

Reporte de Actividad 2

Introducción a Jupyter y Phyton

Michelle Contreras Cossio

7 de Febrero del 2018

1 Introducción

En esta actividad nos encontramos por primera vez con la programación, como tal, en Phyton, usando el entorno de programación de Jupyter. A partir de la base de datos del Servicio Meteorológico Nacional, realizamos interpretación y análisis de datos del clima de cierta ciudad, explorando las diferentes herramientas que nos presenta Phyton.

Phyton es un lenguaje de programación, orientado a objetos, utilizado para aplicaciones o incluso páginas web, además es un lenguaje interpretado, que brinda mayor rapidez en su desarrollo; pandas por su parte, es una librería utilizada para el análisis y estructura de datos. Jupyter Notebook, es el entorno de programación donde estaremos trabajando.

2 Actividad Realizada

Durante la actividad se nos pidió aprender a correr Jupyter Notebook desde una terminal, para poder crear un archivo que alojara el código de esta actividad. Trabajamos con un conjunto de datos, estos fueron obtenidos desde el sitio web del Sistema Meteorológico Nacional, en este caso fue escogida la localidad de Todos Santos, BCS, de la cual obtuvimos un archivo txt con datos climatológicos.

Nos fue brindado un código, el cual debíamos imitar, pero con nuestros datos, para poder entender algunas funciones básicas de como se manejan los datos en pandas, a continuación se muestra:

1. En un principio, se cargaron las bibliotecas que utilizamos, estas fueron pandas, numpy y matplotlib, utilizadas para el manejo de datos, números y gráficas, respectivamente.
2. Posteriormente, se leyó el archivo txt creado de datos meteorológicos de Todos Santos y se imprimieron los primeros renglones para verificar que fueran leídos correctamente.
3. Se dio estructura a los datos como un data frame y se verificaron los tipos de categorías con que contabamos.
4. Combinamos dos columnas que contenían la fecha y hora, dentro de una sola.
5. Con "describe" analizamos aspectos importantes de los datos, como es el promedio por columna, máximos, mínimos, cuartiles, entre otros.
6. Hicimos un filtro o selección de datos, que incluía únicamente los datos con la temperatura dentro del rango entre los 24 y 25 ° C
7. Calculamos la media, por columnas, de los datos.
8. Empezamos a graficar:
 - La variación de la rapidez de los vientos, añadiendo título a la gráfica y ejes.
 - La variación de la temperatura y humedad relativa, ambas en una sola gráfica.
 - La variación de la temperatura con respecto al tiempo.

Posteriormente, se pidieron seis actividades adicionales:

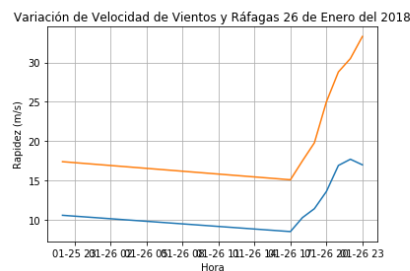
1. Crear una gráfica que muestre la rapidez de los vientos y la rapidez de las ráfagas, como funciones del tiempo. Para esto, se creó un nuevo data frame que incluyera únicamente los datos del 26 de enero del 2018, para poder comparar en un solo día estas rapidezces.

```
# Selecciona los datos del día 26 de Enero del 2018
df1 = df[df.FECHA > '2018-01-25 23:00:00']
df_dia = df_tmp[df_tmp.FECHA < '2018-01-27 00:00:00']
df_dia
```

	DIRS	DIRR	VELS	VELR	TEMP	HR	PB	PREC	RADSOL	FECHA
0	271	259	10.59	17.4	24.8	41	1007.3	0.0	555.5	2018-01-25 22:00:00
19	276	270	8.52	15.1	24.7	29	1010.9	0.0	636.7	2018-01-26 17:00:00
20	273	272	10.29	17.5	25.5	32	1010.2	0.0	784.3	2018-01-26 18:00:00
21	271	272	11.43	19.8	25.9	38	1009.4	0.0	853.7	2018-01-26 19:00:00
22	292	311	13.61	25.0	26.4	38	1008.4	0.0	824.7	2018-01-26 20:00:00
23	304	310	16.92	28.8	27.0	31	1007.4	0.0	729.7	2018-01-26 21:00:00
24	304	297	17.71	30.5	26.1	34	1006.8	0.0	561.8	2018-01-26 22:00:00
25	305	315	17.00	33.3	26.0	33	1006.6	0.0	331.7	2018-01-26 23:00:00

