

# Proyecto Final - Efecto del Clima y Factores Ambientales en la Productividad Agrícola

Est. Apl. 2

## Introducción al Problema

La productividad agrícola es fundamental para la seguridad alimentaria y el desarrollo económico de muchas regiones del mundo. El cambio climático y las fluctuaciones ambientales han generado preocupaciones sobre cómo estos factores influyen en el rendimiento de los cultivos y, en consecuencia, en la sostenibilidad de la agricultura a nivel global.

Este análisis busca comprender el efecto que las condiciones climáticas y los factores ambientales tienen sobre la productividad agrícola. La relevancia de este problema radica en su impacto directo sobre la capacidad de los países para satisfacer las demandas alimenticias de su población, especialmente en un contexto de cambio climático y creciente presión sobre los recursos naturales.

A partir de este contexto, se plantea responder a las siguientes preguntas de investigación mediante un análisis estadístico:

1. ¿Qué factores ambientales tienen un efecto significativo en la productividad agrícola? 2. ¿Qué modelo estadístico ofrece la mejor predicción de la productividad?

El objetivo de este análisis es aportar información que pueda ser útil para diseñar políticas y estrategias agrícolas más resilientes y sostenibles en el futuro.

## Análisis de Datos

El conjunto de datos utilizado en este análisis proviene de Kaggle y está disponible en el siguiente enlace: Global Agriculture Climate Impact Dataset. Este dataset ofrece información detallada sobre la interacción entre factores climáticos, ambientales y agrícolas, y su impacto en la productividad de los cultivos. A continuación, se describen las variables incluidas en el dataset:

- **Crop\_Yield\_MT\_per\_HA**
  - **Tipo:** Numérica continua
    - **Descripción:** Productividad agrícola medida en toneladas métricas por hectárea. Esta es la variable objetivo del análisis.
- **Year**
  - **Tipo:** Numérica entera
    - **Descripción:** Año en que se registraron los datos.
- **Country**
  - **Tipo:** Categórica
    - **Descripción:** País donde se recopiló la información.
- **Region**
  - **Tipo:** Categórica

- **Descripción:** Región específica dentro del país, utilizada para un análisis más granular.
- **Crop\_Type**
  - **Tipo:** Categórica
  - **Descripción:** Tipo de cultivo analizado (e.g., trigo, maíz, arroz).
- **Average\_Temperature\_C**
  - **Tipo:** Numérica continua
  - **Descripción:** Temperatura promedio registrada durante la temporada de cultivo.
- **Total\_Precipitation\_mm**
  - **Tipo:** Numérica continua
  - **Descripción:** Precipitación total (en milímetros) durante la temporada de cultivo.
- **CO2\_Emissions\_MT**
  - **Tipo:** Numérica continua
  - **Descripción:** Emisiones de CO2 (en toneladas métricas) relacionadas con actividades agrícolas o de la región.
- **Extreme\_Weather\_Events**
  - **Tipo:** Numérica entera
  - **Descripción:** Número de eventos climáticos extremos (e.g., sequías, inundaciones) durante la temporada.
- **Irrigation\_Access\_%**
  - **Tipo:** Numérica continua
  - **Descripción:** Porcentaje del área de cultivo que cuenta con acceso a sistemas de riego.
- **Pesticide\_Use\_KG\_per\_HA**
  - **Tipo:** Numérica continua
  - **Descripción:** Uso de pesticidas en kilogramos por hectárea.
- **Fertilizer\_Use\_KG\_per\_HA**
  - **Tipo:** Numérica continua
  - **Descripción:** Cantidad de fertilizantes utilizados por hectárea.
- **Soil\_Health\_Index**
  - **Tipo:** Numérica continua
  - **Descripción:** Índice que evalúa la calidad y salud del suelo.
- **Adaptation\_Strategies**
  - **Tipo:** Categórica
  - **Descripción:** Estrategias adoptadas para mitigar o adaptarse a los efectos del clima (e.g., rotación de cultivos, manejo de agua).
- **Economic\_Impact\_Million\_USD**
  - **Tipo:** Numérica continua
  - **Descripción:** Impacto económico de la venta de los cultivos (estimado en millones de dólares).

Este dataset proporciona una base rica para explorar las relaciones entre los factores ambientales y climáticos, y la productividad agrícola a nivel global.

Cargamos librerías

```

if (!require("tidyverse")) install.packages("tidyverse")
if (!require("effectsize")) install.packages("effectsize") # Para calcular eta squared
if (!require("knitr")) install.packages("knitr")# Instalar y cargar paquetes necesarios
if (!require("vcd")) install.packages("vcd") # Para calcular Cramér's V
if (!require("countrycode")) install.packages("countrycode") # Para agrupar países en continentes
if (!require("naniar")) install.packages("naniar") # Para visualizar datos faltantes
if (!require("patchwork")) install.packages("patchwork")
if (!require("gridExtra")) install.packages("gridExtra")
if (!require("reshape2")) install.packages("reshape2")
if (!require("boot")) install.packages("boot")
library(boot)
library(reshape2)
library(gridExtra)
library(patchwork)
library(naniar)
library(countrycode)
library(vcd)
library(knitr)
library(tidyverse)
library(effectsize)

```

Cargamos ahora los datos y pasamos `Crop_Yield_MT_per_HA` como la primer columna

```

file_path <- "Data/climate_change_impact_on_agriculture_2024.csv"
# Data obtained from https://www.kaggle.com/datasets/talhachoudary/global-agriculture-climate-impact-da
climate_data <- read.csv(file_path)
climate_data <- climate_data[, c("Crop_Yield_MT_per_HA", setdiff(names(climate_data), "Crop_Yield_MT_pe")]

```

## Datos faltantes

```

# Resumen de datos faltantes por columna
missing_summary <- climate_data %>%
  summarise(across(everything(), ~ sum(is.na(.)))) %>%
  pivot_longer(cols = everything(), names_to = "Variable", values_to = "Missing_Count") %>%
  mutate(Missing_Percentage = (Missing_Count / nrow(climate_data)) * 100)

# Mostrar resumen
missing_summary %>%
  arrange(desc(Missing_Percentage)) %>%
  knitr::kable(format = "markdown", caption = "Datos Faltantes por Variable")

```

Table 1: Datos Faltantes por Variable

Variable	Missing_Count	Missing_Percentage
Crop_Yield_MT_per_HA	0	0
Year	0	0
Country	0	0
Region	0	0
Crop_Type	0	0
Average_Temperature_C	0	0
Total_Precipitation_mm	0	0
CO2_Emissions_MT	0	0
Extreme_Weather_Events	0	0

Variable	Missing_Count	Missing_Percentage
Irrigation_Access_.	0	0
Pesticide_Use_KG_per_HA	0	0
Fertilizer_Use_KG_per_HA	0	0
Soil_Health_Index	0	0
Adaptation_Strategies	0	0
Economic_Impact_Million_USD	0	0

Como no hay datos faltantes, continuamos.

## Variables Numéricas: Boxplots e histogramas

Los siguientes boxplots e histogramas muestran la distribución de las variables numéricas en el dataset.

Boxplot e histograma de `Crop_Yield_MT_per_HA`

```
x <- climate_data$Crop_Yield_MT_per_HA

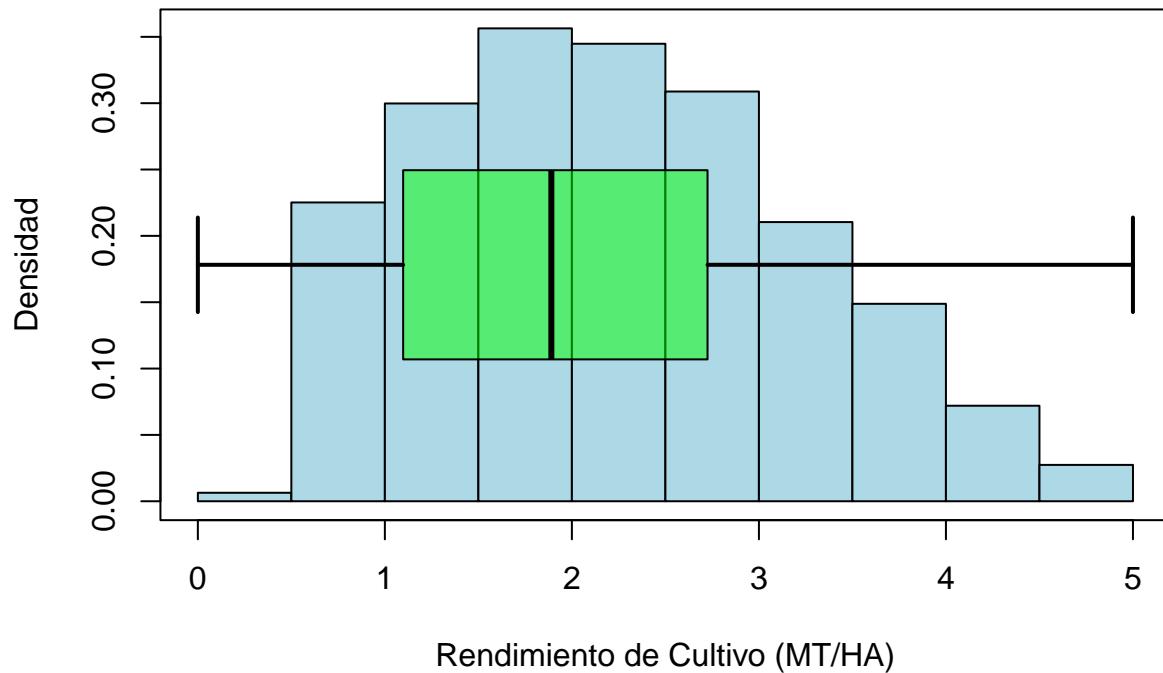
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Productividad Agrícola",
      xlab = "Rendimiento de Cultivo (MT/HA)",
      ylab = "Densidad")

par(new = TRUE)

boxplot(x, horizontal = TRUE, axes = FALSE,
        col = rgb(0, 1, 0, alpha = 0.5),
        at = 0.25,
        height = 0.005,
        whisklty = 1, whisklwd = 2, staplewex = 0.5, staplelwd = 2)

box()
```

## Histograma y Boxplot de Productividad Agrícola



Boxplot e histograma de Year

```
x <- climate_data$Year

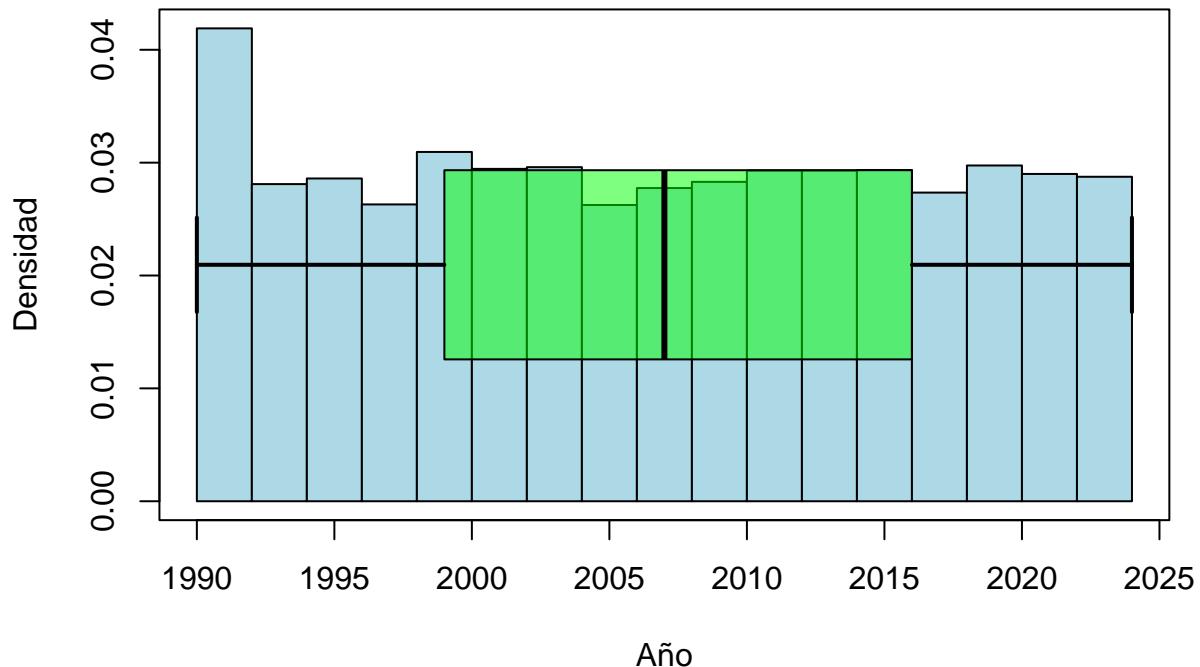
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot del Año",
      xlab = "Año",
      ylab = "Densidad")

par(new = TRUE)

boxplot(x, horizontal = TRUE, axes = FALSE,
        col = rgb(0, 1, 0, alpha = 0.5),
        at = 0.25,
        height = 0.005,
        whisklty = 1, whisklwd = 2, staplewex = 0.5, staplelwd = 2)

box()
```

## Histograma y Boxplot del Año



Boxplot e histograma de Average\_Temperature\_C

```
x <- climate_data$Average_Temperature_C

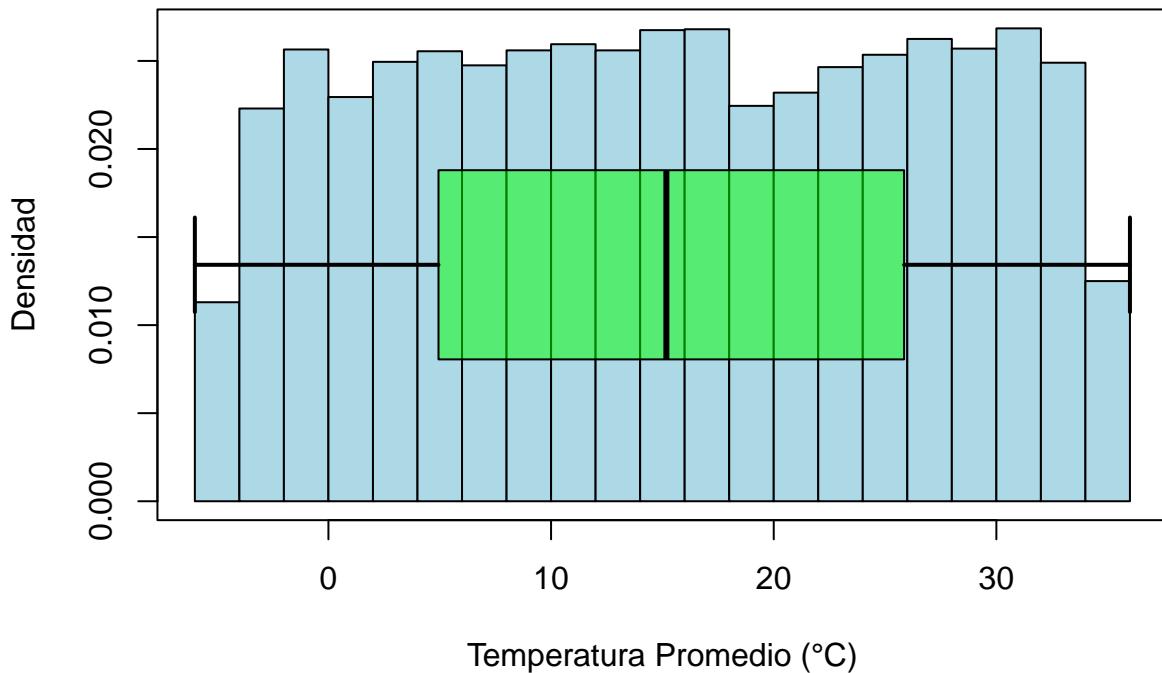
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Temperatura Promedio",
      xlab = "Temperatura Promedio (°C)",
      ylab = "Densidad")

par(new = TRUE)

boxplot(x, horizontal = TRUE, axes = FALSE,
        col = rgb(0, 1, 0, alpha = 0.5),
        at = 0.25,
        height = 0.005,
        whisklty = 1, whisklwd = 2, staplewex = 0.5, staplelwd = 2)

box()
```

