

Actividad 3: Biblioteca Pandas de Phyton

César Andrés Pérez Robinson

Marzo 2019

0.1. Introducción

En esta actividad se exploró la biblioteca de software Pandas, desarrollada en el lenguaje Python. Pandas se utiliza principalmente para la manipulación y análisis de datos, por lo que en la presente actividad se hace un análisis de datos presentados por el Servicio Meteorológico Nacional, donde se pueden obtener datos sobre distintos lugares de México.

Se escogieron los datos sobre Medias y Extremas Diarias de Tulum.

Se realizan una variedad de actividades orientadas a responder las siguientes preguntas:

- ¿Cómo le podrás determinar cuáles son los meses más lluviosos?
- ¿Cuáles son los meses más fríos y cuáles son los más cálidos?
- ¿Cuáles han sido años muy húmedos?
- ¿Cuáles han sido años muy secos?
- ¿Cuáles años han tenido inviernos fríos?
- ¿Cuáles años han tenido veranos más cálidos?
- ¿Cómo ha venido siendo la temperatura mensual promedio en los últimos 20 años?
- ¿Qué ha pasado con la precipitación en los últimos 20 años de datos?

0.2. Desarrollo del Código en Jupyter Notebooks

De manera inicial se descargan los programas a usar y el archivo de datos usando:

```
import plotly.plotly as py
import plotly.tools as tls
import matplotlib.pyplot as plt; plt.rcParamsDefaults()
import pandas as pd
import numpy as np

df = pd.read_csv('tulum.csv', sep=',', engine='python')
```

Se asegura que la variable de FECHA se presente como "datetime" utilizando:

```
[IN]
df['Fecha'] = pd.to_datetime(df.apply(lambda x: x['FECHA'], 1), dayfirst=True)
df = df.drop(['FECHA'], 1)
df.head()
```

```
[OUT]
PRECIP    EVAP    TMAX TMIN Fecha
0      2.3    5.3    34.5 21.5 1964-07-09
1      0.0   10.2    31.0 24.5 1964-07-10
2      0.0    8.8    30.0 24.5 1964-07-11
3      0.0    9.2    30.0 27.5 1964-07-12
4      0.0    9.3    31.0 27.5 1964-07-13
```

Se utilizan distintas funciones sobre el DataFrame que se está analizando:

Encabezado	df.head()
Final	df.tail()
Tipos de Variables	df.dtypes
Promedio	df.mean()
Desviación estándar	df.std()
Mediana	df.median()
Máximo	df.max()
Mínimo	df.min()

Posteriormente, se comienza a modificar el DataFrame para poder manejar y contestar las preguntas iniciales. Primero se agregan columnas de mes y año y se elimina la columna de 'Fecha'.

```
df['Año']=df['Fecha'].dt.year
df['Mes']=df['Fecha'].dt.month
df = df.drop('Fecha', axis=1)
df.head()
```

Se les asignan nombres a cada meses:

```
ENE=df[df.Mes==1]
FEB=df[df.Mes==2]
MAR=df[df.Mes==3]
ABR=df[df.Mes==4]
MAY=df[df.Mes==5]
JUN=df[df.Mes==6]
JUL=df[df.Mes==7]
AGO=df[df.Mes==8]
SEP=df[df.Mes==9]
OCT=df[df.Mes==10]
NOV=df[df.Mes==11]
DIC=df[df.Mes==12]
```

Se agrupa la precipitación acumulada para cada mes y presenta la gráfica del comportamiento en la Figura 1.

