

Parcial Visualización de Datos

Luca Mazzarello, Ignacio Pardo

April 25, 2023

1 Procesamiento de datos del SUACI

1.1 Importación de librerías

```
[ ]: import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

1.2 Carga de datos

```
[ ]: data_2022 = pd.read_csv('suaci/sistema-unico-de-atencion-ciudadana-2022.csv',
    ↪sep=';')
data_2021 = pd.read_csv('suaci/sistema-unico-de-atencion-ciudadana-2021.csv',
    ↪sep=';')
data_2020 = pd.read_csv('suaci/sistema-unico-de-atencion-ciudadana-2020.csv',
    ↪sep=',')
data_2019 = pd.read_csv('suaci/sistema-unico-de-atencion-ciudadana-2019.csv',
    ↪sep=',')
data_2018 = pd.read_csv('suaci/sistema-unico-de-atencion-ciudadana-2018.csv',
    ↪sep=',')
```

```
[ ]: datas = [data_2022, data_2021, data_2020, data_2019, data_2018]
sizes = [len(d) for d in datas]
sizes
```

```
[ ]: [104912, 754638, 574435, 989629, 893291]
```

1.3 Preprocesamiento

Conteo de denuncias por barrio para el año 2022.

```
[ ]: # Count denuncias por barrio
barrio_counts_2022 = data_2022['domicilio_barrio'].value_counts()
```

```
[ ]: bcdf = pd.DataFrame(barrio_counts_2022)

#Change keys to domicilio_barrio,count
```

```
# make index a column

bcd = bcd.reset_index()

# rename columns

bcd.columns = ['domicilio_barrio', 'count']
```

```
[ ]: bcd.to_csv('barrio_counts_2022.csv', index=False)
```

1.3.1 Limpieza de datos de mosquitos

```
[ ]: mosquitos_2022 = data_2022[data_2022['subcategoria'] == 'ENFERMEDADES_
↳TRANSMITIDAS POR MOSQUITOS']
barrios_ms_2022 = mosquitos_2022['domicilio_barrio'].value_counts()[0:10][::-1]
barrios_ms_2022 = mosquitos_2022['domicilio_barrio'].value_counts()[0:10][::-1]
```

```
[ ]: mosquitos_2021 = data_2021[data_2021['subcategoria'] == 'ENFERMEDADES_
↳TRANSMITIDAS POR MOSQUITOS']
barrios_ms_2021 = mosquitos_2021['domicilio_barrio'].value_counts()[0:10][::-1]
barrios_ms_2021 = mosquitos_2021['domicilio_barrio'].value_counts()[0:10][::-1]
```

```
[ ]: mosquitos_2020 = data_2020[data_2020['subcategoria'] == 'ENFERMEDADES_
↳TRANSMITIDAS POR MOSQUITOS']
barrios_ms_2020 = mosquitos_2020['domicilio_barrio'].value_counts()[0:10][::-1]
barrios_ms_2020 = mosquitos_2020['domicilio_barrio'].value_counts()[0:10][::-1]
```

```
[ ]: mosquitos_2019 = data_2019[data_2019['subcategoria'] == 'ENFERMEDADES_
↳TRANSMITIDAS POR MOSQUITOS']
barrios_ms_2019 = mosquitos_2019['domicilio_barrio'].value_counts()[0:10][::-1]
barrios_ms_2019 = mosquitos_2019['domicilio_barrio'].value_counts()[0:10][::-1]
```

```
[ ]: mosquitos_2019 = data_2019[data_2019['subcategoria'] == 'ENFERMEDADES_
↳TRANSMITIDAS POR MOSQUITOS']
barrios_ms_2019 = mosquitos_2019['domicilio_barrio'].value_counts()[0:10][::-1]
barrios_ms_2019 = mosquitos_2019['domicilio_barrio'].value_counts()[0:10][::-1]
```

```
[ ]: mosquitos_2018 = data_2018[data_2018['subcategoria'] == 'ENFERMEDADES_
↳TRANSMITIDAS POR MOSQUITOS']
barrios_ms_2018 = mosquitos_2018['domicilio_barrio'].value_counts()[0:10][::-1]
barrios_ms_2018 = mosquitos_2018['domicilio_barrio'].value_counts()[0:10][::-1]
```

