Visualización de datos

Bioestadística

Marta Coronado (marta.coronado@uab.cat)

Grado en Genética | Curso 2025/26

Facultat Biociències UAB

Outline

1. Gramática de los gráficos

   - Tidy data

2. Gráficos con `ggplot2`

   - Boxplot
   - Histograma
   - Scatterplot

3. Librerías especializadas

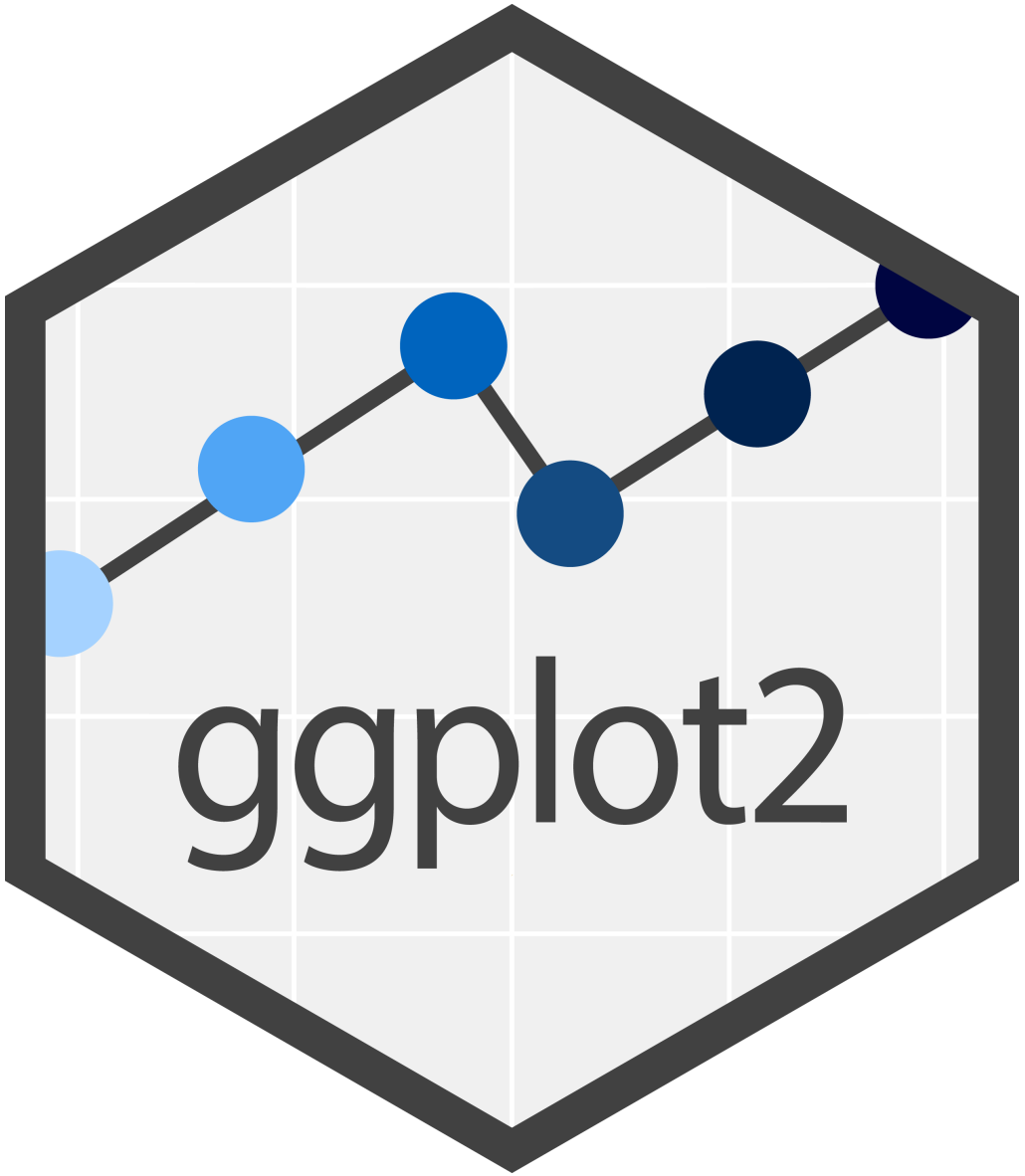
01

La gramática de los gráficos

¿Qué es la visualización de datos?

- La visualización de datos es la representación gráfica de los datos
- El objetivo principal es comunicar información de manera clara y efectiva
- La funcionalidad y la estética han de ir de la mano

Data insights: a visualization (Aisch 2019)



`ggplot2` es un popular paquete de visualización de datos de código abierto para el **lenguaje de programación R**.

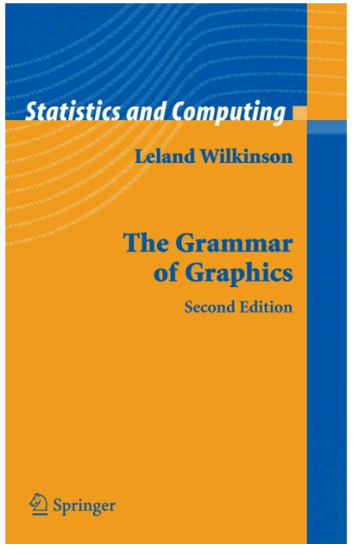
Fue desarrollado por **Hadley Wickham** y se basa en la **gramática de gráficos**, que proporciona un marco coherente y sistemático para crear una amplia gama de visualizaciones de datos.

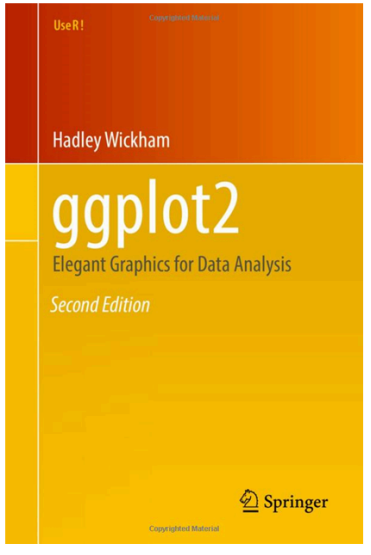
1. Capas (Layering)
2. Flexibilidad
3. Reproducibilidad
4. Comunidad

La gramática de los gráficos

Gramática original

Wilkinson, Leland. The grammar of graphics. Springer Science & Business Media, 2006.





Adaptado a `R` en el paquete `ggplot2`

Hadley Wickham. ggplot2: elegant graphics for data analysis. Springer, 2009.

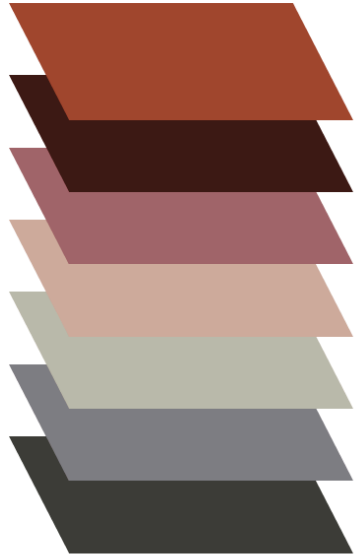
"La gramática nos dice que un **gráfico estadístico** es un **mapeo** de **datos** a **atributos estéticos** (color, forma, tamaño) de **objetos geométricos** (puntos, líneas, barras). El gráfico también puede contener **transformaciones estadísticas** de los datos y se dibuja en un **sistema de coordenadas específico**. Es la **combinación de estos componentes independientes lo que forma un gráfico**."

Sintaxis

En `ggplot2` hay diferentes componentes que podemos añadir a un gráfico:

```
ggplot(data = <DATA>,
    mapping = aes(<MAPPINGS>)) +

    <GEOM_FUNCTION>(stat = <STAT>,
    position = <POSITION>) +

    <SCALE_FUNCTION>() +

    <COORDINATE_FUNCTION>() +

    <FACET_FUNCTION>() +

    <THEME_FUNCTION>
```

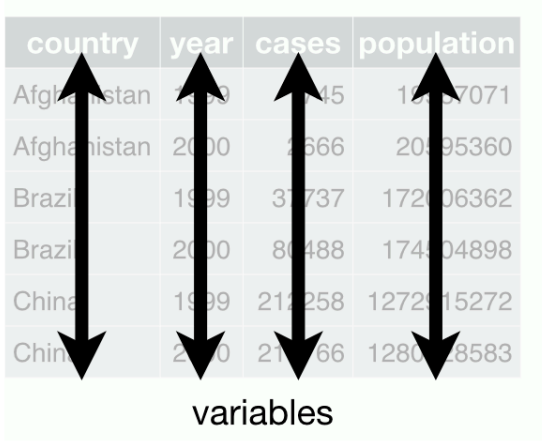
Describes all the non-data ink	**Theme**
Rows and columns of sub-plots	**Facets**
Plotting space for the data	**Coordinates**
Statistical models and summaries	**Statistics**
Shapes used to represent the data	**Geometries**
Scales onto which data is mapped	**Aesthetics**
The actual variables to be plotted	**Data**

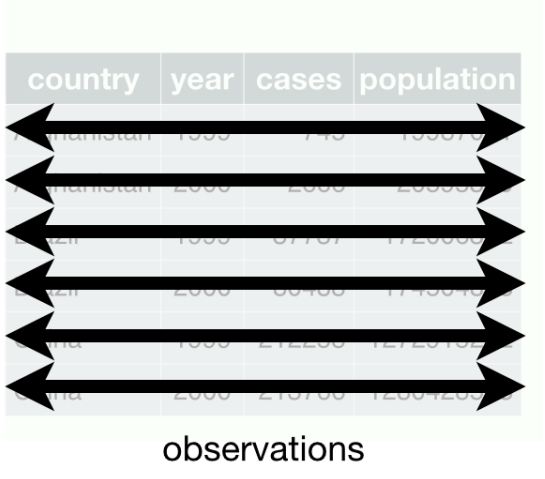


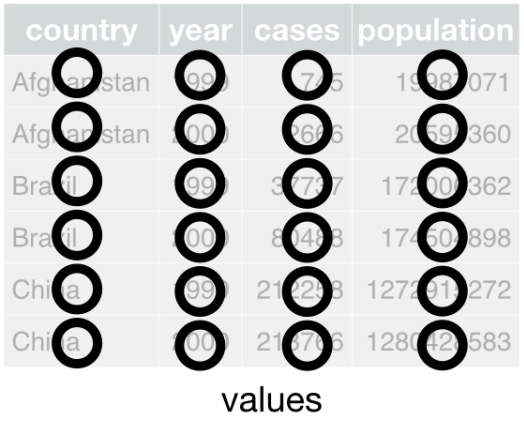
Grammar of Graphics:
A layered approach to elegant visuals

Datos ordenados (tidy data)

Data frames con una observación por fila y una variable por columna.