## Histograma y Boxplot de Temperatura Promedio



Boxplot e historgama de Total\_Precipitation\_mm

```
x <- climate_data$Total_Precipitation_mm

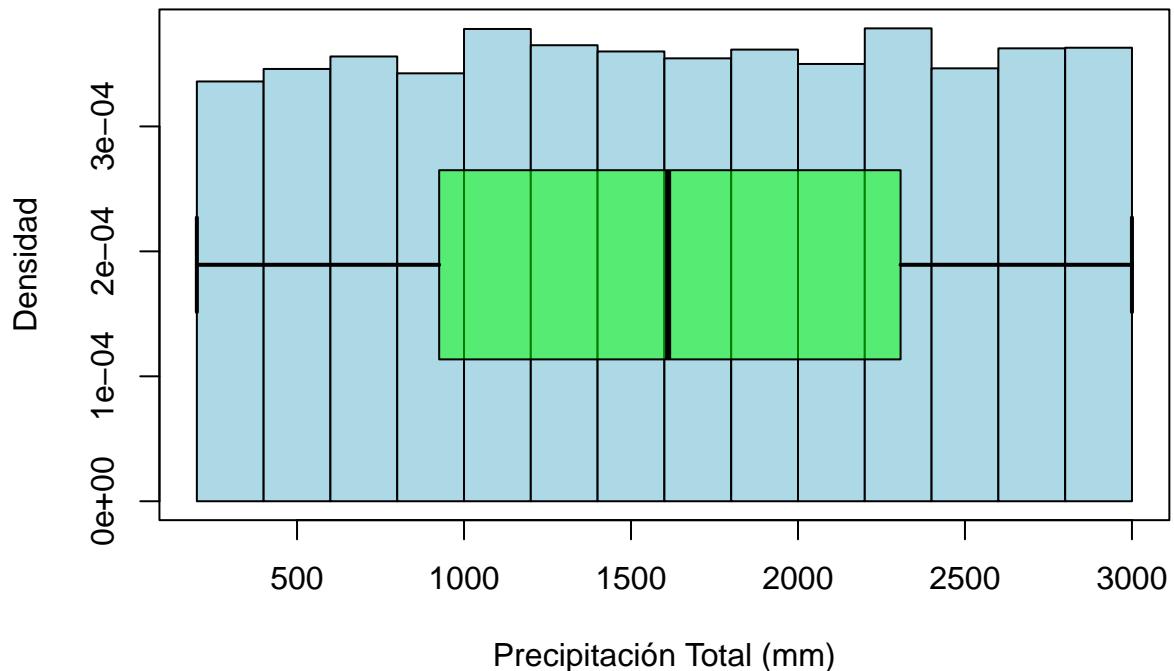
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Precipitación Total",
      xlab = "Precipitación Total (mm)",
      ylab = "Densidad")

par(new = TRUE)

boxplot(x, horizontal = TRUE, axes = FALSE,
        col = rgb(0, 1, 0, alpha = 0.5),
        at = 0.25,
        height = 0.005,
        whisklty = 1, whisklwd = 2, staplewex = 0.5, staplelwd = 2)

box()
```

## Histograma y Boxplot de Precipitación Total



Boxplot e histograma de C02\_Emissions\_MT

```
x <- climate_data$C02_Emissions_MT

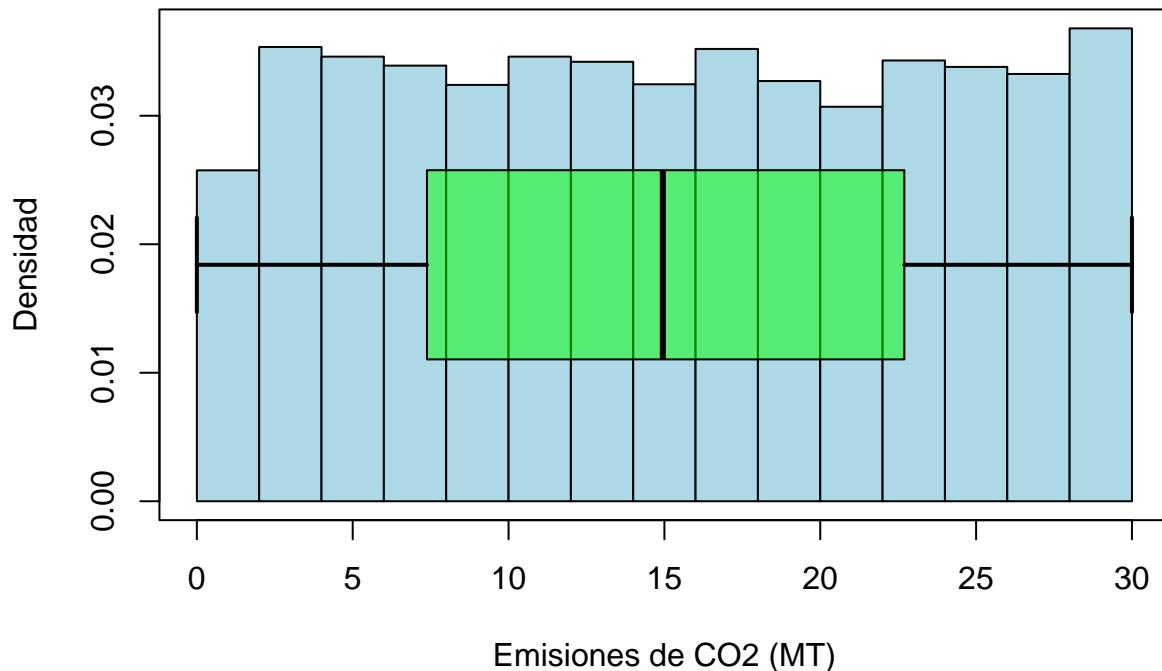
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Emisiones de CO2",
      xlab = "Emisiones de CO2 (MT)",
      ylab = "Densidad")

par(new = TRUE)

boxplot(x, horizontal = TRUE, axes = FALSE,
        col = rgb(0, 1, 0, alpha = 0.5),
        at = 0.25,
        height = 0.005,
        whisklty = 1, whisklwd = 2, staplewex = 0.5, staplelwd = 2)

box()
```

## Histograma y Boxplot de Emisiones de CO2



Boxplot e histograma de Extreme\_Weather\_Events

```
x <- climate_data$Extreme_Weather_Events

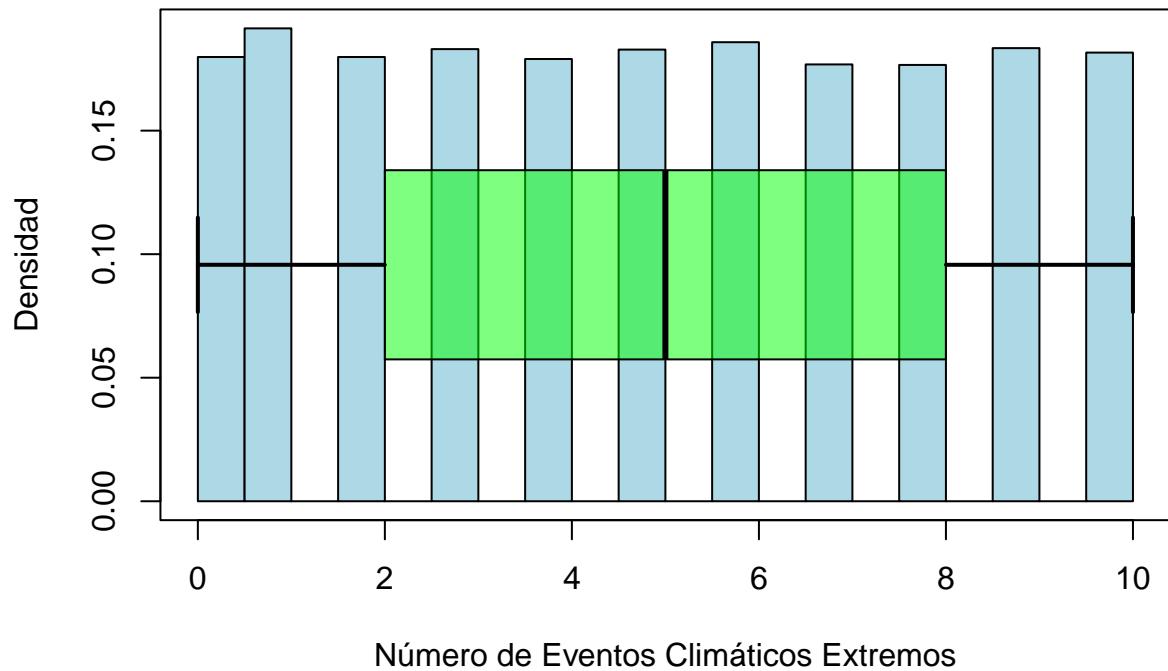
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Eventos Climáticos Extremos",
      xlab = "Número de Eventos Climáticos Extremos",
      ylab = "Densidad")

par(new = TRUE)

boxplot(x, horizontal = TRUE, axes = FALSE,
        col = rgb(0, 1, 0, alpha = 0.5),
        at = 0.25,
        height = 0.005,
        whisklty = 1, whisklwd = 2, staplewex = 0.5, staplelw = 2)

box()
```

## Histograma y Boxplot de Eventos Climáticos Extremos



Boxplot e histograma de Irrigation\_Access\_%

```
x <- climate_data$Irrigation_Access_.

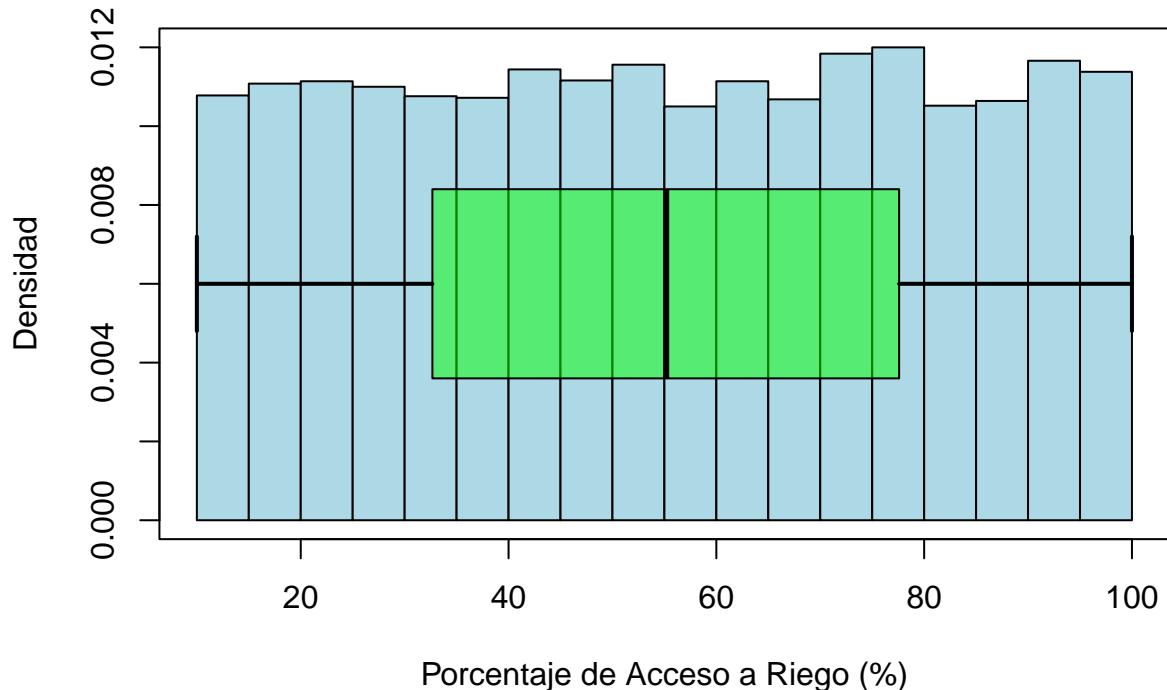
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Acceso a Riego",
      xlab = "Porcentaje de Acceso a Riego (%)",
      ylab = "Densidad")

par(new = TRUE)

boxplot(x, horizontal = TRUE, axes = FALSE,
        col = rgb(0, 1, 0, alpha = 0.5),
        at = 0.25,
        height = 0.005,
        whisklty = 1, whisklwd = 2, staplewex = 0.5, staplelwd = 2)

box()
```

## Histograma y Boxplot de Acceso a Riego



Boxplot e histograma de Pesticide\_Use\_KG\_per\_HA

```
x <- climate_data$Pesticide_Use_KG_per_HA

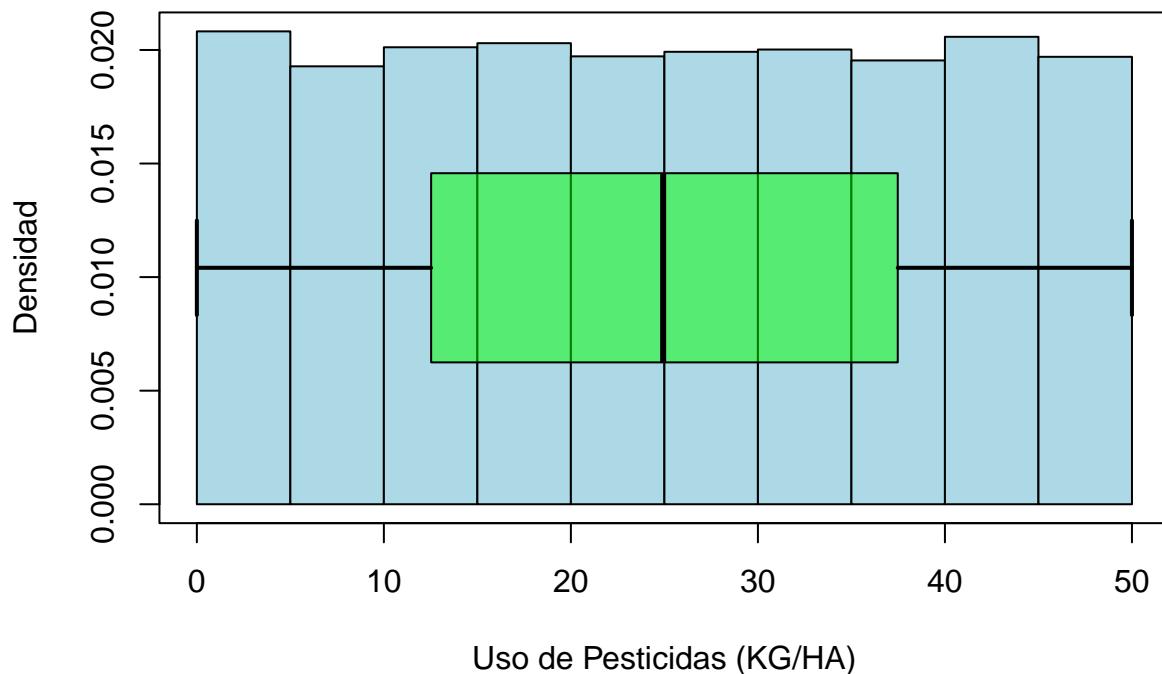
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Uso de Pesticidas",
      xlab = "Uso de Pesticidas (KG/HA)",
      ylab = "Densidad")

par(new = TRUE)

boxplot(x, horizontal = TRUE, axes = FALSE,
        col = rgb(0, 1, 0, alpha = 0.5),
        at = 0.25,
        height = 0.005,
        whisklty = 1, whisklwd = 2, staplewex = 0.5, staplelwd = 2)

box()
```

