

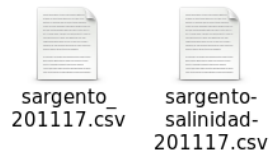
Evaluación 1: Análisis de las mareas y salinidad en el Manglar El Sargento.

Melissa Matrecitos Avila

8 de Marzo de 2018

1 Preparación de datos

Para comenzar se descargaron dos archivos que contenían los datos a estudiar, los documentos fueron:



Los cuales tienen la siguiente estructura, el primero corresponde al archivo con nombre sargento_201117.csv y el segundo a sargentosalinidad201117.csv:

```
1 "Plot Title: sargento"
2 "#","Date Time, GMT-07:00","Abs Pres, kPa","Temp, °C","Water Level, meters"
3 1,10/26/2017 13:00:00,105.612,24.448,-0.150
4 2,10/26/2017 13:15:00,105.513,24.351,-0.160
5 3,10/26/2017 13:30:00,105.433,24.351,-0.168
6 4,10/26/2017 13:45:00,105.385,24.351,-0.173

1 "Plot Title: sargento-salinidad"
2 "#","Date Time, GMT-07:00","Cond High Rng, ?S/cm","Temp, °C","Specific Conductance, ?S/cm","Salinity, ppt"
3 1,10/26/2017 12:45:00,54525.5,25.21,54301.2,35.9195
4 2,10/26/2017 13:00:00,54525.5,24.91,54622.1,36.1588
5 3,10/26/2017 13:15:00,54525.5,24.82,54719.0,36.2311
6 4,10/26/2017 13:30:00,54525.5,24.76,54783.8,36.2794
```

Para poder trabajar con ellos en pandas, se les cambió el nombre, eliminando los guiones o cambiándolos por guión bajo. Procediendo en el entorno de jupyter notebook, lo primero que se hizo fue importar las bibliotecas que se utilizarían más adelante:

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

Posteriormente se leyeron los dos archivos que contenían los datos, saltando los dos primeros renglones ya que no contenían datos, además se la asignaron los nombres a las columnas de cada archivo:

```
#Leer el archivo Sargento
dfSargento=pd.read_csv("sargento201117.csv", skiprows=2, header=None, names=['Num', 'Date', 'Abs_Pres', 'Temp', 'Water_Level'])

#Leer el archivo Salinidad
dfSalinidad=pd.read_csv("sargento_salinidad_201117.csv", skiprows=3, header=None, names=['Num', 'Date', 'Cond_High_Rng', 'Temp', 'Specific_Conductance', 'Salinity'])
```

Continuando con la preparación de los datos, se imprimió en la pantalla las dos cabezas y finales de ambas tablas, con el fin de verificar que ambas tuvieran el mismo número de datos, cómo no fue así, se eliminaron los datos extras en el archivo de nombre "sargento", logrando que los dos archivos constaran de la misma cantidad. La siguiente imagen muestra el final de la tabla Sargento (los últimos 5 datos), donde se muestra la cantidad de datos que ésta tiene:

	Num	Date	Abs_Pres	Temp	Water_Level
2389	2390	11/20/2017 10:15:00	106.986	21.855	-0.013
2390	2391	11/20/2017 10:30:00	106.998	21.760	-0.012
2391	2392	11/20/2017 10:45:00	106.998	21.760	-0.012
2392	2393	11/20/2017 11:00:00	106.950	21.760	-0.017
2393	2394	11/20/2017 11:15:00	106.966	21.760	-0.015

Por último, se creo para cada archivo, dos columnas nuevas, de tipo "date", que contenían la fecha y el mes en el que se tomarón los datos, esto con el fin de poder trabajar con los datos en las bibliotecas mencionadas anteriormente:

```
#Cambiar el tipo de date
dfSargento['Ndate'] = pd.to_datetime(dfSargento['Date'], format='%m/%d/%Y %H:%M:%S')
dfSargento['month'] = dfSargento['Ndate'].dt.month
```

La imagen anterior muestra la instrucción para realizar las acciones mencionadas, solo se muestra para el archivo "sargento", ya que fue lo mismo que se realizó para el archivo "salinidad". El formato final de los datos , pra el archivo sargento y salinidad respectivamente, fue:

	Num	Date	Abs_Pres	Temp	Water_Level	Ndate	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

