Visualización de datos con R - ggplot2

Jorge Mario Estrada A.

Presentación tomada de Applied Epi

Hoy: objetivos y agenda

En este módulo pretendemos ayudarle:

- Entender la {ggplot2} "Gramática de los gráficos"
- Construir gráficos sencillos de caja/escala/barra e histogramas
- Ajuste las escalas, los temas y las etiquetas de los gráficos

Time	Location	Topic
20 minutos	Principal	Sintaxis de **ggplot2** ("gramática de los gráficos")
10 minutos	Principal	Demostración en vivo
1h30 minutos	Sesiones de trabajo	Ejercicio: Un gráfico simple
20 minutos	Principal	Escalas, temas y etiquetas
1 hora	Breakouts	Ajustes del gráfico
10 minutos	Principal	Plenaria

Los tiempos de pausas se incorporan arriba aunque no se muestren

Visualización de datos con

Opciones de visualización en R

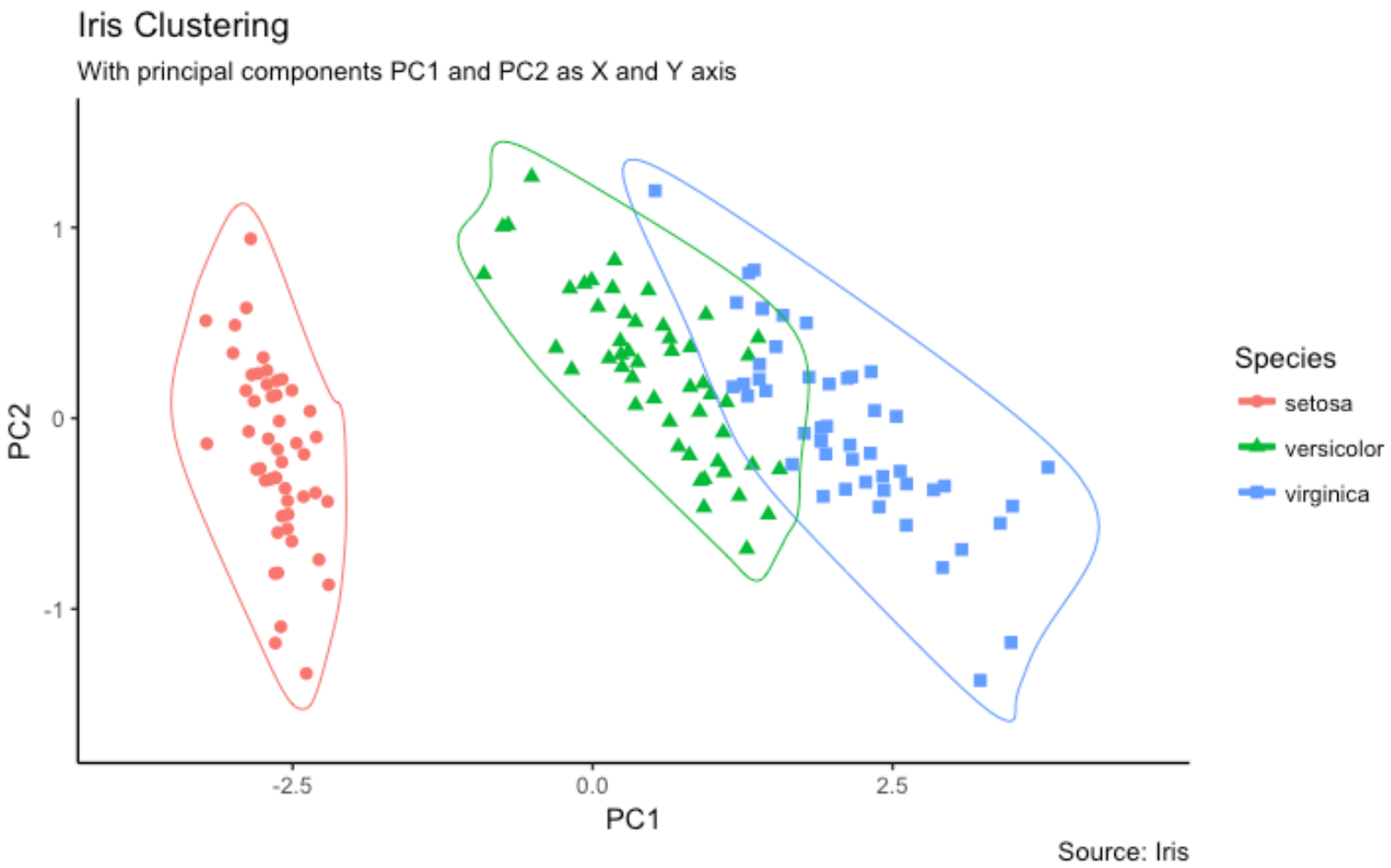
Hoy nos centramos en **ggplot2** porque:

- es bueno para la exploración rápida de datos multidimensionales
- produce resultados finales de muy **alta calidad**
- tiene una gramática bien estructurada => **alta coherencia**.
- se acompaña de muchos paquetes que amplían la funcionalidad

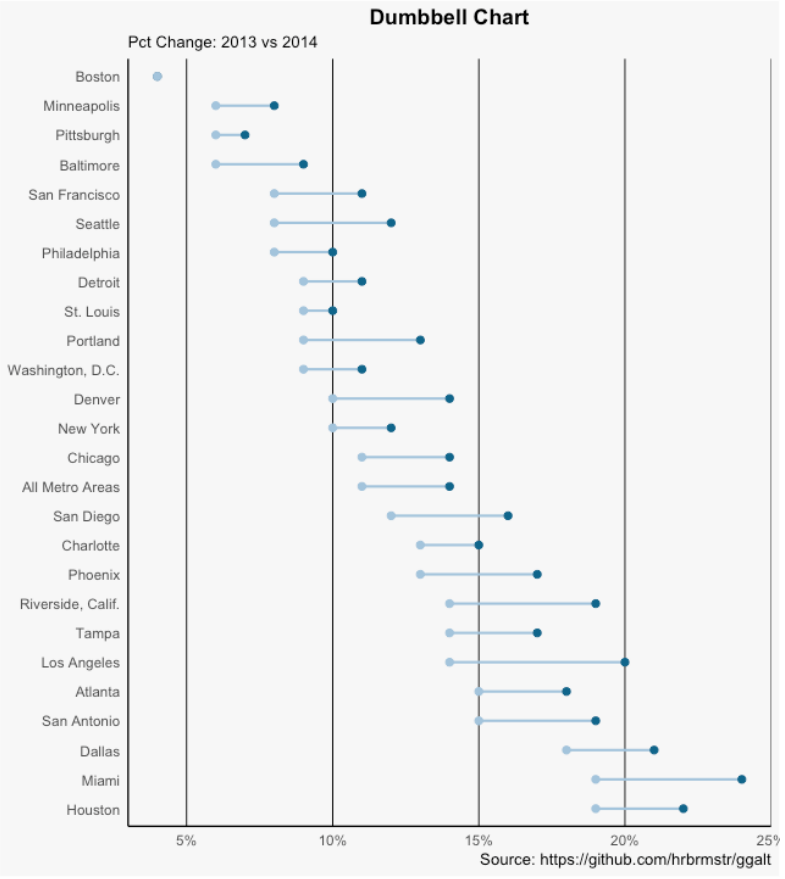
Consulte la galería de gráficos R para inspirarse.

Otras opciones de trazado son **base** R, **lattice** y **plotly**.

