



## Workshop: paquetes ggplot2 y plotly

### Analytics Research Lab

#### Introducción

El presente documento tiene como objetivo ayudarte a mejorar tus habilidades en el software Rstudio. Es una guía practica que te impulsará a desarrollar ejercicios de manera autonoma. A continuación encontrarás las principales funciones contenidas en los paquetes ggplot2 y plotly, así como sus argumentos. Inicialmente debes instalar los paquetes:

```
install.packages(c("plotly", "ggplot2"))
install.packages("gapminder")
#gapminder es el paquete donde se encuentra nuestra base de datos
```

y luego cargarlos:

```
library(ggplot2);library(gapminder)
```

#### Principales funciones del paquete ggplot2

La visualización de datos en R es una de las razones por las que el software es tan utilizado, es muy importante para cualquier científico de datos conocer acerca de ella. El paquete ggplot2 te permite construir gráficos a partir de un conjunto de datos. En esta ocasión utilizaremos la librería **gapminder**, que es un extracto de los datos Gapminder sobre la esperanza de vida, el PIB per cápita, y la población según el país.

Para realizar un grafico debemos llamar la función **ggplot()**. Sus dos principales agumentos son:

**data:** se refiere a nuestros datos.

**mapping:** se refiere a los componentes estéticos a aplicar.

```
p <- ggplot(data=gapminder, mapping = aes(x=gdpPercap,y=lifeExp))
p
```

Al ejecutar este código podemos observar que el grafico generado está vacío, esto es debido a que no le hemos asignado un objeto geométrico.

```
p+geom_point()
```

Se pueden utilizar gran cantidad de objetos geométricos para nuestros graficos que ya están incluidos en el paquete ggplot2. Para consultar las distintas opciones da click **aquí** y ve a la sección “*Layers*” para verlos todos. Para nuestra grafica en específico podríamos utilizar otro geom como **geom\_smooth**.

```
p+geom_point()+geom_smooth()
```

#### Coloreando gráficos

Para embellecer nuestro grafico visualmente y darle un toque más profesional podemos agregarle colores.

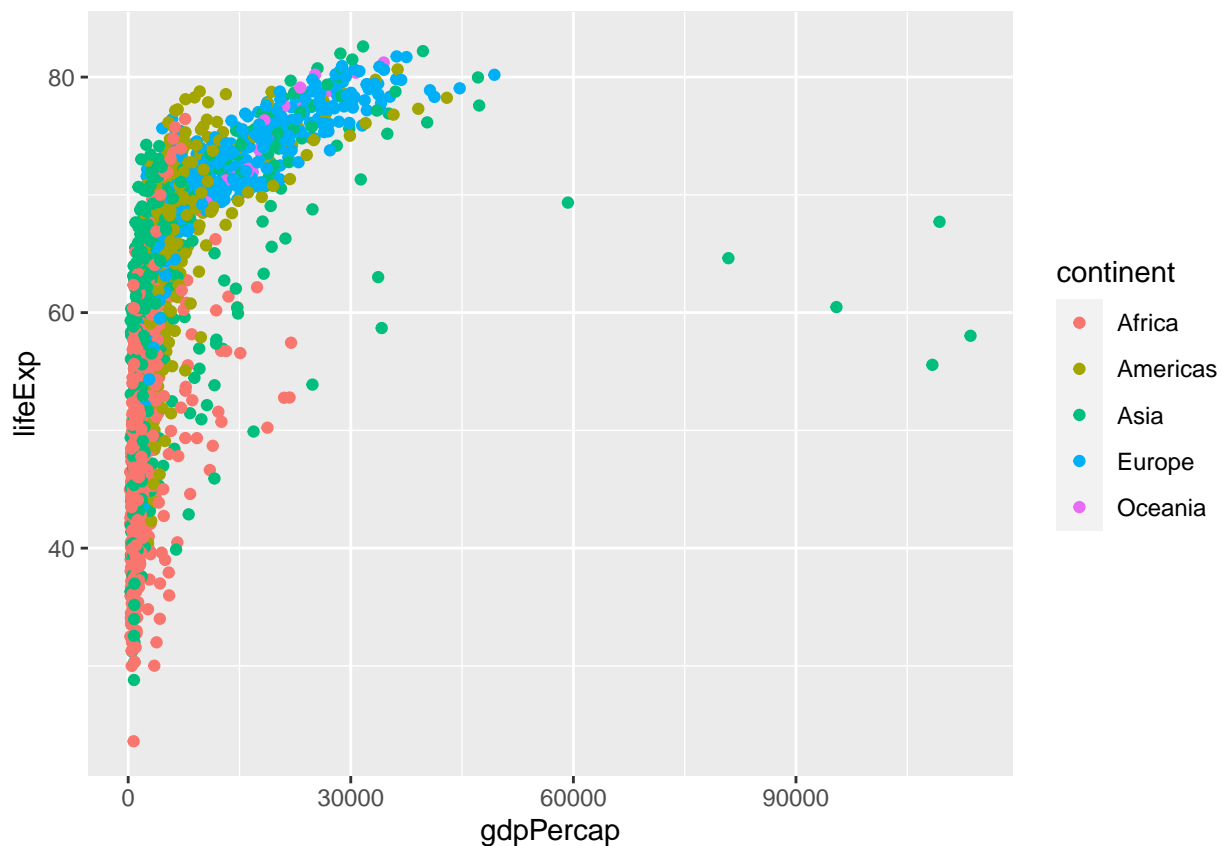
```
p <- ggplot(data=gapminder,mapping=aes(x=gdpPercap,y=lifeExp,color="purple"))
  +geom_point()
p
```

