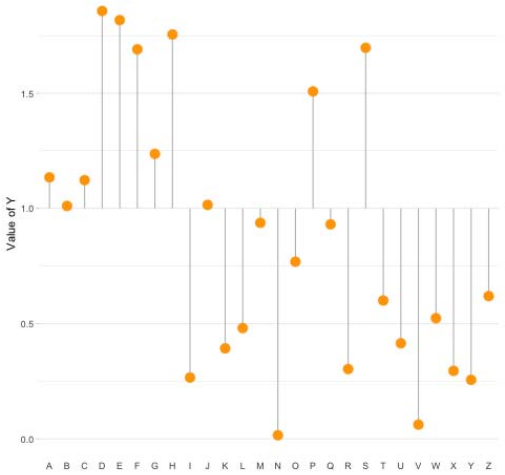
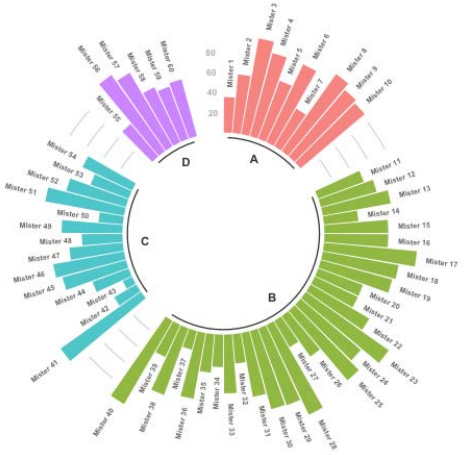
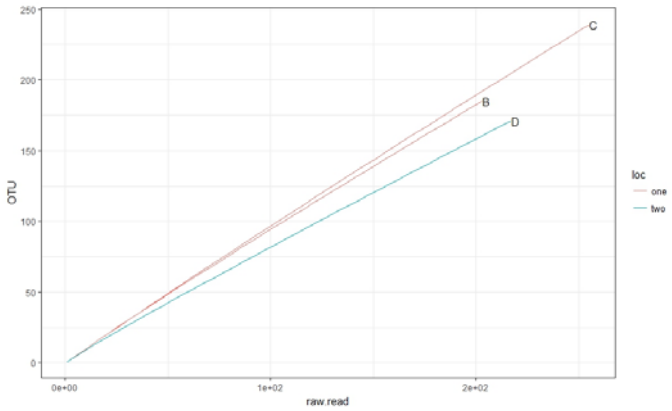
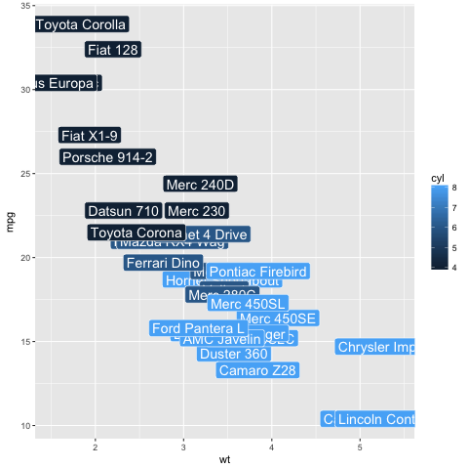
```
library(dplyr)

rladies_global %>%
  filter(city == 'Córdoba')
```

Visualización de datos
ggplot2


Value of Y
1.5
1.0
0.5
0.0
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z


A
B
C
D


OTU
raw.read
0e+00
1e+02
2e+02
C
B
D
loc
one
two


mpg
wt
cyl
Toyota Corolla
Fiat 128
Fiat X1-9
Porsche 914-2
Merc 240D
Datsun 710
Merc 230
Ferrari Dino
Pontiac Firebird
Merc 450SL
Merc 450SE
Duster 360
Camaro Z28
Chrysler Imp
Lincoln Cont

¿Qué es Ggplot2?

- Es un paquete para visualización de datos
- Creado y desarrollado por Hadley Wickham
- Inspirado en el Grammar of Graphics de Leland Wilkinson

¿Por qué ggplot2?

