Momento de Retroalimentación: Módulo 1 Construcción de un modelo estadístico base (Portafolio Implementación)

Descripción:

La contaminación por mercurio de peces en el agua dulce comestibles es una amenaza directa contra nuestra salud. Se llevó a cabo un estudio reciente en 53 lagos de Florida con el fin de examinar los factores que influían en el nivel de contaminación por mercurio. Las variables que se midieron se encuentran en mercurio.csv Descargar mercurio.csv y su descripción es la siguiente:

Alrededor de la principal pregunta de investigación que surge en este estudio: ¿Cuáles son los principales factores que influyen en el nivel de contaminación por mercurio en los peces de los lagos de Florida? pueden surgir preguntas paralelas que desglosan esta pregunta general:

1. ¿Hay evidencia para suponer que la concentración promedio de mercurio en los lagos es dañino para la salud humana? Considera que las normativas de referencia para evaluar los niveles máximos de Hg (Reglamento 34687-MAG y los reglamentos internacionales CE 1881/2006 y Codex Standard 193-1995) establecen que la concentración promedio de mercurio en productos de la pesca no debe superar los 0.5 mg de Hg/kg.
2. ¿Habrá diferencia significativa entre la concentración de mercurio por la edad de los peces?
3. Si el muestreo se realizó lanzando una red y analizando los peces que la red encontraba ¿Habrá influencia del número de peces encontrados en la concentración de mercurio en los peces?
4. ¿Las concentraciones de alcalinidad, clorofila, calcio en el agua del lago influyen en la concentración de mercurio de los peces?

Preparación de los datos

In [72]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
# from scipy.stats import norm
import statsmodels.api as sm
#from scipy import stats
# plt.style.use('ggplot')
from statsmodels.graphics.factorplots import interaction_plot
import pingouin as pg
import scipy.stats as stats
import statsmodels.formula.api as sfm
```

In [2]:

```
df = pd.read_csv("mercurio.csv")
```

In [3]:
```
df
```

Out[3]:

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
0	1	Alligator	5.9	6.1	3.0	0.7	1.23	5	0.85	1.43	1.53	1
1	2	Annie	3.5	5.1	1.9	3.2	1.33	7	0.92	1.90	1.33	0
2	3	Apopka	116.0	9.1	44.1	128.3	0.04	6	0.04	0.06	0.04	0
3	4	Blue Cypress	39.4	6.9	16.4	3.5	0.44	12	0.13	0.84	0.44	0
4	5	Brick	2.5	4.6	2.9	1.8	1.20	12	0.69	1.50	1.33	1
5	6	Bryant	19.6	7.3	4.5	44.1	0.27	14	0.04	0.48	0.25	1
6	7	Cherry	5.2	5.4	2.8	3.4	0.48	10	0.30	0.72	0.45	1
7	8	Crescent	71.4	8.1	55.2	33.7	0.19	12	0.08	0.38	0.16	1
8	9	Deer Point	26.4	5.8	9.2	1.6	0.83	24	0.26	1.40	0.72	1
9	10	Dias	4.8	6.4	4.6	22.5	0.81	12	0.41	1.47	0.81	1
10	11	Dorr	6.6	5.4	2.7	14.9	0.71	12	0.52	0.86	0.71	1
11	12	Down	16.5	7.2	13.8	4.0	0.50	12	0.10	0.73	0.51	1
12	13	Eaton	25.4	7.2	25.2	11.6	0.49	7	0.26	1.01	0.54	1
13	14	East Tohopekaliga	7.1	5.8	5.2	5.8	1.16	43	0.50	2.03	1.00	1
14	15	Farm-13	128.0	7.6	86.5	71.1	0.05	11	0.04	0.11	0.05	0
15	16	George	83.7	8.2	66.5	78.6	0.15	10	0.12	0.18	0.15	1
16	17	Griffin	108.5	8.7	35.6	80.1	0.19	40	0.07	0.43	0.19	1
17	18	Harney	61.3	7.8	57.4	13.9	0.77	6	0.32	1.50	0.49	1
18	19	Hart	6.4	5.8	4.0	4.6	1.08	10	0.64	1.33	1.02	1
19	20	Hatchineha	31.0	6.7	15.0	17.0	0.98	6	0.67	1.44	0.70	1
20	21	Iamonia	7.5	4.4	2.0	9.6	0.63	12	0.33	0.93	0.45	1
21	22	Istokpoga	17.3	6.7	10.7	9.5	0.56	12	0.37	0.94	0.59	1
22	23	Jackson	12.6	6.1	3.7	21.0	0.41	12	0.25	0.61	0.41	0
23	24	Josephine	7.0	6.9	6.3	32.1	0.73	12	0.33	2.04	0.81	1
24	25	Kingsley	10.5	5.5	6.3	1.6	0.34	10	0.25	0.62	0.42	1
25	26	Kissimmee	30.0	6.9	13.9	21.5	0.59	36	0.23	1.12	0.53	1
26	27	Lochloosa	55.4	7.3	15.9	24.7	0.34	10	0.17	0.52	0.31	1
27	28	Louisa	3.9	4.5	3.3	7.0	0.84	8	0.59	1.38	0.87	1
28	29	Miccasukee	5.5	4.8	1.7	14.8	0.50	11	0.31	0.84	0.50	0
29	30	Minneola	6.3	5.8	3.3	0.7	0.34	10	0.19	0.69	0.47	1
30	31	Monroe	67.0	7.8	58.6	43.8	0.28	10	0.16	0.59	0.25	1
31	32	Newmans	28.8	7.4	10.2	32.7	0.34	10	0.16	0.65	0.41	1
32	33	Ocean Pond	5.8	3.6	1.6	3.2	0.87	12	0.31	1.90	0.87	0
33	34	Ocheese Pond	4.5	4.4	1.1	3.2	0.56	13	0.25	1.02	0.56	0

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
34	35	Okeechobee	119.1	7.9	38.4	16.1	0.17	12	0.07	0.30	0.16	1
35	36	Orange	25.4	7.1	8.8	45.2	0.18	13	0.09	0.29	0.16	1
36	37	Panasoffkee	106.5	6.8	90.7	16.5	0.19	13	0.05	0.37	0.23	1
37	38	Parker	53.0	8.4	45.6	152.4	0.04	4	0.04	0.06	0.04	0
38	39	Placid	8.5	7.0	2.5	12.8	0.49	12	0.31	0.63	0.56	1
39	40	Puzzle	87.6	7.5	85.5	20.1	1.10	10	0.79	1.41	0.89	1
40	41	Rodman	114.0	7.0	72.6	6.4	0.16	14	0.04	0.26	0.18	1
41	42	Rousseau	97.5	6.8	45.5	6.2	0.10	12	0.05	0.26	0.19	1
42	43	Sampson	11.8	5.9	24.2	1.6	0.48	10	0.27	1.05	0.44	1
43	44	Shipp	66.5	8.3	26.0	68.2	0.21	12	0.05	0.48	0.16	1
44	45	Talquin	16.0	6.7	41.2	24.1	0.86	12	0.36	1.40	0.67	1
45	46	Tarpon	5.0	6.2	23.6	9.6	0.52	12	0.31	0.95	0.55	1
46	51	Tohopekaliga	25.6	6.2	12.6	27.7	0.65	44	0.30	1.10	0.58	1
47	47	Trafford	81.5	8.9	20.5	9.6	0.27	6	0.04	0.40	0.27	0
48	48	Trout	1.2	4.3	2.1	6.4	0.94	10	0.59	1.24	0.98	1
49	49	Tsala Apopka	34.0	7.0	13.1	4.6	0.40	12	0.08	0.90	0.31	1
50	50	Weir	15.5	6.9	5.2	16.5	0.43	11	0.23	0.69	0.43	1
51	52	Wildcat	17.3	5.2	3.0	2.6	0.25	12	0.15	0.40	0.28	1
52	53	Yale	71.8	7.9	20.5	8.8	0.27	12	0.15	0.51	0.25	1

```
In [4]: columns = np.array(["id", "nombre", "alcalinidad", "PH", "calcio", "clorofila", '
```

