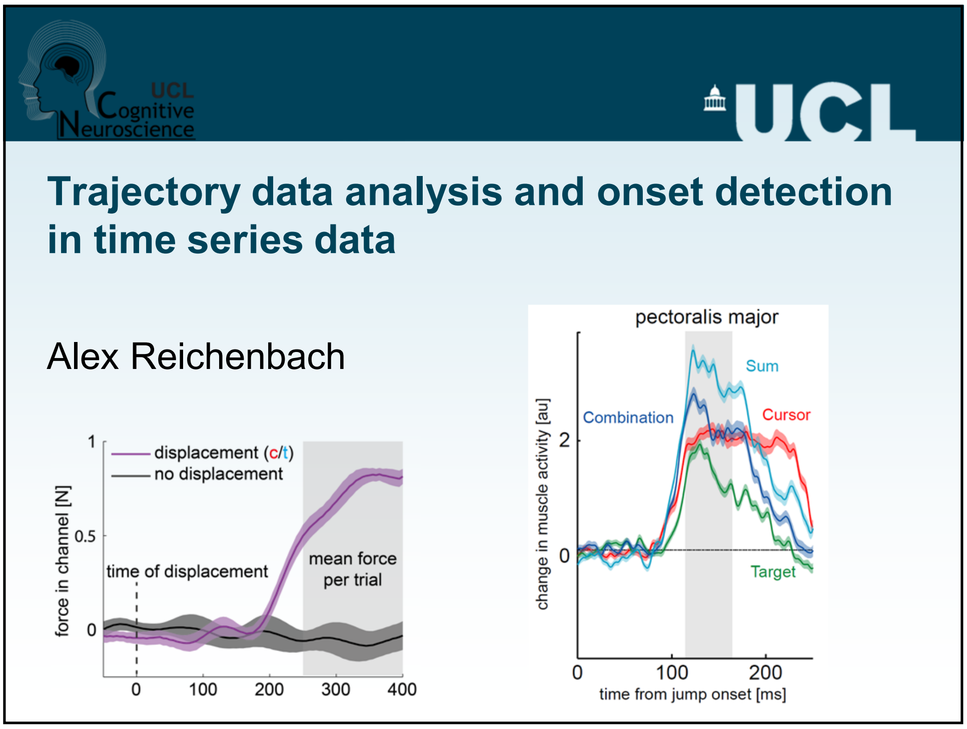
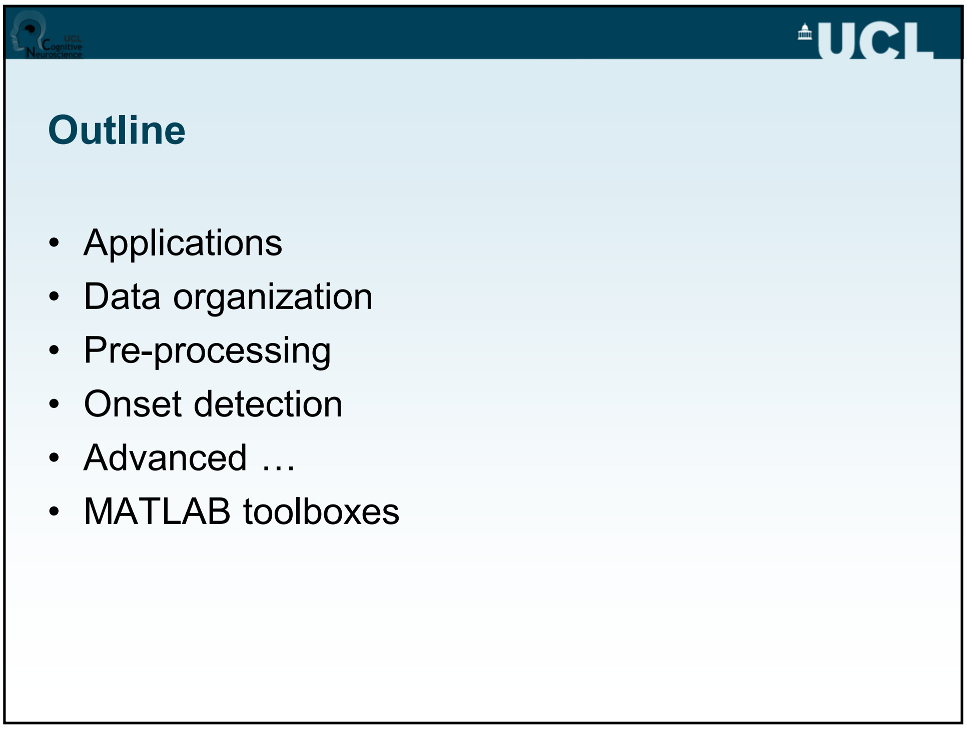


