

UNIVERSIDAD DE SONORA



"El saber de mis hijos
hará mi grandeza"

FÍSICA COMPUTACIONAL I

Actividad 3

Análisis de datos con Pandas y visualización con Plotly

1. Introducción

La presente práctica se enfoca en realizar un análisis de datos de las mareas, salinidad y temperatura registrada por los diversos sensores colocados alrededor del manglar conocido como *El sargento*. Los datos recopilados constan de la presión atmosférica, nivel del mar, temperatura, salinidad y conductividad, así como la fecha y hora correspondiente de cada medición.

Tras analizar los datos proporcionados podremos determinar si existe alguna correlación entre las variables y de qué tipo es, pero antes debemos conocer un poco más acerca de la marea y la salinidad del agua.

La marea, la cual es originada por la atracción gravitacional que ejercen el Sol y la Luna sobre nuestros mares, en conjunto con la erróneamente llamada "fuerza centrífuga", resultante del movimiento angular de la Tierra. Como mencionamos, es principalmente la Luna quien afecta más nuestros mares debido a que se encuentra mucho más cerca del planeta, por lo tanto, la posición en la que se encuentre es de gran importancia pues de ello dependen las mareas bajas (bajamar) y las mareas altas (pleamar), tan así de importante es que durante el día se generan dos mareas de cada tipo cada 12 horas.

La salinidad del mar, como bien dice el mismo nombre, hace referencia a la cantidad de sal disuelta en un litro de agua de mar, sin embargo, es posible que dicho valor sea variante en ciertas áreas debido a la evaporación, estancamiento o flujo de agua con distinta salinidad. Es importante conocer este valor pues tiene repercusiones en el entorno que la contiene, como en su flora y fauna.

2. Desarrollo

Para llevar a cabo el análisis de los datos crearemos un par de gráficas que correspondan con las variables que nos interesan pero antes de ello debemos procesar los datos que nos fueron proporcionados directamente desde la estación en el manglar con el fin de asegurar que sean los necesarios.

2.1. Importación de bibliotecas

Comenzamos importando al cuadernillo de trabajo las bibliotecas que serán necesarias (figura 1).

```
In [1]: 1 import pandas as pd
        2 import numpy as np
        3 import seaborn as sns
        4 import matplotlib.pyplot as plt
        5
        6 #Con este código podremos ver las gráficas desde nuestro cuadernillo de trabajo.
        7 %matplotlib inline
```

Figura 1: Bibliotecas empleadas en el código.

2.2. Lectura de archivos

Después procedemos a leer los archivos .CSV de cada estación del manglar (figura 2).

```
In [2]: 1 Canal = pd.read_csv('./Canal.csv', header=1)
        2 Canal.head()

In [3]: 1 Estacion = pd.read_csv('./Estacion.csv', header=1)
        2 Estacion.head()

In [4]: 1 Salinidad = pd.read_csv('./Salinidad.csv', header=1)
        2 Salinidad.head()
```

Figura 2: Configuración utilizada para la lectura.

2.3. Preparación de los datos

Al momento de leer los archivos .CSV los datos no siempre tienen un formato muy claro o, en ocasiones, presentan ciertos inconvenientes como celdas vacías, información irrelevante, columnas o filas con errores, etcétera. Por eso mismo deben ser procesados previo a trabajar con ellos.

2.3.1. Eliminar datos innecesarios

En esta ocasión, los datos contenían una columna llamada '#', la cual enumeraba cada registro hecho en la base de datos pero ahora no es necesario, así que eliminaremos esta columna (figura 3).

```

In [5]: 1 Temp = Canal.drop('#', axis=1)
        2 Canal = Temp
        3 Canal.head()

In [6]: 1 Temp = Estacion.drop('#', axis=1)
        2 Estacion = Temp
        3 Estacion.head()

In [7]: 1 Temp = Salinidad.drop('#', axis=1)
        2 Salinidad = Temp
        3 Salinidad.head()

```

Figura 3: Removemos la columna '#' de cada archivo.

2.3.2. Cambio de formato

Entre nuestros datos tenemos una columna para la fecha y hora llamada '*Fecha*', sin embargo, el formato que presenta es **object** (se percibe como texto) cuando lo más conveniente es que sea **datetime** para tener un manejo de la información mucho más sencillo (figura 4).

```

In [8]: 1 Canal['Fecha'] = pd.to_datetime(Canal['Fecha'])
        2 Canal.dtypes

In [9]: 1 Estacion['Fecha'] = pd.to_datetime(Estacion['Fecha'])
        2 Estacion.dtypes

In [10]: 1 Salinidad['Fecha'] = pd.to_datetime(Salinidad['Fecha'])
         2 Salinidad.dtypes

```

Figura 4: Modificamos el tipo de la columna '*Fecha*' a datetime.

