

Reporte Seminario de Física Computacional

Itzel Alexia Gacía Monge

6 de diciembre 2018

1 Lectura de Datos

Al descargar los datos que se usarían, se procedió a darles un formato apropiado en *Python*. Primeramente, como el archivo contenía los valores separados de manera no equitativa — había un espacio entre datos, pero tres espacios entre otros — se usó el comando

```
delim_whitespace=True
```

para ayudar con esa inequitatividad al momento de leer los valores.

Una vez que se puedan leer los valores, se procede a cambiar su tipo, ya que *Python* los interpretó como Objeto. Se cambia la fecha a *datetime* y el resto de las columnas a punto flotante:

```
df['Ndate'] = pd.to_datetime(df['Fecha'], dayfirst=True,
format='%d/%m/%Y', errors='coerce')
df[col]= df[col].apply(pd.to_numeric, errors='coerce',
axis=1)
```

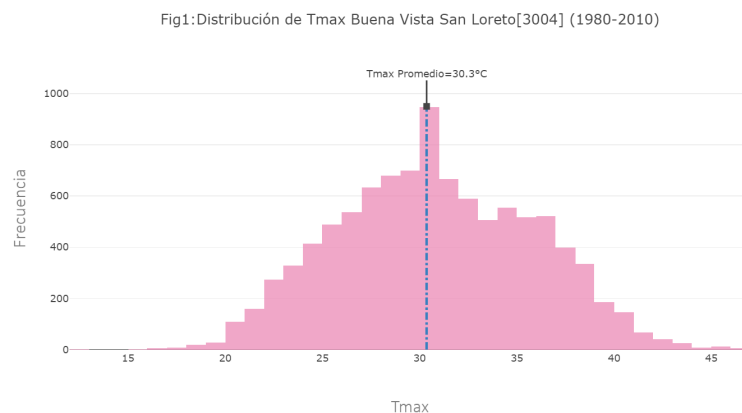
De esta forma, ya se está listo para empezar a manipular los datos del archivo.

Se van a filtrar los años de 1980 a 2010 de la variable principal, creando una nueva, ya que 30 años de datos de clima te da la Normal Climatológica de la ciudad, estado o región.

2 Histogramas y Boxplots

Para graficar los datos, se utilizó la biblioteca *plotly* en línea. Antes de empezar a graficar, se necesita tener un dato de referencia. Este se obtuvo con el promedio de la temperatura máxima, el cuál se representaría en las gráficas por medio de una línea.

La primera gráfica creada fue un histograma que mostraba la distribución de la temperatura máxima contra su frecuencia.



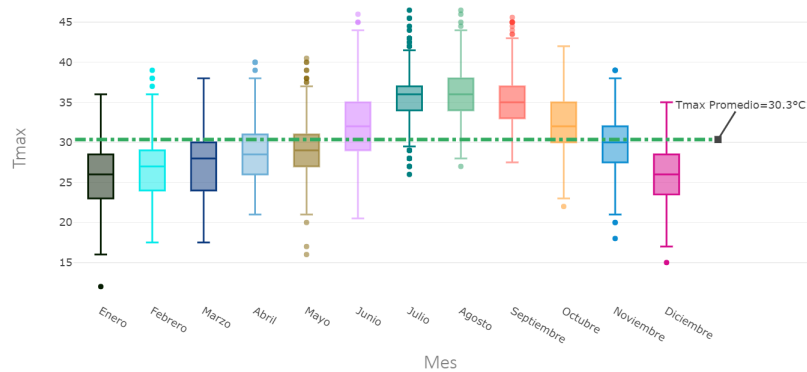
Como también se quiere graficar un boxplot de cada mes, agregamos una nueva etiqueta a nuestro dataframe donde marque del 1 al 12 el mes en que se encuentra. Esto se hace de la forma:

```
df2['month'] = pd.DatetimeIndex(df2['Ndate']).month
```

Con esto, podemos crear un dataframe para cada mes sin importar su año, obteniendo 12 dataframes, es decir, 12 boxplots.

```
y1 = df2.loc[df2['month'] == 1]
.
.
.
y12 = df2.loc[df2['month'] == 12]
```

Fig2: Variabilidad Mensual de Tmax Buena Vista San Loreto[3004] (1980-2010)



Al contar con la Normal Climatológica, ahora podemos comparar nuevos datos con ella y observar si la temperatura ha aumentado. Se creó un nuevo dataframe con los datos que se usaron para comparar con la Normal, siendo estos del 2011 al 2016.

