Evaluacion 1

Jorge Cáceres Barrales

2024-06-18

Objetivo de Investigación

Analizar los datos de permisos de circulación vehicular en Calbuco para identificar patrones y tendencias, y proponer mejoras en la gestión de permisos.

Introducción

Descripción del Conjunto de Datos

El conjunto de datos seleccionado para este análisis proviene de los registros de permisos de circulación en la Municipalidad de Calbuco. Este conjunto de datos contiene información detallada sobre vehículos y sus permisos de circulación, incluyendo variables como el tipo de vehículo, año de fabricación, marca, modelo, color, tipo de combustible, y valor del permiso, entre otros.

Objetivos del Análisis

El objetivo principal de este análisis es investigar y comprender los patrones y tendencias en los permisos de circulación vehicular en Calbuco. Específicamente, se busca:

1. Identificar las características más comunes de los vehículos que obtienen permisos de circulación.
2. Analizar la distribución temporal de la obtención de permisos.
3. Explorar posibles relaciones entre las características de los vehículos y el costo del permiso.
4. Realizar un modelo que permita estimar la cantidad de pagos que se recibiran en un periodo de tiempo

Procesamiento de Datos

El preprocesamiento de datos es un paso crucial para garantizar que los datos estén limpios y listos para el análisis. A continuación, se detallan los pasos tomados para limpiar y preparar los datos

Carga Librerias

```
# Cargar librerías necesarias
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
```

```
##     intersect, setdiff, setequal, union
```

```
library(stringr)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(e1071)
library(xgboost)
```

```
##
## Attaching package: 'xgboost'
```

```
## The following object is masked from 'package:dplyr':
##
##     slice
```

```
library(ROSE)
```

```
## Loaded ROSE 0.0-4
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
library(summarytools)
library(httr)
```

```
##
## Attaching package: 'httr'
```

```
## The following object is masked from 'package:caret':
##
##     progress
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

Cargas funciones y base de datos

```
# Enlace raw al archivo de funciones en GitHub
source("https://raw.githubusercontent.com/jkcrs1/R/main/funciones.R")
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

##
## Attaching package: 'palette'

## The following object is masked from 'package:grDevices':
##
##     palette

##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

## Loading required package: Rcpp

## Loading required package: rlang

##
## Attaching package: 'tibble'

## The following object is masked from 'package:summarytools':
##
##     view

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
# Permisos circulación  pagados en la municipalidad de calbuco
url3 <- "https://github.com/jkcrs1/R/raw/main/permiso_circulacion_calbuco.csv"

# Prueba con diferentes codificaciones
permiso <- read.csv(url3, sep = ";", fileEncoding = "ISO-8859-1")
```

Tipos de Variables

Utilizando el paquete summarytools, se genera un resumen detallado de todas las variables del dataframe. Este informe incluye estadísticas descriptivas, frecuencias, gráficos de barras para variables categóricas y otras visualizaciones útiles para el análisis de datos.

```r
# Configurar el entorno de summarytools para HTML
st_options(style = "rmarkdown", plain.ascii = FALSE)

# Obtener un resumen de los datos utilizando summarytools y renderizar en HTML
dfSummary(permiso) %>%
  print(method = 'render')
```

Normalización de las Variables

Se procede a normalizar y limpiar el conjunto de datos, asegurando que las variables estén en el formato correcto y que los valores nulos se manejen adecuadamente.

```r
# Convertir el dataframe
permiso <- data.frame(permiso)

# Cambio de nombres de columnas
colnames(permiso) <- colnames(permiso) %>% str_replace_all("\\.", "_")

# Normalización de datos
tipo_vehiculo_mapeo <- c(
  "AMBULANCIA" = "AMBULANCIA", "AUTOMOVIL" = "AUTOMOVIL", "BUS" = "BUS",
  "Cabriolet" = "AUTOMOVIL", "CAMION" = "CAMION", "CAMIONETA" = "CAMIONETA",
  "CARRO ARRASTRE A" = "REMOLQUE", "CARRO BOMBA" = "CARRO BOMBA",
  "CASA RODANTE" = "CASA RODANTE", "Comercial" = "COMERCIAL", "CUATRIMOTO" = "CUATRIMOTO",
  "FURGON" = "FURGON", "GRUA" = "MAQUINA PESADA", "Hatchback" = "AUTOMOVIL",
  "JEEP" = "AUTOMOVIL", "MAQUINA INDUSTRIAL" = "MAQUINA INDUSTRIAL",
  "MINIBUS" = "MINIBUS", "MINIBUS ESCOLAR" = "MINIBUS", "MINIBUS PARTICULAR" = "MINIBUS",
  "MINIBUS PRIVADO" = "MINIBUS", "MINIBUS TURISMO" = "MINIBUS", "MOTO" = "MOTOCICLETA",
  "MOTOCICLETA" = "MOTOCICLETA", "OTROS" = "OTROS", "REMOLQUE A" = "REMOLQUE",
  "REMOLQUE B" = "REMOLQUE", "RETROEXCAVADORA" = "MAQUINA PESADA", "Sedan" = "AUTOMOVIL",
  "SEMI REMOLQUE" = "REMOLQUE", "STATION WAGON" = "AUTOMOVIL", "SUV" = "SUV",
  "TAXI EJECUTIVO" = "TAXI", "TAXI BASICO" = "TAXI", "TAXI COLECTIVO" = "TAXI",
  "TRACTOCAMION" = "CAMION", "TRACTOR" = "TRACTOR", "VAN" = "VAN"
)

tipo_combustible_mapeo <- c(
  "Benc" = "Bencina", "Dies" = "Diesel", "NULL" = "NULL",
  "DUAL" = "Hibrido", "Hibr" = "Hibrido", "Elec" = "Electrico"
)

transmision_mapeo <- c(
  "Mec" = "Mecanica", "Aut" = "Automatica", "NULL" = "NULL",
  "CVT" = "Automatica", "DCT" = "Automatica"
)


# Aplicar todas las transformaciones y normalizaciones
permiso <- permiso %>%
  mutate(
    Tipo_Vehiculo = recode(Tipo_Vehiculo, !!!tipo_vehiculo_mapeo),
    Tipo_Combustible = recode(Tipo_Combustible, !!!tipo_combustible_mapeo),
    Transmision = recode(Transmision, !!!transmision_mapeo),
    across(everything(), reemplazar_nulos),
    across(c(Municipalidad, Grupo_Vehiculo, Placa, Digito, Codigo_SII,
```

```
            Forma_Pago, Tipo_Vehiculo, Marca, Modelo, Color,
            Transmision, Tipo_Combustible, Equipamiento), str_to_title)
  ) %>%
  mutate(
    Ano_Vehiculo = as.integer(Ano_Vehiculo),
    Valor_Multa = suppressWarnings(as.numeric(as.character(Valor_Multa))),
    Valor_Neto = suppressWarnings(as.numeric(as.character(Valor_Neto))),
    Valor_Pagado = suppressWarnings(as.numeric(as.character(Valor_Pagado))),
    Fecha_Pago = as.Date(Fecha_Pago, format = "%d-%m-%y"),
    Ano_Pago = year(Fecha_Pago),
    Mes_Pago_texto = month(Fecha_Pago, label = TRUE),
    Mes_Pago = month(Fecha_Pago)
  ) %>%
  filter(!is.na(Valor_Pagado) & !is.na(Fecha_Pago) & Valor_Pagado > 0) %>%
  distinct()
```

Traformación de variables categoricas a Factor

Para asegurar un análisis e interpretación adecuados de los datos, las variables categóricas se transforman a factores.

