1) Data wrangling

August 14, 2023

1 Introducción

El siguiente código tiene la función de limpiar los datasets del sistema público de bicicletas de la Ciudad Autónoma de Buenos Aires para generar unos gráficos. Para analizar el código que produce los gráficos con el dataset limpio que aquí se generará o para leer el informe final con análisis y conclusiones de las visualizaciones, por favor recurra al respositorio GitHub del proyecto: https://github.com/Agustin-Bulzomi/Projects/tree/main/Programming/Gobierno%20Abierto%20(Python)

2 1) Preparación previa

2.0.1 Carga de librerías

[1]:
```
import pandas as pd
import numpy as np
from tqdm import tqdm
import unidecode
import requests
import io
```

2.0.2 Carga de datasets

[2]:
```
# Se almacena dentro de un diccionario cada url de los .csv que deben ser␣
 ↪analizados
url_dict = {
    2015: 'https://www.dropbox.com/scl/fi/q0wlnf3jw10tawc26zeyn/
 ↪recorridos-realizados-2015.csv?rlkey=rbi1z88t3o6w9fr8rzwdgqeu8&dl=1',
    2016: 'https://www.dropbox.com/scl/fi/9s05jse7q8mzzfzjjwn09/
 ↪recorridos-realizados-2016.csv?rlkey=r52xp7sab2bsiwe36gqx50oax&dl=1',
    2017: 'https://www.dropbox.com/scl/fi/b1tkyz57br13wv0vwsmcy/
 ↪recorridos-realizados-2017.csv?rlkey=buavz2lag6kd8y1cywe3702jg&dl=1',
    2018: 'https://www.dropbox.com/scl/fi/fvjrngoize8bbxtut9hyt/
 ↪recorridos-realizados-2018.csv?rlkey=ck7zprfimu8qfwjmz6goll685&dl=1',
    2019: 'https://www.dropbox.com/scl/fi/od2ntsows3939bb19epxn/
 ↪recorridos-realizados-2019.csv?rlkey=rwqm9p0qptgzuqtonlimlwukh&dl=1',
    2020: 'https://www.dropbox.com/scl/fi/iritm62p90zuy5v4uzz4b/
 ↪recorridos-realizados-2020.csv?rlkey=zx2kelus28oftelhb44fgqi7m&dl=1',
```

```
    2021: 'https://www.dropbox.com/scl/fi/vi2f56ju2bdpmeigac78j/
 ↪recorridos-realizados-2021.csv?rlkey=m4q2blrfk7az3jul1aj1d4v0a&dl=1',
  }
```

[3]:
```
# Se crea un progress bar para informar el proceso de descarga
pbar = tqdm(url_dict.keys())

for año in pbar:
    pbar.set_description(f"Procesando el año {año}. Datasets procesados")
    url = url_dict[año]
    # Se usa un condicional para agregar unas líneas necesarias para leer el␣
 ↪dataset 2021, que tiene algunas diferencias con respecto a los otros datasets
    if año == 2021:
        response = requests.get(url)
        f = response.content.replace(b'\x00', b'')
        df = pd.read_csv(io.BytesIO(f), encoding='ISO-8859-1', low_memory=False)
        # Se eliminan las columnas que no serán utilizadas
        df.drop(["ID", "Estado cerrado", "fecha_origen_recorrido",␣
 ↪"fecha_destino_recorrido", "Tipo de ciclista", "Apellido de ciclista",␣
 ↪"Msnbc de bicicleta", "Moto identificador público", "Código QR de␣
 ↪bicicleta", "Modelo de bicicleta", "ID de factura", "ID de línea de␣
 ↪factura", "Correo de ciclista", "Teléfono de ciclista", "ID de producto",␣
 ↪"Origen de viaje", "Nombre de producto"], axis = 1, inplace = True)
        # Se agrega el período, que no estaba en el dataset original
        df["año"] = 2021
    else:
        # Se lee el dataset
        df = pd.read_csv(url, low_memory = False).rename(columns={'periodo':␣
 ↪'año'})
        # Se eliminan las columnas que no serán utilizadas
        df.drop(["fecha_origen_recorrido", "fecha_destino_recorrido",␣
 ↪'domicilio_estacion_origen', 'domicilio_estacion_destino'], axis = 1,␣
 ↪inplace = True)
    globals()[f"dataset_{año}"] = df

print("\nFin del proceso\n\nLas variables de cada dataset se llaman:
 ↪\ndataset_2015\ndataset_2016\n...\n")
```

```
Procesando el año 2021. Datasets procesados: 100%|      | 7/7 [47:26<00:00,
406.69s/it]


Fin del proceso

Las variables de cada dataset se llaman:
dataset_2015
dataset_2016
…
```

3 2) Limpieza

3.0.1 Limpieza N° 1 - Género

Los datasets del 2020 y del 2021 no incluyen esta columna y el dataset del 2019 tiene mayoritariamente nulos, por lo que se obtendrán de distintas formas:

- 1) el dataset 2021 contiene el nombre de los ciclistas, por lo que se puede hacer un merge con una base de datos de nombres y géneros. Los faltantes se imputaran respetando las proporciones de género preexistentes
- 2) una vez completado el susodicho dataset, se aprovechará que los 3 datasets contienen una columna de id_usuario para encontrar en el 2020 y en el 2019 los usuarios cuyo género esté en el 2021. Los faltantes se imputarán de la misma manera que en el anterior caso.

2021:

A) Carga del df de nombres y géneros:

[4]:
```
nombres = pd.read_csv("https://raw.githubusercontent.com/Agustin-Bulzomi/
 ↪Projects/main/Programming/Gobierno%20Abierto%20(Python)/Datasets/nombres.
 ↪csv", delimiter = "\t", encoding= "latin")
nombres.columns = ["nombre", "genero_usuario"]
nombres
```

[4]:

	nombre	genero_usuario
0	aaron	M
1	aaronit	M
2	aba	F
3	abaco	M
4	abalen	M
…	…	…
9216	zulmara	F
9217	zunilda	F
9218	zuza	F
9219	zuzanny	F
9220	zysli	F

[9221 rows x 2 columns]

B) Limpieza de la columna “nombre” en el dataset para aplicar merge:

[5]:
```
dataset_2021["nombre"] = dataset_2021["nombre"].str.lower()

# Se divide el string para separar casos con primer y segundo nombre:
dataset_2021["nombre"] = dataset_2021["nombre"].str.split()
```

```
# Se obtiene solo el primer nombre de la lista generada en la anterior línea de␣
 ↪código:
dataset_2021.loc[:,"nombre"] = dataset_2021["nombre"].map(lambda x: x[0])
# Se limpia el formato de caracteres especiales.
# Este proceso se realizará en todos los nombres pues, como no serán expuestos,␣
 ↪no pesa tanto que una estación se llame "peron" en vez de "Perón" o "pena"␣
 ↪en vez de "Peña".
# En cambio, sí es un riesgo la posibilidad de que se dupliquen estaciones por␣
 ↪diferencias en formato ("Peron" vs "Perón", etc.):
dataset_2021["nombre"] = dataset_2021["nombre"].apply(lambda x : unidecode.
 ↪unidecode(x))
# Se aplica el merge:
dataset_2021 = dataset_2021.merge(nombres, on ='nombre', how ='left')

dataset_2021
```

[5]:

	duracion_recorrido	id_estacion_origen	nombre_estacion_origen	\
0	447	323	240 - ECHEVERRIA	
1	438	167	275 - PLAZA 24 DE SEPTIEMBRE	
2	414	247	282 - Tronador	
3	38	158	158 - VILLARROEL	
4	484	29	029 - Parque Centenario	
…	…	…	…	
1144217	690	277	292 - PLAZA BOLIVIA	
1144218	1360	79	079 - AZUCENA VILLAFLOR	
1144219	1169	79	079 - AZUCENA VILLAFLOR	
1144220	1031	79	079 - AZUCENA VILLAFLOR	
1144221	1795	79	079 - AZUCENA VILLAFLOR	

	id_estacion_destino	nombre_estacion_destino	id_usuario	\
0	289.0	255 - BARRANCAS DE BELGRANO	701665	
1	273.0	223 - GAINZA	752374	
2	400.0	313 - De Los Incas	425502	
3	158.0	158 - VILLARROEL	4519	
4	99.0	099 - Malabia	8197	
…	…	…	…	
1144217	44.0	044 - Ecoparque	62246	
1144218	168.0	168 - Estados Unidos	445201	
1144219	8.0	008 - Congreso	554162	
1144220	75.0	075 - Plaza Primero de Mayo	51005	
1144221	207.0	123 - Armenia	734428	

	nombre	año	genero_usuario
0	andres	2021	M
1	maria	2021	F
2	romina	2021	F
3	javier	2021	M

```
4          victor  2021              M
…             …    …             …
1144217   valeria  2021              F
1144218     fazal  2021            NaN
1144219     amjad  2021            NaN
1144220   delfina  2021              F
1144221     shady  2021            NaN

[1144222 rows x 9 columns]
```

C) Aplicación del merge para imputar nulos:

[6]:
```
# Cálculo de resultados:
print("\nQuedaron:", dataset_2021.genero_usuario.isnull().sum(), "nulos␣
 ↪restantes\nLa imputación anterior completó correctamente el",␣
 ↪round(((len(dataset_2021)-dataset_2021.genero_usuario.isnull().sum())/
 ↪len(dataset_2021)), 2)*100, "% de los datos faltantes\n")
```

```
Quedaron: 68233 nulos restantes
La imputación anterior completó correctamente el 94.0 % de los datos faltantes
```

D) Imputación de los últimos nulos:

[7]:
```
# Cálculo de las proporciones a utilizar
value_counts_2021 = round(dataset_2021.genero_usuario.value_counts(normalize =␣
 ↪True),3)
value_counts_2021
```

[7]:
```
M    0.656
F    0.344
Name: genero_usuario, dtype: float64
```

[8]:
```
# Imputación:
dataset_2021.loc[dataset_2021["genero_usuario"].isnull(), "genero_usuario"] =␣
 ↪np.random.choice(size=dataset_2021["genero_usuario"].isnull().sum(), a=["M",␣
 ↪"F"], p=[value_counts_2021[0], value_counts_2021[1]])
# Se chequea que las proporciones se hayan mantenido similares:
round(dataset_2021.genero_usuario.value_counts(normalize = True),3)
```

[8]:
```
M    0.656
F    0.344
Name: genero_usuario, dtype: float64
```

[9]:
```
# Se corrobora que haya funcionado:
print("\nCantidad de datos nulos:", dataset_2021.genero_usuario.isnull().sum(),␣
 ↪"\n")
```

```
Cantidad de datos nulos: 0
```

[10]:
```
dataset_2021.isnull().sum()
```

[10]:
```
duracion_recorrido         0
id_estacion_origen         0
nombre_estacion_origen     0
id_estacion_destino        2
nombre_estacion_destino    2
id_usuario                 0
nombre                     0
año                        0
genero_usuario             0
dtype: int64
```

[11]:
```
# Se puede eliminar la columna "nombre" que ya no será utilizada
dataset_2021.drop(["nombre"], axis = 1, inplace = True)
```

2020:

A) Creación de dataframe con único id_usuario del 2021:

Este paso es necesario pues muchos usuarios hicieron más de un recorrido. Al aparecer en más de una fila, un merge los duplicaría en el 2020

[12]:
```
# Se eliminan los recorridos con id_usuario duplicado:
dataframe_id_genero = dataset_2021.loc[:, ['id_usuario','genero_usuario']].
 ↪drop_duplicates(subset = "id_usuario")
# Se transforma el dtype a integer para el merge:
id_int = dataframe_id_genero.id_usuario.astype("int64")
# Se reemplaza la serie object por la nueva serie integer:
dataframe_id_genero.id_usuario = id_int
# Se filtra solo los id_usuarios que se necesitan para 2020, pues un merge con␣
 ↪usuarios de 2021 que no estén en 2020 agregaría filas:
dataframe_id_genero_2020 = dataframe_id_genero.loc[dataframe_id_genero.
 ↪id_usuario.isin(dataset_2020.id_usuario), :]
dataframe_id_genero_2020
```

[12]:

	id_usuario	genero_usuario
0	701665	M
2	425502	F
3	4519	M
4	8197	M
5	162618	F
…	…	…
1144137	1649	F

```
1144145       51780              M
1144147      678795              F
1144177      447308              F
1144179      678907              F

[66543 rows x 2 columns]
```

B) Aplicación del merge entre el nuevo dataframe y el dataset:

[13]:
```
dataset_2020 = pd.merge(dataset_2020, dataframe_id_genero_2020, how = "outer")
# Cálculo de resultados:
print("\nQuedaron:", dataset_2020.genero_usuario.isnull().sum(), "nulos␣
 ↪restantes\nLa imputación anterior completó correctamente el",␣
 ↪round(((len(dataset_2020)-dataset_2020.genero_usuario.isnull().sum())/
 ↪len(dataset_2020)), 2)*100, "% de los datos faltantes\n")
```

```
Quedaron: 665173 nulos restantes
La imputación anterior completó correctamente el 67.0 % de los datos faltantes
```

C) Imputación de los últimos nulos:

[14]:
```
# Cálculo de las proporciones a utilizar
value_counts_2020 = round(dataset_2020["genero_usuario"].value_counts(normalize␣
 ↪= True),3)
value_counts_2020
```

[14]:
```
M    0.665
F    0.335
Name: genero_usuario, dtype: float64
```

[15]:
```
# Imputación:
dataset_2020.loc[dataset_2020["genero_usuario"].isnull(), "genero_usuario"] =␣
 ↪np.random.choice(size=dataset_2020["genero_usuario"].isnull().sum(), a=["M",␣
 ↪"F"], p=[value_counts_2020[0], value_counts_2020[1]])
# Se chequea que las proporciones se hayan mantenido similares:
round(dataset_2020.genero_usuario.value_counts(normalize = True),3)
```

[15]:
```
M    0.665
F    0.335
Name: genero_usuario, dtype: float64
```

[16]:
```
# Se corrobora que haya funcionado:
print("\nCantidad de datos nulos:", dataset_2020.genero_usuario.isnull().sum(),␣
 ↪"\n")
```

```
Cantidad de datos nulos: 0
```

2019:

[17]:
```
dataset_2019.genero_usuario.value_counts()
```

[17]:
```
M               149804
F                54268
NO INFORMADO        35
-58.4429517          4
Name: genero_usuario, dtype: int64
```