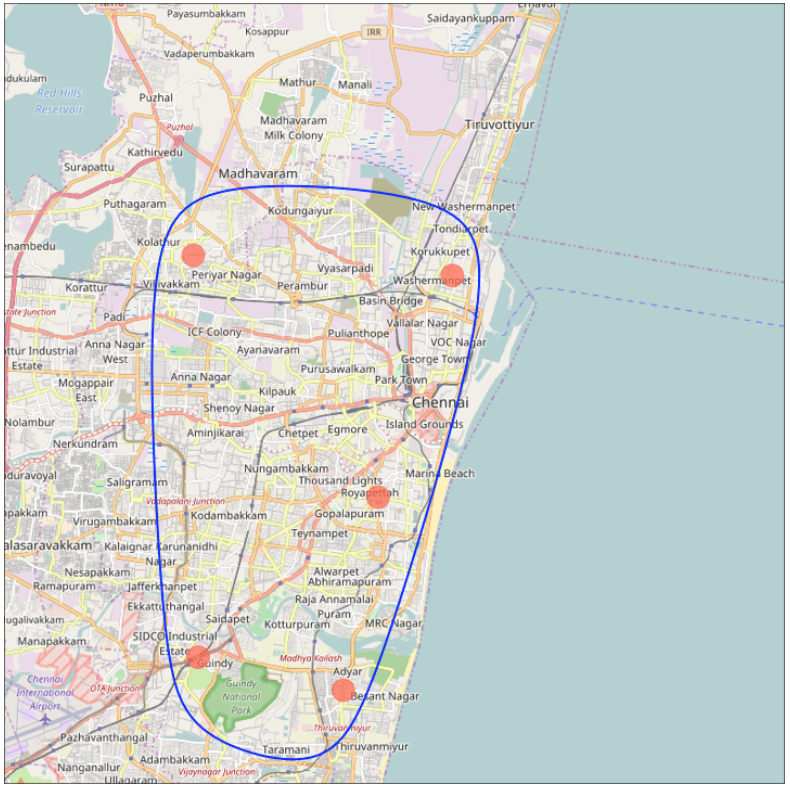
¿Se hizo con ggplot?



¿Se hizo con ggplot?

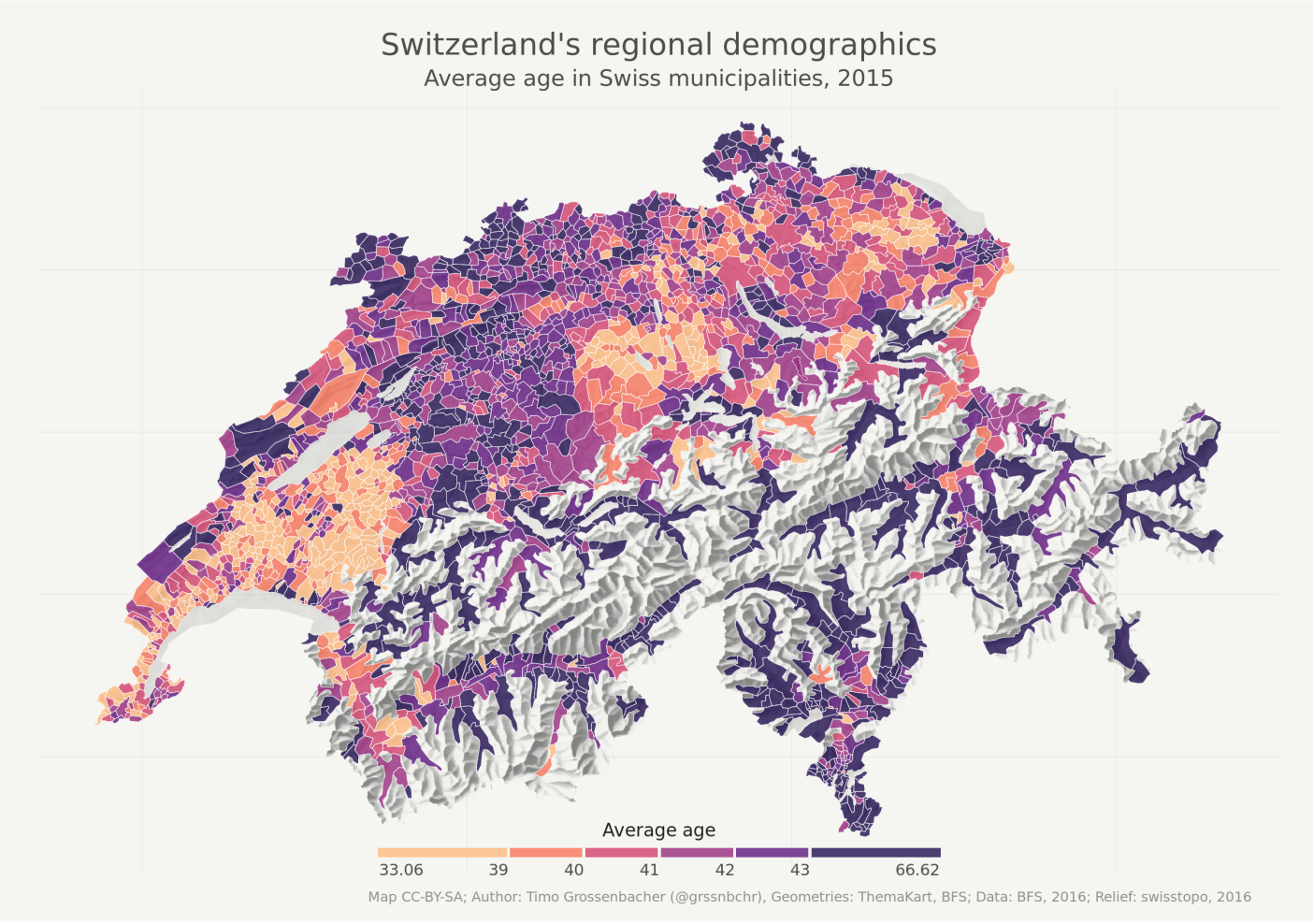


¿Se hizo con ggplot?

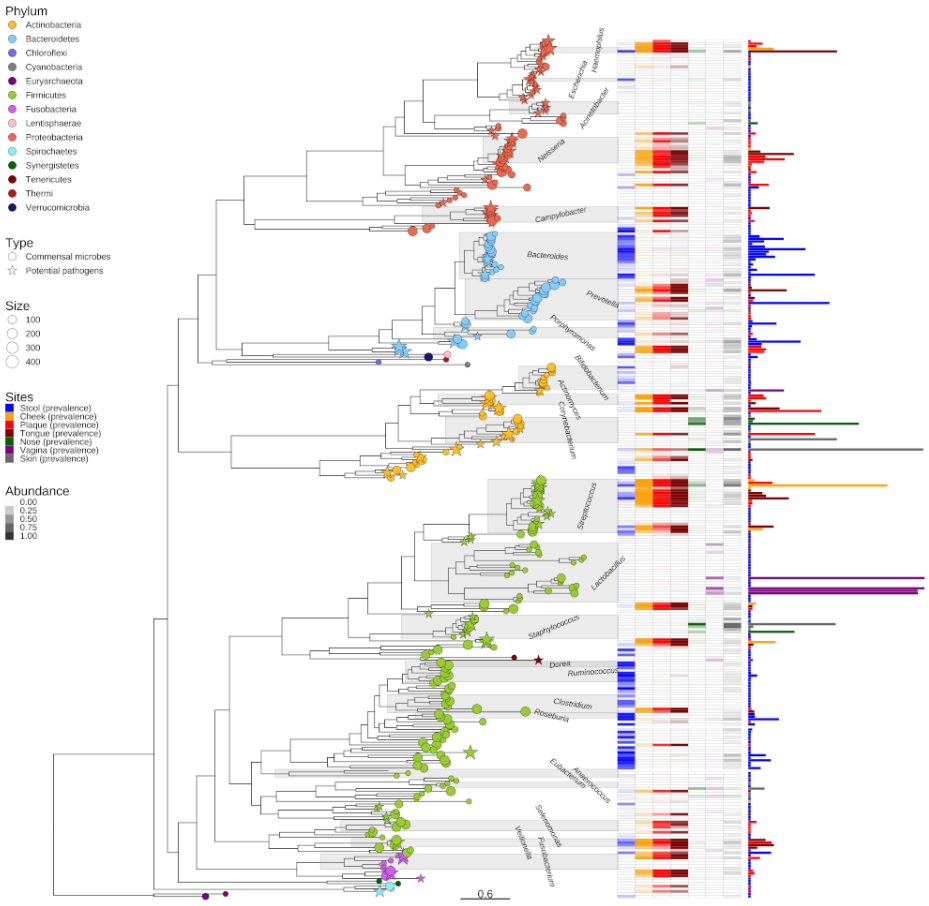


¿Se hizo con ggplot?