```
# Transformar las variables a factor con los niveles definidos
permiso <- permiso %>%
  mutate(across(where(is.character), as.factor))
```

Análisis Exploratorio de Datos (EDA)

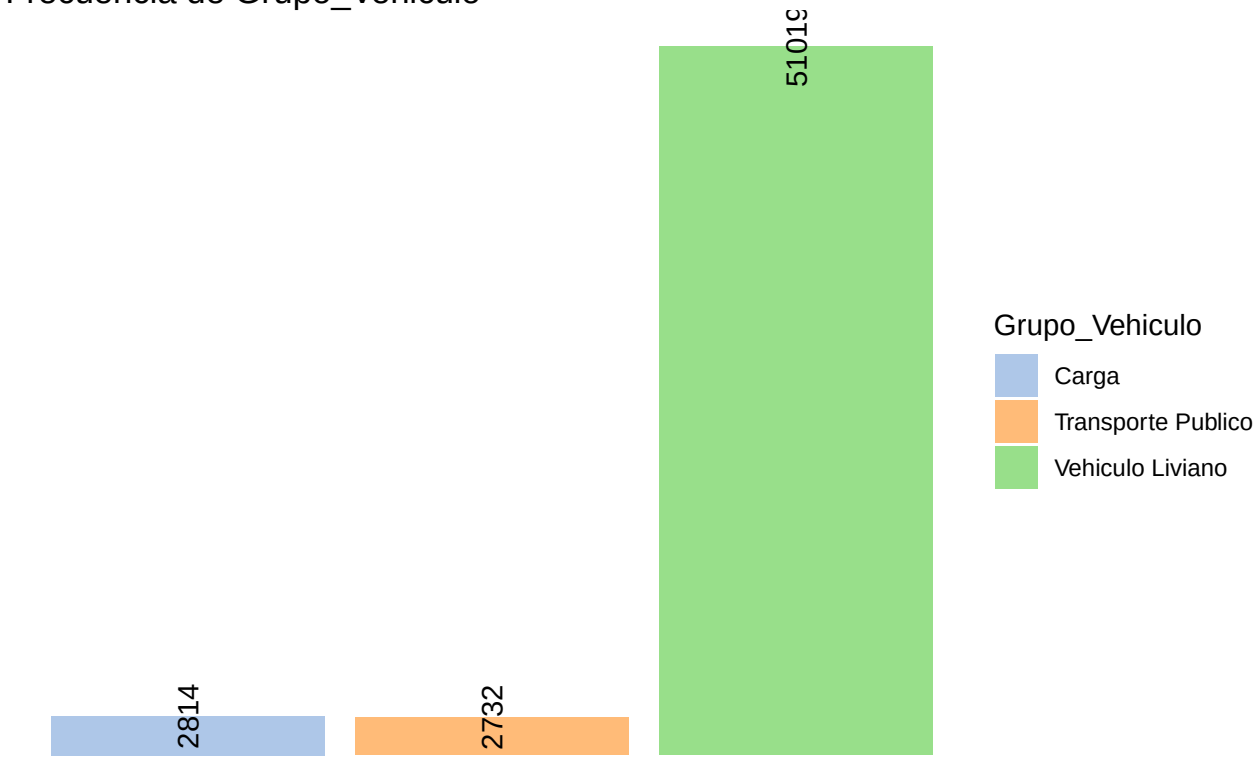
Seleccionar variables relevantes

```
permiso_relevante <-  permiso %>%
  select(Grupo_Vehiculo, Ano_Vehiculo,Tipo_de_Pago,Fecha_Pago,Ano_Pago,Mes_Pago_texto,Mes_Pago,Valor_Ne
```

Visualización de la Distribución de la Variable “Grupo Vehiculo”

Observar la distribución de la variable “Grupo Vehiculo” en el conjunto de datos original para entender mejor la composición del mismo.

```
# Visualización de la distribución de la variable "Grupo Vehiculo"
var_x <- "Grupo_Vehiculo"
var_tipo_grafico <- "barra"

grafico(
  data = permiso_relevante,
  var  = var_x,
  tipo_grafico = var_tipo_grafico
)
```



Este gráfico de barras muestra la frecuencia de Grupo_Vehiculo .

Muestreo

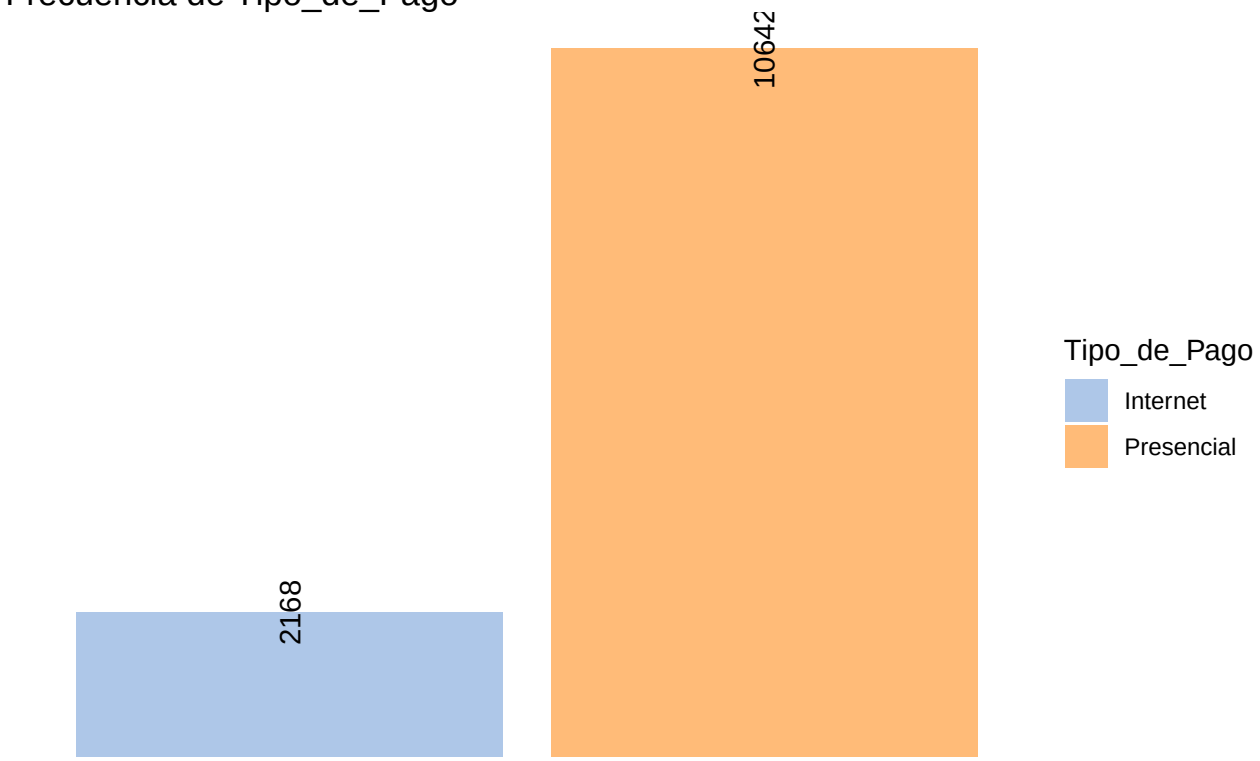
Para asegurar una representación adecuada de cada grupo, realizaremos un muestreo estratificado de los datos. Este proceso garantiza que cada estrato (grupo) esté proporcionalmente representado en la muestra.

