



Tecnológico de Monterrey

TC2029 Ciencia y Analítica de Datos
DRA. María de la Paz Rico Fernández

Juan Pablo Bladinieres Marín del Campo A01793474
Gerardo Quiroga Nájera A00967999

**Limpieza, Análisis, Visualización y K-Means en
datos de Calidad del Agua en sitios de monitoreo
de aguas superficiales**

Noviembre 2022

Limpieza de Base de datos

El dataset que vamos a revisar es el de Datos de Agua de sitios de monitoreo de aguas superficiales de 2020 en México.

El data set incluye 55 columnas, con 4,141 registros de datos sobre la calidad del agua:

#	Column	Non-Null Count	Dtype
0	CLAVE	3493 non-null	object
1	SITIO	3493 non-null	object
2	ORGANISMO_DE_CUENCA	3493 non-null	object
3	ESTADO	3493 non-null	object
4	MUNICIPIO	3493 non-null	object
5	CUENCA	3492 non-null	object
6	CUERPO DE AGUA	3479 non-null	object
7	TIPO	3493 non-null	object
8	SUBTIPO	3479 non-null	object
9	LONGITUD	3493 non-null	float64
10	LATITUD	3493 non-null	float64
11	PERIODO	3493 non-null	float64
12	DBO_mg/L	2581 non-null	object
13	CALIDAD_DBO	2581 non-null	object
14	DQO_mg/L	2581 non-null	object
15	CALIDAD_DQO	2581 non-null	object
16	SST_mg/L	3489 non-null	object
17	CALIDAD_SST	3489 non-null	object
18	COLI_FEC_NMP_100mL	2582 non-null	object
19	CALIDAD_COLI_FEC	2582 non-null	object
20	E_COLI_NMP_100mL	2582 non-null	object
21	CALIDAD_E_COLI	2582 non-null	object
22	ENTEROC_NMP_100mL	904 non-null	object
23	CALIDAD_ENTEROC	904 non-null	object
24	OD_PORC	1797 non-null	object
25	CALIDAD_OD_PORC	1797 non-null	object
26	OD_PORC_SUP	1619 non-null	object
27	CALIDAD_OD_PORC_SUP	1619 non-null	object
28	OD_PORC_MED	487 non-null	object
29	CALIDAD_OD_PORC_MED	487 non-null	object
30	OD_PORC_FON	946 non-null	object
31	CALIDAD_OD_PORC_FON	946 non-null	object
32	TOX_D_48_UT	1816 non-null	object
33	CALIDAD_TOX_D_48	1816 non-null	object
34	TOX_V_15_UT	1819 non-null	object
35	CALIDAD_TOX_V_15	1819 non-null	object
36	TOX_D_48_SUP_UT	762 non-null	object
37	CALIDAD_TOX_D_48_SUP	762 non-null	object
38	TOX_D_48_FON_UT	0 non-null	float64
39	CALIDAD_TOX_D_48_FON	0 non-null	float64
40	TOX_FIS_SUP_15_UT	1674 non-null	object
41	CALIDAD_TOX_FIS_SUP_15	1674 non-null	object
42	TOX_FIS_FON_15_UT	0 non-null	float64
43	CALIDAD_TOX_FIS_FON_15	0 non-null	float64
44	SEMAFORO	3493 non-null	object
45	CONTAMINANTES	2226 non-null	object
46	CUMPLE_CON_DBO	3493 non-null	object
47	CUMPLE_CON_DQO	3493 non-null	object
48	CUMPLE_CON_SST	3493 non-null	object
49	CUMPLE_CON_CF	3493 non-null	object
50	CUMPLE_CON_E_COLI	3493 non-null	object
51	CUMPLE_CON_ENTEROC	3493 non-null	object
52	CUMPLE_CON_OD	3493 non-null	object
53	CUMPLE_CON_TOX	3493 non-null	object
54	GRUPO	3493 non-null	object

Para poder llevar a cabo el análisis, se realiza la separación de tipos de variables:

```
# Separación de tipos de variables
binarias=['CUMPLE_CON_DBO','CUMPLE_CON_DQO','CUMPLE_CON_SST','CUMPLE_CON_CF','CUMPLE_CON_E_COLI','CUMPLE_CON_ENTEROC','CUMPLE_CON_OD']
categoricas=['TIPO','SUBTIPO','SEMAFORO','CONTAMINANTES','GRUPO']
calidades=['CALIDAD_DBO','CALIDAD_DQO','CALIDAD_SST','CALIDAD_COLI_FEC','CALIDAD_E_COLI','CALIDAD_ENTEROC','CALIDAD_OD_PORC','CALIDAD_OD_PORC_SUP']
localizacion=['SITIO','ESTADO','MUNICIPIO','CUENCA','CUERPO DE AGUA','ORGANISMO DE CUENCA','LONGITUD','LATITUD']
numericas=['DBO_mg/L','DQO_mg/L','SST_mg/L','COLI_FEC_NMP_100mL','E_COLI_NMP_100mL','ENTEROC_NMP_100mL','OD_PORC','OD_PORC_SUP','OD_PORC_SUP']
identificador=['CLAVE']
necesarios=['PERIODO']
```

Se determina que el periodo no es necesario dado que solo contiene un dato (del año pero no es diferente) nota: podría ser útil en el caso de que se comparen periodos pero en esta base de datos no es necesario.

```
# Transformación de Binarias
test=df[binarias].copy()
test.replace({'ND':0,'NO':0,'SI':1}, inplace=True)
for i in test.columns:
    print(test.groupby(i).size())
```

```
CUMPLE_CON_DBO
0.0    1174
1.0    2319
dtype: int64
CUMPLE_CON_DQO
0.0    1843
1.0    1650
dtype: int64
CUMPLE_CON_SST
0.0     389
1.0    3104
dtype: int64
CUMPLE_CON_CF
0.0    2545
1.0     948
dtype: int64
CUMPLE_CON_E_COLI
0.0    2040
1.0    1453
dtype: int64
CUMPLE_CON_ENTEROC
0.0    2741
1.0     752
dtype: int64
CUMPLE_CON_OD
0.0     535
1.0    2958
dtype: int64
CUMPLE_CON_TOX
0.0      82
1.0    3411
dtype: int64
```

En esta transformación se determina que el ND y NO son igual que 0 por lo tanto se recomienda juntar estos dos resultados y al final se cambian a valores de 1 y 0 para mejorar la visibilidad.

