# MATLAB

Class 3: Signal processing

# Simulating Data

**Randi()** – returns random number of max i

**Randn()** – returns number from standard normal dist

**Rand()** –returns random number between 0-1

**Sin()** –can be used to simulate oscillatory data

# Data Smoothing

You can try and remove noise from your data by smoothing

**Smooth() and smoothdata()**

**Detrend()**

# Normalizing data

Sometimes it is necessary to normalize your data before you preform an analysis

This could mean scaling the range of the data or mean centering it

**Normalize()**

# Normalizing data

It is important to normalize data specially if you are comparing measures that do not have a defined range (i.e., a personality scale vs neural activity)

**Normalize data if you are comparing two measures of different scales (e.g., in PCA, PLS etc)**

# Data Interpolation

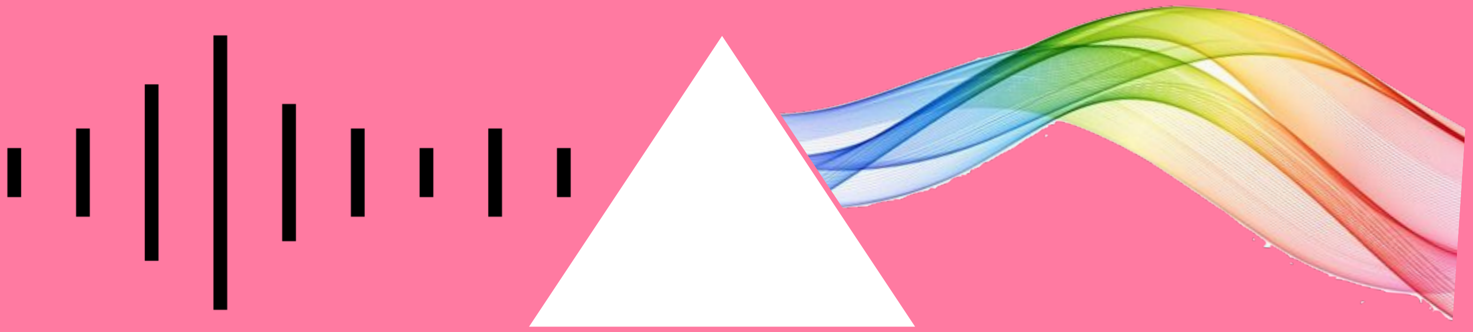
If there are large outliers in your dataset, you may consider interpolation (i.e., removing outliers and guessing what the value should have been)

