EQUIPO 7

PRODUCTO INTEGRADOR DE APRENDIZAJE

GGPLOT

Manuel Mancha Sandoval 1877313
José Roberto Landero Pérez 1990004
Jorge Roberto Flores Zamora 1900477
Javier Eduardo Ramírez Vega 1922597
Alexandra Guadalupe Valdez Mireles 1910361

CREANDO EL OBJETO GGPLOT

CÓDIGO

```
library(dslabs)
data(murders)

murders %>%
ggplot()
```

Se crea el objeto ggplot a partir de la data murders utilizando el operador pipeline %>%. También se debe cargar la data murders de la librería dslabs

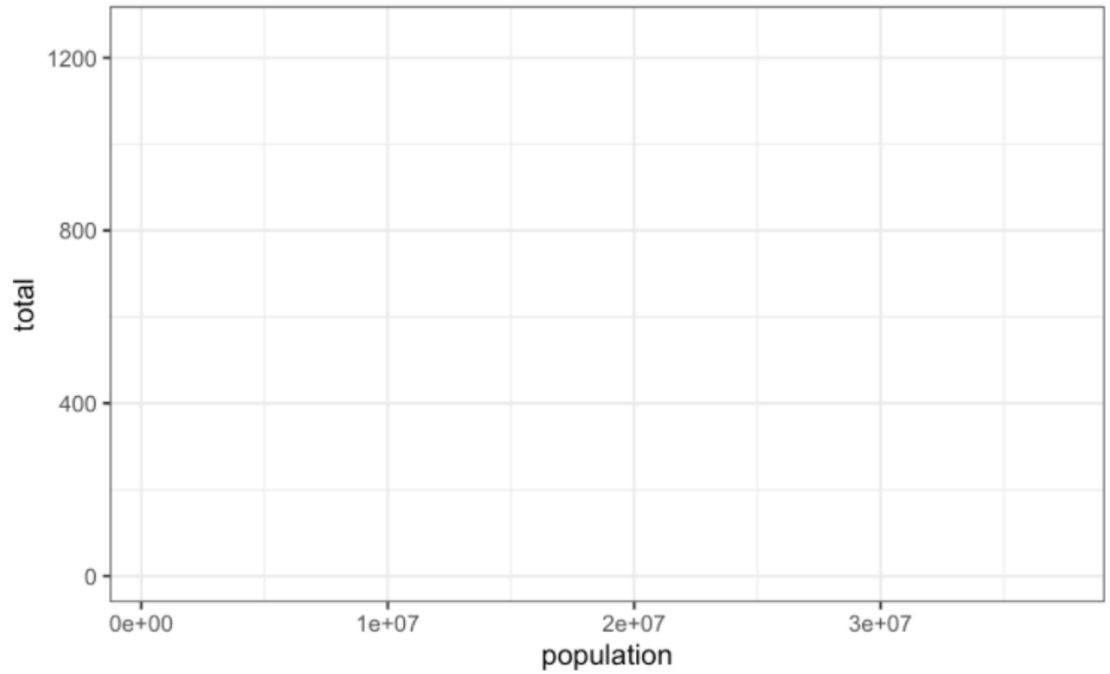
Este código solo nos muestra un cuadro vacío. Esto es porque no le hemos especificado qué variables tomar del data frame ni tampoco qué tipo de gráfico queremos.
Para agregar cada componente del gráfico que estamos creando usaremos capas. El objeto ggplot nos permite ir agregando capa por capa qué componente del gráfico queremos agregar. El símbolo para agregar capas al objeto ggplot es el símbolo +.

CAPA AESTHETIC MAPPING

La estética básica es: qué va en el **eje x** y qué ponemos en el **eje y**. Usaremos la función aesthetic (estética en inglés) que en R es aes().
No tenemos que usar el accesador $ porque la función aes toma como referencia la tabla murders antes del pipeline.