Datos ordenados (tidy data)

Dos tipos de estructuras de datos ordenados:

- Formato ancho (el más común): en un formato ancho, las múltiples medidas de una sola observación se almacenan en una sola fila.

```
##   Student Math Literature PE
## 1       A   99         45 56
## 2       B   73         78 55
## 3       C   12         96 57
```

- Formato largo: cada fila corresponde a una medida de una observación.

```
## # A tibble: 9 × 3
##   Student Subject    Score
##   <chr>   <chr>      <dbl>
## 1 A       Math          99
## 2 A       Literature    45
## 3 A       PE            56
## 4 B       Math          73
## 5 B       Literature    78
## 6 B       PE            55
## 7 C       Math          12
## 8 C       Literature    96
## 9 C       PE            57
```

Datos ordenados (tidy data)

Existen funciones para cambiar del formato ancho al formato largo:

```
# Paquete tidyverse, función pivot_longer
library(tidyverse)
long_df <- pivot_longer(
  wide_df,
  cols = c(Math, Literature, PE), # o  cols = -c(Student),
  names_to = "Subject",
  values_to = "Score"
  )
long_df
```

```
## # A tibble: 9 × 3
##   Student Subject    Score
##   <chr>   <chr>      <dbl>
## 1 A       Math          99
## 2 A       Literature    45
## 3 A       PE            56
## 4 B       Math          73
## 5 B       Literature    78
## 6 B       PE            55
## 7 C       Math          12
## 8 C       Literature    96
```

Trabajando con datos

Leeremos los datos que corresponden a casos de una epidemia de ébola simulada (obtenidos de: https://www.epirhandbook.com/). En R, una tabla se llama **data.frame**.

```
surv_raw <- read_tsv("data/ebola_epidemic.tab", col_names = T)
```

	case_id	generation	date_infection	date_onset	date_hospitalisation	date_outcome	outcome	sex	age	age_unit	age_years	age_cat	age_cat5
1	5fe599	4	2014-05-08	2014-05-13	2014-05-15			m	2	years	2	0-4	0-4
2	8689b7	4		2014-05-13	2014-05-14	2014-05-18	Recover	f	3	years	3	0-4	0-4
3	11f8ea	2		2014-05-16	2014-05-18	2014-05-30	Recover	m	56	years	56	50-69	55-59
4	b8812a	3	2014-05-04	2014-05-18	2014-05-20			f	18	years	18	15-19	15-19
5	893f25	3	2014-05-18	2014-05-21	2014-05-22	2014-05-29	Recover	m	3	years	3	0-4	0-4
6	be99c8	3	2014-05-03	2014-05-22	2014-05-23	2014-05-24	Recover	f	16	years	16	15-19	15-19
7	07e3e8	4	2014-05-22	2014-05-27	2014-05-29	2014-06-01	Recover	f	16	years	16	15-19	15-19

Showing 1 to 25 of 25 entries

02

Gráficos con `ggplot2`

Cómo hacer boxplots avanzados?

Queremos representar la edad (age) de los grupos de outcome de nuestra tabla surv_raw.

```
## # A tibble: 3 × 2
##   outcome n_rows
##   <chr>    <int>
## 1 Death     2582
## 2 Recover   1983
## 3 <NA>      1323
```

Primero, indicamos los datos (el dataframe):

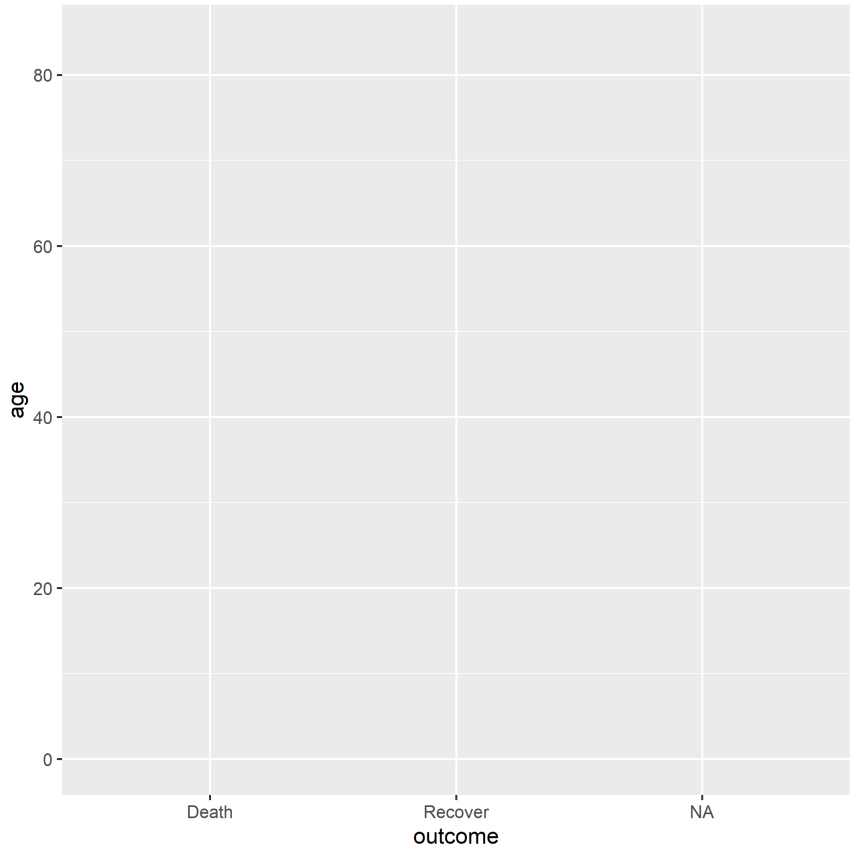
```
ggplot(data = surv_raw)
```

Cómo hacer boxplots avanzados?

Queremos representar la edad (`age`) de los grupos de `outcome` de nuestra tabla `surv_raw`.

Después, indicamos las variables que usamos para los ejes `x` e `y`:

```
ggplot(data = surv_raw,
       mapping = aes(x = outcome, y = age))
```

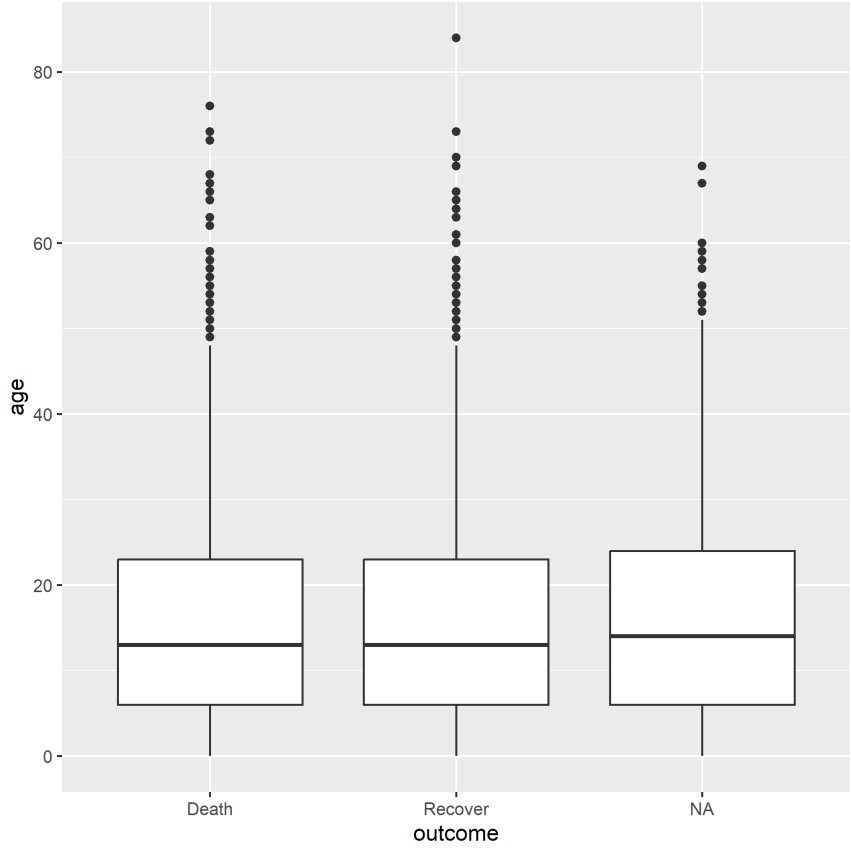


Cómo hacer boxplots avanzados?

Queremos representar la edad (`age`) de los grupos de `outcome` de nuestra tabla `surv_raw`.

