

Reporte de Actividad 2: Introducción a Python, Jupyter y Pandas

Jesús Antonio González Espinosa

Física Computacional 1

7 de Febrero del 2018

La segunda actividad del curso fue usar un ejemplo aportado por el profesor para empezar a identificar e interpretar el lenguaje de programación Python, en el entorno de Jupyter Notebook. También, en base al mismo código, se nos pidió realizar unas actividades más que seguían con el contenido del ejemplo dado. Ambas partes de la actividad fueron con el objetivo de introducirnos al uso de lenguaje de programación Python en el entorno Jupyter Notebook.

1 Introducción

1.1 ¿Qué es Python?

Python es un lenguaje de programación interpretado, o sea, que el código no requiere ser compilado, ya que consiste en ser interpretado en tiempo real. Además, Python es multiparadigma, al permitir varios estilos de programación como programación orientada a objetos, programación imperativa, programación funcional, o muchos otros que son soportados mediante el uso de extensiones. Además, uno de los beneficios de él, es que tiene una sintaxis muy accesible, haciéndolo fácil de escribir y leer. Otra de las ventajas de Python es que al ser de licencia de código abierto, hay una gran comunidad que crea librerías que aportan herramientas extras.

1.2 ¿Qué es Jupyter Notebook?

Antes de hablar de Jupyter, es necesario conocer los documentos "notebook". Estos son documentos que contienen tanto código como elementos de texto, con el fin de dar provecho de ambos en conjunto. Jupyter Notebook es una aplicación que permite editar y correr documentos notebook en exploradores web. No requiere tener acceso a internet para trabajar en él. Al entrar a la aplicación, este te presenta un tablero o panel de control, donde se muestran todos los archivos locales, que permite abrir los documentos tipo notebooks. Hoy en día soporta muchos tipos de lenguajes de programación.

2 Actividad

En esta actividad usamos Jupyter Notebook para trabajar con Python. Lo que haremos es usar un documento de Jupyter Notebook de ejemplo que tiene una código en Python que dan descripción y presentación de varias tablas de datos de una estación del Servicio Meteorológico Nacional; el cual vamos a replicar pero con una nueva estación meteorológica. En este caso, vamos a trabajar con los datos obtenidos por la estación ubicada en el volcán "La Malinche", en Tlaxcala.

Se presentará un poco sobre los comandos y las librerías usadas durante la actividad y la función que tienen.

2.1 Comandos

Al iniciar, lo primero que se hace es abrir las bibliotecas necesarias para el trabajo. En éste caso, se usaron tres, y se les asignó a una variable. Fueron llamadas por el código:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

- **Pandas:** es una biblioteca destinada al análisis de datos, que proporciona estructuras de datos flexibles y permite trabajar con ellos de manera muy eficiente.
- **Numpy:** una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices
- **Matplotlib.pyplot:** biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays

Con las librerías listas, se cargaron los datos meteorológicos de "La Malinche" mediante el comando:

```
df0 = pd.read_csv('LaMalinche.txt', skiprows=4, sep='\s+')
```

Con esto, ya se puede trabajar con los datos. Dentro del archivo ejemplo hay varios comandos que presentan varios análisis de los datos. Algunos comandos fueron `df0.head()`, que presenta tanta cantidad de datos del archivo como pongas entre el paréntesis. También usamos los comandos `df = pd.DataFrame(df0)`, en conjunto con `df = dtypes`, el primero siendo para dar estructura de data frame a los datos, y el segundo para presentar los tipos de datos que Pandas ha reconocido.

	DD/MM/AAAA	HH:MM	DIRS	DIRR	VELS	VELR	TEMP	HR	PB	PREC	RADSOL	TEMPCOMB	HUMCOMB	HUMSUBS	TEMPSUBS
0	25/01/2018	22:00	57	94	5.43	24.1	13.6	57	736.4	0.0	130.7	16.7	43.0	6.0	9.2
1	25/01/2018	23:00	52	274	5.75	34.2	12.2	63	736.6	0.0	106.0	13.9	46.0	6.0	9.3
2	26/01/2018	00:00	42	81	5.98	27.7	10.1	72	736.9	0.0	8.3	9.9	48.0	6.0	9.6
3	26/01/2018	01:00	44	81	5.77	21.6	9.0	75	737.2	0.0	0.0	8.0	54.0	6.0	9.7
4	26/01/2018	02:00	29	82	2.49	20.5	8.4	78	738.1	0.0	0.0	7.6	60.0	6.0	9.7
5	26/01/2018	03:00	28	90	1.91	15.1	7.6	81	738.6	0.0	0.0	5.9	62.0	6.0	9.7
6	26/01/2018	04:00	355	331	1.76	10.8	6.4	86	738.8	0.0	0.0	4.9	67.0	6.0	9.7
7	26/01/2018	05:00	24	315	2.55	10.1	5.7	86	738.7	0.0	0.0	3.9	70.0	6.0	9.7
8	26/01/2018	06:00	9	235	2.97	9.4	4.0	90	738.1	0.0	0.0	2.2	73.0	6.0	9.6
9	26/01/2018	07:00	9	331	2.68	11.9	3.2	92	737.8	0.0	0.0	1.7	77.0	6.0	9.5

