

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")
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-data
climate_data <- read.csv(file_path)
climate_data <- climate_data[, c("Crop_Yield_MT_per_HA", setdiff(names(climate_data), "Crop_Yield_MT_per_HA"))]

```

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
Irrigation_Access_	0	0
Pesticide_Use_KG_per_HA	0	0

Variable	Missing_Count	Missing_Percentage
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

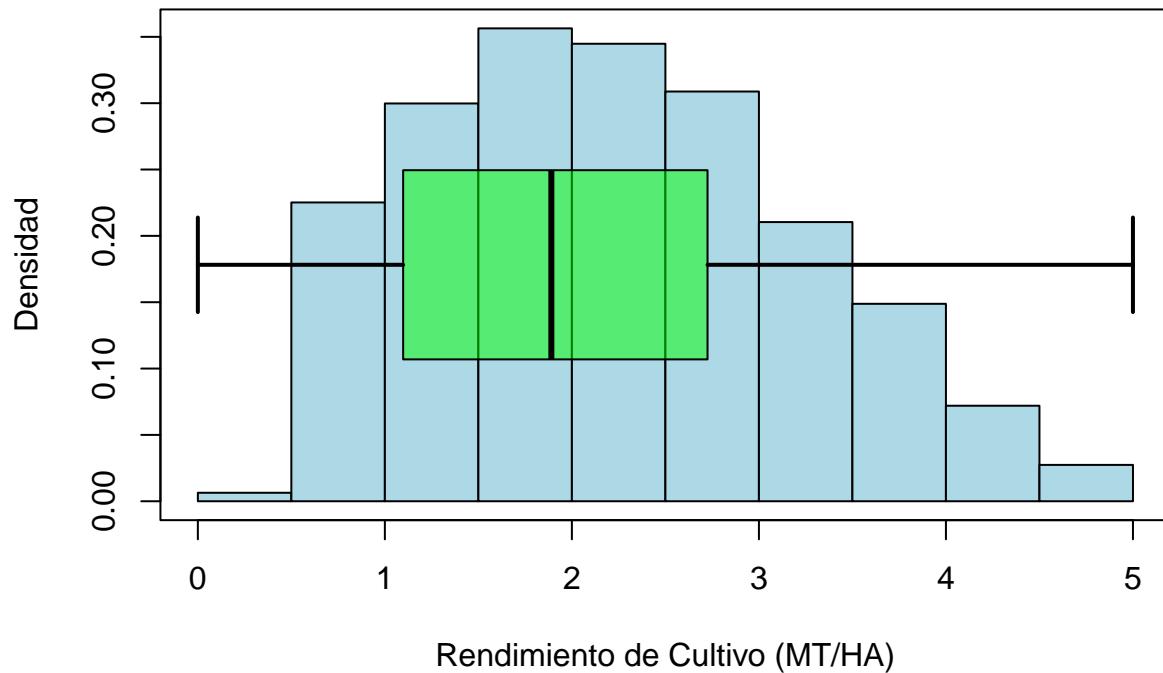
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Productividad Agrícola",
      xlab = "Rendimiento de Cultivo (MT/HA)",
      ylab = "Densidad")

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

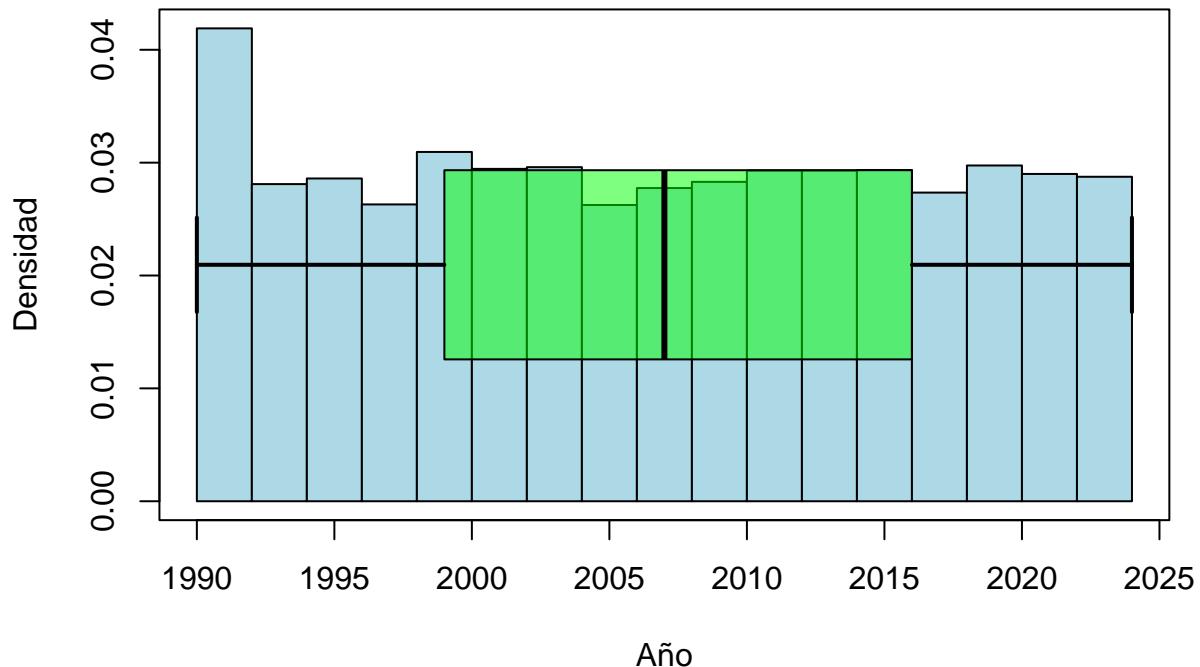
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot del Año",
