

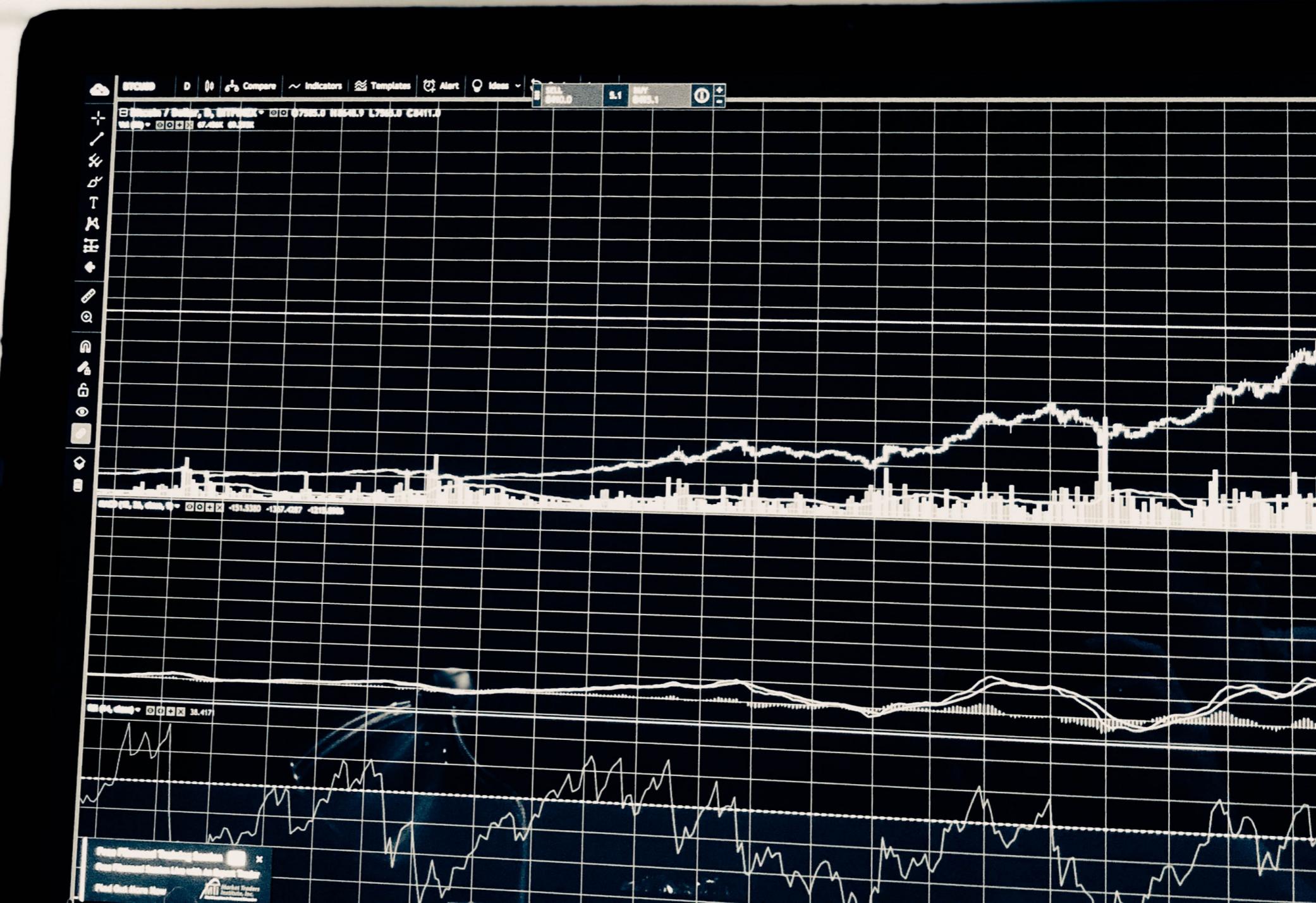


# Time Series For Everyone

Bruno Gonçalves

[www.data4sci.com/newsletter](http://www.data4sci.com/newsletter)  
[graphs4sci.substack.com](https://graphs4sci.substack.com)

<https://github.com/DataForScience/Timeseries>

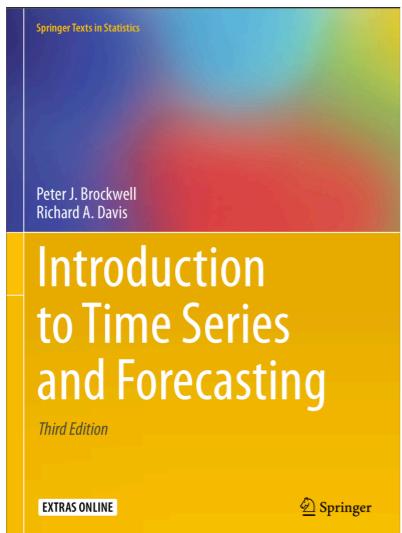




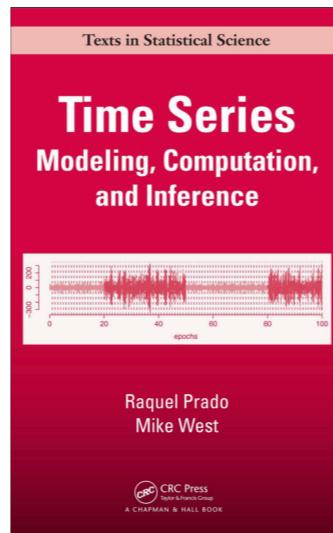
## Lesson I:

# Understanding Timeseries

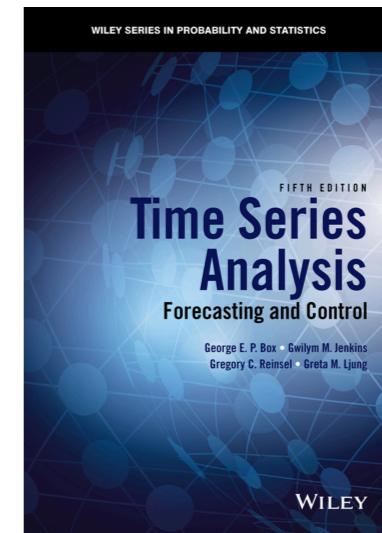
# References



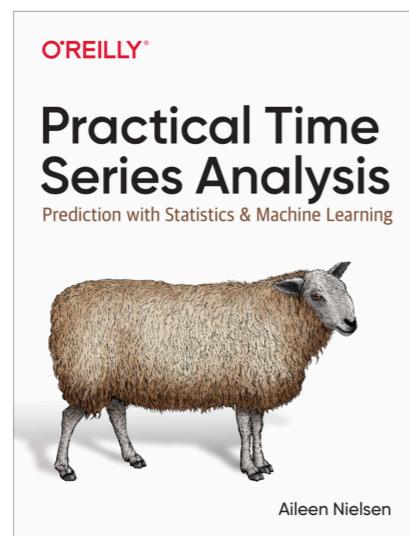
<https://amzn.to/30yEp8l>



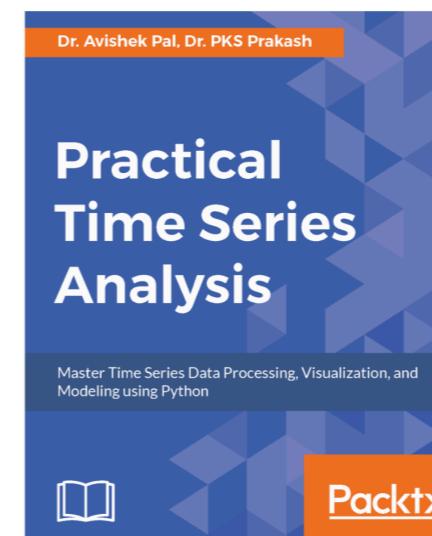
<https://amzn.to/2AhvCxt>



<https://amzn.to/30wY5db>



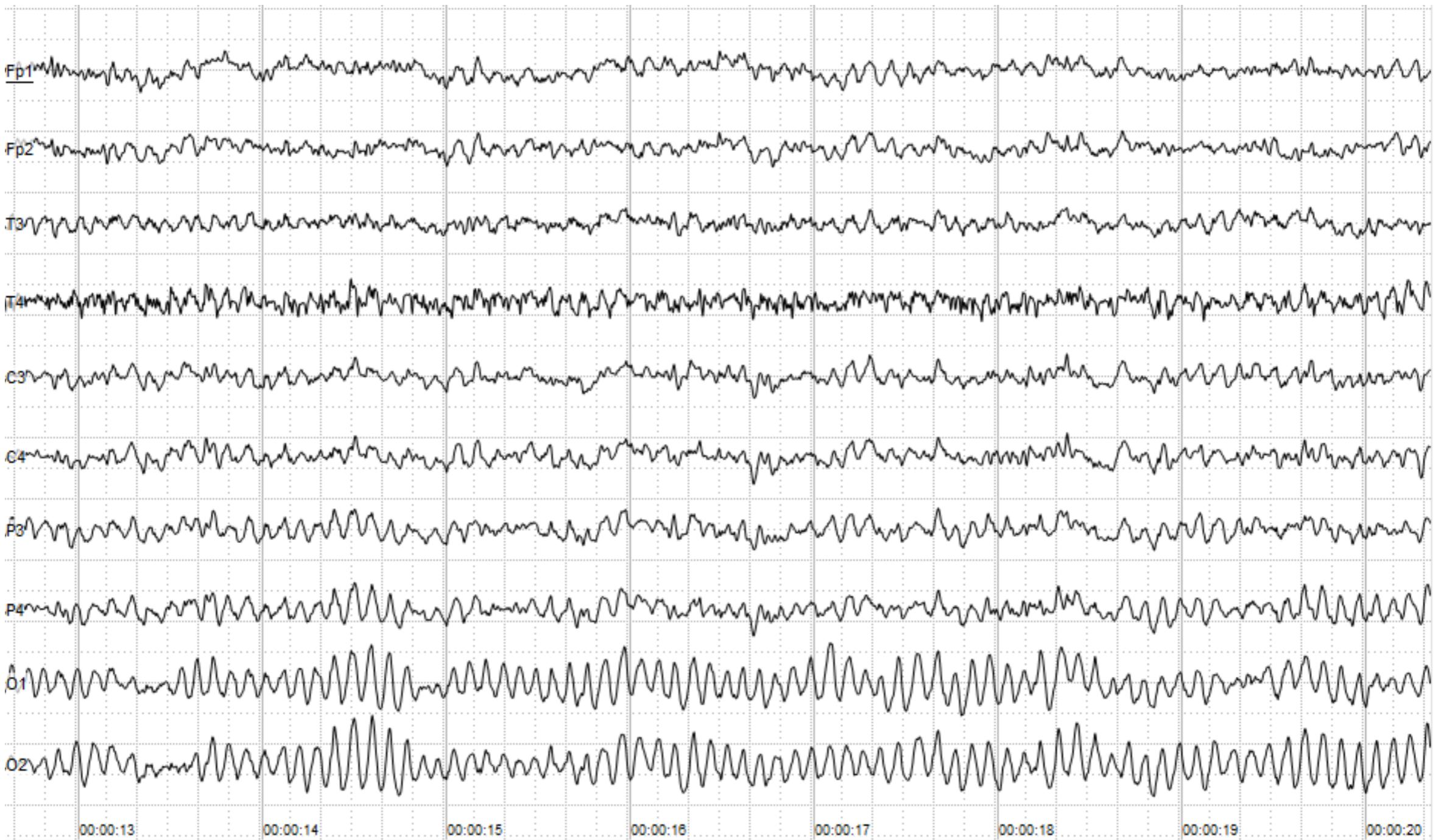
<https://amzn.to/2YoEKs6>



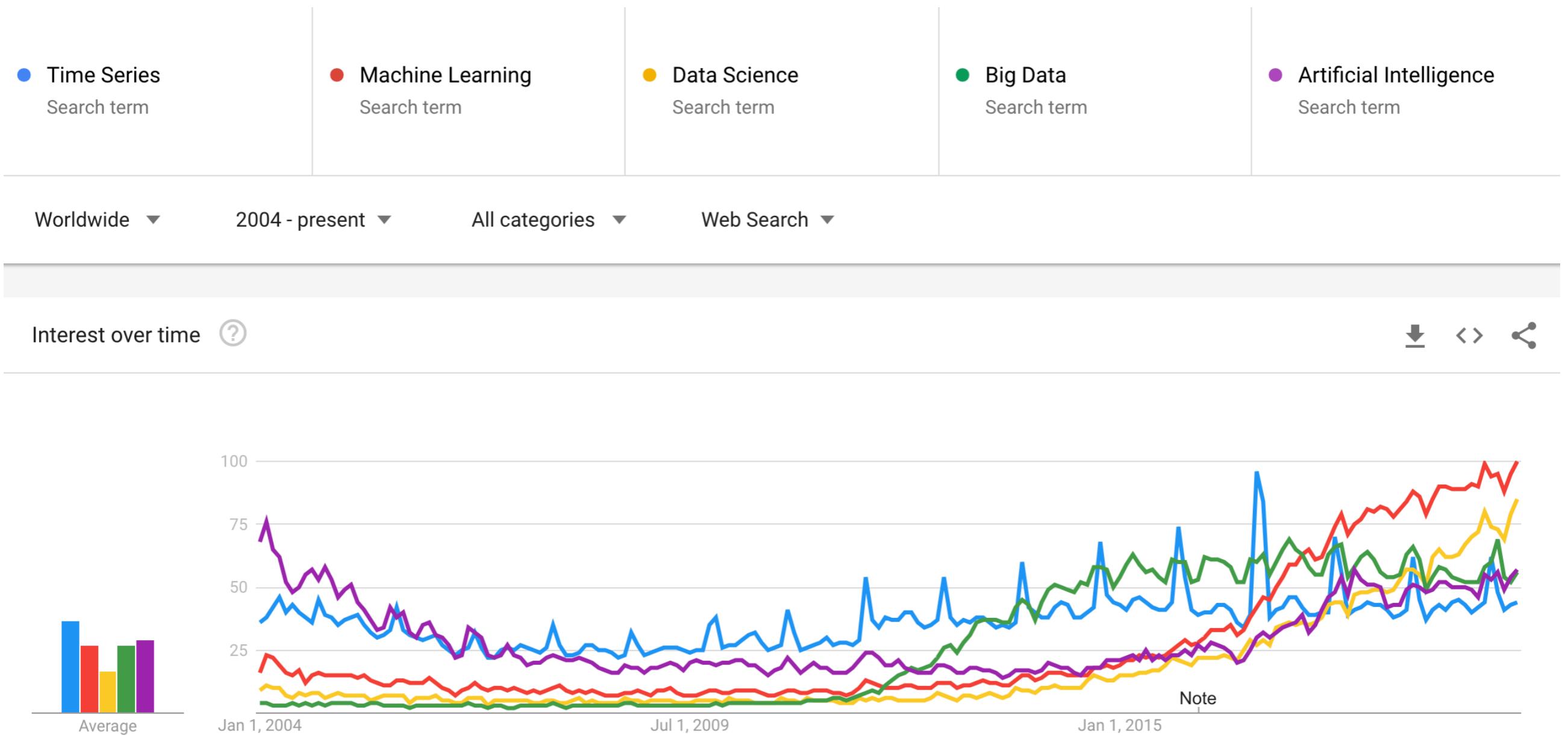
<https://amzn.to/30wY8Wp>

# EEG

<https://en.wikipedia.org/wiki/Electroencephalography>

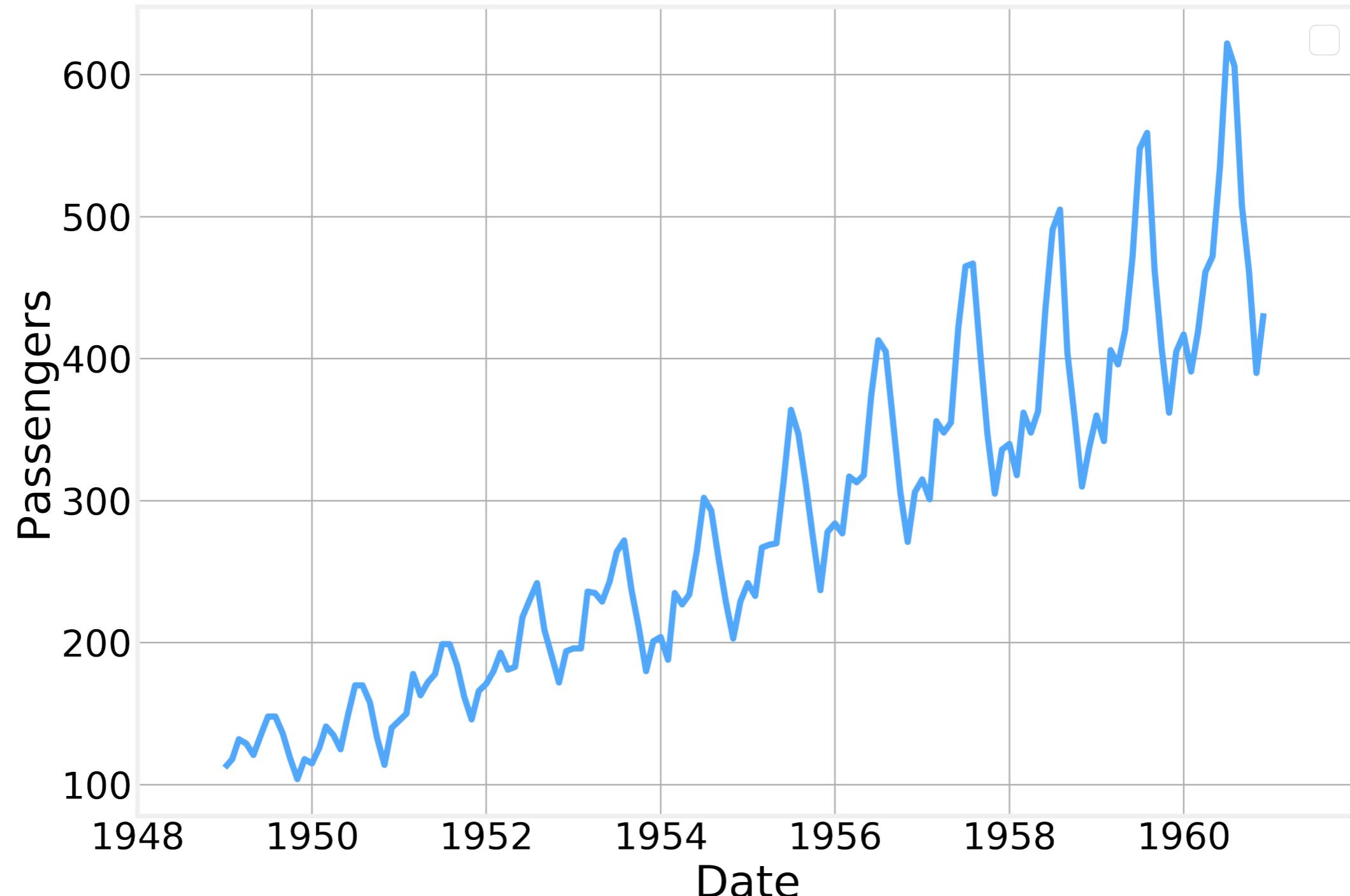


# Public Interest - Search Trends



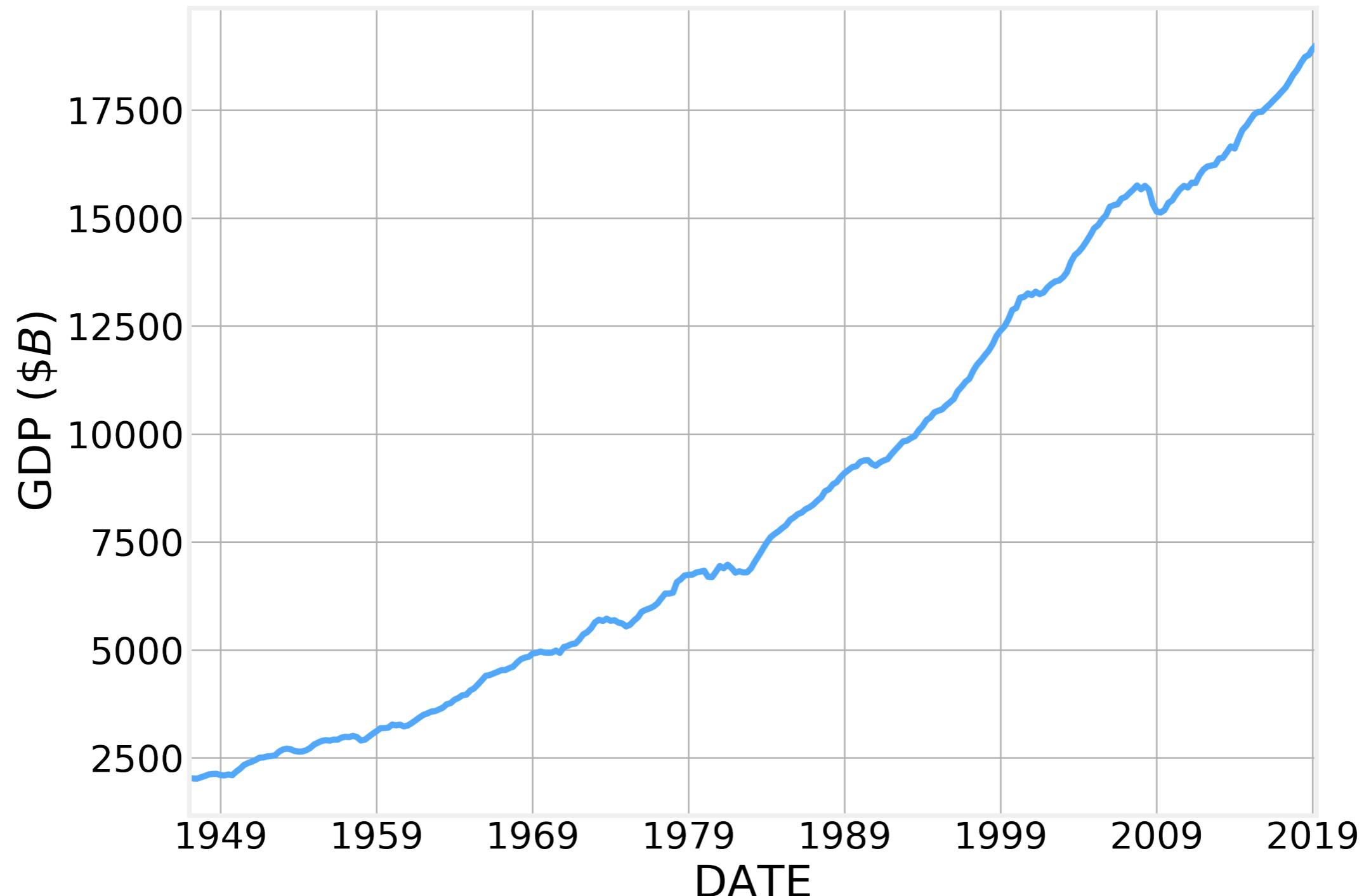
# Airline Passengers

<https://www.kaggle.com/chirag19/air-passengers>



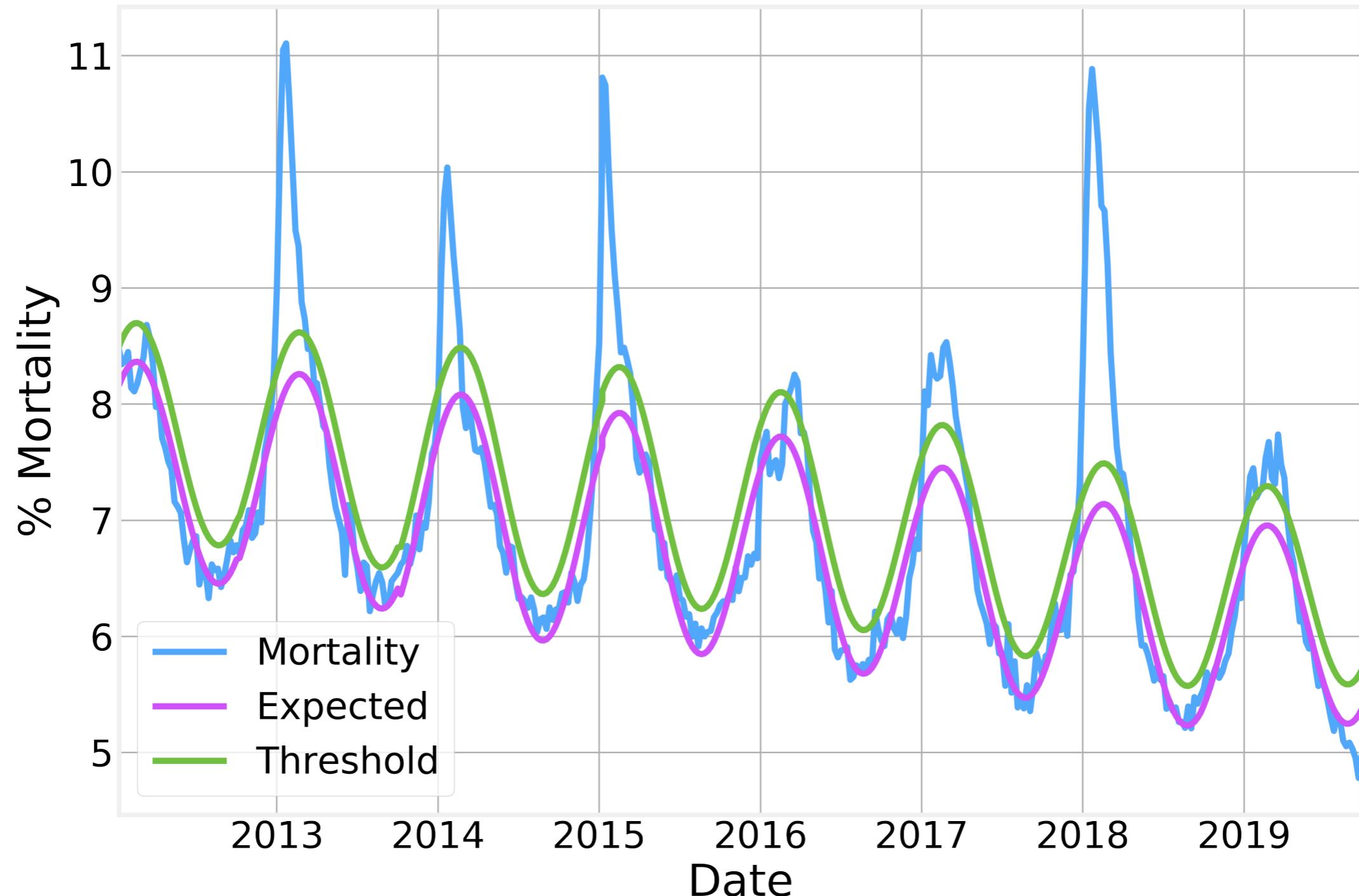
# GDP

<https://fred.stlouisfed.org/series/GDPC1>



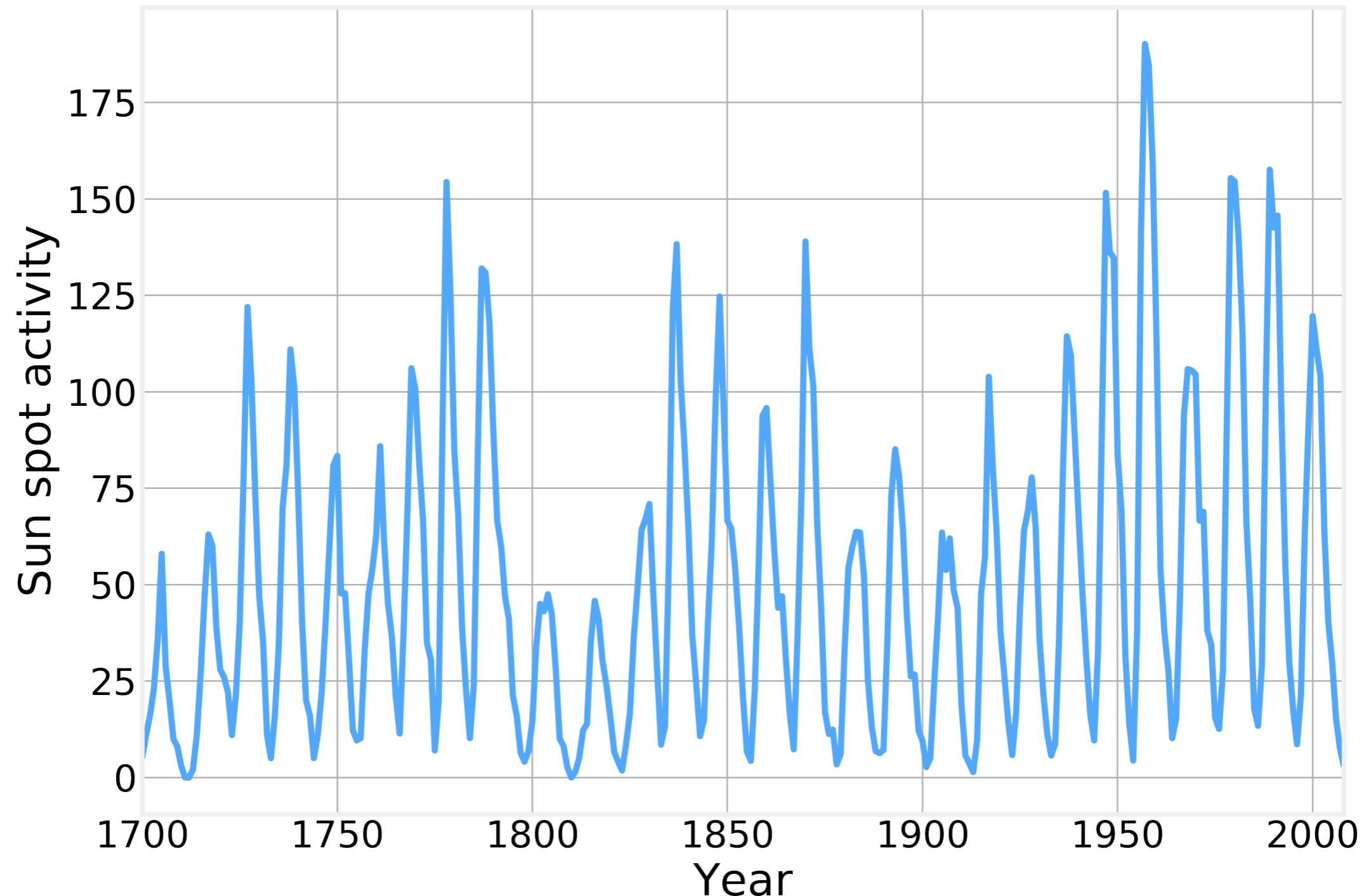
# Influenza

[www.cdc.gov/flu/weekly/](http://www.cdc.gov/flu/weekly/)