Indicamos de qué tipo es el gráfico, en este caso, haremos un boxplot:

```
ggplot(data = surv_raw,
       mapping = aes(x = outcome, y = age)) +
  geom_boxplot()
```

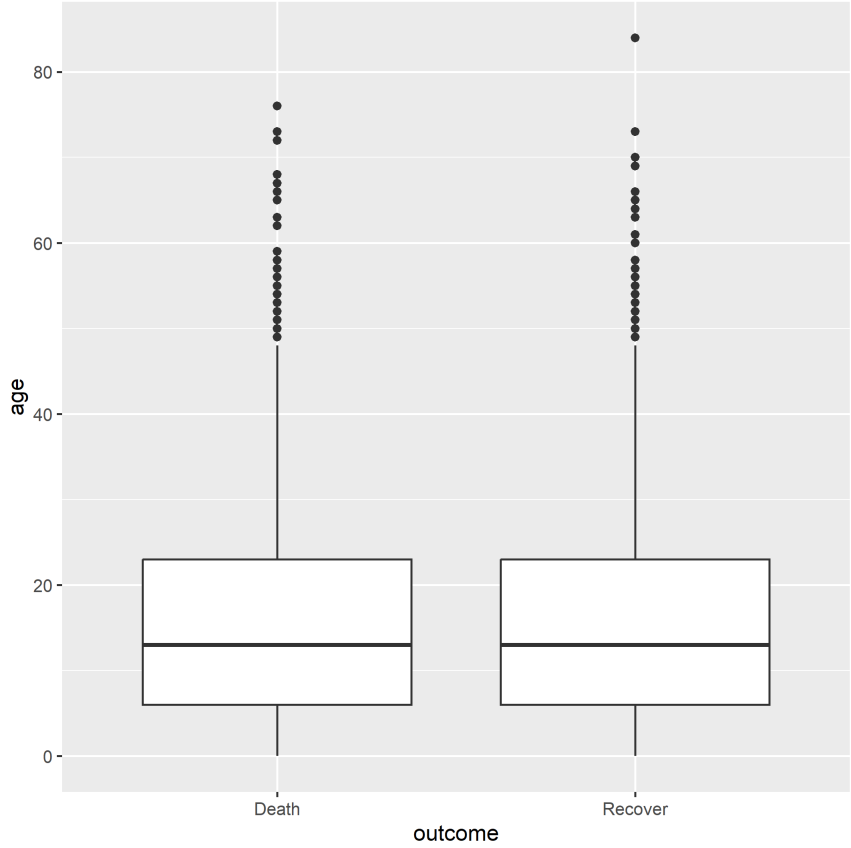


Cómo hacer boxplots avanzados?

Queremos representar la edad (`age`) de los grupos de `outcome` de nuestra tabla `surv_raw`.

Podemos usar la función `drop_na()` para eliminar los casos NA (no tenemos información de la edad):

```
library(tidyr)

ggplot(data = surv_raw %>% drop_na(outcome),
       mapping = aes(x = outcome, y = age)) +
  geom_boxplot()
```

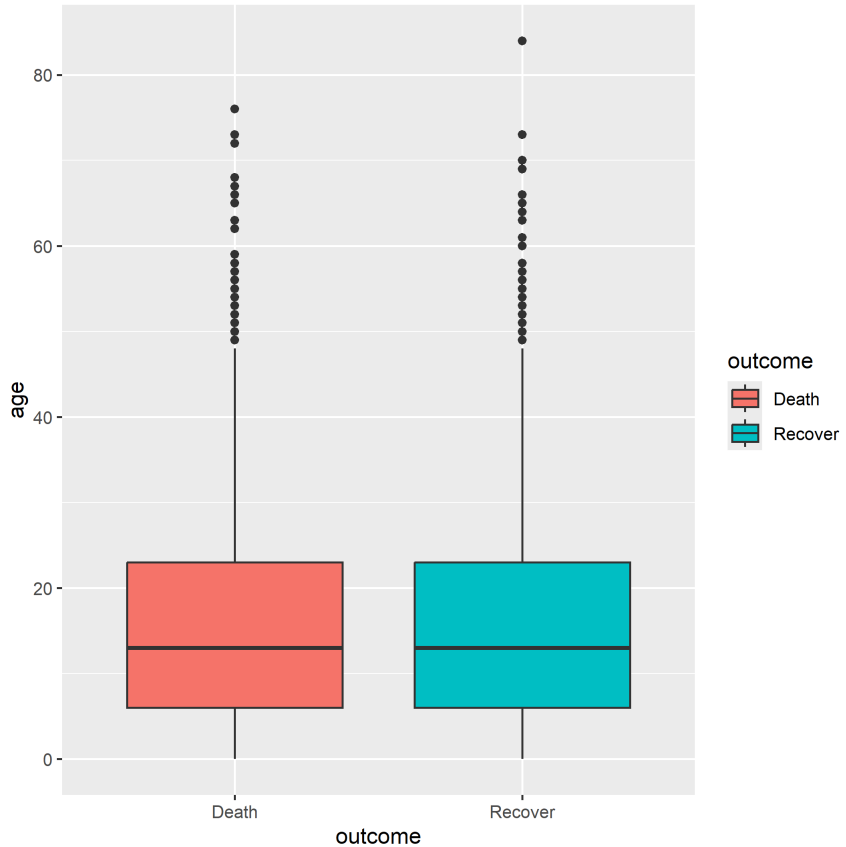


Cómo hacer boxplots avanzados?

Queremos representar la edad (`age`) de los grupos de `outcome` de nuestra tabla `surv_raw`.

Con `fill`, indicamos que queremos colorear según el grupo de `outcome`.

```
ggplot(data = surv_raw %>% drop_na(outcome),
       mapping = aes(x = outcome, y = age,
                     fill = outcome)) +
  geom_boxplot()
```

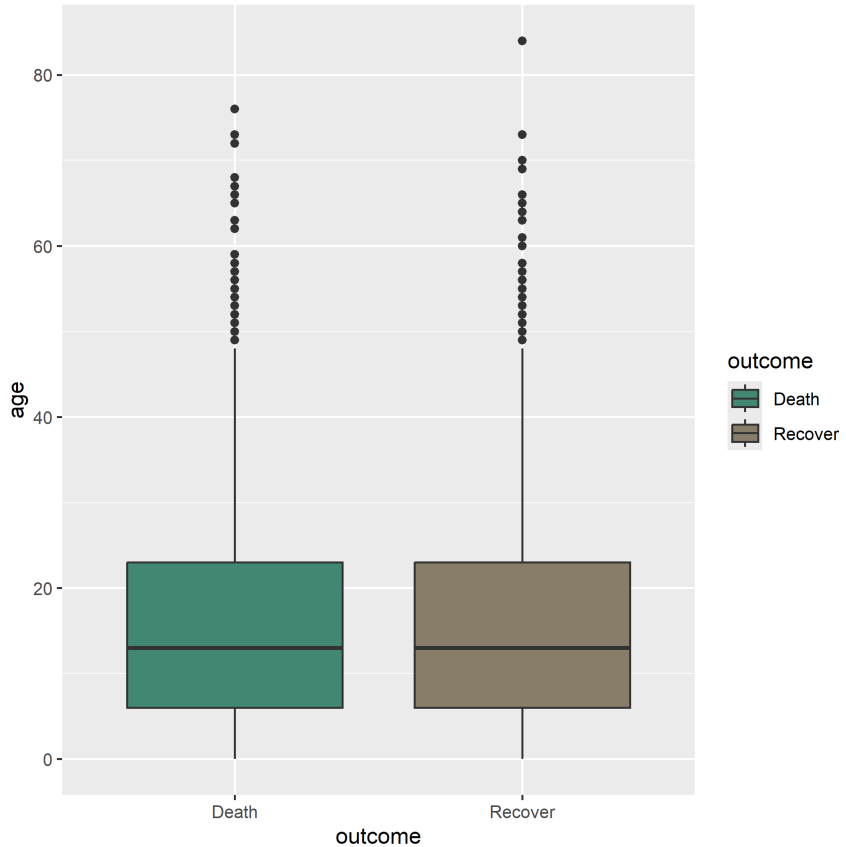


Cómo hacer boxplots avanzados?

Queremos representar la edad (`age`) de los grupos de `outcome` de nuestra tabla `surv_raw`.

Con `scale_fill_manual`, indicamos que de manera manual asignaremos los colores que queremos usar:

```
ggplot(data = surv_raw %>% drop_na(outcome),
       mapping = aes(x = outcome, y = age,
                     fill = outcome)) +
  geom_boxplot() +
    scale_fill_manual(
    values = c("Death" = "aquamarine4",
               "Recover" = "bisque4"))
```



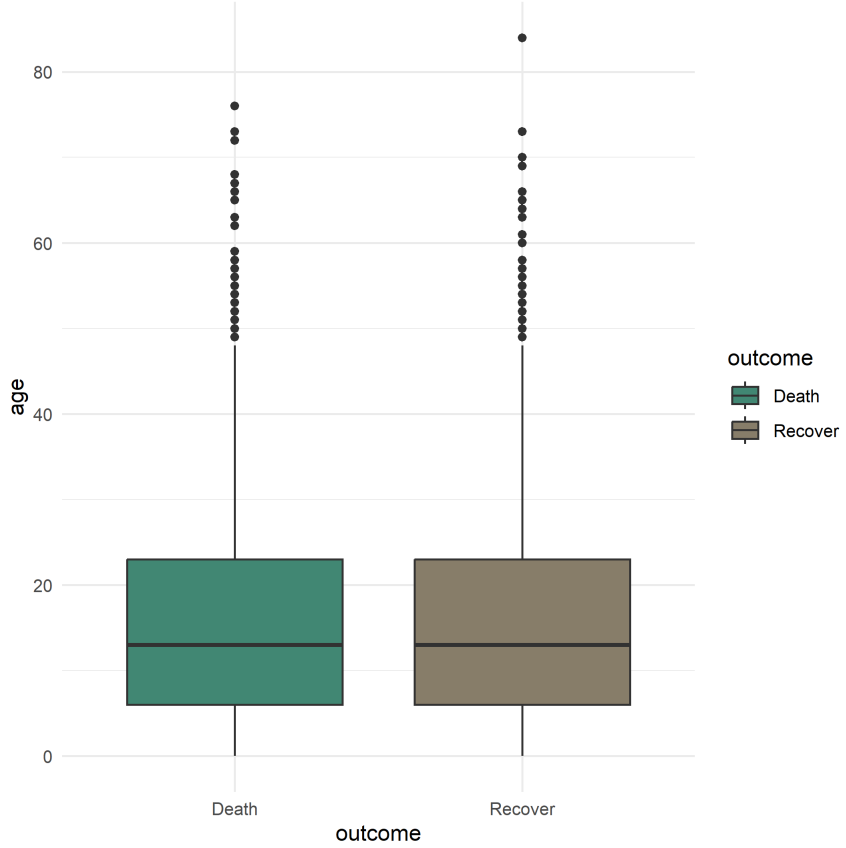
Puedes consultar la guía de colores aquí: https://r-charts.com/colors/

Cómo hacer boxplots avanzados?

