Ciencia de datos

Práctica 3. Estadística descriptiva básica

Alberto Benavides

A partir de los datos limpios, es posible obtener un panorama de los mismos mediante una descripción estadística básica. Los archivos de datos con los que se trabajará son `enf.csv` y `semanalesTodas.csv`. El primero contiene 784660 registros entre los que figuran

In [242]:
```
import pandas as pd
data1 = pd.read_csv("D:/FIME/Epidemia/Data/csvSemanales/enf.csv")
print(data1.sample(10))
```

```
        anio  sem               estado                          enfermedad  \
428761  2011  47   Baja California Sur  influenza a h1n1
714876  2014  36   Hidalgo              anorexia y bulimia
405223  2011  22   Guerrero             escarlatina
663938  2014  24   Guanajuato           tumor maligno del cuello del utero
336686  2010  10   Tlaxcala             sida
415084  2011  33   Oaxaca               sifilis adquirida
572615  2014  2    Morelos              hipertension arterial
777015  2014  52   Queretaro            fiebre tifoidea
569600  2014  1    Veracruz             fiebre del oeste del nilo
638537  2014  18   Chihuahua            adicciones

        casos      cie
428761  0.0    j09
714876  1.0    f50
405223  0.0    a38
663938  3.0    c53
336686  0.0    b20-b24
415084  3.0    a51-a53
572615  157.0  i10-i15
777015  1.0    a01.0
569600  0.0    a92.3
638537  0.0    f10-f19
```

A su vez, `semanalesTodas.csv` tiene 138 registros de los datoa anteriores preprocesados, como por ejemplo

In [243]:
```
data2 = pd.read_csv("D:/FIME/Epidemia/Data/semanalesTodasKmeans.csv")
print(data2.sample(10))
```

```
          # m        f1        f2        f3        f4       ac1       ac2  \
84   0.089752  3.145485  2.641810  2.103169  0.014584 -2.535501  0.872935
115  0.117341 -0.669638 -0.726289 -0.813300 -1.116549 -1.166307 -1.125164
95   0.182978 -0.250573 -0.772247 -0.255807  0.156531  0.403418 -0.120376
93   0.104781 -0.540835  1.591331 -0.624397 -0.316623 -0.125199 -0.004954
40   0.133238 -0.250573 -0.772247  1.710007  1.575993  0.077635  0.515794
42   0.117326 -0.560573 -0.648291 -0.647987  1.585929 -0.870864 -0.001710
3    0.116887  3.110109 -0.571179  0.665668  0.234157 -1.568089 -0.560935
57   0.118971 -0.250573  0.146922  1.587143  0.393108 -0.753710 -1.021002
34   0.112843 -0.540835  1.066092  1.341417  1.339416 -0.278225 -0.255104
113 -0.249967 -0.540835 -0.247008 -0.501534 -1.026354  1.287090  1.015665

          ac3       ac4       ac5       ac6  \
84  -2.123539  1.168302 -1.993798  1.591690
115 -0.853290 -0.935573 -0.687915 -0.623106
95  -0.295535  0.019661  0.411640  0.019916
93  -0.098657 -0.342742  0.189497  0.188404
40   0.247593  0.269593  0.234986 -0.182035
42  -0.620251 -0.717385 -0.675323 -0.442638
3   -1.147260 -0.068722 -1.248616 -0.192296
57  -1.028807 -1.271171 -0.613808 -1.054068
34   0.322193 -0.157418  0.047109  0.017672
113  0.693781  0.178158 -0.096798 -0.402128

                                          enf cie  cluster
84   ENFERMEDAD ALCOHOLICA DEL HIGADO          K   7
115  TETANOS NEONATAL                          A   12
95   ESCABIOSIS                                B   3
93   ENFERMEDAD DE PARKINSON                   G   12
40   DIABETES MELLITUS EN EL EMBARAZO          O   10
42   PALUDISMO POR P. FALCIPARUM               B   0
3    TIFO MURINO                               A   7
57   DISPLASIA CERVICAL SEVERA Y CACU IN SITU  N   6
34   ENFERMEDAD DE ALZHEIMER                   G   6
113  INFLUENZA                                 J   3
```

donde

- m: Pendiente de la regresión lineal,
- $f_i, i \in [1, 4]$: Frecuencia i de Fourier (ordenadas de mayor a menor),
- $ac_j, j \in [1, 6]$: Autocorrelaciones para los retrasos de 2^{j-1} semanas,
- enf: Nombre de la enfermedad,

- cie: Primera letra de la CIE correspondiente a la enfermedad,
- cluster: Agrupación asignada por algoritmo de k-medias.