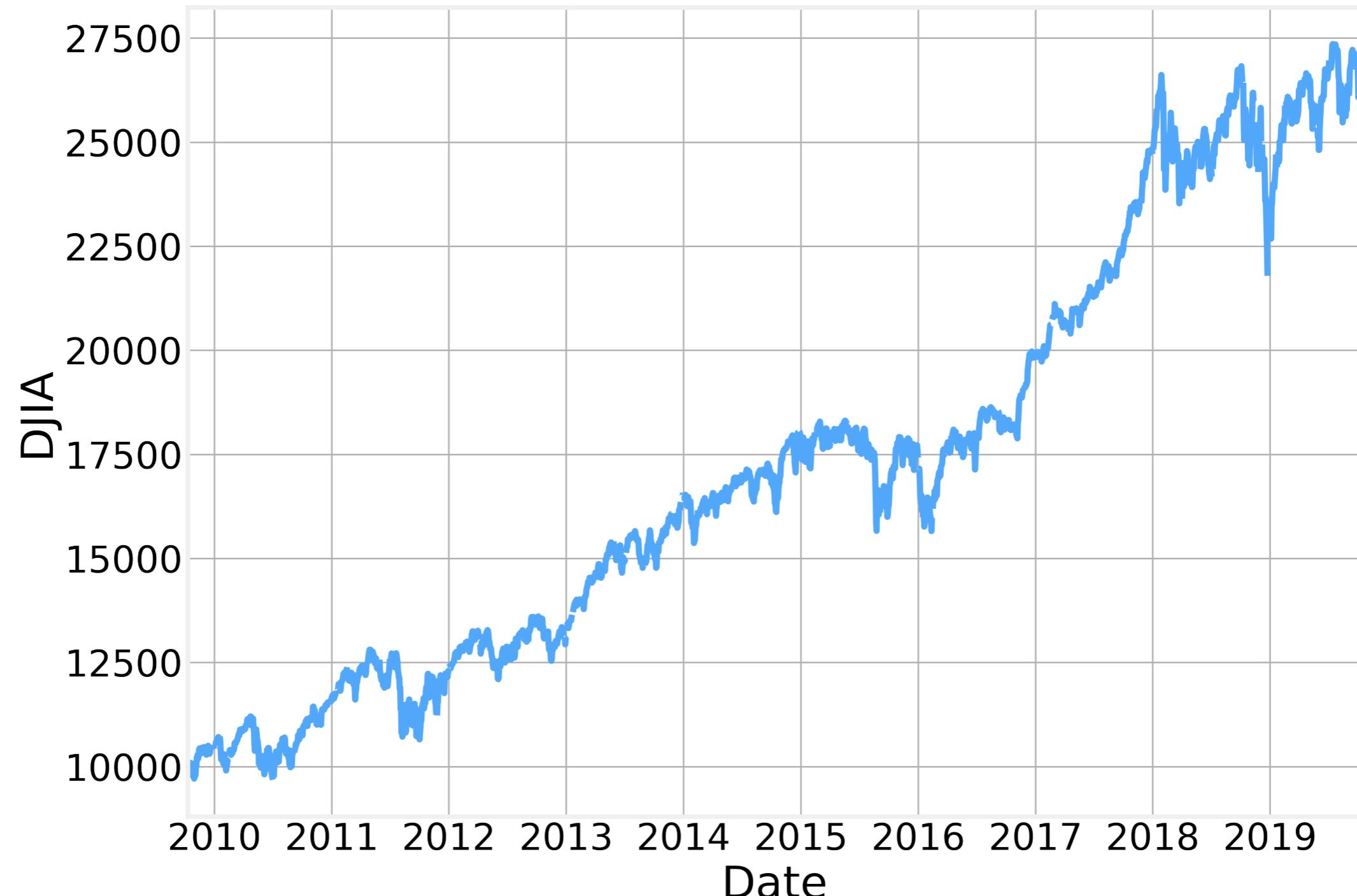
# Sunspot activity

<http://www.sidc.be/silso/datafiles>



# Stock Market - DJIA

<https://fred.stlouisfed.org/series/DJIA>



# Time series

---

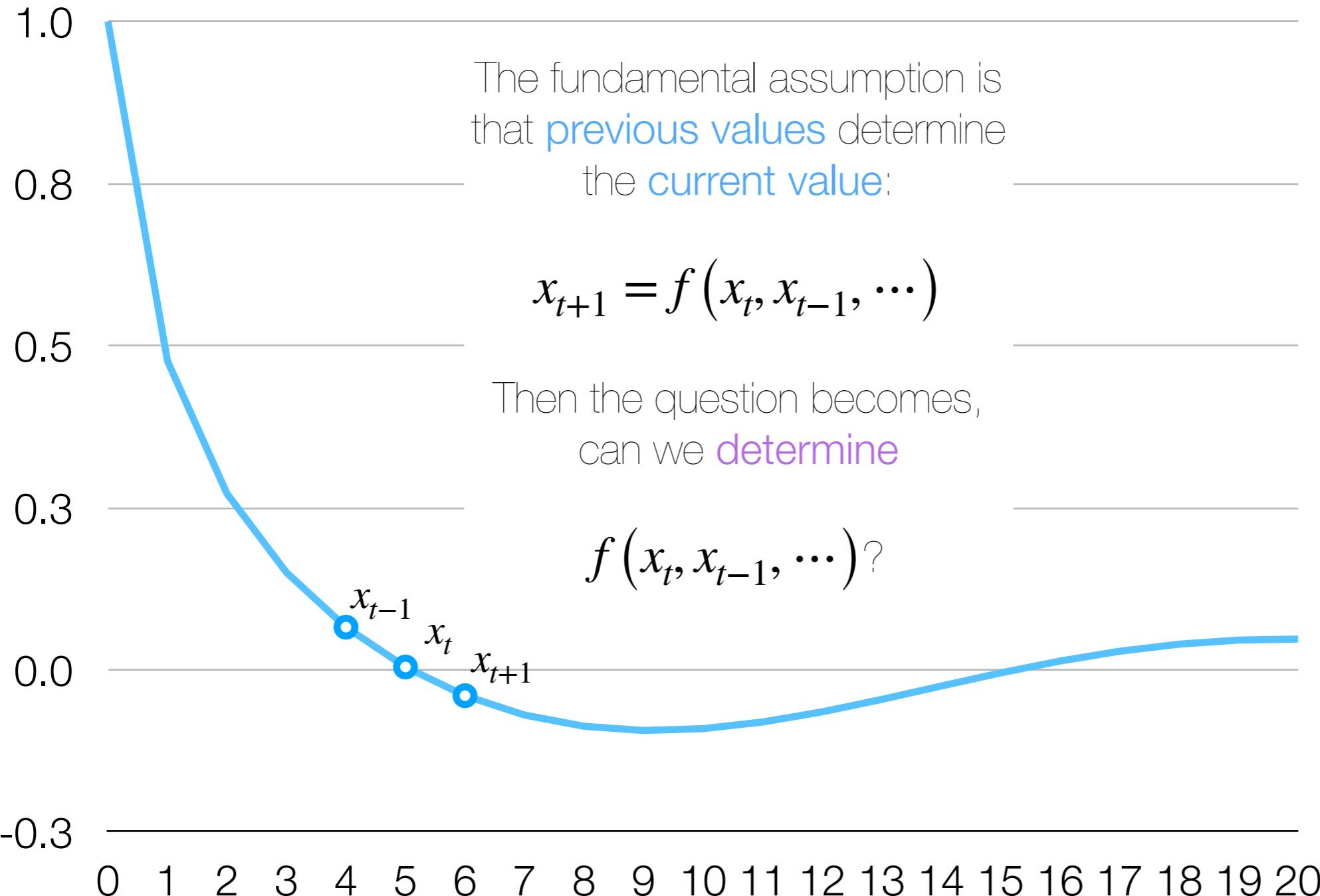
- A set of values measured **sequentially** in **time**
- Values are **typically** (but not always) measured at **equal intervals**,  $x_1, x_2, x_3, x_3$ , etc...
- Values can be:
  - **continuous**
  - **discrete** or **symbolic** (words).
- Associated with **empirical** observation of time varying phenomena:
  - Stock market prices (**day**, **hour**, minute, **tick**, etc...)
  - Temperatures (**day**, **minute**, second, etc...)
  - Number of patients (**week**, **month**, etc...)
  - GDP (**quarter**, **year**, etc...)
- **Forecasting** requires predicting **future** values based on **past** behavior

# Mathematical Conventions

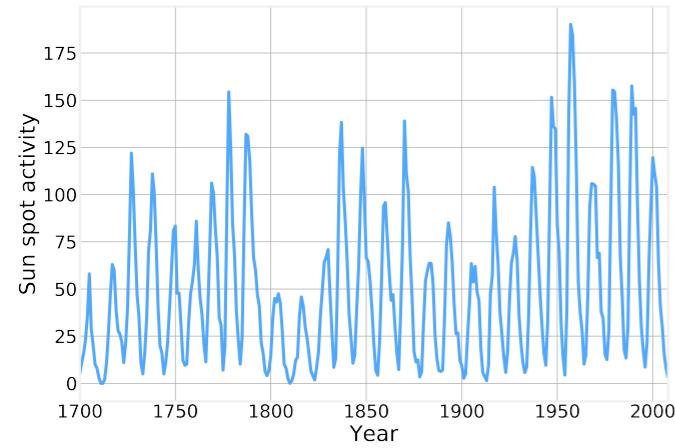
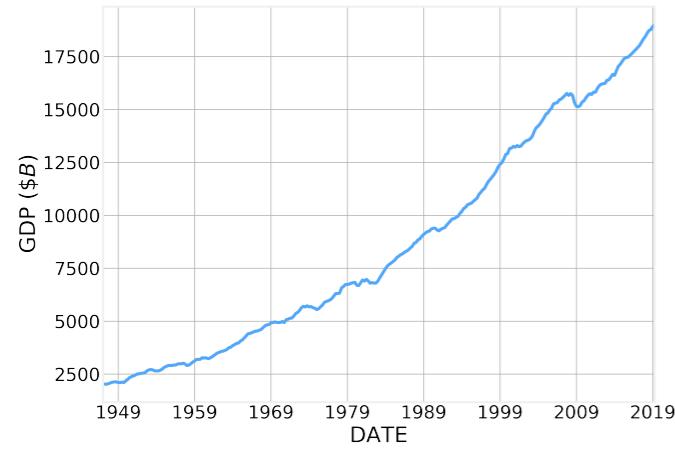
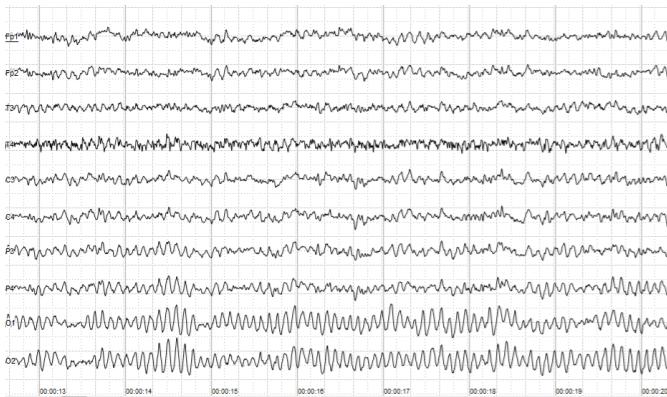
---

- The value of the time series at time  $t$  is given by  $x_t$
- The values at a given lag  $l$  are given by  $x_{t-l}$
- The mean of the overall signal is  $\mu$  and the corresponding running value is  $\mu_t^w$
- The variance of the overall signal is  $\sigma$  and the corresponding running value is  $\sigma_t^w$
- Running values are calculated over a window of width  $w$

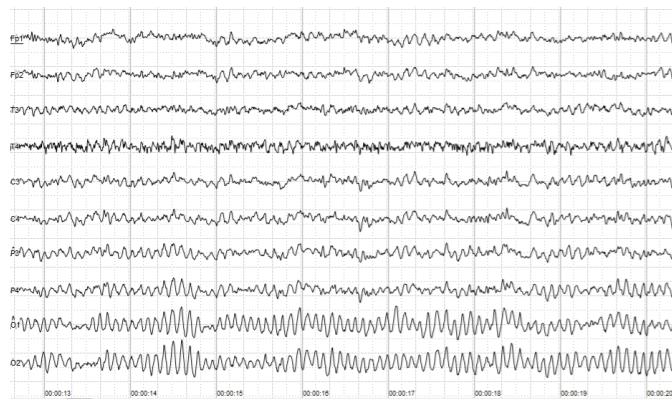
# Time series analysis



# Three fundamental behaviors

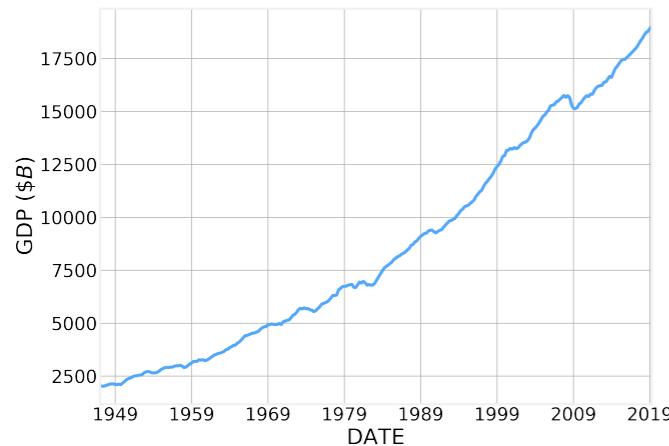


# Three fundamental behaviors



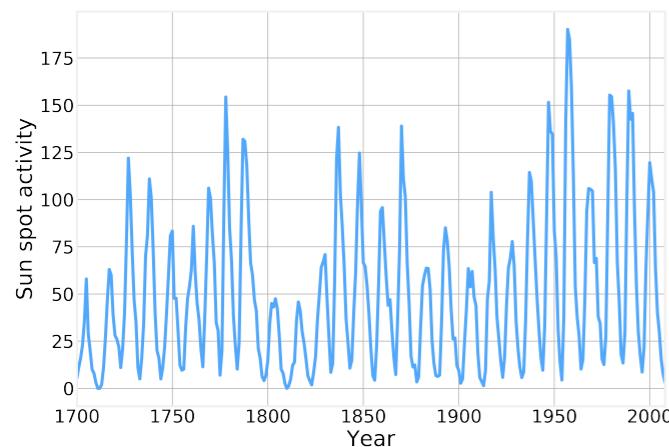
Stationarity

$$\langle x_t \rangle \approx \text{constant}$$



Trend

$$\langle x_t \rangle \approx ct$$



Seasonality

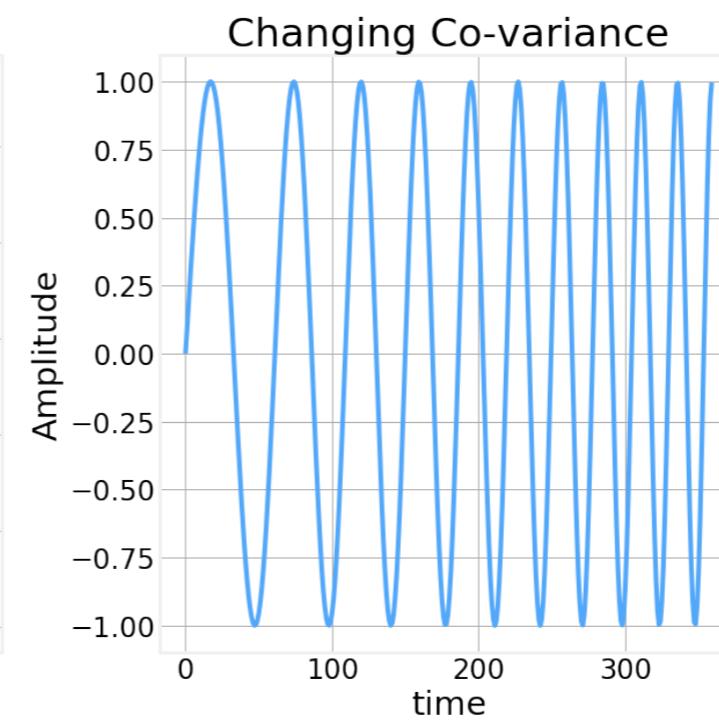
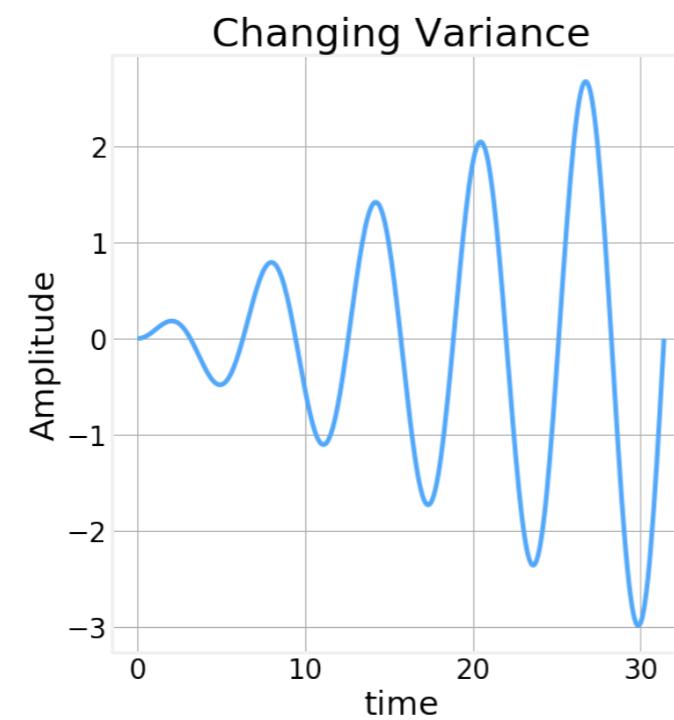
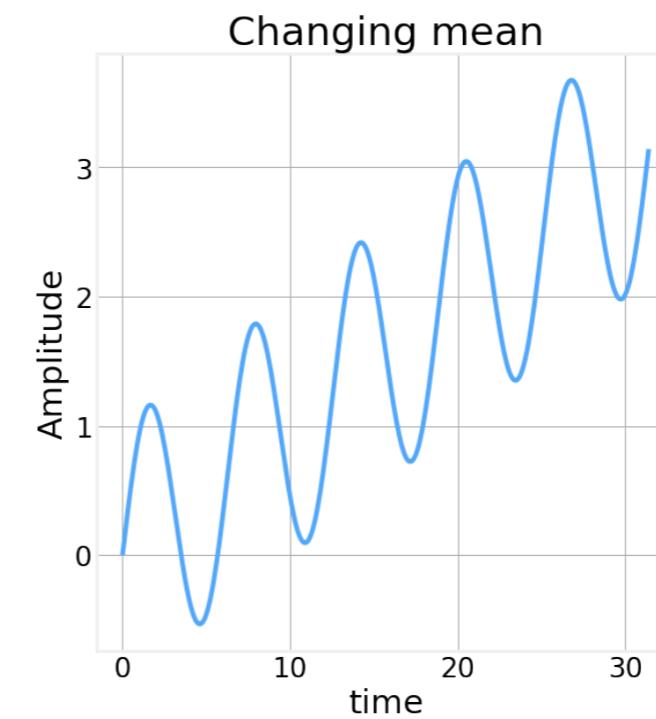
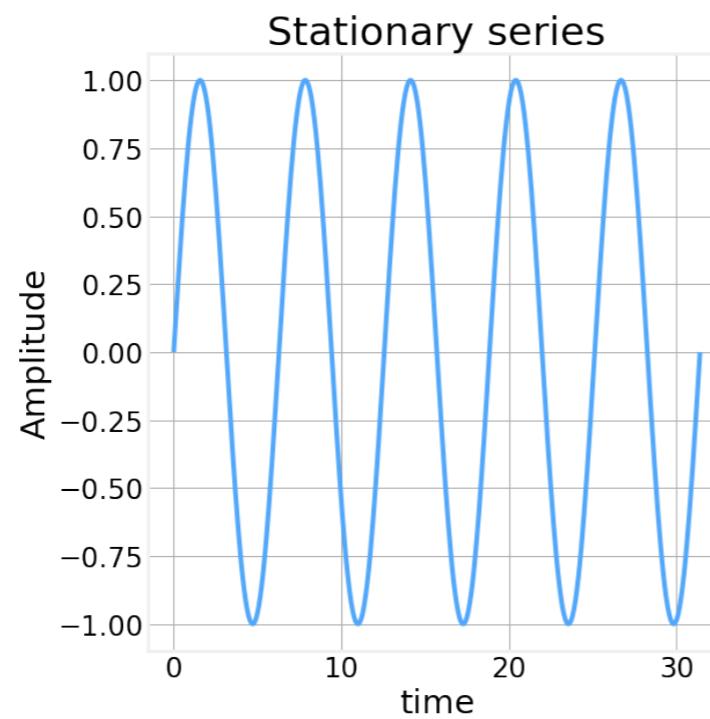
$$x_{t+T} \approx x_t$$

# Stationarity

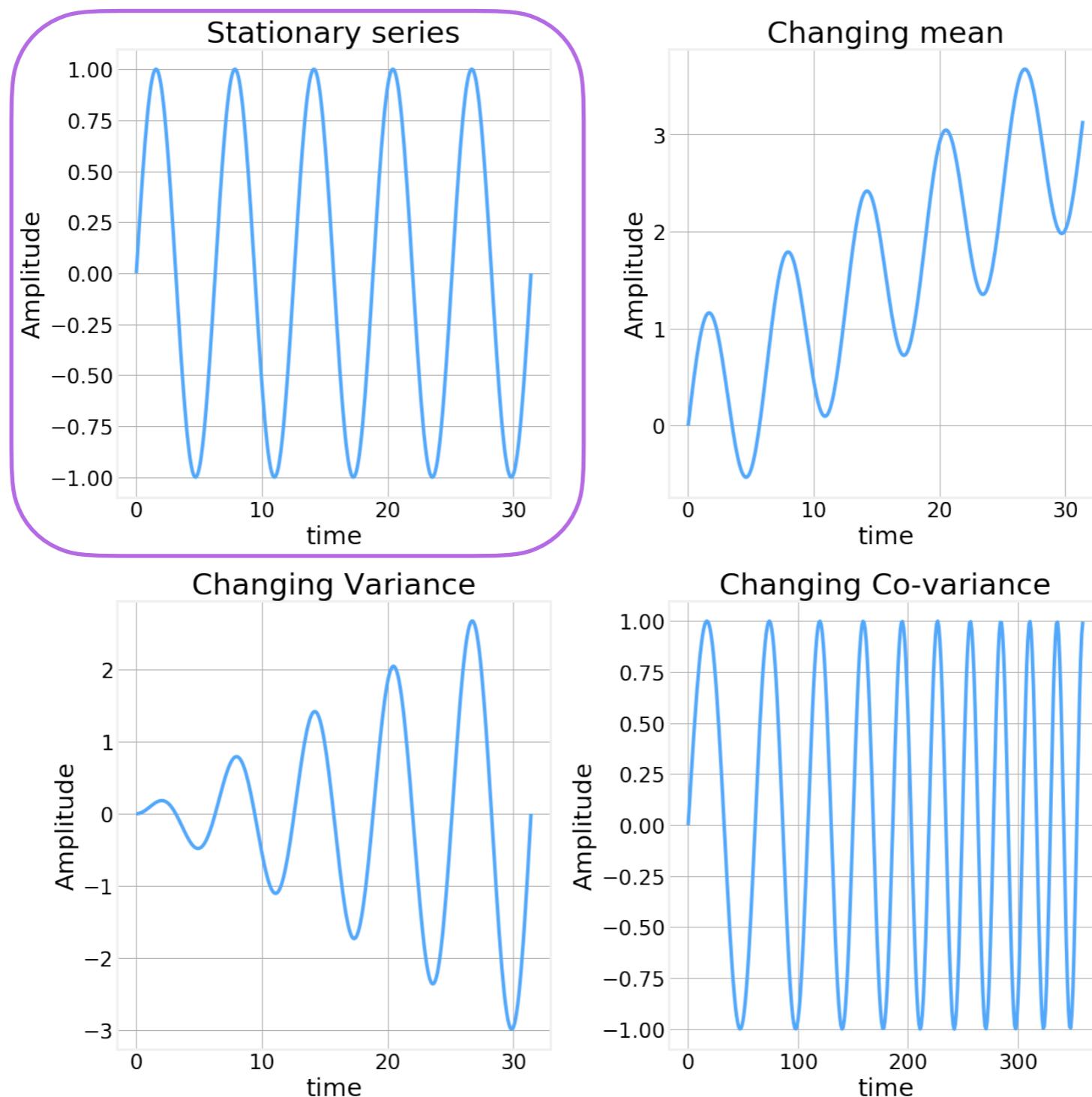
---

- A time series is said to be stationary if its basic statistical properties are **independent of time**
- In particular:
  - **Mean** - Average value stays constant
  - **Variance** - The width of the curve is bounded
  - **Covariance** - Correlation between points is independent of time
- Stationary processes are **easier** to analyze
- Many time series analysis algorithms assume the time series to be stationary
- Several rigorous tests for stationarity have been developed such as the **(Augmented) Dickey-Fuller** and **Hurst Exponent**
- Typically, the first step of any analysis is to transform the series to make it stationary

# Stationarity



# Stationarity



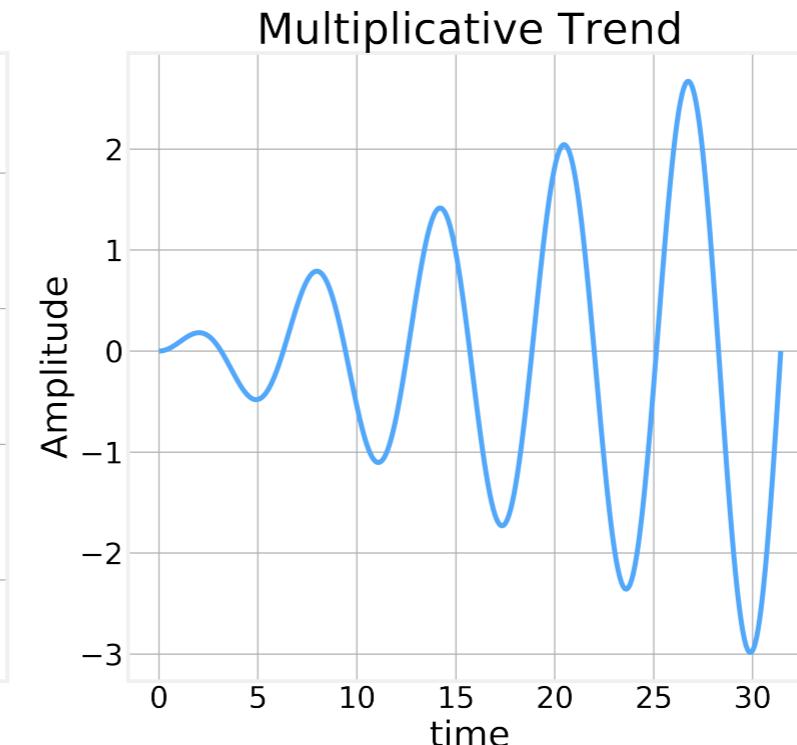
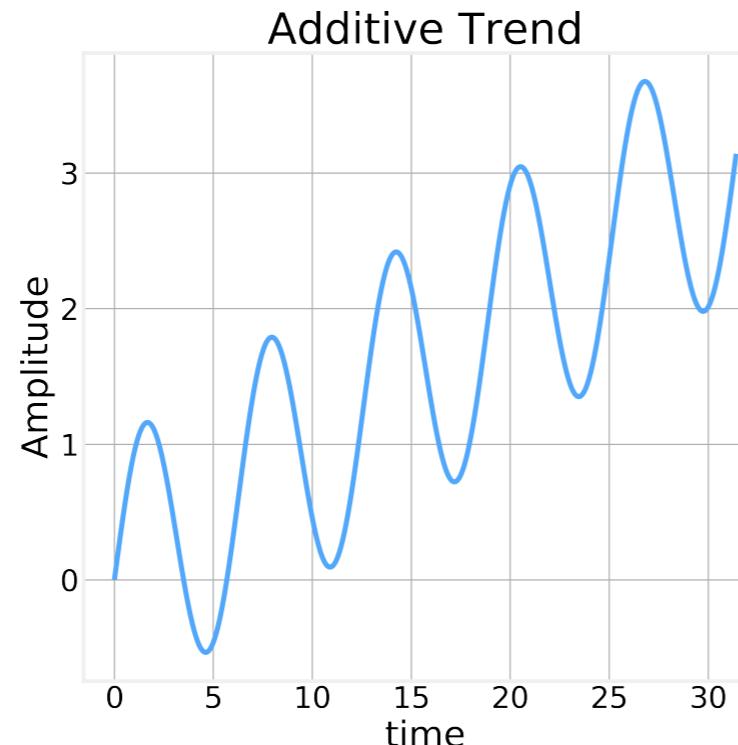


Code - Data Exploration  
<https://github.com/DataForScience/Timeseries>

# Trend

- Many time series have a clear trend or tendency:
  - Stock market indices tend to go up over time
  - Number of cases of preventable diseases tends to go down over time
  - etc
- Trends can be **additive** or **multiplicative**:

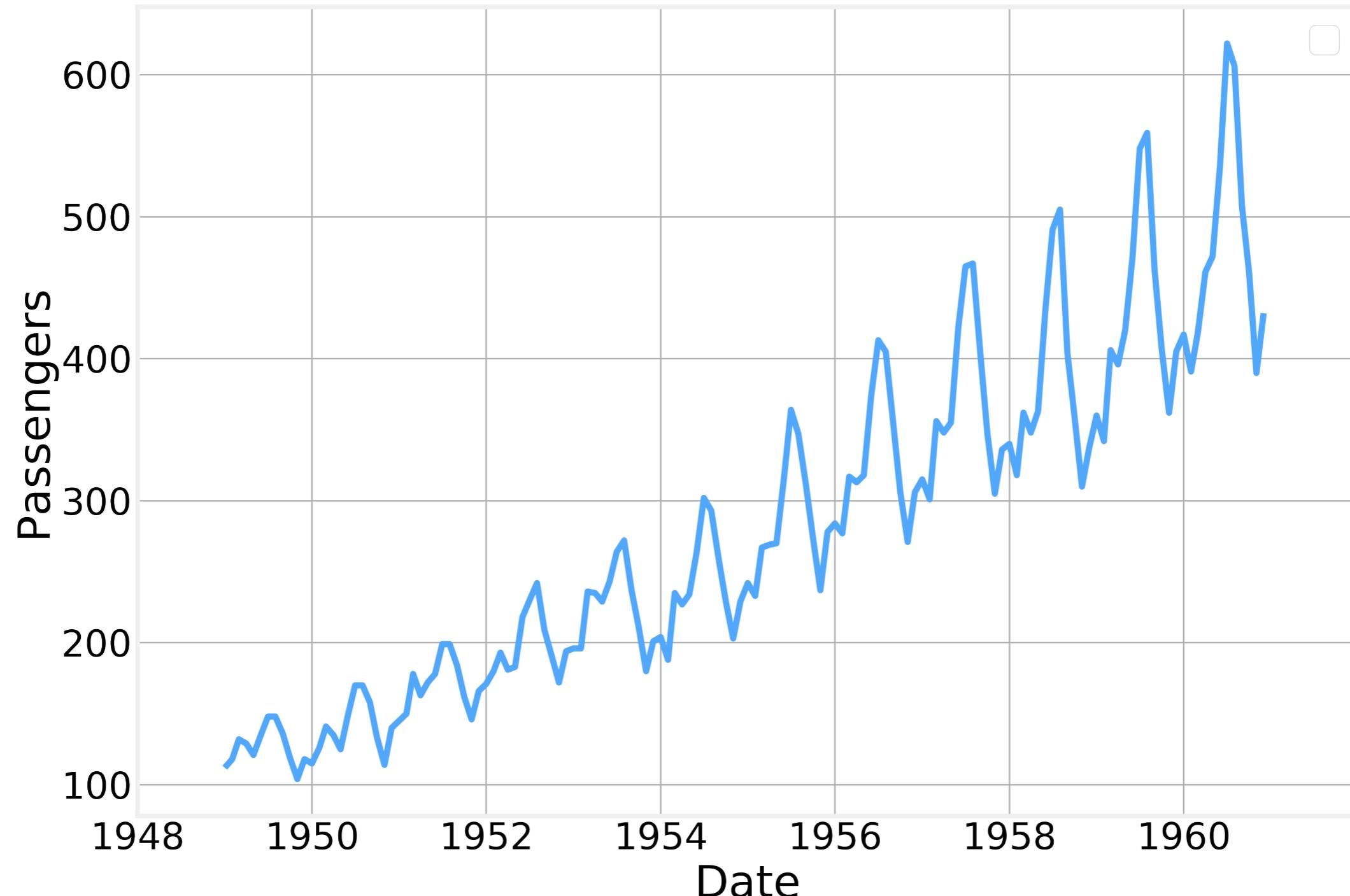
$$x = \frac{t}{10} + \sin(t)$$



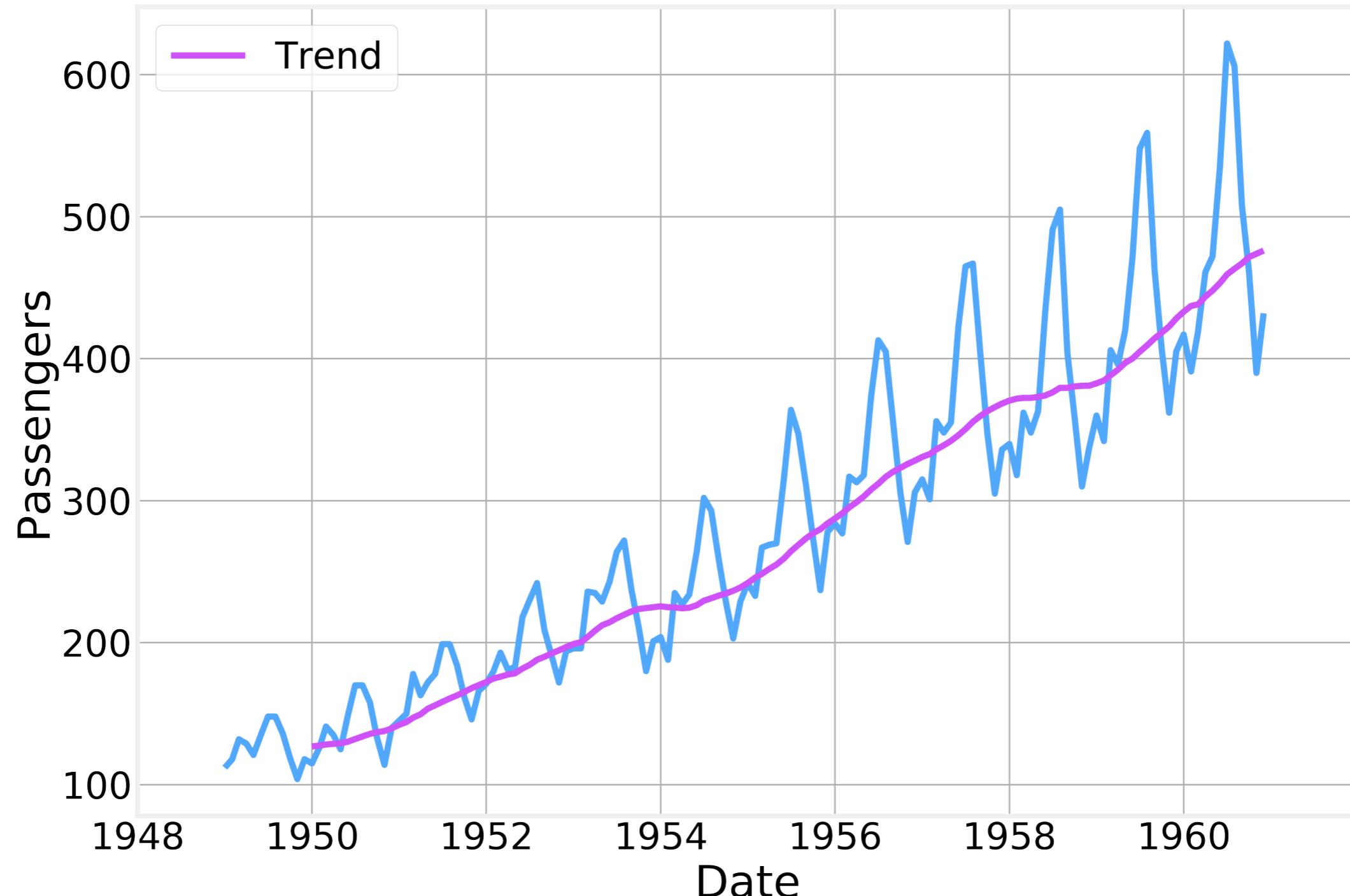
$$x = \frac{t}{10} \sin(t)$$

- Trends can be removed by **subtraction** or **division** of the correct values
- One simple way to determine the trend is to calculate a running average over the series

