Catedra 1

Luis Gutierrez

1- Definición del Problema

Pregunta de Investigación:

1.El Objetivo de la investigacion es verificar si existe alguna diferencia significativa entre la esperanza de vida (lifeExp) entre continentes a o largo del tiempo, ¿como se relaciona esta (esperanza de vida) con el PIB per cápita (gdpPercap) y la población (pop)?

2. Crear un modelo de Machine learnig para predecir la esperanza de vida dado nuevos datos que tratare de recopilar desde el 2007 en adelante, para esto aplicaremos un modelo de randomforest.

Nota: En términos generales, la fórmula del modelo sería similar a una regresión lineal tradicional:

$lifeExp => gdpPercap + pop + continen$

Random Forest no genera una sola fórmula matemática como una regresión lineal. En su lugar, construye muchos árboles de decisión que hacen predicciones y combina sus resultados (por promedio) para obtener el valor final.

Variables Cuantitativas:

- lifeExp: esperanza de vida
- gdpPercap: PIB per cápita
- pop: población total

Variables Cualitativas:

- continet: continente
- country: país

Parámetros a estimar:

Promedio de esperanza de vida y PIB per cápita por continente y año.

2- Introducción

El conjunto de datos para el estudio fueron obtenidos de una libreria de R llamada gapminder, los datos contienen información socioeconómica de diversos países del mundo entre 1952 y 2007. Las variables incluyen esperanza de vida, población y PIB per cápita y agrupadas por país y año.

Descripción de los datos:

1.Cargar librerias para el estudio

```
library(gapminder)
```

```
Warning: package 'gapminder' was built under R version 4.2.3
```

```
library(dplyr)
```

```
Warning: package 'dplyr' was built under R version 4.2.3
```

```
Attaching package: 'dplyr'
```

```
The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(tidyr)
```

```
Warning: package 'tidyr' was built under R version 4.2.3
```

```
library(stringr)
```

```
Warning: package 'stringr' was built under R version 4.2.3
```

```
library(forcats)
library(randomForest)
```

```
Warning: package 'randomForest' was built under R version 4.2.3

randomForest 4.7-1.1

Type rfNews() to see new features/changes/bug fixes.


Attaching package: 'randomForest'

The following object is masked from 'package:ggplot2':

    margin

The following object is masked from 'package:dplyr':

    combine
```

2. Dimensión de los datos

```r
# Saber la dimension del dataset con dim()
dim(gapminder)
```

```
[1] 1704    6
```

```r
# Usando otra libreria llamada dplyr y la funcion glimse
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, ~
```

```r
# Utilizando funciones que retornan filas y columnas
nrow(gapminder) # Número de filas
```

```
[1] 1704
```

```r
ncol(gapminder) # Número de columnas
```

```
[1] 6
```

3. Periodo

```r
# Utilizar la funcion unique que es de los paquetes base de R
unique(gapminder$year)
```

```
 [1] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 2002 2007
```

4. Cobertura de países

```r
# Contar la cantidad de paises de la columna country, de manera unica
n_distinct(gapminder$country)
```

```
[1] 142
```

Variables clave:

• country: nombre del país • continent: continente • year: año • lifeExp: esperanza de vida • pop: población total • gdpPercap: PIB per cápita

No hay datos faltantes en este dataset

```r
# saber si hay nulos en el datat la funcion retorna True o False
any(is.na(gapminder))
```

```
[1] FALSE
```

Nombres de columnas del dataset

```r
names(gapminder)
```

```
[1] "country"   "continent" "year"      "lifeExp"   "pop"       "gdpPercap"
```

3- Preprocesamiento de datos

Para este proceso crearemos una funcion que sirva para cargar el dataset y realizar limpieza del mismo a pesar de que no hay valores NaN

```r
# Función para cargar y limpiar el dataset
cargar_y_limpiar <- function() {
df <- gapminder %>%
filter(!is.na(lifeExp), !is.na(gdpPercap), !is.na(pop)) %>%
mutate(country = str_trim(country))
return(df)
}
```

```r
datos <- cargar_y_limpiar()
glimpse(datos)
```

```
Rows: 1,704
Columns: 6
$ country   <chr> "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, ~
```

4- Análisis Exploratorio de Datos (EDA)

A continuación crearemos 2 funciones una para variables cualitativas y otra para variables cuantiativas y asi poder aplicar un analisis descriptivo a cada una de las variables (atributos), la idea es que esta función reciba la columna a estudiar y me retorne graficos y otros estadisticos.

Función para variables cualitativas

```r
eda_cualitativa <- function(df, col_name) {
col <- df[[col_name]]
cat("=== Análisis EDA Cualitativo para:", col_name, "===\n")
# Tipo de dato
cat("\n Tipo de dato:\n")
print(class(col))
# Valores nulos
cat("\n Valores nulos:\n")
na_count <- sum(is.na(col))
cat("Cantidad:", na_count, "\n")
cat("Porcentaje:", round(100 * na_count / length(col), 2), "%\n")
# Valores únicos
cat("\n Valores únicos:\n")
print(length(unique(col)))
# Duplicados
cat("\n Registros duplicados en la columna:\n")
print(sum(duplicated(col)))
# Frecuencias
```

```
cat("\n Frecuencias de categorías:\n")
print(sort(table(col), decreasing = TRUE))
cat("\n Top 10 categorías:\n")
print(head(sort(table(col), decreasing = TRUE), 10))
# Gráfico de barras
print(
ggplot(df %>% count(!!sym(col_name)) %>%
mutate(!!sym(col_name) := fct_reorder(!!sym(col_name), n)),
aes(x = !!sym(col_name), y = n)) +
geom_col(fill = "coral") +
coord_flip() +
labs(title = paste("Frecuencia de categorías en", col_name),
x = col_name, y = "Frecuencia") +
theme_minimal()
)
}
```

Función para variables cuantitativa

```
eda_cuantitativa <- function(df, col_name) {
col <- df[[col_name]]
cat("=== Análisis EDA Cuantitativo para:", col_name, "===\n")
# Tipo de dato
cat("\n Tipo de dato:\n")
print(class(col))
# Valores nulos
cat("\n Valores nulos:\n")
na_count <- sum(is.na(col))
cat("Cantidad:", na_count, "\n")
cat("Porcentaje:", round(100 * na_count / length(col), 2), "%\n")
# Valores únicos
cat("\n Valores únicos:\n")
cat("\n Registros duplicados en la columna:\n")
print(sum(duplicated(col)))
# Estadísticas
cat("\n Estadísticas descriptivas:\n")
print(summary(col))
cat("\n Percentiles (5, 25, 50, 75, 95):\n")
print(quantile(col, probs = c(0.05, 0.25, 0.5, 0.75, 0.95), na.rm = TRUE))
# Histograma
```

```
print(
ggplot(df, aes(x = !!sym(col_name))) +
geom_histogram(bins = 30, fill = "skyblue", color = "black") +
theme_minimal() +
labs(title = paste("Histograma de", col_name), x = col_name)
)
# Boxplot
print(
ggplot(df, aes(y = !!sym(col_name))) +
geom_boxplot(fill = "lightgreen") +
theme_minimal() +
labs(title = paste("Boxplot de", col_name), y = col_name)
)
}
```

la siguiente función es la que deriva a las 2 funciones anteriores dependiendo del tipo de

variable

```
eda_columna <- function(df, col_name) {
col <- df[[col_name]]
if (is.numeric(col)) {
eda_cuantitativa(df, col_name)
} else if (is.character(col) || is.factor(col)) {
eda_cualitativa(df, col_name)
} else if (inherits(col, "Date") || inherits(col, "POSIXct")){
  cat("=== Análisis EDA para:", col_name, "(Fecha) ===\n")
cat("\n Tipo de dato:\n")
print(class(col))
cat("\n Valores nulos:\n")
na_count <- sum(is.na(col))
cat("Cantidad:", na_count, "\n")
cat("Porcentaje:", round(100 * na_count / length(col), 2), "%\n")
cat("\n Rango de fechas:\n")
print(range(col, na.rm = TRUE))
print(
ggplot(df, aes(x = !!sym(col_name))) +
geom_histogram(bins = 30, fill = "gray") +
theme_minimal() +
labs(title = paste("Distribución temporal de", col_name), x = col_name)
```

```
  )
  } else {
  cat("\n Tipo de dato no soportado actualmente.\n")
  }
  }
```

