Regresión lineal simple II

Agustín Castro

2023-10-14

Contents

Inicio

Píldoras_R. Material de formación

En esta práctica trabajemos con la **regresión lineal simple**

[mi_blog] https://agustincastro.es

La **regresión lineal simple** es un método estadístico que se utiliza para **modelar la relación entre una variable dependiente (o respuesta) y una variable independiente**. La idea principal detrás de la regresión lineal simple es comprender **cómo cambia la variable dependiente cuando lo hace la variable independiente**. El objetivo es encontrar la mejor línea recta que se ajuste a los datos observados.

Vamos a utilizar las siguientes librerías en esta práctica.

```
library(tidyverse)
library(ggplot2)
library(hrbrthemes) # para el theme_ipsum()
library(lmtest) # para el test de Breusch-Pagan
library(psych) # para pairs.panels
```

Iris

Trabajaremos el dataset **iris** que contiene datos de diferentes especies de orquídeas del género Iris.

```
data(iris)
```

El conjunto de datos cuenta con **150 observaciones**, **5 variables** y **3 especies** de Iris. Nos centraremos ahora en las variables **Petal.Length** (longitud de los pétalos) y **Sepal.Length** (longitud de los spalos).

```
head(iris, 5)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
```

```
str(iris)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 ...
```

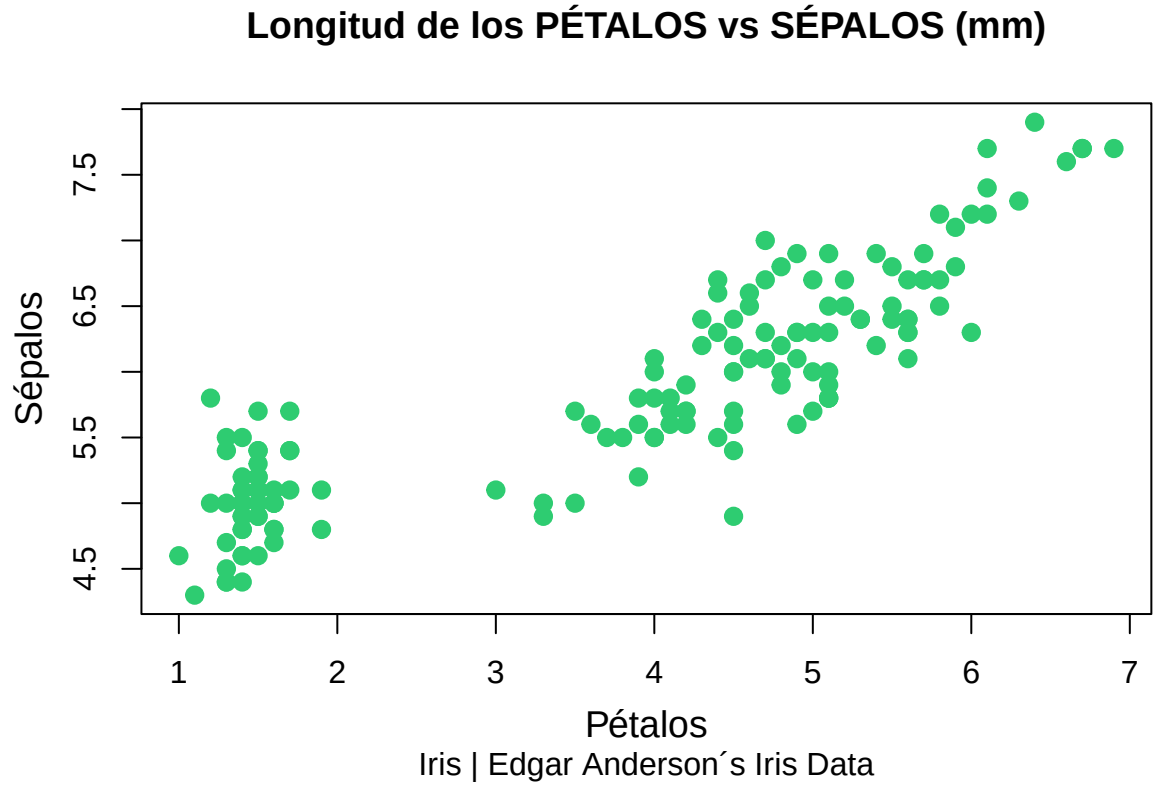
Vamos a cambiar los nombres de las variables con **rename** y quedarnos con un nuevo dataset que contenga únicamente las variables que vamos a analizar, **petalos_long** y **sepalos_long**. Además, aprovechamos para reorganizar el orden de las columnas, poniendo delante la variable **especie** y, después, el resto. Además, ordenamos el dataset resultante **iris** por **especie** y por **longitud de los pétalos**, de forma descendiente.

```
iris <- iris %>%
  rename(sepalos_long = Sepal.Length,
         sepalos_ancho = Sepal.Width,
         petalos_long = Petal.Length,
         petalos_ancho = Petal.Width,
         especie = Species) %>%
  select(especie, petalos_long, sepalos_long) %>%
  arrange(especie, desc(petalos_long))
head(iris, 5)
```

```
##   especie petalos_long sepalos_long
## 1  setosa          1.9          4.8
## 2  setosa          1.9          5.1
## 3  setosa          1.7          5.4
## 4  setosa          1.7          5.7
## 5  setosa          1.7          5.4
```

