

# Evaluación 1

Roberto Benard Orci

08/03/2018

## Actividad a realizar

### Act.1

Para este paso use los comandos *less* y *cat* para ver los primeros datos de los archivos, de esta manera al abrir jupyter notebook en la terminal y al hacer los dataframes me asegure de eliminar los renglones que no me servían (quería que me quedaran el mismo número de datos con la misma fecha y hora), como lo puede ver en la siguiente imagen:

```
dfs = pd.read_csv("sargento-salinidad-201117.csv", header=None, skiprows=3, names=['FECH', 'COND', 'TEMP', 'SPCD', 'SALY'])
dfN = pd.read_csv("sargento_201117.csv", header=None, skiprows=2, names=['FECH', 'PRES', 'TEMP', 'WL'])

dfN=dfN.drop(dfN.index[len(dfN)-1])

dfs=dfs.reset_index()
dfN=dfN.reset_index()

dfs=dfs.drop('index', axis=1)
dfN=dfN.drop('index', axis=1)
```

### Act.2

Antes de eliminar los renglones que no necesitaría, importe las bibliotecas de *Numpy*, *Pandas*, *Seaborn*, *matplotlib.pyplot* y *datetime*.

```
import pandas as pd
import numpy as np
from datetime import datetime
import seaborn as sns
import matplotlib.pyplot as plt
```

Después de importar las bibliotecas y de eliminar los renglones inservibles, utilice el comando *DataFrame* para darles estructura a las tablas de datos. Luego me asegure que estos fueran del "tipo" indicado.

```
dfN=pd.DataFrame(dfN)
dfs=pd.DataFrame(dfs)
```

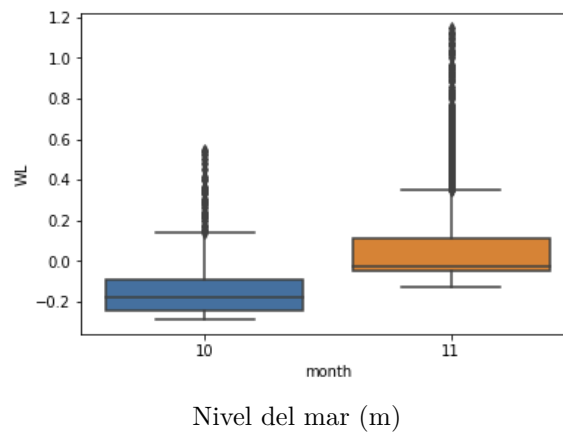
```
dfs.dtypes
dfN.dtypes
```

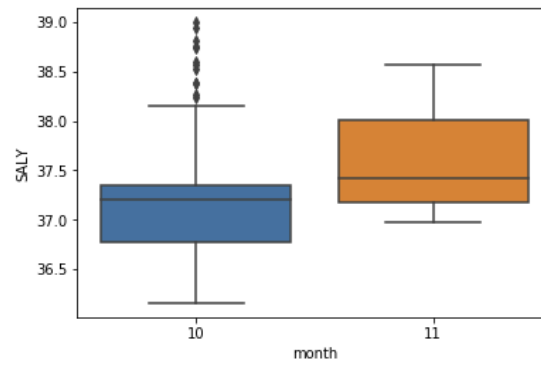
```
FECH      object
PRES      float64
TEMP      float64
WL         float64
Ndate     datetime64[ns]
month      int64
dtype: object
```

### Act.3

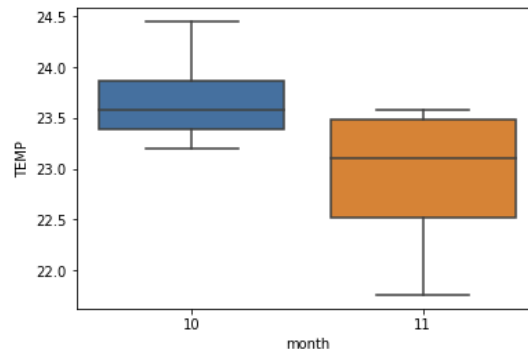
Con la ayuda de la biblioteca *Seaborn* cree diferentes graficas de caja (Boxplot) para diferentes variables.

Nota: en muchos casos se utilizaron dos temperaturas, una temperatura (Temperatura1) es la que aparece en el archivo llamado sargento-201117.csv, en el cual aparece la variable del nivel del mar. La otra (Temperatura2) aparece en el archivo llamado sargento-salinidad-201117.csv, en el cual aparece la variable de salinidad.

