

[Análisis de Series de Tiempo con Pandas en Pitiquito, Sonora

Martínez Cerda Mario Antonio

12 de Febrero de 2021

1 Introducción

Una serie de tiempo es cualquier conjunto de datos ordenados en el tiempo a intervalos iguales. Hemos visto que los datos climatológicos que descargamos del Servicio Meteorológico Nacional son datos diarios, se tiene una serie de tiempo para la Precipitación, una para la Evaporación, y las de las Temperaturas (Máxima y Mínima). Podrían ser datos cada hora, cada minuto o quizá 10 datos por segundo, dependiendo del fenómeno que se desee estudiar.

Se llama Análisis de Series de Tiempo al proceso de explorar patrones, regularidades, periodicidad, o realizar un análisis estadístico de una serie de tiempo (promedio aritmético, desviación estándar, promedios móviles, otras).

Hay varias técnicas para analizar Series de Tiempo:

Una es en el dominio temporal (auto-correlación, correlación-cruzada, tendencia, estacionalidad, descomposición de series, otras). Otra es en el espacio de frecuencias (análisis espectral, análisis de ondículas (wavelets)). Por ejemplo: Análisis de Series de Fourier y Transformadas de Fourier.

En Física se utilizan mucho estas técnicas para el estudio de una gran diversidad de fenómenos, donde las variables físicas dependen del tiempo. Es un campo muy interesante, muy amplio y muy activo.

En nuestro caso nos limitaremos a hacer análisis en el dominio temporal, y nos apoyaremos de Pandas para analizar nuestras series de tiempo de datos climatológicos.

En el siguiente documento veremos las actividades relacionadas a las series de tiempo y las evidencias hechas en Google Colab, disponibles en Github.

2 Actividades y evidencias.

2.1 Actividad 1.

Explora tus datos de Precipitación los últimos años 1990- fin de datos.

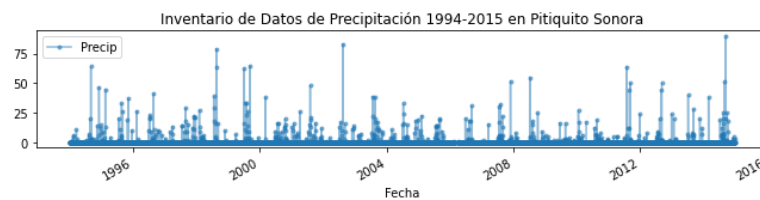
- * Gráfica todo el rango de datos utilizando la función `df.plot()`
- * Analiza un año específico completo, utiliza la función `df.loc()` para seleccionar el rango de datos.
- * Gráfica la estacionalidad de tus datos por mes utilizando la función `sns.barplot()`

```
[4] #Creamos conjunto de datos que separen los datos generales obtenidos.
df_52 = df_A5_ts[(df_A5_ts['Año'] >= 1952) & (df_A5_ts['Año'] < 1973)]
df_73 = df_A5_ts[(df_A5_ts['Año'] >= 1973) & (df_A5_ts['Año'] < 1994)]
df_94 = df_A5_ts[(df_A5_ts['Año'] >= 1994) & (df_A5_ts['Año'] < 2015)]
```

```
df_94.head()
```

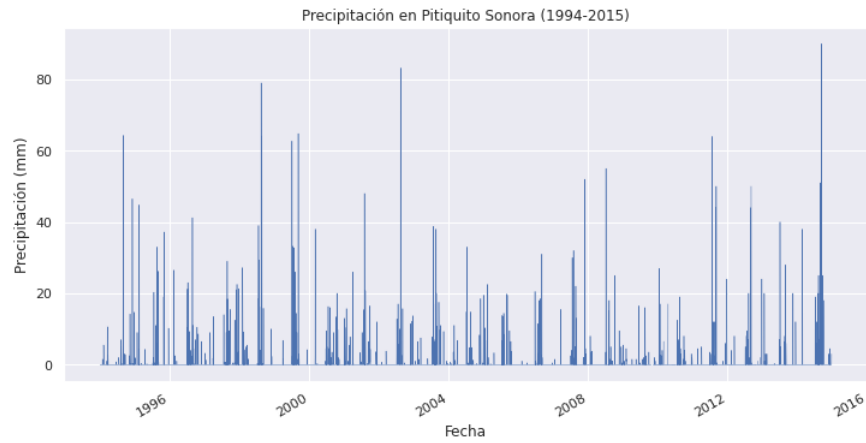
	Precip	Evap	Tmax	Tmin	Año	Mes
Fecha						
1994-01-01	0.0	3.6	25.0	5.5	1994	Jan
1994-01-02	0.0	4.1	24.0	6.0	1994	Jan
1994-01-03	0.0	4.9	27.0	7.0	1994	Jan
1994-01-04	0.0	5.1	30.0	5.0	1994	Jan
1994-01-05	0.0	3.0	26.0	5.0	1994	Jan

```
cols_plot = ['Precip']
ax_ylabels = ['Precip (mm)']
axes = df_94[cols_plot].plot(marker='.', alpha=0.5, linestyle='-', figsize=(11,2), subplots=True)
plt.title('Inventario de Datos de Precipitación 1994-2015 en Pitiquito Sonora');
```

