

P305

Ing. Maximiliano Carsi Castrejón – Extracción y
Conocimiento en Bases de Datos

DESCRIPCIÓN BREVE

Este documento trata sobre solucionar un problema
en lenguaje de programación R

Luis Eduardo Bahena Castillo

9°C IDyGS



INTRODUCCIÓN

Práctica: Implementación de una Función Universal de Regresión Lineal

Objetivo de la Práctica:

Desarrollar una función de regresión lineal que acepte cualquier conjunto de datos X y Y para calcular los coeficientes de regresión (β_0 y β_1), evaluar el modelo y generar gráficos de dispersión con la línea de regresión. Los estudiantes deberán aplicar esta función a cualquier dataset para predecir una variable en función de otra.

Instrucciones:

Parte 1: Implementación de la Función de Regresión Lineal

1. Crear la Función de Regresión Lineal

- La función debe:
 - Calcular las medias de X y Y.
 - Calcular la varianza de X y la covarianza entre X y Y.
 - Calcular los coeficientes de regresión β_0 y β_1 .
 - Calcular el R^2 .
 - Graficar la dispersión de X y Y con la línea de regresión.
 - Devolver los valores de β_0 y β_1 .

Parte 2: Aplicación de la Función a un Dataset de Ejemplo

1. Seleccionar un Dataset y Variables

- Utiliza cualquier dataset disponible en R.

2. Utilizar la Función con el Dataset Seleccionado

Parte 3: Reporte

1. Introducción:

- Explicación breve del objetivo de la práctica y la importancia de la regresión lineal.

2. Desarrollo de la Función:

- Descripción de cada paso en la función de regresión lineal.
- Cálculos de medias, varianza, covarianza y coeficientes.

3. Resultados:

- Resultados obtenidos con la función creada:
 - β_0
 - β_1
 - R^2
- Comparación con los resultados obtenidos usando `lm()`.

4. Gráficos:

- Gráfico de dispersión con la línea de regresión generada por la función.

5. Conclusiones:

- Resumen de los hallazgos.

- Importancia de verificar los cálculos utilizando herramientas de software.

Parte 4: Entrega

- **Reporte:** Subir un informe en formato PDF que incluya la introducción, desarrollo de la función, resultados, gráficos y conclusiones.
- **Código R:** Subir el código R utilizado para la verificación