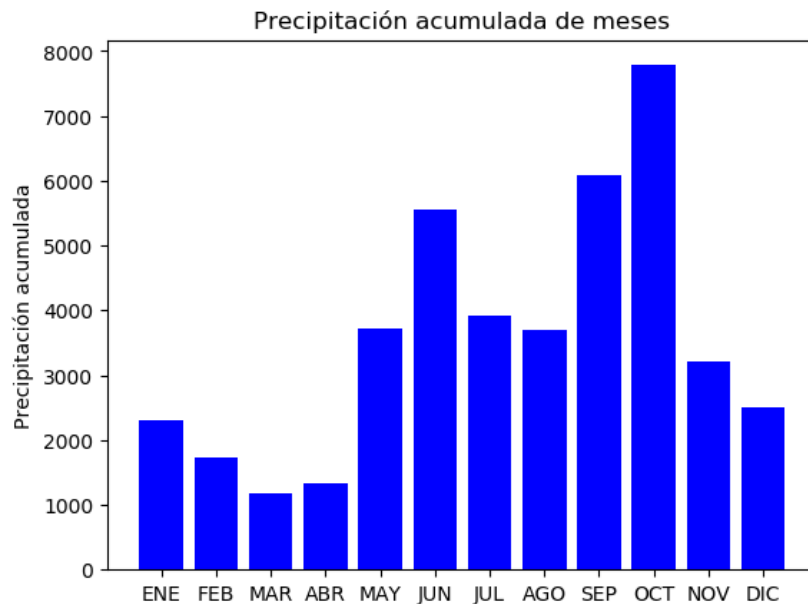
```
acENE = ENE.PRECIP.sum()
acFEB = FEB.PRECIP.sum()
acMAR = MAR.PRECIP.sum()
acABR = ABR.PRECIP.sum()
acMAY = MAY.PRECIP.sum()
acJUN = JUN.PRECIP.sum()
acJUL = JUL.PRECIP.sum()
acAGO = AGO.PRECIP.sum()
acSEP = SEP.PRECIP.sum()
acOCT = OCT.PRECIP.sum()
acNOV = NOV.PRECIP.sum()
acDIC = DIC.PRECIP.sum()

meses = ('ENE', 'FEB', 'MAR', 'ABR', 'MAY', 'JUN', 'JUL', 'AGO',
        'SEP', 'OCT', 'NOV', 'DIC')
nombres = np.arange(len(meses))
precipitacion = [acENE,acFEB,acMAR,acABR,acMAY,acJUN,acJUL,acAGO,
                 acSEP,acOCT,acNOV,acDIC]

plt.bar(nombres, precipitacion, align='center', color='blue')
plt.xticks(nombres, meses)
```

```
plt.ylabel('Precipitación acumulada')
plt.title('Precipitación acumulada de meses')
plt.show()
```

Figura 1: Precipitación mensual acumulada



Después se realiza algo similar para encontrar la temperatura mínima promedio sobre cada mes y se presenta la Figura 2:

```
tmENE = ENE.TMIN.mean()
tmFEB = FEB.TMIN.mean()
tmMAR = MAR.TMIN.mean()
tmABR = ABR.TMIN.mean()
tmMAY = MAY.TMIN.mean()
tmJUN = JUN.TMIN.mean()
tmJUL = JUL.TMIN.mean()
tmAGO = AGO.TMIN.mean()
tmSEP = SEP.TMIN.mean()
tmOCT = OCT.TMIN.mean()
tmNOV = NOV.TMIN.mean()
tmDIC = DIC.TMIN.mean()

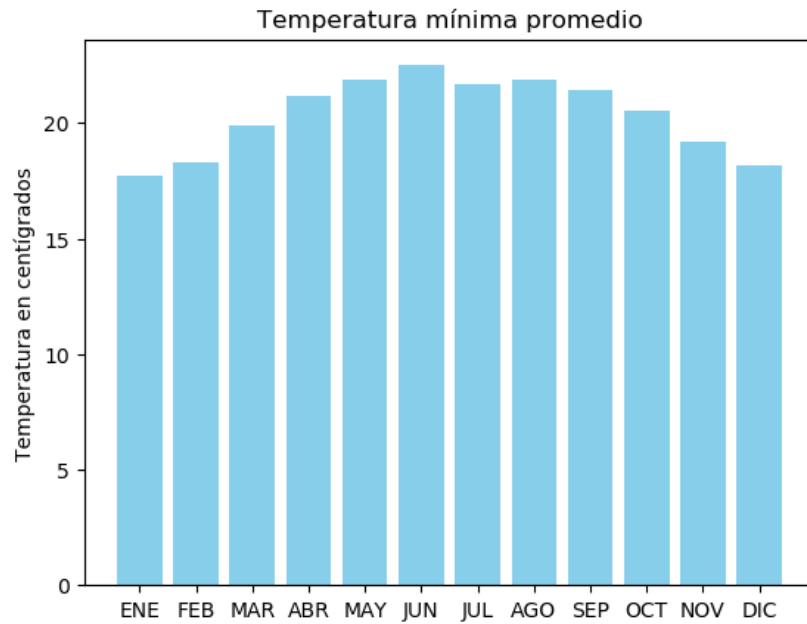
meses = ('ENE', 'FEB', 'MAR', 'ABR', 'MAY', 'JUN', 'JUL', 'AGO',
        'SEP', 'OCT', 'NOV', 'DIC')
nombres = np.arange(len(meses))
tMinprom = [tmENE, tmFEB, tmMAR, tmABR, tmMAY, tmJUN, tmJUL, tmAGO,
            tmSEP, tmOCT, tmNOV, tmDIC]

plt.bar(nombres, tMinprom, align='center', color='skyblue')
plt.xticks(nombres, meses)
plt.ylabel('Temperatura en centígrados')
plt.title('Temperatura mínima promedio')
plt.show()
```

