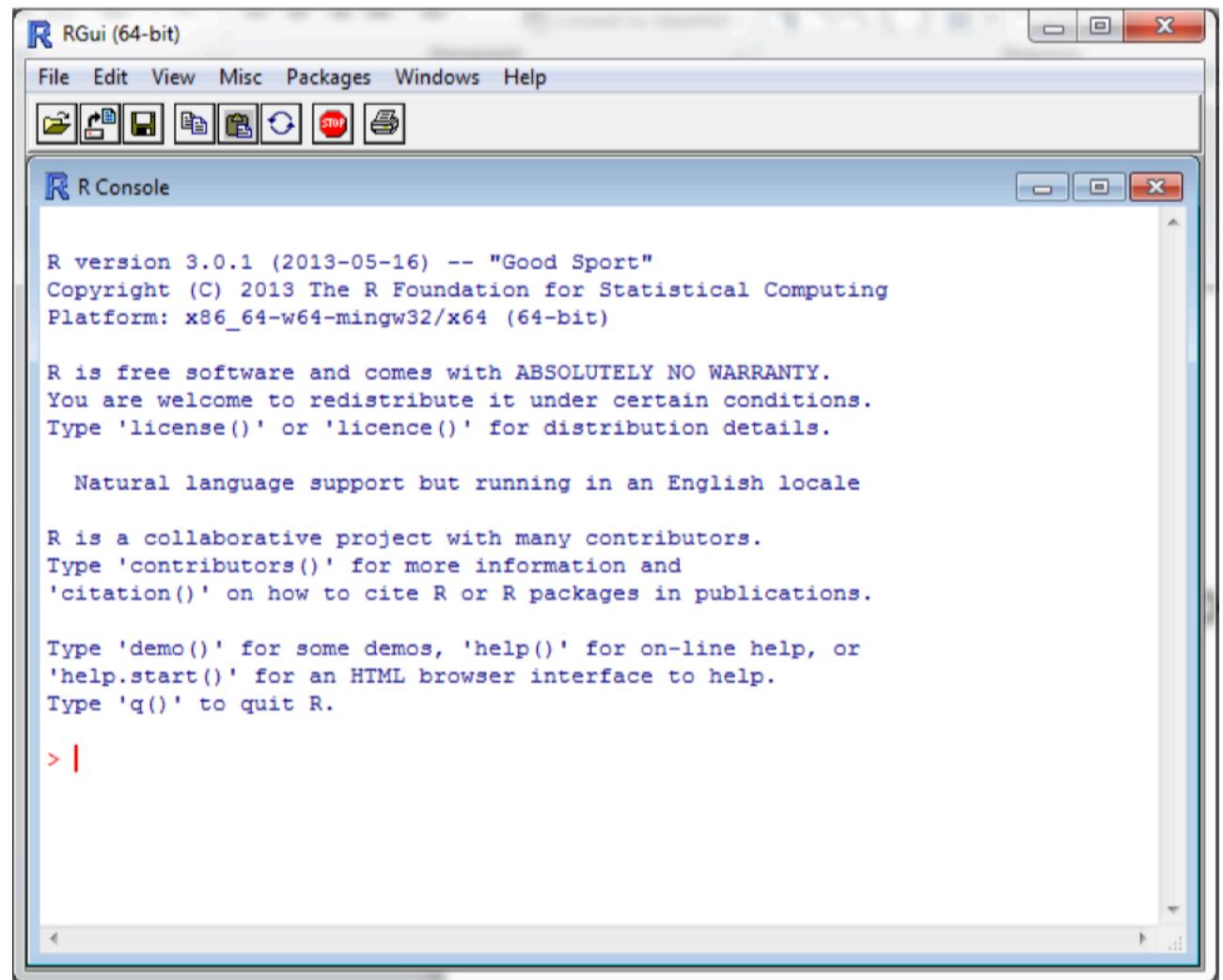
**La mejor manera de aprender R
es usando R!**



Abran R:



Aun no vamos a
trabajar en RStudio:



R Gui (64-bit)