	Num	Date	Cond_High_Rng	Temp	Specific_Conductance	Salinity	Ndate	month
0	2	10/26/2017 13:00:00	54525.5	24.91	54622.1	36.1588	2017-10-26 13:00:00	10
1	3	10/26/2017 13:15:00	54525.5	24.82	54719.0	36.2311	2017-10-26 13:15:00	10
2	4	10/26/2017 13:30:00	54525.5	24.76	54783.8	36.2794	2017-10-26 13:30:00	10
3	5	10/26/2017 13:45:00	54525.5	24.75	54794.6	36.2875	2017-10-26 13:45:00	10
4	6	10/26/2017 14:00:00	54525.5	24.73	54816.2	36.3036	2017-10-26 14:00:00	10

2 Análisis y Resultados de los Datos

Para conocer el comportamiento de las variables a estudiar, se realizaron distintos diagramas que nos proporcionan información muy valiosa. Para los diagramas de caja y Correlación de Pearson se utilizó la biblioteca de Seaborn, mientras que para las demás se usó la biblioteca Matplotlib. la cual nos da más libertad para manejar los datos, siendo las gráficas supuestas las más complejas. A continuación se muestra el código que se utilizó para crear cada uno de los diagramas y su diagrama correspondiente, siendo el primer código que aparece el creador del primer diagrama:

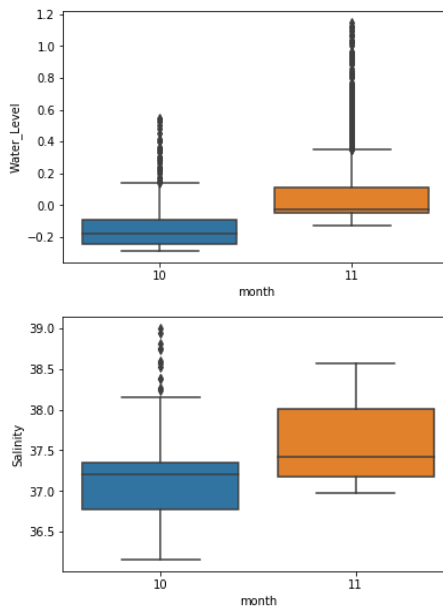
- **Diagrama de caja:**

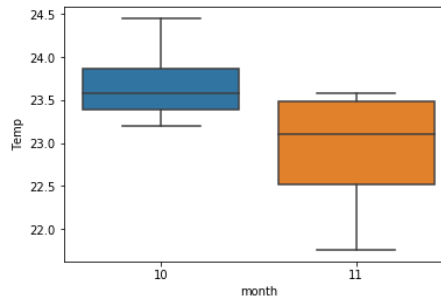
Para el estudio de cada variable se crearon dos cajas, la cuales corresponden al mes de octubre y noviembre. Las variables a estudiar fueron nivel del mar, salinidad y temperatura, todas contra tiempo.

```
#Diagrama de caja Nivel de mar (metros)
ax = sns.boxplot(x="month", y="Water_Level", data=dfSargento)
plt.show()

#Diagrama de caja Salinidad (ppt)
ax = sns.boxplot(x="month", y="Salinity", data=dfSalinidad)
plt.show()

#Diagrama de caja Temperatura de Agua (°C).
ax = sns.boxplot(x="month", y="Temp", data=dfSargento)
plt.show()
```





- **Describe:**

Este comando se utilizo como un apoyo para loa diagramas de caja, dando a conocer valores importantes más exactos:

	Num	Abs_Pres	Temp	Water_Level	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

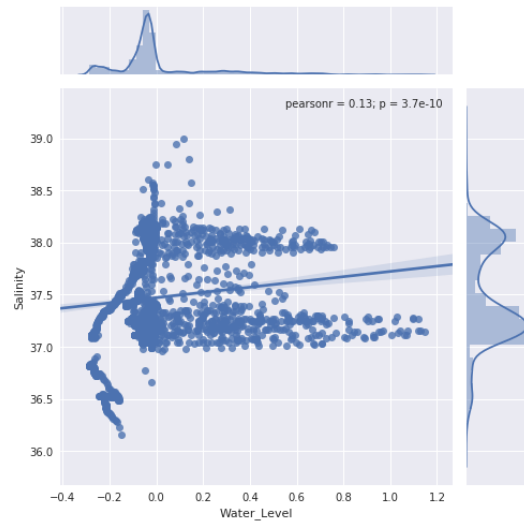
	Num	Cond_High_Rng	Temp	Specific_Conductance	Salinity	month
count	2394.000000	2394.000000	2394.000000	2394.000000	2394.000000	2394.000000
mean	1198.500000	54524.972807	23.316646	56386.831662	37.479737	10.781119
std	691.232595	11.876669	0.547033	619.501987	0.464974	0.413574
min	2.000000	54105.700000	21.490000	54622.100000	36.158800	10.000000
25%	600.250000	54525.500000	22.730000	55949.700000	37.151400	11.000000
50%	1198.500000	54525.500000	23.490000	56185.600000	37.328300	11.000000
75%	1796.750000	54525.500000	23.700000	57053.700000	37.980300	11.000000
max	2395.000000	54525.500000	24.910000	58398.700000	38.994200	11.000000