**Filloutliers()** is a useful function to identify and replace outliers in your data

# Spectral analysis

Attempts to break down a wave or signal into its underlying frequency components

Answers the question how much of each frequency is there in my data

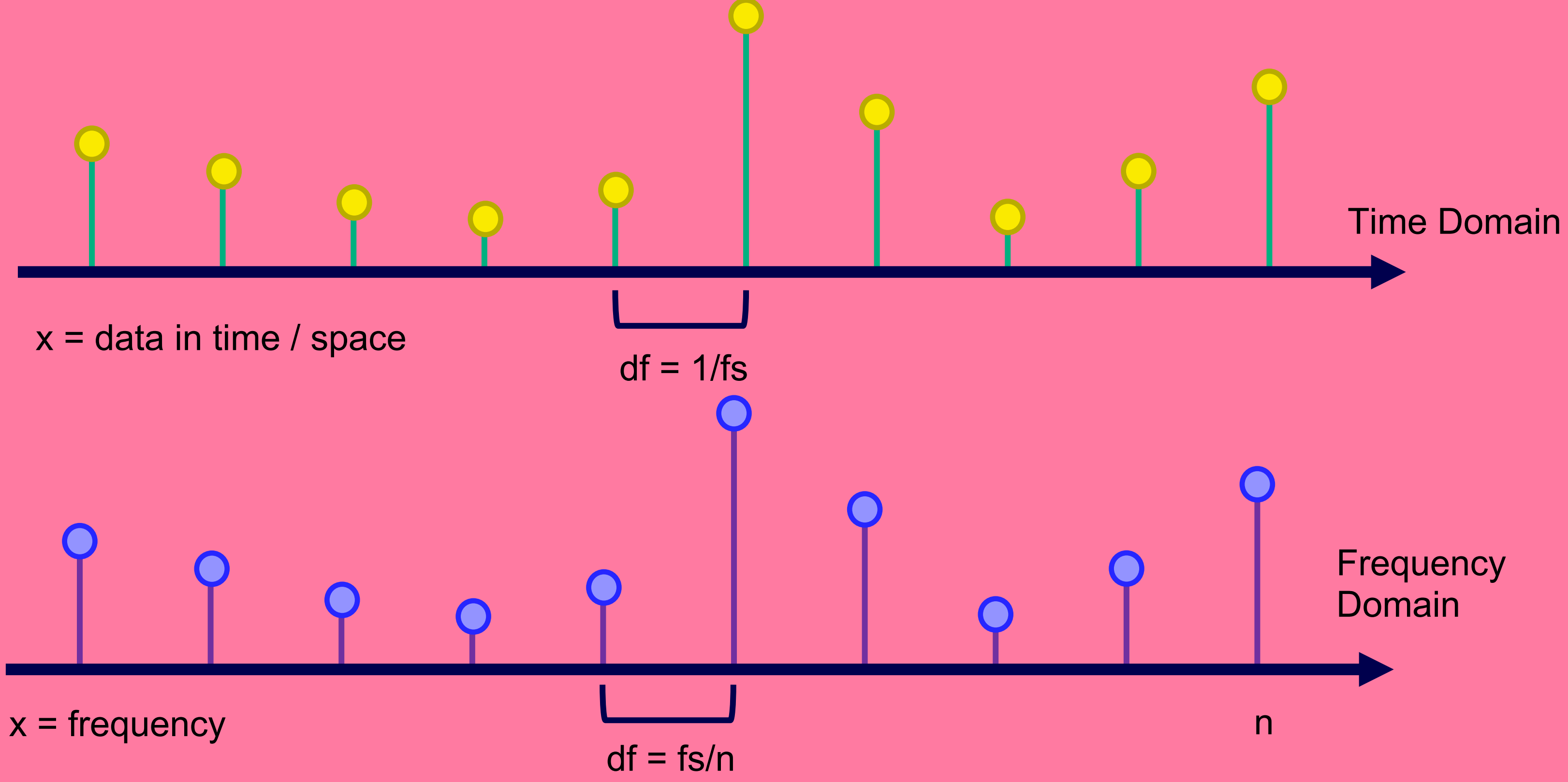


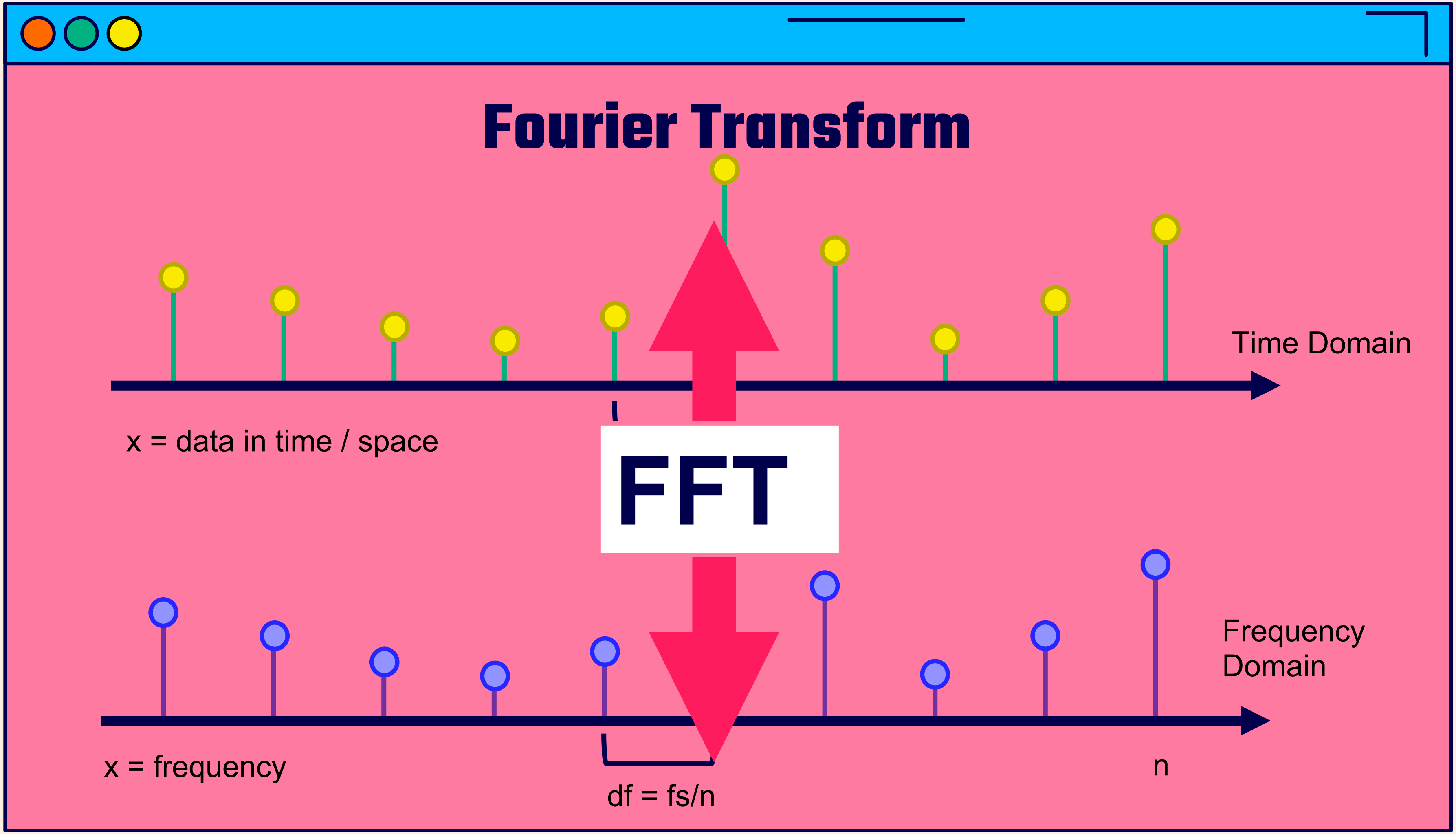