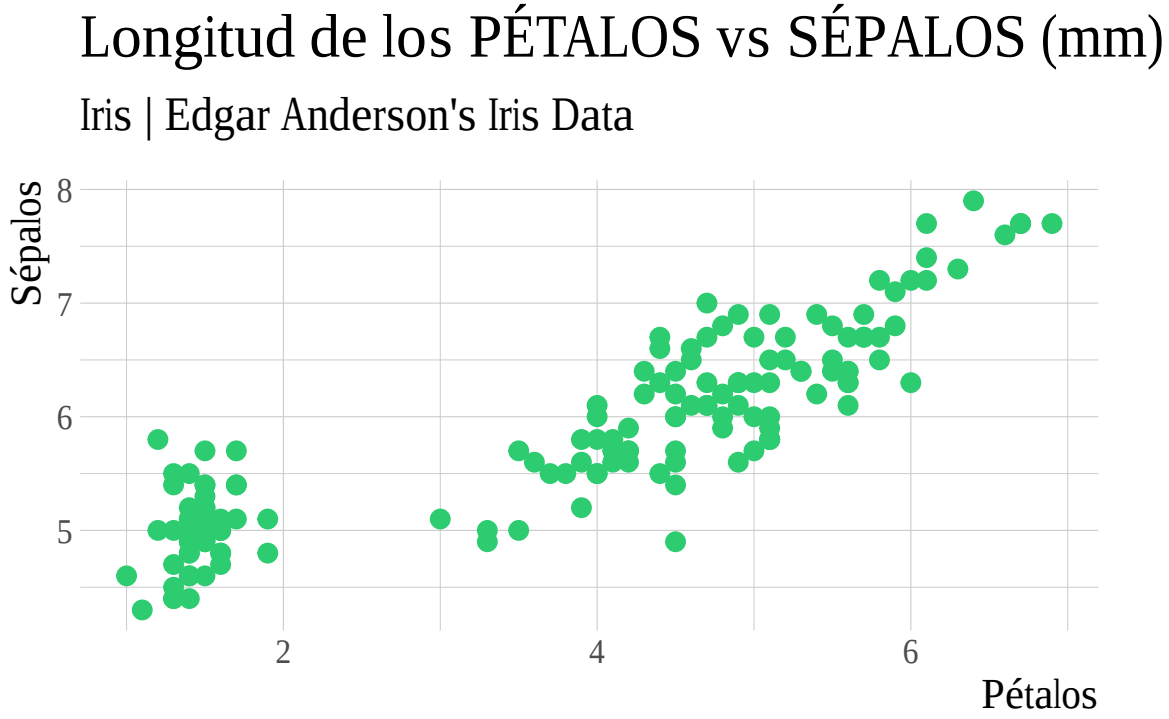
Dibujamos con **plot** la relación entre las variables **petalos_long** y **sepalos_long**. Esta función viene de base con R y permite echar un vistazo rápido al comportamiento de ambas variables. Se observa que hay una relación directa entre ellas. **A medida que aumenta la longitud de los pétalos, lo hacen de la misma forma los sépalos**.

```r
plot(iris$petalos_long, iris$sepalos_long,
     main = "Longitud de los PÉTALOS vs SÉPALOS (mm)",
     sub = "Iris | Edgar Anderson´s Iris Data",
     xlab = "Pétalos",
     ylab = "Sépalos",
     cex.lab = 1.2,
     cex.axis = 1,
     mgp = c(2.4, 1, 0),
     pch = 19,
     col = "#2ECC71",
     cex = 1.2)
```



