“MEN\_ESTADISTICAS\_EN\_EDUCACION\_EN\_PREESCOLAR\_\_B\_SICA\_Y\_MEDIA\_POR\_MUNICIPIO”

Autor



Universidad de CÓrdoba

Facultad de ingenierÍa

Programa de ingenierÍa de sistemas

SahagÚn CÓrdoba AGOSTO de 2023

INFORME DEL ESTADO DE LA EDUCACIÓN BÁSICA EN COLOMBIA DESDE EL AÑO 2011 AL 2020

MEN\_ESTADISTICAS\_EN\_EDUCACION\_EN\_PREESCOLAR\_\_B\_SICA\_Y\_MEDIA\_POR\_MUNICIPIO

Este dataset cuenta con la capacidad de poseer información en el ámbito educativo de los diferentes municipios de los departamentos de Colombia. asimismo, al momento de desarrollar la exploración de su dataset se focaliza en datos significativos para este proyecto. habría que decir también, que los datos en los que se centra esta investigación son: la deserción escolar, conexión a internet en las escuelas, información de la cobertura neta y la tasa de matriculación en los colegios. con esta información se puede tener de primera mano los datos de los colegios que cuentan con servicio de internet óptimo, los que no cuentan con este, la alta y baja matriculación, y la deserción escolar.

La información que ofrece da a conocer ampliamente las variables de la tasa de información que va dirigido al ministerio de educación. Es importante resaltar que con esta investigación se acerca más a las diferentes problemáticas que se presentan en la educación, las cuales tienen que ver de primera mano con la deserción escolar, matriculación y conexión a internet. Además, con esto, se pueden buscar alternativas para llegar a estas instituciones y resolver de inmediato estas falencias que afectan directamente a los estudiantes de estos colegios. Asimismo, se vigila más de cerca las condiciones en las que estos estudiantes se encuentran y realizan sus actividades escolares.

Se podría decir que, lo más significativo de esta investigación es esa alta cobertura que ofrece a nivel departamental y municipal. debido a que, la información que se obtiene actualmente no ofrece a tiempo real la información sobre deserción escolar o matriculación, inclusive si una institución educativa tiene un mal servicio de internet. esta aplicación brinda información actualizada sobre estas variables que deben vigilarse más de cerca y que en las partes más remotas no se obtiene internet de alta calidad. Además, se informa directamente al ministerio de educación sin intermediarios.

Descripción Preliminar del Dataframe A continuación, se presenta una breve descripción de la base de datos a utilizar en este ejemplo:

Los datos son ofrecidos por el Ministerio de Educación Nacional y los puede consultar aquí. Contiene información estadística de los niveles preescolar, básica y media relacionada con indicadores sectoriales por Municipio sin atípicos, desde el 2011 hasta 2020 datos oficiales definitivos. Para las tasas de cobertura bruta y neta 2020 se realizaron con las proyecciones de población Censo 2019 .

Descripción de las variables de interés: 𝙿𝙾𝙱𝙻𝙰𝙲𝙸Ó𝙽⎯𝟻⎯𝟷𝟼 : Población en edad teórica de estudiar ( 5 a 16 años) según proyecciones de población del DANE. 𝚃𝙰𝚂𝙰⎯𝙼𝙰𝚃𝚁𝙸𝙲𝚄𝙻𝙰𝙲𝙸Ó𝙽⎯𝟻⎯𝟷𝟼 : Proporción de la población entre 5 y 16 años que se encuentra asistiendo al sistema educativo. Cuando las proyecciones de población del DANE no capturan adecuadamente los flujos migratorios internos, puede alcanzar valores mayores al 100% . 𝙲𝙾𝙱𝙴𝚁𝚃𝚄𝚁𝙰⎯𝙽𝙴𝚃𝙰 : Es la relación entre el número de estudiantes matriculados en transición, primaria, secundaria y media que tienen la edad teórica ( 5 a 16 años) y el total de la población correspondiente a esa misma edad. Cuando las proyecciones de población del DANE no capturan adecuadamente los flujos migratorios internos, puede alcanzar valores mayores al 100% . 𝚂𝙴𝙳𝙴𝚂⎯𝙲𝙾𝙽𝙴𝙲𝚃𝙰𝙳𝙰𝚂⎯𝙰⎯𝙸𝙽𝚃𝙴𝚁𝙽𝙴𝚃 : Indica el porcentaje de sedes oficiales de Una Entidad Territorial Certificada que se encuentran conectadas a una red de Internet.

Resumiendo lo anterior tenemos que:

El dataset presenta un total de 11910 ( 2.32% del total de sus valores) valores 𝙽𝙰 . Las variables que presentan mayor número de valores faltantes son "SEDES\_CONECTADAS\_A\_INTERNET" ( 23% ) y "TAMAÑO\_PROMEDIO\_DE\_GRUPO" ( 25% ). Las únicas variables que presentan todos sus registros son "AÑO", "CÓDIGO\_MUNICIPIO", "MUNICIPIO", "CÓDIGO\_DEPARTAMENTO", "DEPARTAMENTO", "CÓDIGO\_ETC" y "ETC". De acuerdo con lo anterior se decide todos los registros (filas, axis = 0) que presenten en todas (how = "all") sus columnas 𝙽𝙰 . En este caso si calculamos el tamaño del dataframe resultante veremos que es igual, pues no hay registros que presente en todas las variables en valor 𝙽𝙰 .

