

Reporte Actividad 2: Introducción a Python, Jupyter y Pandas

Melissa Matrecitos Avila

07 de Febrero de 2018

1 Introducción



Figure 1: Logo de Python

El siguiente texto es el reporte de la actividad 02 del curso de Física Computacional, el cual es una introducción en el lenguaje de programación Python con el entorno de programación Jupyter Notebook, así como Pandas y Matplotlib. Para empezar debemos saber qué es con los que vamos a trabajar. Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios

estilos: programación orientada a objetos, programación imperativa y programación funcional. Python es un lenguaje interpretado, es decir sólo realiza la traducción a medida que sea necesaria, típicamente, instrucción por instrucción, a diferencia de los compiladores que traducen un programa desde su descripción en un lenguaje de programación al código de máquina del sistema. Para el análisis de datos en Python, se utiliza un paquete llamado "Pandas", el cual proporciona estructuras de datos rápidas, flexibles y expresivas diseñadas para que trabajar con datos sea fácil e intuitivo. Los principales tipos de datos que pueden representarse con pandas son: datos tabulares con columnas de tipo heterogéneo con etiquetas en columnas y filas, y series temporales.

Ahora con respecto a Jupyter Notebook, es una aplicación web que permite crear y compartir documentos que contienen código fuente, ecuaciones, visualizaciones y texto explicativo. Sus dos componentes principales son el kernel y dashboard. Un kernel es un programa que ejecuta e introspecta el código del usuario. La aplicación Jupyter Notebook tiene un núcleo para el código Python, pero también hay núcleos disponibles para otros lenguajes de programación. El dashboard de la aplicación no solo muestra el documento que ha creado, sino que también se puede utilizar para administrar los kernels:



Figure 2: Logo de Jupyter

puede ver cuáles se están ejecutando y apagarlos de ser necesario.

2 Actividades a realizar

2.1 Respecto al ejemplo

```
# Gráfica de la rapidez de los vientos (m/s)
plt.figure(); df.VELS.plot(); plt.legend(loc='best')
plt.title("Variación de la Rapidez de los Vientos")
plt.ylabel("Rapidez (m/s)")
plt.grid(True)
plt.show()
```

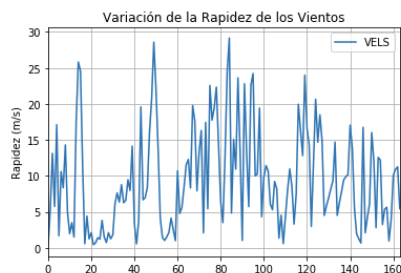


Figure 3: Gráfica de Velocidad de Viento y Velocidad de Rafagas

En la primera parte de la actividad, donde se replicó el contenido del ejemplo brindado por el profesor, se vieron las bases de la programación en python, como cargar a la memoria de trabajo las bibliotecas, leer archivos, mostrar los primeros o últimos datos del archivo, dar estructura a los datos, ver los tipos de datos que el paquete Panda reconoce, combinar columnas, convertir variables, realizar análisis exploratorios de los datos, seleccionar rangos de datos acotando una variable, calcular promedios y hacer gráficas. Todas las acciones se incluyen en la Figura 4, excepto por la sección de gráficas, la cual aparece en la Figura 3. Para realizar las gráficas se utilizó

```
# 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
#

# Dar estructura de datos (DataFrame)
df = pd.DataFrame(df0)

# Ver los tipos de datos que Pandas ha reconocido al leer
df.dtypes

# 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)

# Realiza un análisis exploratorio de datos
df.describe()

# Selecciona los renglones con Temperatura > 24°C y < 25°C
df_tmp = df[df.TEMP > 24]
df_select = df_tmp[df_tmp.TEMP < 25]
df_select

# Descarga los datos de una estación del Servicio Meteorológico Nacional
# http://smn1.conagua.gob.mx/emas/
# Lee un archivo de texto con la función Pandas "read_csv", con elementos separados por mas de
# un espacio, brincándose 4 renglones del inicio (encabezados)
df0 = pd.read_csv('Jalpan.TXT', skiprows=4, sep='\s+')
# "Shift + Enter"

# Lee los primeros 5 renglones del archivo
df.head()
df.tail()
# "Shift+Enter"

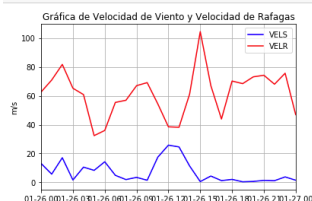
# Calcula el promedio de las columnas, excepto en la FECHA (que no tendría sentido)
df.mean()
```