Primeramente, se crearon del nuevo dataframe sus respectivos histogramas y boxplots, repitiendo los mismos pasos que en las gráficas pasadas.

Fig1: Distribución de Tmax Buena Vista San Loreto[3004] (2011-2016)

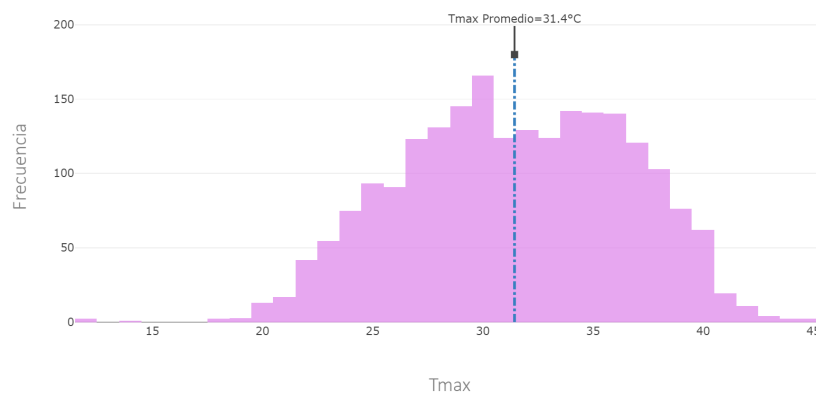
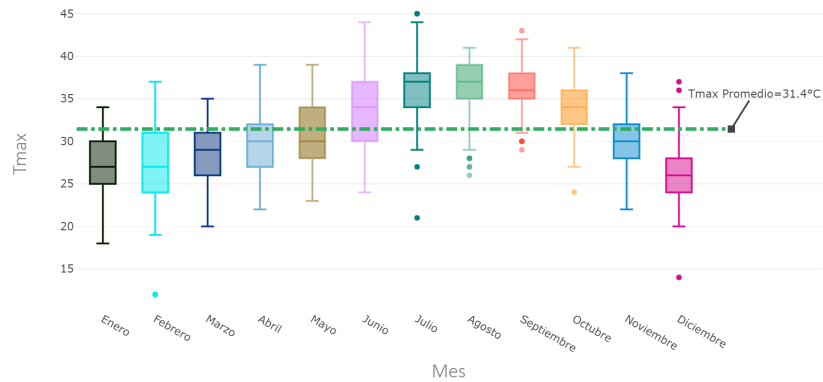


Fig4: Variabilidad Mensual de Tmax Buena Vista San Loreto[3004] (2011-2016)



Posteriormente, se juntan ambos Histogramas y Boxplots para observar cómo ha cambiado la temperatura los últimos 5 años.

Hay que tener en cuenta que como el número de datos de 1981-2010, es mayor al número de datos de 2011-2017 se deben de normalizar las gráficas del Histograma. Además, una vez Normalizadas, también se puede apreciar su contraste en forma porcentual.

Fig5:Distribución de Tmax Buena Vista San Loreto[3004] (Normalizada)

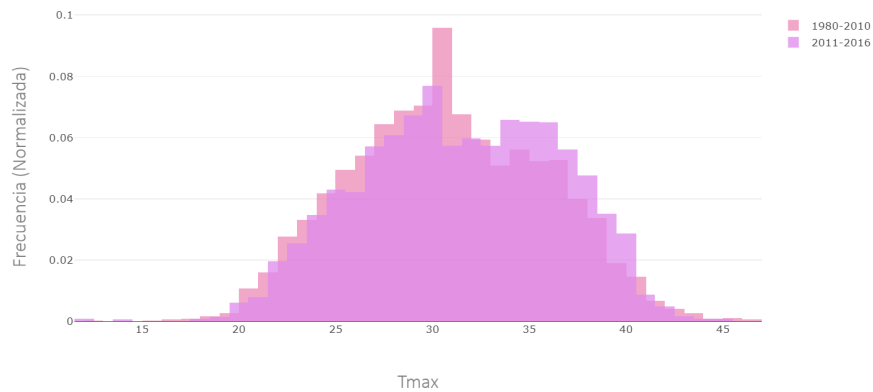


Fig6:Distribución de Tmax Buena Vista San Loreto[3004] (Normalizada en %)

