

# Tarea individual 2

## STAT NT

### Entrega

Esta tarea tiene que estar disponible en su repositorio de GitHub en la carpeta Tareas . Asegurate que tanto Mauro como yo seamos colaboradoras de tu proyecto privado STAT\_NT. Recordar seleccionar en en opciones de proyecto, codificación de texto 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 el archivo .Rmd con la solución de la tarea y los archivos que sean necesarios para su reproducibilidad que se evaluará.

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"`. Luego en el cuerpo del documento podés hacer comentarios extendidos sobre lo que muestra la figura.

### Ejercicio 1

#### Parte 1:

```
data(mpg)

glimpse(mpg)

## Rows: 234
## Columns: 11
## $ manufacturer <chr> "audi", "audi", "audi", "audi", "audi", "audi", "audi", "~
## $ model        <chr> "a4", "a4", "a4", "a4", "a4", "a4", "a4", "a4 quattro", "~
## $ displ        <dbl> 1.8, 1.8, 2.0, 2.0, 2.8, 2.8, 3.1, 1.8, 1.8, 2.0, 2.0, 2.~
## $ year         <int> 1999, 1999, 2008, 2008, 1999, 1999, 2008, 1999, 1999, 200~
## $ cyl          <int> 4, 4, 4, 4, 6, 6, 6, 4, 4, 4, 4, 6, 6, 6, 6, 6, 8, 8, ~
## $ trans        <chr> "auto(l5)", "manual(m5)", "manual(m6)", "auto(av)", "auto~
## $ drv          <chr> "f", "f", "f", "f", "f", "f", "f", "4", "4", "4", "4", "4~
## $ cty          <int> 18, 21, 20, 21, 16, 18, 18, 18, 16, 20, 19, 15, 17, 17, 1~
## $ hwy          <int> 29, 29, 31, 30, 26, 26, 27, 26, 25, 28, 27, 25, 25, 25, 2~
## $ fl           <chr> "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p", "p~
## $ class        <chr> "compact", "compact", "compact", "compact", "compact", "compact", "c~
```

1. Con los datos `mpg` que se encuentran disponible en `ggplot2` hacer un gráfico de barras para la variables `drv` con las siguientes características:
2. Las barras tienen que estar coloreadas por `drv`

3. Incluir el nombre de los ejes y título informativo modificando el `scale` asociado, esto es sin usar `labs`.
4. Seleccioná alguna paleta que sea apropiada para el problema. ¿Qué tipo de paleta hay que seleccionar para la variable `drv`?

Tienen que tener un color apropiado para variables cualitativas. Elegi la paleta Dark2 porque da colores bien distinguibles entre si y no dan noción de orden

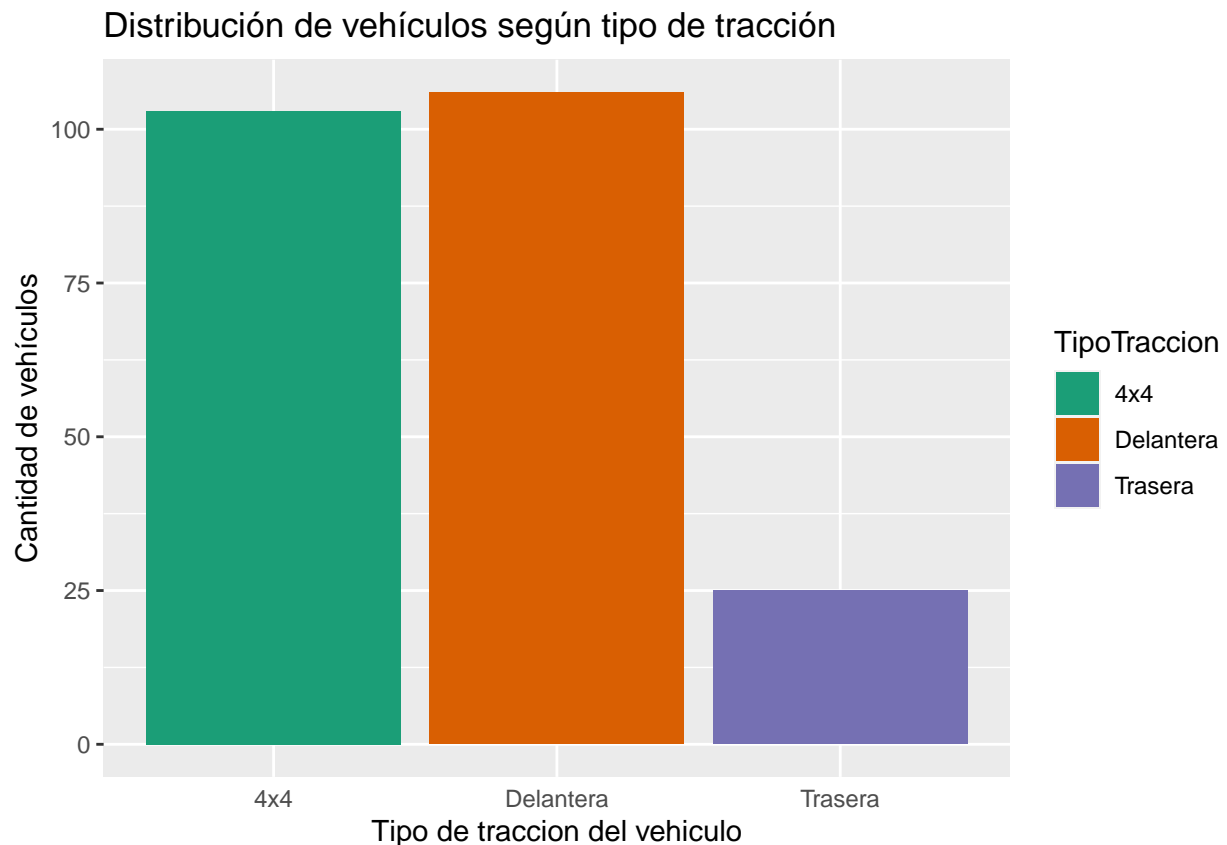
5. Recodificá la variable `drv` a 4x4, Delantera y trasera.

```
mpg <- mpg %>%
  mutate(TipoTraccion = ifelse(drv == "f", "Delantera",
                                ifelse(drv == "r", "Trasera", "4x4")))
```

