

# Reporte de la Actividad 5

## Midiendo el Cambio Climático a Nivel Local

Luis Fernando Duarte Gonzalí  
Universidad de Sonora  
Física Computacional

Febrero del 2019

### 1. Introducción

En el siguiente documento se seguirá practicando con la biblioteca Matplotlib de Python para seguir analizando datos, se mostrarán en gráficas de los cálculos obtenidos de los índices del cambio climático. Seguiremos trabajando con el documento de *Leon.txt*, ahora desde el punto de vista del análisis de datos enfocado al cambio climático a nivel local.

El archivo que se utilizó fue descargado de la página del servicio meteorológico de la columna de medias y extremas diarias.

#### 1.1. Climate Changes Indices

Hay un consenso general dentro de la comunidad del clima que cambia en la frecuencia o la severidad de los eventos del clima extremo que tendrían impactos profundos en la naturaleza y la sociedad. Por eso es muy importante analizar eventos extremos. El monitoreo, la detección y la atribución a los cambios extremos en el clima usualmente requieren datos de resolución diaria. De cualquier manera, la compilación, provisión y actualización de datos de resolución diaria globalmente completa y disponible es una tarea muy difícil. Esto es porque, en parte, no todos los servicios meteorológicos e hidrometeorológicos nacionales tienen la capacidad de distribuir libremente los datos diarios que colectan. Consecuentemente, el ET y su predecesor, el CCL/CLIVAR (WG) en la detección del cambio climático han coordinado un esfuerzo internacional para desarrollar, calcular, y analizar un conjunto de índices con los cuales, los países y regiones individualmente, puedan calcular esos índices.

### 2. Análisis de Datos

Los datos que contiene el archivo de *Leon.txt* van desde Enero de 1961 hasta el 2016, no contiene muchas fechas o ninguna sin datos, es decir, no tiene espacios vacíos entre estos años, es por eso que elegí esa estación. Si es que faltaran datos se podrían interpolar los datos faltantes, pero no es muy recomendable ya que es información casi inventada"porque no sabemos qué días sí llovió o no, ya que son procesos casi aleatorios, otra técnica sería cortar los datos y no tener gráficas con espacios vacíos por falta de información.

Para comenzar, necesitamos abrir JUPYTER LAB desde Anaconda Prompt, abriendo la carpeta creada previamente con el nombre de *Actividad5*. Una vez abierto, en python, primero se importaron las bibliotecas Pandas, Numerical Python y Matplot para generar las gráficas.

```
# Importamos las librerías necesarias para nuestro código
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
plt.rcParams.update({'font.size': 13})
```

Para que python pudiera leer los archivos nulos, fue necesario reemplazar la palabra *Nulo* por NaN. Además de cambiar los tipos de datos para poder trabajar con ellos.

```
# Reemplazamos la palabra "Nulo" del archivo de texto por "NaN" para que python pueda leerlo como dato nulo.
df = df.replace('Nulo', 'NaN')

# Cambiamos los tipos de datos para poder empezar a trabajar con ellos de manera correcta y para que python los reconozca.
# Además cambiamos el tipo de dato de la fecha para que lo reconozca como fecha.
df['Fecha'] = pd.to_datetime(df.apply(lambda x: x['Fecha'], 1), dayfirst=True)
df['Precip'] = df['Precip'].astype(float)
df['Evap'] = df['Evap'].astype(float)
df['Tmax'] = df['Tmax'].astype(float)
df['Tmin'] = df['Tmin'].astype(float)
```

### 3. Resultados

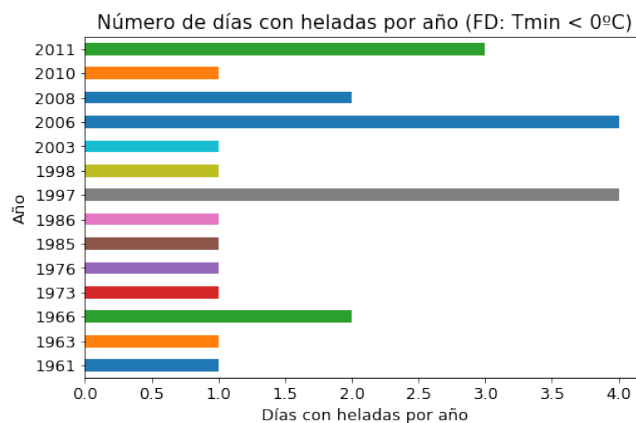
En esta sección se mostrarán las gráficas y el código utilizado para generarlas, analizando los índices del cambio climático con los datos de las dos actividades anteriores