A) Cambios en las categorías "NO INFORMADO" y en los valores erroneos:

Las 4 filas erroneas contienen nulos en las demás columnas, por lo que no se imputarán sino que se eliminarán:

[18]:
```
index_numeros = dataset_2019[dataset_2019["genero_usuario"] == "-58.4429517"].
 ↪index
dataset_2019.drop(index_numeros, inplace = True)
```

Se reemplaza el valor "No informado" por la letra "N" utilizada en otro dataset:

[19]:
```
dataset_2019.loc[:,"genero_usuario"].replace({'NO INFORMADO':'N'}, inplace =␣
 ↪True)

dataset_2019.genero_usuario.value_counts()
```

[19]:
```
M    149804
F     54268
N        35
Name: genero_usuario, dtype: int64
```

B) Cambios en columnas para hacer merge:

[20]:
```
print("Ya hay un", round(((len(dataset_2019)-dataset_2019.genero_usuario.
 ↪isnull().sum())/len(dataset_2019)), 2)*100, "% de los datos válidos")
```

```
Ya hay un 3.0 % de los datos válidos
```

Como en este dataset ya existe la columna género con algunos datos, hay que renombrarla y así evitar que se dupliquen filas

[21]:
```
dataset_2019.rename(columns={"genero_usuario": "genero_usuario_viejo"}, inplace␣
 ↪= True)
```

[22]:
```
dataframe_id_genero_2019 = dataframe_id_genero.loc[dataframe_id_genero.
 ↪id_usuario.isin(dataset_2019.id_usuario), :]
dataframe_id_genero_2019
```

[22]:
```
         id_usuario genero_usuario
2            425502              F
3              4519              M
4              8197              M
5            162618              F
8            361721              M
…               …              …
1144044      293077              F
1144091      319196              F
1144137        1649              F
1144145       51780              M
1144177      447308              F

[47562 rows x 2 columns]
```

B) Aplicación del merge entre el nuevo dataframe y el dataset:

[23]:
```
dataset_2019 = pd.merge(dataset_2019, dataframe_id_genero_2019, how = "outer")
# Cálculo de resultados:
print("\nQuedaron:", dataset_2019.genero_usuario.isnull().sum(), "nulos␣
 ↪restantes\nLa imputación anterior completó correctamente otro",
      (round(((len(dataset_2019)-dataset_2019.genero_usuario.isnull().sum())/
 ↪len(dataset_2019)), 2)*100), "% de los datos faltantes")
```

```
Quedaron: 5100675 nulos restantes
La imputación anterior completó correctamente otro 20.0 % de los datos faltantes
```

C) Unificación de ambas imputaciones

[24]:
```
dataset_2019["genero_usuario"] = dataset_2019["genero_usuario"].
 ↪combine_first(dataset_2019["genero_usuario_viejo"])
```

[25]:
```
dataset_2019.drop("genero_usuario_viejo", axis = 1, inplace = True)
dataset_2019
```

[25]:
```
           año  id_usuario id_estacion_origen nombre_estacion_origen  \
0         2019      115783                168         Estados Unidos
1         2019      115783                141         Solís y Alsina
2         2019      115783                 76                Ayacucho
3         2019      115783                148        Constitución II
4         2019      115783                 88                Misiones
…          …          …                   …                      …
6367305   2019      177162                283        228 - BALDOMERO
6367306   2019      585275                283        228 - BALDOMERO
6367307   2019      494906                283        228 - BALDOMERO
6367308   2019      618404                283        228 - BALDOMERO
6367309   2019      606607                283        228 - BALDOMERO
```

```
         long_estacion_origen  lat_estacion_origen         duracion_recorrido  \
0                   -58.381283           -34.618620  0 days 00:07:02.000000000
1                   -58.390411           -34.611838  0 days 00:10:33.000000000
2                   -58.394927           -34.607573  0 days 00:14:16.000000000
3                   -58.379232           -34.627494  0 days 00:50:08.000000000
4                   -58.404230           -34.612867  0 days 00:27:05.000000000
…                          …                    …                         …
6367305             -58.456035           -34.635505  0 days 00:26:52.000000000
6367306             -58.456035           -34.635505  0 days 00:14:46.000000000
6367307             -58.456035           -34.635505  0 days 00:41:50.000000000
6367308             -58.456035           -34.635505  0 days 00:43:58.000000000
6367309             -58.456035           -34.635505  0 days 00:37:21.000000000

         id_estacion_destino nombre_estacion_destino  long_estacion_destino  \
0                      112.0              9 de Julio             -58.380766
1                       93.0            Carlos Calvo             -58.394769
2                      171.0                 Pasteur             -58.399527
3                       76.0                Ayacucho             -58.394927
4                       18.0          Independencia              -58.380481
…                          …                     …                      …
6367305                283.0         228 - BALDOMERO             -58.456035
6367306                283.0         228 - BALDOMERO             -58.456035
6367307                283.0         228 - BALDOMERO             -58.456035
6367308                283.0         228 - BALDOMERO             -58.456035
6367309                283.0         228 - BALDOMERO             -58.456035

         lat_estacion_destino genero_usuario
0                   -34.612389              F
1                   -34.620601              F
2                   -34.603243              F
3                   -34.607573              F
4                   -34.617373              F
…                          …            …
6367305             -34.635505            NaN
6367306             -34.635505            NaN
6367307             -34.635505            NaN
6367308             -34.635505            NaN
6367309             -34.635505            NaN

[6367310 rows x 12 columns]
```

[26]:
```
# Cálculo de resultados:
proporcion_validos = round((len(dataset_2019)-dataset_2019.genero_usuario.
 ↪isnull().sum())/len(dataset_2019), 3)
print("\nQuedaron:", dataset_2019.genero_usuario.isnull().sum(), "nulos␣
 ↪restantes al unir ambos resultados, es decir, un",
      proporcion_validos*100, "% de los datos son válidos\n")
```

```
Quedaron: 4915794 nulos restantes al unir ambos resultados, es decir, un 22.8 %
de los datos son válidos
```

D) Imputación de los valores restantes

Utilizar un 23% de los datos para imputar el 77% restante no es estadísticamente robusto, por lo que se utilizará un punto medio entre las proporciones del 23% válido de 2019 y las proporciones obtenidas al promediar 2018 y 2020:

[27]:
```
value_counts_2019 = round(dataset_2019.genero_usuario.value_counts(normalize =
↪True),3)
value_counts_2019
```

[27]:
```
M    0.741
F    0.259
N    0.000
Name: genero_usuario, dtype: float64
```

[28]:
```
promedio_2018_2020 = round((dataset_2018.genero_usuario.value_counts(normalize
↪= True) + dataset_2020.genero_usuario.value_counts(normalize = True)) /2, 3)
promedio_2018_2020
```

[28]:
```
F    0.309
M    0.691
N      NaN
Name: genero_usuario, dtype: float64
```

El valor N da NaN por no estar presente en el dataset 2018. Se utilizarán las otras dos proporciones para imputar:

[29]:
```
value_counts_promedio = round((dataset_2019.genero_usuario.
↪value_counts(normalize = True) + promedio_2018_2020) /2, 3)
value_counts_promedio
```

[29]:
```
F    0.284
M    0.716
N      NaN
Name: genero_usuario, dtype: float64
```

[30]:
```
# Imputación:
dataset_2019.loc[dataset_2019["genero_usuario"].isnull(), "genero_usuario"] =
↪np.random.choice(size=dataset_2019["genero_usuario"].isnull().sum(), a=["M",
↪"F"], p=[value_counts_promedio[1], value_counts_promedio[0]])
# Se chequea que las proporciones se hayan mantenido similares:
print("Las proporciones deberían ser:\n")
```

```
print(round(value_counts_promedio * (1-proporcion_validos) + ((dataset_2019.
 ↪genero_usuario.value_counts(normalize = True) + promedio_2018_2020) /2) *␣
 ↪proporcion_validos, 3))
print("\n\ny dieron:\n")
print(round(dataset_2019["genero_usuario"].value_counts(normalize = True),3))
```

```
Las proporciones deberían ser:

F    0.286
M    0.714
N      NaN
Name: genero_usuario, dtype: float64


y dieron:

M    0.722
F    0.278
N    0.000
Name: genero_usuario, dtype: float64
```

[31]:
```
# Se corrobora que haya funcionado:
print("\nCantidad de datos nulos:", dataset_2019.genero_usuario.isnull().sum(),␣
 ↪"\n")
```

```
Cantidad de datos nulos: 0
```

3.0.2 Limpieza N° 2 - Duración

2015 - 2019:

[32]:
```
dataset_2015_2019 = pd.concat([dataset_2015, dataset_2016, dataset_2017,␣
 ↪dataset_2018, dataset_2019])
```

[33]:
```
dataset_2015_2019 = dataset_2015_2019[dataset_2015_2019.duracion_recorrido.
 ↪notnull()]
```

[34]:
```
duracion_recorrido = dataset_2015_2019.loc[:,"duracion_recorrido"]
duracion_split_2015_2019 = duracion_recorrido.str.split(":", expand = True)[1]
dataset_2015_2019 = pd.concat((dataset_2015_2019, duracion_split_2015_2019),␣
 ↪axis = 1).rename(columns= {1 : "minutos"})
dataset_2015_2019.drop(["duracion_recorrido"], axis = 1, inplace = True)
```

[35]:
```
sum(dataset_2015_2019["minutos"].isnull())
```

[35]: 0

[36]:
```
pbar = tqdm([2015, 2016, 2017, 2018, 2019])

for año in pbar:
    temp_dataset = dataset_2015_2019[dataset_2015_2019.año == año]
    globals()[f"dataset_{año}"] = temp_dataset
```

```
100%|      | 5/5 [00:01<00:00,  3.04it/s]
```

2020 - 2021:

[37]:
```
dataset_2020_2021 = pd.concat([dataset_2020, dataset_2021])
```

[38]:
```
dataset_2020_2021["duracion_recorrido"] =␣
 ↪round(dataset_2020_2021["duracion_recorrido"].astype("int") / 60).
 ↪astype("int")
dataset_2020_2021.rename(columns= {"duracion_recorrido": "minutos"}, inplace =␣
 ↪True)
```

[39]:
```
sum(dataset_2020_2021.minutos.isnull())
```

[39]:
```
0
```

[40]:
```
pbar = tqdm([2020, 2021])

for año in pbar:
    temp_dataset = dataset_2020_2021[dataset_2020_2021.año == año]
    globals()[f"dataset_{año}"] = temp_dataset
```

```
100%|      | 2/2 [00:00<00:00,  3.57it/s]
```

3.0.3 Limpieza N° 3 - Estación de origen y destino

Se limpia tanto el id, el nombre, la latitud y la longitud de la estación pues son datos relacionados.

Utilizando los datasets de estaciones que provee el Gobierno de la Ciudad, disponibles en https://data.buenosaires.gob.ar/dataset/estaciones-bicicletas-publicas, se observa que la discrepancia en id de los datasets de bicicletas surge por la presencia de dos números que identifican estaciones: id y código.

Lamentablemente, desde el nuevo sistema (2019 en adelante) el identificador de estación cambió, por lo que el dato "id_estacion" del primer sistema es el código que aparece a veces al principio de "nombre_estacion". Es por eso que se lo tendrá que extraer para reemplazar el id_estacion. En los casos en los que el nombre no incluya el código, deberá ser buscado en otras filas o, de no existir, en la base de datos de estaciones.

[41]:
```
estaciones_nuevo = pd.read_csv("https://cdn.buenosaires.gob.ar/datosabiertos/
 ↪datasets/transporte/estaciones-bicicletas-publicas/
 ↪nuevas-estaciones-bicicletas-publicas.csv", sep=",", encoding='utf-8')
estaciones_viejo = pd.read_csv("https://cdn.buenosaires.gob.ar/datosabiertos/
 ↪datasets/estaciones-bicicletas-publicas/estaciones_sistema_viejo.csv",␣
 ↪sep=",", encoding='utf-8')
```

[42]:
```
estaciones_nuevo
```

[42]:

	WKT	id	codigo	nombre	ubicacion	tipo	horario	anclajes_t
0	POINT (-58.3747109506359 -34.5924239181221)	2	2	002 - Retiro I	Ramos Mejia, Jose Maria, Dr. Av. & Del Liberta…	AUTOMÁTICA	Estación automática: disponibilidad las 24 horas	20
1	POINT (-58.368256111128 -34.611033074021)	3	3	003 - ADUANA	Moreno & Av Paseo Colon	AUTOMÁTICA	Estación automática: disponibilidad las 24 horas	20
2	POINT (-58.3687766674259 -34.6018228613782)	4	4	004 - Plaza Roma	Lavalle & Bouchard	AUTOMÁTICA	Estación automática: disponibilidad las 24 horas	20
3	POINT (-58.420951914897 -34.5805498216605)	5	5	005 - Plaza Italia	Av. Sarmiento 2601	AUTOMÁTICA	Estación automática: disponibilidad las 24 horas	42
4	POINT (-58.3697538990917 -34.6285274659984)	6	6	006 - Parque Lezama	Avenida Martin Garcia, 295	AUTOMÁTICA	Estación automática: disponibilidad las 24 horas	20
..	…	…	…	…	…	…	…	…
224	POINT (-58.4594980806633 -34.5445021121101)	396	227	227 -Club Ciudad de Buenos Aires	Miguel Sanchez y Av Libertadores	AUTOMÁTICA		
225	POINT (-58.3747959781335 -34.6098020140517)	398	16	016 - Legislatura	169 Peru & Roca, Julio A., Presidente Diagonal…	AUTOMÁTICA		
226	POINT (-58.370711799739 -34.6089370142298)	401	61	061-Ministerio de Economia	Balcarce & Yrigoyen, Hipolito Av.	AUTOMÁTICA		
227	POINT (-58.3788579541696 -34.5824223699167)	403	393	393 - Barrio 31	Carlos H. Perette 11	AUTOMÁTICA		
228	POINT (-58.4632508461667 -34.6290840208419)	405	352	352 - San Jose de Flores	Avenida Rivadavia y Fray Cayetano	AUTOMÁTICA		

```
..                                                  …          …
224  Estación automática: disponibilidad las 24 horas          16
225  Estación automática: disponibilidad las 24 horas          20
226  Estación automática: disponibilidad las 24 horas          24
227  Estación automática: disponibilidad las 24 horas          24
228  Estación automática: disponibilidad las 24 horas          24

[229 rows x 8 columns]
```

