Filling Time Series: Rutina de rellenado de datos ausentes en series de tiempos geofísicas en Python

Manual del usuario

Erick Rivera[1,2] & Rolando Duarte[2]

1. Centro de Investigaciones Geofísicas

2. Escuela de Física

Universidad de Costa Rica

Correo electrónico: erick.rivera@ucr.ac.cr, rolando.duartemejias@ucr.ac.cr

Diciembre de 2021

Universidad de Costa Rica

Disponible a través de los repositorios:

https://pypi.org/project/FillingTimeSeries/

https://github.com/cigefi-ucr/FillingTimeSeries

https://github.com/cigefi-ucr/FillingTimeSeriesGUI

Universidad de Costa Rica, San Pedro

Índice

Sección ---------- página

I. Introducción

En los trabajos de Alfaro y Soley (2009) y Ureña, Alfaro y Soley (2016) se desarrolló un procedimiento para el rellenado de datos ausentes en series de tiempo meteorológicas, aplicando un método basado en las componentes principales de la matriz de correlación o covarianza de diversas estaciones cercanas, que presentan alta correlación entre sí. Además, se elaboró otro método basado en la autorregresión aplicado en las series de tiempo, idóneo para los casos cuando no se tiene a disposición información de estaciones cercanas y solo queda la opción de rellenar los datos ausentes usando la información de la propia estación. El proyecto de estos autores culmina con la elaboración de un programa con interfaz gráfica desarrollado en SCILAB, dirigido a personas usuarias en instituciones académicas o que laboran en servicios hidrometeorológicos de la región.

A partir de estos trabajos, varias personas usuarias han destacado la importancia de estos para el rellenado de series de tiempo, pero algunas han mencionado la necesidad de que el código esté disponible en Python, lenguaje que es muy utilizado en diversas áreas, en específico, relacionadas con las ciencias. Por ello, se crea Filling Time Series, paquete de Python y aplicación de escritorio con interfaz gráfica, que recopila alguno de los métodos desarrollados en los trabajos originales y hace que su aplicación sea sencilla e intuitiva y con la capacidad para que la comunidad científica contribuya con el paquete, mediante la posible adición de nuevos métodos de rellenado de series de tiempo.

El presente documento explicará el concepto teórico de los métodos disponibles en Filling Time Series, luego se expondrá la utilización del paquete creado en Python y de la interfaz gráfica, en los sistemas operativos de Linux y Windows. Se discutirán algunos resultados obtenidos en cada método para que la persona usuaria conozca el funcionamiento de la herramienta y pueda aplicarla en sus actividades.

II. Marco Teórico

Filling Time Series dispone de tres métodos para el rellenado de series de tiempo. El primer método está basado en componentes principales de la matriz de correlación o covarianza de datos compuestos por estaciones climatológicamente cercanas, el segundo en la autorregresión de una cierta cantidad de valores anteriores y posteriores, lo cual es ideal para aquellos casos en los que no se posea información de estaciones cercanas y se necesite rellenar los datos ausentes usando la información de la propia estación y un tercer método que integra los métodos anteriormente mencionados.

II.I. Análisis en componentes principales

Alfaro y Soley (2009) explican que el método se puede utilizar en un conjunto de datos que poseen series de tiempo de distintas estaciones climatológicamente

cercanas; esto se traduce, matemáticamente, en los elementos con valores altos obtenidos en la matriz de covarianza o correlación asociado al conjunto de datos. Primero se deben rellenar los datos ausentes con el valor promedio de su respectiva serie de tiempo. Luego, el algoritmo propuesto para la aplicación de este método se basa en:

1- Cálculo de la matriz de covarianza o correlación y sus valores propios con los respectivos vectores propios
2- Cálculo de las componentes principales mediante la fórmula $Y = X_0EL$, donde X_0 representa la matriz original de los datos.
3- Se sabe que los datos originales se obtienen mediante la fórmula $X_0 = YLE^T$. Se pueden estimar los datos ausentes mediante la fórmula $X_a = YL'ET$, usando los primeros k componentes principales, truncando la matriz con los valores propios a un tamaño kxk, siendo k menor a la cantidad de estaciones tomadas en cuenta, es decir, las columnas del conjunto de datos.

Se cambian los datos ausentes, que se habían introducido como el promedio de la serie de tiempo respectiva, por el valor predicho en X_a, permitiendo volver a realizar el procedimiento desde el paso 1. La idea principal es iterar varias veces, hasta que la máxima diferencia entre predicciones sucesivas sea menor a un valor de tolerancia provisto por la persona usuaria, es decir, $\max\{\|X_{i+1} - X_i\|\} < tolerance$, donde X_i es la predicción en la iteración i-ésima. Si no se alcanza el valor de tolerancia, se iterará una cantidad de veces igual a la especificada por el usuario.

El valor de tolerancia y la cantidad máxima de iteraciones dependerán de la selección de la persona usuaria, pero lo más importante es seleccionar la cantidad k de componentes principales a utilizar. El procedimiento recomendado a utilizar se basa en el trabajo de Wiks (1995) y North et al. (1982), a partir del gráfico de Scree con los autovalores y sus respectivas barras de error (ver Figura 1). Dependiendo de los componentes principales, se debe elegir la menor cantidad de componentes principales, con excepción en el caso que las barras de errores del autovalor del menor valor de los componentes principales se traslape con el autovalor del siguiente componente principal. En dado caso, se debe elegir el mayor de los componentes principales entre los que poseen autovalores traslapados.

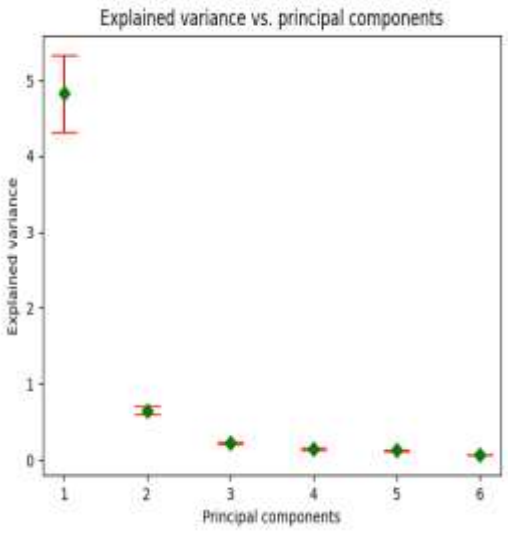

Figura 1. Ejemplo del gráfico de Scree con evidencia de utilizar, únicamente, el primer

Al terminar el algoritmo, si existen valores menores que una cota mínima dado por la persona usuaria, estos valores cambiarán por la cota mínima.