```
[ ]: import functools as ft
```

```
[ ]: mosquitos = {
    "2022": mosquitos_2022,
    "2021": mosquitos_2021,
```

```

    "2020": mosquitos_2020,
    "2019": mosquitos_2019,
    "2018": mosquitos_2018
}

```

```
[ ]: keys = [list(v.keys().to_series()) for k, v in mosquitos.items()]

```

```

for key in zip(keys[0], keys[1], keys[2], keys[3], keys[4]):
    print(key)

```

```

('contacto', 'contacto', 'contacto', 'contacto', 'contacto')
('periodo', 'periodo', 'periodo', 'periodo', 'periodo')
('categoria', 'categoria', 'categoria', 'categoria', 'categoria')
('subcategoria', 'subcategoria', 'subcategoria', 'subcategoria', 'subcategoria')
('prestacion', 'prestacion', 'concepto', 'concepto', 'concepto')
('tipo_prestacion', 'tipo_prestacion', 'tipo_prestacion', 'tipo_prestacion',
'tipo_prestacion')
('fecha_ingreso', 'fecha_ingreso', 'fecha_ingreso', 'fecha_ingreso',
'fecha_ingreso')
('hora_ingreso', 'hora_ingreso', 'hora_ingreso', 'hora_ingreso', 'hora_ingreso')
('domicilio_comuna', 'domicilio_comuna', 'domicilio_cgpc', 'domicilio_cgpc',
'domicilio_cgpc')
('domicilio_barrio', 'domicilio_barrio', 'domicilio_barrio', 'domicilio_barrio',
'domicilio_barrio')
('domicilio_calle', 'domicilio_calle', 'domicilio_calle', 'domicilio_calle',
'domicilio_calle')
('domicilio_altura', 'domicilio_altura', 'domicilio_altura', 'domicilio_altura',
'domicilio_altura')
('domicilio_esquina_proxima', 'domicilio_esquina_proxima',
'domicilio_esquina_proxima', 'domicilio_esquina_proxima',
'domicilio_esquina_proxima')
('lat', 'lat', 'lat', 'lat', 'lat')
('lon', 'lon', 'long', 'long', 'long')
('canal', 'canal', 'canal', 'canal', 'canal')
('genero', 'genero', 'genero', 'genero', 'genero')
('estado_del_contacto', 'estado_del_contacto', 'estado_del_contacto',
'estado_del_contacto', 'estado_del_contacto')
('fecha_cierre_contacto', 'fecha_cierre_contacto', 'fecha_cierre_contacto',
'fecha_cierre_contacto', 'fecha_cierre_contacto')

```

```
[ ]: for k, v in mosquitos.items():
    try:
        v.rename(columns={'long': 'lon'}, inplace=True)
        v.rename(columns={'concepto': 'prestacion'}, inplace=True)
    except:
        pass