- Definido en base a un conjunto de principios
- Estructura consistente independientemente del tipo de gráfico
- Puede producir gráficos de calidad publicable en poco tiempo
- Diseñado para trabajar iterativamente agregando capas, agrupa y faceta

Otras maneras de visualizar gráficos:

- R base
- Lattice

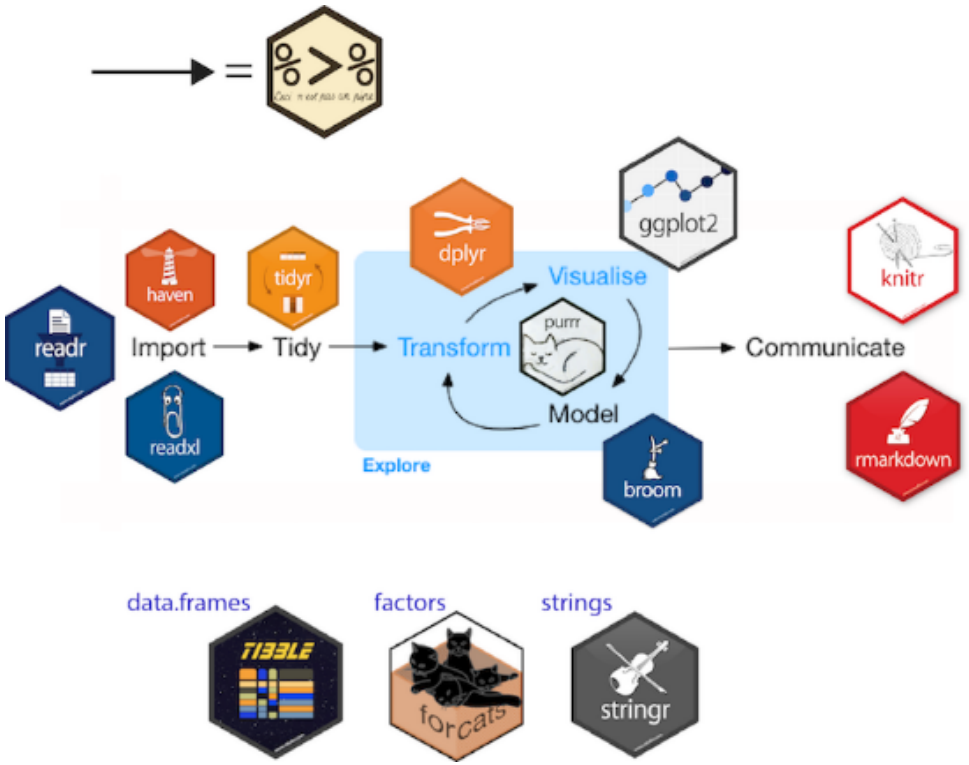
actualmente favorece a ggplot2:

- es el paquete de gráficos estáticos mas prominente de R
- Hay mucha ayuda disponible
- Están desarrolladas muchas extensiones (aprox. 40)

Como empezar

install.packages(“ggplot2”) o install.packages(“tidyverse”)

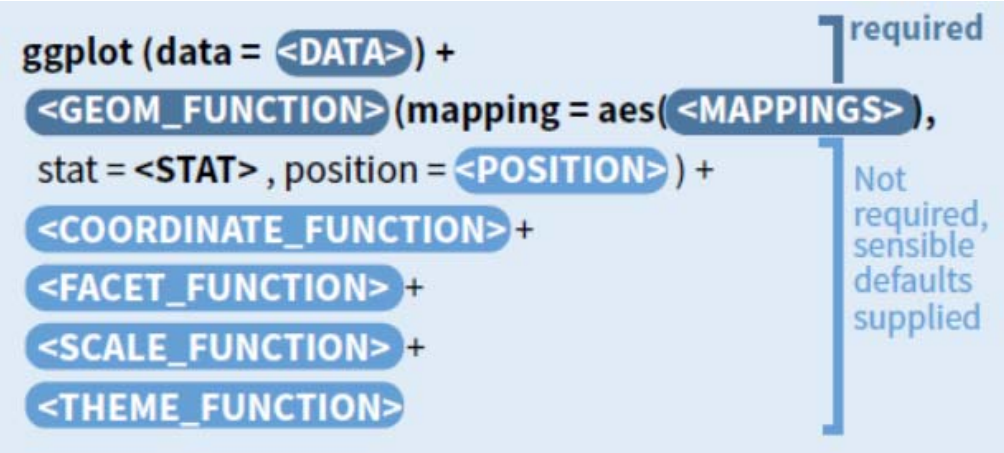
ggplot2 actualmente es parte del grupo de paquetes del “Tidyverse”