II.II. Autorregresión (Ulrych y Clayton método)

Para los casos donde se posea la información de estaciones cercanas, Alfaro y Soley (Alfaro & Soley, 2009), establecen que los datos ausentes se pueden rellenar con la información de los registros anteriores y posteriores con un filtro predictivo $AR(\rho)$, siendo ρ el orden del modelo autorregresivo, con la principal característica que cumple con el principio de Máxima Entropía, siendo los valores rellenados consistentes con la estadística de la serie de tiempo completa.

El modelo autorregresivo de orden ρ sigue la fórmula:

$$\hat{y}[k] = \phi_1 y[k-1] + \phi_2 y[k-2] + \phi_3 y[k-3] + \cdots + \phi_\rho y[k-\rho] \quad 1)$$

$$\hat{y}[k] = \phi_1 y[k+1] + \phi_2 y[k+2] + \phi_3 y[k+3] + \cdots + \phi_\rho y[k+\rho] \quad 2)$$

La ecuación 1) es la relación del registro en la posición k con los valores anteriores y la ecuación 2), la relación del registro en la posición k con los valores siguientes. Los coeficientes ϕ_i están calculados según lo propuesto por Ulrych y Clayton (1976), que establece que estos coeficientes se calculan con el ajuste de los mínimos cuadrados. Entonces al cambiar un dato, este será el promedio entre los modelos obtenidos con los valores anteriores y los siguientes.

Primero se debe rellenar los datos ausentes con el valor promedio de la serie de tiempo, luego, el algoritmo para rellenarlos con este método es:

1- Se calculan los coeficientes del modelo autorregresivo para el número de coeficientes provistos por la persona usuaria.
2- Se cambian los datos que originalmente estaban ausentes por el resultado del modelo obtenido en el paso anterior.
3- De forma similar al método de componentes principales, se itera tantas veces como sea necesario, hasta que la máxima diferencia entre predicciones sucesivas sea menor que un valor de tolerancia provisto por la persona usuaria o se alcance la cantidad máxima de iteraciones establecida.

Finalmente, si algún valor rellenado es menor que una cota mínima dada por la persona usuaria, se cambia por el valor de dicha cota mínima. Además, nótese que el modelo no puede salirse de los límites de la serie de tiempo, por lo que los primeros ρ valores perdidos se cambiarán usando el filtro predictivo con los valores posteriores, los últimos ρ valores perdidos se cambiarán usando el filtro predictivo con los valores anteriores y los valores perdidos intermedios usarán un promedio de ambos filtros predictivos.

II.III. Método integrado

Este método fue agregado en el trabajo de Ureña et al. (2016), el cual consiste en el uso de componentes principales con un cambio. Anteriormente, al aplicar el método de componentes principales, primero se debía hacer un rellenado de los datos usando el promedio de la serie de tiempo. En el método integrado, en cambio, el rellenado inicial se realiza con el valor obtenido por el método de autorregresión.

III. Manual del usuario

III.I. Preprocesamiento de los datos

Antes de usar Filling Time Series, se debe realizar un preprocesamiento para asegurar buenos resultados.

- Se permite archivos .csv o .txt.
- Si existen valores ausentes codificados con algún valor, cambiarlos a nan (recomendado numpy.nan).
- Conocer, de antemano, la matriz de correlación o la autocorrelación según el retraso que se pueda aplicar.
- No debe haber un encabezado de archivo para el uso de la interfaz gráfica. En el caso del paquete de Python, queda a criterio del usuario si mantiene o no el encabezado.
- Las columnas se separan por un espacio.
- No debe existir tendencia en la serie de tiempo.

La Figura 2 es un ejemplo de cómo debe de lucir el archivo con los datos.

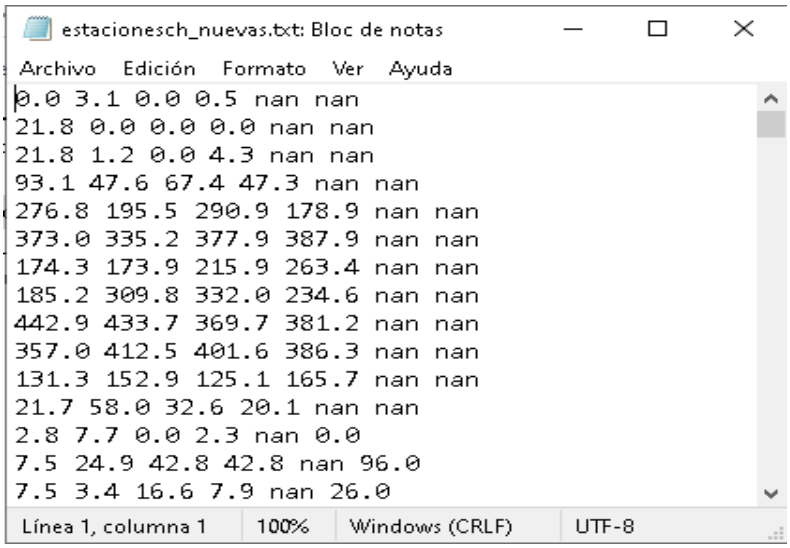

Figura 2. Ejemplo del archivo con los datos de las series de tiempo.

III.II. Paquete de Python: Filling Time Series

(ver https://pypi.org/project/FillingTimeSeries/)

Filling Time Series es un paquete de Python creado en este proyecto para el rellenado de series de tiempo usando los métodos explicados anteriormente. Con pocas líneas de código, el usuario podrá usar el paquete para utilizarlo en sus labores con series de tiempo, con el potencial de poder automatizar los procesos.