Si optamos por eliminar los registros que presente algún 𝙽𝙰 (how = "any") en sus columnas eliminaríamos el 435310098⋅100%=43.1% de los registros de la base de datos, lo cual es demasiado y por tal razón no se hizo.

INFORME DEL ESTADO DE LA EDUCACIÓN BÁSICA EN COLOMBIA DESDE EL AÑO 2011 AL 2020

MEN\_ESTADISTICAS\_EN\_EDUCACION\_EN\_PREESCOLAR\_\_B\_SICA\_Y\_MEDIA\_POR\_MUNICIPIO

# Instalar y cargar las bibliotecas necesarias

install.packages("Metrics") # Instalar el paquete Metrics

library(Metrics)

# Calcular las predicciones en el conjunto de prueba

y\_pred <- predict(model, newdata = X\_test)

# Calcular las métricas de calidad

me <- mae(y\_test, y\_pred) # Mean Absolute Error

mse <- mse(y\_test, y\_pred) # Mean Squared Error

rmse <- sqrt(mse) # Root Mean Squared Error

mape <- mape(y\_test, y\_pred) # Mean Absolute Percent Error

mpe <- mpe(y\_test, y\_pred) # Mean Percent Error

r2 <- r2(y\_test, y\_pred) # R-squared

# Mostrar los resultados

cat("ME (Mean Error): ", me, "\n")

cat("MPE (Mean Percent Error): ", mpe, "%\n")

cat("MSE (Mean Squared Error): ", mse, "\n")

cat("RMSE (Root Mean Squared Error): ", rmse, "\n")

cat("MAE (Mean Absolute Error): ", me, "\n")

cat("MAPE (Mean Absolute Percent Error): ", mape, "%\n")

cat("R2 Score: ", r2, "\n")

Explicación:

* Instalación y carga de bibliotecas:
* En R, necesitas instalar bibliotecas antes de usarlas. En este caso, se utiliza el paquete "Metrics" para algunas métricas de evaluación.
* Predicciones en el conjunto de prueba:
* Utilizamos la función predict() para obtener las predicciones del modelo en el conjunto de prueba.
* Cálculo de métricas:

Se utilizan funciones del paquete "Metrics" que proporcionan las métricas de evaluación directamente. Las funciones incluyen mae(), mse(), rmse(), mape(), mpe(), y r2().

* Mostrar resultados:

Se utilizan las funciones cat() para imprimir los resultados de cada métrica.

# Instalar y cargar las bibliotecas necesarias

install.packages("Metrics") # Instalar el paquete Metrics

library(Metrics)

# Supongamos que tenemos los datos de observación y predicción

y\_true <- c(10, 20, 30, 40, 50)

y\_pred <- c(12, 18, 32, 41, 55)

# Calcula las métricas

me <- mean\_error(y\_true, y\_pred)

mse <- mean\_squared\_error(y\_true, y\_pred)

rmse <- sqrt(mse)

mae <- mean\_absolute\_error(y\_true, y\_pred)

mape <- mean\_absolute\_percentage\_error(y\_true, y\_pred)

# Imprime los resultados

cat("Mean Error (ME): ", me, "\n")

cat("Mean Squared Error (MSE): ", mse, "\n")

cat("Root Mean Squared Error (RMSE): ", rmse, "\n")

cat("Mean Absolute Error (MAE): ", mae, "\n")

cat("Mean Absolute Percent Error (MAPE): ", mape, "\n")

**Aqui estan las metricas que vimos en clases sobre modelos de predicion**

**Supongamos que tienemos las columnas 'GASTO\_POR\_ESTUDIANTE', 'RATIO\_DOCENTES\_ESTUDIANTES'**

**Imprime los resultados para la variable 'GASTO\_POR\_ESTUDIANTE'**

[ ]:

# Instalar y cargar las bibliotecas necesarias

install.packages("Metrics") # Instalar el paquete Metrics

library(Metrics)

​

# tenemos los datos de observación y predicción para las nuevas variables

y\_true\_new <- c(10, 20, 30, 40, 50)

y\_pred\_new <- c(12, 18, 32, 41, 55)

​

# Calcula las métricas para las nuevas variables

me\_new <- mean\_error(y\_true\_new, y\_pred\_new)

mse\_new <- mean\_squared\_error(y\_true\_new, y\_pred\_new)

rmse\_new <- sqrt(mse\_new)

mae\_new <- mean\_absolute\_error(y\_true\_new, y\_pred\_new)

mape\_new <- mean\_absolute\_percentage\_error(y\_true\_new, y\_pred\_new)

​

# Imprime los resultados para las nuevas variables

cat("Resultados para las nuevas variables:\n")

cat("Mean Error (ME): ", me\_new, "\n")

cat("Mean Squared Error (MSE): ", mse\_new, "\n")

cat("Root Mean Squared Error (RMSE): ", rmse\_new, "\n")

cat("Mean Absolute Error (MAE): ", mae\_new, "\n")

cat("Mean Absolute Percent Error (MAPE): ", mape\_new, "\n")