Hacemos el mismo gráfico utilizando la librería de **ggplot**

```r
ggplot(iris, aes(x = petalos_long, y = sepalos_long)) +
  geom_point(color = "#2ECC71", size = 3) +
  labs(title = "Longitud de los PÉTALOS vs SÉPALOS (mm)",
       subtitle = "Iris | Edgar Anderson's Iris Data",
       x = "Pétalos", y = "Sépalos") +
  theme_ipsum(plot_title_size = 23,
              subtitle_size = 18,
              axis_title_size = 16,
              axis_text_size = 13)
```



Correlación lineal Pearson

Podemos estudiar como es la correlación **lineal** entre ambas variables utilizando **cor**. El coeficiente de correlación de Pearson (r) es de **0.872**, y el coeficiente de determinación (R) es **0.76**. Podríamos decir que el 76% de la variabilidad de una variable es explicada por la otra.

```
cor.test(iris$petalos_long, iris$sepalos_long)
```

```
##
##  Pearson's product-moment correlation
##
## data:  iris$petalos_long and iris$sepalos_long
## t = 21.646, df = 148, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8270363 0.9055080
## sample estimates:
##       cor
## 0.8717538
```

```
cor(iris$petalos_long, iris$sepalos_long)
```

```
## [1] 0.8717538
```

```r
(cor(iris$petalos_long, iris$sepalos_long))^2
```

