

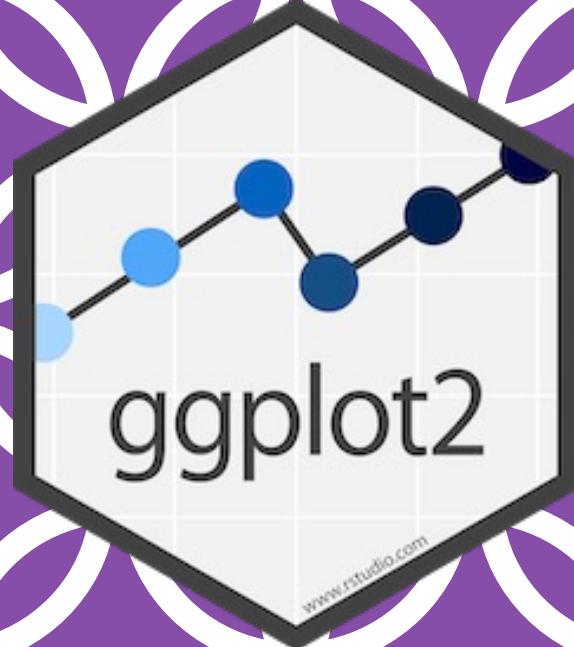


Red Mexicana de
Bioinformática



GGPLOT2

INTERMEDIO



Dra. Ernestina Godoy Lozano
Departamento de
Bioinformática en
Enfermedades Infecciosas
CISEI - INSP



elizabeth.godoy@insp.mx

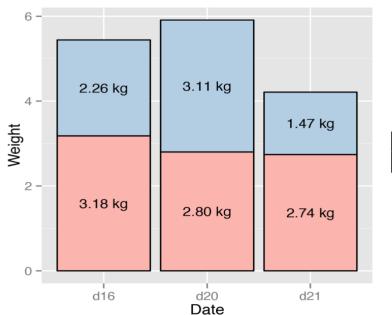
@Tina_Godoy

¿QUÉ APRENDEREMOS?

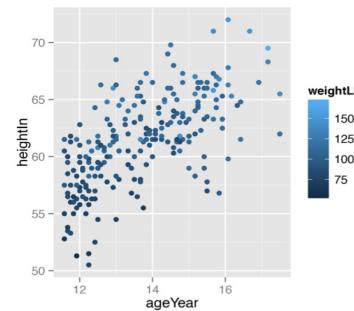
1. Sintaxis de ggplot2

2. Ejemplo de gráficos

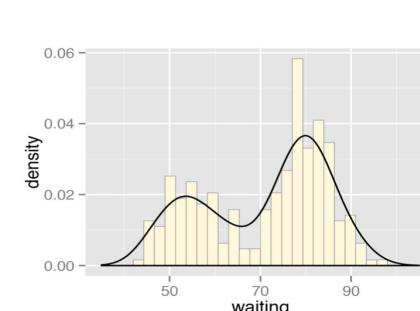
- Gráfica de barras
- Scatterplots
- Histogramas
- Boxplots



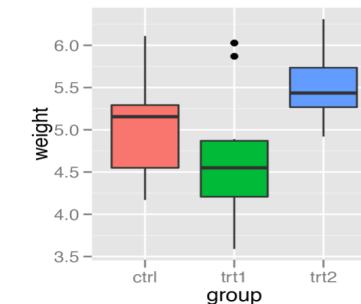
Bar plots



Scatter plots



Histograms



Box plots



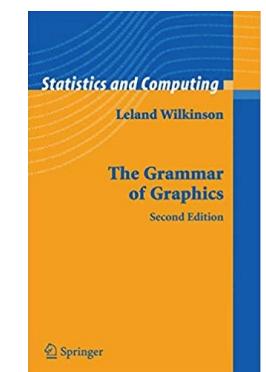
ANTES DE COMENZAR

Material que usaremos en la sesión se encuentra en:

https://github.com/tgodoy/RMB_ggplot2_2022

GGPLOT2

- Es una librería de R implementada por Hadley Wickham para visualización de datos.
- Forma parte de un conjunto de librerías llamado **tidyverse**
- ggplot2 esta basado en “**The Grammar of Graphics**” de Leland Wilkinson (2000).
 - Todos los gráficos pueden generarse mediante un lenguaje regular, con una sintaxis determinada de manera estructurada.
- Tiene más de 10 años y es usado por miles de personas en todo el mundo.
 - **Ventaja:** Hay muchas comunidades que te ayudan a resolver dudas



EXTENSIONES DE GGPLOT2

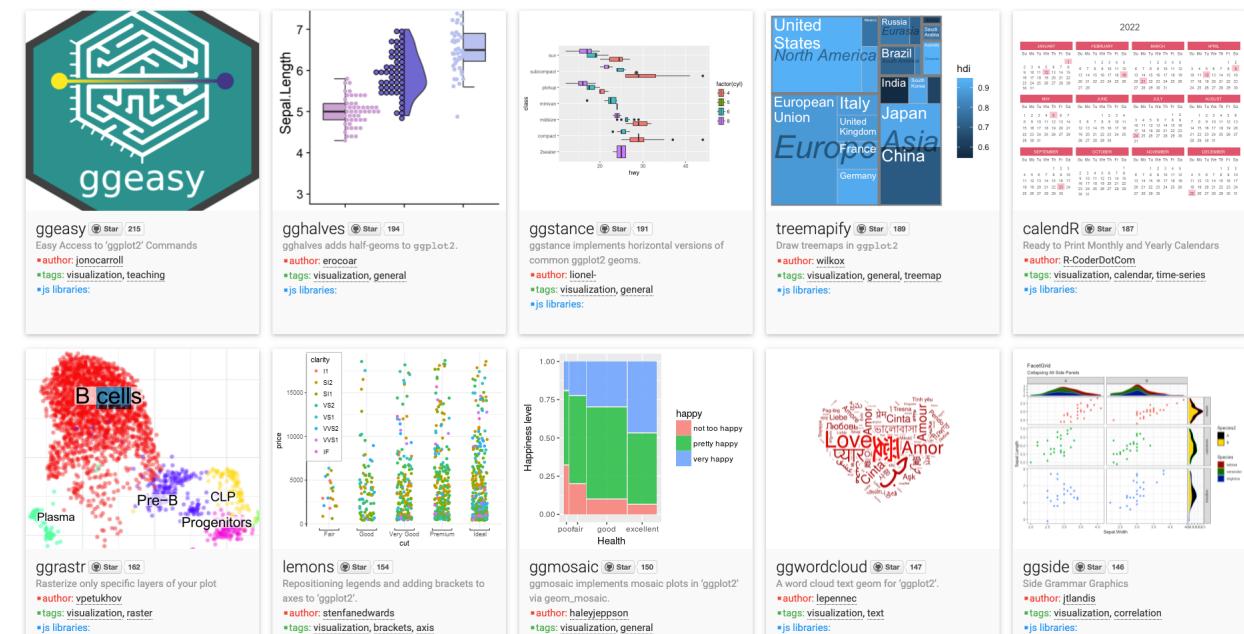
Hay registradas **117 extensiones** de ggplot2

