

December 11, 2018

computacional1, Reporte 3
Juan Jose López
November 2018

1 Introduction

En esta actividad descargamos dos archivos de datos de la pagina computacional1, correspondientes a los niveles de marea de las estaciones canal y sargento que se encuentran a unos 100km de Bahia de Kino, Sonora. Estan en contacto con el agua del golfo de california. Queremos saber que informacion nos dicen los datos respecto al comportamiento del agua en esa zona.

2 Terminologia

A continuación se recogen los principales términos empleados en la descripción de las mareas:

Marea alta o pleamar: momento en que el agua del mar alcanza su máxima altura dentro del ciclo de las mareas.

Marea baja o bajamar: momento opuesto, en que el mar alcanza su menor altura. El tiempo aproximado entre una pleamar y la bajamar es de 6 horas, completando un ciclo de 24 horas 50 minutos.

Flujo: el flujo es el proceso de ascenso lento y continuo de las aguas marinas, debido al incremento progresivo de la atracción lunar o solar o de ambas atracciones en el caso de luna nueva y de luna llena.

Reflujo: el reflujo es el proceso de descenso de las aguas marinas, lento y progresivo, debido a la decadencia de la atracción lunar o solar.

Carrera o amplitud de marea: diferencia de altura entre pleamar y bajamar.

Rango micromareal: cuando la carrera de marea es menor de 2 metros.

Rango mesomareal: cuando la carrera de marea está comprendida entre los 2 metros y los 4 metros.

Rango macromareal: cuando la carrera de marea es mayor de 4 metros.¹

Semiperíodo de marea: diferencia en el tiempo entre pleamar y bajamar.

Estoa de marea: es el momento en el que el nivel permanece fijo en la pleamar o en la bajamar.

Estoa de corriente: es el instante en que la corriente asociada a la marea se anula.

Establecimiento del puerto: es el desfase existente, debido a la inercia de la hidrosfera, entre el paso de la Luna por el meridiano del lugar y la aparición de la pleamar siguiente.

Edad de la marea: es el desfase existente, por la misma razón, entre el paso de la Luna llena por el meridiano del lugar y la máxima pleamar mensual siguiente.

Unidad de altura: promedio durante 19 años (un ciclo nodal o ciclo de metón) de las dos máximas carreras de marea (equinoccios) de cada año del ciclo.

Marea viva, alta o sizigia: son las mareas que se producen con la luna llena y la luna nueva, cuando el Sol, la Luna y la Tierra se encuentran alineados. La Marea Viva que se produce durante la fase de Luna Nueva se denomina

”Marea Viva de Conjunción”; y la que se produce mientras tiene lugar la fase de luna llena se llama ”Marea Viva de Oposición”.

Marea muerta, baja o de cuadratura: son las mareas que se producen durante las fases de Cuarto Creciente y Cuarto Menguante, cuando las posiciones de la Tierra, el Sol y la Luna forman un ángulo aparente de 90° .

Líneas cotidales: las líneas cotidales (del inglés tide: marea) son las líneas que unen los puntos en los cuales la pleamar es simultánea.

Puntos anfidrómicos o puntos de anfidromia: son zonas hacia las cuales convergen las líneas cotidales y en las que la amplitud de la marea es cero.

Puerto patrón: son los puntos geográficos para los cuales se calcula y publica la predicción de fecha y altura de marea. Puerto secundario: son puntos geográficos de interés para el navegante pero que no tienen publicado un cálculo de predicción de mareas, pero sí una corrección en cuanto a hora y altura que los refiere a un puerto patrón y mediante la cual se pueden determinar igualmente los datos de marea.

Tablas de marea: son las publicaciones anuales con la predicción diaria de las alturas de marea. Suministran, entre otros datos, fecha, hora y altura de marea para diferentes puntos a lo largo del litoral marítimo.

3 Etapas del desarrollo

En esta sección se presentan las líneas de código Python que se utilizaron en Jupiter para generar el archivo ”el Manglar El Sargento.ipynb”

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Empiezo por poner las librerías que creo que voy a utilizar. Seleccione la de

"Pandas" que me ayuda a manejar los archivos de datos; "matplotlib.pyplot" y "numpy" los use para las graficas anteriormente y los crei conveniente utilizarlas.

importar datos

En esta seccion leo los archivos de datos que contienen informacion sobre las mareas. Tengo dos *DataFrame* llamados, "canal" y "old", que ofrecen la informacion de un de mes sobre las siguientes caracteisticas de la marea: "presion en abs", "temperatura del agua en grados centigrados" y "nivel de la marea en metros"; ademas hay dos *DataFrame* más a los que llame "canal2" y "old2", que presentan informacion del mes siguiente a este.

Hay un ultimo *DataFrame* llamado "salinidad", aqui se encuentra informacion en las mismas fechas que "canal" y "old", sobre: "conductividad del agua", y "niveles de salinidad"

```
In [2]: old = pd.read_csv('sargento-old-080918.csv', header=None, skiprows=2 )
old.head()
```

Out[2]:

	0	1	2	3	4
0	1	08/16/2018 12:30:00	104.359	29.252	-0.340
1	2	08/16/2018 12:45:00	104.243	28.754	-0.351
2	3	08/16/2018 13:00:00	104.255	28.655	-0.350
3	4	08/16/2018 13:15:00	104.239	28.655	-0.352
4	5	08/16/2018 13:30:00	104.255	28.655	-0.350

```
In [7]: canal = pd.read_csv('sargento-canal.csv', header=None, skiprows=1)
canal.head()
```

Out[7]:

	0	1	2	3	4
0	1	08/16/2018 11:15:01	112.032	29.152	0.360
1	2	08/16/2018 11:30:02	111.990	27.567	0.356
2	3	08/16/2018 11:45:03	111.928	27.272	0.350
3	4	08/16/2018 12:00:04	111.899	27.173	0.347
4	5	08/16/2018 12:15:05	111.870	27.075	0.344

```
In [8]: salinidad=pd.read_csv('salinidad_080918.csv', header=None, skiprows=2)
salinidad.head()
```

```
Out[8]:
```

	0	1	2	3	4	5	6
0	1	08/16/2018 12:45:00	33686	54525.5	28.54	50942.8	33.4310
1	2	08/16/2018 13:00:00	33686	54525.5	28.43	51048.0	33.5085
2	3	08/16/2018 13:15:00	33686	54525.5	28.35	51124.8	33.5651
3	4	08/16/2018 13:30:00	33686	54525.5	28.31	51163.3	33.5935
4	5	08/16/2018 13:45:00	33686	54525.5	28.28	51192.2	33.6148

```
In [9]: old2 = pd.read_csv('sargento-old-061018.csv', header=None, skiprows=2 )
old2.head()
```

```
Out[9]:
```

	0	1	2	3	4
0	1	09/08/2018 11:45:00	111.202	28.853	0.417
1	2	09/08/2018 12:00:00	112.820	28.357	0.578
2	3	09/08/2018 12:15:00	114.550	28.159	0.750
3	4	09/08/2018 12:30:00	116.325	28.258	0.927
4	5	09/08/2018 12:45:00	117.804	28.159	1.074

```
In [10]: canal2 = pd.read_csv('sargento-canal-061018.csv', header=None, skiprows=2 )
canal2.head()
```

```
Out[10]:
```