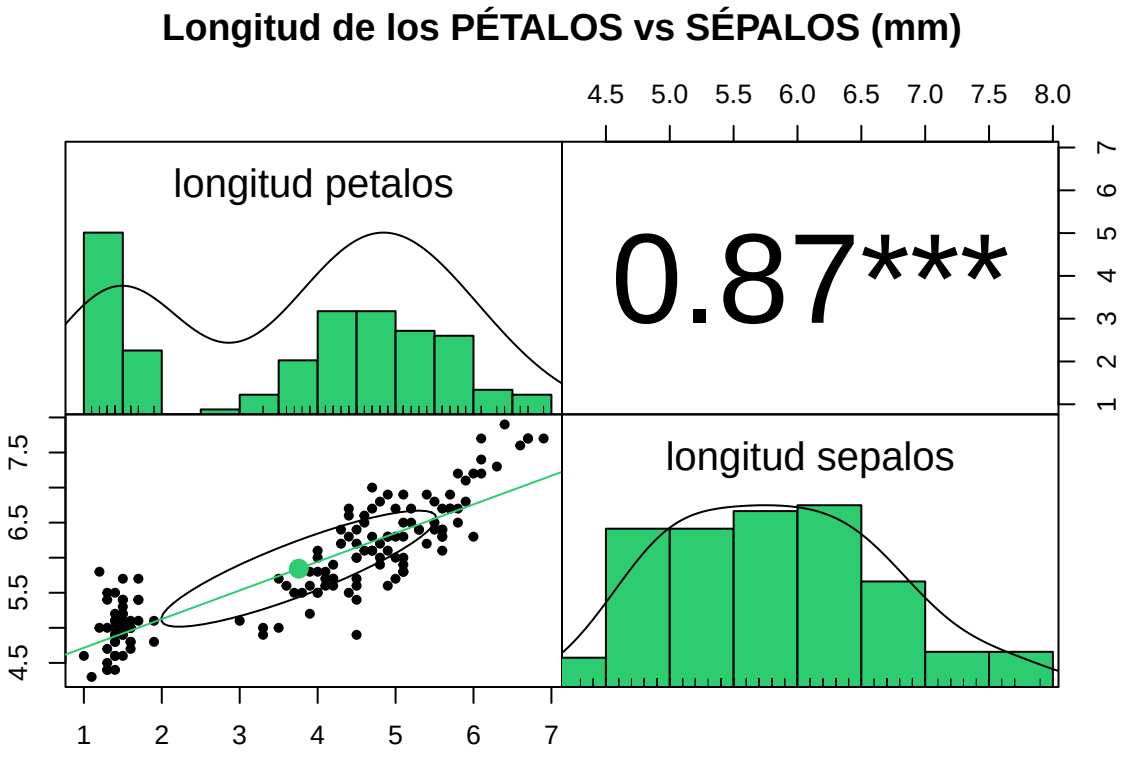
```
## [1] 0.7599546
```

Gráfico de correlación

También podemos representar gráficamente la correlación con **pairs.panels**, una función de la librería **pysch**. Hay otras muchas funciones para representar gráficamente la correlación, como por ejemplo **corrplot** (ver práctica de **correlación lineal de Pearson**).

```r
df <- data.frame(iris$petalos_long, iris$sepalos_long)
colnames(df) <- c("longitud petalos", "longitud sepalos")

pairs.panels(df, method = "pearson",
             main = "Longitud de los PÉTALOS vs SÉPALOS (mm)",
             cex.labels = 1.5,
             cex.cor = 1, stars = TRUE,
             pch = 20,
             gap = 0,
             lm = TRUE, col = "#2ECC71",
             hist.col = "#2ECC71")
```



Regresión lineal

Para crear el modelo de regresión lineal utilizamos la función **lm**, que creará el objeto **lm_iris**. En este objeto es donde se guardarán todos los resultados. La variable dependiente, o respuesta, será **sepalos_long** y la independiente, o predictora, **petalos_long**. Lo que nos interesa es conocer **como es la proporción entre una y otra longitud, y si podemos predecir la medida de una (sépalos) conociendo la otra (pétalos)**.

El objeto creado contiene información de dos valores, el **intercepto** y la **pendiente** de la recta (y = a + bX, donde y = es la variable respuesta (sepalos_long), a = intercepto y b = pendiente (petalos_long)). El modelo sería el siguiente **sepalos_long = 4.3066 + 0.4089 * petalos_long**.

```
lm_iris <- lm(sepalos_long ~ petalos_long, data = iris)
lm_iris
```

```
##
## Call:
## lm(formula = sepalos_long ~ petalos_long, data = iris)
##
## Coefficients:
##  (Intercept)  petalos_long
##       4.3066        0.4089
```

Si queremos más informción sobre el modelo creado, podemos acceder a un resumen utilizando la función **summary**.

```
summary(lm_iris)
```

```
##
## Call:
## lm(formula = sepalos_long ~ petalos_long, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24675 -0.29657 -0.01515  0.27676  1.00269
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.30660    0.07839   54.94   <2e-16 ***
## petalos_long  0.40892    0.01889   21.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4071 on 148 degrees of freedom
## Multiple R-squared:   0.76,  Adjusted R-squared:  0.7583
## F-statistic: 468.6 on 1 and 148 DF,  p-value: < 2.2e-16
```

[importante] Para poder dar por bueno el modelo de regresión lineal es necesario que se cumplan una serie de supuestos. El primero de ellos es que la relación entre las variables sea **lineal**. Por otro lado, es necesario que los **residuos sean normales**, es decir, que sigan una **distribución normal**. Por último, es necesario que la **varianza de los residuos** sea homogenea, es decir, que exista **homocedasticidad**.