CÓDIGO

```
murders %>%
 ggplot() +
 aes(x = population, y = total) +
```

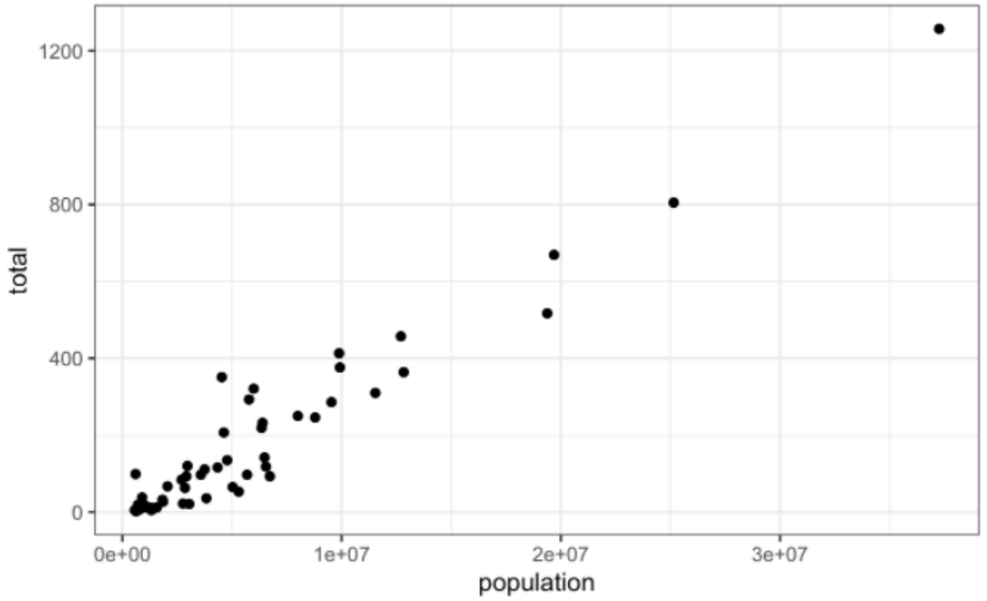


CAPA GEOMS

Agreguemos una capa más que nos indique qué tipo de gráfico queremos. Para ello usaremos los llamados **geoms**. Existen diferentes tipo de geoms. Por ejemplo, un gráfico de dispersión es mostrado con puntos, por ende usaremos la función geom_point().

CÓDIGO

```
murders %>%
  ggplot() +
  aes(x = population, y = total) +
  geom_point()
```

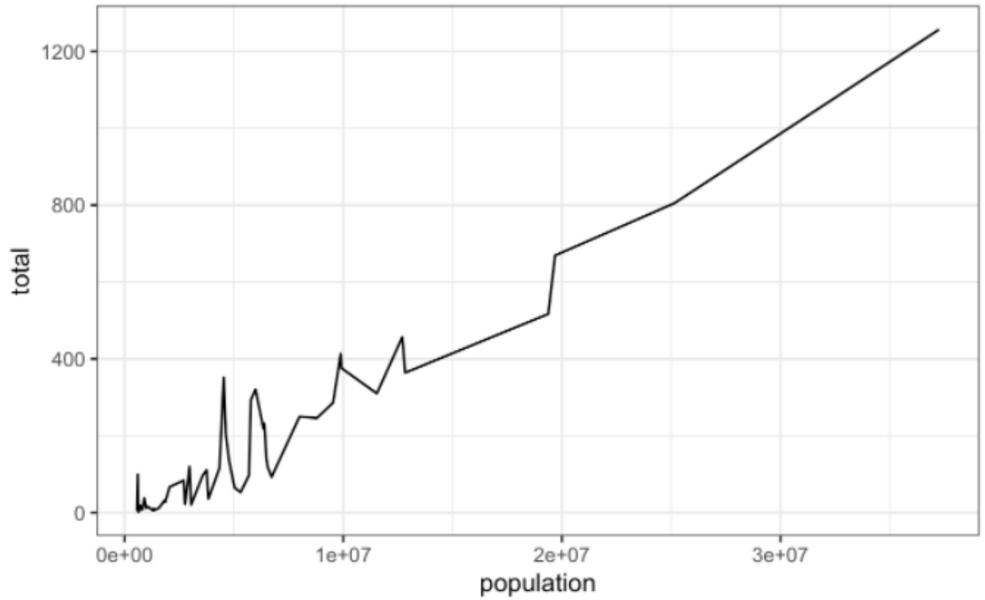


CAPA GEOMS

De la misma forma, podemos mostrar líneas que conecten los datos en vez de puntos con la función geom_line().