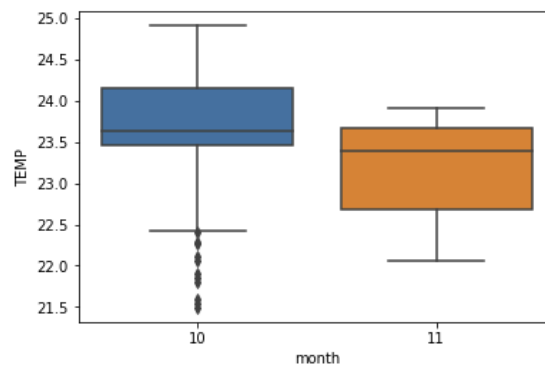




Salinidad (ppt)



Temperatura1 (C°)



Temperatura2 (C°)

Después de graficar, utiliza el comando *describe* para obtener la mediana, media, los cuartiles, el máximo y el mínimo de los datos.

	PRES	TEMP	WL	month
<b>count</b>	2394.000000	2394.000000	2394.000000	2394.000000
<b>mean</b>	107.430007	23.120883	0.030863	10.781119
<b>std</b>	2.371844	0.563555	0.235974	0.413574
<b>min</b>	104.229000	21.760000	-0.288000	10.000000
<b>25%</b>	106.407000	22.525000	-0.071000	11.000000
<b>50%</b>	106.764000	23.388000	-0.035000	11.000000
<b>75%</b>	107.305000	23.484000	0.018750	11.000000
<b>max</b>	118.641000	24.448000	1.146000	11.000000

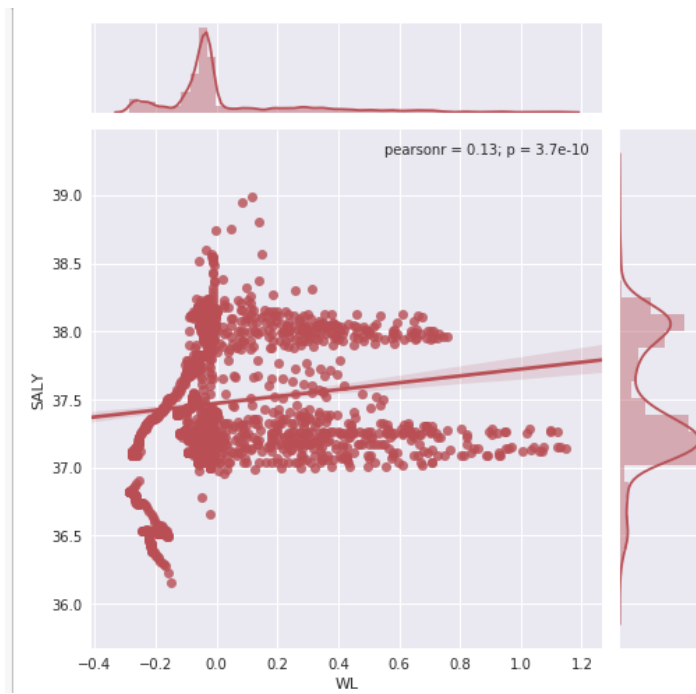
sargento-201117.csv

	COND	TEMP	SPCD	SALY	month
<b>count</b>	2394.000000	2394.000000	2394.000000	2394.000000	2394.000000
<b>mean</b>	54524.972807	23.316646	56386.831662	37.479737	10.781119
<b>std</b>	11.876669	0.547033	619.501987	0.464974	0.413574
<b>min</b>	54105.700000	21.490000	54622.100000	36.158800	10.000000
<b>25%</b>	54525.500000	22.730000	55949.700000	37.151400	11.000000
<b>50%</b>	54525.500000	23.490000	56185.600000	37.328300	11.000000
<b>75%</b>	54525.500000	23.700000	57053.700000	37.980300	11.000000
<b>max</b>	54525.500000	24.910000	58398.700000	38.994200	11.000000

