

Proyecto

Entrega 1

Por: Sofía Catalina Galindo, Diego Alejandro
Herrera, Anamaria Leguizamón.



Programa

- Entendimiento del negocio
- Selección de los datos a utilizar
- Exploración de los datos
- Reporte calidad de datos
- Filtros, limpieza y transformación inicial
- Bono

Entendimiento del negocio



[Volver a la página de agenda](#)

Selección de los datos a utilizar

Arrestos

ARREST_KEY: Clave única que identifica cada arresto.

ARREST_DATE: Fecha en la que ocurrió el arresto.

PD_CD: Código del departamento de policía asociado al delito.

PD_DESC: Descripción del delito.

KY_CD: Código de clasificación del delito.

OFNS_DESC: Descripción de la clasificación del delito.

LAW_CODE: Código de la ley asociada al delito.

LAW_CAT_CD: Categoría legal del delito.

ARREST_BORO: Distrito donde se realizó el arresto.

ARREST_PRECINCT: Número de la comisaría donde se realizó el arresto.

JURISDICTION_CODE: Código de jurisdicción.

AGE_GROUP: Grupo de edad del individuo arrestado.

PERP_SEX: Sexo del individuo arrestado.

PERP_RACE: Raza del individuo arrestado.

X_COORD_CD: Coordenada X de ubicación del arresto.

Y_COORD_CD: Coordenada Y de ubicación del arresto.

Latitude: Latitud de la ubicación del arresto.

Longitude: Longitud de la ubicación del arresto.

New Georeferenced Column: Columna adicional de georreferencia (puede contener información adicional sobre la ubicación).



Arrestos

ARREST_KEY: Clave única que identifica cada arresto.

ARREST_DATE: Fecha en la que ocurrió el arresto.

PD_CD: Código del departamento de policía asociado al delito.

PD_DESC: Descripción del delito.

KY_CD: Código de clasificación del delito.

OFNS_DESC: Descripción de la clasificación del delito.

LAW_CODE: Código de la ley asociada al delito.

LAW_CAT_CD: Categoría legal del delito.

ARREST_BORO: Distrito donde se realizó el arresto.

ARREST_PRECINCT: Número de la comisaría donde se realizó el arresto.

JURISDICTION_CODE: Código de jurisdicción.

AGE_GROUP: Grupo de edad del individuo arrestado.

PERP_SEX: Sexo del individuo arrestado.

PERP_RACE: Raza del individuo arrestado.

X_COORD_CD: Coordenada X de ubicación del arresto.

Y_COORD_CD: Coordenada Y de ubicación del arresto.

Latitude: Latitud de la ubicación del arresto.

Longitude: Longitud de la ubicación del arresto.

New Georeferenced Column: Columna adicional de georreferencia (puede contener información adicional sobre la ubicación).



Accidentes

UNIQUE_ID: Identificación única del registro generado por el sistema. Clave primaria. (Número)

COLLISION_ID: Código de identificación de colisión. Clave foránea, coincide con unique_id de la tabla de Colisiones. (Número)

CRASH_DATE: Fecha de ocurrencia de la colisión. (Fecha y Hora)

CRASH_TIME: Hora de ocurrencia de la colisión. (Texto)

VEHICLE_ID: Código de identificación del vehículo asignado por el sistema. (Texto)

STATE_REGISTRATION: Estado donde está registrado el vehículo. (Texto)

VEHICLE_TYPE: Tipo de vehículo según la categoría de vehículo seleccionada . (Texto)

VEHICLE_MAKE: Marca del vehículo. (Texto)

VEHICLE_MODEL: Modelo del vehículo. (Texto)

VEHICLE_YEAR: Año de fabricación del vehículo. (Texto)

TRAVEL_DIRECTION: Dirección en la que se desplazaba el vehículo. (Texto)

VEHICLE_OCCUPANTS: Número de ocupantes del vehículo. (Número)

DRIVER_SEX: Género del conductor. (Texto)

DRIVER_LICENSE_STATUS: Estado de la licencia del conductor, permiso, sin licencia. (Texto)

DRIVER_LICENSE_JURISDICTION: Estado donde se emitió la licencia de conducir. (Texto)

PRE_CRASH: Acción previa al choque: ir en línea recta, girar a la derecha, pasar, retroceder, etc. (Texto)

POINT_OF_IMPACT: Ubicación en el vehículo del punto inicial de impacto (es decir, lado del conductor, parte trasera del lado del pasajero, etc.). (Texto)

VEHICLE_DAMAGE: Ubicación en el vehículo donde ocurrió la mayor parte del daño. (Texto)

VEHICLE_DAMAGE_1: Ubicaciones de daños adicionales en el vehículo. (Texto)

VEHICLE_DAMAGE_2: Ubicaciones de daños adicionales en el vehículo. (Texto)

VEHICLE_DAMAGE_3: Ubicaciones de daños adicionales en el vehículo. (Texto)

PUBLIC_PROPERTY_DAMAGE: Propiedad pública dañada (Sí o No). (Texto)

PUBLIC_PROPERTY_DAMAGE_TYPE: Tipo de propiedad pública dañada (por ejemplo, señal, cerca, poste de luz, etc.). (Texto)

CONTRIBUTING_FACTOR_1: Factores que contribuyen a la colisión para el vehículo designado. (Texto)

CONTRIBUTING_FACTOR_2: Factores que contribuyen a la colisión para el vehículo designado. (Texto)



Exploración de Datos

The background features a dark teal color with two large, overlapping hexagonal shapes. The top-right hexagon is a medium teal color, and the bottom-left hexagon is a light lime green color. The text 'Exploración de Datos' is centered horizontally and partially overlaid by these shapes.

Datos de arrestos del Departamento de Policía de Nueva York hasta la fecha

Table +

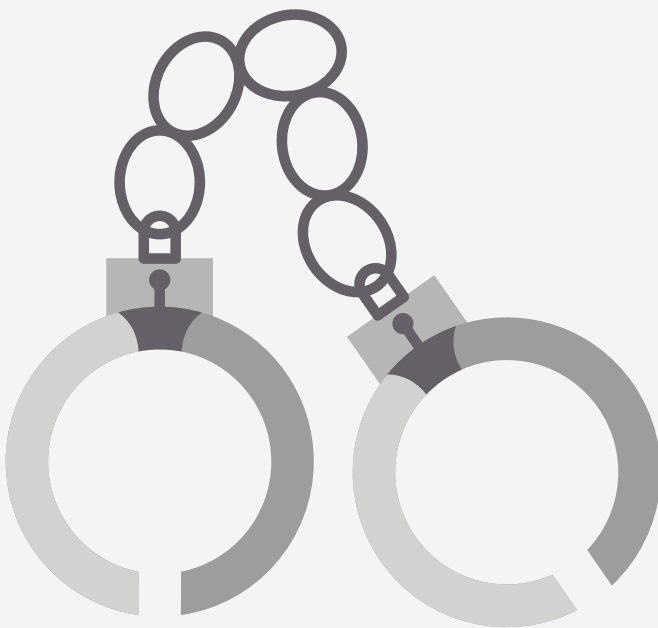
	ARREST_KEY	ARREST_DATE	PD_CD	PD_DESC	RY_CD	OFNS_DESC	LAW_CODE	LAW_CAT_CD	ARREST_BORO	ARREST_PRECINCT	JURISC
1	261265483	2023-01-03	397	ROBBERY,OPEN AREA UNCLASSIFIED	105	ROBBERY	PL 1600500	F	B	49	0
2	261271301	2023-01-03	105	STRANGULATION 1ST	106	FELONY ASSAULT	PL 1211200	F	S	120	0
3	261336449	2023-01-04	397	ROBBERY,OPEN AREA UNCLASSIFIED	105	ROBBERY	PL 1601001	F	K	61	0
4	261328047	2023-01-04	105	STRANGULATION 1ST	106	FELONY ASSAULT	PL 1211200	F	Q	114	0
5	261417496	2023-01-05	244	BURGLARY,UNCLASSIFIED,UNKNOWN	107	BURGLARY	PL 1402000	F	B	44	0
6	261383093	2023-01-08	109	ASSAULT 2,1,UNCLASSIFIED	106	FELONY ASSAULT	PL 1200502	F	K	76	0
7	261611504	2023-01-09	263	ARSON 2,3,4	114	ARSON	PL 1501500	F	B	49	71



10,000 rows | Truncated data | 5.99 seconds runtime

Refreshed 3 hours ago

224872



Datos de Colisiones de vehículos motorizados - Vehículos

Table

	UNIQUE_ID	COLLISION_ID	CRASH_DATE	CRASH_TIME	VEHICLE_ID	STATE_REGISTRATION	VEHICLE_TYPE	VEHICLE_MAKE	VEHICLE_MODEL	VEHICLE_YEAR
1	10385780	100201	2012-09-07	2024-04-10T09:03:00Z	1	NY	PASSENGER VEHICLE	null	null	null
2	19140702	4213082	2019-09-23	2024-04-10T08:15:00Z	0553ab4e-9500-4cbe-8d98-Md7189d5856	NY	Station Wagon/Sport Utility Vehicle	TOYT -CAR/SUV	null	2002
3	14887647	3307608	2015-10-02	2024-04-10T17:18:00Z	2	NY	TAXI	null	null	null
4	14889754	3308693	2015-10-04	2024-04-10T20:34:00Z	1	NY	PASSENGER VEHICLE	null	null	null
5	14400270	207666	2013-04-25	2024-04-10T21:15:00Z	1	NY	PASSENGER VEHICLE	null	null	null
6	17044639	3434155	2016-05-02	2024-04-10T17:35:00Z	219456	NY	4 dr sedan	MERZ -CAR/SUV	null	2015
7	19138701	4229067	2019-10-24	2024-04-10T13:15:00Z	c53b43d9-419a-4ab1-93d1-3f2979078d89	NY	Bus	FHMT-TRUCK/BUS	null	2006