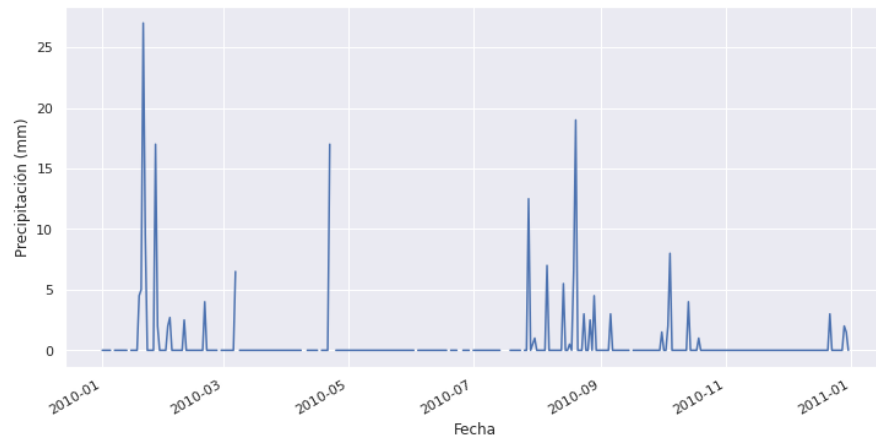


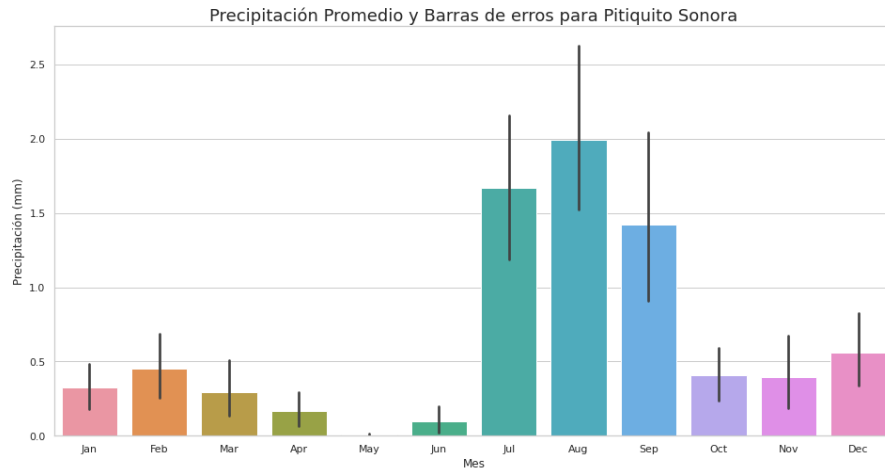
```
sns.set(rc={'figure.figsize':(12,6)})
df_94['Precip'].plot(linewidth=0.5);
plt.ylabel('Precipitación (mm)');
plt.title('Precipitación en Pitiquito Sonora (1994-2015)')
```

Text(0.5, 1.0, 'Precipitación en Pitiquito Sonora (1994-2015)')



```
sns.set(rc={'figure.figsize':(12,6)})
ax = df_94.loc['2010', 'Precip'].plot()
ax.set_ylabel('Precipitación (mm)');
```



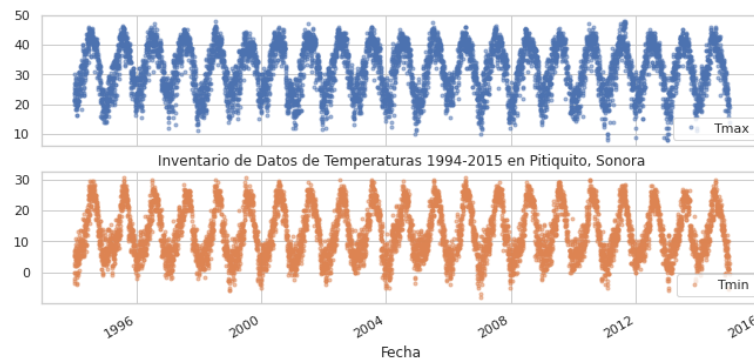


2.2 Actividad 2.

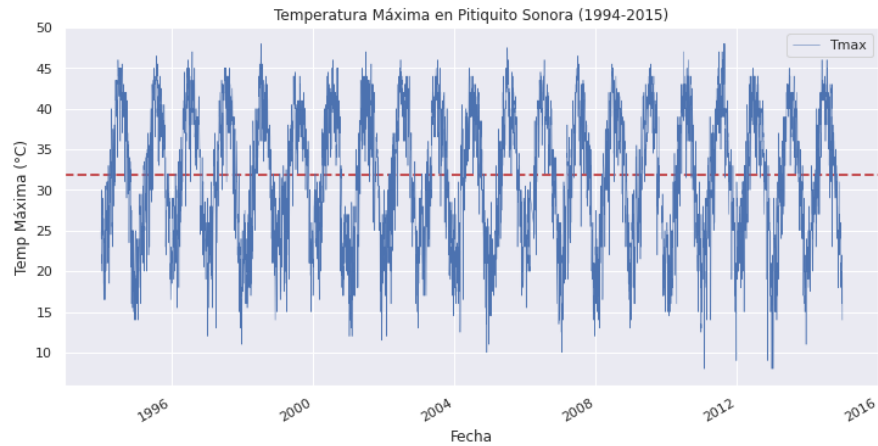
Explora tus datos de Temperaturas Máxima y Mínima de los últimos 30 años datos a partir de 1990. Trata de construir dos subgráficas contiguas.

- * Gráfica todo el rango de datos utilizando la función `df.plot()`, incluye una línea punteada mostrando el promedio.
- * Analiza un año específico completo para las dos Temperaturas, con apoyo de la función `df.loc()`
- * Gráfica la estacionalidad filtrando tus datos por mes y utilizando la función `sns.boxplot()`, incluye una línea punteada mostrando el promedio.