Figure 1: Datos del archivo leído.

```
DD/MM/AAAA    object
HH:MM         object
DIRS          int64
DIRR          int64
VELS          float64
VELR          float64
TEMP          float64
HR            int64
PB            float64
PREC          float64
RADSOL        float64
TEMPCOMB      float64
HUMCOMB       float64
HUMSUBS       float64
TEMPSUBS      float64
dtype: object
```

Figure 2: Tipo de datos del archivo.

Luego, en el ejemplo hay un código que une la columna de 'DD/MM/AAAA' con la de 'HH:MM', para poder crear una variable de tiempo única. Mostrando el código que hace el cambio de unificar y borrar las columnas, para crear una nueva llamada fecha:

```
# Combinar las columnas "DD/MM/AAAA" con "HH:MM" y convertirla a variable de tiempo
# Se crea una nueva columna "Fecha" al final con formato de tiempo.
# Eliminamos las dos primeras columnas que ya no necesitaremos
df['FECHA'] = pd.to_datetime(df.apply(lambda x: x['DD/MM/AAAA'] + ' ' + x['HH:MM'], 1), dayfirst = True)
df = df.drop(['DD/MM/AAAA', 'HH:MM'], 1)

df.head(10)
```

	DIRS	DIRR	VELS	VELR	TEMP	HR	PB	PREC	RADSOL	TEMPCOMB	HUMCOMB	HUMSUBS	TEMPSUBS	FECHA
0	57	94	5.43	24.1	13.6	57	736.4	0.0	130.7	16.7	43.0	6.0	9.2	2018-01-25 22:00:00
1	52	274	5.75	34.2	12.2	63	736.6	0.0	106.0	13.9	46.0	6.0	9.3	2018-01-25 23:00:00
2	42	81	5.98	27.7	10.1	72	736.9	0.0	8.3	9.9	48.0	6.0	9.6	2018-01-26 00:00:00
3	44	81	5.77	21.6	9.0	75	737.2	0.0	0.0	8.0	54.0	6.0	9.7	2018-01-26 01:00:00
4	29	82	2.49	20.5	8.4	78	738.1	0.0	0.0	7.6	60.0	6.0	9.7	2018-01-26 02:00:00
5	28	90	1.91	15.1	7.6	81	738.6	0.0	0.0	5.9	62.0	6.0	9.7	2018-01-26 03:00:00
6	355	331	1.76	10.8	6.4	86	738.8	0.0	0.0	4.9	67.0	6.0	9.7	2018-01-26 04:00:00
7	24	315	2.55	10.1	5.7	86	738.7	0.0	0.0	3.9	70.0	6.0	9.7	2018-01-26 05:00:00
8	9	235	2.97	9.4	4.0	90	738.1	0.0	0.0	2.2	73.0	6.0	9.6	2018-01-26 06:00:00
9	9	331	2.68	11.9	3.2	92	737.8	0.0	0.0	1.7	77.0	6.0	9.5	2018-01-26 07:00:00

Figure 3: Datos del archivo con los datos de la fecha unificados.

A partir de aquí, el código ejemplo muestra algunas herramientas como comandos para hacer análisis exploratorios o para sacar los promedios. En el ejemplo, se muestra el comando que hace el análisis exploratorio de los datos:

```
# Realizo un análisis exploratorio de datos
df.describe()
```

Figure 4: Comando que muestra el análisis exploratorio de los datos.

También se muestran algunas formas de crear restricciones a los datos.

```
# Selecciono los renglones con Temperatura > 24°C y < 25°C
df_temp = df[df.TEMP > 24]
df_select = df[df.TEMP > 24]
```

Figure 5: Comando que hace restricciones.

