Modelos lineales

Armando Ocampo

Librerias de trabajo

Antes de comenzar a trabajar, te recomiendo abrir las siguientes librerias. Si no cuentas con alguna de ellas es posible instalarla mediante la función *install.packages()*

```
library(readr)
library(dplyr)
library(corrplot)
library(ggplot2)
library(scales)
library(ggpubr)
library(broom)
```

Asimismo, se agregan dos conjuntos de datos. Los cuales debes descargar y guardar en la carpeta *clean_data*.

Modelos lineales

Para este apartado utilizaremos el conjunto de datos de vacunación obtenido del portal **Our World in Data**, el cual se conforma por información de 171 países, agregando las variables esperanza de vida, producto interno bruto *per capita* y porcentaje de vacunación de 13 inmunizaciones. Este dataset tiene actualización al 31 de diciembre de 2019. El formato es un archivo de valores separados por coma (*comma separate values, csv*). Antes de comenzar, lo llamaremos a nuestro ambiente de trabajo.

```
vacunas_df <- read_csv('../data/dataset_vacunas.csv')
```

Para facilitar el uso de las variables del conjunto de datos utilizaremos la función *attach()* de la paquetería *base*. Esta permite que cada variable del dataset se vuelva un vector independiente, sin saturar la memoria del ambiente.

```
attach(vacunas_df)
```

Los modelos lineales describen la relación de la variable respuesta (dependiente) y la(s) variable(s) explicativa(s) (independiente). Esta relación puede ser positiva, negativa o estar ausente. En R, este tipo de modelos se realiza mediante la función *lm()* de la paquetería *stats*. Los argumentos que se colocan son los siguientes. *formula* = y_x, describe la relación de la variable "y" con la variable "x" ($_y x$, este elemento indica, en medida que x explica y). El siguiente argumento, *data*= detalla el conjunto de datos a utilizar. En el siguiente ejemplo se creará un modelo lineal que exlique la relación de la esperanza de vida con el producto interno bruto en el conjunto de datos de vacunas.

```
gdp_vs_life <- lm(Life_expectancy~GDP, data = vacunas_df)
```

Para extraer la información del modelo se utilizan las funciones *print()*, *summary()* y *tidy()*.

```
summary(gdp_vs_life)
```

```
##
## Call:
## lm(formula = Life_expectancy ~ GDP, data = vacunas_df)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.901  -3.883   1.317   4.144   8.623
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6.756e+01  5.798e-01  116.53   <2e-16 ***
## GDP         2.485e-04  1.962e-05   12.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.386 on 169 degrees of freedom
## Multiple R-squared:  0.4869, Adjusted R-squared:  0.4839
## F-statistic: 160.4 on 1 and 169 DF,  p-value: < 2.2e-16
```

```
confint(gdp_vs_life)
```

```
##                    2.5 %       97.5 %
## (Intercept) 6.641799e+01 6.870703e+01
## GDP         2.097573e-04 2.872285e-04
```