      xlab = "Año",
      ylab = "Densidad")

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

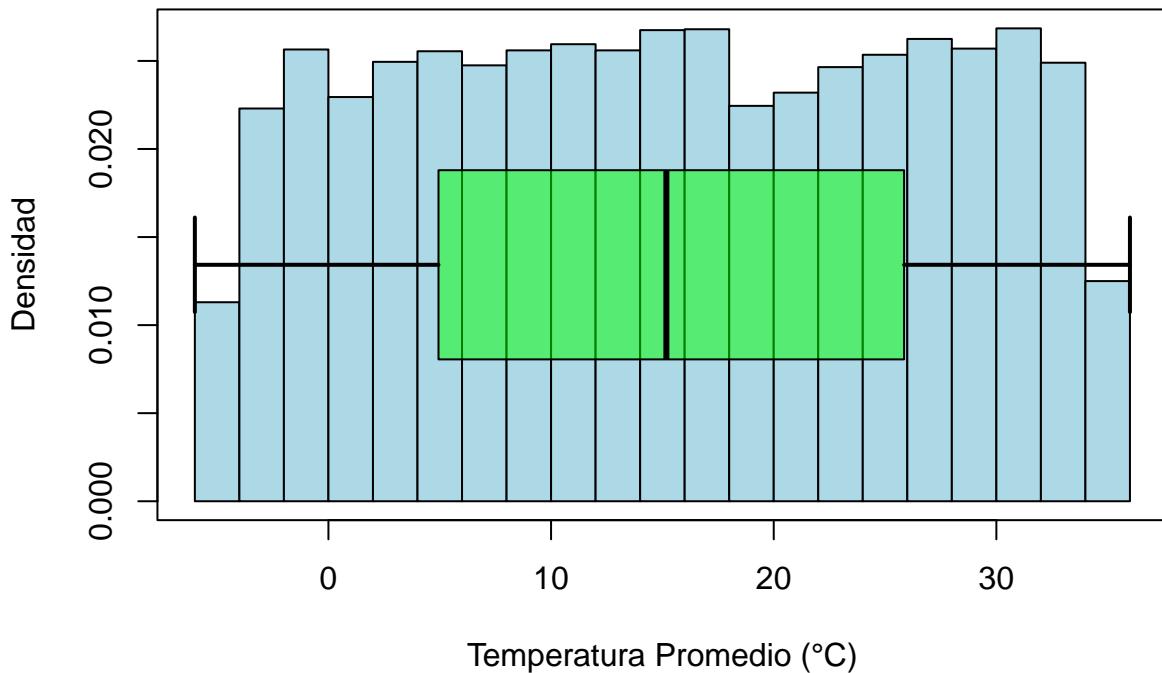
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Temperatura Promedio",
      xlab = "Temperatura Promedio (°C)",
      ylab = "Densidad")

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

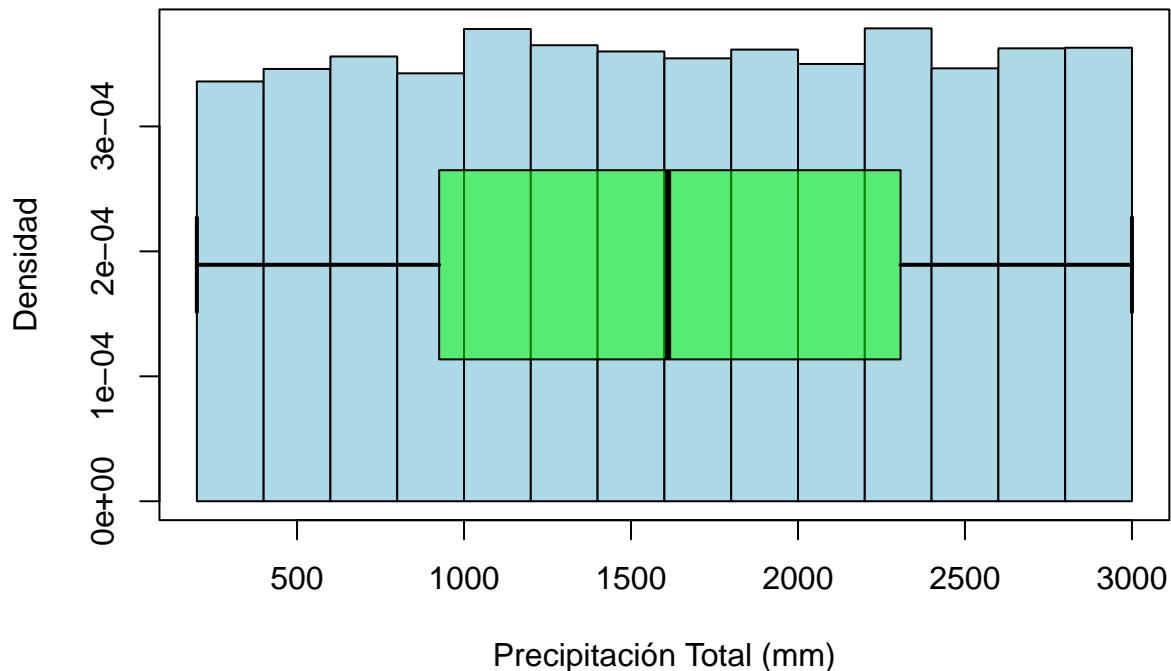
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Precipitación Total",
      xlab = "Precipitación Total (mm)",
      ylab = "Densidad")

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

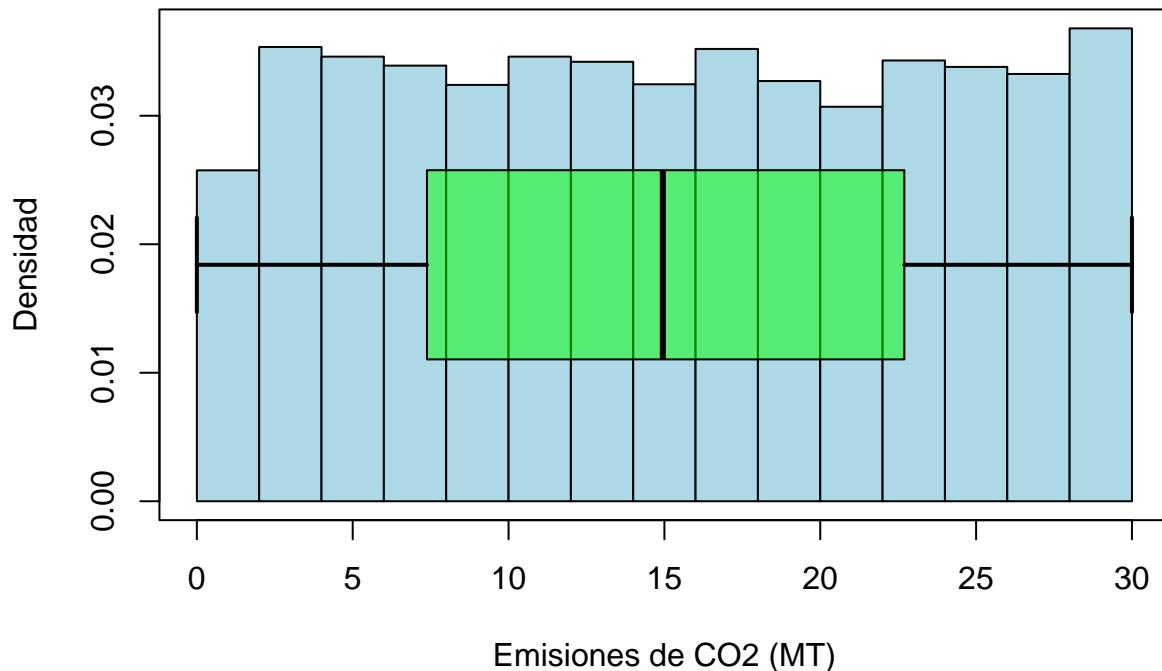
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Emisiones de CO2",
      xlab = "Emisiones de CO2 (MT)",
      ylab = "Densidad")