[43]:
```
estaciones_viejo
```

[43]:
```
     id_estacion          nombre_estacion  long_estacion  lat_estacion  \
0            1.0      Facultad de Derecho     -58.392452    -34.583133
1            2.0                   Retiro     -58.374822    -34.592589
2            3.0                   Aduana     -58.368918    -34.611242
3            4.0              Plaza Roma     -58.368950    -34.601721
4            5.0             Plaza Italia     -58.420997    -34.580127
..           …                        …           …           …
199        200.0         Austria y French     -58.404361    -34.588191
200          NaN                      CMD            NaN           NaN
201          NaN  F. J. Santamaría de Oro            NaN           NaN
202          NaN       Fitz Roy y Gorriti            NaN           NaN
203          NaN                Ecoparque            NaN           NaN

                             domicilio_estacion tipo_estacion  \
0    Av. Pres.Figueroa Alcorta y  Juan A.Bibiloni    AUTOMÁTICA
1     Av. Dr.Jose Ramos Mejia y Del Libertador Av    AUTOMÁTICA
2                  Av. Ing.Huergo y Av. Belgrano    AUTOMÁTICA
3                           Lavalle y Bouchard    AUTOMÁTICA
4                   Av. Santa Fe y Av. Sarmiento    AUTOMÁTICA
..                                           …           …
199           Austria 2080 entre French y Juncal    AUTOMÁTICA
200                                          NaN           NaN
201                                          NaN           NaN
202                                          NaN           NaN
203                                          NaN           NaN

                                  observaciones  \
0    Abril 2015 (pasó de ser Manual a Automática)
1    Abril 2015 (pasó de ser Manual a Automática)
2    Abril 2015 (pasó de ser Manual a Automática)
3    Abril 2015 (pasó de ser Manual a Automática)
4    Abril 2015 (pasó de ser Manual a Automática)
..                                           …
199                          Septiembre 2017
200                                      NaN
201                                      NaN
```

```
202                                                NaN
203                                                NaN

                                       horario_estacion
0    Estación automática: disponibilidad las 24 horas
1    Estación automática: disponibilidad las 24 horas
2    Estación automática: disponibilidad las 24 horas
3    Estación automática: disponibilidad las 24 horas
4    Estación automática: disponibilidad las 24 horas
..                                                 …
199  Estación automática: disponibilidad las 24 horas
200                                                NaN
201                                                NaN
202                                                NaN
203                                                NaN

[204 rows x 8 columns]
```

2021:

[44]:
```
dataset_2021
```

[44]:
```
         minutos  id_estacion_origen        nombre_estacion_origen  \
0              7                 323                240 - ECHEVERRIA
1              7                 167  275 - PLAZA 24 DE SEPTIEMBRE
2              7                 247                 282 - Tronador
3              1                 158                158 - VILLARROEL
4              8                  29       029 - Parque Centenario
…            …                   …                             …
1144217       12                 277           292 - PLAZA BOLIVIA
1144218       23                  79       079 - AZUCENA VILLAFLOR
1144219       19                  79       079 - AZUCENA VILLAFLOR
1144220       17                  79       079 - AZUCENA VILLAFLOR
1144221       30                  79       079 - AZUCENA VILLAFLOR

         id_estacion_destino      nombre_estacion_destino  id_usuario  \
0                      289.0  255 - BARRANCAS DE BELGRANO      701665
1                      273.0                223 - GAINZA      752374
2                      400.0          313 - De Los Incas      425502
3                      158.0             158 - VILLARROEL        4519
4                       99.0                099 - Malabia        8197
…                          …                            …         …
1144217                 44.0              044 - Ecoparque       62246
1144218                168.0         168 - Estados Unidos      445201
1144219                  8.0                008 - Congreso      554162
1144220                 75.0  075 - Plaza Primero de Mayo       51005
1144221                207.0                123 - Armenia      734428
```

```
         long_estacion_origen  lat_estacion_origen  long_estacion_destino  \
0                         NaN                  NaN                    NaN
1                         NaN                  NaN                    NaN
2                         NaN                  NaN                    NaN
3                         NaN                  NaN                    NaN
4                         NaN                  NaN                    NaN
…                         …                    …                      …
1144217                   NaN                  NaN                    NaN
1144218                   NaN                  NaN                    NaN
1144219                   NaN                  NaN                    NaN
1144220                   NaN                  NaN                    NaN
1144221                   NaN                  NaN                    NaN

         lat_estacion_destino   año genero_usuario
0                         NaN  2021              M
1                         NaN  2021              F
2                         NaN  2021              F
3                         NaN  2021              M
4                         NaN  2021              M
…                         …    …             …
1144217                   NaN  2021              F
1144218                   NaN  2021              M
1144219                   NaN  2021              F
1144220                   NaN  2021              F
1144221                   NaN  2021              M

[1144222 rows x 12 columns]
```

[45]:
```
%%capture [--no-stderr]
dataset_2021.loc[dataset_2021.nombre_estacion_origen.isnull() == False,␣
 ↪"nombre_estacion_origen"] = dataset_2021.loc[dataset_2021.
 ↪nombre_estacion_origen.isnull() == False, "nombre_estacion_origen"].
 ↪apply(lambda x : unidecode.unidecode(x))
dataset_2021.loc[dataset_2021.nombre_estacion_origen.isnull() == False,␣
 ↪"nombre_estacion_origen"] = dataset_2021.loc[dataset_2021.
 ↪nombre_estacion_origen.isnull() == False, "nombre_estacion_origen"].str.
 ↪lower()

dataset_2021.loc[dataset_2021.nombre_estacion_destino.isnull() == False,␣
 ↪"nombre_estacion_destino"] = dataset_2021.loc[dataset_2021.
 ↪nombre_estacion_destino.isnull() == False, "nombre_estacion_destino"].
 ↪apply(lambda x : unidecode.unidecode(x))
dataset_2021.loc[dataset_2021.nombre_estacion_destino.isnull() == False,␣
 ↪"nombre_estacion_destino"] = dataset_2021.loc[dataset_2021.
 ↪nombre_estacion_destino.isnull() == False, "nombre_estacion_destino"].str.
 ↪lower()
```

[46]:
```
dataset_2021.isnull().sum()
```

[46]:
```
minutos                        0
id_estacion_origen             0
nombre_estacion_origen         0
id_estacion_destino            2
nombre_estacion_destino        2
id_usuario                     0
long_estacion_origen     1144222
lat_estacion_origen      1144222
long_estacion_destino    1144222
lat_estacion_destino     1144222
año                            0
genero_usuario                 0
dtype: int64
```

[47]:
```
dataset_2021[dataset_2021.id_estacion_destino.isnull()]
```

[47]:
```
        minutos  id_estacion_origen   nombre_estacion_origen  \
209519        0                 359              250 - fleni
250815        0                 432  187 - jose maria moreno

        id_estacion_destino nombre_estacion_destino  id_usuario  \
209519                  NaN                    NaN      753996
250815                  NaN                    NaN      141425

        long_estacion_origen  lat_estacion_origen  long_estacion_destino  \
209519                   NaN                  NaN                    NaN
250815                   NaN                  NaN                    NaN

        lat_estacion_destino   año genero_usuario
209519                   NaN  2021              F
250815                   NaN  2021              M
```

[48]:
```
# Debido a discrepancias entre id y nombre, sumado a la falta de información␣
 ↪sobre latitud y longitud, se borrarán estas filas:
dataset_2021 = dataset_2021[~pd.isnull(dataset_2021.id_estacion_destino)]
```

[49]:
```
# Se eliminan unas filas con información dudosa (estación inexistente en la␣
 ↪base de datos de estaciones y con un código alto)
dataset_2021 = dataset_2021.loc[dataset_2021.nombre_estacion_destino != "balboa␣
 ↪definitivo",:]
dataset_2021
```

[49]:
```
         minutos  id_estacion_origen        nombre_estacion_origen  \
0              7                 323             240 - echeverria
1              7                 167  275 - plaza 24 de septiembre
```

```
2              7                 247                282 - tronador
3              1                 158              158 - villarroel
4              8                  29       029 - parque centenario
…            …                  …                          …
1144217       12                 277           292 - plaza bolivia
1144218       23                  79       079 - azucena villaflor
1144219       19                  79       079 - azucena villaflor
1144220       17                  79       079 - azucena villaflor
1144221       30                  79       079 - azucena villaflor

         id_estacion_destino      nombre_estacion_destino  id_usuario  \
0                      289.0  255 - barrancas de belgrano      701665
1                      273.0                223 - gainza      752374
2                      400.0           313 - de los incas      425502
3                      158.0             158 - villarroel        4519
4                       99.0                099 - malabia        8197
…                       …                          …          …
1144217                44.0              044 - ecoparque       62246
1144218               168.0         168 - estados unidos      445201
1144219                 8.0                008 - congreso      554162
1144220                75.0  075 - plaza primero de mayo       51005
1144221               207.0                123 - armenia      734428

         long_estacion_origen  lat_estacion_origen  long_estacion_destino  \
0                         NaN                  NaN                    NaN
1                         NaN                  NaN                    NaN
2                         NaN                  NaN                    NaN
3                         NaN                  NaN                    NaN
4                         NaN                  NaN                    NaN
…                       …                    …                      …
1144217                   NaN                  NaN                    NaN
1144218                   NaN                  NaN                    NaN
1144219                   NaN                  NaN                    NaN
1144220                   NaN                  NaN                    NaN
1144221                   NaN                  NaN                    NaN

         lat_estacion_destino   año genero_usuario
0                         NaN  2021              M
1                         NaN  2021              F
2                         NaN  2021              F
3                         NaN  2021              M
4                         NaN  2021              M
…                       …    …                 …
1144217                   NaN  2021              F
1144218                   NaN  2021              M
1144219                   NaN  2021              F
1144220                   NaN  2021              F
```

```
1144221                    NaN  2021              M

[1144185 rows x 12 columns]
```

[50]:
```
# Se separa el código del nombre
# Origen:
col_nombre_origen = dataset_2021.loc[:,"nombre_estacion_origen"]
nombre_origen_2021 = col_nombre_origen.str.split("-", expand = True)[1].str.
 ↪strip()
id_origen_2021 = col_nombre_origen.str.split("-", expand = True)[0].
 ↪astype("int")
dataset_2021 = pd.concat((dataset_2021, id_origen_2021, nombre_origen_2021),␣
 ↪axis = 1).rename(columns= {1 : "nombre_origen", 0 : "codigo_origen"})

# Destino:
col_nombre_destino = dataset_2021.loc[:,"nombre_estacion_destino"].str.strip()
nombre_destino_2021 = col_nombre_destino.str.split("-", expand = True)[1]
id_destino_2021 = col_nombre_destino.str.split("-", expand = True)[0].
 ↪astype("int")
dataset_2021 = pd.concat((dataset_2021, id_destino_2021, nombre_destino_2021),␣
 ↪axis = 1).rename(columns= {1 : "nombre_destino", 0 : "codigo_destino"})

# Drop de columnas viejas:
dataset_2021.drop(["nombre_estacion_origen", "id_estacion_origen",␣
 ↪"id_estacion_destino", "nombre_estacion_destino"], axis = 1, inplace = True)
dataset_2021
```

[50]:

	minutos	id_usuario	long_estacion_origen	lat_estacion_origen	long_estacion_destino	lat_estacion_destino	año	genero_usuario
0	7	701665	NaN	NaN	NaN	NaN	2021	M
1	7	752374	NaN	NaN	NaN	NaN	2021	F
2	7	425502	NaN	NaN	NaN	NaN	2021	F
3	1	4519	NaN	NaN	NaN	NaN	2021	M
4	8	8197	NaN	NaN	NaN	NaN	2021	M
…	…	…	…	…	…	…	…	…
1144217	12	62246	NaN	NaN				
1144218	23	445201	NaN	NaN				
1144219	19	554162	NaN	NaN				
1144220	17	51005	NaN	NaN				
1144221	30	734428	NaN	NaN				

```
1144217                   NaN                   NaN  2021              F
1144218                   NaN                   NaN  2021              M
1144219                   NaN                   NaN  2021              F
1144220                   NaN                   NaN  2021              F
1144221                   NaN                   NaN  2021              M

         codigo_origen           nombre_origen  codigo_destino  \
0                  240              echeverria             255
1                  275  plaza 24 de septiembre             223
2                  282                tronador             313
3                  158              villarroel             158
4                   29       parque centenario              99
…                  …                       …              …
1144217            292           plaza bolivia              44
1144218             79       azucena villaflor             168
1144219             79       azucena villaflor               8
1144220             79       azucena villaflor              75
1144221             79       azucena villaflor             123

                nombre_destino
0         barrancas de belgrano
1                       gainza
2                  de los incas
3                    villarroel
4                       malabia
…                            …
1144217               ecoparque
1144218          estados unidos
1144219                congreso
1144220   plaza primero de mayo
1144221                 armenia

[1144185 rows x 12 columns]
```

[51]:
```
drop_duplicados = dataset_2021.drop_duplicates(subset = ["codigo_origen",␣
↪"nombre_origen"])
duplicados_first = drop_duplicados.duplicated(subset = ["codigo_origen"])
drop_duplicados[duplicados_first]
```

[51]:
```
Empty DataFrame
Columns: [minutos, id_usuario, long_estacion_origen, lat_estacion_origen,
long_estacion_destino, lat_estacion_destino, año, genero_usuario, codigo_origen,
nombre_origen, codigo_destino, nombre_destino]
Index: []
```

Los nulos de lat y long en todas las filas se resolverán luego al emprolijar el resto del dataset

2020:

[52]:
```
dataset_2020
```

[52]:

	minutos	id_estacion_origen	nombre_estacion_origen	\
0	15	116	116 - HOSPITAL ALEMÁN	
1	11	14	014 - Pacifico	
2	17	134	134 - DON BOSCO	
3	16	134	134 - DON BOSCO	
4	12	194	194 - PERÓN Y ACUÑA DE FIGUEROA	
…	…	…	…	
2002940	36	151	151 - AIME PAINÉ	
2002941	27	150	150 - VERA PEÑALOZA	
2002942	43	150	150 - VERA PEÑALOZA	
2002943	82	114	114 - DELLA PAOLERA	
2002944	177	353	237 - Madero Office	

	id_estacion_destino	nombre_estacion_destino	id_usuario	\
0	214.0	142 - Armenia y Gorriti	666202	
1	214.0	142 - Armenia y Gorriti	666202	
2	214.0	142 - Armenia y Gorriti	666202	
3	214.0	142 - Armenia y Gorriti	666202	
4	214.0	142 - Armenia y Gorriti	666202	
…	…	…	…	
2002940	179.0	179 - CASA SAN	634757	
2002941	179.0	179 - CASA SAN	641709	
2002942	179.0	179 - CASA SAN	641712	
2002943	179.0	179 - CASA SAN	412605	
2002944	179.0	179 - CASA SAN	630939	

	long_estacion_origen	lat_estacion_origen	long_estacion_destino	\
0	-58.402586	-34.592171	-58.428972	
1	-58.426385	-34.577424	-58.428972	
2	-58.416777	-34.612231	-58.428972	
3	-58.416777	-34.612231	-58.428972	
4	-58.422461	-34.606076	-58.428972	
…	…	…	…	
2002940	-58.361280	-34.611816	-58.364284	
2002941	-58.355744	-34.618841	-58.364284	
2002942	-58.355744	-34.618841	-58.364284	
2002943	-58.372251	-34.594975	-58.364284	
2002944	-58.364690	-34.599037	-58.364284	

	lat_estacion_destino	año	genero_usuario
0	-34.590541	2020	F
1	-34.590541	2020	F
2	-34.590541	2020	F
3	-34.590541	2020	F
4	-34.590541	2020	F

```
…                          …    …              …
2002940             -34.638480  2020              M
2002941             -34.638480  2020              M
2002942             -34.638480  2020              F
2002943             -34.638480  2020              M
2002944             -34.638480  2020              F

[2002945 rows x 12 columns]
```

[53]:
```
%%capture [--no-stderr]
dataset_2020.loc[dataset_2020.nombre_estacion_origen.isnull() == False,␣
 ↪"nombre_estacion_origen"] = dataset_2020.loc[dataset_2020.
 ↪nombre_estacion_origen.isnull() == False, "nombre_estacion_origen"].
 ↪apply(lambda x : unidecode.unidecode(x))
dataset_2020.loc[dataset_2020.nombre_estacion_origen.isnull() == False,␣
 ↪"nombre_estacion_origen"] = dataset_2020.loc[dataset_2020.
 ↪nombre_estacion_origen.isnull() == False, "nombre_estacion_origen"].str.
 ↪lower()

dataset_2020.loc[dataset_2020.nombre_estacion_destino.isnull() == False,␣
 ↪"nombre_estacion_destino"] = dataset_2020.loc[dataset_2020.
 ↪nombre_estacion_destino.isnull() == False, "nombre_estacion_destino"].
 ↪apply(lambda x : unidecode.unidecode(x))
dataset_2020.loc[dataset_2020.nombre_estacion_destino.isnull() == False,␣
 ↪"nombre_estacion_destino"] = dataset_2020.loc[dataset_2020.
 ↪nombre_estacion_destino.isnull() == False, "nombre_estacion_destino"].str.
 ↪lower()
```

[54]:
```
dataset_2020.isnull().sum()
```

[54]:
```
minutos                  0
id_estacion_origen       0
nombre_estacion_origen   0
id_estacion_destino      0
nombre_estacion_destino  0
id_usuario               0
long_estacion_origen     0
lat_estacion_origen      0
long_estacion_destino    0
lat_estacion_destino     0
año                      0
genero_usuario           0
dtype: int64
```

[55]:
```
# Se separa el código del nombre
# Origen:
col_nombre_origen = dataset_2020.loc[:,"nombre_estacion_origen"]
```

```
nombre_origen_2020 = col_nombre_origen.str.split("-", expand = True)[1].str.
 ↪strip()
id_origen_2020 = col_nombre_origen.str.split("-", expand = True)[0].
 ↪astype("int")
dataset_2020 = pd.concat((dataset_2020, id_origen_2020, nombre_origen_2020),␣
 ↪axis = 1).rename(columns= {1 : "nombre_origen", 0 : "codigo_origen"})

# Destino:
col_nombre_destino = dataset_2020.loc[:,"nombre_estacion_destino"]
nombre_destino_2020 = col_nombre_destino.str.split("-", expand = True)[1].str.
 ↪strip()
id_destino_2020 = col_nombre_destino.str.split("-", expand = True)[0].
 ↪astype("int")
dataset_2020 = pd.concat((dataset_2020, id_destino_2020, nombre_destino_2020),␣
 ↪axis = 1).rename(columns= {1 : "nombre_destino", 0 : "codigo_destino"})

# Drop de columnas viejas:
dataset_2020.drop(["nombre_estacion_origen", "id_estacion_origen",␣
 ↪"id_estacion_destino", "nombre_estacion_destino"], axis = 1, inplace = True)
dataset_2020
```

[55]:

	minutos	id_usuario	long_estacion_origen	lat_estacion_origen	long_estacion_destino	lat_estacion_destino	año	genero_usuario
0	15	666202	-58.402586	-34.592171	-58.428972	-34.590541	2020	F
1	11	666202	-58.426385	-34.577424	-58.428972	-34.590541	2020	F
2	17	666202	-58.416777	-34.612231	-58.428972	-34.590541	2020	F
3	16	666202	-58.416777	-34.612231	-58.428972	-34.590541	2020	F
4	12	666202	-58.422461	-34.606076	-58.428972	-34.590541	2020	F
…	…	…	…	…	…	…	…	…
2002940	36	634757	-58.361280	-34.611816	-58.364284	-34.638480	2020	M
2002941	27	641709	-58.355744	-34.618841	-58.364284	-34.638480	2020	M
2002942	43	641712	-58.355744	-34.618841	-58.364284	-34.638480	2020	F
2002943	82	412605	-58.372251	-34.594975	-58.364284	-34.638480	2020	M
2002944	177	630939	-58.364690	-34.599037	-58.364284	-34.638480	2020	F

```
         codigo_origen              nombre_origen  codigo_destino  \
0                  116           hospital aleman             142
1                   14                  pacifico             142
2                  134                 don bosco             142
3                  134                 don bosco             142
4                  194  peron y acuna de figueroa             142
…                  …                         …              …
2002940            151                aime paine             179
2002941            150              vera penaloza            179
2002942            150              vera penaloza            179
2002943            114              della paolera            179
2002944            237              madero office            179

           nombre_destino
0        armenia y gorriti
1        armenia y gorriti
2        armenia y gorriti
3        armenia y gorriti
4        armenia y gorriti
…                      …
2002940          casa san
2002941          casa san
2002942          casa san
2002943          casa san
2002944          casa san

[2002945 rows x 12 columns]
```

[56]:
```
dataset_2020.isnull().sum()
```

[56]:
```
minutos                 0
id_usuario              0
long_estacion_origen    0
lat_estacion_origen     0
long_estacion_destino   0
lat_estacion_destino    0
año                     0
genero_usuario          0
codigo_origen           0
nombre_origen           0
codigo_destino          0
nombre_destino          0
dtype: int64
```

[57]:
```
drop_duplicados = dataset_2020.drop_duplicates(subset = ["codigo_origen",␣
↪"nombre_origen"])
```

```
drop_duplicados
```

[57]:

```
       minutos  id_usuario  long_estacion_origen  lat_estacion_origen  \
0           15      666202            -58.402586           -34.592171
1           11      666202            -58.426385           -34.577424
2           17      666202            -58.416777           -34.612231
4           12      666202            -58.422461           -34.606076
5           27      666202            -58.428972           -34.590541
…          …         …                   …                    …
22202       26      279101            -58.355744           -34.618841
23398       15      690136            -58.378858           -34.582422
26597       51        1827            -58.390478           -34.623924
30309       19       92175            -58.365717           -34.627537
34447       31      588772            -58.356254           -34.628759

       long_estacion_destino  lat_estacion_destino   año genero_usuario  \
0                 -58.428972            -34.590541  2020              F
1                 -58.428972            -34.590541  2020              F
2                 -58.428972            -34.590541  2020              F
4                 -58.428972            -34.590541  2020              F
5                 -58.428972            -34.590541  2020              F
…                      …                   …   …           …
22202             -58.364681            -34.605489  2020              M
23398             -58.368256            -34.611033  2020              M
26597             -58.459297            -34.565409  2020              M
30309             -58.369220            -34.640269  2020              F
34447             -58.416012            -34.622261  2020              M

       codigo_origen              nombre_origen  codigo_destino  \
0                116            hospital aleman             142
1                 14                   pacifico             142
2                134                  don bosco             142
4                194  peron y acuna de figueroa             142
5                142          armenia y gorriti             142
…               …                       …              …
22202            150              rodrigo bueno             111
23398            393                  barrio 31               3
26597            373        jorgelina de simone             240
30309            196          hospital argerich             126
34447            108             usina del arte             199

                        nombre_destino
0                   armenia y gorriti
1                   armenia y gorriti
2                   armenia y gorriti
4                   armenia y gorriti
5                   armenia y gorriti
```

```
…                                    …
22202                       macacha guemes
23398                              aduana
26597                           echeverria
30309  ministerio de justicia y seguridad
34447               estados unidos y boedo

[201 rows x 12 columns]
```

[58]:
```
duplicados_first = drop_duplicados.duplicated(subset = ["codigo_origen"])
duplicados_last = drop_duplicados.duplicated(subset = ["codigo_origen"], keep =␣
↪"last")
drop_duplicados[duplicados_first | duplicados_last]
```

[58]:
```
       minutos  id_usuario  long_estacion_origen  lat_estacion_origen  \
11763       31        6392            -58.355744           -34.618841
22202       26      279101            -58.355744           -34.618841

       long_estacion_destino  lat_estacion_destino   año genero_usuario  \
11763             -58.403865            -34.588329  2020              M
22202             -58.364681            -34.605489  2020              M

       codigo_origen  nombre_origen  codigo_destino    nombre_destino
11763            150  vera penaloza             200  austria y french
22202            150  rodrigo bueno             111    macacha guemes
```

[59]:
```
drop_duplicados_destino = dataset_2020.drop_duplicates(subset =␣
↪["codigo_destino", "nombre_destino"])
drop_duplicados_destino
```

[59]:
```
       minutos  id_usuario  long_estacion_origen  lat_estacion_origen  \
0           15      666202            -58.402586           -34.592171
13          26      666202            -58.390598           -34.583749
14           8      666202            -58.416777           -34.612231
15          21      666202            -58.407738           -34.585443
17          21      666202            -58.411275           -34.572165
…          …          …                      …                    …
19181       34      481819            -58.374538           -34.592106
19889       17      186810            -58.387669           -34.635361
20619       47      432292            -58.411656           -34.602782
26621       42        1827            -58.459297           -34.565409
31397       34      335330            -58.416012           -34.622261

       long_estacion_destino  lat_estacion_destino   año genero_usuario  \
0                 -58.428972            -34.590541  2020              F
13                -58.390598            -34.583749  2020              F
14                -58.421982            -34.598119  2020              F
```

```
15                -58.407738            -34.585443  2020              F
17                -58.426058            -34.592687  2020              F
…                      …                     …   …             …
19181             -58.362127            -34.624256  2020              M
19889             -58.395849            -34.630375  2020              M
20619             -58.378858            -34.582422  2020              F
26621             -58.390478            -34.623924  2020              M
31397             -58.362066            -34.630778  2020              F

       codigo_origen           nombre_origen  codigo_destino  \
0                116         hospital aleman             142
13                 1     facultad de derecho               1
14               134              don bosco              54
15                 9       parque las heras               9
17               335        general urquiza              70
…                …                       …             …
19181            130             retiro ii              34
19889            138     hospital britanico             107
20619             96          carlos gardel             393
26621            240             echeverria             373
31397            199  estados unidos y boedo             153

             nombre_destino
0          armenia y gorriti
13       facultad de derecho
14         acuna de figueroa
15          parque las heras
17                    araoz
…                     …
19181        colonia express
19889      hospital garrahan
20619             barrio 31
26621    jorgelina de simone
31397  juan manuel de blanes

[201 rows x 12 columns]
```

[60]:
```
duplicados_first = drop_duplicados_destino.duplicated(subset =␣
 ↪["codigo_destino"])
duplicados_last = drop_duplicados_destino.duplicated(subset =␣
 ↪["codigo_destino"], keep = "last")
drop_duplicados_destino[duplicados_first | duplicados_last]
```

[60]:
```
      minutos  id_usuario  long_estacion_origen  lat_estacion_origen  \
7059       45      711275            -58.406162           -34.590863
7490       42      643758            -58.390598           -34.583749
```

```
      long_estacion_destino  lat_estacion_destino   año genero_usuario  \
7059             -58.355744            -34.618841  2020              F
7490             -58.355744            -34.618841  2020              M

      codigo_origen        nombre_origen  codigo_destino nombre_destino
7059            193    arenales y aguero             150  rodrigo bueno
7490              1  facultad de derecho             150  vera penaloza
```

[61]:
```
estaciones_nuevo.loc[estaciones_nuevo.codigo == 150, :]
```

[61]:
```
                                          WKT   id  codigo  \
92  POINT (-58.3557441485293 -34.6188407590044)  129     150

                 nombre        ubicacion        tipo  \
92  150 - RODRIGO BUENO  Av. España 2200  AUTOMÁTICA

                                           horario  anclajes_t
92  Estación automática: disponibilidad las 24 horas          30
```

[62]:
```
estaciones_nuevo.loc[estaciones_nuevo.WKT.str.contains('-58.355744'),:]
```

[62]:
```
                                          WKT   id  codigo  \
92  POINT (-58.3557441485293 -34.6188407590044)  129     150

                 nombre        ubicacion        tipo  \
92  150 - RODRIGO BUENO  Av. España 2200  AUTOMÁTICA

                                           horario  anclajes_t
92  Estación automática: disponibilidad las 24 horas          30
```

[63]:
```
# Vera Peñaloza no ofrece resultados. Se cambiará el nombre por Rodrigo Bueno:
dataset_2020.replace({"vera penaloza" : "rodrigo bueno"}, regex=True, inplace =␣
↪True)
```

[64]:
```
drop_duplicados_destino = dataset_2020.drop_duplicates(subset =␣
↪["codigo_destino", "nombre_destino"])
duplicados_first = drop_duplicados_destino.duplicated(subset =␣
↪["codigo_destino"])
duplicados_last = drop_duplicados_destino.duplicated(subset =␣
↪["codigo_destino"], keep = "last")
drop_duplicados_destino[duplicados_first | duplicados_last]
```

