

Evaluación 1

Luisa Casas

08 Marzo 2018

1 Limpieza de Datos

Comencé por descargar los archivos provistos por la página de computacional1.pbworks.com, y luego de renombrarlos, utilicé los comandos de emacs para reemplazar diagonales por espacios.

2 Pandas

2.1 Primera y Segunda parte

Primero importé las bibliotecas de pandas y matplotlib e hice que python me mostrara un encabezado para asegurarme de que la lectura fuera correcta. Después extraje el mes dado que es más fácil manejar los datos con un solo mes. Hice lo mismo para el segundo archivo, el cual incluía los datos de Salinidad.

```
In [1]: #Importando todo lo necesario
import pandas as pd
import numpy as np
from datetime import datetime
```

```
In [2]: #Para leer los archivos
df = pd.read_csv("dfsargento270218.csv", header=None, skiprows=1, names=['#', 'DateTime', 'Pres', 'Temp', 'WaterLvl'])
df.head()
```

```
Out[2]:
```

	#	DateTime	Pres	Temp	WaterLvl
0	1	02 04 2018 09:45:00	108.068	17.094	0.060
1	2	02 04 2018 10:00:00	107.815	16.903	0.035
2	3	02 04 2018 10:15:00	107.791	16.903	0.032
3	4	02 04 2018 10:30:00	107.791	16.903	0.032
4	5	02 04 2018 10:45:00	107.791	16.903	0.032

```
In [3]: #Extraer el mes
df['Ndate'] = pd.to_datetime(df['DateTime'], format='%m %d %Y %H:%M:%S')
df['month'] = df['Ndate'].dt.month
df.head()
```

```
Out[3]:
```

	#	DateTime	Pres	Temp	WaterLvl	Ndate	month
0	1	02 04 2018 09:45:00	108.068	17.094	0.060	2018-02-04 09:45:00	2
1	2	02 04 2018 10:00:00	107.815	16.903	0.035	2018-02-04 10:00:00	2
2	3	02 04 2018 10:15:00	107.791	16.903	0.032	2018-02-04 10:15:00	2
3	4	02 04 2018 10:30:00	107.791	16.903	0.032	2018-02-04 10:30:00	2
4	5	02 04 2018 10:45:00	107.791	16.903	0.032	2018-02-04 10:45:00	2

2.2 Tercera parte

Siguiente, importé la biblioteca de Seaborn para comenzar a hacer gráficas.

```
In [4]: #a.Boxplot de Nivel del agua
import seaborn as sns
import matplotlib.pyplot as plt
ax = sns.boxplot(x="month", y="WaterLvl", data=df)
plt.show()
```

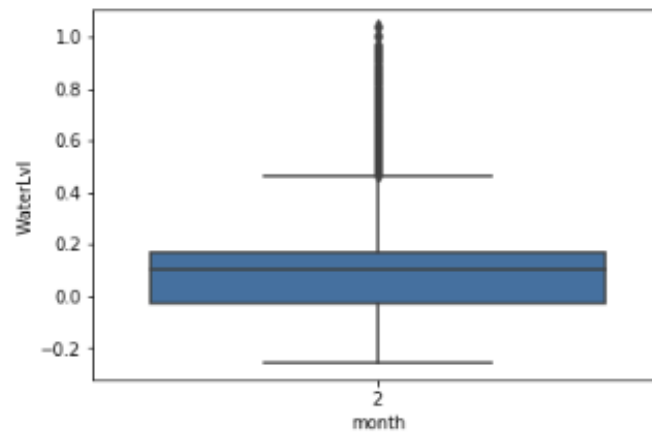


Figure 1: (a)Nivel del agua

```
In [7]: #b.Boxplot de Salinidad
import seaborn as sns
import matplotlib.pyplot as plt
ax = sns.boxplot(x="month", y="Sali", data=dg)
plt.show()
```

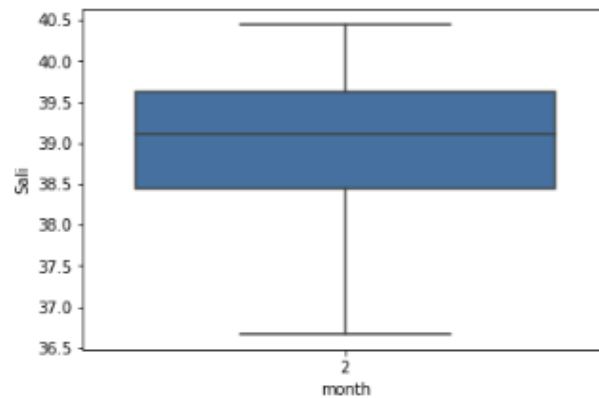
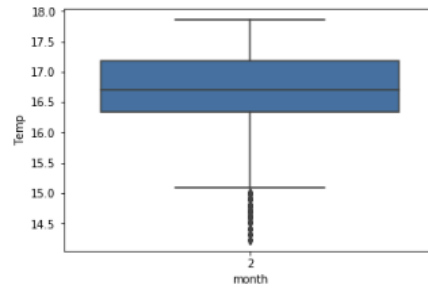


