Tarea individual 2

Entregar el Viernes 26 de Abril

Entrega

Esta tarea tiene que estar disponible en su repositorio de GitHub con el resto de las actividades y tareas del curso el 26 de Abril. Asegurate que tanto Lucía como yo seamos colaboradoras de tu proyecto Tareas_STAT_NT creado hace dos semanas. Recordar seleccionar en en opciones de proyecto, codificación de código UTF-8. La tarea debe ser realizada en RMarkdown, la tarea es individual por lo que cada uno tiene que escribir su propia versión de la misma. El repositorio debe contener unicamente el archivo .Rmd con la solución de la tarea. Vamos a utilizar la librería `gapminder`, por lo que si no la usaste anteriormente tenés que instalarla y luego cargarla. Para obtener la descripción del paquete `library(help = "gapminder")` y para saber sobre la base `?gapminder`.

Recordá que todas las Figuras deben ser autocontenidas, deben tener toda la información necesaria para que se entienda la información que se presenta. Todas las Figuras deben tener leyendas, títulos. El título (caption) debe contener el número de la Figura así como una breve explicación de la información en la misma. Adicionalmente en las Figuras los nombre de los ejes tienen que ser informativos. En el YAML en Tarea_2.Rmd verás `fig_caption: true` para que salgan los `caption` en el chunk de código debes incluir `fig.cap =` "Poner el que tipo de gráfico es y algún comentario interesante de lo que ves".

Idea básica de regresión lineal

Una regresión lineal es una aproximación utilizada para modelar la relación entre dos variables que llamaremos `X` e `Y`. Donde Y es la variable que queremos explicar y X la variable explicativa (regresión simple).

El análisis de regresión ajusta una curva a travéz de los datos que representa la media de Y dado un valor especificado de X. Si ajustamos una regresión lineal a los datos consideramos "la curva media" como aquella que mejor ajusta a los datos.

Algunas veces ajustamos curvas genéricas promediando puntos cercanos entre si con métodos de suavizado no necesariamente lineales. ¿Cómo incluimos una recta de regresión en nuestro gráfico?

ajustamos una recta de regresión a los datos en Para agregar una linea de regresión o una curva tinenes que agregar una capa a tu gráfico `geom_smoth`. Probablemente dos de los argumentos más útiles de `geom_smoth` son:

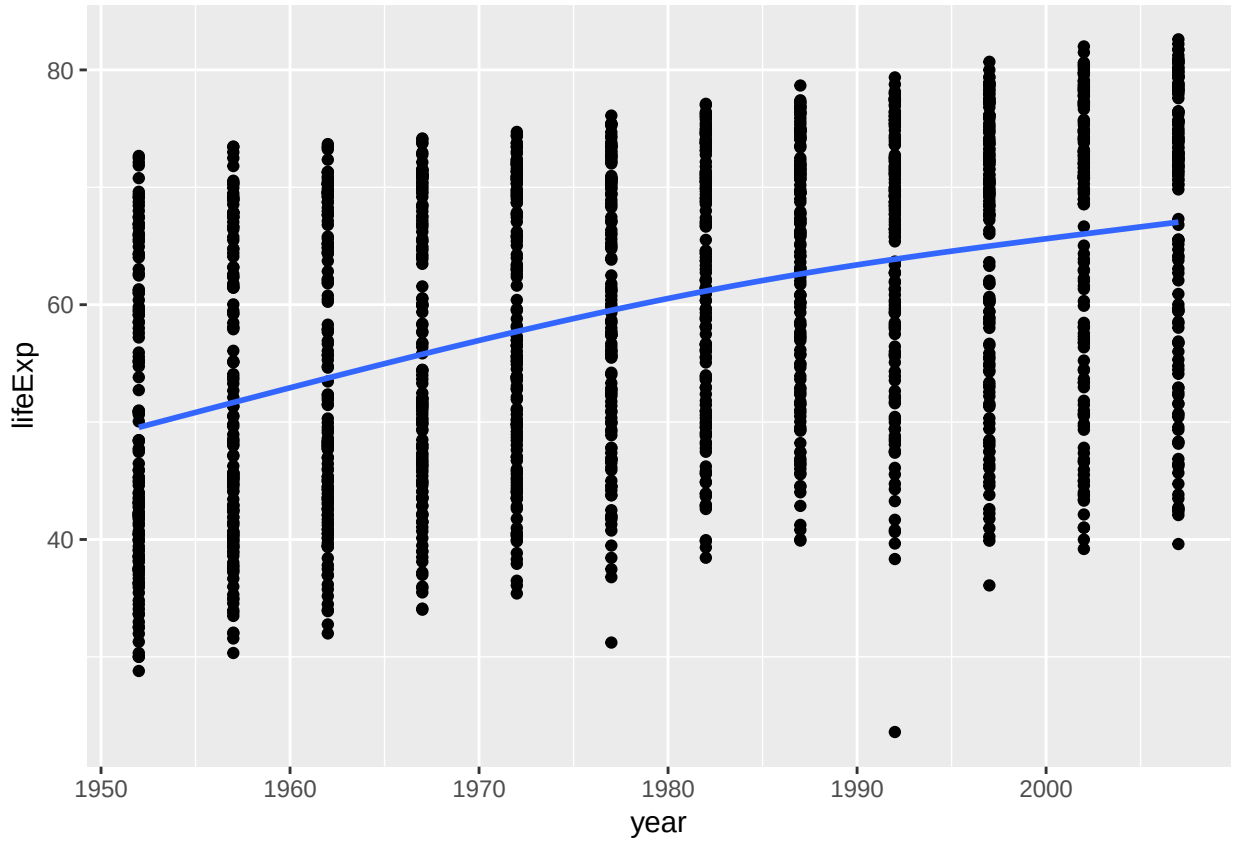
- `method = ...`
  - ... "lm" para una linea recta. `lm` "Linear Model".

