

Reporte de la Evaluación 2

Eduardo Peñuñuri Bolado

Grupo 1

5 de mayo de 2019

Reporte

En este caso trabajamos con una evaluación, pero era como una actividad más, y consistió en hacer una tabla, unos cálculos y unas gráficas básicamente utilizando dos archivos con datos (de los cuales solo utilizamos algunas columnas). Se dividió en tres partes, la primera era en la que leíamos el primer archivo para de el conseguir hacer una tabla, en la segunda con lo obtenido en la parte uno realizabamos unos cálculos para obtener algo que se nos pedía, y finalmente la tercera parte, donde usamos el segundo archivo y consistía en realizar una gráfica.

Lo primero que hicimos de todo fue haecer las invocaciones (llamar a pandas, matplotlib, etc.), y leer los datos con los que trabajaríamos, para acto seguido comenzar a realizar las tres partes, las cuales se dividieron en el programa.

En la parte uno lo primero que se hizo fue definir tres constantes (latitud, longitus y altitud), para acto seguido comenzar a realizar lo más difícil y/o pesado; se nos pidió realizar una tabla de promedios mensuales, y los datos estaban en intervalos de media hora, por lo que se debió de hacer promedios, y para ello primero se creó un arreglo que representaba el mes, y en el se ponían los promedios diarios, para después sacar la media del mes. Los resultados se pusieron en otro arreglo, donde cada elemento representaba un mes.

Aquí un fragmento de ejemplo:

```
#Hacemos el promedio mensual de la velocidad del viento
#Para ello, primero sacamos la de cada día (los días se toman desde las 0:00
hasta las 23:59)
#Vamos a agregar los promedios en un arreglo, estarán los doce meses
dfVVM=[0] * 12
dfVV=df1M[["WS_ms_S_WVT"]]

#ENERO#
dfVENE=[0] * 26

#Dado que el primero no comienza desde las 0:00 horas, se saca diferente al
```

```

resto de días
#Día 1
dfVENE[0]=dfVV[0:14].mean()

#Otros días del mes de enero
j=1

for i in range(14,1214,48):
    dfVENE[j]=dfVV[i:i+48].mean()
    j=j+1

dfVENE=pd.DataFrame(dfVENE)
dfVVM[0]=dfVENE.mean()

#FEBRERO#
dfVFEB=[0] * 28
j=0

for i in range(1214,2558,48):
    dfVFEB[j]=dfVV[i:i+48].mean()
    j=j+1

dfVFEB=pd.DataFrame(dfVFEB)
dfVVM[1]=dfVFEB.mean()

```

Cada vez que se ocupaba sacar promedios mensuales se hacía lo mismo (velocidad del viento, temperatura, humedad relativa y radiación solar), sin embargo, también se pedían mínimas y máximas (de temperatura y humedad relativa), y en esos casos simplemente se sacaba el máximo o mínimo de cada mes, y, como en el caso anterior, se ponían en un arreglo donde cada elemento representaba a su respectivo mes.

Un pedazo del programa para darse una idea:

```

#Ahora vamos por la temperatura máxima
#Para ello agarraremos la temperatura máxima de cada mes
dfTMaxM=[0] * 12
dfT=df1M[["AirTC_Avg"]]

#ENERO#
dfTMaxM[0]=dfT[0:1214].max()

#FEBRERO#
dfTMaxM[1]=dfT[1214:2558].max()

```

Una vez terminado cada una de las columnas, antes de realizar la tabla final, se realizaron unos arreglos más para que quedara más presentable, entre ellas el hacer un arreglo de doce

elementos donde todos fueran la latitud, otro donde fueran la longitud, y uno más para la altitud, además de uno donde estuvieran los meses, esta vez escritos con letras; y para juntar todo en una tabla, se hizo uso del siguiente código:

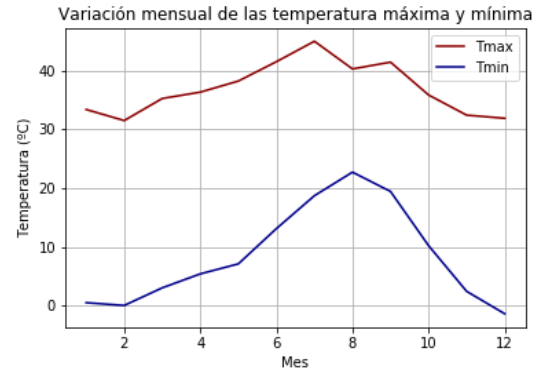
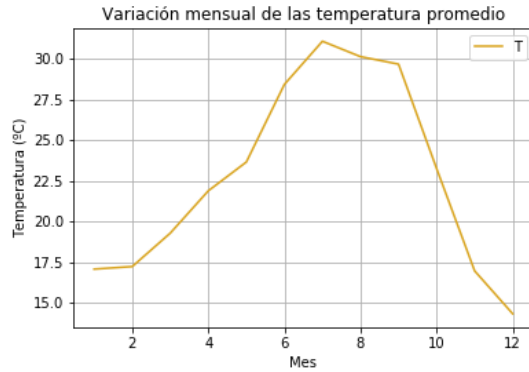
```
#Ahora haremos la tabla
dfTABLA=pd.DataFrame(df0)
dfTABLA["Mes"]=dfMESES
dfTABLA["Latitud (N)"]=dfLATITUD
dfTABLA["Longitud (O)"]=dfLONGITUD
dfTABLA["Elevación (m)"]=dfALTITUD
dfTABLA["Velocidad del Viento (m/s)"]=dfOVVM
dfTABLA["Tmax (C)"]=dfOTMaxM
dfTABLA["Tmin (C)"]=dfOTMinM
dfTABLA["Tmean (C)"]=dfOTM
dfTABLA["RHmax (%)"]=dfORHMaxM
dfTABLA["RHmin (%)"]=dfORHMinM
dfTABLA["RHmean (%)"]=dfORHM
dfTABLA["Rs (MJ/m2/día)"]=dfORSM
dfTABLA
```

Finalmente, la tabla quedó:

0	Mes	Latitud (°N)	Longitud (°O)	Elevación (m)	Velocidad del Viento (m/s)	Tmax (°C)	Tmin (°C)	Tmean (°C)	RHmax (%)	RHmin (%)	RHmean (%)	Rs (MJ/m2/día)	
0	0	Enero	28.918617	-111.310633	101	1.941529	33.35	0.54	17.074827	91.60	5.98	38.372184	32.917208
1	0	Febrero	28.918617	-111.310633	101	1.964189	31.47	0.07	17.230275	99.53	6.76	48.168006	56.250350
2	0	Marzon	28.918617	-111.310633	101	1.926196	35.22	3.06	19.282359	89.73	6.76	36.968353	92.894913
3	0	Abril	28.918617	-111.310633	101	2.101813	36.30	5.43	21.880618	93.40	5.23	40.785667	134.012965
4	0	Mayo	28.918617	-111.310633	101	2.113918	38.18	7.13	23.650034	94.60	8.19	44.233468	162.405343
5	0	Junio	28.918617	-111.310633	101	2.154986	41.47	13.12	28.416187	98.37	5.34	50.810507	163.926069
6	0	Julio	28.918617	-111.310633	101	2.022204	44.94	18.71	31.065726	97.00	13.10	57.639805	157.534402
7	0	Agosto	28.918617	-111.310633	101	1.910853	40.24	22.71	30.120894	98.60	30.16	68.868454	151.273589
8	0	Septiembre	28.918617	-111.310633	101	1.790326	41.39	19.43	29.661271	98.03	21.32	66.619750	136.749910
9	0	Octubre	28.918617	-111.310633	101	1.664435	35.82	10.23	23.254207	98.47	17.32	68.696082	95.513468
10	0	Noviembre	28.918617	-111.310633	101	1.498097	32.39	2.47	16.966396	99.43	8.75	58.085458	60.384285
11	0	Diciembre	28.918617	-111.310633	101	1.628730	31.86	-1.34	14.332823	97.33	8.04	58.333938	44.962366

Además de esto, se nos pidieron tres gráficas de la variación mensual de las temperaturas, la humedad relativa y la radiación solar, para las cuales se utilizaban en el eje de las x los meses (escritos de forma numérica [enero es 1, febrero 2, etc.]), y en el de las y la variación mensual que se pedía. Algo a mencionar es que se hicieron dos gráficas para las temperaturas, una donde salía la media mensual, y otro donde están la máximas y mínimas también de cada mes.