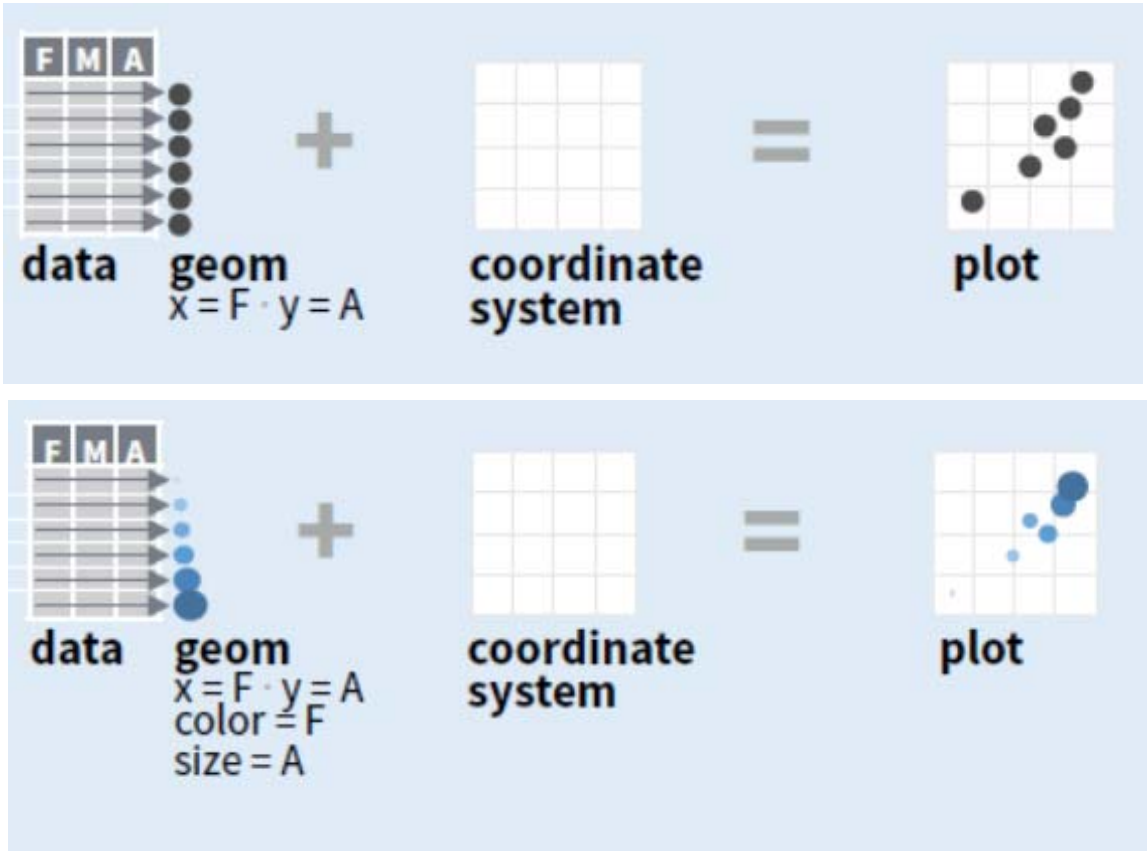
Elementos básicos en visualización en ggplot2

Datos
Mapeos estéticos
Objecto geométrico

Transformaciones estadísticas
Escalas
Sistemas de coordenadas
Ajustes de posición
Facetas
Temas



Los datos deben estar almacenados en un data.frame
Importante: datos ordenados (cada variable en su columna, cada observación en su fila, cada valor en su celda)



