

# Exploración Auxiliar de gapminder

## Análisis complementario para interpretación de resultados

### Introducción

Este documento auxiliar permite realizar exploraciones adicionales sobre el dataset `gapminder`, facilitando la interpretación de resultados y la generación de visualizaciones complementarias.

### Carga de librerías y datos

---

#### 1. Países que cumplen condiciones específicas

Por ejemplo, para la pregunta 1 de la tarea principal:

Filtrar los países asiáticos en 2007 con esperanza de vida mayor a 70 y PBI per cápita mayor a 5000:

```
# Filtrar países según condición
países_filtrados <- gapminder %>%
  filter(continent == "Asia", year == 2007, lifeExp > 70, gdpPercap > 5000)

países_filtrados %>% select(country, lifeExp, gdpPercap)
```

```
# A tibble: 14 x 3
  country      lifeExp gdpPercap
  <fct>      <dbl>    <dbl>
1 Bahrain      75.6    29796.
2 Hong Kong, China 82.2    39725.
3 Iran         71.0    11606.
```

4	Israel	80.7	25523.
5	Japan	82.6	31656.
6	Korea, Rep.	78.6	23348.
7	Kuwait	77.6	47307.
8	Lebanon	72.0	10461.
9	Malaysia	74.2	12452.
10	Oman	75.6	22316.
11	Saudi Arabia	72.8	21655.
12	Singapore	80.0	47143.
13	Taiwan	78.4	28718.
14	Thailand	70.6	7458.

### Visualización: Scatterplot de Esperanza de Vida vs PBI per cápita

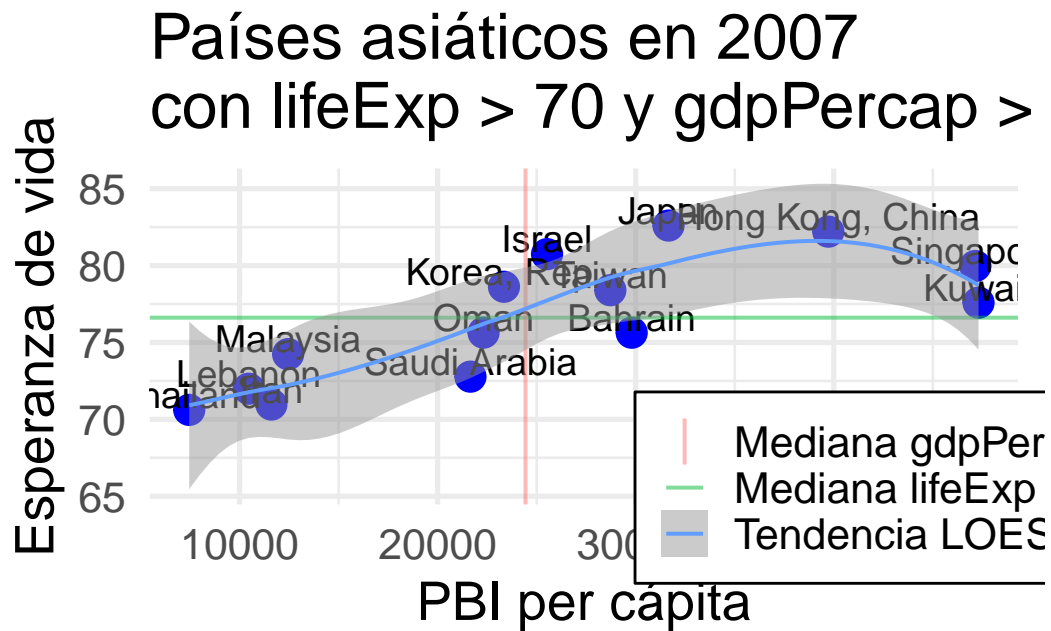
```
# Calcular medianas
mediana_lifeExp <- median(países_filtrados$lifeExp)
mediana_gdpPercap <- median(países_filtrados$gdpPercap)

# Scatterplot con leyenda interior de medianas y tendencia LOESS
ggplot(países_filtrados, aes(x = gdpPercap, y = lifeExp)) +
  geom_point(color = 'blue', size = 5) +
  geom_text(aes(label = country), nudge_y = 1, size = 5) +
  geom_hline(aes(yintercept = mediana_lifeExp, color = "Mediana lifeExp"),
             linewidth = 0.7, alpha = 0.5) +
  geom_vline(aes(xintercept = mediana_gdpPercap, color = "Mediana gdpPercap"),
             linewidth = 0.7, alpha = 0.5) +
  geom_smooth(aes(color = "Tendencia LOESS"), method = "loess", se = TRUE, linewidth = 0.7,
  labs(
    title = "Países asiáticos en 2007\ncon lifeExp > 70 y gdpPercap > 5000",
    x = "PBI per cápita",
    y = "Esperanza de vida",
    color = NULL) +
  theme_minimal(base_size = 20) +
  theme(
    legend.position = c(0.85, 0.05),
    legend.background = element_rect(fill = "white", color = "black", linewidth = 0.5)
  )
```

Warning: A numeric `legend.position` argument in `theme()` was deprecated in ggplot2 3.5.0.

i Please use the `legend.position.inside` argument of `theme()` instead.

```
`geom_smooth()` using formula = 'y ~ x'
```



### Versión interactiva

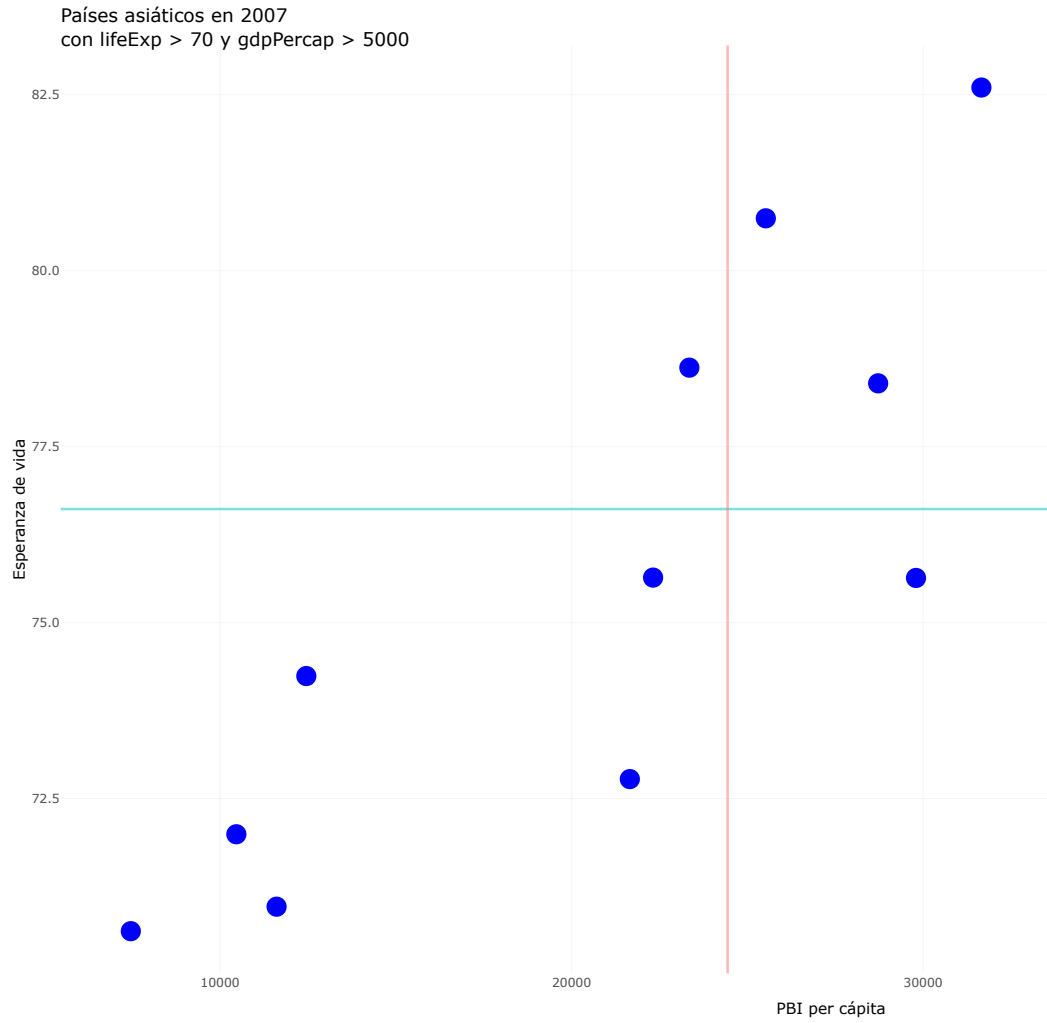
```
# Interactive: replica de la gráfica estática con Plotly
# Calcular medianas
mediana_lifeExp <- median(paises_filtrados$lifeExp)
mediana_gdpPercap <- median(paises_filtrados$gdpPercap)

p <- ggplot(paises_filtrados, aes(x = gdpPercap, y = lifeExp, text = country)) +
  geom_point(color = 'blue', size = 5) +
  geom_hline(aes(yintercept = mediana_lifeExp, color = "Mediana lifeExp"), linewidth = 0.7,
  geom_vline(aes(xintercept = mediana_gdpPercap, color = "Mediana gdpPercap"), linewidth = 0.7,
  labs(
    title = "Países asiáticos en 2007\ncon lifeExp > 70 y gdpPercap > 5000",
    x = "PBI per cápita",
    y = "Esperanza de vida",
    color = NULL) +
  theme_minimal(base_size = 12) +
  theme(
    legend.position.inside = c(0.95, 0.05),
```

```
    legend.background = element_rect(fill = "white", color = "black", linewidth = 0.5)
  )

ggplotly(p, tooltip = "text") %>%
  layout(legend = list(
    x = 0.95, y = 0.05,
    xanchor = "right", yanchor = "bottom",
    bgcolor = "white", bordercolor = "black", borderwidth = 0.5
  ))
```

file:///private/var/folders/fp/fmmgkb5548nf620fw64d0tl40000gn/T/Rtmp7bMyfP/file161e61826077,

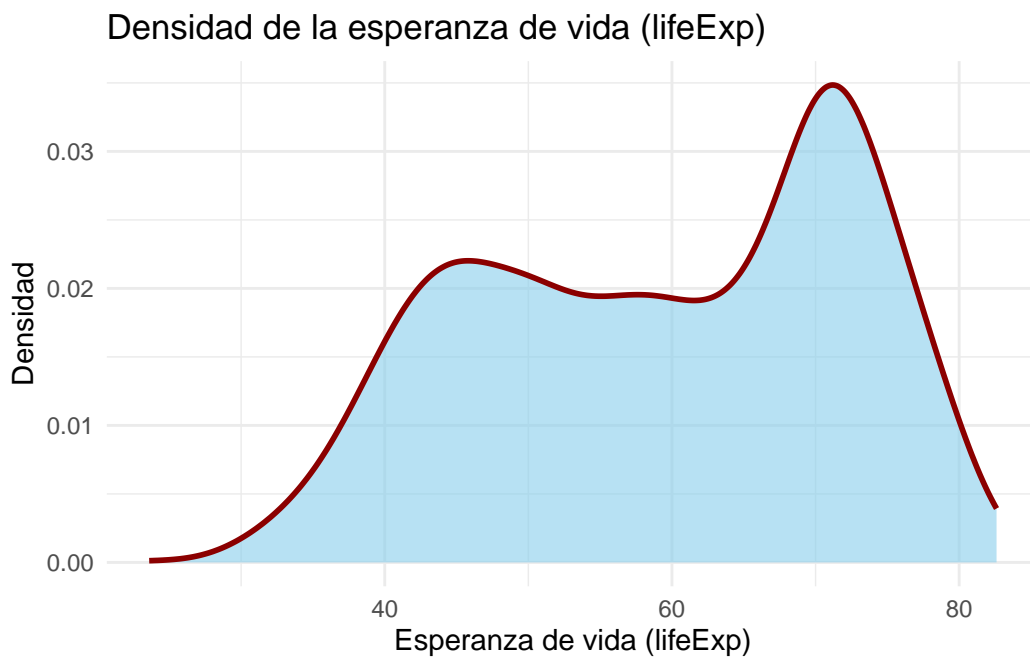


---

## Análisis exploratorio de la variable lifeExp

### Visualización de la distribución de lifeExp

```
ggplot(gapminder, aes(x = lifeExp)) +  
  geom_density(fill = 'skyblue', color = 'darkred', alpha = 0.6, linewidth = 1) +  
  labs(  
    title = 'Densidad de la esperanza de vida (lifeExp)',  
    x = 'Esperanza de vida (lifeExp)',  
    y = 'Densidad'  
  ) +  
  theme_minimal()
```



### Test de normalidad para lifeExp

```
# Tamaño de muestra utilizado para los tests  
tamano_muestra <- min(5000, length(gapminder$lifeExp))  
tamano_muestra
```

[1] 1704

```
# Test de Shapiro-Wilk (solo para n <= 5000)
shapiro_test <- shapiro.test(sample(gapminder$lifeExp, tamano_muestra))
shapiro_test
```

Shapiro-Wilk normality test

```
data: sample(gapminder$lifeExp, tamano_muestra)
W = 0.95248, p-value < 2.2e-16
```

```
# Test de Kolmogorov-Smirnov
ks_test <- ks.test(scale(gapminder$lifeExp), "pnorm")
```

Warning in ks.test.default(scale(gapminder\$lifeExp), "pnorm"): ties should not be present for the one-sample Kolmogorov-Smirnov test

```
ks_test
```

Asymptotic one-sample Kolmogorov-Smirnov test

```
data: scale(gapminder$lifeExp)
D = 0.1049, p-value < 2.2e-16
alternative hypothesis: two-sided
```

```
# Test de Anderson-Darling
ad_test <- ad.test(gapminder$lifeExp)
ad_test
```

Anderson-Darling normality test

```
data: gapminder$lifeExp
A = 28.009, p-value < 2.2e-16
```

**Tamaño de muestra utilizado:** Para ambos tests se utilizó una muestra de 1704 observaciones de la variable `lifeExp`.

**Interpretación de los resultados:**

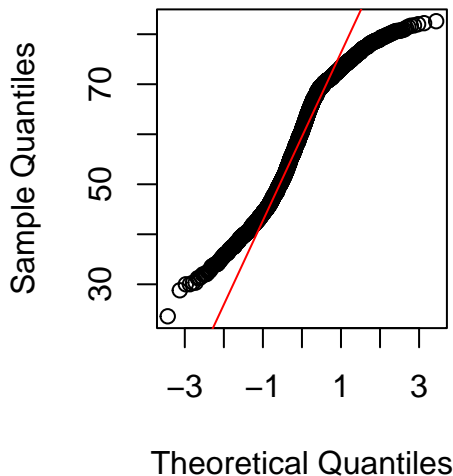
- **Shapiro-Wilk normality test:** - Estadístico  $W = 0.95248$ ,  $p\text{-value} < 2.2e-16$  - El p-valor extremadamente bajo indica que rechazamos la hipótesis nula de normalidad. Por tanto, la distribución de `lifeExp` no es normal (al menos para la muestra evaluada).
- **Kolmogorov-Smirnov test:** - Estadístico  $D = 0.1049$ ,  $p\text{-value} < 2.2e-16$  - También rechaza la hipótesis nula de normalidad para la variable estandarizada.