2.3.3. Renombrar columnas

Modificaremos los encabezados de cada columna, de modo que presenten nombres más cortos, comprensibles y sin caracteres que puedan ocasionar problema al leerlos (figura 5).

2.4. Creación de gráficas

En este punto, la información ya tiene un orden apropiado con el cual podremos crear algunas gráficas para buscar una correlación entre los diversos

```

In [11]: 1 Canal.columns = ['Fecha', 'PresAbs', 'Tempe', 'NivelAgua']
          2 Canal.head()

In [12]: 1 Estacion.columns = ['Fecha', 'PresAbs', 'Tempe', 'NivelAgua']
          2 Estacion.head()

In [13]: 1 Salinidad.columns = ['Fecha', 'ConduLow', 'ConduHigh', 'Tempe', 'ConduEspec', 'Salinidad']
          2 Salinidad.head()

```

Figura 5: Renombramos las columnas para evitar errores posteriormente.

factores analizados. Haremos esto para períodos de tiempo de un día y una semana).

Entre las gráficas a crear tenemos:

- Nivel del mar en la estación y del canal.
- Salinidad y nivel del mar junto a la estación.
- Temperatura del agua en la estación y el canal.
- Correlación entre el nivel de agua y la salinidad.

2.4.1. Datos por períodos

Como mencionamos anteriormente haremos dos nuevos arreglos para trabajar, uno de ellos con un período de un día; el otro, una semana. Para hacer esto tendremos que delimitar el rango de fechas con el cual trabajaremos mediante un par de variables (figura 6).

```

In [14]: 1 #Delimitamos los datos de un sólo día.
          2 LimInfDia = '2018-08-31 23:59:59'
          3 LimSupDia = '2018-09-02 00:00:00'
          4
          5 #Delimitamos los datos de una semana.
          6 LimInfSem = LimInfDia
          7 LimSupSem = '2018-09-08 00:00:00'

```

Figura 6: Limites para los períodos de tiempo.

Una vez hecho esto, lo siguiente es crear nuevos marcos de información con los limites establecidos y además asegurarnos que tengan la misma cantidad de información (filas) pues de no ser así puede ocasionar dificultades (figura 7).

```

In [15]: 1 #Arreglo para un sólo día
2 DiaCanal = Canal[ (Canal['Fecha'] > LimInfDia) &
3                 (Canal['Fecha'] < LimSupDia) ]
4
5 #Arreglo para una semana
6 SemCanal = Canal[ (Canal['Fecha'] > LimInfSem) &
7                 (Canal['Fecha'] < LimSupSem) ]

In [16]: 1 #Arreglo para un sólo día
2 DiaEstacion = Estacion[ (Estacion['Fecha'] > LimInfDia) &
3                       (Estacion['Fecha'] < LimSupDia) ]
4
5 #Arreglo para una semana
6 SemEstacion = Estacion[ (Estacion['Fecha'] > LimInfSem) &
7                       (Estacion['Fecha'] < LimSupSem) ]
8
9 SemEstacion = SemEstacion.drop(SemEstacion.index[671])

In [17]: 1 #Arreglo para un sólo día
2 DiaSalinidad = Salinidad[ (Salinidad['Fecha'] > LimInfDia) &
3                       (Salinidad['Fecha'] < LimSupDia) ]
4
5 #Arreglo para una semana
6 SemSalinidad = Salinidad[ (Salinidad['Fecha'] > LimInfSem) &
7                       (Salinidad['Fecha'] < LimSupSem) ]
8
9 SemSalinidad = SemSalinidad.drop(SemSalinidad.index[671])

```

Figura 7: Creación de nuevos marcos de información y delimitados.

2.4.2. Gráficas

Finalmente es posible proceder a crear las gráficas mencionadas en un principio.

Nivel del mar

Esta gráfica muestra el comportamiento del nivel del mar en la estación y el canal del manglar durante un día y una semana.