[64]:
```
Empty DataFrame
Columns: [minutos, id_usuario, long_estacion_origen, lat_estacion_origen,
long_estacion_destino, lat_estacion_destino, año, genero_usuario, codigo_origen,
nombre_origen, codigo_destino, nombre_destino]
Index: []
```

2019:

[65]:
```
%%capture [--no-stderr]
dataset_2019.loc[dataset_2019.nombre_estacion_origen.isnull() == False,␣
 ↪"nombre_estacion_origen"] = dataset_2019.loc[dataset_2019.
 ↪nombre_estacion_origen.isnull() == False, "nombre_estacion_origen"].
 ↪apply(lambda x : unidecode.unidecode(x))
dataset_2019.loc[dataset_2019.nombre_estacion_origen.isnull() == False,␣
 ↪"nombre_estacion_origen"] = dataset_2019.loc[dataset_2019.
 ↪nombre_estacion_origen.isnull() == False, "nombre_estacion_origen"].str.
 ↪lower()

dataset_2019.loc[dataset_2019.nombre_estacion_destino.isnull() == False,␣
 ↪"nombre_estacion_destino"] = dataset_2019.loc[dataset_2019.
 ↪nombre_estacion_destino.isnull() == False, "nombre_estacion_destino"].
 ↪apply(lambda x : unidecode.unidecode(x))
dataset_2019.loc[dataset_2019.nombre_estacion_destino.isnull() == False,␣
 ↪"nombre_estacion_destino"] = dataset_2019.loc[dataset_2019.
 ↪nombre_estacion_destino.isnull() == False, "nombre_estacion_destino"].str.
 ↪lower()
```

[66]:
```
dataset_2019
```

[66]:

	año	genero_usuario	id_estacion_origen	nombre_estacion_origen	\
0	2019	F	168	estados unidos	
1	2019	F	141	solis y alsina	
2	2019	F	76	ayacucho	
3	2019	F	148	constitucion ii	
4	2019	F	88	misiones	
…	…	…	…	…	
6367305	2019	M	283	228 - baldomero	
6367306	2019	F	283	228 - baldomero	
6367307	2019	M	283	228 - baldomero	
6367308	2019	M	283	228 - baldomero	
6367309	2019	M	283	228 - baldomero	

	long_estacion_origen	lat_estacion_origen	id_estacion_destino	\
0	-58.381283	-34.618620	112.0	
1	-58.390411	-34.611838	93.0	
2	-58.394927	-34.607573	171.0	
3	-58.379232	-34.627494	76.0	
4	-58.404230	-34.612867	18.0	
…	…	…	…	
6367305	-58.456035	-34.635505	283.0	
6367306	-58.456035	-34.635505	283.0	
6367307	-58.456035	-34.635505	283.0	
6367308	-58.456035	-34.635505	283.0	
6367309	-58.456035	-34.635505	283.0	

```
        nombre_estacion_destino  long_estacion_destino  lat_estacion_destino  \
0                     9 de julio             -58.380766            -34.612389
1                   carlos calvo             -58.394769            -34.620601
2                        pasteur             -58.399527            -34.603243
3                      ayacucho             -58.394927            -34.607573
4                   independencia             -58.380481            -34.617373
…                             …                    …                     …
6367305         228 - baldomero             -58.456035            -34.635505
6367306         228 - baldomero             -58.456035            -34.635505
6367307         228 - baldomero             -58.456035            -34.635505
6367308         228 - baldomero             -58.456035            -34.635505
6367309         228 - baldomero             -58.456035            -34.635505

         id_usuario minutos
0          115783.0      07
1          115783.0      10
2          115783.0      14
3          115783.0      50
4          115783.0      27
…             …     …
6367305    177162.0      26
6367306    585275.0      14
6367307    494906.0      41
6367308    618404.0      43
6367309    606607.0      37

[6367165 rows x 12 columns]
```

[67]:
```
# Hay un caso con nombre diferenciado por "&" vs "y":
fitz_roy = dataset_2019.loc[dataset_2019.nombre_estacion_origen.str.
 ↪contains('fitz'),"nombre_estacion_origen"]
fitz_roy
```

[67]:
```
14               fitz roy y gorriti
122                        fitz roy
162        159 - fitz roy & gorriti
177              fitz roy y gorriti
191              fitz roy y gorriti
                      …
6364623    159 - fitz roy & gorriti
6364787    159 - fitz roy & gorriti
6365585    159 - fitz roy & gorriti
6366490    159 - fitz roy & gorriti
6366770          101 - fitz roy
Name: nombre_estacion_origen, Length: 42778, dtype: object
```

[68]:
```
%%capture [--no-stderr]
dataset_2019.replace({"fitz roy y gorriti" : "fitz roy & gorriti"}, regex=True,␣
↪inplace = True)
```

[69]:
```
# Se separa el código del nombre. Lamentablemente no todas las filas lo tienen,␣
↪por lo que se filtra las que tienen "-" y, en los casos negativos se␣
↪mantendrá el id original

# Origen:
col_nombre_origen = dataset_2019.loc[dataset_2019.nombre_estacion_destino.str.
↪contains('-'),"nombre_estacion_origen"]
nombre_origen_2019 = col_nombre_origen.str.split("-", expand = True)[1].str.
↪strip()
id_origen_2019 = col_nombre_origen.str.split("-", expand = True)[0].
↪astype("int")
dataset_2019 = pd.concat((dataset_2019, id_origen_2019, nombre_origen_2019),␣
↪axis = 1).rename(columns= {1 : "nombre_origen", 0 : "codigo_origen"})

# Destino:
col_nombre_destino = dataset_2019.loc[dataset_2019.nombre_estacion_destino.str.
↪contains('-'),"nombre_estacion_destino"]
nombre_destino_2019 = col_nombre_destino.str.split("-", expand = True)[1].str.
↪strip()
id_destino_2019 = col_nombre_destino.str.split("-", expand = True)[0].
↪astype("int")
dataset_2019 = pd.concat((dataset_2019, id_destino_2019, nombre_destino_2019),␣
↪axis = 1).rename(columns= {1 : "nombre_destino", 0 : "codigo_destino"})
```

[70]:
```
dataset_2019
```

[70]:

	año	genero_usuario	id_estacion_origen	nombre_estacion_origen	\
0	2019	F	168	estados unidos	
1	2019	F	141	solis y alsina	
2	2019	F	76	ayacucho	
3	2019	F	148	constitucion ii	
4	2019	F	88	misiones	
…	…	…	…	…	
6367305	2019	M	283	228 - baldomero	
6367306	2019	F	283	228 - baldomero	
6367307	2019	M	283	228 - baldomero	
6367308	2019	M	283	228 - baldomero	
6367309	2019	M	283	228 - baldomero	

	long_estacion_origen	lat_estacion_origen	id_estacion_destino	\
0	-58.381283	-34.618620	112.0	
1	-58.390411	-34.611838	93.0	
2	-58.394927	-34.607573	171.0	

```
3                  -58.379232          -34.627494                 76.0
4                  -58.404230          -34.612867                 18.0
…                        …                   …                    …
6367305            -58.456035          -34.635505                283.0
6367306            -58.456035          -34.635505                283.0
6367307            -58.456035          -34.635505                283.0
6367308            -58.456035          -34.635505                283.0
6367309            -58.456035          -34.635505                283.0

        nombre_estacion_destino  long_estacion_destino  lat_estacion_destino  \
0                     9 de julio             -58.380766            -34.612389
1                   carlos calvo             -58.394769            -34.620601
2                        pasteur             -58.399527            -34.603243
3                      ayacucho             -58.394927            -34.607573
4                  independencia             -58.380481            -34.617373
…                            …                    …                     …
6367305          228 - baldomero             -58.456035            -34.635505
6367306          228 - baldomero             -58.456035            -34.635505
6367307          228 - baldomero             -58.456035            -34.635505
6367308          228 - baldomero             -58.456035            -34.635505
6367309          228 - baldomero             -58.456035            -34.635505

         id_usuario minutos  codigo_origen nombre_origen  codigo_destino  \
0          115783.0      07            NaN           NaN             NaN
1          115783.0      10            NaN           NaN             NaN
2          115783.0      14            NaN           NaN             NaN
3          115783.0      50            NaN           NaN             NaN
4          115783.0      27            NaN           NaN             NaN
…               …     …             …            …              …
6367305    177162.0      26          228.0     baldomero           228.0
6367306    585275.0      14          228.0     baldomero           228.0
6367307    494906.0      41          228.0     baldomero           228.0
6367308    618404.0      43          228.0     baldomero           228.0
6367309    606607.0      37          228.0     baldomero           228.0

        nombre_destino
0                  NaN
1                  NaN
2                  NaN
3                  NaN
4                  NaN
…                  …
6367305      baldomero
6367306      baldomero
6367307      baldomero
6367308      baldomero
6367309      baldomero
```

```
[6367165 rows x 16 columns]
```

[71]:
```
dataset_2019.isnull().sum()
```

[71]:
```
año                           0
genero_usuario                0
id_estacion_origen        14693
nombre_estacion_origen        0
long_estacion_origen      16828
lat_estacion_origen       16828
id_estacion_destino       14803
nombre_estacion_destino       0
long_estacion_destino     16970
lat_estacion_destino      16970
id_usuario                    0
minutos                       0
codigo_origen           2860094
nombre_origen           2860094
codigo_destino          2860094
nombre_destino          2860094
dtype: int64
```

[72]:
```
dataframe_codigo_origen_2019 = dataset_2019.loc[:,["id_estacion_origen",␣
 ↪"codigo_origen","nombre_origen"]]
dataframe_codigo_origen_2019.dropna(subset = ["codigo_origen"], inplace = True)
dataframe_codigo_origen_2019.drop_duplicates(subset = ["codigo_origen"],␣
 ↪inplace = True)
dataframe_codigo_origen_2019.rename(columns = {"codigo_origen" :␣
 ↪"codigo_origen_merge", "nombre_origen" : "nombre_origen_merge"}, inplace =␣
 ↪True)
dataframe_codigo_origen_2019
```

[72]:

	id_estacion_origen	codigo_origen_merge	nombre_origen_merge
114	205	125.0	f.j.santamaria de oro
115	352	236.0	guemes
127	11	11.0	tribunales
128	400	313.0	de los incas
161	279	325.0	castelli y peron
…	…	…	…
29843	127	127.0	santos dumont y otero
32924	356	245.0	arribenos
33265	330	148.0	paternal
38675	283	228.0	baldomero
40244	404	317.0	cero mas infinito

```
[396 rows x 3 columns]
```

[73]:
```
dataset_2019 = dataset_2019.merge(dataframe_codigo_origen_2019,␣
↪on='id_estacion_origen', how='left')
```

[74]:
```
dataframe_codigo_destino_2019 = dataset_2019.loc[:,["id_estacion_destino",␣
↪"codigo_destino","nombre_destino"]]
dataframe_codigo_destino_2019.dropna(subset = ["codigo_destino"], inplace =␣
↪True)
dataframe_codigo_destino_2019.drop_duplicates(subset = ["codigo_destino"],␣
↪inplace = True)
dataframe_codigo_destino_2019.rename(columns = {"codigo_destino" :␣
↪"codigo_destino_merge", "nombre_destino" : "nombre_destino_merge"}, inplace␣
↪= True)
dataframe_codigo_destino_2019
```

[74]:
```
       id_estacion_destino  codigo_destino_merge nombre_destino_merge
114                  206.0                 159.0   fitz roy & gorriti
115                  190.0                 190.0               juncal
119                  352.0                 236.0               guemes
127                  247.0                 282.0             tronador
128                  258.0                 336.0             la pampa
…                      …                     …                    …
33253                330.0                 148.0             paternal
33747                411.0                 368.0             ensenada
36261                203.0                 108.0       usina del arte
38670                283.0                 228.0            baldomero
40355                386.0                 277.0              coghlan

[396 rows x 3 columns]
```

[75]:
```
dataset_2019 = dataset_2019.merge(dataframe_codigo_destino_2019,␣
↪on='id_estacion_destino', how='left')
```

[76]:
```
dataset_2019
```

[76]:
```
          año genero_usuario id_estacion_origen nombre_estacion_origen  \
0        2019              F                168         estados unidos
1        2019              F                141         solis y alsina
2        2019              F                 76                ayacucho
3        2019              F                148        constitucion ii
4        2019              F                 88                misiones
…         …                …                 …                      …
6367160  2019              M                283        228 - baldomero
6367161  2019              F                283        228 - baldomero
6367162  2019              M                283        228 - baldomero
6367163  2019              M                283        228 - baldomero
6367164  2019              M                283        228 - baldomero
```

```
         long_estacion_origen  lat_estacion_origen  id_estacion_destino  \
0                  -58.381283           -34.618620                112.0
1                  -58.390411           -34.611838                 93.0
2                  -58.394927           -34.607573                171.0
3                  -58.379232           -34.627494                 76.0
4                  -58.404230           -34.612867                 18.0
…                         …                    …                    …
6367160            -58.456035           -34.635505                283.0
6367161            -58.456035           -34.635505                283.0
6367162            -58.456035           -34.635505                283.0
6367163            -58.456035           -34.635505                283.0
6367164            -58.456035           -34.635505                283.0

        nombre_estacion_destino  long_estacion_destino  lat_estacion_destino  \
0                     9 de julio             -58.380766           -34.612389
1                   carlos calvo             -58.394769           -34.620601
2                        pasteur             -58.399527           -34.603243
3                      ayacucho             -58.394927           -34.607573
4                  independencia             -58.380481           -34.617373
…                            …                    …                    …
6367160          228 - baldomero             -58.456035           -34.635505
6367161          228 - baldomero             -58.456035           -34.635505
6367162          228 - baldomero             -58.456035           -34.635505
6367163          228 - baldomero             -58.456035           -34.635505
6367164          228 - baldomero             -58.456035           -34.635505

         id_usuario minutos  codigo_origen nombre_origen  codigo_destino  \
0          115783.0      07            NaN           NaN             NaN
1          115783.0      10            NaN           NaN             NaN
2          115783.0      14            NaN           NaN             NaN
3          115783.0      50            NaN           NaN             NaN
4          115783.0      27            NaN           NaN             NaN
…               …     …            …            …              …
6367160    177162.0      26          228.0     baldomero           228.0
6367161    585275.0      14          228.0     baldomero           228.0
6367162    494906.0      41          228.0     baldomero           228.0
6367163    618404.0      43          228.0     baldomero           228.0
6367164    606607.0      37          228.0     baldomero           228.0

        nombre_destino  codigo_origen_merge   nombre_origen_merge  \
0                  NaN                168.0        estados unidos
1                  NaN                141.0        solis y alsina
2                  NaN                 76.0              ayacucho
3                  NaN                323.0  parque avellaneda ii
4                  NaN                 88.0              misiones
…                  …                    …                    …
6367160      baldomero                228.0             baldomero
```

```
6367161      baldomero               228.0            baldomero
6367162      baldomero               228.0            baldomero
6367163      baldomero               228.0            baldomero
6367164      baldomero               228.0            baldomero

         codigo_destino_merge nombre_destino_merge
0                       112.0           9 de julio
1                        93.0         carlos calvo
2                       171.0              pasteur
3                        76.0             ayacucho
4                         NaN                  NaN
…                         …                    …
6367160                 228.0            baldomero
6367161                 228.0            baldomero
6367162                 228.0            baldomero
6367163                 228.0            baldomero
6367164                 228.0            baldomero

[6367165 rows x 20 columns]
```

