Reporte de Matplotlib

Eduardo Peñuñuri Bolado
Grupo 1

17 de marzo de 2019

Reporte

En la actividad realizada, hicimos básicamente gráficos pero utilizando, además de Matplotlib, Seaborn, para poder comparar los gráficos, facilidad de usarlos, etc., y así poder ver cual es mejor o más eficiente y/o sencillo de usar, además de que hicimos otros gráficos con los datos trabajados. Cabe aclarar, que para ello, a parte de invocar Pandas, se hizo lo mismo con Matplotlib y Seaborn.

Lo primero que hice fue trabajar los datos que se nos proporcionaron, ya que contaba con algunos espacios vacíos o regiones innecesarias, que podrían hacer que los resultados finales se vieran perjudicados negativamente, generando errores importantes que luego serían difíciles de erradicar. Para ello, después de leer los datos en Phyton, ejecutamos el siguiente comando para eliminar huecos sin información de las tablas:

```
#Eliminamos las columnas que no contienen datos
df1 = df0.drop(df0.columns[df0.columns.str.contains('unnamed:',case=False)],axis=1)
#df1
```

Posteriormente, revisamos el tipo de variable de las columnas (haciendo uso de .dtypes), y filtramos los datos, para quedarnos con los de 2019:

```
#Eliminamos los datos que sean de 2010
df1.drop(range(52338,52639),axis=0)
df1.head()
```

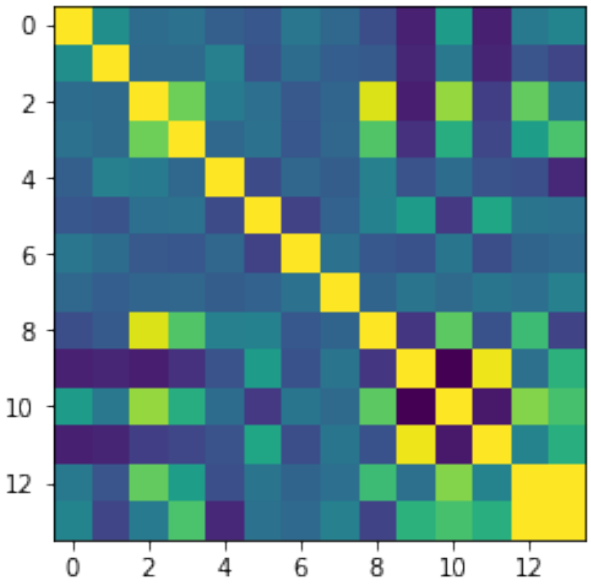
Y creamos otra columna para la fecha, para posteriormente volver a checar los tipos de variables y ver que todo se encuentre en orden:

```
#Creamos un la columna Fecha para la fecha y quitamos la columna de DATE
#Primero hacemos otro DataFrame pero sin el primer renglón (el de las unidades),
para luego trabajar con ella
dfA = df1
dfA = df1.drop(0,0)
```