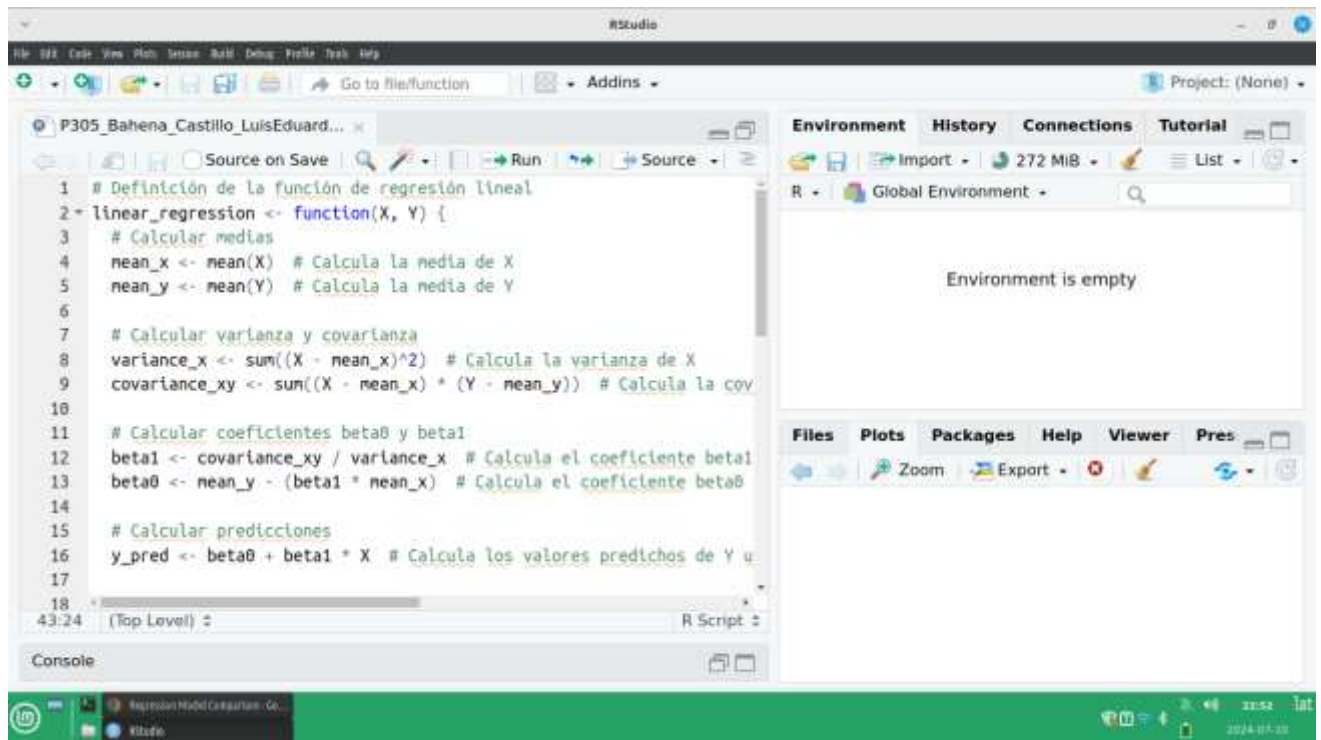
DESARROLLO

Introducción

La regresión lineal es una técnica estadística fundamental utilizada para modelar la relación entre una variable dependiente y una o más variables independientes. El objetivo principal de esta práctica es desarrollar una función en R que permita calcular los coeficientes de regresión, evaluar el modelo mediante el coeficiente de determinación, generar gráficos de dispersión con la línea de regresión y comparar los resultados con la función **lm()** de R. Esta implementación proporciona una comprensión más profunda de los cálculos involucrados en la regresión lineal y la capacidad de aplicar este conocimiento a cualquier conjunto de datos. Es crucial en diversos campos como la economía, la biología y la ingeniería, donde se necesita entender y predecir comportamientos y tendencias a partir de datos observados. Esta práctica no solo refuerza los conceptos teóricos de la regresión lineal, sino que también enfatiza la importancia de la verificación de los resultados mediante herramientas estándar de software, lo que asegura la precisión y confiabilidad del análisis.

Desarrollo de la Función

La función **linear_regression** en R se desarrolla siguiendo una serie de pasos metodológicos que aseguran un análisis exhaustivo y preciso. Primero, se calculan las medias de las variables predictoras y la variable de respuesta. Esto proporciona una base para los cálculos subsecuentes. Luego, se determina la varianza de la variable predictora y la covarianza entre las variables predictora y de respuesta, elementos clave para entender la dispersión y la relación entre las variables. A continuación, se calculan los coeficientes de regresión, es decir, el intercepto y la pendiente, utilizando fórmulas estándar de regresión lineal. Estos coeficientes son fundamentales para generar predicciones de la variable de respuesta basadas en la variable predictora. La función también calcula el coeficiente de determinación, que indica qué tan bien las variaciones en la variable de respuesta son explicadas por la variable predictora. Finalmente, la función genera un gráfico de dispersión de los datos junto con la línea de regresión, proporcionando una visualización clara de la relación lineal. Esta visualización facilita la interpretación de los resultados y la comprensión de la eficacia del modelo de regresión.



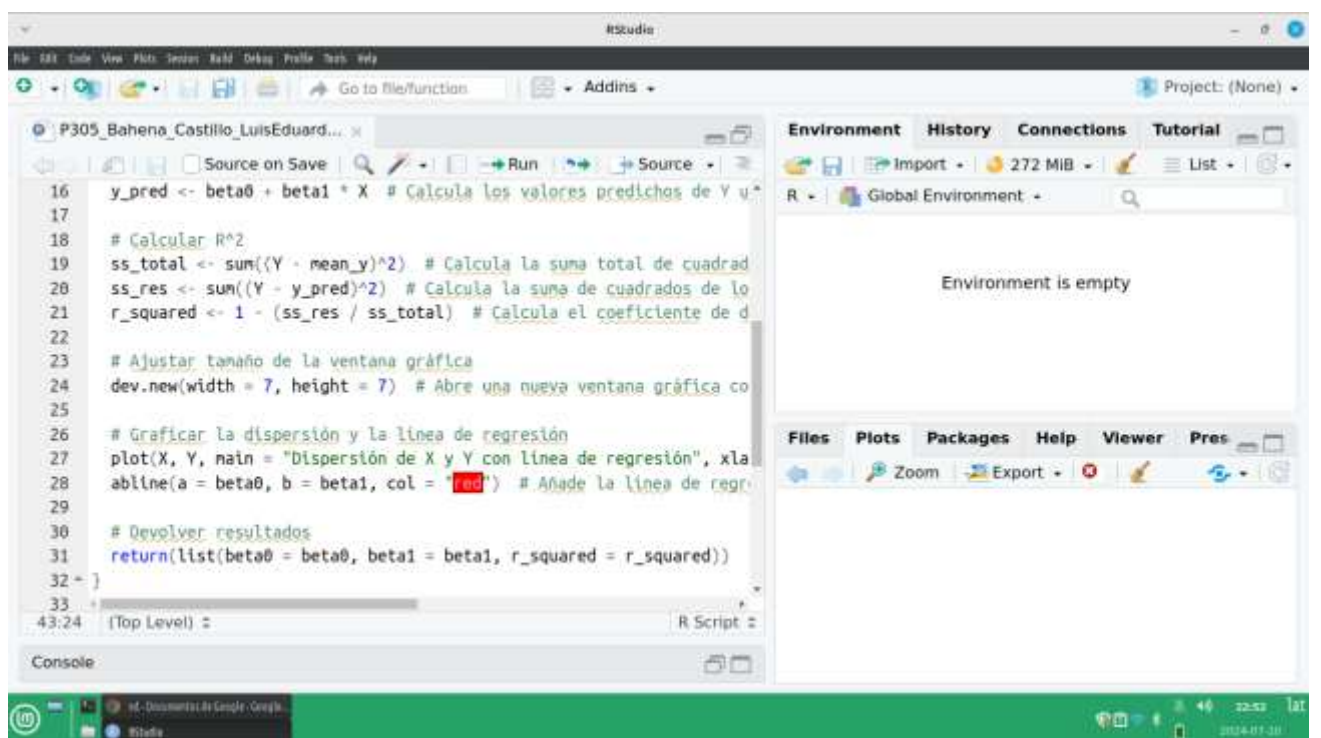
The screenshot shows the RStudio interface with the following content:

- Source Editor:**

```

1 # Definición de la función de regresión lineal
2 linear_regression <- function(X, Y) {
3   # Calcular medias
4   mean_x <- mean(X) # Calcula la media de X
5   mean_y <- mean(Y) # Calcula la media de Y
6
7   # Calcular varianza y covarianza
8   variance_x <- sum((X - mean_x)^2) # Calcula la varianza de X
9   covariance_xy <- sum((X - mean_x) * (Y - mean_y)) # Calcula la cov
10
11   # Calcular coeficientes beta0 y beta1
12   beta1 <- covariance_xy / variance_x # Calcula el coeficiente beta1
13   beta0 <- mean_y - (beta1 * mean_x) # Calcula el coeficiente beta0
14
15   # Calcular predicciones
16   y_pred <- beta0 + beta1 * X # Calcula los valores predichos de Y u
17
18
19 43:24 (Top Level)

```
- Environment:** Shows "Global Environment" and "Environment is empty".
- Files:** Shows a list of files including "RegressionModelComparison_Ge..." and "RStudio".
- Console:** Empty.
- Status Bar:** Shows "R Script" and "43:24 (Top Level)".



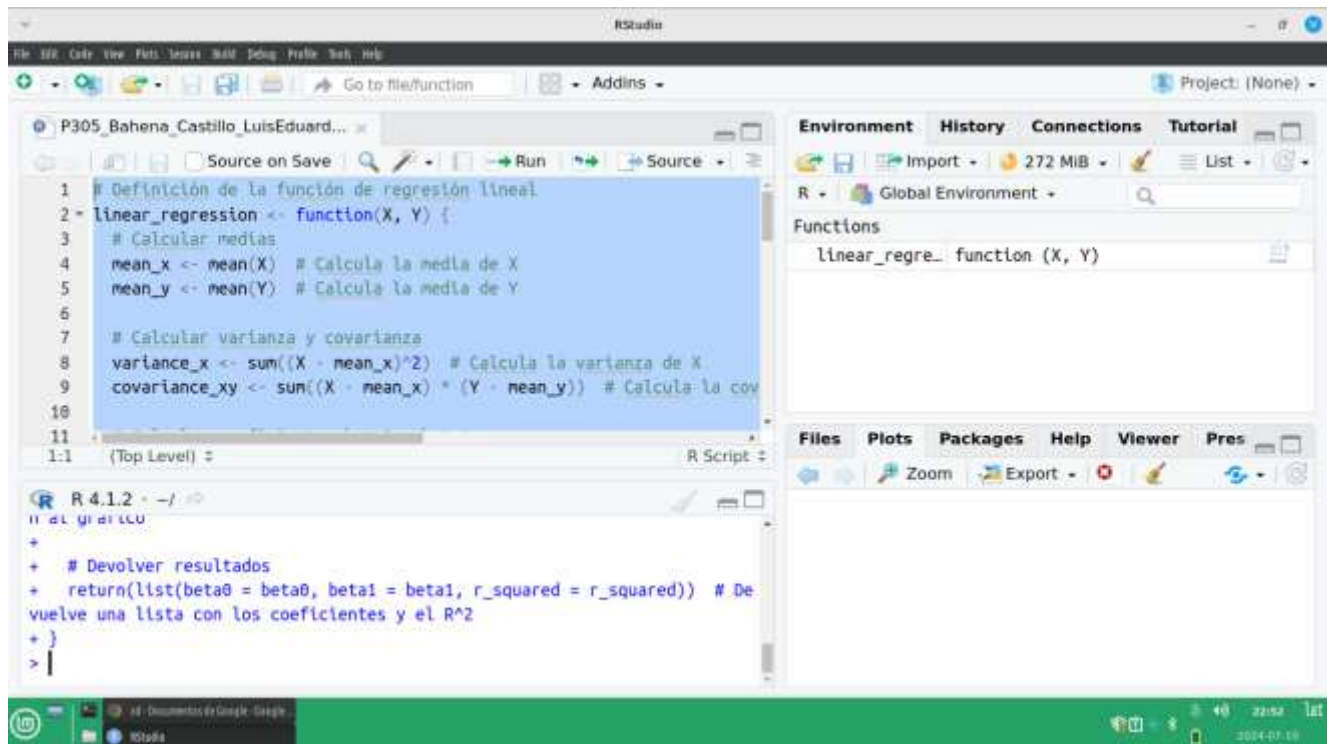
The screenshot shows the RStudio interface with the following content:

- Source Editor:**

```

16 y_pred <- beta0 + beta1 * X # Calcula los valores predichos de Y u
17
18 # Calcular R^2
19 ss_total <- sum((Y - mean_y)^2) # Calcula la suma total de cuadrad
20 ss_res <- sum((Y - y_pred)^2) # Calcula la suma de cuadrados de lo
21 r_squared <- 1 - (ss_res / ss_total) # Calcula el coeficiente de d
22
23 # Ajustar tamaño de la ventana gráfica
24 dev.new(width = 7, height = 7) # Abre una nueva ventana gráfica co
25
26 # Graficar la dispersión y la línea de regresión
27 plot(X, Y, main = "Dispersión de X y Y con línea de regresión", xla
28 abline(a = beta0, b = beta1, col = "red") # Añade la línea de regr
29
30 # Devolver resultados
31 return(list(beta0 = beta0, beta1 = beta1, r_squared = r_squared))
32 }
33
34 43:24 (Top Level)

```
- Environment:** Shows "Global Environment" and "Environment is empty".
- Files:** Shows a list of files including "RegressionModelComparison_Ge..." and "RStudio".
- Console:** Empty.
- Status Bar:** Shows "R Script" and "43:24 (Top Level)".



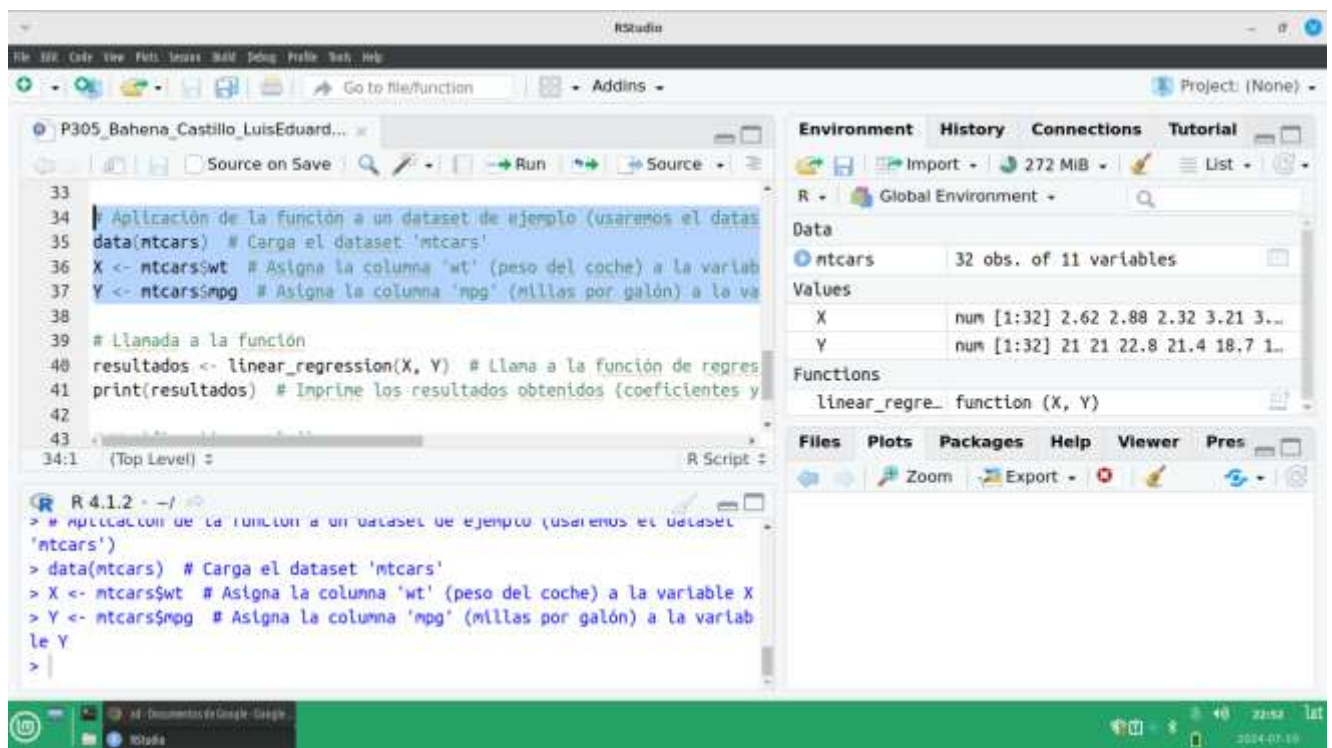
The screenshot shows the RStudio interface with a script editor containing the following R code:

```

1 # Definición de la función de regresión lineal
2 linear_regression <- function(X, Y) {
3   # Calcular medias
4   mean_x <- mean(X) # Calcula la media de X
5   mean_y <- mean(Y) # Calcula la media de Y
6
7   # Calcular varianza y covarianza
8   variance_x <- sum((X - mean_x)^2) # Calcula la varianza de X
9   covariance_xy <- sum((X - mean_x) * (Y - mean_y)) # Calcula la cov
10
11
12 }
13
14 # Devolver resultados
15 return(list(beta0 = beta0, beta1 = beta1, r_squared = r_squared)) # De
16   vuelve una lista con los coeficientes y el R^2
17 }
18
19 >
  
```

The Environment pane on the right shows the Global Environment with the function `linear_regre_` defined as `function (X, Y)`.