- **Anderson-Darling normality test:**

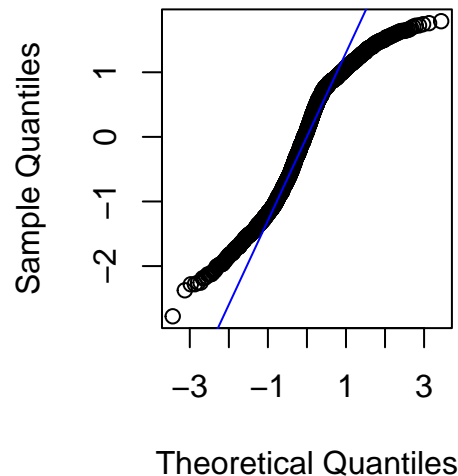
- Estadístico  $A = 28.009$ ,  $p\text{-value} < 2.2e-16$
- El resultado refuerza la evidencia contra la normalidad, con un p-valor extremadamente bajo que indica una fuerte discrepancia respecto a la distribución normal, especialmente en las colas.

```
# Q-Q plot
par(mfrow = c(1, 2))
qqnorm(gapminder$lifeExp, main = "QQ-plot de lifeExp")
qqline(gapminder$lifeExp, col = "red")
qqnorm(scale(gapminder$lifeExp), main = "QQ-plot de lifeExp estandarizada")
qqline(scale(gapminder$lifeExp), col = "blue")
```

**QQ-plot de lifeExp**



**QQ-plot de lifeExp estandariz**





```
par(mfrow = c(1, 1))  
  
kurtosis(gapminder$lifeExp)
```

```
[1] -1.129098
```

### Análisis visual (Q-Q plots):

En ambos gráficos (para `lifeExp` y su versión estandarizada) se observa una **curvatura en forma de S**, lo que indica que los datos se desvían sistemáticamente de la línea teórica que representaría una distribución normal. En particular: - En la cola inferior (cuantiles más bajos), los puntos están **por debajo** de la línea → hay más valores bajos de lo esperado bajo normalidad. - En la cola superior (cuantiles más altos), los puntos están **por encima** de la línea → hay más valores altos de lo esperado.

Sin embargo, el **exceso de curtosis negativo** observado ( $-1.13$ ) indica que la distribución es **platicúrtica**, es decir, tiene colas **más livianas** y una **menor concentración alrededor de la media** en comparación con una distribución normal. La forma de los Q-Q plots puede entonces explicarse por esta menor curtosis y una posible **asimetría leve**.

**Conclusión:** Tanto los tests estadísticos como la evidencia visual aportada por los Q-Q plots indican de forma consistente que la variable `lifeExp` **no sigue una distribución normal**. Esto debe tenerse en cuenta al seleccionar métodos estadísticos posteriores, especialmente aquellos que asumen normalidad. Dado el exceso de curtosis negativo y la desviación en los cuantiles extremos, podrían considerarse métodos no paramétricos, transformaciones o enfoques robustos.

## 2. Nueva variable ingreso\_total

```
# Crear variable ingreso_total y filtrar valores válidos  
aux_gapminder <- gapminder %>%  
  mutate(  
    ingreso_total = gdpPercap * pop,  
    log_ingreso_total = log(ingreso_total),  
    log_lifeExp = log(lifeExp)  
  ) %>%  
  filter(is.finite(log_ingreso_total), is.finite(log_lifeExp))
```

## Test de normalidad para ingreso\_total

```
# Tamaño de muestra utilizado para los tests
tamano_muestra <- min(5000, length(aux_gapminder$ingreso_total))
tamano_muestra
```

```
[1] 1704
```

```
# Test de Shapiro-Wilk (solo para n <= 5000)
shapiro_test <- shapiro.test(sample(aux_gapminder$ingreso_total, tamano_muestra))
shapiro_test
```

Shapiro-Wilk normality test

```
data:  sample(aux_gapminder$ingreso_total, tamano_muestra)
W = 0.23835, p-value < 2.2e-16
```

```
# Test de Kolmogorov-Smirnov
ks_test <- ks.test(scale(aux_gapminder$ingreso_total), "pnorm")
ks_test
```

Asymptotic one-sample Kolmogorov-Smirnov test

```
data:  scale(aux_gapminder$ingreso_total)
D = 0.39683, p-value < 2.2e-16
alternative hypothesis: two-sided
```

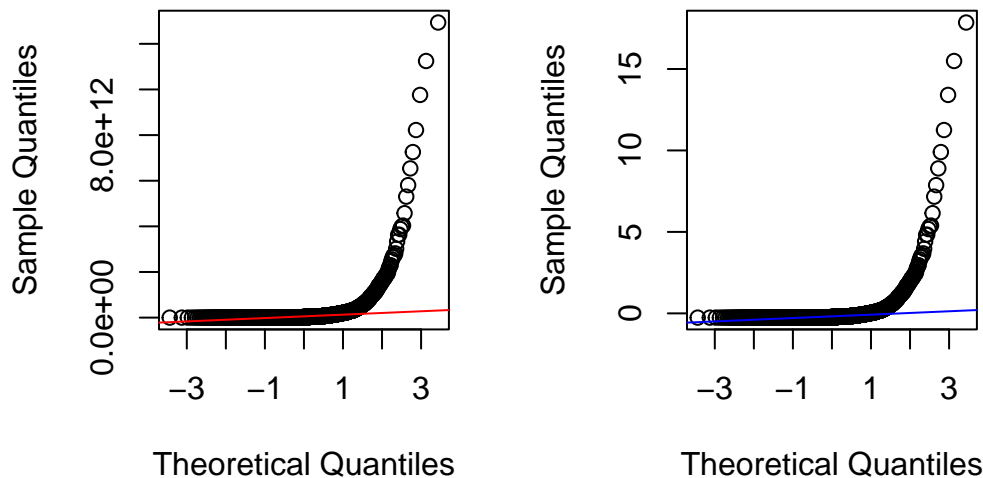
```
# Test de Anderson-Darling
ad_test <- ad.test(aux_gapminder$ingreso_total)
ad_test
```

Anderson-Darling normality test

```
data:  aux_gapminder$ingreso_total
A = 419.21, p-value < 2.2e-16
```

```
# Q-Q plot
par(mfrow = c(1, 2))
qqnorm(aux_gapminder$ingreso_total, main = "QQ-plot de ingreso_total")
qqline(aux_gapminder$ingreso_total, col = "red")
qqnorm(scale(aux_gapminder$ingreso_total), main = "QQ-plot de ingreso_total estandarizada")
qqline(scale(aux_gapminder$ingreso_total), col = "blue")
```

## QQ-plot de ingreso\_total-plot de ingreso\_total estand



```
par(mfrow = c(1, 1))

kurtosis(aux_gapminder$ingreso_total)
```

[1] 136.1171

### Normalidad de la variable ingreso\_total:

Se evaluó la normalidad de la variable `ingreso_total` mediante los tests de Shapiro-Wilk, Kolmogorov-Smirnov y Anderson-Darling, utilizando una muestra de 1704 observaciones. En todos los casos, el **p-valor fue menor a  $2.2e-16$** , lo que indica una **fuerte desviación respecto a la normalidad**. El valor extremadamente alto de curtosis ( 136) revela una distribución con colas extremadamente pesadas y fuerte asimetría, lo cual también se refleja en los Q-Q plots.

Dado este comportamiento, se aplicó una **transformación logarítmica** a la variable. Esta transformación reduce la asimetría y la curtosis, acercando la distribución a la normalidad, como se observa en los nuevos Q-Q plots y puede corroborarse mediante nuevos tests de normalidad.

## Test de normalidad para log\_ingreso\_total y log\_lifeExp

### Tests para log\_ingreso\_total

```
# Tamaño de muestra (máximo 5000) según log_ingreso_total
tamano_muestra <- min(5000, nrow(aux_gapminder))
tamano_muestra
```

```
[1] 1704
```

```
# Tests para log_ingreso_total

# Shapiro-Wilk
shapiro_log_ing <- shapiro.test(sample(aux_gapminder$log_ingreso_total, tamano_muestra))
shapiro_log_ing
```

Shapiro-Wilk normality test

```
data:  sample(aux_gapminder$log_ingreso_total, tamano_muestra)
W = 0.99843, p-value = 0.1131
```

```
# Kolmogorov-Smirnov
ks_log_ing <- ks.test(scale(aux_gapminder$log_ingreso_total), "pnorm")
ks_log_ing
```

Asymptotic one-sample Kolmogorov-Smirnov test

```
data:  scale(aux_gapminder$log_ingreso_total)
D = 0.023894, p-value = 0.285
alternative hypothesis: two-sided
```

```
# Anderson-Darling
ad_log_ing <- ad.test(aux_gapminder$log_ingreso_total)
ad_log_ing
```

Anderson-Darling normality test

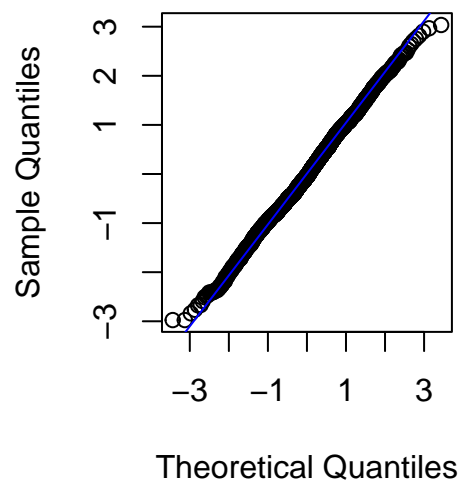
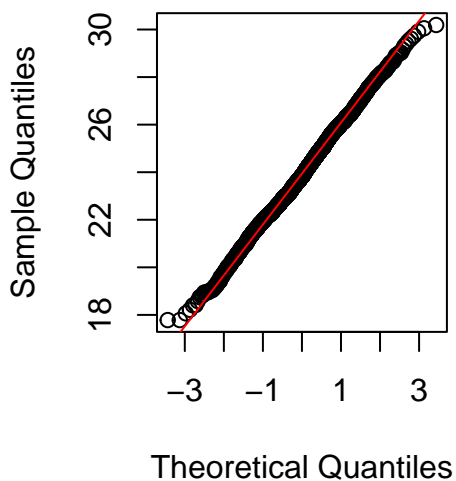
data: aux\_gapminder\$log\_ingreso\_total  
A = 0.91735, p-value = 0.01964

```
# Curtosis
curtosis_log_ing <- kurtosis(aux_gapminder$log_ingreso_total)
cat("Curtosis (log_ingreso_total):", curtosis_log_ing, "\n")
```

Curtosis (log\_ingreso\_total): -0.1419643

```
# Q-Q plots
par(mfrow = c(1, 2))
qqnorm(aux_gapminder$log_ingreso_total, main = "QQ-plot de log_ingreso_total")
qqline(aux_gapminder$log_ingreso_total, col = "red")
qqnorm(scale(aux_gapminder$log_ingreso_total), main = "QQ-plot de log_ingreso_total (est.)")
qqline(scale(aux_gapminder$log_ingreso_total), col = "blue")
```

## QQ-plot de log\_ingreso\_toQQ-plot de log\_ingreso\_total (



```
par(mfrow = c(1, 1))
```

### Tests para log\_lifeExp

```
# Shapiro-Wilk
shapiro_log_life <- shapiro.test(sample(aux_gapminder$log_lifeExp, tamaño_muestra))
shapiro_log_life
```

Shapiro-Wilk normality test

```
data: sample(aux_gapminder$log_lifeExp, tamaño_muestra)
W = 0.93724, p-value < 2.2e-16
```

```
# Kolmogorov-Smirnov
ks_log_life <- ks.test(scale(aux_gapminder$log_lifeExp), "pnorm")
```

Warning in ks.test.default(scale(aux\_gapminder\$log\_lifeExp), "pnorm"): ties should not be present for the one-sample Kolmogorov-Smirnov test

```
ks_log_life
```

Asymptotic one-sample Kolmogorov-Smirnov test

```
data: scale(aux_gapminder$log_lifeExp)
D = 0.11777, p-value < 2.2e-16
alternative hypothesis: two-sided
```

```
# Anderson-Darling
ad_log_life <- ad.test(aux_gapminder$log_lifeExp)
print(ad_log_life)
```