CÓDIGO

```
murders %>%
ggplot() +
aes(x = population, y = total) +
geom_line()
```



murders %>%
ggplot() +
aes(x = population, y = total) +
geom_line()

CAPA DE ESCALA

Visualmente aun podemos mejorar más nuestro gráfico. Vemos varios datos concentrados en valores menores y solo algunos extremos. En esos casos es mejor tener una vista escalando los ejes usando logaritmos. Para ello, usaremos las capas scale_x_continuous() y scale_y_continuous(). Por ejemplo, si queremos transformar la escala a logaritmo en base 2 tendríamos que agregar capas, pero también cambiar el valor de nudge_x, por el cambio de escala:

CÓDIGO

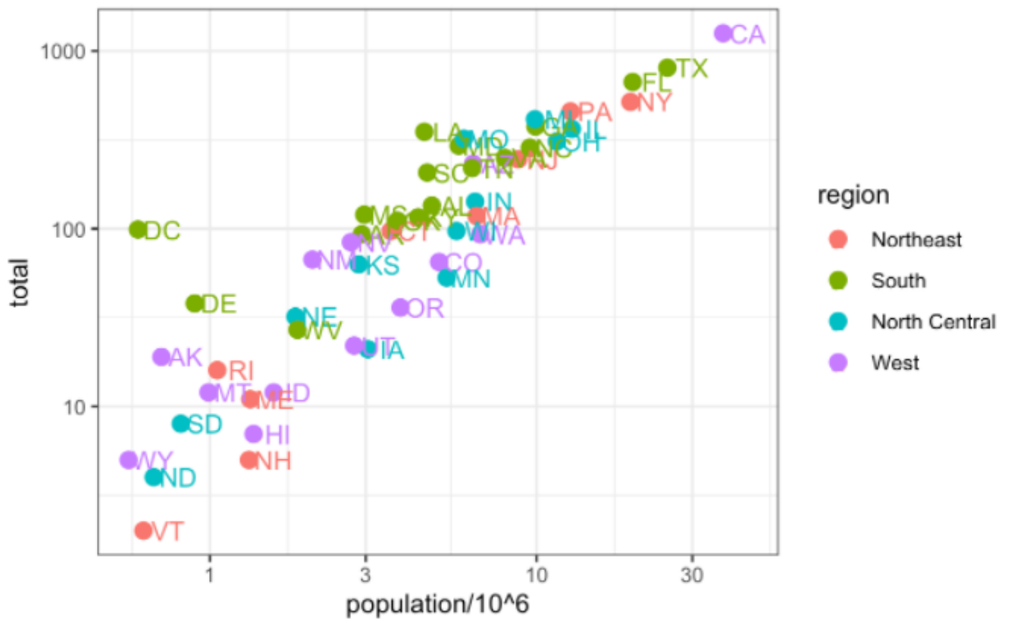
```
murders %>%
ggplot() +
aes(x = population/10^6, y = total,
label=abb, color=region) +
 geom_point(size=3) +
 geom_text(nudge_x = 0.23) +
 scale_x_continuous(trans = "log2")
+  scale_y_continuous(trans =
"log2")
```

CAPA DE ESCALA

De la misma forma, podríamos hacer la transformación a logaritmo en base 10:

CÓDIGO

```
murders %>%
ggplot() +
aes(x = population/10^6, y = total, label=abb,
color=region) +
geom_point(size=3) +
 geom_text(nudge_x = 0.075) +
 scale_x_continuous(trans = "log10") +
 scale_y_continuous(trans = "log10")
```

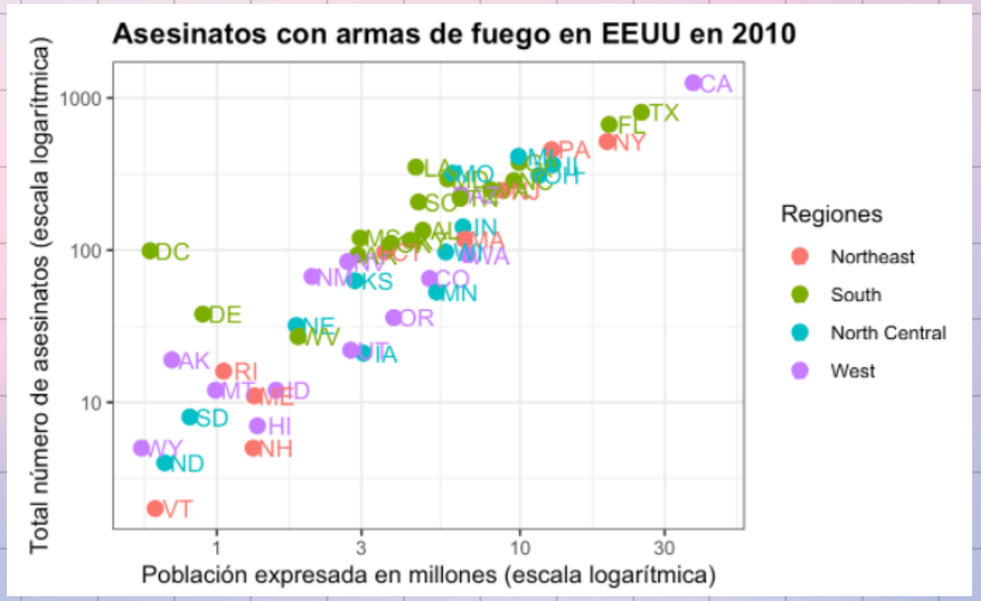


CAPA DE ETIQUETAS, TÍTULO Y LEYENDAS

También podemos cambiar las etiquetas (label en inglés) al gráfico. Hasta el momento en el eje x vemos que aparece **population/10^6** y lo podemos cambiar con la función xlab(). De la misma forma podemos cambiar en el eje y usando ylab(). Para agregar un títutlo al gráfico usaremos la función ggtitle(). Para cambiar el nombre a la leyenda usaremos la función scale_color_discrete().

CÓDIGO

```
murders %>%
ggplot() +
aes(x = population/10^6, y = total, label=abb,
color=region) +
geom_point(size=3) +
 geom_text(nudge_x = 0.075) +  scale_x_log10()
+
 scale_y_log10() +
 xlab("Población expresada en millones (escala
logarítmica)") +
 ylab("Total número de asesinatos (escala
logarítmica)") +
ggtitle("Asesinatos con armas de fuego en
EEUU en 2010") +
scale_color_discrete(name = "Regiones")
```



