Actividad Individual I

Ciencia de Datos con R

Introducción

El objetivo de este trabajo es aplicar herramientas de **data wrangling** con `tidyverse` y realizar **visualizaciones** utilizando `ggplot2` y `plotly`.

Dataset: `gapminder` (incluido en `{gapminder}`)

El dataset `gapminder` del paquete `{gapminder}` contiene datos de desarrollo mundial de los paises entre los años 1952 y 2007, con información de todos los continentes.

Variables principales:

- `country`: país
- `continent`: continente
- `year`: año de observación
- `lifeExp`: esperanza de vida
- `pop`: población
- `gdpPercap`: PBI per cápita

---

1. Data Wrangling

1.1. Filtrar los países del continente asiático en el año 2007 con esperanza de vida mayor a 70 y PBI per cápita mayor a 5000; y luego calcula la mediana. Interprete el resultado. La mediana guardala en un nuevo objeto llamado `mediana`.

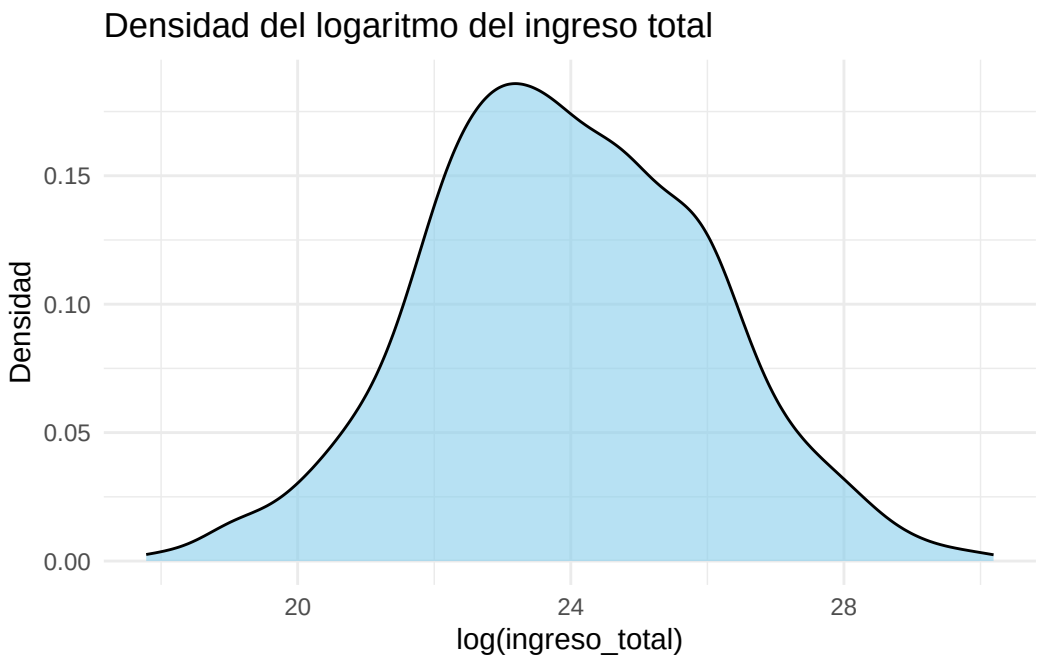
```
mediana <- gapminder %>%
  filter(continent == "Asia", year == 2007, lifeExp > 70, gdpPercap > 5000) %>%
  summarise(mediana = median(lifeExp)) %>%
  pull(mediana)

mediana
```

```
[1] 76.614
```

> La mediana de 76.6 años refleja la esperanza de vida típica entre los países asiáticos que en 2007, tenían `lifeExp > 70` y `gdpPercap > 5000`. No representa a toda Asia, sino solo a este subconjunto de paises, por lo que debe interpretarse como un valor central condicionado por esos criterios.

1.2. Crear una nueva variable que represente el ingreso total (`gdpPercap * pop`) y visualizar la densidad del logaritmo de la variable creada. Obs: todo el proceso tiene que estar unido por pipes. Interprete la visualización obtenida. Guarde el gráfico en un objeto llamado `densidad_ingreso_total_log` y luego imprimalo.

```
densidad_ingreso_total_log <- gapminder %>%
  mutate(
    ingreso_total = gdpPercap * pop,
    log_ingreso_total = log(ingreso_total)
  ) %>%
  filter(is.finite(log_ingreso_total)) %>%
  ggplot(aes(x = log_ingreso_total)) +
  geom_density(fill = "skyblue", alpha = 0.6) +
  labs(
    title = "Densidad del logaritmo del ingreso total",
    x = "log(ingreso_total)",
    y = "Densidad"
  ) +
  theme_minimal()

densidad_ingreso_total_log
```

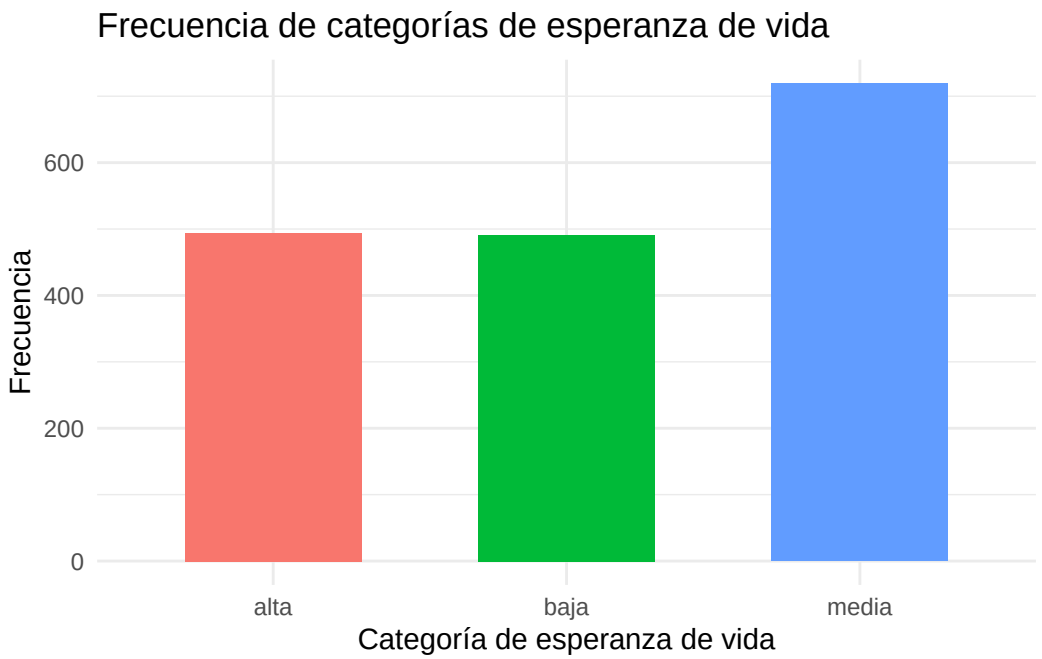


La curva de densidad del logaritmo del ingreso total muestra que la mayoría de los países están concentrados entre valores logarítmicos de 23 y 24, es decir, ingresos totales de aproximadamente 10 a 30 miles de millones. La cola hacia la derecha indica que hay unos pocos países con grandes economías y/o muy poblados que concentran ingresos más altos que el resto. La transformación logarítmica reduce la dispersión y permite observar mejor la estructura central de la distribución, que en escala lineal quedaría opacada por los valores extremos. Este tipo de transformaciones es habitual en economía y análisis de datos, especialmente cuando se trabaja con variables altamente asimétricas o con outliers.

