

**Analítica de datos**

# **Introducción a la analítica de datos**



Pontificia Universidad  
**JAVERIANA**  
Bogotá

Profesor: Nicolás Velásquez

# Introducción personal

---

- Nicolás Velásquez ([nikvel250@gmail.com](mailto:nikvel250@gmail.com) ó [velasquez\\_juan@javeriana.edu.co](mailto:velasquez_juan@javeriana.edu.co) )



M.A. en Economía, Universidad de los Andes

- Áreas de interés:



Analítica de Datos en Marketing Digital (A/B Testing y Modelación  )



Economía Experimental (Prosocialidad en contextos de postconflicto  )

- Más información: [www.linkedin.com/in/juan-nicolas-velasquez/](https://www.linkedin.com/in/juan-nicolas-velasquez/)

Mi nombre es \_\_\_\_\_.

Algo que disfruto mucho hacer es \_\_\_\_\_.

**Preferirías analizar datos de:**

- ☐ Una empresa de chocolates
- ☐ Un equipo de fútbol
- ☐ Una campaña de Marketing

# NORMAS DE CONVIVENCIA

---







 Celulares en silencio.

 Si llega tarde, ingrese con respeto.

 Si debe traer hijos/as, ubíquese cerca de la puerta.

 ♂ Mantenga respeto por quien habla (profesor o compañeros).

# RESULTADOS DE APRENDIZAJE ESPERADOS

- 1  Interpretar estadísticos, gráficos, pruebas y regresiones.
- 2  Aplicar principios de visualización efectiva.
- 3  Distinguir causalidad y correlación.
- 4  Construir modelos de regresión para decisiones.
- 5  Evaluar técnicas aplicadas a negocios.
- 6  Diseñar experimentos sencillos. (A/B testing)

💡 Intuición y experiencia para escoger ruta.



vs

📊 Waze o Maps ayudan a tomar una decisión informada.



**Quienes toman decisiones deben adaptarse al mundo con cada vez más datos disponibles.**

# Estructura del Curso



Comenzamos suponiendo que ya tenemos los datos.

- Exploraremos formas de resumir datos usando **R**:



Visualizaciones.



Estadísticos descriptivos.



Aprenderemos a construir **intervalos de confianza**, que nos dicen con qué tanta precisión estamos calculando un promedio.



Aprenderemos a elaborar **pruebas de hipótesis**, que nos dicen si dos grupos son distintos en cuanto a alguna característica.

# Estructura del Curso



Usaremos **regresiones**, que explotan las relaciones entre dos o más variables para predecir y para prescribir.



Dando un paso atrás, veremos distintas formas de **recolectar datos** y cómo hacerlo apropiadamente mediante encuestas y experimentos.



Si nos da tiempo veremos Modelo Bayesianos y como los modelos son afectados por creencias (Priors).



# Nada de nervios...

---

**Pánico a la Estadística:** Bloqueo mental que resulta de pensar en una clase basada en estadística

- En ocasiones el resultado es ni siquiera intentar entender.
- Se presentarán los materiales a un nivel introductorio:
  - Sin mucha teoría ni expresiones matemáticas.
  - Se explica el concepto en términos muy generales (importante entender) y se ilustra con muchos ejemplos.



# Que **NO** vamos a ver en este curso

---

- **Big data:** Aplicación de Analytics a problemas que involucren enormes cantidades de datos
  - Muchas observaciones
  - Muchas variables
  - Datos no estructurados
- Característica distintiva: Uso de métodos de ciencias de la computación
- Mayoría de problemas de Analytics a la que están expuestos ejecutivos involucran **“Small data”**.

# Curso de R en Datacamp



Otorga  
Certificado

- Para que podamos avanzar rápidamente en el curso, deberán realizar un **curso introductorio de R**.
- La plataforma es intuitiva y está basada en el enfoque de **learning by doing**.
- Es importante completarlo, ya que condiciona la nota de las tareas. Si no se finaliza antes de la fecha especificada, la nota empezará a disminuir.

# Uso de Casos Harvard

---

Utilizaremos casos de la vida real, con datos reales, para elaborar nuestro análisis.

- Es importante hacer el curso llamado Case Companion (Brightspace).
- Hay que leerlos previo a cada sesión (ver syllabus).

Discutiremos el caso, definiremos una estrategia para resolverlo, y ustedes harán el análisis en R.

# Cadena de valor de Business Analytics

Actividad	Software	Objetivo	Método
Recolección, almacenamiento y organización de datos.	<b>SQL / Hadoop / MS / BigQuery / Google Cloud</b>	Construcción y arquitectura de bases de datos. Ejemplo base de datos de transacciones de los clientes. Amazon.	No cubierto
Análisis de datos	<b>MS Excel, R , Stata, Python, SPSS</b>	Análisis estratégico. Ejemplo: Estrategia de ventas.	MS Excel / Casos
Comunicación de los resultados	<b>MS Power Point / MS Word / Latex / Tableau /Power BI</b>	Diseño de presentación de resultados. Reportes y presentaciones.	Power point / Reportes de talleres y trabajo final

# Reglas para el desarrollo del taller/caso

---

Usaremos GitHub(tendremos una sesión virtual) y R:

- **Grupos de 2 o 3 personas.**

- **Participación activa:**

Cada estudiante debe hacer al menos **2 commits significativos** en el repositorio de GitHub.

- **Entregables:** Deben entregar:

- Aparte del código en Github deben entregar un documento con justificación de respuestas Word/PDF.
- Justifiquen sus respuestas y decisiones.
- Expliquen brevemente el enfoque metodológico utilizado.
- Señalen dificultades encontradas y cómo las resolvieron.

# Reglas para el desarrollo del taller/caso

---

- **Honestidad y replicabilidad:**

- El trabajo debe ser propio y **replicable por otros**, con código claro, documentado y funcional.

- **Buen uso de GitHub:**

- Commits descriptivos.

- Repositorio organizado y con instrucciones claras (README).

# Evaluación

Evaluación basada en:

- 1** Parcial 1 y 2. 15% cada uno.
- 2** Tareas y Quizzes. 30%.
- 3** Proyecto caso final (en equipos de 2, más información pronto). 40%.

*Parcial 1: 20 de Agosto de 2025.*

*Parcial 2: 08 de Octubre de 2025.*



# Introducción



# ¿Qué es la analítica de datos para los negocios?

Es la aplicación de tecnologías informáticas y herramientas estadísticas que permiten analizar datos relevantes para la toma de decisiones dentro una empresa/organización.