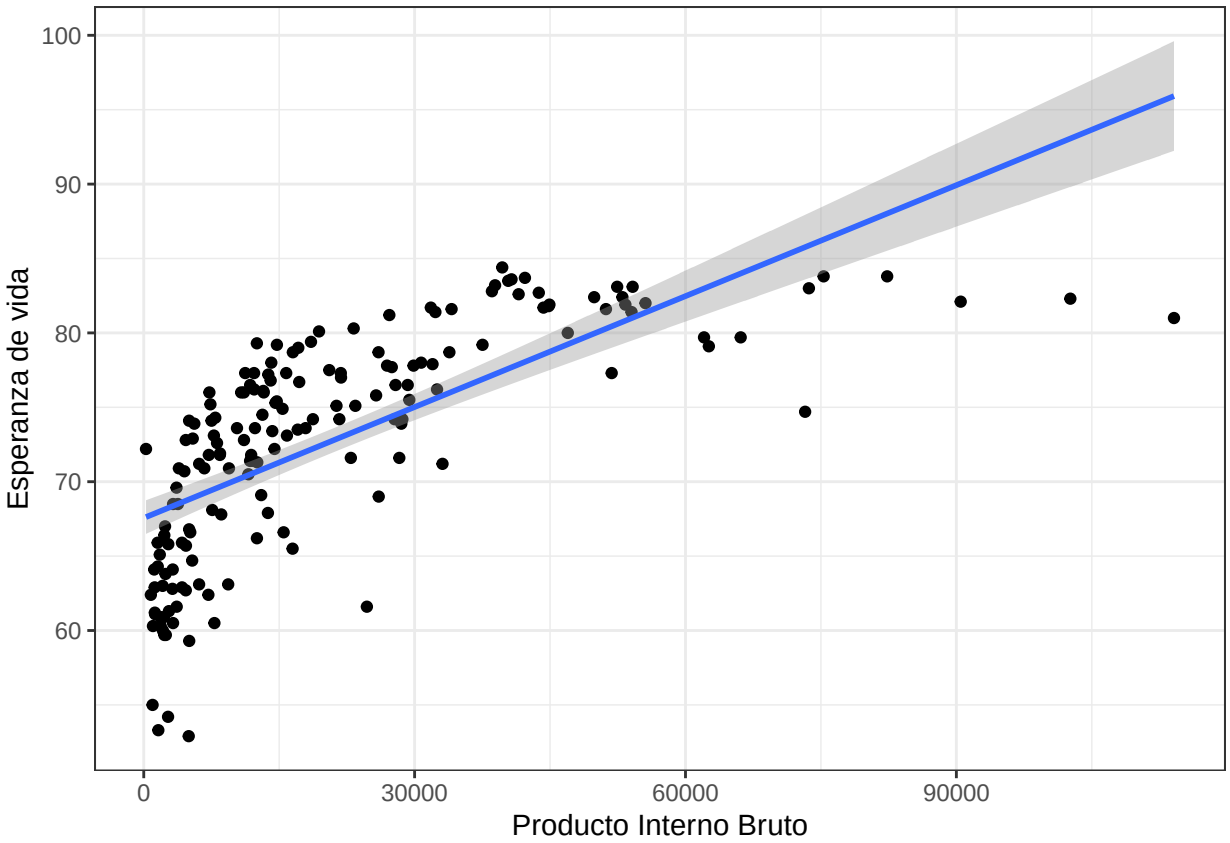
```
tidy(gdp_vs_life)
```

```
## # A tibble: 2 x 5
##   term         estimate std.error statistic   p.value
##   <chr>           <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept) 67.6      0.580         117.  2.30e-163
## 2 GDP          0.000248 0.0000196      12.7 2.83e- 26
```

Además de los estadísticos de confianza, estas funciones generan los coeficientes que permiten generar la función que explica el modelo. Proporcionando la intercepción con el eje de las "y", y la relación de entre variables.