## Histograma y Boxplot de Uso de Pesticidas



Boxplot e histograma de Fertilizer\_Use\_KG\_per\_HA

```
x <- climate_data$Fertilizer_Use_KG_per_HA

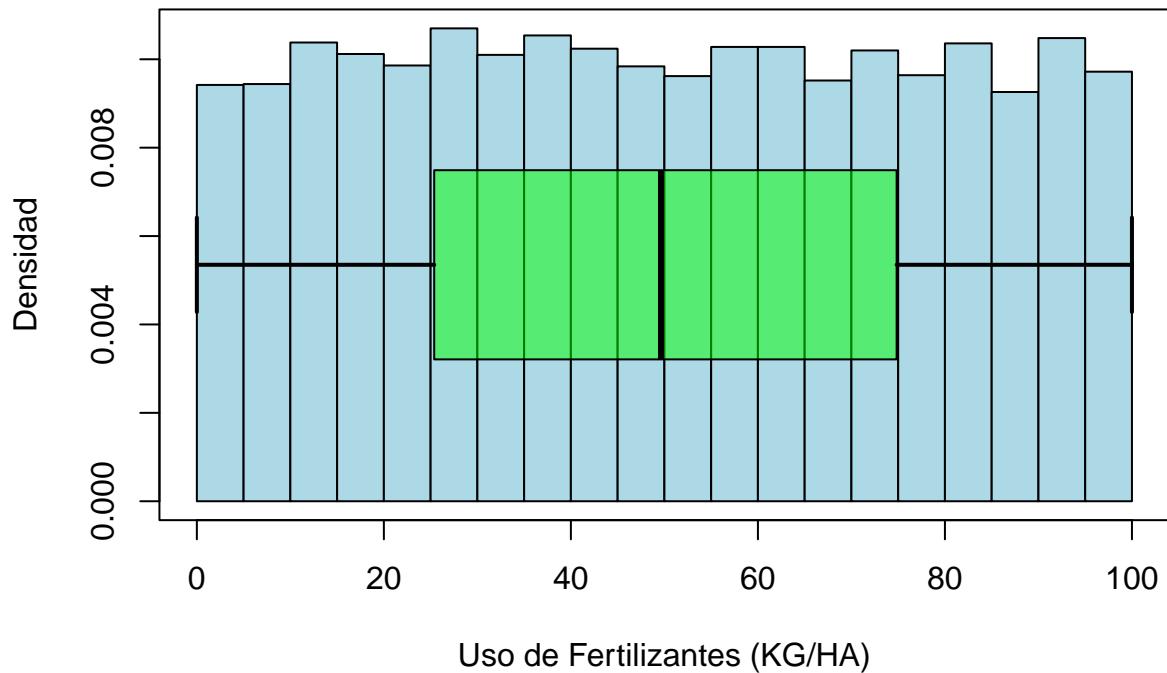
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Uso de Fertilizantes",
      xlab = "Uso de Fertilizantes (KG/HA)",
      ylab = "Densidad")

par(new = TRUE)

boxplot(x, horizontal = TRUE, axes = FALSE,
        col = rgb(0, 1, 0, alpha = 0.5),
        at = 0.25,
        height = 0.005,
        whisklty = 1, whisklwd = 2, staplewex = 0.5, staplelwd = 2)

box()
```

## Histograma y Boxplot de Uso de Fertilizantes



Boxplot e histograma de Soil\_Health\_Index

```
x <- climate_data$Soil_Health_Index

hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot del Índice de Salud del Suelo",
      xlab = "Índice de Salud del Suelo",
      ylab = "Densidad")

par(new = TRUE)

boxplot(x, horizontal = TRUE, axes = FALSE,
        col = rgb(0, 1, 0, alpha = 0.5),
        at = 0.25,
        height = 0.005,
        whisklty = 1, whisklwd = 2, staplewex = 0.5, staplelwd = 2)

box()
```

## Histograma y Boxplot del Índice de Salud del Suelo



Boxplot e histograma de Economic\_Impact\_Million\_USD

```
x <- climate_data$Economic_Impact_Million_USD

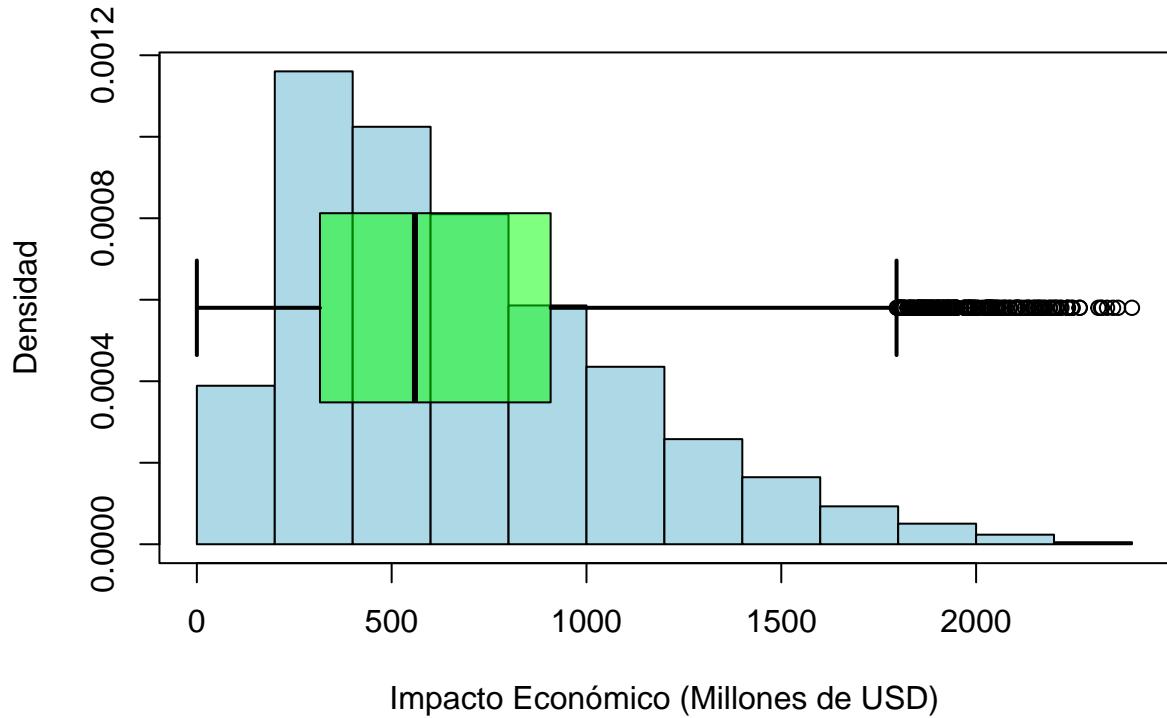
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot del Impacto Económico",
      xlab = "Impacto Económico (Millones de USD)",
      ylab = "Densidad")

par(new = TRUE)

boxplot(x, horizontal = TRUE, axes = FALSE,
        col = rgb(0, 1, 0, alpha = 0.5),
        at = 0.25,
        height = 0.005,
        whisklty = 1, whisklwd = 2, staplewex = 0.5, staplelwd = 2)

box()
```

## Histograma y Boxplot del Impacto Económico



### Variables Categóricas: Frecuencias

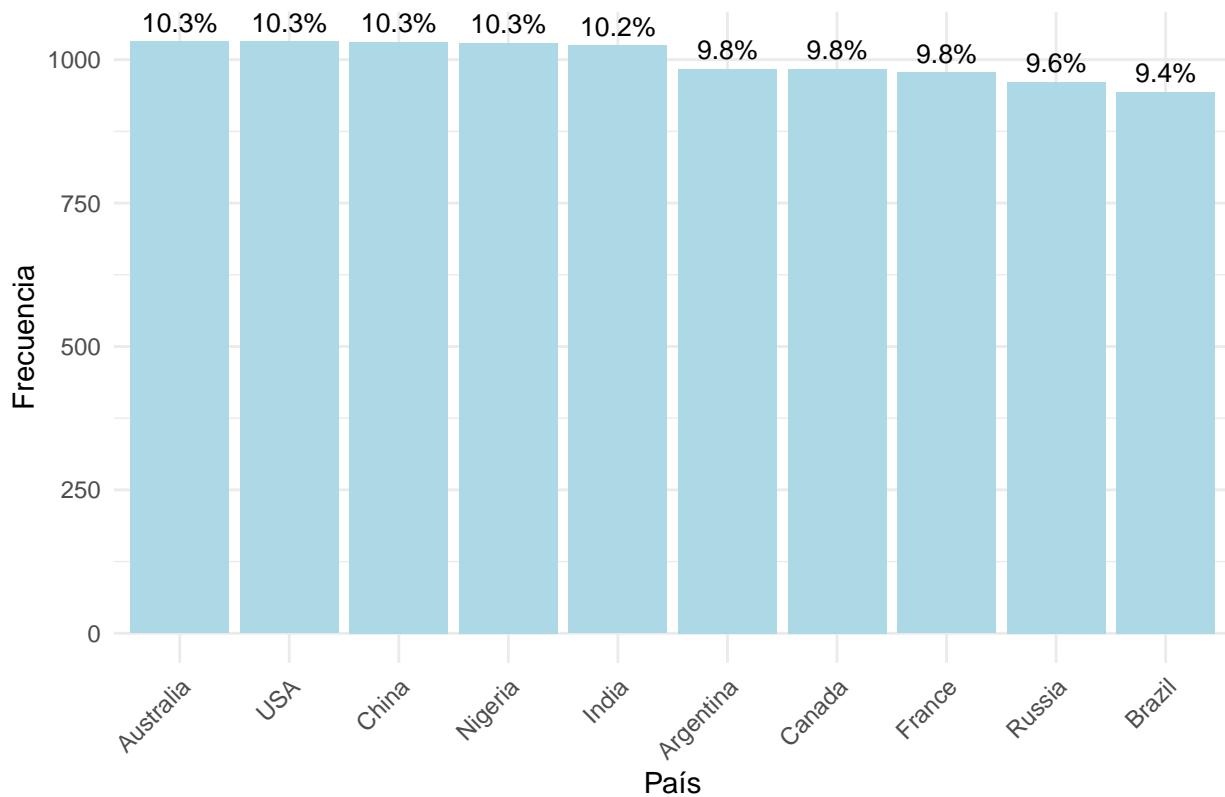
A continuación, se presentan las frecuencias para las variables categóricas del dataset.

#### Frecuencia de Países Country

```
# Crear la tabla de frecuencias y calcular el porcentaje
var_freq <- as.data.frame(table(climate_data$Country))
colnames(var_freq) <- c("Country", "Frequency")
var_freq <- var_freq %>%
  mutate(Percentage = (Frequency / sum(Frequency)) * 100)

# Crear la gráfica combinada de frecuencia y porcentaje
ggplot(var_freq, aes(x = reorder(Country, -Frequency), y = Frequency)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  geom_text(aes(label = paste0(round(Percentage, 1), "%")),
            vjust = -0.5, size = 3.5, color = "black") +
  theme_minimal() +
  labs(
    title = "Frecuencia y Porcentaje de Países",
    x = "País",
    y = "Frecuencia"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

## Frecuencia y Porcentaje de Países

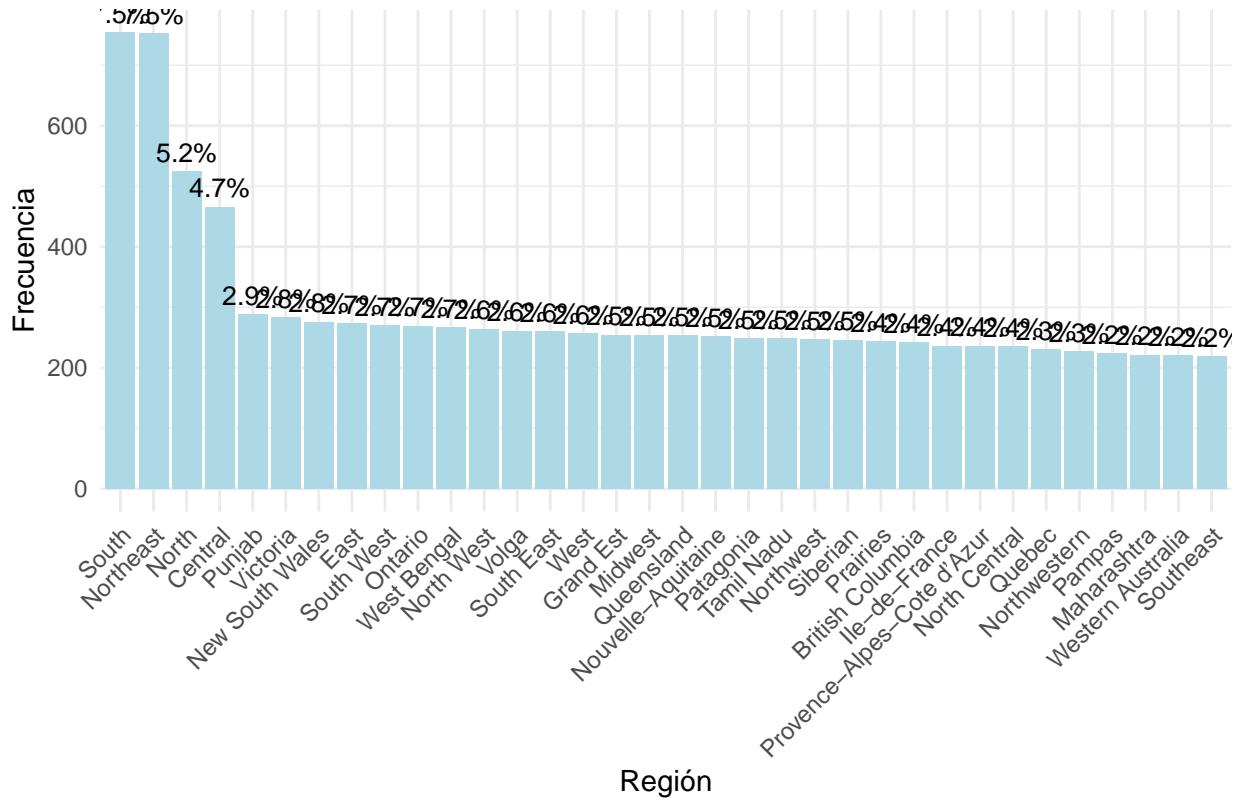


### Frecuencia de Regiones Region

```
# Crear la tabla de frecuencias y calcular el porcentaje
var_freq <- as.data.frame(table(climate_data$Region))
colnames(var_freq) <- c("Region", "Frequency")
var_freq <- var_freq %>%
  mutate(Percentage = (Frequency / sum(Frequency)) * 100)

# Crear la gráfica combinada de frecuencia y porcentaje
ggplot(var_freq, aes(x = reorder(Region, -Frequency), y = Frequency)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  geom_text(aes(label = paste0(round(Percentage, 1), "%")),
            vjust = -0.5, size = 3.5, color = "black") +
  theme_minimal() +
  labs(
    title = "Frecuencia y Porcentaje de Regiones",
    x = "Región",
    y = "Frecuencia"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