# Fourier Transform

The Fast Fourier Transform is one way we can mathematically express and solve this problem.

It returns the **spectrum** of **frequencies** that make up a signal

# Fourier Transform





Fourier Transform
Time Domain
x = data in time / space
FFT
Frequency
Domain
x = frequency
df = fs/n
n

# Fourier Transform

In MATLAB, the function **fft()** returns the spectral estimate of a timeseries

There are many methods of spectral decomposition, the **Welch** method is one often used to increase the SNR of your output.

Here your signal is broken down into smaller windows, an FFT of these windows is computed and then averaged

# What is a Filter (not Instagram)

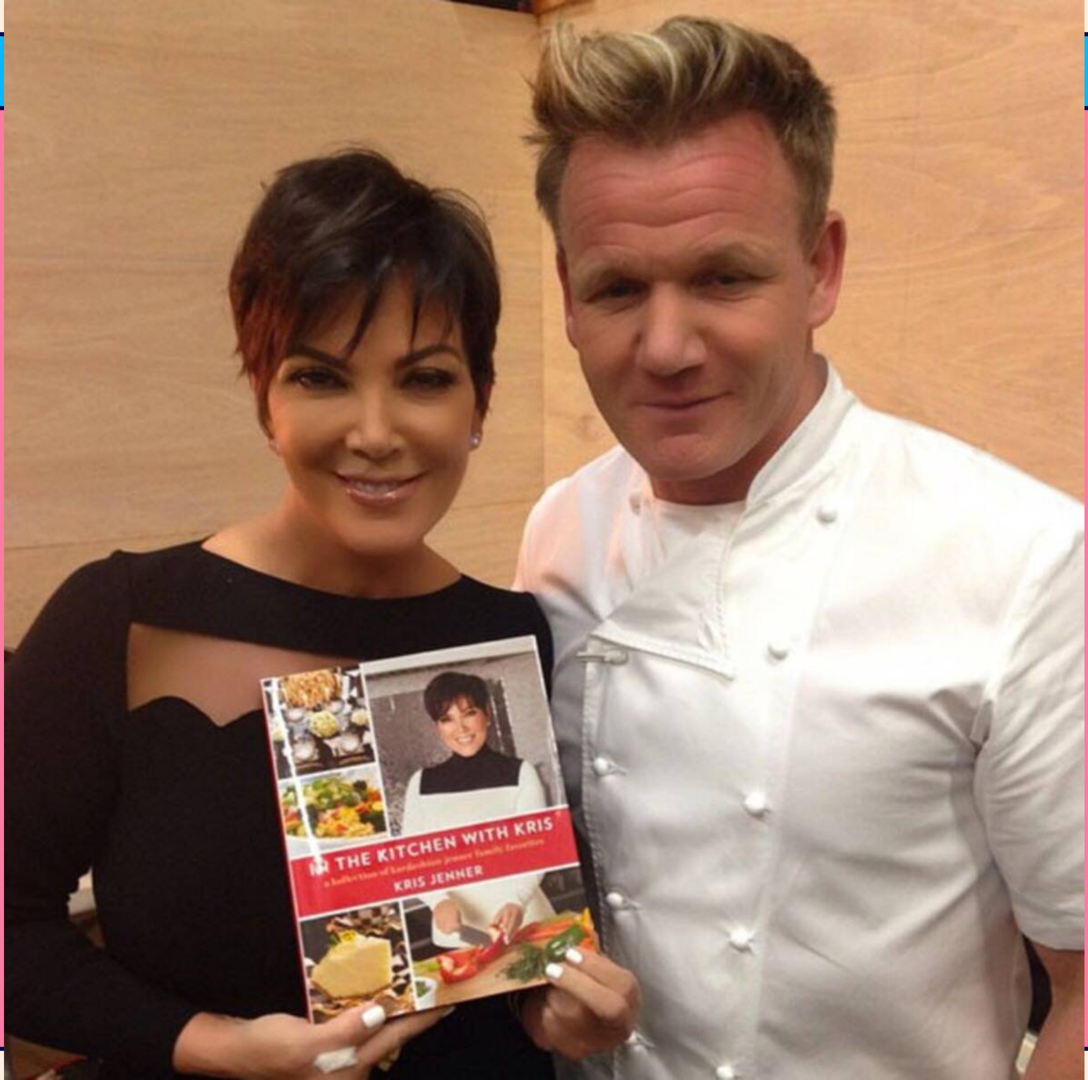
A collection of methods to remove unwanted signals/artifacts/aspects of your data