6. Eliminá los ticks del eje x.

```
graficoBase <- mpg %>% ggplot(aes(TipoTraccion))

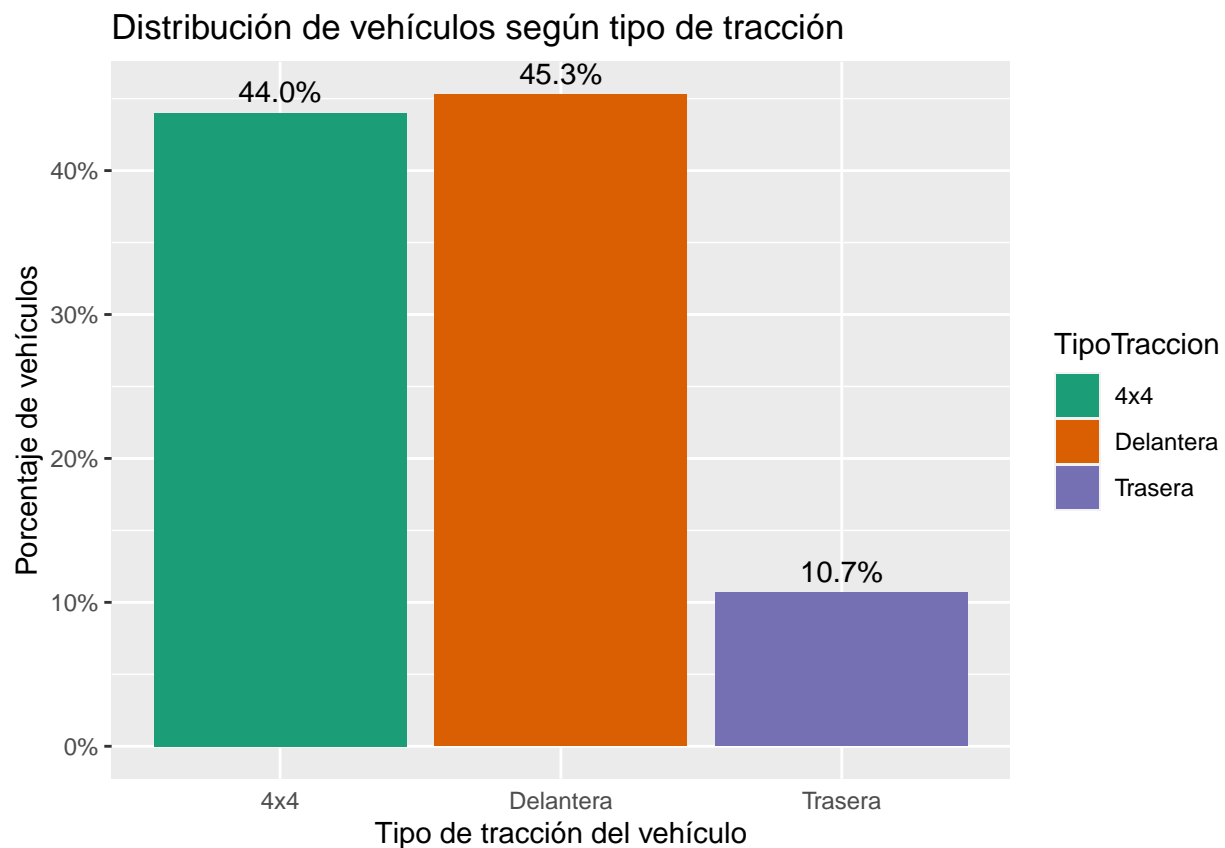
(grafico1Parte1 <- graficoBase +
  geom_bar(aes(fill= TipoTraccion)) +
  ggtitle(label = "Distribución de vehículos según tipo de tracción")+
  scale_x_discrete(name="Tipo de traccion del vehiculo") +
  scale_y_continuous(name = "Cantidad de vehículos") +
  scale_fill_brewer(palette = "Dark2") +
  theme(axis.ticks.x = element_blank())
)
```