Otro de los cambios en las variables Binarias es dejar las longitudes y latitudes, a fin que sea más fácil graficar los datos:

```
test3=df[localizacion].copy()
test3.drop(['SITIO','ESTADO','MUNICIPIO','CUENCA','CUERPO DE AGUA','ORGANISMO DE CUENCA'],axis=1,inplace=True)
for i in test3.columns:
    print(test3.groupby(i).size())
```

```
LONGITUD
-117.12403    1
-117.10789    1
-117.10715    1
-117.09717    1
-117.08115    1
..
-86.75637     1
-86.75567     1
-86.74517     1
-86.73982     1
-86.73215     1
Length: 3486, dtype: int64
LATITUD
14.53491     1
14.54128     1
14.55447     1
14.61337     1
14.61567     2
..
32.66399     1
32.66450     1
32.66608     1
32.70583     1
32.70650     1
Length: 3485, dtype: int64
```

Dentro de las variables categóricas, observamos lo siguiente:

- Las variables Tipo y Grupo son muy parecidas, por tanto hemos decidido eliminar Tipo y sólo quedarnos con grupo
- La columna Contaminantes es redundante en el sentido que en caso se encuentren contaminantes, éstos se encuentran en las columnas específicas. Adicional, que es más útil tener cada contaminante en su columna que juntas en una sola
- La columna Subtipo contiene errores de escritura, por lo que decidimos convertir todos los datos a mayúsculas para evitar los errores encontrados
- La Columna Semáforo se puede transformar para que más adelante se utilice con colores (verde, amarillo y rojo) más adelante.

```
# Categoricas
test2=df[categoricas].copy()

test2.drop(['TIPO', 'CONTAMINANTES'],axis=1,inplace=True)
test2['SUBTIPO'] = test2['SUBTIPO'].str.upper()
test2['SEMAFORO'].replace({'Amarillo':'y','Rojo':'r','Verde':'g'},inplace=True)

for i in test2.columns:
    print(test2.groupby(i).size())
```

En la columna de calidades, nos encontramos con datos bastante pequeños, por tanto reemplazamos los valores:

```
calidades1=['CALIDAD_DBO', 'CALIDAD_DQO', 'CALIDAD_SST', 'CALIDAD_COLI_FEC', 'CALIDAD_E_COLI', 'CALIDAD_ENTEROC', 'CALIDAD_OD_PORC', 'CALI
calidades2=['CALIDAD_TOX_V_15', 'CALIDAD_TOX_D_48', 'CALIDAD_TOX_D_48_SUP', 'CALIDAD_TOX_D_48_FON', 'CALIDAD_TOX_FIS_SUP_15', 'CALIDAD_T

test4=df[calidades].copy()
test4[calidades1] = test4[calidades1].replace({np.NaN:'Excelente'})
test4[calidades2] = test4[calidades2].replace({np.NaN:'No Toxic'})

for i in test4.columns:
    print(test4.groupby(i).size())
```

Explorar los Datos

En la siguiente tabla podremos ver un extracto de algunas columnas y de los primeros 3 registros (para más información, revisar el código en [Git Hub](#)):

```
df.head(3)
```

	CLAVE	SITIO	ORGANISMO_DE_CUENCA	ESTADO	MUNICIPIO	CUENCA	CUERPO DE AGUA	TIPO	SUBTIPO	LONGITUD	...	CONTAMINANTES
0	DLAGU8	PRESA EL SAUCILLO 100M AGUAS ARRIBA DE LA CORTINA	LERMA SANTIAGO PACIFICO	AGUASCALIENTES	RINCON DE ROMOS	RIO SAN PEDRO	PRESA EL SAUCILLO	LENTICO	PRESA	-102.33911	...	DQO,CF,
1	DLBAJ100	LOS CABOS SEG 22, 2 ISA10B	PENINSULA DE BAJA CALIFORNIA	BAJA CALIFORNIA SUR	LOS CABOS	SAN JOSE DEL CABO	OCEANO PACIFICO	COSTERO	OCEANO-MAR	-109.84290	...	NaN
2	DLBAJ101	LOS CABOS SEG 22, 1 ISA10B	PENINSULA DE BAJA CALIFORNIA	BAJA CALIFORNIA SUR	LOS CABOS	SAN LUCAS	OCEANO PACIFICO	COSTERO	OCEANO-MAR	-109.86442	...	NaN

3 rows x 55 columns

A continuación, algunos datos estadísticos:

	LONGITUD	LATITUD	DBO_mg/L	DQO_mg/L	SST_mg/L	COLI_FEC_NMP_100mL	E_COLI_NMP_100mL	ENTEROC_NMP_100mL	OD_PORC	OD_PORC_SUP	...
count	3493.000000	3493.000000	4141.000000	4141.000000	4141.000000	4.141000e+03	4.141000e+03	4141.000000	4141.000000	4141.000000	...
mean	-100.359969	21.046992	9.933834	38.747509	83.939839	5.966389e+04	4.946844e+04	236.897778	28.739918	31.812050	...
std	6.122773	3.893696	52.175019	122.697137	407.697458	9.240891e+05	8.309977e+05	2060.532745	39.224503	43.602828	...
min	-117.124030	14.534910	0.000000	0.000000	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	...
25%	-103.882310	18.396070	0.000000	0.000000	0.000000	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	...
50%	-99.795530	20.148980	0.000000	0.000000	18.000000	2.100000e+02	1.100000e+01	0.000000	0.000000	0.000000	...
75%	-96.860230	22.828930	5.000000	35.750000	48.000000	4.600000e+03	1.100000e+03	0.000000	61.600000	70.600000	...
max	-86.732150	32.706500	1500.000000	2871.250000	9430.000000	2.419600e+07	2.419600e+07	24196.000000	226.100000	289.000000	...

