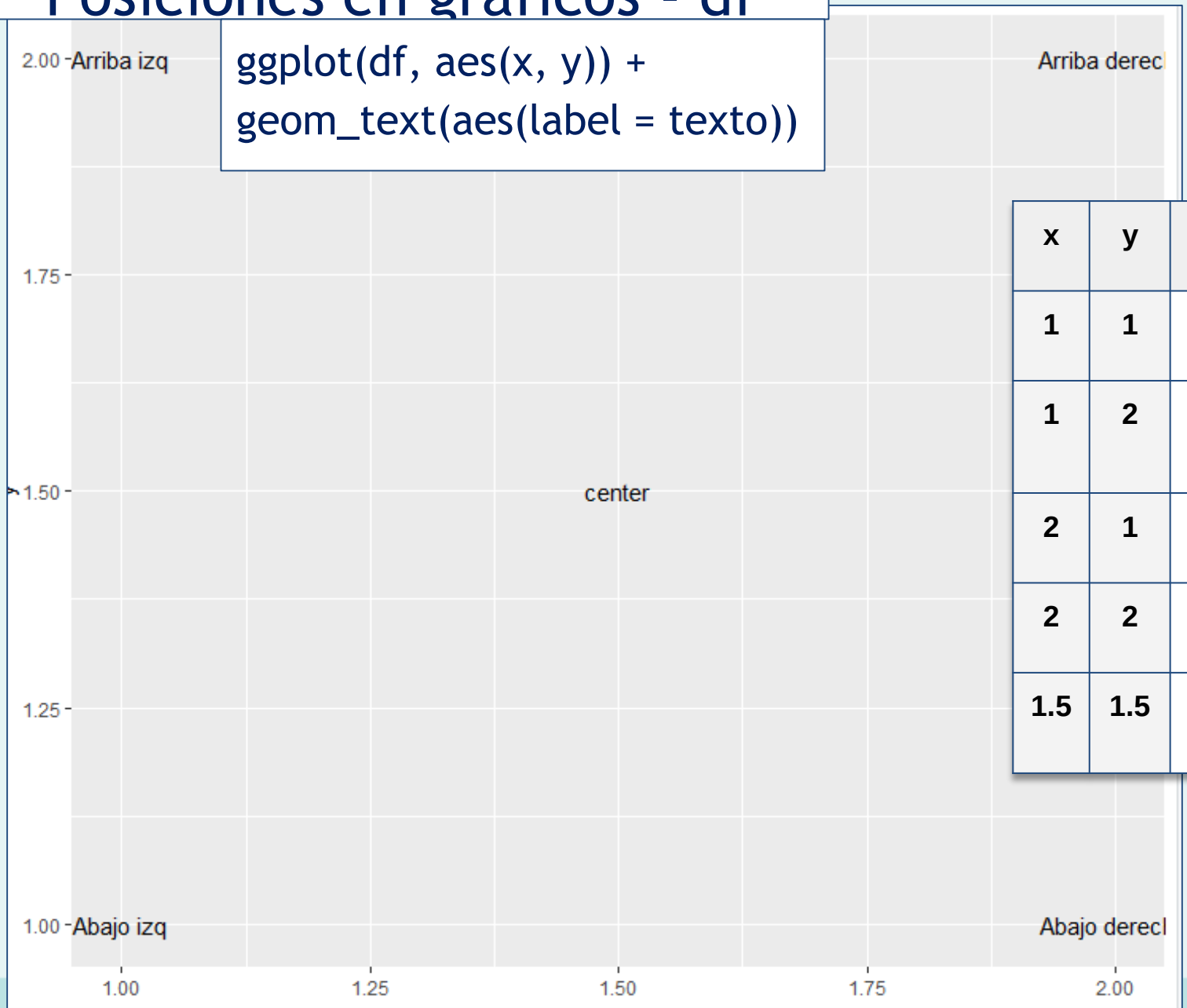
Sección 4

4.3 Principios de visualización de datos

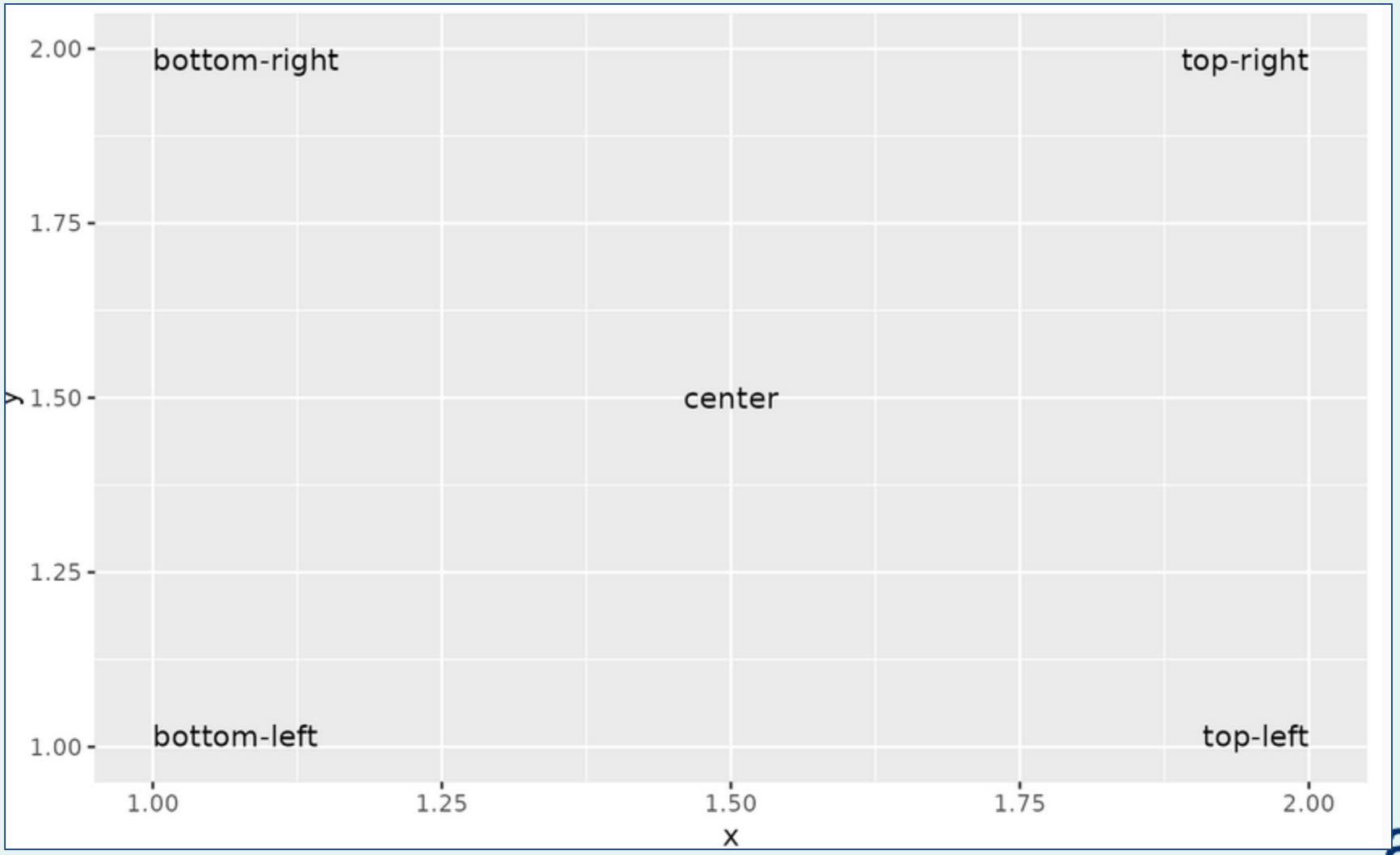
Justificación de datos
Posiciones en gráficos - df

```
ggplot(df, aes(x, y)) +
geom_text(aes(label = texto))
```



x	y	texto
1	**1**	Abajo izq
1	**2**	Arriba izq
2	**1**	Abajo derecha
2	**2**	Arriba derecha
1.5	**1.5**	Centro

```
ggplot(df, aes(x, y)) + geom_text(aes(label = text),
vjust = "inward", hjust = "inward"
```



