

# Graficos con buena calidad

*Luciano Selzer*

*28 June, 2018*

## Sistemas gráficos en R

Los gráficos son una de las maneras de explorar nuestros datos rápidamente.

En R existen varios sistemas gráficos

- Sistema gráfico de base
- Lattice
- ggplot2
- Otros

## ggplot2

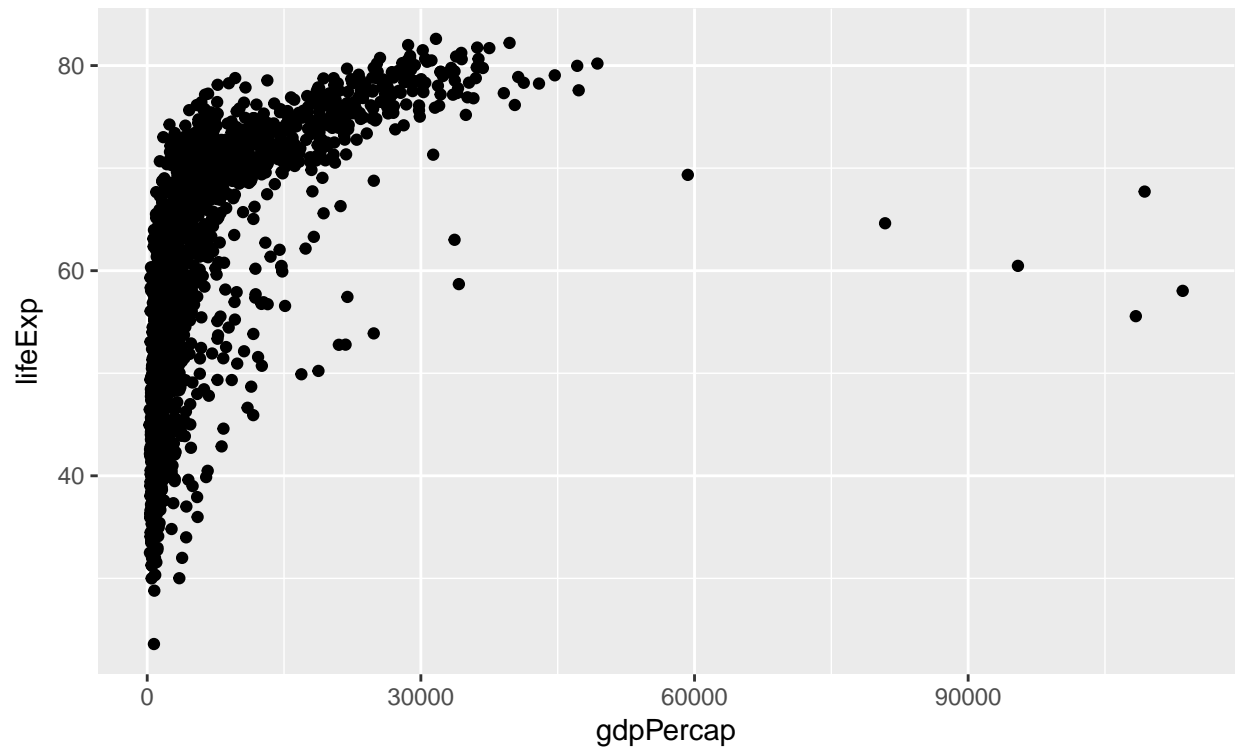
El sistema de gráficos de ggplot está contruido sobre la idea de que los gráficos pueden construirse a partir de un conjunto de componentes:

- Los datos
- El sistema de coordenadas
- Los objetos geométricos **geom** – representaciones de los datos

Una clave es que los gráficos se contruyen por capas.

---

```
library("ggplot2")
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +
  geom_point()
```



---

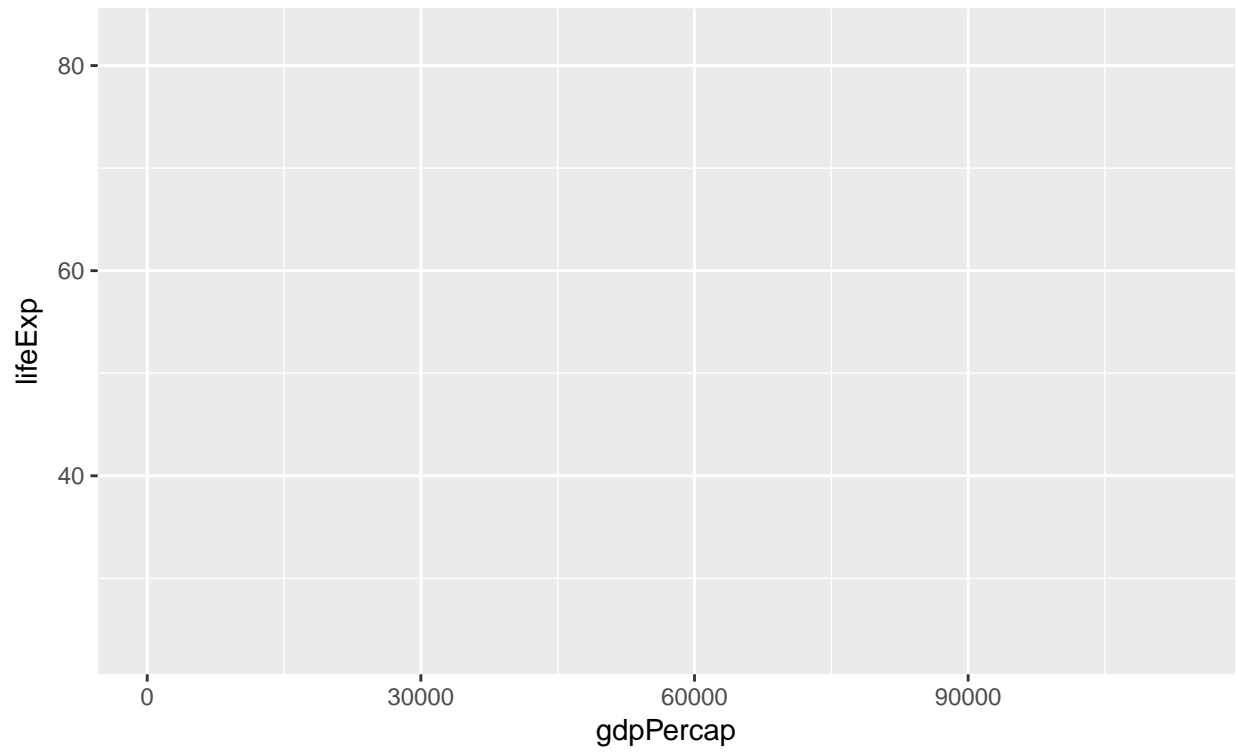
La función `ggplot` le dice a R que queremos hacer un gráfico y establece todas las opciones para el resto de las capas que agreguemos. Es posible, sobrescribir estas opciones en cada capa si es lo que queremos.

