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", <u>https://www.anaconda.com/</u>



2. Elegimos nuestra plataforma: Windows, Mac o Linux

	Anaconda Installe	rs
Windows 📲	MacOS 🗯	Linux 🛆
Python 3.9	Python 3.9	Python 3.9
64-Bit Graphical Installer (621 MB)	64-Bit Graphical Installer (688 MB)	64-Bit (x86) Installer (737 MB)
	64-Bit (M1) Graphical Installer (484 MB)	MB)
	64-Bit (M1) Command Line Installer (472 MI	64-Bit (AWS Graviton2 / ARM64) Installer (534 MB)
	What can we use data for?	4 Z & LinuxONE) Installer
Only Ess	entials Functional Analytics	Ads and Analytics Have any questions?
This website uses cookies to		Let's get you connect to an expert!

- 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.

	,О _а г	naconda							/ a
T4	← (Todo Aplicaciones Document	os (Web	Configuración	Personas	►	С	it" h h
	Mejor	coincidencia							ara
		naconda Prompt (Anaconda3) plicación				CA			
	Aplica	ciones			Anaconda	Prompt (Ana	conda3))	
	0	naconda Navigator (Anaconda3)	>			Aplicacion			
nb nb	م ب	naconda Powershell Prompt Anaconda3)			Abrir				
aci b	С Л	upyter Notebook (Anaconda 3)		6	Ejecutar como ad Abrir ubicación d	ministrador e archivo			
2.с	🔀 S	pyder (Anaconda 3)			Anclar a Inicio				
3.c 4 -	R (/	eset Spyder Settings Anaconda3)			Anclar a la barra o Desinstalar	de tareas			
	Buscar	en Internet							
Ι.	,О а	naconda - Ver resultados web							
	,С а	naconda python							

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.

Archivo Editar Ver Historial Marcadores Herramiențas Ayuda C Home Page - Select or create a X +	-	· Ø	× ~
$\leftarrow \rightarrow \mathbf{C}$ O D localhost:8889/tree	☆ ©	• 4	€ ≡
💭 Jupyter	Quit Logout		
Files Running Clusters			
Select items to perform actions on them.	Upload New- 2		
	Name Last Modified File size		
Taller_CONCIAS.docx	hace unos segundos 134 kB		

7. Creamos un nuevo Notebook.

<u>Archivo Editar Ver Historial Marcadores Herramiențas Ayuda</u>			- c	7	\times
C Home Page - Select or create a X +					~
$\leftarrow \rightarrow \mathbf{C}$ O D localhost.8889/tree	☆	${f igodot}$	•	4	=
💭 Jupyter	Quit	Logout			
Files Running Clusters					
Select items to perform actions on them.	Upload	New - 2			
□ 0 ▼ ■/ Name	 Notebook: Python 3 (ipyke) 	ernel) :e			
Taller_CONCIAS.docx	Other:	ИB			
	Text File				
	Folder				
	Terminal				

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.

Archivo <u>E</u> ditar <u>V</u> er <u>H</u> istorial <u>M</u> arcadores Herramien <u>t</u> as Ayuda		_	
C Home Page - Select or create a × 🧧 01_web_crawler - Jupyter Noteb × +			~
$\leftarrow \rightarrow \mathbf{C}$ \bigcirc \bigcirc b localhost:8889/notebooks/01_web_crawler.ipynb	ជ	⊌ 🍕	∉ ≡
Jupyter 01_web_crawler (autosaved)		2	Logout
File Rename Notebook			× 1) O
Enter a new notebook name:			
01_web_crawler			
	Cancel	Rename	

9. Se habrá cambiado el nombre del notebook.

<u>A</u> rchivo <u>E</u> ditar <u>V</u> er <u>H</u> istorial <u>M</u> arcadores Herramien <u>t</u> as A <u>v</u> uda	- 🗆 ×
C Home Page - Select or create a × 🧧 01_web_crawler - Jupyter Noteb × +	~
$\leftarrow \rightarrow \mathbf{C}$ O D localhost:8889/notebooks/01_web_crawler.ipynb	⊠ 🔹 🧉 ≡
Jupyter 01_web_crawler (autosaved)	Cogout
File Edit View Insert Cell Kernel Widgets Help Trusted	Python 3 (ipykernel) O
P → % 2	
In []:	

10. Instalamos Beautiful Soup con el comando "pip install beautifulsoup4"



11. Importamos las librerías "BeautifulSoup" y "request"

In [5]:	from bs4 import BeautifulSoup as bs
	import requests

- 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,'ht ml.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.

TALLER_CONCIAS				– o ×
🕀 Nuevo ~ 🔏 🔲	î E) & Ū ∿·	Ordenar ~ 🗮 Ver ~		
$\leftarrow \rightarrow \sim \uparrow$ \simeq Est	e equipo > DATA (D:) > TALLER_CONCIAS		~ C	
> 🛧 Acceso rápido	Nombre	Fecha de modificación	Тіро	Tamaño
/ Access replace	ipynb_checkpoints	04/12/2022 01:15 p. m.	Carpeta de archivos	
> 🌰 OneDrive - Personal	🚞 contaminantes	03/12/2022 10:32 p. m.	Carpeta de archivos	
> 🗦 soapsc Dropbox	🚞 procesar	04/12/2022 12:45 p. m.	Carpeta de archivos	
🗸 💻 Este equipo	01_web_crawler.ipynb	03/12/2022 10:47 p. m.	Archivo de origen	30 KB
> V Descargas	02_quitar_nulos.ipynb	04/12/2022 01:24 p. m.	Archivo de origen	2 KB
> Documentos	🐞 contaminantes_2012.csv	03/12/2022 10:42 p. m.	Archivo de valores	51,538 KB
Escritorio	🔊 contaminantes_2013.csv	03/12/2022 10:41 p.m.	Archivo de valores	54,041 KB
	🔊 contaminantes_2014.csv	03/12/2022 10:40 p. m.	Archivo de valores	54,724 KB
Márica	🔊 contaminantes_2015.csv	03/12/2022 10:40 p. m.	Archivo de valores	59,380 KB
	🔊 contaminantes_2016.csv	03/12/2022 10:39 p. m.	Archivo de valores	62,988 KB
	🔊 contaminantes_2017.csv	03/12/2022 10:38 p. m.	Archivo de valores	63,344 KB
> 🖆 05 (C:)	🔊 contaminantes_2018.csv	03/12/2022 10:37 p.m.	Archivo de valores	63,253 KB
> 🚍 DAIA (D:)	🔊 contaminantes_2019.csv	03/12/2022 10:36 p. m.	Archivo de valores	67,202 KB
> 🎦 Red	🔊 contaminantes_2020.csv	03/12/2022 10:35 p. m.	Archivo de valores	67,639 KB
	🕼 contaminantes_2021.csv	03/12/2022 10:34 p.m.	Archivo de valores	67,693 KB
	a contaminantes_2022.csv	03/12/2022 10:33 p. m.	Archivo de valores	56,513 KB
18 elementos 1 elemento selecc	ionado 55.1 MB			

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.

contaminantes_2021.csv	03/12/2022 10:34 p. m.	Archivo de valores	67,693 KB
a contaminantes_2022.csv	03/12/2022 10:33 p. m.	Archivo de valores	56,513 KB
🔊 noNulls_contaminantes_2012.csv	04/12/2022 12:45 p. m.	Archivo de valores	56,507 KB
Taller_CONCIAS.docx	04/12/2022 11:33 a.m.	Documento de Mi	563 KB

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.



 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).



