

Reporte de la Evaluación 1

Rolando A. Fimbres G.

8 de marzo de 2018

1. Descripción

Se tiene una estación de monitoreo de variables atmosféricas, CO₂, radiación solar, nivel de agua y salinidad en el Manglar El Sargento, en una bahía en la costa frente a la parte norte de la Isla Tiburón.

Nos interesa explorar los datos de Febrero de 2018 de nivel de mar, salinidad y temperatura del agua. Se proporcionan los datos de cada 15 minutos en formato CSV, que puedes descargar. Nivel de agua: sargento-270218.csv, Salinidad: sargento-salinidad-270218.csv.

Se te pide que por favor que crees una carpeta llamada Evaluacion1 en tu espacio de trabajo de Física Computacional 1. Todas las actividades las desarrollarás desde allí.

Descarga los archivos de datos de El Sargento a tu carpeta Evaluacion1.

Inicia un archivo para el reporte en LaTeX para esta Actividad, donde irás describiendo las actividades realizadas y productos generados.

2. Actividades a Realizar

1.- Usa los comandos de Linux y Emacs para que los 2 archivos abarquen el mismo periodo de tiempo (mismo rango fechas, mismo número de renglones). Describe brevemente la estructura de los archivos. Cada archivo es generado por un sensor distinto.

2.- Ejecuta tu Jupyter Notebook, para leer los dos archivos de datos con el apoyo de Pandas, como lo has hecho con anterioridad. No olvides cargar Pandas, Numpy, Pyplot de Matplotlib y la biblioteca datetime.

3.- Con la ayuda de la biblioteca Seaborn, por favor crea un gráfica de caja (boxplot) para visualizar la variabilidad de los datos de Febrero: a) Nivel de mar (metros); b) Salinidad (Partes por mil - ppt); y c) Temperatura de Agua (°C). Con la ayuda de la función describe puedes saber con exactitud la posición de la mediana, cuarteles, máximos y mínimos.

4.- De nuevo con la ayuda de Seaborn, también explora si hay una correlación de Pearson entre cada pareja de variables (Regresión lineal con las distribuciones marginales): Nivel de mar-Salinidad, Nivel de mar-Temperatura del agua,

Salinidad-Temperatura del agua.

5.- Con la ayuda de Matplotlib, realice ahora 3 gráficas independientes de las variables: Nivel del mar, Salinidad y Temperatura del Agua, para ver su variabilidad como función del tiempo.

6.- Enseguida, de igual forma, produce gráficas superpuestas con doble eje vertical (izquierda, derecha): a) Nivel de mar y Salinidad; b) Nivel de mar y Temperatura.

7.- Con ayuda de la función `xlim` de `pyplot`, analiza las gráficas del punto anterior para 5 días y trata de explicar si hay o no una clara manifestación de dependencia de Salinidad y Nivel de mar o de Nivel de Mar y Temperatura del agua.

8.- Completa el reporte en LaTeX de la actividad realizada, con segmentos de código y figuras producidas, para después descargar el reporte de actividad en formato PDF a tu carpeta.

9.- Como es costumbre sube tu carpeta Evaluacion1 a tu repositorio en Github.

3. Procedimientos Realizados

Tras crear la carpeta de la evaluación y descargar los archivos en ella, analicé ambos con `emacs`. Limpié los primeros renglones de ambos para así trabajar solo con números. Después noté que el archivo `csv` de salinidad comenzaba sus mediciones quince minutos antes y después que el otro archivo. Eliminé ambas filas y continué el análisis en `jupyter`.

Primero cargué los paquetes a utilizar:

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

Después leí cada archivo `csv` por separado.

```
In [3]: df1 = pd.read_csv("sargento-270218.csv", header=None, names=['Date',
df1.AP=pd.to_numeric(df1.AP, errors='coerce')
df1.TEMP=pd.to_numeric(df1.TEMP, errors='coerce')
df1.WL=pd.to_numeric(df1.WL, errors='coerce')
df1.head()
```

```
Out[3]:
```

