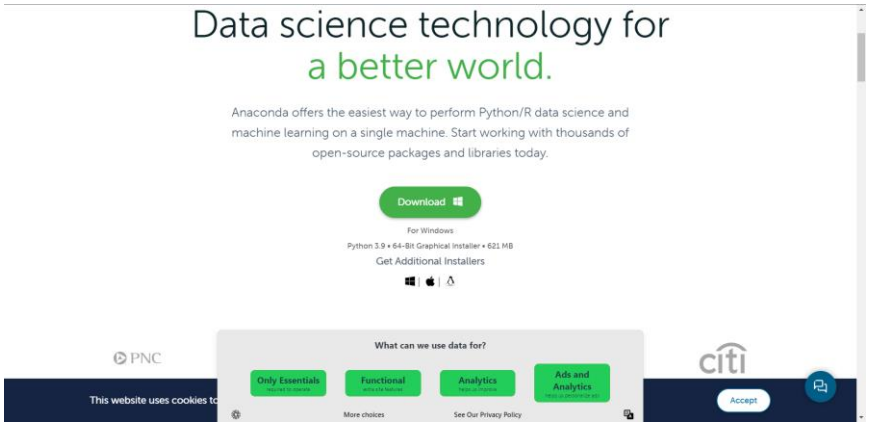
MANUAL DE INSTALACIÓN DEL EXTRACTOR DE DATOS DE CALIDAD DEL AIRE

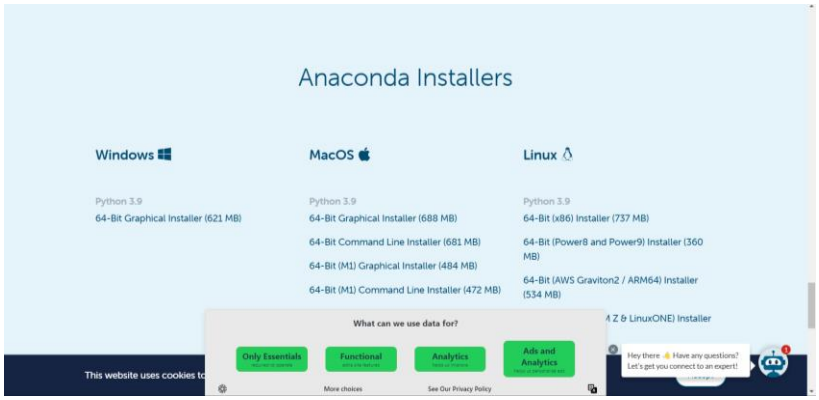
El presente manual corresponde a la instalación de Python a través del IDE Anaconda para procesar los datos obtenidos por las estaciones meteorológicas y obtener la correlación existente entre la concentración de estos contaminantes y el número de defunciones ocurridas por cáncer en la Ciudad de México y el Área Metropolitana. De igual manera se utiliza el programa QGIS como sistema de Información Geográfica para el análisis geoestadístico de las defunciones por cáncer en relación a su ocurrencia geográfica y el nivel de concentración de contaminantes en ese mismo punto. Es importante mencionar que la información georreferenciada de las defunciones ocurridas por cáncer fue modificada de sus coordenadas originales por cuestiones de confidencialidad.

INSTALACIÓN DE ANACONDA (PYTHON IDE)

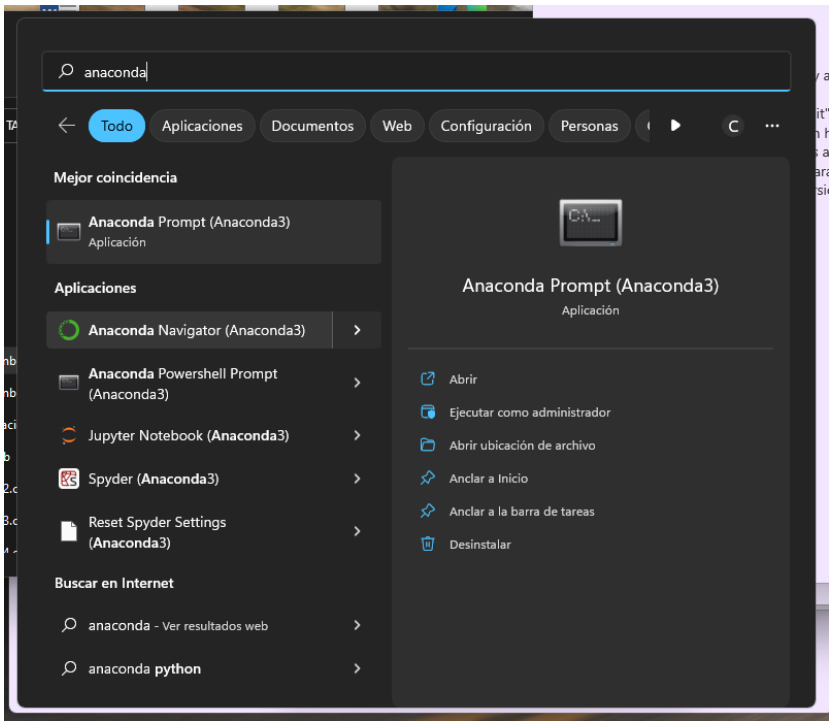
1. Ir a la siguiente dirección Web y buscar el botón “Donwnload”, https://www.anaconda.com/



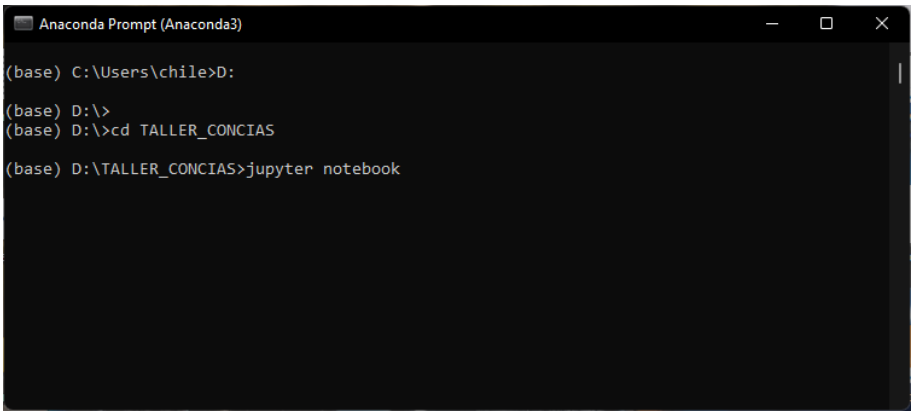
2. Elegimos nuestra plataforma: Windows, Mac o Linux



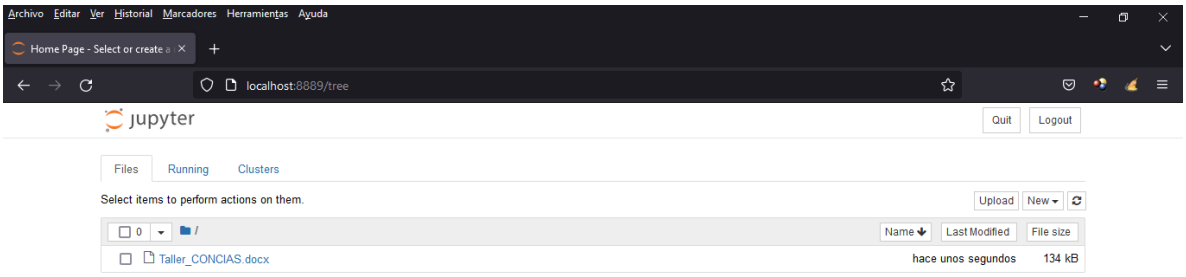
3. Ejecutamos el archivo que descargamos haciendo doble click. Se abrirá un “Típico Wizard” de instalación. Seguiremos los pasos, podemos seleccionar instalación sólo para nuestro usuario, seleccionar la ruta en disco donde instalaremos y listo.
4. Ya instalado, presionamos el botón de Windows en nuestro teclado y escribimos “anaconda promt” (sin las comillas) para ejecutar la consola de Anaconda.



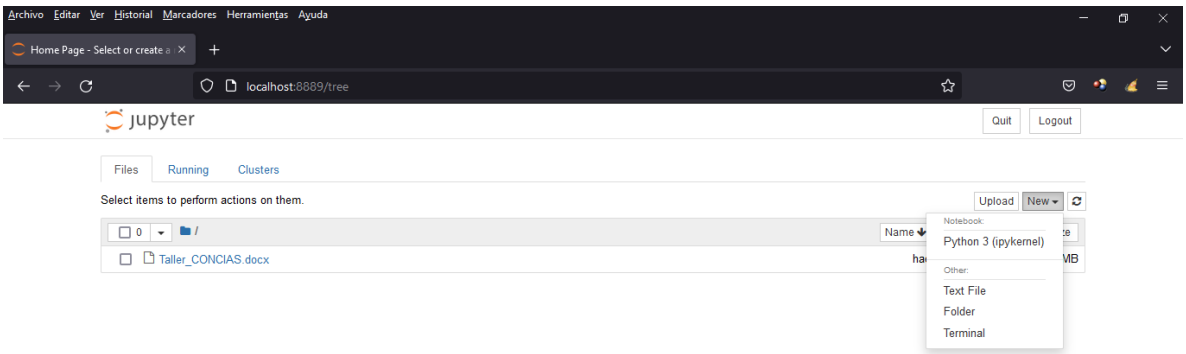
5. Cambiamos del directorio por default a la ruta donde vamos a trabajar y ejecutamos el comando “jupyter notebook” para que se abra en el directorio de trabajo.



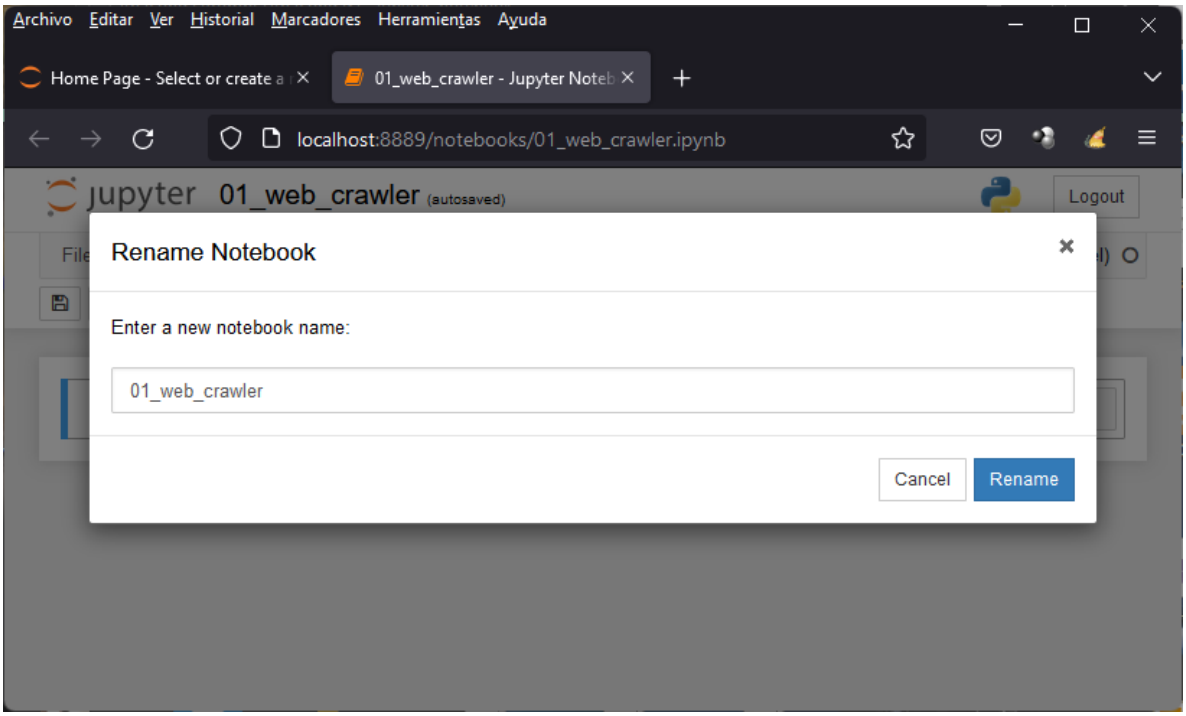
6. Se abrirá en un navegador Jupyter Notebook en el directorio especificado.



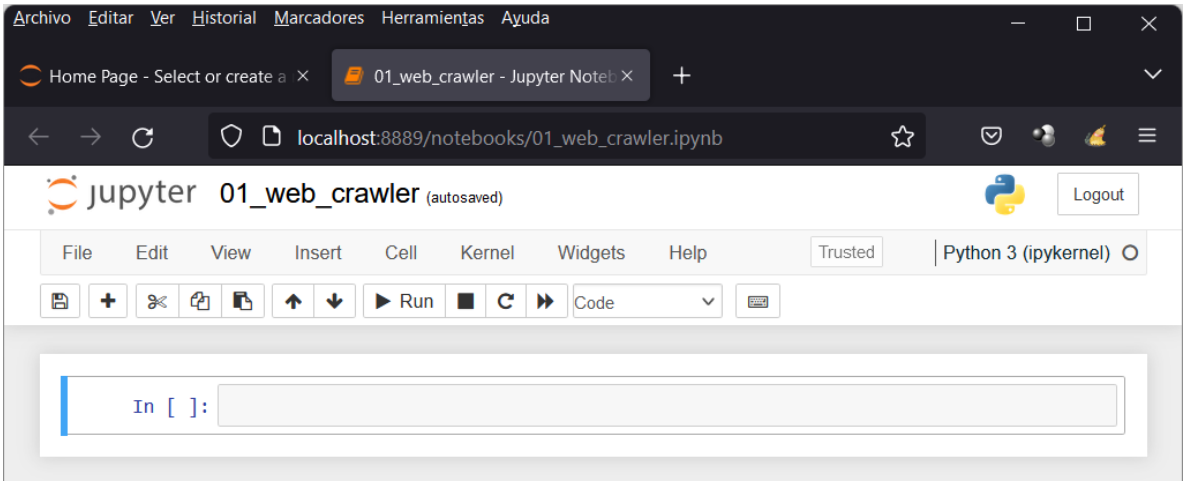
7. Creamos un nuevo Notebook.



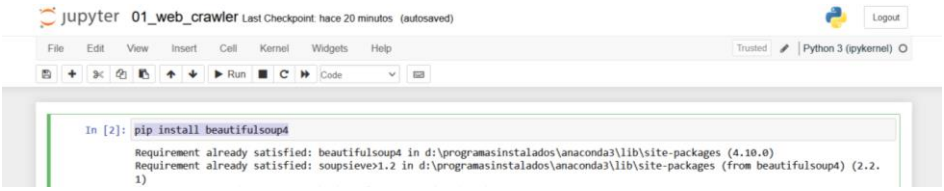
8. Hacemos clic en donde dice “Untitled” para renombrar el nuevo notebook, lo nombramos como “01_web_crawler” (Sin las comillas) con el cual extraeremos los archivos .csv desde la página Web de la red de estaciones meteorológicas.



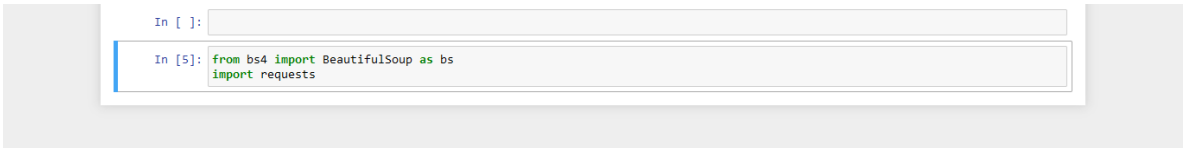
9. Se habrá cambiado el nombre del notebook.