R Console

```
R version 3.0.1 (2013-05-16) -- "Good Sport"
Copyright (C) 2013 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

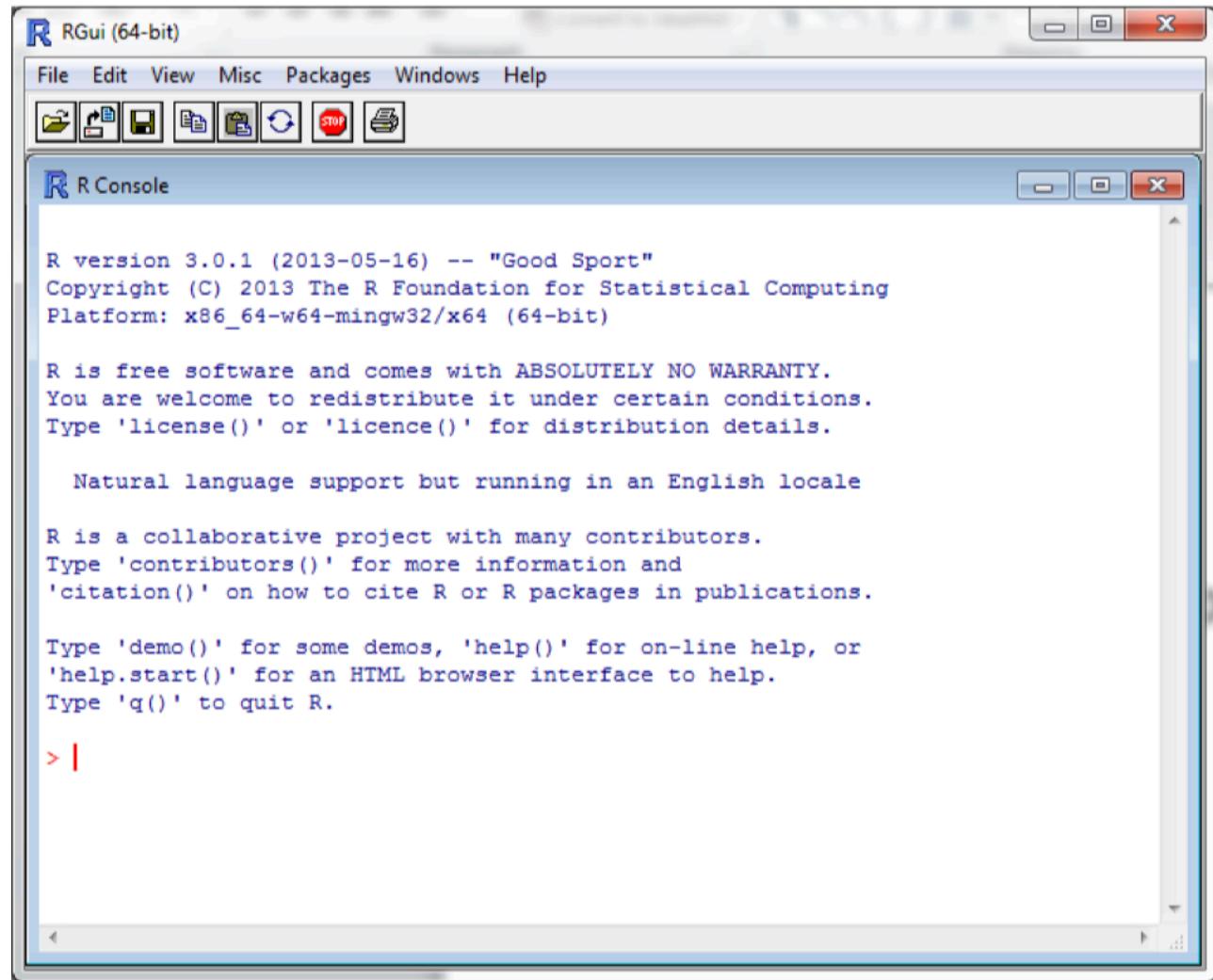
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```



- La forma mas básica de usar R es como calculadora:

>5+5

Y enter!!!

[1] 10

- La forma mas básica de usar R es como calculadora:

```
>5+5
```



Esta letra azul significa que es un **comando de R**

Y enter!!!

