

```
library(dplyr)  
library(magrittr)
```

```
Curso_R %>% filter(city == 'Corrientes')
```

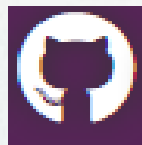
BIENVENIDOS AL TALLER

AprندهR desde Cero

Lunes 26 de agosto de 2019.



@patriloto



PatriLoto



Hoy hablamos sobre...



PARTE 1: Breve Repaso

¿Cómo instalamos un paquete?

¿Cómo importamos datos?

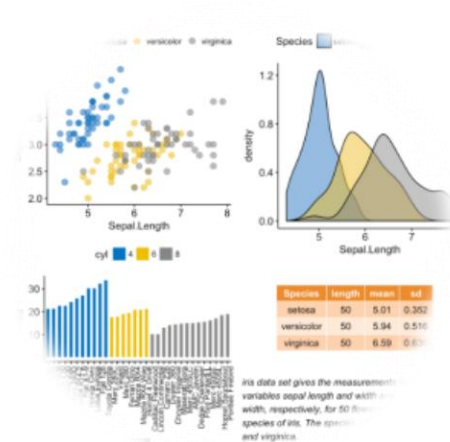
Visualización

Importancia de la visualización

Paquete ggplot2

Elementos de la visualización

Introducción a gráficos interactivos
con plotly



PARTE 2: #Datosdemiercoles

- Presentación de #Datosdemiercoles
- ¿Cómo Participo?

Hands-on con ggplot2

Practicamos con:

Dataset propina

Dataset gapminder

Dataset datos

PARTE 1

IMPORTACIÓN DE DATOS

Recomendaciones para trabajar más efectivamente con R.

- Hacete un gran favor y trabaja con proyectos, es por tu bien.
- Utiliza control de versiones, te vas a ahorrar mucho tiempo y además tendrás todo actualizado. No sucumbas a las copias con diferentes nombres version1, version1_2, version1esteSi, version1ultimo.

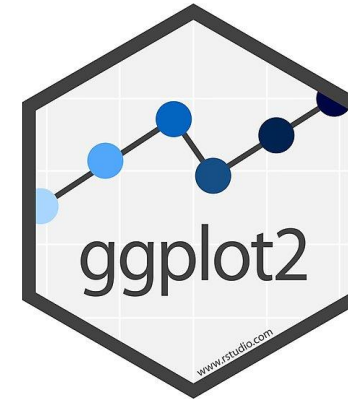


Breve Repaso

¿Cómo instalamos un paquete?



- **install.packages**("nombre_del_paquete")
- **library** (nombre_del_paquete)
- **install.packages** ("ggplot2")
- **library**(ggplot2)



Importación de datos en R

Para archivos con formato csv → package readr()

```
tiempo_pantalla <- readr::read_csv("E:/DATASETS/tiempo_pantalla.csv")
```

```
View(tiempo_pantalla)
```



Para archivos con formato txt en R → utilizamos el paquete read.table.

Ejemplo:

```
mtcars <- read.table("E:/DATASETS/mtcars.txt")
```

```
mtcars <- read.table("E:/DATASETS/mtcars.txt", header=TRUE)
```

```
View(mtcars)
```



Importación de datos en R

Para archivos con formato (.xls and .xlsx) → utilizamos el paquete **readxl**.

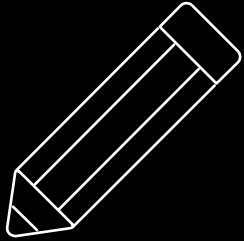
Ejemplo:

- `install.packages("readxl")`
- `library(readxl)`
- `estadis <- read_xlsx("estadistica2009.xlsx")`





PARTE 2



Visualización con ggplot2



Premisas para visualización

Una imagen vale más que mil palabras pero...



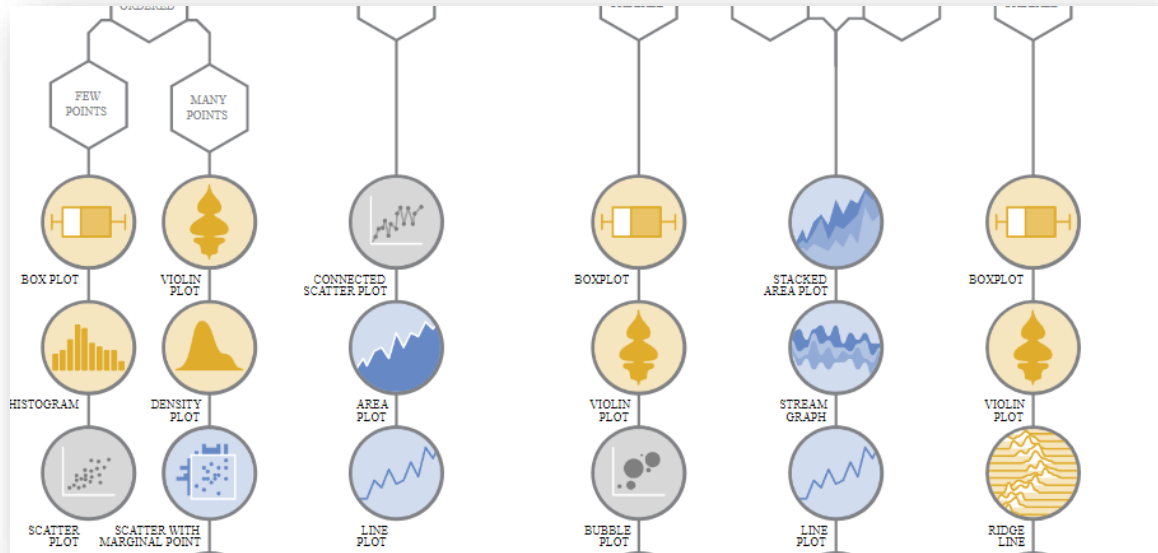
- ✓ No sólo interesa la belleza de los gráficos sino también es importante la funcionalidad.

Premisas para visualización

- ✓ Sé perseverante: la única manera de aprender es experimentando y fallando, experimentando y fallando y así sucesivamente, hasta obtener aquel gráfico que comunique efectivamente tus datos.
- ✓ Trabaja de manera incremental, comienza por un pequeño gráfico y luego en cada iteración mejóralo.

Premisas para visualización

- Es muy útil, seguir en twitter a gente de la comunidad que se dedica a lo que vos querés aprender.



Yan Holtz: @R_Graph_Gallery

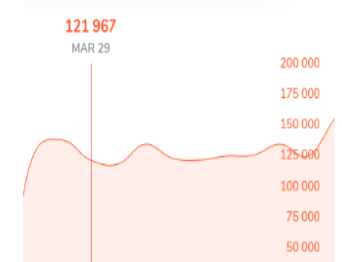
<https://www.data-to-viz.com/>

Data, Code and Visualization

April 8, 2019

When charts are integrated in the web page

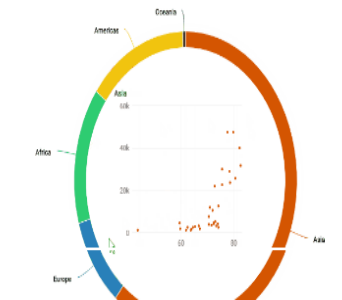
I saw a chart yesterday, a chart without frame, without title or captions just a very hidden chart at bottom of the web page. It will be a nice time to get the most interesting visual features of it.



Feb. 4, 2019

Using tooltips in unexpected ways

Highcharts tooltips are one of most underrated features. In the tooltips you can easily render html, that means we can put images, tables even charts. So the possibilities are almost limitless.



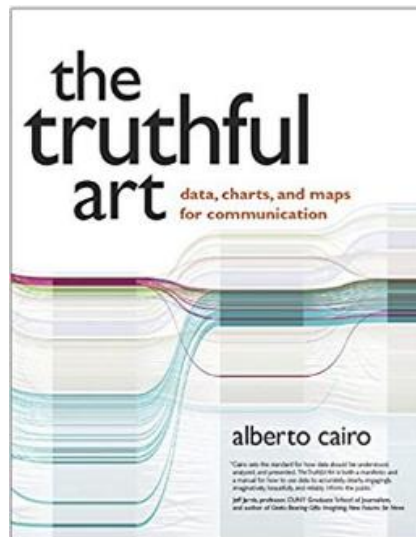
Joshua Kunst: @jbkunst

<http://jkunst.com/blog/>

¿Para qué aprendemos visualización?