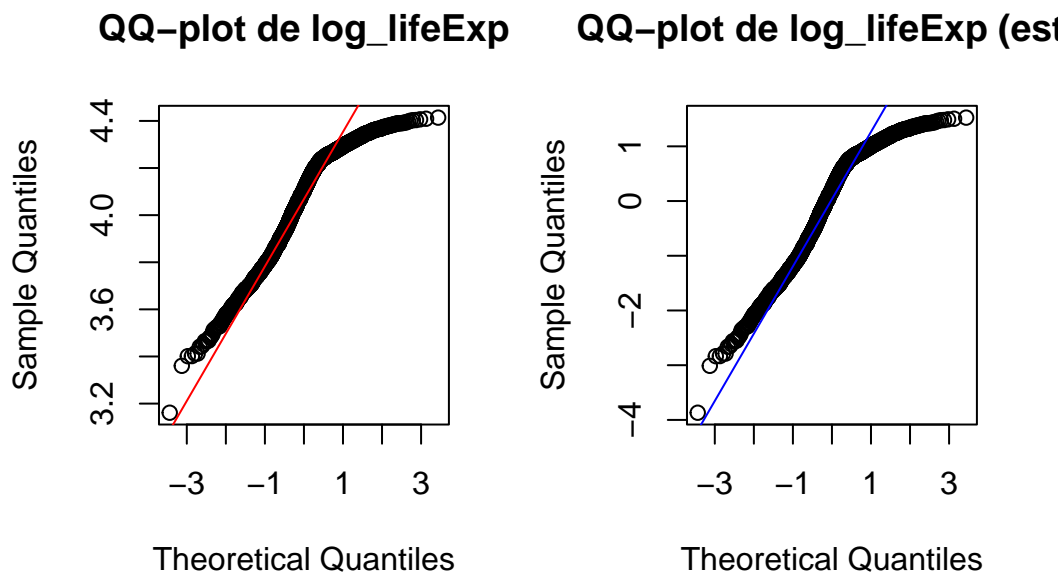
Anderson-Darling normality test

```
data: aux_gapminder$log_lifeExp
A = 36.148, p-value < 2.2e-16
```

```
# Curtosis
curtosis_log_life <- kurtosis(aux_gapminder$log_lifeExp)
cat("Curtosis (log_lifeExp):", curtosis_log_life, "\n")
```

Curtosis (log\_lifeExp): -0.6551934

```
# Q-Q plots
par(mfrow = c(1, 2))
qqnorm(aux_gapminder$log_lifeExp, main = "QQ-plot de log_lifeExp")
qqline(aux_gapminder$log_lifeExp, col = "red")
qqnorm(scale(aux_gapminder$log_lifeExp), main = "QQ-plot de log_lifeExp (est.)")
qqline(scale(aux_gapminder$log_lifeExp), col = "blue")
```



```
par(mfrow = c(1, 1))
```

### Interpretación de las transformaciones de lifeExp e ingreso\_total

#### Transformación de lifeExp:

Aunque se aplicó la transformación logarítmica, los tests de normalidad (Shapiro-Wilk, Kolmogorov-Smirnov y Anderson-Darling) siguen rechazando la hipótesis de

normalidad de `log_lifeExp` con p-valores  $< 2.2\text{e-}16$ . Además, la curtosis solo pasó de -1.13 a -0.66, indicando una platicurtosis aún moderada. Visualmente, los Q-Q plots muestran que la transformación no alteró sustancialmente la forma o la tendencia de la distribución original. **Por tanto, en este caso, la transformación logarítmica no consiguió acercar `lifeExp` a la normalidad.**

#### Transformación de `ingreso_total`:

En contraste, `ingreso_total` presentaba inicialmente una distribución extremadamente sesgada y leptocúrtica (curtosis = 136), lo que fue confirmado por todos los tests de normalidad y los Q-Q plots.

Tras aplicar la transformación logarítmica (`log_ingreso_total`), se observó una **mejora significativa**: - El p-valor del test de Shapiro-Wilk aumentó a 0.1131 (no se rechaza normalidad). - El Kolmogorov-Smirnov entregó un p-valor de 0.285 (no rechaza normalidad). - El Anderson-Darling aún rechaza marginalmente ( $p = 0.01964$ ), pero con un estadístico mucho menor ( $A = 0.92$ ). - La curtosis (en exceso) bajó a -0.14, acercándose a la de una distribución normal.

**Conclusión:** Mientras que la transformación logarítmica tuvo poco impacto sobre `lifeExp`, en `ingreso_total` logró corregir severas desviaciones, acercando la variable a una distribución normal. Este contraste ilustra la importancia de analizar caso por caso el efecto de las transformaciones.

### Ajuste de modelo log-log y análisis de residuos

```
# Ajustar regresión lineal simple
modelo <- lm(log_lifeExp ~ log_ingreso_total, data = aux_gapminder)

# Mostrar resumen del modelo
summary(modelo)
```

Call:

```
lm(formula = log_lifeExp ~ log_ingreso_total, data = aux_gapminder)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-0.79267	-0.12553	0.02024	0.14398	0.44172

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	2.392098	0.051407	46.53	<2e-16 ***



```
log_ingreso_total 0.069707    0.002141    32.56    <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1824 on 1702 degrees of freedom
Multiple R-squared:  0.3838,    Adjusted R-squared:  0.3835
F-statistic: 1060 on 1 and 1702 DF,  p-value: < 2.2e-16
```

### Modelo ajustado:

Se ajustó un modelo de regresión lineal simple con `log_lifeExp` como variable dependiente y `log_ingreso_total` como variable independiente. El modelo resultante fue:

$$\widehat{\log(\text{lifeExp})} = 2.392 + 0.070 \cdot \log(\text{ingreso\_total})$$

donde ambos coeficientes resultaron altamente significativos (p-valor < 2.2e-16).

### Interpretación económica:

En este contexto log-log, el coeficiente de `log_ingreso_total` ( 0.07) representa una **elasticidad**: un aumento del 1% en el ingreso total se asocia, en promedio, con un aumento de aproximadamente **0.07% en la esperanza de vida**.

### Bondad de ajuste:

El modelo explica cerca del 38% de la variabilidad en `log_lifeExp` ( $R^2 = 0.3838$ ), lo cual es moderado para datos agregados a nivel país-año, donde es esperable una gran heterogeneidad.

```
# Normalidad de los residuos
lillie_test <- lillie.test(residuals(modelo))
lillie_test
```

Lilliefors (Kolmogorov-Smirnov) normality test

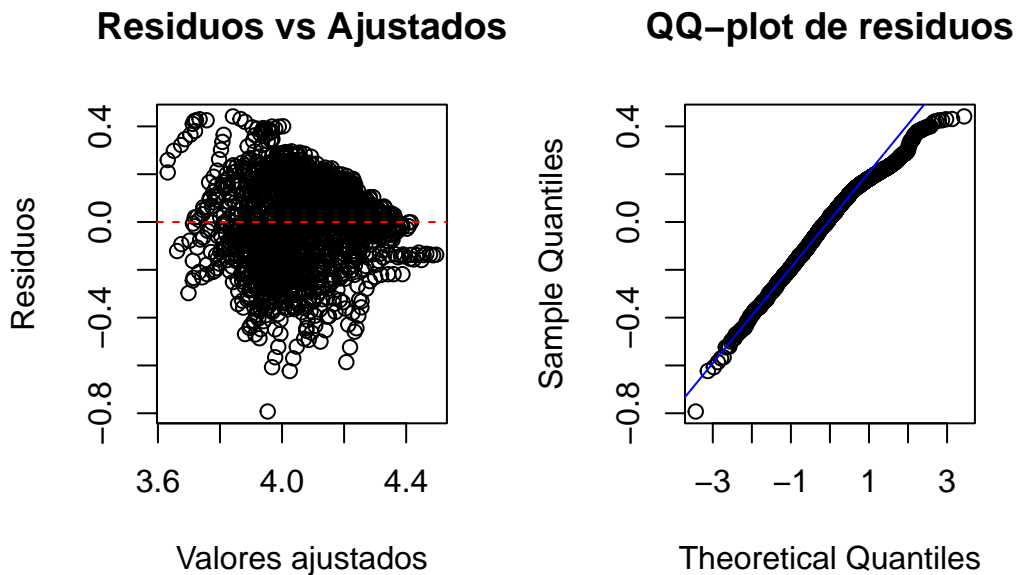
```
data: residuals(modelo)
D = 0.050137, p-value = 9.586e-11
```

```
# Homocedasticidad: Test de Breusch-Pagan
bptest(modelo)
```

studentized Breusch-Pagan test

```
data: modelo
BP = 68.254, df = 1, p-value < 2.2e-16
```

```
# Visualización de residuos
par(mfrow = c(1, 2))
plot(modelo$fitted.values, residuals(modelo),
     main = "Residuos vs Ajustados", xlab = "Valores ajustados", ylab = "Residuos")
abline(h = 0, lty = 2, col = 'red')
qqnorm(residuals(modelo), main = "QQ-plot de residuos")
qqline(residuals(modelo), col = 'blue')
```



```
par(mfrow = c(1, 1))
```

#### Normalidad:

El test de Lilliefors ( $D = 0.0501$ ,  $p < 1e-10$ ) rechaza la normalidad de los residuos. Esto indica que, aunque la relación entre las variables es lineal en escala logarítmica, los residuos **no son perfectamente normales**.

#### Homocedasticidad:

El test de Breusch-Pagan ( $BP = 68.25$ ,  $p < 2.2e-16$ ) también rechaza la hipótesis de homocedasticidad, indicando **presencia de heterocedasticidad**. Esto se confirma visualmente en el gráfico de “residuos vs ajustados”, donde se aprecia una dispersión creciente.

#### Visualización:

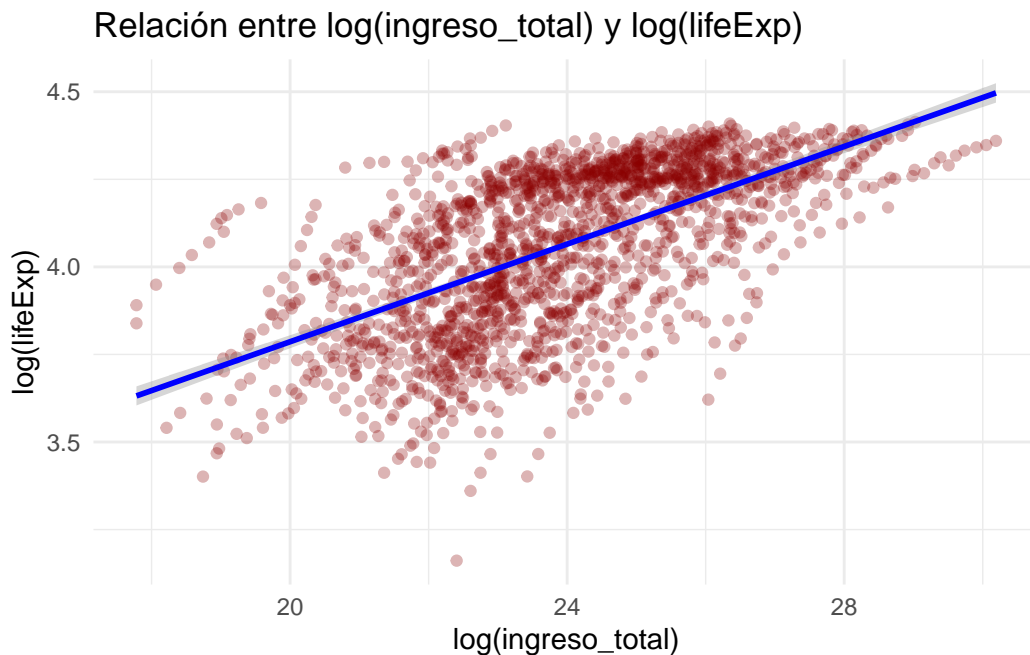
- El gráfico de residuos vs valores ajustados muestra un patrón de abanico,

típico de heterocedasticidad. - El **Q-Q plot de los residuos** revela desviaciones de la normalidad en las colas, consistentes con los resultados del test de Lilliefors.

### Visualización de la relación entre $\log(\text{ingreso\_total})$ y $\log(\text{lifeExp})$

```
# Visualizar la relación y la recta de regresión
ggplot(aux_gapminder, aes(x = log_ingreso_total, y = log_lifeExp)) +
  geom_point(alpha = 0.3, color = 'darkred') +
  geom_smooth(method = "lm", se = TRUE, color = "blue") +
  labs(
    title = "Relación entre log(ingreso_total) y log(lifeExp)",
    x = "log(ingreso_total)",
    y = "log(lifeExp)"
  ) +
  theme_minimal()
```

`geom\_smooth()` using formula = 'y ~ x'



El gráfico de dispersión entre  $\log(\text{ingreso\_total})$  y  $\log(\text{lifeExp})$ , junto con la recta de regresión, confirma una relación **positiva y aproximadamente lineal** en la escala logarítmica. El uso del logaritmo mejora claramente la linealidad y permite interpretar la pendiente en términos de elasticidad.

## Conclusión general de las transformaciones y el ajuste log-log

- Existe una relación positiva y estadísticamente significativa entre el ingreso total y la esperanza de vida cuando ambas variables son logarítmicas.
- El modelo capta una proporción moderada de la variabilidad, pero sus residuos presentan problemas de normalidad y homocedasticidad.
- A pesar de estos problemas, el ajuste log-log es conceptualmente apropiado para capturar la relación económica subyacente. Para una modelización más rigurosa, se podría considerar:
  - Uso de errores robustos (HC standard errors),
  - Modelos de regresión robusta,
  - O transformaciones adicionales para corregir la heterocedasticidad.

---

## Visualizaciones sobre la categorización de esperanza de vida (Ejercicio 1.3)

### 1. Línea apilada: proporción de categorías a lo largo del tiempo

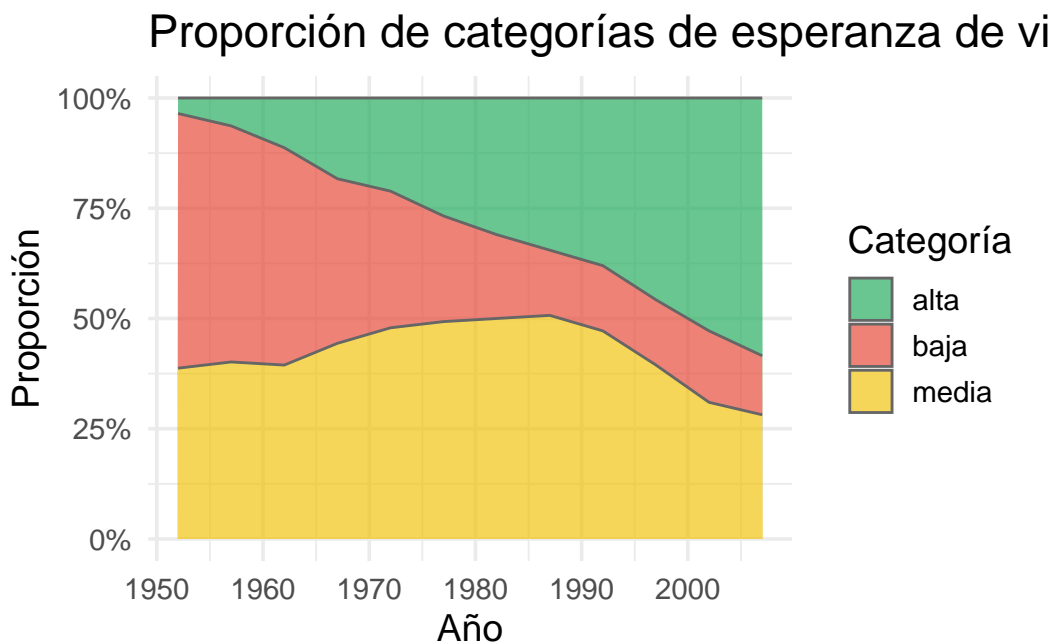
```
# Data frame con categoría
exp_cat_df <- gapminder %>%
  mutate(
    exp_cat = case_when(
      lifeExp < 50 ~ "baja",
      lifeExp >= 50 & lifeExp < 70 ~ "media",
      lifeExp >= 70 ~ "alta"
    )
  )