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

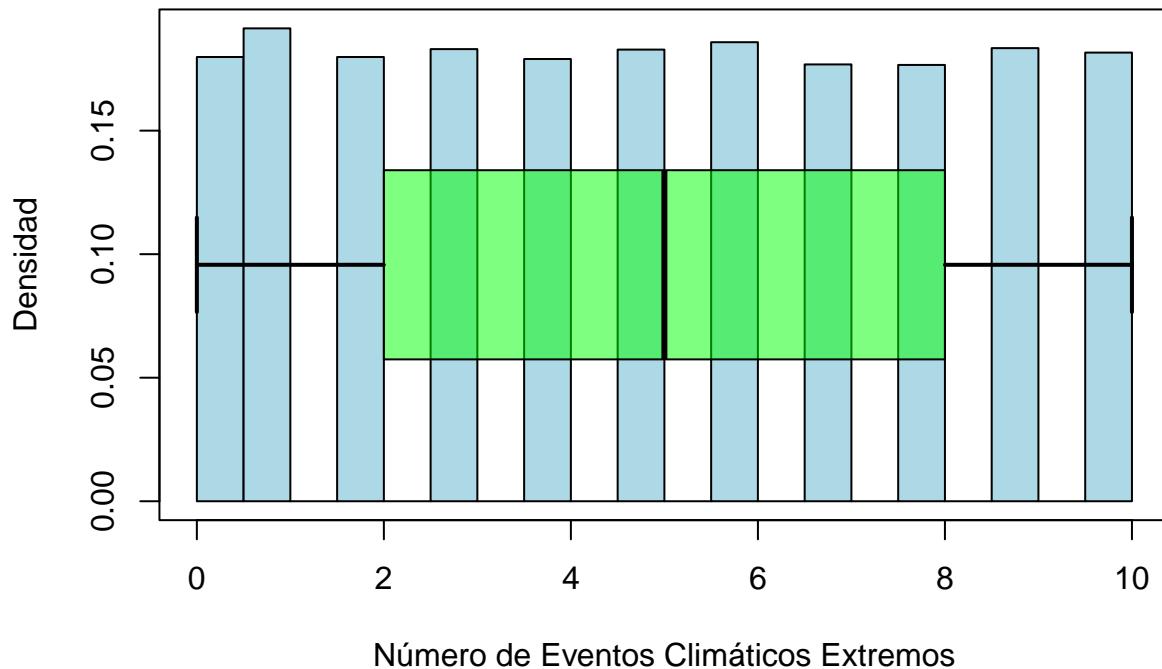
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Eventos Climáticos Extremos",
      xlab = "Número de Eventos Climáticos Extremos",
      ylab = "Densidad")

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_.

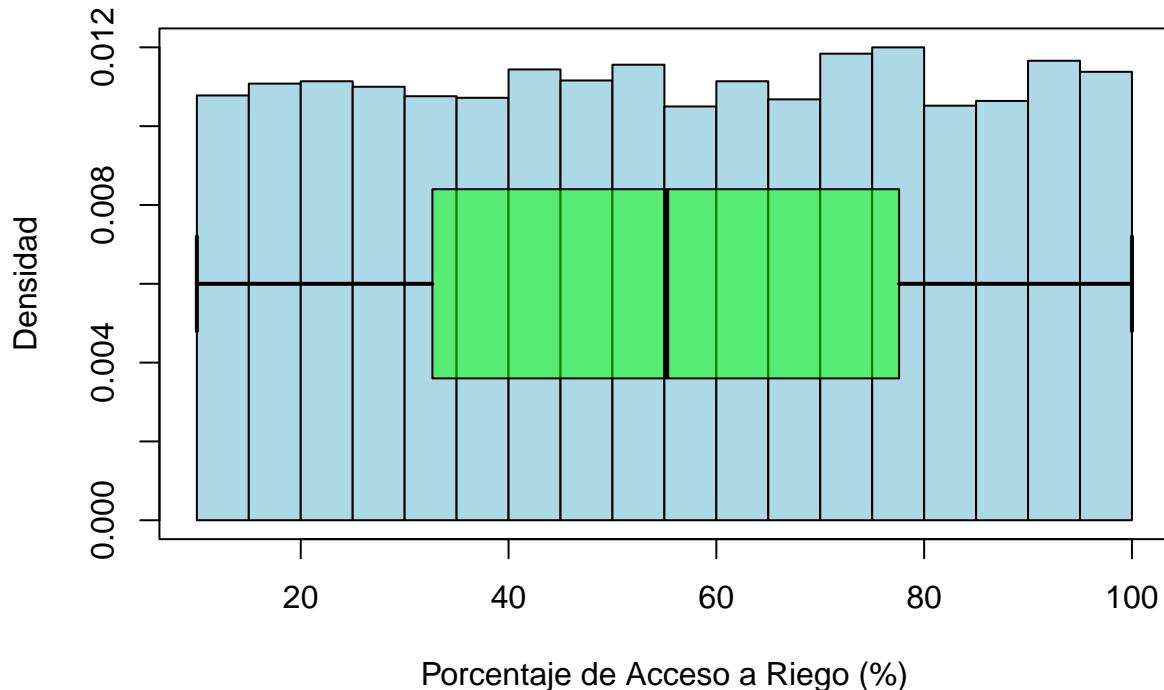
hist(x, prob = TRUE,
  col = "lightblue",
  main = "Histograma y Boxplot de Acceso a Riego",
  xlab = "Porcentaje de Acceso a Riego (%)",
  ylab = "Densidad")

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

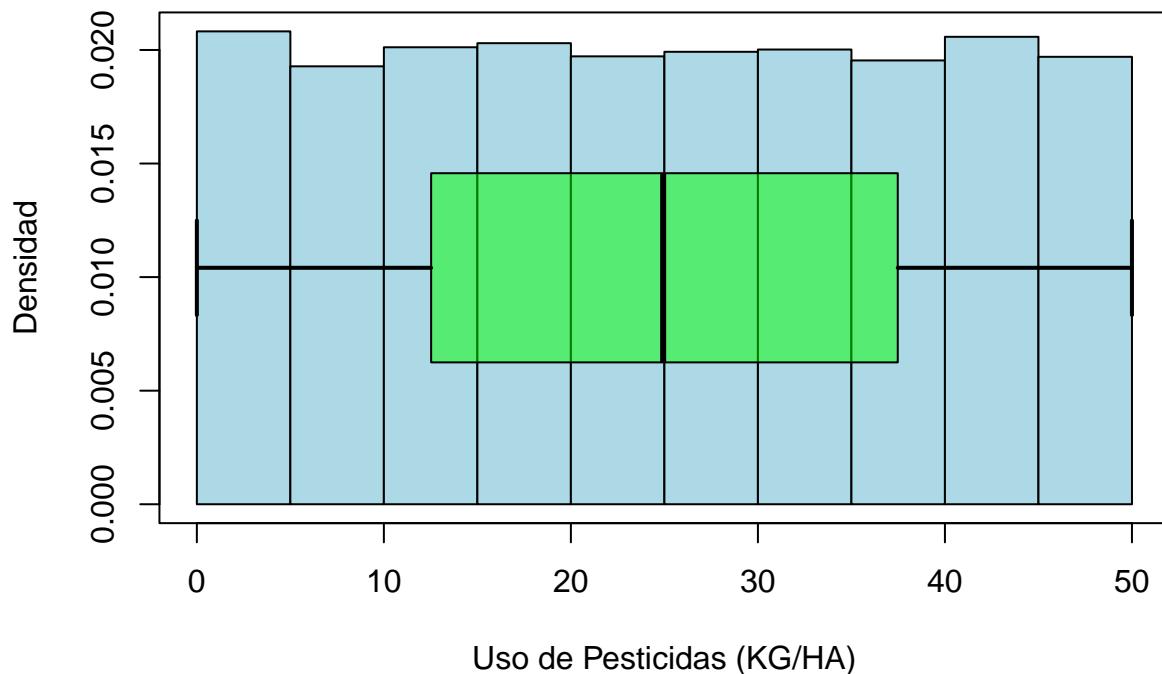
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Uso de Pesticidas",
      xlab = "Uso de Pesticidas (KG/HA)",
      ylab = "Densidad")

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