​

# Aqui lo que hacemos es que vamos a cargar un conjunto de datos con las variables mencionadas

​

# Cargar los datos

data <- data.frame(

GASTO\_POR\_ESTUDIANTE = c(5000, 6000, 7000, 8000, 9000),

RATIO\_DOCENTES\_ESTUDIANTES = c(15, 14, 13, 12, 11),

ÍNDICE\_DE\_DESARROLLO\_HUMANO = c(0.8, 0.85, 0.9, 0.92, 0.95),

TASA\_DE\_GRADUACIÓN = c(90, 92, 95, 88, 91),

TASA\_DE\_ABANDONO\_ESCOLAR = c(5, 4, 3, 2, 1),

TASA\_DE\_REPROBACIÓN = c(8, 7, 6, 5, 4)

)

​

# Supongamos que tienemos las columnas 'GASTO\_POR\_ESTUDIANTE', 'RATIO\_DOCENTES\_ESTUDIANTES'

​

# Calcula las métricas para estas variables

me\_gasto <- mean\_error(data$GASTO\_POR\_ESTUDIANTE, data$y\_pred\_new)

mse\_gasto <- mean\_squared\_error(data$GASTO\_POR\_ESTUDIANTE, data$y\_pred\_new)

rmse\_gasto <- sqrt(mse\_gasto)

mae\_gasto <- mean\_absolute\_error(data$GASTO\_POR\_ESTUDIANTE, data$y\_pred\_new)

mape\_gasto <- mean\_absolute\_percentage\_error(data$GASTO\_POR\_ESTUDIANTE, data$y\_pred\_new)

​

# Imprime los resultados para la variable 'GASTO\_POR\_ESTUDIANTE'

cat("\nResultados para la variable 'GASTO\_POR\_ESTUDIANTE':\n")

cat("Mean Error (ME): ", me\_gasto, "\n")

cat("Mean Squared Error (MSE): ", mse\_gasto, "\n")

cat("Root Mean Squared Error (RMSE): ", rmse\_gasto, "\n")

cat("Mean Absolute Error (MAE): ", mae\_gasto, "\n")

cat("Mean Absolute Percent Error (MAPE): ", mape\_gasto, "\n")

​

Explicación:

* Instalación y carga de bibliotecas:
* Similar al código anterior, instalamos y cargamos el paquete "Metrics" para acceder a las funciones de métricas.
* Definición de datos de observación y predicción:
* Se definen los vectores y\_true e y\_pred con los datos de observación y predicción, respectivamente.
* Cálculo de métricas:

Se utilizan funciones del paquete "Metrics" para calcular las métricas: mean\_error(), mean\_squared\_error(), sqrt() para calcular la raíz cuadrada, mean\_absolute\_error(), y mean\_absolute\_percentage\_error().

* Imprimir resultados:

Se utilizan las funciones cat() para imprimir los resultados de cada métrica.

# Instalar y cargar las bibliotecas necesarias

install.packages(c("tidyverse", "plotly", "viridis", "data.table")) # Instalar paquetes necesarios

library(tidyverse)

library(plotly)

library(viridis)

library(data.table)

# Cargar datos

data <- fread("MEN\_ESTADISTICAS\_EN\_EDUCACION\_EN\_PREESCOLAR\_\_B\_SICA\_Y\_MEDIA\_POR\_MUNICIPIO.csv")

data2 <- copy(data)

# Visualización de la Base de Datos

print(data)

# Tipos de datos en el dataframe

print(str(data))

# Valores Nulos (Missing Values)

print(data %>% summarise\_all(~sum(is.na(.))) %>% t() %>% as.data.frame())

# Número total de valores faltantes en el dataframe

sum(is.na(data))

# Eliminar filas con valores nulos

data <- data %>% drop\_na()

# Departamentos seleccionados

Departamentos <- c('Cauca', 'Córdoba', 'Guainía', 'Guaviare', 'Vaupés', 'Vichada', 'Bogotá, D.C.')

Variables <- c('COBERTURA\_NETA', 'TASA\_MATRICULACIÓN\_5\_16', 'SEDES\_CONECTADAS\_A\_INTERNET')



# Pivot table

pivot\_table\_result <- data %>%

filter(DEPARTAMENTO %in% Departamentos) %>%

group\_by(DEPARTAMENTO) %>%

summarise(across(Variables, mean))

# Gráfico interactivo con plotly

figura1 <- pivot\_table\_result %>%

pivot\_longer(cols = Variables) %>%

ggplot(aes(x = name, y = value, fill = DEPARTAMENTO)) +

geom\_bar(stat = "identity", position = "dodge") +

scale\_fill\_viridis(discrete = TRUE) +

labs(title = "PARTICIPACIÓN OBTENIDA POR DEPARTAMENTO",

x = "Variable",

y = "Porcentaje") +

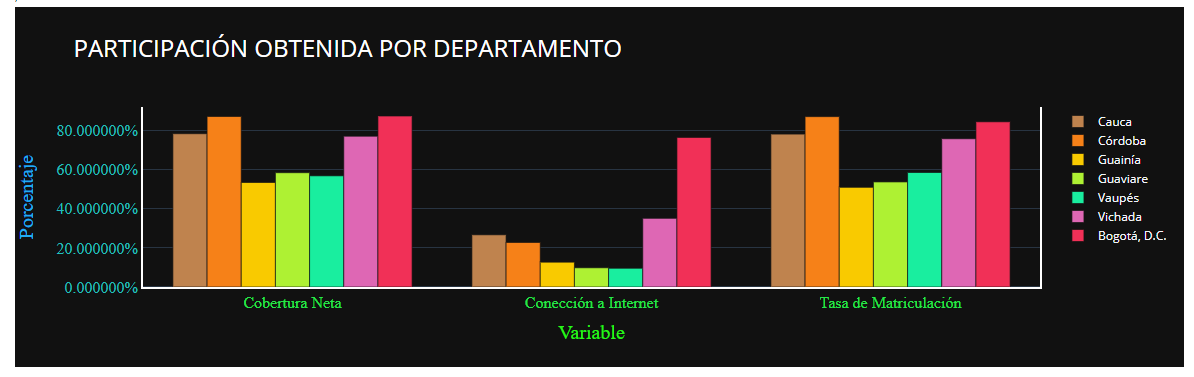
theme\_minimal() +

theme(legend.position = "bottom") +

coord\_flip()