UCL Cognitive Neuroscience

UCL

Trajectory data analysis and onset detection in time series data

Alex Reichenbach

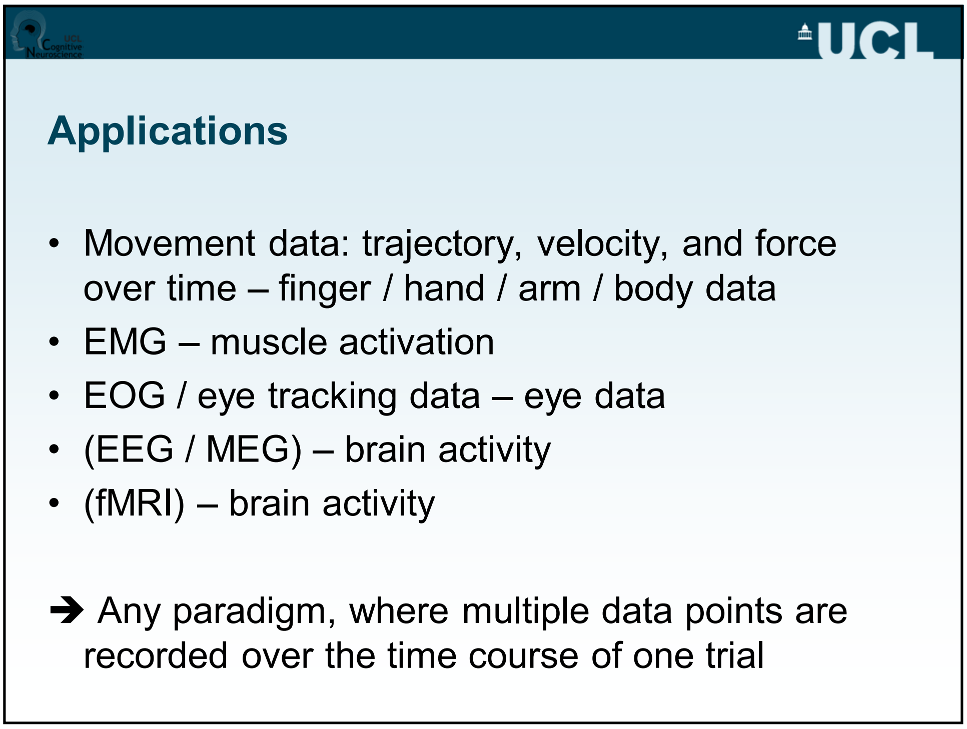


UCL Cognitive Neuroscience

UCL

Outline

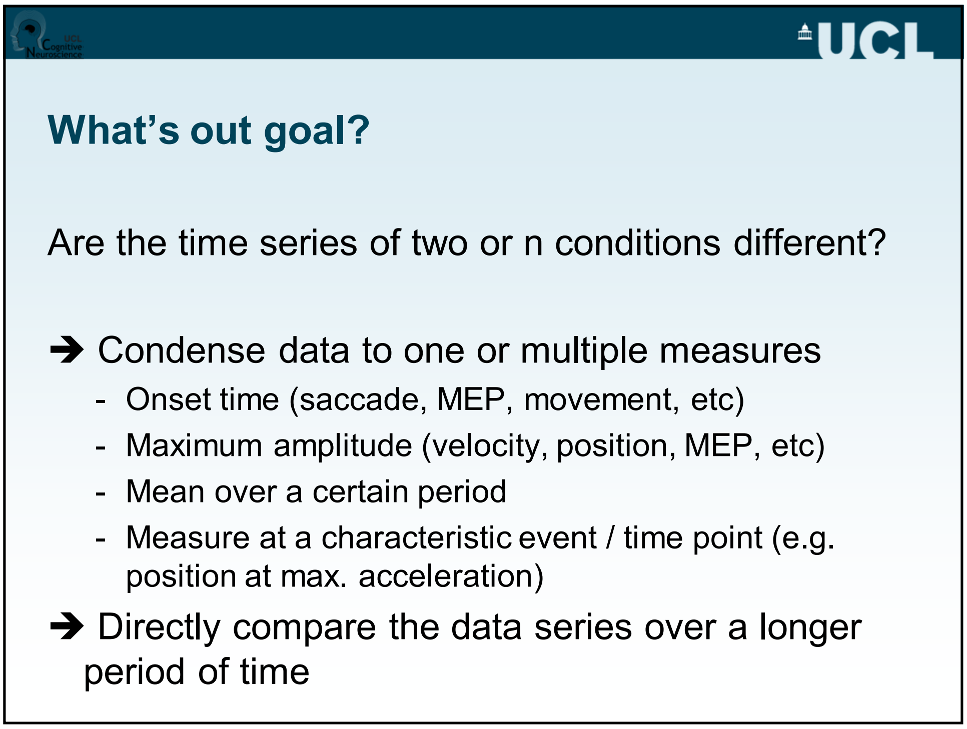
- Applications
- Data organization
- Pre-processing
- Onset detection
- Advanced …
- MATLAB toolboxes



Applications

