Ciencia de datos

Práctica 5. Pruebas estadísticas

Alberto Benavides

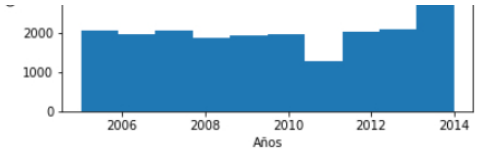
De los datos correspondientes a los registros obtenidos de los PDFs de la Secretaría de Salud de México

In [11]:
```
import pandas as pd

data1 = pd.read_csv("D:/FIME/Epidemia/Data/csvSemanales/nacional.csv")
data1.loc[:, 'cie'] = data1['cie'].astype(str).str[0]

print(data1.sample(3))
```

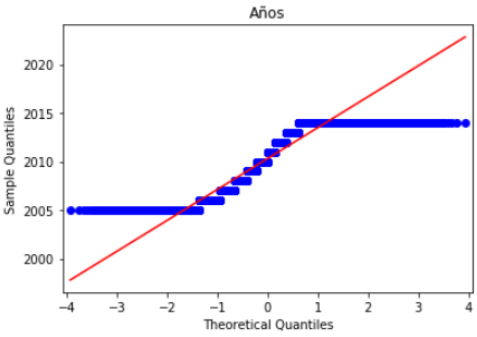
```
       anio  sem             enfermedad  casos cie
1356   2005   35                   sida     59   b
18059  2014    8  hipertension arterial  10243   i
22072  2014   40  leishmaniasis cutanea      7   b
```

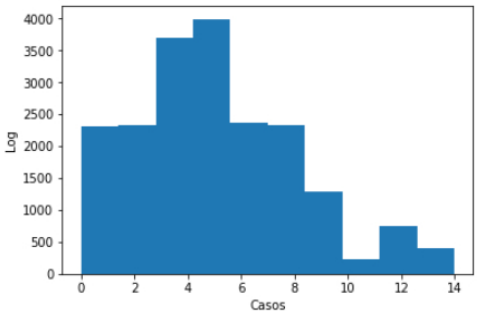
se muestran los histogramas y diatramas de cuartil-cuartil

In [107]:
```
import matplotlib as mpl
import matplotlib.pyplot as plt
import numpy as np

fig = plt.figure()
plt.hist(data1['anio'])
plt.xlabel('Años')
plt.ylabel('Cantidad')
plt.savefig("P5/hist1Anio.png")
plt.show()
plt.close(fig)

from statsmodels.graphics.gofplots import qqplot
fig = plt.figure()
qqplot(data1['anio'], line='s')
plt.title('Años')
plt.savefig("P5/qq1Anio.png")
plt.show()
plt.close(fig)


fig = plt.figure()
plt.hist(np.log(data1[data1['casos'] > 0]['casos']))
plt.xlabel('Casos')
plt.ylabel('Log')
plt.savefig("P5/hist1Casos.png")
plt.show()
plt.close(fig)

fig = plt.figure()
qqplot(np.log(data1[data1['casos'] > 0]['casos']), line='s')
plt.title('Casos')
plt.savefig("P5/qq1Casos.png")
plt.show()
plt.close(fig)


# https://stackoverflow.com/a/34682518
classnames, indices = np.unique(data1['cie'], return_inverse=True)
fig = plt.figure()
plt.hist(indices, bins=len(classnames))
plt.xticks(range(len(classnames)), classnames)
plt.xlabel('CIE')
plt.ylabel('Cantidad')
plt.savefig("P5/hist1Cie.png")
plt.show()
plt.close(fig)

fig = plt.figure()
qqplot(indices, line='s')
plt.title('CIEs')
plt.savefig("P5/qq1Cie.png")
plt.show()
plt.close(fig)
```

