






Visualización de datos con ggplot2

Ronny A. Hernández Mora.

 @RonnyHdezM

 ronnyhdez

 <http://ronnyhdez.rbind.io/>

Materiales del curso

README.md

¡Bienvenido al curso de ciencia de datos con R!

Este es un curso libre y gratis que puede usar para dar sus primeros pasos con el lenguaje de programación R.



Artwork by @allison_horst

Materiales del curso

Sesión	Presentación	Video
0-Preparación	No hay	https://www.youtube.com/watch?v=NCvJwJSMq60
1- Introducción a las herramientas	Por subir	Por subir



https://github.com/ronnyhdez/curso_ciencia_datos_r

main 4 branches 0 tags

Go to file Add file **Code**

ronnyhdez Merge pull request #11 from ronnyhdez/T8

- img Ref #1 estructura del curso
- presentaciones Ref #8 material presentacion
- sesion_01 Ref 1 material sesion 1
- sesion_02 Ref #8 orden en script segunda ses
- .gitignore Ref #2 materiales sesion 2

Clone

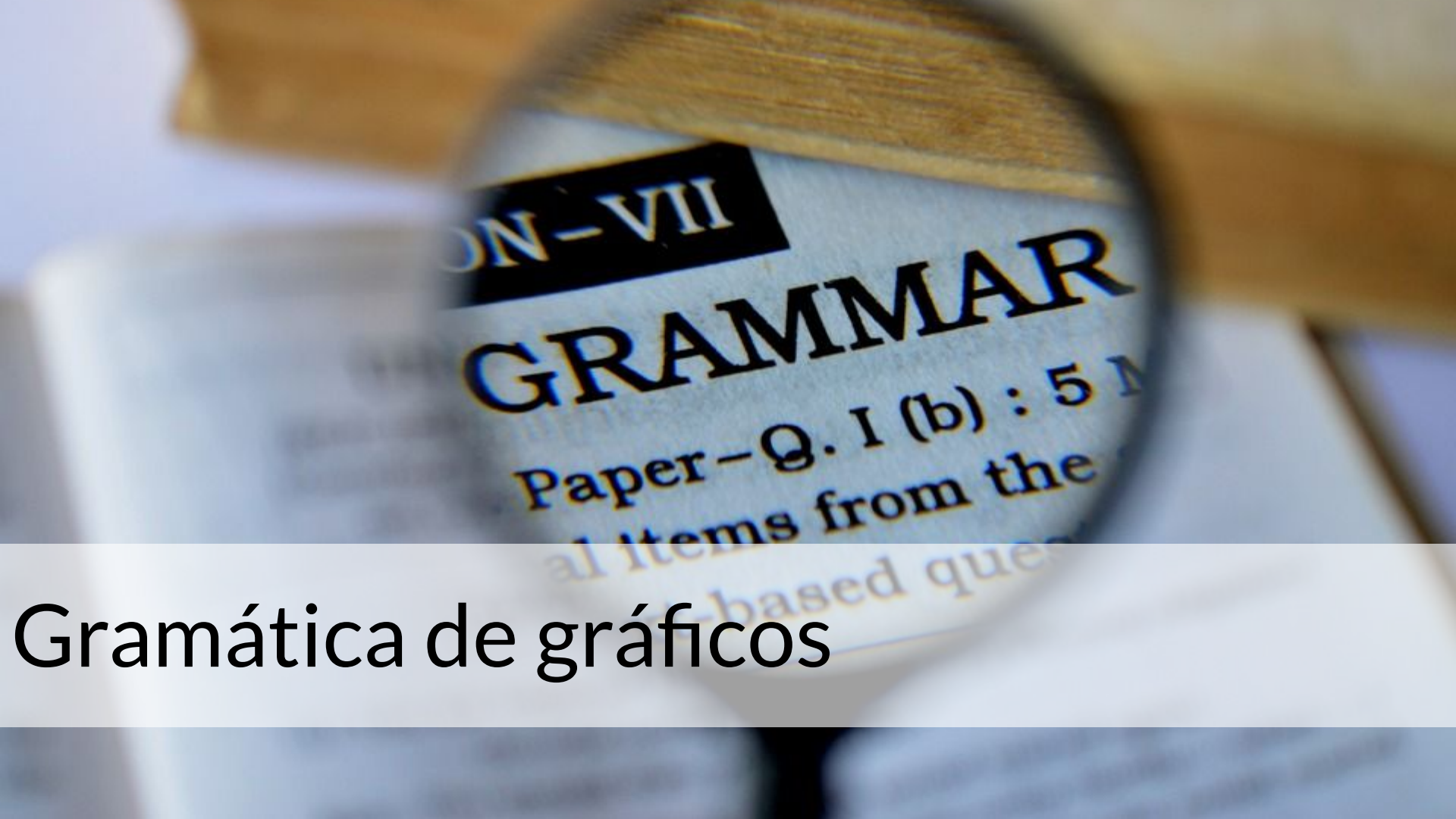
HTTPS SSH GitHub CLI

https://github.com/ronnyhdez/curso_cie

Use Git or checkout with SVN using the web URL.

