

Boring but important disclaimers:

- ▶ If you are not getting this from the GitHub repository or the associated Canvas page (e.g. CourseHero, Chegg etc.), you are probably getting the substandard version of these slides Don't pay money for those, because you can get the most updated version for free at

<https://github.com/julianmak/academic-notes>

The repository principally contains the compiled products rather than the source for size reasons.

- ▶ Associated Python code (as Jupyter notebooks mostly) will be held on the same repository. The source data however might be big, so I am going to be naughty and possibly just refer you to where you might get the data if that is the case (e.g. JRA-55 data). I know I should make properly reproducible binders etc., but I didn't...
- ▶ I do not claim the compiled products and/or code are completely mistake free (e.g. I know I don't write Pythonic code). Use the material however you like, but use it at your own risk.
- ▶ As said on the repository, I have tried to honestly use content that is self made, open source or explicitly open for fair use, and citations should be there. If however you are the copyright holder and you want the material taken down, please flag up the issue accordingly and I will happily try and swap out the relevant material.

OCES 3301 : basic Data Analysis in ocean sciences

Session 7: fairly basic time-series analysis

Outline

(Just overview here; for actual content see Jupyter notebooks)

- ▶ time-series data
 - recall: frequency, wavelength, wavenumbers etc.
- ▶ basic manipulations
 - filtering (high and low pass) and kernels
 - trends (just regression)
 - lag analysis (also just regression)
 - power spectrum (PSD)

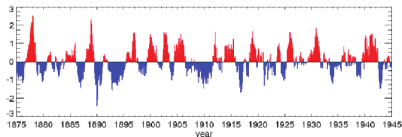
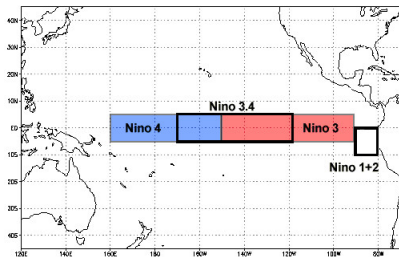
Time-series data



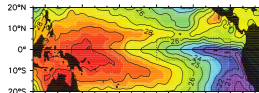
Figure: The eternal bendy boi.

Time-series data

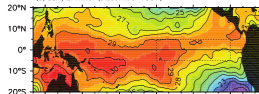
- recall El-Niño data (lec 02)
 - SST data over time



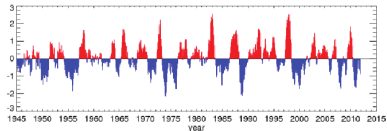
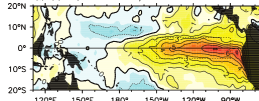
(a) SST, 'Normal' (December 1996)



(b) SST, El Niño (December 1997)



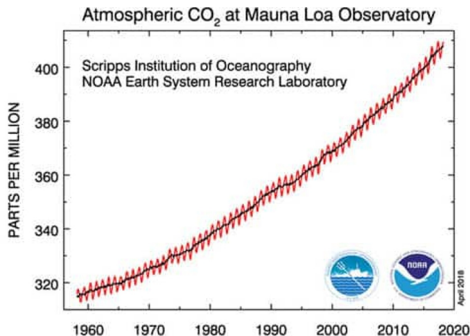
(b) - (a) SST, El Niño anomaly



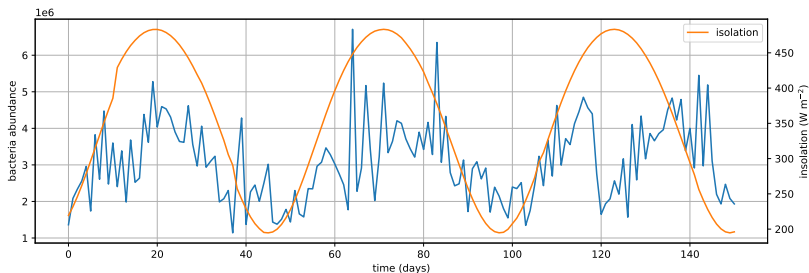
- detrended averaged signal here

Time-series data

- Keeling curve
→ atmospheric CO₂ concentration at one spot
- rolling average
given as black line
→ non-linear trend?



Time-series data



► bacteria data from Charmaine (see assignment 2)

→ some oscillation **period/frequency**?

→ some **correlation**?