LÍNEAS DE REFERENCIA

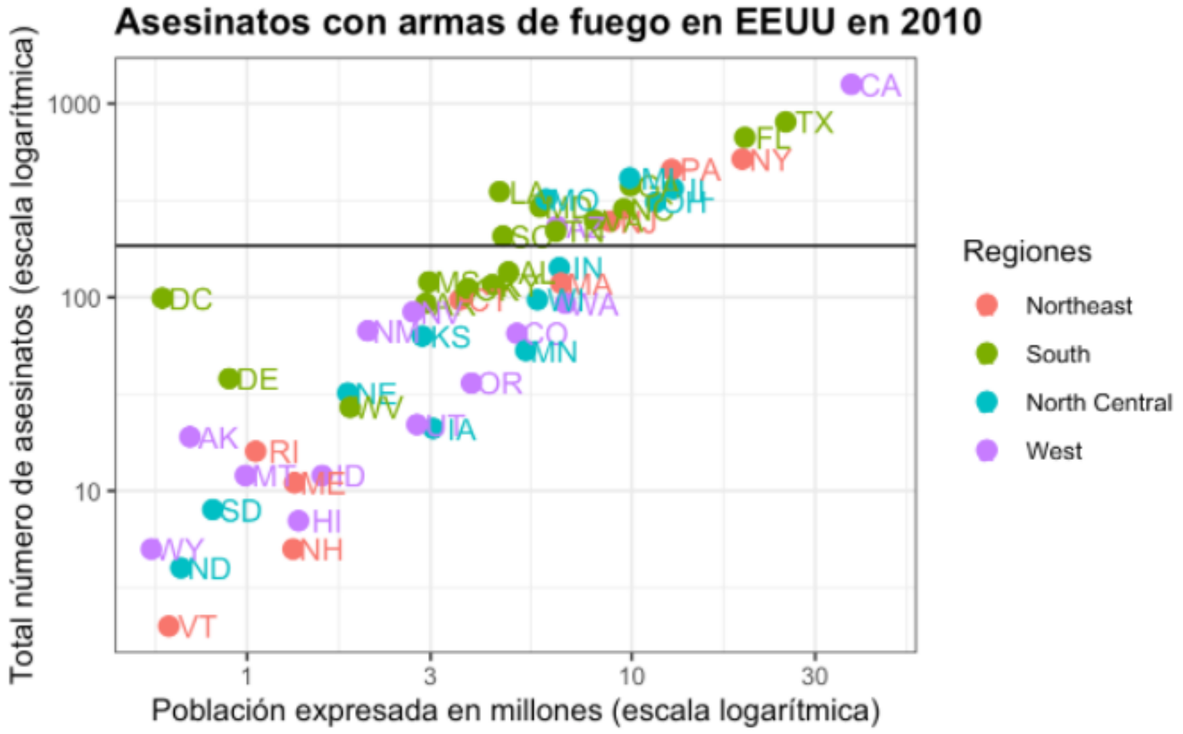
Podemos agregar líneas de referencia, ya sean verticales con geom_vline(xintercept =), horizontales con geom_hline(yintercept = ...) o diagonales con geom_abline(intercept =), ésta última nos pide en qué punto corta el eje y y dibuja una línea con pendiente por default de 1. Por ejemplo, podríamos calcular el promedio del total de asesinatos y dibujar una línea de referencia horizontal.

CÓDIGO

```
#Calculamos el promedio del total
promedio_total <- mean(murders$total)
#Y agregamos la línea de referencia horizontal
murders %>%
ggplot() +
aes(x = population/10^6, y = total, label=abb,
color=region) +
geom_point(size=3) +
geom_text(nudge_x = 0.075) +
scale_x_log10() +
scale_y_log10() +
xlab("Población expresada en millones (escala
logarítmica)") +
ylab("Total número de asesinatos (escala
logarítmica)") +
ggtitle("Asesinatos con armas de fuego en EEUU
en 2010") +
scale_color_discrete(name = "Regiones") +
geom_hline(yintercept = promedio_total)
```