Trend



# Trend



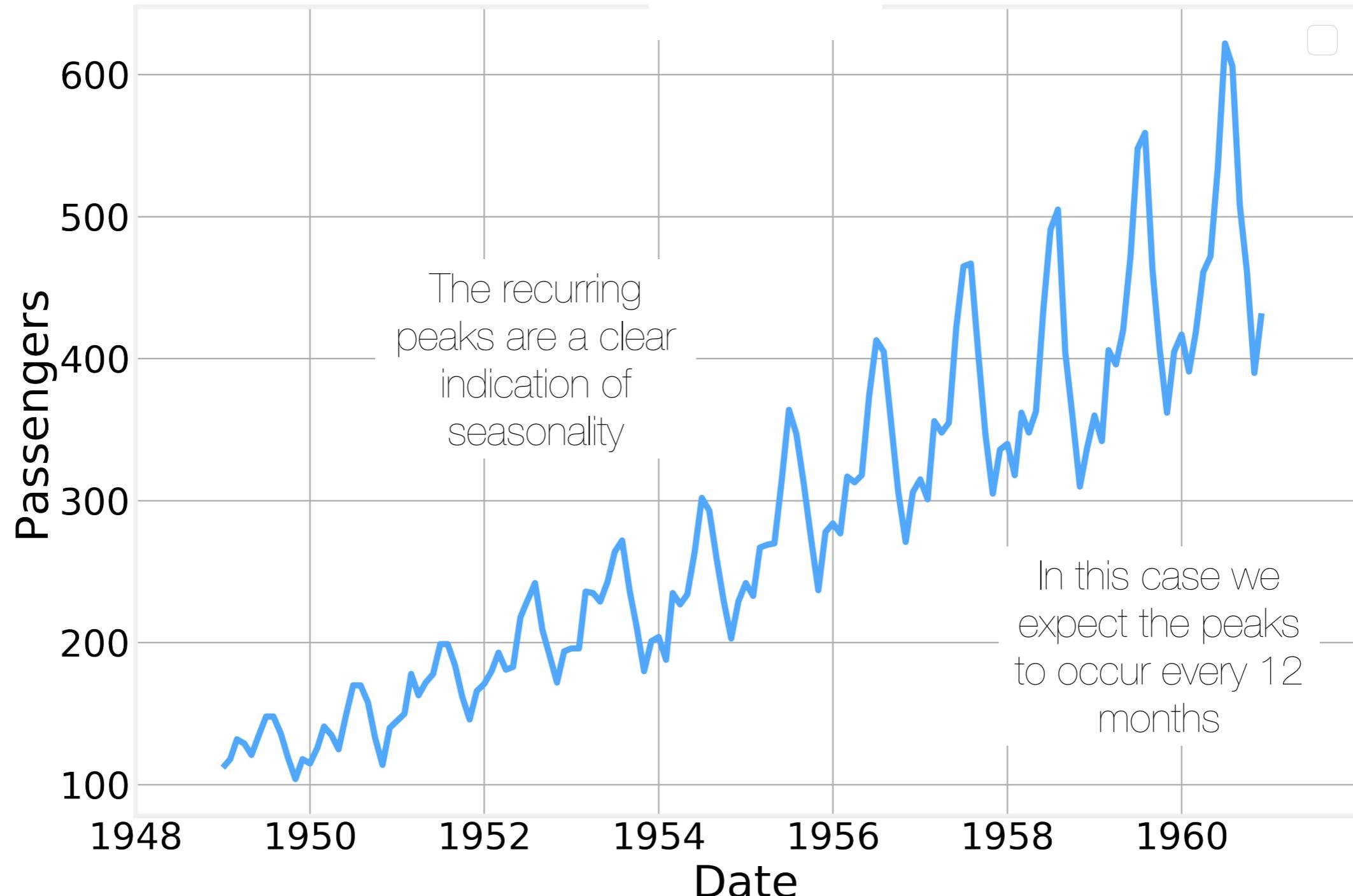
# Seasonality

---

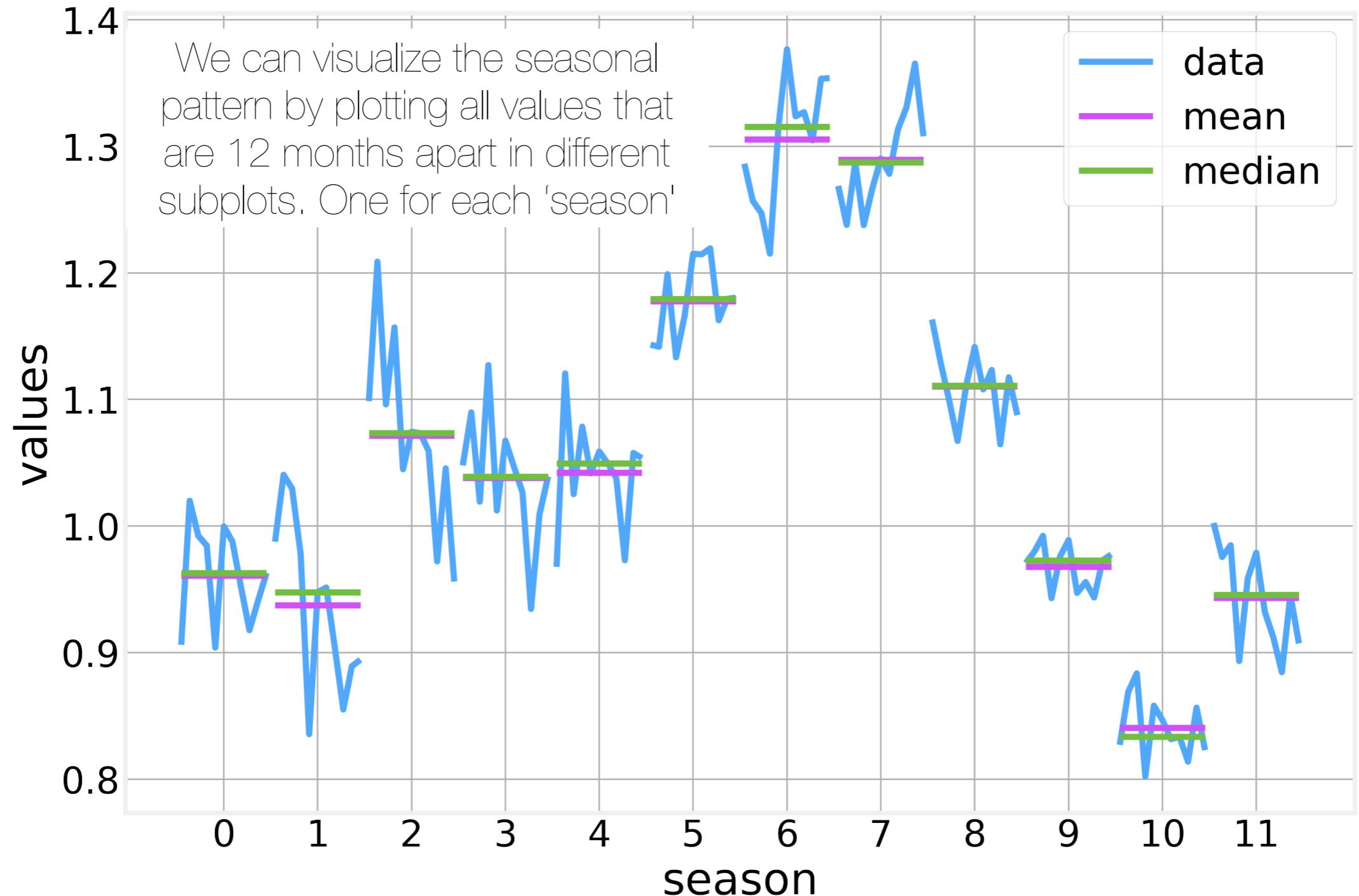
- Many of the phenomena we might be interested in vary in time in a **cyclical** or **seasonal** fashion
  - Ice-cream sales peak in the **summer** and drop in the **winter**
  - Number of cell phone calls made is larger during the day than at night
  - Many types of crime are more frequent at **night** than during the **day**
  - Visits to museums are more frequent in the **weekend** than in **weekdays**
  - The stock market grows during **bull** periods and shrinks during **bear** periods
  - etc
- Understanding the seasonality of a time series provides important information about its long term behavior and is extremely useful in predicting future values
- If the period is fixed it's called **seasonality**, while if the period is **irregular** it's called cyclical

# Seasonality

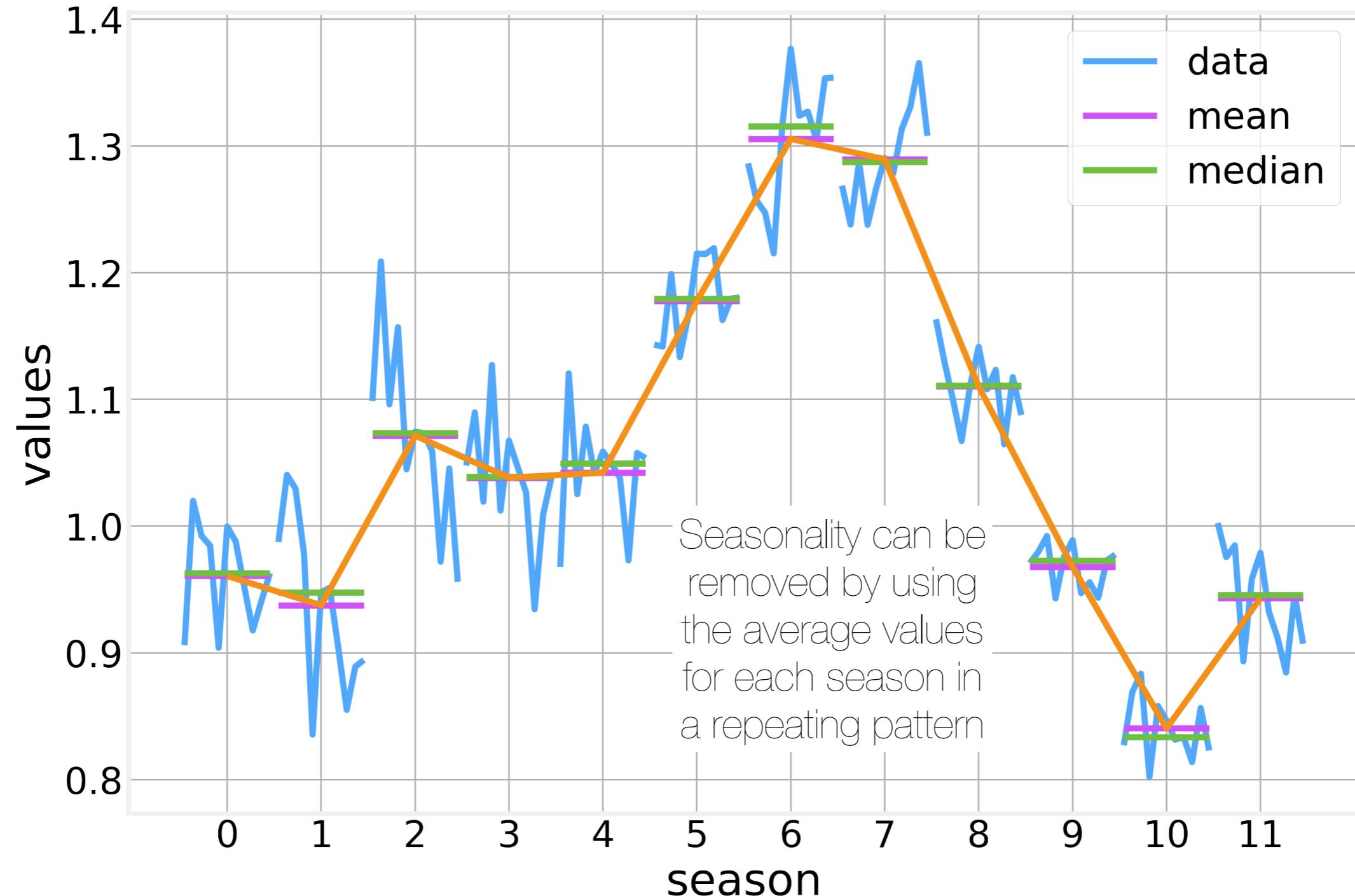
<https://www.kaggle.com/chirag19/air-passengers>



# Seasonality



# Seasonality

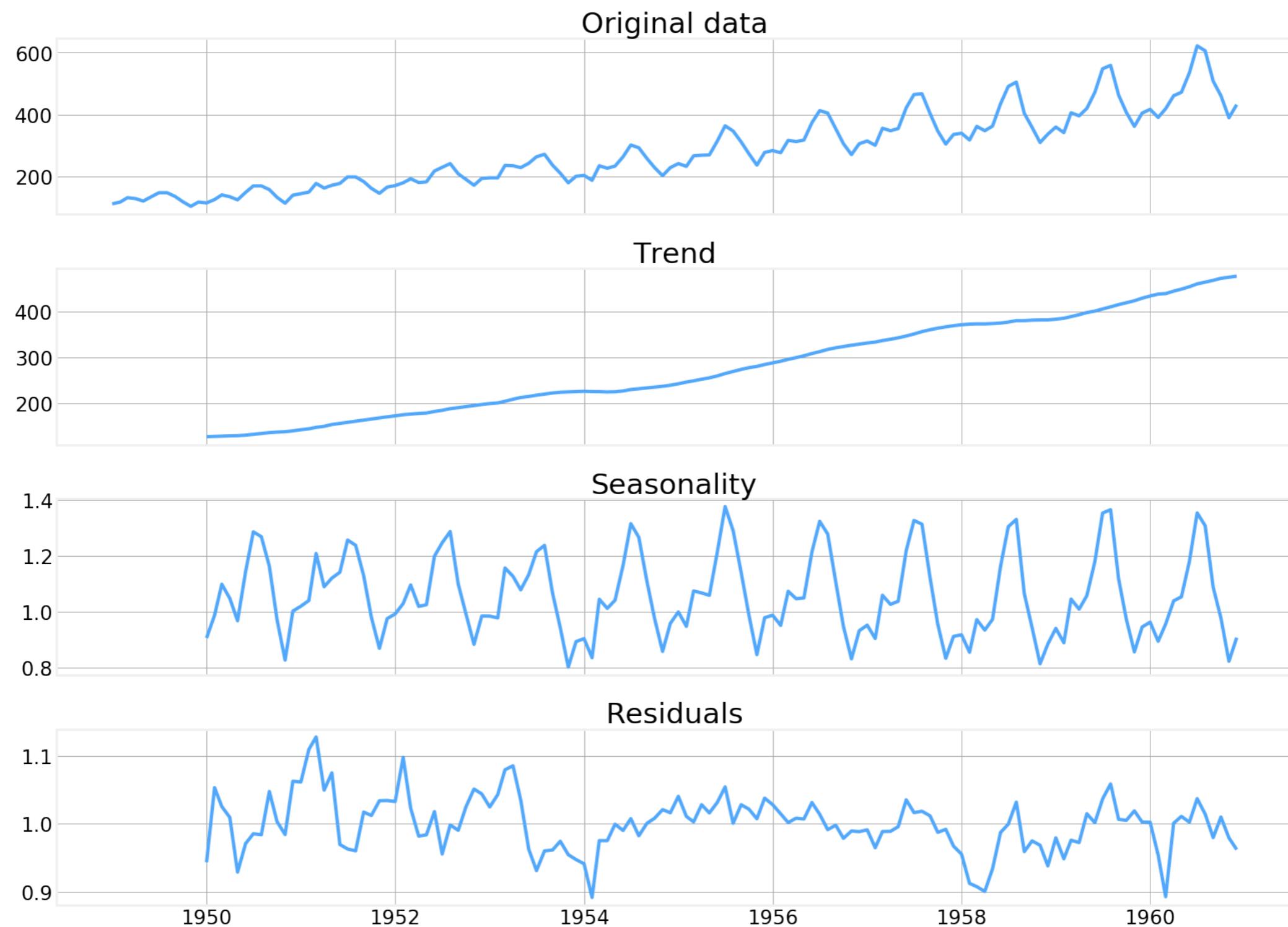


# Time series decomposition

---

- A time series can be decomposed into **three components**:
  - Trend,  $T_t$
  - Seasonality,  $S_t$
  - Residuals,  $R_t$
- Decompositions can be
  - **additive** -  $x_t = T_t + S_t + R_t$
  - **multiplicative** -  $x_t = T_t \cdot S_t \cdot R_t$
- The residuals are simply what is left of the original signal after we **remove the trend and the seasonality**
- Residuals are typically **stationary**

# Time series decomposition





Code - Decomposition  
<https://github.com/DataForScience/Timeseries>



## Lesson II:

### Processing Timeseries data

# Lagged values

---

- While analyzing time series, we often refer to values that our time series took **1, 2, 3**, etc time steps in the past
- These are known as lagged values and denoted:

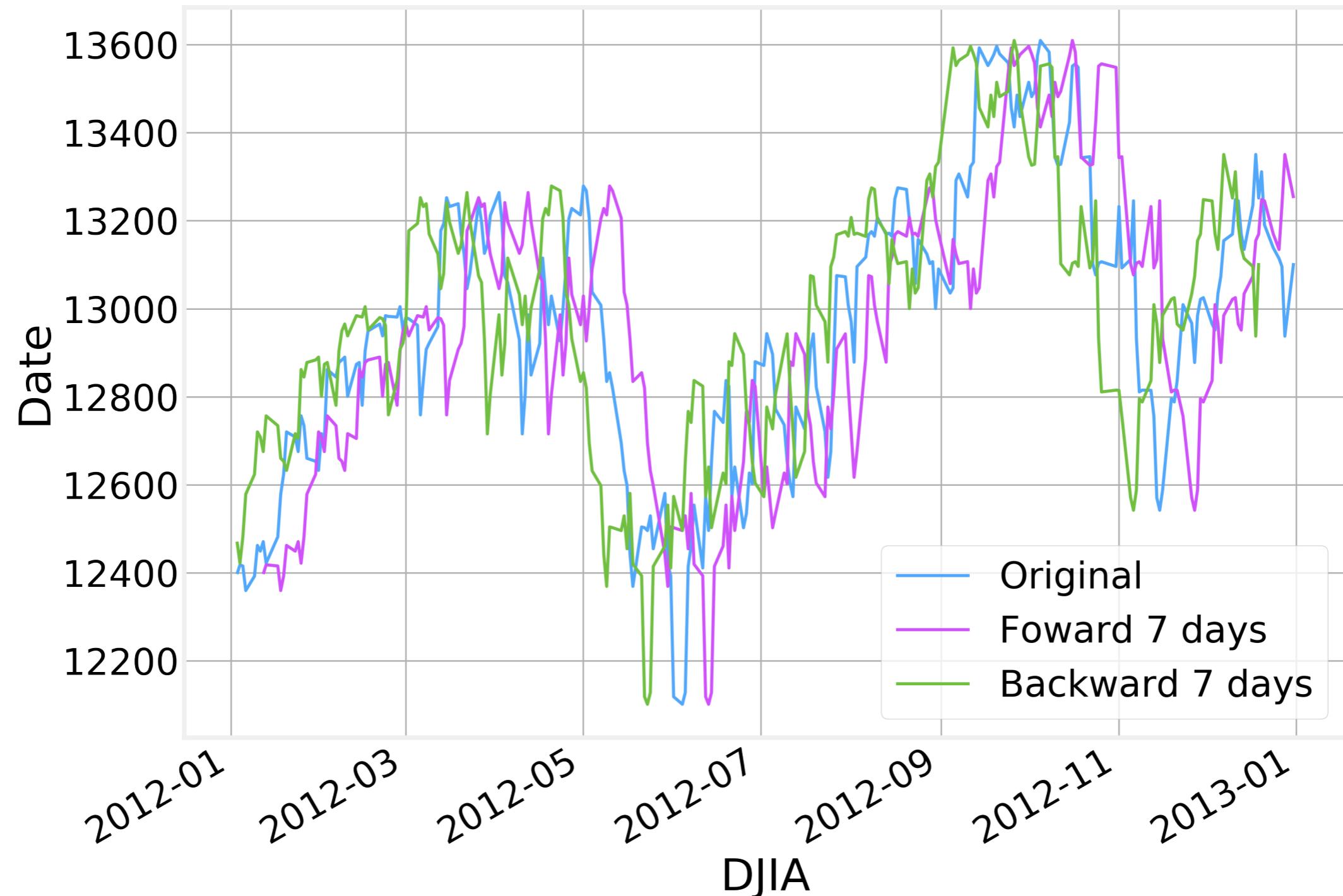
$$x_{t-l}$$

- where  $l$  is the value of the lag we are considering.

## Lagged values



# Lagged values



# Differences

---

- Perhaps the most common use case for lagged values is for the calculation of **differences** of the form:

$$x_t - x_{t-l}$$

- Where  $l \geq 1$  is the value of the lag we are interested in.
- Naturally, higher order differences can also be used, in which case, the difference of the difference is calculated:

$$y_t = x_t - x_{t-l}$$

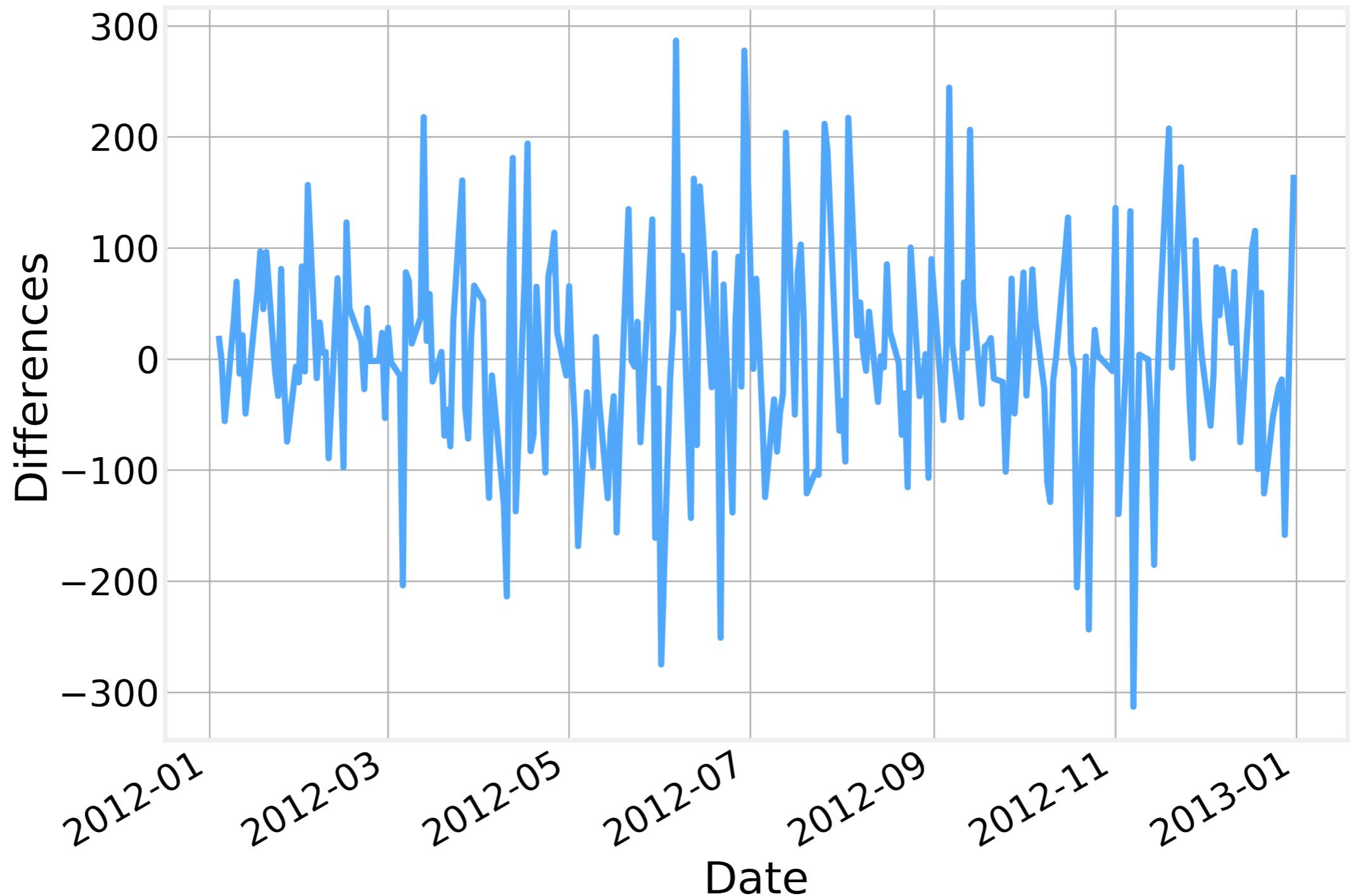
$$z_t = y_t - y_{t-l} \equiv x_t - 2x_{t-l} + x_{t-2l}$$

- This can be thought of as a discrete version of the usual derivative of a function.
- Differences are also a particularly simple way to **detrend** a time series

# Differences

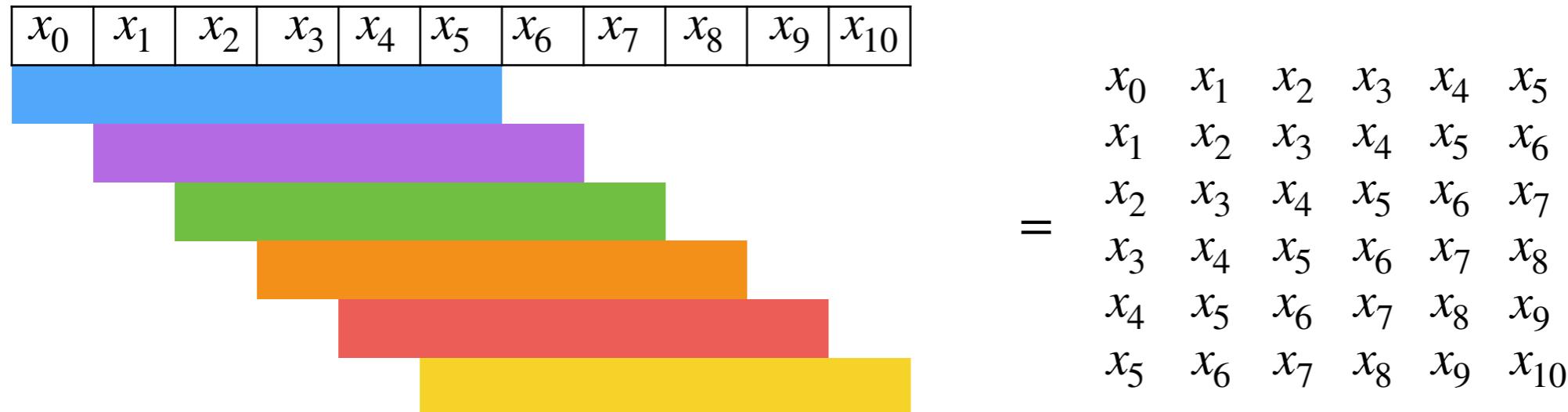


# Differences



# Windowing

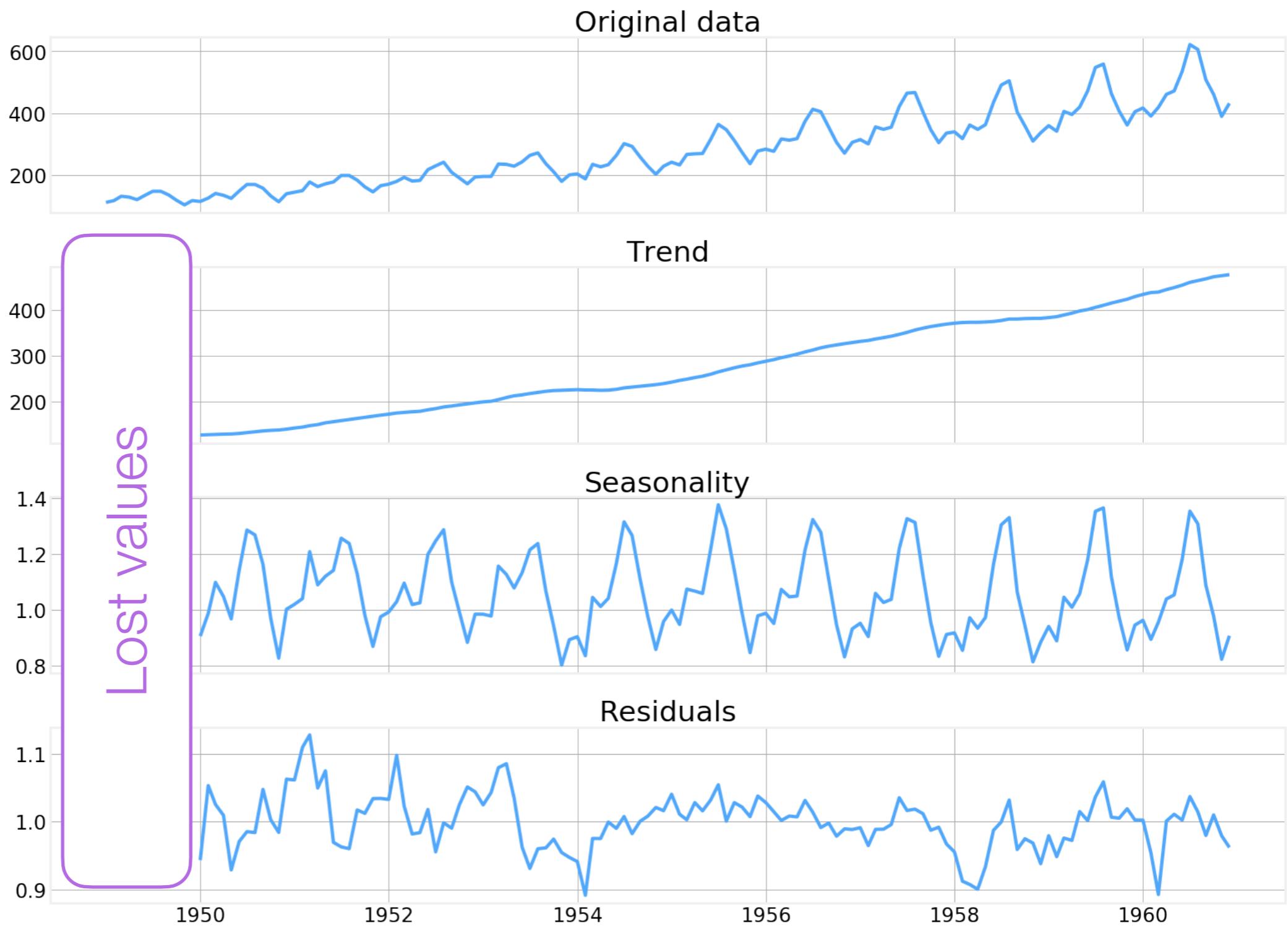
- When analyzing the temporal behavior of a signal, we often need to evaluate if specific quantities are **time varying** or not
- A common approach is to use **sliding windows** of a given length to evaluate the required values
- So a sliding window of width **6** on a series of length **11** would look like:



- and we would calculate the metric of interest **within each window**.