[77]:
```
dataset_2019.isnull().sum()
```

[77]:
```
año                           0
genero_usuario                0
id_estacion_origen        14693
nombre_estacion_origen        0
long_estacion_origen      16828
lat_estacion_origen       16828
id_estacion_destino       14803
nombre_estacion_destino       0
long_estacion_destino     16970
lat_estacion_destino      16970
id_usuario                    0
minutos                       0
codigo_origen           2860094
nombre_origen           2860094
codigo_destino          2860094
nombre_destino          2860094
codigo_origen_merge       48446
nombre_origen_merge       48446
codigo_destino_merge      48026
nombre_destino_merge      48026
dtype: int64
```

[78]:
```
drop_duplicados_origen = dataset_2019.drop_duplicates(subset =␣
 ↪["codigo_origen_merge", "nombre_origen_merge"])
```

```
duplicados_first = drop_duplicados_origen.duplicated(subset =␣
 ↪["codigo_origen_merge"])
duplicados_last = drop_duplicados_origen.duplicated(subset =␣
 ↪["codigo_origen_merge"], keep = "last")
drop_duplicados_origen[duplicados_first | duplicados_last]
```

[78]:
```
Empty DataFrame
Columns: [año, genero_usuario, id_estacion_origen, nombre_estacion_origen,
long_estacion_origen, lat_estacion_origen, id_estacion_destino,
nombre_estacion_destino, long_estacion_destino, lat_estacion_destino,
id_usuario, minutos, codigo_origen, nombre_origen, codigo_destino,
nombre_destino, codigo_origen_merge, nombre_origen_merge, codigo_destino_merge,
nombre_destino_merge]
Index: []
```

[79]:
```
drop_duplicados_destino = dataset_2019.drop_duplicates(subset =␣
 ↪["codigo_destino_merge", "nombre_destino_merge"])
duplicados_first = drop_duplicados_destino.duplicated(subset =␣
 ↪["codigo_destino_merge"])
duplicados_last = drop_duplicados_destino.duplicated(subset =␣
 ↪["codigo_destino_merge"], keep = "last")
drop_duplicados_destino[duplicados_first | duplicados_last]
```

[79]:
```
Empty DataFrame
Columns: [año, genero_usuario, id_estacion_origen, nombre_estacion_origen,
long_estacion_origen, lat_estacion_origen, id_estacion_destino,
nombre_estacion_destino, long_estacion_destino, lat_estacion_destino,
id_usuario, minutos, codigo_origen, nombre_origen, codigo_destino,
nombre_destino, codigo_origen_merge, nombre_origen_merge, codigo_destino_merge,
nombre_destino_merge]
Index: []
```

[80]:
```
dataset_2019.dropna(subset = ["codigo_origen_merge", "nombre_origen_merge",␣
 ↪"codigo_destino_merge", "nombre_destino_merge"], inplace = True)
dataset_2019.drop(["nombre_estacion_origen", "id_estacion_origen",␣
 ↪"nombre_estacion_destino", "id_estacion_destino", "codigo_origen",␣
 ↪"nombre_origen", "codigo_destino", "nombre_destino"], axis = 1, inplace =␣
 ↪True)
dataset_2019.rename(columns = {"codigo_origen_merge" : "codigo_origen",␣
 ↪"nombre_origen_merge" : "nombre_origen", "codigo_destino_merge" :␣
 ↪"codigo_destino", "nombre_destino_merge" : "nombre_destino"}, inplace = True)
dataset_2019
```

[80]:

	año	genero_usuario	long_estacion_origen	lat_estacion_origen	\
0	2019	F	-58.381283	-34.618620	
1	2019	F	-58.390411	-34.611838	
2	2019	F	-58.394927	-34.607573	

```
3        2019              F            -58.379232           -34.627494
5        2019              F            -58.395897           -34.615196
…        …                 …             …                   …
6367160  2019              M            -58.456035           -34.635505
6367161  2019              F            -58.456035           -34.635505
6367162  2019              M            -58.456035           -34.635505
6367163  2019              M            -58.456035           -34.635505
6367164  2019              M            -58.456035           -34.635505

         long_estacion_destino  lat_estacion_destino  id_usuario minutos  \
0                   -58.380766            -34.612389    115783.0      07
1                   -58.394769            -34.620601    115783.0      10
2                   -58.399527            -34.603243    115783.0      14
3                   -58.394927            -34.607573    115783.0      50
5                   -58.406617            -34.609355    115783.0      08
…                          …                     …         …     …
6367160             -58.456035            -34.635505    177162.0      26
6367161             -58.456035            -34.635505    585275.0      14
6367162             -58.456035            -34.635505    494906.0      41
6367163             -58.456035            -34.635505    618404.0      43
6367164             -58.456035            -34.635505    606607.0      37

         codigo_origen         nombre_origen  codigo_destino nombre_destino
0                168.0        estados unidos           112.0     9 de julio
1                141.0        solis y alsina            93.0   carlos calvo
2                 76.0               ayacucho           171.0        pasteur
3                323.0  parque avellaneda ii             76.0       ayacucho
5                106.0                rincon           163.0        once ii
…                …                      …              …            …
6367160          228.0             baldomero           228.0      baldomero
6367161          228.0             baldomero           228.0      baldomero
6367162          228.0             baldomero           228.0      baldomero
6367163          228.0             baldomero           228.0      baldomero
6367164          228.0             baldomero           228.0      baldomero

[6281083 rows x 12 columns]
```

[81]:
```
dataset_2019.isnull().sum()
```

[81]:
```
año                       0
genero_usuario            0
long_estacion_origen      0
lat_estacion_origen       0
long_estacion_destino   313
lat_estacion_destino    313
id_usuario                0
minutos                   0
```

```
codigo_origen              0
nombre_origen              0
codigo_destino             0
nombre_destino             0
dtype: int64
```

[82]:
```
dataset_2019[dataset_2019.lat_estacion_destino.isnull()]
```

[82]:
```
         año genero_usuario  long_estacion_origen  lat_estacion_origen  \
37      2019              F            -58.405790           -34.601780
50      2019              M            -58.405790           -34.601780
3163    2019              M            -58.393317           -34.589150
3421    2019              F            -58.439709           -34.603162
4028    2019              M            -58.413859           -34.594629
…       …              …                 …                 …
446865  2019              M            -58.425816           -34.592673
458426  2019              F            -58.405313           -34.582472
461226  2019              M            -58.374282           -34.595881
462786  2019              M            -58.420997           -34.580127
462789  2019              M            -58.420997           -34.580127

        long_estacion_destino  lat_estacion_destino  id_usuario minutos  \
37                        NaN                   NaN    570578.0      19
50                        NaN                   NaN    588687.0      19
3163                      NaN                   NaN     31574.0      17
3421                      NaN                   NaN     83434.0      40
4028                      NaN                   NaN     28354.0      22
…                       …                     …          …     …
446865                    NaN                   NaN       370.0      49
458426                    NaN                   NaN    203384.0      13
461226                    NaN                   NaN    112067.0      27
462786                    NaN                   NaN    584730.0      17
462789                    NaN                   NaN    224422.0      16

        codigo_origen           nombre_origen  codigo_destino nombre_destino
37              144.0              pueyrredon            44.0      ecoparque
50              144.0              pueyrredon            44.0      ecoparque
3163            166.0  cementerio de recoleta            44.0      ecoparque
3421             31.0                 padilla            44.0      ecoparque
4028             66.0            billinghurst            44.0      ecoparque
…               …                       …             …            …
446865           70.0                   araoz            44.0      ecoparque
458426           89.0                 cabello            44.0      ecoparque
461226           53.0           ricardo rojas            44.0      ecoparque
462786            5.0            plaza italia            44.0      ecoparque
462789            5.0            plaza italia            44.0      ecoparque
```

```
[313 rows x 12 columns]
```

[83]:
```
eco_lat = dataset_2019.loc[(dataset_2019.nombre_destino == "ecoparque") &␣
 ↪(dataset_2019.lat_estacion_destino.notnull()), "lat_estacion_destino"].
 ↪iloc[0]
eco_long = dataset_2019.loc[(dataset_2019.nombre_destino == "ecoparque") &␣
 ↪(dataset_2019.long_estacion_destino.notnull()), "long_estacion_destino"].
 ↪iloc[0]

dataset_2019.fillna(value = {'lat_estacion_destino': eco_lat,␣
 ↪'long_estacion_destino': eco_long}, inplace = True)
dataset_2019.isnull().sum()
```

[83]:
```
año                    0
genero_usuario         0
long_estacion_origen   0
lat_estacion_origen    0
long_estacion_destino  0
lat_estacion_destino   0
id_usuario             0
minutos                0
codigo_origen          0
nombre_origen          0
codigo_destino         0
nombre_destino         0
dtype: int64
```

3.0.4 2018 - 2015:

En estos años "id_estacion" es el código de los años 2019 a 2021, por lo que la limpieza anterior no es necesaria

- A) 2018

[84]:
```
%%capture [--no-stderr]
dataset_2018.loc[dataset_2018.nombre_estacion_origen.isnull() == False,␣
 ↪"nombre_estacion_origen"] = dataset_2018.loc[dataset_2018.
 ↪nombre_estacion_origen.isnull() == False, "nombre_estacion_origen"].
 ↪apply(lambda x : unidecode.unidecode(x))
dataset_2018.loc[dataset_2018.nombre_estacion_origen.isnull() == False,␣
 ↪"nombre_estacion_origen"] = dataset_2018.loc[dataset_2018.
 ↪nombre_estacion_origen.isnull() == False, "nombre_estacion_origen"].str.
 ↪lower()

dataset_2018.loc[dataset_2018.nombre_estacion_destino.isnull() == False,␣
 ↪"nombre_estacion_destino"] = dataset_2018.loc[dataset_2018.
 ↪nombre_estacion_destino.isnull() == False, "nombre_estacion_destino"].
 ↪apply(lambda x : unidecode.unidecode(x))
```

```
dataset_2018.loc[dataset_2018.nombre_estacion_destino.isnull() == False,␣
 ↪"nombre_estacion_destino"] = dataset_2018.loc[dataset_2018.
 ↪nombre_estacion_destino.isnull() == False, "nombre_estacion_destino"].str.
 ↪lower()
```

[85]:
```
dataset_2018.isnull().sum()
```

[85]:
```
año                          0
genero_usuario               0
id_estacion_origen       29661
nombre_estacion_origen       0
long_estacion_origen     29661
lat_estacion_origen      29661
id_estacion_destino      29997
nombre_estacion_destino      0
long_estacion_destino    29997
lat_estacion_destino     29997
id_usuario                   0
minutos                      0
dtype: int64
```

[86]:
```
dataset_2018[dataset_2018.id_estacion_destino.isnull()].nombre_estacion_destino.
 ↪value_counts()
```

[86]:
```
ecoparque             15588
fitz roy y gorriti    14409
Name: nombre_estacion_destino, dtype: int64
```

[87]:
```
dataset_2018[dataset_2018.id_estacion_origen.isnull()].nombre_estacion_origen.
 ↪value_counts()
```

[87]:
```
ecoparque             15367
fitz roy y gorriti    14294
Name: nombre_estacion_origen, dtype: int64
```

[88]:
```
%%capture [--no-stderr]
# Ecoparque:
dataset_2018.loc[dataset_2018.nombre_estacion_origen == "ecoparque",␣
 ↪"id_estacion_origen"] = dataset_2019.loc[dataset_2019.nombre_destino ==␣
 ↪"ecoparque", "codigo_destino"].iloc[0]
dataset_2018.loc[dataset_2018.nombre_estacion_destino == "ecoparque",␣
 ↪"id_estacion_destino"] = dataset_2019.loc[dataset_2019.nombre_destino ==␣
 ↪"ecoparque", "codigo_destino"].iloc[0]
dataset_2018.loc[dataset_2018.nombre_estacion_origen == "ecoparque",␣
 ↪"lat_estacion_origen"] = eco_lat
dataset_2018.loc[dataset_2018.nombre_estacion_origen == "ecoparque",␣
 ↪"long_estacion_origen"] = eco_long
```

```
dataset_2018.loc[dataset_2018.nombre_estacion_destino == "ecoparque",␣
 ↪"lat_estacion_destino"] = eco_lat
dataset_2018.loc[dataset_2018.nombre_estacion_destino == "ecoparque",␣
 ↪"long_estacion_destino"] = eco_long

# Fitz Roy y Gorriti:
dataset_2018.loc[dataset_2018.nombre_estacion_origen == "fitz roy y gorriti",␣
 ↪"id_estacion_origen"] = dataset_2019.loc[dataset_2019.nombre_destino ==␣
 ↪"fitz roy & gorriti", "codigo_destino"].iloc[0]
dataset_2018.loc[dataset_2018.nombre_estacion_destino == "fitz roy y gorriti",␣
 ↪"id_estacion_destino"] = dataset_2019.loc[dataset_2019.nombre_destino ==␣
 ↪"fitz roy & gorriti", "codigo_destino"].iloc[0]
dataset_2018.loc[dataset_2018.nombre_estacion_origen == "fitz roy y gorriti",␣
 ↪"lat_estacion_origen"] = dataset_2019.loc[dataset_2019.nombre_destino ==␣
 ↪"fitz roy & gorriti", "lat_estacion_destino"].iloc[0]
dataset_2018.loc[dataset_2018.nombre_estacion_origen == "fitz roy y gorriti",␣
 ↪"long_estacion_origen"] = dataset_2019.loc[dataset_2019.nombre_destino ==␣
 ↪"fitz roy & gorriti", "long_estacion_destino"].iloc[0]
dataset_2018.loc[dataset_2018.nombre_estacion_destino == "fitz roy y gorriti",␣
 ↪"lat_estacion_destino"] = dataset_2019.loc[dataset_2019.nombre_destino ==␣
 ↪"fitz roy & gorriti", "lat_estacion_destino"].iloc[0]
dataset_2018.loc[dataset_2018.nombre_estacion_destino == "fitz roy y gorriti",␣
 ↪"long_estacion_destino"] = dataset_2019.loc[dataset_2019.nombre_destino ==␣
 ↪"fitz roy & gorriti", "long_estacion_destino"].iloc[0]
```