- ...otro para una curva genérica (llamada de suavizado; por defecto, es la parte `smooth` de `geom_smooth`).
- `se=...` controla si los intervalos de confianza son dibujados o no.

Ejemplo:

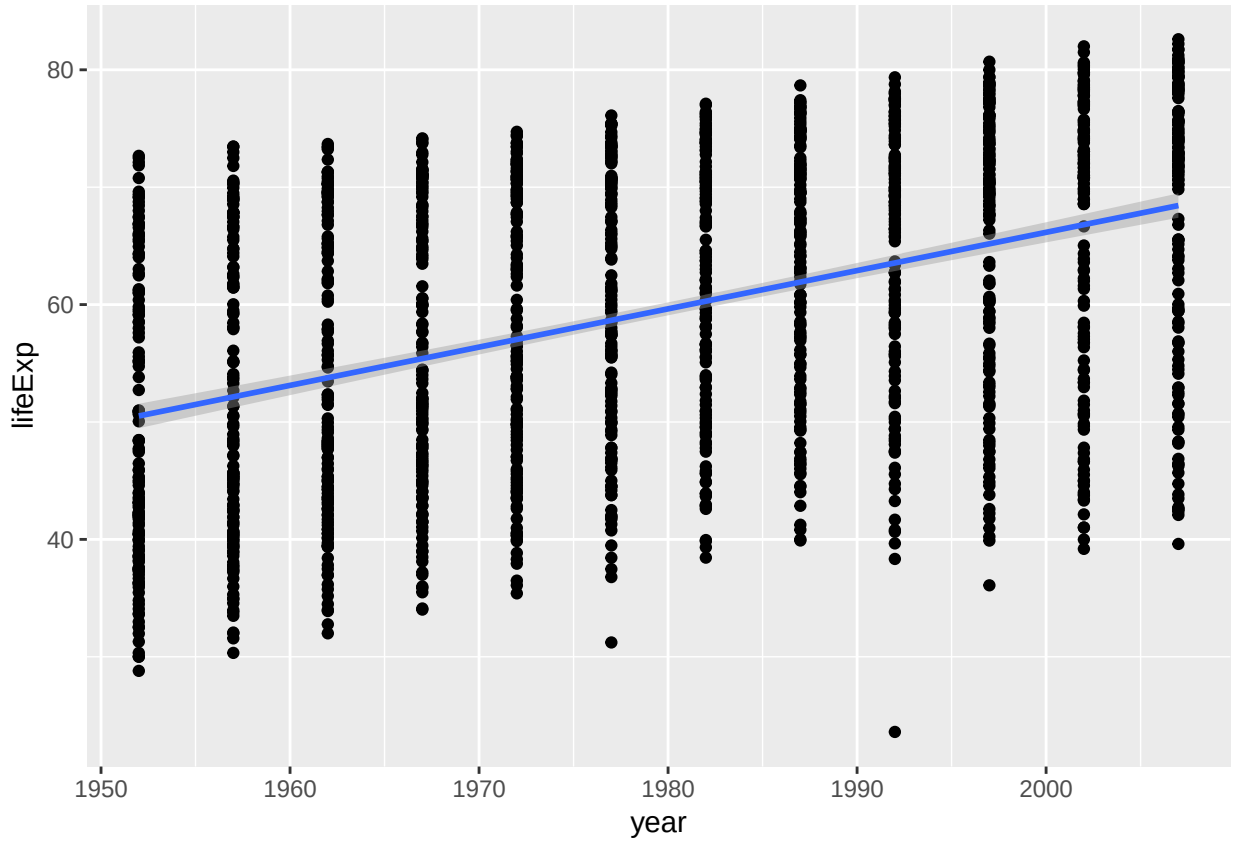
```
vc1 <- ggplot(gapminder, aes(year, lifeExp)) +
    geom_point()
vc1 + geom_smooth(se = FALSE)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



En este caso `geom_smooth()` está usando `method = 'gam'`

```
vc1 + geom_smooth(method = "lm")
```



Ejercicio 1

Hacer un gráfico de dispersión que tenga en el eje `x year` y en el eje `y lifeExp`, los puntos deben estar coloreados por la variable `continent`. Para este plot ajustá una recta de regresión para cada continente sin incluir las barras de error. Las etiquetas de los ejes deben ser claras y describir las variables involucradas. Incluir un `caption` en la Figura con algún comentario de interés que describa el gráfico. El resto de los comentarios del gráfico se realizan en el texto.