1.3. Crear un nuevo data frame con las variables `country`, `year` y una variable categórica de esperanza de vida (baja <menos de 50 años>, media <entre 50 y menos de 70 años>, alta <70 años o más>). Luego, realizar la tabla de frecuencias de la variable categórica y la visualización de dicha tabla. La tabla de frecuencias almacenala en un objeto llamado `freq_exp_country_year` y la visualización en un objeto llamado `plot_freq_exp_country_year`

```
# Proceso de categorización y visualización:
#
# 1. case_when() categoriza lifeExp en tres niveles usando puntos de corte fijos:
#    - baja: < 50 años
#    - media: entre 50 y 70 años
#    - alta: >= 70 años
```

```r
#
# 2. count(exp_cat) genera la tabla de frecuencias requerida
#
# 3. ggplot crea gráfico de barras con:
#    - altura = frecuencia (n)
#    - color = categoría (exp_cat)
#    - estética minimalista (sin leyenda, ancho 0.6)

# Crear data frame con variable categórica de esperanza de vida
exp_cat_df <- gapminder %>%
  mutate(
    exp_cat = case_when(
      lifeExp < 50 ~ "baja",
      lifeExp >= 50 & lifeExp < 70 ~ "media",
      lifeExp >= 70 ~ "alta"
    )
  ) %>%
  select(country, year, exp_cat)

# Tabla de frecuencias
freq_exp_country_year <- exp_cat_df %>%
  count(exp_cat)

# Visualización de la tabla de frecuencias
library(ggplot2)
plot_freq_exp_country_year <- ggplot(freq_exp_country_year, aes(x = exp_cat, y = n, fill = ex
  geom_col(show.legend = FALSE, width = 0.6) +
  labs(
    title = "Frecuencia de categorías de esperanza de vida",
    x = "Categoría de esperanza de vida",
    y = "Frecuencia"
  ) +
  theme_minimal()

plot_freq_exp_country_year
```



1.4. Usar `mutate(across())` para categorizar todas las variables numéricas en tres niveles: bajo, medio y alto. Observación: el objetivo es que cada categoria tenga aproximadamente las mismas cantidad de observaciones. El resultado debe ser un nuevo data.frame con las variables originales y las categorizadas. Guardar el nuevo data.frame en un objeto llamado `gapminder_cat`. Luego, visualizar la tabla de frecuencias de cada una de las variables categorizadas.

```
# Explicación del proceso de categorización:
#
# 1. mutate(across()) permite aplicar una transformación a múltiples columnas:
#    - .cols = where(is.numeric) selecciona todas las columnas numéricas
#    - .fns define la función a aplicar (usando cut() para la categorización)
#    - .names = "cat_{.col}" genera nuevos nombres con prefijo "cat_"
#
# 2. cut(..., breaks = quantile(...)) divide los valores en tres niveles:
#    - Usa los cuantiles (33% y 66%) como puntos de corte
#    - Las categorías resultantes son "bajo", "medio" y "alto"
#
# 3. La estructura resultante tendrá:
#    - Las variables originales intactas
#    - Nuevas variables categorizadas con prefijo "cat_"

# Crear data frame con variables originales y categorizadas
```

```r
gapminder_cat <- gapminder %>%
  mutate(
    across(
      .cols = where(is.numeric),
      .fns = ~{
        breaks <- quantile(., probs = c(0, 1/3, 2/3, 1))
        cut(.,
            breaks = breaks,
            labels = c("bajo", "medio", "alto"),
            include.lowest = TRUE)
      },
      .names = "cat_{.col}"
    )
  )

# Proceso de visualización de frecuencias:
#
# 1. Identificar variables categorizadas:
#    - names(gapminder_cat) obtiene todos los nombres de columnas
#    - startsWith(..., "cat_") filtra solo las que empiezan con "cat_"
#    - Esto es necesario para separar las nuevas variables de las originales
#
# 2. Crear tablas de frecuencia usando map():
#    - map() aplica la misma función a cada variable categórica
#    - count() cuenta frecuencias por categoría
#    - mutate() agrega nombre de variable y calcula proporciones
#    - knitr::kable() formatea la tabla para mejor visualización

# Obtener nombres de variables categorizadas
var_categoricas <- names(gapminder_cat)[startsWith(names(gapminder_cat), "cat_")]

# Crear tabla de frecuencias con rangos para cada variable
walk(var_categoricas, ~{
  # Obtener nombre de la variable original (sin el prefijo cat_)
  var_original <- sub("cat_", "", .x)

  # Crear tabla de frecuencias con rangos por categoría
  gapminder_cat %>%
    group_by(.data[[.x]]) %>%
    summarise(
      categoria = first(.data[[.x]]),
      n = n(),
```

```
      valor_min = min(.data[[var_original]]),
      valor_max = max(.data[[var_original]]),
      .groups = "drop"
    ) %>%
  arrange(valor_min) %>%  # Ordenar por valor mínimo
  mutate(
    proporcion = n/sum(n),
    minimo = valor_min,
    maximo = valor_max
  ) %>%
  select(categoria, n, proporcion, minimo, maximo) %>%
  knitr::kable(
    caption = paste("Frecuencias y rangos para", var_original),
    col.names = c("Categoría", "Frecuencia", "Proporción", "Mínimo", "Máximo"),
    digits = 3
  ) %>%
  print()
})
```

Table: Frecuencias y rangos para year

Categoría	Frecuencia	Proporción	Mínimo	Máximo
bajo	568	0.333	1952	1967
medio	568	0.333	1972	1987
alto	568	0.333	1992	2007

Table: Frecuencias y rangos para lifeExp

Categoría	Frecuencia	Proporción	Mínimo	Máximo
bajo	568	0.333	23.599	52.098
medio	568	0.333	52.102	68.900
alto	568	0.333	68.930	82.603

Table: Frecuencias y rangos para pop

|Categoría | Frecuencia| Proporción| Mínimo| Máximo|

```
|:---------|----------:|----------:|--------:|----------:|
|bajo      |        568|      0.333|    60011|    3838168|
|medio     |        568|      0.333|  3840161|   11882916|
|alto      |        568|      0.333| 11911819| 1318683096|
```

Table: Frecuencias y rangos para gdpPercap

```
|Categoría | Frecuencia| Proporción|   Mínimo|     Máximo|
|:---------|----------:|----------:|--------:|----------:|
|bajo      |        568|      0.333|  241.166|   1654.457|
|medio     |        568|      0.333| 1654.989|   6619.551|
|alto      |        568|      0.333| 6631.459| 113523.133|
```

1.5. Crear una variable `cat_pop` que clasifique los países según su población (`< 1 millón`, `1-10 millones`, `> 10 millones`). Remplaza el data.frame `gapminder_cat` por el nuevo data.frame que contiene la variable `cat_pop`.

```r
# Proceso de categorización por población:
#
# 1. Actualizar gapminder_cat manteniendo sus columnas y agregando cat_pop:
#    - Usar mutate() sobre el data.frame existente
#    - case_when() con umbrales de población en millones
#    - Mantener el orden lógico: menor a mayor población

# Actualizar data frame con nueva categorización de población
gapminder_cat <- gapminder_cat %>%
  mutate(
    cat_pop = case_when(
      pop < 1e6 ~ "< 1 millón",
      pop >= 1e6 & pop <= 1e7 ~ "1-10 millones",
      pop > 1e7 ~ "> 10 millones"
    )
  )

# Visualizar distribución de categorías de población
gapminder_cat %>%
  count(cat_pop) %>%
  mutate(prop = n/sum(n)) %>%
  knitr::kable(
    caption = "Distribución de países por categoría de población",
    col.names = c("Categoría", "Frecuencia", "Proporción"),
```

```
    digits = 3
  )
```

Table 1: Distribución de países por categoría de población

Categoría	Frecuencia	Proporción
1-10 millones	865	0.508
< 1 millón	180	0.106
> 10 millones	659	0.387

1.6. Seleccionar las variables que contengan la letra **c**. Luego, guardar el resultado en un nuevo objeto llamado `gapminder_c`.

```
# Proceso de selección de variables:
#
# 1. Usar select() con matches():
#    - matches() busca patrones en nombres de columnas
#    - (?i)c hace la búsqueda insensible a mayúsculas/minúsculas
#    - Incluye tanto "c" como "C" en cualquier posición

# Seleccionar variables que contengan 'c' o 'C'
gapminder_c <- gapminder_cat %>%
  select(matches("(?i)c"))

# Mostrar nombres de variables seleccionadas
colnames(gapminder_c) %>%
  paste(collapse = ", ") %>%
  cat("Variables seleccionadas:\n", ., "\n")
```

```
Variables seleccionadas:
 country, continent, gdpPercap, cat_year, cat_lifeExp, cat_pop, cat_gdpPercap
```