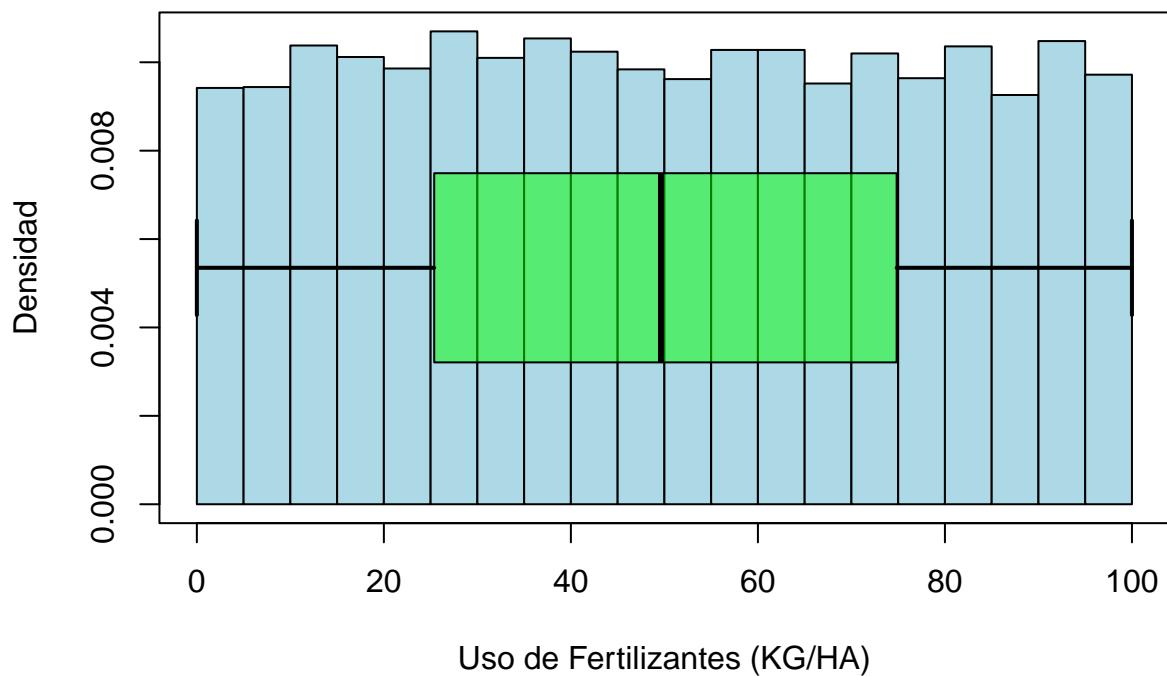
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot de Uso de Fertilizantes",
      xlab = "Uso de Fertilizantes (KG/HA)",
      ylab = "Densidad")

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

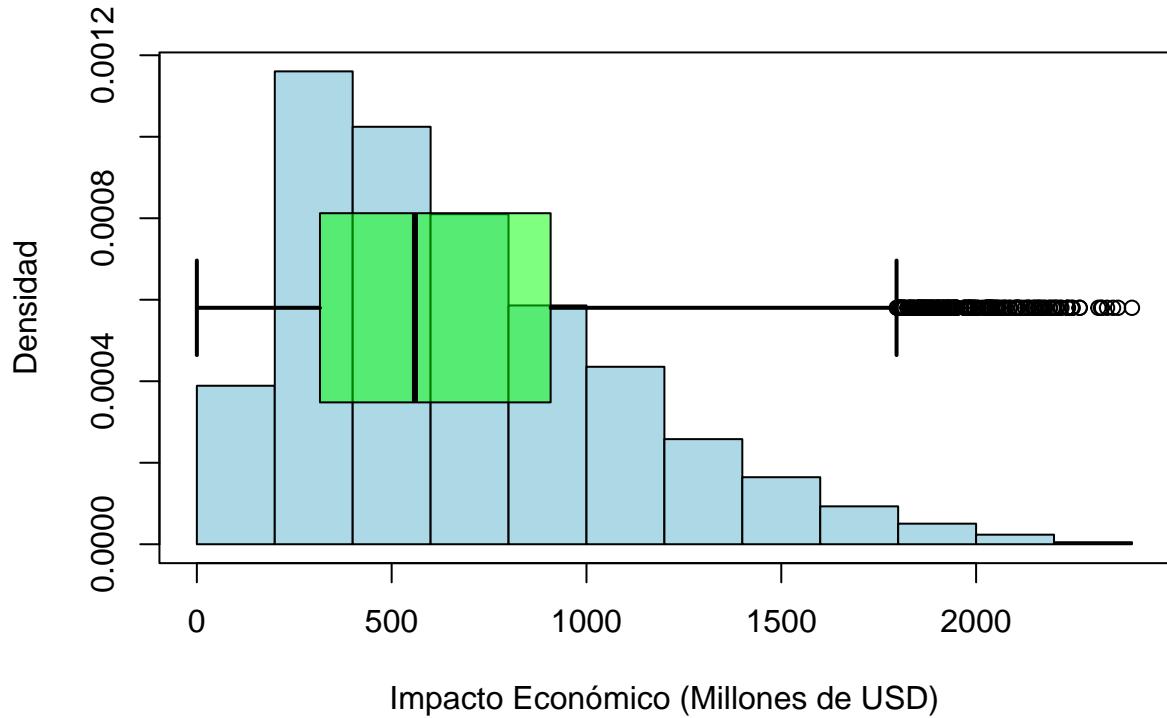
hist(x, prob = TRUE,
      col = "lightblue",
      main = "Histograma y Boxplot del Impacto Económico",
      xlab = "Impacto Económico (Millones de USD)",
      ylab = "Densidad")

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

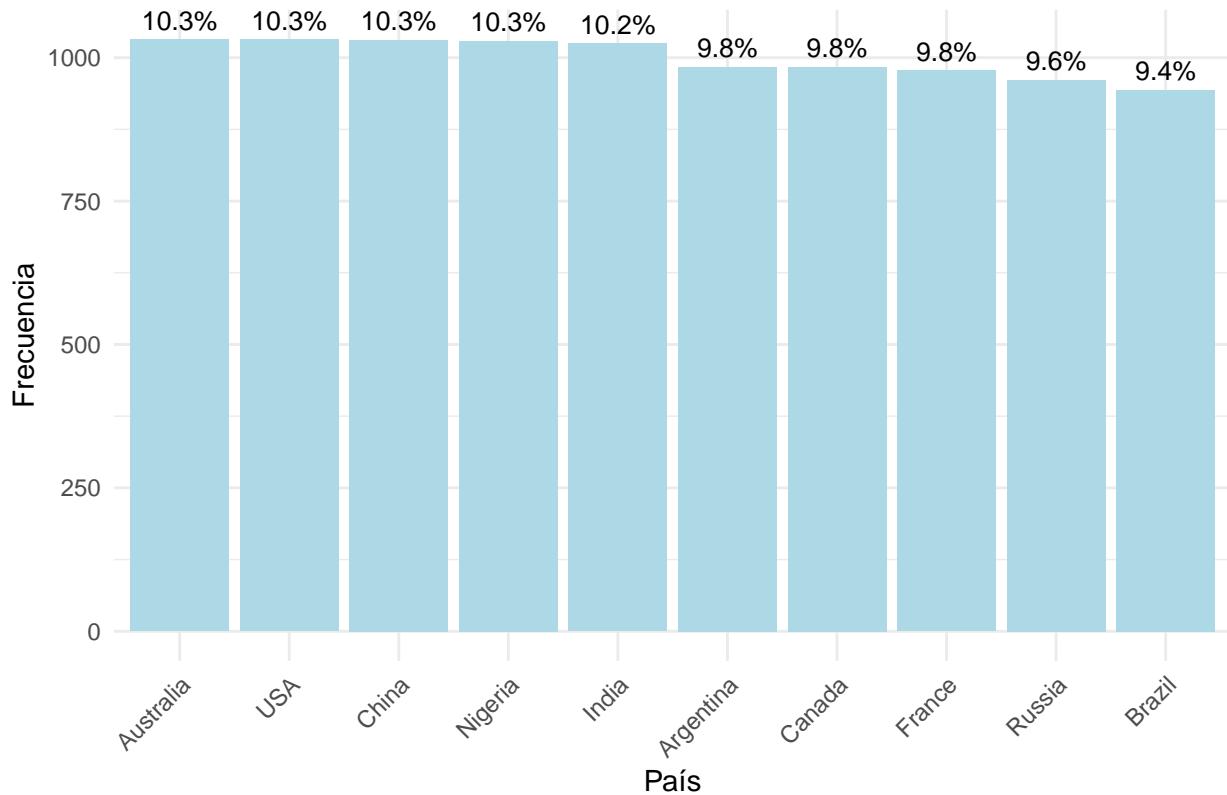
A continuación, se presentan las frecuencias para las variables categóricas del dataset.

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

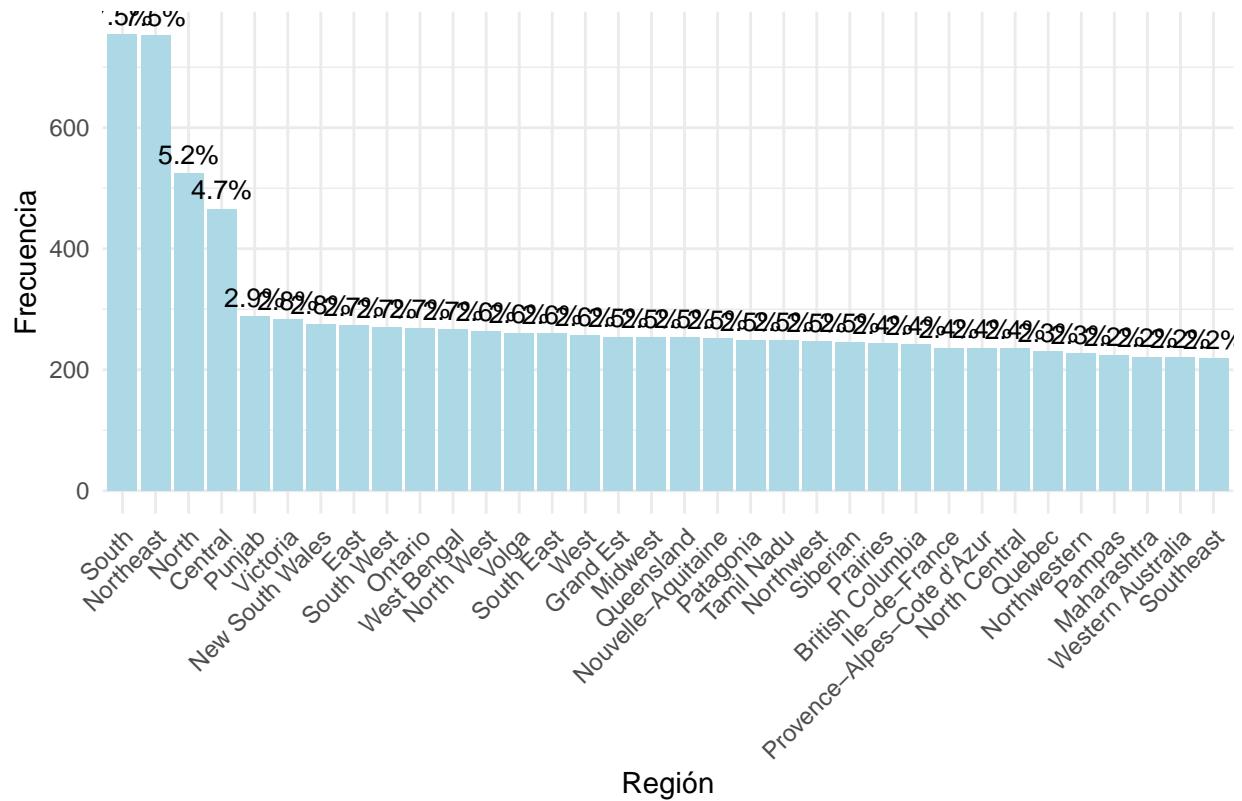


Frecuencia de Regiones Region

