



Universidad de Sonora

Departamento de Física

Licenciatura en Física

Computacional I

2017-1

Reporte

Actividad #4: Graficando con Python

Leonardo Coronado Arvayo

Profesor: Carlos Lizárraga Celaya

28 de febrero de 2017

Índice

1. Introducción	2
2. Desarrollo	2
2.1. Limpieza de datos	2
2.2. Graficando con Python	3
2.3. Tephigram	8
3. Conclusiones	11

Resumen

Los datos usados en esta actividad corresponden al 21 de febrero del 2017 a las 12 horas Greenwich para la estación meteorológica de la Paz Baja California. La practica consiste en dos partes principalmente. La primera es graficar usando el modulo "matplotlib", que se uso para visualizar relaciones entre la presión y altura; la altura y temperatura; la altura y temperatura de rocío (DWPT); y las ultimas dos gráficas juntas. Para la segunda parte se uso la librería "tephi" para graficar un "tephigrama" que es una representación gráfica de las observaciones de temperatura, presión y humedad que se hacen a partir de muestreos atmosfericos [6], generalmente es usado para estudios meteorológicos.

1. Introducción

La interfaz virtual de python permite utilizar un gran numero de herramientas en la forma de módulos/librerías. En este caso solo nos enfocamos en el uso de los módulos gráficos matplotlib y teph; estos tienen características distintas. Matplotlib es un modulo muy amplio que se encarga en general de realizar gráficos mientras teph es muy especifico para el tipo de resultado.

Para esta actividad se usaron las variables de altura, presión, temperatura y DWPT para el 21 de febrero del 2017 a las 12 horas Greenwich para la estación meteorológica de la Paz Baja California obtenidos en [3]. Antes de ser utilizados en el ambiente grafico, se limpiaron de irregularidades usando emacs (principalmente el comando query-replace).

Una vez que los datos se encontraron limpios (separados por coma o coma y espacio), se paso a usar matplotlib para graficar la presión contra altura; la altura contra temperatura; la altura contra temperatura de rocío (DWPT); y temperatura y DWPT contra altura.

Para el caso del modulo teph se encontró el problema de que su funcionamiento/código esta dirigido para python 2 por lo que se hizo uso de una computadora personal para su uso.

Como se menciono anteriormente un "tephigrama" es una representación gráfica de las observaciones de temperatura, presión y humedad que se hacen a partir de muestreos atmosfericos [6], generalmente es usado para estudios meteorológicos. Las axis en este son temperatura (T) y entropia (θ) [7].

2. Desarrollo

2.1. Limpieza de datos

Para limpiar los archivos se uso emacs y su comando "query replace" buscando dejar las observaciones según los requerimientos de cada sub-sección.

Para graficar sección 2.2 se dejaron todas las variables y observaciones para el 21 de febrero a las 12 horas Greenwich.

Para el caso de tephigrama se crearon dos archivos uno con presión y temperatura solamente (separados por una coma) y otro con presión y el indicador DWPT (separados igualmente).

2.2. Graficando con Python

Primero se importaron las siguientes librerías, se agregaron y visualizaron las primeras observaciones. Como se ve en:

```
import pandas as pd
import numpy as np
import matplotlib as plt
import matplotlib.pyplot as mplt
%matplotlib inline
```

```
df = pd.read_csv ("/home/lcoronado/computacional1/actividad4/12Z.csv")
df.head(20)
```

En la siguiente figura se ven las observaciones.