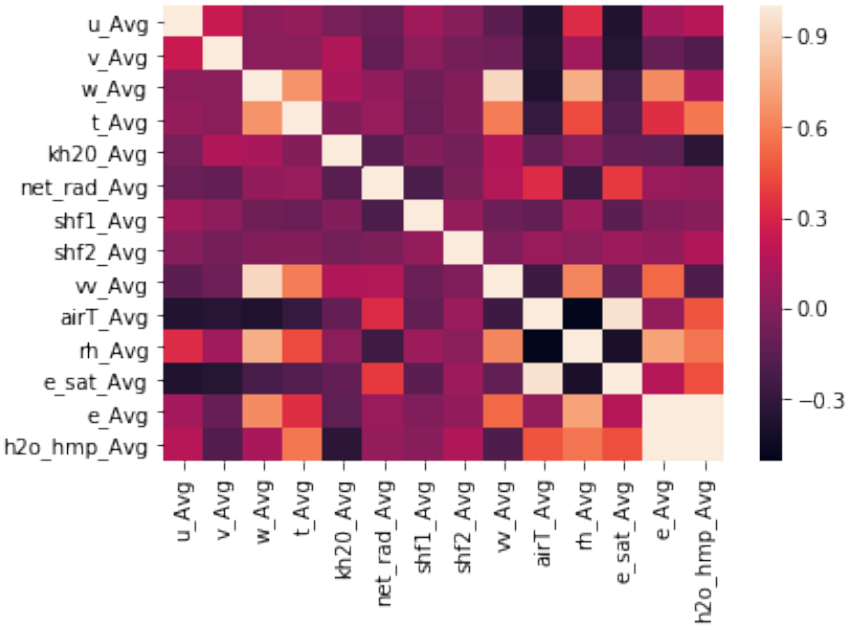
```
#Creamos un la columna Fecha para la fecha y quitamos la columna de DATE
dfA["FECHA"] = pd.to_datetime(dfA.apply(lambda x:x["DATE"],1),dayfirst=True)
df2 = dfA.drop(["DATE"],1)
df2.head()
```

Lo que siguió en la actividad fue mostrar la correlación, y fue tan sencillo como utilizar la función corr(), y una vez obtenido esto, hicimos una gráfica o mapa de correlaciones utilizando Matplotlib y Seaborn, dando lo siguiente:

Matplotlib



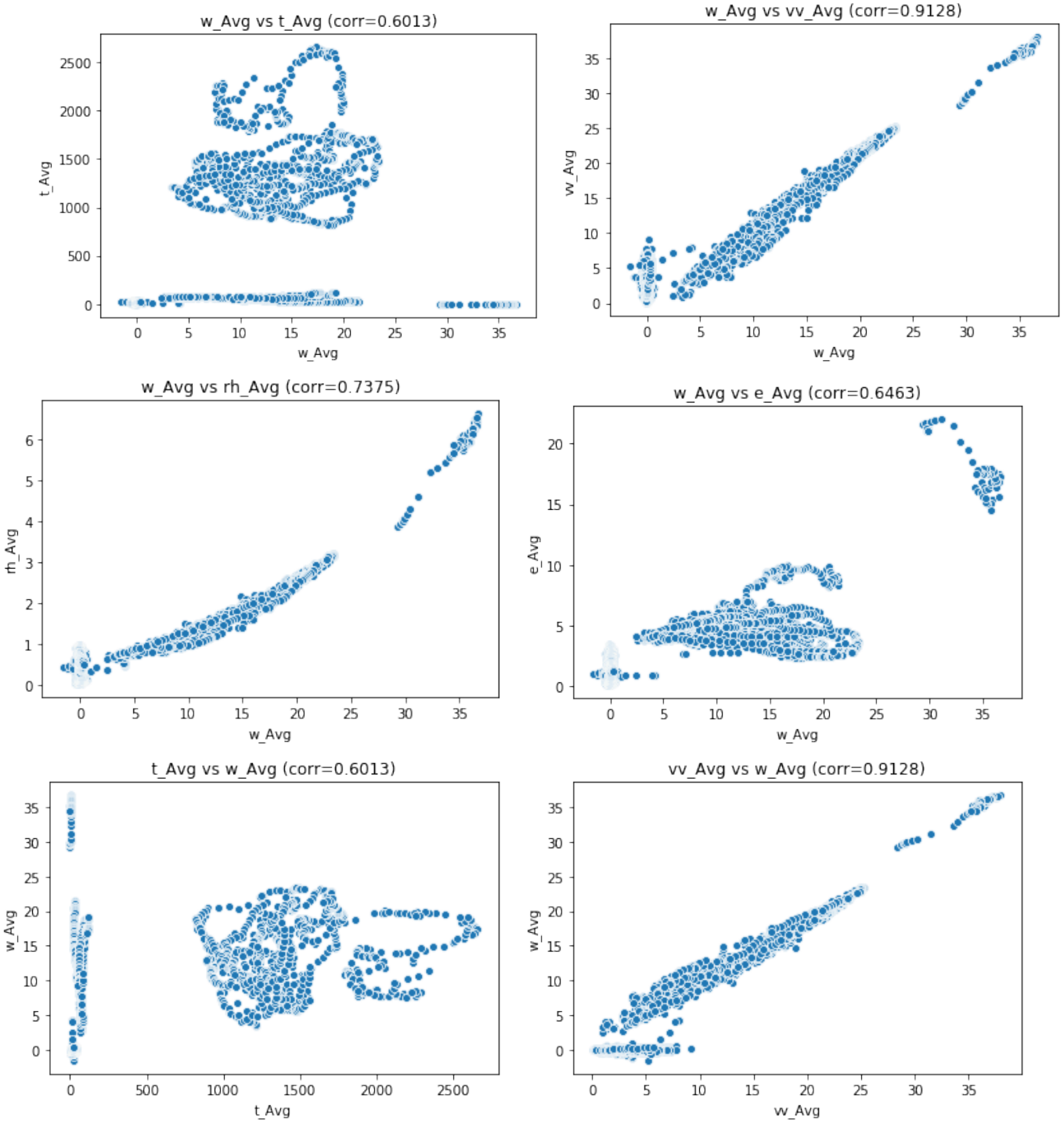
Seaborn

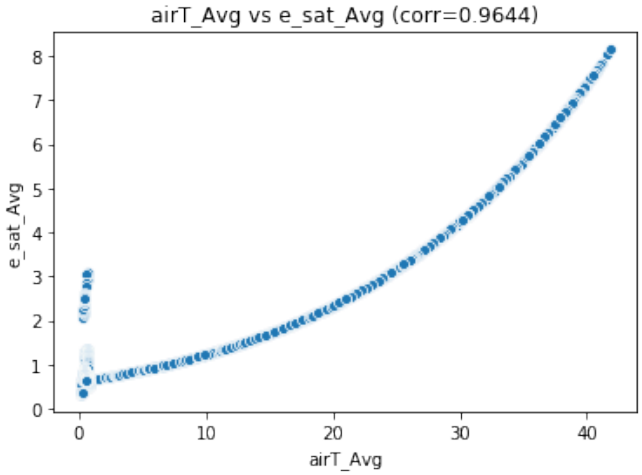


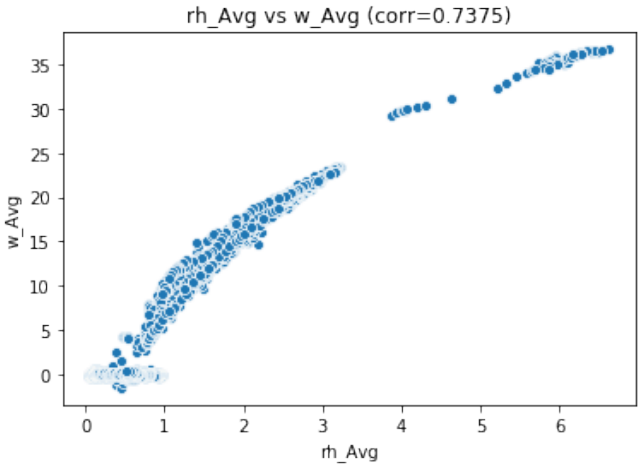
Por último, hicimos gráficas de dispersión de puntos de "Variable 1 vs. Variable 2", para