1.7. Calcular el promedio y la desviación estándar de `lifeExp` y `gdpPercap` por continente y década. Guardar el resultado en un nuevo objeto llamado `summary_exp_continet_decade`

```
# Proceso de cálculo de estadísticas por grupo:
#
# 1. Crear variable década a partir de year:
#    - floor(year/10)*10 redondea hacia abajo al inicio de década
#
```

```r
# 2. Usar group_by() + summarise() para calcular estadísticas:
#    - Agrupar por continente y década
#    - Calcular media y desviación estándar para cada grupo

# Calcular estadísticas por continente y década
summary_exp_continet_decade <- gapminder %>%
  mutate(decada = floor(year/10)*10) %>%
  group_by(continent, decada) %>%
  summarise(
    lifeExp_mean = mean(lifeExp),
    lifeExp_sd = sd(lifeExp),
    gdpPercap_mean = mean(gdpPercap),
    gdpPercap_sd = sd(gdpPercap),
    .groups = "drop"
  )

# Mostrar resultados formateados
knitr::kable(
  summary_exp_continet_decade,
  caption = "Estadísticas por continente y década",
  digits = 2
)
```

Table 2: Estadísticas por continente y década

continent	decada	lifeExp_mean	lifeExp_sd	gdpPercap_mean	gdpPercap_sd
Africa	1950	40.20	5.47	1318.90	1058.37
Africa	1960	44.33	6.04	1824.22	2263.88
Africa	1970	48.52	6.67	2462.78	3723.04
Africa	1980	52.47	7.64	2382.13	2911.67
Africa	1990	53.61	9.24	2330.28	2720.96
Africa	2000	54.07	9.59	2844.21	3304.23
Americas	1950	54.62	9.19	4347.55	3140.19
Americas	1960	59.40	8.19	5284.90	3790.05
Americas	1970	63.39	7.19	6921.67	5030.81
Americas	1980	67.16	6.28	7650.07	6063.01
Americas	1990	70.36	5.04	8467.12	7407.76
Americas	2000	73.02	4.62	10145.35	9258.59
Asia	1950	47.82	9.51	5491.61	18930.73
Asia	1960	53.11	9.79	5850.27	15167.04
Asia	1970	58.46	9.87	7989.39	15752.33

continent	decada	lifeExp_mean	lifeExp_sd	gdpPercap_mean	gdpPercap_sd
Asia	1980	63.73	8.38	7521.18	8336.86
Asia	1990	67.28	8.06	9236.89	10370.12
Asia	2000	69.98	8.14	11323.56	12696.23
Europe	1950	65.56	5.92	6312.04	3441.87
Europe	1960	69.14	4.07	9254.66	4521.59
Europe	1970	71.36	3.21	13381.78	5719.36
Europe	1980	73.22	3.19	16416.10	6974.23
Europe	1990	74.97	3.18	18069.17	9571.92
Europe	2000	77.17	2.96	23383.11	11528.77
Oceania	1950	69.78	0.61	10948.30	942.84
Oceania	1960	71.20	0.25	13595.74	1109.97
Oceania	1970	72.38	0.75	16850.65	1038.07
Oceania	1980	74.80	1.07	19501.38	1773.70
Oceania	1990	77.57	1.02	22459.11	3664.85
Oceania	2000	80.23	0.87	28374.48	5136.12

1.8. Mostrar los 10 países con mayor esperanza de vida en 2007 usando `dplyr`. Calcula la media de la variable `gdpPercap` para dichos paises y comparala con la media global de `gdpPercap` para el año 2007. Comentar los resultados obtenidos. El resultado con los promedios debe ser un nuevo data.frame llamado `summary_top10_exp`

```
# Proceso de análisis:
#
# 1. Filtrar datos de 2007 y obtener top 10 por lifeExp
# 2. Calcular media de gdpPercap para:
#    - Los 10 países con mayor esperanza de vida
#    - Todos los países en 2007 (media global)

# Obtener top 10 países por esperanza de vida en 2007
top10_2007 <- gapminder %>%
  filter(year == 2007) %>%
  arrange(desc(lifeExp)) %>%
  slice_head(n = 10) %>%
  select(country, lifeExp, gdpPercap)

# Calcular medias y crear data frame resumen
summary_top10_exp <- gapminder %>%
  filter(year == 2007) %>%
  summarise(
    gdp_mean_global = mean(gdpPercap),
```

```
    gdp_mean_top10 = mean(top10_2007$gdpPercap),
    diff_porcentual = (gdp_mean_top10 - gdp_mean_global) / gdp_mean_global * 100
  )

# Mostrar top 10 países
knitr::kable(
  top10_2007,
  caption = "10 países con mayor esperanza de vida en 2007",
  col.names = c("País", "Esperanza de vida", "PIB per cápita"),
  digits = c(0, 2, 2)
)
```

Table 3: 10 países con mayor esperanza de vida en 2007

País	Esperanza de vida	PIB per cápita
Japan	82.60	31656.07
Hong Kong, China	82.21	39724.98
Iceland	81.76	36180.79
Switzerland	81.70	37506.42
Australia	81.24	34435.37
Spain	80.94	28821.06
Sweden	80.88	33859.75
Israel	80.75	25523.28
France	80.66	30470.02
Canada	80.65	36319.24

```
# Mostrar comparación de medias
knitr::kable(
  summary_top10_exp,
  caption = "Comparación de PIB per cápita: Top 10 vs Global (2007)",
  col.names = c("Media global", "Media top 10", "Diferencia %"),
  digits = 2
)
```

Table 4: Comparación de PIB per cápita: Top 10 vs Global (2007)

Media global	Media top 10	Diferencia %
11680.07	33449.7	186.38

Análisis de resultados Los 10 países con mayor esperanza de vida en 2007 tienen un PIB per cápita 186.4% superior a la media global. Esto sugiere una fuerte relación positiva entre el desarrollo económico y la longevidad de la población.

1.9. Realizar un `left_join()` entre `gapminder` y el siguiente tibble. El resultado almacenalo en un nuevo objeto llamado `gapminder_join`.

```
region_labels <- tibble(
  continent = c("Africa", "Americas", "Asia", "Europe","Oceania"),
  region = c("África", "Américas", "Asia", "Europa", "Oceanía"))
```

```
# Realizar left join entre gapminder y region_labels
gapminder_join <- gapminder %>%
  left_join(region_labels, by = "continent")

# Mostrar algunas filas del resultado
gapminder_join %>%
  group_by(region) %>%
  slice(1) %>%  # Una fila por región
  ungroup() %>%
  knitr::kable(caption = "Una fila por región para ilustrar variedad")
```

Table 5: Una fila por región para ilustrar variedad

country	continent	year	lifeExp	pop	gdpPercap	region
Argentina	Americas	1952	62.485	17876956	5911.3151	Américas
Afghanistan	Asia	1952	28.801	8425333	779.4453	Asia
Albania	Europe	1952	55.230	1282697	1601.0561	Europa
Australia	Oceania	1952	69.120	8691212	10039.5956	Oceanía
Algeria	Africa	1952	43.077	9279525	2449.0082	África

1.10. Mover la variable `pop` al inicio del dataset. El resultado almacenalo en un nuevo objeto llamado `gapminder_pop_first`.

```
gapminder_pop_first <- gapminder %>%
  select(pop, everything())

gapminder_pop_first %>%
  head() %>%
  knitr::kable(
    caption = "Primeras filas del dataset con 'pop' como primera columna",
```

```
    digits = 2
  )
```

Table 6: Primeras filas del dataset con 'pop' como primera columna

pop	country	continent	year	lifeExp	gdpPercap
8425333	Afghanistan	Asia	1952	28.80	779.45
9240934	Afghanistan	Asia	1957	30.33	820.85
10267083	Afghanistan	Asia	1962	32.00	853.10
11537966	Afghanistan	Asia	1967	34.02	836.20
13079460	Afghanistan	Asia	1972	36.09	739.98
14880372	Afghanistan	Asia	1977	38.44	786.11