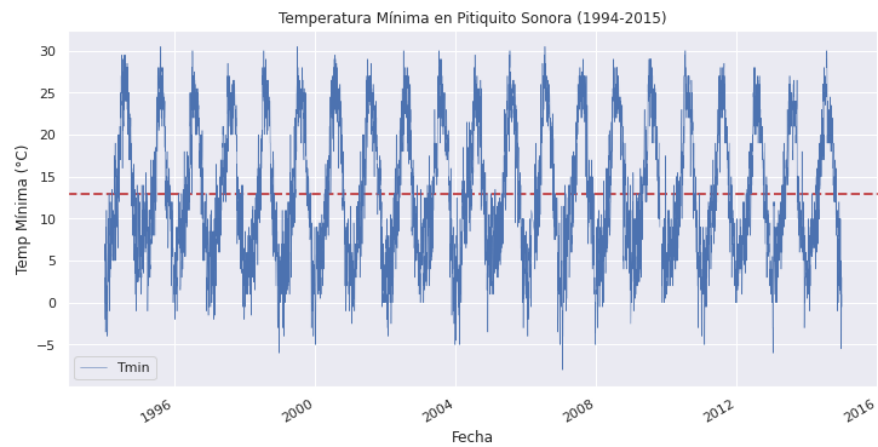
```
cols_plot = ['Tmax', 'Tmin']
ax_ylabels = ['Temperatura Máxima (°C)', 'Temperatura Mínima (°C)']
axes = df_94[cols_plot].plot(marker='.', alpha=0.5, linestyle='None', figsize=(11,5), subplots=True)
plt.title('Inventario de Datos de Temperaturas 1994-2015 en Pitiquito, Sonora');
```



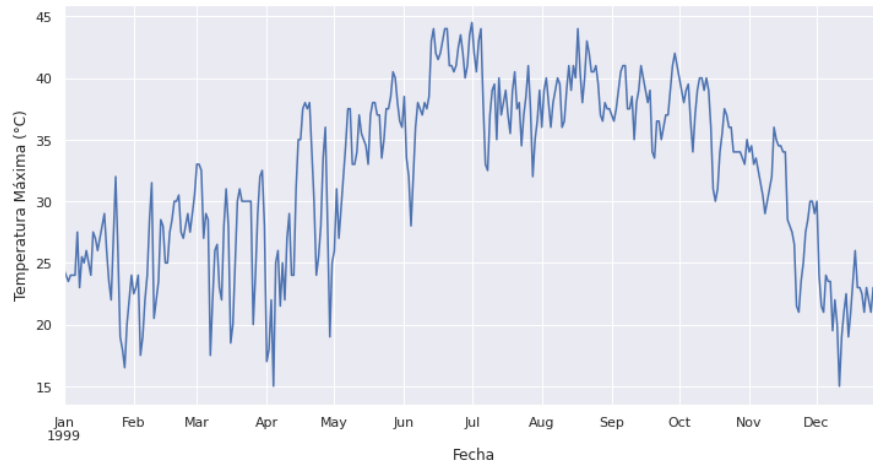
```
#Gráfica de la temperatura máxima de 1994-2015.
sns.set(rc={'figure.figsize':(12,6)})
Tmax_ave = df_94.Tmax.mean()
plt.axhline(Tmax_ave, color = 'r', linestyle='--', linewidth=2);
df_94['Tmax'].plot(linewidth=0.5);
plt.ylabel('Temp Máxima (°C)');
plt.title('Temperatura Máxima en Pitiquito Sonora (1994-2015)')
plt.legend();
```



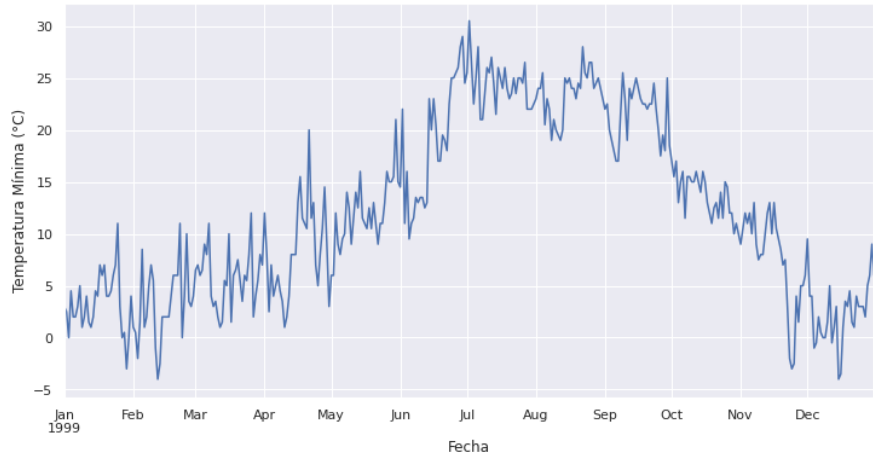
```
#Ahora, la gráfica mínima.
sns.set(rc={'figure.figsize':(12,6)})
Tmin_ave = df_94.Tmin.mean()
plt.axhline(Tmin_ave, color = 'r', linestyle='--', linewidth=2);
df_94['Tmin'].plot(linewidth=0.5);
plt.ylabel('Temp Mínima (°C)');
plt.title('Temperatura Mínima en Pitiquito Sonora (1994-2015)')
plt.legend();
```



```
#Analizando la temperatura máxima del año 1999.
sns.set(rc={'figure.figsize':(12,6)})
ax = df_94.loc['1999', 'Tmax'].plot()
ax.set_ylabel('Temperatura Máxima (°C)');
```

