

# Manual para el cálculo de estadísticas móviles a partir de series sintéticas

Alessio Bocco

19 April, 2021

## Contents

<b>1</b>	<b>Introducción</b>	<b>1</b>
<b>2</b>	<b>Agregación de series temporales</b>	<b>2</b>
<b>3</b>	<b>Ejemplo de aplicación</b>	<b>4</b>

El presente documento describe el procedimiento para el cálculo de las estadísticas móviles en alta frecuencia, cada 5 días, a partir de series climáticas sintéticas. Estas estadísticas corresponden a los totales acumulados de precipitación y a la media de temperaturas máxima y mínima agregada a distintas escalas temporales. Si bien se hará hincapié en el uso de series sintéticas, lo mostrado a continuación es también válido para series observadas ya que pueden ser consideradas como series sintéticas con una única realización.

El cálculo de índices de sequía está dividido en dos etapas: a) cálculo de los estadísticas mensuales de precipitación y temperatura y b) cálculo de los índices propiamente dichos.

## 1 Introducción

El insumo primario principal para el cálculo de índices de sequía son los valores de variables climáticas (precipitación, temperaturas máxima y mínima) simulados u observados a paso diario. Estos valores se agregan para las distintas escalas temporales definidas previamente. La agregación involucra (a) la suma de precipitaciones diarias y (b) los promedios de temperaturas máximas y mínimas diarias, para la escala temporal seleccionada. Los valores agregados o estadísticos para cada escala temporal se calculan para toda la serie sintética, repitiéndose el cálculo para cada péntada. Para el cálculo de valores agregados se implementó un script en lenguaje R (R Development Core Team, 2008) que procesa los datos diarios simulados de las tres variables definidas para todas las localidades (o puntos geográficos) y realizaciones generadas. Este script se ejecuta cada vez que un nuevo set de datos es simulado utilizando

el generador estocástico de datos meteorológicos. Si se realiza una nueva simulación con otra configuración del generador, ya sea para otras fechas o localidades, las estadísticas también deben recalcularse porque las nuevas series generadas no serán necesariamente iguales a las originales dado el carácter estocástico del generador. Una vez que ha finalizado el proceso de agregación, los datos obtenidos, de aquí en más llamados estadísticos, se guardan en una base de datos local. Dada una fecha de fin de una péntada (días 5, 10, 15, 20, 25 o fin del mes), el proceso se encarga de generar ventanas del ancho correspondiente a cada escala temporal y de computar los estadísticos previamente enumerados con los valores dentro de esa ventana. Luego, el proceso se mueve una péntada hacia adelante y se vuelve a realizar el mismo cómputo. Este proceso iterativo finaliza una vez que se completa la última péntada de la serie simulada. Además de los estadísticos, se guarda un registro de todas las tareas realizadas por el script. En este registro se detalla el nombre y origen de las series sintéticas para así asegurar la trazabilidad de los datos de inicio y evitar confusiones en el caso que se realicen múltiples simulaciones mediante el generador. Por último, notamos que el uso de series sintéticas tiene la ventaja que éstas no poseen datos faltantes por lo que no es necesario un procesamiento especial para el cálculo de los estadísticos.

## 2 Agregación de series temporales

El primer paso para el cálculo de los índices de sequía es la agregación de los datos diarios, ya sean series sintéticas u observadas, a escala mensual. En la implementación inicial – antes de la creación del SISSA – la agregación se realizaba por mes calendario. Por ejemplo, para calcular la precipitación acumulada, insumo del SPI, con una escala de 2 meses para Pehuajó en mayo de 2017, se considera la suma de los totales mensuales de lluvia en Pehuajó para abril y mayo de 2017 (2 meses). Esta lógica de cálculo se muestra en la fila (a) de la Figura 1 donde los datos de lluvia que se incluyen en el cálculo se muestran en amarillo. El uso de meses calendarios implica que los índices de sequía calculados a partir de estas series se actualizan una vez al mes, luego de terminado un mes calendario: el círculo naranja al final de un período indica cuándo está disponible el valor del índice (ignorando retrasos en el cálculo). Con el cálculo tradicional, el siguiente valor de un índice (Figura 1, fila b) recién se podría calcular a fines de junio de 2017, un mes después. En este caso, el valor del índice se calcula con la suma de totales de lluvia para mayo-junio 2017 (marcados con amarillo). Una actualización más frecuente de los índices permite tomar mejores decisiones ya que no es necesario esperar al cierre del mes en curso para obtener el valor de los índices. La nueva implementación para el cálculo frecuente de estadísticas permite tener un nuevo valor para un índice/escala cada 5 días aproximadamente.

