

# Evaluaci?n 1

March 9, 2018

```
In [1]: # Cargamos las bibliotecas
import pandas as pd
import numpy as np
from datetime import datetime
import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [2]: # Leemos los dos archivos, saltandonos las filas necesarias
df1 = pd.read_csv("sargento_201117.csv", header=None, names=['N', 'Date', 'Press', 'Temp', 'WL'])
df2 = pd.read_csv("sargento-salinidad-201117.csv", header=None, names=['N', 'Date', 'CHR', 'Temp', 'SC', 'Sal'],
                  skiprows=3, sep=',')
```

```
#Eliminamos la última fila un archivo
df1=df1.drop(df1.index[len(df1)-1])
```

```
#Se crean los data frames
df1=pd.DataFrame(df1)
df2=pd.DataFrame(df2)
```

```
In [3]: # Nuestros Datos
df1.head()
```

```
Out[3]:
```

	N	Date	Press	Temp	WL
0	1	10/26/2017 13:00:00	105.612	24.448	-0.150
1	2	10/26/2017 13:15:00	105.513	24.351	-0.160
2	3	10/26/2017 13:30:00	105.433	24.351	-0.168
3	4	10/26/2017 13:45:00	105.385	24.351	-0.173
4	5	10/26/2017 14:00:00	105.321	24.351	-0.179

```
In [4]: # Nuestros Datos
df2.head()
```

```
Out[4]:
```

	N	Date	CHR	Temp	SC	Sal
0	2	10/26/2017 13:00:00	54525.5	24.91	54622.1	36.1588
1	3	10/26/2017 13:15:00	54525.5	24.82	54719.0	36.2311
2	4	10/26/2017 13:30:00	54525.5	24.76	54783.8	36.2794
3	5	10/26/2017 13:45:00	54525.5	24.75	54794.6	36.2875
4	6	10/26/2017 14:00:00	54525.5	24.73	54816.2	36.3036

```
In [5]: # Tipo de datos
df1.dtypes
```

```
Out[5]: N          int64
Date         object
Press       float64
Temp        float64
WL          float64
dtype: object
```

```
In [6]: # Tipo de datos
df2.dtypes
```

```
Out[6]: N          int64
Date         object
CHR         float64
Temp        float64
SC          float64
Sal         float64
dtype: object
```

```
In [7]: # Convertir la cadena de caracteres 'Date' en variable temporal 'NDateTime'
df1['Ndt'] = pd.to_datetime(df1['Date'],format='%m/%d/%Y %H:%M:%S')
df1['month'] = df1['Ndt'].dt.month
df1.head()
```

```
Out[7]:
```

	N	Date	Press	Temp	WL	Ndt	month
0	1	10/26/2017 13:00:00	105.612	24.448	-0.150	2017-10-26 13:00:00	10
1	2	10/26/2017 13:15:00	105.513	24.351	-0.160	2017-10-26 13:15:00	10
2	3	10/26/2017 13:30:00	105.433	24.351	-0.168	2017-10-26 13:30:00	10
3	4	10/26/2017 13:45:00	105.385	24.351	-0.173	2017-10-26 13:45:00	10
4	5	10/26/2017 14:00:00	105.321	24.351	-0.179	2017-10-26 14:00:00	10

```
In [8]: # Convertir la cadena de caracteres 'Date' en variable temporal 'NDateTime'
df2['Ndt'] = pd.to_datetime(df2['Date'],format='%m/%d/%Y %H:%M:%S')
df2['month'] = df2['Ndt'].dt.month
df2.head()
```

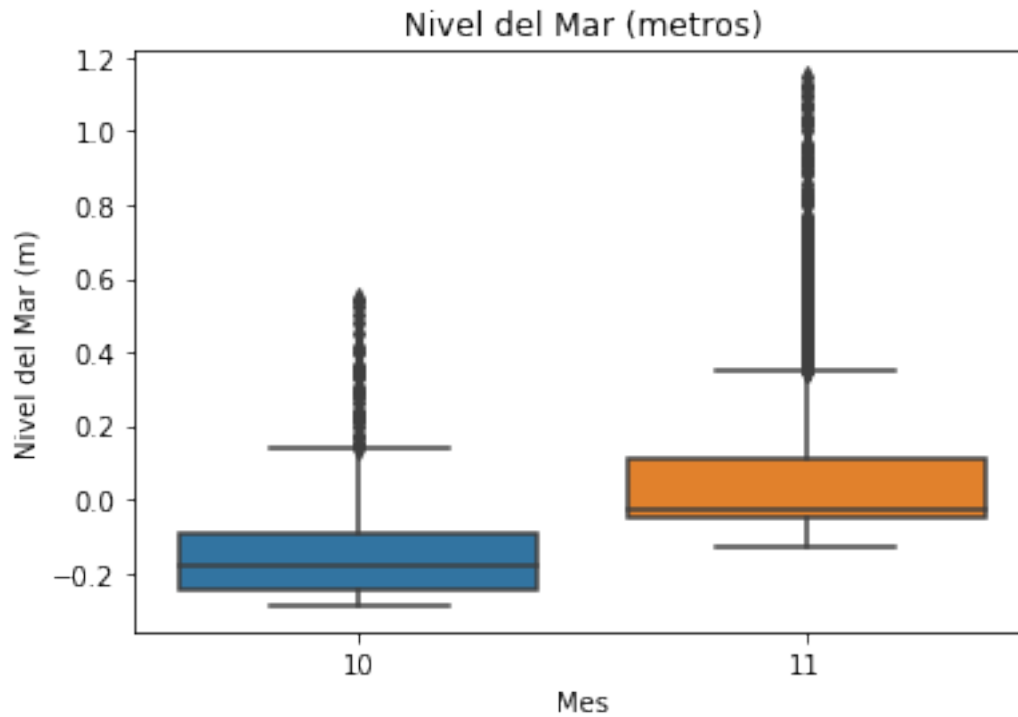
```
Out[8]:
```

	N	Date	CHR	Temp	SC	Sal	\
0	2	10/26/2017 13:00:00	54525.5	24.91	54622.1	36.1588	
1	3	10/26/2017 13:15:00	54525.5	24.82	54719.0	36.2311	
2	4	10/26/2017 13:30:00	54525.5	24.76	54783.8	36.2794	
3	5	10/26/2017 13:45:00	54525.5	24.75	54794.6	36.2875	
4	6	10/26/2017 14:00:00	54525.5	24.73	54816.2	36.3036	