Idealised example

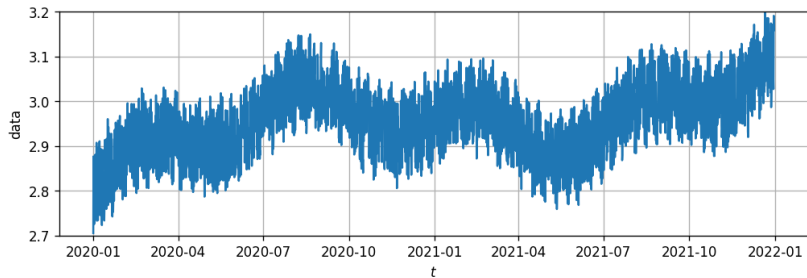


Figure: A dense signal.

Idealised example

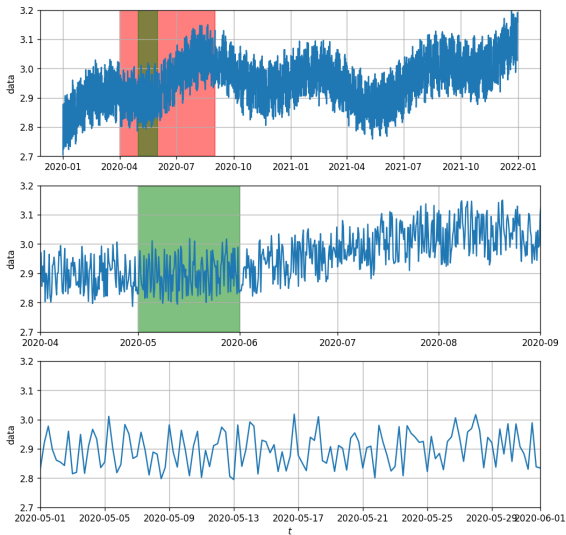


Figure: A dense signal: zoomed in.

Idealised example: filtering

- ▶ noise/fluctuations coexisting with 'real' signal (cf. 05, 06)
 - how to pick these out?
 - average over some window?

Idealised example: filtering

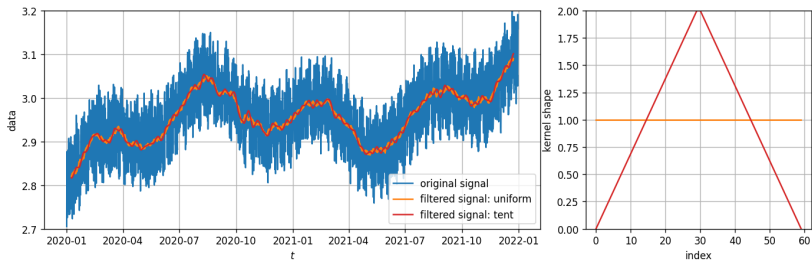


Figure: A dense signal: moving window average.

- uniform window (moving boxcar)
- tent
- others

Idealised example: convolution with kernel

- ▶ **low-pass filtering** is when you filter out the high frequencies (so leaving the low frequencies intact) by some averaging etc.
- ▶ formally down through a **convolution**

$$f^{<}(t) = (f * G)(t) = \int f(\tau)G(t - \tau) \, d\tau,$$

→ G is the **kernel**

→ averaging \sim integral (sums)

→ boxcar is taking $G \equiv 1/T$ in some interval of length T , and 0 outside of interval

- ▶ $f = f^{>} + f^{<}$, the **high-passed** and **low-passed** signal respectively

→ low pass filter kills the high frequencies, and vice-versa

Convolutions

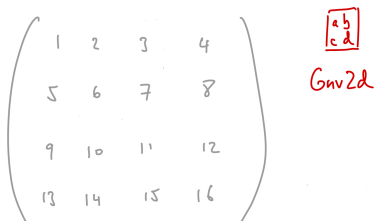


Figure: Sample array to be convolved with a kernel (the red stuff). Kernel here is of size (2,2).

Convolutions

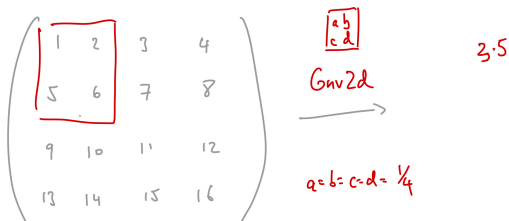


Figure: Choose a kernel, throw down kernel, element-wise multiplication, then sum.

Convolutions

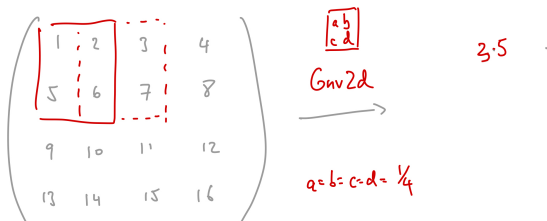


Figure: Move the kernel with some **stride** (1 here), then continue same process.