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

1:	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	N02	18.567919	1.0	
3	2012-01-31	ACO	NOX	30.674224	1.0	
4	2012-01-31	ACO	03	26.706664	1.0	
]: ler	n(df_temp)					

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

$- \rightarrow \sim \uparrow \square$	Este equipo > DATA (D:) > TALLER_CONCIAS > noNull_Me	в 		~ C	
🛧 Acceso rápido	Nombre	Fecha de modificación	Тіро	Tamaño	
left oneDrive - Personal	noNullsMes_noNulls_contaminantes_2012.csv	04/12/2022 02:51 p.m.	Archivo de valores	99 KB	
🔐 soapsc Dropbox	noNullsMes_noNulls_contaminantes_2013.csv	04/12/2022 03:51 p.m.	Archivo de valores	103 KB	
💻 Este equipo	noNullsMes_noNulls_contaminantes_2014.csv	04/12/2022 03:51 p.m.	Archivo de valores	105 KB	
🛓 Descargas	noNullsMes_noNulls_contaminantes_2015.csv	04/12/2022 03:52 p.m.	Archivo de valores	114 KB	
Documentos	noNullsMes_noNulls_contaminantes_2016.csv	04/12/2022 03:52 p.m.	Archivo de valores		
Escritorio	noNullsMes_noNulls_contaminantes_2017.csv	04/12/2022 03:52 p.m.	Archivo de valores	122 KB	
Música	noNullsMes_noNulls_contaminantes_2018.csv	04/12/2022 03:52 p.m.	Archivo de valores	122 KB	
Vídeos	noNullsMes_noNulls_contaminantes_2019.csv	04/12/2022 03:52 p.m.	Archivo de valores	130 KB	
늘 OS (C:)	noNullsMes noNulls contaminantes 2020.csv	04/12/2022 03:52 p.m.	Archivo de valores_	131 KB	
🚍 DATA (D:)					
📜 Red					

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

ile Edit	View	Insert Cell	Kernel W	/idgets H	lelp					Т	rusted	ø	Pytho
+ * 4	b 🖪	↑ ↓ Run	■ C ₩	Code	× 🖼								
Out[56]:	2371												
	CONC	ATENAMOS TODO	S LOS ARCH	IVOS OBTE	NIDOS PARA	TENER U	N ÚNICO	ARCHIVO					
In [65]:	import import	glob os											
In [67]:	files	= os.path.join("noNull_Mes,	/", "*.csv	") #Estable	cemos rut	a donde	se encuent	ran los arci	rivos			
In [68]:	files	<pre>= glob.glob(fil</pre>	es) #Librer	ía que pue	de manipula	r archive	s del mi	smo tipo					
In [68]: In [70]:	files df_con print(= glob.glob(fil cat = pd.concat df_concat)	es) <i>#Librer</i>	ía que pue d_csv, fil	de manipula es), ignore	r archive	os del mi rue) #con	smo tipo catenamos	los archivos	s y Leemos el	resu	Ltad	0
In [68]: In [70]:	files df_con print(= glob.glob(fil cat = pd.concat df_concat) date id_	es) #Librer (map(pd.read	<i>ía que pue</i> d_csv, fil parameter	de manipula es), ignore value	unit	es del mi	ismo tipo ncatenamos	Los archivos	s y Leemos el	. resu	Ltad	0
In [68]: In [70]:	files df_con print(= glob.glob(fil cat = pd.concat df_concat) date id_ 2012-01-31	es) #Librer (map(pd.read station id_ ACO	<i>ía que pue</i> d_csv, fil parameter CO	de manipula es), ignore value 0.611230	unit 15.0	es del mi	smo tipo acatenamos	los archivos	s y Leemos el	. resu	Ltad	0
In [68]: In [70]:	files df_con print(0 1	<pre>= glob.glob(fil cat = pd.concat df_concat)</pre>	es) #Librer (map(pd.read station id_ ACO ACO	<i>ía que pue</i> d_csv, fil parameter CO NO	de manipula es), ignore value 0.611230 12.097804	unit 15.0 1.0	os del mi rue) #con	smo tipo acatenamos	los archivos	s y Leemos el	. resu	Ltad	D
In [68]: In [70]:	files df_con print(0 1 2	= glob.glob(fil cat = pd.concat df_concat) date id_ 2012-01-31 2012-01-31 2012-01-31	es) #Librer (map(pd.read station id_ ACO ACO ACO	ia que pue d_csv, fil parameter CO NO NO2	de manipula es), ignore value 0.611230 12.097804 18.567919	unit 15.0 1.0 1.0	os del mi	smo tipo	los archivos	s y Leemos el	. resu	Ltad	D
In [68]: In [70]:	files df_con print() 0 1 2 3 4	= glob.glob(fil cat = pd.concat df_concat) date id_ 2012-01-31 2012-01-31 2012-01-31 2012-01-31 2012-01-31	es) #Librer (map(pd.read station id_ ACO ACO ACO ACO ACO	<pre>ía que pue d_csv, fil parameter CO NO2 NO2 NO3</pre>	de manipula es), ignore 0.611230 12.097804 18.567919 30.674224 26.706564	unit 15.0 1.0 1.0 1.0	os del mi	smo tipo	Los archivos	s y Leemos el	. resu	Ltad	D
In [68]: In [70]:	files df_con print(0 1 2 3 4	<pre>= glob.glob(fil cat = pd.concat df_concat)</pre>	es) #Librer (map(pd.read station id_ ACO ACO ACO ACO ACO ACO	<pre>ia que pue d_csv, fil parameter CO NO NO2 NOX O3</pre>	de manipula es), ignore 0.611230 12.097804 18.567919 30.674224 26.706664	unit 15.0 1.0 1.0 1.0 1.0	es del mi	smo tipo	Los archivos	s y leemos el	. resu	Ltad	D
In [68]: In [70]:	files df_con print(0 1 2 3 4 25070	<pre>= glob.glob(fil cat = pd.concat df_concat)</pre>	es) #Librer (map(pd.read station id_ ACO ACO ACO ACO ACO ACO ACO ACO ACO ACO	<pre>ia que pue d_csv, fil parameter CO NO NO2 NOX O3 03</pre>	de manipula es), ignore value 0.611230 12.097804 18.567919 30.674224 26.706664 27.843763	unit 15.0 1.0 1.0 1.0 1.0 1.0 1.0	es del mi	smo tipo	Los archivos	s y leemos el	. resu	Ltad	D
In [68]: In [70]:	files df_con print(0 1 2 3 4 25070 25071	<pre>= glob.glob(fil) cat = pd.concat df_concat)</pre>	es) #Librer (map(pd.read station id_] ACO ACO ACO ACO ACO ACO ACO ACO ACO ACO	ia que pue d_csv, fil parameter CO NO2 NO2 NOX O3 O3 PM10	de manipula es), ignore value 0.611230 12.097804 18.567919 30.674224 26.706664 27.843763 47.325384	unit 15.0 1.0 1.0 1.0 1.0 1.0 2.0	es del mi	ismo tipo neatenamos	Los archivos	s y leemos el	. resu	Ltade	D
In [68]: In [70]:	files df_con print() 0 1 2 3 4 25070 25071 25072	<pre>= glob.glob(fil) cat = pd.concat df_concat)</pre>	(map(pd.read station id_ ACO ACO ACO ACO ACO ACO ACO ACO ACO XAL XAL XAL	ia que pue d_csv, fil parameter CO NO NO2 NO3 O3 PM10 PM12.5	de manipula es), ignore 0.611230 12.097804 18.567919 30.674224 26.70644 27.843763 47.325384	unit 15.0 1.0 1.0 1.0 1.0 1.0 2.0 2.0	s del mi	smo tipo ncatenamos	los archivos	s y leemos el	. resu	Ltad	D
In [68]: In [70]:	files df_con print() 0 1 2 3 4 25070 25071 25072 25073	<pre>= glob.glob(fil cat = pd.concat df_concat)</pre>	es) #Librer (map(pd.read station id_ ACO ACO ACO ACO ACO ACO ACO ACO ACO ACO	ia que pue d_csv, fil parameter CO NO2 NO2 NO3 O3 O3 PM10 PM12.5 PMC0	de manipula es), ignore 0.611230 12.097804 18.567919 30.674224 26.706664 27.843763 47.325384 21.758440 21.485654	r archive index=Tr 15.0 1.0 1.0 1.0 1.0 1.0 1.0 2.0 2.0 2.0	is del mi	ismo tipo acatenamos	Los archivos	s y leemos el	. resu	Ltad	D
In [68]: In [70]:	files df_con print(0 1 2 3 4 25070 25071 25072 25073 25074	<pre>= glob.glob(fil cat = pd.concat df_concat) date id 2012-01-31 2012-01-31 2012-01-31 2012-01-31 2012-01-31 2012-01-31 2020-12-31 2020-12-31 2020-12-31</pre>	es) #Librer (map(pd.read station id_ ACO ACO ACO ACO ACO ACO ACO ACO ACO ACO	ia que pue d_csv, fil parameter CO NO2 NO2 O3 03 PM10 PM2.5 PMC0 SO2	de manipula es), ignore 0.611230 12.097804 18.567919 30.674224 26.706664 27.843763 47.325384 21.485654 4.200645	r archive index=Tr unit 15.0 1.0 1.0 1.0 1.0 1.0 2.0 2.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1	is del mi	ismo tipo acatenamos	Los archivos	; y leemos el	. resu	Ltad	D