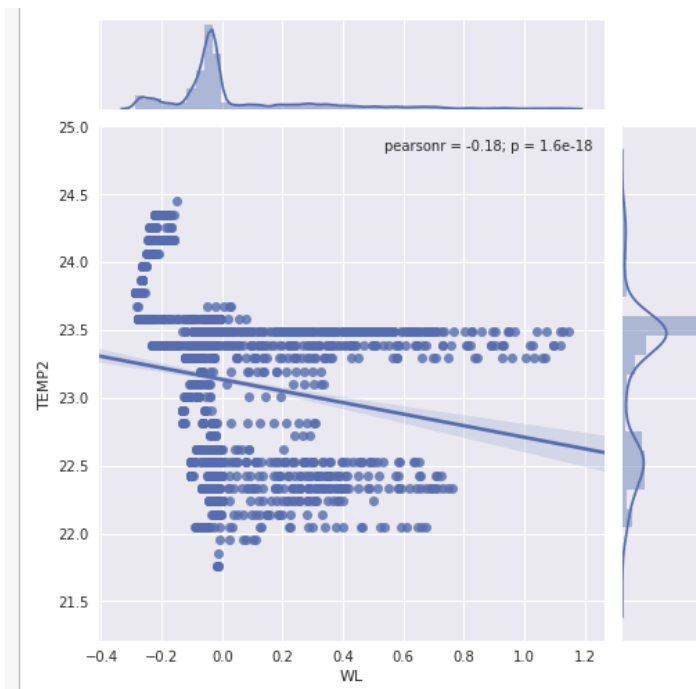
sargento-salinidad-201117.csv

## Act.4

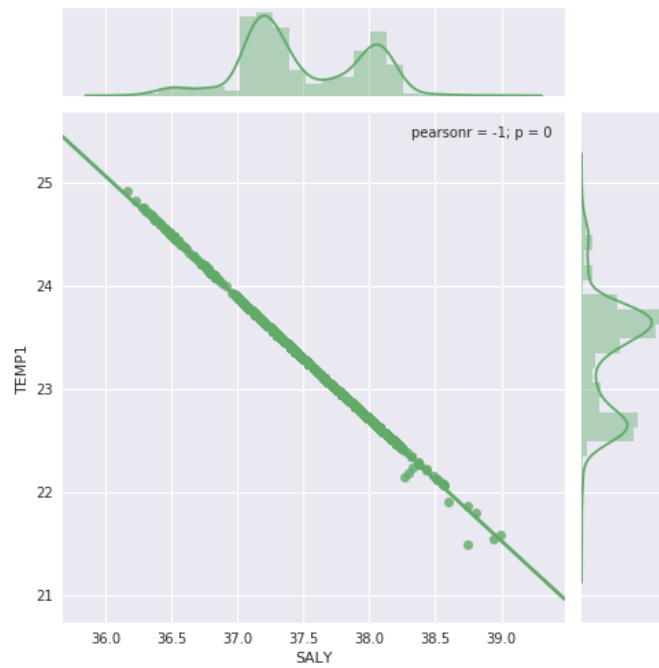
Para esta parte, volvemos a utilizar *Seaborn* para graficar los datos, solo que esta vez creamos una gráfica de regresión lineal con distribuciones marginales.