Figura 4. Actividades realizadas

2.2 Actividades Adicionales

2.2.1 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?

```
# Gráfica de Velocidad de Viento y Velocidad de Ráfagas
plt.plot_date(x=df.FECHA, y=df.VELS, fmt='b-')
plt.plot_date(x=df.FECHA, y=df.VELR, fmt='r-')
plt.legend(loc='best')
plt.xlim(("26/01/2018 00:00", "27/01/2018 00:00"))
plt.title("Gráfica de Velocidad de Viento y Velocidad de Ráfagas")
plt.ylabel("m/s")
plt.grid(True)
plt.show()
```



Se puede observar que por la tarde, aproximadamente entre las 16:00-18:00 horas es cuando los vientos tienen mayor rapidez, aun que también por la mañana, a las primeras horas del día se forman picos en las gráficas, lo que quiere decir que también hay un aumento en su rapidez, pero no tan notable como en la tarde.

También se puede observar en la gráfica que los vientos (color azul) siempre tienen mayor velocidad que las ráfagas (color rojo).

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

Claramente se nota como la mayoría de los vientos mantienen una dirección constante al rededor de 150° , aun que se puede destacar como en ciertos periodos, la dirección cambia brutalmente, aumentando incluso hasta un poco más de los 350° . Sin embargo también se ven esos cambios drásticos en disminución, cayendo hasta los 0° .

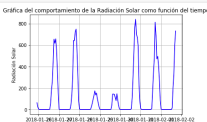
No hay que olvidar que los datos en la gráfica son los de una semana, por lo que probablemente los cambios de dirección sean debido a las ráfagas de viento que hay en cada día de la semana.

```
# Gráfica de la dirección de los vientos como función del tiempo
plt.plot_date(x=df.FECHA, y=df.DIR, fmt='b-')
plt.title("Gráfica de la dirección de los vientos como función del tiempo")
plt.ylabel("Grados")
plt.grid(True)
plt.show()
```



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

```
# Gráfica del comportamiento de la Radiación Solar como función del tiempo.
plt.plot_date(x=df.FECHA, y=df.ADPOL, fmt='b-')
plt.title("Gráfica del comportamiento de la Radiación Solar como función del tiempo")
plt.ylabel("Radiación Solar")
plt.grid(True)
plt.show()
```



Al igual que en la gráfica anterior, se pueden observar los "picos", los cuales llegan hasta más de 800, de manera casi constante cada cierto intervalo de tiempo, lo cual también se puede deber a que los datos en el eje x son varios días y que los

"picos" se pueden dar a las mismas horas del día.

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

Para poder contestar esta parte de la actividad, se tuvieron que separar un conjunto de datos, ya que se necesitaba solo un día, esto se logro mediante la entrada que me nuestra debajo del párrafo. Después se buscó el máximo y mínimo del conjunto de datos para verificar que fueran los datos correctos, Para finalizar, se realizó la resta de la temperatura máxima menos la mínima para así poder encontrar la diferencia que se buscaba.

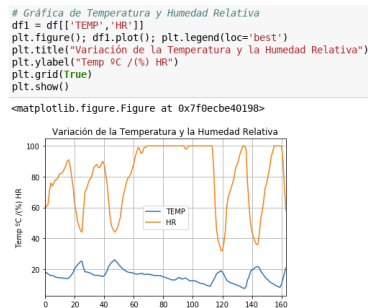
```
#Datos de un día
df_dia = df[df.FECHA >= '26/01/2018 00:00']
df_select = df_dia[df_dia.FECHA < '27/01/2018 00:00']
df_select
```

```
Tmax=df_select.TEMP.max()
Tmax
25.199999999999999

Tmin=df_select.TEMP.min()
Tmin
13.800000000000001

DTemp=Tmax-Tmin
DTemp
11.399999999999999
```

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



Se nota una clara relación entre ambas, ya que los puntos más altos de la humedad relativa, corresponden a los más bajos de la temperatura, y de igual forma, los puntos más alto de temperatura corresponden a los más bajos de la humedad.