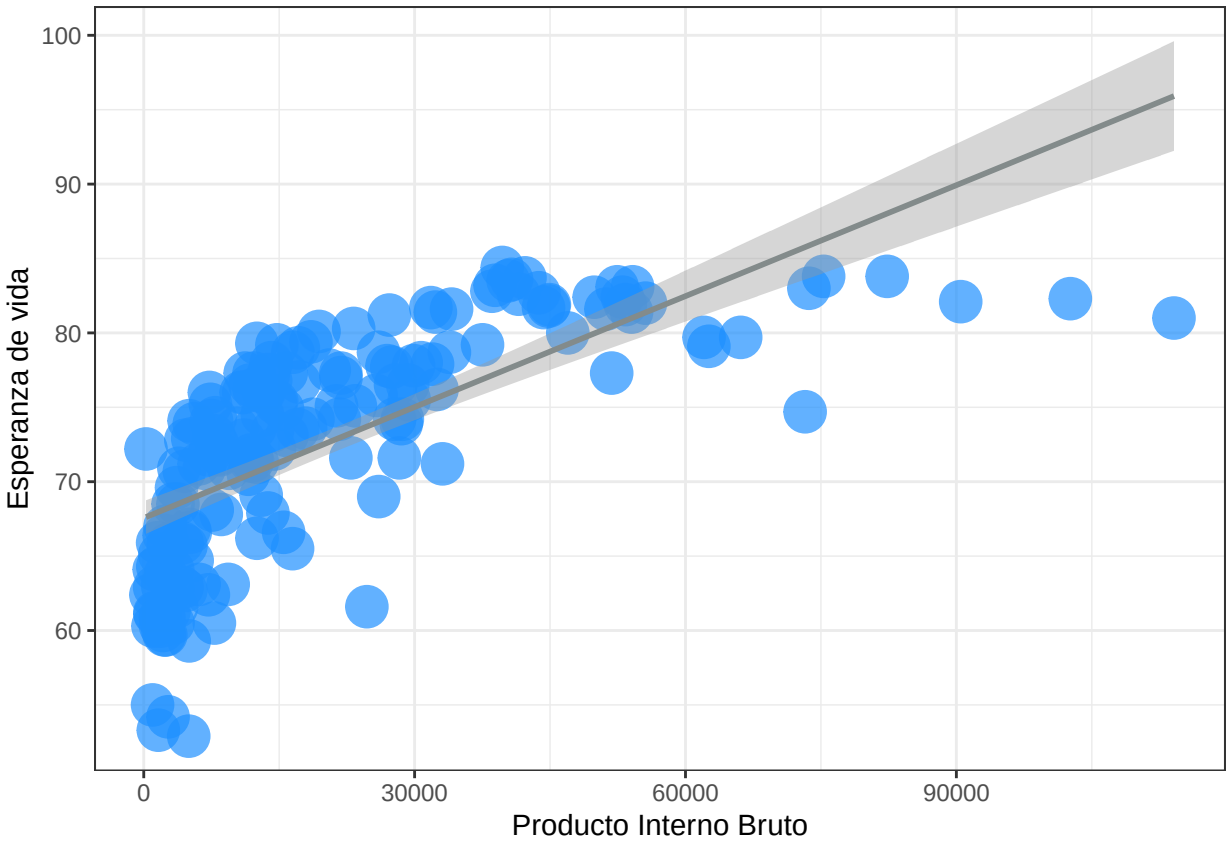
De esta manera obtenemos la siguiente fórmula $y = (0.000248)x + 67.6$. Posteriormente, podemos graficar el modelo. Para esto generaremos un gráfico de dispersión en la paquetería *ggplot2*, agregando la función *geom_smooth(method = 'glm')*, este argumento crea un modelo lineal generalizado.

```
ggplot(vacunas_df, aes(x = GDP, y = Life_expectancy)) +
  geom_point()+
  geom_smooth(method = 'glm')+
  xlab('Producto Interno Bruto')+
  ylab('Esperanza de vida')+
  theme_bw()
```



Para una mejor visualización se cambiará el color y tamaño de los puntos. Así como el color de la línea del modelo.