```
# 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

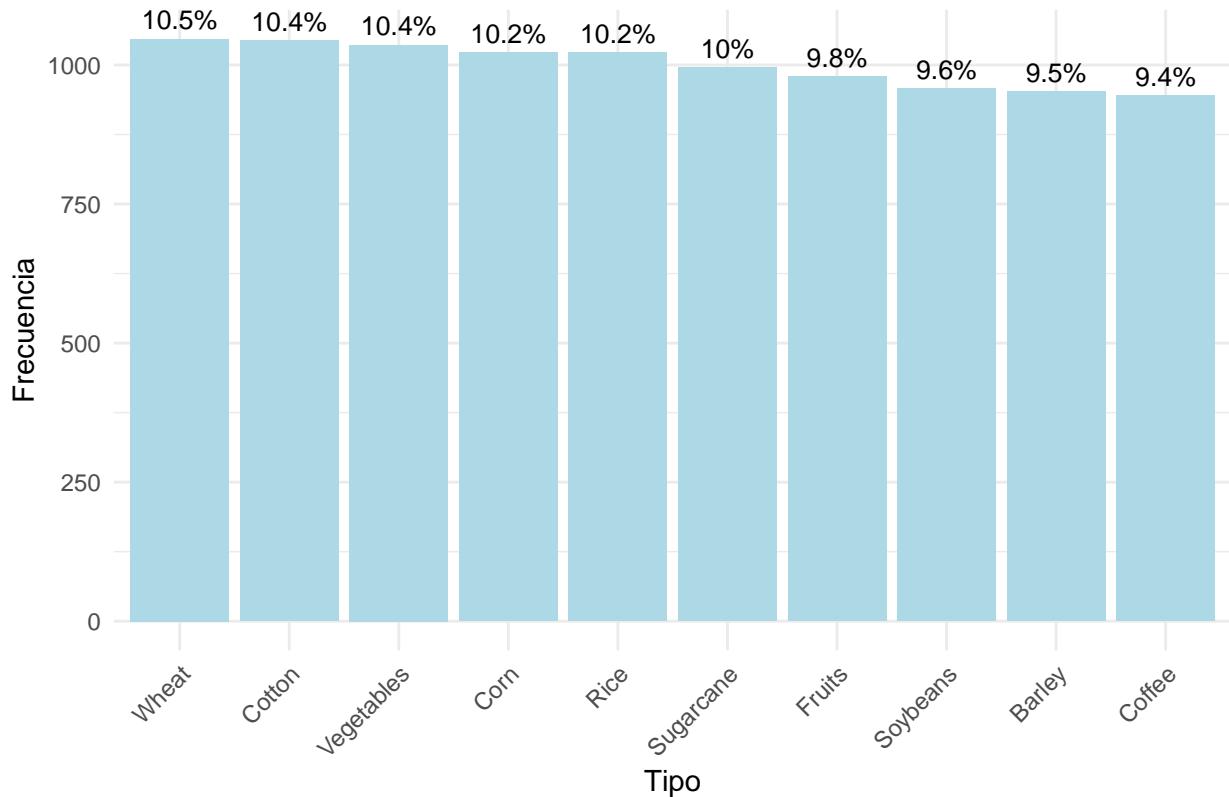


Frecuencia de Tipos de Cultivo Crop_Type

```
# 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))
```