## Frecuencia y Porcentaje de Regiones

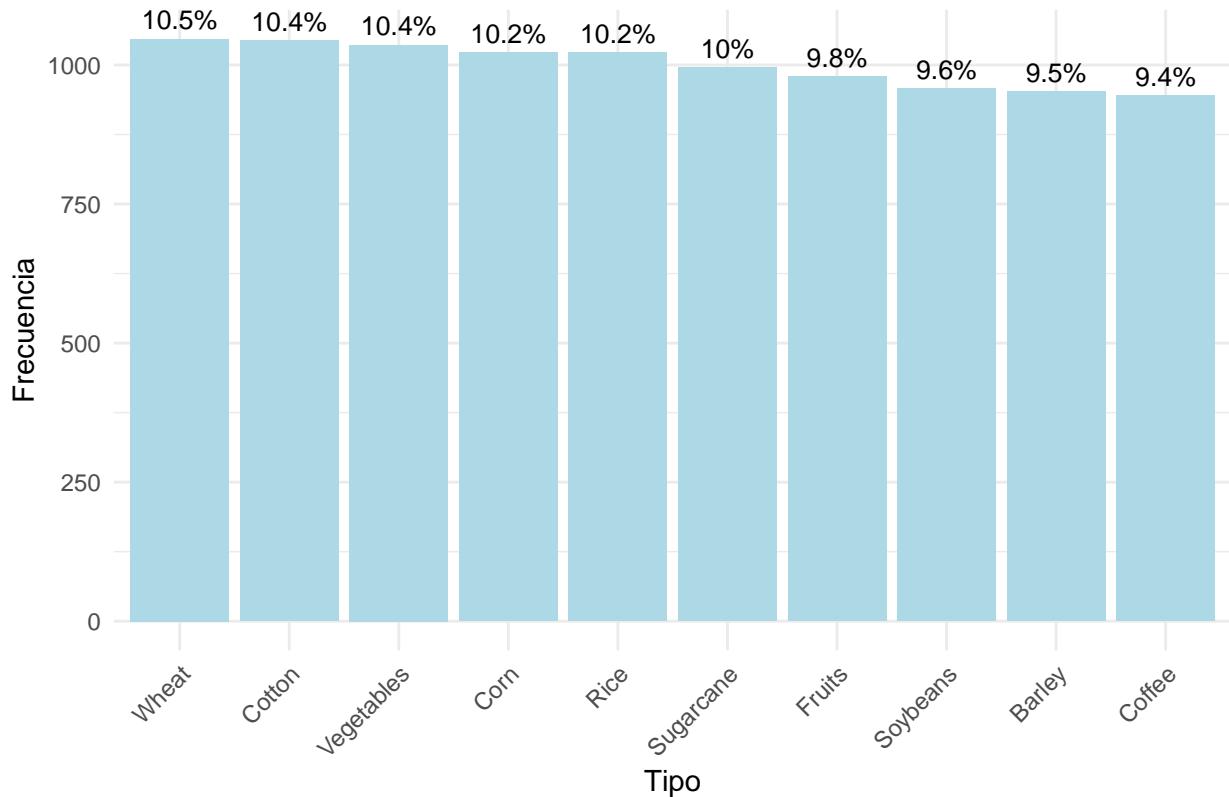


## Frecuencia de Tipos de Cultivo Crop\_Type

```
# Crear la tabla de frecuencias y calcular el porcentaje
var_freq <- as.data.frame(table(climate_data$Crop_Type))
colnames(var_freq) <- c("Crop_Type", "Frequency")
var_freq <- var_freq %>%
  mutate(Percentage = (Frequency / sum(Frequency)) * 100)

# Crear la gráfica combinada de frecuencia y porcentaje
ggplot(var_freq, aes(x = reorder(Crop_Type, -Frequency), y = Frequency)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  geom_text(aes(label = paste0(round(Percentage, 1), "%")),
            vjust = -0.5, size = 3.5, color = "black") +
  theme_minimal() +
  labs(
    title = "Frecuencia y Porcentaje de Tipos de Cultivo",
    x = "Tipo",
    y = "Frecuencia"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

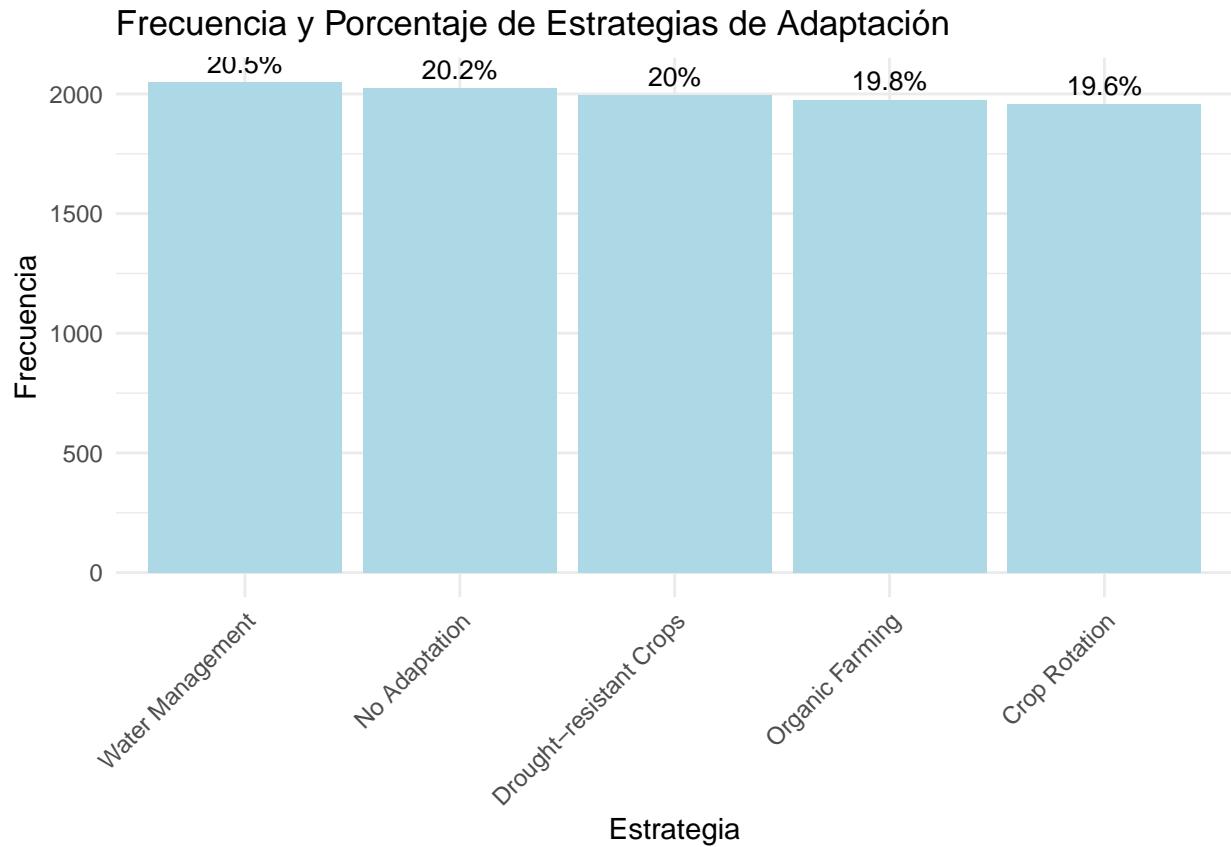
## Frecuencia y Porcentaje de Tipos de Cultivo



Frecuencia de Estrategias de Adaptación `Adaptation_Strategies`

```
# Crear la tabla de frecuencias y calcular el porcentaje
var_freq <- as.data.frame(table(climate_data$Adaptation_Strategies))
colnames(var_freq) <- c("Adaptation_Strategies", "Frequency")
var_freq <- var_freq %>%
  mutate(Percentage = (Frequency / sum(Frequency)) * 100)

# Crear la gráfica combinada de frecuencia y porcentaje
ggplot(var_freq, aes(x = reorder(Adaptation_Strategies, -Frequency), y = Frequency)) +
  geom_bar(stat = "identity", fill = "lightblue") +
  geom_text(aes(label = paste0(round(Percentage, 1), "%")),
            vjust = -0.5, size = 3.5, color = "black") +
  theme_minimal() +
  labs(
    title = "Frecuencia y Porcentaje de Estrategias de Adaptación",
    x = "Estrategia",
    y = "Frecuencia"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



## Pairwise scatter plots

Los gráficos de dispersión por pares son una herramienta visual poderosa para analizar relaciones entre múltiples variables numéricas. Al graficar cada par de variables en un conjunto de datos, podemos identificar patrones, tendencias, y posibles correlaciones lineales o no lineales entre ellas.

Este enfoque es particularmente útil para:

- Detectar relaciones lineales o no lineales entre variables.
- Evitar multicolinealidad.

```
numeric_vars <- climate_data[, sapply(climate_data, is.numeric)]
```

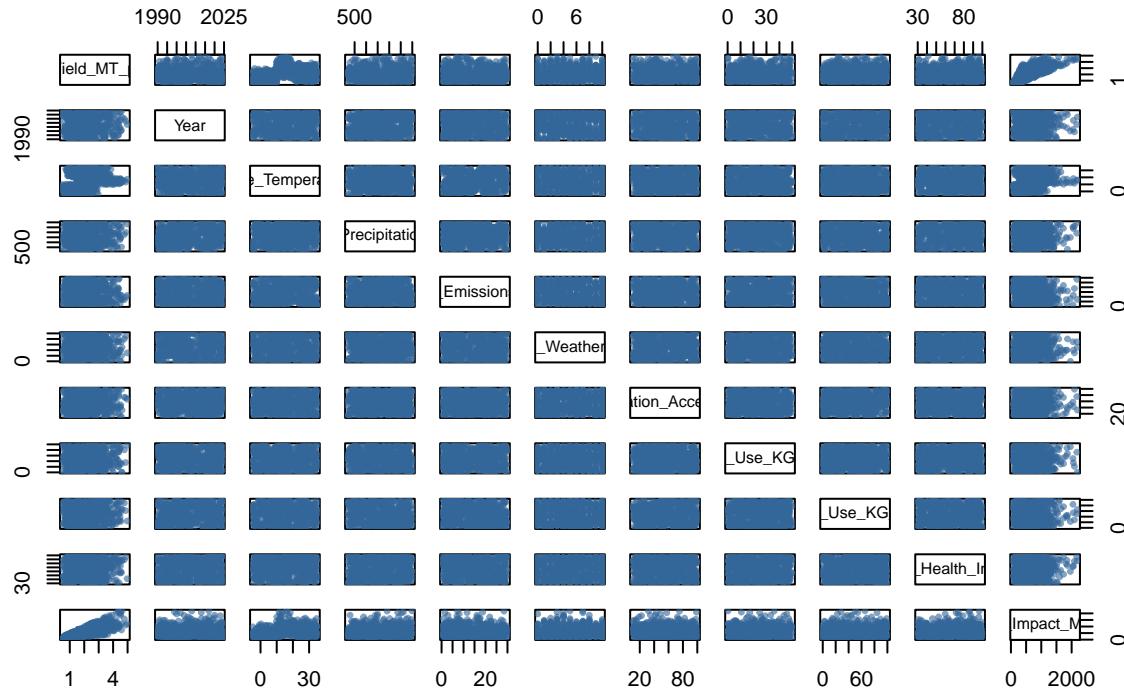
Hagamos el plot

```
# Fijar la semilla para reproducibilidad
set.seed(123)

# Seleccionar una muestra aleatoria de 500 filas
sample_size <- 500
climate_data_sample <- numeric_vars[sample(nrow(numeric_vars), sample_size), ]

# Dibujar el gráfico de pares con la muestra
pairs(climate_data_sample,
      main = "Pares para una Muestra Aleatoria de Datos",
      pch = 19,
      cex = 0.5,
      col = rgb(0.2, 0.4, 0.6, 0.6))
```

## Pares para una Muestra Aleatoria de Datos



### Coeficientes empíricos de Pearson

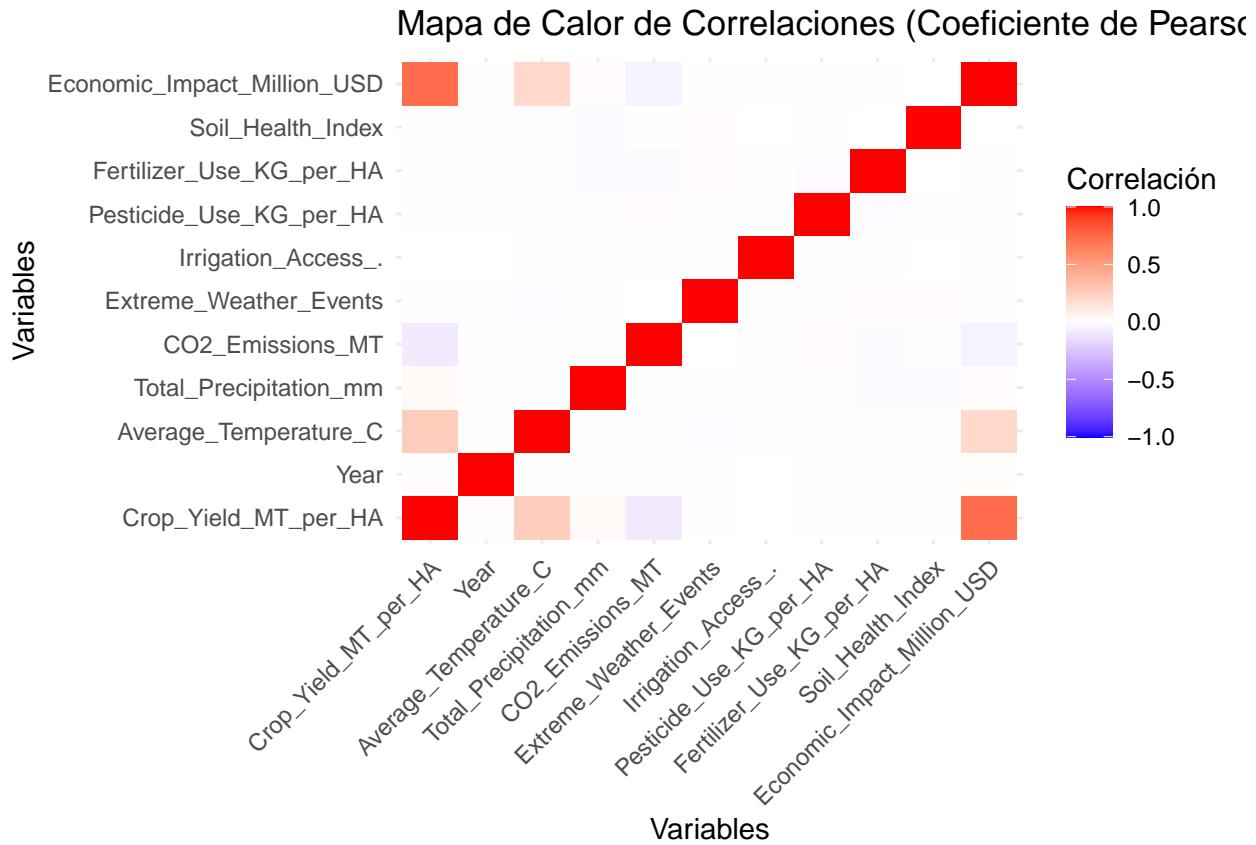
El coeficiente de correlación de Pearson mide la relación lineal entre variables numéricas. Un valor cercano a 1 o -1 indica una fuerte correlación positiva o negativa, respectivamente, mientras que un valor cercano a 0 indica una correlación débil o inexistente.

A continuación, calculamos la matriz de correlaciones para nuestras variables numéricas:

```
# Calcular la matriz de correlaciones
cor_matrix <- cor(numeric_vars, use = "complete.obs") # Ignorar valores NA

# Convertir la matriz en un formato largo para ggplot
cor_long <- melt(cor_matrix)

# Crear el mapa de calor
ggplot(cor_long, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0,
                       limit = c(-1, 1), space = "Lab", name = "Correlación") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(
    title = "Mapa de Calor de Correlaciones (Coeficiente de Pearson)",
    x = "Variables",
    y = "Variables"
  )
```



### Cálculo de Eta Squared para Variables Numéricas y Categóricas

El coeficiente Eta Squared mide la proporción de la varianza explicada de una variable numérica por una variable categórica. Este cálculo es útil para evaluar qué tan fuerte es la relación entre variables categóricas y numéricas en un conjunto de datos.

