

Análisis Exploratorio de la Demanda de Viajes SUBE - Año 2024/2025

Una mirada a los patrones de movilidad en
el transporte público argentino

Materia: Captación de Datos
Especialización en Ciencia de Datos
Universidad Nacional de La Matanza

GRUPO 04

Fica Millán, Yesica – DNI 27.624.956

Petraroia, Franco – DNI 27.161.862

Miranda Charca, Florencia – DNI 41.398.768

De Los Rios, Raúl – DNI 37.741.686



CONTEXTO Y PROBLEMÁTICA

- **El transporte público es esencial para la movilidad diaria en Argentina.**
- **Comprender su demanda según el tiempo y el tipo de transporte es clave para planificar mejor el sistema.**
- **En 2025 comienzan a incorporarse nuevos medios de pago, lo que puede afectar los registros con tarjeta SUBE.**

HIPÓTESIS

- Los días hábiles concentran la mayor cantidad de viajes.
- El colectivo es el medio más utilizado.
- En 2025, podría haber una caída en los viajes registrados con SUBE por el uso de medios de pago alternativos.

OBJETIVOS

- Analizar cómo fue la demanda de transporte público en 2024.
- Comparar enero a mayo de 2024 vs 2025 para detectar cambios por la incorporación de nuevos medios de pago.

Fuente: Datos abiertos de la Secretaría de Transporte / Dataset SUBE - Cantidad de transacciones (usos) por fecha

Diccionario de Datos

Columna	Tipo	Descripción
dia_transporte	Fecha (date)	Día de transporte informado (formato ISO-8601)
nombre_empresa	Texto (string)	Nombre de la empresa de transporte
linea	Texto (string)	Descripción de la línea
amba	Texto (string)	Indica si es AMBA (SI/NO)
tipo_transporte	Texto (string)	Colectivo, tren, subte, lanchas
jurisdiccion	Texto (string)	Tipo de jurisdicción de la línea (NACIONAL, PROVINCIAL, MUNICIPAL); vacío para subte
provincia	Texto (string)	Nombre de la provincia; si jurisdicción nacional figura 'JN'; para subte vacío
municipio	Texto (string)	Nombre del municipio; para jurisdicción nacional y provincial figura 'SD' o 'SN' respectivamente.
cantidad	Integer	Cantidad de transacciones de uso
dato_preliminar	Texto (string)	Indica si el dato es preliminar (SI/NO)
dia_semana	Texto (string)	Nombre del día de la semana correspondiente a dia_transporte (LUNES, MARTES, etc.)
tipo_dia	Texto (string)	Clasificación del día: HÁBIL, FERIADO, FIN_DE_SEMANA
motivo_feriado	Texto (string)	Motivo del feriado si corresponde (ej: AÑO NUEVO, NAVIDAD, etc.)
cantidad_log	Float	Logaritmo natural de cantidad para análisis estadístico
MES	Entero (int)	Número del mes correspondiente a la fecha en DIA_TRANSPORTE (1 a 12)
MES_ANO	Texto (string)	Representación del mes y año en formato MM-AAAA o similar
ES_HABIL	Booleano (bool)	Indica si el día es hábil (True) o no (False); útil para análisis binario

EXPLORACIÓN PRELIMINAR

Características generales:

- Ambos datasets presentan las mismas columnas (13 variables).
- Dimensiones:
 - SUBE 2024: 504.676 filas
 - SUBE 2025: 205.633 filas (hasta el 29/05)
- Rango de fechas:
 - 2024: del 01/01 al 31/12
 - 2025: del 01/01 al 29/05

Observaciones exploratorias:

- Estadísticas descriptivas revelan valores mínimos negativos en la columna CANTIDAD.
- La distribución de la variable CANTIDAD presenta asimetría positiva (sesgo hacia la derecha).
- En 2025 se observan inconsistencias gramaticales:
 - JURISDICCIÓN: "C.A.B.A" → "CABA"
 - PROVINCIA: "C.A.B.A" → "CIUDAD AUTÓNOMA DE BUENOS AIRES"

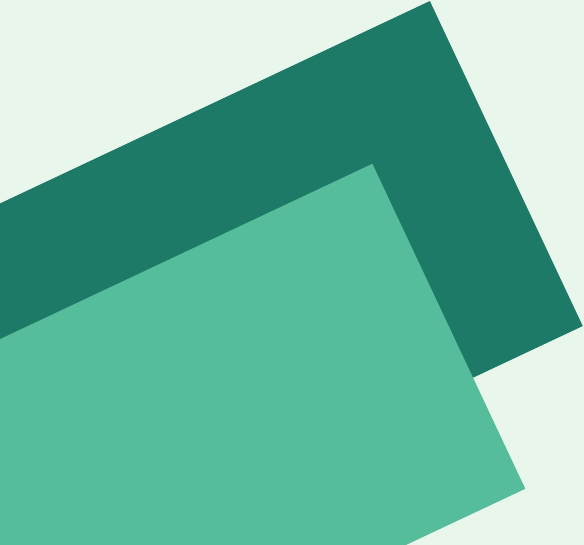
Calidad de los datos:

- Duplicados: 0
- No hay columnas con desviación estándar = 0:
- Valores faltantes en: JURISDICCION, PROVINCIA y MUNICIPIO



ENRIQUECIMIENTO: API Y WEB SCRAPING

Se utilizó una API para identificar feriados nacionales y web scraping para obtener el motivo de cada feriado, enriqueciendo el análisis temporal del dataset.

- **API Nager.Date** para clasificar feriados.
 - **Scraping de La Nación** para agregar motivo de feriado.
- 

API Nager.Date

para clasificar día de la semana y tipo de día

```
api_tipo_dias2024.py X
api_tipo_dias2024.py > ...
1  import requests          # Para realizar peticiones HTTP a una API
2  import pandas as pd      # Para manipulación y análisis de datos en DataFrames
3
4  # 1) Leer el dataset y convierte la columna DIA_TRANSPORTE a formato de fecha
5  df_sube = pd.read_csv("dat-ab-usos-2024.csv", parse_dates=["DIA_TRANSPORTE"])
6
7  # 2) Obtener la lista de feriados en Argentina para 2024 desde la API pública de Nager.Date
8  feriados_2024 = requests.get("https://date.nager.at/api/v3/PublicHolidays/2024/AR").json()
9
10 # Extraer solo las fechas en formato string y guardarlas en un set para evitar duplicados
11 fechas_feriados = set(item["date"] for item in feriados_2024)
12
13 # 3) Función para clasificar cada fecha como FERIADO, FIN_DE_SEMANA o HÁBIL
14 def clasificar_fecha(fecha):
15     fecha_str = fecha.strftime("%Y-%m-%d") # Convertir la fecha a string con formato YYYY-MM-DD
16     if fecha_str in fechas_feriados:
17         return "FERIADO"
18     elif fecha.weekday() >= 5:             # devuelve un número Sábado (5) o domingo (6)
19         return "FIN_DE_SEMANA"
20     else:
21         return "HÁBIL"
22
23 # 4) Diccionario para traducir el nombre del día de la semana al español
24 dias_traduccion = {
25     "Monday": "LUNES", "Tuesday": "MARTES", "Wednesday": "MIÉRCOLES",
26     "Thursday": "JUEVES", "Friday": "VIERNES", "Saturday": "SÁBADO", "Sunday": "DOMINGO"
27 }
28 # Agregar una nueva columna con el nombre del día de la semana en español
29 df_sube["DIA_SEMANA"] = df_sube["DIA_TRANSPORTE"].dt.day_name().map(dias_traduccion)
30
31 # 5) Aplicar la función de clasificación a cada fila del DataFrame
32 df_sube["TIPO_DIA"] = df_sube["DIA_TRANSPORTE"].apply(clasificar_fecha)
33
34 # 6) Guardar el nuevo dataset con las columnas agregadas
35 df_sube.to_csv("df-sube-2024-tipo-dia.csv", index=False, encoding="utf-8-sig")
36 # index=False          → No guarda el índice del DataFrame como columna en el CSV
37 # encoding="utf-8-sig" → Asegura que Excel reconozca correctamente los caracteres especiales
38
39 # Mensaje de confirmación
40 print('✅ Proceso finalizado. Se han agregado las columnas "DIA_SEMANA" y "TIPO_DIA" (desde API) al Dataset.')
41 print('📁 Archivo guardado como: df-sube-2024-tipo-dia.csv')
```


Scraping Diario La Nación

1° Consultar a robots.txt

```
scraping_consulta_robots2024.py
1  import requests
2
3  # Consulta el archivo robots.txt del sitio de Wikipedia
4  print(requests.get("https://www.lanacion.com.ar/robots.txt").text)
5
```

2° Agregar motivo de feriado

```
scraping.feriado2024.py X
scraping.feriado2024.py > ...
1  import requests                # Para realizar peticiones HTTP
2  from bs4 import BeautifulSoup  # Para parsear el contenido HTML
3  import pandas as pd            # Para manipulación de datos con DataFrames
4  from datetime import datetime  # Para manejar fechas
5  import re                      # Para trabajar con expresiones regulares
6
7  # Mapeo de nombres de meses en español a su número correspondiente
8  meses = {
9      "enero": 1,
10     "febrero": 2,
11     "marzo": 3,
12     "abril": 4,
13     "mayo": 5,
14     "junio": 6,
15     "julio": 7,
16     "agosto": 8,
17     "septiembre": 9,
18     "octubre": 10,
19     "noviembre": 11,
20     "diciembre": 12
21 }
22
23 # Diccionario vacío para almacenar los feriados encontrados
24 feriados = {}
25
26 # 1. Descargar y parsear la página de feriados de La Nación para el año 2024
27 url = "https://www.lanacion.com.ar/feriados/2024/"
28 headers = {"User-Agent": "Mozilla/5.0"}
29 resp = requests.get(url, headers=headers, timeout=10)
30 soup = BeautifulSoup(resp.content, "html.parser")
31
32 # 2. Buscar todos los bloques de calendario mensual en la página
33 calendarios = soup.find_all("div", class_="holidays-card-calendar")
34
```

2° Agregar motivo de feriado

raping.feriado2024.py > ...

```
# 2.1 Iterar sobre cada mes encontrado
for calendario in calendarios:
    # Extraer el encabezado del calendario mensual
    encabezado = calendario.find("div", class_="labeled-calendar")
    if not encabezado:
        continue    # Saltar si no se encuentra el encabezado

    # Buscar el nombre del mes dentro del enlace
    link_mes = encabezado.find("a", class_="com-link")
    if not link_mes:
        continue    # Saltar si no se encuentra el enlace

    # Obtener el nombre y número del mes
    nombre_mes = link_mes.text.strip().lower()
    numero_mes = meses.get(nombre_mes)
    if not numero_mes:
        print(f"⚠ Mes no reconocido: {nombre_mes}")
        continue

    # Buscar la lista de feriados del mes
    ul = calendario.find("ul", class_="holidays-list")
    if not ul:
        continue    # Saltar si no se encuentra la lista

    # Iterar sobre cada feriado del mes
    for li in ul.find_all("li"):
        # Buscar el día del feriado y el motivo
        dia_tag = li.find("span", class_=re.compile(r"--"))    # Tag con
        motivo_tag = li.find("h4", class_="com-text")    # Tag con
        if not dia_tag or not motivo_tag:
            continue    # Saltar si falta alguno de los datos
```

```
try:
    # Parsear día y motivo, y construir la fecha completa en formato YYYY-MM-DD
    dia = int(dia_tag.text.strip())
    motivo = motivo_tag.text.strip().upper()
    fecha = datetime(2024, numero_mes, dia).strftime("%Y-%m-%d")

    # Guardar la fecha y el motivo en el diccionario
    feriados[fecha] = motivo
except Exception as e:
    # Mostrar error si ocurre al intentar procesar una entrada
    print(f"❌ Error procesando {nombre_mes} {li}: {e}")
```

```
# 3. Mostrar los feriados encontrados en consola
print("Feriados encontrados:")
for fecha, motivo in sorted(feriados.items()):
    print(f"{fecha} → {motivo}")
```

```
# 4. Leer archivo CSV con datos de SUBE y etiquetar si el día es feriado
df_sube = pd.read_csv("df-sube-2024-tipo-dia.csv", parse_dates=["DIA_TRANSPORTE"])
```

```
# Crear nueva columna con el motivo del feriado (o "NO FERIADO" si no aplica)
df_sube["MOTIVO_FERIADO"] = df_sube["DIA_TRANSPORTE"].dt.strftime("%Y-%m-%d").map(feriados).fillna("NO FERIADO")
```

```
# 5. Guardar el DataFrame resultante en un nuevo archivo CSV
df_sube.to_csv("df-sube-2024.csv", index=False, encoding="utf-8-sig")
```

```
# Mensajes de confirmación final
print("✅ Scraping finalizado. Archivo guardado como df-sube-2024.csv")
print(f"✅ Total de feriados encontrados: {len(feriados)}")
```