1.11. Extraer una muestra aleatoria de aproximadamente el 10% observaciones y comparar el promedio de `lifeExp` por continente con el de la base completa. El resultado de la muestra guardalo en `sample10_gapminder` Explicar de forma "intuitiva" porque existen diferencias

```
# Extraer muestra aleatoria del 10%
sample10_gapminder <- gapminder %>%
  slice_sample(prop = 0.1)

# Calcular promedio de lifeExp por continente en la muestra y contar observaciones
sample_avg <- sample10_gapminder %>%
  group_by(continent) %>%
  summarise(
    lifeExp_mean_sample = mean(lifeExp),
    sample_size = n()
  )

# Calcular promedio de lifeExp por continente en la base completa y contar observaciones
complete_avg <- gapminder %>%
  group_by(continent) %>%
  summarise(
    lifeExp_mean_complete = mean(lifeExp),
    complete_size = n()
  )

# Unir los resultados para comparar
comparison <- sample_avg %>%
  left_join(complete_avg, by = "continent") %>%
  mutate(difference = lifeExp_mean_sample - lifeExp_mean_complete) %>%
```

```
  # Reorganizar columnas para mejor visualización
  select(continent, complete_size, lifeExp_mean_complete, sample_size, lifeExp_mean_sample, d

# Mostrar la comparación
comparison %>%
  knitr::kable(caption = "Comparación de esperanza de vida promedio por continente: muestra v
```

Table 7: Comparación de esperanza de vida promedio por continente: muestra vs. base completa (con tamaños de muestra)

continent	complete_size	lifeExp_mean_complete	sample_size	lifeExp_mean_sample	difference
Africa	624	48.86533	65	50.57148	1.7061468
Americas	300	64.65874	25	62.81220	-1.8465367
Asia	396	60.06490	43	60.91115	0.8462447
Europe	360	71.90369	36	70.82917	-1.0745194
Oceania	24	74.32621	1	72.22000	-2.1062083

Las diferencias entre la muestra (10%) y la base completa en esperanza de vida por continente se deben al tamaño reducido de la muestra, la aleatoriedad del muestreo, posibles sesgos o valores atípicos, y la baja representación de algunos continentes, lo que aumenta la variabilidad y aleja los promedios muestrales de los verdaderos.

1.12. Filtrar por el año 2007, crear una variable categórica de `gdpPercap` (al igual que lo realizado en el punto 1.4), agrupar según `continent` y categoría, y calcular el promedio de `lifeExp`. Guardar el resultado en un nuevo objeto llamado `summary_gdpPercap_lifeExp`.

```
# Filtrar por el año 2007
# Crear variable categórica de gdpPercap (bajo, medio, alto) usando ntile
# Agrupar por continent y categoría de gdpPercap
# Calcular promedio de lifeExp

summary_gdpPercap_lifeExp <- gapminder %>%
  filter(year == 2007) %>%
  mutate(gdpPercap_cat = cut(
    gdpPercap,
    breaks = quantile(gdpPercap, probs = c(0, 1/3, 2/3, 1), na.rm = TRUE),
    labels = c("bajo", "medio", "alto"),
    include.lowest = TRUE
```

```
  )) %>%
  group_by(continent, gdpPercap_cat) %>%
  summarise(lifeExp_mean = mean(lifeExp), .groups = "drop")

# Mostrar el resultado
summary_gdpPercap_lifeExp %>%
  knitr::kable(caption = "Promedio de esperanza de vida por continente y categoría de PIB pe
```

Table 8: Promedio de esperanza de vida por continente y categoría de PIB per cápita (2007)

continent	gdpPercap_cat	lifeExp_mean
Africa	bajo	52.40649
Africa	medio	61.62545
Africa	alto	58.24850
Americas	bajo	60.91600
Americas	medio	73.04465
Americas	alto	76.78971
Asia	bajo	62.78920
Asia	medio	70.64300
Asia	alto	78.03927
Europe	medio	73.86829
Europe	alto	78.79913
Oceania	alto	80.71950

1.13. Estandarizar las variables `lifeExp`, `gdpPercap` y `pop` y luego calcular sus medias y desvíos estándar. Guardar el resultado en un nuevo objeto llamado `summary_estandarizado`.

```
# Estandarizar variables y calcular estadísticas
summary_estandarizado <- gapminder %>%
  # Estandarizar las variables seleccionadas
  mutate(
    across(
      c(lifeExp, gdpPercap, pop),
      ~scale(.x),
      .names = "z_{.col}"
    )
  ) %>%
  # Calcular media y desvío estándar de las variables estandarizadas
  summarise(
    across(
```

```
      starts_with("z_"),
      list(
        media = ~mean(.x, na.rm = TRUE),
        desvio = ~sd(.x, na.rm = TRUE)
      ),
      .names = "{.fn}_{.col}"
    )
  ) %>%
  # Pasar a formato long
  pivot_longer(
    cols = everything(),
    names_to = c("estadistico", "variable"),
    names_sep = "_z_",
    values_to = "valor"
  ) %>%
  pivot_wider(
    names_from = estadistico,
    values_from = valor
  )

# Mostrar resultados
summary_estandarizado %>%
  knitr::kable(
    caption = "Estadísticas de las variables estandarizadas",
    digits = 4
  )
```

Table 9: Estadísticas de las variables estandarizadas

variable	media	desvio
lifeExp	0	1
gdpPercap	0	1
pop	0	1

1.14. Calcular el cambio en esperanza de vida entre años (utilizar la función `lag`) para cada país y agrupar según si aumentó o no. Comparar el `gdpPercap` promedio en 2007 según esa clasificación. Guardar el resultado en un nuevo objeto llamado `summary_lifeExp_gdpPercap`.

```
# Calcular cambio en esperanza de vida por país y comparar gdpPercap en 2007
summary_lifeExp_gdpPercap <- gapminder %>%
  group_by(country) %>%
```

```r
  arrange(year) %>%
  mutate(lifeExp_diff = lifeExp - lag(lifeExp)) %>%
  filter(year == 2007) %>%
  mutate(change = if_else(lifeExp_diff > 0, "Aumentó", "No aumentó")) %>%
  group_by(change) %>%
  summarise(avg_gdpPercap_2007 = mean(gdpPercap), .groups = "drop")

# Mostrar comparación
summary_lifeExp_gdpPercap %>%
  knitr::kable(caption = "Promedio de gdpPercap en 2007 según cambio en esperanza de vida")
```

Table 10: Promedio de gdpPercap en 2007 según cambio en esperanza de vida

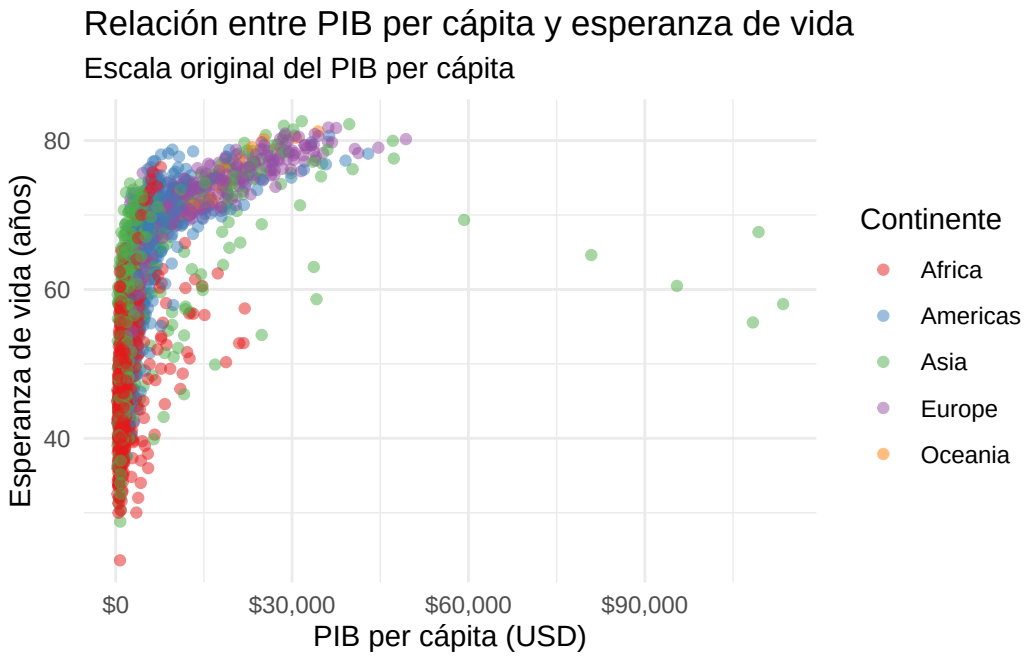
change	avg_gdpPercap_2007
Aumentó	11891.880
No aumentó	5876.528