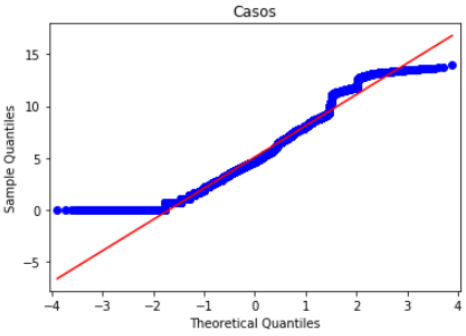


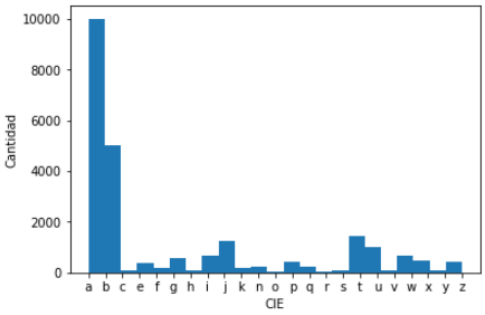

```
<Figure size 432x288 with 0 Axes>
```



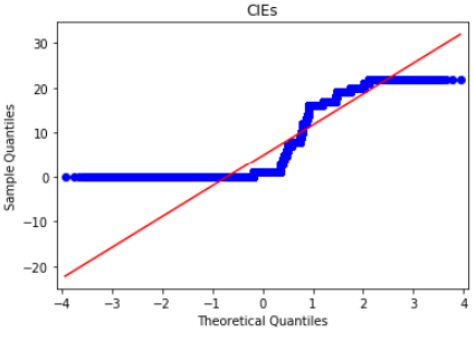



```
<Figure size 432x288 with 0 Axes>
```





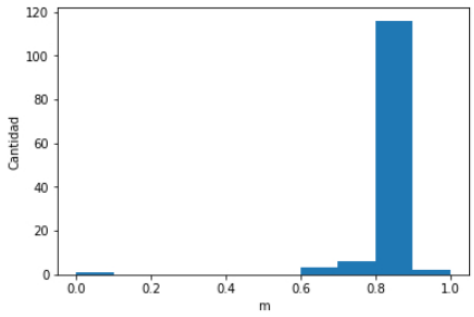

```
<Figure size 432x288 with 0 Axes>
```



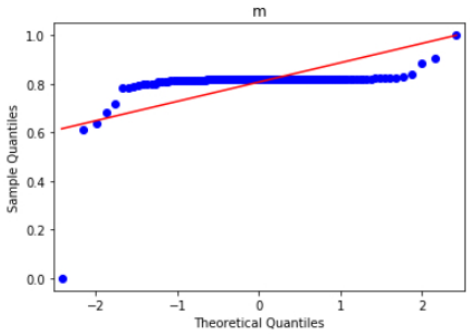
y lo mismo para los preprocesados

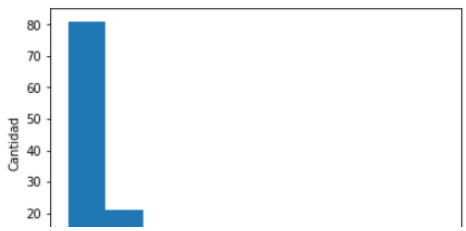
In [174]:
```
data2 = pd.read_csv("D:/FIME/Epidemia/Data/semanalesTodasKmeans.csv")
print(data2.sample(3))

for col in data2:
    if col != 'enf' and col != 'cie' and col != 'cluster':
        fig = plt.figure()
        plt.hist(data2[col])
        plt.xlabel(col)
        plt.ylabel('Cantidad')
        plt.savefig("P5/hist2{}.png".format(col))
        plt.show()
        plt.close(fig)

        fig = plt.figure()
        qqplot(data2[col], line='s')
        plt.title(col)
        plt.savefig("P5/qq2{}.png".format(col))
        plt.show()
        plt.close(fig)

classnames, indices = np.unique(data2['cie'], return_inverse=True)
fig = plt.figure()
plt.hist(indices, bins=len(classnames))
plt.xticks(range(len(classnames)), classnames)
plt.xlabel('CIE')
plt.ylabel('Cantidad')
plt.savefig("P5/hist2Cie.png")
plt.show()
plt.close(fig)

fig = plt.figure()
plt.hist(data2['cluster'], bins=len(classnames))
plt.xlabel('Cluster')
plt.ylabel('Cantidad')
plt.savefig("P5/hist2Cluster.png")
plt.show()
plt.close(fig)
```

