

Actividad5

César Andrés Pérez Robinson

Abril 2019

0.1. Introducción

En esta actividad se trabajó con datos disponibles de temperatura y precipitación de la fuente oficial de datos sobre condiciones de estados de tiempo de México, el Servicio Meteorológico Nacional de la Comisión Nacional del Agua. Calculando una serie de índices que serían indicadores de los efectos del Cambio Climático. Utilizando los datos de las dos actividades anteriores.

Se grafican una variedad de índices, como el número de días con heladas, de verano, de noches tropicales, temperaturas máximas y mínimas, promedio mensual de la diferencia de temperaturas, etc.

0.2. Código en Phyton

Primero se descargan las bibliotecas que se utilizarán:

```
import pandas as pd
import numpy as np
```

Se sube el archivo de datos de la región seleccionada, en este caso es Tulum.

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

Se crea una variable de 'FECHA' que está en formato datetime.

```
df['NFecha'] = pd.to_datetime(df['FECHA'], format='%d/%m/%Y')
```

Se utiliza la variable anterior para generar las columnas de 'AÑO' y 'MES' para agregarse al DataFrame.

```
df['AÑO'] = df['NFecha'].dt.year
df['MES'] = df['NFecha'].dt.month
```

Se requiere identificar cuantos años de datos tenemos disponibles así que se utiliza:

```
NumA = len(df['AÑO'].unique())
```

Y se cuentan cuantas heladas han habido por año en la región con:

```
[IN]
# 1) Número de días con heladas por año (FD: Tmin < 0°C)
df.TMIN.min()
dhelada = df[df.TMIN < 0]
print(dhelada)
```

```
[OUT]
Empty DataFrame
Columns: [FECHA, PRECIP, EVAP, TMAX, TMIN, NFecha, AÑO, MES]
Index: []
```

Lo que indica que no han habido heladas en todos los años de datos.

Después se observa el día con la temperatura más baja. Una manera diferente de hacer esto es utilizar:

```
# Observaremos el día con la temperatura más baja.
df.TMIN.min()
```

```
dhelada = df[df.TMIN < df.TMIN.min() +.01]
print(dhelada)
```

```
[OUT]
   FECHA  PRECIP  EVAP  TMAX  TMIN  NFecha  AÑO  MES
7223  10/02/1991    0.0   4.9  29.3    4.3  1991-02-10  1991    2
```

Para contar los números de días de verano se usa el siguiente *for*:

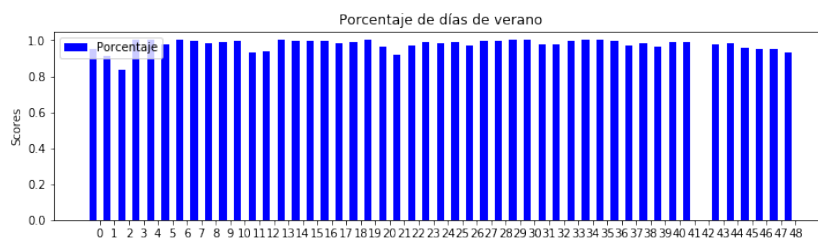
```
# 2) Número de días de verano (SU: Tmax > 25°C)
dverano = df['TMAX'][df.TMAX > 25]
print(dverano.count())
print('')
# Creando arreglos vacios
PctVerano = []
DatosV = []
CantidadV = []
# Para la cantidad de días de verano por año
for i in range(1963,2012):
    NumVerano= dverano[df['AÑO']==[i+1]].count()
    NumDatos= df['TMAX'][df['AÑO']==[i+1]].count()

    DatosV.append(NumDatos)
    CantidadV.append(NumVerano)
    PctVerano.append(NumVerano/NumDatos)

    print(i+1, NumVerano, ";", "Datos totales", NumDatos,
          'Porcentaje Válido', ';', np.round(NumVerano/(NumDatos+0.01),
                                              decimals=2))
```

Donde se muestra el resultado en la Figura 1, que muestra porcentajes altos de días de verano. Para