- `data` datos
- `aes` como mapear los datos

---

Si solo usamos `ggplot` muestra el gráfico vacío.

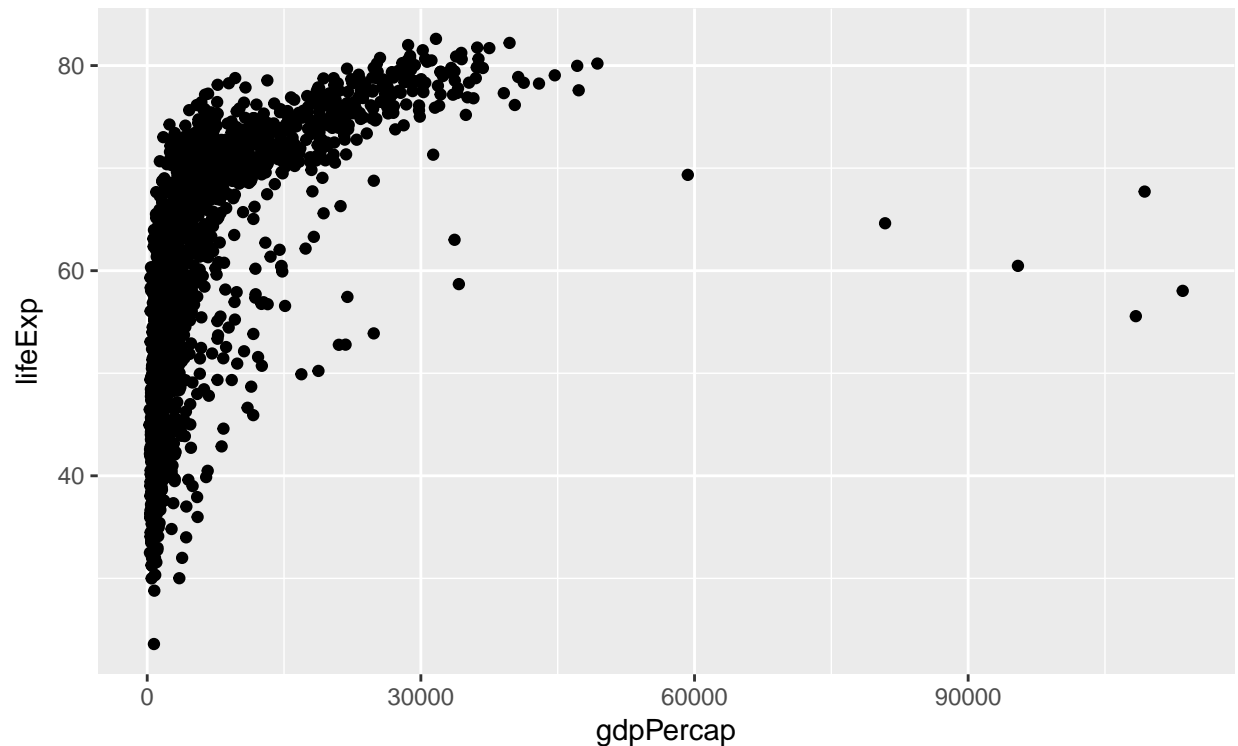
```
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp))
```



---

Tenemos que decirle como queremos representar nuestros datos. Con una nueva capa **geom**. Por ejemplo, puntos:

```
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +  
  geom_point()
```



## Ejercicio 1

Modifica el gráfico de ejemplo para mostrar como ha cambiado la expectativa de vida con el tiempo:

```
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +  
  geom_point()
```

Pista: el set de datos `gapminder` tiene una columna `year`(año).

