



UNIVERSIDAD DE SONORA

DIVISIÓN DE CIENCIAS EXACTAS Y NATURALES

FÍSICA COMPUTACIONAL

28 DE FEBRERO DEL 2017

Act 4 Física Computacional: Visualizando datos con Pandas y Matplotlib

Alumna: Chávez Gutiérrez Yanneth Tzitzin
Profesor: Carlos Lizárraga Celaya.

1. Breve resumen

En esta práctica de física moderna se utilizaron herramientas de prácticas anteriores para poder interpretar los datos de la atmósfera en diferentes regiones. Y haciendo uso de estos datos, se representó graficamente mediante la biblioteca en Python de Matplotlib y despues haciendo uso de Tephí de recreo un tefigrama representando los datos de la atmósfera para cierta ciudad y hora específicos.

2. Introducción

Para la realización de esta práctica, se analizaron datos del 15 de febrero en la ciudad de Chihuahua, Chihuahua a la hora 12Z. Los datos se obtuvieron de ".Atmospheric Sounding" de la universidad de Wyoming.

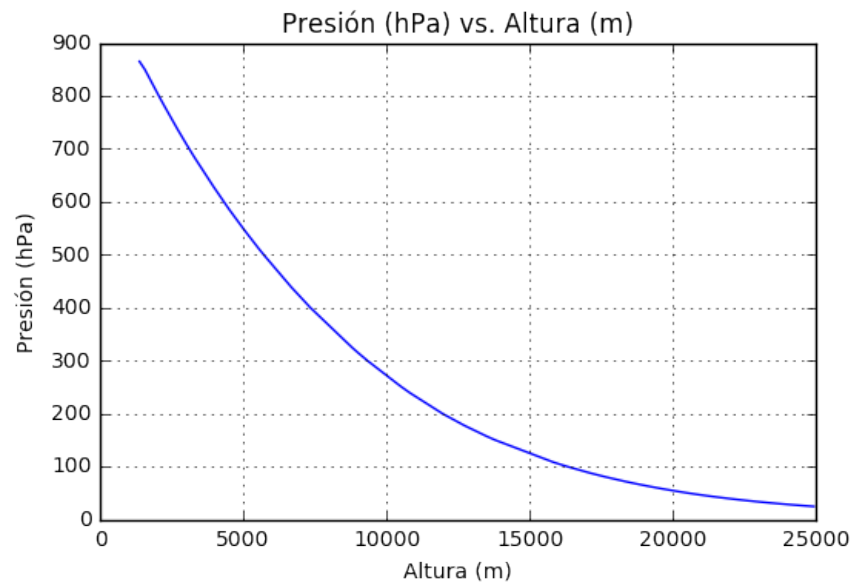
Habiendo descargado los datos de este día, se trabajó con ellos para obtener representaciones graficas del comportamiento de los datos. Haciendo uso de las bibliotecas de Python: *Matplotlib* e instalango *Tephi*, un paquete que ayuda a producir tefigramas. Los cuales son diagramas meteorológicos en el que se representa, mediante coordenadas, las temperaturas absolutas y los logaritmos neperianos de las potenciales.

Resultados

2.1. Primera parte: Utilizando Matplotlib

Con la ayuda de Python y la biblioteca de gráficas *Matplotlib* se realizaron las siguientes graficas para interpretar los datos.

Presión (hPa) vs. Altura (m)