Usar la funcion en una variable cualitata:

la función a la columna "country" (nombre del país)

En este ejemplo la columna nombre pais tiene 12 repeticiones de cada pais dado que son 12 años por lo cual la mayoria de los graficos y estadisticos no entregan mucha información, veremos como se comportan las otras columnas, ademas el grafico de barra final esta sobrepoblado dado que son muchos paises y todos se repiten 12 veces.

```
  # Para las variables categoricas (cualitivativas) las debo transformar
  datos$country <- as.factor(datos$country) # para probar categórica

  # Luego hacemos un llamado a la funcion eda columna
  eda_columna(datos, "country") # Variable categór
```

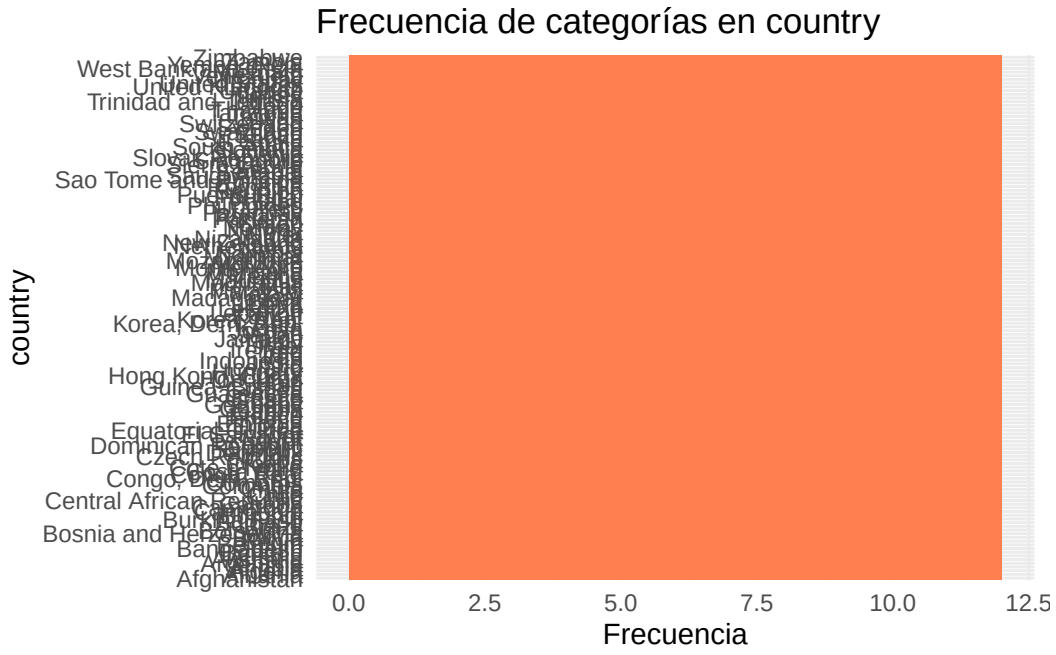
```
=== Análisis EDA Cualitativo para: country ===

 Tipo de dato:
[1] "factor"

 Valores nulos:
Cantidad: 0
Porcentaje: 0 %

 Valores únicos:
[1] 142

 Registros duplicados en la columna:
[1] 1562

 Frecuencias de categorías:
col
             Afghanistan                  Albania                  Algeria
                      12                       12                       12
                  Angola                Argentina                Australia
```

12	12	12
Austria	Bahrain	Bangladesh
12	12	12
Belgium	Benin	Bolivia
12	12	12
Bosnia and Herzegovina	Botswana	Brazil
12	12	12
Bulgaria	Burkina Faso	Burundi
12	12	12
Cambodia	Cameroon	Canada
12	12	12
Central African Republic	Chad	Chile
12	12	12
China	Colombia	Comoros
12	12	12
Congo, Dem. Rep.	Congo, Rep.	Costa Rica
12	12	12
Cote d'Ivoire	Croatia	Cuba
12	12	12
Czech Republic	Denmark	Djibouti
12	12	12
Dominican Republic	Ecuador	Egypt
12	12	12
El Salvador	Equatorial Guinea	Eritrea
12	12	12
Ethiopia	Finland	France
12	12	12
Gabon	Gambia	Germany
12	12	12
Ghana	Greece	Guatemala
12	12	12
Guinea	Guinea-Bissau	Haiti
12	12	12
Honduras	Hong Kong, China	Hungary
12	12	12
Iceland	India	Indonesia
12	12	12
Iran	Iraq	Ireland
12	12	12
Israel	Italy	Jamaica
12	12	12
Japan	Jordan	Kenya
12	12	12

Korea, Dem. Rep.	Korea, Rep.	Kuwait
12	12	12
Lebanon	Lesotho	Liberia
12	12	12
Libya	Madagascar	Malawi
12	12	12
Malaysia	Mali	Mauritania
12	12	12
Mauritius	Mexico	Mongolia
12	12	12
Montenegro	Morocco	Mozambique
12	12	12
Myanmar	Namibia	Nepal
12	12	12
Netherlands	New Zealand	Nicaragua
12	12	12
Niger	Nigeria	Norway
12	12	12
Oman	Pakistan	Panama
12	12	12
Paraguay	Peru	Philippines
12	12	12
Poland	Portugal	Puerto Rico
12	12	12
Reunion	Romania	Rwanda
12	12	12
Sao Tome and Principe	Saudi Arabia	Senegal
12	12	12
Serbia	Sierra Leone	Singapore
12	12	12
Slovak Republic	Slovenia	Somalia
12	12	12
South Africa	Spain	Sri Lanka
12	12	12
Sudan	Swaziland	Sweden
12	12	12
Switzerland	Syria	Taiwan
12	12	12
Tanzania	Thailand	Togo
12	12	12
Trinidad and Tobago	Tunisia	Turkey
12	12	12
Uganda	United Kingdom	United States

```
                  12                   12                   12
             Uruguay              Venezuela              Vietnam
                  12                   12                   12
       West Bank and Gaza         Yemen, Rep.               Zambia
                  12                   12                   12
            Zimbabwe
                  12
```

```
 Top 10 categorías:
col
Afghanistan     Albania     Algeria      Angola   Argentina   Australia
         12          12          12          12          12          12
    Austria     Bahrain  Bangladesh     Belgium
         12          12          12          12
```