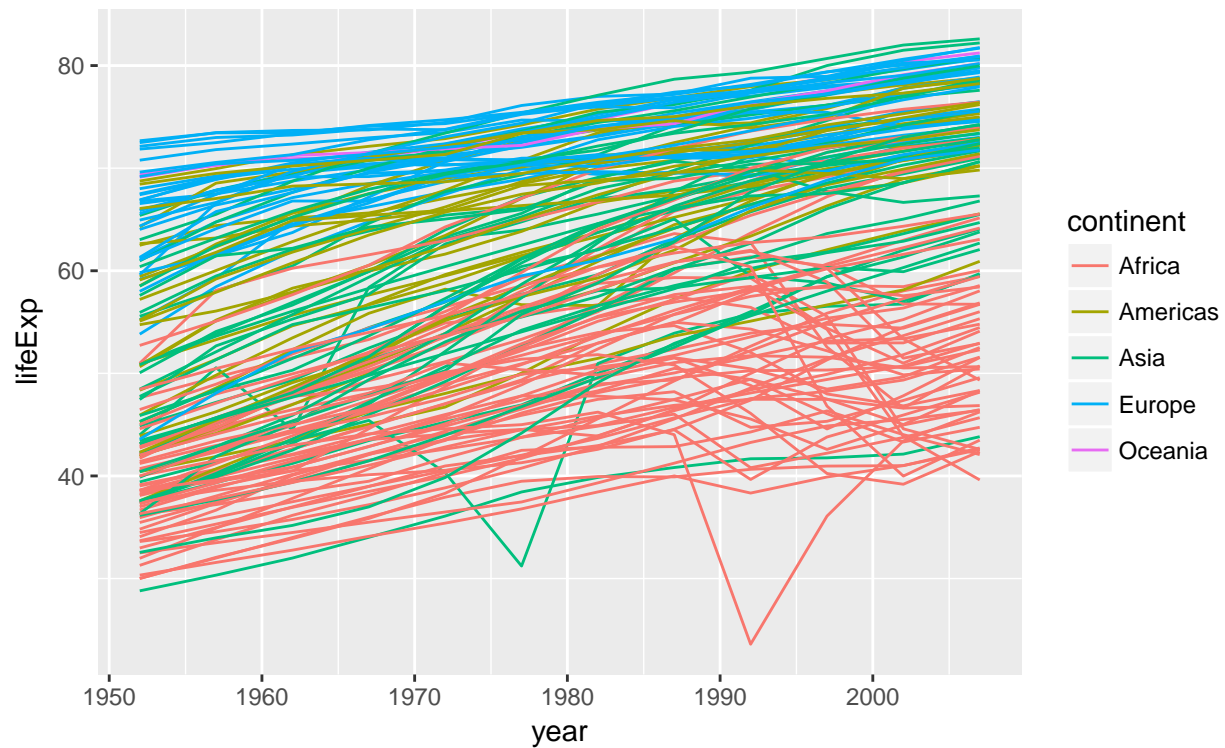
## Ejercicio 2

En el ejemplo anterior y el ejercicio usamos la función `aes` para decirle al `geom_point()` la ubicación x e y de cada punto. Hay otra propiedad estetica que podemos modificar en el punto, el `color`. Modifica el gráfico anterior para agregar color a cada punto según el continente. ¿Ves alguna tendencia en tus datos? ¿Son lo que esperabas?

## Capas

Un diagrama de puntos no es la mejor forma para mostrar el cambio en el tiempo. Es mejor usar una gráfica de líneas.

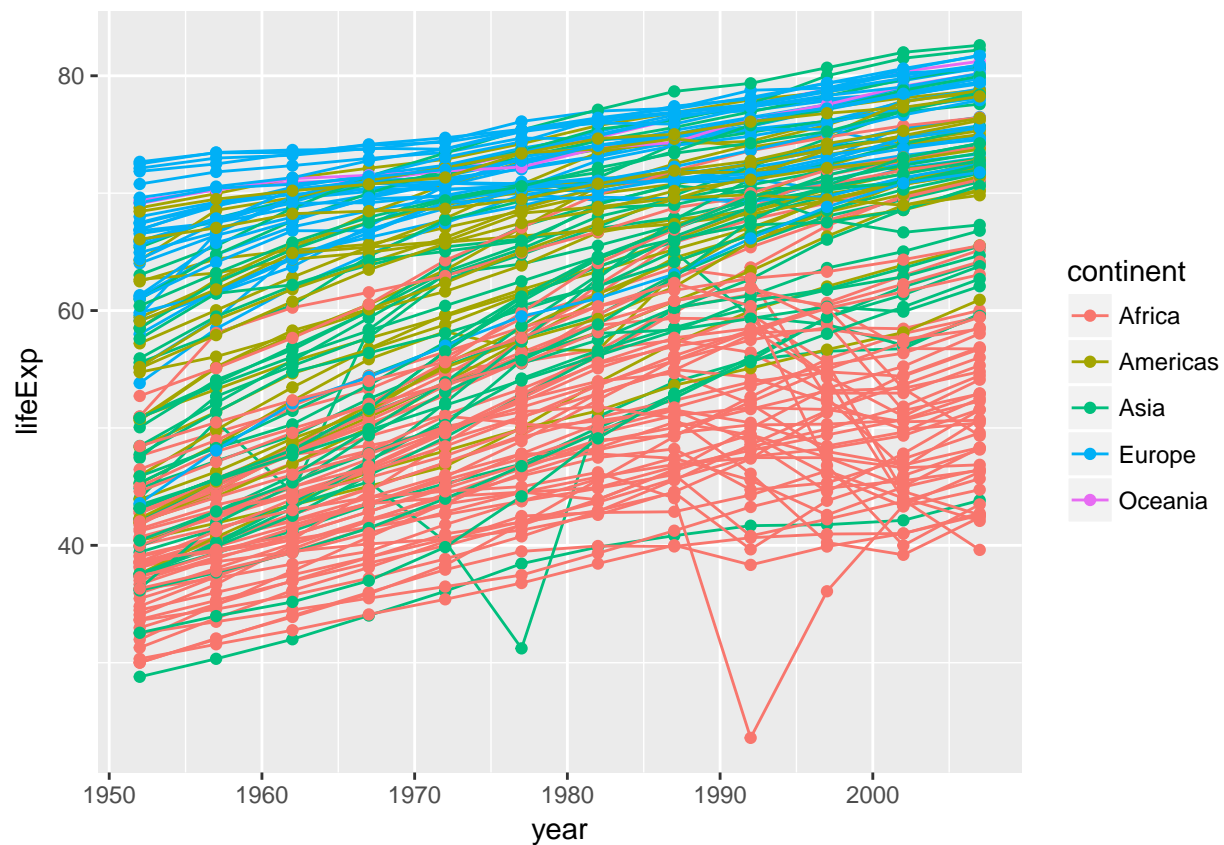
```
ggplot(data = gapminder,  
  aes(x = year, y = lifeExp,  
    by = country, color = continent)) +  
  geom_line()
```



