PRESENTACIÓN 1: Dataset Relacionado con la Calidad del Agua

1. ammonia
2. arsenic
3. barium
4. cadmium
5. chloramine
6. chromium
7. copper
8. flouride
9. bacteria
10. viruses
11. lead
12. nitrates
13. nitrites
14. mercury
15. perchlorate
16. radium
17. selenium
18. silver
19. uranium
20. is_safe

Integrantes:

- Alondra Villanueva
- Alberto San Martín Concha

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#to ignore warnings
import warnings
warnings.filterwarnings('ignore')
```

Haz doble clic (o pulsa Intro) para editar

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/Minor-DataScience/water-quality/waterQuality1.csv')
df.head()
```

	aluminium	ammonia	arsenic	barium	cadmium	chloramine	chromium	copper	flouride
0	1.65	9.08	0.04	2.85	0.007	0.35	0.83	0.17	0.05
1	2.32	21.16	0.01	3.31	0.002	5.28	0.68	0.66	0.90
2	1.01	14.02	0.04	0.58	0.008	4.24	0.53	0.02	0.99
3	1.36	11.33	0.04	2.96	0.001	7.23	0.03	1.66	1.08
4	0.92	24.33	0.03	0.20	0.006	2.67	0.69	0.57	0.61

5 rows × 21 columns

```
df.tail()
```

	aluminium	ammonia	arsenic	barium	cadmium	chloramine	chromium	copper	flouri
7994	0.05	7.78	0.00	1.95	0.04	0.10	0.03	0.03	1.
7995	0.05	24.22	0.02	0.59	0.01	0.45	0.02	0.02	1.
7996	0.09	6.85	0.00	0.61	0.03	0.05	0.05	0.02	0.
7997	0.01	10	0.01	2.00	0.00	2.00	0.00	0.09	0.
7998	0.04	6.85	0.01	0.70	0.03	0.05	0.01	0.03	1.

5 rows × 21 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7999 entries, 0 to 7998
Data columns (total 21 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   aluminium    7999 non-null   float64
 1   ammonia      7999 non-null   object
 2   arsenic      7999 non-null   float64
 3   barium       7999 non-null   float64
 4   cadmium      7999 non-null   float64
 5   chloramine   7999 non-null   float64
 6   chromium     7999 non-null   float64
 7   copper       7999 non-null   float64
 8   flouride     7999 non-null   float64
 9   bacteria     7999 non-null   float64
 10  viruses      7999 non-null   float64
 11  lead         7999 non-null   float64
 12  nitrates     7999 non-null   float64
 13  nitrites     7999 non-null   float64
 14  mercury      7999 non-null   float64
 15  perchlorate  7999 non-null   float64
 16  radium       7999 non-null   float64
 17  selenium     7999 non-null   float64
 18  silver       7999 non-null   float64
 19  uranium      7999 non-null   float64
 20  is_safe      7999 non-null   object
dtypes: float64(19), object(2)
memory usage: 1.3+ MB
```

Haz doble clic (o pulsa Intro) para editar

```
df.nunique()
```

```
aluminium       495
ammonia        2564
arsenic         107
barium          480
cadmium          23
chloramine      812
chromium         91
copper          201
flouride        151
bacteria        101
viruses          61
lead            200
nitrates       1803
nitrites        280
mercury          11
perchlorate    2999
radium          735
selenium         11
silver           51
uranium          10
is_safe           3
dtype: int64
```

El siguiente codigo es para evaluar, sumar los valores null por cada columna

```
df.isnull().sum()
```

```
aluminium      0
ammonia        0
arsenic        0
```

```
barium         0
cadmium        0
chloramine     0
chromium       0
copper         0
flouride       0
bacteria       0
viruses        0
lead           0
nitrates       0
nitrites       0
mercury        0
perchlorate    0
radium         0
selenium       0
silver         0
uranium        0
is_safe        0
dtype: int64
```

Porcentaje de valores perdidos por cada columna

```
(df.isnull().sum()/(len(df)))*100
```

```
aluminium      0.0
ammonia        0.0
arsenic        0.0
barium         0.0
cadmium        0.0
chloramine     0.0
chromium       0.0
copper         0.0
flouride       0.0
bacteria       0.0
viruses        0.0
lead           0.0
nitrates       0.0
nitrites       0.0
mercury        0.0
perchlorate    0.0
radium         0.0
selenium       0.0
silver         0.0
uranium        0.0
is_safe        0.0
dtype: float64
```

describe()– Provide a statistics summary of data belonging to numerical datatype such as int, float