```
In [5]: df.columns = columns
```

```
In [6]: df = df.set_index('id')
```

In [7]:
```
df
```

Out[7]:

	nombre	alcalinidad	PH	calcio	clorofila	mercurio	npeces	minmercurio	maxmercurio	e
id										
1	Alligator	5.9	6.1	3.0	0.7	1.23	5	0.85	1.43	
2	Annie	3.5	5.1	1.9	3.2	1.33	7	0.92	1.90	
3	Apopka	116.0	9.1	44.1	128.3	0.04	6	0.04	0.06	
4	Blue Cypress	39.4	6.9	16.4	3.5	0.44	12	0.13	0.84	
5	Brick	2.5	4.6	2.9	1.8	1.20	12	0.69	1.50	
6	Bryant	19.6	7.3	4.5	44.1	0.27	14	0.04	0.48	
7	Cherry	5.2	5.4	2.8	3.4	0.48	10	0.30	0.72	
8	Crescent	71.4	8.1	55.2	33.7	0.19	12	0.08	0.38	
9	Deer Point	26.4	5.8	9.2	1.6	0.83	24	0.26	1.40	
10	Dias	4.8	6.4	4.6	22.5	0.81	12	0.41	1.47	
11	Dorr	6.6	5.4	2.7	14.9	0.71	12	0.52	0.86	
12	Down	16.5	7.2	13.8	4.0	0.50	12	0.10	0.73	
13	Eaton	25.4	7.2	25.2	11.6	0.49	7	0.26	1.01	
14	East Tohopekaliga	7.1	5.8	5.2	5.8	1.16	43	0.50	2.03	
15	Farm-13	128.0	7.6	86.5	71.1	0.05	11	0.04	0.11	
16	George	83.7	8.2	66.5	78.6	0.15	10	0.12	0.18	
17	Griffin	108.5	8.7	35.6	80.1	0.19	40	0.07	0.43	
18	Harney	61.3	7.8	57.4	13.9	0.77	6	0.32	1.50	
19	Hart	6.4	5.8	4.0	4.6	1.08	10	0.64	1.33	
20	Hatchineha	31.0	6.7	15.0	17.0	0.98	6	0.67	1.44	
21	Iamonia	7.5	4.4	2.0	9.6	0.63	12	0.33	0.93	
22	Istokpoga	17.3	6.7	10.7	9.5	0.56	12	0.37	0.94	
23	Jackson	12.6	6.1	3.7	21.0	0.41	12	0.25	0.61	
24	Josephine	7.0	6.9	6.3	32.1	0.73	12	0.33	2.04	
25	Kingsley	10.5	5.5	6.3	1.6	0.34	10	0.25	0.62	
26	Kissimmee	30.0	6.9	13.9	21.5	0.59	36	0.23	1.12	
27	Lochloosa	55.4	7.3	15.9	24.7	0.34	10	0.17	0.52	
28	Louisa	3.9	4.5	3.3	7.0	0.84	8	0.59	1.38	
29	Miccasukee	5.5	4.8	1.7	14.8	0.50	11	0.31	0.84	
30	Minneola	6.3	5.8	3.3	0.7	0.34	10	0.19	0.69	
31	Monroe	67.0	7.8	58.6	43.8	0.28	10	0.16	0.59	
32	Newmans	28.8	7.4	10.2	32.7	0.34	10	0.16	0.65	

	nombre	alcalinidad	PH	calcio	clorofila	mercurio	npeces	minmercurio	maxmercurio
id									
33	Ocean Pond	5.8	3.6	1.6	3.2	0.87	12	0.31	1.90
34	Ocheese Pond	4.5	4.4	1.1	3.2	0.56	13	0.25	1.02
35	Okeechobee	119.1	7.9	38.4	16.1	0.17	12	0.07	0.30
36	Orange	25.4	7.1	8.8	45.2	0.18	13	0.09	0.29
37	Panasoffkee	106.5	6.8	90.7	16.5	0.19	13	0.05	0.37
38	Parker	53.0	8.4	45.6	152.4	0.04	4	0.04	0.06
39	Placid	8.5	7.0	2.5	12.8	0.49	12	0.31	0.63
40	Puzzle	87.6	7.5	85.5	20.1	1.10	10	0.79	1.41
41	Rodman	114.0	7.0	72.6	6.4	0.16	14	0.04	0.26
42	Rousseau	97.5	6.8	45.5	6.2	0.10	12	0.05	0.26
43	Sampson	11.8	5.9	24.2	1.6	0.48	10	0.27	1.05
44	Shipp	66.5	8.3	26.0	68.2	0.21	12	0.05	0.48
45	Talquin	16.0	6.7	41.2	24.1	0.86	12	0.36	1.40
46	Tarpon	5.0	6.2	23.6	9.6	0.52	12	0.31	0.95
51	Tohopekaliga	25.6	6.2	12.6	27.7	0.65	44	0.30	1.10
47	Trafford	81.5	8.9	20.5	9.6	0.27	6	0.04	0.40
48	Trout	1.2	4.3	2.1	6.4	0.94	10	0.59	1.24
49	Tsala Apopka	34.0	7.0	13.1	4.6	0.40	12	0.08	0.90
50	Weir	15.5	6.9	5.2	16.5	0.43	11	0.23	0.69
52	Wildcat	17.3	5.2	3.0	2.6	0.25	12	0.15	0.40
53	Yale	71.8	7.9	20.5	8.8	0.27	12	0.15	0.51

Exploración de los Datos

In [8]:

```
df.describe()
```

Out[8]:

	alcalinidad	PH	calcio	clorofila	mercurio	npeces	minmercurio	maxmercı
count	53.000000	53.000000	53.000000	53.000000	53.000000	53.000000	53.000000	53.000
mean	37.530189	6.590566	22.201887	23.116981	0.527170	13.056604	0.279811	0.874
std	38.203527	1.288449	24.932574	30.816321	0.341036	8.560677	0.226406	0.522
min	1.200000	3.600000	1.100000	0.700000	0.040000	4.000000	0.040000	0.060
25%	6.600000	5.800000	3.300000	4.600000	0.270000	10.000000	0.090000	0.480
50%	19.600000	6.800000	12.600000	12.800000	0.480000	12.000000	0.250000	0.840
75%	66.500000	7.400000	35.600000	24.700000	0.770000	12.000000	0.330000	1.330
max	128.000000	9.100000	90.700000	152.400000	1.330000	44.000000	0.920000	2.040

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 53 entries, 1 to 53
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   nombre       53 non-null     object 
 1   alcalinidad  53 non-null     float64
 2   PH           53 non-null     float64
 3   calcio       53 non-null     float64
 4   clorofila    53 non-null     float64
 5   mercurio     53 non-null     float64
 6   npeces       53 non-null     int64  
 7   minmercurio  53 non-null     float64
 8   maxmercurio  53 non-null     float64
 9   estimacion   53 non-null     float64
 10  edad         53 non-null     int64  
dtypes: float64(8), int64(2), object(1)
memory usage: 5.0+ KB
```

In [10]:

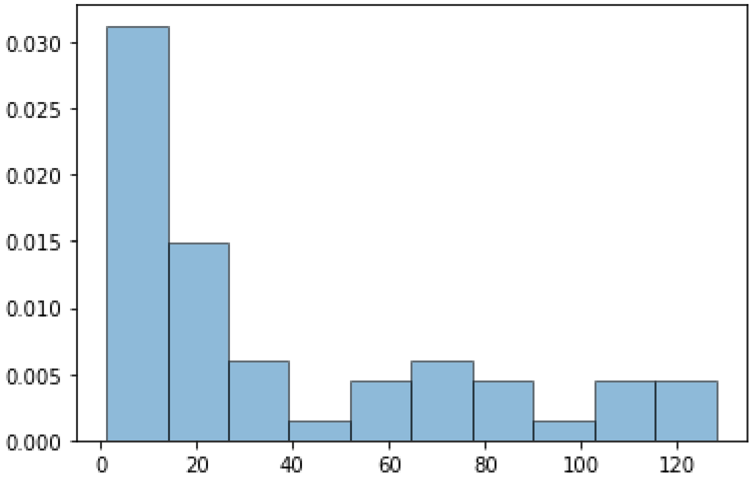
```
"""sigma = df.alcalinidad.std()
mu = df.alcalinidad.mean()
domain = np.linspace(df.alcalinidad.min(), df.alcalinidad.max())"""
```

Out[10]:

```
'sigma = df.alcalinidad.std()\nmu = df.alcalinidad.mean()\ndomain = np.linspace
(df.alcalinidad.min(), df.alcalinidad.max())'
```

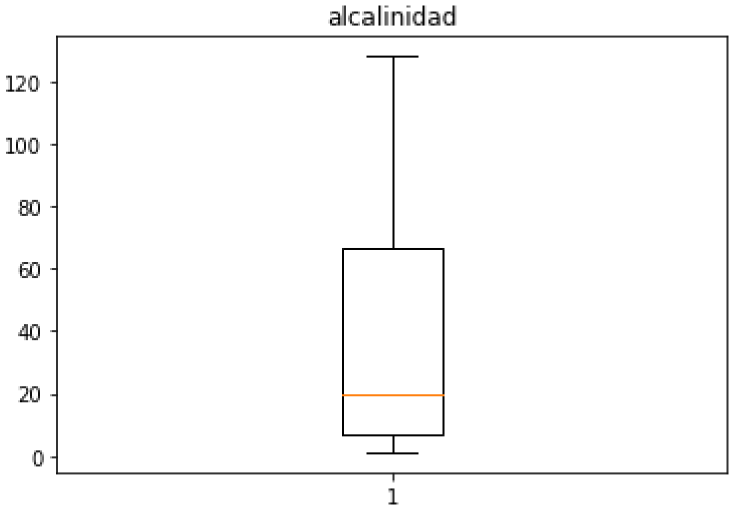
In [11]:

```
# plt.plot(domain, norm.pdf(domain, mu, sigma))
plt.hist(df.alcalinidad, edgecolor = "black", alpha = 0.5, density = True)
plt.show()
```



In [12]:

```
plt.title('alcalinidad')
plt.boxplot(df.alcalinidad)
plt.show()
```