En vez de usar `geom_point()` usamos `geom_line()` y hay que agregar el argumento `by` a `aes` para que una las líneas según el país.

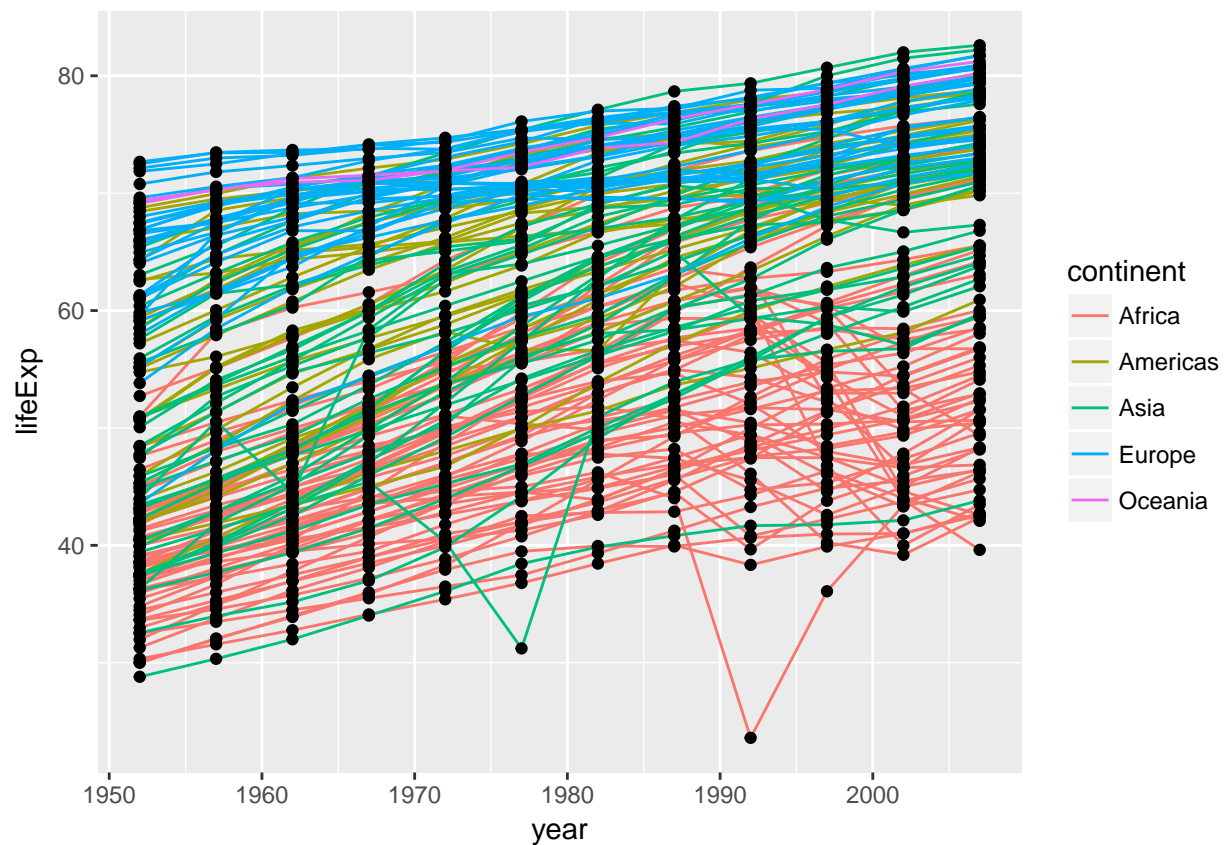
¿Pero si queremos ver los puntos y las líneas a la vez? Añadimos otra capa a nuestro gráfico:

```
ggplot(data = gapminder,
       aes(x = year, y = lifeExp,
           by = country, color = continent)) +
  geom_line() + geom_point()
```



Cada capa es dibujada por encima de la anterior

```
ggplot(data = gapminder,
       aes(x = year, y = lifeExp, by = country)) +
  geom_line(aes(color = continent)) + geom_point()
```

