Análisis Estadístico

Derek Martell

Descargamos librerías y la base de datos ya limpia

Ahora vamos a analizar

```
# Verificar nombres de variables
colnames(df)
```

```
## [1] "Country"       "Index.Year"    "capi"          "art"
## [5] "capcorr"       "labor"         "gdp"           "Overall.Score"
```

```
# Revisar la estructura
glimpse(df)
```

```
## Rows: 135
## Columns: 8
## $ Country       <chr> "Bolivia", "Bolivia", "Bolivia", "Bolivia", "Bolivia", "~
## $ Index.Year    <int> 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 20~
## $ capi          <dbl> 16.17637, 18.96895, 23.15322, 19.09732, 17.88748, 13.926~
## $ art           <dbl> 28.73, 39.32, 39.05, 53.82, 37.82, 42.27, 46.61, 50.80, ~
## $ capcorr       <dbl> 1196557766, 1503429162, 1967443290, 1582225491, 15021646~
## $ labor         <int> 3436412, 3529691, 3599373, 3669870, 3741182, 3814211, 38~
## $ gdp           <dbl> 922.1133, 970.3518, 1022.1451, 979.3445, 975.7771, 930.1~
## $ Overall.Score <dbl> 65.2, 65.1, 68.8, 65.6, 65.0, 68.0, 65.1, 64.3, 64.5, 58~
```

```
# Número de países y años
length(unique(df$Country))
```

```
## [1] 5
```

```
length(unique(df$Index.Year))
```

```
## [1] 27
```

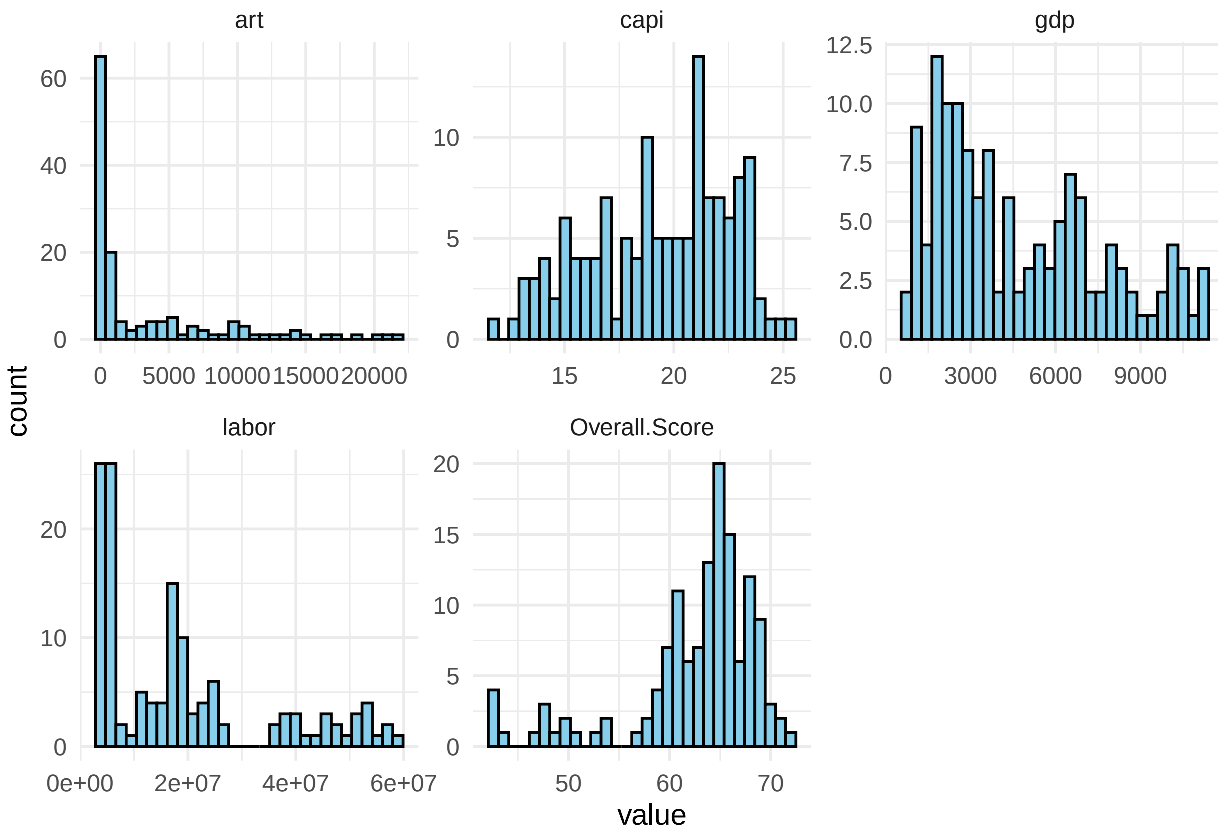
2. Estadísticos descriptivos

```r
describe(df[, c("capi", "art", "labor", "gdp", "Overall.Score")])
```

```
##               vars   n        mean          sd      median     trimmed
## capi             1 135       19.35        3.22       19.91       19.51
## art              2 135     3208.45     5144.17      406.22     2059.98
## labor            3 135 18618090.83 15919779.88 15961805.00 16258772.17
## gdp              4 135     4679.09     2931.20     3689.14     4420.70
## Overall.Score    5 135       62.59        6.40       64.50       63.69
##                       mad        min         max       range  skew kurtosis
## capi                 3.63      11.69       25.30       13.61 -0.36    -0.89
## art                544.37      23.84    21753.88    21730.04  1.84     2.69
## labor         14859147.97 3436412.00 58544385.00 55107973.00  1.07    -0.05
## gdp               2959.97     888.22    11391.38    10503.16  0.63    -0.74
## Overall.Score        4.60      42.30       71.70       29.40 -1.57     2.08
##                        se
## capi                 0.28
## art                442.74
## labor          1370156.50
## gdp                252.28
## Overall.Score        0.55
```

3. Visualización de distribuciones

```r
df %>%
  pivot_longer(cols = c(capi, art, labor, gdp, Overall.Score)) %>%
  ggplot(aes(x = value)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  facet_wrap(~name, scales = "free") +
  theme_minimal()
```



Ralizamos los test de normalidad

```
shapiro.test(df$capi)   # Solo si n < 5000
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$capi
## W = 0.9622, p-value = 0.0008555
```

```
shapiro.test(df$art)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$art
## W = 0.67653, p-value = 7.528e-16
```

```
shapiro.test(df$labor)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$labor
## W = 0.82885, p-value = 3.049e-11
```

```r
shapiro.test(df$gdp)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$gdp
## W = 0.92, p-value = 6.94e-07
```

```r
shapiro.test(df$Overall.Score)
```

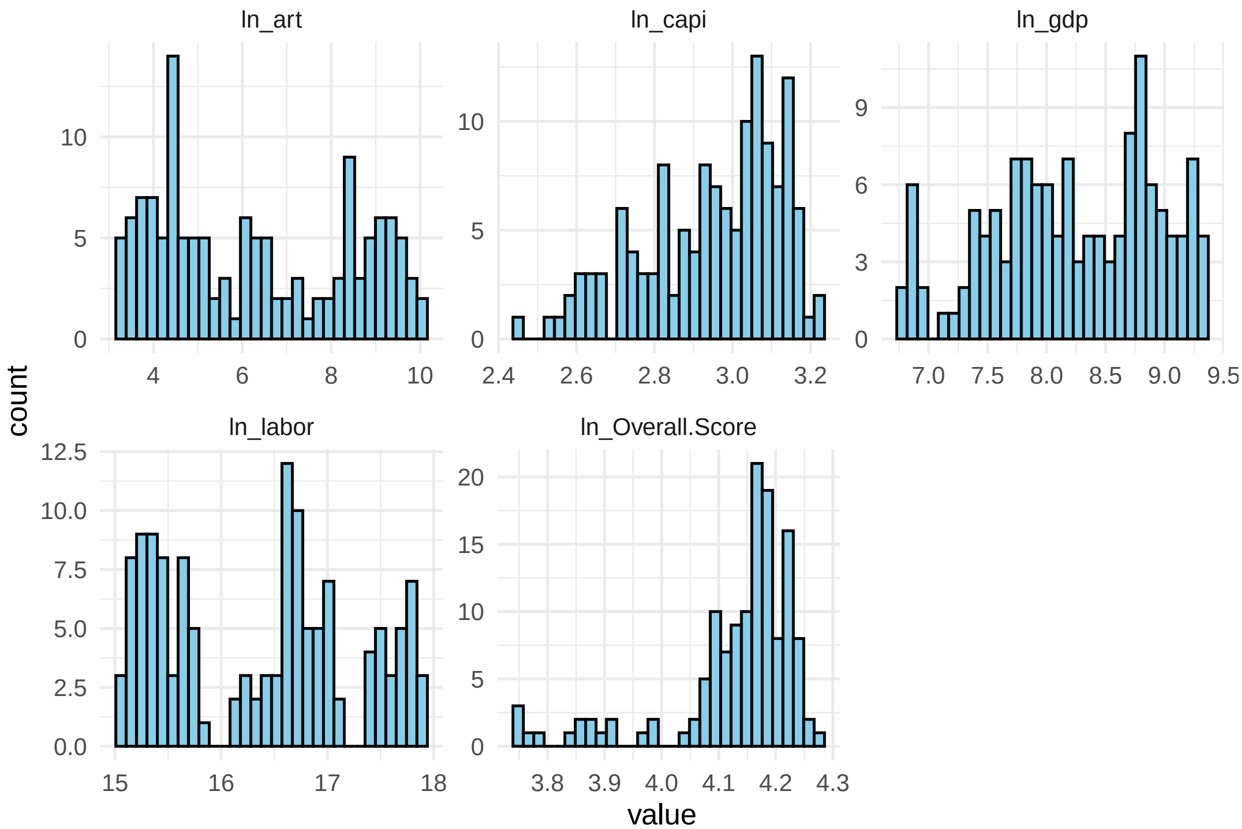
```
##
##  Shapiro-Wilk normality test
##
## data:  df$Overall.Score
## W = 0.83079, p-value = 3.618e-11
```

4. Transformación logarítmica

```r
df <- df %>%
  mutate(
    ln_gdp = log(gdp),
    ln_capi = log(capi),
    ln_art = log(art + 1),      # Se suma 1 en caso de ceros
    ln_labor = log(labor),
    ln_Overall.Score = log(Overall.Score)
  )
```

5. Visualización de distribuciones luego de las transformaciones

```r
df %>%
  pivot_longer(cols = c(ln_capi, ln_art, ln_labor, ln_gdp, ln_Overall.Score)) %>%
  ggplot(aes(x = value)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  facet_wrap(~name, scales = "free") +
  theme_minimal()
```



Realizamos el test de normalidad

```r
shapiro.test(df$ln_capi)  # Solo si n < 5000
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$ln_capi
## W = 0.94238, p-value = 2.188e-05
```

```r
shapiro.test(df$ln_art)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$ln_art
## W = 0.91151, p-value = 2.163e-07
```

```r
shapiro.test(df$ln_labor)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$ln_labor
## W = 0.91721, p-value = 4.698e-07
```

```r
shapiro.test(df$ln_gdp)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$ln_gdp
## W = 0.95942, p-value = 0.0004897
```

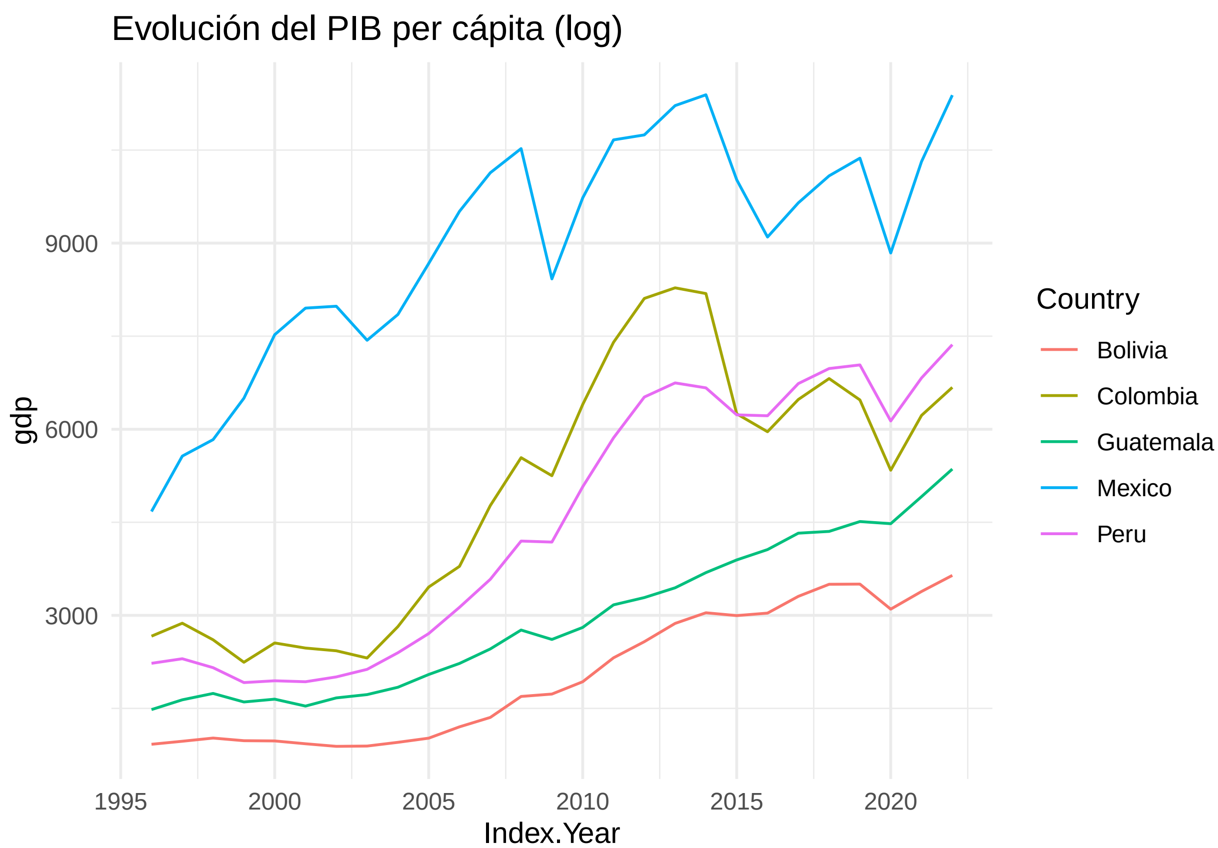
```r
shapiro.test(df$ln_Overall.Score)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$ln_Overall.Score
## W = 0.78169, p-value = 6.705e-13
```

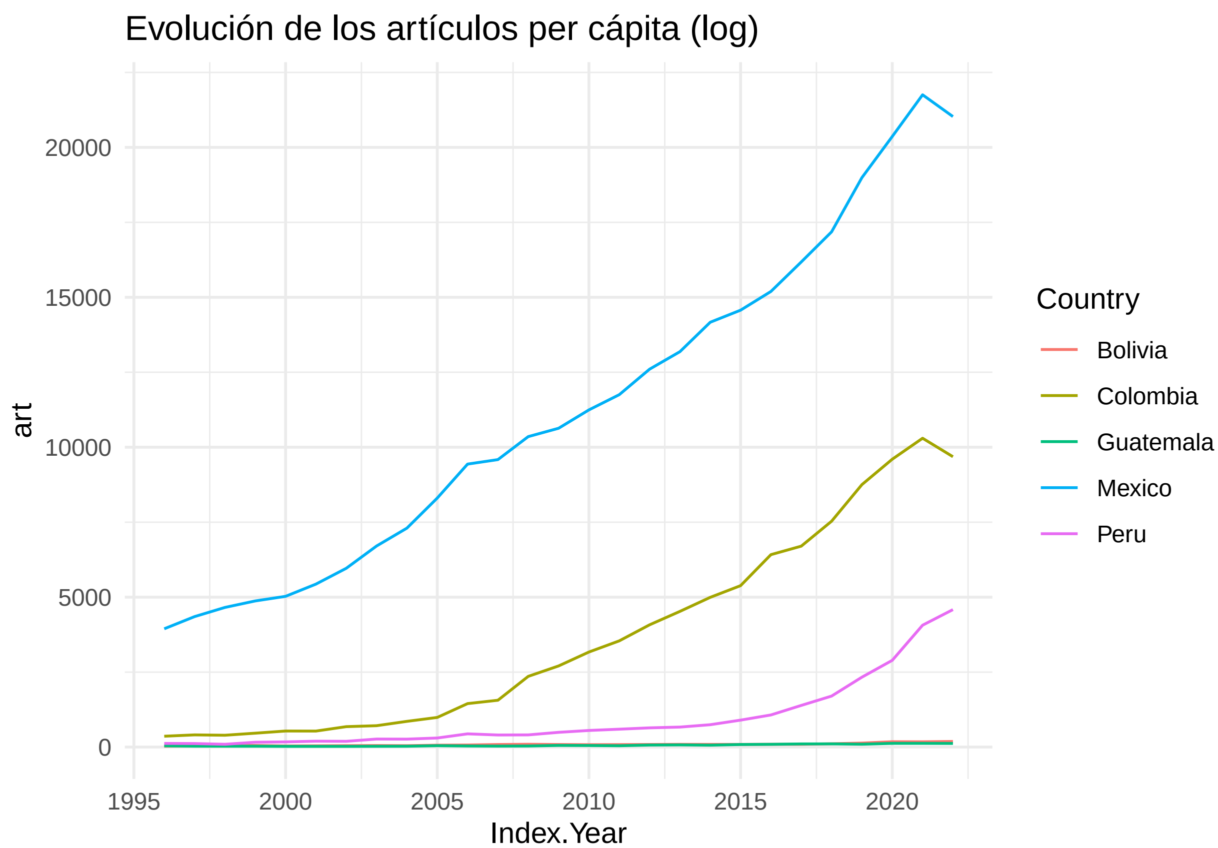
6. Visualización de series de tiempo

6.1. Visualización antes de la transformación

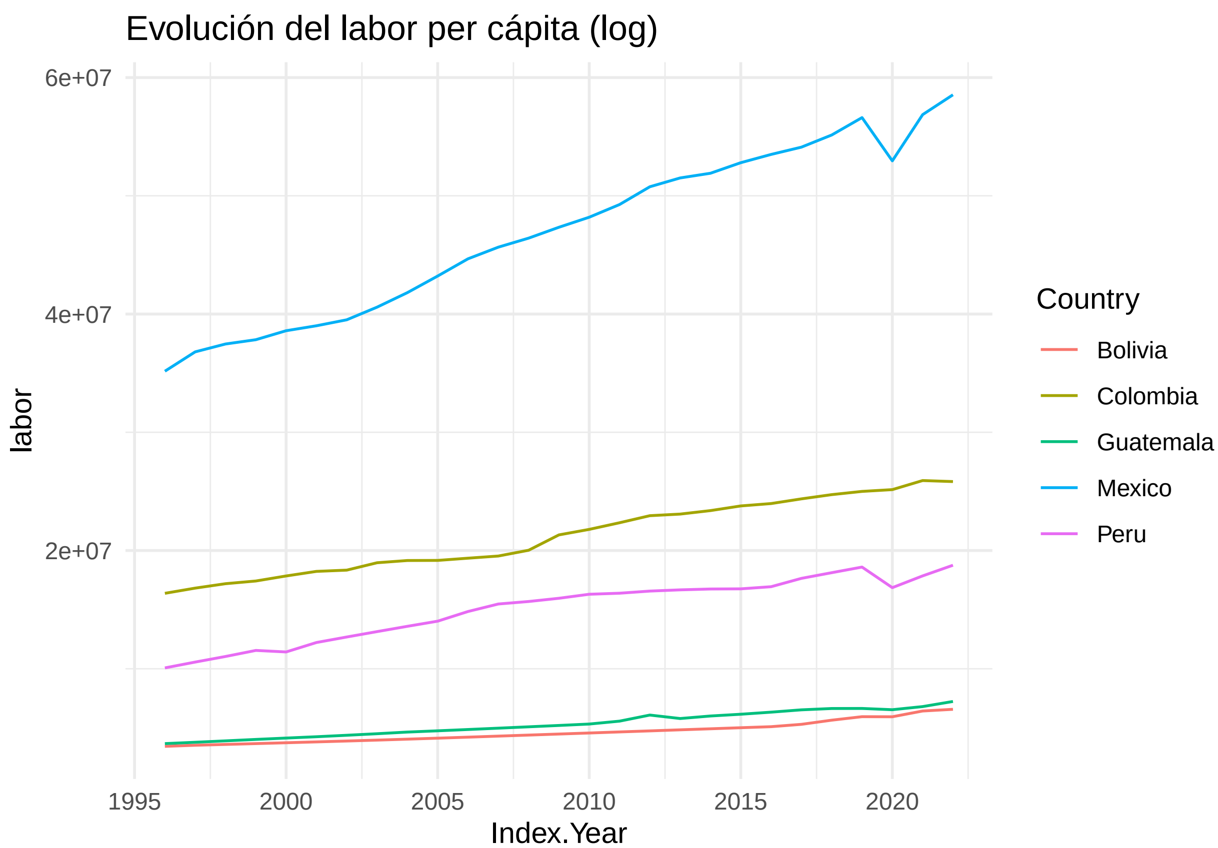
```r
ggplot(df, aes(x = Index.Year, y = gdp, color = Country)) +
  geom_line() +
  labs(title = "Evolución del PIB per cápita (log)") +
  theme_minimal()
```



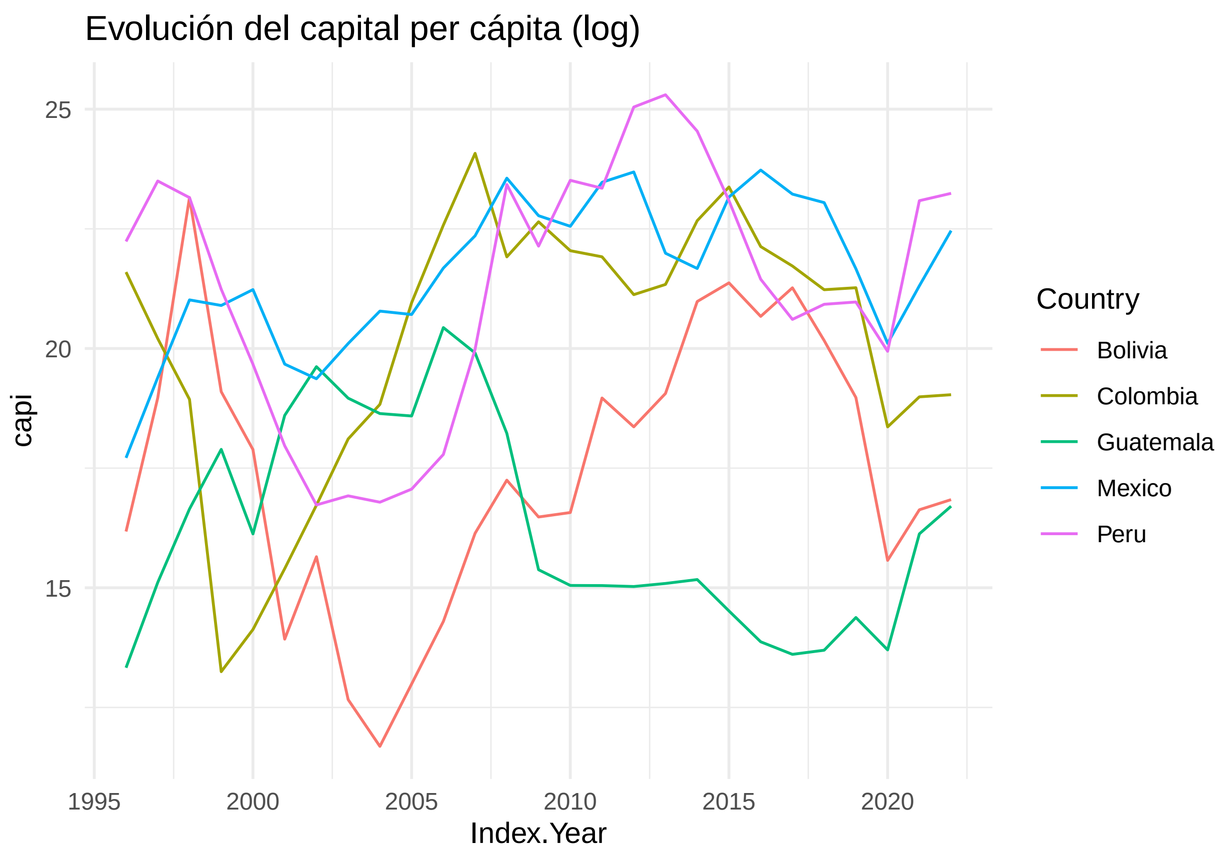

```r
ggplot(df, aes(x = Index.Year, y = art, color = Country)) +
  geom_line() +
  labs(title = "Evolución de los artículos per cápita (log)") +
  theme_minimal()
```