**Parte 2:** En base al gráfico anterior

7. Hacer el gráfico en porcentaje en vez de conteos
8. Usando `scale_y_continuous()` cambiar la escala del eje y a porcentajes
9. Usando `geom_text()` incluir texto con porcentajes arriba de cada barra

```
(grafico1Parte2 <- graficoBase +  
  geom_bar(aes(y = prop.table(..count..), fill = TipoTraccion)) +  
  ggtitle(label = "Distribución de vehículos según tipo de tracción") +  
  scale_x_discrete(name = "Tipo de tracción del vehículo") +  
  scale_y_continuous(name = "Porcentaje de vehículos", labels = scales::percent) +  
  scale_fill_brewer(palette = "Dark2") +  
  theme(axis.ticks.x = element_blank()) +  
  geom_text(aes(y = prop.table(..count..), label = scales::percent(prop.table(..count..)) ), stat = "count",  
    vjust = -0.5  
  )  
)
```



## Ejercicio 2

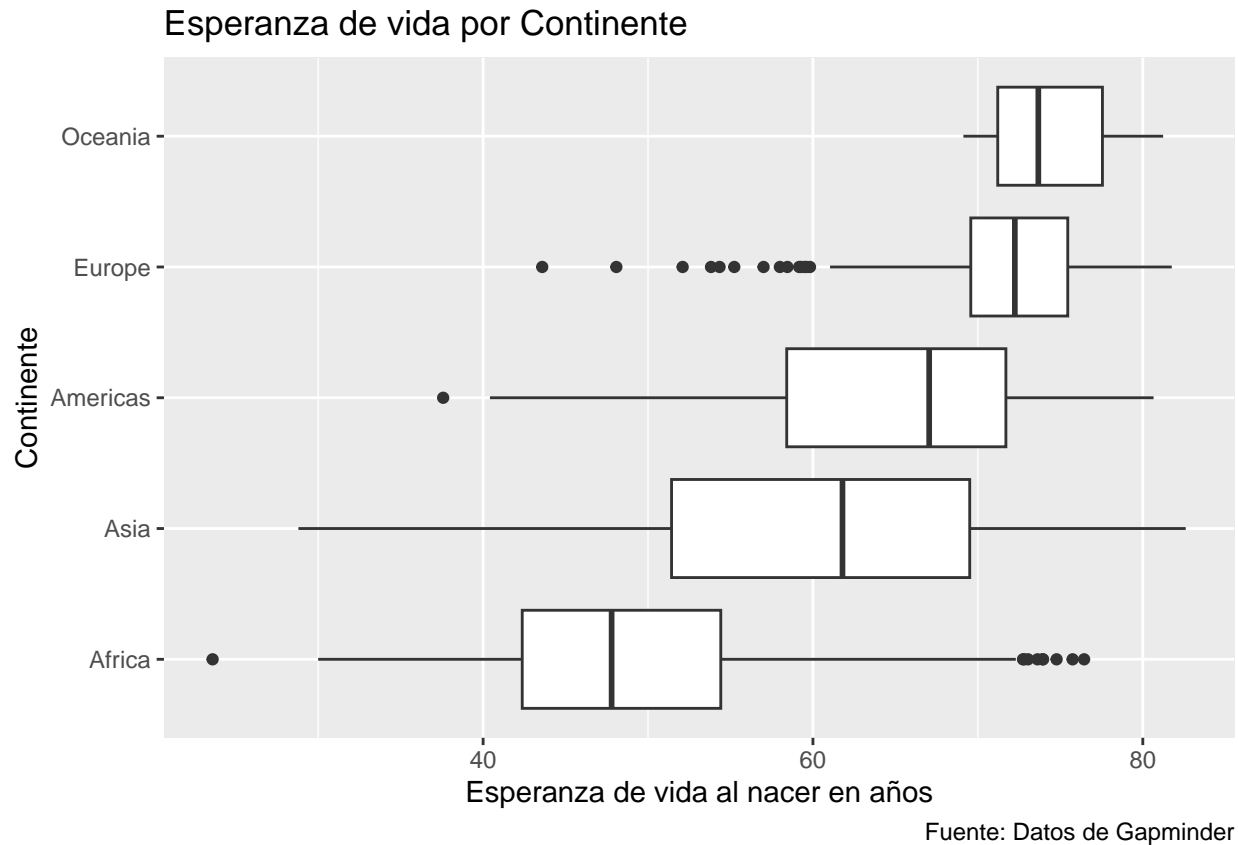
Vamos a utilizar la biblioteca `gapminder`, por lo que si no la usaste anteriormente tenés que instalarla y luego cargarla. Para obtener la descripción del paquete `library(help = "gapminder")` y para saber sobre la base `?gapminder`.

```
data(gapminder)
glimpse(gapminder)
```

```
## Rows: 1,704
## Columns: 6
## $ country   <fct> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan", ~
## $ continent <fct> Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, Asia, ~
## $ year      <int> 1952, 1957, 1962, 1967, 1972, 1977, 1982, 1987, 1992, 1997, ~
## $ lifeExp   <dbl> 28.801, 30.332, 31.997, 34.020, 36.088, 38.438, 39.854, 40.8~
## $ pop       <int> 8425333, 9240934, 10267083, 11537966, 13079460, 14880372, 12~
## $ gdpPercap <dbl> 779.4453, 820.8530, 853.1007, 836.1971, 739.9811, 786.1134, ~
```