Correlación:

	LONGITUD	LATITUD	DBO_mg/L	DQO_mg/L	SST_mg/L	COLI_FEC_NMP_100mL	E_COLI_NMP_100mL	ENTEROC_NMP_100mL	OD_PORC	OD_PORC_SUP	...	TOX_FIS_SUP_15_UT	TOX_FIS_FON_15_UT	CUMPLE_CON_DBO	CUMPLE_CON_DQO	CUMPLE_CON_SST	CUMPLE_CON_CF
LONGITUD	1.00000	-0.71366	-0.00485	-0.01314	0.000766	-0.017869	-0.015878	-0.014317	-0.022233	-0.081512	...	0.018955	NaN	0.007980	0.010767	0.042338	0.021044
LATITUD	-0.71366	1.00000	-0.020814	-0.024895	-0.047574	0.004967	0.004417	-0.047095	0.067502	0.094280	...	-0.014043	NaN	-0.007730	0.026398	0.081863	0.016972
DBO_mg/L	-0.00485	-0.020814	1.00000	0.883136	0.078177	0.013631	0.088488	-0.021895	-0.047819	-0.087627	...	0.012397	NaN	-0.179309	-0.156833	-0.194836	-0.051679
DQO_mg/L	-0.01314	-0.024895	0.883136	1.00000	0.141835	0.029332	0.081151	-0.036316	-0.024982	-0.130308	...	0.039049	NaN	-0.121527	-0.219944	-0.255178	-0.006053
SST_mg/L	0.000766	-0.047574	0.078177	0.141835	1.00000	0.038488	0.036999	-0.011137	0.138408	-0.093747	...	-0.000675	NaN	0.044036	-0.071791	-0.448025	-0.088172
COLI_FEC_NMP_100mL	-0.017869	0.004967	0.013631	0.029332	0.038488	1.00000	0.932232	-0.007425	-0.028053	-0.045480	...	-0.003491	NaN	-0.075993	-0.058833	-0.127415	-0.042765
E_COLI_NMP_100mL	-0.015878	0.004417	0.088488	0.081151	0.036999	0.932232	1.00000	-0.008446	-0.028979	-0.043098	...	-0.003241	NaN	-0.073780	-0.056487	-0.114791	-0.039502
ENTEROC_NMP_100mL	-0.014317	-0.047095	-0.021895	-0.036316	-0.011137	-0.007425	-0.008446	1.00000	-0.084259	0.104014	...	-0.002705	NaN	-0.176172	-0.118604	0.003343	-0.070503
OD_PORC	-0.022233	0.067502	-0.047619	-0.022492	0.138408	-0.028953	-0.028979	-0.084259	1.00000	-0.534700	...	-0.040533	NaN	0.504051	0.448657	-0.188023	-0.050979
OD_PORC_SUP	-0.081512	0.094280	-0.067627	-0.103008	-0.093747	-0.045480	-0.043098	0.104014	-0.534700	1.00000	...	0.060292	NaN	-0.368370	-0.262887	0.231953	0.132686
OD_PORC_MED	-0.140326	0.041442	-0.050708	-0.059928	-0.056273	-0.021475	-0.020109	0.006345	-0.248047	0.470098	...	0.057182	NaN	-0.036410	0.011919	0.125368	0.210980
OD_PORC_FON	-0.134322	0.098824	-0.072061	-0.092186	-0.071595	-0.031043	-0.028939	0.019289	-0.357194	0.607112	...	0.019629	NaN	-0.220430	-0.132743	0.172788	0.123290
TOX_D_48_UT	-0.019231	0.019784	0.328426	0.369477	0.044492	0.199492	0.183022	-0.010812	-0.024861	-0.068812	...	-0.005201	NaN	-0.049898	-0.076987	-0.048758	0.003449
TOX_V_15_UT	0.019173	-0.029032	0.591717	0.538263	0.035530	0.355003	0.386881	-0.010316	-0.038255	-0.065466	...	-0.004963	NaN	-0.121557	-0.081953	-0.063894	-0.050928
TOX_D_48_SUP_UT	0.041060	-0.030568	-0.001596	0.004055	-0.003534	-0.002506	-0.002315	-0.004493	-0.028637	0.035901	...	0.086338	NaN	0.024432	0.019822	0.012984	0.058865
TOX_D_48_FON_UT	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
TOX_FIS_SUP_15_UT	0.019955	-0.014043	0.012397	0.009049	-0.000675	-0.003491	-0.003241	-0.002705	-0.040533	0.060292	...	1.000000	NaN	-0.009458	-0.014694	0.003146	0.036038
TOX_FIS_FON_15_UT	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN
CUMPLE_CON_DBO	0.007980	-0.007730	-0.179309	-0.121527	0.044036	-0.075993	-0.073780	-0.176172	0.504051	-0.368370	...	-0.009458	NaN	1.000000	0.669587	-0.043816	0.379738
CUMPLE_CON_DQO	0.010767	0.026398	-0.156833	-0.219944	-0.071791	-0.058833	-0.056487	-0.118604	0.448657	-0.262887	...	-0.014694	NaN	0.669587	1.000000	0.074288	0.265897
CUMPLE_CON_SST	0.042338	0.081863	-0.194836	-0.255178	-0.448025	-0.127415	-0.114791	0.003343	-0.188023	0.231953	...	0.003146	NaN	-0.043816	0.074288	1.000000	0.152618
CUMPLE_CON_CF	0.021044	0.016972	-0.051679	-0.006053	-0.088172	-0.042765	-0.039502	-0.070503	-0.050979	0.132686	...	0.036038	NaN	0.379738	0.265897	0.152618	1.000000
CUMPLE_CON_E_COLI	-0.044727	0.119817	-0.087758	-0.049777	-0.060550	-0.056647	-0.054004	-0.105788	0.124034	0.038929	...	0.038992	NaN	0.543922	0.425404	0.182446	0.696356