 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.

In [100]:	df_con print(catenado = p df_concatena	d.concat(m do)	ap(pd.read_csv	, files), i	gnore_index	x=True) #concatenamos los archivos y leemos el resultado
		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	03	26.706664	1.0	
	25070	2020-12-31	XAL	03	27.843763	1.0	
	25071	2020-12-31	XAL	PM10	47.325384	2.0	
	25072	2020-12-31	XAL	PM2.5	21.758440	2.0	
	25073	2020-12-31	XAL	PMCO	21.485654	2.0	
	25074	2020-12-31	XAL	502	4.200645	1.0	
	[25075	rows x 5 co	lumns]				
In [101]:	#df_co	ncatenado.dn concatenado	op (column [df concat	s = df_concate enado.columns[nado.column 0]]	[0], axis	= 1, inplace = True)

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"

istorial Marcadores Herramientas Apuda or oreste s × 🦉 03.geolocalizar estaciones y, s × 🏾 🖉 02. quitar nulos - Jupiter Notes × 🛛 🦉 01. web. crawler - Jupiter Notes × +	
C D Iocalhost 8889/notebooks/03.geolocalizar_estaciones_y_valores.jpynb	☆
Jupyter 03_geolocalizar_estaciones_y_valores Last Checkpoint: hace 2 minutos (autosaved)	nt Logout
File Edit View Insert Cell Kernel Widgets Help	Trusted Python 3 (ipykernel) O
E + 3< 2 K ↑ ↓ ► Run ■ C ► Code ∨	
In []:	

2. Abrimos el archivo generado en la fase anterior.



 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"

Archivo Editar Ver Historial M	arcadores Herramie	entas Ay	(uda							-	Ø)	× .
C Home Page - Select or create a	× 🦉 03_geoloc		taciones_y_	ya x 😑 d	2_quitar_nulo	s - Jupyt	er Notel: × 🛛 🖉 01_web_crawler - Jupyter Notel: × 🛛 🌞 Nueva pestaña					
$\leftarrow \ \rightarrow \ \mathbf{G}$		calhost:8							☆		. 🤞 🗉	Ξ
	💭 jupyter	03_	geoloca	alizar_es	taciones	_y_va	IOTES Last Checkpoint: hace 41 minutos (autosaved)		Cogout Logout			
	File Edit	View	Insert	Cell	Kernel	Widgets	Help	Trusted	Python 3 (ipykernel) O			
	B + × 6	ბ ნ	↑ ↓	► Run	C >>	Code	 Image: A state of the state of					
												ī
	In [1]:	impor	t pandas	s as pd								I
	In [4]:	df=pd	.read_c	sv("noNull	_Mes/medi	a_2010	2020_noNulls _sinColumna.csv", encoding='latin-1')					I
		ur.ne	au()									I
	Out[4]:	id	_station i	d_parameter	value	unit						I
		0	ACO	CO	0.611230	15						I
		1	ACO	NO	12.097804	1						I.
		2	ACO	NO2	18.567919	1						1
		3	ACO	NOX	30.674224	1						1
		4	ACO	03	26.706664	1						I.
	In [5]:	len(d	f)									1
	Out[5]:	25075										I
	In [7]:	df_gr	ped = dt	f.groupby(['id_stat	ion','	id_parameter'])					I
	In [8]:	df_gr	ped['val	lue'].agg(['mean'])	.to_cs	v('noNull_Mes/estacionesConMediaPorParametro.csv')					I
	·											1
												I.

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.

I	. 5.0	- 💪 🗃 -							cat_es	tacion.csv -
Are	hivo Inicio	Insertar D	Diseño de págir	a Fórmulas	Datos F	levisar Vista	Ayuda 🎧	Qué desea haر	er?	
ſ	Cortar	Calib	ri - 1	1 - A A		P - ab Aju	star texto	General	*	
Peg	ar	ormato N	к <u></u> .	<u>ð</u> - <u>A</u> -	= = = 3	Con	nbinar y centrar	- \$ - % 001	•,0 •,0	Formato
	Portapapeles	5	Fuente	5		Alineación		15 Número	5	
0	POSIBLE PÉR	DIDA DE DATO	S Algunas cara	cterísticas del	libro se pueder	n perder si lo gu	arda como CSV	(delimitado por	comas). Pai	ra conservar
A1		: × •	√ fx C	atalogo de es	taciones					
4	A	В	С	D	E	F	G	н	1	J
1	Catalogo de	staciones								
2	cve_estac	nom_estac	longitud	latitud	alt	obs_estac	id_station			
3	ACO	Acolman	-98.912003	19.635501	2198		4.8415E+11			
4	AJU	Ajusco	-99.162611	19.154286	2942		4.8409E+11			
5	AJM	Ajusco Medic	-99.207744	19.272161	2548		4.8409E+11			
6	ARA	Aragón	-99.074549	19.470218	2200	Finalizó oper	4.8409E+11			
7	ATI	Atizapan	-99.254133	19.576963	2341		4.8415E+11			

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.

5	+ 34 0	··	↓ Run	C #	Code				
		Agregamos	el archivo del o	atálogo de	estaciones	s meteo	rológicas		
	In [13]:	df_cat_sta df_cat_sta	tions=pd.reations.head()	d_csv("co	ontaminan	tes/cat	t_estacion.csv", enc	oding='latin-1')	
	Out[13]:	id_station	n nom_estac	longitud	latitud	alt	obs_estac	id_station.1	
		0 ACC	O Acolman	-98.912003	19.635501	2198.0	NaN	4.840000e+11	
		1 AJ	U Ajusco	-99.162611	19.154286	2942.0	NaN	4.840000e+11	
		2 A.II	M Ajusco Medio	-99.207744	19.272161	2548.0	NaN	4.840000e+11	
		3 AR	A Aragón	-99.074549	19.470218	2200.0	Finalizó operación en 2010	4.840000e+11	
		4 A3	n Atizapan	-99.254133	19.576963	2341.0	NaN	4.840000e+11	
-	In [10]:	df_estac_m df_estac_m	edia_param = edia_param.†	pd.read_ ead()	csv("noN	ull_Me	s/estacionesConMedia	PorParametro.csv",	encoding='latin-
	Out[10]:	id_station	n id_parameter	mean					
		0 ACC	o co	0.383156					
		1 ACC	ON CON	10.760317					
		2 ACC	D NO2	17.401988					
		3 ACC	XON C	28.247428					

6.

