



"El saber de mis hijos
hará mi grandeza"

Universidad de Sonora

Departamento de Física

Licenciatura en Física

Física Computacional-1

2016-2

Visualizando datos con Pandas y Matplotlib

Danira Rios Quijada

Profesor: Carlos Lizárraga Celaya

29 de septiembre de 2016

Resumen

En el siguiente reporte de trabajo, se describe a detalle los pasos que seguimos en la actividad 4 de Física computacional, para poder analizar datos graficamente, utilizando Python y la biblioteca Mathplotlib. Los datos que utilizamos los descargamos de la página de ciencias atmosféricas de la Universidad de Wyoming[2], las gráficas concistieron en graficar:

- Presión (hPa) vs. Altura (m)
- Temperatura ($^{\circ}\text{C}$) vs. Altura (m)
- Temperatura de Rocío (DWPT $^{\circ}\text{C}$) vs. Altura (m)
- Gráficas de Temperatura y Temperatura de Rocío en un solo gráfico.

1. Introducción

El análisis puramente estadístico de los datos, no siempre es suficiente, a veces es necesario hacer un análisis visual de los datos, dependiendo de la información que necesitemos obtener, para hacer este tipo de análisis con Python, regularmente se utiliza la biblioteca Matplotlib, la cual nos brinda muchas alternativas para representar series de datos visualmente.

2. Descargando un día de datos y preparando el archivo

El archivo de datos que descargamos, lo descargamos directamente de la página web de la universidad de Wyoming [2], entonces el archivo no estaba separado por comas (csv), si no que después de limpiarlo con emacs, nos quedaba un archivo separado por espacios, pero el problema era que cada dato difería en el número de espacios de separación;

Yo opté por tratar el archivo con una hoja de cálculo y así poder definir las columnas manualmente, para después exportarlo como un archivo csv, con comas como separación, pero también existe una función en Python, para trabajar con archivos de este tipo, se utiliza de la siguiente forma:

```
pd.read_csv ("ruta del archivo/archivo.csv",sep='\s+', header=None)
```

Lo que indica que como separador, el archivo presenta uno o más espacios.

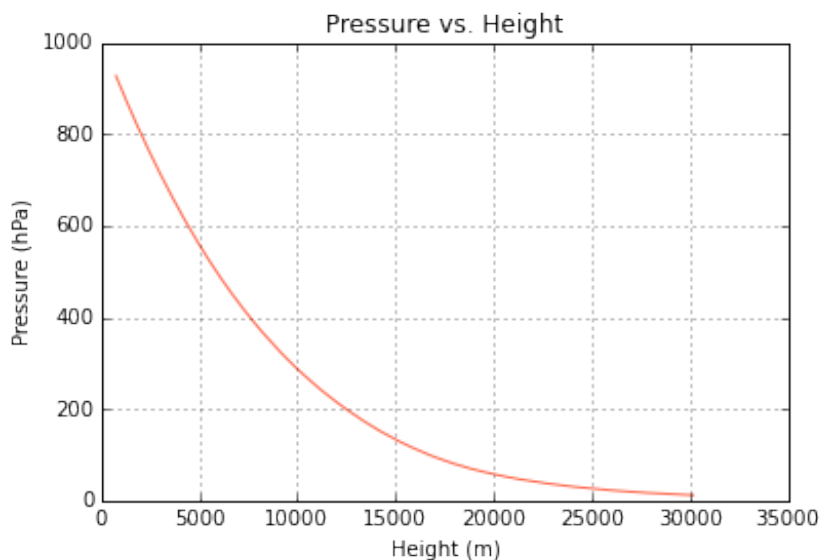
3. Graficando plots básicos Y vs X

- Presión (hPa) vs. Altura (m)

Este gráfico consistió en poner a la presión como función de la altura, el código que utilizamos fue:

```
#Presión (hPa) vs. Altura (m)
y = df[u'PRES']
x = df[u'HGHT']
mplt.plot(x, y, c='tomato')
mplt.ylabel('Pressure (hPa)')
mplt.xlabel('Height (m)')
mplt.title('Pressure vs. Height')
mplt.grid(True)
```