Un valor de Eta Squared más alto indica que la variable categórica tiene un mayor efecto sobre la variable numérica, mientras que un valor cercano a 0 sugiere que la variable categórica no tiene un efecto significativo.

Este análisis es particularmente valioso para:

- Explorar la fuerza de asociación entre grupos definidos por una categoría y una medida cuantitativa.
- Identificar variables categóricas relevantes para el análisis predictivo o explicativo.

```
# Separar variables numéricas y categóricas
numeric_vars <- climate_data[, sapply(climate_data, is.numeric)]
categorical_vars <- climate_data[, sapply(climate_data, is.character)]
```

```
# Función para calcular eta squared
calculate_eta_squared <- function(numeric_col, categorical_col) {
  # Crear un data frame temporal para trabajar
  temp_data <- data.frame(
    numeric_col = numeric_col,
    categorical_col = categorical_col
  )
  # Remover NA para evitar errores
  temp_data <- na.omit(temp_data)
```

```

# Calcular eta squared
model <- aov(numeric_col ~ categorical_col, data = temp_data)
eta_squared <- eta_squared(model)
return(eta_squared$Eta2[1]) # Devuelve el eta squared para el efecto principal
}

# Crear un data frame para almacenar los resultados
results <- expand.grid(
  Numeric = colnames(numeric_vars),
  Categorical = colnames(categorical_vars)
) %>%
  rowwise() %>%
  mutate(Eta_Squared = calculate_eta_squared(
    numeric_vars[[Numeric]],
    categorical_vars[[Categorical]]
  )) %>%
  ungroup()

# Mostrar resultados en una tabla ordenada
results <- results %>% arrange(desc(Eta_Squared))
kable(
  results,
  caption = "Resultados de Eta Squared para cada par de variables numéricas y categóricas"
)

```

Table 2: Resultados de Eta Squared para cada par de variables numéricas y categóricas

Numeric	Categorical	Eta_Squared
Irrigation_Access_	Region	0.0035757
Extreme_Weather_Events	Region	0.0034193
Fertilizer_Use_KG_per_HA	Region	0.0033998
Average_Temperature_C	Region	0.0033496
Soil_Health_Index	Region	0.0032554
Economic_Impact_Million_USD	Region	0.0031778
Crop_Yield_MT_per_HA	Region	0.0029636
Year	Region	0.0028643
Total_Precipitation_mm	Region	0.0026705
CO2_Emissions_MT	Region	0.0026143
Pesticide_Use_KG_per_HA	Region	0.0024162
Extreme_Weather_Events	Country	0.0016987
Total_Precipitation_mm	Country	0.0016608
Crop_Yield_MT_per_HA	Crop_Type	0.0013293
Pesticide_Use_KG_per_HA	Crop_Type	0.0012310
Economic_Impact_Million_USD	Crop_Type	0.0012216
Fertilizer_Use_KG_per_HA	Country	0.0011640
Average_Temperature_C	Crop_Type	0.0011014
Average_Temperature_C	Country	0.0009446
Soil_Health_Index	Country	0.0007507
Year	Crop_Type	0.0007394
Economic_Impact_Million_USD	Adaptation_Strategies	0.0007364
Irrigation_Access_	Crop_Type	0.0007361
Total_Precipitation_mm	Crop_Type	0.0007300

Numeric	Categorical	Eta_Squared
Pesticide_Use_KG_per_HA	Country	0.0007269
Year	Country	0.0007260
Economic_Impact_Million_USD	Country	0.0007090
CO2_Emissions_MT	Country	0.0006896
Extreme_Weather_Events	Crop_Type	0.0006787
Irrigation_Access_.	Country	0.0006506
Pesticide_Use_KG_per_HA	Adaptation_Strategies	0.0006409
CO2_Emissions_MT	Crop_Type	0.0005718
Fertilizer_Use_KG_per_HA	Crop_Type	0.0005520
Average_Temperature_C	Adaptation_Strategies	0.0005228
Year	Adaptation_Strategies	0.0005058
Crop_Yield_MT_per_HA	Country	0.0004081
Crop_Yield_MT_per_HA	Adaptation_Strategies	0.0003941
Extreme_Weather_Events	Adaptation_Strategies	0.0003434
CO2_Emissions_MT	Adaptation_Strategies	0.0003225
Soil_Health_Index	Crop_Type	0.0002248
Irrigation_Access_.	Adaptation_Strategies	0.0001925
Total_Precipitation_mm	Adaptation_Strategies	0.0001671
Soil_Health_Index	Adaptation_Strategies	0.0001101
Fertilizer_Use_KG_per_HA	Adaptation_Strategies	0.0000921

## Cálculo de Cramér's V para Variables Categóricas

A continuación, se calcula el valor de Cramér's V ( $V$ ) para cada combinación posible de dos variables categóricas del dataset.

```
# Seleccionar solo variables categóricas
categorical_vars <- climate_data[, sapply(climate_data, is.character)]


# Función para calcular Cramér's V
calculate_cramers_v <- function(var1, var2) {
  # Crear tabla de contingencia
  contingency_table <- table(var1, var2)
  # Calcular Cramér's V
  cramers_v <- assocstats(contingency_table)$cramer
  return(cramers_v)
}

# Generar combinaciones de todas las variables categóricas
categorical_combinations <- combn(colnames(categorical_vars), 2, simplify = FALSE)

# Calcular Cramér's V para cada combinación
results_cramers_v <- map_dfr(
  categorical_combinations,
  ~ tibble(
    Var1 = .x[1],
    Var2 = .x[2],
    Cramers_V = calculate_cramers_v(categorical_vars[[.x[1]]], categorical_vars[[.x[2]]]))
  )
)

# Ordenar resultados por el valor de Cramér's V
```

```

results_cramers_v <- results_cramers_v %>% arrange(desc(Cramers_V))
kable(
  results_cramers_v,
  caption = "Resultados de Cramér's V para cada par de variables categóricas"
)

```

Table 3: Resultados de Cramér's V para cada par de variables categóricas

Var1	Var2	Cramers_V
Country	Region	0.9124265
Region	Crop_Type	0.0622969
Region	Adaptation_Strategies	0.0578097
Crop_Type	Adaptation_Strategies	0.0336675
Country	Crop_Type	0.0327953
Country	Adaptation_Strategies	0.0278957

## Modificaciones a los datos

Como podemos observar, contamos con una amplia variedad de categorías. Para evitar multicolinealidad o una segmentación excesiva de nuestro conjunto de datos, optaremos por eliminar la variable `Region` y agrupar los valores de la variable `Country` por continentes.

```

# Elimina 'Region'
climate_data <- climate_data %>% select(-Region)

# Agrupar países en continentes
climate_data$Continent <- countrycode(climate_data$Country, "country.name", "continent")

# Elimina 'Country'
climate_data <- climate_data %>% select(-Country)

```

Eliminaremos la columna `Year` del dataset porque su inclusión podría introducir una dependencia temporal que no es el enfoque principal de este análisis.

```

# Elimina 'Year'
climate_data <- climate_data %>% select(-Year)

```

La variable `Economic_Impact_Million_USD` representa el impacto económico estimado asociado a la productividad agrícola. Sin embargo, esta métrica se calcula posteriormente a la cosecha, ya que depende directamente de los rendimientos obtenidos y de factores externos como los precios de mercado y las políticas económicas. Por esta razón, no es adecuada para incluirla como predictor en este análisis.

```

# Elimina 'Economic_Impact_Million_USD'
climate_data <- climate_data %>% select(-Economic_Impact_Million_USD)

```

Estandarizaremos los datos antes de ajustarlos al modelo. La estandarización transforma las variables numéricas para que tengan media 0 y desviación estándar 1, lo que es especialmente importante cuando las variables tienen diferentes unidades o rangos. Esto permite que el modelo lineal generalizado (GLM) considere todas las variables en una escala comparable, reduciendo el impacto de aquellas con valores más grandes en la estimación de los coeficientes.

```

# Función para estandarizar solo las variables predictoras
standardize_predictors <- function(data, response_var) {
  numeric_cols <- sapply(data, is.numeric) # Identificar columnas numéricas

```

```

predictors <- numeric_cols & names(data) != response_var # Excluir la variable objetivo
data[, predictors] <- scale(data[, predictors]) # Estandarizar solo las predictoras
return(data)
}

# Aplicar la estandarización al dataset
response_var <- "Crop_Yield_MT_per_HA" # Nombre de la variable objetivo
climate_data <- standardize_predictors(climate_data, response_var)

```

## Modelos

El histograma de la variable Crop\_Yield\_MT\_per\_HA muestra una distribución asimétrica positiva, lo que indica que los datos no son normales y están mejor representados por una distribución perteneciente a la familia exponencial, como la distribución gamma. Dado este comportamiento, utilizaremos un modelo lineal generalizado (GLM) con una distribución gamma para modelar esta variable.

```

# Ajustar el modelo GLM con enlace logarítmico
glm_log <- glm(
  Crop_Yield_MT_per_HA ~ .,
  data = climate_data,
  family = Gamma(link = "log")
)

```

Inicialmente, empleamos una función liga logarítmica (log). Ahora, también exploraremos el uso de la función de enlace inversa (inverse).

```

# Ajustar el modelo GLM con enlace inverso
glm_inverse <- glm(
  Crop_Yield_MT_per_HA ~ .,
  data = climate_data,
  family = Gamma(link = "inverse")
)
summary(glm_inverse)

##
## Call:
## glm(formula = Crop_Yield_MT_per_HA ~ ., family = Gamma(link = "inverse"),
##      data = climate_data)
##
## Coefficients:
## (Intercept)          Estimate Std. Error t value
## Crop_TypeCoffee     4.290e-01  9.068e-03 47.312
## Crop_TypeCorn        1.619e-02  8.767e-03  1.846
## Crop_TypeCotton     1.555e-02  8.599e-03  1.808
## Crop_TypeFruits     2.429e-02  8.654e-03  2.807
## Crop_TypeRice        3.566e-03  8.556e-03  0.417
## Crop_TypeSoybeans   9.387e-03  8.544e-03  1.099
## Crop_TypeSugarcane  1.469e-02  8.729e-03  1.683
## Crop_TypeVegetables 5.346e-03  8.576e-03  0.623
## Crop_TypeWheat       1.877e-02  8.613e-03  2.180
## Average_Temperature_C -5.252e-02 1.945e-03 -26.997
## Total_Precipitation_mm -5.208e-03 1.925e-03 -2.705
## CO2_Emissions_MT     1.706e-02  1.923e-03  8.872
## Extreme_Weather_Events 8.289e-05 1.923e-03  0.043

```

```

## Irrigation_Access_.          -6.560e-04 1.923e-03 -0.341
## Pesticide_Use_KG_per_HA      9.835e-04 1.920e-03  0.512
## Fertilizer_Use_KG_per_HA     -1.883e-03 1.921e-03 -0.980
## Soil_Health_Index             3.817e-04 1.919e-03  0.199
## Adaptation_StrategiesDrought-resistant Crops 3.736e-03 6.066e-03  0.616
## Adaptation_StrategiesNo Adaptation       2.627e-03 6.055e-03  0.434
## Adaptation_StrategiesOrganic Farming      4.159e-03 6.087e-03  0.683
## Adaptation_StrategiesWater Management    8.742e-03 6.076e-03  1.439
## ContinentAmericas            9.926e-03 6.635e-03  1.496
## ContinentAsia                 6.215e-03 7.234e-03  0.859
## ContinentEurope               1.379e-02 7.360e-03  1.873
## ContinentOceania              9.123e-03 8.412e-03  1.084
##
## Pr(>|t|)
##   < 2e-16 ***
##   0.06488 .
##   0.07063 .
##   0.00500 **
##   0.67682
##   0.27191
##   0.09246 .
##   0.53306
##   0.02931 *
##   0.27182
##   < 2e-16 ***
##   0.00684 **
##   < 2e-16 ***
##   0.96562
##   0.73303
##   0.60849
##   0.32704
##   0.84231
##   0.53792
##   0.66441
##   0.49445
##   0.15024
##   0.13466
##   0.39027
##   0.06108 .
##   0.27819
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.1878978)
##
## Null deviance: 2291.5 on 9999 degrees of freedom
## Residual deviance: 2132.3 on 9974 degrees of freedom
## AIC: 27172
##
## Number of Fisher Scoring iterations: 5

```

Planteamos adicionalmente dos modelos reducidos

```

# Ajustar el modelo reducido GLM con enlace logaritmico
glm_log.red <- glm(
  Crop_Yield_MT_per_HA ~ Average_Temperature_C + CO2_Emissions_MT + Total_Precipitation_mm + Crop_Type

```

```

    data = climate_data,
    family = Gamma(link = "log")
)

# Ajustar el modelo reducido GLM con enlace inverso
glm_inverse.red <- glm(
  Crop_Yield_MT_per_HA ~ Average_Temperature_C + CO2_Emissions_MT + Total_Precipitation_mm + Crop_Type
  data = climate_data,
  family = Gamma(link = "inverse")
)

```

## Bondad de Ajuste

La bondad de ajuste también implica verificar que el modelo cumple con los supuestos teóricos bajo los cuales se construyó. En el caso de un modelo lineal generalizado (GLM) con distribución gamma y enlace logarítmico, es importante asegurarse de que:

1. **La dispersión residual** es razonable y consistente con los supuestos.
2. **El enlace logarítmico** es adecuado para relacionar las variables predictoras con la variable objetivo.
3. **No existen valores atípicos extremos** que influyan de manera desproporcionada en el modelo.

A continuación, verificaremos estos aspectos mediante visualizaciones y métricas clave para asegurar que el modelo respeta sus supuestos fundamentales. Esto es crucial para garantizar la validez de las inferencias y predicciones derivadas del modelo.

## Investigación de Residuos

1. Los **residuos de Pearson** verifican si la varianza está correctamente especificada. Estos deben estar centrados en 0 y tener una varianza aproximada de 1.
2. Los **residuos de Deviance** verifican si la función de enlace g() es apropiada. Si hay valores grandes en los residuos de deviance, podría ser que el enlace no sea correcto.

```

# Función para crear gráficos de residuos vs predictor lineal
create_residual_vs_linear_predictor <- function(model, model_name) {
  # Residuos de Pearson
  pearson_residuals <- residuals(model, type = "pearson")
  linear_predictor <- predict(model, type = "link") # Predictor lineal

  scatter_pearson <- ggplot() +
    aes(x = linear_predictor, y = pearson_residuals) +
    geom_point(alpha = 0.5) +
    geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
    labs(title = paste("Residuos de Pearson -", model_name),
         x = "Predictor Lineal (\u03B7)", 
         y = "Residuos de Pearson") +
    theme_minimal()

  # Residuos de Deviance
  deviance_residuals <- residuals(model, type = "deviance")

  scatter_deviance <- ggplot() +
    aes(x = linear_predictor, y = deviance_residuals) +
    geom_point(alpha = 0.5) +
    labs(title = paste("Residuos de Deviance -", model_name),

```

```

x = "Predictor Lineal (\u03B7)",
y = "Residuos de Deviance") +
theme_minimal()

return(list(scatter_pearson, scatter_deviance))
}

# Crear las gráficas para cada modelo
plots_log <- create_residual_vs_linear_predictor(glm_log, "Log")
plots_inverse <- create_residual_vs_linear_predictor(glm_inverse, "Inverse")
plots_log_red <- create_residual_vs_linear_predictor(glm_log.red, "Log Reducido")
plots_inverse_red <- create_residual_vs_linear_predictor(glm_inverse.red, "Inverse Reducido")

# Combinar todas las gráficas en una sola lista
all_plots <- c(
  plots_log,
  plots_inverse,
  plots_log_red,
  plots_inverse_red
)

# Crear una salida de dimensiones más grandes para que las gráficas no se vean aplastadas
ggsave(
  filename = "residual_plots_grid.png", # Guardar como imagen
  plot = grid.arrange(
    grobs = all_plots,
    nrow = 4,
    ncol = 2,
    top = "Comparación de Residuos para los Modelos",
    heights = unit(c(1, 1, 1, 1), "null"), # Ajustar proporciones para filas
    widths = unit(c(1, 1), "null") # Ajustar proporciones para columnas
  ),
  width = 12, # Ancho total en pulgadas
  height = 16 # Alto total en pulgadas
)

```

En nuestra investigación de residuos (Pearson y Deviance), encontramos que, en general, los residuos se comportan como se espera, mostrando una distribución aproximadamente centrada en cero con varianza constante, lo cual es consistente con los supuestos del modelo. Sin embargo, identificamos un pico inusual en los residuos hacia la mitad de los datos, lo que podría indicar la presencia de patrones no explicados completamente por el modelo. Resolver este problema requeriría un análisis más profundo y complejo con técnicas y conceptos que quedan fuera del alcance de este curso. Por lo tanto, procederemos con los modelos actuales, conscientes de esta limitación.