- **Correlación Pearson:**

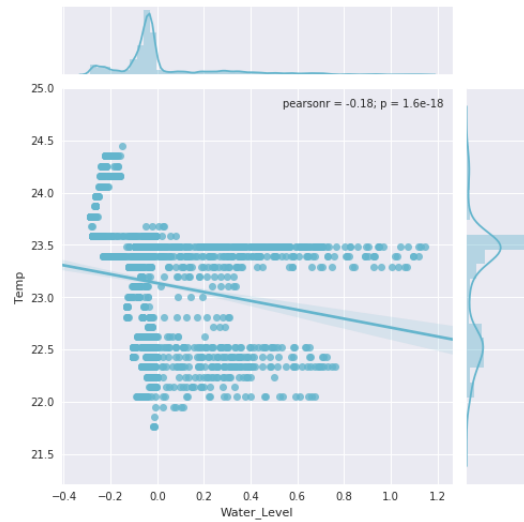
Para llevar a cabo la primera gráfica fue necesario guardar los datos de interés un nuevo Data Frame donde se combinaran los datos de Sargento y Salinidad. En la primera gráfica se muestra la relación de entre nivel de mar y salinidad, en la segunda nivel de mar y temperatura del agua y en la última salinidad y temperatura del agua:

```
#Crear dataframe para graficar las correlaciones
dfCombinados=pd.concat([dfSargento,dfSalinidad], axis=1, join_axes=[dfSalinidad.index])

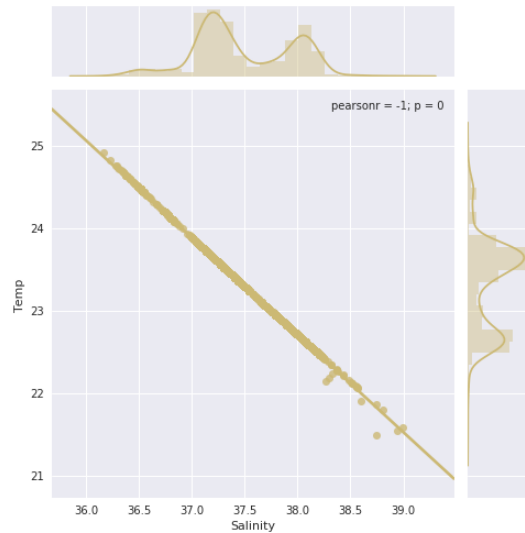
#Grafica de Nivel del Agua vs Salinidad
sns.set(style="darkgrid", color_codes=True)
g = sns.jointplot("Water_Level", "Salinity", data=dfCombinados, kind="reg",
                  color="b", size=7)
plt.show(g)
```



```
#Grafica de Nivel del Agua vs Temperatura
sns.set(style="darkgrid", color_codes=True)
g = sns.jointplot("Water_Level", "Temp", data=dfSargento, kind="reg", color="c", size=7)
plt.show(g)
```



```
#Grafica de Nivel del Agua vs Temperatura
sns.set(style="darkgrid", color_codes=True)
g = sns.jointplot("Salinity", "Temp", data=dfSalinidad, kind="reg", color="y", size=7)
plt.show(g)
```