2. Visualizaciones

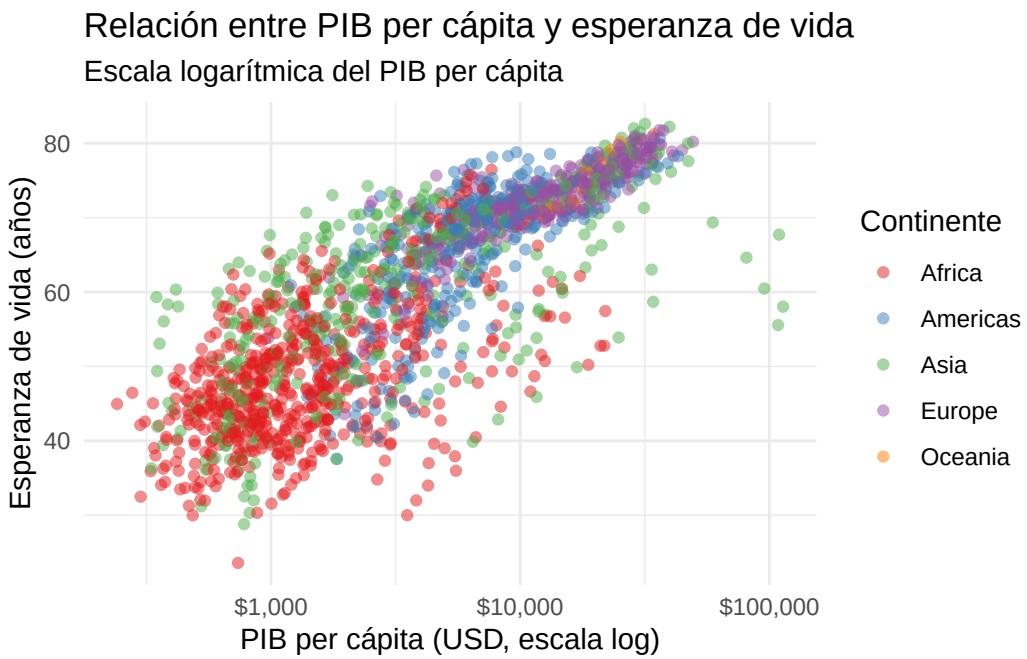
2.1. Gráfico de dispersión entre `gdpPercap` y `lifeExp`, coloreado por `continent`. Interpretar los resultados obtenidos.

```r
# Paleta para continentes con colores consistentes
paleta_continentes <- scale_color_brewer(palette = "Set1")
paleta_fill_continentes <- scale_fill_brewer(palette = "Set1")


# Gráfico con escala original
p1 <- ggplot(gapminder, aes(x = gdpPercap, y = lifeExp, color = continent)) +
  geom_point(alpha = 0.5) +
  scale_x_continuous(labels = scales::dollar_format()) +
  labs(
    title = "Relación entre PIB per cápita y esperanza de vida",
    subtitle = "Escala original del PIB per cápita",
    x = "PIB per cápita (USD)",
    y = "Esperanza de vida (años)",
    color = "Continente"
  ) +
  theme_minimal() +
  paleta_continentes
```

```r
# Gráfico con escala logarítmica
p2 <- ggplot(gapminder, aes(x = gdpPercap, y = lifeExp, color = continent)) +
  geom_point(alpha = 0.5) +
  scale_x_log10(labels = scales::dollar_format()) +
  labs(
    title = "Relación entre PIB per cápita y esperanza de vida",
    subtitle = "Escala logarítmica del PIB per cápita",
    x = "PIB per cápita (USD, escala log)",
    y = "Esperanza de vida (años)",
    color = "Continente"
  ) +
  theme_minimal() +
  paleta_continentes

# Mostrar ambos gráficos
p1
```

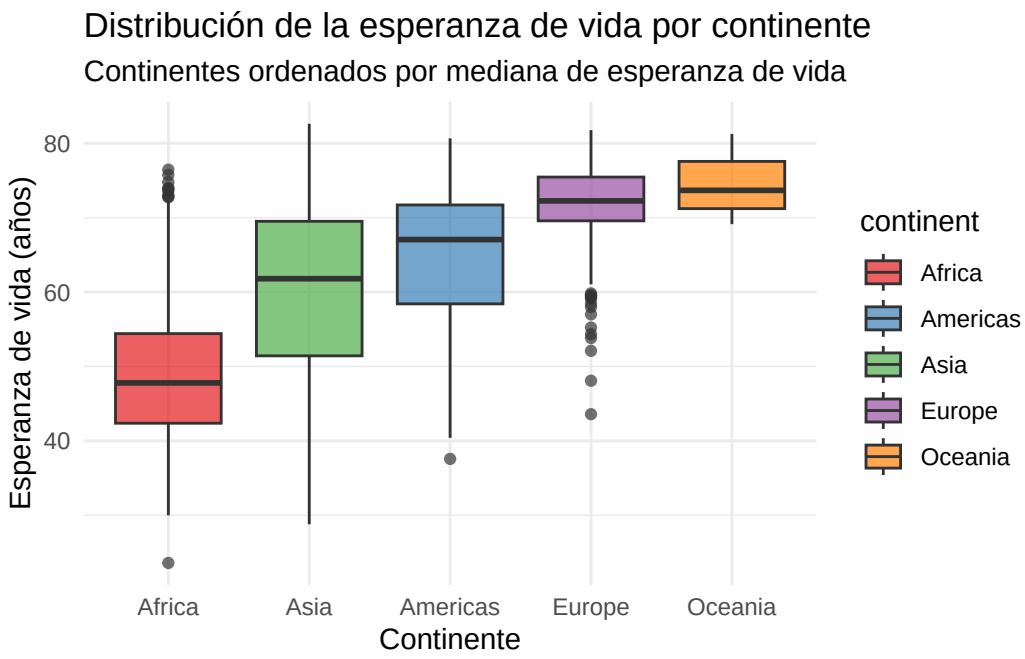



```r
p2
```



2.2. Boxplot de `lifeExp` por continente. Interpretar los resultados obtenidos

```
# Calcular medianas por continente para ordenar
continents_ordered <- gapminder %>%
  group_by(continent) %>%
  summarise(median_lifeExp = median(lifeExp)) %>%
  arrange(median_lifeExp) %>%
  pull(continent)

# Crear boxplot con continentes ordenados
ggplot(gapminder, aes(x = factor(continent, levels = continents_ordered), y = lifeExp, fill =
  geom_boxplot(alpha = 0.7) +
  labs(
    title = "Distribución de la esperanza de vida por continente",
    subtitle = "Continentes ordenados por mediana de esperanza de vida",
    x = "Continente",
    y = "Esperanza de vida (años)"
  ) +
  theme_minimal() +
  paleta_fill_continentes
```



2.3. Facet wrap de `gdpPercap` según año.

En vez de tener que elegir un solo tipo de gráfico, probé cuatro enfoques que iluminan la misma historia desde ángulos distintos:

1. **Violines facetados (escala log)** *Cumple la consigna palabra por palabra* y atenúa el efecto de los outliers.

2. **Histogramas facetados (escala log)** Mismo paneo temporal, pero sin separar por continente. Ideal para ver el desplazamiento global de la masa hacia la derecha.
3. **Áreas apiladas** Muestra cómo se reparte el *PIB mundial* entre continentes.

   Tip: el área se calcula con `ingreso_total = gdpPercap * pop`, así cada país aporta su parte según población×PIB per cápita.

4. **Serie temporal de la mediana** Una línea por continente con la mediana anual de `gdpPercap` (escala original para lectura rápida).