#### 3.1. Número de días con heladas por año.

##### 3.1.1. Código

```
# Número de días con heladas por año (FD: Tmin < 0°C)
fig = plt.figure(figsize=(8,5))
df_FD = df[df.Tmin < 0]
df_FD['Tmin'].groupby(df.Fecha.dt.year).agg('count').plot(kind='barh', legend='')
plt.title('Número de días con heladas por año (FD: Tmin < 0°C)')
plt.ylabel('Año')
plt.xlabel('Días con heladas por año')
```

##### 3.1.2. Gráfica



Se puede observar que los años con mayor número de heladas fueron 1997 y 2006, el último registrado con mayor número de heladas fue en 2011, es decir, desde el 2011 hasta el 2016 no se registraron heladas ( $T_{min} < 0^{\circ}C$ ), mientras que antes de esos años se ha visto una periodicidad de por lo menos cada 2 años ver cierto número de heladas.

## 3.2. Número de Días de Verano por Año

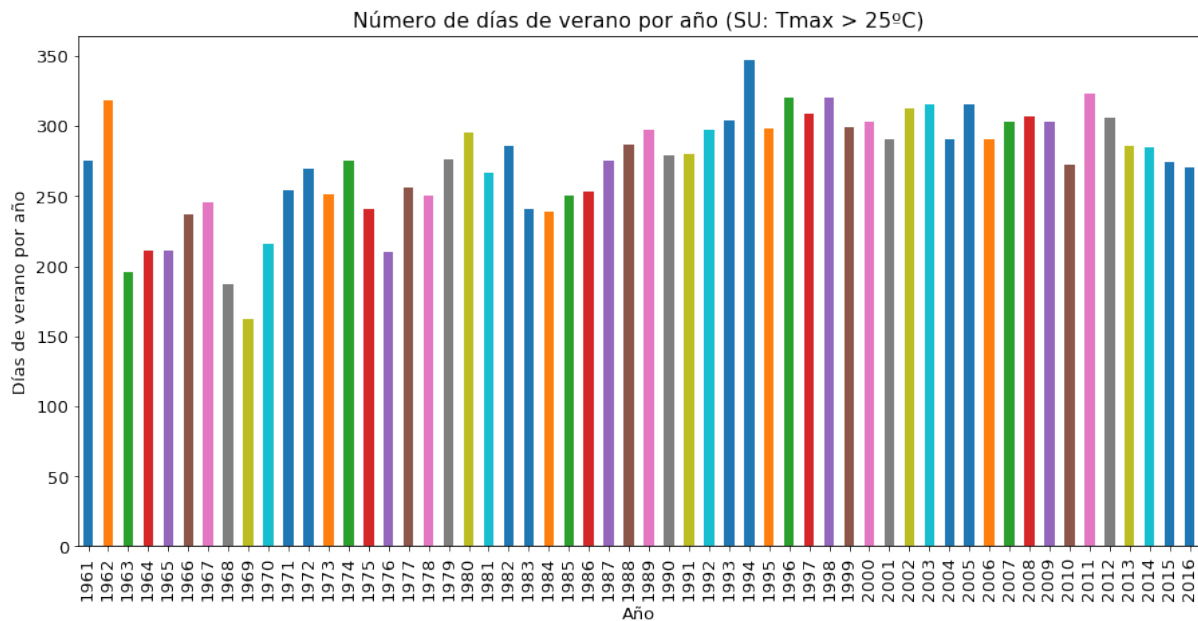
### 3.2.1. Código

```
# Número de días de verano por año (SU: Tmax > 25°C)

df_SU = df[df.Tmax > 25]

fig = plt.figure(figsize=(15,7))
df_SU['Tmax'].groupby(df.Fecha.dt.year).agg('count').plot(kind = 'bar', legend = '')
plt.title('Número de días de verano por año (SU: Tmax > 25°C)')
plt.ylabel('Días de verano por año')
plt.xlabel('Año')
```

### 3.2.2. Gráfica



Como se puede observar en el gráfico anterior, se ve cierta tendencia en las últimas dos décadas donde el número de días de verano ha sido mayor que 260.