Queremos representar la edad (`age`) de los grupos de `outcome` de nuestra tabla `surv_raw`.

Podemos aplicar un tema con `theme_*`:

```
ggplot(data = surv_raw %>% drop_na(outcome),
       mapping = aes(x = outcome, y = age,
                     fill = outcome)) +
  geom_boxplot() +
    scale_fill_manual(
    values = c("Death" = "aquamarine4",
               "Recover" = "bisque4")) +
  theme_minimal()
```



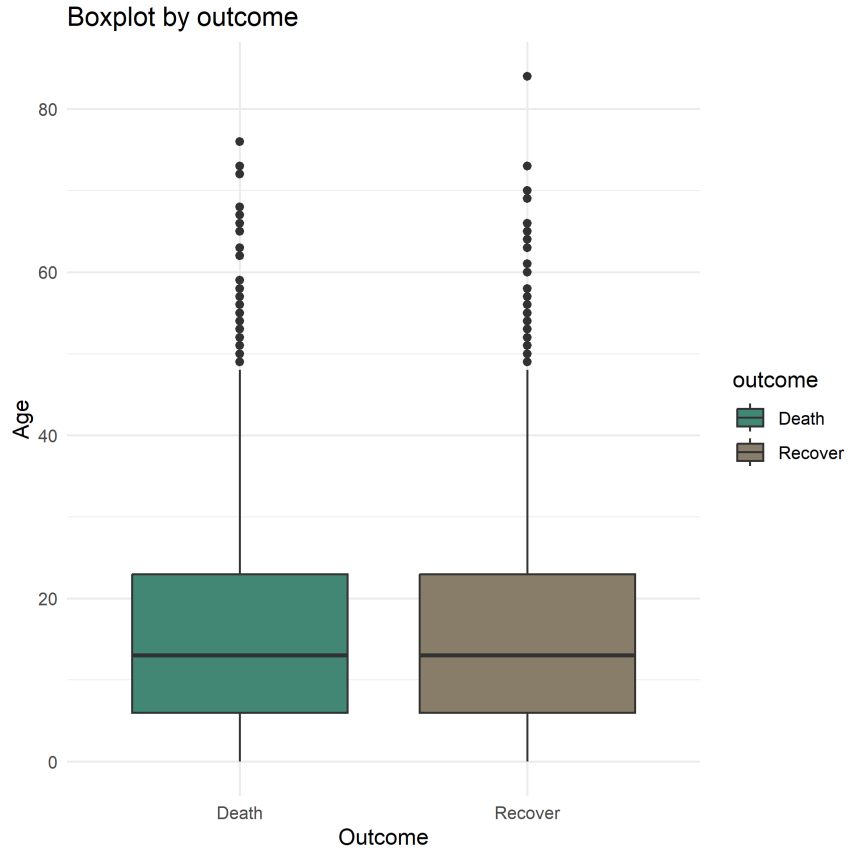
Puedes consultar los temas disponibles en: https://ggplot2.tidyverse.org/reference/ggtheme.html

Cómo hacer boxplots avanzados?

Queremos representar la edad (`age`) de los grupos de `outcome` de nuestra tabla `surv_raw`.

Con `labs()`, indicamos los títulos para los ejes y un título general.

```
ggplot(data = surv_raw %>% drop_na(outcome),
       mapping = aes(x = outcome, y = age,
                     fill = outcome)) +
  geom_boxplot() +
    scale_fill_manual(
    values = c("Death" = "aquamarine4",
               "Recover" = "bisque4")) +
  theme_minimal() +
  labs(y = "Age", x = "Outcome",
       title = "Boxplot by outcome")
```

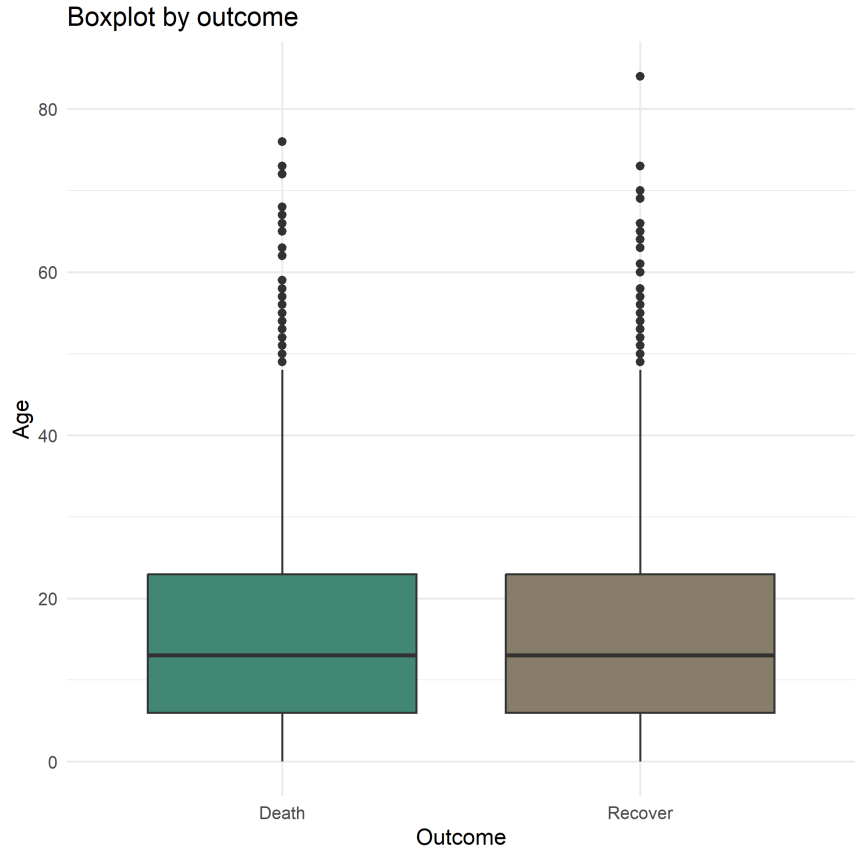


Cómo hacer boxplots avanzados?

Queremos representar la edad (`age`) de los grupos de `outcome` de nuestra tabla `surv_raw`.

Con `theme()`, podemos aplicar cualquier modificación al gráfico (¡avanzado!). Por ejemplo, indicar que no queremos la leyenda.

```
ggplot(data = surv_raw %>% drop_na(outcome),
       mapping = aes(x = outcome, y = age,
                     fill = outcome)) +
  geom_boxplot() +
    scale_fill_manual(
    values = c("Death" = "aquamarine4",
               "Recover" = "bisque4")) +
  theme_minimal() +
  labs(y = "Age", x = "Outcome",
       title = "Boxplot by outcome") +
  theme(legend.position = "none")
```

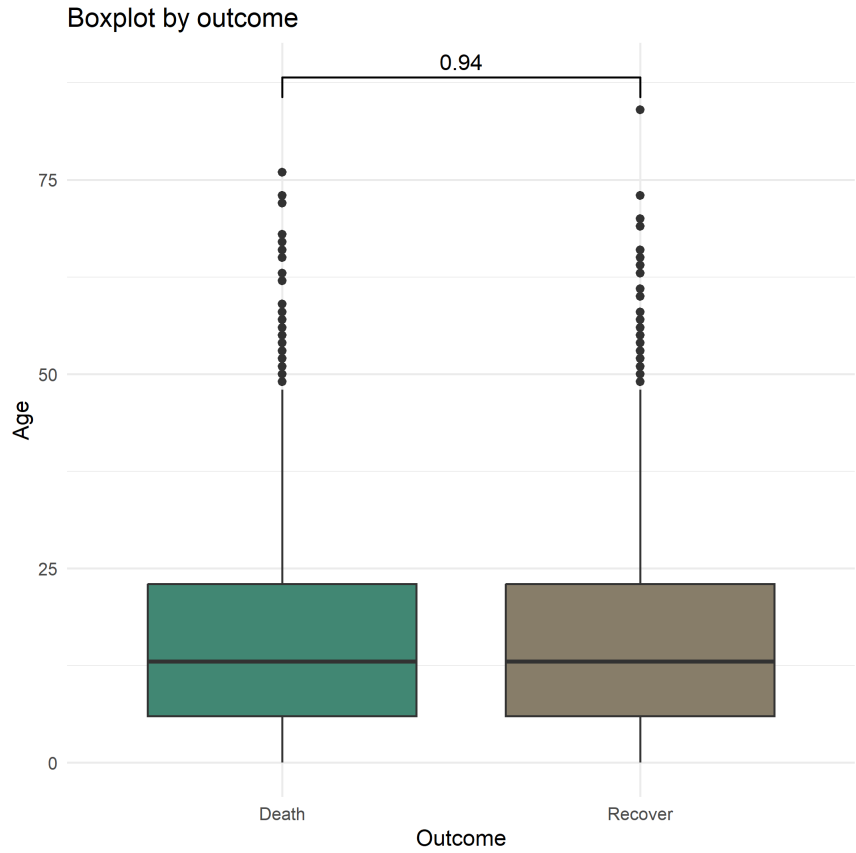


Cómo hacer boxplots avanzados?

Queremos representar la edad (age) de los grupos de outcome de nuestra tabla surv_raw.

Podemos agregar la significancia con el paquete ggsignif.

```
library(ggsignif)

ggplot(data = surv_raw %>% drop_na(outcome),
       mapping = aes(x = outcome, y = age,
                     fill = outcome)) +
  geom_boxplot() +
    scale_fill_manual(
    values = c("Death" = "aquamarine4",
               "Recover" = "bisque4")) +
  theme_minimal() +
  labs(y = "Age", x = "Outcome",
       title = "Boxplot by outcome") +
  theme(legend.position = "none") +
  geom_signif(comparisons =
                list(c("Death", "Recover")),
              test = "wilcox.test")
```



Cómo hacer un histograma avanzado?