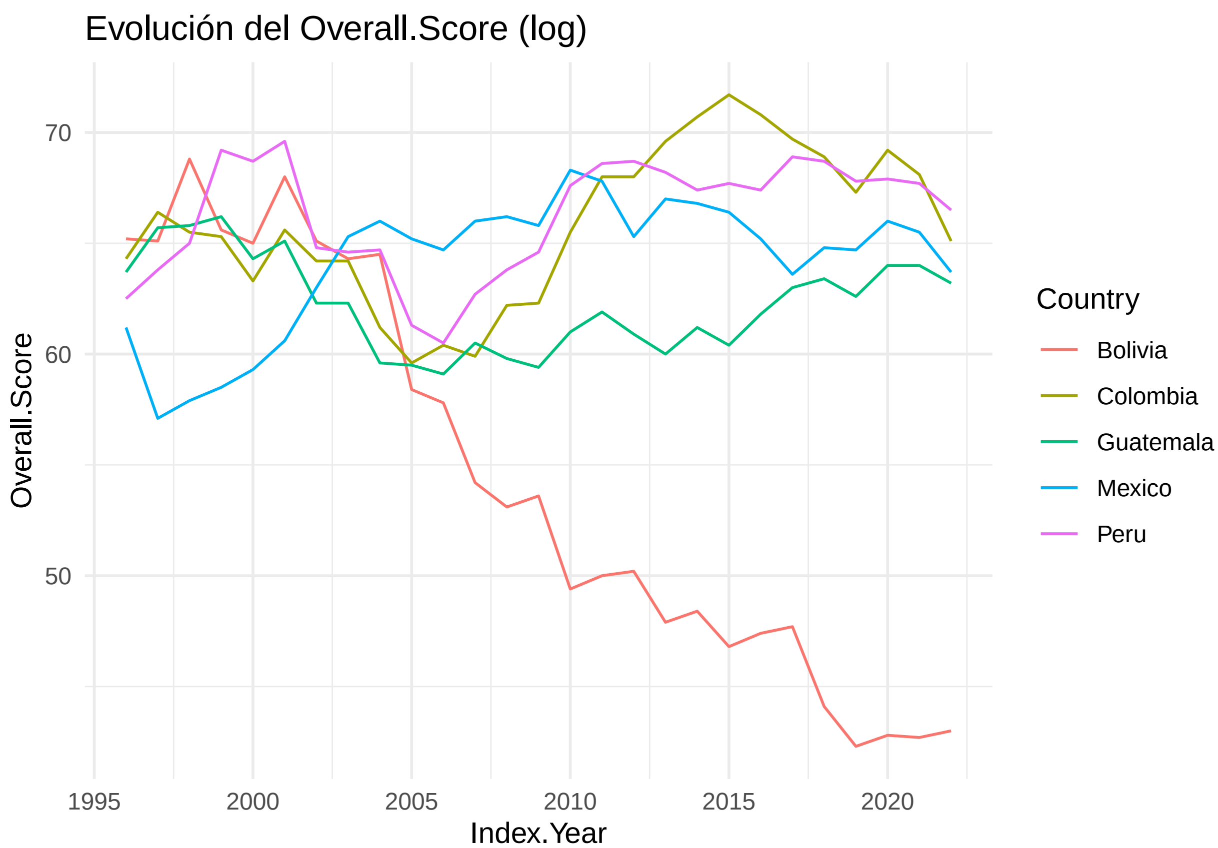

```r
ggplot(df, aes(x = Index.Year, y = labor, color = Country)) +
  geom_line() +
  labs(title = "Evolución del labor per cápita (log)") +
  theme_minimal()
```




```r
ggplot(df, aes(x = Index.Year, y = capi, color = Country)) +
  geom_line() +
  labs(title = "Evolución del capital per cápita (log)") +
  theme_minimal()
```

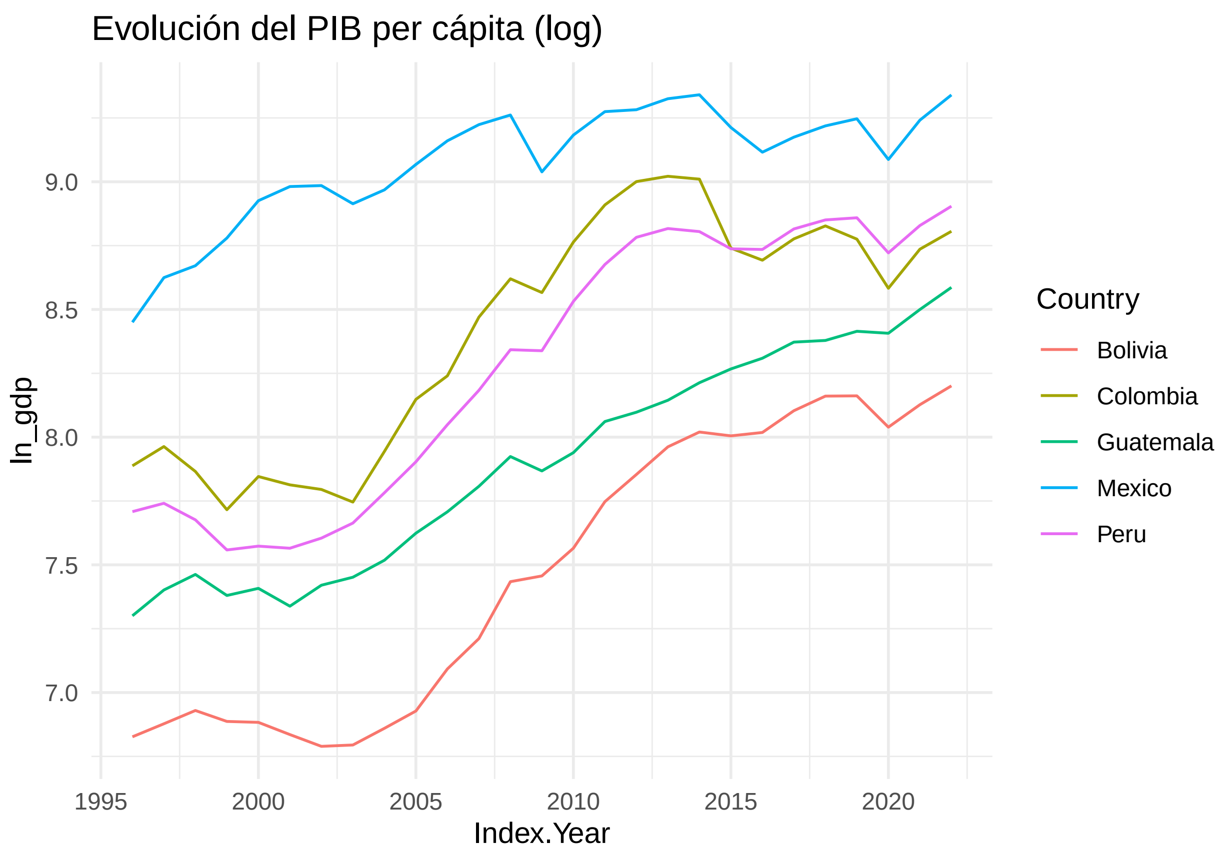



```r
ggplot(df, aes(x = Index.Year, y = Overall.Score, color = Country)) +
  geom_line() +
  labs(title = "Evolución del Overall.Score (log)") +
  theme_minimal()
```

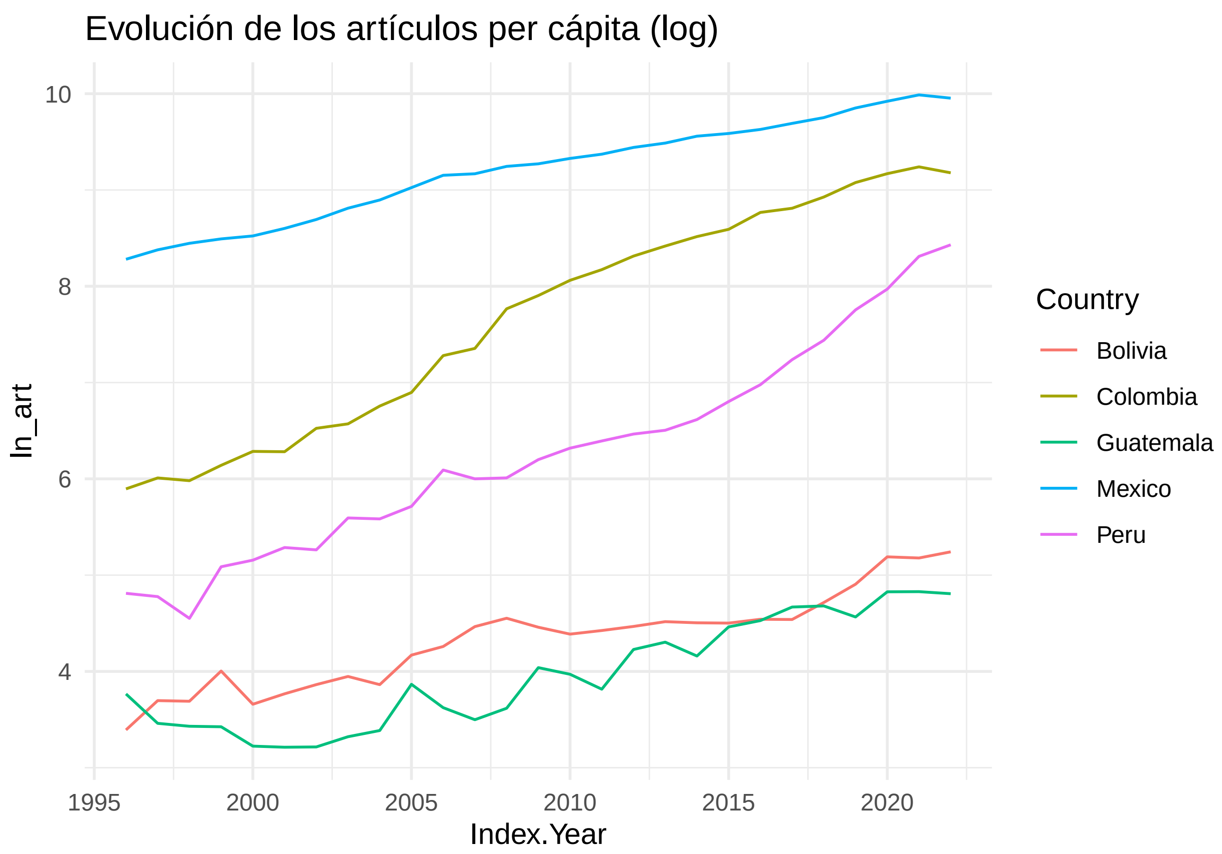


6.2. Visualización después de la transformación

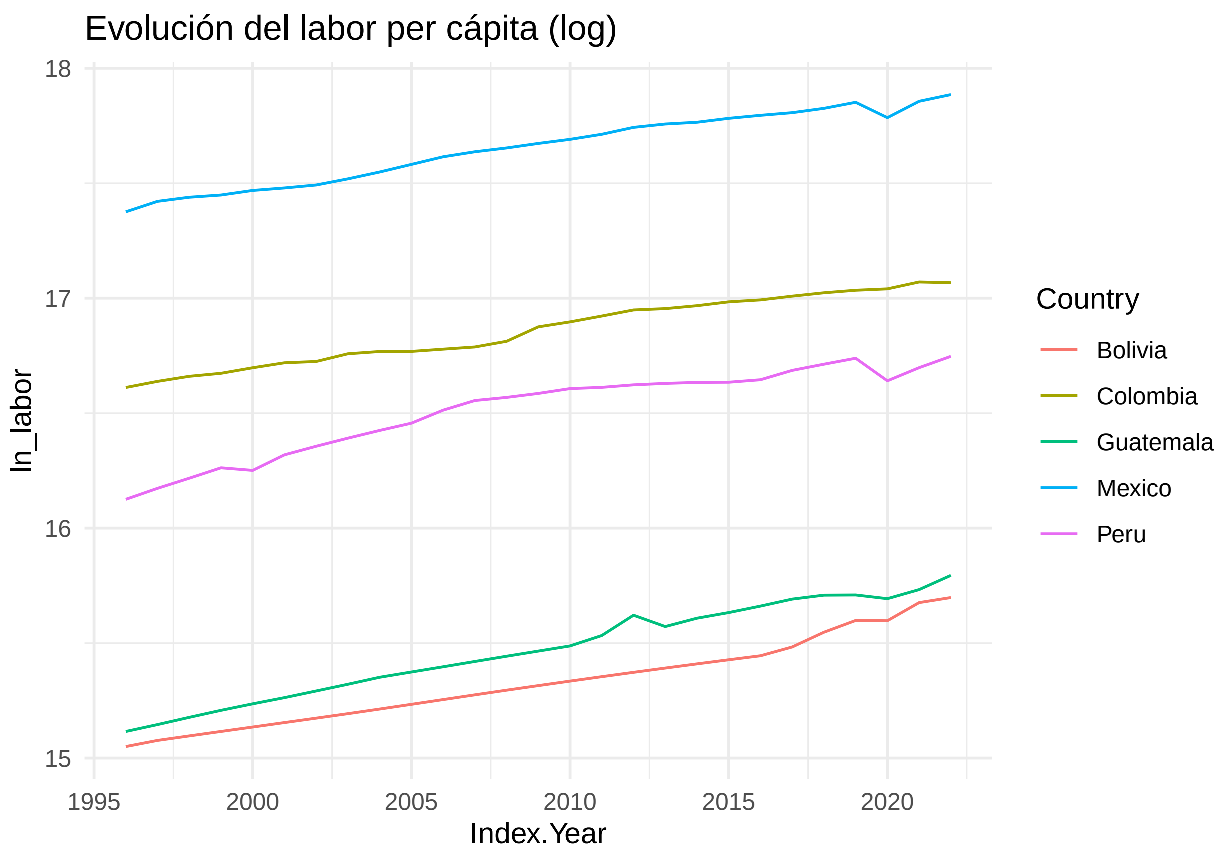
```
ggplot(df, aes(x = Index.Year, y = ln_gdp, color = Country)) +
  geom_line() +
  labs(title = "Evolución del PIB per cápita (log)") +
  theme_minimal()
```



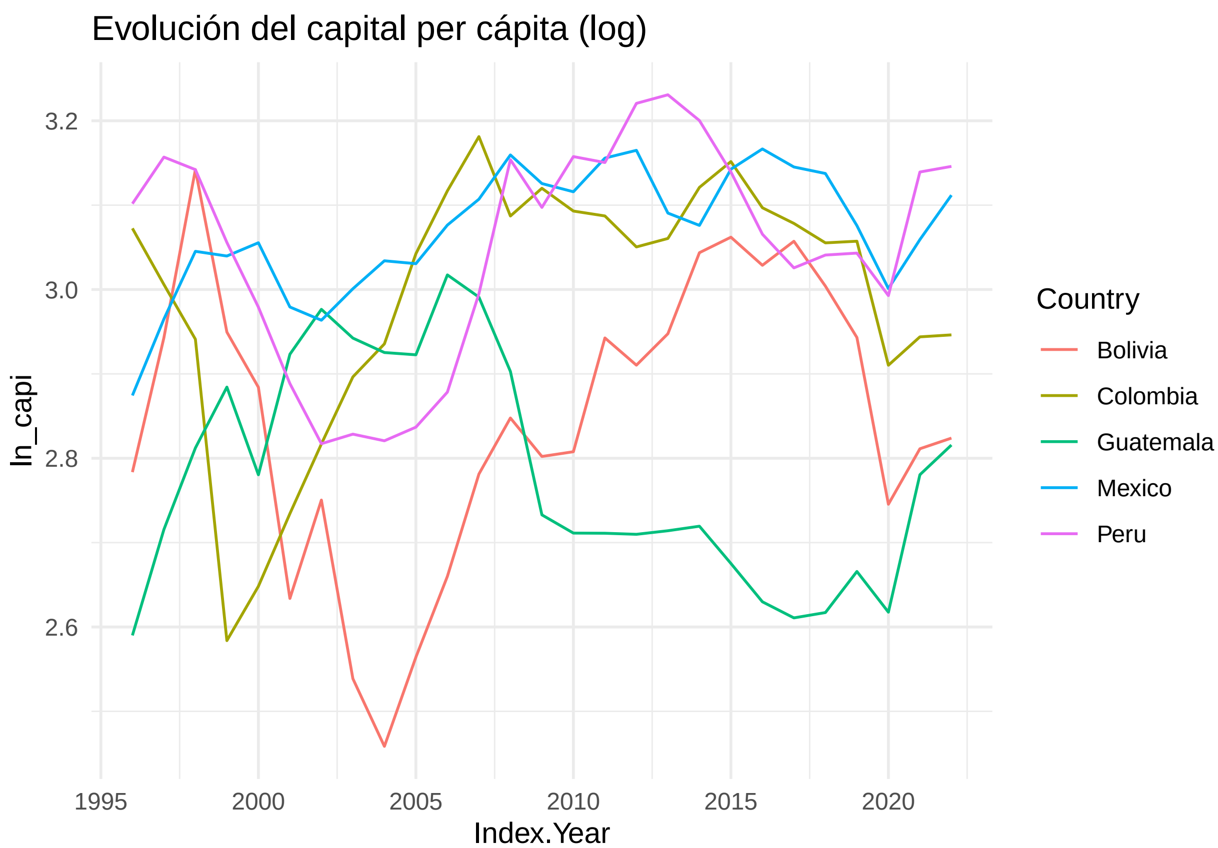

```r
ggplot(df, aes(x = Index.Year, y = ln_art, color = Country)) +
  geom_line() +
  labs(title = "Evolución de los artículos per cápita (log)") +
  theme_minimal()
```



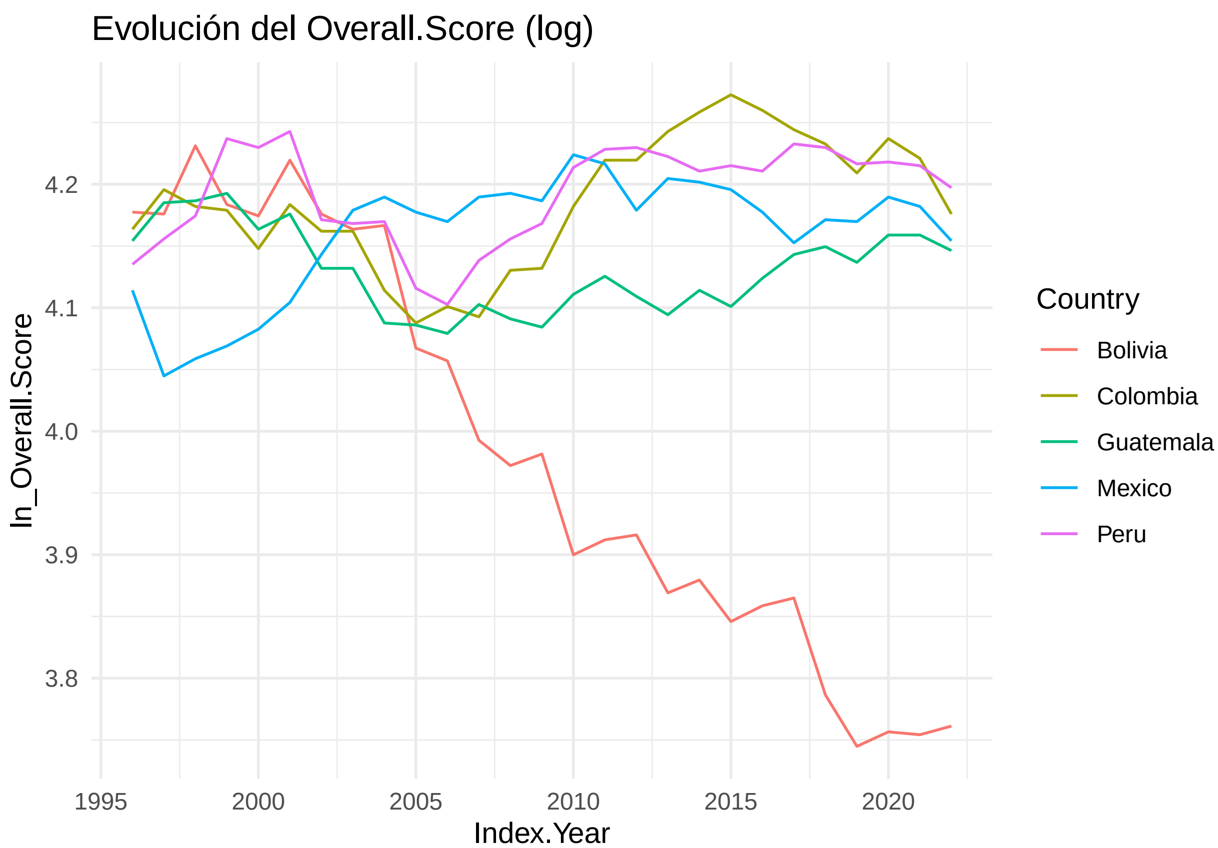

```r
ggplot(df, aes(x = Index.Year, y = ln_labor, color = Country)) +
  geom_line() +
  labs(title = "Evolución del labor per cápita (log)") +
  theme_minimal()
```




```r
ggplot(df, aes(x = Index.Year, y = ln_capi, color = Country)) +
  geom_line() +
  labs(title = "Evolución del capital per cápita (log)") +
  theme_minimal()
```




```r
ggplot(df, aes(x = Index.Year, y = ln_Overall.Score, color = Country)) +
  geom_line() +
  labs(title = "Evolución del Overall.Score (log)") +
  theme_minimal()
```