# Datos y Toma de Decisiones

---

## Estilo Vieja Escuela



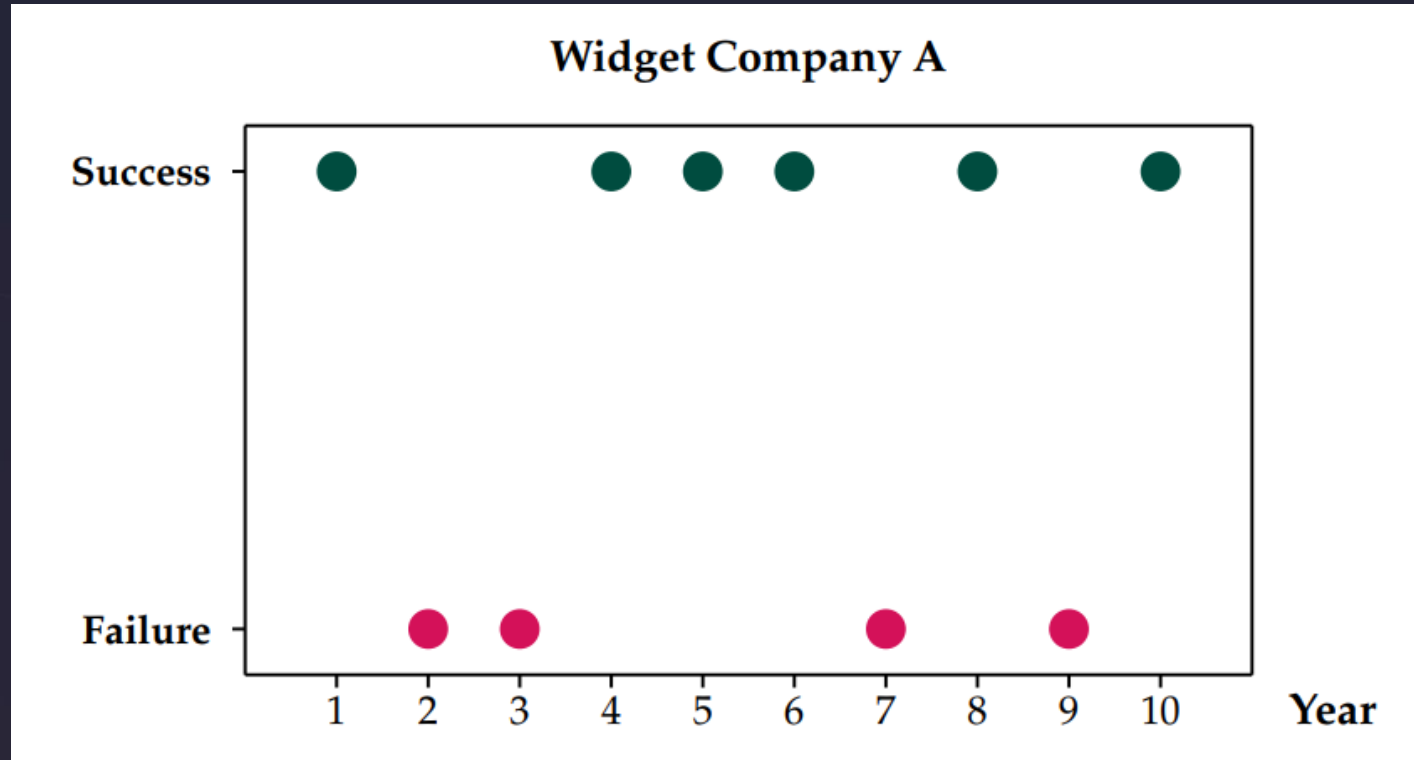
Las decisiones las toman **HiPPOs**  
(Highest Paid Person in Organization)  
basados en intuición.

QualPro encontró que  
este tipo de decisiones  
tenían peores resultados  
que si hubiesen sido  
tomadas aleatoriamente.



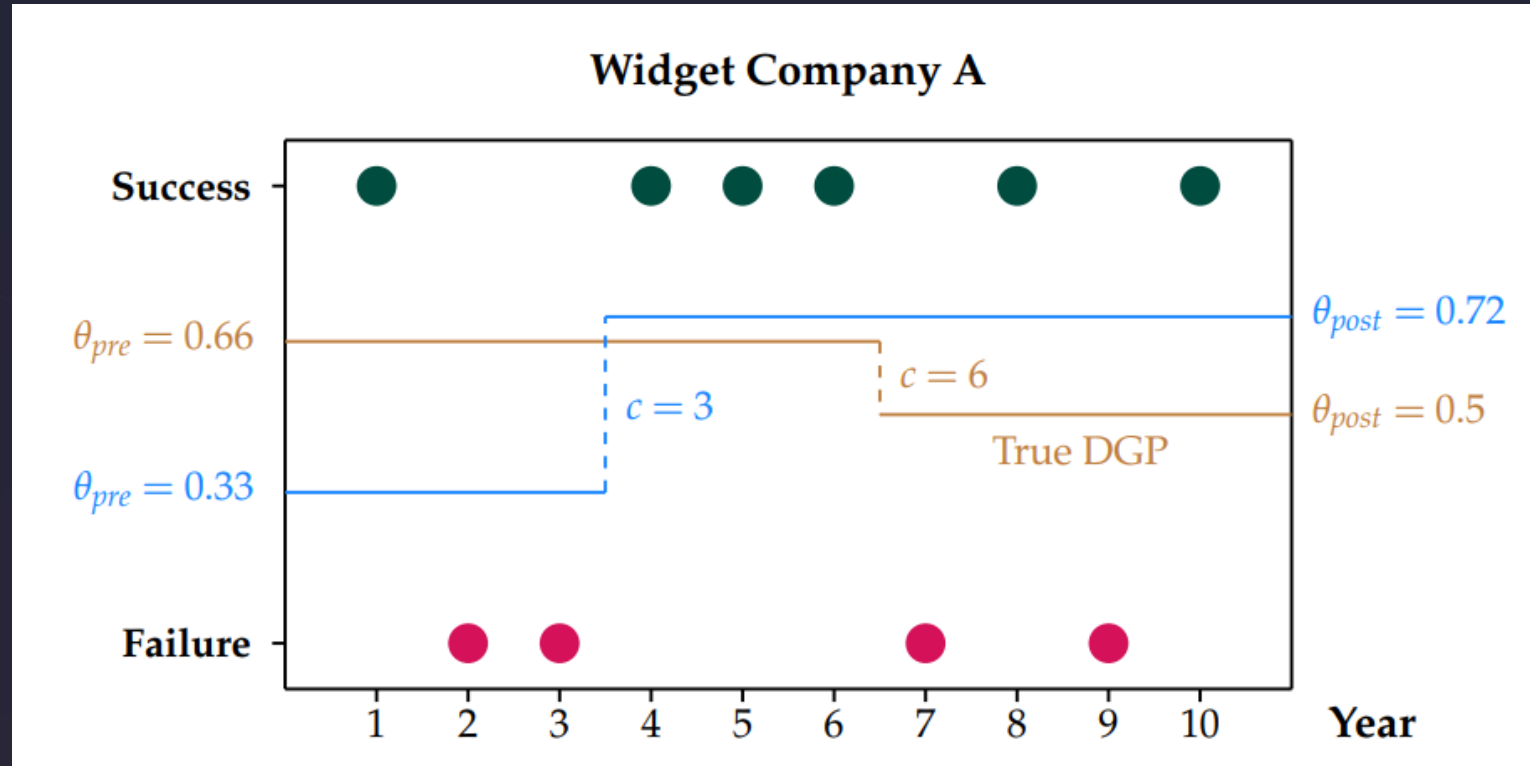
**Este problema se puede  
corregir usando datos para  
argumentar las decisiones.**

# Los datos por sí solos no son suficientes...



Fuente: Barron & Fries (2022)  
Narrative persuasion.

