EASC2410 Lecture 22

Introduction to Time Series Analysis 2: Time Series Decomposition using Python

Dr. Binzheng Zhang
Department of Earth Sciences

Review of Lecture 21:

- learn basic concepts of data types
  - Section data
  - Time series data
  - Panel data
- the time domain versus frequency domain
- spectrum and power spectrum (periodogram)
- how to generate a periodogram using Python

In Lecture 22, you will learn:

- Internal structures of time series
  - Trend
  - Seasonal
  - Cyclic
  - Random/Unexpected
- how to degenerate a time series using Python
  - moving average method
  - tsa module

Recall: What is a time series

Definition

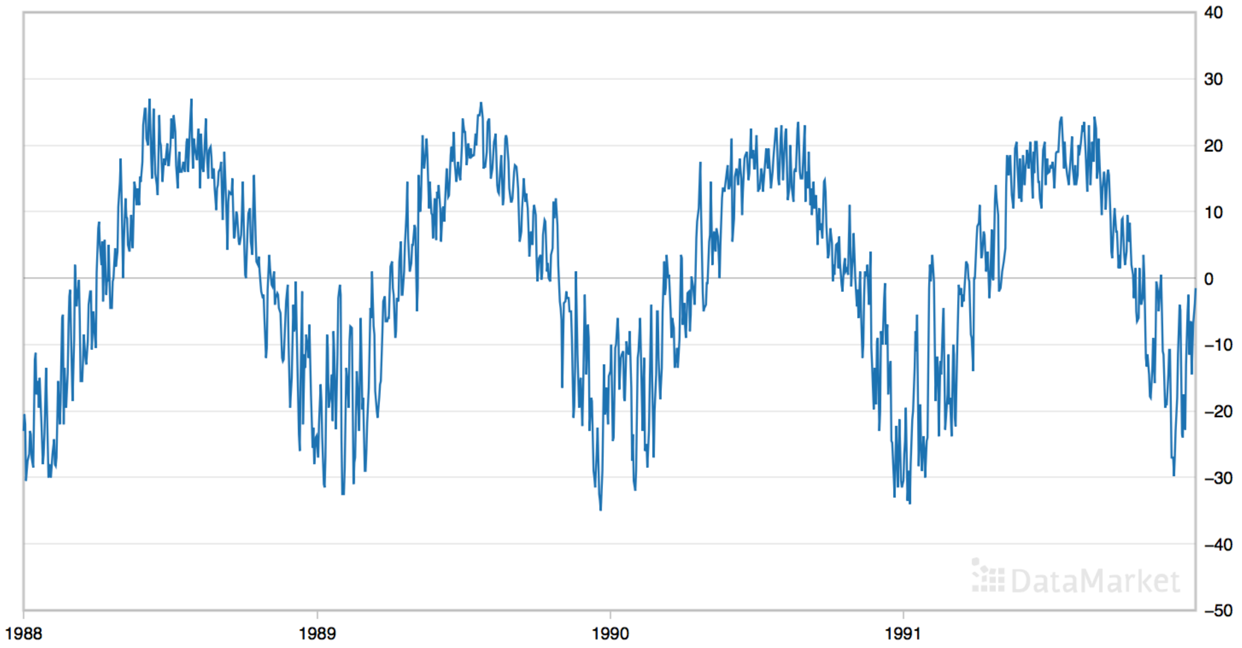
- The data is a time series in the form of a sequence of quantitative observations about a system or process and made at successive points in time.
- Usually time series data is equally spaced in time (if not, then resamplings are needed)

Examples

- gross domestic product versus year
- sales volumes versus seasons
- stock prices versus day
- weather attributes versus hours

Mean daily temperature, Fisher River near Dallas, Jan 01, 1988 to Dec 31, 1991

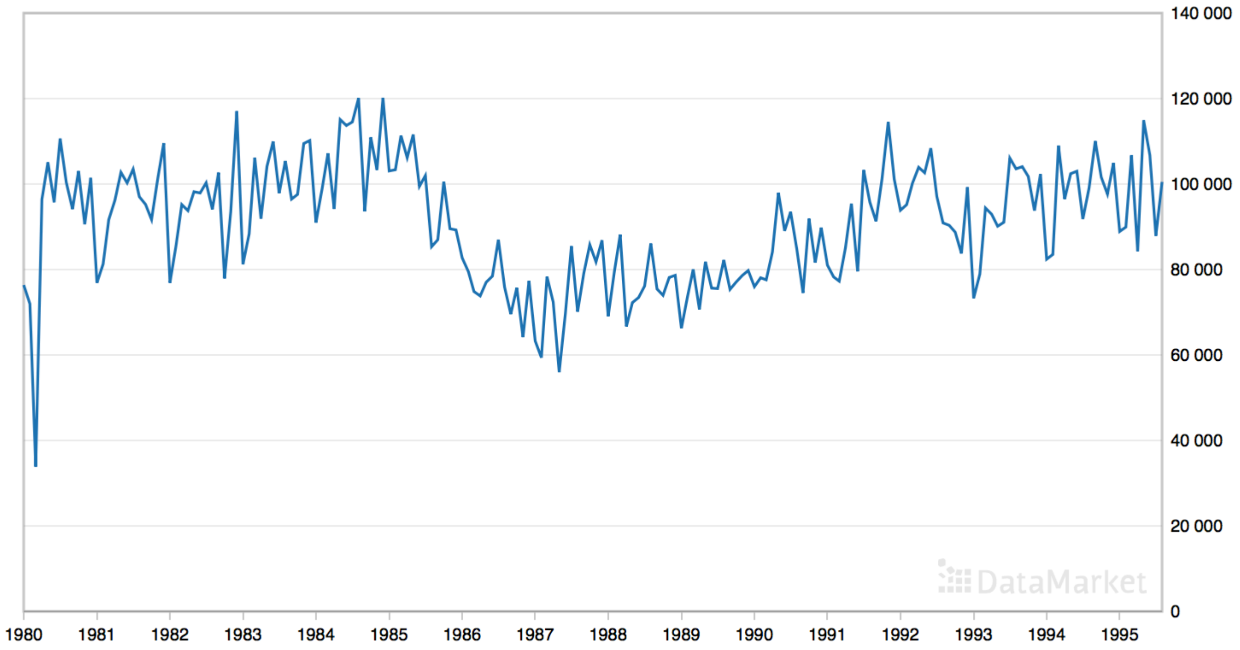
Units: Degrees Celsius



Source: Time Series Data Library (citing: Hipel and McLeod (1994))

Monthly total number of pigs slaughtered in Victoria. Jan 1980 – August 1995

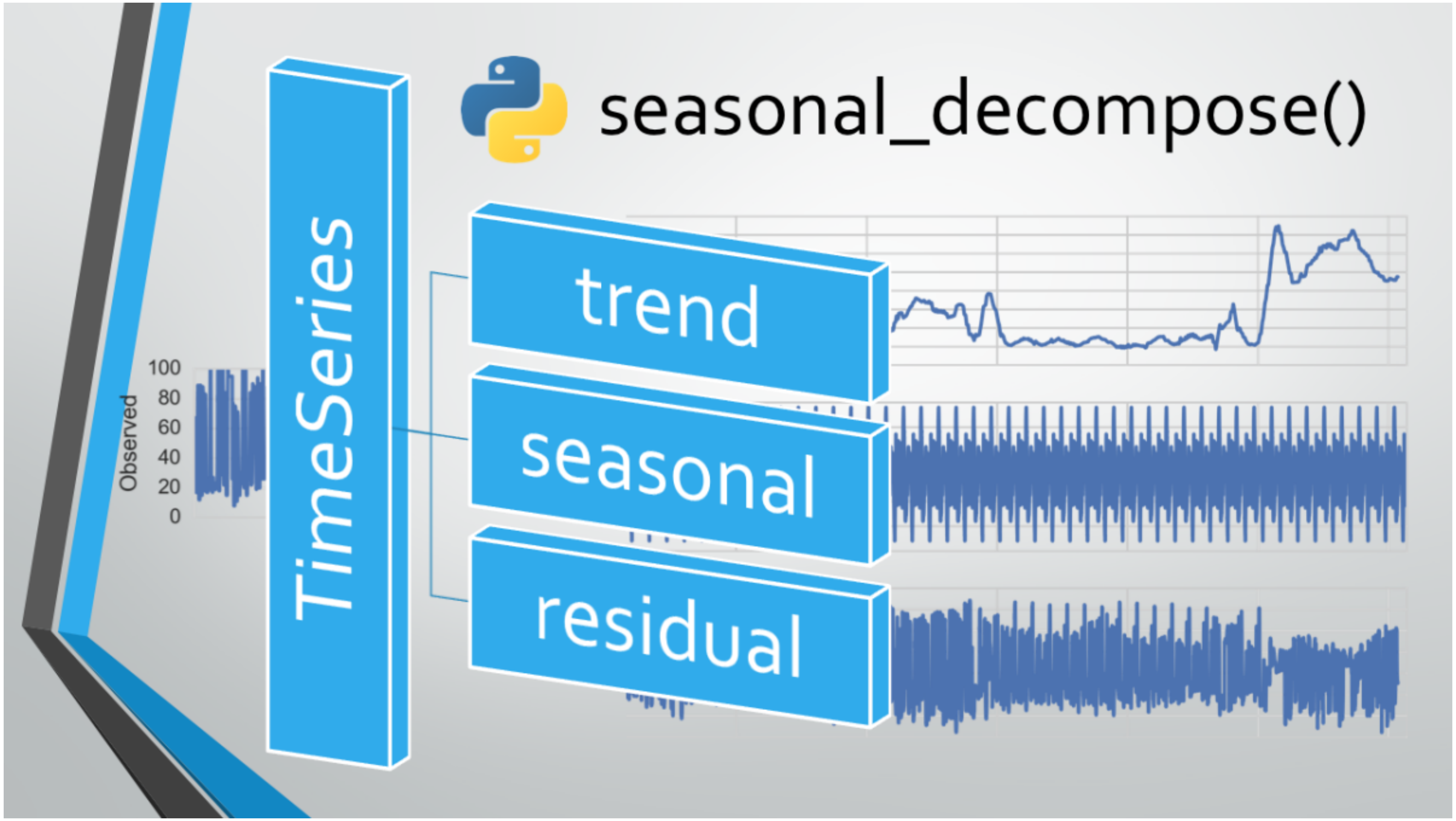
Units: Number of pigs



Source: Time Series Data Library (citing: Australian Bureau of Statistics)

Why is a time series analysis useful?