- **Para aprender** cómo usar y transmitir los datos de manera precisa, clara, atractiva, imaginativa, bella y confiable con el objetivo de informar al público, a nuestro público.

Alberto Cairo, escritor del libro The Truthful Art.



Cualidades de grandes visualizaciones

- Que sea agradable a la vista
- Que sea funcional
- Que muestre hallazgos
- Que esté basada en datos confiables, es decir, que transmita la verdad
- Que sea llamativa

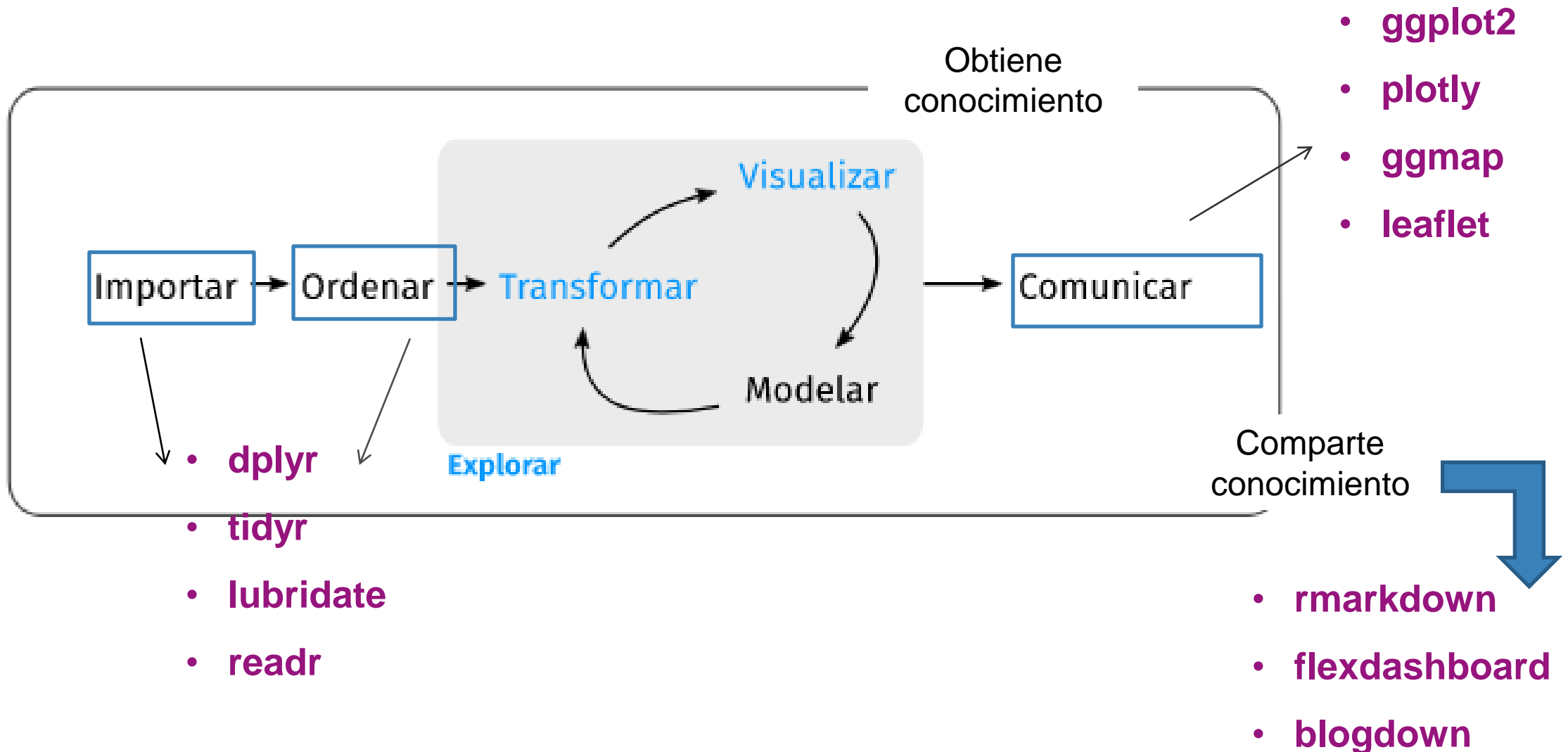
Alberto Cairo, escritor del libro The Truthful Art.

Visualizaciones efectivas

Para que una visualización sea fácilmente entendible:

- Pensá siempre en el público objetivo
- ¿Qué deseas mostrar con tu gráfico?
- Resaltá la información más importante
- Proporcioná contexto a los datos
- Usa el color de forma inteligente

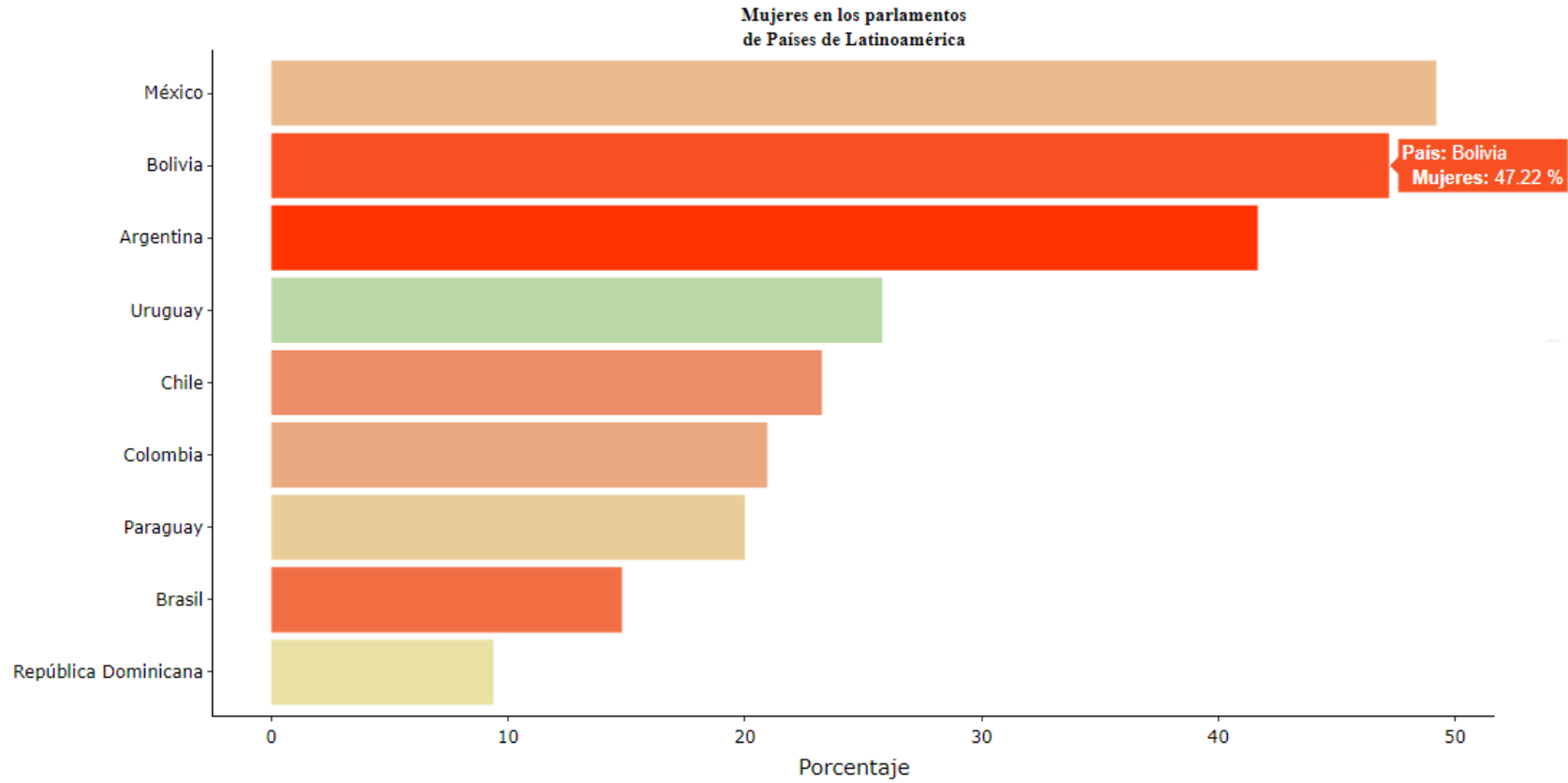
La visualización dentro del esquema de trabajo en ciencia de datos