¿Qué pasó? ¿Por qué si agregamos `color="purple"` se coloreó de un rojo-rosado? Este error es bastante común. Sucede que al agregar color dentro de la función “aes” se espera que se le dé a R una variable dentro de los datos, debido a que “purple” no está como una variable dentro de los datos, R pone otro color. La forma correcta de hacerlo es colocar el color dentro del geom:

```
p <- ggplot(data=gapminder,mapping = aes(x=gdpPercap,y=lifeExp))
  + geom_point(color="purple")
p
```

En caso que querramos utilizar variables para darle color a nuestro grafico lo hacemos de la siguiente forma:

```
library(ggplot2); library(gapminder)
p <- ggplot(data=gapminder,mapping=aes(x=gdpPercap,y=lifeExp,color=continent))+
  geom_point()
p
```



Note que RStudio automáticamente le asigna el color del continente al que pertenece el respectivo punto.

### Construyendo gráficos de barras

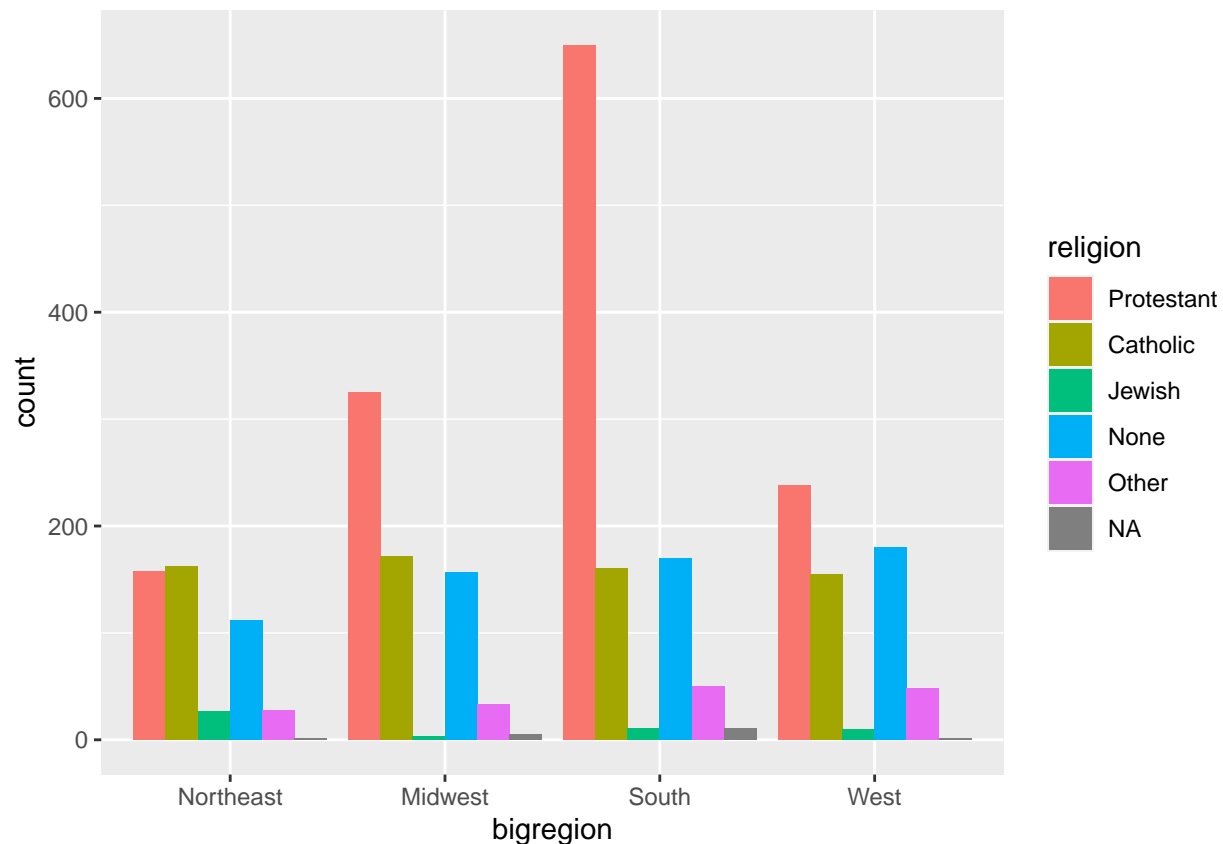
Veamos cómo construir uno de los gráficos más utilizados en ggplot2. Además de algunos factores a tener en cuenta al momento de construirlos. Para este caso, requerimos del paquete "socviz" para los datos.

```
install.packages("socviz")
library(socviz)
#Paquete donde se encuentran los datos
```

```
p2 <- ggplot(data = gss_sm,mapping=aes(x=bigregion,fill=religion))+
  geom_bar()
p2
```