Es de interés en este estudio conocer los datos obtenidos nacionalmente, por lo que primero se importan y eligen los registros que cuenten el total de casos de enfermedades registrados por año y semana epidemiológica, lo que se logran con

In [244]:
```
# https://cmdlinetips.com/2018/02/how-to-subset-pandas-dataframe-based-on-values-of-a-column/
data = data1[data1.estado == "TOTAL"]
print(data.sample(10))
```

```
        anio  sem estado                         enfermedad     casos      cie
29038   2005  24    TOTAL  tuberculosis respiratoria          242.0     a15-a16
550946  2013  38    TOTAL  hepatitis virica c                  58.0      b17.1
49696   2005  39    TOTAL  infecciones respiratorias agudas  444482.0  j00-j06
738006  2014  42    TOTAL  leishmaniasis cutanea               6.0       b55.1
562043  2013  46    TOTAL  meningitis meningococica            6.0       a39.0
684704  2014  29    TOTAL  enfermedad cerebrovascular          736.0     i60-i67
335344  2010  9     TOTAL  brucelosis                          36.0      a23
742024  2014  43    TOTAL  escabiosis                          1915.0    b86
41247   2005  33    TOTAL  colera                              0.0       a00
264096  2009  2     TOTAL  varicela                            6500.0    b01
```

Ahora, se sobreescribe la primer letra de la CIE asignada a cada enfermedad resultante en lugar de su CIE

In [245]:
```
# https://stackoverflow.com/a/35552899
# https://stackoverflow.com/a/28541443
data.loc[:, 'cie'] = data['cie'].astype(str).str[0]
print(data.sample(10))
```

```
        anio  sem estado                               enfermedad     casos cie
263667  2009  2     TOTAL  intoxicacion alimentaria bacteriana   508.0     a
481904  2012  39    TOTAL  parotiditis infecciosa                83.0      b
447144  2012  12    TOTAL  meningitis tuberculosa                2.0       a
321947  2009  49    TOTAL  rubeola congenita                     0.0       p
232748  2008  28    TOTAL  intoxicacion por veneno de escorpion  5230.0    t
267033  2009  5     TOTAL  meningitis tuberculosa                4.0       a
742684  2014  43    TOTAL  hipertension arterial                9344.0    i
282643  2009  17    TOTAL  hepatitis virica c                    25.0      b
280928  2009  16    TOTAL  infecciones respiratorias agudas      458439.0  j
736793  2014  42    TOTAL  neumonias y bronconeumonias           2863.0    j
```

Con ello, se puede obtener la cantidad de registros existentes por cada letra inicial de CIE

In [246]:
```
 # https://stackoverflow.com/a/22391554
print(data['cie'].value_counts())
```

```
a    10017
b    5010
t    1460
j    1242
u    999
w    681
i    665
g    548
x    454
p    440
z    406
e    364
q    208
n    208
f    207
k    181
y    111
s    104
h    104
v    104
c    104
o    52
r    52
Name: cie, dtype: int64
```

De donde se puede observar que los que inician con la letra A son los más numerosos. Los porcentajes de aparición son

In [247]:
```
# https://stackoverflow.com/a/24167876
print(data['cie'].value_counts(normalize=True))
```

```
a    0.422284
b    0.211205
t    0.061549
j    0.052359
u    0.042115
w    0.028709
i    0.028034
g    0.023102
x    0.019139
p    0.018549
z    0.017116
e    0.015345
q    0.008769
n    0.008769
f    0.008726
k    0.007630
y    0.004679
s    0.004384
h    0.004384
```

```
v    0.004384
c    0.004384
o    0.002192
r    0.002192
Name: cie, dtype: float64
```

De modo que las enfermedades cuya CIE inicia con la letra A ocupan el 42.23% de los resgistros.

Ahora bien, se puede obtener una descripción de los datos agrupados por número de casos registrados y letra inicial de CIE