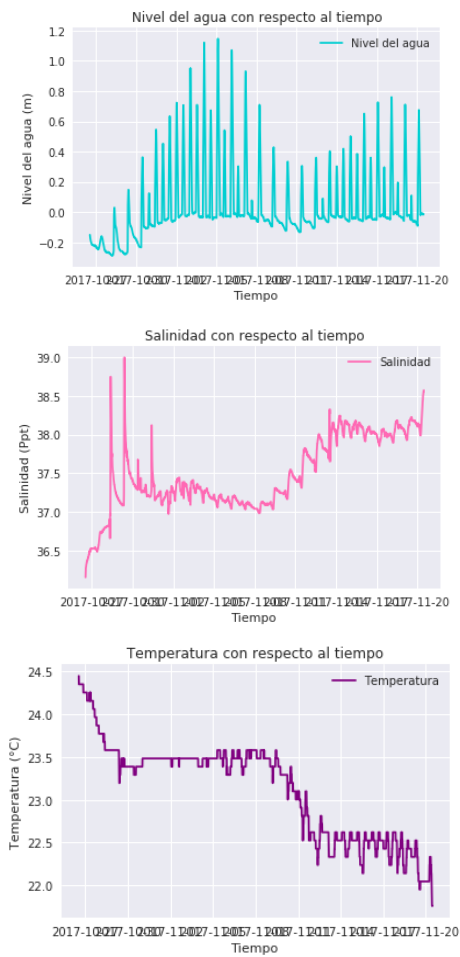
- **Gráficas Independientes:**

En esta actividad se realizaron 3 gráficas independientes de las variables nivel del mar, salinidad y temperatura del agua (respectivamente), para ver como cambiaban en función del tiempo.

```
#Gráfica de Nivel del mar contra tiempo
WL=dfSargento['Water_Level']
Tiempo=dfSargento['Ndate']
plt.plot(Tiempo, WL, "darkturquoise", label='Nivel del agua'); plt.legend(loc='best')
plt.title('Nivel del agua con respecto al tiempo')
plt.ylabel('Nivel del agua (m)')
plt.xlabel('Tiempo')
plt.grid(True)
plt.show()

#Gráfica de Salinidad contra tiempo
Sal=dfSalinidad['Salinity']
Tiempo=dfSalinidad['Ndate']
plt.plot(Tiempo, Sal, "hotpink", label='Salinidad'); plt.legend(loc='best')
plt.title('Salinidad con respecto al tiempo')
plt.ylabel('Salinidad (Ppt)')
plt.xlabel('Tiempo')
plt.grid(True)
plt.show()

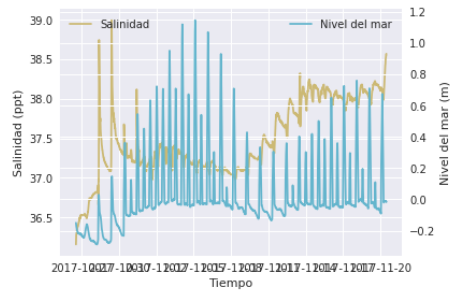
#Gráfica de Temperatura del agua contra tiempo
Temp=dfSargento['Temp']
Tiempo=dfSargento['Ndate']
plt.plot(Tiempo, Temp, "purple", label='Temperatura'); plt.legend(loc='best')
plt.title('Temperatura con respecto al tiempo')
plt.ylabel('Temperatura (°C)')
plt.xlabel('Tiempo')
plt.grid(True)
plt.show()
```



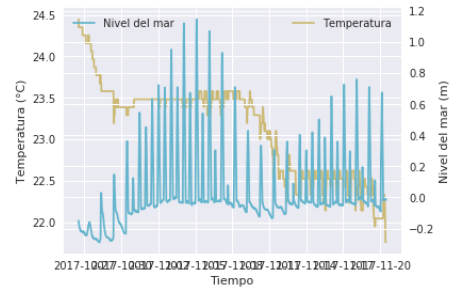
- **Gráficas supuestas:**

Las gráficas realizadas en esta sección muestran el comportamiento de dos variables con respecto a una misma, las variables a estudiar por gráfica fueron: nivel de mar y salinidad y en la otra nivel de mar y temperatura. Este tipo de gráficas no son las convencionales de Matplotlib, por lo que tienen una estructura distinta, tal como se muestra en el código:

```
#Grafica de doble eje Salinidad y nivel del mar
fig, ax1 = plt.subplots()
tiempo=dfSargento['Ndate']
salinidad=dfSargento.Salinity
agua=dfSargento.Water_Level
ax1.plot(tiempo,salinidad,'y-', label='Salinidad'); plt.legend(loc='upper left')
ax1.set_xlabel('Tiempo')
ax1.set_ylabel('Salinidad (ppt)')
ax2 = ax1.twinx()
ax2.plot(tiempo, agua , 'c-', label='Nivel del mar'); plt.legend(loc='best')
ax2.set_ylabel('Nivel del mar (m)')
fig.tight_layout()
plt.show()
```