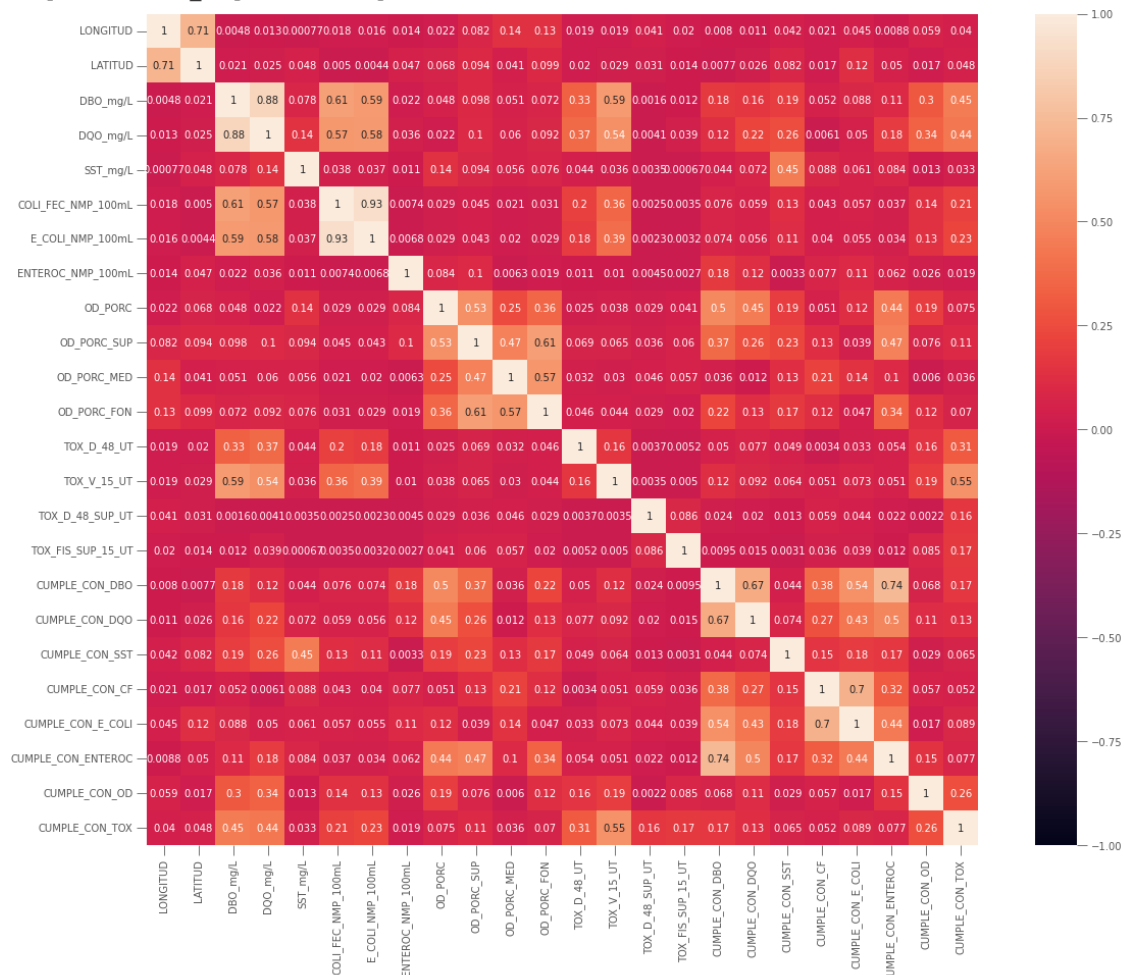
```
[41] correlacion = dfcorr.corr().abs()
```

S

```
f, ax = plt.subplots(figsize = (20,15))
```

```
sb.heatmap(correlacion, vmax = 1, vmin = -1, square = True, annot = True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fec8f24990>
```



A continuación realizamos la exploración de cada dato, con ayuda de `describe()`, nos encontramos con algunos datos nulos y datos <1 o <3, vamos a cambiarlos a 0 en el siguiente paso:

```
# Transformación completa de los resultados del analisis
df=pd.read_csv(path1)

df.rename(mapper=df['CLAVE'],axis=0,inplace=True)

df['Coordinates']= list(zip(df.LONGITUD, df.LATITUD))
df['Coordinates']= df['Coordinates'].apply(Point)

df.drop(['CLAVE','TIPO','CONTAMINANTES','PERIODO','SITIO','ESTADO','MUNICIPIO','CUENCA','CUERPO DE AGUA','ORGANISMO DE CUENCA'],axis=1,inplace=True)

df['SUBTIPO']=df['SUBTIPO'].astype("string")
df['SEMAFORO']=df['SEMAFORO'].astype("string")
df['GRUPO']=df['GRUPO'].astype("string")
df['SUBTIPO']=df['SUBTIPO'].str.upper()
df['SEMAFORO']=df['SEMAFORO'].replace({'Amarillo':'y','Rojo':'r','Verde':'g'})

localizacion=['LATITUD','LONGITUD']
for i in localizacion:
    df[i]=df[i].astype("float")

numericas=['DBO_mg/L','DQO_mg/L','SST_mg/L','COLI_FEC_NMP_100mL','E_COLI_NMP_100mL','ENTEROC_NMP_100mL','OD_PORC','OD_PORC_SUP','OD_PORC_INF']
for i in numericas:
    df[i]=df[i].replace((np.NaN:0))
    df[i]=df[i].replace({'<2':0,'<10':0,'<3':0,'<1':0})
    df[i]=df[i].astype("float")

calidades1=['CALIDAD_DBO','CALIDAD_DQO','CALIDAD_SST','CALIDAD_COLI_FEC','CALIDAD_E_COLI','CALIDAD_ENTEROC','CALIDAD_OD_PORC','CALIDAD_OD_PORC_SUP','CALIDAD_OD_PORC_INF']
for i in calidades1:
    df[i]=df[i].replace((np.NaN:'Sin Medida'))
    df[i]=df[i].astype("string")

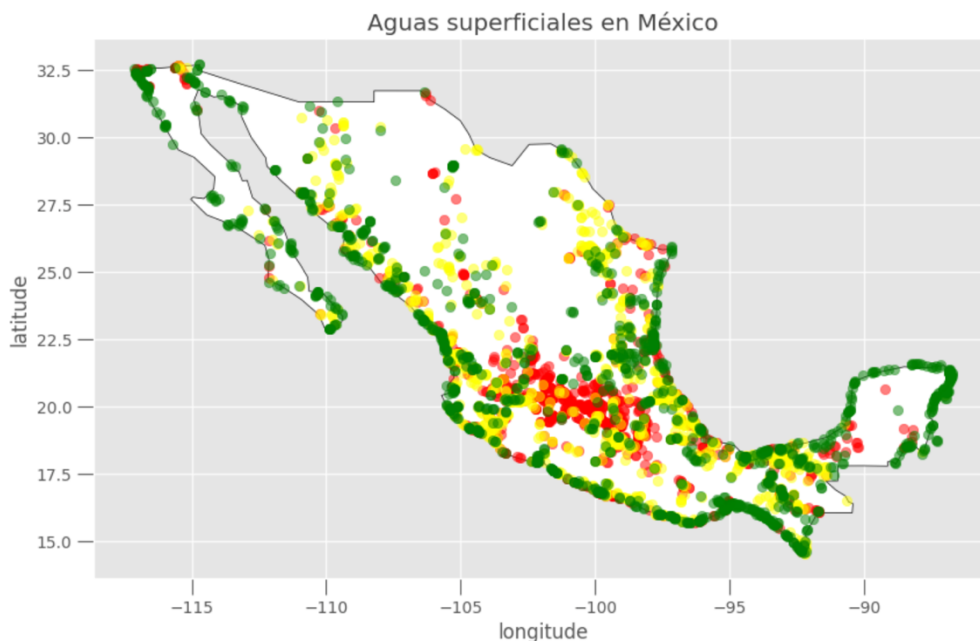
calidades2=['CALIDAD_TOX_V_15','CALIDAD_TOX_D_48','CALIDAD_TOX_D_48_SUP','CALIDAD_TOX_D_48_FON','CALIDAD_TOX_FIS_SUP_15','CALIDAD_TOX_FIS_SUP_48']
for i in calidades2:
    df[i]=df[i].replace((np.NaN:'Sin Medida'))
    df[i]=df[i].astype("string")

binarias=['CUMPLE_CON_DBO','CUMPLE_CON_DQO','CUMPLE_CON_SST','CUMPLE_CON_CF','CUMPLE_CON_E_COLI','CUMPLE_CON_ENTEROC','CUMPLE_CON_OD_PORC']
for i in binarias:
    df[i]=df[i].replace({'ND':0,'NO':0,'SI':1})
    df[i]=df[i].astype("float")

df.describe(include='all')
df.info()
```