Download ZIP

10 days ago



Gramática de gráficos

Hoja de referencia

Data Visualization with ggplot2 : CHEAT SHEET



Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data set**, a **coordinate system**, and **geoms**—visual marks that represent data points.



To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.



Complete the template below to build a graph.

```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),  
  stat = <STAT>, position = <POSITION>) +  
  <COORDINATE_FUNCTION> +  
  <SCALE_FUNCTION> +  
  <THEME_FUNCTION>
```

Not required, sensible defaults supplied

ggplot(data = mpg, aes(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

aesthetic mappings data geom
qplot(x = cty, y = hwy, data = mpg, geom = "point") Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

last_plot() Returns the last plot

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5 x 5 file named "plot.png" in working directory. Matches file type to file extension.

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

GRAPHICAL PRIMITIVES

```
a <- ggplot(economics, aes(date, unemploy))  
b <- ggplot(seals, aes(x = long, y = lat))  
  
a + geom_blank() # Useful for expanding limits  
b + geom_curve(aes(yend = lat + 1, xend = long + 1, curvature = 1), x, yend, y, alpha, angle, color, curvature, linetype, size)  
a + geom_path(linetype = "butt", linejoin = "round", linewidth = 1)  
x, y, alpha, color, group, linetype, size  
a + geom_polygon(aes(group = group))  
x, y, alpha, color, fill, group, linetype, size  
b + geom_rect(aes(xmin = long, ymin = lat, xmax = long + 1, ymax = lat + 1), xmin, ymin, xmax, ymax, alpha, color, fill, linetype, size)  
a + geom_ribbon(aes(lmin = unemploy - 900, lmax = unemploy + 900), x, y, ymax, ymin, alpha, color, fill, group, linetype, size)
```

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size
b + geom_abline(aes(intercept = 0, slope = 1))
b + geom_hline(aes(yintercept = lat))
b + geom_vline(aes(xintercept = long))
b + geom_segment(aes(yend = lat + 1, xend = long + 1))
b + geom_spoke(aes(angle = 1:1155, radius = 1))

ONE VARIABLE continuous

```
c <- ggplot(mpg, aes(hwy))  
c2 <- ggplot(mpg)  
  
c + geom_area(stat = "bin")  
x, y, alpha, color, fill, linetype, size  
c + geom_density(kernel = "gaussian")  
x, y, alpha, color, fill, group, linetype, size, weight  
c + geom_dotplot()  
x, y, alpha, color, fill  
c + geom_freqpoly(x, y, alpha, color, group, linetype, size)  
c + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight  
c2 + geom_qq(aes(sample = hwy)) x, y, alpha, color, fill, linetype, size, weight
```

discrete

```
d <- ggplot(mpg, aes(fit))  
d + geom_bar() x, alpha, color, fill, linetype, size, weight
```

TWO VARIABLES

```
continuous x, continuous y  
e <- ggplot(mpg, aes(cty, nudge_x = 1, merge_y = 1, check_overlap = TRUE), x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust  
e + geom_jitter(height = 2, width = 2) x, y, alpha, color, fill, shape, size, stroke  
e + geom_point(x, y, alpha, color, fill, shape, size, stroke)  
e + geom_quantile(x, y, alpha, color, group, linetype, size, weight)  
e + geom_rug(sides = "bl", x, y, alpha, color, linetype, size)  
e + geom_smooth(method = lm, x, y, alpha, color, fill, group, linetype, size, weight)  
e + geom_text(aes(label = cty, nudge_x = 1, merge_y = 1, check_overlap = TRUE), x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust)
```

discrete x, continuous y

```
f <- ggplot(mpg, aes(class, hwy))  
f + geom_col(x, y, alpha, color, fill, group, linetype, size)  
f + geom_boxplot(x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight)  
f + geom_dotplot(binaxis = "y", stackdir = "center", x, y, alpha, color, fill, group)  
f + geom_violin(scale = "area", x, y, alpha, color, fill, group, linetype, size, weight)
```

discrete x, discrete y

```
g <- ggplot(diamonds, aes(cut, color))  
g + geom_count() x, y, alpha, color, fill, shape, size, stroke
```