```
# Calcular la muestra aleatoria según Desviación Estándar
cant <- nrow(permiso_relevante)
sd <- sd(permiso_relevante$Valor_Pagado)
n <- tam.muestra(alfa = 0.05, epsilon = 1200, s = sd, N = cant)
set.seed(2)
cant <- sample(nrow(permiso_relevante), n)
permiso_muestra <- permiso_relevante[cant, ]

cat("La cantidad de registros de muestra es:", nrow(permiso_muestra))
```

```
## La cantidad de registros de muestra es: 12810
```

Variable: Tipo de Pago

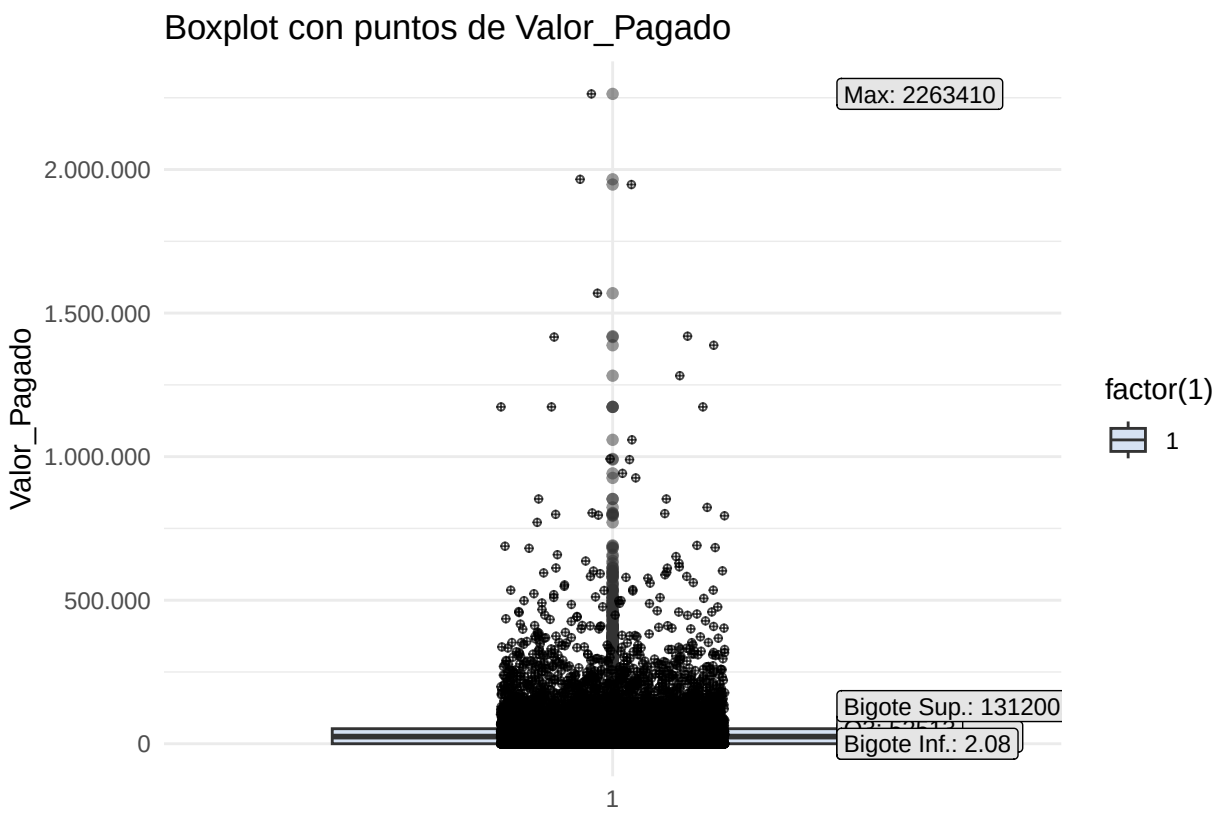
```
var_x <- "Tipo_de_Pago"
var_tipo_grafico <- "barra"

grafico(
  data = permiso_muestra,
  var  = var_x,
  tipo_grafico = var_tipo_grafico
)
```



Este gráfico de barras muestra la frecuencia de Tipo_de_Pago .

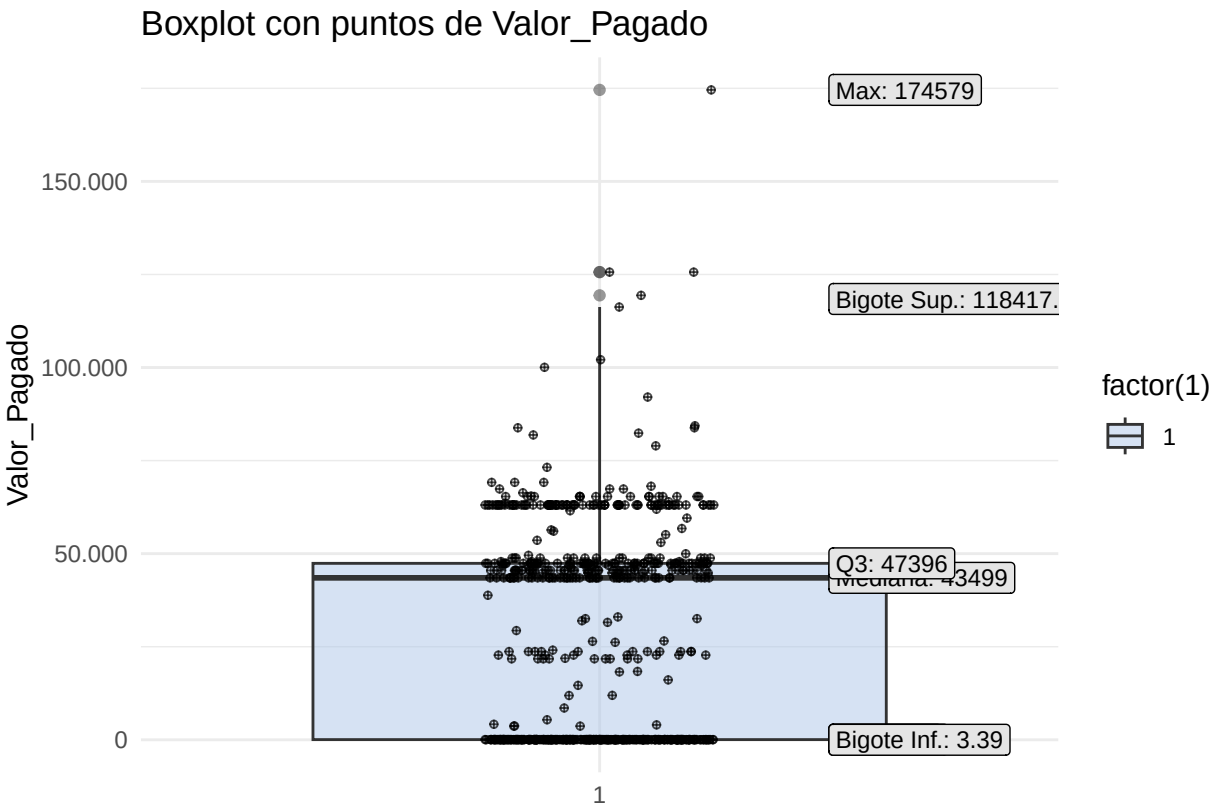
Variable: Valor Pagado