	0	1	2	3	4
0	1	09/08/2018 11:15:00	116.031	28.258	0.505
1	2	09/08/2018 11:30:01	117.781	27.468	0.679
2	3	09/08/2018 11:45:02	118.999	27.370	0.800
3	4	09/08/2018 12:00:03	120.469	27.370	0.946
4	5	09/08/2018 12:15:04	122.265	27.370	1.125

Trabajo de datos

Anteriormente me salte el encabezado de las columnas al momento de leerlo en Jupyter, y en esta seccion me dedico a darle un nombre, porque Jupyter solamente las enumera.

```
In [11]: canal.columns = ['#', 'Fecha', 'Presion', 'Temp', 'Nivel']
canal.head()
```

```
Out[11]:
```

	#	Fecha	Presion	Temp	Nivel
0	1	08/16/2018 11:15:01	112.032	29.152	0.360
1	2	08/16/2018 11:30:02	111.990	27.567	0.356
2	3	08/16/2018 11:45:03	111.928	27.272	0.350
3	4	08/16/2018 12:00:04	111.899	27.173	0.347
4	5	08/16/2018 12:15:05	111.870	27.075	0.344

```
In [12]: old.columns = ['#', 'Fecha', 'Presion', 'Temp', 'Nivel']
old.head()
```

```
Out[12]:
```

	#	Fecha	Presion	Temp	Nivel
0	1	08/16/2018 12:30:00	104.359	29.252	-0.340
1	2	08/16/2018 12:45:00	104.243	28.754	-0.351
2	3	08/16/2018 13:00:00	104.255	28.655	-0.350
3	4	08/16/2018 13:15:00	104.239	28.655	-0.352
4	5	08/16/2018 13:30:00	104.255	28.655	-0.350

```
In [13]: salinidad.columns = ['#', 'Fecha', 'Conductividad baja', 'Conductividad baja',
                              'Temp', 'conductancia especifica, s/cm,',
                              'sal, ppt',]
salinidad.head()
```

```
Out[13]:
```

	#	Fecha	Conductividad baja	Conductividad baja	Temp	conductancia especifica, s/cm,	sal, ppt
0	1	08/16/2018 12:45:00	33686	54525.5	28.54	50942.8	33.4310
1	2	08/16/2018 13:00:00	33686	54525.5	28.43	51048.0	33.5085
2	3	08/16/2018 13:15:00	33686	54525.5	28.35	51124.8	33.5651
3	4	08/16/2018 13:30:00	33686	54525.5	28.31	51163.3	33.5935
4	5	08/16/2018 13:45:00	33686	54525.5	28.28	51192.2	33.6148

```
In [14]: canal2.columns = ['#', 'Fecha', 'Presion', 'Temp', 'Nivel']
canal2.head()
```

```
Out[14]:
```

	#	Fecha	Presion	Temp	Nivel
0	1	09/08/2018 11:15:00	116.031	28.258	0.505
1	2	09/08/2018 11:30:01	117.781	27.468	0.679
2	3	09/08/2018 11:45:02	118.999	27.370	0.800
3	4	09/08/2018 12:00:03	120.469	27.370	0.946
4	5	09/08/2018 12:15:04	122.265	27.370	1.125

```
In [15]: old2.columns = ['#', 'Fecha', 'Presion', 'Temp', 'Nivel']
old2.head()
```

```
Out[15]:
```

	#	Fecha	Presion	Temp	Nivel
0	1	09/08/2018 11:45:00	111.202	28.853	0.417
1	2	09/08/2018 12:00:00	112.820	28.357	0.578
2	3	09/08/2018 12:15:00	114.550	28.159	0.750
3	4	09/08/2018 12:30:00	116.325	28.258	0.927
4	5	09/08/2018 12:45:00	117.804	28.159	1.074

cambiar formato de fecha a tiempo

Como lo dice el nombre de esta seccion, cambie el formato de la primer columna para que Jupyter la lea como si fuera una fecha. La columna fecha paso de ser "float64" a "datetime64"

```
In [16]: canal['Fecha'] = pd.to_datetime(canal['Fecha'])
         canal.dtypes
```

```
Out[16]: #                int64
         Fecha      datetime64[ns]
         Presion      float64
         Temp        float64
         Nivel        float64
         dtype: object
```

```
In [17]: old['Fecha'] = pd.to_datetime(old['Fecha'])
         old.dtypes
```

```
Out[17]: #                int64
         Fecha      datetime64[ns]
         Presion      float64
         Temp        float64
         Nivel        float64
         dtype: object
```

```
In [18]: salinidad['Fecha'] = pd.to_datetime(salinidad['Fecha'])
         salinidad.dtypes
```

```
Out[18]: #                int64
         Fecha      datetime64[ns]
         Conductividad baja      int64
         Conductividad baja      float64
         Temp        float64
         conductancia especifica, s/cm,      float64
         sal, ppt      float64
         dtype: object
```

```
In [19]: canal2['Fecha'] = pd.to_datetime(canal2['Fecha'])
         canal2.dtypes
```

```
Out[19]: #                int64
         Fecha      datetime64[ns]
         Presion      float64
         Temp        float64
         Nivel        float64
         dtype: object
```

```
In [20]: old2['Fecha'] = pd.to_datetime(old2['Fecha'])
         old2.dtypes
```

```
Out[20]: #          int64
         Fecha      datetime64[ns]
         Presion    float64
         Temp       float64
         Nivel      float64
         dtype: object
```

Datos de la estacion, del canal y de la salinidad

Como ya tenia los *DataFrames* de "old" y "canal" en formato "datetime64", en los siguientes dos renglones de codigo cree dos nuevos *DataFrame* para para contener la informacion de una semana y un dia.

```
In [21]: #un dia
         diacanal = canal[(canal['Fecha'] > '09/07/2018 00:04') & (canal['Fecha'] < '09/07/2018 23:51')]
         diaold = old[(old['Fecha'] < '09/07/2018 23:45') & (old['Fecha'] > '09/07/2018 00:00')]
         diasal = salinidad[(salinidad['Fecha'] < '09/07/2018 23:45') & (salinidad['Fecha'] > '09/07/2018 00:00')]
```

```
In [22]: #una semana
         semanacanal = canal[(canal['Fecha'] < '09/08/2018 10:51') & (canal['Fecha'] > '09/01/2018 10:40')]
         semanaold = old[(old['Fecha'] < '09/08/2018 11:30') & (old['Fecha'] > '09/01/2018 11:30')]
         semanasal = salinidad[(salinidad['Fecha'] < '09/08/2018 11:15') & (salinidad['Fecha'] > '09/01/2018 11:15')]
```

Quize que Jupyter me mostrara la informacion en la pantalla y me ayude con la accion *DataFrame.head()*, pero como era extensos esa opcion no me permitio visualizarlos como yo queria, entonces me servi de la accion *print(DataFrame)*

```
In [23]: diacanal.head()
```

```
Out[23]:
```