## Valores Atípicos

En esta sección, nos enfocaremos en la detección de valores atípicos (*outliers*) que podrían estar influyendo de manera significativa en los resultados de nuestros modelos. Identificar y manejar estos valores es crucial para garantizar la robustez y confiabilidad de las estimaciones.

Para ello, utilizaremos tres representaciones (gráficas y estadísticas) clave:

1. **Leverage:** Nos permite identificar observaciones con un alto impacto en la estimación de los coeficientes debido a su posición en el espacio de las variables predictoras.
2. **Distancia de Cook:** Evalúa la influencia combinada de leverage y residuos en la calidad del ajuste del

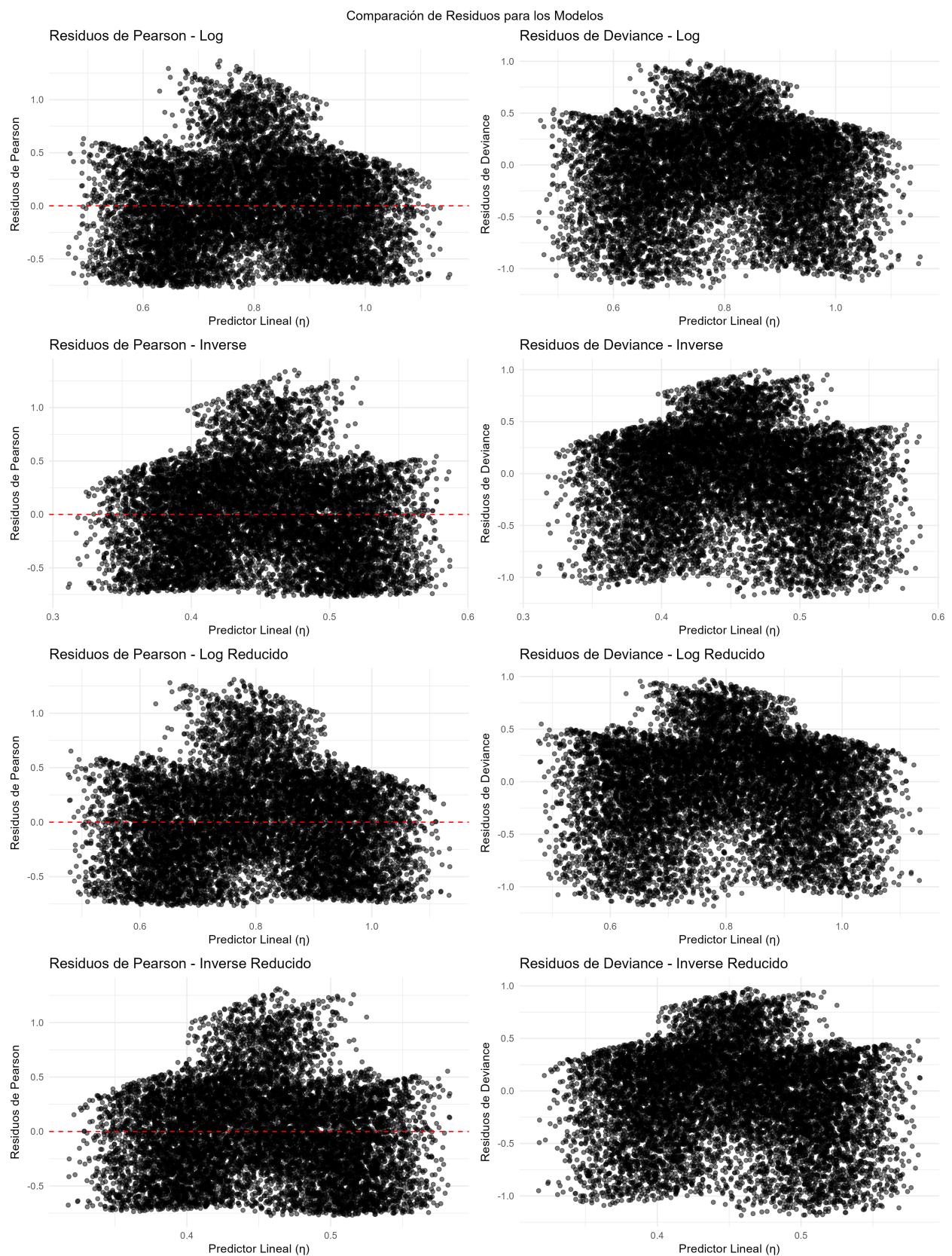


Figure 1: Comparación de Residuos

modelo. Valores altos indican observaciones con un impacto significativo en las predicciones.

3. **Gráfico Leverage vs Distancia de Cook:** Combina ambas métricas para proporcionar una visión completa de las observaciones influyentes.

Con base en estas visualizaciones, identificaremos las observaciones que representen valores atípicos y procederemos a eliminarlas del conjunto de datos para asegurar un análisis más preciso y representativo. Este proceso nos permitirá ajustar modelos que reflejen de manera más fiel las tendencias y patrones en los datos.

Calculemos los outliers

```
# Función para calcular leverage y distancia de Cook
detect_outliers <- function(model, model_name) {
  leverage <- hatvalues(model)
  cook_distance <- cooks.distance(model)

  # Umbrales
  threshold_leverage <- 2 * mean(leverage) # Umbral para leverage
  threshold_cook <- 1 # Umbral fijo para Cook's distance

  # Identificar outliers
  leverage_outliers <- which(leverage > threshold_leverage)
  cook_outliers <- which(cook_distance > threshold_cook)

  # Combinar resultados
  all_outliers <- unique(c(leverage_outliers, cook_outliers))

  # Manejo de casos donde no hay outliers
  if (length(leverage_outliers) == 0) {
    leverage_outliers <- "Ninguno"
  }
  if (length(cook_outliers) == 0) {
    cook_outliers <- "Ninguno"
  }
  if (length(all_outliers) == 0) {
    all_outliers <- "Ninguno"
  }

  # Total de outliers
  total_leverage <- if (is.numeric(leverage_outliers)) length(leverage_outliers) else 0
  total_cook <- if (is.numeric(cook_outliers)) length(cook_outliers) else 0
  total_all <- if (is.numeric(all_outliers)) length(all_outliers) else 0

  # Resultados
  list(
    modelo = model_name,
    threshold_leverage = threshold_leverage,
    threshold_cook = threshold_cook,
    leverage_outliers = leverage_outliers,
    cook_outliers = cook_outliers,
    all_outliers = all_outliers,
    total_leverage = total_leverage,
    total_cook = total_cook,
    total_all = total_all
  )
}
```

```

# Aplicar la función a los 4 modelos
outliers_log <- detect_outliers(glm_log, "Log")
outliers_inverse <- detect_outliers(glm_inverse, "Inverse")
outliers_log_red <- detect_outliers(glm_log.red, "Log Reducido")
outliers_inverse_red <- detect_outliers(glm_inverse.red, "Inverse Reducido")

# Mostrar resultados de cada modelo
resultados <- list(outliers_log, outliers_inverse, outliers_log_red, outliers_inverse_red)

for (res in resultados) {
  cat("\n--- Modelo:", res$modelo, "---\n")
  cat("Umbral de Leverage:", res$threshold_leverage, "\n")
  cat("Umbral de Cook:", res$threshold_cook, "\n")
  cat("Total de Outliers por Leverage:", res$total_leverage, "\n")
  cat("Total de Outliers por Cook:", res$total_cook, "\n")
  cat("Total de Outliers combinados:", res$total_all, "\n")
}

## --- Modelo: Log ---
## Umbral de Leverage: 0.0052
## Umbral de Cook: 1
## Total de Outliers por Leverage: 0
## Total de Outliers por Cook: 0
## Total de Outliers combinados: 0
##
## --- Modelo: Inverse ---
## Umbral de Leverage: 0.0052
## Umbral de Cook: 1
## Total de Outliers por Leverage: 55
## Total de Outliers por Cook: 0
## Total de Outliers combinados: 55
##
## --- Modelo: Log Reducido ---
## Umbral de Leverage: 0.0026
## Umbral de Cook: 1
## Total de Outliers por Leverage: 0
## Total de Outliers por Cook: 0
## Total de Outliers combinados: 0
##
## --- Modelo: Inverse Reducido ---
## Umbral de Leverage: 0.0026
## Umbral de Cook: 1
## Total de Outliers por Leverage: 43
## Total de Outliers por Cook: 0
## Total de Outliers combinados: 43

# Obtener el total de índices únicos de outliers combinados
unique_outliers <- unique(c(
  if (is.numeric(outliers_log$all_outliers)) outliers_log$all_outliers else numeric(0),
  if (is.numeric(outliers_inverse$all_outliers)) outliers_inverse$all_outliers else numeric(0),
  if (is.numeric(outliers_log_red$all_outliers)) outliers_log_red$all_outliers else numeric(0),
  if (is.numeric(outliers_inverse_red$all_outliers)) outliers_inverse_red$all_outliers else numeric(0)
))

```

```

# Imprimir el total de índices únicos de outliers
cat("\n--- Resumen General ---\n")

## 
## --- Resumen General ---

cat("Total de índices únicos de outliers combinados:", length(unique_outliers), "\n")

## Total de índices únicos de outliers combinados: 80

Grafiquemos Leverage

# Función para graficar leverage
plot_leverage <- function(model, model_name) {
  # Calcular leverage
  leverage <- hatvalues(model)

  # Umbral de leverage
  threshold_leverage <- 2 * mean(leverage)

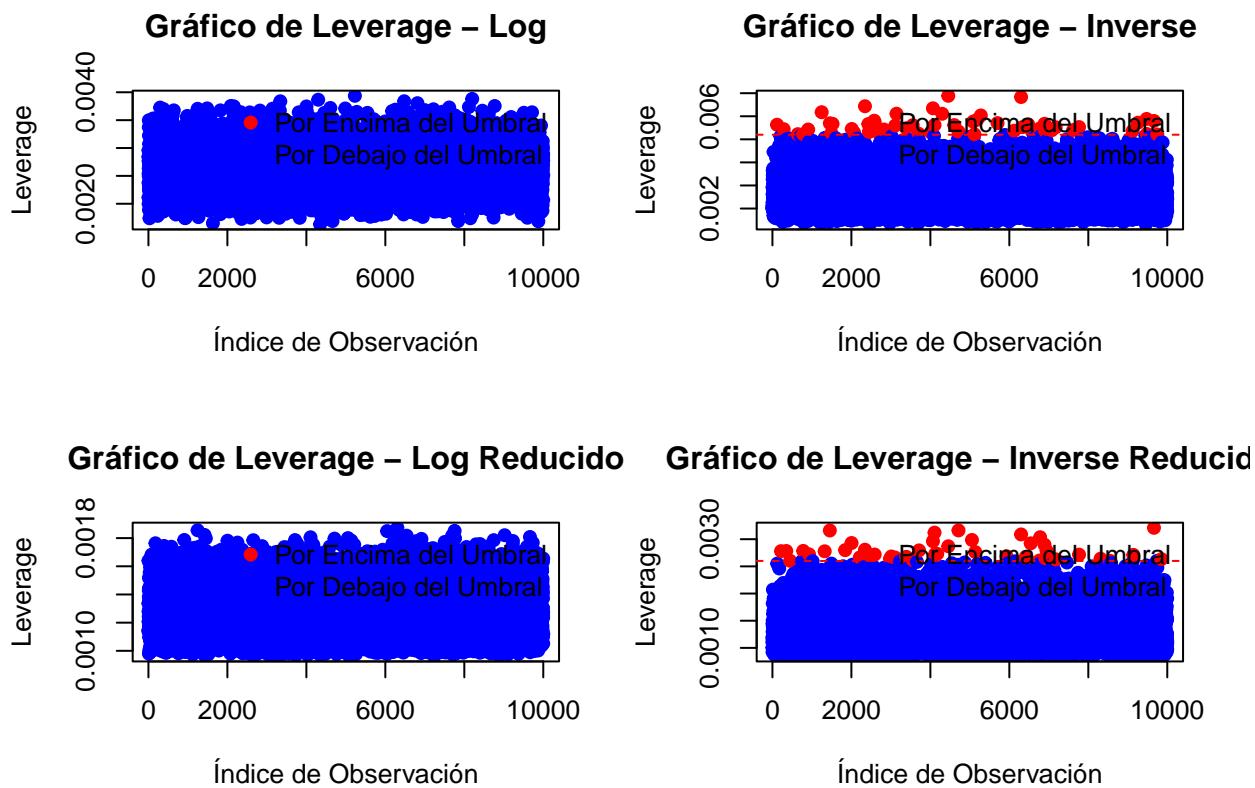
  # Crear el gráfico
  plot(
    leverage,
    main = paste("Gráfico de Leverage -", model_name),
    xlab = "Índice de Observación",
    ylab = "Leverage",
    pch = 19, col = ifelse(leverage > threshold_leverage, "red", "blue")
  )

  # Añadir línea del umbral
  abline(h = threshold_leverage, col = "red", lty = 2)

  # Añadir leyenda
  legend(
    "topright",
    legend = c("Por Encima del Umbral", "Por Debajo del Umbral"),
    col = c("red", "blue"),
    pch = 19,
    bty = "n"
  )
}

# Crear las gráficas para los 4 modelos
par(mfrow = c(2, 2)) # Configurar una cuadrícula de 2x2
plot_leverage(glm_log, "Log")
plot_leverage(glm_inverse, "Inverse")
plot_leverage(glm_log.red, "Log Reducido")
plot_leverage(glm_inverse.red, "Inverse Reducido")

