

Páctica 8

Uniendo dataframes en Pandas

Jesús Roberto Araujo Sánchez

3 de Abril, 2019

Abstract

Se tomo un conjunto de datos y se manipularon para poder manejarlos como una serie, despues se graficaron e interpretaron.

1 Introduction

Cuando trabajamos datos lo mas facil es interpretarlos (salvo algunos casos), lo realmente complicado es darles el tratamiento necesario para poder trabajar con ellos.

El objetivo de esta actividad es el de tomar dos series de datos de diferente tamaño pero características similares, darles un tratamiento y despues unirlos para manejar una unica serie.

En la segunda parte de la actividad se explica el metodo que se utilizo y las dificultades presentadas en el proceso y se presentaran los resultados y graficos.

En la tercera parte se da una conclusion de los resultados obtenidos.

2 Desarrollo

Para comenzar hay que aclarar que las series tenian diferente longitud, asi que se tuvo que cortar la serie mas larga para poder unirla con la otra.

Otra dificultad fue que mientras una serie mostraba un año de datos en periodos de diez minutos, la otra mostraba un año en secciones de media hora y por lo tanto se tuvo que usar el siguiente condicional logico para poder hacer el trabajo:

```
df1 = df1[((df1["MIN"]==30.0) — (df1["MIN"]==0.0)) (df1["FECHA"] > 2010-01-01)]
df1 = df1.reset_index(drop = true)
df1.tail()
```

La primera serie de datos ya contaba con la variable fecha, sin embargo la segunda no y fue necesario crearla a partir de los datos.

Para organizar los datos en secciones de media hora en la segunda serie se utilizaron los siguientes comandos:

```
h=[] Arreglo de horas  
m=[] Arreglo de minutos
```

```
for i in range (0, len(df2)):  
    if (len(str(df2['4 Hour_Minute_RTM L'][i]))==4):  
        if (str(df2['4 Hour_Minute_RTM L'][i])[0:2]=='24'):  
            h.append('00')  
            m.append('00')  
        else:  
            h.append(str(df2['4 Hour_Minute_RTM L'][i])[0:2])  
            m.append(str(df2['4 Hour_Minute_RTM L'][i])[2:4])  
    elif (len(str(df2['4 Hour_Minute_RTM L'][i]))==3):  
        h.append(str(df2['4 Hour_Minute_RTM L'][i])[0:1])  
        m.append(str(df2['4 Hour_Minute_RTM L'][i])[1:3])  
    elif (len(str(df2['4 Hour_Minute_RTM L'][i]))==2):  
        h.append('00')  
        m.append(str(df2['4 Hour_Minute_RTM L'][i])[0:2])  
  
    t = [] #Arreglo de tiempo  
for i in range (0,len(df2)):  
    t.append( h[i]+' ':'+m[i])  
  
df2['TIEMPO'] = t
```

El codigo anterior escribe la serie en un formato HH/MM/SS

Ahora se creo una variable tipo date para que Phyton pudiera asignar un valor temporal a los datos, esto se hizo de la siguiente forma:

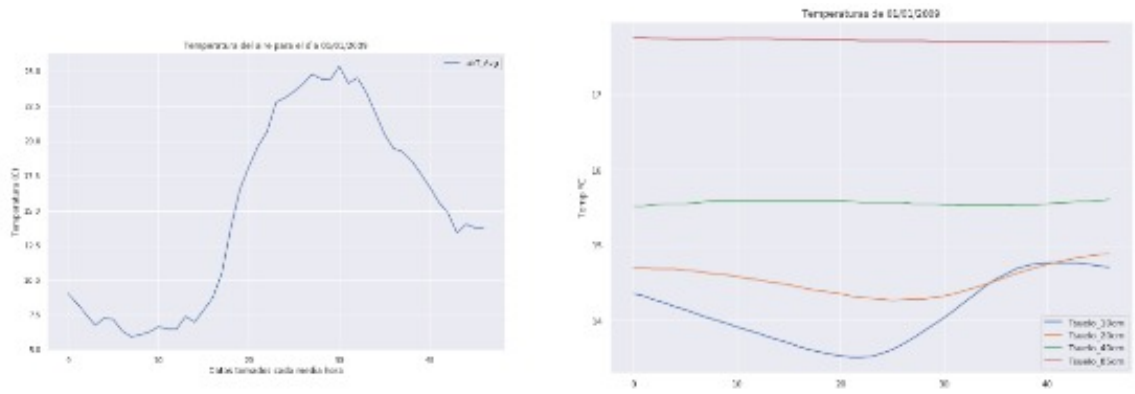


Figura 2: Temperatura del aire y del suelo para el día 1ero de enero de 2009.

