



UNIVERSIDAD DE SONORA

DIVISIÓN DE CIENCIAS EXACTAS Y NATURALES

DEPARTAMENTO DE FÍSICA

LICENCIATURA EN FÍSICA

FÍSICA COMPUTACIONAL I

Visualizando datos con Pandas y Matplotlib

Martinez López Lizbeth Vanessa

Profesor del curso
Dr. Carlos Lizárraga

27 de Febrero del 2017

Resumen

En el siguiente reporte se muestran los resultados obtenidos al usar la biblioteca matplotlib y pandas para visualizar y graficar datos. En este caso las gráficas son de variables respecto a otras, para analizar el comportamiento entre dichas variables. También, se presenta un tefigrama.

1. Introducción

Actualmente el desarrollo tecnológico ha permitido que existan muchas herramientas para visualizar y graficar datos. Por medio de Python se puede graficar e incluso es muy completo, pues se cuenta con bibliotecas o paquetes de distintos gráficos que pueden ser de utilidad.

En esta práctica el manejo de las bibliotecas de Python fue una clave importante para la buena realización de dicha actividad. Esta ocasión, la finalidad de la práctica es mostrar por medio de gráficas el comportamiento que tiene la presión, temperatura y temperatura de rocío de la atmósfera, respecto a la altura; nuevamente se pretende demostrar que lo mencionado en la actividad 1 sucede realmente.

Los datos utilizados para esta actividad son los obtenidos por la estación de Albuquerque el día 15 de febrero del 2017.

2. Desarrollo

Para llevar a cabo esta actividad, lo primero que se debió hacer, fue extraer los datos de sondeo de la página de la Universidad de Wyoming, para ello se hizo uso

del script que en anteriores actividades utilizamos, sólo fue necesario editar el link para poder compilar el script y obtener los datos del día que se requería.

Ya con el archivo de datos, por medio de Emacs se hace limpieza, quitando los espacios entre columnas y sustituyendo por comas, para así crear un archivo .csv. Este archivo es con el cual se trabajó en jupyter notebook para la creación de gráficas. Jupyter Notebook es una aplicación web que permite crear y compartir documentos con código en vivo, ecuaciones, visualizaciones y texto explicativo.

En Jupyter se empleó el código necesario para la creación de gráficas de variables respecto a otras. Como se mencionó anteriormente, en esta práctica se hizo uso de bibliotecas, las bibliotecas usadas fueron:

1. **Matplotlib:** Es una biblioteca de graficación que produce figuras de publicación de calidad en una variedad de formatos impresos y entornos interactivos a través de plataformas.
2. **Pandas:** Es una biblioteca de código abierto, con licencia BSD que proporciona estructuras de datos de alto rendimiento y fácil de usar y herramientas de análisis de datos para el lenguaje de programación Python.
3. **Numpy:** Es un paquete de procesamiento de matrices de uso general, diseñado para manipular de forma eficiente grandes matrices multidimensionales de registros arbitrarios, sin sacrificar demasiada velocidad para matrices multidimensionales pequeñas. NumPy se basa en la base de código numérico.

Código básico utilizado para graficación

Primero se usó el código para la importación de las bibliotecas:

```
import pandas as pd
```

```
import numpy as np
import matplotlib.pyplot as plt
import pylab as plt
```

PyLab permite darle características a las gráficas, como el nombre de los ejes. Pyplot permite que las gráficas se creen con matplotlib y no se use algún otro programa. Pandas permite leer los archivos y visualizarlos.

```
df = pd.read_csv("/home/vanemtzl/Actividad 4/15feb17.csv")
```

Con esta parte de código, leemos los datos del archivo de sondeos para el día seleccionado.

```
df.columns= ['Presión', 'Altura', 'Temperatura', 'DWPT', 'RELH', 'MIXR', 'DRCT',
'SKNT', 'THTA', 'THTE', 'THTV']
```