# Los datos por sí solos no son suficientes...

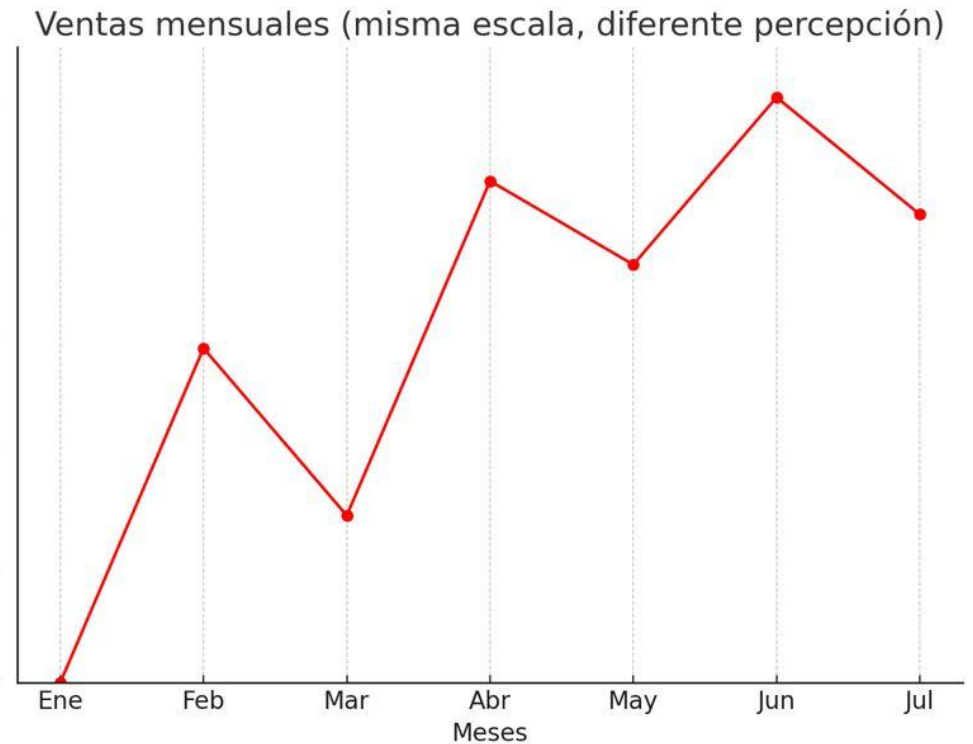
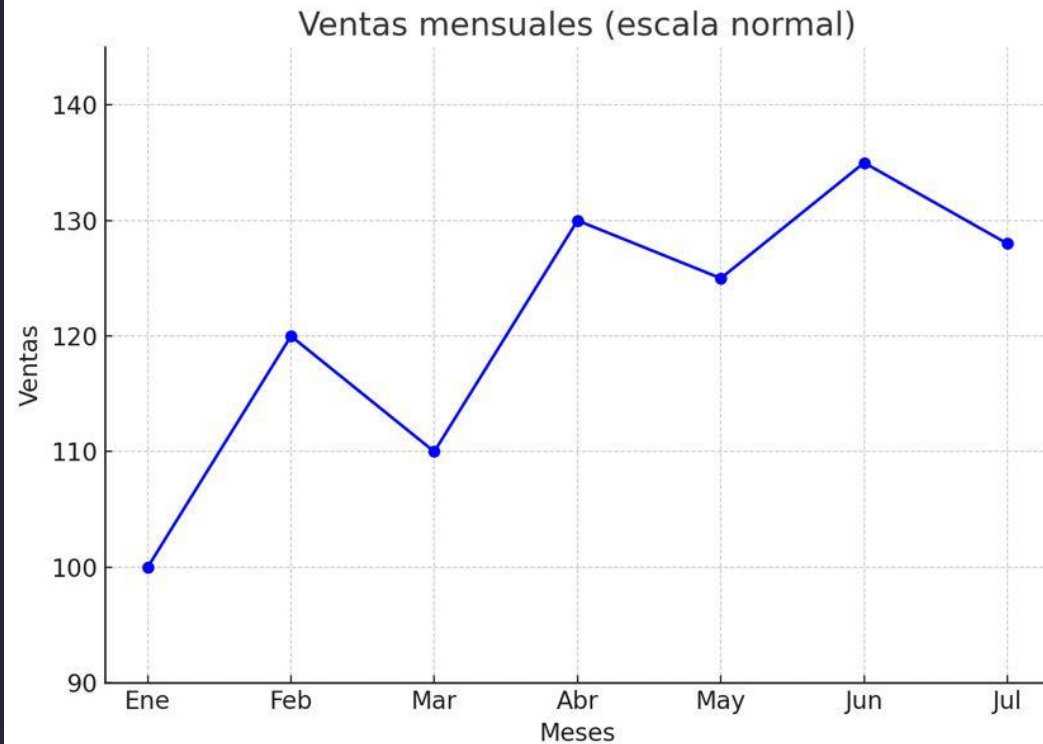


Fuente: Barron & Fries (2022)  
Narrative persuasion.





La interpretación de los datos es crucial para contar una historia que persuade!!

# Otro caso visual

Manipulación visual: misma escala de datos con diferente percepción



# Ejemplos de preguntas que podríamos responder con Analítica de Negocios

-  ¿Cuál es la probabilidad de incumplimiento de crédito para un solicitante de préstamo?
-  ¿Cómo podemos optimizar la asignación de capital para maximizar los retornos dada una tolerancia al riesgo específica?
-  ¿Qué factores influyen más en el precio de las acciones de una empresa?
-  ¿Cómo podemos detectar transacciones financieras fraudulentas en tiempo real?

# Ejemplos de preguntas que podríamos responder con Analítica de Negocios



71.680  
TOTAL IMPRESSIONS

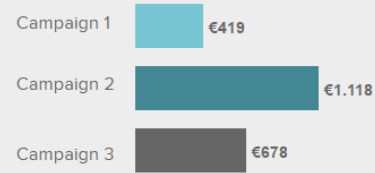
2.203  
TOTAL CLICKS

106  
TOTAL ACQUISITIONS

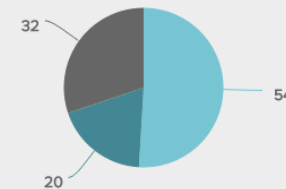
€ 631  
COST PER ACQUISITION



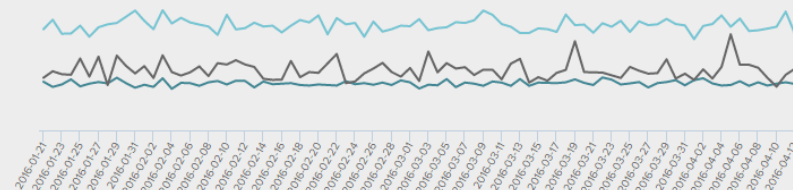
CPA by Campaign - Last 12 weeks



Acquisitions by Campaign - Last 12 weeks







Daily Impressions by Campaign - Last 12 weeks





# Ejemplos desde la Marketing Digital

-  ¿Qué diseño de anuncio (A/B testing) maximiza la tasa de conversión en una campaña digital?
-  ¿Cuál es el Valor de Vida del Cliente (CLTV) para diferentes segmentos de clientes?
-  ¿Cómo podemos optimizar el gasto publicitario entre diversos canales digitales para maximizar el ROI?
-  ¿Qué contenido o promoción personalizada debería ofrecer a un cliente para aumentar la probabilidad de compra?

# Ejemplos (con uso de grandes cantidades de datos)

---

Alpha house & House of Cards



Target



# Aspectos de la analítica

---

**Analítica descriptiva:** visualizar y tabular datos que ya se tienen para entender cambios o la situación actual de un negocio (básicamente describir la información que se tiene)

*¿Han crecido las ventas después de la introducción de un nuevo plan de mercadeo?*

*¿Qué regiones son las más débiles en ventas?*

*¿Cuáles son las características de mis clientes?*

*¿Cuánto ha variado el precio de las acciones de Coca Cola en los últimos 5 años?*

# Aspectos de la analítica

---

**Analítica predictiva:** predecir qué pasará, explotando relaciones entre variables. Para predecir, es suficiente que las variables que estamos estudiando estén correlacionadas, no es necesario que una cause otra.

- *Puntaje de crédito (probabilidad de morosidad)*
- *Retención de clientes (probabilidad de perderlo, intervención temprana)*
- *¿Cuánto me costará atender a un paciente de acuerdo a sus características?*
- *Detección de fraudes (probabilidad de que sea fraudulenta)*
- *Protección de infantes (probabilidad de que el niño sea maltratado)*
- *Predicción de inventarios*

# Aspectos de la analítica

---

**Analítica prescriptiva:** se enfoca en hacer uso de la analítica para determinar cuál es la mejor decisión que se podría tomar. Aquí es importante determinar causalidad. Muchas veces se hace con experimentos.