Gráficos de rectas

- No hay una geometría específica para este tipo de gráficos
- Podemos utilizar la función geom_line() para crear rectas
- Este gráfico sirve para informar diferencias al comparar variables del mismo tipo pero para un número relativamente pequeño de comparaciones
- geom_line(): Conecta los puntos con una recta en el orden en que aparecen en el eje x.

Comparamos expectativa de vida entre 2010 y 2015 - Gráfico de rectas

```
library(tidyverse)
library(dslabs)
data(gapminder)

west <- c("Western Europe", "Northern Europe", "Southern Europe", "Northern America", "Australia and New Zealand")

dat <- gapminder %>% filter(year %in% c(2010, 2015) & region %in% west & !is.na(life_expectancy) & population > 10^7)

dat %>% mutate(location = ifelse(year == 2010, 1, 2),
  location = ifelse(year == 2015 & country %in% c("United Kingdom", "Portugal"),
  location + 0.22, location), hjust = ifelse(year == 2010, 1, 0)) %>%
  mutate(year = as.factor(year)) %>%
  ggplot(aes(year, life_expectancy, group = country)) +
  geom_line(aes(color = country), show.legend = FALSE) +
  geom_text(aes(x = location, label = country, hjust = hjust), show.legend = FALSE) +
  xlab("") + ylab("Life Expectancy")
```

Comparamos expectativa de vida entre 2010 y 2015 - Gráfico de rectas

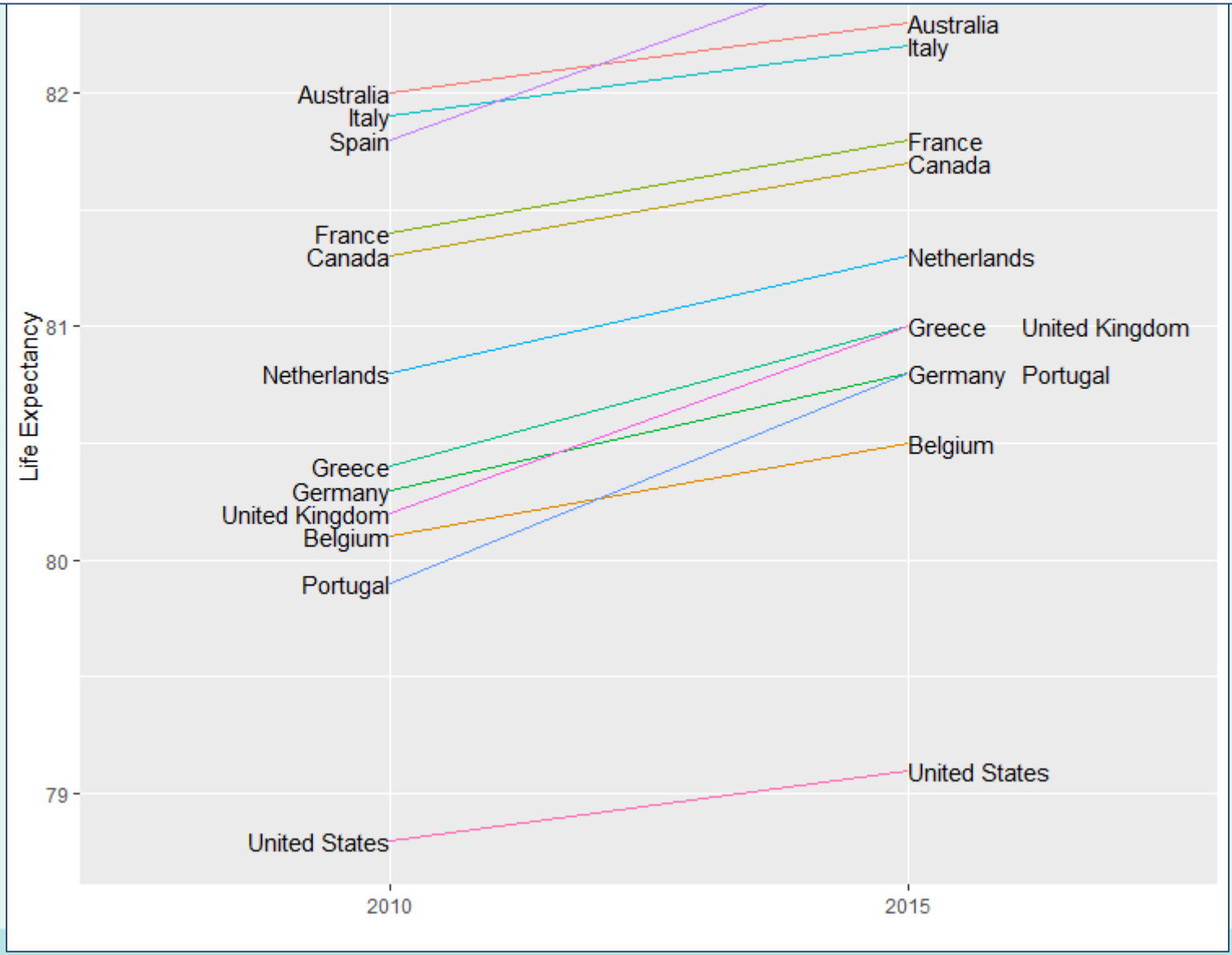


Gráfico Bland Altman - De diferencias - MA
Visualización de diferencias

- Este gráfico muestra en el eje y las diferencias entre los datos vs los promedios entre los datos en el eje x
- Para el mismo utilizamos la librería library(ggrepel) con la función geom_text_repel para agregar texto al gráfico sin superponerse entre sí

  Este gráfico es más apropiado que el gráfico de rectas para comparar datos para un número largo de observaciones