```



Grafiquemos Distancia de Cook

```
# Función para graficar distancia de Cook
plot_cooks_distance <- function(model, model_name) {
  # Calcular distancia de Cook
  cook_distance <- cooks.distance(model)

  # Umbral de Cook
  threshold_cook <- 1 # Umbral fijo

  # Crear el gráfico
  plot(
    cook_distance,
    main = paste("Gráfico de Distancia de Cook - ", model_name),
    xlab = "Índice de Observación",
    ylab = "Distancia de Cook",
    pch = 19, col = ifelse(cook_distance > threshold_cook, "red", "blue")
  )

  # Añadir línea del umbral
  abline(h = threshold_cook, col = "red", lty = 2)

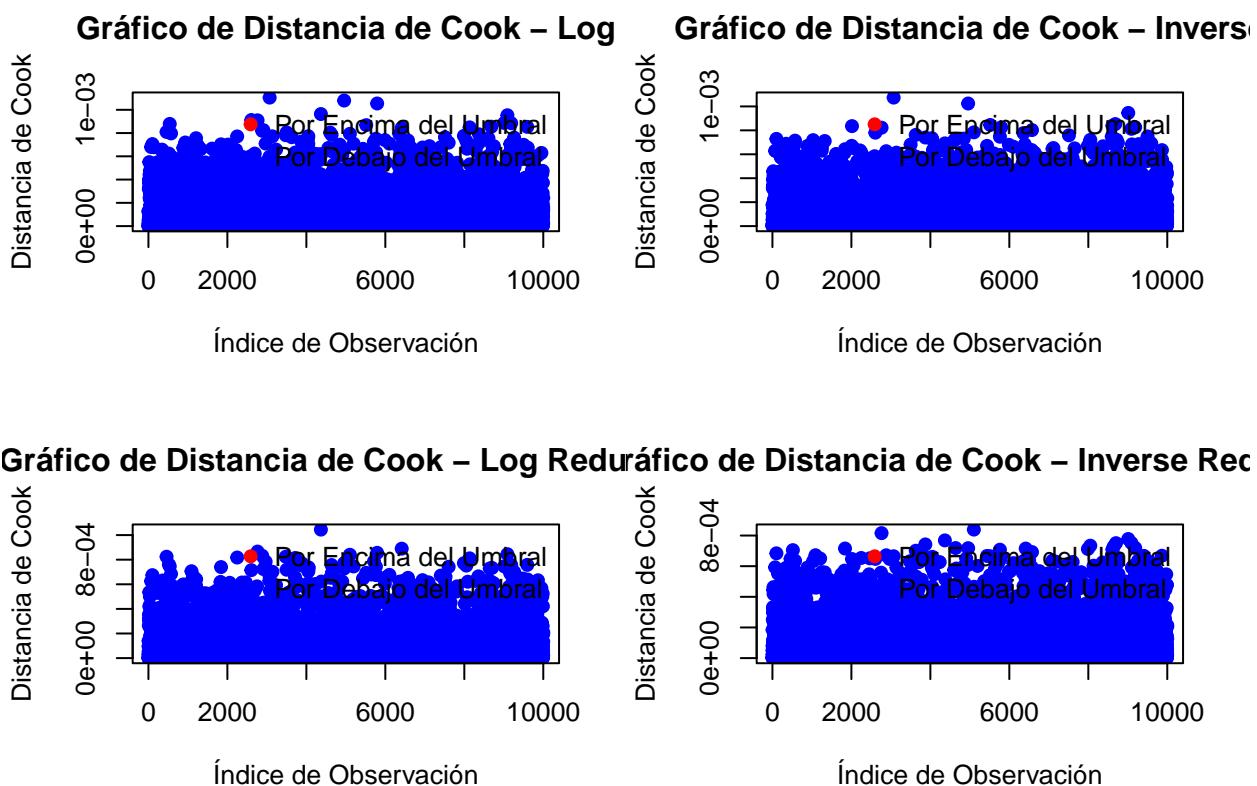
  # Añadir leyenda
  legend(
    "topright",
    legend = c("Por Encima del Umbral", "Por Debajo del Umbral"),
    col = c("red", "blue"),
    bty = "n"
  )
}
```

```

    pch = 19,
    bty = "n"
  }

# Crear las gráficas para los 4 modelos
par(mfrow = c(2, 2)) # Configurar una cuadricula de 2x2
plot_cooks_distance(glm_log, "Log")
plot_cooks_distance(glm_inverse, "Inverse")
plot_cooks_distance(glm_log.red, "Log Reducido")
plot_cooks_distance(glm_inverse.red, "Inverse Reducido")

```



Ahora graficaremos Leverage vs. Distancia de Cook

```

# Función para graficar leverage vs Cook's distance
plot_leverage_cook <- function(model, model_name) {
  leverage <- hatvalues(model) # Calcular leverage
  cook_distance <- cooks.distance(model) # Calcular distancia de Cook

  # Umbral
  threshold_leverage <- 2 * mean(leverage)
  threshold_cook <- 1

  # Crear el gráfico
  plot(
    leverage, cook_distance,

```

```

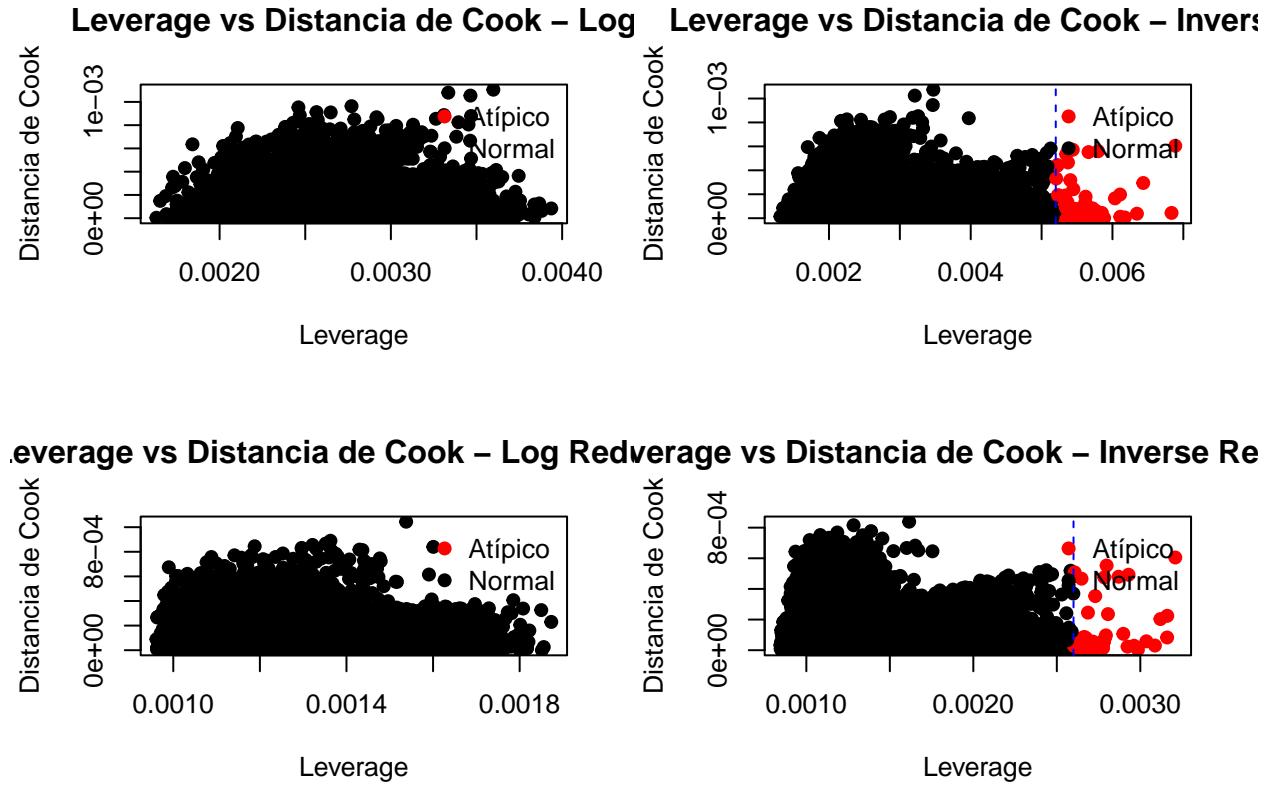
main = paste("Leverage vs Distancia de Cook - ", model_name),
xlab = "Leverage",
ylab = "Distancia de Cook",
pch = 19, col = ifelse(leverage > threshold_leverage | cook_distance > threshold_cook, "red", "black")
)

# Añadir líneas de umbral
abline(v = threshold_leverage, col = "blue", lty = 2)
abline(h = threshold_cook, col = "blue", lty = 2)

# Leyenda
legend(
  "topright", legend = c("Atípico", "Normal"),
  col = c("red", "black"), pch = 19, bty = "n"
)
}

# Crear las gráficas para los 4 modelos
par(mfrow = c(2, 2)) # Configurar una cuadrícula de 2x2 para las gráficas
plot_leverage_cook(glm_log, "Log")
plot_leverage_cook(glm_inverse, "Inverse")
plot_leverage_cook(glm_log.red, "Log Reducido")
plot_leverage_cook(glm_inverse.red, "Inverse Reducido")

```



En nuestra base de datos de 10,000 observaciones, identificamos 80 valores atípicos a través de los cuatro modelos. Estos outliers representan el 0.8% del total de los datos. Dado que pueden afectar al ajuste de

nuestro modelo de manera desproporcionada y, considerando su bajo porcentaje, procedemos a eliminarlos.

```
# Función para eliminar outliers del dataset
remove_outliers <- function(data, outliers_indices) {
  # Verificar si los índices de outliers son válidos
  if (length(outliers_indices) == 0) {
    return(data) # Devolver el dataset original si no hay outliers
  }
  # Excluir las filas identificadas como outliers
  data_clean <- data[-outliers_indices, ]
  return(data_clean)
}

# Verificar y combinar los índices de outliers
outliers_combined <- unique(c(
  if (is.numeric(outliers_log$log$all_outliers)) outliers_log$log$all_outliers else numeric(0),
  if (is.numeric(outliers_inverse$inverse$all_outliers)) outliers_inverse$inverse$all_outliers else numeric(0),
  if (is.numeric(outliers_log_red$log$all_outliers)) outliers_log_red$log$all_outliers else numeric(0),
  if (is.numeric(outliers_inverse_red$inverse$all_outliers)) outliers_inverse_red$inverse$all_outliers else numeric(0)))
))

# Crear un nuevo dataset sin outliers
climate_data_clean <- remove_outliers(climate_data, outliers_combined)

# Mostrar el número de observaciones antes y después
cat("Número de observaciones antes de eliminar outliers:", nrow(climate_data), "\n")

## Número de observaciones antes de eliminar outliers: 10000
cat("Número de observaciones después de eliminar outliers:", nrow(climate_data_clean), "\n")

## Número de observaciones después de eliminar outliers: 9920
cat("Total de outliers eliminados:", length(outliers_combined), "\n")

## Total de outliers eliminados: 80

Volvemos a ajustar con datasets sin outliers

# Ajustamos
glm_log_clean <- glm(
  Crop_Yield_MT_per_HA ~ .,
  data = climate_data_clean,
  family = Gamma(link = "log")
)

glm_inverse_clean <- glm(
  Crop_Yield_MT_per_HA ~ .,
  data = climate_data_clean,
  family = Gamma(link = "inverse")
)

glm_log_red_clean <- glm(
  Crop_Yield_MT_per_HA ~ Average_Temperature_C + CO2_Emissions_MT + Total_Precipitation_mm + Crop_Type,
  data = climate_data_clean,
  family = Gamma(link = "log")
)
```

```

glm_inverse_red_clean <- glm(
  Crop_Yield_MT_per_HA ~ Average_Temperature_C + CO2_Emissions_MT + Total_Precipitation_mm + Crop_Type,
  data = climate_data_clean,
  family = Gamma(link = "inverse")
)

```

## Comparación de modelos

En esta sección, evaluaremos y compararemos los cuatro modelos ajustados utilizando diversas métricas de desempeño y técnicas de contraste. Para identificar el modelo que mejor se ajusta a nuestros datos, analizaremos el **AIC** (Criterio de Información de Akaike) y el **BIC** (Criterio de Información Bayesiano), que penalizan la complejidad del modelo en favor de su simplicidad. También incluiremos la **pseudo  $R^2$**  de McFadden como una medida relativa de la calidad del ajuste, proporcionando una visión general de la capacidad explicativa de cada modelo.

Adicionalmente, compararemos los modelos reducidos con sus modelos padres a través de dos pruebas estadísticas para modelos anidados: la **Prueba de Razón de Verosimilitudes (LRT)** y el **Deviance Test**, que nos permitirán evaluar si los modelos reducidos son significativamente peores que sus contrapartes completas o si la reducción es válida sin pérdida sustancial de información. Este análisis integral nos ayudará a seleccionar el modelo más adecuado desde una perspectiva tanto estadística como práctica.

### Comparaciones con AIC, BIC y pseudo $R^2$

```

# Función para calcular la pseudo R^2 (McFadden's R^2)
calculate_pseudo_r2 <- function(model) {
  null_deviance <- model$null.deviance
  residual_deviance <- model$deviance
  pseudo_r2 <- 1 - (residual_deviance / null_deviance)
  return(pseudo_r2)
}

# Crear una lista con los modelos refitteados
models <- list(
  Log = glm_log_clean,
  Inverse = glm_inverse_clean,
  Log_Reducido = glm_log_red_clean,
  Inverse_Reducido = glm_inverse_red_clean
)

# Calcular AIC, BIC y pseudo R^2 para cada modelo
comparison <- data.frame(
  Modelo = names(models),
  AIC = sapply(models, AIC),
  BIC = sapply(models, BIC),
  Pseudo_R2 = sapply(models, calculate_pseudo_r2)
)

# Ordenar por AIC
comparison <- comparison[order(comparison$AIC), ]

# Mostrar la tabla comparativa
print(comparison)

```

##	Modelo	AIC	BIC	Pseudo_R2
----	--------	-----	-----	-----------

```

## Log_Reducido      Log_Reducido 26824.27 26925.10 0.07888189
## Log                  Log 26840.35 27034.81 0.07977137
## Inverse_Reducido Inverse_Reducido 26922.17 27023.00 0.07005929
## Inverse            Inverse 26939.34 27133.80 0.07085810

```

El modelo Log Reducido tiene el menor AIC (26824.27), lo que sugiere que es el modelo que mejor predice sin aumentar demasiado su complejidad. El modelo Log Reducido tiene el menor BIC (26925.10), lo que lo hace el preferible si lo que se busca es un modelo más general que preciso. El modelo Log tiene el mayor pseudo  $R^2$  (0.0798), lo que significa que explica ligeramente más variabilidad en los datos que los otros modelos.