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

]: df	_merged.h	ead()							
]:	id_station	id_parameter	mean	nom_estac	longitud	latitud	alt	obs_estac	id_station.1
0	ACO	со	0.383156	Acolman	-98.912003	19.635501	2198.0	NaN	4.840000e+11
1	ACO	NO	10.760317	Acolman	-98.912003	19.635501	2198.0	NaN	4.840000e+11
2	ACO	NO2	17.401988	Acolman	-98.912003	19.635501	2198.0	NaN	4.840000e+11
3	ACO	NOX	28.247428	Acolman	-98.912003	19.635501	2198.0	NaN	4.840000e+11
4	ACO	O3	29.533586	Acolman	-98.912003	19.635501	2198.0	NaN	4.840000e+11

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.

	А	В	С	D	E	F	G	Н	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 Tecno	-99.194279	19.414393	2287	Finalizó oper	4.84E+11
54	52	СНО	со	0.63222757	Chalco	-98.886088	19.266948	2253		4.84E+11
55	53	СНО	NO	18.4200269	Chalco	-98.886088	19.266948	2253		4.84E+11
56	54	СНО	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	СНО	O3	29.6132454	Chalco	-98.886088	19.266948	2253		4.84E+11
59	57	СНО	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	со	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
60	67	CUA	NO2	21 0452165	Cupiimalap	00 201705	10 265212	2704		A 9/E±11
	• • • • • • • • • • • • • • • • • • •	estacsConM	edia_Geo	(+)						•

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



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.

df_e	stacs_ge	eo_01.head()			resta_sta			,	- A		
	Innamed: (Unnamed: 0.	1 id_station	id_paramet	er me	n nom_e	stac	longitud	latitud	alt	obs_estac	id_station.1
0	()	0 ACC) (0.3831	56 Acol	man	-98.912003	19.635501	2198.0	NaN	4.840000e+11
1	3		1 ACC	4 (IO 10.7603	7 Acol	man	-98.912003	19.635501	2198.0	NaN	4.840000e+11
2	1	1	2 ACC) NO	02 17.4019	38 Acol	man	-98.912003	19.635501	2198.0	NaN	4.840000e+11
3	3	1	3 ACC) NC	OX 28.2474	28 Acol	man	-98.912003	19.635501	2198.0	NaN	4.840000e+11
4	4		4 ACC) (03 29.5335	6 Acol	man	-98.912003	19.635501	2198.0	NaN	4.840000e+11
df_e	stacs_ge	eo_01=pd.read(ad_csv("ne)	Null_Mes/€	istacscon	ledia_Geo	o_nol	vulls.csv	, encodi	ing= 1a	(tin-1')	
	d_station	id_parameter	mean	nom_estac	longitud	latitud	alt	obs_estac	id_statio	on.1		
0	ACO	CO	0.383156	Acolman -	98.912003	19.635501	2198	NaN	4.8400006	1+11		
		8100	10 760317	Acolman -	98.912003	19.635501	2198	NaN	4.8400006	2+11		
1	ACO	NO	10.100011									
1 2	ACO	NO2	17.401988	Acolman	98.912003	19.635501	2198	NaN	4.840000	+11		
	df_e df_e	Unnamed: 0 0 0 1 1 1 2 2 3 3 4<	Unnamed: 0 Unnamed: 0 0 0 1 1 2 2 3 3 4 4 df_estacs_geo_01=pd.red df_estacs_geo_01.head() id_station id_station	Unnamed:0 Unnamed:0 Unnamed:0 Unnamed:0 Unnamed:0 Unnamed:0 I I static I I I Static I <th< td=""><td>Unnamed: Unnamed: (d_station (d_parametic) 0 0 0 ACO CO 1 1 1 ACO NO 2 2 2 ACO NO 3 3 3 ACO NO 4 4 4 ACO CO df_estacs_geo_01=pd.read_csv("noNull_Nes/df_estacs_geo_01=head() Image: The station Image: The station</td><td>Unnamed:0 Unnamed:0: Id_station Id_garameter mem 0 0 0 ACO CO 03811 1 1 1 ACO NO 107003 2 2 2 ACO NO 12174019 3 3 3 ACO NO 222474 4 4 ACO 03 2853351 df_estacs_geo_01=pd,read_csv("nobull_Mes/estacsCore df_estacs_geo_01.head() Ide_station id_station id_parameter mean_nom_estac longitud</td><td>Unnamed: 0 Unnamed: 0.1 id_station id_parameter mean non_e 0 0 0 ACO CO 0.83156 Acot 1 1 ACO NO 10.70017 Acot 2 2 2 ACO NO2 17.40198 Acot 3 3 3 ACO NO2 29.217.40198 Acot 4 4 4 ACO O.3 29.533586 Acot df_estacs_geo_01=pd.read_csv("noNull_Mes/estacsConMedia_Geo df_estacs_geo_01.head() Heady Acot Image: Acot Ac</td><td>Unnamed: 0 Unnamed: 0.1 id_station id_parameter mean nom_estacc 0 0 0 ACO CO 0.83156 Acoiman 1 1 ACO NO 10.700317 Acoiman 2 2 ACO NO2 17.401968 Acoiman 3 3 3 ACO NOX 28.247428 Acoiman 4 4 ACO O.3.28.53566 Acoiman df_estacs_geo_01.pd.read_csv("nokul1_Mes/estacsConMedia_Geo_nod Acoiman df_estacs_geo_01.head() atitud atitud</td><td>Unnamed: 0 Unnamed: 0.1 id_station id_parameter mean nom_estac longitud 0 0 0 ACO CO 0.83166 Acciman -0.93176 1 1 ACO NO 10.700317 Acciman -0.6912003 2 2 ACO NO2 17.40198 Acciman -0.6912003 3 3 3 ACO NO2 17.40198 Acciman -0.6912003 4 4 ACO O.3 29.53568 Acciman -0.6912003 df_estacs_geo_01.pd,read_csv("nokiul1_Mes/estacsConfieldia_Geo_nokiul1s.csv/df_estacs_geo_01.head() -</td><td>Unnamed: 0 Unnamed: 0.1 id_station id_parameter mean nom_estac longitud latitud 0 0 0 ACO CO 0.363166 Acolman -0.01200 19.635601 1 1 1 ACO NO 10.760317 Acolman -0.012003 19.635601 2 2 ACO NO 17.40108 Acolman -0.0012003 19.635601 3 3 3 ACO NOX 20.247428 Acolman -0.0012003 19.635601 4 4 ACO O.3.29.53568 Acolman -0.0012003 19.635601 df_estacs_geo_01.pd.read_csv("noNull_Mes/estacsConMedia_Geo_noNulls.csv", encodid -0.012101 -0.012101 -0.012101 -0.012101 id_station id_aparameter mean nom_estac longitud latitud at obs_estac id_station</td><td>unamed: 0 Unnamed: 0.1 id_station id_parameter mean nom_estac longitud latitud ait 0 0 0 ACO CO 0.85156 Acolman -06.912003 19.855501 2196.0 1 1 ACO NO 10.700317 Acolman -06.912003 19.855501 2196.0 2 2 2 ACO NO2 17.401908 Acolman -06.912003 19.855501 2198.0 3 3 ACO NO2 17.401908 Acolman -06.912003 19.855501 2198.0 4 4 ACO O.3 29.53586 Acolman -06.912003 19.855501 2198.0 df_estacs_geo_01=pd.read_csv(*notkull_Mes/estacsCorrMedia_Geo_notkulls.csv*, encodings*1a df_estacs_geo_01.head() -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111</td><td>Unnamed: 0 Unnamed: 0.1 id_station id_parameter mean nom_estac longitud latitud ait obs_estac 0 0 0 ACO CO 0.88156 Acolman -08.912003 19.855601 2198.0 NaN 1 1 ACO NO 10.700317 Acolman -08.912003 19.855601 2198.0 NaN 2 2 2 ACO NO2 217.401908 Acolman -08.912003 19.855601 2198.0 NaN 3 3 ACO NO2 217.401908 Acolman -08.912003 19.855601 2198.0 NaN 4 4 ACO O.3 29.535806 Acolman -08.912003 19.855601 2198.0 NaN df_estacs_geo_01=pd.read_csv(*notkull_Mes/estacsConMedia_Geo_notkulls.csv*, encodings 'latin-1') df_estacs_geo_01.head() NaN</td></th<>	Unnamed: Unnamed: (d_station (d_parametic) 0 0 0 ACO CO 1 1 1 ACO NO 2 2 2 ACO NO 3 3 3 ACO NO 4 4 4 ACO CO df_estacs_geo_01=pd.read_csv("noNull_Nes/df_estacs_geo_01=head() Image: The station Image: The station	Unnamed:0 Unnamed:0: Id_station Id_garameter mem 0 0 0 ACO CO 03811 1 1 1 ACO NO 107003 2 2 2 ACO NO 12174019 3 3 3 ACO NO 222474 4 4 ACO 03 2853351 df_estacs_geo_01=pd,read_csv("nobull_Mes/estacsCore df_estacs_geo_01.head() Ide_station id_station id_parameter mean_nom_estac longitud	Unnamed: 0 Unnamed: 0.1 id_station id_parameter mean non_e 0 0 0 ACO CO 0.83156 Acot 1 1 ACO NO 10.70017 Acot 2 2 2 ACO NO2 17.40198 Acot 3 3 3 ACO NO2 29.217.40198 Acot 4 4 4 ACO O.3 29.533586 Acot df_estacs_geo_01=pd.read_csv("noNull_Mes/estacsConMedia_Geo df_estacs_geo_01.head() Heady Acot Image: Acot Ac	Unnamed: 0 Unnamed: 0.1 id_station id_parameter mean nom_estacc 0 0 0 ACO CO 0.83156 Acoiman 1 1 ACO NO 10.700317 Acoiman 2 2 ACO NO2 17.401968 Acoiman 3 3 3 ACO NOX 28.247428 Acoiman 4 4 ACO O.3.28.53566 Acoiman df_estacs_geo_01.pd.read_csv("nokul1_Mes/estacsConMedia_Geo_nod Acoiman df_estacs_geo_01.head() atitud atitud	Unnamed: 0 Unnamed: 0.1 id_station id_parameter mean nom_estac longitud 0 0 0 ACO CO 0.83166 Acciman -0.93176 1 1 ACO NO 10.700317 Acciman -0.6912003 2 2 ACO NO2 17.40198 Acciman -0.6912003 3 3 3 ACO NO2 17.40198 Acciman -0.6912003 4 4 ACO O.3 29.53568 Acciman -0.6912003 df_estacs_geo_01.pd,read_csv("nokiul1_Mes/estacsConfieldia_Geo_nokiul1s.csv/df_estacs_geo_01.head() -	Unnamed: 0 Unnamed: 0.1 id_station id_parameter mean nom_estac longitud latitud 0 0 0 ACO CO 0.363166 Acolman -0.01200 19.635601 1 1 1 ACO NO 10.760317 Acolman -0.012003 19.635601 2 2 ACO NO 17.40108 Acolman -0.0012003 19.635601 3 3 3 ACO NOX 20.247428 Acolman -0.0012003 19.635601 4 4 ACO O.3.29.53568 Acolman -0.0012003 19.635601 df_estacs_geo_01.pd.read_csv("noNull_Mes/estacsConMedia_Geo_noNulls.csv", encodid -0.012101 -0.012101 -0.012101 -0.012101 id_station id_aparameter mean nom_estac longitud latitud at obs_estac id_station	unamed: 0 Unnamed: 0.1 id_station id_parameter mean nom_estac longitud latitud ait 0 0 0 ACO CO 0.85156 Acolman -06.912003 19.855501 2196.0 1 1 ACO NO 10.700317 Acolman -06.912003 19.855501 2196.0 2 2 2 ACO NO2 17.401908 Acolman -06.912003 19.855501 2198.0 3 3 ACO NO2 17.401908 Acolman -06.912003 19.855501 2198.0 4 4 ACO O.3 29.53586 Acolman -06.912003 19.855501 2198.0 df_estacs_geo_01=pd.read_csv(*notkull_Mes/estacsCorrMedia_Geo_notkulls.csv*, encodings*1a df_estacs_geo_01.head() -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111 -106.00111	Unnamed: 0 Unnamed: 0.1 id_station id_parameter mean nom_estac longitud latitud ait obs_estac 0 0 0 ACO CO 0.88156 Acolman -08.912003 19.855601 2198.0 NaN 1 1 ACO NO 10.700317 Acolman -08.912003 19.855601 2198.0 NaN 2 2 2 ACO NO2 217.401908 Acolman -08.912003 19.855601 2198.0 NaN 3 3 ACO NO2 217.401908 Acolman -08.912003 19.855601 2198.0 NaN 4 4 ACO O.3 29.535806 Acolman -08.912003 19.855601 2198.0 NaN df_estacs_geo_01=pd.read_csv(*notkull_Mes/estacsConMedia_Geo_notkulls.csv*, encodings 'latin-1') df_estacs_geo_01.head() NaN