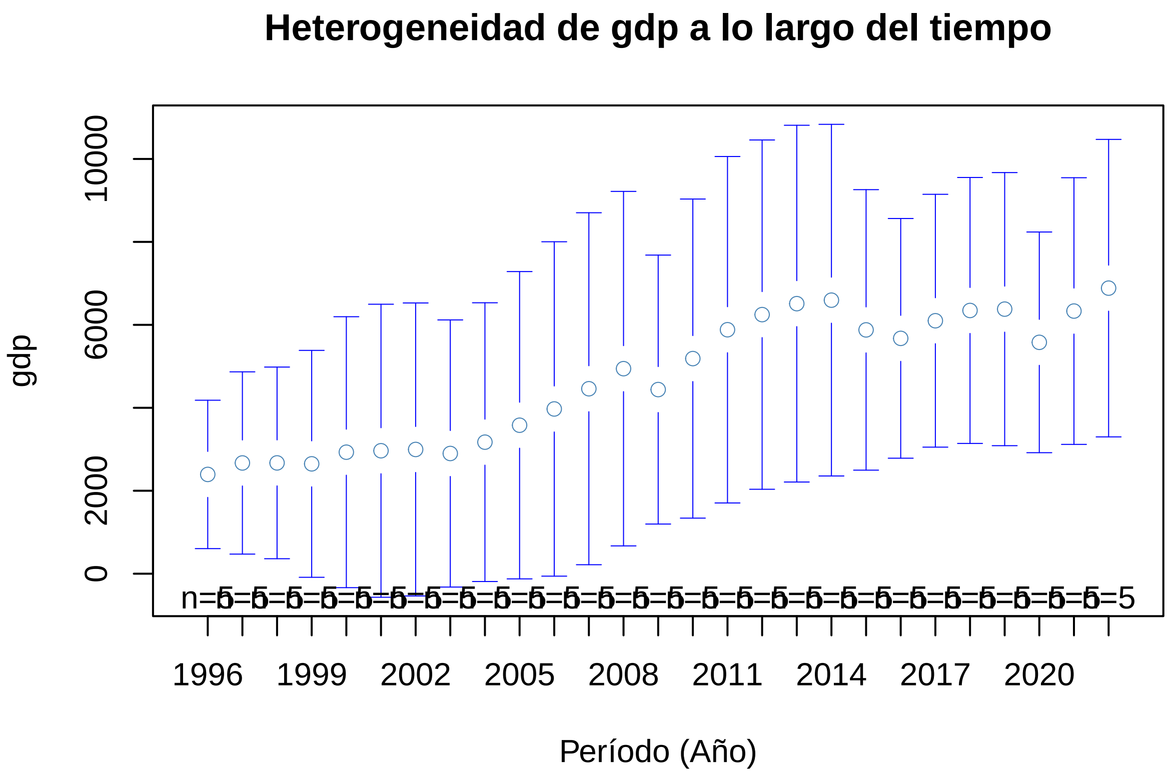
7. Visualización para heterogeneidad

```r
# install.packages("gplots") # Descomenta y ejecuta si no tienes gplots instalado
library(gplots)
```

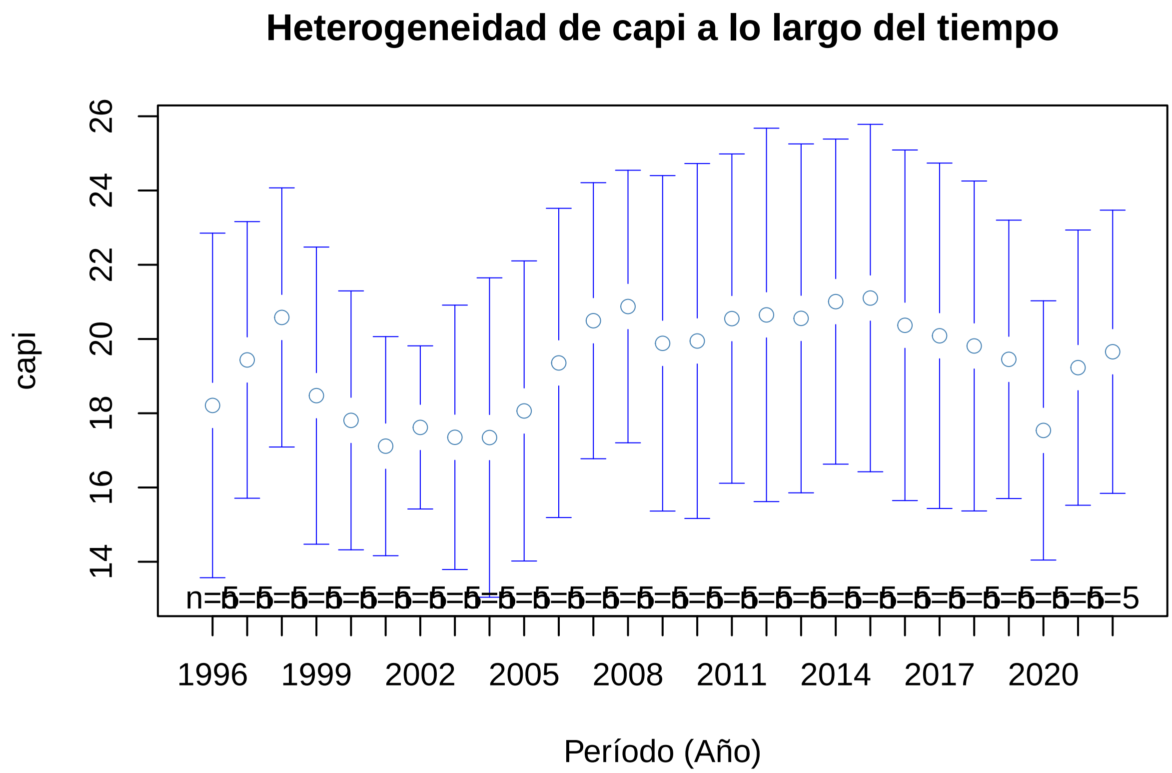
```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```r
library(dplyr) # Para usar el operador %>% si lo necesitas para otras operaciones
# Lista de las variables de tu modelo que quieres visualizar


variables_modelo <- c("gdp", "capi", "art", "labor", "Overall.Score")

# Asumiendo que tu variable de tiempo en 'df' se llama 'year'.
# ¡¡¡IMPORTANTE!!! Reemplaza 'df$year' si tu variable de tiempo tiene otro nombre.
```

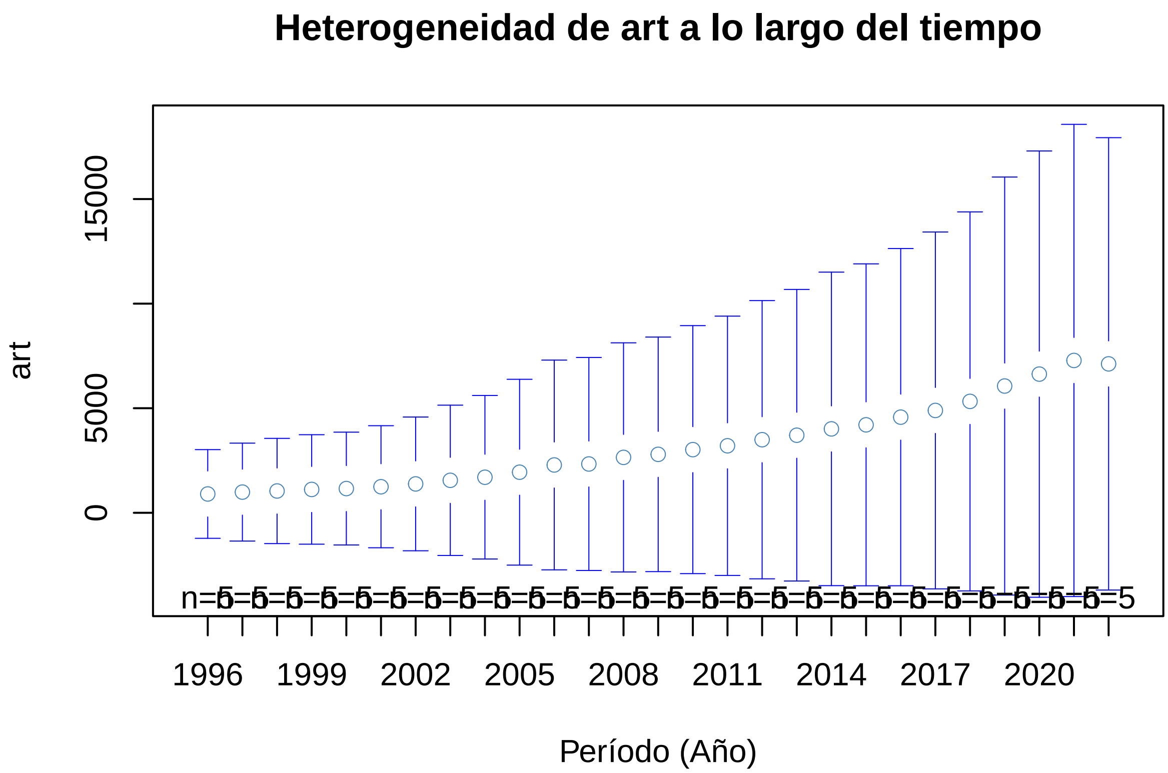
```r
time_variable <- df$Index.Year # O df$tu_nombre_de_variable_de_tiempo

# Bucle para generar un gráfico para cada variable
for (var_name in variables_modelo) {
  # Construye la fórmula: variable_actual ~ variable_de_tiempo
  formula_str <- paste(var_name, "~ time_variable")

  # Define el título del gráfico
  plot_title <- paste("Heterogeneidad de", var_name, "a lo largo del tiempo")

  # Define la etiqueta del eje Y (puedes personalizarla más si lo deseas)
  ylab_text <- paste(var_name)

  # Genera el gráfico
  plotmeans(
    as.formula(formula_str), # Convierte la cadena de texto a una fórmula
    data = df,               # Usa tu dataframe df
    main = plot_title,
    xlab = "Período (Año)", # Puedes cambiar "Año" por "Período" si tu variable no son años
    ylab = ylab_text,
    connect = FALSE,         # Para no conectar los puntos con líneas (útil si los años no son consecut
    barwidth = 0.5,          # Ancho de las barras de los intervalos de confianza
    col = "steelblue",       # Color de los puntos y barras
    ccol = "darkblue"        # Color de las líneas que conectan la media (si connect=TRUE)
  )
}
```

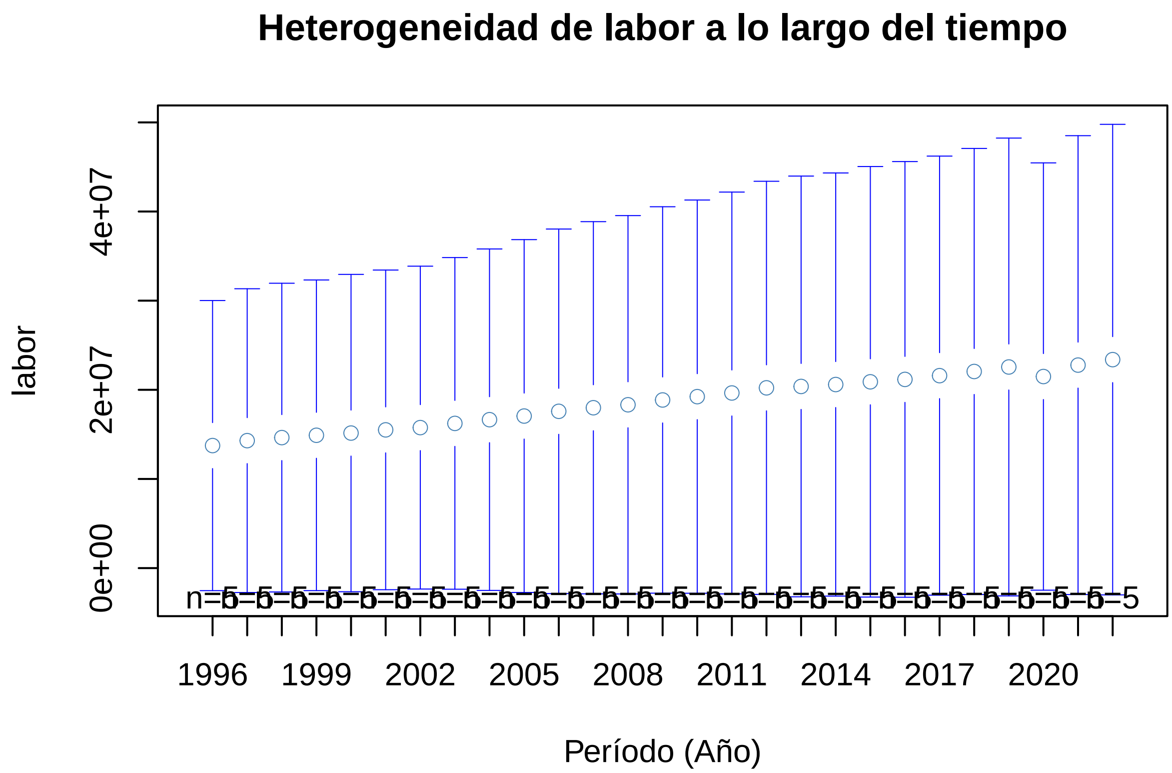


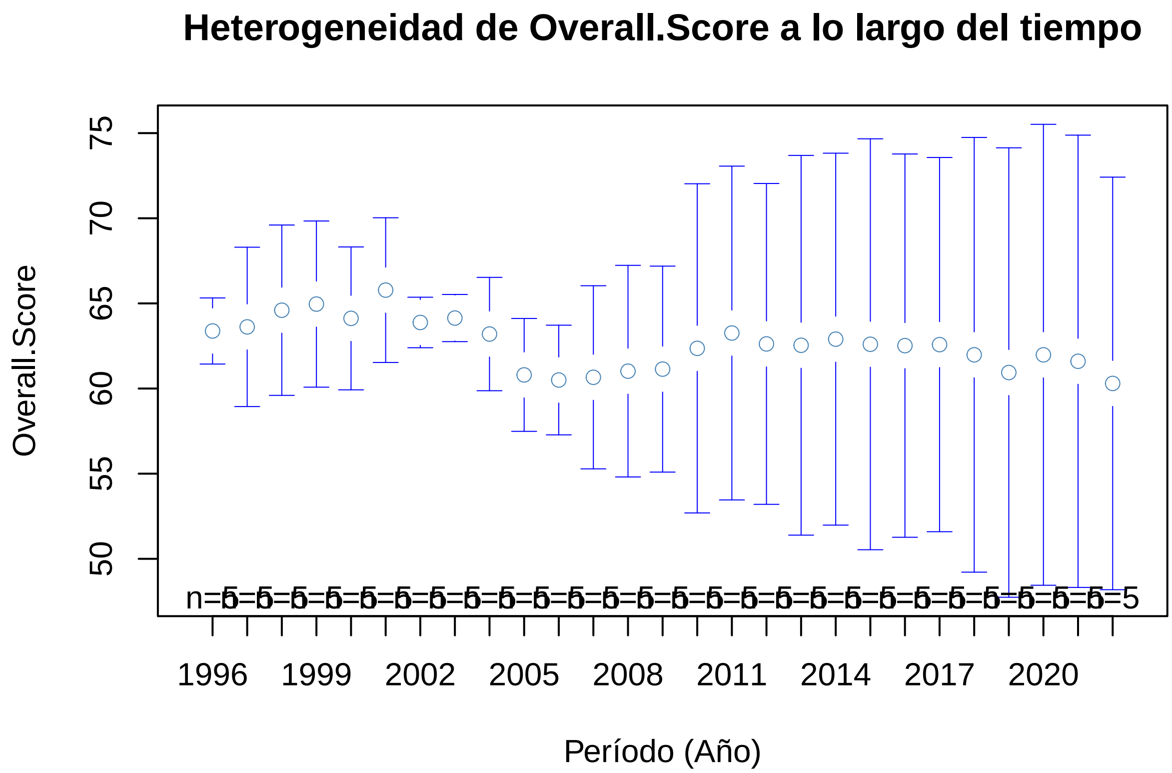
Heterogeneidad de gdp a lo largo del tiempo
gdp
0
2000
6000
10000
1996
1999
2002
2005
2008
2011
2014
2017
2020
Período (Año)



Heterogeneidad de capi a lo largo del tiempo
capi
14
16
18
20
22
24
26
n=5=5=5=5=5=5=5=5=5=5=5=5=5=5=5=5=5=5=5=5=5=5=5=5=5=5=5
1996
1999
2002
2005
2008
2011
2014
2017
2020
Período (Año)

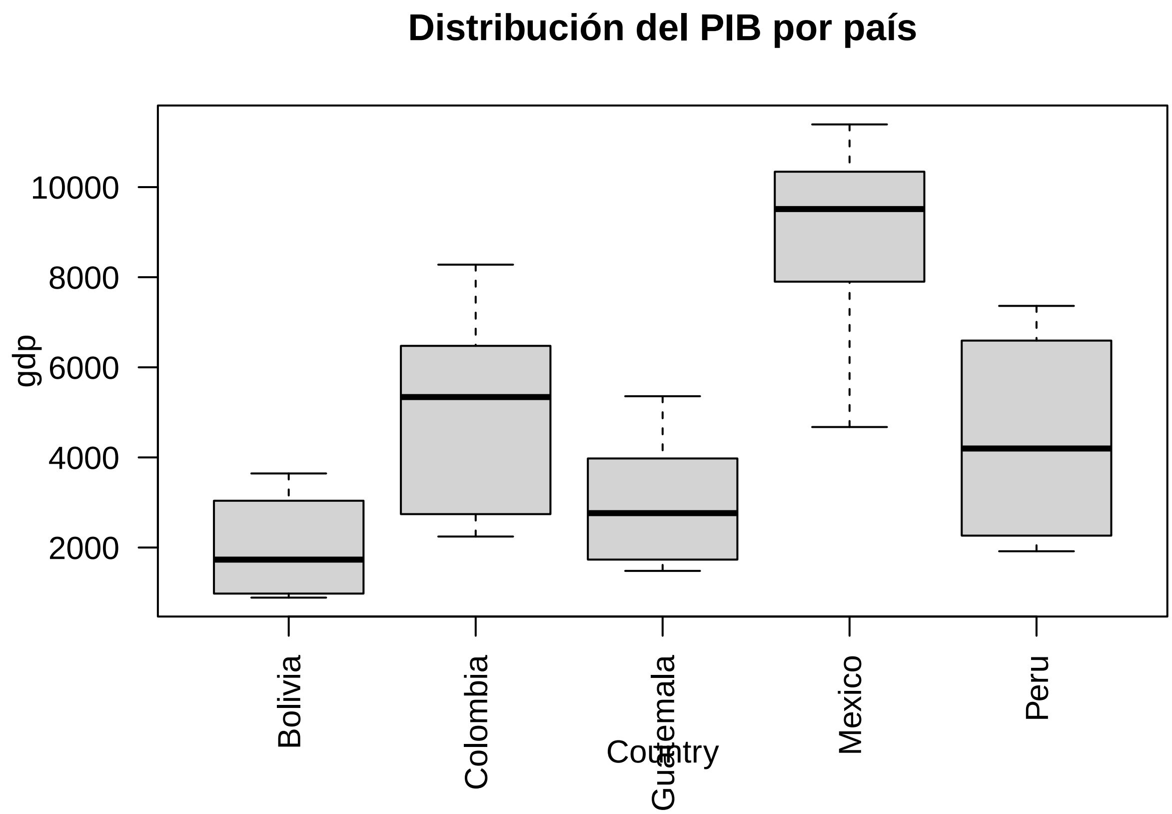

Heterogeneidad de art a lo largo del tiempo
art
0
5000
15000
1996
1999
2002
2005
2008
2011
2014
2017
2020
Período (Año)


Heterogeneidad de labor a lo largo del tiempo
labor
0e+00
2e+07
4e+07
1996
1999
2002
2005
2008
2011
2014
2017
2020
Período (Año)

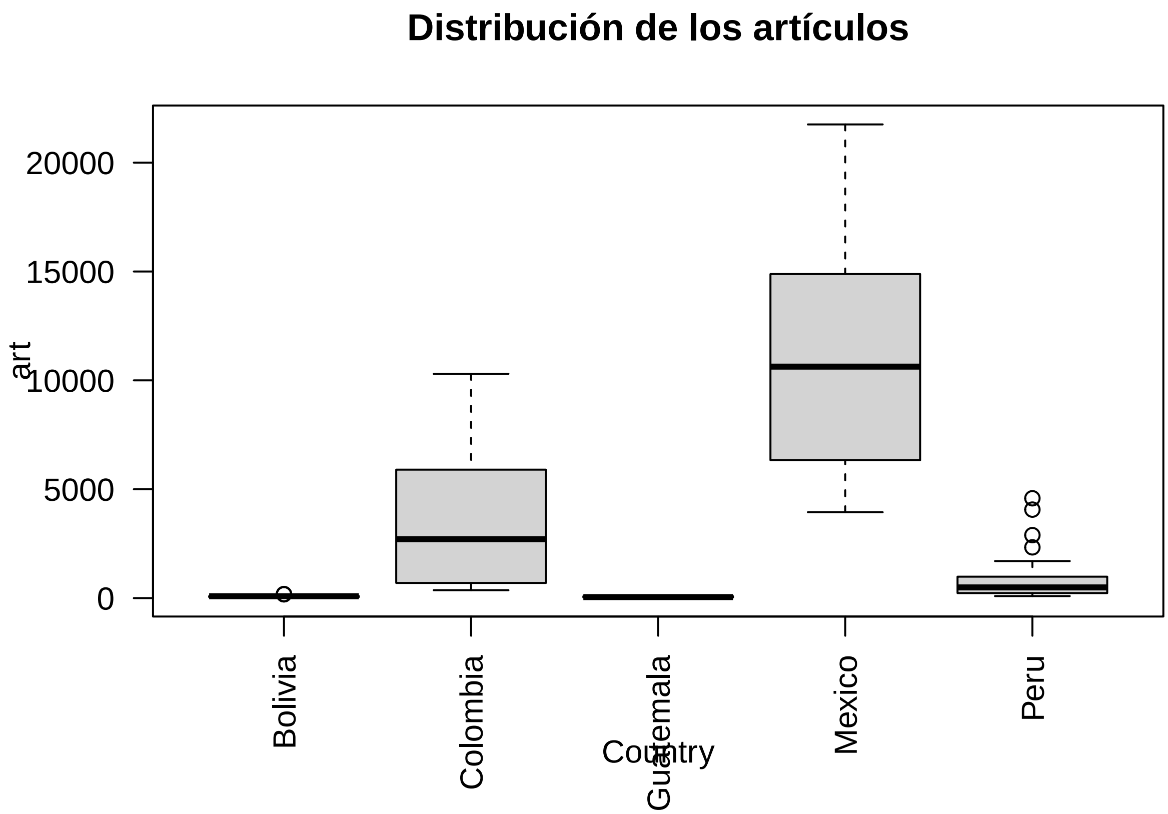


7.1. Antes de la transformación

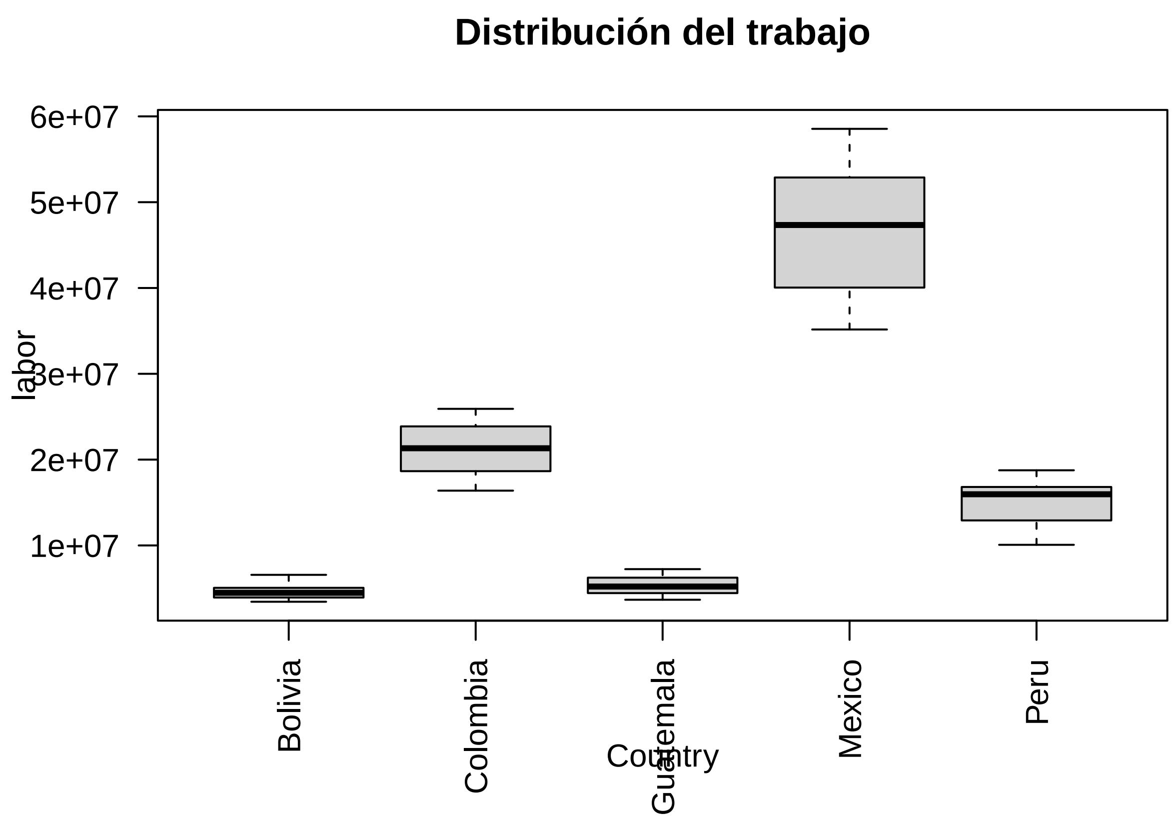
```
boxplot(gdp ~ Country, data = df, main = "Distribución del PIB por país", las = 2)
```