- Código empleado para un día (figura 8).

```
In [18]: 1 #Datos a graficar
2 x = DiaEstacion['Fecha']
3 y1 = DiaEstacion['NivelAgua']
4 y2 = DiaCanal['NivelAgua']
5
6 plt.title('Nivel del Agua (1 Día)', fontweight='bold') #Título
7 plt.xlabel('Hora', fontweight='bold') #Eje X
8 plt.xticks(rotation=45) #Datos del eje X
9
10 #plt.xlim(['2018-09-01 00:00:00', '2018-09-01 23:59:59'])
11
12 EjeIzq = plt.subplot() #Creamos el primer eje, tomando en cuenta que serán dos.
13 EjeDer = EjeIzq.twinx() #Creamos el segundo, indicando que comparten X.
14
15 #Primer eje -----
16 EjeIzq.set_ylabel('Estación (m)', color='b', fontweight='bold')
17 EjeIzq.tick_params(axis='y', labelcolor='b')
18 EjeIzq.plot(x, y1, color='b')
19
20 #Segundo eje -----
21 EjeDer.set_ylabel('Canal (m)', color='r', fontweight='bold')
22 EjeDer.tick_params(axis='y', labelcolor='r')
23 EjeDer.plot(x, y2, color='r')
24
25 plt.grid()
26 plt.tight_layout()
27 plt.savefig('Nivel del Agua - Dia')
```

Figura 8: Configuración para generar la gráfica del nivel del mar de un día.

- Gráfico generado para un día (figura 9).
- Código empleado para una semana (figura 10).
- Gráfico generado para una semana (figura 11).

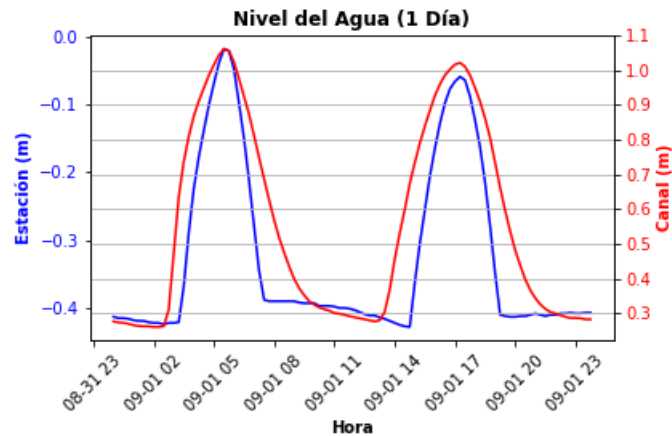


Figura 9: Gráfica resultante para el nivel del mar durante un día.

```
In [19]: 1 #Datos a graficar
2 x = SemEstacion['Fecha']
3 y1 = SemEstacion['NivelAgua']
4 y2 = SemCanal['NivelAgua']
5
6 plt.title('Nivel del Agua (1 Semana)', fontweight='bold') #Título
7 plt.xlabel('Día', fontweight='bold') #Eje X
8 plt.xticks(rotation=45) #Datos del eje X
9
10 EjeIzq = plt.subplot() #Creamos el primer eje, tomando en cuenta que serán dos.
11 EjeDer = EjeIzq.twinx() #Creamos el segundo, indicando que comparten X.
12
13 #Primer eje -----
14 EjeIzq.set_ylabel('Estación (m)', color = 'b', fontweight='bold')
15 EjeIzq.tick_params(axis='y', labelcolor = 'b')
16 EjeIzq.plot(x,y1, color='b')
17
18 #Segundo eje -----
19 EjeDer.set_ylabel('Canal (m)', color='r', fontweight='bold')
20 EjeDer.tick_params(axis='y', labelcolor= 'r')
21 EjeDer.plot(x, y2, color='r')
22
23 plt.grid()
24 plt.tight_layout()
25 plt.savefig('Nivel del Agua - Semana')
```

Figura 10: Configuración para generar la gráfica del nivel del mar de una semana.

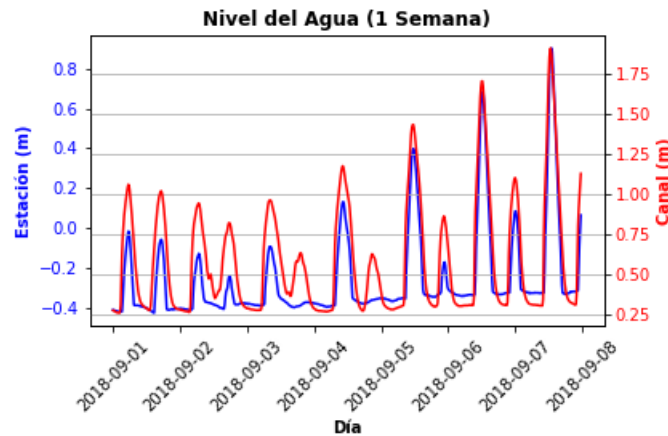


Figura 11: Gráfica resultante para el nivel del mar durante una semana.