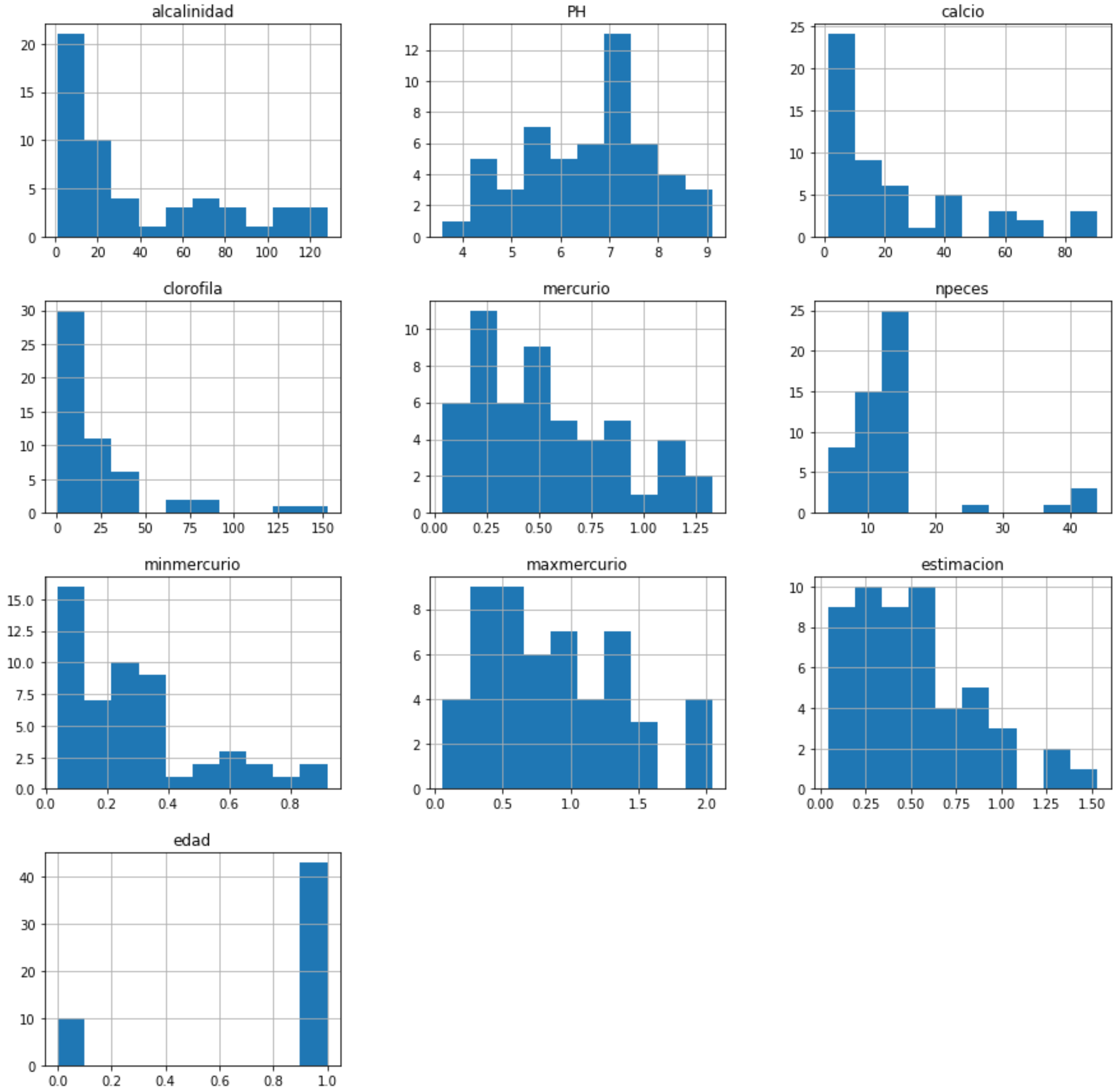


In [13]:

```
df.hist(figsize = (15,15))
```

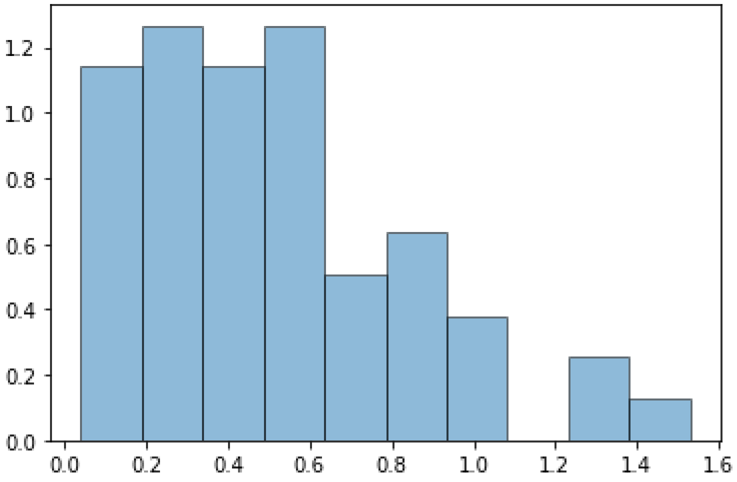
Out[13]:

```
array([[<AxesSubplot:title={'center':'alcalinidad'}>,
        <AxesSubplot:title={'center':'PH'}>,
        <AxesSubplot:title={'center':'calcio'}>],
       [<AxesSubplot:title={'center':'clorofila'}>,
        <AxesSubplot:title={'center':'mercurio'}>,
        <AxesSubplot:title={'center':'npeces'}>],
       [<AxesSubplot:title={'center':'minmercurio'}>,
        <AxesSubplot:title={'center':'maxmercurio'}>,
        <AxesSubplot:title={'center':'estimacion'}>],
       [<AxesSubplot:title={'center':'edad'}>, <AxesSubplot:>,
        <AxesSubplot:>]], dtype=object)
```



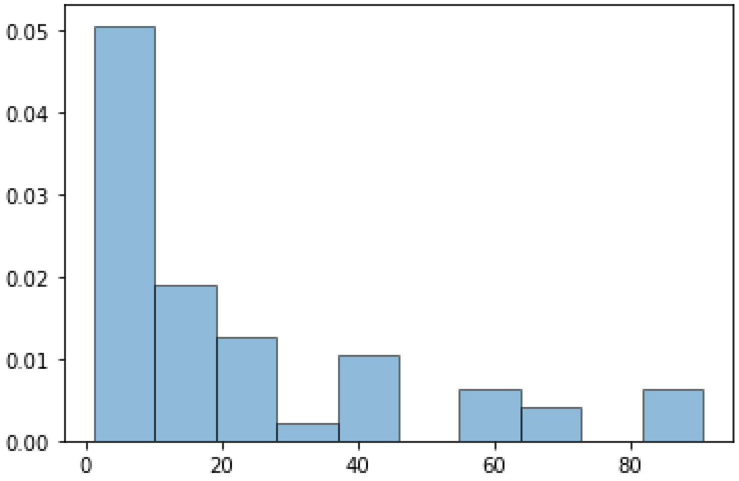
In [14]:

```
plt.hist(df.estimacion, edgecolor = "black", alpha = 0.5, density = True)
plt.show()
```



In [15]:

```
plt.hist(df.calcio, edgecolor = "black", alpha = 0.5, density = True)
plt.show()
```



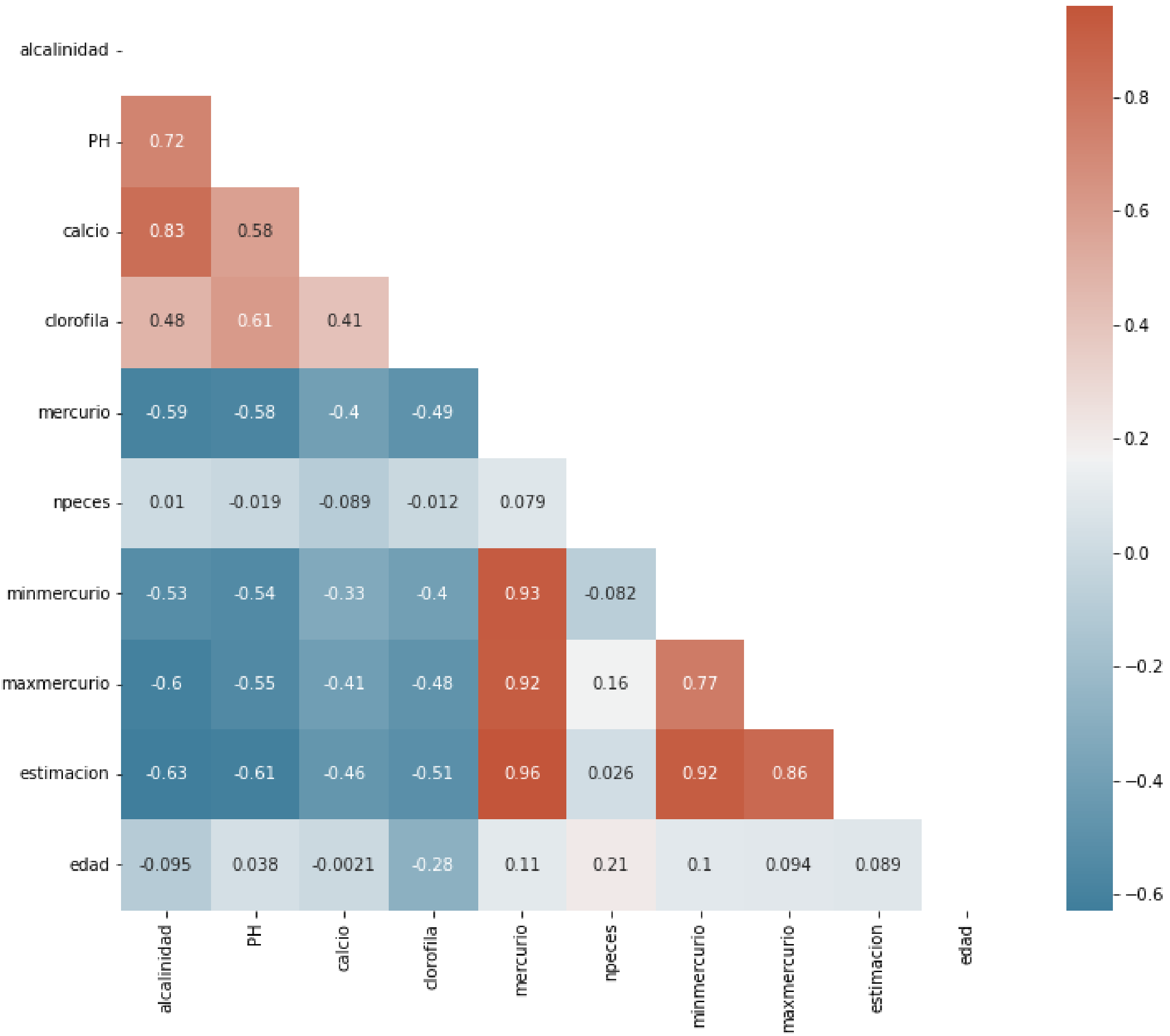
In [16]:
```
df.corr()
```

Out[16]:

	alcalinidad	PH	calcio	clorofila	mercurio	npeces	minmercurio	maxn
alcalinidad	1.000000	0.719166	0.832604	0.477531	-0.593897	0.010291	-0.525357	-0
PH	0.719166	1.000000	0.577133	0.608483	-0.575400	-0.018606	-0.541965	-0
calcio	0.832604	0.577133	1.000000	0.409914	-0.400680	-0.089379	-0.332476	-0
clorofila	0.477531	0.608483	0.409914	1.000000	-0.491375	-0.011820	-0.400459	-0
mercurio	-0.593897	-0.575400	-0.400680	-0.491375	1.000000	0.079034	0.927205	0
npeces	0.010291	-0.018606	-0.089379	-0.011820	0.079034	1.000000	-0.081653	0
minmercurio	-0.525357	-0.541965	-0.332476	-0.400459	0.927205	-0.081653	1.000000	0
maxmercurio	-0.604796	-0.551815	-0.407917	-0.484972	0.915864	0.161092	0.765353	1
estimacion	-0.627958	-0.612849	-0.464409	-0.506442	0.959215	0.025800	0.919089	0
edad	-0.094939	0.038000	-0.002111	-0.283002	0.108739	0.207956	0.100662	0

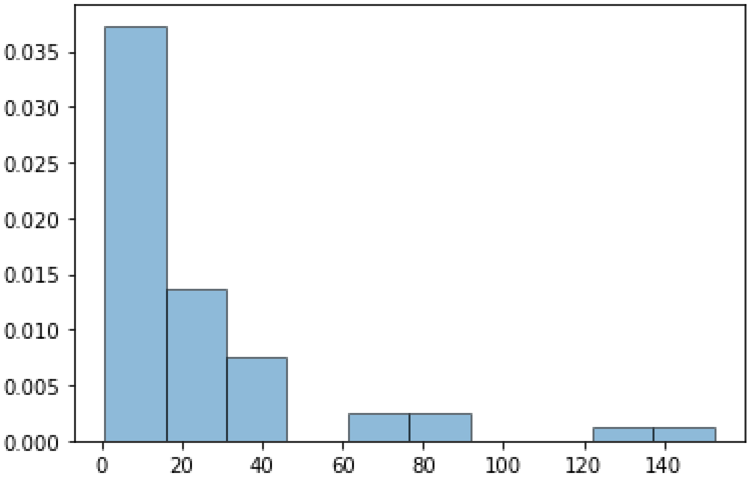
In [17]:

```
# clf = clf.select_dtypes(include=['float64','int'])
corr = df.corr(method = 'pearson')
f, ax = plt.subplots(figsize=(12, 10))
mask = np.triu(np.ones_like(corr, dtype=bool))
cmap = sns.diverging_palette(230, 20, as_cmap=True)
sns.heatmap(corr, annot=True, mask = mask, cmap=cmap)
plt.show()
```



In [18]:

```
plt.hist(df.clorofila, edgecolor = "black", alpha = 0.5, density = True)
plt.show()
```

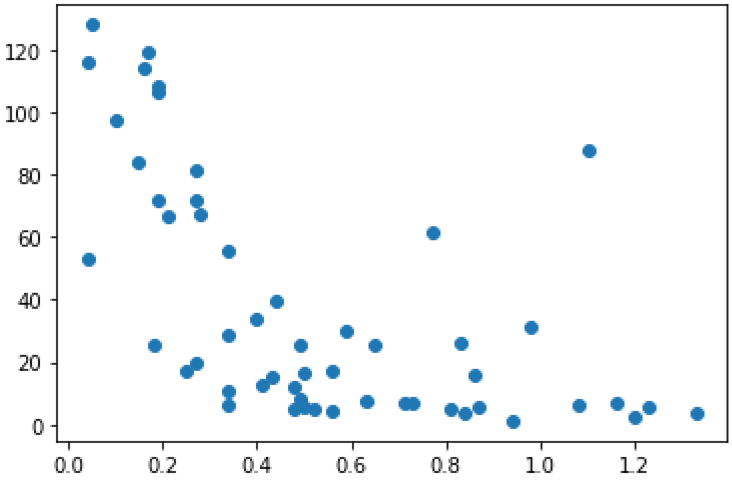


In [19]:

```
x = df['mercurio']
y = df['alcalinidad']
fig,ax = plt.subplots()
ax.scatter(x,y)
```

Out[19]:

```
<matplotlib.collections.PathCollection at 0x28fd330eeb0>
```

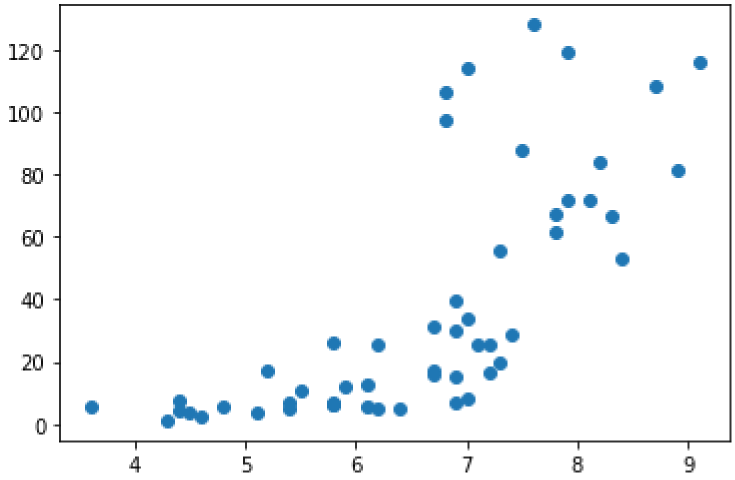


In [20]:

```
x = df['PH']
y = df['alcalinidad']
fig,ax = plt.subplots()
ax.scatter(x,y)
```

Out[20]:

```
<matplotlib.collections.PathCollection at 0x28fd392f700>
```

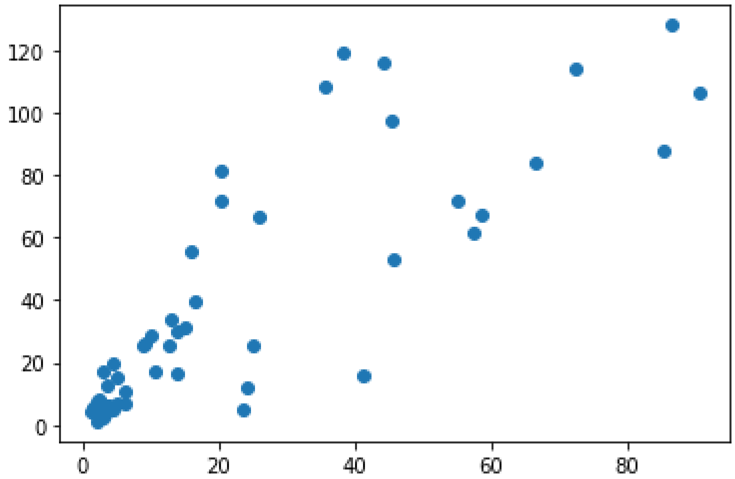


In [21]:

```
x = df['calcio']
y = df['alcalinidad']
fig,ax = plt.subplots()
ax.scatter(x,y)
```

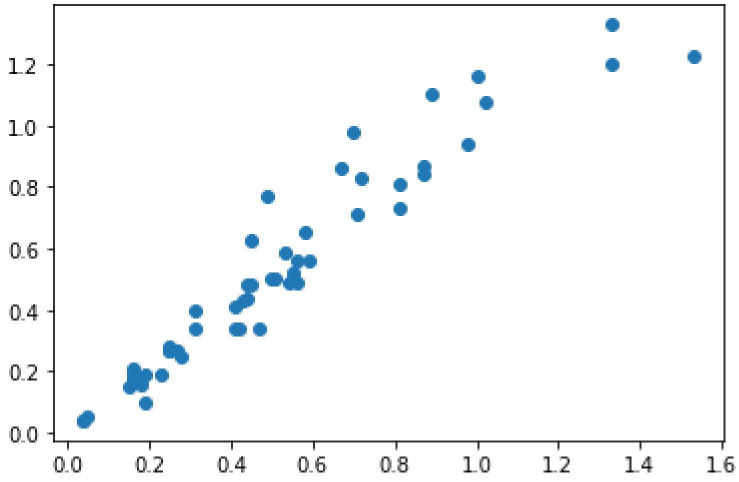
Out[21]:

```
<matplotlib.collections.PathCollection at 0x28fd3a85040>
```

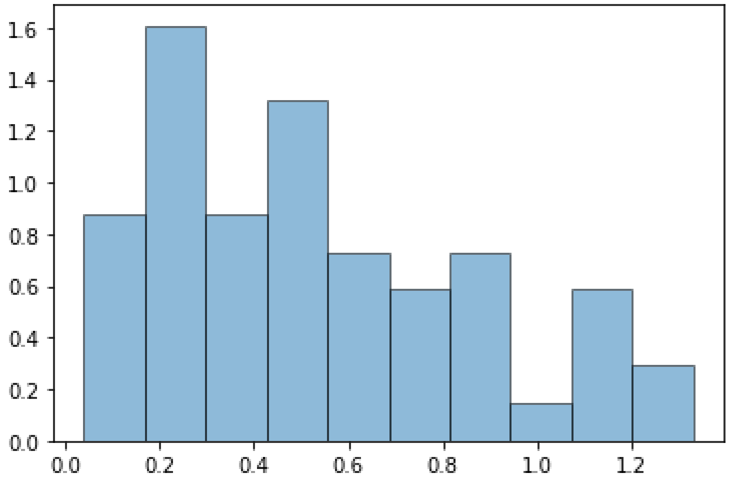


In [22]:
```
x = df['estimacion']
y = df['mercurio']
fig,ax = plt.subplots()
ax.scatter(x,y)
```