## 3.3. Número de Noches Tropicales por Año

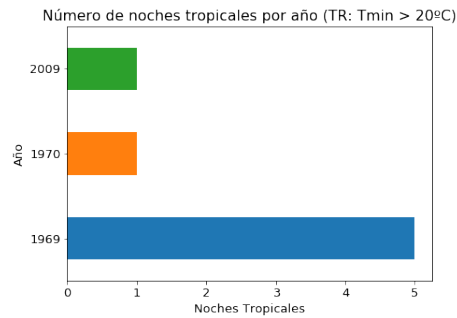
### 3.3.1. Código

```
# Número de noches tropicales por año (TR: Tmin > 20°C)

df_TR = df[df.Tmin > 20]

fig = plt.figure(figsize=(7,5))
df_TR['Tmin'].groupby(df.Fecha.dt.year).agg('count').plot(kind = 'barh', legend = '')
plt.title('Número de noches tropicales por año (TR: Tmin > 20°C)')
plt.ylabel('Año')
plt.xlabel('Noches Tropicales')
```

### 3.3.2. Gráfica



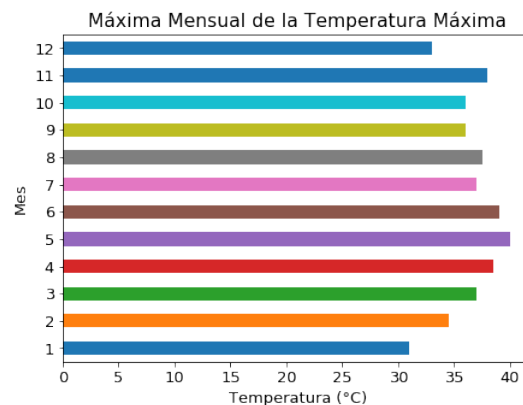
Por otro lado podemos ver que el número de noches tropicales es muy pequeño en el periodo de tiempo (1961-2016), sólo 1969 tuvo 5 noches tropicales ( $T_{min} > 20^{\circ}C$ ) y sólo una noche tropical en 1970 y 2009.

## 3.4. La Máxima Mensual de la Temperatura Máxima

### 3.4.1. Código

```
# TXx: La máxima mensual de la temperatura máxima, max(Tmax).  
  
fig = plt.figure(figsize=(7,5))  
df.groupby(df.Fecha.dt.month)['Tmax'].max().plot(kind = 'barh', legend = '')  
plt.title('Máxima Mensual de la Temperatura Máxima')  
plt.ylabel('Mes')  
plt.xlabel('Temperatura (°C)')
```

### 3.4.2. Gráfica



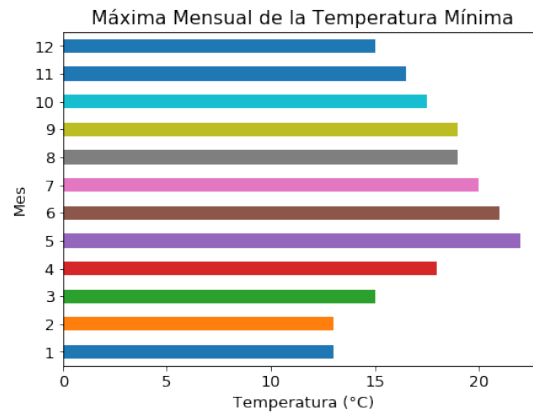
Aquí vemos la máxima mensual histórica en el periodo de tiempo antes mencionado. Donde por obvias razones vemos que el mes con mayor temperatura máxima es en Mayo, pero por falta de conocimiento no me fue posible ubicar el año en que fue esta máxima temperatura máxima.