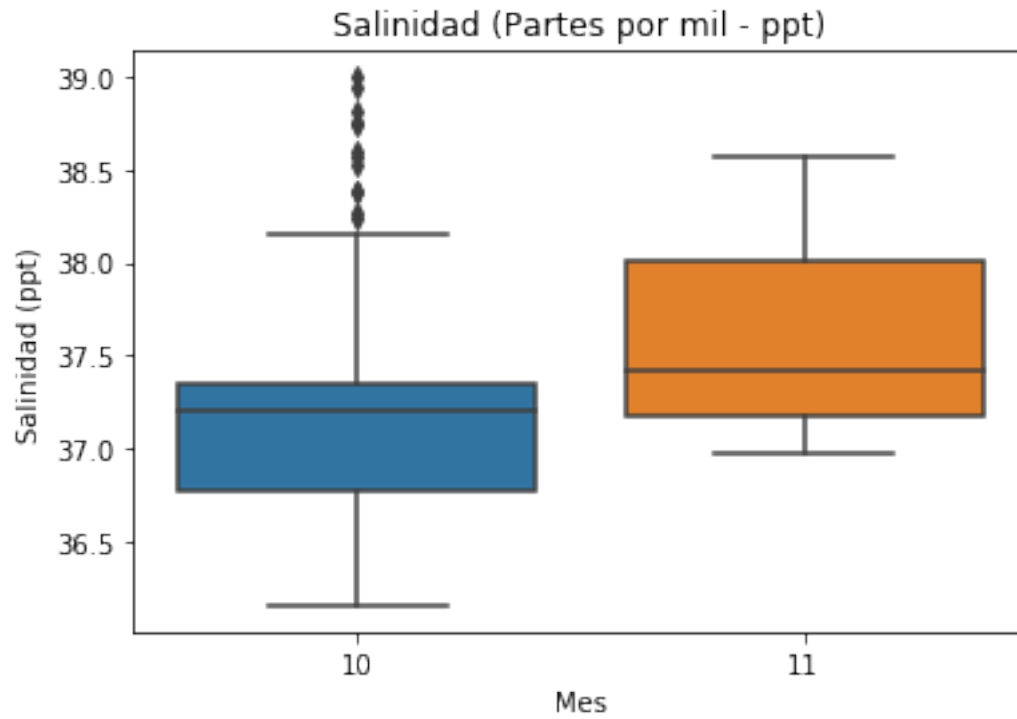
  

	Ndt	month
0	2017-10-26 13:00:00	10
1	2017-10-26 13:15:00	10
2	2017-10-26 13:30:00	10
3	2017-10-26 13:45:00	10
4	2017-10-26 14:00:00	10

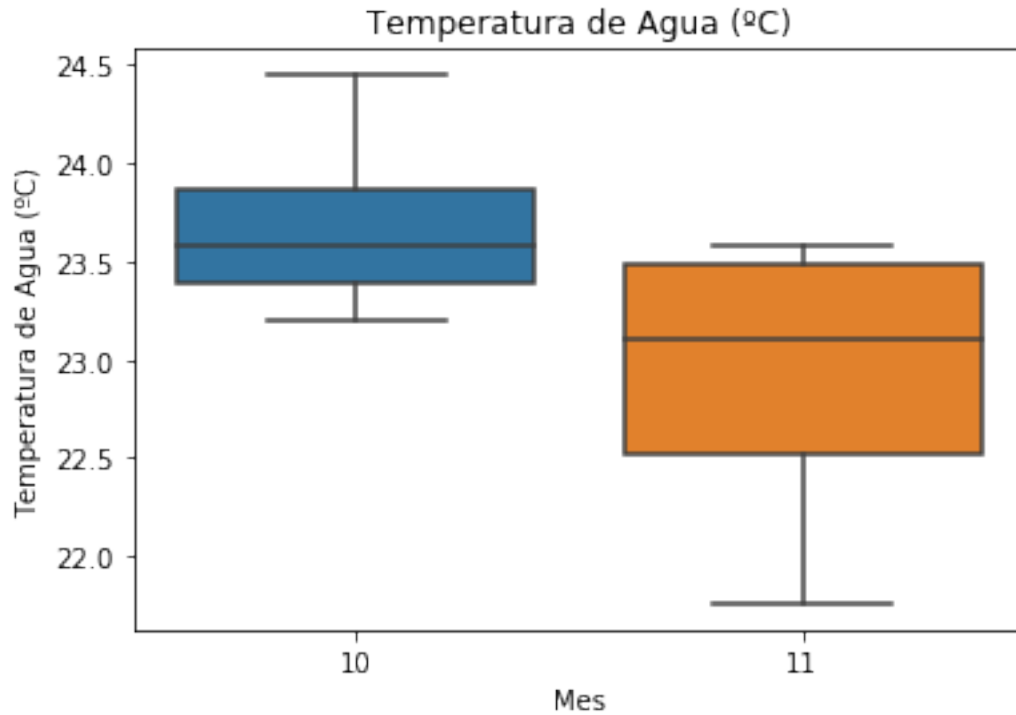
```
In [9]: # Gráfica de Nivel de mar (metros)
ax = sns.boxplot(x="month", y="WL", data=df1)
plt.title('Nivel del Mar (metros)')
plt.ylabel('Nivel del Mar (m)')
plt.xlabel('Mes')
plt.show()
```



```
In [10]: # Gráfica de Salinidad (Partes por mil - ppt)
ax = sns.boxplot(x="month", y="Sal", data=df2)
plt.title('Salinidad (Partes por mil - ppt)')
plt.ylabel('Salinidad (ppt)')
plt.xlabel('Mes')
plt.show()
```



```
In [11]: # Gráfica de Temperatura de Agua (°C)
ax = sns.boxplot(x="month", y="Temp", data=df1)
plt.title('Temperatura de Agua (°C)')
plt.ylabel('Temperatura de Agua (°C)')
plt.xlabel('Mes')
plt.show()
```



```
In [12]: # Descripción de datos
df1.describe()
```

```
Out[12]:
```

	N	Press	Temp	WL	month
count	2394.000000	2394.000000	2394.000000	2394.000000	2394.000000
mean	1197.500000	107.430007	23.120883	0.030863	10.781119
std	691.232595	2.371844	0.563555	0.235974	0.413574
min	1.000000	104.229000	21.760000	-0.288000	10.000000
25%	599.250000	106.407000	22.525000	-0.071000	11.000000
50%	1197.500000	106.764000	23.388000	-0.035000	11.000000
75%	1795.750000	107.305000	23.484000	0.018750	11.000000
max	2394.000000	118.641000	24.448000	1.146000	11.000000

```
In [13]: # Descripción de datos
df2.describe()
```

```
Out[13]:
```

	N	CHR	Temp	SC	Sal \
count	2394.000000	2394.000000	2394.000000	2394.000000	2394.000000
mean	1198.500000	54524.972807	23.316646	56386.831662	37.479737
std	691.232595	11.876669	0.547033	619.501987	0.464974
min	2.000000	54105.700000	21.490000	54622.100000	36.158800
25%	600.250000	54525.500000	22.730000	55949.700000	37.151400
50%	1198.500000	54525.500000	23.490000	56185.600000	37.328300
75%	1796.750000	54525.500000	23.700000	57053.700000	37.980300

```
max      2395.000000  54525.500000      24.910000  58398.700000      38.994200
```

```

              month
count  2394.000000
mean    10.781119
std      0.413574
min     10.000000
25%     11.000000
50%     11.000000
75%     11.000000
max     11.000000

```

```
In [14]: # Juntamos ambos archivos
pd.concat([df1, df2], axis=1, join_axes=[df2.index])
```

```
Out[14]:
```