Salinidad y nivel del mar

La gráfica muestra el comportamiento de la salinidad y el nivel del agua en la estación del manglar durante un día y una semana.

- Código empleado para un día (figura 12).
- Gráfico generado para un día (figura 13).
- Código empleado para una semana (figura 14).
- Gráfico generado para una semana (figura 15).

Temperatura del agua

La gráfica muestra el comportamiento de la temperatura del agua en la estación y el canal del manglar durante un día y una semana.

- Código empleado para un día (figura 16).
- Gráfico generado para un día (figura 17).
- Código empleado para una semana (figura 18).

```

In [20]: 1 #Datos a graficar
2 x = DiaEstacion['Fecha']
3 y1 = DiaSalinidad['Salinidad']
4 y2 = DiaEstacion['NivelAgua']
5
6 plt.title('Datos de la Estación (1 Día)', fontweight='bold') #Título
7 plt.xlabel('Hora', fontweight='bold') #Eje X
8 plt.xticks(rotation=45) #Datos del eje X
9
10 EjeIzq = plt.subplot() #Creamos el primer eje, tomando en cuenta que serán dos.
11 EjeDer = EjeIzq.twinx() #Creamos el segundo, indicando que comparten X.
12
13 #Primer eje -----
14 EjeIzq.set_ylabel('Salinidad (ppt)', color = 'g', fontweight='bold')
15 EjeIzq.tick_params(axis='y', labelcolor = 'g')
16 EjeIzq.plot(x,y1, color='g')
17
18 #Segundo eje -----
19 EjeDer.set_ylabel('Nivel del Agua (m)', color='m', fontweight='bold')
20 EjeDer.tick_params(axis='y', labelcolor= 'm')
21 EjeDer.plot(x, y2, color='m')
22
23 plt.grid()
24 plt.tight_layout()
25 plt.savefig('Salinidad y Nivel del Agua - Dia')

```

Figura 12: Configuración para generar la gráfica de salinidad y nivel del mar de un día.

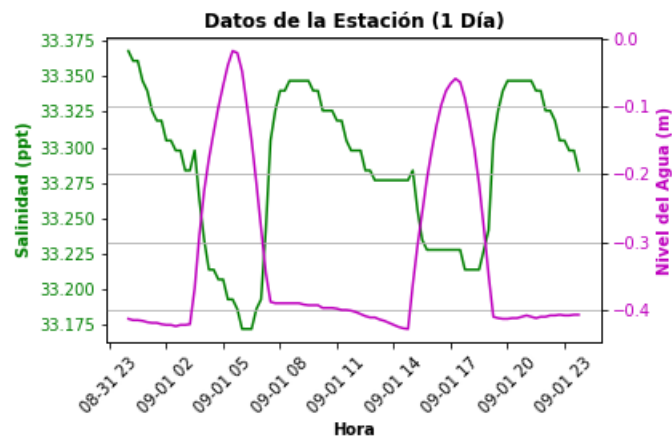


Figura 13: Gráfica resultante de la salinidad y nivel del mar durante un día.

```

In [21]: 1 #Datos a graficar
2 x = SemEstacion['Fecha']
3 y1 = SemSalinidad['Salinidad']
4 y2 = SemEstacion['NivelAgua']
5
6 plt.title('Datos de la Estación (1 Semana)', fontweight='bold') #Título
7 plt.xlabel('Hora', fontweight='bold') #Eje X
8 plt.xticks(rotation=45) #Datos del eje X
9
10 EjeIzq = plt.subplot() #Creamos el primer eje, tomando en cuenta que serán dos.
11 EjeDer = EjeIzq.twinx() #Creamos el segundo, indicando que comparten X.
12
13 #Primer eje -----
14 EjeIzq.set_ylabel('Salinidad (ppt)', color = 'g', fontweight='bold')
15 EjeIzq.tick_params(axis='y', labelcolor = 'g')
16 EjeIzq.plot(x,y1, color='g')
17
18 #Segundo eje -----
19 EjeDer.set_ylabel('Nivel del Agua (m)', color='m', fontweight='bold')
20 EjeDer.tick_params(axis='y', labelcolor= 'm')
21 EjeDer.plot(x, y2, color='m')
22
23 plt.grid()
24 plt.tight_layout()
25 plt.savefig('Salinidad y Nivel del Agua - Semana')

```

Figura 14: Configuración para generar la gráfica de salinidad contra nivel del mar por una semana.