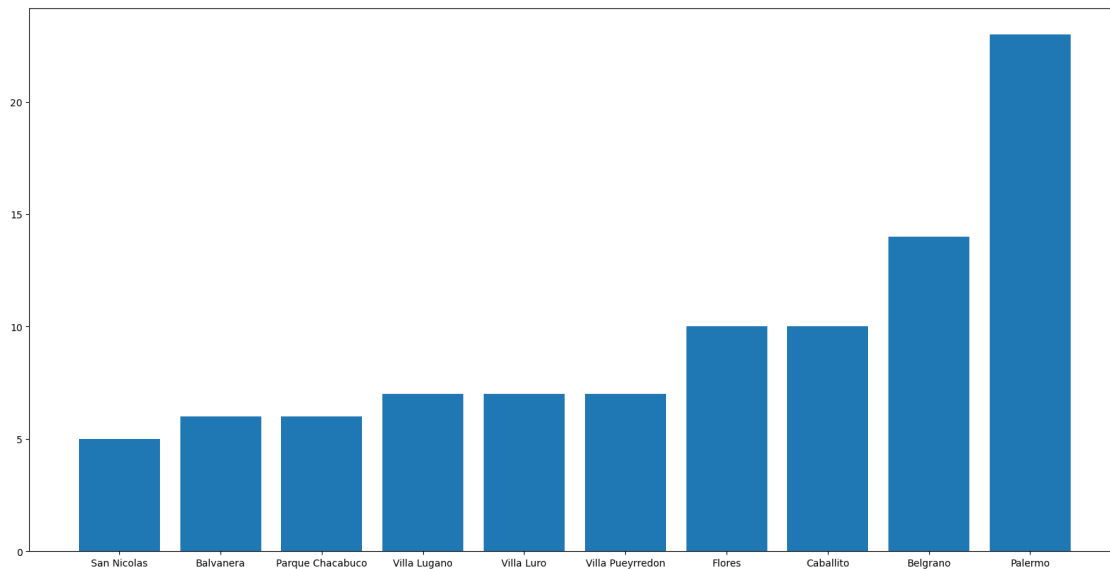
```

1.4 Exportación de datos

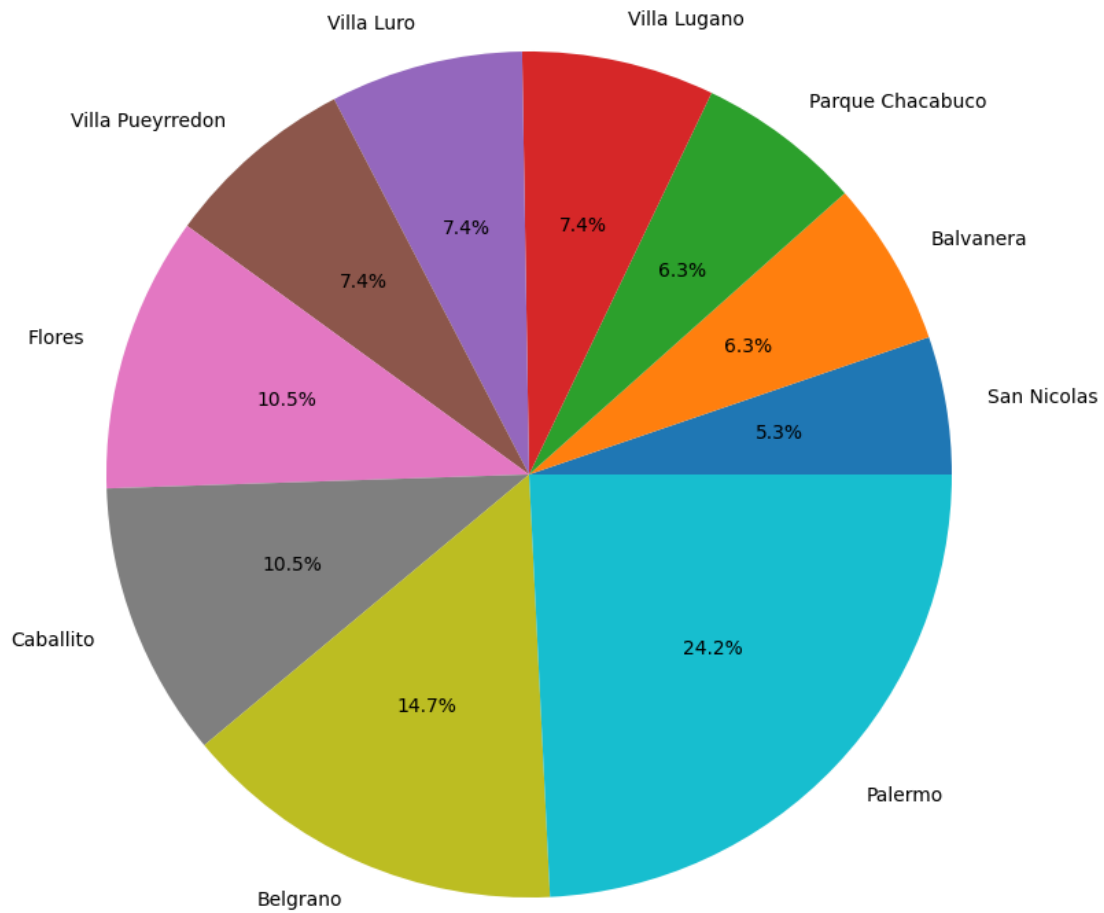
```
[ ]: for k, v in mosquitos.items():  
      v.to_csv(f"mosquitos-{k}.csv", index=False, sep=";")
```

1.5 Análisis exploratorio

