

# Tema 1 - Trabajando con R

Juan Gabriel Gomila & María Santos

Conociendo R

# ¿Qué es R?

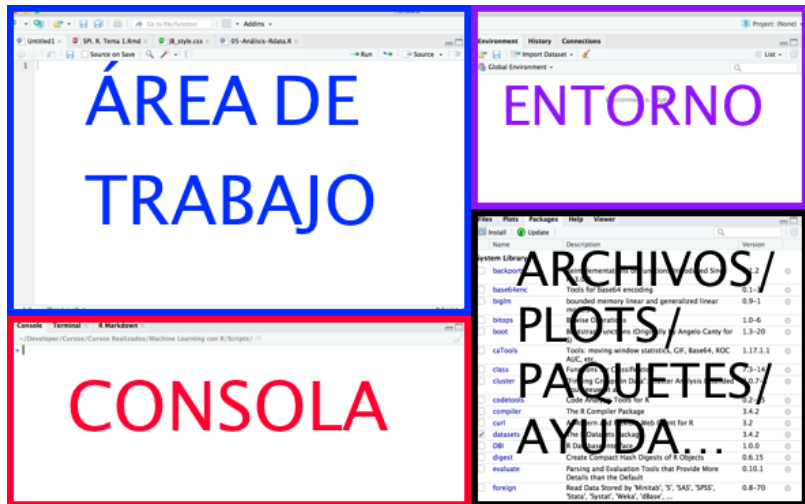


- ▶ Entorno de programación para el análisis estadístico y gráfico de datos
- ▶ Software libre
- ▶ Sintaxis sencilla e intuitiva
- ▶ Enorme comunidad de usuarios (Comprehensive R Archive Network, CRAN)
- ▶ ¿Aún tenéis dudas de por qué usarlo? Haz click aquí

# ¿Qué es RStudio?

En este curso usaremos RStudio como interfaz gráfica de usuario de R para todos los sistemas operativos

Es un entorno integrado para utilizar y programar con R



# Cómo instalar R

## **Si sois de Windows o Mac**

1. Id a CRAN
2. Pulsad sobre el enlace correspondiente a vuestro sistema operativo
3. Seguid las instrucciones de instalación correspondientes

## **Si trabajáis con Ubuntu o Debian**

1. Abrid la terminal, estando conectados a internet
2. Introducid lo siguiente: `sudo aptitude install r-base`

## Cómo instalar RStudio

1. Obtener RStudio
2. **Solo si utilizáis Linux**, ejecutad en una terminal la siguiente instrucción para completar la instalación: `sudo dpkg -i rstudio-<version>-i386.deb`, donde `version` refiere a la versión concreta que se haya descargado



# Trabajando con RStudio



## Cómo pedir ayuda

- ▶ `help()`: obtener ayuda por consola
- ▶ `??...`: obtener ayuda por consola
- ▶ Pestaña Help de Rstudio
- ▶ Cheat Sheet de RStudio
- ▶ Buscar en San Google (stackoverflow, R project...)
- ▶ Foro del curso





# Paquetes: cómo instalarlos y cargarlos

Paquete. Librería con funciones y datos que no necesariamente vienen instaladas de serie

- ▶ `install.packages("nombre_paquete", dep = TRUE)`: instala o actualiza un paquete de R
- ▶ `library(nombre_del_paquete)`: carga un paquete ya instalado

Utilizando R como calculadora

## Calculadora básica - Operaciones

Código	Operación
+	Suma
-	Resta
*	Multiplicación
/	División
^	Potencia
%/%	Cociente entero
%%	Resto de división entera

# Calculadora básica - Operaciones

Código	Significado
pi	$\pi$
Inf	$\infty$
NaN	Indeterminación (Not a Number)
NA	Valor desconocido (Not Available)

# Calculadora básica - Operaciones

```
2+2
```

```
[1] 4
```

```
77%/5
```

```
[1] 15
```

```
77%%5
```

```
[1] 2
```