Análisis relación entre la calidad del agua y su ubicación geográfica por medio de K- Means

Con los datos obtenidos, realizamos una gráfica con todos los puntos mapeados dentro de México, de acuerdo al valor de la columna Semáforo:

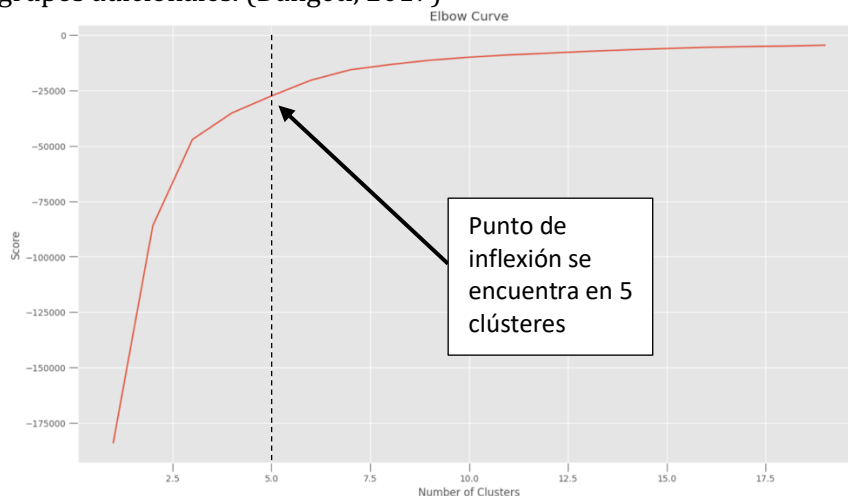


Con los datos del dataset hemos graficado todos los registros de acuerdo a su ubicación dentro de la República Mexicana y con base en el color del atributo 'Semáforo el color: verde, amarillo y rojo, el primero si no hay ningún indicador de contaminación; amarillo cuando uno o más indicadores indican contaminación y rojo cuando no se cumplen.

Los indicadores son Fluoruros, coliformes fecales, nitrógeno de nitratos, arsénico, cadmio, cromo, mercurio o plomo.

Gráfica de Codo

El método del codo se utiliza para determinar el número óptimo de conglomerados -*clusters*- en el agrupamiento de k-means. El método del codo traza el valor de la función de costo producida por diferentes valores de k; si la variable k aumenta, la distorsión promedio disminuirá, cada grupo tendrá menos instancias constituyentes y las instancias estarán más cerca de sus respectivos centroides. Sin embargo, las mejoras en la distorsión promedio disminuirán a medida que aumente el parentesco. El valor de k en el que la mejora en la distorsión disminuye más se denomina codo, en el que debemos dejar de dividir los datos en grupos adicionales. (Dangeti, 2017)



En nuestra gráfico podemos observar que el punto de inflexión es en el número 5.