Limpieza de datos

Se depuraron los datos para asegurar un análisis confiable, corrigiendo errores y completando información faltante.

```
eda_sube2024.py x
eda_sube2024.py > identificar_outliers_por_grupo
39 # 3 ----- ESTADÍSTICAS DESCRIPTIVAS -----
40 print("-" * 80 + "\nESTADÍSTICAS DESCRIPTIVAS (NUMÉRICAS)\n")
41 print(df.describe(include=[np.number]).round(2)) # filtra que solo se incluyan columnas de tipo numérico y con 2 decimales
42
43 # ♦ Detectar filas con valores negativos en la columna 'CANTIDAD'
44 negativos = df[df["CANTIDAD"] < 0]
45 print("\nCantidad de filas con valores negativos en CANTIDAD:", len(negativos))
46 print(negativos[["DIA_TRANSPORTE", "TIPO_TRANSPORTE", "CANTIDAD"]].head(10))
47 print("\n⚠ Se encontraron 3 valores anómalos en CANTIDAD (valores negativos). Estos se consideran errores o correcciones no documentadas.")
48
49 # ♦ Eliminar filas con valores negativos en 'CANTIDAD' para evitar problemas en el análisis
50 df = df[df["CANTIDAD"] >= 0]
51 print("✅ Filas con valores negativos en 'CANTIDAD' eliminadas correctamente.")
52
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

ESTADÍSTICAS DESCRIPTIVAS (NUMÉRICAS)

	CANTIDAD
count	504676.00
mean	8194.59
std	18748.33
min	-105.00
25%	545.00
50%	2127.00
75%	7617.00
max	516002.00

Cantidad de filas con valores negativos en CANTIDAD: 3

	DIA_TRANSPORTE	TIPO_TRANSPORTE	CANTIDAD
65337	2024-02-20	TREN	-3
84741	2024-03-05	TREN	-105
147242	2024-04-20	TREN	-1

⚠ Se encontraron 3 valores anómalos en CANTIDAD (valores negativos). Estos se consideran errores o correcciones no documentadas.
✅ Filas con valores negativos en 'CANTIDAD' eliminadas correctamente.

Valores faltantes

```
90 # 5 ----- VALORES FALTANTES -----
91 print("-" * 50 + "\nVALORES FALTANTES POR COLUMNA\n" + "-" * 50)
92 print(df.isna().sum()) # Devuelve un df del mismo tamaño con valores booleanos, luego suma los True por columna
93
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

VALORES FALTANTES POR COLUMNA

DIA_TRANSPORTE	0
NOMBRE_EMPRESA	0
LINEA	0
AMBA	0
TIPO_TRANSPORTE	0
JURISDICCION	2522
PROVINCIA	2537
MUNICIPIO	2537
CANTIDAD	0
DATO_PRELIMINAR	0
DIA_SEMANA	0
TIPO_DIA	0
MOTIVO_FERIADO	0
CANTIDAD_LOG	0
dtype:	int64

```
eda_sube2024.py > ...
94 print("\n----- ANÁLISIS DE VALORES NULOS -----")
95
96 # Filas con JURISDICCION nula
97 nulos_jur = df[df["JURISDICCION"].isna()]
98 print("\nTipos de transporte con JURISDICCION nula:")
99 print(nulos_jur["TIPO_TRANSPORTE"].value_counts())
100
101 # Filas con PROVINCIA nula
102 nulos_prov = df[df["PROVINCIA"].isna()]
103 print("\nTipos de transporte con PROVINCIA nula:")
104 print(nulos_prov["TIPO_TRANSPORTE"].value_counts())
105
106 # Filas con MUNICIPIO nulo
107 nulos_mun = df[df["MUNICIPIO"].isna()]
108 print("\nTipos de transporte con MUNICIPIO nulo:")
109 print(nulos_mun["TIPO_TRANSPORTE"].value_counts())
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

----- ANÁLISIS DE VALORES NULOS -----

Tipos de transporte con JURISDICCION nula:

TIPO_TRANSPORTE	count
SUBTE	2522

Name: count, dtype: int64

Tipos de transporte con PROVINCIA nula:

TIPO_TRANSPORTE	count
SUBTE	2522
COLECTIVO	11
TREN	4

Name: count, dtype: int64

Tipos de transporte con MUNICIPIO nulo:

TIPO_TRANSPORTE	count
SUBTE	2522
COLECTIVO	11
TREN	4

Name: count, dtype: int64

Análisis y Corrección

```
eda_sube2024.py X
eda_sube2024.py > ...
111 # Correccion datos nulos
112 es_subte = df["TIPO_TRANSPORTE"] == "SUBTE" # Filtrar filas donde TIPO_TRANSPORTE es SUBTE
113 df.loc[es_subte, "JURISDICCION"] = df.loc[es_subte, "JURISDICCION"].fillna("CABA") # Completar valores nulos
114 df.loc[es_subte, "PROVINCIA"] = df.loc[es_subte, "PROVINCIA"].fillna("CIUDAD AUTÓNOMA DE BUENOS AIRES")
115 df.loc[es_subte, "MUNICIPIO"] = df.loc[es_subte, "MUNICIPIO"].fillna("CABA")
116
117 # Verificación final de nulos
118 print("\n----- Verificación de valores nulos tras corrección en SUBTE -----")
119 print(df[["JURISDICCION", "PROVINCIA", "MUNICIPIO"]].isna().sum())
120
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

----- Verificación de valores nulos tras corrección en SUBTE -----

	count
JURISDICCION	0
PROVINCIA	15
MUNICIPIO	15

dtype: int64

Análisis y Corrección

```
eda_sube2024.py X
eda_sube2024.py > ...

121 # Filas con PROVINCIA nula
122 nulos_prov = df[df["PROVINCIA"].isna()]
123 print("\nValores Nulos por Provincia:")
124 print(nulos_prov[["TIPO_TRANSPORTE", "NOMBRE_EMPRESA", "LINEA", "AMBA"]].to_string(index=False))
125
126 # Filas con MUNICIPIO nulo
127 nulos_mun = df[df["MUNICIPIO"].isna()]
128 print("\nValores Nulos por Municipio:")
129 print(nulos_mun[["TIPO_TRANSPORTE", "NOMBRE_EMPRESA", "LINEA", "AMBA"]].to_string(index=False))
130
131 # COLECTIVO - Empresa 9 de Julio SRL - Línea 500 Santa Fe
132 colectivo_sfe = (df["TIPO_TRANSPORTE"] == "COLECTIVO") & (df["LINEA"] == "LINEA_500I_SFE")
133 df.loc[colectivo_sfe, "PROVINCIA"] = "SANTA FE"
134 df.loc[colectivo_sfe, "MUNICIPIO"] = "SANTA FE"
135
136 # TREN - Tren del Valle
137 tren_valle = (df["TIPO_TRANSPORTE"] == "TREN") & (df["LINEA"] == "FFCC TREN DEL VALLE")
138 df.loc[tren_valle, "PROVINCIA"] = "JN"
139 df.loc[tren_valle, "MUNICIPIO"] = "SD"
140
141 print("\n----- Verificación final tras imputación específica: -----")
142 print(df[["JURISDICCION", "PROVINCIA", "MUNICIPIO"]].isna().sum())
143 print("\n✅ Filas con valores nulos corregidas correctamente.")
```

Valores Nulos por Provincia:

TIPO_TRANSPORTE	NOMBRE_EMPRESA	LINEA	AMBA
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO

Valores Nulos por Municipio:

TIPO_TRANSPORTE	NOMBRE_EMPRESA	LINEA	AMBA
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
COLECTIVO	EMPRESA 9 DE JULIO SRL	LINEA_500I_SFE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO
TREN	SOFSE - TREN DEL VALLE FFCC	TREN DEL VALLE	NO

----- Verificación final tras imputación específica: -----

```
JURISDICCION    0
PROVINCIA       0
MUNICIPIO       0
dtype: int64
```

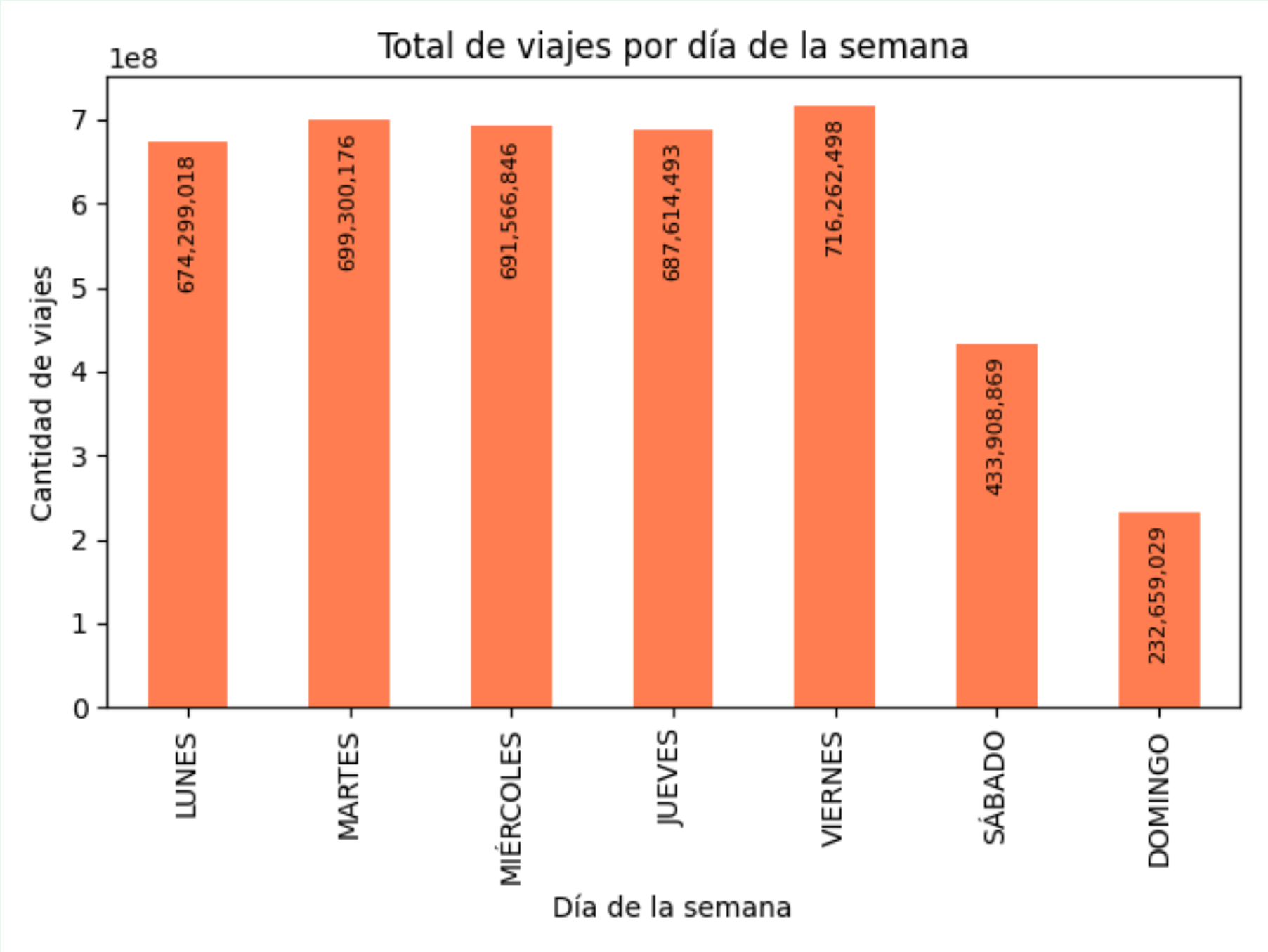
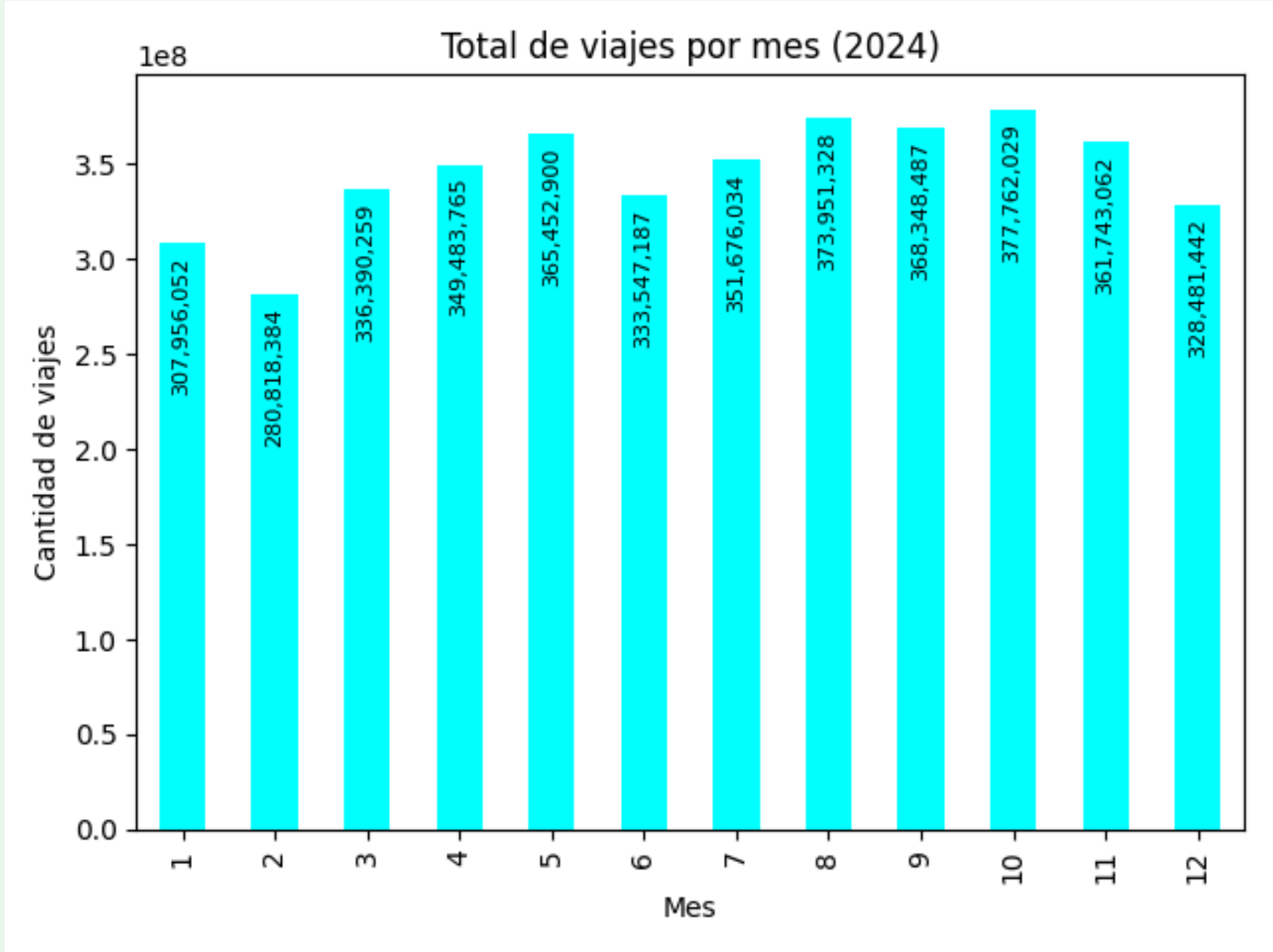
✅ Filas con valores nulos corregidas correctamente.