```
[1] 10
```



Esta letra negra es el resultado del **comando**

```
>5 * 2
```

```
[1] 10
```

```
>5 *
```

```
2
```



Si escribes un comando incompleto R va esperar que lo completes!

```
>5 *
```



```
+
```



Si quieres cancelar el comando presiona Esc

Importante!

```
> 2+3*4
```

```
[1] 14
```

```
> (2+3)*4
```

```
[1] 20
```

```
> 3^2
```

```
[1] 9
```

```
> 5e3 #notación científica
```

```
[1] 5000
```

Comentar los scripts

- Para comentar los scripts se usa el símbolo `#`

```
# Paulo Intro to R  
2 + 2 # suma
```

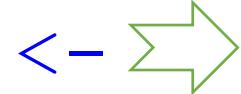
Por qué comentar?



```
fmA1=lm(Y~X1) # fits a linear model on genotypes (additive  
effects)  
Z1=1.0*(X1==1) # Create a variable for A1A2 genotype  
fmAD1=lm(Y~X1+Z1) # fits a linear model on genotypes  
(dominant effects)
```

Asignar variables

```
> x <- 1 # definir variable x  
> x  
[1] 1
```

 asigna lo de la derecha a la variable de la izquierda

También se puede usar `=` para asignar variables, pero puede ser confuso

Nombres de las variables

```
resultado <- 44
```

```
mi nombre <- "paulo"
```

```
Error: unexpected symbol in "mi  
nombre"
```

Nombre de las variables no
pueden tener espacios!

```
miNombre <- 'paulo'
```

```
mi.nombre <- 'paulo'
```

```
mi_nombre <- 'paulo'
```

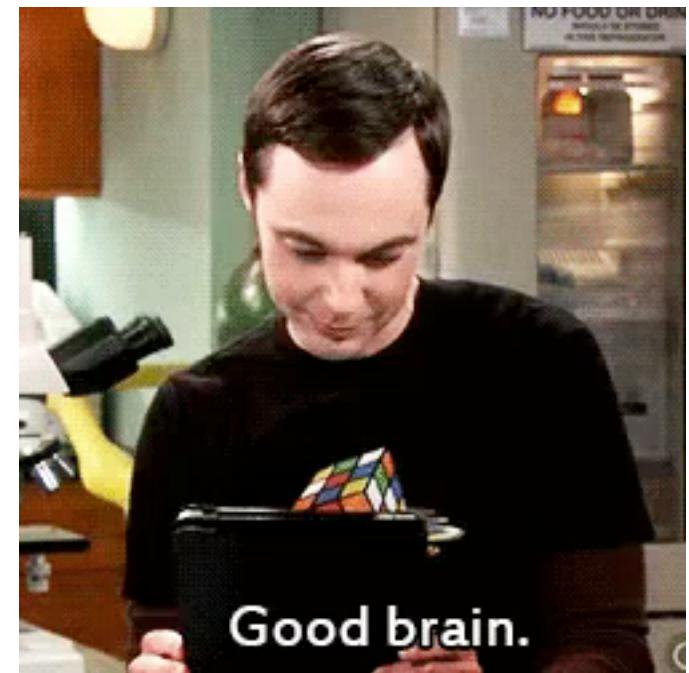
```
minombre <- 'paulo'
```

```
minombre  
[1] "paulo"
```

Memoria de trabajo

- Una vez nombrada una variable, esta se guarda en la **memoria de trabajo** de R

```
> ls() # ver los elementos que están  
en la memoria
```



Memoria de trabajo

Para remover elementos en la memoria:

```
>rm()
```

```
>rm(x)
```

Es buena practica chequear siempre:

```
>ls()
```

```
>rm(list = ls()) # remover todos los elementos de la memoria!
```

```
>ls()
```

Vectores

Los vectores en R son una colección de elementos

```
a <- c(1, 2, 5, 10) # el objeto a ahora es un vector de  
cuatro números
```

```
> a  
[1] 1 2 5 10
```

Vectores

```
a <- c(1, 2, 5, 10)  
[1] 1 2 5 10
```

```
b <- 2*a  
b  
[1] 2 4 10 20
```

```
> 2*a-1      # Recuerden que primero se hacen las divisiones y  
multiplicaciones  
[1] 1 3 9 19
```

Vectores

