

# Reporte de la Actividad 7

## Visualización de datos con la biblioteca Seaborn

Luis Fernando Duarte Gonzalí  
Universidad de Sonora  
Física Computacional

Marzo del 2019

### 1. Introducción

#### 1.1. Seaborn

Seaborn es una librería de visualización de datos de Python basado en matplotlib. Provee una interfaz de alto nivel para dibujar gráficas estadísticas atractivas e informativas.

Seaborn tiene como objetivo el hacer la visualización una parte central de exploración y entendimiento de datos. Sus funciones para graficar orientadas en datasets operan en *Dataframes* y arreglos que contienen datasets completos e internamente desarrollan la semántica de mapeos y agregación estadística necesaria para producir gráficas informativas.

#### 1.2. A la Actividad

En el siguiente documento se seguirá practicando con la biblioteca Matplotlib y ahora con Seaborn de Python para seguir analizando datos, se mostrarán en Heatmaps y gráficas de dispersión. Esta vez estaremos analizando datos de una estación de meteorología ubicada en un campo de nogal con datos del año 2009 y parte del 2010.

### 2. Análisis de Datos

Primero tenemos que cargar a nuestro código las librerías que vamos a necesitar durante la actividad.

```
# Importamos las librerías necesarias para nuestro código
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Leemos el archivo de manera un poco diferente a la que hemos estado usando:

```
# Leemos el archivo "Nogal.csv".
df = pd.DataFrame( pd.read_csv("Nogal.csv", engine="python" ) )
df.head()
```

Eliminamos las columnas con datos nulos, para mejor manejo del DataFrame.

```
# Tomamos las columnas relevantes.
# Eliminamos las columnas innecesarias.
df.drop( df.columns[18:36], axis=1, inplace=True )
df.drop( df.columns[2:4], axis=1, inplace=True )
df.head()
```

Creamos una nueva columna FECHA a partir de las columnas de DATE y TIME:

```
# Creamos una nueva columna para la Fecha
df["Fecha"] = df["DATE"] + " " + df["TIME"]
df.drop( ["DATE","TIME"], axis=1, inplace=True )
df.head()
```

Cambiamos el tipo de datos a float para poder manejar los números y operar con ellos.