```
gapminder %>%
   ggplot(aes(y = lifeExp, x = year, color = continent)) +
   geom_point() +
   geom_smooth(method = "lm", se = FALSE) +
   labs(y = "Life expectancy at birth (in years)", x = "Time (in years)", color = NULL)
   theme(axis.title = element_text(face = "bold"))
```

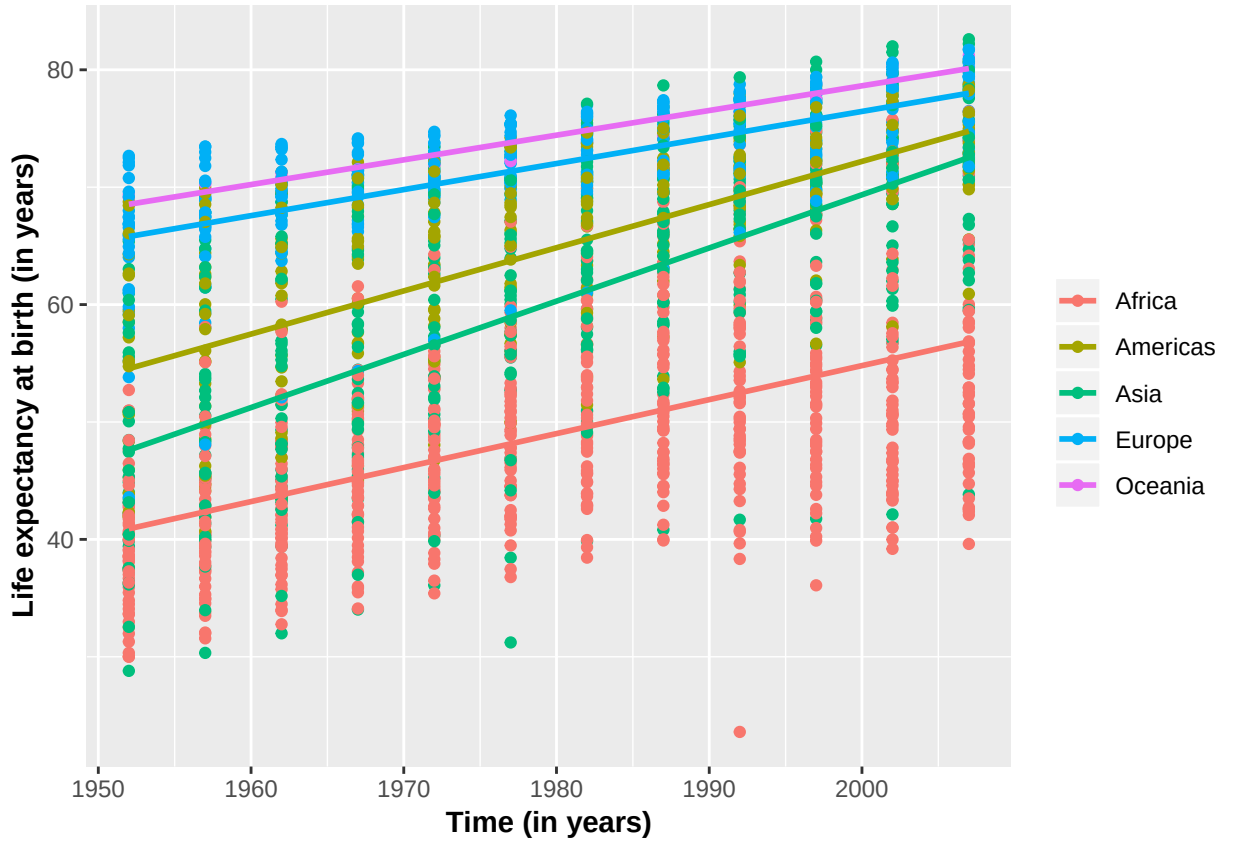


Figure 1: Scatter plot of life expectancy at birth (in year) and years with OLS regression lines, by continent

Ejercicio 2

Omitir la capa de `geom_point()` del gráfico anterior. Las lineas aún aparecen aunque los puntos no. ¿Porqué sucede esto? `aes` and `data` in ggplot call.