	#	Fecha	Presion	Temp	Nivel
2066	2067	2018-09-07 00:19:00	119.499	27.173	1.103
2067	2068	2018-09-07 00:34:00	119.369	27.173	1.090
2068	2069	2018-09-07 00:49:00	119.083	27.173	1.062
2069	2070	2018-09-07 01:04:00	118.667	27.173	1.020
2070	2071	2018-09-07 01:19:00	118.160	27.173	0.970


```
In [24]: diaold['Fecha'].head()
```

```
Out[24]: 2063    2018-09-07 00:15:00
          2064    2018-09-07 00:30:00
          2065    2018-09-07 00:45:00
          2066    2018-09-07 01:00:00
          2067    2018-09-07 01:15:00
          Name: Fecha, dtype: datetime64[ns]
```

```
In [25]: diacanal['Fecha'].head()
```

```
Out[25]: 2066    2018-09-07 00:19:00
          2067    2018-09-07 00:34:00
          2068    2018-09-07 00:49:00
          2069    2018-09-07 01:04:00
          2070    2018-09-07 01:19:00
          Name: Fecha, dtype: datetime64[ns]
```

```
In [26]: diasal.head()
```

```
Out[26]:
```

	#	Fecha	Conductividad baja	Conductividad baja	Temp	conductancia especifica, s/cm,	sal, ppt
2062	2063	2018-09-07 00:15:00	33686	54525.5	28.27	51201.8	33.6219
2063	2064	2018-09-07 00:30:00	33686	54525.5	28.27	51201.8	33.6219
2064	2065	2018-09-07 00:45:00	33686	54525.5	28.27	51201.8	33.6219
2065	2066	2018-09-07 01:00:00	33686	54525.5	28.27	51201.8	33.6219
2066	2067	2018-09-07 01:15:00	33686	54525.5	28.28	51192.2	33.6148

```
In [30]: print(semancanal)
```

#	Fecha	Presion	Temp	Nivel
1531	1534 2018-09-01 10:55:00	111.405	27.075	0.307
1532	1535 2018-09-01 11:10:00	111.430	27.075	0.306
1533	1536 2018-09-01 11:25:00	111.404	27.075	0.298
1534	1537 2018-09-01 11:40:00	111.378	27.075	0.295
1535	1538 2018-09-01 11:55:00	111.339	27.075	0.291
1536	1539 2018-09-01 12:10:00	111.301	27.075	0.288
1537	1540 2018-09-01 12:25:00	111.275	27.075	0.285
1540	1541 2018-09-01 12:40:00	111.249	27.075	0.282
1541	1542 2018-09-01 12:55:00	111.210	27.075	0.279
1542	1543 2018-09-01 13:10:00	111.184	27.075	0.276
1543	1544 2018-09-01 13:25:00	111.208	27.075	0.279
1544	1545 2018-09-01 13:40:00	111.456	27.075	0.303
1545	1546 2018-09-01 13:55:00	112.083	27.075	0.307
1546	1547 2018-09-01 14:10:00	112.083	27.075	0.455
1547	1548 2018-09-01 14:25:00	111.784	27.075	0.538
1548	1549 2018-09-01 14:40:00	114.420	27.075	0.598
1549	1550 2018-09-01 14:55:00	115.172	27.075	0.673
1550	1551 2018-09-01 15:10:00	115.782	27.075	0.733
1551	1552 2018-09-01 15:25:00	116.330	27.075	0.789
1552	1553 2018-09-01 15:40:00	116.833	27.075	0.838
1553	1554 2018-09-01 15:55:00	117.387	27.075	0.883
1554	1555 2018-09-01 16:10:00	117.715	27.075	0.926
1555	1556 2018-09-01 16:25:00	118.053	27.075	0.959
1556	1557 2018-09-01 16:40:00	118.313	27.075	0.985
1557	1558 2018-09-01 16:55:00	118.482	27.075	1.002
1558	1559 2018-09-01 17:10:00	118.625	27.075	1.016
1559	1560 2018-09-01 17:25:00	118.676	27.075	1.021
1560	1561 2018-09-01 17:41:00	118.560	27.075	1.018
1561	1562 2018-09-01 17:56:00	118.313	27.075	0.985
1562	1563 2018-09-01 18:11:00	117.940	27.075	0.969
...

```
In [31]: print(semanasold)
```

#	Fecha	Presion	Temp	Nivel
1530	1534 2018-08-01 11:45:00	101.739	28.953	-0.480
1534	1535 2018-08-01 12:00:00	101.723	28.953	-0.483
1535	1536 2018-08-01 12:15:00	101.692	28.953	-0.480
1536	1537 2018-08-01 12:30:00	101.660	28.953	-0.480
1537	1538 2018-08-01 12:45:00	101.644	28.953	-0.411
1538	1539 2018-08-01 13:00:00	101.644	28.953	-0.411
1539	1540 2018-08-01 13:15:00	101.612	28.953	-0.414
1540	1541 2018-08-01 13:30:00	101.586	28.953	-0.414
1541	1542 2018-08-01 13:45:00	101.564	28.953	-0.419
1542	1543 2018-08-01 14:00:00	101.511	28.953	-0.421
1543	1544 2018-08-01 14:15:00	101.500	28.953	-0.425
1544	1545 2018-08-01 14:30:00	101.444	28.953	-0.427
1545	1546 2018-08-01 14:45:00	101.408	28.953	-0.428
1546	1547 2018-08-01 15:00:00	104.123	28.953	-0.383
1547	1548 2018-08-01 15:15:00	104.703	29.853	-0.385
1548	1549 2018-08-01 15:30:00	105.198	29.853	-0.250
1549	1550 2018-08-01 15:45:00	105.698	29.152	-0.286
1550	1551 2018-08-01 16:00:00	106.115	29.152	-0.105
1551	1552 2018-08-01 16:15:00	106.483	29.152	-0.128
1552	1553 2018-08-01 16:30:00	106.787	29.152	-0.898
1553	1554 2018-08-01 16:45:00	106.905	29.152	-0.877
1554	1555 2018-08-01 17:00:00	107.111	29.252	-0.866
1555	1556 2018-08-01 17:15:00	107.116	29.252	-0.859
1556	1557 2018-08-01 17:30:00	107.128	29.252	-0.864
1557	1558 2018-08-01 17:45:00	106.387	29.252	-0.888
1558	1559 2018-08-01 18:00:00	106.539	29.352	-0.123
1559	1560 2018-08-01 18:15:00	106.110	29.352	-0.163
1560	1561 2018-08-01 18:30:00	105.618	29.352	-0.215
1561	1562 2018-08-01 18:45:00	104.970	29.352	-0.279
1562	1563 2018-08-01 19:00:00	104.293	29.252	-0.348
...


```
In [32]: print(semanasal)
```