```
[ ]: _ = plt.figure(figsize=(20,10))  
      _ = plt.bar(barrios_ms_2022.index, barrios_ms_2022.values)
```



```
[ ]: _ = plt.figure(figsize=(20,10))  
      _ = plt.pie(barrios_ms_2022.values, labels=barrios_ms_2022.index, autopct='%1.  
      ↪1f%%')
```



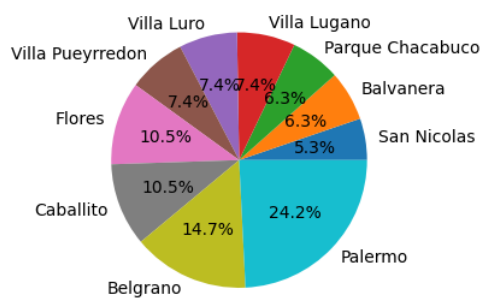
```
[ ]: # Plot all years in one fig, with a subplot for each year

fig, axs = plt.subplots(5, 1, figsize=(20, 20))

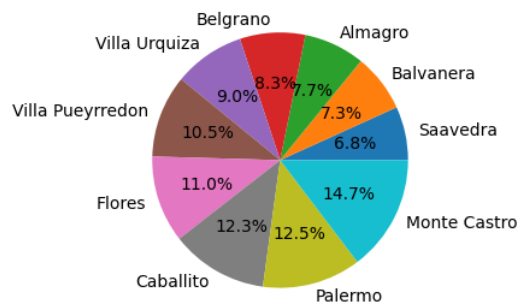
for i, (k, v) in enumerate(mosquitos.items()):
    axs[i].pie(v['domicilio_barrio'].value_counts()[0:10][::-1].values,
    labels=v['domicilio_barrio'].value_counts()[0:10][::-1].index, autopct='%1.1f%%')
    axs[i].set_title(k)

plt.show()
```

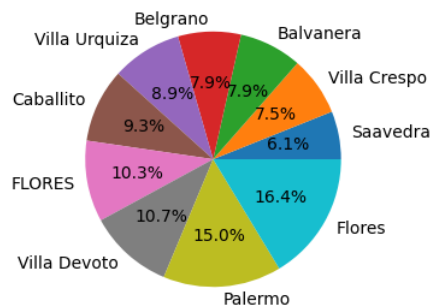
2022



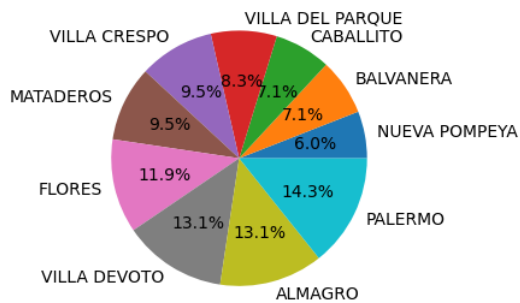
2021



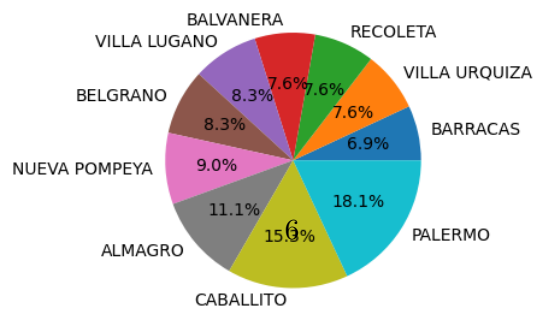
2020



2019



2018