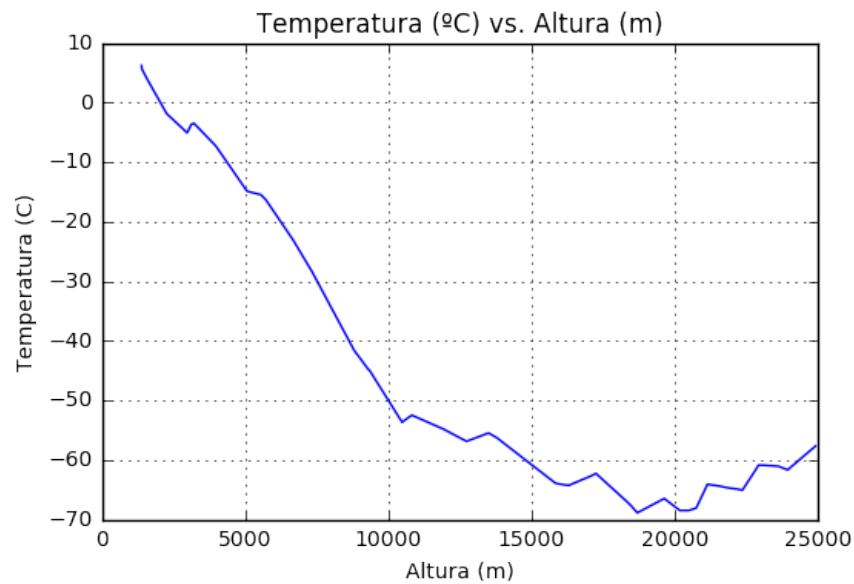


Podemos notar como varia la presión conforme a la altura. Entre mayor sea la altura, menor presión y entre menor sea la altura, mayor presión.

Comandos utilizados:

```
from pylab import figure, show, legend, xlabel, ylabel
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import rc
y=df_clean[u'Pres']
x=df_clean[u'HGHT']
plt.plot(x,y)
plt.grid(True)
pl.xlabel('Altura (m)')
pl.ylabel('Presión (hPa)')
pl.title('Presión (hPa) vs. Altura (m)')
pl.show()
```

Temperatura (°C) vs. Altura (m)

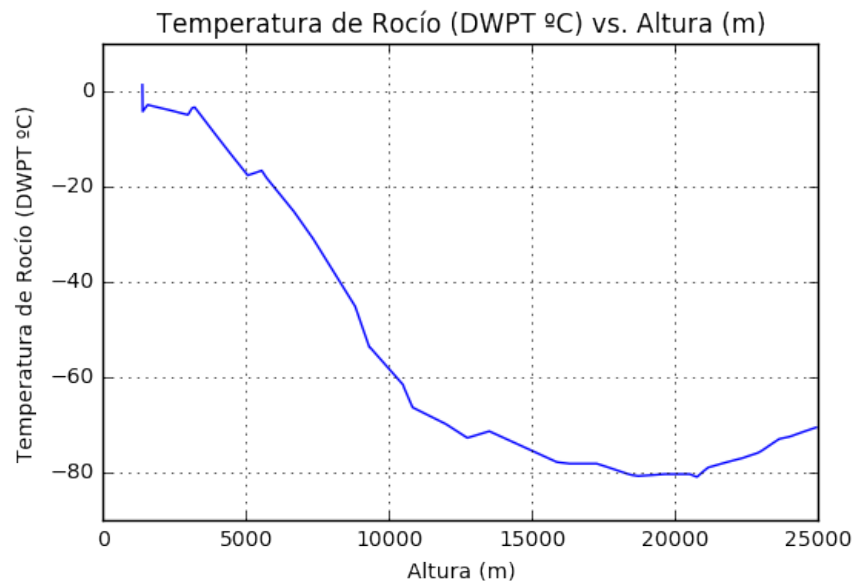


Esta gráfica nos muestra las variaciones de temperatura conforme a la altura, podemos ver que a no tan grandes variaciones de altura, la temperatura muestra muchos cambios.

Comandos utilizados:

```
y=df_clean[u'TEMP']
x=df_clean[u'HGHT']
plt.plot(x,y)
plt.grid(True)
pl.xlabel('Altura (m)')
pl.ylabel('Temperatura (C)')
pl.title('Temperatura (°C) vs. Altura (m)')
pl.show()
```