Out[22]:
```
<matplotlib.collections.PathCollection at 0x28fd3ae8370>
```



In [23]:
```
plt.hist(df.mercurio, edgecolor = "black", alpha = 0.5, density = True)
plt.show()
```



Moda

In [24]:
```
from scipy import stats

stats.mode(df)
```

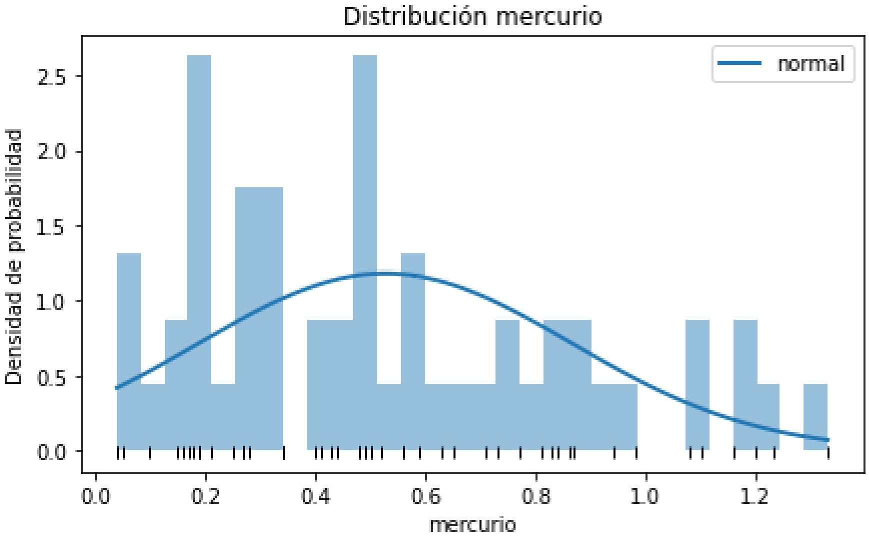
Out[24]:
```
ModeResult(mode=array([['Alligator', 17.3, 5.8, 3.0, 1.6, 0.34, 12, 0.04, 0.06,
0.16, 1]],
      dtype=object), count=array([[ 1,  2,  4,  2,  3,  4, 20,  6,  2,  4, 4
3]]))
```

Distribuciones

In [25]:
```
#datos = datos[(datos.age > 15) & (datos.male ==0)]
mercurio = df['mercurio']
alcalinidad = df['alcalinidad']
PH = df['PH']
calcio = df['calcio']
npeces = df['npeces']
```

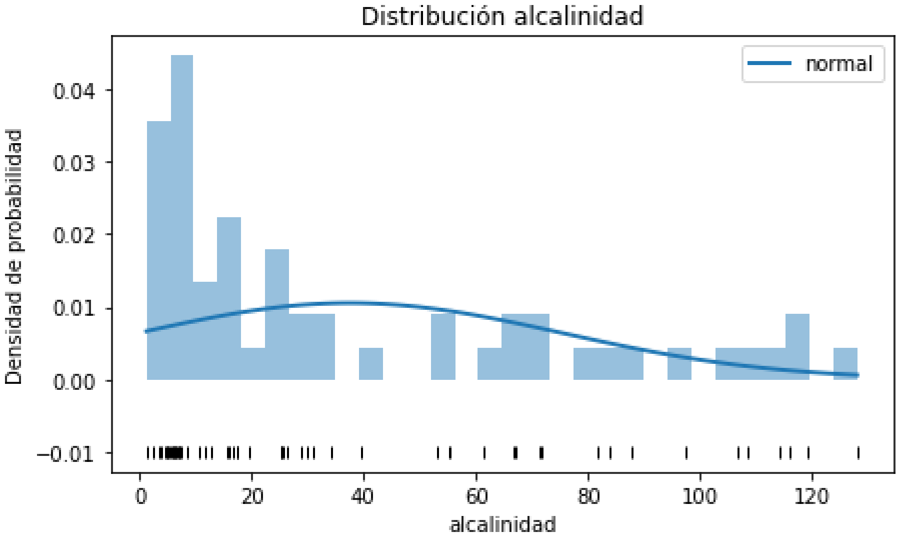
In [26]:

```
# Histograma + curva normal teórica
# ==============================================================================

# Valores de la media (mu) y desviación típica (sigma) de los datos
mu, sigma = stats.norm.fit(mercurio)

# Valores teóricos de la normal en el rango observado
x_hat = np.linspace(min(mercurio), max(mercurio), num=100)
y_hat = stats.norm.pdf(x_hat, mu, sigma)

# Gráfico
fig, ax = plt.subplots(figsize=(7,4))
ax.plot(x_hat, y_hat, linewidth=2, label='normal')
ax.hist(x=mercurio, density=True, bins=30, color="#3182bd", alpha=0.5)
ax.plot(mercurio, np.full_like(mercurio, -0.01), '|k', markeredgewidth=1)
ax.set_title('Distribución mercurio')
ax.set_xlabel('mercurio')
ax.set_ylabel('Densidad de probabilidad')
ax.legend();
```



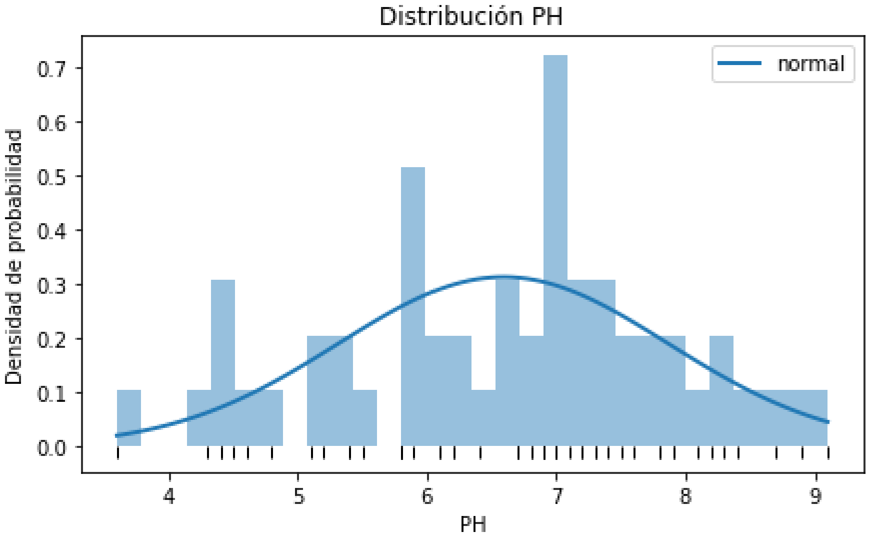
In [27]:

```
# Histograma + curva normal teórica
# ==============================================================================

# Valores de la media (mu) y desviación típica (sigma) de los datos
mu, sigma = stats.norm.fit(alcalinidad)

# Valores teóricos de la normal en el rango observado
x_hat = np.linspace(min(alcalinidad), max(alcalinidad), num=100)
y_hat = stats.norm.pdf(x_hat, mu, sigma)

# Gráfico
fig, ax = plt.subplots(figsize=(7,4))
ax.plot(x_hat, y_hat, linewidth=2, label='normal')
ax.hist(x=alcalinidad, density=True, bins=30, color="#3182bd", alpha=0.5)
ax.plot(alcalinidad, np.full_like(alcalinidad, -0.01), '|k', markeredgewidth=1)
ax.set_title('Distribución alcalinidad')
ax.set_xlabel('alcalinidad')
ax.set_ylabel('Densidad de probabilidad')
ax.legend();
```



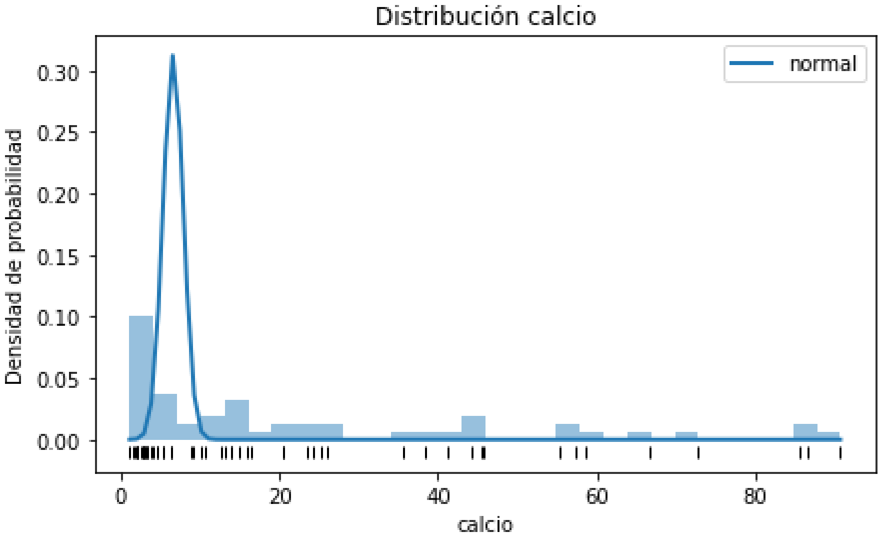
In [28]:

```
# Histograma + curva normal teórica
# ==============================================================================

# Valores de la media (mu) y desviación típica (sigma) de los datos
mu, sigma = stats.norm.fit(PH)

# Valores teóricos de la normal en el rango observado
x_hat = np.linspace(min(PH), max(PH), num=100)
y_hat = stats.norm.pdf(x_hat, mu, sigma)

# Gráfico
fig, ax = plt.subplots(figsize=(7,4))
ax.plot(x_hat, y_hat, linewidth=2, label='normal')
ax.hist(x=PH, density=True, bins=30, color="#3182bd", alpha=0.5)
ax.plot(PH, np.full_like(PH, -0.01), '|k', markeredgewidth=1)
ax.set_title('Distribución PH')
ax.set_xlabel('PH')
ax.set_ylabel('Densidad de probabilidad')
ax.legend();
```