1. Hacer un gráfico de caja de `lifeExp` según continente. Las etiquetas de los ejes deben ser claras y describir las variables involucradas (Esperanza de vida al nacer en años y Continente). Ordená las cajas usando el paquete `forcats` de menor a mayor valor en la mediana de `lifeExp`. Incluir un `caption` en la figura con algún comentario de interés que describa el gráfico. Rotar los ejes para que en el eje y quede Continente.

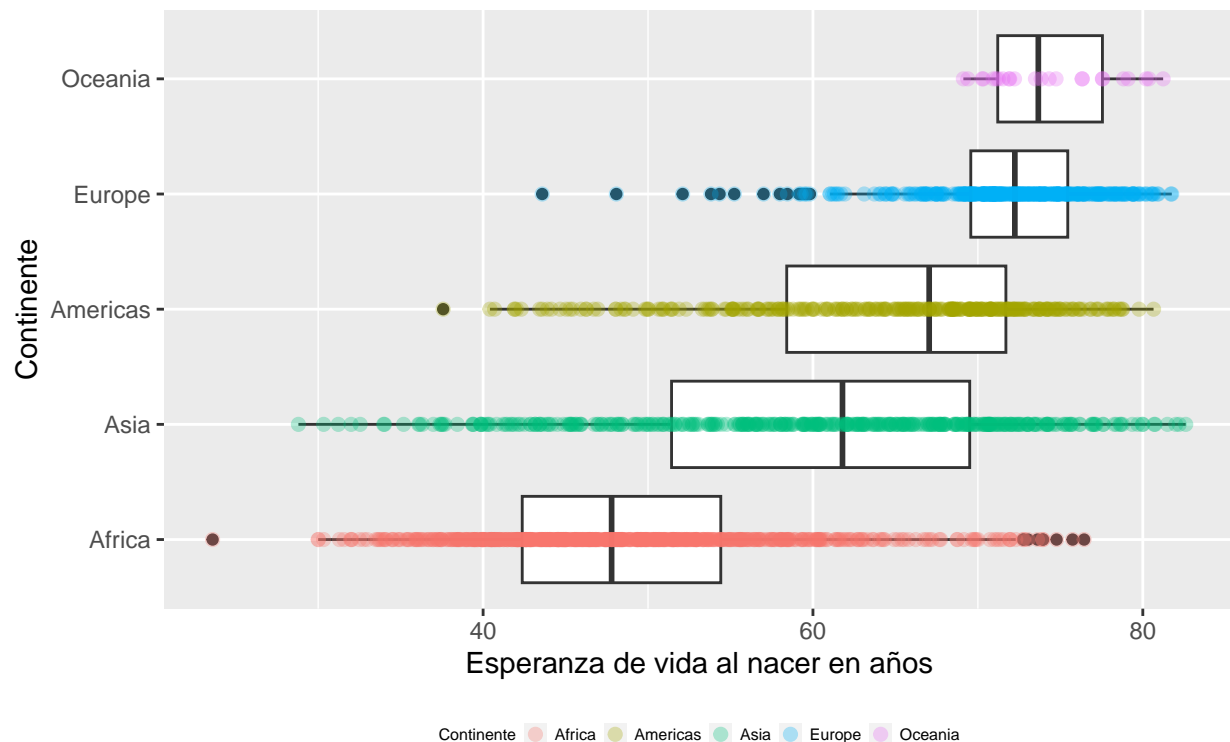
```
( Grafico2Parte1 <- gapminder %>%
  ggplot(aes(x=fct_reorder(continent, lifeExp, median), y=lifeExp,)) +
  geom_boxplot() +
  ggtitle("Esperanza de vida por Continente") +
  scale_x_discrete(name = "Continente") +
  scale_y_continuous(name = "Esperanza de vida al nacer en años") +
  labs(caption = "Fuente: Datos de Gapminder") +
  coord_flip()
)
```