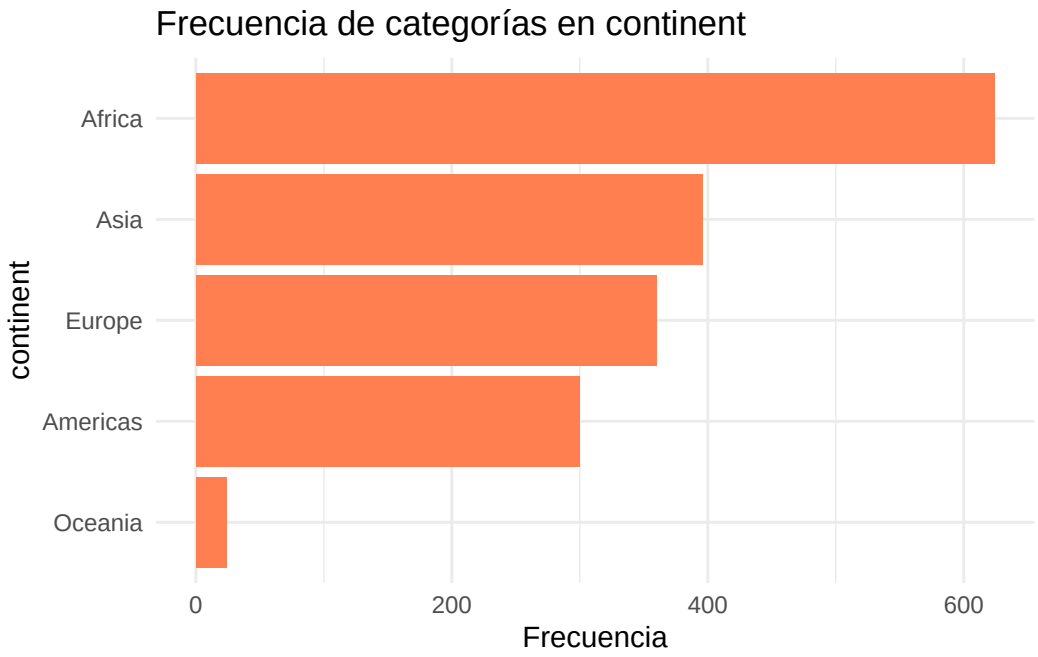


Aplicar la función a la columna "continent" (continente)

En esta columna o atributo se tendra la frecuencia segun la cantidad de paises que existe por continente y los años de estudio que son 12 años, ejemplo Africa tiene 52 paises, por lo cual su frencuencia de datos seria 52*12=624 lo cual se muestra en el grafico.

```r
# Para las variables categoricas (cualitivativas) las debo transformar
datos$continent <- as.factor(datos$continent) # para probar categórica
# Luego hacemos un llamado a la funcion eda columna
eda_columna(datos, "continent") # Variable categórica
```

```
=== Análisis EDA Cualitativo para: continent ===

 Tipo de dato:
[1] "factor"

 Valores nulos:
Cantidad: 0
Porcentaje: 0 %

 Valores únicos:
[1] 5

 Registros duplicados en la columna:
[1] 1699

 Frecuencias de categorías:
col
  Africa     Asia   Europe Americas  Oceania
     624      396      360      300       24

 Top 10 categorías:
col
  Africa     Asia   Europe Americas  Oceania
     624      396      360      300       24
```

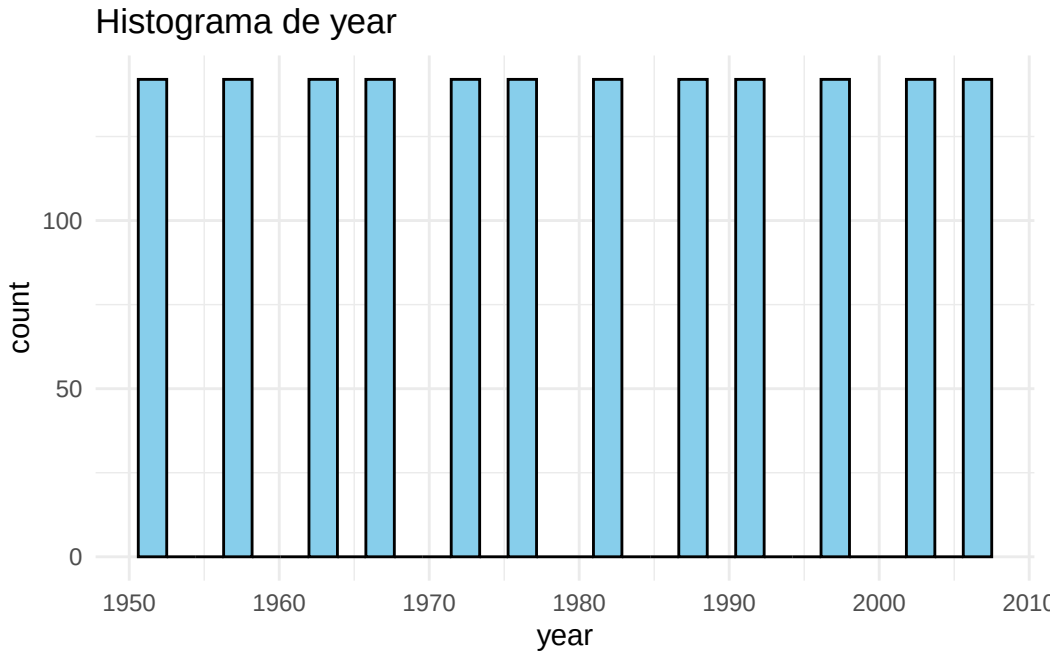


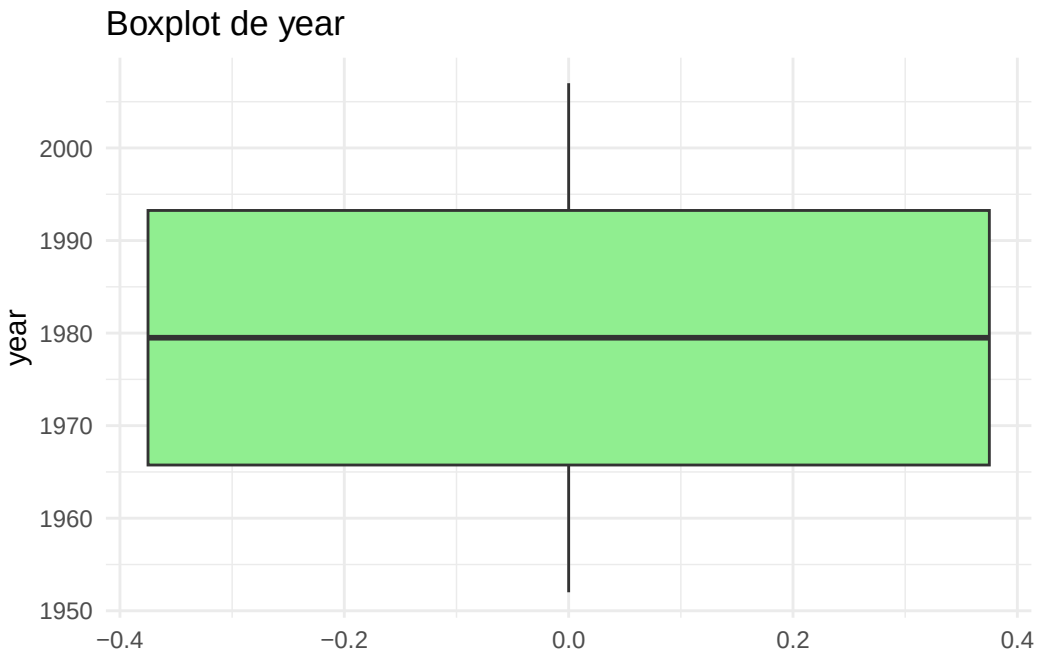
Aplicar la función a la columna “year” (año)

```
# En el caso de variables numericas solo llamamos a la funcion sin transformar nada
# Luego hacemos un llamado a la funcion eda columna
eda_columna(datos, "year")
```

```
=== Análisis EDA Cuantitativo para: year ===

 Tipo de dato:
[1] "integer"

 Valores nulos:
Cantidad: 0
Porcentaje: 0 %

 Valores únicos:

 Registros duplicados en la columna:
[1] 1692
```

```
Estadísticas descriptivas:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1952    1966    1980    1980    1993    2007

Percentiles (5, 25, 50, 75, 95):
     5%     25%     50%     75%     95%
1952.00 1965.75 1979.50 1993.25 2007.00
```