*¿Cuál de los diseños de página deberíamos implementar?*

*¿Si subimos el precio aumentarían las ganancias?*

*¿Sería efectivo implementar un sistema de compensación basado en desempeño?*

# Verifiquemos que aprendimos...

Vayan a [www.menti.com](https://www.menti.com) código **1529 5380**



# Double check...

Vayan a [www.menti.com](https://www.menti.com) código **4873 7527**

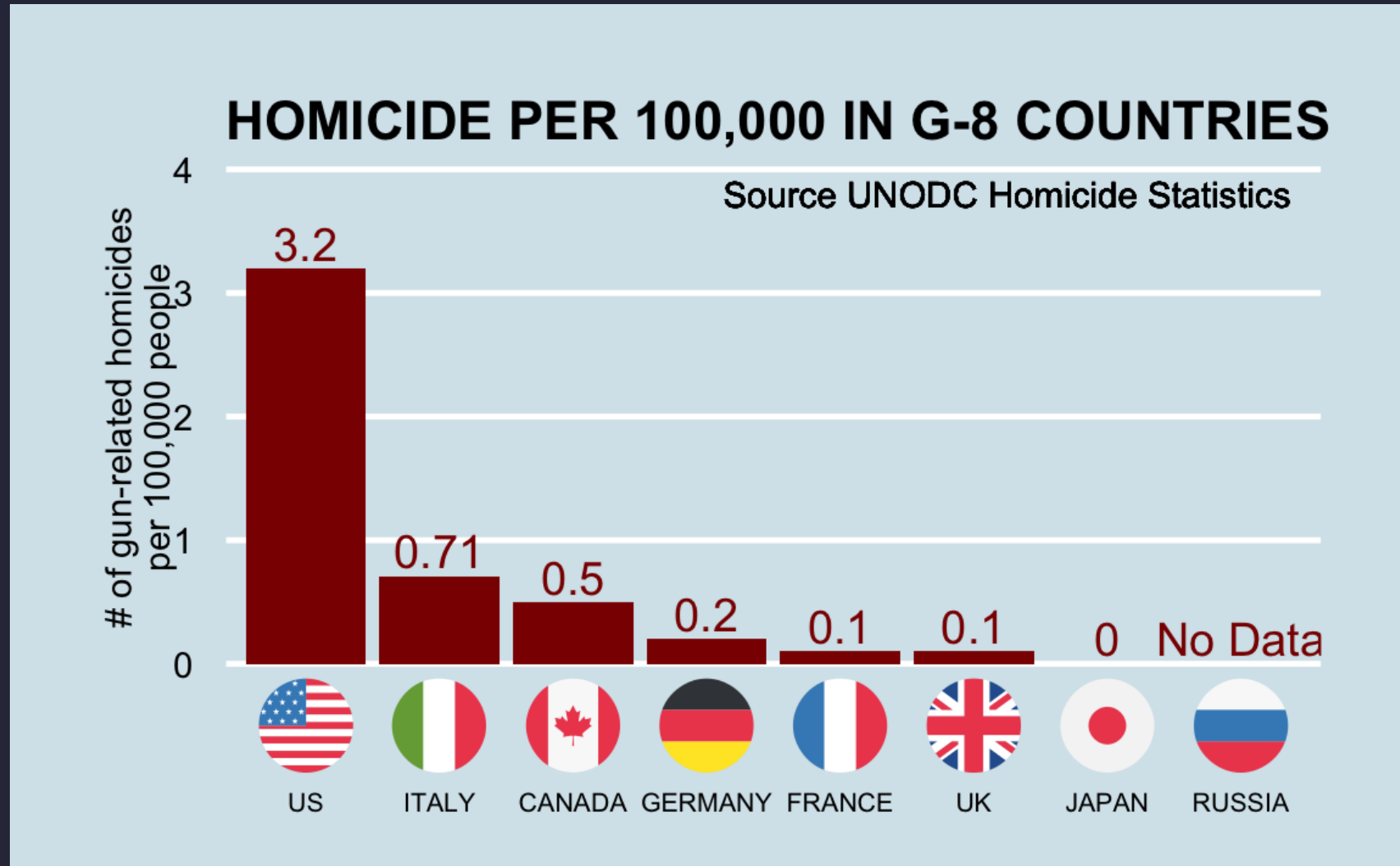


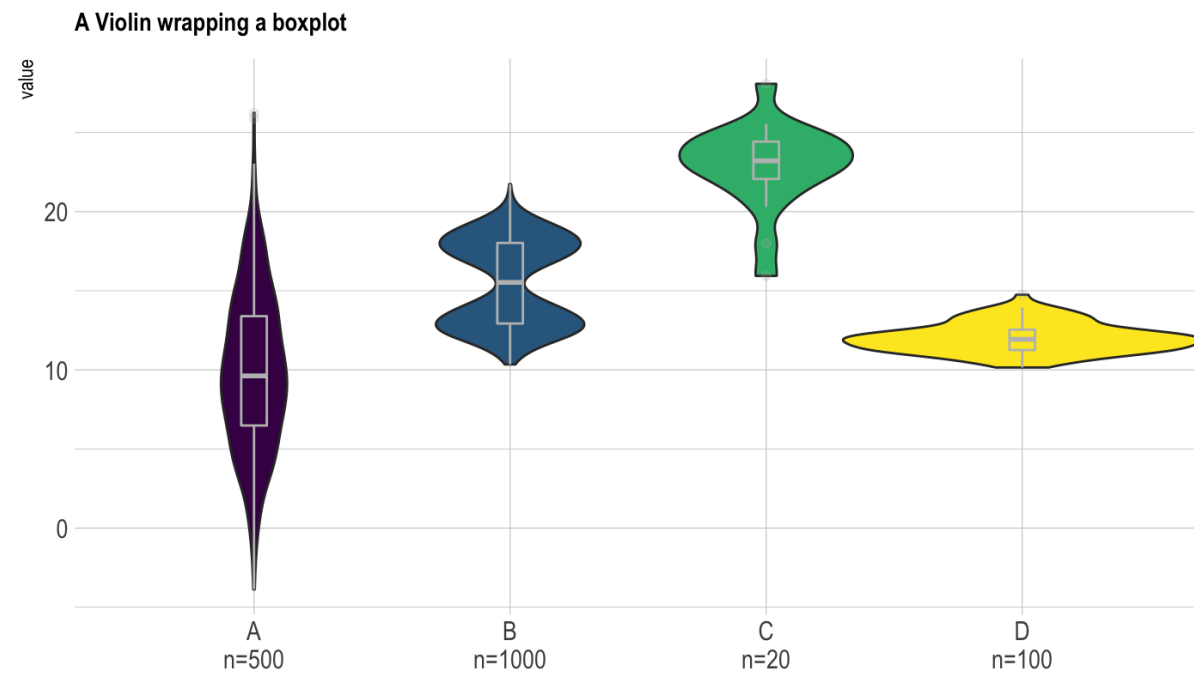
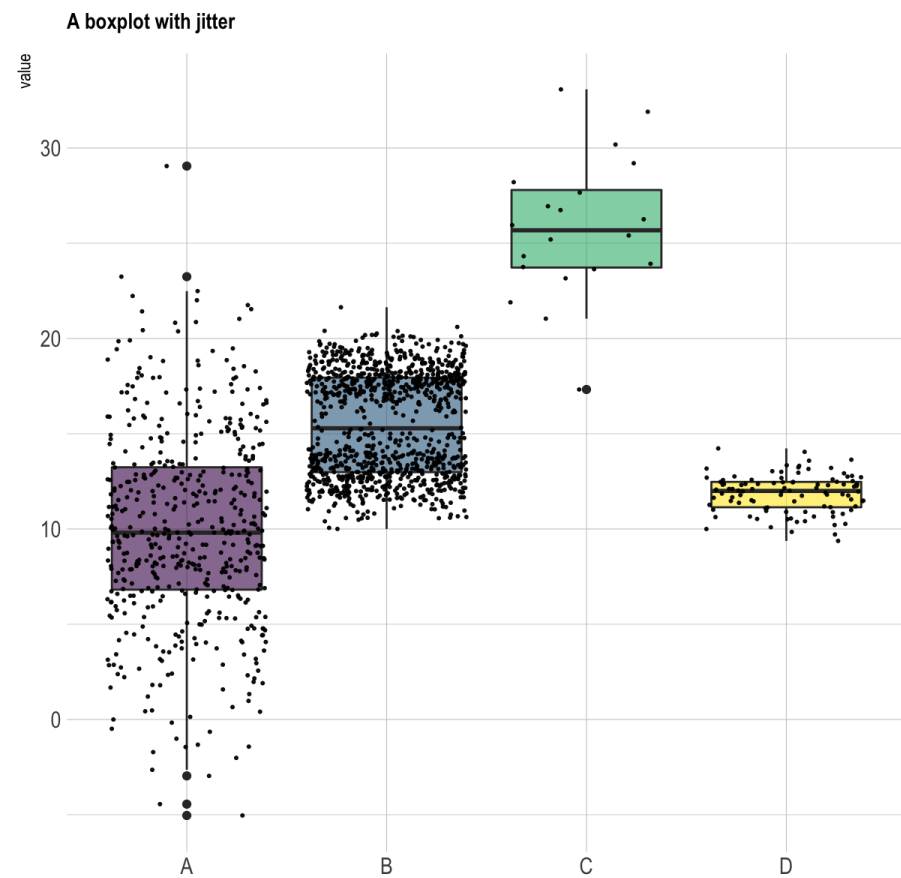
# R Básico