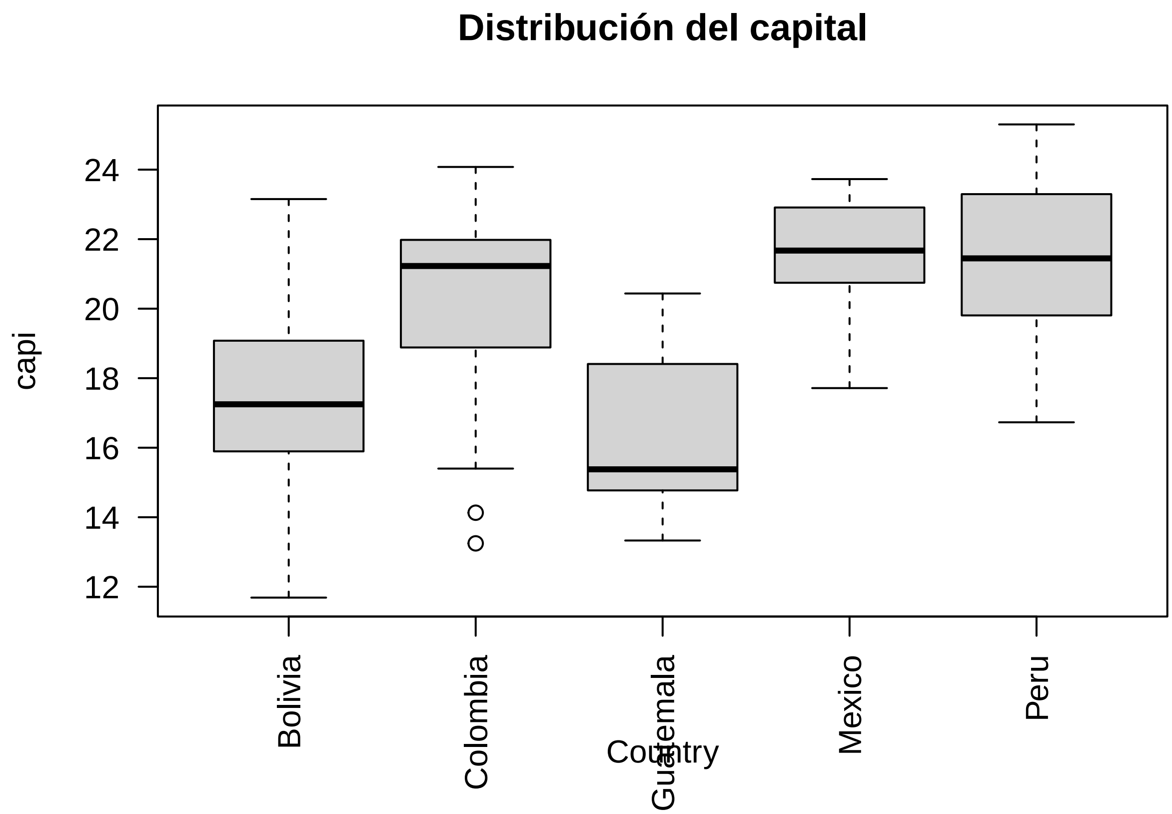

```r
boxplot(art ~ Country, data = df, main = "Distribución de los artículos", las = 2)
```



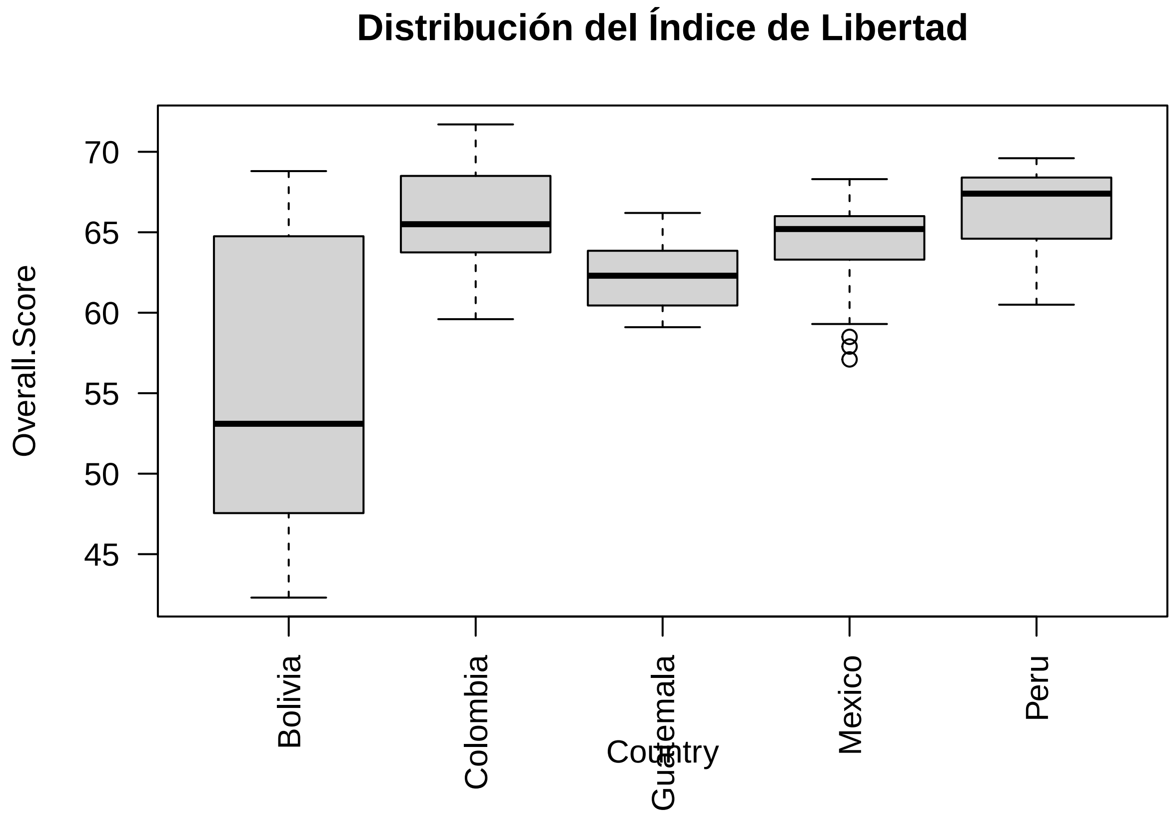

```
boxplot(labor ~ Country, data = df, main = "Distribución del trabajo", las = 2)
```




```r
boxplot(capi ~ Country, data = df, main = "Distribución del capital", las = 2)
```

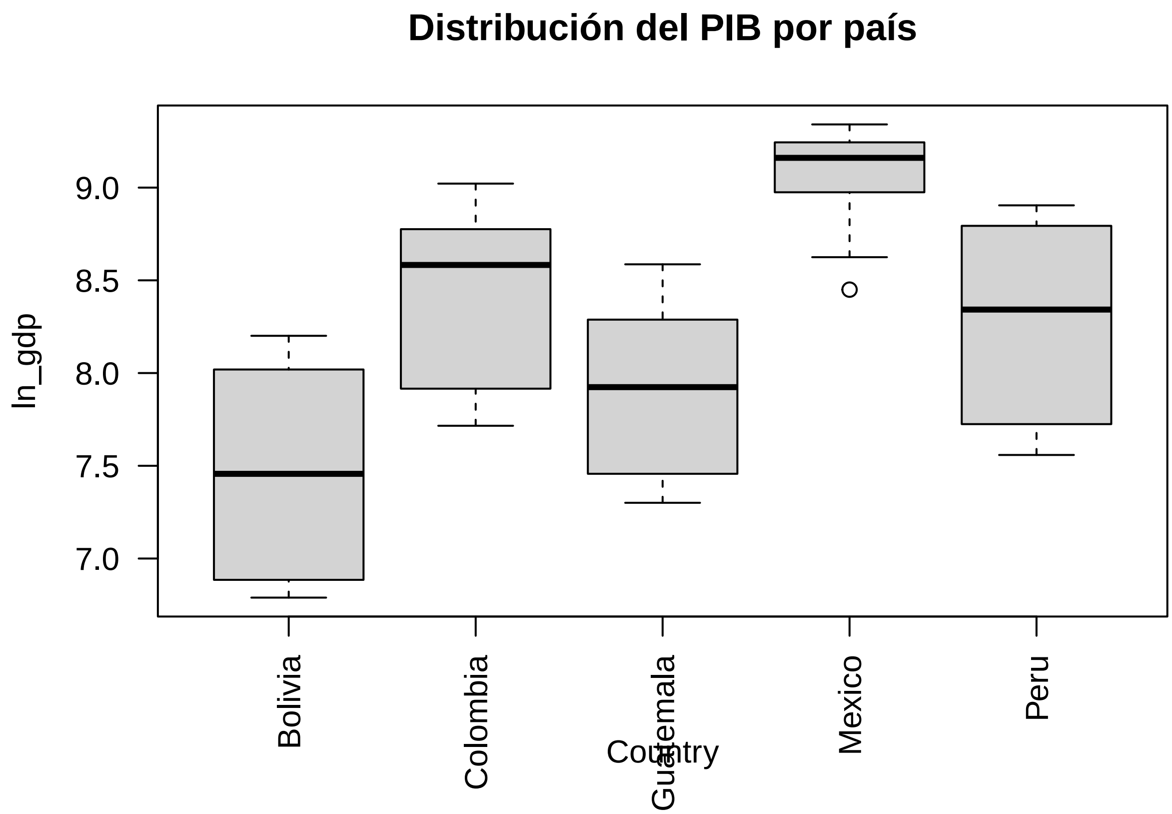



```r
boxplot(Overall.Score ~ Country, data = df, main = "Distribución del Índice de Libertad", las = 2)
```

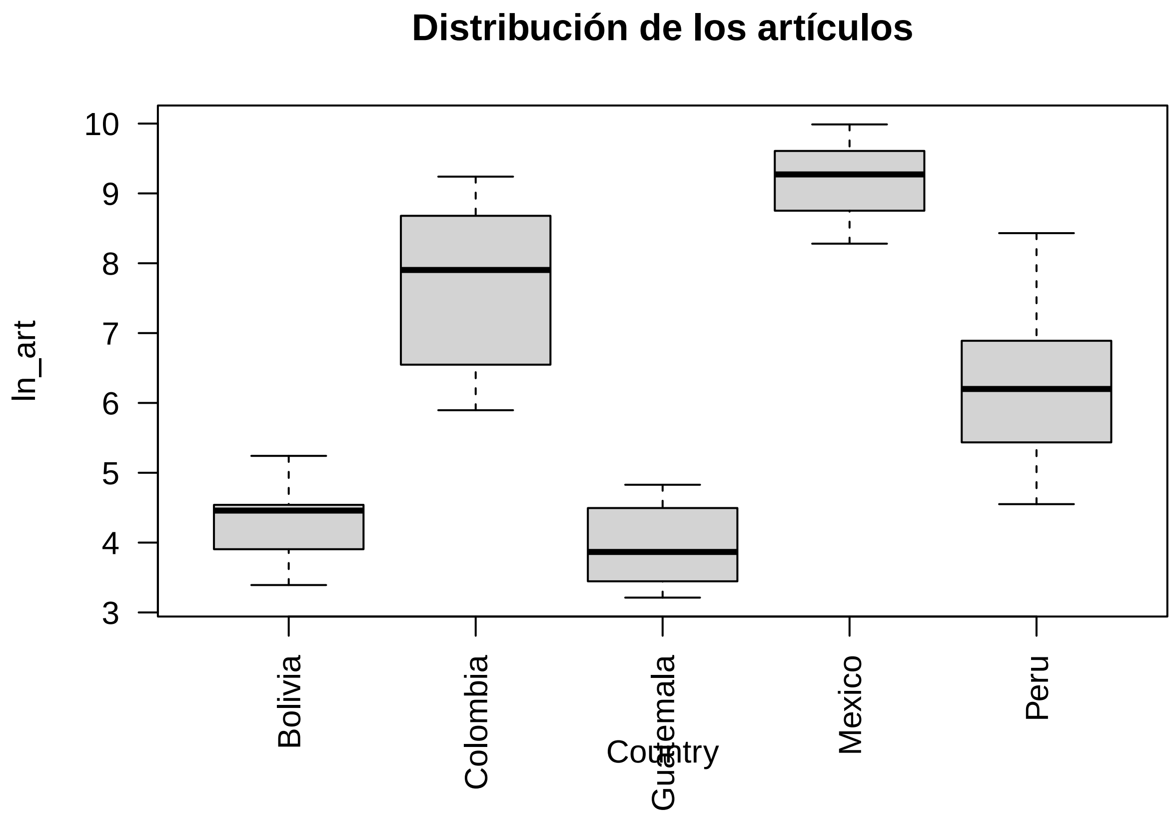


7.2. Después de la transformación (es decir en logaritmos)

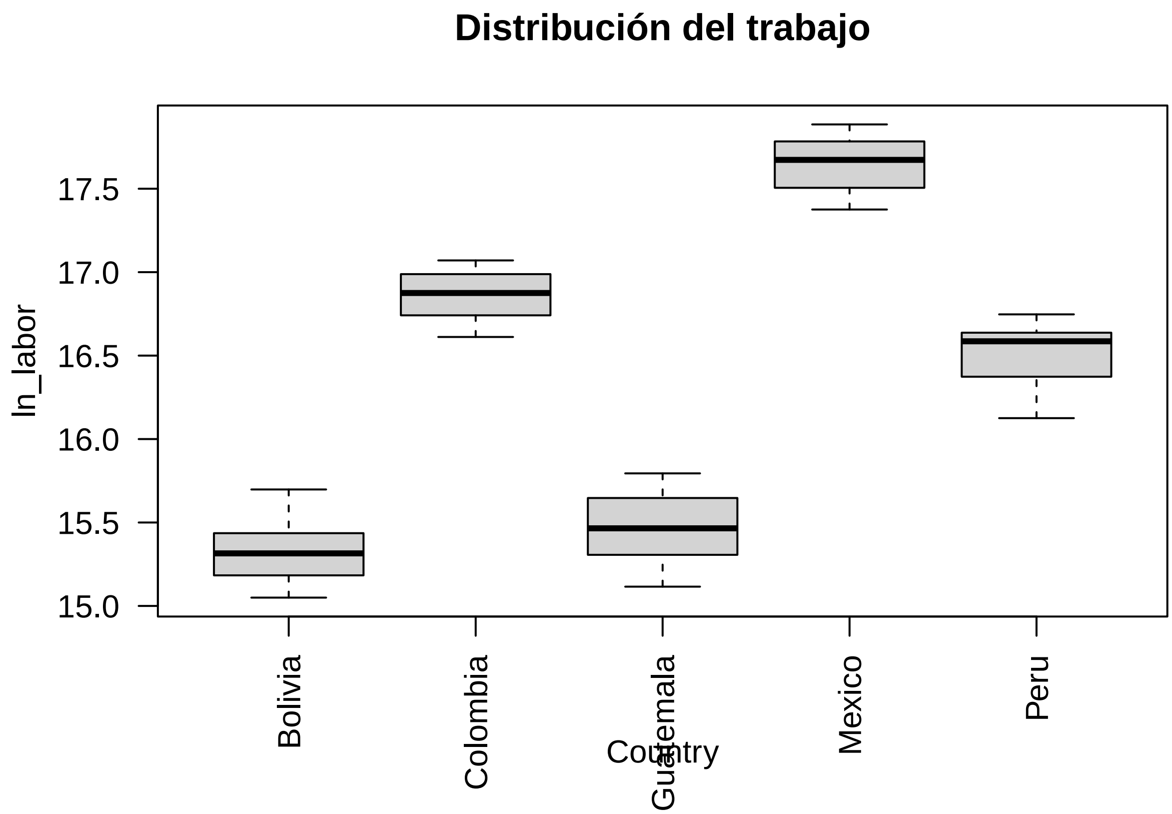
```
boxplot(ln_gdp ~ Country, data = df, main = "Distribución del PIB por país", las = 2)
```



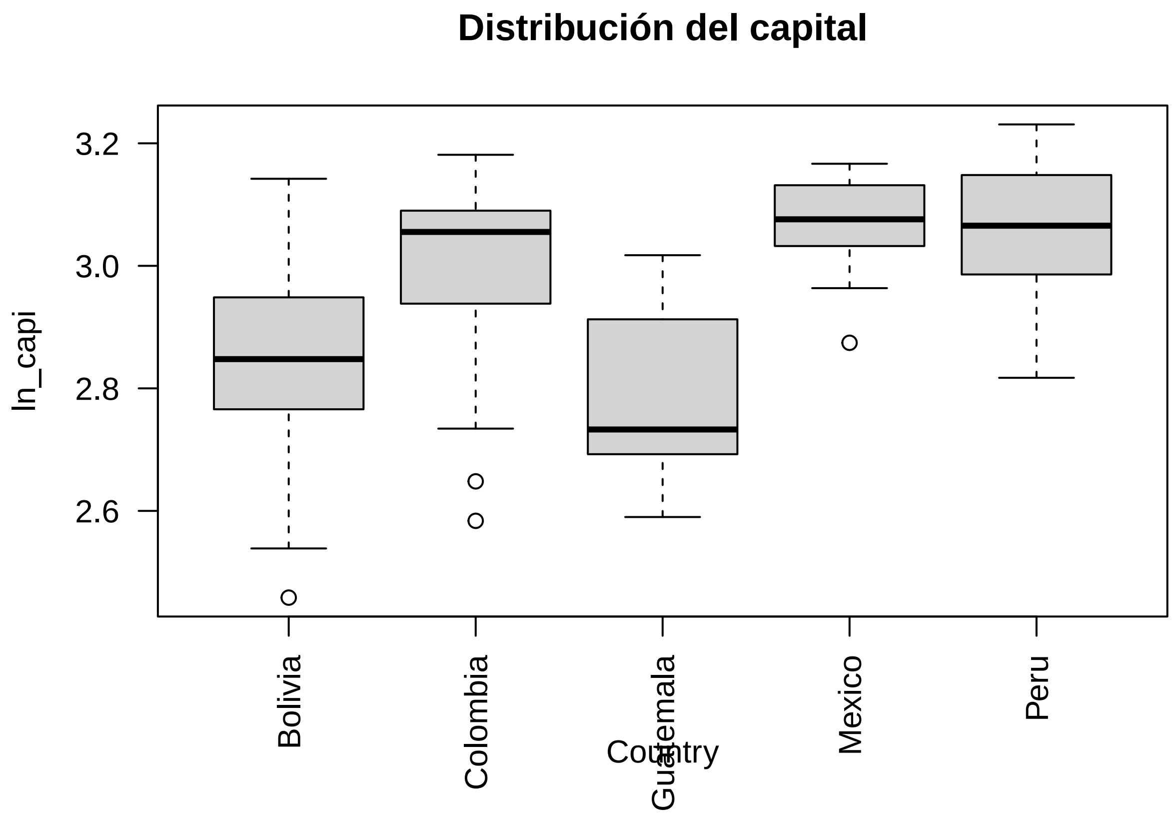

```
boxplot(ln_art ~ Country, data = df, main = "Distribución de los artículos", las = 2)
```



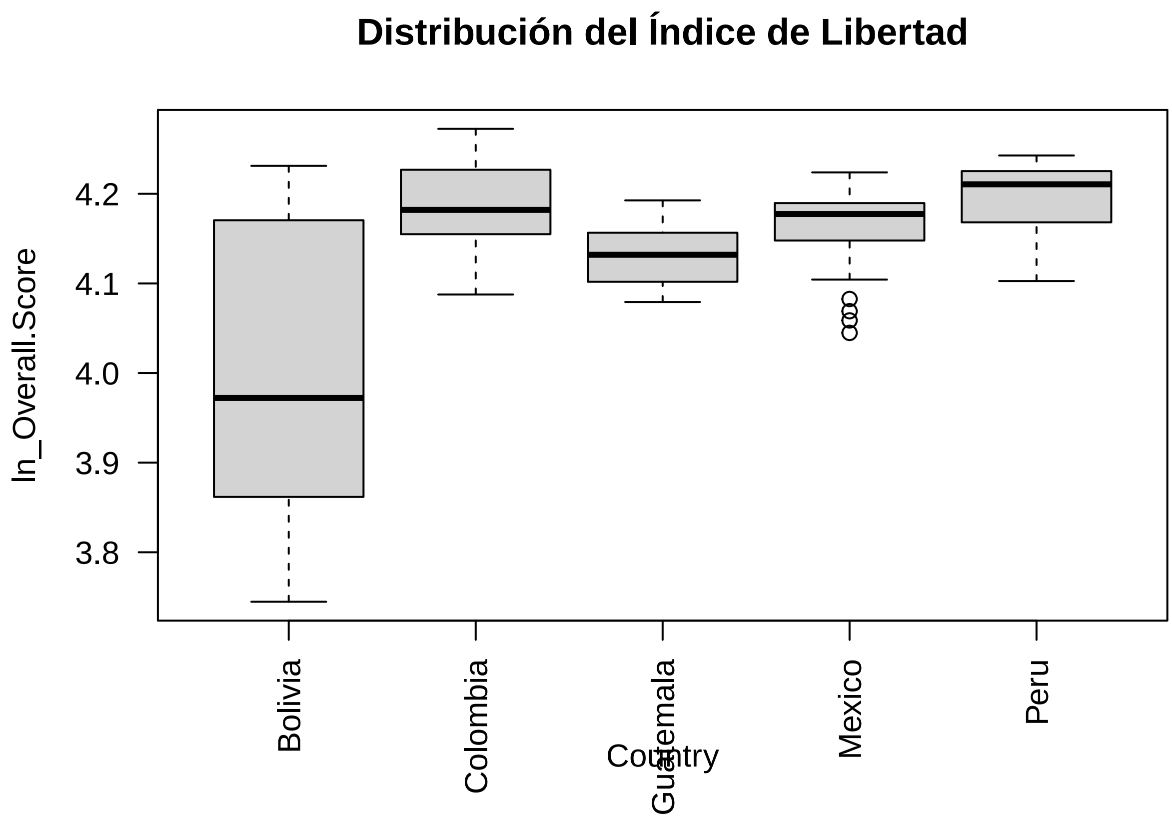

```r
boxplot(ln_labor ~ Country, data = df, main = "Distribución del trabajo", las = 2)
```




```
boxplot(ln_capi ~ Country, data = df, main = "Distribución del capital", las = 2)
```



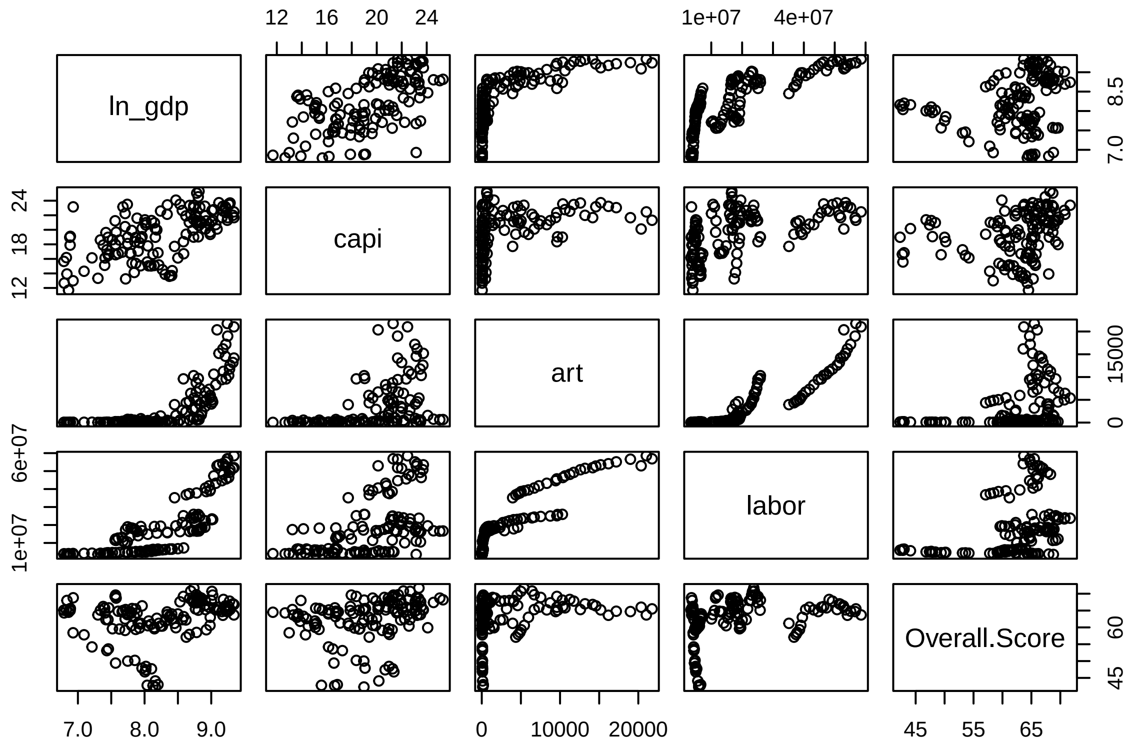

```
boxplot(ln_Overall.Score ~ Country, data = df, main = "Distribución del Índice de Libertad", las = 2)
```



Con estos gráficos podemos intuir que vamos a tener que usar el modelo de efectos fijos.