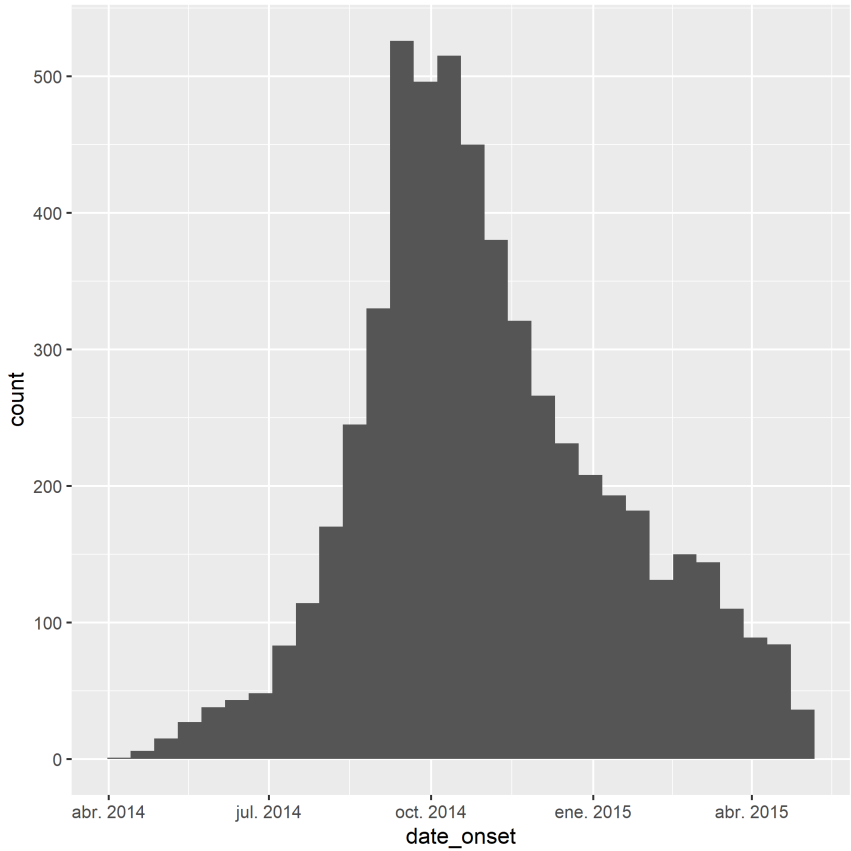
Queremos hacer una curva epidémica para cada hospital.

```
ggplot(data = surv_raw)
```

Cómo hacer un histograma avanzado?

Queremos hacer una curva epidémica para cada hospital.

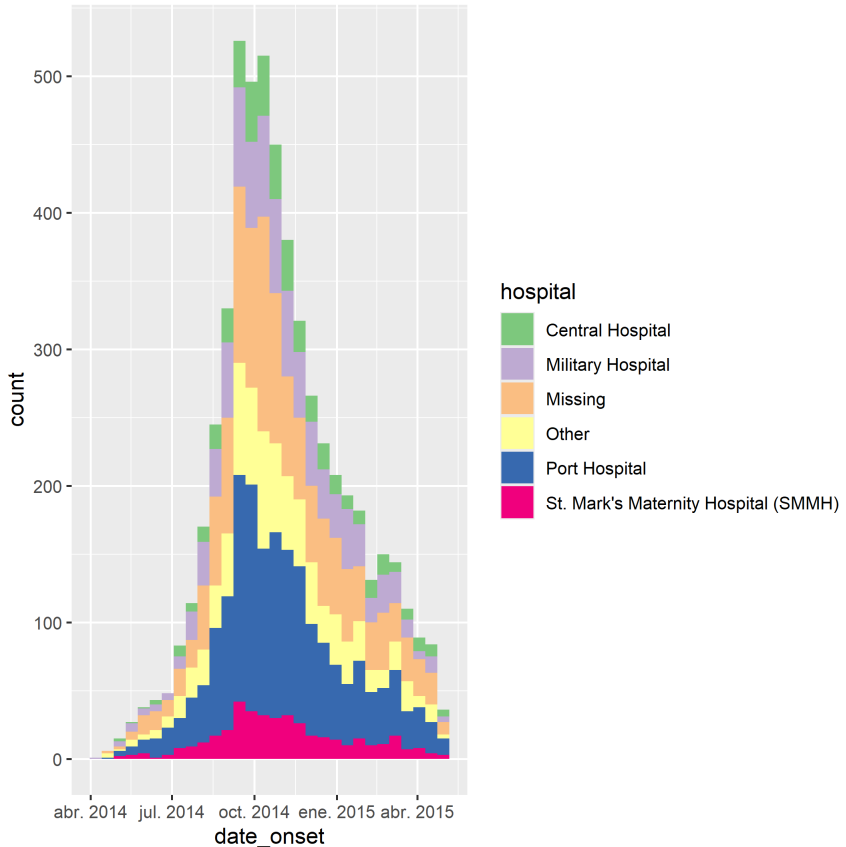
```
ggplot(data = surv_raw) +
    geom_histogram(aes(x = date_onset))
```



Cómo hacer un histograma avanzado?

Queremos hacer una curva epidémica para cada hospital.

```
ggplot(data = surv_raw) +
    geom_histogram(aes(x = date_onset,
                       fill = hospital)) +
   scale_fill_brewer(type = "qual",
                     palette = 1)
```

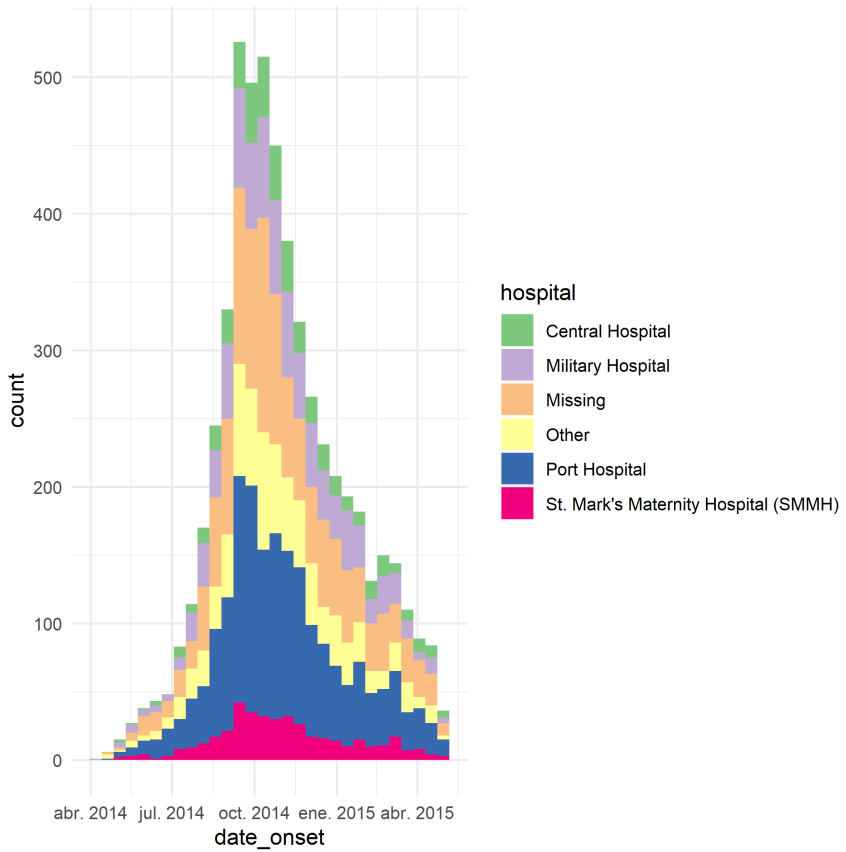


Hemos utilizado una paleta de colores preexistente. Puedes consultarlas en: https://ggplot2.tidyverse.org/reference/scale_brewer.html

Cómo hacer un histograma avanzado?

Queremos hacer una curva epidémica para cada hospital.

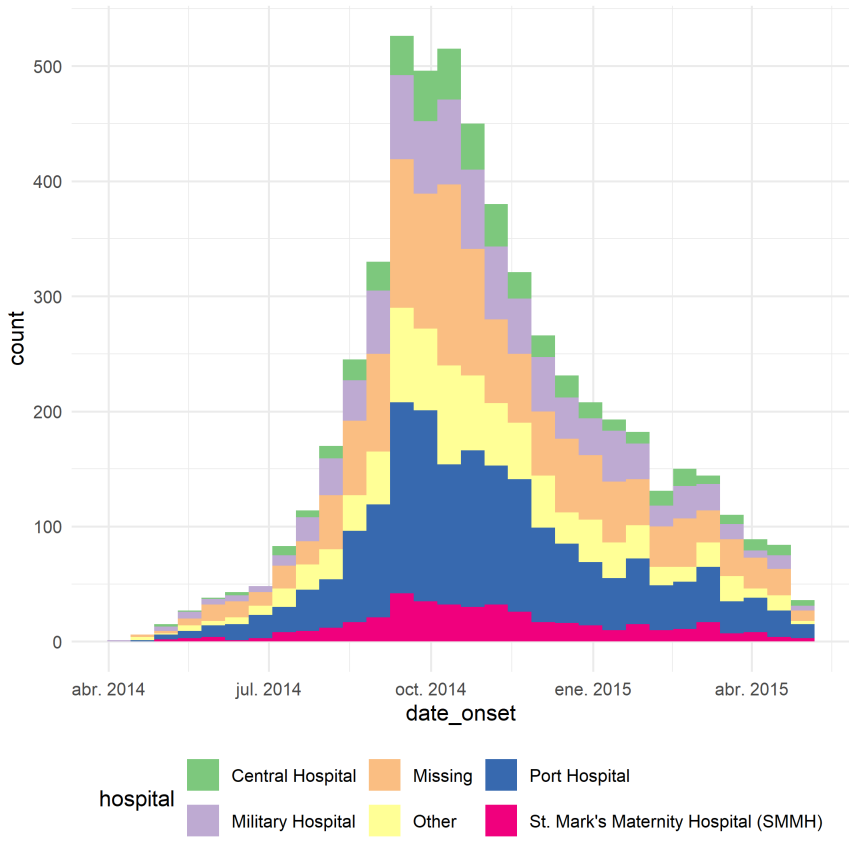
```
ggplot(data = surv_raw) +
    geom_histogram(aes(x = date_onset,
                       fill = hospital)) +
   scale_fill_brewer(type = "qual", palette =
  theme_minimal()
```



Cómo hacer un histograma avanzado?

