# Reporte de Actividad 5
# Preparando datos con ayuda de Emacs

Michelle Contreras Cossio

6 Marzo del 2018

## 1 Introducción

En esta actividad, de nuevo se trabajó con la base de datos atmosféricos de la Universidad de Wyoming, utilizando la misma estación en el Aeropuerto Internacional de Adelaida, Australia. Sin embargo, se dió un enfoque bastante diferente.

El objetivo de esta práctica fue comprender mejor el uso de Emacs para editar archivos, automatizando así un proceso que, al trabajar con archivos muy grandes, nos podría tomar mucho tiempo. Estos datos, utilizados de nuevo de un año completo, se editaron en Emacs y también con un poco de ayuda del bash, para trabajar únicamente información de las variables CAPE y PW, explicadas en la siguiente sección, y posteriormente se trabajó con estos archivos jupyter.

## 2 Descripción de los conceptos físicos: CAPE y PW

Durante esta práctica se utilizan datos de las variables "CAPE" y "PW" de la base de datos atmosféricos de la Universidad de Wyoming, durante esta sección se van a definir para poder entender los datos y gráficas que se analizarán.

CAPE, por sus siglas en inglés, significa Convective Available Potential Energy, Energía Potencial Convectiva Disponible. Es la cantidad de energía que una parcela de aire tendría si se levantara cierta distancia, verticalmente. Es un indicador de inestabilidad atmosférica y es muy importante al momento de predecir catástrofes atmosféricas, como son tormentas, tornados, granizo, remolinos, etc.

PW, por sus siglas en inglés, Precipitable Water, es decir Agua Precipitable, es la cantidad de agua, expresada como altura o masa, que se obtendría si todo el vapor de agua contenido en una columna específica de la atmósfera, se condensara y precipitara. Se mide en milímetros o pulgadas.

## 3 Descripción del proceso de limpieza y preparación de los datos

En esta sección se explica el proceso que se llevó a cabo para crear los archivos ingresados a pandas.

1. Primeramente, se inició tomando el archivo de sondeos.txt creado en la actividad anterior y copiándolo a nuestra nueva carpeta de Actividad5.
2. Posteriormente, se realizó un script, similar al utilizando en la Actividad 4 para crear el df2017. Pero este nos permitía quedarnos únicamente con los datos de la fecha, CAPE y Precipitable Water, para 00Z y 12Z.

```
#!/bin/bash

egrep -v 'Station' sondeos.txt | egrep '94672|CAPE|Precip' > df2017CAPE_PW.csv
```

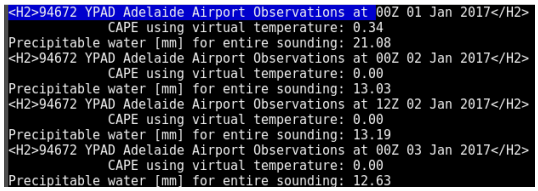
3. Una vez corrido el script, se creó un solo archivo llamado df2017CAPE_PW.csv.


4. Este archivo se corrió con emacs, con lo que se pudo observar que contenía información innecesaria, ya que el objetivo era tener un archivo como el siguiente:

```
$ head df2017CAPE_PW_00Z.csv
01 01 2017,    0.00,   9.42
02 01 2017,    17.60,   9.39
03 01 2017,    0.00,   6.29
04 01 2017,    0.00,   12.83
05 01 2017,    0.00,   9.22
06 01 2017,    0.00,   8.87
07 01 2017,    0.00,   4.91
08 01 2017,    0.00,   6.70
09 01 2017,    0.00,   11.46
10 01 2017,    0.00,   8.03
```

   Donde la primera columna representara la fecha, la segunda la variable CAPE y la tercera PW.

5. Para ello, se hizo uso de comandos en emacs. Primero se utilizó el comanto ctrl+ tecla de espacio, que nos permite seleccionar cierta parte del código y, moviendo con las flechas, se seleccionó la parte de código que se deseaba eliminar.

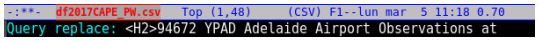


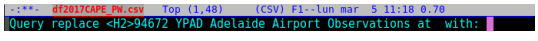

6. Posteriormente se utilizó el comando ctrl+w, que borra esa parte seleccionada pero la manda a la memoria. Y se recuperó, en el mismo lugar donde estaba, con ctrl+y, que vacía en el lugar indicado la memoria.