```
gapminder %>%
   ggplot(aes(y = lifeExp, x = year, color = continent)) +
   geom_smooth(method = "lm", se = FALSE) +
   labs(y = "Life expectancy at birth (in years)", x = "Time (in years)", color = NULL)
   theme(axis.title = element_text(face = "bold"))
```

Bien, ggplot2 trabaja con layers.

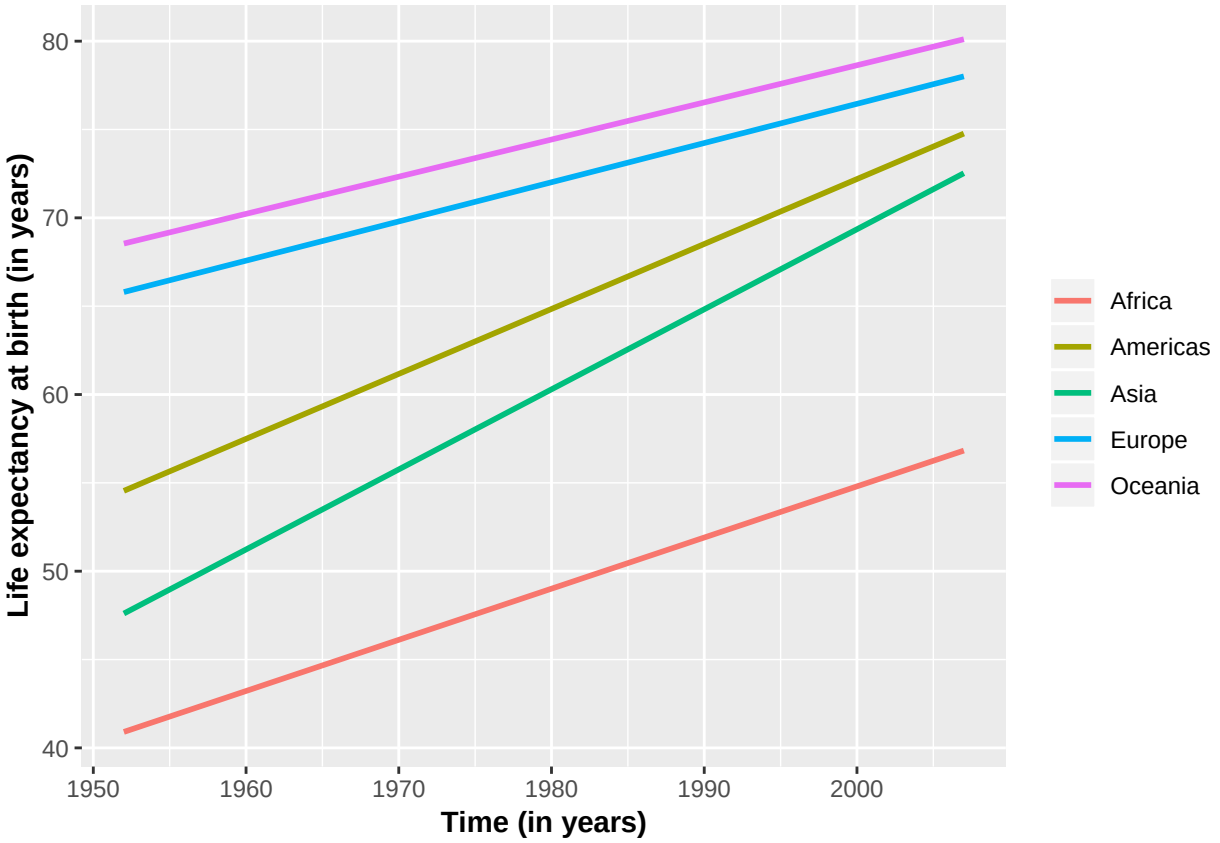
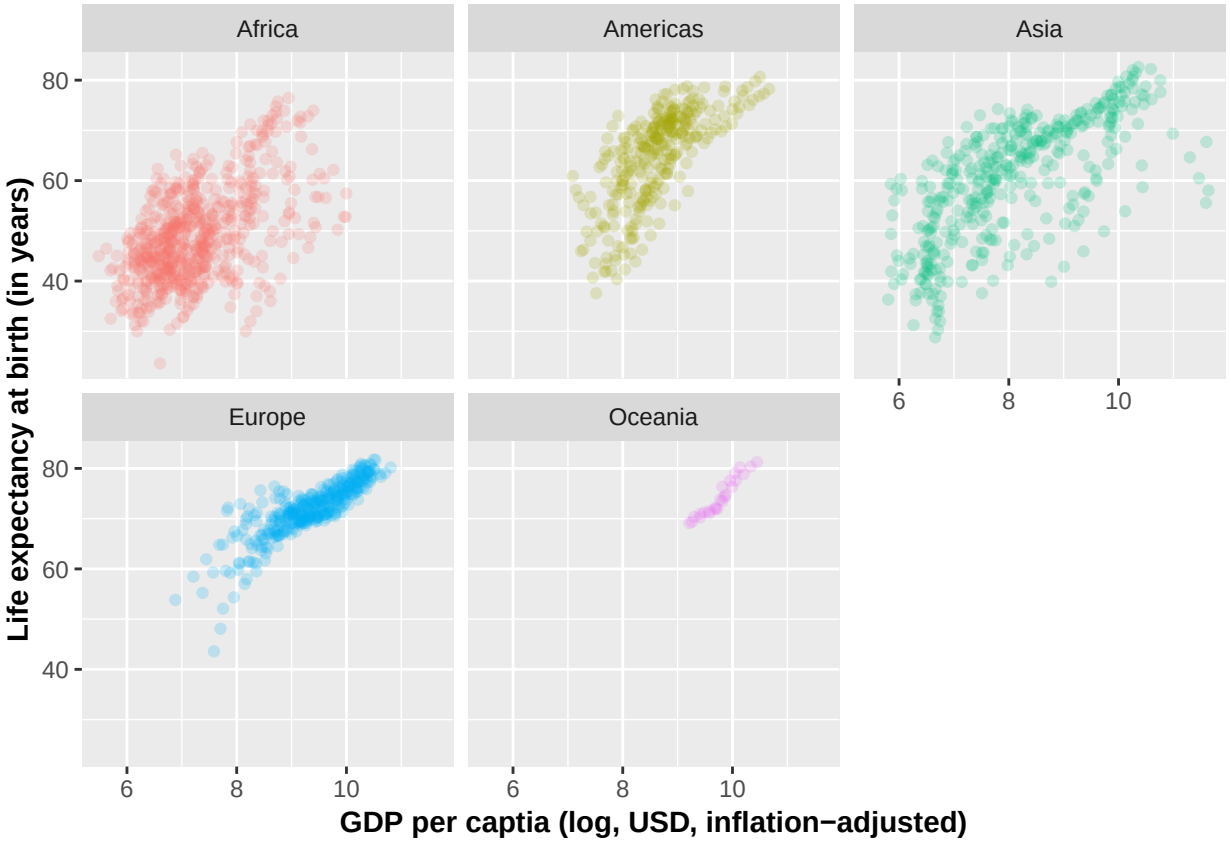


Figure 2: OLS regression of years and life expectancy at birth (in years), by continent

Ejercicio 3

El siguiente es un gráfico de dispersión entre `lifeExp` y `gdpPercap` coloreado por la variable `continent`. Usando como elemento estético color (`aes()`) nosotros podemos distinguir los distintos continentes usando diferentes colores de similar manera usando forma (`shape`).



El gráfico anterior está sobrecargado, ¿de que forma modificarías el gráfico para que sea más clara la comparación para los distintos continentes y porqué? Usar `gdpPercap` en logaritmos, fijar `alpha` en 0.2 y `facet_wrap` por `continent`.

Las etiquetas de los ejes deben ser claras y describir las variables involucradas. Comentá alguna característica interesante que describa lo que aprendes viendo el gráfico. `log(gdpPercap)` es buen predictor lineal de `lifeExp`.

Bien, el log de gdpPercap es un posible buen predictor.

Ejercicio 4

Hacer un gráfico de lineas que tenga en el eje `x year` y en el eje `y gdpPercap` para cada continente en una misma ventana gráfica. En cada continente, el gráfico debe contener una linea para cada país a lo largo del tiempo (serie de tiempo de `gdpPercap`). Las etiquetas de los ejes deben ser claras y describir las variables involucradas. Incluir un `caption` en la Figura con algún comentario de interés que describa el gráfico.