	PRES	HGHT	TEMP	DWPT	RELH	MIXR	DRCT	SKNT	THTA	THTE	THTV
0	1016.0	18	17.4	13.0	75	9.34	35	7	289.2	315.7	290.9
1	1008.0	87	18.2	12.2	68	8.93	40	10	290.7	316.2	292.2
2	1000.0	157	17.8	11.8	68	8.76	45	13	290.9	316.0	292.5
3	950.0	595	15.2	8.4	64	7.32	10	25	292.6	313.8	293.9
4	939.0	694	14.6	7.6	63	7.02	14	26	293.0	313.4	294.2
5	925.0	822	15.4	6.4	55	6.56	20	27	295.1	314.4	296.2
6	899.0	1063	15.2	3.2	45	5.39	32	23	297.3	313.4	298.2
7	893.0	1119	17.0	-3.0	25	3.45	35	22	299.7	310.4	300.3
8	878.0	1263	17.4	-4.6	22	3.11	42	19	301.6	311.3	302.1
9	865.0	1390	20.2	-21.8	5	0.78	48	17	305.8	308.4	305.9
10	850.0	1540	20.6	-19.4	5	0.97	55	14	307.7	311.1	307.9
11	825.0	1795	18.5	-17.1	8	1.22	70	10	308.1	312.3	308.4

Figura 1: Archivo

Luego se describieron las variables con:

```
df.describe()
```

Resultando en:

	PRES	HGHT	TEMP	DWPT	RELH	MIXR	DRCT	SKNT	THTA
count	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000
mean	350.151948	11544.701299	-36.762338	-59.593506	11.402597	0.973117	139.636364	13.519481	374.653247
std	321.000310	7551.621316	36.676552	35.502907	17.123591	2.248525	96.342988	7.833145	77.069749
min	30.000000	18.000000	-81.100000	-95.900000	1.000000	0.000000	5.000000	1.000000	289.200000
25%	77.300000	3915.000000	-71.900000	-92.000000	4.000000	0.000000	50.000000	8.000000	314.200000
50%	232.000000	11398.000000	-44.900000	-70.400000	5.000000	0.010000	125.000000	12.000000	346.500000
75%	640.000000	18051.000000	3.400000	-22.400000	8.000000	0.780000	225.000000	19.000000	399.500000
max	1016.000000	23630.000000	20.600000	13.000000	75.000000	9.340000	355.000000	38.000000	579.100000

Figura 2: Estadísticos de los indicadores

Se busco si tenían datos no definidos (NA) con:

```
df.apply(lambda x: sum(x.isnull()),axis=0)
```

Y no se encontró ninguno como se puede ver en las siguiente figura:

```
PRES      0
HGHT      0
TEMP      0
DWPT      0
RELH      0
MIXR      0
DRCT      0
SKNT      0
THTA      0
THTE      0
THTV      0
dtype: int64
```

Figura 3: Datos no definidos

Luego se verifico el nombre de las variables con:

```
df.columns
```

Que resultado en:

```
Index(['PRES', 'HGHT', 'TEMP', 'DWPT', 'RELH', 'MIXR', 'DRCT', 'SKNT',
      'THTA', 'THTE', 'THTV'],
      dtype='object')
```

Figura 4: Títulos de las variables

Para graficar presión contra altura se uso el siguiente código:

```
#Presión (hPa) vs. Altura (m)
y = df[' HGHT']
x = df[' PRES']
mplt.plot(x, y)
mplt.ylabel('Altura (m)')
mplt.xlabel('Presión (hPa)')
mplt.title('Presión vs. Altura')
```

Que resultado en:

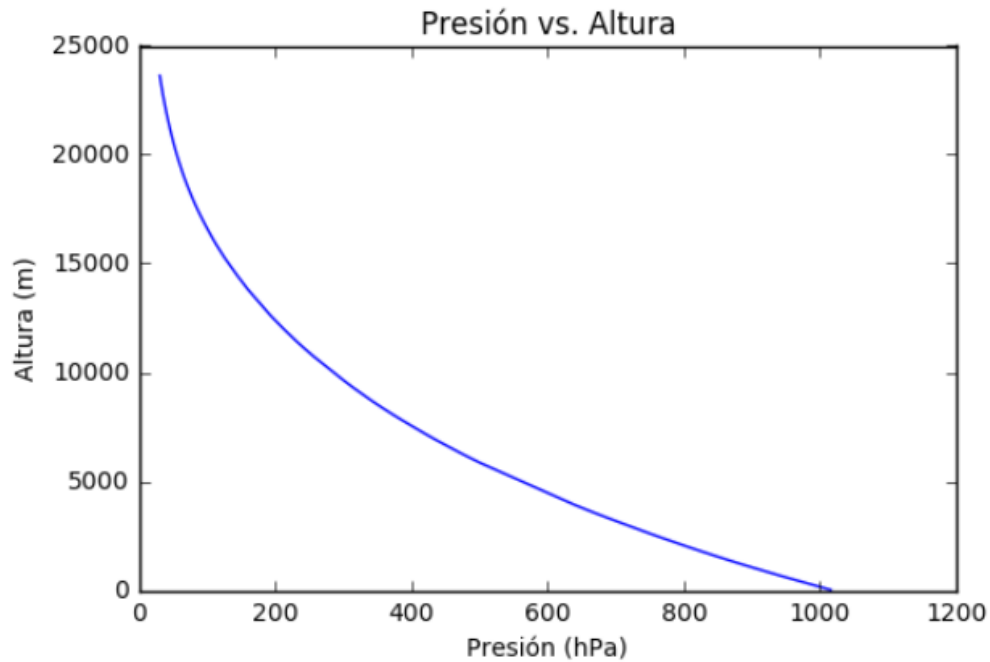


Figura 5: Presión (hPa) vs. Altura (m)

Para el de temperatura contra altura el código es:

```
#Temperatura (C) vs. Altura (m)
y = df[' HGHT']
x = df[' TEMP']
mplt.plot(x, y)
mplt.ylabel('Altura (m)')
mplt.xlabel('Temperatura (C)')
mplt.title('Temperatura vs. Altura')
```

Cuya gráfica es:

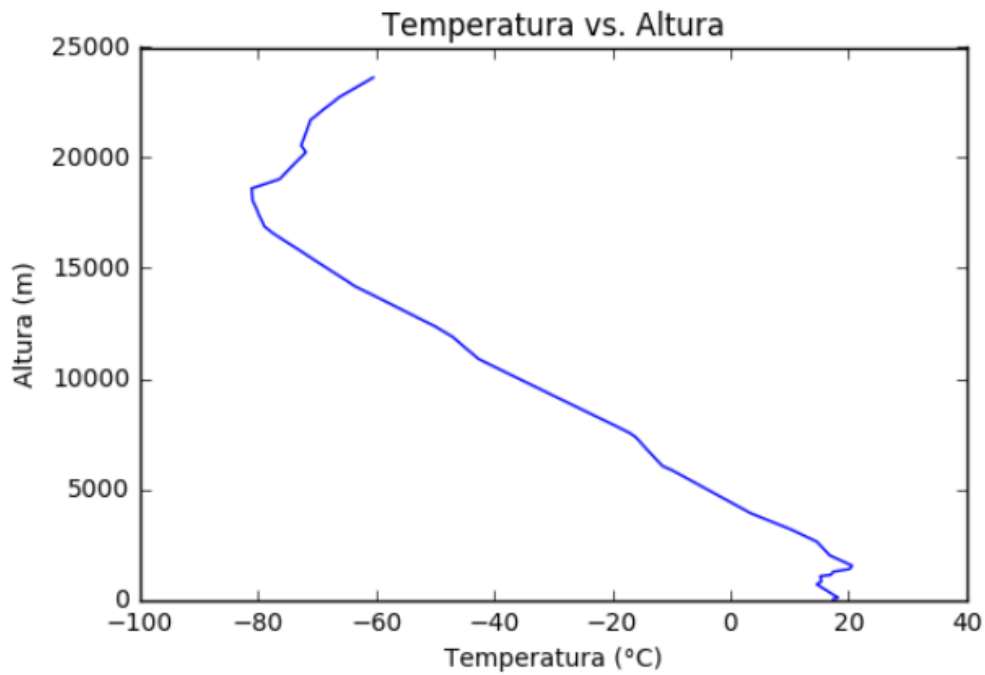


Figura 6: Temperatura (C) vs. Altura (m)

