

Actividad 7

Física Computacional

Brayan Alexis Ramírez Camacho
Lic. en Física
Universidad de Sonora

17 de Marzo de 2019

1. Introducción

En esta actividad se analizan datos recogidos en una estación meteorológica ubicada en un campo de nogal, con registros del año 2009 y parte del 2010.

2. Desarrollo

Antes de poder trabajar con los datos, es necesario eliminar las variables o columnas que no aportan información al análisis, para ello se utiliza la función `drop()`, después de haber leído el archivo de texto con los datos y de haber guardado la información en un `DataFrame`. También es necesario eliminar el primer renglón, que contiene las unidades de medición de cada variable.

Las variables del `DataFrame` son de tipo *Object*, y es necesario transformarlas a variables tipo *float*; esto se logra con la función `.apply(pd.to_numeric)`.

Una función de gran utilidad en el análisis de datos es `corr()`, que genera una matriz de correlaciones entre las diferentes variables del `DataFrame`. Esto permite conocer cuáles variables presentan cierto nivel de correlación. En el siguiente fragmento de código se muestra el uso de esta función:

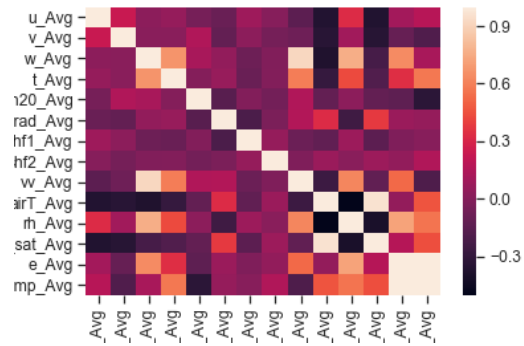
```
df1 = df.corr(method='pearson', min_periods=1)
```

Una herramienta complementaria a la utilización de la función `corr()` es un *heatmap* (mapa de calor), que es un diagrama de color que ilustra precisamente la correlación entre las variables.

Ésta puede ser elaborada utilizando la librería *Seaborn*, como se muestra a continuación:

```
hm_sb = sb.heatmap( df1 )
```

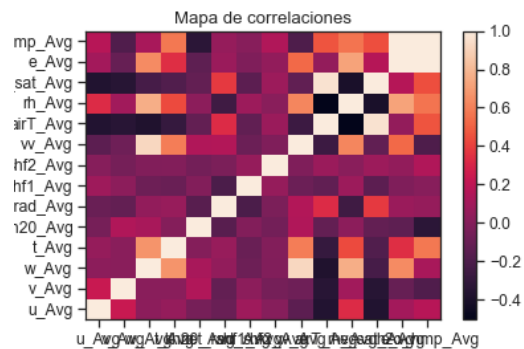
produciendo el siguiente diagrama:



Seaborn no es la única opción para elaborar éste tipo de diagramas. A continuación se muestra una alternativa utilizando Matplotlib:

```
#Heatmap hecho con Matplotlib, usando la función pcolor()
plt.pcolor(df1)
plt.yticks(np.arange(0.5, len(df1.index), 1), df1.index)
plt.xticks(np.arange(0.5, len(df1.columns), 1), df1.columns)
plt.colorbar()
plt.title('Mapa de correlaciones')
```

que produce la gráfica:



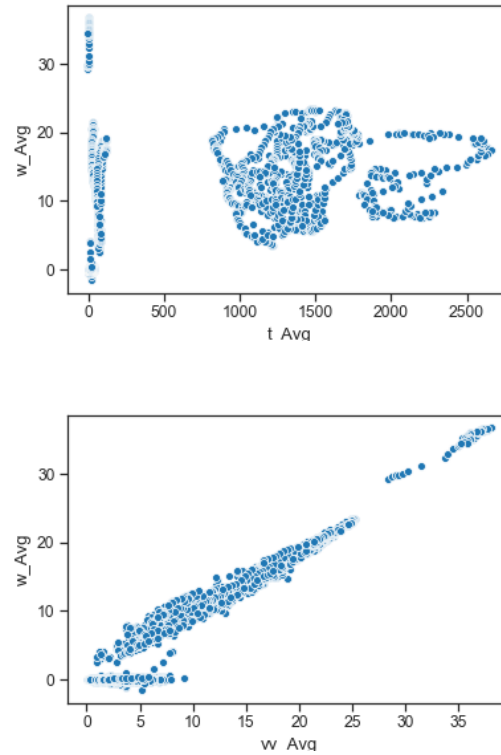
En ambas gráficas, es posible observar cierta correlación entre algunos pares de variables. Lo siguiente es graficar cada caso donde se cumpla que $\text{abs}(\text{corr}) > 0,6$. Los casos donde ésto se cumple son:

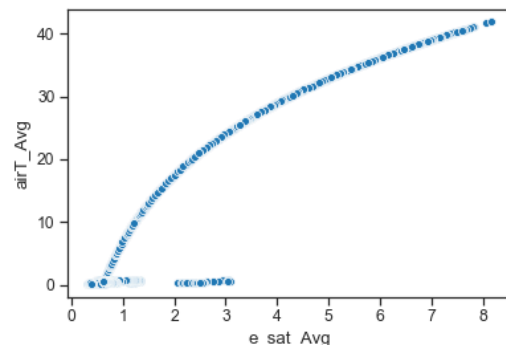
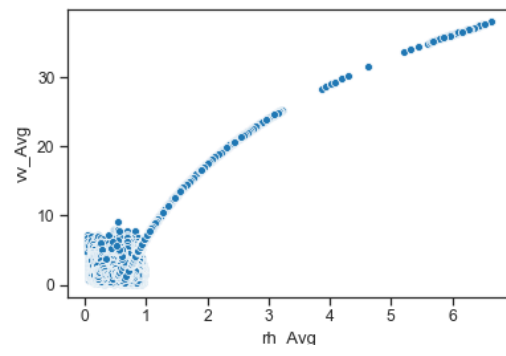
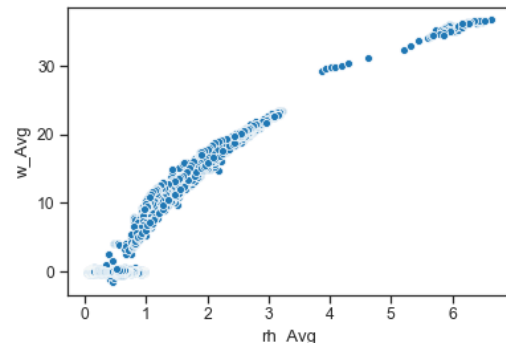
- t_Avg y w_Avg
- vv_Avg y w_Avg
- rh_Avg y w_Avg
- rh_Avg y vv_Avg

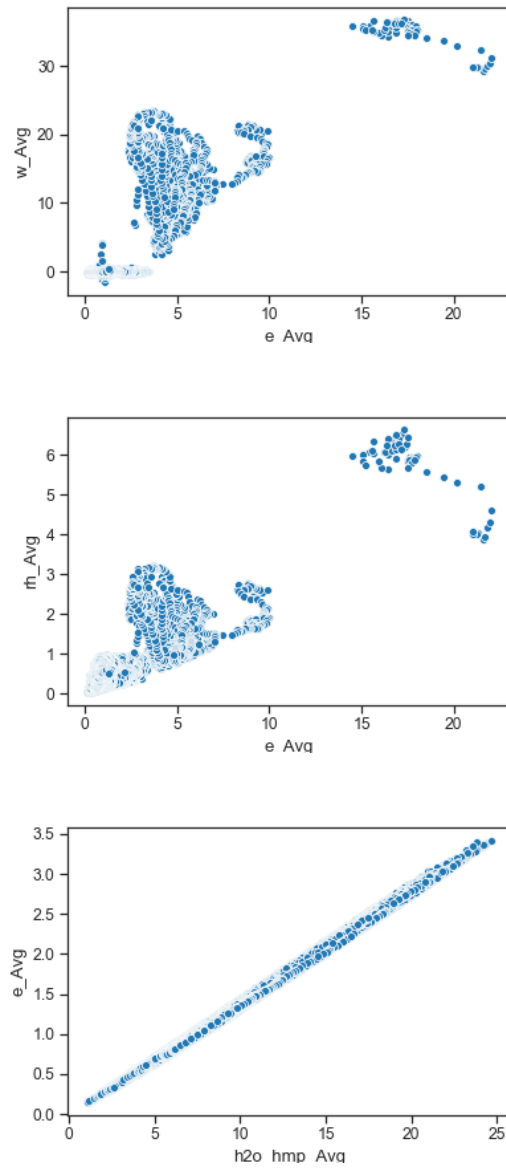
- e_{sat_Avg} y $airT_Avg$
- e_Avg y w_Avg
- e_Avg y rh_Avg
- $h2o_hmp_Avg$ y e_Avg

3. Resultados

Al realizar una gráfica de dispersión de puntos (*ScatterPlot*) para cada par de entre las variables mencionadas anteriormente, es posible observar la relación entre ellas. A continuación se muestran dichas gráficas:







4. Conclusiones

La función `corr()` de Python es muy poderosa en análisis de datos, ya que facilita en gran medida la tarea de encontrar correlaciones entre variables, aún cuando no se tiene información sobre las mismas.

A su vez, la biblioteca de visualización Seaborn ofrece sencillas maneras de elaborar gráficas para ilustrar relaciones entre variables.