9,445 rows | Truncated data | 26.31 seconds runtime

Refreshed 3 hours ago



EXPLORACIÓN DE LOS DATOS

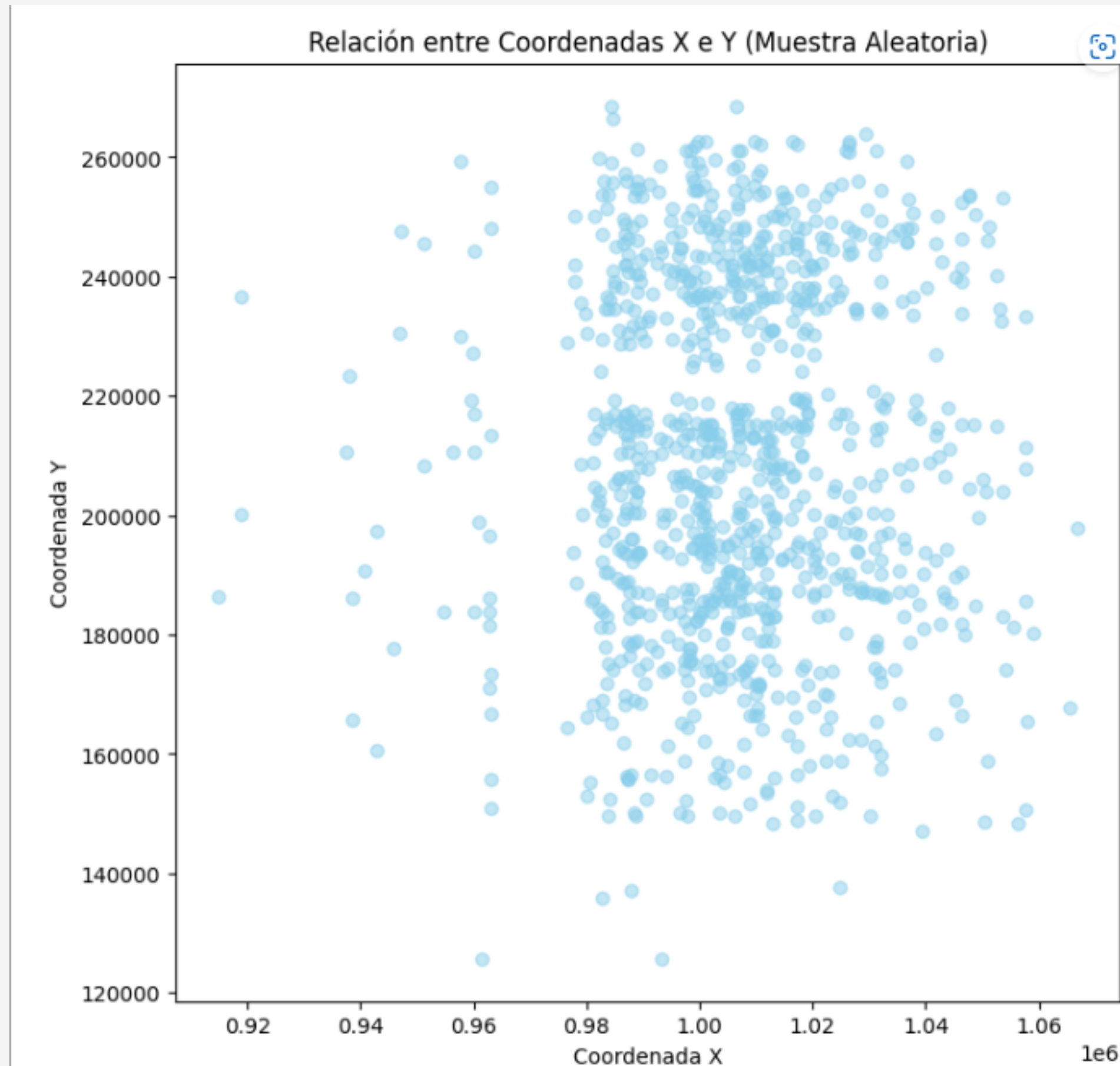
Arrestos

	summary	ARREST_KEY	PO_CD	PO_DESC	KEY_CD	OFFS_DESC	LAW_CODE	LAW_CAT_CD	ARREST_BORO	ARREST_PRECINCT	JURISDICTION_CODE
1	count	226872	226870	226872	226855	226872	226872	225273	226872	226872	226872
2	mean	2.70647924840011168	424.7344011989245	null	249.3451323532653	null	null	9.0	null	63.43052910892486	0.9285367960788462
3	stdev	5304010.298147567	274.4753806048603	null	147.6867326476052	null	null	0.0	null	34.635045257003999	7.538548508006556
4	min	261180920	1	(null)	101	(null)	(null)	(null)	8	1	0
5	max	279779734	997	WEAPONS,MFR,TRANSPORT,ETC.	995	VEHICLE AND TRAFFIC LAWS	YTL2130045	V	5	123	97

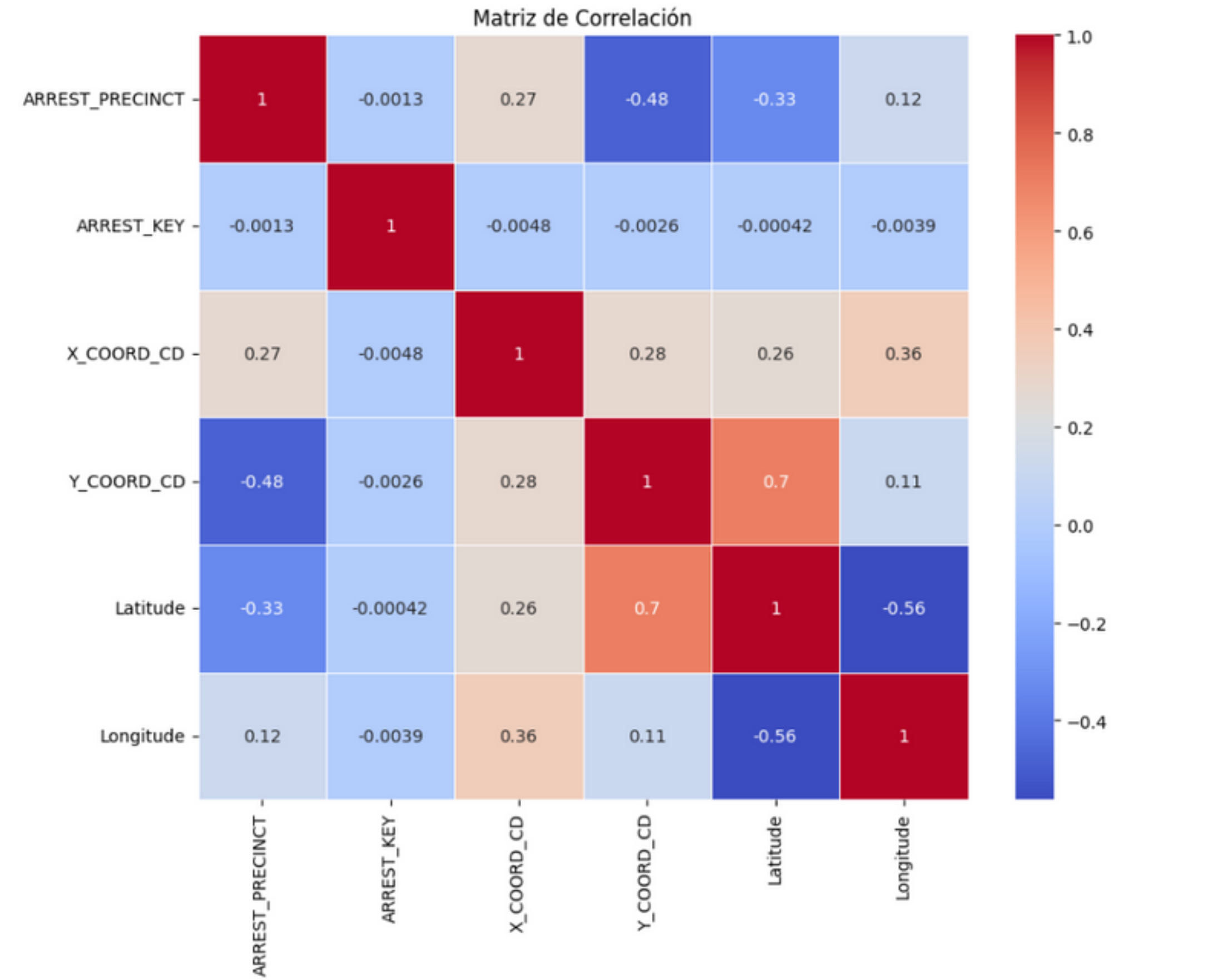
Colisiones

	summary	UNIQUE_ID	COLLISION_ID	VEHICLE_ID	STATE_REGISTRATION	VEHICLE_TYPE	VEHICLE_MAKE	VEHICLE_MODEL	VEHICLE_YEAR	TRAVEL_DIRECTION	VEH
1	count	4168842	4168842	4168842	3866446	3933610	2290650	51430	2270871	2501876	2368
2	mean	1.6541496848649338E7	3175097.827277215	128216.67765904864	null	189024.83333333334	Infinity	1184.5892857142858	2015.123295717999	null	885.4
3	stddev	3346797.193883878	1497127.8992604834	304779.2461164517	null	1161277.609248284	NaN	1540.0758575027387	148.2844405723924	null	9093
4	min	111711	22	000019b8-1c6f-45da-9913-efaa6e183326	AB	Time mope	'CHINA'	'FLEET'	1000	-	0
5	max	20626524	4713865	8886c7-5e7c-47de-b447-d8f54609ee9	ZZ	qum	zstv	yct6	20063	West	9999

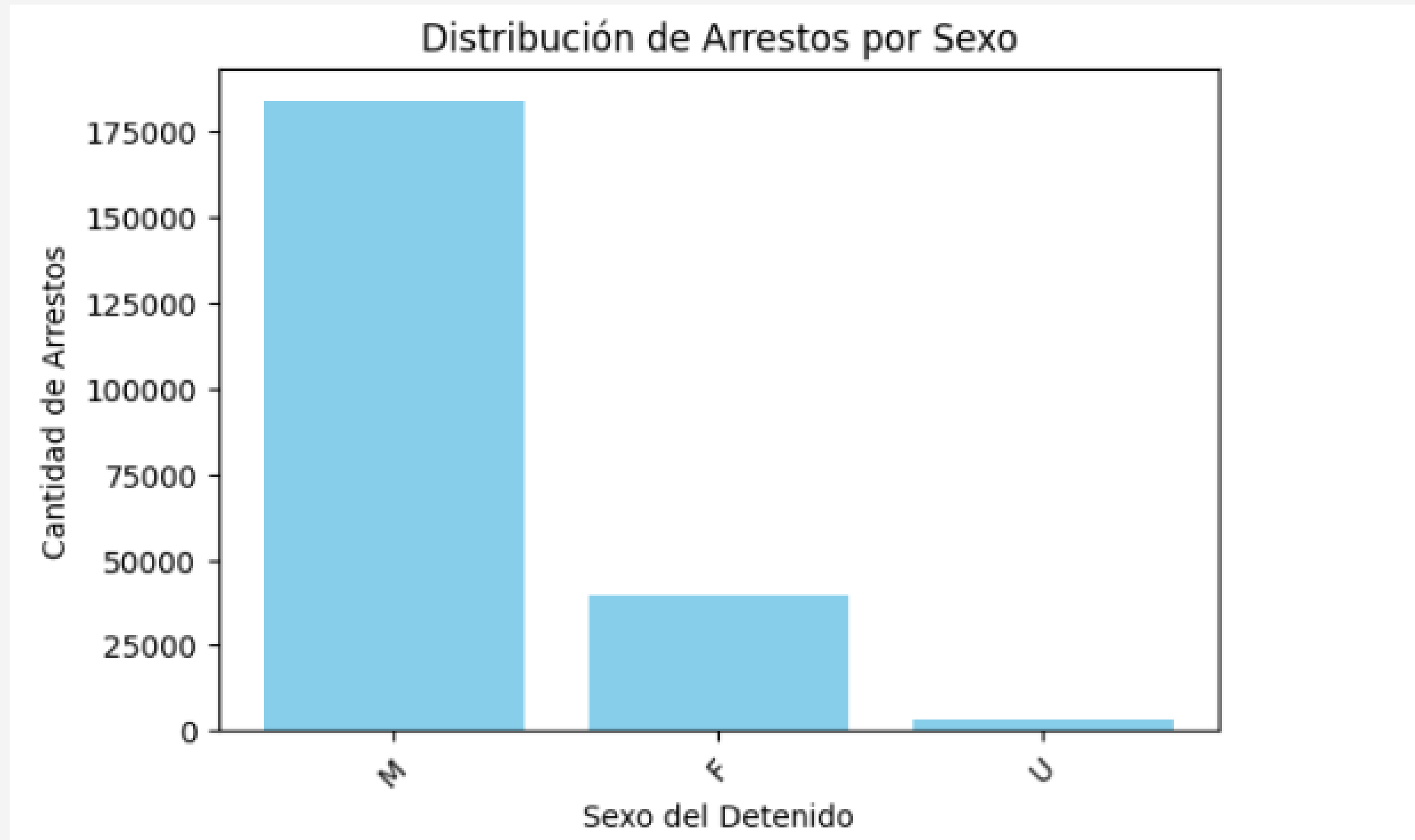
DISTRIBUCIÓN DE LOS DATOS ARRESTOS EN NY