## 3.5. La Máxima Mensual de la Temperatura Mínima

### 3.5.1. Código

```
# TNx: La máxima mensual de la temperatura mínima, max(Tmin).  
  
fig = plt.figure(figsize=(7,5))  
df.groupby(df.Fecha.dt.month)['Tmin'].max().plot(kind = 'barh', legend = '')  
plt.title('Máxima Mensual de la Temperatura Mínima')  
plt.ylabel('Mes')  
plt.xlabel('Temperatura (°C)')
```

### 3.5.2. Gráfica



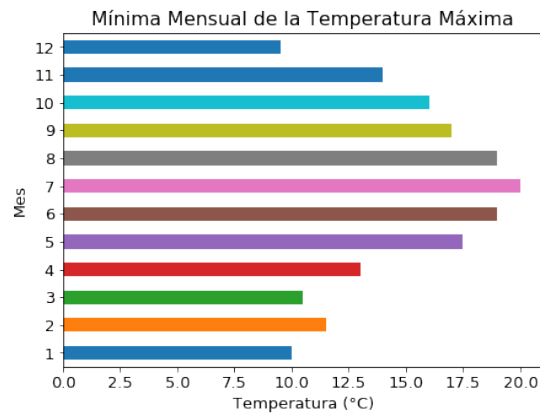
Aquí podemos observar de nuevo que el mes con récord histórico en la máxima temperatura mínima es mayo, igual, no se puede mostrar el año que tuvo el récord.

## 3.6. El Mínimo Mensual de la Temperatura Máxima

### 3.6.1. Código

```
# TXn: El mínimo mensual de la temperatura máxima, min(Tmax).
fig = plt.figure(figsize=(7,5))
df.groupby(df.Fecha.dt.month)['Tmax'].min().plot(kind = 'barh', legend = '')
plt.title('Mínima Mensual de la Temperatura Máxima')
plt.ylabel('Mes')
plt.xlabel('Temperatura (°C)')
```

### 3.6.2. Gráfica



Por otro lado, podemos observar que la mínima temperatura máxima se dió en entre Diciembre y Enero, que es muy lógico.

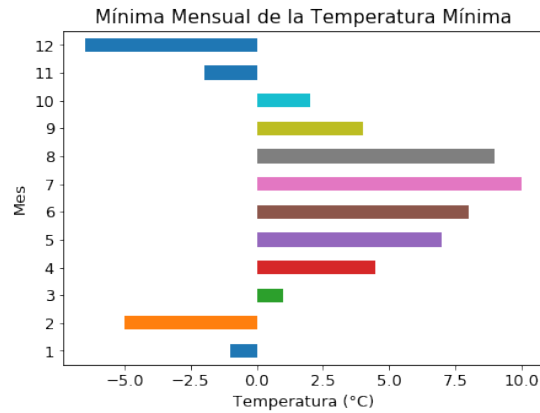
### 3.7. El Mínimo Mensual de la Temperatura Mínima

#### 3.7.1. Código

```
# Tmn: El mínimo mensual de la temperatura mínima, min(Tmin).

fig = plt.figure(figsize=(7,5))
df.groupby(df.Fecha.dt.month)['Tmin'].min().plot(kind = 'barh', legend = '')
plt.title('Mínima Mensual de la Temperatura Mínima')
plt.ylabel('Mes')
plt.xlabel('Temperatura (°C)')
```

#### 3.7.2. Gráfica



Igual, en Diciembre se dio la mínima temperatura mínima, alcanzando los menor que -5.0 °C.

### 3.8. El Promedio Mensual de la Diferencia de Temperaturas

#### 3.8.1. Código

```
# DTR: El promedio mensual de la diferencia de temperaturas (Tmax-Tmin).
df['Tdif'] = df.Tmax - df.Tmin

df.head()
```