```
ggplot(gapminder) +
   geom_line(aes(x = lubridate::make_date(year), y = gdpPercap/1000,
                 color = continent, group = country), show.legend = FALSE) +
   facet_wrap(~continent) +
```

```
   labs(x = "Time (in years)", y = "GDP Per capita (thousands USD, inflation-adjusted)",
        color = NULL) +
   theme(axis.title = element_text(face = "bold"))
```

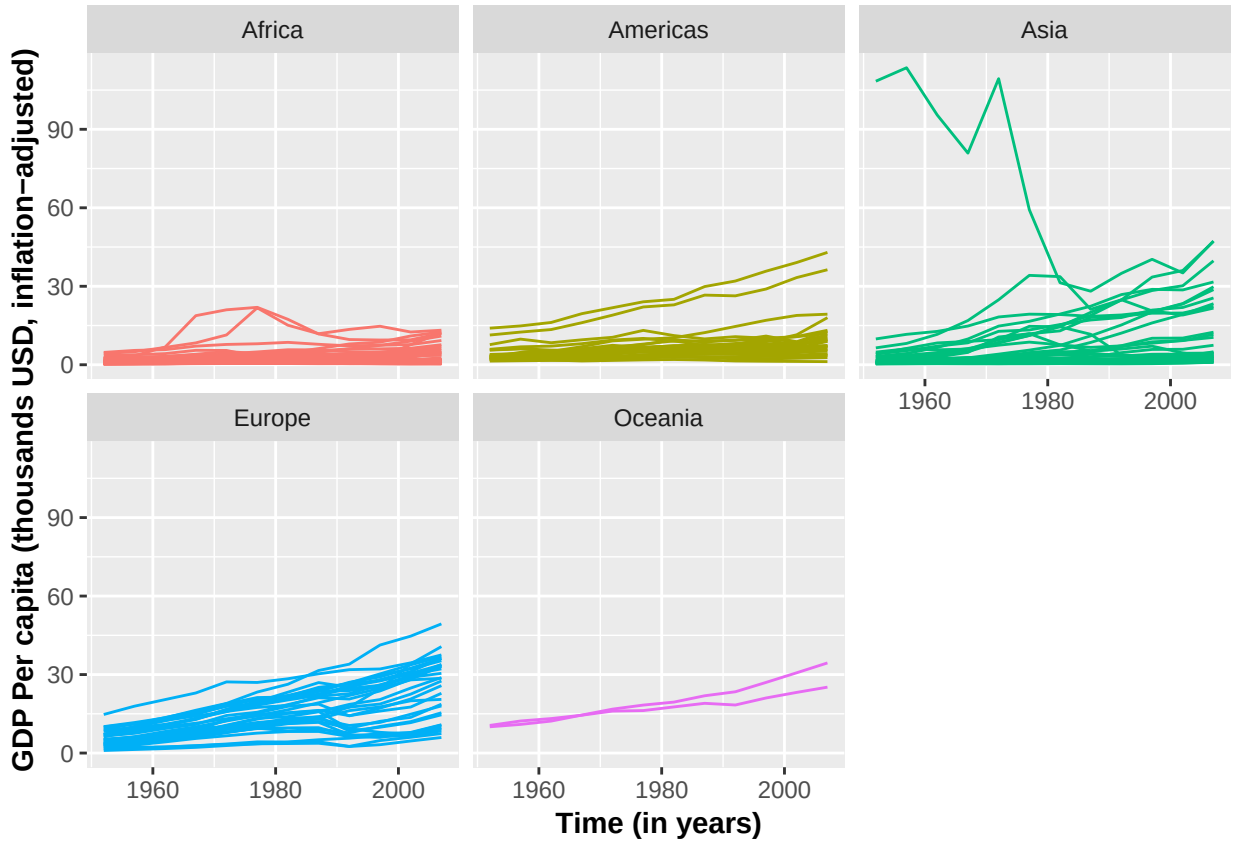


Figure 3: Time series of GDP per capita per country by continent.

Bien.

Ejercicio 5

Usando los datos `gapminder` seleccione una visualización que describa algún aspecto de los datos que no exploramos. Comente algo interesante que se puede aprender de su gráfico.

```
gapminder %>%
   filter(year %in% c(1952, 1962, 1972, 1982, 1992, 2007)) %>%
ggplot(aes(x = log(pop), y = log(gdpPercap) , colour = continent)) +
   geom_point(alpha = 0.3) +
   labs(x = "Population size (in logs)",
```