In [248]:
```
print(data.groupby(['cie'])['casos'].describe())
```

```
       count           mean            std     min      25%      50%  \
cie
a    10017.0  8238.886593    27267.391977    0.0     0.00     28.0
b    5010.0   1085.947305    2440.398179     0.0     5.00     37.0
c    104.0    119.490385     74.104622       24.0    49.75    80.0
e    364.0    2311.829670    3049.646005     5.0     69.00    295.0
f    207.0    543.859903     545.763387      1.0     52.00    156.0
g    548.0    26.885036      27.256107       0.0     13.00    18.0
h    104.0    9997.173077    2382.303338     1685.0  8332.75  10394.5
i    665.0    1000.296241    2406.767605     2.0     14.00    19.0
j    1242.0   166282.132045  241732.275705   0.0     1787.00  3366.0
k    181.0    12312.386740   12732.622177    0.0     37.00    6570.0
n    208.0    20159.552885   34267.122104    25.0    113.75   865.5
o    52.0     186.134615     32.418240       103.0   173.25   187.5
p    440.0    0.061364       0.249570        0.0     0.00     0.0
q    208.0    10.543269      14.104786       0.0     2.00     4.0
r    52.0     1563.538462    1384.661047     86.0    483.75   1204.5
s    104.0    547.019231     430.921793      13.0    148.75   395.0
t    1460.0   1960.066438    2398.014683     0.0     58.00    764.0
u    999.0    49.612613      94.266013       0.0     5.00     15.0
v    104.0    1277.625000    814.812709      271.0   486.50   1127.0
w    681.0    1501.004405    958.612725      90.0    225.00   1939.0
x    454.0    74.259912      32.156191       15.0    48.00    72.5
y    111.0    153.657658     248.992022      3.0     24.50    79.0
z    406.0    86.403941      35.275416       0.0     63.00    83.5

           75%        max
cie
a    296.00     149993.0
b    298.00     23013.0
c    194.00     321.0
e    5450.25    8973.0
f    896.00     1698.0
g    26.00      159.0
h    11686.50   13637.0
i    758.00     10642.0
j    384079.00  1184372.0
k    23942.00   32657.0
n    12181.25   91116.0
o    210.00     242.0
p    0.00       2.0
q    12.75      60.0
r    2120.75    5174.0
s    973.25     1193.0
t    4019.50    8743.0
u    54.00      1097.0
v    2067.75    2517.0
w    2282.00    3021.0
x    98.75      167.0
y    183.50     1370.0
z    105.00     189.0
```

Al preprocesar los datos, se agrupan las enfermedades por primera letra de la CIE y se descubre que los grupos A y B contienen la mayoría de los registros, contando un 31.19% y un 19.57% respectivamente.

In [249]:
```
# https://stackoverflow.com/a/51453257
pd.options.display.max_columns
# pd.set_option('display.max_columns', 500)
data = data2
print(data['cie'].value_counts())
```

```
A    44
B    27
T    8
E    7
J    6
I    5
Q    4
F    4
N    4
K    4
W    3
G    3
U    3
Y    3
V    2
H    2
C    2
S    2
X    1
Z    1
O    1
P    1
R    1
Name: cie, dtype: int64
```

In [250]:
```
print(data['cie'].value_counts(normalize = True))
```

```
A    0.318841
B    0.195652
T    0.057971
E    0.050725
J    0.043478
I    0.036232
Q    0.028986
F    0.028986
N    0.028986
K    0.028986
W    0.021739
G    0.021739
U    0.021739
Y    0.021739
V    0.014493
H    0.014493
C    0.014493
S    0.014493
X    0.007246
Z    0.007246
O    0.007246
P    0.007246
R    0.007246
Name: cie, dtype: float64
```

También se obtuvo la correlación existente entre la primera letra de la CIE por enfermedad y el cluster asignado por k-medias

In [251]:
```
# https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.factorize.html
data['cie'], uniques = pd.factorize(data['cie'])

print(data['cluster'].corr(data['cie']))
```

```
-0.1378082827244919
```

Finalmente, se seleccionan las características de los datos

In [252]:
```
features = ["# m", "f1", "f2", "f3", "f4", "ac1", "ac2", "ac3", "ac4", "ac5", "ac6"]
x = data.loc[:, features].values
```

se normalizan

In [253]:
```
# https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
from sklearn.preprocessing import MinMaxScaler
x = MinMaxScaler().fit(x).transform(x)
```

y con estas características normalizadas se puede hacer una selección a partir del umbral de varianza

In [254]:
```
# https://stackoverflow.com/a/7670325
print("Columnas iniciales = {}".format(x.shape[1]))

# https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html#sklearn.feature_selection.VarianceThresho
from sklearn.feature_selection import VarianceThreshold
th = 0.05 # .8 * (1 - .8)
print("Umbral de varianza = {}".format(th))
sel = VarianceThreshold(threshold=th)
x = sel.fit_transform(x)
print("Columnas finales")
# https://stackoverflow.com/q/39812885
print(data[data.columns[sel.get_support(indices=True)]].sample(10))
```

```
Columnas iniciales = 11
Umbral de varianza = 0.05
Columnas finales
           f1        f2        f3        f4       ac3       ac4
26   0.039688  1.328711  1.832870  1.221127 -0.207999 -0.650651
45   0.329949 -0.772247  2.078597 -0.789777 -0.431074 -0.332867
25  -0.468270 -0.755833 -0.785655 -1.152036  1.814904  1.658516
124 -0.540835  0.278232  1.710007  0.038242 -0.546520 -0.300950
90  -0.653311 -0.844468 -0.716545 -0.837093 -0.247101  0.050031
77  -0.294113 -0.194484  0.997399  0.972721 -0.505118 -0.696216
80  -0.653311 -0.770606 -0.937699 -0.531021  0.488841  0.531799
12   1.456164 -0.352056  0.083296 -0.397060 -0.848144 -0.947685
46  -0.669638 -0.741061 -0.965343 -1.143164  0.961513  1.015570
56  -0.540835  0.015612  0.849963  0.984550  0.027232 -0.386948
```