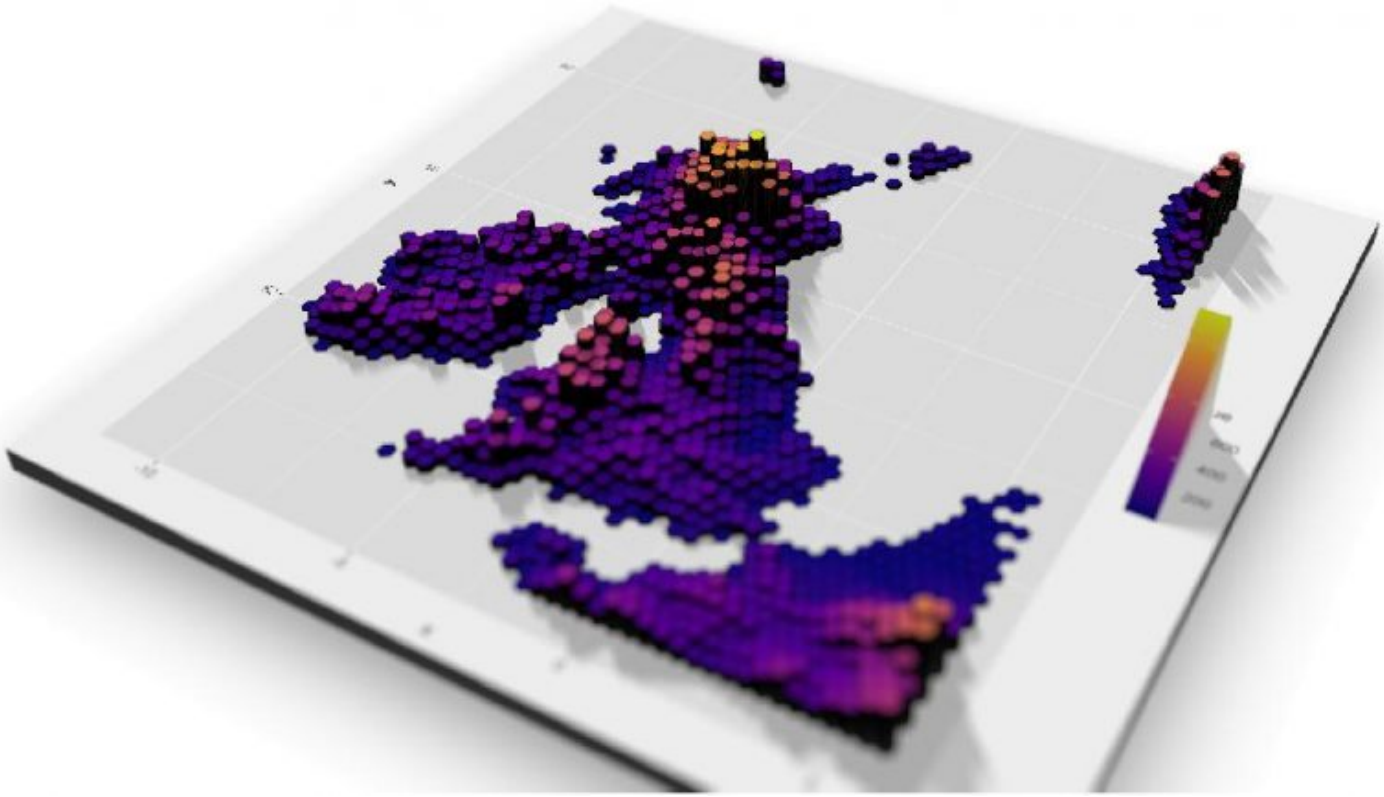
¿Se hizo con ggplot?



¿Se hizo con ggplot?



¿Se hizo con ggplot?



¿Se hizo con ggplot?

¿Se hizo con ggplot?

¿Gg-qué?

- El *paquete* {ggplot2} es la herramienta de visualización de datos más popular en R
- La función `ggplot()` es el núcleo de este paquete
- Todo este enfoque se conoce coloquialmente como "ggplotting"
- Los gráficos realizados se llaman a veces cariñosamente "ggplots"

Hay un gran número de paquetes que amplian las funcionalidades de {ggplot2} como {gganimate}, {ggthemr}, {ggdendro}, {gghighlight}, {ggforce}...

Pregunta adicional: ¿Qué representa la "gg" en estos nombres?

Gramática de los gráficos

Construir un objeto del gráfico "añadiendo" comandos uno encima de otro que especifiquen las capas del gráfico y los elementos de diseño

El orden de las capas suele ser el siguiente:

1) **Abra el gráfico** con el comando `ggplot()` y especifique el conjunto de datos

2) **"Mapear"** (especifique) las columnas de datos a las características "estéticas" del gráfico, tales como ejes, color, tamaño, forma, relleno, transparencia

3) **Añadir (+) y capas "geom "** que visualizan los datos geométricamente como formas

4) **Modificar las "escalas "**, como la escala de colores o las interrupciones del eje Y

5) **Añadir elementos de "diseños temáticos "** como etiquetas de ejes, título, leyenda, fuentes, tamaños de texto, temas de fondo o rotación de ejes.

Estas capas se "añaden" secuencialmente con los símbolos **+**. Los comandos {ggplot2} pueden ser bastante largos.

Abrir el gráfico

```
ggplot()
```

`ggplot()` crea un lienzo vacío. (un cuadro en blanco)

Añadir los datos

```
ggplot(data = surv)
```

Asigna el data frame a ser utilizado.

Alternativamente, utilice el operador pipe `%>%` para "canalizar" un marco de datos *en* `ggplot()`.

```
surv %>%
  ggplot()
```

Añadir los datos

```
ggplot(
  data = surv)
```

Las nuevas líneas y las sangrías no afectarán a la ejecución del código.

Pueden hacer que los comandos más largos sean más fáciles de leer...

Mapeos con `aes()`

```
ggplot(
  data = surv,
  mapping = aes())
```

La "estética" de los gráficos son características como la posición, el color, la forma...