10. Instalamos Beautiful Soup con el comando “pip install beautifulsoup4”



11. Importamos las librerías “BeautifulSoup” y “request”



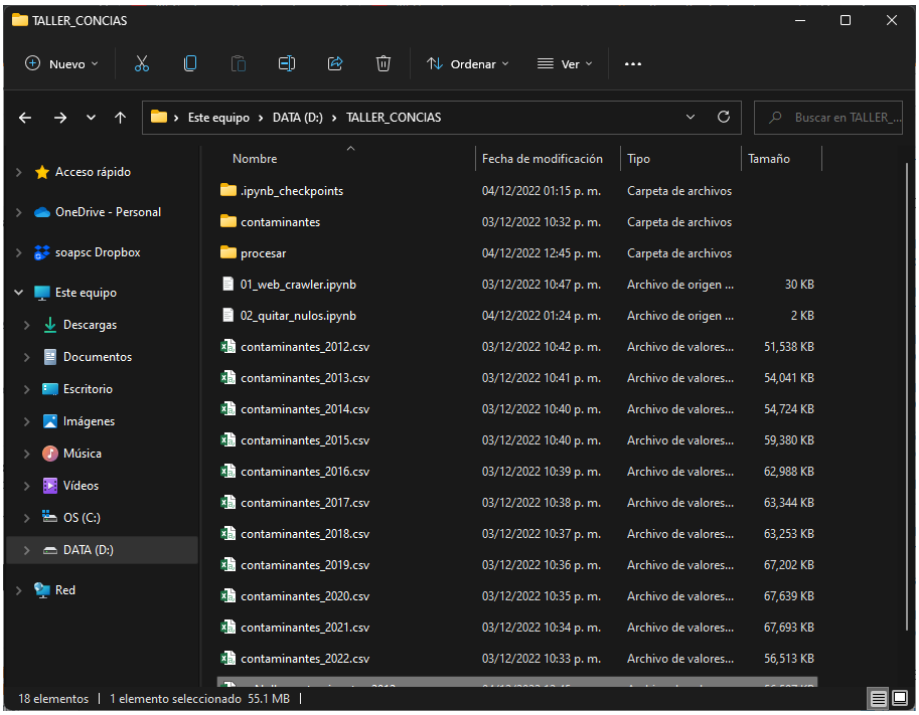
12. Creamos una carpeta en el directorio de trabajo llamada “contaminantes” que es donde se descargarán los archivos de lecturas de las estaciones meteorológicas en formatos CSV.
13. Ejecutamos el siguiente código y veremos cómo la carpeta recién creada se empieza a llenar con los archivos extraídos de manera remota.

```
DOMAIN =   'http://www.aire.cdmx.gob.mx/'
FILETYPE = '.csv'
QUITARFILETYPE = '.gz'

def get_soup(url):
   return bs(requests.get('url').text,'html.parser')
```

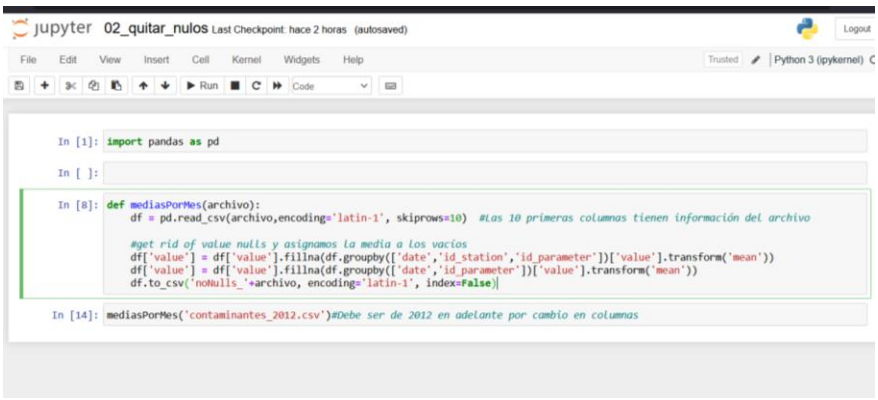
```
html                                                                    =
bs(requests.get('http://www.aire.cdmx.gob.mx/default.php?opc=%27aKBhnmI=%27&opcion=Zg==').text,'html.parser')

for tabla in html.findAll('table'):
    for link in tabla.findAll('a'):
        if link.get('href'):
            file_link= link.get('href')
            print(file_link)
            if FILETYPE in file_link:
                if QUITARFILETYPE not in file_link:
                    url= DOMAIN+file_link
                    local_filename = url.split('/')[-1]
                    r = requests.get(url,stream=True)
                    r.raise_for_status()
                    with open('contaminantes/'+local_filename, 'wb') as f:
                        for chunk in r.iter_content(chunk_size=1024):
                            if chunk:
                                f.write(chunk)
```

DATA WRANGLING (LIMPIEZA, EXPLORACIÓN Y VISUALIZACIÓN DE DATOS)

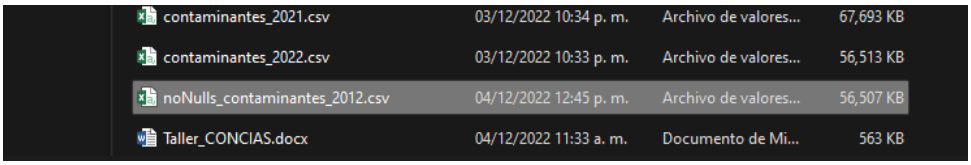
1. De la carpeta “contaminantes” copiamos al directorio de trabajo los archivos csv de 2012 en adelante.



2. Creamos un nuevo notebook llamado “02_quitar_nulos” y agregamos las líneas de código que aparecen en la imagen.



3. El resultado de la ejecución del código anterior será un nuevo archivo llamado “noNull_contaminantes_2012.csv”. Ejecutamos el código para cada uno de los archivos de 2012 a 2022.



4. Verificamos el número de registros con valores nulos y vemos que no existen, de igual manera revisamos cuántos registros hay en total sólo para el año 2012 y observamos que son 1, 722, 000 registros.

5. Modificamos la función y la renombramos como “mediasPorMes(archivo)”. Como vamos a leer los archivos ya sin valores nulos generados por la función anterior, sólo asignamos al campo “date” como tipo datetime para poder

agrupar por mes los registros y el campo “value” se le asigna la media mensual por estación y por parámetro (contaminante).

```
In [51]: def mediasPorMes(archivo):
             df = pd.read_csv(archivo,encoding='latin-1')
             df['date']= pd.to_datetime(df['date'],dayfirst=True, errors='coerce') #Lo convertimos a datetime

             df_mes = df.set_index('date').groupby([pd.Grouper(freq='M'),'id_station','id_parameter']).mean()
             df_mes.to_csv('noNullsMes_'+archivo, encoding='latin-1')

In [52]: mediasPorMes('noNulls_contaminantes_2012.csv')
```

6. Verificamos el resultado y comprobamos cuántos registros tenemos ahora.

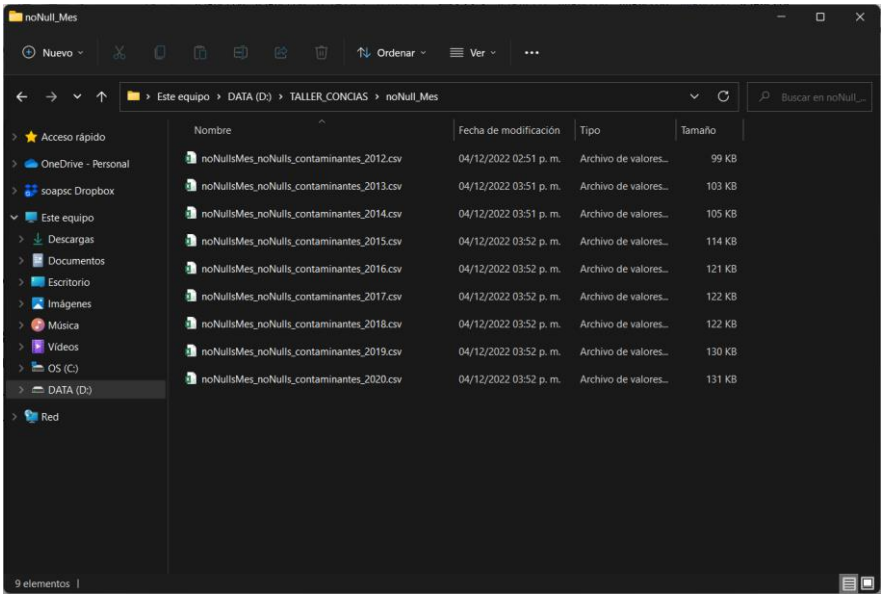
```
In [55]: df_temp = pd.read_csv('noNullsMes_noNulls_contaminantes_2012.csv',encoding='latin-1')
         df_temp.head()
```

Out[55]:

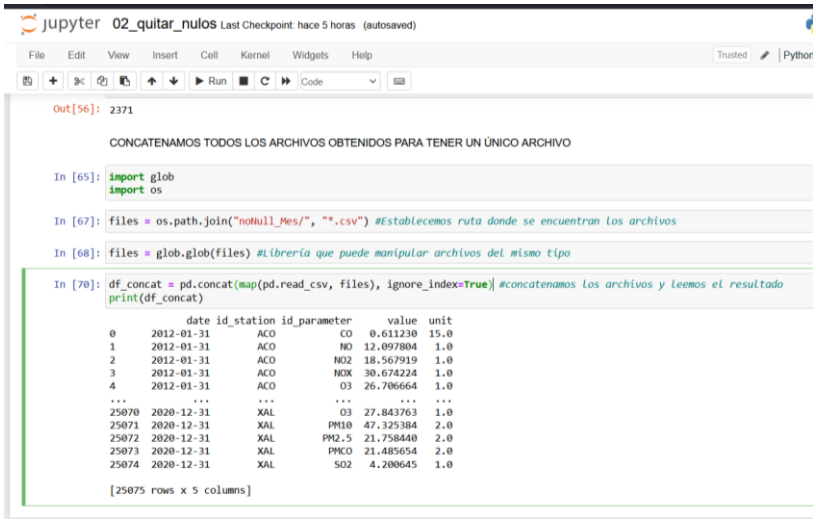
	date	id_station	id_parameter	value	unit
0	2012-01-31	ACO	CO	0.611230	15.0
1	2012-01-31	ACO	NO	12.097804	1.0
2	2012-01-31	ACO	NO2	18.567919	1.0
3	2012-01-31	ACO	NOX	30.674224	1.0
4	2012-01-31	ACO	O3	26.706664	1.0

```
In [56]: len(df_temp)
Out[56]: 2371
```

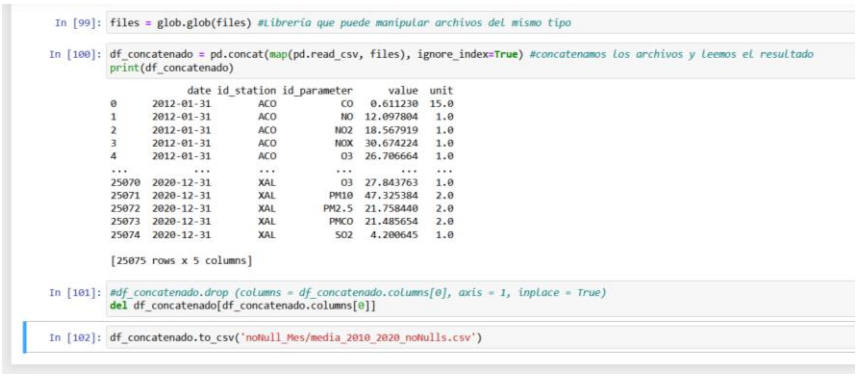
7. Creamos una carpeta llamada “noNull_Mes” y agregamos los archivos resultantes del paso anterior.



8. Concatenamos todos los archivos resultantes para que nos quedemos con un único archivo con el cual trabajar.



9. Guardamos el resultado en un nuevo archivo el cual llamaremos “media_2010_2022_noNulls.csv” el cual contendrá ahora sólo 25075 registros. Como no ocupamos la primera columna la eliminamos.

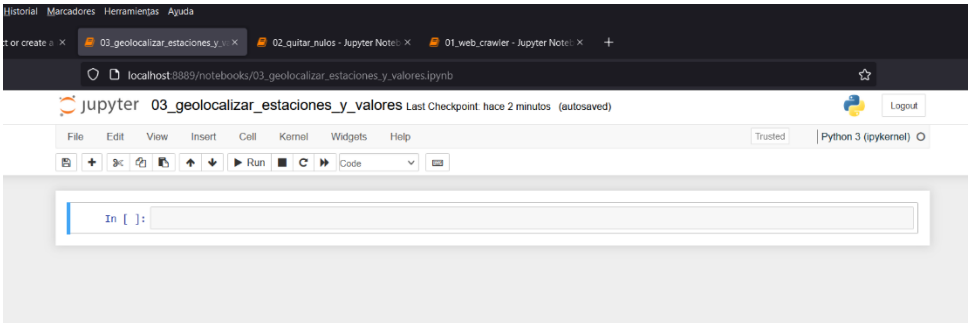


Este archivo resultante podría ser utilizado en algún SGBD que es como se crearon varias de las tablas en SQL Server utilizadas para el Visor de mapas Espacio Temporal, así como el Visor de mapas de Enfermedades Respiratorias.

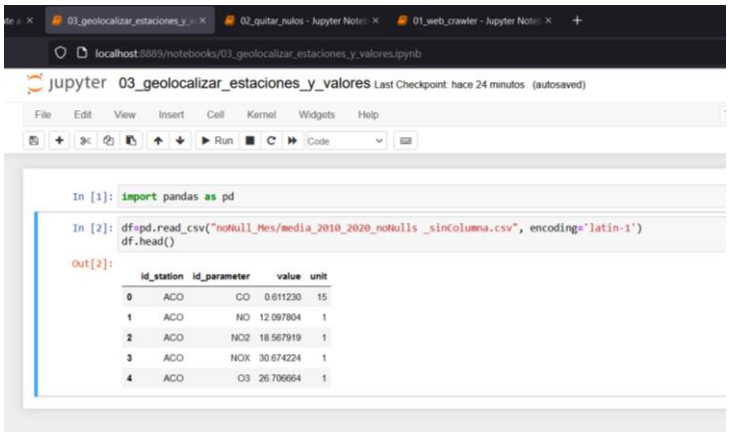
GEOLOCALIZAR ESTACIONES Y VALORES

De los archivos generados en el paso anterior, es necesario que podamos asignar a cada estación los valores de las mediciones para que éstas puedan ser georreferenciadas.

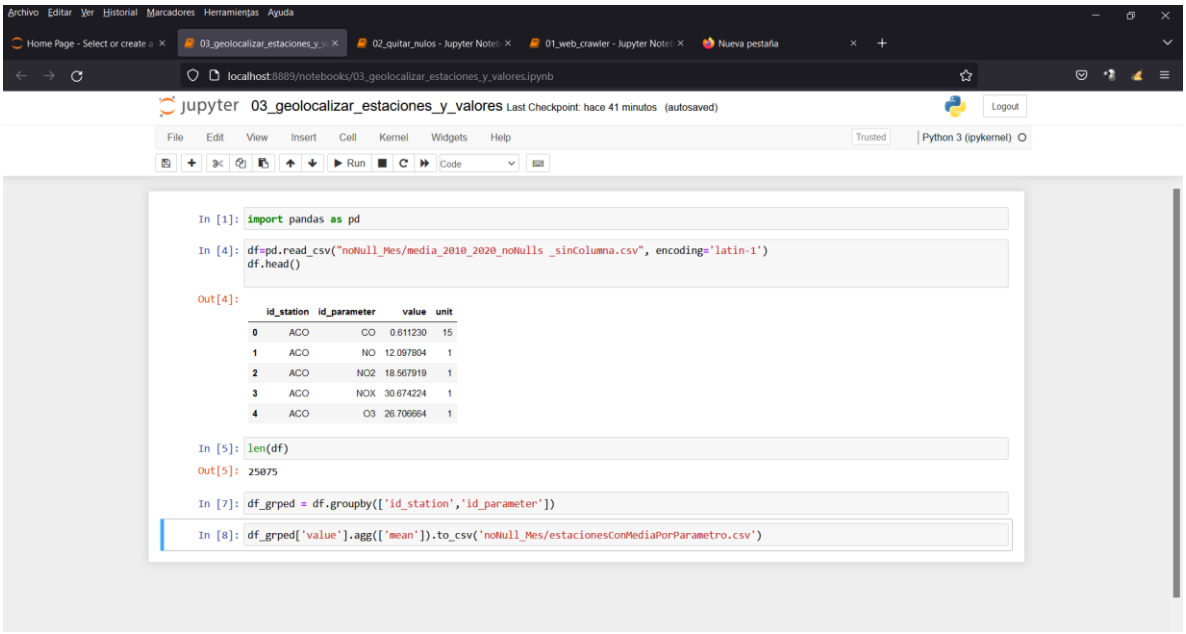
1. Creamos un nuevo notebook llamado “geolocalizar_estaciones_y_valores”



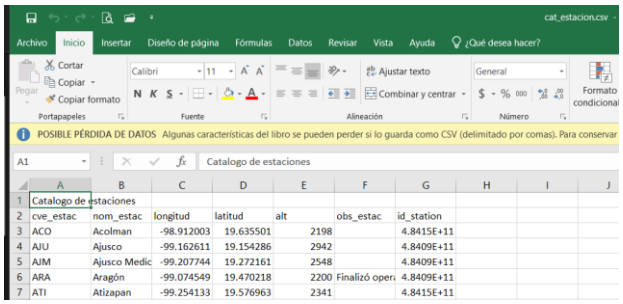
2. Abrimos el archivo generado en la fase anterior.