Como se construye un gráfico con ggplot2

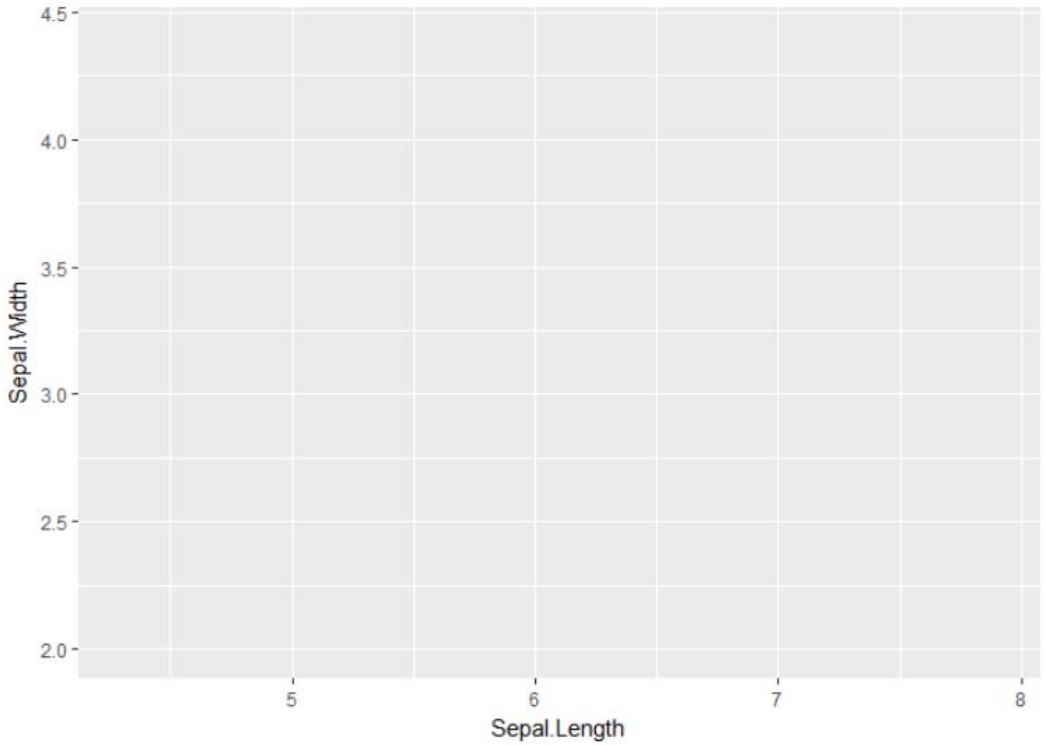
1. Especificar los datos con los que se van a trabajar

```
ggplot(data=iris)
```

Como se construye un gráfico con ggplot2

2.Especificar el mapeo de las variables a componentes “estéticos”

```
ggplot(data=iris, aes(x=Sepal.Length, y= Sepal.Width))
```

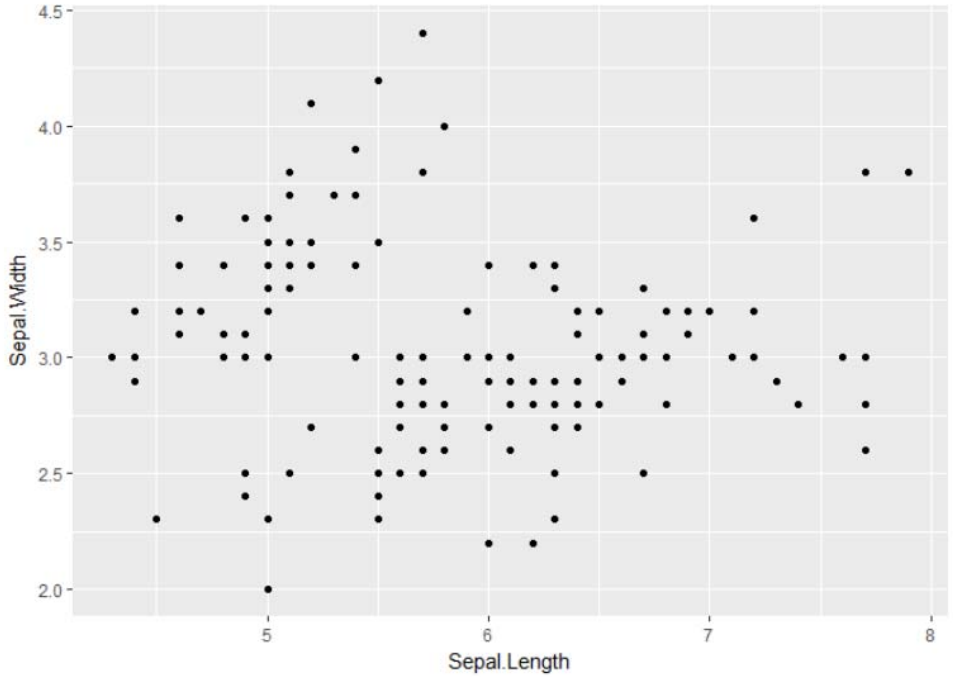


componentes estéticos:

- X
- Y
- Color
- Fill
- Alpha
- Linetype
- Shape
- Size

Como se construye un gráfico con ggplot2

3.Añadir capa: especificar el tipo de objeto geométrico
ggplot(data=iris, aes(x=Sepal.Length, y= Sepal.Width)) + geom_point()



Si el proceso básico es….