```
a <- c(1, 2, 5, 10)
```

```
b <- c(-1, 1)
```

```
> b*a # R multiplica los vectores y recicla el mas corto
```

```
[1] -1 2 -5 10
```

Funciones matemáticas

- R tiene muchas funciones matemáticas integradas!
- Para utilizar una función, simplemente escriba el nombre, seguido de paréntesis abiertos y cerrados:

```
> log(10)
```

log es el nombre
de la función

10 es el
argumento

Funciones matemáticas

```
>sqrt(25) #raíz cuadrada
```

```
[1] 5
```

```
>exp(1) # Exponente
```

```
[1] 2.718282
```

```
> x <- c(1,2,3)
```

```
> mean(x)
```

```
[1] 2
```

```
> min(x)
```

```
[1] 1
```

Que funciones existen?

Como las uso?



- GOOGLE es tu amigo!
 - Puedes obtener ayuda de R escribiendo:
? Y el nombre de la función
- ```
> ?log # usar ? Y el nombre de la función para
obtener ayuda
```

`log {base}`

R Documentation

## Logarithms and Exponentials

### Description

### What the function does

`log` computes logarithms, by default natural logarithms, `log10` computes common (i.e., base 10) logarithms, and `log2` computes binary (i.e., base 2) logarithms. The general form `log(x, base)` computes logarithms with base `base`.

`log1p(x)` computes  $\log(1+x)$  accurately also for  $|x| \ll 1$  (and less accurately when  $x$  is approximately -1).

`exp` computes the exponential function.

`expm1(x)` computes  $\exp(x) - 1$  accurately also for  $|x| \ll 1$ .

### Usage

### How to use the function

```
log(x, base = exp(1))
logb(x, base = exp(1))
log10(x)
log2(x)

log1p(x)

exp(x)
expm1(x)
```

### Arguments

### What does the function need

`x` a numeric or complex vector.

`base` a positive or complex number: the base with respect to which logarithms are computed. Defaults to `e=exp(1)`.

### Details

All except `logb` are generic functions: methods can be defined for them individually or via the [Math](#) group generic.

`log10` and `log2` are only convenience wrappers, but logs to bases 10 and 2 (whether computed via `log` or the wrappers) will be computed more efficiently and accurately where supported by the OS. Methods can be set for them individually (and otherwise methods for `log` will be used).

`logb` is a wrapper for `log` for compatibility with S. If (S3 or S4) methods are set for `log` they will be dispatched. Do not set S4 methods on `logb` itself.

All except `log` are [primitive](#) functions.

# ?log

**Value**

### What does the function return

A vector of the same length as `x` containing the transformed values. `log(0)` gives `-Inf`, and `log(x)` for negative values of `x` is `NaN`. `exp(-Inf)` is 0.

For complex inputs to the log functions, the value is a complex number with imaginary part in the range  $[-\pi i, \pi i]$ : which end of the range is used might be platform-specific.

### S4 methods

`exp`, `expm1`, `log`, `log10`, `log2` and `log1p` are S4 generic and are members of the [Math](#) group generic.

Note that this means that the S4 generic for `log` has a signature with only one argument, `x`, but that `base` can be passed to methods (but will not be used for method selection). On the other hand, if you only set a method for the [Math](#) group generic then `base` argument of `log` will be ignored for your class.

### Source

`log1p` and `expm1` may be taken from the operating system, but if not available there are based on the Fortran subroutine `dlnrel` by W. Fullerton of Los Alamos Scientific Laboratory (see <http://www.netlib.org/slatec/fnlib/dlnrel.f> and (for small `x`) a single Newton step for the solution of `log1p(y) = x` respectively).

### References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole. (for `log`, `log10` and `exp`.)

Chambers, J. M. (1998) *Programming with Data. A Guide to the S Language*. Springer. (for `logb`.)

### See Also

### Discover other related functions

[Trig](#), [sqrt](#), [Arithmetic](#).

### Examples

### Sample code -how it works

