UNIVERSIDAD DE SONORA
Licenciatura en Física para la clase de Física Computacional
17 de Marzo de 2019
Grupo 3

Visualizacion de Datos con la Biblioteca Seaborn

Castillo Espinoza Aarón Sebastián

Introducción

Seaborn es una biblioteca para hacer gráficos estadísticos en Python. Está construido sobre matplotlib y está estrechamente integrado con las estructuras de datos de pandas.

Estas son algunas de las funcionalidades que ofrece Seaborn:

- Una API orientada a conjuntos de datos para examinar relaciones entre múltiples variables.
- Soporte especializado para el uso de variables categóricas para mostrar observaciones o estadísticas agregadas.
- Opciones para visualizar distribuciones univariadas o bivariadas y para compararlas entre subconjuntos de datos.
- Estimación automática y trazado de modelos de regresión lineal para diferentes tipos de variables dependientes.
- Vistas convenientes sobre la estructura general de conjuntos de datos complejos.
- Abstracciones de alto nivel para estructurar cuadrículas de múltiples parcelas que le permiten crear visualizaciones complejas con facilidad.
- Control conciso sobre el estilo de figura de matplotlib con varios temas incorporados.
- Herramientas para elegir paletas de colores que revelen fielmente patrones en sus datos

Seaborn tiene como objetivo hacer de la visualización una parte central de la exploración y comprensión de los datos. Sus funciones de trazado orientadas a los conjuntos de datos operan en marcos de datos y matrices que contienen conjuntos de datos completos y realizan internamente el mapeo semántico y la agregación estadística necesarios para producir gráficos informativos.

Actividades desarrolladas

Para empezar esta actividad, a diferencia de las anteriores, debo importar la biblioteca de Seaborn a la que llamé "sns", junto con las bibliotecas que ya he utilizado antes, pandas, numpy y matplotlib. Posteriormente, le doy lectura al documento de Excel llamado "meteonogal-09.csv" y lo guardo en un DataFrame. Notó que en este documento existen columnas sin nombre que son innecesarias para mi trabajo por lo que, utilizando el comando $df.drop(df.columns[:])$, me deshago es ellas. De manera similar me deshago del primer renglón del DataFrame pues en este se encuentran las unidades de cada columna.

Una vez hecho esto, tomo las columnas llamas "DATE" y "TIME" y las junto en una sola columna llamada "Fecha", esto con el propósito de poder eliminar todos aquellos datos que no fueran del año 2009.

Posteriormente tome el DataFrame y le aplique la función $df.corr()$ la cual me creo una matriz de correlación entre las columnas de este. Esa matriz quedó de la siguiente manera:

	u_Avg	v_Avg	w_Avg	t_Avg	kh20_Avg	net_rad_Avg	shf1_Avg	shf2_Avg	vv_Avg	airT_Avg	rh_Avg	e_sat_Avg	e_Avg	h2o_hmp_Avg
u_Avg	1.000000	0.241494	0.015906	0.044331	-0.049445	-0.094394	0.089411	0.003542	-0.163176	-0.363548	0.331126	-0.370644	0.103343	0.173845
v_Avg	0.241494	1.000000	0.009572	0.011734	0.155188	-0.117176	0.029321	-0.053846	-0.088159	-0.344386	0.098515	-0.347170	-0.109168	-0.189840
w_Avg	0.015906	0.009572	1.000000	0.601335	0.128426	0.133556	-0.079466	-0.010182	0.912885	-0.338470	0.737544	-0.192107	0.646316	0.125133
t_Avg	0.044331	0.011734	0.601335	1.000000	-0.009929	0.266318	-0.094414	-0.006335	0.529160	-0.231403	0.367252	-0.139445	0.306703	0.576896
kh20_Avg	-0.049445	0.155188	0.128426	-0.009929	1.000000	-0.157759	-0.005268	-0.061590	0.152027	-0.122687	0.024498	-0.123180	-0.126972	-0.326760
net_rad_Avg	-0.094394	-0.117176	0.133556	0.266318	-0.157759	1.000000	-0.206065	-0.027206	0.162470	0.334375	-0.266374	0.385945	0.074294	0.056376
shf1_Avg	0.089411	0.029321	-0.079466	-0.094414	-0.005268	-0.206065	1.000000	0.059003	-0.089608	-0.121637	0.084782	-0.138901	-0.016448	0.014207
shf2_Avg	0.003542	-0.053846	-0.010182	-0.006335	-0.061590	-0.027206	0.059003	1.000000	-0.017229	0.078023	0.017651	0.081718	0.051809	0.151206
vv_Avg	-0.163176	-0.088159	0.912885	0.529160	0.152027	0.162470	-0.089608	-0.017229	1.000000	-0.223757	0.582047	-0.091182	0.521232	-0.193064
airT_Avg	-0.363548	-0.344386	-0.338470	-0.231403	-0.122687	0.334375	-0.121637	0.078023	-0.223757	1.000000	-0.484177	0.964464	0.088203	0.464621
rh_Avg	0.331126	0.098515	0.737544	0.367252	0.024498	-0.266374	0.084782	0.017651	0.582047	-0.484177	1.000000	-0.386199	0.728634	0.563712
e_sat_Avg	-0.370644	-0.347170	-0.192107	-0.139445	-0.123180	0.385945	-0.138901	0.081718	-0.091182	0.964464	-0.386199	1.000000	0.198066	0.445780
e_Avg	0.103343	-0.109168	0.646316	0.306703	-0.126972	0.074294	-0.016448	0.051809	0.521232	0.088203	0.728634	0.198066	1.000000	0.999154
h2o_hmp_Avg	0.173845	-0.189840	0.125133	0.576896	-0.326760	0.056376	0.014207	0.151206	-0.193064	0.464621	0.563712	0.445780	0.999154	1.000000