- Ahora agregale al gráfico anterior una capa con las observaciones individuales con una paleta apropiada para la variable `continente` (usando `algunsacale_<>_<>`), asignando un tamaño a los puntos de 2 y transparencia 0.3. Usando `theme`, poné la leyenda abajo y modificá el tamaño del título de la leyenda y el texto de la misma a tamaño 6. Adicionalmente reducí el tamaño de la caja `key` a 0.3 cm.

```
(
  Grafico2Parte2 <- Grafico2Parte1 +
    geom_point(aes(color = continente), size = 2, alpha = 0.3) +
    scale_color_discrete(name = "Continente") +
    theme(
      legend.position = "bottom",
      legend.title = element_text(size = 6),
      legend.text = element_text(size = 6),
      legend.key.size = unit(0.3, "cm")
    )
)
```

## Esperanza de vida por Continente



Fuente: Datos de Gapminder

- Usando funciones de `dplyr` calculá la media de la variable `lifeExp` y guardala en un objeto `world_avg`. Replicá el siguiente gráfico donde la línea vertical es `world_avg` los puntos grises son el valor medio de `lifeExp` para cada continente que deberás dibujarlo usando `stat_summary` con transparencia 0.6 y tamaño 2 para los puntos. Usar el `geom` apropiado para incorporar una variación aleatoria en los puntos de 0.2.

```
(
  world_avg <- gapminder %>% summarise(world_avg = mean(lifeExp))
)
```