```
vehiculo_liviano_dataset <- permiso_muestra %>% filter(`Grupo_Vehiculo` == "Vehiculo Liviano")
transporte_publico_dataset <- permiso_muestra %>% filter(`Grupo_Vehiculo` == "Transporte Publico")
carga_dataset <- permiso_muestra %>% filter(`Grupo_Vehiculo` == "Carga")



var_x <- "Valor_Pagado"
var_tipo_grafico <- "boxplot"

grafico(
  data = vehiculo_liviano_dataset,
  var  = var_x,
  tipo_grafico = var_tipo_grafico
)
```



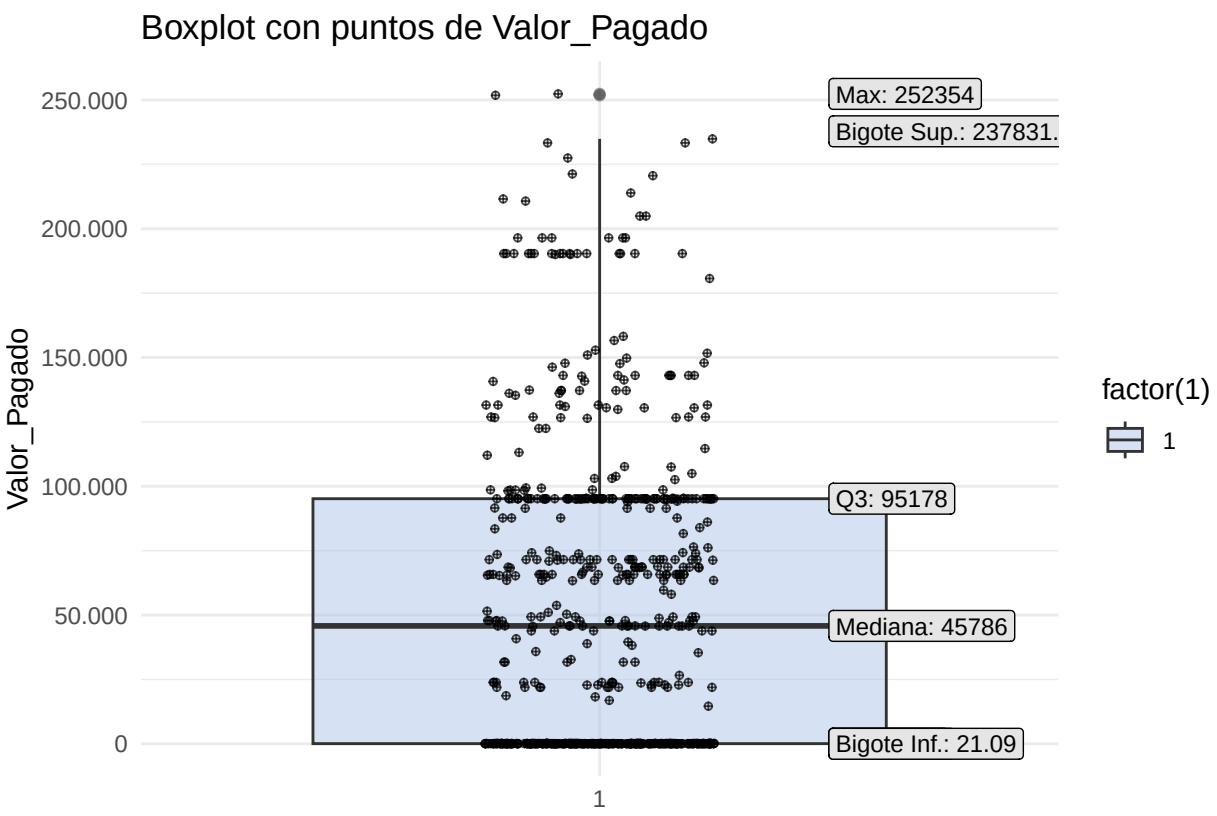
Este boxplot con puntos muestra la distribución de Valor_Pagado .

```
grafico(
  data = transporte_publico_dataset,
  var  = var_x,
  tipo_grafico = var_tipo_grafico
)
```



Este boxplot con puntos muestra la distribución de Valor_Pagado .

```
grafico(
  data = carga_dataset,
  var  = var_x,
  tipo_grafico = var_tipo_grafico
)
```



Este boxplot con puntos muestra la distribución de Valor_Pagado .

Tratamiento de Outliers

```
# Tratamiento de outliers
# Calcular el IQR y los límites para identificar valores atípicos
IQR_Valor_Pagado <- IQR(permiso_muestra$Valor_Pagado, na.rm = TRUE)
Q1 <- quantile(permiso_muestra$Valor_Pagado, 0.25, na.rm = TRUE)
Q3 <- quantile(permiso_muestra$Valor_Pagado, 0.75, na.rm = TRUE)
lower_bound <- Q1 - 1.5 * IQR_Valor_Pagado
upper_bound <- Q3 + 1.5 * IQR_Valor_Pagado

# Filtrar los valores atípicos
outliers <- permiso_muestra %>%
  filter(Valor_Pagado < lower_bound | Valor_Pagado > upper_bound)

# Eliminar los valores atípicos
permiso_sin_outliers <- permiso_muestra %>%
  filter(Valor_Pagado >= lower_bound & Valor_Pagado <= upper_bound)

cat("Número de registros originales:", nrow(permiso_muestra), "\n")
```

```
## Número de registros originales: 12810
```

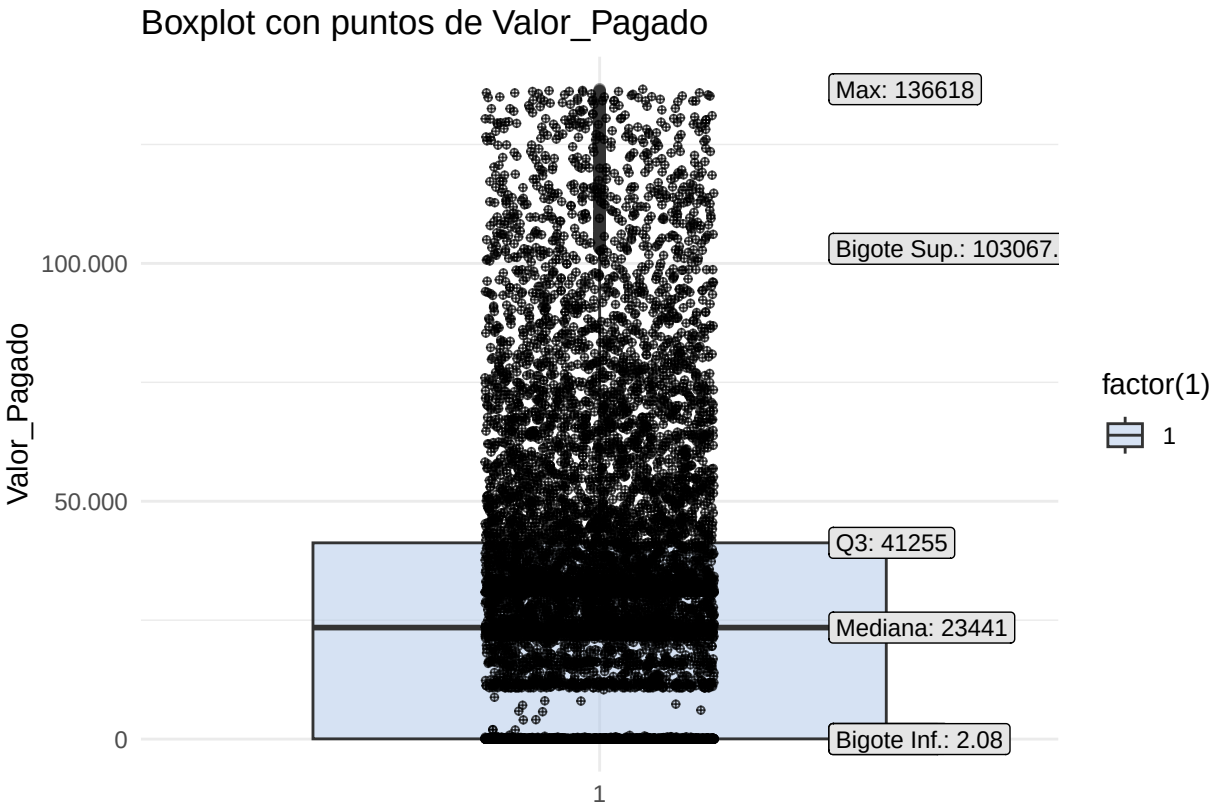
```
cat("Número de registros sin outliers:", nrow(permiso_sin_outliers), "\n")
```