Time series analysis applies different statistical methods to explore and model the **internal structures of the time series data** such as periodicity, trends, seasonal fluctuations, cyclical behavior, and random/irregular/unexpected changes.



Internal structures of a time series

Typical Characteristics of time series data y(t) that requires its special mathematical treatment:

- General trend
- Seasonality
- Cyclical movements
- Unexpected (random) variations

Most time series has of one or more of the aforementioned internal structures. Based on this notion, a time series can be expressed as y(t) = T(t) + S(t) + C(t) + R(t), which is a sum of the **trend**, **seasonal**, **cyclical**, and **random** components in that order. Here, t is the time index at which observations about the series have been taken at t = 1,2,3 ...N successive and equally spaced points in time.

As you have learned from last lecture, one objective of time series analysis is to understand the frequency properties (spectrum and power spectrum, aka periodogram), which goes from time domain to frequency domain. SciPy provides a bunch of nice tools for doing the frequency analysis.

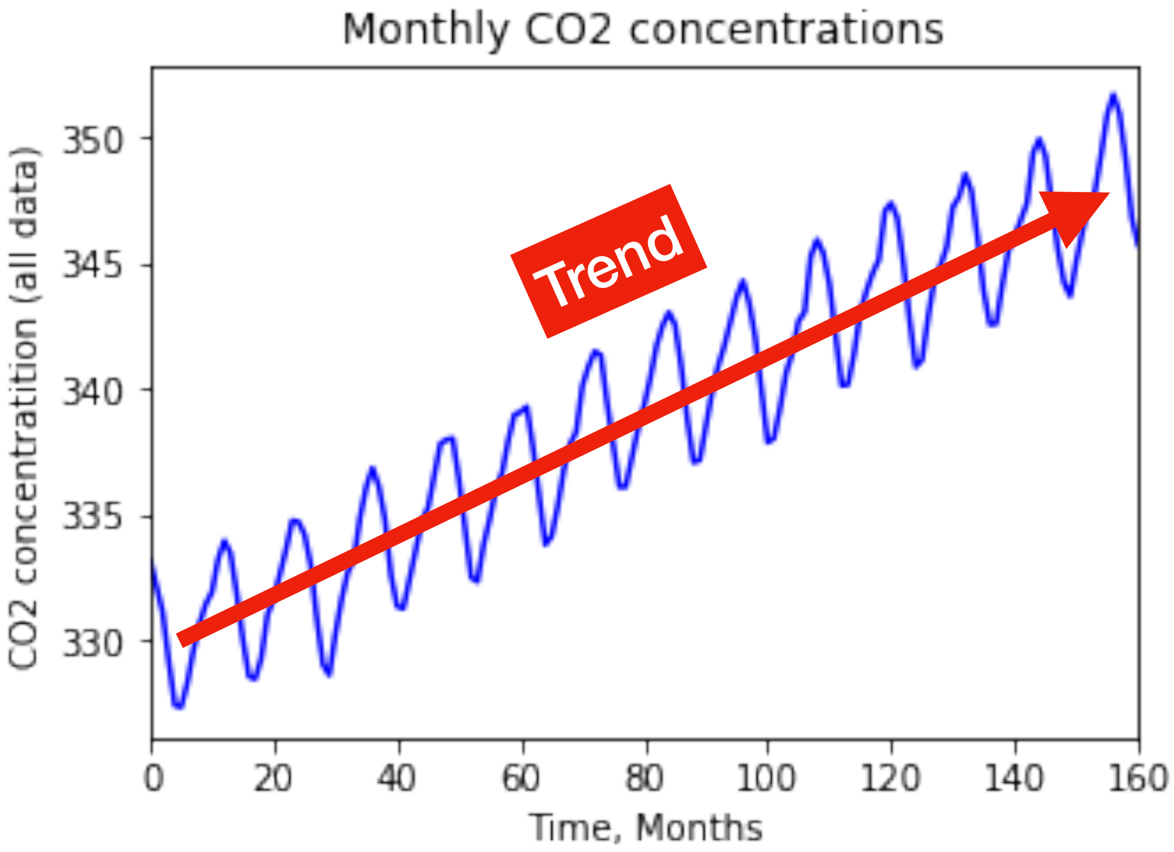
Another **objective** of time series analysis is to decompose a time series into its constituent characteristics and develop mathematical models for each. This is related to the frequency domain analysis but all results are shown in the time domain - the frequency analysis is under the hood (you don't need to worry about the frequency analysis, but you need to know what it is). These models are then used to understand what causes the observed behavior of the time series and to predict the series for future points in time.

General Trend of a time series

Definition:

When a time series exhibits an upward or downward movement in the long run, it is said to have a general trend.

Example:



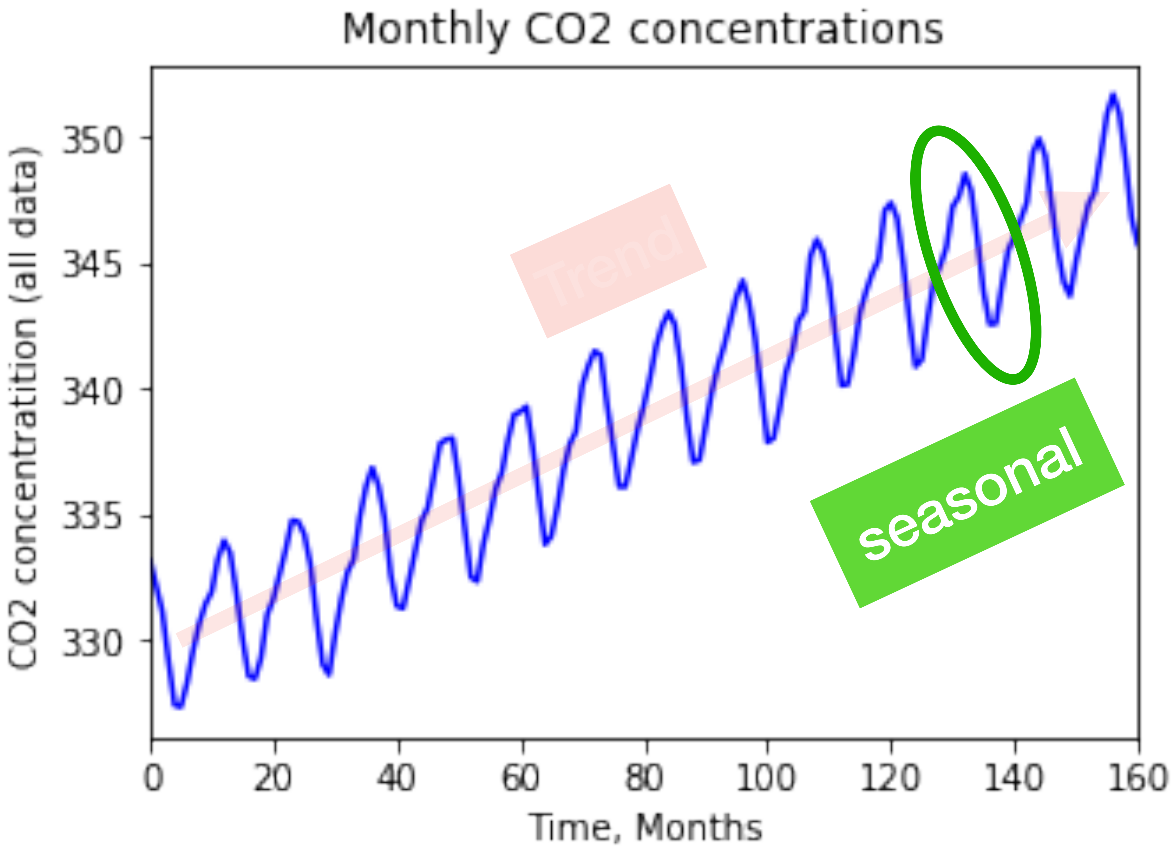
In the CO2 concentration data you've worked on yesterday (periodograms), there is a long-term (more than 12-months periodicity) increase in the CO2 concentration as the data curve goes upward in general. This is called the "trend" of a time series

Seasonality of a time series

Definition:

Seasonality manifests as **repetitive** and **period** variations in a time series. It is called seasonal but is not necessary with respect to the actually “seasons”. Seasonality is *regular*, which means that any seasonal variations in a time series is predictable - you know exactly when a peak or a dip occurs with respect to time - just like seasons.