Shapiro-Wilk

Para medir la normalidad de los residuos podemos utilizar el test de **Shapiro-Wilk**. Como el p-valor (0.6767) es mayor que el nivel de significancia (0.05), concluimos que no hay evidencia para rechazar la hipótesis nula de normalidad. Por lo tanto, **los residuos se distribuyen normalmente**.

```
shapiro.test(residuals(lm_iris))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lm_iris)
## W = 0.99298, p-value = 0.6767
```

Breusch-Pagan

La homocedasticidad se puede comprobar visualmente con un gráfico de dispersión o, con el test de **Breusch-Pagan**. Para realizar este test **es necesario haber instalado** previamente la librería **lmtest**. El valor p (0.09688) es mayor que 0.05, lo que sugiere que no hay suficiente evidencia para rechazar la hipótesis nula de homocedasticidad. Esto implica que **la varianza de los residuos es constante en todas las variables independientes**.

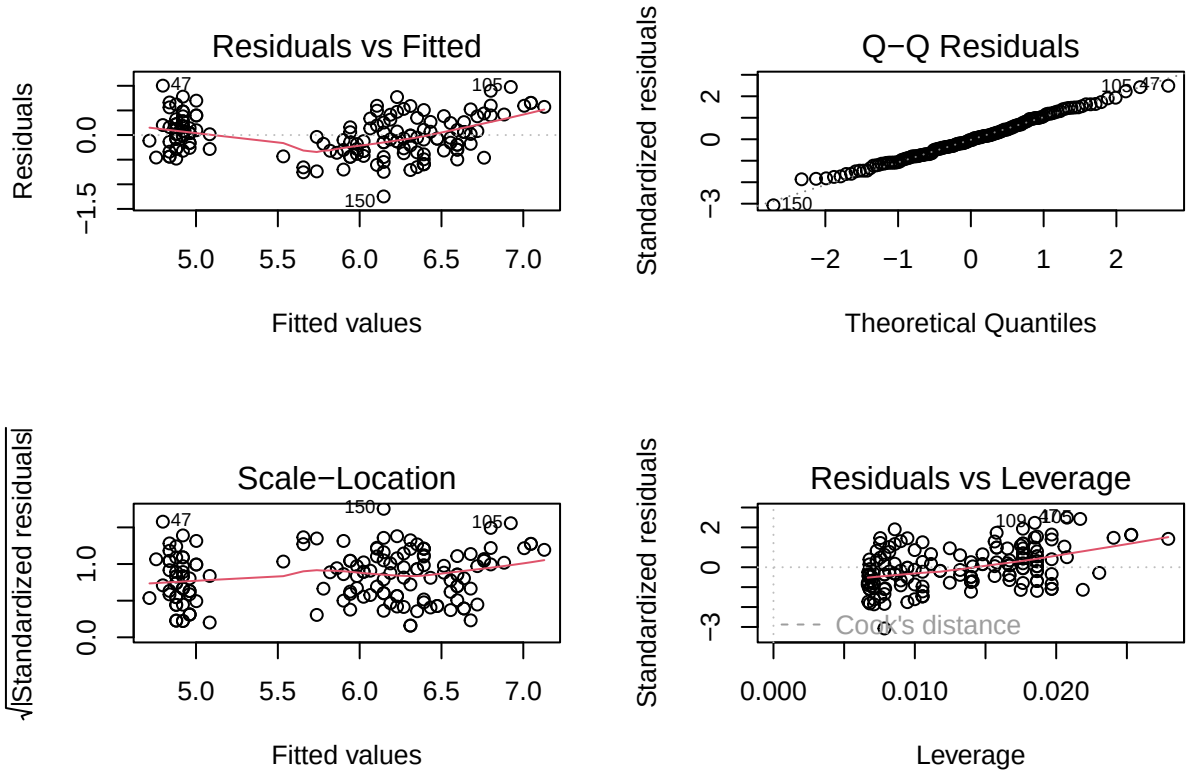
```
bptest(lm_iris)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  lm_iris
## BP = 2.7561, df = 1, p-value = 0.09688
```