En el caso de los datos de "La Malinche", no había datos que cumplieran las restricciones, por lo que no hubo tabla para mostrar.

Lo último que se mostró en el ejemplo, fueron como hacer gráficas. Se presentaron varias gráficas de acuerdo a los datos de "La Malinche"; algunos incluso con respecto al tiempo. Algunas gráficas con sus respectivos códigos fueron:

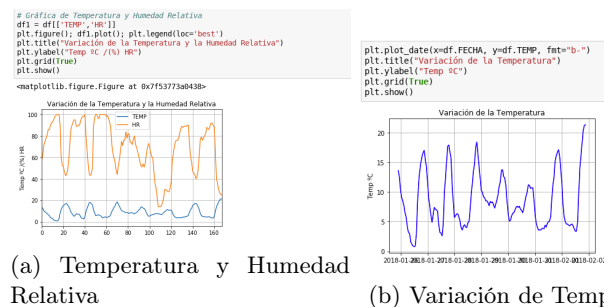


Figure 6: Gráficas de los datos de La Malinche

Esto dio fin al ejemplo e inicio a una actividad para el alumno en el cual en base al código de ejemplo, debíamos lograr obtener más gráficas y análisis de datos.

2.2 Actividad Adicional

1. **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?**

Para esta gráfica use pedazos de código de los ejemplos anteriores, un pedazo fue de la gráfica de "Variación de la Temperatura y la Humedad Relativa" para usar dos gráficas en una; y el otro pedazo fue de la gráfica de la "Variación de la Temperatura" para hacerla con respecto al tiempo. Finalmente, limite el tiempo a un día, para poder ver las horas del día con más viento. Podemos notar que alrededor de la media noche, se da el pico más alto en la gráfica del viento, ya que se ve entre las 21 horas y el inicio del día 30.

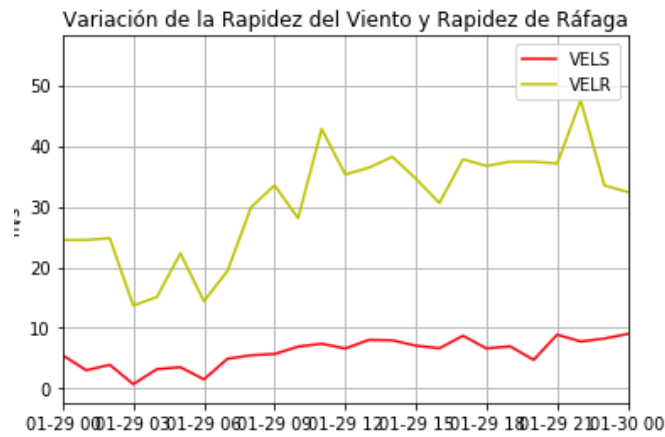


Figure 7: Gráfica de Velocidad de los Vientos y Rapidez de las Ráfagas.

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

En la gráfica podemos ver que está muy variado, pero se puede notar fácilmente que en varios días la dirección domina en 0 grados.

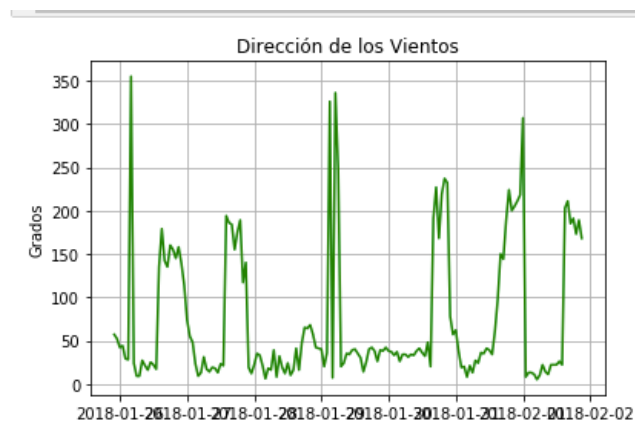


Figure 8: Gráfica de la Dirección de los Vientos.

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

Esta gráfica la limite a un solo día para poder observar el comportamiento del sol durante el día. Podemos ver que muestra en la gráfica radiación solar durante la noche. Esto puede

ser a causa de problemas con la adaptación del uso horario en el código, o de alguna manera llega la radiación solar a esa zona en la noche.