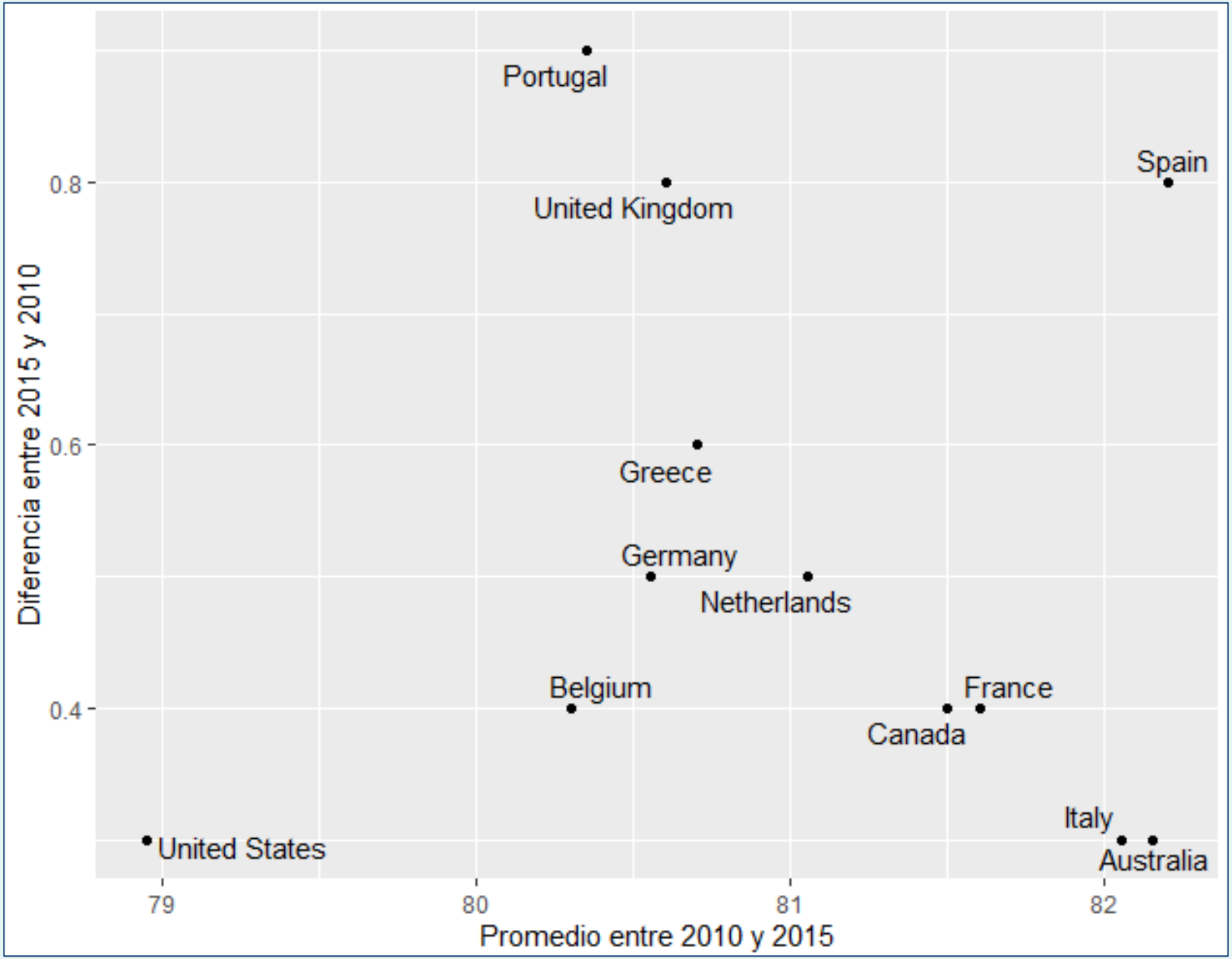
```
library(ggrepel)
```

1	life_expectancy_1960
2	life_expectancy_1960

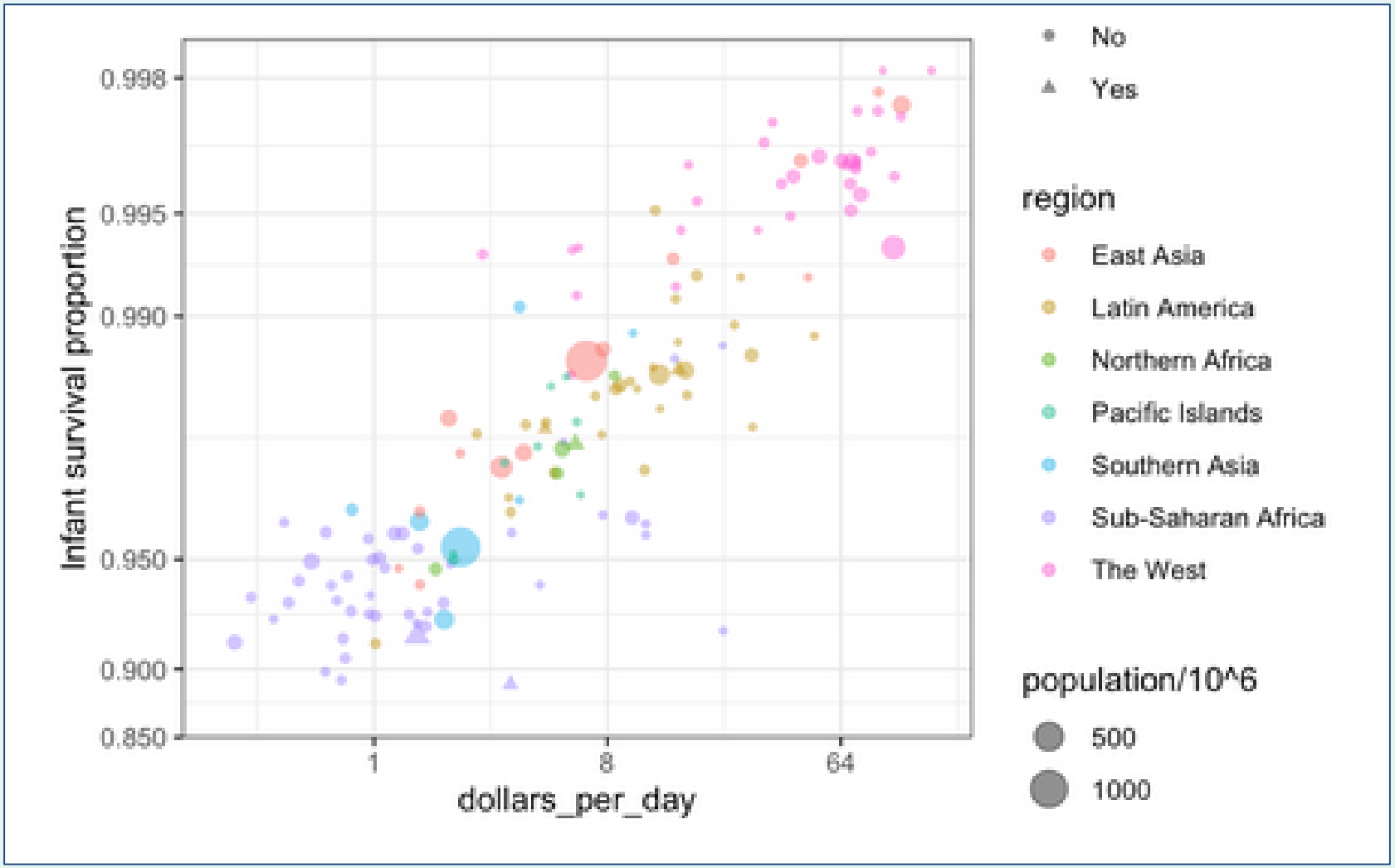
```
dat %>%
  mutate(year = paste0("life_expectancy_", year)) %>%
        select(country, year, life_expectancy) %>%
```

	country	life_expectancy_2010	life_expectancy_2015
1	Australia	82.0	82.3
2	Belgium	80.1	80.5

```
spread(year, life_expectancy) %>%
  mutate(average = (life_expectancy_2015 +
life_expectancy_2010)/2,
  difference = life_expectancy_2015 - life_expectancy_2010)
%>%
  ggplot(aes(average, difference, label = country)) +
  geom_point() +  geom_text_repel() +
  xlab("Promedio entre 2010 y 2015") +
  ylab("Diferencia entre 2015 y 2010")
```