Gráficas de residuos

Para comprobar la **normalidad** y la **homocedasticidad** podemos utilizar también estas gráficas de residuos. En las dos gráficas de la izquierda se observa que estos se distribuyen de forma aleatoria, sin seguir ningún patrón. La aparición de dispersiones importantes al inicio o final del gráfico son indicadores de lo contrario. Por otro lado, en el gráfico situado en la zona superior derecha (Q-Q Plot) se observa visualmente que los residuos se distribuyen de forma normal.

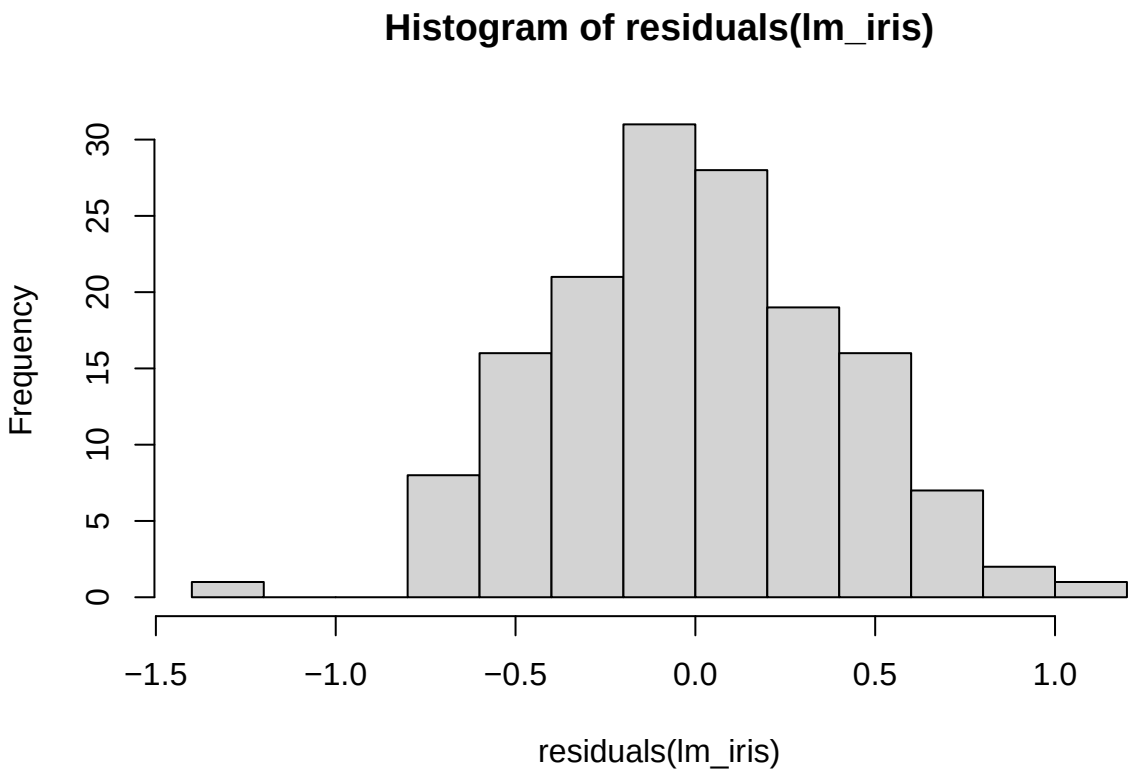
```
par(mfrow = c(2,2)) # ventana en 2 filas y 2 columnas
plot(lm_iris)
```



Histograma de residuos

Podemos observar también el histograma de los residuos para comprobar visualmente si se distribuyen de forma **normal**.

