

Páctica 7

Visualización de datos con la biblioteca Seaborn

Jesús Roberto Araujo Sánchez

15 de marzo, 2019

Abstract

Se tomo un conjunto de datos de una region, se graficaron y se analizo su correlacion para un mejor entendimiento de ellos como magnitudes fisicas interrelacionadas.

1 Introduction

Es muy comun en en analisis de datos que estos vengan sucios, ya sea con vacios y datos desordenados, lo cual es computacionalmente un problema y hay que saber como manejarlo.

Por otro lado, cuando se estudian datos estos son relacion entre magnitudes y dichas magnitudes presentan una correlacion que puede ser directa o inversa y la calidad de dicha correlacion se pondera con valores de -1 a 1, en este caso las bibliotecas utilizadas para las graficas seran Matplotlib y Seaborn.

2 Desarrollo

Para comenzar, cargamos las librerias y leimos el archivo de datos dado en la actividad.

Puesto que los datos presentaban caracteres especiales que Phyton no podia reconocer se utilizo el siguiente comando:

```
Df=pd.DataFrame(pd.read_csv("meteo - nogal - 09.csv", engine = "Phyton"))
```

En esta ocasion se utilizo el comando Engyne, pues a diferencia de las otras actividades esta presenta datos con caracteres especiales que Phyton no reconoce.

Una vez leído el DataFrame se eliminaron algunas columnas que no eran necesarias para nuestro analisis utilizando los siguientes comandos:

```
df.drop(df.columns[18:36], axis=1, Inplace=True)  
df.drop(df.columns[2:4], axis=1, Inplace=True)
```

```
df.head()
```

Ahora para finalizar se quiere encontrar la relacion entre nuestras variables y para eso se usa la funcion `corr()` y le asignamos un dataframe como a continuacion se muestra:

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

La funcion `method` sirve para encontrar la correlacion entre las varibales y `min-periods` el valor entre los que oscila la relacion.

Por ultimo se grafico la correlacion usando las librerias Matplotlib y Seaborn

En Seaborn:

```
sns_plot = sns.heatmap(df_corr, cmap="viridis", robust=True, square=True, annot=False)
```

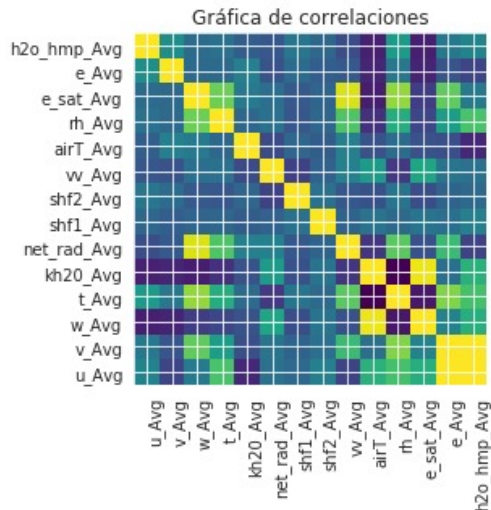
En Matplotlib:

```
fig, ax = plt.subplots()
ax.set_xticks(np.arange(len(df_corr)))
ax.set_yticks(np.arange(len(df_corr)))
ax.set_xticklabels(df_corr)
ax.set_yticklabels(df_corr.columns[::-1])
```

```
plt.setp(ax.get_xticklabels(), rotation=90, ha="right", rotation_mode="anchor")
ax.set_title("Gráfica de correlaciones")
plt.imshow(datos, cmap='viridis', interpolation='nearest')
```

3 Resultados

En base al analisis anterior pudimos generar las siguientes graficas y correlaciones:



Estas graficas muestran la correlacion que hay entre las variables estudiadas.

