



UNIVERSIDAD DE SONORA

DIVISIÓN DE CIENCIAS EXACTAS Y NATURALES

FÍSICA COMPUTACIONAL

Buscando Regularidad en las Mareas

Autor:

Carlos Alí Medina Leal

Profesor:

Carlos Lizárraga Celaya

1. Un breve resumen...

En esta actividad se analizarán datos de los niveles de las mareas en distintos puertos y se tratará de conseguir alguna regularidad ante estas mediciones; aprenderemos a interpretar datos haciendo que python detecte fechas y a establecer un rango en las gráficas, además de interpolar datos faltantes, etc.

2. Introducción

Es posible ver las mareas como una superposición de diferentes modos debidos a la atracción gravitatoria del Sol, la Luna y la rotación terrestre. En cada sitio, las condiciones serán distintas, pero se preservan ciertas similitudes.

Siguiendo nuestro interés por estudiar las mareas del Mar de Cortez o Golfo de California, encontramos que el CICESE de Ensenada, produce los pronósticos de mareas para una colección de sitios en las costas de México; La costa que analizaremos en este caso será Guaymas.

Recordamos que como cuando la Tierra, la Luna y el Sol están en línea (sol-Tierra-luna, o sol-luna-Tierra) las dos influencias principales se combinan para producir mareas de primavera o mareas vivas; Cuando las dos fuerzas se oponen entre sí como cuando el ángulo de la luna-Tierra-sol está cerca de noventa grados, se producen las mareas muertas o marea de apogeo.

Las mareas son máximas cuando las dos pleamares son iguales. Eso solo ocurre cuando el eje mayor de los elipsoides es paralelo al plano ecuatorial. Es decir, cuando el sol se encuentra en el plano ecuatorial. Esto ocurre durante los equinoccios. Las mareas de equinoccio son las mayores del año.

3. Desarrollo

Empezamos esta actividad descargando datos de la costa que se eligió. Los datos contienen el mes, el día, la hora y la altura, los cuales analizaremos en python para graficar las mareas y verlas más claramente.

Primero cargaremos el archivo e importaremos las librerías con:

```
%matplotlib inline
import pandas as pd
```

```
import matplotlib as plt
import numpy as np
import matplotlib.pyplot as mplt
df = pd.read_csv("/home/camedina/Computacional1/Computacional1/
Actividad6/DatosG.csv")
```

Ahora, recordemos que el archivo contiene la fecha y la hora en archivo de texto, esto nos causará problemas ya que pandas no reconoce los números como fecha y hora. Para ajustar los valores como una fecha y hora, se usará lo siguiente:

```
df_clean = df.drop(df.columns[[0,1,2,3]], axis=1)

cols = df_clean.columns.tolist()
cols = cols[-1:] + cols[:-1]

df_clean = pd.DataFrame(df_clean[cols]):
```

Lo que modificará la tabla y la pondrá de la siguiente manera:

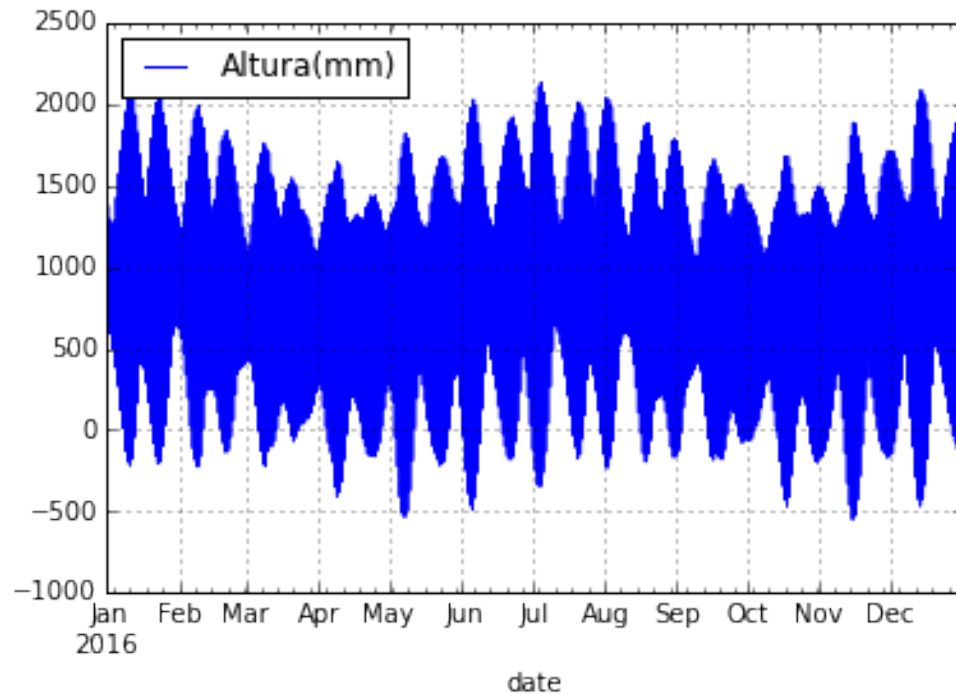
	date	Altura(mm)
0	2016-01-01 00:00:00	979
1	2016-01-01 01:00:00	982
2	2016-01-01 02:00:00	948
3	2016-01-01 03:00:00	889
4	2016-01-01 04:00:00	820
5	2016-01-01 05:00:00	761
6	2016-01-01 06:00:00	736
7	2016-01-01 07:00:00	757
8	2016-01-01 08:00:00	828
9	2016-01-01 09:00:00	955
10	2016-01-01 10:00:00	1120
11	2016-01-01 11:00:00	1268
12	2016-01-01 12:00:00	1357
13	2016-01-01 13:00:00	1376
14	2016-01-01 14:00:00	1331

Ahora, recordemos que en los datos nos pueden faltar valores, ya que en algún día no se midió el nivel del mar, o no estuvo en funcionamiento las instalaciones, etc. Este problema no se puede solucionar añadiendo ceros o borrando la fila, ya que el cero también sería un valor importante y borrar la fila alteraría los rangos, por lo que ambos métodos modificarían todos los datos, así que se usará el método de interpolación.

La interpolación nos servirá para reemplazar los datos faltantes por datos estimados por el comportamiento de la función, y puede ser de distintas formas, pero el método que usaremos será el spline, el cual se realizará con el siguiente código:

```
df_clean[u'Altura(mm)'] = df_clean[u'Altura(mm)'].astype(float).  
interpolate(method='spline', order=2)
```

Graficando nos quedará de la siguiente forma:

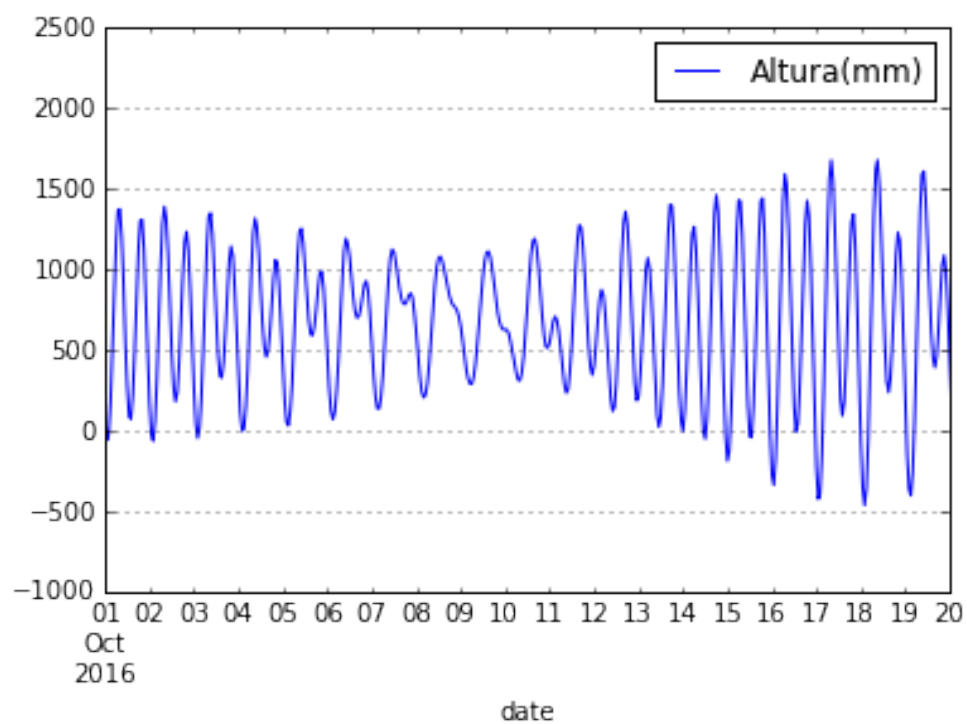


Para ver más cerca la gráfica, definimos un intervalo a graficar de la siguiente forma:

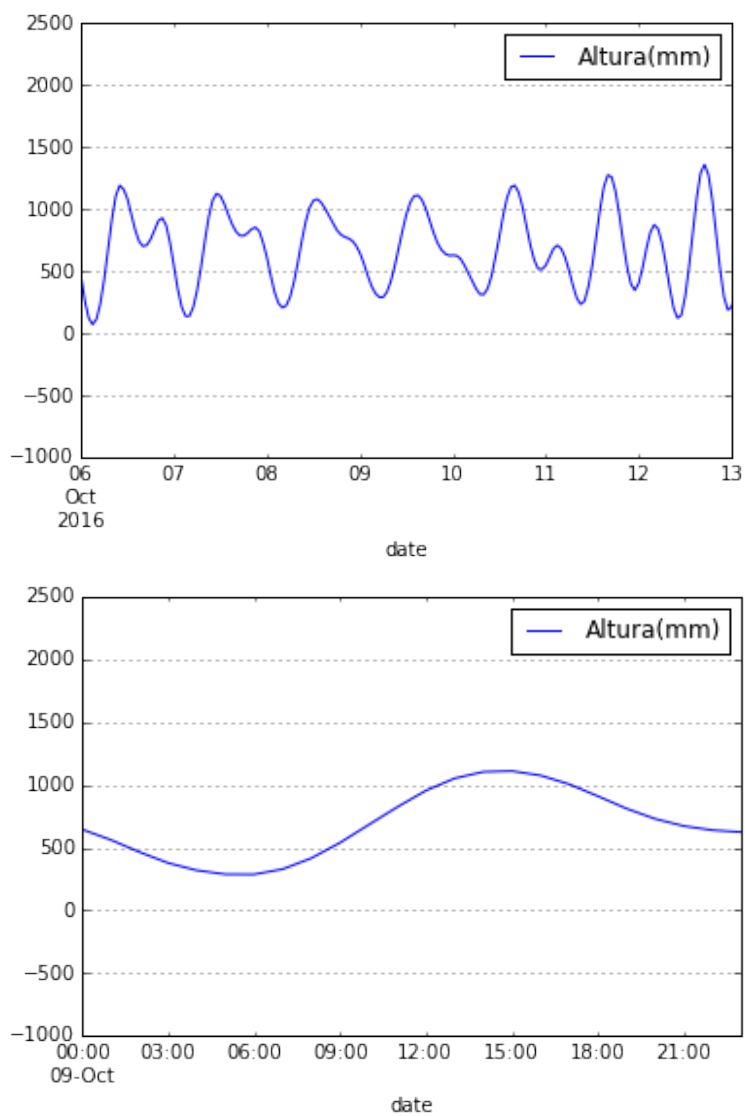
```
ax=df_clean.plot()  
ax.set_xlim(pd.Timestamp('2016-10-01'), pd.Timestamp('2016-10-20'))
```

Buscando Regularidad en las Mareas

Y graficando nos quedará:



Con esto se pueden observar cierto patrón de regularidad escondido en el nivel de las mareas. Acortamos más el rango y podemos ver las mareas más claramente.



Además de estos datos, en esta actividad también analizaremos los datos de la costa de El Sargento, Sonora, los cuales son de la siguiente forma:

	Date	Time	Abs Pres, kPa	Temp, C	Water Levels (m)	DOY
0	10/13/2013	15:30	103.956	24.255	-0.097	286
1	10/13/2013	16:00	103.893	24.158	-0.104	286
2	10/13/2013	16:30	103.842	24.062	-0.109	286
3	10/13/2013	17:00	103.806	24.062	-0.112	286
4	10/13/2013	17:30	103.757	24.062	-0.117	286
5	10/13/2013	18:00	103.720	24.062	-0.121	286
6	10/13/2013	18:30	103.696	24.062	-0.123	286
7	10/13/2013	19:00	103.659	24.062	-0.127	286
8	10/13/2013	19:30	103.647	24.062	-0.128	286
9	10/13/2013	20:00	103.572	23.966	-0.136	286
10	10/13/2013	20:30	103.562	24.062	-0.137	286