La agregación frecuente de valores mensuales se basa en el concepto de péntadas (período de 5 días consecutivos). Sin embargo, esta definición es flexible para adaptarse a distintas circunstancias: (a) todos los meses tienen 6 péntadas y (b) las cinco primeras péntadas dentro de un mes tienen una duración de 5 días, en tanto que la sexta y última péntada de un mes puede tener 3, 4, 5 o 6 días – para meses con 28, 29, 30 o 31 días, respectivamente. Por ejemplo, la primera péntada de un mes incluye los días 1 a 5 de ese mes, la segunda péntada los días 6 a 10, y así sucesivamente. Una ventaja de esta definición es que las

péntadas no cruzan límites de meses, lo que hace fácil la agregación de datos a diferentes escalas. Por ejemplo, el cálculo de totales mensuales de precipitación basados es la suma de totales para las 6 péntadas en cada mes/año. Con este nuevo enfoque, las estadísticas mensuales se calculan para cada péntada del año. La escala de agregación de las series diarias, las futuras escalas de los índices, se definen en función de péntadas, y no de meses. Para continuar con el ejemplo anterior para Pehuajó, en el cálculo frecuente la agregación se realiza desde la primera péntada de abril de 2017 (1-5 de abril de 2017) hasta la sexta péntada del mes de mayo de 2017 (26-31 de mayo de 2017). (Figura 1, fila c); las 12 péntadas utilizadas para el cálculo se indican en amarillo. Este período de 12 péntadas o dos meses se denomina como “Abril-P1 a Mayo-P6”, donde P1 y P6 indican la primera y sexta década de un mes, respectivamente. Este intervalo coincide con dos meses calendario, por lo que el valor del índice para la fila (c) debe coincidir con el valor calculado en la fila (a), es decir con el cálculo para meses calendario.

A continuación, el período de 12 péntadas que corresponde a una escala de dos meses se corre una péntada hacia adelante (líneas punteadas azules en la Figura 1. ). Ahora el período utilizado (Figura 1, fila d) incluye las 12 péntadas entre la segunda péntada de Abril 2017 (Abril-P2) y la primera péntada de Junio 2017 (Junio-P1). En este período, el índice tiene la misma escala temporal que en la fila (c) pero ya no coincide con los meses calendario: comienza el 6 mayo 2017 y termina el 5 junio 2017. En la Figura 1. fila e, se vuelve a desplazar una péntada el período para el cual se calcula el SPI-2, y ahora el cálculo comprende el período entre la tercera péntada de abril y la segunda péntada de Junio 2017 (o sea entre el 11 de Abril y el 10 de Junio de 2017).

Como se ilustró antes, la ventaja más importante del cálculo por péntadas es que los diagnósticos de sequía están disponibles cada cinco días aproximadamente, o sea al final de cada péntada (círculos naranja en las filas c–e de la Figura 1). Es decir, se dispone de diagnósticos de las condiciones de sequía mucho más frecuentemente que antes (seis veces por mes, en lugar de una).