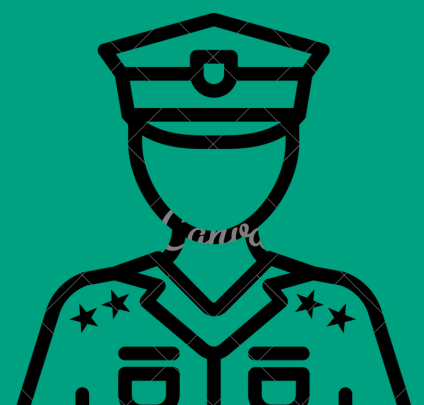
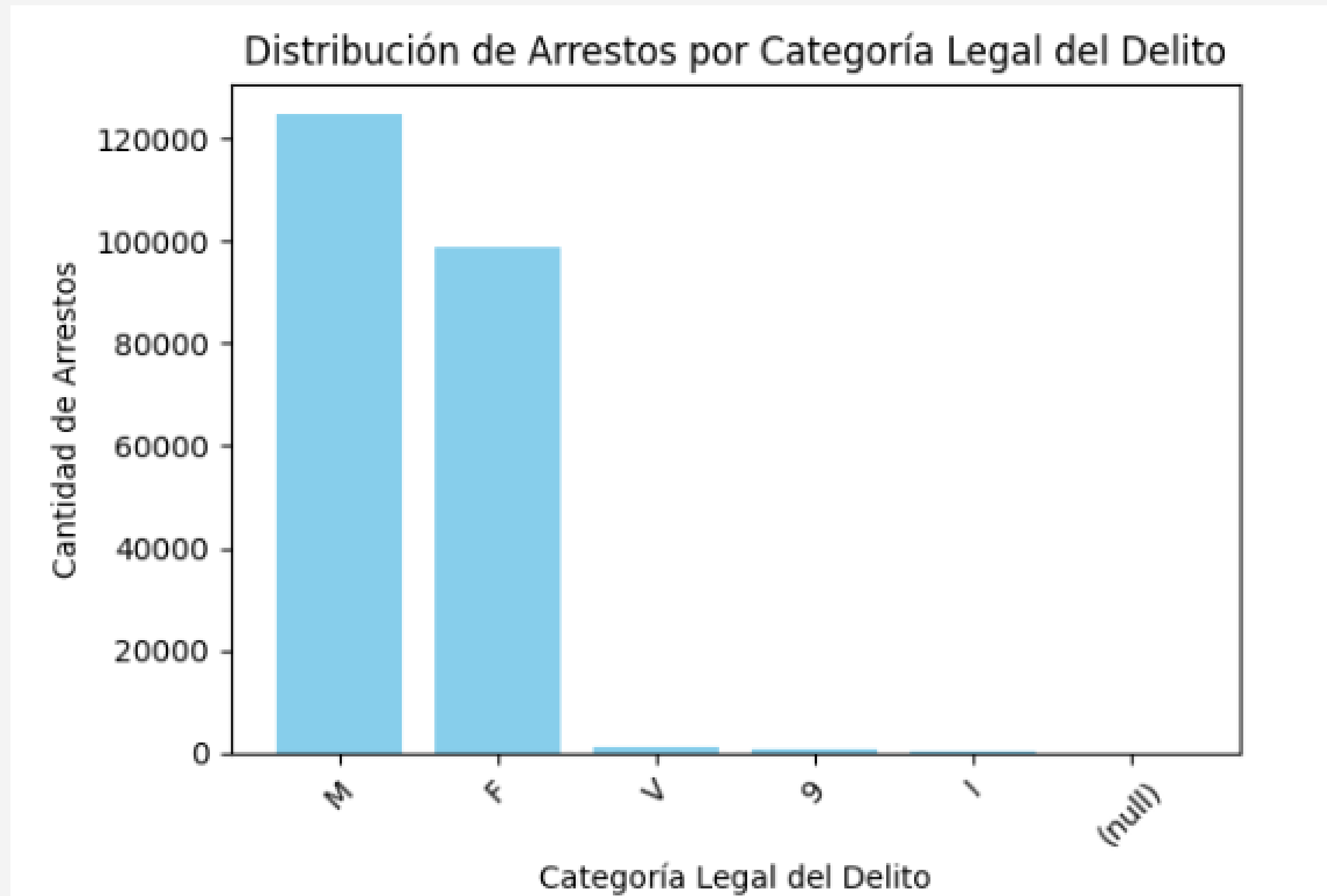
DISTRIBUCIÓN DE LOS DATOS ARRESTOS EN NY



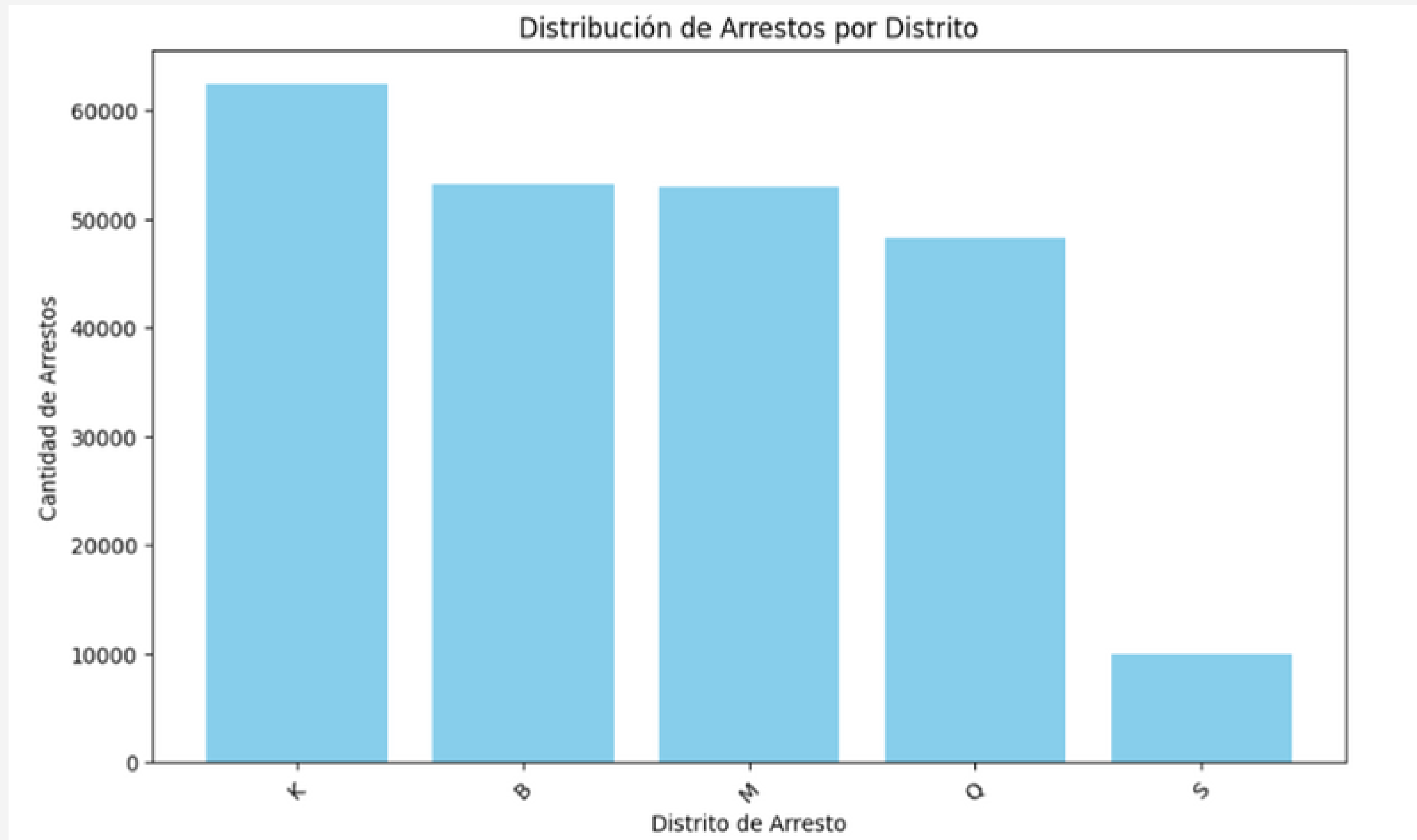
DISTRIBUCIÓN DE LOS DATOS ARRESTOS EN NY