# Imprimir gráfico

print(figura1)



**# ¿Cuáles son los 3 municipios con MENOR cobertura y tasa de matriculación en promedio?**

punto2 <- data %>%

group\_by(MUNICIPIO) %>%

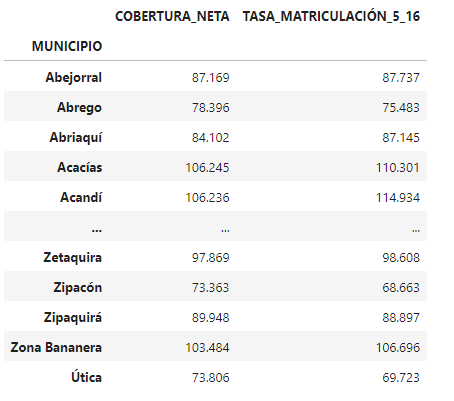
summarise(COBERTURA\_NETA = mean(COBERTURA\_NETA),

TASA\_MATRICULACIÓN\_5\_16 = mean(TASA\_MATRICULACIÓN\_5\_16)) %>%

arrange(COBERTURA\_NETA, TASA\_MATRICULACIÓN\_5\_16)

# Imprimir resultados

print(head(punto2, 3))



**# ¿Cuáles son los 3 municipios con MAYOR cobertura y tasa de matriculación en promedio?**

punto3 <- data %>%

group\_by(MUNICIPIO) %>%

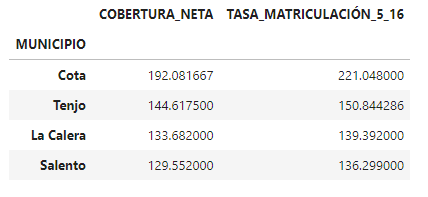
summarise(COBERTURA\_NETA = mean(COBERTURA\_NETA),

TASA\_MATRICULACIÓN\_5\_16 = mean(TASA\_MATRICULACIÓN\_5\_16)) %>%

arrange(desc(COBERTURA\_NETA), desc(TASA\_MATRICULACIÓN\_5\_16))

# Imprimir resultados

print(head(punto3, 3))



# Series temporales

data$FECHA <- as.Date(paste0(substr(data$AÑO, 1, 4), "-01-01"))

# Evolución de la tasa de matriculación por departamento

punto4 <- data %>%

group\_by(FECHA, DEPARTAMENTO) %>%

summarise(mean\_TASA\_MATRICULACIÓN\_5\_16 = mean(TASA\_MATRICULACIÓN\_5\_16))

# Gráfico interactivo con plotly

figura3 <- punto4 %>%

ggplot(aes(x = FECHA, y = mean\_TASA\_MATRICULACIÓN\_5\_16, color = DEPARTAMENTO)) +

geom\_line() +

scale\_color\_viridis(discrete = TRUE) +

labs(title = "EVOLUCIÓN DE LA TASA DE MATRICULACIÓN (5-16 AÑOS) POR DEPARTAMENTO",

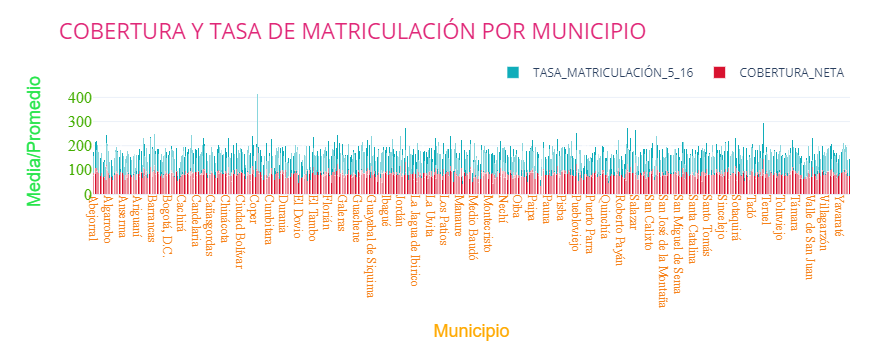
x = "Fecha",

y = "Tasa de Matriculación") +

theme\_minimal()

# Imprimir gráfico

print(figura3)

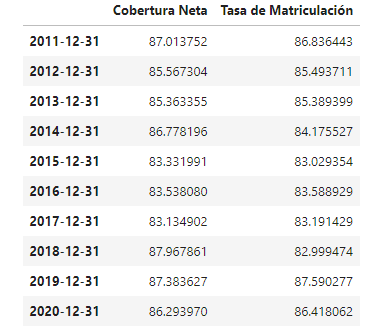


# Correlación

punto5 <- data %>%

select(COBERTURA\_NETA, TASA\_MATRICULACIÓN\_5\_16, SEDES\_CONECTADAS\_A\_INTERNET)