WL = Nivel del mar, SALY = Salinidad



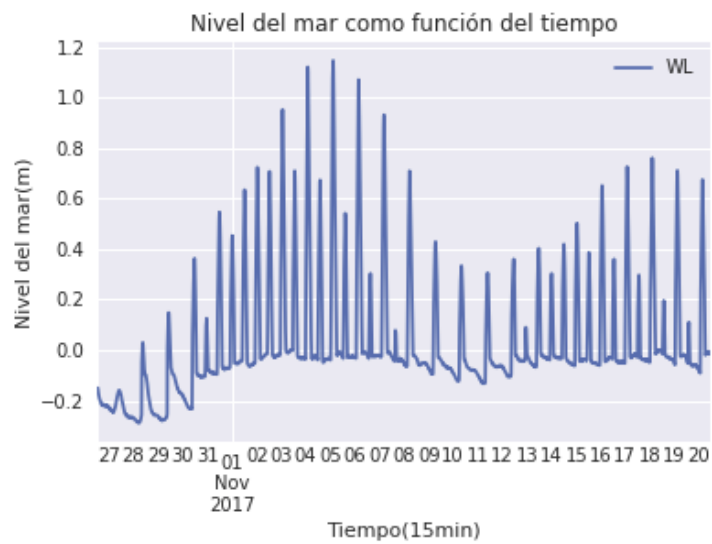
WL = Nivel del mar, TEMP = Temperatura

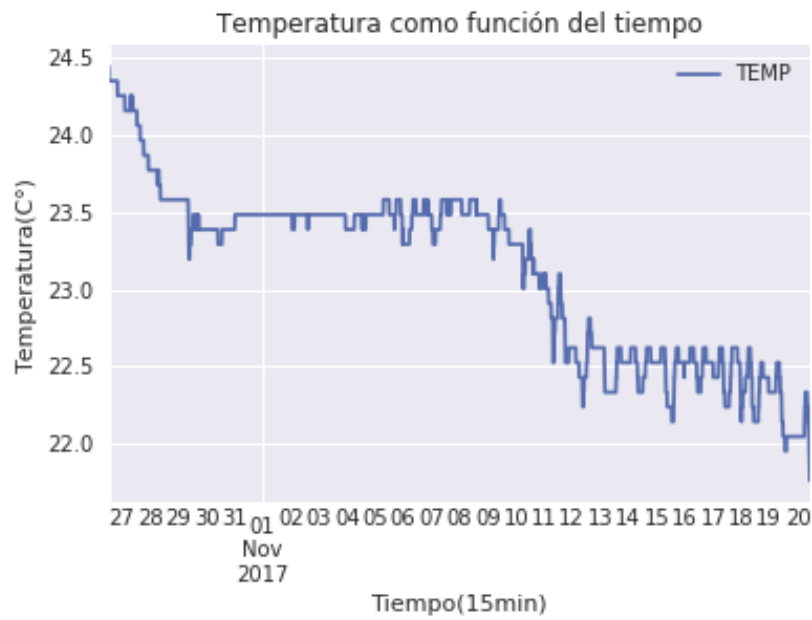
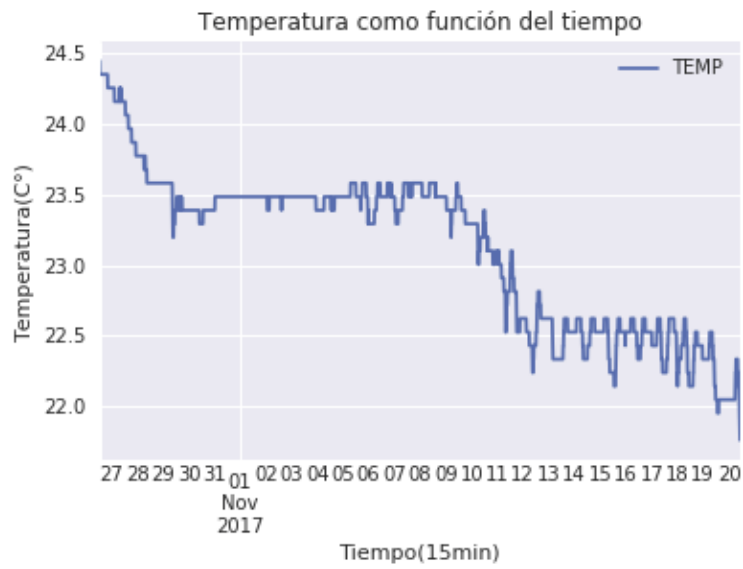


SALY = Salinidad , TEMP = Temperatura2

## Act.5

En esta parte utilizamos *Matplotlib* para hacer tres graficas independientes de ciertas variables como función del tiempo.