# Proporción por año y categoría
prop_exp_time <- exp_cat_df %>%
  group_by(year, exp_cat) %>%
  summarise(n = n(), .groups = "drop") %>%
  group_by(year) %>%
  mutate(prop = n / sum(n))

# Definir paleta consistente
colores_exp_cat <- c("baja" = "#e74c3c", "media" = "#f1c40f", "alta" = "#27ae60")

# Línea apilada con colores fijos
```

```
ggplot(prop_exp_time, aes(x = year, y = prop, fill = exp_cat)) +
  geom_area(alpha = 0.7, color = "grey40") +
  labs(
    title = "Proporción de categorías de esperanza de vida a lo largo del tiempo",
    x = "Año",
    y = "Proporción",
    fill = "Categoría"
  ) +
  scale_y_continuous(labels = scales::percent_format()) +
  scale_fill_manual(values = colores_exp_cat) +
  theme_minimal(base_size = 14)
```



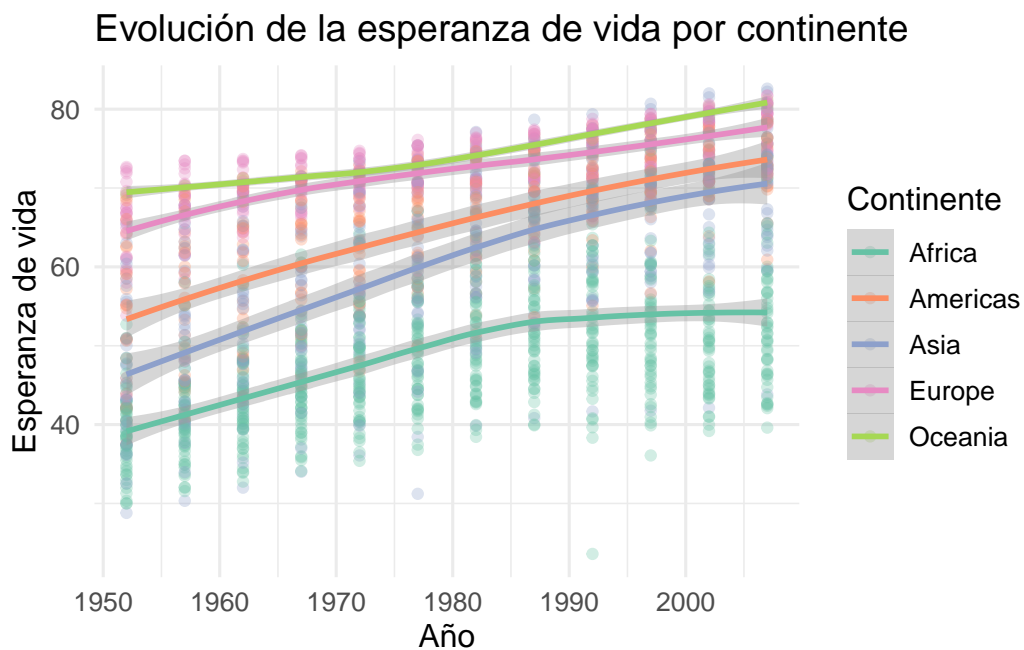
Este gráfico permite visualizar cómo la distribución de las categorías de esperanza de vida (baja, media, alta) evoluciona a lo largo del tiempo, facilitando la identificación de tendencias globales y transiciones demográficas en el conjunto de países analizados.

## Análisis de tendencias temporales por continente

### Evolución de la esperanza de vida por continente

```
# Gráfico de esperanza de vida vs tiempo por continente
ggplot(gapminder, aes(x = year, y = lifeExp, color = continent)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "loess", se = TRUE) +
  labs(
    title = "Evolución de la esperanza de vida por continente",
    x = "Año",
    y = "Esperanza de vida",
    color = "Continente"
  ) +
  theme_minimal(base_size = 12) +
  scale_color_brewer(palette = "Set2")
```

`geom\_smooth()` using formula = 'y ~ x'



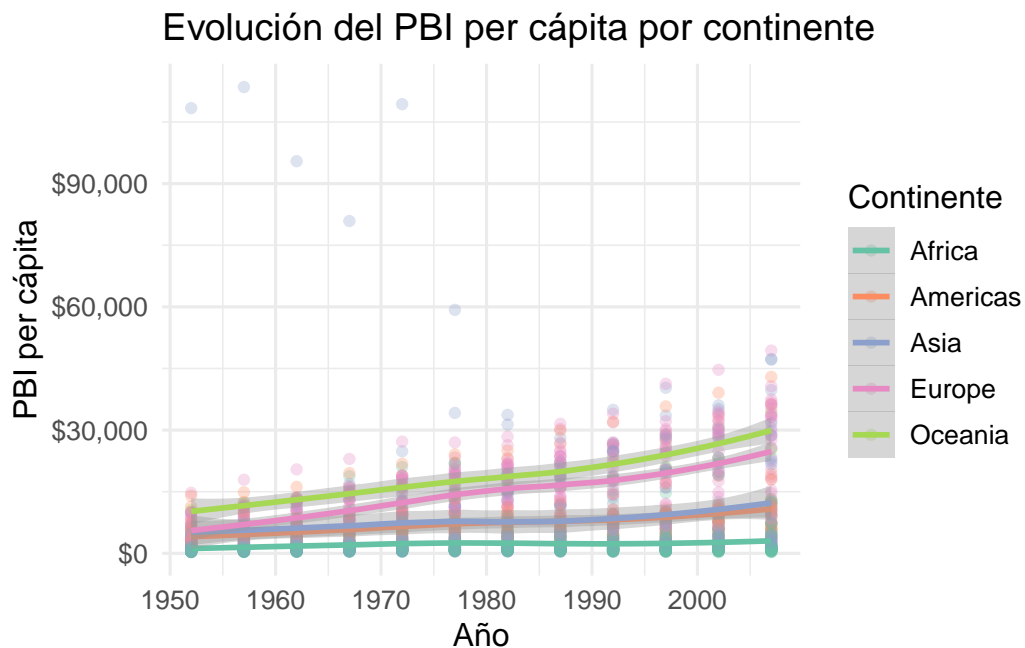
Este gráfico muestra cómo la esperanza de vida ha evolucionado de manera diferente en cada continente. Las bandas sombreadas representan el intervalo de confianza

de la tendencia LOESS, revelando tanto la tendencia central como la variabilidad en cada región.

### Evolución del PBI per cápita por continente

```
# Gráfico de PBI per cápita vs tiempo por continente
ggplot(gapminder, aes(x = year, y = gdpPercap, color = continent)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "loess", se = TRUE) +
  labs(
    title = "Evolución del PBI per cápita por continente",
    x = "Año",
    y = "PBI per cápita",
    color = "Continente"
  ) +
  theme_minimal(base_size = 12) +
  scale_color_brewer(palette = "Set2") +
  scale_y_continuous(labels = scales::dollar_format())
```

`geom\_smooth()` using formula = 'y ~ x'



La visualización del PBI per cápita revela patrones distintos de crecimiento económico entre continentes. Las tendencias suavizadas LOESS y sus intervalos de confianza permiten apreciar tanto la magnitud como la variabilidad del desarrollo económico en cada región.

### Relación entre esperanza de vida y PIB per cápita (2007)

```
# Identificar top 10 países por esperanza de vida en 2007
top10_2007 <- gapminder %>%
  filter(year == 2007) %>%
  arrange(desc(lifeExp)) %>%
  slice_head(n = 10) %>%
  mutate(is_top10 = TRUE)

# Preparar datos para visualización
data_2007 <- gapminder %>%
  filter(year == 2007) %>%
  left_join(select(top10_2007, country, is_top10), by = "country") %>%
  mutate(
    is_top10 = replace_na(is_top10, FALSE),
    label = if_else(is_top10, country, NA_character_)
  )

# Crear gráfico
ggplot(data_2007, aes(x = gdpPercap, y = lifeExp)) +
  # Puntos base
  geom_point(aes(color = is_top10, size = is_top10, alpha = is_top10)) +
  # Etiquetas para top 10
  geom_text_repel(
    aes(label = label),
    nudge_x = 0.05,
    size = 3,
    segment.color = "grey50"
  ) +
  # Línea de tendencia
  geom_smooth(method = "lm", color = "grey30", se = FALSE) +
  # Configuración estética
  scale_color_manual(
    values = c("FALSE" = "grey70", "TRUE" = "#e41a1c")
  ) +
  scale_size_manual(
```



```

    values = c("FALSE" = 2, "TRUE" = 3)
  ) +
  scale_alpha_manual(
    values = c("FALSE" = 0.5, "TRUE" = 1)
  ) +
  scale_x_log10(labels = scales::dollar_format()) +
  labs(
    title = "Relación entre PIB per cápita y esperanza de vida (2007)",
    subtitle = "Destacando los 10 países con mayor esperanza de vida",
    x = "PIB per cápita (escala logarítmica)",
    y = "Esperanza de vida"
  ) +
  theme_minimal(base_size = 12) +
  theme(
    legend.position = "none",
    plot.subtitle = element_text(color = "grey40")
  )

```

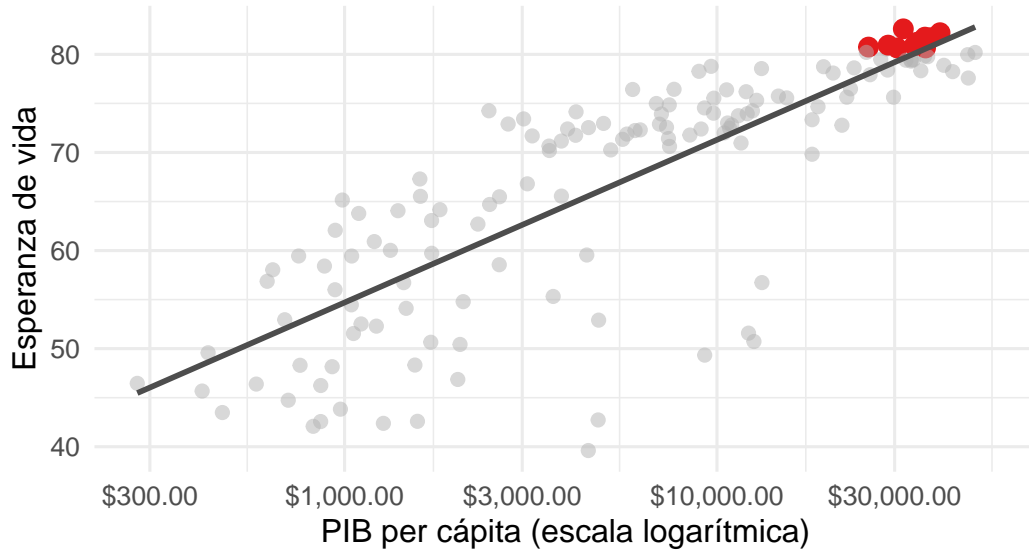
```
`geom_smooth()` using formula = 'y ~ x'
```

Warning: Removed 132 rows containing missing values or values outside the scale range (``geom_text_repel()``).

Warning: ggrepel: 10 unlabeled data points (too many overlaps). Consider increasing `max.overlaps`

## Relación entre PIB per cápita y esperanza de vida (2007)

Destacando los 10 países con mayor esperanza de vida



Esta visualización muestra la relación entre PIB per cápita y esperanza de vida en 2007, destacando en rojo los 10 países con mayor esperanza de vida. Los países del top 10 tienden a concentrarse en la parte superior derecha del gráfico, confirmando la asociación entre alto desarrollo económico y mayor longevidad.

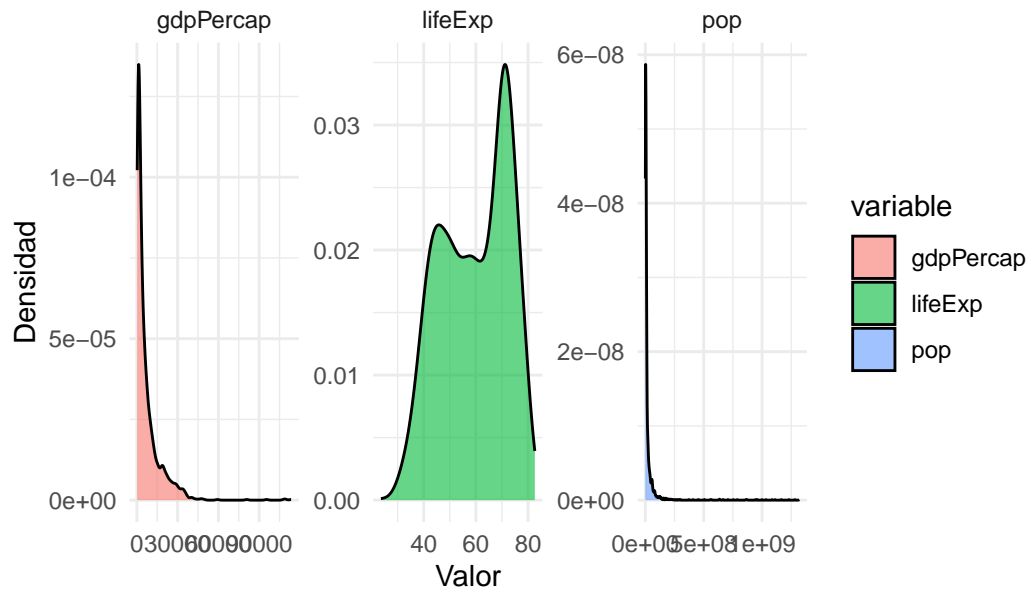
---

## Visualizaciones de distribuciones antes y después de estandarizar variables