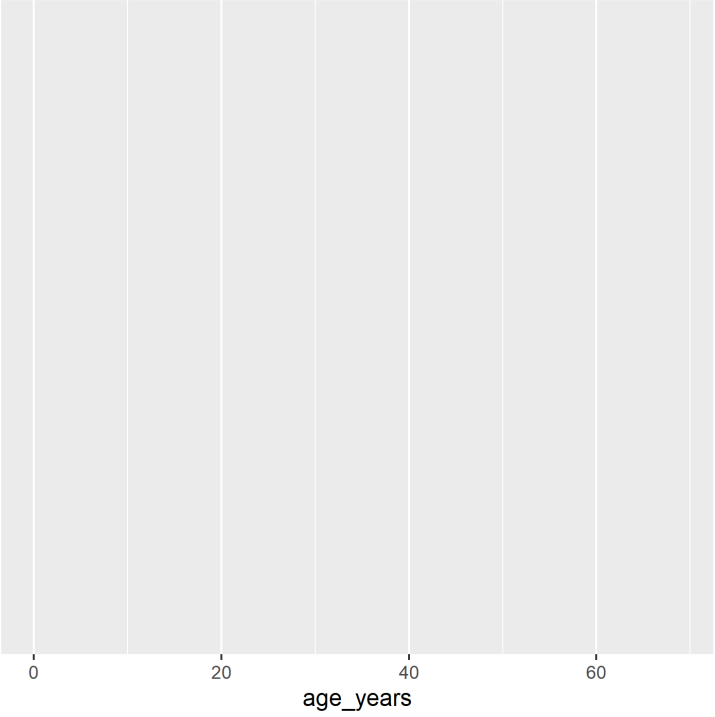
`mapping = aes()` asigna la "estética" a las columnas de los datos.

Mapeos con `aes()`

```
ggplot(
  data = surv,
  mapping = aes(
    x = age_years
    ))
```

Los mapeos estéticos se colocan dentro de `aes()`.

Dos mapeos básicos son los ejes a las columnas, a través de: `x =`

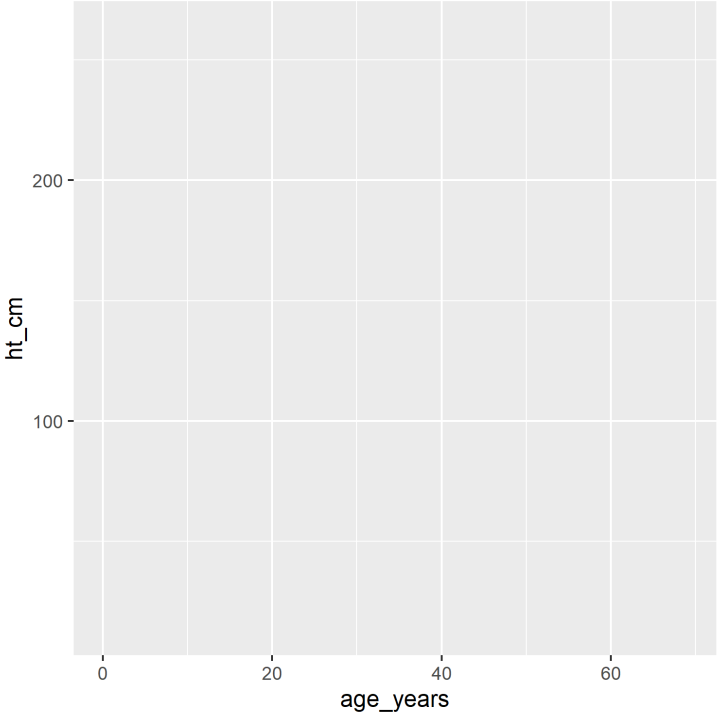


Mapeos con `aes()`

```
ggplot(
  data = surv,
  mapping = aes(
    x = age_years,
    y = ht_cm))
```

Los mapeos estéticos se colocan dentro de `aes()`.

Dos mapeos básicos son los ejes a las columnas, a través de: `x =` y `y =`

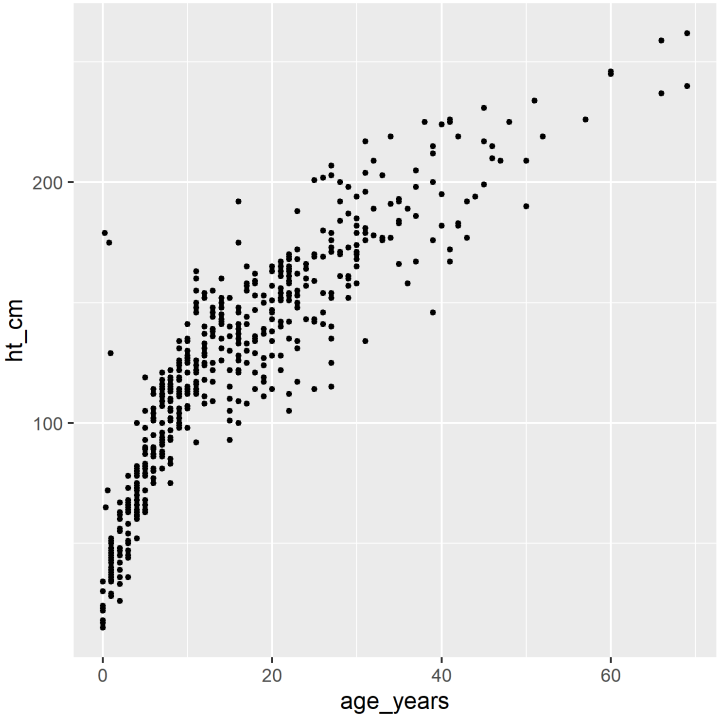


Añadir geometría

```
ggplot(
  data = surv,
  mapping = aes(
    x = age_years,
    y = ht_cm)) +
geom_point()
```

Los datos se visualizan mediante comandos "geom", como `geom_point()`.

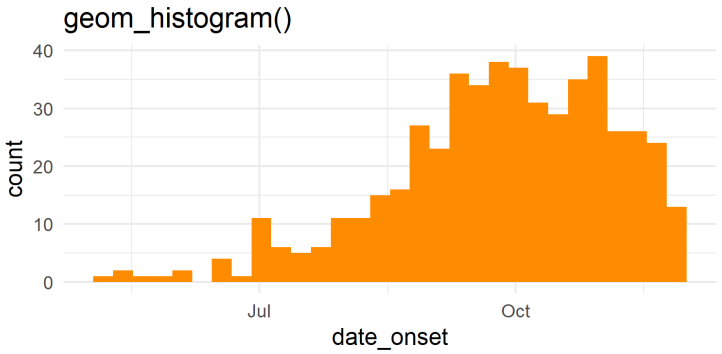
Estos comandos se "añaden" con un **+** al comando `ggplot()`.

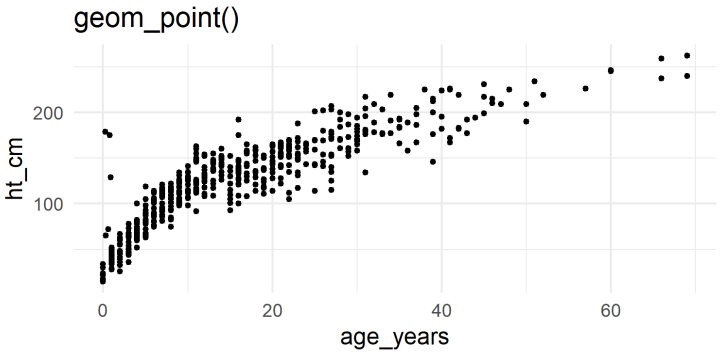


Geometrías

Algunos "geoms" clásicos son:

Geometría	Geom
Histogramas	`geom_histogram()`
Puntos	`geom_point()`



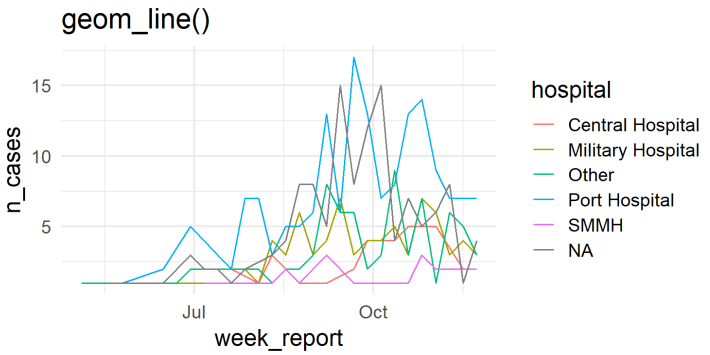


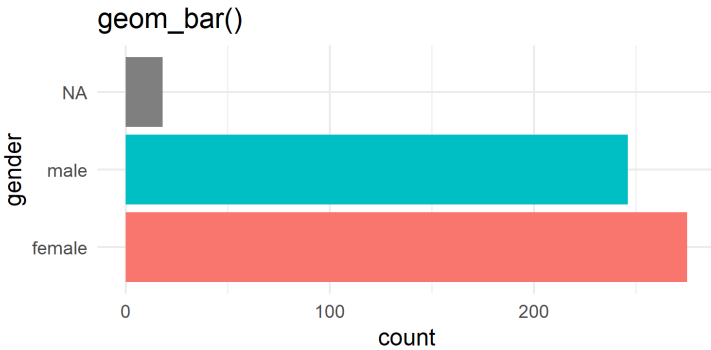
Lista completa aquí

Geometrías

Algunos "geoms" clásicos son:

Geometría	Geom
Líneas	`geom_line()`
Gráficos de barras	`geom_bar()` or `geom_col()`



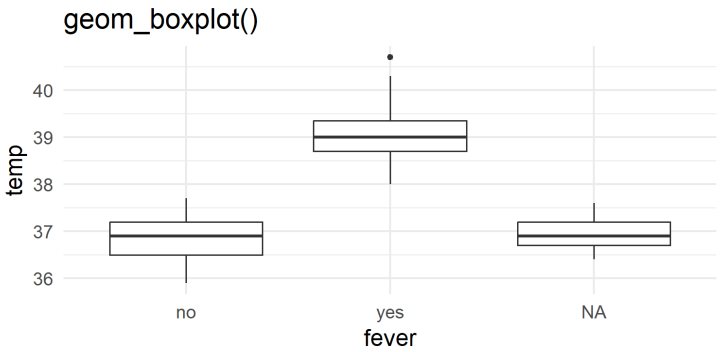


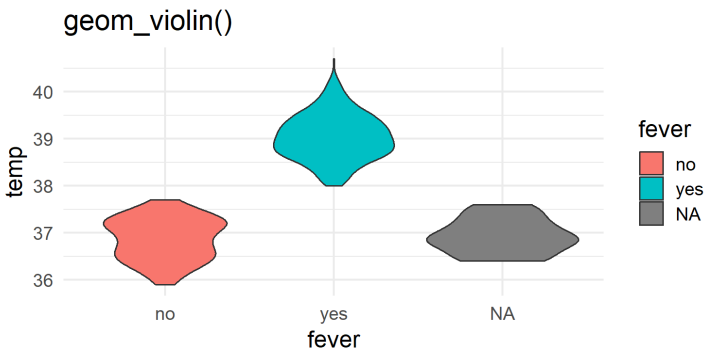
Lista completa aquí

Geometrías

Algunos "geoms" clásicos son:

Geometría	Geom
Boxplots	`geom_boxplot()`
Gráfico de violin	`geom_violin()`



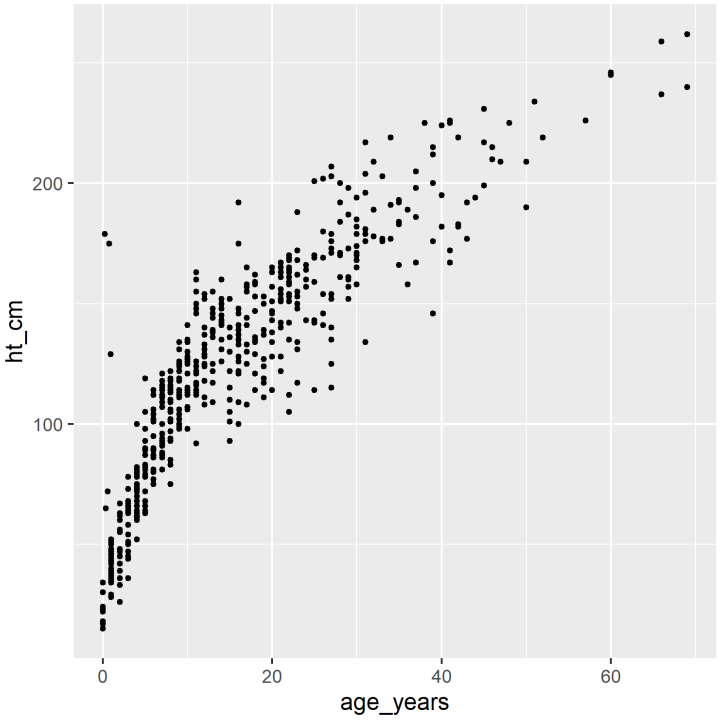


Lista completa aquí

Añadiendo Geoms

```
ggplot(
  data = surv,
  mapping = aes(
   x = age_years,
   y = ht_cm)) +
geom_point()
```

Con los ejes ahora mapeados, `geom_point()` muestra los datos como puntos.

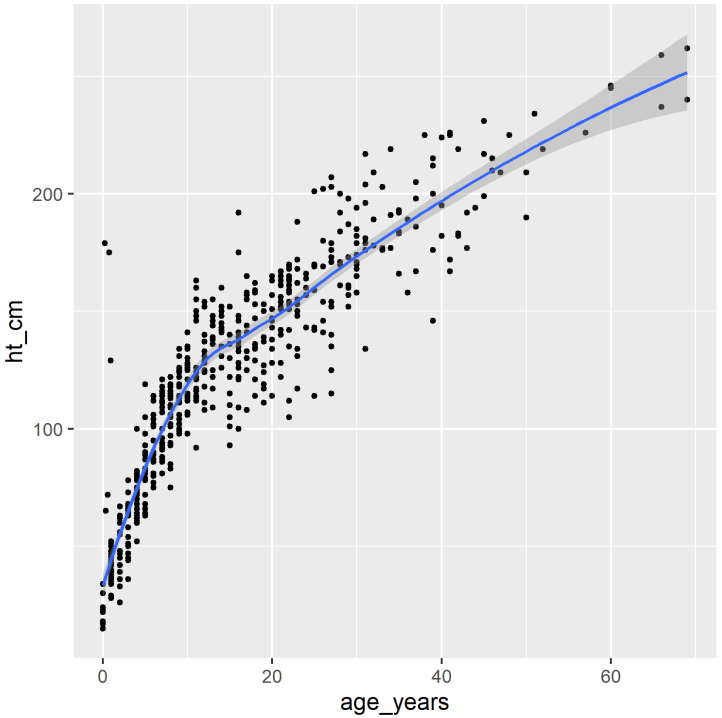


Añadiendo Geoms

```
ggplot(
  data = surv,
  mapping = aes(
   x = age_years,
   y = ht_cm)) +
geom_point() +
geom_smooth()
```

Podemos añadir geoms adicionales al gráfico actual con `+`.

Los Geoms aparecen en el orden en que se escriben: la línea suavizada aparece sobre los puntos.