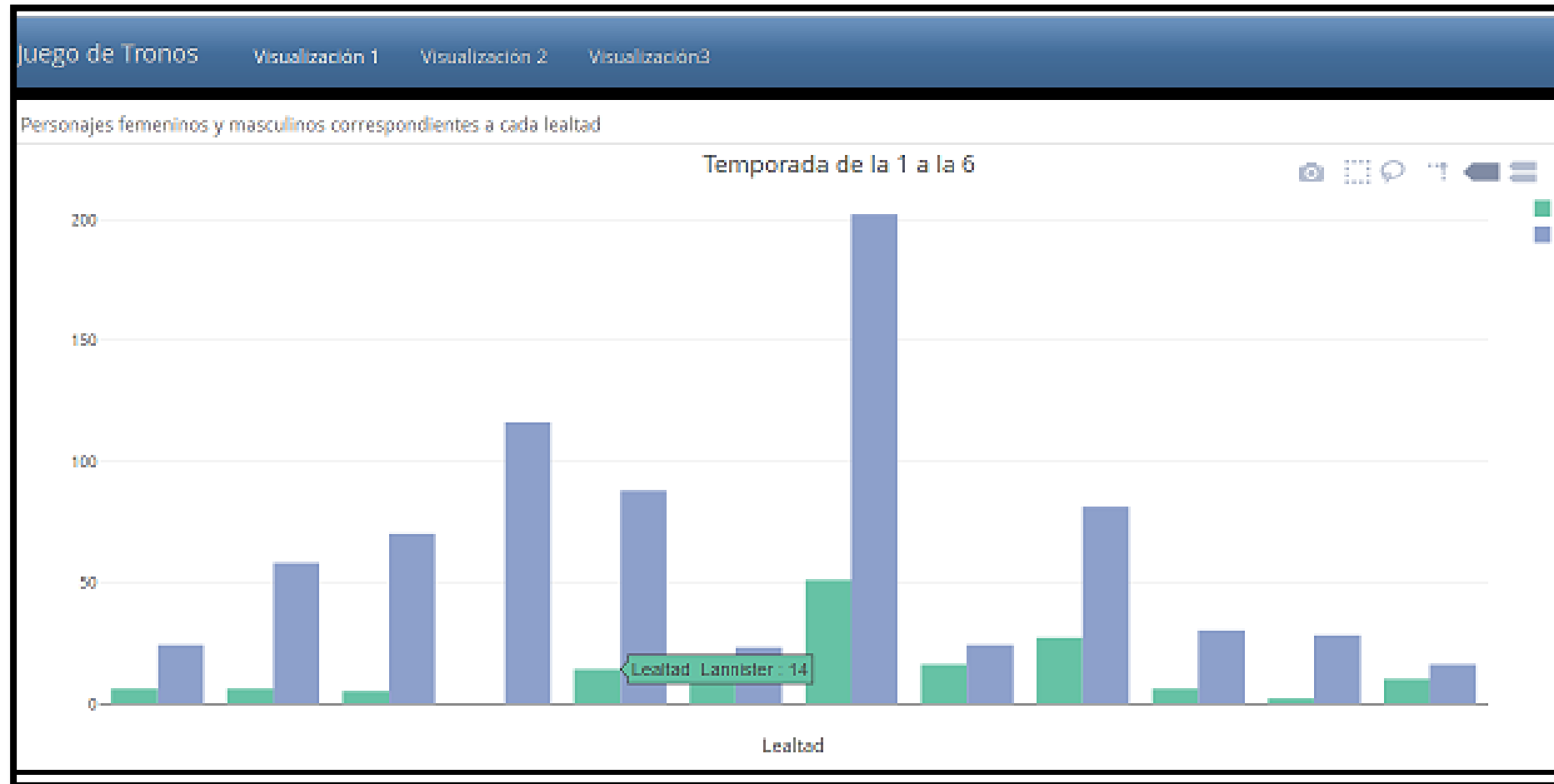
Más alla de ggplot2

Paquetes Similares:

-  plotly



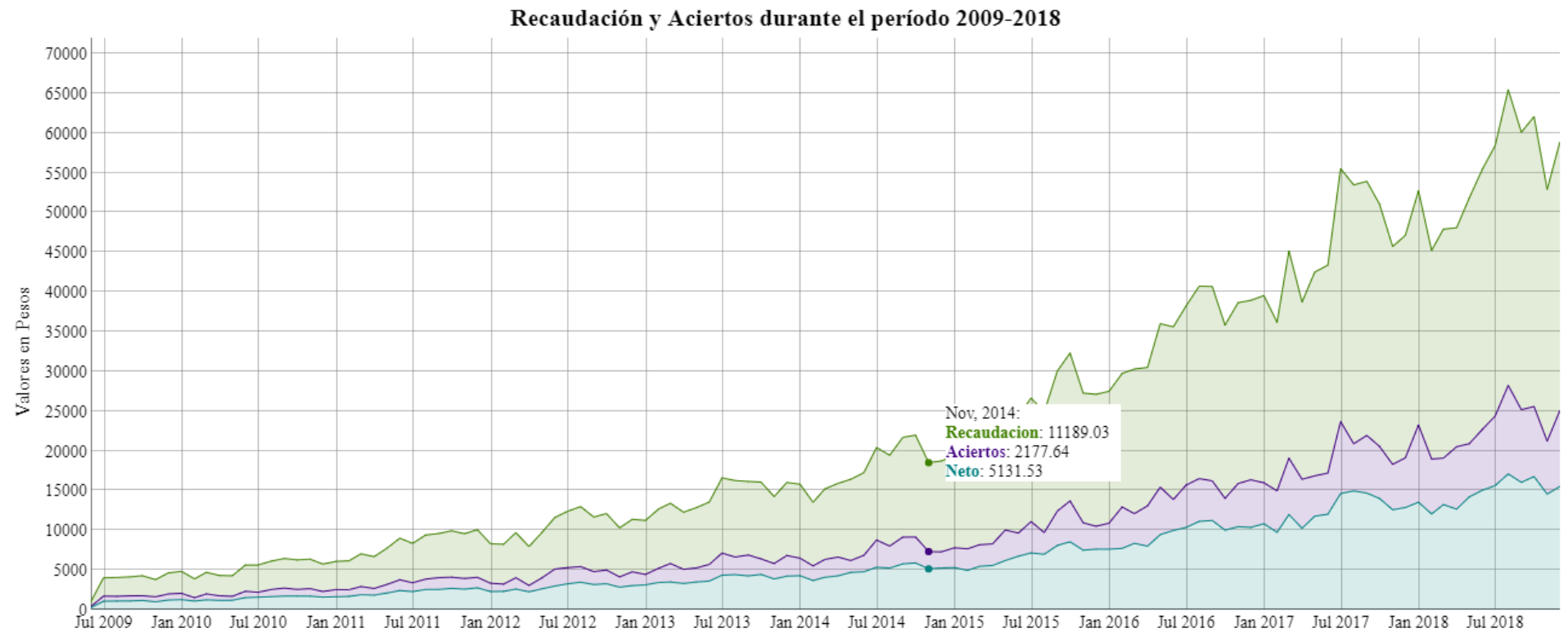
Más alla de ggplot2



Más alla de ggplot2

Series temporales:

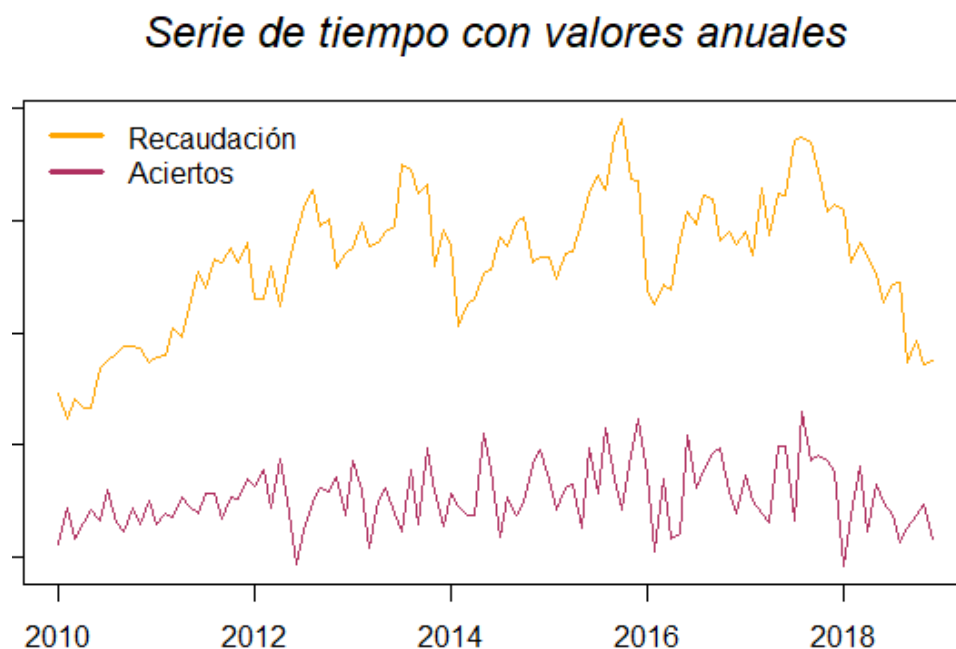
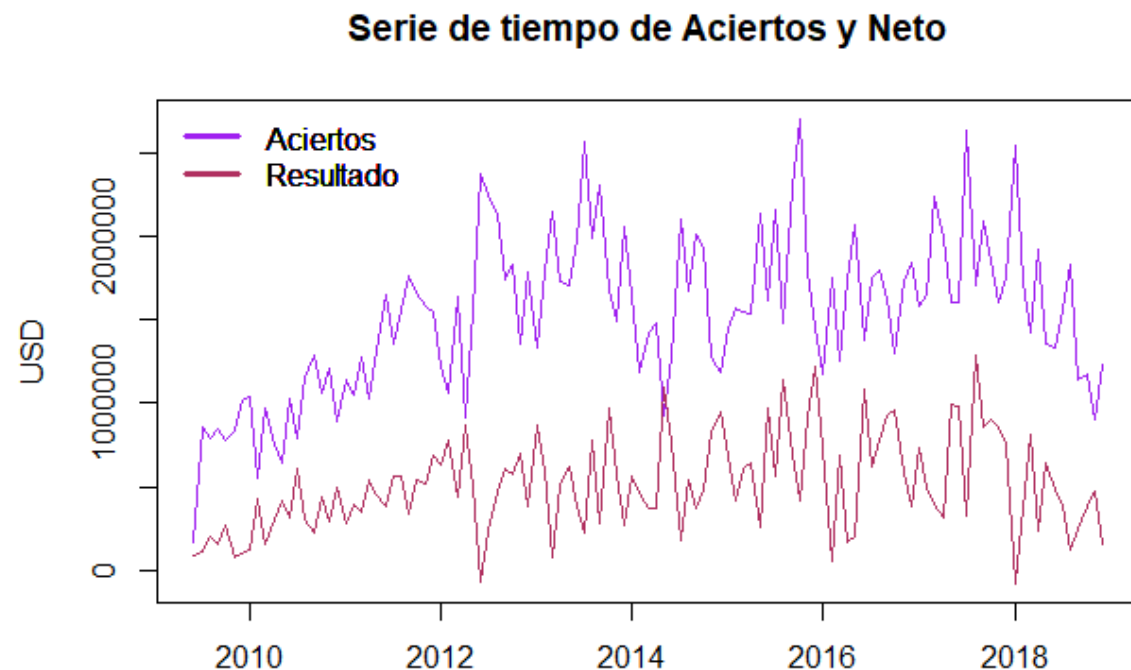
- dygraphs
- forecast



Más alla de ggplot2

Series temporales:

- ts

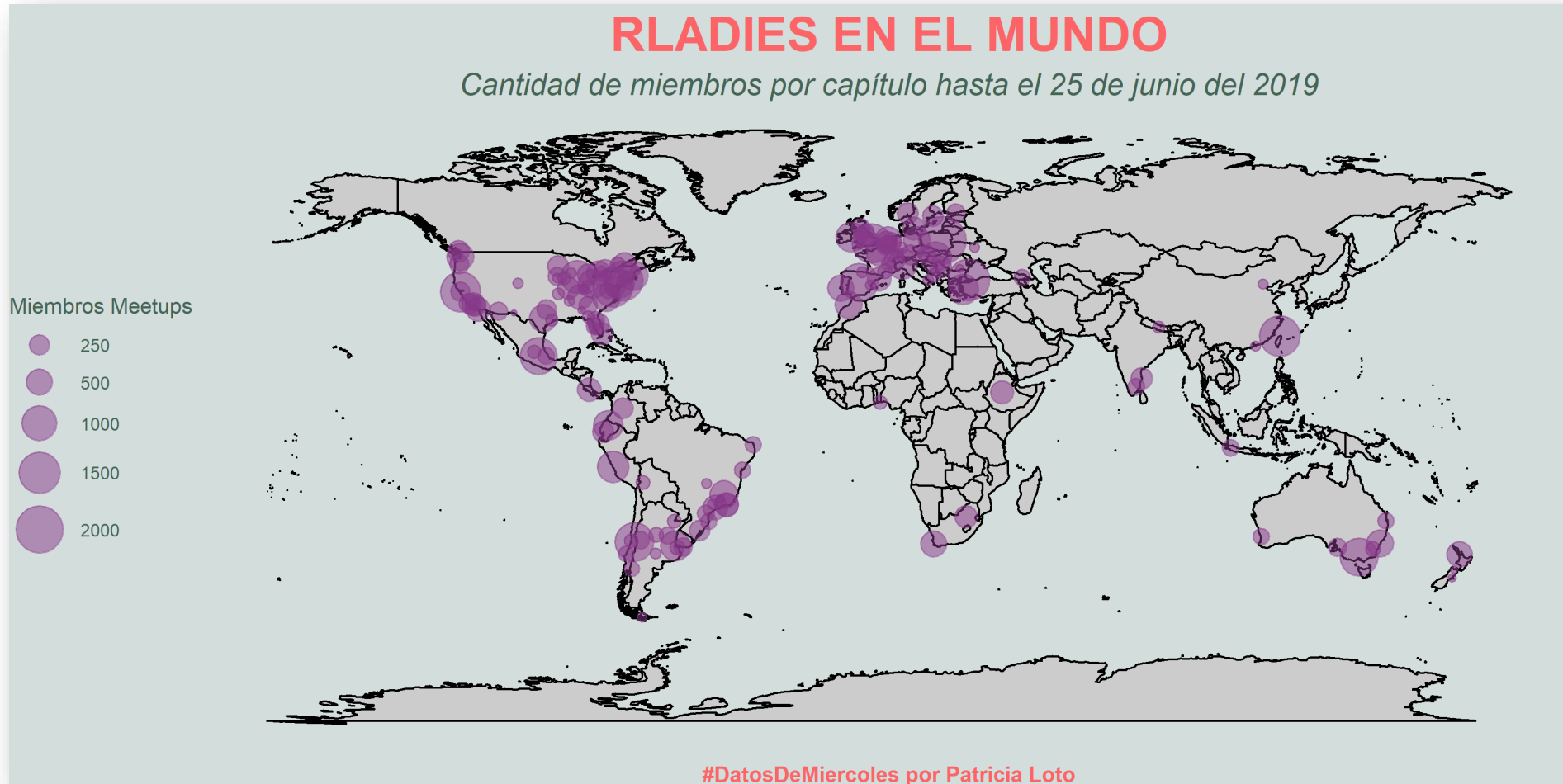


Periodo: 2010-2018

Con la función **seqplot.ts** del paquete **ts**

Más alla de ggplot2

Mapas con leaflet y ggmap



¿Porqué usar el ggplot2?