Instalación

Filling Time Series podrá ser utilizado en cualquier sistema operativo con alguna versión de Python 3.6 en adelante, El método de instalación es mediante la terminal, ejecutando el siguiente comando:

```
pip install FillingTimeSeries
```

Además, este procedimiento instalará las dependencias necesarias para su funcionamiento.

Dependencias:

- NumPy (Harris, Millman, & van der Walt, 2020)
- Pandas (Reback, y otros, 2021)
- Matplotlib (Hunter, 2007)
- Scikit-Learn (Pedregosa, y otros, 2011)
- Statsmodels (Seabold & Perktold, 2010)

Importación del paquete

La forma de importar el paquete se puede hacer mediante un alias:

```
import FillingTimeSeries.FillingMethods as fts
```

O importando las clases que se van a utilizar directamente:

```
from FillingTimeSeries.FillingMethods import PrincipalComponentAnalysis, Autoregression,
ComponentsAutoregression
```

FillingTimeSeries.FillingMethods.Autoregression(df**)**

Inicializa el método de autorregresión de Ulrich & Clayton

Parámetros

df: pandas-dataframe

pandas dataframe a rellenar con valores nulos

Métodos:

- **ULCLMethod(**lags, tol, itermax, valueMin**)**

Aplica el método de autorregresión de Ulrich & Clayton
Parámetros:
lags: entero (int)
Valor de rezago para la aplicación de la autorregresión.
tol: flotante (float)
Valor de tolerancia de la diferencia entre la serie de tiempo rellenada anterior y la serie llena actual a través de las iteraciones.
itermax: entero(int)
Número máximo de iteraciones para encontrar una serie de tiempo rellenada que cumpla con la condición de tolerancia.
valueMin: flotante(float)
El valor mínimo permitido después de aplicar el método de autorregresión.
Regresa:
dfPF: pandas-dataframe
Dataframe de Pandas utilizando valores pasados y futuros para completar los valores faltantes.

FillingTimeSeries.FillingMethods.PrincipalComponentAnalysis(df**)**

Inicializa el método de componentes principales.

Parámetros

df: pandas-dataframe

pandas dataframe a rellenar con valores nulos

Métodos:

- **checkPrincipalComponents ():**
  Muestra los gráficos de la razón de varianza explicada de los componentes principales.
  Regresa:
  upperRange: entero(int)
  Valor máximo a elegir entre los componentes principales.
- **PCAMethod(**components, tol, itermax, valueMin**)**
  Aplica el método de componentes principales.
  Parámetros:
  components: entero(int)
  número de componentes principales.
  tol: flotante(float)
  Valor de tolerancia de la diferencia entre la serie de tiempo rellenada anterior y la serie llena actual a través de las iteraciones.
  itermax: entero(int)
  Número máximo de iteraciones para encontrar una serie de tiempo rellenada que cumpla con la condición de tolerancia.
  valueMin: flotante(float)

El valor mínimo permitido después de aplicar el método de componentes principales.
Regresa:
dfPF: pandas-dataframe
Dataframe de Pandas luego de haber sido aplicado el método.

FillingTimeSeries.FillingMethods.ComponentsAutoregression(df**)**

Inicializa el método de componentes principales.

Parámetros

df: pandas-dataframe

pandas dataframe a rellenar con valores nulos

Métodos:

- **checkPrincipalComponents ():**
  Muestra los gráficos de la razón de varianza explicada de los componentes principales.
  Regresa:
  upperRange: entero(int)
  Valor máximo a elegir entre los componentes principales.
- **FullMethod(**lags, components, tol, itermax, valueMin**)**
  Aplica el método de autorregresión y luego aplica el método de componentes principales.
  Parámetros:
  lags: entero (int)
  Valor de rezago para la aplicación de la autorregresión.
  components: entero(int)
  número de componentes principales.
  tol: flotante(float)
  Valor de tolerancia de la diferencia entre la serie de tiempo rellenada anterior y la serie llena actual a través de las iteraciones.
  itermax: entero(int)
  Número máximo de iteraciones para encontrar una serie de tiempo rellenada que cumpla con la condición de tolerancia.
  valueMin: flotante(float)
  El valor mínimo permitido después de aplicar el método de componentes principales.
  Regresa:
  dfPF: pandas-dataframe
  Dataframe de Pandas luego de haber sido aplicado el método.

Ejemplos del uso del paquete Filling Time Series

Método de autorregresión:

```
from FillingTimeSeries.FillingMethods import Autoregression
import pandas as pd
df = pd.read_csv("estacionesch_nuevas.txt", delimiter="\s+", header=0)
AR = AutoRegression("datest.txt")
dfFilled = AR.ULCLMethod(lags = 1, tol = 0.001, itermax = 1000, valueMin = 0)
```

Método de componentes principales:

```
from FillingTimeSeries.FillingMethods import PrincipalComponentAnalysis
import pandas as pd
df = pd.read_csv("estacionesch_nuevas.txt", delimiter="\s+", header=0)
pca = PrincipalComponentAnalysis(df)
upperRange = pca.checkPrincipalComponents() #This method will plot principal components
dfFilled = pca.PCAMethod(components = 1, tol = 0.001, itermax = 1000, valueMin = 0)
```

Método de componentes principales:

```
from FillingTimeSeries.FillingMethods import ComponentsAutoregression
import pandas as pd
df = pd.read_csv("estacionesch_nuevas.txt", delimiter="\s+", header=0)
full = ComponentsAutoregression(df)
upperRange = full.checkPrincipalComponents() #This method will plot principal components
dfFilled = FullMethod(lags = 1, components = 1, tol = 0.001, itermax = 1000, valueMin = 0)
```

III.III. Interfaz gráfica: FTS - *Filling Time Series*

FTS – Filling Time Series es la interfaz gráfica creada en este proyecto para darle una herramienta intuitiva y sencilla las personas usuarias que necesiten rellenar archivos con series de tiempo, sin la necesidad de conocer conceptos de programación. Está basado en el paquete de Python, Filling Time Series, explicado en la sección anterior.