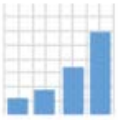
Data

Aes()

Geom_

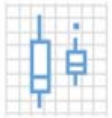
Geom_

- **geom_bar()**



d + geom_bar()
x, alpha, color, fill, linetype, size, weight

- **geom_boxplot()**

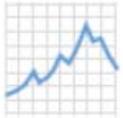


f + geom_boxplot(), x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

- **geom_histogram()**

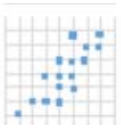
c + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight

- **geom_line():**



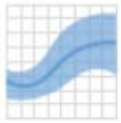
i + geom_line()
x, y, alpha, color, group, linetype, size

- **geom_point():**



e + geom_point(), x, y, alpha, color, fill, shape, size, stroke

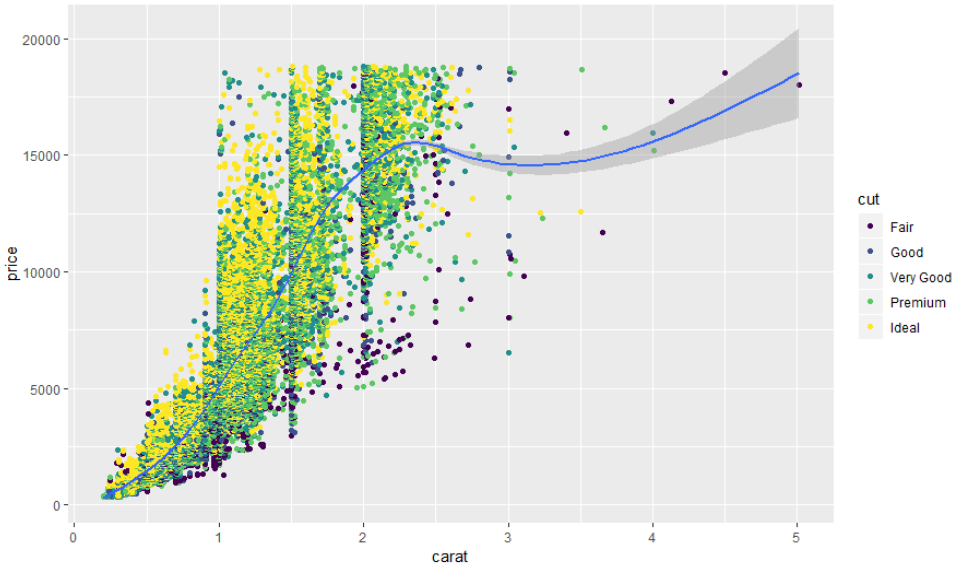
- **geom_smooth():**



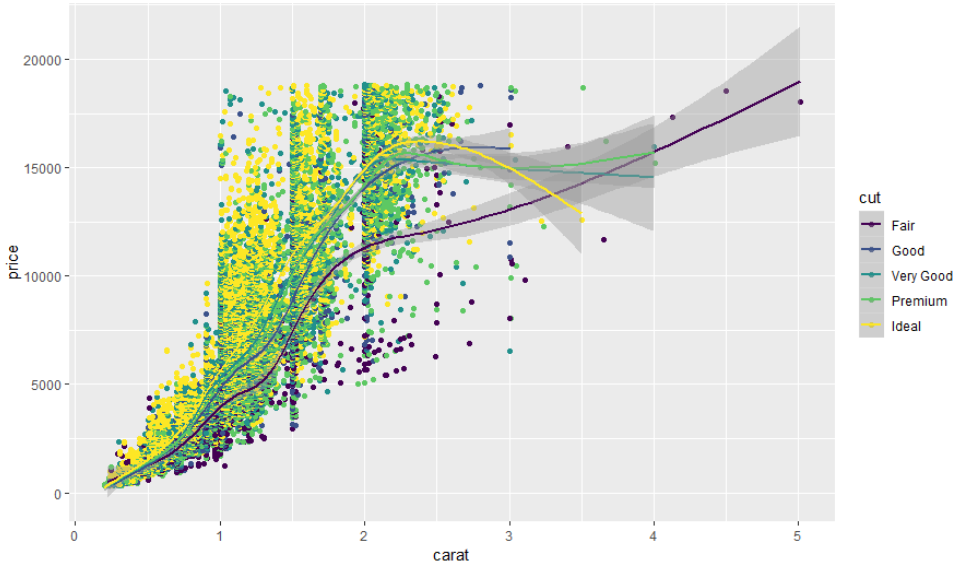
e + geom_smooth(method = lm), x, y, alpha, color, fill, group, linetype, size, weight