- Gráfico generado para una semana (figura 19).

Correlación

La siguiente tabla mostrará la correlación entre cada una de las variables con el resto de ellas.

- Código empleado para un día (figura 20).
- Gráfico generado para un día (figura 21).
- Código empleado para una semana (figura 22).
- Gráfico generado para una semana (figura 23).

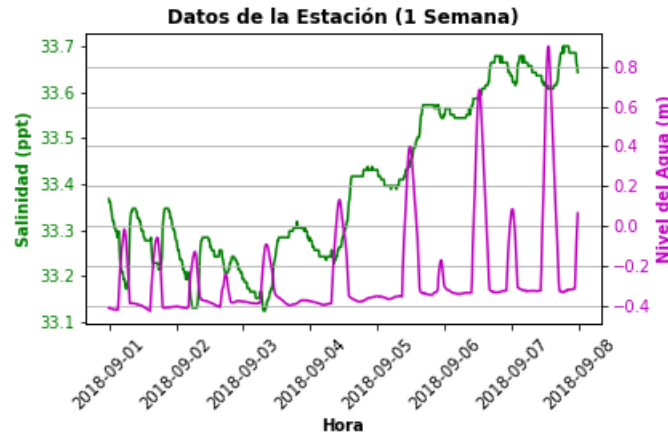


Figura 15: Gráfica resultante de la salinidad y el nivel del mar durante una semana.

2.5. Nuevos datos

Agregamos unos cuantos datos más para tener una vista más amplia de estos y ver posibles ciclos a lo largo de los meses, y para hacer esto haremos nuevamente cada unos de los pasos iniciales más uno extra para agregar todo.

2.5.1. Procesamiento de los datos

Los pasos realizados para estos nuevos datos serán los mostrados en la figura 24.

2.5.2. Adición de datos

Para proceder en este punto debemos agregar los datos originales y los nuevos en un sólo marco de datos (figura 25).

2.5.3. Gráfico resultante

La siguiente gráfica muestra el nivel del mar en la estación y el canal a lo largo de aproximadamente 2 meses.

- Código empleado para la configuración del gráfico (figura 26).

```

In [22]: 1 #Datos a graficar
2 x = DiaEstacion['Fecha']
3 y1 = DiaEstacion['Tempe']
4 y2 = DiaCanal['Tempe']
5
6 plt.title('Temperatura del Agua (1 Día)', fontweight='bold') #Título
7 plt.xlabel('Hora', fontweight='bold') #Eje X
8 plt.xticks(rotation=45) #Datos del eje X
9
10 EjeIzq = plt.subplot() #Creamos el primer eje, tomando en cuenta que serán dos.
11 EjeDer = EjeIzq.twinx() #Creamos el segundo, indicando que comparten X.
12
13 #Primer eje -----
14 EjeIzq.set_ylabel('Estación (°C)', color = 'Navy', fontweight='bold')
15 EjeIzq.tick_params(axis='y', labelcolor = 'Navy')
16 EjeIzq.plot(x,y1, color='Navy')
17
18 #Segundo eje -----
19 EjeDer.set_ylabel('Canal (°C)', color='y', fontweight='bold')
20 EjeDer.tick_params(axis='y', labelcolor='y')
21 EjeDer.plot(x, y2, color='y')
22
23 plt.grid()
24 plt.tight_layout()
25 plt.savefig('Temperatura del Agua - Dia')

```

Figura 16: Configuración para generar la gráfica de la temperatura del mar en la estación y el canal del manglar por un día.