```
df.describe().T
```

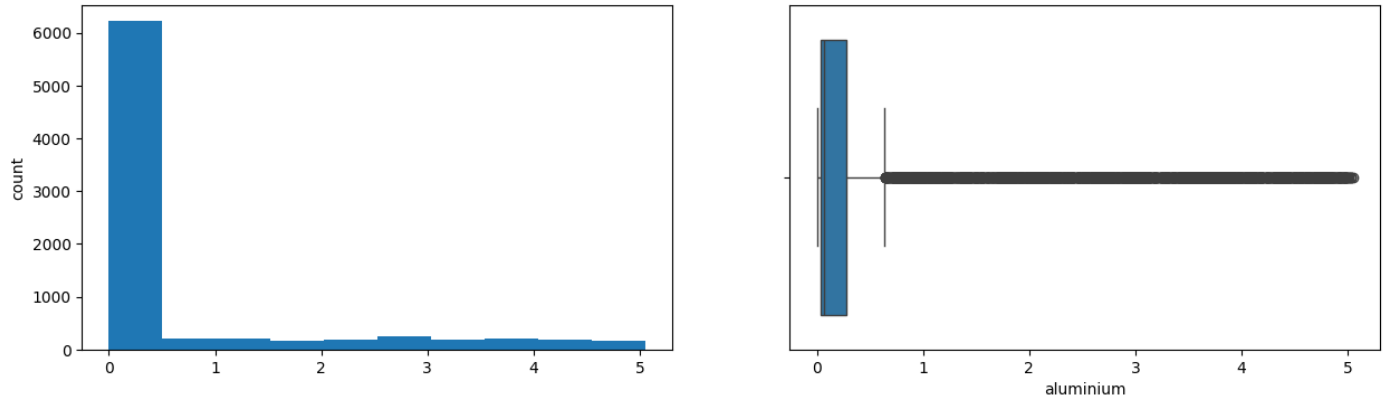
	count	mean	std	min	25%	50%	75%	max
aluminium	7999.0	0.666158	1.265145	0.0	0.040	0.070	0.280	5.05
arsenic	7999.0	0.161445	0.252590	0.0	0.030	0.050	0.100	1.05
barium	7999.0	1.567715	1.216091	0.0	0.560	1.190	2.480	4.94
cadmium	7999.0	0.042806	0.036049	0.0	0.008	0.040	0.070	0.13
chloramine	7999.0	2.176831	2.567027	0.0	0.100	0.530	4.240	8.68
chromium	7999.0	0.247226	0.270640	0.0	0.050	0.090	0.440	0.90
copper	7999.0	0.805857	0.653539	0.0	0.090	0.750	1.390	2.00
flouride	7999.0	0.771565	0.435373	0.0	0.405	0.770	1.160	1.50
bacteria	7999.0	0.319665	0.329485	0.0	0.000	0.220	0.610	1.00
viruses	7999.0	0.328583	0.378096	0.0	0.002	0.008	0.700	1.00
lead	7999.0	0.099450	0.058172	0.0	0.048	0.102	0.151	0.20
nitrates	7999.0	9.818822	5.541331	0.0	5.000	9.930	14.610	19.83
nitrites	7999.0	1.329961	0.573219	0.0	1.000	1.420	1.760	2.93
mercury	7999.0	0.005194	0.002967	0.0	0.003	0.005	0.008	0.01
perchlorate	7999.0	16.460299	17.687474	0.0	2.170	7.740	29.480	60.01
radium	7999.0	2.920548	2.323009	0.0	0.820	2.410	4.670	7.99
selenium	7999.0	0.049685	0.028770	0.0	0.020	0.050	0.070	0.10
silver	7999.0	0.147781	0.143551	0.0	0.040	0.080	0.240	0.50
uranium	7999.0	0.044673	0.026904	0.0	0.020	0.050	0.070	0.09