7. Se utilizó el comando esc+<, que nos llevó al inicio del documento.

8. El comando esc+% nos permitió abrir el Query replace, comando que nos permite remplazar alguna cadena de código por otra.


9. En el Query replace vaciamos la memoria con el ctrl+y, que es la cadena que queremos reemplazar en el código y damos enter. Después, nos pregunta por cual otra cadena la queremos reemplazar, en el primer caso, sólo se buscaba eliminar esa parte del código, por lo que se reemplazo con nada. Posterior a eso nos marca las partes del código que se van a reemplazar y damos aceptar con el símbolo !.



10. Así, se eliminó una cadena de código que se repetía una vez para cada uno de los días.

```
00Z 01 Jan 2017</H2>
            CAPE using virtual temperature: 0.34
Precipitable water [mm] for entire sounding: 21.08
00Z 02 Jan 2017</H2>
            CAPE using virtual temperature: 0.00
Precipitable water [mm] for entire sounding: 13.03
12Z 02 Jan 2017</H2>
            CAPE using virtual temperature: 0.00
Precipitable water [mm] for entire sounding: 13.19
00Z 03 Jan 2017</H2>
            CAPE using virtual temperature: 0.00
Precipitable water [mm] for entire sounding: 12.63
12Z 03 Jan 2017</H2>
            CAPE using virtual temperature: 0.00
Precipitable water [mm] for entire sounding: 15.61
```

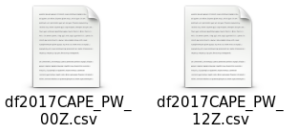
11. El proceso realizado de del paso 5 al 10 se repitió para cada una de las cadenas que se quería eliminar o reemplazar, ya fuera por una coma o, en el caso de los meses, se reemplazaron por números.

12. Finalmente, se obtuvo un archivo con renglones de la siguiente manera:

```
00Z 01 01 2017, 0.34, 21.08
00Z 02 01 2017, 0.00, 13.03
12Z 02 01 2017, 0.00, 13.19
00Z 03 01 2017, 0.00, 12.63
12Z 03 01 2017, 0.00, 15.61
00Z 04 01 2017, 0.00, 17.10
12Z 04 01 2017, 0.00, 19.36
00Z 05 01 2017, 0.08, 29.36
12Z 05 01 2017, 240.18, 34.8
00Z 06 01 2017, 52.44, 33.44
12Z 06 01 2017, 0.00, 24.70
00Z 07 01 2017, 0.00, 26.00
12Z 07 01 2017, 6.89, 45.50
00Z 08 01 2017, 0.00, 37.59
12Z 08 01 2017, 0.00, 40.69
00Z 09 01 2017, 0.00, 45.87
```

13. Como el objetivo era tener dos archivos, uno con los datos del 00Z y otro del 12Z, se utilizaron los siguientes comandos, con grep, lo cual separó el archivo df2017CAPE_PW.csv, en dos archivos: df2017CAPE_PW_00Z.csv y df2017CAPE_PW_12Z.csv.

```
michellecc@ltsp51:~/Computacional1/Actividad5$ grep '00Z' df2017CAPE_PW.csv  > df2017CAPE_PW_00Z.csv
michellecc@ltsp51:~/Computacional1/Actividad5$ grep '12Z' df2017CAPE_PW.csv  > df2017CAPE_PW_12Z.csv
```




14. Los archivos creados tenían al inicio de cada renglón 00Z o 12Z, respectivamente, por lo que utilizando los mismos comando del paso 5 al 10, se eliminaron, para finalmente lograr dos archivos con la misma estructura que el mostrado en el paso 4, para después poder trabajar con los datos en pandas.

# 4 Análisis de datos utilizando Pandas

En esta sección se muestra como se trabajaron los datos, anteriormente limpiados, para poder realizar gráficas para una mejor visualización y análisis de estos.

1. Primeramente, y como de costumbre, se cargaron las bibliotecas con las que se trabajó, los procesos posteriores se realizaron para cada uno de los dos archivos creados en la sección anterior.

```
import pandas as pd
import numpy as np
from datetime import datetime
```

2. Posteriormente, se leyeron los datos, se asignó nombre a cada columna y se le dió formato de número a la columna CAPE.

```
# Leer archivo de datos de 00Z
# Convertir la columna CAPE de objeto a número
df00 = pd.read_csv("df2017CAPE_PW_00Z.csv", header=None, names=['Date', 'CAPE', 'PW'])
df00.CAPE=pd.to_numeric(df00.CAPE, errors='coerce')
df00.head()
```