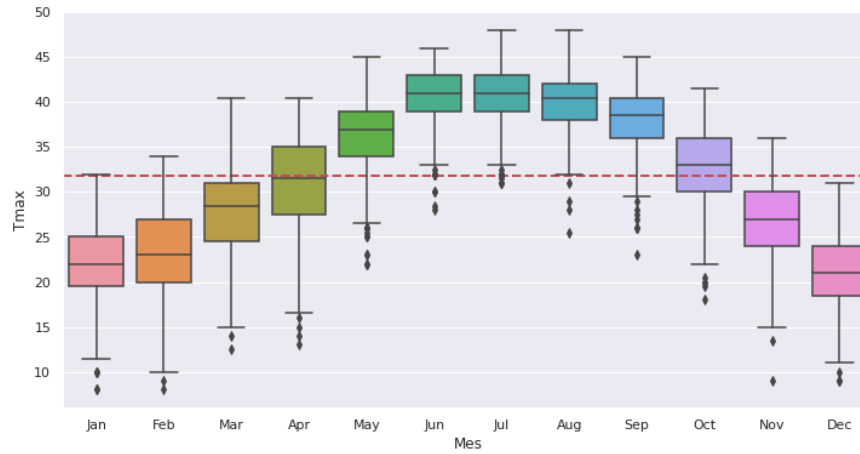


```
#Ahora la temperatura mínima.
sns.set(rc={'figure.figsize':(12,6)})
ax = df_94.loc['1999', 'Tmin'].plot()
ax.set_ylabel('Temperatura Mínima (°C)');
```



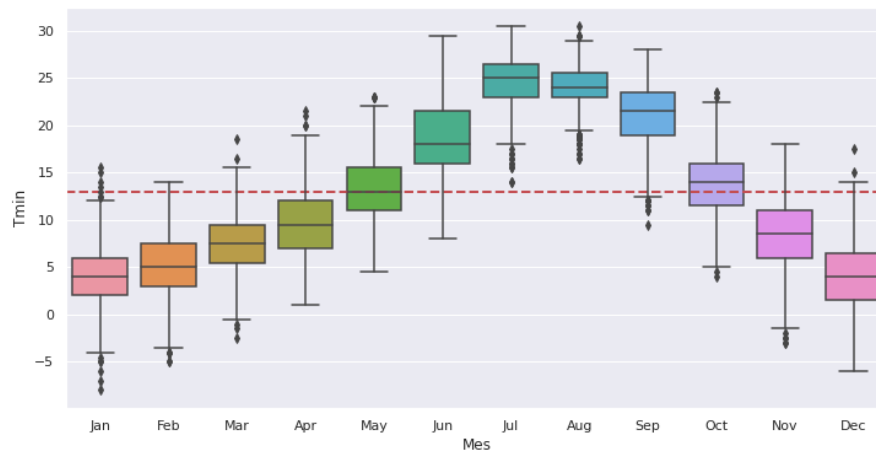
```
#Graficando diagramas de caja de Tmax.
Tmax_ave = df_94.Tmax.mean()
plt.axhline(Tmax_ave, color = 'r', linestyle='--', linewidth=2);
sns.boxplot(data=df_94, x='Mes', y='Tmax')
ax.set_title('Tmax')
```

Text(0.5, 1.0, 'Tmax')



```
#Graficando diagramas de caja de Tmin.
Tmin_ave = df_94.Tmin.mean()
plt.axhline(Tmin_ave, color = 'r', linestyle='--', linewidth=2);
sns.boxplot(data=df_94, x='Mes', y='Tmin')
ax.set_title('Tmin')
```

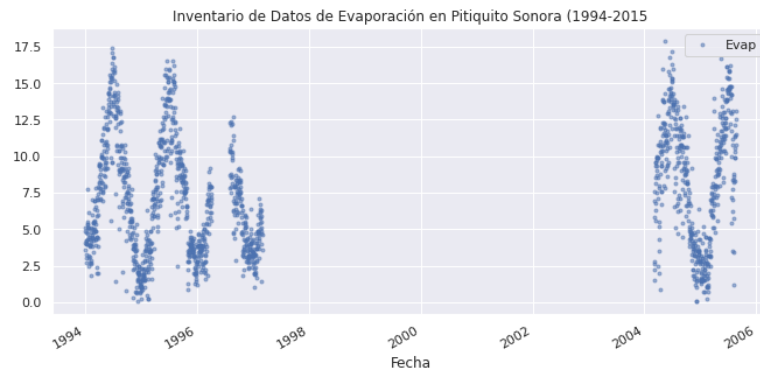
Text(0.5, 1.0, 'Tmin')



2.3 Actividad 3.

Realiza el mismo análisis que en la Actividad 2, pero para la variable de Evaporación.

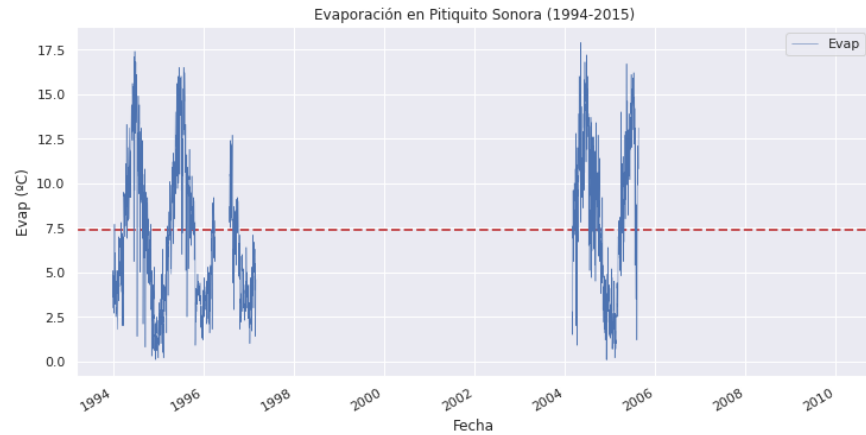
```
cols_plot = ['Evap']
ax_ylabels = ['Evap (mm)']
axes = df_94[cols_plot].plot(marker='.', alpha=0.5, linestyle='None', figsize=(11,5), subplots=True)
plt.title('Inventario de Datos de Evaporación en Pitiquito Sonora (1994-2015)');
```