ANÁLISIS TEMPORAL

Variable CANTIDAD

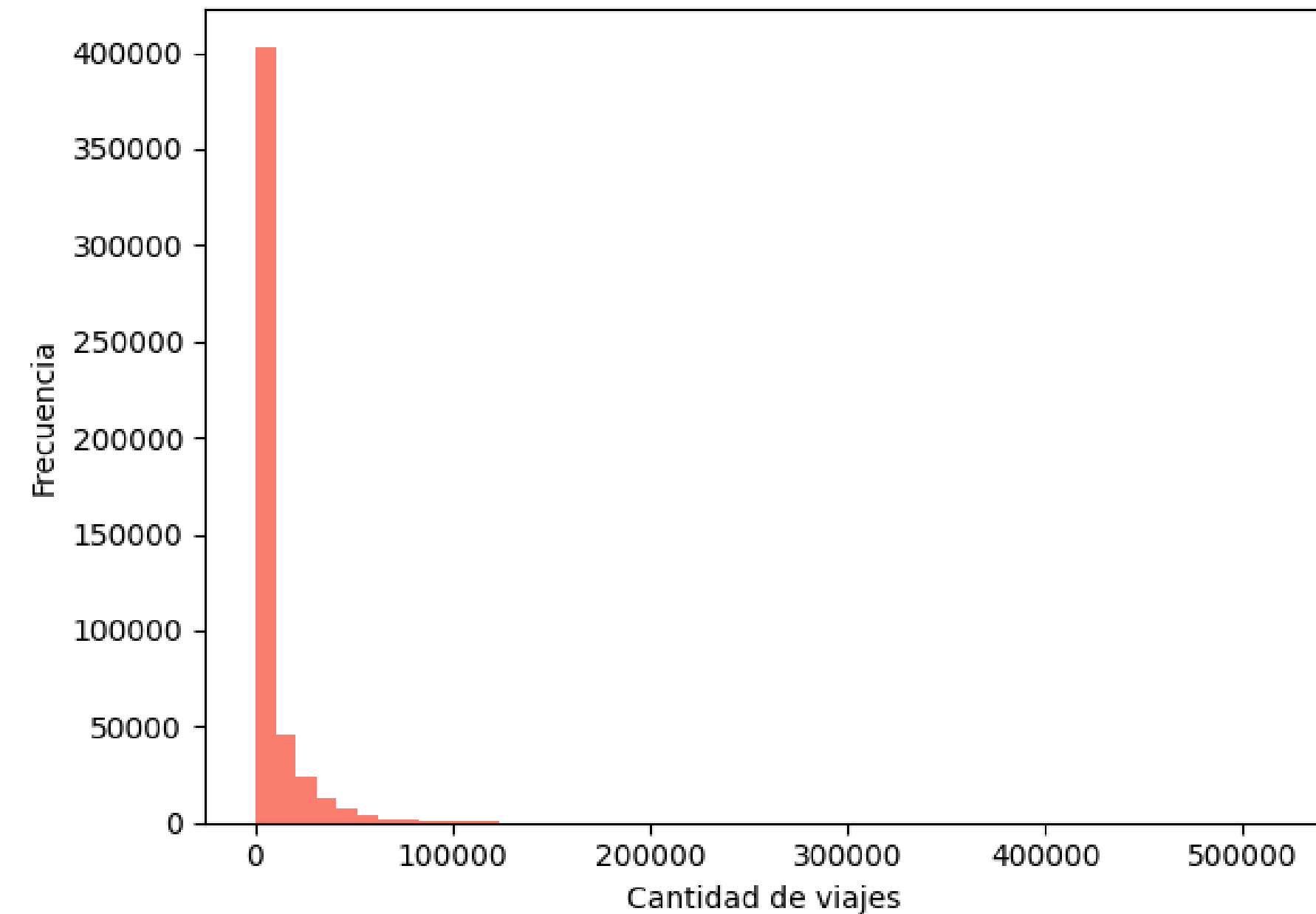
- Qué tan concentrada/distribuida está?
- Tiene sesgos, outliers, estacionalidad?

Series Temporales por Mes y Días de la Semana



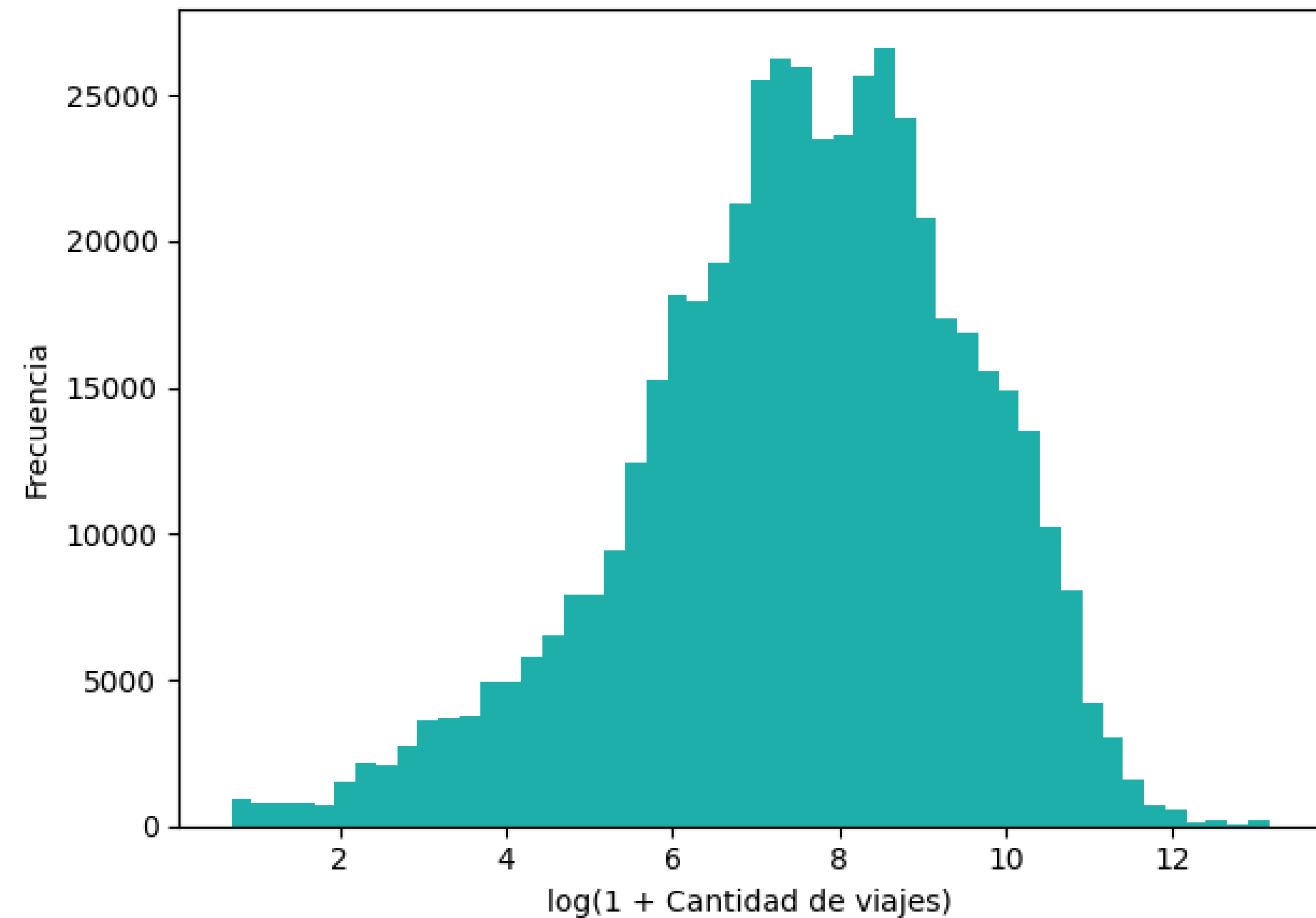
Cantidad de Viajes

Distribución de CANTIDAD de viajes



Distribución sesgada a la derecha

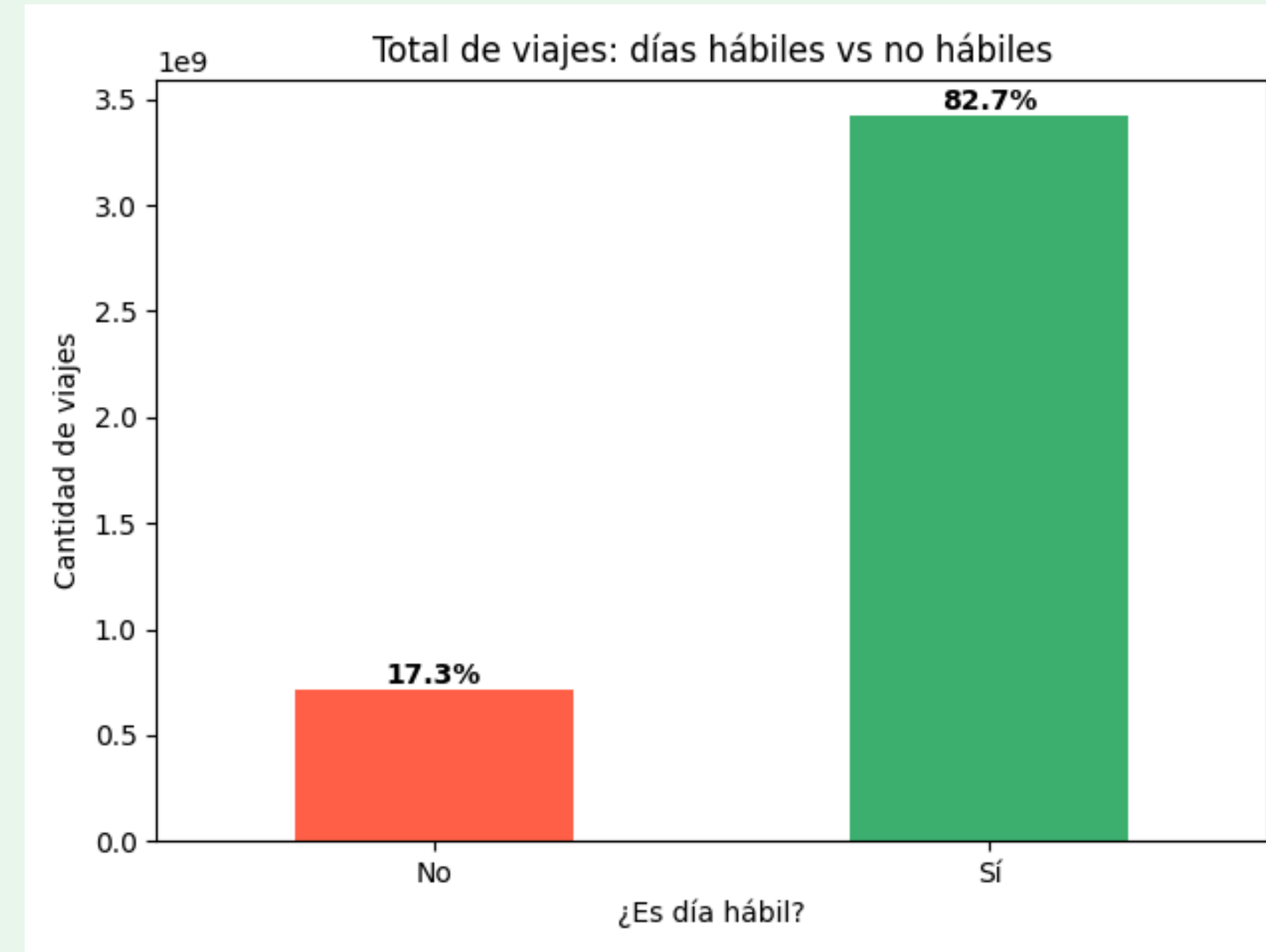
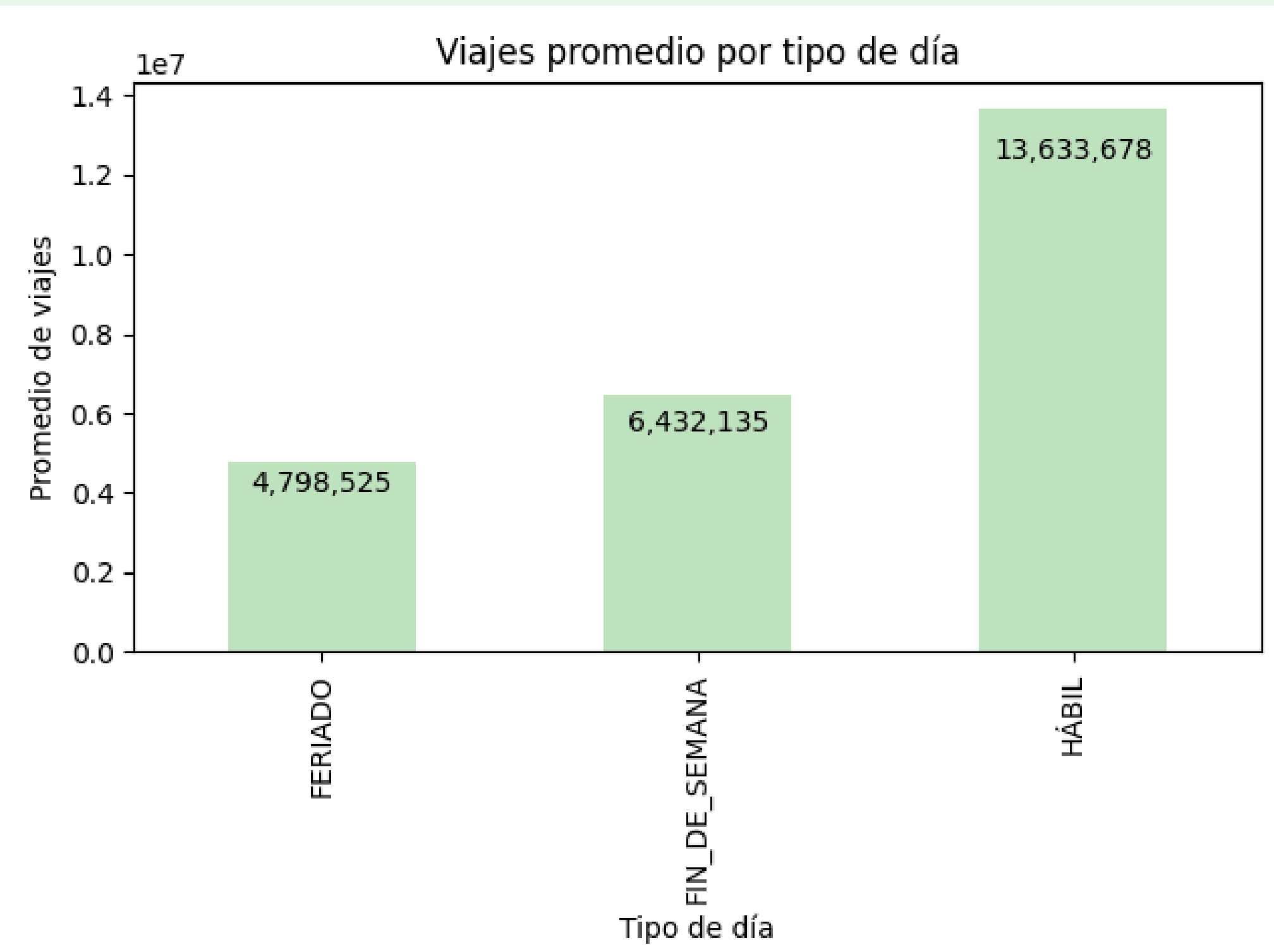
Distribución logarítmica de CANTIDAD