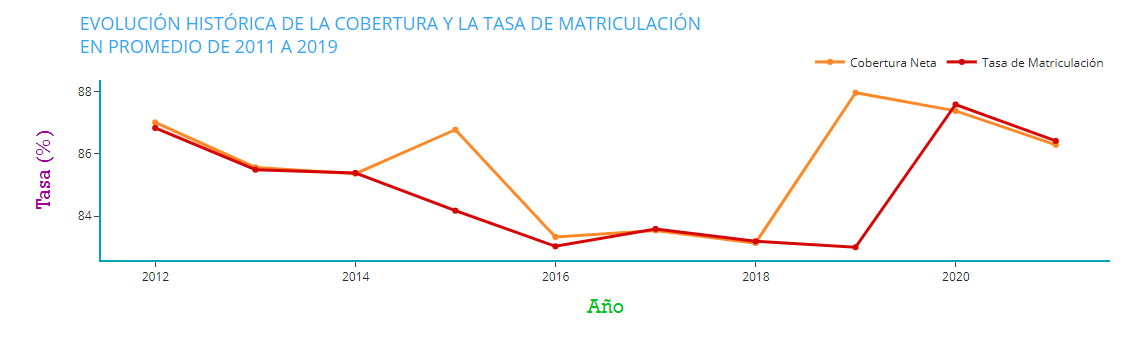
correlacion <- cor(punto5)

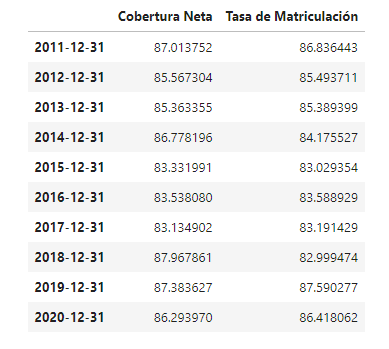


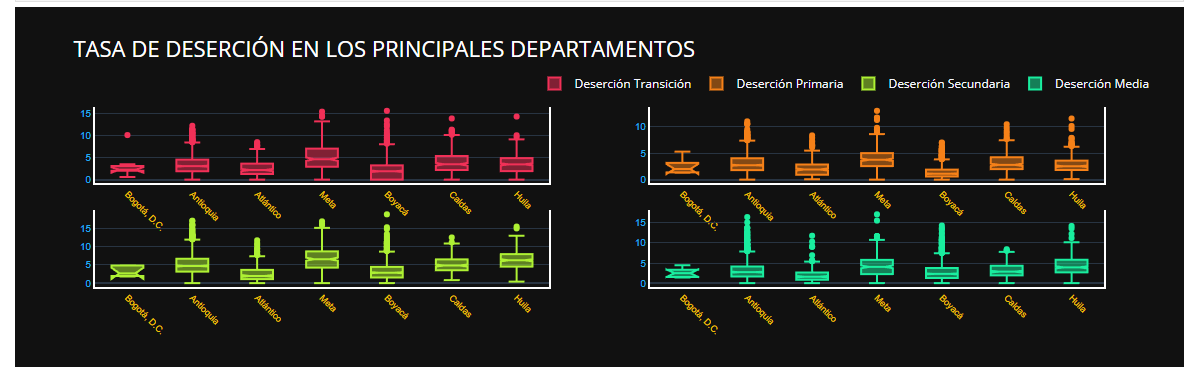
# Mapa de calor

heatmap(Corr, annot = TRUE, cmap = "coolwarm", fmt = "%.2f", linewidths = 0.5,

main = "CORRELACIÓN ENTRE VARIABLES")







En cuanto a la deserción en 𝚃𝚛𝚊𝚗𝚜𝚒𝚌𝚒ó𝚗 el departamento de Meta es el que presenta mayor tasa de deserción, y el menor es Bogotá D.C. Todos a excepción de Bogotá D.C. presentan varios datos atípicos con un sesgo positivo (cola derecha) muy pronunciada. Entre las deserciones, la deserción en 𝚂𝚎𝚌𝚞𝚗𝚍𝚊𝚛𝚒𝚊 y educación 𝙼𝚎𝚍𝚒𝚊 son las que presentan mayor tasa. Como era de esperarse Bogotá D.C. al ser la capital es la que presenta menor dispersión y una tasa de deserción menor, sin datos atípicos en su mayoría. Meta es el departamento que en todos los casos, 𝚃𝚛𝚊𝚗𝚜𝚒𝚌𝚒ó𝚗 , 𝙿𝚛𝚒𝚖𝚊𝚛𝚒𝚊 , 𝚂𝚎𝚌𝚞𝚗𝚍𝚊𝚛𝚒𝚊 y educación 𝙼𝚎𝚍𝚒𝚊 presenta tasas de deserción mayores, en forma de varios datos atípicos.