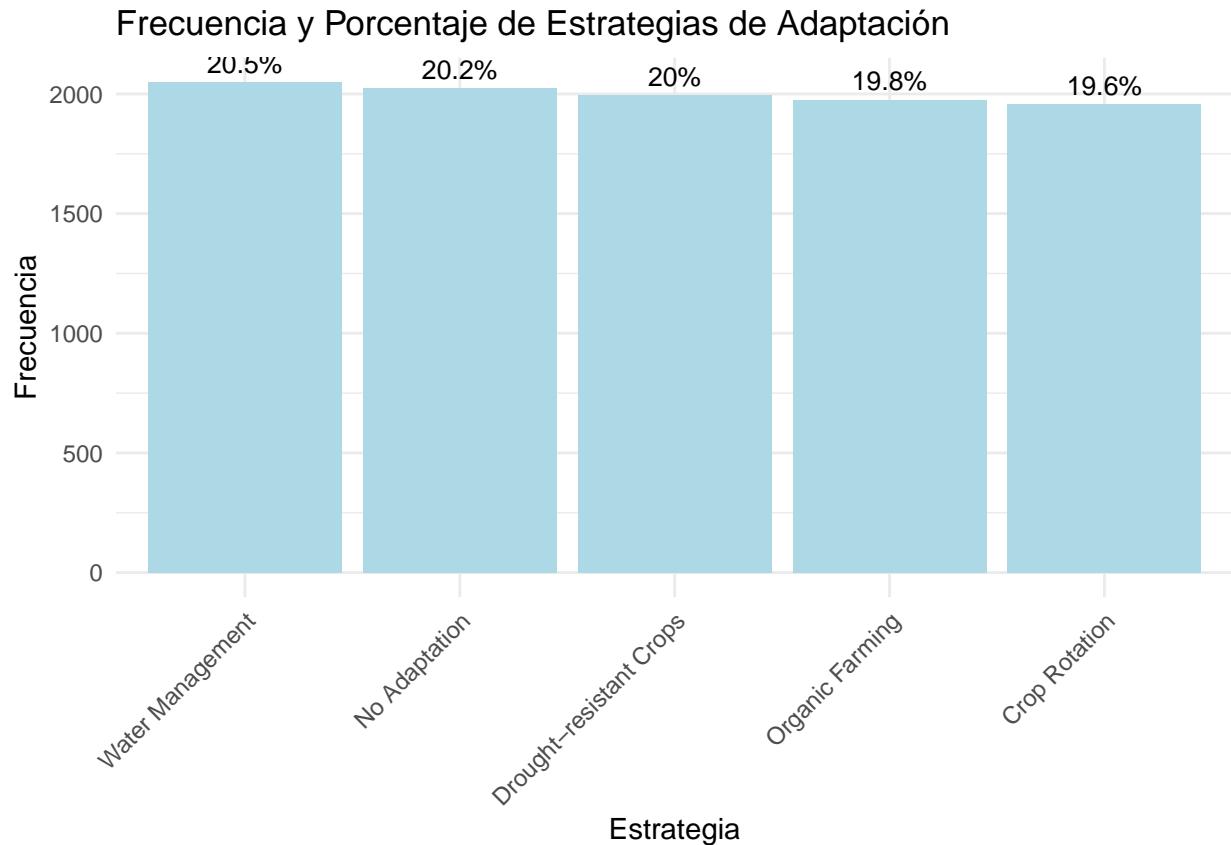
Frecuencia y Porcentaje de Tipos de Cultivo



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)]
```

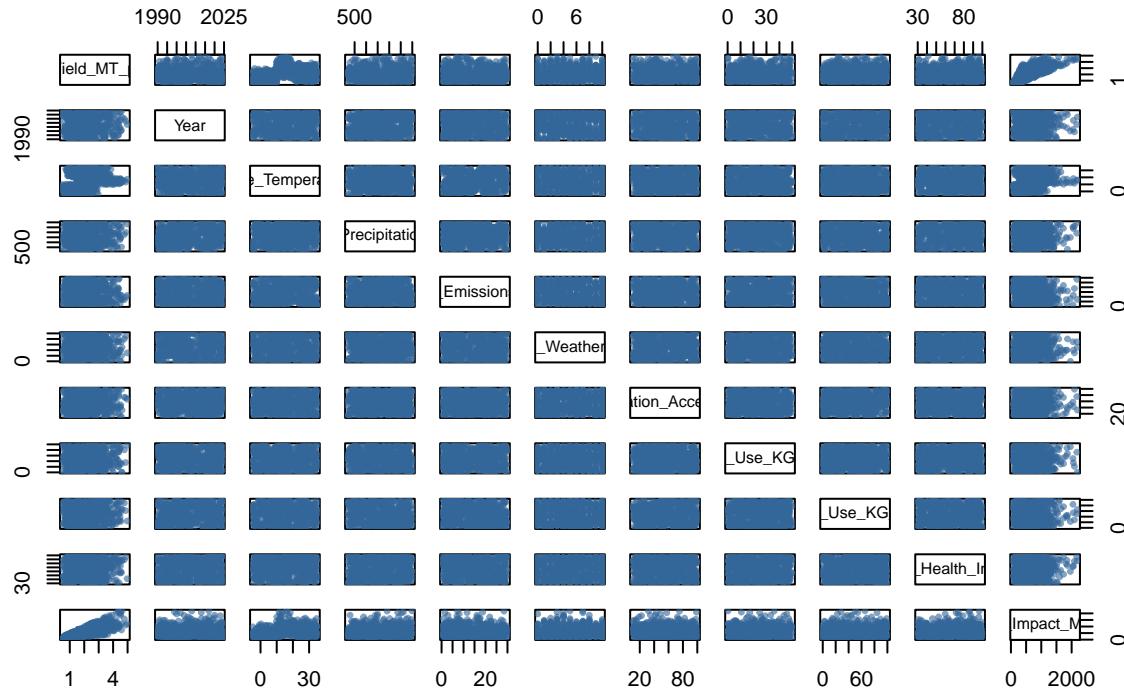
Hagamos el plot

```
# 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

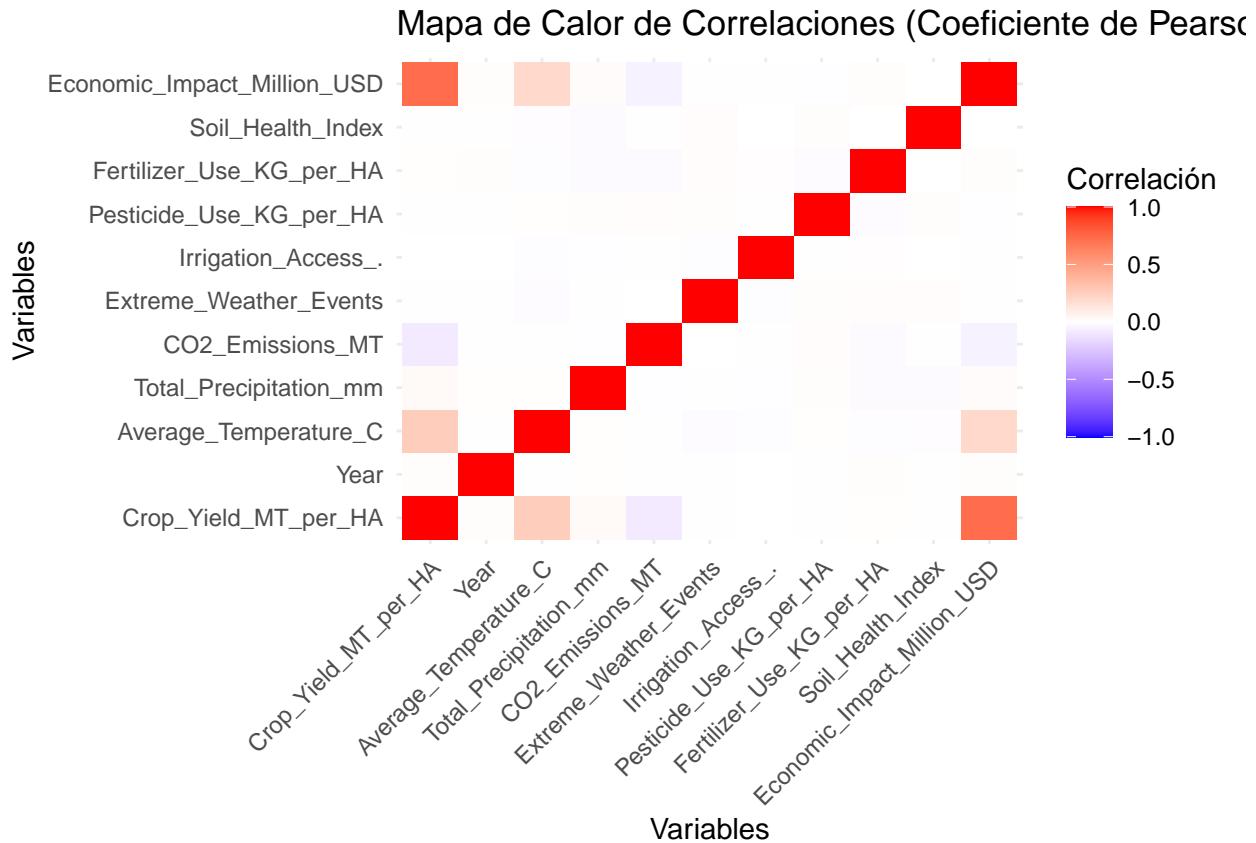
El coeficiente de correlación de Pearson mide la relación lineal entre variables numéricas. Un valor cercano a 1 o -1 indica una fuerte correlación positiva o negativa, respectivamente, mientras que un valor cercano a 0 indica una correlación débil o inexistente.