Capas

ggplot(diamonds) + geom_point(aes(x=carat, y=price, color=cut)) + geom_smooth(aes(x=carat, y=price))



ggplot(diamonds) + geom_point(aes(x=carat, y=price, color=cut)) + geom_smooth(aes(x=carat, y=price,color=cut))

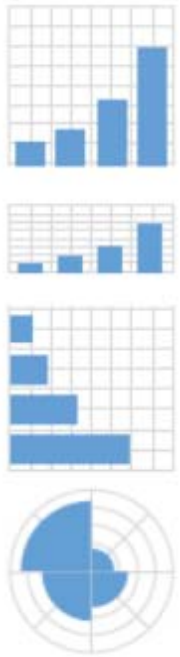


Elementos de ggplot2

Sistema de coordenadas

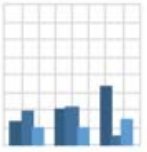
- Define los planos en los cuales los objetos van a estar posicionados en el espacio en el gráfico.

  - **coord_cartesian** - (por defecto)
  - **coord_fixed**
  - **coord_flip** – invierte las coordenadas cartesianas
  - **coord_polar** - x (o y) es mapeado a ángulo (theta)



Posición

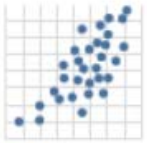
- Los ajustes de posición determinan como ubicar geoms que de otro modo ocuparían el mismo espacio.



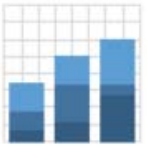
s <- ggplot(mpg, aes(fl, fill = drv))

s + geom_bar(position = "dodge")
Arrange elements side by side

s + geom_bar(position = "fill")
Stack elements on top of one another, normalize height



e + geom_point(position = "jitter")
Add random noise to X and Y position of each element to avoid overplotting



s + geom_bar(position = "stack")
Stack elements on top of one another

Facetas (facets)

Las Facetas dividen un gráfico en múltiples sub-gráficos basado en una o varias variables discretas

- ¿Que diferencia facet_grid de facet_wrap ?

facet_grid(x~y) muestra todos los gráficos x*y incluso si hay gráficos vacíos

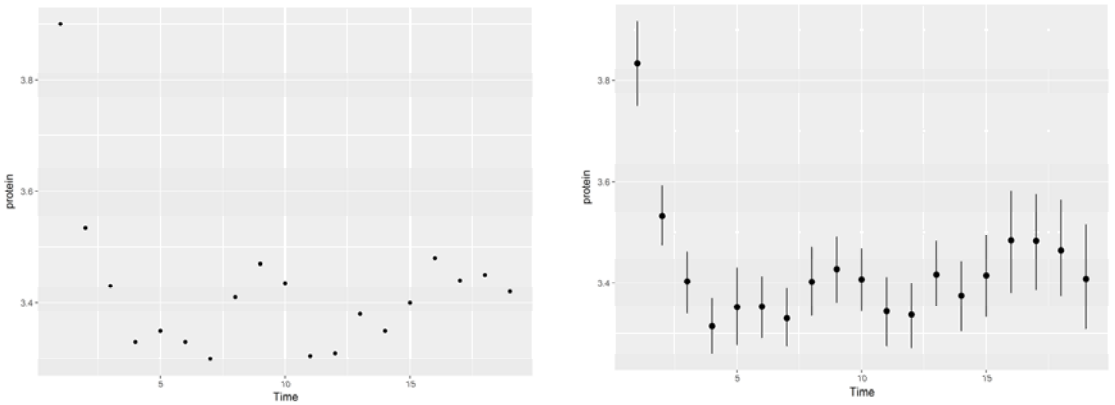
facet_wrap(x~y) solo muestra gráficos que tengan datos