	N	Date	Press	Temp	WL	Ndt \
0	1	10/26/2017 13:00:00	105.612	24.448	-0.150	2017-10-26 13:00:00
1	2	10/26/2017 13:15:00	105.513	24.351	-0.160	2017-10-26 13:15:00
2	3	10/26/2017 13:30:00	105.433	24.351	-0.168	2017-10-26 13:30:00
3	4	10/26/2017 13:45:00	105.385	24.351	-0.173	2017-10-26 13:45:00
4	5	10/26/2017 14:00:00	105.321	24.351	-0.179	2017-10-26 14:00:00
5	6	10/26/2017 14:15:00	105.273	24.351	-0.184	2017-10-26 14:15:00
6	7	10/26/2017 14:30:00	105.225	24.351	-0.189	2017-10-26 14:30:00
7	8	10/26/2017 14:45:00	105.177	24.351	-0.193	2017-10-26 14:45:00
8	9	10/26/2017 15:00:00	105.145	24.351	-0.196	2017-10-26 15:00:00
9	10	10/26/2017 15:15:00	105.113	24.351	-0.200	2017-10-26 15:15:00
10	11	10/26/2017 15:30:00	105.097	24.351	-0.201	2017-10-26 15:30:00
11	12	10/26/2017 15:45:00	105.065	24.351	-0.204	2017-10-26 15:45:00
12	13	10/26/2017 16:00:00	105.033	24.351	-0.208	2017-10-26 16:00:00
13	14	10/26/2017 16:15:00	105.017	24.351	-0.209	2017-10-26 16:15:00
14	15	10/26/2017 16:30:00	104.985	24.351	-0.212	2017-10-26 16:30:00
15	16	10/26/2017 16:45:00	104.953	24.351	-0.216	2017-10-26 16:45:00
16	17	10/26/2017 17:00:00	104.969	24.351	-0.214	2017-10-26 17:00:00
17	18	10/26/2017 17:15:00	104.969	24.351	-0.214	2017-10-26 17:15:00
18	19	10/26/2017 17:30:00	104.937	24.351	-0.217	2017-10-26 17:30:00
19	20	10/26/2017 17:45:00	104.937	24.351	-0.217	2017-10-26 17:45:00
20	21	10/26/2017 18:00:00	104.953	24.351	-0.216	2017-10-26 18:00:00
21	22	10/26/2017 18:15:00	104.969	24.351	-0.214	2017-10-26 18:15:00
22	23	10/26/2017 18:30:00	104.969	24.351	-0.214	2017-10-26 18:30:00
23	24	10/26/2017 18:45:00	104.953	24.351	-0.216	2017-10-26 18:45:00
24	25	10/26/2017 19:00:00	104.937	24.351	-0.217	2017-10-26 19:00:00
25	26	10/26/2017 19:15:00	104.921	24.351	-0.219	2017-10-26 19:15:00
26	27	10/26/2017 19:30:00	104.889	24.351	-0.222	2017-10-26 19:30:00
27	28	10/26/2017 19:45:00	104.889	24.351	-0.222	2017-10-26 19:45:00
28	29	10/26/2017 20:00:00	104.889	24.351	-0.222	2017-10-26 20:00:00
29	30	10/26/2017 20:15:00	104.889	24.351	-0.222	2017-10-26 20:15:00
...	...	...	...	...	...	...
2364	2365	11/20/2017 04:00:00	112.375	22.046	0.523	2017-11-20 04:00:00

2365	2366	11/20/2017	04:15:00	111.766	22.046	0.462	2017-11-20	04:15:00
2366	2367	11/20/2017	04:30:00	111.156	22.046	0.402	2017-11-20	04:30:00
2367	2368	11/20/2017	04:45:00	110.531	22.046	0.339	2017-11-20	04:45:00
2368	2369	11/20/2017	05:00:00	109.970	22.046	0.284	2017-11-20	05:00:00
2369	2370	11/20/2017	05:15:00	109.345	22.046	0.221	2017-11-20	05:15:00
2370	2371	11/20/2017	05:30:00	108.417	22.046	0.129	2017-11-20	05:30:00
2371	2372	11/20/2017	05:45:00	107.457	22.046	0.034	2017-11-20	05:45:00
2372	2373	11/20/2017	06:00:00	106.933	22.142	-0.019	2017-11-20	06:00:00
2373	2374	11/20/2017	06:15:00	106.920	22.238	-0.020	2017-11-20	06:15:00
2374	2375	11/20/2017	06:30:00	106.952	22.238	-0.017	2017-11-20	06:30:00
2375	2376	11/20/2017	06:45:00	106.972	22.333	-0.015	2017-11-20	06:45:00
2376	2377	11/20/2017	07:00:00	106.972	22.333	-0.015	2017-11-20	07:00:00
2377	2378	11/20/2017	07:15:00	107.020	22.333	-0.010	2017-11-20	07:15:00
2378	2379	11/20/2017	07:30:00	106.988	22.333	-0.013	2017-11-20	07:30:00
2379	2380	11/20/2017	07:45:00	107.004	22.333	-0.012	2017-11-20	07:45:00
2380	2381	11/20/2017	08:00:00	107.020	22.333	-0.010	2017-11-20	08:00:00
2381	2382	11/20/2017	08:15:00	107.004	22.333	-0.012	2017-11-20	08:15:00
2382	2383	11/20/2017	08:30:00	107.000	22.238	-0.012	2017-11-20	08:30:00
2383	2384	11/20/2017	08:45:00	107.032	22.238	-0.009	2017-11-20	08:45:00
2384	2385	11/20/2017	09:00:00	107.032	22.238	-0.009	2017-11-20	09:00:00
2385	2386	11/20/2017	09:15:00	107.012	22.142	-0.011	2017-11-20	09:15:00
2386	2387	11/20/2017	09:30:00	106.996	22.142	-0.012	2017-11-20	09:30:00
2387	2388	11/20/2017	09:45:00	107.025	22.046	-0.009	2017-11-20	09:45:00
2388	2389	11/20/2017	10:00:00	107.021	21.951	-0.010	2017-11-20	10:00:00
2389	2390	11/20/2017	10:15:00	106.986	21.855	-0.013	2017-11-20	10:15:00
2390	2391	11/20/2017	10:30:00	106.998	21.760	-0.012	2017-11-20	10:30:00
2391	2392	11/20/2017	10:45:00	106.998	21.760	-0.012	2017-11-20	10:45:00
2392	2393	11/20/2017	11:00:00	106.950	21.760	-0.017	2017-11-20	11:00:00
2393	2394	11/20/2017	11:15:00	106.966	21.760	-0.015	2017-11-20	11:15:00

	month	N	Date	CHR	Temp	SC	Sal	\
0	10	2	10/26/2017 13:00:00	54525.5	24.91	54622.1	36.1588	
1	10	3	10/26/2017 13:15:00	54525.5	24.82	54719.0	36.2311	
2	10	4	10/26/2017 13:30:00	54525.5	24.76	54783.8	36.2794	
3	10	5	10/26/2017 13:45:00	54525.5	24.75	54794.6	36.2875	
4	10	6	10/26/2017 14:00:00	54525.5	24.73	54816.2	36.3036	
5	10	7	10/26/2017 14:15:00	54525.5	24.72	54827.0	36.3117	
6	10	8	10/26/2017 14:30:00	54525.5	24.70	54848.7	36.3279	
7	10	9	10/26/2017 14:45:00	54525.5	24.69	54859.5	36.3360	
8	10	10	10/26/2017 15:00:00	54525.5	24.67	54881.2	36.3521	
9	10	11	10/26/2017 15:15:00	54525.5	24.67	54881.2	36.3521	
10	10	12	10/26/2017 15:30:00	54525.5	24.66	54892.0	36.3602	
11	10	13	10/26/2017 15:45:00	54525.5	24.64	54913.7	36.3764	
12	10	14	10/26/2017 16:00:00	54525.5	24.64	54913.7	36.3764	
13	10	15	10/26/2017 16:15:00	54525.5	24.64	54913.7	36.3764	
14	10	16	10/26/2017 16:30:00	54525.5	24.63	54924.6	36.3845	
15	10	17	10/26/2017 16:45:00	54525.5	24.62	54935.4	36.3926	
16	10	18	10/26/2017 17:00:00	54525.5	24.62	54935.4	36.3926	