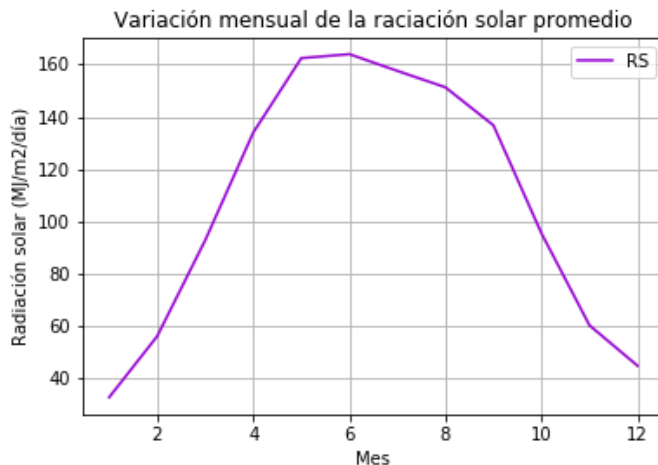
En las de temperaturas podemos ver como va variando según el mes, comenzando de poco y aumentando gradualmente hasta que cerca de los 2/3 comienza a bajar nuevamente, hasta llegar a algo similar a como comenzó, y es debido simplemente a que la temperatura varía según la estación del año.



Siguiendo con la humedad relativa, aunque similar al caso anterior aumente y luego disminuya, podemos ver que al final queda bastante más alta que como comenzó, además de ser algo más impredecible y “violenta” en cuanto a la forma de la curva.



Finalmente la radiación solar si es más parecida a la temperatura, además de que termina bastante parecido a como comenzó; seguramente es debido a que depende la época del año hay más o menos horas de sol, además de que el sol se encuentra a diferente distancia de la tierra según lo mismo.



Después de esto comenzamos con la parte dos, la cual consistió en sacar la evaporación mensual promedio (ET0) con tres ecuaciones diferentes: la de Jansen Haise, Valiantzas 1 y Valiantzas 4.

La de Jansen Haise es simplemente $ET0 = (0.0252T + 0.078)Rs$, por lo que el código fue sencillo:

```
dfET0JH=[0] * 12

#ENERO#
dfET0JH[0]=(0.0252*dfTM[0].item() + 0.078)*dfRSM[0].item()

#FEBRERO#
dfET0JH[1]=(0.0252*dfTM[1].item() + 0.078)*dfRSM[1].item()
```

Con Valiantzas comenzó a complicarse un poco; Valiantzas 1 no fue difícil, ya que es $ET0 = 0.0393Rs(Tmean + 9.5)**0.05 - (0.19Rs**0.6)(phi**0.15) + 0.0061(Tmean + 20)(1.12Tmean - Tmin - 2)**0.7$, y simplemente ocupamos la altitud en radianes ($phi=0.5047$), por lo que también fue sencillamente lo y rápido:

```
dfET0V1=[0] * 12
phi=0.5047

#ENERO#
dfET0V1[0]=0.0393*dfRSM[0].item()*(dfTM[0].item()+9.5)**0.05
-(0.19*dfRSM[0].item()**0.6)*(phi**0.15)
+0.0061*(dfTM[0].item()+20)*(1.12*dfTM[0].item()-dfTMinM[0].item()-2)**0.7

#FEBRERO#
dfET0V1[1]=0.0393*dfRSM[1].item()*(dfTM[1].item()+9.5)**0.05
-(0.19*dfRSM[1].item()**0.6)*(phi**0.15)
+0.0061*(dfTM[1].item()+20)*(1.12*dfTM[1].item()-dfTMinM[1].item()-2)**0.7
```