Convolutions

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix} \xrightarrow[\substack{\text{Conv2d} \\ a=b=c=d=\frac{1}{4}}]{\begin{matrix} a & b \\ c & d \end{matrix}} \begin{pmatrix} 3.5 & . & 5.5 \\ . & . & . \\ . & . & . \end{pmatrix}$$

Figure: Repeat until whole array is done. By default convolution leads to an array with **reduced** the size.

Convolutions

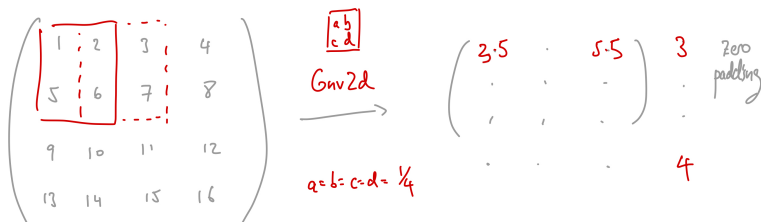


Figure: Can do **padding** for various reasons. Zero padding here bulks the **input** array size.

Idealised example: filtering

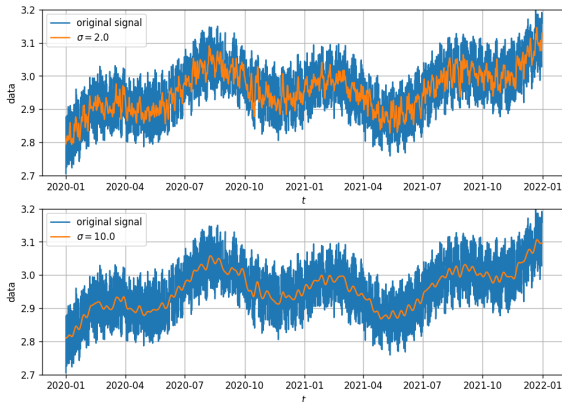


Figure: A dense signal: Gaussian filter.

- Gaussian kernel here (cf. lec05; see two slides)
 - often used in image processing (e.g. 'blurring')

Idealised example: trends

- normally **linear trends**
 - just linear regression...
 - preserved by filtering?

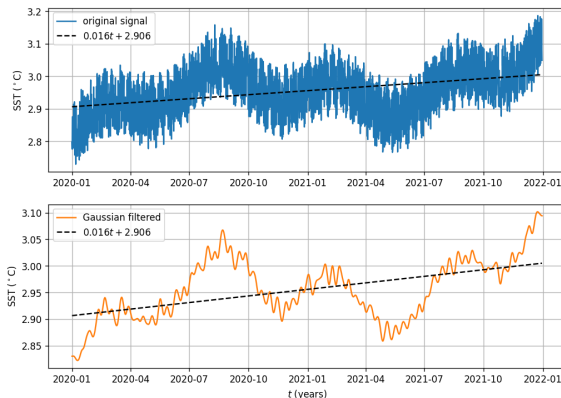


Figure: A dense signal: linear trend.

Idealised example: trends

► **detrend** to get **anomalies** with respect to the linear trend

→ come back to this in 08

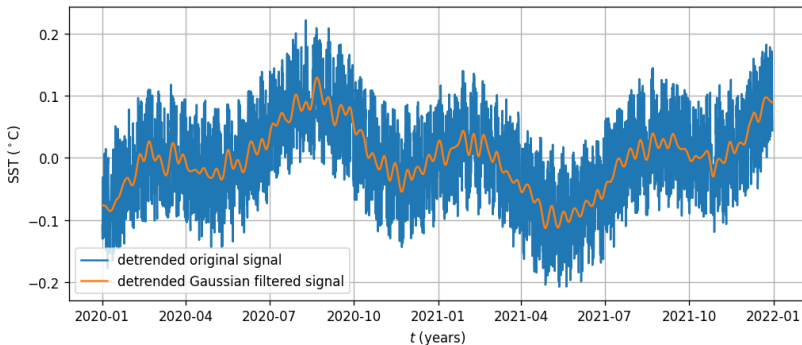


Figure: A dense signal: linear trend.