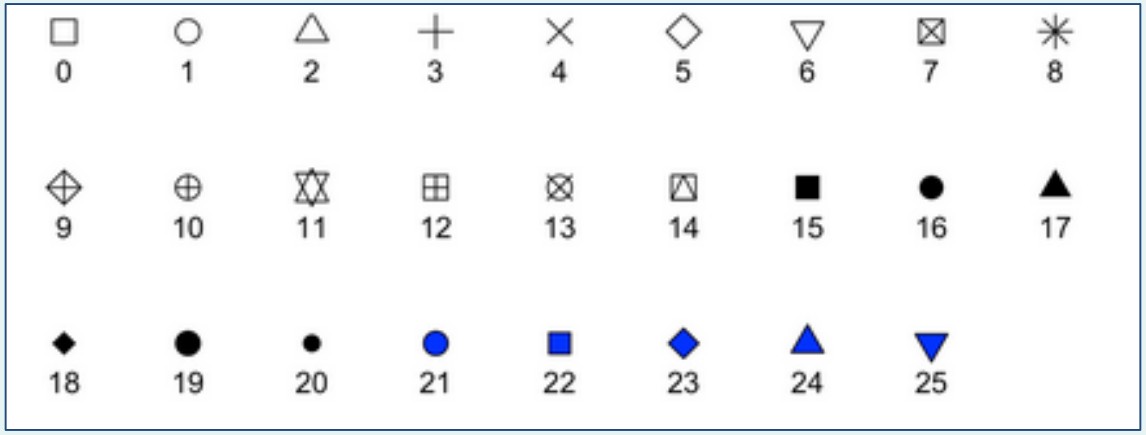

Portugal
Spain
United Kingdom
Greece
Germany
Netherlands
Belgium
France
Canada
Italy
Australia
United States
0.8
0.6
0.4
Diferencia entre 2015 y 2010
79
80
81
82
Promedio entre 2010 y 2015

Comparar varias variables - Visualizar relación entre 3 variables



Codificar variables categóricas

- Las variables categóricas las podemos codificar por color y forma.
  - Las formas pueden ser controladas mediante el argumento **shape**
  - Las disponibles en R son las siguientes:

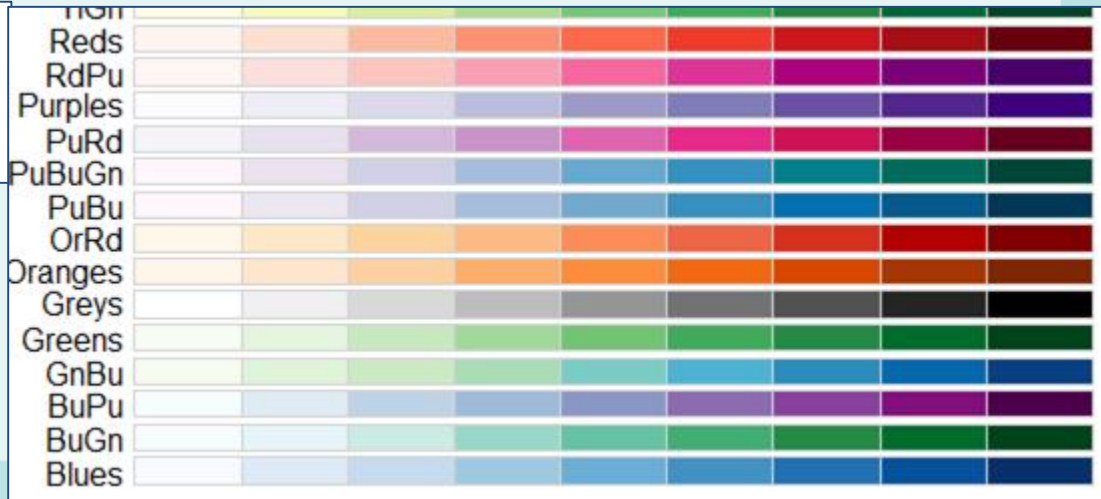


Notar que para las últimas 5 los colores van dentro de la forma

Codificar variables numéricas

- Para las variables continuas podemos utilizar color, intensidad o tamaño para codificar los datos
- Para cuantificar podemos elegir entre dos opciones, secuencial y divergente:
  - Secuencial: Ideal para datos que aumentan de un mínimo a un máximo. En estas escalas los valores altos son distinguidos claramente de los bajos. Ejemplos del paquete RColorBrewer:

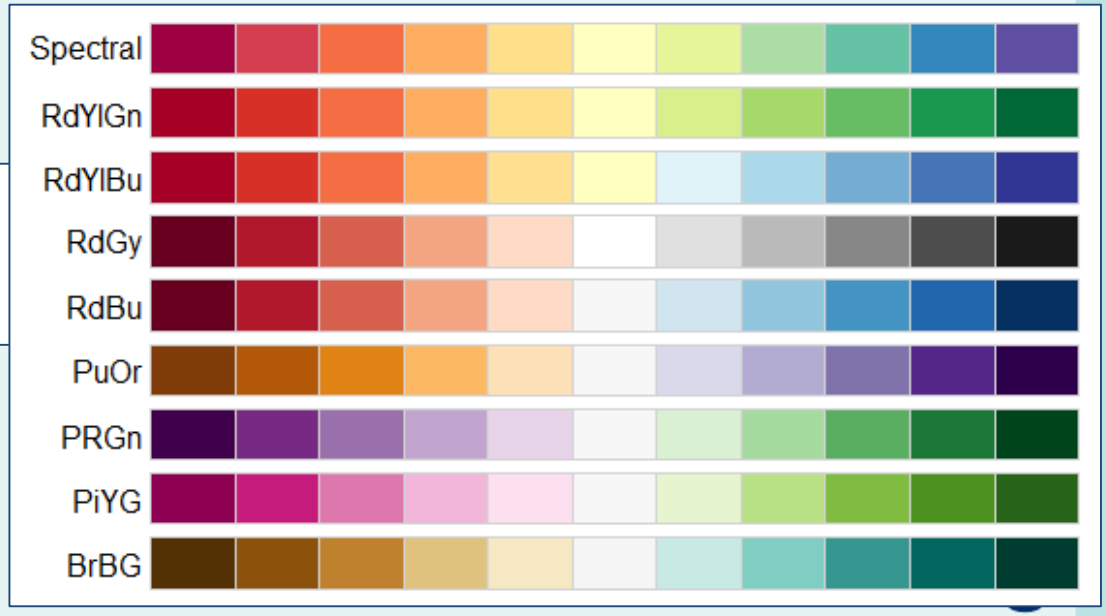
```
library(RColorBrewer)
display.brewer.all(type="seq")
)
```