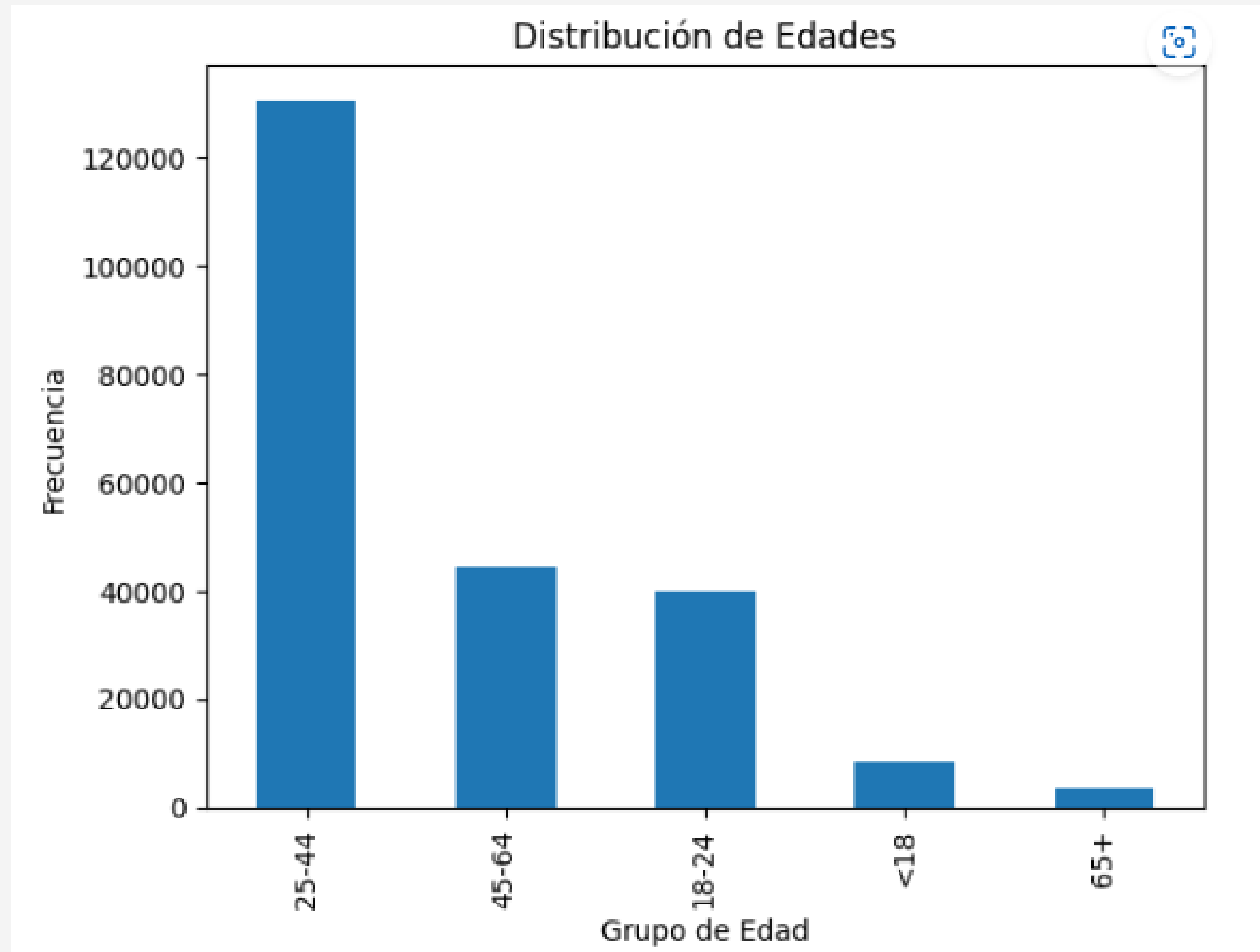
DISTRIBUCIÓN DE LOS DATOS ARRESTOS EN NY



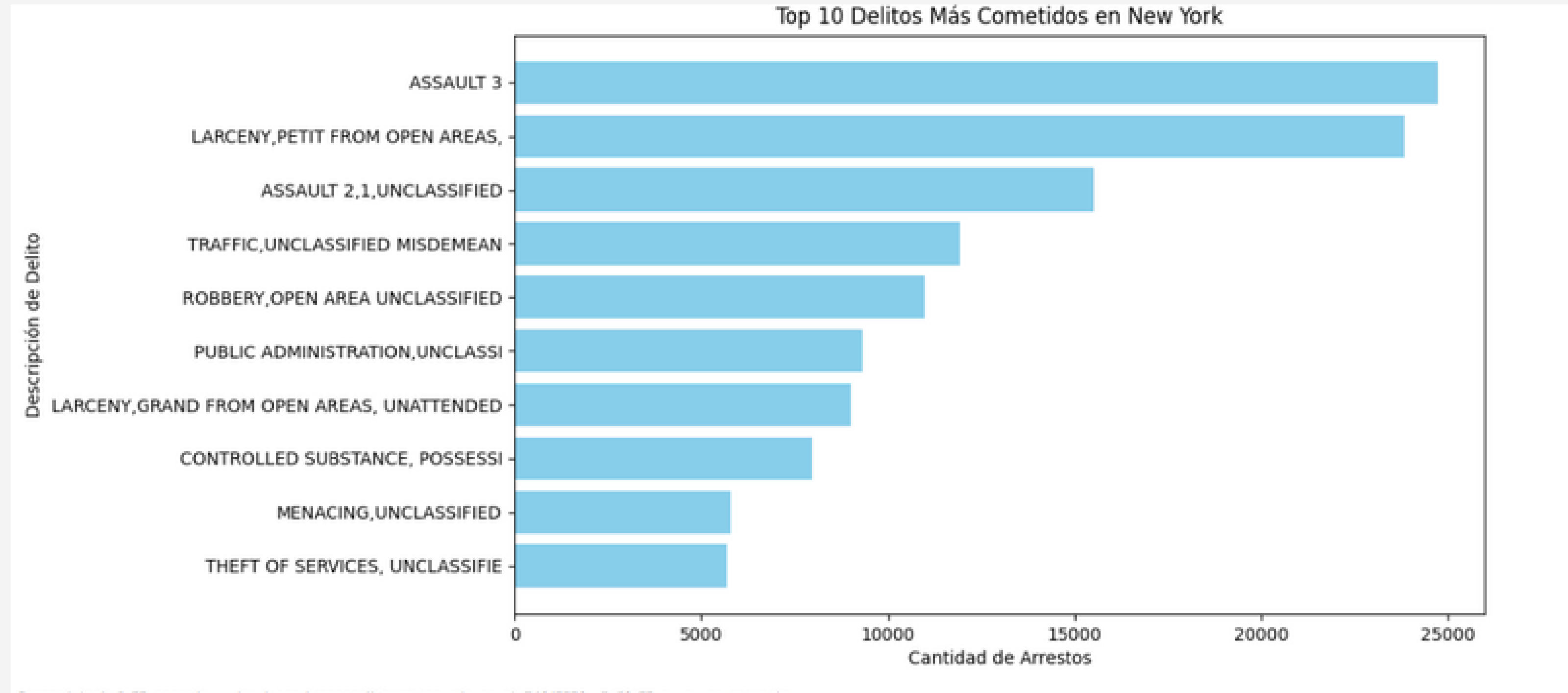
DISTRIBUCIÓN DE LOS DATOS ARRESTOS EN NY



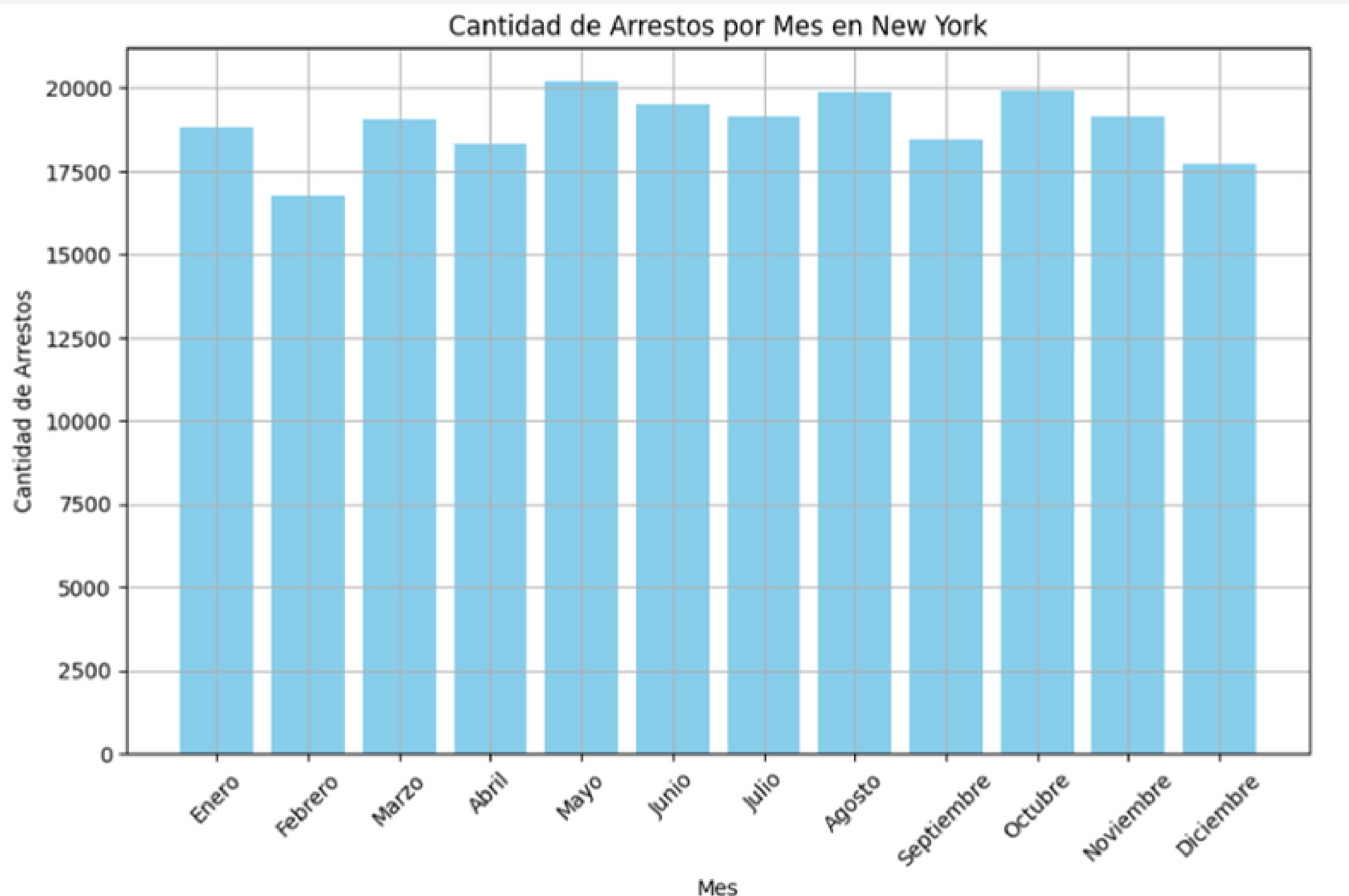
DISTRIBUCIÓN DE LOS DATOS ARRESTOS EN NY