```
log(exp(3))
log10(1e7) # = 7

x <- 10^{-(1+2*1:9)}
cbind(x, log(1+x), log1p(x), exp(x)-1, expm1(x))
```

# Funciones matemáticas

```
>log(2.718282) #2.71 es la base del logaritmo natural (valor e)
[1] 1
```

```
> log(2)
[1] 0.6931472
```

Argumento opcional

```
> log(2, base = 2.718281828459)
[1] 0.6931472
```

```
> log(2, base=10) == > log10(2)
[1] 0.30103
```

```
[1] 0.30103
```

# Argumentos

**round** toma 2 argumentos

- x (el número que desea redondear) - Requerido
- dígitos (la cantidad de lugares decimales que desea redondear) – opcional – **predeterminado=0**

```
> round(10.123)
```

```
[1] 10
```

```
> round(10.123, 1)
```

```
[1] 10.1
```

```
> round(x=10.123, digits = 1)
```

```
[1] 10.1
```

R tiene bases de datos incorporadas que se pueden usar para practicar

```
> help(package = datasets)
```

```
> cars # ?cars para obtener info de los datos
```

|   | speed | dist |
|---|-------|------|
| 1 | 4     | 2    |
| 2 | 4     | 10   |
| 3 | 7     | 4    |

.....

```
> dim(cars) # para mirar el numero de columnas y filas
```

```
[1] 50 2
```

```
> names(cars) # nombres
[1] "speed" "dist"
```

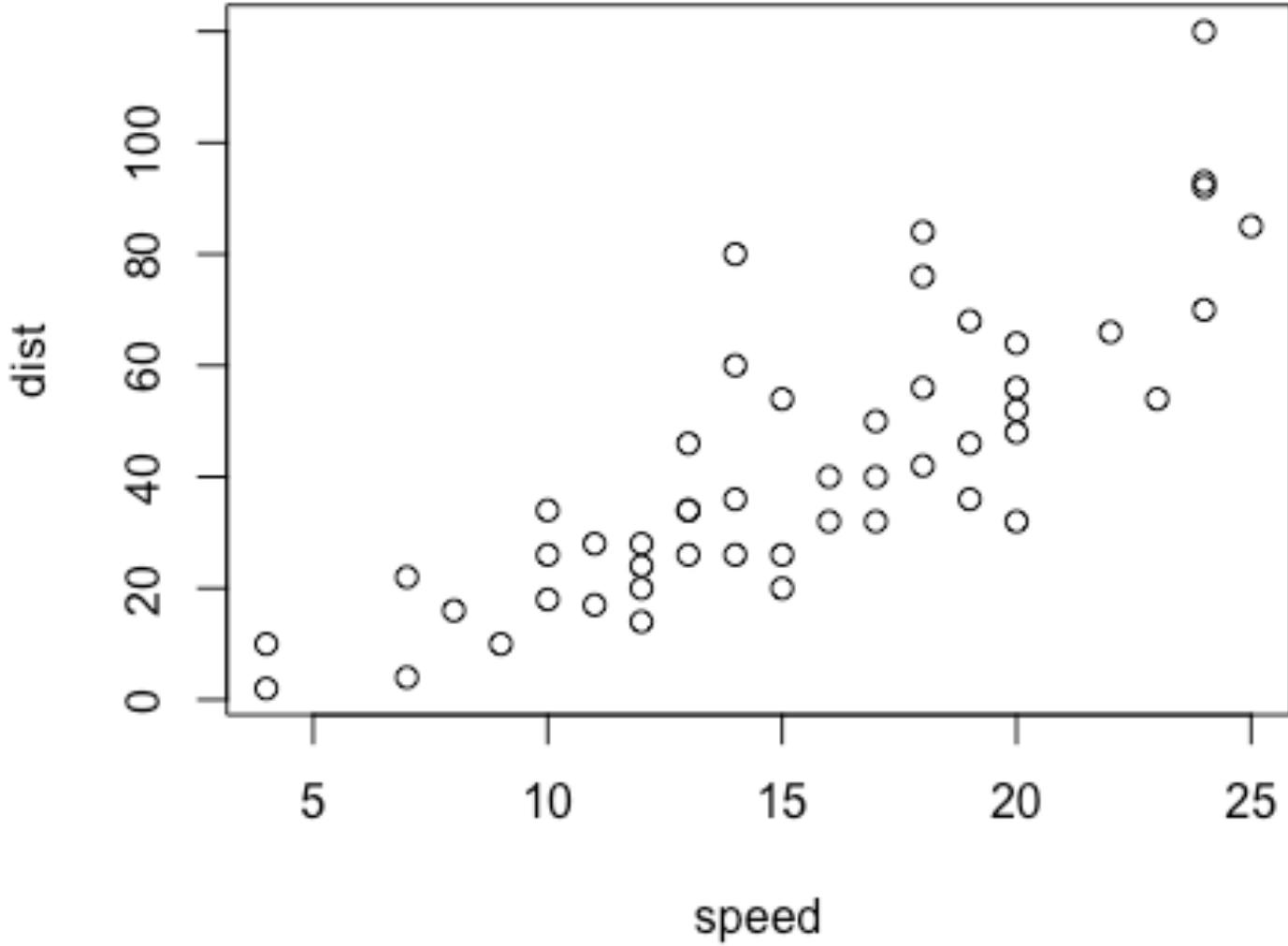
```
> mean(cars$speed) # promedio
[1] 15.4
```