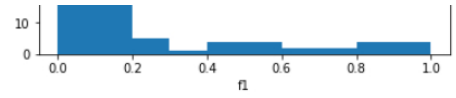
```
           m        f1        f2        f3        f4        ac1       ac2  \
71  0.815983  0.040446  0.218968  0.175484  0.352679  0.656818  0.351859
54  0.817569  0.019061  0.014412  0.023623  0.051339  0.642375  0.236239
10  0.816827  0.033008  0.479312  0.070494  0.293155  0.735702  0.407004

         ac4       ac8       ac16      ac32                  enf cie  cluster
71  0.059247  0.479243  0.203499  0.255384            ASCARIASIS   B        5
54  0.196733  0.329632  0.272969  0.346732               COLERA   A        6
10  0.481491  0.405325  0.139636  0.203136  DESNUTRICION MODERADA   E        6
```



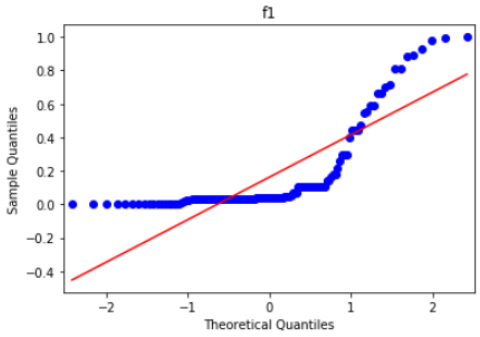

```
<Figure size 432x288 with 0 Axes>
```

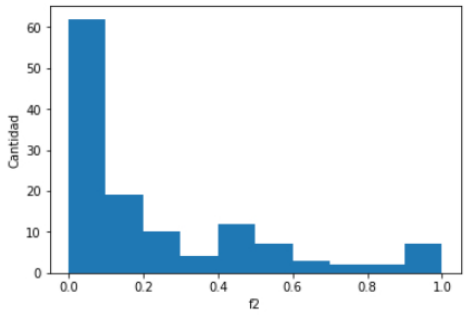




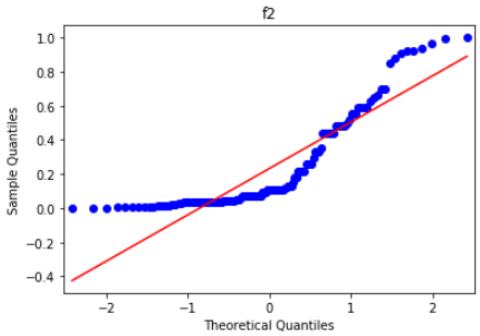


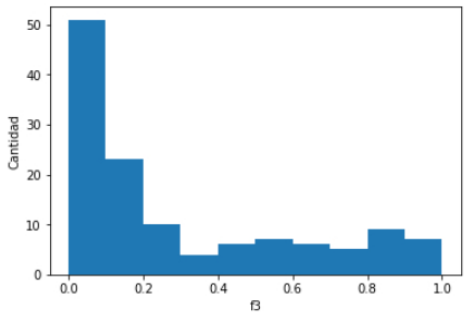

```
<Figure size 432x288 with 0 Axes>
```