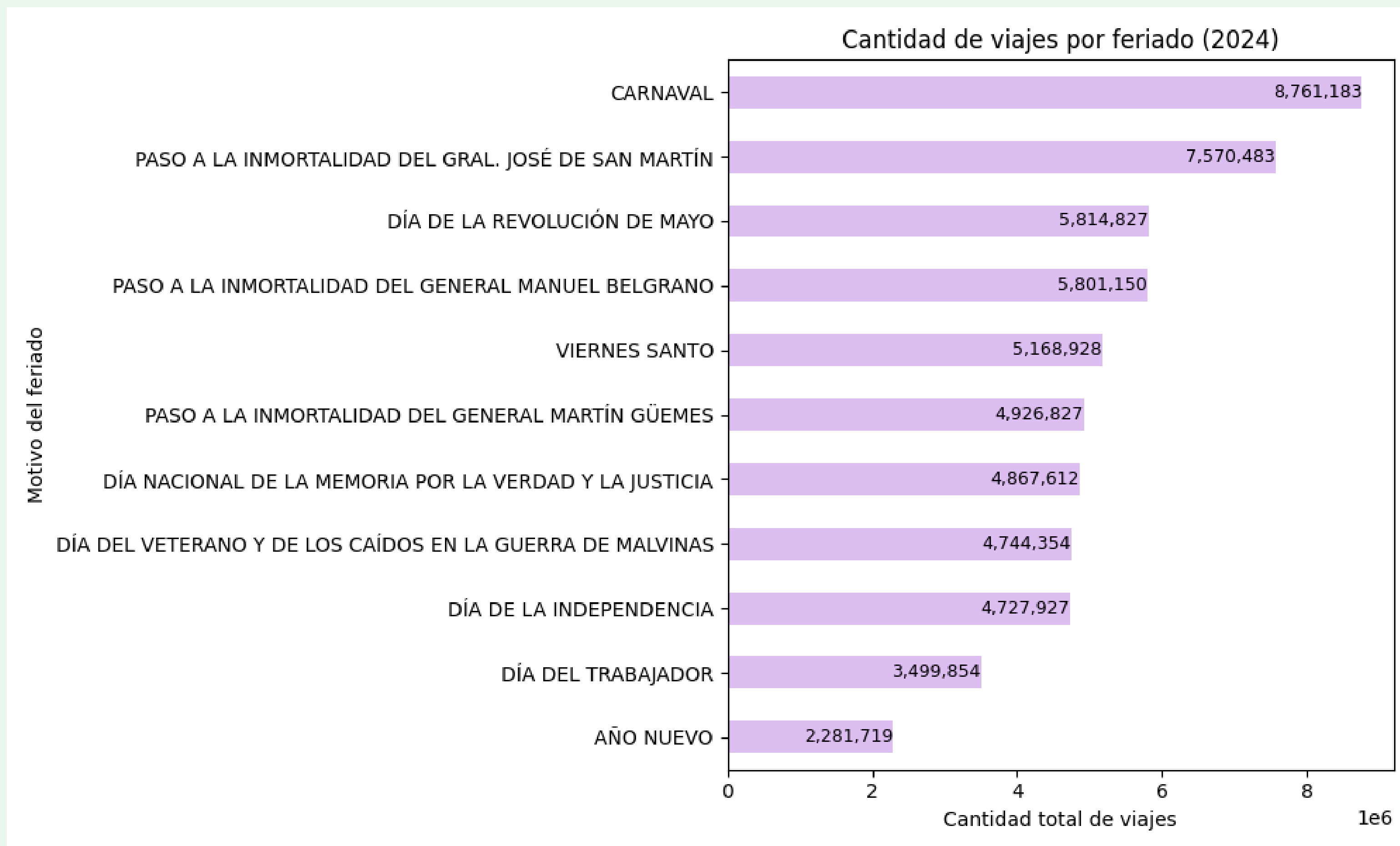


Distribución normalizada

Se observa una alta concentración en valores bajos y pocos valores extremos, lo que motivó aplicar una transformación logarítmica.

Series Temporales por Tipo de Día

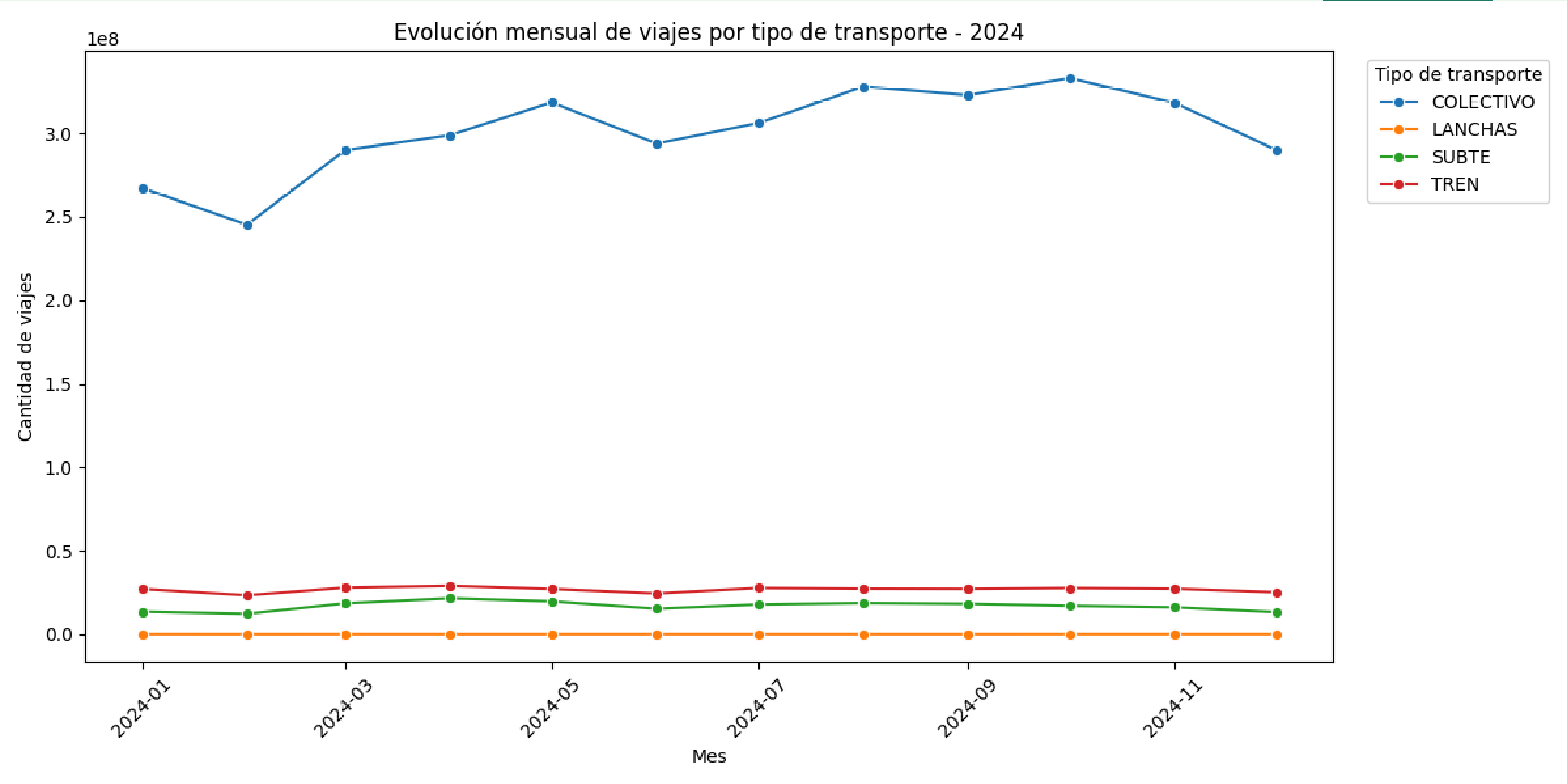




ANÁLISIS POR TIPO DE TRANSPORTE

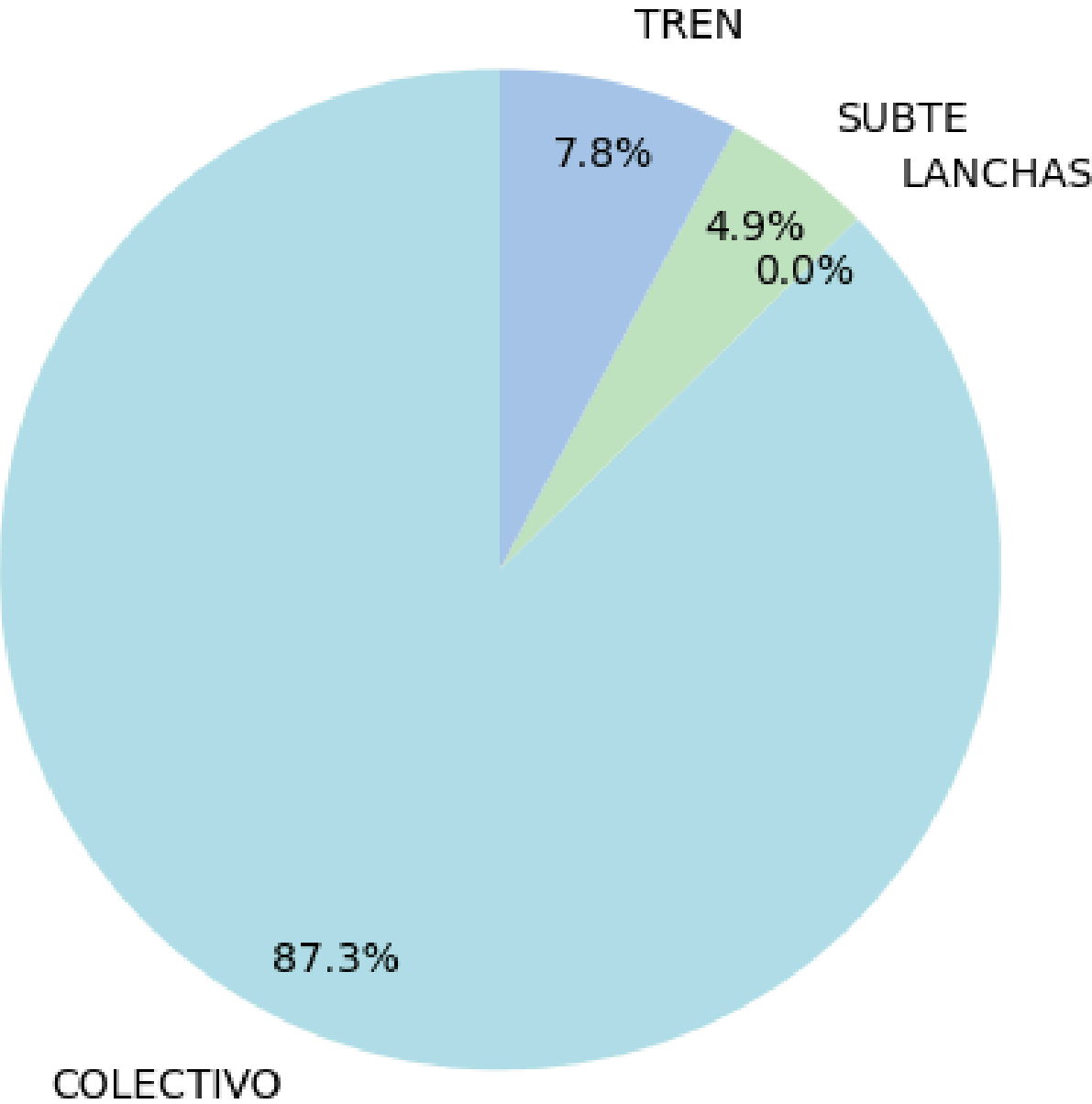
- Cual es el más utilizado?
- Como se distribuyen?
- Cantidad de viajes por tipo de transporte y tipo de día de la semana