	Date	AP	TEMP	WL
1	02/04/2018 09:45:00	108.068	17.094	0.060
2	02/04/2018 10:00:00	107.815	16.903	0.035
3	02/04/2018 10:15:00	107.791	16.903	0.032
4	02/04/2018 10:30:00	107.791	16.903	0.032
5	02/04/2018 10:45:00	107.791	16.903	0.032

```
In [9]: df2 = pd.read_csv("sargento-salinidad-270218.csv", header=None, names
df2.CH=pd.to_numeric(df2.CH, errors='coerce')
df2.TEMP=pd.to_numeric(df2.TEMP, errors='coerce')
df2.SC=pd.to_numeric(df2.SC, errors='coerce')
df2.SAL=pd.to_numeric(df2.SAL, errors='coerce')
df2.head()
```

```
Out[9]:
```

	Date	CH	TEMP	SC	SAL
2	02/04/2018 09:45:00	50617.2	17.33	59362.4	39.7235
3	02/04/2018 10:00:00	50646.3	17.23	59526.8	39.8482
4	02/04/2018 10:15:00	50772.9	17.17	59753.7	40.0203
5	02/04/2018 10:30:00	51007.8	17.14	60068.6	40.2595
6	02/04/2018 10:45:00	50958.7	17.11	60050.6	40.2458

Después numeré el mes de cada archivo.

```
In [18]: df1['Ndate'] = pd.to_datetime(df1['Date'], format='%m/%d/%Y %H:%M:%S')
df1['month'] = df1['Ndate'].dt.month
df1.head()
```

```
Out[18]:
```

	Date	AP	TEMP	WL	Ndate	month
1	02/04/2018 09:45:00	108.068	17.094	0.060	2018-02-04 09:45:00	2
2	02/04/2018 10:00:00	107.815	16.903	0.035	2018-02-04 10:00:00	2
3	02/04/2018 10:15:00	107.791	16.903	0.032	2018-02-04 10:15:00	2
4	02/04/2018 10:30:00	107.791	16.903	0.032	2018-02-04 10:30:00	2
5	02/04/2018 10:45:00	107.791	16.903	0.032	2018-02-04 10:45:00	2

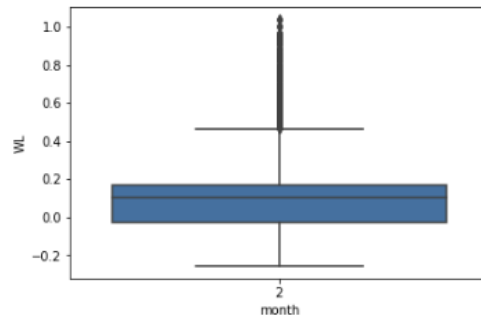
```
In [19]: df2['Ndate'] = pd.to_datetime(df2['Date'], format='%m/%d/%Y %H:%M:%S')
df2['month'] = df2['Ndate'].dt.month
df2.head()
```

```
Out[19]:
```

	Date	CH	TEMP	SC	SAL	Ndate	month
2	02/04/2018 09:45:00	50617.2	17.33	59362.4	39.7235	2018-02-04 09:45:00	2
3	02/04/2018 10:00:00	50646.3	17.23	59526.8	39.8482	2018-02-04 10:00:00	2
4	02/04/2018 10:15:00	50772.9	17.17	59753.7	40.0203	2018-02-04 10:15:00	2
5	02/04/2018 10:30:00	51007.8	17.14	60068.6	40.2595	2018-02-04 10:30:00	2
6	02/04/2018 10:45:00	50958.7	17.11	60050.6	40.2458	2018-02-04 10:45:00	2

Siguió hacer tres gráficas de caja; mes contra nivel del mar, mes contra salinidad y mes contra temperatura.

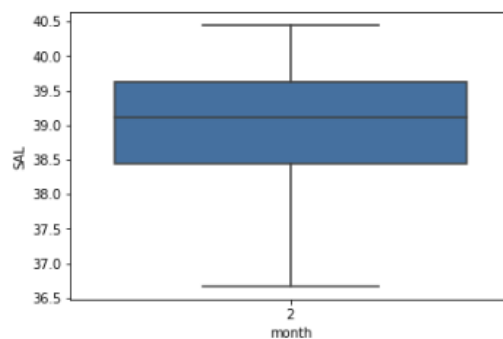
```
In [21]: import seaborn as sns
ax = sns.boxplot(x="month", y="WL", data=df1)
plt.show()
df1.describe()
```



```
Out[21]:
```