## Calculadora básica - Funciones

Código	Función
<code>sqrt(x)</code>	$\sqrt{x}$
<code>exp(x)</code>	$e^x$
<code>log(x)</code>	$\ln(x)$
<code>log10(x)</code>	$\log_{10}(x)$
<code>log(x,a)</code>	$\log_a(x)$
<code>abs(x)</code>	$ x $

## Calculadora básica - Funciones

```
sqrt(9)
```

```
[1] 3
```

```
log(exp(1))
```

```
[1] 1
```

```
log(1000,10)
```

```
[1] 3
```

```
log10(1000)
```

```
[1] 3
```

## Calculadora básica - Combinatoria

Código	Operación
<code>factorial(x)</code>	$x!$
<code>choose(n,m)</code>	$\binom{n}{m}$

- ▶ Número factorial. Se define como número factorial de un número entero positivo  $n$  como  $n! = n \cdot (n - 1) \cdot \dots \cdot 2 \cdot 1$
- ▶ Coeficiente binomial. Se define el coeficiente binomial de  $n$  sobre  $m$  como

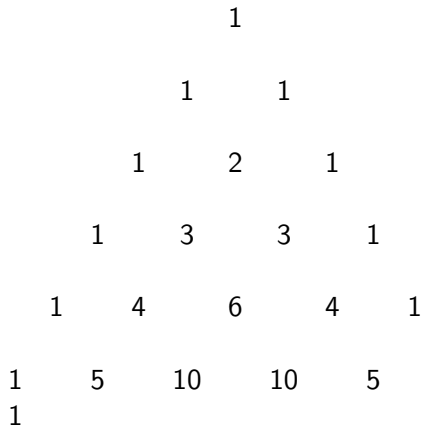
$$\binom{n}{m} = \frac{n!}{m!(n - m)!}$$



## Calculadora básica - Combinatoria

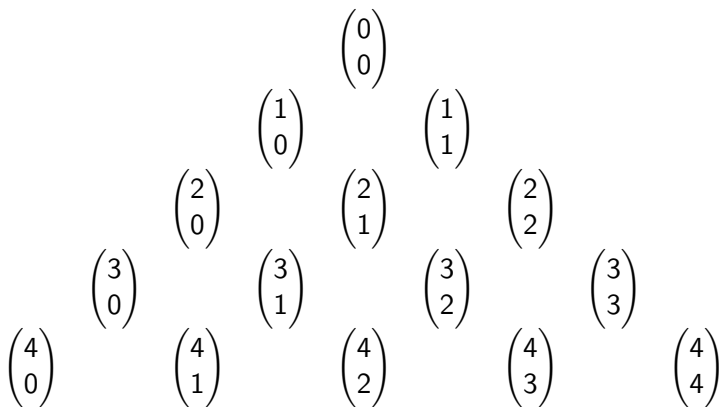
Triángulo de Pascal.

mathdots ymath mathdots MnSymbol



que se corresponde con ...

## Calculadora básica - Combinatoria



A diagram of Pascal's Triangle showing the first five rows (n=0 to n=4). Each entry is a binomial coefficient  $\binom{n}{k}$ , where n is the row index and k is the column index within that row. The triangle is symmetric and each entry is the sum of the two entries directly above it.

			$\binom{0}{0}$			
		$\binom{1}{0}$		$\binom{1}{1}$		
	$\binom{2}{0}$		$\binom{2}{1}$		$\binom{2}{2}$	
$\binom{3}{0}$		$\binom{3}{1}$		$\binom{3}{2}$		$\binom{3}{3}$
$\binom{4}{0}$	$\binom{4}{1}$		$\binom{4}{2}$		$\binom{4}{3}$	$\binom{4}{4}$

## Calculadora básica - Combinatoria

```
factorial(5)
```

```
[1] 120
```

```
choose(4,2)
```

```
[1] 6
```

```
factorial(6)
```

```
[1] 720
```

```
factorial(5)*6
```

```
[1] 720
```

## Trigonometría en radianes

Código	Función
<code>sin(x)</code>	$\sin(x)$
<code>cos(x)</code>	$\cos(x)$
<code>tan(x)</code>	$\tan(x)$
<code>asin(x)</code>	$\arcsin(x)$
<code>acos(x)</code>	$\arccos(x)$
<code>atan(x)</code>	$\arctan(x)$