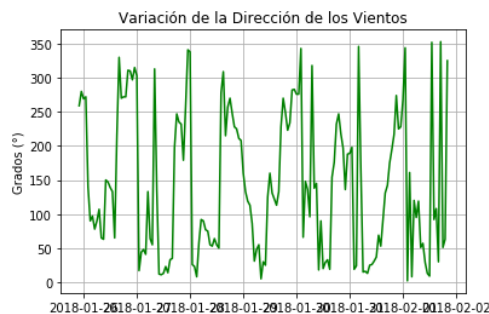
Y se graficaron ambas como función de tiempo. Se encontró que las horas del día con más viento son entre la 22 y 23 horas, sin embargo, también se notó que esta hora proporcionada es en UTC, por lo que a la zona horaria que en verdad le corresponde al municipio de Todos Santos, sería a entre 3 y 4 pm.

```
# Gráfica de velocidad de vientos y ráfagas 26/01/18
y=df_dia[['VELS','VELR']]
x=df_dia['FECHA']
plt.plot(x,y)
plt.title("Variación de Velocidad de Vientos y Ráfagas 26 de Enero del 2018")
plt.ylabel("Rapidez (m/s)")
plt.xlabel("Hora")
plt.grid(True)
plt.show()
```

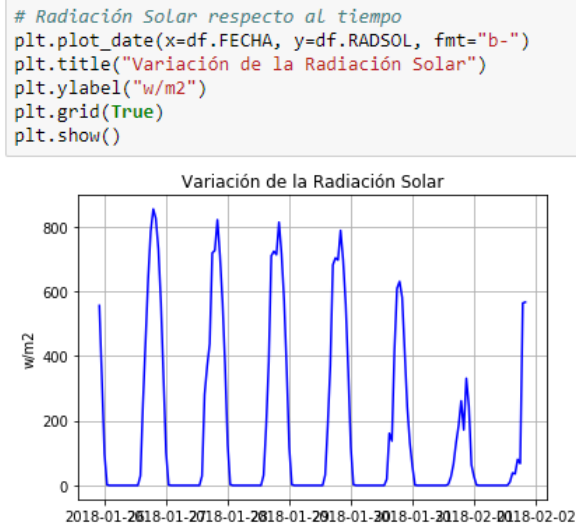


2. Crear una gráfica con la dirección de los vientos como función del tiempo. En esta gráfica se utilizaron todos los datos del archivo txt y se encontró que la dirección de los vientos, típicamente tiene una dirección con un ángulo de entre 260 y 300 grados, los cuales se podrían denominar como los vientos "dominantes" en este caso.

```
# Gráfica de dirección de vientos
plt.plot_date(x=df.FECHA, y=df.DIRR, fmt="g-")
plt.title("Variación de la Dirección de los Vientos")
plt.ylabel("Grados (°)")
plt.grid(True)
plt.show()
```

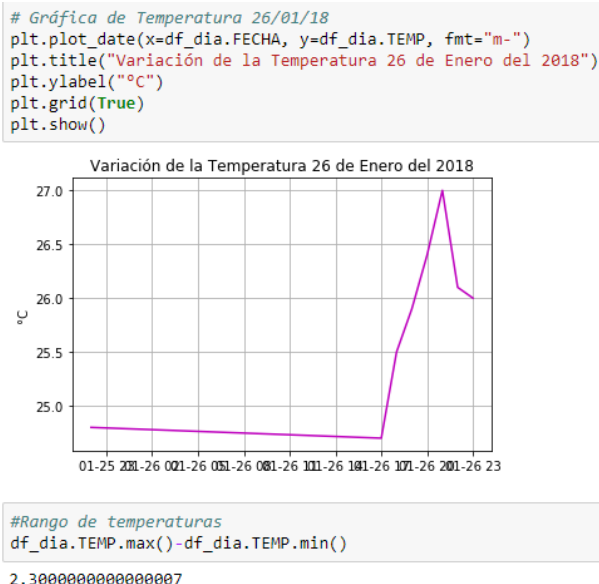


- Muestre el comportamiento de la Radiación Solar como función del tiempo. Se creó, de igual manera, una gráfica que incluyera las fechas, utilizando todos los datos y se encontró que la radiación diariamente y como es de esperar, decreciendo diariamente hasta cero, y es necesario comentar nuevamente que por la zona horaria que están manejando, incorrectamente, estos datos, muestra estas ausencias de radiación solar en un rango entre las 2:00 y 13:00 horas, que convertido a horario en Todos Santos, nos da entre las 7:00 pm y 6:00 am, aproximadamente.