3. Se creó una nueva columna, con formato de fecha.

```
# Convertir la cadena de caracteres 'Date' en variable temporal 'NDate' (00Z)
df00['Ndate'] = pd.to_datetime(df00['Date'], format='%d %m %Y')
df00['month'] = df00['Ndate'].dt.month
df00.head()
```

4. Haciendo uso de la biblioteca seaborn y matplotlib realizaron las gráficas de boxplot, por mes, la primera del CAPE y la segunda del PW.

```
# graficar Boxplots por mes de CAPE (00Z)
# Biblioteca Seaborn
import seaborn as sns
import matplotlib.pyplot as plt
ax = sns.boxplot(x="month", y="CAPE", data=df00)
plt.show()
```

```
# graficar Boxplots por mes de PW (00Z)
# Biblioteca Seaborn
import seaborn as sns
import matplotlib.pyplot as plt
ax = sns.boxplot(x="month", y="PW", data=df00)
plt.show()
```

5. Utilizando la misma biblioteca, seaborn, se creó el jointplot, que compara las variables CAPE y PW.

```
#Gráfica jointplot (00Z)
import seaborn as sns
sns.set(style="darkgrid", color_codes=True)

g = sns.jointplot("CAPE", "PW", data=df00, kind="reg",
                  color="r", size=7)
plt.show(g)
```

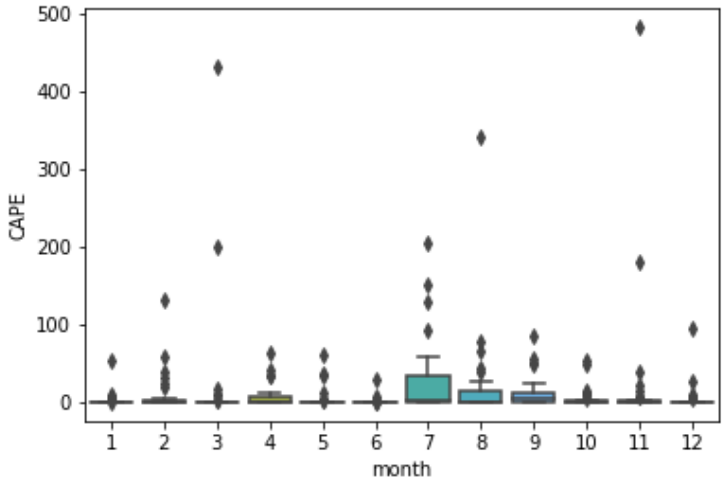
6. Finalmente, se creó la gráfica de lmplot de PW vs CAPE, donde crea un modelo de regresión lineal para cada uno de los meses.

```
#Gráfica lmplot (00Z)
g = sns.lmplot(x="CAPE", y="PW", hue="month",
               truncate=True, size=5, data=df00)
plt.show(g)
```

# 5 Resultados del análisis

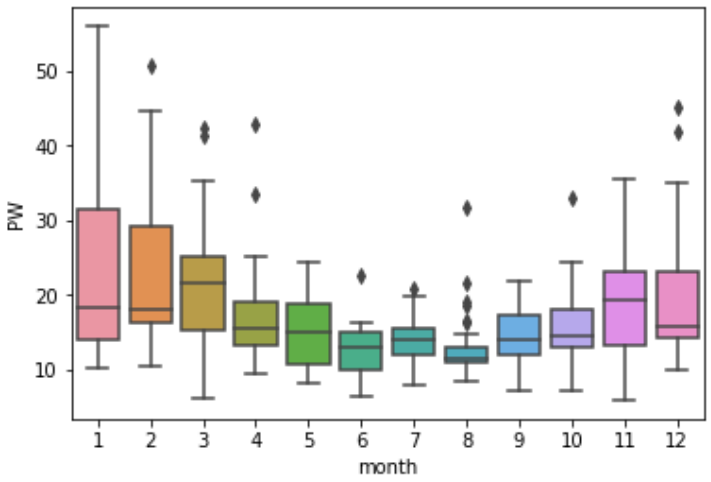
Los pasos mostrados en la sección anterior crearon un total de 8 gráficas, analizadas a continuación:

- **Archivo de datos df2017CAPE_PW_00Z.csv:** Debido a la zona horaria en la que se encuentra, los sondeos en 00Z se realizaron a las 12pm.
  - **Boxplot:**
    * CAPE:



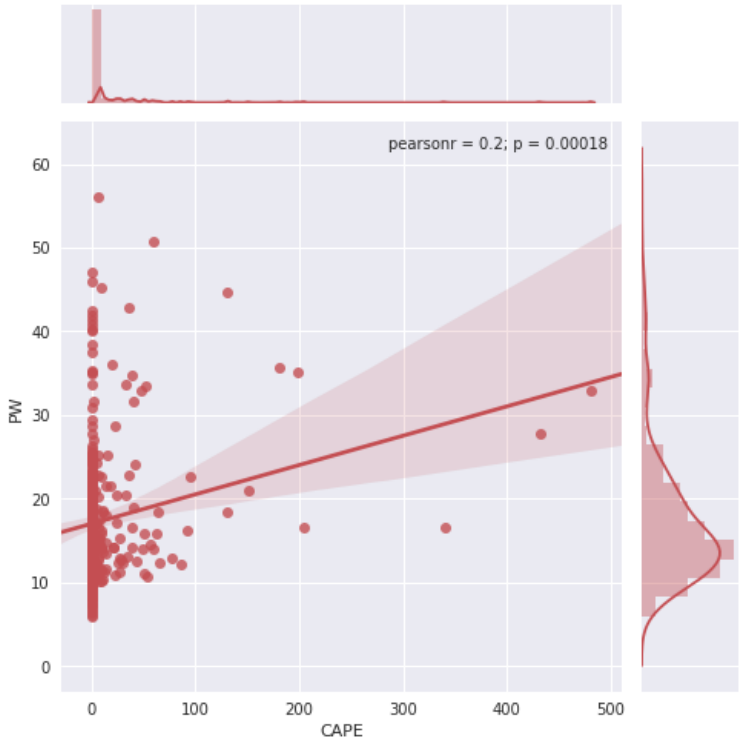

Este diagrama de caja muestra que en todo el año, a las 12 pm, la variable CAPE esta alrededor del cero, aunque existen valores muy fuera del rango, son pocos.

    * PW:



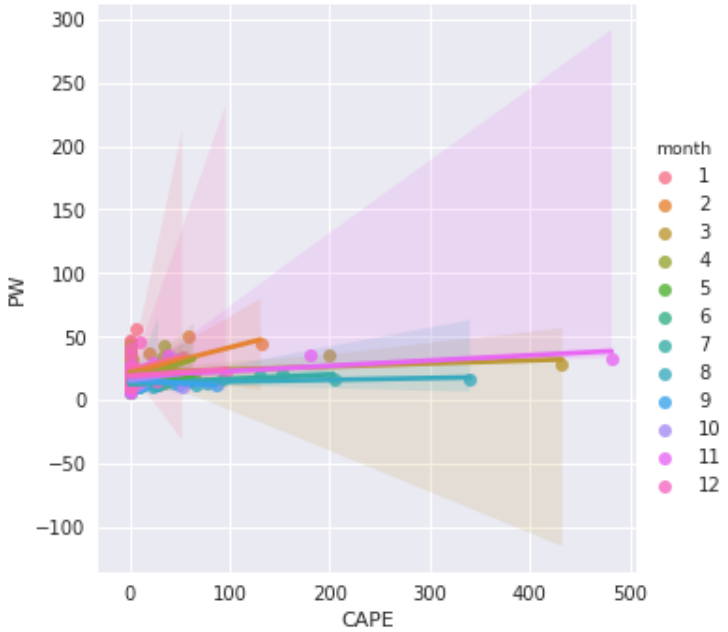

Por su parte, este diagrama de caja del agua precipitable, muestra que la media en general esta alrededor de los 15 y 25 mm. Sin embargo, se nota un poco la variación, donde de diciembre a febrero, es decir, el verano en Australia, este índice es más alto que de junio a agosto, donde es invierno.

– **Jointplot**:



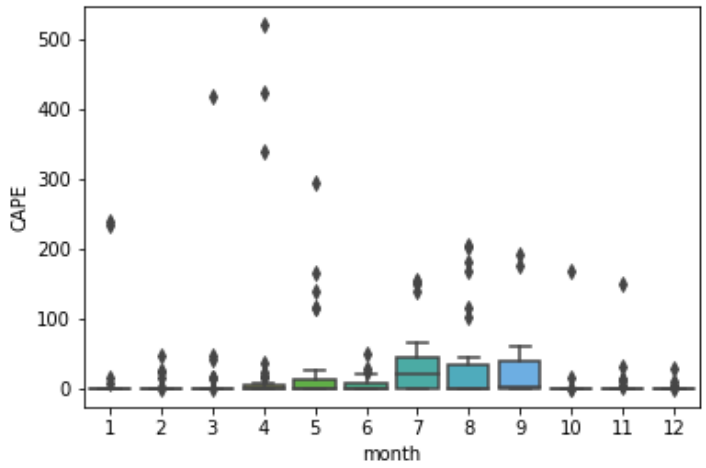

Esta gráfica de PW vs CAPE nos muestra el coeficiente de correlación de Pearson, donde podemos observar que las variables si estan ligadas linealmente y positivamente, ya que este es igual a 0.2. La gráfica además muestra la recta que mejor se aproxima, así como la distribución de cada una de las variables. Podemos observar, que la variable CAPE se distribuye en su mayoría alrededor del 0 y la variable PW alrededor del 20.

– **Lmplot:**



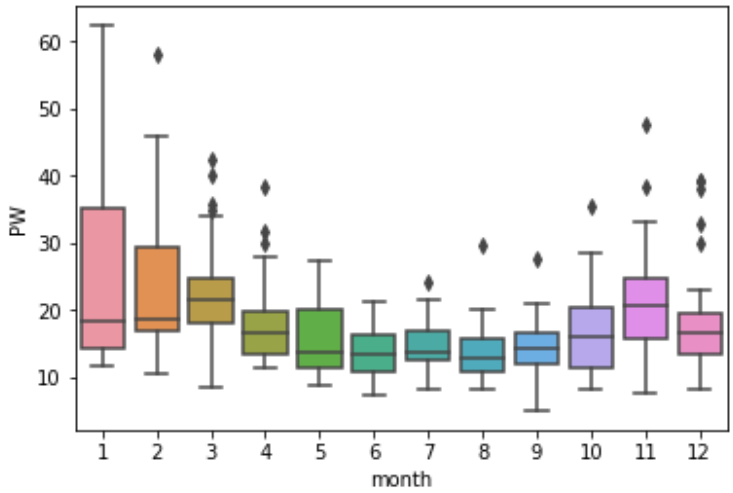
El Lmplot grafíca PW contra CAPE, pero de cada mes, para así ver como varían ambas, según cambian las estaciones del año. Para cada mes creó una recta de regresión lineal, se puede observar que los meses donde el CAPE es más alto, el PW es más bajo.

- **Archivo de datos df2017CAPE_PW_12Z.csv:** Debido a la zona horaria en la que se encuentra, los sondeos e 12 e realizaron a las 12am.
  - **Boxplot:**
    * CAPE:



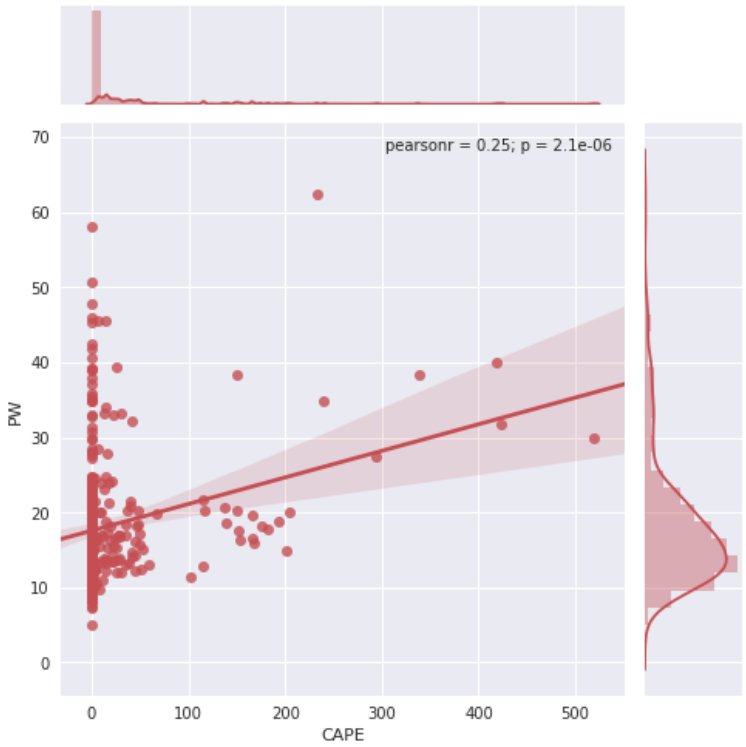

Podemos observar que a las 12 am, existe una pequeña variación que con la gráfica de las 12 pm. El CAPE ya no se encuentra tan cercano al cero, en los meses de invierno, este aumenta.