```
## Número de registros sin outliers: 11850
```

```
permiso_muestra<- permiso_sin_outliers
```

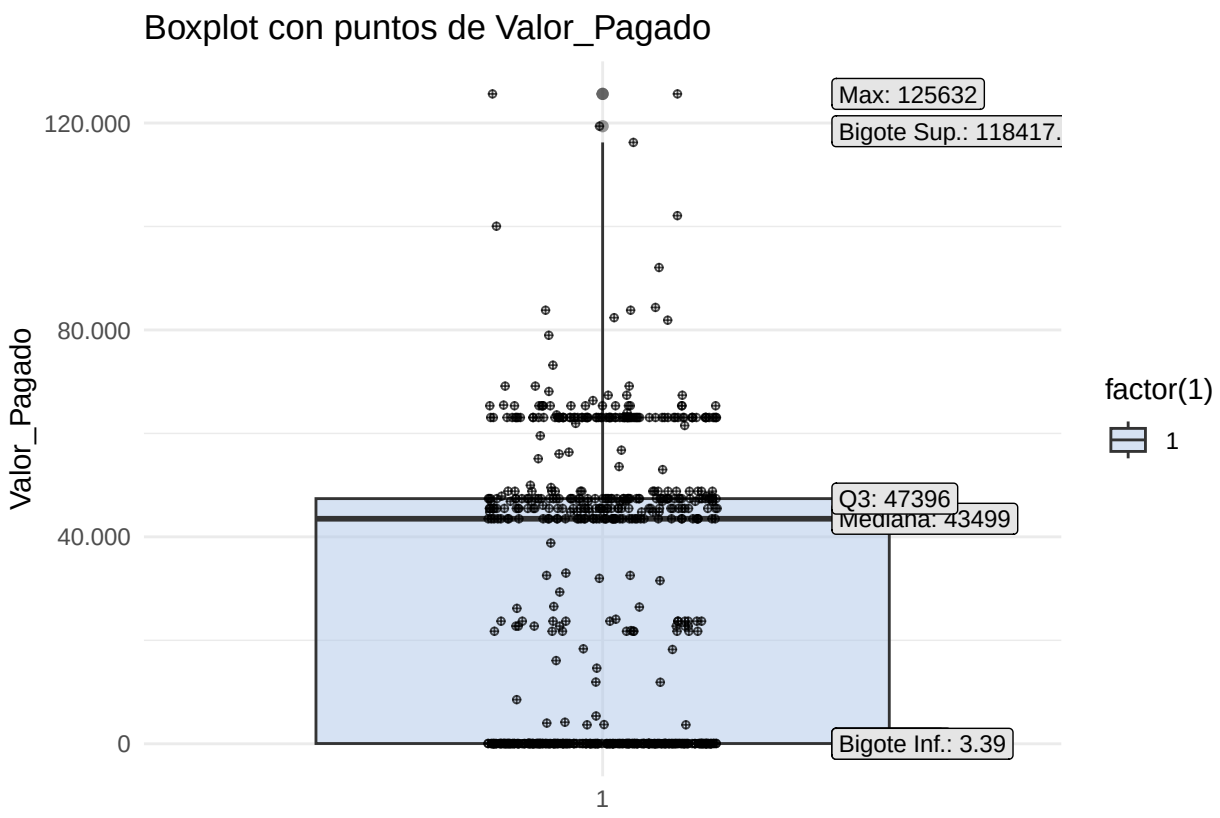
Variable: Valor Pagado sin outliers

```
vehiculo_liviano_dataset <- permiso_muestra %>% filter(`Grupo_Vehiculo` == "Vehiculo Liviano")
transporte_publico_dataset <- permiso_muestra %>% filter(`Grupo_Vehiculo` == "Transporte Publico")
carga_dataset <- permiso_muestra %>% filter(`Grupo_Vehiculo` == "Carga")



var_x <- "Valor_Pagado"
var_tipo_grafico <- "boxplot"

grafico(
  data = vehiculo_liviano_dataset,
  var  = var_x,
  tipo_grafico = var_tipo_grafico
)
```



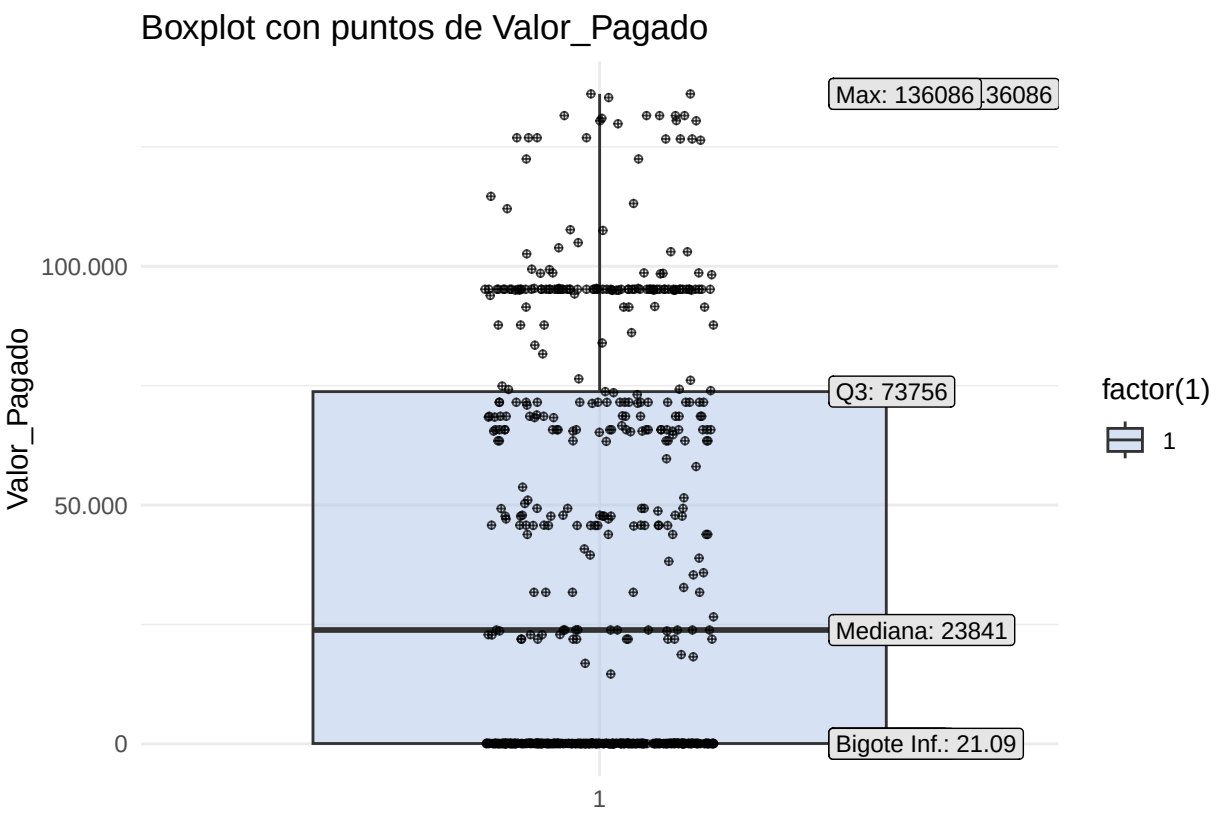
Este boxplot con puntos muestra la distribución de Valor_Pagado .