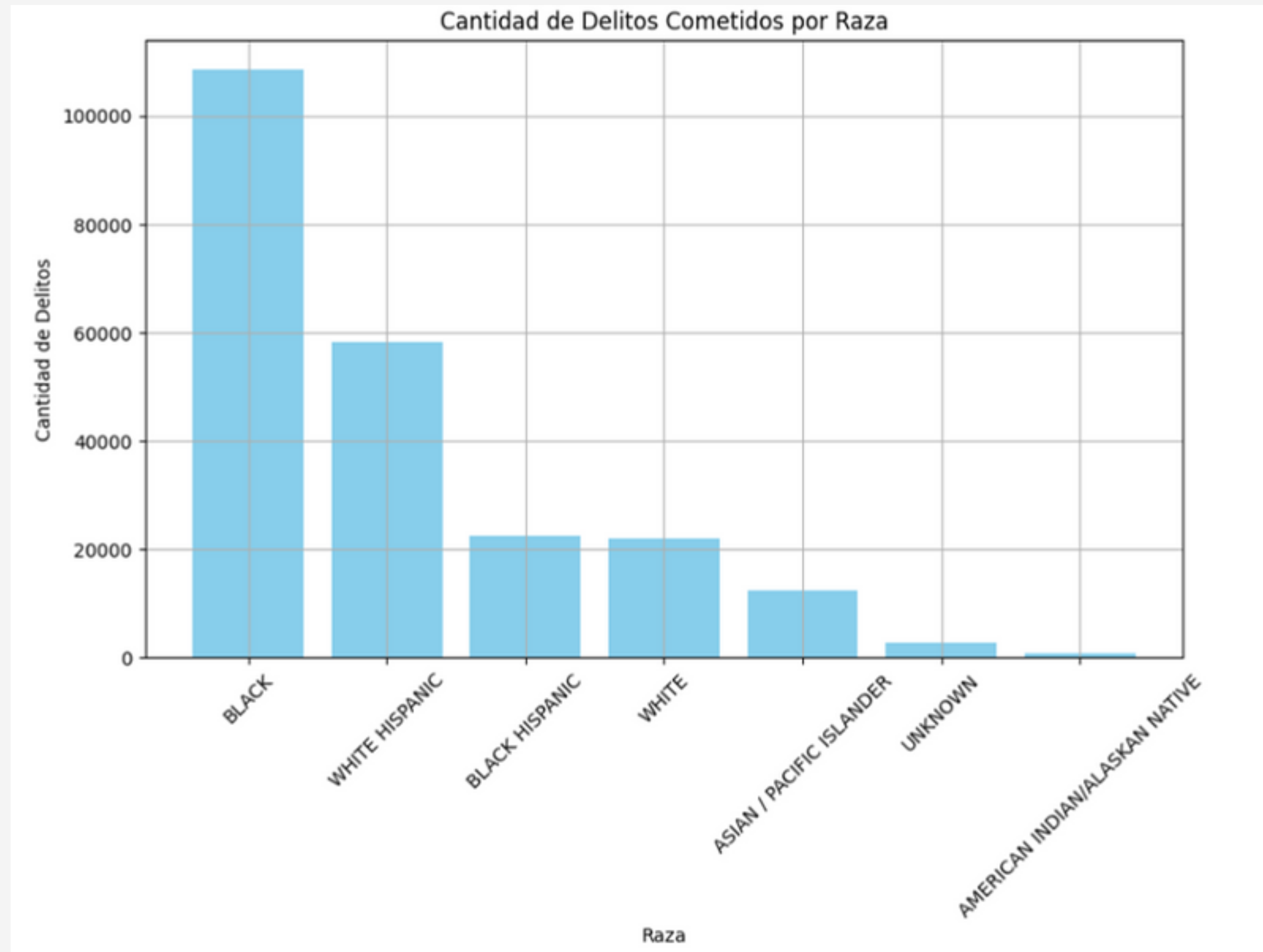
DISTRIBUCIÓN DE LOS DATOS ARRESTOS EN NY



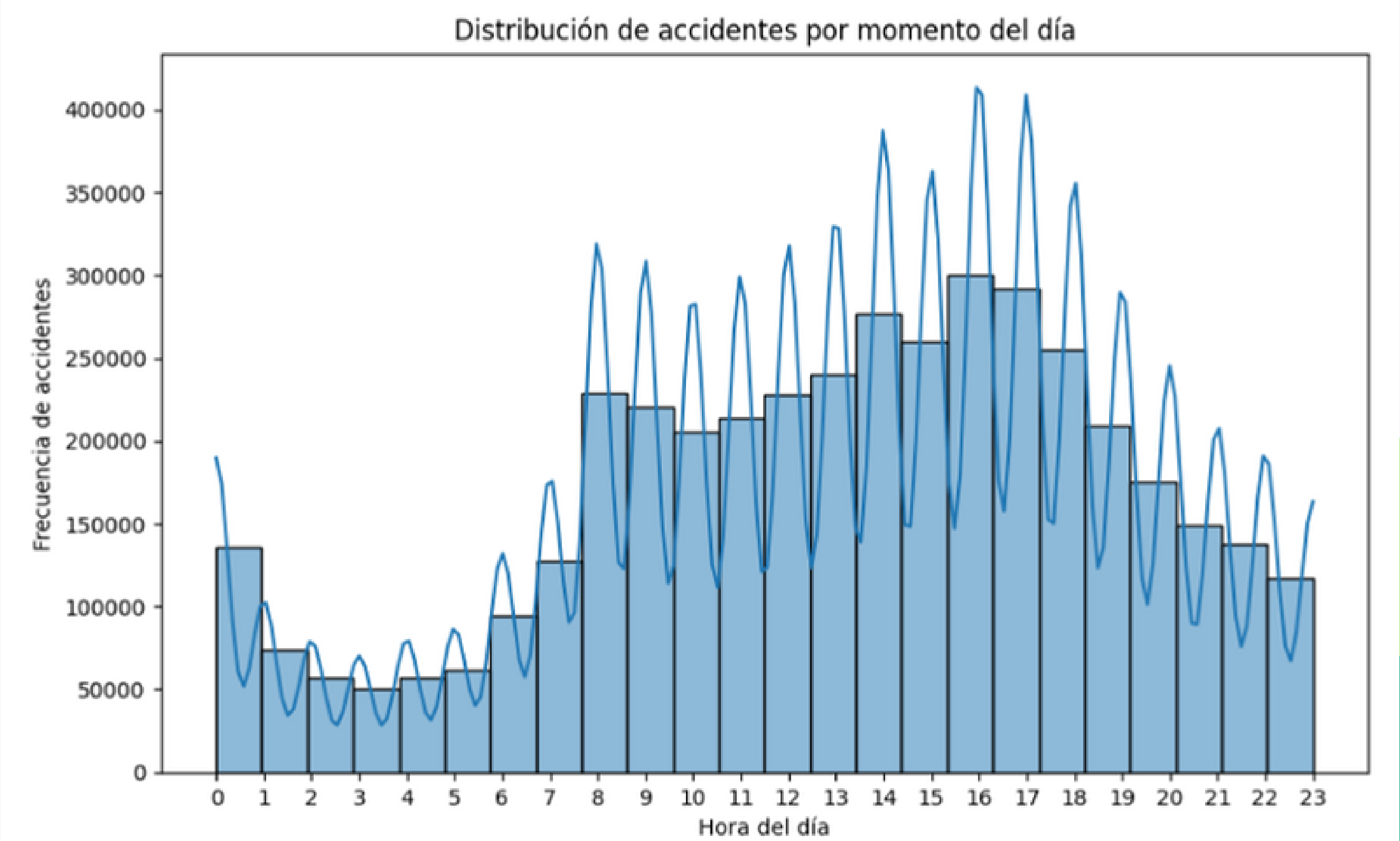
DISTRIBUCIÓN DE LOS DATOS ARRESTOS EN NY



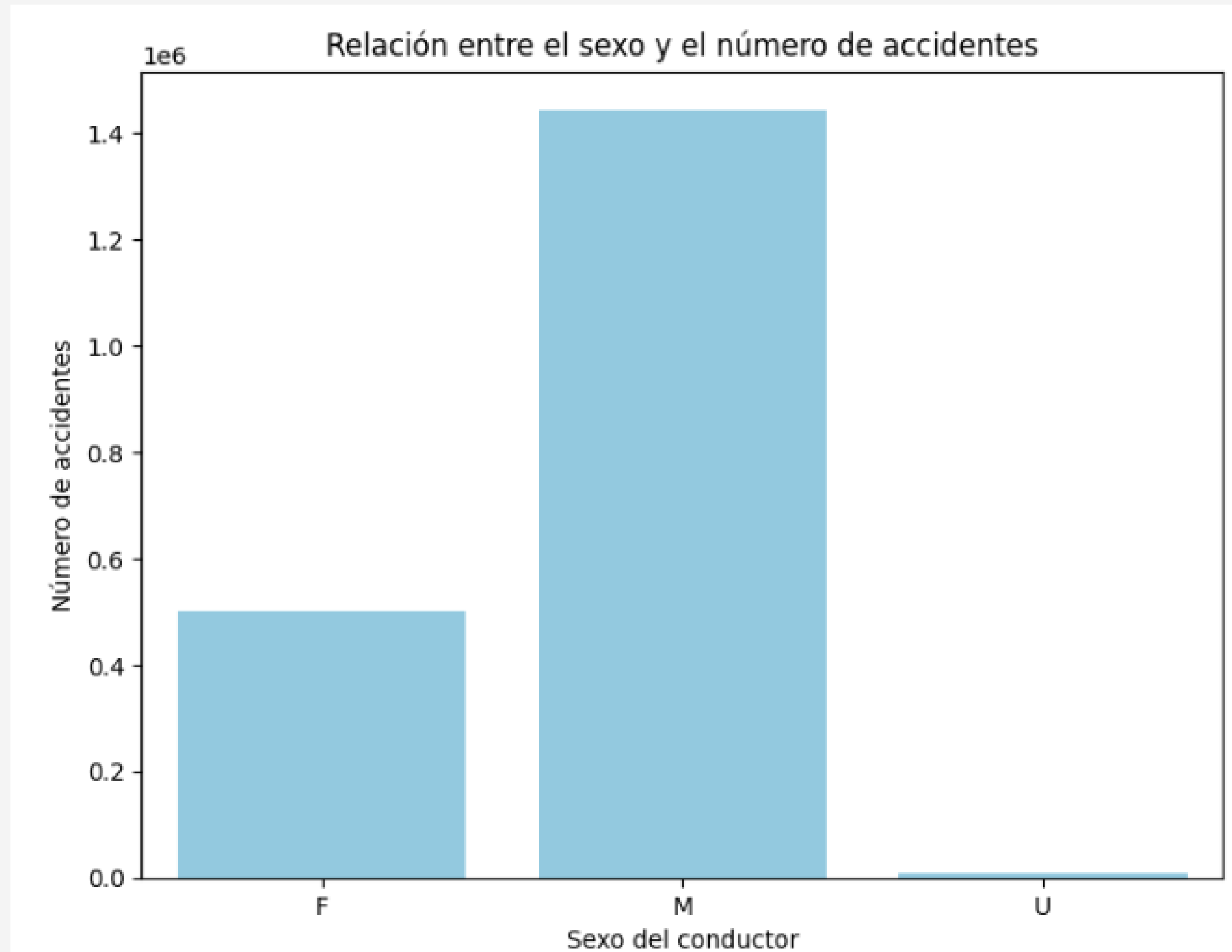
DISTRIBUCIÓN DE LOS DATOS ARRESTOS EN NY