17	10	19	10/26/2017	17:15:00	54525.5	24.60	54957.1	36.4089
18	10	20	10/26/2017	17:30:00	54525.5	24.59	54968.0	36.4170
19	10	21	10/26/2017	17:45:00	54525.5	24.59	54968.0	36.4170
20	10	22	10/26/2017	18:00:00	54525.5	24.59	54968.0	36.4170
21	10	23	10/26/2017	18:15:00	54525.5	24.57	54989.7	36.4332
22	10	24	10/26/2017	18:30:00	54525.5	24.56	55000.6	36.4413
23	10	25	10/26/2017	18:45:00	54525.5	24.56	55000.6	36.4413
24	10	26	10/26/2017	19:00:00	54525.5	24.54	55022.4	36.4576
25	10	27	10/26/2017	19:15:00	54525.5	24.54	55022.4	36.4576
26	10	28	10/26/2017	19:30:00	54525.5	24.53	55033.3	36.4657
27	10	29	10/26/2017	19:45:00	54525.5	24.53	55033.3	36.4657
28	10	30	10/26/2017	20:00:00	54525.5	24.50	55065.9	36.4901
29	10	31	10/26/2017	20:15:00	54525.5	24.50	55065.9	36.4901
...	...	...	...	...	...	...	...	...
2364	11	2366	11/20/2017	04:00:00	54525.5	22.64	57157.9	38.0587
2365	11	2367	11/20/2017	04:15:00	54525.5	22.67	57123.1	38.0325
2366	11	2368	11/20/2017	04:30:00	54525.5	22.70	57088.4	38.0064
2367	11	2369	11/20/2017	04:45:00	54525.5	22.72	57065.2	37.9890
2368	11	2370	11/20/2017	05:00:00	54525.5	22.72	57065.2	37.9890
2369	11	2371	11/20/2017	05:15:00	54525.5	22.70	57088.4	38.0064
2370	11	2372	11/20/2017	05:30:00	54525.5	22.72	57065.2	37.9890
2371	11	2373	11/20/2017	05:45:00	54525.5	22.72	57065.2	37.9890
2372	11	2374	11/20/2017	06:00:00	54525.5	22.70	57088.4	38.0064
2373	11	2375	11/20/2017	06:15:00	54525.5	22.67	57123.1	38.0325
2374	11	2376	11/20/2017	06:30:00	54525.5	22.64	57157.9	38.0587
2375	11	2377	11/20/2017	06:45:00	54525.5	22.62	57181.2	38.0762
2376	11	2378	11/20/2017	07:00:00	54525.5	22.56	57250.9	38.1287
2377	11	2379	11/20/2017	07:15:00	54525.5	22.51	57309.1	38.1725
2378	11	2380	11/20/2017	07:30:00	54525.5	22.48	57344.1	38.1988
2379	11	2381	11/20/2017	07:45:00	54525.5	22.45	57379.2	38.2252
2380	11	2382	11/20/2017	08:00:00	54525.5	22.41	57425.9	38.2604
2381	11	2383	11/20/2017	08:15:00	54525.5	22.38	57461.0	38.2869
2382	11	2384	11/20/2017	08:30:00	54525.5	22.34	57507.9	38.3222
2383	11	2385	11/20/2017	08:45:00	54525.5	22.29	57566.6	38.3664
2384	11	2386	11/20/2017	09:00:00	54525.5	22.28	57578.3	38.3752
2385	11	2387	11/20/2017	09:15:00	54525.5	22.22	57648.9	38.4284
2386	11	2388	11/20/2017	09:30:00	54525.5	22.21	57660.6	38.4373
2387	11	2389	11/20/2017	09:45:00	54525.5	22.16	57719.6	38.4817
2388	11	2390	11/20/2017	10:00:00	54525.5	22.13	57755.0	38.5084
2389	11	2391	11/20/2017	10:15:00	54525.5	22.12	57766.8	38.5173
2390	11	2392	11/20/2017	10:30:00	54525.5	22.09	57802.3	38.5440
2391	11	2393	11/20/2017	10:45:00	54525.5	22.08	57814.1	38.5530
2392	11	2394	11/20/2017	11:00:00	54525.5	22.08	57814.1	38.5530
2393	11	2395	11/20/2017	11:15:00	54525.5	22.06	57837.8	38.5708