Figure 2: (b)Salinidad

```
In [8]: #c.Boxplot de Temperatura p/primer archivo
import seaborn as sns
import matplotlib.pyplot as plt
ax = sns.boxplot(x="month", y="Temp", data=df)
plt.show()
```



```
In [9]: #c.Boxplot de Temperatura p/segundo archivo
import seaborn as sns
import matplotlib.pyplot as plt
ax = sns.boxplot(x="month", y="Temp", data=dg)
plt.show()
```

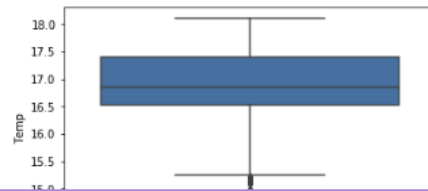


Figure 3: (c)Temperatura

2.3 Correlación de Pearson

Empleando de nuevo la ayuda de Seaborn, cree gráficas para explorar si había una correlación entre cada pareja de variables.

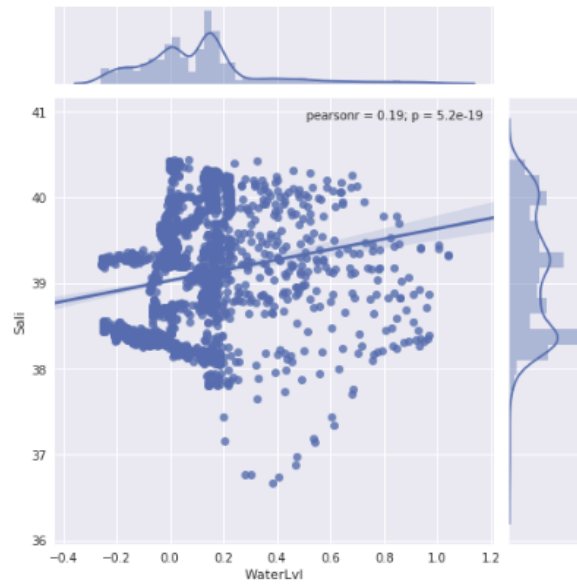


Figure 4: Nivel del mar-Salinidad

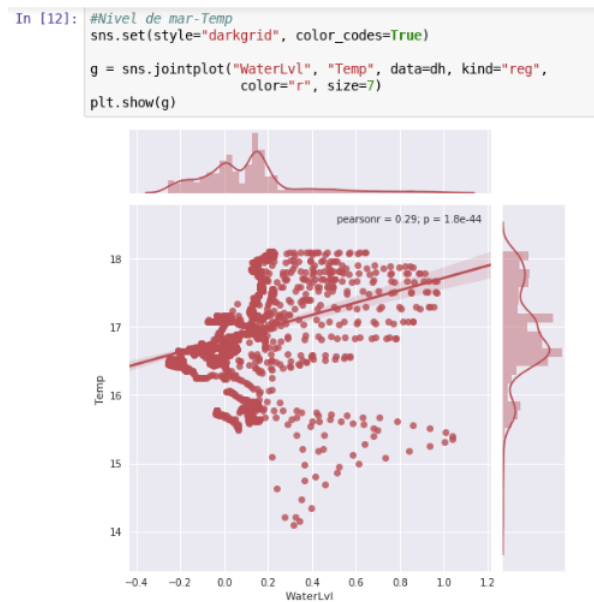


Figure 5: Nivel del mar-Temperatura

```
In [13]: #Salinidad-Temperatura
sns.set(style="darkgrid", color_codes=True)

g = sns.jointplot("Sali", "Temp", data=dh, kind="reg",
                  color="r", size=7)
plt.show(g)
```

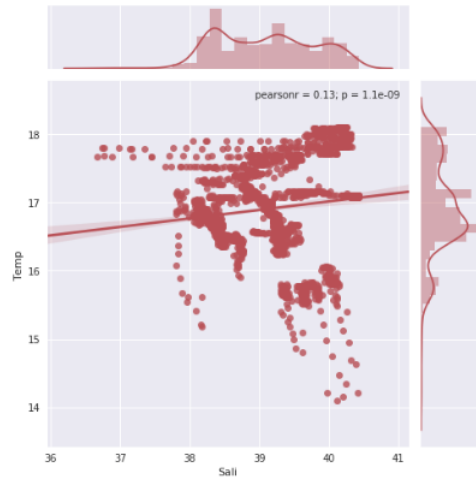


Figure 6: Salinidad-Temperatura

2.4 Gráficas Independientes

En esta sección se pide realizar gráficas independientes de distintas variables como función del tiempo.

```
In [14]: #Gráfica Nivel del mar
plt.figure(); dh.WaterLvl.plot(); plt.legend(loc='best')
plt.title("Variación del Nivel del mar")
plt.ylabel("Nivel del mar (m)")
plt.grid(True)
plt.show()
```