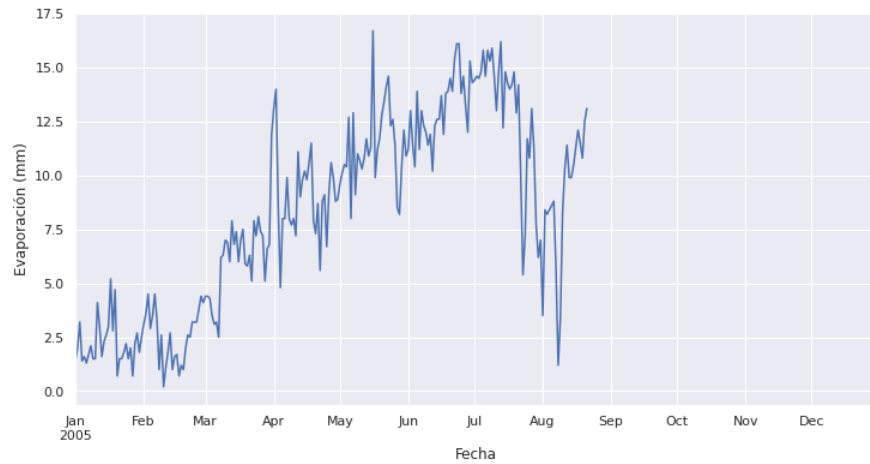
```
# Calculo de promedio
Evap_ave = df_94.Evap.mean()
plt.axhline(Evap_ave, color = 'r', linestyle='--', linewidth=2);

df_94['Evap'].plot(linewidth=0.5);
plt.ylabel('Evap (°C)');
plt.title('Evaporación en Pitiquito Sonora (1994-2015)')

plt.legend();
```




```
sns.set(rc={'figure.figsize':(12, 6)})
ax = df_94.loc['2005', 'Evap'].plot()
ax.set_ylabel('Evaporación (mm)');
```



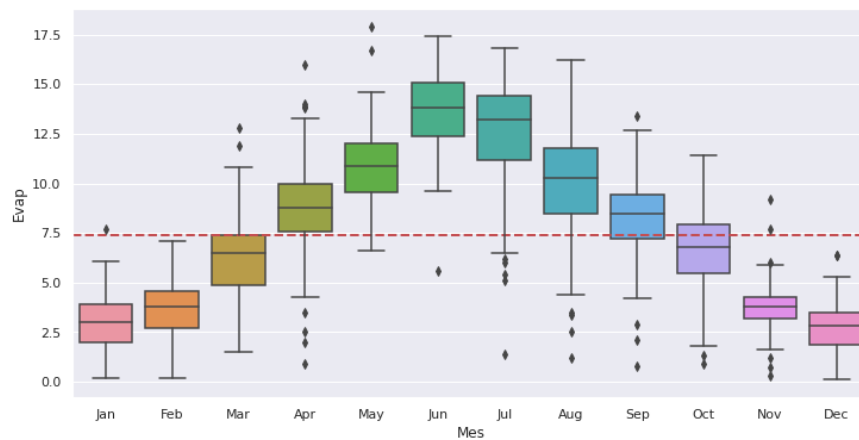
```
evap_ave = df_94.Evap.mean()

plt.axhline(Evap_ave, color = 'r', linestyle='--', linewidth=2);

sns.boxplot(data=df_94, x='Mes', y='Evap')

ax.set_title('Evap')

Text(0.5, 1.0, 'Evap')
```



2.4 Actividad 4.

Crea nuevos DataFrames de las Temperaturas y la Evaporación con promedios móviles de 7, 30 y 365 días, utilizando la función `df.rolling()`. Explora tus datos:

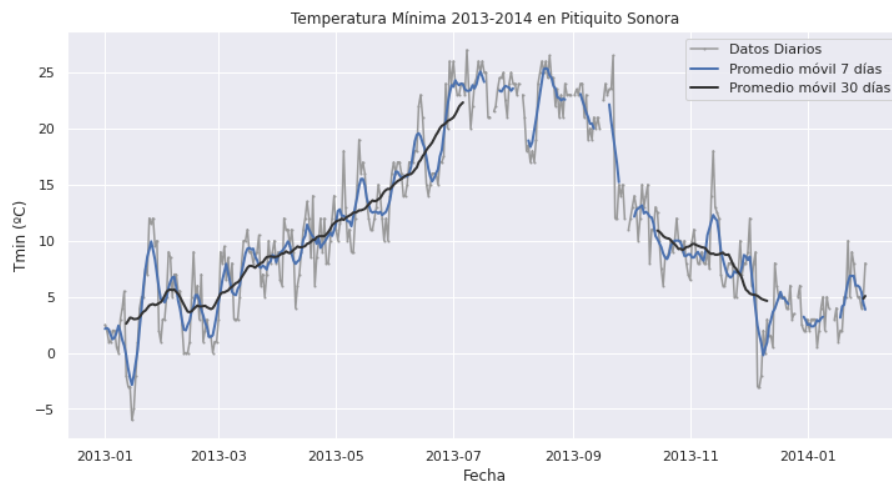
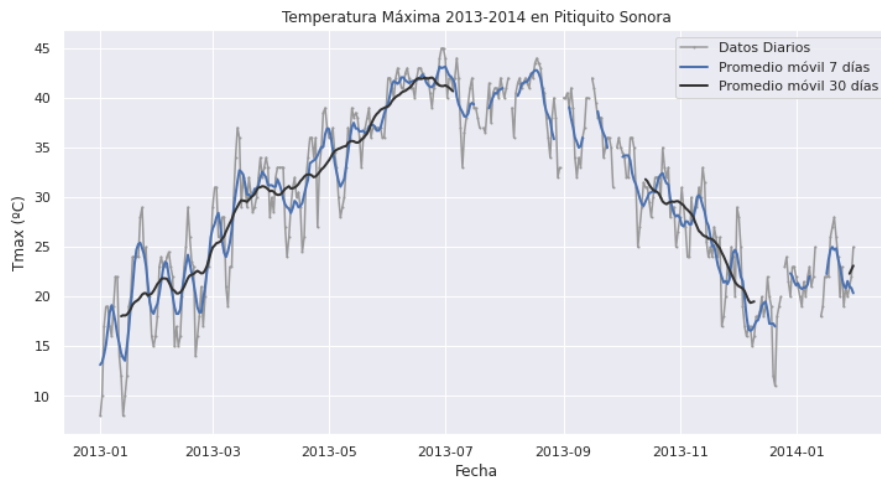
- * Para un año de datos, contrastando los datos diarios con el promedio móvil de 7 y 30 días.

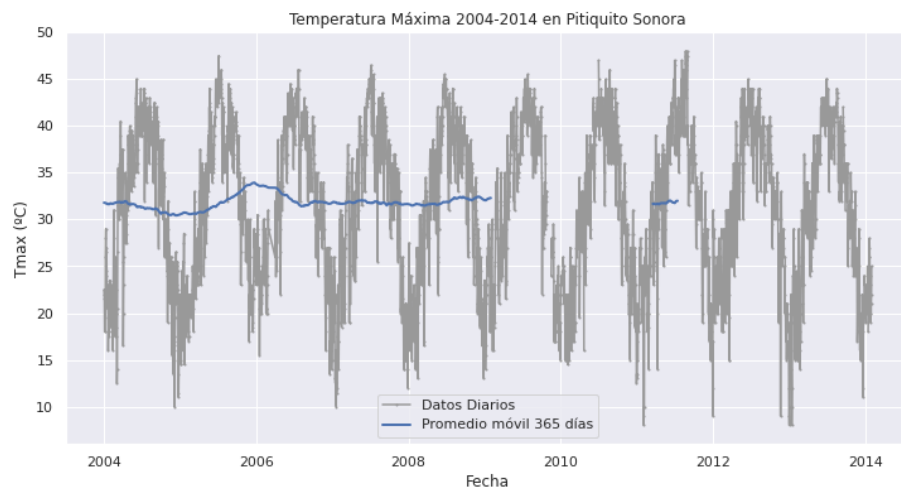
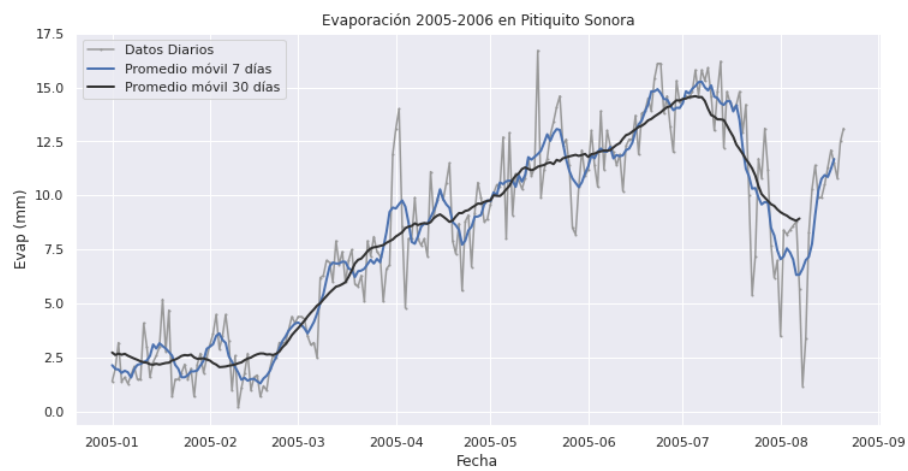
- * Explora para 10 o más años de datos, contrastando los datos diarios con las gráficas de promedio móvil de 365 días.