	Ndt	month
0	2017-10-26 13:00:00	10
1	2017-10-26 13:15:00	10

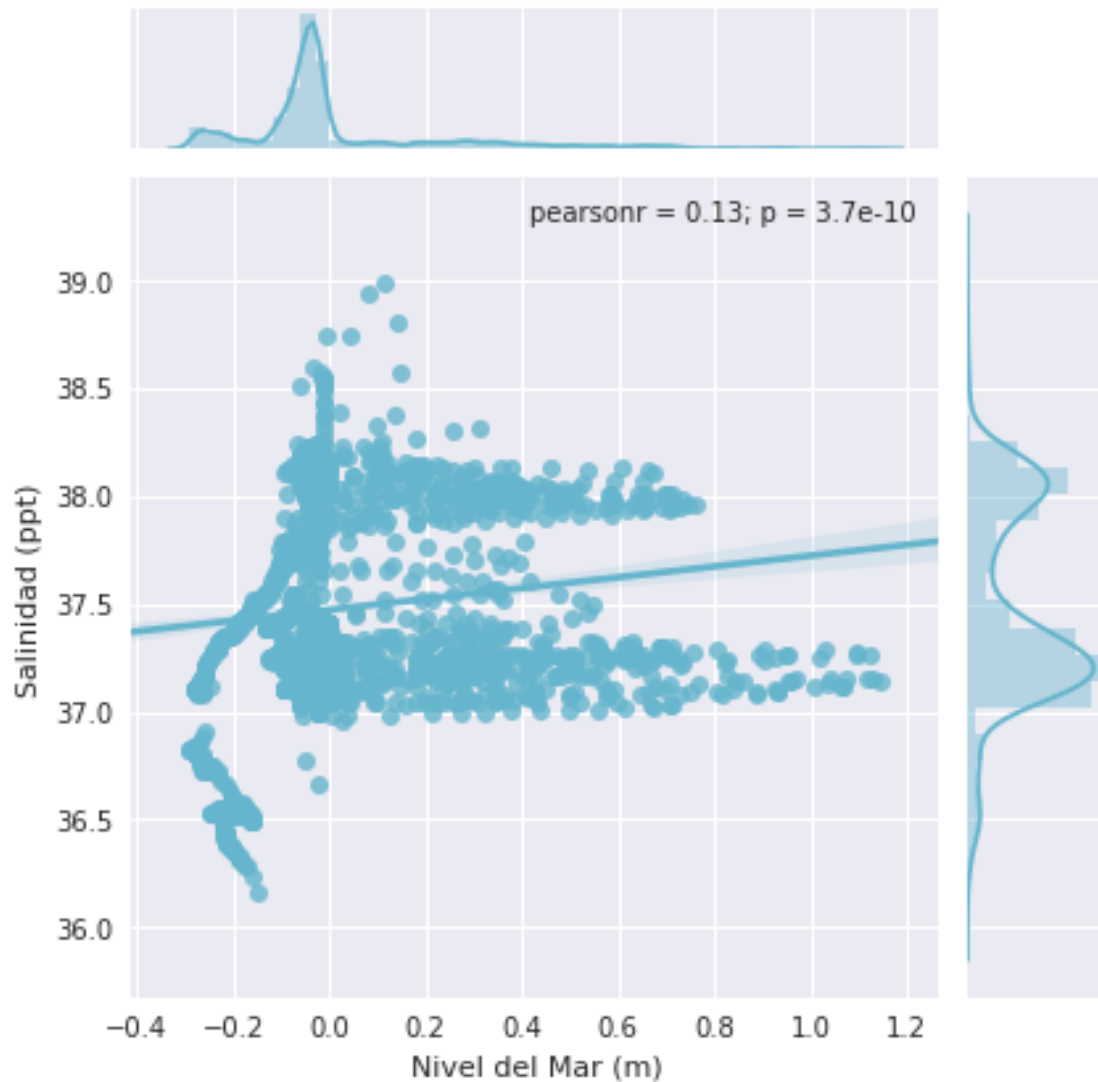


2	2017-10-26	13:30:00	10
3	2017-10-26	13:45:00	10
4	2017-10-26	14:00:00	10
5	2017-10-26	14:15:00	10
6	2017-10-26	14:30:00	10
7	2017-10-26	14:45:00	10
8	2017-10-26	15:00:00	10
9	2017-10-26	15:15:00	10
10	2017-10-26	15:30:00	10
11	2017-10-26	15:45:00	10
12	2017-10-26	16:00:00	10
13	2017-10-26	16:15:00	10
14	2017-10-26	16:30:00	10
15	2017-10-26	16:45:00	10
16	2017-10-26	17:00:00	10
17	2017-10-26	17:15:00	10
18	2017-10-26	17:30:00	10
19	2017-10-26	17:45:00	10
20	2017-10-26	18:00:00	10
21	2017-10-26	18:15:00	10
22	2017-10-26	18:30:00	10
23	2017-10-26	18:45:00	10
24	2017-10-26	19:00:00	10
25	2017-10-26	19:15:00	10
26	2017-10-26	19:30:00	10
27	2017-10-26	19:45:00	10
28	2017-10-26	20:00:00	10
29	2017-10-26	20:15:00	10
...	...	...	...
2364	2017-11-20	04:00:00	11
2365	2017-11-20	04:15:00	11
2366	2017-11-20	04:30:00	11
2367	2017-11-20	04:45:00	11
2368	2017-11-20	05:00:00	11
2369	2017-11-20	05:15:00	11
2370	2017-11-20	05:30:00	11
2371	2017-11-20	05:45:00	11
2372	2017-11-20	06:00:00	11
2373	2017-11-20	06:15:00	11
2374	2017-11-20	06:30:00	11
2375	2017-11-20	06:45:00	11
2376	2017-11-20	07:00:00	11
2377	2017-11-20	07:15:00	11
2378	2017-11-20	07:30:00	11
2379	2017-11-20	07:45:00	11
2380	2017-11-20	08:00:00	11
2381	2017-11-20	08:15:00	11
2382	2017-11-20	08:30:00	11

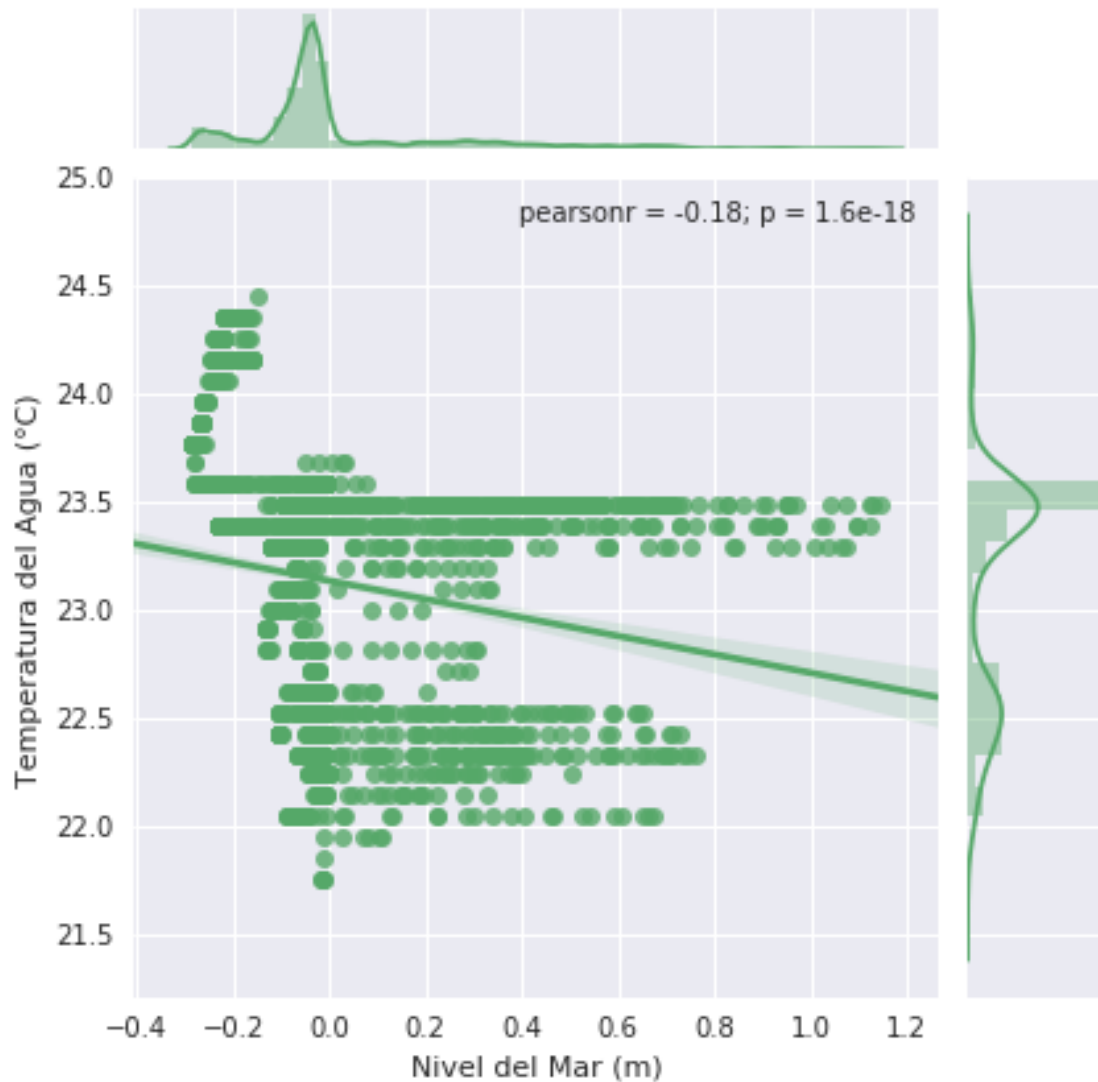
2383	2017-11-20	08:45:00	11
2384	2017-11-20	09:00:00	11
2385	2017-11-20	09:15:00	11
2386	2017-11-20	09:30:00	11
2387	2017-11-20	09:45:00	11
2388	2017-11-20	10:00:00	11
2389	2017-11-20	10:15:00	11
2390	2017-11-20	10:30:00	11
2391	2017-11-20	10:45:00	11
2392	2017-11-20	11:00:00	11
2393	2017-11-20	11:15:00	11