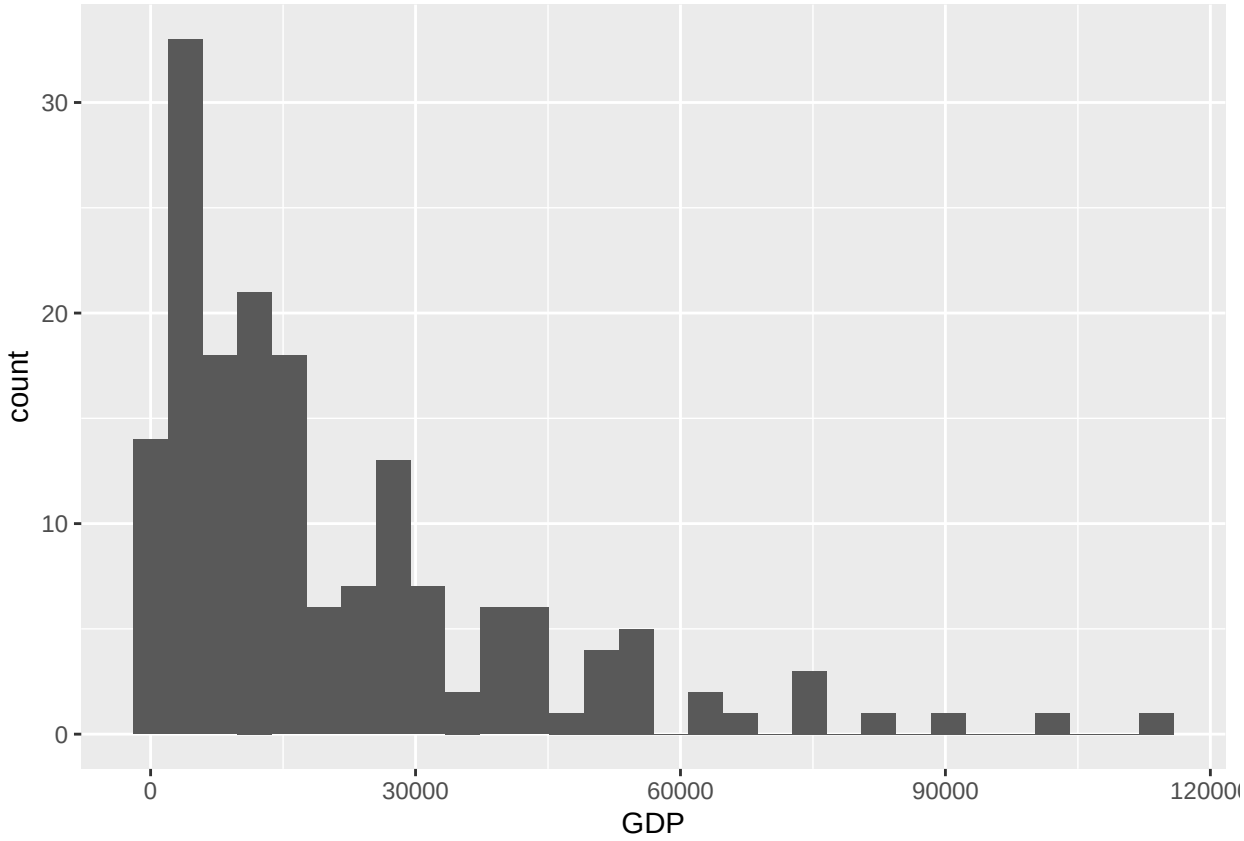
```
ggplot(vacunas_df, aes(x = GDP, y = Life_expectancy)) +
  geom_point(size = 7, color = 'dodgerblue', alpha = 0.7)+
  geom_smooth(method = 'glm', color = 'azure4')+
  xlab('Producto Interno Bruto')+
  ylab('Esperanza de vida')+
  theme_bw()
```



Uno de los puntos a resaltar es que la mayoría de los datos del producto interno bruto se encuentran desplazados a la izquierda. Para una mejor visualización se graficará la distribución de los datos.

```
ggplot(vacunas_df, aes(GDP)) +
  geom_histogram()
```