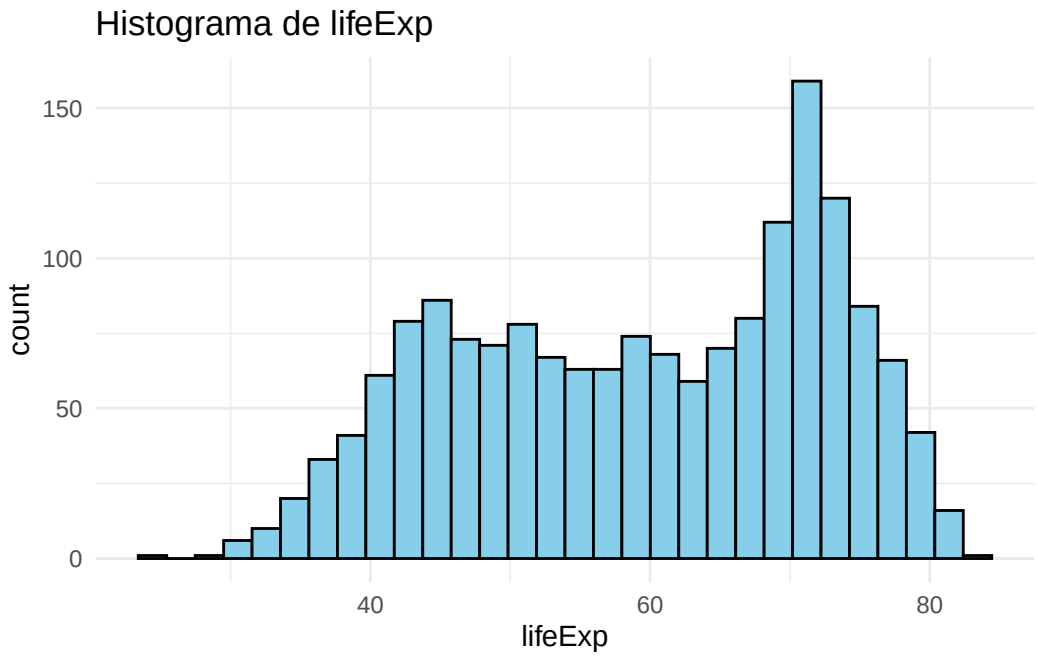


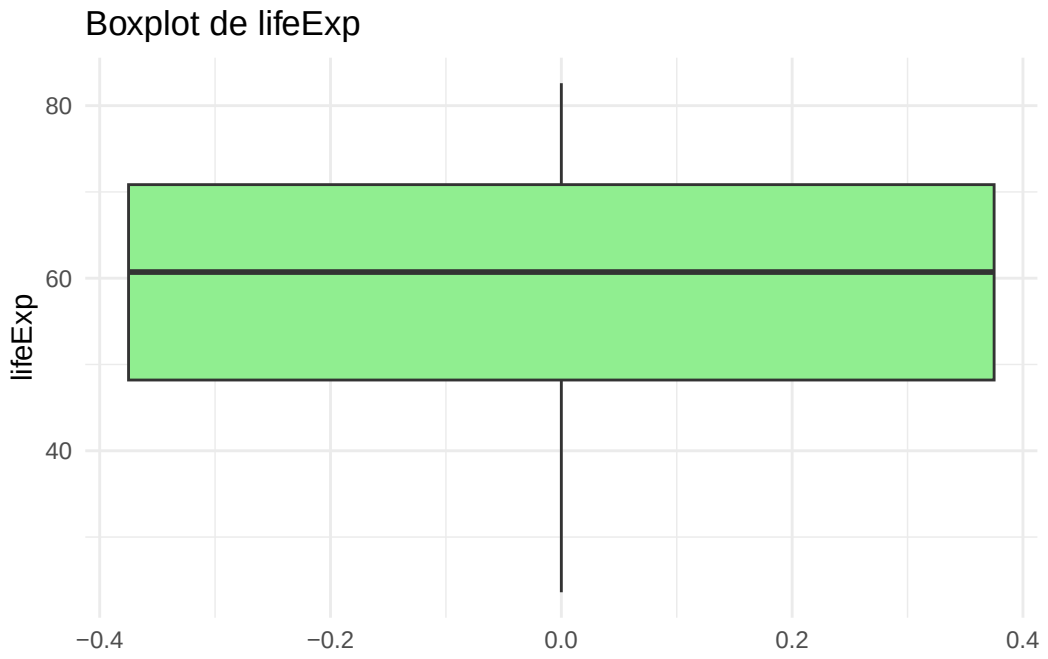
Aplicar la función a la columna "lifeExp" (esperanza de vida)

```
# En el caso de variables numericas solo llamamos a la funcion sin transformar nada
# Luego hacemos un llamado a la funcion eda columna
eda_columna(datos, "lifeExp")
```

```
=== Análisis EDA Cuantitativo para: lifeExp ===

 Tipo de dato:
[1] "numeric"

 Valores nulos:
Cantidad: 0
Porcentaje: 0 %

 Valores únicos:

 Registros duplicados en la columna:
[1] 78

 Estadísticas descriptivas:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
```

```
  23.60   48.20   60.71   59.47   70.85   82.60

 Percentiles (5, 25, 50, 75, 95):
     5%     25%     50%     75%     95%
38.4924 48.1980 60.7125 70.8455 77.4370
```