	AP	TEMP	WL	month
count	2216.000000	2216.000000	2216.000000	2216.0
mean	108.472155	16.676223	0.100208	2.0
std	2.170574	0.756794	0.215948	0.0
min	104.878000	14.230000	-0.257000	2.0
25%	107.199750	16.332000	-0.026000	2.0
50%	108.488000	16.713000	0.102000	2.0
75%	109.178000	17.189000	0.170000	2.0
max	117.899000	17.855000	1.038000	2.0

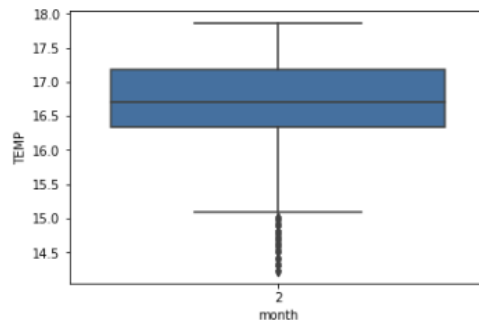
```
In [22]: import seaborn as sns
ax = sns.boxplot(x="month", y="SAL", data=df2)
plt.show()
df2.describe()
```



Out[22]:

	CH	TEMP	SC	SAL	month
count	2216.000000	2216.000000	2216.000000	2216.000000	2216.0
mean	49435.399819	16.902455	58526.228520	39.091686	2.0
std	1147.675142	0.670285	922.339947	0.697235	0.0
min	46569.900000	14.100000	55305.400000	36.669100	2.0
25%	48410.500000	16.530000	57655.400000	38.433300	2.0
50%	49294.000000	16.850000	58557.450000	39.114250	2.0
75%	50068.200000	17.390000	59244.200000	39.634000	2.0
max	52095.300000	18.100000	60306.400000	40.440200	2.0

```
In [23]: import seaborn as sns
ax = sns.boxplot(x="month", y="TEMP", data=df1)
plt.show()
df1.describe()
```



```
Out[23]:
```

	AP	TEMP	WL	month
count	2216.000000	2216.000000	2216.000000	2216.0
mean	108.472155	16.676223	0.100208	2.0
std	2.170574	0.756794	0.215948	0.0
min	104.878000	14.230000	-0.257000	2.0
25%	107.199750	16.332000	-0.026000	2.0
50%	108.488000	16.713000	0.102000	2.0
75%	109.178000	17.189000	0.170000	2.0
max	117.899000	17.855000	1.038000	2.0

Ahora se pide una gráfica de regresión lineal que relacione la salinidad con el nivel del mar. Por ésto mismo es necesario crear un nuevo data frame que relacione ambas variables.

```
In [50]: df3 = pd.concat([df1['WL'], df2['SAL']], axis=1)
df3.head()
```

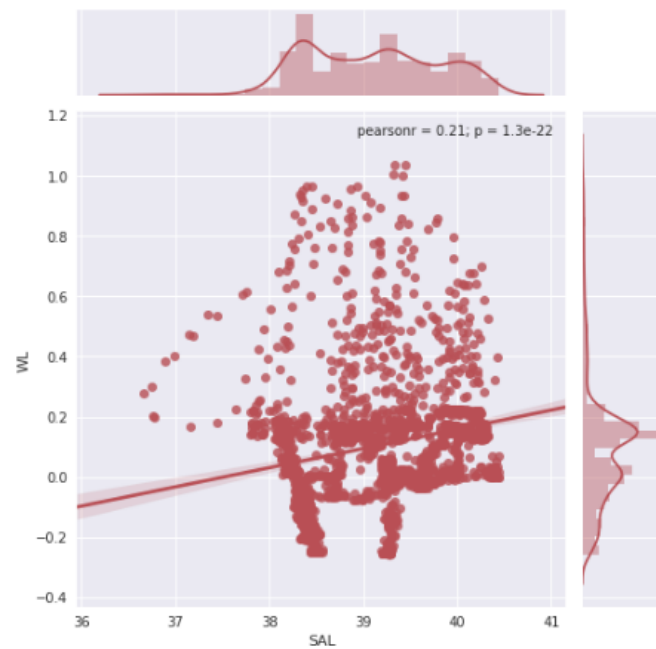
```
Out[50]:
```

	WL	SAL
1	0.060	NaN
2	0.035	39.7235
3	0.032	39.8482
4	0.032	40.0203
5	0.032	40.2595

Después realizamos la grafica.