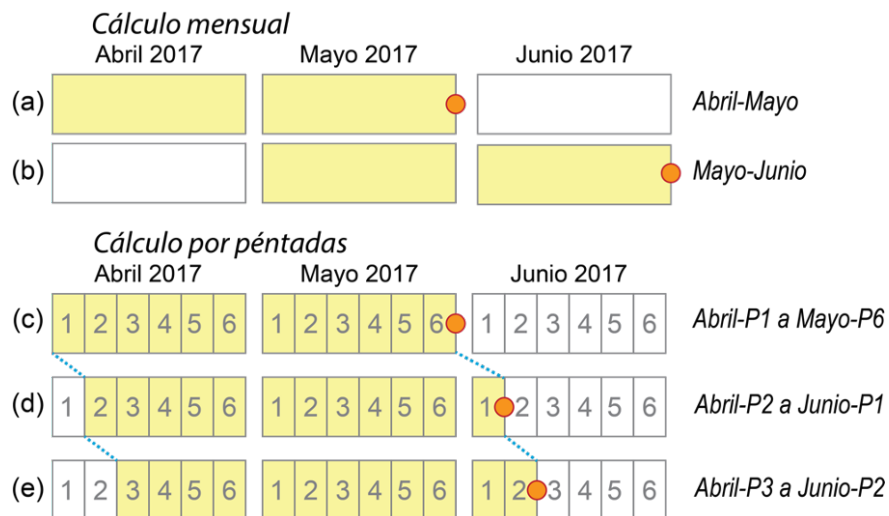


Figure 1: Ilustración de la agregación de series temporales por mes calendario (filas a y b) y por péntadas filas c-e).

### 3 Ejemplo de aplicación

Como se mencionó anteriormente, el primer paso para el cálculo de los índices es la agregación de los valores diarios a series mensuales en alta frecuencia. Para ello se debe correr el script `01_estadisticas_moviles.R`. No es necesario modificar nada en el código ya que éste es válido para la agregación de cualquier tipo de serie diaria de precipitación y temperaturas máxima y mínima. Solo se deben modificar los archivos `configuracion_estadisticas_moviles.yml` y `parametros_estadisticas_moviles.yml`. Estos archivos se pueden editar con cualquier editor de texto, incluso con el propio RStudio. A continuación se muestra la estructura de los archivos de texto y como modificarlos según cada usuario. El primero que modificaremos es `configuracion_estadisticas_moviles.yml`. Este archivo define las rutas a las distintas carpetas donde se almacenarán los resultados y la cantidad de máxima de procesos que pueden realizarse en paralelo.

```
# -----
# Directorios de trabajo
# -----
dir:
  base: /Users/alessiobocco/Documents/Documentos/SISSA/Devel/indices-eventos/Estadistica
  run: /Users/alessiobocco/Documents/Documentos/SISSA/Devel/indices-eventos/Estadisticas
  lib: /Users/alessiobocco/Documents/Documentos/SISSA/Devel/indices-eventos/lib/R/
  data: /Users/alessiobocco/Documents/Documentos/SISSA/Devel/indices-eventos/data/
# -----

# -----
# Cantidad maxima de procesos paralelos
# -----
max.procesos: 8
# -----
```

El primer bloque corresponde a las rutas. Se deben definir cuatro rutas:

- **base:** corresponde a la ruta donde se encuentra el proyecto.
- **run:** corresponde a la ruta donde se guardarán los logs de cada una de las tareas realizadas. Puede crearla manualmente como una subcarpeta dentro del directorio `base` o solo especificarla y el script la creará automáticamente. Siempre debe ser una subcarpeta dentro de `base`.
- **lib:** corresponde a la ruta donde se encuentran los scripts desarrollados por el SISSA. Esta carpeta no se encuentra junto al proyecto sino un nivel más arriba dentro de `indices-eventos`.
- **data:** corresponde a la ruta donde se encuentran los datos diarios de entrada para el cálculo de las estadísticas móviles. Esta carpeta no se encuentra junto al proyecto sino un nivel más arriba dentro de `indices-eventos`.

En el segundo bloque se deben definir la cantidad de procesos en paralelo que pueden llevarse adelante. La mayoría de las computadoras personales hoy poseen más de un núcleo lo que aumenta en gran medida el poder de cómputo. Se recomienda elegir una cantidad de núcleos tal que permita seguir utilizando la computadora con fluidez, en el caso que se trate de una computadora personal. Si se desconoce la cantidad de núcleos puede escribir lo siguiente en la consola.

```
require(parallel)
parallel::detectCores()
```

```
## [1] 8
```

La función `detectCores()` del paquete `parallel` permite identificar la cantidad de núcleos disponibles. En este caso son 8 pero se recomendaría usar 6 para así dejar dos disponibles para otras tareas.