Codificar variables numéricas - Divergente

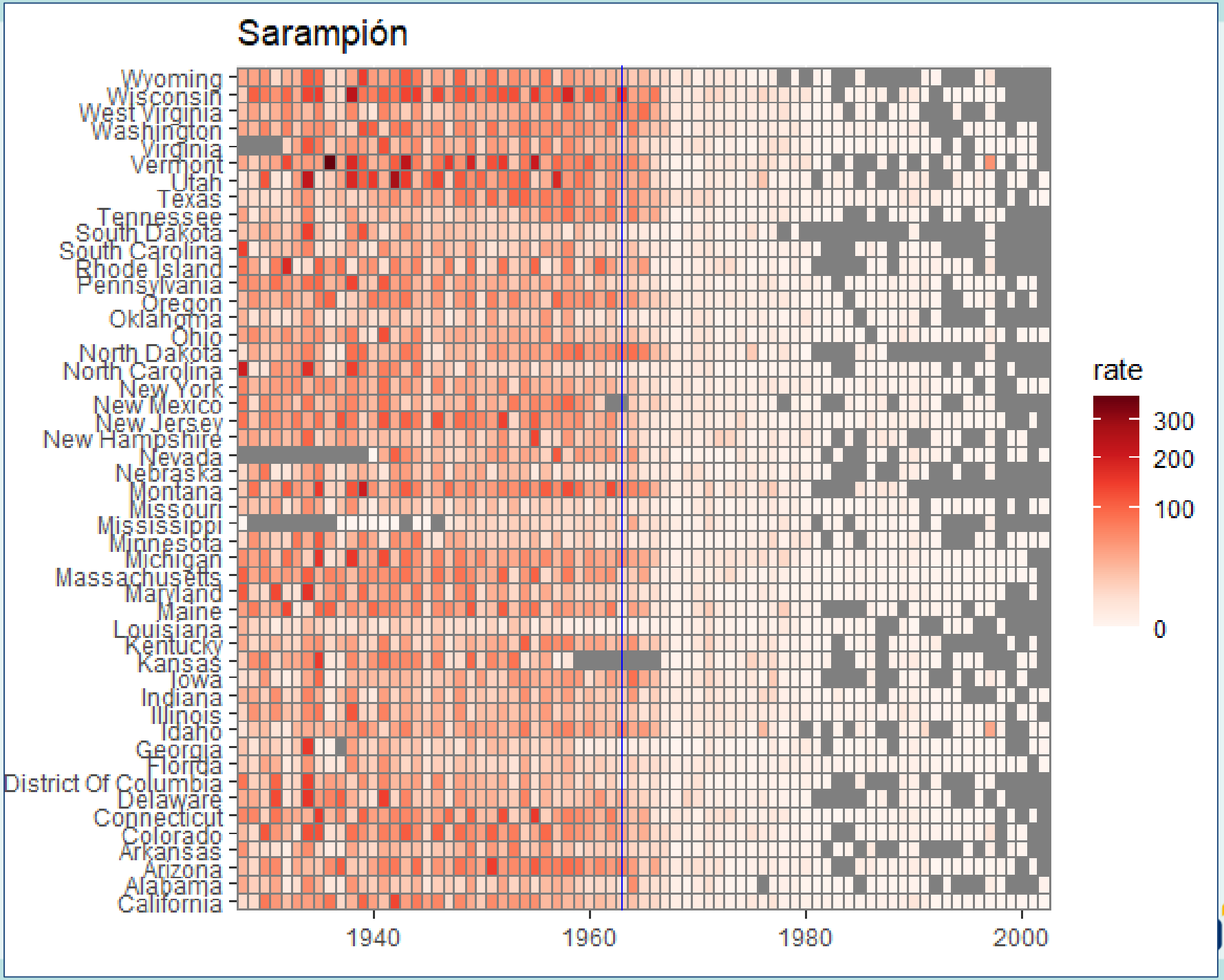
- Los colores divergentes son utilizados para representar valores que divergen del centro.
- Colocamos un emphasis igual entre los valores extremos. La comparación visual a realizar es entre los valores menores que el centro y los mayores al centro.

```
library(RColorBrewer)
display.brewer.all(type="div")
```



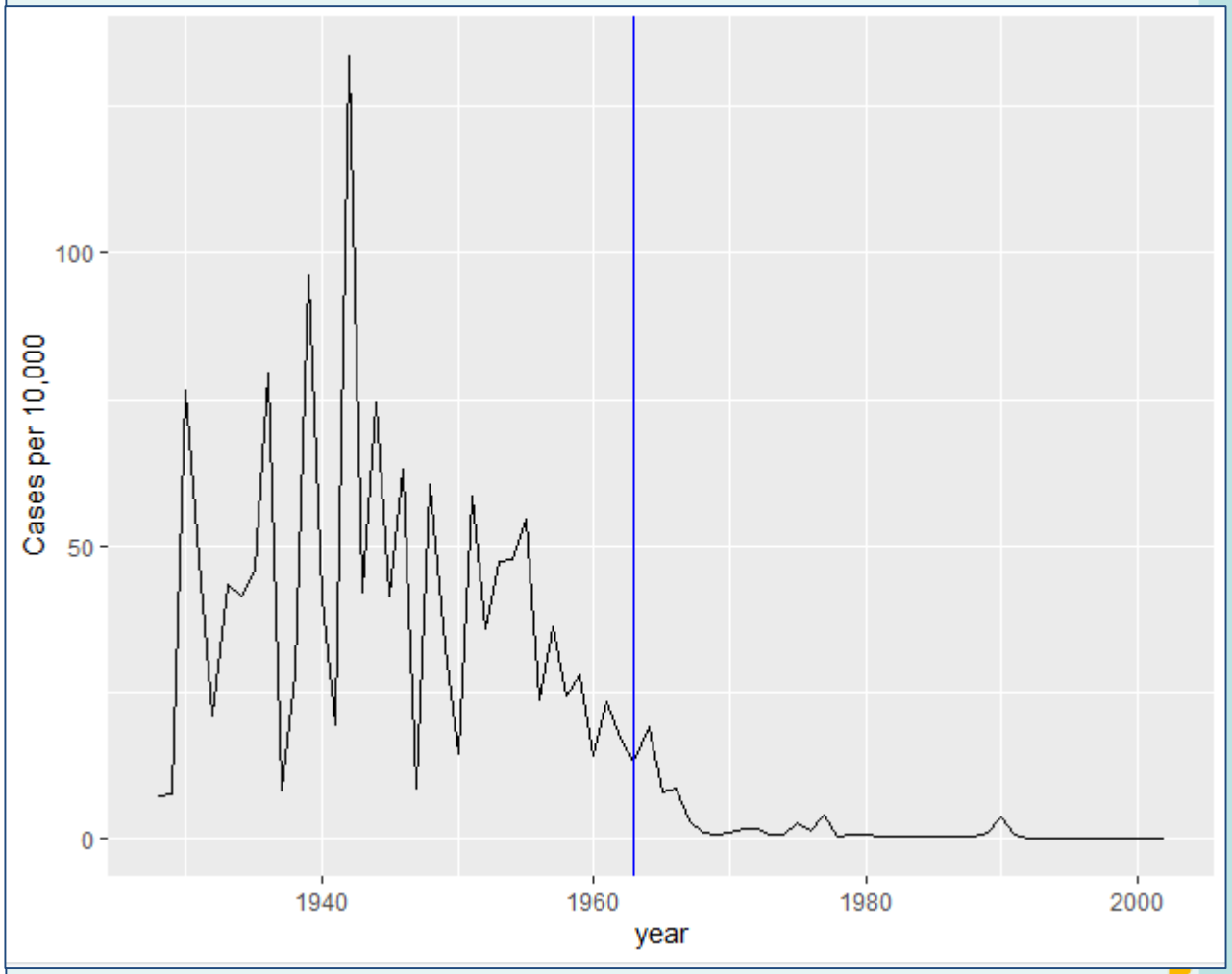
Ejemplo - Vacunación de sarampión

```
library(tidyverse)
library(dslabs)
data(us_contagious_diseases)
the_disease <- "Measles"
dat <- us_contagious_diseases %>%
   filter(!state %in% c("Hawaii", "Alaska") & disease == the_disease) %>%
   mutate(rate = count / population * 10000 * 52/weeks_reporting) %>%
   mutate(state = reorder(state, rate))

dat %>% ggplot(aes(year, state, fill=rate)) +
  geom_tile(color = "grey50") +
  scale_x_continuous(expand = c(0,0)) +
  scale_fill_gradientn(colors = RColorBrewer::brewer.pal(9, "Reds"), trans =
"sqrt") +
  geom_vline(xintercept = 1963, col = "blue") +
  ggtitle("Sarampión") +
  ylab("") +
  xlab("")
```