Ahora lo cargaremos, lo limpiamos y lo procesaremos:

```
dg = pd.read_csv("/home/camedina/Computacional1/Computacional1/Actividad6/smOct2013Mar2014cp.csv")
```

```
from datetime import datetime
df['date']= df.apply(lambda x:datetime.strptime("{0} {1} {2} {3}".format(x[u'Anio'],x[u'Mes'], x[u'Dia'], x[u'Hora(utc)']),
"%Y %m %d %H"),axis=1)
```

```
dg=dg.drop(u'#',1)
```

```
dg_clean = dg.dropna()
```

```
dg_clean=dg_clean.drop(u'Abs Pres, kPa',1)
```

```
dg_clean=dg_clean.drop(u'Temp, C',1)
```

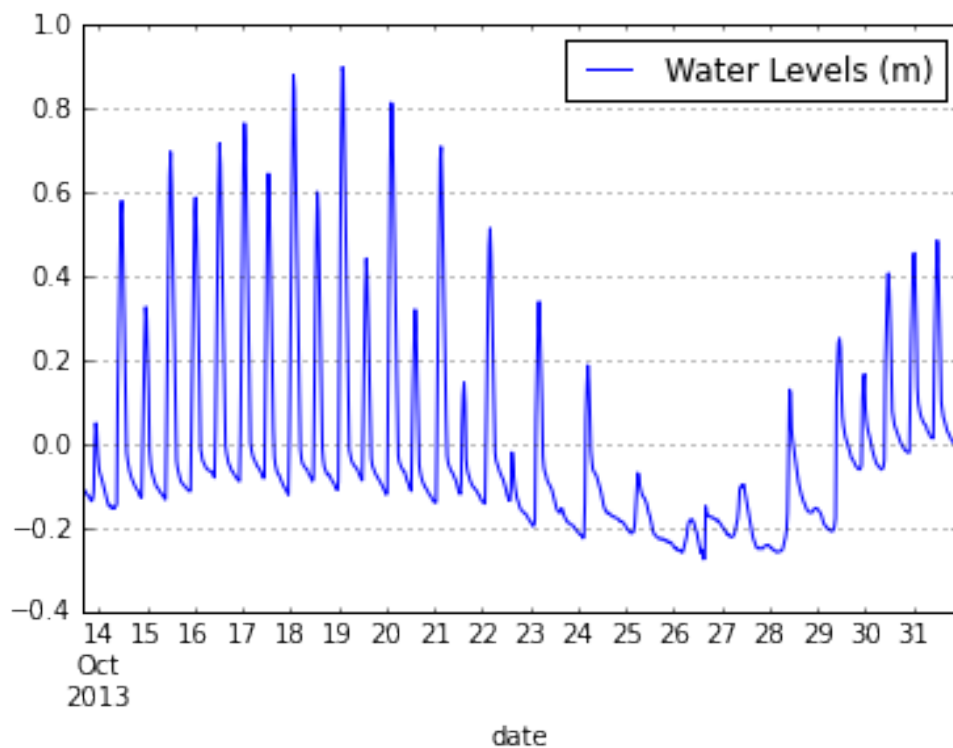
```
dg_clean=dg_clean.drop(u'DOY',1)
```

En ese código quitamos las columnas que no nos sirven y lo limpiamos, ahora vamos

a convertir los valores a tiempo:

```
dg_clean['fecha'] = dg_clean[u'Date'] + ' ' + dg_clean[u'Time']  
  
dg_clean['date'] = pd.to_datetime(dg_clean['fecha'], format='%m/%d/%Y %H:%M')  
  
dg_clean.set_index(['date'], inplace=True)  
  
del dg_clean['fecha']  
del dg_clean[u'Date']  
del dg_clean[u'Time']
```

Lo que nos lanza la siguiente gráfica:



Con esto hecho, nos dimos cuenta de las regularidades de las mareas que se saben en la teoría de las mareas y las observamos en la vida real, así como la interpolación y la transformación de valores a tiempo en Python/Pandas.

4. Bibliografía

CICESE *Pronostico de mareas*, (2015), recuperado el 17 de Octubre del 2016.
Desde: [http : //predmar.cicese.mx/calendarios/](http://predmar.cicese.mx/calendarios/)