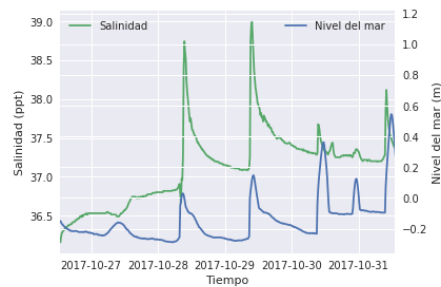
```
#Grafica de doble eje Nivel de mar y Temperatura.
fig, ax1 = plt.subplots()
tiempo=dfSargento['Ndate']
temperatura=dfSargento.Temp
agua=dfSargento.Water_Level
ax1.plot(tiempo,temperatura,'y-', label='Temperatura'); plt.legend(loc='upper right')
ax1.set_xlabel('Tiempo')
ax1.set_ylabel('Temperatura (°C)')
ax2 = ax1.twinx()
ax2.plot(tiempo, agua , 'c-', label='Nivel del mar'); plt.legend(loc='upper left')
ax2.set_ylabel('Nivel del mar (m)')
fig.tight_layout()
plt.show()
```



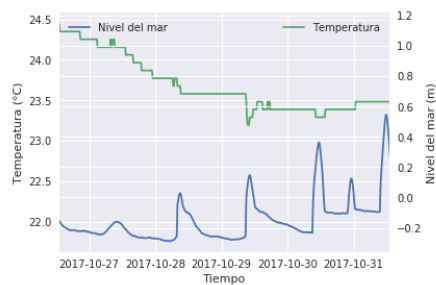
- **Gráficas Independientes con xlim:**

Para poder observar mejor el comportamiento de las variables estudiadas en la parte anterior, se utilizo el comando xlim, para tener un rango de datos más pequeño que estudiar, tomando un rango de 5 días, que van del 26 de octubre al 31 del mismo mes:


```
#Grafica de doble eje Salinidad y nivel del mar
fig, ax1 = plt.subplots()
tiempo=dfSargento['Ndate']
salinidad=dfSargento.Salinity
agua=dfSargento.Water_Level
ax1.plot(tiempo,salinidad,'g-', label='Salinidad'); plt.legend(loc='upper left')
ax1.set_xlabel('Tiempo')
ax1.set_ylabel('Salinidad (ppt)')
ax2 = ax1.twinx()
ax2.plot(tiempo, agua , 'b-', label='Nivel del mar'); plt.legend(loc='best')
ax2.set_ylabel('Nivel del mar (m)')
fig.tight_layout()
plt.xlim(("2017-10-26 13:00:00","2017-10-31 13:00:00"))
plt.show()
```



```
#Grafica de doble eje Nivel de mar y Temperatura.
fig, ax1 = plt.subplots()
tiempo=dfSargento['Ndate']
temperatura=dfSargento.Temp
agua=dfSargento.Water_Level
ax1.plot(tiempo,temperatura,'g-', label='Temperatura'); plt.legend(loc='upper right')
ax1.set_xlabel('Tiempo')
ax1.set_ylabel('Temperatura (°C)')
ax2 = ax1.twinx()
ax2.plot(tiempo, agua , 'b-', label='Nivel del mar'); plt.legend(loc='upper left')
ax2.set_ylabel('Nivel del mar (m)')
fig.tight_layout()
plt.xlim(("2017-10-26 13:00:00","2017-10-31 13:00:00"))
plt.show()
```



En la primera gráfica se puede observar claramente la relación, ya que los picos de una variable corresponden a los picos de la otra. Mientras que en la segunda gráfica se observa como mientras el nivel del mar aumenta, la temperatura va disminuyendo.

3 Conclusión

Con esta evaluación me doy cuenta de lo mucho que he aprendido en lo que va del semestre, pero sobre todo, lo mucho que me falta, ya que las bibliotecas pueden tener muchas funciones que facilitan el análisis de los datos.

Me gustaron mucho las actividades realizadas y espero haberlas completado correctamente.