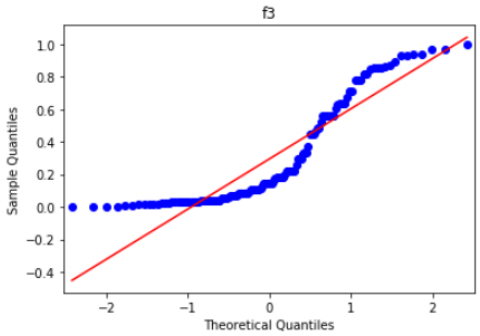


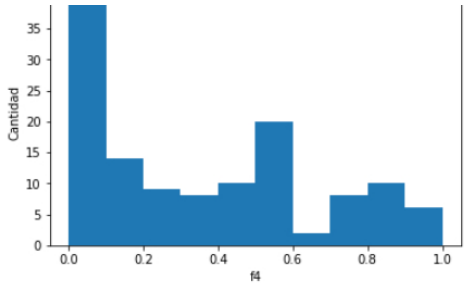

```
<Figure size 432x288 with 0 Axes>
```



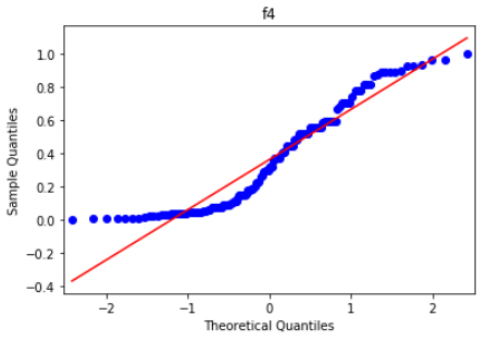


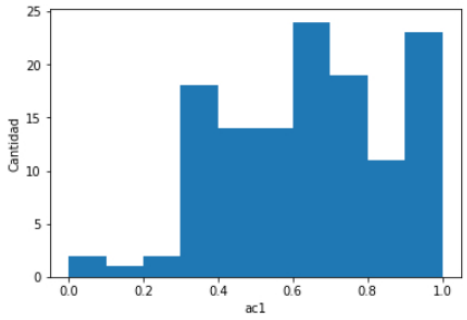

```
<Figure size 432x288 with 0 Axes>
```