Figura 1: Porcentaje de días de verano



encontrar el número de noches tropicales por año, donde la temperatura es mayor a 20°C, se usa:

```
# 3) Número de noches tropicales por año (TR: Tmin > 20°C)
dTropicales = df['TMIN'][df.TMIN > 20]
print('Total de días tropicales registrados =', dTropicales.count())
print('')

# Creando arreglos vacios
PctTropical = []
DatosV = []
CantidadV = []

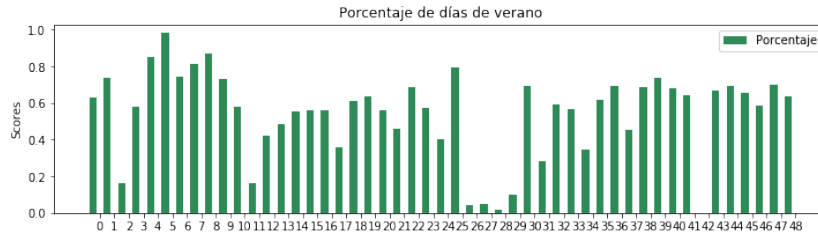
# Para la cantidad de días tropicales por por año
for i in range(1963,2012):
    NumTropical= dTropicales[df['AÑO']==[i+1]].count()
    NumDatos= df['TMIN'][df['AÑO']==[i+1]].count()

    DatosV.append(NumDatos)
    CantidadV.append(NumTropical)
    PctTropical.append(NumTropical/NumDatos)
```

```
print(i+1, NumTropical, "-", "Datos totales", NumDatos, '-', ,
      'Porcentaje Válido', ';',
      np.round(NumTropical/(NumDatos+0.01), decimals=2))
```

Y el resultado se observa en la Figura 2, que muestra cual es el porcentaje días tropicales. Ahora para

Figura 2: Porcentaje de días tropicales



la gráfica de la máxima temperatura mensual de la temperatura máxima se utiliza lo siguiente.

5)

```
tmaxm=[]
inicial=df['AÑO'][0]
nAños=df['AÑO'].nunique()

#El primer loop crea un DataFrame del año
for i in range(0,nAños):
    daño = df[df['AÑO']==inicial]
    #El segundo loop crea un DataFrame por mes
    for j in range (1,13):
        dmes = daño[daño['MES']==j]
        #Se busca la máxima de Tmax
        TmaxMes = dmes.TMAX.max()
        tmaxm.append(TmaxMes)
    inicial=inicial+1
```

Donde de cada año se toma la temperatura máxima de cada mes y eso es lo que se muestra en la Figura 3. Se hace algo similar para la Figura 4, que muestra cómo ha variado la máxima temperatura mensual de la temperatura mínima. Se hace el mismo procedimiento para obtener la Figura 5, que muestra cómo ha variado el mínimo mensua de la temperatura máxima Mientras tanto para el mínimo mensua de la temperatura mínima se utiliza.

```
mmtmin=[]
inicial=df['AÑO'][0]
nAños=df['AÑO'].nunique()

for i in range(0,nAños):
    daño = df[df['AÑO']==inicial]
    for j in range (1,13):
        dmes = daño[daño['MES']==j]
        TminMes = dmes.TMIN.min()
        mmtmin.append(TminMes)

    inicial=inicial+1
```

Para encontrar la diferencia te la temperatura máxima y la mínima, simplemente se tiene que restar la columna de la temperatura mínima a la columna de la temperatura máxima utilizando:

Figura 3: Temperatura máxima mensual

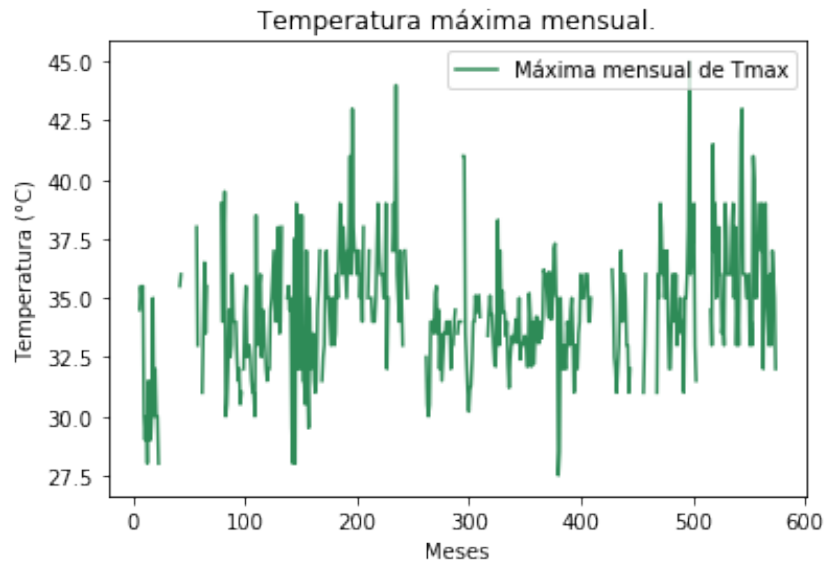
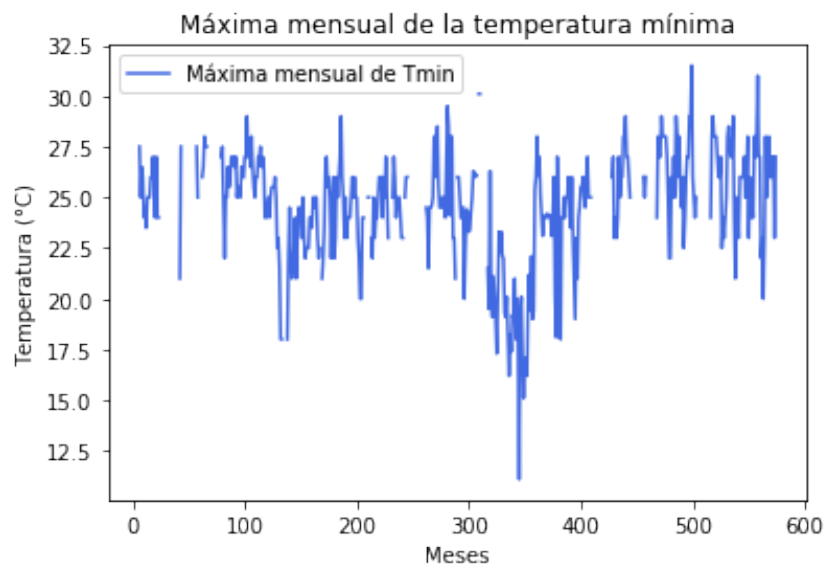


Figura 4: Máxima mensual de temperatura mínima



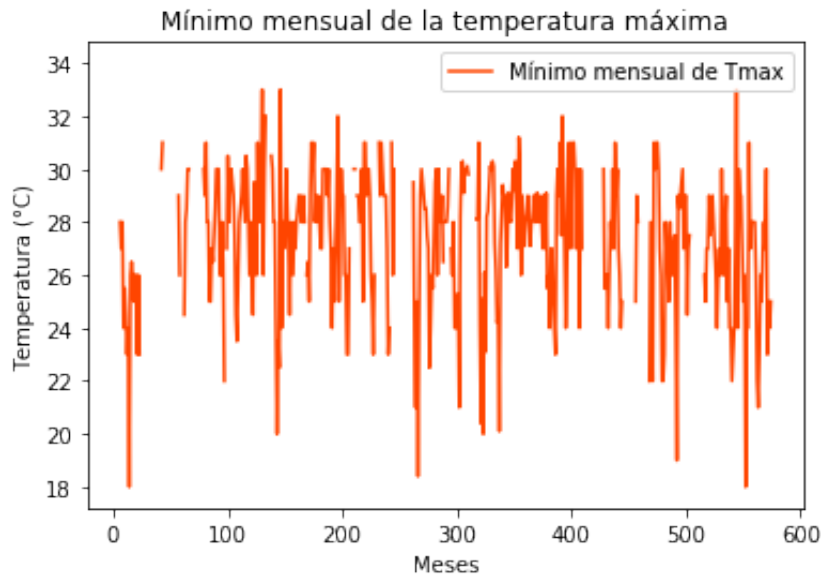
```
df['difTEMP']=df['TMAX']-df['TMIN']
```

Y después para hacer la gráfica de promedio mensual de la diferencia tenemos que hacer una columna con esos datos por lo que podemos utilizar:

```
dift=[]
inicial=df['AÑO'][0]
nAños=df['AÑO'].nunique()

for i in range(0,nAños):
    daño = df[df['AÑO']==inicial]
    for j in range (1,13):
        dmes = daño[daño['MES']==j]
        DifTmean = dmes.difTEMP.mean()
        dift.append(DifTmean)
```