- <https://exts.ggplot2.tidyverse.org/gallery/>

Estas extensiones son librerías de complemento

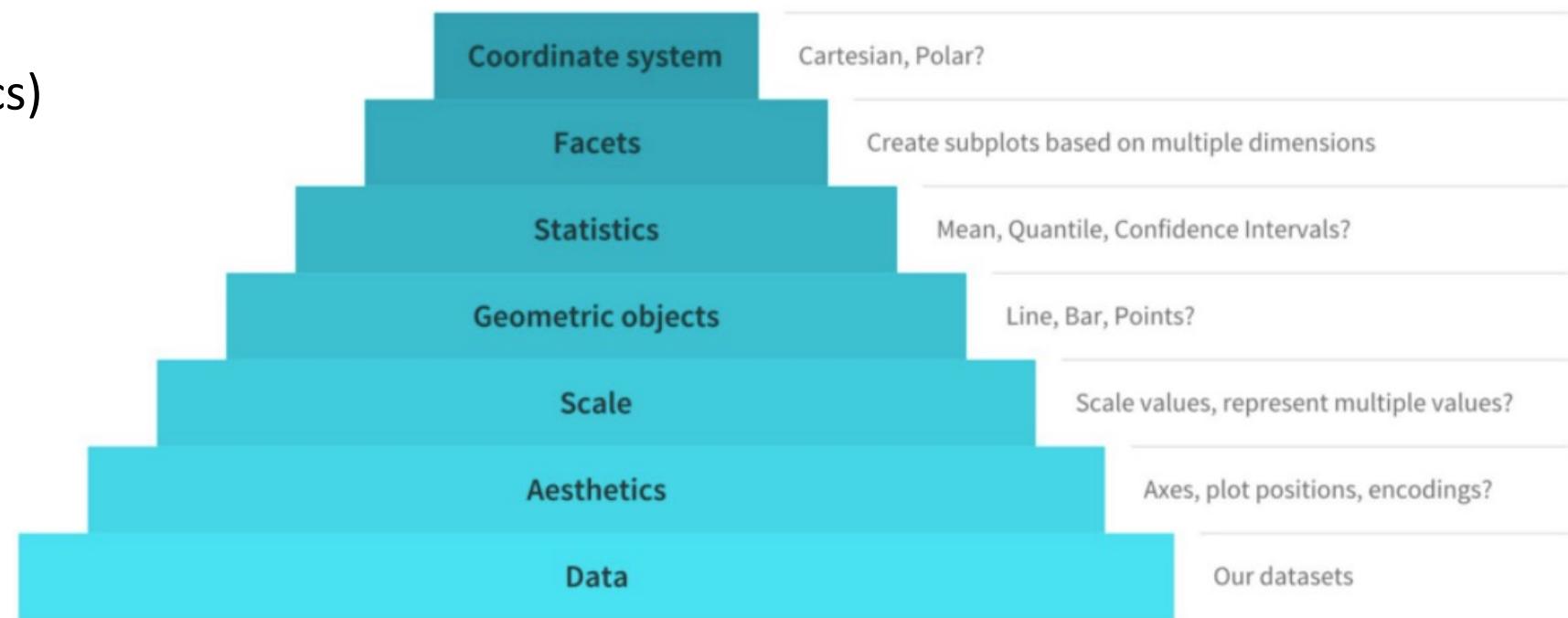
- Tienen la misma sintaxis que ggplot2
- Complementan la librería de ggplot2
- Nos permiten hacer gráficas más complejas
- Nos permiten explorar nuestros datos

Cada extensión es una librería con su documentación y ejemplos propios.

