

Biblioteca Pandas.

Sealtiel Gallardo Martínez

11 de Febrero de 2019

A lo largo de este curso de Física Computacional, nos hemos dado cuenta que hay siempre otras alternativas para realizar los trabajos científicos de una manera sencilla y eficiente tanto para el escritor como para el lector. Python siempre será una buena idea, ya que es de extrema utilidad para realizar nuestro trabajo de manera satisfactoria y con cada actividad que realizamos nos damos cuenta que Fortran tal vez se haya quedado un poco atrás.

En la tercera actividad nos pide realizar un análisis climatológico de una región elegida, preferidamente de Sonora; a causa de esto elegí realizar el análisis en base a los datos recopilados en la región de Hermosillo. Esto con motivo de darnos cuenta en otros sentidos diferentes, la variación de estas temperaturas, ya que este documento está dirigido a la comunidad de física de la Universidad de Sonora, situada en el municipio de Hermosillo.

Para realizar el análisis de una manera eficiente utilizaremos una librería que nos ayudará permitiendo utilizar sus funciones para obtener más información acerca de una tabla de datos externa. Esta última será brindada de la página web oficial del Servicio Meteorológico Nacional y contiene una cantidad muy útil de datos sobre la situación meteorológica de la capital de Sonora.

Para leer estos datos utilizaremos la siguiente sección de código:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
import calendar
import datetime
```

```

#Este programa est destinado a leer los datos de la CONAGUA y
    determinar
#la informacin de cada mes en diferentes parmetros.

os.chdir('/Users/Usuario/Python')

#Este bloque determina los elementos nulos de la tabla de datos.
missing=['NULO']
df = pd.read_csv('fuentes.txt', sep='\s+', na_values=missing)
pd.isnull(df)

```

Donde nos podemos dar cuenta que, en ciertas secciones de código, vemos funciones dirigidas a eliminar valores considerados como nulos en la tabla de datos leídos, así como el formato que le damos a la columna de fechas que podemos encontrar en la tabla de datos obtenidas en la red del SMN a través del siguiente comando:

```
df['FECHA'] = pd.to_datetime(df['FECHA'])
```

Después de realizar estos podemos encontrar una función de la librería Pandas, que nos ayudará a agrupar los datos de la manera que a nosotros nos convenga, de la cual recopilamos los siguientes calculos:

- A partir de cálculos estadísticos determinaremos cuales son los meses más lluviosos, así como los más húmedos por su estrecha relación. Usaremos el siguiente bloque de códigos para poder agrupar los datos mensuales y determinar el promedio de precipitaciones en cada periodo mencionado:

```
df.groupby(df['FECHA'].dt.strftime('%B'))['PRECIP'].mean().sort_values()
```

- Ahora de la misma manera podemos determinar el promedio mensual de las máximas temperaturas así como las mínimas temperaturas de cada mes, brindándonos la tabla mostrada en la sección de codigo subida a mi repertorio de GITHUB:

```
df.groupby(df['FECHA'].dt.strftime('%B'))['TMAX'].mean().sort_values()
```

```
df.groupby(df['FECHA'].dt.strftime('%B'))['TMIN'].mean().sort_values()
```

A continuación, veremos como podemos cambiar el periodo con el que podemos analizar los datos. Anteriormente utilizamos meses como separación, ahora lo intentaremos con años y temperaturas mínimas para poder identificar el año más frío contenido en la tabla de datos.

- Sección de código que nos permite visualizar numéricamente la columna de la temperatura mínima en contra del tiempo así como las temperaturas mínimas contra este mismo tiempo.

```
df.groupby(df['FECHA'].dt.strftime('%Y'))['TMIN'].mean().sort_values()
```

```
df.groupby(df['FECHA'].dt.strftime('%Y'))['TMAX'].mean().sort_values()
```

Gracias a otras funciones como `df.describe` y `df.dtypes`, podemos obtener los datos estadísticos más importante de cada columna, y aquí los implementamos de manera que determinemos los principales parámetros:

```
df.describe()
```

```
df.dtypes
```

De esta manera, podemos realizar otra visualización de los datos.

Otra herramienta que volveremos a utilizar en esta actividad es la graficadora de Matplotlib, esto con el fin de mostrar un apoyo gráfico de los datos, entonces mostraré la relación que pudiera existir entre los índices de evaporación y las temperaturas máximas de las regiones. Como podemos ver,

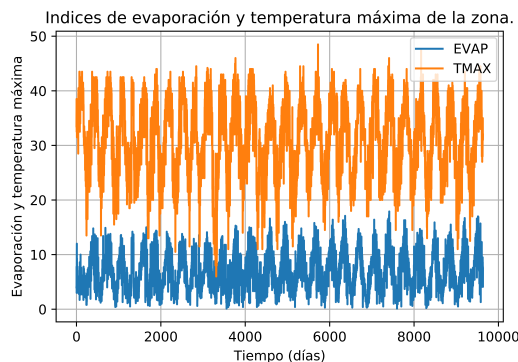


Figura 1: Posible relación entre evaporación y temperatura.

esta librería de Python llamada Pandas siempre nos será de gran utilidad a la hora de realizar un análisis completo a todo un conjunto de datos. Desde el ambiente de programación en Jupyter hasta las librerías que siempre estarán disponibles en Internet, podemos obtener un trabajo sin limitaciones técnicas.