Sintaxis, Rstudio, y operaciones básicas

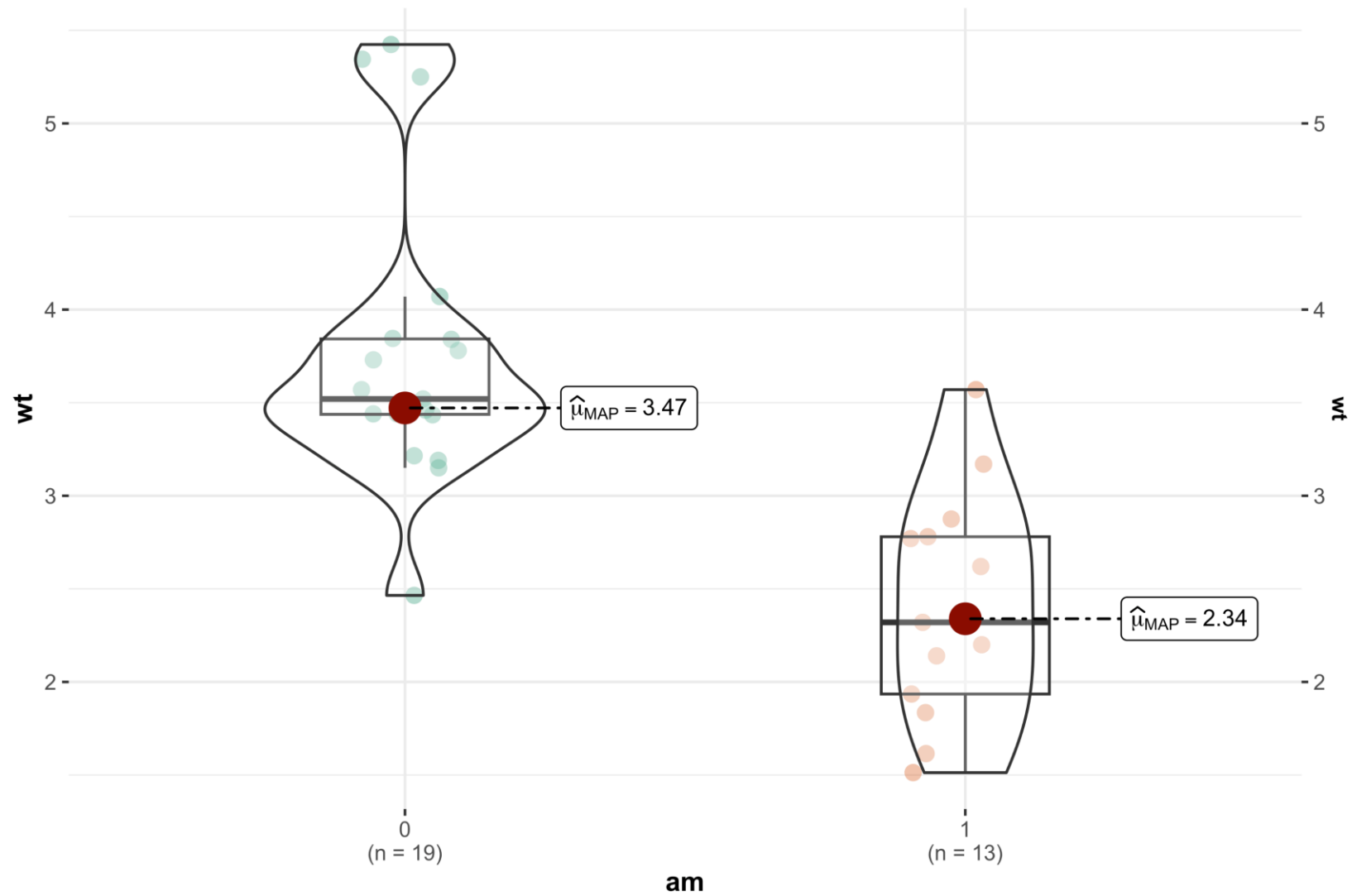


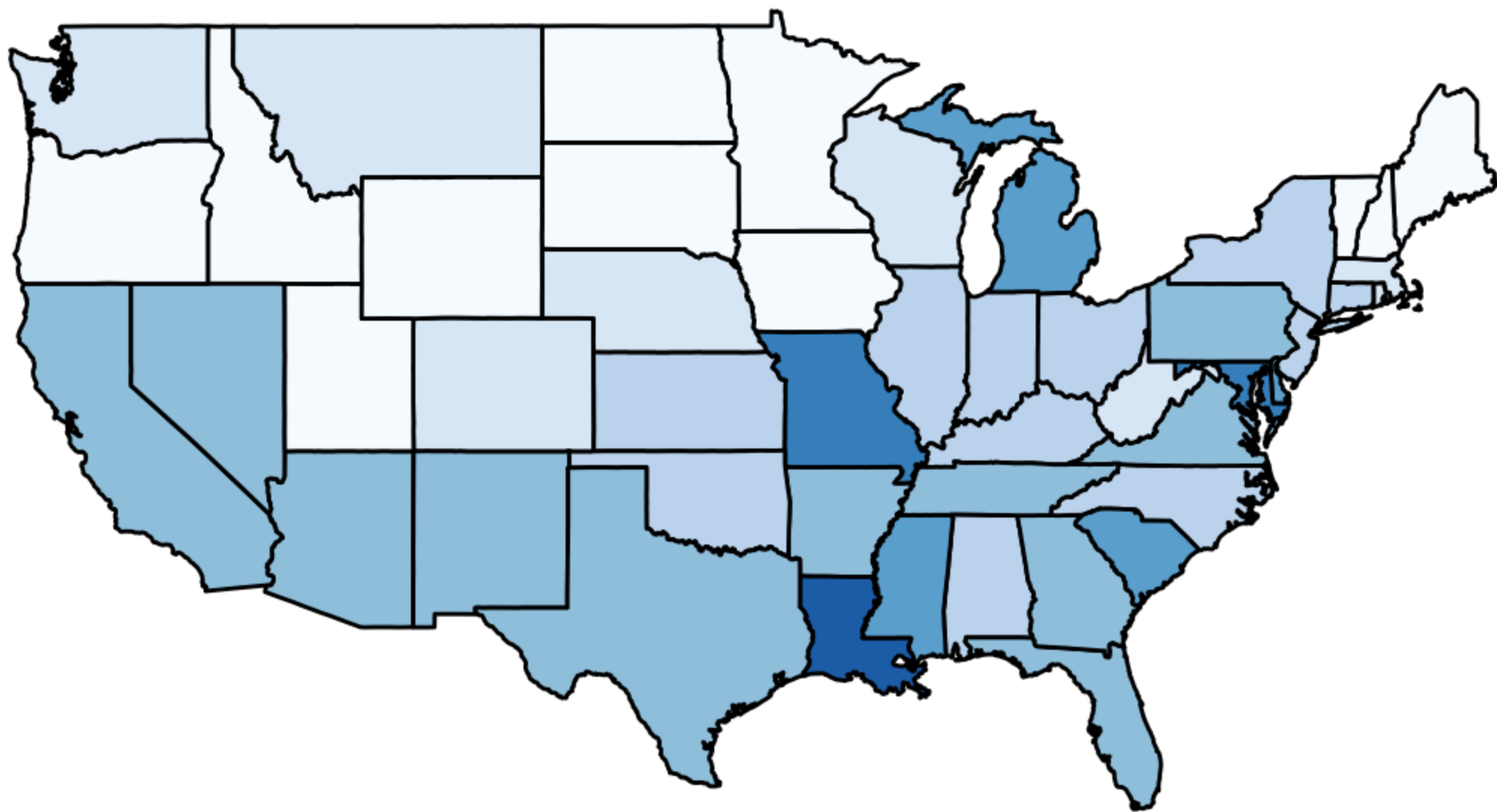
# Ejemplos de motivación sencillos





$\log_e(\text{BF}_{01}) = -7.23$ ,  $\hat{\delta}_{\text{difference}}^{\text{posterior}} = 1.27$ ,  $\text{CI}_{95\%}^{\text{ETI}} [0.74, 1.79]$ ,  $r_{\text{Cauchy}}^{\text{JZS}} = 0.71$





# Recursos

- Grolemund & Wickham 2017: R for Data Science (avail. online for free)
- Kabacoff 2015: R in Action
- Salganik 2017: Bit by Bit (avail. online for free)
- Chang 2013: R Graphics Cookbook
- interactive SWIRL tutorials
- R programming course by John Hopkins Univ. / Roger Peng at Coursera
- <http://rafalab.dfci.harvard.edu/dslibro/r-basics.html>

# ¿Qué es R?



- Un lenguaje de programación estadística y un entorno informático gratuitos y de código abierto