# Windowing

One common approach is to place all “**lost values**” at the beginning as it avoids “**future leaking**” when splitting the dataset

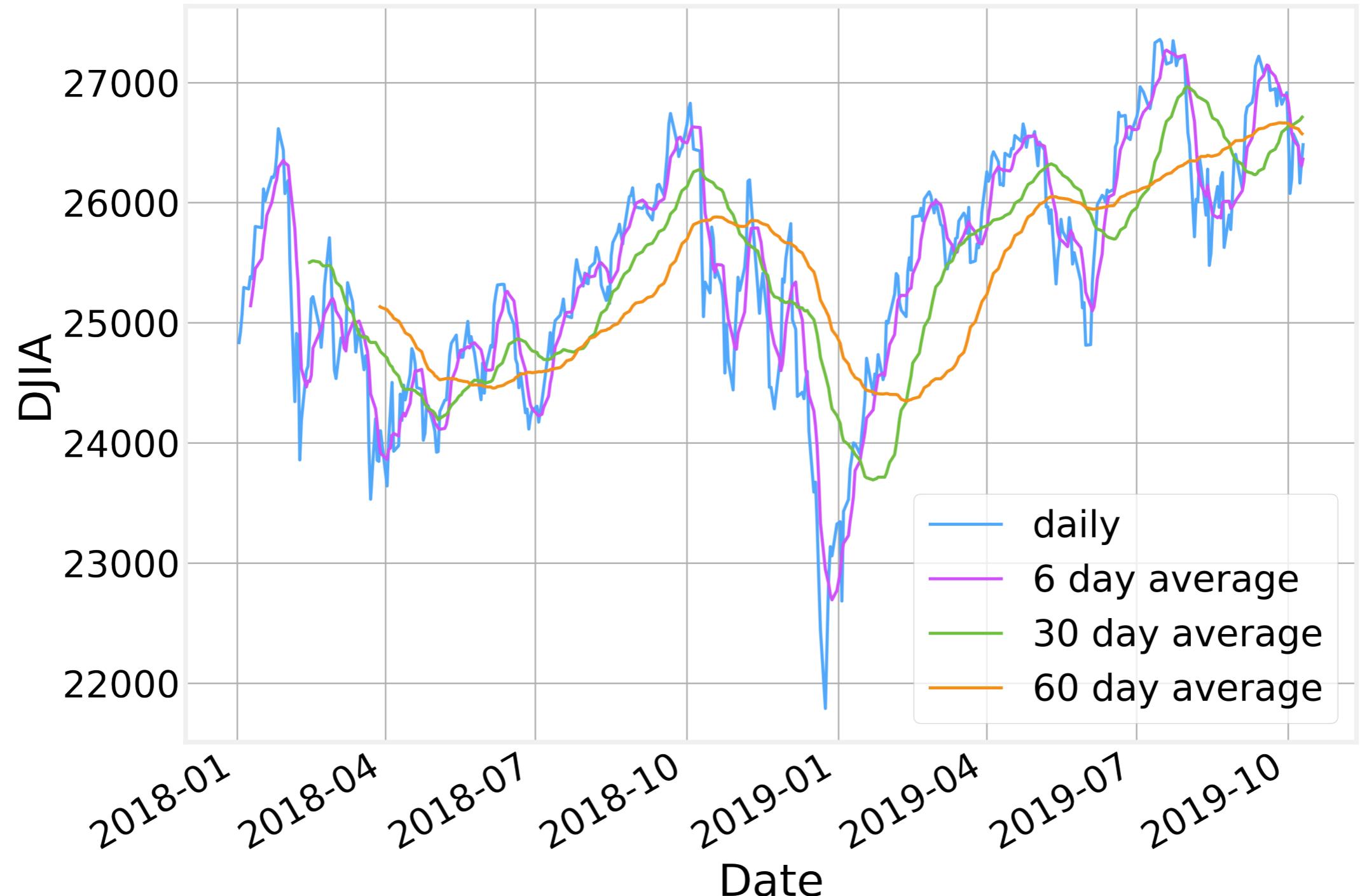


# Running Values

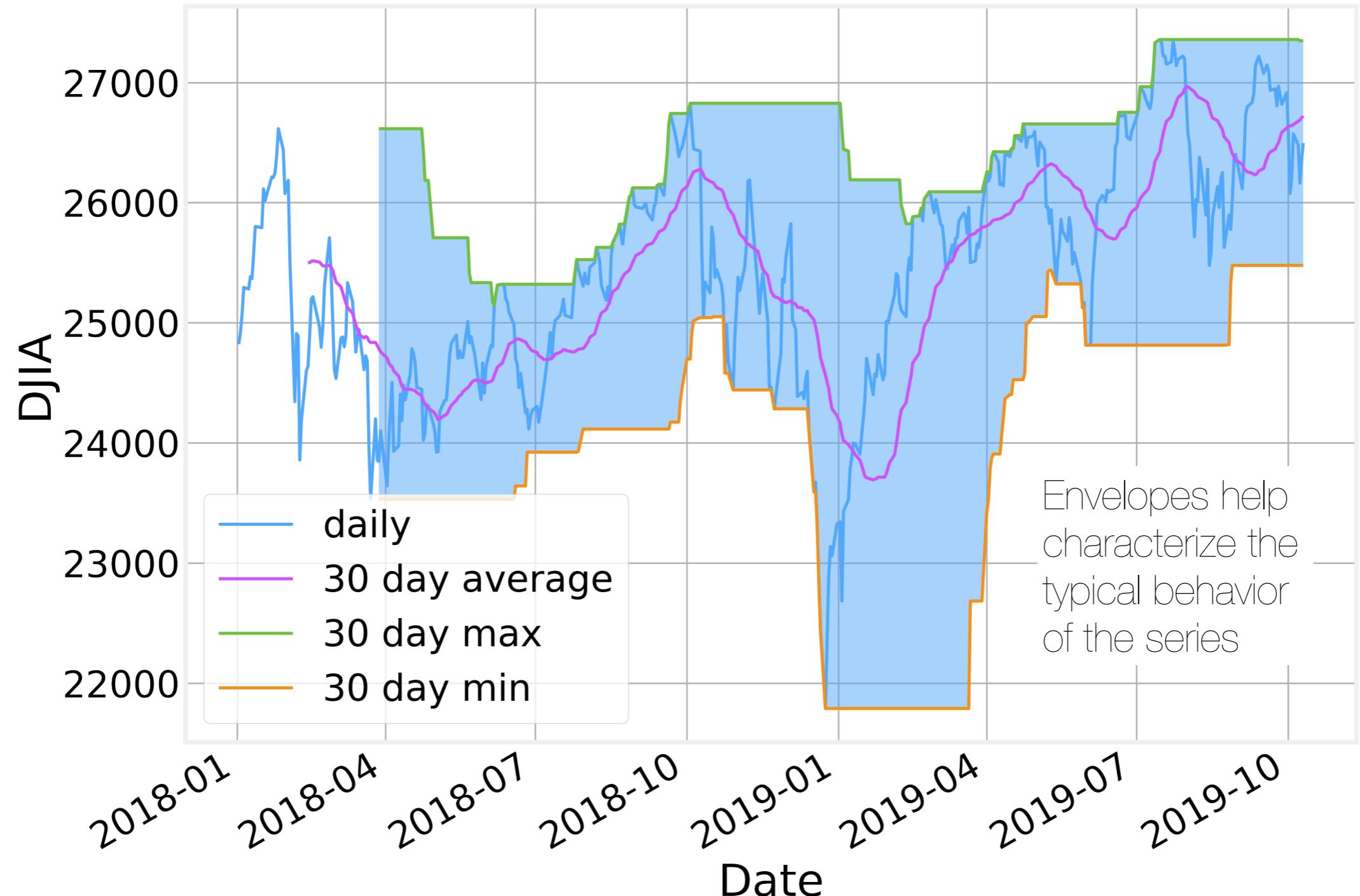
---

- In the first part of the lecture, we already used **running averages** to **detrend** a time series
- Other common metrics are:
  - Variance
  - Maximum value
  - Minimum value
  - etc...
- One important detail to note is that while using windowing to calculate running values **we “lose” a number of points** equal to the width of the window
- Depending on the application we can choose to place the missing values in either or (or even both) extremes of the time interval

# Running Values



# Envelopes

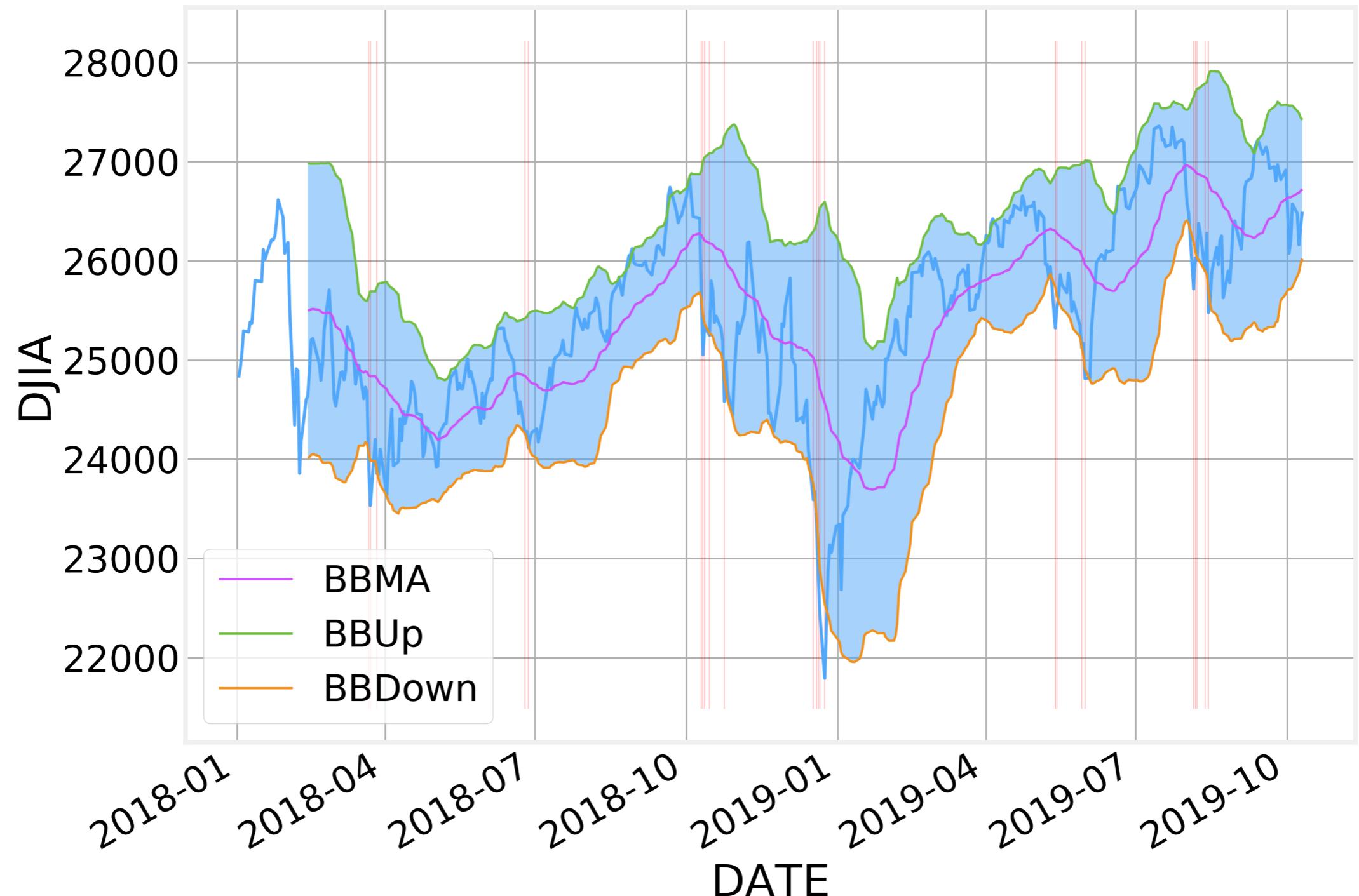


# Bollinger Bands

[https://en.wikipedia.org/wiki/Bollinger\\_Bands](https://en.wikipedia.org/wiki/Bollinger_Bands)

- A common use for application for running values is the calculation of **Bollinger Bands**.
- Introduced by **John Bollinger** in the 1980s as a complement to more traditional time series technical analysis techniques.
- **Bollinger Bands** are defined by two components:
  - A  $N$  period moving average,  $\mu_N$
  - The area  $K$  standard deviations above and below the moving average  $\mu_N \pm K\sigma_N$
- Both  $\mu_N$  and  $\sigma_N$  are computed on a **running window** of size  $N$
- The values of  $N$  and  $K$  are application specific. For stock trading,  $N = 20$  and  $K = 2$  are typical values.
- Whenever the time series steps out of the Bollinger Band that's a clear indication of a **change in the temporal behavior**.

# Bollinger Bands

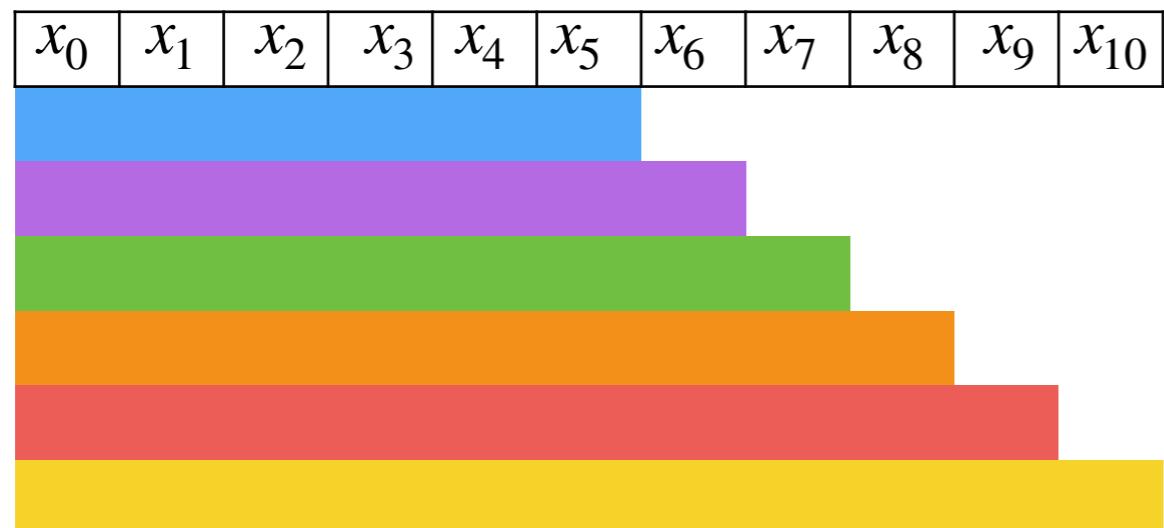


# Exponential Running Average

- One alternative to a simple running average is **Exponential Smoothing**
- The exponentially "smooth" version of a time series is given by:

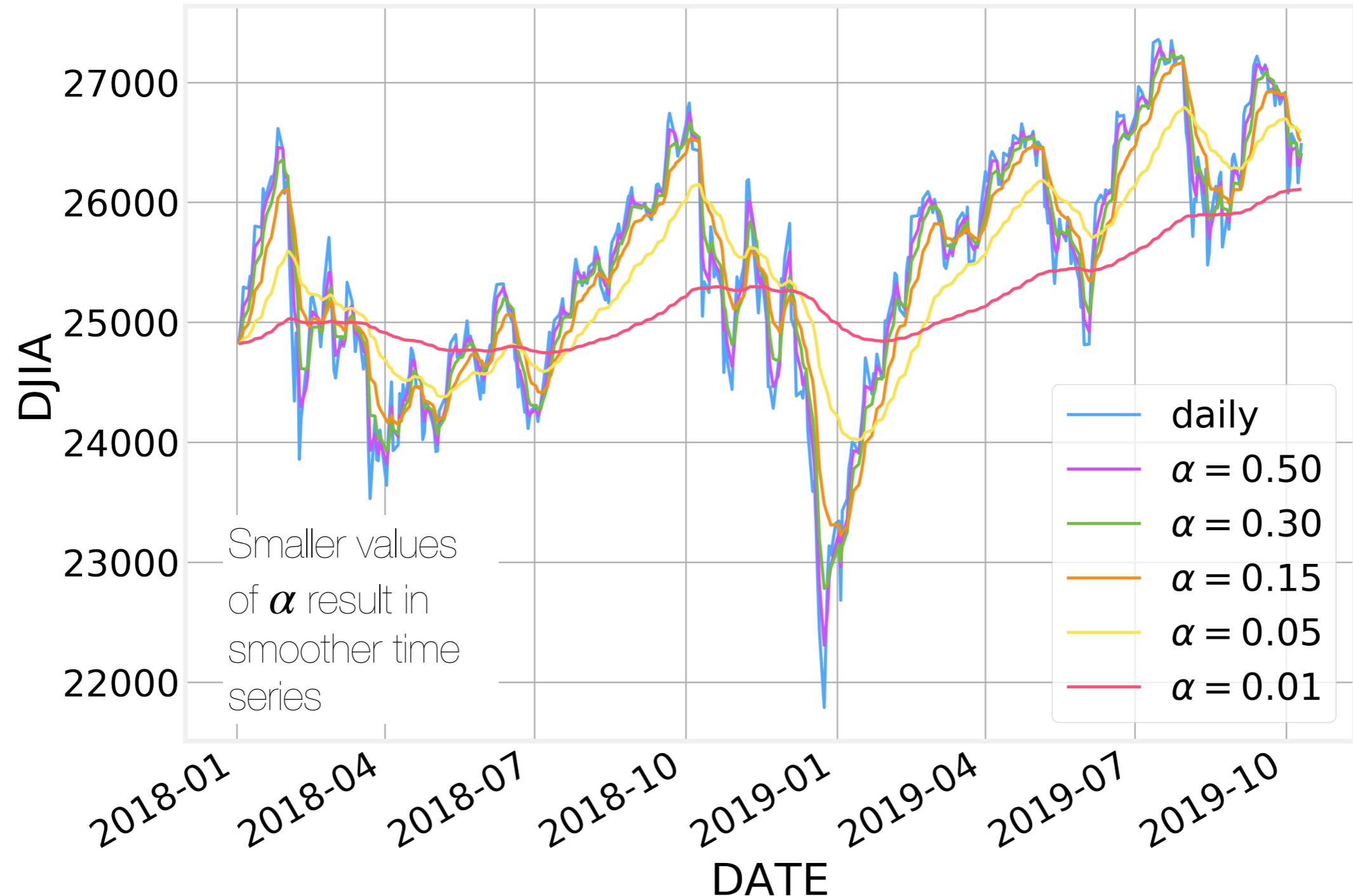
$$z_t = \alpha x_t + (1 - \alpha) z_{t-1}$$

- The smaller the value of the weight  $\alpha$ , the less influence each point has on the transformed time series.
- Each point depends implicitly on **all previous points**



$$\begin{pmatrix} \alpha & & & & \\ \alpha(1-\alpha)^1 & \alpha & & & \\ \alpha(1-\alpha)^2 & \alpha(1-\alpha)^1 & \alpha & & \\ \vdots & \vdots & \vdots & \ddots & \\ \alpha(1-\alpha)^{n-1} & \alpha(1-\alpha)^{n-2} & \alpha(1-\alpha)^{n-3} & & \alpha \end{pmatrix}$$

# Exponential Running Average



# Forecasting

---

- We can also use the Exponential Moving Averages as a simple forecasting tool.

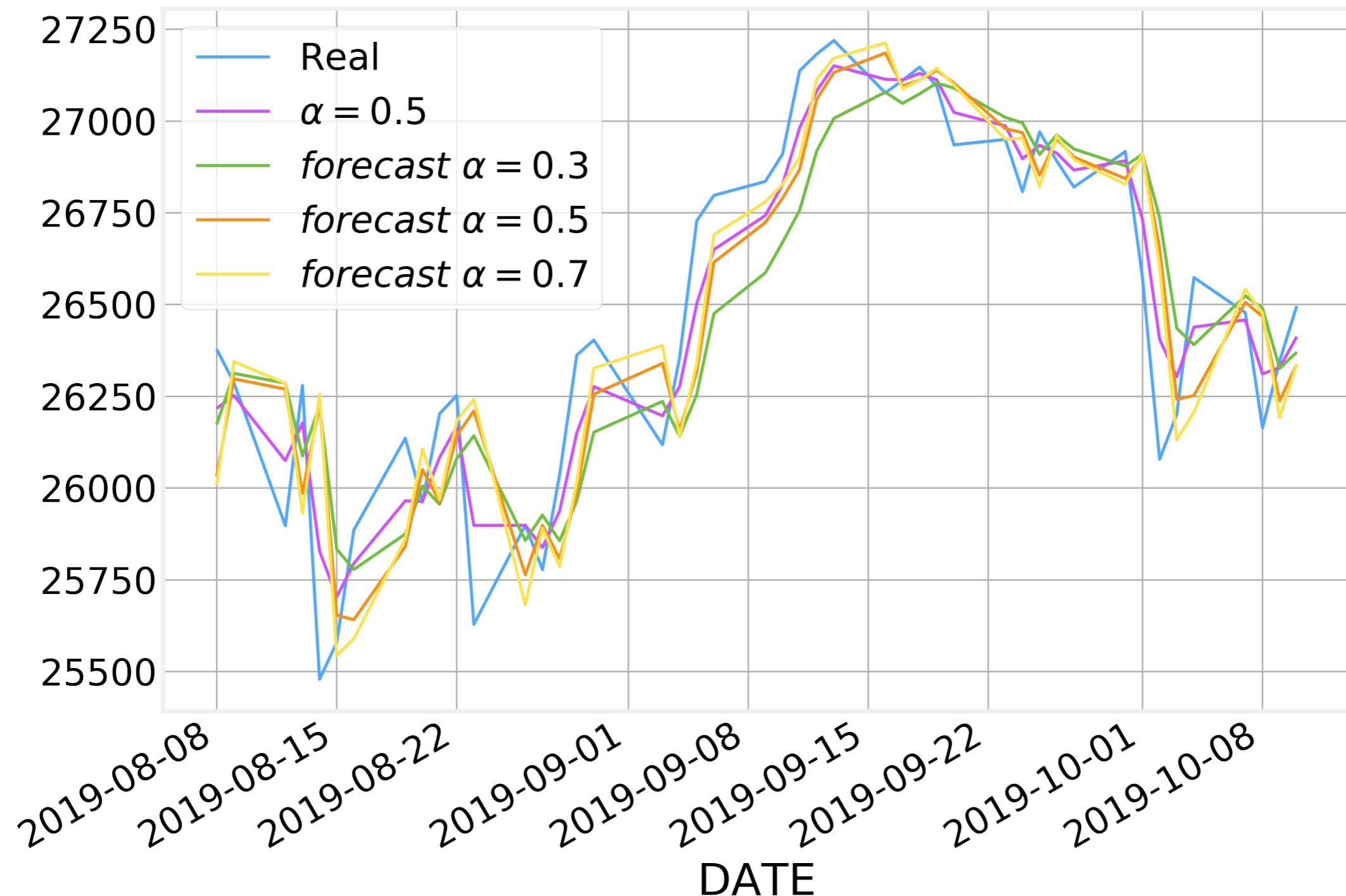
- The value at time  $t + 1$  is given by:

$$z_{t+1} = \alpha x_t + (1 - \alpha) z_t$$

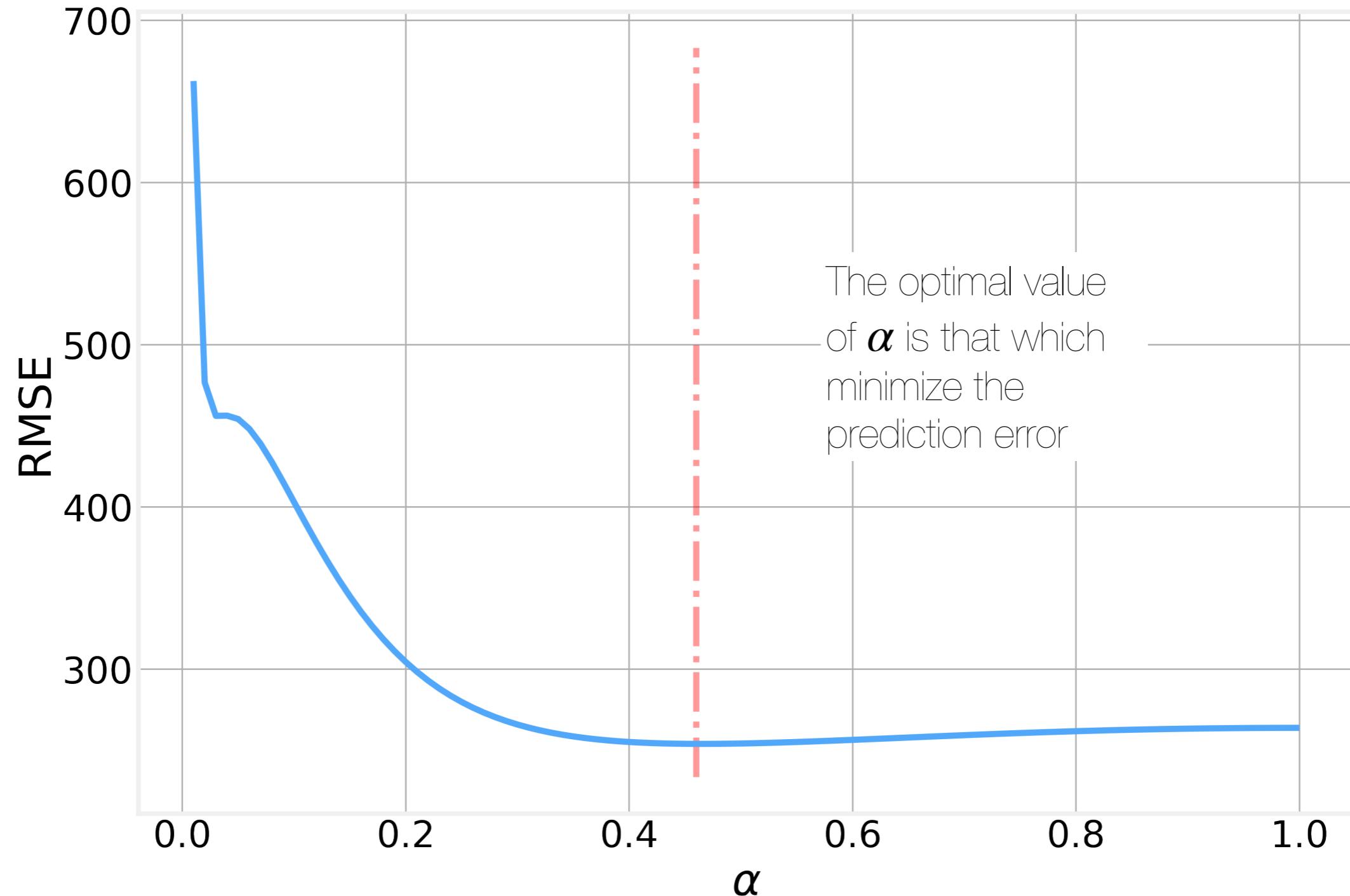
- Which we can consider to be a prediction on the value of  $x_{t+1}$ , based on the current value of  $z_t$  and some fraction of our current error value  $x_t - z_t$ :

$$z_{t+1} = z_t + \alpha (x_t - z_t)$$

# Forecasting

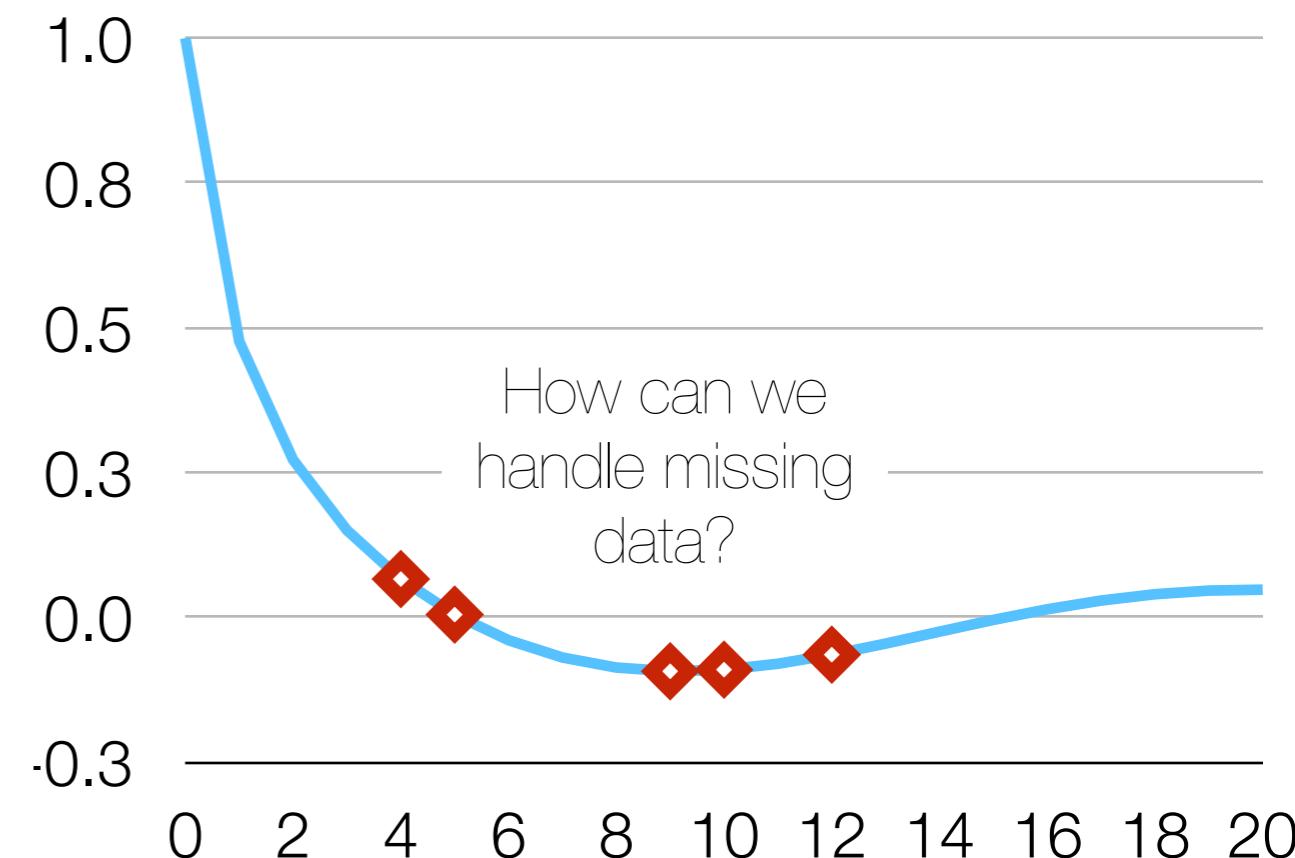


# Forecasting

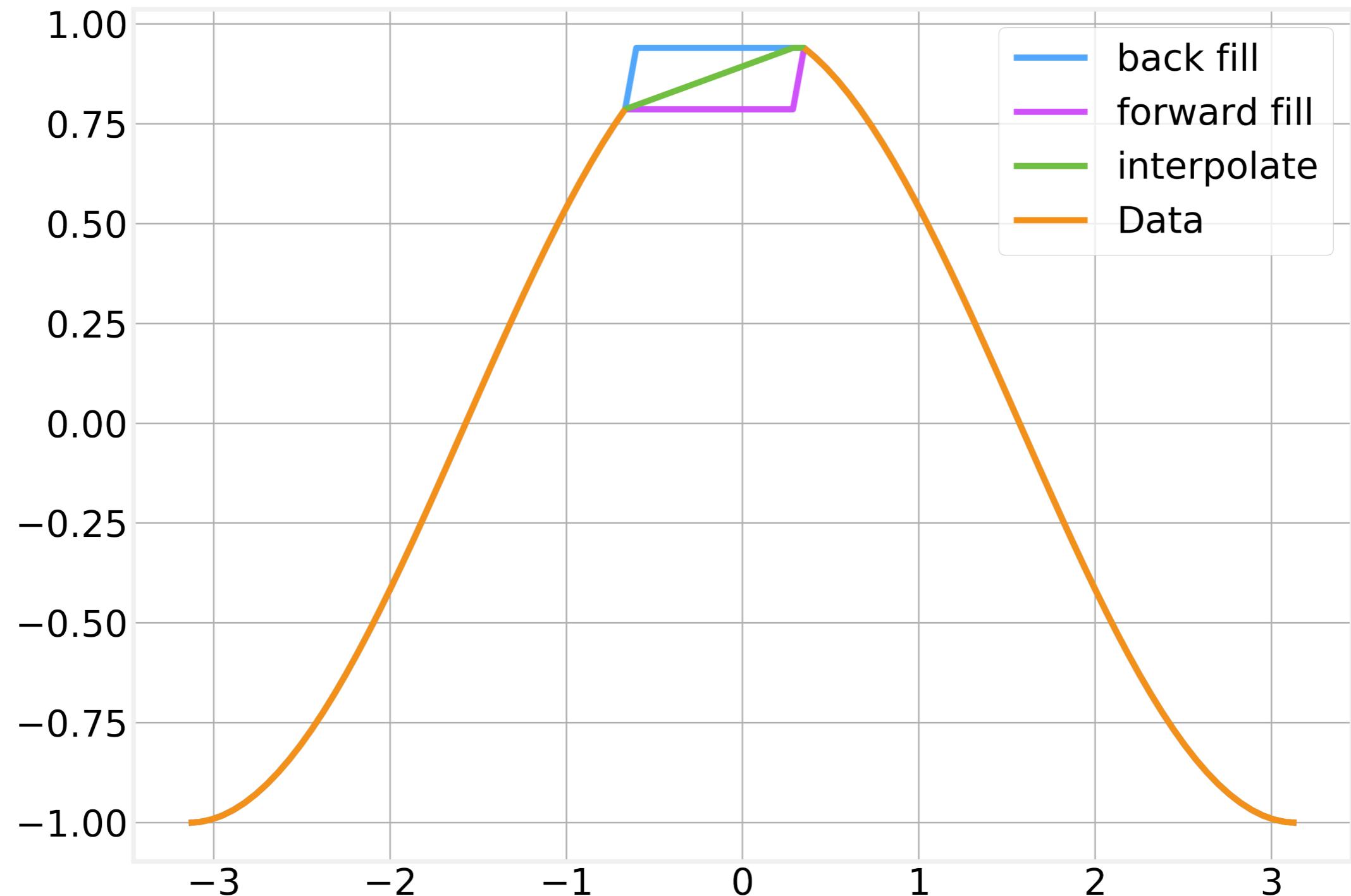


# Fill methods

- Sometimes the time series is **incomplete**.
- Missing data points can be due to data corruption, data collection issues, etc.
- Missing values are represented as **nan**
- Several techniques have been developed to handle this case:
  - **forward fill** - keep the last valid value
  - **back fill** - keep the next valid value
  - **interpolate** - add values by interpolating between the previous and the next value
  - **imputation** - add values based on what we expect the missing values to be