    * PW:



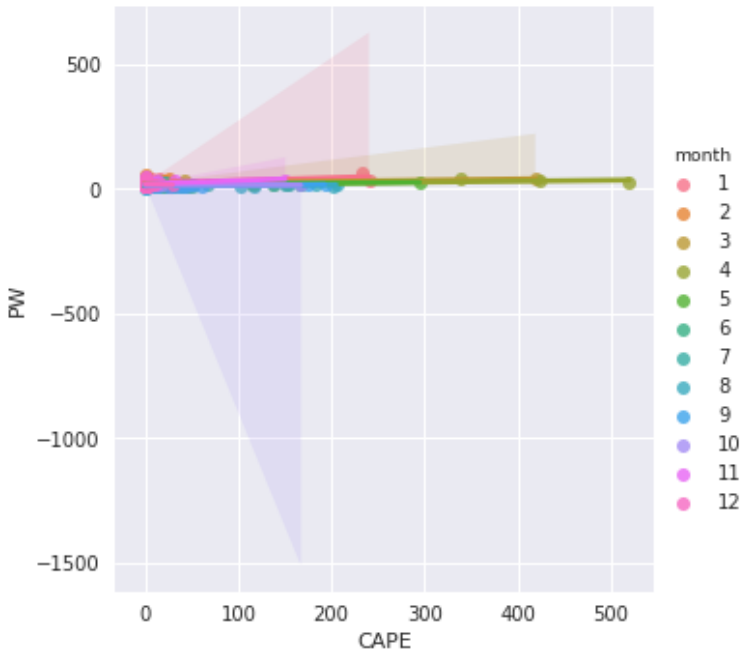

Este boxplot si es bastante similar al de las 12 pm, la variación existe en el tamaño del rango intercuartílico, que en este caso se ve un poco más extenso.

– **Jointplot:**




Esta gráfica muestra una mayor correlación lineal, con un tamaño de 0.25 y las distribuciones se encuentran similares a las de las 12 pm, alrededor de los mismo valores.

– **Lmplot:**




Por último, esta gráfica difiere un poco con la de las 12 pm, ya que muestra que el PW no varió mucho entre los meses.

# 6 Conclusiones

Como conclusión, me gustaría agregar que el objetivo fue cumplido, ya que se pudo realizar una limpia de datos bastante extensos, como se mencionó, eran de un año; tal y como se requerían, de una manera muy rápida, se logró en menos de 40 minutos. Si estos datos se hubieran ingresado en pandas antes de ser editados y limpiados, el proceso probablemente hubiera sido más largo y tedioso, con muchos tropiezos e incluso sin haber podido llegar a lo que se requería.

La moraleja que me queda tras esto es que subestime mucho el uso que se le podía dar a Emacs, sobre todo porque nunca había trabajado en un procesamiento de datos con una cantidad grande de estos, lo comparaba mucho con el Bloc de Notas en Windows, pero veo que permite realizar procesos de una manera más automática y nos permite "ser flojos".

# 7 Bibliografía


- Convective available potential energy (2018). Consultado: 2 de Marzo del 2018, de Wikipedia. Sitio web: https://en.wikipedia.org/wiki/Convective_available_potential_energy
- Precipitable Water (2018). Consultado: 2 de Marzo del 2018, de Wikipedia. Sitio web: https://en.wikipedia.org/wiki/Precipitable_water
- Agua precipitable. Consultado: 2 de Marzo del 2018, de Agua Market. Sitio web: http://www.aguamarket.com/diccionario/terminos.asp?Id=4550


# 8 Apéndice

1. ¿Cómo se te hizo esta actividad? ¿Compleja, Difícil, Sencilla?

   Se me hizo bastante sencilla.

2. ¿Qué te llamó más la atención?

   El uso de emacs, nunca me había dado la oportunidad de usarlo o aprender sus comandos y es bastante útil.

3. ¿Qué parte fue la que menos te interesó hacer?

   No hubo nada que me desinteresara, hubo cosas neutrales, pero no con desinterés o que no me gustaran.

4. ¿Cómo mejorarías esta actividad? ¿Qué le faltó? ¿Qué sobró?

   Me hubiera gustado hacer más uso de pandas, que nos pidiera algo que no viniera en el ejemplo.

5. ¿Hasta este punto, que te parece el uso de Jupyter para programar en Python?

   Me ha gustado mucho y me parece sencillo, hasta el momento.