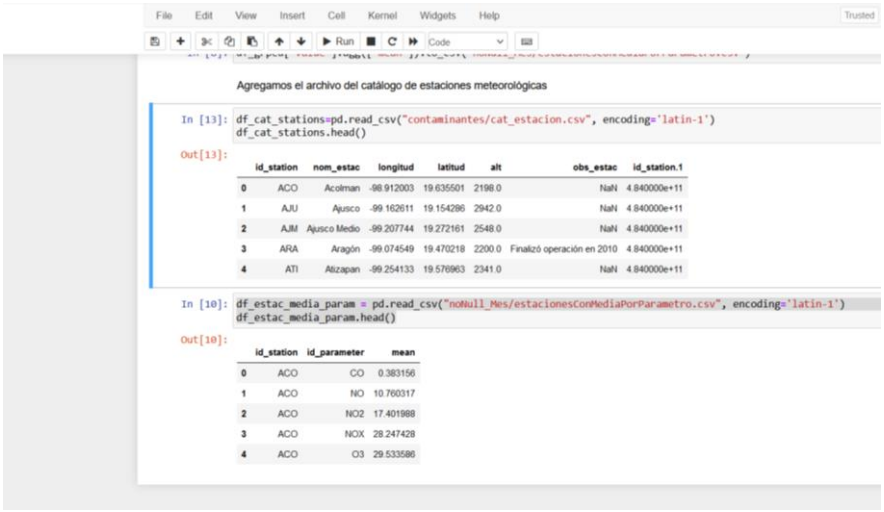
3. Agrupamos por estación y por parámetro, tomamos el valor de la media del campo “value” y lo asignamos a un nuevo campo “mean” que tendrá el valor de la media resultante y guardamos el resultado como un nuevo archivo CSV llamado “estacionesConMediaPorParametro.csv”



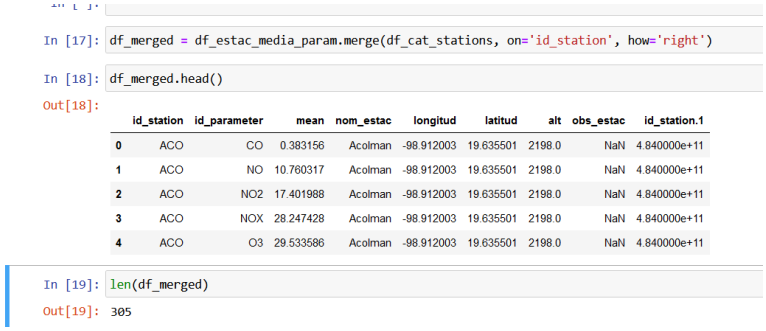
4. Necesitamos asociar este último archivo del paso 3 con la ubicación (coordenadas geográficas) de las estaciones meteorológicas. Para ello, en la carpeta "contaminantes" debemos tener el archivo "cat_estaciones.csv" el cual fue descargado mediante el "Extractor de datos de calidad del aire", podemos abrirlo manualmente y eliminar la primer fila que no necesitamos.



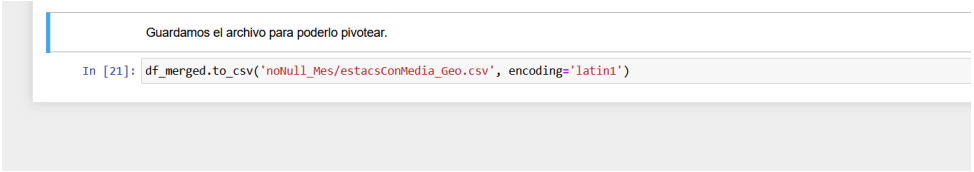
5. Lo cargamos y vemos el contenido. También cargamos el archivo con las medias por parámetro. Vemos que en este archivo la columna "cve_estac" no coincide con la columna "id_station" del archivo resultante en el paso anterior. Cambiamos manualmente el nombre de la columna en el catálogo de estaciones para que coincidan.

6. 

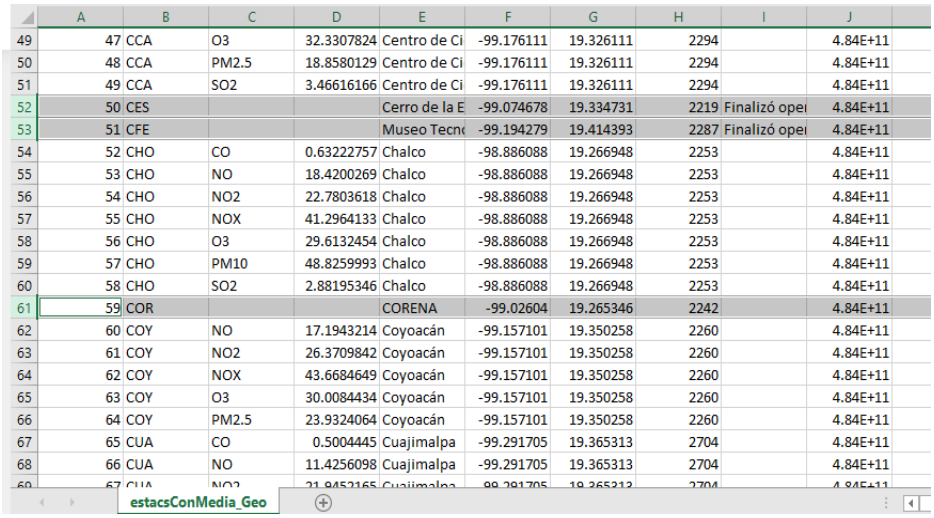
7. Como ahora los dos campos coinciden, los dos archivos se podrán mezclar (merge).



8. Guardamos el resultado en un archivo CSV llamado “estacsConMedia_Geo.csv”



9. Si abrimos el archivo resultante podemos ver que existe algunos vacíos en los campos “id_parameter” y “mean”. Necesitamos eliminarlos antes de ajustar la tabla ya que se debe a estaciones que por ya no existir dejaron de medir esos parámetros.



	A	B	C	D	E	F	G	H	I	J
49	47	CCA	O3	32.3307824	Centro de Ci	-99.176111	19.326111	2294		4.84E+11
50	48	CCA	PM2.5	18.8580129	Centro de Ci	-99.176111	19.326111	2294		4.84E+11
51	49	CCA	SO2	3.46616166	Centro de Ci	-99.176111	19.326111	2294		4.84E+11
52	50	CES			Cerro de la E	-99.074678	19.334731	2219	Finalizó oper	4.84E+11
53	51	CFE			Museo Tecn	-99.194279	19.414393	2287	Finalizó oper	4.84E+11
54	52	CHO	CO	0.63222757	Chalco	-98.886088	19.266948	2253		4.84E+11
55	53	CHO	NO	18.4200269	Chalco	-98.886088	19.266948	2253		4.84E+11
56	54	CHO	NO2	22.7803618	Chalco	-98.886088	19.266948	2253		4.84E+11
57	55	CHO	NOX	41.2964133	Chalco	-98.886088	19.266948	2253		4.84E+11
58	56	CHO	O3	29.6132454	Chalco	-98.886088	19.266948	2253		4.84E+11
59	57	CHO	PM10	48.8259993	Chalco	-98.886088	19.266948	2253		4.84E+11
60	58	CHO	SO2	2.88195346	Chalco	-98.886088	19.266948	2253		4.84E+11
61	59	COR			CORENA	-99.02604	19.265346	2242		4.84E+11
62	60	COY	NO	17.1943214	Coyoacán	-99.157101	19.350258	2260		4.84E+11
63	61	COY	NO2	26.3709842	Coyoacán	-99.157101	19.350258	2260		4.84E+11
64	62	COY	NOX	43.6684649	Coyoacán	-99.157101	19.350258	2260		4.84E+11
65	63	COY	O3	30.0084434	Coyoacán	-99.157101	19.350258	2260		4.84E+11
66	64	COY	PM2.5	23.9324064	Coyoacán	-99.157101	19.350258	2260		4.84E+11
67	65	CUA	CO	0.5004445	Cuajimalpa	-99.291705	19.365313	2704		4.84E+11
68	66	CUA	NO	11.4256098	Cuajimalpa	-99.291705	19.365313	2704		4.84E+11
69	67	CUA	NO2	21.9452165	Cuajimalpa	-99.291705	19.365313	2704		4.84E+11

estacsConMedia_Geo

10. Abrimos el archivo, quitamos los valores del campo “mean” que no contengan valor y guardamos el resultado como “estacsConMedia_Geo_noNulls.csv”

```
In [ ]:
In [22]: df_estacs_geo=pd.read_csv("noNull_Mes/estacsConMedia_Geo.csv", encoding='latin-1')
In [23]: df_estacs_geo_noNulls = df_estacs_geo[df_estacs_geo['mean'].notna()]
In [28]: df_estacs_geo_noNulls.to_csv('noNull_Mes/estacsConMedia_Geo_noNulls.csv', encoding='latin1')
```

11. Al abrir el archivo vemos que hay dos columnas que no necesitamos, las podemos eliminar desde el archivo original y lo volvemos a cargar.

```
In [ ]:
In [30]: df_estacs_geo_01=pd.read_csv("noNull_Mes/estacsConMedia_Geo_noNulls.csv", encoding='latin-1')
         df_estacs_geo_01.head()
```

Out[30]:

	Unnamed: 0	Unnamed: 0.1	id_station	id_parameter	mean	nom_estac	longitud	latitud	alt	obs_estac	id_station.1
0	0	0	ACO	CO	0.383156	Acolman	-98.912003	19.635501	2198.0	NaN	4.840000e+11
1	1	1	ACO	NO	10.760317	Acolman	-98.912003	19.635501	2198.0	NaN	4.840000e+11
2	2	2	ACO	NO2	17.401988	Acolman	-98.912003	19.635501	2198.0	NaN	4.840000e+11
3	3	3	ACO	NOX	28.247428	Acolman	-98.912003	19.635501	2198.0	NaN	4.840000e+11
4	4	4	ACO	O3	29.533586	Acolman	-98.912003	19.635501	2198.0	NaN	4.840000e+11

```
In [32]: df_estacs_geo_01=pd.read_csv("noNull_Mes/estacsConMedia_Geo_noNulls.csv", encoding='latin-1')
         df_estacs_geo_01.head()
```

Out[32]:

	id_station	id_parameter	mean	nom_estac	longitud	latitud	alt	obs_estac	id_station.1
0	ACO	CO	0.383156	Acolman	-98.912003	19.635501	2198	NaN	4.840000e+11
1	ACO	NO	10.760317	Acolman	-98.912003	19.635501	2198	NaN	4.840000e+11
2	ACO	NO2	17.401988	Acolman	-98.912003	19.635501	2198	NaN	4.840000e+11
3	ACO	NOX	28.247428	Acolman	-98.912003	19.635501	2198	NaN	4.840000e+11
4	ACO	O3	29.533586	Acolman	-98.912003	19.635501	2198	NaN	4.840000e+11

12. Ahora sí podemos pivotear (convertir los contaminantes a columnas) el dataframe.

```
In [ ]:
```

```
In [38]: df_estacs_geo_01.pivot_table('mean',
['id_station','nom_estac','latitud','longitud'],'id_parameter').to_csv('noNull_Mes/estacsMedia_Geo_pivoted.csv', encoding='latin-
```

```
In [39]: df_estacs_geo_pivot=pd.read_csv("noNull_Mes/estacsMedia_Geo_pivoted.csv", encoding='latin-1')
df_estacs_geo_pivot.head()
```

Out[39]:

	id_station	nom_estac	latitud	longitud	CO	NO	NO2	NOX	O3	PM10	PM2.5	PMCO	SO2
0	ACO	Acolman	19.635501	-98.912003	0.383156	10.760317	17.401988	28.247428	29.533586	40.882878	NaN	NaN	2.828110
1	AJM	Ajusco Medio	19.272161	-99.207744	0.416988	5.955789	17.793649	23.817452	39.134903	34.591827	19.503123	14.892344	3.260812
2	AJU	Ajusco	19.154286	-99.162611	NaN	NaN	NaN	NaN	34.440474	NaN	19.633707	NaN	NaN
3	ATI	Atizapan	19.576963	-99.254133	0.524848	14.294743	22.365726	36.682922	27.950104	41.278804	NaN	NaN	5.707607
4	BJU	Benito Juárez	19.370464	-99.159596	0.526351	NaN	21.485584	NaN	31.259555	39.579430	22.474065	17.099596	4.388510

```
In [ ]:
```

13. Como se puede observar existen vacío en los valores de medición de algunos elementos ya que las estaciones meteorológicas no miden todos los contaminantes. Para resolver esto podemos asignarles a los vacíos el valor medio del resto de las mediciones para ese elemento. Tenemos que hacerlo, elemento por elemento. En este caso es para 'CO'.