El segundo archivo de texto que debe modificarse es `parametros_estadisticas_moviles.yml`. Este archivo define las características que tendrá el proceso de agregación desde el ancho de las ventanas (escalas de interés), pasando por la política de faltantes hasta las funciones de agregación.

```
# -----
# Definicion de ventana de calculo
# -----
# Este parametro define los ancho de la ventana de tiempo (en pentadas) para
# realizar la agregacion. Se utilizan anchos de ventana en relacion a las escalas
# a ser utilizadas para el calculo de índices de sequía, estas escalas son las
# siguientesL 1, 2, 3, 6, 9, 12, 18, 24, 36 y 48 MESES. 1 mes = 6 pentadas, por lo
# tanto, al pasar los meses (de las escalas), a pentadas, se obtienen los siguientes
# valores: 6, 12, 18, 36, 54, 72, 108, 144, 216 y 288.
ancho.ventana.pentadas: [18]
# -----

# -----
# Parametros relacionados con el numero aceptable de datos faltantes en un mes
# -----
# Temperatura maxima y minima: hay dos criterios para definir si el numero de dias
# faltantes permite calcular una agregacion. Primero, no puede haber mas de
# un numero total de dias faltantes en un mes. Segundo, no puede haber mas de X dias
# seguidos faltantes. Si alguna de estas dos condiciones falla, el valor resultante
# de la agregacion es NA. Para el caso de la precipitacion, solo aplica el primer
# criterio. Para los distintos anchos de ventana la politica de faltantes varia
# ya que la cantidad de datos disponibles es diferente.
faltantes:
```

```

- { ancho.ventana.pentadas: 6, politica: { tmax: { maximo: 5, consecutivos: 3 }, tmin:
- { ancho.ventana.pentadas: 18, politica: { tmax: { maximo: 15, consecutivos: 3 }, tmi
#-----

# -----
# Parametros relacionados con los estadisticos aplicables a cada variable
# -----
estadisticos:
  tmax:
    - { id: Media, funcion: mean, parametros: { na.rm: TRUE }, validable: TRUE }
    - { id: Mediana, funcion: median, parametros: { na.rm: TRUE }, validable: TRUE }
    - { id: DesviacionEstandar, funcion: sd, parametros: { na.rm: TRUE }, validable: TRUE }
    - { id: MAD, funcion: mad, parametros: { na.rm: TRUE }, validable: TRUE }
    - { id: NFaltantes, funcion: ContarFaltantes, parametros: { }, validable: FALSE }
    - { id: NDisponibles, funcion: ContarDisponibles, parametros: { }, validable: FALSE }
  tmin:
    - { id: Media, funcion: mean, parametros: { na.rm: TRUE }, validable: TRUE }
    - { id: Mediana, funcion: median, parametros: { na.rm: TRUE }, validable: TRUE }
    - { id: DesviacionEstandar, funcion: sd, parametros: { na.rm: TRUE }, validable: TRUE }
    - { id: MAD, funcion: mad, parametros: { na.rm: TRUE }, validable: TRUE }
    - { id: NFaltantes, funcion: ContarFaltantes, parametros: { }, validable: FALSE }
    - { id: NDisponibles, funcion: ContarDisponibles, parametros: { }, validable: FALSE }
  prcp:
    - { id: Suma, funcion: sum, parametros: { na.rm: TRUE }, validable: TRUE }
    - { id: Ocurrencia, funcion: CalcularOcurrencia, parametros: { umbral: 0.1 }, validable: TRUE }
    - { id: Media, funcion: mean, parametros: { na.rm: TRUE }, validable: TRUE }
    - { id: Mediana, funcion: median, parametros: { na.rm: TRUE }, validable: TRUE }
    - { id: DesviacionEstandar, funcion: sd, parametros: { na.rm: TRUE }, validable: TRUE }
    - { id: MAD, funcion: mad, parametros: { na.rm: TRUE }, validable: TRUE }
    - { id: NFaltantes, funcion: ContarFaltantes, parametros: { }, validable: FALSE }
    - { id: NDisponibles, funcion: ContarDisponibles, parametros: { }, validable: FALSE }
# -----

```

El primer bloque de este documento define las escalas de agregación de los valores diarios. Es decir, la cantidad de péntadas que tendrá la ventana de agregación. Se puede utilizar varias escalas con un mínimo de 6 (1 mes) hasta 288 (48 meses), sólo se deben separar los valores con comas tal como muestra el ejemplo.