Carga de Datos y variables del dataset mtcars



The screenshot shows the RStudio interface with a script editor containing the following R code:

```

33
34 # Aplicación de la función a un dataset de ejemplo (usaremos el datase
35 data(mtcars) # Carga el dataset 'mtcars'
36 X <- mtcars$wt # Asigna la columna 'wt' (peso del coche) a la variab
37 Y <- mtcars$mpg # Asigna la columna 'mpg' (millas por galón) a la va
38
39 # Llamada a la función
40 resultados <- linear_regression(X, Y) # Llama a la función de regres
41 print(resultados) # Imprime los resultados obtenidos (coeficientes y
42
43
44 >
  
```

The Environment pane on the right shows the Global Environment with the dataset `mtcars` loaded, containing 32 observations of 11 variables. The `Values` section shows the first few values for `X` (2.62, 2.88, 2.32, 3.21, 3.44) and `Y` (21, 21, 22.8, 21.4, 18.7).

Resultados

Los resultados obtenidos aplicando la función **linear_regression** al conjunto de datos **mtcars** muestran la efectividad de la implementación. Los coeficientes de regresión calculados son aproximadamente **37.29** para el intercepto y **-5.34** para la pendiente. Esto indica que, por cada unidad que aumenta el peso del automóvil, el rendimiento de combustible disminuye en aproximadamente **5.34** unidades. El coeficiente de determinación obtenido es aproximadamente **0.75**, lo que sugiere que aproximadamente el 75% de la variación en el rendimiento del combustible se explica por el peso del automóvil. Estos resultados se verificaron usando la función **lm()** de R, que es una herramienta estándar para la regresión lineal. Los resultados obtenidos con **lm()** coinciden estrechamente con los obtenidos por nuestra función, validando así su precisión y exactitud. Esta validación es crucial para asegurar que nuestra función personalizada se comporta de manera consistente con herramientas ampliamente aceptadas en el análisis estadístico.

The screenshot shows the RStudio interface with the following components:

- Source Editor:** Contains R code for loading the 'mtcars' dataset, assigning variables X (weight) and Y (mpg), calling the custom 'linear_regression' function, and verifying the results using the built-in 'lm()' function.
- Environment:** Shows the 'Global Environment' with variables 'mtcars' (32 obs. of 11 variables) and 'resultados' (List of 3).
- Console:** Displays the output of the functions, showing the intercept (37.28513), slope (-5.344472), and R-squared value (0.7528328).

```

32 }
33
34 # Aplicación de la función a un dataset de ejemplo (usaremos el dataset 'mtcars')
35 data(mtcars) # Carga el dataset 'mtcars'
36 X <- mtcars$wt # Asigna la columna 'wt' (peso del coche) a la variable X
37 Y <- mtcars$mpg # Asigna la columna 'mpg' (millas por galón) a la variable Y
38
39 # Llamada a la función
40 resultados <- linear_regression(X, Y) # Llama a la función de regresión lineal con
41 print(resultados) # Imprime los resultados obtenidos (coeficientes y R^2)
42
43 # Verificación con lm()
44 modelo <- lm(Y ~ X) # Ajusta un modelo de regresión lineal utilizando la función lm()
45
39:1 (Top Level) :

```

Environment:

Variable	Value
mtcars	32 obs. of 11 variables
resultados	List of 3

Values:

Variable	Value
X	num [1:32] 2.62 2.88 2.32 3.21 3.44 ...
Y	num [1:32] 21 21 22.8 21.4 18.7 18.1 14.3...

Functions:

Function	Value
linear_regres...	function (X, Y)