Example:



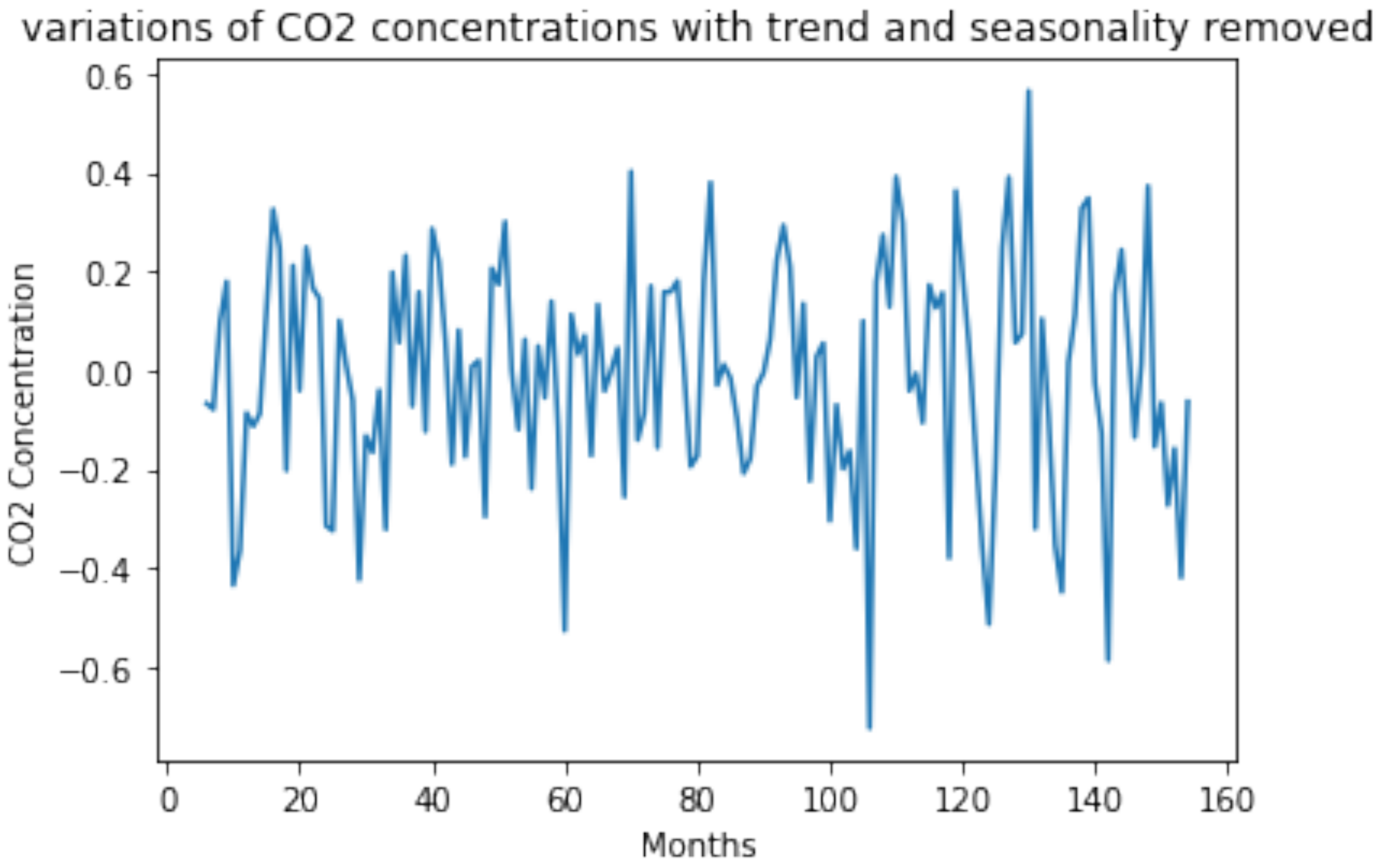
In the CO2 concentration data you’ve worked on yesterday (periodograms), there is a long-term (more than 12-months periodicity) increase in the CO2 concentration as the data curve goes upward in general. This is called the “trend” of a time series

Cyclic variations of a time series

Definition:

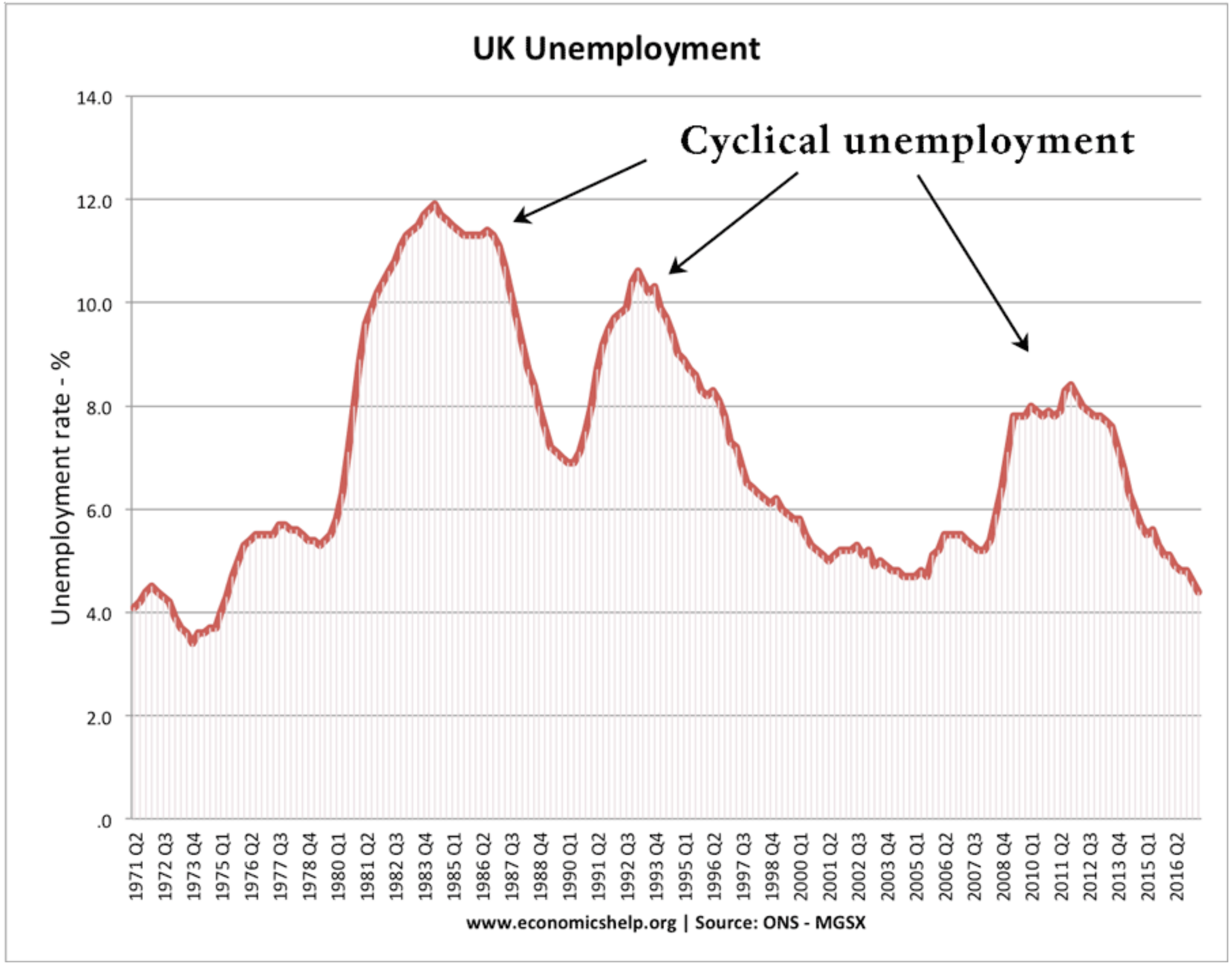
Cyclical changes are movements observed after every few units of time, but they occur less frequently than seasonal fluctuations. **Unlike seasonality, cyclical changes might not have a fixed period of variations.**

Example:



In the CO2 concentration data you've worked on yesterday (periodograms), there is a short-term (less than 12 months periodicity) in the CO2 concentration as the data curve goes up and down in an un-predictable way. This is called the "cyclic" variation of a time series