`geom_smooth()` proporciona medias condicionales suavizadas, lo que ayuda a mostrar tendencias en presencia de "sobretrazado" (véase documentación)

Una nota rápida sobre las sangrías

Las sangrías, los espacios y las nuevas líneas no afectan a la ejecución del código y pueden variarse para mejorar la legibilidad.

```
ggplot(data = surv, mapping = aes(x = age_years, y = ht_cm))+geom_point()
```

es lo mismo que:

```
ggplot(data = surv,
       mapping = aes(x = age_yers, y = ht_cm)) +
geom_point()
```

es lo mismo que:

```
ggplot(
  data = surv,          # caso de uso lista de líneas
  mapping = aes(        # hacer mapeos estéticos para todos los geoms
    x = age_years,      # asignar el eje x a la columna de edad
    y = ht_cm)) +       # asignar el eje y a la altura
geom_point()            # mostrar los datos en forma de puntos
```

Otra estética

Además de los ejes, otras "estéticas" comunes son:

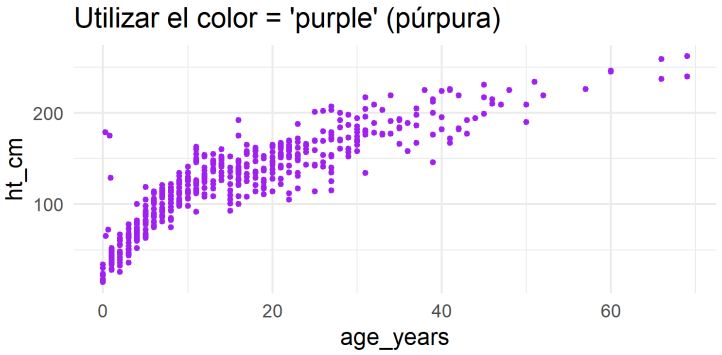
Argumento	Controla
`shape`	Visualización del punto como punto, estrella, triángulo, cuadrado...
`fill`	El color *interior* (por ejemplo, de barra o boxplot)
`color`	El color *exterior* o de barra, boxplot - O de punto
`size`	Grosor de la línea, tamaño del punto...
`alpha`	Transparencia: de 0 (invisible) a 1 (opaco)
`width`	Anchura de las barras del "bar plot"
`linetype`	Ya sea sólido, punteado, punteado, etc.
`binwidth`	Anchura de las franjas del histograma

asignaciones de parametros de estética

La estética puede asignarse a cualquiera de los dos:

- **Valores estáticos**: `color = "púrpura"`
  - Asignado *fuera* del conjunto `aes()`
  - La misma visualización para todos los datos

Algunos ejemplos:



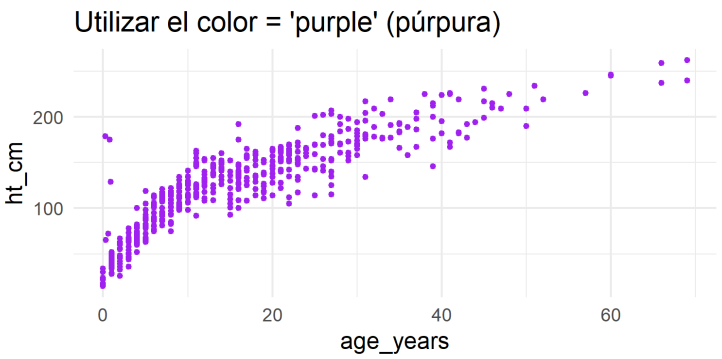
asignaciones de parametros de estética

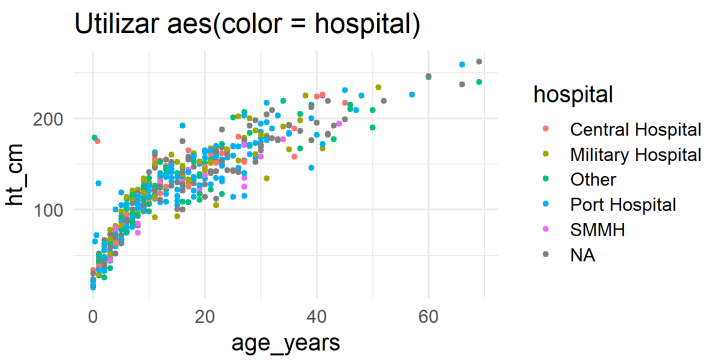
La estética puede asignarse a cualquiera de los dos:

- **Valores estáticos**: `color = "púrpura"`
  - Asignado *fuera* del conjunto `aes()`
  - La misma visualización para todos los datos

- **Una columna de datos**: `aes(color = hospital)`
  - Asignado *dentro* de `aes()`
  - Muestra los datos como "grupos"

Algunos ejemplos:





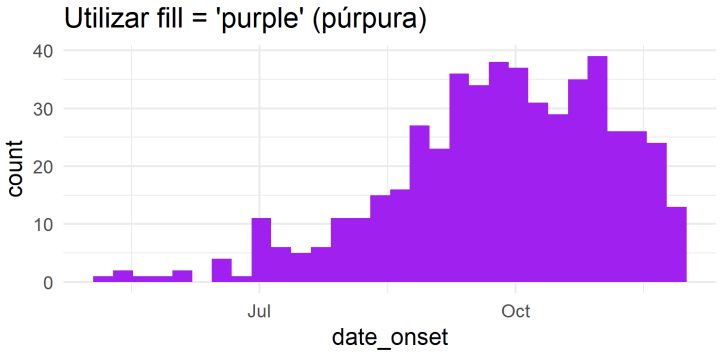
asignaciones de parametros de estética

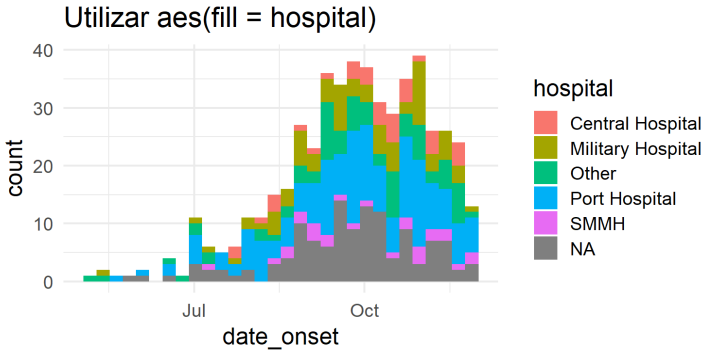
La estética puede asignarse a cualquiera de los dos:

- **Valores estáticos**: `color = "púrpura"`
  - Asignado *fuera* del conjunto `aes()`
  - La misma visualización para todos los datos

- **Una columna de datos**: `aes(color = hospital)`
  - Asignado *dentro* de `aes()`
  - Muestra los datos como "grupos"

Más ejemplos:



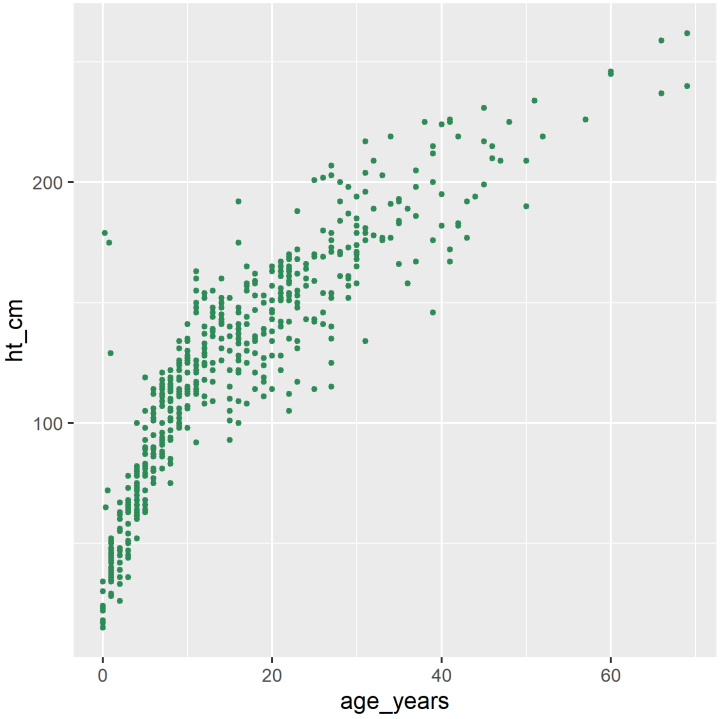


Estética estática

```
ggplot(
  data = surv,
  mapping = aes(
    x = age_years,
    y = ht_cm)) +
geom_point(color = "seagreen")
```

Una estética es *estática* si se aplica la misma visualización a todos los puntos de datos de la geom o del gráfico. Las estéticas estáticas se definen *fuera* de `aes()` con un *valor numérico o de caracteres*.