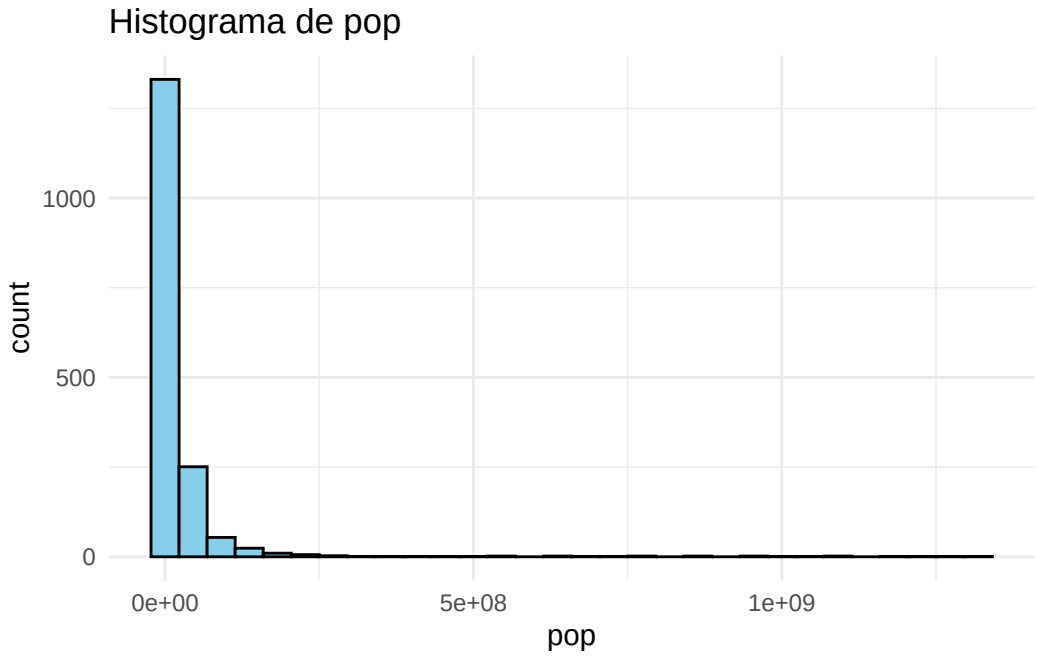


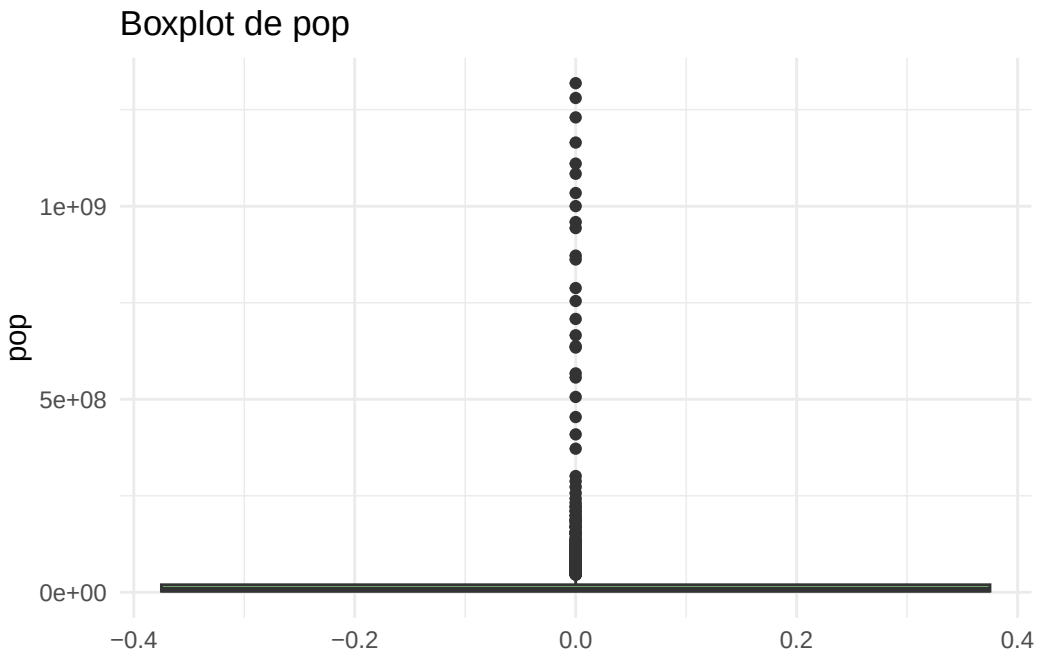
Aplicar la función a la columna “pop” (población total)

```
# En el caso de variables numericas solo llamamos a la funcion sin transformar nada
# Luego hacemos un llamado a la funcion eda columna
eda_columna(datos, "pop")
```

```
=== Análisis EDA Cuantitativo para: pop ===

 Tipo de dato:
[1] "integer"

 Valores nulos:
Cantidad: 0
Porcentaje: 0 %

 Valores únicos:

 Registros duplicados en la columna:
[1] 0
```

```
 Estadísticas descriptivas:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
6.001e+04 2.794e+06 7.024e+06 2.960e+07 1.959e+07 1.319e+09

 Percentiles (5, 25, 50, 75, 95):
       5%        25%        50%        75%        95%
  475458.9  2793664.0  7023595.5 19585221.8 89822054.5
```



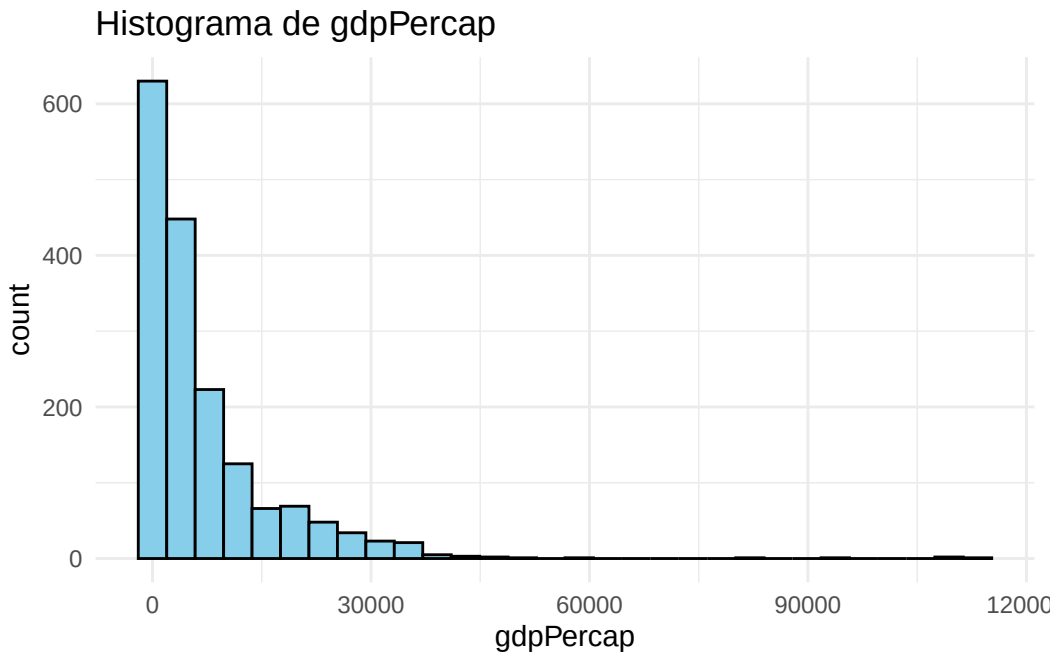


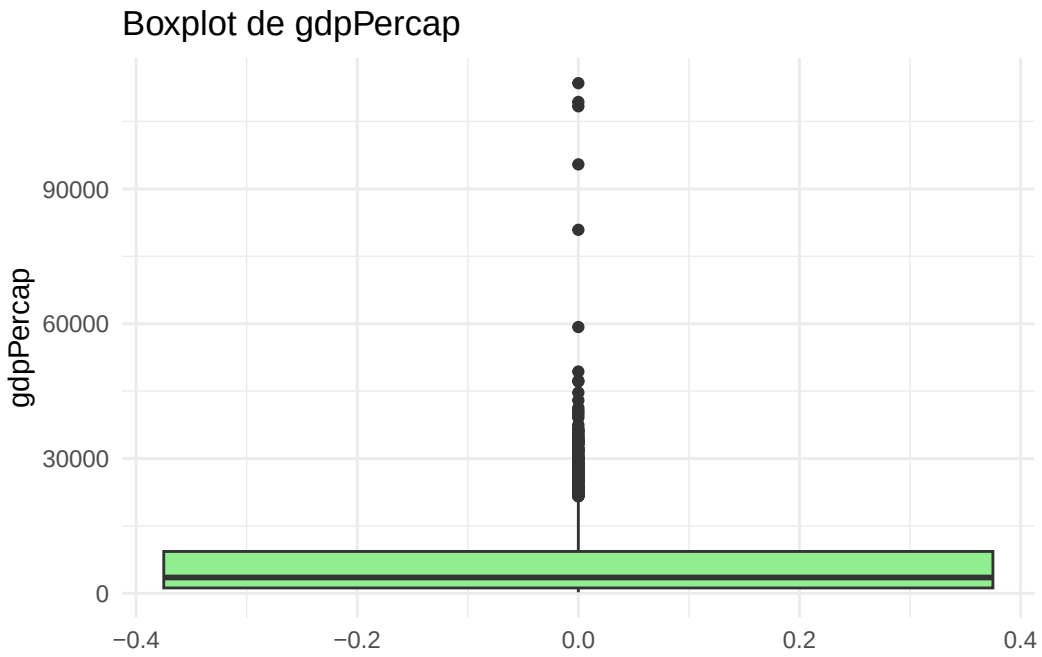
Aplicar la función a la columna "gdpPercap" (PIB per cápita)

```
# En el caso de variables numericas solo llamamos a la funcion sin transformar nada
# Luego hacemos un llamado a la funcion eda columna
eda_columna(datos, "gdpPercap")
```

```
=== Análisis EDA Cuantitativo para: gdpPercap ===

 Tipo de dato:
[1] "numeric"

 Valores nulos:
Cantidad: 0
Porcentaje: 0 %

 Valores únicos:

 Registros duplicados en la columna:
[1] 0

 Estadísticas descriptivas:
   Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
```

```
  241.2   1202.1   3531.8   7215.3   9325.5 113523.1

Percentiles (5, 25, 50, 75, 95):
       5%        25%        50%        75%        95%
 547.9964  1202.0603  3531.8470  9325.4623 26608.3333
```