Mientras que para la temperatura máxima de la Figura 3, se utiliza:

```
tmENE = ENE.TMAX.mean()
```

Figura 2: Temperatura mínima



```
tmFEB = FEB.TMAX.mean()
tmMAR = MAR.TMAX.mean()
tmABR = ABR.TMAX.mean()
tmMAY = MAY.TMAX.mean()
tmJUN = JUN.TMAX.mean()
tmJUL = JUL.TMAX.mean()
tmAGO = AGO.TMAX.mean()
tmSEP = SEP.TMAX.mean()
tmOCT = OCT.TMAX.mean()
tmNOV = NOV.TMAX.mean()
tmDIC = DIC.TMAX.mean()

meses = ('ENE', 'FEB', 'MAR', 'ABR', 'MAY', 'JUN', 'JUL', 'AGO',
        'SEP', 'OCT', 'NOV', 'DIC')
nombres = np.arange(len(meses))
tMinprom = [tmENE, tmFEB, tmMAR, tmABR, tmMAY, tmJUN, tmJUL, tmAGO,
            tmSEP, tmOCT, tmNOV, tmDIC]

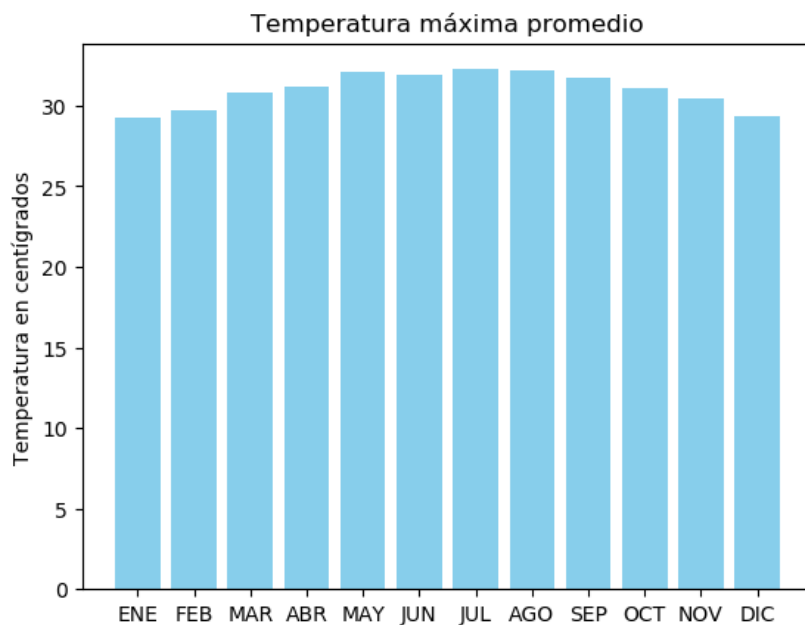
plt.bar(nombres, tMinprom, align='center', color='skyblue')
plt.xticks(nombres, meses)
plt.ylabel('Temperatura en centígrados')
plt.title('Temperatura máxima promedio')
plt.show()
```

Para analizar la precipitación, conviene generar un nuevo DataFrame que solamente contiene los años y la precipitación acumulada para cada año. Esto se hace utilizando:

```
[IN]
años=[]
precip=[]

for i in range(1964,2013):
    a = i
    p = df[df.Año==(i)].PRECIP.sum()
```