```
[ ]: len(mosquitos_2021)
```

```
[ ]: 1060
```

1.6 Análisis de cantidad de denuncias totales

```
[ ]: mosquitos_total = pd.concat([mosquitos_2022, mosquitos_2021, mosquitos_2020,
    ↪ mosquitos_2019, mosquitos_2018])
```

```
[ ]: mosquitos_total["domicilio_barrio"] = mosquitos_total["domicilio_barrio"].str.
    ↪ upper()
```

```
[ ]: mes_f = lambda x: x % 100
    anio_f = lambda x: x // 100
```

```
[ ]: #mosquitos_total_palermo = mosquitos_total[mosquitos_total["domicilio_barrio"]
    ↪ == "PALERMO"]
mosquitos_total["mes"] = mosquitos_total["periodo"].apply(mes_f)
mosquitos_total["anio"] = mosquitos_total["periodo"].apply(anio_f)
mosquitos_total
```

```
[ ]:
      contacto  periodo  categoria \
1785  00002269/22  202201  SALUD Y SERVICIOS SOCIALES
4356  00006028/22  202201  SALUD Y SERVICIOS SOCIALES
4848  00006269/22  202201  SALUD Y SERVICIOS SOCIALES
4871  00005409/22  202201  SALUD Y SERVICIOS SOCIALES
5365  00005196/22  202201  SALUD Y SERVICIOS SOCIALES
...
883066  01606962/18  201812  SALUD Y SERVICIOS SOCIALES
884487  01609557/18  201812  SALUD Y SERVICIOS SOCIALES
885678  01611514/18  201812  SALUD Y SERVICIOS SOCIALES
887110  01613995/18  201812  SALUD Y SERVICIOS SOCIALES
890825  01620381/18  201812  SALUD Y SERVICIOS SOCIALES
```

```

                                subcategoria \
1785  ENFERMEDADES TRANSMITIDAS POR MOSQUITOS
4356  ENFERMEDADES TRANSMITIDAS POR MOSQUITOS
4848  ENFERMEDADES TRANSMITIDAS POR MOSQUITOS
4871  ENFERMEDADES TRANSMITIDAS POR MOSQUITOS
5365  ENFERMEDADES TRANSMITIDAS POR MOSQUITOS
...
883066  ENFERMEDADES TRANSMITIDAS POR MOSQUITOS
884487  ENFERMEDADES TRANSMITIDAS POR MOSQUITOS
885678  ENFERMEDADES TRANSMITIDAS POR MOSQUITOS
887110  ENFERMEDADES TRANSMITIDAS POR MOSQUITOS
```

890825 ENFERMEDADES TRANSMITIDAS POR MOSQUITOS

		prestacion	tipo_prestacion	\
1785		Criaderos de mosquitos	Reporte	
4356		Criaderos de mosquitos	Reporte	
4848		Criaderos de mosquitos	Reporte	
4871	Información sobre enfermedades transmitidas po...		Solicitud	
5365		Criaderos de mosquitos	Reporte	
...		
883066		CRIADEROS DE MOSQUITOS	SOLICITUD	
884487		CRIADEROS DE MOSQUITOS	SOLICITUD	
885678		CRIADEROS DE MOSQUITOS	SOLICITUD	
887110		CRIADEROS DE MOSQUITOS	SOLICITUD	
890825		CRIADEROS DE MOSQUITOS	SOLICITUD	

	fecha_ingreso	hora_ingreso	domicilio_comuna	domicilio_barrio	...	\
1785	2022-01-04	08:59:23	3.0	SAN CRISTOBAL	...	
4356	2022-01-06	18:03:11	6.0	CABALLITO	...	
4848	2022-01-06	20:49:44	14.0	PALERMO	...	
4871	2022-01-06	12:25:45	1.0	SAN NICOLAS	...	
5365	2022-01-06	10:43:06	10.0	FLORESTA	...	
...	
883066	2018-12-26	05:45:19 p.m.	NaN	PALERMO	...	
884487	2018-12-27	10:03:58 a.m.	NaN	RECOLETA	...	
885678	2018-12-27	01:49:37 p.m.	NaN	PARQUE AVELLANEDA	...	
887110	2018-12-27	09:22:15 p.m.	NaN	BOCA	...	
890825	2018-12-29	01:34:20 p.m.	NaN	PALERMO	...	

	lat	lon	canal	genero	estado_del_contacto	\
1785	-34.627571	-58.395442	Boti	Masculino	Abierto	
4356	-34.616716	-58.454184	GCS Web	Femenino	Abierto	
4848	-34.569974	-58.420226	GCS Web	Masculino	Abierto	
4871	-34.600055	-58.378397	App BA 147	Femenino	Abierto	
5365	-34.631738	-58.479428	App BA 147	Masculino	Abierto	
...	
883066	-34.59148	-58.414135	147	femenino	Abierto	
884487	-34.593322	-58.400479	App	masculino	Abierto	
885678	-34.640531	-58.478674	147	femenino	Abierto	
887110	-34.639052	-58.366368	147	femenino	Abierto	
890825	-34.579045	-58.405114	App	masculino	Abierto	

	fecha_cierre_contacto	domiclio_calle	domicilio_cgpc	mes	anio
1785	NaN	NaN	NaN	1	2022
4356	NaN	NaN	NaN	1	2022
4848	NaN	NaN	NaN	1	2022
4871	NaN	NaN	NaN	1	2022
5365	NaN	NaN	NaN	1	2022

...
883066	NaN	NaN	COMUNA 14	12	2018	
884487	NaN	NaN	COMUNA 2	12	2018	
885678	NaN	NaN	COMUNA 9	12	2018	
887110	NaN	NaN	COMUNA 4	12	2018	
890825	NaN	NaN	COMUNA 14	12	2018	

[2255 rows x 23 columns]