Queremos hacer una curva epidémica para cada hospital.

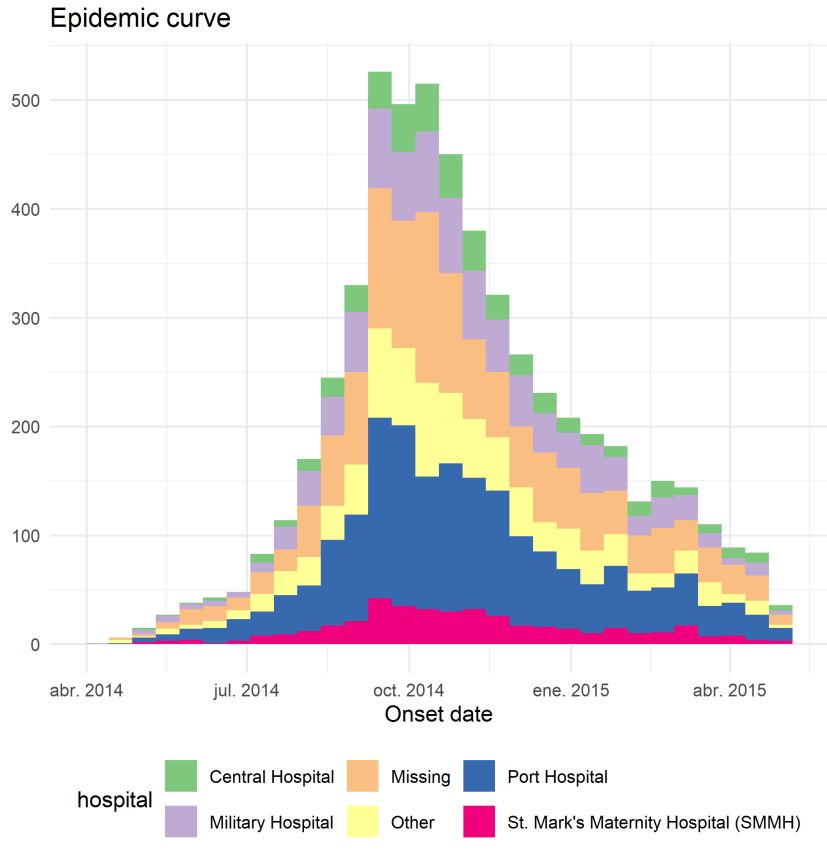
```
ggplot(data = surv_raw) +
    geom_histogram(aes(x = date_onset,
                       fill = hospital)) +
   scale_fill_brewer(type = "qual", palette =
  theme_minimal() +
  theme(legend.position = "bottom")
```



Cómo hacer un histograma avanzado?

Queremos hacer una curva epidémica para cada hospital.

```
ggplot(data = surv_raw) +
    geom_histogram(aes(x = date_onset,
                       fill = hospital)) +
   scale_fill_brewer(type = "qual", palette =
  theme_minimal() +
  theme(legend.position = "bottom") +
  labs(x = "Onset date", y = "",
       title = "Epidemic curve")
```



Cómo hacer un scatterplot?

Queremos representar la relación entre altura y peso de los pacientes.

```
library(dplyr)

# Calcular los cuantiles para 5 grupos
q <- seq(0, 1, 1/5)
age_quints <- quantile(surv_raw$age, probs = q, na.rm = TRUE)
age_quints
```

```
##   0%  20%  40%  60%  80% 100%
##    0    5   10   17   26   84
```

```
# Crear etiquetas con los rangos de edad
age_labels <- paste0("(",age_quints[-length(age_quints)], "-", age_quints[-1],"]")

age_labels
```

```
## [1] "(0-5]"   "(5-10]"  "(10-17]" "(17-26]" "(26-84]"
```

Cómo hacer un scatterplot?

Queremos representar la relación entre altura y peso de los pacientes.

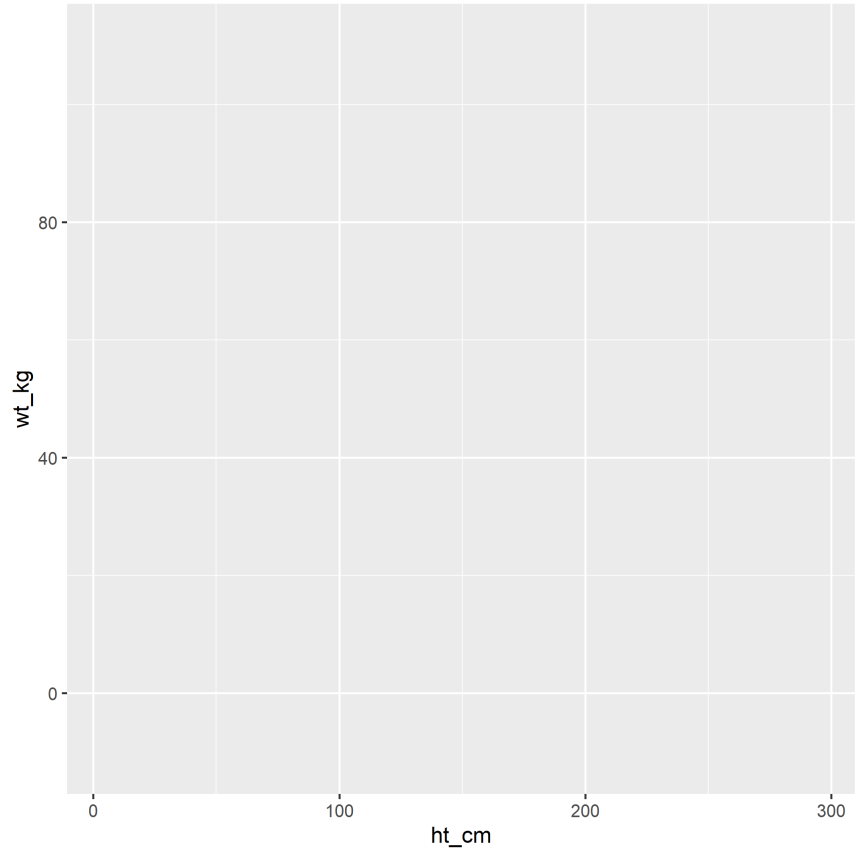
```
# Crear la columna age_cat_5 con etiquetas de rango
surv_raw <- surv_raw %>%
  mutate(age_cat_5 = cut(age,
                         breaks = age_quints,
                         include.lowest = TRUE,
                         labels = age_labels))
surv_raw[,c("case_id", "age", "age_cat_5")]
```

```
## # A tibble: 5,888 × 3
##    case_id   age age_cat_5
##    <chr>   <dbl> <fct>
##  1 5fe599      2 (0-5]
##  2 8689b7      3 (0-5]
##  3 11f8ea     56 (26-84]
##  4 b8812a     18 (17-26]
##  5 893f25      3 (0-5]
##  6 be99c8     16 (10-17]
##  7 07e3e8     16 (10-17]
##  8 369449      0 (0-5]
##  9 f393b4     61 (26-84]
```

Cómo hacer un scatterplot?

Queremos representar la relación entre altura y peso de los pacientes.

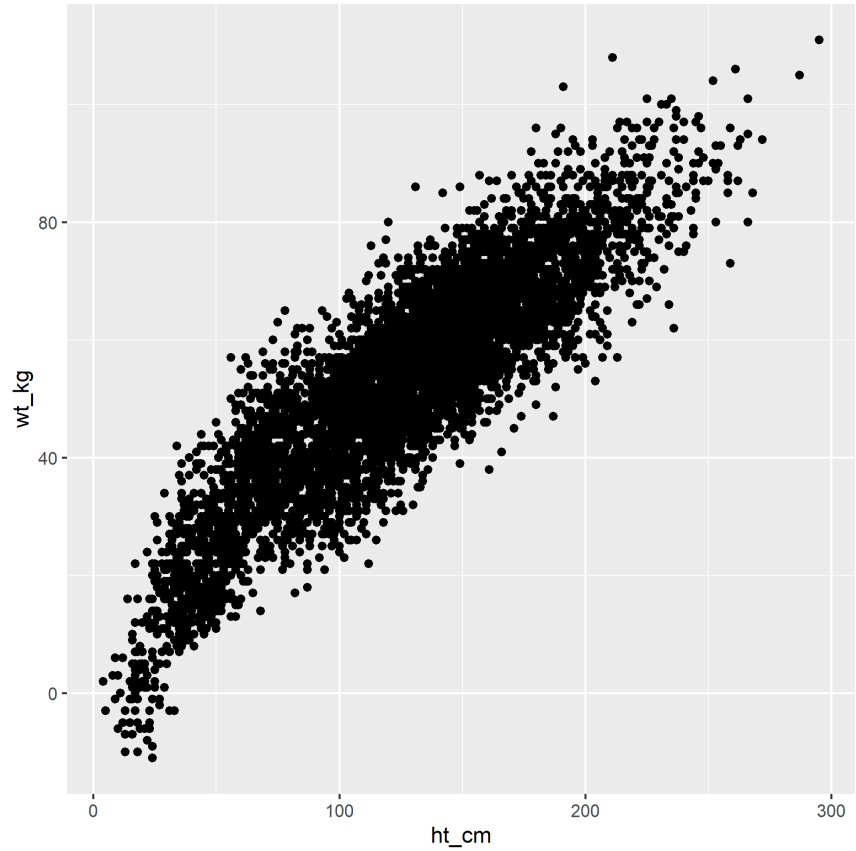
```
ggplot(surv_raw,
       aes(x = ht_cm, y = wt_kg))
```



Cómo hacer un scatterplot?

Queremos representar la relación entre altura y peso de los pacientes.