Cyclic variations of a time series

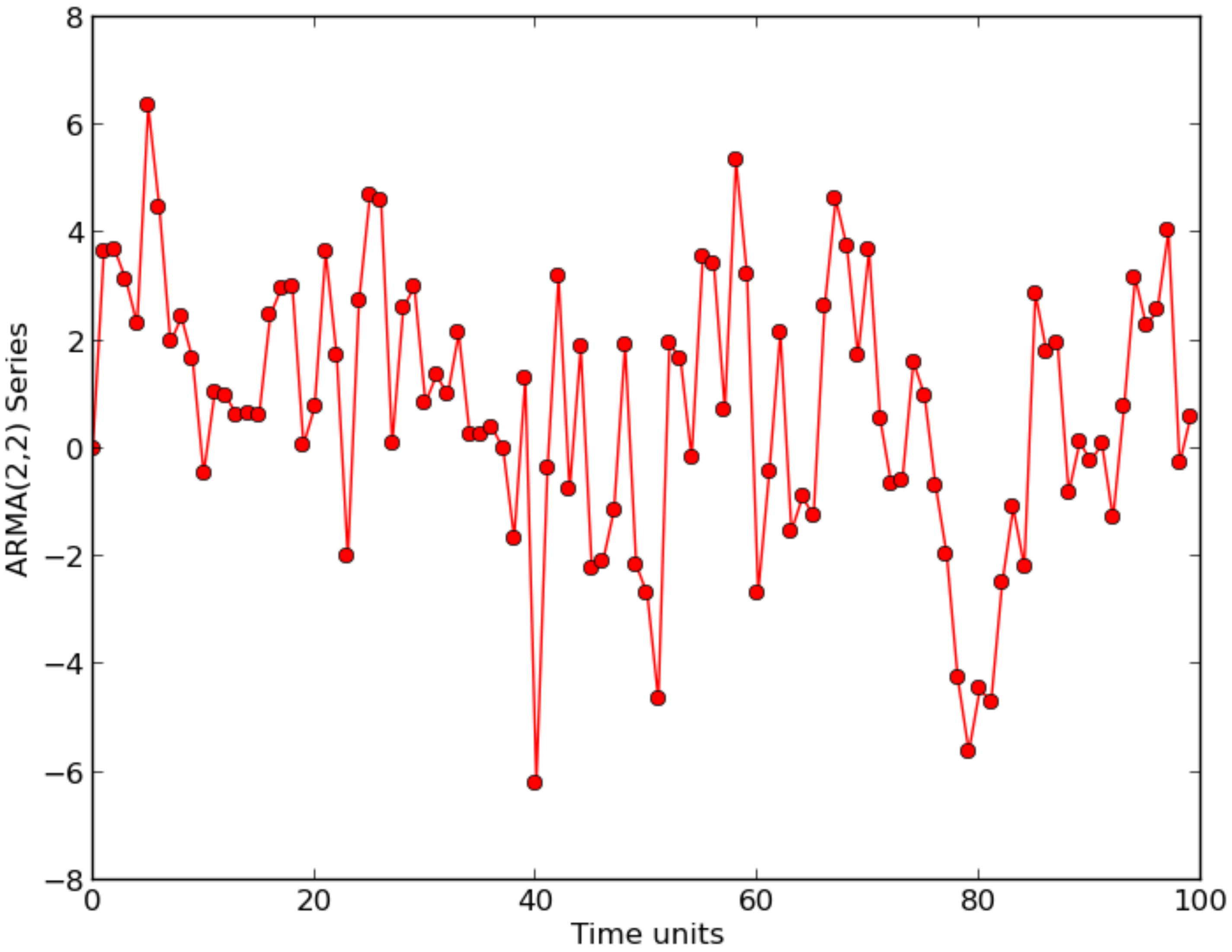


www.economicshelp.org | Source: ONS - MGSX

The difference between “seasonal” variations and “cyclic” variations of a data set is clear: fixed periodicity or not

Random/irregular/unexpected variations of a time series

Referring to our model that expresses a time series as a sum of four components, it is noteworthy that in spite of being able to account for the three other components, we might still be left with an irreducible error component that is random and does not exhibit systematic dependency on the time index. This fourth component reflects unexpected variations in the time series. Unexpected variations are stochastic and cannot be framed in a mathematical model for a definitive future prediction. This type of error is due to lack of information about explanatory variables that can model these variations or due to presence of a random noise.



When *trend* and *seasonal variations* are removed from a set of time series data, the ***residual*** left, which may or may not be random. Various techniques for analyzing series of this type examine to see "if irregular variation may be explained in terms of probability models such as *moving average* or *autoregressive models*, i.e. we can see if any *cyclical variation* is still left in the *residuals.*These variation occur due to sudden causes are called *residual variation* (*irregular variation* or *accidental* or *erratic fluctuations*) and are unpredictable, for example rise in prices of steel due to strike in the factory, accident due to failure of break, flood, earth quick, war etc.

Decompose a time series using Python

The objective of time series decomposition is to model the long-term trend and seasonality and estimate the overall time series as a combination of them. Two popular models for time series decomposition are

Additive model

The additive model formulates the original time series a time series y(t) as the sum of trend T(t), seasonality S(t), and residual variation R(t) components as:

$$y(t) = T(t) + S(t) + R(t)$$

R(t) here is called the "residual" of the time series obtained after subtracting the trend and seasonal components from the original time series. This "residual" part usually include random and/or cyclic changes, and the analysis on R(t) is **problem-dependent**

Multiplicative model

The multiplicative decomposition model, which gives the time series as product of the trend, seasonal, and irregular components is useful when there is time-varying seasonality:

$$y(t) = T(t) \times S(t) \times R(t)$$

In the following part, we will discuss the following two popular methods for estimating the trend and seasonal components:

- Trend modeling using Moving Averages (**pandas**)
- Seasonal and Trend Decomposition using the Python package (**statsmodels.tsa**)

Decompose a time series: Moving Average Method

Moving averages (MA) at a time index t estimates the average trend component T(t) and is calculated by taking average of over the time period of $t \pm k$ where k is the range of the MA:

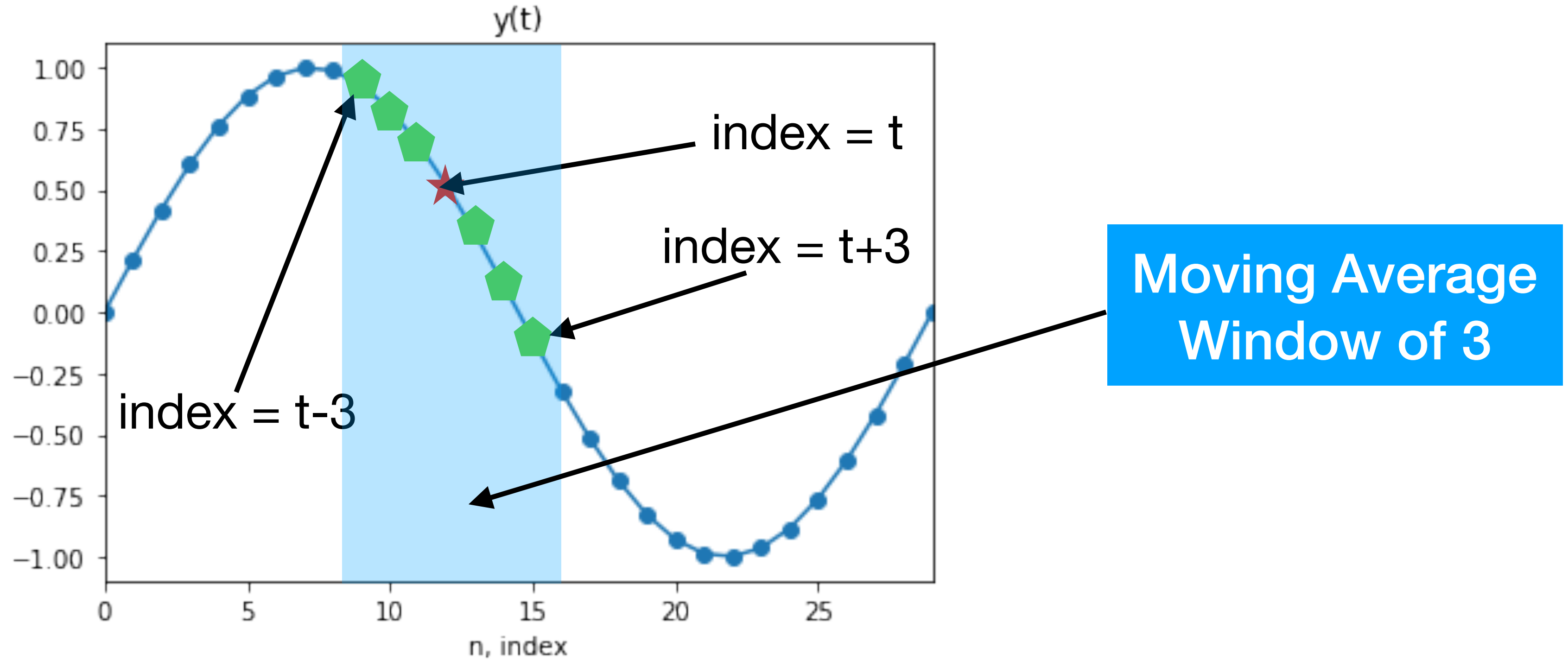
$$T(t)^{[k]} = \frac{1}{2k+1}\left[y(t-k) + y(t-k+1) + \ldots + y(t) + y(t+1) + \ldots + y(t+k-1) + y(t+k)\right]$$

For example, a Moving average for k = 3 is

$$T(t)^{[3]} = \frac{1}{2k+1}\left[y(t-3) + y(t-2) + y(t-1) + y(t) + y(t+1) + y(t+2) + y(t+3)\right]$$

Taking moving averages have an effect of smoothing the original time series by eliminating random noise. Commonly the total number of observations m = 2k + 1 is used to describe the moving average as m-order MA, which henceforth will be denoted as $T(t)^{[k]}$