# Fill methods

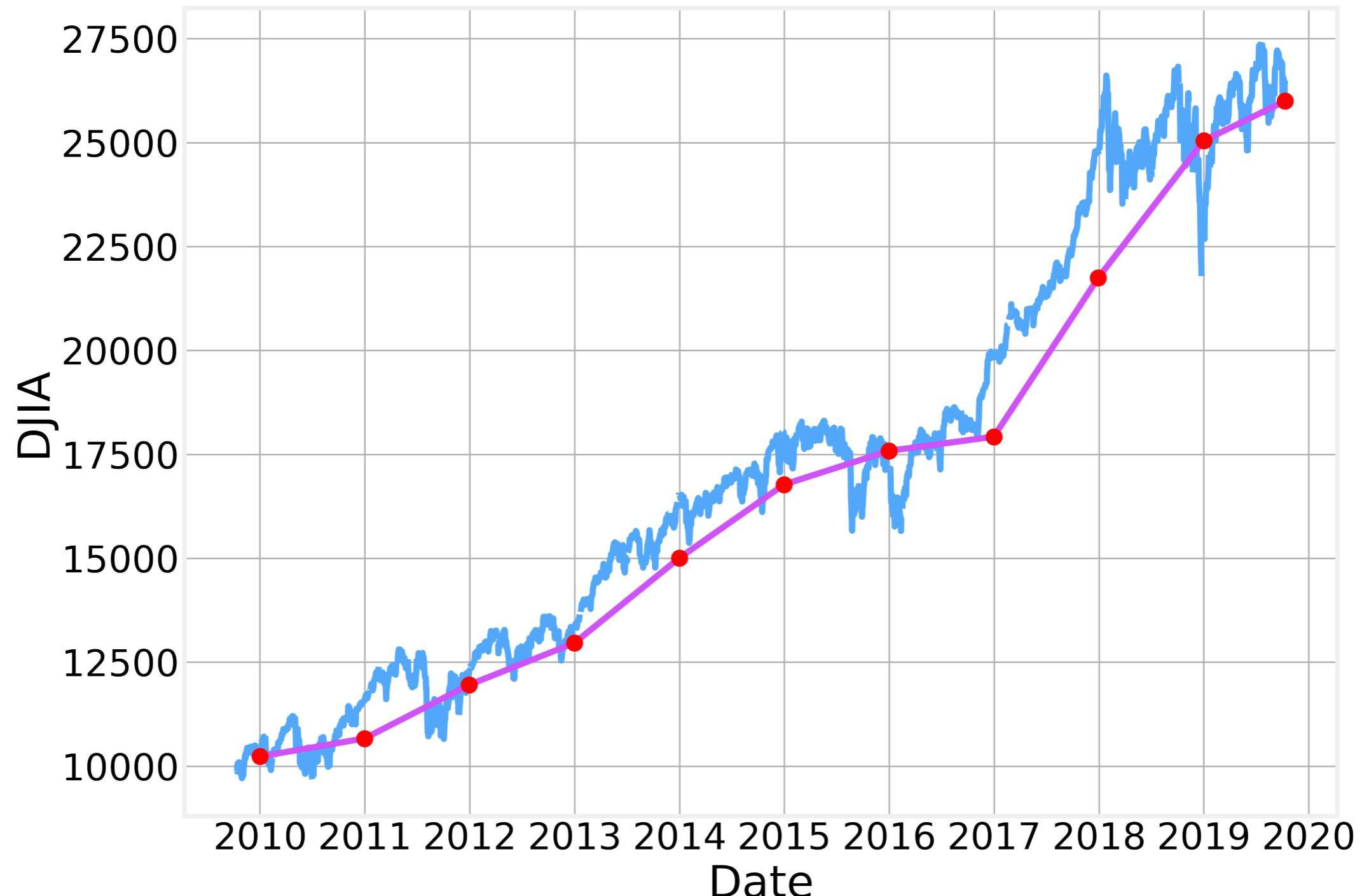


# Resampling

---

- Time series typically have an **intrinsic time scale** at which the data was collected: ticks, seconds, days, months, etc
- In many cases, our analysis requires that we **resample** the data to a different time scale
- Resampling to a longer timescale is relatively simple and similar to aggregation:
  - Transforming from daily to weekly frequency requires simply aggregating by week
- Resampling to shorter timescales requires **interpolation or imputation** to make up for the missing values
  - Going from weekly to daily frequency requires **specifying how to allocate** the values for each day of the week

# Resampling



# Jackknife Estimation

---

- Also known as '**leave one out**' estimation
- Commonly used for mean and variance estimates
- The **Jackknife estimate** of a parameter is the average value of the parameter calculated by omitting each of the values one at a time.
- If  $\mu_i$  is the mean calculated by omitting the  $i^{th}$  value, then the Jackknife estimate of the mean is given by:

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^N \mu_i$$

- And the **variance** of the estimate is:

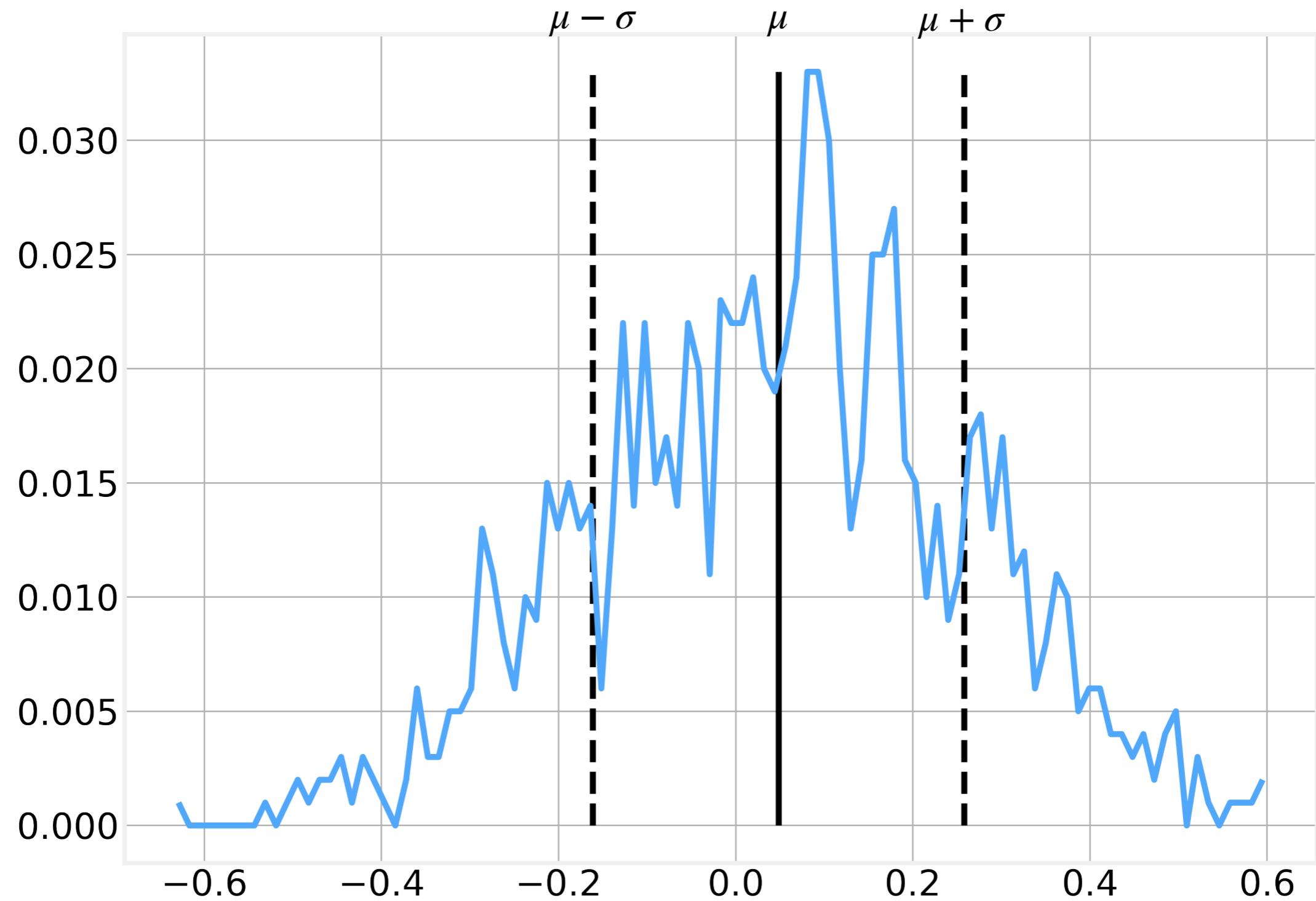
$$\hat{\sigma}(\hat{\mu}) = \frac{N-1}{N} \sum_{i=1}^N (\mu_i - \hat{\mu})^2$$

# Bootstrapping

---

- Bootstrapping is another alternative to **estimate statistical properties** such as mean and variance
- Bootstrapping measures the desired property in a **large number of samples** (with replacement), of the observed dataset.
- Each sample has **equal size** to the observed dataset.
- From the entire population of samples, the **empirical bootstrap distribution** of the expected values can be obtained to provide information about the distribution in the total population

# Bootstrapping





Code - Transformations

<https://github.com/DataForScience/Timeseries>



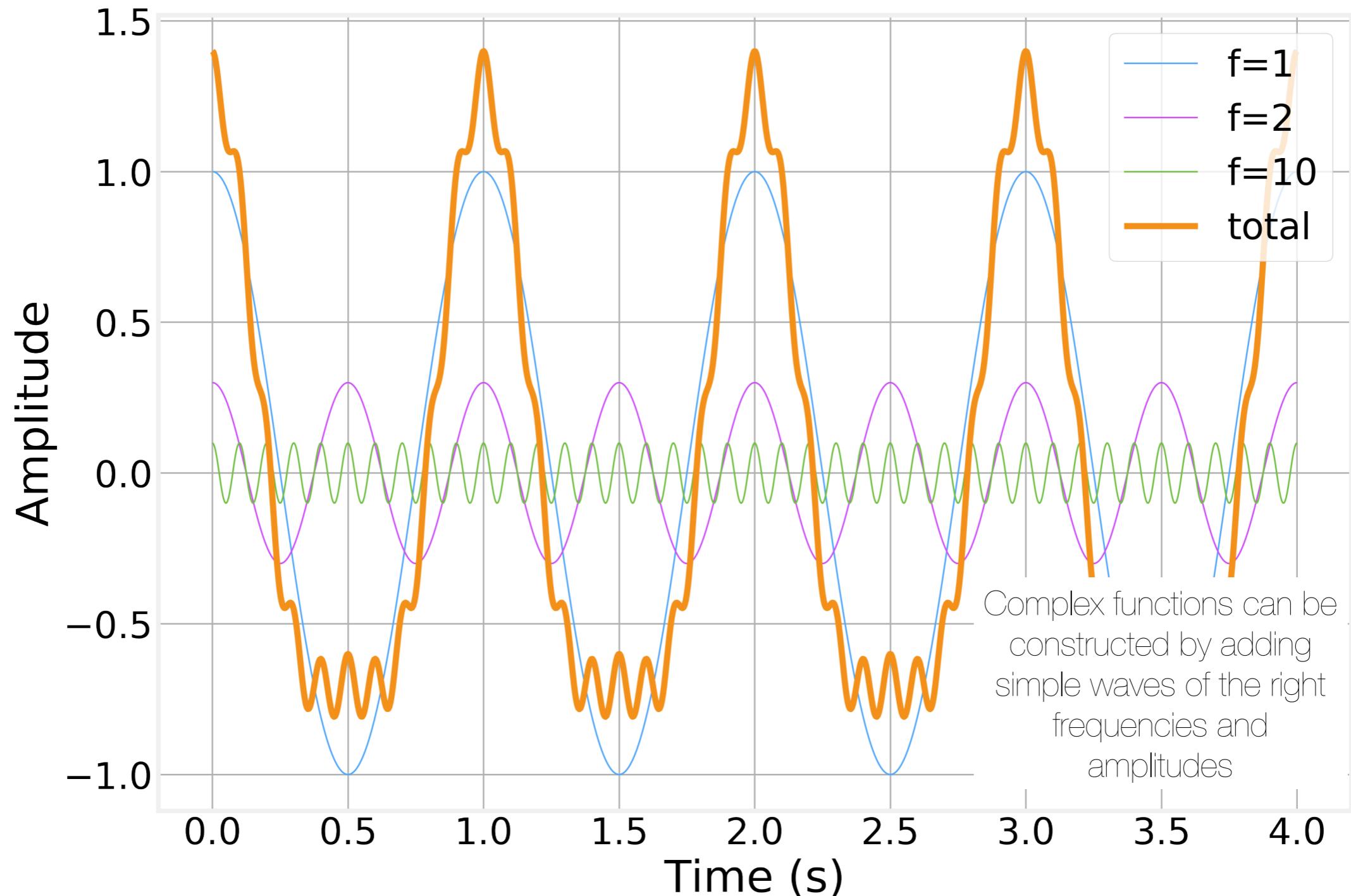
## Lesson III: Fourier Transformations and Filtering

# Frequency Domain

---

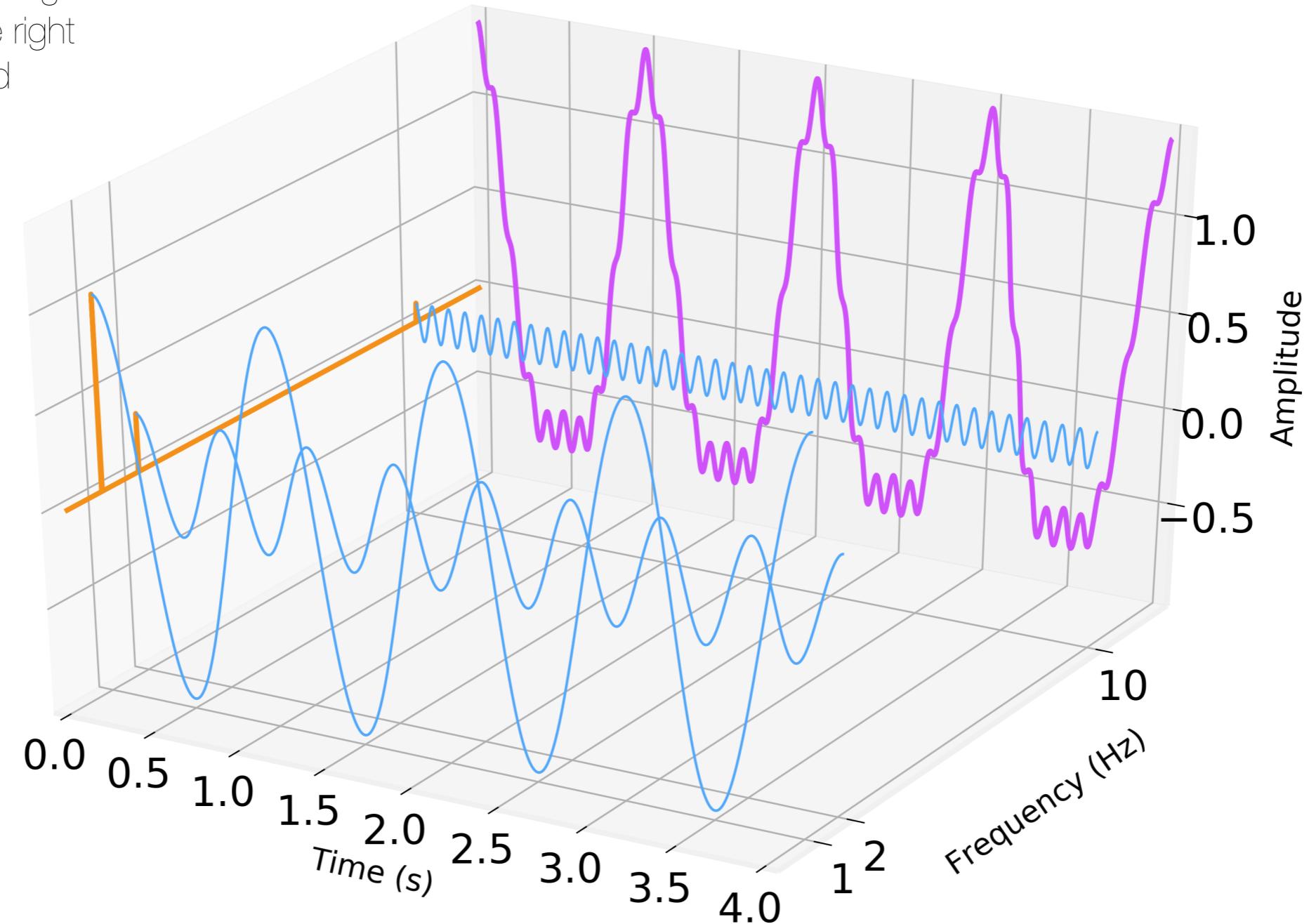
- So far we have focused on the natural (time based) representation of a time series.
- An alternative representation is based on frequencies and was first introduced by [Jean Fourier](#) in 1807
- Fourier showed that periodic functions can be decomposed as a [sum of trigonometric functions](#)
- Fourier's original result was later extended to [all functions](#)
- The [Discrete Fourier Transforms](#) provides us with a simple and convenient way to move from the [time-domain](#) to the [frequency-domain](#) and back.

# Adding Frequencies



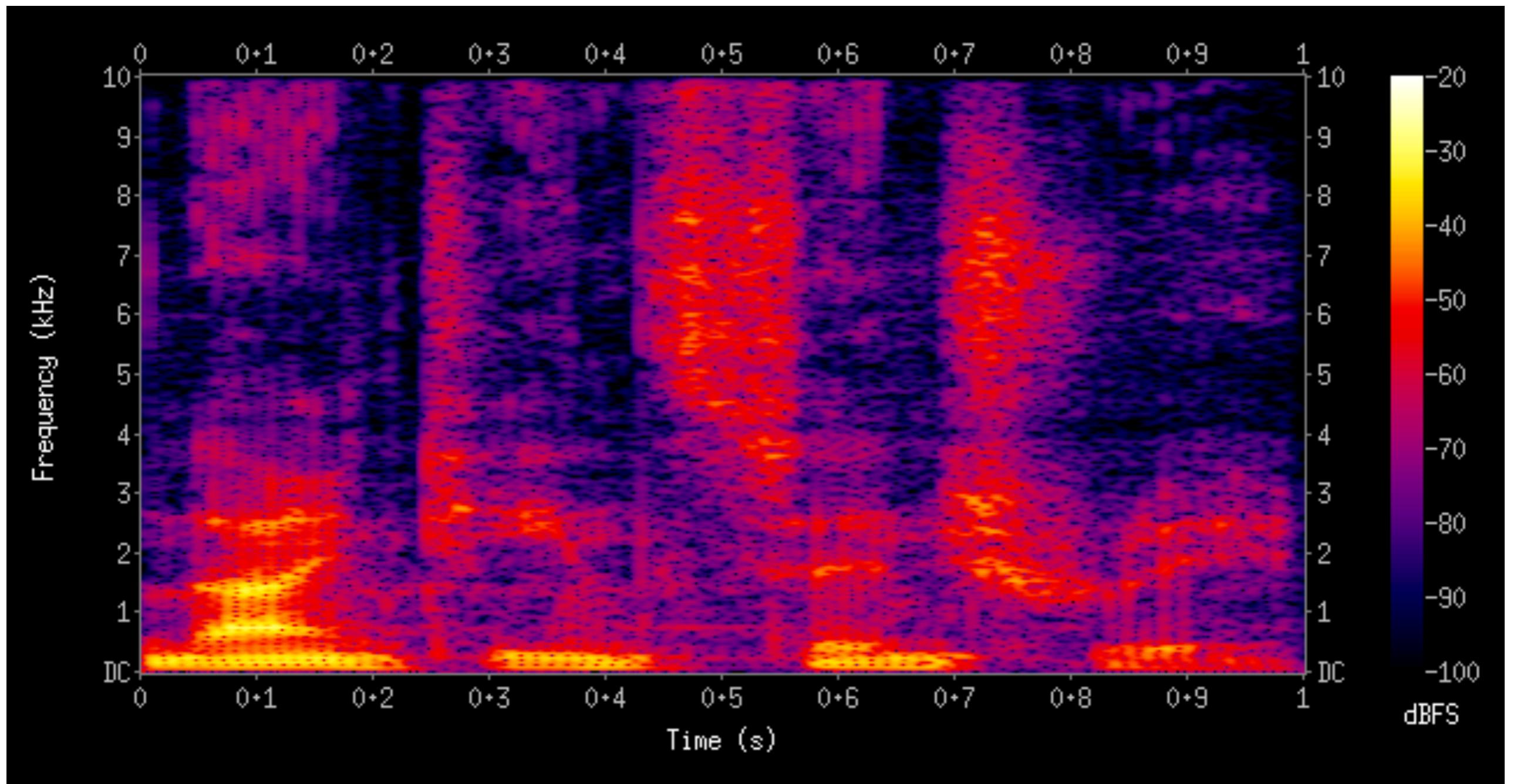
# 3D Visualization

Complex functions can be constructed by adding simple waves of the right frequencies and amplitudes



# Spectrogram

<https://en.wikipedia.org/wiki/Spectrogram>



# (Discrete) Fourier Transform

[https://en.wikipedia.org/wiki/Discrete-time\\_Fourier\\_transform](https://en.wikipedia.org/wiki/Discrete-time_Fourier_transform)

- The **DFT** maps a sequence of  $N$  values  $\mathbf{x}_n$  representing a time series  $\mathbf{x}(t)$  into  $N$  complex numbers  $X_k$  defined as:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i \frac{2\pi kn}{N}}$$

- where:

$$e^{i\theta} = \cos(\theta) + i \sin(\theta)$$

- To recover the original values we use the **Inverse DFT**, defined as:

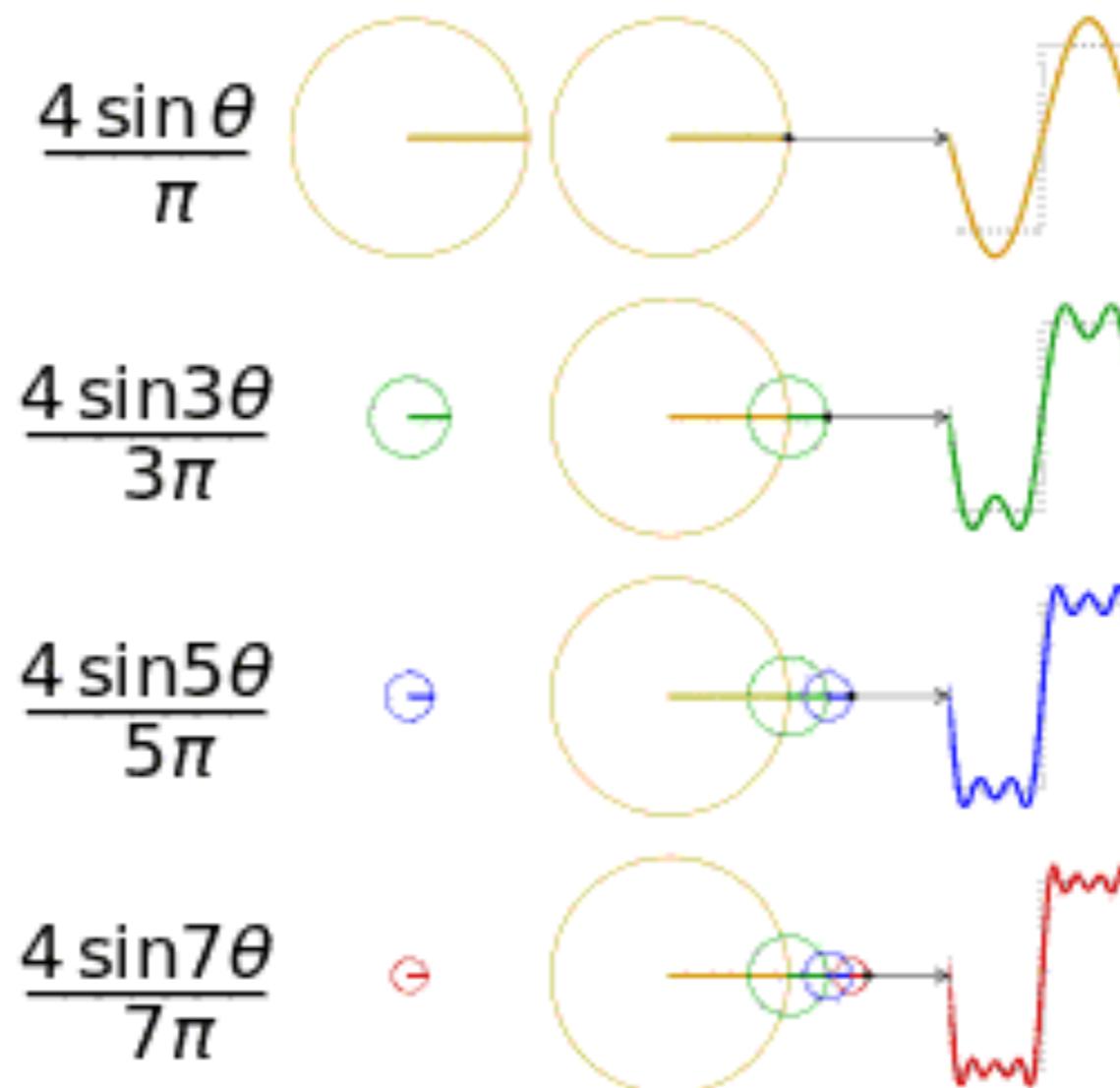
$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{i \frac{2\pi kn}{N}}$$

- The DFT represents the continuous series  $\mathbf{x}(t)$  as a **sum of discrete frequencies**:

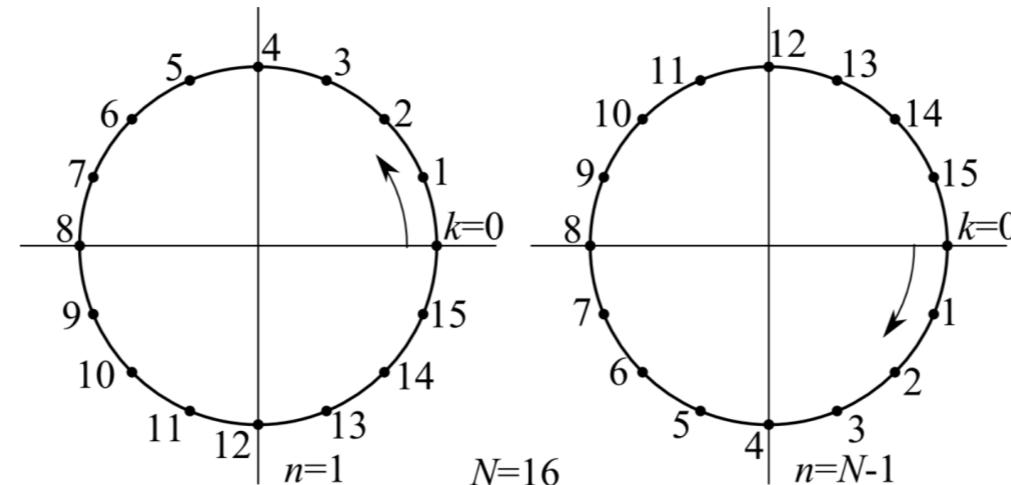
$$\omega_n = 2\pi \frac{k}{N}$$

# (Discrete) Fourier Transform

[https://en.wikipedia.org/wiki/Discrete-time\\_Fourier\\_transform](https://en.wikipedia.org/wiki/Discrete-time_Fourier_transform)



- Python-based ecosystem of open-source software for mathematics, science, and engineering.
- Provides practical implementation of the **Fast Fourier Transform** an efficient algorithm to compute the **DFT** and **IDFT**. See [https://en.wikipedia.org/wiki/Fast\\_Fourier\\_transform](https://en.wikipedia.org/wiki/Fast_Fourier_transform)
- `scipy.fftpack.fft()`/`scipy.fftpack.ifft()` - **DFT** and **IFT**
- `scipy.fftpack.freq()` - return the list of frequencies
- `scipy.fftpack.fftshift()`/`scipy.fftpack.ifftshift()` - Shift the zero-frequency component to the center of the spectrum and back.