```
[ ]: mosquitos_total["fecha"] = pd.to_datetime(mosquitos_total["anio"].astype(str) +
↪ "-" + mosquitos_total["mes"].astype(str) + "-01")
```

```
[ ]: mosquitos_total["fecha_ingreso"] = pd.
↪ to_datetime(mosquitos_total["fecha_ingreso"], infer_datetime_format=True)
```

```
[ ]: # Scatter plot of occurring dates, in the x axis is the date, in the y axis is
↪ the number of denuncias
# The column "fecha_ingreso" is the date when the denuncia was made

data_key = "fecha"

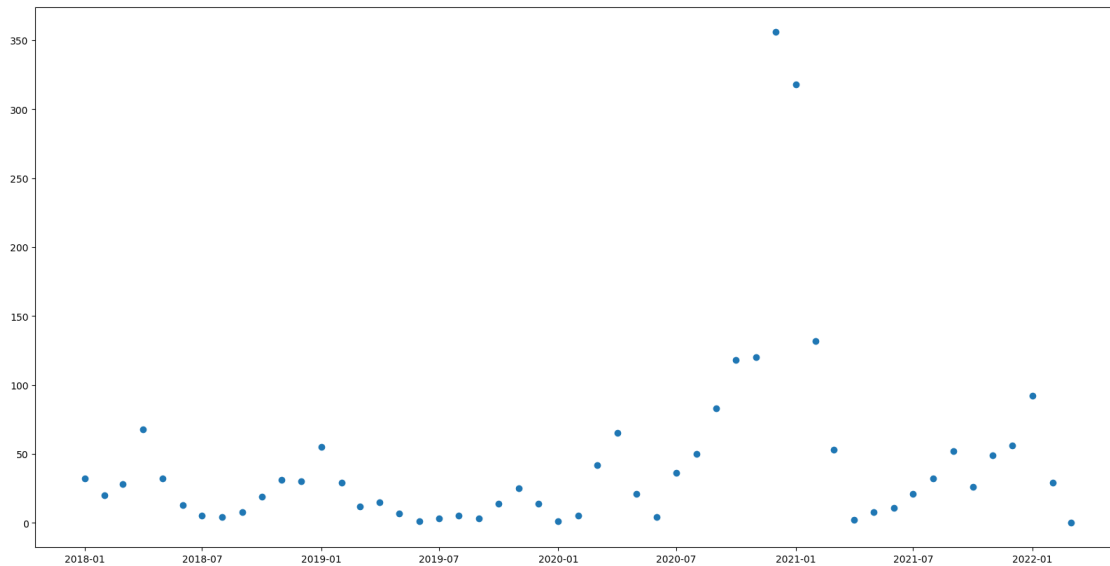
data_x = mosquitos_total[data_key].value_counts().sort_index().index
data_y = mosquitos_total[data_key].value_counts().sort_index().values