Figure 7: Nivel del mar

```
In [15]: plt.figure(); dh.Sali.plot(); plt.legend(loc='best')
plt.title("Variación de la Salinidad")
plt.ylabel("Salinidad(ppt)")
plt.grid(True)
plt.show()
```



Figure 8: Salinidad

```
In [16]: plt.figure(); dh.Temp.plot(); plt.legend(loc='best')
plt.title("Variación de la Temperatura")
plt.ylabel("Temperatura (°C)")
plt.grid(True)
plt.show()
```

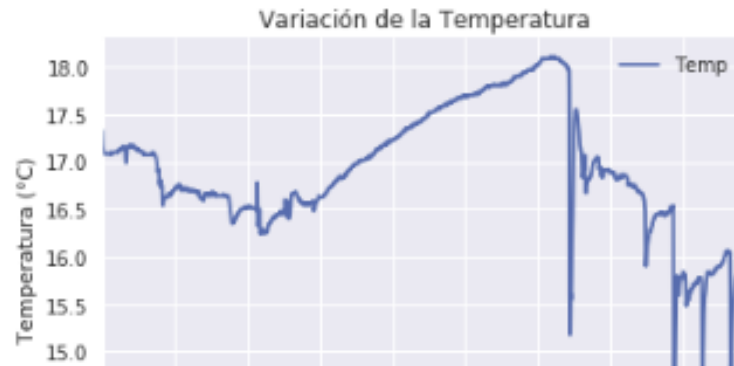


Figure 9: Temperatura

2.5 Gráficas superpuestas

Para esta sección, me fue necesario crear una concatenación de ambos archivos, dejando solamente las variables de Fecha, Temperatura, Nivel del mar, Salinidad y número de muestra.


```
In [10]: #Concatenando dataframes
dh=pd.concat([df['#'],df['DateTime'],df['WaterLvl'],dg['Sa
dh.head()
```

Out[10]:

	#	DateTime	WaterLvl	Sali	Temp
0	1.0	02 04 2018 09:45:00	0.060	39.7235	17.33
1	2.0	02 04 2018 10:00:00	0.035	39.8482	17.23
2	3.0	02 04 2018 10:15:00	0.032	40.0203	17.17
3	4.0	02 04 2018 10:30:00	0.032	40.2595	17.14
4	5.0	02 04 2018 10:45:00	0.032	40.2458	17.11

Figure 10: Concatenación

```
In [18]: #Nivel del mar-salinidad
df2 = dh[['WaterLvl','Sali']]
plt.figure(); df2.plot(); plt.legend(loc='best')
plt.title("Variación del nivel del mar y temperatura")
plt.ylabel("Nivel(m) / Sal(ppt)")
plt.grid(True)
plt.show()
```

<matplotlib.figure.Figure at 0x7fbf89402518>



Figure 11: Nivel del mar contra Salinidad

2.6 Limitando rangos

En esta sección utilicé la función de "xlim" para limitar el análisis de los datos a mediciones de 5 días, lo cual era un aproximado de 500 datos.

```
In [20]: #5 días nivel del mar-sali
df3 = dh[['WaterLvl', 'Sali']]
t=dh['#']
s=dh['WaterLvl']
plt.figure(); plt.plot(t,s); plt.legend(loc='best')
plt.title("Nivel del mar y salinidad")
axes = plt.gca()
axes.set_xlim([1000,1500])
plt.grid(True)
plt.show()
```



Figure 12: 5 días Nivel del mar-Salinidad

```
In [21]: #5 días nivel del mar-temp
df4 = dh[['WaterLvl', 'Temp']]
t=dh['#']
s=dh['WaterLvl']
plt.figure(); plt.plot(t,s); plt.legend(loc='best')
plt.title("Nivel del mar y temperatura a 5 días")
axes = plt.gca()
axes.set_xlim([1000,1500])
plt.grid(True)
plt.show()
```

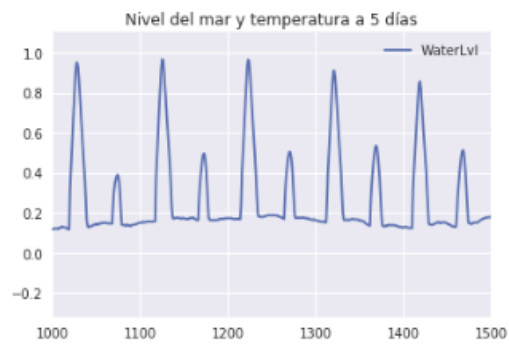


Figure 13: 5 días Nivel del mar-Temperatura