Si no queremos nuestro grafico de barras apilado debemos mover la posición:

```
library(socviz)
p2 <- ggplot(data = gss_sm,mapping=aes(x=bigregion,fill=religion))+
  geom_bar(position = "dodge")
p2
```



Por último vamos a construir un gráfico simple, pero profesional, a modo de ejemplo en ggplot2. Los datos tratan la expectativa de vida en las regiones que pertenecen a la OCDE. Vamos a visualizar la brecha entre este grupo de países en comparación con Estados Unidos a lo largo del tiempo. Construimos el grafico añadiendo los datos y las estéticas.

```
library(socviz)
p3 <- ggplot(data=oecd_sum,mapping =aes(x=year,y=diff,fill=hi_lo))
```

Como objeto geométrico usaremos barras y eliminaremos las guías o convenciones:

```
p3+geom_col()+guides(fill=FALSE)
```

Finalmente, vamos a añadir todo lo referente a títulos, subtítulos, nombres de los ejes, y demás elementos que explican al gráfico.

```
p3 <- ggplot(data=oecd_sum,mapping =aes(x=year,y=diff,fill=hi_lo))
p3+geom_col()+guides(fill=FALSE)+
  labs(x=NULL,y="Diferencia en años",
       title = "Brecha en esperanza de vida con USA",
```

```

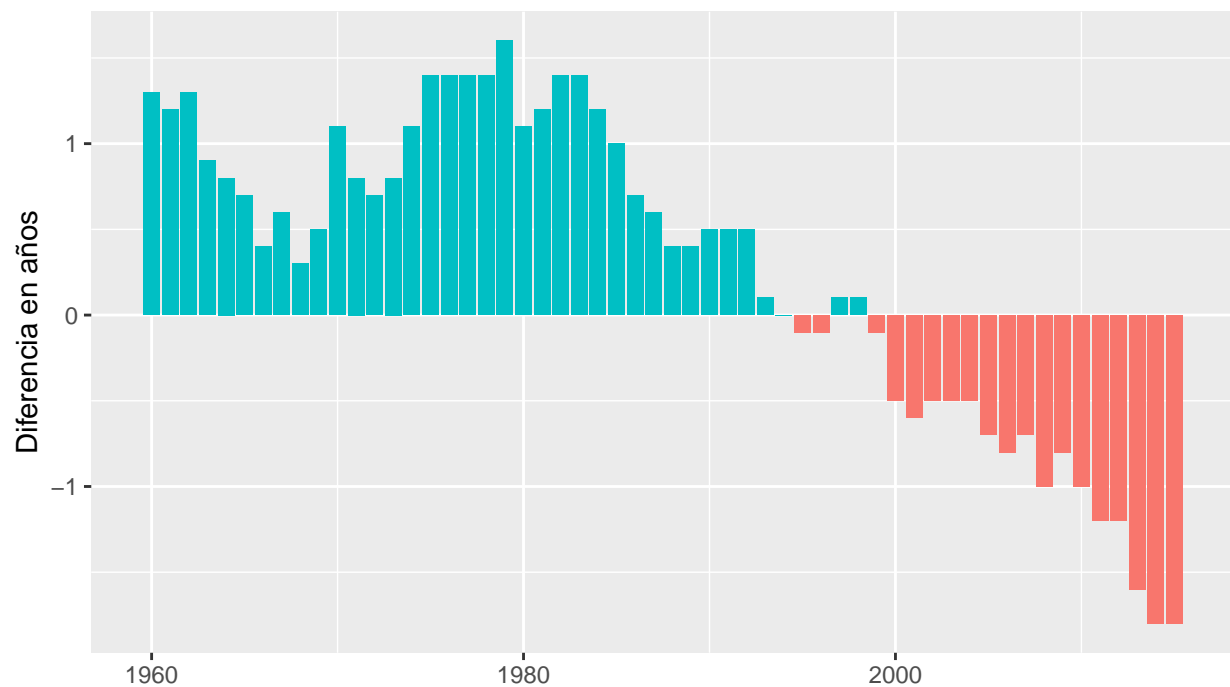
subtitle = "Diferencia de la exeperanza de vida promedio entre USA y países de la OCDE
1960-2015",
caption = "DATOS:OECD. After a chart by Christopher Ingraham, Washington Post, December 27th 2017"

```

## Warning: Removed 1 rows containing missing values (position\_stack).

## Brecha en esperanza de vida con USA

Diferencia de la exeperanza de vida promedio entre USA y países de la OCDE  
1960-2015



DATOS:OECD. After a chart by Christopher Ingraham, Washington Post, December 27th 2017

### *Bibliografía recomendada*

Libro R4ds:

R Markdown the definitive guide

Healy, K. (2018). Data visualization: a practical introduction. Princeton University Press.

RStudio -> Help -> Cheatsheets -> Data Vizualitation with ggplot2

Plotly