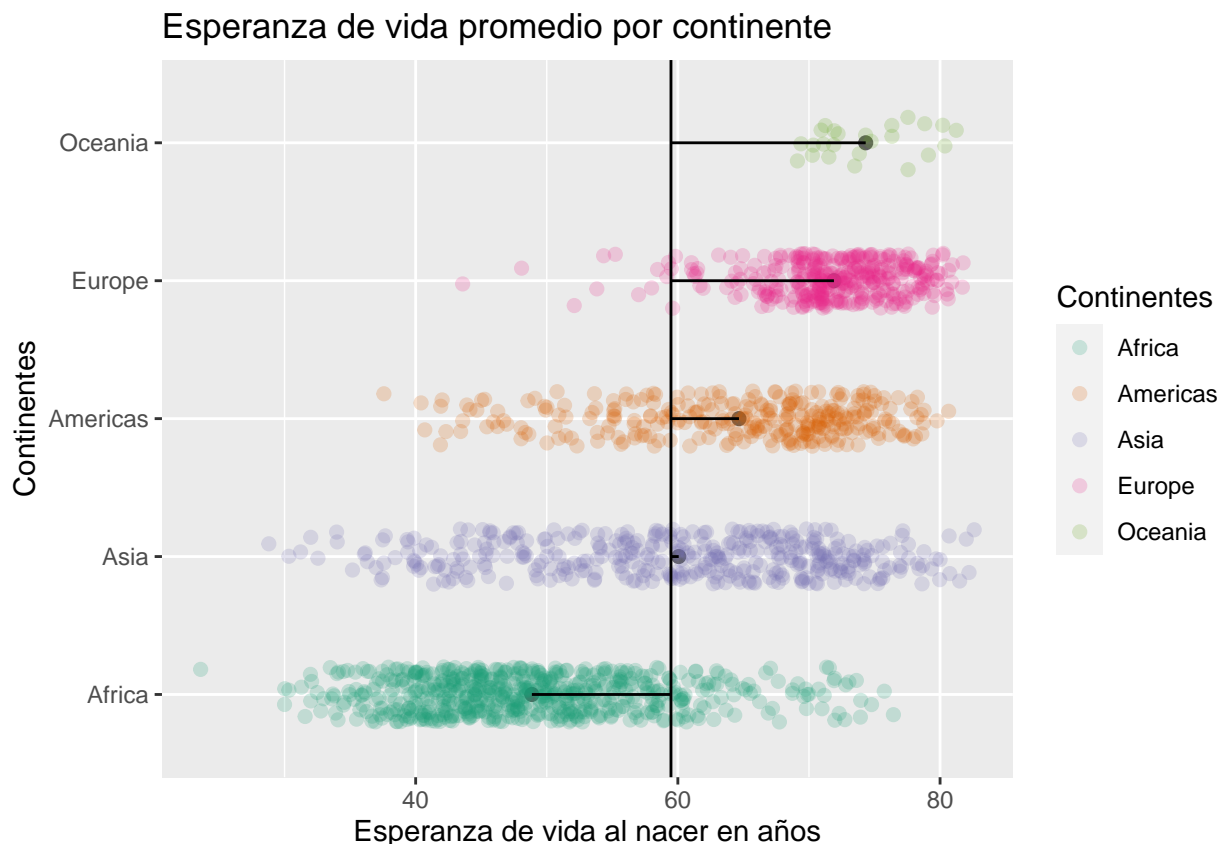
```
## # A tibble: 1 x 1
##   world_avg
##   <dbl>
## 1      59.5
```

```
(
  continent_avg <- gapminder %>% group_by(continent) %>% summarise(continent_avg = mean(lifeExp)) %>% arrange()
)
```

```
## # A tibble: 5 x 2
##   continent continent_avg
##   <fct>          <dbl>
## 1 Africa          48.9
## 2 Asia            60.1
```

```
## 3 Americas          64.7
## 4 Europe            71.9
## 5 Oceania           74.3
```

```
gapminder %>% ggplot(aes(x = lifeExp, y = fct_reorder(continent, lifeExp, median))) +
  stat_summary(fun = mean, geom = "point", size = 2, alpha = 0.6) +
  ggtitle(label = "Esperanza de vida promedio por continente") +
  scale_x_continuous(name = "Esperanza de vida al nacer en años") +
  scale_y_discrete(name = "Continentes") +
  geom_point(aes(color = continent), alpha = 0.2, size = 2, position = position_jitter(w = 0, h = 0.2)) +
  scale_color_brewer(name = "Continentes", palette = "Dark2") +
  geom_vline(aes(xintercept = world_avg), data = world_avg) +
  geom_segment(data = continent_avg, aes(x=continent_avg, y=continent, xend=world_avg$world_avg, yend=c
```