En cuanto a Valiantzas 4 fue más tedioso y complicado, ya que se requirieron muchas fórmulas, las cuales expliqué en el mismo código, así que considero más sencillo simplemente poner el fragmento:

```
#Ahora vamos con la ET0 de Valiantzas 4
#La ecuación es:  $ET0 = 0.051(1 - \alpha)Rs(Tmean + 9.5)**0.5 - 2.4(Rs/Ra)**2 + 0.048(Tmean + 20)(1 - RH/100)(0.5 + 0.536u2) + 0.00012z$ 
#Donde alpha es el albedo (Albedo_Avg)
#Ra es la radiación solar en la parte alta de la atmósfera
#u2 es la velocidad del viento a 2 m de altura (WS_ms_S_WVT)
#z es la altura sobre el nivel mar

#Para sacar Ra usamos la siguiente ecuación:
#Ra = ((24*60)/pi)Gsc*dr[omega_s*Sen(phi)Sen(delta) +
```

```

Cos(phi)Cos(delta)Sen(omega_s)]
#Donde Gsc es la constante solar, y vale 0.0820 MJ m-2 min-1
#dr es la inversa relativa de la distancia Tierra-Sol (dr = 1 +
0.033Cos((2pi/365)*J)) --- (J es el número del día en el año)
#omega_s es el ángulo de la hora del atardecer en radianes
(omega_s = Arccos[-Tan(phi)Tan(delta)])
#phi es la latitud en radianes (0.5047 rad)
#delta es la declinación solar (delta = 0.409Sen((2pi/365)*J - 1.39)) --- (J
es el número del día en el año)

```

Finalmente, acomodamos los resultados en una tabla para que resultaran mejor a la vista y se vieran más ordenados.

0	Mes	ET0 de Jansen & Haise	ET0 de Valiantzas 1	ET0 de Valiantzas 4
0 0	Enero	16.731344	1.743964	8.558203
1 0	Febrero	28.811594	2.346682	12.450646
2 0	Marzon	52.384876	3.426181	17.398591
3 0	Abril	84.346631	4.879617	23.513833
4 0	Mayo	109.458090	5.931224	27.500213
5 0	Junio	130.171712	6.190545	34.608303
6 0	Julio	135.614481	5.864700	25.857740
7 0	Agosto	126.623032	5.088093	29.961431
8 0	Septiembre	112.882131	4.880626	26.059452
9 0	Octubre	63.421517	3.483240	17.716811
10 0	Noviembre	30.527467	2.255607	10.472542
11 0	Diciembre	19.746892	1.807847	6.462251

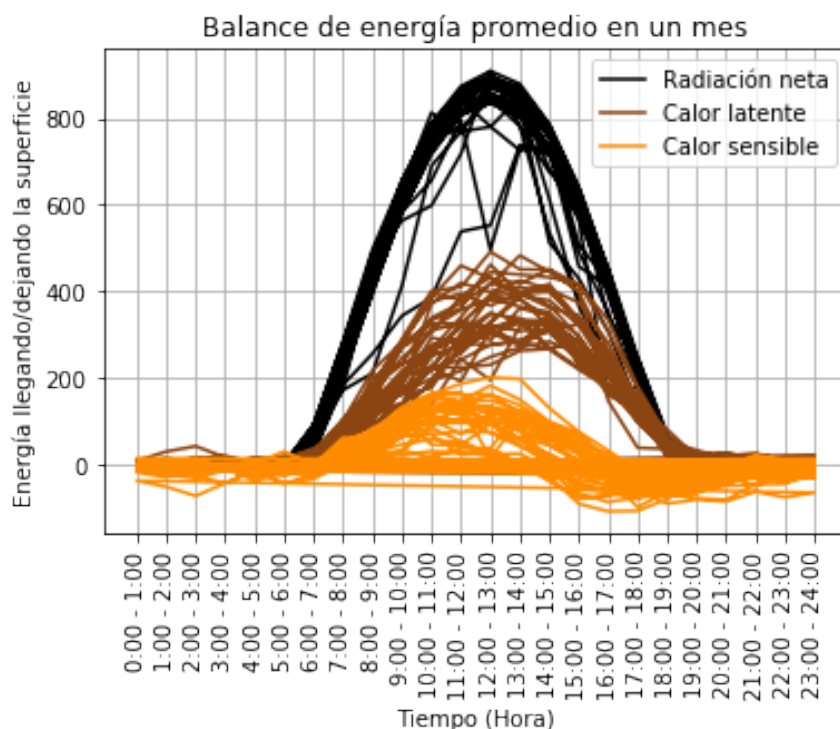
Por último, terminamos con la parte tres, y realmente es la menos tardada y la de menor dificultad, ya que consiste simplemente en una gráfica. Esta vez trabajamos con el segundo archivo, así que hicimos los procedimientos necesarios (leerlo, examinarlo, etc.), y sacamos las columnas que nos interesaban.