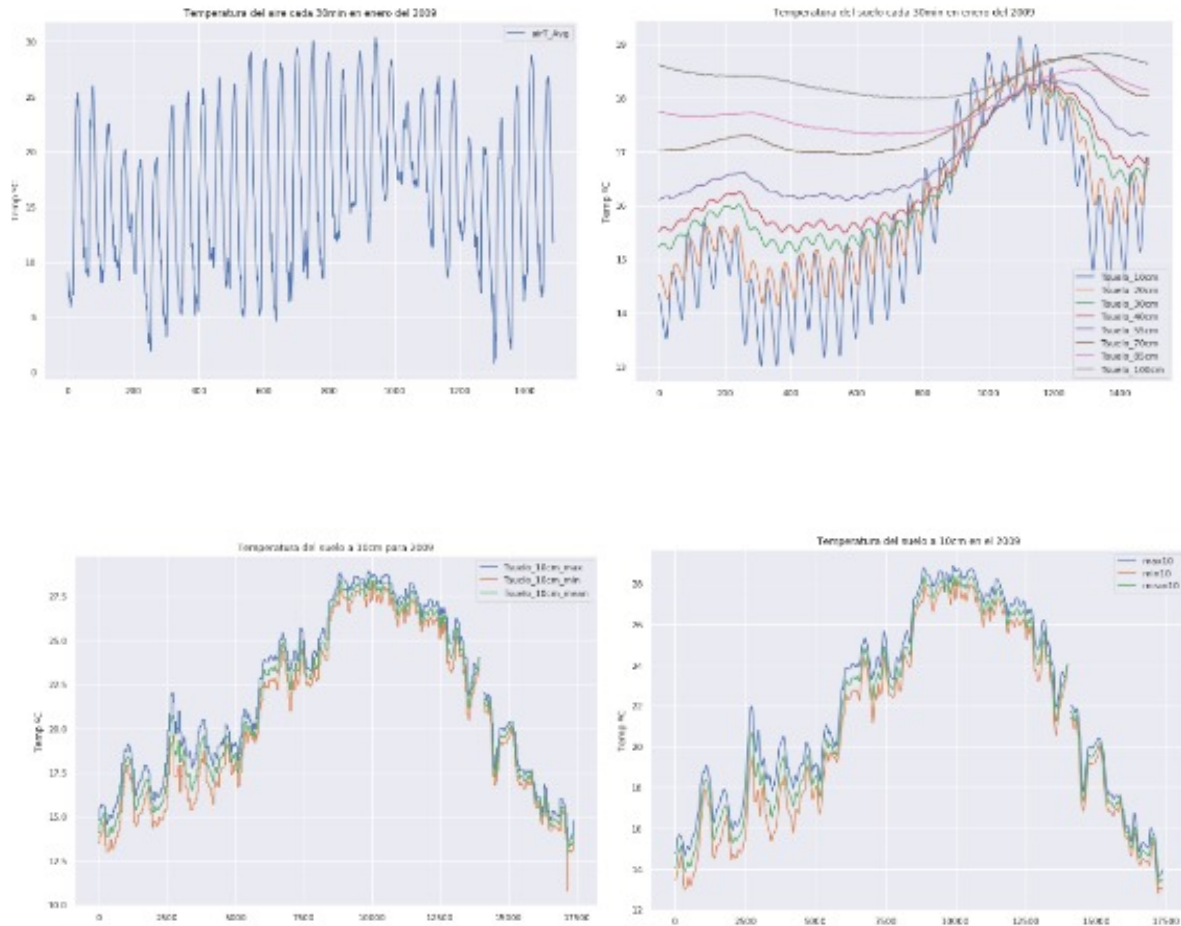


Figura 4: Gráficas de las temperaturas medidas 10cm bajo suelo, antes y después de aplicar el promedio móvil.

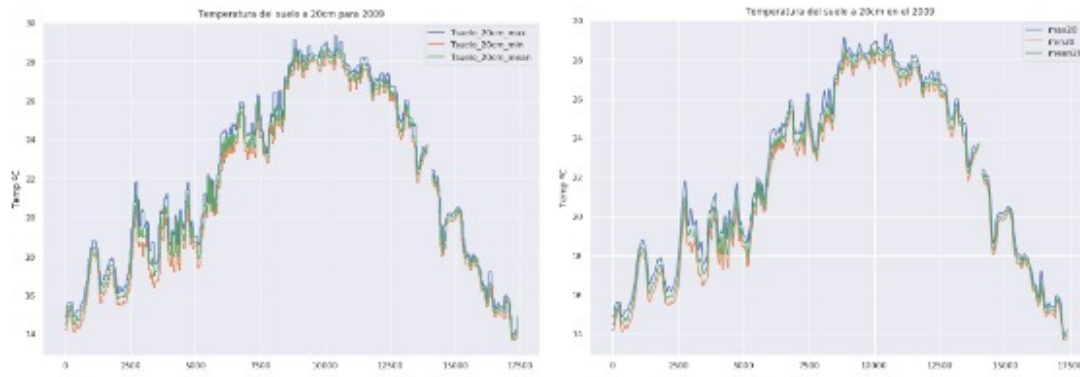


Figura 5: Gráficas de las temperaturas medidas 20cm bajo suelo, antes y después de aplicar el promedio móvil.