Idealised example: correlations

- ▶ again, just linear correlations
- compare two data at the same time

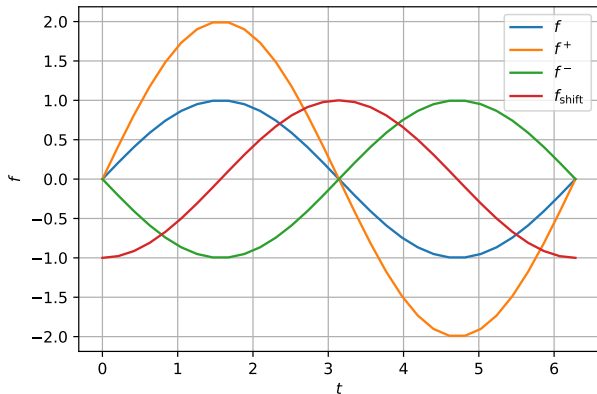


Figure: What is the expected linear correlation here (with respect to f)?

Idealised example: correlations

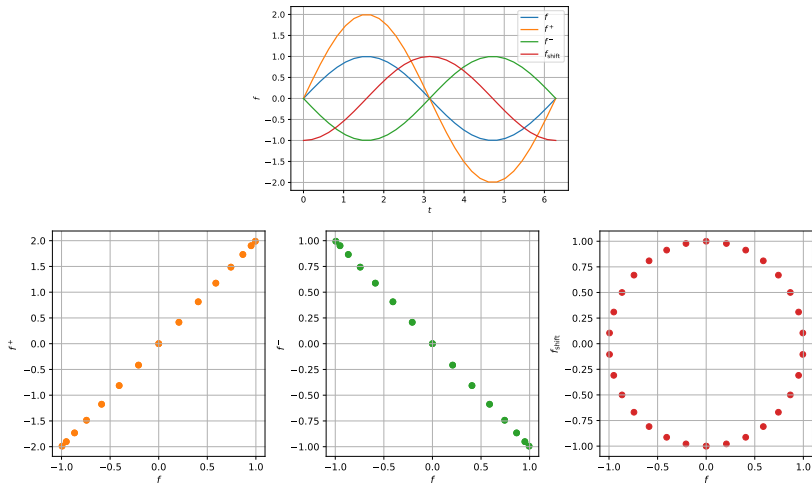


Figure: As above figure but shown in a slightly different way, might be easier to see...

Idealised example: lag correlations

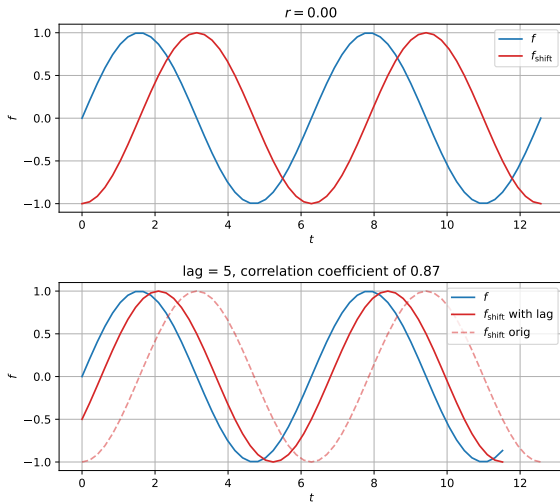


Figure: Correlation of original and shifted signals.

Idealised example: lag correlations

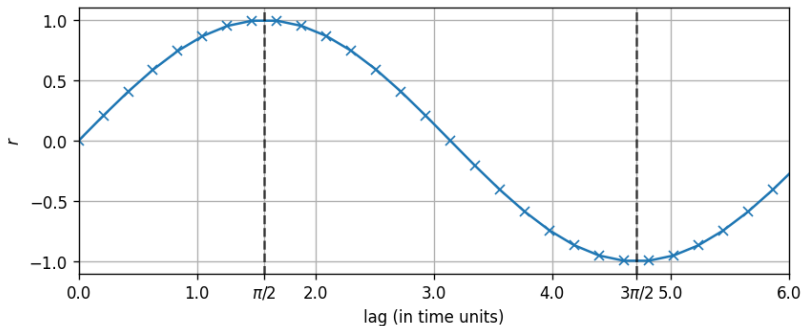


Figure: Correlation of above signal as a function of lag (are max and min values consistent with expectations?)

Idealised example: auto-correlations

- ▶ **auto** means 'self'
 - correlation of lagged versions of itself
- ▶ one interpretation of how 'predictable' something is
 - the `statsmodel` version does something slightly different

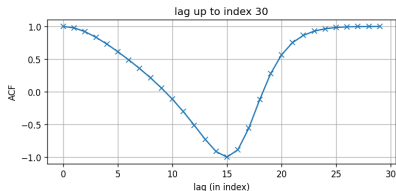
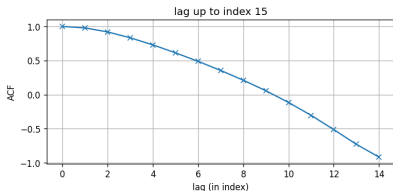


Figure: Auto-correlation of a simple signal (why is there an apparent asymmetry?)