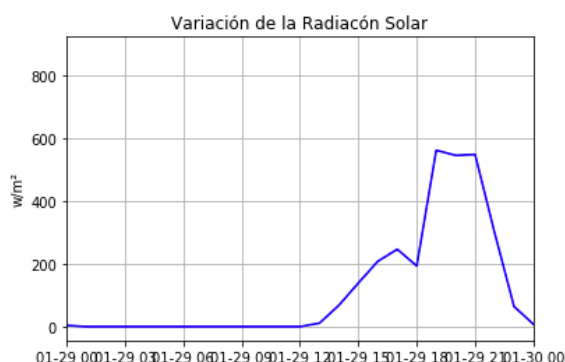


Figure 9: Gráfica de la Radiación Solar.

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

Ahora para poder ver el lapso de temperatura diaria, se limitó el rango a un solo día, basado en el ejemplo, y con eso, se evaluó la temperatura máxima y mínima con comandos que buscan esos datos específicos. Luego se depositaron en una variable, y luego se hizo la resta de ambos, para ver el lapso de temperatura, resultando en 6.1 Grados Centígrados.

```
df_DTE = df[df.FECHA >= '2018-01-30 00:00']
df_select = df_DTE[df_DTE.FECHA < '2018-01-30 23:59']
df_select
```

Figure 10: Código para limitar los datos a un día.

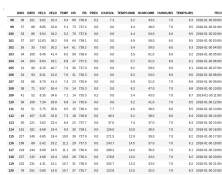


Figure 11: Mostrando los datos de un solo día.

```
TMin = df_select.TEMP.min()
TMin
```

5.099999999999999996

```
TMax = df_select.TEMP.max()
TMax
```

11.1999999999999999

$$TDif = TMax - TMin$$

6.099999999999999996

Figure 12: Diferencia entre Temperatura máxima y mínima.

5. **¿Puedes comentar sobre la relación entre la temperatura y la humedad relativa?**
Viendo la gráfica de Variación de la Temperatura y la Humedad Relativa, podemos inferir que cada vez que la temperatura sube, la humedad va a bajar, y viceversa. En otras palabras, son inversas una a la otra.

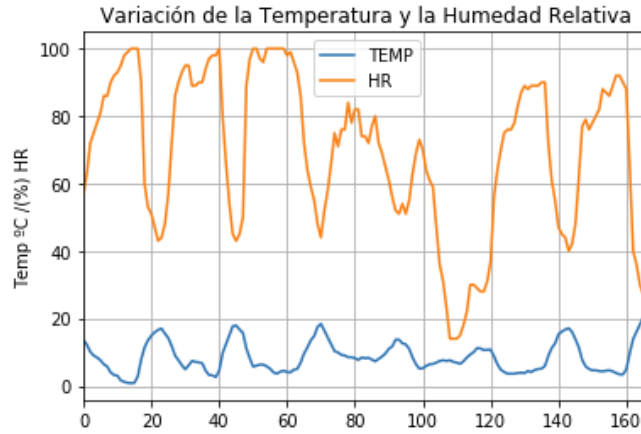


Figure 13: Gráfica de Temperatura y Humedad Relativa.

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

Los resultados del análisis exploratorio fueron:

	DIRS	DIRR	VELS	VELR	TEMP	HR	PI	PREC	RADSOL	TEMPCOMB	HUMCOMB	HUMSUBS
count	168.000000	168.000000	168.000000	168.000000	168.000000	168.000000	167.000000	167.000000	167.000000	167.000000	167.000000	167.000000
mean	72.851190	142.232143	4.593333	20.659552	8.652381	69.142857	732.270833	0.046707	91.546707	8.857485	68.68683	6.359281
std	78.495253	93.788140	2.335603	11.013463	4.617514	23.702317	56.843514	0.113956	172.676027	7.092312	19.863623	0.481233
min	5.000000	0.000000	0.420000	5.000000	0.000000	14.000000	0.000000	0.000000	0.000000	-0.600000	32.000000	6.000000
25%	21.000000	74.750000	2.910000	10.400000	4.875000	51.750000	735.900000	0.000000	0.000000	3.150000	51.500000	6.000000
50%	36.000000	103.500000	3.655000	18.700000	7.850000	74.000000	736.650000	0.000000	0.000000	6.300000	71.000000	6.000000
75%	121.500000	231.250000	6.515000	29.275000	11.400000	89.000000	737.300000	0.000000	105.400000	13.150000	84.000000	7.000000
max	355.000000	353.000000	10.030000	55.400000	21.300000	100.000000	739.000000	0.800000	880.000000	32.300000	99.000000	7.000000

Figure 14: Tabla de análisis exploratorio de los dato.