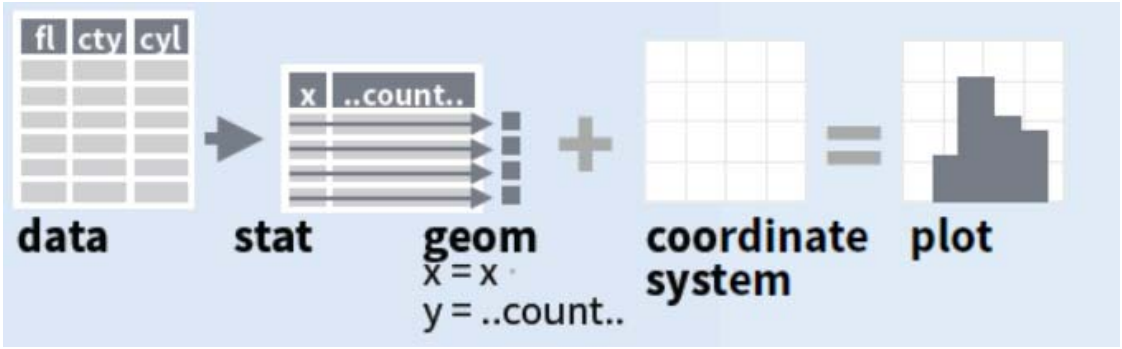
Transformar datos

```
ggplot(Milk, aes(x=Time, y=protein)) +
stat_summary(fun.data="mean_cl_normal")
```

- la función stat_ transforma estadísticamente los datos, normalmente como algún tipo de resumen, como frecuencias, media, limite de confianza

- Cada función stat_ esta asociada por defecto a un geom_ por lo que no habría que agregar geom_





Por ejemplo:
- stat_bin → geom_histogram
- stat_summary → geom_pointrange

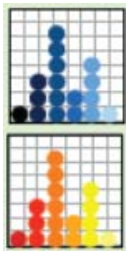
Escalas

scale_<aesthetic>_<type>

- Las **escalas** asignan los valores que hay en los datos a los valores visuales de una estética.

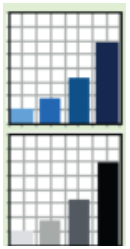
- **Escalas para Color y Relleno (Continuas)**

n + scale_fill_distiller(palette = "Blues")



n + scale_fill_gradient(low="red", high="yellow")

- **Escalas para Color y Relleno (Discretas)**
- n + scale_fill_brewer
- n + scale_fill_grey



Escalas de localización para X e Y

- scale_x_discrete
- scale_y_sqrt()

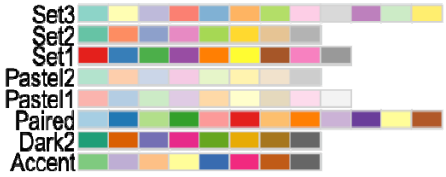
Paletas de colores

los colores pueden ser especificados por su nombre (e.g **col = “red”**)
o como triplete hexadecimal RGB (**col = “#FFCC00”**)

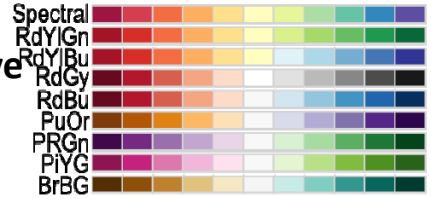
Rcolorbrewer

Sequential

Diverging



Qualitative



ggsci

- NPG
- AAAS
- NEJM
- Lancet
- JAMA
- JCO
- UCSCGB
- D3
- LocusZoom
- IGV
- UChicago
- Star Trek
- Tron Legacy
- Futurama
- Rick and Morty
- The Simpsons

Ghibli

Wes anderson

sumar paletas de colores a ggplot

- Scale_color_ brewer:

```
d + scale_colour_brewer(palette = "Greens")
```

- ggsci :

```
p1 + scale_color_npg ()
p1 + scale_fill_npg()
```

- Ghibli:

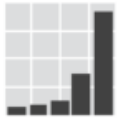
```
P3 + scale_color_manual(values = ghibli_palette("MarnieMedium1"))
```

- Escala de grises:

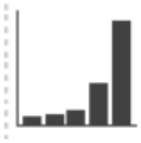
```
p + scale_colour_grey()
```

Temas (Themes)

Themes controla todos los elementos del grafico que no son datos. Puede usarse un theme completo como theme_bw(), o elegir cambiar alguna parte individualmente usando theme() y la funcion element_

theme_grey() (por defecto) 

theme_bw() 

theme_classic() 

theme_dark() 

Extensiones

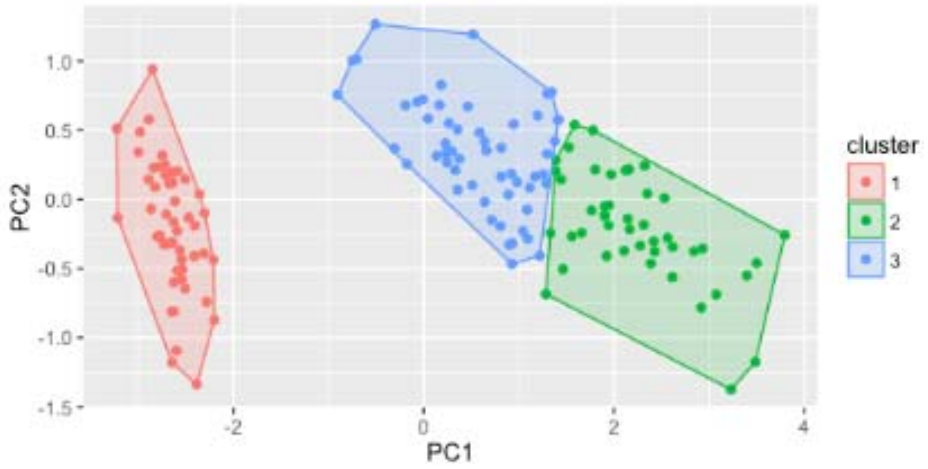
http://www.ggplot2-exts.org/gallery/

Ggrepel ,ggfortify, Gganimate, Ggally , ggridges, ggsci, ggTimeSeries, ggmuller

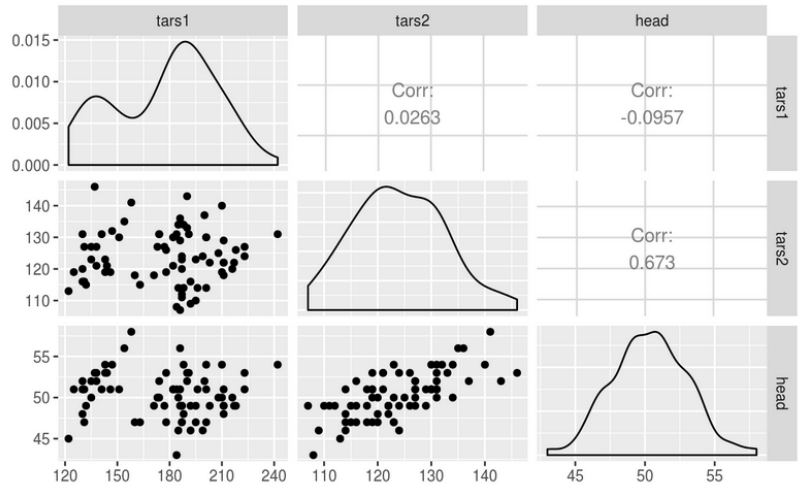
No son extensiones pero pueden ser útiles :Patchwork, gridExtra

Extensiones

GGFORTIFY



GGALLY



Exportar gráficos

Desde Rstudio
- Plot: save

Función ggsave
- ggsave("Fig 4.tiff", plot = Fig4, width=5 fig.asp = 0.618, units="in", dpi=300, compression = "lzw")

Ayuda

- http://stackoverflow.com
- https://ggplot2.tidyverse.org/reference/

- http://www.rstudio.com/resources/cheatsheets/.
- www.data-to-viz.com

- Libro R4DS
- Libro Ggplot2
- https://stats.idre.ucla.edu/stat/data/intro_ggplot2/ggplot2_intro_slidy.html