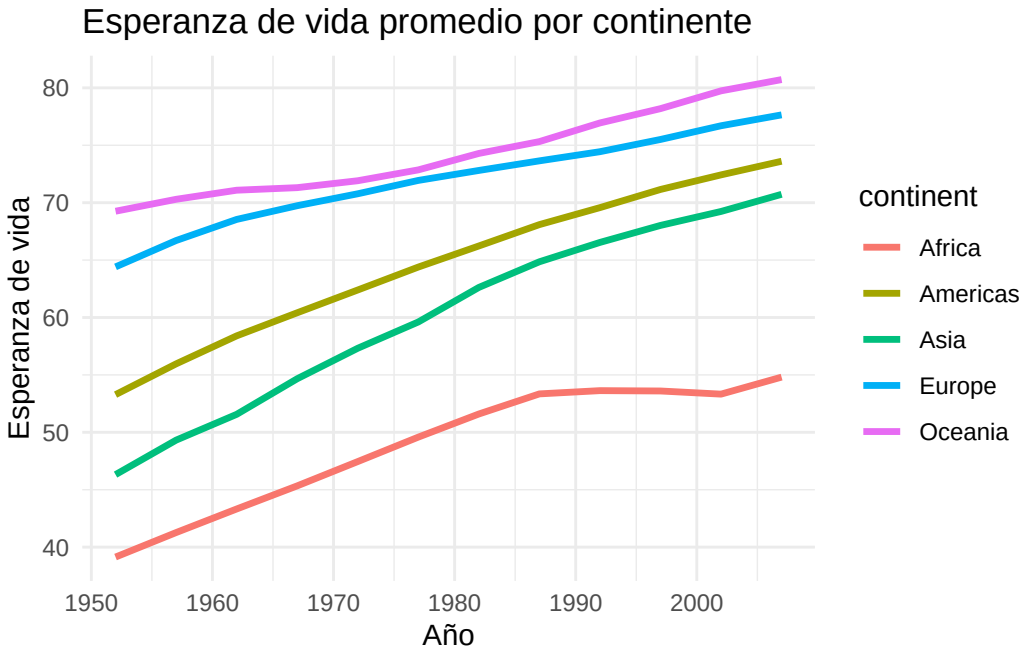




4.1 Esperanza de vida por continente

```
# Calcular promedio por año y continente
vida_media <- datos %>%
group_by(continent, year) %>%
summarise(lifeExp_mean = mean(lifeExp), .groups = "drop")
# Gráfico con ggplot2 clásico
ggplot(vida_media, aes(x = year, y = lifeExp_mean, color = continent)) +
geom_line(size = 1.2) +
labs(title = "Esperanza de vida promedio por continente",
x = "Año",
y = "Esperanza de vida") +
theme_minimal()
```

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



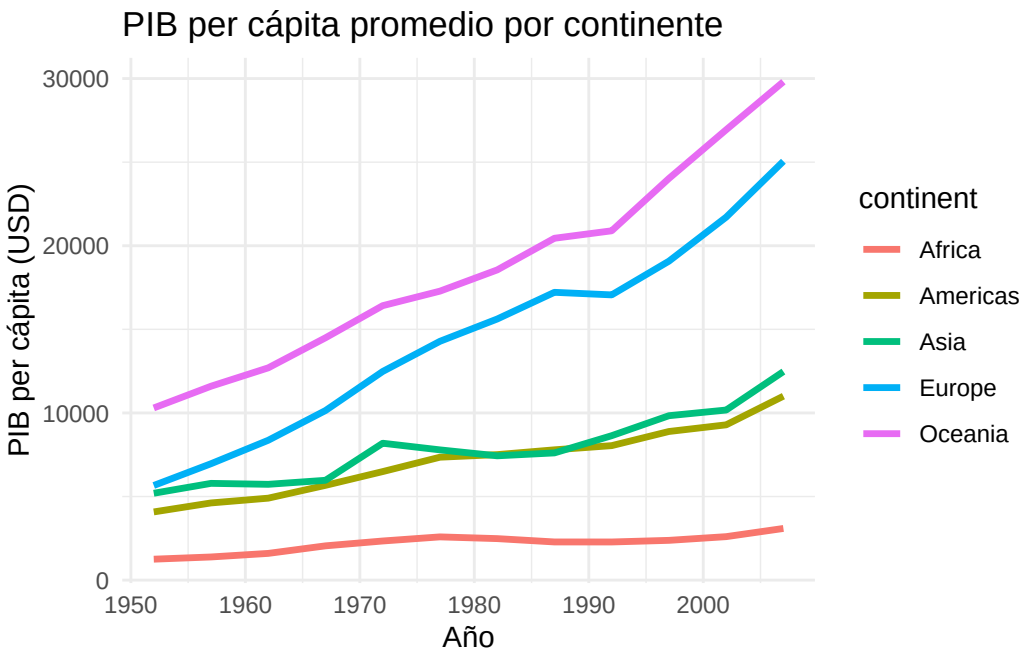
Interpretación:

En este gráfico observamos que la esperanza de vida ha aumentado consistentemente en todos los continentes desde 1952 hasta 2007. • Oceanía y Europa mantienen los niveles más altos de esperanza de vida durante todo el periodo. • África, en cambio, muestra la esperanza de vida más baja y un crecimiento más lento. En algunas décadas incluso se ve una ligera caída, probablemente relacionada con enfermedades o conflictos (como la epidemia de VIH en los 90s). • Esto sugiere una brecha significativa en la salud y el desarrollo humano entre continentes.

4.2 PIB per cápita a través del tiempo

```
pib_media <- datos %>%
group_by(continent, year) %>%
summarise(gdpPercap_mean = mean(gdpPercap), .groups = "drop")
ggplot(pib_media, aes(x = year, y = gdpPercap_mean, color = continent)) +
geom_line(size = 1.2) +
labs(title = "PIB per cápita promedio por continente",
```

```
    x = "Año",
y = "PIB per cápita (USD)") +
theme_minimal()
```