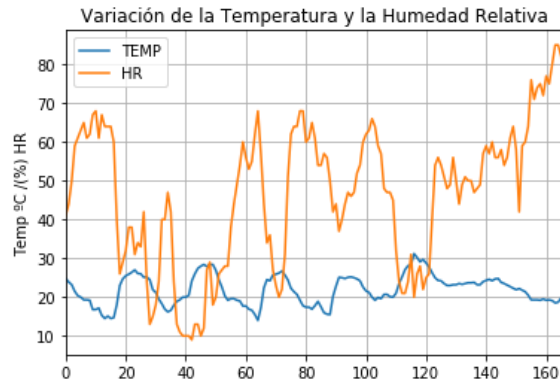
También podemos notar que por algunos días, el máximo diario de radiación solar comenzó a descender y este se ubica, aproximadamente, a las 12:00 horas.

- ¿Cuál es el lapso de temperatura diaria? Para responder esta pregunta, se creó, nuevamente, una gráfica de la variación de la temperatura, con respecto al 26 de enero del 2018 y para calcular el rango de variación de la temperatura, se restó la Temperatura máxima menos la mínima. Esto nos dio un total de 2.3° C.



- ¿Puedes comentar sobre la relación entre la temperatura y la humedad relativa? Para ello se utilizó la gráfica creada mediante el uso del ejemplo.

En la gráfica no pude notar ningún patrón de dependencia entre estas dos variables, en ciertos puntos tienen picos al mismo tiempo, pero en otros momentos, mientras la temperatura disminuye, la humedad relativa aumenta.



6. Realiza el análisis exploratorio de datos, que resuma el sitio estudiado, se realizó el describe sobre los datos que contienen a todo el archivo txt:

	DIRS	DIRR	VELS	VELR	TEMP	HR	PB	PREC	RADSOL
count	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.0	167.000000
mean	154.383234	144.580838	6.908862	15.406587	21.982036	46.425150	1006.810180	0.0	170.393413
std	107.384611	103.975572	4.989053	7.274350	3.755659	18.282063	1.507477	0.0	262.159527
min	2.000000	2.000000	0.340000	2.700000	14.000000	9.000000	1003.700000	0.0	0.000000
25%	52.000000	52.000000	2.390000	9.700000	19.200000	31.000000	1005.800000	0.0	0.000000
50%	135.000000	133.000000	6.310000	15.100000	22.300000	49.000000	1006.500000	0.0	0.000000
75%	250.000000	232.000000	10.110000	19.100000	24.700000	60.000000	1007.600000	0.0	302.350000
max	359.000000	353.000000	20.530000	34.700000	31.200000	85.000000	1010.900000	0.0	853.700000

3 Investigación

En esta sección se añade la información extra, recabada de internet, para el mayor y mejor entendimiento de las funciones y características de la biblioteca y entorno utilizados.

3.1 Jupyter

Como fue mencionado con anterioridad, Jupyter Notebook es un ambiente de programación que permite editar código, widgets, texto, ecuaciones, imágenes, video, los cuales se pueden convertir a diferentes formatos y compartir fácilmente.

3.1.1 Características

Jupyter Notebook contiene tres componentes principales:

- Aplicación web: su aplicación interactiva, donde se puede editar y crear código y documentos "notebook". Esta edita y corre código, permite crear widgets de JavaScript e incluye ecuaciones matemática.
- Kernels: son procesos separados que corren el código del usuario, en cierto lenguaje, en la aplicación web y muestran la salida directamente en la aplicación. Cuenta con kernels que soportan: Phyton, R, Julia, Ruby, Haskell, Scala, node.js, Go.
- Documentos "notebook": documentos que contienen una representación de lo que se ve en la aplicación web, todas las entradas y salidas, imágenes, ecuaciones, etc. Cada notebook tiene su propio kernel. Los notebooks consisten en una sucesión de celdas y tienen la extensión ".ipynb", pero pueden exportarse en diferentes formatos como PDF o LaTeX.

3.1.2 Ventajas

- Se puede usar en prácticamente cualquier lugar, es un software libre disponible en diferentes sistemas operativos y fácil de instalar.
- Soporta una amplia variedad de lenguajes de programación, entre los que destaca Python, R, Ruby, entre otros.
- Es interactivo, te brinda inmediatamente el "feedback" o salida del código.
- Es open source, permite a los usuarios revisarlo y proponer cambios o extensiones, incluso es posible personalizarlo para un uso específico.
- Produce gráficas, tablas entre otras capacidades matemáticas, es muy utilizado en la ciencia.

3.1.3 Desventajas

- Aunque es útil que sea interactivo, es un poco difícil acostumbrarse a esto.
- Es más complicado instalarlo y acomodarlo todo para un primer uso, que otros entornos de programación.
- No es fácil configurarlo para poder compartirlo y editar en otra computadora, es necesario realizar ciertos pasos que no podrían resultar triviales para cualquier usuario.
- Es difícil poder trabajar de manera colaborativa con este entorno, por lo mismo que no es sencillo "abrir" el código en otra computadora, como se revisó en el punto anterior.