[89]:
```
%%capture [--no-stderr]
dataset_2018.rename(columns = {"id_estacion_origen" : "codigo_origen",␣
 ↪"id_estacion_destino" : "codigo_destino", "nombre_estacion_origen" :␣
 ↪"nombre_origen", "nombre_estacion_destino" : "nombre_destino"}, inplace =␣
 ↪True)
```

[90]:
```
dataset_2018.isnull().sum()
```

[90]:
```
año                     0
genero_usuario          0
codigo_origen           0
nombre_origen           0
long_estacion_origen    0
lat_estacion_origen     0
codigo_destino          0
nombre_destino          0
long_estacion_destino   0
lat_estacion_destino    0
id_usuario              0
minutos                 0
dtype: int64
```

- B) 2017

[91]:
```
%%capture [--no-stderr]
dataset_2017.loc[dataset_2017.nombre_estacion_origen.isnull() == False,␣
 ↪"nombre_estacion_origen"] = dataset_2017.loc[dataset_2017.
 ↪nombre_estacion_origen.isnull() == False, "nombre_estacion_origen"].
 ↪apply(lambda x : unidecode.unidecode(x))
dataset_2017.loc[dataset_2017.nombre_estacion_origen.isnull() == False,␣
 ↪"nombre_estacion_origen"] = dataset_2017.loc[dataset_2017.
 ↪nombre_estacion_origen.isnull() == False, "nombre_estacion_origen"].str.
 ↪lower()

dataset_2017.loc[dataset_2017.nombre_estacion_destino.isnull() == False,␣
 ↪"nombre_estacion_destino"] = dataset_2017.loc[dataset_2017.
 ↪nombre_estacion_destino.isnull() == False, "nombre_estacion_destino"].
 ↪apply(lambda x : unidecode.unidecode(x))
dataset_2017.loc[dataset_2017.nombre_estacion_destino.isnull() == False,␣
 ↪"nombre_estacion_destino"] = dataset_2017.loc[dataset_2017.
 ↪nombre_estacion_destino.isnull() == False, "nombre_estacion_destino"].str.
 ↪lower()
```

[92]:
```
dataset_2017.isnull().sum()
```

[92]:
```
año                          0
genero_usuario               0
id_estacion_origen        3419
nombre_estacion_origen       0
long_estacion_origen      3419
lat_estacion_origen       3419
id_estacion_destino       3496
nombre_estacion_destino      0
long_estacion_destino     3496
lat_estacion_destino      3496
id_usuario             1048158
minutos                      0
dtype: int64
```

[93]:
```
dataset_2017[dataset_2017.id_estacion_destino.isnull()].nombre_estacion_destino.
 ↪value_counts()
```

[93]:
```
f. j. santamaria de oro    1920
fitz roy y gorrtiti        1576
Name: nombre_estacion_destino, dtype: int64
```

[94]:
```
%%capture [--no-stderr]
# Se corrige el error de tipeo en "gorrtiti
dataset_2017.replace({"fitz roy y gorrtiti" : "fitz roy & gorriti"},␣
 ↪regex=True, inplace = True)
```

[95]:
```
%%capture [--no-stderr]
# Ecoparque:
dataset_2017.loc[dataset_2017.nombre_estacion_origen == "f. j. santamaria de␣
 ↪oro", "id_estacion_origen"] = dataset_2019.loc[dataset_2019.nombre_destino␣
 ↪== "f.j.santamaria de oro", "codigo_destino"].iloc[0]
dataset_2017.loc[dataset_2017.nombre_estacion_destino == "f. j. santamaria de␣
 ↪oro", "id_estacion_destino"] = dataset_2019.loc[dataset_2019.nombre_destino␣
 ↪== "f.j.santamaria de oro", "codigo_destino"].iloc[0]
dataset_2017.loc[dataset_2017.nombre_estacion_origen == "f. j. santamaria de␣
 ↪oro", "lat_estacion_origen"] = dataset_2019.loc[dataset_2019.nombre_destino␣
 ↪== "f.j.santamaria de oro", "lat_estacion_origen"].iloc[0]
dataset_2017.loc[dataset_2017.nombre_estacion_origen == "f. j. santamaria de␣
 ↪oro", "long_estacion_origen"] = dataset_2019.loc[dataset_2019.nombre_destino␣
 ↪== "f.j.santamaria de oro", "long_estacion_origen"].iloc[0]
dataset_2017.loc[dataset_2017.nombre_estacion_destino == "f. j. santamaria de␣
 ↪oro", "lat_estacion_destino"] = dataset_2019.loc[dataset_2019.nombre_destino␣
 ↪== "f.j.santamaria de oro", "lat_estacion_destino"].iloc[0]
dataset_2017.loc[dataset_2017.nombre_estacion_destino == "f. j. santamaria de␣
 ↪oro", "long_estacion_destino"] = dataset_2019.loc[dataset_2019.
 ↪nombre_destino == "f.j.santamaria de oro", "long_estacion_destino"].iloc[0]

# Fitz Roy y Gorriti:
dataset_2017.loc[dataset_2017.nombre_estacion_origen == "fitz roy & gorriti",␣
 ↪"id_estacion_origen"] = dataset_2019.loc[dataset_2019.nombre_destino ==␣
 ↪"fitz roy & gorriti", "codigo_destino"].iloc[0]
dataset_2017.loc[dataset_2017.nombre_estacion_destino == "fitz roy & gorriti",␣
 ↪"id_estacion_destino"] = dataset_2019.loc[dataset_2019.nombre_destino ==␣
 ↪"fitz roy & gorriti", "codigo_destino"].iloc[0]
dataset_2017.loc[dataset_2017.nombre_estacion_origen == "fitz roy & gorriti",␣
 ↪"lat_estacion_origen"] = dataset_2019.loc[dataset_2019.nombre_destino ==␣
 ↪"fitz roy & gorriti", "lat_estacion_destino"].iloc[0]
dataset_2017.loc[dataset_2017.nombre_estacion_origen == "fitz roy & gorriti",␣
 ↪"long_estacion_origen"] = dataset_2019.loc[dataset_2019.nombre_destino ==␣
 ↪"fitz roy & gorriti", "long_estacion_destino"].iloc[0]
dataset_2017.loc[dataset_2017.nombre_estacion_destino == "fitz roy & gorriti",␣
 ↪"lat_estacion_destino"] = dataset_2019.loc[dataset_2019.nombre_destino ==␣
 ↪"fitz roy & gorriti", "lat_estacion_destino"].iloc[0]
dataset_2017.loc[dataset_2017.nombre_estacion_destino == "fitz roy & gorriti",␣
 ↪"long_estacion_destino"] = dataset_2019.loc[dataset_2019.nombre_destino ==␣
 ↪"fitz roy & gorriti", "long_estacion_destino"].iloc[0]
```

[96]:
```
%%capture [--no-stderr]
dataset_2017.rename(columns = {"id_estacion_origen" : "codigo_origen",␣
 ↪"id_estacion_destino" : "codigo_destino", "nombre_estacion_origen" :␣
 ↪"nombre_origen", "nombre_estacion_destino" : "nombre_destino"}, inplace =␣
 ↪True)
```

[97]:
```
dataset_2017.isnull().sum()
```

[97]:
```
año                          0
genero_usuario               0
codigo_origen                0
nombre_origen                0
long_estacion_origen         0
lat_estacion_origen          0
codigo_destino               0
nombre_destino               0
long_estacion_destino        0
lat_estacion_destino         0
id_usuario             1048158
minutos                      0
dtype: int64
```

- C) 2016

[98]:
```
%%capture [--no-stderr]
dataset_2016.loc[dataset_2016.nombre_estacion_origen.isnull() == False,␣
 ↪"nombre_estacion_origen"] = dataset_2016.loc[dataset_2016.
 ↪nombre_estacion_origen.isnull() == False, "nombre_estacion_origen"].
 ↪apply(lambda x : unidecode.unidecode(x))
dataset_2016.loc[dataset_2016.nombre_estacion_origen.isnull() == False,␣
 ↪"nombre_estacion_origen"] = dataset_2016.loc[dataset_2016.
 ↪nombre_estacion_origen.isnull() == False, "nombre_estacion_origen"].str.
 ↪lower()

dataset_2016.loc[dataset_2016.nombre_estacion_destino.isnull() == False,␣
 ↪"nombre_estacion_destino"] = dataset_2016.loc[dataset_2016.
 ↪nombre_estacion_destino.isnull() == False, "nombre_estacion_destino"].
 ↪apply(lambda x : unidecode.unidecode(x))
dataset_2016.loc[dataset_2016.nombre_estacion_destino.isnull() == False,␣
 ↪"nombre_estacion_destino"] = dataset_2016.loc[dataset_2016.
 ↪nombre_estacion_destino.isnull() == False, "nombre_estacion_destino"].str.
 ↪lower()
```

[99]:
```
dataset_2016.isnull().sum()
```

[99]:
```
año                        0
genero_usuario             0
id_estacion_origen         0
nombre_estacion_origen     0
long_estacion_origen       0
lat_estacion_origen        0
id_estacion_destino        0
nombre_estacion_destino    0
long_estacion_destino      0
```

```
lat_estacion_destino            0
id_usuario                 596807
minutos                         0
dtype: int64
```

[100]:
```
%%capture [--no-stderr]
dataset_2016.rename(columns = {"id_estacion_origen" : "codigo_origen",␣
 ↪"id_estacion_destino" : "codigo_destino", "nombre_estacion_origen" :␣
 ↪"nombre_origen", "nombre_estacion_destino" : "nombre_destino"}, inplace =␣
 ↪True)
```

- D) 2015

[101]:
```
%%capture [--no-stderr]
dataset_2015.loc[dataset_2015.nombre_estacion_origen.isnull() == False,␣
 ↪"nombre_estacion_origen"] = dataset_2015.loc[dataset_2015.
 ↪nombre_estacion_origen.isnull() == False, "nombre_estacion_origen"].
 ↪apply(lambda x : unidecode.unidecode(x))
dataset_2015.loc[dataset_2015.nombre_estacion_origen.isnull() == False,␣
 ↪"nombre_estacion_origen"] = dataset_2015.loc[dataset_2015.
 ↪nombre_estacion_origen.isnull() == False, "nombre_estacion_origen"].str.
 ↪lower()

dataset_2015.loc[dataset_2015.nombre_estacion_destino.isnull() == False,␣
 ↪"nombre_estacion_destino"] = dataset_2015.loc[dataset_2015.
 ↪nombre_estacion_destino.isnull() == False, "nombre_estacion_destino"].
 ↪apply(lambda x : unidecode.unidecode(x))
dataset_2015.loc[dataset_2015.nombre_estacion_destino.isnull() == False,␣
 ↪"nombre_estacion_destino"] = dataset_2015.loc[dataset_2015.
 ↪nombre_estacion_destino.isnull() == False, "nombre_estacion_destino"].str.
 ↪lower()
```

[102]:
```
dataset_2015.isnull().sum()
```

[102]:
```
año                            0
genero_usuario                 0
id_estacion_origen             0
nombre_estacion_origen         0
long_estacion_origen        3168
lat_estacion_origen         3168
id_estacion_destino            0
nombre_estacion_destino        0
long_estacion_destino       3462
lat_estacion_destino        3462
id_usuario                503252
minutos                        0
dtype: int64
```

[103]:
```
dataset_2015[dataset_2015.lat_estacion_destino.isnull()].
 ↪nombre_estacion_destino.value_counts()
```

[103]:
```
santiago del estero    2288
cochabamba             1174
Name: nombre_estacion_destino, dtype: int64
```

[104]:
```
estaciones_nuevo.loc[estaciones_nuevo.nombre.str.contains("stero"), :]
estaciones_viejo.loc[estaciones_viejo.nombre_estacion.str.contains("stero"), :]

estaciones_nuevo.loc[estaciones_nuevo.nombre.str.contains("chaba"), :]
estaciones_viejo.loc[estaciones_viejo.nombre_estacion.str.contains("chaba"), :]
```

[104]:
```
    id_estacion nombre_estacion  long_estacion  lat_estacion  \
14         15.0      Cochabamba            NaN           NaN

   domicilio_estacion tipo_estacion observaciones horario_estacion
14                NaN           NaN           NaN               NaN
```

[105]:
```
%%capture [--no-stderr]
# Como las dos estaciones con nulos no se encuentran en los otros datasets o en␣
 ↪el de estaciones del sistema nuevo y el dataset del sistema viejo contiene␣
 ↪ambos pero sin la información de latitud y longitud, se procederá a␣
 ↪eliminarlos.
dataset_2015.dropna(subset = ["long_estacion_origen", "lat_estacion_origen",␣
 ↪"long_estacion_destino", "lat_estacion_destino"], inplace = True)
```

[106]:
```
%%capture [--no-stderr]
dataset_2015.rename(columns = {"id_estacion_origen" : "codigo_origen",␣
 ↪"id_estacion_destino" : "codigo_destino", "nombre_estacion_origen" :␣
 ↪"nombre_origen", "nombre_estacion_destino" : "nombre_destino"}, inplace =␣
 ↪True)
```

[107]:
```
dataset_2015.isnull().sum()
```

[107]:
```
año                        0
genero_usuario             0
codigo_origen              0
nombre_origen              0
long_estacion_origen       0
lat_estacion_origen        0
codigo_destino             0
nombre_destino             0
long_estacion_destino      0
lat_estacion_destino       0
id_usuario            496973
minutos                    0
```

```
dtype: int64
```