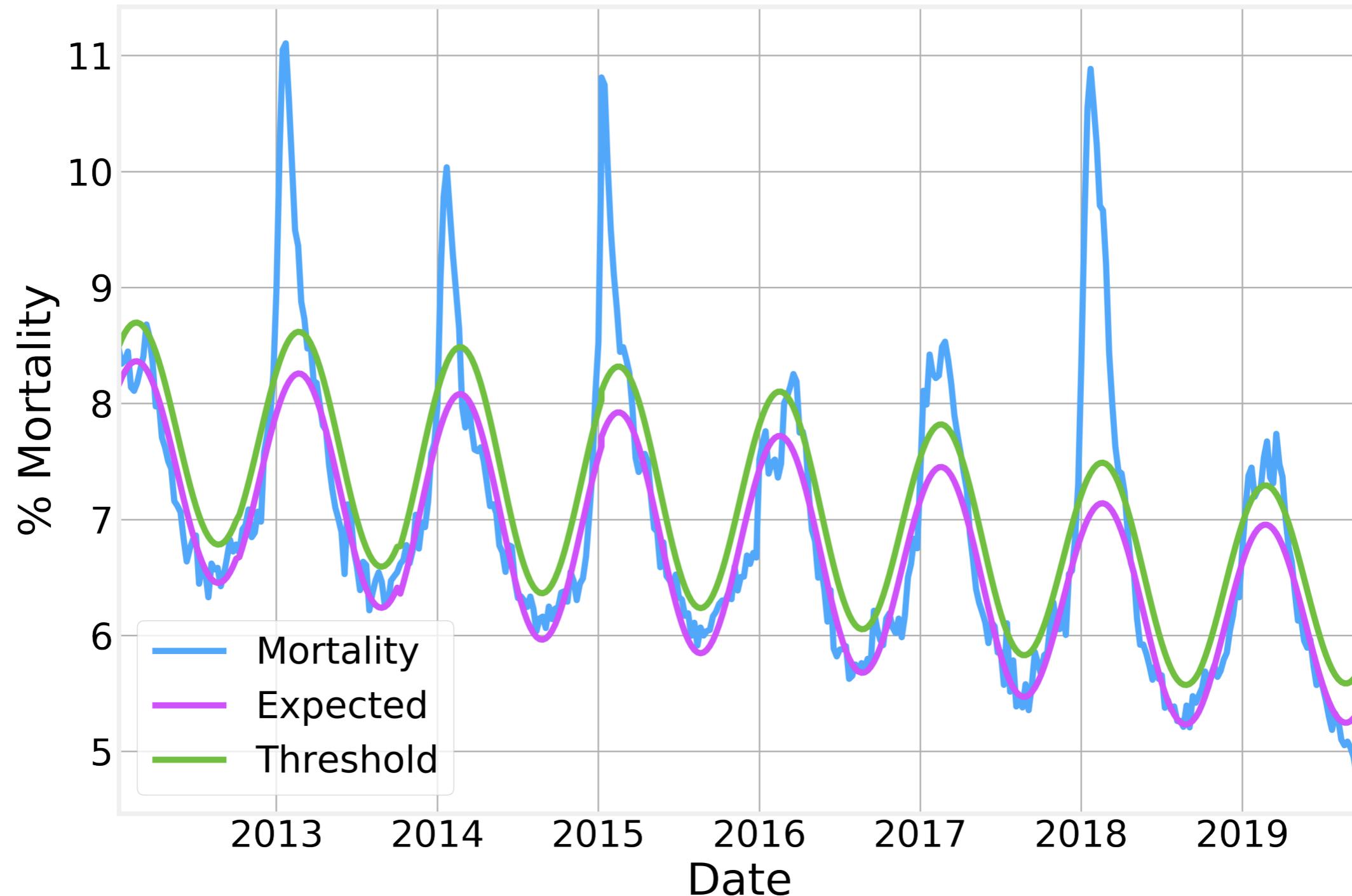


# Filtering

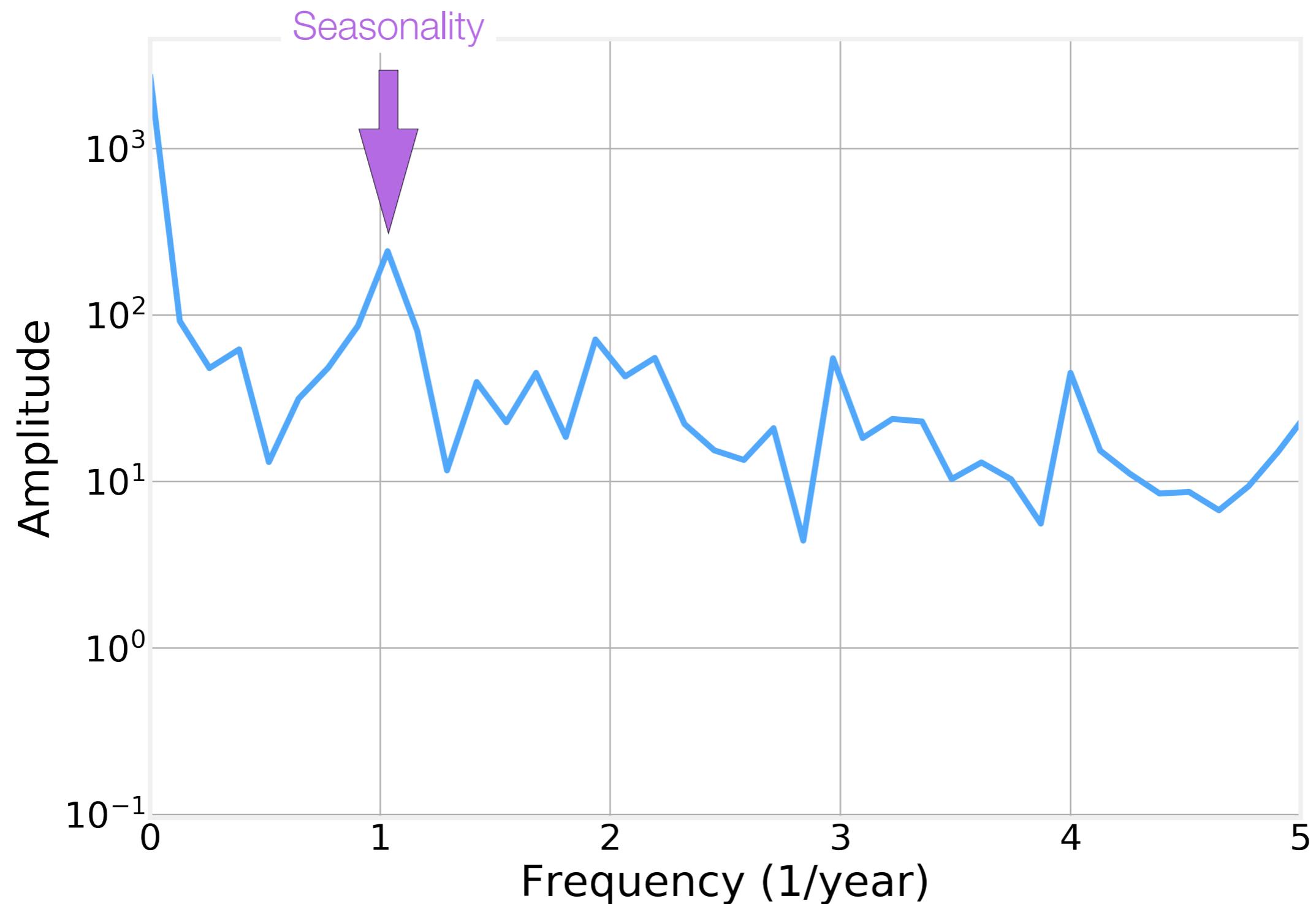
---

- Common applications of **Fourier Analysis** are:
  - **Seasonality** - determine the main frequency underlying a time series
  - **Filtering** - remove higher order frequencies to eliminate noise
  - **Processing** - Several signal processing operations are simpler to compute in the frequency-space

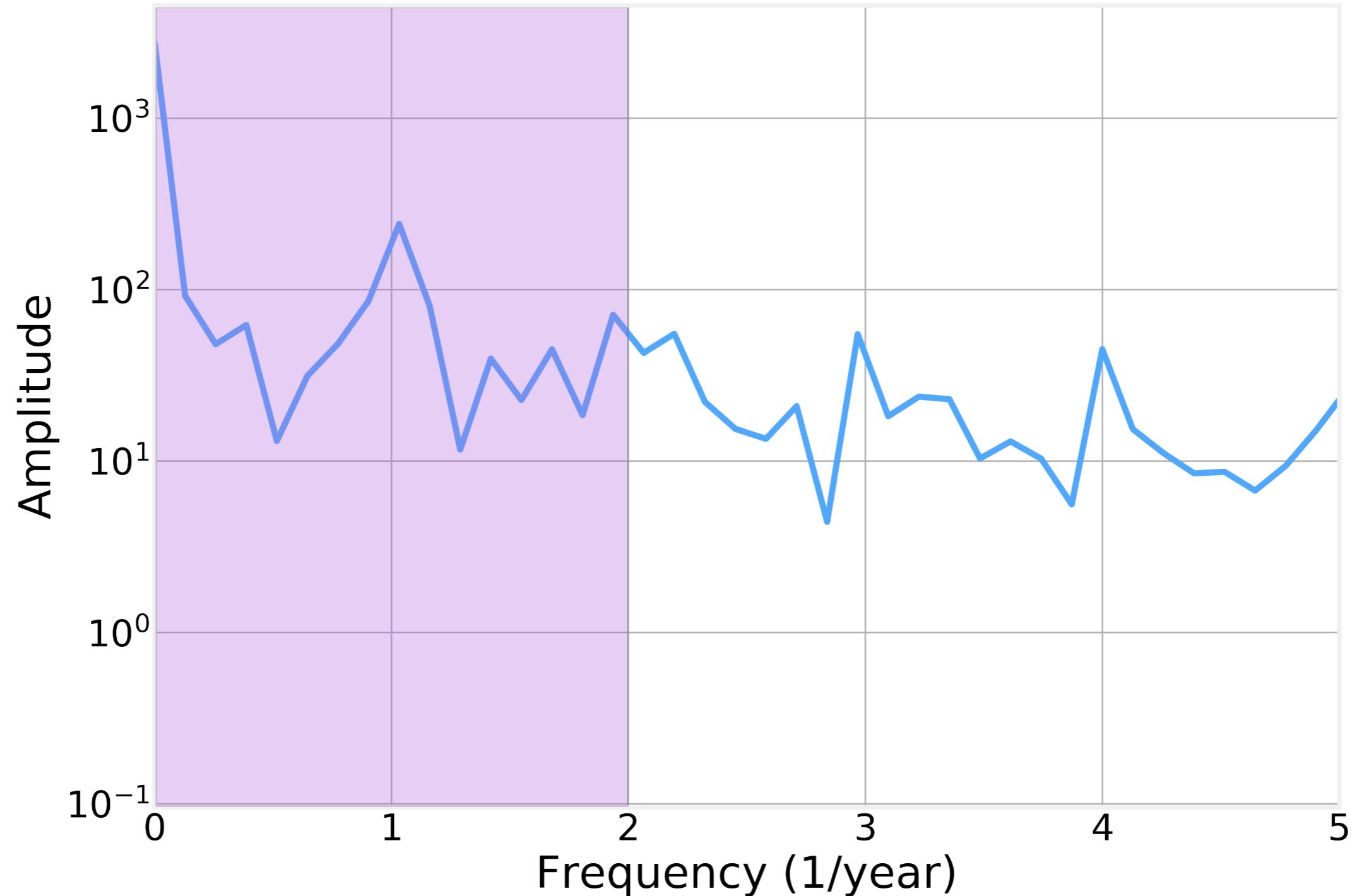
# Filtering



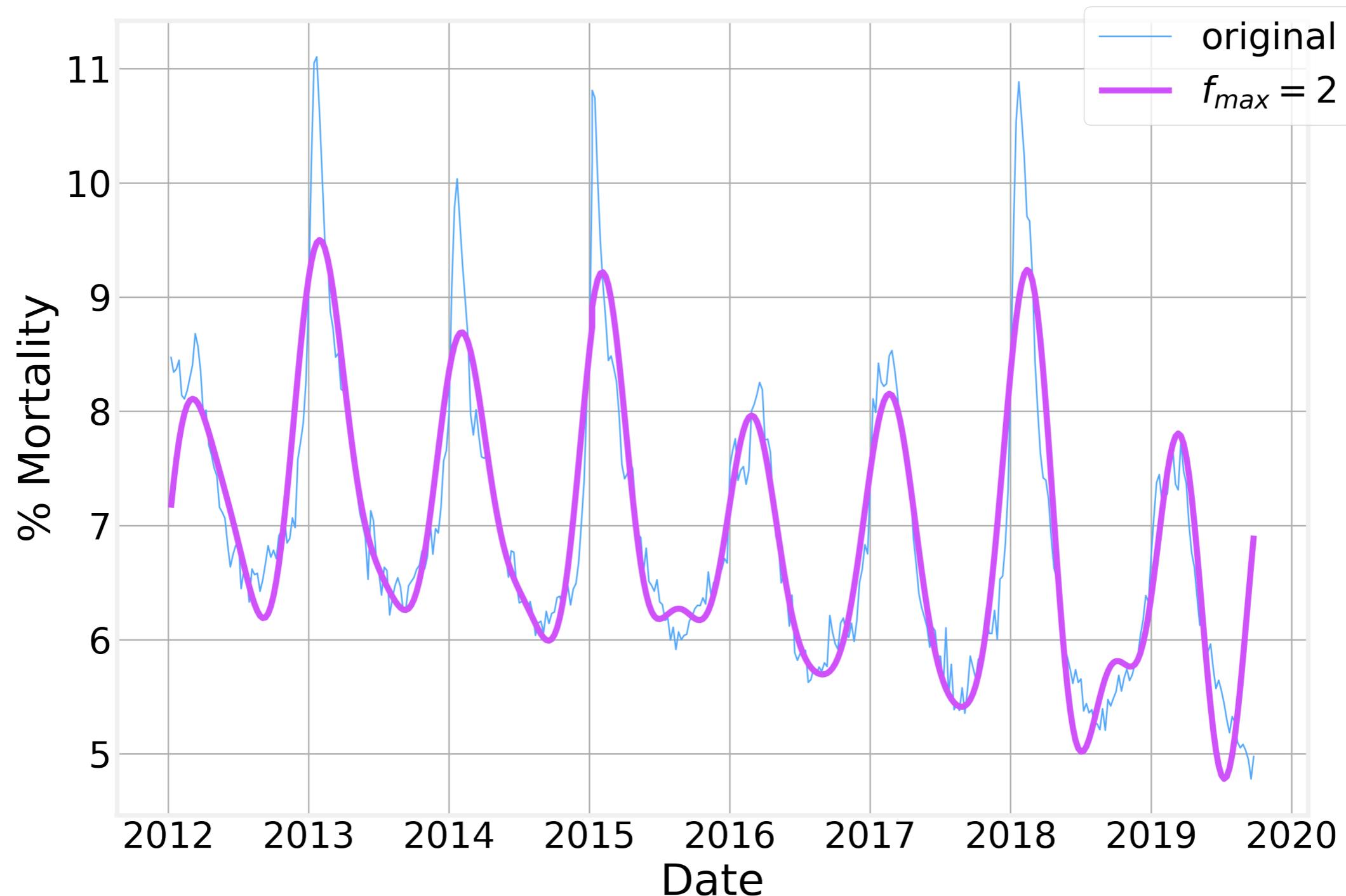
# Filtering



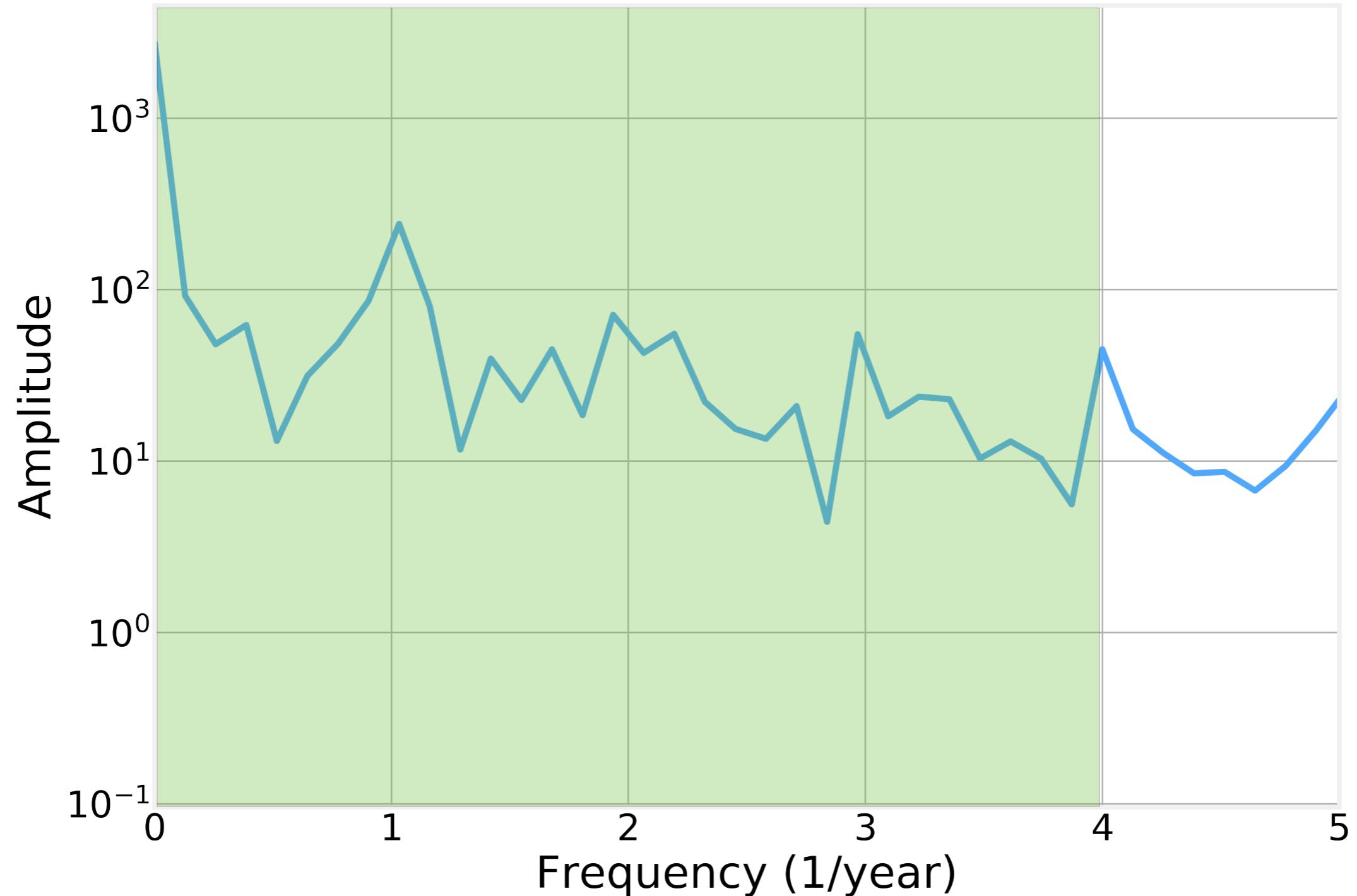
# Filtering



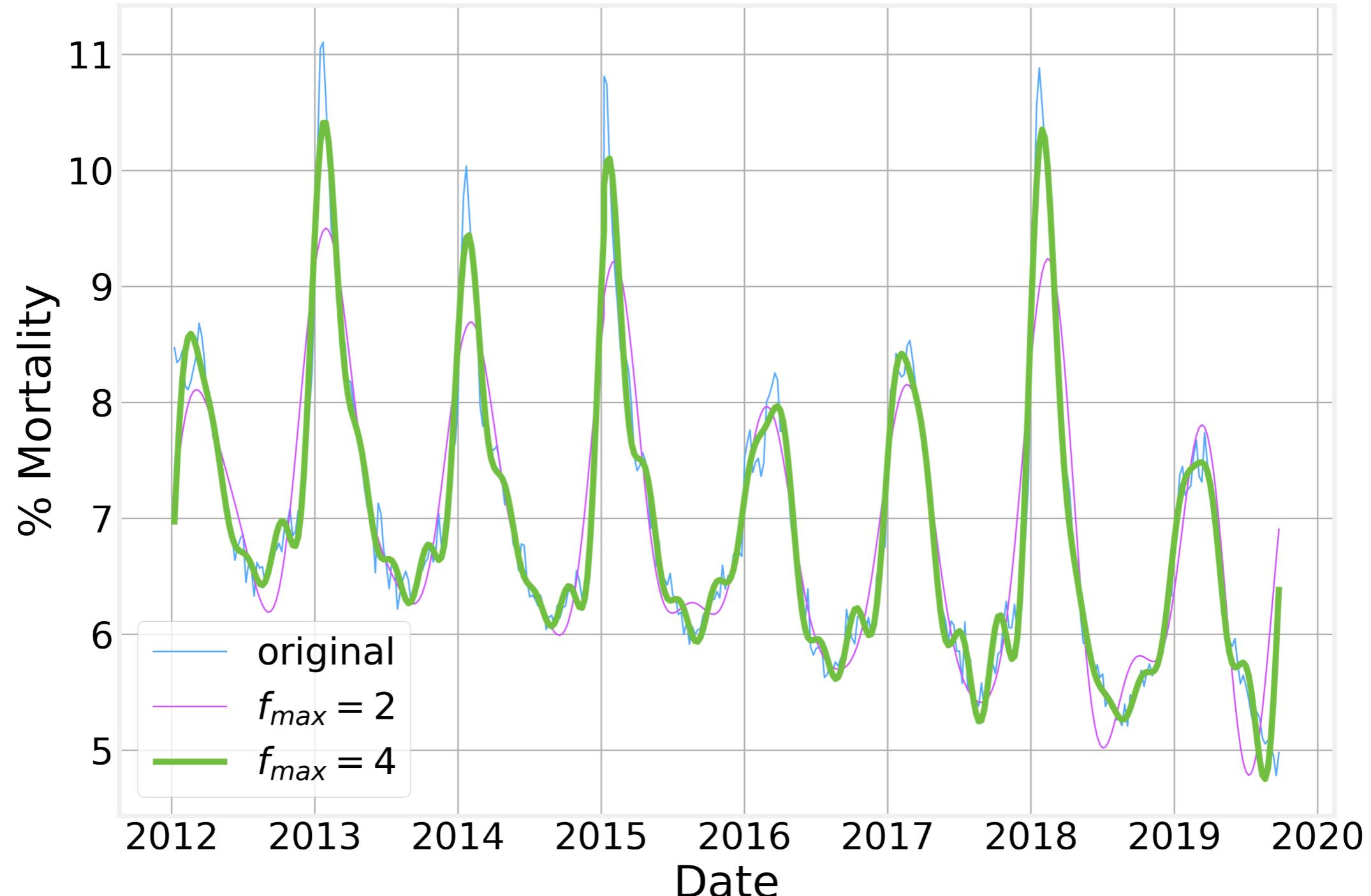
# Filtering



# Filtering



# Filtering



# Extrapolation

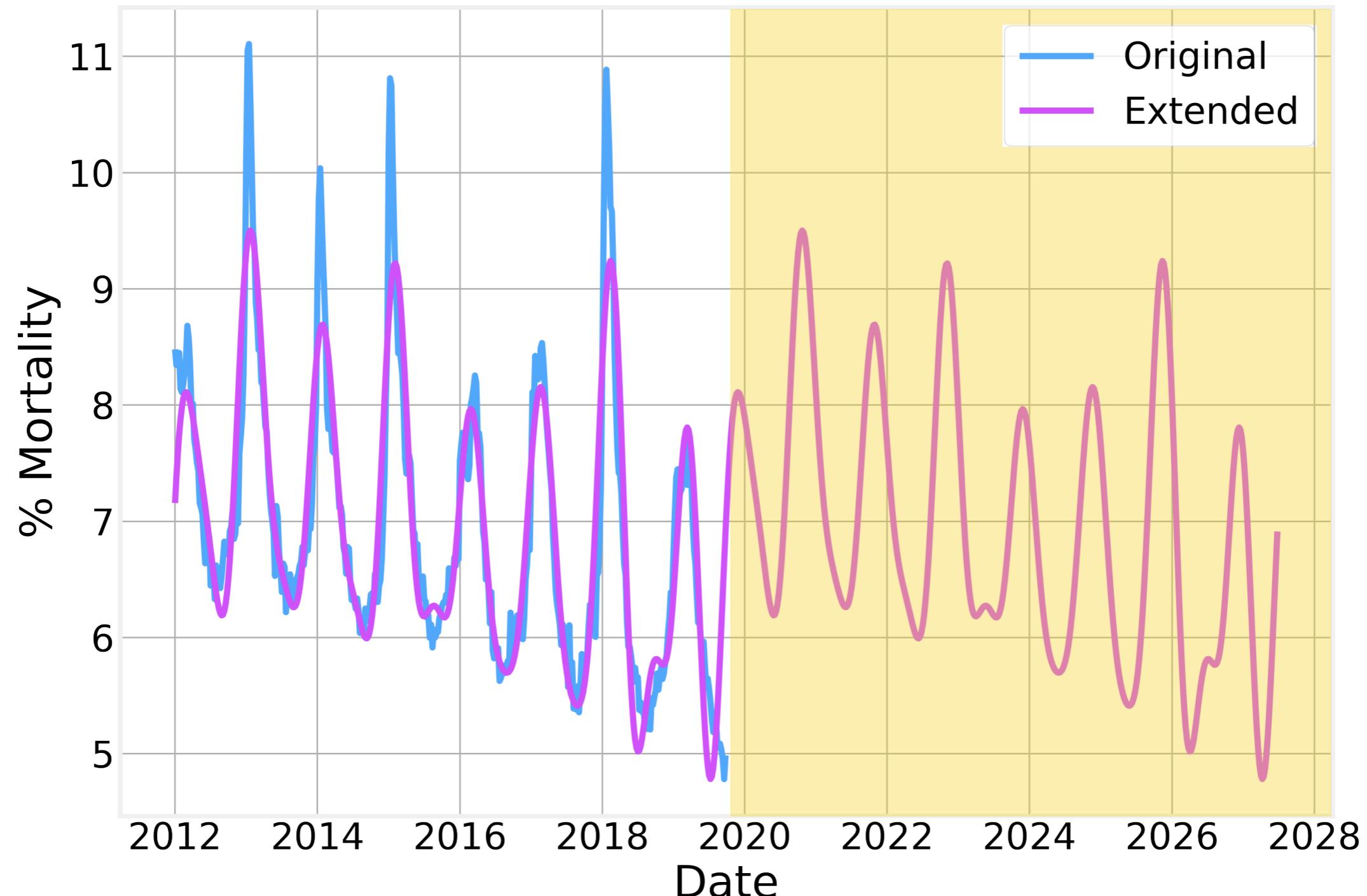
---

- As we saw above, we can recover the original signal from the FFT values by using:

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} x_k e^{i \frac{2\pi k n}{N}}$$

- Where  $n$  is our time variable.
- There's nothing stopping us from **extending** the values of  $n$  **beyond the original domain** of the signal
- The resulting extrapolated values **correspond to a forecast** into the future.

# Extrapolation





Code - Fourier Analysis

<https://github.com/DataForScience/Timeseries>



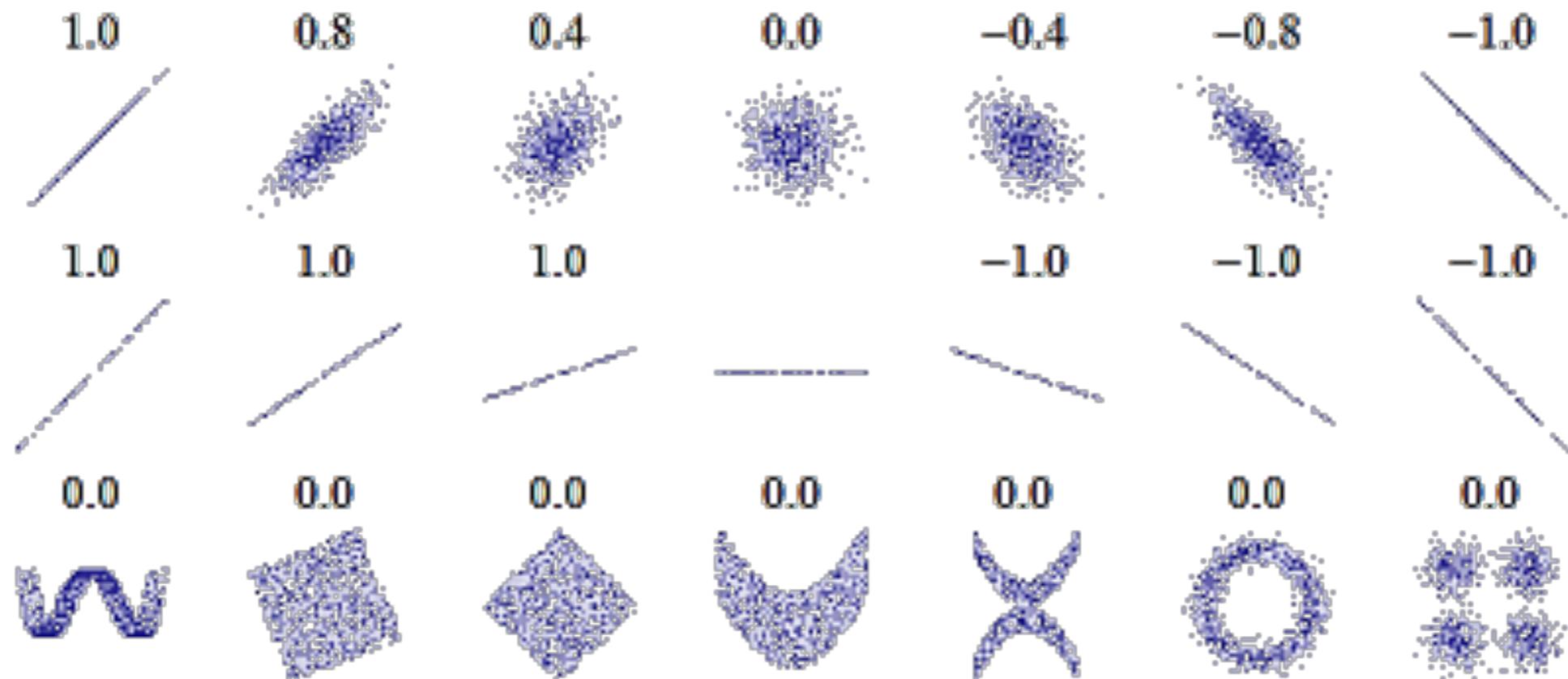
## Lesson IV: Correlations

# Correlation

- Many correlation measures have been proposed over the years
- The most well known one is the **Pearson Correlation**

$$\rho(x, y) = \sum_{i=1}^N \frac{(x_i - \mu_x)(y_i - \mu_y)}{\sigma_x \sigma_y}$$

- Assumes a **linear relationship** between  $x$  and  $y$ .