Fig. 1. Matriz de Correlación

Como es evidente, notar si entre las columnas existe algún tipo de relación y en cuales es mayor o menor la relación existente es muy difícil en una tabla. Para hacerlo mas fácil realice las siguientes dos gráficas de correlación, una con la biblioteca de matplotlib y la otra con Seaborn.

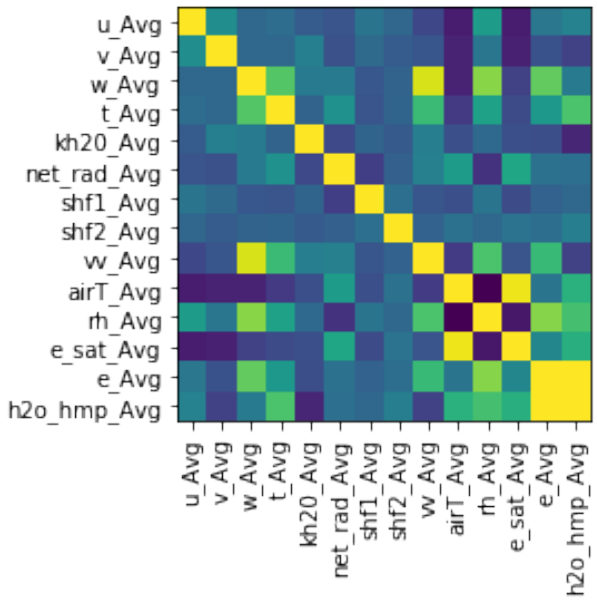

Fig. 2. Gráfica de Correlación por Matplotlib

Los colores en esta gráfica anterior indican que, entre mas amarillo sea mayor es la relación que existe entre los esas dos columnas de datos, entre mas oscuro sea el azul menor es la relación. Cabe mencionar que la diagonal que se nota es debido a que ahí se esta sacando la relación que hay entre una columna consigo misma.

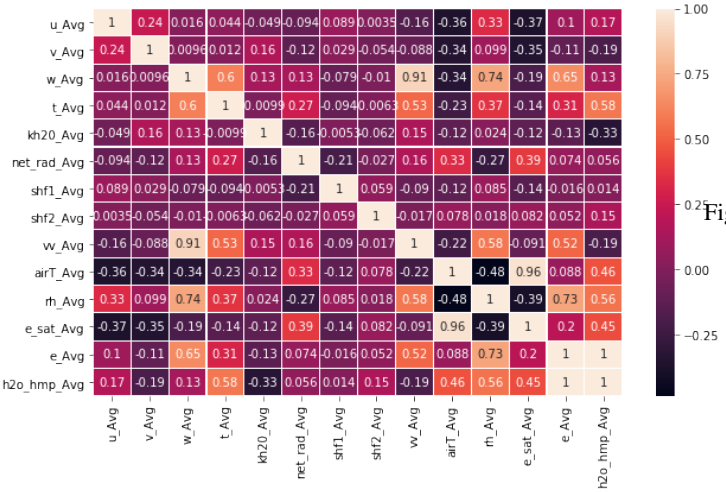

Fig. 3. Gráfica de Correlación por Seaborn

La gráfica anterior esta hecha por la biblioteca de seaborn, como se puede ver es mas fácil notar la relación que existe entre columnas pues aquí es posible ver los números en cada cuadro. Entre mas cercano sea el numero a 1, mayor es la relación existente.

Lo siguiente que se nos pide hacer es tomar aquellas columnas cuya relacion sea mayor o igual a 0.6 y hacer una grafica de dispersion de puntos o "scatter plot" entre ellas usando la biblioteca Seaborn.
Las graficas obtenidas fueron las siguientes:

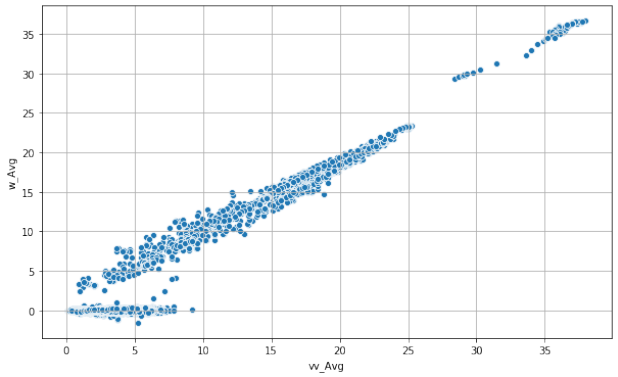

Fig. 4. Grafica de vv_Avg vs. w_Avg

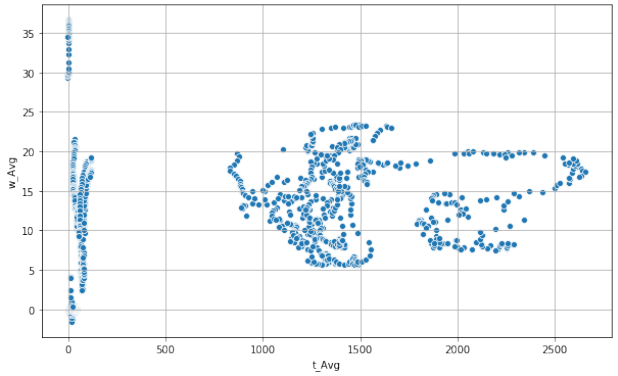

Fig. 5. Gráfica de t_Avg vs. w_Avg

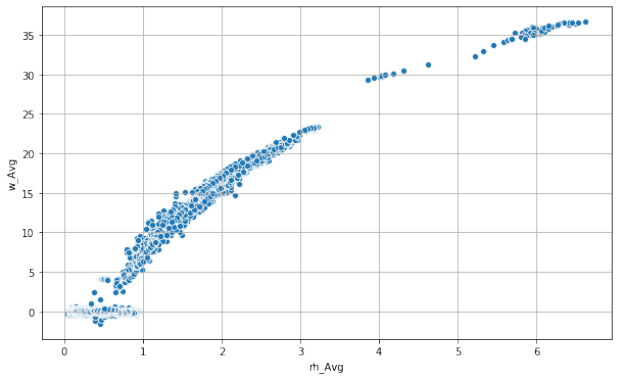

Fig. 6. Gráfica de rh_Avg vs. w_Avg

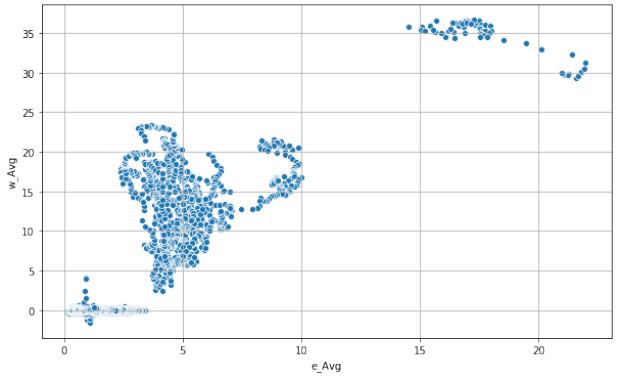

Fig. 7. Gráfica de e_Avg vs. w_Avg

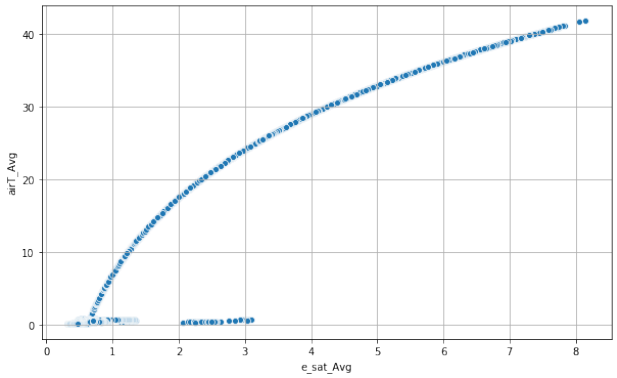

Fig. 8. Gráfica de e_sat_Avg vs. airT_Avg

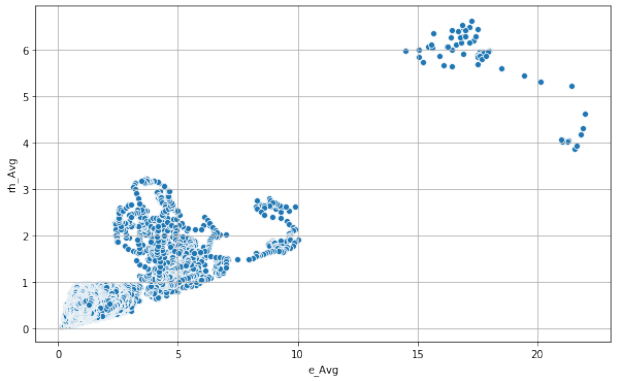

Fig. 9. Gráfica de e_Avg vs. rh_Avg

Como se puede apreciar, entre más cercano sea la correlación a 1, la gráfica de la dispersión tiende a una linea recta.

Conclusión

Al realizar esta practica aprendí una nueva herramienta para el manejo e interpretación de datos, pues muchas veces se nos dan tablas llenas de datos y no sabemos que representan ni de donde salieron pero lo que podemos hacer es obtener la matriz y gráfica de correlación y así sera posible darle un tipo de interpretación a esos datos. También conocí una nueva biblioteca para hacer gráficas en Python con la cual es mas fácil de trabajar pues requiere menos lineas de código y es más versátil.