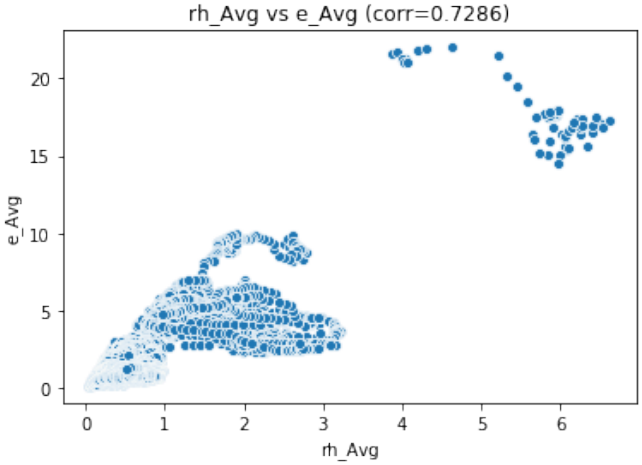
los casos donde la correlación abs(corr) mayor a 6, siendo estos casos:
$(\mathrm{w}_A vg, t_A vg), (w_A vg, vv_A vg), (w_A vg, rh_A vg), (w_A vg, e_A vg), (t_A vg, w_A vg), (vv_A vg, w_A vg)$,
$(\mathrm{airT}_A vg, e_s at_A vg), (rh_A vg, w_A vg), (rh_A vg, e_A vg), (e_s at_A vg, airT_A vg), (e_A vg, w_A vg)$,
$(\mathrm{e}_A vg, rh_A vg), (e_A vg, h2o_h mp_A vg), (h2o_h mp_A vg, e_A vg)$.

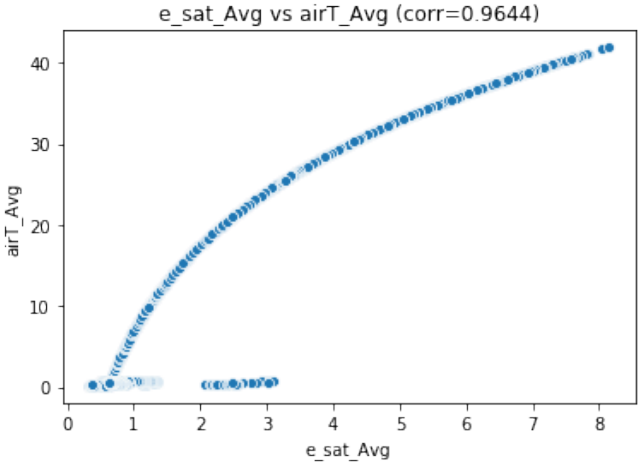
El resultado es las gráficas que se mostrarán a continuación:

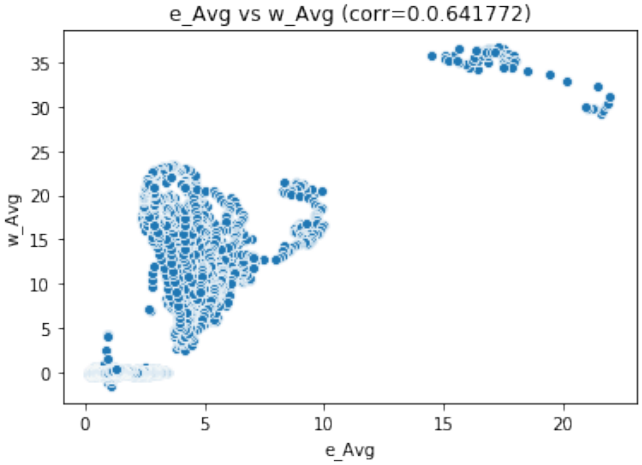


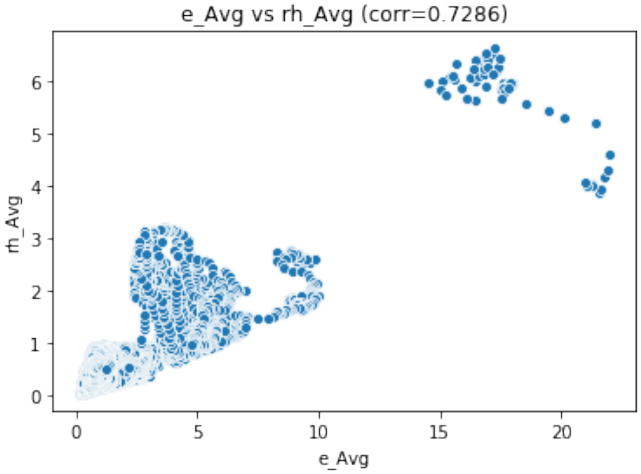

airT_Avg vs e_sat_Avg (corr=0.9644)
e_sat_Avg
airT_Avg

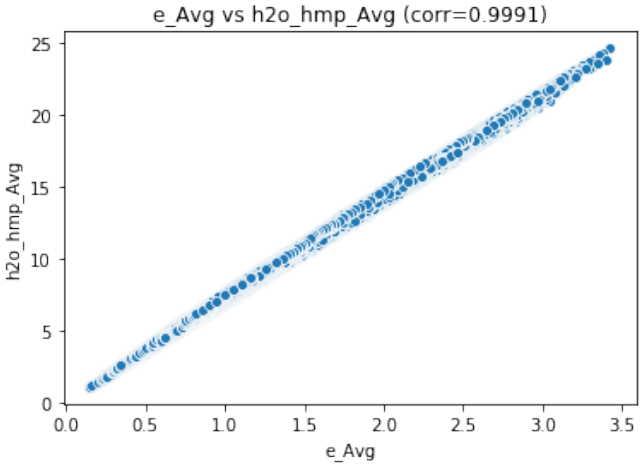

rh_Avg vs w_Avg (corr=0.7375)
w_Avg
rh_Avg

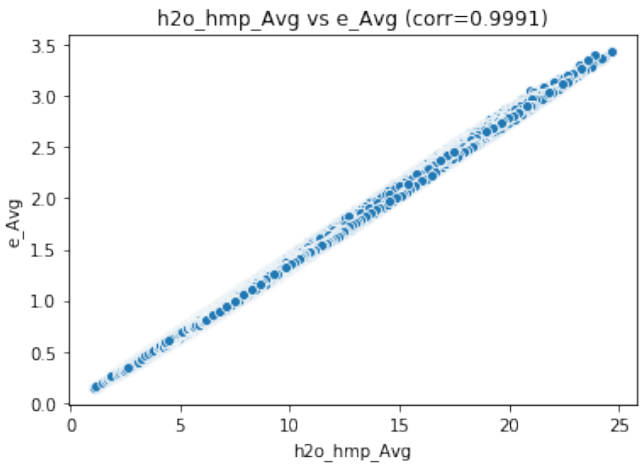

rh_Avg vs e_Avg (corr=0.7286)
e_Avg
rh_Avg


e_sat_Avg vs airT_Avg (corr=0.9644)
airT_Avg
e_sat_Avg


e_Avg vs w_Avg (corr=0.0.641772)
w_Avg
e_Avg


e_Avg vs rh_Avg (corr=0.7286)
rh_Avg
e_Avg





Conclusión

La actividad mostró otra forma de graficar a parte de Matplotlib, la cual es bastante útil y variada, aunque para ser sincero no encontré mucha diferencia en cuanto a la dificultad para usarla, pero si en los resultados, ya que considero que son más completas o presentables con Seaborn que con Matplotlib, sin necesidad de agregar mucho código, mientras que con el otro si queríamos columnas con nombres o cosas similares requería un esfuerzo extra, por lo que Seaborn entrega mejores resultados con menos esfuerzo o dificultad, pero fuera de eso, ambos son similares y pueden utilizarse sin problema, dando una buena variedad de opciones si se sabe buscar.

Referencias

[1] Michael Waskom (2012).*Seaborn: Statistical data visualization*. Recuperado en marzo de 2019 de: https://seaborn.pydata.org/

[2] Matplotlib development team (2012).*Matplotlib*. Recuperado en marzo de 2019 de: https://matplotlib.org/

[3] The python graph gallery (2017).*Seaborn*. Recuperado en marzo de 2019 de: https://python-graph-gallery.com/seaborn/