Distribución Mensual por Tipo de Transporte

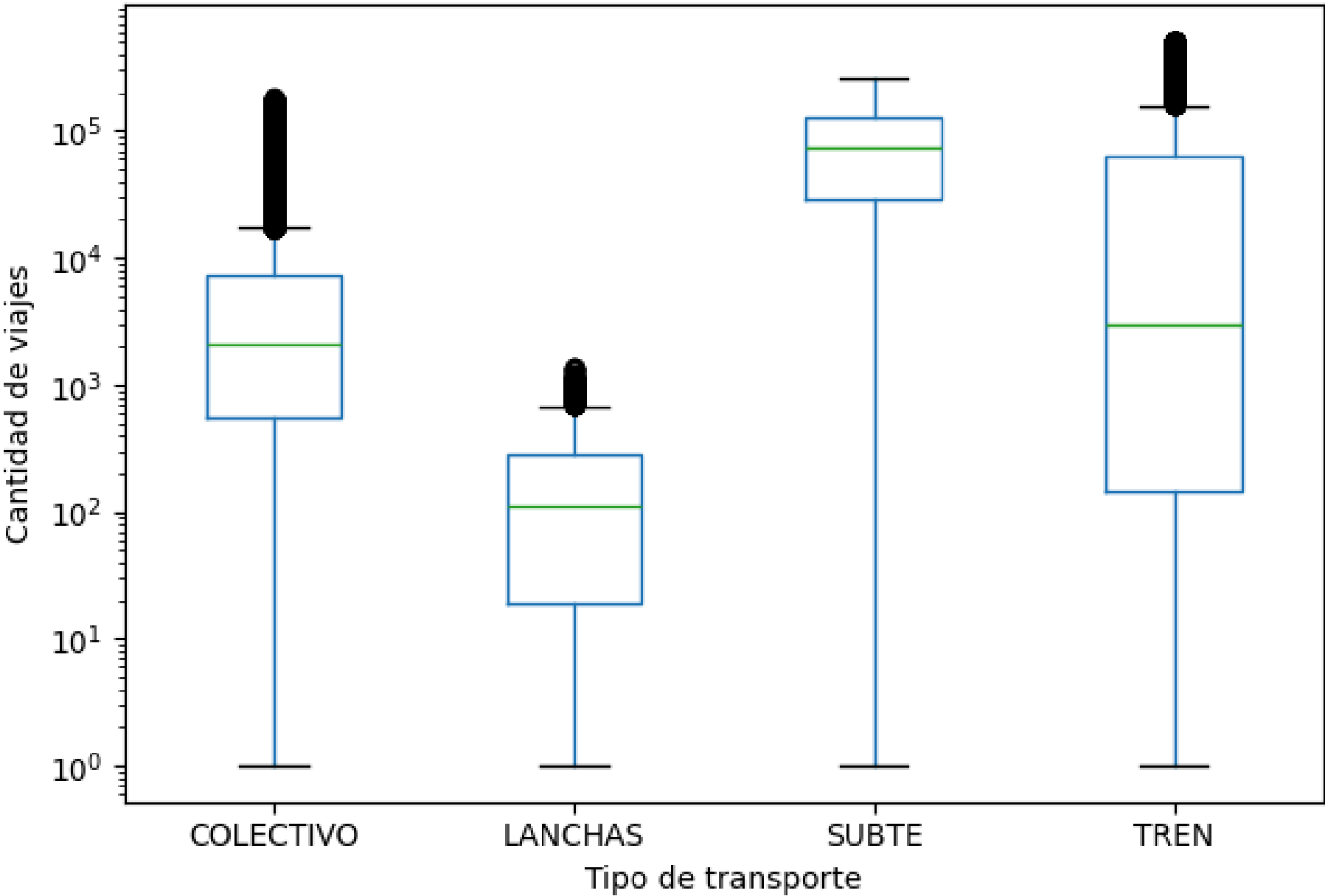


Relación entre variables TIPO_TRANSPORTE vs CANTIDAD

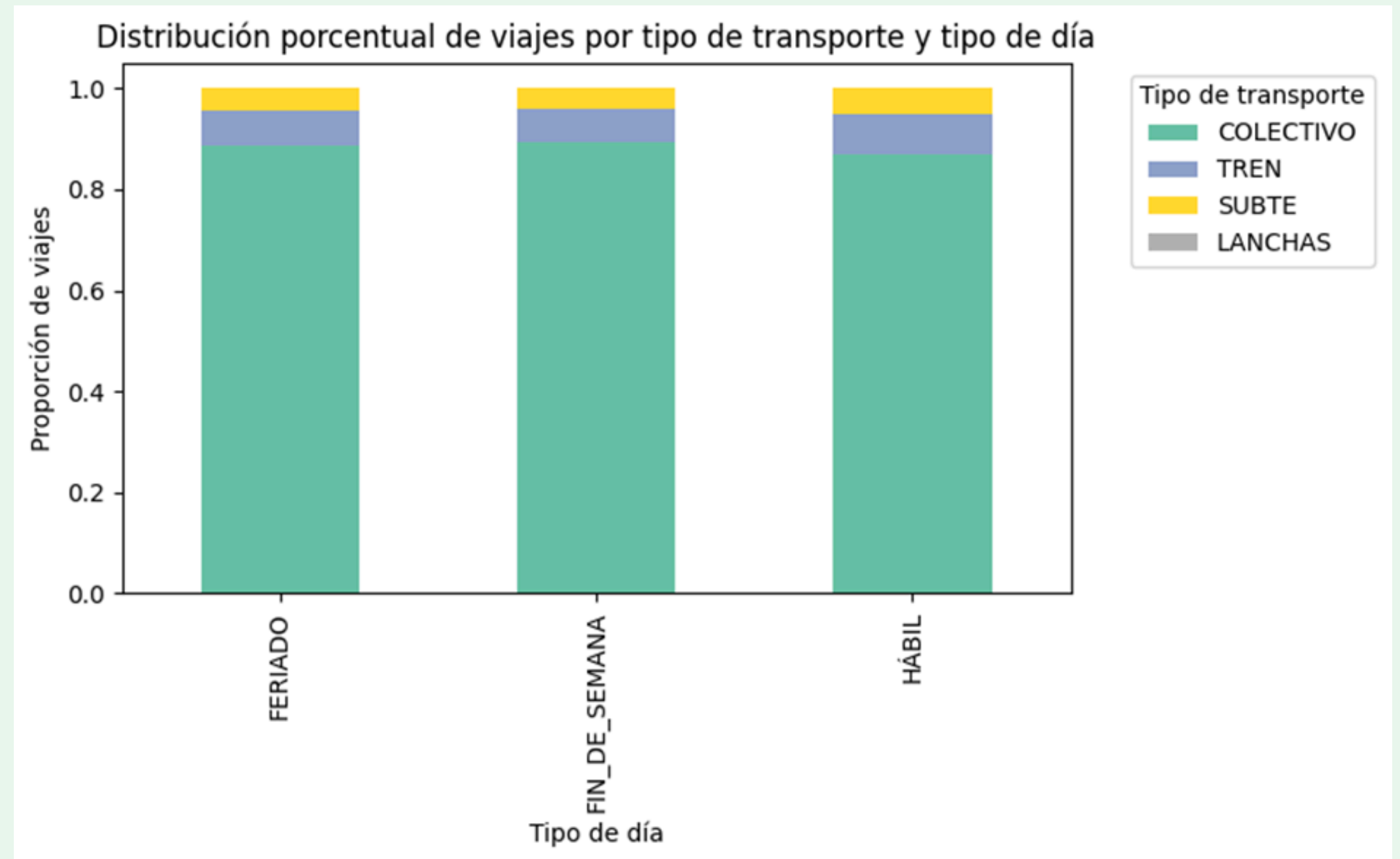
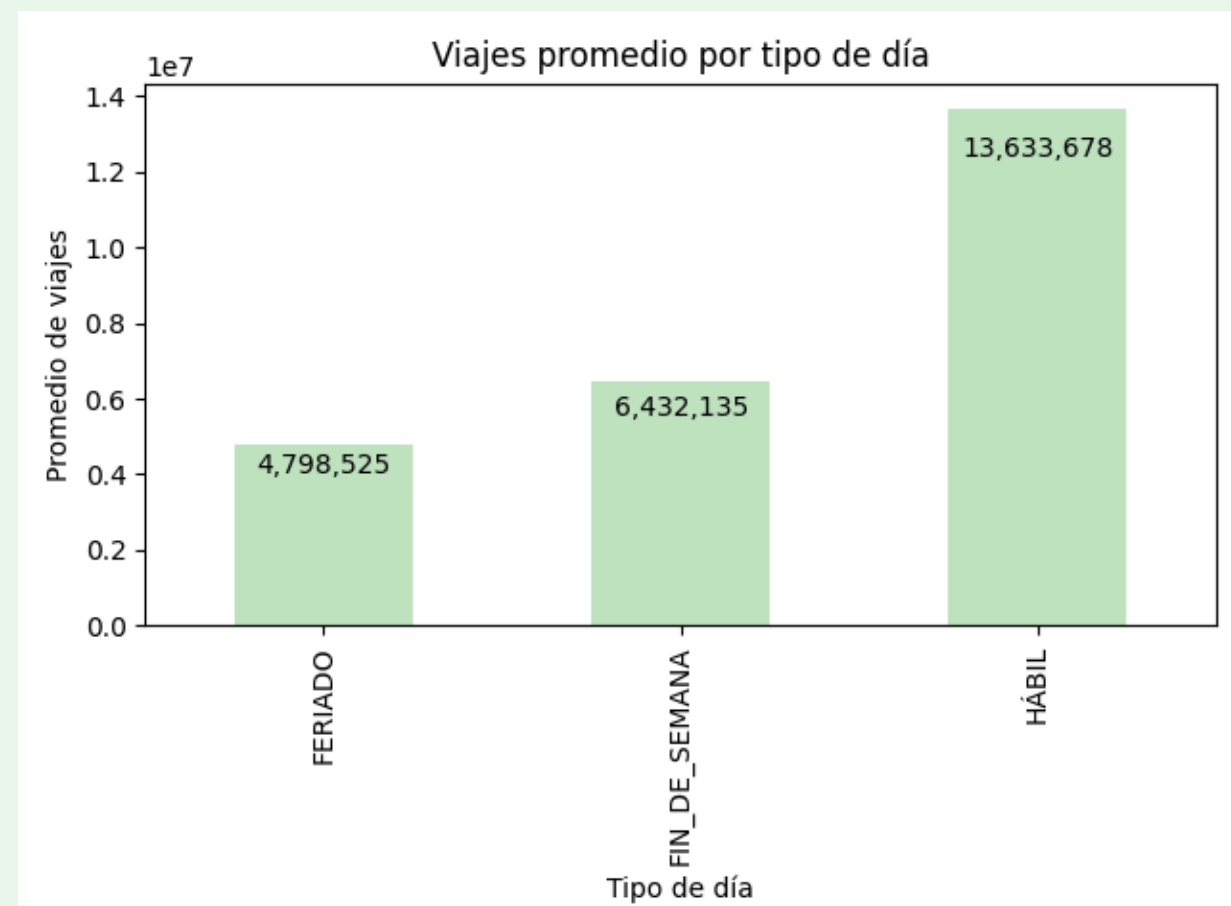
Distribución de viajes por tipo de transporte