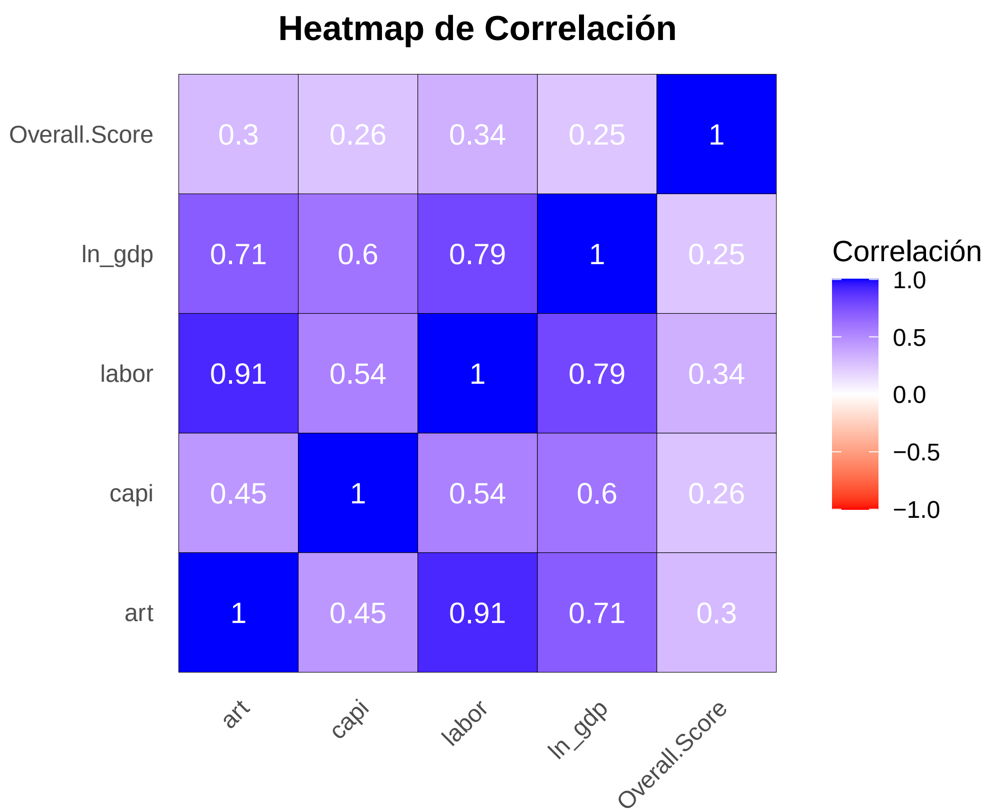
8. Correlación y colinealidad

```r
# Seleccionar las variables de interés
data_for_pairs <- df %>%
  select(ln_gdp, capi, art, labor, Overall.Score)

# Crear el gráfico pairs básico
pairs(data_for_pairs)
```

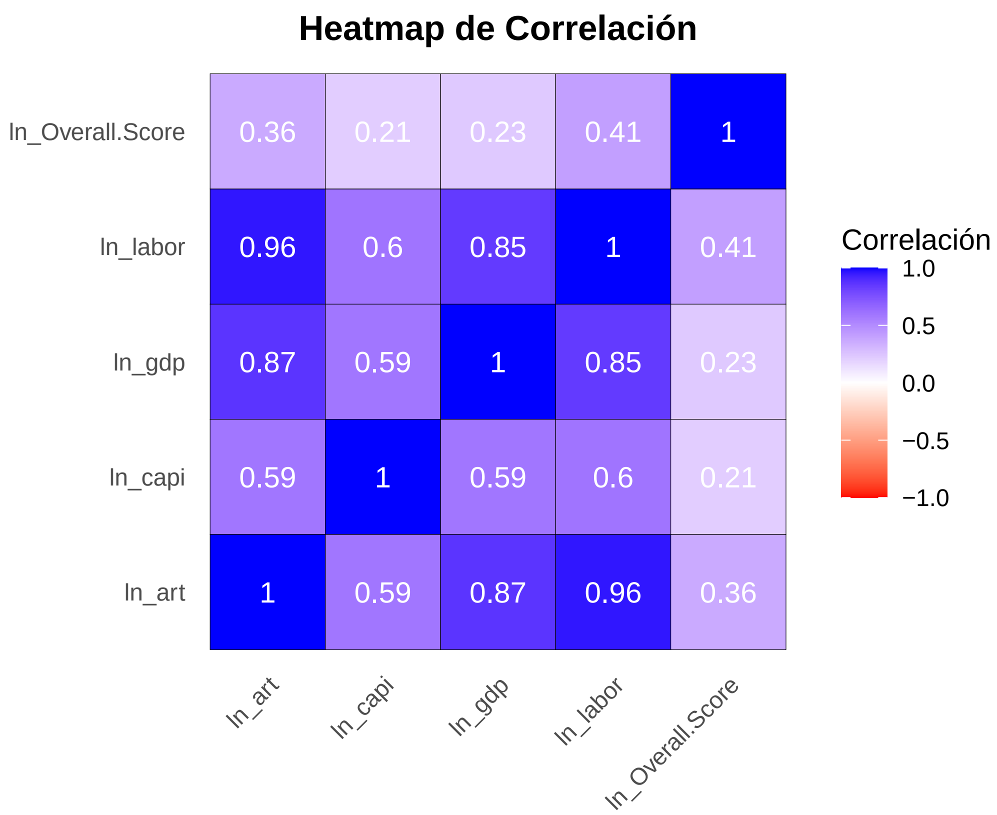


Vamos a ver la matriz de correlación, entre

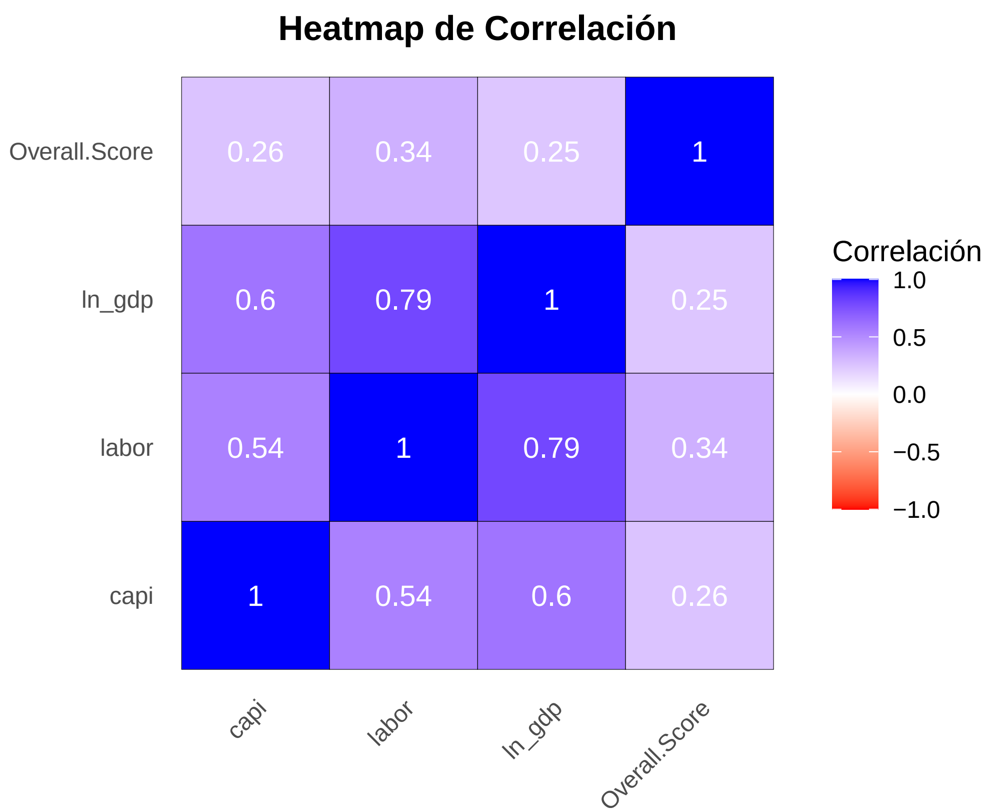
```
library(dplyr); library(tidyr); library(ggplot2)
# Matriz de correlación


df %>%
  select(ln_gdp, capi, art, labor, Overall.Score) %>%
  cor() %>%
  as.data.frame() %>%
  tibble::rownames_to_column("Var1") %>%
  pivot_longer(cols = -Var1, names_to = "Var2", values_to = "value") %>%
  ggplot(aes(Var1, Var2, fill = value)) +
  geom_tile(color = "black") +
  geom_text(aes(label = round(value, 2)), color = "white", size = 4) +
  scale_fill_gradient2(low = "red", mid = "white", high = "blue", midpoint = 0, limit = c(-1, 1), name =
  coord_fixed() +
  labs(title = "Heatmap de Correlación", x = "", y = "") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), panel.grid.major = element_blank(), panel.gr
```



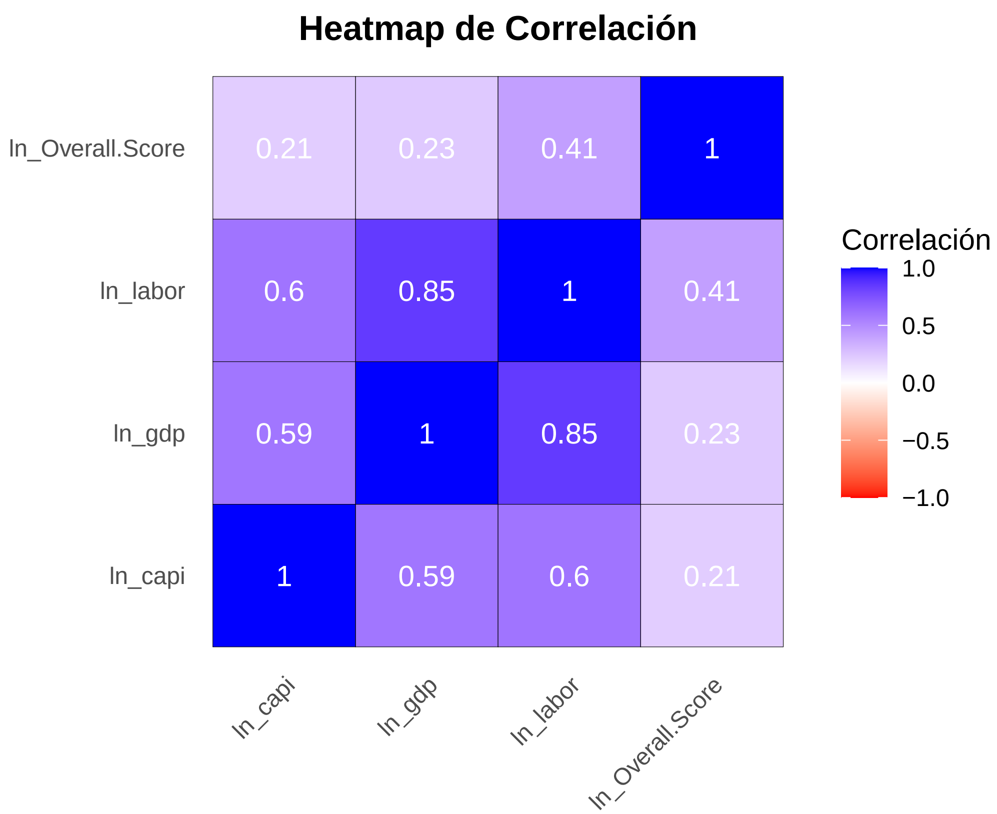

```
# Se puede observar que hay una gran correlación entre ln_art y ln_labor


# Matriz de correlación
df %>%
  select(ln_gdp, ln_capi, ln_art, ln_labor, ln_Overall.Score) %>%
  cor() %>%
  as.data.frame() %>%
  tibble::rownames_to_column("Var1") %>%
  pivot_longer(cols = -Var1, names_to = "Var2", values_to = "value") %>%
  ggplot(aes(Var1, Var2, fill = value)) +
  geom_tile(color = "black") +
  geom_text(aes(label = round(value, 2)), color = "white", size = 4) +
  scale_fill_gradient2(low = "red", mid = "white", high = "blue", midpoint = 0, limit = c(-1, 1), name =
  coord_fixed() +
  labs(title = "Heatmap de Correlación", x = "", y = "") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), panel.grid.major = element_blank(), panel.gri
```



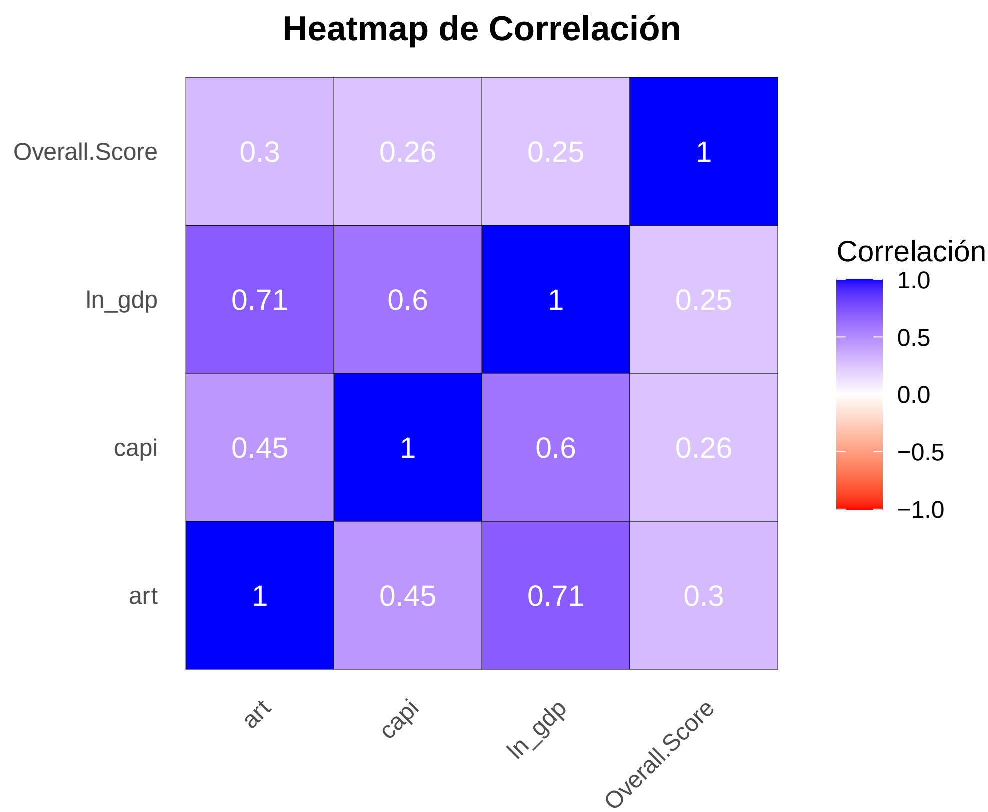

```r
#Se mantiene la correlación con los logaritmos


# Por lo que borramos la variable art, y vemos que desaparece este problema de correlación
# entre variables, luego haremos lo mismo con las pruebas VIF, y con la eliminación
# de una variable solucionaremos el problema

# Matriz de correlación, sin art
df %>%
  select(ln_gdp, capi, labor, Overall.Score) %>%
  cor() %>%
  as.data.frame() %>%
  tibble::rownames_to_column("Var1") %>%
  pivot_longer(cols = -Var1, names_to = "Var2", values_to = "value") %>%
  ggplot(aes(Var1, Var2, fill = value)) +
  geom_tile(color = "black") +
  geom_text(aes(label = round(value, 2)), color = "white", size = 4) +
  scale_fill_gradient2(low = "red", mid = "white", high = "blue", midpoint = 0, limit = c(-1, 1), name =
  coord_fixed() +
  labs(title = "Heatmap de Correlación", x = "", y = "") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), panel.grid.major = element_blank(), panel.gr
```



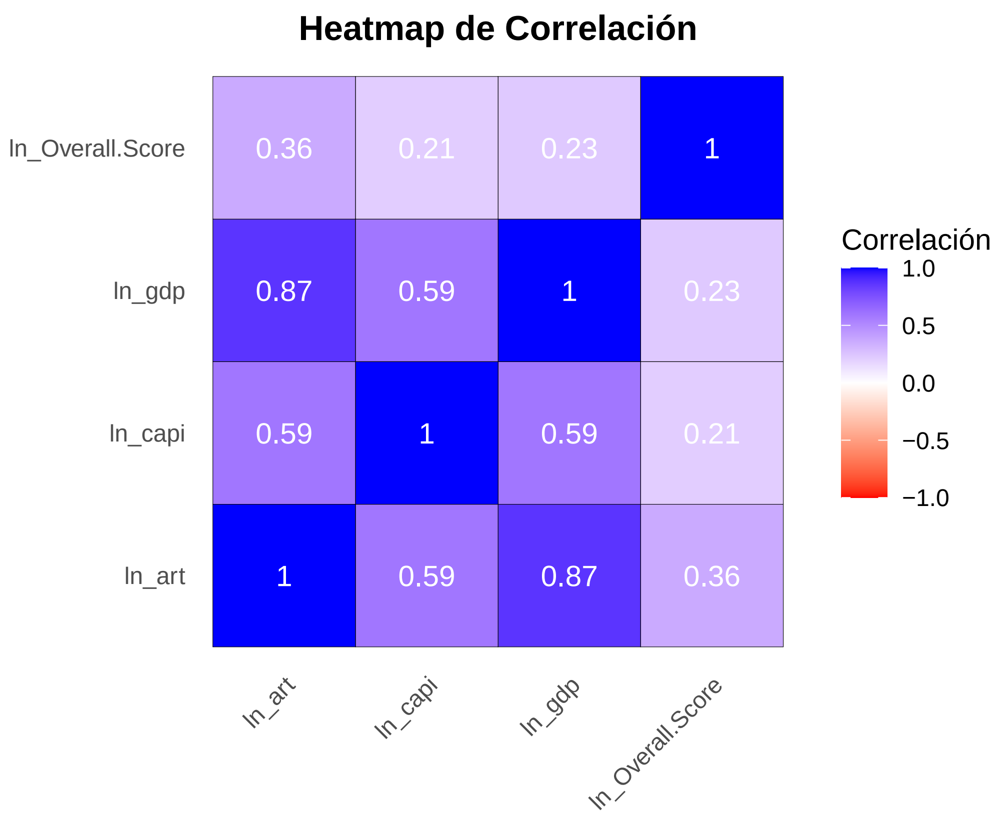

```r
# Matriz de correlación, sin art (con log)
df %>%
  select(ln_gdp, ln_capi, ln_labor, ln_Overall.Score) %>%
  cor() %>%
  as.data.frame() %>%
  tibble::rownames_to_column("Var1") %>%
  pivot_longer(cols = -Var1, names_to = "Var2", values_to = "value") %>%
  ggplot(aes(Var1, Var2, fill = value)) +
  geom_tile(color = "black") +
  geom_text(aes(label = round(value, 2)), color = "white", size = 4) +
  scale_fill_gradient2(low = "red", mid = "white", high = "blue", midpoint = 0, limit = c(-1, 1), name =
  coord_fixed() +
  labs(title = "Heatmap de Correlación", x = "", y = "") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), panel.grid.major = element_blank(), panel.gri
```




```r
# Matriz de correlación sin labor

df %>%
  select(ln_gdp, capi,art, Overall.Score) %>%
  cor() %>%
  as.data.frame() %>%
  tibble::rownames_to_column("Var1") %>%
  pivot_longer(cols = -Var1, names_to = "Var2", values_to = "value") %>%
  ggplot(aes(Var1, Var2, fill = value)) +
  geom_tile(color = "black") +
  geom_text(aes(label = round(value, 2)), color = "white", size = 4) +
  scale_fill_gradient2(low = "red", mid = "white", high = "blue", midpoint = 0, limit = c(-1, 1), name =
  coord_fixed() +
  labs(title = "Heatmap de Correlación", x = "", y = "") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), panel.grid.major = element_blank(), panel.gr
```




```r
#Con logaritmos
df %>%
  select(ln_gdp, ln_capi,ln_art, ln_Overall.Score) %>%
  cor() %>%
  as.data.frame() %>%
  tibble::rownames_to_column("Var1") %>%
  pivot_longer(cols = -Var1, names_to = "Var2", values_to = "value") %>%
  ggplot(aes(Var1, Var2, fill = value)) +
  geom_tile(color = "black") +
  geom_text(aes(label = round(value, 2)), color = "white", size = 4) +
  scale_fill_gradient2(low = "red", mid = "white", high = "blue", midpoint = 0, limit = c(-1, 1), name =
  coord_fixed() +
  labs(title = "Heatmap de Correlación", x = "", y = "") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1), panel.grid.major = element_blank(), panel.gr
```