Out[42]:

	id_station	nom_estac	latitud	longitud	CO	NO	NO2	NOX	O3	PM10	PM2.5	PMCO	SO2
0	ACO	Acolman	19.635501	-98.912003	0.383156	10.760317	17.401988	28.247428	29.533586	40.882878	NaN	NaN	2.828110
1	AJM	Ajusco Medio	19.272161	-99.207744	0.416988	5.955789	17.793649	23.817452	39.134903	34.591827	19.503123	14.892344	3.260812
2	AJU	Ajusco	19.154286	-99.162611	NaN	NaN	NaN	NaN	34.440474	NaN	19.633707	NaN	NaN
3	ATI	Atizapan	19.576963	-99.254133	0.524848	14.294743	22.365726	36.682922	27.950104	41.278804	NaN	NaN	5.707607
4	BJU	Benito Juárez	19.370464	-99.159596	0.526351	NaN	21.485584	NaN	31.259555	39.579430	22.474065	17.099596	4.388510

```
In [ ]:
```

```
In [44]: df_estacs_geo_pivot['CO']=df_estacs_geo_pivot['CO'].fillna(df_estacs_geo_pivot['CO'].mean())
df_estacs_geo_pivot.head()
```

Out[44]:

	id_station	nom_estac	latitud	longitud	CO	NO	NO2	NOX	O3	PM10	PM2.5	PMCO	SO2
0	ACO	Acolman	19.635501	-98.912003	0.383156	10.760317	17.401988	28.247428	29.533586	40.882878	NaN	NaN	2.828110
1	AJM	Ajusco Medio	19.272161	-99.207744	0.416988	5.955789	17.793649	23.817452	39.134903	34.591827	19.503123	14.892344	3.260812
2	AJU	Ajusco	19.154286	-99.162611	0.576028	NaN	NaN	NaN	34.440474	NaN	19.633707	NaN	NaN
3	ATI	Atizapan	19.576963	-99.254133	0.524848	14.294743	22.365726	36.682922	27.950104	41.278804	NaN	NaN	5.707607
4	BJU	Benito Juárez	19.370464	-99.159596	0.526351	NaN	21.485584	NaN	31.259555	39.579430	22.474065	17.099596	4.388510

```
In [ ]: df_estacs_geo_pivot=df_estacs_geo_pivot.fillna(df_estacs_geo_pivot['CO'].mean())
df_estacs_geo_pivot.head()
```

14. Guardamos el resultado en un archivo llamado "estacsMedia_Geo_pivoted_NO_Nulls.csv" para ya poderlo utilizar en QGIS.

```
In [54]: df_estacs_geo_pivot['SO2']=df_estacs_geo_pivot['SO2'].fillna(df_estacs_geo_pivot['SO2'].mean())
         df_estacs_geo_pivot.head()
```

Out[54]:

	id_station	nom_estac	latitud	longitud	CO	NO	NO2	NOX	O3	PM10	PM2.5	PMCO	SO2	NX
0	ACO	Acolman	19.635501	-98.912003	0.383156	10.760317	17.401988	28.247428	29.533586	40.882878	22.107702	20.538719	2.828110	28.247428
1	AJM	Ajusco Medio	19.272161	-99.207744	0.416988	5.955789	17.793649	23.817452	39.134903	34.591827	19.503123	14.892344	3.260812	23.817452
2	AJU	Ajusco	19.154286	-99.162611	0.576028	16.929812	23.530309	41.153246	34.440474	43.090817	19.633707	20.538719	4.197430	41.153246
3	ATI	Atizapan	19.576963	-99.254133	0.524848	14.294743	22.365726	36.682922	27.950104	41.278804	22.107702	20.538719	5.707607	36.682922
4	BJU	Benito Juárez	19.370464	-99.159596	0.526351	16.929812	21.485584	41.153246	31.259555	39.579430	22.474065	17.099596	4.388510	41.153246

GUARDAMOS EL RESULTADO COMO UN ARCHIVO CSV QUE SE PODRÁ UTILIZAR EN QGIS.

```
In [55]: df_estacs_geo_pivot.to_csv('noNull_Mes/estacsMedia_Geo_pivoted_NO_Nulls.csv', encoding='latin-1')
```

SISTEMAS DE INFORMACIÓN GEOGRÁFICA

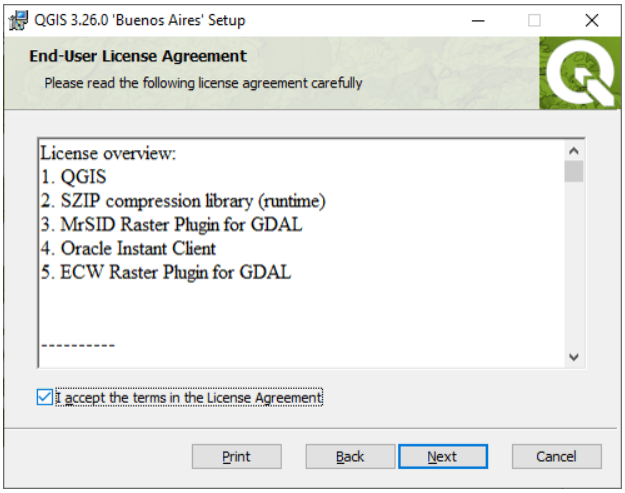
Un Sistema de Información Geográfica (SIG o GIS en inglés) son un conjunto de software, hardware y procesos elaborados que facilitan el análisis, gestión y representación de datos georreferenciados o con información geoespacial.

QGIS es un SIG el cual se puede descargar desde http://www.qgis.org/ En la sección de descargas seleccionamos el instalador de acuerdo a nuestro sistema operativo.

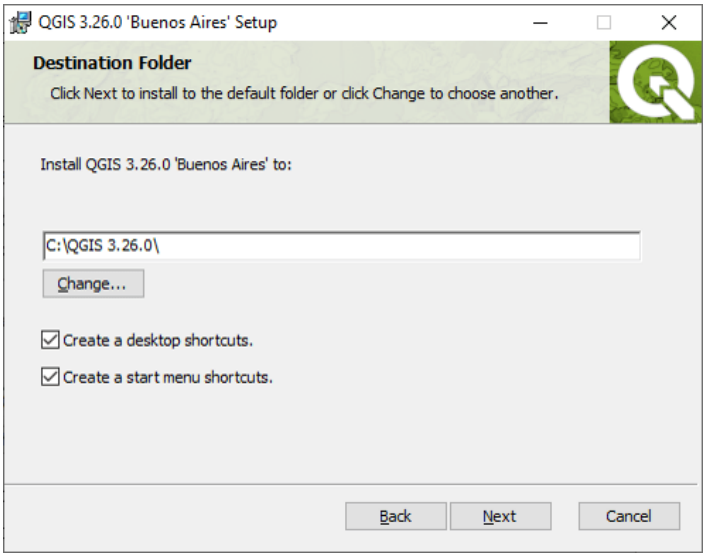
Instalación de QGIS

1. Ya descargado el instalador de QGIS lo ejecutamos. Aparecerá la siguiente pantalla.

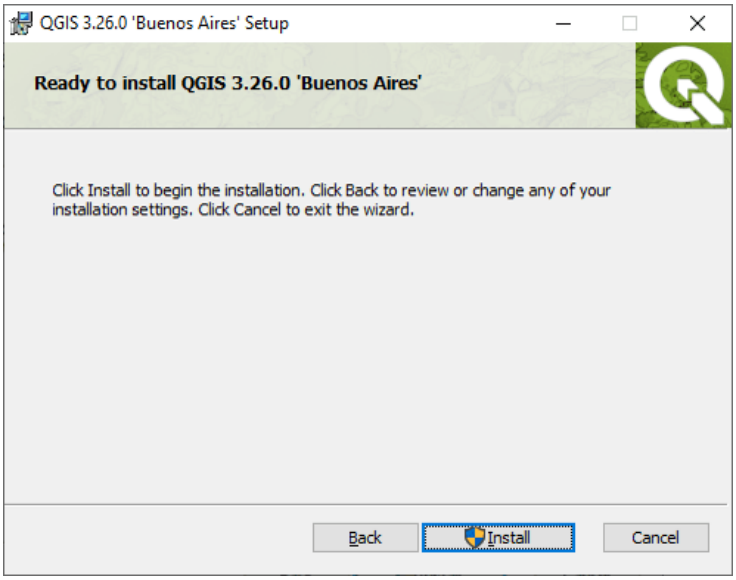
2. Aceptamos los términos de licencia y continuamos.



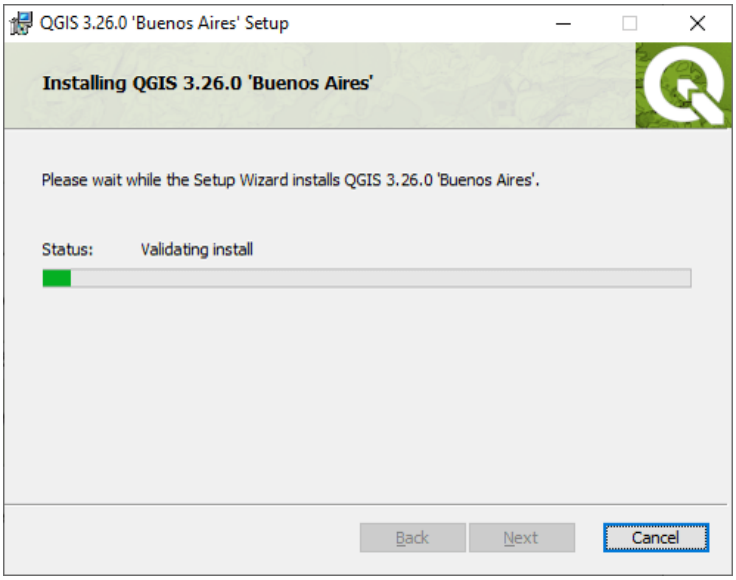
3. Aceptamos la ruta de instalación por defecto.



