

Reporte de Actividad 2

Roberto Benard Orci

06/02/2018

1 Introducción

En esta actividad se nos introdujo a Jupyter Notebook como entorno de programación, y con la ayuda de un ejemplo, exploramos las diferentes cosas que podíamos hacer con el lenguaje de programación Python. Después de terminar el ejemplo, se nos pidió realizar ciertas actividades adicionales para practicar lo que aprendimos con el ejemplo.

2 Resumen

Al comparar la forma de programación que usamos en Python, a través de Jupyter notebook, con la forma en la que programamos en FORTRAN, puedo decir que Python no se sintió muy diferente a FORTRAN, se sentía como si mezclara la terminal con el archivo de texto donde escribía el código, lo cual me pareció muy conveniente.

En esta sesión practicamos distintos comandos de Python en Jupyter notebook, vimos que funciones podíamos usar para analizar una gran lista de datos y graficamos estos datos para compararlos y hacer conclusiones al respecto.

2.1 Bibliotecas

Antes que nada, tenemos que cargar a la memoria de trabajo de Python diferentes bibliotecas. Las bibliotecas son paquetes que te permitirán usar distintas funciones. En nuestro caso, cargamos 3 bibliotecas: pandas, NumPy y matplotlib.pyplot.

```
In [1]: # Cargar a la memoria de trabajo las bibliotecas: Pandas (manejo de datos,  
# Numpy (numerical python) y la biblioteca de gráficas Matplotlib  
# Se asignan nombres cortos.  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
#  
# Usar "Shift+Enter" para procesar la información de la celda  
#
```

Bibliotecas

2.1.1 Pandas

Pandas es un paquete de Python que proporciona estructuras de datos rápidas, flexibles y expresivas diseñadas para que el trabajo con datos "relacionales" o "etiquetados" sea fácil e intuitivo.

2.1.2 NumPy

NumPy es el paquete fundamental para la programación científica con Python. Su uso principal es el de un contenedor multidimensional eficiente de datos genéricos.

2.1.3 Matplotlib.pyplot

Matplotlib es una biblioteca de bosquejo 2D de Python que produce figuras en una variedad de formatos impresos y entornos interactivos en todas las plataformas. Matplotlib intenta hacer las cosas fáciles. Se pueden generar gráficos, histogramas, espectros de potencia, gráficos de barras, diagramas de errores, diagramas de dispersión, etc.



Graficas

2.2 Lectura de un archivo

Después de cargar las bibliotecas descargamos un archivo con los datos que use usaran para practicar diferentes funciones, para esto entramos a [http : //smn1.conagua.gob.mx/emas/](http://smn1.conagua.gob.mx/emas/) en donde escoges un estado, luego un municipio para ver diferentes datos del clima en esa región en la última semana (aproximadamente).

Luego utilizas la función para leer el archivo después de los primeros 4 renglones (ya que en estos no hay datos, solo el encabezado del archivo), y los separa por más de un espacio entre cada columna, para efectos estéticos del arreglo.

2.3 Funciones

Posteriormente utilizamos varias funciones para organizar los datos y ver distintas características de las columnas, como los tipos de datos que son, el número de datos, su promedio, desviación estándar, mínimo, máximo, los cuartiles, etc. También utilizamos una función para unir las columnas de la fecha y la hora en una sola para hacer más sencillo el crear graficas.

2.4 Graficas

No fue muy difícil ver lo que se necesitaba para hacer una gráfica, a estas siempre incluimos 4 comandos, uno para nombrar la gráfica, otro para nombrar el eje y , otro para que la gráfica tuviera una cuadrícula, y la más importante, para que se mostrara la gráfica. Pero había uno o a veces 2 renglones que siempre cambiaban o se modificaban, este dictaba que era lo que se graficaba, que columna, y en este podías personalizar su color, si se graficaban 2 columnas, o incluso, si se usaba otra columna para el eje vertical, en otras palabras, graficar una o más columnas en función de otra.

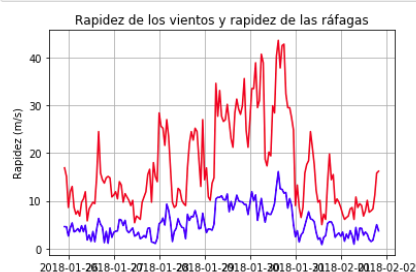
2.5 Actividades adicionales

Después de haber copiado los ejemplos de las gráficas y deducir para que servía cada comando intente hacer estas *actividades adicionales* por mi cuenta. Varias eran prácticamente idénticas a los ejemplos, pero intente practicarlas y moverles ciertas cosas.