- basado en el lenguaje **S** desarrollado en los Laboratorios Bell
- desarrollado inicialmente en 1993 por Ross Ihaka y Robert Gentleman en la Universidad de Auckland, Nueva Zelanda

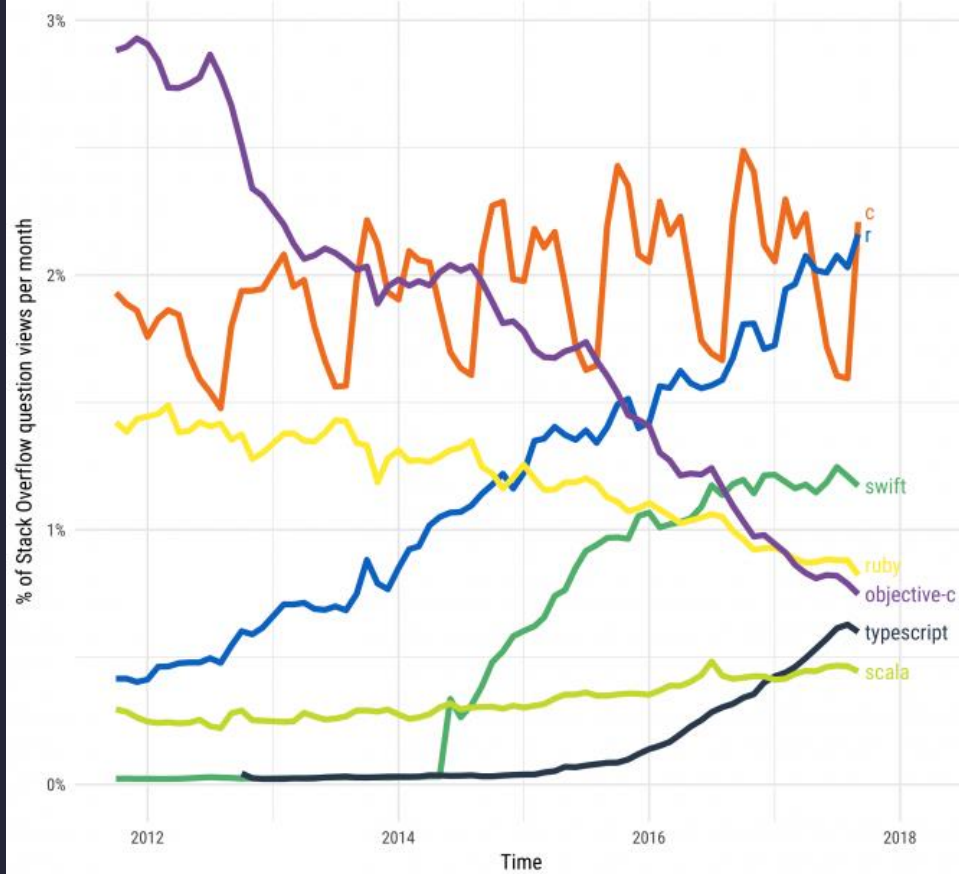
# ¿Por qué R?

- gratuito y de código abierto.
- funciona en los principales sistemas operativos
- base de software probada y fiable.
- combina un modelo de programación flexible con una amplia gama de métodos estadísticos.
- desarrollo activo y amplia comunidad.
- fácilmente ampliable mediante paquetes R.

# ¿Por qué R?

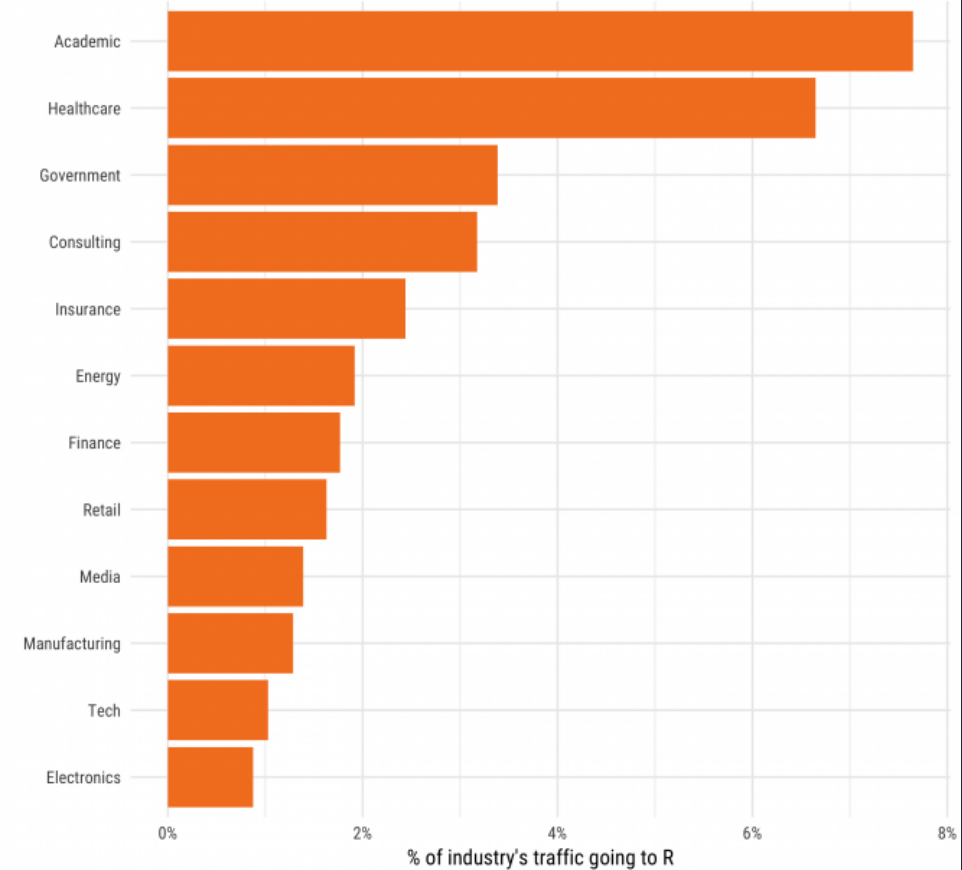
## Stack Overflow Traffic to Programming Languages

Based on visits to Stack Overflow questions from World Bank high-income countries. The more-visited languages of Python, JavaScript, Java, C#, and PHP were omitted.