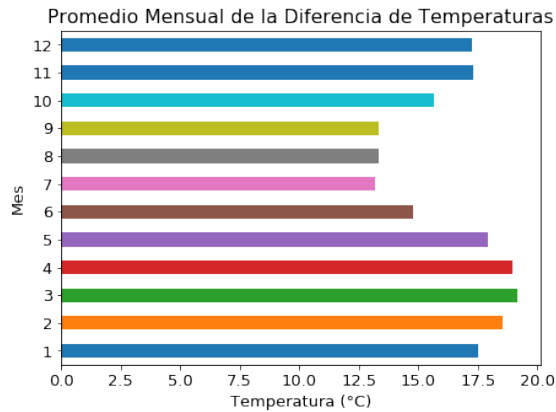
	Fecha	Precip	Evap	Tmax	Tmin	Tsum	Tprom	Mes	Día	Tdif
0	1961-01-01	0.0	3.7	25.0	9.0	34.0	17.00	1	1	16.0
1	1961-01-02	0.0	4.7	25.5	8.5	34.0	17.00	1	2	17.0
2	1961-01-03	0.0	5.4	25.0	6.0	31.0	15.50	1	3	19.0
3	1961-01-04	0.0	4.0	24.0	6.0	30.0	15.00	1	4	18.0
4	1961-01-05	0.0	2.8	24.5	6.0	30.5	15.25	1	5	18.5

```
# DTR: El promedio mensual de la diferencia de temperaturas (Tmax-Tmin).

fig = plt.figure(figsize=(7,5))
df.groupby(df.Fecha.dt.month)['Tdif'].mean().plot(kind = 'barh', legend = '')
plt.title('Promedio Mensual de la Diferencia de Temperaturas')
plt.ylabel('Mes')
plt.xlabel('Temperatura (°C)')
```

En este caso, cree otra columna que nos mostrara la diferencia entre la temperatura mínima y máxima diaria, para poder después hacer el código que nos genere la siguiente gráfica.

### 3.8.2. Gráfica



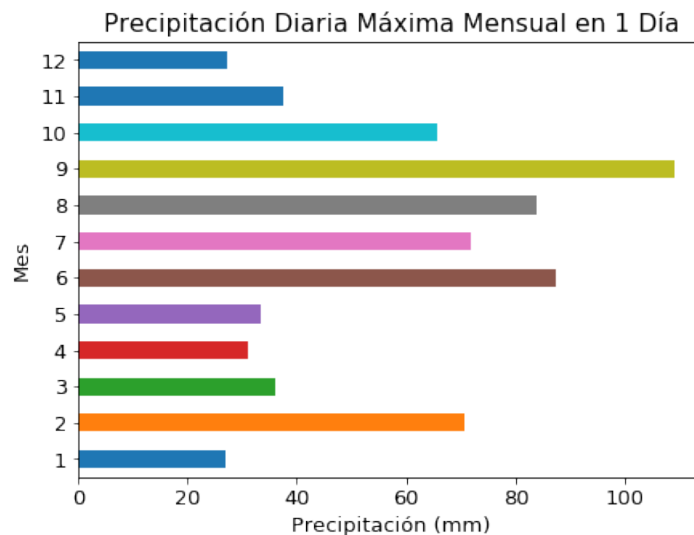
Esta vez se muestra la gráfica de promedio mensual de la columna que se ha creado previamente para la diferencia de temperaturas mínima y máxima, donde vemos que en la ciudad de León el promedio de la diferencia de temperaturas se encuentra por debajo de los 20°C todo el año.

## 3.9. Precipitación Diaria Máxima Mensual en 1 Día.

### 3.9.1. Código

```
# Rx1day: Precipitación diaria máxima mensual en 1 día.
fig = plt.figure(figsize=(7,5))
df.groupby(df.Fecha.dt.month)['Precip'].max().plot(kind = 'barh', legend = '')
plt.title('Precipitación Diaria Máxima Mensual en 1 Día')
plt.ylabel('Mes')
plt.xlabel('Precipitación (mm)')
```

### 3.9.2. Gráfica



Ahora observamos la precipitación máxima en un solo día históricamente, donde vemos que el récord se encuentra en el mes de Septiembre, pero otra vez no pude visualizar el año en que se dio ésta máxima.