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

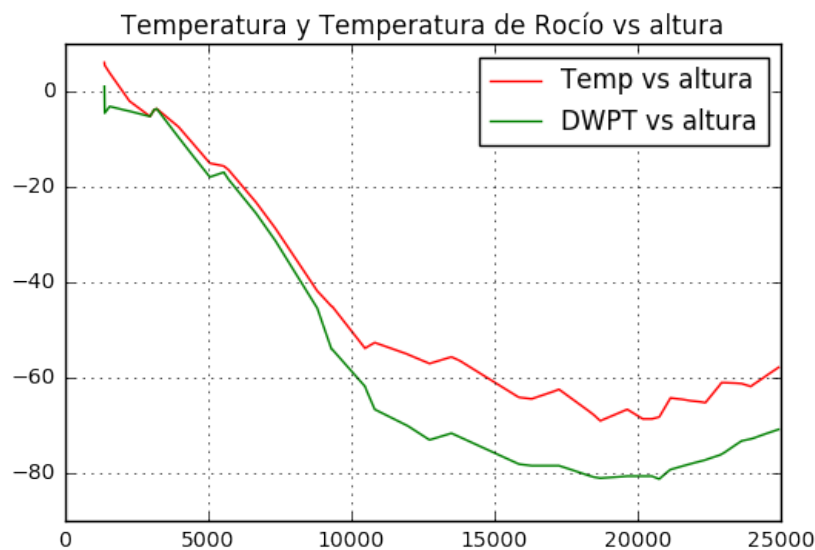


La temperatura de rocío es la temperatura a la que empieza a condensarse el vapor de agua contenido en el aire, produciendo rocío, neblina, cualquier tipo de nube o, en caso de que la temperatura sea lo suficientemente baja, escarcha. Podemos ver que a comparación a la gráfica anterior, esta no muestra cambios muy drásticos de la temperatura conforme a la altura.

Comandos utilizados:

```
y=df_clean[u'DWPT']
x=df_clean[u'HGHT']
mplt.plot(x,y)
mplt.grid(True)
pl.xlabel('Altura (m)')
pl.ylabel('Temperatura de Rocío (DWPT °C)')
pl.title('Temperatura de Rocío (DWPT °C) vs. Altura (m)')
pl.show()
```

Gráficas de Temperatura (Rosa) y Temperatura de Rocío (Verde) vs altura en una sola gráfica



Aquí simplemente se trata de comparar/contrastar los cambios notables de Temp de rocío y Temperatura contra altura en una sola gráfica. La línea azul representa la Temperatura de Rocío, y la línea morada la Temperatura.

Comandos utilizados:

```
y=df_clean[u'TEMP']
x=df_clean[u'HGHT']
plt.plot(x,y, label='Temp vs altura',color='red')
mplt.grid(True)
y=df_clean[u'DWPT']
x=df_clean[u'HGHT']
plt.plot(x,y, label='DWPT vs altura',color='green')
plt.grid(True)
mplt.grid(True)
pl.title('Temperatura y Temperatura de Rocío vs altura')
plt.legend( loc='upper right', numpoints = 1 )
pl.show()
```

Resultados

2.2. Segunda parte: Utilizando Tephi

Pasos a seguir:

1. Primero desde el repositorio en Github, se realizo un Fork del repositorio tephi.
2. Se creo la carpeta Actividad 4 en la computadora para realizar esta actividad.
3. Se entro a la carpeta desde una terminal, y se clono el repositorio de tephi en Github.com, utilizando el comando: git clone y definiendo la carpeta donde se encuentra.

```
git clone https://github.com/Tzitzin/tephi.git
```