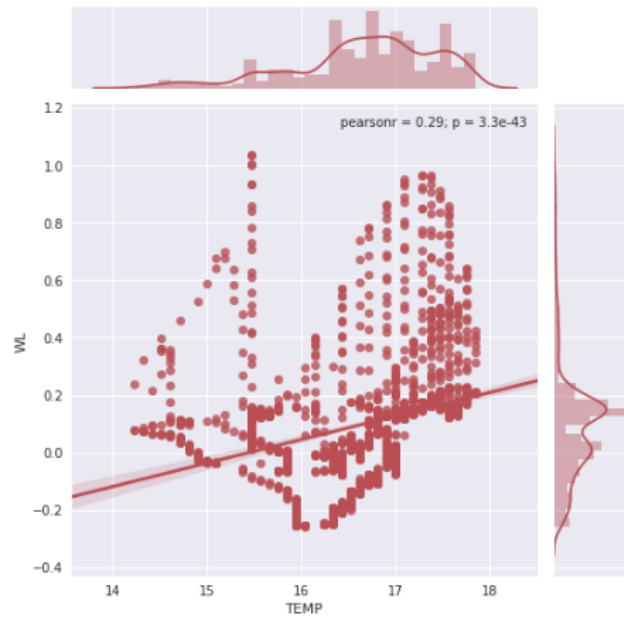
```
In [55]: import seaborn as sns
sns.set(style="darkgrid", color_codes=True)

g = sns.jointplot("SAL", "WL", data=df3, kind="reg",
                  color="r", size=7)
plt.show(g)
```



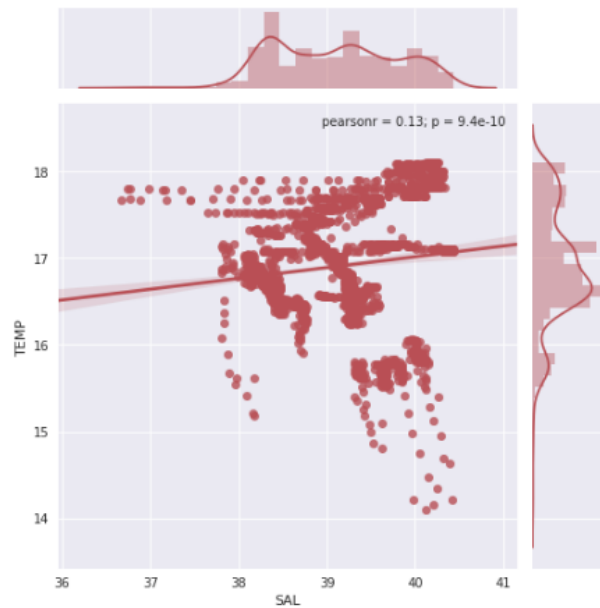
Ahora la segunda que relaciona nivel del mar y temperatura.


```
In [56]: sns.set(style="darkgrid", color_codes=True)
g = sns.jointplot("TEMP", "WL", data=df1, kind="reg",
                  color="r", size=7)
plt.show(g)
```



Y la que relaciona salinidad y temperatura del agua.

```
In [54]: sns.set(style="darkgrid", color_codes=True)
g = sns.jointplot("SAL", "TEMP", data=df2, kind="reg",
                  color="r", size=7)
plt.show(g)
```

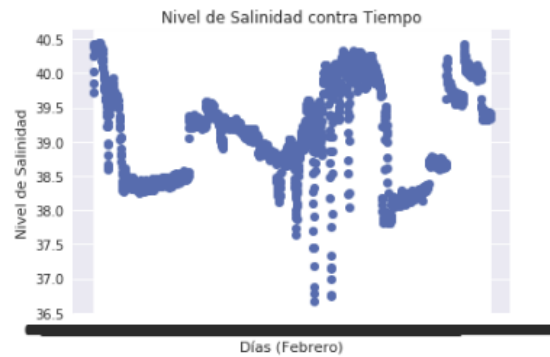


Ahora se piden tres graficas con respecto al tiempo: nivel del mar, salinidad y temperatura.