#	Fecha	Presion	Temp	Nivel
1546	1547 2018-08-01 15:15:00	33686	54525.5	28.79
1547	1548 2018-08-01 15:30:00	33686	54525.5	28.82
1548	1549 2018-08-01 15:45:00	33686	54525.5	28.83
1549	1550 2018-08-01 16:00:00	33686	54525.5	28.83
1550	1551 2018-08-01 16:15:00	33686	54525.5	28.83
1551	1552 2018-08-01 16:30:00	33686	54525.5	28.83
1552	1553 2018-08-01 16:45:00	33686	54525.5	28.83
1553	1554 2018-08-01 17:00:00	33686	54525.5	28.83
1554	1555 2018-08-01 17:15:00	33686	54525.5	28.83
1555	1556 2018-08-01 17:30:00	33686	54525.5	28.83
1556	1557 2018-08-01 17:45:00	33686	54525.5	28.85
1557	1558 2018-08-01 18:00:00	33686	54525.5	28.85
1558	1559 2018-08-01 18:15:00	33686	54525.5	28.85
1559	1560 2018-08-01 18:30:00	33686	54525.5	28.85
1560	1561 2018-08-01 18:45:00	33686	54525.5	28.83
...
2172	2173 2018-08-08 03:45:00	33686	54525.5	28.24
2173	2174 2018-08-08 04:00:00	33686	54525.5	28.24
2174	2175 2018-08-08 04:15:00	33686	54525.5	28.22
2175	2176 2018-08-08 04:30:00	33686	54525.5	28.22

En esta parte lo que hice fue juntar los *DataFrame* "canal" y "canal2" en un solo *DataFrame* que llame "canalcomp"; hice lo mismo con "old" y "old2".

```
In [33]: frames = [canal,canal2,]
canalcomp = pd.concat(frames)
canalcomp.head()
```

```
Out[33]:
```

#	Fecha	Presion	Temp	Nivel
0	1 2018-08-16 11:15:01	112.032	29.152	0.360
1	2 2018-08-16 11:30:02	111.990	27.567	0.356
2	3 2018-08-16 11:45:03	111.928	27.272	0.350
3	4 2018-08-16 12:00:04	111.899	27.173	0.347
4	5 2018-08-16 12:15:05	111.870	27.075	0.344

```
In [34]: print(canalcomp)
```

#	Fecha	Presion	Temp	Nivel
0	1 2018-08-16 11:15:01	112.032	29.152	0.360
1	2 2018-08-16 11:30:02	111.990	27.567	0.356
2	3 2018-08-16 11:45:03	111.928	27.272	0.350
3	4 2018-08-16 12:00:04	111.899	27.173	0.347
4	5 2018-08-16 12:15:05	111.870	27.075	0.344
5	6 2018-08-16 12:30:06	111.844	27.075	0.342
6	7 2018-08-16 12:45:07	111.802	26.977	0.337
7	8 2018-08-16 13:00:08	111.802	26.977	0.337
8	9 2018-08-16 13:15:09	111.789	26.977	0.336
9	10 2018-08-16 13:30:10	111.789	26.977	0.336
10	11 2018-08-16 13:45:11	111.773	26.879	0.335
11	12 2018-08-16 14:00:12	111.773	26.879	0.335
12	13 2018-08-16 14:15:13	111.747	26.879	0.332
13	14 2018-08-16 14:30:14	111.798	26.879	0.337
14	15 2018-08-16 14:45:15	112.342	26.879	0.391
15	16 2018-08-16 15:00:16	113.610	26.879	0.517
16	17 2018-08-16 15:15:17	114.081	27.173	0.654
17	18 2018-08-16 15:30:18	116.061	27.272	0.761
18	19 2018-08-16 15:45:19	116.539	27.272	0.848
19	20 2018-08-16 16:00:20	117.511	27.173	0.905
20	21 2018-08-16 16:15:21	118.085	27.173	0.955
21	22 2018-08-16 16:30:22	118.450	27.075	0.999
22	23 2018-08-16 16:45:23	118.804	27.075	1.042
23	24 2018-08-16 17:00:24	119.222	27.075	1.076
24	25 2018-08-16 17:15:25	119.121	27.075	1.105
25	26 2018-08-16 17:30:26	119.755	27.075	1.129
26	27 2018-08-16 17:45:27	119.973	26.977	1.158
27	28 2018-08-16 18:00:28	120.142	26.977	1.187
28	29 2018-08-16 18:15:29	120.877	26.977	1.161
29	30 2018-08-16 18:30:30	119.620	26.879	1.116
...

```
In [35]: frames45 = [old,old2,]
oldcomp = pd.concat(frames45)
oldcomp.head()
```

```
Out[35]:
```

	#	Fecha	Presion	Temp	Nivel
0	1	2018-08-16 12:30:00	104.359	29.252	-0.340
1	2	2018-08-16 12:45:00	104.243	28.754	-0.351
2	3	2018-08-16 13:00:00	104.255	28.655	-0.350
3	4	2018-08-16 13:15:00	104.239	28.655	-0.352
4	5	2018-08-16 13:30:00	104.255	28.655	-0.350

```
In [36]: print(oldcomp)
```

#	Fecha	Presion	Temp	Nivel
0	1 2018-08-16 12:30:00	104.359	29.252	-0.340
1	2 2018-08-16 12:45:00	104.243	28.754	-0.351
2	3 2018-08-16 13:00:00	104.255	28.655	-0.350
3	4 2018-08-16 13:15:00	104.239	28.655	-0.352
4	5 2018-08-16 13:30:00	104.255	28.655	-0.350
5	6 2018-08-16 13:45:00	104.251	28.555	-0.350
6	7 2018-08-16 14:00:00	104.251	28.555	-0.350
7	8 2018-08-16 14:15:00	104.235	28.555	-0.352
8	9 2018-08-16 14:30:00	104.235	28.555	-0.354
9	10 2018-08-16 14:45:00	104.187	28.555	-0.357
10	11 2018-08-16 15:00:00	104.150	28.555	-0.360
11	12 2018-08-16 15:15:00	104.140	28.555	-0.361
12	13 2018-08-16 15:30:00	104.790	28.555	-0.296
13	14 2018-08-16 15:45:00	105.595	28.555	-0.217
14	15 2018-08-16 16:00:00	100.375	28.655	-0.339
15	16 2018-08-16 16:15:00	106.671	28.655	-0.188
16	17 2018-08-16 16:30:00	107.104	28.655	-0.067
17	18 2018-08-16 16:45:00	107.536	28.655	0.024
18	19 2018-08-16 17:00:00	107.980	28.655	0.013
19	20 2018-08-16 17:15:00	108.140	28.754	0.037
20	21 2018-08-16 17:30:00	108.358	28.754	0.058
21	22 2018-08-16 17:45:00	108.500	28.754	0.077
22	23 2018-08-16 18:00:00	108.678	28.754	0.090
23	24 2018-08-16 18:15:00	108.554	28.853	0.078
24	25 2018-08-16 18:30:00	108.281	28.853	0.051
25	26 2018-08-16 18:45:00	107.897	28.853	0.012
26	27 2018-08-16 19:00:00	107.448	28.853	-0.012
27	28 2018-08-16 19:15:00	106.909	28.853	-0.077
28	29 2018-08-16 19:30:00	106.430	28.853	-0.133
29	30 2018-08-16 19:45:00	105.799	28.853	-0.196