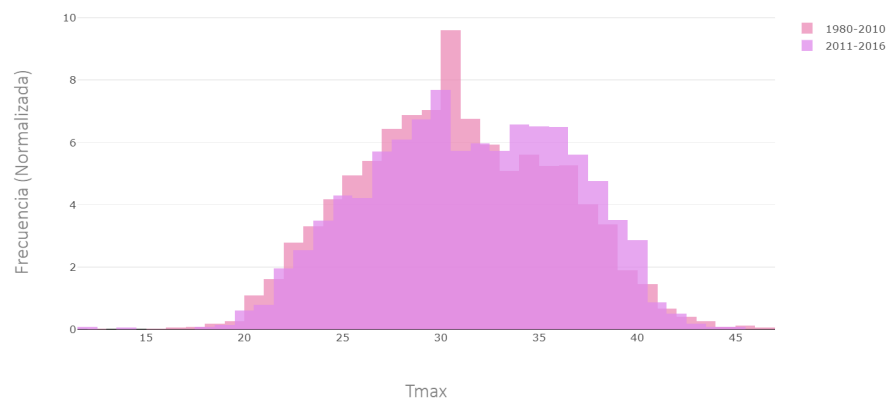
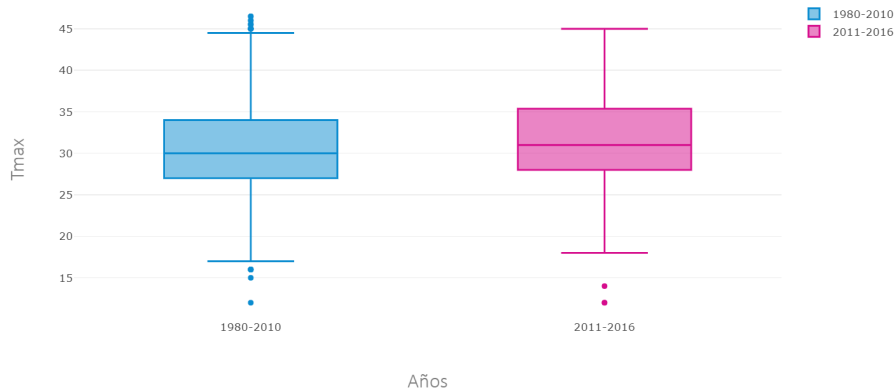


Fig7: Variabilidad de Tmax Respecto a la Normal Climatológica base Buena Vista San Loreto[3004] (1980-2010)



3 Loess Smooth Graph

Para crear la gráfica de suavisamiento, necesitamos agregar una nueva columna que nos exprese solamente el año.

```
df2['year'] = df2['Ndate'].dt.year
```

Primero, en un dataframe se eligen los datos del año de 1980, y en otro, que tome los datos que sean mayores a 30 grados.

```
df1980 = df2['year'] == 1980
hot1980 = df2['Tmax'] > 30
```

Entonces, vamos a pedir que en una nuevo dataframe aparezcan todos los valores donde coincidan los dos dataframes anteriores. Es decir, que aparezcan los valores donde estén los datos de 1980 que tenían una temperatura mayor a 30 grados.

```
dfhot1980 = df2[df1980 & hot1980]
```

Como se quiere saber cuántos números fueron los días en los que 1980 tuvo temperaturas mayores a 30, podemos usar dos funciones para ver su número y guardarlo en otro dataframe. Una es:

```
d80 = len(dfhot1980.index)
```

La otra es:

```
d80 = dfhot1980.shape
```

Se reproduce el mismo proceso para todos los demás años hasta llegar al 2010, creando después un arreglo que tenga año y número de días calientes. Para esto, primero debemos tener un nuevo dataframe con un arreglo vacío donde se le agregarán los años desde 1980 a 2010, siendo este la primera fila.

Del mismo modo, se crea otro dataframe, este con los números de días calientes que le corresponden a cada año, siendo esta la segunda fila.

```
HotList = [d80, d81, d82, d83, d84, d85, d86, d87, d88, d89, d90, d91, d92,
d93, d94, d95, d96, d97, d98, d99, d00, d01, d02, d03, d04, d05, d06, d07,
d08, d09, d10]
```

```
df_HotDays = pd.DataFrame(HotList, columns = ['HotDays'])
```

Concatenamos ambas filas en un nuevo dataframe, obteniendo el dataframe que se utilizará para la gráfica de Loess.

```
df_HotClima = pd.concat([df_HotYears, df_HotDays], axis = 1)
```

Así, graficamos los años calientes contra sus días calientes, pasando en ellas la gráfica de suavizado.