```
grafico(
  data = transporte_publico_dataset,
  var  = var_x,
  tipo_grafico = var_tipo_grafico
)
```



Este boxplot con puntos muestra la distribución de Valor_Pagado .

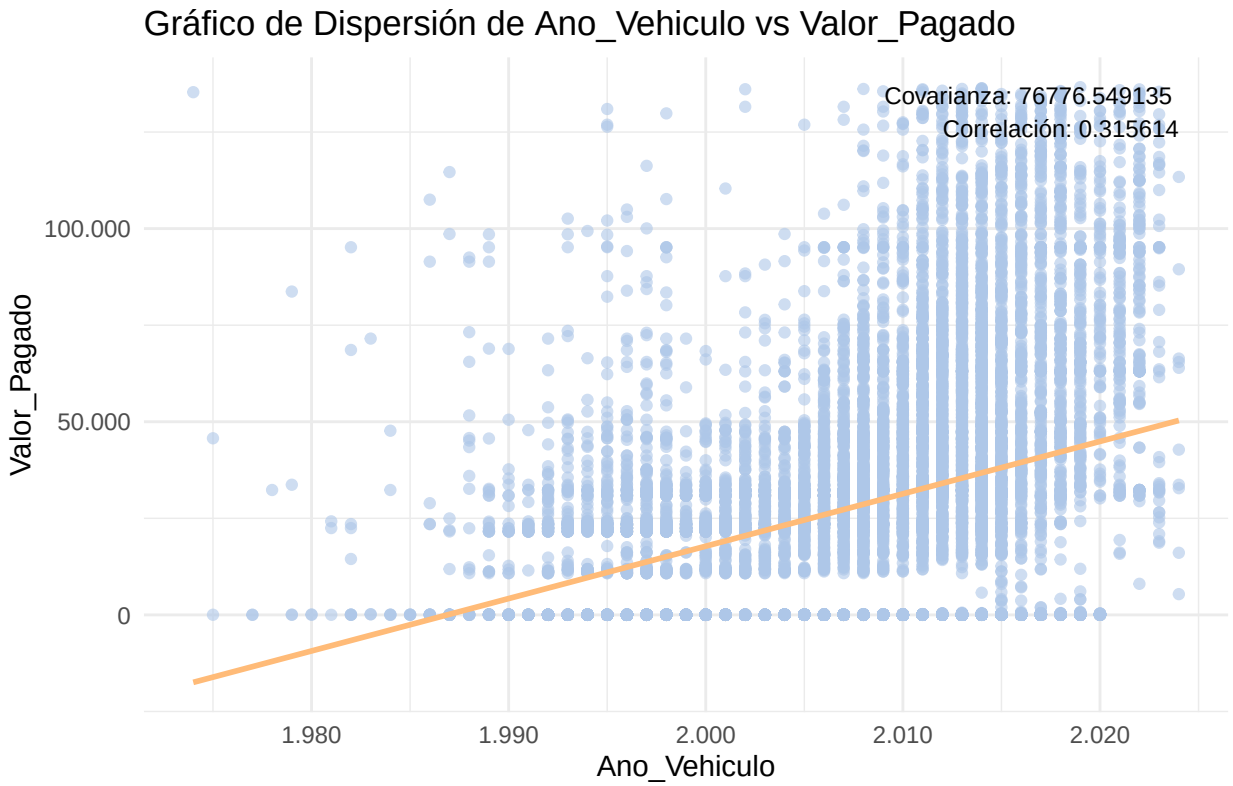
```
grafico(
  data = carga_dataset,
  var  = var_x,
  tipo_grafico = var_tipo_grafico
)
```



Este boxplot con puntos muestra la distribución de Valor_Pagado .

Relacion entre Gastos e Ingresos

```
var_x <- "Ano_Vehiculo"
var_y <- "Valor_Pagado"

grafico_dispersion(
  data = permiso_muestra,
  var_x = var_x,
  var_y = var_y
)
```

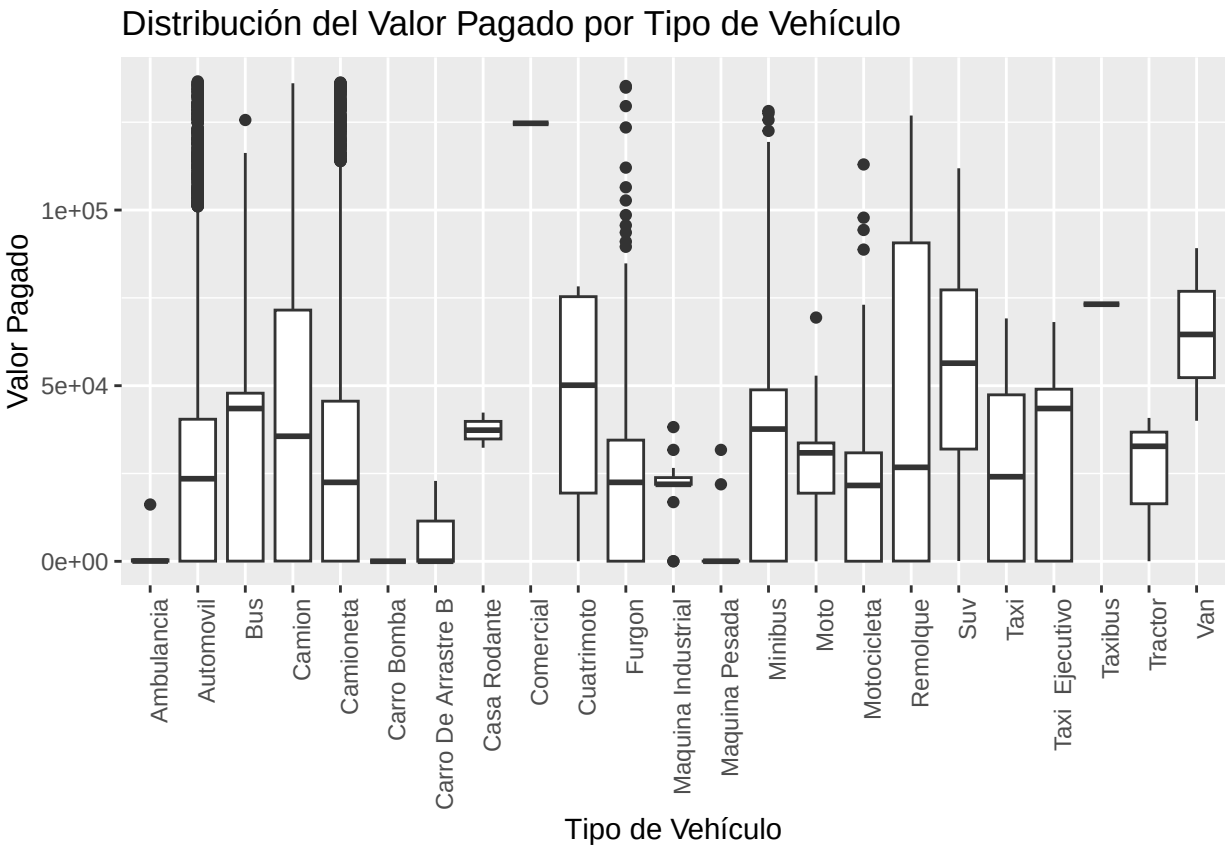


Este gráfico de dispersión muestra la relación entre Ano_Vehiculo y Valor_Pagado
La relaciòn es: Correlación moderada con relación positiva (directa)