Distribución de viajes por tipo de transporte

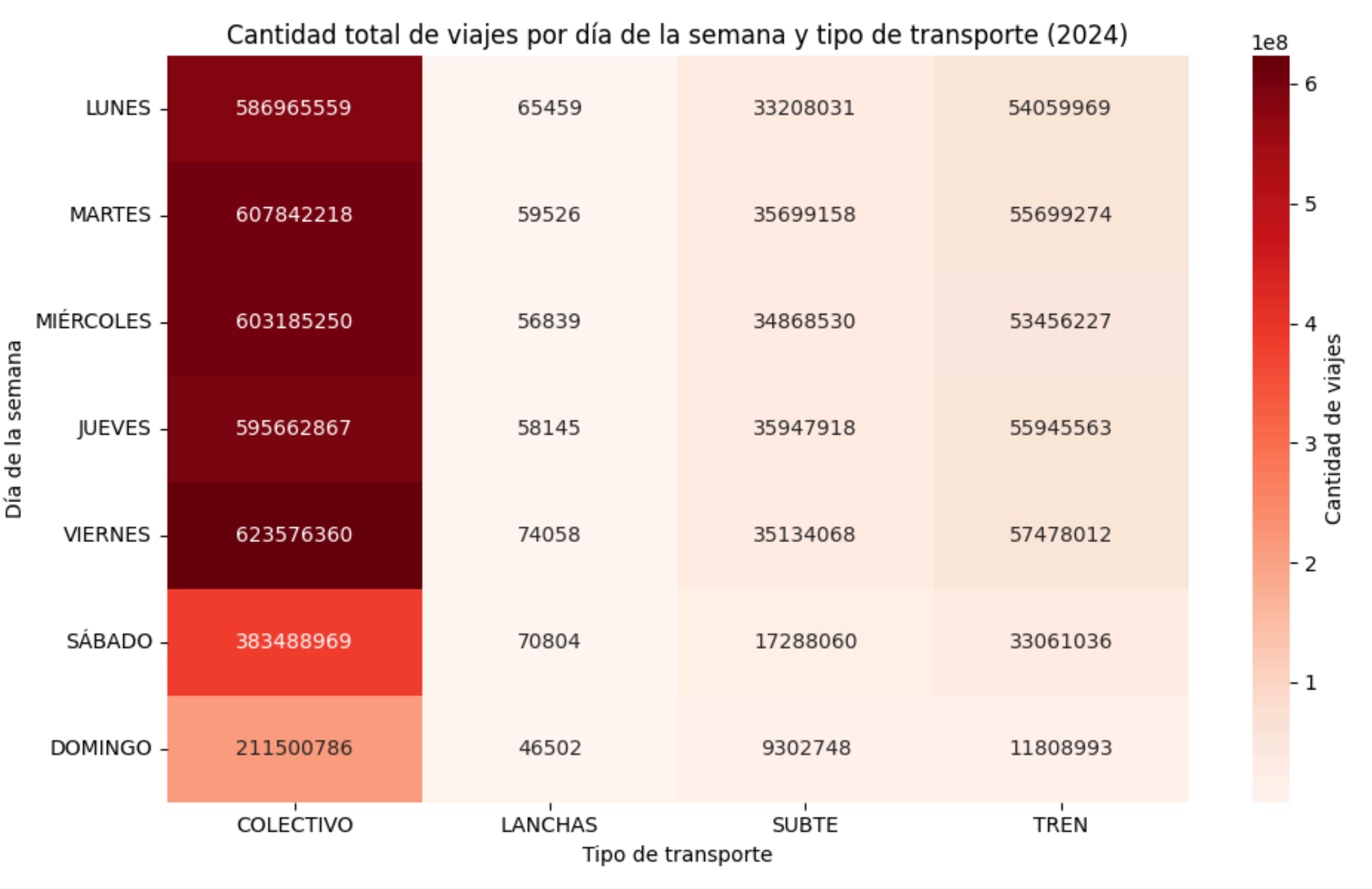


Distribución de la Cantidad de Viajes por Tipo de Transporte



Distribución de la cantidad de viajes por día de semana y tipo de transporte

TENDENCIA EN LA CANTIDAD DE VIAJES REALIZADOS

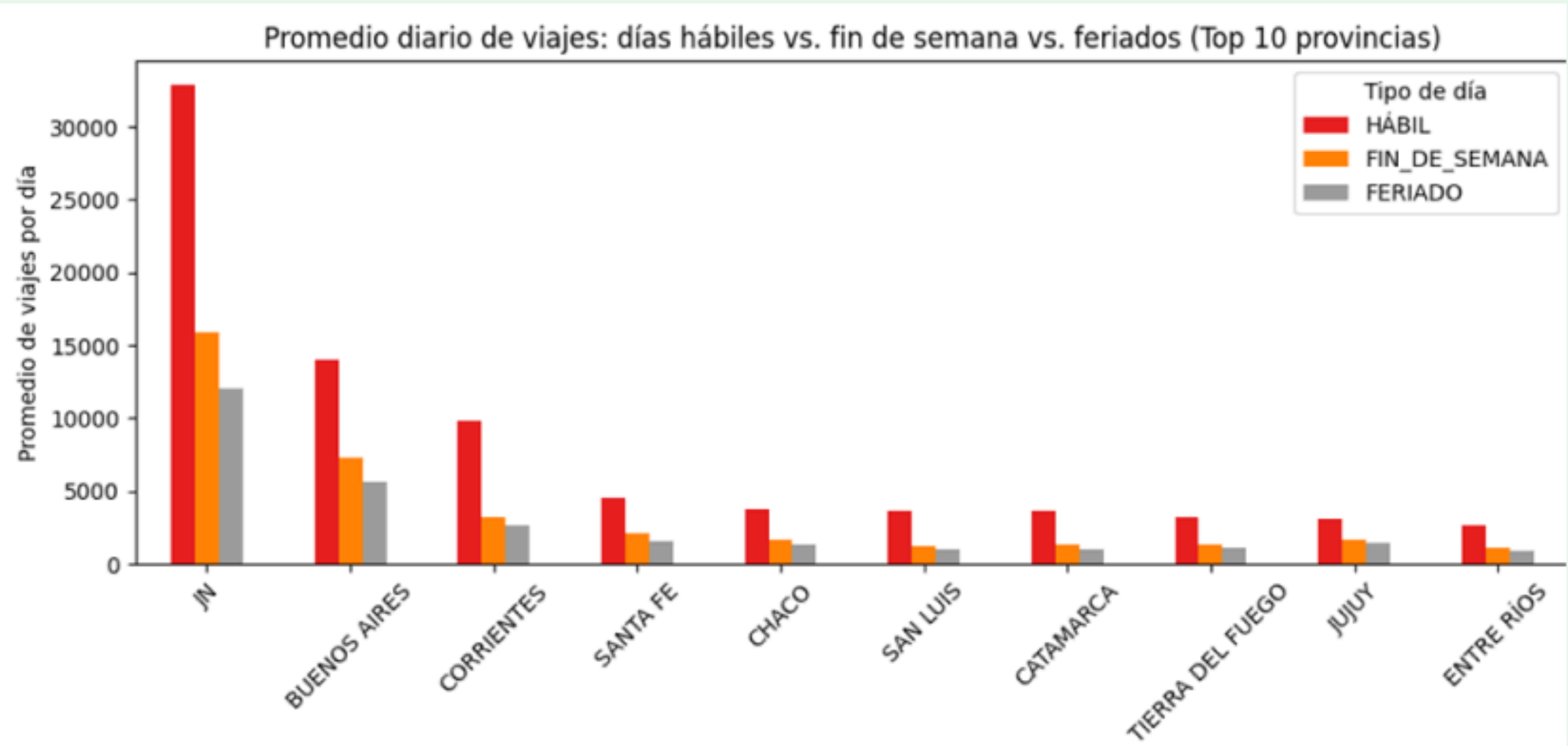


- El viernes concentra el mayor volumen total de viajes: colectivos, trenes y lanchas.
- El subte alcanza su uso máximo los jueves, posiblemente por dinámicas laborales y recreativas en CABA. La modalidad de trabajo flexible podría explicar una menor movilización los viernes, lo que requiere un análisis más profundo.

Distribución de la cantidad de viajes por tipo de día y tipo de transporte a lo largo del año 2024 (Top 10 provincias)

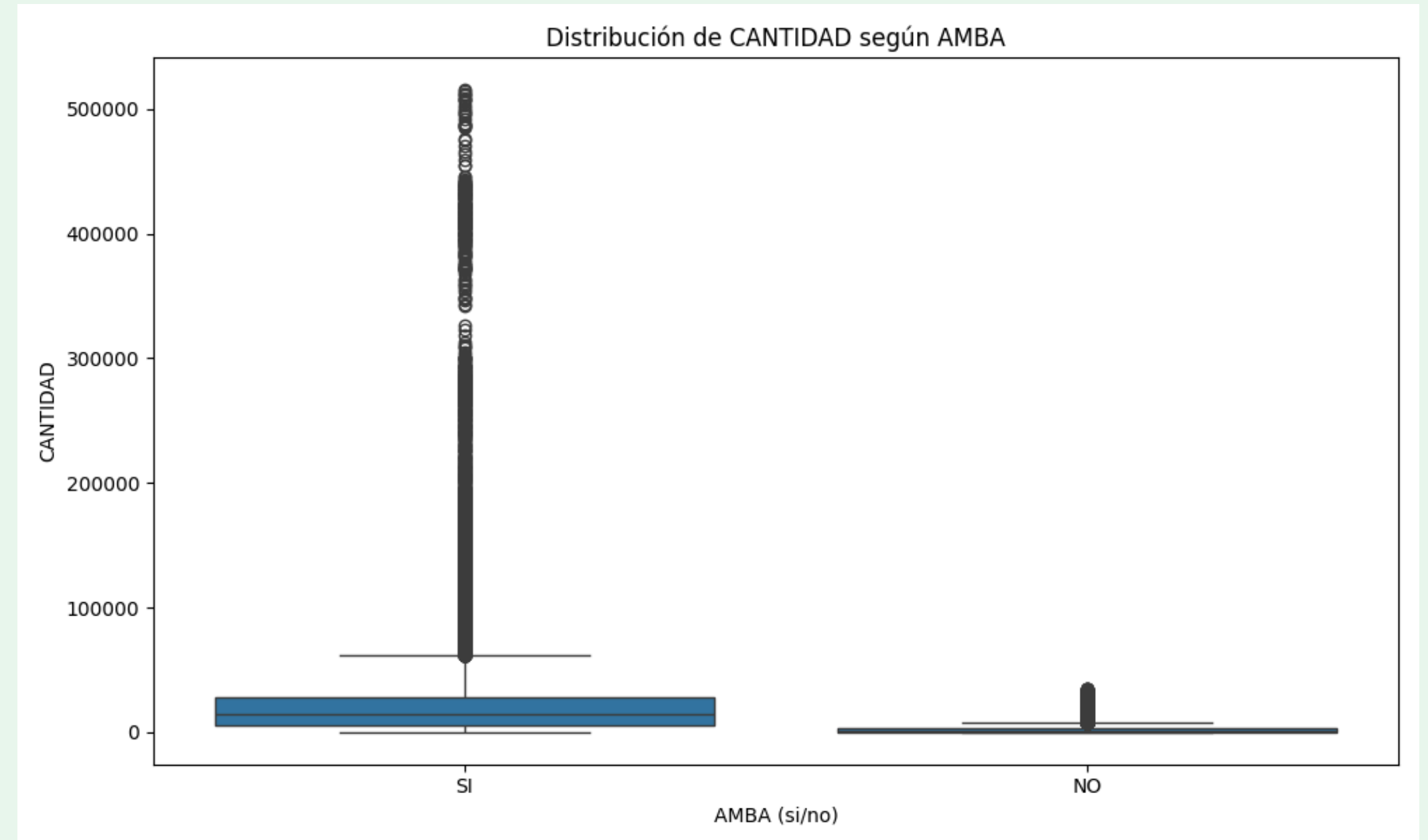
ANÁLISIS DE DATOS

- Los días hábiles concentran la mayor cantidad de viajes diarios, seguidos por los fines de semana y finalmente los feriados.
- La caída en la demanda durante feriados se explica por la suspensión de actividades laborales y escolares, lo que demuestra la estrecha relación entre la movilidad y el ritmo de la vida productiva



