Tecnológico de Monterrey

TC2029 Ciencia y Analitica 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 monitoreeo 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_O
categoricas=['TIPO','SUBTIPO','SEMAFORO','CONTAMINANTES','GRUPO']
calidades=['CALIDAD_DBO','CALIDAD_DQO','CALIDAD_SST','CALIDAD_COLI_FEC','CALIDAD_E_COLI','CALIDAD_ENTEROC','CALIDAD_OD_PORC','CALID
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
identificador=['CLAVE']
nonecesarios=['PERIODO']
```

Se determina que el periodo no es necesario dado que solo contiene un dato (del año pero no es diferente) nota: podría ser util 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 contamientante en cu solumna 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 Toxico'})

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

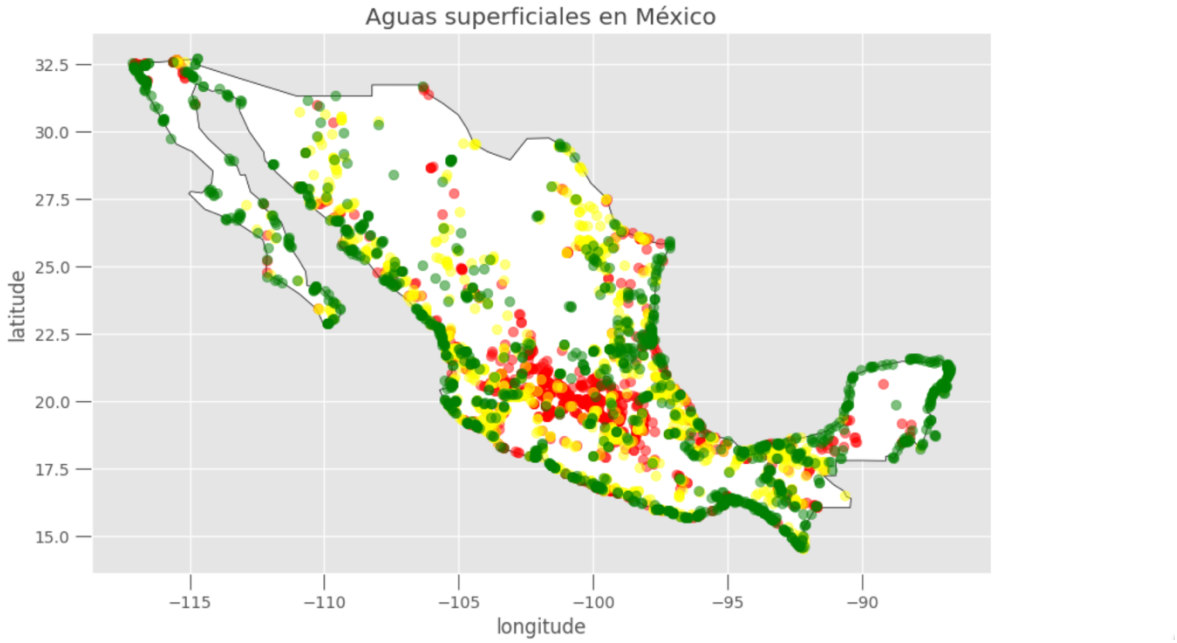
Explorar los Datos

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

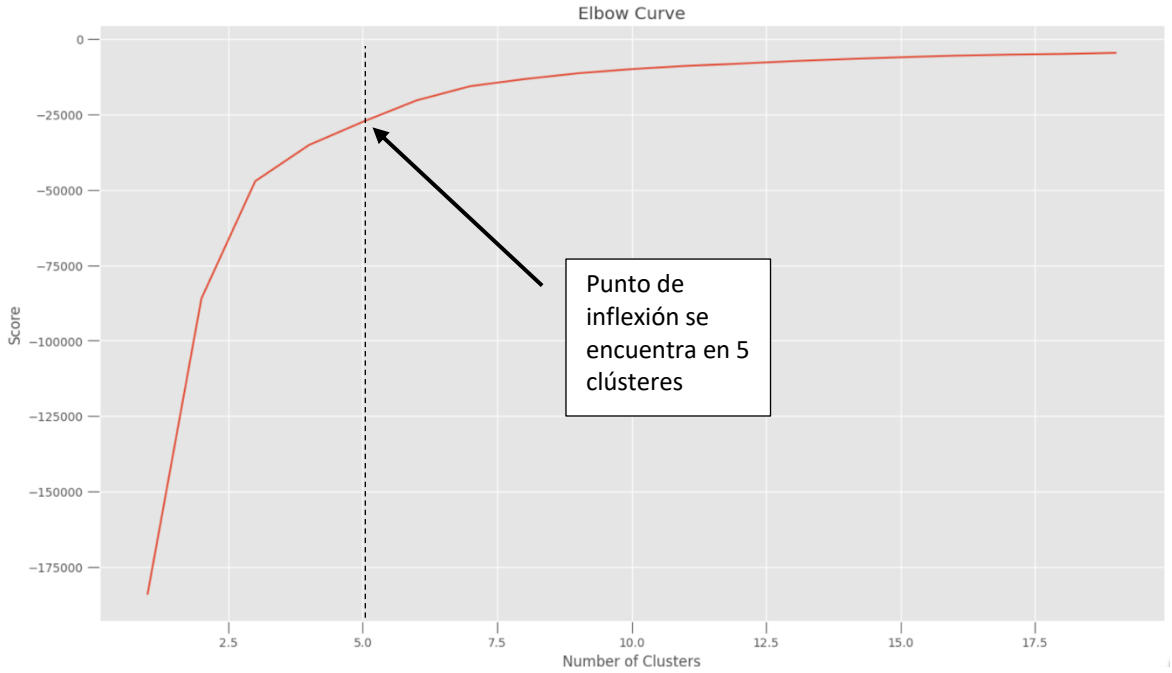
```
# Transformación completa de los resultados del analisis
df=pd.read_csv(path1)

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'],axi

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_
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','CALI
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_T
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_O
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 ubicción 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:



Gráfica de Codo



Agrupamiento de latitudes y longitudes con K-Means en México