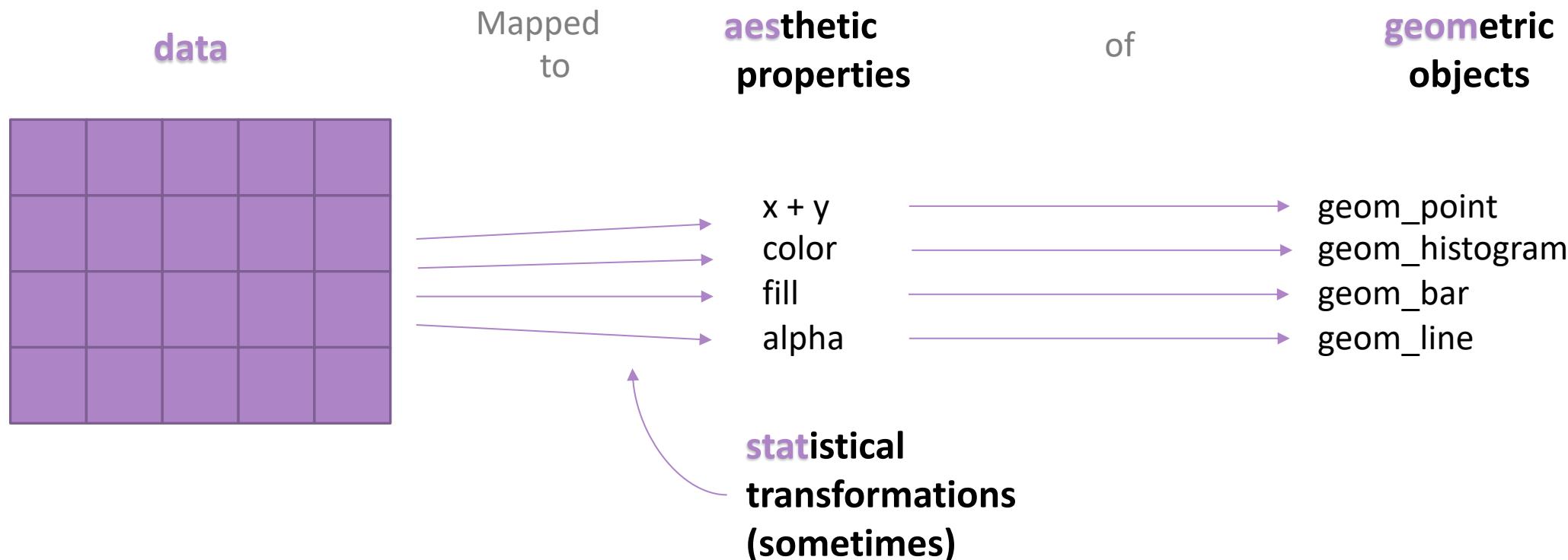


ELEMENTOS DE UN GRÁFICO EN GG PLOT

- Datos
- Estéticas (aes thetics)
- Capas
- Facetas
- Temas

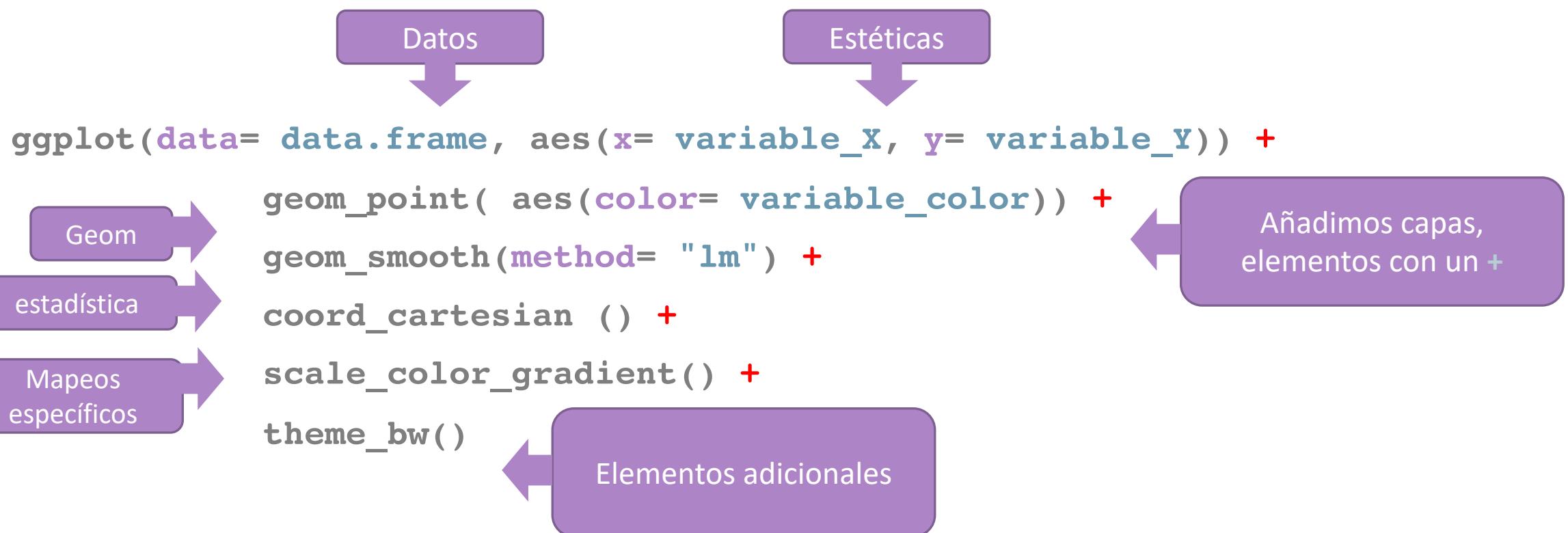


THE GRAMMAR OF THE GRAPHICS Y GGPlot2



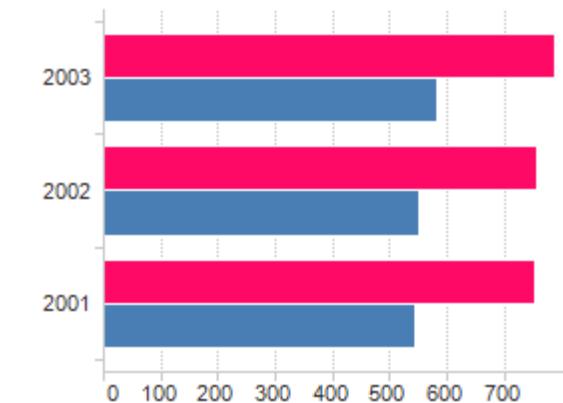
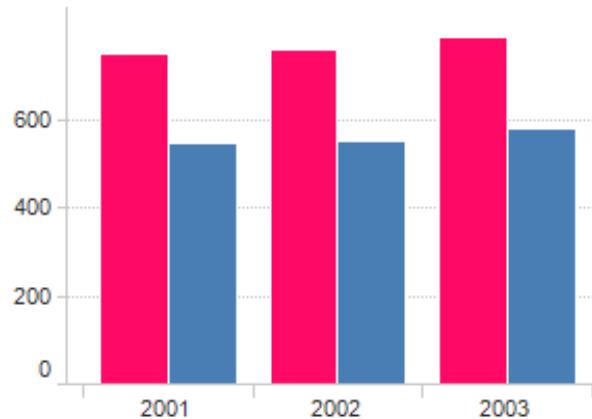
SINTÁXIS GG PLOT()

Crea un gráfico al que se le añaderán capas. Sin valores por defecto, pero que proporciona más control.



BAR PLOTS

- Un gráfico de barras es una forma de **resumir un conjunto de datos por categorías**.
- Muestra los datos usando **varias barras de la misma anchura**, cada una de las cuales representa una categoría concreta.
- La altura de cada barra es proporcional a una agregación específica.



EJERCICIO: BAR PLOTS

cabbage_exp → Es el recopilado de un experimento de un cultivo de "Col o repollo (cabbage)".