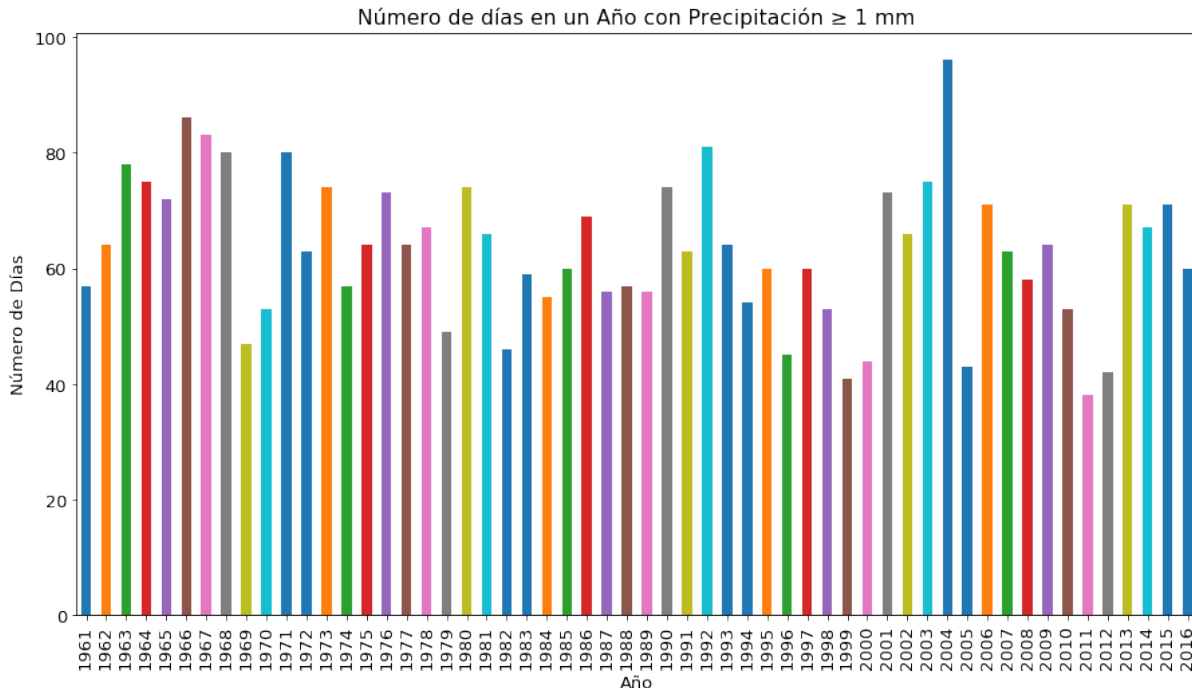
### 3.10. Número de Días en un Año con Precipitación Mayor Igual a 1mm.

#### 3.10.1. Código

```
# Número de días en un año con precipitación mayor igual a 1mm
df_SDII = df[df.Precip >= 1]

fig = plt.figure(figsize=(15,8))
df_SDII['Precip'].groupby(df.Fecha.dt.year).agg('count').plot(kind = 'bar', legend = '')
plt.title('Número de días en un Año con Precipitación ≥ 1 mm')
plt.ylabel('Número de Días')
plt.xlabel('Año')
```

#### 3.10.2. Gráfica



Aquí vemos el número de días donde hubo precipitación mayor igual a un milímetro, donde observamos que la máxima está en el 2004, pero no se ve gran disminución de precipitaciones a años anteriores, tal vez en las últimas dos décadas sí se ha visto un poco, pero necesitaríamos ver de los últimos dos años o más para asegurar que ha habido una disminución significativa.