gráfica de niveles de mar del sensor junto a la estación y el

En esta seccion pongo el codigo utilizado para generar las graficas. En el eje X va el tiempo y en el Y los niveles del mar en metros corresspondientes a el canal y el sargento.

```
In [41]: Run día fuera de línea
import plotly
import plotly.graph_objs as go

import plotly.offline as offline
import plotly.graph_objs as go
plotly.offline.init_notebook_mode(connected=True)

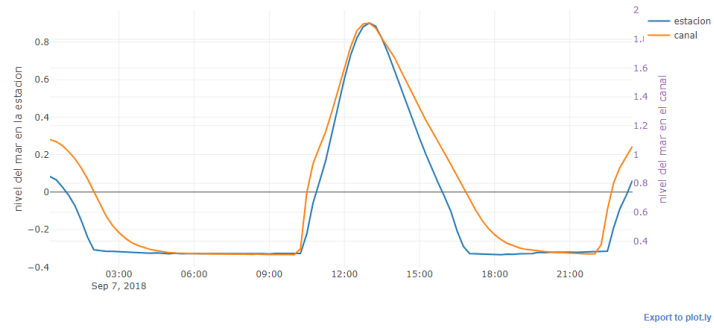
trace1 = go.Scatter(
    x=diabla['fecha'],
    y=diabla['nivel'],
    name='estacion',
)

trace2 = go.Scatter(
    x=diabla['fecha'],
    y=diabla['nivel'],
    name='canal',
    yAxis='y2'
)

data = [trace1, trace2]
layout = go.Layout(
    title='Niveles de mar del sensor junto a la estación y el del canal de un día',
    yaxis=dict(
        title='nivel del mar en la estacion'
    ),
    yaxis2=dict(
        title='nivel del mar en el canal',
        color='rgb(148, 103, 189)'
    ),
    tickfont=dict(
        color='rgb(148, 103, 189)'
    ),
    overlaying='y',
    side='right'
)

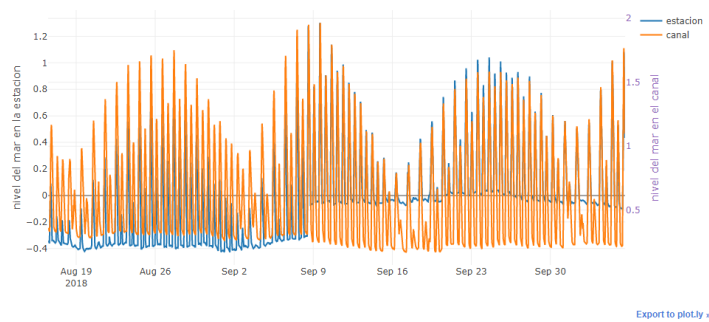
fig = go.Figure(data=data, layout=layout)
plot_url = offline.plot(fig, filename='multiple-axes-double')
```

niveles de mar del sensor junto a la estación y el del canal de un día

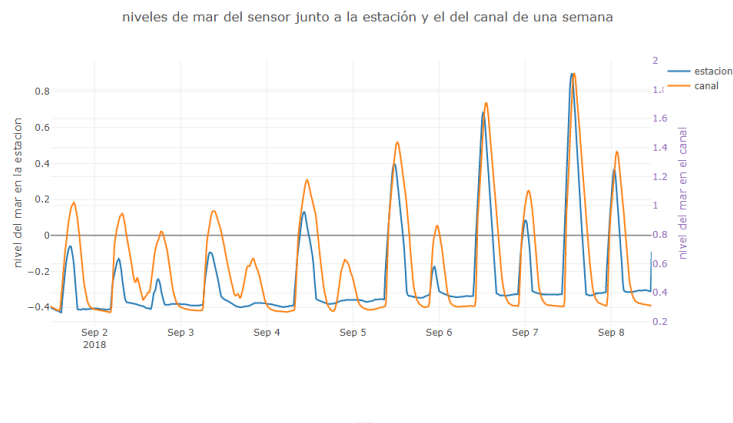


```
In [42]: #los dos archivos juntos fuera de línea
import plotly
import plotly.graph_objs as go
import plotly.offline as offline
import plotly.graph_objs as go
plotly.offline.init_notebook_mode(connected=True)
trace1 = go.Scatter(
    x=oldcomp['fecha'],
    y=oldcomp['nivel'],
    name='estacion',
)
trace2 = go.Scatter(
    x=oldcomp['fecha'],
    y=canalcomp['nivel'],
    name='canal',
    yaxis='y2'
)
data = [trace1, trace2]
layout = go.Layout(
    title='niveles de mar del sensor junto a la estación y el del canal dos archivos',
    yaxis=dict(
        title='nivel del mar en la estación'
    ),
    yaxis2=dict(
        title='nivel del mar en el canal',
        titlefont=dict(
            color='rgb(148, 183, 189)'
        ),
        tickfont=dict(
            color='rgb(148, 183, 189)'
        ),
        overlaying='y',
        side='right'
    )
)
fig = go.Figure(data=data, layout=layout)
plot_url = offline.iplot(fig, filename='multiple-axes-double')
```

niveles de mar del sensor junto a la estación y el del canal dos archivos



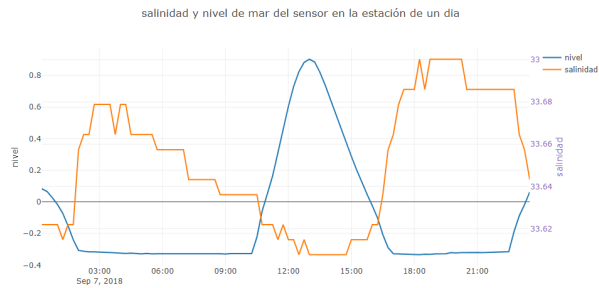
```
In [68]: #run sensor offline
import plotly
import plotly.graph_objs as go
import plotly.offline as offline
import plotly.graph_objs as go
plotly.offline.init_notebook_mode(connected=True)
traces = go.Scatter(
    x=semanaal['fecha'],
    y=semanaal['nivel'],
    name='estacion',
)
traces2 = go.Scatter(
    x=semanaal['fecha'],
    y=semanacanal['nivel'],
    name='canal',
    yAxis='y2'
)
data = [traces, traces2]
layout = go.Layout(
    title='niveles de mar del sensor junto a la estación y el del canal de una semana',
    yaxis=dict(
        title='nivel del mar en la estación'
    ),
    yaxis2=dict(
        title='nivel del mar en el canal',
        titlefont=dict(
            color='rgb(148, 103, 189)'
        ),
        tickfont=dict(
            color='rgb(148, 103, 189)'
        ),
        overlaying='y',
        side='right'
    )
)
fig = go.Figure(data=data, layout=layout)
plot_url = offline.iplot(fig, filename='multiple-axes-double')
```



En este apartado podemos observar el código utilizado para generar las gráficas de salinidad contra el nivel del mar que llegaba en la estación.