- Fueron 2 condiciones C39 y C52;
- Se recolectaron las coles a los días 16, 20 y 21
- se pesaron 10 coles de la condición
- Se obtuvo la desviación estandar y el error estandar de la media del peso.



```
cabbage_exp <- data.frame(Cultivar=c(rep("c39", 3), rep("c52", 3)),  
                           Date=rep(c("d16", "d20", "d21"), 2),  
                           Weight=c(3.18, 2.80, 2.74, 2.26, 3.11, 1.47),  
                           sd= c(0.9566144, 0.2788867, 0.9834181, 0.4452215, 0.7908505, 0.2110819),  
                           n=rep(10, 6),  
                           se=c(0.30250803, 0.08819171, 0.31098410, 0.14079141, 0.25008887, 0.06674995))  
  
# Explorar datos  
cabbage_exp
```

EJERCICIO: BAR PLOTS

```

# Grafico de Bar plots basico
ggplot(data= cabbage_exp, aes(x=Date, y= Weight)) +
  geom_bar(stat="identity")

# Cambio de color de las barras
ggplot(data= cabbage_exp, aes(x=Date, y= Weight)) +
  geom_bar(stat="identity", fill="dodgerblue")

# Cambio de tema
ggplot(data= cabbage_exp, aes(x=Date, y= Weight)) +
  geom_bar(stat="identity", fill="dodgerblue") +
  theme_bw()

### Grafico con barras apiladas coloreado por condicion
ggplot(data= cabbage_exp, aes(x=Date, y=Weight, fill=Cultivar)) +
  geom_bar(stat="identity")

### Añadir texto
ggplot(data= cabbage_exp, aes(x=Date, y=Weight, fill=Cultivar)) +
  geom_bar(stat="identity") +
  geom_text(aes(label=Weight), vjust=0.4)

### Grafico con barras separadas por condicion
ggplot(data= cabbage_exp, aes(x=Date, y=Weight, fill=Cultivar)) +
  geom_bar(stat="identity", position = "dodge")

### Usa otra paleta de colores
ggplot(data= cabbage_exp, aes(x=Date, y=Weight, fill=Cultivar)) +
  geom_bar(stat="identity", position="dodge") +
  scale_fill_brewer(palette="Set1")

```

```

## Añadir texto Posicion "dodge"
ggplot(data= cabbage_exp, aes(x=Date, y=Weight, fill=Cultivar)) +
  geom_bar(stat="identity", position="dodge") +
  scale_fill_brewer(palette="Set1") +
  geom_text(aes(label=Weight), vjust=-0.4,
            position=position_dodge(0.9), size=3, colour="green4")

### Añadir barras de error y cambia tema
ggplot(data= cabbage_exp, aes(x=Date, y=Weight, fill=Cultivar)) +
  geom_bar(stat="identity", position="dodge") +
  scale_fill_brewer(palette="Set1") +
  geom_errorbar(aes(ymin=Weight-se, ymax=Weight+se),
                width=0.2, position=position_dodge(0.9)) +
  theme_bw()

### Cambiar los titulos de los ejes
ggplot(data= cabbage_exp, aes(x=Date, y=Weight, fill=Cultivar)) +
  geom_bar(stat="identity", position="dodge") +
  scale_fill_brewer(palette="Set1") +
  geom_errorbar(aes(ymin=Weight-se, ymax=Weight+se),
                width=0.2, position=position_dodge(0.9)) +
  labs(y="Day", x="Weight (lb)", fill="Experiment") +
  theme_bw()

ggsave("figuras/barplot_cultivos.png", device = "png")

```

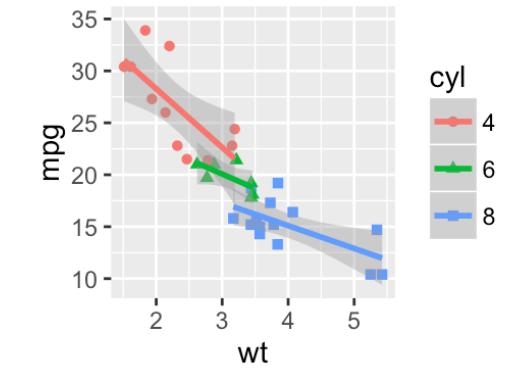
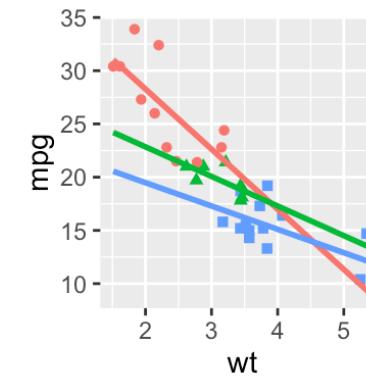
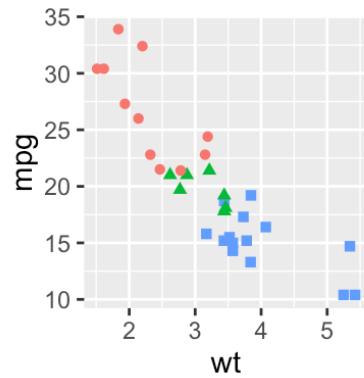
PALETAS DE COLORES EN R

- R tiene paletas de colores ya preestablecidas
- Y tiene una librería que se dedica exclusivamente al color
 - `library(RColorBrewer)`