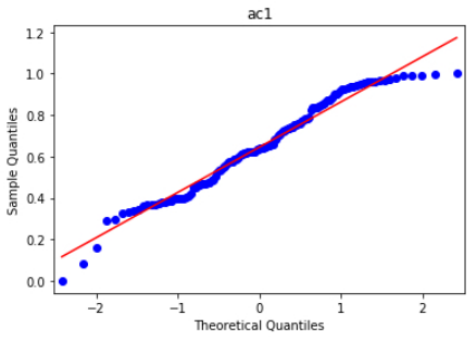


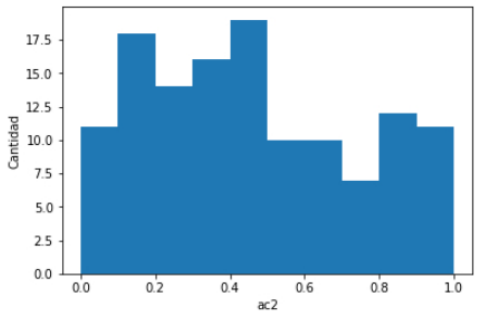
<Figure size 432x288 with 0 Axes>



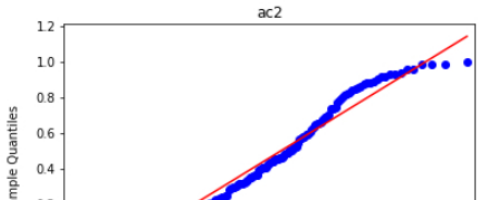


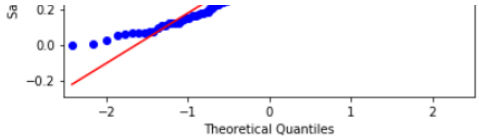
<Figure size 432x288 with 0 Axes>

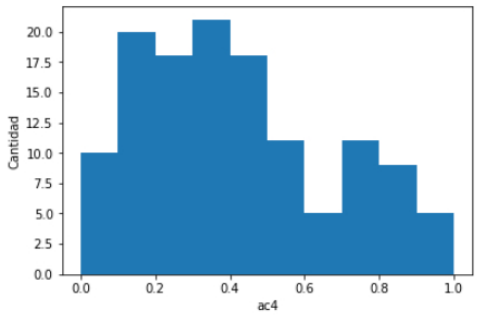




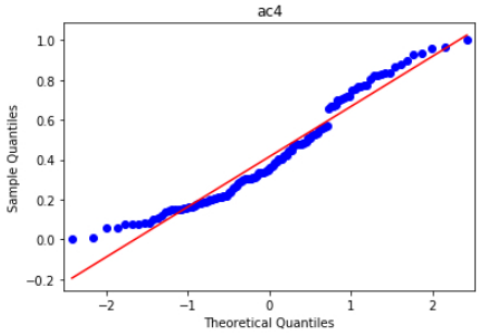
<Figure size 432x288 with 0 Axes>

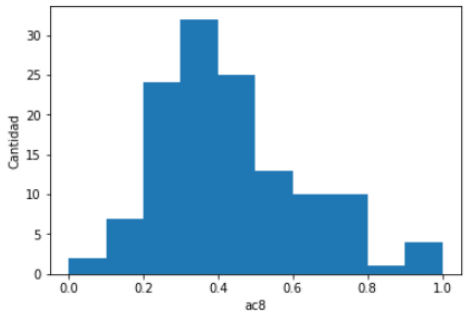




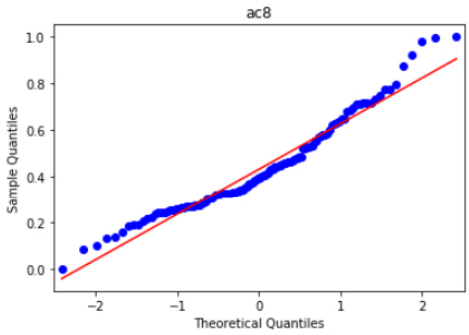


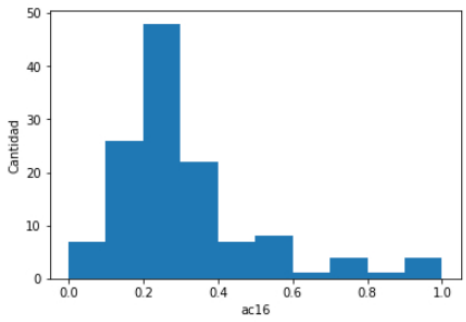

```
<Figure size 432x288 with 0 Axes>
```



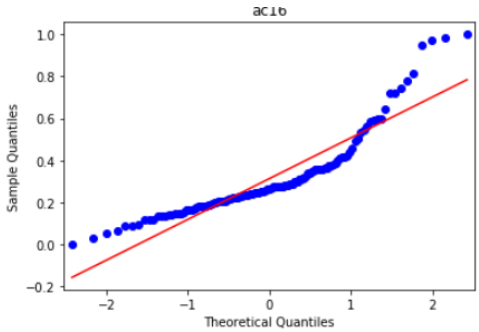


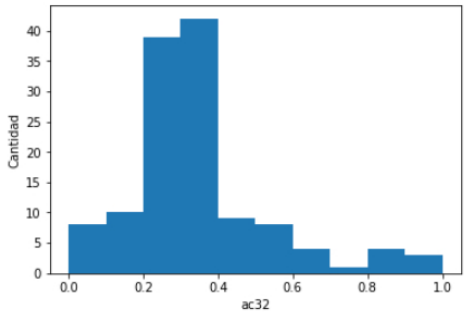

```
<Figure size 432x288 with 0 Axes>
```