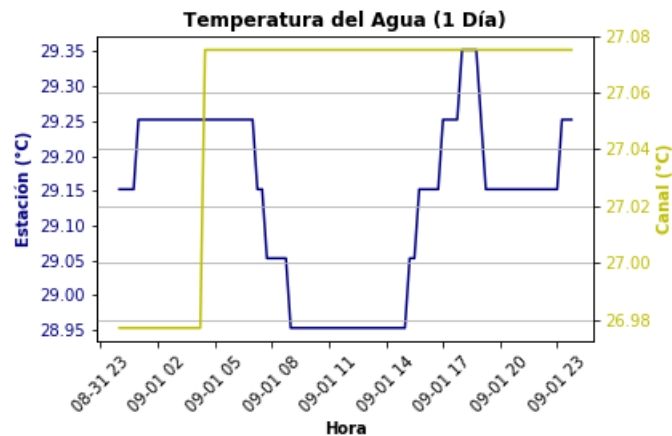


Figura 17: Gráfica resultante para la temperatura del mar durante un día.

```

In [23]: 1 #Datos a graficar
2 x = SemEstacion['Fecha']
3 y1 = SemEstacion['Tempe']
4 y2 = SemCanal['Tempe']
5
6 plt.title('Temperatura del Agua (1 Semana)', fontweight='bold') #Título
7 plt.xlabel('Día', fontweight='bold') #Eje X
8 plt.xticks(rotation=45) #Datos del eje X
9
10 EjeIzq = plt.subplot() #Creamos el primer eje, tomando en cuenta que serán dos.
11 EjeDer = EjeIzq.twinx() #Creamos el segundo, indicando que comparten X.
12
13 #Primer eje -----
14 EjeIzq.set_ylabel('Estación (°C)', color = 'Navy', fontweight='bold')
15 EjeIzq.tick_params(axis='y', labelcolor = 'Navy')
16 EjeIzq.plot(x,y1, color='Navy')
17
18 #Segundo eje -----
19 EjeDer.set_ylabel('Canal (°C)', color='y', fontweight='bold')
20 EjeDer.tick_params(axis='y', labelcolor= 'y')
21 EjeDer.plot(x, y2, color='y')
22
23 plt.grid()
24 plt.tight_layout()
25 plt.savefig('Temperatura del Agua - Semana')

```

Figura 18: Configuración para generar una gráfica de temperatura del mar de la estación y el canal del manglar durante una semana.

- Gráfica generada para el nivel del mar (figura 27).

3. Conclusión

Para dar una conclusión habrá que basarse en los resultados observados en las tabla de correlación de un día y una semana.

Valores de correlación para un día

- Presión atmosférica absoluta y nivel del mar (1)
- Conductividad específica y salinidad (1)
- Presión atmosférica absoluta y temperatura del mar (0.5)
- Temperatura del mar y nivel del mar (0.5)
- Temperatura del mar y conductividad específica (-0.43)
- Temperatura del mar y salinidad (-0.43)
- Presión atmosférica absoluta y conductividad específica (-0.86)

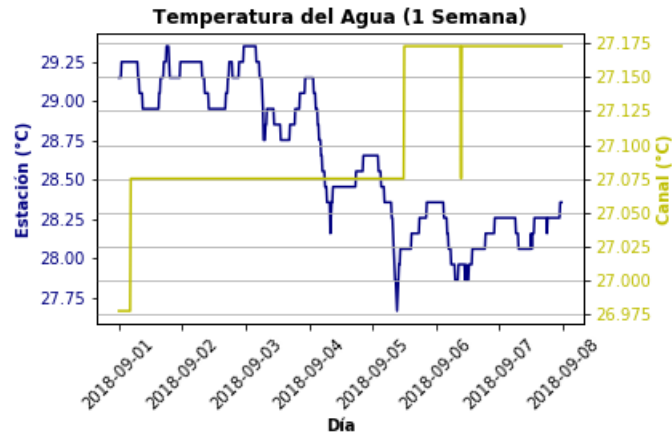


Figura 19: Gráfico correspondiente a la temperatura del mar en la estación y canal del manglar durante una semana.

```
In [24]: 1 #Crearemos un nuevo DataFrame con los datos del NIVEL DEL AGUA de la ESTACIÓN.
2 DatosCorr = DiaEstacion.drop('Fecha', axis=1)
3
4 #Las columnas restantes son información de la SALINIDAD.
5 DatosCorr['ConduEspec'] = DiaSalinidad.ConduEspec
6 DatosCorr['Salinidad'] = DiaSalinidad.Salinidad
7
8 #Eliminamos aquellos renglones donde exista un NaN.
9 DatosCorr = DatosCorr.dropna()
10
11 #Calculos la correlación entre las variables
12 Corre = DatosCorr.corr()
13
14 #Creamos el mapa de calor.
15 Mapa = sns.heatmap(Corre, square=True, annot=True, annot_kws={'size': 12}, linewidths=1,
16                     linecolor='black', cmap='YlGnBu')
17
18 Imagen = Mapa.get_figure()
19 Imagen.tight_layout()
20 Imagen.savefig('Correlacion Agua y Salinidad - 1 Dia')
```

Figura 20: Configuración para la tabla de correlación por un día.