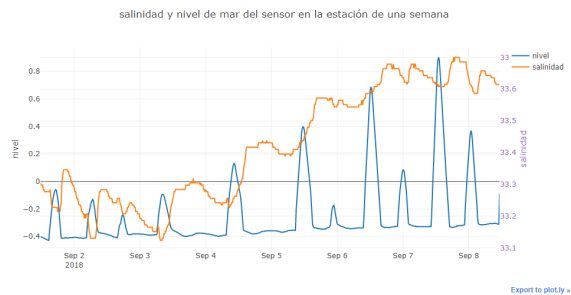
```
#run dia offline
import plotly.plotly as py
import plotly.graph_objs as go

traces = go.Scatter(
    x=diad['fecha'],
    y=diad['nivel'],
    name='nivel',
)
traces2 = go.Scatter(
    x=diad['fecha'],
    y=diad['sal_ppm'],
    name='salinidad',
    yAxis='y2'
)
data = [traces, traces2]
layout = go.Layout(
    title='salinidad y nivel de mar del sensor en la estación de un día',
    yaxis=dict(
        title='nivel'
    ),
    yaxis2=dict(
        title='salinidad',
        titlefont=dict(
            color='rgb(148, 103, 189)'
        ),
        tickfont=dict(
            color='rgb(148, 103, 189)'
        ),
        overlaying='y',
        side='right'
    )
)
fig = go.Figure(data=data, layout=layout)
plot_url = offline.iplot(fig, filename='multiple-axes-double')
```



```
In [46]: Run semana offline
import plotly.offline as go
import plotly.graph_objs as go

trace1 = go.Scatter(
    x=semana['fecha'],
    y=semana['nivel'],
    name='nivel'
)
trace2 = go.Scatter(
    x=semana['fecha'],
    y=semana['sal'],
    name='salinidad',
    yaxis='y2'
)
data = [trace1, trace2]
layout = go.Layout(
    title='salinidad y nivel de mar del sensor en la estación de una semana',
    yaxis=dict(
        title='nivel'
    ),
    yaxis2=dict(
        title='salinidad',
        titlefont=dict(
            color='rgb(148, 181, 189)'
        ),
        tickfont=dict(
            color='rgb(148, 181, 189)'
        ),
        overlaying='y',
        side='right'
    )
)
fig = go.Figure(data=data, layout=layout)
plot_url = offline.plot(fig, filename='multiple-axes-double')
```

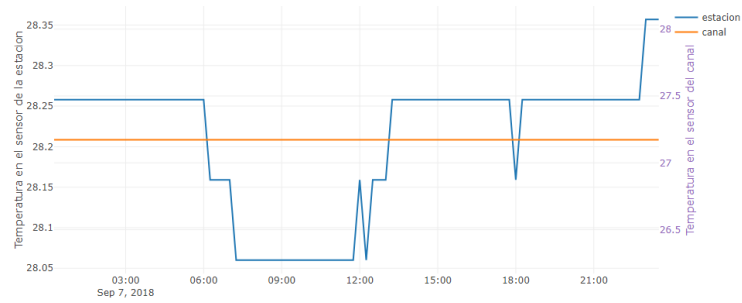


Gráfica de temperatura de agua entre el sensor junto a la estación y el del canal.

Se muestra el código que se utilizó seguido por la gráfica

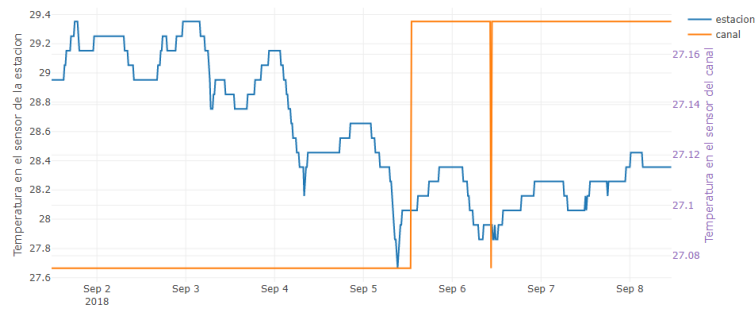
```
In [39]: Run día fuera de línea
import plotly
import plotly.graph_objs as go
import plotly.offline as offline
import plotly.graph_objs as go
plotly.offline.init_notebook_mode(connected=True)
trace1 = go.Scatter(
    x=dia['fecha'],
    y=dia['temp'],
    name='estacion',
)
trace2 = go.Scatter(
    x=dia['fecha'],
    y=dia['temp'],
    name='canal',
)
data = [trace1, trace2]
layout = go.Layout(
    title='Gráfica de temperatura de agua entre el sensor junto a la estación y el del canal de un día',
    yaxis=dict(
        title='temperatura en el sensor de la estación'
    ),
    yaxis2=dict(
        title='temperatura en el sensor del canal',
        color='rgb(148, 181, 189)'
    ),
    tickfont=dict(
        color='rgb(148, 181, 189)'
    ),
    overlaying='y',
    side='right'
)
fig = go.Figure(data=data, layout=layout)
plot_url = offline.plot(fig, filename='multiple-axes-double')
```

Gráfica de temperatura de agua entre el sensor junto a la estación y el del canal de un día



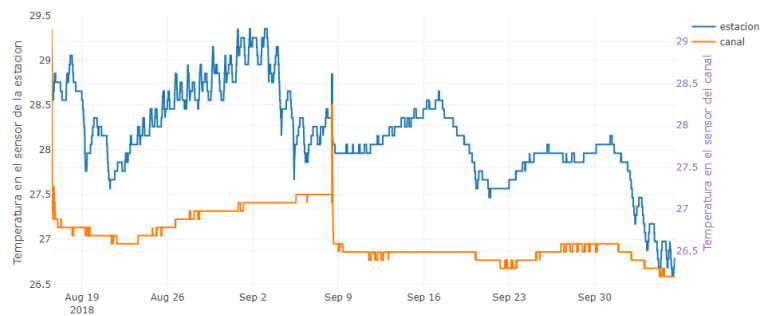
```
In [71]: #una semana de fuera línea
import plotly
import plotly.graph_objs as go
import plotly.offline as offline
import plotly.graph_objs as go
plotly.offline.init_notebook_mode(connected=True)
trace1 = go.Scatter(
    x=senanald['fecha'],
    y=senanald['temp'],
    name='estacion',
)
trace2 = go.Scatter(
    x=senanald['fecha'],
    y=senanald['temp'],
    name='canal',
)
axis='y2'
data = [trace1, trace2]
layout = go.Layout(
    title='Gráfica de temperatura de agua entre el sensor junto a la estación y el del canal de una semana',
    yaxis=dict(
        title='Temperatura en el sensor de la estacion'
    ),
    yaxis2=dict(
        title='Temperatura en el sensor del canal',
        titlefont=dict(
            color='rgb(148, 183, 189)'
        ),
        tickfont=dict(
            color='rgb(148, 183, 189)'
        ),
        overlaying='y',
        side='right'
    )
)
fig = go.Figure(data=data, layout=layout)
plot_url = offline.plot(fig, filename='multiple-axes-double')
```

Gráfica de temperatura de agua entre el sensor junto a la estación y el del canal de una semana



```
In [45]: #dos archivos fuera de línea
import plotly
import plotly.graph_objs as go
import plotly.offline as offline
import plotly.graph_objs as go
plotly.offline.init_notebook_mode(connected=True)
trace1 = go.Scatter(
    x=oldcomp['fecha'],
    y=oldcomp['temp'],
    name='estacion',
)
trace2 = go.Scatter(
    x=oldcomp['fecha'],
    y=canalcomp['temp'],
    name='canal',
    yaxis='y2'
)
data = [trace1, trace2]
layout = go.Layout(
    title='Gráfica de temperatura de agua entre el sensor junto a la estación y el del canal dos archivos',
    yaxis=dict(
        title='Temperatura en el sensor de la estación'
    ),
    yaxis2=dict(
        title='Temperatura en el sensor del canal',
        titlefont=dict(
            color='rgb(148, 183, 189)'
        ),
        tickfont=dict(
            color='rgb(148, 183, 189)'
        ),
        overlaying='y',
        side='right'
    )
)
fig = go.Figure(data=data, layout=layout)
plot_url = offline.plot(fig, filename='multiple-axes-double')
```

Gráfica de temperatura de agua entre el sensor junto a la estación y el del canal dos archivos



correlación entre el nivel de agua y la salinidad: correlation matrix python heat map

En este apartado se procede a crear un mapa de calor donde se utilicen las variables que utilizamos anteriormente para saber la correlacion que hay entre cada una de ellas.