## Visits to R by industry

Based on visits to Stack Overflow questions from the US/UK in January-August 2017. The denominator in each is the total traffic from that industry.

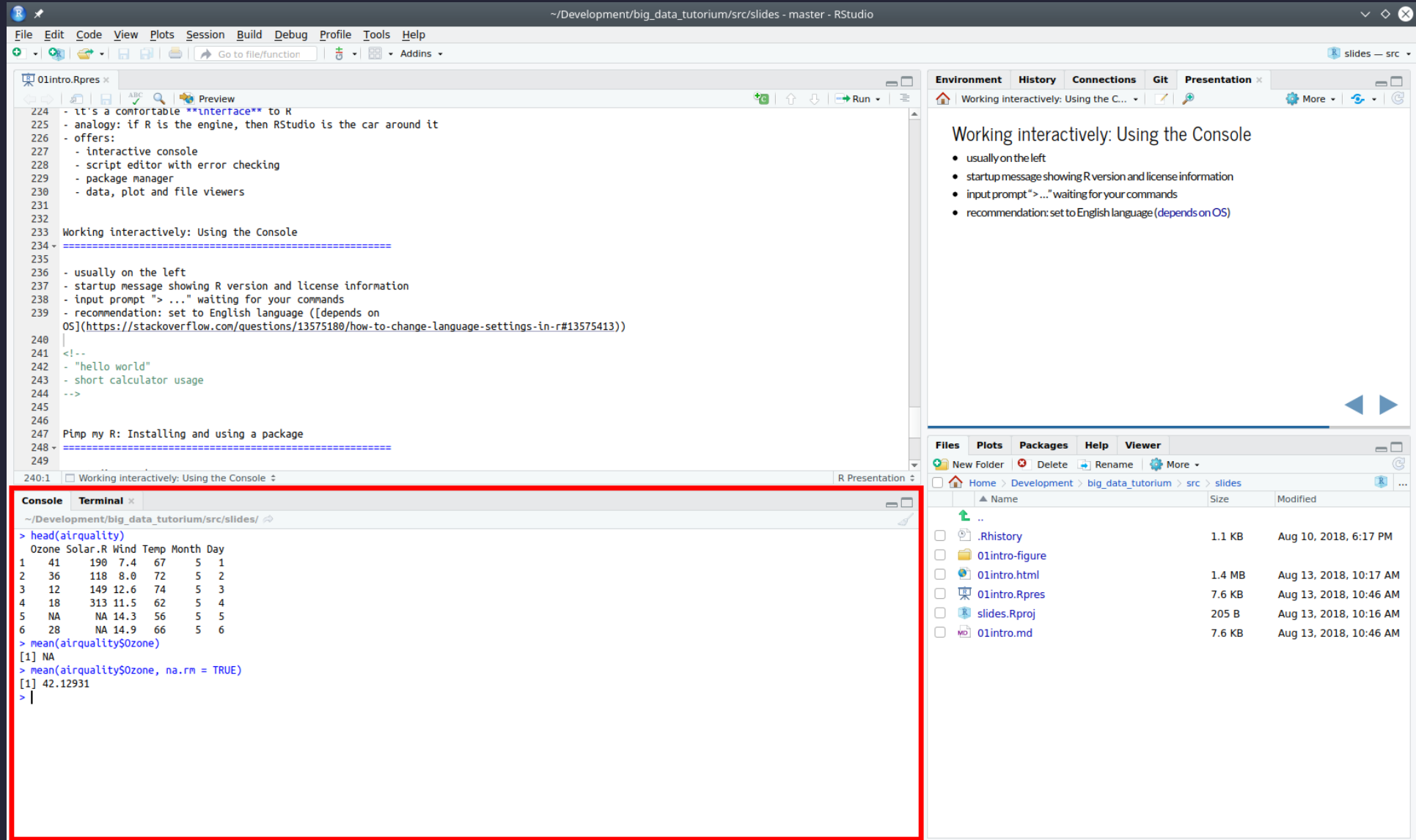






- RStudio es un entorno de desarrollo integrado (IDE) para R
- es una interfaz cómoda para R
- analogía: si R es el motor, RStudio es el coche que lo rodea
- ofrece:
  - consola interactiva
  - editor de scripts con comprobación de errores
  - gestor de paquetes
  - visores de datos, gráficos y archivos

# Trabajar de forma interactiva: Uso de la consola



The screenshot displays the RStudio interface with a presentation slide on the right and a console window at the bottom. The slide, titled "Working interactively: Using the Console", lists the following points:

- usually on the left
- startup message showing R version and license information
- input prompt "> ..." waiting for your commands
- recommendation: set to English language (depends on OS)

The console window shows the execution of R commands and the output of the `airquality` dataset:

```
> head(airquality)
  Ozone Solar.R Wind Temp Month Day
1   41    190  7.4   67     5    1
2   36    118  8.0   72     5    2
3   12    149 12.6   74     5    3
4   18    313 11.5   62     5    4
5   NA     NA 14.3   56     5    5
6   28     NA 14.9   66     5    6
> mean(airquality$Ozone)
[1] NA
> mean(airquality$Ozone, na.rm = TRUE)
[1] 42.12931
> |
```

The console window is highlighted with a red border.

# Trabajar de forma interactiva: Uso de la consola

- normalmente en la parte inferior izquierda
- mensaje de inicio mostrando la versión de R y la información de la licencia
- prompt de entrada "> ..." esperando sus comandos (los comandos se emiten usando ENTER)
- salida: depende del tipo de datos
- **consejo general:** todos los comandos distinguen entre mayúsculas y minúsculas
- **recomendación:** configure el idioma inglés (depende del sistema operativo)

# Algunas reglas generales de sintaxis de R

- Una sintaxis describe las reglas de un lenguaje de programación
- Un lenguaje de programación es muy estricto en cuanto a la gramática
- Algunas reglas generales:
  1. Cada línea es una sentencia ("comando"), varias sentencias se evalúan de arriba a abajo.

```
c <- a + b  
d <- sqrt(c)
```

- Excepción: Si una expresión no está cerrada (véase la regla de la paréntesis más abajo), puede abarcar varias líneas:

```
a * (b  
+ c  
+ d)
```

# Algunas reglas generales de sintaxis de R

2. Los espacios suelen ignorarse.

• Todos son equivalentes:

```
a+b  
a + b  
a  + b
```

• Los espacios y las tabulaciones sirven para hacer nuestro código más leíble.

# Algunas reglas generales de sintaxis de R

3. Las expresiones deben estar cerradas.

- Existen diferentes caracteres especiales que marcan el principio y el final de algo, por ejemplo, el principio y el final de una cadena de caracteres o de una expresión:

```
"hello world"
```

```
a * (b + c)
```

```
x[1]
```

- Las sentencias más complejas contienen expresiones anidadas. Las expresiones anidadas se evalúan de dentro a fuera.

```
y[c(1, 3)]
```

- Para cada paréntesis abierta, comilla, etc. debe haber una contrapartida de cierre en el orden correcto. Esto sería incorrecto:

```
y[c(1, 3)]
```

```
## Error: unexpected ']'
```

# Algunas reglas generales de sintaxis de R

## 4. Coma y puntos

Las comas dividen cosas: Principalmente argumentos (parámetros) de funciones.

```
log(x, 5)
```

→ pasa los parámetros x y 5 para calcular el logaritmo de base 5 de x.