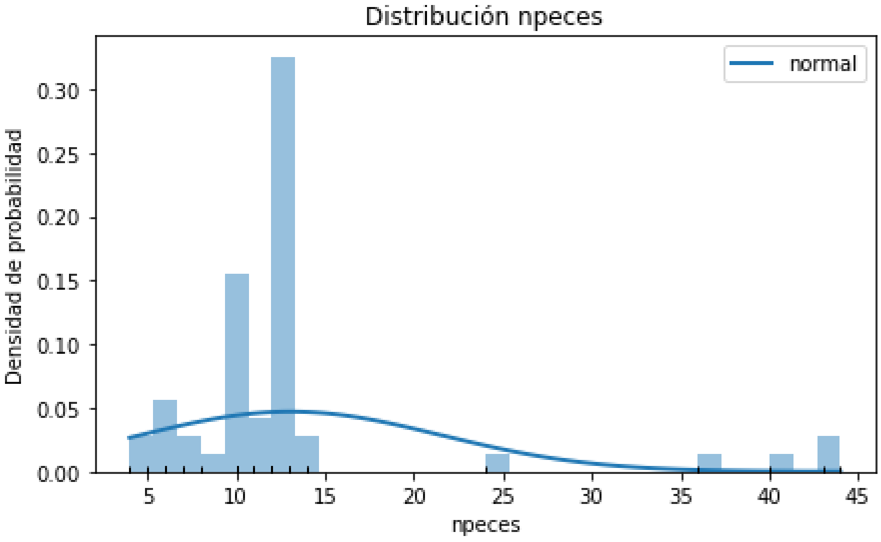
In [29]:

```
# Histograma + curva normal teórica
# ==============================================================================

# Valores de la media (mu) y desviación típica (sigma) de los datos
mu, sigma = stats.norm.fit(PH)

# Valores teóricos de la normal en el rango observado
x_hat = np.linspace(min(calcio), max(calcio), num=100)
y_hat = stats.norm.pdf(x_hat, mu, sigma)

# Gráfico
fig, ax = plt.subplots(figsize=(7,4))
ax.plot(x_hat, y_hat, linewidth=2, label='normal')
ax.hist(x=calcio, density=True, bins=30, color="#3182bd", alpha=0.5)
ax.plot(calcio, np.full_like(calcio, -0.01), '|k', markeredgewidth=1)
ax.set_title('Distribución calcio')
ax.set_xlabel('calcio')
ax.set_ylabel('Densidad de probabilidad')
ax.legend();
```



In [30]:

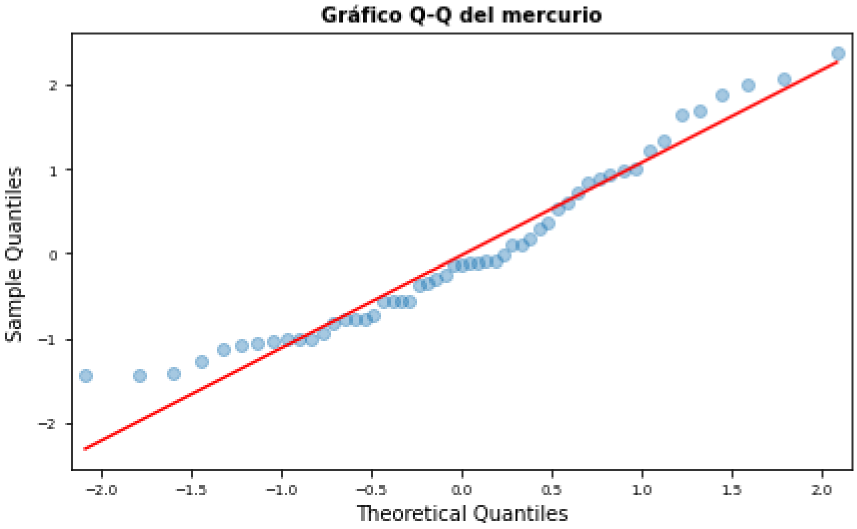
```
# Histograma + curva normal teórica
# ==============================================================================

# Valores de la media (mu) y desviación típica (sigma) de los datos
mu, sigma = stats.norm.fit(npeces)

# Valores teóricos de la normal en el rango observado
x_hat = np.linspace(min(npeces), max(npeces), num=100)
y_hat = stats.norm.pdf(x_hat, mu, sigma)

# Gráfico
fig, ax = plt.subplots(figsize=(7,4))
ax.plot(x_hat, y_hat, linewidth=2, label='normal')
ax.hist(x=npeces, density=True, bins=30, color="#3182bd", alpha=0.5)
ax.plot(npeces, np.full_like(npeces, -0.01), '|k', markeredgewidth=1)
ax.set_title('Distribución npeces')
ax.set_xlabel('npeces')
ax.set_ylabel('Densidad de probabilidad')
ax.legend();
```



ANOVA

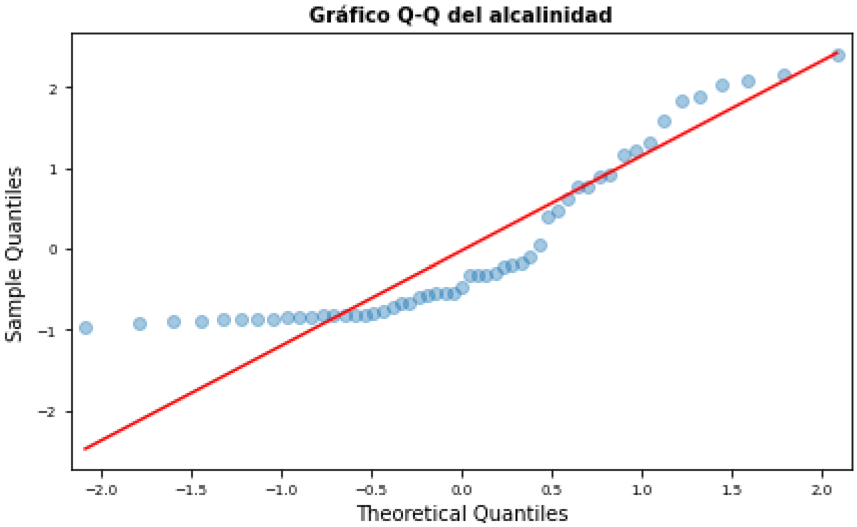
QQplots

In [31]:
```
# Gráfico Q-Q
# ==============================================================================
fig, ax = plt.subplots(figsize=(7,4))
sm.qqplot(
    mercurio,
    fit   = True,
    line  = 'q',
    alpha = 0.4,
    lw    = 2,
    ax    = ax
)
ax.set_title('Gráfico Q-Q del mercurio', fontsize = 10,
             fontweight = "bold")
ax.tick_params(labelsize = 7)
```

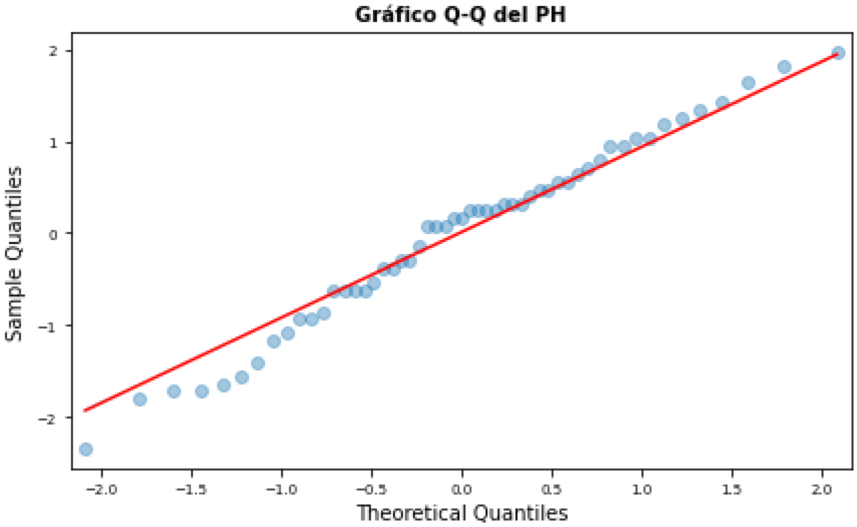


In [32]:

```
# Gráfico Q-Q
# ===============================================================================
fig, ax = plt.subplots(figsize=(7,4))
sm.qqplot(
    alcalinidad,
    fit   = True,
    line  = 'q',
    alpha = 0.4,
    lw    = 2,
    ax    = ax
)
ax.set_title('Gráfico Q-Q del alcalinidad', fontsize = 10,
             fontweight = "bold")
ax.tick_params(labelsize = 7)
```