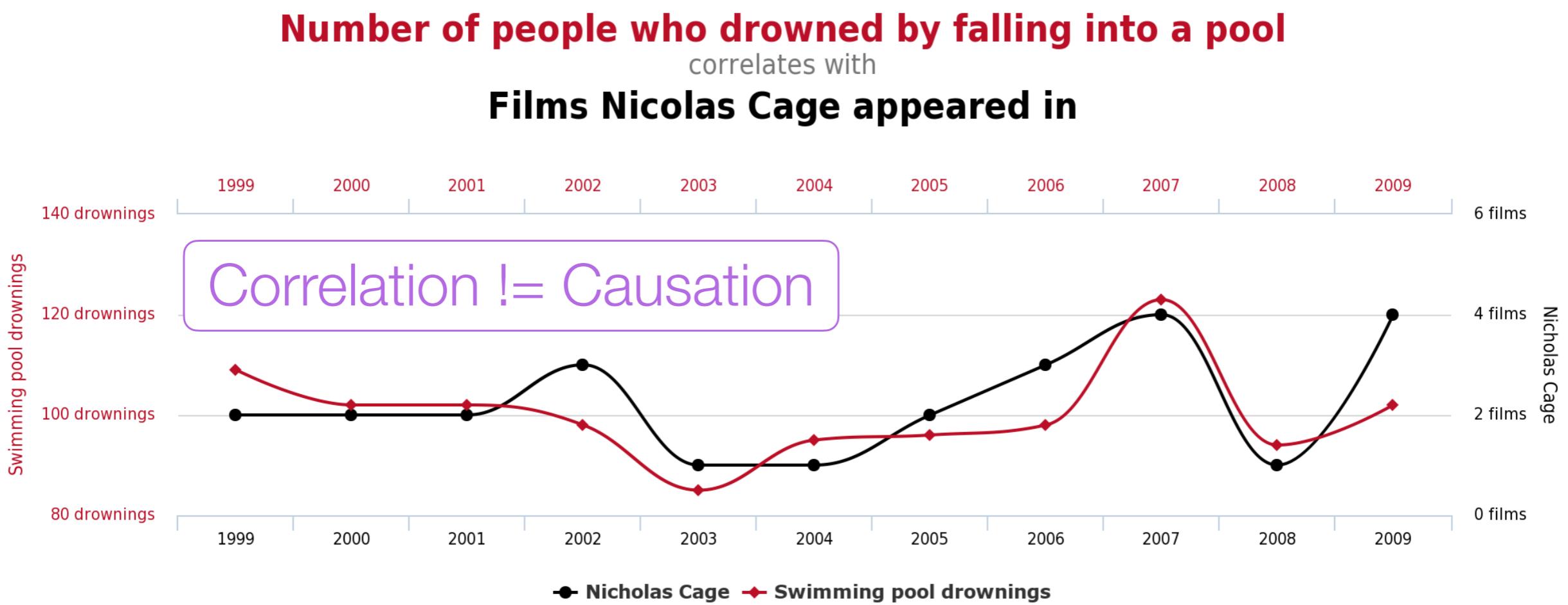
# Correlations of 2 time series

---

- The correlation of two time series gives you an indication of how similar their behavior is
- Two completely unrelated time series (say, two sequences of random numbers) will have a Pearson correlation coefficient of **0**

# Correlations of 2 time series

- The correlation of two time series gives you an indication of how similar their behavior is
- Two completely unrelated time series (say, two sequences of random numbers) will have a Pearson correlation coefficient of **0**



# Correlations of 2 time series

- The correlation of two time series gives you an indication of how similar their behavior is
- Two completely unrelated time series (say, two sequences of random numbers) will have a Pearson correlation coefficient of **0**
- Correlation != Causation!
- Adding a trend to both series we immediately observe a significant correlation

The Pearson correlation of two trending series is overwhelmed by the trend

# Auto-correlation

<https://en.wikipedia.org/wiki/Correlogram>

- It follows from the previous slide that a series will have a perfect correlation with itself, but what about lagged versions of itself?
- We define the Auto-correlation function as the Pearson correlation between values of the time series at different lags, as a function of the lag:

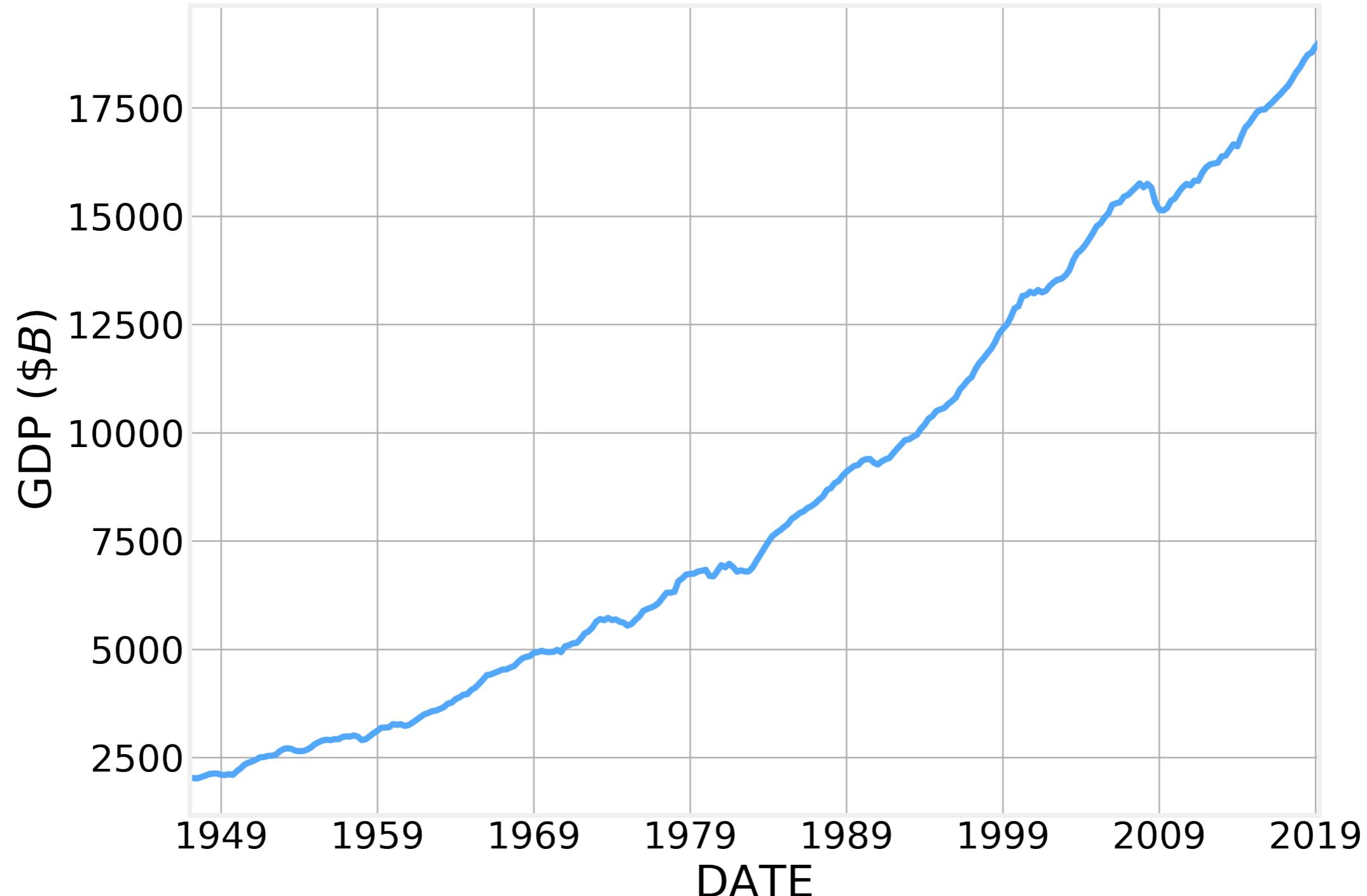
$$ACF_x(l) = \rho(x_t, x_{t-l})$$

- By definition,  $ACF_x(0) \equiv 1$
- And as the lag  $l$  increases the value of the  $ACF$  tends to decrease.
- We can calculate the confidence interval for the  $ACF$  using:

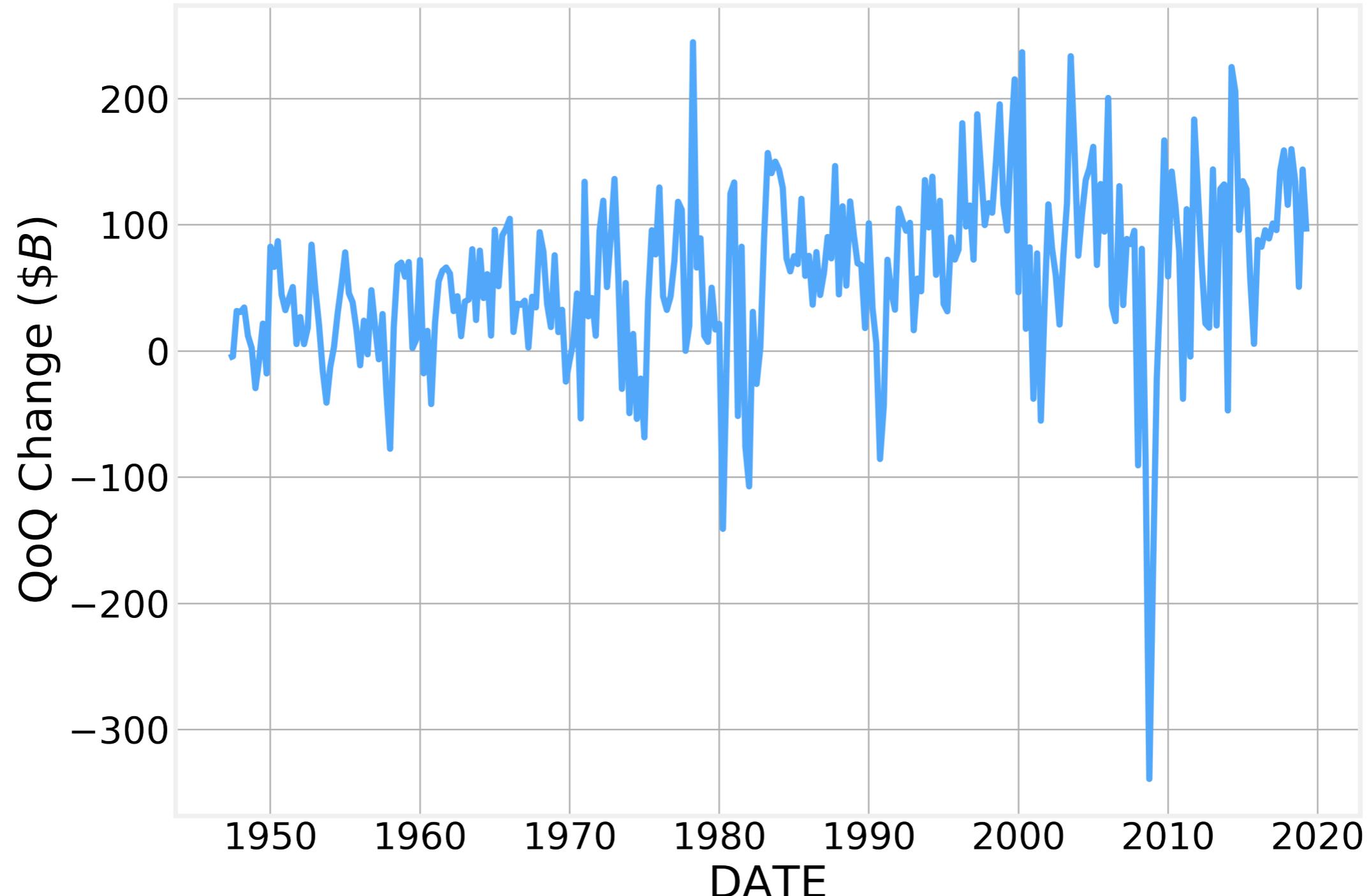
$$CI = \pm z_{1-\alpha/2} \sqrt{\frac{1}{N} \left( 1 + 2 \sum_{l=1}^k r_l^2 \right)}$$

- where  $z_{1-\alpha/2}$  is the quantile of the normal distribution corresponding to significance level  $\alpha$  and  $r_l$  are the values of the  $ACF$  for a specific lag  $l$

# Auto-correlation

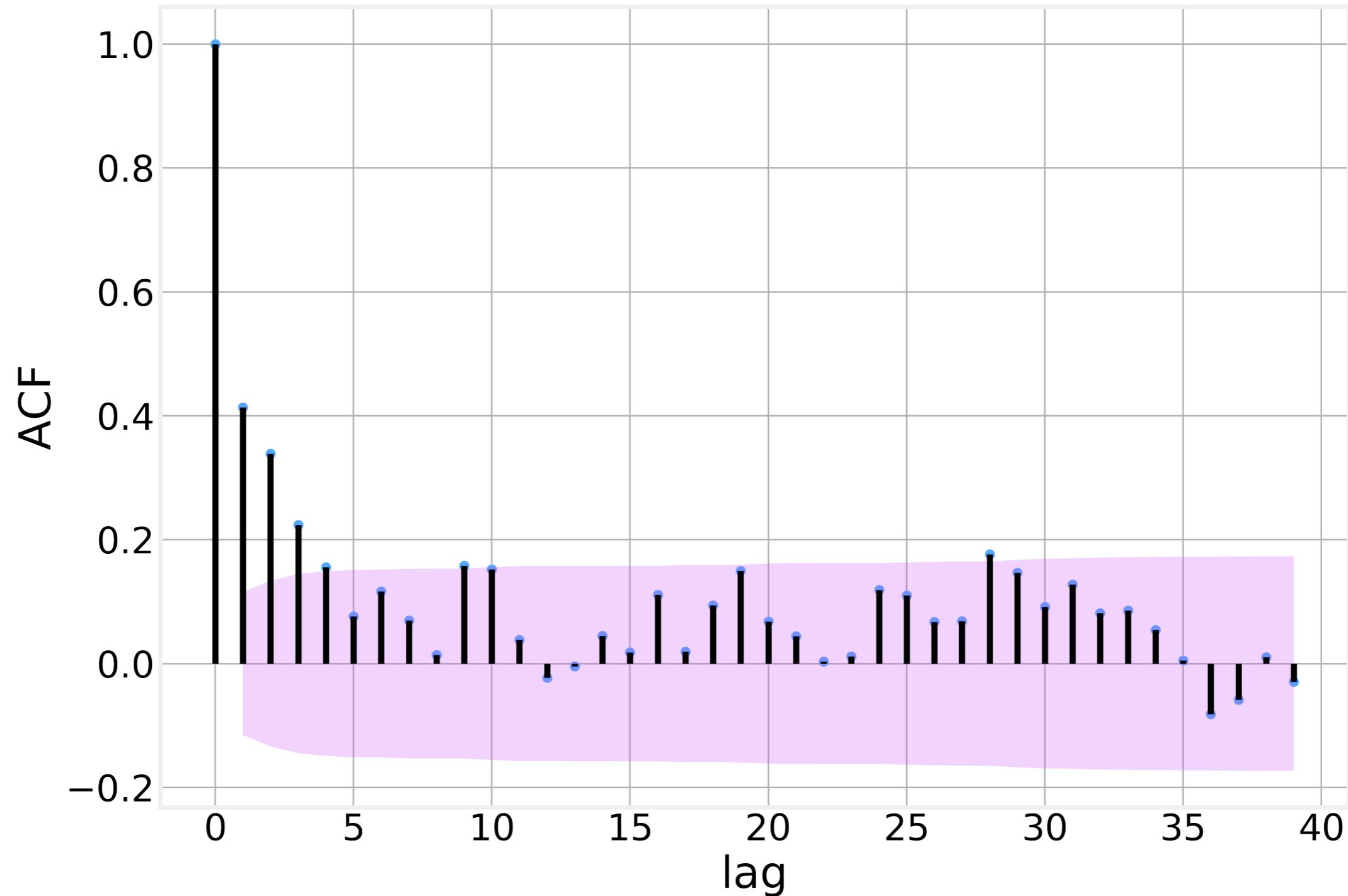


# Auto-correlation



# Auto-correlation

<https://en.wikipedia.org/wiki/Correlogram>



# Partial Autocorrelation

[https://en.wikipedia.org/wiki/Partial\\_autocorrelation\\_function](https://en.wikipedia.org/wiki/Partial_autocorrelation_function)

- One of the disadvantages of the Autocorrelation function is that it still considers the intermediate values
- The Partial Autocorrelation function calculate the correlation function between  $x_t$  and  $x_{t-l}$  after explaining away all the intermediate values  $x_{t-1} \cdots x_{t-l+1}$
- Intermediate values are "explained away" by fitting a linear model of the form:

$$\hat{x}_t = f(x_{t-1} \cdots x_{t-l+1})$$

$$\hat{x}_{t-l} = f(x_{t-1} \cdots x_{t-l+1})$$

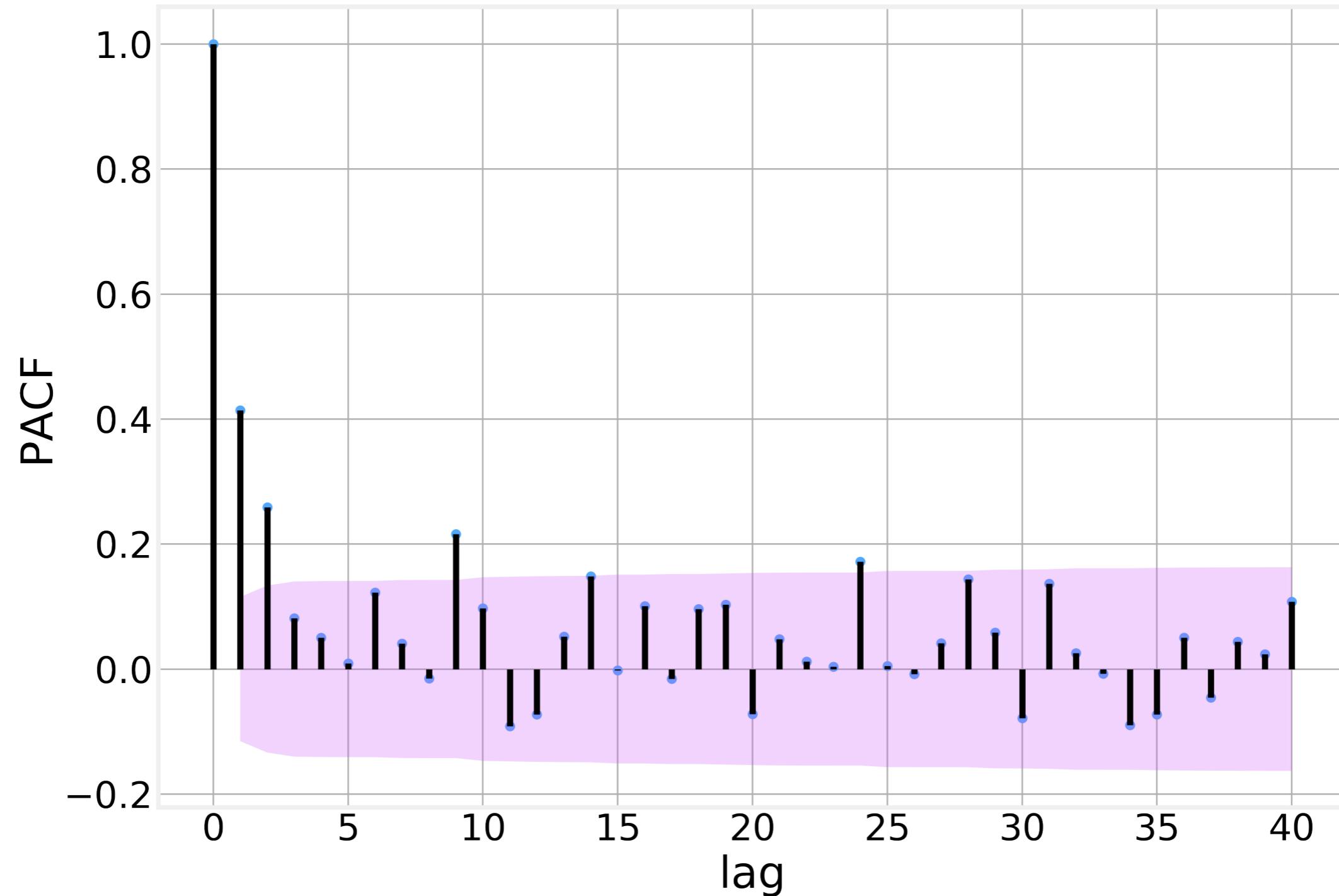
- And then calculating the Pearson correlation function between the values and their residuals:

$$PACF_x(l) = \rho(x_t - \hat{x}_t, x_{t-l} - \hat{x}_{t-l})$$

- Confidence intervals can be computed using the same formula used for the **ACF**

# Partial Autocorrelation

[https://en.wikipedia.org/wiki/Partial\\_autocorrelation\\_function](https://en.wikipedia.org/wiki/Partial_autocorrelation_function)





Code - Correlations

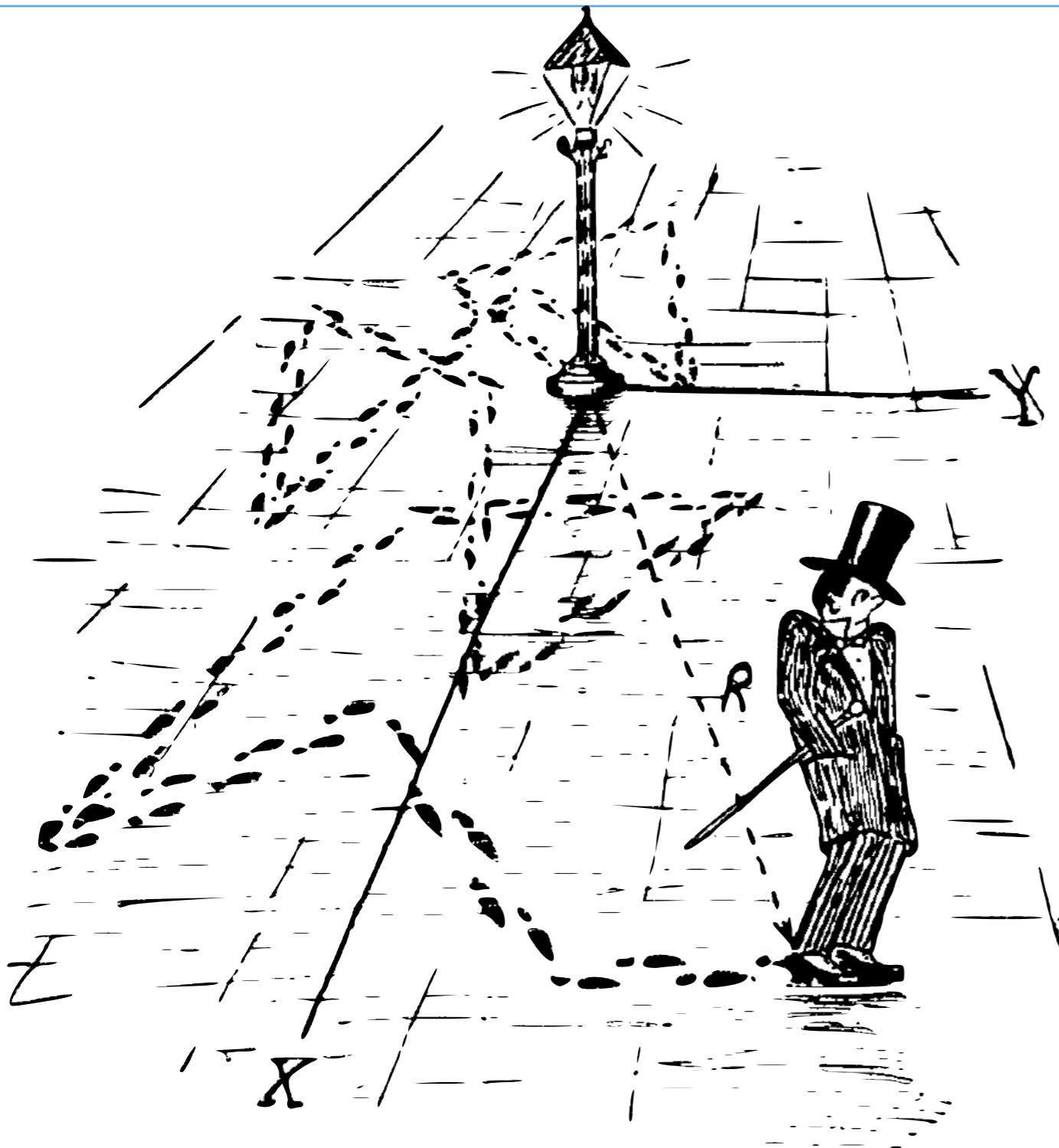
<https://github.com/DataForScience/Timeseries>



## Lesson V: Random Walks

# Random Walks

Illustration by George Gamow



- At each step flip a coin
  - Heads: Move right
  - Tails: Move left
- If you start at position **0**, do you ever reach position **L**?
- On average, we expect the position to be always close to **0**.
- What if the coin is biased as in the previous example?

# Random Walks

- Mathematically, we can describe the **position** of our random walker at time  $t$  as:

$$x_t = x_{t-1} + \epsilon_t$$

- Where  $\epsilon_t$  is the **stochastic value** generated by our coin flip ( $+1$  or  $-1$ )

- We can further write:

$$x_t = x_0 + \sum_i \epsilon_i$$

- which shows that the current position is just the **sum** across all **coin flips** in our walk.
- Naturally, we can treat a random walk as a realization of a time series, but **is it stationary?**
- The mean position is:

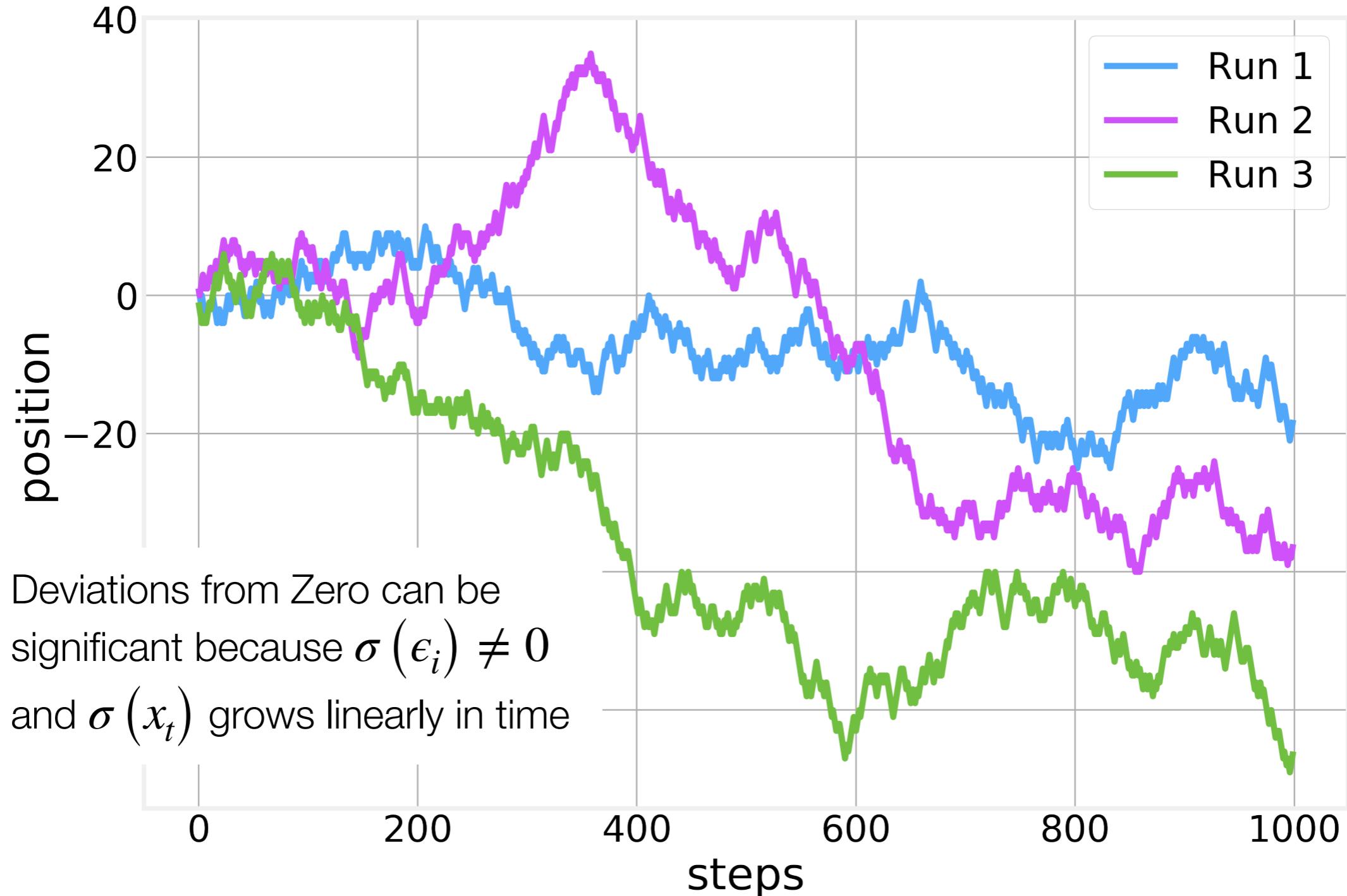
$$\mu = \langle x_t \rangle = \langle x_0 \rangle + \sum_i \langle \epsilon_i \rangle$$

- If the coin is unbiased,  $\langle \epsilon_i \rangle = 0$  and **the mean is constant**. On the other hand, the variance is:

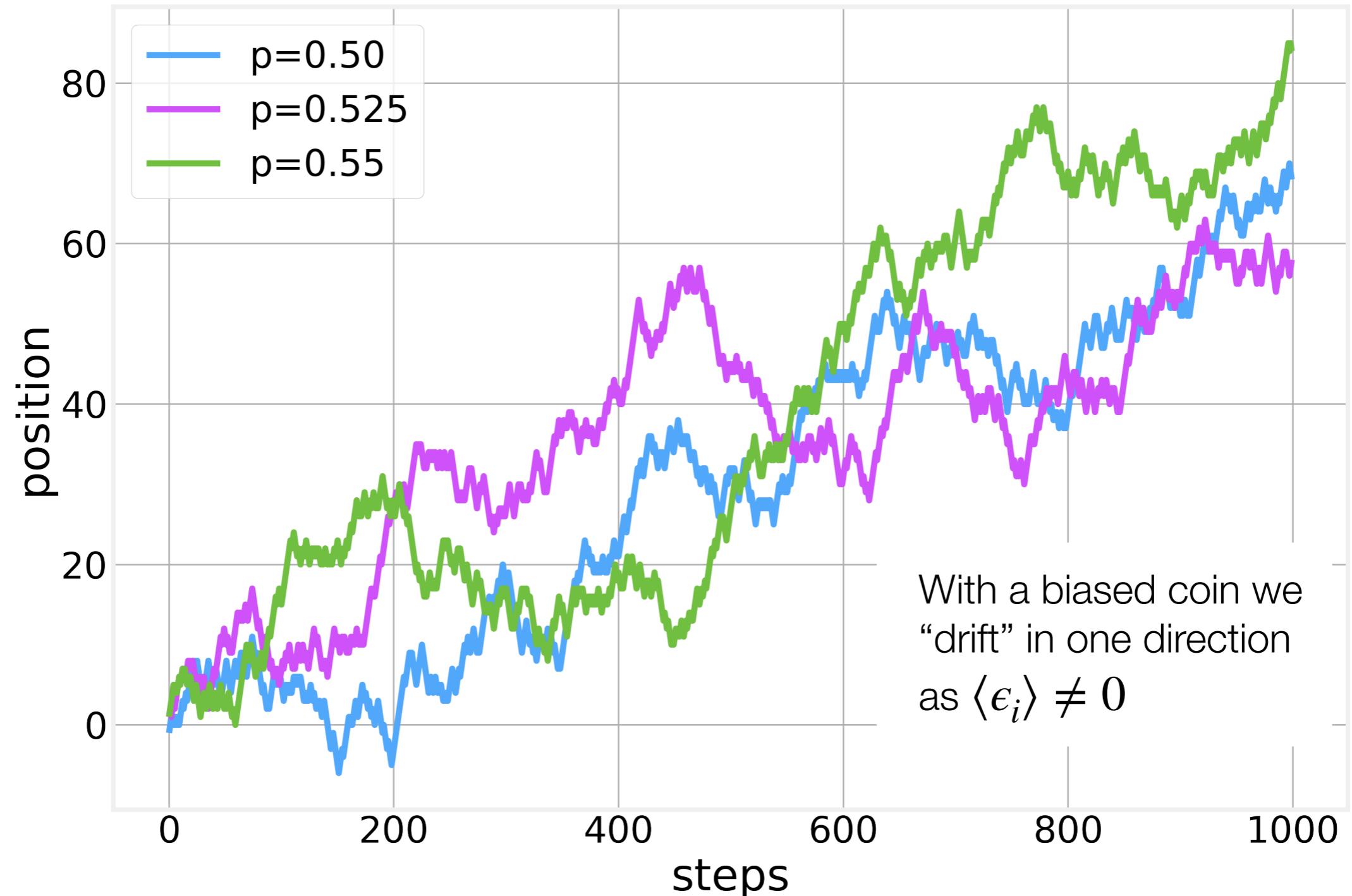
$$\sigma = \sigma(x_t) = \sigma(x_0) + \sum_i \sigma(\epsilon_i) = \sigma(x_0) + t \cdot \sigma(\epsilon)$$

- which **is not constant**. So even the simple random walk is **not a stationary process**.