THREE VARIABLES

```
sealsSz <- with(seals, sqrt(delta_long^2 + delta_lat^2))  
i <- ggplot(seals, aes(long, lat))  
i + geom_contour(aes(z = z)) x, y, z, alpha, color, group, linetype, size, weight  
i + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE) x, y, alpha, fill  
i + geom_tile(aes(fill = z), x, y, alpha, color, fill, linetype, size, width)
```

continuous bivariate distribution

```
h <- ggplot(diamonds, aes(carat, price))  
h + geom_bin2d(binwidth = c(0.25, 500)) x, y, alpha, color, fill, linetype, size, weight  
h + geom_density2d() x, y, alpha, color, group, linetype, size  
h + geom_hex() x, y, alpha, color, fill, size
```

continuous function

```
i <- ggplot(economics, aes(date, unemploy))  
i + geom_area() x, y, alpha, color, fill, linetype, size  
i + geom_line() x, y, alpha, color, group, linetype, size  
i + geom_step(direction = "hv") x, y, alpha, color, group, linetype, size
```

visualizing error

```
df <- data.frame(r = c("A", "B"), fit = 4.5, se = 1.2)  
j <- ggplot(df, aes(r, fit, ymin = fit - se, ymax = fit + se))  
j + geom_crossbar(latten = 2) x, y, ymax, ymin, alpha, color, fill, group, linetype, size  
j + geom_errorbar(x, y, ymax, ymin, alpha, color, group, linetype, size, width (also geom_errorbarh))  
j + geom_linerange(x, y, ymin, ymax, alpha, color, group, linetype, size)  
j + geom_pointrange(x, y, ymin, ymax, alpha, color, fill, group, linetype, size)
```

maps

```
data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))  
map <- map_data("state")  
k <- ggplot(data, aes(fill = murder))  
k + geom_map(aes(map_id = state), map = map) x, y, alpha, color, fill, linetype, size
```

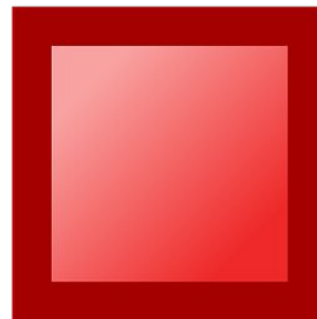
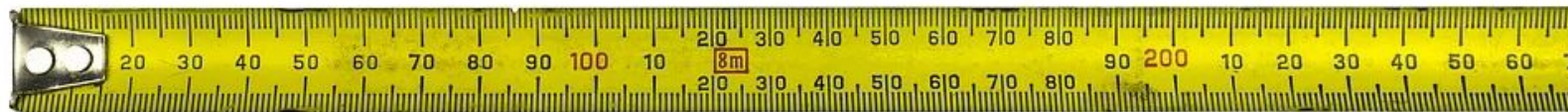


```
ggplot( datos, aes(ejes) ) +  
  geom_*( )
```

datos +
mapeo variables +
geometría +
elementos extra



Tomar en cuenta:





Café 20:08 a 20:18

Para aprender más

 README.md 

Curso Profesional Visualización de datos con R

Este repositorio contiene el material usado durante el curso. Todos los ejemplos así como la presentación están aquí contenidos.



Data Latam

<https://github.com/datalatam/visualizacion>

Programación en C++
Algoritmos, estructuras de datos y objetos

L. Joyanes Aguilar

EI385

Learning Python

FIFTH EDITION

Lutz

LINUX BASICS FOR HACKERS

O'REILLY

Fundamentals of Data Visualization

Wilke

O'REILLY

Mastering Ubuntu Server
Third Edition

Jay LaCroix

<|

R for Data Science

John Fox
John Fox

O'REILLY

THINK LIKE A PROGRAMMER

SPRAT



The Pragmatic Programmer

20th ANNIVERSARY
EDITION



Zuur • Ieno • Walker
Saveliev • Smith



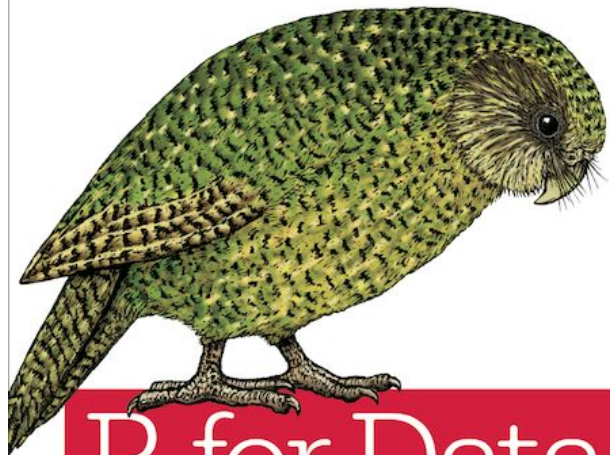
Mixed Effects Models
Extensions in Ecology with R

Wichham

Advanced R

Recursos

O'REILLY®



R for Data Science

VISUALIZE, MODEL, TRANSFORM, TIDY, AND IMPORT DATA


Hadley Wickham &
Garrett Grolemund

<https://es.r4ds.hadley.nz/>



¡Gracias !

Ronny A. Hernández Mora.

 @RonnyHdezM

 ronnyhdez

 <http://ronnyhdez.rbind.io/>

ronny.hernandezm@gmail.com