[2394 rows x 15 columns]

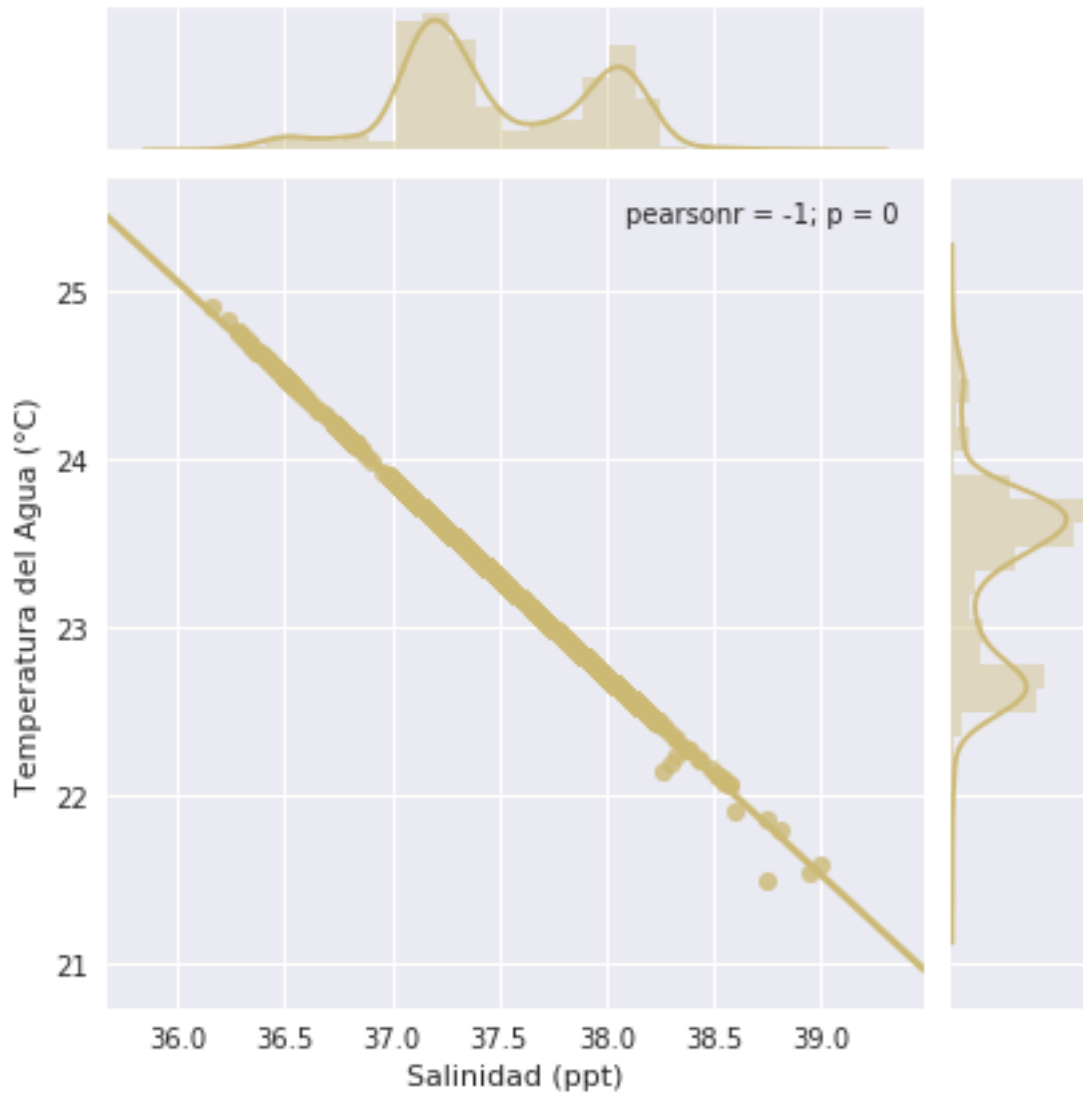
```
In [15]: # Gráfica de Nivel de mar-Salinidad
sns.set(style="darkgrid", color_codes=True)
df3=pd.concat([df1, df2], axis=1, join_axes=[df2.index])
g = sns.jointplot("WL", "Sal", data=df3,kind="reg", color="c")
#plt.title('Nivel de mar-Salinidad')
plt.ylabel('Salinidad (ppt)')
plt.xlabel('Nivel del Mar (m)')
plt.show(g)
```



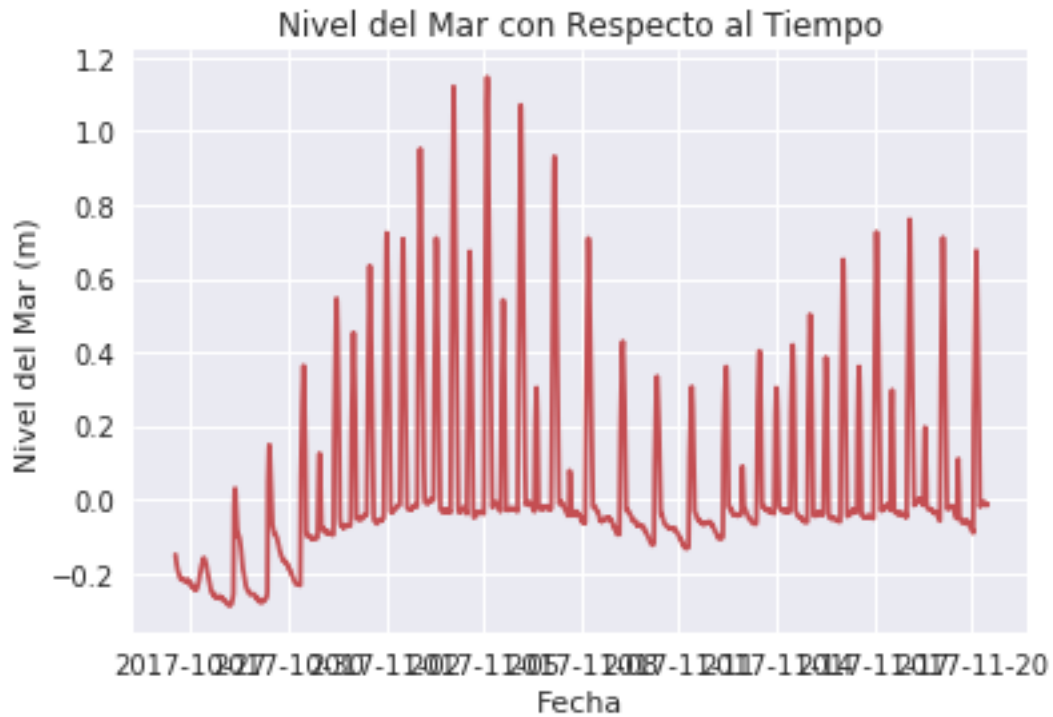
```
In [16]: # Gráfica de Nivel de mar-Temperatura del agua
sns.set(style="darkgrid", color_codes=True)
g = sns.jointplot("WL", "Temp", data=df1, kind="reg", color="g")
#plt.title('Nivel de mar-Temperatura del Agua')
plt.ylabel('Temperatura del Agua (°C)')
plt.xlabel('Nivel del Mar (m)')
plt.show(g)
```



```
In [17]: # Gráfica de Salinidad-Temperatura del agua
sns.set(style="darkgrid", color_codes=True)
g = sns.jointplot("Sal", "Temp", data=df2, kind="reg", color="y")
#plt.title('Salinidad-Temperatura del agua')
plt.ylabel('Temperatura del Agua (°C)')
plt.xlabel('Salinidad (ppt)')
plt.show(g)
```