```
# Medidas de tendencia central y dispersión
tendencia_central <- permiso_muestra %>%
  summarise(
    media_valor_pagado = mean(`Valor_Pagado`, na.rm = TRUE),
    mediana_valor_pagado = median(`Valor_Pagado`, na.rm = TRUE),
    desviacion_estandar_valor_pagado = sd(`Valor_Pagado`, na.rm = TRUE),
    varianza_valor_pagado = var(`Valor_Pagado`, na.rm = TRUE)
  )

print(tendencia_central)
```

```
##   media_valor_pagado mediana_valor_pagado desviacion_estandar_valor_pagado
## 1           29241.82                23510                          32329.62
##   varianza_valor_pagado
## 1            1045204079
```

```
# Boxplot de valor pagado por tipo de vehículo
ggplot(permiso_muestra, aes(x = `Tipo_Vehiculo`, y = `Valor_Pagado`)) +
  geom_boxplot() +
  labs(title = "Distribución del Valor Pagado por Tipo de Vehículo", x = "Tipo de Vehículo", y = "Valor
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

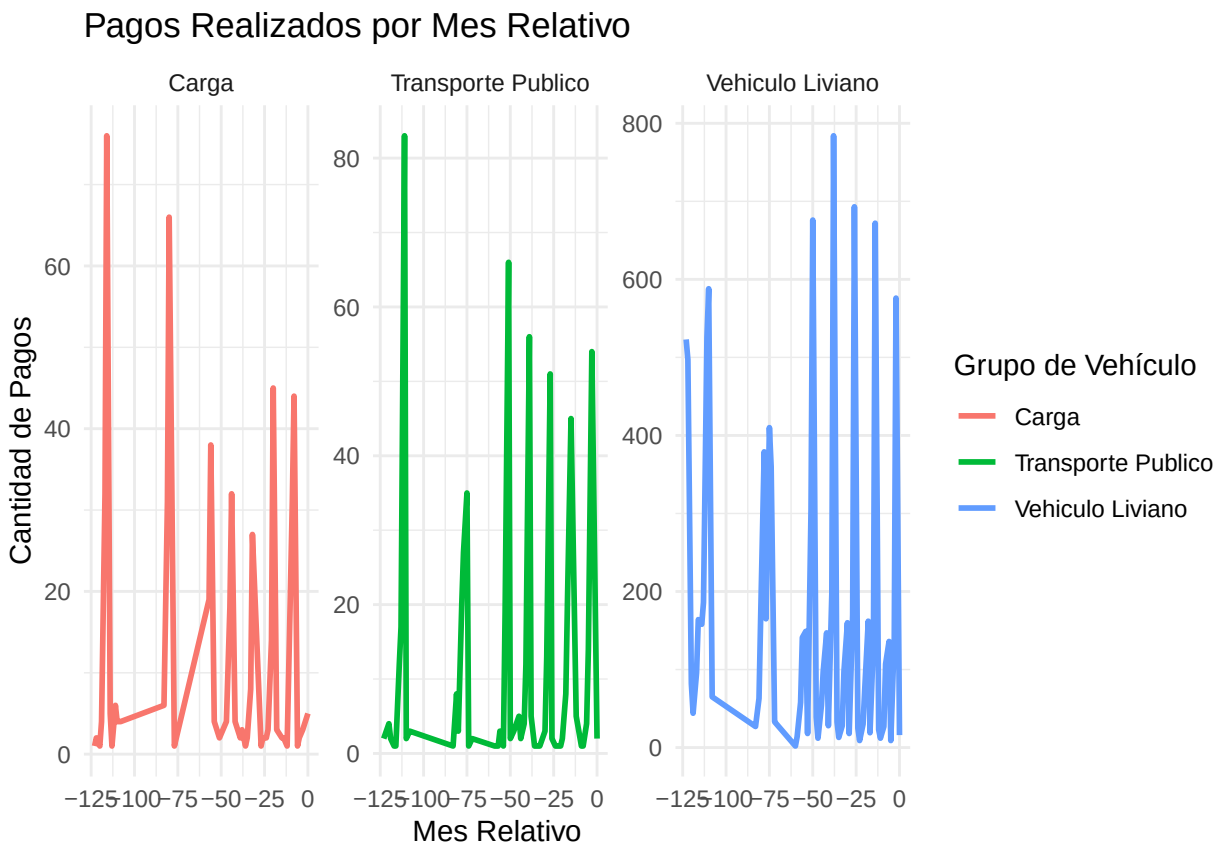


Distribucion de Pagos en Grupos de Vehiculos por Meses Relativos

```r
# Filtrar y transformar los datos

# Calcular el mes relativo
permiso_muestra <- permiso_muestra %>%
  arrange(Fecha_Pago) %>%
  group_by(Grupo_Vehiculo) %>%
  mutate(
    Fecha_Relativa = as.yearmon(Fecha_Pago),
    Mes_Relativo = as.integer((min(Fecha_Relativa) - Fecha_Relativa) * 12)
  ) %>%
  ungroup()

# Resumir los datos por mes relativo y grupo de vehículo
pagos_por_mes_relativo <- permiso_muestra %>%
  group_by(Grupo_Vehiculo, Mes_Relativo) %>%
  summarise(Cantidad_Pagos = n(), .groups = 'drop')

# Crear el gráfico
ggplot(pagos_por_mes_relativo, aes(x = Mes_Relativo, y = Cantidad_Pagos, color = Grupo_Vehiculo)) +
  geom_line(size = 1) +
  facet_wrap(~ Grupo_Vehiculo, scales = "free_y") +
  labs(title = "Pagos Realizados por Mes Relativo",
       x = "Mes Relativo",
       y = "Cantidad de Pagos",
```

```
      color = "Grupo de Vehículo") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Valores Nulos y reemplazar por la Mediana

En caso que se encuentren valores, se reemplazan por el valor de la mediana de cada variable.

```
#Imputar valores nulos por la mediana
permiso_muestra <- permiso_muestra %>%
  mutate(across(everything(), reemplazar_por_mediana))
```

Descripcion de Variables normalizadas

```
# Configurar el entorno de summarytools para HTML
st_options(style = "rmarkdown", plain.ascii = FALSE)