Agrupamiento de latitudes y longitudes con K-Means en México

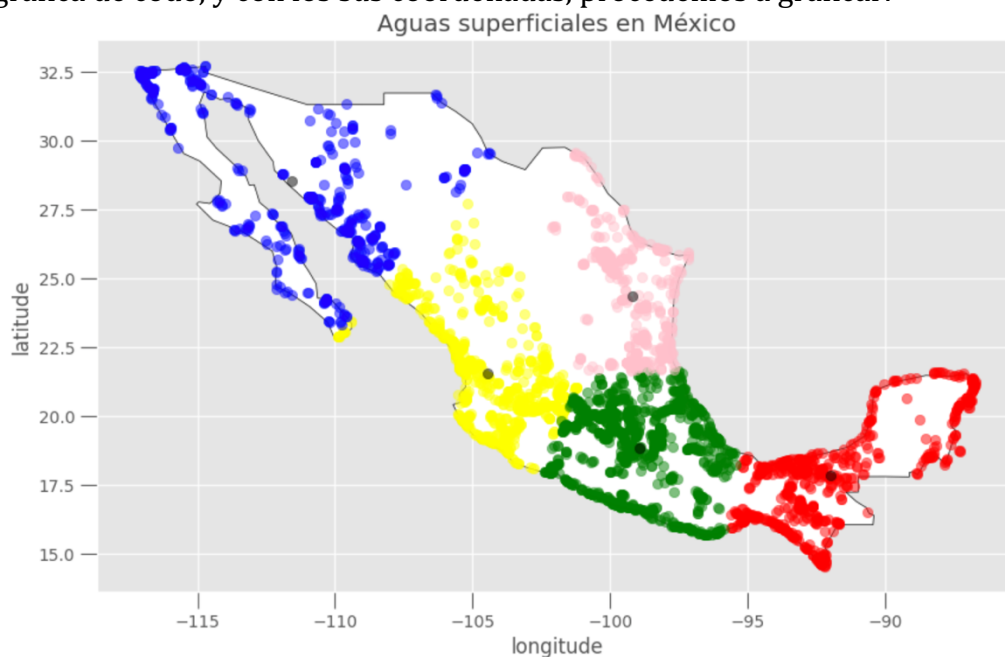
```
kmeans = KMeans(n_clusters=5).fit(latlong)
centroids = kmeans.cluster_centers_
print(centroids)
```

```
[[ 17.84328155 -91.98520227]
 [ 21.55627989 -104.46702415]
 [ 24.3756531  -99.18655067]
 [ 28.54666738 -111.59246864]
 [ 18.84640806 -98.92112128]]
```

```
df2=pd.DataFrame(centroids)
df2.columns=['latitude','longitude']
df2["Coordinates"] = list(zip(df2.longitude, df2.latitude))
df2["Coordinates"] = df2["Coordinates"].apply(Point)
df2.head()
```

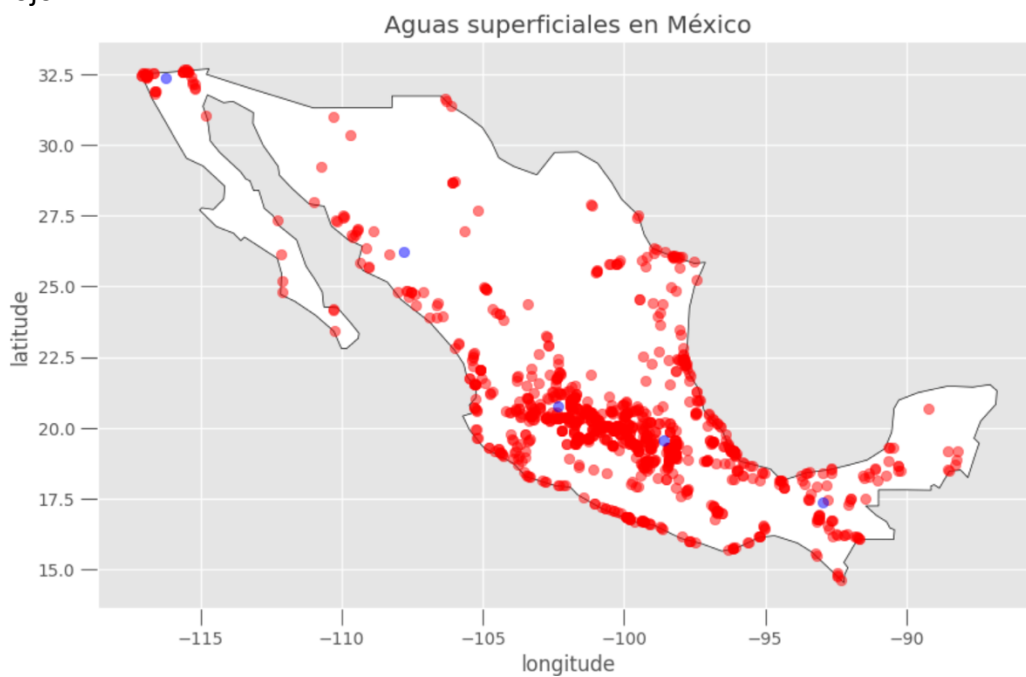
	latitude	longitude	Coordinates
0	17.843282	-91.985202	POINT (-91.98520227340268 17.843281545319467)
1	21.556280	-104.467024	POINT (-104.46702415286624 21.556279885350317)
2	24.375653	-99.186551	POINT (-99.18655066997519 24.375653101736972)
3	28.546667	-111.592469	POINT (-111.5924686440678 28.54666738498789)
4	18.846408	-98.921121	POINT (-98.92112127973749 18.846408055783428)

Teniendo en cuenta que 5, es el número óptimo de clústers, de acuerdo a nuestra gráfica de codo, y con los sus coordenadas, procedemos a graficar:

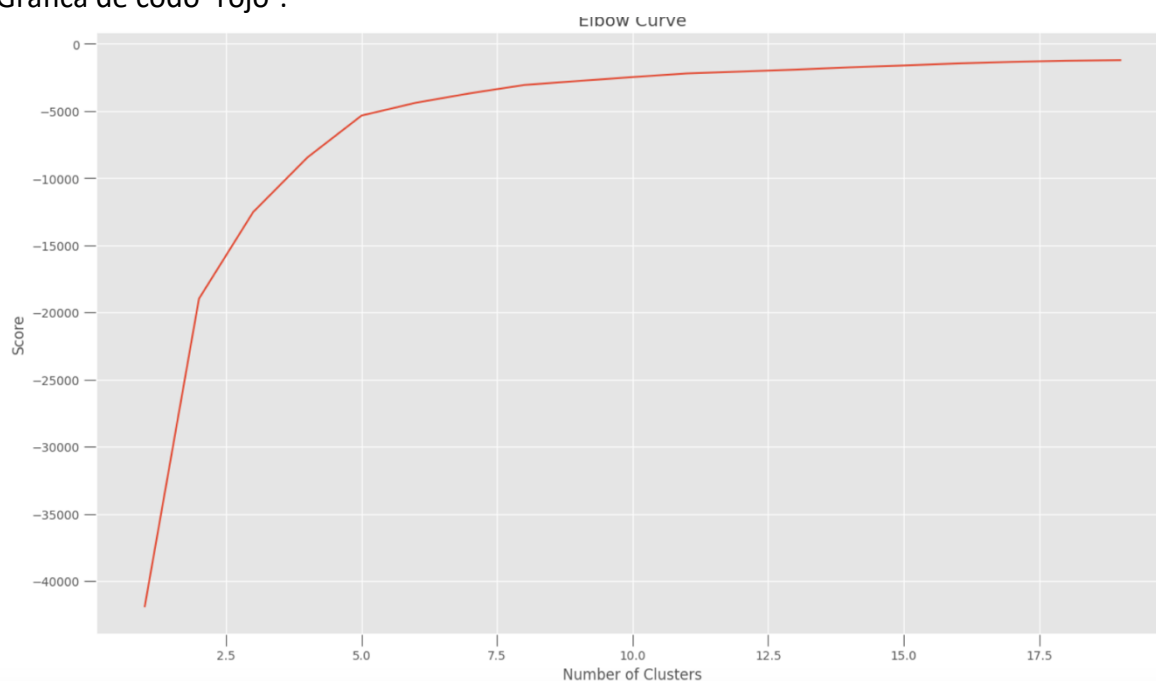


Agrupamiento por Cluster, de acuerdo al semáforo

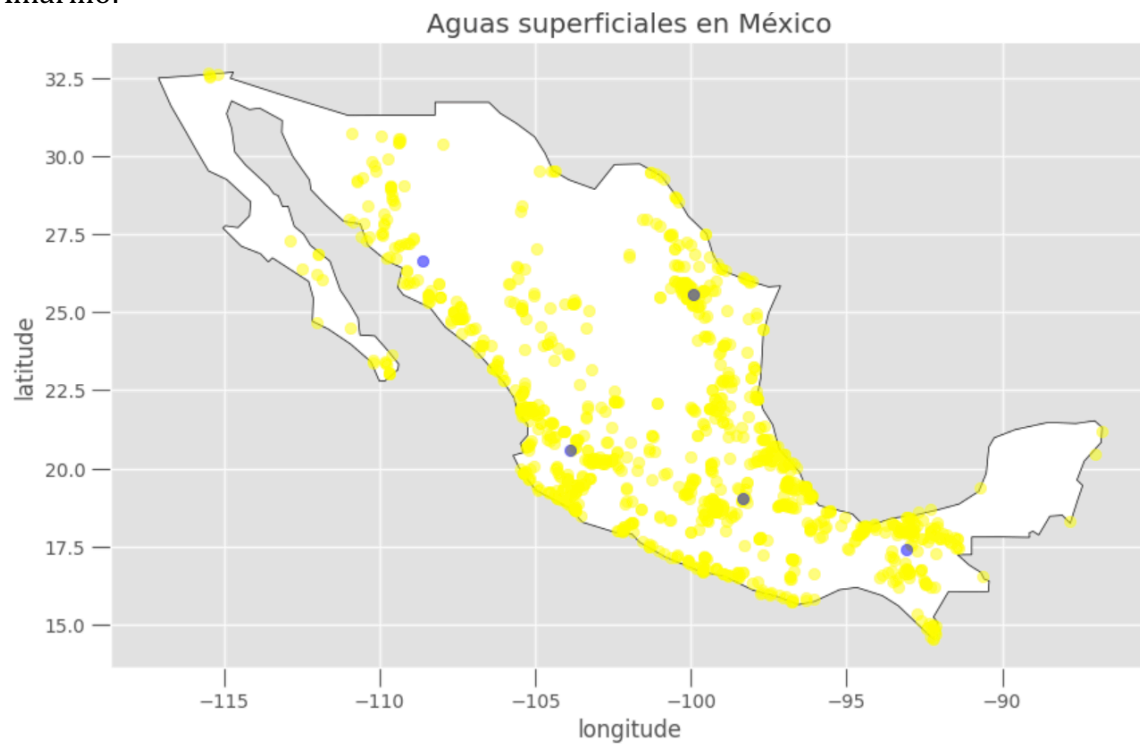
Rojo:



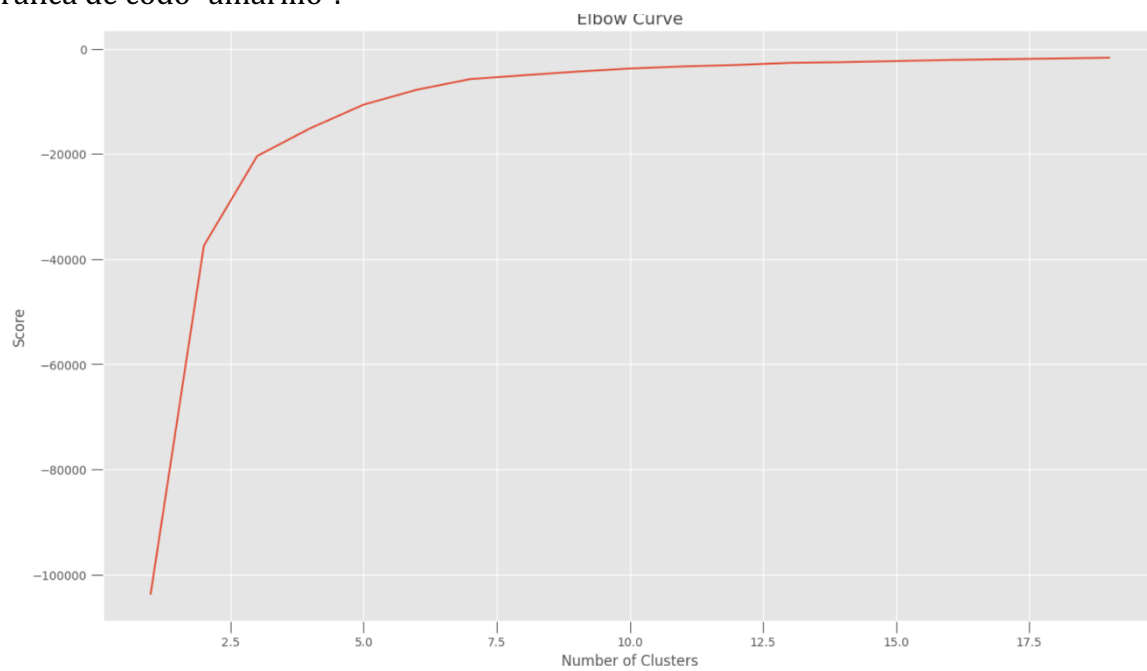
Gráfica de codo -rojo-:



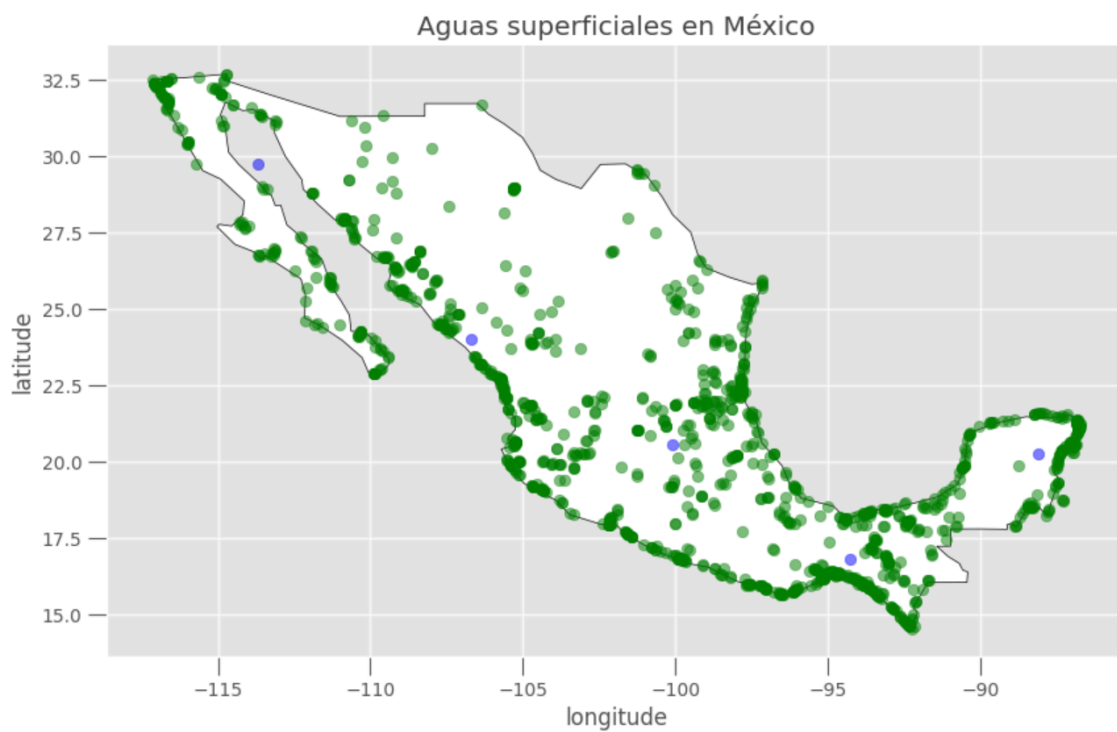
Amarillo:



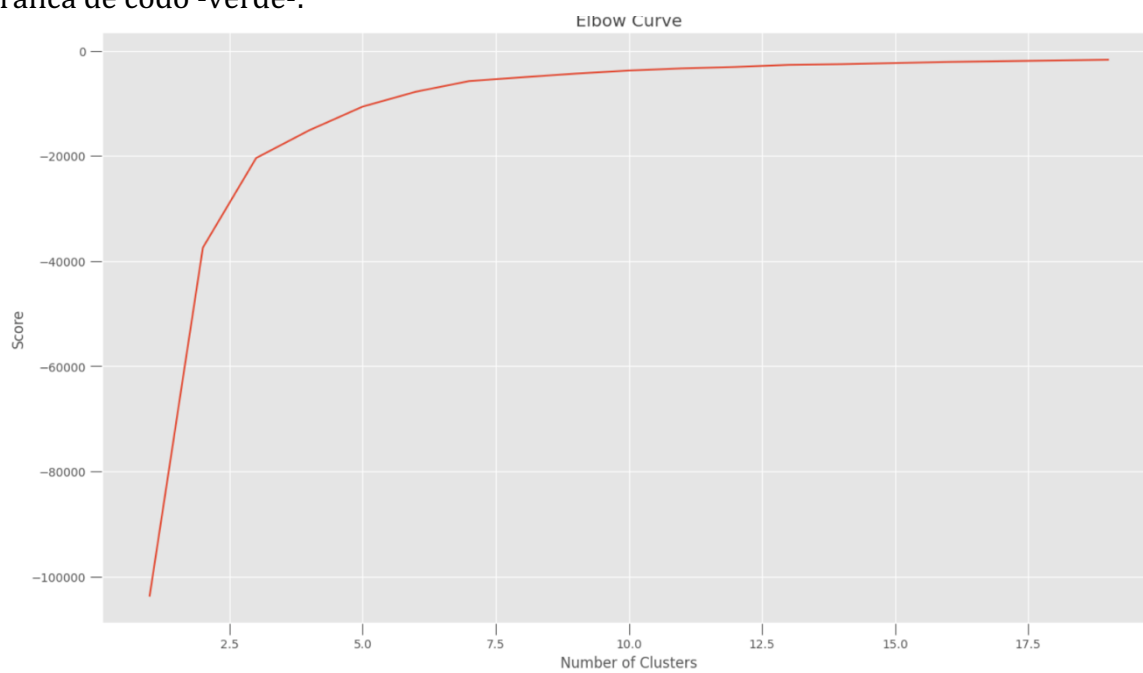
Gráfica de codo -amarillo-:



Verde:



Gráfica de codo -verde-:



Como pudimos observar en las 3 gráficas, cada una por cada color del semáforo, en las 3 pudimos observar, en conjunto que el número de clusters es 5.

REFERENCIAS

Dangeti,P. (2017). Statistics for Machine Learning. Packt Publishing
Código completo en [GitHub](#)