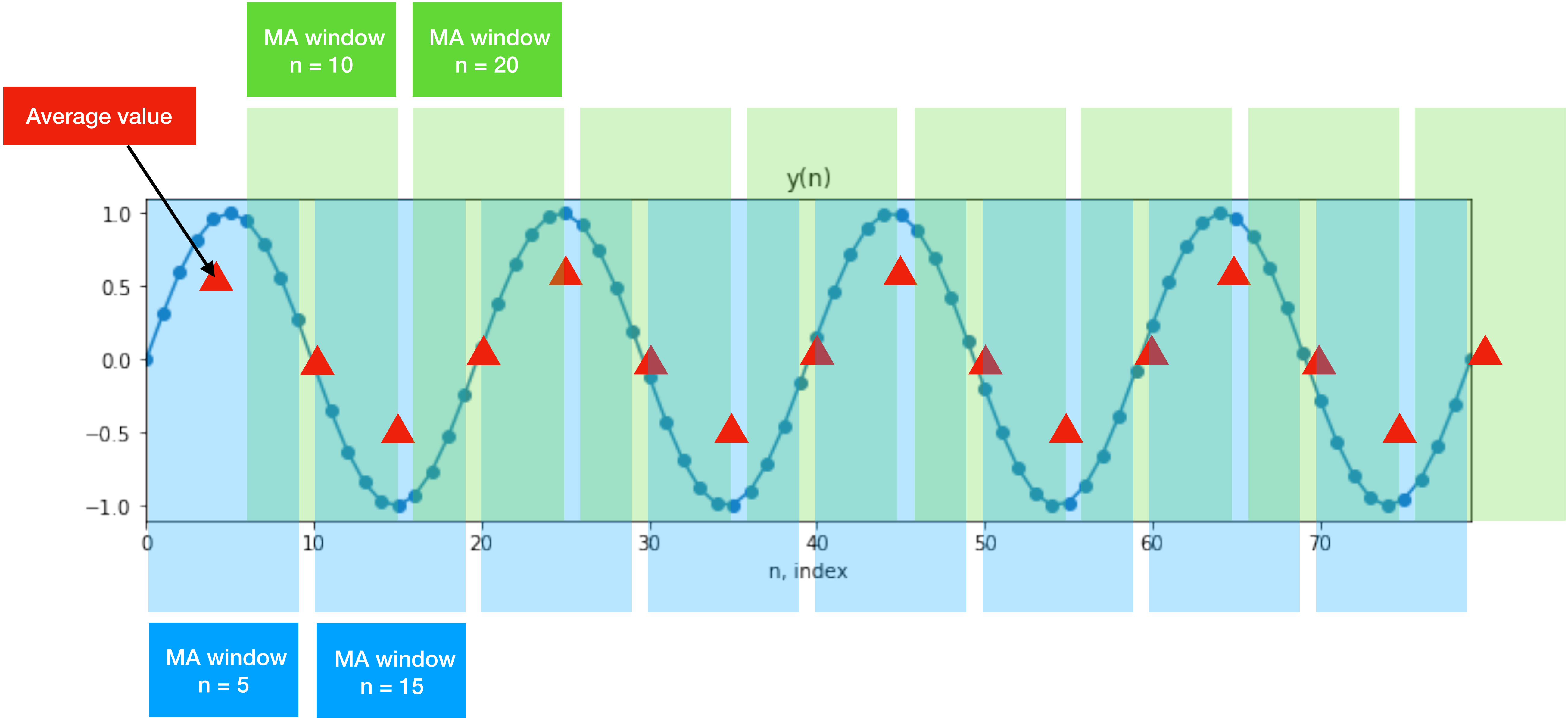
Here's what a moving average window of 3 (k=3) looks like graphically:



Decompose a time series: Moving Average Method

So what's the effect of a moving average window? it smooths out variations with periodicity smaller than the moving average window.

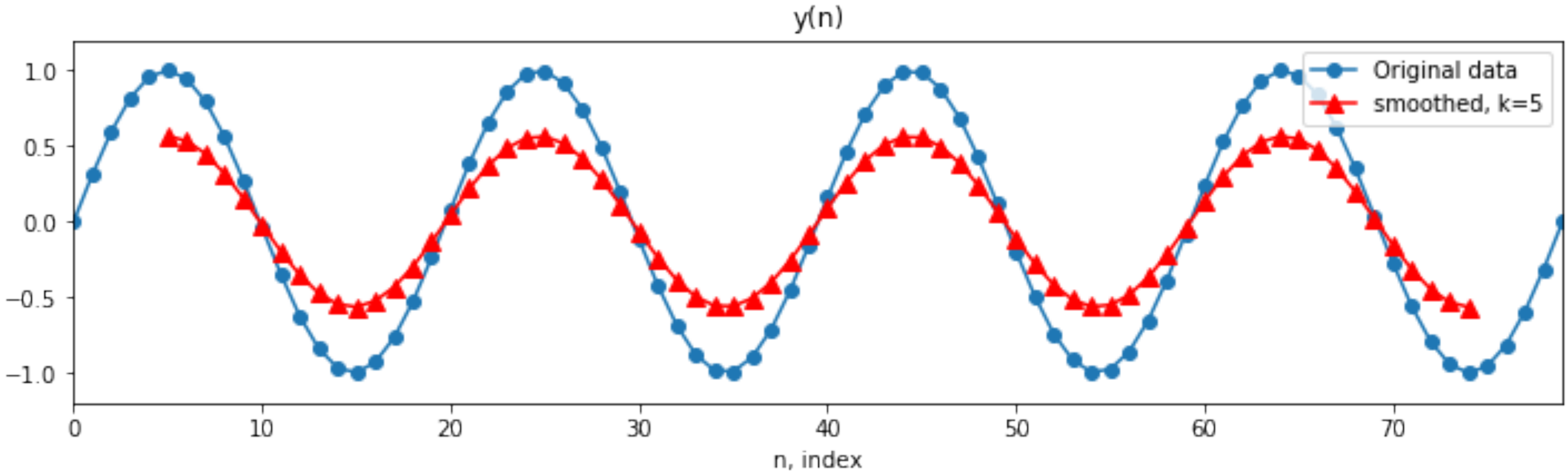
let's take a look at the following example with a MA window of 9 (k=4)



Decompose a time series: Moving Average Method

So what's the effect of a moving average window? it smooths out variations with periodicity smaller than the moving average window.

let's take a look at the following example with a MA window of 9 (k=4)

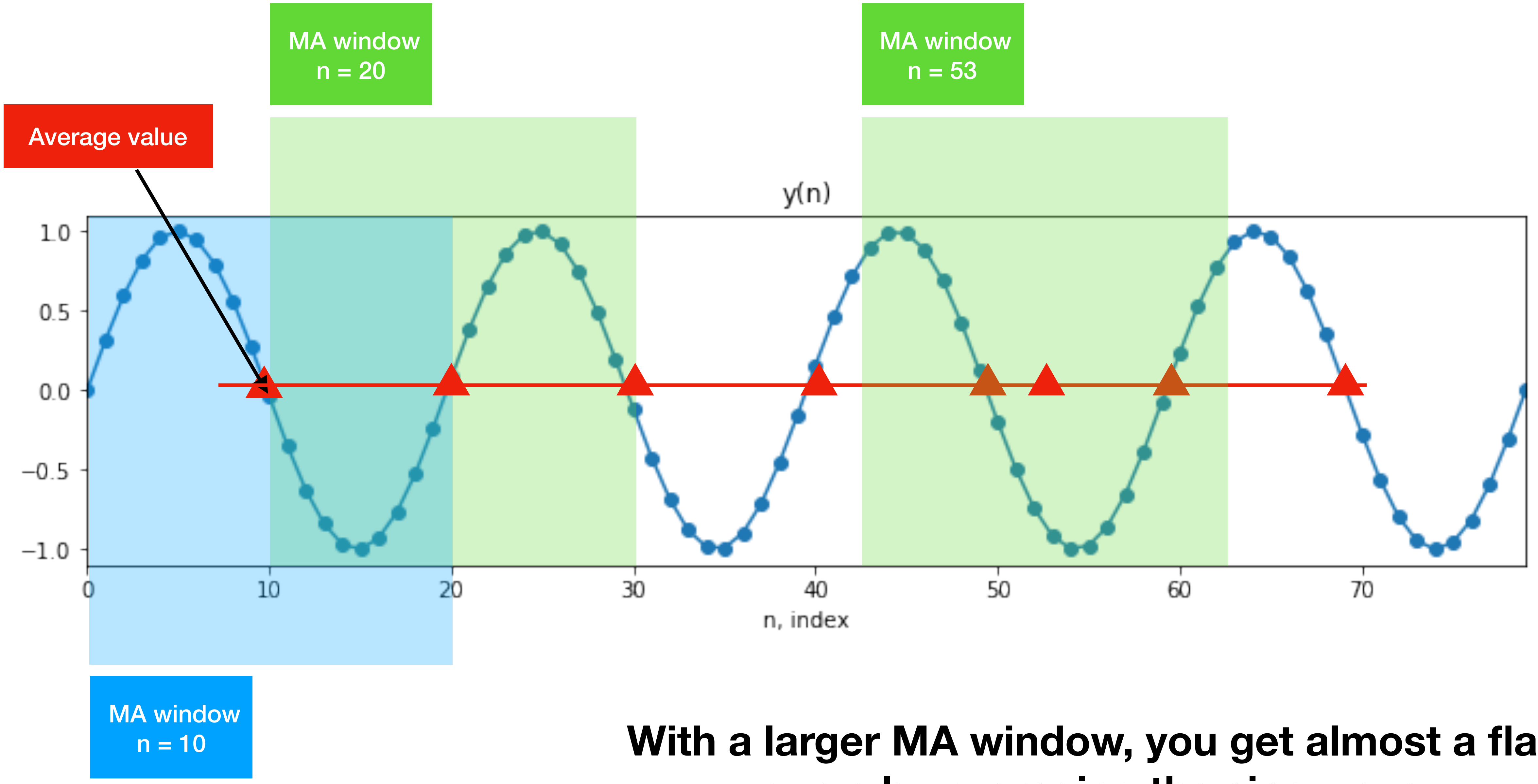


What you found here is that the k=5 moving average smooth window reduces the peak of the variations But it sounds pointless - the wave shape is not significantly changed after the moving average. Why is that?