• La coma no se puede utilizar para agrupar dígitos en números grandes:

```
population <- 3,350,000  
## Error: unexpected ',' in "population <- 3,"
```

Se utiliza un punto como punto decimal:

```
3.1415
```

# Algunas reglas generales de sintaxis de R

- Si una línea de código R comienza con el símbolo #, no se evalúa. Podemos usar esto para escribir recordatorios de por qué escribimos un código particular. Por ejemplo, en el script anterior, podríamos añadir:

```
## Código para calcular la solución a la  
## ecuación cuadrática de la forma  $ax^2 + bx + c$   
## definir las variables  
a <- 3  
b <- 2  
c <- -1  
  
## ahora calcule la solución  
(-b + sqrt(b^2 - 4*a*c)) / (2*a)  
(-b - sqrt(b^2 - 4*a*c)) / (2*a)
```



# Pimp my R: Instalación y uso de un paquete

The screenshot shows the RStudio IDE interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The main editor window displays a presentation slide titled "RStudio" with the following content:

- RStudio is an **Integrated development environment (IDE)** for R
- it's a comfortable **interface** to R
- analogy: if R is the engine, then RStudio is the car around it
- offers:
  - interactive console
  - script editor with error checking
  - package manager
  - data, plot and file viewers

The bottom right pane shows the "User Library" of installed packages, which is highlighted with a red border. The table lists the following packages:

Name	Description	Version
abind	Combine Multidimensional Arrays	1.4-5
arules	Mining Association Rules and Frequent Itemsets	1.6-1
assertthat	Easy Pre and Post Assertions	0.2.0
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.2
base64enc	Tools for base64 encoding	0.1-3
BH	Boost C++ Header Files	1.66.0-1
bindr	Parametrized Active Bindings	0.1.1
bindrcpp	An 'Rcpp' Interface to Active Bindings	0.2.2
bit	A Class for Vectors of 1-Bit Booleans	1.1-14
bit64	A S3 Class for Vectors of 64bit Integers	0.9-7
bitops	Bitwise Operations	1.0-6
blob	A Simple S3 Class for Representing Vectors of Binary Data ('BLOBS')	1.1.1
broom	Convert Statistical Analysis Objects into Tidy Tibbles	0.5.0
carData	Companion to Applied Regression Data Sets	3.0-1
caret	Classification and Regression Training	6.0-80
caTools	Tools: moving window statistics, GIF, Base64, ROC AUC, etc.	1.17.1
cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0
cli	Helpers for Developing Command Line Interfaces	1.0.0
colorspace	Color Space Manipulation	1.3-2

# Pimp my R: Gestor de paquetes en RStudio

- Los paquetes (también conocidos como "bibliotecas") amplían la funcionalidad de R.
- a la derecha, la pestaña "Paquetes" permite ver, instalar y actualizar paquetes de R desde CRAN
- primera tarea: instalar los siguientes paquetes
  - tidyverse (se trata de un metapaquete que contiene muchos otros paquetes - tardará un rato)
  - swirl (es un paquete para ejercicios interactivos que usaremos más adelante)

# Pimp my R: Consejos y trucos para el gestor de paquetes

- alternativa: usar comando en Consola:
- `install.packages("<NOMBRE_PAQUETE>")`
- luego, para cargar un paquete
- `library(<NOMBRE_PAQUETE>)` (¡sin quotation marks!)

```
install.packages("tidyverse")  
library(tidyverse)
```

# Pimp my R: Consejos y trucos para el gestor de paquetes

- Si olvida cargar un paquete, se encontrará con errores como éstos:

```
qplot()  
## Error in qplot() : could not find function "qplot"  
diamonds  
## Error: object 'diamonds' not found
```

# Saber dónde está R: El concepto de directorio de trabajo

- El directorio o ruta de trabajo es la ubicación en la unidad de disco de su ordenador, en la que su sesión actual de **R** está
  - trabajar, leer archivos, escribir archivos, etc. es relativo a esta ruta
  - Para averiguar la ruta de trabajo actual: `getwd()`
  - establecer la ruta de trabajo: `setwd("<PATH>")`
- **ruta absoluta:** la ruta empieza por / (MacOS / Unix) o C:\.
  - depende de su estructura personal de carpetas
- **ruta relativa:** la ruta comienza directamente con el nombre de un archivo o carpeta
  - relativa a otra ruta, por ejemplo, la ruta de trabajo actual

# Saber dónde está R: Un ejemplo

- Saber dónde está R: `getwd()` devuelve `"/Usuarios/SinNombre/Documentos"`
- el fichero que quieres cargar está en `/Usuarios/SinNombre/Documentos/MiProyecto/datos.csv`
- puede cargar el archivo con: `read.csv("MiProyecto/datos.csv")`
- ¿y si la ruta de trabajo estuviera en...
- `/Usuarios/SinNombre/Documentos/MiProyecto?`
- `/Usuarios/SinNombre/Investigación?`

# Consejos para los nombres de archivos y carpetas

- no utilice espacios (utilice\_esto\_mejor)
- "nombre de archivo gracioso .xlsx" - ¿cuántos espacios cuenta?
- Intente utilizar el alfabeto inglés y evite los caracteres especiales.
- Sea breve.
- Puede distinguir entre mayúsculas y minúsculas.
- Para un mismo proyecto, utilice el mismo directorio raíz para scripts y datos.
- No utilice rutas absolutas en su código → ¡sólo se ejecutará en su ordenador!

# Obtener ayuda

- Usando el sistema de ayuda interno de R:
- `help(<SÍMBOLO>)` / atajo: `?<SÍMBOLO>`
- `<SÍMBOLO>` puede ser cualquier cosa: una función, un paquete, un conjunto de datos
- se muestra en la parte inferior derecha de la pestaña "Ayuda" en Rstudio
- ejemplo: `?getwd` o `?mean`



# Obtener ayuda

The screenshot shows the RStudio interface with a presentation slide titled "Calling the Rescue team: Getting help". The slide content is as follows:

```
296 - do not use spaces (use `` instead)
297 - 'funny file name .xlsx'
298 - try to stick to the English alphabet, avoid special characters
299 - keep it short
300 - case matters!
301 - for a single project, use the same root directory for scripts and data
302 - do not use absolute paths in your code & rarr; it will only run on your computer!
303
304 RStudio file manager
305 =====
306
307 - on the right, "Files" tab
308 - "More" button allows to "Set as Working Directory" and "Go to Working Directory"
309
310 ![RStudio file manager](01intro-figure/rstudio-screenshot-files.png)
311
312 Calling the Rescue team: Getting help
313 =====
314
315 - R's internal help system:
316
317 - 'help(<SYMBOL>)' / shortcut: '?<SYMBOL>'
318 - shown on the right lower side in the "Help" tab in RStudio
319 - example: '?getwd'
320
321
322
```

The bottom right pane shows the R Documentation for `getwd()`, which is highlighted with a red box. The documentation includes the following sections:

- Description**  
`getwd` returns an absolute filepath representing the current working directory of the R process; `setwd(dir)` is used to set the working directory to `dir`.
- Usage**  
`getwd()`  
`setwd(dir)`
- Arguments**  
`dir` A character string: [tilde expansion](#) will be done.
- Value**  
`getwd` returns a character string or NULL if the working directory is not available. On Windows the path returned will use `\\` for the path separator and be encoded in UTF-8. The path will not have a trailing `/` unless it is the root directory (e.g. `C:/`).

# Otras funciones de ayuda útiles

- Mostrar ejemplos de uso: `example(<SIMBOLO>)`

```
example(mean)
## mean> x <- c(0:10, 50)
## mean> xm <- mean(x)
## mean> c(xm, mean(x, trim = 0.10))
## [1] 8.75 5.50
```

- lista todas las funciones disponibles que contienen una palabra clave:  
`apropos("<búsqueda>")`

```
apropos('matrix')
## [1] "anyDuplicated.matrix"      "as.data.frame.matrix" ...
## [4] "as.matrix"                "as.matrix.data.frame" ...
```

# Empecemos de a poco a aprender a interactuar con R

```
library("swirl")  
swirl()  
  
install_course_github("swirldev", "R_Programming_E")
```