

Evaluación 2

Física Computacional

Brayan Alexis Ramírez Camacho
Lic. en Física
Universidad de Sonora

05 de Mayo de 2019

1. Introducción

La **Evapotranspiración** (ET) es el proceso por el cual se acumula vapor de agua en la atmósfera debido tanto de la transpiración de las plantas como de la evaporación de la humedad del suelo.

Este proceso es de suma importancia en Agricultura, por lo que la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO) publicó en su reporte número 56 una estimación del uso de agua en cultivos, para lo cual utilizó la ecuación de Penman-Monteith (PM).

La principal limitante de la ecuación de PM es la disponibilidad de mediciones de las variables que involucra, como la temperatura diaria promedio, la velocidad del viento, la humedad relativa y la radiación solar.

En ésta actividad se evalúan las alternativas a la ecuación de PM, apoyándose en el artículo ".Evaluation of the Penman-Monteith and other 34 reference evapotranspiration equations under limited data in a semiarid dry climate" de Koffi Djaman, además de datos meteorológicos tomados de un viñedo ubicado en el kilómetro 41 de la carretera de Hermosillo a Bahía Kino (Latitud $28^{\circ} 55.117'$ N, Longitud $111^{\circ} 18.638'$ W, altitud 101m), en todo el año 2018.

2. Desarrollo

Ésta actividad se divide en tres partes:

1. Construcción de una tabla de promedios mensuales similar a la Tabla 1 del artículo de Djaman.
2. Estimación de la Evapotranspiración mensual promedio (ET_0) utilizando las ecuaciones de los siguientes 3 autores que aparecen en el artículo de K. Djaman: Ec 7, Jansen Haise (1963); Ec. 31, Valiantzas 1 (2012); y Ec. 34, Valiantzas 4 (2013).
3. Elaboración de gráficas para el balance promedio en un mes arbitrario.

2.1. Parte 1

- Primeramente se importan las librerías *Pandas*, *Numpy*, *Matplotlib.pyplot*, *Math* y *Datetime* de Python.

- Se leen los archivos de texto con los datos meteorológicos y se almacenan en Dataframes de Pandas:

```
df1 = pd.DataFrame( pd.read_csv("meteo-vid-2018.csv", engine="python" ) )
df2 = pd.DataFrame( pd.read_csv("flujos-vid-2018.csv", engine="python" ) )
```

- Se eliminan los renglones que contienen solamente datos nulos y se toman las columnas de interés:

```
#Se eliminan los renglones a partir del renglón 17246
df1 = df1.drop(df1.index[17246:]).reset_index(drop=True)
```

```
df1 = df1.filter(['WS_ms_S_WVT', 'AirTC_Avg', 'RH', 'Rn_Avg', 'Date', 'Time', 'albedo_A'])
```

- Se crea una nueva columna "FECHA.^a a partir de las columnas "Date" y "Time", que posteriormente se convierte a una variable de tipo DATE utilizando la función `pd.to_datetime`. Además, de ésta columna se extraen otras dos para almacenar el DIA y MES:

```
df1["DATETIME"] = df1["Date"] + " " + df1["Time"]

df1['FECHA'] = pd.to_datetime(df1.apply(lambda x: x['DATETIME'], 1), dayfirst=True)

df1['DIA'] = df1['FECHA'].dt.day
df1['MES'] = df1['FECHA'].dt.month
```

- Se crean columnas para los valores máximos, mínimos y promedios diarios para la temperatura (T) y la humedad relativa (RH):

```
df1["Tmax"] = df1.groupby(["MES", "DIA"])["AirTC_Avg"].transform("max")
df1["Tmin"] = df1.groupby(["MES", "DIA"])["AirTC_Avg"].transform("min")
df1["Tmean"] = df1.groupby(["MES", "DIA"])["AirTC_Avg"].transform("mean")

df1["RHmax"] = df1.groupby(["MES", "DIA"])["RH"].transform("max")
df1["RHmin"] = df1.groupby(["MES", "DIA"])["RH"].transform("min")
df1["RHmean"] = df1.groupby(["MES", "DIA"])["RH"].transform("mean")
```