Sarampión
Wyoming
Wisconsin
West Virginia
Washington
Virginia
Vermont
Utah
Texas
Tennessee
South Dakota
South Carolina
Rhode Island
Pennsylvania
Oregon
Oklahoma
Ohio
North Dakota
North Carolina
New York
New Mexico
New Jersey
New Hampshire
Nevada
Nebraska
Montana
Missouri
Mississippi
Minnesota
Michigan
Massachusetts
Maryland
Maine
Louisiana
Kentucky
Kansas
Iowa
Indiana
Illinois
Idaho
Georgia
Florida
District Of Columbia
Delaware
Connecticut
Colorado
Arkansas
Arizona
Alabama
California
1940
1960
1980
2000
rate
300
200
100
0

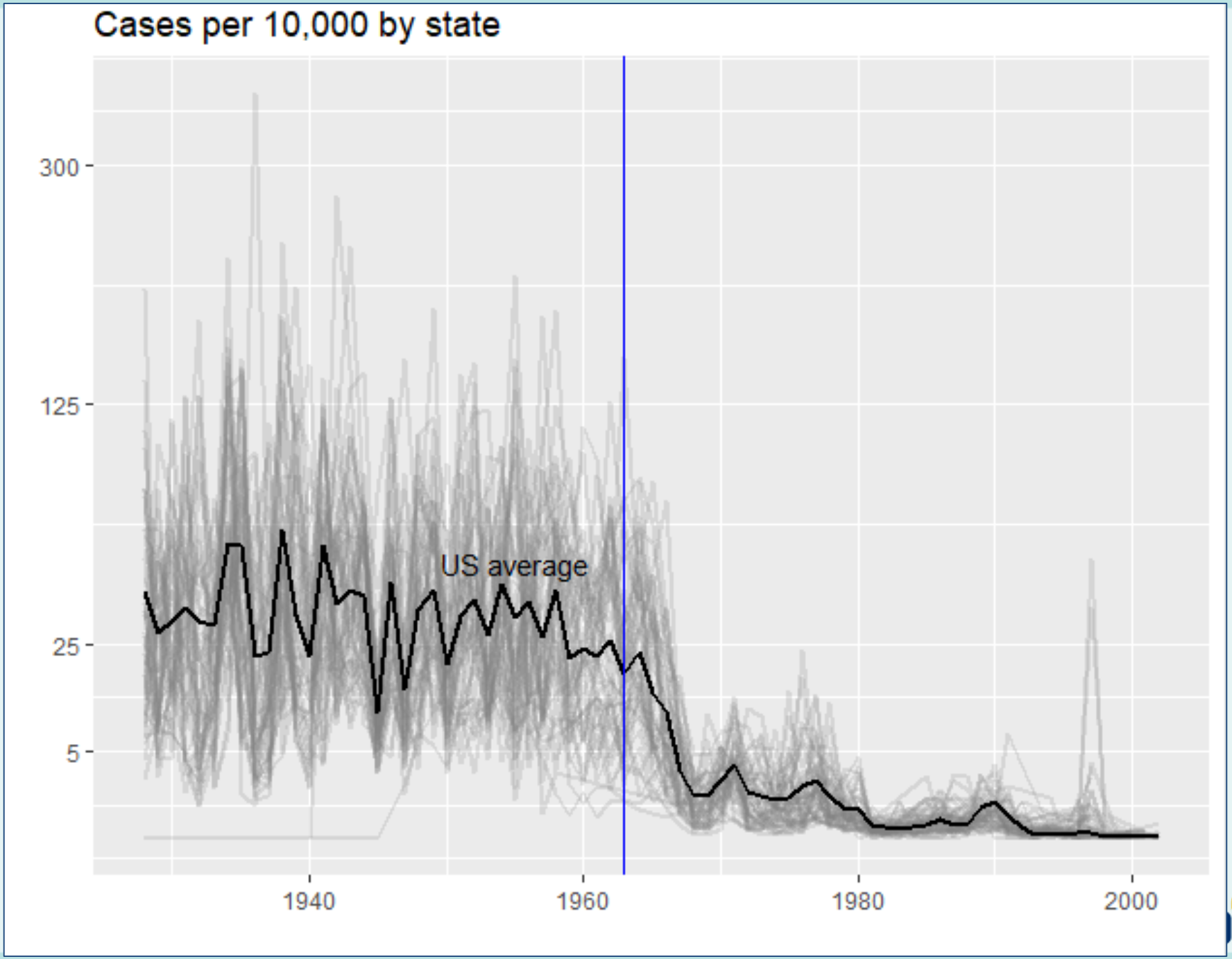
Si queremos ver detalles por estado

```
dat %>%
filter(state
=="California"
& !is.na(rate))
        %>%
ggplot(aes(year,
rate))
          +
geom_line()
          +
ylab("Cases per
10,000")
          +
geom_vline(xinterc
ept=1963, col =
"blue")
```