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

[38]: d f ['	f_estacs_ge 'id_station	eo_01.pivot n','nom_est	_table('m ac','lati	iean', tud','lon	gitud'],	'id_para	meter').t	:o_csv('n	oNull_Mes	/estacsMe	edia_Geo_	pivoted.c	:sv', end
[39]: df df	f_estacs_ge f_estacs_ge	eo_pivot=pd eo_pivot.he	.read_csv ad()	("noNull_	Mes/esta	acsMedia_	Geo_pivot	ed.csv",	encoding	='latin-1	')		
t[39]:	id station	nom estac	latitud	longitud	CO	NO	NO2	NOV	03	DMAG	DM0.5	PMCO	500
				IUIIUIUU					0.0	PIVITU	PIVIZ.3	PNICO	302
0) ACO	Acolman	19.635501	-98.912003	0.383156	10.760317	17.401988	28.247428	29.533586	40.882878	NaN	NaN	2.828110
0	ACO AJM	Acolman Ajusco Medio	19.635501 19.272161	-98.912003 -99.207744	0.383156	10.760317 5.955789	17.401988 17.793649	28.247428 23.817452	29.533586 39.134903	40.882878 34.591827	NaN 19.503123	NaN 14.892344	2.828110 3.260812
0 1 2	2 ACO	Acolman Ajusco Medio Ajusco	19.635501 19.272161 19.154286	-98.912003 -99.207744 -99.162611	0.383156 0.416988 NaN	10.760317 5.955789 NaN	17.401988 17.793649 NaN	28.247428 23.817452 NaN	29.533586 39.134903 34.440474	40.882878 34.591827 NaN	NaN 19.503123 19.633707	NaN 14.892344 NaN	2.828110 3.260812 NaN
0 1 2 3	ACO AJM AJU AJU AJU AJU	Acolman Ajusco Medio Ajusco Atizapan	19.635501 19.272161 19.154286 19.576963	-98.912003 -99.207744 -99.162611 -99.254133	0.383156 0.416988 NaN 0.524848	10.760317 5.955789 NaN 14.294743	17.401988 17.793649 NaN 22.365726	28.247428 23.817452 NaN 36.682922	29.533586 39.134903 34.440474 27.950104	40.882878 34.591827 NaN 41.278804	NaN 19.503123 19.633707 NaN	NaN 14.892344 NaN NaN	2.828110 3.260812 NaN 5.707607

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'.