4. Instalamos el programa.

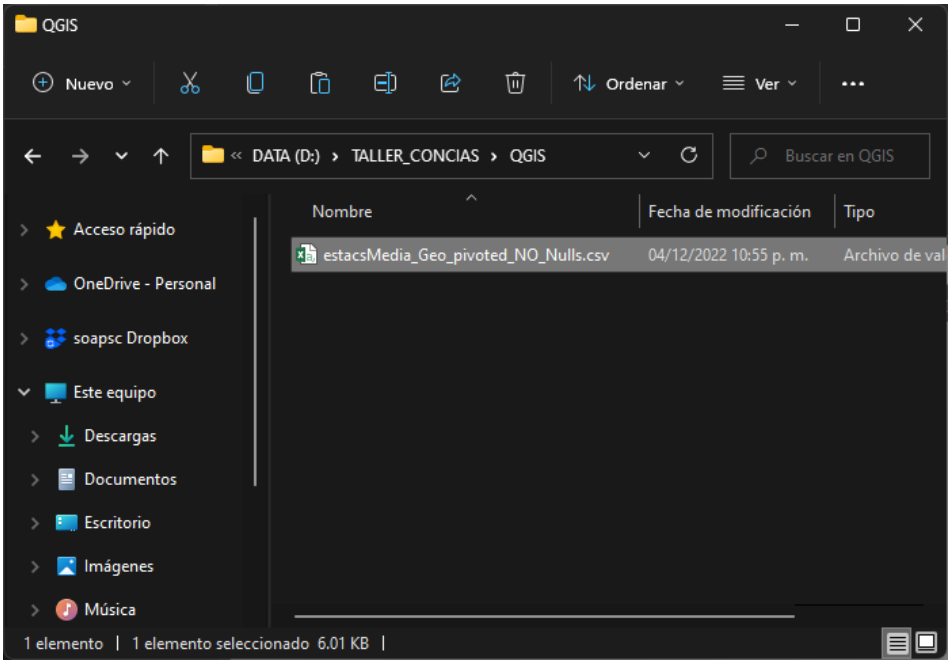


5. Comienza el proceso de instalación.

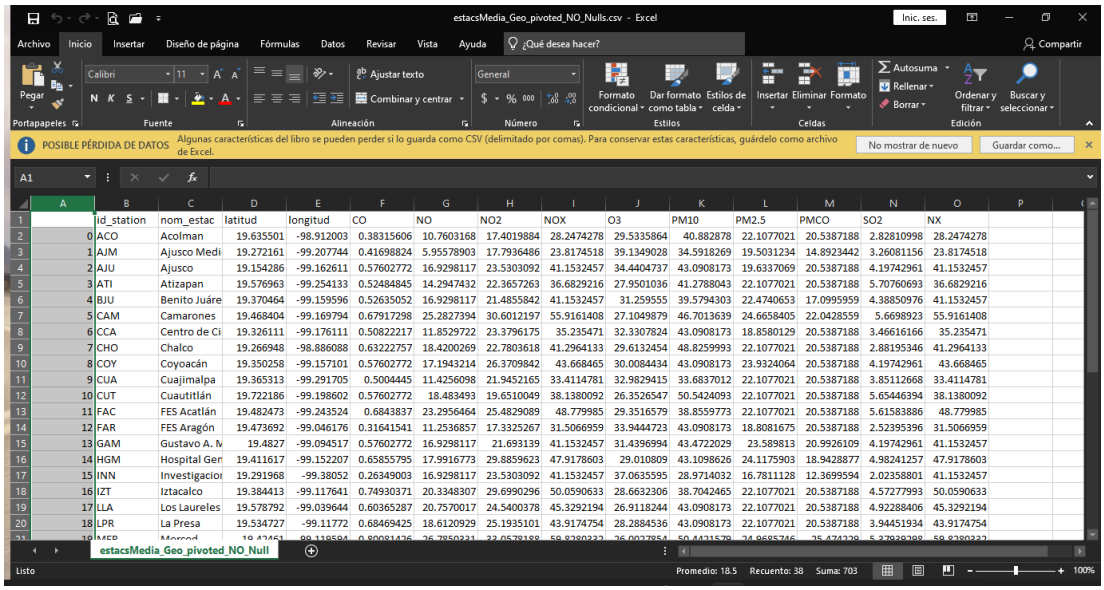


6. Terminamos la instalación.

7. Habiendo instalado QGIS, copiamos el archivo generado en la fase anterior a la carpeta QGIS

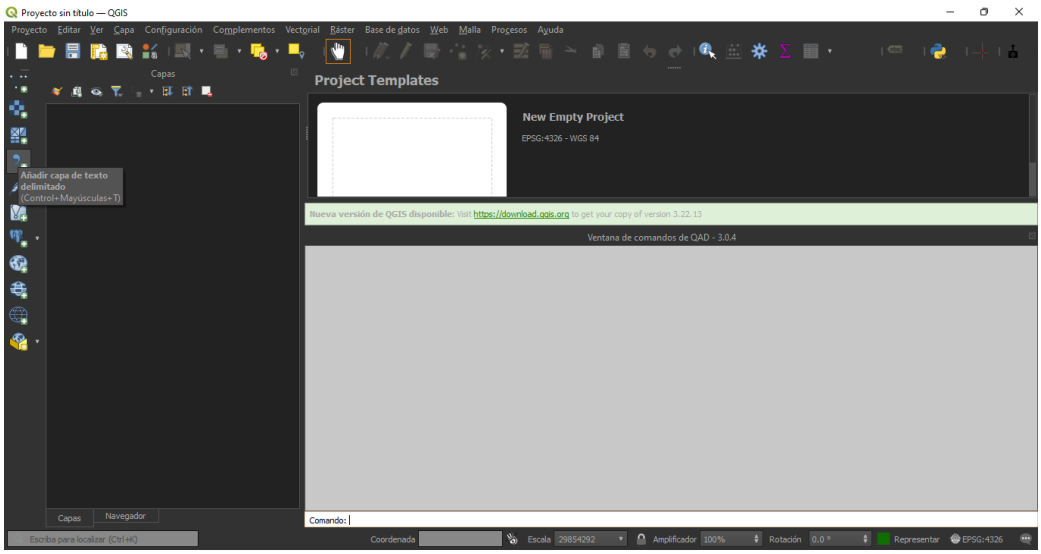


8. Revisamos el archivo y en caso de que se encuentre una columna extra la eliminamos.

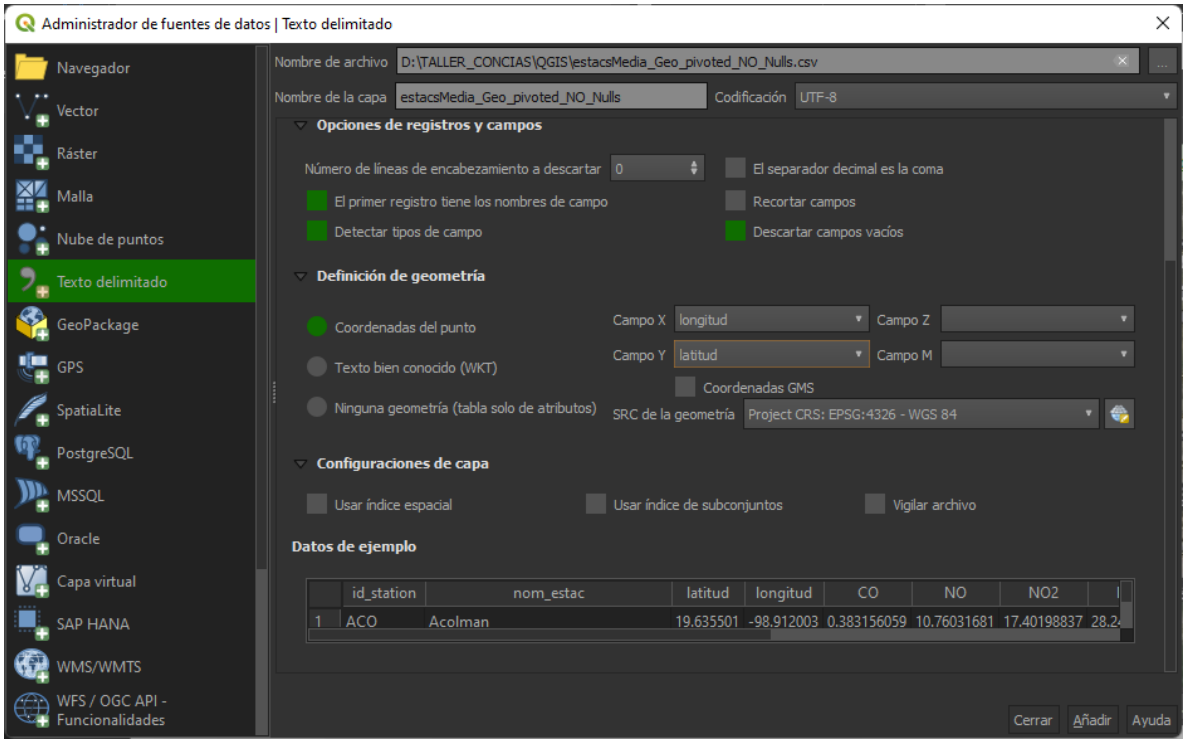


	id_station	nom_estac	latitud	longitud	CO	NO	NO2	NOX	O3	PM10	PM2.5	PMCO	SO2	NX
0	ACO	Acolman	19.635501	-98.912003	0.38315606	10.7603168	17.4019884	28.2474278	29.5335864	40.882878	22.1077021	20.5387188	2.82810998	28.2474278
1	AJM	Ajusco Medi	19.272161	-99.207744	0.41698824	5.95578903	17.7936486	23.8174518	39.1349028	34.5918269	19.5031234	14.8923442	3.26081156	23.8174518
2	AJU	Ajusco	19.154286	-99.162611	0.57602772	16.9298117	23.5303092	41.1532457	34.4404737	43.0908173	19.6337069	20.5387188	4.19742961	41.1532457
3	ATI	Atizapan	19.576963	-99.254133	0.52484845	14.2947432	22.3657263	36.6829216	27.9501036	41.2788043	22.1077021	20.5387188	5.70760693	36.6829216
4	BJU	Benito Juáre	19.370464	-99.159596	0.52635052	16.9298117	21.4855842	41.1532457	31.259555	39.5794303	22.4740653	17.0995959	4.38850976	41.1532457
5	CAM	Camarones	19.468404	-99.169794	0.67917298	25.2827394	30.6012197	55.9161408	27.1049879	46.7013639	24.6658405	22.0428559	5.6698923	55.9161408
6	CCA	Centro de Ci	19.326111	-99.176111	0.50822217	11.8529722	23.3796175	35.235471	32.3307824	43.0908173	18.8580129	20.5387188	3.46616166	35.235471
7	CHO	Chalco	19.266948	-98.886088	0.63222757	18.4200269	22.7803618	41.2964133	29.6132454	48.8259993	22.1077021	20.5387188	2.88195346	41.2964133
8	COY	Coyoacán	19.350258	-99.157101	0.57602772	17.1943214	26.3709842	43.668465	30.0084434	43.0908173	23.9324064	20.5387188	4.19742961	43.668465
9	CUA	Cuajimalpa	19.365313	-99.291705	0.5004445	11.4256098	21.9452165	33.4114781	32.9829415	33.6837012	22.1077021	20.5387188	3.85112668	33.4114781
10	CUT	Cuautitlán	19.722186	-99.198602	0.57602772	18.483493	19.6510049	38.1380092	26.3526547	50.5424093	22.1077021	20.5387188	5.65446394	38.1380092
11	FAC	FES Acatlán	19.482473	-99.243524	0.6843837	23.2956464	25.4829089	48.779985	29.3516579	38.8559773	22.1077021	20.5387188	5.61583886	48.779985
12	FAR	FES Aragón	19.473692	-99.046176	0.31641541	11.2536857	17.3325267	31.5066959	33.9444723	43.0908173	18.8081675	20.5387188	2.52395396	31.5066959
13	GAM	Gustavo A. M	19.4827	-99.094517	0.57602772	16.9298117	21.693139	41.1532457	31.4396994	43.4722029	23.589813	20.9926109	4.19742961	41.1532457
14	HGM	Hospital Ger	19.411617	-99.152207	0.65855795	17.9916773	29.8859623	47.9178603	29.010809	43.1098626	24.1175903	18.9428877	4.98241257	47.9178603
15	INN	Investigacior	19.291968	-99.38052	0.26349003	16.9298117	23.5303092	41.1532457	37.0635595	28.9714032	16.7811128	12.3699594	2.02358801	41.1532457
16	IZT	Iztacalco	19.384413	-99.117641	0.74930371	20.3348307	29.6990296	50.0590633	28.6632306	38.7042465	22.1077021	20.5387188	4.57277993	50.0590633
17	LLA	Los Laureles	19.578792	-99.039644	0.60365287	20.7570017	24.5400378	45.3292194	26.9118244	43.0908173	22.1077021	20.5387188	4.92288406	45.3292194
18	LPR	La Presa	19.534727	-99.11772	0.68469425	18.6120929	25.1935101	43.9174754	28.2884536	43.0908173	22.1077021	20.5387188	3.94451934	43.9174754

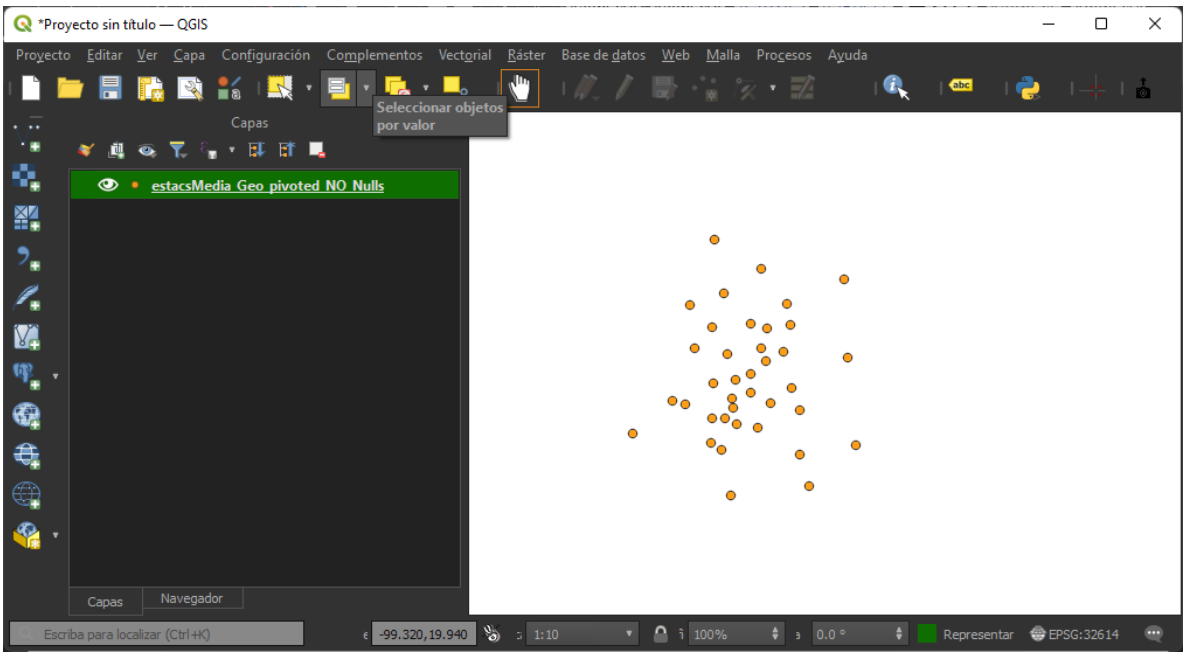
9. Lo importamos en QGIS como archivo CSV con el botón del panel izquierdo.



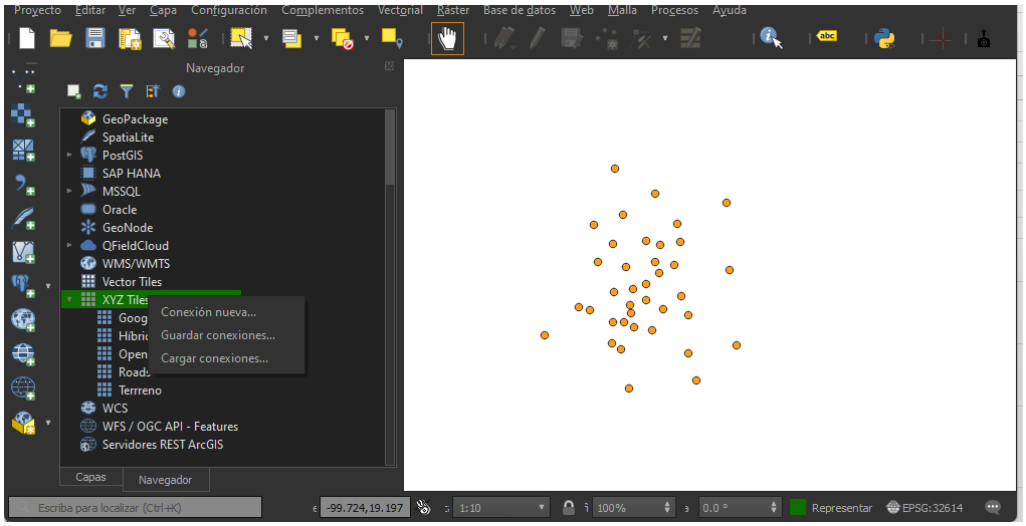
10. Le damos la siguiente configuración.



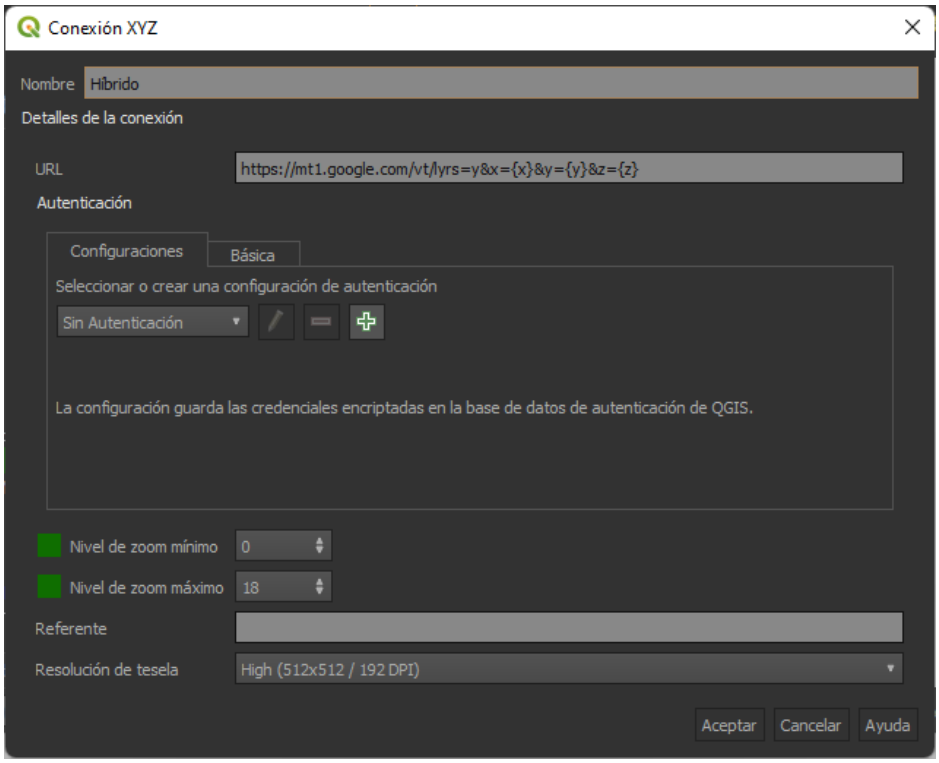
11. Aparecerán los puntos en la interfaz de QGIS.



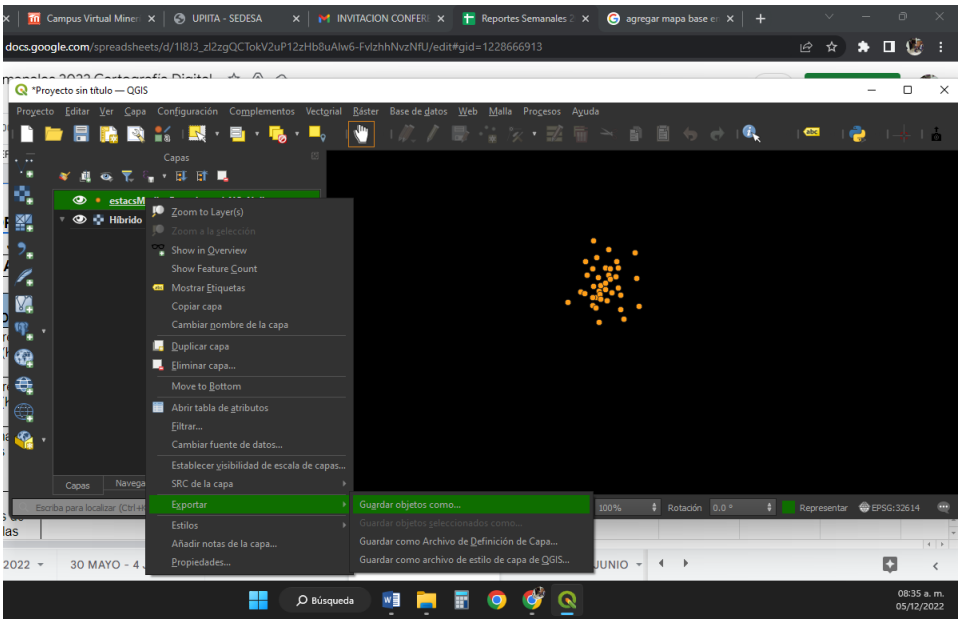
12. Agregamos un mapa base. Vamos al panel “Navegador”. En “XYZ Tiles” hacemos click derecho y seleccionamos “Conexión nueva”.



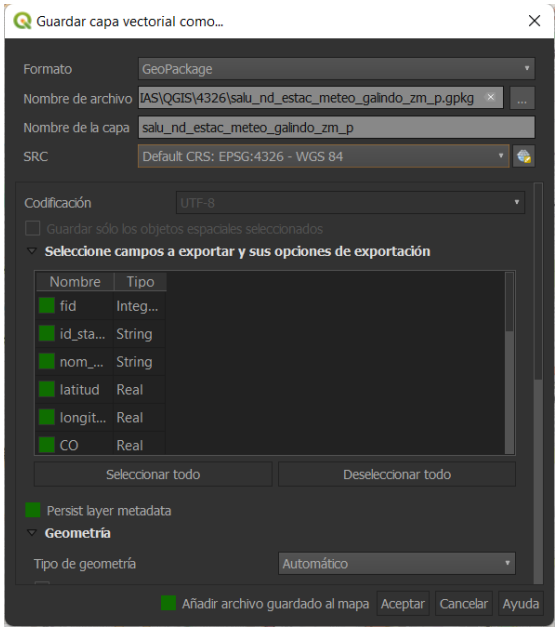
13. Realizamos la configuración de la conexión como se observa en la imagen. La url que asignamos será esta: https://mt1.google.com/vt/lyrs=y&x={x}&y={y}&z={z}



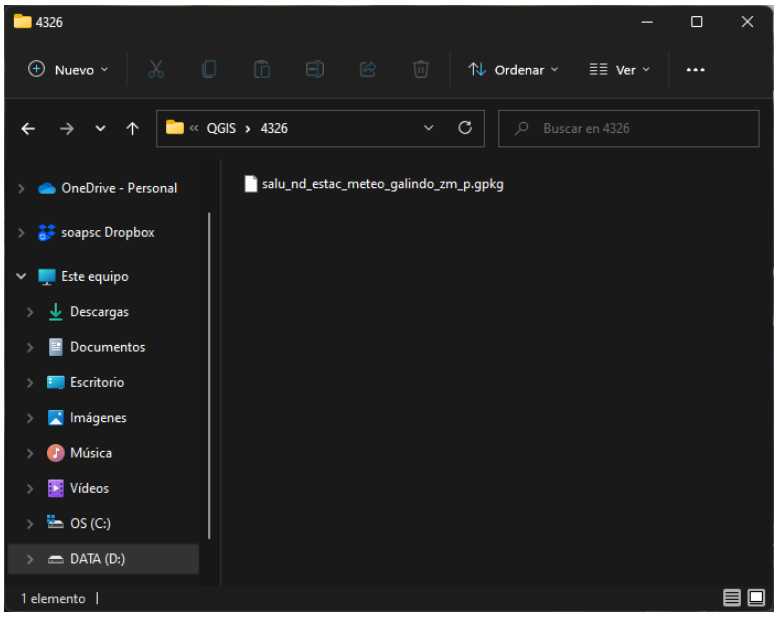
14. Exportamos la capa CSV para poder visualizarla con el Sistema de Referencia adecuado. Click derecho sobre la capa -> Exportar -> Guardar objetos como.