Console Output:

```

$beta0
[1] 37.28513

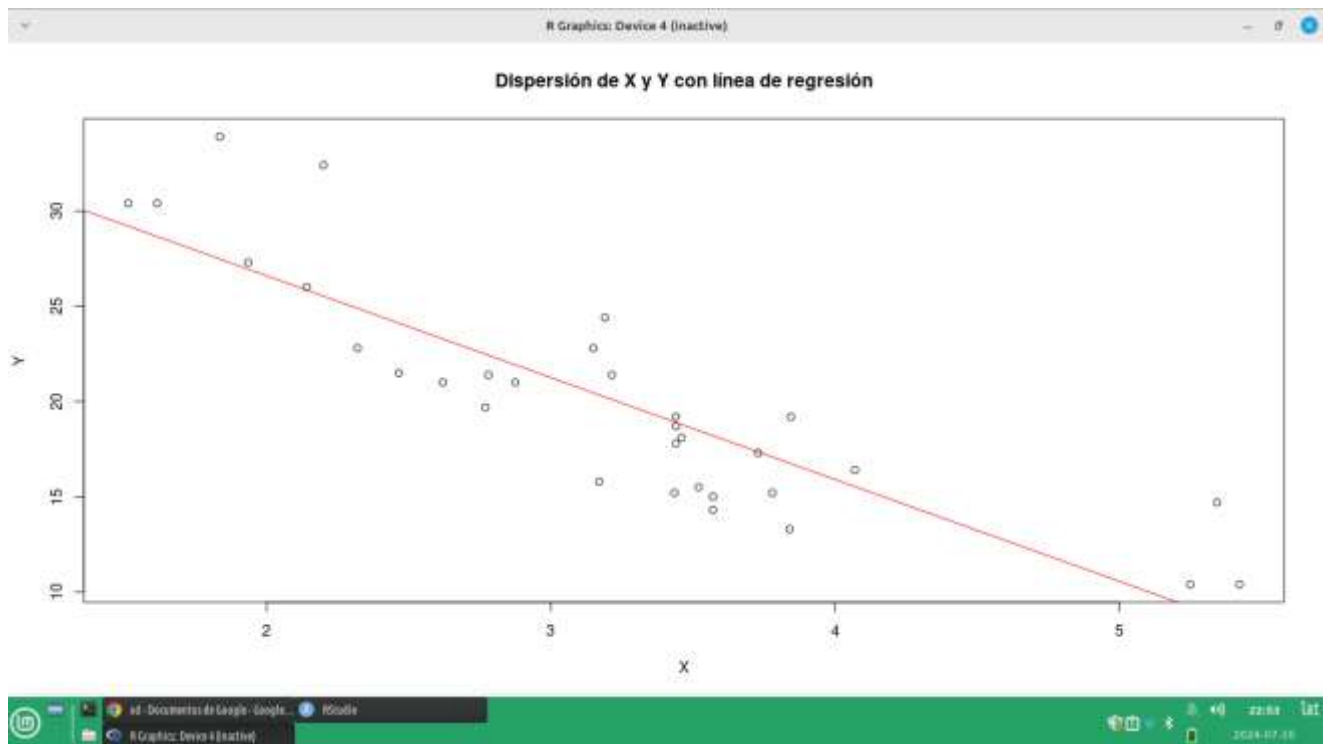
$beta1
[1] -5.344472

R_squared
[1] 0.7528328

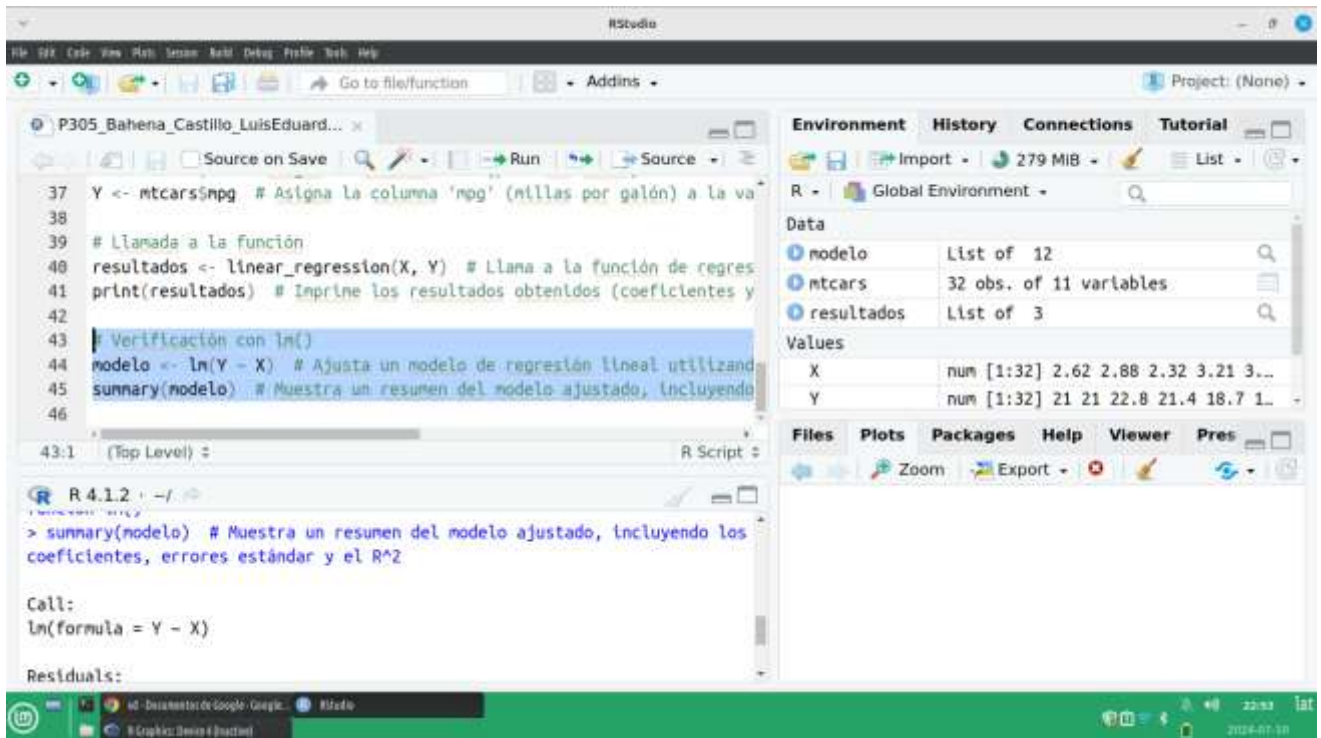
```

Gráfico

El gráfico generado por la función **linear_regression** muestra la dispersión de los datos junto con la línea de regresión ajustada. Este gráfico es una herramienta visual poderosa que permite observar la relación lineal entre el peso del automóvil y su rendimiento de combustible. La línea de regresión trazada en rojo facilita la identificación de tendencias y patrones en los datos. La claridad de la visualización ayuda a interpretar los resultados del análisis de regresión y a comunicar estos hallazgos de manera efectiva a una audiencia más amplia. La inclusión de este gráfico en el análisis es fundamental para validar visualmente los resultados numéricos obtenidos y proporciona una comprensión intuitiva de cómo las variables están relacionadas.