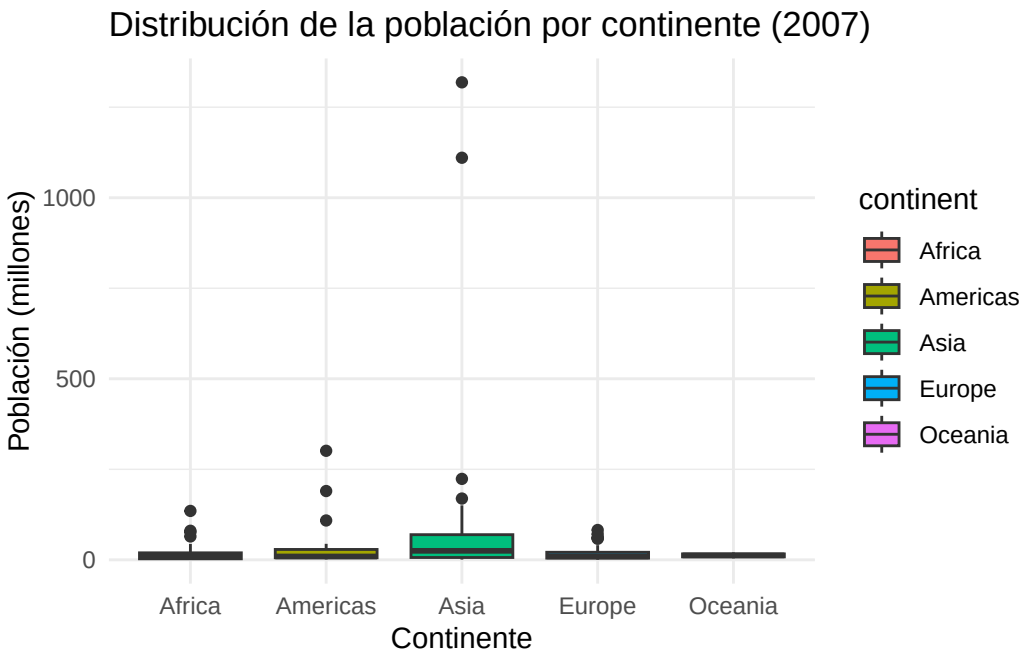


Interpretación:

Este gráfico muestra la evolución del PIB per cápita (una medida del desarrollo económico) por continente. • América y Europa presentan un aumento sostenido, especialmente Europa, que supera ampliamente a los otros continentes hacia 2007. • África mantiene un PIB per cápita mucho menor y con menor crecimiento, evidenciando desigualdades económicas estructurales. • Oceanía, aunque tiene pocos países representados, muestra valores altos y estables. Esto refleja que el crecimiento económico es desigual a nivel global, lo cual puede estar vinculado con diferencias en acceso a educación, salud, comercio y recursos naturales.

4.3 Distribución de la población año 2007

```r
datos_2007 <- filter(datos, year == 2007)
ggplot(datos_2007, aes(x = continent, y = pop / 1e6, fill = continent)) +
geom_boxplot() +
labs(title = "Distribución de la población por continente (2007)",
x = "Continente",
y = "Población (millones)") +
theme_minimal()
```



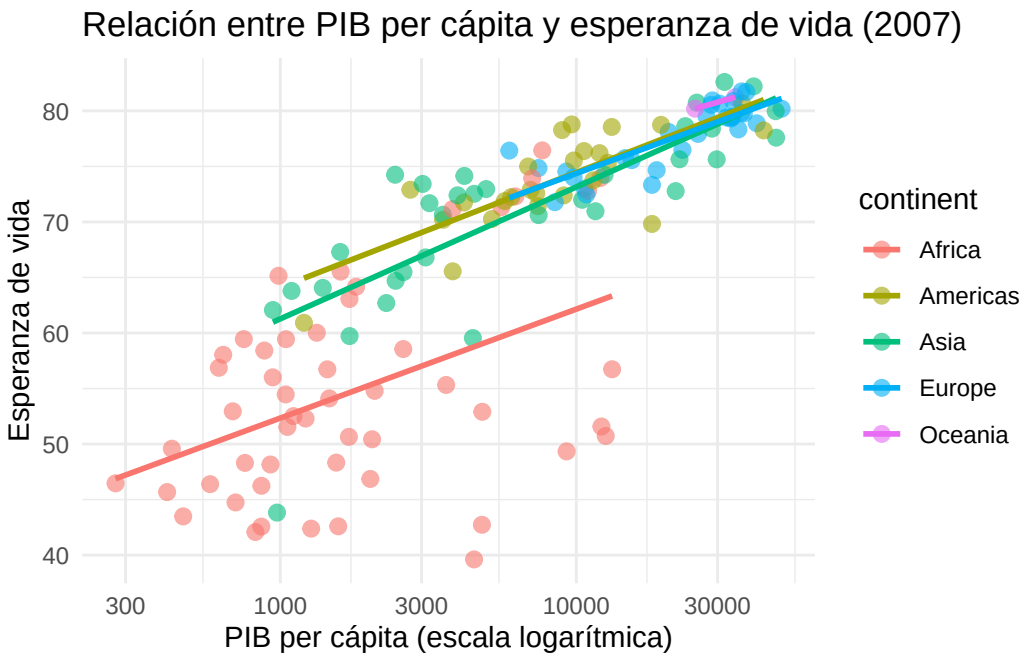
Interpretación:

Este boxplot muestra la distribución de la población por país dentro de cada continente en 2007. • Asia tiene una alta variabilidad poblacional, con algunos países extremadamente poblados como China e India, y otros con mucha menos población. • Europa y América tienen poblaciones más homogéneas entre países. • Oceanía presenta valores bajos, y África muestra también una gran dispersión. Esto refleja que dentro de cada continente hay países con características muy distintas en términos de población, lo que puede tener un impacto en el desarrollo y políticas públicas.

4.4 Relación entre el PIB per cápita y la esperanza de vida

```
ggplot(datos_2007, aes(x = gdpPercap, y = lifeExp, color = continent)) +
geom_point(alpha = 0.6, size = 2.5) +
geom_smooth(method = "lm", se = FALSE, linewidth = 1) +
scale_x_log10() +
labs(title = "Relación entre PIB per cápita y esperanza de vida (2007)",
x = "PIB per cápita (escala logarítmica)",
y = "Esperanza de vida") +
theme_minimal()
```

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



Este gráfico muestra la relación entre el PIB per cápita y la esperanza de vida en el año 2007. • Se observa una fuerte correlación positiva: a mayor PIB per cápita, mayor esperanza de vida. • Los países con bajo PIB per cápita (a la izquierda) tienden a tener menor esperanza de vida, mientras que los países más ricos viven más. • Sin embargo, se nota una desaceleración del crecimiento de la esperanza de vida en países con PIB muy alto, lo que indica un punto de saturación. • También se observan diferencias por continente, mostrando que los países europeos y oceánicos concentran altos valores en ambas variables. Este gráfico respalda la idea de que el desarrollo económico está fuertemente asociado con mejores condiciones de salud y longevidad, pero no es el único factor.

5- Planificación del proyecto de Ciencia de datos

Mi idea es realizar un modelo de machine learnig para predecir una variable del modelo como la esperanza de vida, para este desarrollo utilizaremos el paquete de R llamado RandomForest, si nos da el tiempo del proyecto quizas comparar los resultado con otro modelo.

1. **División de datos en entrenamiento y prueba**

```
# Filtrar datos hasta 2002 para entrenamiento, dejar 2007 para prueba
train_data <- datos %>% filter(year <= 2002)
test_data <- datos %>% filter(year == 2007)

# Verificar dimensiones
dim(train_data)
```

```
[1] 1562    6
```

```
dim(test_data)
```

```
[1] 142   6
```

2. Construcción del modelo Random Forest y evaluacion

```
# Modelo Random Forest con variables seleccionadas
set.seed(123)  # Para reproducibilidad
rf_model <- randomForest(
  lifeExp ~ gdpPercap + pop + continent + year,
  data = train_data,
  ntree = 500,       # Número de árboles
  mtry = 2,          # Variables consideradas en cada división
  importance = TRUE, # Para calcular importancia de variables
  na.action = na.omit,
  keep.forest = TRUE # Para mantener el modelo para predicciones
)
```