```
cat_cols=df.select_dtypes(include=['object']).columns
num_cols = df.select_dtypes(include=np.number).columns.tolist()
print("Categorical Variables:")
print(cat_cols)
print("Numerical Variables:")
print(num_cols)
```

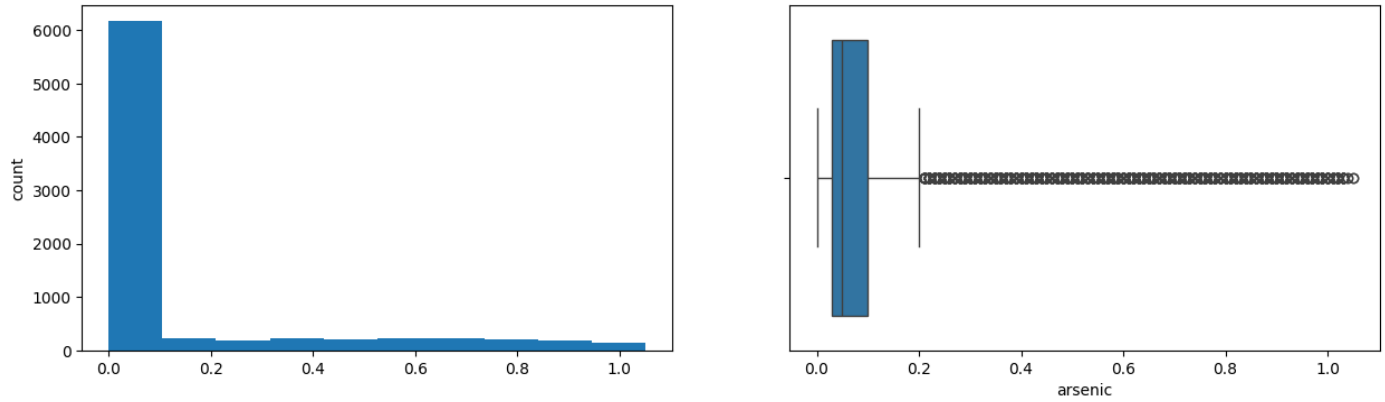
```
Categorical Variables:
Index(['ammonia', 'is_safe'], dtype='object')
Numerical Variables:
['aluminium', 'arsenic', 'barium', 'cadmium', 'chloramine', 'chromium', 'copper', 'flouride', 'bacteria', 'viruses', 'lead', 'nitrates',
```

```
for col in num_cols:
    print(col)
    print('Skew :', round(df[col].skew(), 2))
    plt.figure(figsize = (15, 4))
    plt.subplot(1, 2, 1)
    df[col].hist(grid=False)
    plt.ylabel('count')
    plt.subplot(1, 2, 2)
    sns.boxplot(x=df[col])
    plt.show()
```

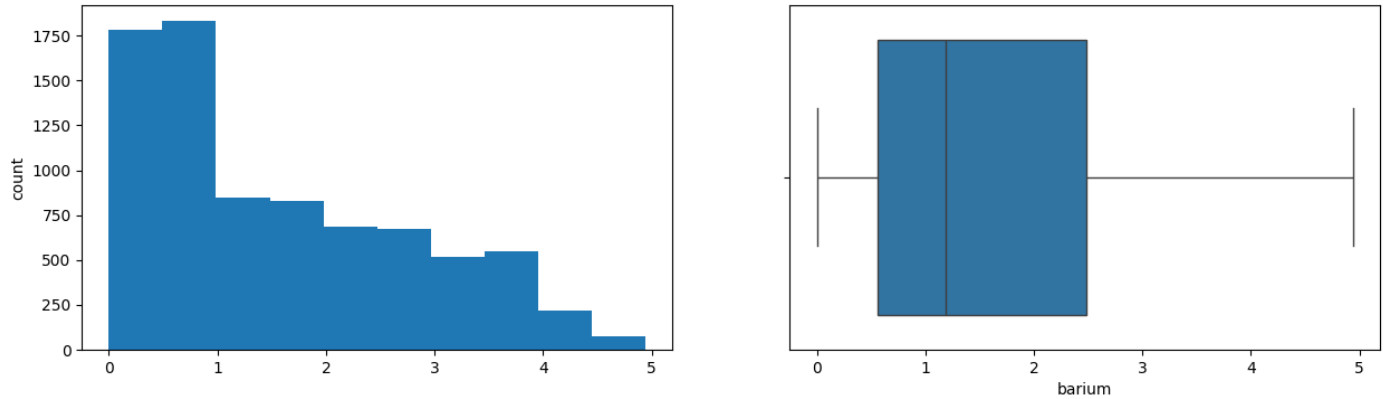
```
aluminium
Skew : 2.01
```



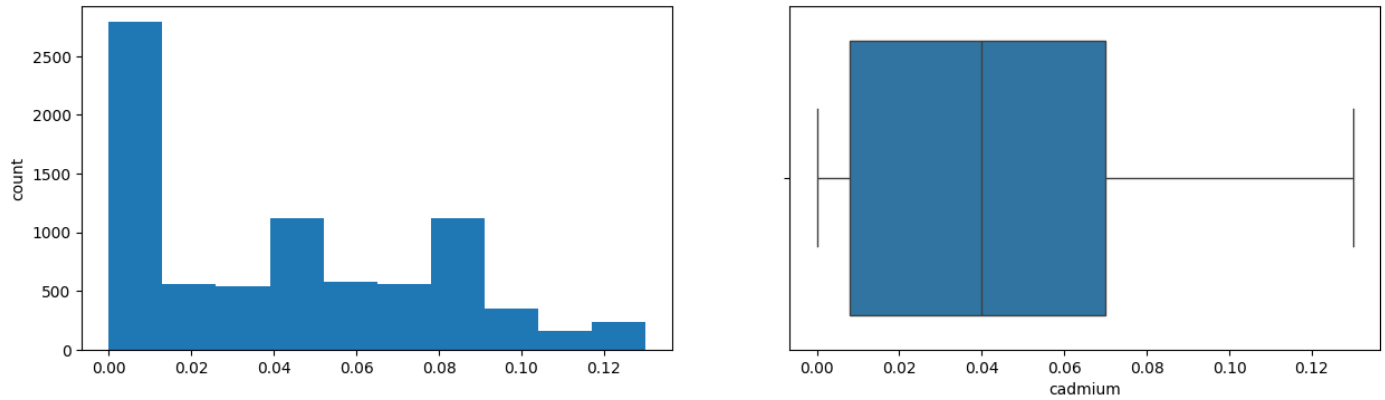