```
# Visualización de distribuciones originales
tmp_orig <- gapminder %>%
  pivot_longer(cols = c(lifeExp, gdpPercap, pop),
               names_to = "variable", values_to = "valor")
ggplot(tmp_orig, aes(x = valor, fill = variable)) +
  geom_density(alpha = 0.6) +
  facet_wrap(~ variable, scales = "free") +
  labs(
    title = "Distribuciones de variables originales",
    x = "Valor",
    y = "Densidad"
```

```
) +  
theme_minimal()
```

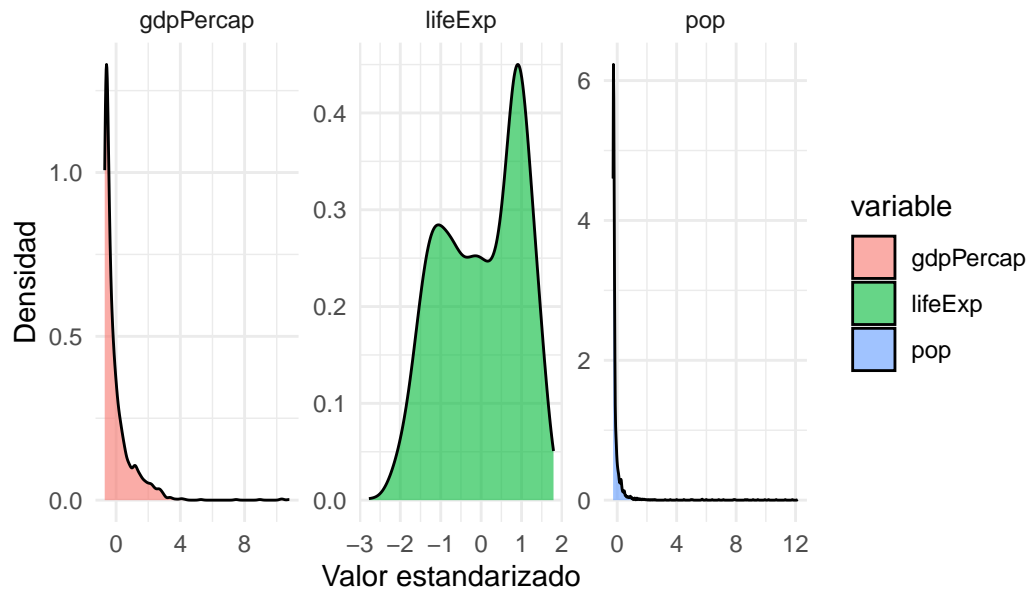
## Distribuciones de variables originales



```
# Visualización de distribuciones estandarizadas  
tmp_scaled <- gapminder %>%  
  mutate(  
    z_lifeExp = as.numeric(scale(lifeExp)),  
    z_gdpPercap = as.numeric(scale(gdpPercap)),  
    z_pop = as.numeric(scale(pop))  
  ) %>%  
  pivot_longer(  
    cols = starts_with("z_"),  
    names_to = "variable", values_to = "valor"  
  ) %>%  
  mutate(variable = sub("^z_", "", variable))  
ggplot(tmp_scaled, aes(x = valor, fill = variable)) +  
  geom_density(alpha = 0.6) +  
  facet_wrap(~ variable, scales = "free") +  
  labs(  
    title = "Distribuciones de variables estandarizadas",  
    x = "Valor estandarizado",  
    y = "Densidad"
```

```
) +  
theme_minimal()
```

## Distribuciones de variables estandarizadas



## Confirmar media y desviación de variables estandarizadas

```
# Calcular media y sd usando el objeto tmp_scaled generado anteriormente  
summary_estandarizado_aux <- tmp_scaled %>%  
  group_by(variable) %>%  
  summarise(  
    media = mean(valor, na.rm = TRUE),  
    desvio = sd(valor, na.rm = TRUE),  
    .groups = "drop"  
  )  
  
# Mostrar resultados  
summary_estandarizado_aux %>%  
  knitr::kable(  
    caption = "Media y desviación estándar de variables estandarizadas"  
  )
```

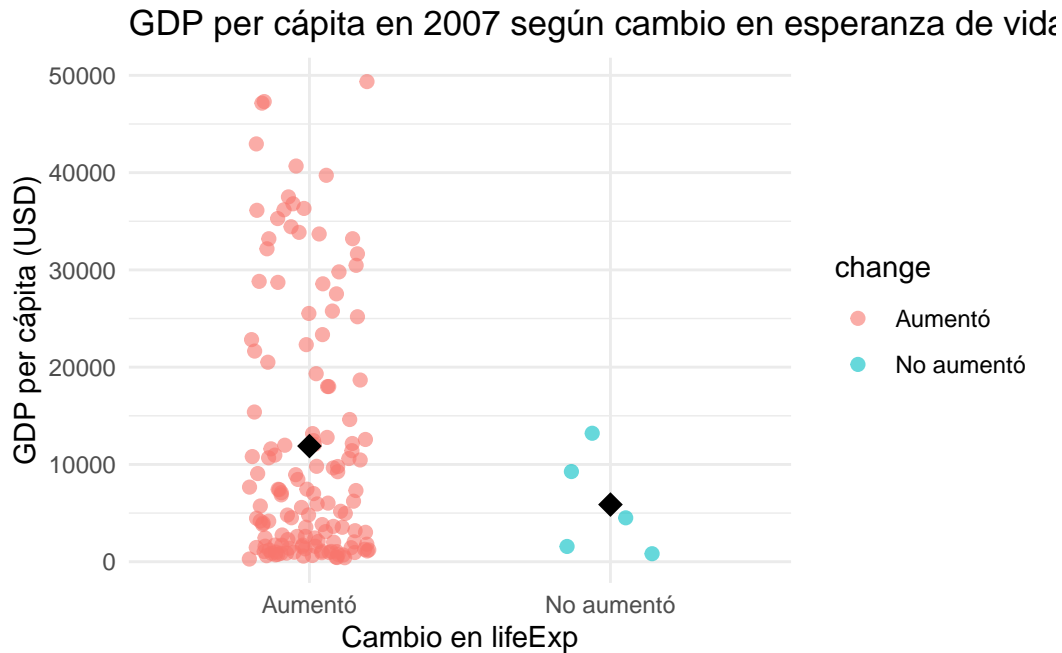
Table 1: Media y desviación estándar de variables estandarizadas

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

## Visualizaciones para ejercicio 1.14

```
data_change <- gapminder %>%
  group_by(country) %>%
  arrange(year) %>%
  mutate(lifeExp_diff = lifeExp - lag(lifeExp)) %>%
  filter(year == 2007) %>%
  mutate(change = if_else(lifeExp_diff > 0, "Aumentó", "No aumentó"))

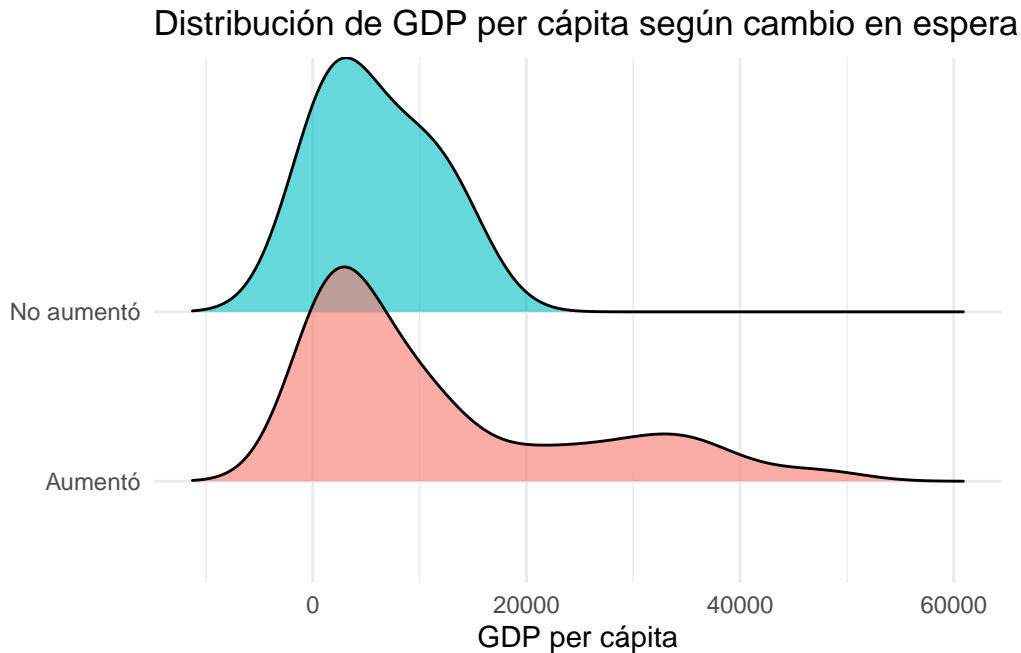
data_change %>%
  ggplot(aes(x = change, y = gdpPercap, color = change)) +
  geom_jitter(width = 0.2, alpha = 0.6, size = 2) +
  stat_summary(fun = mean, geom = "point", shape = 18, size = 4, color = "black") +
  labs(
    title = "GDP per cápita en 2007 según cambio en esperanza de vida",
    x = "Cambio en lifeExp",
    y = "GDP per cápita (USD)"
  ) +
  theme_minimal()
```



Se observa que la mayoría de los países experimentaron un aumento en la esperanza de vida entre 2002 y 2007. Dentro de ese grupo, hay una gran dispersión en el ingreso per cápita, incluyendo tanto países de ingresos bajos como muy altos. En contraste, los pocos países donde no aumentó la esperanza de vida presentan niveles consistentemente bajos de ingreso. Esto refuerza la asociación entre desarrollo económico y mejora en indicadores de salud.

```
data_change %>%
  ggplot(aes(x = gdpPercap, y = change, fill = change)) +
  geom_density_ridges(scale = 1.5, alpha = 0.6, show.legend = FALSE) +
  labs(
    title = "Distribución de GDP per cápita según cambio en esperanza de vida (2007)",
    x = "GDP per cápita",
    y = ""
  ) +
  theme_minimal()
```

Picking joint bandwidth of 3850



Aunque el gráfico es estéticamente atractivo, puede ser engañoso, especialmente en el grupo “No aumentó”, que contiene muy pocos países. La suavización genera una ilusión de densidad o volumen que en realidad no representa la escasez de datos. Por eso, es clave aclarar que las curvas no reflejan tamaños muestrales proporcionales y que deben interpretarse con precaución.

---

## Paleta para continentes con colores consistentes

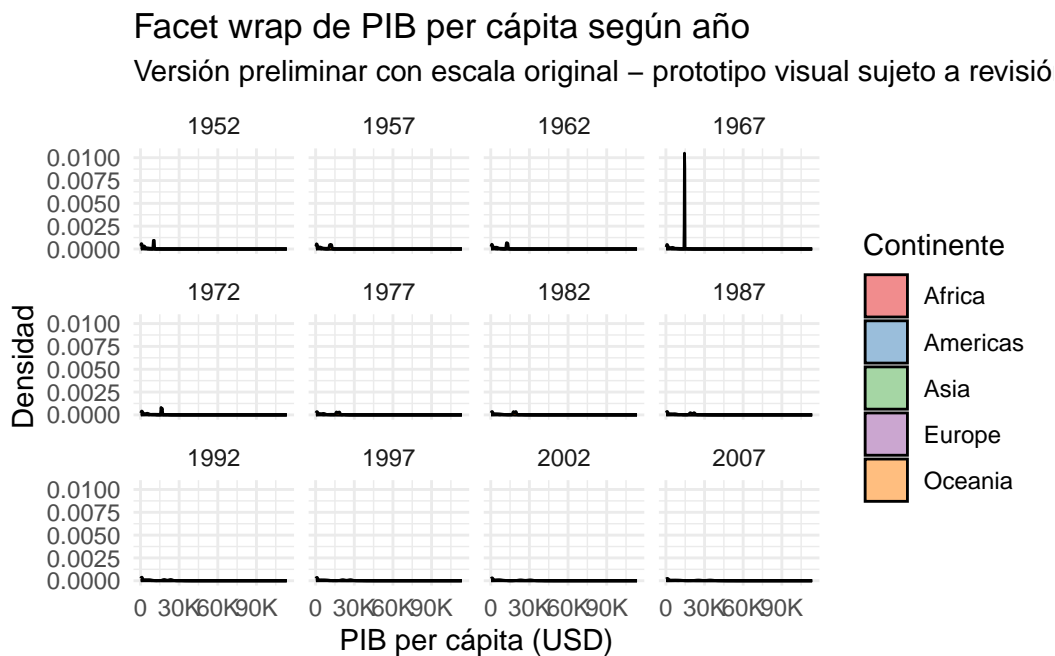
```
paleta_continentes <- scale_color_brewer(palette = "Set1")  
paleta_fill_continentes <- scale_fill_brewer(palette = "Set1")
```

### 2.3: Facet wrap de gdpPercap según año

#### Gráfico 1: Escala original (prototipo exploratorio)