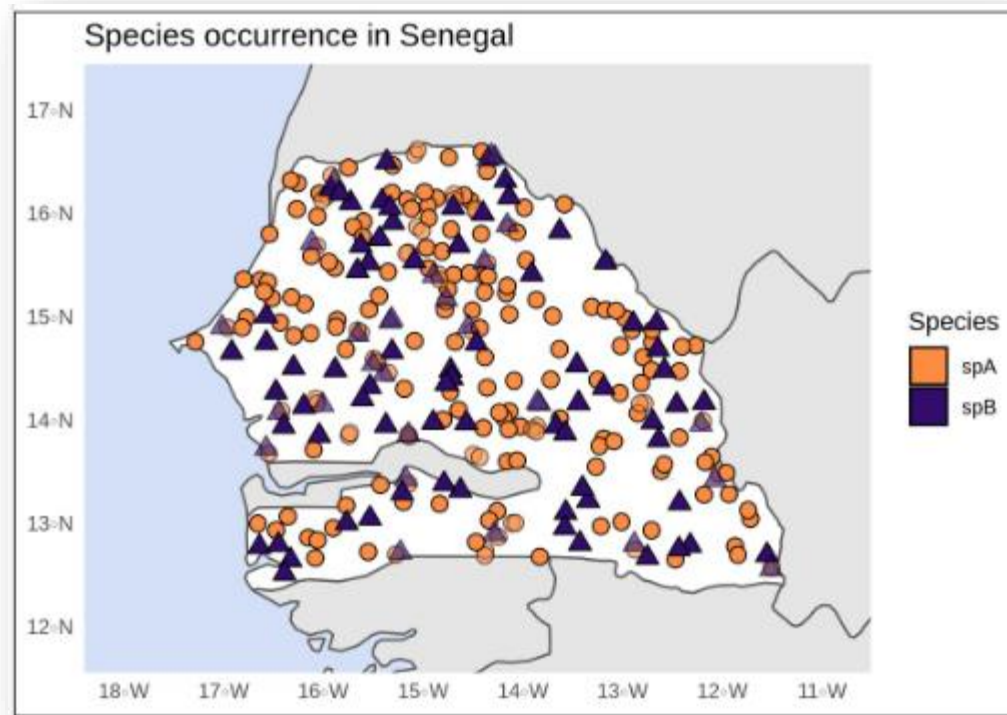
- **ggplot2** Es uno de los paquetes más populares para visualización de datos dentro de la comunidad R.
- Fue desarrollado por Hadley Wickham.
- Es parte de un conjunto de paquetes que tiene foco en la ciencia de datos llamado *Tidyverse*.

Gramática de gráficos o *Grammar of Graphics*

- La Gramática de los gráficos nos responde las preguntas:
- ¿Qué es un gráfico?
- ¿Cómo describir un gráfico?
- ¿Cómo crear un gráfico?

Gramática de gráficos

- Un gráfico es un mapeo de los datos a atributos estéticos de objetos geométricos.



Gramática de gráficos - Componentes

- **Datos (Data):** con mapeos (aes) entre los datos y los elementos gráficos
- **Capas (Layers):** elementos geométricos (geoms, son puntos, líneas, rectángulos, etc.) y transformaciones estadísticas (identidades, promedio, mediana, conteos, etc.)
- **Escalas (Scales):** mapea valores en el espacio de los datos a valores en el espacio estético (ej. color, tamaño, forma o posición)
- **Coord (coordenadas):** normalmente Cartesianas, pero por ej; polares.
- **Facets (facetas):** como se arregla el display cuando son muchos gráficos
- **Theme (temas),** items para mejorar el gráfico como fuente, tamaño, color, background.

Instalar ggplot2

```
install.packages ("ggplot2")  
library (ggplot2)
```

**Ventaja: descargamos la
última versión**

```
install.packages ("devtools")  
library (devtools)  
install_github ("tidyverse/ggplot2")
```

Versión en desarrollo: <https://github.com/tidyverse/ggplot2>

Ejemplo: set de datos de propina

##	total	propina	sexo	fuma	dia	momento	cantidad
##	<dbl>	<dbl>	<chr>	<chr>	<chr>	<chr>	<int>
## 1	17.0	1.01	F	No	Do	Noche	2
## 2	10.3	1.66	M	No	Do	Noche	3
## 3	21.0	3.5	M	No	Do	Noche	3
## 4	23.7	3.31	M	No	Do	Noche	2
## 5	24.6	3.61	F	No	Do	Noche	4
## 6	25.3	4.71	M	No	Do	Noche	4

Tres componentes de todo gráfico

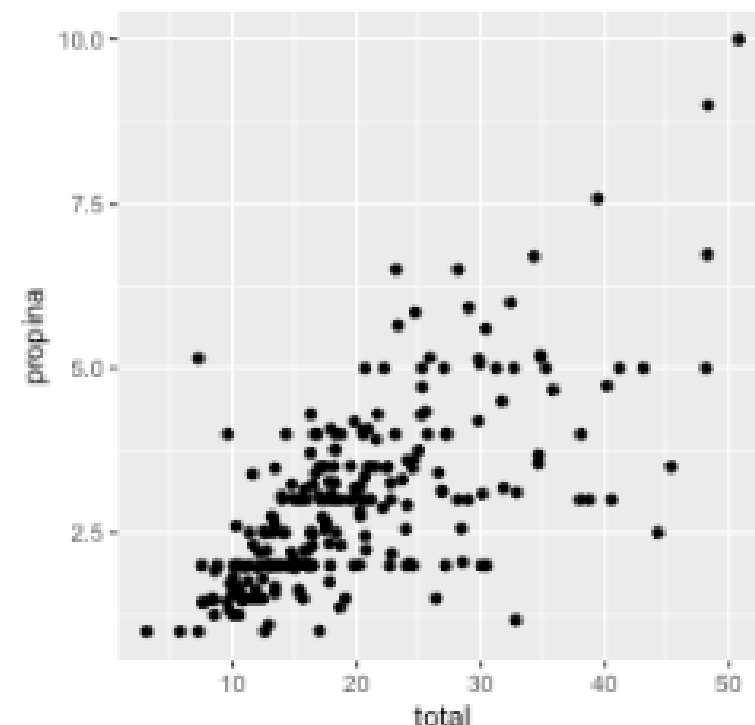
1. **Datos (data):** nuestro set de datos.
2. **Atributos estéticos (aes)** : un conjunto de mapeos estéticos entre las variables en los datos y las propiedades visuales.
3. **Capas (layers):** al menos una capa que describe cómo representar cada observación. Las capas son creadas con la función **geom**.

Tres componentes de todo gráfico:

Veamos un ejemplo

- **data:** `propina`
- **aes:** `total` mapeado a la posición x , `monto de propina` mapeado a la posición y.
- **layer:** puntos mediante `geom_point`.

```
ggplot(data = propinas, aes(x = total, y = propina)) -  
  geom_point() +  
  theme(aspect.ratio = 1)
```



¿Qué hay atrás de este gráfico?

- Cada observación está representada como un punto, cuya posición está de acuerdo a dos variables (posición horizontal y vertical: x e y).
- Cada punto puede tener tamaño, color y forma estos atributos son denominados elementos estéticos (aesthetics aes)
- Los aes son propiedades que pueden ser percibidas en el gráfico, cada aes puede ser mapeado a una variable o fijado en un valor constante
- **total** es mapeado a la posición horizontal, **propina** a la posición vertical y forma al color. Tamaño y forma no son mapeados a variables (valor por defecto)

Color, tamaño, forma y otros atributos estéticos

- Para agregar otras variables al gráfico podemos usar otros **aes** o **componentes estéticos** como color, forma o tamaño:

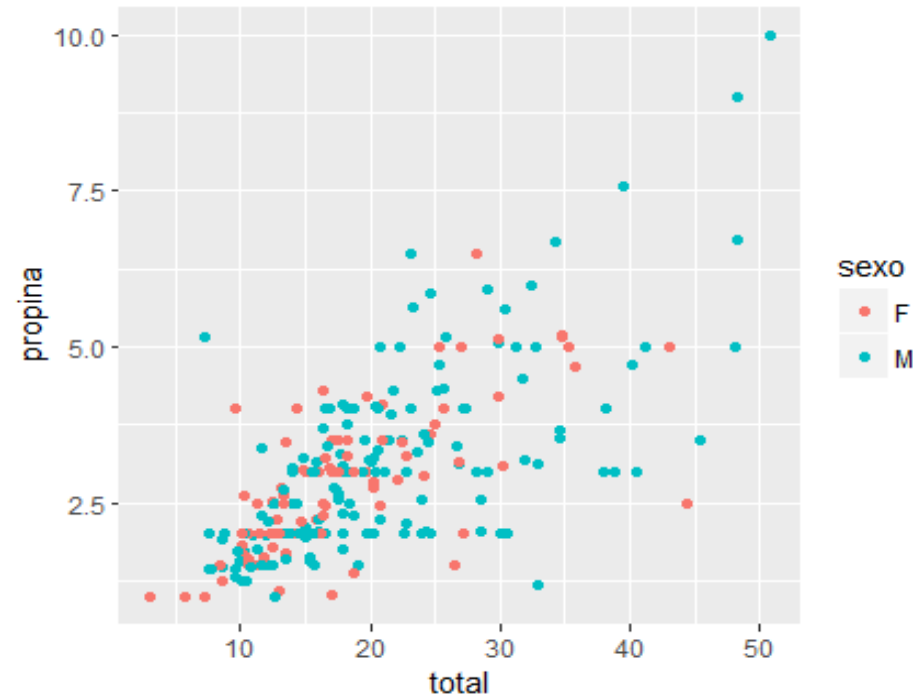
```
aes(x = total, y = propina, colour = sexo)
```

```
aes(x = total, y = propina, shape = sexo)
```

```
aes(x = total, y = propina, size = cantidad)
```

Ejemplo de la aes Color

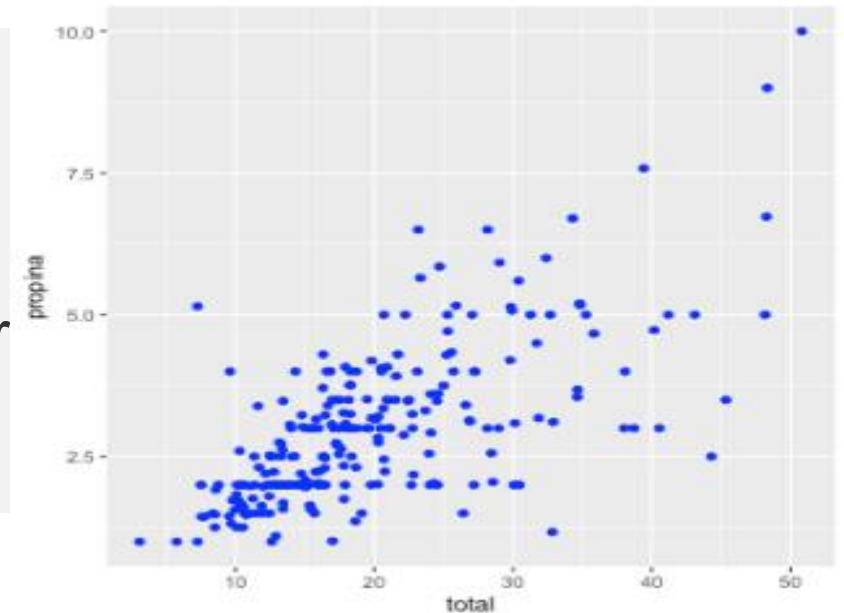
```
ggplot(data = propinas, aes(x = total, y = propina, colour = sexo)) +  
  geom_point() + theme(aspect.ratio = 1)
```



Color fijo

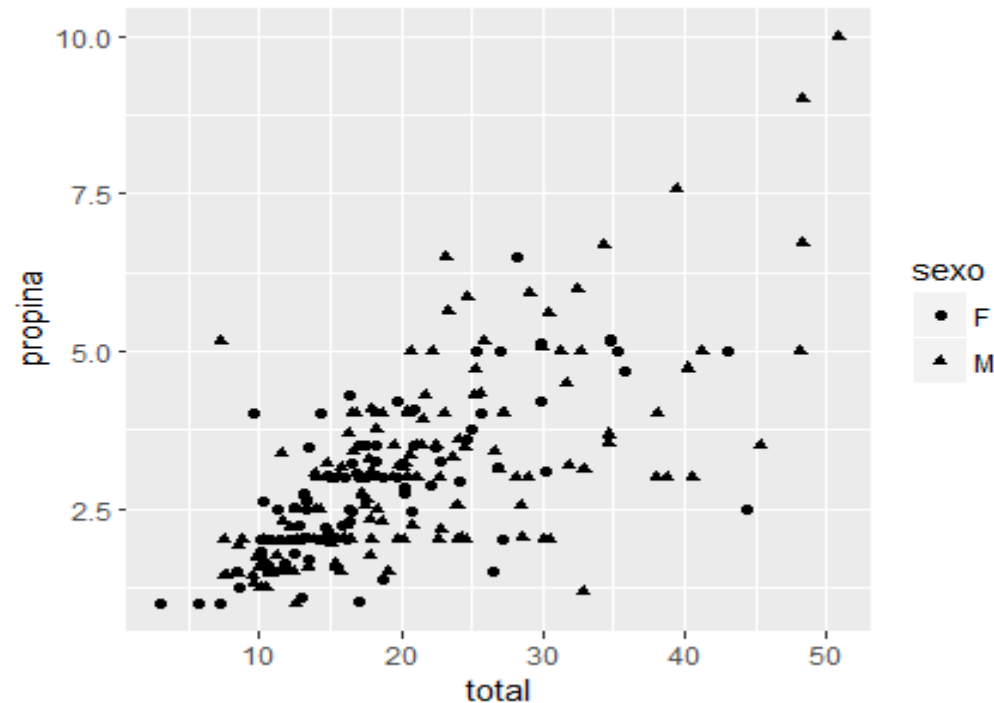
- Si queremos fijar un elemento estético a un valor fijo hay que hacerlo en la capa de afuera sin aes o usar I('blue') en aes

```
ggplot(data = propinas,  
aes(x = total, y = propina) )+  
  geom_point(colour = "blue") + theme(aspect.r  
= 1)
```



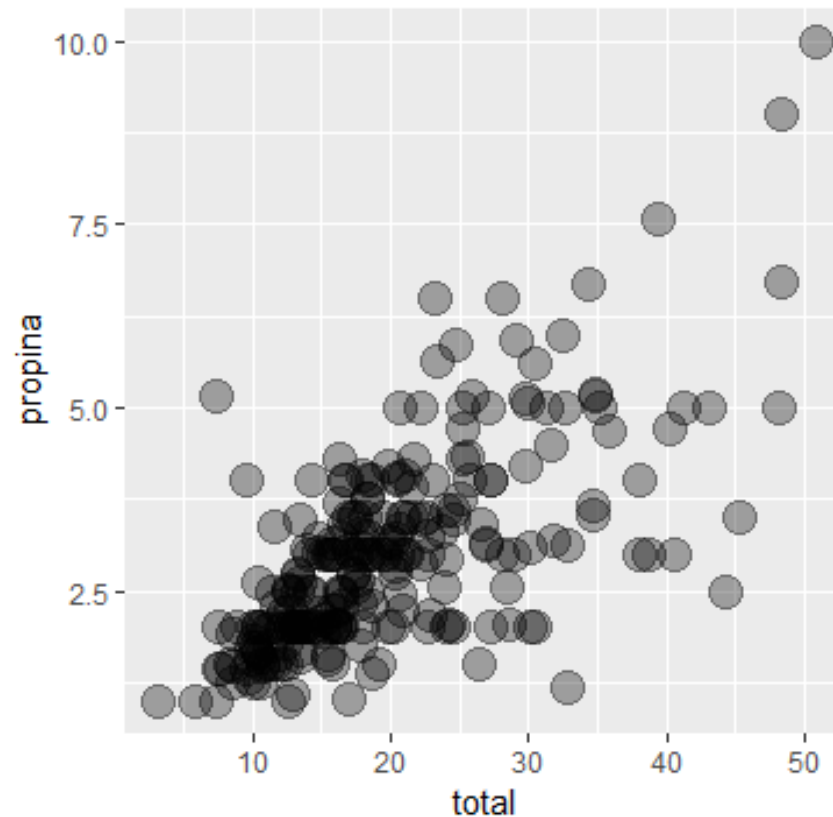
Ejemplo de la aes Forma

- `ggplot(data = propinas, aes(x = total, y = propina, shape = sexo)) +
geom_point() + theme(aspect.ratio = 1)`