- Presión atmosférica absoluta y salinidad (-0.86)
- Nivel del mar y conductividad específica (-0.86)
- Nivel del mar y salinidad (-0.86)

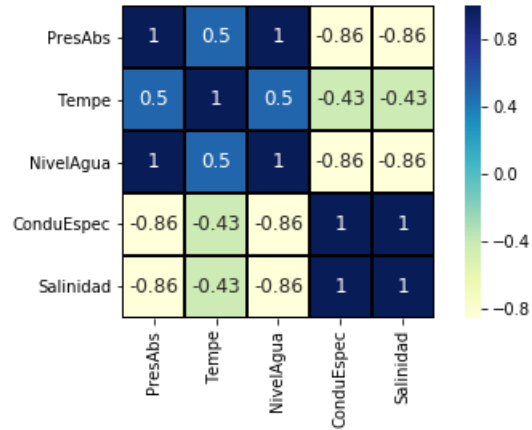


Figura 21: Tabla generada mostrando la correlación entre las variables durante un día.

```
In [25]: 1 #Crearemos un nuevo DataFrame con los datos del NIVEL DEL AGUA de la ESTACIÓN.
2 DatosCorr = SemEstacion.drop('Fecha', axis=1)
3
4 #Las columnas restantes son información de la SALINIDAD.
5 DatosCorr['ConduEspec'] = SemSalinidad.ConduEspec
6 DatosCorr['Salinidad'] = SemSalinidad.Salinidad
7
8 #Eliminamos aquellos renglones donde exista un NaN.
9 DatosCorr = DatosCorr.dropna()
10
11 #Calculos la correlación entre las variables
12 Corre = DatosCorr.corr()
13
14 #Creamos el mapa de calor.
15 Mapa = sns.heatmap(Corre, square=True, annot=True, annot_kws={'size': 12}, linewidths=1,
16                     linecolor='black', cmap='YlGnBu')
17
18 Imagen = Mapa.get_figure()
19 Imagen.tight_layout()
20 Imagen.savefig('Correlacion Agua y Salinidad - 1 Semana')
```

Figura 22: Configuración para la tabla de correlación por una semana.

Valores de correlación para una semana

- Presión atmosférica absoluta y nivel del mar (1)
- Conductividad específica y salinidad (1)
- Presión atmosférica absoluta y conductividad específica (0.28)
- Presión atmosférica absoluta y salinidad (0.28)

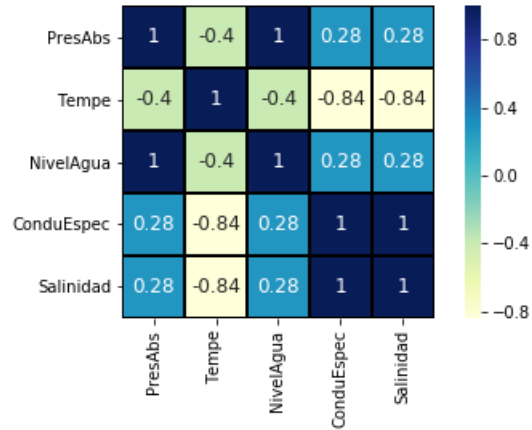


Figura 23: Tabla generada mostrando la correlación entre las variables durante una semana.

```
In [26]: 1 #Lectura de los nuevos datos
2
3 Canal2 = pd.read_csv('./Canal-2.csv', skiprows=2, header=None)
4 Canal2.drop([0], axis=1, inplace=True)
5
6 Estacion2 = pd.read_csv('./Estacion-2.csv', skiprows=2, header=None)
7 Estacion2.drop([0], axis=1, inplace=True)

In [28]: 1 #Ajuste del formato de datos
2 Canal2[1] = pd.to_datetime(Canal2[1])
3
4 Estacion2[1] = pd.to_datetime(Estacion2[1])

In [29]: 1 #Renombrar columnas
2 Canal2.columns = ['Fecha', 'PresAbs', 'Tempe', 'NivelAgua']
3
4 Estacion2.columns = ['Fecha', 'PresAbs', 'Tempe', 'NivelAgua']
```

Figura 24: Tratamiento entero para los nuevos datos.