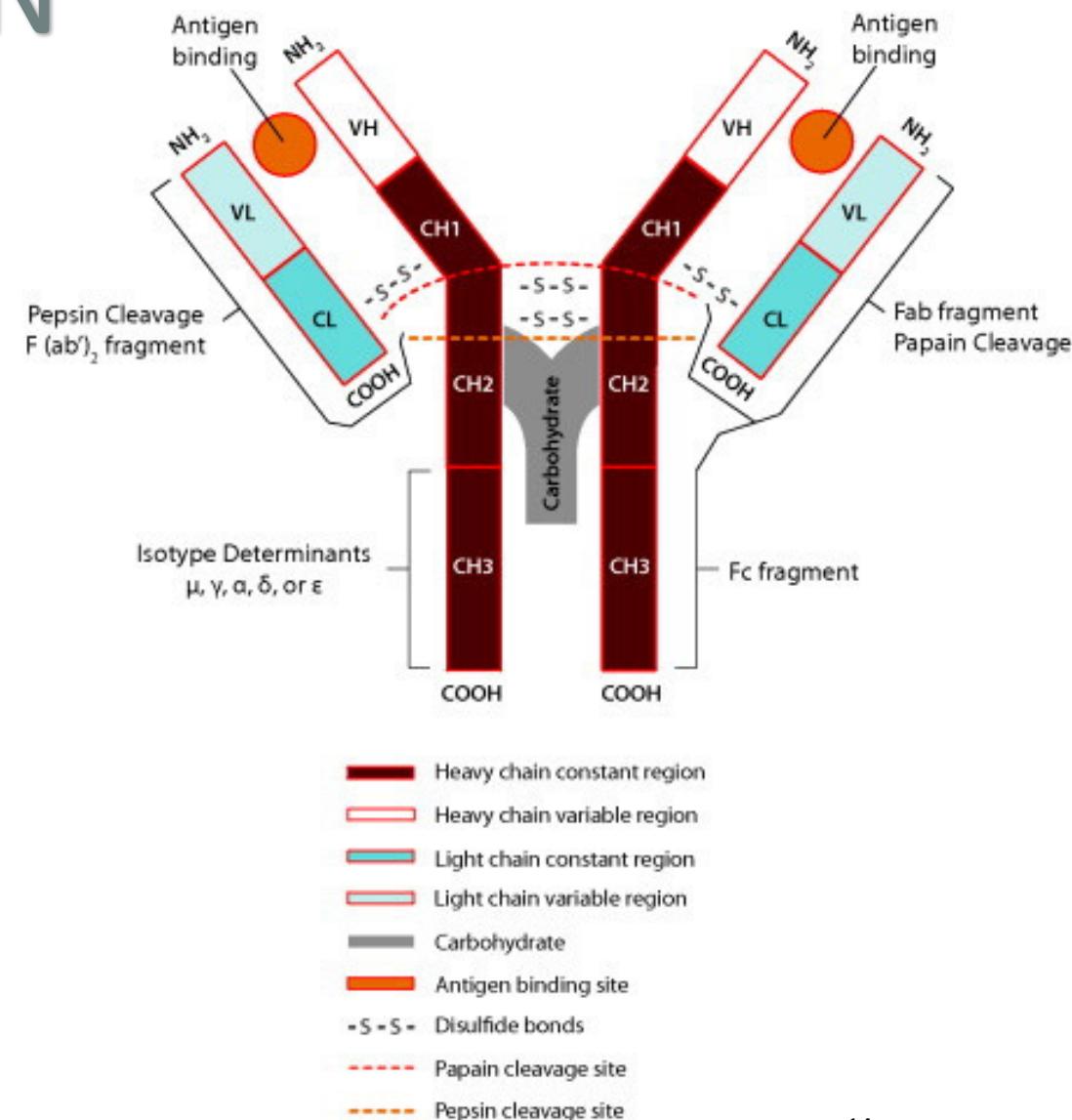
SCATERPLOTS

- El diagrama de dispersión es un tipo de diagrama matemático que utiliza las **coordenadas cartesianas** para mostrar los valores de **dos variables** para un conjunto de datos y observar sus relaciones.
- Uno de los aspectos más poderosos de un gráfico de dispersión es su capacidad para mostrar **las relaciones no lineales entre las variables**.



DATOS PARA LA SESIÓN

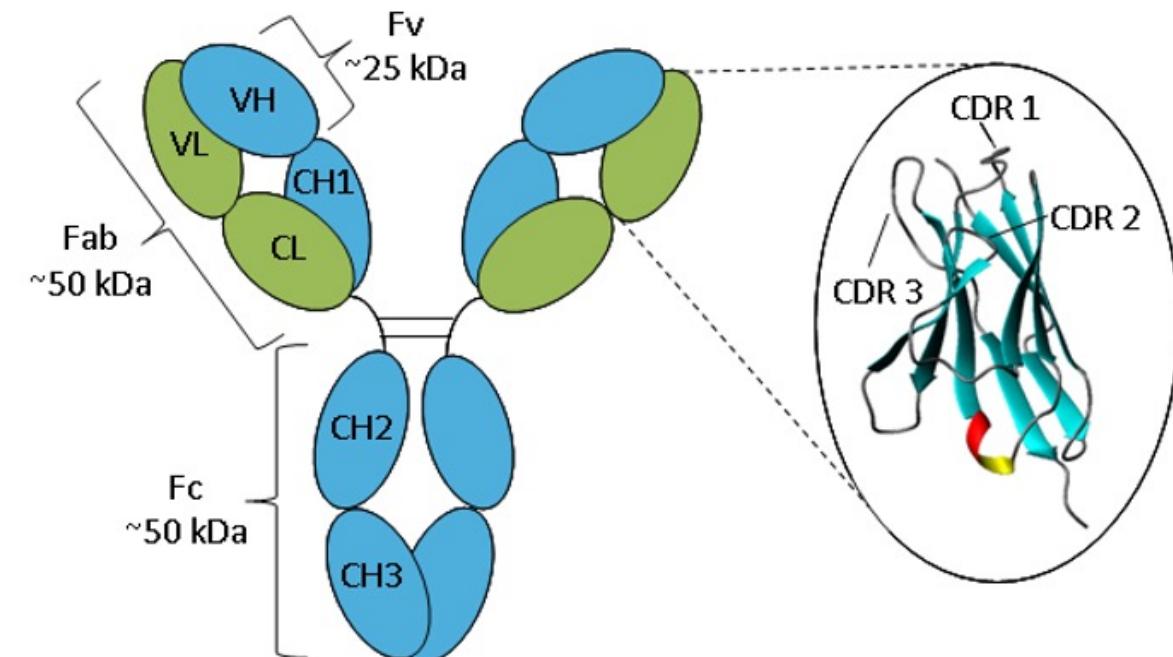
- Los datos son los elementos más importantes de un gráfico
- ggplot2 solo acepta un tipo de datos “**data.frames**”
- Usaremos un set de datos simulados de anticuerpos



DATOS -> ANTIBODIES

Set de datos extraídos de pacientes recuperados de Dengue

- Gen IGHV
- Longitud del CDR3 (nt)
- Porcentaje de identidad con la línea germinal del segmento IGHV-IGHJ
- Número de mutaciones en la región VH



```
## Usaremos un set de datos simulado "antibodies".  
#Datos que provienen de pacientes recuperados de Dengue  
# Explorar los datos  
antibodies <- read.delim("tablas/antibodies.txt", header=T)  
View(antibodies)  
summary(antibodies)  
dim(antibodies)
```