2.5.1 I

Crear una gráfica que muestre la rapidez de los vientos y la rapidez de las ráfagas, como funciones del tiempo. ¿Cuáles son las horas del día con más viento?.

```
In [39]: # Gráfica de la rapidez de los vientos y la rapidez de las ráfagas, como funciones del tiempo.
plt.plot_date(x=df.FECHA, y=df.VELS, fmt="b-")
plt.plot_date(x=df.FECHA, y=df.VELR, fmt="r-")
plt.title("Rapidez de los vientos y rapidez de las ráfagas")
plt.ylabel("Rapidez (m/s)")
plt.grid(True)
plt.show()
```



Rapidez de los vientos y rapidez de las ráfagas

Para esta primera actividad simplemente cree un *df1* en el cual estuvieran las columnas de la rapidez de los vientos y la rapidez de las ráfagas. Las horas del día con más viento parecen ser las 3 y las 4 de la tarde.

2.5.2 II

Crear una gráfica con la dirección de los vientos como función del tiempo y comentar sobre los vientos dominantes en el sitio de estudio.

```
In [31]: # Gráfica de la dirección de los vientos como función del tiempo.
plt.plot_date(x=df.FECHA, y=df.DIRS, fmt="g-")
plt.title("Dirección de los vientos")
plt.ylabel("Grados(°)")
plt.grid(True)
plt.show()
```



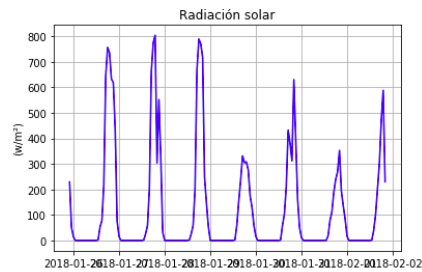
Dirección de los vientos

A pesar de que la dirección de los vientos parece ser muy dispersa, estos tienen una dirección entre 100 y 110 grados en promedio.

2.5.3 III

Muestre el comportamiento de la Radiación Solar como función del tiempo. ¿Que puedes comentar?

```
In [32]: # Gráfica de la radiación solar como función del tiempo.
plt.plot_date(x=df.FECHA, y=df.RADSOL, fmt="b-")
plt.title("Radiación solar")
plt.ylabel("(w/m²)")
plt.grid(True)
plt.show()
```



Radiación Solar

Que la es muy distintivo el nivel de radiación solar que recibe la región en el día a comparación de la noche, y que parece que en los últimos días del mes de enero esa región recibió más radiación que al inicio del mes de febrero.

2.5.4 IV

¿Cuál es el lapso de temperatura diaria? (Diferencia entre la temperatura máxima y la mínima).

```
In [33]: df.TEMP.describe()
Out[33]: count    167.000000
         mean      15.028144
         std       4.978664
         min       8.500000
         25%      11.250000
         50%      13.300000
         75%      17.900000
         max      27.400000
         Name: TEMP, dtype: float64
```

```
In [34]: df.TEMP.max()-df.TEMP.min()
Out[34]: 18.899999999999999
```

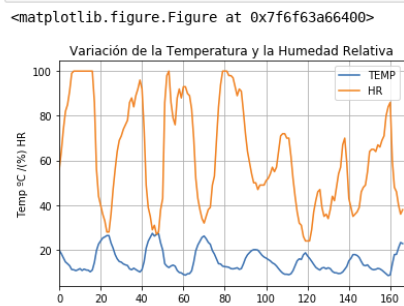
Diferencia de temperatura

La diferencia es de 18.9 grados Celsius, la mínima siendo 8.5 y la máxima 27.4.

2.5.5 V

¿Puedes comentar sobre la relación entre la temperatura y la humedad relativa?

```
In [35]: # Gráfica de Temperatura y Humedad Relativa
         df1 = df[['TEMP', 'HR']]
         plt.figure(); df1.plot(); plt.legend(loc='best')
         plt.title("Variación de la Temperatura y la Humedad Relativa")
         plt.ylabel("Temp °C /(%) HR")
         plt.grid(True)
         plt.show()
```



Temperatura y humedad relativa

Son inversamente proporcionales, cuando una disminuye la otra aumenta.