Inicialización de la interfaz gráfica FTS - Filling Time Series en Windows

Luego de descargarse los archivos necesarios en Windows, para inicializar la aplicación Filling Time Series, se debe ingresar en la carpeta "*Windows/FTS-Filling_Time_Series/*" y dar doble clic sobre el archivo "*FTS-Filling_Time_Series.exe"* (ver Figura 3).

Figura 3. Archivo "*FTS-Filling_Time_Series.exe"* y pantalla inicial de la aplicación de Filling Time Series.

Inicialización de la interfaz gráfica FTS - Filling Time Series en Linux

Luego de descargarse los archivos necesarios en Linux, en la terminal, verifique que se encuentre en el mismo directorio de los archivos y digitando el comando *". run.sh"*, la aplicación se inicializará (ver Figura 4).

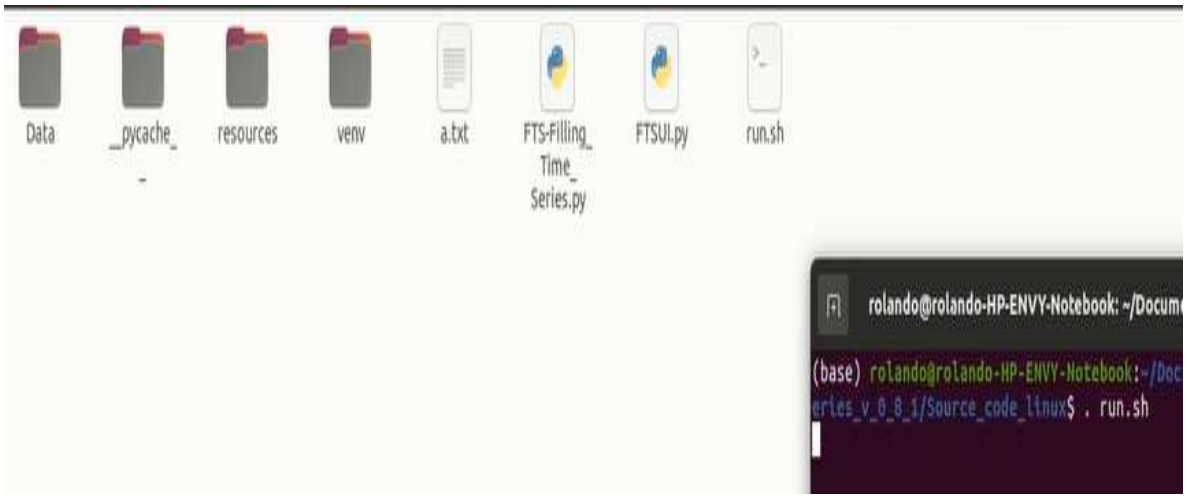

Figura 4. Archivos en Linux y línea de comando para inicializar FTS - Filling Time Series en Linux

Uso de la interfaz gráfica de Filling Time Series en Windows y Linux

El desarrollo de la aplicación de Filling Time Series ha sido pensado de manera que los pasos a realizar, para hacer el rellenado de las series de tiempo geofísicas, sean intuitivos. Primero se debe dar click en el botón "*Search...*" para buscar el archivo que se desea rellenar, cuya extensión debe ser "*.txt" o "*.csv" (ver Figura 5).

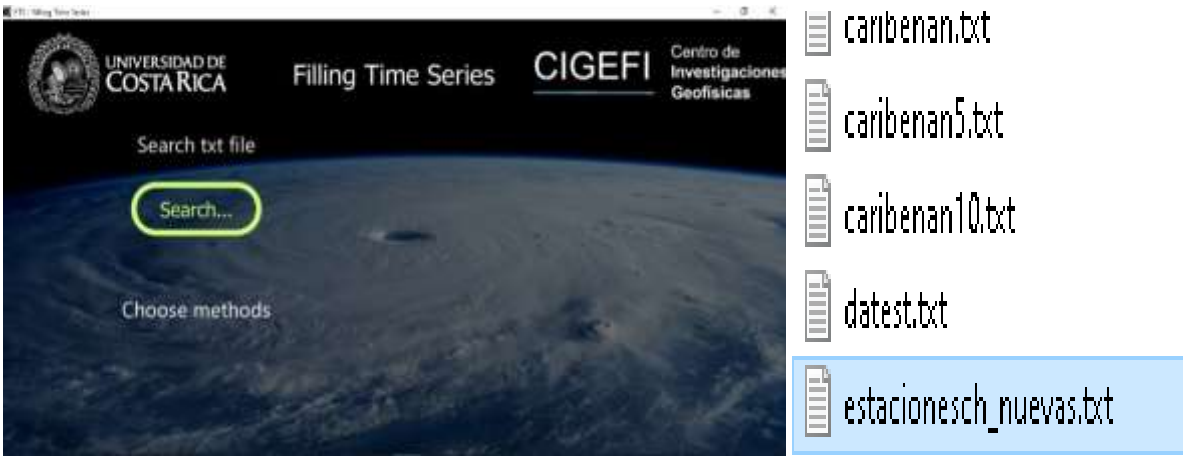

Figura 5. Click en el botón "Search…" y buscar el archivo que se desea rellenar

Luego, aparecerán tres botones con los métodos disponibles en esta aplicación, "*PCA*" para el método de componentes principales, "*ULCL*" para el método de Ulrych y Clayton de autorregresión y "*Full*" para el método integrado. En este caso, se procederá a dar click sobre el botón de "*PCA*" para aplicar el método de componentes principales, seguido, aparecerá el gráfico de Scree, respectivo al caso, con los componentes principales y las proporciones de las varianzas explicadas para que el usuario tome la mejor decisión al escoger la cantidad de componentes principales que serán usados para el rellenado (ver Figura 6).