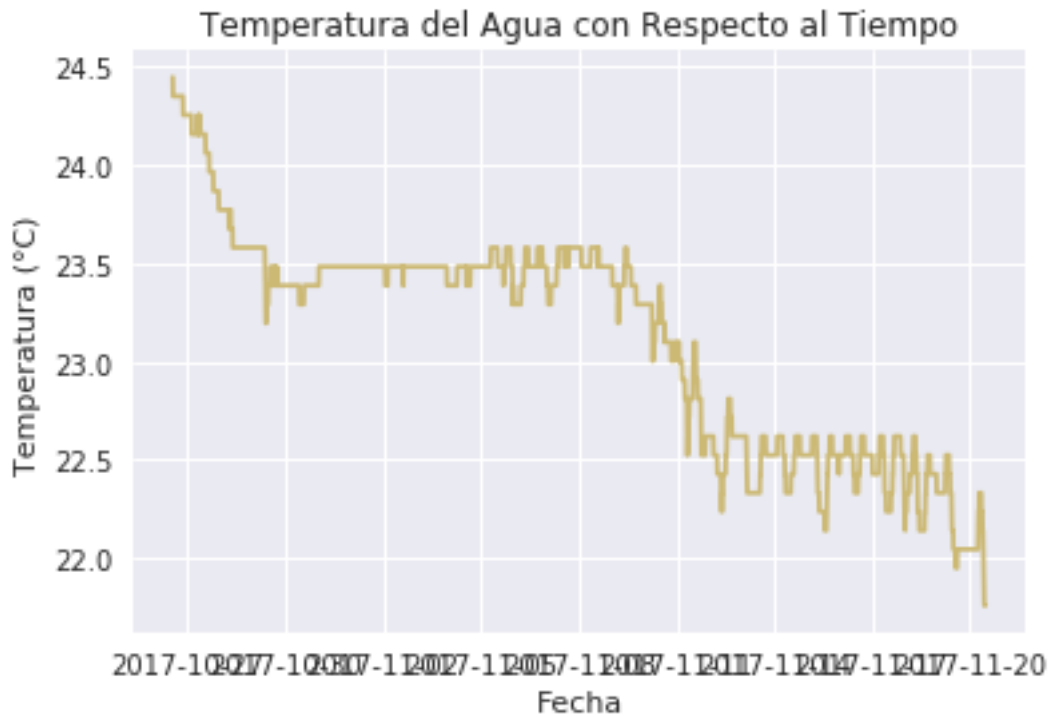
```
In [18]: # Gráfica de Nivel del mar como función del tiempo
WL = df1['WL']
Date = df1['Ndt']
plt.plot_date(x=Date, y=WL, fmt='r-')
plt.title('Nivel del Mar con Respecto al Tiempo')
plt.ylabel('Nivel del Mar (m)')
plt.xlabel('Fecha')
plt.grid(True)
plt.show()
```



```
In [19]: # Gráfica de Salinidad como función del tiempo
Sal = df2['Sal']
Date = df2['Ndt']
plt.plot_date(x=Date, y=Sal, fmt='g-')
plt.title('Salinidad con Respecto al Tiempo')
plt.ylabel('Salinidad (ppt)')
plt.xlabel('Fecha')
plt.grid(True)
plt.show()
```

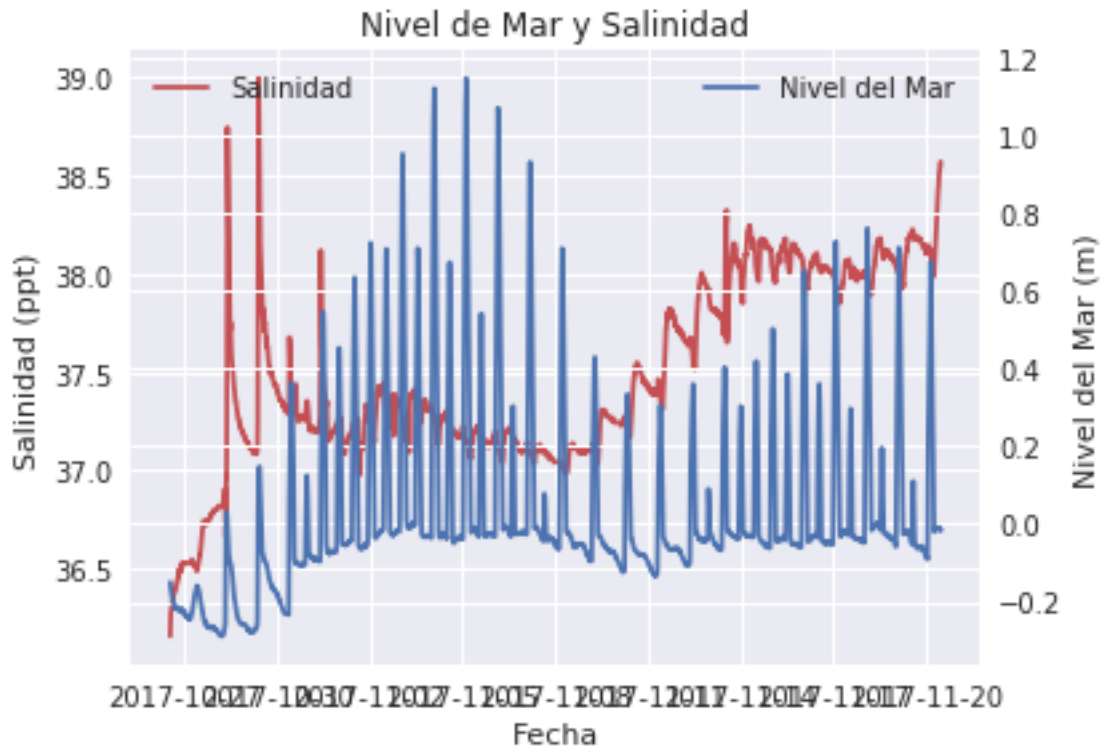


```
In [20]: # Gráfica de Temperatura del Agua como función del tiempo
Temp = df1['Temp']
Date = df1['Ndt']
plt.plot_date(x=Date, y=Temp, fmt='y-')
plt.title('Temperatura del Agua con Respecto al Tiempo')
plt.ylabel('Temperatura (°C)')
plt.xlabel('Fecha')
plt.grid(True)
plt.show()
```