Hacemos la prubea de Varianza Inflada (VIF)

```r
# Primero mantenemos todas las variabkes

# VIF (con todos sin la transforamción del logaritmo)
modelo_vif <- lm(gdp ~ capi + art + labor + Overall.Score, data = df)
vif(modelo_vif)
```

```
##          capi           art         labor Overall.Score
##      1.456625      6.158046      7.099257      1.140191
```

```r
# VIF (con la condición del logaritmo)
modelo_vif <- lm(ln_gdp ~ capi + art + labor + Overall.Score, data = df)
vif(modelo_vif)
```

```
##          capi           art         labor Overall.Score
##      1.456625      6.158046      7.099257      1.140191
```

```r
# Observamos que están demasiado infladas, labor y art (son mayores a 5). Por lo que
# Con esto veremos que es necesario eliminar una de las variables, para que la regresión
# no pierda significancia
```

```r
# VIF (quitamos a labor, ya que está muy correlacionada con art)
modelo_vif <- lm(ln_gdp ~ capi + art  + Overall.Score, data = df)
vif(modelo_vif)
```

```
##          capi           art Overall.Score
##      1.278141      1.308834      1.119778
```

```r
 # Con logaritmos
modelo_vif <- lm(ln_gdp ~ ln_capi + ln_art  + ln_Overall.Score, data = df)
vif(modelo_vif)
```

```
##          ln_capi           ln_art ln_Overall.Score
##         1.545163         1.699527         1.152147
```

```r
# VIF (quitamos a art, ya que está muy correlacionada con labor)
modelo_vif <- lm(ln_gdp ~ capi + labor + Overall.Score, data = df)
vif(modelo_vif)
```

```
##          capi         labor Overall.Score
##      1.429459      1.508879      1.139594
```

```r
 # Con logaritmos
modelo_vif <- lm(ln_gdp ~ ln_capi + ln_labor  + ln_Overall.Score, data = df)
vif(modelo_vif)
```

```
##          ln_capi         ln_labor ln_Overall.Score
##         1.569569         1.806432         1.208404
```

```r
# Con esto vemos que se soluciona el problema de correlación entre las variables exógenas
```

Hacemos las regresiones

Con labor

Pooled

```r
knitr::opts_chunk$set(error = TRUE)
reg_pool = plm(ln_gdp ~ capi+ labor+ Overall.Score, data = df, model = "pooling")
summary(reg_pool)
```

```
## Pooling Model
##
## Call:
## plm(formula = ln_gdp ~ capi + labor + Overall.Score, data = df,
##     model = "pooling")
##
## Balanced Panel: n = 5, T = 27, N = 135
```

```
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -1.026705 -0.256018 -0.013113  0.305596  0.843490
##
## Coefficients:
##                  Estimate  Std. Error t-value  Pr(>|t|)
## (Intercept)    6.9663e+00  4.0193e-01 17.3323 < 2.2e-16 ***
## capi           5.3420e-02  1.2982e-02  4.1149 6.805e-05 ***
## labor          2.9672e-08  2.6963e-09 11.0047 < 2.2e-16 ***
## Overall.Score -5.1392e-03  5.8307e-03 -0.8814    0.3797
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    65.451
## Residual Sum of Squares: 21.436
## R-Squared:      0.6725
## Adj. R-Squared: 0.665
## F-statistic: 89.665 on 3 and 131 DF, p-value: < 2.22e-16
```

Regresión de efectos fijos

```r
#ESTIMACIÓN DE EFECTOS FIJOS
reg_fe = plm(ln_gdp ~ capi+ labor+ Overall.Score, data = df, model = "within")
summary(reg_fe)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = ln_gdp ~ capi + labor + Overall.Score, data = df,
##     model = "within")
##
## Balanced Panel: n = 5, T = 27, N = 135
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -0.596207 -0.267807  0.023937  0.235258  0.586621
##
## Coefficients:
##                  Estimate  Std. Error t-value  Pr(>|t|)
## capi           4.0416e-02  1.1922e-02  3.3899  0.000932 ***
## labor          6.9453e-08  7.9967e-09  8.6851 1.588e-14 ***
## Overall.Score -3.4200e-02  5.9578e-03 -5.7404 6.580e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    26.608
## Residual Sum of Squares: 13.014
## R-Squared:      0.51089
## Adj. R-Squared: 0.48394
## F-statistic: 44.2192 on 3 and 127 DF, p-value: < 2.22e-16
```

Efectos Aleatorios

```
#ALEATORIOS
reg_re = plm(ln_gdp ~ capi+ labor+ Overall.Score, data = df, model = "random")
# En caso no corra, cambiamos de método
#reg_re <- plm(gdp ~ capi + labor + Overall.Score,
#               data = df,
#               model = "random",
#               random.method = "amemiya")
summary(reg_re)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = ln_gdp ~ capi + labor + Overall.Score, data = df,
##     model = "random")
##
## Balanced Panel: n = 5, T = 27, N = 135
##
## Effects:
##                  var std.dev share
## idiosyncratic 0.1025  0.3201     1
## individual    0.0000  0.0000     0
## theta: 0
##
## Residuals:
##      Min.   1st Qu.    Median   3rd Qu.      Max.
## -1.026705 -0.256018 -0.013113  0.305596  0.843490
##
## Coefficients:
##                  Estimate  Std. Error z-value  Pr(>|z|)
## (Intercept)    6.9663e+00  4.0193e-01 17.3323 < 2.2e-16 ***
## capi           5.3420e-02  1.2982e-02  4.1149 3.873e-05 ***
## labor          2.9672e-08  2.6963e-09 11.0047 < 2.2e-16 ***
## Overall.Score -5.1392e-03  5.8307e-03 -0.8814    0.3781
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    65.451
## Residual Sum of Squares: 21.436
## R-Squared:      0.6725
## Adj. R-Squared: 0.665
## Chisq: 268.995 on 3 DF, p-value: < 2.22e-16
```

Validación del modelo

```
#HIpotesis nula = se prefiere el modelo Pooled
#Hipotesis alternativa= se prefiere el modelo de efectod fijos
pFtest(reg_fe, reg_pool)
```

```
##
##  F test for individual effects
##
## data:  ln_gdp ~ capi + labor + Overall.Score
## F = 20.545, df1 = 4, df2 = 127, p-value = 4.495e-13
## alternative hypothesis: significant effects
```

```
#Se rechaza la hipótesis nula

#Test de Hausman
phtest(reg_re, reg_fe)
```

```
##
##  Hausman Test
##
## data:  ln_gdp ~ capi + labor + Overall.Score
## chisq = 133.71, df = 3, p-value < 2.2e-16
## alternative hypothesis: one model is inconsistent
```

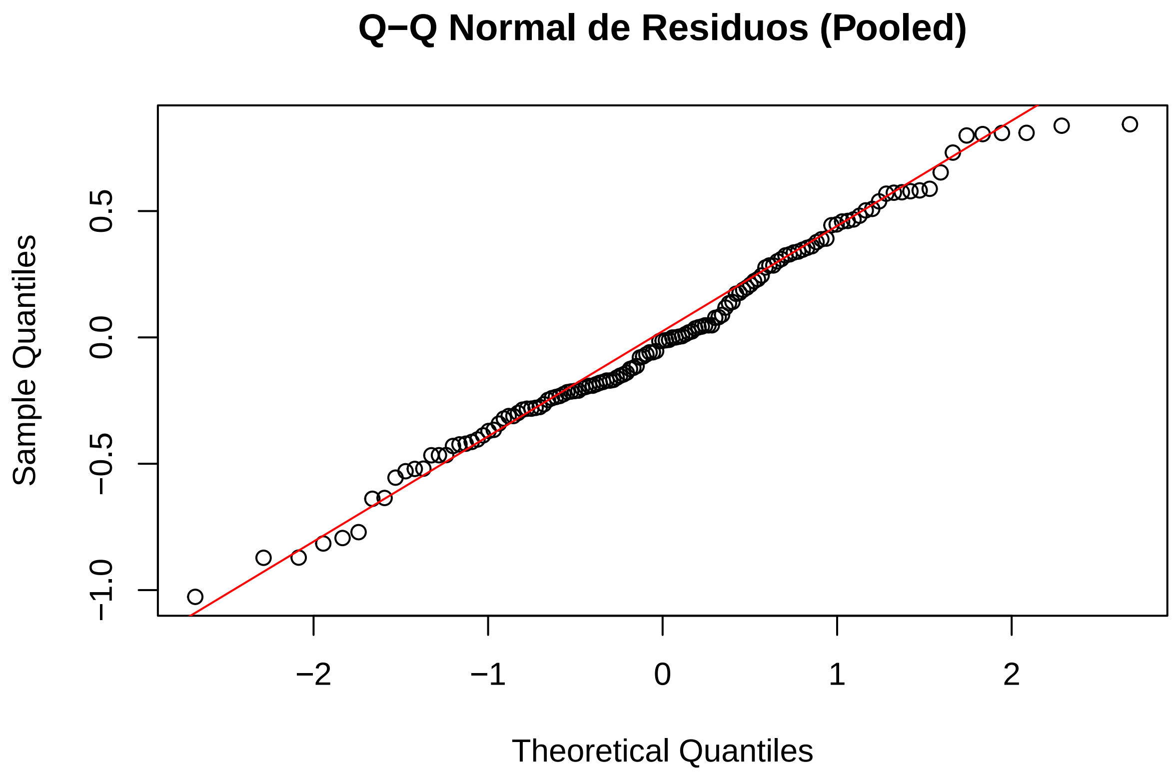
```
#HA = REF_FE
#HO = REG_R
```

#Graficos

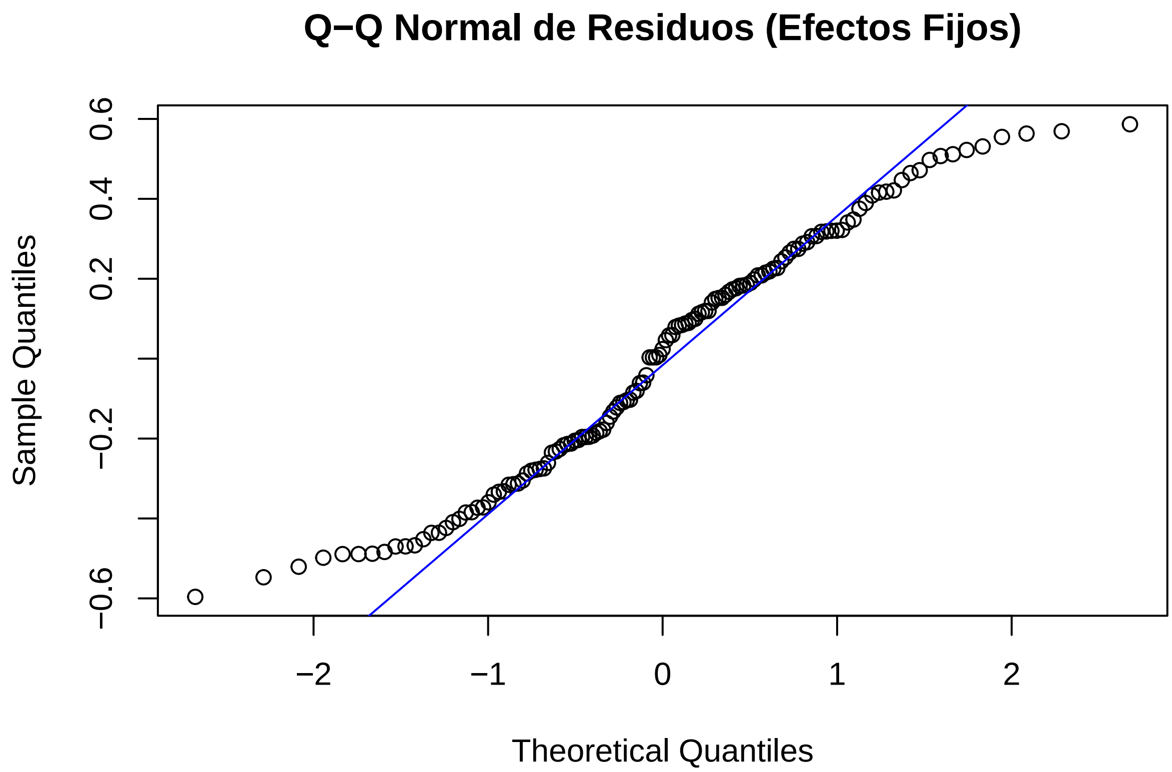
Gráfico de residuos

QQNORM

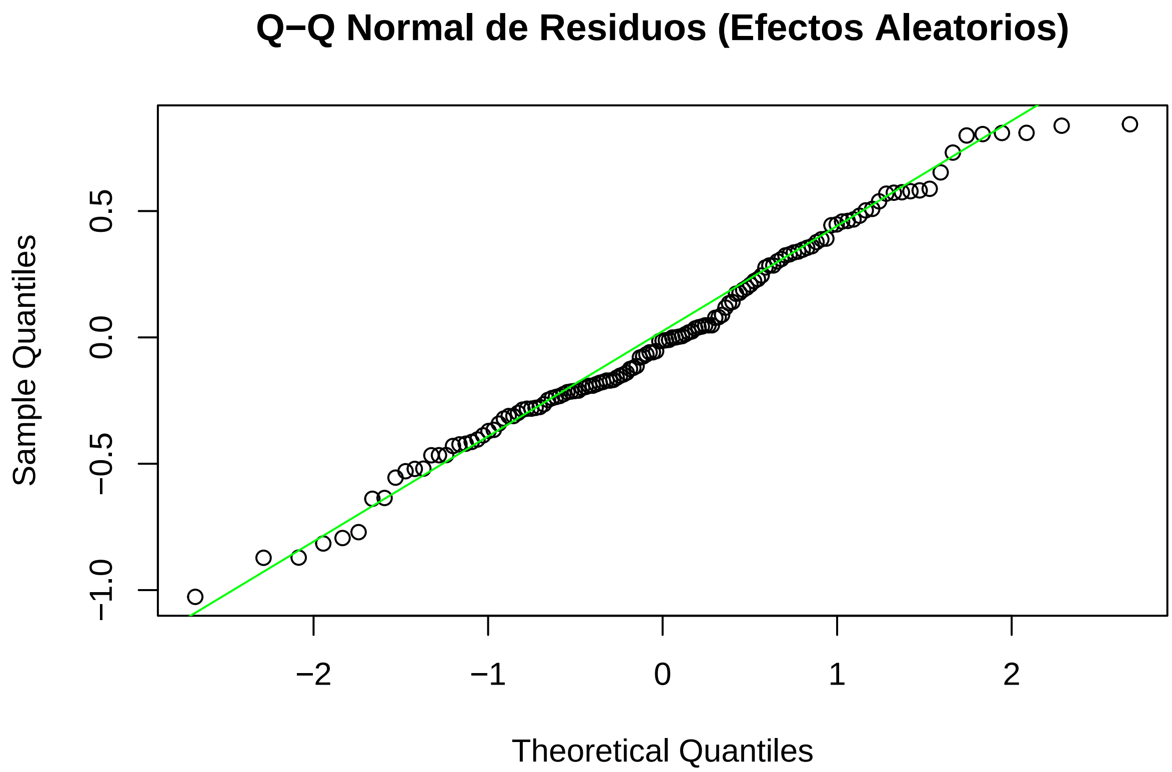
```
# --- Gráfico Q-Q Normal para el modelo Pooled ---
# Extraer los residuos del modelo Pooled
residuos_pool <- residuals(reg_pool)

# Crear el gráfico Q-Q normal
qqnorm(residuos_pool, main = "Q-Q Normal de Residuos (Pooled)")
qqline(residuos_pool, col = "red") # Agrega la línea de referencia
```




```r
# --- Gráfico Q-Q Normal para el modelo de Efectos Fijos ---
# Extraer los residuos del modelo de Efectos Fijos
residuos_fe <- residuals(reg_fe)

# Crear el gráfico Q-Q normal
qqnorm(residuos_fe, main = "Q-Q Normal de Residuos (Efectos Fijos)")
qqline(residuos_fe, col = "blue") # Puedes usar un color diferente si quieres
```

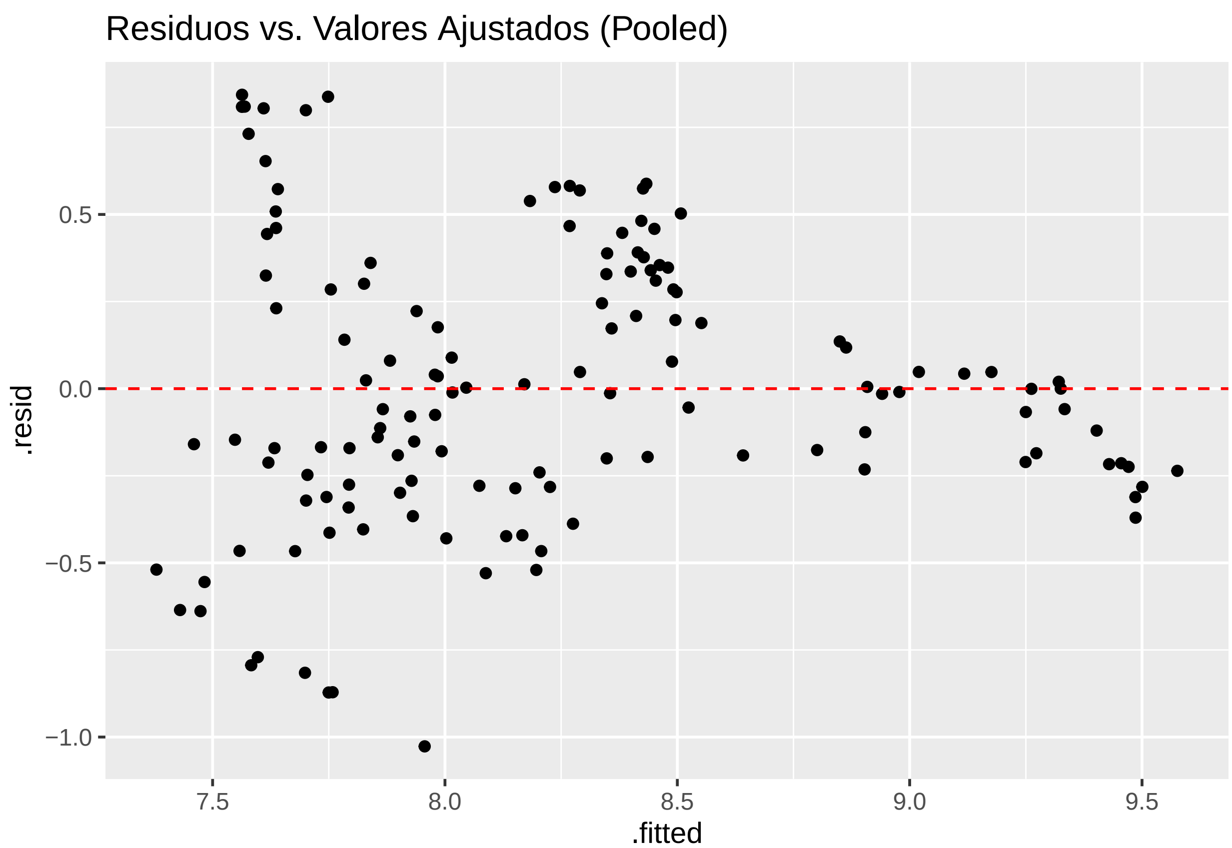



```r
# --- Gráfico Q-Q Normal para el modelo de Efectos Aleatorios ---
# Extraer los residuos del modelo de Efectos Aleatorios
residuos_re <- residuals(reg_re)

# Crear el gráfico Q-Q normal
qqnorm(residuos_re, main = "Q-Q Normal de Residuos (Efectos Aleatorios)")
qqline(residuos_re, col = "green") # Otro color para diferenciarlos
```

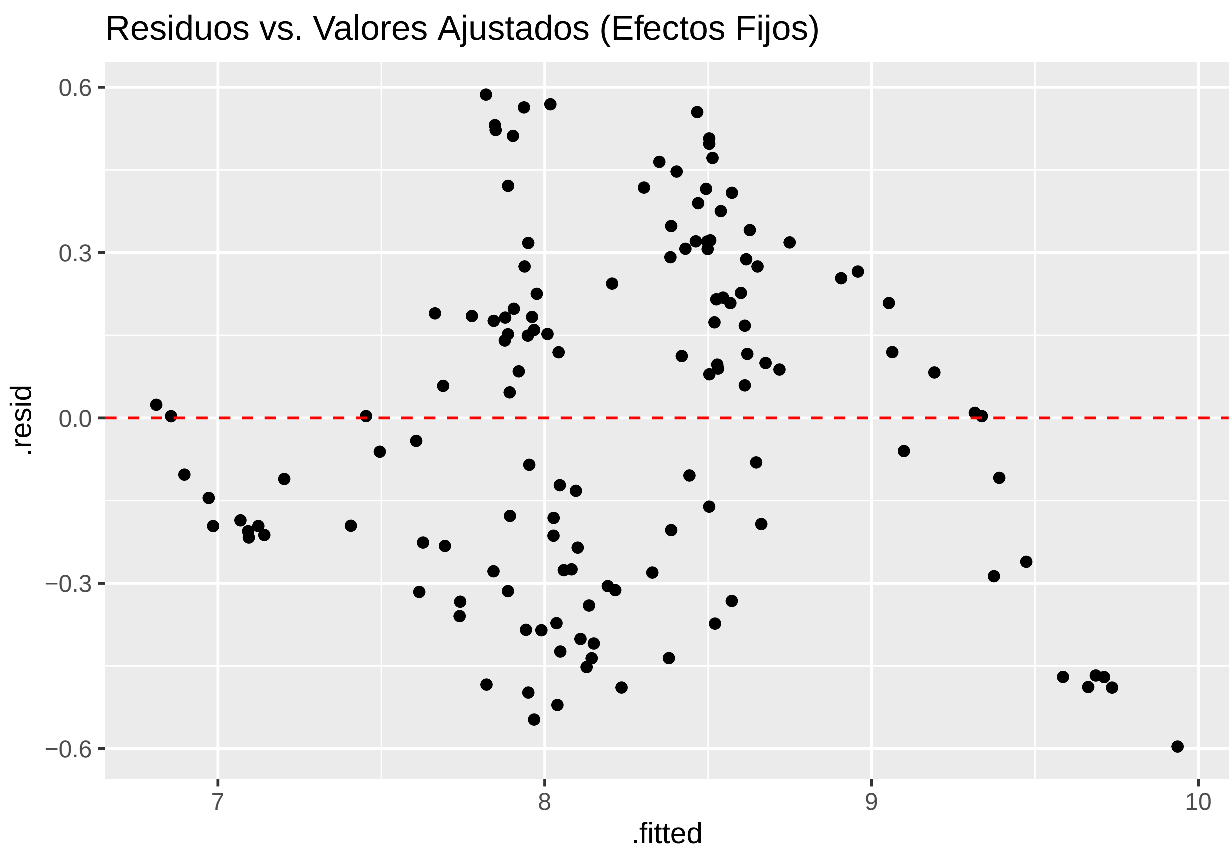


Residuos

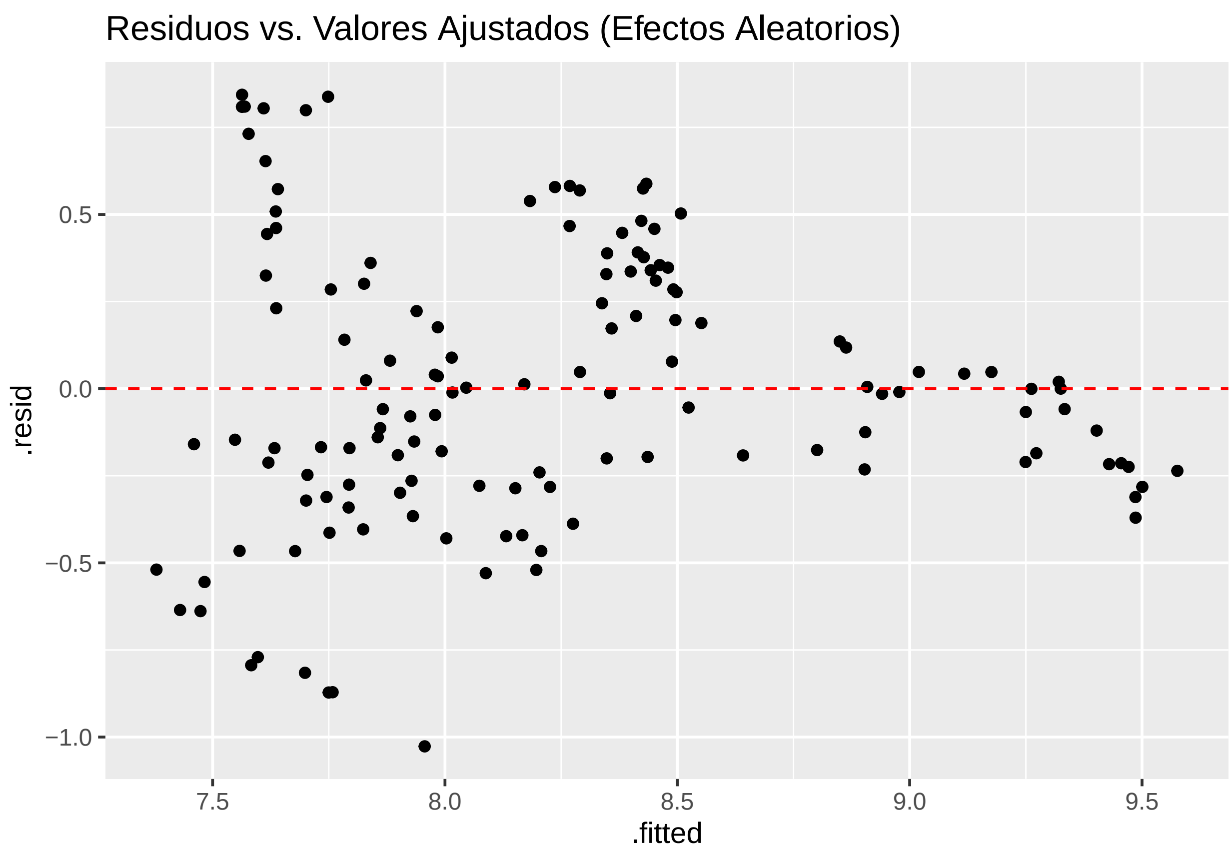
```
library(ggplot2) # Para gráficos más bonitos

# Para el modelo Pooled
library(broom)
aug_pool <- augment(reg_pool, df) # 'df' es tu conjunto de datos original
ggplot(aug_pool, aes(.fitted, .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, lty = 2, color = "red") +
  labs(title = "Residuos vs. Valores Ajustados (Pooled)")
```




```r
# Para el modelo de Efectos Fijos
aug_pool <- augment(reg_fe, df) # 'df' es tu conjunto de datos original
ggplot(aug_pool, aes(.fitted, .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, lty = 2, color = "red") +
  labs(title = "Residuos vs. Valores Ajustados (Efectos Fijos)")
```