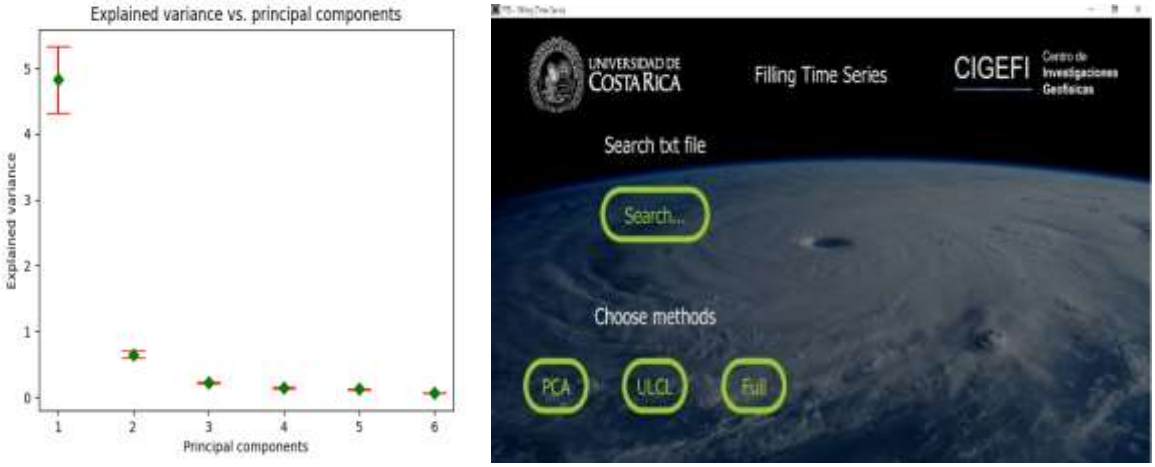

Figura 6. Click en el botón "*PCA*" y aparición del gráfico de Scree.

Seguidamente, aparecerán los campos para los parámetros necesarios para aplicar el método de componentes principales. En el campo de "*Principal components*" se inserta el número de componentes principales que serán usados, en "*Maximum iterations*" se digita el número de iteraciones máxima, si no se ha encontrado una diferencia entre predicciones que sea menor a la tolerancia fijada y en "*Tolerance*" se ingresa la tolerancia de la diferencia entre predicciones. El proceso se termina si la diferencia de predicciones es menor que la tolerancia fijada o hasta que se cumpla la cantidad máxima de iteraciones. "*Mimimum value*" es la cota mínima permitida para el rellenado de la serie de tiempo, esto para preservar el sentido físico del problema que se está estudiando. Ya completados estos valores, se da clic sobre el botón "*Apply & save*", apareciendo una ventana para escribir el nombre del archivo que se guardará con la información de las series de tiempo rellenadas. Los formatos permitidos son "*.txt" y "*.csv" (ver Figura 7).

Figura 7. Insertar los parámetros necesarios para el método de componentes principales, click en el botón "*Apply & save*" y escribir el nombre del archivo con la serie de tiempo rellenada

Finalmente, aparecerá una pantalla con el mensaje "*Waiting the process…*" indicando que el programa está realizando los cálculos respectivos, hasta que se despliega el mensaje en pantalla "File is saved!", que significa que el proceso terminó y el archivo fue guardado. En esta última pantalla aparecerán dos botones, "*Restart*" que permite rellenar algún otro archivo y "*Quit*" que será para salir del programa (ver Figura 8).

Para la aplicación del método autorregresivo, "*ULCL*", y el método integrado, "*Full*", los pasos son similares, teniendo en cuenta que los parámetros para el método autorregresivo son *"Lags*", "*Maximum iterations*", "*Tolerance*" y "Minimum value" y para el método integrado son todos los parámetros que se han explicado en esta sección.

Figura 8. Pantalla final indicando que el archivo ha sido guardado.