	i	d_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 []: In [44]:	df_e df_e	estacs_g estacs_g	eo_pivot['C eo_pivot.he	<mark>0']=df_es</mark> ad()	stacs_geo_	pivot['(co'].fill	na(df_est	acs_geo_	pivot['CO	'].mean())		
In []: In [44]: Out[44]:	df_e df_e	estacs_g estacs_g d _station	eo_pivot['C eo_pivot.he nom_estac	0']=df_es ad() latitud	stacs_geo_ longitud	pivot['(co	co'].fill NO	na(df_est NO2	acs_geo_l	pivot['CO O3	'].mean() PM10) PM2.5	РМСО	SO2
In []: In [44]: Out[44]:	df_e df_e i	estacs_g estacs_g id_station ACO	eo_pivot['C eo_pivot.he nom_estac Acolman	0']=df_es ad() latitud 19.635501	stacs_geo_ longitud -98.912003	pivot['(CO 0.383156	0'].fill NO 10.760317	na(df_est NO2 17.401988	acs_geo_ NOX 28.247428	pivot['CO O3 29.533586	'].mean() PM10 40.882878) PM2.5 NaN	PMCO NaN	SO2 2.828110
In []: In [44]: Out[44]:	df_e df_e i 0	estacs_g estacs_g id_station ACO AJM	eo_pivot['C eo_pivot.he nom_estac Acolman Ajusco Medio	0']=df_es ad() latitud 19.635501 19.272161	longitud -98.912003 -99.207744	pivot['(co 0.383156 0.416988	NO 10.760317 5.955789	na(df_est NO2 17.401988 17.793649	acs_geo_ NOX 28.247428 23.817452	pivot['CO O3 29.533586 39.134903	'].mean() PM10 40.882878 34.591827) PM2.5 NaN 19.503123	PMCO NaN 14.892344	SO2 2.828110 3.260812
In []: In [44]: Out[44]:	df_e df_e i 0 1 2	estacs_g estacs_g id_station ACO AJM AJU	eo_pivot['C eo_pivot.he nom_estac Acolman Ajusco Medio Ajusco	0']=df_es ad() latitud 19.635501 19.272161 19.154286	longitud -98.912003 -99.207744 -99.162611	pivot['(co 0.383156 0.416988 0.576028	CO'].fill NO 10.760317 5.955789 NaN	na (df_est NO2 17.401988 17.793649 NaN	acs_geo_ NOX 28.247428 23.817452 NaN	pivot['CO O3 29.533586 39.134903 34.440474	'].mean() PM10 40.882878 34.591827 NaN) PM2.5 NaN 19.503123 19.633707	PMCO NaN 14.892344 NaN	SO2 2.828110 3.260812 NaN
In []: In [44]: Out[44]:	df_e df_e i 0 1 2 3	estacs_g estacs_g d_station ACO AJM AJU ATI	eo_pivot['C eo_pivot.he nom_estac Acolman Ajusco Medio Ajusco Atizapan	0']=df_es ad() latitud 19.635501 19.272161 19.154286 19.576963	longitud -98.912003 -99.207744 -99.162611 -99.254133	pivot['(<u>co</u> 0.383156 0.416988 0.576028 0.524848	NO 10.760317 5.955789 NaN 14.294743	na(df_est NO2 17.401988 17.793649 NaN 22.365726	acs_geo_ NOX 28.247428 23.817452 NaN 36.682922	oivot['CO O3 29.533586 39.134903 34.440474 27.950104	'].mean() PM10 40.882878 34.591827 NaN 41.278804) PM2.5 NaN 19.503123 19.633707 NaN	PMCO NaN 14.892344 NaN NaN	SO2 2.828110 3.260812 NaN 5.707607

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

	id_station	nom_estac	latitud	longitud	co	NO	NO2	NOX	03	PM10	PM2.5	PMCO	SO2	N
	ACO	Acolman	19.635501	-98.912003	0.383156	10.760317	17.401988	28.247428	29.533586	40.882878	22.107702	20.538719	2.828110	28.24742
	MLA I	Ajusco Medio	19.272161	-99 207744	0.416988	5.955789	17.793649	23.817452	39.134903	34.591827	19.503123	14.892344	3.260812	23.81745
	Z AJU	Ajusco	19.154286	-99.162611	0.576028	16.929812	23.530309	41.153246	34.440474	43.090817	19.633707	20.538719	4.197430	41.15324
	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	36.68292
	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	41.15324
0					HIVOCS		PODRÁU		NOGIS					

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 <u>http://www.qgis.org/</u> En la sección de descargas seleccionamos el instalador de acuerdo a nuestro sistema operativo.

Descarga QGIS Paquetes binarios (instaladores) e La versión actual es QGIS 3.26.0 Los repositorios a largo plazo actu QGIS está disponible en Windows	Vescarga QGIS para tu plataforma puetes binarios (instaladores) están disponibles desde esta página. versión actual es QGIS 3 26 0 Buenos Aires' y fue lanzada en 17.06 2022. repositorios a largo plazo actualmente ofecen QGIS 3 22.8 Bialowieza' AIS está disponible en Windows, macOS, Linux y Android.								
DESCARGAS DE INSTALACIÓ	TODOS LOS LANZAMIENTOS FUENTES								
Los instaladores auto	nomos (MSI) de los paquetes OSGeo4W (re	comendado para nuevos usuarios)							
Versión mas reciente (ma	is rica en características):								
* Q	QGIS Versión Instalador Autónomo 3.26								
sha256									
Versión con Soporte a La	rgo Plazo (mas estable):								
▲ 	QGIS Versión Instalador Autónomo 3.22								
sha256									

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.

🖟 QGIS 3.26.0 'Buenos Aires' Setup		-		×
Destination Folder Click Next to install to the default folder o	r click Change to choose	another.		R
Install QGIS 3.26.0 'Buenos Aires' to:				
C:\QGIS 3.26.0\]
☑ Create a desktop shortcuts.				
Create a start menu shortcuts.				
	<u>B</u> ack <u>N</u> e	ext	Can	cel

4. Instalamos el programa.



5. Comienza el proceso de instalación.

🛃 QG	S 3.26.0 'Buenos Aires' Setup		_		×
In	stalling QGIS 3.26.0 'Buenos Air	es'			R
Ple	ase wait while the Setup Wizard installs C	QGIS 3.26.0 'Bue	nos Aires'.		
Sta	tus: Validating install				
		Back	Next	Can	cel

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.

8	5- ð- 🖻 🖆	÷				estacs	Media_Geo_piv	roted_NO_Nulls	.csv - Excel				Inic. s	ses. 🖬	- 0	×
Archivo	Inicio Insertar	Diseño de pá	gina Fórmu	ılas Datos	Revisar	Vista Ayu	da 📿 įQue	é desea hacer?							,Q₊ Cor	npartir
Pegar Portapape	Calibri ■ → N K <u>S</u> → eles G Fue	- 11 - A [*] ■ - <u>-</u>		≡ ॐ • ≡	환 Ajustar tex 텔 Combinar ación	to y centrar 👻	General \$ - % 000 Número	→ Si Si Fe Cone	ormato Dar f dicional ▼ como Estil	ormato Estilos o tabla + celda los	de Insertar E	Timinar Format	> Autosu	ma * AZT r* Ordenau filtrar Edición	y Buscary seleccionar*	
1 PO:	POSIBLE PÉRDIDA DE DATOS Algunas características del libro se pueden perder si lo guarda como CSV (delimitado por comas). Para conservar estas características, guárdelo como archivo No mostrar de nuevo Guardar como ×															
A1																
	АВ											м	N			(🔺
1	lid station	nom estac	latitud	longitud	со	NO	NO2	NOX	03	PM10	PM2.5	PMCO	SO2	NX		
2	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		_ 12
3	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		
4	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		
5	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		
6	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		
7	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		
8	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		
9	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		
10	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		
11	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		
12	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		
13	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		
14	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		
15	13 GAM	Gustavo A. N	19.4827	-99.094517	0.57602772	16.9298117	21.693139	41.1532457	31.4396994	43.4722029	23.589813	20.9926109	4.19742961	41.1532457		
16	14 HGM	Hospital Gen	19.411617	-99.152207	0.65855795	17.9916773	29.8859623	47.9178603	29.010809	43.1098626	24.1175903	18.9428877	4.98241257	47.9178603		
17	15 INN	Investigacion	19.291968	-99.38052	0.26349003	16.9298117	23.5303092	41.1532457	37.0635595	28.9714032	16.7811128	12.3699594	2.02358801	41.1532457		
18	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		
19	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		
20	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		
21	10 1400	Morcod	10 42461	00 11050/	0.90091436	26 7950221	22 0570100	50 0000000	26.0027954	50 4421570	24 9695746	25.474229	5 27020200	50 0000000		
	estacsMedia	a_Geo_pivoted		Ð									_			
Listo										Promedio: 18.	Recuento:	38 Suma: 703	E			+ 100%