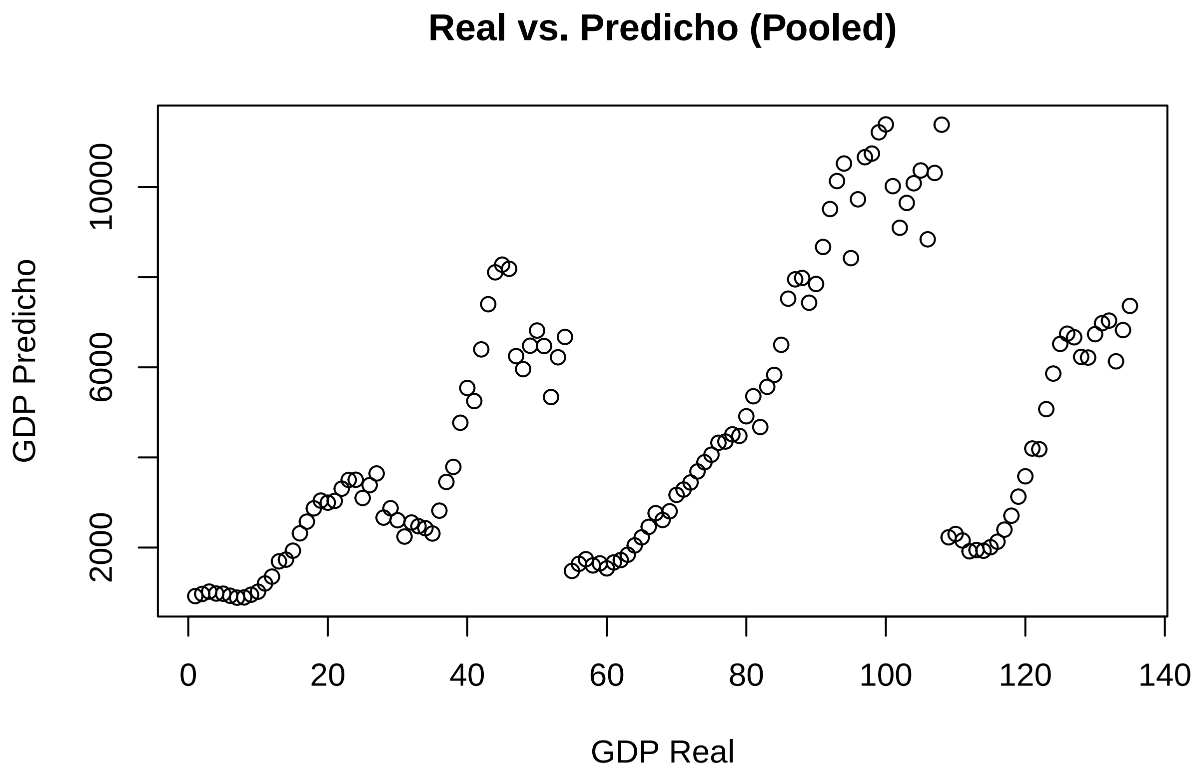



```r
# Para el modelo de Efectos Aleatorios
aug_pool <- augment(reg_re, df) # 'df' es tu conjunto de datos original
ggplot(aug_pool, aes(.fitted, .resid)) +
   geom_point() +
   geom_hline(yintercept = 0, lty = 2, color = "red") +
   labs(title = "Residuos vs. Valores Ajustados (Efectos Aleatorios)")
```

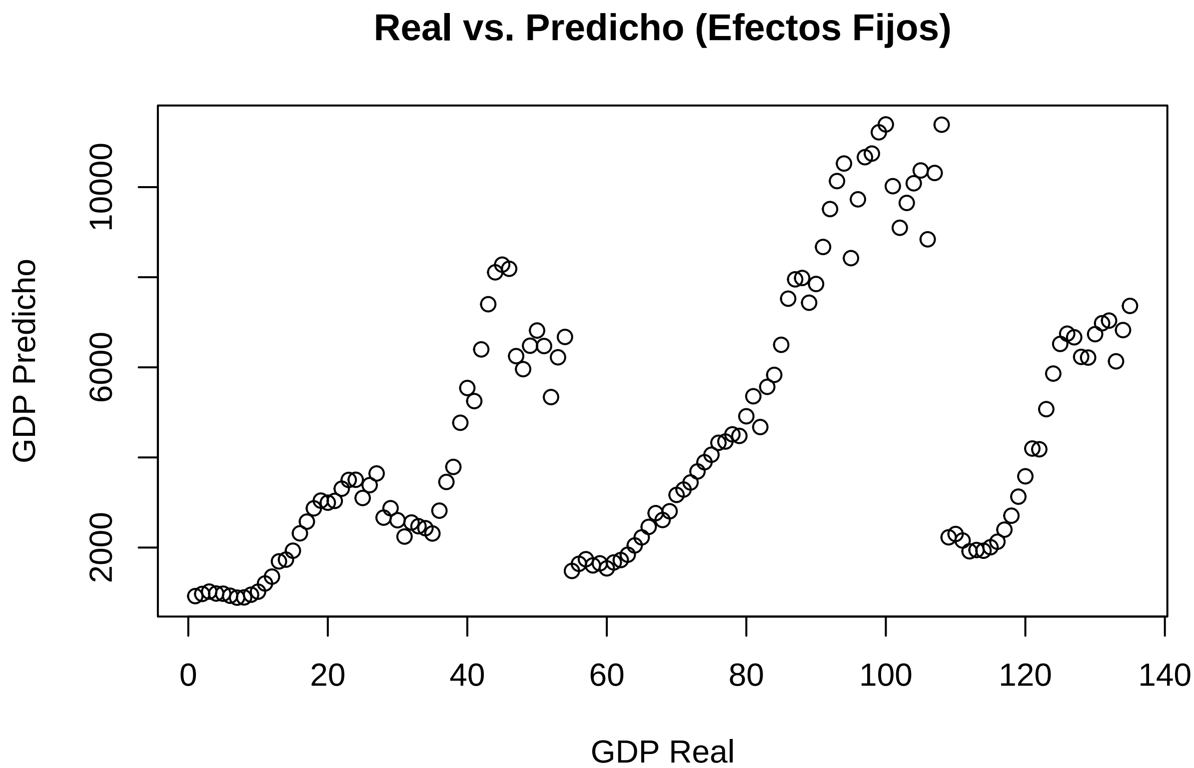


Valores predichos vs valores residuales

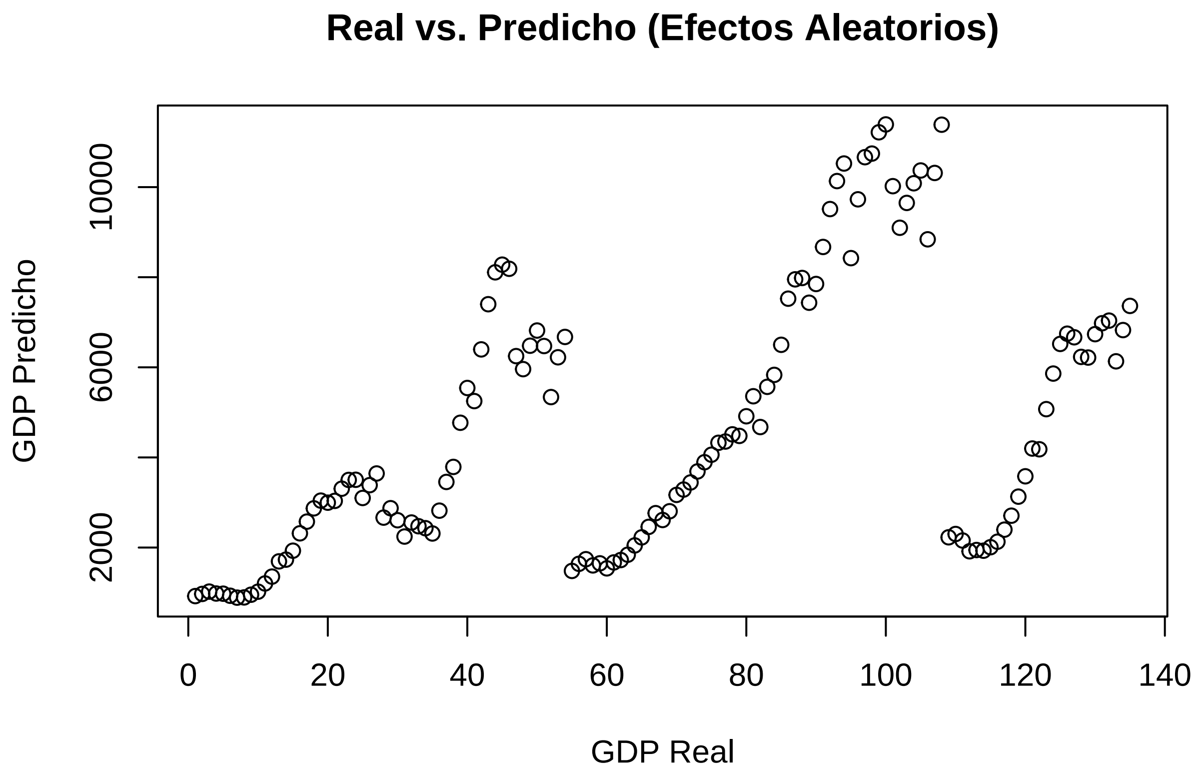
```r
# Para el modelo Pooled
plot(df$gdp, reg_pool$fitted.values,
     xlab = "GDP Real", ylab = "GDP Predicho",
     main = "Real vs. Predicho (Pooled)")
abline(a = 0, b = 1, col = "blue", lty = 2) # Línea de 45 grados
```




```r
# Para el modelo de Efectos Fijos
plot(df$gdp, reg_fe$fitted.values,
     xlab = "GDP Real", ylab = "GDP Predicho",
     main = "Real vs. Predicho (Efectos Fijos)")
abline(a = 0, b = 1, col = "blue", lty = 2)
```




```
# Para el modelo de Efectos Aleatorios
plot(df$gdp, reg_re$fitted.values,
     xlab = "GDP Real", ylab = "GDP Predicho",
     main = "Real vs. Predicho (Efectos Aleatorios)")
abline(a = 0, b = 1, col = "blue", lty = 2)
```



Con art

Pooled

```r
reg_pool = plm(ln_gdp ~ capi+ art+ Overall.Score, data = df, model = "pooling")
summary(reg_pool)
```

```
## Pooling Model
##
## Call:
## plm(formula = ln_gdp ~ capi + art + Overall.Score, data = df,
##     model = "pooling")
##
## Balanced Panel: n = 5, T = 27, N = 135
##
## Residuals:
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -1.35329 -0.32070  0.01705  0.31101  0.84548
##
## Coefficients:
##                  Estimate  Std. Error t-value  Pr(>|t|)
## (Intercept)    6.5526e+00  4.3597e-01 15.0297 < 2.2e-16 ***
## capi           7.7518e-02  1.3565e-02  5.7145 7.064e-08 ***
```

```
## art            7.4598e-05  8.5878e-06  8.6865 1.305e-14 ***
## Overall.Score -9.7818e-04  6.3869e-03 -0.1532    0.8785
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    65.451
## Residual Sum of Squares: 26.175
## R-Squared:      0.60008
## Adj. R-Squared: 0.59093
## F-statistic: 65.5229 on 3 and 131 DF, p-value: < 2.22e-16
```

Regresión de efectos fijos

```
#ESTIMACIÖN DE EFECTOS FIJOS
reg_fe = plm(ln_gdp ~ capi+ art+ Overall.Score, data = df, model = "within")
summary(reg_fe)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = ln_gdp ~ capi + art + Overall.Score, data = df,
##     model = "within")
##
## Balanced Panel: n = 5, T = 27, N = 135
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -0.7071098 -0.2927096  0.0062864  0.2456395  0.7000305
##
## Coefficients:
##                  Estimate  Std. Error t-value  Pr(>|t|)
## capi           5.3489e-02  1.3343e-02  4.0087 0.0001035 ***
## art            6.2727e-05  1.1548e-05  5.4318 2.736e-07 ***
## Overall.Score -3.3991e-02  6.8844e-03 -4.9374 2.439e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    26.608
## Residual Sum of Squares: 16.833
## R-Squared:      0.36737
## Adj. R-Squared: 0.3325
## F-statistic: 24.5826 on 3 and 127 DF, p-value: 1.3117e-12
```

Efectos Aleatorios

```
#ALEATORIOS
reg_re = plm(ln_gdp ~ capi+art+Overall.Score, data = df, model = "random")

# En caso no corra, cambiamos de método
```

```r
#reg_re <- plm(gdp ~ capi + art + Overall.Score,
#               data = df,
#               model = "random",
#               random.method = "amemiya")
summary(reg_re)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = ln_gdp ~ capi + art + Overall.Score, data = df,
##     model = "random")
##
## Balanced Panel: n = 5, T = 27, N = 135
##
## Effects:
##                  var std.dev share
## idiosyncratic 0.1325  0.3641     1
## individual    0.0000  0.0000     0
## theta: 0
##
## Residuals:
##     Min.  1st Qu.   Median  3rd Qu.     Max.
## -1.35329 -0.32070  0.01705  0.31101  0.84548
##
## Coefficients:
##                  Estimate  Std. Error z-value Pr(>|z|)
## (Intercept)    6.5526e+00  4.3597e-01 15.0297  < 2e-16 ***
## capi           7.7518e-02  1.3565e-02  5.7145  1.1e-08 ***
## art            7.4598e-05  8.5878e-06  8.6865  < 2e-16 ***
## Overall.Score -9.7818e-04  6.3869e-03 -0.1532   0.8783
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    65.451
## Residual Sum of Squares: 26.175
## R-Squared:      0.60008
## Adj. R-Squared: 0.59093
## Chisq: 196.569 on 3 DF, p-value: < 2.22e-16
```

Validación del modelo

```r
#HIpotesis nula = se prefiere el modelo Pooled
#Hipotesis alternativa= se prefiere el modelo de efectod fijos
pFtest(reg_fe, reg_pool)
```

```
##
##  F test for individual effects
##
## data:  ln_gdp ~ capi + art + Overall.Score
```

```
## F = 17.62, df1 = 4, df2 = 127, p-value = 1.585e-11
## alternative hypothesis: significant effects
```

```r
#Se rechaza la hipótesis nula

#Test de Hausman
phtest(reg_re, reg_fe)
```

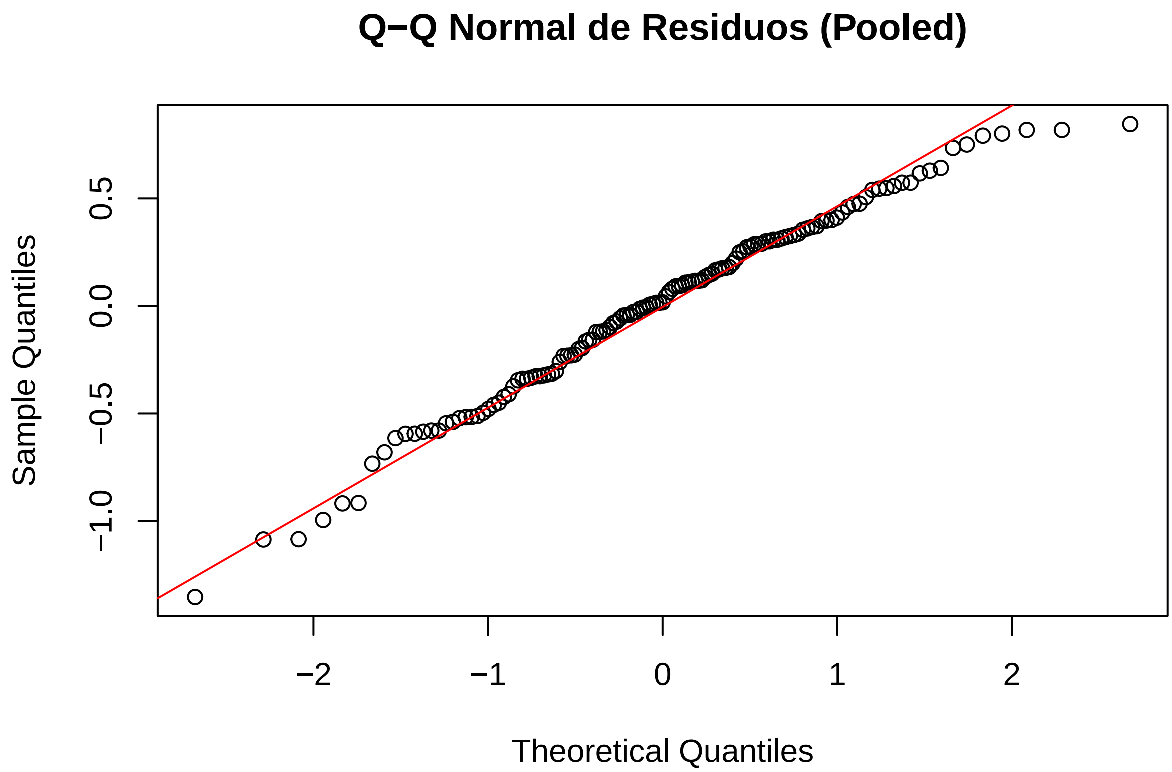
```
##
##  Hausman Test
##
## data:  ln_gdp ~ capi + art + Overall.Score
## chisq = 71.748, df = 3, p-value = 1.803e-15
## alternative hypothesis: one model is inconsistent
```

```r
#HA = REF_FE
#HO = REG_R
```

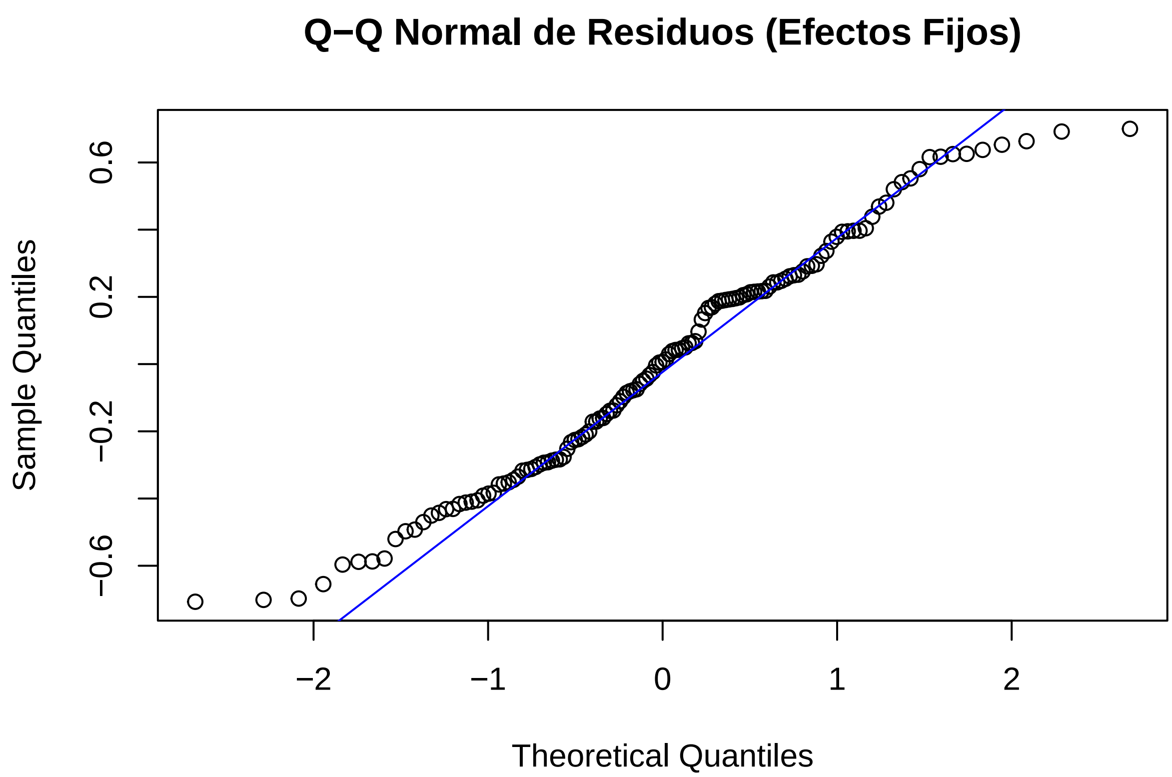
#Graficos

QQNORM

```r
# --- Gráfico Q-Q Normal para el modelo Pooled ---
# Extraer los residuos del modelo Pooled
residuos_pool <- residuals(reg_pool)

# Crear el gráfico Q-Q normal
qqnorm(residuos_pool, main = "Q-Q Normal de Residuos (Pooled)")
qqline(residuos_pool, col = "red") # Agrega la línea de referencia
```




```r
# --- Gráfico Q-Q Normal para el modelo de Efectos Fijos ---
# Extraer los residuos del modelo de Efectos Fijos
residuos_fe <- residuals(reg_fe)

# Crear el gráfico Q-Q normal
qqnorm(residuos_fe, main = "Q-Q Normal de Residuos (Efectos Fijos)")
qqline(residuos_fe, col = "blue") # Puedes usar un color diferente si quieres
```




```r
# --- Gráfico Q-Q Normal para el modelo de Efectos Aleatorios ---
# Extraer los residuos del modelo de Efectos Aleatorios
residuos_re <- residuals(reg_re)

# Crear el gráfico Q-Q normal
qqnorm(residuos_re, main = "Q-Q Normal de Residuos (Efectos Aleatorios)")
qqline(residuos_re, col = "green") # Otro color para diferenciarlos
```