Otros ejemplos: `size = 3`
`alpha = 0.5`
`width = 1.2`



Estética dinámica

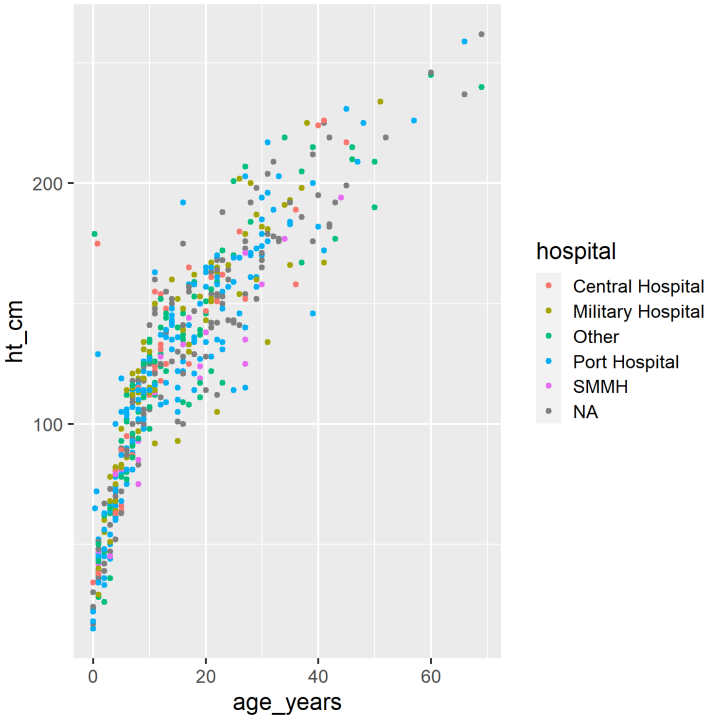
```
ggplot(
  data = surv,
  mapping = aes(
    x = age_years,
    y = ht_cm,
    color = hospital)) +
geom_point()
```

La estética *Dinámica* se asigna a un nombre de columna del data frame.

Esto crea "grupos" y genera una leyenda.

Esta asignación ocurre *dentro* de `aes()`.

Arriba, `color=` se asigna a la columna `hospital`.

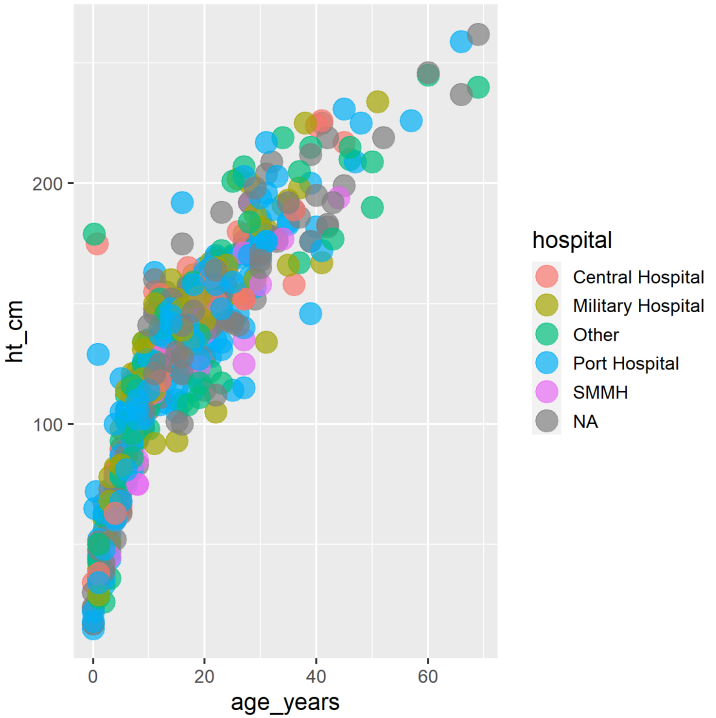


Estática y dinámica

```
ggplot(
  data = surv,
  mapping = aes(
    x = age_years,
    y = ht_cm,
    color = hospital)) +
geom_point(
  size = 7,
  alpha = 0.7)
```

Arriba, `size = 7` y `alpha = 0.7` se asignan estáticamente, fuera de `aes()`.

Se asigna `color=` a la columna `hospital`, dentro de `aes()`.



Más información sobre la estética de ggplot aquí

Facetas (mini gráficos)

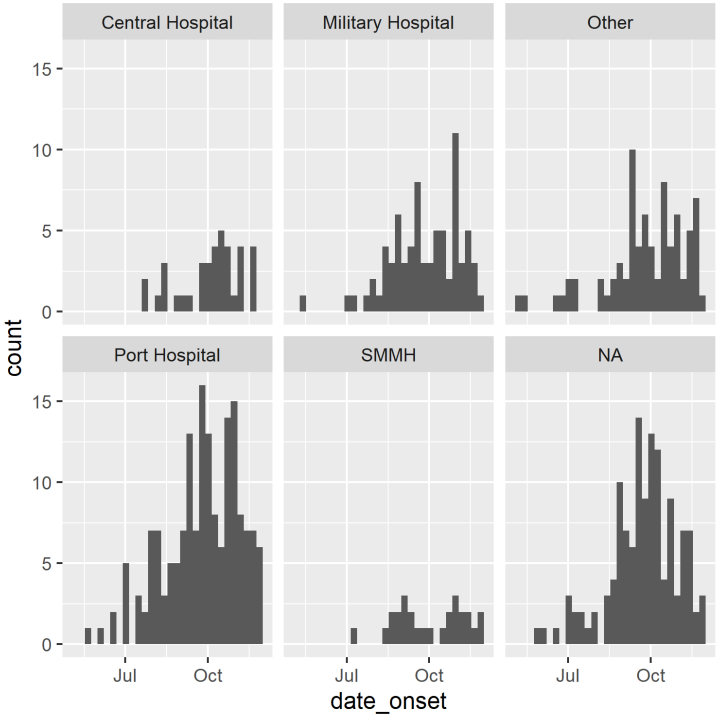
```
ggplot(
  data = surv,
  mapping = aes(x = date_onset)) +
geom_histogram() +
facet_wrap(~hospital)
```

Las facetas son "mini-gráficos": Gráficos para un valor único

Se pueden utilizar para mostrar grupos en los datos.

`facet_wrap()` produce una faceta por valor único de la columna.

Coloque un "~" antes del nombre de la columna.