```
       y = "GDP per captia (log, USD, inflation-adjusted)",
       color = NULL) +
  facet_wrap(~year) +
  theme(axis.title = element_text(face = "bold"))
```

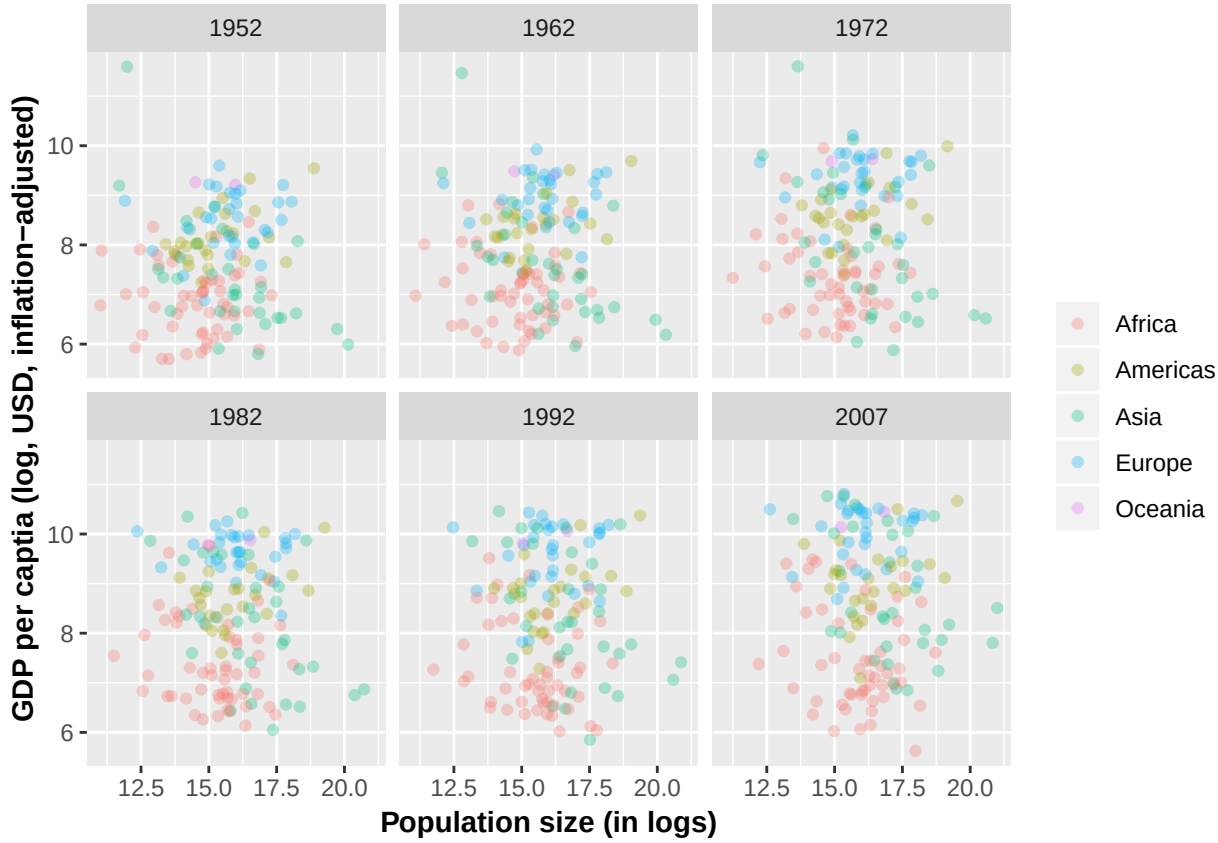


Figure 4: Scatter plot of Population sizes and GDP per capita by continent. Larger countries are not, and have not always been, the richest ones.

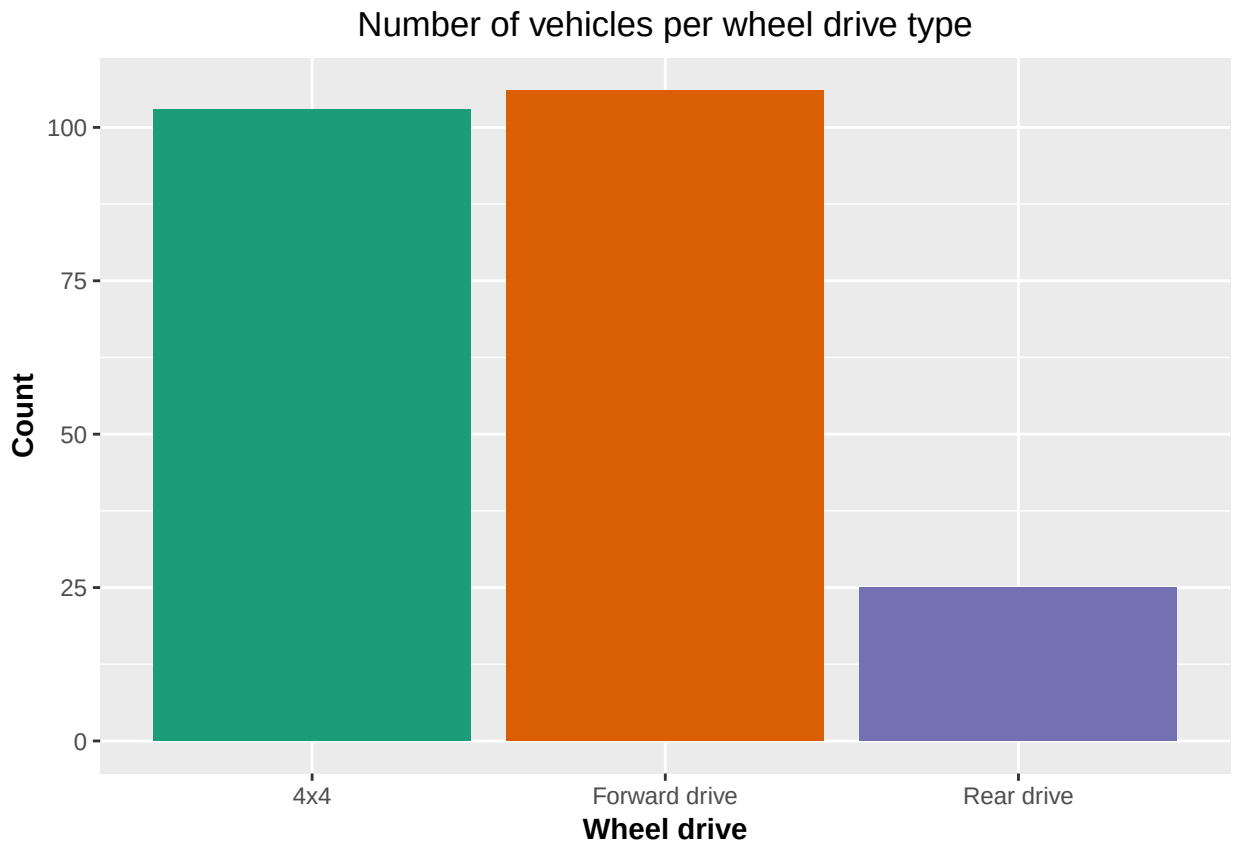
Bien.

Ejercicio 6

Con los datos `mpg` que se encuentran disponible en `ggplot2` hacer un gráfico de barras para la variables `drv` con las siguientes características:

- Las barras tienen que estar coloreadas por `drv`
- Incluir usando `labs()` el nombre de los ejes y título informativo.
- Usá la paleta de colores `Dark2`, mirá la ayuda de `scale_colour_brewer()`.