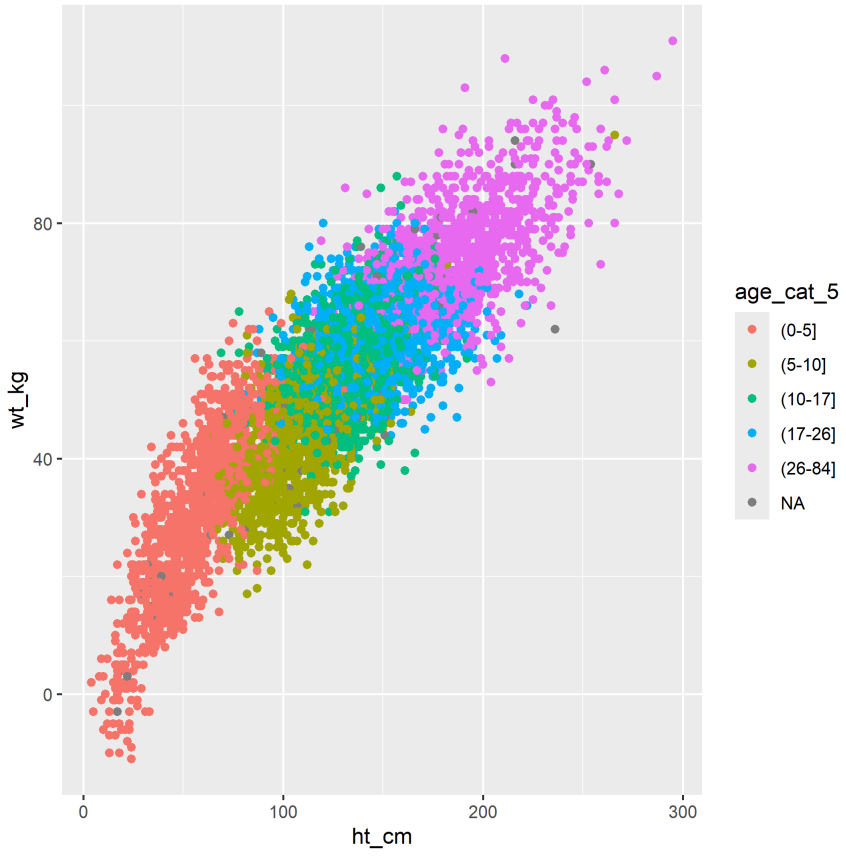
```
ggplot(surv_raw,
       aes(x = ht_cm, y = wt_kg)) +
    geom_point()
```



Cómo hacer un scatterplot?

Queremos representar la relación entre altura y peso de los pacientes.

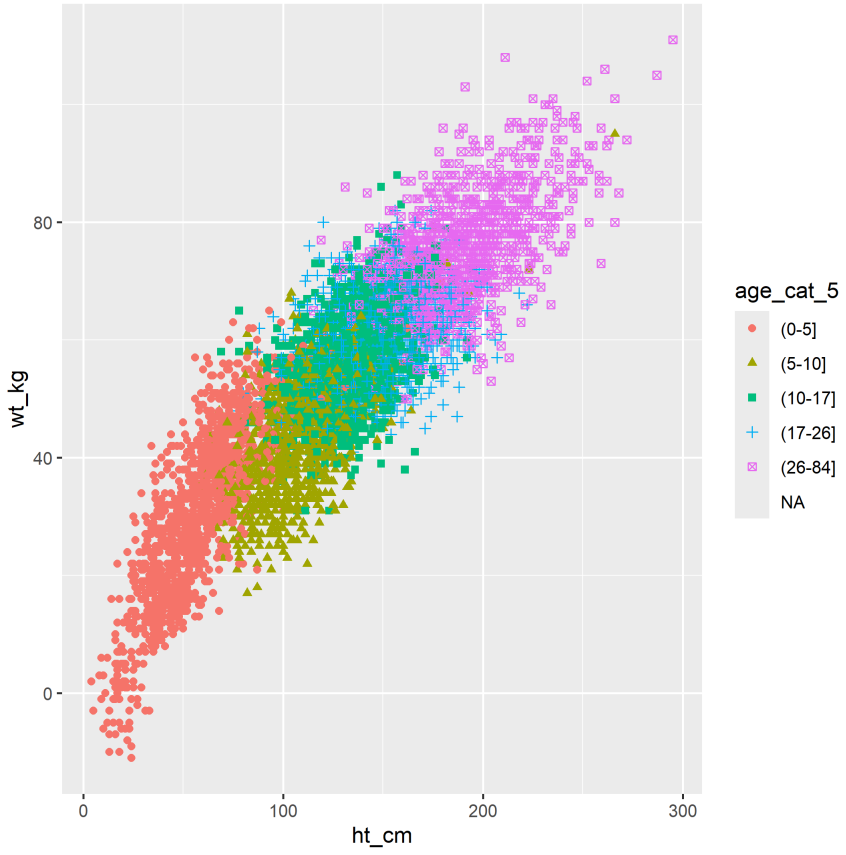
```
ggplot(surv_raw,
       aes(x = ht_cm, y = wt_kg,
           color=age_cat_5)) +
    geom_point()
```



Cómo hacer un scatterplot?

Queremos representar la relación entre altura y peso de los pacientes.

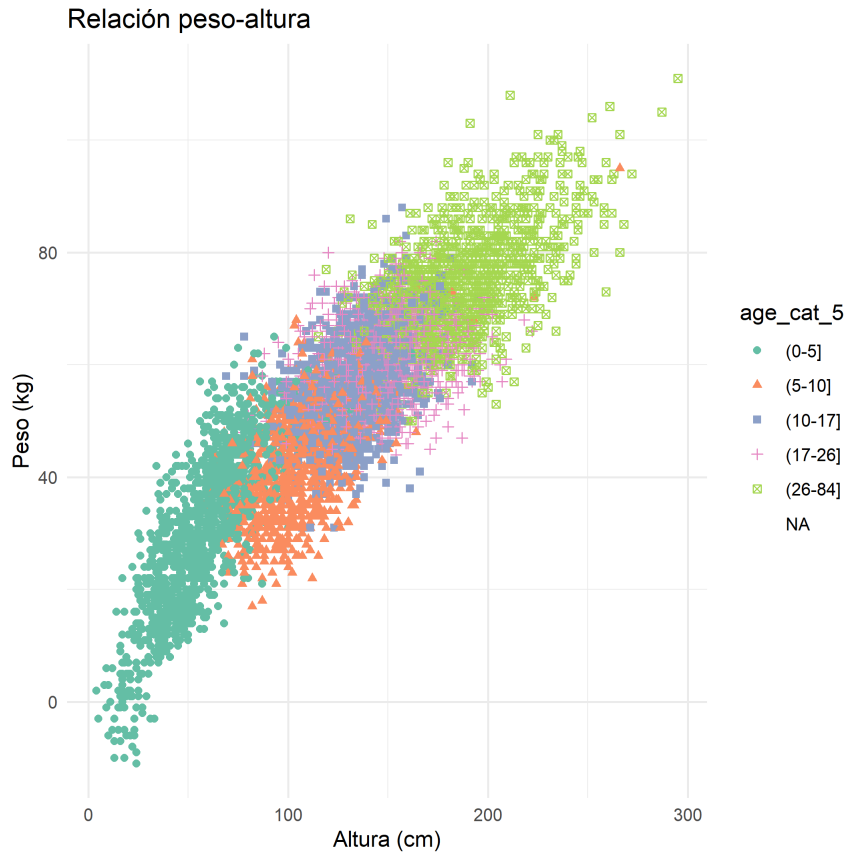
```
ggplot(surv_raw,
       aes(x = ht_cm, y = wt_kg,
           color=age_cat_5,
           shape=age_cat_5)) +
    geom_point()
```



Cómo hacer un scatterplot?

Queremos representar la relación entre altura y peso de los pacientes.

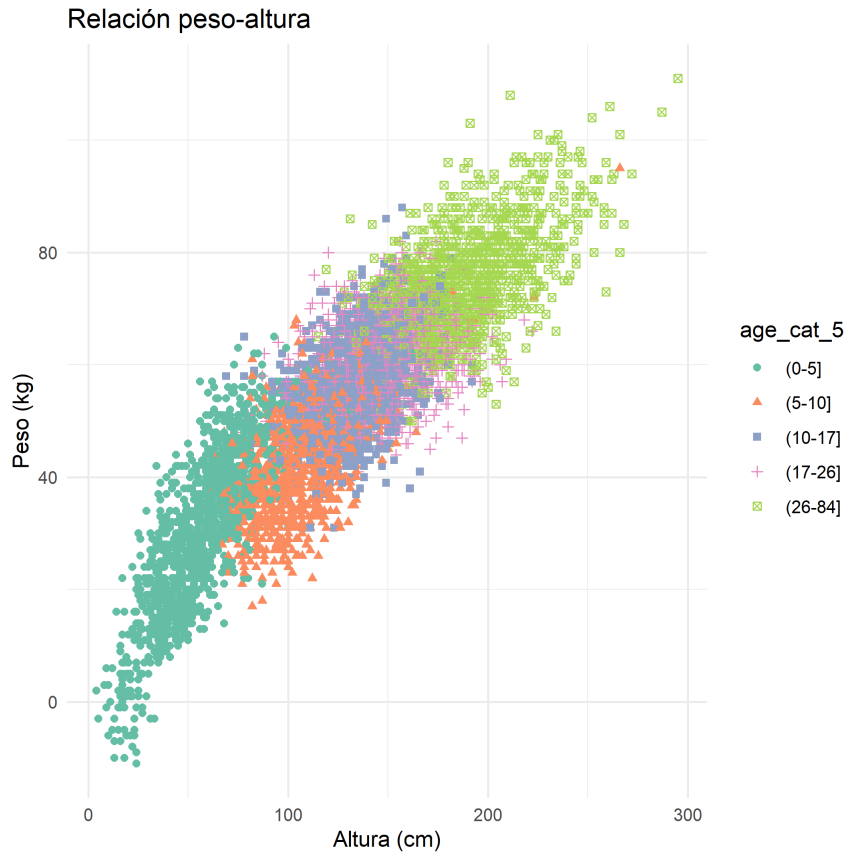
```
ggplot(surv_raw,
       aes(x = ht_cm, y = wt_kg,
           color=age_cat_5,
           shape=age_cat_5)) +
    geom_point() +
    labs(title = "Relación peso-altura",
         x = "Altura (cm)",
         y = "Peso (kg)") +
    scale_color_brewer(palette = "Set2") +
    theme_minimal()
```



Cómo hacer un scatterplot?