GGPLOT

GRÁFICO



CAMBIANDO ESTILO DEL GRÁFICO

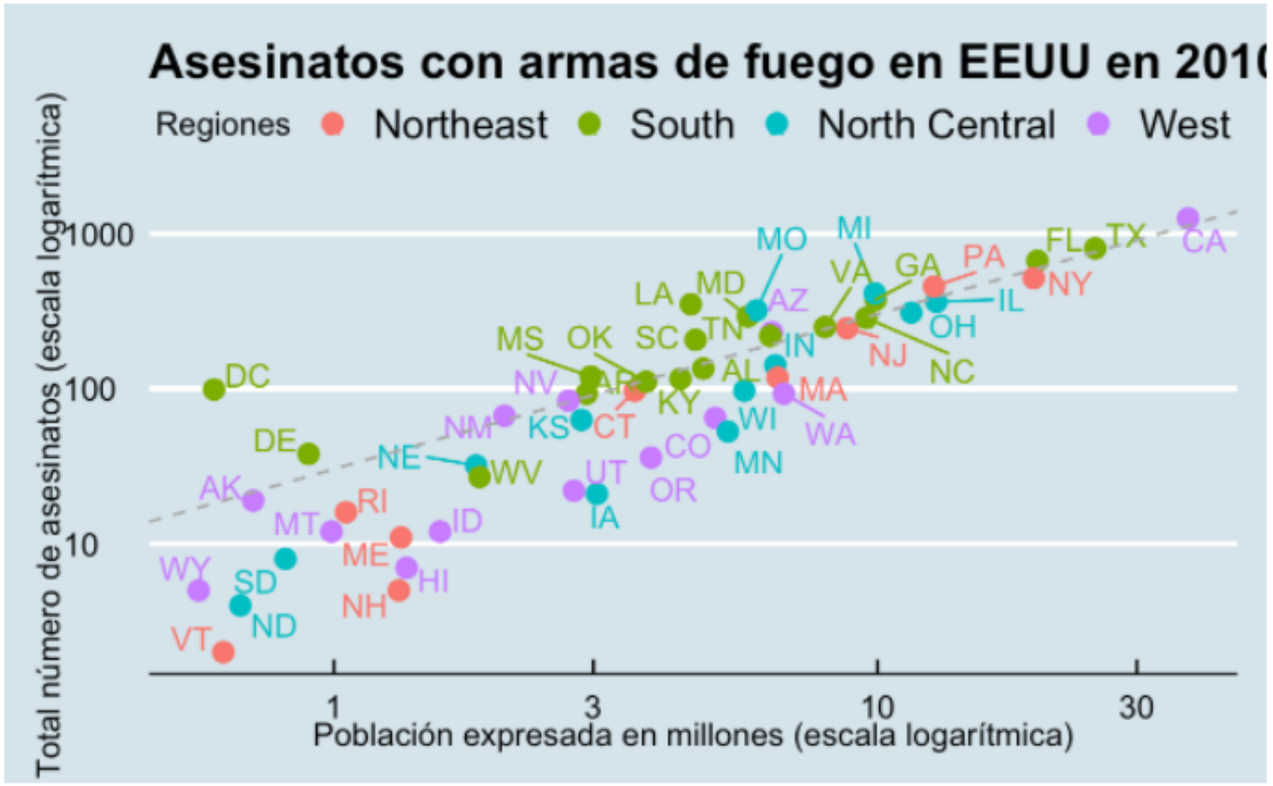
El estilo del gráfico usando ggplot() puede ser fácilmente cambiado. Existen múltiples temas que podemos usar cargando la librería **ggthemes**. Podemos, por ejemplo utilizar un tema muy utilizado: el tema de the economist agregando la capa theme_economist(). Podemos hacer que los nombres se repelan utilizando la función geom_text_repel() en vez de geom_text() que estamos usando actualmente. Para utilizar esta función necesitamos llamar a la librería ggrepel.

CÓDIGO

```
library(ggthemes)
library(ggrepel)
murders %>%
ggplot() +
aes(x = population/10^6, y = total, label=abb, color=region) +
 geom_point(size=3) +
geom_text_repel() +
scale_x_log10() +
 scale_y_log10() +
xlab("Población expresada en millones (escala
logarítmica)") +
ylab("Total número de asesinatos (escala logarítmica)") +
 ggtitle("Asesinatos con armas de fuego en EEUU en
2010") +
scale_color_discrete(name = "Regiones") +
 geom_abline(intercept = ratio_log10, lty = 2, color =
"darkgrey") +
theme_economist()
```

GGPLOT

GRÁFICO



¡Gracias!

Seguimos con un breve ejemplo de ggplot en R