```
arsenic
Skew : 1.99
```



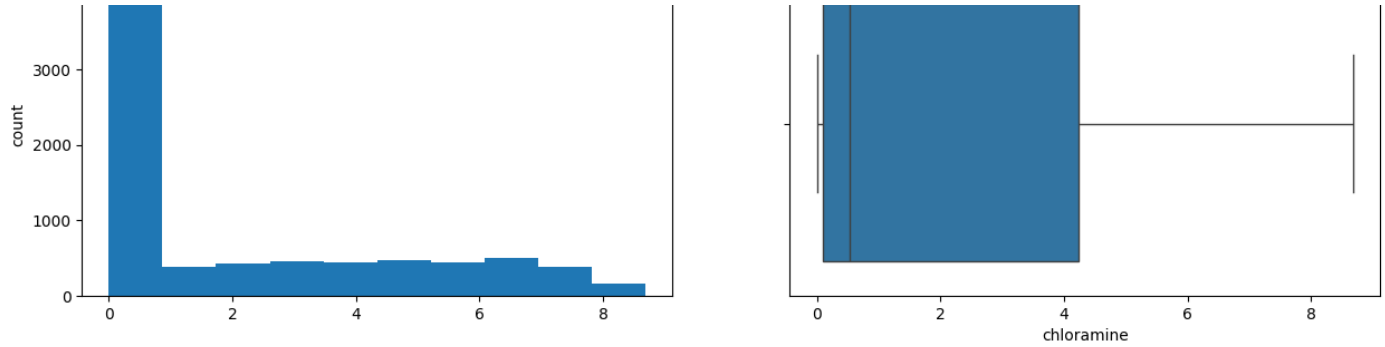

```
barium
Skew : 0.66
```



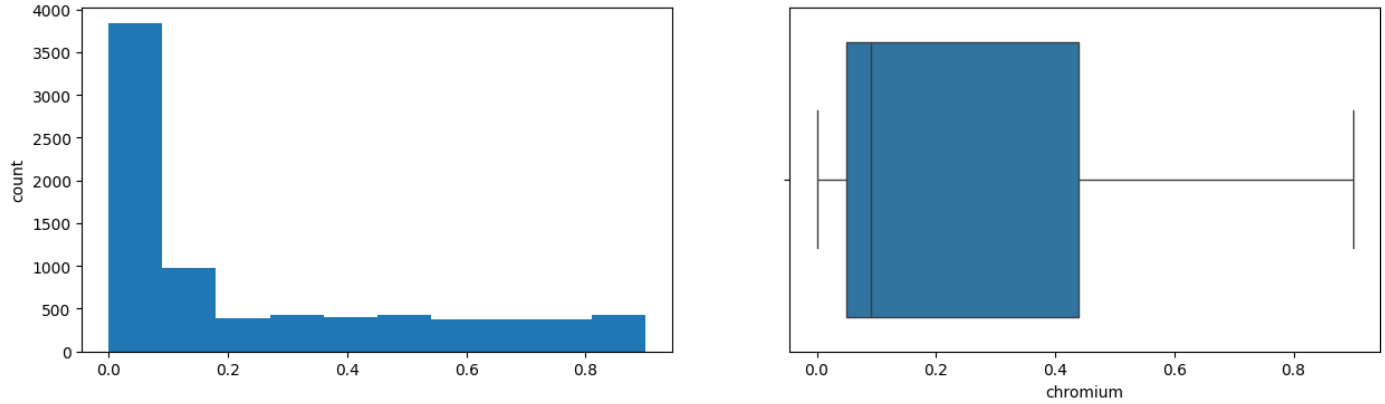

```
cadmium
Skew : 0.48
```



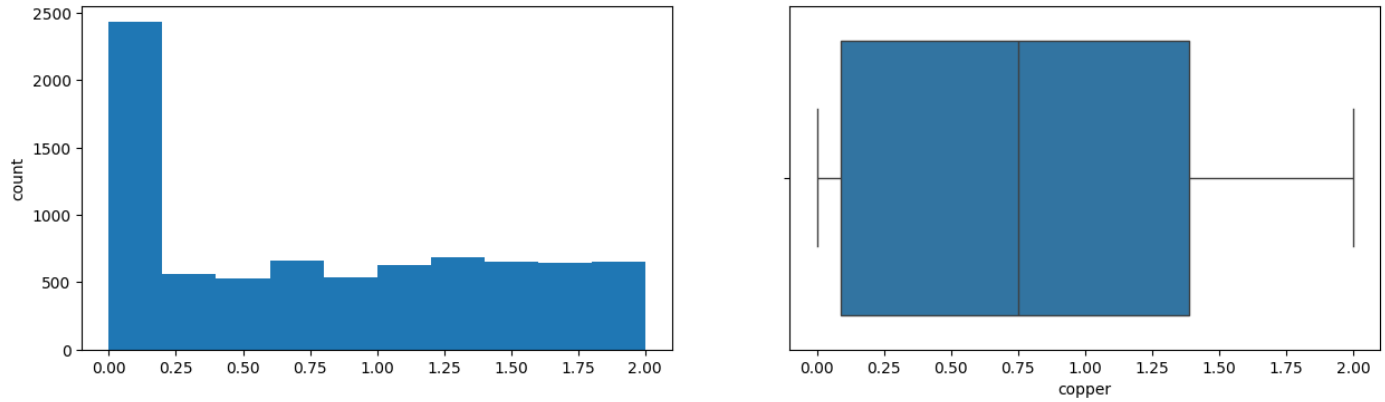