### Comparaciones anidadas

```

# Función para realizar las pruebas y formatear resultados en texto
nested_model_tests_text <- function(full_model, reduced_model, model_name) {
  # Prueba de Razón de Verosimilitudes (LRT)
  lrt <- anova(reduced_model, full_model, test = "LRT")

  # Deviance Test
  deviance_diff <- reduced_model$deviance - full_model$deviance
  df_diff <- reduced_model$df.residual - full_model$df.residual
  p_value_deviance <- pchisq(deviance_diff, df = abs(df_diff), lower.tail = FALSE)

  # Formatear y mostrar resultados
  cat("\n\n\n--- Comparación de Modelos (", model_name, ") ---\n", sep = "")

  cat("\nPrueba de Razón de Verosimilitudes (LRT):\n")
  cat("  - Grados de libertad residuales (reducido): ", reduced_model$df.residual, "\n")
  cat("  - Grados de libertad residuales (completo): ", full_model$df.residual, "\n")
  cat("  - Deviancia residual (reducido): ", round(reduced_model$deviance, 2), "\n")
  cat("  - Deviancia residual (completo): ", round(full_model$deviance, 2), "\n")
  cat("  - Diferencia de chi-cuadrado: ", round(lrt[2, "Deviance"], 2), "\n")
  cat("  - P-valor: ", format.pval(lrt[2, "Pr(>Chi)"]), "\n")

  cat("\nPrueba de Deviancia:\n")
  cat("  - Diferencia de deviancia: ", round(deviance_diff, 2), "\n")
  cat("  - Diferencia de grados de libertad: ", abs(df_diff), "\n")
  cat("  - P-valor: ", format.pval(p_value_deviance), "\n")
}

# Comparar modelos completos vs reducidos
nested_model_tests_text(glm_log_clean, glm_log_red_clean, "Modelo Log")

##
##
##
## --- Comparación de Modelos (Modelo Log) ---
##
## Prueba de Razón de Verosimilitudes (LRT):
##   - Grados de libertad residuales (reducido): 9907
##   - Grados de libertad residuales (completo): 9894
##   - Deviancia residual (reducido): 2098.33
##   - Deviancia residual (completo): 2096.31
##   - Diferencia de chi-cuadrado: 2.03
##   - P-valor: 0.62072
##

```

```

## Prueba de Deviancia:
##   - Diferencia de deviancia: 2.03
##   - Diferencia de grados de libertad: 13
##   - P-valor: 0.99976
nested_model_tests_text(glm_inverse_clean, glm_inverse_red_clean, "Modelo Inverse")

##
##
##
## --- Comparación de Modelos (Modelo Inverse) ---
##
## Prueba de Razón de Verosimilitudes (LRT):
##   - Grados de libertad residuales (reducido): 9907
##   - Grados de libertad residuales (completo): 9894
##   - Deviancia residual (reducido): 2118.43
##   - Deviancia residual (completo): 2116.61
##   - Diferencia de chi-cuadrado: 1.82
##   - P-valor: 0.72018
##
## Prueba de Deviancia:
##   - Diferencia de deviancia: 1.82
##   - Diferencia de grados de libertad: 13
##   - P-valor: 0.99987

```

Los resultados de las pruebas de comparación entre los modelos completos y reducidos muestran valores  $p$  altos ( $p > 0.05$ ) en ambas pruebas, indicando que no hay evidencia estadística suficiente para rechazar la hipótesis nula. Esto sugiere que los modelos reducidos ofrecen un ajuste comparable al de los modelos completos, por lo que podemos utilizarlos sin pérdida significativa de información.

### Cross validation

En esta sección, realizamos un análisis de validación cruzada para evaluar el desempeño predictivo de los modelos ajustados. Utilizamos un esquema de **10 folds** repetido **100 veces** para obtener una medida confiable de la deviancia promedio. La *deviancia total* se calcula como una métrica que evalúa qué tan bien cada modelo ajusta los datos observados, penalizando las discrepancias entre los valores predichos y los reales.

```

# Definir la función de deviance gamma
gamma_deviance <- function(y, mu) {
  return(2 * sum(((y - mu) / mu) - log(y / mu)))
}

# Realizar validación cruzada usando cv.glm
perform_cv_glm <- function(model, data, k_folds = 10) {
  set.seed(456) # Semilla para reproducibilidad

  # Validación cruzada con cv.glm
  cv_results <- cv.glm(
    data = data,
    glmfit = model,
    cost = gamma_deviance, # Utilizamos la deviance gamma como métrica
    K = k_folds
  )

  # Retornar el costo (Deviancia)
  return(cv_results$delta[1]) # delta[1] es el error
}

```

```

}

# Aplicar validación cruzada a cada modelo
cv_deviance_results <- data.frame(
  Modelo = c("Log", "Log Reducido", "Inverse", "Inverse Reducido"),
  Deviance = c(
    perform_cv_glm(glm_log_clean, climate_data_clean),
    perform_cv_glm(glm_log_red_clean, climate_data_clean),
    perform_cv_glm(glm_inverse_clean, climate_data_clean),
    perform_cv_glm(glm_inverse_red_clean, climate_data_clean)
  )
)

# Mostrar resultados
print(cv_deviance_results)

##           Modelo Deviance
## 1          Log 210.8267
## 2      Log Reducido 210.3297
## 3        Inverse 212.8565
## 4 Inverse Reducido 212.3263

```

El modelo Log Reducido tiene, en promedio, un Deviance menor al evaluar el modelo, lo que indica que es el modelo que ofrece el mejor ajuste para nuestros datos. Los modelos con enlace inverso presentan un peor desempeño (mayor deviance). Los modelos reducidos presentan un mejor desempeño que sus contrapartes (modelos completos).

## Pruebas sobre coeficientes

Primero, evaluaremos la significancia estadística de los coeficientes estimados en los modelos ajustados. Esto nos permitirá identificar cuáles variables tienen un impacto significativo en la variable de respuesta y cuáles podrían considerarse irrelevantes. Para ello, utilizaremos pruebas z (asintóticamente) expuestas en el summary del fit, donde la hipótesis nula establece que el coeficiente es igual a cero ( $H_0 : \beta_i = 0$ ). Este análisis nos ayudará a comprender mejor las relaciones entre las variables predictoras y la productividad agrícola, y a justificar la inclusión o exclusión de variables en los modelos reducidos.

```

cat("\n--- Summary de modelo con liga log ---\n", sep = "")

##
## --- Summary de modelo con liga log ---
summary(glm_log_clean)

##
## Call:
## glm(formula = Crop_Yield_MT_per_HA ~ ., family = Gamma(link = "log"),
##      data = climate_data_clean)
##
## Coefficients:
##                               Estimate Std. Error t value
## (Intercept)                0.8672494  0.0209744 41.348
## Crop_TypeCoffee             -0.0461198  0.0199852 -2.308
## Crop_TypeCorn               -0.0414572  0.0195696 -2.118
## Crop_TypeCotton              0.0646688  0.0194701 -3.321
## Crop_TypeFruits             -0.0125376  0.0198452 -0.632
## Crop_TypeRice              -0.0290877  0.0195784 -1.486

```

```

## Crop_TypeSoybeans           -0.0393508  0.0199229 -1.975
## Crop_TypeSugarcane          -0.0214926  0.0197431 -1.089
## Crop_TypeVegetables         -0.0507270  0.0195052 -2.601
## Crop_TypeWheat              -0.0295711  0.0194699 -1.519
## Average_Temperature_C       0.1370538  0.0043664 31.388
## Total_Precipitation_mm      0.0123986  0.0043509  2.850
## CO2_Emissions_MT            -0.0407877  0.0043548 -9.366
## Extreme_Weather_Events      0.0003541  0.0043414  0.082
## Irrigation_Access_.         0.0023964  0.0043407  0.552
## Pesticide_Use_KG_per_HA     -0.0037725  0.0043448 -0.868
## Fertilizer_Use_KG_per_HA    0.0045724  0.0043444  1.052
## Soil_Health_Index            -0.0013456  0.0043416 -0.310
## Adaptation_StrategiesDrought-resistant Crops -0.0081595  0.0138107 -0.591
## Adaptation_StrategiesNo Adaptation        -0.0056041  0.0137518 -0.408
## Adaptation_StrategiesOrganic Farming       -0.0094332  0.0138350 -0.682
## Adaptation_StrategiesWater Management     -0.0220434  0.0137028 -1.609
## ContinentAmericas           -0.0296991  0.0152934 -1.942
## ContinentAsia                -0.0187319  0.0166685 -1.124
## ContinentEurope              -0.0373884  0.0168270 -2.222
## ContinentOceania             -0.0257225  0.0192388 -1.337
## Pr(>|t|)
## (Intercept)                  < 2e-16 ***
## Crop_TypeCoffee               0.021036 *
## Crop_TypeCorn                 0.034162 *
## Crop_TypeCotton               0.000899 ***
## Crop_TypeFruits               0.527553
## Crop_TypeRice                 0.137389
## Crop_TypeSoybeans              0.048278 *
## Crop_TypeSugarcane             0.276351
## Crop_TypeVegetables            0.009318 **
## Crop_TypeWheat                 0.128843
## Average_Temperature_C         < 2e-16 ***
## Total_Precipitation_mm        0.004385 **
## CO2_Emissions_MT              < 2e-16 ***
## Extreme_Weather_Events        0.935001
## Irrigation_Access_.           0.580901
## Pesticide_Use_KG_per_HA       0.385265
## Fertilizer_Use_KG_per_HA      0.292606
## Soil_Health_Index              0.756625
## Adaptation_StrategiesDrought-resistant Crops 0.554662
## Adaptation_StrategiesNo Adaptation 0.683639
## Adaptation_StrategiesOrganic Farming 0.495360
## Adaptation_StrategiesWater Management 0.107720
## ContinentAmericas             0.052171 .
## ContinentAsia                  0.261130
## ContinentEurope                0.026311 *
## ContinentOceania               0.181251
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.1862076)
##
## Null deviance: 2278.0  on 9919  degrees of freedom
## Residual deviance: 2096.3  on 9894  degrees of freedom

```

```

## AIC: 26840
##
## Number of Fisher Scoring iterations: 4
cat("\n\n--- Summary de modelo con liga inv ---\n", sep = "")

##
##
## --- Summary de modelo con liga inv ---
summary(glm_inverse_clean)

##
## Call:
## glm(formula = Crop_Yield_MT_per_HA ~ ., family = Gamma(link = "inverse"),
##      data = climate_data_clean)
##
## Coefficients:
##                               Estimate Std. Error t value
## (Intercept)                4.233e-01  9.178e-03 46.125
## Crop_TypeCoffee             1.985e-02  8.844e-03  2.244
## Crop_TypeCorn               1.949e-02  8.650e-03  2.253
## Crop_TypeCotton              2.803e-02  8.703e-03  3.220
## Crop_TypeFruits              6.967e-03  8.651e-03  0.805
## Crop_TypeRice                1.272e-02  8.600e-03  1.479
## Crop_TypeSoybeans             1.776e-02  8.807e-03  2.017
## Crop_TypeSugarcane            7.571e-03  8.644e-03  0.876
## Crop_TypeVegetables            2.266e-02  8.660e-03  2.616
## Crop_TypeWheat                 1.269e-02  8.520e-03  1.490
## Average_Temperature_C        -5.365e-02  1.969e-03 -27.255
## Total_Precipitation_mm       -5.677e-03  1.943e-03 -2.921
## CO2_Emissions_MT                1.807e-02  1.944e-03  9.298
## Extreme_Weather_Events          5.055e-05  1.934e-03  0.026
## Irrigation_Access_.           -7.080e-04  1.937e-03 -0.366
## Pesticide_Use_KG_per_HA         1.207e-03  1.933e-03  0.624
## Fertilizer_Use_KG_per_HA        -2.026e-03  1.934e-03 -1.048
## Soil_Health_Index                  4.314e-04  1.932e-03  0.223
## Adaptation_StrategiesDrought-resistant Crops 4.086e-03  6.102e-03  0.670
## Adaptation_StrategiesNo Adaptation 2.611e-03  6.082e-03  0.429
## Adaptation_StrategiesOrganic Farming 4.450e-03  6.120e-03  0.727
## Adaptation_StrategiesWater Management 9.677e-03  6.101e-03  1.586
## ContinentAmericas                1.245e-02  6.725e-03  1.852
## ContinentAsia                     7.544e-03  7.325e-03  1.030
## ContinentEurope                   1.593e-02  7.447e-03  2.139
## ContinentOceania                  1.079e-02  8.521e-03  1.266
##
## (Intercept) < 2e-16 ***
## Crop_TypeCoffee 0.02484 *
## Crop_TypeCorn 0.02426 *
## Crop_TypeCotton 0.00128 **
## Crop_TypeFruits 0.42059
## Crop_TypeRice 0.13917
## Crop_TypeSoybeans 0.04375 *
## Crop_TypeSugarcane 0.38110
## Crop_TypeVegetables 0.00890 **

```

```

## Crop_TypeWheat          0.13634
## Average_Temperature_C < 2e-16 ***
## Total_Precipitation_mm 0.00349 **
## CO2_Emissions_MT       < 2e-16 ***
## Extreme_Weather_Events 0.97914
## Irrigation_Access_.    0.71468
## Pesticide_Use_KG_per_HA 0.53237
## Fertilizer_Use_KG_per_HA 0.29488
## Soil_Health_Index       0.82332
## Adaptation_StrategiesDrought-resistant Crops 0.50313
## Adaptation_StrategiesNo Adaptation 0.66768
## Adaptation_StrategiesOrganic Farming 0.46717
## Adaptation_StrategiesWater Management 0.11279
## ContinentAmericas      0.06407 .
## ContinentAsia           0.30309
## ContinentEurope          0.03249 *
## ContinentOceania         0.20554
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.1880606)
##
## Null deviance: 2278.0  on 9919  degrees of freedom
## Residual deviance: 2116.6  on 9894  degrees of freedom
## AIC: 26939
##
## Number of Fisher Scoring iterations: 5
cat("\n\n--- Summary de modelo reducido con liga log ---\n", sep = "")

##
##
## --- Summary de modelo reducido con liga log ---
summary(glm_log_red_clean)

##
## Call:
## glm(formula = Crop_Yield_MT_per_HA ~ Average_Temperature_C +
##       CO2_Emissions_MT + Total_Precipitation_mm + Crop_Type, family = Gamma(link = "log"),
##       data = climate_data_clean)
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.832242  0.014137 58.869 < 2e-16 ***
## Average_Temperature_C 0.136689  0.004361 31.346 < 2e-16 ***
## CO2_Emissions_MT     -0.041106  0.004351 -9.447 < 2e-16 ***
## Total_Precipitation_mm 0.012214  0.004345  2.811 0.00495 **
## Crop_TypeCoffee        -0.045873  0.019968 -2.297 0.02162 *
## Crop_TypeCorn           -0.040751  0.019554 -2.084 0.03718 *
## Crop_TypeCotton          -0.063884  0.019452 -3.284 0.00103 **
## Crop_TypeFruits          -0.011871  0.019824 -0.599 0.54929
## Crop_TypeRice            -0.028593  0.019565 -1.461 0.14393
## Crop_TypeSoybeans         -0.038611  0.019908 -1.939 0.05247 .
## Crop_TypeSugarcane        -0.020581  0.019721 -1.044 0.29669

```

```

## Crop_TypeVegetables      -0.051170   0.019485  -2.626  0.00865 **
## Crop_TypeWheat           -0.029446   0.019457  -1.513  0.13020
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.1861042)
##
## Null deviance: 2278.0  on 9919  degrees of freedom
## Residual deviance: 2098.3  on 9907  degrees of freedom
## AIC: 26824
##
## Number of Fisher Scoring iterations: 4
cat("\n\n--- Summary de modelo reducido con liga inv ---\n", sep = "")

##
##
## --- Summary de modelo reducido con liga inv ---
summary(glm_inverse_red_clean)

##
## Call:
## glm(formula = Crop_Yield_MT_per_HA ~ Average_Temperature_C +
##       CO2_Emissions_MT + Total_Precipitation_mm + Crop_Type, family = Gamma(link = "inverse"),
##       data = climate_data_clean)
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.438342  0.006161 71.151 < 2e-16 ***
## Average_Temperature_C -0.053556  0.001966 -27.239 < 2e-16 ***
## CO2_Emissions_MT       0.018206  0.001942  9.376 < 2e-16 ***
## Total_Precipitation_mm -0.005620  0.001940 -2.897 0.00378 **
## Crop_TypeCoffee         0.019709  0.008836  2.231 0.02573 *
## Crop_TypeCorn           0.019115  0.008641  2.212 0.02698 *
## Crop_TypeCotton          0.027751  0.008694  3.192 0.00142 **
## Crop_TypeFruits          0.006817  0.008641  0.789 0.43018
## Crop_TypeRice            0.012475  0.008594  1.452 0.14665
## Crop_TypeSoybeans         0.017507  0.008799  1.990 0.04667 *
## Crop_TypeSugarcane        0.007238  0.008634  0.838 0.40192
## Crop_TypeVegetables       0.022737  0.008651  2.628 0.00859 **
## Crop_TypeWheat            0.012613  0.008514  1.482 0.13850
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.1879511)
##
## Null deviance: 2278.0  on 9919  degrees of freedom
## Residual deviance: 2118.4  on 9907  degrees of freedom
## AIC: 26922
##
## Number of Fisher Scoring iterations: 5

```