Figura 3: Temperatura máxima



```
años.append(a)
precip.append(p)
```

```
new = {'AÑO': años, 'Precipitación_Acumulada': precip}
dfp = pd.DataFrame(data=new)
dfp.tail()
```

```
[OUT]
AÑO Precipitación_Acumulada
44 2008          1127.0
45 2009           566.7
46 2010          1643.5
47 2011           798.4
48 2012          1315.2
```

Se presenta la Figura 4, donde se muestra la precipitación acumulada para cada año:

```
duracion = años
n = np.arange(len(duracion))
PRECIPIT= precip

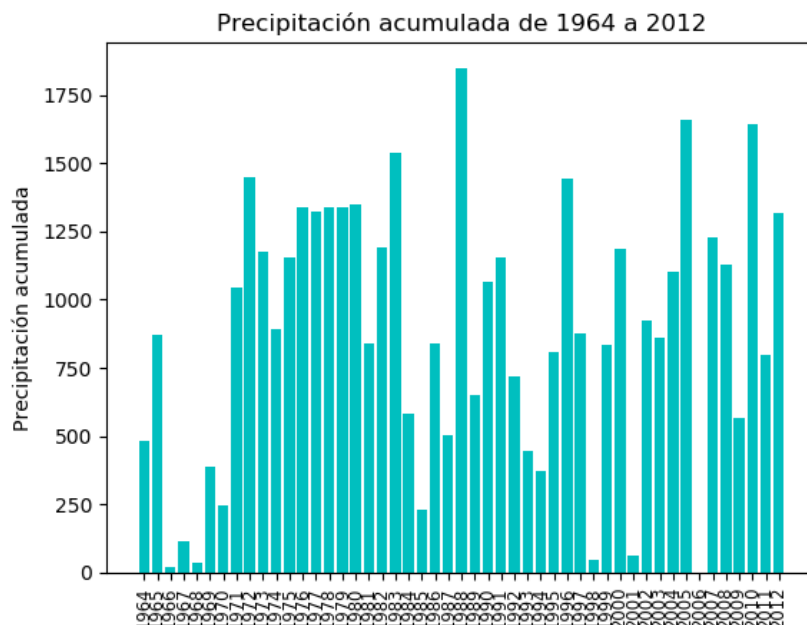
plt.bar(n, PRECIPIT,align='center',color='c')
plt.xticks(n, duracion, size = 'small', color = 'black', rotation = 90)
plt.ylabel('Precipitación acumulada')
plt.title('Precipitación acumulada de 1964 a 2012')
plt.show()
```

Se encuentran los años con mayor y menor precipitación utilizando:

```
[IN]
max=dfp.Precipitación_Acumulada.idxmax()
dfp.iloc[[max]]
```

```
[OUT]
```

Figura 4: Temperatura máxima



```
AÑO Precipitación_Acumulada
24 1988          1849.0
```

```
[IN]
min=dfp.Precipitación_Acumulada.idxmin()
dfp.iloc[[min]]
```

```
[OUT]
AÑO Precipitación_Acumulada
42 2006              0.0
```

Se muestran también los cinco años más y menos lluviosos usando:

```
[IN]
# para los años más lluviosos
dfp.nlargest(5,'Precipitación_Acumulada', keep='first')
```

```
[OUT]
AÑO Precipitación_Acumulada
24 1988          1849.0
41 2005          1657.9
46 2010          1643.5
19 1983          1537.1
8  1972          1446.7
```

```
[IN]
# para los años menos lluviosos
dfp.nsmallest(5,'Precipitación_Acumulada', keep='first')
```

```
[OUT]
AÑO Precipitación_Acumulada
42 2006              0.0
2  1966             19.1
4  1968             36.7
```

```
34 1998          47.0
37 2001          59.5
```