Ejemplo de la aes Tamaño

- `ggplot(data = propinas, aes(x = total, y = propina)) +
 geom_point (size = 5, alpha = 1 / 3) + theme(aspect.ratio = 1)`



Escalas (scales)

- Los datos pueden ser convertidos desde unidades de datos como moneda, sexo, día, etc. a unidades gráficas como color, forma, tamaño, etc. que se pueden representar en la computadora
- La conversión se llama escalado y es realizada por scales o escalas.
- Cada color es representado por una cadena de seis letras y números, cada size por números y cada shape por enteros.
- Estas especificaciones estéticas que tienen sentido para R se describen en `vignette("ggplot2-specs")`

Escalas (scales)

- El Mapeo estético indica que una variable debe ser mapeada a un elemento estético pero no dice cómo debe ser mapeada.
- Cuando mapeo una variable a shape usando `aes(shape = x)` no se indica la forma específica (shape) que debe tomar.
- Cuando uso `aes(color = z)` no se especifica de que color debe ser.
- Para describir el color, la forma o el tamaño (colour, shape, size) deseado se hace mediante transformaciones en scale.

Gramática de gráficos

```
ggplot(propinas, aes(x= total, y= propina, colour = fuma)) +  
  geom_point()
```



Capas de un gráfico

- Los datos, mapeos estéticos, objetos geométricos y las transformaciones estadísticas forman una **capa**.
- Podemos tener un gráfico con muchas capas.

La gramática de capas define componentes de un gráfico:

- datos y conjunto de mapeos de variables a elementos estéticos
- una o más capas, cada capa tiene un elemento geométrico, una transformación estadística y una posición.

Capas de un gráfico

ggplot() +

layer(

data = propinas, mapping = aes(x = total, y = propina),

geom = "point", stat = "identity", position = "identity") +

scale_x_continuous() +

scale_y_continuous() +

coord_cartesian()

Capas de un gráfico

Equivalente a

```
ggplot(data = propinas, aes(x = total, y = propina)) +  
geom_point()
```

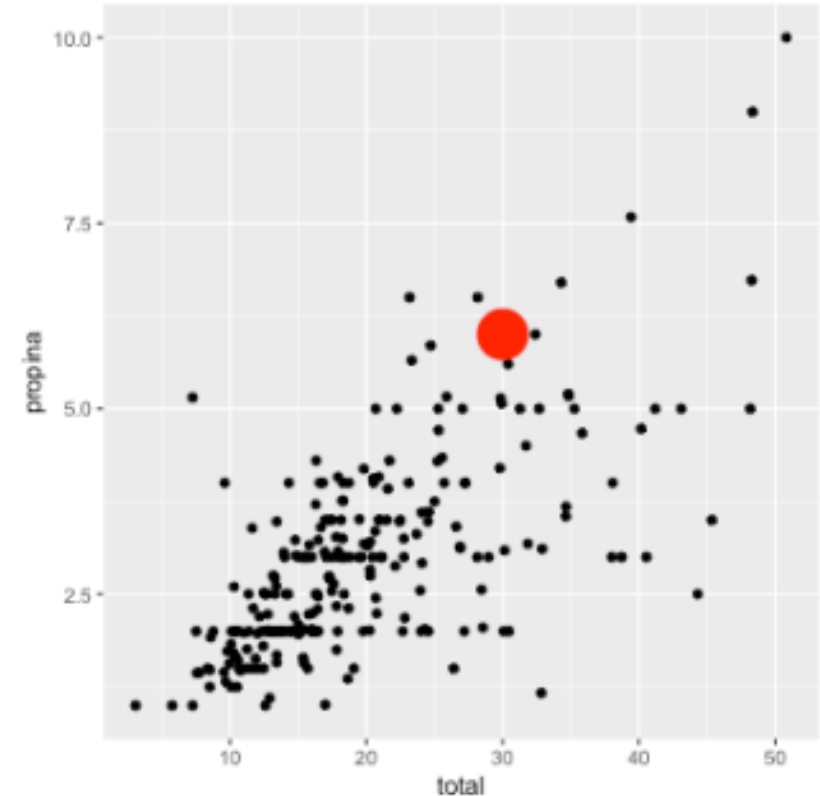
Capas de un gráfico

ggplot() +

geom_point (data = propinas, aes(x = total, y = propina)) +

geom_point(data = data.frame(x = 30, y = 6), aes(x, y),

color = "red", size = 10)

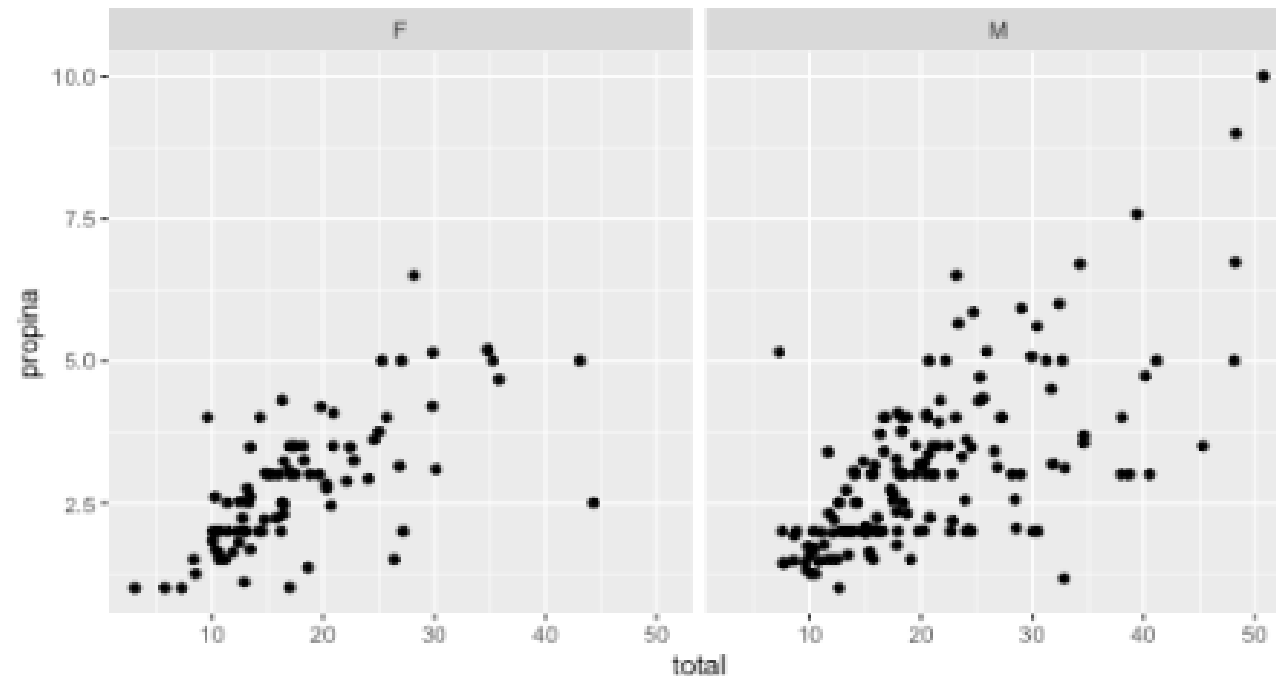


Paneles

- Se pueden desplegar variables categóricas adicionales en un gráfico particionando el panel.
- Crea tablas de gráficos partiendo los datos en subconjuntos y mostrándolos el mismo gráfico para cada subconjunto.
- Dos tipos: `facet_grid` y `facet_wrap`