# fill data x with missing dates

for d in pd.date_range(start=data_x.min(), end=data_x.max(), freq='MS'):
    if d not in data_x:
        data_x = data_x.append(pd.Index([d]))
        data_y = np.append(data_y, 0)

data_x = data_x.sort_values()
data_y = data_y[data_x.argsort()]

_ = plt.figure(figsize=(20,10))
_ = plt.scatter(data_x, data_y)
```



```
[ ]: data_x, len(data_x)
```

```
[ ]: (DatetimeIndex(['2018-01-01', '2018-02-01', '2018-03-01', '2018-04-01',
                    '2018-05-01', '2018-06-01', '2018-07-01', '2018-08-01',
                    '2018-09-01', '2018-10-01', '2018-11-01', '2018-12-01',
                    '2019-01-01', '2019-02-01', '2019-03-01', '2019-04-01',
                    '2019-05-01', '2019-06-01', '2019-07-01', '2019-08-01',
                    '2019-09-01', '2019-10-01', '2019-11-01', '2019-12-01',
                    '2020-01-01', '2020-02-01', '2020-03-01', '2020-04-01',
                    '2020-05-01', '2020-06-01', '2020-07-01', '2020-08-01',
                    '2020-09-01', '2020-10-01', '2020-11-01', '2020-12-01',
                    '2021-01-01', '2021-02-01', '2021-03-01', '2021-04-01',
                    '2021-05-01', '2021-06-01', '2021-07-01', '2021-08-01',
                    '2021-09-01', '2021-10-01', '2021-11-01', '2021-12-01',
                    '2022-01-01', '2022-02-01', '2022-03-01'],
                  dtype='datetime64[ns]', freq=None),
```

51)

1.7 Proyección de denuncias por barrio para los próximos años

```
[ ]: # Quadratic regression with numpy

# ignore every month with 0 denuncias

#data_x = data_x[data_y != 0]
#data_y = data_y[data_y != 0]

x = np.array(range(len(data_x)))
```

```

y = data_y

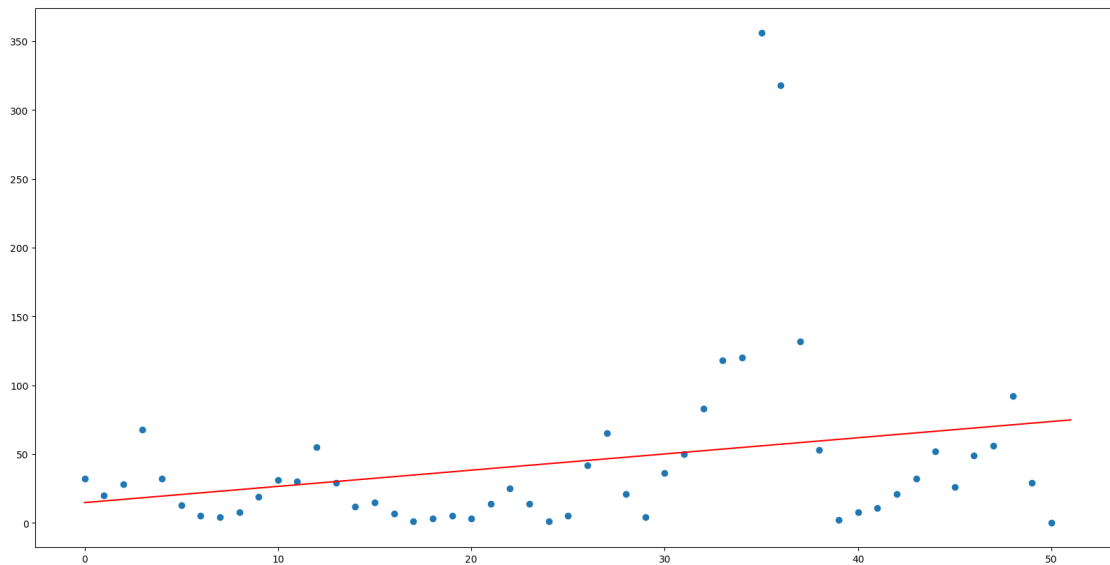
z = np.polyfit(x, y, 1)

p = np.poly1d(z)

xp = np.linspace(0, len(data_x), 100)