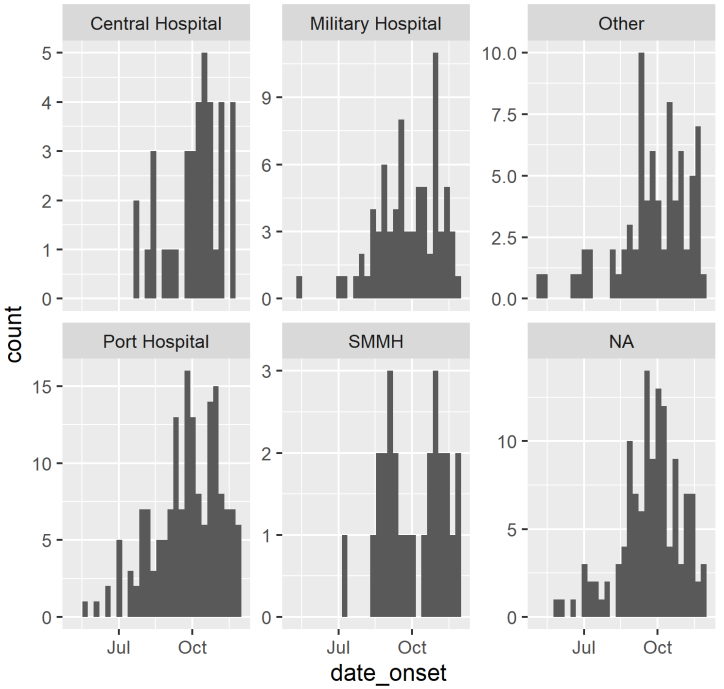


Facetas (mini gráficos)

```
ggplot(
  data = surv,
  mapping = aes(x = date_onset)) +
geom_histogram() +
facet_wrap(~hospital,
           scales = "free_y")
```

Ejes autoescalables "libres" con `scales=`

- "free_y"
- "free_x"
- "free" (both x and y)



Obsérvese el eje y libre

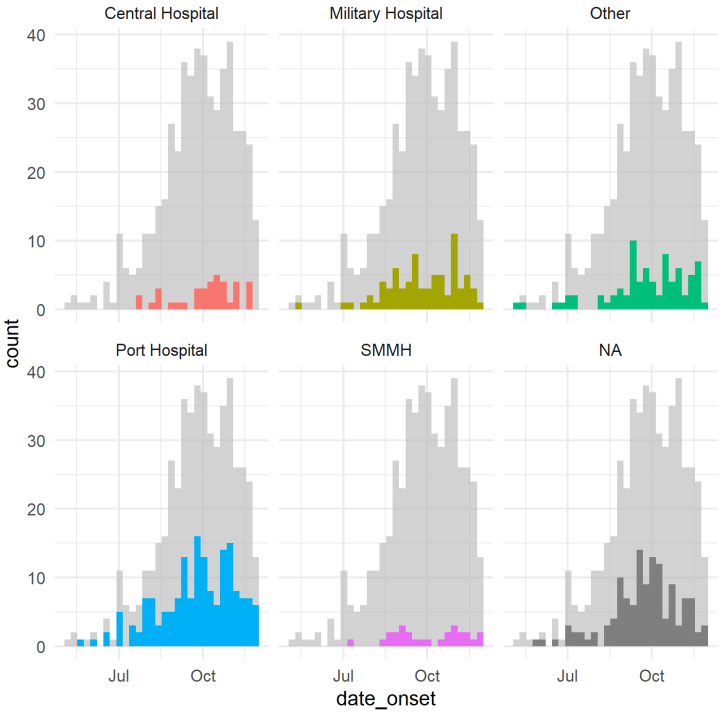
¡Alerte a su público si utiliza ejes libres! Prueba también con `ncol=` y `nrow=`

Facetas + `gghighlight()`

Los paquetes de extensión de ggplot como {gghighlight} son útiles.

`gghighlight()` puede mostrar una "sombra" detrás de cada faceta.

```
ggplot(
  data = surv,
  mapping = aes(
    x = date_onset,
    fill = hospital)) +
geom_histogram() +
facet_wrap(~ hospital) +
gghighlight()
```

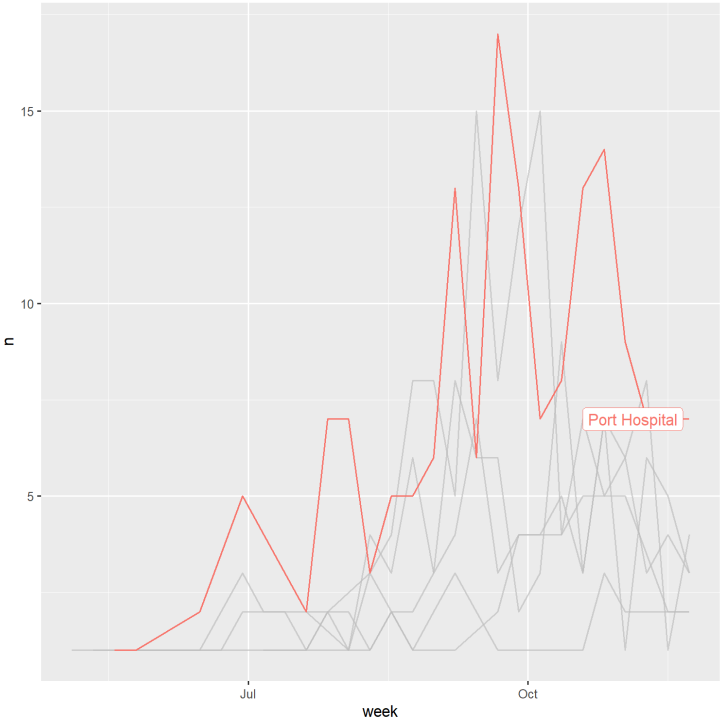


gghighlight

Añandiendo `gghighlight()` a otros gráficos, para especificar los valores a ser destacados

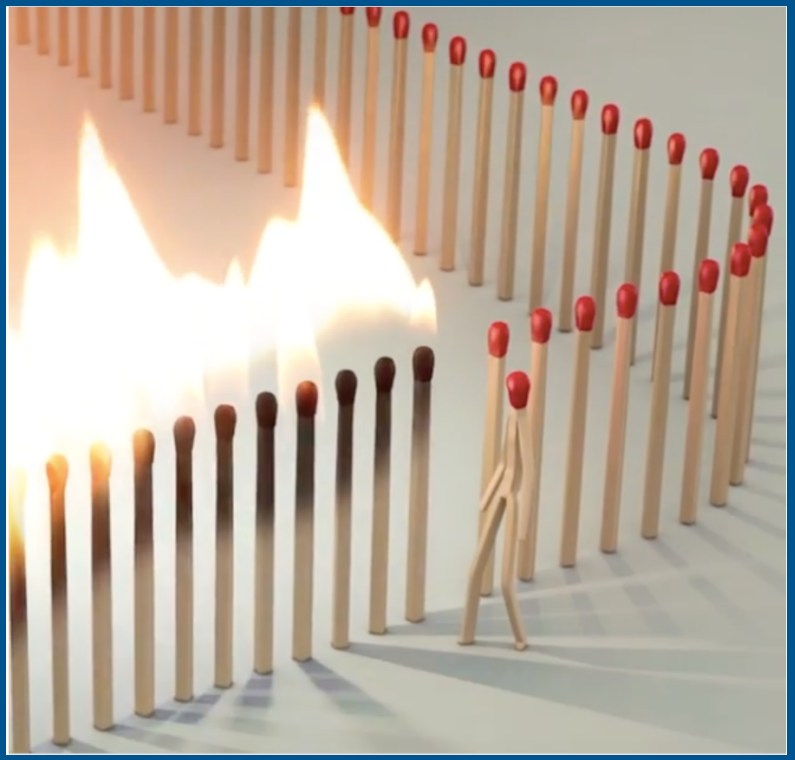
```
surv %>%
  # Obtenga los recuentos semanales por hos
  count(
     hospital,
     week = floor_date(date_onset, "week"))

  # parcela
  ggplot(
    mapping = aes(
      x = week,
      y = n,
      color = hospital)) +
  geom_line() +
  gghighlight(
    hospital == "Port Hospital")+
  theme(legend.position = "none")
```



Agradecimientos

imágenes ggplot de:

- http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html
- https://jcheshire.com/r-spatial-data-hints/great-maps-ggplot2/

Sesiones de trabajo