```
# Facet wrap de `gdpPercap` según año - escala original
g1 <- ggplot(gapminder, aes(x = gdpPercap, fill = continent)) +
  geom_density(alpha = 0.5) +
  scale_x_continuous(labels = scales::label_number(scale_cut = scales::cut_short_scale())) +
  facet_wrap(~year, nrow = 3, ncol = 4) +
  labs(
    title = "Facet wrap de PIB per cápita según año",
    subtitle = "Versión preliminar con escala original - prototipo visual sujeto a revisión",
    x = "PIB per cápita (USD)",
    y = "Densidad",
    fill = "Continente"
  ) +
  theme_minimal() +
  paleta_fill_continentes

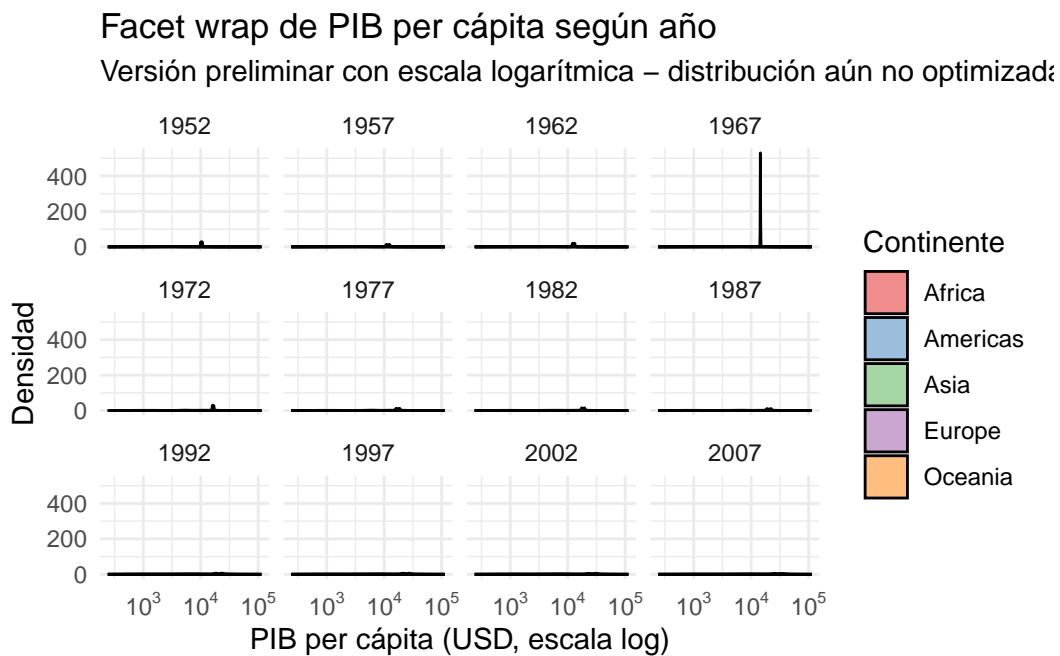
g1
```



**Gráfico 2: Escala logarítmica (prototipo exploratorio)**



```
# Facet wrap de `gdpPercap` según año - escala logarítmica
g2 <- ggplot(gapminder, aes(x = gdpPercap, fill = continent)) +
  geom_density(alpha = 0.5) +
  scale_x_log10(labels = scales::label_log()) +
  facet_wrap(~year, nrow = 3, ncol = 4) +
  labs(
    title = "Facet wrap de PIB per cápita según año",
    subtitle = "Versión preliminar con escala logarítmica - distribución aún no optimizada",
    x = "PIB per cápita (USD, escala log)",
    y = "Densidad",
    fill = "Continente"
  ) +
  theme_minimal() +
  paleta_fill_continentes
g2
```

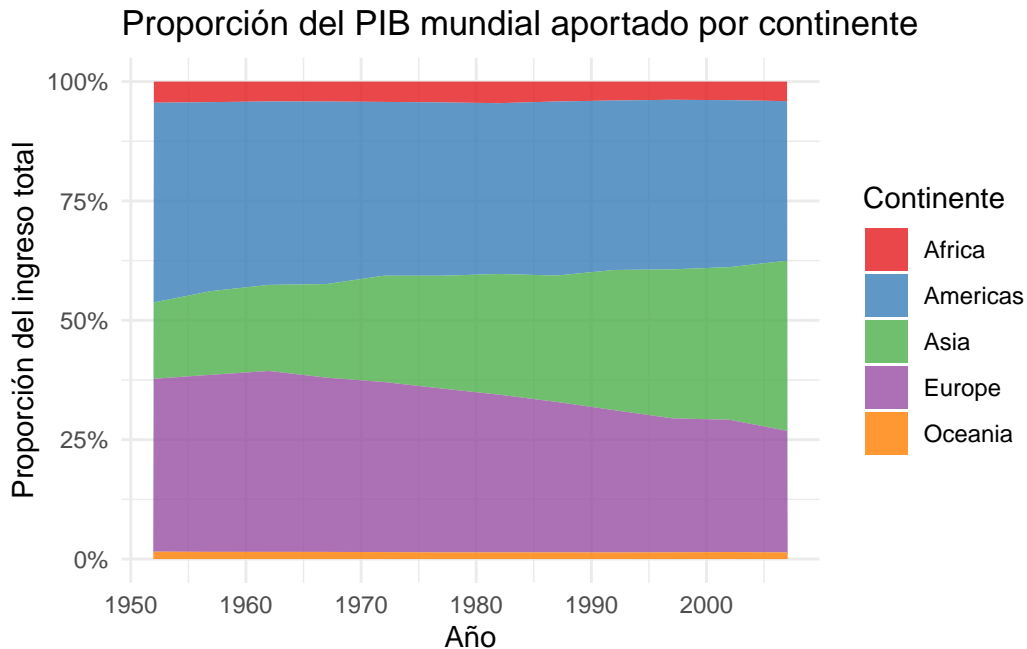


## Alternativa 1: Gráfico de línea apilada (área) de proporciones de $\text{gdpPercap} \times \text{pop}$ por continente

Qué muestra:

- Proporción del ingreso total mundial aportado por cada continente en cada año (ingreso total =  $\text{gdpPercap} \times \text{pop}$ ).
- Facetear por año no tiene sentido acá, así que **esta opción es complementaria, no sustituta** de 2.3.

```
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()) +
  scale_fill_brewer(palette = "Set1") +
  labs(
    title = "Proporción del PIB mundial aportado por continente",
    x = "Año",
    y = "Proporción del ingreso total",
    fill = "Continente"
  ) +
  theme_minimal()
```



## Alternativa 2: Histograma facetado (gdpPercap con continentes)

Qué muestra:

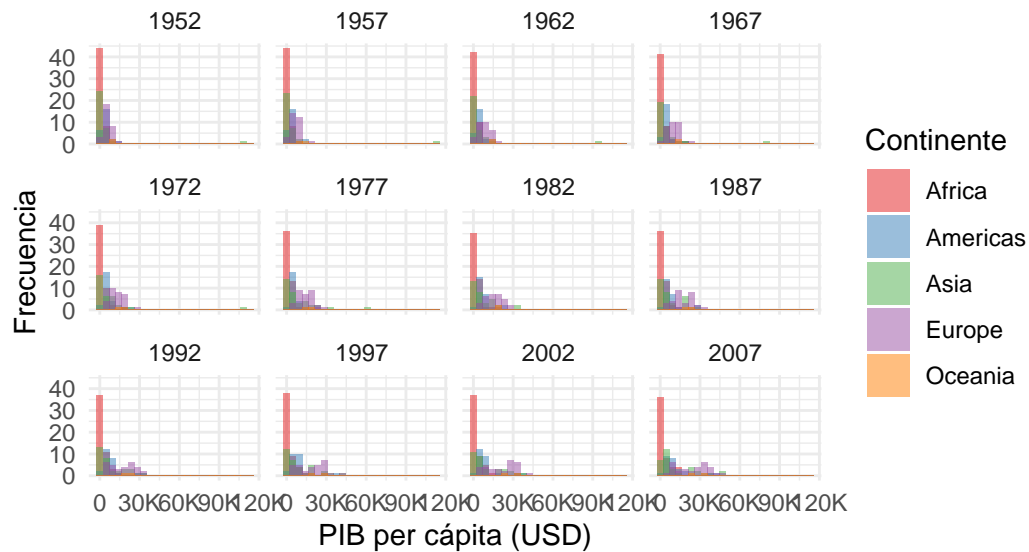
- Distribución de países según su PIB per cápita.
- Escala logarítmica para evitar la distorsión por outliers.
- Cada panel representa un año → **respetar literalmente la consigna.**

```
g1 <- gapminder %>%
  ggplot(aes(x = gdpPercap, fill = continent)) +
  geom_histogram(bins = 25, alpha = 0.5, position = "identity") +
  scale_x_continuous(labels = scales::label_number(scale_cut = scales::cut_short_scale())) +
  facet_wrap(~ year, ncol = 4) +
  labs(
    title = "Distribución del PIB per cápita por año",
    subtitle = "Escala original",
    x = "PIB per cápita (USD)",
    y = "Frecuencia",
    fill = "Continente"
  ) +
  theme_minimal() +
  paleta_fill_continentes
```

g1

## Distribución del PIB per cápita por año

Escala original

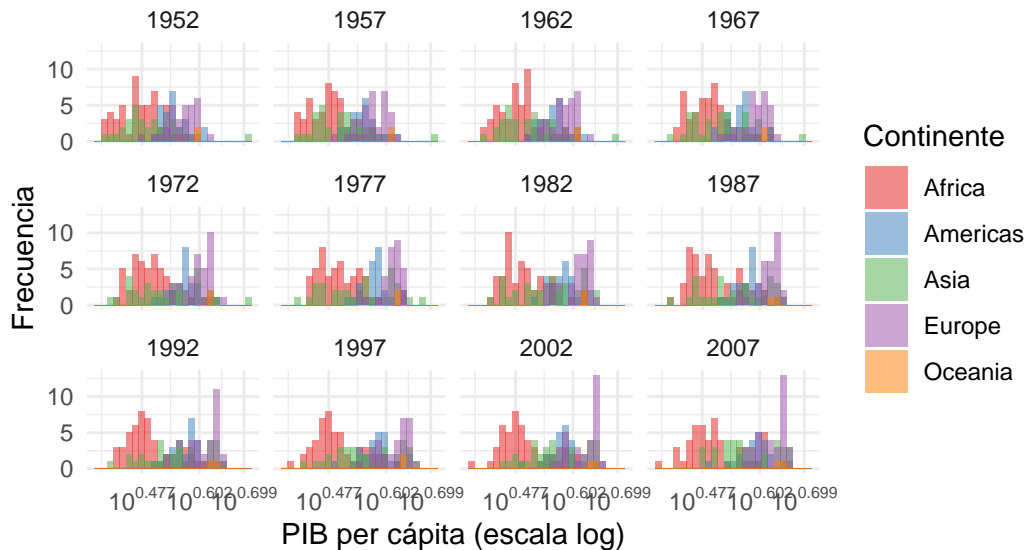


```
g2 <- gapminder %>%
  ggplot(aes(x = log10(gdpPercap), fill = continent)) +
  geom_histogram(bins = 25, alpha = 0.5, position = "identity") +
  scale_x_log10(labels = scales::label_log()) +
  facet_wrap(~ year, ncol = 4) +
  labs(
    title = "Distribución logarítmica del PIB per cápita por año",
    subtitle = "Facetas por año, escala log10",
    x = "PIB per cápita (escala log)",
    y = "Frecuencia",
    fill = "Continente"
  ) +
  theme_minimal() +
  paleta_fill_continentes
```

g2

## Distribución logarítmica del PIB per cápita por año

Facetas por año, escala log10



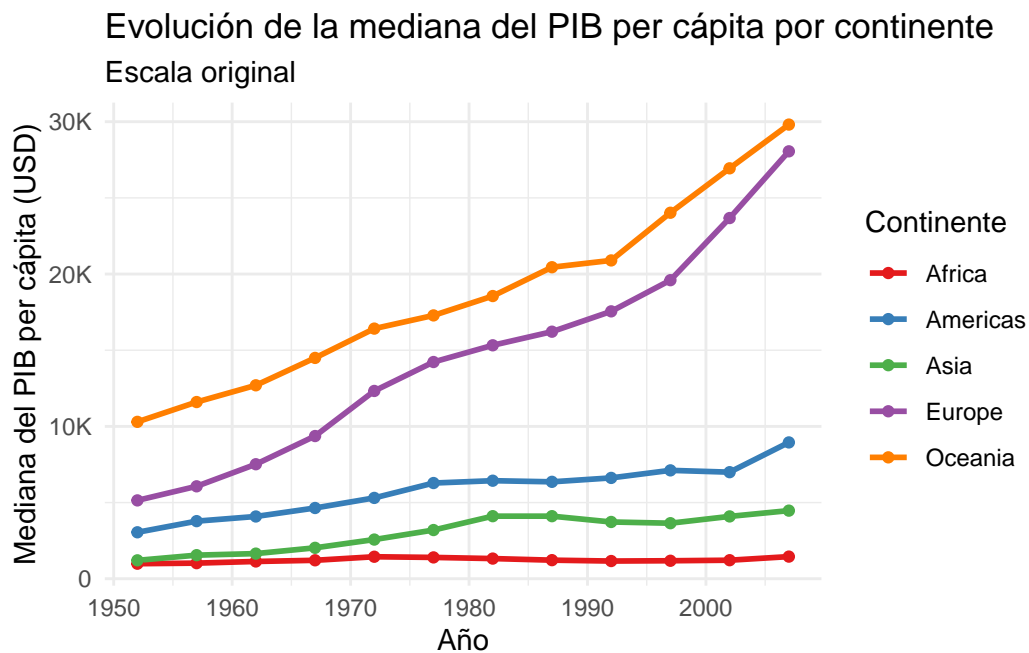
### Alternativa 3: Evolución de la mediana de gdpPercap por continente (línea temporal)

Este gráfico muestra cómo evoluciona el nivel típico (mediana) del PIB per cápita en cada continente. Es ideal para mostrar tendencias macroeconómicas.