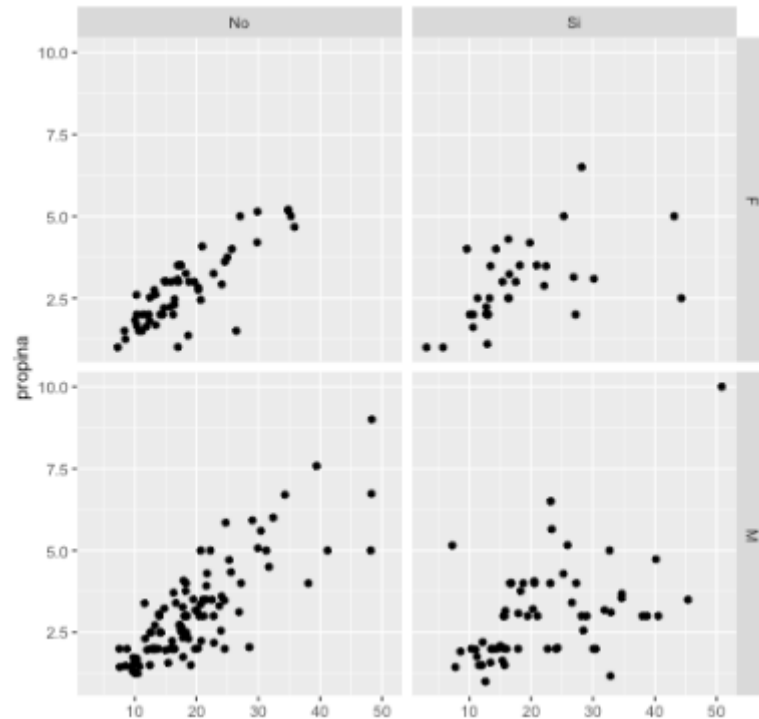
Facetas: facet_wrap

```
ggplot(data = propinas, aes(x = total, y = propina)) +  
  geom_point() + theme(aspect.ratio = 1) +  
  facet_wrap( ~sexo)
```



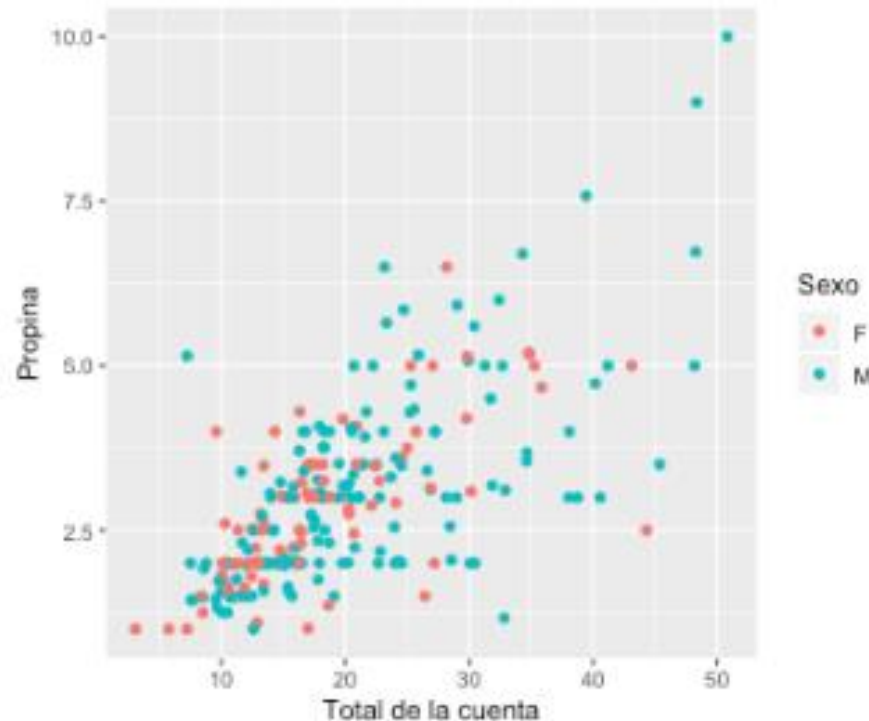
Facetas: facet_grid

```
ggplot(data = propinas, aes(x = total, y = propina)) +  
  geom_point() + theme(aspect.ratio = 1) +  
  facet_grid(sexo ~ fuma)
```



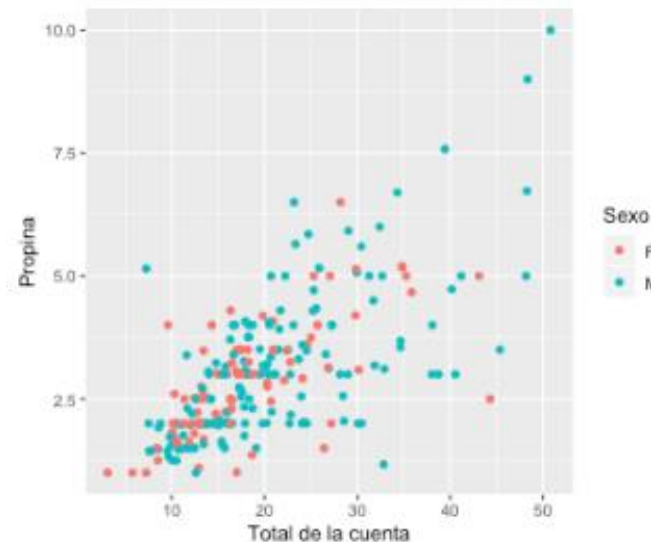
Rótulos (labs) – Opción 1

```
ggplot(data = propinas, aes(x = total, y = propina, color = sexo)) +  
  geom_point() + theme(aspect.ratio = 1) +  
  labs (x = "Total de la cuenta", y = "Propina", color = "Sexo")
```



Incluir rótulos (*labs*) - Opción 2

```
ggplot(data = propinas, aes(x = total, y = propina, colour = sexo)) +  
  geom_point() + theme(aspect.ratio = 1) +  
  scale_x_continuous (name = "Total de la cuenta") +  
  scale_y_continuous (name = "Propina") +  
  scale_color_discrete ("Sexo")
```



Temas: *Theme*

- **theme** en un gráfico permite controlar los elementos que no son datos en el mismo.
- Te ayuda a que tu gráfico sea estéticamente como lo querés sin afectar el mapeo de datos.
- theme te da control sobre **fuentes, tamaño, color, background**.

COMUNIDAD DE APRENDIZAJE ORIENTADA POR PROYECTO



**“R for Data Science”
en español**

#Datosdemiércoles - Un proyecto semanal de datos en R

Datos de miércoles es la versión en español de #tidytuesday, un proyecto semanal de datos organizado por la comunidad de R, que busca que sus participantes desarrollen sus habilidades procesando datos para crear gráficos significativos usando ggplot2, tidyr, dplyr y otras herramientas del tidyverse. El proyecto original surgió en el marco de la comunidad de aprendizaje online R4DS y desde abril de 2019 la comunidad sudamericana de R impulsa una versión local en español.

MAS INFO #LATINR2018



Sitio Web: <http://latin-r.com/>



#LatinR2019

https://twitter.com/LatinR_Conf
[conf](#)



[https://github.com/LatinR/presentaciones-](https://github.com/LatinR/presentaciones-LatinR2018)
[LatinR2018](#)

Bibliografía Gratuita Online

- R4DS de Hadley Wickham, existe la versión en inglés y actualmente se está traduciendo la versión en español.
- Link al proyecto en inglés: <https://r4ds.had.co.nz/>
- Link al proyecto en español: <https://es.r4ds.hadley.nz/> - <https://es.r4ds.hadley.nz/index.html> (índice con nombres)
- R Graphics Cookbook de Winston Chang: <http://www.cookbook-r.com/>
- Fundamentals of Data Visualization de Claus O. Wilke. Link: <https://serialmentor.com/dataviz/>

Bibliografía Online Gratuita

- Data analytics and visualization de Yan Holtz. Link: <https://www.yan-holtz.com/teaching>
- R para profesionales de los datos: una introducción de Carlos J. Gil Bellosta. Link: https://www.datanalytics.com/libro_r/index.html
- Fundamentals of Data Visualization de Claus O. Wilke. Link: <https://serialmentor.com/dataviz/>
- Libro ggplot2 Book: <https://github.com/hadley/ggplot2-book>.

Material de ayuda

- **Más geoms**

<http://ggplot2.tidyverse.org/reference/>

- **Extensiones**

Hay 40 extensiones de ggplot2

<http://www.ggplot2-exts.org/gallery/>

- **ggplot2 ayuda**

Lista de mails: <http://groups.google.com/group/ggplot2>

stackoverflow: <http://stackoverflow.com>

Referencias

- <http://natydasilva.com>
- https://natydasilva.github.io/taller_LatinR
- Libro: <https://github.com/hadley/ggplot2-book>.