IV. Resultados

IV.I. Análisis de componentes principales

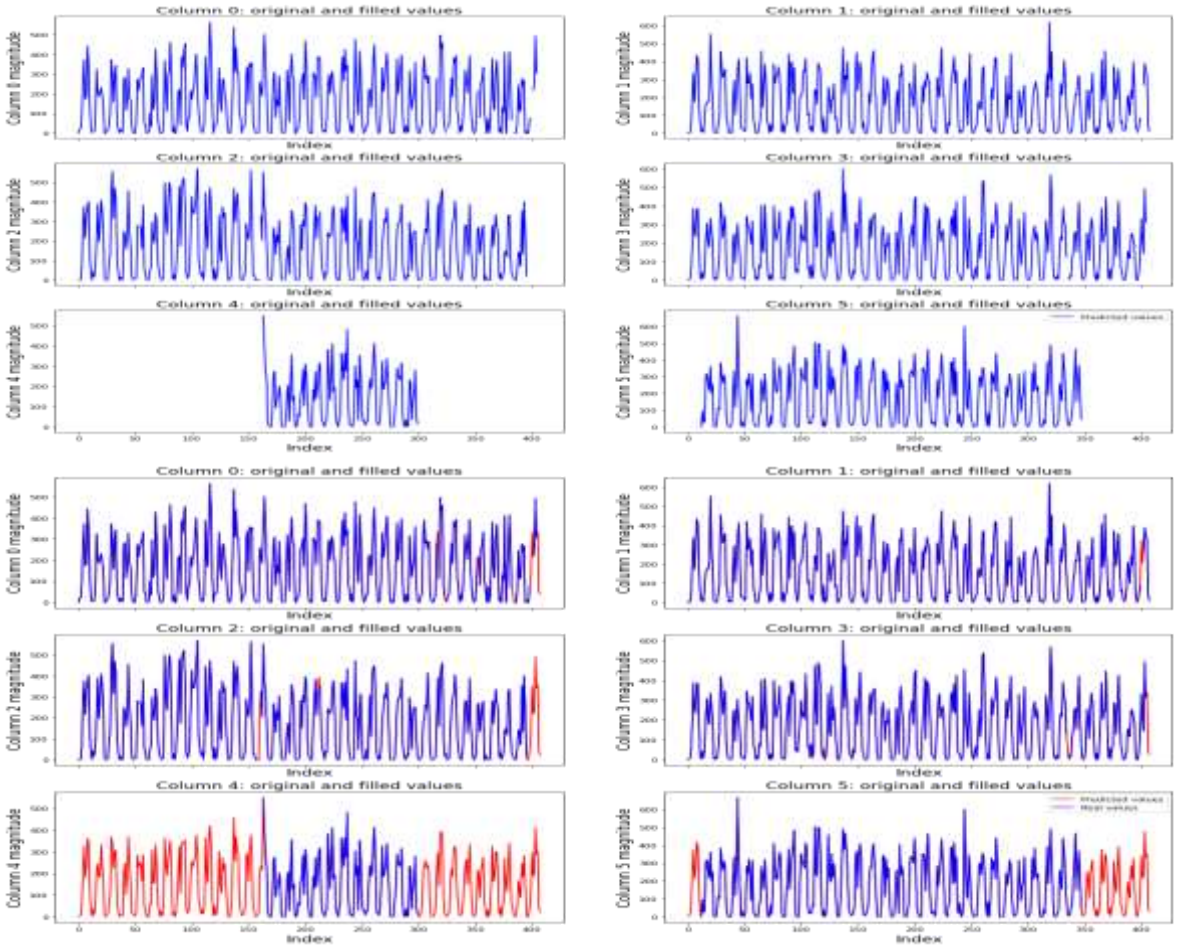

Figura 9. Demostración del rellenado de los datos perdidos en el conjunto de datos en el método de componentes principales.

En este conjunto de datos de 408 registros (34 años) se encuentran 6 columnas con la información de la precipitación medidas en las estaciones meteorológicas de Juan Santamaría, San José, Argentina de Grecia, Fabio Baudrit, Pavas y Embalse de Garita, correlacionadas entre sí con valores que van de 0.86 a 0.94, por lo que es aplicable el método de componentes principales. El porcentaje de valores faltantes en la columna 0 es de 1.96%, columna 1 de 0.98%, columna 2 de 3.43%, columna 3 de 1.23%, columna 4 de 66.42% y la columna 5 de 17.89%. En la utilización de Filling Time Series, se utilizaron los parámetros de 1 componente, con una tolerancia de 0.001, iteración máxima de 1000 y una cota mínima de 0. Se puede observar en la Figura 9, como es que Filling Time Series rellena los datos, obteniendo la información de las demás estaciones para completar los datos, incluso en la columna 4, la cual posee el mayor porcentaje de datos perdidos, facilitando el análisis posterior.

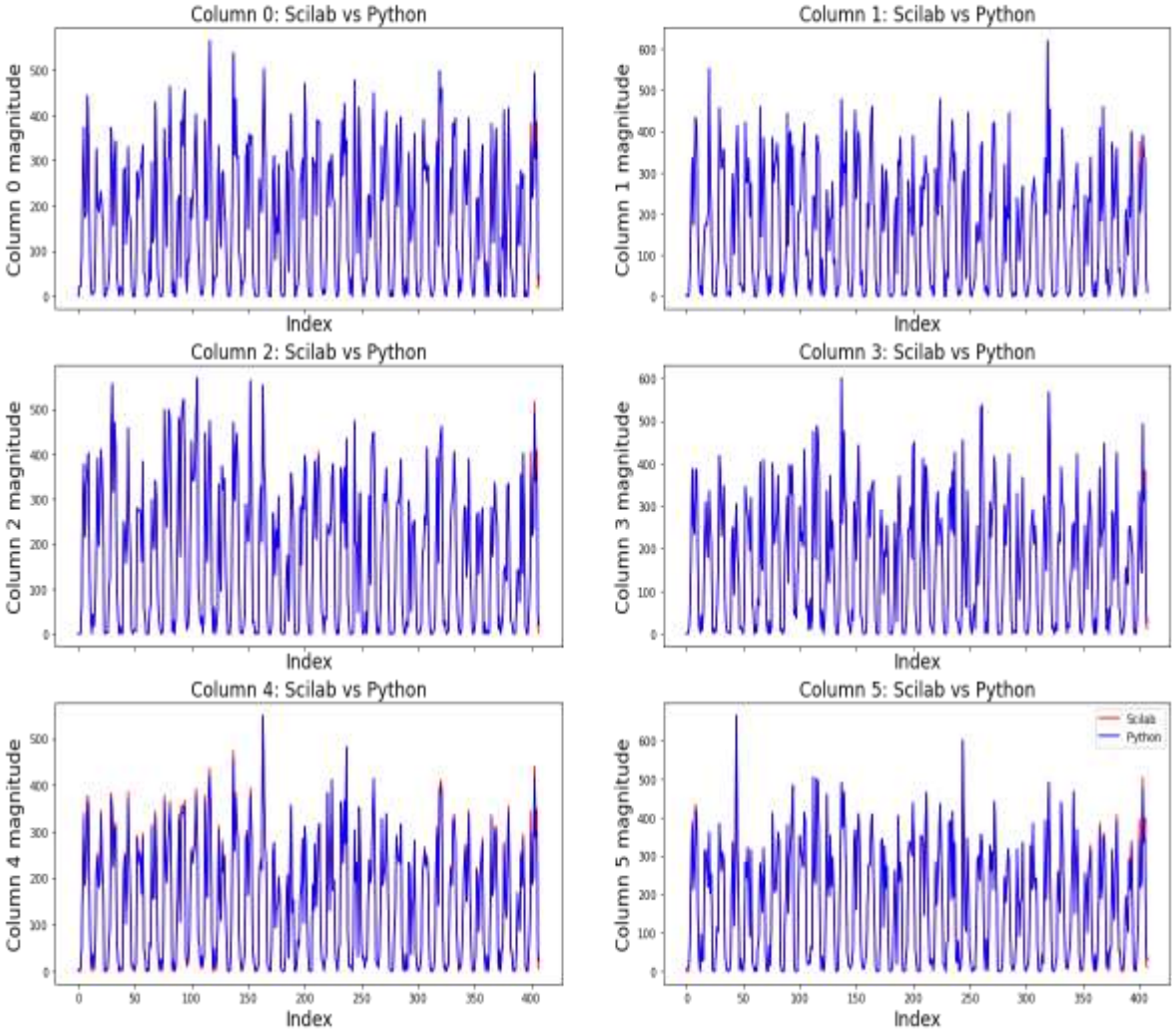

Figura 10. Comparación entre Filling Time Series y la rutina original en SCILAB en el método de componentes principales.