Para el caso de la temperatura de rocío contra altura se uso:

```
#Temperatura de rocío (C) vs. Altura (m)
y = df[' HGHT']
x = df[' DWPT']
matplotlib.pyplot.plot(x, y)
matplotlib.pyplot.ylabel('Altura (m)')
matplotlib.pyplot.xlabel('Temperatura de rocío (C)')
matplotlib.pyplot.title('Temperatura de rocío vs. Altura')
```

Que resultado en:

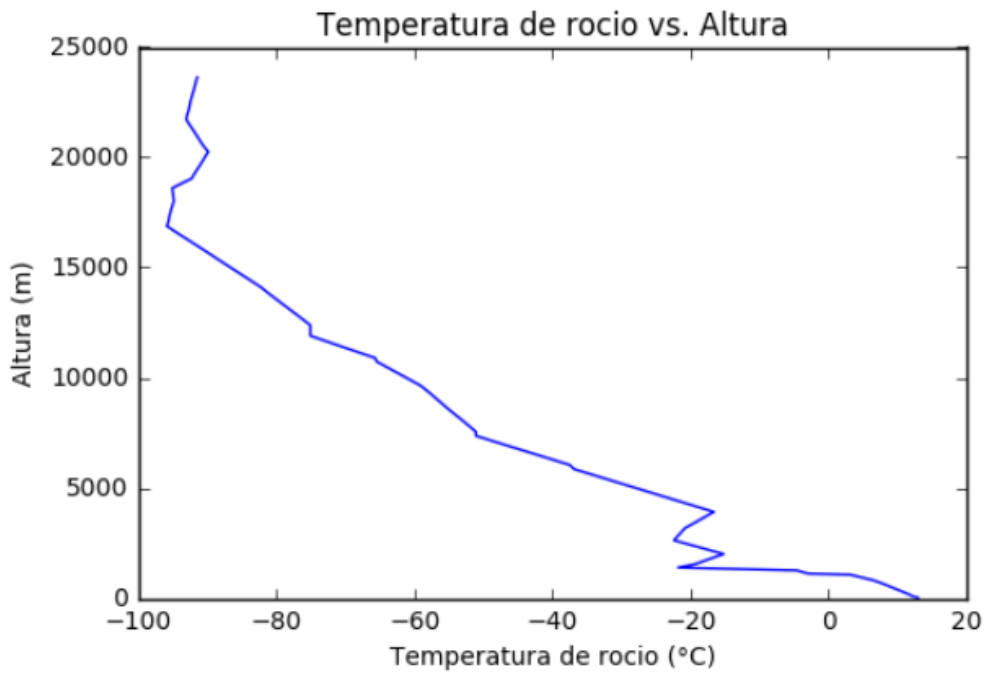


Figura 7: Temperatura de rocío (C) vs. Altura (m)

Por ultimo para graficar altura contra Temperatura/Temperatura de rocío se recurrió a las siguientes ordenes:

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

matplotlib.title('Altura (m) vs. Temperatura/Temperatura de rocío (C)')
matplotlib.ylabel('Altura (m)')
matplotlib.xlabel('Temperatura/Temperatura de rocío (C)')