3 Apéndice

- (a) **¿Cuál es tu primera impresión de Jupyter Notebook?**

Ha sido interesante conocer éste nuevo entorno de programación y la forma en la que trabaja. Me impresiona y me es agradable poder correr cada línea a mi gusto, ya que anteriormente la única forma que conocía para correr códigos era con FORTRAN y era compilar cada vez que querías correr el programa lo cual se podía poner tedioso.

- (b) **¿Se te dificultó leer código en Python?**

Al principio sí, ya que es un nuevo lenguaje que no conozco nada, pero mediante le fui buscando y moviendo a las líneas de código y buscando en Internet, las cosas empezaron a tomar sentido.

- (c) **¿En base a tu experiencia de programación en Fortran, que te parece el entorno de trabajar en Python?**

Es mucho más sencillo, ya que no hay que estar haciendo cosas como compilar cada vez que quieres probar tu código. También siento mayor rapidez al entender los códigos, tal vez sea porque FORTRAN fue el primer lenguaje de programación que aprendí, pero a pesar de eso, siento una mayor comodidad desde un inicio al trabajar con Python.

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

Es impresionante, ya que a tres semanas del curso ya estamos pudimos ver como graficar, ya que con FORTRAN en un semestre, nunca creamos una gráfica en si.

(e) **En general, ¿qué te pareció el entorno de trabajo en Python?**

Ha sido interesante, había escuchado mucho del lenguaje, pero nunca lo había trabajado, y ahora que lo hice, fue interesante notar que cuando dicen que es muy fácil de entender, veo que tienen razón, es muy directo en cuanto a sus comandos y Jupyter Notebook lo hace todavía más atractivo a la vista.

(f) **¿Qué opinas de la actividad? ¿Estuvo compleja? ¿Mucho material nuevo? ¿Que le faltó o que le sobró? ¿Qué modificarías para mejorar?**

El salto que hubo del ejemplo a la Actividad Adicional fue algo que sentí muy pesado al inicio, no sabía que era lo que tenía que lograr o por donde empezar, pero después de volver a analizar los datos y buscar en Internet sobre los comandos, fue creando un mejor acercamiento, después de unos cuantos minutos ya me sentí más seguro en que no era algo difícil; al finalizarla, se puede decir que fue un poquito difícil, pero es algo que sí se puede lograr. La actividad fue una buena introducción a ambos Python y Jupyter Notebook.

(g) **¿Comentarios adicionales que desees compartir?**

Hasta el momento, hubo mucha comodidad trabajando con Python, no sentí la misma presión que sentí cuando aprendí Python. Tal vez sea porque es el segundo lenguaje que estamos aprendiendo, pero espero que lo que dicen de que Python es un lenguaje muy sencillo y elegante.

4 Conclusión

Jupyter Notebook ha resultado ser muy interesante, la versatilidad que trae es mucha gracias a su capacidad de usar múltiples lenguajes de programación, también gracias a su capacidad de ser modificado y usar bibliotecas para abrir más posibilidades a los usuarios. Su interfaz también es muy agradable a la vista, además de que en lo personal sentí muy cómodo al estar trabajando en un explorador web. Por otra parte, las limitaciones de Jupyter Notebook parecen ser pocas, pero entre esas, está la dificultad del trabajo colaborativo, así como la incapacidad de recuperar una celda que se haya borrado si es que no se guardó correctamente. Existen otras pequeñas limitaciones pero, la comunidad siempre está a la vanguardia buscando mantenerlas muy pequeñas, esto gracias a otra bondad de este entorno, que es su código libre.

5 Bibliografía

Willems, K. (2016) *Jupyter Notebook Tutorial: The Definitive Guide*. Recuperado el 5 de Febrero del 2018 desde <https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook>

Alouini, Y. (2017) *What are the pros and cons of using Python Jupyter versus a normal Python development environment?*. Recuperado el 6 de Febrero del 2018 desde <https://www.quora.com/What-are-the-pros-and-cons-of-using-Python-Jupyter-versus-a-normal-Python-development-environment>

Wikipedia (s.f.) *Python*. Recuperado el 6 de Febrero del 2018 desde <https://es.wikipedia.org/wiki/Python>

Ingargiola, A. (2015) *What is the Jupyter Notebook?*. Recuperado el 6 de Febrero del 2018 desde http://jupyternotebookbeginnerguidereadthedocs.io/en/latest/what_is_jupyter.html