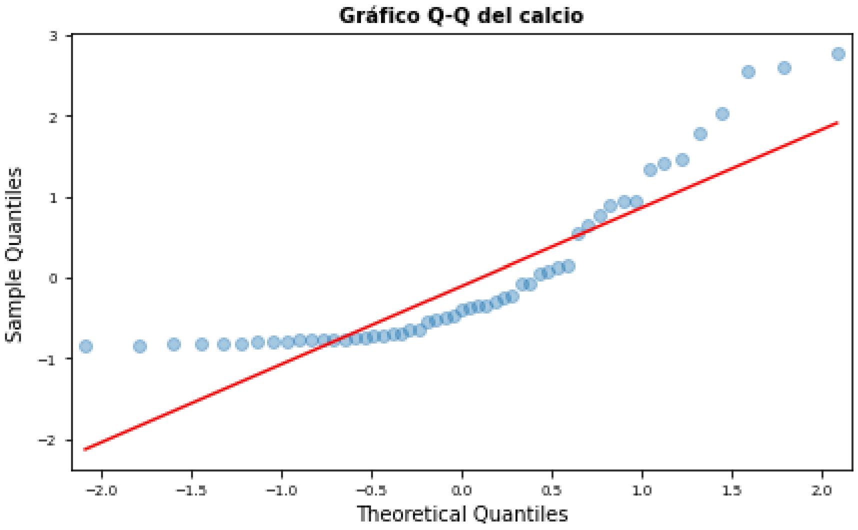


In [33]:
```
# Gráfico Q-Q
# ==============================================================================
fig, ax = plt.subplots(figsize=(7,4))
sm.qqplot(
    PH,
    fit   = True,
    line  = 'q',
    alpha = 0.4,
    lw    = 2,
    ax    = ax
)
ax.set_title('Gráfico Q-Q del PH', fontsize = 10,
             fontweight = "bold")
ax.tick_params(labelsize = 7)
```

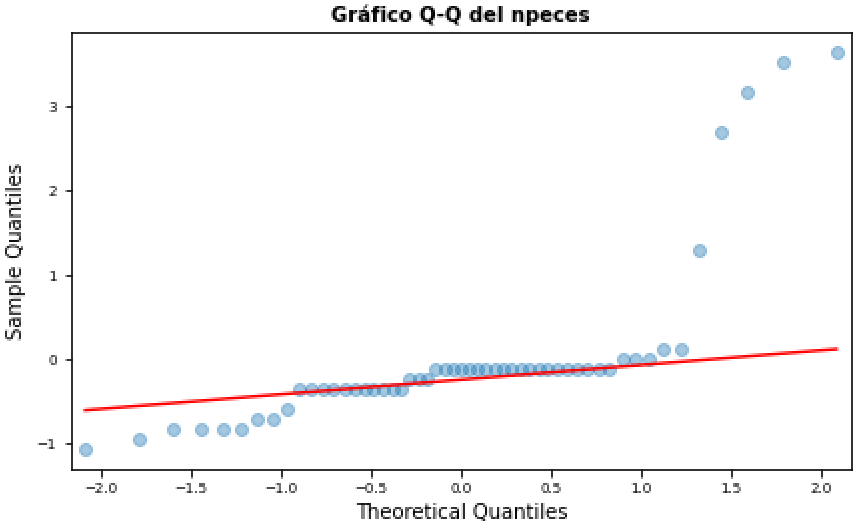


In [34]:

```
# Gráfico Q-Q
# ==============================================================================
fig, ax = plt.subplots(figsize=(7,4))
sm.qqplot(
    calcio,
    fit   = True,
    line  = 'q',
    alpha = 0.4,
    lw    = 2,
    ax    = ax
)
ax.set_title('Gráfico Q-Q del calcio', fontsize = 10,
             fontweight = "bold")
ax.tick_params(labelsize = 7)
```



In [35]:
```
# Gráfico Q-Q
# ==============================================================================
fig, ax = plt.subplots(figsize=(7,4))
sm.qqplot(
    npeces,
    fit   = True,
    line  = 'q',
    alpha = 0.4,
    lw    = 2,
    ax    = ax
)
ax.set_title('Gráfico Q-Q del npeces', fontsize = 10,
             fontweight = "bold")
ax.tick_params(labelsize = 7)
```



Análisis de Hipótesis

In [36]:
```
print('Kursotis:', stats.kurtosis(mercurio))
print('Skewness:', stats.skew(mercurio))
```

```
Kursotis: -0.5392793261283986
Skewness: 0.615985316604268
```

In [37]:
```
print('Kursotis:', stats.kurtosis(alcalinidad))
print('Skewness:', stats.skew(alcalinidad))
```

```
Kursotis: -0.3723123427917301
Skewness: 0.995971540190347
```

In [38]:
```
print('Kursotis:', stats.kurtosis(PH))
print('Skewness:', stats.skew(PH))
```

```
Kursotis: -0.5316990819046379
Skewness: -0.25300371445201947
```

In [39]:
```
print('Kursotis:', stats.kurtosis(calcio))
print('Skewness:', stats.skew(calcio))
```

```
Kursotis: 0.7533350422627985
Skewness: 1.3423994221171138
```

In [40]:
```
print('Kursotis:', stats.kurtosis(npeces))
print('Skewness:', stats.skew(npeces))
```

```
Kursotis: 6.358775112078357
Skewness: 2.655682430492345
```

##

In [41]:
```
# Shapiro-Wilk test
# ==============================================================================
shapiro_test = stats.shapiro(mercurio)
shapiro_test
```

Out[41]:
```
ShapiroResult(statistic=0.9421269297599792, pvalue=0.012508947402238846)
```

In [42]:
```
# Shapiro-Wilk test
# ==============================================================================
shapiro_test = stats.shapiro(alcalinidad)
shapiro_test
```

Out[42]:
```
ShapiroResult(statistic=0.8203012943267822, pvalue=1.5374524764411035e-06)
```

In [43]:
```
# Shapiro-Wilk test
# ==============================================================================
shapiro_test = stats.shapiro(PH)
shapiro_test
```

Out[43]:
```
ShapiroResult(statistic=0.9809678196907043, pvalue=0.5551783442497253)
```

In [44]:

```
# Shapiro-Wilk test
# ==============================================================================
shapiro_test = stats.shapiro(calcio)
shapiro_test
```

Out[44]:

```
ShapiroResult(statistic=0.7913432121276855, pvalue=3.090418942974793e-07)
```

In [45]:

```
# Shapiro-Wilk test
# ==============================================================================
shapiro_test = stats.shapiro(npeces)
shapiro_test
```

Out[45]:

```
ShapiroResult(statistic=0.5829699039459229, pvalue=4.9876252433689316e-11)
```

In [46]:

```
# D'Agostino's K-squared test
# ==============================================================================
k2, p_value = stats.normaltest(mercurio)
print(f"Estadístico = {k2}, p-value = {p_value}")
```

```
Estadístico = 4.220529123327874, p-value = 0.1212058957577724
```

In [47]:

```
# D'Agostino's K-squared test
# ==============================================================================
k2, p_value = stats.normaltest(alcalinidad)
print(f"Estadístico = {k2}, p-value = {p_value}")
```

```
Estadístico = 8.45127698441738, p-value = 0.014615999527870203
```

In [48]:

```
# D'Agostino's K-squared test
# ==============================================================================
k2, p_value = stats.normaltest(calcio)
print(f"Estadístico = {k2}, p-value = {p_value}")
```

```
Estadístico = 15.16551242729508, p-value = 0.0005091559447168802
```

In [49]:

```
# D'Agostino's K-squared test
# ==============================================================================
k2, p_value = stats.normaltest(npeces)
print(f"Estadístico = {k2}, p-value = {p_value}")
```

```
Estadístico = 47.47518669393924, p-value = 4.9078829412583656e-11
```

In [50]:

```
# D'Agostino's K-squared test
# ==============================================================================
k2, p_value = stats.normaltest(PH)
print(f"Estadístico = {k2}, p-value = {p_value}")
```

```
Estadístico = 1.1920185572316295, p-value = 0.5510061663930924
```

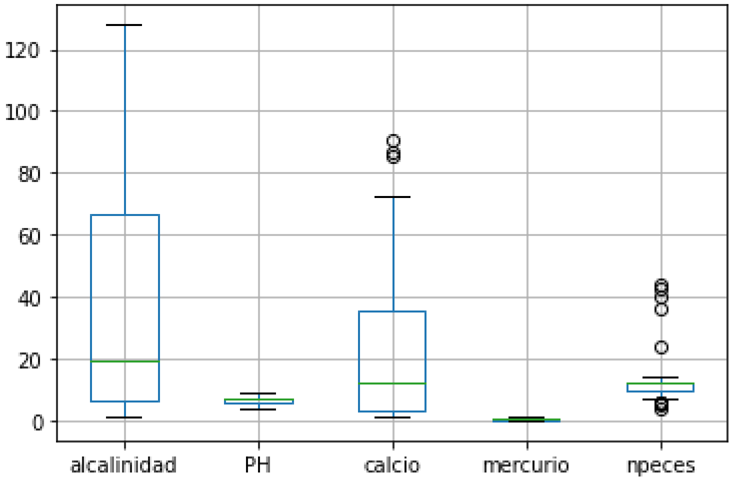
De acuerdo a los análisis realizados, en las variables de alcalinidad y calcio se rechaza la normalidad.