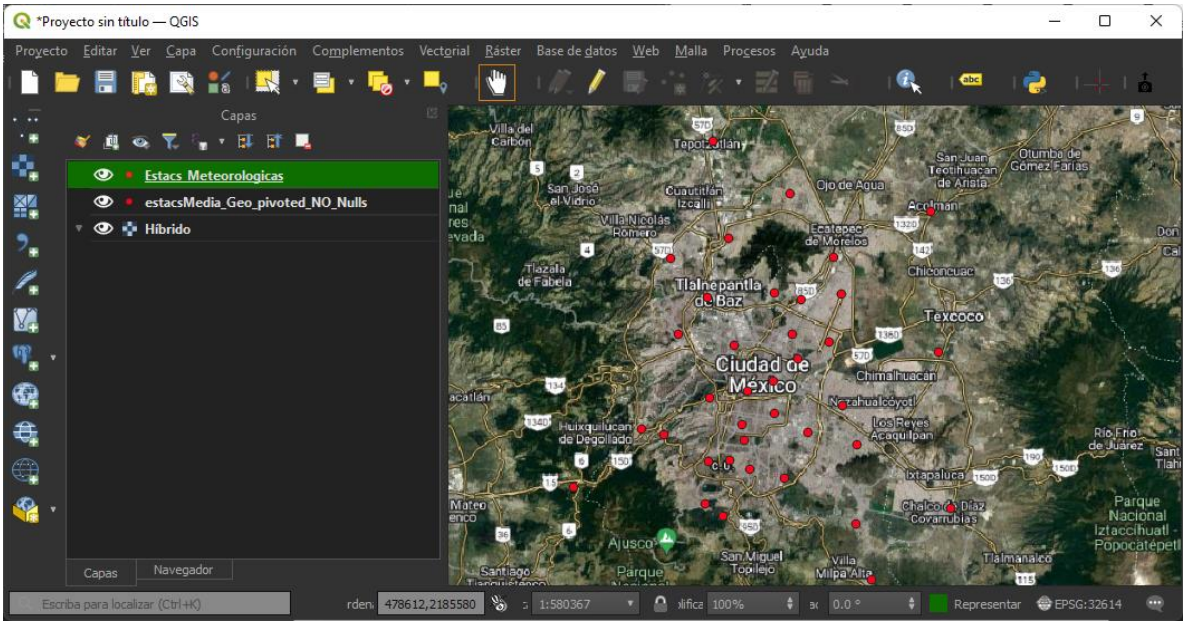
15. Configuramos la capa de la siguiente manera.



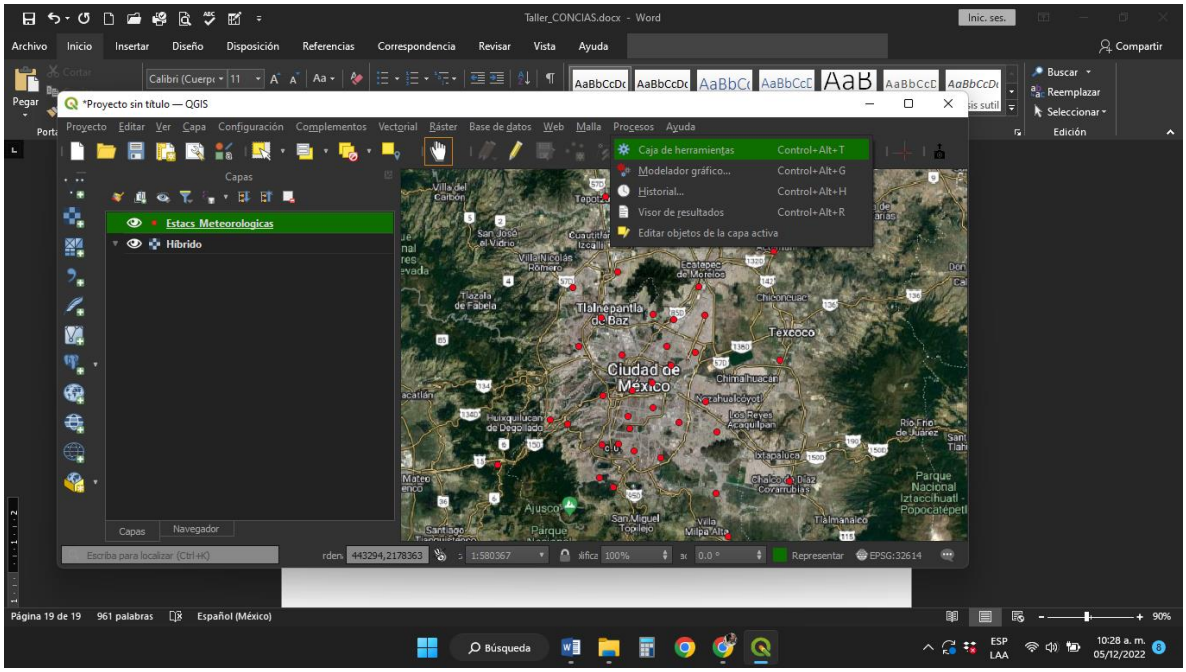
16. Con esto se generará el archivo geopackage en la carpeta.



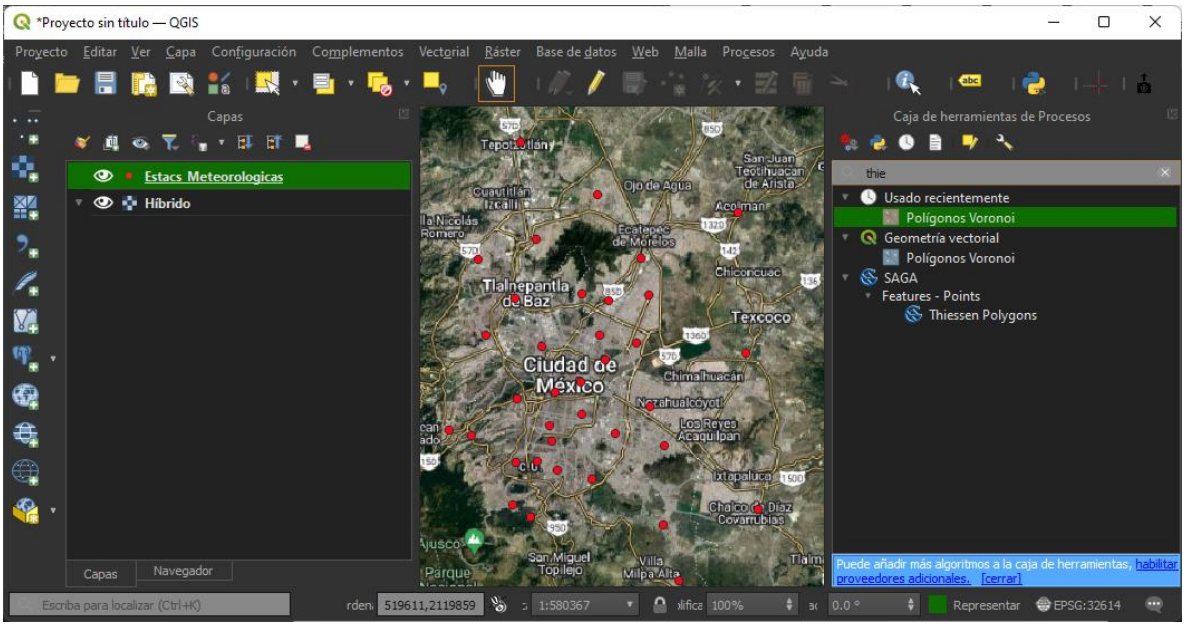
Así como en la interfaz de QGIS. Con esto ya podemos quitar el archivo CSV importado previamente.



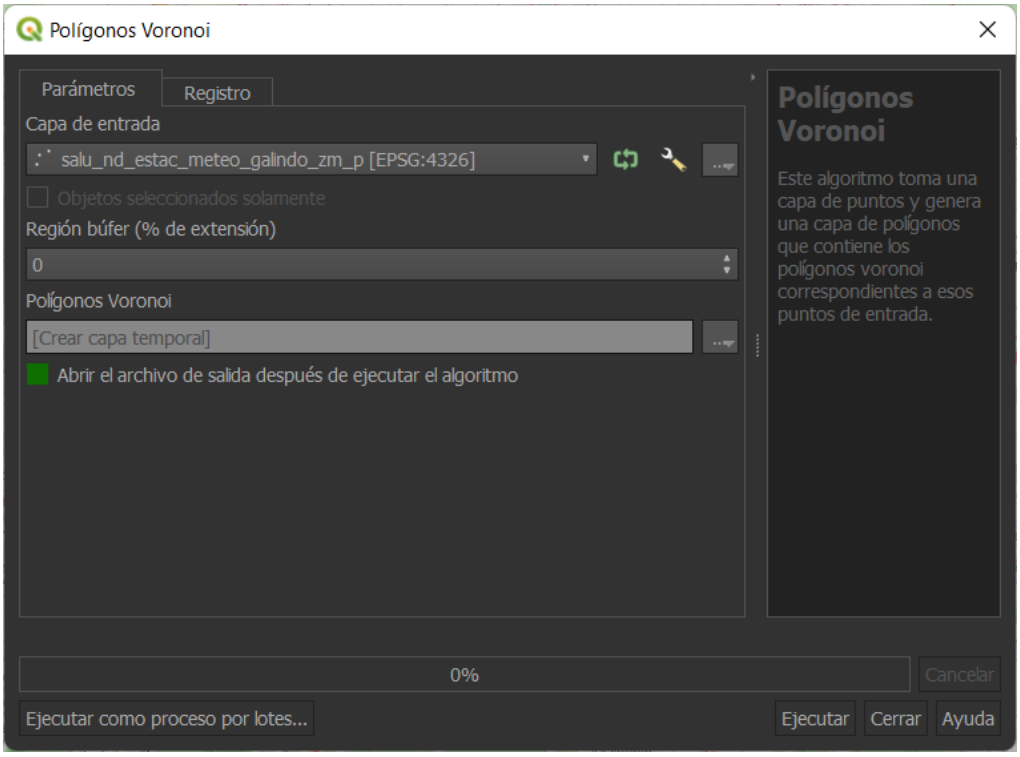
17. Abrimos la caja de herramientas para poder generar los polígonos de Voronoi (Thiessen)



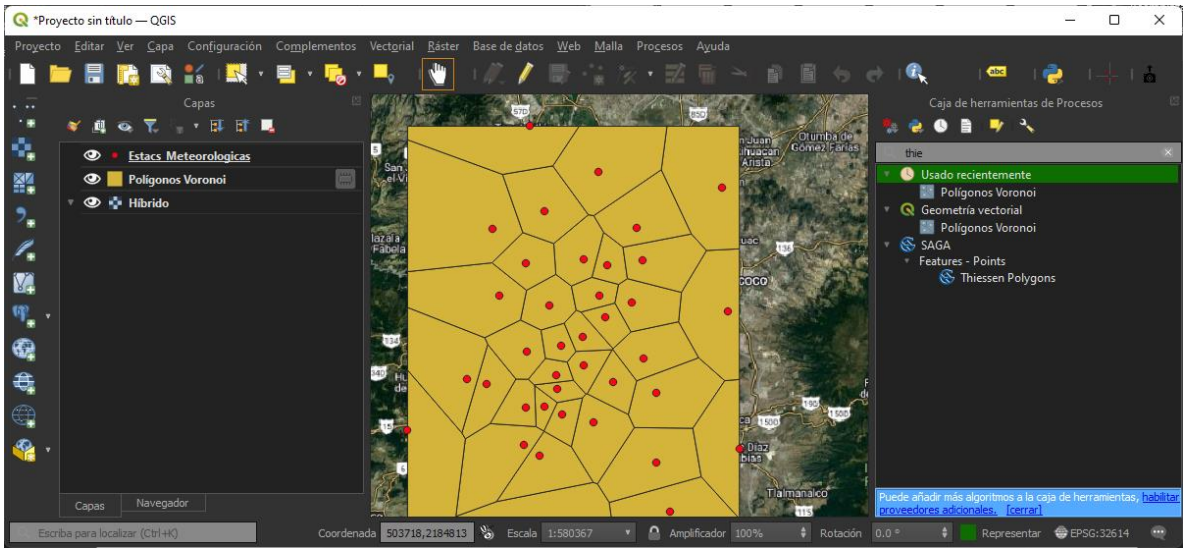
18. Aparecerá el módulo de la derecha.



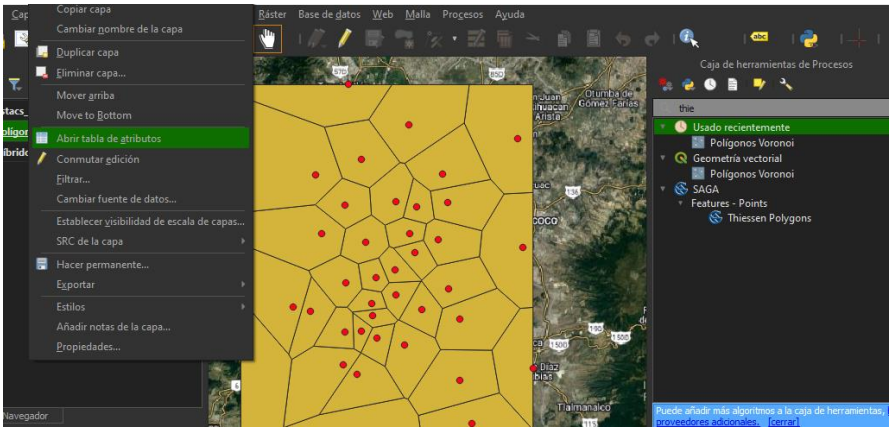
19. Al tener seleccionada la capa, hacemos doble click en la opción de Polígonos de Voronoi con lo que aparecerá la siguiente opción. Sólo seleccionamos “Ejecutar”.



20. Con esto habremos generado los Polígonos de Voronoi.



21. Con click derecho sobre la capa de polígonos podemos ver los atributos.



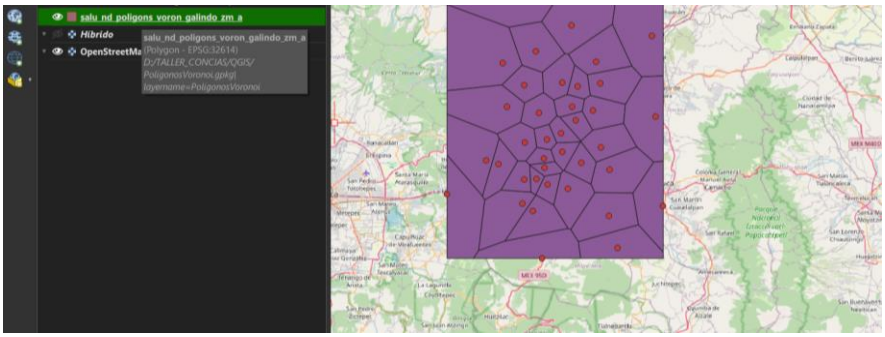
22. De esta manera vemos cómo a cada estación meteorológica le corresponde un polígono de Voronoi y éste contiene la información de cada elemento medido por dicha estación con los valores promedio.