Comence por crear una matris que contenga los datos que queria para un dia.


```
In [37]: dia= {'salinidad':diasal['sal, ppt'] ,
'conductacia especifica':diasal['conductancia especifica, s/cm,'],
'Temp': diasal['Temp'],
'Nivel en la estacion':diaold['Nivel'],
'Temp de la estacion': diaold['Temp'],
'presion en la estacion':diaold['Presion'],
'Nivel en el canal':diacanal['Nivel'],
'Temp en el canal': diacanal['Temp'],
'presion en el canal':diacanal['Presion']}
Dia = pd.DataFrame(data=dia)
Dia
```

Out[37]:

	salinidad	conductacia especifica	Temp	Nivel en la estacion	Temp de la estacion	presion en la estacion	Nivel en el canal	Temp en el canal	presion en el canal
2062	33.6219	51201.8	28.27	NaN	NaN	NaN	NaN	NaN	NaN
2063	33.6219	51201.8	28.27	0.083	28.258	108.610	NaN	NaN	NaN
2064	33.6219	51201.8	28.27	0.066	28.258	108.434	NaN	NaN	NaN
2065	33.6219	51201.8	28.27	0.027	28.258	108.049	NaN	NaN	NaN
2066	33.6148	51192.2	28.28	-0.017	28.258	107.600	1.103	27.173	118.499
2067	33.6219	51201.8	28.27	-0.073	28.258	107.640	1.090	27.173	118.499
2068	33.6219	51201.8	28.27	-0.151	28.258	106.255	1.062	27.173	118.063
2069	33.6574	51250.9	28.22	-0.240	28.258	105.359	1.020	27.173	118.607
2070	33.6646	51259.7	28.21	-0.307	28.258	104.888	0.970	27.173	118.160
2071	33.6646	51259.7	28.21	-0.312	28.258	104.640	0.907	27.173	117.524
2072	33.6788	51279.9	28.19	-0.318	28.258	104.592	0.832	27.173	116.771
2073	33.6788	51279.9	28.19	-0.318	28.258	104.592	0.744	27.173	115.889
2074	33.6788	51279.9	28.19	-0.318	28.258	104.576	0.654	27.173	114.981
2075	33.6788	51279.9	28.19	-0.320	28.258	104.560	0.571	27.173	114.152
2076	33.6646	51259.7	28.21	-0.321	28.258	104.544	0.508	27.173	113.517
2077	33.6788	51279.9	28.19	-0.324	28.258	104.512	0.459	27.173	113.025
2078	33.6788	51279.9	28.19	-0.324	28.258	104.512	0.421	27.173	112.637
2079	33.6646	51259.7	28.21	-0.326	28.258	104.496	0.390	27.173	112.326
2080	33.6646	51259.7	28.21	-0.324	28.258	104.512	0.370	27.173	112.132
2081	33.6646	51259.7	28.21	-0.328	28.258	104.480	0.354	27.173	111.964
2082	33.6646	51259.7	28.21	-0.329	28.258	104.484	0.342	27.173	111.847

Este renglon toma la matriz creada con los datos de un dia y saca la correlacion lineal que existe entre las columnas y renglones.

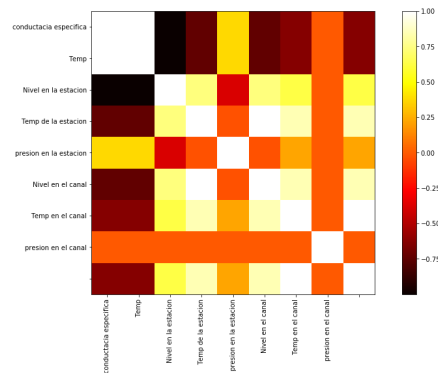
In [38]: Dia.corr()

Out[38]:

	salinidad	conductacia especifica	Temp	Nivel en la estacion	Temp de la estacion	presion en la estacion	Nivel en el canal	Temp en el canal	presion en el canal
salinidad	1.000000e+00	9.999998e-01	-9.999994e-01	-7.384638e-01	3.780866e-01	-7.385280e-01	-6.294800e-01	-1.048046e-12	-6.293958e-01
conductacia especifica	9.999998e-01	1.000000e+00	-9.999997e-01	-7.386937e-01	3.779267e-01	-7.387580e-01	-6.296975e-01	-1.523137e-13	-6.296134e-01
Temp	-9.999994e-01	-9.999997e-01	1.000000e+00	7.388517e-01	-3.778820e-01	7.389160e-01	6.298912e-01	6.616442e-13	6.298071e-01
Nivel en la estacion	-7.384638e-01	-7.386937e-01	7.388517e-01	1.000000e+00	-2.763648e-02	9.999996e-01	8.481529e-01	1.542314e-16	8.481531e-01
Temp de la estacion	3.780866e-01	3.779267e-01	-3.778820e-01	-2.763648e-02	1.000000e+00	-2.773116e-02	2.237263e-01	3.756704e-13	2.238163e-01
presion en la estacion	-7.385280e-01	-7.387580e-01	7.389160e-01	9.999996e-01	-2.773116e-02	1.000000e+00	8.481471e-01	1.372186e-14	8.481474e-01
Nivel en el canal	-6.294800e-01	-6.296975e-01	6.298912e-01	8.481529e-01	2.237263e-01	8.481471e-01	1.000000e+00	-6.078504e-17	9.999998e-01
Temp en el canal	-1.048046e-12	-1.523137e-13	6.616442e-13	1.542314e-16	3.756704e-13	1.372186e-14	-6.078504e-17	1.000000e+00	1.249665e-14
presion en el canal	-6.293958e-01	-6.296134e-01	6.298071e-01	8.481531e-01	2.238163e-01	8.481474e-01	9.999998e-01	1.249665e-14	1.000000e+00

Este renglon de codigo es el necesario para formar un mapa de calor

```
In [32]: fig = plt.figure(figsize=(15,8))
ax1 = fig.add_subplot(111)
plt.imshow(Dia.corr(), cmap='hot', interpolation='nearest')
plt.colorbar()
labels = Dia.columns.tolist()
ax1.set_xticklabels(labels,rotation=90, fontsize=10)
ax1.set_yticklabels(labels,fontsize=10)
plt.show()
```