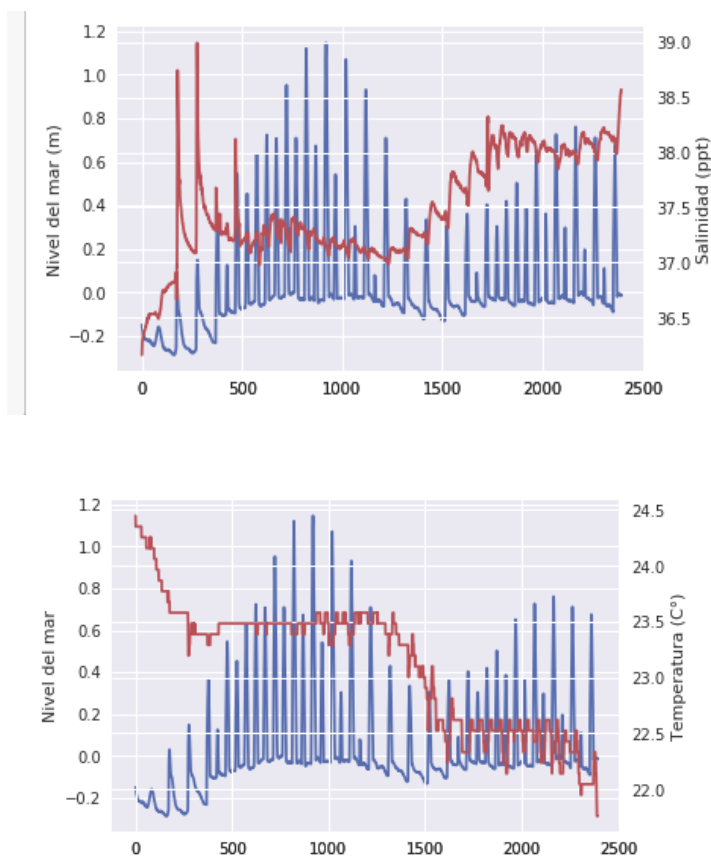






## Act.6

Esta parte es muy parecida a la parte anterior, solo que aquí hicimos graficas superpuestas con doble eje vertical. Estas graficas son muy útiles si quieres comparar dos variables y estas variables tienen rangos diferentes o una es simplemente mucho mayor o menor que la otra para poderlas comparar una encima de la otra de manera "normal".



Estas graficas también están en función del tiempo. En total hay 2394 datos, cada unidad en el eje x representa 15 minutos empezando del 26 de octubre a las 13:00 hasta las 11:15 del 20 de noviembre.

## Act.7

En esta actividad utilizamos las gráficas de la sección anterior, solo que tomamos un pequeño intervalo de tiempo de 5 días (480 unidades) para visualizar mejor la relación entre estas variables en función del tiempo.

Como se puede ver en el código, lo único que se le agrego a este fue `plt.xlim([1000,1480])` para tomar datos que estuvieran prácticamente a la mitad de la lista.

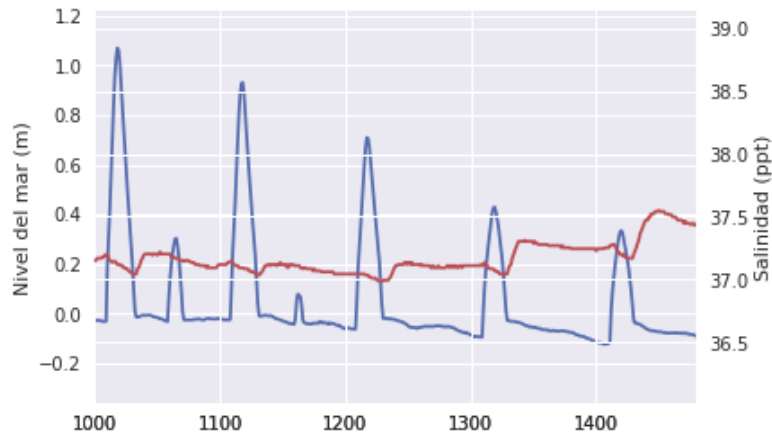
```
from pylab import figure, show, legend, ylabel

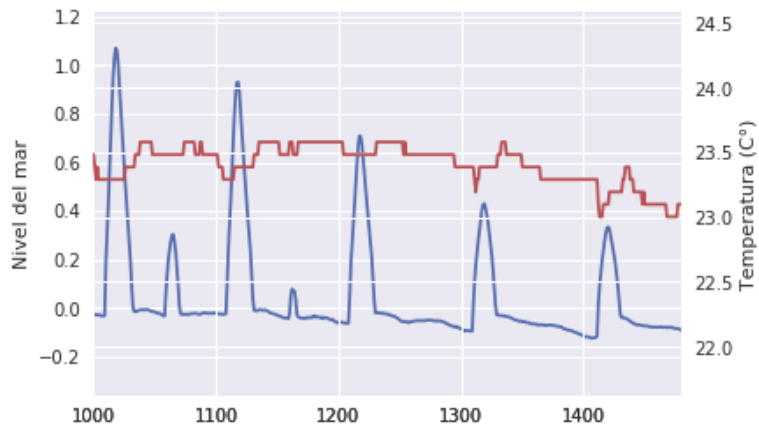
# create the general figure
fig1 = figure()

# and the first axes using subplot populated with data
ax1 = fig1.add_subplot(111)
line1 = ax1.plot(GRAF['WL'], 'b-')
ylabel("Nivel del mar (m)")

# now, the second axes that shares the x-axis with the ax1
ax2 = fig1.add_subplot(111, sharex=ax1, frameon=False)
line2 = ax2.plot(GRAF['SALY'], 'xr-')
ax2.yaxis.tick_right()
ax2.yaxis.set_label_position("right")
ylabel("Salinidad (ppt)")

plt.xlim([1000,1480])
show()
```





Analizando las gráficas uno puede observar que hay una ligera relación entre la temperatura y la salinidad con el nivel del mar. Cuando el nivel del mar aumenta de manera considerable, la salinidad y la temperatura presentan una pequeña disminución.