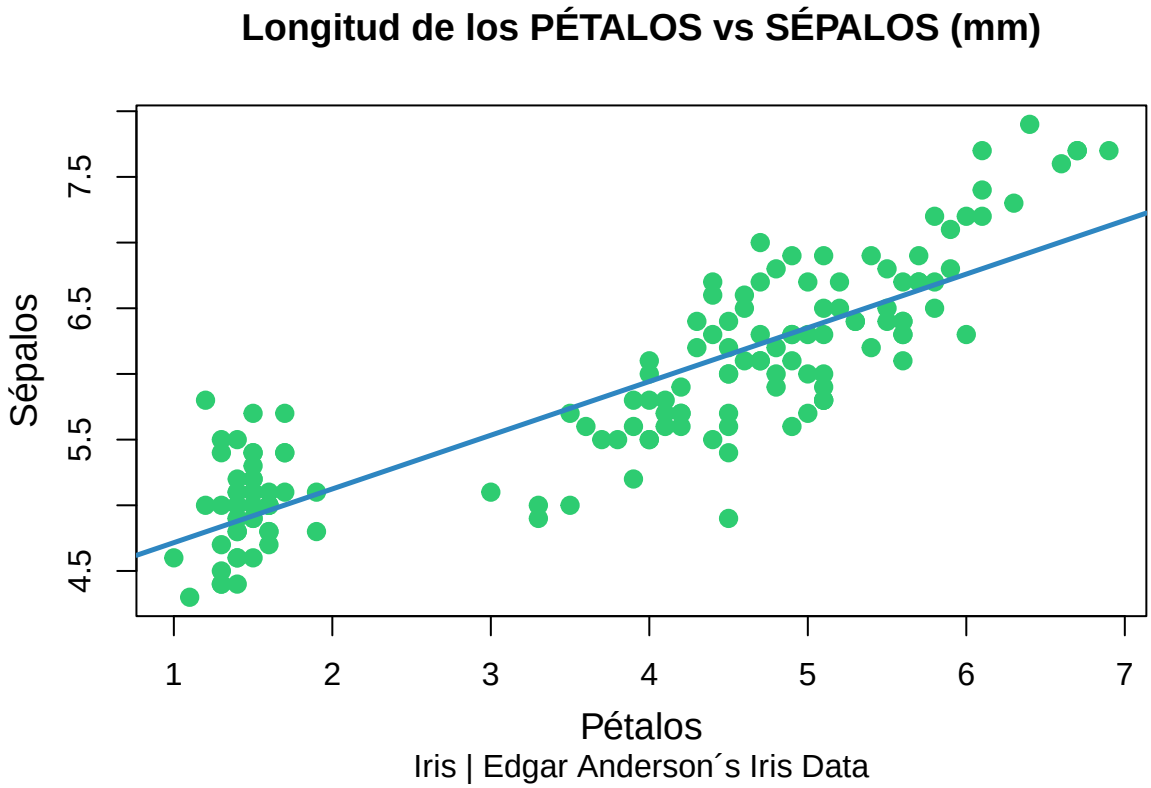
```
hist(residuals(lm_iris))
```



Gráfica de la regresión (plot)

Una vez que damos por bueno el modelo podemos, por ejemplo, representar la **recta de regresión** en el gráfico anterior. Para ello utilizamos la función **abline**, que superpondrá la línea el gráfico generado por **plot**. Ambas funciones están de forma nativa en R.