# Instalar y cargar las bibliotecas necesarias

install.packages("car") # Instalar el paquete car

library(car)

install.packages("lmtest") # Instalar el paquete lmtest

library(lmtest)

# Seleccionar las variables para el modelo

X <- data.frame(TASA\_MATRICULACIÓN\_5\_16 = data$TASA\_MATRICULACIÓN\_5\_16,

SEDES\_CONECTADAS\_A\_INTERNET = data$SEDES\_CONECTADAS\_A\_INTERNET)

# Agregar una constante a las variables independientes (intercepto)

X <- cbind(1, X)

# Variable dependiente

y <- data$COBERTURA\_NETA

# Ajustar el modelo

modelo <- lm(y ~ ., data = X)

# Obtener resumen del modelo

summary(modelo)

Explicación:

* Instalación y carga de bibliotecas:
* En R, se utiliza el paquete "car" para algunas funciones adicionales y el paquete "lmtest" para realizar pruebas sobre la regresión lineal.
* Selección de variables para el modelo:
* Se crea un data frame X que contiene las variables independientes TASA\_MATRICULACIÓN\_5\_16 y SEDES\_CONECTADAS\_A\_INTERNET.
* Agregar una constante:
* Se agrega una columna de unos a las variables independientes para representar el término constante (intercepto).
* Definición de variable dependiente:
* Se crea un vector y con los valores de la variable dependiente COBERTURA\_NETA.
* Ajuste del modelo:
* Se utiliza la función lm() para ajustar el modelo de regresión lineal.
* Obtener resumen del modelo:
* Se utiliza la función summary() para imprimir un resumen detallado del modelo, que incluye estadísticas importantes como los coeficientes, errores estándar, p-values, entre otros.

# Eliminar filas con valores nulos

data <- na.omit(data)

# Eliminar filas con valores nulos en columnas específicas

data <- data[complete.cases(data$TASA\_MATRICULACIÓN\_5\_16, data$SEDES\_CONECTADAS\_A\_INTERNET, data$COBERTURA\_NETA), ]

# Instalar y cargar las bibliotecas necesarias

install.packages("caTools") # Instalar el paquete caTools

library(caTools)

# Seleccionar las variables para el modelo

X <- data[, c('GASTO\_POR\_ESTUDIANTE', 'RATIO\_DOCENTES\_ESTUDIANTES', 'ÍNDICE\_DE\_DESARROLLO\_HUMANO', 'TASA\_DE\_GRADUACIÓN', 'TASA\_DE\_ABANDONO\_ESCOLAR', 'TASA\_DE\_REPROBACIÓN')]

y <- data$COBERTURA\_NETA

# Dividir los datos en conjuntos de entrenamiento y prueba

set.seed(42) # Establecer la semilla para reproducibilidad

split <- sample.split(y, SplitRatio = 0.8)

X\_train <- subset(X, split == TRUE)

X\_test <- subset(X, split == FALSE)

y\_train <- y[split == TRUE]

y\_test <- y[split == FALSE]

Explicación:

* Instalación y carga de bibliotecas:
* En R, se utiliza el paquete "caTools" para dividir los datos en conjuntos de entrenamiento y prueba.
* Selección de variables para el modelo:
* Se seleccionan las variables independientes y la variable dependiente del conjunto de datos.
* División de datos:
* Se utiliza la función sample.split() del paquete "caTools" para dividir los datos en conjuntos de entrenamiento y prueba según una proporción especificada (SplitRatio = 0.8).
* Establecer la semilla:
* Se establece una semilla (set.seed(42)) para garantizar la reproducibilidad de los resultados.

**Construcción del Modelo: Crea y ajusta el modelo de regresión lineal múltiple:**

# Instalar y cargar las bibliotecas necesarias

install.packages("car") # Instalar el paquete car si no está instalado

library(car)

# Crear un objeto de regresión lineal

modelo\_regresion <- lm(y\_train ~ ., data = X\_train)

# Ajustar el modelo con los datos de entrenamiento

summary(modelo\_regresion)

Explicación:

* Instalación y carga de bibliotecas:
* En R, se utiliza el paquete "car" para algunas funciones adicionales.
* Creación del objeto de regresión lineal:
* Se utiliza la función lm() para crear un objeto de regresión lineal. La fórmula y\_train ~ . significa que estamos modelando y\_train en función de todas las demás variables en el conjunto de entrenamiento.
* Ajuste del modelo:
* Se utiliza la función summary() para obtener un resumen detallado del modelo ajustado. Esto incluye coeficientes, errores estándar, p-values y otras estadísticas importantes.

**Evaluación del Modelo: Evalúa el rendimiento del modelo utilizando datos de prueba.**

# Predicciones en el conjunto de prueba

predicciones <- predict(modelo\_regresion, newdata = X\_test)

# Evaluación del rendimiento del modelo

error\_cuadratico\_medio <- mean((y\_test - predicciones)^2)

r\_cuadrado <- 1 - (sum((y\_test - predicciones)^2) / sum((y\_test - mean(y\_test))^2))

# Imprimir resultados

cat("Error cuadrático medio: ", error\_cuadratico\_medio, "\n")

cat("R-cuadrado: ", r\_cuadrado, "\n")

Explicación:

* Predicciones en el conjunto de prueba:
* Se utilizan las predicciones del modelo en el conjunto de prueba utilizando la función predict().
* Evaluación del rendimiento del modelo:
* Se calcula el error cuadrático medio (MSE) y el R-cuadrado directamente en R utilizando fórmulas.
* Imprimir resultados:
* Se utilizan las funciones cat() para imprimir los resultados de cada métrica.