Because the window is too small!

Decompose a time series: Moving Average Method

let’s take a look at the same example with a MA window of 9 (k=20)

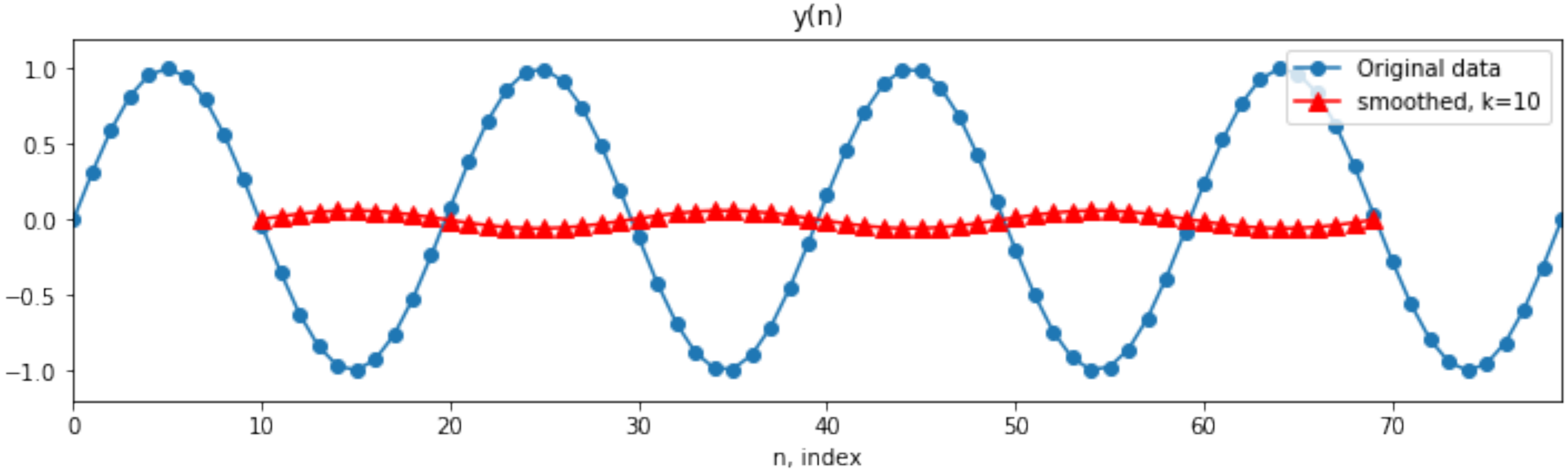


With a larger MA window, you get almost a flat curve by averaging the sine wave why?

Decompose a time series: Moving Average Method

So what’s the effect of a much larger moving average window?

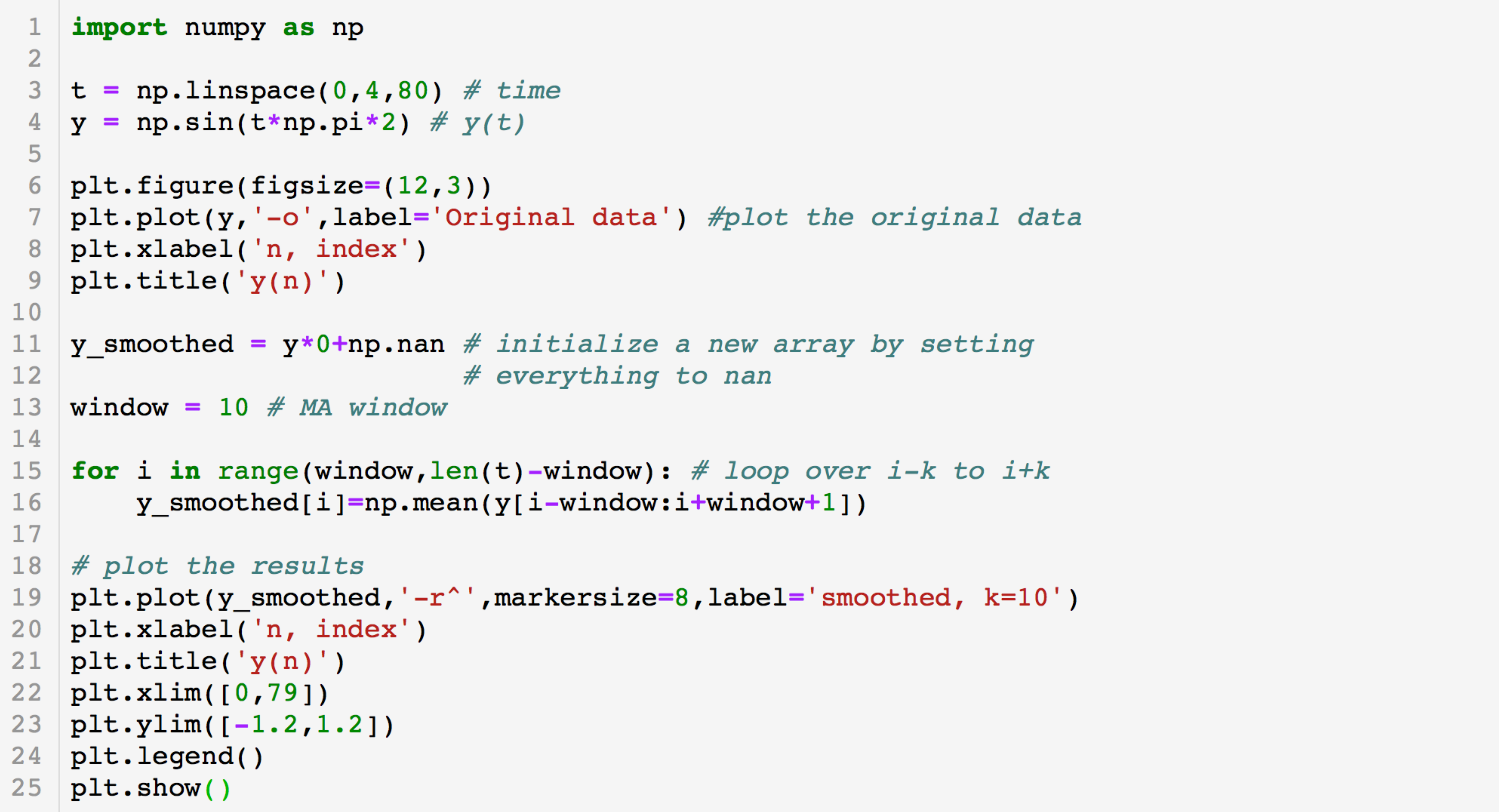
let’s take a look at the following example with a MA window of 21 (k=10). In this case, the MA window is k=10, which involves 21 neighourboring cells (approximately the periodicity of the signal)



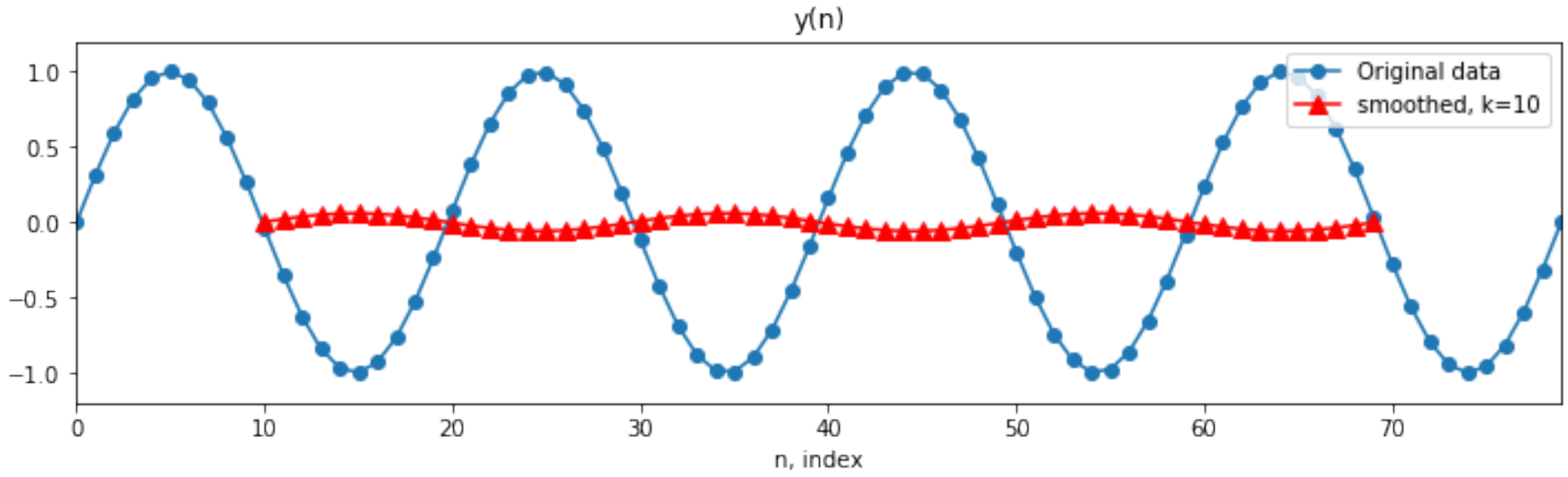
What you found here is that the k=10 moving average smooth window reduces almost all the peaks of the variations - any variations **within** 20 data points are significantly reduced. Here the red curve is basically the **trend**.