```
chloramine
Skew : 0.89
```



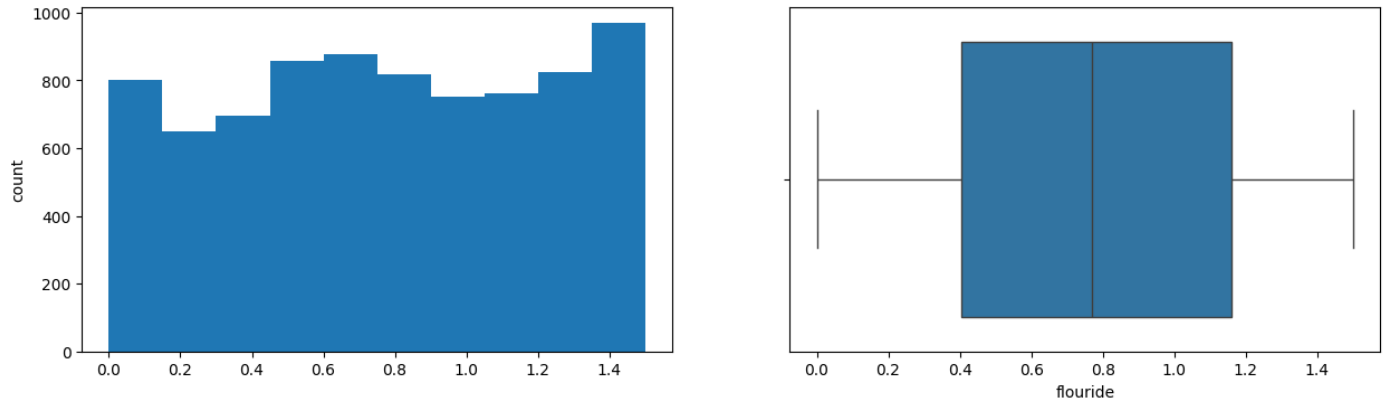

```
chromium
Skew : 1.03
```




```
copper
Skew : 0.25
```




```
flouride
Skew : -0.04
```




```
bacteria
Skew : 0.55
```