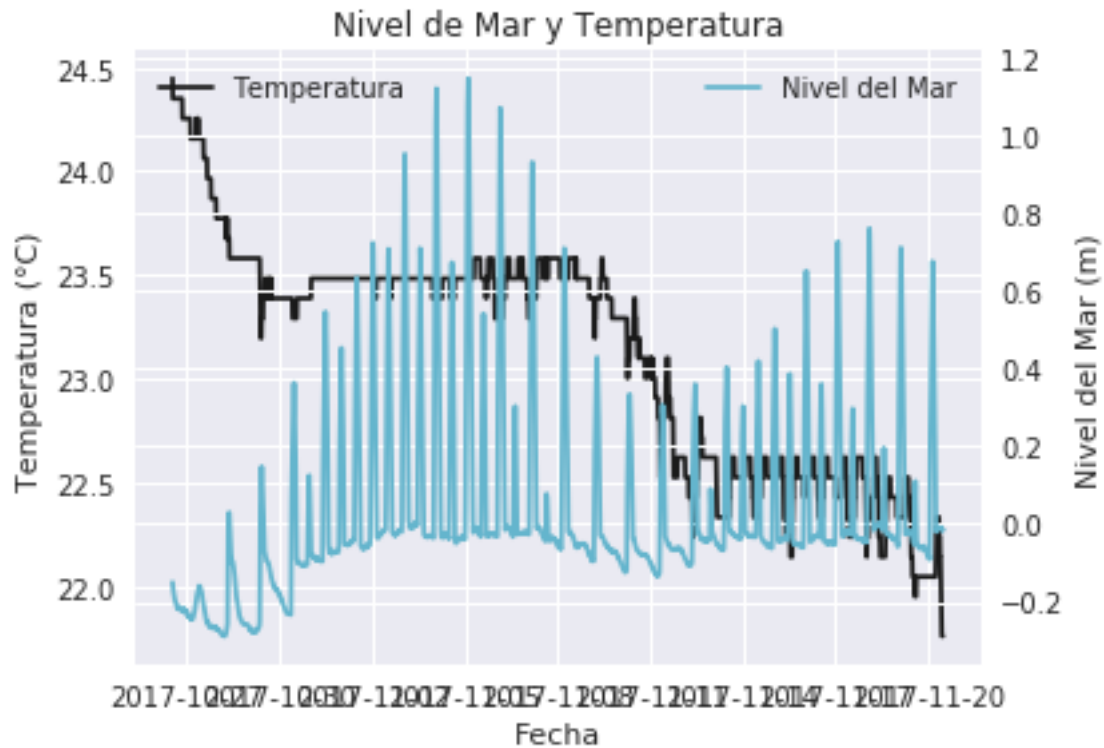


```
In [21]: # Gráfica de Nivel de mar y Salinidad
fig, ax1 = plt.subplots()
Fecha=df1['Ndt']
sal=df2.Sal
WL=df1.WL
ax1.plot(Fecha,sal,'r-', label='Salinidad'); plt.legend(loc='upper left')
ax1.set_xlabel('Fecha')
ax1.set_ylabel('Salinidad (ppt)')
ax2 = ax1.twinx()
ax2.plot(Fecha, WL , 'b-', label='Nivel del Mar'); plt.legend(loc='best')
ax2.set_ylabel('Nivel del Mar (m)')
fig.tight_layout()
plt.title('Nivel de Mar y Salinidad')
plt.show()
```

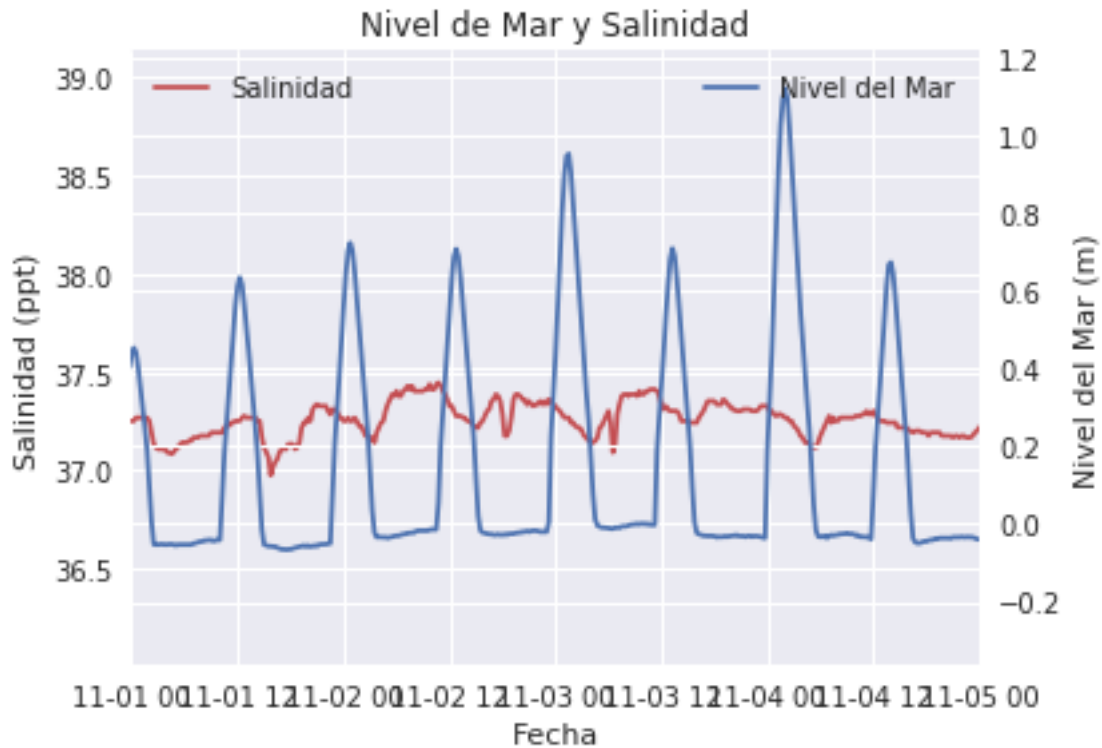




```
In [22]: # Gráfica de Nivel de mar y Temperatura
fig, ax1 = plt.subplots()
Fecha=df1['Ndt']
Temp=df1.Temp
WL=df1.WL
ax1.plot(Fecha,Temp,'k-', label='Temperatura'); plt.legend(loc='upper left')
ax1.set_xlabel('Fecha')
ax1.set_ylabel('Temperatura (°C)')
ax2 = ax1.twinx()
ax2.plot(Fecha, WL , 'c-', label='Nivel del Mar'); plt.legend(loc='best')
ax2.set_ylabel('Nivel del Mar (m)')
fig.tight_layout()
plt.title('Nivel de Mar y Temperatura')
plt.show()
```



```
In [23]: fig, ax1 = plt.subplots()
Fecha=df1['Ndt']
sal=df2.Sal
WL=df1.WL
ax1.plot(Fecha,sal,'r-', label='Salinidad'); plt.legend(loc='upper left')
ax1.set_xlabel('Fecha')
ax1.set_ylabel('Salinidad (ppt)')
ax2 = ax1.twinx()
ax2.plot(Fecha, WL , 'b-', label='Nivel del Mar'); plt.legend(loc='upper right')
ax2.set_ylabel('Nivel del Mar (m)')
fig.tight_layout()
plt.xlim(("2017-11-1 00:00:00","2017-11-5 00:00:00"))
plt.title('Nivel de Mar y Salinidad')
plt.show()
```



```
In [24]: # Gráfica de Nivel de Mar y Temperatura del agua
fig, ax1 = plt.subplots()
Fecha=df1['Ndt']
Temp=df1.Temp
WL=df1.WL
ax1.plot(Fecha,Temp,'k-', label='Temperatura'); plt.legend(loc='upper left')
ax1.set_xlabel('Fecha')
ax1.set_ylabel('Temperatura (°C)')
ax2 = ax1.twinx()
ax2.plot(Fecha, WL , 'c-', label='Nivel del Mar'); plt.legend(loc='upper right')
ax2.set_ylabel('Nivel del Mar (m)')
fig.tight_layout()
plt.xlim(("2017-11-1 00:00:00","2017-11-5 00:00:00"))
plt.title('Nivel de Mar y Temperatura del Agua')
plt.show()
```