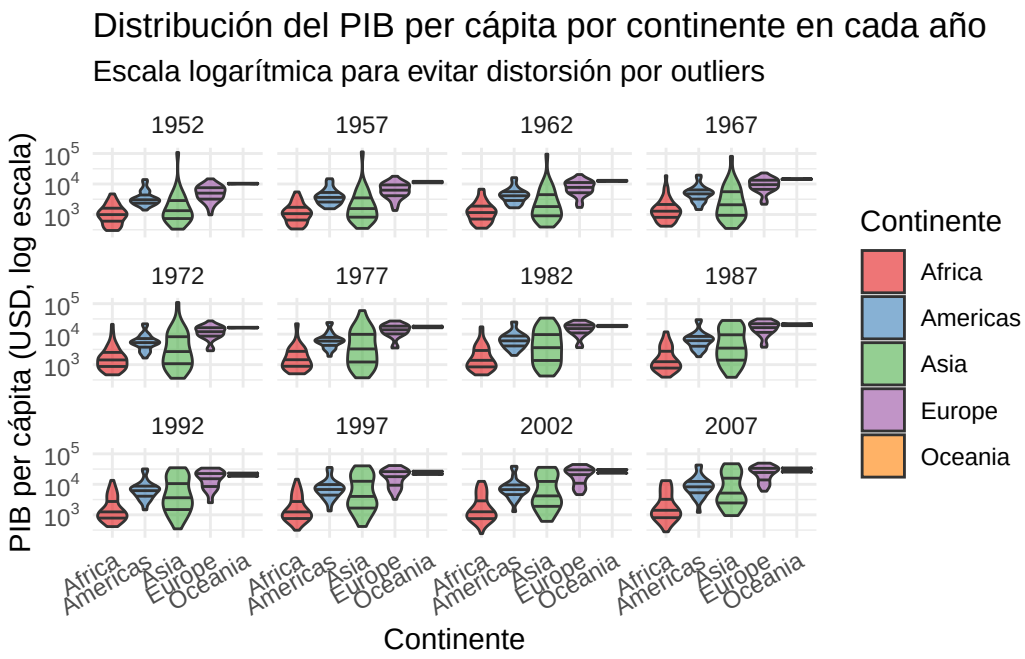
```
# ---- 1) Violines facetados por año (log) ----
g_violin <- gapminder %>%
  ggplot(aes(x = continent, y = gdpPercap, fill = continent)) +
  geom_violin(scale = "width", draw_quantiles = c(0.25, 0.5, 0.75), alpha = 0.6) +
  scale_y_log10(labels = scales::label_log()) +
  facet_wrap(~year, ncol = 4) +
```

```
  labs(
    title = "Distribución del PIB per cápita por continente en cada año",
    subtitle = "Escala logarítmica para evitar distorsión por outliers",
    x = "Continente",
    y = "PIB per cápita (USD, log escala)",
    fill = "Continente"
  ) +
  theme_minimal() +
  paleta_fill_continentes +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))

# ---- 2) Histogramas facetados por año (log) ----
g_hist <- gapminder %>%
  ggplot(aes(x = log10(gdpPercap))) +
  geom_histogram(bins = 25, fill = "steelblue", color = "white", alpha = 0.8) +
  scale_x_log10(labels = scales::label_log()) +
  facet_wrap(~ year, ncol = 4) +
  labs(
    title = "Distribución del PIB per cápita (escala log) por año",
    subtitle = "Distribución global sin separar continentes",
    x = "PIB per cápita (escala log)",
    y = "Frecuencia"
  ) +
  theme_minimal()

# ---- 3) Área apilada: proporción del PIB mundial por continente ----
g_area <- gapminder %>%
  mutate(ingreso_total = gdpPercap * pop) %>%
  group_by(year, continent) %>%
  summarise(ingreso = sum(ingreso_total), .groups = "drop") %>%
  group_by(year) %>%
  mutate(prop = ingreso / sum(ingreso)) %>%
  ggplot(aes(x = year, y = prop, fill = continent)) +
  geom_area(alpha = 0.8) +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(
    title = "Proporción del PIB mundial aportado por continente",
    x = "Año",
    y = "Participación (%)",
    fill = "Continente"
  ) +
  theme_minimal() +
```

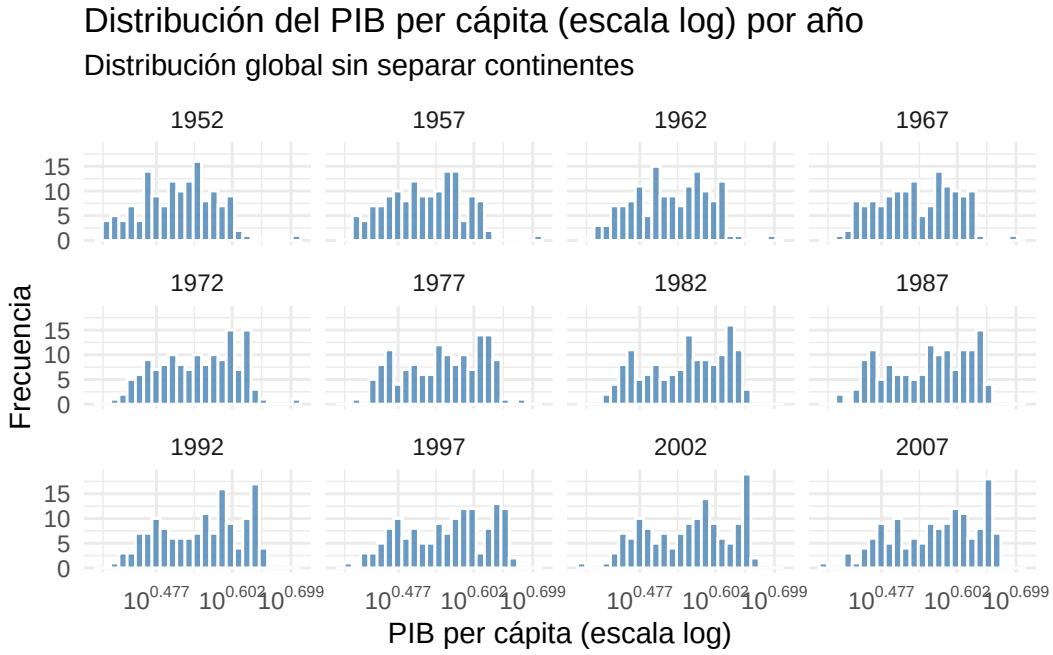
```
  paleta_fill_continentes

# ---- 4) Línea de la mediana de gdpPercap por continente ----
g_median <- gapminder %>%
  group_by(year, continent) %>%
  summarise(median_gdp = median(gdpPercap), .groups = "drop") %>%
  ggplot(aes(x = year, y = median_gdp, color = continent)) +
  geom_line(size = 1) +
  geom_point(size = 1.5) +
  scale_y_continuous(labels = scales::label_number(scale_cut = scales::cut_short_scale())) +
  labs(
    title = "Evolución de la mediana del PIB per cápita por continente",
    x = "Año",
    y = "Mediana del PIB per cápita (USD)",
    color = "Continente"
  ) +
  theme_minimal() +
  paleta_continentes
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

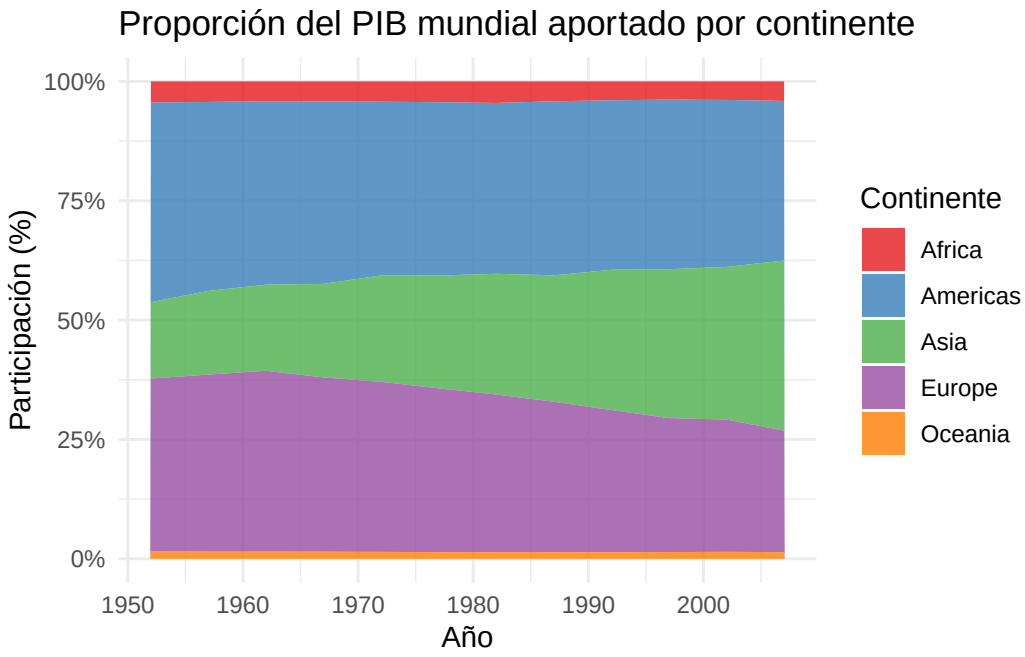
```
# ---- Mostrar gráficos ----
g_violin
```


Distribución del PIB per cápita por continente en cada año
Escala logarítmica para evitar distorsión por outliers
PIB per cápita (USD, log escala)
1952
1957
1962
1967
1972
1977
1982
1987
1992
1997
2002
2007
Africa
Americas
Asia
Europe
Oceania
Continente

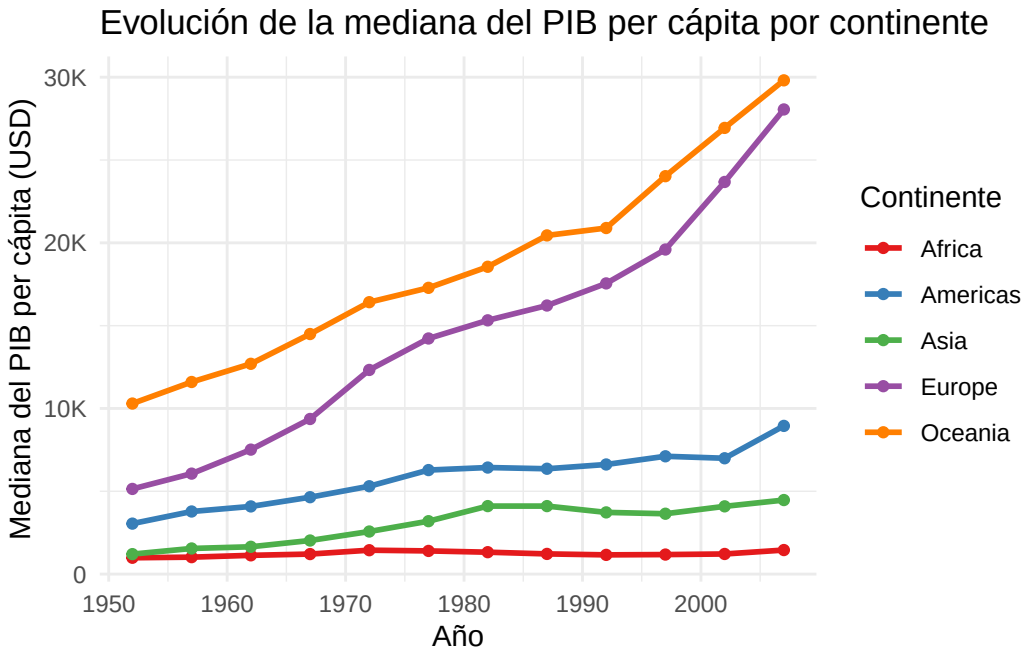
```
g_hist
```


Distribución del PIB per cápita (escala log) por año
Distribución global sin separar continentes
Frecuencia
PIB per cápita (escala log)

