

computacional 1

juanjo_lopez1997

November 2018

computacional1, Reporte 2 Juan Jose López November 2018

1 Introduction



Esta actividad, fue el primer código que se escribió formalmente en el curso. Su propósito fue explorar la librería de *Pandas*, que es muy útil para el manejo

de datos en archivos adjuntos, así como *pyplot*, que nos ofrece herramientas para poder representar los datos de manera gráfica. Los datos que utilizamos en esta práctica son directos de la *EMAS: Estaciones Meteorológicas Automaticas*, los descargamos desde su sitio web: <http://smn.cna.gob.mx/es/emas>

2 Etapas de desarrollo

Comencé por escribir las librerías que mencione en la introducción y fui agregando más mientras fui probando el código de algunos sitios web

```
In [ ]: %matplotlib inline
import pandas as pd
import numpy as np
from numpy import ma
import matplotlib.pyplot as plt
import plotly.plotly as py
```

Trabajo de datos

En esta sección de mi código trabajé con la librería *Pandas* para trabajar con los archivos que descargue de la *EMAS: Estaciones Meteorológicas Automaticas*.

En el siguiente renglón cree un *DataFrame*, y di la instrucción de almacenar la información del archivo leído en él.

```
In [10]: df = pd.read_csv("90dias.csv", sep=',', skiprows=9, header=None)
```

En el renglón anterior le ordene que no usara encabezados en la columna, *headers*, y en automático me enumeró las columnas.

```
In [12]: df.head()
```

Out[12]:

	0	1	2	3	4	5	6	7
0	2018-08-30 17:50	193.0	142	50.0	0.0	865.0	35.3	1006.9
1	2018-08-30 17:40	174.0	138	51.0	0.0	859.0	35.0	1006.8
2	2018-08-30 17:30	176.0	141	52.0	0.0	848.0	34.8	1006.9
3	2018-08-30 17:20	180.0	137	52.0	0.0	823.0	34.7	1006.9
4	2018-08-30 17:10	156.0	119	55.0	0.0	810.0	33.6	1006.9

En la siguiente línea de código, nombro las columnas de mi *DataFrame*

```
In [13]: df.columns = ['Fecha', 'DirRaf', 'DirVien', 'Hum', 'Precip', 'Radia', 'Temp', 'Pres']
```

Uso la siguiente instrucción para que me muestre el formato en el cual se guardó la información y como confinación del nombre de las columnas que un renglón atrás cambie.

```
In [15]: df.dtypes
Out[15]: Fecha      object
DirRaf      float64
DirVien      int64
Hum         float64
Precip      float64
Radia       float64
Temp        float64
Pres        float64
dtype: object
```

En los siguientes renglones de código, cambio el formato de la columna *Fecha* de *float64* a *float64*, esto lo hago porque me facilitara unas funciones que vienen a continuación

```
In [16]: df['Fecha'] = pd.to_datetime(df['Fecha'])
```

```
In [17]: df.dtypes
Out[17]: Fecha      datetime64[ns]
DirRaf      float64
DirVien      int64
Hum         float64
Precip      float64
Radia       float64
Temp        float64
Pres        float64
dtype: object
```

En el siguiente renglón de código lo que hago es crear un nuevo *DataFrame* donde me pongo los datos del anterior *DataFrame* pero invertidos en su orden

```
In [18]: reversed_df = df.iloc[::-1]
```

```
In [19]: reversed_df.tail()
```

```
Out[19]:
```

	Fecha	DirRaf	DirVien	Hum	Precip	Radia	Temp	Pres
4	2018-08-30 17:10:00	156.0	119	55.0	0.0	810.0	33.6	1006.9
3	2018-08-30 17:20:00	180.0	137	52.0	0.0	823.0	34.7	1006.9
2	2018-08-30 17:30:00	176.0	141	52.0	0.0	848.0	34.8	1006.9
1	2018-08-30 17:40:00	174.0	138	51.0	0.0	859.0	35.0	1006.8
0	2018-08-30 17:50:00	193.0	142	50.0	0.0	865.0	35.3	1006.9

A continuación, ya con el nuevo *DataFrame* invertido, uso los siguientes renglones de código para crear un nuevo *DataFrame* donde tome los datos que me proporcione el archivo pero seleccionando de acuerdo a la fecha, esto es posible gracias a que cambie el formato de la columna Fecha a *datetime64* anteriormente.

```
In [27]: reversed_df = reversed_df.reset_index(drop=True)
```

```
In [28]: df_10days = reversed_df[(reversed_df['Fecha'] > '2018-08-20 17:50:00') & (reversed_df['Fecha'] <= '2018-08-30 17:50:00')]
```

```
In [29]: df_10days.head()
```

```
Out[29]:
```