Question: For a general data curve dominated by trend and seasonality, how to choose the MA window?

How to do the Moving Average using NumPy



```
import numpy as np

t = np.linspace(0,4,80) # time
y = np.sin(t*np.pi*2) # y(t)

plt.figure(figsize=(12,3))
plt.plot(y,'-o',label='Original data') #plot the original data
plt.xlabel('n, index')
plt.title('y(n)')

y_smoothed = y*0+np.nan # initialize a new array by setting
                        # everything to nan
window = 10 # MA window

for i in range(window,len(t)-window): # loop over i-k to i+k
    y_smoothed[i]=np.mean(y[i-window:i+window+1])

# plot the results
plt.plot(y_smoothed,'-r^',markersize=8,label='smoothed, k=10')
plt.xlabel('n, index')
plt.title('y(n)')
plt.xlim([0,79])
plt.ylim([-1.2,1.2])
plt.legend()
plt.show()
```



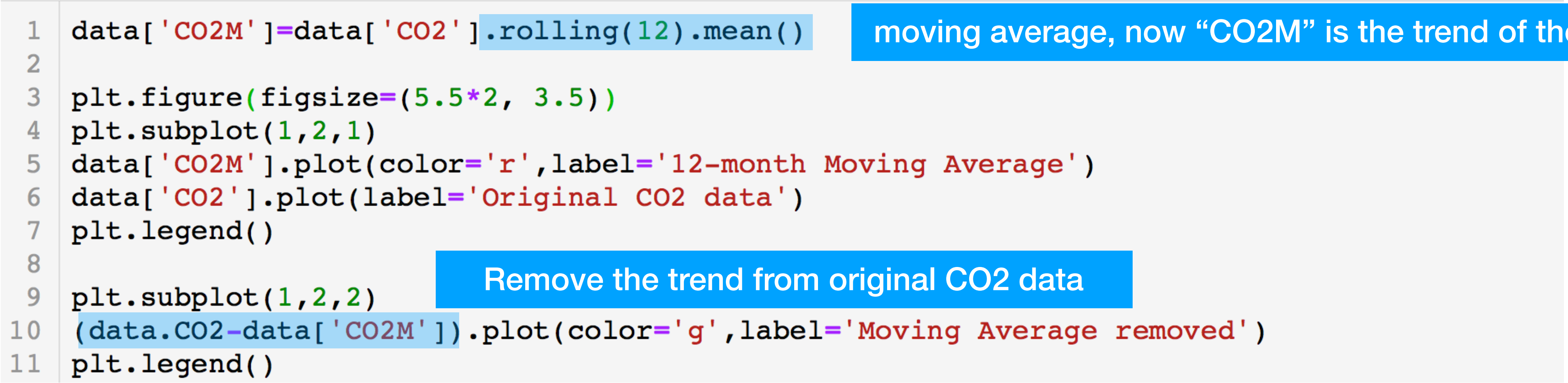
How to do the Moving Average using Pandas

In Pandas, you can use the **rolling()** function to compute a moving average in a very straightforward way

`DataFrame.rolling`(*window*, *min_periods=None*, *center=False*, *win_type=None*, *on=None*, *axis=0*, *closed=None*)

Here's the documentation page of the rolling() function:

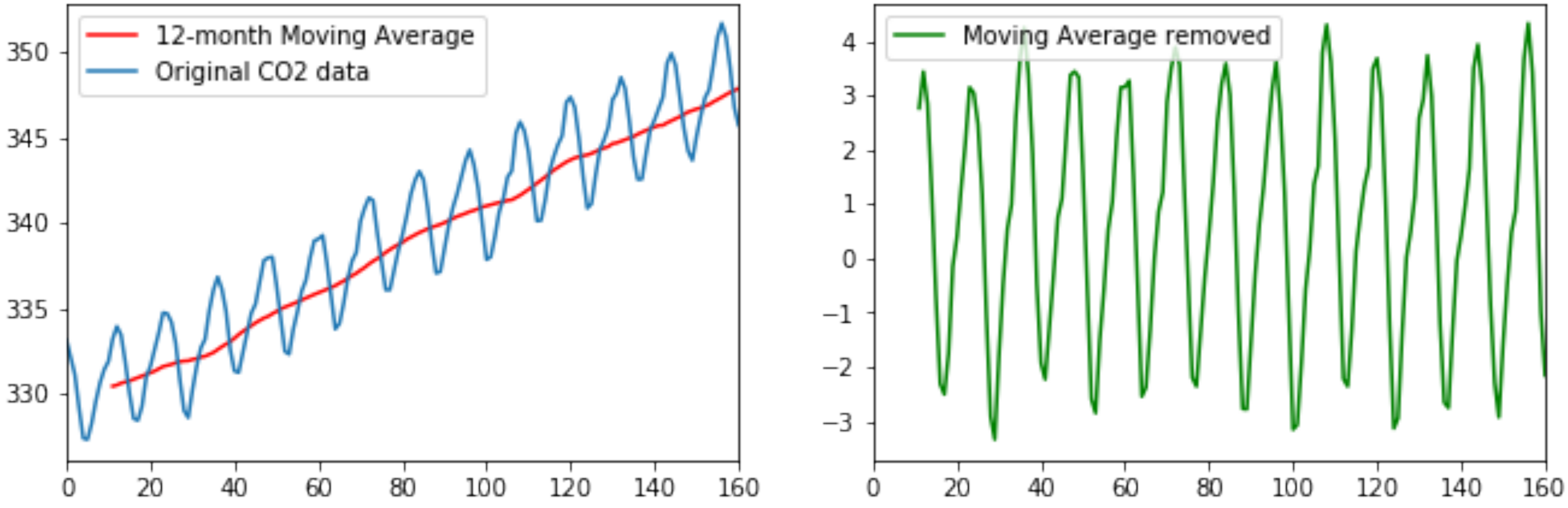
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.rolling.html



```python
data['CO2M']=data['CO2'].rolling(12).mean()

plt.figure(figsize=(5.5*2, 3.5))
plt.subplot(1,2,1)
data['CO2M'].plot(color='r',label='12-month Moving Average')
data['CO2'].plot(label='Original CO2 data')
plt.legend()

plt.subplot(1,2,2)
(data.CO2-data['CO2M']).plot(color='g',label='Moving Average removed')
plt.legend()
```

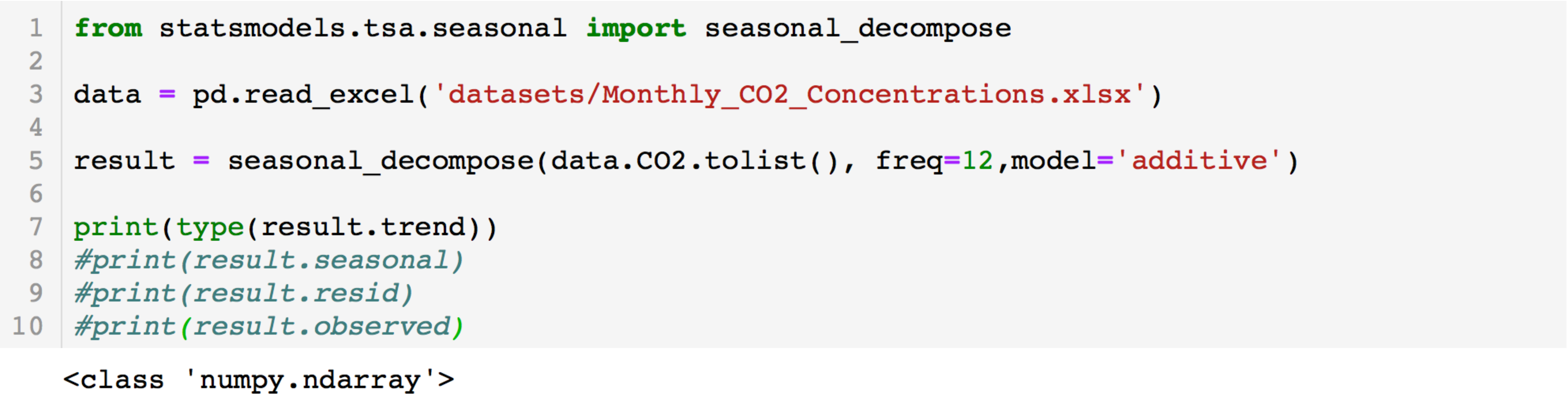
moving average, now "CO2M" is the trend of the CO2 data

Remove the trend from original CO2 data



Using the statsmodels.tsa module

The statsmodels library provides an implementation of the naive, or classical, decomposition method in a function called **seasonal_decompose()**. It requires that you specify whether the model is additive or multiplicative.



```python
from statsmodels.tsa.seasonal import seasonal_decompose

data = pd.read_excel('datasets/Monthly_CO2_Concentrations.xlsx')

result = seasonal_decompose(data.CO2.tolist(), freq=12,model='additive')

print(type(result.trend))
#print(result.seasonal)
#print(result.resid)
#print(result.observed)
```

```
<class 'numpy.ndarray'>
```

The argument **freq** in the **seasonal.seasonal_decompose** is the periodicity of the seasonal behavior and the original time series being monthly observations we suspect a periodicity of 12 (why?).