Debido a que la rutina original en SCILAB implementa el método de componentes principales, es importante hacer una comparación entre esta rutina y el paquete Filling Time Series para poder identificar si la implementación en Python se ha desarrollado de forma correcta. Por ello, se pueden ver, en la Figura 10, las series de tiempo obtenidas por ambos códigos con los mismos parámetros, siendo difícil apreciar alguna diferencia en ciertos casos. La diferencia absoluta promedio entre los dos códigos es en la columna 0 de 0.38 mm, columna 1 de 0.14 mm, columna 2 de 0.57 mm, columna 3 de 0.23 mm, columna 4 de 4.28 mm y la columna 5 de 1.56 mm, presentándose una mayor diferencia conforme aumenta la cantidad de datos faltantes.

IV.II. Autorregresión: Método de Ulrych y Clayton

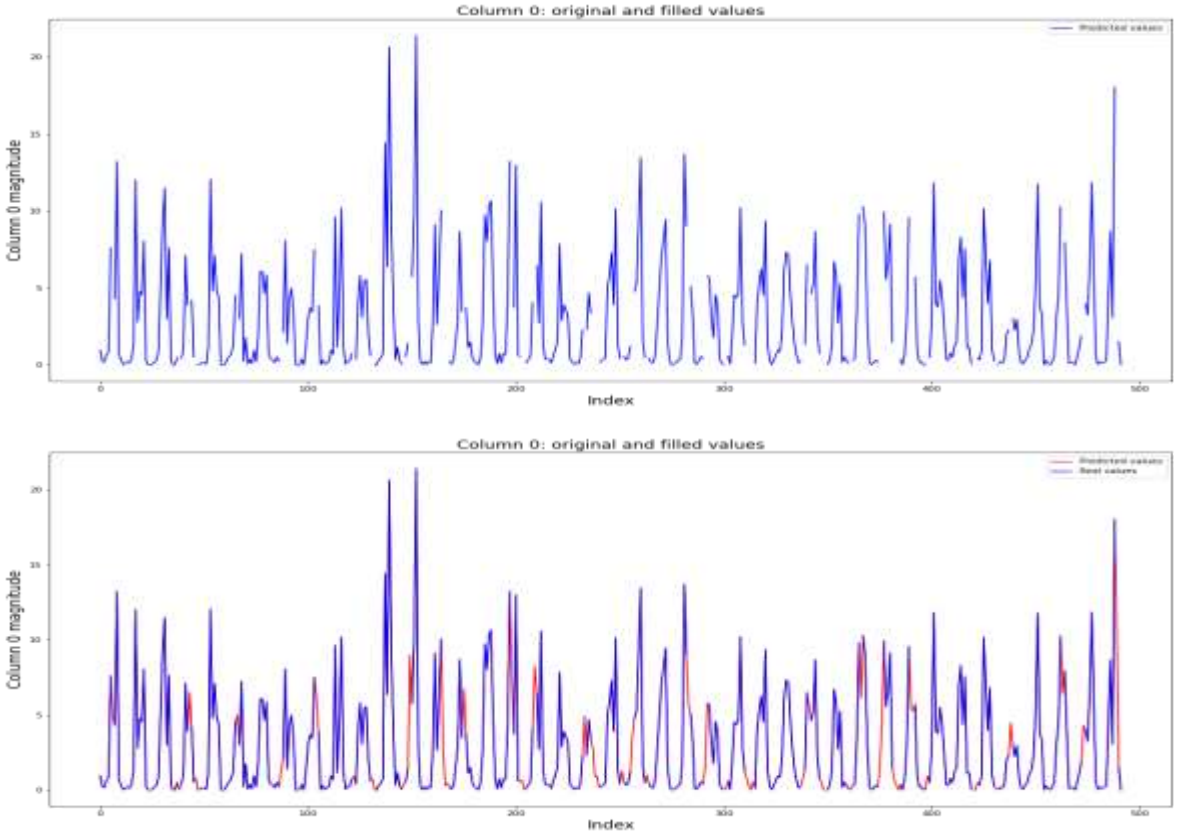

Figura 11. Demostración del rellenado de los datos perdidos en el conjunto de datos en el método de autorregresión.

En este caso, el conjunto de datos contiene 492 registros con una columna, aplicable el método de autorregresión ya que no tenemos información de otras estaciones meteorológicas. En Filling Time Series, se utilizó un valor de k de 12 con una tolerancia de 0.001, máxima iteración de 1000 y cota mínima de 0. El porcentaje de valores perdidos en la columna 0 es del 10.16%. En la figura 11. Se puede ver que Filling Time Series respeta la estadística de la propia serie de tiempo obtenida por la estación meteorológica, incluso recobra algunos valores estacionales de la misma.

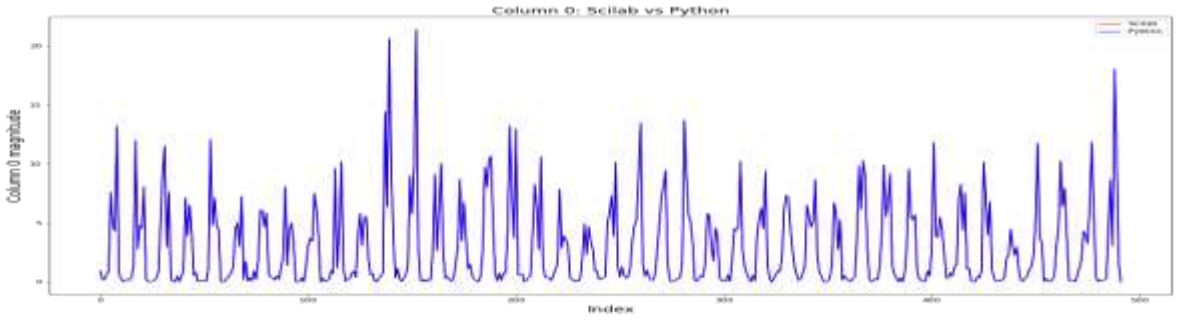

Figura 12. Comparación entre Filling Time Series y la rutina original en SCILAB en el método de autorregresión.

De igual forma, al hacer una comparación en el método autorregresivo de la rutina de SCILAB y el paquete de Filling Time Series en la columna 0 con los mismos parámetros, se obtiene una diferencia absoluta promedio de 0.002, siendo imperceptible alguna diferencia entre los dos códigos, tal y como se puede observar de la Figura 12.