EJERCICIO: SCATTERPLOTS

```
#### Scatter plots basicos
ggplot(data= antibodies, aes(x= CDR3.length, y= VJ.identity)) +
  geom_point()

# Color por condicion
ggplot(data= antibodies, aes(x= CDR3.length, y= VJ.identity, colour=V.GENE)) +
  geom_point()

# Forma por condicion
ggplot(data= antibodies, aes(x= CDR3.length, y= VJ.identity, shape=V.GENE)) +
  geom_point()

# Forma y color por condicion
ggplot(data= antibodies,
       aes(x= CDR3.length, y= VJ.identity, shape=V.GENE, colour=V.GENE)) +
  geom_point()

# Cambiar los valores de forma
ggplot(data= antibodies,
       aes(x= CDR3.length, y= VJ.identity, shape=V.GENE, colour=V.GENE)) +
  geom_point() +
  scale_shape_manual(values=c(1,2,3))

# Cambiar colores manuales
# Con nombre de color
ggplot(data= antibodies, aes(x= CDR3.length, y= VJ.identity, shape=V.GENE, colour=V.GENE)) +
  geom_point() +
  scale_shape_manual(values=c(1,2,3)) +
  scale_colour_manual(values=c("deepskyblue", "seagreen2", "salmon")) +
  theme_bw()

# Con codigo HEX
ggplot(data= antibodies, aes(x= CDR3.length, y= VJ.identity, shape=V.GENE, colour=V.GENE)) +
  geom_point() +
  scale_shape_manual(values=c(15,16,17)) +
  scale_colour_manual(values=c("#24761A", "#9B79D9", "#2468E6")) +
  theme_bw()

### Quitar leyenda, agregar texto y modificar ejes
ggplot(data= antibodies, aes(x= CDR3.length, y= VJ.identity, shape=V.GENE, colour=V.GENE)) +
  geom_point(show.legend = FALSE) +
  scale_shape_manual(values=c(1,2,3)) +
  scale_colour_manual(values=c("#24761A", "#9B79D9", "#2468E6")) +
  annotate("text", x=75, y=85, label="IGHV1-69", colour="#24761A") +
  annotate("text", x=35, y=88, label="IGHV2-5", colour="#9B79D9") +
  annotate("text", x=60, y=82, label="IGHV3-73", colour="#2468E6") +
  labs(x="CDR3 length (bp)", y= "VJ identity (%)", colour="")+
  theme_bw()

ggsave("figuras/scatterplot_antibodies.png", device = "png")
```

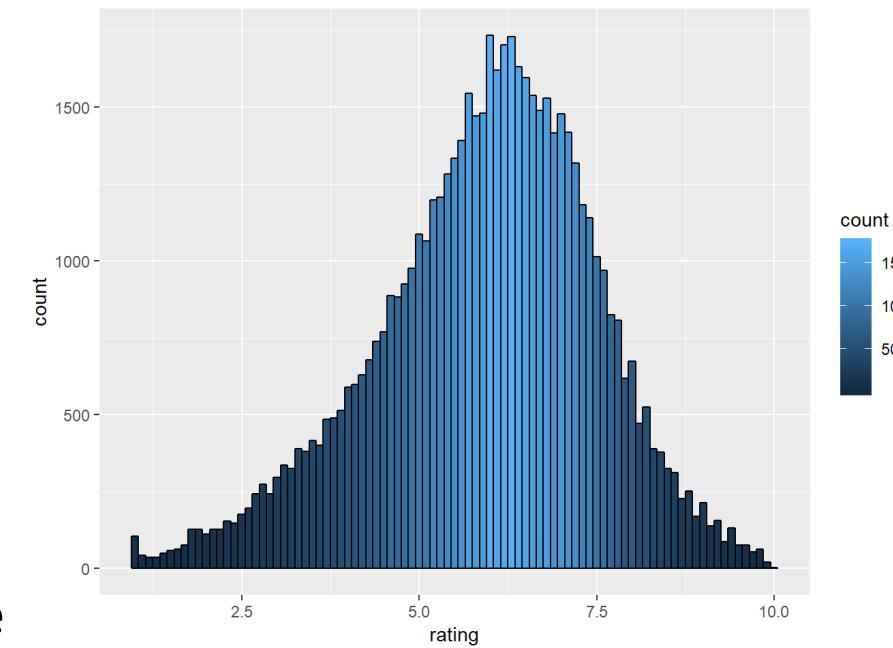
SHAPES IN R

- R tiene un código de **números y símbolos** que pueden cambiar la forma de los objetos que graficamos.

0	□	6	▽	12	田	18	◆	24	△	0	0
1	○	7	⊗	13	⊗	19	●	25	▽	+	+
2	△	8	*	14	☒	20	●	*	*	-	-
3	+	9	◊	15	■	21	○	.	.		
4	×	10	⊕	16	●	22	□	0	○	%	%
5	◇	11	✡	17	▲	23	◆	0	○	#	#

HISTOGRAMAS

- Es una representación gráfica de **una variable en forma de barras**, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados.
- Sirven para obtener una "primera vista" general, o panorama, de la **distribución de la población**, o de la muestra, respecto a una característica, cuantitativa y continua.
- De esta manera podemos **observar alguna tendencia**, por parte de la muestra o población por ubicarse hacia una determinada región de valores dentro del espectro de valores posibles (sean infinitos o no) que pueda adquirir la variable.



EJERCICIO: HISTOGRAMAS

```
## Histograma basico
ggplot(data= antibodies, aes(x= VJ.identity)) +
  geom_histogram()

# Elegir un mejor ancho de barra (binwidth)
ggplot(data= antibodies, aes(x= VJ.identity)) +
  geom_histogram(binwidth=0.3)

# Divide el rango x en 20 barras (bins)
summary(antibodies$VJ.identity)
binsize <- diff(range(antibodies$VJ.identity))/20

# Cambiar el color de las barras
ggplot(data= antibodies, aes(x= VJ.identity)) +
  geom_histogram(binwidth= binsize, colour="black", fill="royalblue4")

# Varios histogramas con diferentes colores de relleno
ggplot(data= antibodies, aes(x= VJ.identity, fill=V.GENE)) +
  geom_histogram(binwidth= binsize, colour="black")

# Cambiar el valor de "alpha" (transparencia)
ggplot(data= antibodies, aes(x= VJ.identity, fill=V.GENE)) +
  geom_histogram(binwidth= binsize, position="identity", alpha=0.4)
```