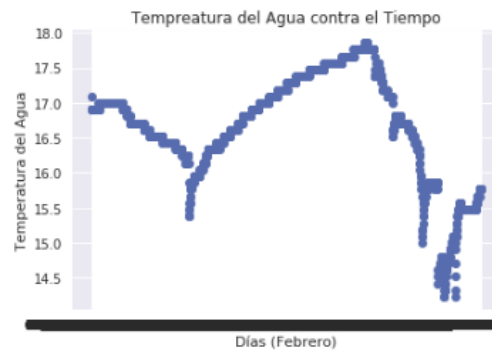
```
In [63]: plt.plot_date(x=df1.Date, y=df1['WL'])
plt.title("Nivel del Mar contra Tiempo")
plt.ylabel("Nivel del Mar")
plt.xlabel("Días (Febrero)")
plt.show()
```



```
In [64]: plt.plot_date(x=df2.Date, y=df2['SAL'])
plt.title("Nivel de Salinidad contra Tiempo")
plt.ylabel("Nivel de Salinidad")
plt.xlabel("Días (Febrero)")
plt.show()
```



```
In [65]: plt.plot_date(x=df1.Date, y=df1['TEMP'])
plt.title("Tempreatura del Agua contra el Tiempo")
plt.ylabel("Temperatura del Agua")
plt.xlabel("Días (Febrero)")
plt.show()
```



Después debemos hacer dos graficas de superposición en el eje y. La primera de nivel del mar y salinidad, mientras que la segunda de nivel del mar y temperatura del agua.

```
In [78]: fig = plt.figure()
ax1 = fig.add_subplot(111)
ax1.plot(df1.Ndate, df1.WL)
ax1.set_ylabel('Nivel del Mar')

ax2 = ax1.twinx()
ax2.plot(df2.Ndate, df2.SAL, 'r-')
ax2.set_ylabel('Salinidad', color='r')
for tl in ax2.get_yticklabels():
    tl.set_color('r')
plt.show()
```



```
In [79]: fig = plt.figure()
ax1 = fig.add_subplot(111)
ax1.plot(df1.Ndate, df1.WL)
ax1.set_ylabel('Nivel del Mar')

ax2 = ax1.twinx()
ax2.plot(df2.Ndate, df2.TEMP, 'r-')
ax2.set_ylabel('Temperatura', color='r')
for tl in ax2.get_yticklabels():
    tl.set_color('r')
plt.show()
```



Finalmente, debemos hacer utilizar la función `xlim` de `matplotlib` para checar si se relacionan el nivel del mar con la salinidad y/o temperatura del agua.

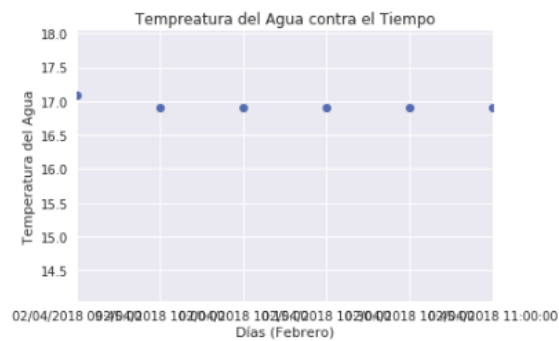
```
In [68]: plt.plot_date(x=df1.Date, y=df1['WL'])
plt.title("Nivel del Mar contra Tiempo")
plt.ylabel("Nivel del Mar")
plt.xlabel("Días (Febrero)")
plt.xlim(0, 5)
plt.show()
```



```
In [69]: plt.plot_date(x=df2.Date, y=df2['SAL'])
plt.title("Nivel de Salinidad contra Tiempo")
plt.ylabel("Nivel de Salinidad")
plt.xlabel("Días (Febrero)")
plt.xlim(0, 5)
plt.show()
```



```
In [70]: plt.plot_date(x=df1.Date, y=df1['TEMP'])
plt.title("Tempreatura del Agua contra el Tiempo")
plt.ylabel("Temperatura del Agua")
plt.xlabel("Días (Febrero)")
plt.xlim(0, 5)
plt.show()
```



Tras ésto podemos notar que la salinidad y la temperatura del agua cambian cuando una de éstas variables lo hace. En cuanto a la variable nivel del mar, no pude notar relación alguna.