OUTLIERS

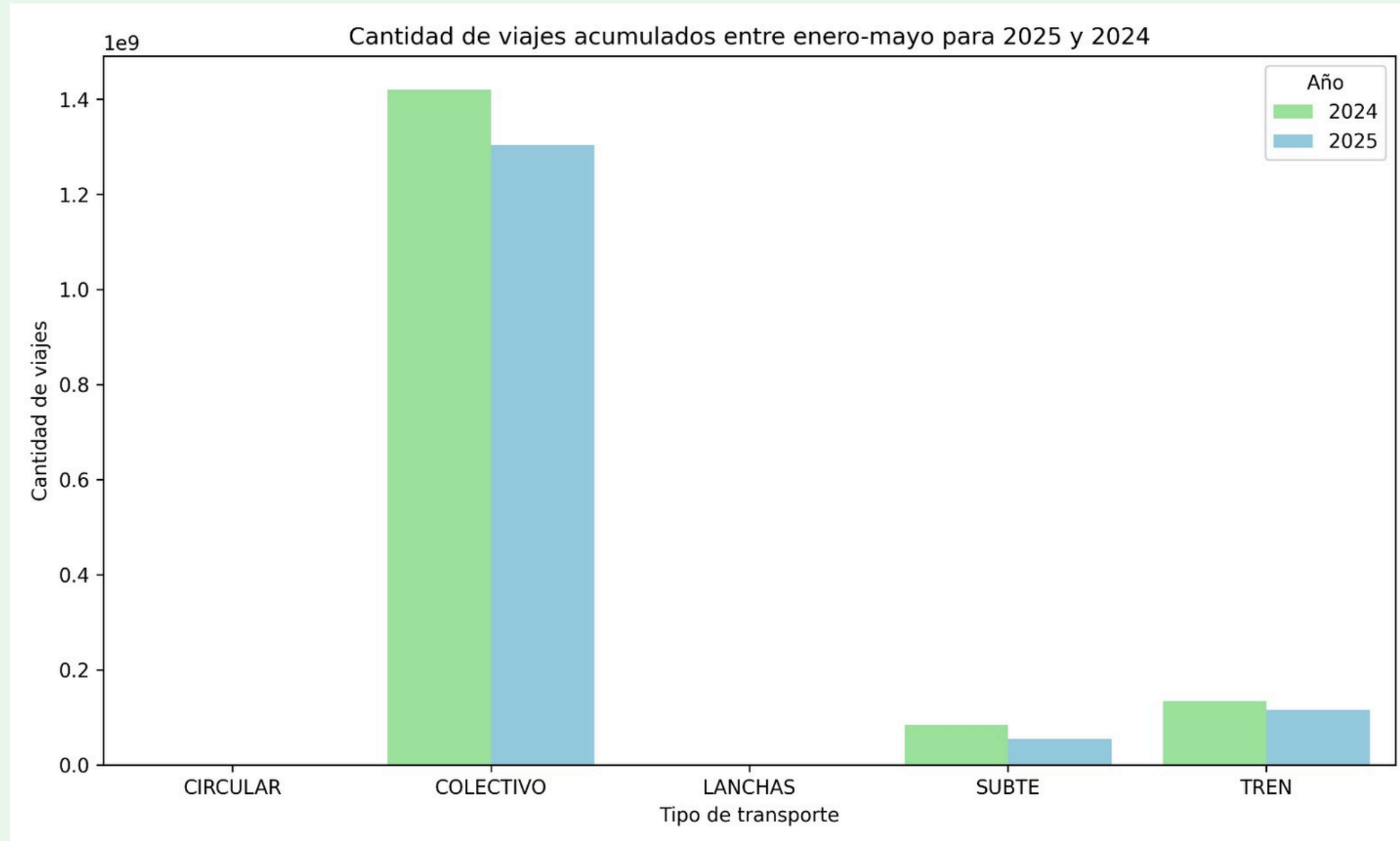
- En este caso decidimos no eliminarlos, ya que representan fenómenos reales que resultan fundamentales para entender el sistema de movilidad.
- Un ejemplo claro es el Área Metropolitana de Buenos Aires (AMBA), que concentra aproximadamente el 30% de la población nacional.



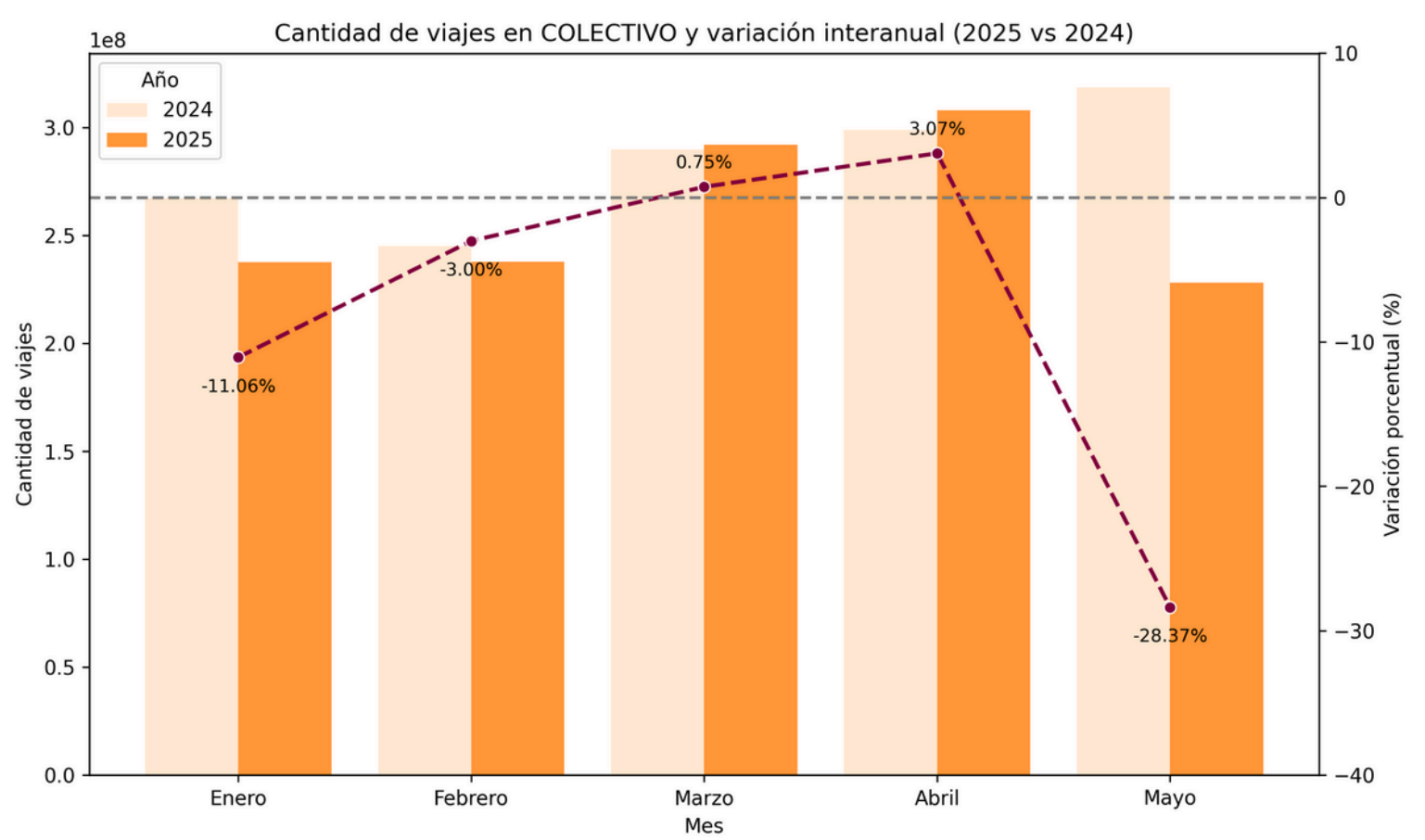
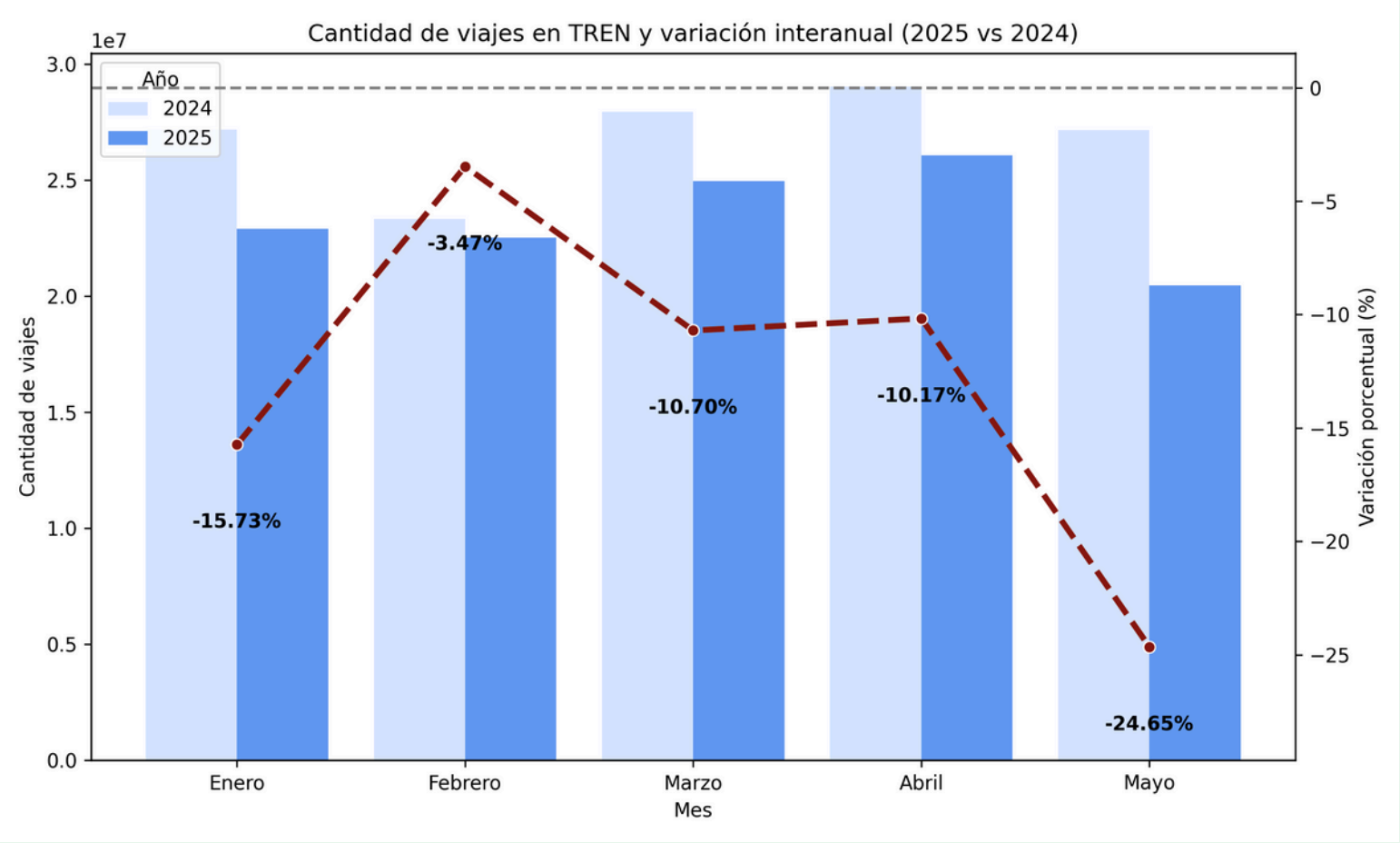
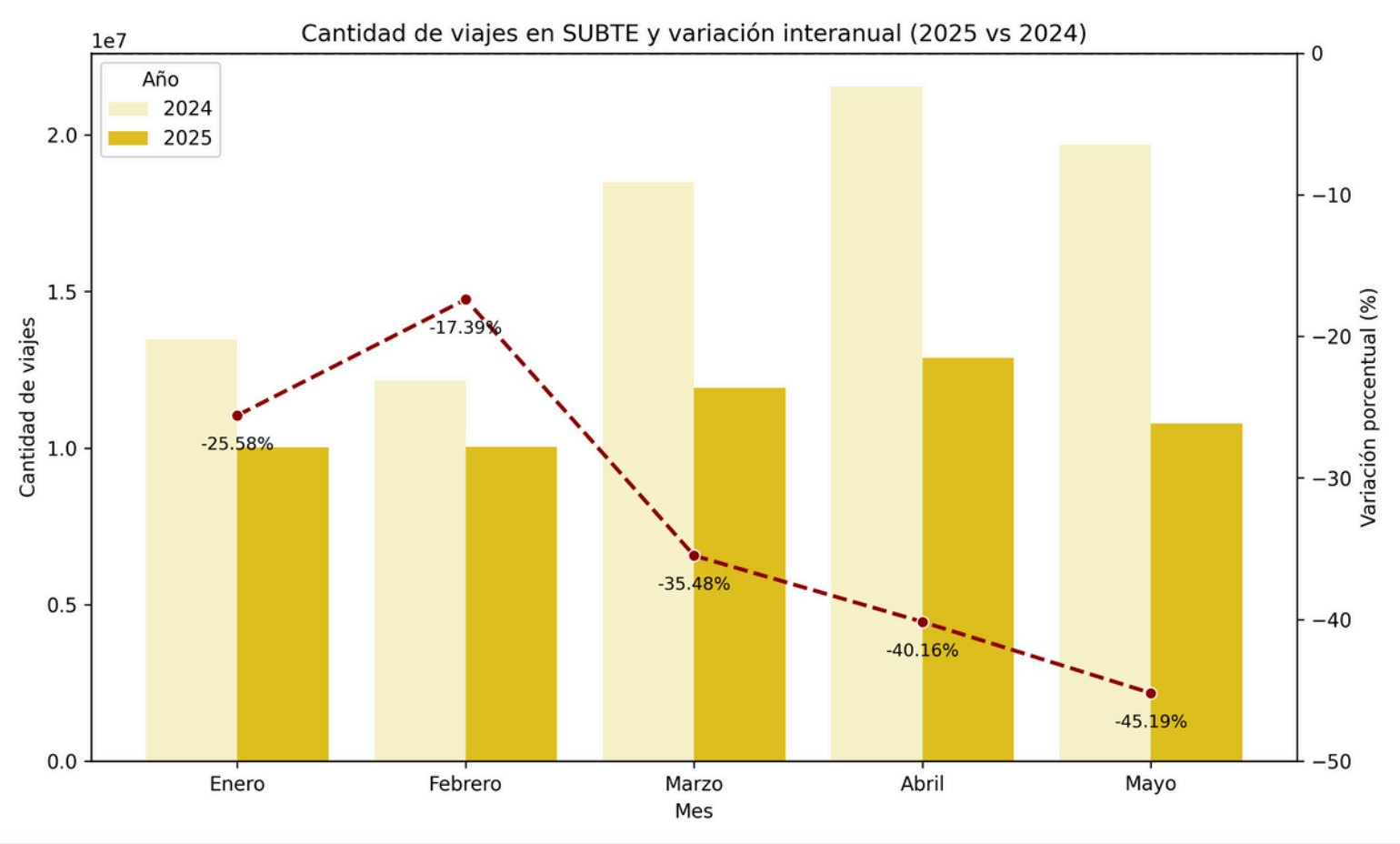
COMPARATIVO