- Nivel del mar y conductividad específica (0.28)
- Nivel del mar y salinidad (0.28)
- Presión atmosférica absoluta y temperatura del mar (-0.4)
- Temperatura del mar y nivel del mar (-0.4)
- Temperatura del mar y conductividad específica (-0.84)
- Temperatura del mar y salinidad (-0.84)

```
In [30]: 1 CanalALL = Canal
2 CanalALL = CanalALL.append(Canal2, ignore_index=True)
3 CanalALL = CanalALL[:-4]
4
5 EstacionALL = Estacion
6 EstacionALL = EstacionALL.append(Estacion2, ignore_index=True)
```

Figura 25: Código empleado para la limpieza de los datos.

```
In [31]: 1 #Grafica del nivel del mar
2
3 x = EstacionALL['Fecha']# + CanalALL['Fecha']
4 y1 = EstacionALL['NivelAgua']
5 y2 = CanalALL['NivelAgua']
6
7 plt.title('Nivel del Mar', fontweight='bold') #Título
8 plt.xlabel('Fecha', fontweight='bold') #Eje X
9 plt.xticks(rotation=45) #Datos del eje X
10
11 EjeIzq = plt.subplot() #Creamos el primer eje, tomando en cuenta que serán dos.
12 EjeDer = EjeIzq.twinx() #Creamos el segundo, indicando que comparten X.
13
14 #Primer eje -----
15 EjeIzq.set_ylabel('Estación (m)', color = 'r', fontweight='bold')
16 EjeIzq.tick_params(axis='y', labelcolor = 'r')
17 EjeIzq.plot(x,y1, color='r')
18
19 #Segundo eje -----
20 EjeDer.set_ylabel('Canal (m)', color='y', fontweight='bold')
21 EjeDer.tick_params(axis='y', labelcolor= 'y')
22 EjeDer.plot(x, y2, color='y')
23
24 plt.grid()
25 plt.tight_layout()
26 plt.savefig('Nivel del Mar')
```

Figura 26: Código empleado para la gráfica del nivel del mar con todos los datos.

Podemos apreciar que en el período de un día la presión atmosférica absoluta y el nivel del mar, así como la conductividad específica y la salinidad están directamente relacionadas, mientras que la presión atmosférica esta anticorrelacionada con la conductividad específica y salinidad, y de igual manera el nivel del mar con la conductividad específica y salinidad.

Durante el período de una semana los resultados arrojados cambiaron

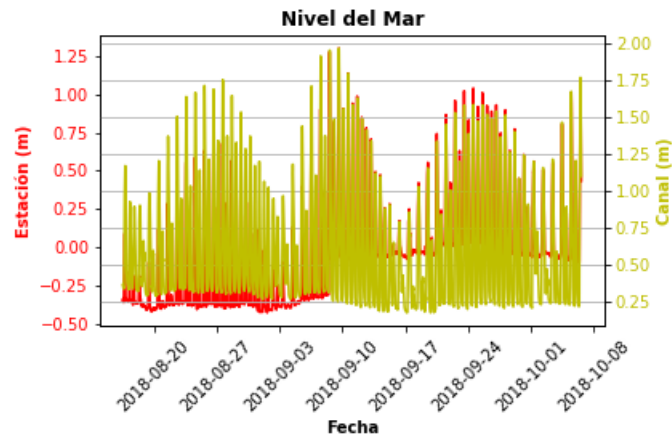


Figura 27: Gráfica del nivel del mar registrados en la estación y canal del manglar del registro total.

pues si bien los valores correlacionados son los mismos, los anticorrelacionados ahora son la temperatura del mar con la conductividad específica y salinidad.

Finalmente podemos concluir que la salinidad esta anticorrelacionada con el nivel del mar a corto plazo pero conforme aumentamos el período analizado nos damos cuenta que la salinidad deja de relacionarse tanto con el nivel del mar y más con la temperatura de este.

Referencias

- [1] (2016, Noviembre 4). LA MAREA, QUÉ ES Y CÓMO SE FORMA, MAREAS VIVAS Y MAREAS MUERTAS, PLEAMAR Y BAJAMAR. 9 de diciembre del 2018, de Sail and Trip. Recuperado de <https://sailandtrip.com/la-marea/>
- [2] (2013, Octubre 7). SALINIDAD. 9 de diciembre del 2018, de EcuRed. Recuperado de <https://www.ecured.cu/Salinidad>