Polígonos Voronoi— Objetos Totales: 38, Filtrados: 38, Seleccionados: 1

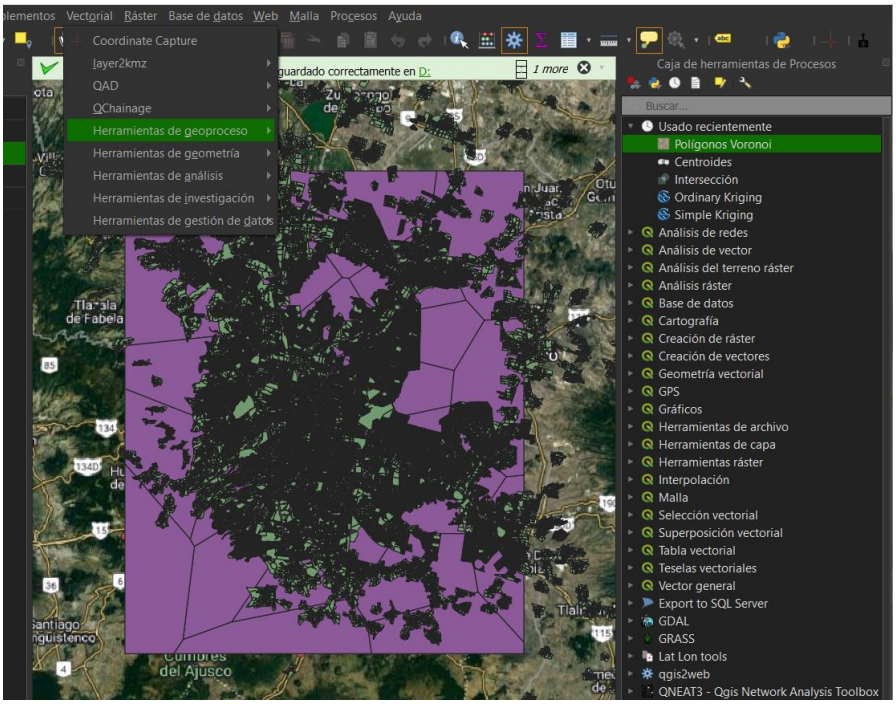
	fid	id_station	nom_estac	latitud	longitud	CO	NO	NO2	NOX	O3
1	23	MPA	Milpa Alta	19.1769	-98.990189	0.29080774	16.92981171	8.385830028	41.15324567	43.11909608
2	31	TAH	Tl◆huac	19.246459	-99.010564	0.487674068	9.244423834	18.9202449	28.15650187	34.17103125
3	3	AJU	Ajusco	19.154286	-99.162611	0.576027721	16.92981171	23.5303092	41.15324567	34.44047366
4	34	TPN	Tlalpan	19.257041	-99.184177	0.75568782	11.944663	18.58920232	30.52312764	32.00861084
5	35	UAX	UAM Xochimilco	19.304441	-99.103629	0.573042182	11.69960899	23.06694024	34.7734295	32.12493347
6	2	AJM	Ajusco Medio	19.272161	-99.207744	0.416988237	5.955789034	17.79364859	23.81745184	39.13490283
7	30	SUR	Santa Ursula	19.31448	-99.149994	0.955883786	20.28458324	29.64946184	49.91894834	28.80930337
8	7	CCA	Centro de Cien...	19.326111	-99.176111	0.508222166	11.85297222	23.37961748	35.23547099	32.33078238
9	25	PED	Pedregal	19.325146	-99.204136	0.443798858	9.61205339	22.8388828	32.42765605	33.4452325
10	16	INN	Investigaciones...	19.291968	-99.38052	0.26349003	16.92981171	23.5303092	41.15324567	37.06355951
11	28	SFE	Santa FE	19.357357	-99.262865	0.489712327	11.14726103	22.49874579	33.65323422	32.60598699
12	26	SAC	Santiago Acahu...	19.34561	-99.009381	0.408397378	10.59670078	20.44565526	31.09673226	33.55090501
13	8	CHO	Chalco	19.266948	-98.886088	0.632227574	18.4200269	22.78036183	41.29641331	29.61324536
14	9	COY	Coyoac◆n	19.350258	-99.157101	0.576027721	17.19432135	26.3709842	43.66846495	30.00844338
15	17	IZT	Iztacalco	19.384413	-99.117641	0.749303712	20.33483072	29.69902962	50.05906328	28.66323055
16	36	UIZ	UAM Iztapalapa	19.360794	-99.07388	0.65937396	16.73041268	27.24688678	43.98968169	29.24313861

Mostrar todos los objetos espaciales

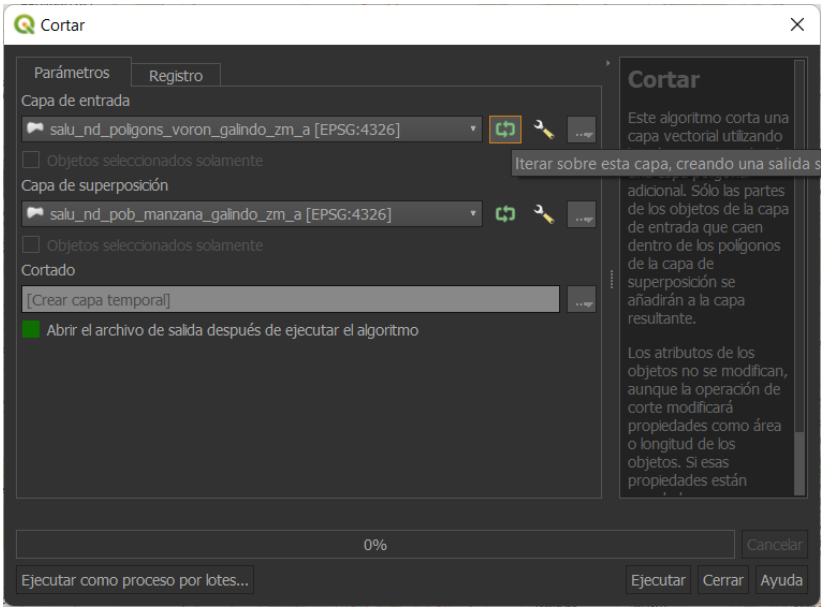
23. Guardamos la capa temporal de polígonos de Voronoi y la llamamos “salu_nd_poligons_voron_galindo_zm_a.gpkg”.



24. Aparte de los datos que contiene la capa de polígonos de Voronoi necesitamos agregarle el número de pobladores que hay en cada polígono. Para ello agregamos al mapa la capa “salu_nd_pob_manzana_galindo_zm_a.gpkg” para ejecutar un “clip” entre ambas capas. Seleccionamos la capa “salu_nd_poligons_voron_galindo_zm_a.gpkg” -> Menú Vectorial -> Herramientas de Geoproceso -> Cortar.

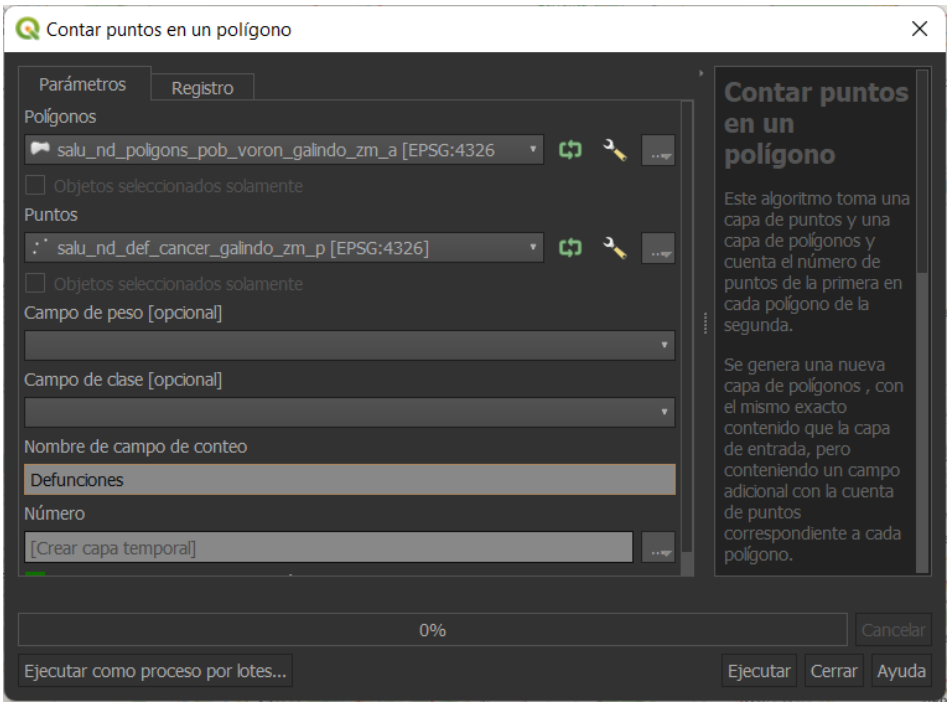


25. Asignamos los siguientes valores y hacemos click en “Ejecutar”. Es importante seleccionar que se iterará por cada polígono.

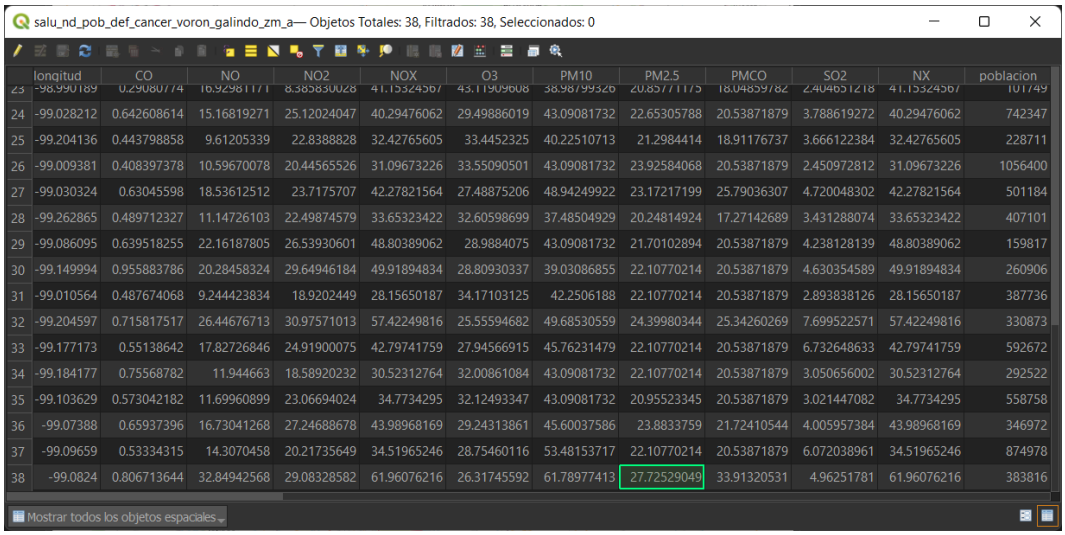


Nota: Debido a que este procesamiento lleva bastante tiempo, se proporciona la capa “salu_nd_pob_def_cancer_voron_galindo_zm_a.gpkg” la cual ya contiene un campo con la población correspondiente a cada polígono de Voronoi.

26. El siguiente paso es asignar el número de defunciones que hayan ocurrido dentro de cada polígono. Para ello activamos la capa de defunciones. En la barra de menús seleccionamos Vectorial -> Herramienta de Análisis -> Contar puntos en un polígono y aparecerá la siguiente ventana la cual debemos asigna los valores como aparecen. Ejecutamos.

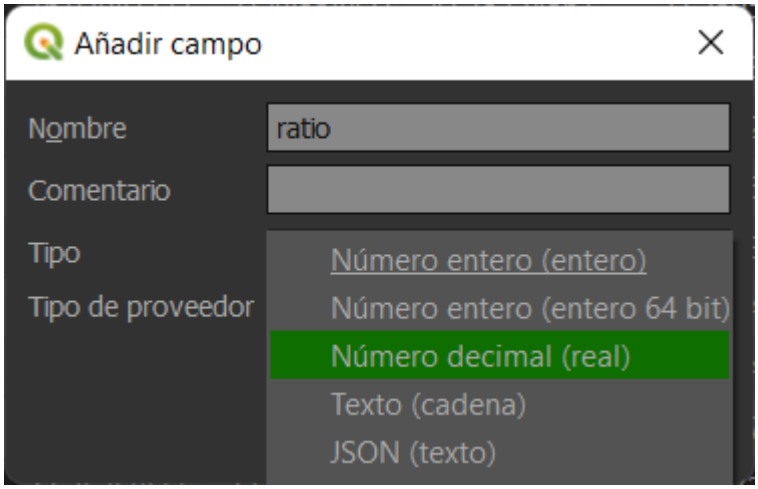


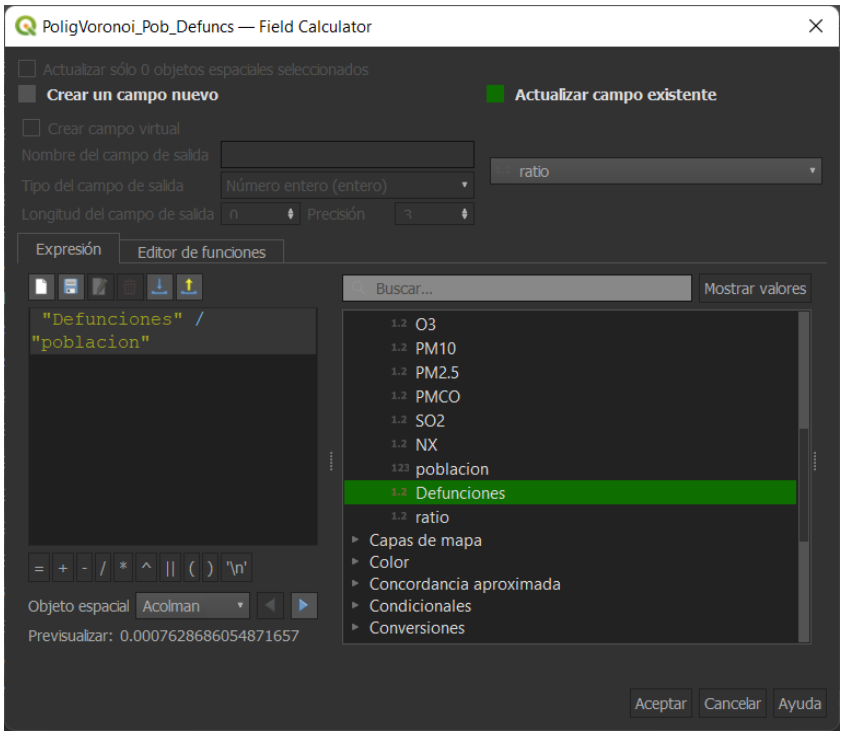
27. Se habrá creado una capa temporal que ahora contiene un campo llamado “Defunciones”. Guardamos esta capa temporal como “PoligVoronoi_Pob_Defuncs” y ya que se haya creado la capa, eliminamos la carpeta temporal “Numero”.