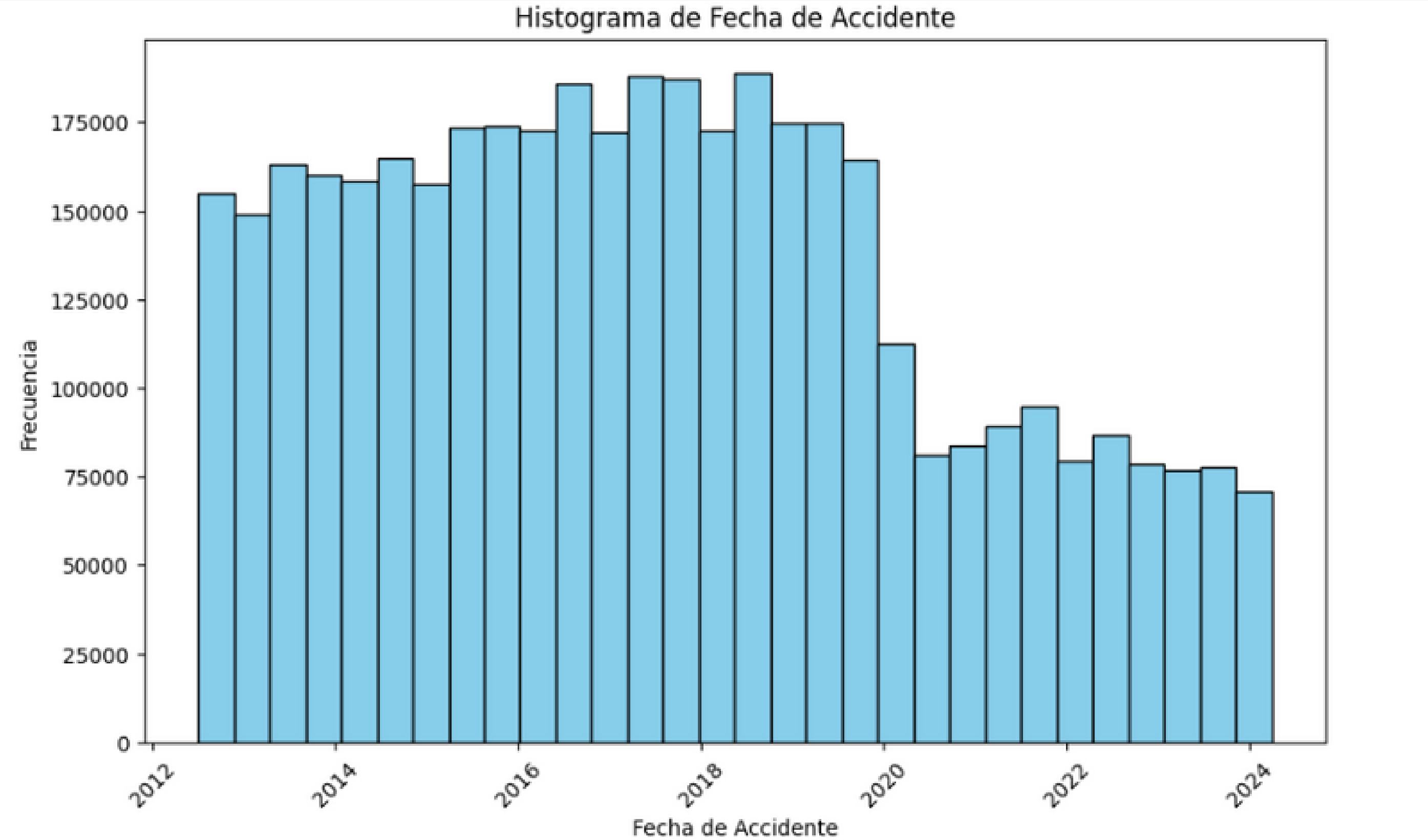
DISTRIBUCIÓN DE LOS DATOS COLISIONES DE VEHÍCULOS MOTORIZADOS



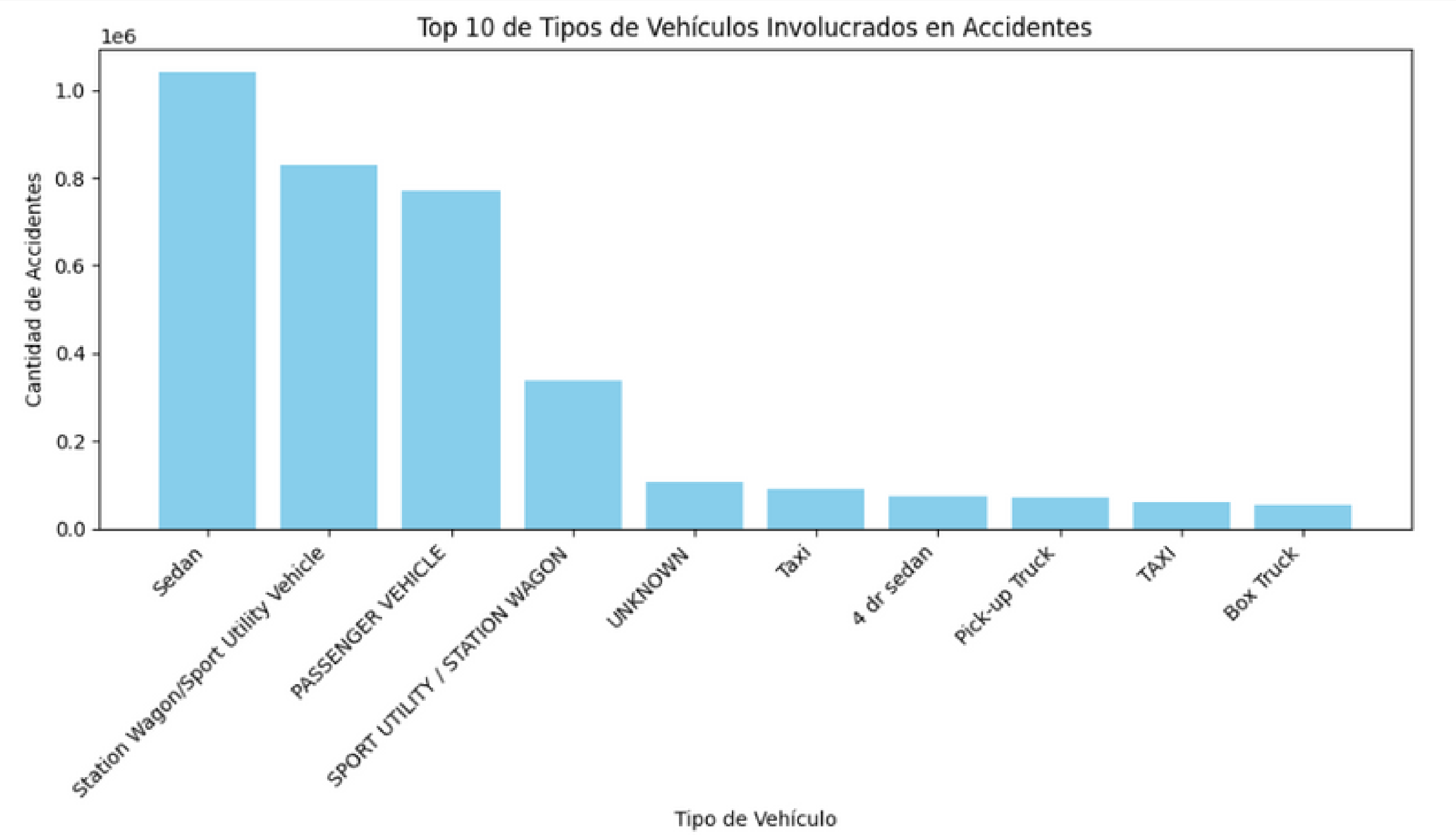
DISTRIBUCIÓN DE LOS DATOS COLISIONES DE VEHÍCULOS MOTORIZADOS



DISTRIBUCIÓN DE LOS DATOS COLISIONES DE VEHÍCULOS MOTORIZADOS



DISTRIBUCIÓN DE LOS DATOS COLISIONES DE VEHÍCULOS MOTORIZADOS



DISTRIBUCIÓN DE LOS DATOS COLISIONES DE VEHÍCULOS MOTORIZADOS

Ocupantes_Categoria	count
4 Ocupantes	37934
Más de 5 Ocupantes	12349
3 Ocupantes	91083
2 Ocupantes	317069
Desconocido	2190359
5 Ocupantes	13872
1 Ocupante	1506176

VEHICLE_TYPE	count
Sedan	1041870
Station Wagon/Spo...	830637
PASSENGER VEHICLE	770753
SPORT UTILITY / S...	337927
NULL	235232
UNKNOWN	105463
Taxi	91734
4 dr sedan	73577
Pick-up Truck	69211
TAXI	60785
Box Truck	53537
VAN	49720
Bike	47132
OTHER	45958
Bus	41444
LARGE COM VEH(6 O...	28604
SMALL COM VEH(4 T...	26683
BUS	25956

Posibles valores para PUBLIC_PROPERTY_DAMAGE_TYPE:

THREE ORANGE CONSTRUCTION BARRIERS

BUS STOP

FENCING DIVIDING W/B BELT PARKWAY @ ENTRANCE TO W/B BELT PARKWAY FROM SHORE PARKWAY

VEHICLE 1 WHITE 2013 INFINITI NY PLATE HEA8152

FENCE AND HOUSE STRUCTURE

METAL PIPE HANGING OFF OF OVERPASS

VEHICLE #1 RAN INTO FRONT LAWN OF 622 EAST 22 STREET CAUSING DAMAGE TO THE GARDEN

ONE BABY STROLLER

SMALL CHIGUAGUA DOG

IRON FENCE

FENCE SURROUNDING TREE ON SIDEWALK IN FRONT OF 788 HANCOCK

DAMAGE TO DRIVERS SIDE FENDER

YIELD SIGN

TREES, SIDEWALK, GRASS, FENCE, PLANETS, WIRES, GUTTER, SIDING FOR THE HOUSE, CHIMNEY, HOSE.

DAMAGE TO BRICK LANDING FOR BASEMENT STAIRS. UNK AMOUNT OF DAMAGE.

BRICK FENCING AND LAWN

BRICK FENCE DAMAGE AND WOODEN FRAME AROUND TREE

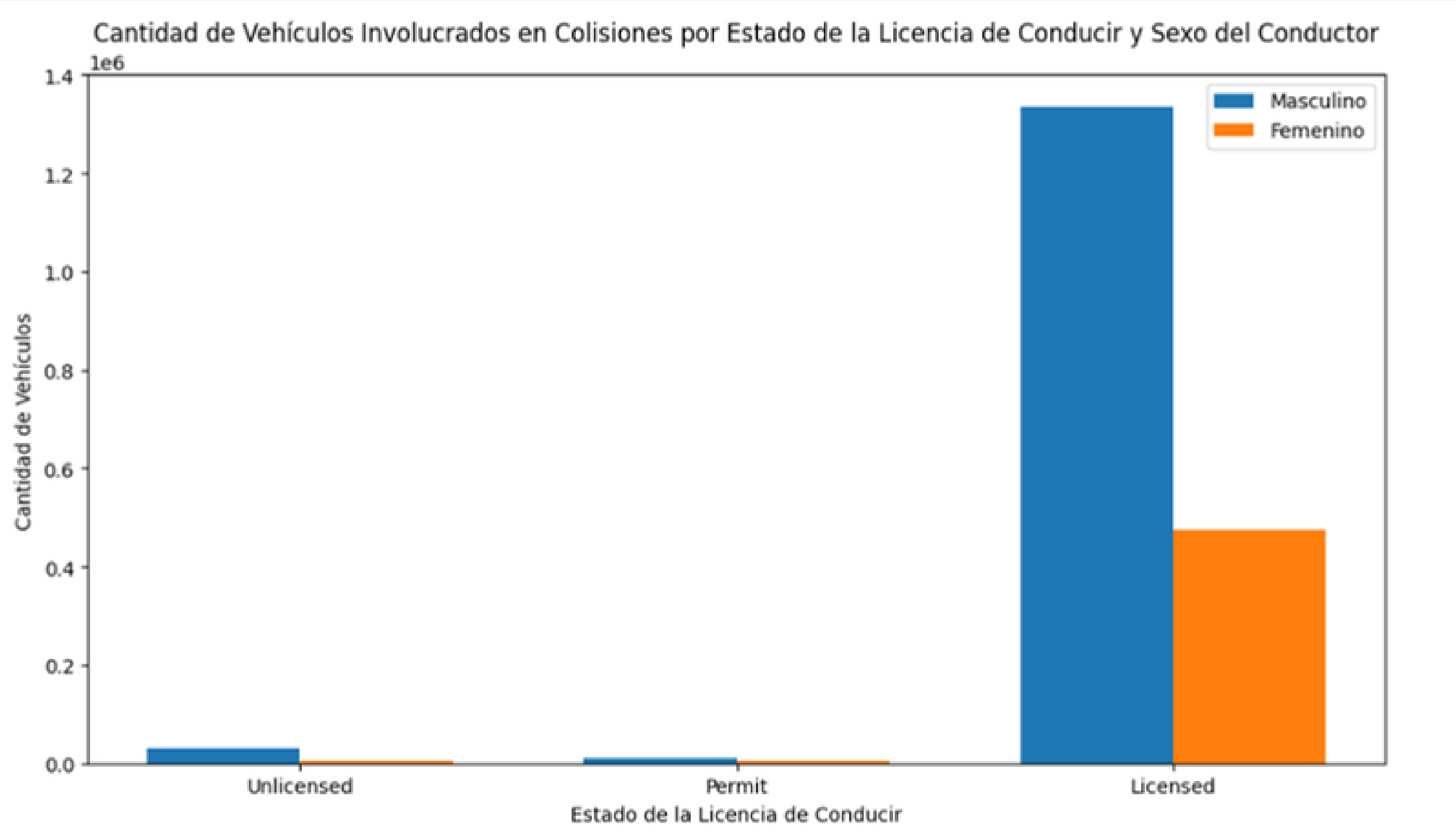
BRICK FRONT STOOP/AWNING/RAILING/FRONT DOOR