matplotlib.grid(True)
matplotlib.plot(x1,y,c='purple',label='Temperatura de rocío')
matplotlib.plot(x2,y,c='red', label='Temperatura')
matplotlib.legend(fancybox=True, shadow=True)
```

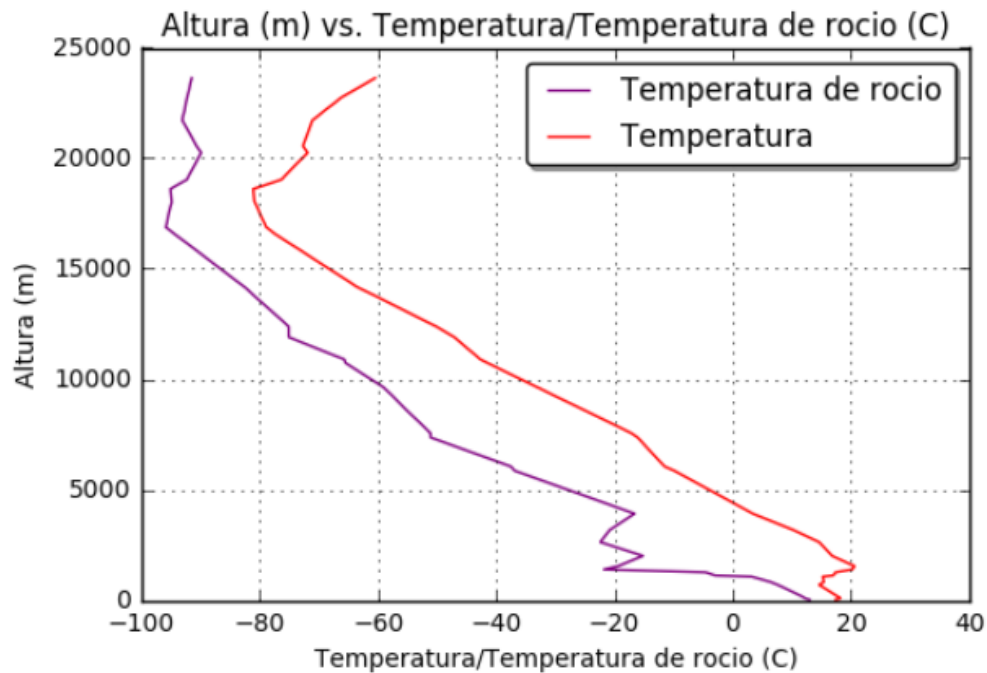



Figura 8: Altura (m) vs. Temperatura/Temperatura de rocío (C)

2.3. Tephigram

No me fue posible correr el modulo tephi en python 3, por lo que en mi computadora personal realice lo siguiente, primero instale múltiples python (buscando en especifico el 2) con este código[2]:

```
sudo dnf install python33 # to get CPython 3.3
sudo dnf install python34 # to get CPython 3.4
sudo dnf install python26 # to get CPython 2.6
sudo dnf install pypy pypy3 jython python35 # to get more at once
```

Luego active la interface virtual para estos con [1]:

```
conda create -n py27 python=2.7
source activate py27
conda install notebook ipykernel
ipython kernel install --user
```

Antes de esto tuve que instalar anaconda y activar el comando conda, para esto los refiero a [4] y [5]. Instale los modulos ".s", "matplotlib", "pandas" "tephicon":

```
python2 -m pip install scipy
python2 -m pip install pandas
python2 -m pip install tephi
python2 -m pip install os
python2 -m pip install matplotlib
```

Una vez instalado tephi y los demás módulos, se uso el siguiente código para graficar (desde la interface virtual de python 2 dentro de la carpeta del modulo tephi):

```
#modulos
import matplotlib.pyplot as plt
import os.path
import pandas as pd
import tephi as tph

#archivos
dew_point = pd.read_csv("/home/noxd/computacional/a4/dews.csv", names=["PRES", "DWPT"])
dry_bulb = pd.read_csv("/home/noxd/computacional/a4/temps.csv", names=["PRES", "TEMP"])

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

Donde se puede observar que el cambio al código presentado en tephi fue el uso de panda para darle a tephi las variables que este gráfica.