IV.III. Método integrado

Los datos a rellenar en esta sección son los mismos utilizados en el método de componentes principales. Al ser rellenados con el método integrado, y haciendo una comparación con el método de componentes principales implementado en Filling Time Series, se obtiene la diferencia absoluta promedio de cada columna, siendo para la columna 0 de 8.96×10^{-6} mm, columna 1 de 9.78×10^{-7} mm, columna 2 de 9.72×10^{-6} mm, columna 3 de 7.99×10^{-6} mm, columna 4 de 1.53×10^{-5} mm y la columna 5 de 9.99×10^{-6} mm. Las diferencias son imperceptibles y esto se puede observar en la Figura 13.

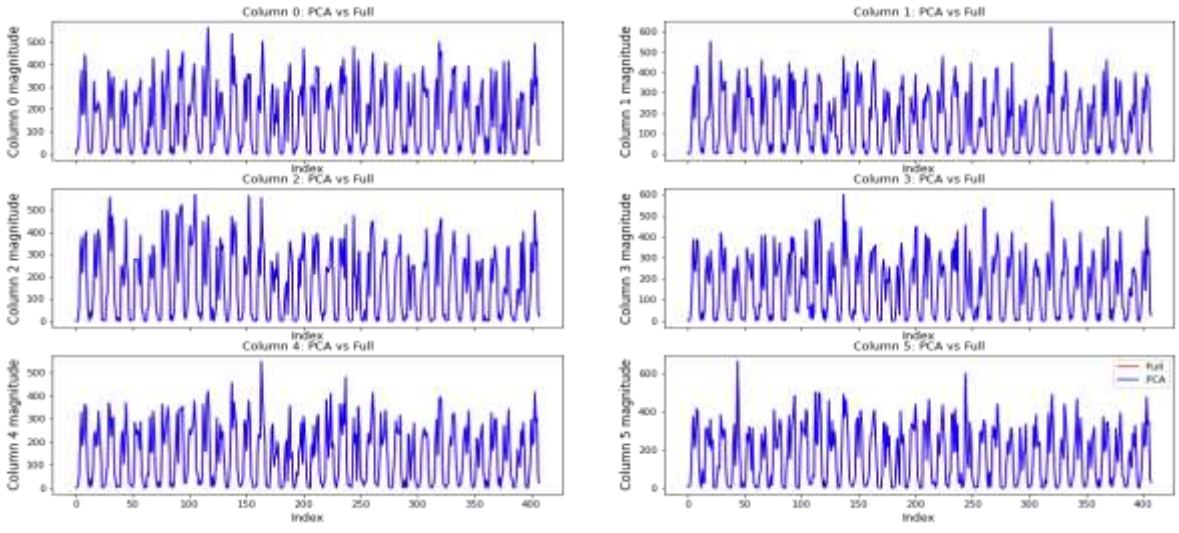

Figura 13. Comparación entre los métodos de componentes principales y el integrado.

V. Conclusiones

Filling Time Series logra implementar de forma efectiva el método de componentes principales, el método autorregresivo y el método integrado desarrollados en los trabajos de Alfaro y Soley (2009) y Ureña, Alfaro y Soley (2016), lo que permitirá a las personas usuarias que están habituadas a los ambientes basados en Python 3, poder rellenar las series de tiempo meteorológicas, mediante métodos avanzados que recobran información importante a pesar de que los datos originales se perdieron en el momento de la medición. Además, Filling Time Series ofrece una forma sencilla de aplicación de los métodos, mediante el uso del paquete de Python y de la aplicación de escritorio, siendo intuitiva para el usuario y teniendo un diseño atractivo para lograr una buena experiencia por parte del usuario. Por último, la creación del paquete de Python, Filling Time Series, fomenta la contribución de la comunidad científicas para introducir nuevas técnicas de rellanado de series de tiempo meteorológicas, para que el paquete posea una gran variedad de métodos para distintos casos de uso.

Agradecimientos

Se agradece a los proyectos VI-805-B0-810 / B8-604 / B9-454, al Dr. Hugo Hidalgo y al Dr. Eric Alfaro por su apoyo e incentivar el desarrollo de la implementación de los métodos de rellenado de series de tiempo en Python: Filling Time Series.

Referencias

Alfaro, E., & Soley, J. (2009). descripción de dos métodos de rellenado de datos ausentes en series de tiempo metereológicos. *Revista de matemática: Teoría y Aplicaciones, 16*, 60 - 75.

Harris, C., Millman, K., & van der Walt, S. (16 de Septiembre de 2020). Array programming with NumPy. *Nature, 585*, 357 - 362. doi:10.1038/s41586-020-2649-2

Hunter, J. (18 de Junio de 2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering, 9*, 90 - 95. doi:10.1109/MCSE.2007.55

North, G., Bell, T., Cahalan, R., & Moeng, F. (01 de Julio de 1982). Sampling errors in the estimation of empirical orthogonal functions. *Monthly Weather Review, 110*, 699 - 706. doi:10.1175/1520-0493(1982)110<0699:SEITEO>2.0.CO;2

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., & Grisel, O. (01 de Febrero de 2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825 - 2830. doi:10.5555/1953048.2078195

Reback, J., jbrockmendel, McKinney, W., Bossche, J. V., Augspurger, T., Cloud, P., . . . Garcia, M. (17 de Octubre de 2021). pandas - dev / pandas: Pandas. *Zenodo*. doi:10.5281/zenodo.3509134

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with Python. *Python in Science Conference, 9*. doi:10.25080/Majora-92bf1922-011

Ulrych, T., & Clayton, R. (Agosto de 1976). Time Series Modeling and Maximum Entropy. *Physics of the Earth and Planetary Interiors,, 12*, 188 - 200. doi:10.1016/0031-9201(76)90047-9

Ureña, P., Alfaro, E., & Soley, J. (2016). Propuestas metodológicas para el rellenado de datos ausentes en series de tiempo geofísicas. Guía Práctica de uso. *Universidad de Costa Rica*.

Wilks, D. (1995). *Statistical Methods in the Atmospheric Sciences.* New York: Academic Press.