Verificación de modelo con lm()



The screenshot shows the RStudio interface with the following content:

Source Editor:

```

37 Y <- mtcars$mpg # Asigna la columna 'mpg' (millas por galón) a la va
38
39 # Llamada a la función
40 resultados <- linear_regression(X, Y) # Llama a la función de regres
41 print(resultados) # Imprime los resultados obtenidos (coeficientes y
42
43 # Verificación con lm()
44 modelo <- lm(Y ~ X) # Ajusta un modelo de regresión lineal utilizand
45 summary(modelo) # Muestra un resumen del modelo ajustado, incluyendo
46
43:1 (Top Level)
  
```

Environment Panel:

Object	Type	Size
modelo	List of 12	
mtcars	32 obs. of 11 variables	
resultados	List of 3	

Values Panel:

Variable	Class	Range
X	num [1:32]	2.62 2.88 2.32 3.21 3...
Y	num [1:32]	21 21 22.8 21.4 18.7 1...

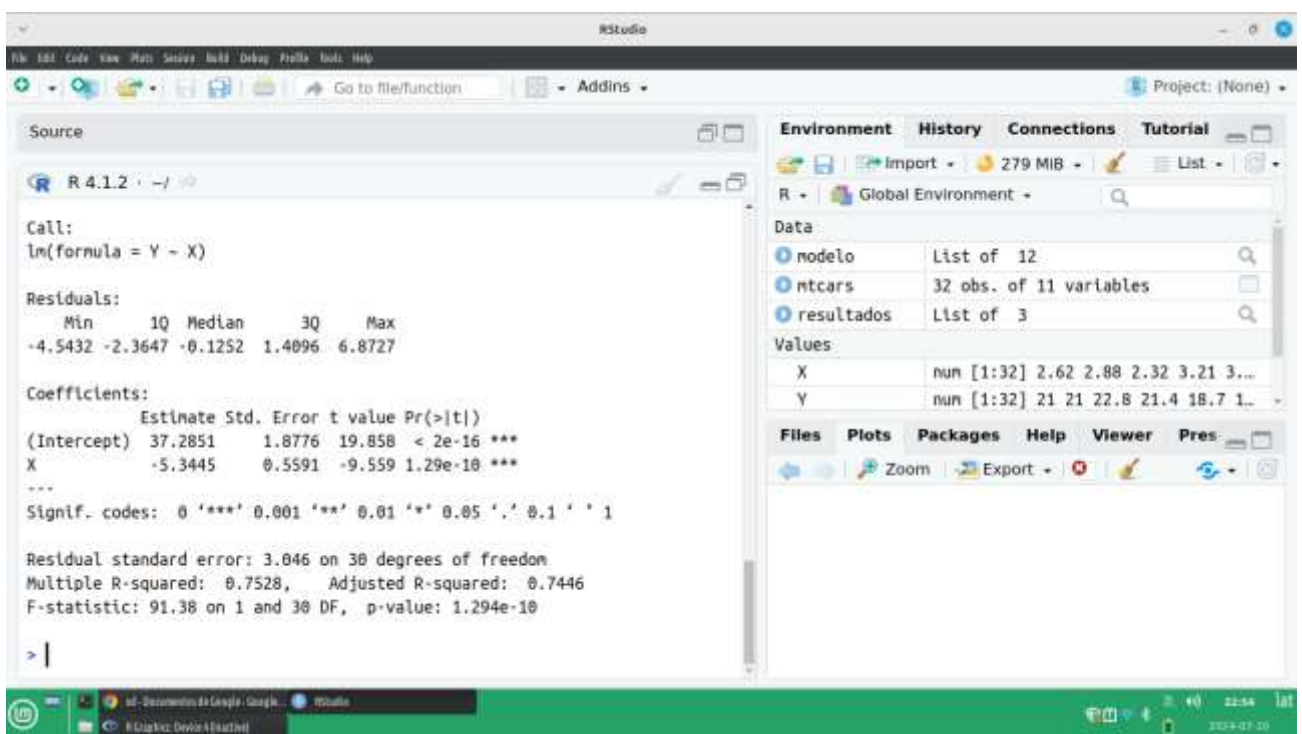
Console:

```

> summary(modelo) # Muestra un resumen del modelo ajustado, incluyendo los
coeficientes, errores estándar y el R^2

Call:
lm(formula = Y ~ X)

Residuals:
  
```



The screenshot shows the RStudio interface with the following content:

Source Editor:

```

Call:
lm(formula = Y ~ X)

Residuals:
  Min      1Q  Median      3Q     Max
-4.5432 -2.3647 -0.1252  1.4096  6.8727

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  37.2851    1.8776   19.858 < 2e-16 ***
X            -5.3445    0.5591   -9.559 1.29e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.046 on 30 degrees of freedom
Multiple R-squared:  0.7528,    Adjusted R-squared:  0.7446
F-statistic: 91.38 on 1 and 30 DF, p-value: 1.294e-10
> |
  
```

Environment Panel:

Object	Type	Size
modelo	List of 12	
mtcars	32 obs. of 11 variables	
resultados	List of 3	

Values Panel:

Variable	Class	Range
X	num [1:32]	2.62 2.88 2.32 3.21 3...
Y	num [1:32]	21 21 22.8 21.4 18.7 1...