### 3.11. Número de Días en el Año con Precipitación Diaria Mayor Igual a 10 mm

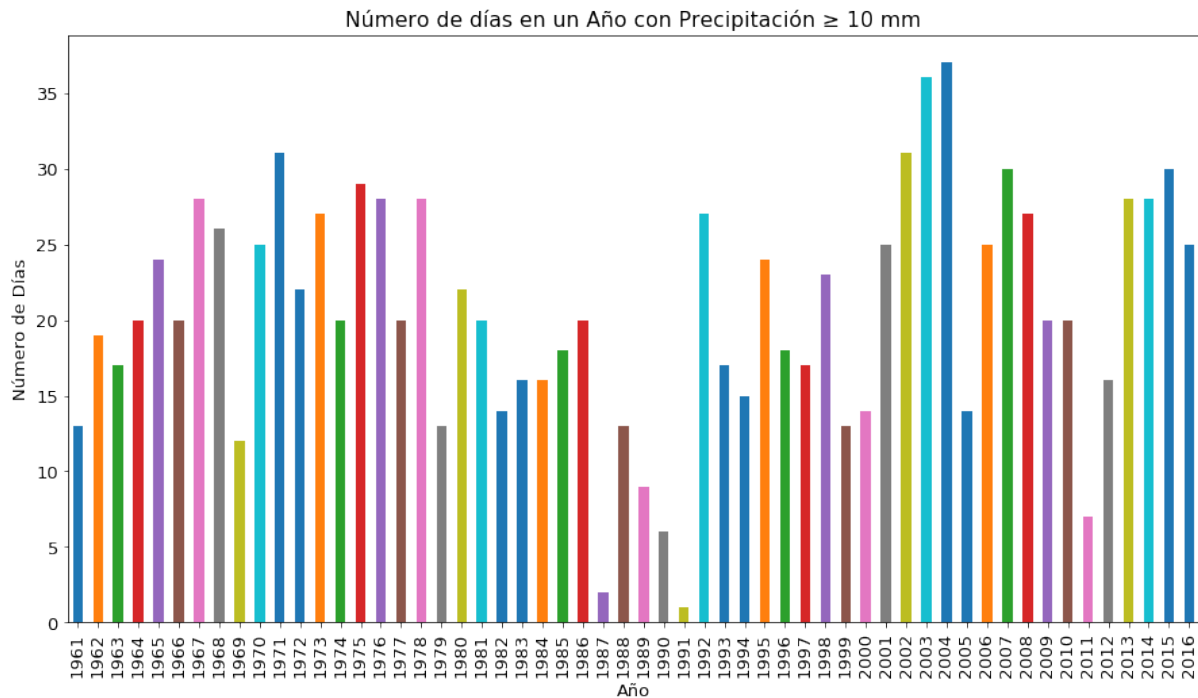
#### 3.11.1. Código

```
# Número de días en el año con precipitación diaria mayor igual a 10 mm.
df_R10 = df[df.Precip >= 10]

fig = plt.figure(figsize=(15,8))
df_R10['Precip'].groupby(df.Fecha.dt.year).agg('count').plot(kind = 'bar', legend = '')
plt.title('Número de días en un Año con Precipitación ≥ 10 mm')
plt.ylabel('Número de Días')
plt.xlabel('Año')
```



### 3.11.2. Gráfica



En la gráfica anterior pudimos observar que la mayoría de los años ha tenido precipitación mayor a 1 mm, pero como hablamos de cantidad, esta vez, se ve reducido el número de días con precipitación mayor igual a 10 mm, donde 1991 ha sido el más bajo hasta ahora, y el mínimo más reciente fue en el 2011.

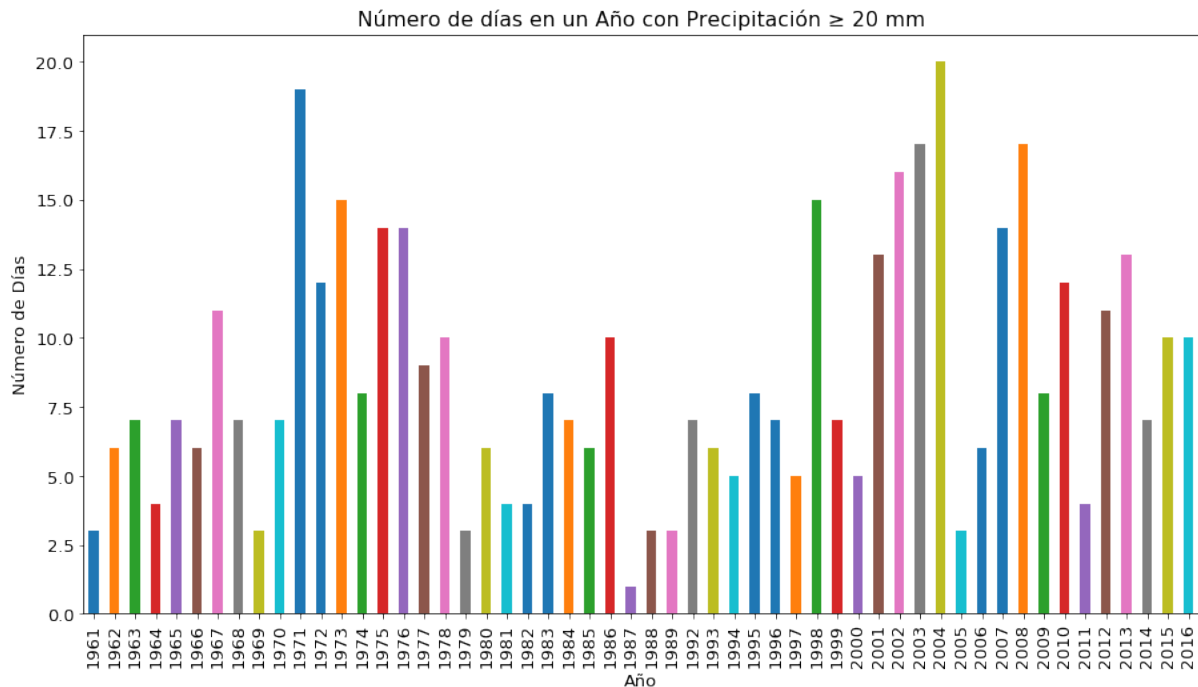
### 3.12. Número de Días en el Año con Precipitación Diaria Mayor Igual a 20mm