Aquí se definen las columnas como variables. Ya con las variables definidas, se inicia la graficación, primero definiendo cuál es la variable en x y cuál en y. Un ejemplo del código empleado es el siguiente:

```
x=df[u'Presión']
y=df[u'Altura']
```

Después, le damos las características a la gráfica, como qué es lo que se quiere graficar y ponerle el nombre a los ejes.

```
matplotlib.pyplot(x,y)
matplotlib.grid(True)
plt.xlabel('Presión (hPa)')
plt.ylabel('Altura (m)')
```

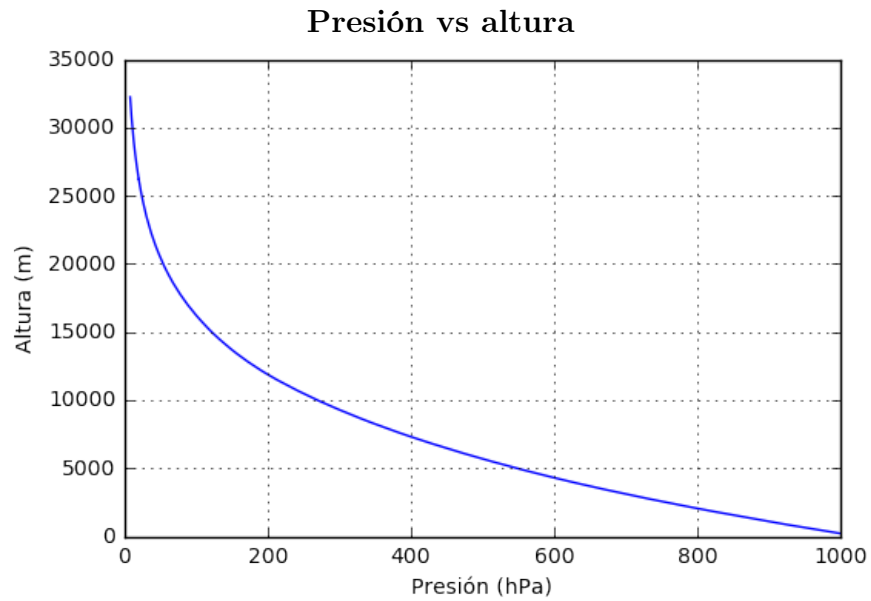
Finalmente, para mostrar la gráfica, se usa lo siguiente:

```
plt.show()
```

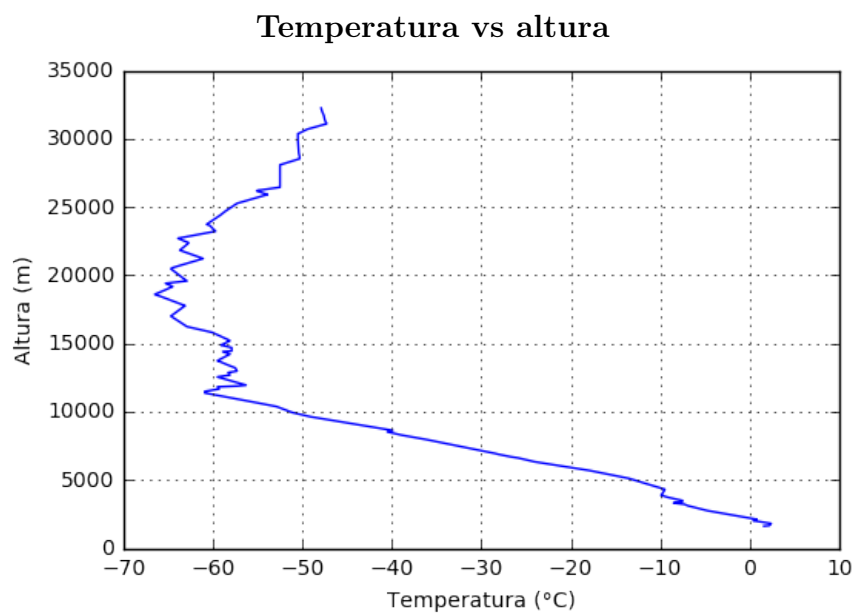
Este es el código básico que se utilizó para hacer las gráficas pedidas en la actividad.

