



UNIVERSIDAD DE SONORA

DIVISIÓN DE CIENCIAS EXACTAS Y NATURALES

FÍSICA COMPUTACIONAL

Actividad4: Visualizando datos con Pandas y Matplotlib

Diana Iveth López Atienzo

Profesor: Carlos Lizárraga Celaya

29 septiembre del 2016

1. Introducción

Como pudimos ver en la actividad anterior, python es una herramienta muy útil a la hora de analizar una base de datos. Pero también existen complementos para estos programas que ayudan a facilitar su uso. En esta actividad utilizaremos Pandas, el cual es una herramienta de python que proporciona y analiza datos de manera más rápida. Pandas es muy adecuado para muchos tipos diferentes de datos, principalmente los archivos csv como los de excel. En esta actividad utilizaremos pandas para graficar algunos de los datos que obtuvimos al bajar un día de datos de la página de la universidad de Wyoming.

Actividad4: Visualizando datos con Pandas y Matplotlib

2. Paso I

Primero descargamos los datos de un día de la pagina de la universidad de Wyoming. En mi caso, escogí el dia 21 de septiembre del 2016, en la ciudad de Chihuahua. Después depuramos los datos y creamos un archivo csv.

3. Paso II

Desde la terminal abrimos un nuevo archivo en ipython notebook y lo llamamos Actividad 4. Lo primero que escribimos es el siguiente código, el cual sirve para dar especificaciones de los comandos, y lo corremos.

```
import pandas as pd
import matplotlib as plt
import numpy as np
import matplotlib.pyplot as mplt
df = pd.read_csv("/home/dilopez/Escritorio/Computacional/Actividad4/Datos2016.csv")
```

```
In [9]: import pandas as pd
import matplotlib as plt
import numpy as np
import matplotlib.pyplot as mplt
df = pd.read_csv("/home/dilopez/Escritorio/Computacional/Actividad4/Datos2016.csv")
```

4. Paso III

En la siguiente celda, ponemos

```
df.describe()
```

y lo corremos. Obtendremos una tabla con todos los datos del archivo csv que creamos.

```
In [10]: df.describe()
Out[10]:
```

	PRES	HGHT	TEMP	DWPT	RELH	MXR	DRCT	SKNT	THTA	THTE	THTV
count	118.000000	118.000000	118.000000	118.000000	118.000000	118.000000	117.000000	117.000000	118.000000	118.000000	118.000000
mean	256.322034	14034.593220	-40.048305	-57.812712	24.355932	1.349153	132.683761	14.547009	414.476271	418.894915	414.736441
std	248.308985	7581.748841	31.617186	35.872577	22.185514	2.762376	112.313957	9.007177	107.540254	103.875476	107.307616
min	18.000000	1372.000000	-76.300000	-99.100000	1.000000	0.000000	0.000000	1.000000	307.200000	331.000000	309.200000
25%	51.475000	6980.500000	-66.025000	-87.500000	1.000000	0.000000	50.000000	7.000000	334.625000	340.825000	334.825000
50%	114.500000	15876.000000	-53.900000	-79.100000	23.500000	0.015000	95.000000	13.000000	366.200000	366.200000	366.200000
75%	434.750000	20592.250000	-9.425000	-27.275000	40.750000	0.922500	176.000000	22.000000	488.300000	488.300000	488.300000
max	863.000000	27247.000000	23.000000	12.400000	82.000000	10.590000	360.000000	35.000000	702.300000	702.500000	702.300000

5. Paso IV

Después escribimos el siguiente comando y lo corremos. En seguida ponemos aparecerán los nombres que utilizaremos para cada columna al momento de escribir un código.

```
df.columns
```

```
In [15]: df.columns
Out[15]: Index([u' PRES', u' HGHT', u' TEMP', u' DWPT', u' RELH', u' MIXR', u' DRCT', u' SKNT', u' THTA', u' THTE', u' THT V'], dtype='object')
```

6. Paso V

En la siguiente celda colocamos

```
%matplotlib inline
```

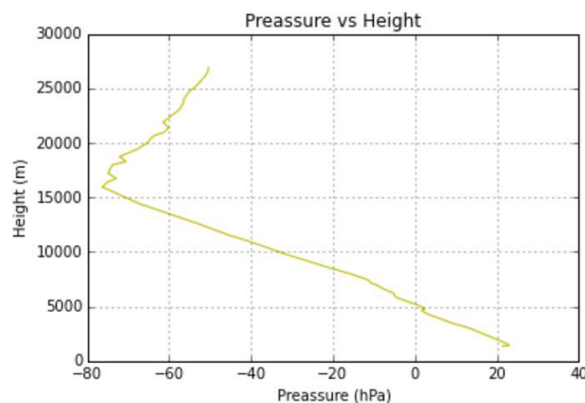
lo cual nos permitirá usar la siguiente función

```
x = df_complete[u' TEMP']
y = df_complete[u' HGHT']

matplotlib.title('Preassure vs Height')
matplotlib.xlabel('Preassure (hPa)')
matplotlib.ylabel('Height (m)')
matplotlib.grid(True)
```

```
matplotlib.plot(x,y, c='y')
```

Cuando la corremos, nos da una gráfica de Presión (hPa) vs. Altura (m)