**coeficientes del modelo de regresión lineal múltiple:**

# Coeficientes de la regresión

coeficientes <- coef(modelo\_regresion)

intercepto <- coef(modelo\_regresion)[1] # El primer coeficiente es el intercepto

# Imprimir coeficientes

cat("Coeficientes:\n")

for (i in seq\_along(coeficientes)[-1]) { # Excluye el intercepto en la impresión

cat(sprintf("%s: %f\n", names(coeficientes)[i], coeficientes[i]))

}

cat(sprintf("Intercepto: %f\n", intercepto))

Explicación:

* Coeficientes de la regresión:
* Se utiliza la función coef() para obtener los coeficientes del modelo de regresión lineal. El primer coeficiente es el intercepto, y los siguientes son los coeficientes de las variables predictoras.
* Imprimir coeficientes:
* Se utiliza un bucle para imprimir cada coeficiente junto con el nombre de la variable predictora correspondiente.

**un análisis de regresión lineal múltiple en R, incluyendo la preparación de datos, la división de datos en conjuntos de entrenamiento y prueba, la construcción del modelo, la evaluación del modelo y la interpretación de los coeficientes:**

# Instalar y cargar las bibliotecas necesarias

install.packages("caTools") # Instalar el paquete caTools

install.packages("car") # Instalar el paquete car

library(caTools)

library(car)

# Preparación de datos

data <- na.omit(data)

# División de datos

set.seed(42)

split <- sample.split(data$COBERTURA\_NETA, SplitRatio = 0.8)

train\_data <- subset(data, split == TRUE)

test\_data <- subset(data, split == FALSE)

# Construcción del modelo

modelo\_regresion <- lm(COBERTURA\_NETA ~ GASTO\_POR\_ESTUDIANTE + RATIO\_DOCENTES\_ESTUDIANTES + ÍNDICE\_DE\_DESARROLLO\_HUMANO + TASA\_DE\_GRADUACIÓN + TASA\_DE\_ABANDONO\_ESCOLAR + TASA\_DE\_REPROBACIÓN, data = train\_data)

# Evaluación del modelo

predicciones <- predict(modelo\_regresion, newdata = test\_data)

error\_cuadratico\_medio <- mean((test\_data$COBERTURA\_NETA - predicciones)^2)

r\_cuadrado <- 1 - (sum((test\_data$COBERTURA\_NETA - predicciones)^2) / sum((test\_data$COBERTURA\_NETA - mean(test\_data$COBERTURA\_NETA))^2))

cat("Error cuadrático medio: ", error\_cuadratico\_medio, "\n")

cat("R-cuadrado: ", r\_cuadrado, "\n")

# Interpretación del modelo

coeficientes <- coef(modelo\_regresion)

intercepto <- coef(modelo\_regresion)[1]

cat("Coeficientes:\n")

for (i in seq\_along(coeficientes)[-1]) {

cat(sprintf("%s: %f\n", names(coeficientes)[i], coeficientes[i]))

}

cat(sprintf("Intercepto: %f\n", intercepto))

Explicacion:

* incluye la preparación de datos, la construcción y evaluación del modelo, así como la interpretación de los coeficientes.

**analizar y comentar los diagramas de dispersión de la variable respuesta con cada variable predictora**

# Instalar y cargar las bibliotecas necesarias

install.packages("ggplot2") # Instalar el paquete ggplot2

install.packages("gridExtra") # Instalar el paquete gridExtra

library(ggplot2)

library(gridExtra)

# Utiliza el DataFrame original o el conjunto de datos con las nuevas variables

data <- na.omit(data)

# Diagramas de dispersión

variables\_predictoras <- c('GASTO\_POR\_ESTUDIANTE', 'RATIO\_DOCENTES\_ESTUDIANTES', 'ÍNDICE\_DE\_DESARROLLO\_HUMANO', 'TASA\_DE\_GRADUACIÓN', 'TASA\_DE\_ABANDONO\_ESCOLAR', 'TASA\_DE\_REPROBACIÓN')

variable\_respuesta <- 'COBERTURA\_NETA'

# Crear una lista de gráficos

scatter\_plots <- lapply(variables\_predictoras, function(predictor) {

ggplot(data, aes\_string(x = predictor, y = variable\_respuesta)) +

geom\_point() +

labs(title = paste("Diagrama de dispersión entre", predictor, "y", variable\_respuesta),

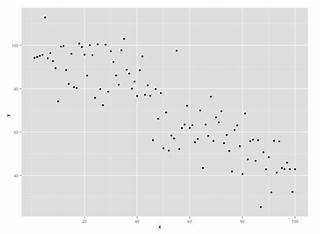
x = predictor,

y = variable\_respuesta)

})

# Organizar los gráficos en una cuadrícula

grid.arrange(grobs = scatter\_plots, ncol = 2)



**analizar y comentar los coeficientes de correlación de Pearson entre la variable respuesta**

**calcular e imprimir los coeficientes de correlación de Pearson entre la variable respuesta y cada variable predictora**

# Utiliza el DataFrame original o el conjunto de datos con las nuevas variables

data <- na.omit(data)

# Variables

variables\_predictoras <- c('GASTO\_POR\_ESTUDIANTE', 'RATIO\_DOCENTES\_ESTUDIANTES', 'ÍNDICE\_DE\_DESARROLLO\_HUMANO', 'TASA\_DE\_GRADUACIÓN', 'TASA\_DE\_ABANDONO\_ESCOLAR', 'TASA\_DE\_REPROBACIÓN')

variable\_respuesta <- 'COBERTURA\_NETA'