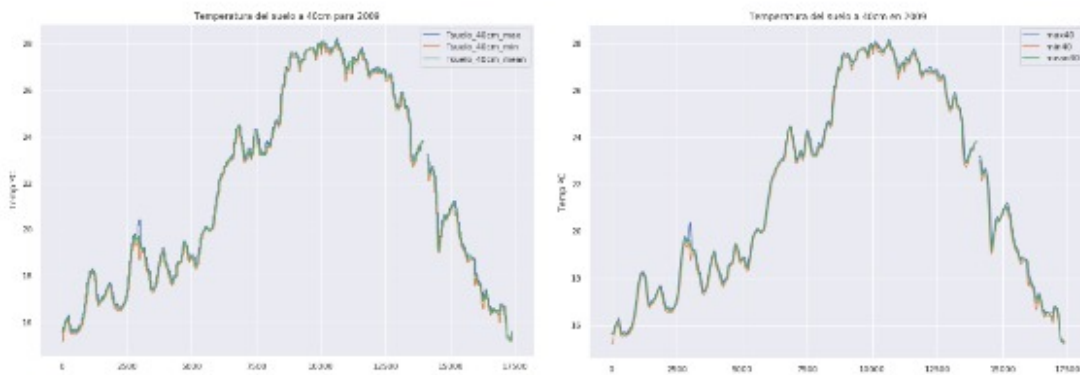


Figura 6: Gráficas de las temperaturas medidas 40cm bajo suelo, antes y después de aplicar el promedio móvil.

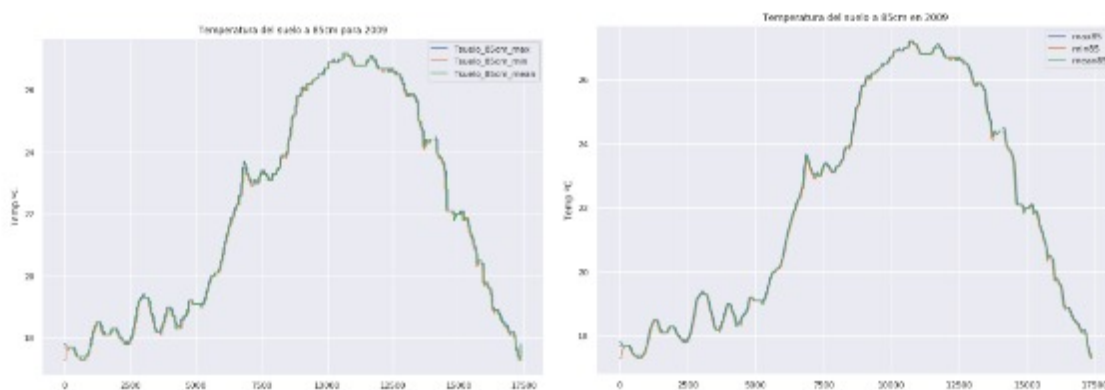


Figura 7: Gráficas de las temperaturas medidas 85cm bajo suelo, antes y después de aplicar el promedio móvil.

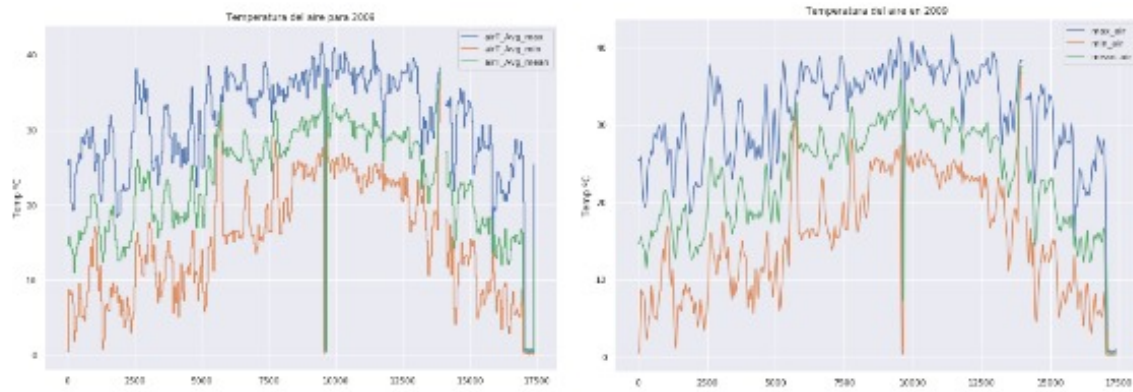


Figura 8: Temperatura del aire para el año 2009, antes y después de aplicar el promedio móvil.

4 Conclusions

Las graficas nos indican que el comportamiento del aire a lo largo del dia es oscilatorio, pues a una hora esta en su mínimo y a otra en su máximo.

Con respecto al suelo se puede apreciar una relacion inversa entre las temperaturas maxima minima y promedio conforme vamos aumentando la profundidad.

Esta actividad fue dificil en el sentido de ajustar las series de datos para poder unirlos.