#### 3.12.1. Código

```
# Número de días en el año con precipitación diaria mayor igual a 20mm.
df_R20 = df[df.Precip >= 20]

fig = plt.figure(figsize=(15,8))
df_R20['Precip'].groupby(df.Fecha.dt.year).agg('count').plot(kind = 'bar', legend = '')
plt.title('Número de días en un Año con Precipitación  $\geq 20$  mm')
plt.ylabel('Número de Días')
plt.xlabel('Año')
```

### 3.12.2. Gráfica



Ahora, la gráfica se ve muy diferente a la de número de días con precipitación mayor igual a 1 y 10 milímetros. En esta gráfica, ni siquiera aparece el año 1991, y el año 1987 sigue apareciendo, al parecer no hubo tantos días con precipitaciones, pero hubo gran cantidad de milímetros de precipitación.

## 4. Conclusión

Se cumplió con el objetivo principal de la actividad, la cual era seguir conociendo la librería de Matplotlib en Python, ahora obteniendo los índices del cambio climático al nivel local. También alimenta la curiosidad sobre el análisis de datos con Python y seguir buscando, por ejemplo, otros tipos de gráficos y cómo mostrar más información en una sola imagen.

En la ciudad de León no hay cambios tan bruscos o que se puedan notar fácilmente, pienso que se debe hacer un estudio más exhaustivo para encontrar irregularidades, variaciones o anomalías en el clima de León, Guanajuato. Aunque también debe ser un análisis correcto de los datos, ya que por falta de conocimiento y desarrollo en el entorno de las librerías de gráficos y código en python no se hizo un análisis completo.

Se pudieron observar muy pequeñas variaciones en los últimos años en la temperatura y precipitación, pero no fueron fáciles de encontrar. Tampoco se está siguiendo cierto tipo de patrón o algo que haga evidente el cambio climático, por lo menos en la ciudad de León.

## 5. Bibliografía

- Definitions of the 27 core indices. (2009). Consultado el 24 de Febrero del 2019, de Climate Change Indices. Sitio web:  
[http://etccdi.pacificclimate.org/list\\_27\\_indices.shtml](http://etccdi.pacificclimate.org/list_27_indices.shtml)
- Gallery. (2019). Consultado: 15 de Febrero del 2019, de Matplotlib.org. Sitio web:  
<https://matplotlib.org/gallery/index.html>

- Matplotlib Tutorial: Python Plotting. (2017). Consultado: 15 de Febrero del 2019, de DataCamp. Sitio web:  
<https://www.datacamp.com/community/tutorials/matplotlib-tutorial-python>
- What is Matplotlib?. (2017). Consultado: 18 de Febrero del 2019, de edureka!. Sitio web:  
<https://www.edureka.co/blog/what-is-matplotlib/>
- Boxplots. (2019). Consultado: 17 de Febrero del 2019, de Matplotlib.org. Sitio web:  
[https://matplotlib.org/gallery/statistics/boxplot\\_demo.html](https://matplotlib.org/gallery/statistics/boxplot_demo.html)
- Representación gráfica de funciones y datos. (2018). Consultado: 18 de Febrero del 2019, de IAC. Sitio web:  
<http://www.iac.es/sieinvens/python-course/source/matplotlib.html>
- Agrupación de datos por fecha en pandas. (2018). Consultado: 15 de Febrero del 2019, de Analytics Lane. Sitio web:  
<https://www.analyticslane.com/2018/07/06/agrupacion-de-datos-por-fecha-en-pandas/>