Donde definimos x,y, como los datos de las columnas HGHT y PRES , hacemos el plot, etiquetamos los ejes, damos un título al gráfico y ponemos cuadrícula, obteniendo:

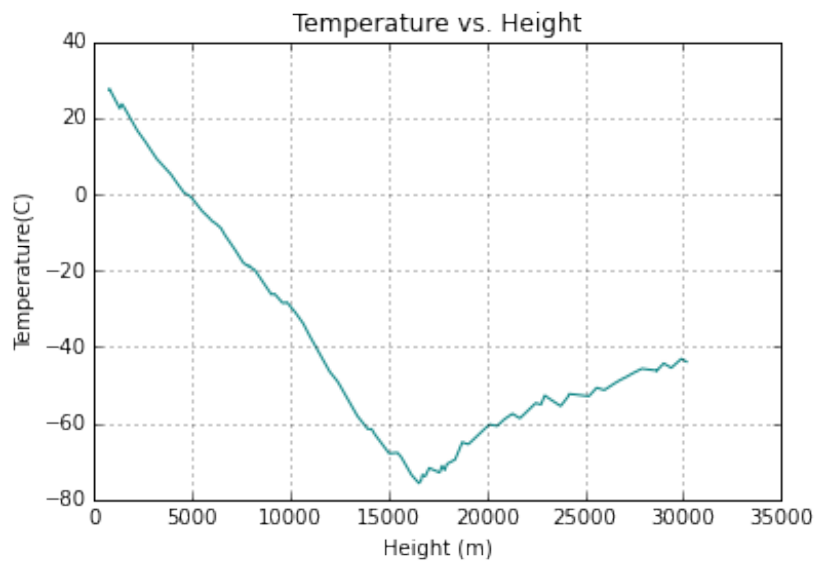


- Temperatura (°C) vs. Altura (m)

Este gráfico consistió en poner a la temperatura como función de la altura, el código que utilizamos fue:

```
#Temperatura (°C) vs. Altura (m)
y= df[u'TEMP']
x= df[u'HGHT']
plt.plot(x, y, c='teal')
plt.ylabel('Temperature(C)')
plt.xlabel('Height (m)')
plt.title('Temperature vs. Height ')
plt.grid(True)
```

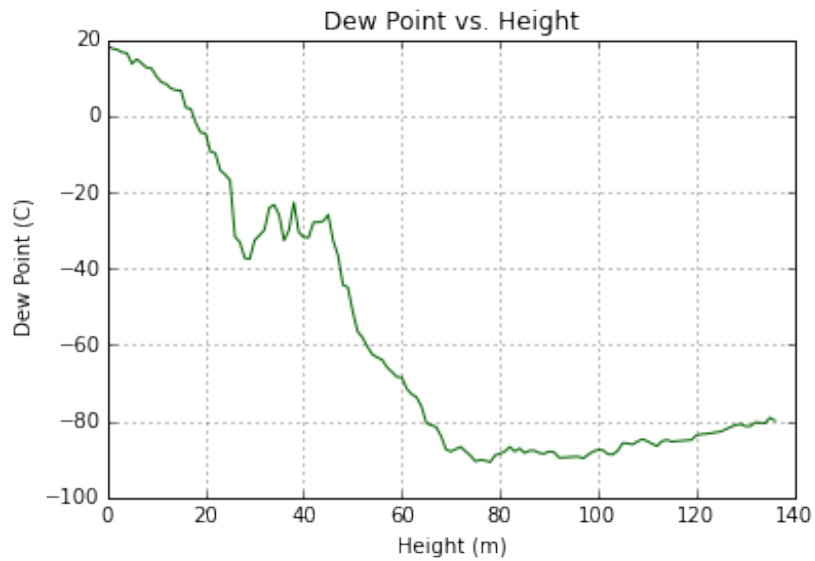
Donde definimos x,y, como los datos de las columnas HGHT y TEMP, hacemos el plot, etiquetamos los ejes, damos un título al gráfico y ponemos cuadrícula, obteniendo:



- Temperatura de Rocío (DWPT °C) vs. Altura (m)
Este gráfico consistió en poner a la temperatura de rocío como función de la altura, el código que utilizamos fue:

```
#Temperatura de Rocío (DWPT °C) vs. Altura (m)
y = df[u'DWPT']
x = df[u'HGHT']
plt.plot(y, c='darkgreen')
plt.ylabel('Dew Point (C)')
plt.xlabel('Height (m)')
plt.title('Dew Point vs. Height')
plt.grid(True)
```

Donde definimos x,y, como los datos de las columnas HGHT y DWPT, hacemos el plot, etiquetamos los ejes, damos un título al gráfico y ponemos cuadrícula, obteniendo:



4. Graficando “Temperature” y “Dew Point” en un solo gráfico.

Temperatura de Rocío (DWPT °C) vs. Altura (m)

Este gráfico concistió en graficar la temperatura y a la temperatura de rocío en el eje X y a la altura en el eje Y, el código que utilizamos fue:

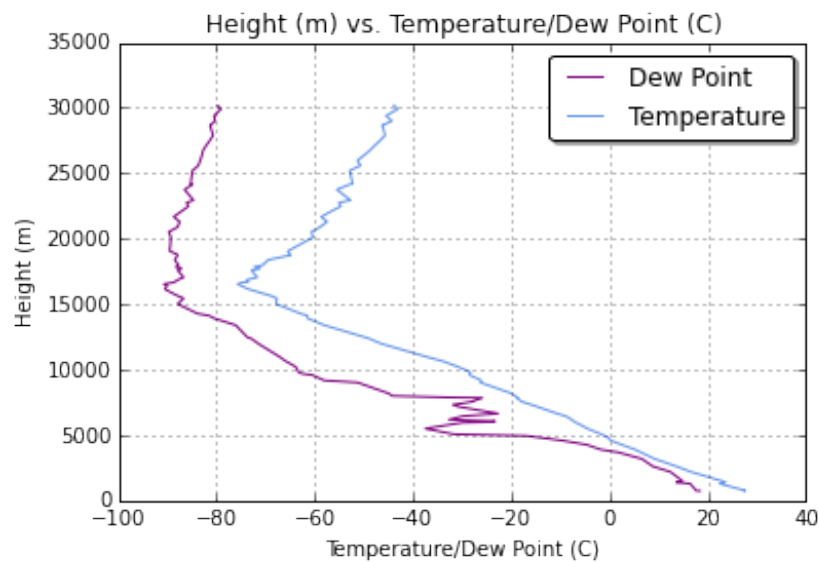
#Gráficas de Temperatura y Temperatura de Rocío en una sola gráfica.

```
y = df[u'HGHT']
x1 = df[u'DWPT']
x2 = df[u'TEMP']

mplt.title('Height (m) vs. Temperature/Dew Point (C)')
mplt.ylabel('Height (m)')
mplt.xlabel('Temperature/Dew Point (C)')

mplt.grid(True)
mplt.plot(x1,y,c='purple',label='Dew Point')
mplt.plot(x2,y,c='cornflowerblue', label='Temperature')
mplt.legend(fancybox=True, shadow=True)
```

Donde definimos Y, como los datos de la columna HGHT y definimos dos X, como los datos de las columnas TEMP y DWPT, antes de hacer el plot, dimos título al gráfico y etiquetamos los ejes, después hicimos un doble plot, X1,Y y X2,Y, para que de esta forma cada curva pudiese ser etiquetada independientemente, utilizamos la función legend, para que apareciera la etiqueta de cada curva[3], obteniendo:



5. Bibliografía

Referencias

- [1] Carlos Lizárraga Celaya, *Actividad cuatro, curso de computacional 1*, (2016, 27 de septiembre).Recuperado (2016, 27 de septiembre), Desde: <http://computacional1.pbworks.com/w/page/111225265/Actividad4>
- [2] University of Wyoming; Department of Atmospheric Science *Datos, sondeo atmosférico*, (2015), recuperado (2016, 25 de septiembre). Desde: <http://weather.uwyo.edu/upperair/sounding.html>
- [3] Matplotlib, *lines bars and markers example code*, (2016, 20 de septiembre). Recuperado (2016,27 de septiembre), Desde: http://matplotlib.org/examples/lines_bars_and_markers/scatter_with_legend.html