```
g1 <- 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",
    subtitle = "Escala original",
    x = "Año",
    y = "Mediana del PIB per cápita (USD)",
    color = "Continente"
  ) +
  theme_minimal() +
  paleta_continentes
```

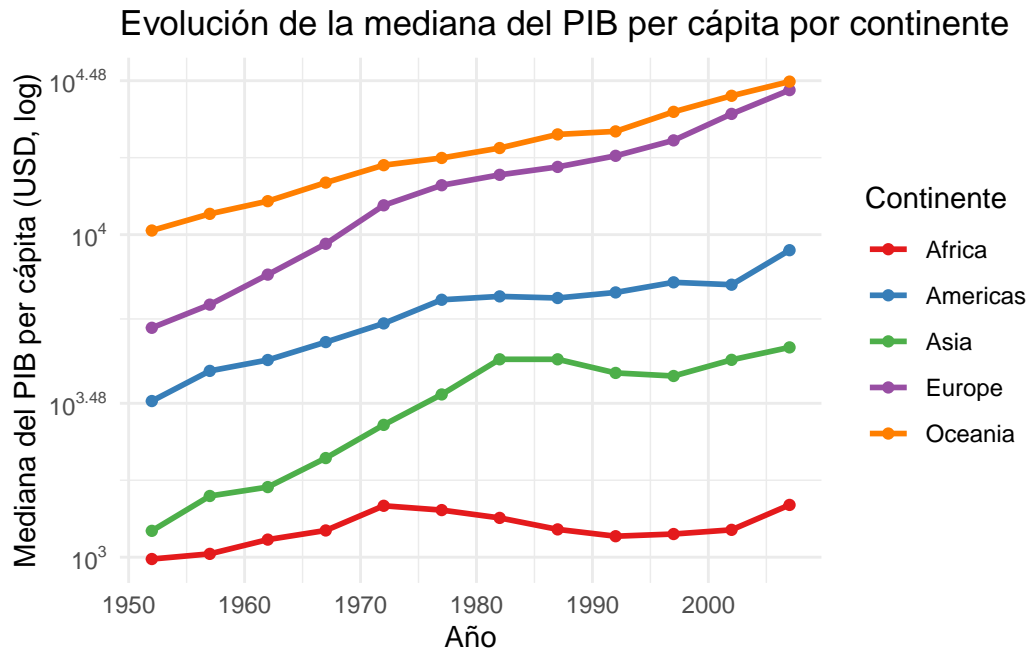
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
 i Please use `linewidth` instead.

g1



```
g2 <- 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_log10(labels = scales::label_log()) +
  labs(
    title = "Evolución de la mediana del PIB per cápita por continente",
    x = "Año",
    y = "Mediana del PIB per cápita (USD, log)",
    color = "Continente"
  ) +
  theme_minimal() +
  paleta_continentes
```

g2



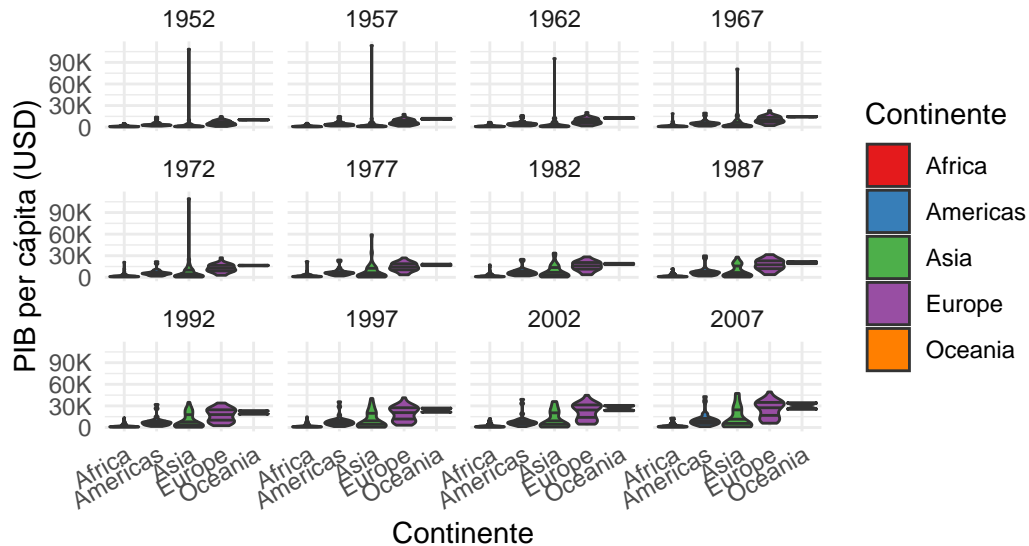
#### Alternativa 4: Evolución de toda la distribución — faceteado por año + violines

Esto sí cumple estrictamente con la consigna “*Facet wrap según año*”, pero en lugar de usar `geom_density()`, usamos **violin plots** para mostrar la dispersión y forma en cada panel:

```
g1 <- gapminder %>%
  ggplot(aes(x = continent, y = gdpPercap, fill = continent)) +
  geom_violin(scale = "width", draw_quantiles = c(0.25, 0.5, 0.75)) +
  scale_y_continuous(labels = scales::label_number(scale_cut = scales::cut_short_scale())) +
  facet_wrap(~ year, ncol = 4) +
  labs(
    title = "Distribución del PIB per cápita por continente en cada año",
    subtitle = "Escala original",
    x = "Continente",
    y = "PIB per cápita (USD)",
    fill = "Continente"
  ) +
  theme_minimal() +
  paleta_fill_continentes +
  theme(axis.text.x = element_text(angle = 30, hjust = 1))
```

g1

## Distribución del PIB per cápita por continente en cada año Escala original



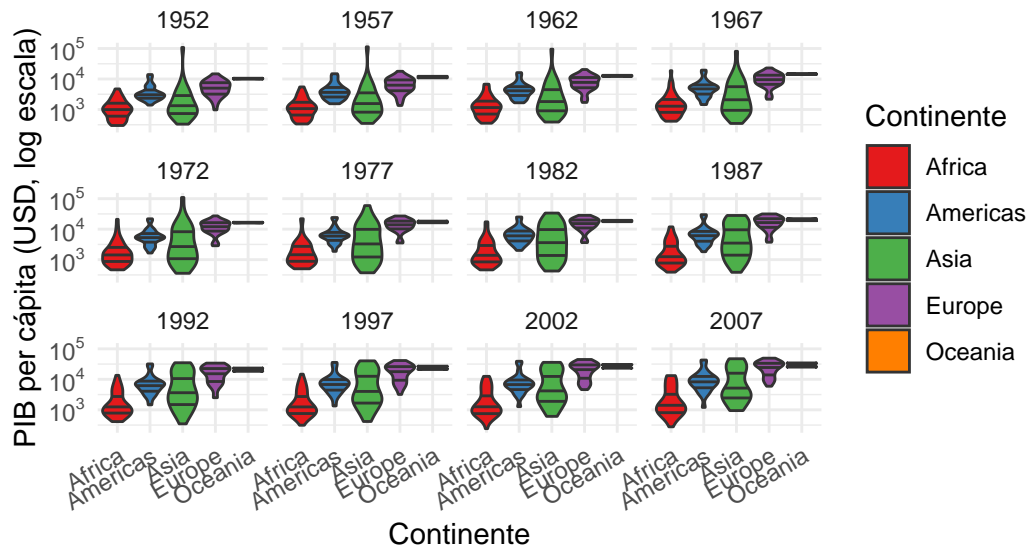
```
g2 <- gapminder %>%
  ggplot(aes(x = continent, y = gdpPercap, fill = continent)) +
  geom_violin(scale = "width", draw_quantiles = c(0.25, 0.5, 0.75)) +
  scale_y_log10(labels = scales::label_log()) +
  facet_wrap(~year, ncol = 4) +
  labs(
    title = "Distribución del PIB per cápita por continente 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))
```

g2



## Distribución del PIB per cápita por continente en cada año

Escala logarítmica para evitar distorsión por outliers



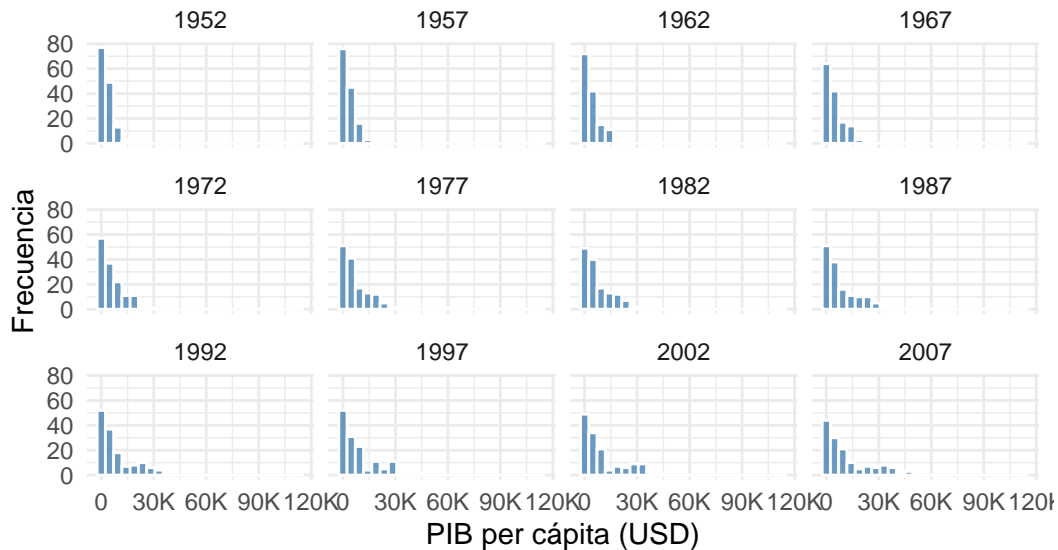
## Alternativa 5: Histograma facetado de $\log(\text{gdpPercap})$ por año (sin continentes)

```
g1 <- gapminder %>%
  ggplot(aes(x = gdpPercap)) +
  geom_histogram(bins = 25, fill = "steelblue", color = "white", alpha = 0.8) +
  scale_x_continuous(labels = scales::label_number(scale_cut = scales::cut_short_scale())) +
  facet_wrap(~ year, ncol = 4) +
  labs(
    title = "Distribución del PIB per cápita por año",
    subtitle = "Sin separar por continente - escala original",
    x = "PIB per cápita (USD)",
    y = "Frecuencia"
  ) +
  theme_minimal()
```

g1

## Distribución del PIB per cápita por año

Sin separar por continente – escala original

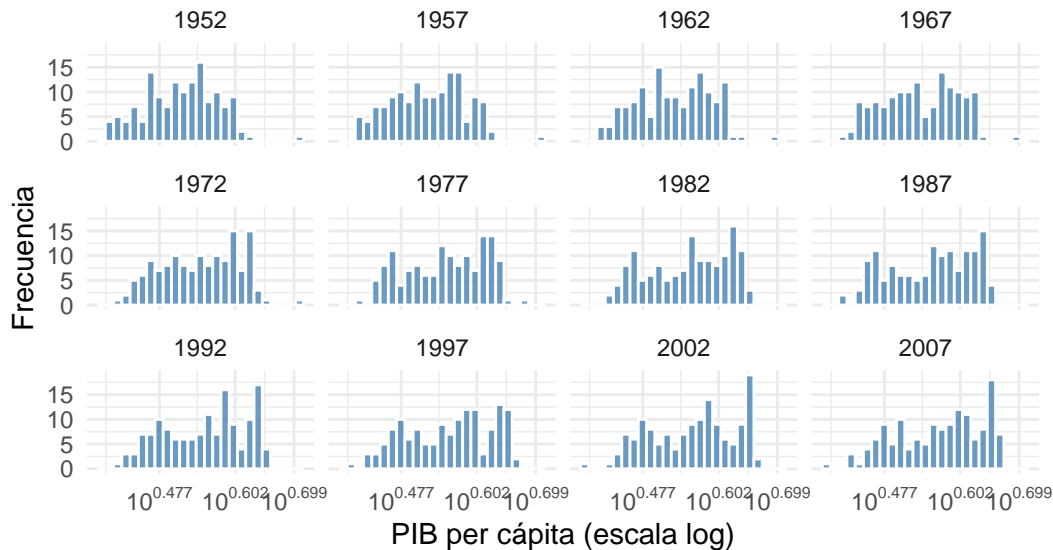


```
g2 <- gapminder %>%
  ggplot(aes(x = log10(gdpPercap))) +
  geom_histogram(bins = 25, fill = "steelblue", color = "white", alpha = 0.8) +
  scale_x_log10(labels = scales::label_log()) +
  facet_wrap(~ year, ncol = 4) +
  labs(
    title = "Distribución del PIB per cápita (escala log) por año",
    subtitle = "Se muestra la distribución de países sin separar por continente",
    x = "PIB per cápita (escala log)",
    y = "Frecuencia"
  ) +
  theme_minimal()
```

g2

## Distribución del PIB per cápita (escala log) por año

Se muestra la distribución de países sin separar por continente



## Alternativa 6: Diagrama de violín facetado por año (sin continentes)