Para identificar los inviernos más fríos se seleccionan los meses de invierno y después se agregan en un nuevo DataFrame:

```
[IN]
dfinv = pd.DataFrame(df.loc[df['Mes'].isin([12,1,2,3])])

tempmax = []
tempmin = []

for i in range(1964,2013):
    ma = dfinv[dfinv.Año==(i)].TMAX.mean()
    mi = dfinv[dfinv.Año==(i)].TMIN.mean()
    tempmax.append(ma)
    tempmin.append(mi)

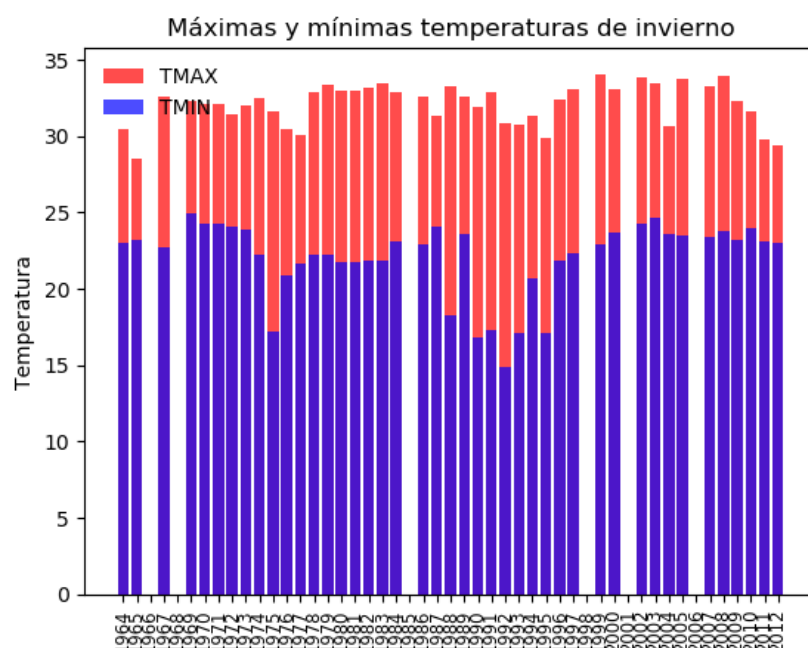
nuevo = {'AÑO': años, 'Temperatura_máxima_promedio': tempmax,
        'Temperatura_mínima_promedio': tempmin}
dft = pd.DataFrame(data=nuevo)
dft.tail()
```

[OUT]

	AÑO	Temperatura_máxima_promedio	Temperatura_mínima_promedio
44	2008	31.009259	19.051402
45	2009	29.389423	18.742718
46	2010	28.672414	16.163793
47	2011	29.023529	18.951807
48	2012	28.009009	19.211712

Se genera la gráfica de temperaturas mínimas y máximas de invierno y se presenta en la Figura 5. Se

Figura 5: Temperatura máxima y mínima de invierno



encuentran los años de temperatura máxima y mínima usando:


```
[IN]
# encontrando el máximo
max=dft.Temperatura_máxima_promedio.idxmax()
dft.iloc[[max]]
[OUT]
AÑO Temperatura_máxima_promedio Temperatura_mínima_promedio
18 1982                31.417355                18.809917
```

```
[IN]
# encontrando el mínimo
min=dft.Temperatura_mínima_promedio.idxmin()
dft.iloc[[min]]
[OUT]
AÑO Temperatura_máxima_promedio Temperatura_mínima_promedio
28 1992                29.819672                13.301639
```

También se buscan los cinco años donde se presentan los inviernos más fríos:

```
# Inviernos más frios, donde la temperatura mínima es menor
dft.nsmallest(5,'Temperatura_mínima_promedio', keep='first')
```

Para encontrar las temperaturas máximas y mínimas para los meses de verano, se repite el procedimiento, solamente que ahora, los meses seleccionados son; junio, julio, agosto y septiembre:

```
dfver = pd.DataFrame(df.loc[df['Mes'].isin([6,7,8,9])])
```

```
tempmax = []
tempmin = []
```

```
for i in range(1964,2013):
    ma = dfver[dfver.Año==(i)].TMAX.mean()
    mi = dfver[dfver.Año==(i)].TMIN.mean()
    tempmax.append(ma)
    tempmin.append(mi)

nuevo = {'AÑO': años, 'Temperatura_máxima_promedio': tempmax,
        'Temperatura_mínima_promedio': tempmin}
dfv = pd.DataFrame(data=nuevo)
```

Y se muestra la gráfica en la Figura 6.

0.3. Conclusiones

Pandas representa una manera rápida de realizar análisis de datos, resulta ser muy eficiente si se conoce el lenguaje y se tiene la habilidad de buscar aquello que no se conoce. La limitante, como todo, es la habilidad de buscar y encontrar aquello que se desea.

Figura 6: Temperatura máxima