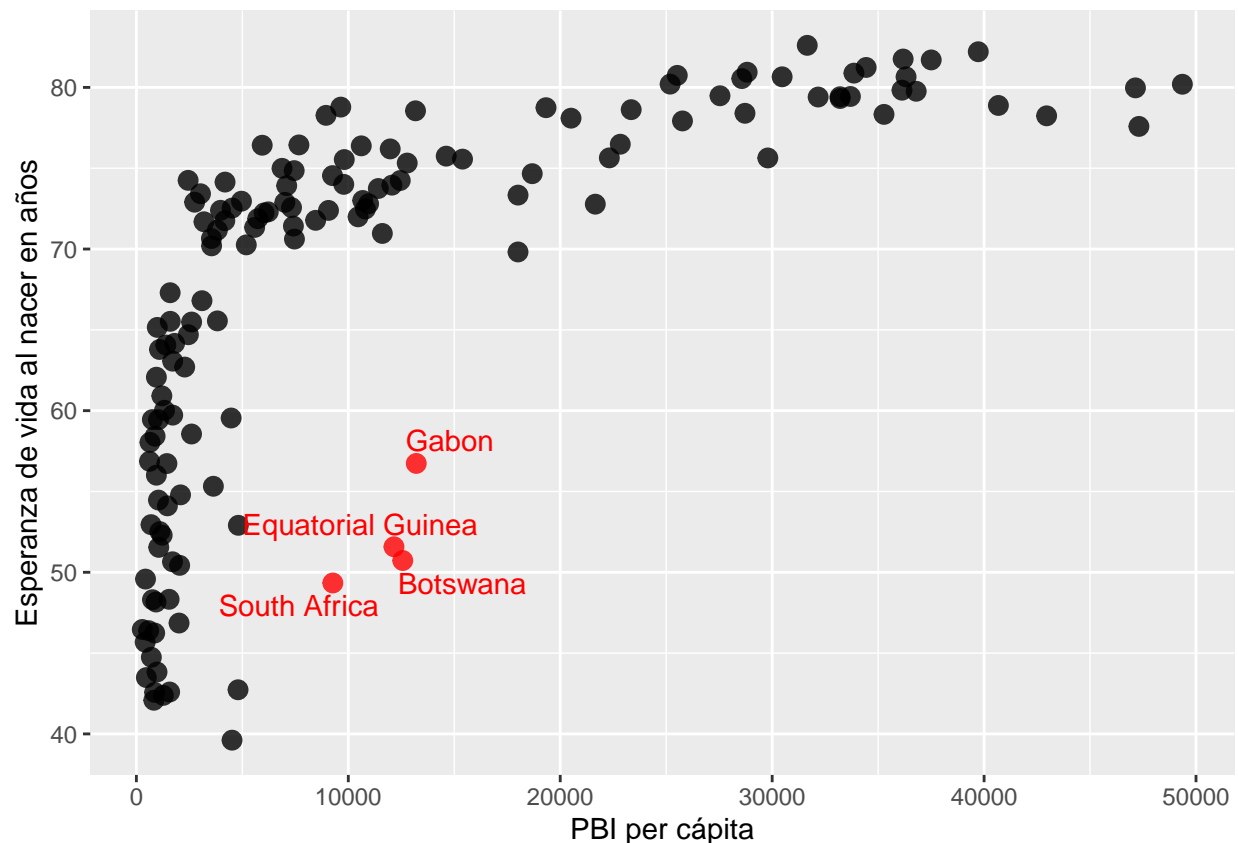
4. Agregar una capa de dato adicional para generar el lollipop plot usando `geom_segment()` para replicar el siguiente gráfico.
5. Hacer un diagrama de dispersión con `lifeExpe` en el eje y y `gdpPercap` en el eje x para el año 2007. Resaltando los países que cumplen la condición que `gdpPercap > 5000 & lifeExp < 60` en rojo. Escribir sus nombres usando `geom_text_repel` del paquete `ggrepel` para evitar texto solapado tal que quede de la siguiente manera.

```
library(ggrepel)
(
gapminder_2007 <- gapminder %>% filter(year==2007)
)
```

```
## # A tibble: 142 x 6
##   country      continent year lifeExp      pop gdpPercap
##   <fct>        <fct>    <int>  <dbl>    <int>    <dbl>
## 1 Afghanistan Asia      2007   43.8  31889923    975.
## 2 Albania      Europe    2007   76.4   3600523   5937.
## 3 Algeria      Africa    2007   72.3  33333216   6223.
## 4 Angola       Africa    2007   42.7  12420476   4797.
## 5 Argentina    Americas  2007   75.3  40301927  12779.
## 6 Australia    Oceania   2007   81.2  20434176  34435.
## 7 Austria      Europe    2007   79.8   8199783   36126.
## 8 Bahrain      Asia      2007   75.6    708573   29796.
## 9 Bangladesh   Asia      2007   64.1 150448339   1391.
## 10 Belgium     Europe    2007   79.4  10392226  33693.
## # i 132 more rows
```

```
gapminder_2007 %>% ggplot(aes(y=lifeExp, x= gdpPercap, label = country)) +
  geom_point(
    aes(color = (gdpPercap > 5000 & lifeExp < 60)),
    alpha = 0.8,
    size = 3
  ) +
  scale_color_manual(
    values = c("FALSE" = "black", "TRUE" = "red")
  ) +
  geom_text_repel(
    color = "red",
    data = subset(gapminder_2007, gdpPercap > 5000 & lifeExp < 60)
  ) +
  scale_x_continuous(name="PBI per cápita") +
  scale_y_continuous(name="Esperanza de vida al nacer en años") +
  theme(legend.position = "none")
```