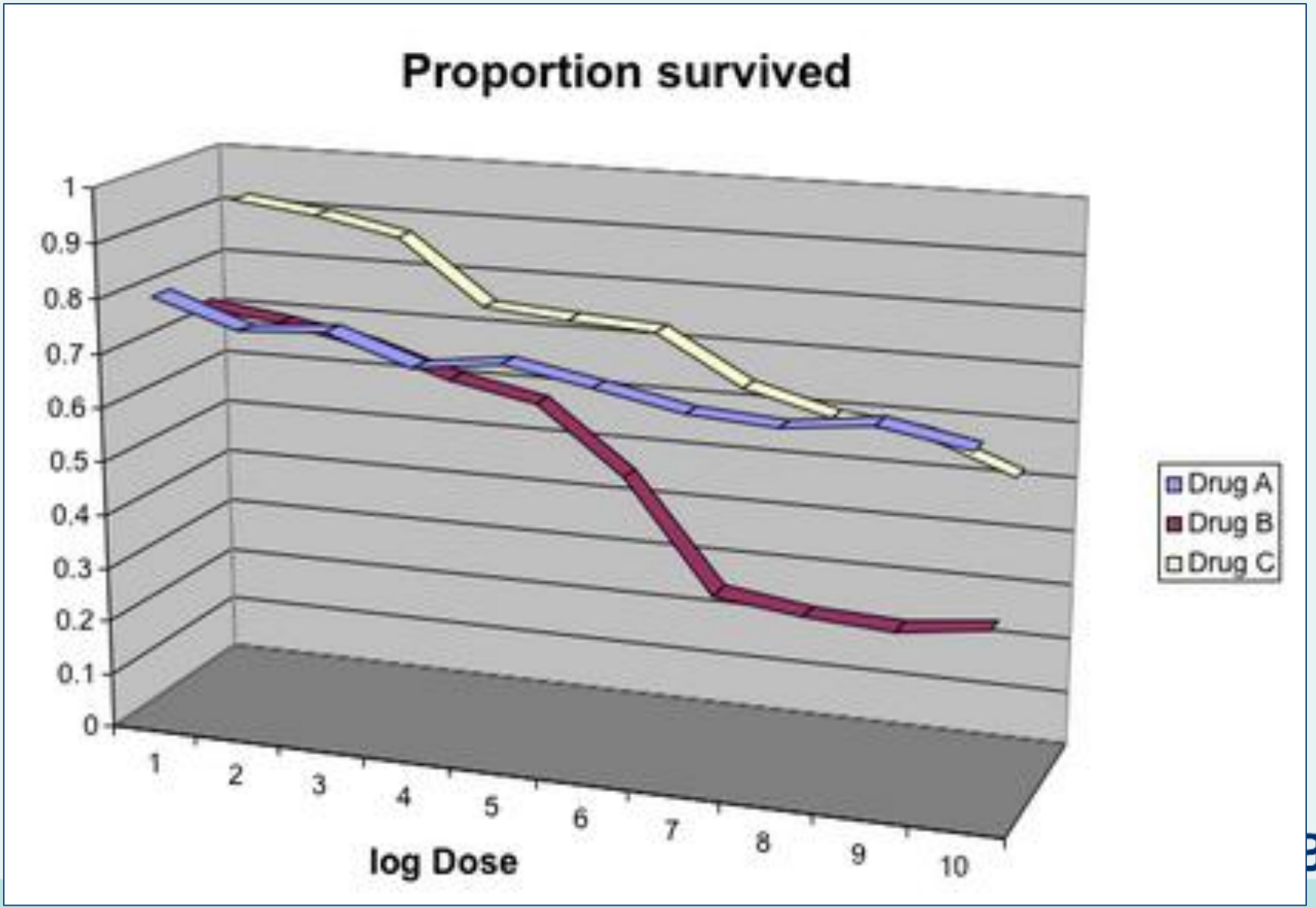


En caso de querer visualizar los datos

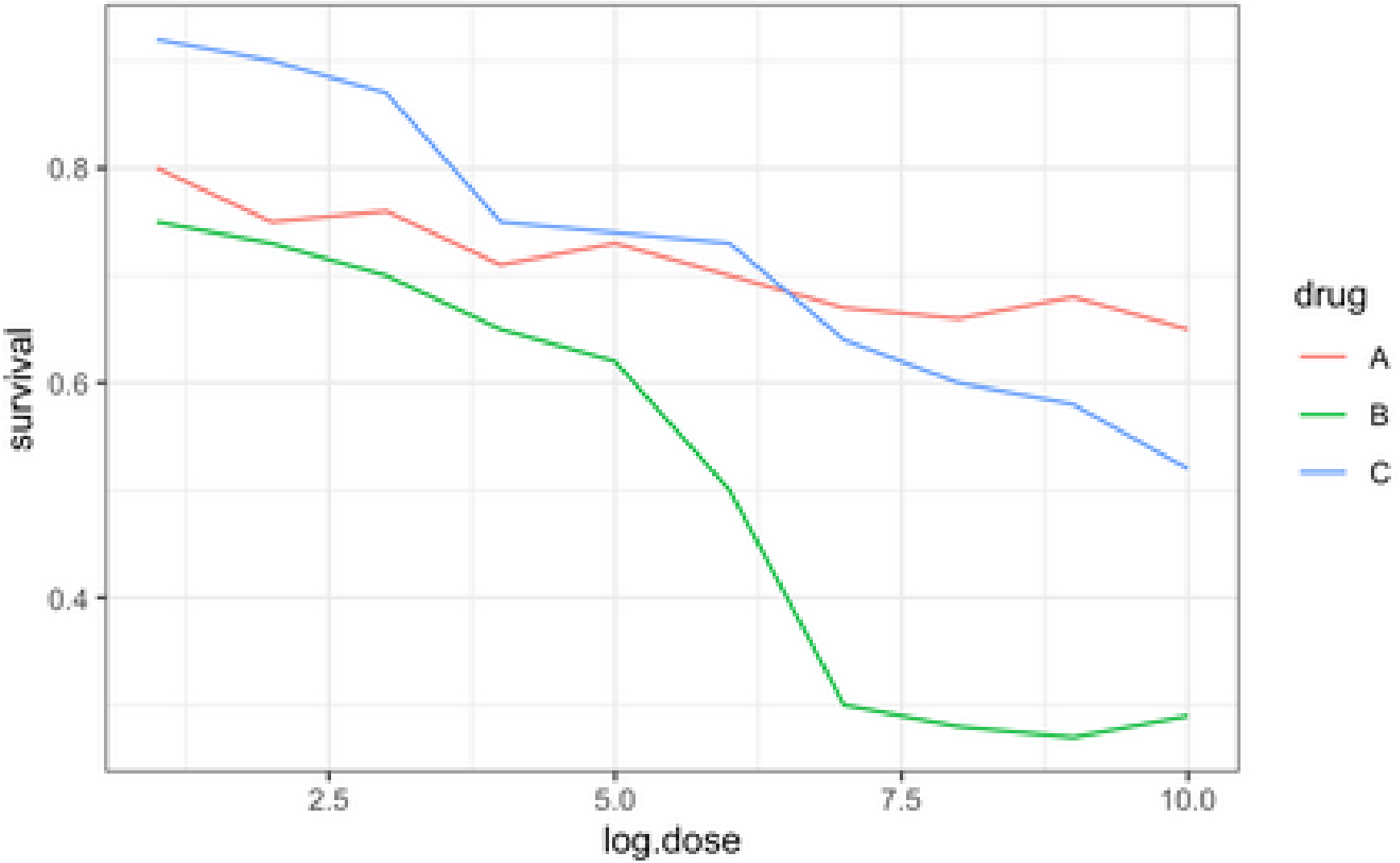
```
# Calculamos el promedio por year
avg <- us_contagious_diseases %>%  filter(disease == the_disease) %>%
group_by(year) %>% summarize(us_rate = sum(count, na.rm =
TRUE)/sum(population, na.rm = TRUE)*10000)

#Realizamos las líneas del gráfico por año y por estado
dat %>% filter(!is.na(rate)) %>% ggplot() +
  geom_line(aes(year, rate, group = state), color = "grey50",
            show.legend = FALSE, alpha = 0.2, size = 1) +
  geom_line(aes(year, us_rate), data = avg, size = 1, col = "black") +
  scale_y_continuous(trans = "sqrt", breaks = c(5, 25, 125, 300)) +
  ggtitle("Cases per 10,000 by state") +
  xlab("") +
  ylab("") +
  geom_text(data = data.frame(x = 1955, y = 50),
            mapping = aes(x, y, label = "US average"), color = "black") +
  geom_vline(xintercept = 1963, col = "blue")
```