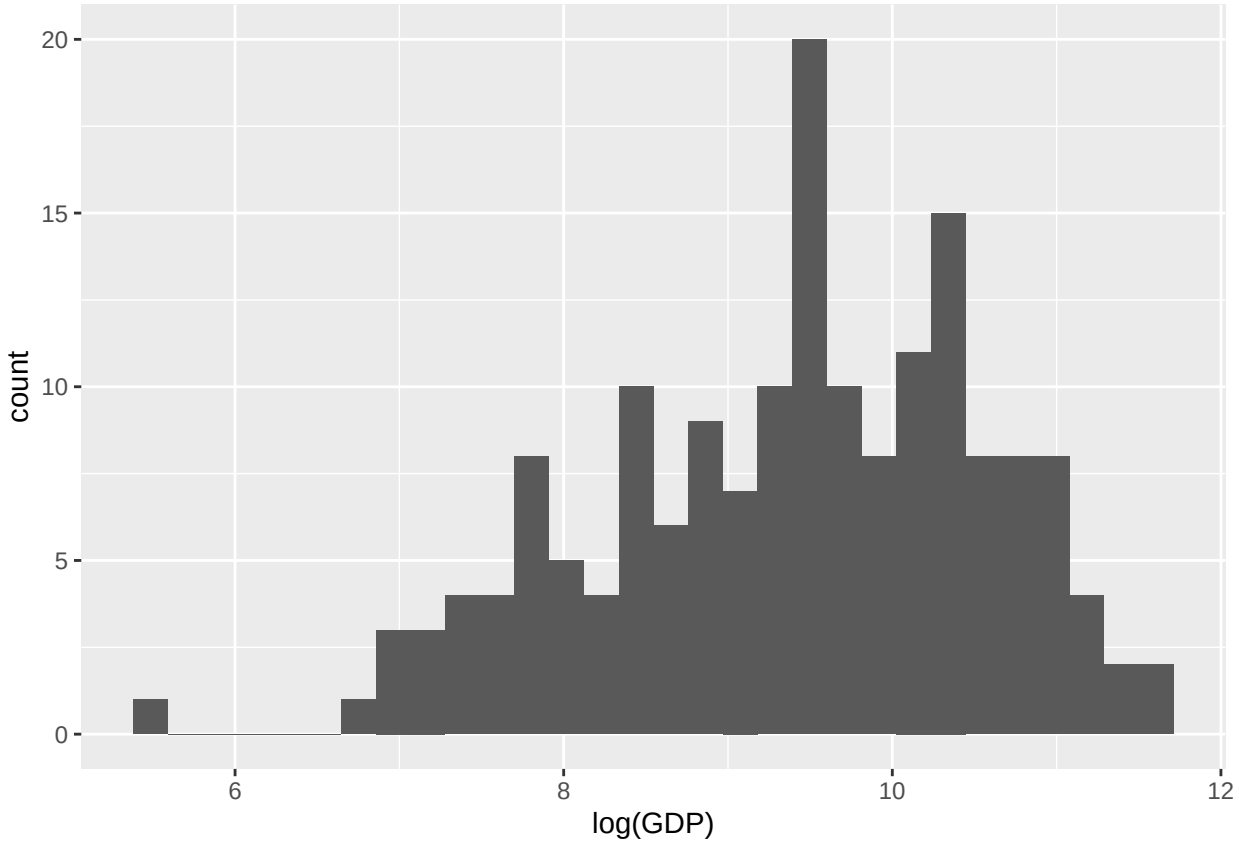
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Al observar la distribución se confirma el sesgo que presentan los datos. Cuando esto ocurre es válido realizar una transformación de los datos. Para esto se utiliza la transformación logarítmica o cuadrática. En este caso se realizará una transformación logarítmica de los datos.

```
ggplot(vacunas_df, aes(log(GDP)))+
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Los datos no siguen una distribución simétrica. No obstante, se ha reducido el sesgo de distribución. A continuación, se repetirá el modelo lineal, no obstante, los datos de PIB presentarán una transformación logarítmica.

```
gdp_vs_life_log <- lm(Life_expectancy~log(GDP), data = vacunas_df)
```

De la misma forma, se extraerán los estadísticos e información del modelo. Donde la nueva función se explica de la siguiente manera $y = (5.24)x + 23.6$

```
summary(gdp_vs_life_log)
```

```
##
## Call:
## lm(formula = Life_expectancy ~ log(GDP), data = vacunas_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2579  -2.0333   0.7346   3.0888  19.6932
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  23.5638     2.6597    8.86 1.08e-15 ***
## log(GDP)      5.2378     0.2812   18.63  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.303 on 169 degrees of freedom
## Multiple R-squared:  0.6725, Adjusted R-squared:  0.6706
```