_ = plt.figure(figsize=(20,10))
_ = plt.scatter(x, y)
_ = plt.plot(xp, p(xp), '-', color="red")

```



```
[ ]: # get betas of the regression
```

```
z
```

```
[ ]: array([ 1.17819005, 14.76093514])
```

```
[ ]: # Cant of denuncias for a given date
```

```
given_date = "2025-01-01"
```

```
# months from "2018-01-01" to given_date
```

```
n_months = len(pd.date_range(start="2018-01-01", end=given_date, freq="MS"))
```

```
n_months
```

```
[ ]: 85
```

Proyección para el año 2025

```
[ ]: p(n_months)
```

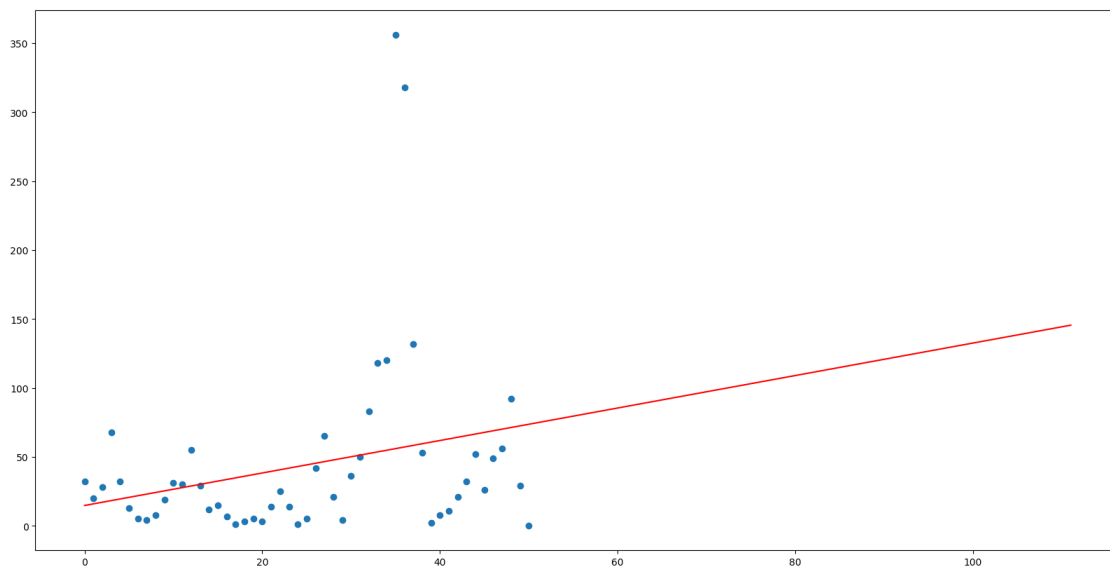
```
[ ]: 114.90708898944189
```

Gráfico de la regresión

```
[ ]: # project 60 months into the future

xp = np.linspace(0, len(data_x) + 60, 100)

_ = plt.figure(figsize=(20,10))
_ = plt.scatter(x, y)
_ = plt.plot(xp, p(xp), '-', color="red")
```



1.8 Canales de comunicación del SUACI

```
[ ]: canales = mosquitos_2021["canal"].value_counts()

total = sum(canales.values)

# Percentage of denuncias by canal

for k, v in canales.items():
    print(f"{k}: {round(v/total*100, 2)}%")
```

GCS Web: 47.55%
App BA 147: 39.15%
Call Center: 5.47%
Boti: 4.72%
Operador GCBA: 1.7%
Comuna: 0.85%
Mail 147: 0.47%
Operador FIXIT: 0.09%