3.2 Python: Pandas

Pandas es un paquete de Python que facilita el trabajar con datos.

Provee estructuras de datos de manera rápida y flexible y promete ser el analista y manipulador de datos, en open source, más poderoso, disponible en cualquier lenguaje.

Trabaja muy bien en el manejo de varios tipos de datos, ya sean tabulares, datos observacionales o estadísticos, entre otros.

3.2.1 Ventajas

- Reporta cuando se tienen datos perdidos.
- Permite el cambio de tamaño de los datos, se pueden intertar o eliminar columnas del data frame.
- Buenas herramientas IO que permiten cargar datos desde archivos planos, bases de datos, archivos excel y guardarlos en un formato HDF5 ultra rápido.
- Pandas es rápido.
- Se usa en la producción de aplicaciones financieras.
- Permite un manejo de bases de datos bastante extensas.
- Es fácil de usar y fácil de aprender.

3.2.2 Desventajas

- Cuenta con mucha competencia, como es excel, con el que es bastante comparado, ya que también permite el uso de bases de datos, de una manera un poco más sencilla para un usuario inexperimentado en el uso de código.

3.2.3 Herramientas útiles

A continuación se presentan algunas funciones básicas que pueden parecer útiles al momento de trabajar con pandas:

- Para análisis o inspección de datos:
 - `df.mean()`, muestra el promedio de todas las columnas.
 - `df.corr()`, muestra la correlación entre columnas.
 - `df.count()`, muestra el número de datos.
 - `df.max()`, muestra el máximo de cada columna.
 - `df.min()`, muestra el mínimo de cada columna.
 - `df.median()`, muestra la mediana de cada columna.
 - `df.std()`, muestra la desviación estándar de cada columna.
- Para selección de datos:
 - `(df[col])`, selecciona una columna como un nuevo data frame.
 - `(df[[col1, col2]])`, selecciona dos columnas como un nuevo data frame.
 - `(s.iloc[0])`, selecciona por posición.
- Para el filtro de datos:
 - `df[df[year] > 1984]`, brindará las columnas donde el año es mayor que 1984.
 - `df.groupby(col)`, separa, según un criterio específico, en grupos, aplica una función a cada grupo, de manera independiente y combina el resultado en una estructura de datos.

4 Apéndice

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

Al principio no entendía de que se trataba, ya que el entorno donde estaba acostumbrada a trabajar era en una aplicación descargada en la computadora (Geany), estaba acostumbrada a trabajar con el lenguaje Fortran, que es un lenguaje compilado; por otro lado, en Phyton no es necesario compilar y eso me resultó más fácil.

2. ¿Se te dificultó leer código en Python?

Si, pero con los letreros que presentó el maestro he ido entendiendo, aunque no del todo, entiendo la estructura, pero no algunas banderas u opciones que se presentan.

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

Mucho más sencillo ya que la estructura y sintaxis es menos complicada, sin embargo, como ya mencione, hay muchos comandos que aun no entiendo.

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

Lás gráficas son visualmente más atractivas y fáciles de producir que con Gnuplot, Phyton es un lenguaje con mejor estética que Fortran y eso me agrada.

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

Por lo pronto, es sencillo de usar, fácil de instalar en cualquier computadora con cualquier SO.

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

Me gustó la actividad como iniciación a Phyton, no fue compleja pero si es un poco difícil encontrar resultados web de cierto problema, ya que Phyton es un lenguaje bastante recurrido y hay demasiada información, que en cierto punto es difícil de digerir. Si fue mucho material nuevo, pero el suficiente.

7. ¿Comentarios adicionales que desees compartir?

Phyton, hasta este punto, me esta gustando, pero tengo que leer mucho de las funciones que ofrece, para poder entender mejor cuando busque soluciones, en foros, por ejemplo.

5 Referencias Bibliográficas

- <https://desarrolloweb.com/articulos/1325.php>
- <https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/What%20is%20the%20Jupyter%20Notebook.html>
- <https://www.quora.com/What-are-the-pros-and-cons-of-using-Python-Jupyter-versus-a-normal-Python-development-environment>
- <https://www.slant.co/topics/366/viewpoints/23/> best-python-ides jupyter
- <https://www.dataquest.io/blog/jupyter-notebook-tips-tricks-shortcuts/>
- <https://pandas.pydata.org/pandas-docs/stable/index.html>
- <https://towardsdatascience.com/a-quick-introduction-to-the-pandas-python-library-f1b678f34673>