3. Gráficas

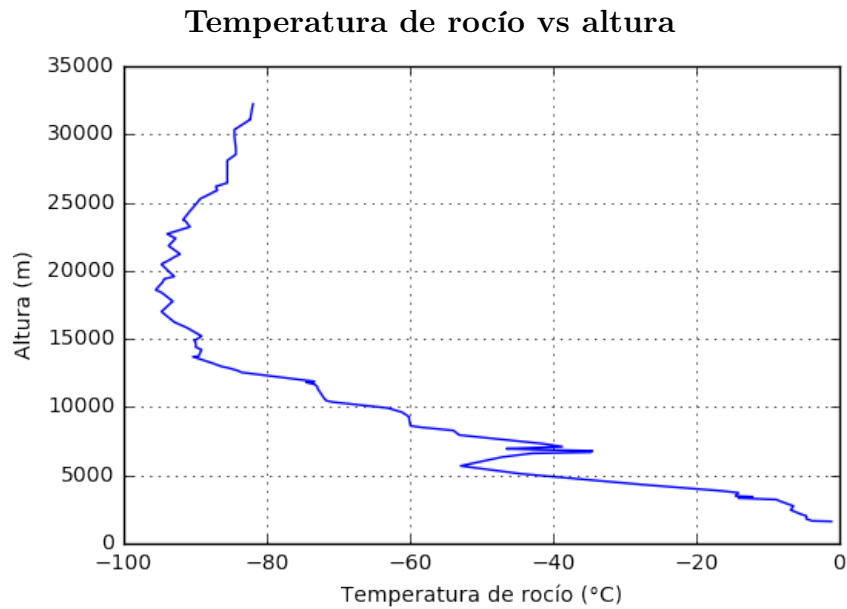
A continuación, se muestran las gráficas obtenidas en Jupyter:



Como se mencionó en anteriores actividades, la presión atmosférica disminuye conforme la presión aumenta, ésto se puede ver claramente en la gráfica, además la forma en la que disminuye es exponencial.

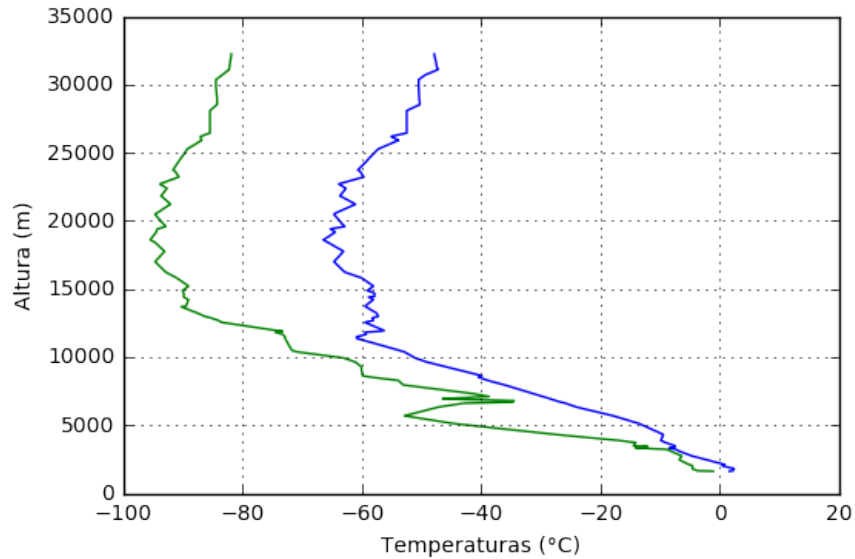


En el caso de la temperatura, tiene un comportamiento más variado, pues en ciertas regiones disminuye conforme la altura aumenta, y en otras regiones la temperatura aumenta conforme la altura lo hace, sin embargo, se puede observar que las temperaturas en 35000 m de altura, se mantienen en valores bastante bajos.