```
g_area
```




```
g_median
```



Interpretación

- Los **violines** confirman el ranking histórico: Europa y Oceanía siempre arriba. Asia arranca abajo pero muestra cierto despegue, sobre todo en la segunda y tercera fila (1972-2007). África, en cambio, parece casi estancada: su forma apenas cambia y sigue muy por debajo del resto.

- El **histograma log** deja ver el corrimiento sistemático a la derecha la "clase media" mundial es cada vez más rica, aunque sigue habiendo gran dispersión.

- El **área apilada** revela el salto de Asia que se queda con una porción cada vez más grande del PIB mundial, aún con una mediana modesta. Esto refleja que gran parte del impulso económico viene de sus países más poblados. Europa pierde participación relativa, aunque en realidad su PIB p/c mediano sube (tiene bajo crecimiento poblacional), así que pesa menos en el agregado mundial. Oceanía sigue el mismo patrón: ingresos altos pero población marginal proporción global bajísima.

- La **línea de medianas** muestra la carrera: Europa y Oceanía llevan el liderazgo en crecimiento, Asia avanza pero de forma más lenta, lo que cuadra con los violines: mejora progresiva sin despegar del todo. África casi no mejora y sigue rezagada. América tiene una trayectoria intermedia, más estable y sin grandes sobresaltos.

2.4. Versión interactiva con `ggplotly()` del diagrama de dispersión realizado en el punto 2.1.

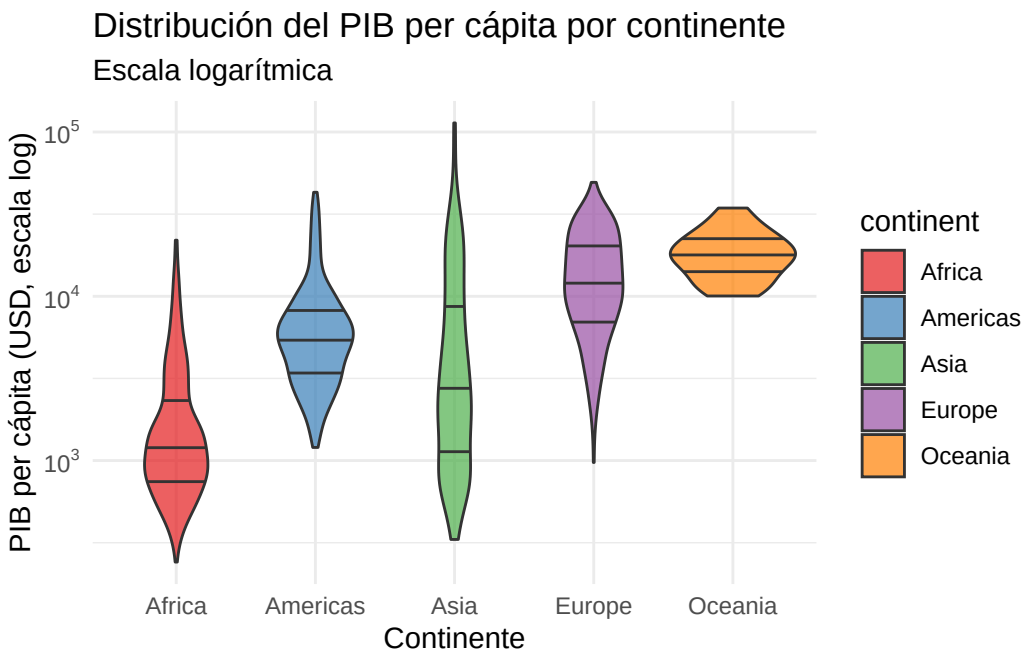
```
grafico_interactivo <- ggplot(gapminder, aes(x = gdpPercap, y = lifeExp, color = continent, t
  geom_point(alpha = 0.5) +
  scale_x_log10(labels = scales::dollar_format()) +
  labs(
    title = "Relación interactiva entre PIB per cápita y esperanza de vida",
    subtitle = "Escala logarítmica del PIB per cápita",
    x = "PIB per cápita (USD, escala log)",
    y = "Esperanza de vida (años)",
    color = "Continente"
  ) +
  theme_minimal() +
  paleta_continentes

ggplotly(grafico_interactivo, tooltip = "text")
```

file:////private/var/folders/fp/fmmgkb5548nf620fw64d0tl40000gn/T/Rtmpdu2DyI/file112f51ddddbc/

Relación interactiva entre PIB per cápita y esperanza de vida

2.5. Violin plot de `gdpPercap` por continente.