A continuacion se muestra el mismo procedimiento aplicado para los datos de una semana.

```
In [72]: semana= {'salinidad':semanasal['sal, ppt'] ,
                  'conductancia especifica':semanasal['conductancia especifica, s/cm,'],
                  'Temp': semanasal['Temp'],
                  'Nivel en la estacion':semanaold['Nivel'],
                  'Temp de la estacion': semanaold['Temp'],
                  'presion en la estacion':semanaold['Presion'],
                  'Nivel en el canal':semanacanal['Nivel'],
                  'Temp en el canal': semanacanal['Temp'],
                  'presion en el canal':semanacanal['Presion']}]
Semana= pd.DataFrame(data=semana)
Semana
```

Out[72]:

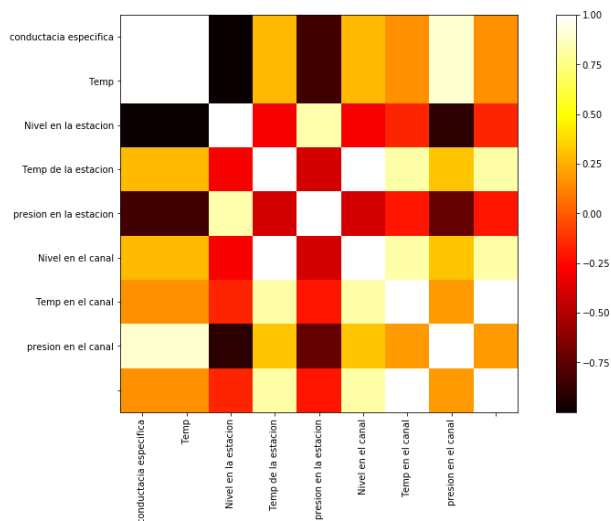
	salinidad	conductancia especifica	Temp	Nivel en la estacion	Temp de la estacion	presion en la estacion	Nivel en el canal	Temp en el canal	presion en el canal
1531	33.3047	50771.4	28.72	NaN	NaN	NaN	NaN	NaN	NaN
1532	33.2977	50761.9	28.73	NaN	NaN	NaN	NaN	NaN	NaN
1533	33.2977	50761.9	28.73	-0.401	28.953	103.739	0.307	27.075	111.495
1534	33.2977	50761.9	28.73	-0.403	28.953	103.723	0.300	27.075	111.430
1535	33.2837	50742.9	28.75	-0.406	28.953	103.692	0.298	27.075	111.404
1536	33.2837	50742.9	28.75	-0.409	28.953	103.660	0.295	27.075	111.378
1537	33.2768	50733.4	28.76	-0.411	28.953	103.644	0.291	27.075	111.339
1538	33.2768	50733.4	28.76	-0.411	28.953	103.644	0.288	27.075	111.301
1539	33.2768	50733.4	28.76	-0.414	28.953	103.612	0.285	27.075	111.275
1540	33.2768	50733.4	28.76	-0.416	28.953	103.596	0.282	27.075	111.249
1541	33.2768	50733.4	28.76	-0.419	28.953	103.564	0.279	27.075	111.210
1542	33.2768	50733.4	28.76	-0.422	28.953	103.532	0.276	27.075	111.184
1543	33.2768	50733.4	28.76	-0.425	28.953	103.500	0.279	27.075	111.210
1544	33.2768	50733.4	28.76	-0.427	28.953	103.484	0.303	27.075	111.456
1545	33.2837	50742.9	28.75	-0.428	28.953	103.468	0.367	27.075	112.103
1546	33.2558	50705.0	28.79	-0.363	28.953	104.123	0.455	27.075	112.983
1547	33.2349	50676.5	28.82	-0.305	29.053	104.703	0.530	27.075	113.734
1548	33.2279	50667.1	28.83	-0.256	29.053	105.198	0.598	27.075	114.420
1549	33.2279	50667.1	28.83	-0.206	29.152	105.698	0.673	27.075	115.172
1550	33.2279	50667.1	28.83	-0.165	29.152	106.115	0.733	27.075	115.782

In [74]: `Semana.corr()`

Out[74]:

	salinidad	conductancia especifica	Temp	Nivel en la estacion	Temp de la estacion	presion en la estacion	Nivel en el canal	Temp en el canal	presion en el canal
salinidad	1.000000	1.000000	-0.999995	0.285846	-0.830495	0.285873	0.156666	0.905845	0.156611
conductancia especifica	1.000000	1.000000	-0.999996	0.285824	-0.830539	0.285851	0.156657	0.905766	0.156602
Temp	-0.999995	-0.999996	1.000000	-0.285811	0.831044	-0.285839	-0.156693	-0.905152	-0.156638
Nivel en la estacion	0.285846	0.285824	-0.285811	1.000000	-0.392996	0.999999	0.824701	0.316856	0.824679
Temp de la estacion	-0.830495	-0.830539	0.831044	-0.392996	1.000000	-0.392996	-0.208843	-0.731134	-0.208801
presion en la estacion	0.285873	0.285851	-0.285839	0.999999	-0.392996	1.000000	0.824697	0.316879	0.824675
Nivel en el canal	0.156666	0.156657	-0.156693	0.824701	-0.208843	0.824697	1.000000	0.193554	1.000000
Temp en el canal	0.905845	0.905766	-0.905152	0.316856	-0.731134	0.316879	0.193554	1.000000	0.193529
presion en el canal	0.156611	0.156602	-0.156638	0.824679	-0.208801	0.824675	1.000000	0.193529	1.000000

```
In [75]: fig = plt.figure(figsize=(15,8))
ax1 = fig.add_subplot(111)
plt.imshow(Semana.corr(), cmap='hot', interpolation='nearest')
plt.colorbar()
labels = Semana.columns.tolist()
ax1.set_xticklabels(labels,rotation=90, fontsize=10)
ax1.set_yticklabels(labels,fontsize=10)
plt.show()
```



4 Conclusion

Después de analizar las gráficas para el canal y la estación podemos decir que el estudio revelo que el 7 de septiembre la marea alta fue de 0.8m, si la comparamos con la marea alta de esa semana, correspondiente del 2 de septiembre al 8, fue uno de los días más que altura tuvo la marea más alta, reflejando que esa semana se presentó un comportamiento creciente, lo cual concuerda con el calendario lunar de esa fecha, que dice que el 2 de septiembre fue luna nueva y durante esa semana se llenaba. En los dos meses de datos recopilados de agosto y septiembre, tenemos que la marea más alta fue de 1.2m, y durante esos dos meses presento un comportamiento de acuerdo a los esperado, por lo que podemos concluir que el tipo de marea que se presenta en esta zona es micromareal. Se encontró que el nivel de salinidad tiene correlación lineal negativo con respecto al nivel del mar.

La temperatura del agua en la estación es mucho más variante que la del canal, pienso que esto se debe a que como la de la estación esta aparte en el manglar y casi no llega el sol, favorece el cambio de temperatura.

References

- [1] TERMINOLOGIA DE MAREAS <https://es.wikipedia.org/wiki/Marea>.