# Obtener un resumen de los datos utilizando summarytools y renderizar en HTML
dfSummary(permiso_muestra) %>%
  print(method = 'render')
```

4. Modelado de Datos

Aplicación de técnicas de modelado de datos y algoritmos de aprendizaje automático para estimar la cantidad de permisos que se pagarán en los próximos 6 meses por grupo y tipo de vehículo.

Técnicas y Algoritmos

1. Selección del Modelo

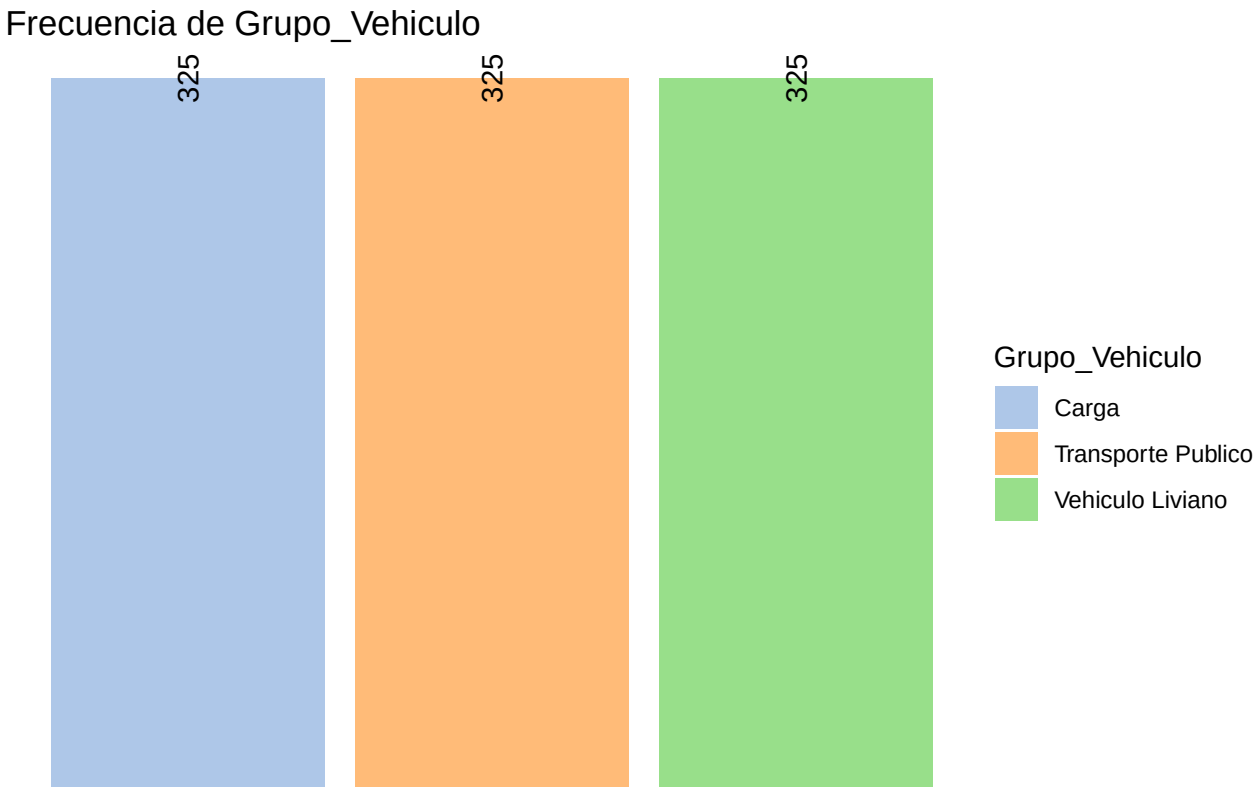
Para este análisis, seleccionaremos tres modelos:

1. Regresión Lineal Múltiple: Este modelo es fácil de interpretar y puede proporcionar una línea base para la comparación con otros modelos más complejos.
2. Árboles de Decisión: Los árboles de decisión son útiles para capturar interacciones no lineales entre las variables y proporcionan interpretaciones claras de las decisiones del modelo.
3. XGBoost: Un algoritmo de boosting que combina múltiples árboles de decisión para mejorar la precisión y el rendimiento del modelo.

2. Entrenamiento y Evaluación

Seleccionar los datos

```
# Configurar semilla para reproducibilidad
set.seed(123)

# Filtrar los datos hasta el año 2018
filtered_permiso_muestra <- permiso_muestra %>%
  filter(Ano_Pago <= 2018)

# Definir el tamaño de la muestra por cada categoría de Grupo_Vehiculo
tamano_muestra <- min(filtered_permiso_muestra %>% count(Grupo_Vehiculo) %>% pull(n))

# Muestreo estratificado para asegurar la misma cantidad de datos por categoría
filtered_permiso_muestra <- filtered_permiso_muestra %>%
  group_by(Grupo_Vehiculo) %>%
  sample_n(tamano_muestra) %>%
  ungroup()


var_x <- "Grupo_Vehiculo"
var_tipo_grafico <- "barra"

grafico(
  data = filtered_permiso_muestra,
  var  = var_x,
  tipo_grafico = var_tipo_grafico
)
```



Este gráfico de barras muestra la frecuencia de Grupo_Vehiculo .

Seleccionar Datos

```
# Calcular el mes relativo
filtered_permiso_muestra <- filtered_permiso_muestra %>%
  arrange(Fecha_Pago) %>%
  group_by(Grupo_Vehiculo) %>%
  mutate(
    Fecha_Relativa = as.yearmon(Fecha_Pago),
    Mes_Relativo = as.integer((min(Fecha_Relativa) - Fecha_Relativa) * 12)
  ) %>%
  ungroup()

# Crear la variable de cantidad de permisos por grupo de vehículo
cantidad_permisos <- filtered_permiso_muestra %>%
  group_by(Grupo_Vehiculo, Mes_Relativo) %>%
  summarise(Cantidad_Permisos = n(), .groups = 'drop')

# Preparar datos con todas las variables
datos_completos <- filtered_permiso_muestra %>%
  left_join(cantidad_permisos, by = c("Grupo_Vehiculo", "Mes_Relativo"))

# Convertir variables categóricas a factores
datos_completos <- datos_completos %>%
  mutate_if(is.character, as.factor)

# Dividir los datos en conjuntos de entrenamiento (80%) y prueba (20%)
```

```r
set.seed(1234)
trainIndex <- createDataPartition(datos_completos$Cantidad_Permisos, p = .8,
                                  list = FALSE,
                                  times = 1)
permisoTrain <- datos_completos[trainIndex,]
permisoTest  <- datos_completos[-trainIndex,]

# Asegurar que los niveles de las variables categóricas sean consistentes entre entrenamiento y prueba
permisoTest <- permisoTest %>%
  mutate(across(where(is.factor), ~ factor(.x, levels = levels(permisoTrain[[cur_column()]]))))

# Convertir variables categóricas a indicadores binarios (dummies)
permisoTrain_matrix <- model.matrix(~ . - 1, data = permisoTrain %>% select(-Cantidad_Permisos))
permisoTest_matrix <- model.matrix(~ . - 1, data = permisoTest %>% select(-Cantidad_Permisos))

# Crear DMatrix para XGBoost
dtrain_cantidad <- xgb.DMatrix(data = permisoTrain_matrix, label = permisoTrain$Cantidad_Permisos)
dtest_cantidad <- xgb.DMatrix(data = permisoTest_matrix, label = permisoTest$Cantidad_Permisos)

# Definir parámetros para el modelo XGBoost
params <- list(booster = "gbtree", objective = "reg:squarederror", eta = 0.3, max_depth = 6)

# Entrenar el modelo XGBoost
modelo_xgb_cantidad <- xgb.train(params, dtrain_cantidad, nrounds = 100)

# Evaluar el modelo
pred_xgb_cantidad <- predict(modelo_xgb_cantidad, dtest_cantidad)
rmse_xgb_cantidad <- RMSE(pred_xgb_cantidad, permisoTest$Cantidad_Permisos)
rsq_xgb_cantidad <- R2(pred_xgb_cantidad, permisoTest$Cantidad_Permisos)

# Mostrar resultados del modelo
resultados_cantidad <- tibble(
  Modelo = "XGBoost",
  RMSE = rmse_xgb_cantidad,
  R2 = rsq_xgb_cantidad
)

print(resultados_cantidad)
```

```
## # A tibble: 1 x 3
##   Modelo   RMSE    R2
##   <chr>   <dbl> <dbl>
## 1 XGBoost  3.90 0.952
```

5. Interpretación de Resultados

Discusión de los Resultados del Análisis:

- Interpretación de los coeficientes del modelo de regresión.
- Identificación de las variables más influyentes.
- Recomendaciones basadas en los hallazgos del análisis.