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



10. Le damos la siguiente configuración.

Q Administrador de fuentes de date	os Texto delimitado	×
Navegador	Nombre de archivo D:\TALLER_CONCIAS\QGIS\estacsMedia_Geo_pivoted_NO_Nulls.csv	×
V- Vector	Nombre de la capa estacsMedia_Geo_pivoted_NO_Nulls Codificación UTF-8	
Ráster	Número de líneas de encabezamiento a descartar 0 🛊 📃 El separador decimal es la coma	
Malla	El primer registro tiene los nombres de campo Recortar campos	
P: Nube de puntos	Detectar tipos de campo	
🤊 🚦 Texto delimitado	▽ Definición de geometría	
🤗 GeoPackage	Coordenadas del punto Campo X longitud Campo Z	
🚛 gps	Texto bien conocido (WKT) Campo Y latitud Campo M	
Castial ita	Coordenadas GMS	
	Project CRS: EPSG:4326 - WGS 84	· 😨
PostgreSQL	▽ Configuraciones de capa	
MSSQL	Usar índice espacial Usar índice de subconjuntos Vigilar archivo	
📮 Oracle	Datos de ejemplo	
Vapa virtual	id station nom estac latitud longitud CO NO NO2	
SAP HANA	1 ACO Acolman 19.635501 -98.912003 0.383156059 10.76031681 17.401988	37 28.2
WFS / OGC API - Funcionalidades	Cerrar	<u>A</u> ñadir Ayuda

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: <u>https://mt1.google.com/vt/lyrs=y&x={x}&y={y}&z={z}</u>

Q Conexión XYZ		×
Nombre Hibrido		
Detalles de la conexión		
URL	https://mt1.google.com/vt/lyrs=y&x={x}&y={y}&z={z}	
Autenticación		
Configuraciones	Básica	
Seleccionar o crear una c	onfiguración de autenticación	
	· / - +	
La configuración guarda l	as credenciales encriptadas en la base de datos de autenticación de QGIS.	
Nivel de zoom mínimo		
Nivel de zoom máximo		
Referente		
Resolución de tesela	High (512x512 / 192 DPI)	
	Aceptar Cancelar	Ayuda

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.

Q Guardar capa vectorial como									
Formato									
Nombre de archivo	IAS\QGIS\4326\salu_nd_	_estac_meteo_ga	lindo_zm_p.gpkg	×					
Nombre de la capa	salu_nd_estac_meteo_g	galindo_zm_p							
SRC				وي ا					
Codificación									
Guardar sólo lo									
Seleccione ca	mpos a exportar y sus o	opciones de exp	ortación						
Nombre									
📕 fid 🛛 Ini									
id_sta St									
nom St									
latitud Re	eal								
longit Re	eal								
CO Re	eal								
Sele	eccionar todo	Desel	eccionar todo						
Persist layer m	ietadata								
Geometría									
Tipo de geometri									
	Añadir archivo gu	uardado al mapa	Aceptar Cance						

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.



- Carp Copar capa
 Baster Base de glatos Web Malla Progesos Ayuda

 Carbining rombre de la capa
 Duplicar capa

 Duplicar capa
 Carbining rombre de la capa

 Mover griba
 Eliminar capa...

 Mover griba
 Abrit tabla de glinbatos

 arité
 Commutar glición

 Elitrar...
 Commutar glición

 Elitrar...
 Commutar glición

 Elitrar...
 Socio

 SRC de la capa
 Socio

 Hascer permanente...
 Exportar

 Exportar
 Socio

 Hascer permanente...
 Exportar

 Exportar
 Socio

 Hascer permanente...
 Exportar

 Exportar
 Exportar

 Exportar
 Exportar

 Exportar
 Exportar

 Exportar
 Exportar
 </t
- 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.

Q	Polígonos Voronoi	— Objetos Totales:	: 38, Filtrados: 38, Sel	eccionados: 1						- 0	×
/	z 🛃 📾 🖷	i 🗠 🗿 🛍 .	🖻 🗏 📉 👡	🍸 🖽 🕸 🔎	III III 🗾 🛄	🗄 🗿 🎕					
	fid	id_station	nom_estac	latitud	longitud	CO		NO2	NOX		
	23	MPA	Milpa Alta	19.1769	-98.990189	0.29080774					08
2			Tlohuac	19.246459		0.487674068	9.244423834		28.15650187	34.171031	
		AJU	Ajusco	19.154286	-99.162611		16.92981171	23.5303092	41.15324567	34.4404736	
		TPN	Tlalpan	19.257041	-99.184177	0.75568782	11.944663	18.58920232	30.52312764		
5		UAX	UAM Xochimilco	19.304441	-99.103629	0.573042182	11.69960899	23.06694024	34.7734295	32.124933	
6		MLA	Ajusco Medio	19.272161	-99.207744	0.416988237		17.79364859	23.81745184	39.134902	
			Santa Ursula	19.31448	-99.149994		20.28458324	29.64946184	49.91894834	28.809303	
8		CCA	Centro de Cien	19.326111	-99.176111		11.85297222	23.37961748	35.23547099	32.330782	
9		PED	Pedregal	19.325146	-99.204136	0.443798858	9.61205339	22.8388828	32.42765605	33.44523	
10			Investigaciones	19.291968	-99.38052	0.26349003		23.5303092			
		SFE	Santa FE	19.357357	-99.262865	0.489712327	11.14726103	22.49874579	33.65323422	32.605986	
			Santiago Acahu	19.34561		0.408397378	10.59670078	20.44565526		33.5509050	01
		СНО	Chalco	19.266948	-98.886088	0.632227574	18.4200269	22.78036183	41.29641331	29.613245	
		соу	Coyoac ۞ n	19.350258	-99.157101		17.19432135	26.3709842	43.66846495	30.008443	
		IZT	Iztacalco	19.384413	-99.117641	0.749303712	20.33483072	29.69902962	50.05906328	28.663230	55
16				19.360794			16.73041268	27.24688678	43.98968169	29.243138	61
										B	•

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.

Q Contar puntos en un polígono	×
Parámetros Registro Polígonos ● ■ salu_nd_polígons_pob_voron_galindo_zm_a [EPSG:4326 ・ 口 2010] ↓ 2010 □ Objetos seleccionados solamente ●	Contar puntos en un polígono
Puntos Puntos Salu_nd_def_cancer_gaindo_zm_p [EPSG:4326] Objetos seleccionados solamente Campo de peso [opcional]	
Campo de clase [opcional]	
Defunciones Número [Crear capa temporal]	conteniendo un campo adicional con la cuenta de puntos correspondiente a cada polígono.
0%	Cancelar Elecutor Cerror Avudo

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".