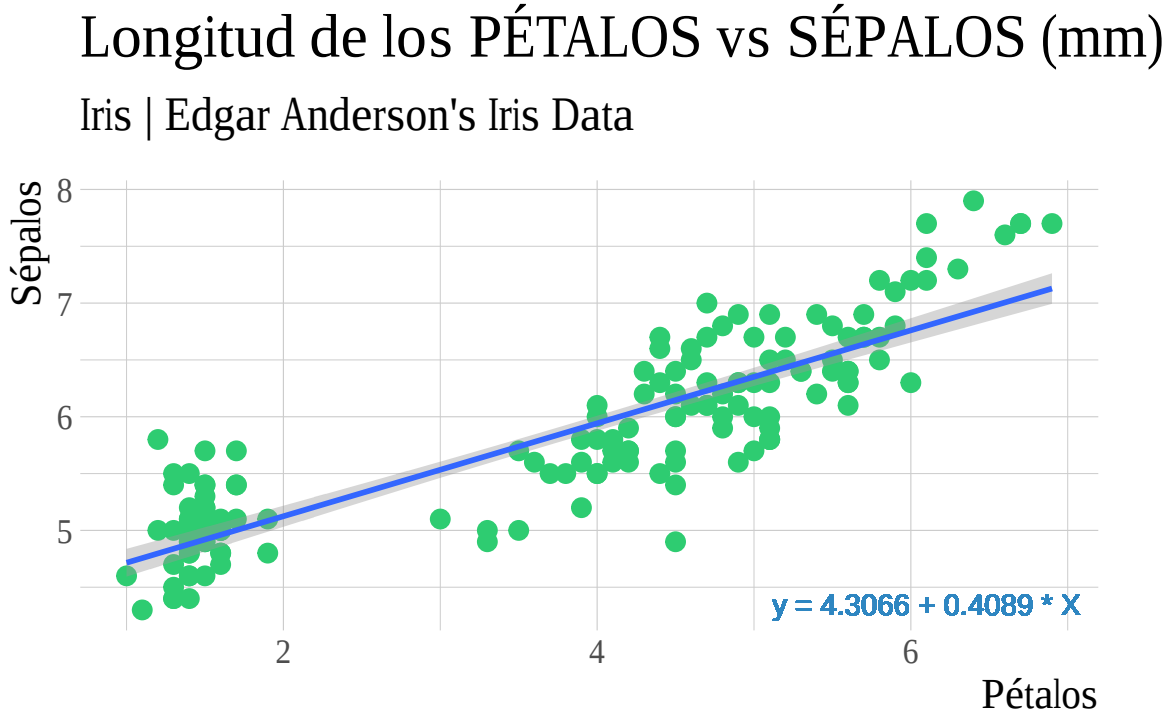
```
par(mfrow = c(1,1))
plot(iris$petalos_long, iris$sepalos_long,
     main = "Longitud de los PÉTALOS vs SÉPALOS (mm)",
     sub = "Iris | Edgar Anderson´s Iris Data",
     xlab = "Pétalos",
     ylab = "Sépalos",
     cex.lab = 1.2,
     cex.axis = 1,
     mgp = c(2.4, 1, 0),
     pch = 19,
     col = "#2ECC71",
     cex = 1.2)
abline(lm_iris, col = "#2E86C1", lwd = 2.5)
```



Gráfica de la regresión (ggplot)

Podemos utilizar la librería **ggplot2** para representar la recta de regresión. En este caso utilizamos la función **geom_smooth**, seleccionando el metodo **lm** para que, con el método de mínimos cuadrados ordinarios (MCO), ajuste una línea de regresión lineal a los datos.

```
ggplot(iris, aes(x = petalos_long, y = sepalos_long)) +
  geom_point(color = "#2ECC71", size = 3) +
  geom_smooth(method = "lm", se = TRUE) +
  labs(title = "Longitud de los PÉTALOS vs SÉPALOS (mm)",
       subtitle = "Iris | Edgar Anderson's Iris Data",
       x = "Pétalos", y = "Sépalos") +
  theme_ipsum(plot_title_size = 23,
              subtitle_size = 18,
              axis_title_size = 16,
              axis_text_size = 13) +
  geom_text(x = 6.1, y = 4.35, label = "y = 4.3066 + 0.4089 * X", color = "#2E86C1")
```



Estimación

por último, podemos hacer el cálculo de una predicción de la longitud de los sépalos para unos pétalos de 3.5 mm. Con la recta generada en el modelo **sepalos_long = 4.3066 + 0.4089 * petalos_long** el resultado sería de **5.73775**.

```
petalos_long = 3.5
sepalos_long = 4.3066 + 0.4089 * petalos_long
print(sepalos_long)
```

```
## [1] 5.73775
```

eof, 14/10/2023