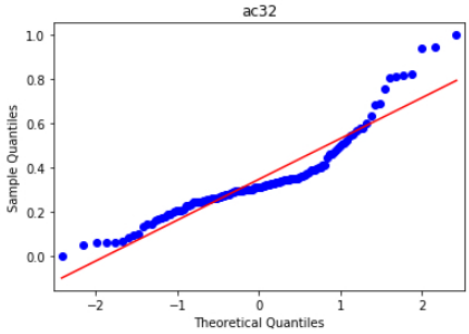


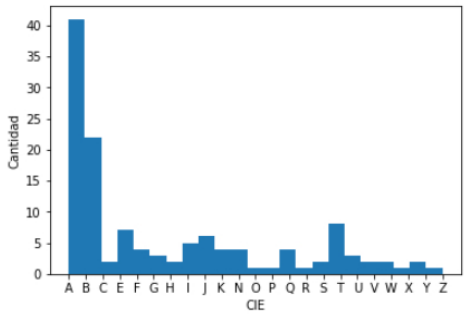

```
<Figure size 432x288 with 0 Axes>
```

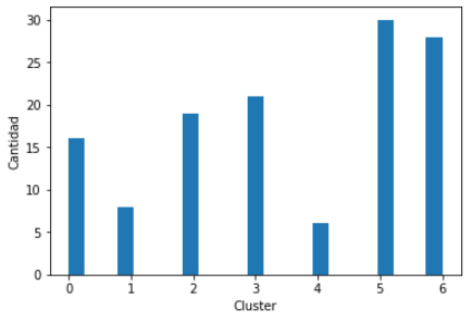





```
<Figure size 432x288 with 0 Axes>
```







Pruebas de normalidad

Primero se comprueba la normalidad de los datos extraídos y depurados de los PDFs de las semanas epidemiológicas. Por tratarse de datos con valor superior a 5000, se utilizará la prueba de Anderson-Darling

In [145]:
```
from scipy.stats import anderson
for col in data1:
    if col != 'enfermedad':
        if col == 'cie':
```

```
            classnames, indices = np.unique(data1['cie'], return_inverse=True)
            s, p, l = anderson(indices)
        else:
            s, p, l = anderson(data1[col], dist='norm')
        print('{} {}'.format(col, p))
        # https://stackoverflow.com/a/10580782
        for i in range(len(l)):
            if p[i] < l[i]:
                print('Parece ser normal con niveles de significancia {}\n'.format(l[i]))
            else:
                print('No parece ser normal con niveles de significancia {}\n'.format(l[i]))
```

```
anio [0.576 0.656 0.787 0.918 1.092]
Parece ser normal con niveles de significancia 15.0

Parece ser normal con niveles de significancia 10.0

Parece ser normal con niveles de significancia 5.0

Parece ser normal con niveles de significancia 2.5

No parece ser normal con niveles de significancia 1.0

sem [0.576 0.656 0.787 0.918 1.092]
Parece ser normal con niveles de significancia 15.0

Parece ser normal con niveles de significancia 10.0

Parece ser normal con niveles de significancia 5.0

Parece ser normal con niveles de significancia 2.5

No parece ser normal con niveles de significancia 1.0

casos [0.576 0.656 0.787 0.918 1.092]
Parece ser normal con niveles de significancia 15.0

Parece ser normal con niveles de significancia 10.0

Parece ser normal con niveles de significancia 5.0

Parece ser normal con niveles de significancia 2.5

No parece ser normal con niveles de significancia 1.0

cie [0.576 0.656 0.787 0.918 1.092]
Parece ser normal con niveles de significancia 15.0

Parece ser normal con niveles de significancia 10.0

Parece ser normal con niveles de significancia 5.0

Parece ser normal con niveles de significancia 2.5

No parece ser normal con niveles de significancia 1.0
```

Ahora se realiza la prueba de Shapiro-Wilk para comprobar la normalidad del segundo conjunto de datos, que serán los que se utilizarán como parámetros de k-medias