Debíamos realizar una gráfica del balance de energía promedio en un mes típico (promedio por hora en un mes), así que primero sacamos los datos que necesitábamos del documento (ya que ocupábamos radiación neta, calor latente y calor sensible), para después, como estaban en intervalos de media hora, hacer medias de cada hora (tomábamos dos y les sacábamos el promedio), y se puso todo en un arreglo.

Después hicimos un arreglo donde salieran las horas de las 0:00 a las 24:00 (1:00 - 2:00, 16:00 - 17:00, etc.), e hicimos que se repitiera 30 veces (por los 30 días del mes), e hicimos otro con los días del mes, repitiéndose cada uno 24 veces (por las horas del día); se realizó

de esta manera simplemente porque al final quise acomodarlo en una tabla y que en esta estuviera presentable, pero en realidad para la gráfica solo ocupaba la de las horas.

Con esto, como ya se comentó, se hizo una tabla (la cual no se anexa porque es demasiado grande, pero en el programa se puede llegar a ver), para posteriormente graficar lo que se pedía, haciendo que el eje x fuera el arreglo de las horas, y el eje y la radiación neta, el calor latente y el calor sensible, separando las curvas por colores, pero todas dentro del mismo gráfico.



Conclusión

La actividad no fue muy diferente al resto de actividades, pero si estuvo tediosa y bastante tardada, en especial las dos primeras partes, más que nada porque eran muchos promedios que sacar, y en la segunda parte por Valiantzas 4, ya que requirió de demasiadas ecuaciones, ya que unas requerían otras, que a su vez requerían otras, además de que requerí sacar más medias, y era algo confuso, pero al final se logró terminar; fuera de esto, la actividad pudo ser peor (supongo), aunque si consumió mucho tiempo, tardando más de un día entero sin descanso en terminarse (me refiero al total, no a que haya estado literalmente sin dormir más de un día entero, aunque si fueron intervalos largos y tuve que sacrificar horas de sueño, ya que tenía otros pendientes y compromisos). Al final se terminó correctamente (o al menos eso creo yo), tras un duro trabajo y mucha investigación.

Referencias

- [1] Allen, G. R., Pereira, L. S., Raes, D. and Smith, M. (1998). *Crop evapotranspiration - Guidelines for computing crop water requirements - FAO Irrigation and drainage paper 56*. Recuperado en mayo de 2019 de: <http://www.fao.org/3/X0490E/x0490e00.htm>
- [2] Djaman, K., O'Neill, M., Diop, L., Bodian, A., Allen, S., Koudahe, K., and Lombard, K. (2 September 2018).. *Evaluation of the Penman-Monteith and other 34 reference evapotranspiration equations under limited data in a semiarid dry climate*. Springer, 15 pp. Mayo de 2019.
- [3] Matplotlib development team (2012). *Matplotlib*. Recuperado en marzo de 2019 de: <https://matplotlib.org/>