Power spectrum

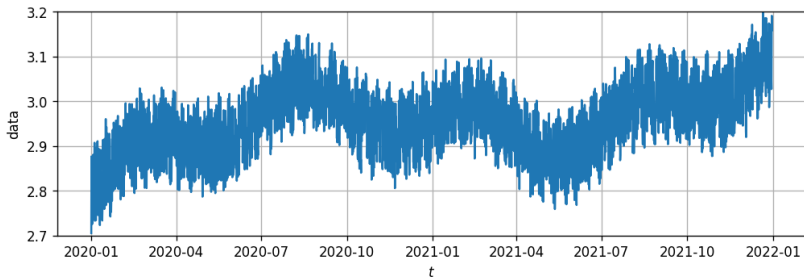


Figure: A dense signal.

- ▶ interested in quantifying the oscillations in the data
→ period? amplitude?
- ▶ do this with a **power spectrum**
→ the maths of it a bit complicated (not really, but problems every year...)
→ focus here on the interpretation

Power spectrum (PSD = Power Spectrum Density)

- ▶ magic command is `signal.periodogram`
→ for syntax and subtleties, see notebook

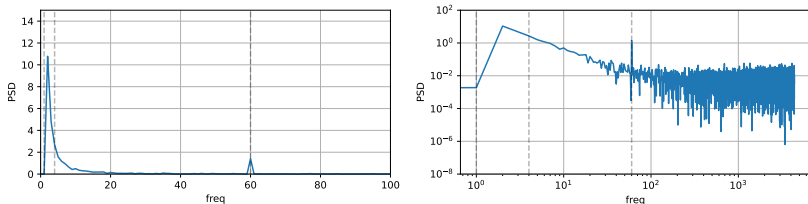


Figure: PSD of the idealised signal.

- ▶ a peak of sorts at 60 here
→ indicating a signal oscillating at 60 units
→ deliberately vague about the units here

Q. vertical lines indicate other oscillations I did put in, but no peaks in PSD?

Power spectrum (PSD = Power Spectrum Density)

- ▶ tidying up and using sensible units gives the following

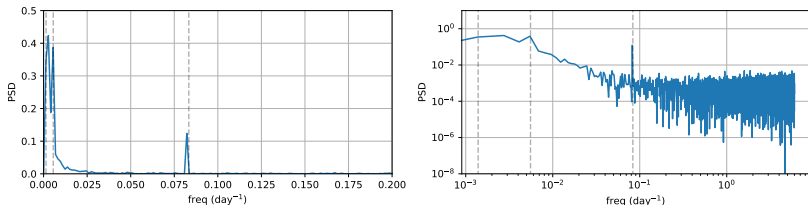


Figure: PSD of the idealised signal with 'right' units and sampling rate.

- ▶ a peak at around 0.08, and $1/0.08 = 12.5$ days
→ I put one it at 12 days
- ▶ other peaks at values corresponding to 180 and 720 days
→ some issues with detrending required

Power spectrum: El-Niño 3.4 data

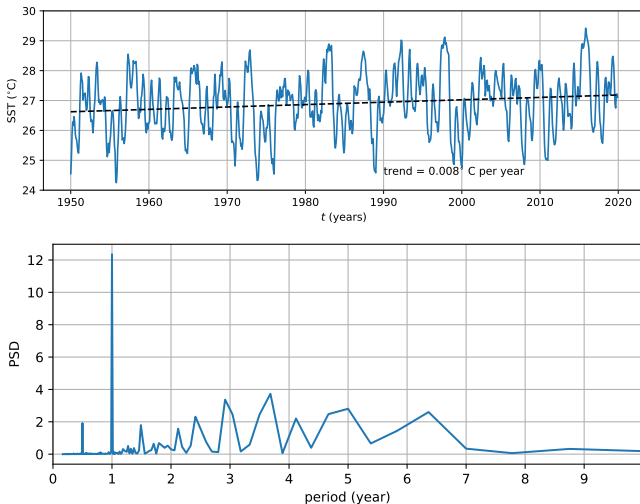


Figure: Application to ENSO SST data (took one over the frequency to get period). PSD computed from detrended data.

Jupyter notebook

go to 07 Jupyter notebook to get some code practise

- ▶ other things with El-Niño 3.4 or tide data
- ▶ could also try it with the bacteria data in assignment 2
→ just comment out my categorisation step

(Look up [Fourier series/transforms](#) if you want more details about the PSD)