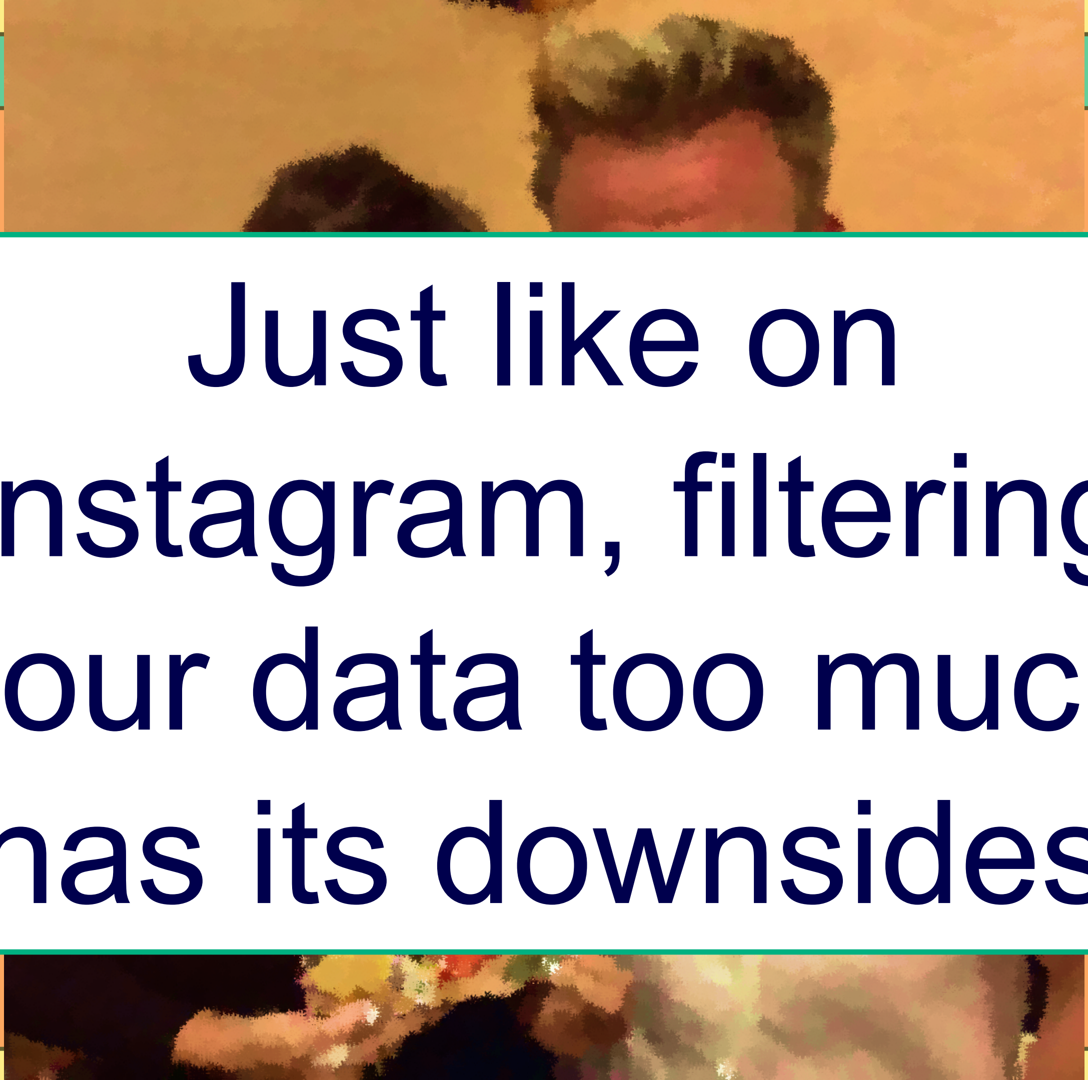
# Filtering

The ***smooth*** function (i.e., moving average, can be considered a type of filter).

Similarly, removing outliers can also act as a ***Low Pass*** filter, removing 'spikes' in your data

Filters come in many shapes and forms, but ultimately all aim to *'clean'* your data... but there is such thing as too much of a good thing


THE KITCHEN WITH KRIS
KRIS JENNER



Just like on Instagram, filtering your data too much has its downsides

# Filtering

Filters can be considered either:

**Lowpass:** only keeps information below the defined frequency

**Highpass:** only keeps the information above the defined frequency

**Bandpass:** only keeps the information within the defined range

# Filtering

There are **digital** and analog **filters**, for the most part what you will care about are digital filters

Infinite Impulse Response **IIR** are filters whose impulse response do not fall to zero (infinite series)

Finite Impulse Response **FIR** are filters whose impulse response are finite and will reach zero

# Filtering

Infinite Impulse Response **IIR** work faster and take up less memory on a computer while executing

Finite Impulse Response **FIR** do not introduce nonlinear phase responses (i.e., all frequencies are shifted by the same amount in phase, creating no phase distortions)

# Spectral analysis cont

FFT and the Welch method are not the only options for spectral analysis, we can compute changes in frequency across time.

Yet, there is an inherent tradeoff between frequency and time resolution

# Frequency Analysis

A note on frequency analysis:

While there are many methods to compute power (e.g., FFT, morlet waves, Hilbert, etc) every measure is limited by a tradeoff between accuracy in **time** vs **frequency**

This **tradeoff** means that you can either be precise in your measure of frequency or time but never both

# Hilbert Transform

Transformation that takes a real timeseries and returns the analytic signal represented by the **instantaneous amplitude** of the time series and the **phase** of the oscillation

This is one way of computing spectral power over time!

# Hilbert Transform

The Hilbert transform takes the signal **s(t)** and tries to express it as an analytic signal with both composed of both **real** and **imaginary** parts

$$s_A(t) = A(t)e^{j\psi(t)}$$

# Hilbert Transform

In MATLAB, the function **Hilbert()** returns the complex number output of your signal. The real part **abs(hil)** reflecting the **instantaneous amplitude**, and the imaginary part **angle(hil)** being the **phase** of the signal

# Find peaks

Useful function in MATLAB to find local maxima or minima in your data based on several criteria

Can modify function to use different local peak criteria

Can input descriptive statistics for reference (e.g. std)

# Find peaks

'MinPeakDistance' –distance between peaks

'Npeaks' –maximum number of peaks

'MinPeakHeight' –minimum height of peak

'MinPeakProminence' –minimum prominence of peak

'Threshold' –minimum difference between peak and neighbours

Etc..

# MATLAB Toolboxes

There are many toolboxes in addition to the basic functions of MATLAB, some are developed my MATLAB and others are **external** and need downloading

# MATLAB Toolboxes

Most external toolboxes can be downloaded from a website and installed either by running a command in the MATLAB terminal or by **adding** the toolbox to your **MATLAB path**

# Example Toolboxes

EEGLAB

SPM

Brainstorm

# Example Toolboxes

NOTE: not all toolboxes are compatible with one another, some may share functions with the same name. Be aware of which functions/ toolboxes are on your MATLAB path