El segundo bloque corresponde a las políticas de datos faltantes. Si los datos de entrada son series sintéticas, éstas no tendrán faltantes y el usuario puede obviar este bloque ya no tendrá ningún efecto. Si en cambio se utilizan series observadas se debe proceder de la siguiente manera. Por cada ancho de ventana y variable se debe definir una política. Por ejemplo, como se observa en la primera fila de este bloque, el ancho de ventana 6 tiene tres políticas, dos para temperaturas máxima y mínima y uno para precipitación. Las

temperaturas tienen dos argumentos, la cantidad máxima dentro de la ventana y la cantidad de faltantes consecutivos mientras que para la precipitación sólo se especifica una cantidad máxima de faltantes. Siguiendo con el ejemplo para el ancho de ventana 6, la temperatura máxima (“tmax”) solo permite un cantidad de máxima (“maximo”) de 5 faltantes con no más de 3 faltantes consecutivos (“consecutivos”). Si esta condición no se cumple, toda la ventana será reemplazada por NA. La temperatura mínima (“tmin”) tiene la misma política de faltantes. Por último, para la precipitación (“prcp”) sólo se especifica la cantidad máxima de faltantes tolerados, en este ejemplo, 3. Cabe mencionar que a medida que el ancho de ventana es mayor, la cantidad de faltantes máxima para cada una de las variables se flexibiliza. Esta política del ejemplo es solo a título demostrativo y el usuario puede definirla como mejor crea conveniente.

En último bloque del documento se definen las funciones de agregación para cada una de las variables junto a algunos diagnósticos como la cantidad de faltantes. Para el caso de las temperaturas máxima y mínima se calcularán las siguientes variables: \* Media: media de los valores de temperatura por ventana \* Mediana: mediana de los valores de temperatura por ventana \* Desvío estándar: desvío estándar de los valores de temperatura por ventana \* MAD: desvío absoluto medio de los valores de temperatura por ventana (Mean absolute deviation) \* NFaltantes: cantidad de faltantes por ventana \* NDisponibles: cantidad de datos disponibles por ventana

Para el caso de la precipitación las variables calculadas son:

- Suma: total acumulado por ventana
- Ocurrencia: cantidad de días lluviosos por ventana
- Media: media de la precipitación por ventana
- Mediana: mediana de la precipitación por ventana
- Desvío estándar: desvío estándar de la precipitación por ventana
- MAD:desvío absoluto medio por ventana (Mean absolute deviation)
- NFaltantes: cantidad de faltantes por ventana
- NDisponibles: cantidad de datos disponibles por ventana

Una vez completado los archivos de configuración y parámetros, sólo queda abrir el script y ejecutarlo. A modo de ejemplo se mostrarán a continuación algunos resultados. En la subcarpeta **data** dentro del directorio raíz **indices-eventos** se creará una carpeta llamada **parital**. Dentro de la misma se guardarán los resultados intermedios de todos los procesos.

```
estadisticas_moviles <- vroom::vroom('/Users/alessiobocco/Documents/Documentos/SISSA/Dev

## Rows: 8,474,000
## Columns: 9
## Delimiter: ","
## chr [2]: variable_id, estadistico
## dbl [5]: station_id, ancho_ventana_pentadas, realizacion, metodo_imputacion_id, valo
## date [2]: fecha_desde, fecha_hasta
```

```
##
## Use 'spec()' to retrieve the guessed column specification
## Pass a specification to the 'col_types' argument to quiet this message
```

A continuación, se muestran las primeras 10 filas de la tabla con los resultados del proceso de agregación. Las variables que posee la Tabla son las siguientes:

- **station\_id**: Código unívoco que identifica a la estación meteorológica
- **point\_id**: Código unívoco que representa a cada punto de la simulación. Esta variable cobra sentido cuando se simula sobre un grilla o en locaciones arbitrarias donde no hay un identificador previo para el punto. Si los datos a agregar solo corresponden a estaciones meteorológicas, esta columna podría descartarse.
- **variable\_id**: Identificador de las variables agregadas que corresponde a precipitación (prcp), temperaturas máxima (tmax) y mínima (tmin).
- **ancho\_ventana\_pentadas**: Cantidad de pñtadas usadas para la agregación. En el caso del ejemplo, 6 pñtadas (6 pñtadas x 5 días/pñtada = 30 días) quiere decir que los valores de las variables se agregan a una escala mensual. Este valor se determina en el archivo de configuración como se mostró más arriba.
- **fecha\_desde**: Fecha de comienzo de la ventana, es decir, primer día de la ventana.
- **fecha\_hasta**: Fecha de fin de la ventana de agregación, es decir, último día de la ventana.
- **realización**: Identificador de la realización. En el caso de que se trato de series observadas, la cantidad de realizaciones será una sola.
- **metodo\_imputacion\_id**: Código unívoco de identificación del método de imputación.
- **estadistico**: Nombre del estadístico utilizado para la agregación.
- **valor**: Valor del estadístico correspondiente a cada ventana/año.

station_id	variable_id	ancho_ventana_pentadas	fecha_desde	fecha_hasta	realizacion	metodo_imputacion_id	estadistico	valor
87448	tmax	6	1961-01-01	1961-01-31	1	0	DesviacionEstandar	3.373502
87448	tmax	6	1961-01-01	1961-01-31	1	0	MAD	2.371786
87448	tmax	6	1961-01-01	1961-01-31	1	0	Media	31.314426
87448	tmax	6	1961-01-01	1961-01-31	1	0	Mediana	30.917285
87448	tmax	6	1961-01-01	1961-01-31	1	0	NDisponibles	31.000000
87448	tmax	6	1961-01-01	1961-01-31	1	0	NFaltantes	0.000000
87448	tmax	6	1961-01-01	1961-01-31	2	0	DesviacionEstandar	3.262904
87448	tmax	6	1961-01-01	1961-01-31	2	0	MAD	2.475527
87448	tmax	6	1961-01-01	1961-01-31	2	0	Media	29.403774
87448	tmax	6	1961-01-01	1961-01-31	2	0	Mediana	28.818150

Como se observa, la tabla tiene un formato largo. Es decir, las variables y los estadísticos usados en el proceso de agregación se encuentran en las filas y no en las columnas. Para continuar con el proceso de cálculo de índices, éste es el formato que debe tener pero si se desea explorar los resultados en un formato más amigable se puede hacer lo siguiente.

Este dataset de ejemplo contiene 50 realizaciones para la estación meteorológica de Villa Reynolds (San Luis, Argentina) y agregada con dos ancho de ventana diferente: 6 pñtadas (1 mes) y 18 pñtadas (3 meses).



```
# Cantidad de realizaciones
unique(estadisticas_moviles$realizacion)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
```

```
# Ancho de ventana de agregación
unique(estadisticas_moviles$ancho_ventana_pentadas)
```

```
## [1] 6 18
```

Si el usuario desea utilizar los valores agregados puede transformar la tabla de formato largo a ancho. En el siguiente ejemplo se utilizan las medias mensuales de temperaturas máxima y mínima y los totales acumulados de precipitación.

```
# Filtrar estadísticos por cada variable:
# Suma de precipitación y media de temperaturas máxima y mínima.
estadisticas_moviles_ancho <- estadisticas_moviles %>%
  dplyr::filter((
    # Filtrar la temperatura máxima media
    (variable_id == 'tmax' & estadistico == 'Media') |
    # Filtrar la temperatura mínima media
    (variable_id == 'tmin' & estadistico == 'Media') |
    # Filtrar los totales de precipitación
    (variable_id == 'prcp' & estadistico == 'Suma'))) &
  # Filtrar los valores agregados a una ventana de 6 pentadas (30 días)
  ancho_ventana_pentadas == 6) %>%
  # Seleccionar las variables station_id, realizacion, variable_id, fecha_desde, fecha_hasta, valor
  dplyr::select(station_id, realizacion, variable_id, fecha_desde, fecha_hasta, valor) %>%
  # Convertir la tabla de formato largo a ancho
  tidyr::spread(variable_id, valor)

# Estructura de la tabla transformada
estadisticas_moviles_ancho %>%
  dplyr::slice(1:10) %>%
  knitr::kable() %>%
  kableExtra::kable_styling() %>%
  kableExtra::scroll_box(width = "100%", height = "200px")
```