In [51]:

```
df.boxplot(column=['alcalinidad', 'PH', 'calcio', 'mercurio', 'npeces'])
```

Out[51]:

```
<AxesSubplot:>
```

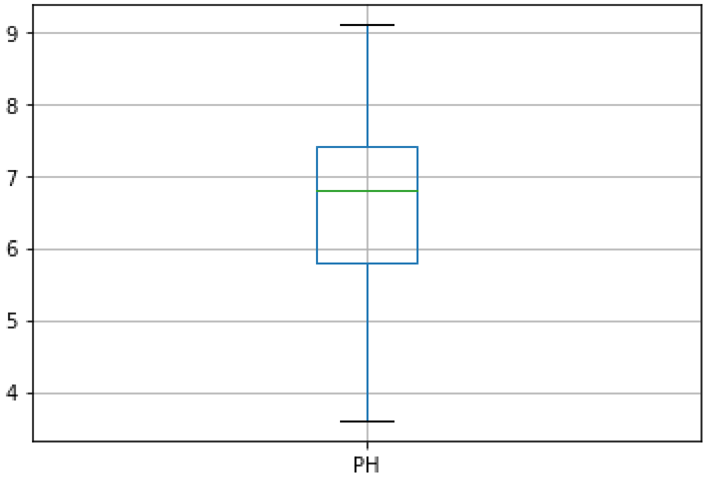


In [52]:

```
df.boxplot(column='PH')
```

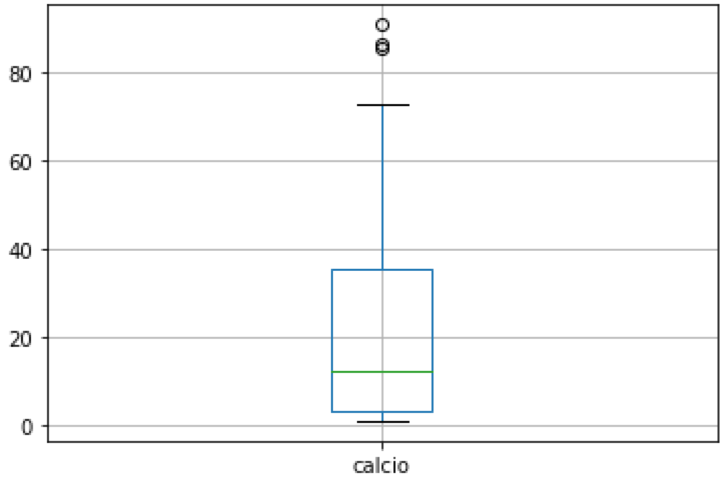
Out[52]:

```
<AxesSubplot:>
```



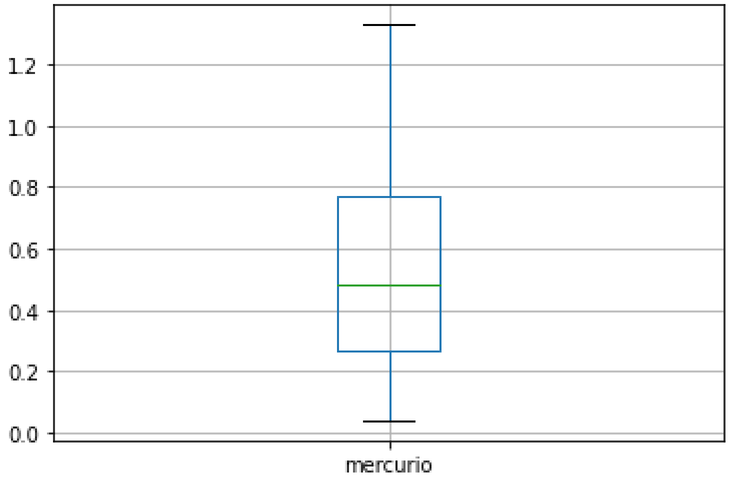
In [53]:
```
df.boxplot(column='calcio')
```

Out[53]:
```
<AxesSubplot:>
```



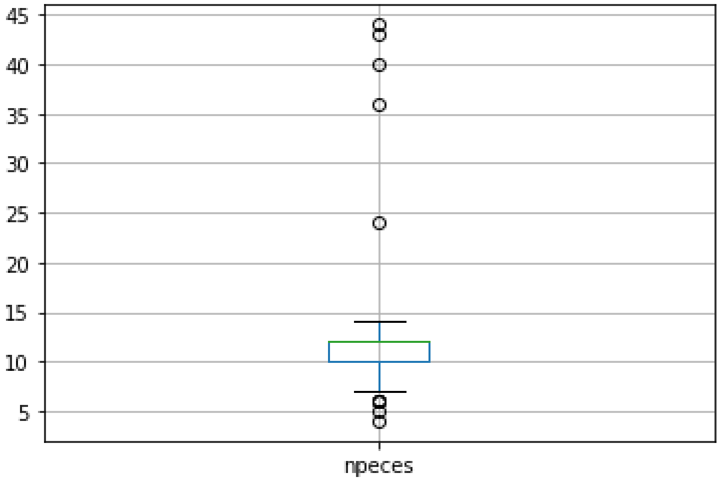
In [54]:
```
df.boxplot(column='mercurio')
```

Out[54]:
```
<AxesSubplot:>
```



```
In [55]: df.boxplot(column='npeces')
```

```
Out[55]: <AxesSubplot:>
```



La variable con valores atípicos notables es npeces.

In [56]:

```
df.groupby('mercurio').size()
```

Out[56]:

```
mercurio
0.04    2
0.05    1
0.10    1
0.15    1
0.16    1
0.17    1
0.18    1
0.19    3
0.21    1
0.25    1
0.27    3
0.28    1
0.34    4
0.40    1
0.41    1
0.43    1
0.44    1
0.48    2
0.49    2
0.50    2
0.52    1
0.56    2
0.59    1
0.63    1
0.65    1
0.71    1
0.73    1
0.77    1
0.81    1
0.83    1
0.84    1
0.86    1
0.87    1
0.94    1
0.98    1
1.08    1
1.10    1
1.16    1
1.20    1
1.23    1
1.33    1
dtype: int64
```

fig, ax = plt.subplots(1, 1, figsize=(8, 4)) sns.boxplot(x="alcalinidad", y="mercurio", data=df, ax=ax) sns.swarmplot(x="alcalinidad", y="mercurio", data=df, color='black', alpha = 0.5, ax=ax);

In [57]:

```
fvalue, pvalue = stats.f_oneway(df['alcalinidad'], df['PH'], df['calcio'], df['np
```

In [58]:
```
print(fvalue, pvalue)
```

```
17.701446360772287 2.847726889143591e-10
```

In [59]:
```
fvalue, pvalue = stats.f_oneway(df['npeces'], df['mercurio'])
```

In [60]:
```
print(fvalue, pvalue)
```

```
113.35321181638591 2.416358468853054e-18
```

In [61]:
```
fvalue, pvalue = stats.f_oneway(df['alcalinidad'], df['mercurio'])
```

In [62]:
```
print(fvalue, pvalue)
```

```
49.717429234590995 2.0203527876566043e-10
```

El p-value es menor a 0.05 por lo que se rechaza la hipótesis nula.

Regression

In [63]:
```
df.drop(['estimacion', 'nombre'], axis=1, inplace=True)
```

In [64]:
```
X = df.iloc[:,df.columns != 'mercurio']
y = df.mercurio
```

In [65]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn import preprocessing
from sklearn.metrics import accuracy_score, classification_report, confusion_matr
```

In [66]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_
```

In [70]:
```
df.head()
```

Out[70]:

	alcalinidad	PH	calcio	clorofila	mercurio	npeces	minmercurio	maxmercurio	edad
id									
1	5.9	6.1	3.0	0.7	1.23	5	0.85	1.43	1
2	3.5	5.1	1.9	3.2	1.33	7	0.92	1.90	0
3	116.0	9.1	44.1	128.3	0.04	6	0.04	0.06	0
4	39.4	6.9	16.4	3.5	0.44	12	0.13	0.84	0
5	2.5	4.6	2.9	1.8	1.20	12	0.69	1.50	1

In [88]:
```
lm = sfm.ols(formula="mercurio~alcalinidad+PH+calcio+clorofila+npeces+edad", data
```

In [89]:
```
lm.pvalues
```

Out[89]:
```
Intercept      0.001232
alcalinidad    0.007774
PH             0.428976
calcio         0.090786
clorofila      0.124741
npeces         0.297340
edad           0.750024
dtype: float64
```

```
In [90]: lm.summary()
```

```
Out[90]:
OLS Regression Results

     Dep. Variable:         mercurio            R-squared:    0.464
            Model:               OLS        Adj. R-squared:    0.395
           Method:     Least Squares          F-statistic:    6.650
             Date:  Sat, 17 Sep 2022  Prob (F-statistic): 4.12e-05
             Time:          16:21:54       Log-Likelihood:  -1.1337
 No. Observations:                53                  AIC:    16.27
     Df Residuals:                46                  BIC:    30.06
         Df Model:                 6
  Covariance Type:         nonrobust

                coef  std err       t   P>|t|  [0.025  0.975]
  Intercept   0.9289    0.270   3.444   0.001   0.386   1.472
 alcalinidad -0.0060    0.002  -2.784   0.008  -0.010  -0.002
         PH  -0.0391    0.049  -0.798   0.429  -0.138   0.059
     calcio   0.0048    0.003   1.728   0.091  -0.001   0.010
  clorofila  -0.0026    0.002  -1.564   0.125  -0.006   0.001
    npeces    0.0048    0.005   1.054   0.297  -0.004   0.014
       edad  -0.0348    0.108  -0.321   0.750  -0.253   0.184

      Omnibus:  6.335     Durbin-Watson:  1.468
Prob(Omnibus):  0.042  Jarque-Bera (JB):  5.988
         Skew:  0.823          Prob(JB): 0.0501
     Kurtosis:  3.042          Cond. No.   518.


Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

El valor R-squared de 0.46 por lo que se explica el 46% del modelo. Además, las variables obtuvieron un p-value menor a 0.05, por lo que se encuentran en la zona de aceptación, con valores negativos en algunos casos que afectan negativamente la concentración del mercurio.

```
In [ ]:
```