# Random Walks

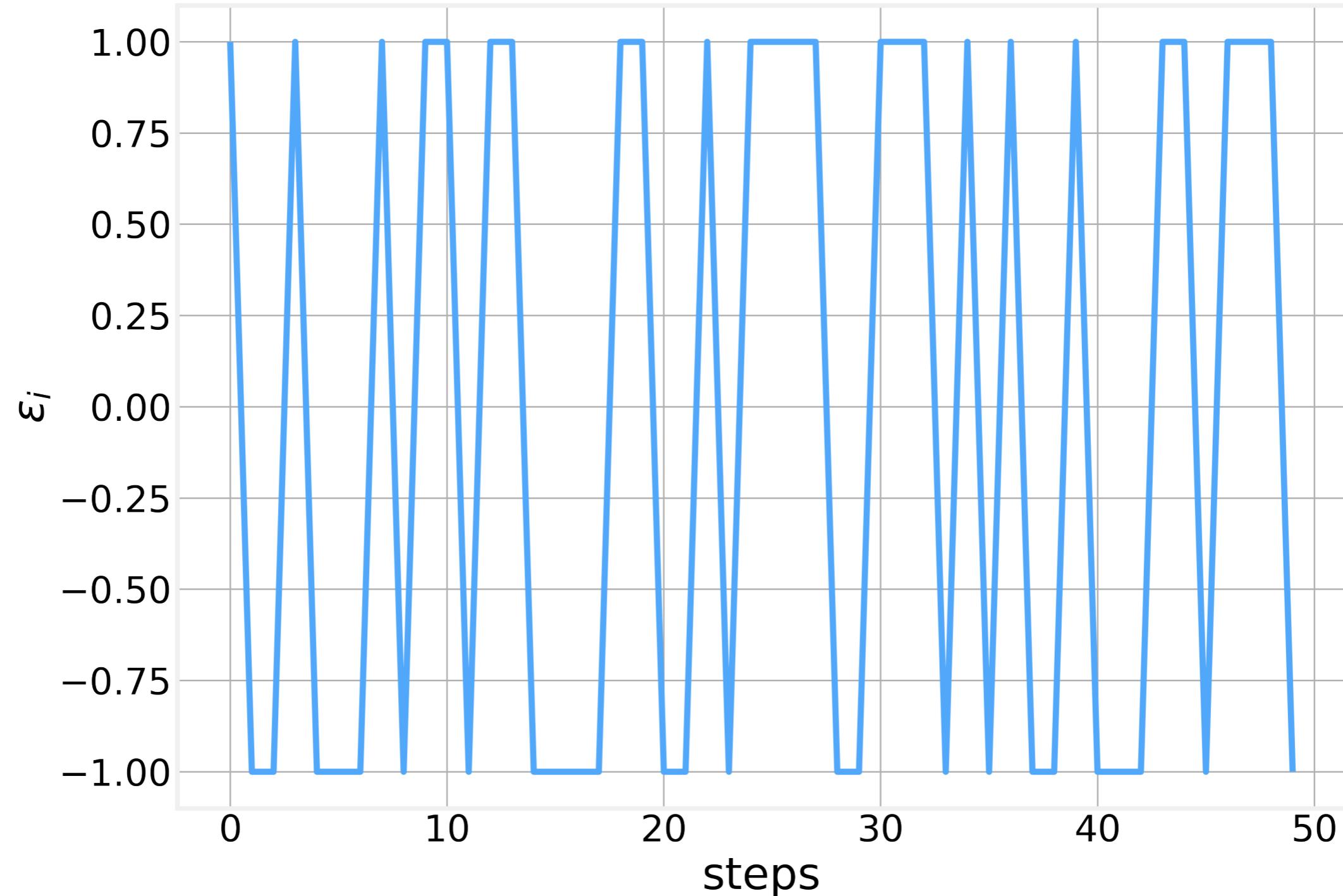


# Random Walk with a drift



# White Noise

Let's take a deep look at our stochastic variables (the outcomes of the “coin flips”)

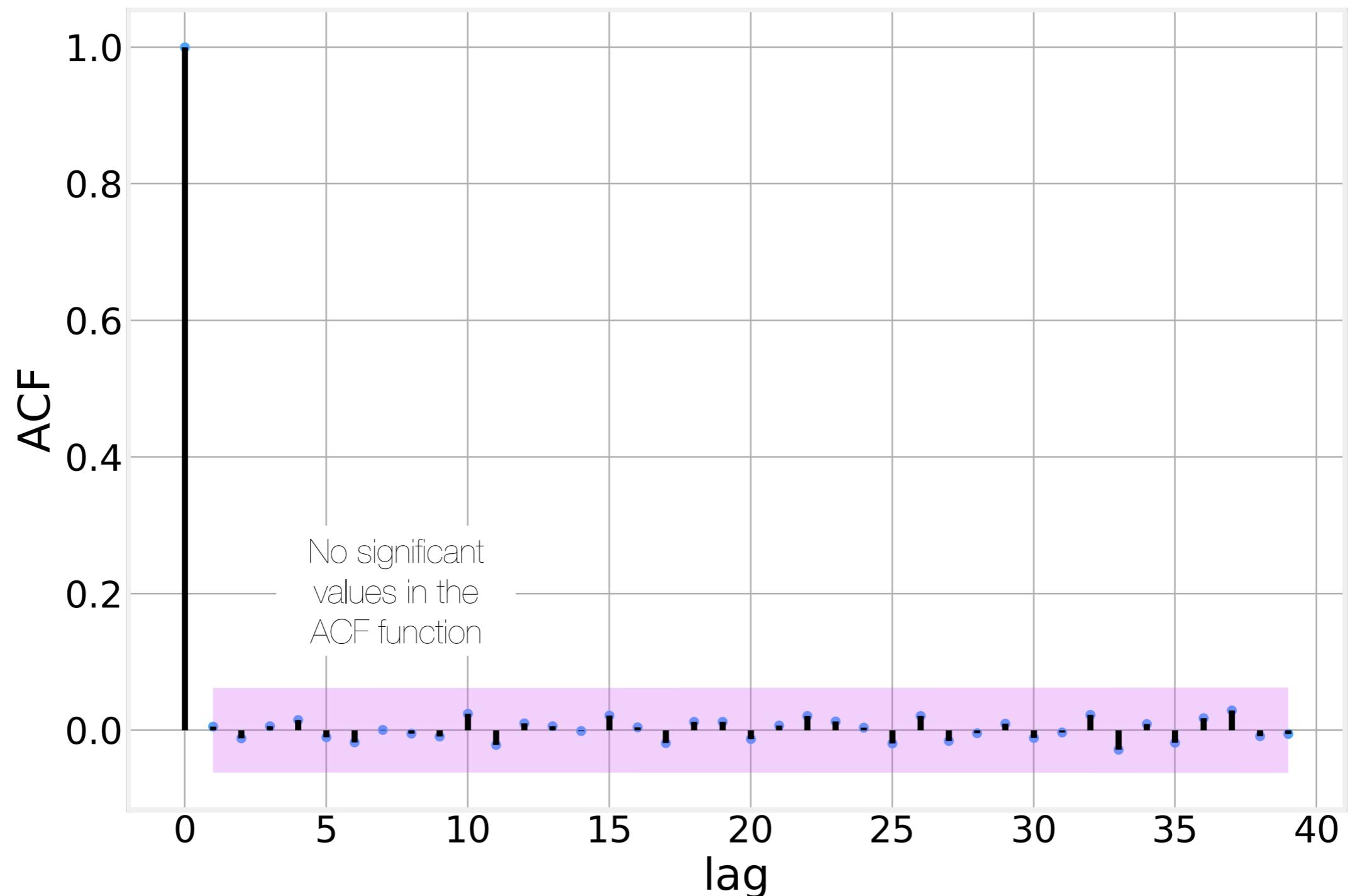


# White Noise

---

- White Noise is defined as:
  - a signal with **equal amplitudes at all frequencies**
  - a sequence of **uncorrelated random variables** (such as coin flips)
- Referred to as "noise" because it contains no information or **correlations**

# White Noise



# Stationary vs Non-Stationary

---

- We can also describe this process in the same mathematical framework as:

$$x_t = \epsilon_i$$

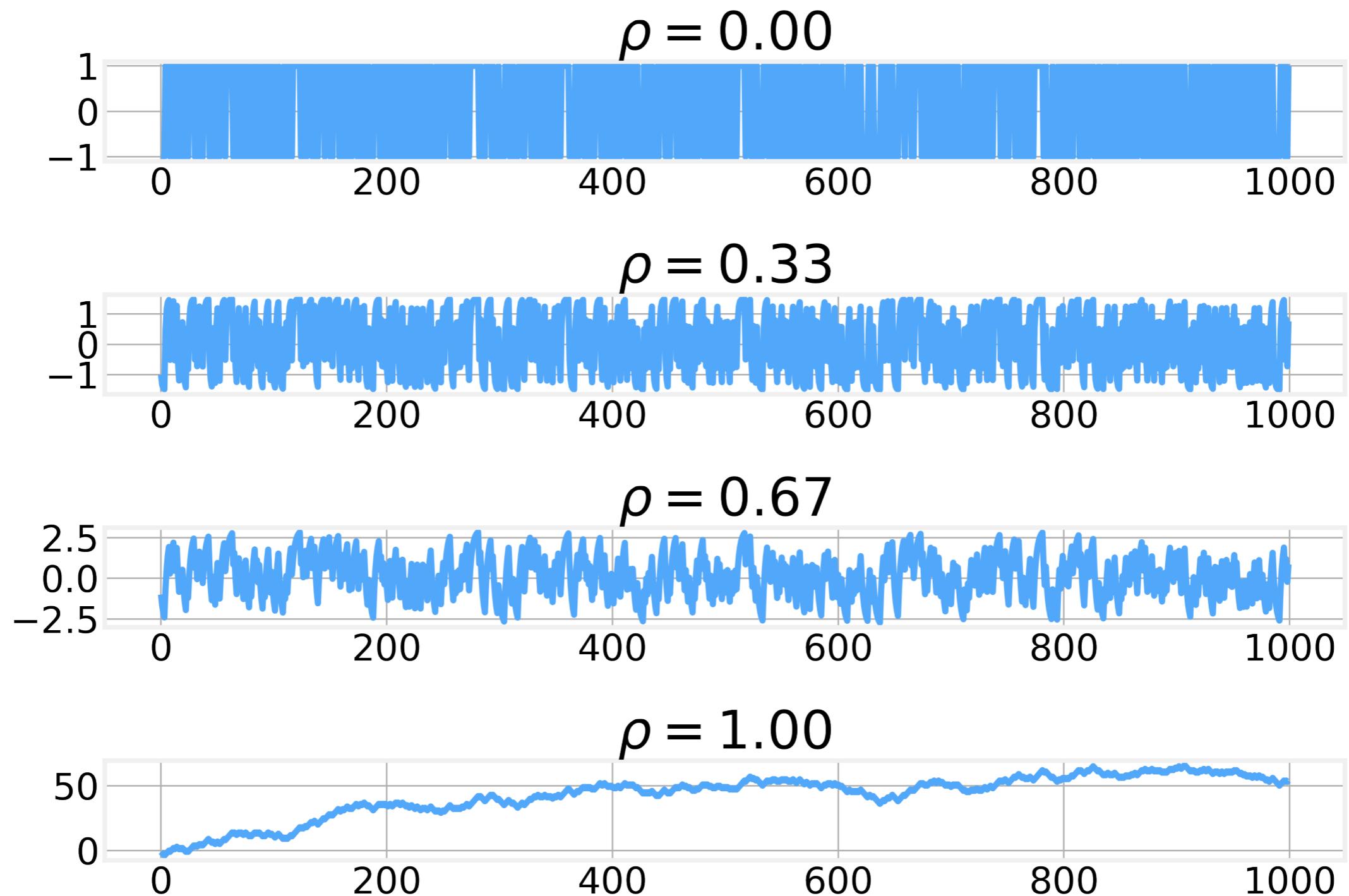
- which is clearly a **stationary process**.

- We can combine both expressions into a single one:

$$x_t = \rho x_{t-1} + \epsilon_i$$

- Where  $\rho$  gives us a “knob” to **interpolate** between the two extremes, **between a stationary and a non-stationary process**.

# Stationary vs Non-Stationary



# Dickey-Fuller Test

---

- The Dickey-Fuller Test is a test of stationarity inspired by a simple: Can we show that

$$\rho \neq 1$$

- with some degree of certainty?
- Numerically, we can express this as a linear regression fit

$$x_t - x_{t-1} = \gamma x_{t-1} + \epsilon_t$$

- where  $\gamma = \rho - 1$
- The slope of the regression is then our expected value for  $\rho - 1$ .
- If the process is non-stationary then we expect  $\gamma \neq 0$ .

# Dickey-Fuller Test

- From the residuals of  $\gamma$  we compute the **Dickey-Fuller statistic**:

$$DF = \frac{\hat{\gamma}}{SE(\gamma)}$$

- The value of this statistic is then compared with a critical values table.
- In general, the more negative it is, the more certain we can be that we can **reject the null hypothesis**
- The Dickey-Fuller Test has many variants.
- The most common one is the known as the **Augmented-Dickey-Fuller** test and is able to account for multiple lags, trends, etc.

Critical values for Dickey-Fuller t-distribution.				
	Without trend		With trend	
Sample size	1%	5%	1%	5%
T = 25	-3.75	-3.00	-4.38	-3.60
T = 50	-3.58	-2.93	-4.15	-3.50
T = 100	-3.51	-2.89	-4.04	-3.45
T = 250	-3.46	-2.88	-3.99	-3.43
T = 500	-3.44	-2.87	-3.98	-3.42
T = $\infty$	-3.43	-2.86	-3.96	-3.41

Source [2]:373

# Hurst Exponent

- The Hurst Exponent is another metric that allows us to determine whether or not a time series is stationary.
- It measures the “speed of diffusion” defined as:

$$Var(\tau) = \langle |z(t + \tau) - z(t)|^2 \rangle \sim \tau^{2H}$$

- where  $H$  is the **Hurst exponent**
  - $H < 0.5$  - mean reverting series
  - $H = 0.5$  - geometric random walk
  - $H > 0.5$  - trending series
- Smaller values indicate stronger levels of **mean reversion**, while larger values represent **stronger trends**



Code - Random Walks  
<https://github.com/DataForScience/Timeseries>



## Lesson VI: ARIMA Models

# Moving Average (MA) Model

- We start our exploration of the ARIMA family of models by considering the [Moving Average](#) model.
- The simplest moving average model can be written as:

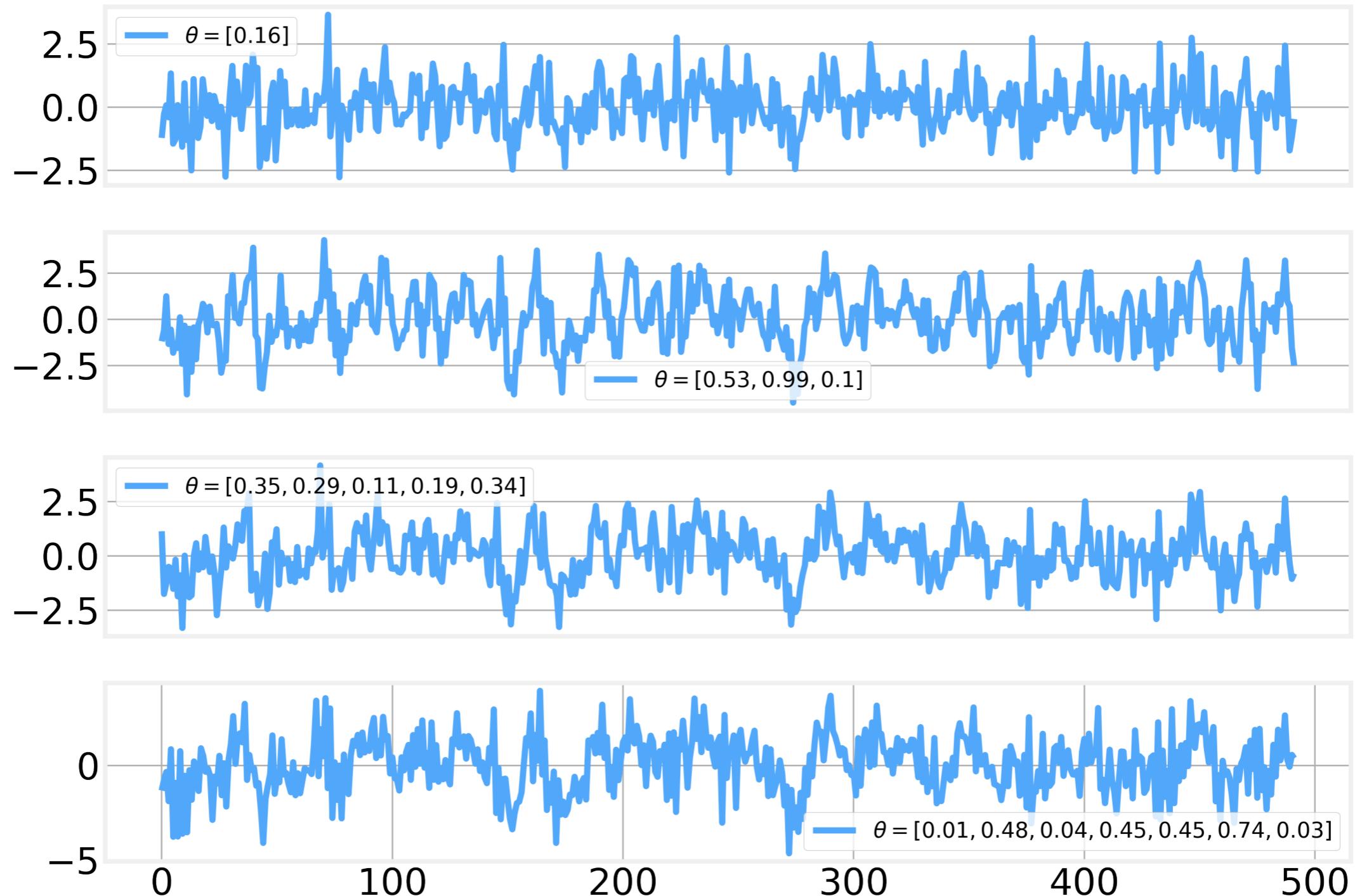
$$x_t = \epsilon_t$$

- Which we already saw in our discussion of random walks.
- A more general case can be written as:

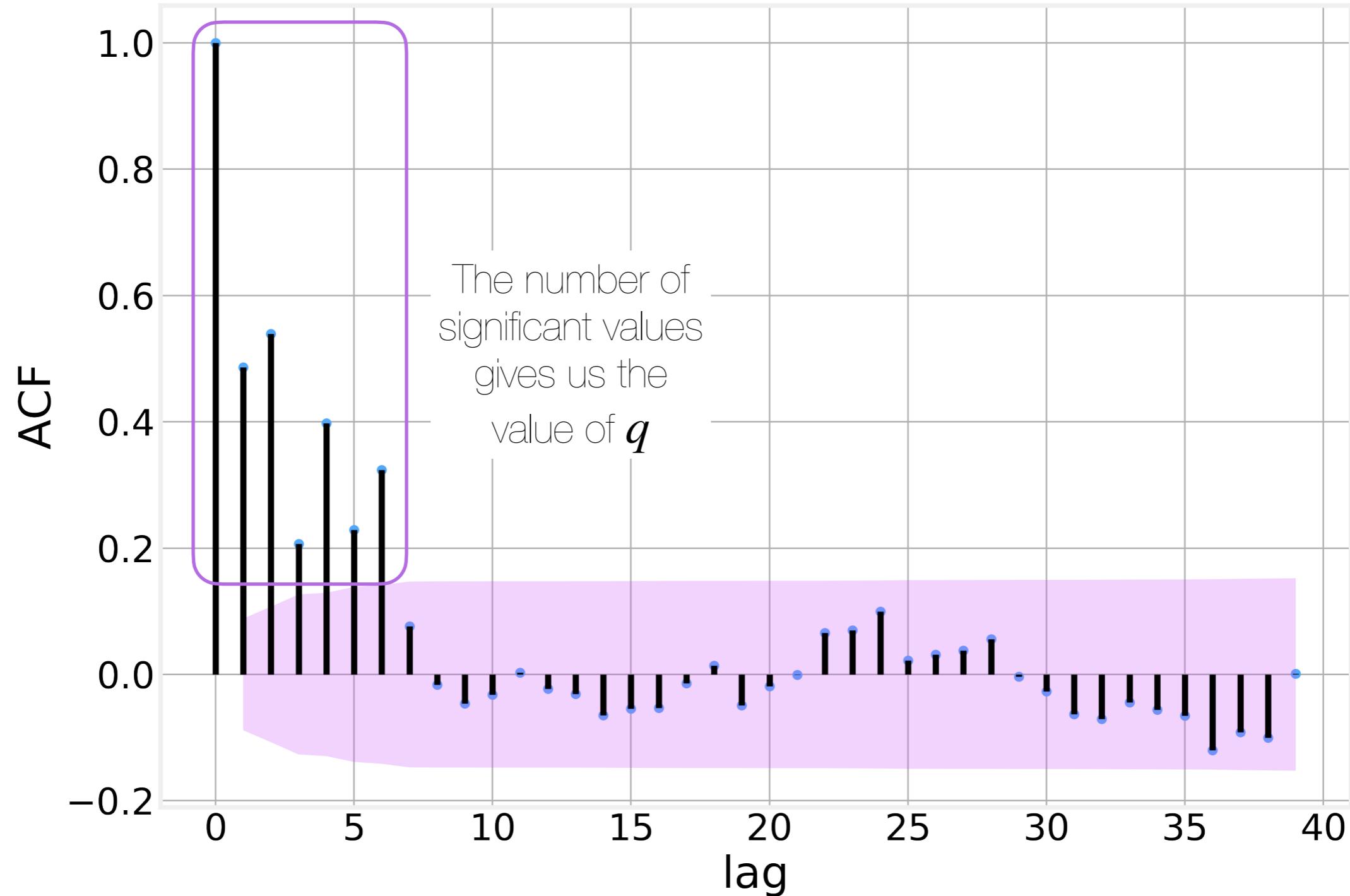
$$x_t = \beta + \sum_{l=0}^q \omega_l \epsilon_{t-l}$$

- where  $\beta$  is a constant offset,  $\omega_l$  are the weights for the values at lag  $l$  and  $q$  is the moving window size.  $\omega_0 \equiv 1$ .
- The  $\epsilon_t$  values are [stochastic variables](#) (often referred to as "errors") rather than the actual [observed values](#)  $x_t$
- The generated  $x_t$  is [uncorrelated](#) with itself for any lag  $l > q$

# Moving Average (MA) Model



# Moving Average (MA) Model



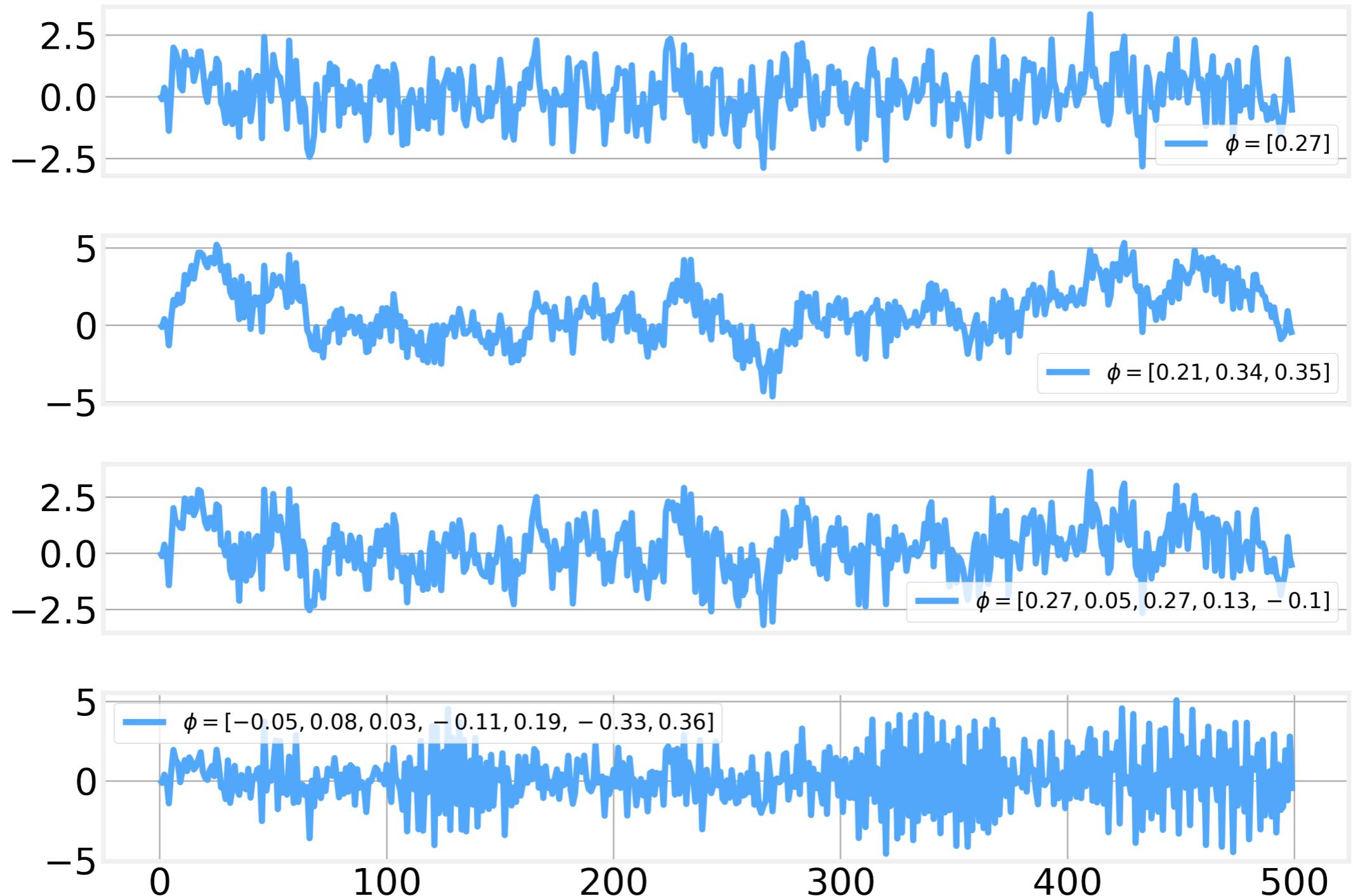
# Auto-Regressive (AR) Models

- Auto-Regressive models rely on the fact that in stationary models any deviation from the mean must be compensated. The series must “revert to the mean”.
- AR models can be defined as:

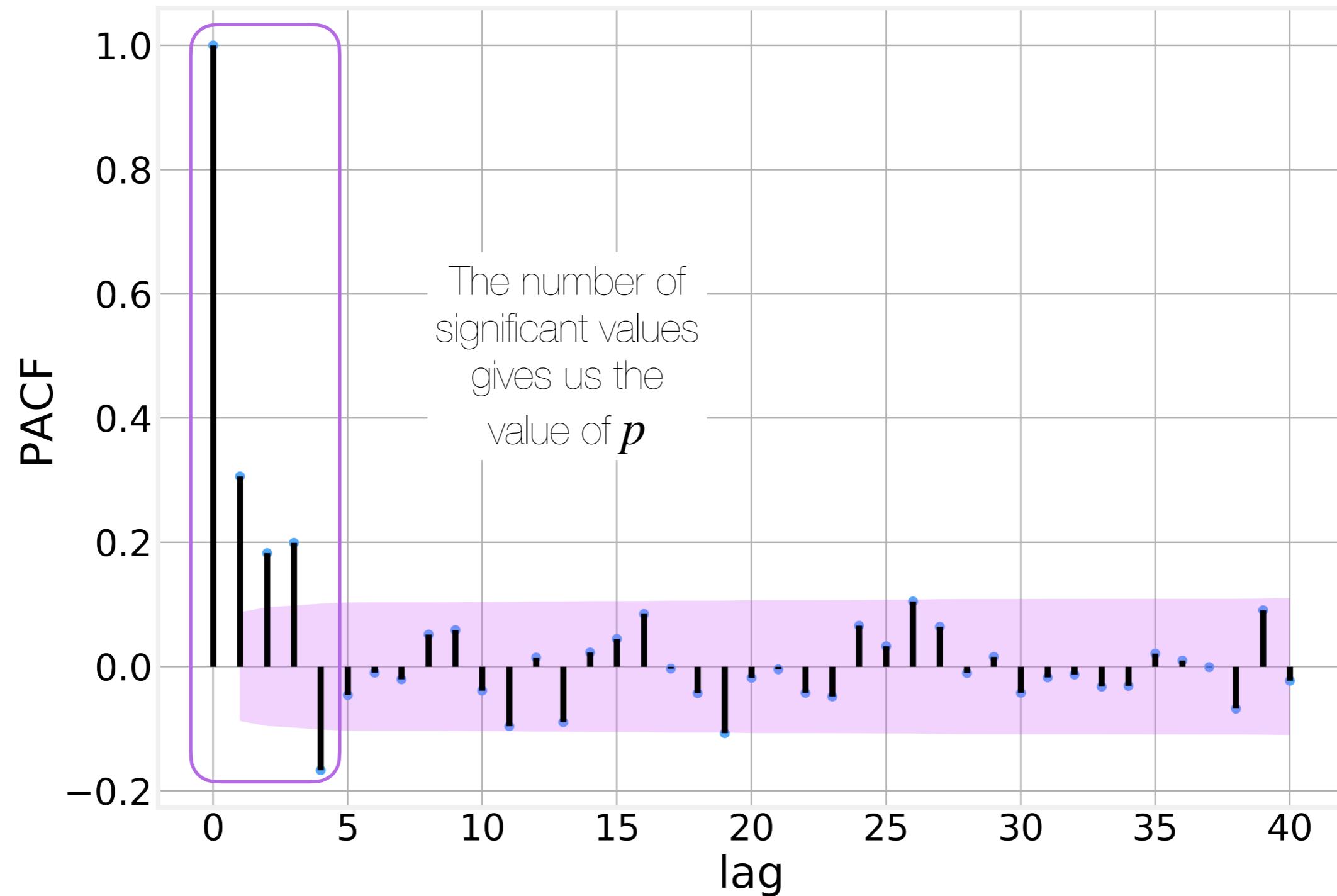
$$x_t = \alpha + \epsilon_t \sum_{l=1}^p \phi_l x_{t-l}$$

- Where the constant  $\alpha$  represents the process' average value and the  $x_{t-l}$  are the observed values at a given lag  $l$  and  $\phi_l$  are the corresponding weights.

# Auto-Regressive (AR) Models



# Auto-Regressive (AR) Models



## Integrative (I) "model"

---

- We already saw that we can take differences to "stationarize" the time series.
- To recover the original values, we must then integrate
- While not a model by itself, it is often an important first step in modeling time series

# ARIMA model

- The three classes of models we described above can be integrated into a single model:

A<sub>uto</sub>  
R<sub>egressive</sub>  
I<sub>ntegrated</sub>  
M<sub>oving</sub>  
A<sub>verage</sub>

$$x_t = c + \sum_i^p \phi_i x_{t-i} + \epsilon_t$$

$$x_t = \mu + \epsilon_t + \sum_i^q \theta_i \epsilon_i$$

- The complete model can be written as:

$$\hat{x}_t = c + \mu + \sum_i^p \phi_i x_{t-i} + \sum_j^q \theta_i \epsilon_{t-i} + \epsilon_t$$

- where  $\hat{x}_t$  is the properly differentiated time series

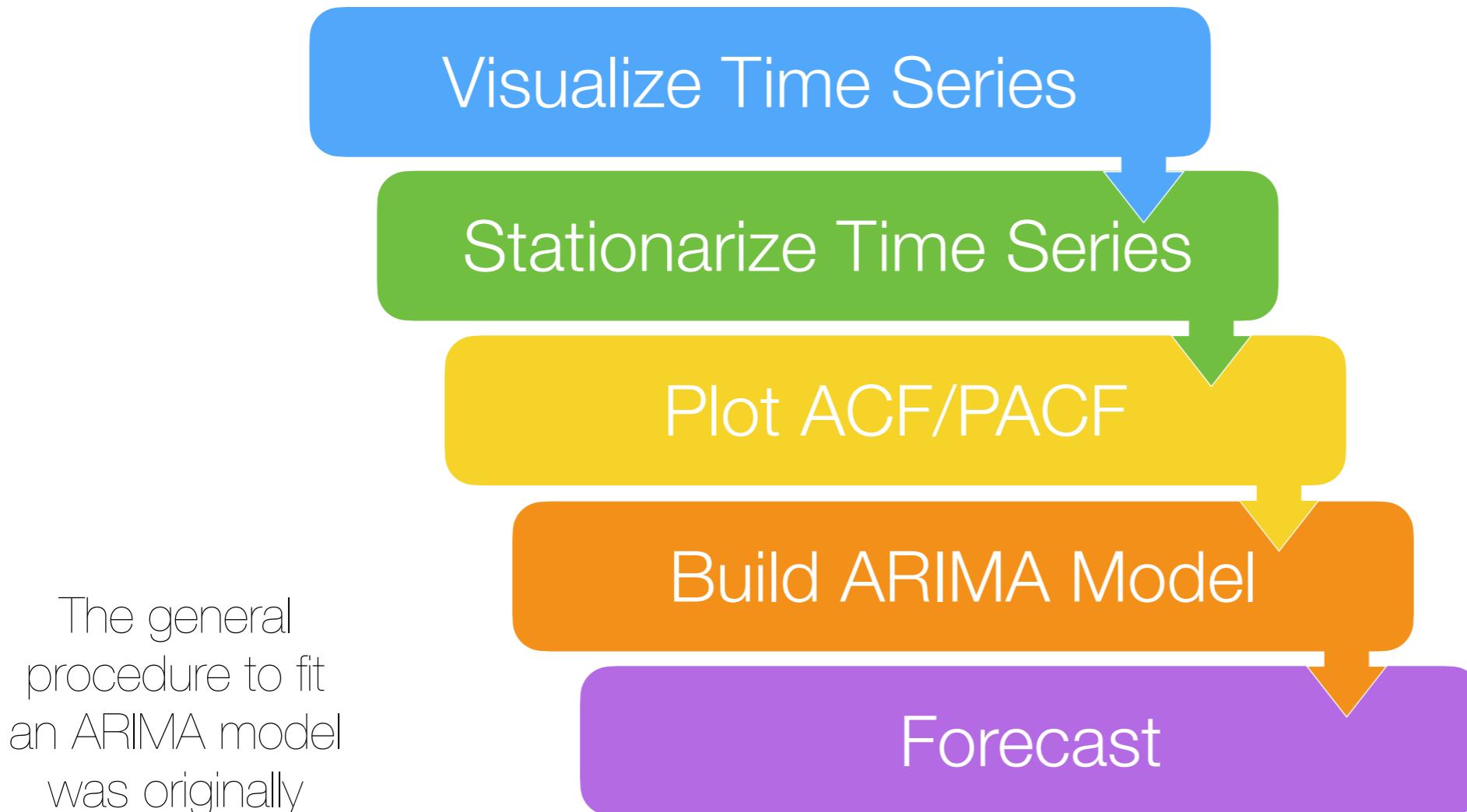
# ARIMA model

---

- From this simple definition we can easily recover several interesting special cases:
- **$ARIMA(0,1,0)$**  - Random Walk (with or without drift)
  - $x_t - x_{t-1} = c + \epsilon_t$
- **$ARIMA(0,0,0)$**  - White noise (the sequence of stochastic variables)
  - $x_t = \epsilon_t$
- **$ARIMA(0,1,1)$**  - Exponential Smoothing
  - $x_t - x_{t-1} = \epsilon_t + \theta_1 \epsilon_{t-1}$
- **$ARIMA(0,2,2)$**  - Double exponential Smoothing
  - $x_t - 2x_{t-1} + x_{t-2} = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2}$

# Fitting ARIMA models

<https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/>



- $d$  - degree of differencing
- $p$  - number of lag observations included in the model (**PACF**)
- $q$  - size of the moving average window (**ACF**)



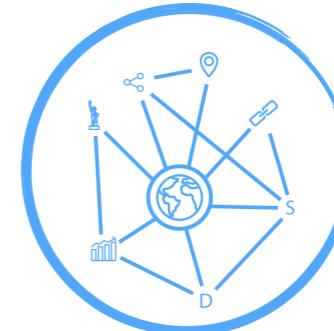
Code - ARIMA

<https://github.com/DataForScience/Timeseries>

# Events



<http://paypal.me/data4sci>



[graphs4sci.substack.com](https://graphs4sci.substack.com)

## Advanced Time Series for Everyone

May 20, 2021 - 5am-9am (PST)

## Natural Language Processing for Everyone

Jun 9, 2021 - 5am-9am (PST)

## Advanced NLP for Everyone

Jun 25, 2021 - 5am-9am (PST)

## Natural Language Processing (NLP) from Scratch

<http://bit.ly/LiveLessonNLP> - On Demand