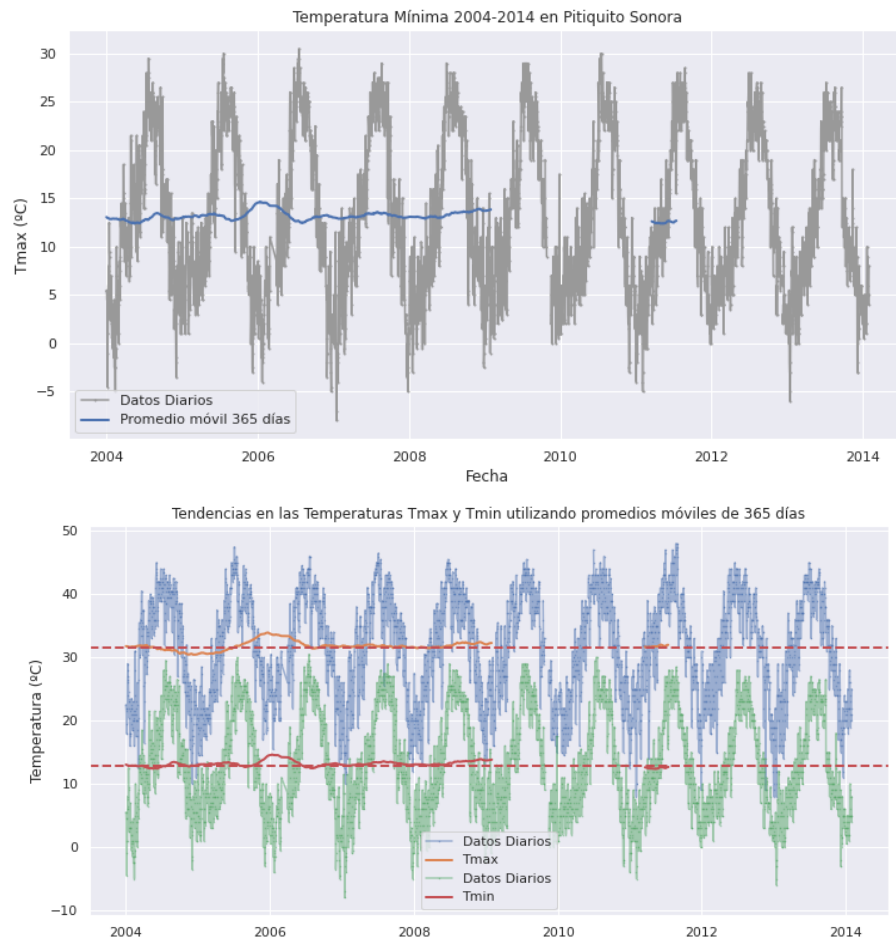
```
data_cols = ['Evap', 'Tmax', 'Tmin']
df_94_7d = df_94[data_cols].rolling(7, center=True).mean()
df_94_30d = df_94[data_cols].rolling(30, center=True).mean()
df_94_365d = df_94[data_cols].rolling(365, center=True).mean()

start, end = '2013-01', '2014-01'
sns.set(rc={'figure.figsize':(12,6)})
fig, ax = plt.subplots()
ax.plot(df_94.loc[start:end, 'Tmax'],
        marker='.', markersize=2, linestyle='-', color='0.6',label='Datos Diarios')
ax.plot(df_94_7d.loc[start:end, 'Tmax'], linestyle='-', linewidth=2, label='Promedio móvil 7 días')
ax.plot(df_94_30d.loc[start:end, 'Tmax'], linestyle='-', color='0.2', linewidth=2, label='Promedio móvil 30 días')

ax.set_xlabel('Fecha')
ax.set_ylabel('Tmax (°C)')
ax.set_title('Temperatura Máxima 2013-2014 en Pitiquito Sonora')
ax.legend();
```



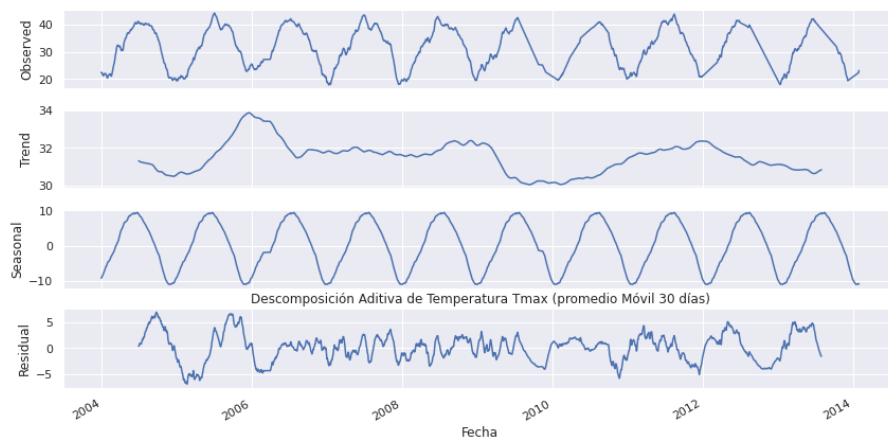




2.5 Actividad 5.

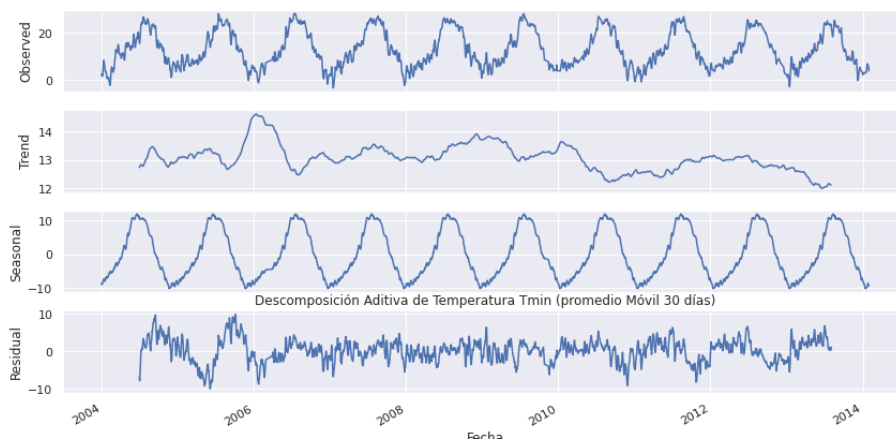
Utiliza la biblioteca `statsmodels.tsa` de Python, para realizar una descomposición de una serie de tiempo: $\text{Serie Observada} = \text{Tendencia} + \text{Estacionalidad} + \text{Residuo}$.