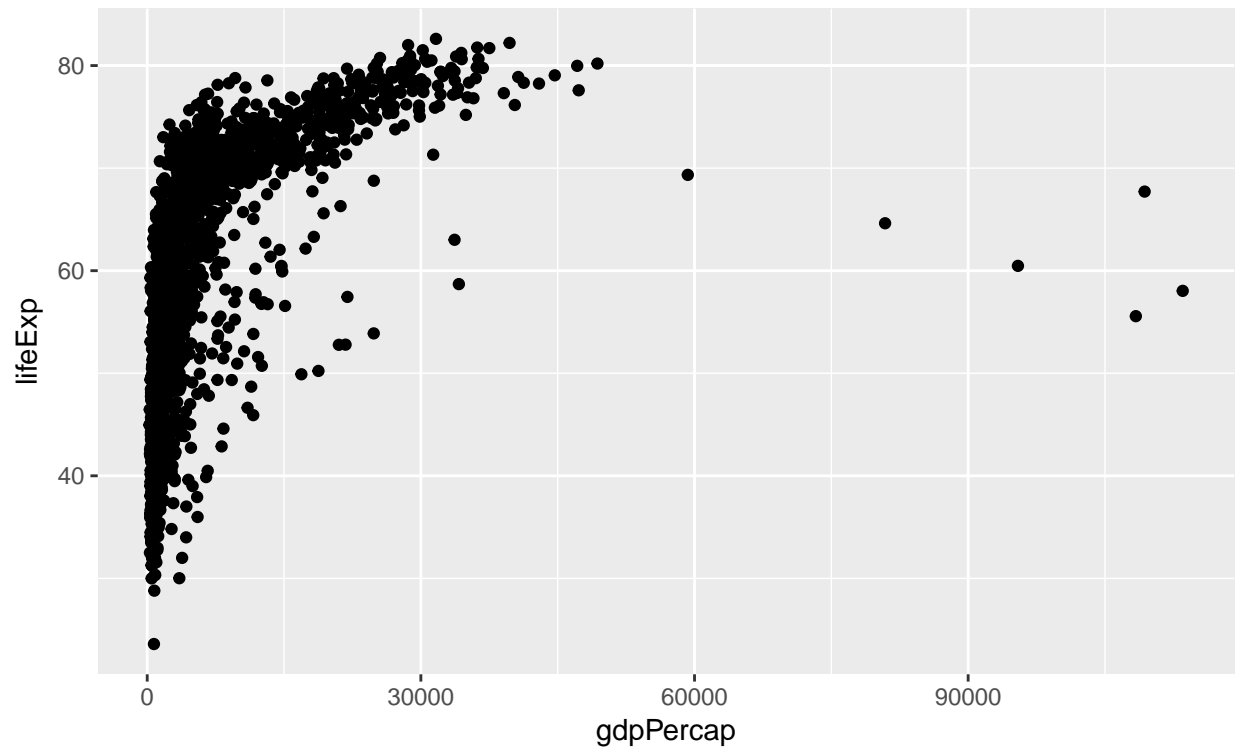


### Ejercicio 3

Cambia el orden de las capas de líneas y puntos ¿Qué sucede?

### Transformaciones y Estadísticas

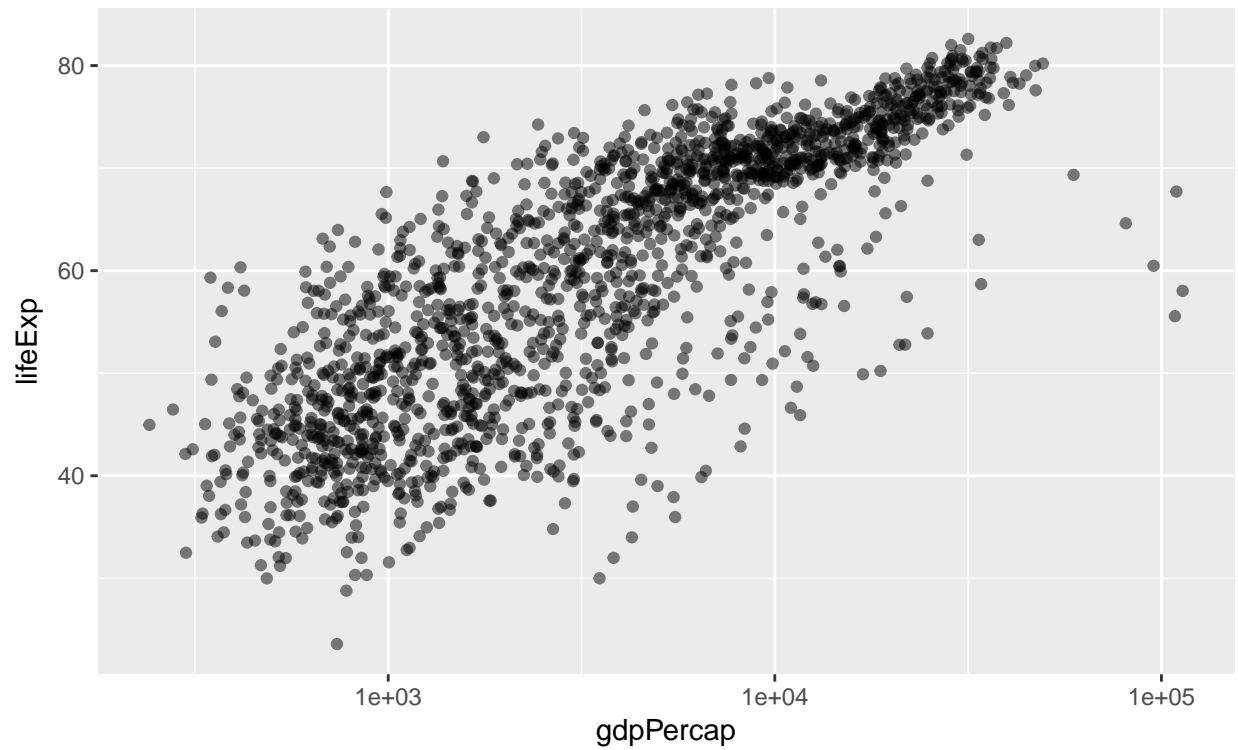
También es fácil realizar tranformaciones de los datos y superponer modelos estadísticos.