```
ggplot2::mpg %>%
   ggplot() +
   geom_bar(aes(drv, fill = drv), show.legend = FALSE) +
   scale_x_discrete(labels = c("4x4", "Forward drive", "Rear drive")) +
   scale_fill_brewer(palette = "Dark2", type = "qual") +
   labs(x = "Wheel drive", y = "Count", title = "Number of vehicles per wheel drive type
        fill = NULL) +
   theme(plot.title = element_text(hjust = 0.5),
         axis.title = element_text(face = "bold"))
```

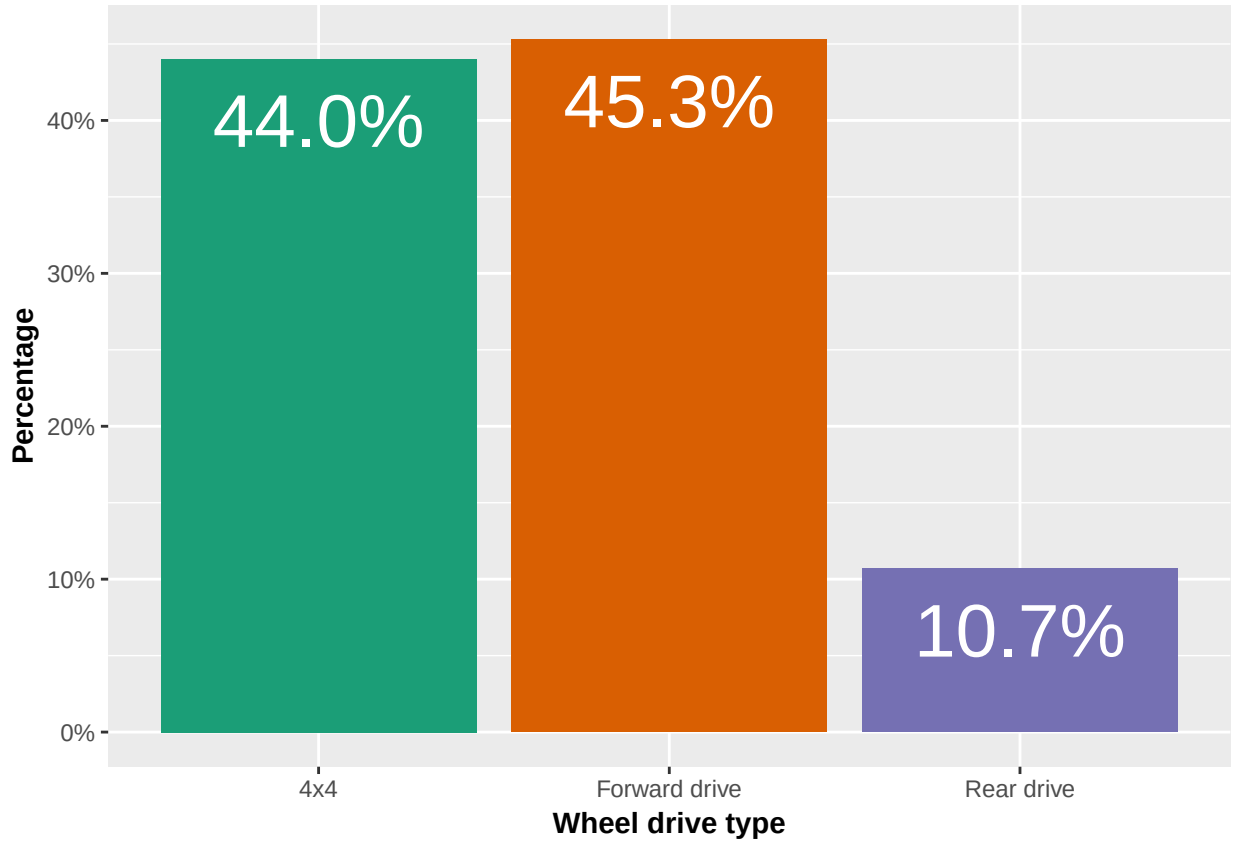


Ejercicio 7

Usando como base el gráfico anterior:

- Incluir en el eje `y` porcentaje en vez de conteos
- Usando `scale_y_continuous()` cambiar la escala del eje `y` a porcentajes
- Usando `geom_text()` incluir texto con porcentajes arriba de cada barra

```
ggplot(mpg, aes(x = drv)) +
   geom_bar(aes(y = ..prop.., fill = factor(..x..), group = "mira_que_lindos_grupos"), s
            show.legend = FALSE) +
   geom_text(aes(label = scales::percent(..prop..), y = ..prop..-0.075, group = "mira_qu
             stat= "count", vjust = -.5, size = 10, color = "White") +
   scale_x_discrete(labels = c("4x4", "Forward drive", "Rear drive")) +
   scale_fill_brewer(palette = "Dark2", type = "qual") +
   scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
   labs(y = "Percentage", x="Wheel drive type") +
   theme(axis.title = element_text(face = "bold"))
```



Muy buen trabajo. 10/10. Sin comentarios adicionales.