Q	😡 salu_nd_pob_def_cancer_voron_galindo_zm_a— Objetos Totales: 38, Filtrados: 38, Seleccionados: 0 – 🛛												×
/	2 🗈	B, B ≥ 0	🔹 🖻 🗏 🛯	N 🛼 🍸 💷 🕴	🕂 🏴 🖪	🕺 🗰 📰 📲	e,						
23	longitud -98.990189	CO 0.29080774		NO2 8.385830028	NOX 41.15324567	O3 43.11909608	PM10 38.98799326	PM2.5 20.85771175	PMCO 18.04859782	SO2 2.404651218	NX 41.15324567		cion UT749
24					40.29476062								42347
25	-99.204136	0.443798858	9.61205339		32.42765605	33.4452325	40.22510713	21.2984414			32.42765605		28711
26													56400
27		0.63045598		23.7175707	42.27821564	27.48875206	48.94249922	23.17217199	25.79036307	4.720048302	42.27821564		01184
28	-99.262865	0.489712327											07101
29			22.16187805			28.9884075		21.70102894	20.53871879	4.238128139			59817
30			20.28458324										60906
31		0.487674068	9.244423834	18.9202449	28.15650187	34.17103125	42.2506188	22.10770214	20.53871879	2.893838126	28.15650187		87736
32	-99.204597		26.44676713										30873
33	-99.177173	0.55138642	17.82726846		42.79741759	27.94566915	45.76231479	22.10770214	20.53871879	6.732648633	42.79741759		
34													
35	-99.103629	0.573042182		23.06694024	34.7734295	32.12493347		20.95523345	20.53871879		34.7734295		
36						29.24313861							
37			14.3070458	20.21735649		28.75460116	53.48153717	22.10770214	20.53871879				
38								27.72529049					
	lostrar todos	los obietos espa	riales_										

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	×
N <u>o</u> mbre	ratio
Comentario	
Тіро	Número entero (entero)
Tipo de proveedor	Número entero (entero 64 bit)
	Número decimal (real)
	Texto (cadena)
	JSON (texto)

Q PoligVoronoi_Pob_Defuncs — Field Calcu	lator	×
Actualizar sólo 0 objetos espaciales seleccion Crear un campo nuevo	ados Actualizar campo existente	
Crear campo virtual Nombre del campo de salida Tipo del campo de salida Longitud del campo de salida	entero) •	Ŧ
Expresión Editor de funciones		
<pre>"Defunciones" / "poblacion" = + - / * ^ () '\n' Objeto espacial Acolman</pre>	Buscar Mostrar valores 1.2 O3 1.2 PM10 1.2 PM2.5 1.2 PMCO 1.2 SO2 1.2 NX 122 poblacion 1.2 poblacion 1.2 ratio * Capas de mapa * Condicionales * Condicionales	1000 C
	Aceptar Cancelar Ayuc	da

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.

Q	salu_nd_pob	_def_cancer_ra	tio_voron_galin	do_zm_a— Obj	etos Totales: 38	3, Filtrados: 38, S	Seleccionados:	0			-		×
/	z 🖩 C	≣ ≣ × ø	🖹 🖆 🗮 関	🛚 🛼 🍸 🕮 🎙	• 🗩 🖪 🖪	🕺 🗮 🗮	e,						
		NO2	NOX				PMCO						
1	0.76031681	17.40198837	28.2474278		40.882878	22.10770214	20.53871879		28.2474278	364414			86
2								3.260811556					77
3			41.15324567	34.44047366		19.63370694	20.53871879	4.197429609	41.15324567	63540			81
4	4.29474317					22.10770214							41
5		21.48558421	41.15324567	31.259555	39.57943032	22.47406528		4.388509755	41.15324567				607
6	5.28273937				46.70136393								72
7	1.85297222	23.37961748	35.23547099	32.33078238		18.85801288	20.53871879	3.466161655	35.23547099				36
8	18.4200269		41.29641331			22.10770214			41.29641331				375
9	7.19432135	26.3709842	43.66846495	30.00844338			20.53871879	4.197429609	43.66846495			0.015044	71
10						22.10770214							88
11	8.48349303	19.65100494	38.13800918	26.35265471	50.54240925	22.10770214	20.53871879	5.654463935	38.13800918				86
12	3.29564637	25.48290894	48.77998503			22.10770214			48.77998503				71
13	1.25368566	17.33252672	31.50669593	33.94447227		18.80816745	20.53871879		31.50669593			0.004601	16
14													18
15	7.99167728		47.91786028		43.10986259	24.11759029	18.94288765		47.91786028				i02
16	6.92981171	23.5303092	41.15324567	37.06355951	28.97140321	16.78111276	12.3699594	2.02358801	41.15324567				i01
	Mostrar todos	los objetos espa	ciales 🖵									E	3

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

Q Guardar capa vectorial como X						
Formato Nombre de archivo Nombre de la capa	Valores separados por comas [CSV] _salu_nd_pob_def_cancer_ratio_voron_gaindo_zm_a.csv	• ×				
SRC	Default CRS: EPSG:4326 - WGS 84	•				
Codificación ☐ Guardar sólo los ▼ Seleccione can	UTF-8 objetos espaciales seleccionados npos a exportar y sus opciones de exportación					
Nombre fid id_station nom_estac latitud longitud CO	Tipo emplazar con los valores mostrac Integer64 String String Real Real Real					
Sele Sustituir todos Persist layer met ⊂ Geometría	ccionar todo Deseleccionar todo ; los valores de campo en bruto seleccionados por los valores tadata Añadir archivo guardado al mapa Aceptar Cancel	: mostrado ar Ayuda				

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.

jupyter	04_0	Correlatio	ON Last Che	ckpoint: 05	/12/2022 (u	nsaved ch	anges)								?	ogou
File Edit	View	Insert	Cell Ke	ernel W	idgets H	lelp						Ν	lot Trusted	🖉 🛛 Pytho	on 3 (ipykerr	nel)
) + × 4	6	^	Run	C))	Code	~	0									
In [1]: import pandas as pd In [2]: df=pd.read_csv("kabla_salu_nd_pob_def_cancer_ratio_voron_galindo_zm_a.csv", encoding='latin-1') df.head() Out[2]: 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	
	12	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	
	2 3 3 4	AJU ATI	Ajusco Atizapan	19.154286 19.576963	-99.162611 -99.254133	0.576028 0.524848	16.929812 14.294743	23.530309 22.365726	41.153246 36.682922	34.440474 27.950104	43.090817 41.278804	19.633707 22.107702	20.538719 20.538719	4.197430 5.707607	41.153246 36.682922	
	 2 3 3 4 4 5 	AJU ATI BJU	Ajusco Atizapan Benito Juī¿½rez	19.154286 19.576963 19.370464	-99.162611 -99.254133 -99.159596	0.576028 0.524848 0.526351	16.929812 14.294743 16.929812	23.530309 22.365726 21.485584	41.153246 36.682922 41.153246	34.440474 27.950104 31.259555	43.090817 41.278804 39.579430	19.633707 22.107702 22.474065	20.538719 20.538719 17.099596	4.197430 5.707607 4.388510	41.153246 36.682922 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.

	IMPORTAMOS LA LIBRERÍA PEARSONR DE SCIPY
In [4]:	<pre>from scipy.stats import pearsonr</pre>
In [6]:	<pre>pearsonr(df['CO'], df['ratio'])</pre>
Out[6]:	(0.30272138338657545, 0.06469270520639792)

3. Realizamos la misma operación para los contaminantes faltantes.

```
🖺 | + | ≫ 🖆 🗈 | ↑ ↓ | ▶ Run | ■ C ▶ Code
                                                          ~
       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['03'], 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 NO₂ y NO_x a un nivel de significancia mucho menor al 1 %.

In [8]:	<pre>pearsonr(df['NO2'], df['ratio'])</pre>
Out[8]:	(0.4985339905942887, 0.001444661823247135)
In [9]:	<pre>pearsonr(df['NOX'], df['ratio'])</pre>
Out[9]:	(0.4169889069750331, 0.009201867990165107)