En nuestro primer gráfico era difícil ver la relación entre GDP y la expectativa de vida

```
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +  
  geom_point(alpha = 0.5) + scale_x_log10()
```



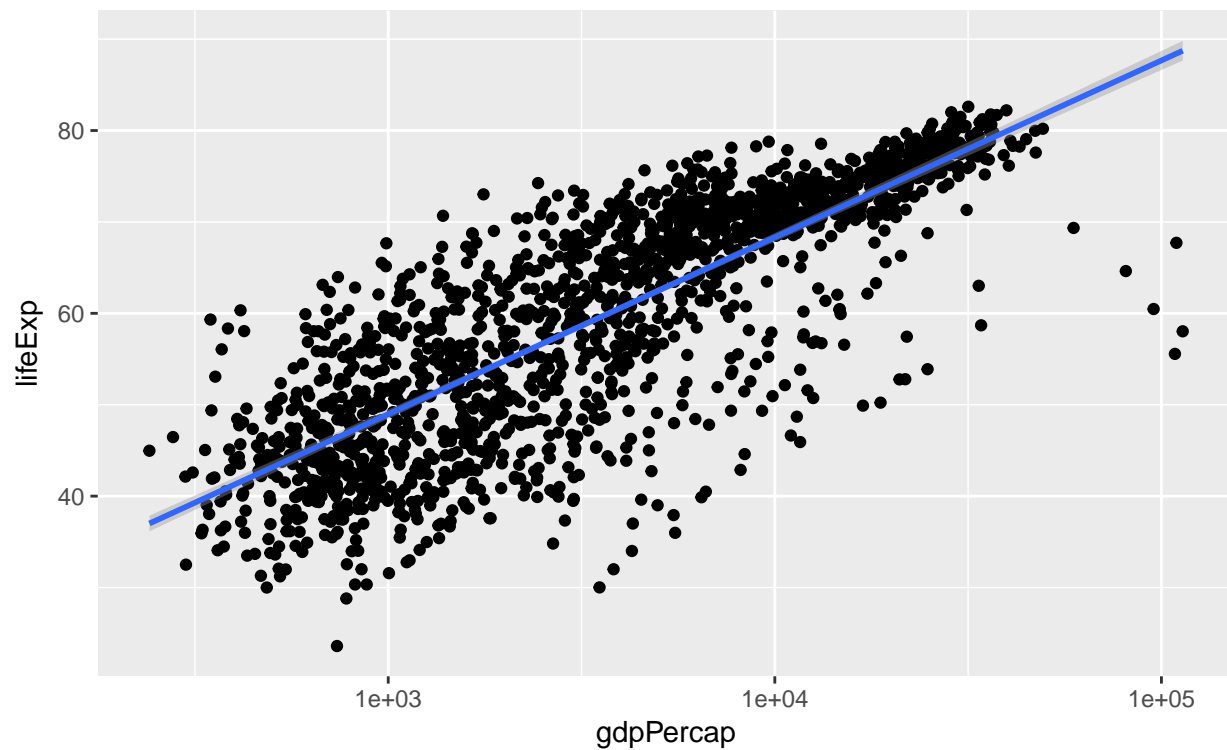


Transformamos los datos de GDP antes de graficarlos con log10 y también agregamos transparencia para que se vean los puntos donde están agrupados

---

Podemos mostrar la relación lineal que hay entre nuestros datos.