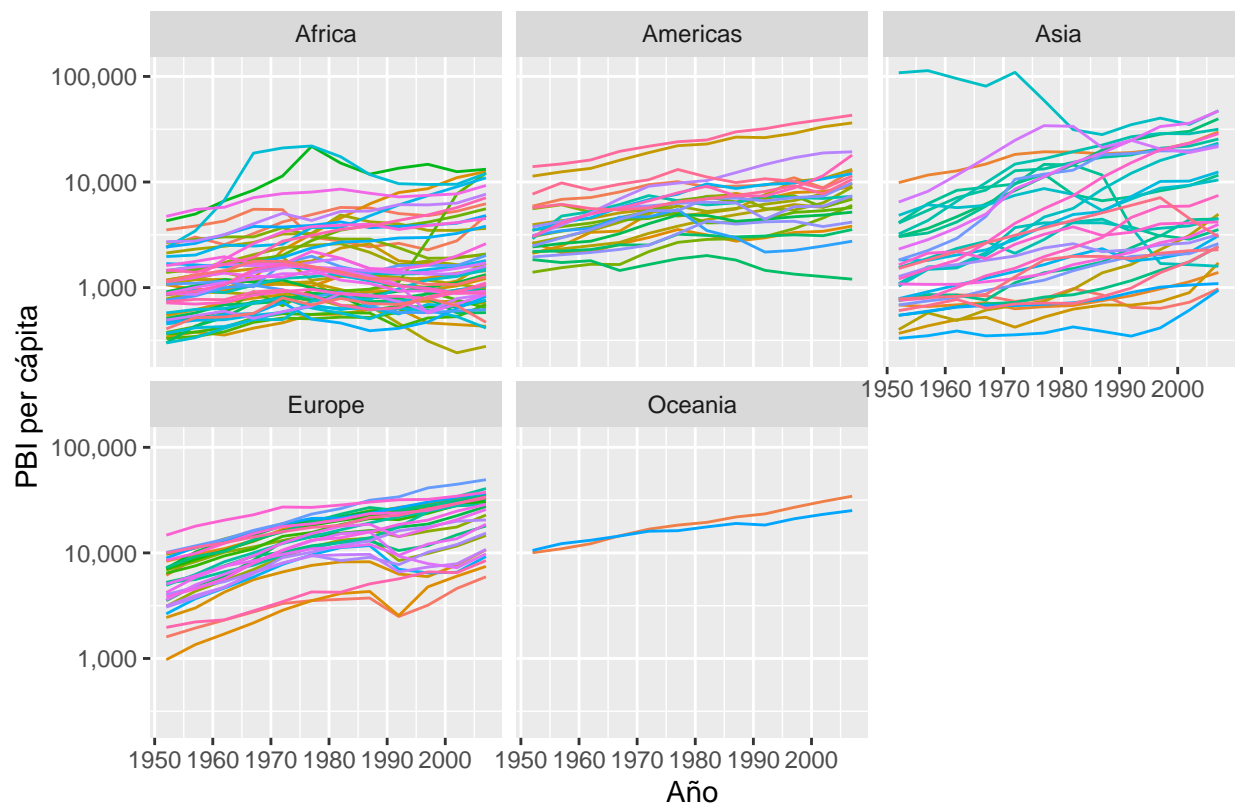
6. Hacer un gráfico de líneas que tenga en el eje x `year` y en el eje y `gdpPercap` para cada continente en una misma ventana gráfica. En cada continente, el gráfico debe contener una línea para cada país a lo largo del tiempo (serie de tiempo de `gdpPercap`). Las etiquetas de los ejes deben ser claras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

```
(
  continent_data <- gapminder %>%
    group_by(continent, country) %>%
    arrange(year)
)
```

```
## # A tibble: 1,704 x 6
## # Groups:   continent, country [142]
##   country    continent  year lifeExp    pop gdpPercap
##   <fct>      <fct>    <int> <dbl>    <int> <dbl>
## 1 Afghanistan Asia      1952  28.8  8425333  779.
## 2 Albania    Europe    1952  55.2  1282697 1601.
## 3 Algeria    Africa    1952  43.1  9279525 2449.
## 4 Angola     Africa    1952  30.0  4232095 3521.
## 5 Argentina  Americas  1952  62.5 17876956 5911.
## 6 Australia  Oceania   1952  69.1  8691212 10040.
## 7 Austria    Europe    1952  66.8  6927772 6137.
## 8 Bahrain    Asia      1952  50.9  120447 9867.
## 9 Bangladesh Asia      1952  37.5 46886859 684.
```

```
## 10 Belgium      Europe      1952      68      8730405      8343.
## # i 1,694 more rows
```

```
ggplot(continent_data, aes(x = year, y = gdpPercap, color = country)) +
  geom_line() +
  facet_wrap(~continent, ncol = 3) +
  scale_x_continuous(name = "Año") +
  scale_y_log10(
    name = "PBI per cápita",
    labels = scales::number_format(big.mark = ",")
  ) +
  labs(caption = "Fuente: Gapminder") +
  theme(legend.position = "none")
```



Fuente: Gapminder