2.5.6 VI

Realiza el análisis exploratorio de datos, que resuma el sitio estudiado (Usar la función `describe()` sobre tu data frame).

```
In [36]: df.describe()
Out[36]:
```

	DIRS	DIRR	VELS	VELR	TEMP	HR	PB	PREC	RADSOL	TEMPCOMB	HUMCOMB	HUMSUBS
count	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000
mean	105.874251	109.748503	5.355509	17.105988	15.028144	62.562874	853.723353	0.056287	122.010180	15.137126	66.574850	0.532934
std	84.686180	86.919414	3.111111	9.537483	4.978664	23.454274	1.782409	0.349103	210.660269	6.645421	24.448226	0.500415
min	5.000000	13.000000	0.830000	5.000000	8.500000	24.000000	848.800000	0.000000	0.000000	7.400000	18.000000	0.000000
25%	51.500000	51.000000	3.135000	9.550000	11.250000	42.000000	852.600000	0.000000	0.000000	10.500000	47.000000	0.000000
50%	75.000000	75.000000	4.460000	14.400000	13.300000	59.000000	853.800000	0.000000	0.200000	12.600000	66.000000	1.000000
75%	120.500000	144.000000	7.190000	24.500000	17.900000	85.500000	854.950000	0.000000	180.750000	18.800000	90.500000	1.000000
max	355.000000	352.000000	16.120000	43.600000	27.400000	100.000000	857.300000	3.800000	803.700000	34.100000	100.000000	1.000000

Esta función parece ser muy útil a la hora de realizar un análisis estadístico de un conjunto de datos, ya que te da muchos datos con los que puedes generar diferentes conclusiones, y con la inclusión de graficas uno puede tener una gran herramienta para estudiar conjuntos de datos.

(n.d.). Retrieved February 03, 2018, from <http://smn1.conagua.gob.mx/emas/>

Pandas: powerful Python data analysis toolkit¶. (n.d.). Retrieved February 03, 2018, from <https://pandas.pydata.org/pandas-docs/stable/index.html>

Introduction. (n.d.). Retrieved February 03, 2018, from <https://matplotlib.org/>

NumPy¶. (n.d.). Retrieved February 03, 2018, from <http://www.numpy.org/>

1.-¿Cuál es tu primera impresión de Jupyter Notebook?

Como mencione al principio en el resumen, me dio la impresión de una terminal combinada con un archivo de texto de código, eso me agrado mucho, se siente más fácil trabajar de esa manera.

2.-¿Se te dificultó leer código en Python?

Un poco, pero eso es de esperarse ya que es un lenguaje nuevo para mí. Recordando la primera vez que use FORTRAN, me parecía imposible hacer algo, y no entendía muy bien lo que hacía, pero después de un año de programar en FORTRAN me siento muy cómodo al usarlo. Se que con Python no será muy diferente, lo único que necesito es tiempo y práctica.

3.-¿En base a tu experiencia de programación en Fortran, que te parece el entorno de trabajar en Python?

Me pareció un poco menos riguroso, lo único que diría que no me gusto es que a la hora de cometer un error no tengo idea de que hice mal y siento que FORTRAN hace un mejor trabajo en señalar donde se encuentra el error.

4.-A diferencia de Fortran, ahora se producen las gráficas utilizando la biblioteca Matplotlib. ¿Cómo fue tu experiencia?.

Esto fue algo que me gustó mucho de Python, su habilidad de cargar distintos paquetes de funciones (bibliotecas) para el uso que se le vaya a dar al programa, uno de los que usamos fue Matplotlib y, al ver los diferentes tipos de graficas que se pueden generar, estoy seguro de que será una herramienta muy útil.

5.-En general, ¿qué te pareció el entorno de trabajo en Python?

En este momento siento que el entorno de trabajo es más relajado en FORTRAN, pero, como mencione, esto probablemente se deba a que no he usado Python tanto como FORTRAN, con el tiempo es probable que cambie de opinión.

6.-¿Qué opinas de la actividad? ¿Estuvo compleja? ¿Mucho material nuevo? ¿Que le faltó o que le sobró? ¿Qué modificarías para mejorar?

Agregaría otro archivo con los comandos más básicos de Python, y más ejemplos con pequeños cambios para hacer énfasis en que comando o función hace que cosa. Empezando con el ejemplo más básico he ir subiendo de ahí.

7.-¿Comentarios adicionales que desees compartir?

Ninguno.