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"
)

```

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 2: 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)

```

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.

Para nuestro primer modelo, tomaremos la matriz de diseño siguiente

```

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

##
## Call:
## glm(formula = Crop_Yield_MT_per_HA ~ ., family = Gamma(link = "log"),
##      data = climate_data)
##
## Coefficients:
##                               Estimate Std. Error t value
## (Intercept)                7.212e-01  3.201e-02 22.526
## Crop_TypeCoffee            -3.981e-02  1.982e-02 -2.008
## Crop_TypeCorn              -3.474e-02  1.944e-02 -1.787
## Crop_TypeCotton            -5.819e-02  1.934e-02 -3.008
## Crop_TypeFruits            -6.598e-03  1.966e-02 -0.336
## Crop_TypeRice              -2.334e-02  1.944e-02 -1.201
## Crop_TypeSoybeans           3.409e-02  1.975e-02 -1.726
## Crop_TypeSugarcane         -1.751e-02  1.957e-02 -0.894
## Crop_TypeVegetables        -4.401e-02  1.938e-02 -2.270
## Crop_TypeWheat              2.367e-02  1.933e-02 -1.225
## Average_Temperature_C      1.174e-02  3.767e-04 31.156
## Total_Precipitation_mm     1.442e-05  5.367e-06  2.688
## CO2_Emissions_MT           -4.542e-03  5.027e-04 -9.034
## Extreme_Weather_Events     8.653e-05  1.365e-03  0.063
## Irrigation_Access_.        8.682e-05  1.661e-04  0.523
## Pesticide_Use_KG_per_HA   -2.344e-04  2.982e-04 -0.786
## Fertilizer_Use_KG_per_HA   1.499e-04  1.505e-04  0.996
## Soil_Health_Index           -6.358e-05  2.138e-04 -0.297
## Adaptation_StrategiesDrought-resistant Crops -7.652e-03  1.374e-02 -0.557
## Adaptation_StrategiesNo Adaptation          -5.648e-03  1.369e-02 -0.413
## Adaptation_StrategiesOrganic Farming         -8.934e-03  1.377e-02 -0.649
## Adaptation_StrategiesWater Management       -2.052e-02  1.365e-02 -1.503
## ContinentAmericas          -2.528e-02  1.512e-02 -1.672
## ContinentAsia               -1.635e-02  1.649e-02 -0.991
## ContinentEurope             -3.360e-02  1.666e-02 -2.017
## ContinentOceania            2.280e-02  1.904e-02 -1.197
##                               Pr(>|t|)
## (Intercept)                < 2e-16 ***
## Crop_TypeCoffee             0.04465 *
## Crop_TypeCorn                0.07402 .
## Crop_TypeCotton              0.00264 **
## Crop_TypeFruits              0.73713
## Crop_TypeRice                 0.22984
## Crop_TypeSoybeans             0.08436 .
## Crop_TypeSugarcane            0.37119
## Crop_TypeVegetables           0.02320 *
## Crop_TypeWheat                  0.22077
## Average_Temperature_C       < 2e-16 ***
## Total_Precipitation_mm       0.00720 **

```

```

## CO2_Emissions_MT < 2e-16 ***
## Extreme_Weather_Events 0.94946
## Irrigation_Access_. 0.60126
## Pesticide_Use_KG_per_HA 0.43191
## Fertilizer_Use_KG_per_HA 0.31929
## Soil_Health_Index 0.76616
## Adaptation_StrategiesDrought-resistant Crops 0.57760
## Adaptation_StrategiesNo Adaptation 0.67997
## Adaptation_StrategiesOrganic Farming 0.51644
## Adaptation_StrategiesWater Management 0.13291
## ContinentAmericas 0.09455 .
## ContinentAsia 0.32148
## ContinentEurope 0.04373 *
## ContinentOceania 0.23117
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.186067)
##
## Null deviance: 2291.5 on 9999 degrees of freedom
## Residual deviance: 2111.7 on 9974 degrees of freedom
## AIC: 27072
##
## Number of Fisher Scoring iterations: 4

```

Inicialmente, empleamos una función liga logarítmica (log). Ahora, también exploraremos el uso de la función de enlace inverso (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:
##                               Estimate Std. Error t value
## (Intercept) 4.806e-01 1.425e-02 33.729
## Crop_TypeCoffee 1.619e-02 8.767e-03 1.846
## Crop_TypeCorn 1.555e-02 8.599e-03 1.808
## Crop_TypeCotton 2.429e-02 8.654e-03 2.807
## Crop_TypeFruits 3.566e-03 8.556e-03 0.417
## Crop_TypeRice 9.387e-03 8.544e-03 1.099
## Crop_TypeSoybeans 1.469e-02 8.729e-03 1.683
## Crop_TypeSugarcane 5.346e-03 8.576e-03 0.623
## Crop_TypeVegetables 1.877e-02 8.613e-03 2.180
## Crop_TypeWheat 9.303e-03 8.466e-03 1.099
## Average_Temperature_C -4.580e-03 1.696e-04 -26.997
## Total_Precipitation_mm -6.469e-06 2.391e-06 -2.705

```

```

## CO2_Emissions_MT           1.986e-03  2.239e-04  8.872
## Extreme_Weather_Events    2.618e-05  6.075e-04  0.043
## Irrigation_Access_.       -2.524e-05 7.400e-05 -0.341
## Pesticide_Use_KG_per_HA   6.787e-05  1.325e-04  0.512
## Fertilizer_Use_KG_per_HA  -6.557e-05 6.690e-05 -0.980
## Soil_Health_Index          1.890e-05  9.501e-05  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|)
## (Intercept)                < 2e-16 ***
## Crop_TypeCoffee             0.06488 .
## Crop_TypeCorn               0.07063 .
## Crop_TypeCotton              0.00500 **
## Crop_TypeFruits             0.67682
## Crop_TypeRice               0.27191
## Crop_TypeSoybeans            0.09246 .
## Crop_TypeSugarcane          0.53306
## Crop_TypeVegetables         0.02931 *
## Crop_TypeWheat              0.27182
## Average_Temperature_C      < 2e-16 ***
## Total_Precipitation_mm      0.00684 **
## CO2_Emissions_MT            < 2e-16 ***
## Extreme_Weather_Events      0.96562
## Irrigation_Access_.         0.73303
## Pesticide_Use_KG_per_HA    0.60849
## Fertilizer_Use_KG_per_HA   0.32704
## Soil_Health_Index            0.84231
## Adaptation_StrategiesDrought-resistant Crops 0.53792
## Adaptation_StrategiesNo Adaptation        0.66441
## Adaptation_StrategiesOrganic Farming       0.49445
## Adaptation_StrategiesWater Management     0.15024
## ContinentAmericas          0.13466
## ContinentAsia              0.39027
## ContinentEurope            0.06108 .
## ContinentOceania           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 dos modelos reducidos

```

# Ajustar el modelo GLM con enlace inverso
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")
)
summary(glm_log.red)

##
## 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)
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)           6.940e-01  1.898e-02 36.565 < 2e-16 ***
## Average_Temperature_C 1.172e-02  3.763e-04 31.140 < 2e-16 ***
## CO2_Emissions_MT      -4.582e-03 5.022e-04 -9.123 < 2e-16 ***
## Total_Precipitation_mm 1.424e-05  5.359e-06  2.656  0.00791 **
## Crop_TypeCoffee        -3.963e-02 1.980e-02 -2.001  0.04539 *
## Crop_TypeCorn          -3.409e-02 1.942e-02 -1.755  0.07931 .
## Crop_TypeCotton        -5.752e-02 1.933e-02 -2.977  0.00292 **
## Crop_TypeFruits        -5.929e-03 1.963e-02 -0.302  0.76265
## Crop_TypeRice          -2.289e-02 1.942e-02 -1.179  0.23860
## Crop_TypeSoybeans      -3.343e-02 1.973e-02 -1.694  0.09030 .
## Crop_TypeSugarcane    -1.674e-02 1.955e-02 -0.856  0.39178
## Crop_TypeVegetables   -4.445e-02 1.936e-02 -2.296  0.02171 *
## Crop_TypeWheat         -2.353e-02 1.932e-02 -1.218  0.22320
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.1859394)
##
## Null deviance: 2291.5 on 9999 degrees of freedom
## Residual deviance: 2113.4 on 9987 degrees of freedom
## AIC: 27054
##
## Number of Fisher Scoring iterations: 4

# Ajustar el modelo 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")
)
summary(glm_inverse.red)

##
## 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)
##
## Coefficients:

```

```

##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)           4.913e-01  8.479e-03 57.949 < 2e-16 ***
## Average_Temperature_C -4.578e-03  1.695e-04 -27.012 < 2e-16 ***
## CO2_Emissions_MT      2.005e-03  2.236e-04  8.966 < 2e-16 ***
## Total_Precipitation_mm -6.428e-06  2.387e-06 -2.693  0.00710 **
## Crop_TypeCoffee        1.615e-02  8.759e-03  1.844  0.06522 .
## Crop_TypeCorn          1.527e-02  8.589e-03  1.777  0.07555 .
## Crop_TypeCotton        2.412e-02  8.645e-03  2.790  0.00529 **
## Crop_TypeFruits        3.417e-03  8.547e-03  0.400  0.68929
## Crop_TypeRice          9.216e-03  8.537e-03  1.079  0.28040
## Crop_TypeSoybeans      1.451e-02  8.722e-03  1.664  0.09617 .
## Crop_TypeSugarcane    5.099e-03  8.566e-03  0.595  0.55171
## Crop_TypeVegetables   1.891e-02  8.603e-03  2.198  0.02796 *
## Crop_TypeWheat         9.245e-03  8.459e-03  1.093  0.27445
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.18776)
##
## Null deviance: 2291.5  on 9999  degrees of freedom
## Residual deviance: 2133.7  on 9987  degrees of freedom
## AIC: 27153
##
## Number of Fisher Scoring iterations: 5

```

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 distribución de los residuos** sigue el supuesto de la familia gamma.
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.
4. **La dispersión residual** es razonable y consistente con los supuestos.

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 valores predichos
create_residual_vs_predicted <- function(model, model_name) {
  # Residuos de Pearson
  pearson_residuals <- residuals(model, type = "pearson")
  pred_values <- predict(model, type = "response")

  scatter_pearson <- ggplot() +

```

```

aes(x = pred_values, 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 = "Valores Predichos",
y = "Residuos de Pearson") +
theme_minimal()

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

scatter_deviance <- ggplot() +
aes(x = pred_values, y = deviance_residuals) +
geom_point(alpha = 0.5) +
geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
labs(title = paste("Residuos de Deviance -", model_name),
x = "Valores Predichos",
y = "Residuos de Deviance") +
theme_minimal()

return(list(scatter_pearson, scatter_deviance))
}

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

```

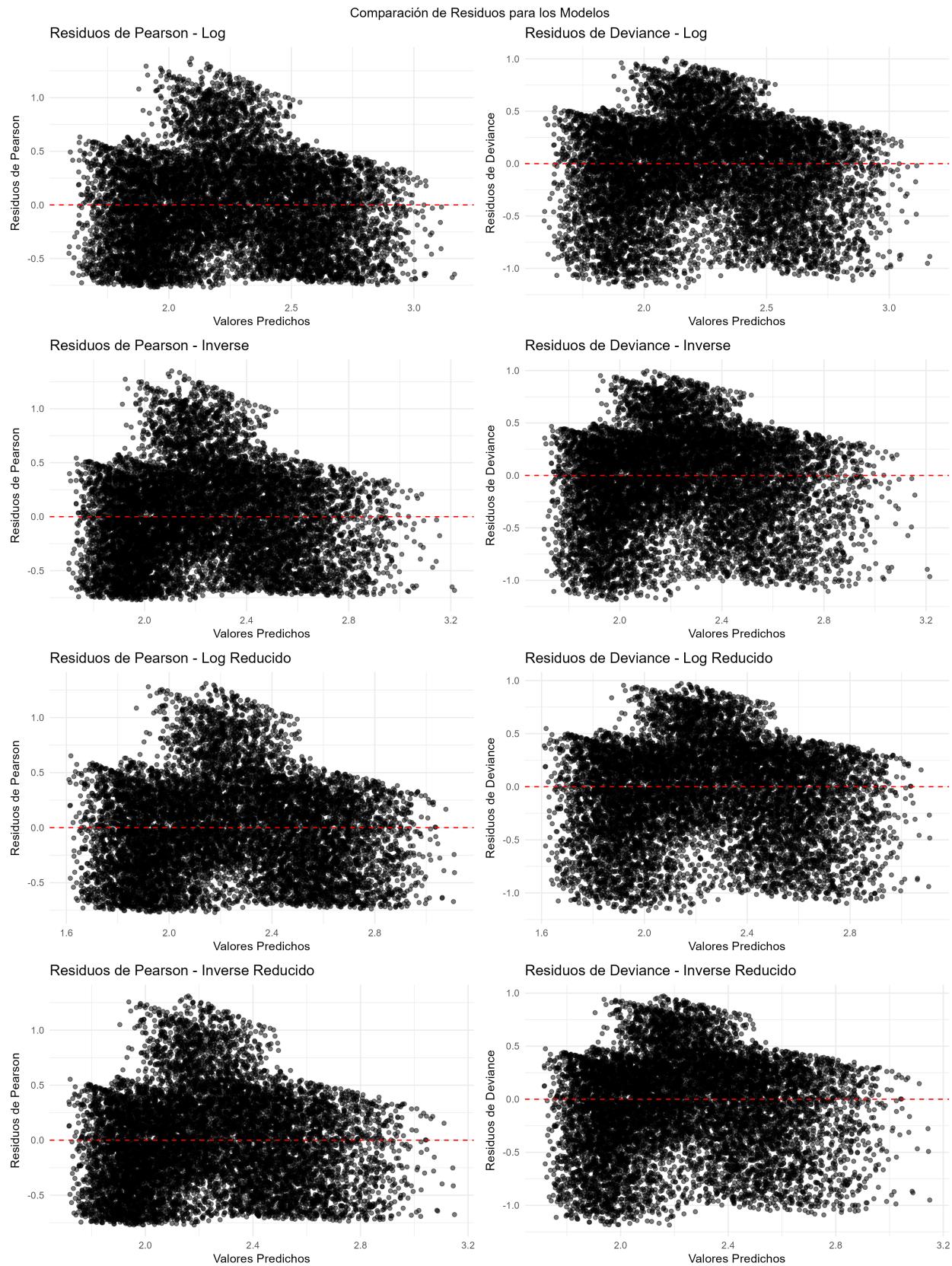


Figure 1: Comparación de Residuos

Valores Atípicos

```
# 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
resultados <- list(outliers_log, outliers_inverse, outliers_log_red, outliers_inverse_red)

# Imprimir resumen de resultados
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

# 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

  # Umbrales
  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")

```