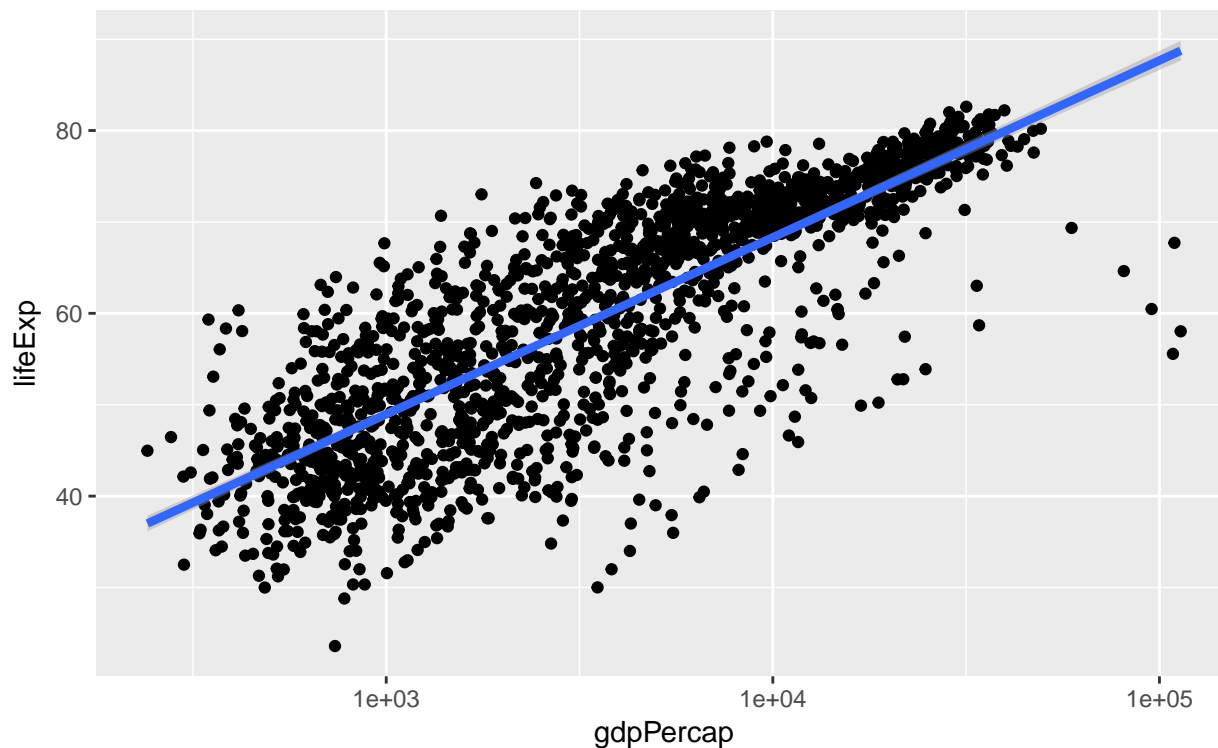
```
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +  
  geom_point() + scale_x_log10() + geom_smooth(method = "lm")
```



---

Hacer la linea más gruesa (o más fina) con el argumento **size**:

```
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +  
  geom_point() + scale_x_log10() + geom_smooth(method = "lm", size = 1.5)
```



## Aesthetics

Hay dos formas de especificar la estética de una geometría:

- Definiendola como un valor: dentro de `geom_*()`
- Mapeandola a los datos: dentro de `aes()`

## Ejercicio 4

1. Modifica el color y tamaño de los puntos en el ejemplo anterior.

Pista: no uses `aes()`

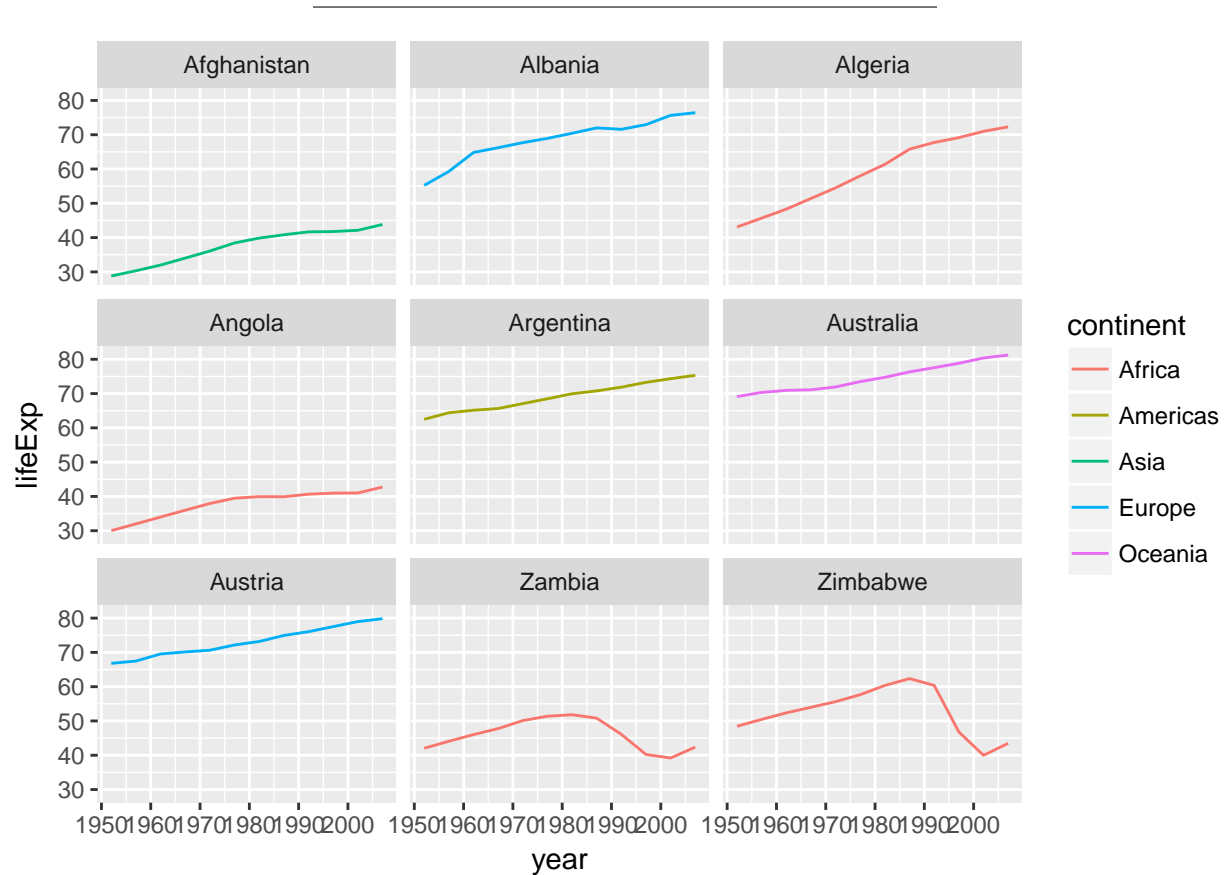
2. Modifica la solución anterior para que ahora los puntos sean de distinto color y forma según el continente agregando una nueva línea de tendencia.