El resultado fue:

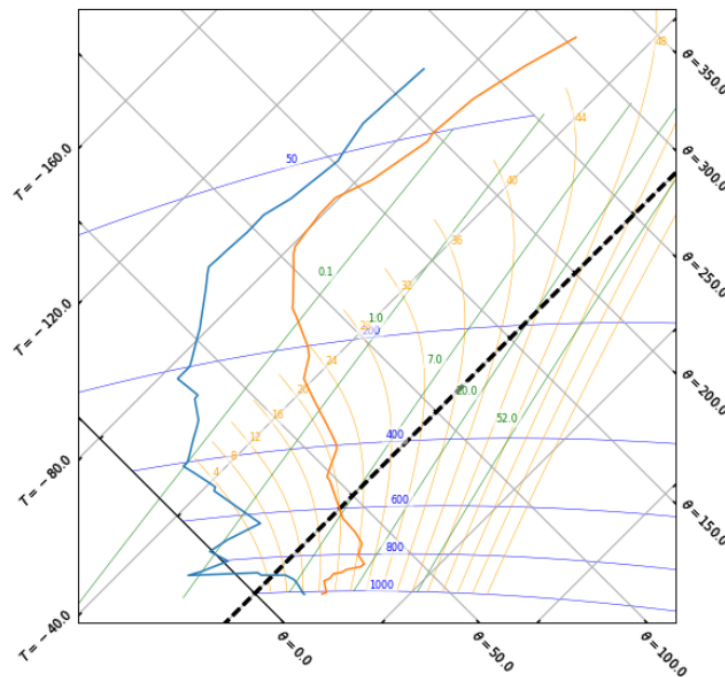


Figura 9: grafica con tephi

Y se puede decorar agregando lo siguiente al codigo:

```
#variables
```

```

dews = pd.read_csv("/home/noxd/computacional/a4/dews.csv", names=["PRES", "DWPT"])
temp = pd.read_csv("/home/noxd/computacional/a4/temps.csv", names=["PRES", "TEMP"])

#graficando con detalles
tpg = tp.Tephigram()
#temperaturas
tpg.plot(dews, label='Dew-point temperature', color='blue', linewidth=2, linestyle='--', ma
tpg.plot(temp, label='Temperature', color='red', linewidth=2, linestyle='--', marker='s')
#iso
tephi.ISOBAR_LINE.update({'color': 'purple', 'linewidth': 3, 'linestyle': '--'})
tephi.WET_ADIABAT_LINE.update({'color': 'yellow', 'linewidth': 3, 'linestyle': '--'})

plot.show()

```

Que resulta en:

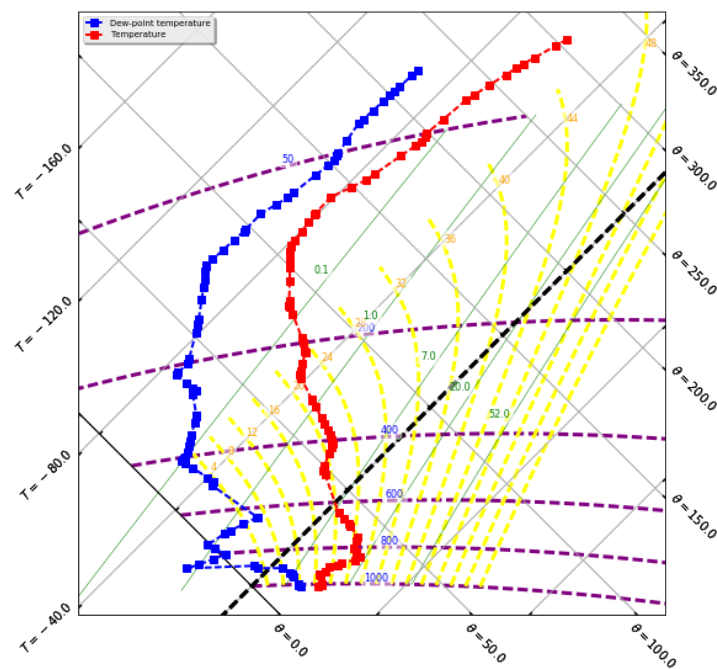


Figura 10: grafica con tephigram con detalles

Por ultimo, comparando con la gráfica de [3] se ve:

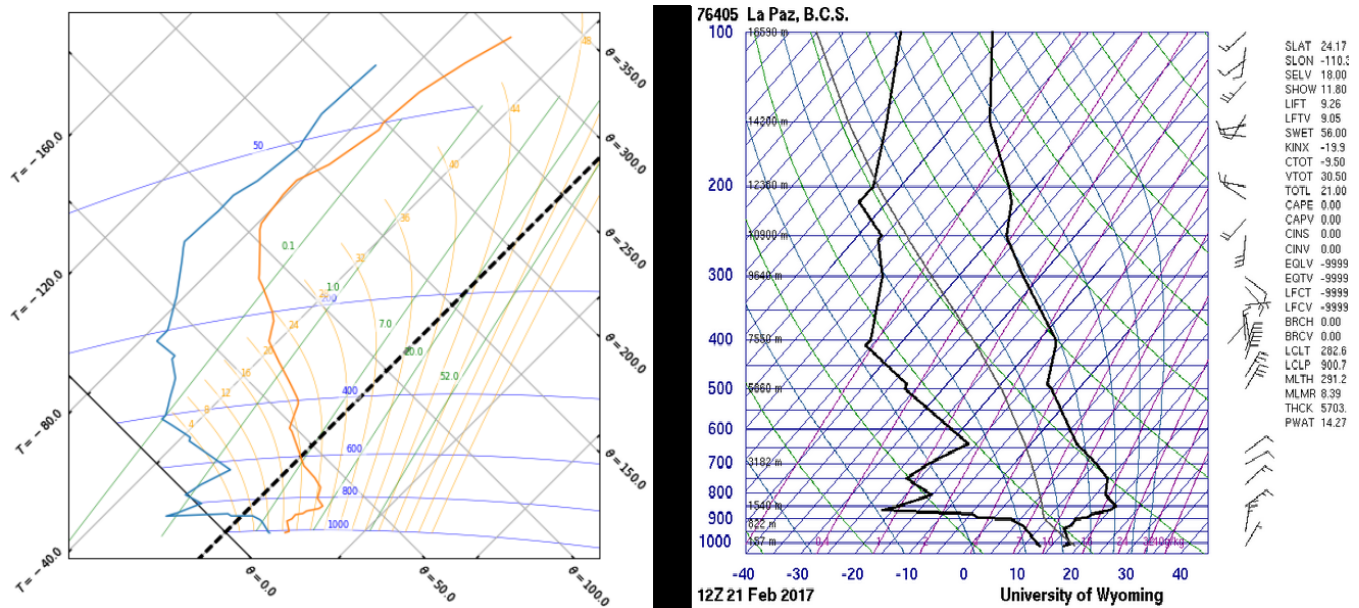


Figura 11: Comparación entre tephigramas

3. Conclusiones

Batalle tanto graficando en tephigramas que no tuve tiempo de estudiar de forma correcta la lectura de estos gráficos. Sin embargo me gusto la experiencia de tener que hacer uso de python en mi computadora personal ya que imagino que es relativamente común encontrarse con módulos desactualizados.

Por otra parte se me hizo que puede ser muy útil usar la librería matplotlib para graficar.

Referencias

- [1] <http://stackoverflow.com/questions/30492623/using-both-python-2-x-and-python-3-x-in-ipynb>
- [2] <https://developer.fedoraproject.org/tech/languages/python/multiple-pythons.html>
- [3] Department of Atmospheric Science, University of Wyoming <http://weather.uwyo.edu/upperair/sounding.html>
- [4] <http://stackoverflow.com/questions/20081338/how-to-activate-an-anaconda-environment>
- [5] <https://docs.continuum.io/anaconda/install>
- [6] modulo tephigram <http://tephi.readthedocs.io/en/latest/introduction.html>
- [7] Tephigram. Wikipedia: <https://en.wikipedia.org/wiki/Tephigram>