station_id	realizacion	fecha_desde	fecha_hasta	precip	tmax	tmin
87448	1	1961-01-01	1961-01-31	100.74139	31.31443	15.06608
87448	1	1961-01-06	1961-02-05	99.22001	31.05173	14.67761
87448	1	1961-01-11	1961-02-10	108.87293	29.76217	14.58293
87448	1	1961-01-16	1961-02-15	54.53124	29.48730	14.08263
87448	1	1961-01-21	1961-02-20	39.02928	28.97840	13.89912
87448	1	1961-01-26	1961-02-25	39.02727	28.90377	13.72856
87448	1	1961-02-01	1961-02-28	36.87749	29.75551	13.89199
87448	1	1961-02-06	1961-03-05	53.46446	29.39177	14.31148
87448	1	1961-02-11	1961-03-10	63.40437	30.50988	14.00751
87448	1	1961-02-16	1961-03-15	43.67057	29.46385	14.21133

Para visualizar los efectos de la agregación en alta frecuencia se muestra un fragmento de 5 años de la tabla anterior. Como se mencionó, este dataset posee 50 realizaciones pero aquí solo se filtra una para optimizar la visualización.

```
highcharter::highchart() %>%
  highcharter::hc_add_series(data = estadisticas_moviles_ancho %>%
    dplyr::filter(realizacion == 1, fecha_desde < as.Date('19
    showInLegend = TRUE, name = "Temperatura máxima", zIndex =
    tooltip = list(valueSuffix = " [°C]", valueDecimals = 1),
    mapping = highcharter::hcaes(x = fecha_desde, y = tmax)) %>%
  highcharter::hc_add_series(data = estadisticas_moviles_ancho %>%
    dplyr::filter(realizacion == 1, fecha_desde < as.Date('19
    tooltip = list(valueSuffix = " [°C]", valueDecimals = 1),
    showInLegend = TRUE, name = "Temperatura mínima",
    mapping = highcharter::hcaes(x = fecha_desde, y = tmin)) %>%
  highcharter::hc_xAxis(type = 'datetime', title = list(text = "Fecha")) %>%
  highcharter::hc_yAxis_multiples(
    list(title = list(text = "Temperatura [°C]"), min = -10, max = 50),
    list(opposite = FALSE, title = list(text = " "))
  ) %>%
  highcharter::hc_chart(type = 'line', zoomType = 'x', panning = TRUE, panKey = 'shift')
  highcharter::hc_legend(enabled = TRUE, layout = "horizontal") %>%
  highcharter::hc_tooltip(shared = TRUE) %>%
  highcharter::hc_colors(c("#fc8d62", "#a6cee3")) %>%
  highcharter::hc_title(text = "Temperaturas máxima y mínima de Villa Reynolds (San Luis
  highcharter::hc_add_theme(highcharter::hc_theme_elementary()) %>%
  highcharter::hc_plotOptions(
    series = list(
      marker = list(
        enabled = TRUE,
        symbol = 'diamond',
        radius = 5,
```

```

        lineWidth = 0,
        states = list(
          hover = list(
            enabled = F
          )
        )
      ),
    ),
    column = list(
      borderWidth = 0
    )
  )
)

```

```

## Registered S3 method overwritten by 'quantmod':
##   method      from
## as.zoo.data.frame zoo

```

## Temperaturas máxima y mínima de Villa Reynolds (San Luis, Argentina)