```
# - Violin plot en escala logarítmica -
gapminder %>%
  ggplot(aes(x = continent, y = gdpPercap, fill = continent)) +
  geom_violin(alpha = 0.7, draw_quantiles = c(0.25, 0.5, 0.75)) +
  scale_y_log10(labels = scales::label_log()) +
  labs(
    title    = "Distribución del PIB per cápita por continente",
    subtitle = "Escala logarítmica",
    x        = "Continente",
    y        = "PIB per cápita (USD, escala log)"
  ) +
  theme_minimal() +
  paleta_fill_continentes
```



2.6. Gráfico de mosaico de `continent` contra una categoría derivada de `lifeExp` (por ejemplo: baja, media, alta).

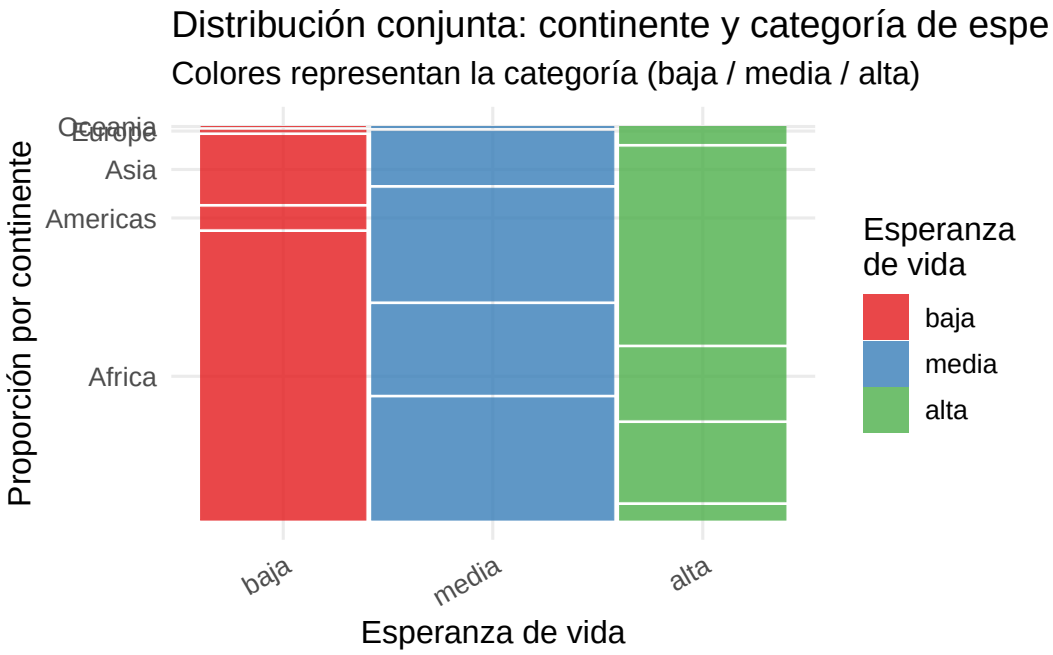
```
data_mosaic <- gapminder %>%
  mutate(
    lifeExp_cat = case_when(
      lifeExp < 50 ~ "baja",
```

```
      lifeExp < 70 ~ "media",
      TRUE         ~ "alta"
    ),
    lifeExp_cat = factor(lifeExp_cat, levels = c("baja", "media", "alta"))
  )

ggplot(data_mosaic) +
  geom_mosaic(
    aes(weight = 1, x = product(continent, lifeExp_cat), fill = lifeExp_cat),
    na.rm = TRUE
  ) +
  labs(
    title = "Distribución conjunta: continente y categoría de esperanza de vida",
    subtitle = "Colores representan la categoría (baja / media / alta)",
    x     = "Esperanza de vida",
    y     = "Proporción por continente",
    fill  = "Esperanza\nde vida"
  ) +
  theme_minimal(base_size = 12) +
  paleta_fill_continentes +
  theme(
    axis.text.x = element_text(angle = 30, hjust = 1)
  )
```

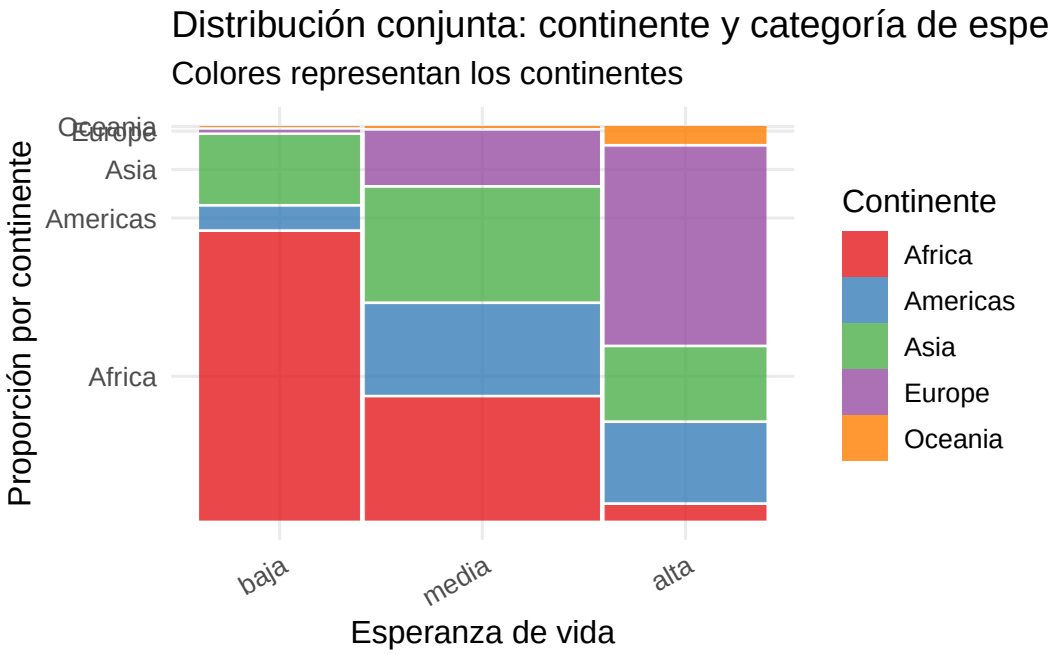
```
Warning: The `scale_name` argument of `continuous_scale()` is deprecated as of ggplot2
3.5.0.
```

```
Warning: The `trans` argument of `continuous_scale()` is deprecated as of ggplot2 3.5.0.
i Please use the `transform` argument instead.
```

```
Warning: `unite_()` was deprecated in tidyr 1.2.0.
i Please use `unite()` instead.
i The deprecated feature was likely used in the ggmosaic package.
  Please report the issue at <https://github.com/haleyjeppson/ggmosaic>.
```




```
ggplot(data_mosaic) +
  geom_mosaic(
    aes(weight = 1, x = product(continent, lifeExp_cat), fill = continent),
    na.rm = TRUE
  ) +
  labs(
    title = "Distribución conjunta: continente y categoría de esperanza de vida",
    subtitle = "Colores representan los continentes",
    x     = "Esperanza de vida",
    y     = "Proporción por continente",
    fill  = "Continente"
  ) +
  theme_minimal(base_size = 12) +
  paleta_fill_continentes +
  theme(
    axis.text.x = element_text(angle = 30, hjust = 1)
  )
```



Entrega

- **La fecha limite de entrega es el 30 de abril de 2025.**
- Las respuestas deben estar en este mismo archivo `.qmd`, el contenido deber ser completamente reproducible, es decir, cada chunk debe de funcionar sin errores para poder replicar los resultados.
- No se aceptan archivos `.Rmd` o `.R` para la entrega. Solamente subir al repositorio el archivo `.qmd` con las respuestas.
- Cada respuesta del ejercicio debe estar en el chunk correspondiente, no borrar la etiqueta del chunk `#| label: ejercicio_XX`.
- Puede realizar pasos intermedios los que sean necesarios dentro del chunk pero debe de respetar el nombre del objeto final en el caso que se indique.
- Los gráficos deben ser guardados en objetos y luego impresos en el caso que se indique que lo almacenen en un objeto. En el caso que no se indique, pueden ser impresos directamente.

- Para comenzar la tarea deben de ir al siguiente link: GitHub Classroom. Una vez allí les va a pedir que indiquen su cuenta de GitHub y luego les va a crear un repositorio en su cuenta. Una vez creado el repositorio, deben de clonar el repositorio en su computadora y abrirlo con RStudio.
