



# R desde Cero



## Practica 2 – Clase 3

---

### Usando otros gráficos

1. Vamos a generar otros tipos de gráficos utilizando los datos de gapminder, para ello vamos a generar algunos otros set de datos filtrando información del dataset original. El siguiente código filtra los datos de Argentina, lo copiamos y ejecutamos:

```
#Tomar solo los datos de Argentina

datosAR <- gapminder %>%

  filter (country=="Argentina")
```

2. Luego escribimos el código para hacer un gráfico de líneas, usando datosAR que es el dataset filtrado que generamos recién:

```
#Gráfico de líneas

ggplot(data = datosAR, mapping = aes(x = year, y = pop)) +

  geom_line()
```

3. Ahora vamos a generar un histograma, usando el set de datos completo, con el siguiente código:

```
#Histograma

ggplot(data = gapminder, mapping = aes(x = lifeExp)) +

  geom_histogram()
```

Aparece el siguiente mensaje en la consola:

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Este mensaje nos dice que el histograma se construyó utilizando 30 intervalos con espacios iguales. Esto se conoce en computación como un valor predeterminado; a menos que se

especifique un número distinto, R graficará con 30 intervalos de forma predeterminada. Para cambiar ese número se utiliza el parámetro `bins = numero de intervalos` de la siguiente manera (en el ejemplo 40 divisiones):

```
ggplot(data = weather, mapping = aes(x = temp)) +  
  geom_histogram(bins = 40)
```

4. Para acomodar la visualización del histograma podemos indicarle que se demarquen los intervalos con una línea blanca:

```
ggplot(data = gapminder, mapping = aes(x = lifeExp)) +  
  geom_histogram(color = "white")
```

5. También se puede cambiar los colores de las figuras:

```
ggplot(data = gapminder, mapping = aes(x = lifeExp)) +  
  geom_histogram(color = "white", fill = "steelblue")
```

6. Otro tipo de gráfico interesante para analizar diferentes distribuciones es el gráfico de cajas:

```
ggplot(data = gapminder, mapping = aes(x = continent, y=lifeExp))  
+geom_boxplot()
```

7. El gráfico de barras también es un tipo de gráfico muy utilizado, para generarlo podemos usar este código:

```
ggplot(data = gapminder, mapping = aes(x = continent)) +  
  geom_bar()
```

## Usando otras capas de los gráficos

8. Las facetas permiten generar el mismo tipo de gráfico para una categoría especificada, por ejemplo, seguro que la distribución de la expectativa de vida cambia de acuerdo al continente, para poder realizar un gráfico para cada uno, utilizamos las facetas:

```
#Facetas
```

```
ggplot(data = gapminder, mapping = aes(x = lifeExp)) +
```



# R desde Cero



```
geom_histogram(binwidth = 5, color = "white", fill =  
"steelblue") +
```

```
facet_wrap(~ continent)
```

9. Para indicar la cantidad de gráficos por fila, podemos hacerlo con el siguiente código:

```
ggplot(data = gapminder, mapping = aes(x = lifeExp)) +  
  
  geom_histogram(binwidth = 5, color = "white", fill =  
"steelblue") +  
  
  facet_wrap(~ continent, nrow=3)
```

10. Vamos a realizar un gráfico de puntos como en la primera práctica, para ello copiamos el siguiente código y lo ejecutamos:

```
ggplot(data = gapminder,  
  
  mapping = aes(x = gdpPercap, y = lifeExp))+  
  
  geom_point(alpha = 0.2)+  
  
  labs(y="Life Expentancy", x= "GDP per Capita")
```

11. Ahora vamos a agregar las facetas por continente, indicando que queremos todos los gráficos en la misma fila:

```
ggplot(data = gapminder,  
  
  mapping = aes(x = gdpPercap, y = lifeExp))+  
  
  geom_point(alpha = 0.2)+  
  
  labs(y="Life Expentancy", x= "GDP per Capita")+  
  
facet_wrap(~ continent, nrow=1)
```

12. En general en este tipo de gráfico suele interesar colocar una línea de tendencia, para ello usamos la capa de estadísticas, con el siguiente código:

```
ggplot(data = gapminder, mapping = aes(x = gdpPercap, y =  
lifeExp))+  
  
  geom_point(alpha = 0.2)+  
  
  labs(y="Life Expentancy", x= "GDP per Capita")+  
  
  # geom_smooth()
```

```
facet_wrap(~ continent, nrow=1)+
```

```
stat_smooth(method = "lm", se = F, col = "red")
```

13. El tema permite personalizar todos los elementos del gráfico, en este código de ejemplo se presentan una serie de los elementos posibles de configuración, los cuales están configurados para que no aparezcan en el gráfico con la opción **element\_blank()**:

```
ggplot(data = gapminder,  
       mapping = aes(x = gdpPercap, y = lifeExp))+  
geom_point(alpha = 0.2)+  
labs(y="Life Expentancy", x= "GDP per Capita")+  
facet_wrap(~ continent, nrow=1)+  
stat_smooth(method = "lm", se = F, col = "red")+  
theme( panel.background = element_blank(),  
       plot.background = element_blank(),  
       legend.background = element_blank(),  
       legend.key = element_blank(),  
       strip.background = element_blank(),  
       axis.text = element_text(colour = "black"),  
       axis.ticks = element_line(colour = "black"),  
       panel.grid.major = element_blank(),  
       panel.grid.minor = element_blank(),  
       axis.line = element_line(colour = "black"),  
       strip.text = element_blank()  
)
```

14. Para conocer las diferentes opciones que existe probemos la ayuda: **?theme**, buscar la propiedad **panel.background()** y buscar que valor puede tener además de **element\_blank()**. Cambiar el valor y ejecutar para analizar cómo cambia el gráfico.