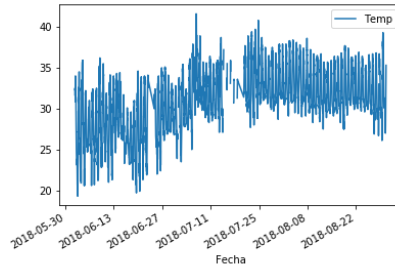
	Fecha	DirRaf	DirVien	Hum	Precip	Radia	Temp	Pres
10354	2018-08-20 19:40:00	165.0	144	58.0	0.0	892.0	35.9	1004.7
10355	2018-08-20 19:50:00	163.0	147	55.0	0.0	917.0	36.3	1004.6
10356	2018-08-20 20:00:00	165.0	148	58.0	0.0	668.0	35.4	1004.5
10357	2018-08-20 20:10:00	166.0	146	60.0	0.0	622.0	35.4	1004.5
10358	2018-08-20 20:20:00	165.0	151	63.0	0.0	532.0	34.5	1004.5

graficas

En las siguientes imágenes, tome los *DataFrame* que hice anteriormente, que contienen la información de un día, un año y diez días, y los grafique usando la *matplotlib.pyplot*

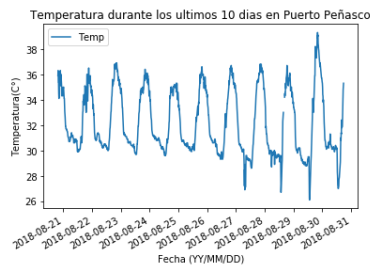
```
In [30]: reversed_df.plot(x="Fecha", y="Temp")
```

```
Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x158353387f0>
```



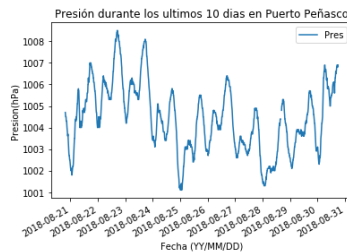
```
In [31]: df_10days.plot(x="Fecha", y="Temp")
plt.title('Temperatura durante los ultimos 10 días en Puerto Peñasco')
plt.xlabel('Fecha (YY/MM/DD)')
plt.ylabel('Temperatura(C°)')
```

```
Out[31]: Text(0,0.5,'Temperatura(C°)')
```



```
In [32]: df_10days.plot(x="Fecha", y="Pres")
plt.title('Presión durante los ultimos 10 días en Puerto Peñasco')
plt.xlabel('Fecha (YY/MM/DD)')
plt.ylabel('Presion(hPa)')
```

```
Out[32]: Text(0,0.5,'Presion(hPa)')
```



3 Conclusion

Pandas me resulto muy útil al momento de trabajar con datos. Presenta un método de lectura muy útil que te permite organizar la información de manera efectiva, aún más que utilizando Excel. La información que se encontró en las gráficas es favorable, se encontró una correlación inversa en los datos de temperatura y presión, cosa que respalda la teoría al respecto.

Ley de Gauss-Lussac

La ley de Gauss-Lussac establece que la presión de un volumen fijo de un gas, es directamente proporcional a su temperatura.

4 Preguntas

¿Cuál es tu impresión de trabajar con Python en Jupyter?

El entorno de trabajo se me hizo amigable, creo que es por su diseño simple que lo organiza por renglones, se siente como si fuera un cuaderno de escuela. Me gusta la facilidad que te da el poder instalar las librerías que ocupas a través de Jupyter.

¿Cuál es tu primera impresión de trabajar con Pandas?

Resulta una herramienta muy útil, pero se me hace algo confuso, porque aun me resulta desconocido.

¿Qué partes se te dificultaron más?

El tener que estar creando *Dataframes* para almacenar la información, por ejemplo: tenía el archivo completo, y para invertir el orden de la información tuve que crear otro *dataframe* donde le ponía la información que quería del otro *dataframe* pero invertida.

¿Qué te pareció esta Actividad? ¿Alcanzó el tiempo?

Estaba relativamente sencilla, lo que más me ocasionó problema fue la ortografía de mi código y una resistencia inicial que tenía a probar código pre-escrito de internet, me ocasionaba desconfianza.

¿Qué le agregarías o quitarías a esta actividad?

Me parece que la actividad está bien en sí. No se me ocurre nada más.

¿Qué cosas te parecieron interesantes y qué se te hizo aburrido?

Escribir el código está entretenido, y estar trabajando con datos reales me ofrece un significado añadido muy personal. Lo aburrido es cuando no resulta tu código y no sabes qué hacer, o no sabes cómo buscarlo en internet, principalmente lo del final, puede ser muy tedioso.

References

- [1] DATOS DE LA EMAS *<http://smn.cna.gob.mx/es/estaciones-meteorologicas-automaticas-2>*.
- [2] TEORIA DE LEY DE GAUSS-LUSSAC *https://es.wikipedia.org/wiki/Ley_de_Gay-Lussac*