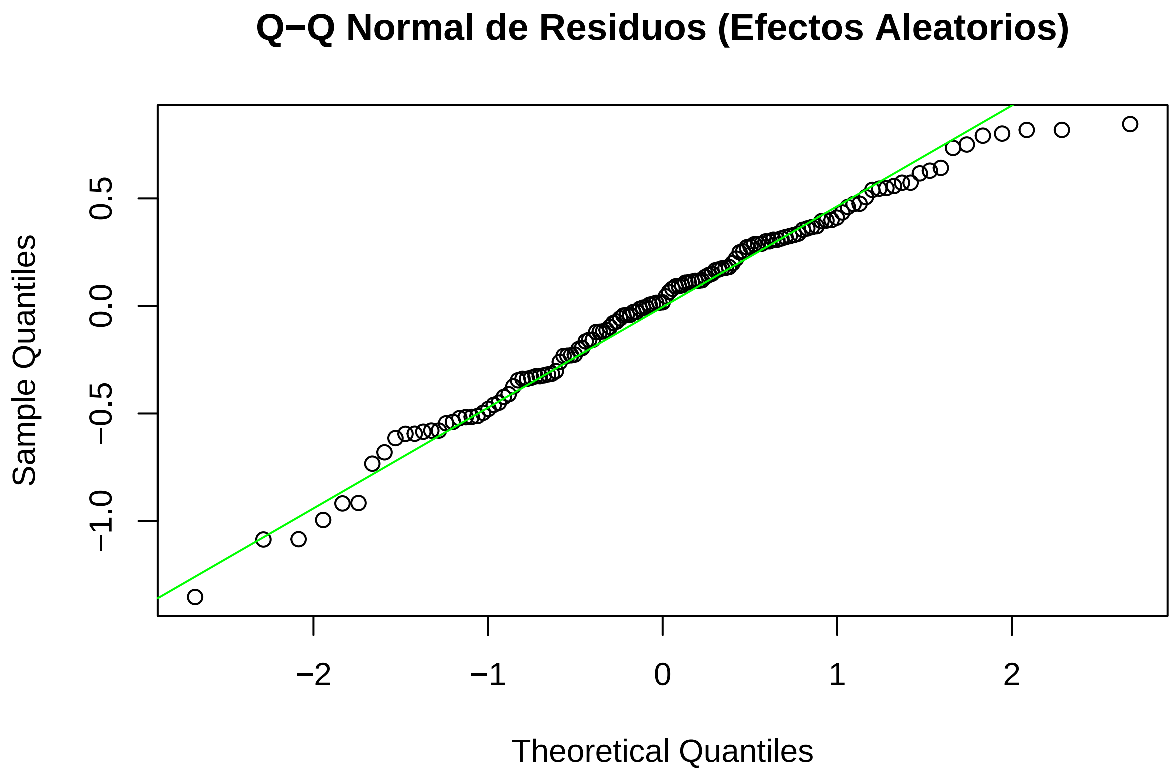
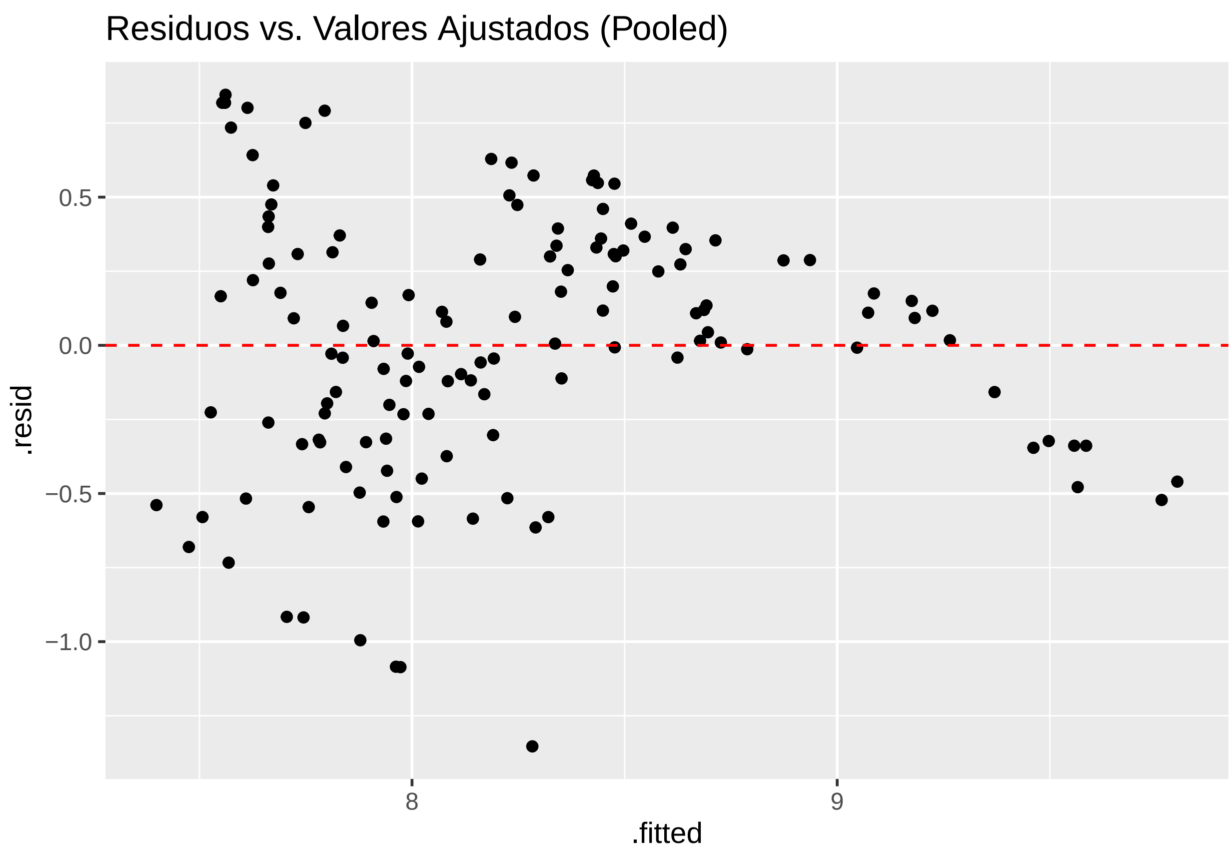
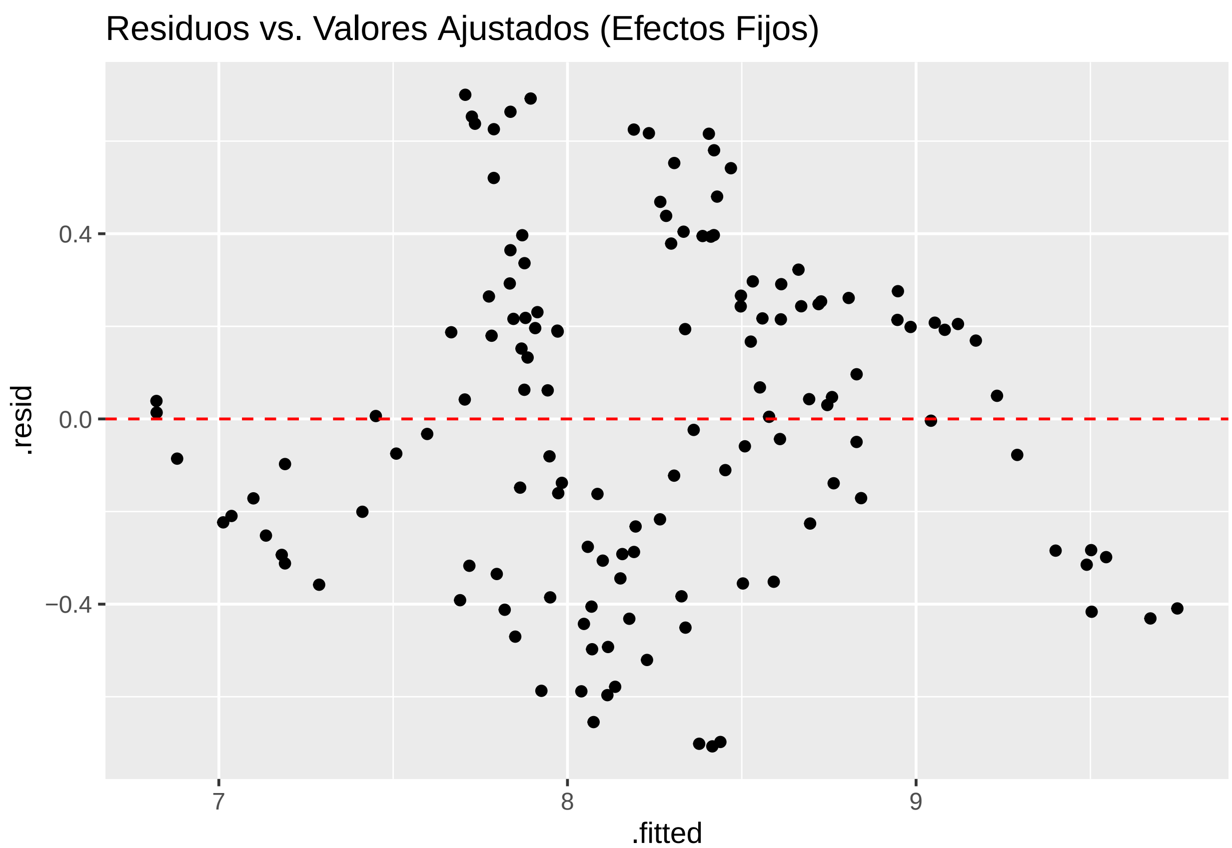


Gráfico de residuos

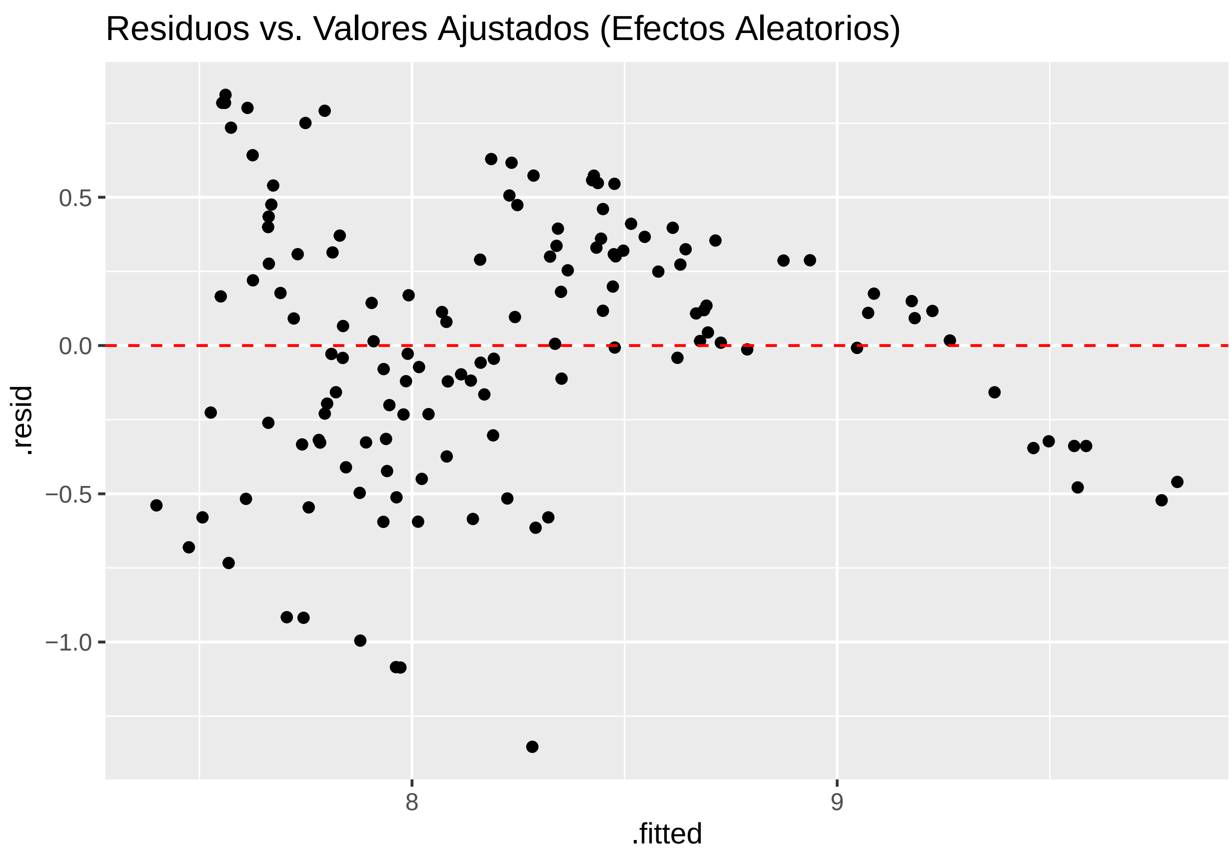
```
library(ggplot2) # Para gráficos más bonitos

# Para el modelo Pooled
library(broom)
aug_pool <- augment(reg_pool, df) # 'df' es tu conjunto de datos original
ggplot(aug_pool, aes(.fitted, .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, lty = 2, color = "red") +
  labs(title = "Residuos vs. Valores Ajustados (Pooled)")
```




```r
# Para el modelo de Efectos Fijos
aug_pool <- augment(reg_fe, df) # 'df' es tu conjunto de datos original
ggplot(aug_pool, aes(.fitted, .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, lty = 2, color = "red") +
  labs(title = "Residuos vs. Valores Ajustados (Efectos Fijos)")
```

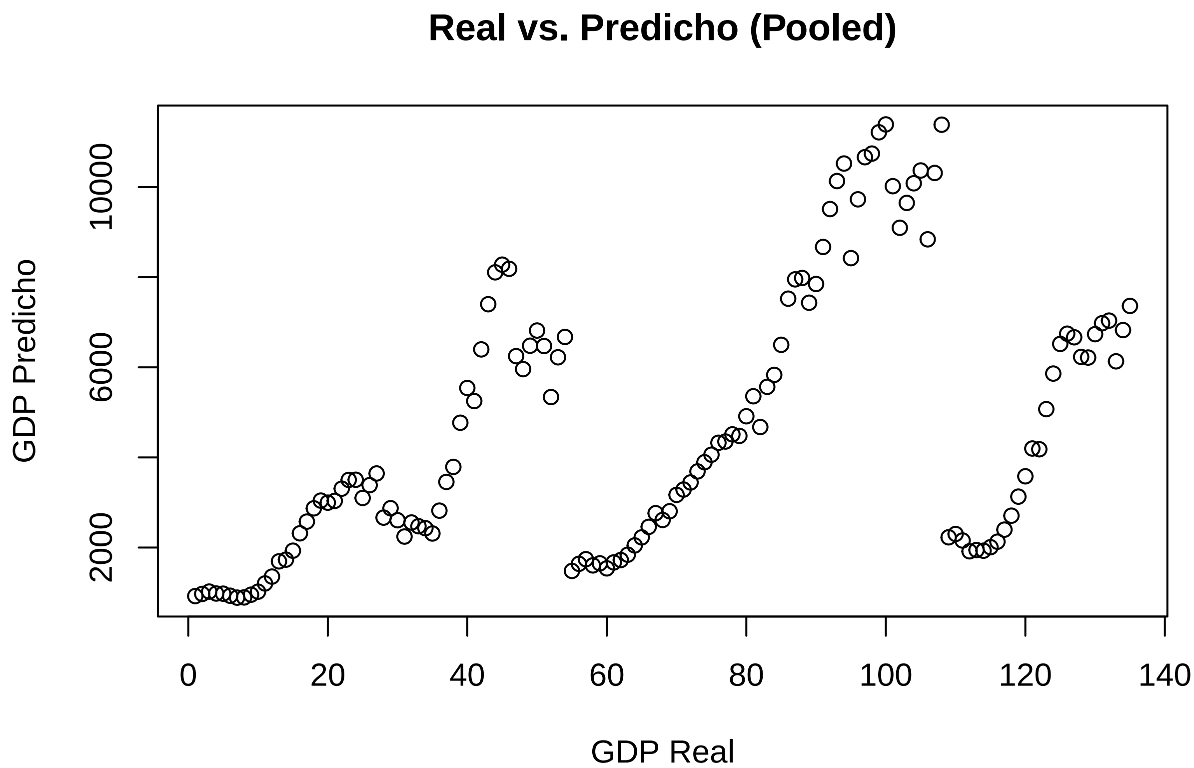



```r
# Para el modelo de Efectos Aleatorios
aug_pool <- augment(reg_re, df) # 'df' es tu conjunto de datos original
ggplot(aug_pool, aes(.fitted, .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, lty = 2, color = "red") +
  labs(title = "Residuos vs. Valores Ajustados (Efectos Aleatorios)")
```

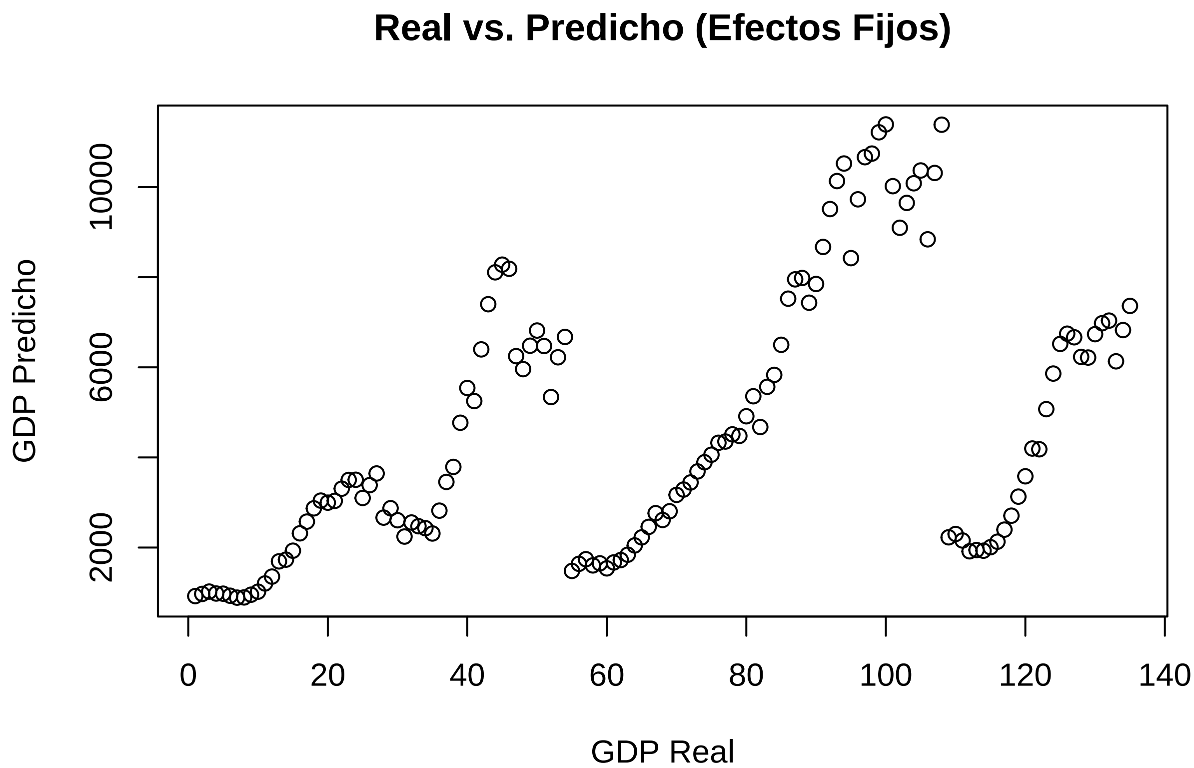


Valores predichos vs valores residuales

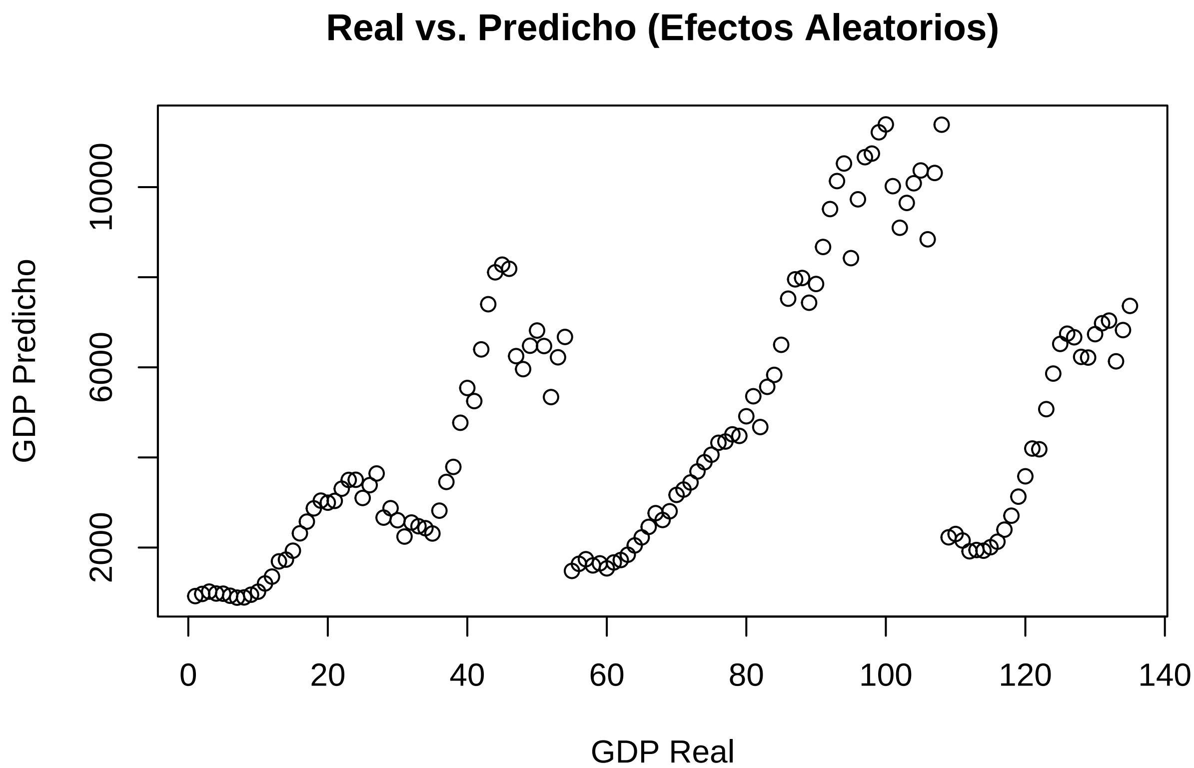
```r
# Para el modelo Pooled
plot(df$gdp, reg_pool$fitted.values,
     xlab = "GDP Real", ylab = "GDP Predicho",
     main = "Real vs. Predicho (Pooled)")
abline(a = 0, b = 1, col = "blue", lty = 2) # Línea de 45 grados
```




```r
# Para el modelo de Efectos Fijos
plot(df$gdp, reg_fe$fitted.values,
     xlab = "GDP Real", ylab = "GDP Predicho",
     main = "Real vs. Predicho (Efectos Fijos)")
abline(a = 0, b = 1, col = "blue", lty = 2)
```




```r
# Para el modelo de Efectos Aleatorios
plot(df$gdp, reg_re$fitted.values,
     xlab = "GDP Real", ylab = "GDP Predicho",
     main = "Real vs. Predicho (Efectos Aleatorios)")
abline(a = 0, b = 1, col = "blue", lty = 2)
```


Real vs. Predicho (Efectos Aleatorios)
GDP Predicho
2000
6000
10000
0
20
40
60
80
100
120
140
GDP Real