EJERCICIO: HISTOGRAMAS Y DENSITY PLOTS

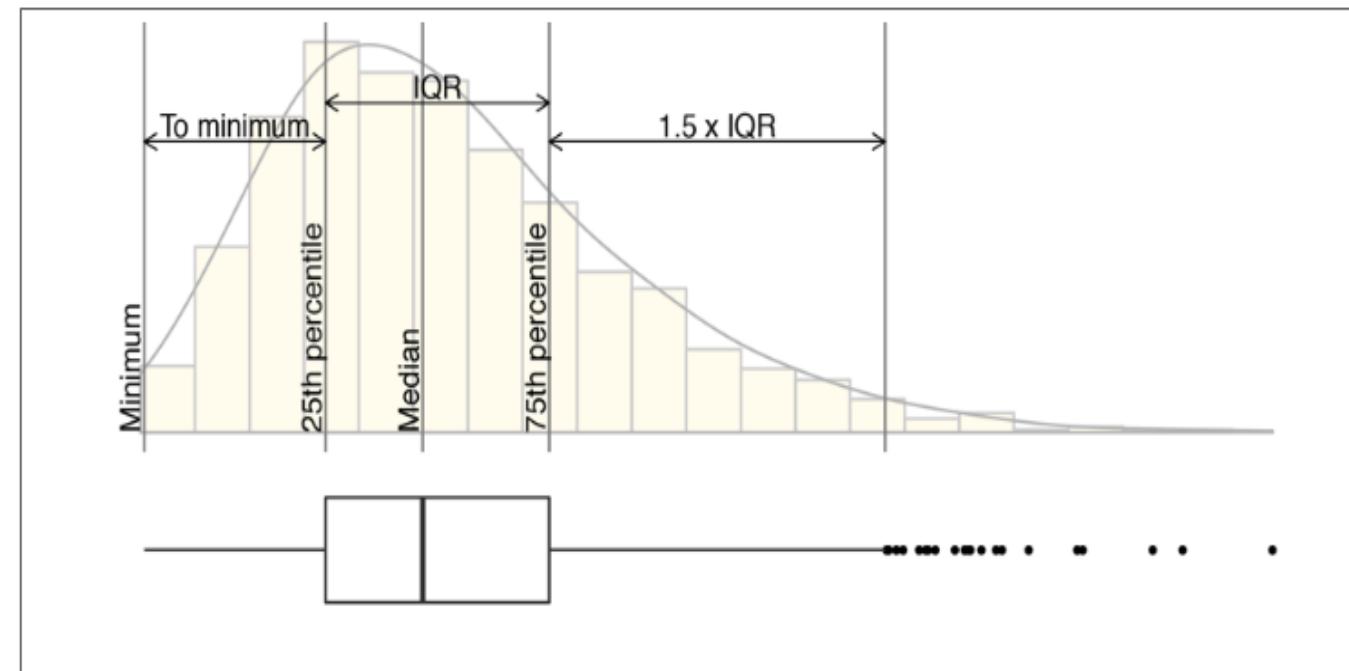
```
# Añadir un Density plot -- Plot de Densidad
ggplot(data= antibodies, aes(x= VJ.identity, fill=V.GENE, y= ..density..)) +
  geom_histogram(binwidth= binsize, position="identity", alpha=0.4, colour="gray") +
  geom_density(alpha=0.4)

# Cambiar la apariencia
ggplot(data= antibodies, aes(x= VJ.identity, fill=V.GENE, y= ..density..)) +
  geom_histogram(binwidth= binsize, position="identity", alpha=0.4, colour="gray") +
  geom_density(alpha=0.4) +
  scale_fill_manual(values=c("deepskyblue", "seagreen2", "salmon")) +
  labs(x="VJ identity (%)", y= "Density", fill "") +
  theme_bw() +
  theme(legend.position = "top")

ggsave("figuras/histogram_antibodies.png", device = "png")
```

BOXPLOTS

- Un boxplot consta de una **caja con bigotes**
- Los boxplots representan **la distribución** de una variable en una población a través de sus **cuartiles**.
- El boxplot esta compuesto por los siguientes elementos:
 - Rango (sin outliers)
 - Outliers
 - Rango intercuartílico
 - Cuartiles (Q1, Q2, y Q3)
 - Mediana (Q2)
 - Mínimo y máximo



EJERCICIO: BOXPLOTS

```
# Boxplot basico
ggplot(data=antibodies, aes(y= V.Nb.mutations, x= V.GENE)) +
  geom_boxplot()

# Con muescas (notch)
ggplot(data=antibodies, aes(y= V.Nb.mutations, x= V.GENE)) +
  geom_boxplot(notch= TRUE)

# Cambiar la apariencia
ggplot(data=antibodies, aes(y= V.Nb.mutations, x= V.GENE, fill=V.GENE)) +
  geom_boxplot(notch= TRUE) +
  scale_fill_manual(values=c("deepskyblue", "seagreen2", "salmon")) +
  theme_bw()

## Cambiar las etiquetas de los ejes
ggplot(data=antibodies, aes(y= V.Nb.mutations, x= V.GENE, fill=V.GENE)) +
  geom_boxplot(notch= TRUE) +
  scale_fill_manual(values=c("deepskyblue", "seagreen2", "salmon")) +
  labs (title= " Boxplot V.GENE", y= "VH mutations (nt)", x= "", fill= "") +
  theme_bw()
```

EJERCICIO: VIOLINS

```
# Grafico de Violin
ggplot(data=antibodies, aes(y= V.Nb.mutations, x= V.GENE, fill=V.GENE)) +
  geom_violin() +
  scale_fill_manual(values=c("deepskyblue", "seagreen2", "salmon")) +
  labs (title= " Boxplot V.GENE", y= "VH mutations (nt)", x= "", fill= "")

# Quitar leyenda y mejorar apariencia
ggplot(data=antibodies, aes(y= V.Nb.mutations, x= V.GENE, fill=V.GENE)) +
  geom_violin(show.legend = FALSE, trim = FALSE) +
  scale_fill_manual(values=c("deepskyblue", "seagreen2", "salmon")) +
  labs (title= " Boxplot V.GENE", y= "VH mutations (nt)", x= "", fill= "") +
  theme_linedraw()

# Añadir una linea de tendencia
# en este caso añadiremos una linea de la media de V.Nb.mutations
v_media <- mean(antibodies$V.Nb.mutations)

ggplot(data=antibodies, aes(y= V.Nb.mutations, x= V.GENE, fill=V.GENE)) +
  geom_violin(show.legend = FALSE, trim = FALSE) +
  scale_fill_manual(values=c("deepskyblue", "seagreen2", "salmon")) +
  labs (title= " Boxplot V.GENE", y= "VH mutations (nt)", x= "", fill= "") +
  geom_hline(yintercept = v_media, col="gold", linetype="dashed") +
  annotate("text", x = 0.7, y=30, label="Mean", colour="gold") +
  theme_linedraw()

ggsave("figuras/boxplot_antibodies.png", device = "png")
```