Figura 5: Mínimo mensual de la temperatura máxima



```
inicial=inicial+1
```

```
Y = dift
plt.plot(Y, label = "Promedio mensual de Tmax-Tmin", color = 'black')
plt.xlabel("Meses")
plt.ylabel("Temperatura (C)")
plt.legend()
plt.title('DTR:Promedio mensual de la diferencia de temperaturas (1980-2011).')
plt.savefig('grafica9',plt=2000)
plt.show()
```

Y se muestra el resultado en la Figura 7. Y en cuanto a la precipitación diaria máxima mensual de un día se hace lo siguiente:

```
pdmm1 = []
inicial=df['AÑO'][0]
nAños=df['AÑO'].nunique()

for i in range(0,nAños):
    daño = df[df['AÑO']==inicial]
    for j in range(1,13):
        dmes = daño[daño['MES']==j]
        if(dmes.PRECIP.count()!=0 and dmes.PRECIP.mean()!=0):
            pmaxid = dmes['PRECIP'].idxmax()
            pmax = dmes['PRECIP'][pmaxid]
            pdmm1.append(pmax)
        else:
            pdmm1.append(0.0)

    inicial=inicial+1
```

Que es el mismo procedimiento de las figuras anteriores, y en este caso genera la Figura 8, donde se indica la precipitación diaria máxima mensual.

Figura 6: Mínimo mensual de la temperatura mínima

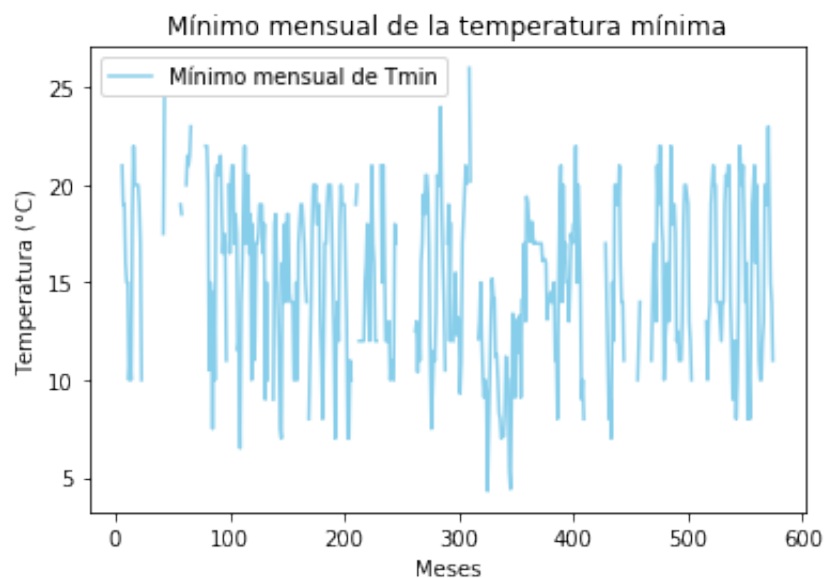


Figura 7: Mínimo mensual de la temperatura mínima

DTR: Promedio mensual de la diferencia de temperaturas (1980-2011).

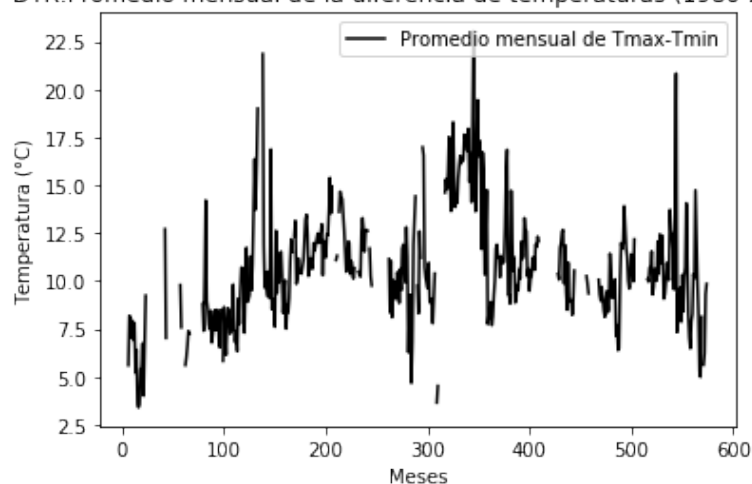


Figura 8: Mínimo mensual de la temperatura mínima