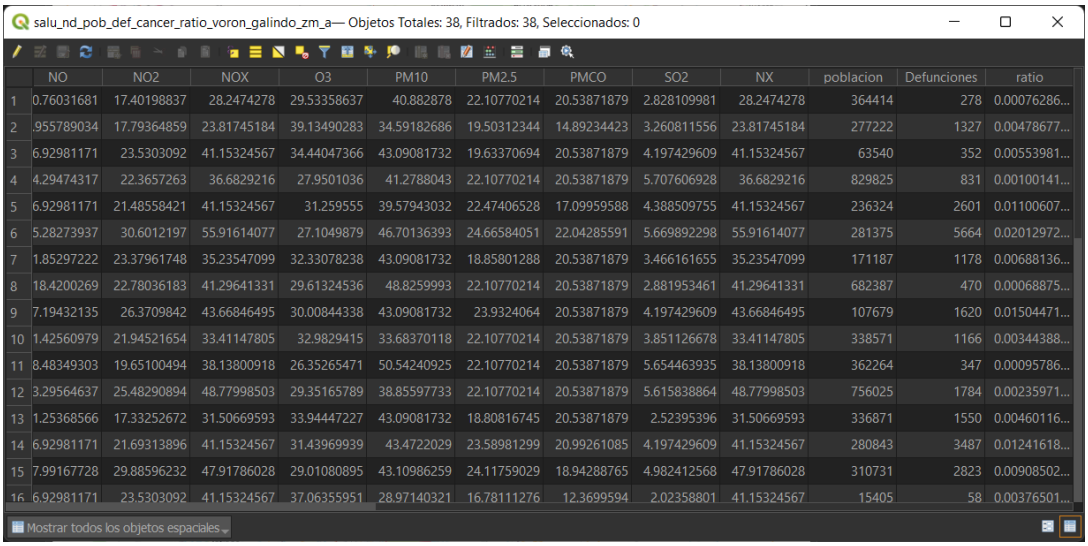
	longitud	CO	NO	NO2	NOX	O3	PM10	PM2.5	PMCO	SO2	NX	poblacion
24	-99.028212	0.642608614	15.16819271	25.12024047	40.29476062	29.49886019	43.09081732	22.65305788	20.53871879	3.788619272	40.29476062	742347
25	-99.204136	0.443798858	9.61205339	22.8388828	32.42765605	33.4452325	40.22510713	21.2984414	18.91176737	3.666122384	32.42765605	228711
26	-99.009381	0.408397378	10.59670078	20.44565526	31.09673226	33.55090501	43.09081732	23.92584068	20.53871879	2.450972812	31.09673226	1056400
27	-99.030324	0.63045598	18.53612512	23.7175707	42.27821564	27.48875206	48.94249922	23.17217199	25.79036307	4.720048302	42.27821564	501184
28	-99.262865	0.489712327	11.14726103	22.49874579	33.65323422	32.60598699	37.48504929	20.24814924	17.27142689	3.431288074	33.65323422	407101
29	-99.086095	0.639518255	22.16187805	26.53930601	48.80389062	28.9884075	43.09081732	21.70102894	20.53871879	4.238128139	48.80389062	159817
30	-99.149994	0.955883786	20.28458324	29.64946184	49.91894834	28.80930337	39.03086855	22.10770214	20.53871879	4.630354589	49.91894834	260906
31	-99.010564	0.487674068	9.244423834	18.9202449	28.15650187	34.17103125	42.2506188	22.10770214	20.53871879	2.893838126	28.15650187	387736
32	-99.204597	0.715817517	26.44676713	30.97571013	57.42249816	25.55594682	49.68530559	24.39980344	25.34260269	7.699522571	57.42249816	330873
33	-99.177173	0.55138642	17.82726846	24.91900075	42.79741759	27.94566915	45.76231479	22.10770214	20.53871879	6.732648633	42.79741759	592672
34	-99.184177	0.75568782	11.944663	18.58920232	30.52312764	32.00861084	43.09081732	22.10770214	20.53871879	3.050656002	30.52312764	292522
35	-99.103629	0.573042182	11.69960899	23.06694024	34.7734295	32.12493347	43.09081732	20.95523345	20.53871879	3.021447082	34.7734295	558758
36	-99.07388	0.65937396	16.73041268	27.24688678	43.98968169	29.24313861	45.60037586	23.8833759	21.72410544	4.005957384	43.98968169	346972
37	-99.09659	0.53334315	14.3070458	20.21735649	34.51965246	28.75460116	53.48153717	22.10770214	20.53871879	6.072038961	34.51965246	874978
38	-99.0824	0.806713644	32.84942568	29.08328582	61.96076216	26.31745592	61.78977413	27.72529049	33.91320531	4.96251781	61.96076216	383816

28. Teniendo los datos de población y número de defunciones por polígono, tenemos que obtener la densidad de fallecimientos. Para ello creamos un nuevo campo en la tabla de atributos de tipo double y mediante la calculadora de campos actualizamos el campo “ratio” con el cociente de defunciones y población.


Añadir campo
Nombre
ratio
Comentario
Tipo
Tipo de proveedor
Número entero (entero)
Número entero (entero 64 bit)
Número decimal (real)
Texto (cadena)
JSON (texto)

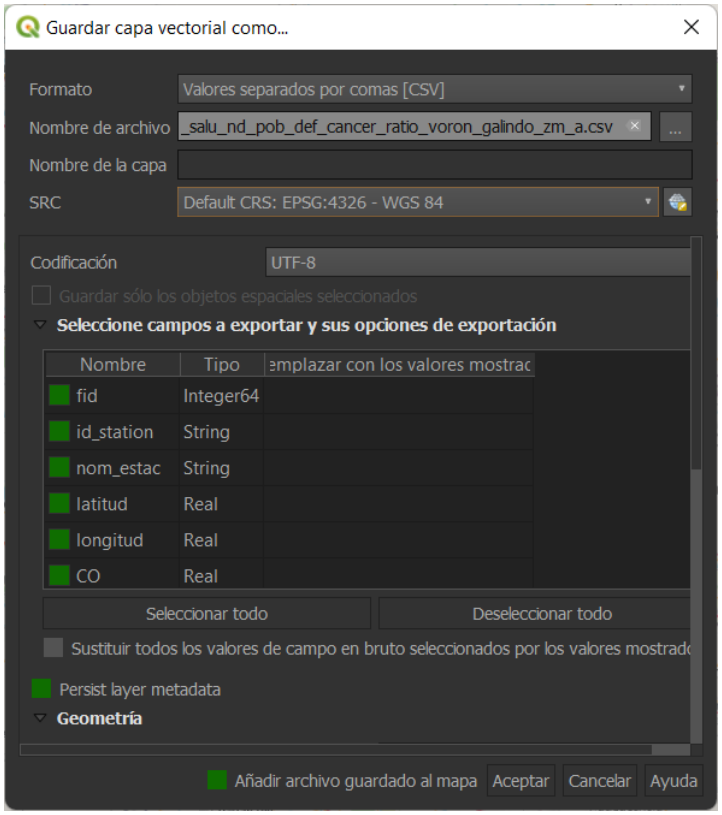

PoligVoronoi_Pob_Defuncs — Field Calculator
Actualizar sólo 0 objetos espaciales seleccionados
Crear un campo nuevo
Actualizar campo existente
Crear campo virtual
Nombre del campo de salida
Tipo del campo de salida
Número entero (entero)
Longitud del campo de salida
Precisión
ratio
Expresión
Editor de funciones
"Defunciones" / "poblacion"
Buscar...
Mostrar valores
O3
PM10
PM2.5
PMCO
SO2
NX
poblacion
Defunciones
ratio
Capas de mapa
Color
Concordancia aproximada
Condicionales
Conversiones
Objeto espacial
Acolman
Previsualizar: 0.0007628686054871657
Aceptar
Cancelar
Ayuda

29. Con esto ya tenemos la información que necesitamos para poder realizar la correlación entre el nivel de contaminantes y el número de defunciones (densidad) por cáncer en la Zona Metropolitana del Valle de México.



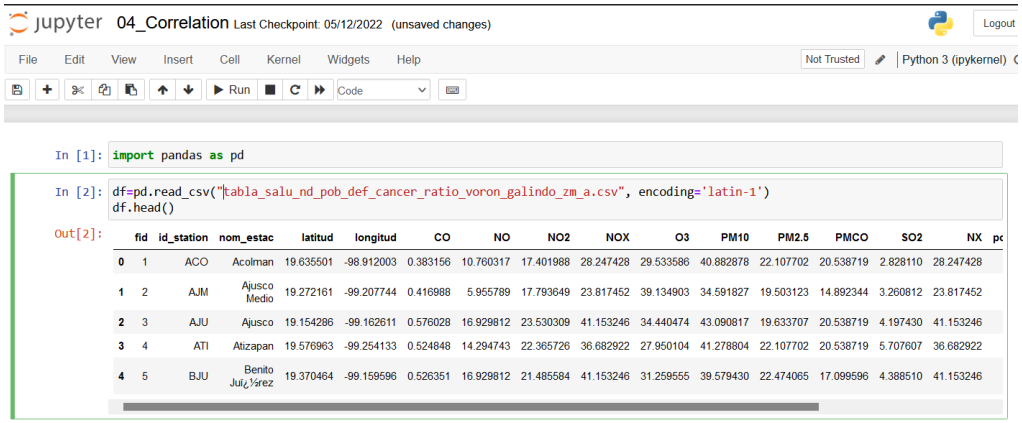
	NO	NO2	NOX	O3	PM10	PM2.5	PMCO	SO2	NX	poblacion	Defunciones	ratio
1	0.76031681	17.40198837	28.2474278	29.53358637	40.882878	22.10770214	20.53871879	2.828109981	28.2474278	364414	278	0.00076286...
2	.955789034	17.79364859	23.81745184	39.13490283	34.59182686	19.50312344	14.89234423	3.260811556	23.81745184	277222	1327	0.00478677...
3	6.92981171	23.5303092	41.15324567	34.44047366	43.09081732	19.63370694	20.53871879	4.197429609	41.15324567	63540	352	0.00553981...
4	4.29474317	22.3657263	36.6829216	27.9501036	41.2788043	22.10770214	20.53871879	5.707606928	36.6829216	829825	831	0.00100141...
5	6.92981171	21.48558421	41.15324567	31.259555	39.57943032	22.47406528	17.09959588	4.388509755	41.15324567	236324	2601	0.01100607...
6	5.28273937	30.6012197	55.91614077	27.1049879	46.70136393	24.66584051	22.04285591	5.669892298	55.91614077	281375	5664	0.02012972...
7	1.85297222	23.37961748	35.23547099	32.33078238	43.09081732	18.85801288	20.53871879	3.466161655	35.23547099	171187	1178	0.00688136...
8	18.4200269	22.78036183	41.29641331	29.61324536	48.8259993	22.10770214	20.53871879	2.881953461	41.29641331	682387	470	0.00068875...
9	7.19432135	26.3709842	43.66846495	30.00844338	43.09081732	23.9324064	20.53871879	4.197429609	43.66846495	107679	1620	0.01504471...
10	1.42560979	21.94521654	33.41147805	32.9829415	33.68370118	22.10770214	20.53871879	3.851126678	33.41147805	338571	1166	0.00344388...
11	8.48349303	19.65100494	38.13800918	26.35265471	50.54240925	22.10770214	20.53871879	5.654463935	38.13800918	362264	347	0.00095786...
12	3.29564637	25.48290894	48.77998503	29.35165789	38.85597733	22.10770214	20.53871879	5.615838864	48.77998503	756025	1784	0.00235971...
13	1.25368566	17.33252672	31.50669593	33.94447227	43.09081732	18.80816745	20.53871879	2.52395396	31.50669593	336871	1550	0.00460116...
14	6.92981171	21.69313896	41.15324567	31.43969939	43.4722029	23.58981299	20.99261085	4.197429609	41.15324567	280843	3487	0.01241618...
15	7.99167728	29.88596232	47.91786028	29.01080895	43.10986259	24.11759029	18.94288765	4.982412568	47.91786028	310731	2823	0.00908502...
16	6.92981171	23.5303092	41.15324567	37.06355951	28.97140321	16.78111276	12.3699594	2.02358801	41.15324567	15405	58	0.00376501...

30. Como último paso en QGIS exportamos la tabla de atributos como archivo CSV para poderlo procesar en Python.



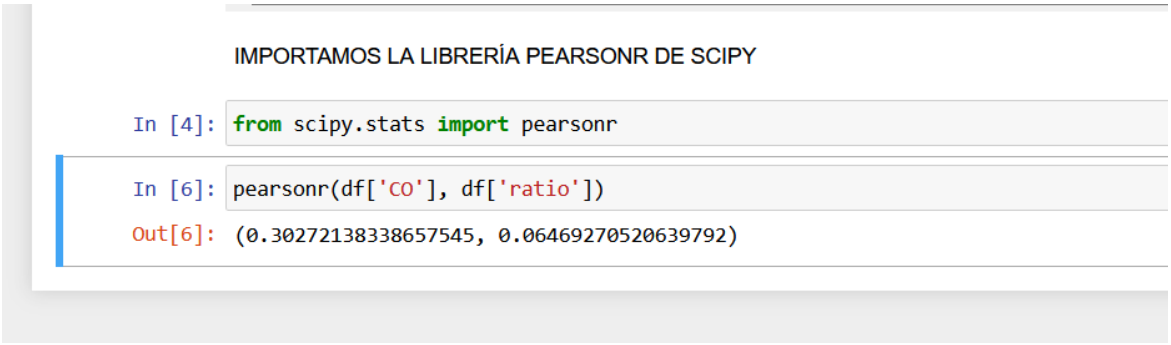
CORRELACIÓN GEOGRÁFICA

1. Creamos un nuevo notebook llamado “04_Correlation” e importamos el archivo “PoligVoronoi_Pob_Def_Ratio.csv” como se muestra en la siguiente imagen.



	fid	id_station	nom_estac	latitud	longitud	CO	NO	NO2	NOX	O3	PM10	PM2.5	PMCO	SO2	NX	pc
0	1	ACO	Acolman	19.635501	-98.912003	0.383156	10.760317	17.401988	28.247428	29.533586	40.882878	22.107702	20.538719	2.828110	28.247428	
1	2	AJM	Ajusco Medio	19.272161	-99.207744	0.416988	5.955789	17.793649	23.817452	39.134903	34.591827	19.503123	14.892344	3.260812	23.817452	
2	3	AJU	Ajusco	19.154286	-99.162611	0.576028	16.929812	23.530309	41.153246	34.440474	43.090817	19.633707	20.538719	4.197430	41.153246	
3	4	ATI	Atizapan	19.576963	-99.254133	0.524848	14.294743	22.365726	36.682922	27.950104	41.278804	22.107702	20.538719	5.707607	36.682922	
4	5	BJU	Benito Juí¿½rez	19.370464	-99.159596	0.526351	16.929812	21.485584	41.153246	31.259555	39.579430	22.474065	17.099596	4.388510	41.153246	

2. Importamos la librería “Pearsonr” de Scipy y generamos nuestra primera correlación entre la media de CO en cada polígono de Voronoi y la densidad de defunciones (ratio) en cada polígono como se observa en la imagen. El resultado nos muestra como primer valor la correlación y como segundo valor el p-valor.



3. Realizamos la misma operación para los contaminantes faltantes.

```
In [7]: pearsonr(df['NO'], df['ratio'])
Out[7]: (0.25866812981624765, 0.1168593606438872)

In [8]: pearsonr(df['NO2'], df['ratio'])
Out[8]: (0.4985339905942887, 0.001444661823247135)

In [9]: pearsonr(df['NOX'], df['ratio'])
Out[9]: (0.4169889069750331, 0.009201867990165107)

In [10]: pearsonr(df['O3'], df['ratio'])
Out[10]: (-0.10033917423613892, 0.5489155688293377)

In [11]: pearsonr(df['PM10'], df['ratio'])
Out[11]: (-0.07920020773226746, 0.6364576241556921)

In [12]: pearsonr(df['PM2.5'], df['ratio'])
Out[12]: (0.24597005028255678, 0.1365958684018983)

In [13]: pearsonr(df['PMCO'], df['ratio'])
Out[13]: (-0.060504171581754695, 0.7182157343132777)

In [14]: pearsonr(df['SO2'], df['ratio'])
Out[14]: (0.07035214063308765, 0.674694878770963)
```

CONCLUSIONES

Como se puede observar en el resultado de la correlación de Pearson en Python, podemos concluir que existe una correlación positiva entre los índices de los contaminantes NO_2 y NO_X a un nivel de significancia mucho menor al 1 %.

```
In [8]: pearsonr(df['NO2'], df['ratio'])
Out[8]: (0.4985339905942887, 0.001444661823247135)

In [9]: pearsonr(df['NOX'], df['ratio'])
Out[9]: (0.4169889069750331, 0.009201867990165107)
```