Queremos representar la relación entre altura y peso de los pacientes.

```
ggplot(surv_raw %>% drop_na(age_cat_5),
       aes(x = ht_cm, y = wt_kg,
           color=age_cat_5,
           shape=age_cat_5)) +
    geom_point() +
    labs(title = "Relación peso-altura",
         x = "Altura (cm)",
         y = "Peso (kg)") +
    scale_color_brewer(palette = "Set2") +
    theme_minimal()
```



Cómo hacer un scatterplot?

Queremos representar la relación entre altura y peso de los pacientes.

```
ggplot(surv_raw %>% drop_na(age_cat_5),
       aes(x = ht_cm, y = wt_kg,
           color=age_cat_5,
           shape=age_cat_5)) +
    geom_point() +
    labs(title = "Relación peso-altura",
         x = "Altura (cm)",
         y = "Peso (kg)",
         color="Categoría edad",
         shape="Categoría edad") +
    scale_color_brewer(palette = "Set2") +
    theme_minimal()
```

03

Librerías especializadas

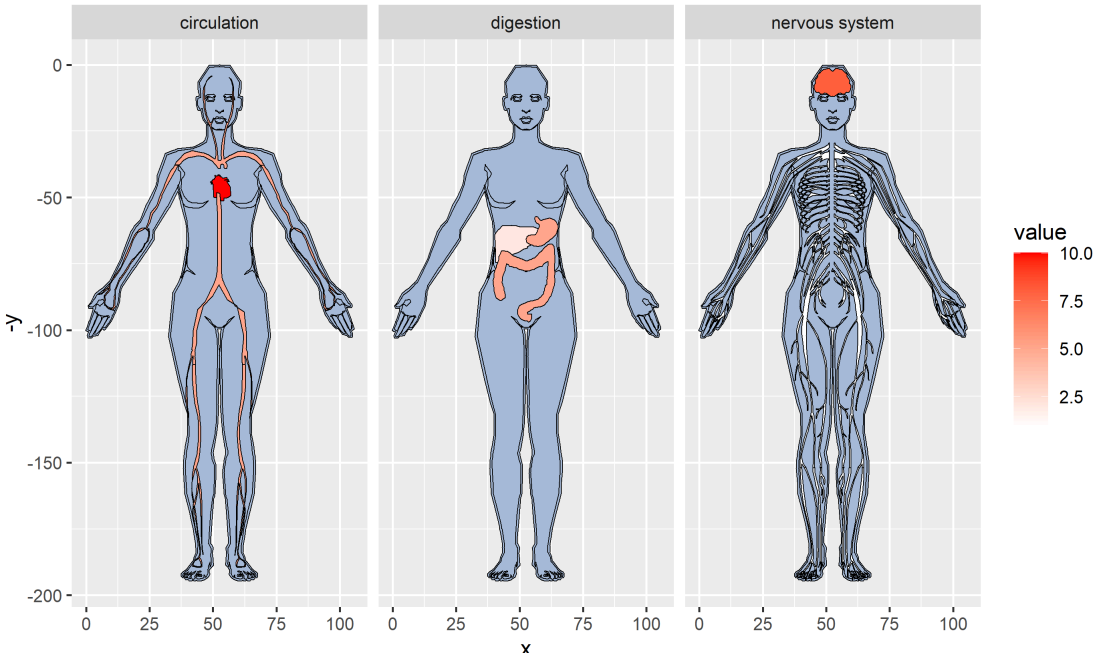
Bibliotecas y software especializados

- Paquetes de software integrados
- Javascript
  - BioJS
- Bibliotecas de R
  - Repositorios especializados Bioconductor
  - Extensiones de `ggplot2`
  - `htmlwidgets`, algunos utilizando bibliotecas de BioJS

Estructuras anatómicas: gganatogram

https://github.com/jespermaag/gganatogram

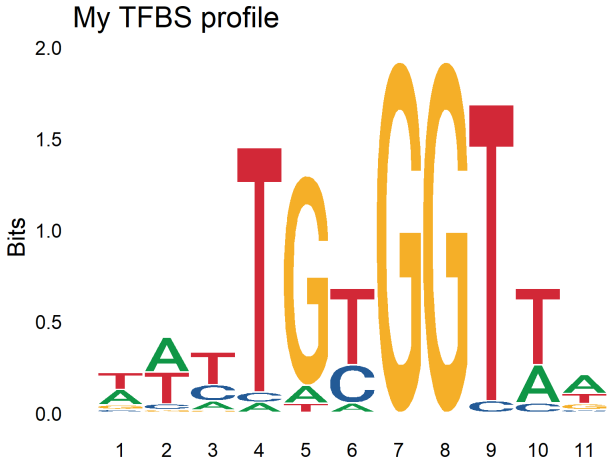
```
library(gganatogram)
gganatogram(data=organ_df, fillOutline='#a6bddb', organism='human',
            sex='female', fill="value") +
    scale_fill_gradient(low = "white", high = "red") +
    facet_wrap(~ type)
```

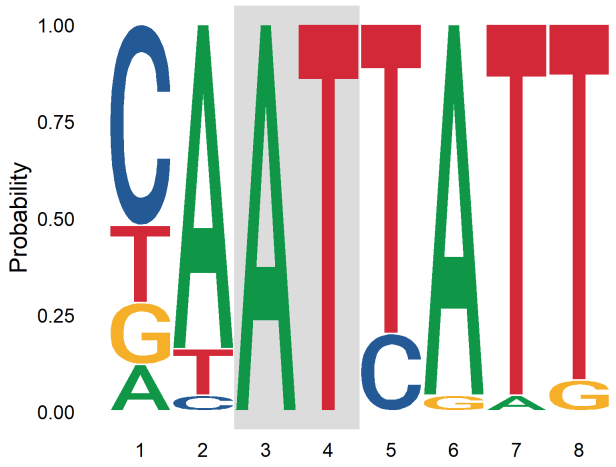


Logos: ggseqlogo

https://omarwagih.github.io/ggseqlogo/

```
library(ggseqlogo)
ggplot() + geom_logo(seqs_dna$MA0002.1) +
    theme_logo() + labs(title = "My TFBS profile")
ggplot() +
    annotate(geom = "rect", xmin = 2.5, xmax = 4.5,
             ymin = -Inf, ymax = Inf, alpha = 0.2) +
    geom_logo(seqs_dna$MA0008.1, method = "probability") +
    theme_logo()
```





Estructuras de genes: gggenes y gggenomes

https://github.com/wilkox/gggenes - https://thackl.github.io/gggenomes/

```
library(gggenes)
ggplot(example_genes, aes(xmin = start, xmax =
  geom_feature(
    data = example_features,
    aes(x = position, y = molecule, forward =
    ) +
  geom_feature_label(
    data = example_features,
    aes(x = position, y = molecule, label = nar
    ) +
  geom_gene_arrow() +
  geom_blank(data = example_dummies) +
  facet_wrap(~ molecule, scales = "free", ncol
  scale_fill_brewer(palette = "Set3") +
  theme_genes()
```

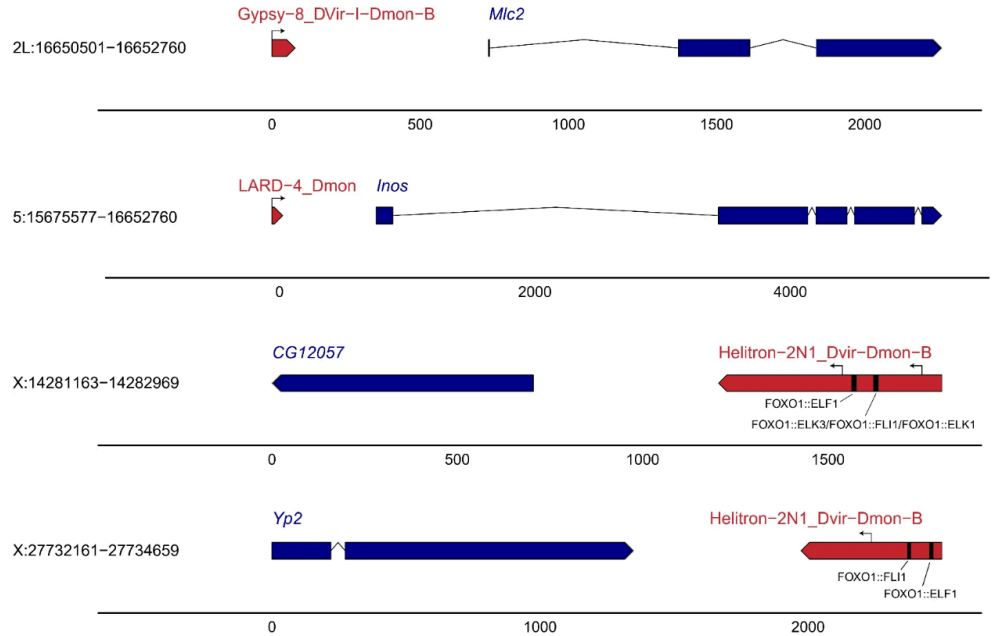


Figure 7 - Tahami, M.S., Vargas-Chavez, C., Poikela, N., Coronado-Zamora, M., et al. Transposable elements in Drosophila montana from harsh cold environments. Mobile DNA 15, 18 (2024). https://doi.org/10.1186/s13100-024-00328-7

Resumen

- La mayoría de los gráficos exploratorios y de comunicación básicos en biología se pueden generar con herramientas generales de gráficos estadísticos (`ggplot2`) --> Datos cuantitativos y cualitativos
    - Diagramas de dispersión (scatterplots)
    - Diagramas de barras (barplots)
    - Diagramas de caja (boxplots)
    - Histoggramas
    - Mapas de calor (heatmaps), ...
- La complejidad y las características de algunos datos biológicos requieren herramientas especializadas.
    - Si los requisitos son estáticos, las extensiones de `ggplot2` pueden ser útiles.
    - Revisa las herramientas utilizadas en estudios similares.