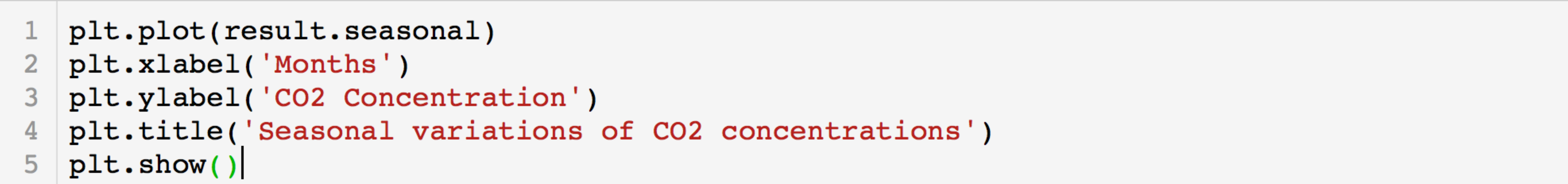
Now the variable "result" is an objecto of the tsa.seasonal.DecomposeResult class, which contains four arrays:

- *result.trend*: the **trend** component
- *result.seasonal*: the **seasonal** variations
- *result.resid*: the **residual** component by removing trend and seasonal, basically "random + cyclic"
- *result.observed*: the **original data**

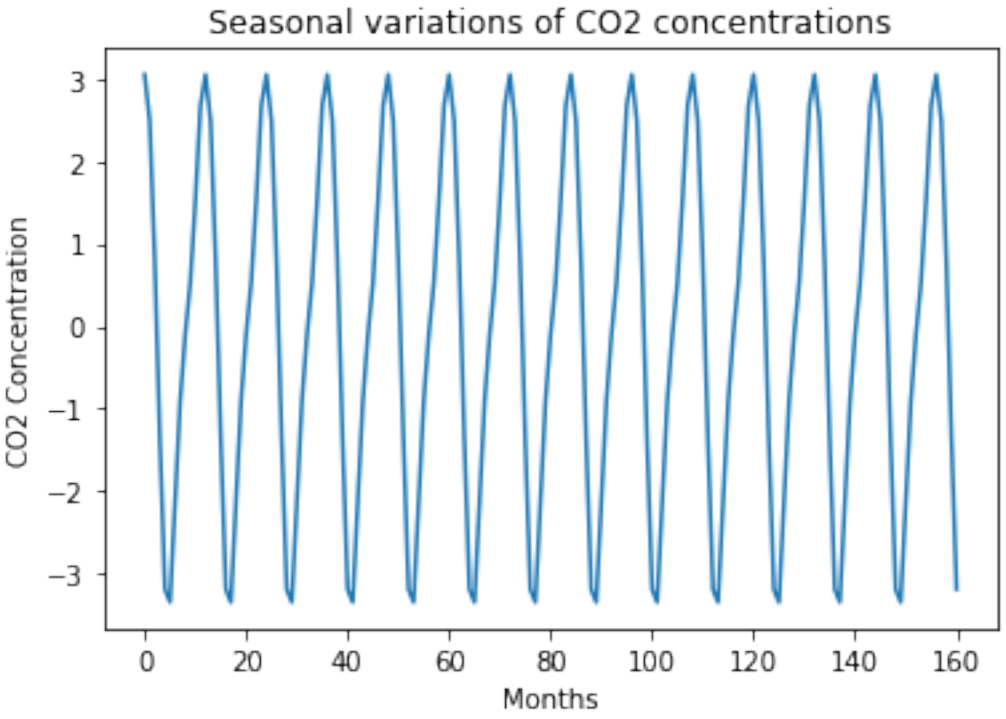
each of the above variables are NumPy arrays

Using the statsmodels.tsa module

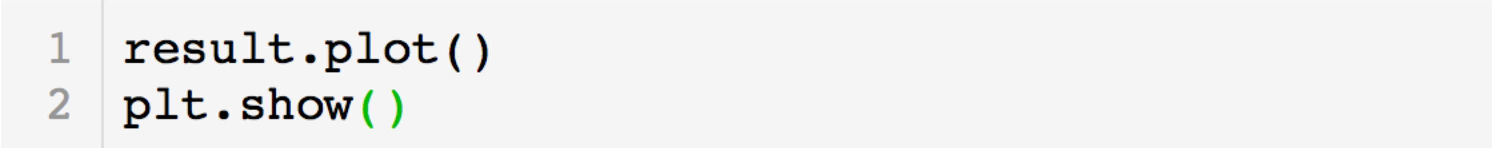
After calling the seasonal_decompose() function, you can plot the results as numpy arrays:



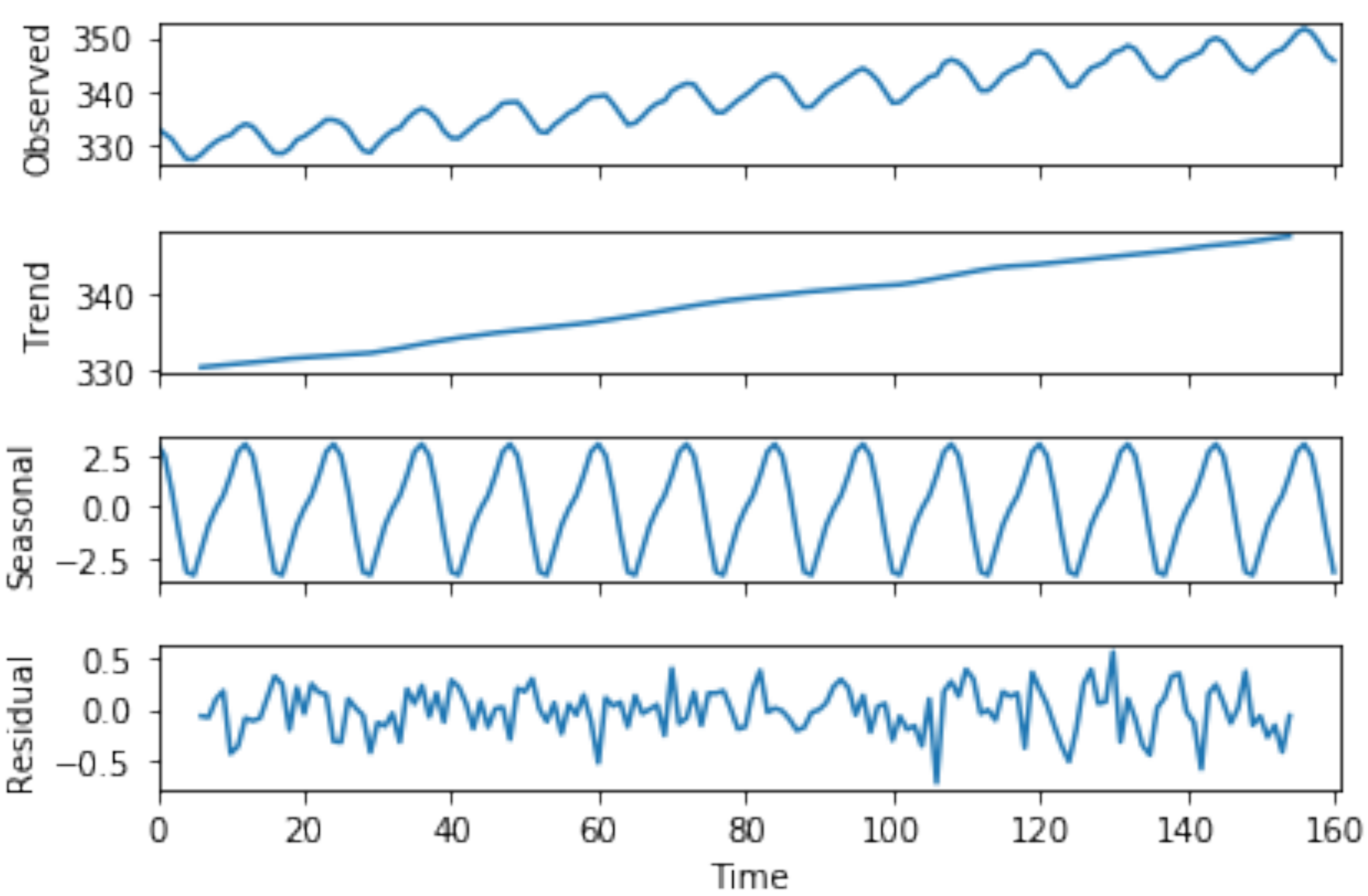
```python
plt.plot(result.seasonal)
plt.xlabel('Months')
plt.ylabel('CO2 Concentration')
plt.title('Seasonal variations of CO2 concentrations')
plt.show()
```



You can also use the .plot() method for the result object:



```python
result.plot()
plt.show()
```



Now let’s decompose some time series