- Se crea la tabla con los promedios mensuales para cada variable

```
tabla = pd.DataFrame()

tabla['Vel_prom'] = df1.groupby(['MES'])['WS_ms_S_WVT'].transform('mean')
```

```

tabla['Tmax'] = df1.groupby(['MES'])['Tmax'].transform('mean')
tabla['Tmin'] = df1.groupby(['MES'])['Tmin'].transform('mean')
tabla['Tmean'] = df1.groupby(['MES'])['AirTC_Avg'].transform('mean')
tabla['RHmax'] = df1.groupby(['MES'])['RHmax'].transform('mean')
tabla['RHmin'] = df1.groupby(['MES'])['RHmin'].transform('mean')
tabla['RHmean'] = df1.groupby(['MES'])['RH'].transform('mean')
tabla['Rs'] = df1.groupby(['MES'])['Rn_Avg'].transform('mean')
tabla['Albedo'] = df1.groupby(['MES'])['albedo_Avg'].transform('mean')

```

- Se generan gráficas para la variación mensual de los máximos, mínimos y promedios de las temperaturas, la humedad relativa y para la radiación solar. Éstas se muestran en la sección de Resultados.

2.2. Parte 2

- La primera ecuación en ser evaluada es la ecuación 7, de Jansen Haise:

$$ET_0 = (0,0252T_{mean} + 0,078)R_s \quad (1)$$

- Seguidamente se evalúa la ecuación 31 de Valiantzas:

$$ET_0 = 0,0393R_s(T_{mean}+9,5)^{0,5} - 0,19(R_s^{0,6})(\phi^{0,15}) + 0,0061(T_{mean}+20)(1,11T_{mean}-T_{min}-2)^{0,7} \quad (2)$$

donde ϕ es la latitud en radianes.

- Finalmente la tercera ecuación que se evalúa es la número 34 de Valiantzas:

$$ET_0 = 0,051(1-\alpha)R_s(T_{mean}+9,5)^{0,5} - 2,4(R_s/R_a)^2 + 0,048(T_{mean}+20)(1-RH/100)(0,5+0,536u2)+0,00012z \quad (3)$$

donde α es el albedo superficial, R_a es la radiación solar en lo alto de la atmósfera, $u2$ es la z es la elevación sobre el nivel del mar.

Para realizar el cálculo anterior, primero es necesario conocer el valor de R_a , que se calcula como

$$R_a = \frac{24 \cdot 60}{\pi} \cdot G_{sc} \cdot dr(\omega \sin \phi \sin \delta + \cos \phi \cos \delta \sin \omega)$$

donde G_{sc} es la constante solar ($0.0820 \text{ MJm}^{-2}\text{min}^{-1}$), dr es la distancia relativa entre la Tierra y el Sol, ω es el ángulo de la hora donde se oculta el sol, ϕ es la latitud en rad y δ es la declinación solar en rad.

Para calcular los parámetros dr , δ y ω se utiliza:

$$dr = 1 + 0,033 \cos \left(\frac{2\pi}{365} J \right)$$

$$\delta = 0,409 \sin \left(\frac{2\pi}{365} J - 1,39 \right)$$

$$\omega = \arccos(-\tan(\phi) \tan(0,409 \sin(\frac{2\pi}{365}J - 1,39)))$$

donde J es el día juliano y está dado por

```
J = int( 30.4*k - 15 ) for k in range( 0, len(tabla) )
```

con R_a calculada ahora procedemos a calcular ET_0 .

2.3. Parte 3

En esta sección se trabaja con el archivo de datos de flujos meteorológicos.