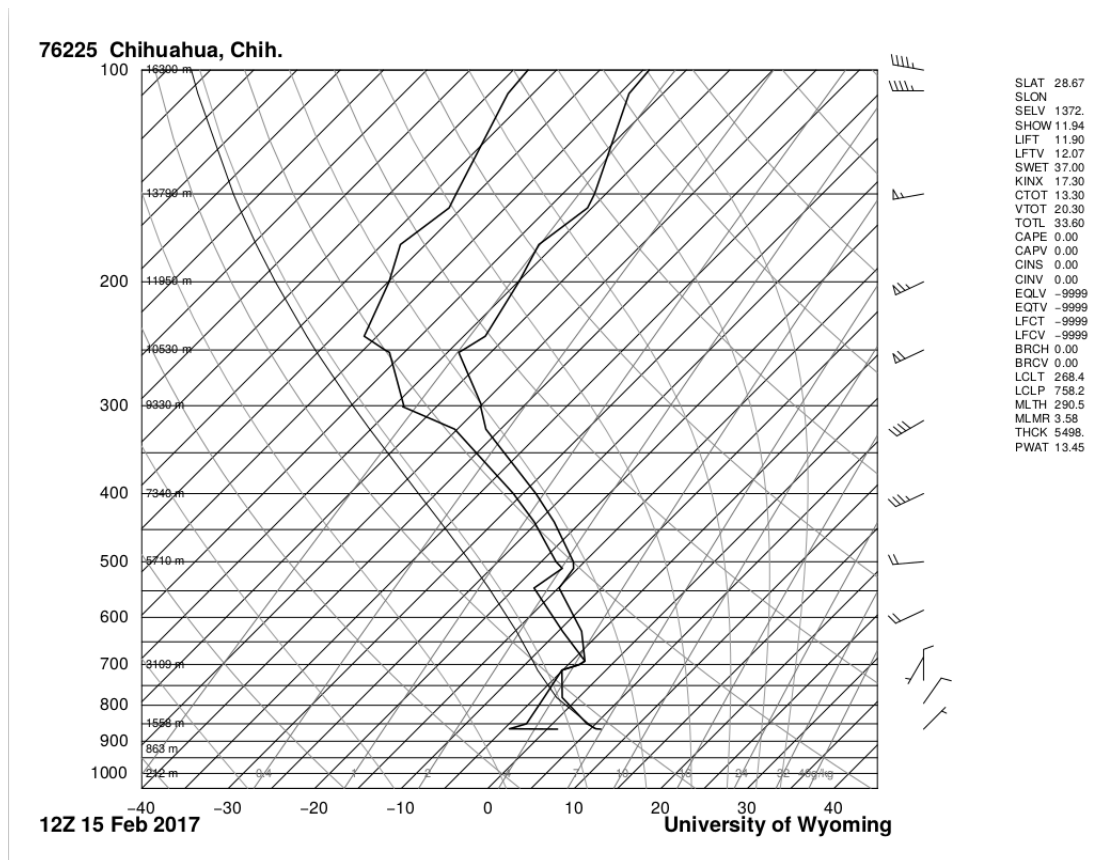
4. Se requerio instalar la biblioteca tephi en el entorno de programación, utilizando el comando pip. como:

```
pip install --user /home/01010100/TZITZIN/Computacional/Act_4/tephi
```

Se tuvo que definir la carpeta en la que específicamente se encontraba tephi, pues anterior mente al solo utilizar el comando pip install, marcaba error.

Se selecciono un día y hora específica de un sondeo del sitio analizado y descargaron los datos en un archivo de texto. Con esos datos se busco obtener una gráfica similar a las que se producen automáticamente con la opción PDF: Skew-T del sitio de la Universidad de Wyoming. A continuacion se muestra la gráfica producida por el sitio de la Universidad de Wyoming:

Gráfica Wyoming:



Para poder obtener esta grafica pero ahora haciendo uso de **tephi**, se importaron las sig bibliotecas de python:

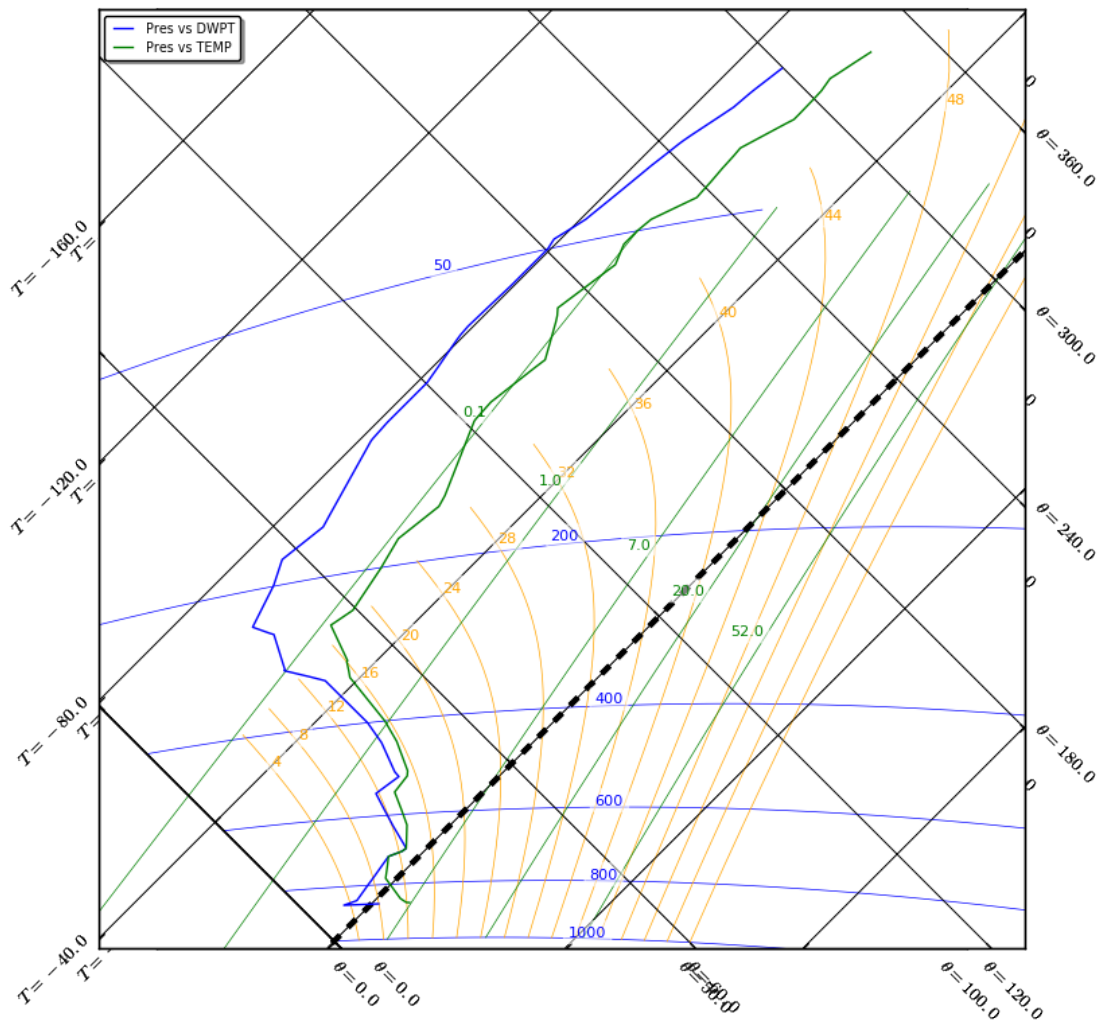
```
import matplotlib.pyplot as plt
import os.path
import tephhi as tph
```

Y para graficar los datos. Se tuvieron que poner los datos en pares, pues tephhi no leia el documento con más de dos columnas. En Emacs se filtraron los datos dejando solamente en un archivo: "Presión" vs "Tempz otro archivo con "Presión" vs "DWPT" que son los datos que se representarían en el Tephigrama. Después de hacer esto, ya podría leerlos tephhi usando:


```
dew_point = pd.read_csv("/home/01010100/TZITZIN/Computacional/Act_4/presDWPT1.csv",
dry_bulb = pd.read_csv("/home/01010100/TZITZIN/Computacional/Act_4/prestemp1.csv",
tpg = tpg.Tephigram()
tpg.plot(dew_point, label="Pres vs DWPT", color="blue")
tpg.plot(dry_bulb, label="Pres vs TEMP", color="green")
plt.show()
```

Y mediante esos comandos se obtuvo la siguiente gráfica:

Gráfica Teph



3. Conclusiones

Usando comandos de prácticas anteriores, pudimos desarrollar las gráficas utilizando Python y su biblioteca de Matplotlib. Es una manera muy sencilla de leer datos de un solo archivo, interpretar los datos y apartir de ellos deducir comportamientos sobre la base de datos utilizada. Usar tephí en un inicio fue una experiencia difícil, me dio muchos problemas primero al intalar tephí desde la terminal y despues de eso, importar tephí en Python. Y también a la hora de lograr todo eso, tephí no leía archivos con más de dos columnas, dio mucho problema el hecho de tener que estar separando tus datos en datos pares para que los leyera el programa, en vez de leer todo el archivo y tu como usuario asigrar las columnas a utilizar. Una ventaja de todo esto fue que el tephigrama al final sale muy bonito, y más completo que el de la universidad, pues al parecer ellos recortaron datos o recortaron la gráfica pues una de las diferencias fue esa, que la grafica producida con tephí muestra poco más de informacion.

4. Referencias

1. <http://weather.uwyo.edu/upperair/sounding.html>
2. <http://computacional1.pbworks.com/w/page/115266988/>
3. <http://stackoverflow.com/questions/4396974/sed-or-awk-delete-n-lines-following-a-pattern>
4. Tefigramas: <http://tephi.readthedocs.io/en/latest/plotting.html>
Actividad %203 %20(2017-1)