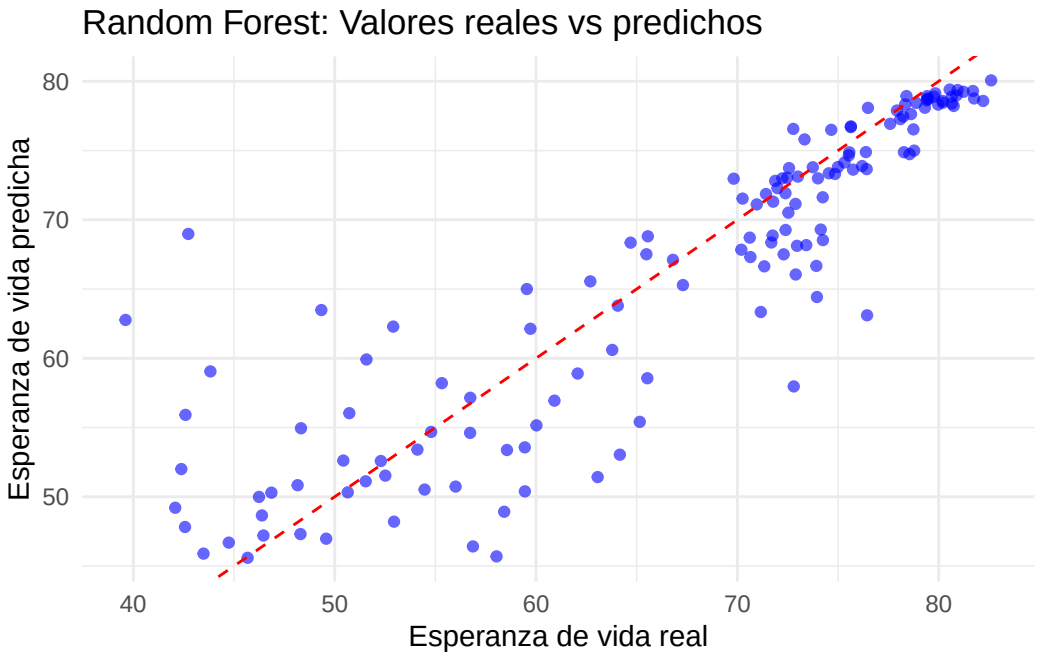
```r
# Ver resumen del modelo
print(rf_model)
```

```
Call:
 randomForest(formula = lifeExp ~ gdpPercap + pop + continent +      year, data = train_data
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 22.9984
                    % Var explained: 85.9
```

```r
# Evaluación del modelo sin caret
evaluar_modelo <- function(modelo, datos_test) {
  pred <- predict(modelo, datos_test)
  real <- datos_test$lifeExp

  # Calcular métricas manualmente
  rmse <- sqrt(mean((pred - real)^2))
  r2 <- 1 - (sum((real-pred)^2)/sum((real-mean(real))^2))
  mae <- mean(abs(pred - real))

  cat("Métricas de evaluación:\n")
  cat("RMSE:", round(rmse, 3), "\n")
  cat("R²:", round(r2, 3), "\n")
  cat("MAE:", round(mae, 3), "\n")

  return(data.frame(Predicho = pred, Real = real))
}

# Evaluar el modelo
resultados <- evaluar_modelo(rf_model, test_data)
```

```
Métricas de evaluación:
RMSE: 5.61
R²: 0.783
MAE: 3.678
```

```r
# Gráfico de valores reales vs predichos
ggplot(resultados, aes(x = Real, y = Predicho)) +
  geom_point(alpha = 0.6, color = "blue") +
  geom_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +
  labs(title = "Random Forest: Valores reales vs predichos",
       x = "Esperanza de vida real",
       y = "Esperanza de vida predicha") +
  theme_minimal()
```




```r
# Optimización de hiperparámetros sin caret
optimizar_rf <- function(train_data, ntree_range = c(500, 1000), mtry_range = 2:4) {
  resultados <- data.frame()

  for (nt in ntree_range) {
    for (mt in mtry_range) {
      set.seed(123)
      modelo <- randomForest(
        lifeExp ~ gdpPercap + pop + continent + year,
        data = train_data,
        ntree = nt,
        mtry = mt
```

```
      )

      pred <- predict(modelo, train_data)
      rmse <- sqrt(mean((pred - train_data$lifeExp)^2))

      resultados <- rbind(resultados, data.frame(ntree = nt, mtry = mt, RMSE = rmse))
    }
  }

  return(resultados)
}

# Ejecutar optimización
resultados_opt <- optimizar_rf(train_data)
print(resultados_opt)
```

```
  ntree mtry     RMSE
1   500    2 2.406958
2   500    3 2.165963
3   500    4 2.119984
4  1000    2 2.403234
5  1000    3 2.168058
6  1000    4 2.123479
```

```
# Seleccionar mejor combinación
mejor_combinacion <- resultados_opt[which.min(resultados_opt$RMSE), ]
cat("\nMejor combinación de parámetros:\n")
```

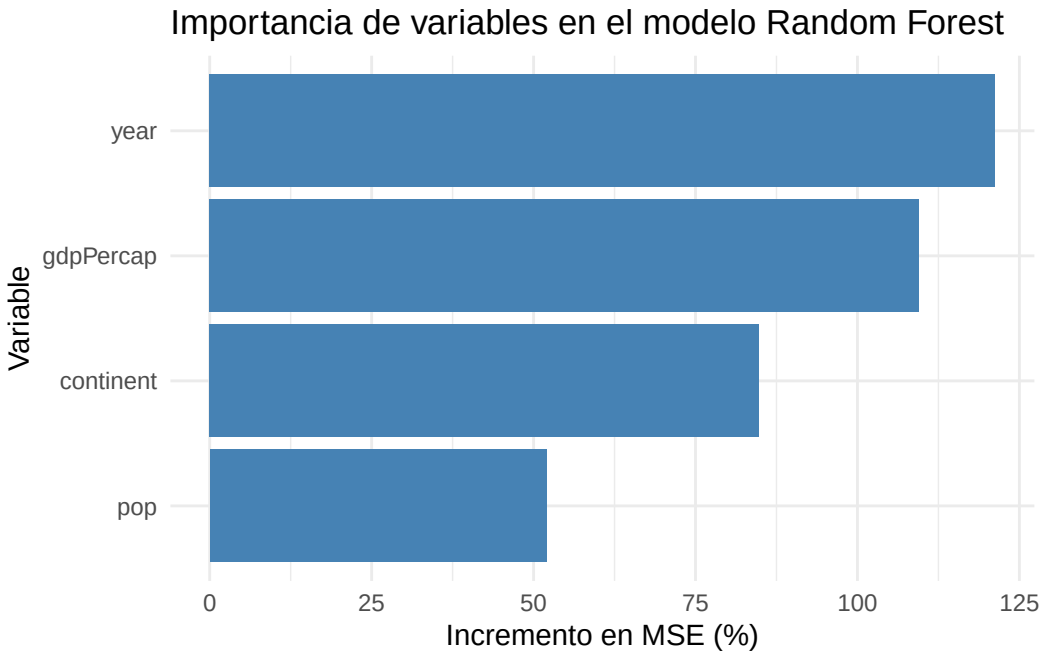
```
Mejor combinación de parámetros:
```

```
print(mejor_combinacion)
```

```
  ntree mtry     RMSE
3   500    4 2.119984
```

3. Importancia de variables

```r
# Importancia de variables
importance_df <- as.data.frame(importance(rf_model))
importance_df$Variable <- rownames(importance_df)

# Gráfico de importancia
ggplot(importance_df, aes(x = reorder(Variable, `%IncMSE`), y = `%IncMSE`)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Importancia de variables en el modelo Random Forest",
       x = "Variable",
       y = "Incremento en MSE (%)") +
  theme_minimal()
```



4. Interpretación de resultados

Los resultados del modelo Random Forest mostrarán:

1. **Exactitud del modelo**: El R^2 y RMSE indicarán cuán bien el modelo predice la esperanza de vida.

2. **Importancia de variables**: El análisis de importancia revelará qué variables (PIB per cápita, población, continente o año) tienen mayor impacto en la predicción.