- Primero se toman las columnas relevantes para el análisis y se elimina el renglón que contiene las unidades de medición de cada variable.

```
df2 = df2.filter(['Year', 'DoY', 'Hour', 'Rg_f', 'LE_f', 'H_f'],axis=1)
```

```
df2 = df2.drop( df2.index[0] ).reset_index(drop=True)
```

- Se crea una variable de fecha a partir de las columnas "Year", "DoY", "Hour", además, se crean listas para el Día, la Hora y el Minuto:

```
hora=[]
minuto=[]
```

```
for i in range( 0, len(df2) ):
```

```
    #Si contiene 1 dígito:
    if (len(str(df2['Hour'][i]))==1):
        hora.append(str(df2['Hour'][i])[0:1])
        minuto.append('00')
```

```
    #Si contiene 2 dígitos:
    if (len(str(df2['Hour'][i]))==2):
        if (str(df2['Hour'][i])[0:2]=='24'):
            hora.append('00')
            minuto.append('00')
        else:
            hora.append(str(df2['Hour'][i])[0:2])
            minuto.append('00')
```

```
    #Si contiene 3 dígitos:
    elif (len(str(df2['Hour'][i]))==3):
        hora.append(str(df2['Hour'][i])[0:1])
        minuto.append('30')
```

```

#Si contiene 4 dígitos:
elif(len(str(df2['Hour'] [i]))==4):
    hora.append(str(df2['Hour'] [i]) [0:2])
    minuto.append('30')

dias =[df2['DoY'] [i] for i in range(0,len(df2))]

```

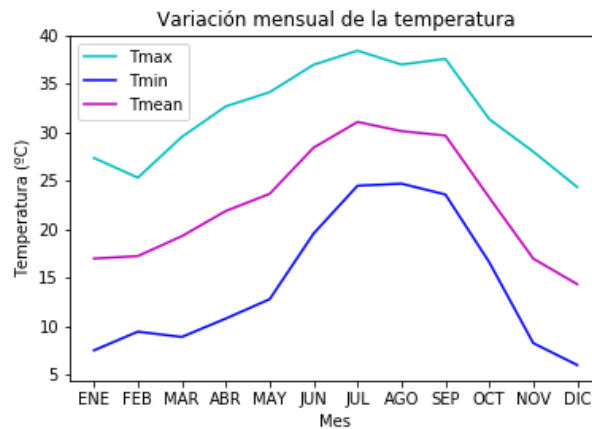
y lo anterior se agrupa en un Dataframe.

- Por último se generan gráficas para el balance de energía entre la radiación neta, el calor latente y el calor sensible. Estas visualizaciones se muestran en la sección de Resultados.

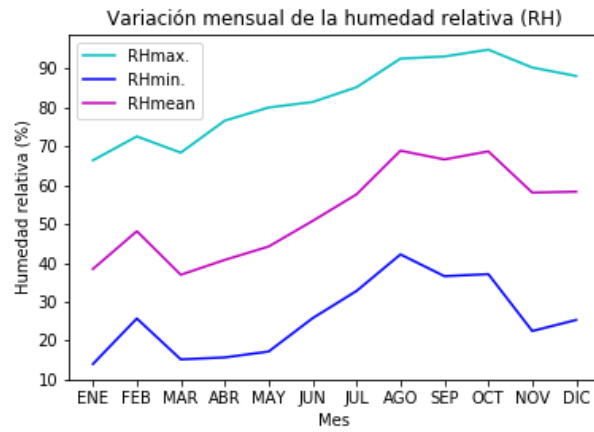
3. Resultados

3.1. Variables meteorológicas

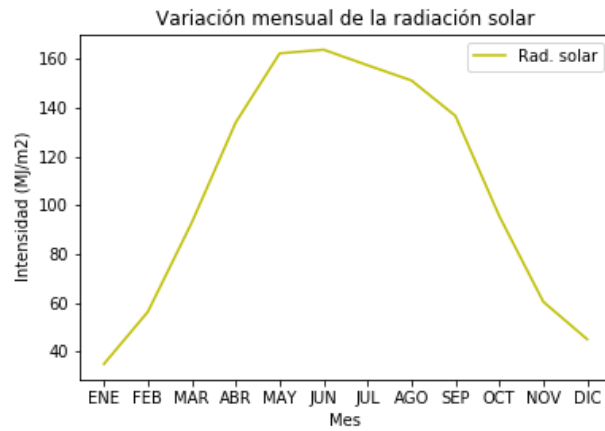
Las gráficas que muestran la variación mensual del máximo, mínimo y promedio de la temperatura.