De Enero – Mayo: 2025 vs 2024

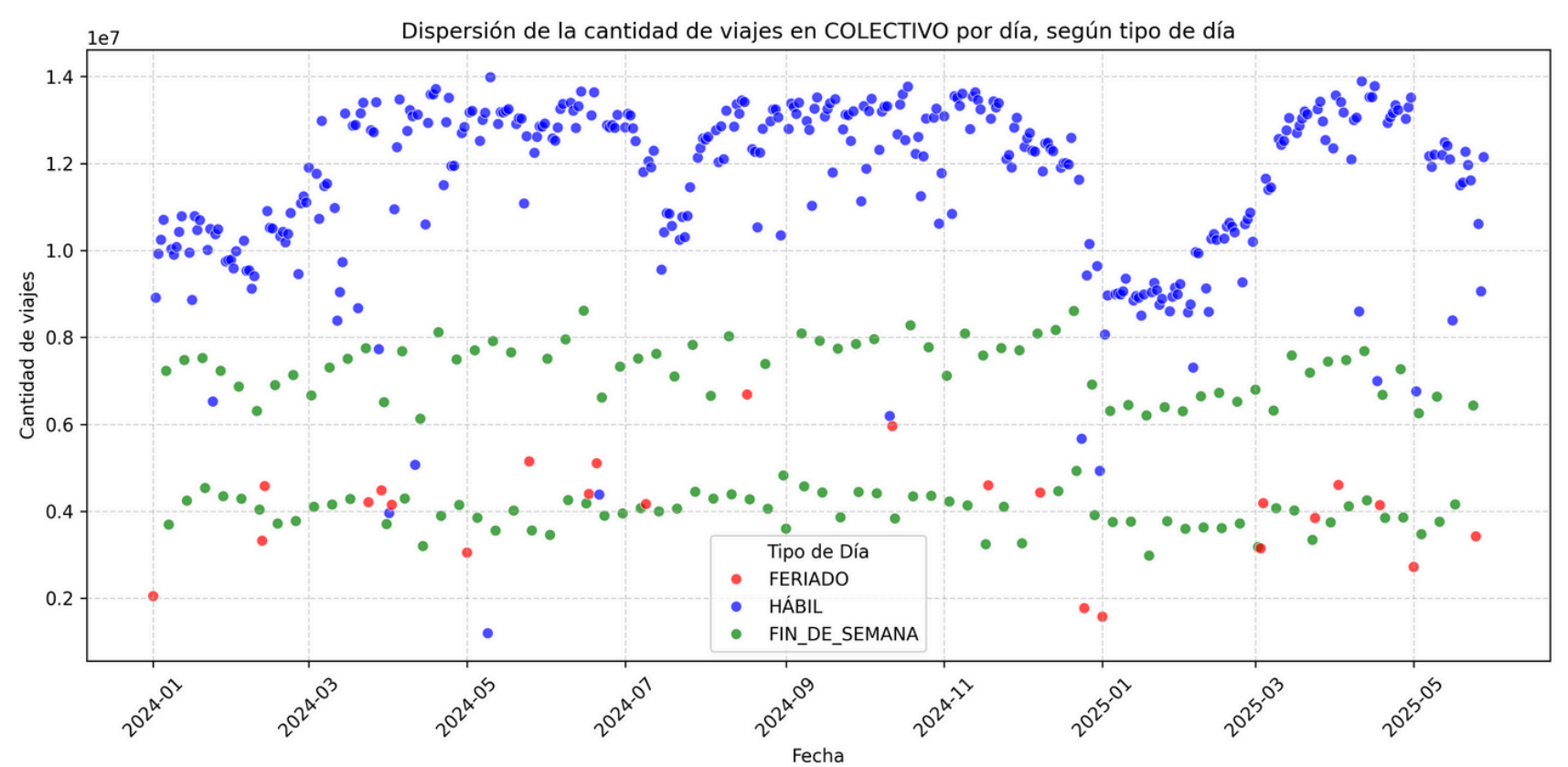
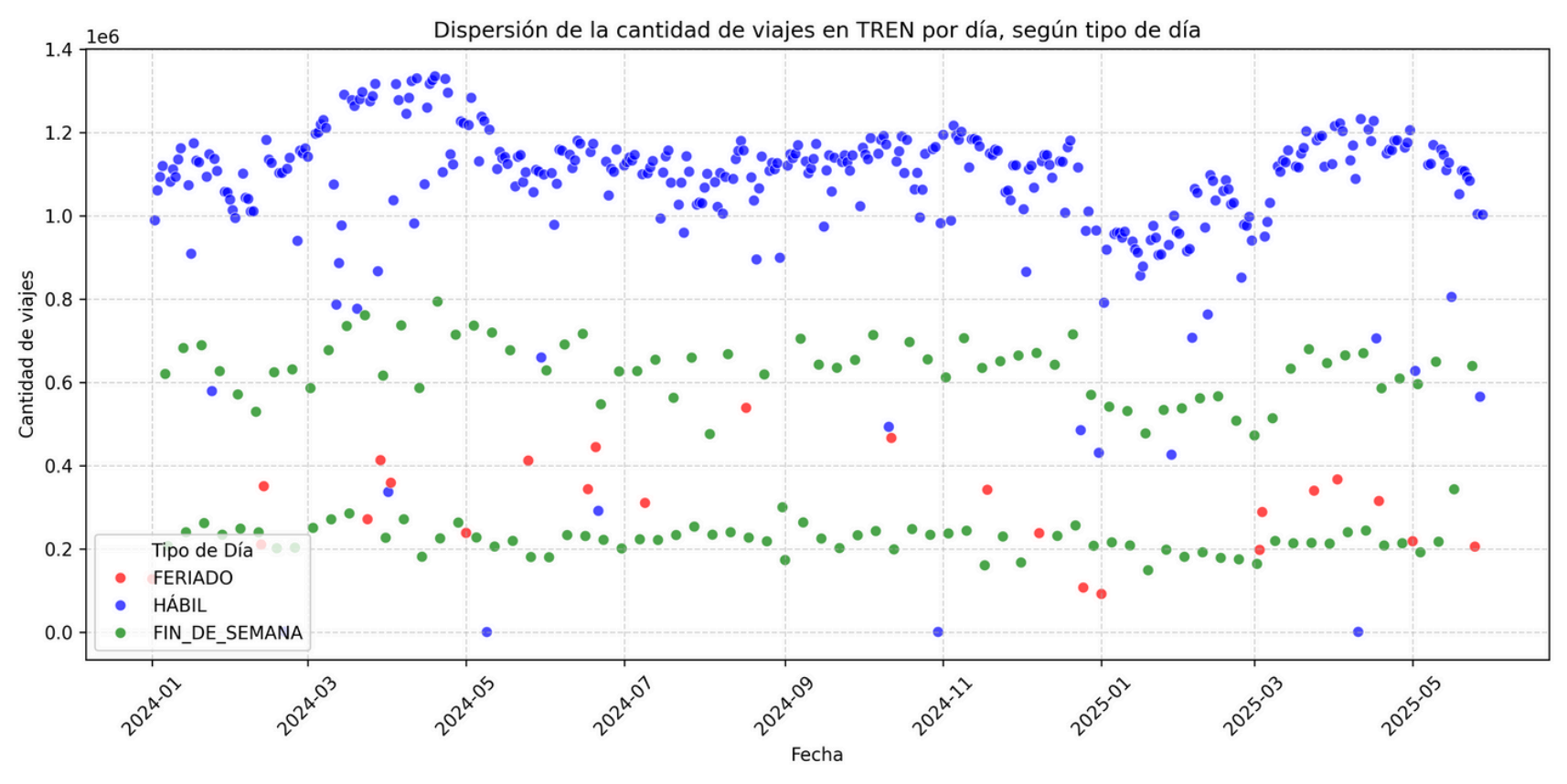
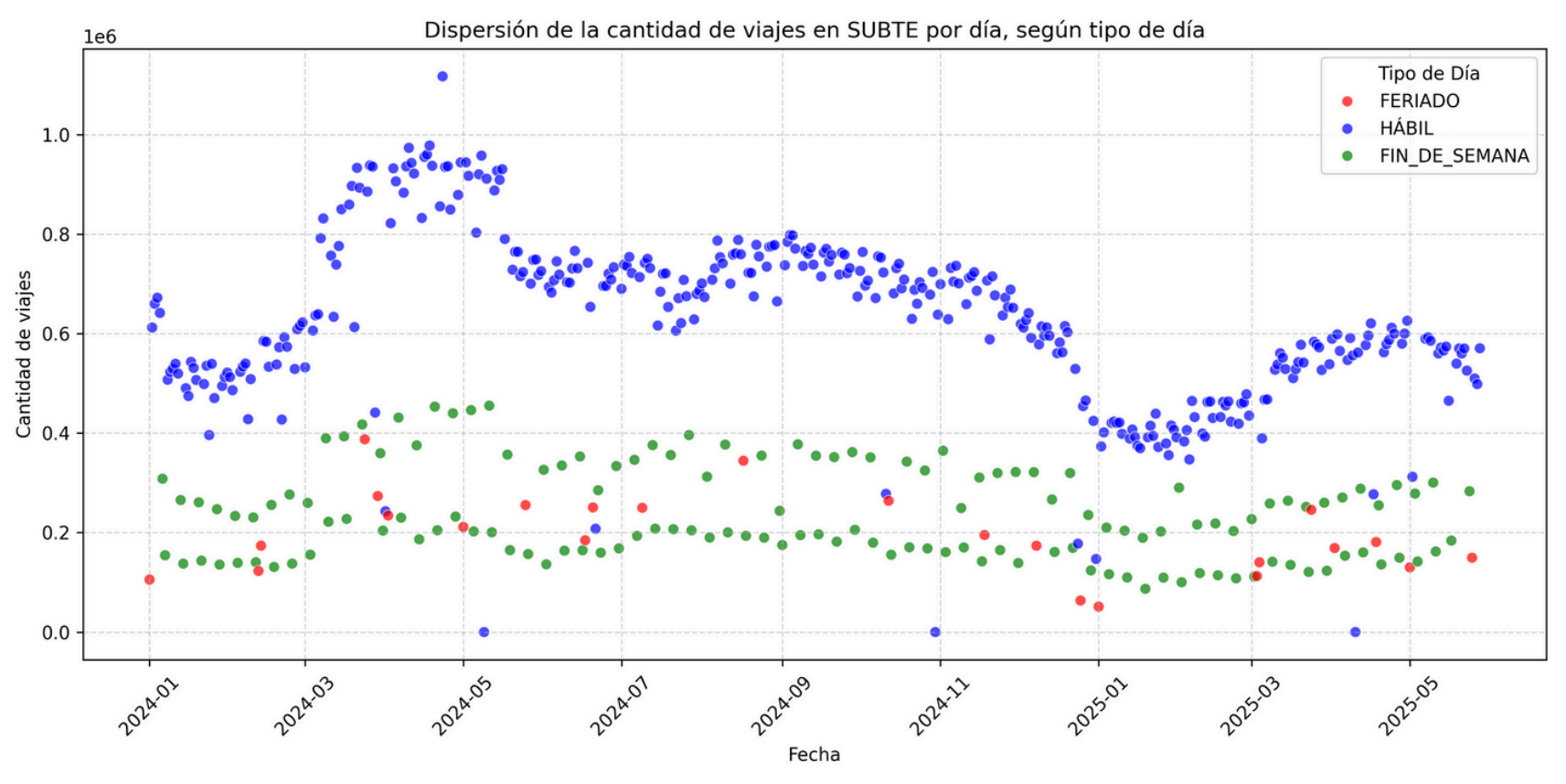
Cantidad de viajes acumuladas por tipo de transporte durante enero – mayo: 2025 vs 2024



Evolución de la cantidad de viajes por tipo de transporte durante enero – mayo: 2025 vs 2024



Evolución de la cantidad de viajes por día y por tipo de transporte durante enero – mayo: 2025 vs 2024



CONCLUSIONES

Análisis Exploratorio de la Demanda de Viajes SUBE
Año 2024/2025

CONCLUSIONES FINALES EDA

VIAJES REGISTRADOS TARJETA SUBE 2024



Estacionalidad marcada: menos viajes en enero, febrero, junio y diciembre por vacaciones escolares y laborales.



Concentración en días hábiles: representan más del 80% de los viajes; fines de semana y feriados caen al 50% y 30%.



Dominio del colectivo: cerca del 90% de los viajes a nivel nacional, con patrón similar en todos los días.



Valores atípicos esperables: por volumen y variabilidad nacional; normalización aplicada para análisis posterior.



Diferencias por zona: subte con baja dispersión, restringido a CABA.

Conclusiones Finales: Utilidad y Aportes del Análisis Exploratorio de Datos (EDA)

Las 10 provincias más activas,



Se evidencia una tendencia clara en los días hábiles con mayor concentración.



Caída de la demanda durante los feriados.



Este patrón es especialmente marcado en provincias como Buenos Aires y Corrientes, donde la movilidad está muy atada al ritmo productivo.



CONCLUSIONES

Conclusiones Finales: Utilidad y Aportes del Análisis Exploratorio de Datos (EDA)

Caída acumulada i.a.

Cambios en los patrones de conducta

- Nuevos medios de pago
- Trabajo remoto y mixto

- Futuros estudios: incorporación de datos adicionales.
- Políticas públicas, planificación de infraestructura





Muchas gracias.

GRUPO 04

Fica Millán, Yesica – DNI 27.624.956

Petraroia, Franco – DNI 27.161.862

Miranda Charca, Florencia – DNI 41.398.768

De Los Rios, Raúl – DNI 37.741.686