## Figuras multipanel

Podemos hacer figuras con un panel por cada uno de los países. Nos vamos a enfocar en los que empiezan con A o Z.

Con `facet_*()` podemos hacer que cada país tenga su panel.

```
starts.with <- substr(gapminder$country, start = 1, stop = 1)
az.countries <- gapminder[starts.with %in% c("A", "Z"), ]
ggplot(data = az.countries,
       aes(x = year, y = lifeExp, color = continent)) +
  geom_line() + facet_wrap(~ country)
```



## Modificando el texto

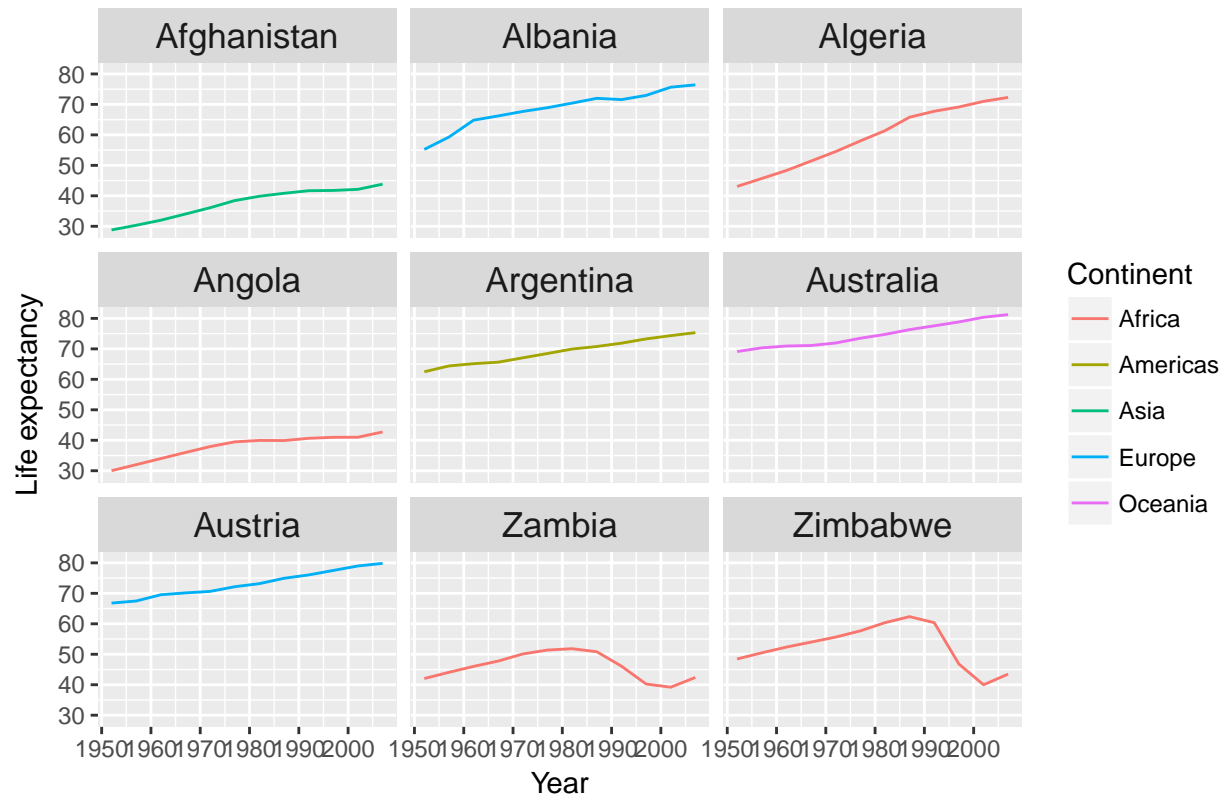
Para obtener una figura para publicar es necesario cambiar algunas cosas. Por ejemplo el nombre de los ejes debería estar completo.

Para controlar la apariencia general del gráfico usamos `theme()`. Para controlar cosas específicas a los ejes, escalas de colores, formas, etc. usamos `scale_*()`.

El nombre de los ejes lo podemos controlar desde `scale_*()` o desde `xlab()` `ylab()`

```
ggplot(data = az.countries,
       aes(x = year, y = lifeExp, color = continent)) +
  geom_line() +
  facet_wrap( ~ country) +
  xlab("Year") +
  ylab("Life expectancy") +
  ggtitle("Figura 1") +
  scale_colour_discrete( name = "Continent") +
  theme(strip.text = element_text(size = 13))
```

Figura 1



Hay muchas más cosas que se pueden hacer con ggplot.

- 
- Hay documentación detallada en el sitio web: <http://docs.ggplot2.org/current/>
  - Hay un machete muy util en RStudio.
  - Y en `cookbookR` hay instrucciones detalladas de como hacer el 99% de las cosas.