El comportamiento de la temperatura de rocío vs la altura, se puede describir con lo dicho en el párrafo anterior, pues es un comportamiento similar, tal vez los valores no son iguales, pero también podemos concluir que en ciertas regiones disminuye y en otras aumenta con respecto a la altura. También se mantiene en temperaturas muy bajas, incluso más que la temperatura normal.

Temperatura y temperatura de rocío vs altura

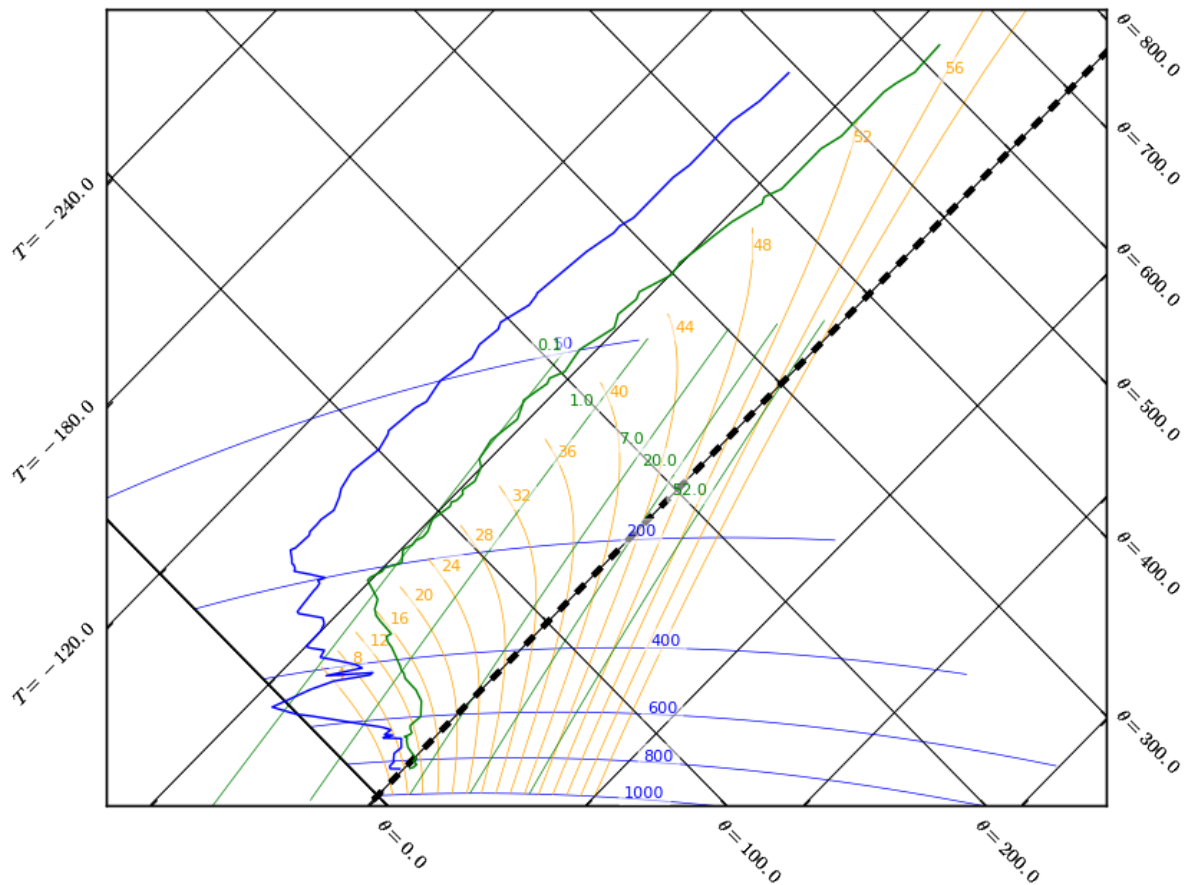


Verde: Temperatura de rocío. Azul: Temperatura.