TRUCK TOO TALL FOR THE RAILROAD TRESTLE BRIDGE, WHILE PASSING UNDER THE BRIDGE, THE TRUCK GOT STUCK

VEHICLE ACCIDENT WHICH COLLIDED IN TO COMPLAINTS FENCE DESTROYING



DISTRIBUCIÓN DE LOS DATOS COLISIONES DE VEHÍCULOS MOTORIZADOS



Reporte Calidad de Datos

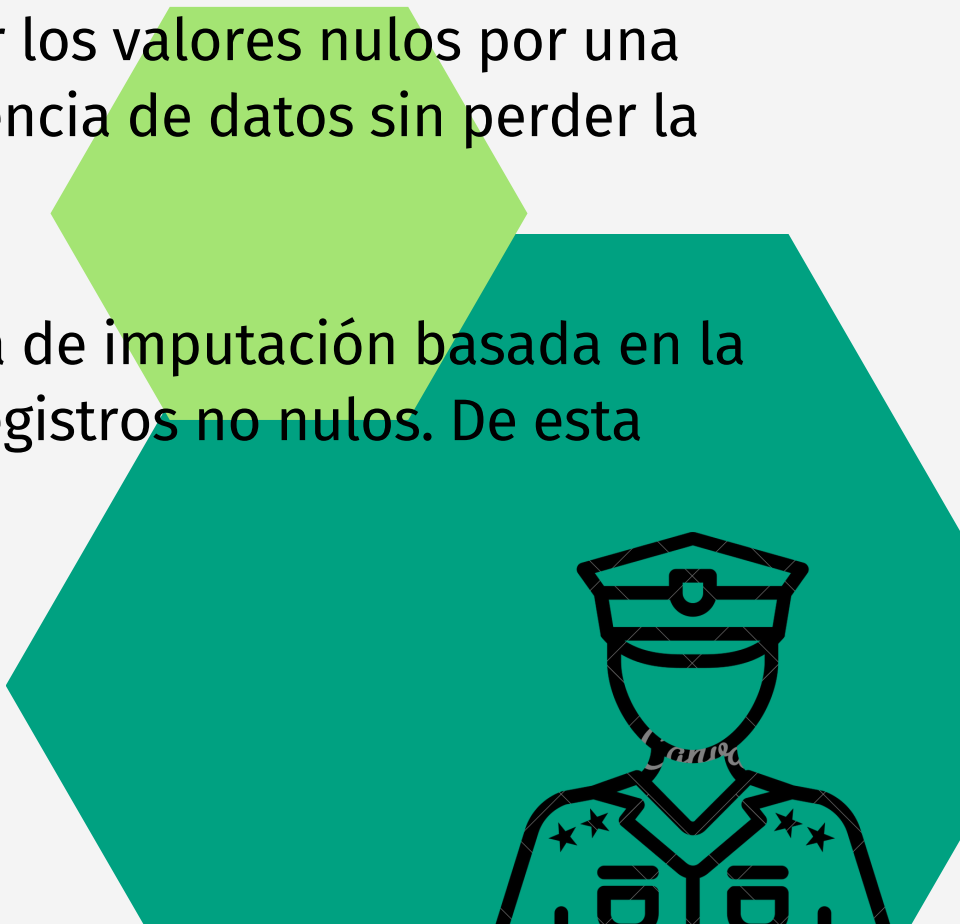
REPORTE DE CALIDAD DE DATOS SOBRE ARRESTOS:

```
1 from pyspark.sql.functions import count, when, isnan
2
3 # Luego puedes usar count y otras funciones sin problemas
4 df1.select([count(when(col(c).isNull(), c)).alias(c) for c in df1.columns]).show()
```

[ARREST_KEY]	[ARREST_DATE]	[PD_CD]	[PD_DESC]	[KY_CD]	[OFNS_DESC]	[LAW_CODE]	[LAW_CAT_CD]	[ARREST_BORO]	[ARREST_PRECINCT]	[JURISDICTION_CODE]	[AGE_GROUP]	[PERP_SEX]	[PERP_RACE]	[X_COORD_CD]	[Y_COORD_CD]	[Latitude]	[Longitude]	[New Georeferenced Column]	[ARREST_YEAR]	[ARREST_MONTH]
0	0	2	0	17	0	0	1599	0	0	0	0	0	0	0	0	0	0	0	0	0

PD_CD y KY_CD: Estas columnas representan variables categóricas con códigos. Una opción sería reemplazar los valores nulos por una nueva categoría que indique "Código desconocido". Esto nos permite conservar la información sobre la ausencia de datos sin perder la integridad del conjunto.

LAW_CAT_CD: Esta variable categórica también presenta valores faltantes. Aquí, podríamos aplicar la técnica de imputación basada en la moda. Esto implica reemplazar los valores faltantes por la categoría legal más frecuente observada en los registros no nulos. De esta manera, mantenemos la coherencia en nuestros datos y minimizamos la pérdida de información.



REPORTE DE CALIDAD DE DATOS SOBRE ACCIDENTES VEHICULARES

=====														

UNIQUE_ID COLLISION_ID CRASH_DATE CRASH_TIME VEHICLE_ID STATE_REGISTRATION VEHICLE_TYPE VEHICLE_MAKE VEHICLE_MODEL VEHICLE_YEAR TRAVEL_DIRECTION VEHICLE_OCCUPANTS DRIVER_SEX DRIVER_L														
ICENSE_STATUS DRIVER_LICENSE_JURISDICTION PRE_CRASH POINT_OF_IMPACT VEHICLE_DAMAGE VEHICLE_DAMAGE_1 VEHICLE_DAMAGE_2 VEHICLE_DAMAGE_3 PUBLIC_PROPERTY_DAMAGE PUBLIC_PROPERTY_DAMAGE_TYP														
E CONTRIBUTING_FACTOR_1 CONTRIBUTING_FACTOR_2 CRASH_HOUR														

	0	0	0	0	0	302396	235232	1878192	4117412	1897871	1666966	1780789	2215737	
2304191		2299404	920140	1699970	1724222	2594754	2983615	3260601		1528858			4142850	
147285		1686788	0											

- para variables categóricas como **VEHICLE_TYPE**, **DRIVER_SEX** o **DRIVER_LICENSE_STATUS**, se podría asignar una categoría especial para los valores faltantes o utilizar técnicas de imputación basadas en la moda.
- Para variables numéricas como **VEHICLE_OCCUPANTS** o **VEHICLE_YEAR**, se podrían utilizar métodos de imputación como la media, la mediana o modelos de regresión.



Planteamiento de las preguntas

The background features a dark teal color with two large, overlapping hexagonal shapes. One hexagon is a medium teal color and points towards the right. The other is a light lime green color and is positioned below and to the left of the first one.

Accidentes

1. ¿Es frecuente que los autos tengan daños en lugares específicos después de un accidente?
2. ¿Existe un patrón de accidentes relacionado con modelos específicos de vehículos?
3. ¿Los accidentes tienden a ocurrir más en alguna dirección de viaje específica?
4. ¿Los vehículos involucrados en accidentes presentan patrones específicos en las fechas y horas de los choques que podrían ayudar a prevenir futuros incidentes?
5. ¿Existe un patrón en las acciones que realizaban los vehículos justo antes de los accidentes?

Arrestos

1. ¿Qué tipos de delitos son los más frecuentes y cómo se distribuyen geográficamente?
2. ¿Existe alguna relación entre el nivel de delito y las características demográficas de los sospechosos, como su grupo de edad o género?
3. ¿Cómo han variado las tasas de arrestos a lo largo del tiempo y si existen patrones temporales o estacionales en los diferentes tipos de delitos?
4. ¿Qué distritos policiales o localidades (boroughs) tienen las mayores tasas de arrestos y cuáles son los tipos de delitos predominantes en esas áreas?



Filtros, limpieza y transformación inicial

Arrestos

```
1 from pyspark.sql.functions import col
2
3 # Eliminar registros duplicados basados en las columnas especificadas
4 df1_sin_duplicados = df1.dropDuplicates(['ARREST_KEY', 'ARREST_DATE', 'PD_CD', 'PD_DESC', 'KY_CD', 'OFNS_DESC', 'LAW_CODE', 'LAW_CAT_CD'])
5
6 # Mostrar los registros duplicados si existen
7 registros_duplicados = df1.subtract(df1_sin_duplicados)
8 if registros_duplicados.count() > 0:
9     print("Registros duplicados:")
10     registros_duplicados.show(truncate=False)
11 else:
12     print("No hay datos duplicados.")
```

```
# Imputar valores 'Desconocido' en la columna PD_CD y KY_CD
df1 = df1.fillna({'PD_CD': 'Desconocido'})
df1 = df1.fillna({'KY_CD': 'Desconocido'})
```

LAW_CODE: Código de la ley asociada al delito.
LAW_CAT_CD: Categoría legal del delito.