```
> min(cars$speed) # mínimo - $ identifica la columna
[1] 4
```

```
> sd(cars$speed) # desviación estándar
[1] 5.287644
```

```
plot(cars)
```

> plot(cars)



# Funciones de estadística básicas

```
> help(package = stats)
```

Documentation for package 'stats' version 4.0.2

- [DESCRIPTION file.](#)
- [Code demos](#). Use `demo()` to run them.

Help Pages

[A](#) [B](#) [C](#) [D](#) [E](#) [F](#) [G](#) [H](#) [I](#) [K](#) [L](#) [M](#) [N](#) [O](#) [P](#) [Q](#) [R](#) [S](#) [T](#) [U](#) [V](#) [W](#) [X](#) [misc](#)

[stats-package](#)

The R Stats Package

-- A --

[acf](#)

Auto- and Cross- Covariance and -Correlation Function Estimation

[acf2AR](#)

Compute an AR Process Exactly Fitting an ACF

[add.scope](#)

Compute Allowed Changes in Adding to or Dropping from a Formula

[add1](#)

Add or Drop All Possible Single Terms to a Model

[addmargins](#)

Puts Arbitrary Margins on Multidimensional Tables or Arrays

[aggregate](#)

Compute Summary Statistics of Data Subsets

[AIC](#)

Akaike's An Information Criterion

[alias](#)

Find Aliases (Dependencies) in a Model

[anova](#)

Anova Tables

[anova.glm](#)

Analysis of Deviance for Generalized Linear Model Fits

[anova.lm](#)

ANOVA for Linear Model Fits

# Fin de la primera parte.....

- Aprender R es como aprender un idioma, la curva de aprendizaje puede ser empinada
- Sea paciente y recuerde que trabajar en R le ayudara mucho en sus análisis.
- Después de un tiempo trabajar en R es divertido



# Que hemos aprendido hasta ahora?

- Qué es R?
- Asignar variables
- Vectorizar operaciones
- Funciones básicas
- Bases de datos integrados y otras funciones.



# Algunas referencias utiles

- An introduction to R (Venables et al.)
  - <http://cran.r-project.org/doc/manuals/R-intro.pdf>
  - Chapters 1, 2, 5.1-5.4
- YARI—yet another R introduction (Handel)
  - <http://ahandel.myweb.uga.edu/software/yari.pdf>
  - Sections 3.1, 3.2, 4.1, 4.3, 5.1, 5.2, 5.5
- R reference card 2.0 (Baggott)
  - <http://cran.r-project.org/doc/contrib/Baggott-refcard-v2.pdf>
  - Print out: tape up within view of your desk
- Getting started with R: an introduction for biologists (Beckerman & Petchey)