In [115]:
```
from scipy.stats import shapiro
# https://docs.scipy.org/doc/numpy/reference/generated/numpy.asarray.html
for alpha in [0.05, 0.01]:
    for col in data2:
        if col != 'enf' and col != 'cluster':
            if col == 'cie':
                classnames, indices = np.unique(data2['cie'], return_inverse=True)
                s, p = shapiro(indices)
            else:
                s, p = shapiro(data2[col])
            print('{} {} {}'.format(col, s, p))
            if p > alpha:
                print('Aceptablemente normal con nivel de significancia {}\n'.format(alpha))
            else:
                print('No parece ser normal con nivel de significancia {}\n'.format(alpha))
```

```
m 0.2348303347826004 6.350159261984893e-23
No parece ser normal con nivel de significancia 0.05

f1 0.6251736879348755 1.379031445065371e-16
No parece ser normal con nivel de significancia 0.05

f2 0.7796123027801514 1.379099633970482e-12
No parece ser normal con nivel de significancia 0.05

f3 0.8172591328620911 2.542906336888695e-11
No parece ser normal con nivel de significancia 0.05

f4 0.9010974168777466 1.06176564429461e-07
No parece ser normal con nivel de significancia 0.05

ac1 0.9681503176689148 0.0041356817819178104
No parece ser normal con nivel de significancia 0.05

ac2 0.9490779638290405 0.00010960957297356799
No parece ser normal con nivel de significancia 0.05

ac4 0.944659948348999 5.162579327588901e-05
No parece ser normal con nivel de significancia 0.05
```

```
ac8 0.9543476700782776 0.0002801059454213828
No parece ser normal con nivel de significancia 0.05

ac16 0.8447107076644897 2.779554086895786e-10
No parece ser normal con nivel de significancia 0.05

ac32 0.8849678635597229 1.6211028963653007e-08
No parece ser normal con nivel de significancia 0.05

cie 0.8060136437416077 1.0233016690952557e-11
No parece ser normal con nivel de significancia 0.05

m 0.2348303347826004 6.350159261984893e-23
No parece ser normal con nivel de significancia 0.01

f1 0.6251736879348755 1.379031445065371e-16
No parece ser normal con nivel de significancia 0.01

f2 0.7796123027801514 1.379099633970482e-12
No parece ser normal con nivel de significancia 0.01

f3 0.8172591328620911 2.542906336888695e-11
No parece ser normal con nivel de significancia 0.01

f4 0.9010974168777466 1.06176564429461e-07
No parece ser normal con nivel de significancia 0.01

ac1 0.9681503176689148 0.0041356817819178104
No parece ser normal con nivel de significancia 0.01

ac2 0.9490779638290405 0.00010960957297356799
No parece ser normal con nivel de significancia 0.01

ac4 0.9446599483489999 5.162579327588901e-05
No parece ser normal con nivel de significancia 0.01

ac8 0.9543476700782776 0.0002801059454213828
No parece ser normal con nivel de significancia 0.01

ac16 0.8447107076644897 2.779554086895786e-10
No parece ser normal con nivel de significancia 0.01

ac32 0.8849678635597229 1.6211028963653007e-08
No parece ser normal con nivel de significancia 0.01

cie 0.8060136437416077 1.0233016690952557e-11
No parece ser normal con nivel de significancia 0.01
```

Como ninguno de los parámetros de los datos de entrada del algoritmo de k-medias parece normal, se utilizarán pruebas no paramétricas

Pruebas no paramétricas

Para el primer grupo de datos se desea saber si los datos agrupados por presencias predominantes (año 2014 y CIEs **A** y **B**) tienen la misma distribución que el resto de los datos. Para ello se utiliza la prueba de Kruskal-Wallis. Primero la diferencia de distribución respecto al año 2014 con el resto de los años