Incluso en el lapso de tiempo en el que la humedad relativa se mantiene constante, la temperatura también se mantiene sin cambios tan bruscos como los que había presentado antes.

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

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

	DIRS	DIRR	VELS	VELR	TEMP	HR	PB	PREC	RADSOL	NIVEL
count	164.000000	164.000000	164.000000	164.000000	164.000000	164.000000	164.000000	164.0	164.000000	164.000000
mean	137.835366	205.939024	9.431707	57.098171	15.504878	80.347561	932.492073	0.0	128.796951	16.729183
std	97.800741	90.771992	6.982523	14.616901	4.079348	20.511734	4.003832	0.0	225.356187	0.013924
min	0.000000	0.000000	0.290000	27.700000	7.400000	32.000000	925.200000	0.0	0.000000	16.700000
25%	57.750000	164.750000	3.890000	46.350000	12.900000	66.500000	929.275000	0.0	0.000000	16.716000
50%	151.000000	177.000000	8.115000	57.400000	15.750000	86.000000	932.500000	0.0	0.000000	16.733000
75%	168.000000	180.000000	13.967500	67.000000	17.725000	100.000000	934.625000	0.0	143.350000	16.741000
max	359.000000	359.000000	29.150000	104.400000	26.100000	100.000000	941.100000	0.0	839.500000	16.751000

3 Apéndice

1. ¿Cuál es tu primera impresión de Jupyter Notebook? Al inicio me costó trabajo familiarizarme con el entorno, ya que era mi primera vez trabajando en un ambiente así, sin embargo con la practica me acostumbré y me pareció más sencillo de trabajar.
2. ¿Se te dificultó leer código en Python? No se me dificultó tanto por los comentarios que cada entrada tenía, además de que se utilizan palabras que tienen un significado literal.
3. ¿En base a tu experiencia de programación en Fortran, que te parece el entorno de trabajar en Python? Me parece un entorno más amigable y sencillo de aprender.
4. A diferencia de Fortran, ahora se producen las gráficas utilizando la biblioteca Matplotlib. ¿Cómo fue tu experiencia?. Me pareció excelente como es que se hacían las gráficas en Matplotlib, por que se generaban al momento y se podían personalizar facilmente.
5. En general, ¿qué te pareció el entorno de trabajo en Python? Por el momento me parece sencillo de utilizar.
6. ¿Qué opinas de la actividad? ¿Estuvo compleja? ¿Mucho material nuevo? ¿Que le faltó o que le sobró? ¿Qué modificarías para mejorar? Me parece un buen comienzo el haber tomado como ejemplo un código ya listo, por que así sé que estoy haciendo y no te pierdes con facilidad. Siento que fue mucha información nueva y sería conveniente hacer como un resuemen de las partes más importantes, algo como resultados sobresalientes.
7. ¿Comentarios adicionales que desees compartir? Me hubiera gustado que tuvieramos una sesión más en el laboratorio por la que se perdió el día lunes.

4 Bibliografía

- <https://bioinf.comav.upv.es/courses/linux/python/pandas.html>
- <https://pandas.pydata.org/pandas-docs/stable/index.html>
- <https://es.wikipedia.org/wiki/Python>
- <https://www.datacamp.com/community/tutorials/tutorial-jupyter-notebook>
- https://live.osgeo.org/es/quickstart/jupyter_quickstart.html

4.1 Imágenes

- <https://commons.wikimedia.org/wiki/File:Python.svg>
- <https://gitlab.eurecom.fr/zoe-apps/zapp-jupyter/blob/master/logo.png>