Accidentes

```
df3 = df3.withColumn("VEHICLE_OCCUPANTS", df3["VEHICLE_OCCUPANTS"].cast("string"))
df3 = df3.withColumn("Ocupantes_Categoria", when(df3["VEHICLE_OCCUPANTS"] == "1", "1 Ocupante")
| | | | | .when(df3["VEHICLE_OCCUPANTS"] == "2", "2 Ocupantes")
| | | | | .when(df3["VEHICLE_OCCUPANTS"] == "3", "3 Ocupantes")
| | | | | .when(df3["VEHICLE_OCCUPANTS"] == "4", "4 Ocupantes")
| | | | | .when(df3["VEHICLE_OCCUPANTS"] == "5", "5 Ocupantes")
| | | | | .otherwise("Más de 5 Ocupantes"))
df3 = df3.fillna({"VEHICLE_OCCUPANTS": "0"})
df3 = df3.withColumn("Ocupantes_Categoria", when(df3["VEHICLE_OCCUPANTS"] == "0", "Desconocido")
| | | | | .otherwise(df3["Ocupantes_Categoria"]))
df3.groupby("Ocupantes_Categoria").count().show()
```

```
# Imputar valores 'Desconocido' en la columna PD_CD y KY_CD
df3 = df3.fillna({'STATE_REGISTRATION': 'Desconocido'})
df3 = df3.fillna({'VEHICLE_MAKE': 'Desconocido'})
df3 = df3.fillna({'VEHICLE_MODEL': 'Desconocido'})
df3 = df3.fillna({'DRIVER_SEX': 'N'})
df3 = df3.fillna({'DRIVER_LICENSE_JURISDICTION': 'Desconocido'})
```

```
from pyspark.sql.functions import col, mode

moda_damagev = df3.select(mode(col("VEHICLE_DAMAGE"))).rdd.flatMap(lambda x: x).collect()[0]
df3 = df3.fillna({"VEHICLE_DAMAGE": moda_damagev})

moda_damagep = df3.select(mode(col("PUBLIC_PROPERTY_DAMAGE"))).rdd.flatMap(lambda x: x).collect()[0]
df3 = df3.fillna({"PUBLIC_PROPERTY_DAMAGE": moda_damagep})

moda_damagept = df3.select(mode(col("PUBLIC_PROPERTY_DAMAGE_TYPE"))).rdd.flatMap(lambda x: x).collect()[0]
df3 = df3.fillna({"PUBLIC_PROPERTY_DAMAGE_TYPE": moda_damagept})
```





Bono

Web Scrapping

- Se utilizo la librería ´pandas´

```
import pandas as pd
#importar pandas, no se utiliza selenium ya que la informacion esta en formato tabla y se puede extraer facilmente.

tablas = pd.read_html("https://www.health.ny.gov/statistics/cancer/registry/appendix/neighborhoodpop.htm")
#vinculo
```

- Se llenan los valores nulos, con el dato NO nulo anterior

```
# Llenar los valores nulos
df['Borough'] = df['Borough'].ffill() # Forward fill
```

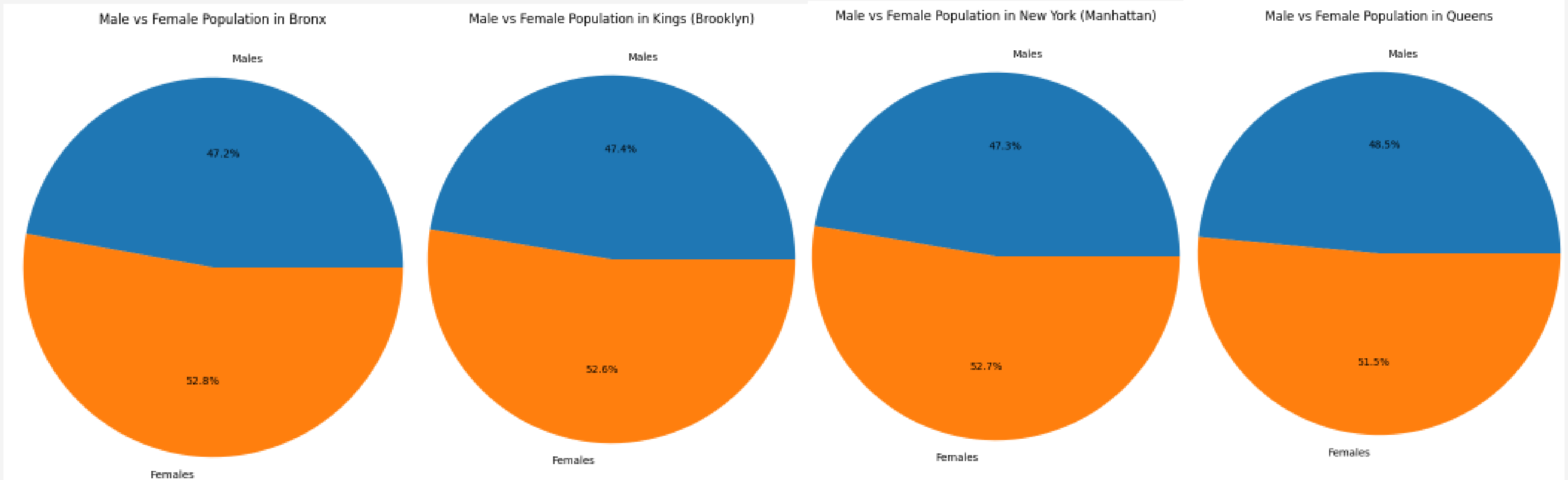
	Borough	region	Males	Females	Total Population
0	Bronx	Riverdale, Fieldston & Kingsbridge	52133	61937	114070
1	NaN	Wakefield, Williamsbridge & Woodlawn	65087	77848	142935
2	NaN	Co-op City, Pelham Bay & Schuylerville	55615	65929	121544
3	NaN	Pelham Parkway, Morris Park & Laconia	61233	67896	129130
4	NaN	Belmont, Crotona Park East & East Tremont	75963	87740	163704

	Borough	region	Males	Females	Total Population
0	Bronx	Riverdale, Fieldston & Kingsbridge	52133	61937	114070
1	Bronx	Wakefield, Williamsbridge & Woodlawn	65087	77848	142935
2	Bronx	Co-op City, Pelham Bay & Schuylerville	55615	65929	121544
3	Bronx	Pelham Parkway, Morris Park & Laconia	61233	67896	129130
4	Bronx	Belmont, Crotona Park East & East Tremont	75963	87740	163704

[Volver a la página de agenda](#)

Web Scrapping

El porcentaje de mujeres y hombres en cada localidad



[Volver a la página de agenda](#)

API

- Se llama api con longitud y latitud de nueva york.

```
import requests

def get_weather(lat, lon, api_key):
    url = f"http://api.openweathermap.org/data/2.5/weather?lat={lat}&lon={lon}&appid={api_key}" #Función que toma latitud, longitud y API Key
    response = requests.get(url)
    return response.json()

#En este caso se usan los datos de New York
latitude = '40.712776'
longitude = '-74.005974'

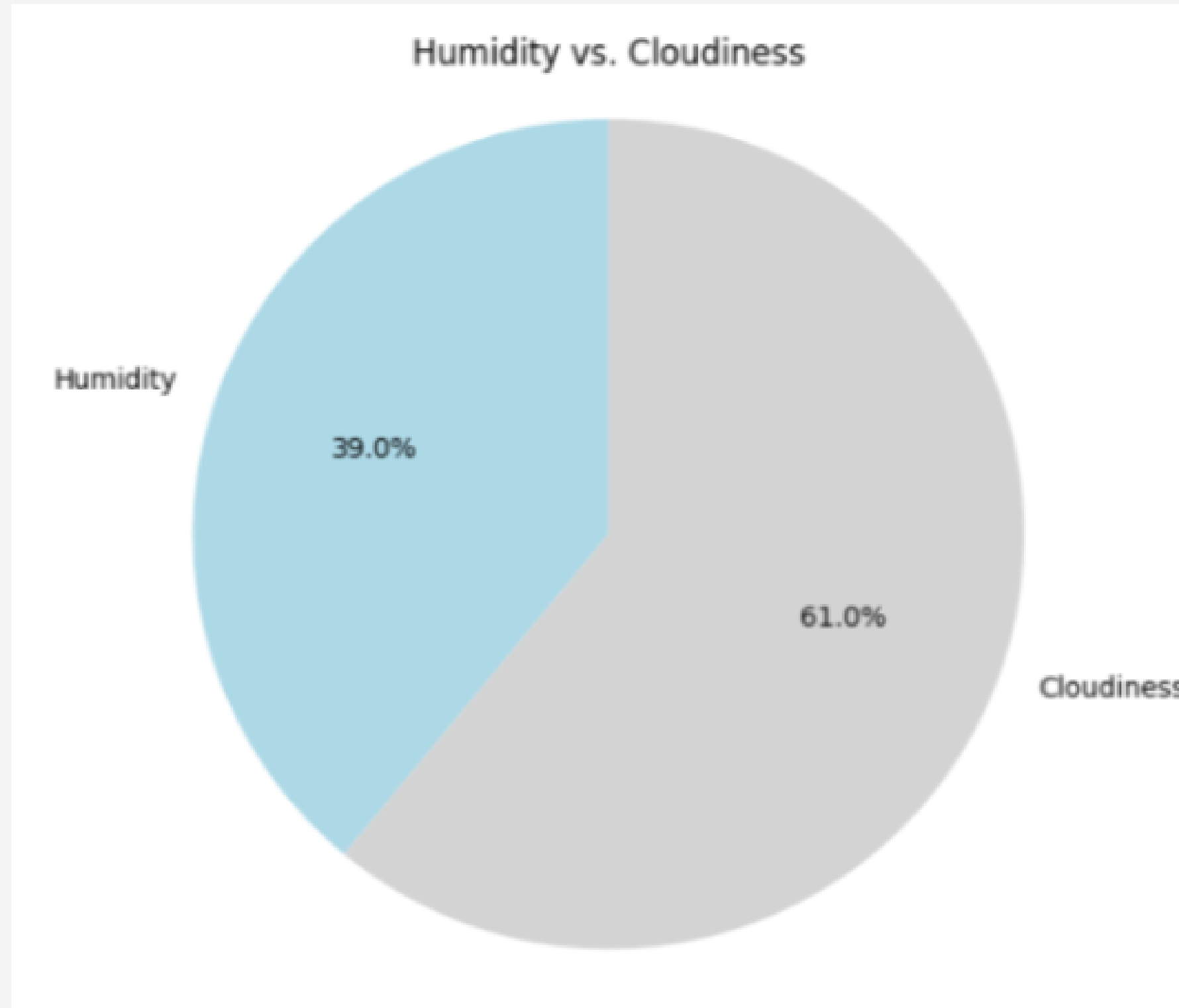
# llamar la función para obtener los datos
weather_data = get_weather(latitude, longitude, CONST_WEATHER_API_KEY)

# Verificar que el llamada sea exitoso, es decir 200
if weather_data.get('cod') == 200:
    print("Successfully retrieved weather data:")
    print(weather_data)
else:
    print("Failed to retrieve weather data. Error code:", weather_data.get('cod'))
```

[Volver a la página de agenda](#)

API

Visualización 01-04-24: Humedad vs Nubosidad



BIBLOGRAFIA

<https://kinsta.com/es/base-de-conocimiento/que-es-web-scraping/#:~:text=El%20web%20scraping%20se%20refiere,precios%20de%20varias%20tiendas%20online.>

<https://www.kyoceradocumentsolutions.es/es/smart-r-workspaces/business-challenges/the-cloud/procesamiento-datos-grandes-servidores.html>

<https://spark.apache.org/docs/3.3.1/api/python/index.html>

<https://catalog.data.gov/dataset/nypd-arrest-data-year-to-date>

The image features a light gray background with decorative geometric elements. In the top right corner, there are two overlapping hexagonal shapes: a dark teal one in front of a lighter teal one. In the bottom right corner, there is a single light green hexagonal shape.

GRACIAS