```
## F-statistic:   347 on 1 and 169 DF,  p-value: < 2.2e-16
```

```
confint(gdp_vs_life_log)
```

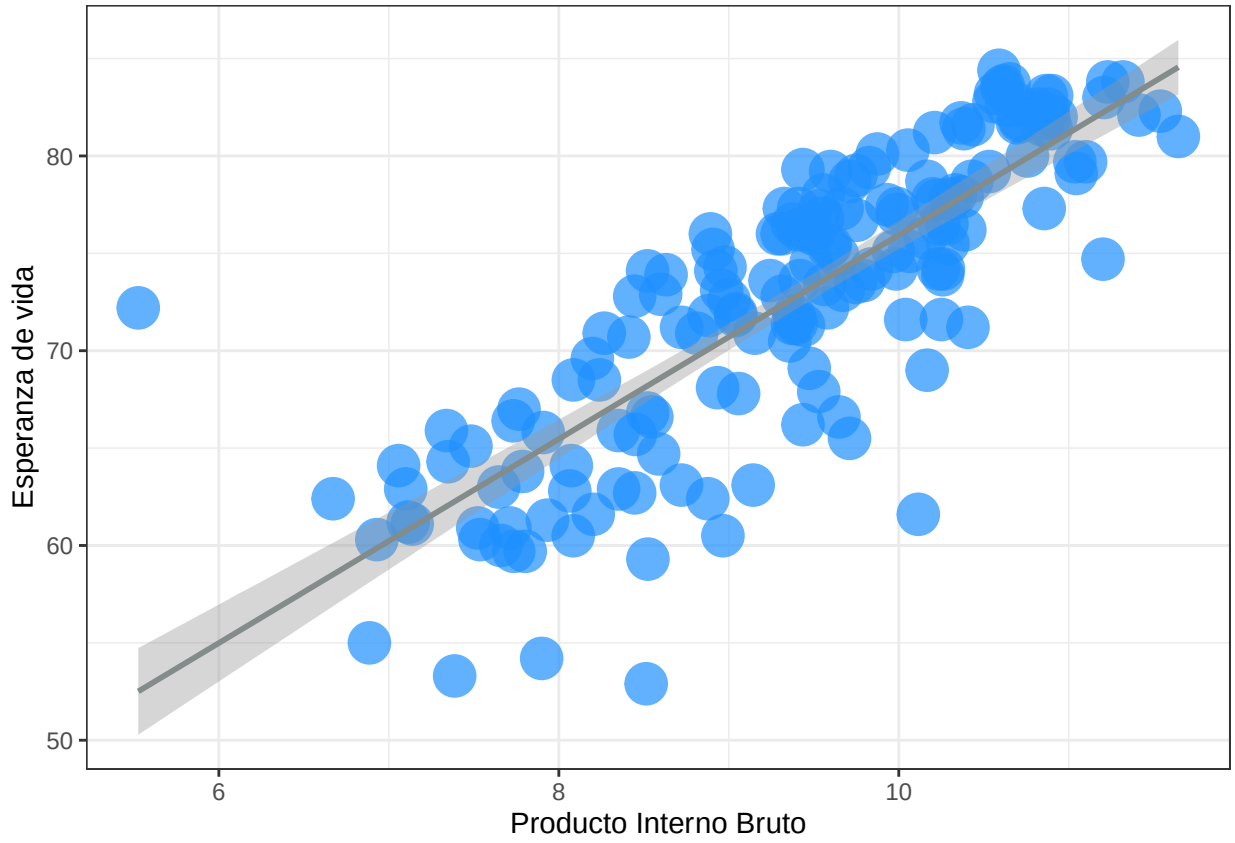
```
##                 2.5 %    97.5 %
## (Intercept) 18.313344 28.814272
## log(GDP)     4.682732  5.792827
```

```
tidy(gdp_vs_life_log)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    23.6      2.66       8.86 1.08e-15
## 2 log(GDP)        5.24     0.281     18.6  8.08e-43
```

En este punto, graficaremos el nuevo modelo.

```
ggplot(vacunas_df, aes(x = log(GDP), y = Life_expectancy)) +
  geom_point(size = 7, color = 'dodgerblue', alpha = 0.7)+
  geom_smooth(method = 'glm', color = 'azure4')+
  xlab('Producto Interno Bruto')+
  ylab('Esperanza de vida')+
  theme_bw()
```



Al observar el gráfico se detalla que los datos ya no se encuentran agrupados en un solo sitio. Por último compararemos ambos modelos.

```
tidy(gdp_vs_life)
```

```
## # A tibble: 2 x 5
##   term         estimate std.error statistic   p.value
##   <chr>           <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept) 67.6      0.580         117.  2.30e-163
## 2 GDP          0.000248 0.0000196      12.7 2.83e- 26
```

```
tidy(gdp_vs_life_log)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    23.6      2.66       8.86 1.08e-15
## 2 log(GDP)        5.24     0.281     18.6  8.08e-43
```

El modelo con transformación logarítmica explica de manera más clara la relación entre las variables.