4 3) Unificación, limpieza final y exportación del dataset

4.0.1 A) Concatenación parcial + imputación de longitudes y latitudes de 2021:

[108]:
```
# No se concatena el año 2021 pues se le tiene que imputar las latitudes y longitudes
dataset = pd.concat([dataset_2015, dataset_2016, dataset_2017, dataset_2018, dataset_2019, dataset_2020])
dataset
```

[108]:

	año	genero_usuario	codigo_origen	nombre_origen	\
0	2015	M	25	plaza guemes	
1	2015	F	17	plaza almagro	
2	2015	M	17	plaza almagro	
3	2015	M	29	parque centenario	
4	2015	M	29	parque centenario	
…	…	…	…	…	
2002940	2020	M	151	aime paine	
2002941	2020	M	150	rodrigo bueno	
2002942	2020	F	150	rodrigo bueno	
2002943	2020	M	114	della paolera	
2002944	2020	F	237	madero office	

	long_estacion_origen	lat_estacion_origen	codigo_destino	\
0	-58.416065	-34.589521	29.0	
1	-58.418832	-34.606399	25.0	
2	-58.418832	-34.606399	25.0	
3	-58.434577	-34.608459	25.0	
4	-58.434577	-34.608459	25.0	
…	…	…	…	
2002940	-58.361280	-34.611816	179.0	
2002941	-58.355744	-34.618841	179.0	
2002942	-58.355744	-34.618841	179.0	
2002943	-58.372251	-34.594975	179.0	
2002944	-58.364690	-34.599037	179.0	

	nombre_destino	long_estacion_destino	lat_estacion_destino	\
0	parque centenario	-58.434577	-34.608459	
1	plaza guemes	-58.416065	-34.589521	
2	plaza guemes	-58.416065	-34.589521	
3	plaza guemes	-58.416065	-34.589521	
4	plaza guemes	-58.416065	-34.589521	
…	…	…	…	
2002940	casa san	-58.364284	-34.638480	
2002941	casa san	-58.364284	-34.638480	

```
2002942           casa san             -58.364284            -34.638480
2002943           casa san             -58.364284            -34.638480
2002944           casa san             -58.364284            -34.638480

         id_usuario minutos
0               NaN      26
1               NaN      57
2               NaN      03
3               NaN      04
4               NaN      09
…               …      …
2002940    634757.0      36
2002941    641709.0      27
2002942    641712.0      43
2002943    412605.0      82
2002944    630939.0     177

[13002211 rows x 12 columns]
```

[109]:
```
%%capture [--no-stderr]
# Se crea un df con latitudes y longitudes por código de los años 2015 a 2020

dataframe_lat_long_origen = dataset.loc[dataset.año != "2021",["codigo_origen",␣
 ↪"lat_estacion_origen","long_estacion_origen"]]
dataframe_lat_long_origen.drop_duplicates(subset = ["codigo_origen"], inplace =␣
 ↪True)
dataframe_lat_long_origen.rename(columns = {"lat_estacion_origen" :␣
 ↪"lat_estacion_origen_merge", "long_estacion_origen" :␣
 ↪"long_estacion_origen_merge"}, inplace = True)
dataframe_lat_long_origen.dropna(subset = ["lat_estacion_origen_merge",␣
 ↪"long_estacion_origen_merge"], inplace = True)

dataframe_lat_long_destino = dataset.loc[dataset.año !=␣
 ↪"2021",["codigo_destino", "lat_estacion_destino","long_estacion_destino"]]
dataframe_lat_long_destino.drop_duplicates(subset = ["codigo_destino"], inplace␣
 ↪= True)
dataframe_lat_long_destino.rename(columns = {"lat_estacion_destino" :␣
 ↪"lat_estacion_destino_merge", "long_estacion_destino" :␣
 ↪"long_estacion_destino_merge"}, inplace = True)
dataframe_lat_long_destino.dropna(subset = ["lat_estacion_destino_merge",␣
 ↪"long_estacion_destino_merge"], inplace = True)
```

[110]:
```
dataframe_lat_long_destino
```

[110]:
```
       codigo_destino  lat_estacion_destino_merge  long_estacion_destino_merge
0                29.0                   -34.608459                   -58.434577
1                25.0                   -34.589521                   -58.416065
```

```
6                30.0                -34.590394                -58.397378
8                18.0                -34.617373                -58.380481
9                 7.0                -34.605840                -58.380990
…                 …                        …                         …
21391          362.0                -34.543005                -58.436117
21542          213.0                -34.597210                -58.474211
24615          387.0                -34.638584                -58.399683
31013          277.0                -34.565643                -58.475512
38670          228.0                -34.635505                -58.456035

[397 rows x 3 columns]
```

[111]:
```
# Se hace un merge para adjudicar a cada estación de origen/destino su latitud
 ↪y longitud según su código

dataset_2021 = dataset_2021.merge(dataframe_lat_long_origen,
 ↪on='codigo_origen', how='left')
dataset_2021 = dataset_2021.merge(dataframe_lat_long_destino,
 ↪on='codigo_destino', how='left')
dataset_2021
```

[111]:
```
         minutos  id_usuario  long_estacion_origen  lat_estacion_origen  \
0              7      701665                   NaN                  NaN
1              7      752374                   NaN                  NaN
2              7      425502                   NaN                  NaN
3              1        4519                   NaN                  NaN
4              8        8197                   NaN                  NaN
…            …           …                     …                    …
1144180       12       62246                   NaN                  NaN
1144181       23      445201                   NaN                  NaN
1144182       19      554162                   NaN                  NaN
1144183       17       51005                   NaN                  NaN
1144184       30      734428                   NaN                  NaN

         long_estacion_destino  lat_estacion_destino   año genero_usuario  \
0                          NaN                   NaN  2021              M
1                          NaN                   NaN  2021              F
2                          NaN                   NaN  2021              F
3                          NaN                   NaN  2021              M
4                          NaN                   NaN  2021              M
…                          …                     …    …            …
1144180                    NaN                   NaN  2021              F
1144181                    NaN                   NaN  2021              M
1144182                    NaN                   NaN  2021              F
1144183                    NaN                   NaN  2021              F
1144184                    NaN                   NaN  2021              M
```

```
        codigo_origen           nombre_origen  codigo_destino  \
0                 240              echeverria             255
1                 275  plaza 24 de septiembre             223
2                 282                tronador             313
3                 158              villarroel             158
4                  29       parque centenario              99
…                 …                       …              …
1144180           292           plaza bolivia              44
1144181            79       azucena villaflor             168
1144182            79       azucena villaflor               8
1144183            79       azucena villaflor              75
1144184            79       azucena villaflor             123

                nombre_destino  lat_estacion_origen_merge  \
0         barrancas de belgrano                 -34.565275
1                       gainza                 -34.615064
2                de los incas                  -34.583990
3                  villarroel                  -34.592839
4                     malabia                  -34.608459
…                         …                        …
1144180             ecoparque                  -34.563465
1144181        estados unidos                  -34.611721
1144182              congreso                  -34.611721
1144183   plaza primero de mayo                -34.611721
1144184               armenia                  -34.611721

         long_estacion_origen_merge  lat_estacion_destino_merge  \
0                        -58.459401                  -34.559793
1                        -58.428781                  -34.616680
2                        -58.466630                  -34.579751
3                        -58.445099                  -34.592839
4                        -58.434577                  -34.596281
…                               …                           …
1144180                  -58.436037                  -34.575485
1144181                  -58.363969                  -34.618620
1144182                  -58.363969                  -34.609930
1144183                  -58.363969                  -34.612124
1144184                  -58.363969                  -34.585424

         long_estacion_destino_merge
0                          -58.448432
1                          -58.446667
2                          -58.471003
3                          -58.445099
4                          -58.435615
…                                 …
1144180                    -58.414595
```

```
1144181                   -58.381283
1144182                   -58.389253
1144183                   -58.398905
1144184                   -58.421044

[1144185 rows x 16 columns]
```

[112]:
```
# Se hace un drop de las columnas con nulos y se reemplazan los nombres de las␣
 ↪columnas nuevas

dataset_2021.drop(["lat_estacion_origen", "long_estacion_origen",␣
 ↪"lat_estacion_destino", "long_estacion_destino"], axis = 1, inplace = True)
dataset_2021.rename(columns = {"lat_estacion_origen_merge" :␣
 ↪"lat_estacion_origen", "long_estacion_origen_merge" :␣
 ↪"long_estacion_origen", "lat_estacion_destino_merge" :␣
 ↪"lat_estacion_destino", "long_estacion_destino_merge" :␣
 ↪"long_estacion_destino"}, inplace = True)
print("\nQuedaron:", dataset_2021.lat_estacion_origen.isnull().sum(), "nulos␣
 ↪restantes\nLa imputación anterior completó correctamente el",
(round(((len(dataset_2021)-(dataset_2021.lat_estacion_origen.isnull().
 ↪sum()+dataset_2021.long_estacion_origen.isnull().sum()))/len(dataset_2021)),␣
 ↪3)*100), "% de los datos faltantes\n")
```

```
Quedaron: 560 nulos restantes
La imputación anterior completó correctamente el 99.9 % de los datos faltantes
```

[113]:
```
dataset_2021.dropna(subset = ["lat_estacion_origen", "long_estacion_origen",␣
 ↪"lat_estacion_destino", "long_estacion_destino"], inplace = True)
```

4.0.2 B) Concatenación final y corrección de algunos textos:

[114]:
```
dataset = pd.concat([dataset, dataset_2021])
dataset
```

[114]:

	año	genero_usuario	codigo_origen	nombre_origen
0	2015	M	25	plaza guemes
1	2015	F	17	plaza almagro
2	2015	M	17	plaza almagro
3	2015	M	29	parque centenario
4	2015	M	29	parque centenario
…	…	…	…	…
1144180	2021	F	292	plaza bolivia
1144181	2021	M	79	azucena villaflor
1144182	2021	F	79	azucena villaflor
1144183	2021	F	79	azucena villaflor

```
1144184  2021              M            79  azucena villaflor

         long_estacion_origen  lat_estacion_origen  codigo_destino  \
0                   -58.416065           -34.589521            29.0
1                   -58.418832           -34.606399            25.0
2                   -58.418832           -34.606399            25.0
3                   -58.434577           -34.608459            25.0
4                   -58.434577           -34.608459            25.0
…                          …                    …             …
1144180             -58.436037           -34.563465            44.0
1144181             -58.363969           -34.611721           168.0
1144182             -58.363969           -34.611721             8.0
1144183             -58.363969           -34.611721            75.0
1144184             -58.363969           -34.611721           123.0

                 nombre_destino  long_estacion_destino  lat_estacion_destino  \
0             parque centenario             -58.434577            -34.608459
1                  plaza guemes             -58.416065            -34.589521
2                  plaza guemes             -58.416065            -34.589521
3                  plaza guemes             -58.416065            -34.589521
4                  plaza guemes             -58.416065            -34.589521
…                           …                    …                     …
1144180               ecoparque             -58.414595            -34.575485
1144181          estados unidos             -58.381283            -34.618620
1144182                congreso             -58.389253            -34.609930
1144183   plaza primero de mayo             -58.398905            -34.612124
1144184                 armenia             -58.421044            -34.585424

         id_usuario minutos
0               NaN      26
1               NaN      57
2               NaN      03
3               NaN      04
4               NaN      09
…               …     …
1144180     62246.0      12
1144181    445201.0      23
1144182    554162.0      19
1144183     51005.0      17
1144184    734428.0      30

[14145371 rows x 12 columns]
```

[115]:
```
dataset.dtypes
```

[115]:
```
año                        int64
genero_usuario            object
```

```
codigo_origen              object
nombre_origen              object
long_estacion_origen      float64
lat_estacion_origen       float64
codigo_destino            float64
nombre_destino             object
long_estacion_destino     float64
lat_estacion_destino      float64
id_usuario                float64
minutos                    object
dtype: object
```

[116]:
```
%%capture [--no-stderr]
dataset.codigo_origen = dataset.codigo_origen.astype("int")
dataset.codigo_destino = dataset.codigo_destino.astype("int")
dataset.minutos = dataset.minutos.astype("int")
dataset.loc[dataset.id_usuario.isnull() == False, "id_usuario"] = dataset.
 ↪loc[dataset.id_usuario.isnull() == False, "id_usuario"].astype("int")
```

[117]:
```
%%capture [--no-stderr]
# Se corrige el espaciado en "f.j.santamaria" y se unifica "&" con "y"
dataset.replace({"f.j.santamaria de oro" : "f. j. santamaria de oro", " & " : "␣
 ↪y "}, regex=True, inplace = True)
```

[118]:
```
dataset.isnull().sum()
```

[118]:
```
año                          0
genero_usuario               0
codigo_origen                0
nombre_origen                0
long_estacion_origen         0
lat_estacion_origen          0
codigo_destino               0
nombre_destino               0
long_estacion_destino        0
lat_estacion_destino         0
id_usuario             2141938
minutos                      0
dtype: int64
```

Lamentablemente los id_usuario nulos son imposibles de imputar, pero se deja esa columna pues se puede utilizar los años que tienen ese dato (los cuales no presentan nulos)

4.0.3 C) Eliminación de outliers:

La duración de los recorridos presenta algunos números erróneos:

[119]:
```
print((dataset["minutos"]).max())
```

```
53168
```

[120]:
```
q75,q25 = np.percentile(dataset["minutos"],[75,25])
iqr = q75-q25
max_limit = q75+(1.5*iqr)
min_limit = q25-(1.5*iqr)
print("límite superior:", max_limit, "\nlímite inferior:", min_limit)
```

```
límite superior: 55.0
límite inferior: -17.0
```

[121]:
```
# Como no se desea quitar recorridos dentro de los 60 minutos permitidos, se␣
 ↪definirá el límite superior como 60 minutos.
# Como carece de sentido utilizar un límite inferior negativo, se le asignará␣
 ↪el valor de 0 minutos
max_limit = 60
min_limit = 0
```

[122]:
```
cant_valores = len(dataset)
dataset = dataset.loc[(dataset["minutos"] < max_limit) & (dataset["minutos"]  >␣
 ↪min_limit),:]

print("Se borraron", cant_valores - len(dataset), "outliers, quedando un total␣
 ↪de", len(dataset), "valores válidos entre 0 y 60 minutos")
```

```
Se borraron 176121 outliers, quedando un total de 13969250 valores válidos entre
0 y 60 minutos
```

4.0.4 D) Exportación de dataset:

[123]:
```
dataset.to_csv("dataset.csv", index = False)
```