Para correlaciones con valor absoluto mayor a 0.5 se realizaron las siguientes graficas:

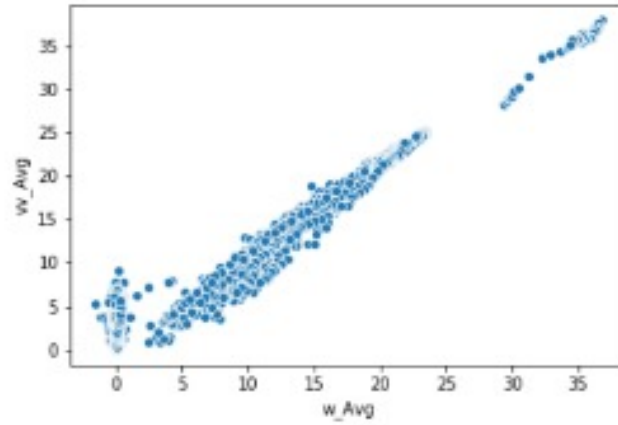


Figura 2: Correlación entre $wAvg$ y $vvAvg$

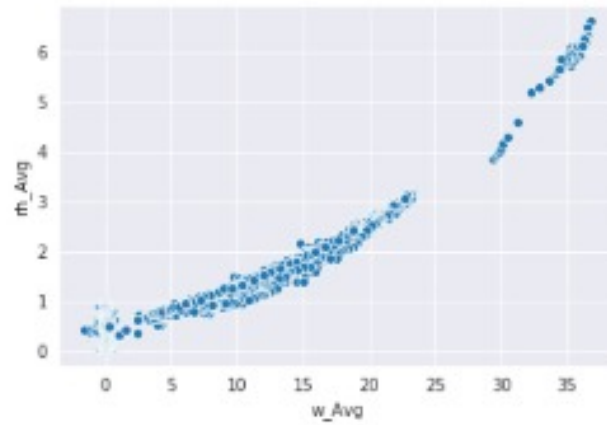


Figura 3: Correlación entre $rhAvg$ y $wAvg$

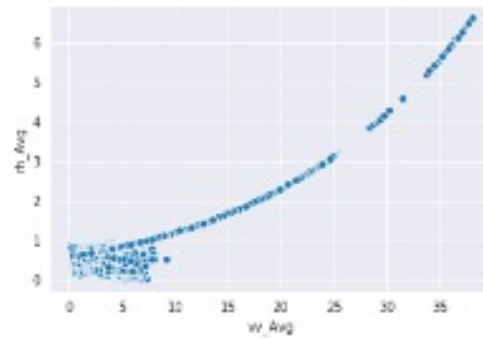


Figura 4: Correlación entre esatAvg y airTAvg

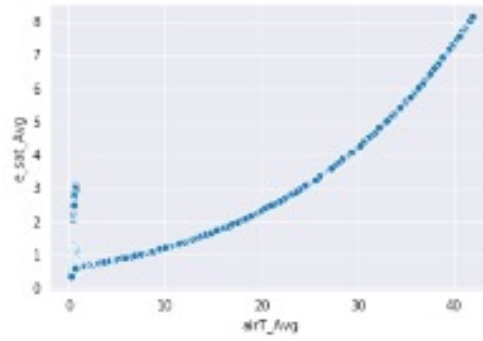


Figura 5: Correlación entre h2ohmpAvg y eAvg

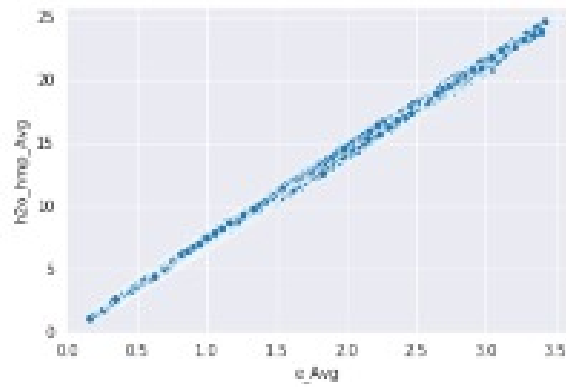


Figura 6: Correlación entre rhAvg y vvAvg

4 Conclusions

Como podemos apreciar mientras mas se aproxima el valor de la correlacion a 1, los datos mas se aproximan a una recta mientras que mientras mas se aleje la correlacion de 1 los datos se curvan.

Por lo tanto la correlacion cercana a uno es una medida de la linealidad de los datos.