EJEMPLO DE VIOLINS USANDO FACET_WRAP()

```
#### Cargar libreria extra
library(reshape2)
cdrs <- antibodies[,1:4]
View(cdrs)
tmp <- melt(cdrs)

# Explorar los datos ¿Que hace la funcion melt?
View(tmp)
dim(tmp)
summary(tmp)

# boxplot con ajuste de faceta (facet wrap)
ggplot(data= tmp, aes(x= V.GENE, y= value)) +
  geom_violin() +
  facet_wrap(~variable)

# Cambiar la apariencia
ggplot(data= tmp, aes(x= V.GENE, y= value, fill= V.GENE)) +
  geom_violin(trim = FALSE) +
  facet_wrap(~variable) +
  scale_fill_manual(values=c("deepskyblue", "seagreen2", "salmon")) +
  labs(y="length (nt)", x="", fill "") +
  theme_linedraw() +
  theme(axis.text.x = element_text(angle = 45, vjust = 0.5, hjust=0.5),
        legend.position = "bottom")

ggsave("figuras/extraviolin_antibodies.png", device = "png")
```

RESHAPE2::MELT ()

¿Qué le hace la función melt a mis datos?

Mis datos			
ID	Tiempo	Var1	Var2
1	1	5	10
1	2	3	20
2	1	6	15
2	2	2	25

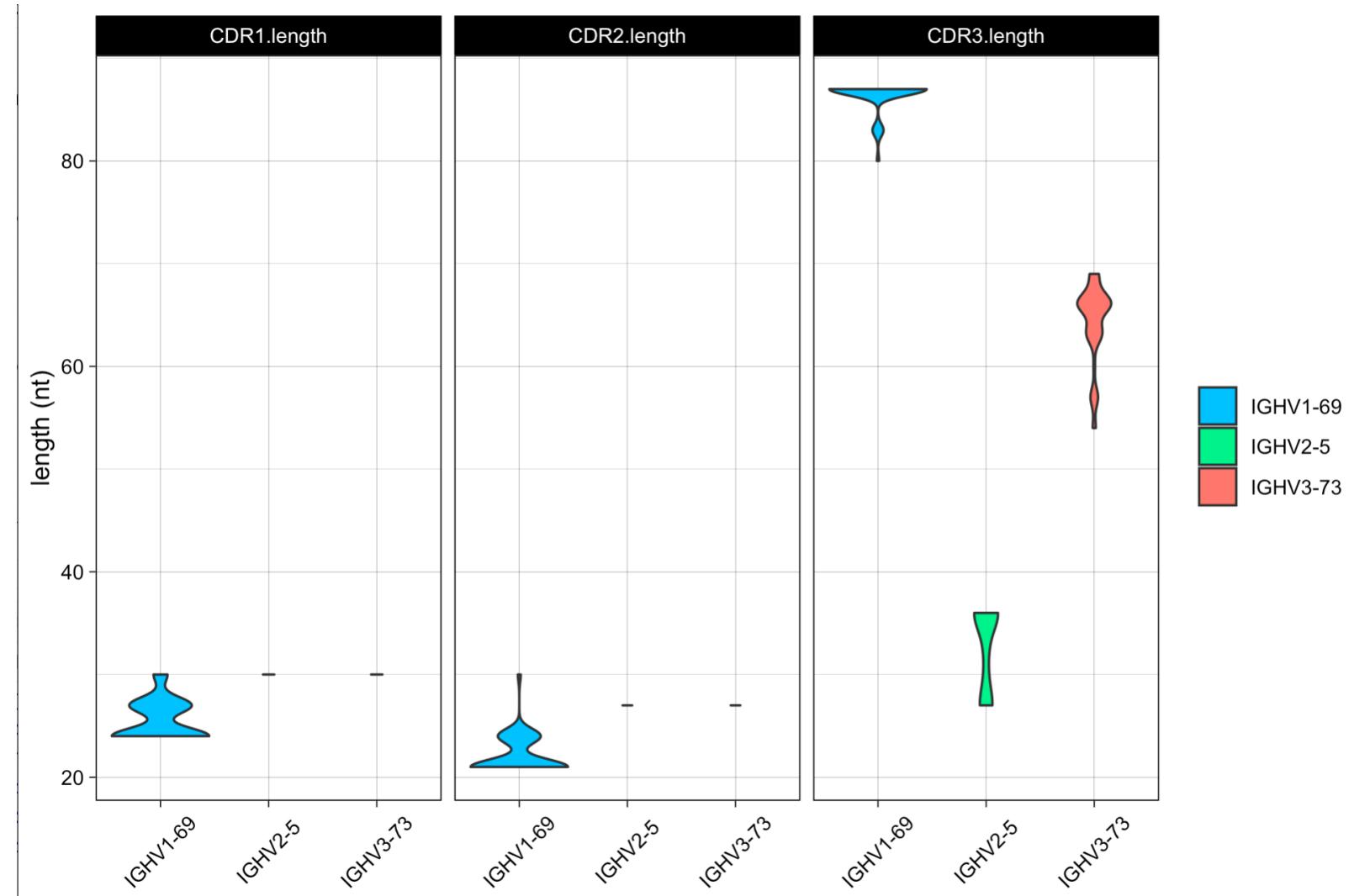


Mis datos transformados

ID	Tiempo	Variable	Value
1	1	Var1	5
1	2	Var1	3
2	1	Var1	6
2	2	Var1	2
1	1	Var2	10
1	2	Var2	20
2	1	Var2	15
2	2	Var2	25

EXTRA DE VIOLINS

Usando la función `melt` y aplicando el `facet_warp`



RECURSOS EXTRAS

- R Graphics Cookbook de Winston Chang
 - <https://r-graphics.org/>
- Página de “tidyverse”
 - <https://ggplot2.tidyverse.org/>
- STHDA (Statistical tools for hig-througput data analysis)
 - <http://www.sthda.com/english/wiki/ggplot2-essentials>
- Tutoriales en línea
 - <http://r-statistics.co/Complete-Ggplot2-Tutorial-Part1-With-R-Code.html>
 - <http://r-statistics.co/ggplot2-Tutorial-With-R.html>
- Cheatsheets
 - <https://rstudio.com/wp-content/uploads/2015/04/ggplot2-spanish.pdf>
- Google
 - “How to in ggplot2“

COLORES EN R

- Selector de colores (RGB y HEX)

<https://htmlcolorcodes.com/es/selector-de-color/>

- R chart color

<http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf>

- Paletas de colores en R

<https://r-charts.com/color-palettes/>