Aquí se muestra una gráfica que nos permite comparar con mayor facilidad la temperatura y la temperatura de rocío de la atmósfera. Se puede observar que la temperatura de rocío disminuye, en ciertas regiones, más rápido que la otra temperatura. Además, la temperatura de rocío se mantiene en valores aún más bajos que la temperatura normal, pues entre 30000 y 35000 metros, la temperatura de rocío se mantiene en -80°C y la temperatura normal en aproximadamente -50°C .

4. Tefigrama

El tefigrama es un diagrama termodinámico empleado para trazar perfiles verticales de temperatura, humedad y viento atmosféricos. El tefigrama elaborado en esta actividad, muestra la temperatura y temperatura de rocío con respecto a la presión. A continuación se muestra el tefigrama obtenido:



“Como la presión atmosférica disminuye de forma logarítmica a medida que aumenta la altitud en la atmósfera, en el tefigrama las líneas de presión constante (isobaras) se trazan en sentido aproximadamente horizontal y van disminuyendo logarítmicamente con la altitud a lo largo del eje de las ordenadas (eje y). Tal orientación produce líneas de temperatura constante y de temperatura potencial oblicuas que emanan del eje de las abscisas (eje x) a un ángulo aproximado de $+45^\circ$ y -45° , respectivamente.”₆

Como se puede observar las líneas azules horizontales, son las denotadas como isobaras, mientras que las líneas inclinadas hacia la derecha son las isotermas (temperatura constante), las líneas inclinadas hacia la izquierda son las adiabáticas secas (no hay cambio de calor) las cuales representan la temperatura potencial constante. Las curvas amarillas son las adiabáticas saturadas, que representan la temperatura

potencial equivalente constante. Por último, las dos líneas (azul y verde) son las gráficas de temperatura de rocío y temperatura con respecto a la presión.

Para poder crear el tefigrama, se hizo lo siguiente: primero se guardó el paquete de tephi por medio de github y la terminal, en la carpeta donde se tienen todos los archivos de la actividad. Después se instaló tephi por medio del comando pip, el código empleado en la terminal fue el siguiente:

```
pip install --user /home/vanemtzl/"Actividad 4"/tephi
```

Ya con tephi instalado, se importó en el archivo de jupyter para comenzar a graficar

```
import tephi as tph
```

Para el tefigrama, se tuvo que hacer dos archivos aparte con las variables a graficar, uno contiene presión y temperatura y otro contiene presión y temperatura de rocío. Estos archivos deben ser leídos por pandas para poder graficar, además se deben definir las variables. El código empleado fue el siguiente:

```
dew_point = pd.read_csv("/home/vanemtzl/Actividad 4/dwpt.csv",  
names=["Presión", "DWPT"])  
dry_bulb = pd.read_csv("/home/vanemtzl/Actividad 4/temp.csv",  
names=["Presión", "Temperatura"])
```

Por último, se grafica con el siguiente código:

```
tpg = tph.Tephigram()  
tpg.plot(dew_point)  
tpg.plot(dry_bulb)  
plt.show()
```

5. Conclusión

La creación de gráficos es de bastante utilidad, pues te permite observar con mayor facilidad el comportamiento de las variables a graficar, en este caso, se pudo ver cómo es que la temperatura, presión y temperatura de rocío de la atmósfera, se comportaron el día 15 de febrero del 2017, con respecto a la altura.

Python brinda bastantes funciones interesantes y fáciles para crear gráficas, tan sólo es cuestión de investigar y conocer el código a emplear. Como se pudo observar en esta actividad, el uso de bibliotecas es importante, pues estas contienen muchas herramientas que nos son útiles, por ejemplo teph, que contiene una nueva forma de graficar datos de sondeo.

6. Referencias

- [1] *<https://jupyter.org/>*
- [2] *<http://matplotlib.org/>*
- [3] *<http://pandas.pydata.org/>*
- [4] *<https://pypi.python.org/pypi/numpy>*
- [5] *<http://weather.wyo.edu/upperair/sounding.html>*
- [6] *http://www.meted.ucar.edu/mesoprim/tephigram_es/navmenu.php*