Para la humedad relativa (RH) se obtuvo:

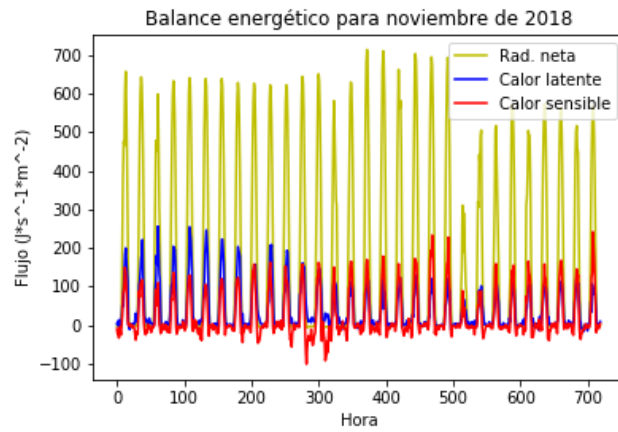


Mientras que para la radiación solar recibida

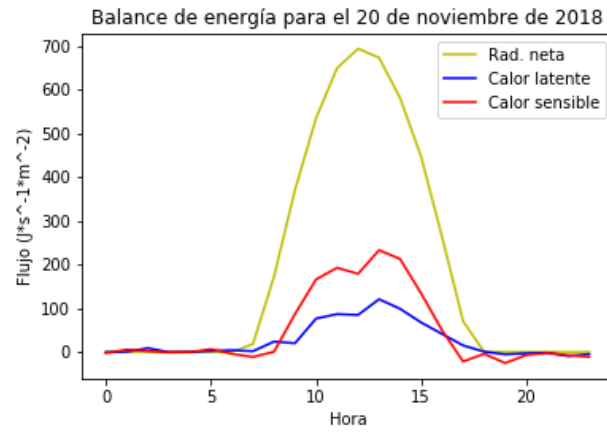


3.2. Balance Energético

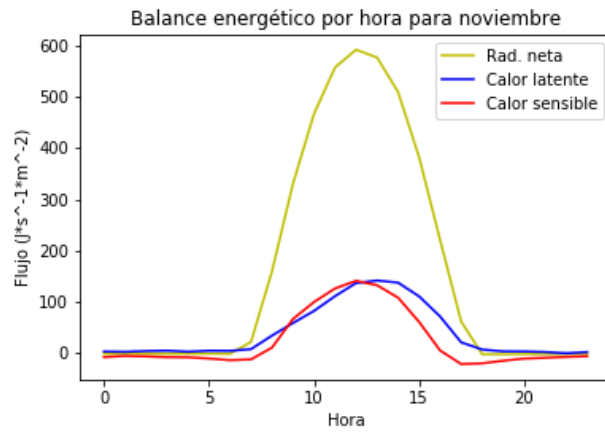
La gráfica obtenida del balance de energía para el mes de noviembre del 2018 es:



en tanto que el balance energético para el 20 de noviembre del 2018 es el siguiente:



y haciendo el promedio por hora para el mismo mes:



4. Conclusión

Del anterior análisis se puede concluir que las ecuaciones Ec 7, Jansen Haise (1963); Ec. 31, Valiantzas 1 (2012) y Ec. 34, Valiantzas 4 (2013) producen buenos resultados al ser utilizadas en la zona árida del centro de Sonora.