Aplica la función `seasonal_decompose` para lograr esto, a los datos de Temperaturas Tmax y Tmin, y si tienes datos de Evaporación, selecciona un periodo que haya datos. La función de `seasonal_decompose` aplica su propio promedio móvil a las series de datos para su análisis.



```
resTmax.trend.describe()
```

```
count    3265.000000
mean      31.493870
std        0.806979
min        30.031455
25%        30.840108
50%        31.625936
75%        31.928037
max        33.854338
Name: Tmax, dtype: float64
```



```
resTmin.trend.describe()

count    3265.000000
mean      13.068060
std        0.458545
min       12.011546
25%       12.776321
50%       13.067319
75%       13.292564
max       14.631703
Name: Tmin, dtype: float64
```

2.6 Actividad 6.

Por favor explora visualmente si se observa un cambio en la distribución de Temperaturas (histogramas y kde), sobreponiendo dos histogramas de dos periodos de 30 años contiguos. Utiliza la función de Seaborn: `sns.distplot`. Agrega una línea vertical punteada indicando el valor del promedio de cada grupo de datos.

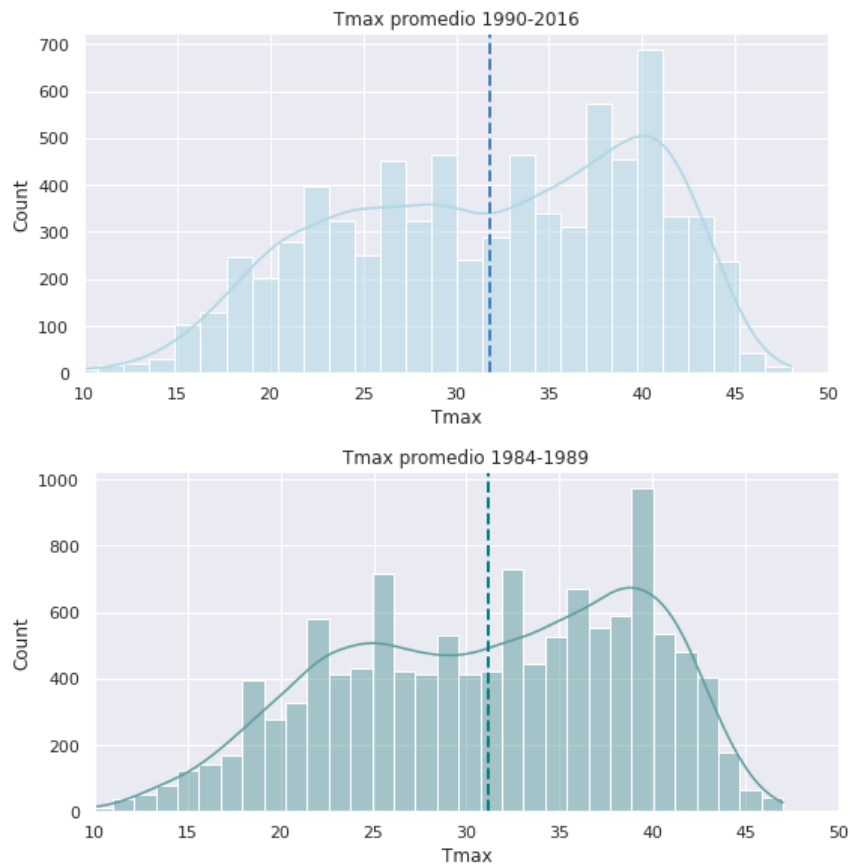
```
g1 = sns.distplot(df_94.Tmax, kde=True, color='lightblue', height=4, aspect=2);
plt.axvline(df_94.Tmax.mean(), linestyle='--', linewidth=2, color='steelblue');

df_primeros = df_A5_ts[(df_A5_ts['Año'] < 1990)]
g2 = sns.distplot(df_primeros.Tmax, kde=True, color='cadetblue', height=4, aspect=2);
plt.axvline(df_primeros.Tmax.mean(), linestyle='--', linewidth=2, color='teal');

g1.set(title='Tmax promedio 1990-2016');
g2.set(title='Tmax promedio 1984-1989');
g1.set(xlim=(10,50))
g2.set(xlim=(10,50))

plt.show();

print('Temperatura máxima promedio del periodo 1990-2016 = ', df_94.Tmax.mean(), '°C' )
print('Temperatura máxima promedio del periodo 1984-1986 = ', df_primeros.Tmax.mean(), '°C' )
```



3 Conclusión

Al realizar la actividad se me dificultó el saber la función de cada comando para editar el tipo de gráfica, debido a que es extenso. Fue una actividad complicada por lo largo que era hacer las celdas y los fallos que se pueden encontrar en el desarrollo del mismo, el tiempo se vió un poco ajustado pero se pudo realizar con éxito. En este documento vimos un poco de complemento de los anteriores trabajos. Sin embargo, me pareció un poco repetitivo. Mismos datos, con gráficas muy parecidas y es bueno darse cuenta que saltaremos a un nuevo tema.

4 Referencias.

- 1.-[http://computacional1.pbworks.com/w/page/143144913/Actividad%20\(2021-1\)](http://computacional1.pbworks.com/w/page/143144913/Actividad%20(2021-1))
- 2.-<https://www.dataquest.io/blog/tutorial-time-series-analysis-with-pandas/>