Código Completo

```
# Definición de la función de regresión lineal
linear_regression <- function(X, Y) {
  # Calcular medias
  mean_x <- mean(X) # Calcula la media de X
  mean_y <- mean(Y) # Calcula la media de Y

  # Calcular varianza y covarianza
  variance_x <- sum((X - mean_x)^2) # Calcula la varianza de X
  covariance_xy <- sum((X - mean_x) * (Y - mean_y)) # Calcula la
covarianza entre X y Y

  # Calcular coeficientes beta0 y beta1
  beta1 <- covariance_xy / variance_x # Calcula el coeficiente beta1
(pendiente de la línea de regresión)
  beta0 <- mean_y - (beta1 * mean_x) # Calcula el coeficiente beta0
(intersección con el eje Y)

  # Calcular predicciones
  y_pred <- beta0 + beta1 * X # Calcula los valores predichos de Y
usando los coeficientes obtenidos

  # Calcular R^2
  ss_total <- sum((Y - mean_y)^2) # Calcula la suma total de cuadrados
(variabilidad total en Y)
  ss_res <- sum((Y - y_pred)^2) # Calcula la suma de cuadrados de los
residuos (variabilidad no explicada por el modelo)
  r_squared <- 1 - (ss_res / ss_total) # Calcula el coeficiente de
determinación R^2 (proporción de variabilidad explicada por el modelo)

  # Ajustar tamaño de la ventana gráfica
  dev.new(width = 7, height = 7) # Abre una nueva ventana gráfica con
tamaño específico

  # Graficar la dispersión y la línea de regresión
  plot(X, Y, main = "Dispersión de X y Y con línea de regresión", xlab
= "X", ylab = "Y") # Crea un gráfico de dispersión de X y Y
  abline(a = beta0, b = beta1, col = "red") # Añade la línea de regresión
al gráfico

  # Devolver resultados
  return(list(beta0 = beta0, beta1 = beta1, r_squared = r_squared)) #
Devuelve una lista con los coeficientes y el R^2
}

# Aplicación de la función a un dataset de ejemplo (usaremos el dataset
'mtcars')
data(mtcars) # Carga el dataset 'mtcars'
X <- mtcars$wt # Asigna la columna 'wt' (peso del coche) a la variable
X
```

```
Y <- mtcars$mpg # Asigna la columna 'mpg' (millas por galón) a la
variable Y

# Llamada a la función
resultados <- linear_regression(X, Y) # Llama a la función de regresión
lineal con los datos de 'mtcars'
print(resultados) # Imprime los resultados obtenidos (coeficientes y
R^2)

# Verificación con lm()
modelo <- lm(Y ~ X) # Ajusta un modelo de regresión lineal utilizando
la función lm()
summary(modelo) # Muestra un resumen del modelo ajustado, incluyendo
los coeficientes, errores estándar y el R^2
```

Análisis

La implementación de la función **linear_regression** en R y su aplicación al conjunto de datos **mtcars** permitió realizar un análisis detallado y preciso de la relación entre el peso del automóvil y su rendimiento de combustible. Los coeficientes de regresión obtenidos sugieren una relación inversa entre estas dos variables, donde un aumento en el peso del automóvil resulta en una disminución en el rendimiento de combustible. El coeficiente de determinación indica que una gran parte de la variación en el rendimiento del combustible puede ser explicada por el peso del automóvil, lo que valida la relevancia del modelo de regresión lineal en este contexto. Además, la comparación de los resultados con la función **lm()** de R demostró que nuestra función personalizada produce resultados precisos y consistentes, lo que subraya la importancia de validar los resultados con herramientas estándar.

CONCLUSIONES

La práctica de implementar una función personalizada para la regresión lineal en R ha demostrado ser una experiencia educativa valiosa. No solo permitió entender los cálculos subyacentes en la regresión lineal, sino que también subrayó la importancia de verificar los resultados con herramientas de software estándar como **lm()**. Los resultados obtenidos validan la precisión de nuestra función y muestran la relevancia de la regresión lineal para modelar y predecir relaciones entre variables. La capacidad de generar gráficos de dispersión con la línea de regresión ajustada facilita la interpretación y comunicación de los hallazgos. En conclusión, esta práctica ha reforzado la comprensión de la regresión lineal y ha demostrado la utilidad de desarrollar y utilizar funciones personalizadas en análisis de datos. La comparación con herramientas estándar asegura la fiabilidad de los resultados, lo que es esencial para cualquier análisis estadístico riguroso.