## Trigonometría en radianes

```
sin(pi/2)
```

```
[1] 1
```

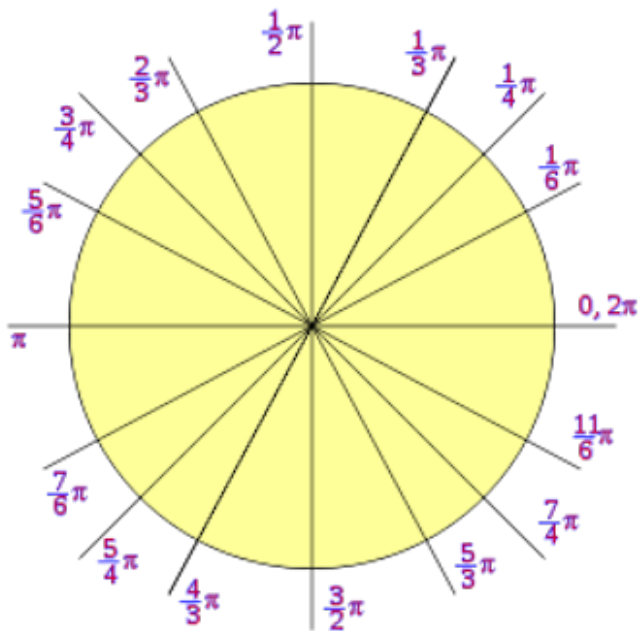
```
cos(pi)
```

```
[1] -1
```

```
tan(0)
```

```
[1] 0
```

## Trigonometría en radianes



## Un pequeño adelanto

```
x = seq(0,2*pi,0.1)
plot(x,sin(x),type="l",col="blue",lwd=3, xlab=expression(x))
lines(x,cos(x),col="green",lwd=3)
lines(x, tan(x), col="purple",lwd=3)
legend("bottomleft",col=c("blue","green","purple"),
      legend=c("Seno","Coseno", "Tangente"), lwd=3, bty="l")
```

## Números en coma flotante

Código	Función
<code>print(x,n)</code>	Muestra las $n$ cifras significativa del número $x$
<code>round(x,n)</code>	Redondea a $n$ cifras significativas un resultado o vector numérico $x$
<code>floor(x)</code>	$\lfloor x \rfloor$ , parte entera por defecto de $x$
<code>ceiling(x)</code>	$\lceil x \rceil$ , parte entera por exceso de $x$
<code>trunc(x)</code>	Parte entera de $x$ , eliminando la parte decimal



## Números en coma flotante

```
print(pi,5)
```

```
[1] 3.1416
```

```
round(pi,5)
```

```
[1] 3.14159
```

```
floor(pi)
```

```
[1] 3
```

```
ceiling(pi)
```

```
[1] 4
```

# Variables y funciones

- ▶ `nombre_variable = valor`: define una variable con dicho valor
- ▶ `nombre_función = function(variable){función}`: define una función

```
miVariable = 4  
doble = function(x){2*x}  
doble(miVariable)
```

```
[1] 8
```

```
cuadrado = function(x){x^2}  
cuadrado(miVariable)
```

```
[1] 16
```

# Números complejos

Código	Función
<code>a+bi</code>	Número complejo
<code>complex(real=...,imaginary=...)</code>	Número complejo en forma binómica
<code>complex(modulus=...,argument=...)</code>	Número complejo en forma polar

# Números complejos

Código	Función
<code>sqrt(as.complex(-x))</code>	$\sqrt{-x}$
<code>Re(x)</code>	Parte real de $x$
<code>Im(x)</code>	Parte imaginaria de $x$
<code>Mod(x)</code>	Módulo de $x$
<code>Arg(x)</code>	Argumento de $x$
<code>Conj(x)</code>	Conjugado de $x$

# Números complejos

```
z = 2+3i  
z2 = complex(real = 2, imaginary = -3)  
Re(z)
```

```
[1] 2
```

```
Im(z)
```

```
[1] 3
```

```
Conj(z2)
```

```
[1] 2+3i
```

## Números complejos