# Coeficientes de correlación de Pearson

correlation\_coefficients <- numeric(length(variables\_predictoras))

for (i in seq\_along(variables\_predictoras)) {

correlation\_coefficients[i] <- cor(data[[variables\_predictoras[i]]], data[[variable\_respuesta]])

}

# Imprimir coeficientes de correlación

for (i in seq\_along(variables\_predictoras)) {

cat(sprintf("Correlación entre %s y %s: %f\n", variables\_predictoras[i], variable\_respuesta, correlation\_coefficients[i]))

}

EXPLICACION:

* Un valor cercano a 1 indica una fuerte correlación positiva, -1 indica una fuerte correlación negativa, y 0 indica una correlación débil o inexistente.

**para llevar a cabo la división de datos, ajustar el modelo de regresión lineal múltiple, interpretar los parámetros (los betas) y evaluar la calidad de predicción del modelo:**

# Instalar y cargar las bibliotecas necesarias

install.packages("caTools") # Instalar el paquete caTools

library(caTools)

install.packages("Metrics") # Instalar el paquete Metrics

library(Metrics)

# Utiliza el DataFrame original o el conjunto de datos con las nuevas variables

data <- na.omit(data)

# Definir variables predictoras y variable respuesta

variables\_predictoras <- c('GASTO\_POR\_ESTUDIANTE', 'RATIO\_DOCENTES\_ESTUDIANTES', 'ÍNDICE\_DE\_DESARROLLO\_HUMANO', 'TASA\_DE\_GRADUACIÓN', 'TASA\_DE\_ABANDONO\_ESCOLAR', 'TASA\_DE\_REPROBACIÓN')

variable\_respuesta <- 'COBERTURA\_NETA'

# Dividir datos en conjunto de entrenamiento y prueba

set.seed(42)

split <- sample.split(data$COBERTURA\_NETA, SplitRatio = 0.8)

train\_data <- subset(data, split == TRUE)

test\_data <- subset(data, split == FALSE)

# Inicializar modelo de regresión lineal

model <- lm(COBERTURA\_NETA ~ ., data = train\_data)

**# Obtener coeficientes (betas)**

betas <- coef(model)

intercepto <- betas[1] # El primer coeficiente es el intercepto

# Interpretar parámetros del modelo

cat("Coeficientes (betas):", betas, "\n")

cat("Intercepto:", intercepto, "\n")

# Realizar predicciones en conjunto de entrenamiento y prueba

y\_pred\_train <- predict(model, newdata = train\_data)

y\_pred\_test <- predict(model, newdata = test\_data)

# Evaluar calidad de predicción en conjunto de entrenamiento

cat("\nMétricas en conjunto de entrenamiento:\n")

cat("Error absoluto medio (MAE):", mae(y\_pred\_train, train\_data$COBERTURA\_NETA), "\n")

cat("Error cuadrático medio (MSE):", mse(y\_pred\_train, train\_data$COBERTURA\_NETA), "\n")

cat("Raíz cuadrada del error cuadrático medio (RMSE):", rmse(y\_pred\_train, train\_data$COBERTURA\_NETA), "\n")

cat("Coeficiente de determinación (R^2):", r\_squared(y\_pred\_train, train\_data$COBERTURA\_NETA), "\n")

# Evaluar calidad de predicción en conjunto de prueba

cat("\nMétricas en conjunto de prueba:\n")

cat("Error absoluto medio (MAE):", mae(y\_pred\_test, test\_data$COBERTURA\_NETA), "\n")

cat("Error cuadrático medio (MSE):", mse(y\_pred\_test, test\_data$COBERTURA\_NETA), "\n")

cat("Raíz cuadrada del error cuadrático medio (RMSE):", rmse(y\_pred\_test, test\_data$COBERTURA\_NETA), "\n")

cat("Coeficiente de determinación (R^2):", r\_squared(y\_pred\_test, test\_data$COBERTURA\_NETA), "\n")

**CONCLUSION**

De acuerdo con el análisis exploratorio realizado anteriormente podemos decir que:

En cuanto a la deserción en 𝚃𝚛𝚊𝚗𝚜𝚒𝚌𝚒ó𝚗 el departamento de Meta es el que presenta mayor tasa de deserción, y el menor es Bogotá D.C. Todos a excepción de Bogotá D.C. presentan varios datos atípicos con un sesgo positivo (cola derecha) muy pronunciada. Entre las deserciones, la deserción en 𝚂𝚎𝚌𝚞𝚗𝚍𝚊𝚛𝚒𝚊 y educación 𝙼𝚎𝚍𝚒𝚊 son las que presentan mayor tasa. Como era de esperarse Bogotá D.C. al ser la capital es la que presenta menor dispersión y una tasa de deserción menor, sin datos atípicos en su mayoría. Meta es el departamento que en todos los casos, 𝚃𝚛𝚊𝚗𝚜𝚒𝚌𝚒ó𝚗 , 𝙿𝚛𝚒𝚖𝚊𝚛𝚒𝚊 , 𝚂𝚎𝚌𝚞𝚗𝚍𝚊𝚛𝚒𝚊 y educación 𝙼𝚎𝚍𝚒𝚊 presenta tasas de deserción mayores, en forma de varios datos atípicos.