In [179]:
```
from scipy.stats import kruskal

anio2014 = data1[data1['anio'] == 2014]['casos']
anioNo2014 = data1[data1['anio'] != 2014]['casos']

for alpha in [0.05, 0.01]:
    s, p = kruskal(anio2014, anioNo2014)
    if p > alpha:
        print('el factor no parece causar diferencia con nivel de significancia', alpha)
    else:
        print('hay por lo menos un factor que causa diferencia con nivel de significancia', alpha)
```

```
hay por lo menos un factor que causa diferencia con nivel de significancia 0.05
hay por lo menos un factor que causa diferencia con nivel de significancia 0.01
```

y ahora entre las CIEs **A** y **B** con el resto

In [192]:
```
cieAB = data1[data1['cie'] == 'a']['casos']
cieAB = cieAB.append(data1[data1['cie'] == 'b']['casos'])
cieNoAB = data1[data1['cie'] != 'a']
cieNoAB = cieNoAB[cieNoAB['cie'] != 'b']['casos']

for alpha in [0.05, 0.01]:
    s, p = kruskal(cieAB, cieNoAB)
    if p > alpha:
        print('el factor no parece causar diferencia con nivel de significancia', alpha)
    else:
        print('hay por lo menos un factor que causa diferencia con nivel de significancia', alpha)
```

```
hay por lo menos un factor que causa diferencia con nivel de significancia 0.05
hay por lo menos un factor que causa diferencia con nivel de significancia 0.01
```

In [ ]:
```
de donde se ve que en ambos casos hay diferencias en las distribuciones de los conjuntos de datos.
```

In [172]:
```
from scipy.stats import mannwhitneyu
nCols = data2.shape[1]
for alpha in [0.05, 0.01]:
    for i in range(nCols - 1): # Todos entre sí
```

```
        if data2.columns[i] != 'enf' and data2.columns[i] != 'cie' and data2.columns[i] != 'cluster':
            for j in range(i + 1, nCols - 2):
                if data2.columns[j] != 'enf' and data2.columns[j] != 'cie' and data2.columns[j] != 'cluster':
                    s, p = mannwhitneyu(data2[data2.columns[i]], data2[data2.columns[j]])
                    if p > alpha:
                        print('{} y {}'.format(data2.columns[i], data2.columns[j]))
                        print('son igualmente distribuidos con nivel de significancia', alpha)
                    #else:
                    #    print('se ven diferentemente distribuidos con nivel de significancia', alpha)
```

```
f4 y ac16
son igualmente distribuidos con nivel de significancia 0.05
f4 y ac32
son igualmente distribuidos con nivel de significancia 0.05
ac2 y ac4
son igualmente distribuidos con nivel de significancia 0.05
ac2 y ac8
son igualmente distribuidos con nivel de significancia 0.05
ac4 y ac8
son igualmente distribuidos con nivel de significancia 0.05
f2 y f3
son igualmente distribuidos con nivel de significancia 0.01
f3 y f4
son igualmente distribuidos con nivel de significancia 0.01
f4 y ac4
son igualmente distribuidos con nivel de significancia 0.01
f4 y ac16
son igualmente distribuidos con nivel de significancia 0.01
f4 y ac32
son igualmente distribuidos con nivel de significancia 0.01
ac2 y ac4
son igualmente distribuidos con nivel de significancia 0.01
ac2 y ac8
son igualmente distribuidos con nivel de significancia 0.01
ac4 y ac8
son igualmente distribuidos con nivel de significancia 0.01
ac4 y ac32
son igualmente distribuidos con nivel de significancia 0.01
```

Enseguida se comprueba si las CIEs asignadas por la OMS tienen la misma distribución que los agrupamientos definidos por el algoritmo de k-medias

In [175]:
```
from scipy.stats import wilcoxon
classnames, indices = np.unique(data2['cie'], return_inverse=True)
for alpha in [0.05, 0.01]:
    s, p = wilcoxon(indices, data2['cluster'])
    print('{:.2f} {:.3f}'.format( s, p))
    if p > alpha:
        print('son igualmente distribuidos con nivel de significancia', alpha)
    else:
        print('se ven diferentemente distribuidos con nivel de significancia', alpha)
```

```
2749.50 0.076
son igualmente distribuidos con nivel de significancia 0.05
2749.50 0.076
son igualmente distribuidos con nivel de significancia 0.01
```