7. Paso VI

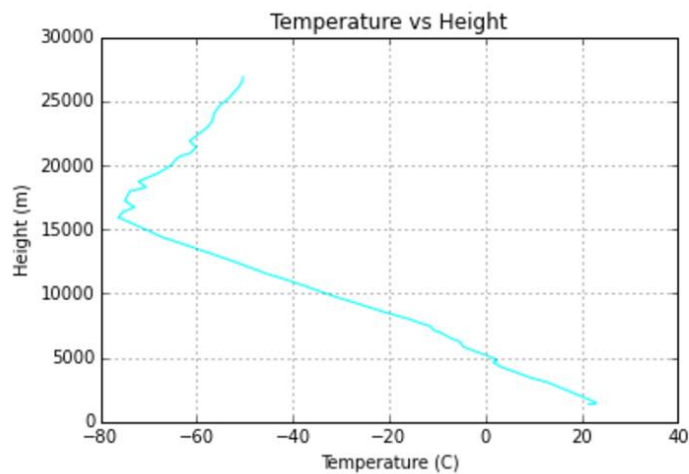
Después colocamos

```
x = df_complete[u' TEMP']
y = df_complete[u' HGHT']

plt.title('Temperature vs Height')
plt.xlabel('Temperature (C)')
plt.ylabel('Height (m)')
plt.grid(True)

plt.plot(x,y,c='aqua')
```

Y aparecera una gráfica de Temperatura (°C) vs. Altura (m)



8. Paso VII

Ahora colocamos el siguiente código

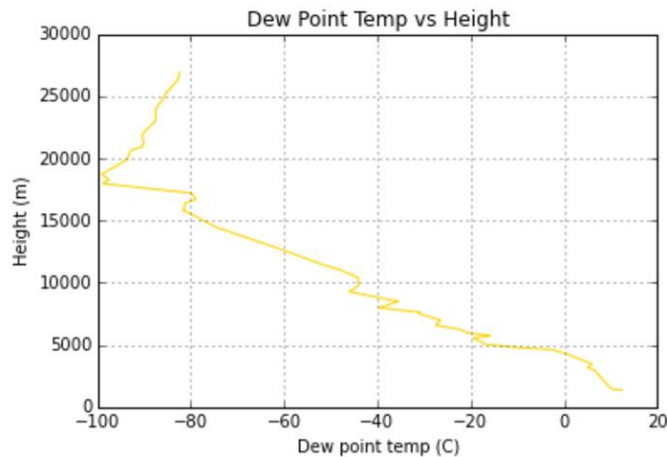
```
x = df_complete[u' DWPT']
y = df_complete[u' HGHT']

plt.title('Dew Point Temp vs Height')
plt.xlabel('Dew point temp (C)')
plt.ylabel('Height (m)')
```

```
mplt.grid(True)
```

```
mplt.plot(x,y,c='gold')
```

El cual nos graficará Temperatura de Rocío (DWPT °C) vs. Altura (m)

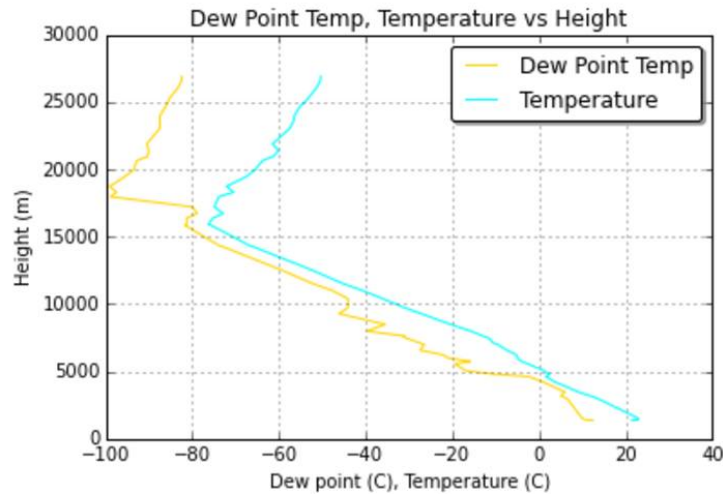


9. Paso VIII

Y por último, corremos el siguiente código, para obtener las gráficas Temperatura y Temperatura de Rocío en una sola gráfica.

```
mplt.title('Dew Point Temp, Temperature vs Height')
mplt.xlabel('Dew point (C), Temperature (C)')
mplt.ylabel('Height (m)')
mplt.grid(True)
```

```
mplt.plot(df_complete[u' DWPT'],df_complete[u' HGHT'], 'gold',label="Dew Point Tem
mplt.plot(df_complete[u' TEMP'],df_complete[u' HGHT'], 'aqua',label="Temperature")
mplt.legend(fancybox=True,shadow=True)
```



10. Conclusión

Python es una herramienta muy útil a la hora de editar documentos que contienen y con la ayuda de pandas, es mucho más fácil llevar a cabo muchos procesos, ya sean de tabulación, graficación entre muchos otros.

11. Bibliografía

University of Wyoming/ College of Engineering/ Department of Atmospheric Science : [http : //weather.uwyo.edu/upperair/sounding.html](http://weather.uwyo.edu/upperair/sounding.html)

Pydata/ Python Data Analysis Library : [http : //pandas.pydata.org/pandas-docs/stable/tutorials.h](http://pandas.pydata.org/pandas-docs/stable/tutorials.h)