```
# Escala original
g1 <- gapminder %>%
  ggplot(aes(x = factor(year), y = gdpPercap)) +
  geom_violin(fill = "darkorange", alpha = 0.7,
              draw_quantiles = c(0.25, 0.5, 0.75)) +
  scale_y_continuous(labels = scales::label_number(scale_cut = scales::cut_short_scale())) +
  labs(
    title = "Distribución del PIB per cápita por año",
    subtitle = "Violines verticales - escala original",
    x = "Año",
    y = "PIB per cápita (USD)"
  ) +
  theme_minimal()

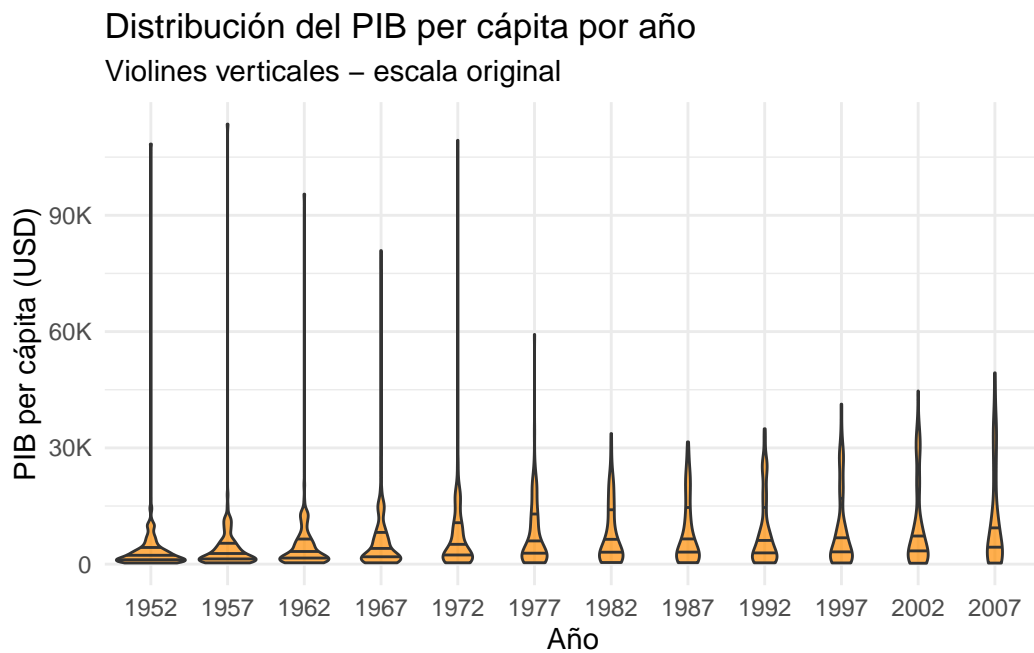
# Escala logarítmica
g2 <- gapminder %>%
  ggplot(aes(x = factor(year), y = gdpPercap)) +
```

```

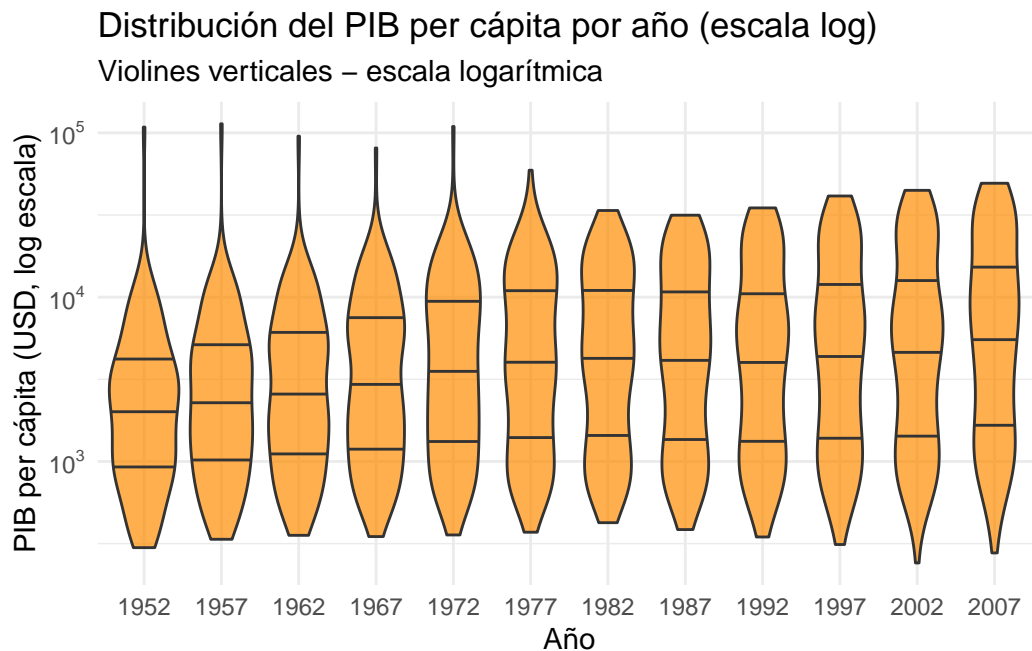
geom_violin(fill = "darkorange", alpha = 0.7,
            draw_quantiles = c(0.25, 0.5, 0.75)) +
scale_y_log10(labels = scales::label_log()) +
labs(
  title = "Distribución del PIB per cápita por año (escala log)",
  subtitle = "Violines verticales - escala logarítmica",
  x = "Año",
  y = "PIB per cápita (USD, log escala)"
) +
theme_minimal()

# Visualización
g1

```



g2



**\*\*Alternativa 7: Split-violin de gdpPercap vs lag(gdpPercap) por continente**

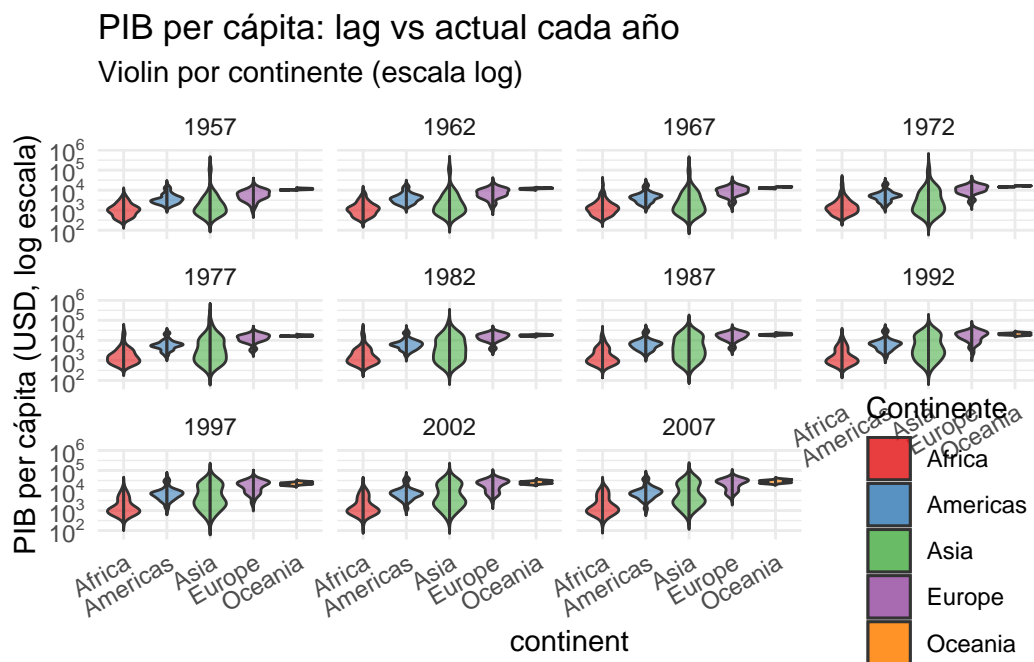
```
# Preparo datos con lag de 1 año por país
df_halves <- gapminder %>%
  group_by(country) %>%
  arrange(year) %>%
  mutate(lag_gdp = lag(gdpPercap)) %>%
  ungroup() %>%
  filter(!is.na(lag_gdp))

# Split-violin en log-escala, faceteado por año
ggplot(df_halves, aes(x = continent, fill = continent)) +
  geom_half_violin(
    aes(y = lag_gdp),
    side = "l",
    alpha = 0.6,
    scale = "width",
    trim = FALSE
  ) +
  geom_half_violin(
    aes(y = gdpPercap),
    side = "r",
    alpha = 0.6,
    scale = "width",
    trim = FALSE
  )
```

```

    side = "r",
    alpha = 0.6,
    scale = "width",
    trim = FALSE
  ) +
  scale_y_log10(labels = label_log()) +
  facet_wrap(~year, ncol = 4) +
  labs(
    title = "PIB per cápita: lag vs actual cada año",
    subtitle = "Violin por continente (escala log)",
    y = "PIB per cápita (USD, log escala)",
    fill = "Continente"
  ) +
  theme_minimal() +
  paleta_fill_continentes +
  theme(
    axis.text.x = element_text(angle = 30, hjust = 1),
    legend.position = c(0.9, 0.025)
  )
)

```

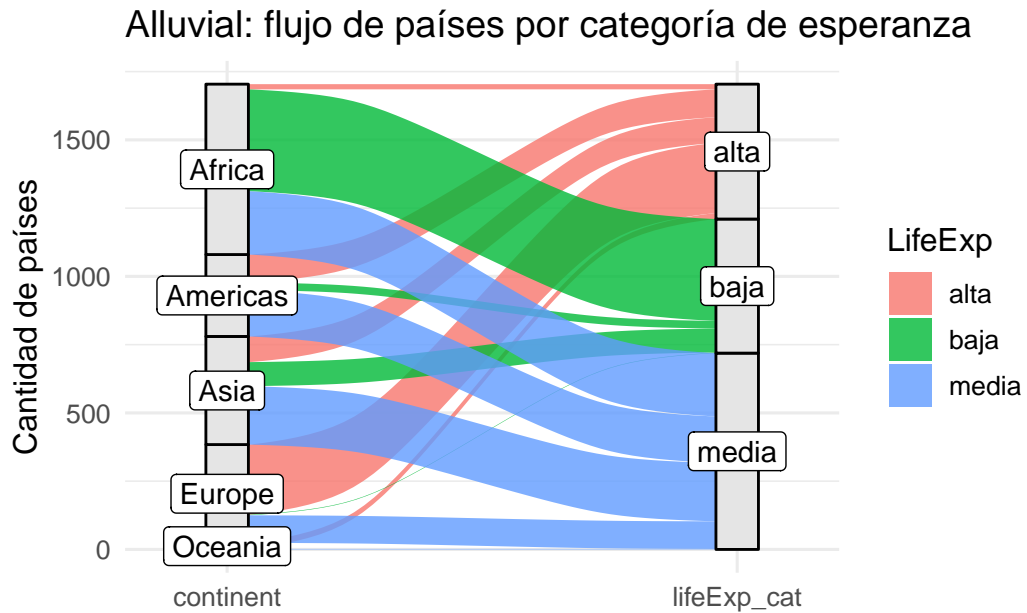


## Diagrama de mosaico de continent contra una categoría derivada de lifeExp

```
library(ggalluvial)

data_mosaic <- gapminder %>%
  mutate(
    lifeExp_cat = case_when(
      lifeExp < 50 ~ "baja",
      lifeExp < 70 ~ "media",
      TRUE       ~ "alta"
    )
  )

data_mosaic %>%
  count(continent, lifeExp_cat) %>%
  ggplot(aes(axis1 = continent, axis2 = lifeExp_cat, y = n)) +
  geom_alluvium(aes(fill = lifeExp_cat), width = 1/12, alpha = 0.8) +
  geom_stratum(width = 1/12, fill = "grey90") +
  geom_label(stat = "stratum", aes(label = after_stat(stratum))) +
  scale_x_discrete(
    limits = c("continent", "lifeExp_cat"),
    expand = c(.1, .1)
  ) +
  labs(
    title = "Alluvial: flujo de países por categoría de esperanza",
    x      = "",
    y      = "Cantidad de países",
    fill   = "LifeExp"
  ) +
  theme_minimal(base_size = 12)
```



## Propuesta de visualización animada al estilo Hans Rosling

### Concepto

#### Bubble chart:

- $x = \text{gdpPercap}$  (escala log),
- $y = \text{lifeExp}$ ,
- $\text{size} = \text{pop}$ ,
- $\text{color} = \text{continent}$ ,
- $\text{frame} = \text{year}$ .

Esto replica el clásico *Gapminder* interactivo. Se puede exportar como GIF (ganimate) o mantenerlo interactivo (plotly).

### Opción A — ganimate (GIF)



```

library(gganimate)

gap_bubbles <- gapminder %>%
  ggplot(aes(
    x = gdpPercap, y = lifeExp,
    size = pop, colour = continent
  )) +
  geom_point(alpha = 0.8, show.legend = FALSE) +
  scale_x_log10(labels = scales::dollar_format()) +
  scale_size(range = c(2, 20), guide = "none") +
  labs(
    title = 'Año: {frame_time}',
    x = 'PIB per cápita (log10 USD)',
    y = 'Esperanza de vida (años)'
  ) +
  theme_minimal() +
  transition_time(year) +
  ease_aes('linear')

animate(gap_bubbles)

```

---

## Opción B — plotly (HTML interactivo)

```

library(plotly)

plot_ly(
  gapminder,
  x = ~gdpPercap, y = ~lifeExp,
  size = ~pop, color = ~continent,
  frame = ~year, text = ~country,
  type = 'scatter', mode = 'markers',
  sizes = c(10, 100),
  marker = list(line = list(width = 0))
) %>%
  layout(
    title = 'Gapminder dinámico',
    xaxis = list(type = 'log', title = 'PIB per cápita (USD, log)'),

```

```
yaxis = list(title = 'Esperanza de vida (años)')
) %>%
animation_opts(
  frame = 1000, easing = "linear", redraw = FALSE
)
```

[illegible]