```
# Convertimos las variables de object a float64.
df[df.columns[0:14]] = df[df.columns[0:14]].apply(pd.to_numeric, errors='coerce')
```

Usamos la función de `corr()` para encontrar la correlación entre variables en una matriz:

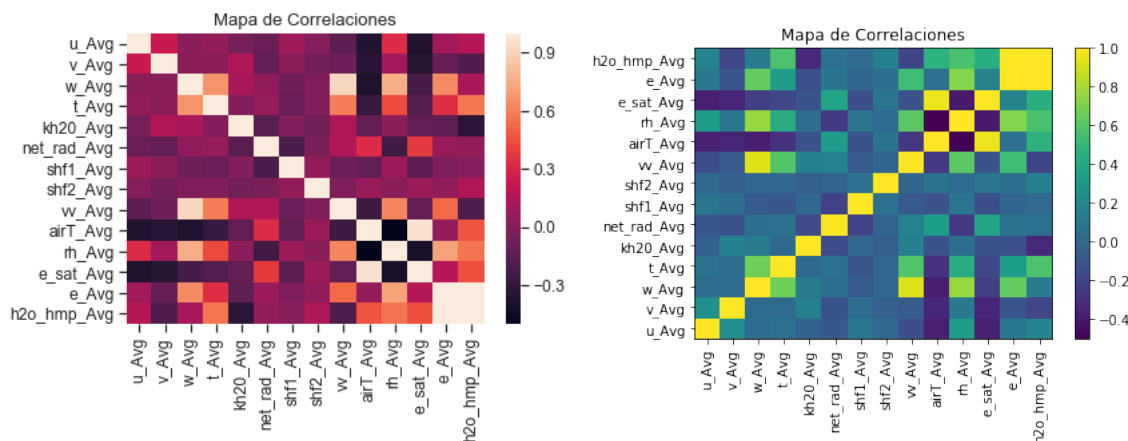
```
# Utilizamos la función corr() para encontrar correlaciones entre las variables.
df1 = df.corr(method='pearson', min_periods=1)
df1.head(14)
```

Generamos dos Heatmaps con los resultados de las correlaciones para comparando el de Matplotlib y el de Seaborn.

También nos fijamos en los datos donde la correlación es mayor a 0.6 y generamos las gráficas de dispersión para estas.

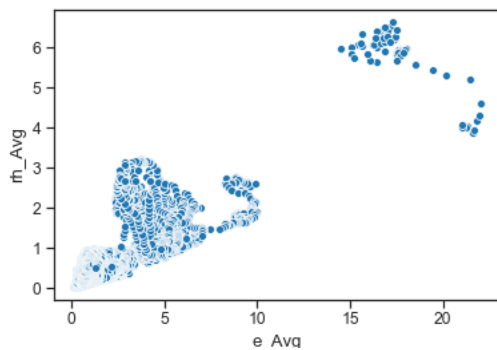
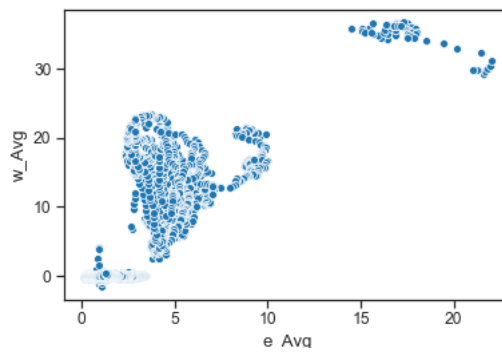
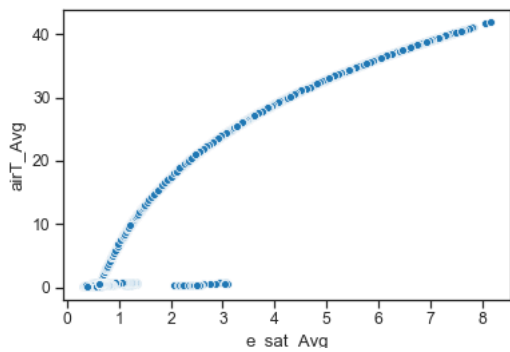
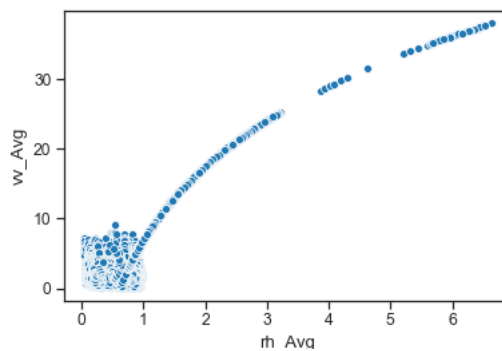
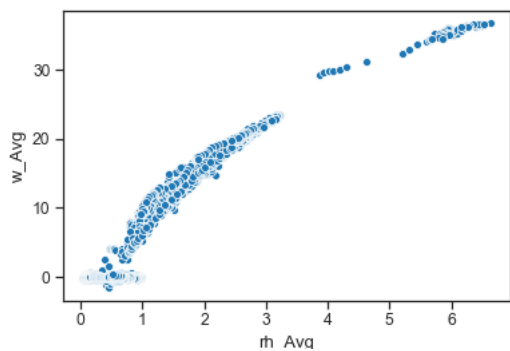
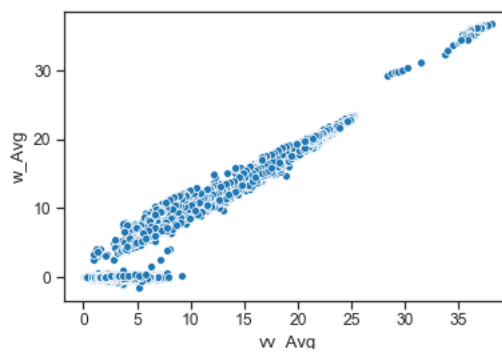
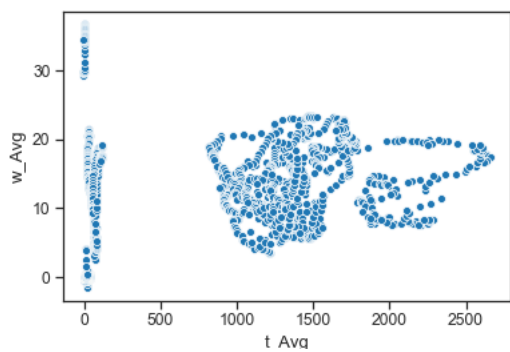
### 3. Resultados

En las siguientes figuras se muestran los heatmap generados con Seaborn y Matplotlib en ese orden:

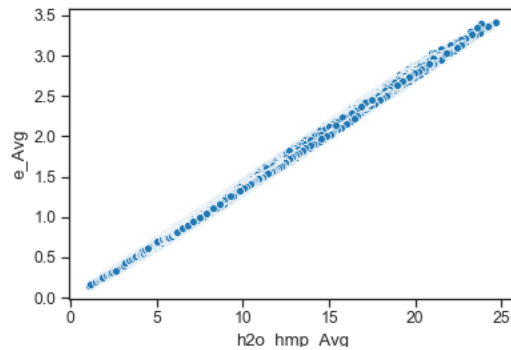


La primera diferencia que vemos es que necesitamos poner manualmente los nombres al eje X y eje Y, también se puede poner manualmente la barra de colores que nos indica el valor de cada color aproximado.

Se obtuvieron muchas gráficas donde la correlación es mayor a 0.6, en seguida se muestran las gráficas generadas.



En la siguiente, la correlación es muy cercana a 1 por lo tanto la gráfica será aproximadamente lineal.



## 4. Conclusión

En Seaborn las gráficas se vuelven una tarea mucho más sencilla que con matplotlib, ya trae implícita algunas funciones que en cambio con matplotlib lo tienes que introducir manualmente lo que nos quita tiempo y además hace más lento el proceso de la lectura del código y generación de gráficas.

En cuanto a las gráficas que se generaron en esta actividad, la correlación nos muestra que tan parecidos o relacionados están ciertos valores, en este caso encontramos que sólo una gráfica tenía una correlación lineal.

## 5. Bibliografía

- Seaborn: statistical data visualization. (2018). Consultado el 20 de Marzo del 2019, de Seaborn. Sitio web:  
<https://seaborn.pydata.org/>
- Probabilidad y Estadística con Python (2015). Consultado el 24 de Marzo del 2019, de Raul E. Lopez Briega Matemáticas, análisis de datos y python. Sitio web:  
<https://relopezbriega.github.io/blog/2015/06/27/probabilidad-y-estadistica-con-python/>