Cases per 10,000 by state
300
125
25
5
US average
1940
1960
1980
2000

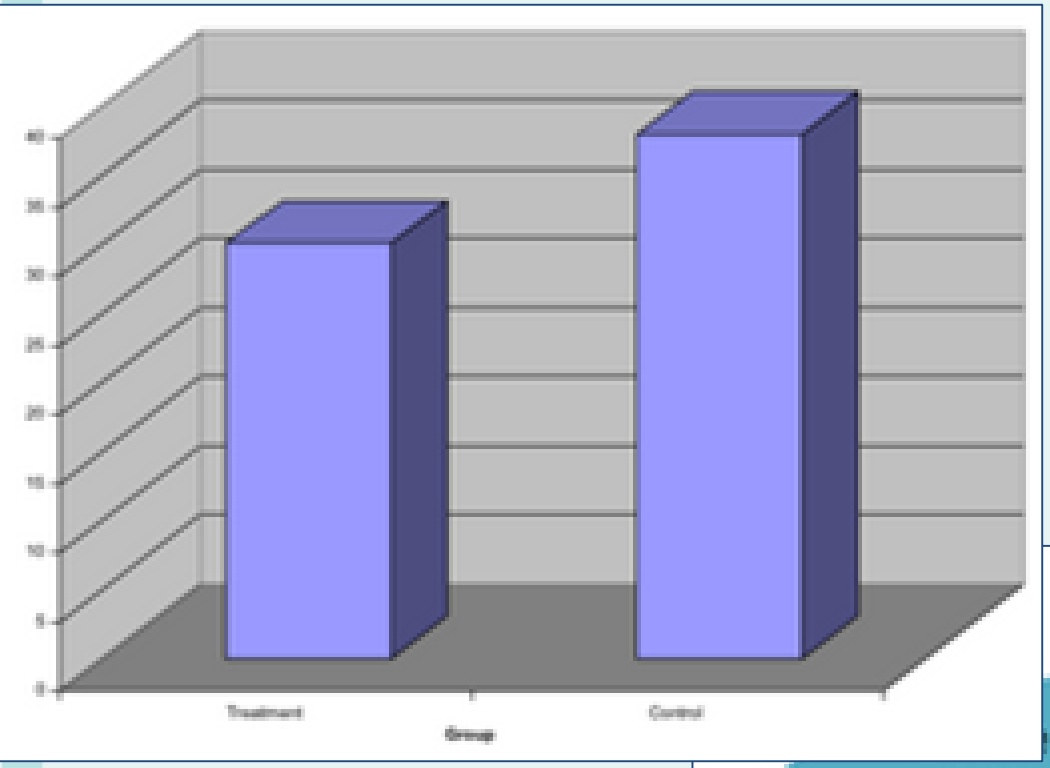
Evita gráficos pseudo gráficos 3D

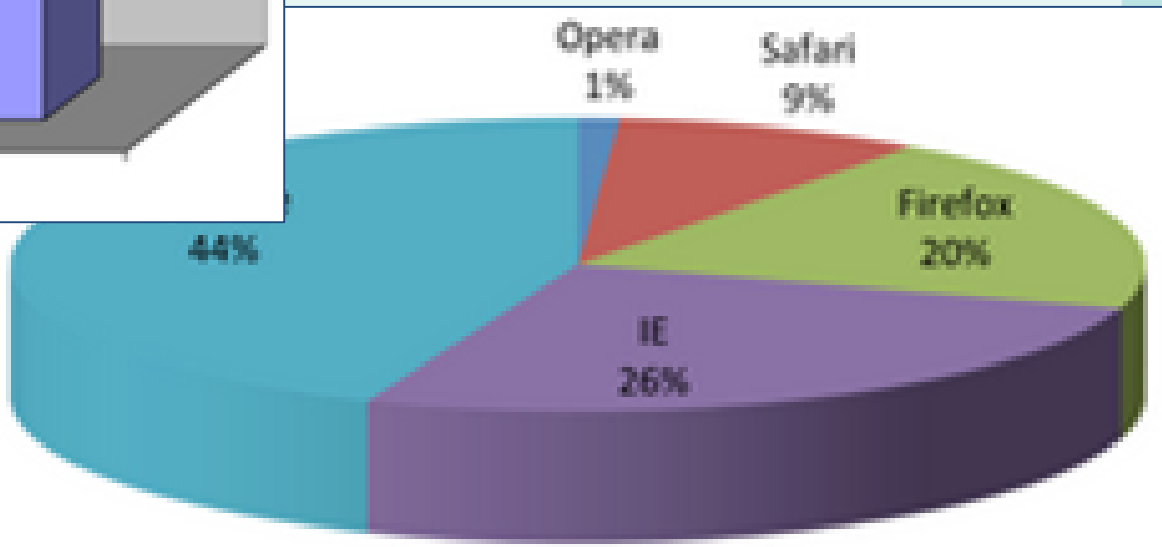


Evita gráficos pseudo gráficos 3D



Chequea si 3ra dimensión no representa una cantidad





Evita muchos dígitos significativos

- R por defecto retorna 7 dígitos significativos
- Ejemplo:
  - Totales de frecuencias cada 10.000 personas:

    La tabla se satura volviéndose más difícil de leer

state	year	Measles	Pertussis	Polio
California	1940	37.8826320	18.3397861	0.8266512
California	1950	13.9124205	4.7467350	1.9742639
California	1960	14.1386471	NA	0.2640419
California	1970	0.9767889	NA	NA
California	1980	0.3743467	0.0515466	NA

Piensa en la precisión requerida para los datos a mostrar

- En tabla anterior teníamos una precisión de 0.00001 casos cada 10,000 habitantes, vemos que 1 cifra es suficiente

state	year	Measles	Pertussis	Polio
California	1940	37.9	18.3	0.8
California	1950	13.9	4.7	2.0
California	1960	14.1	NA	0.3

- Funciones signif y round nos permiten cambiar el número de dígitos significativos o redondear
- Podemos definir los dígitos globalmente seteando options(digits = 3)

Mostrar valores a comparar en filas no en columnas

state	disease	1940	1950	1960	1970	1980
California	Measles	37.9	13.9	14.1	1	0.4
California	Pertussis	18.3	4.7	NA	NA	0.1
California	Polio	0.8	2.0	0.3	NA	NA