- Movement data: trajectory, velocity, and force over time – finger / hand / arm / body data
- EMG – muscle activation
- EOG / eye tracking data – eye data
- (EEG / MEG) – brain activity
- (fMRI) – brain activity

➔ Any paradigm, where multiple data points are recorded over the time course of one trial



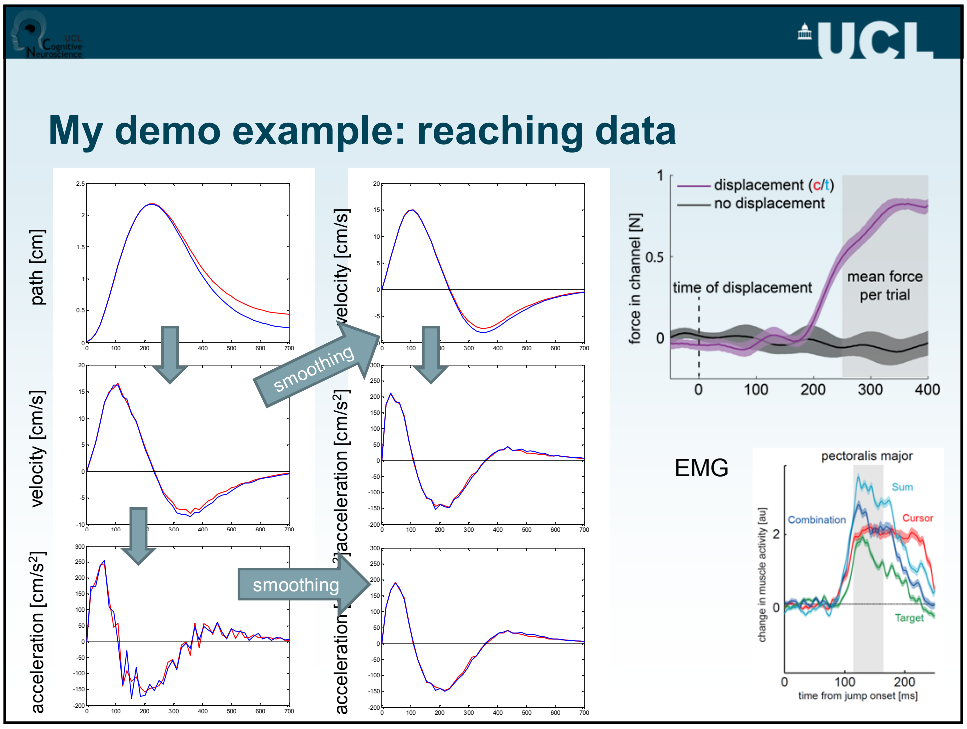
What’s out goal?

Are the time series of two or n conditions different?

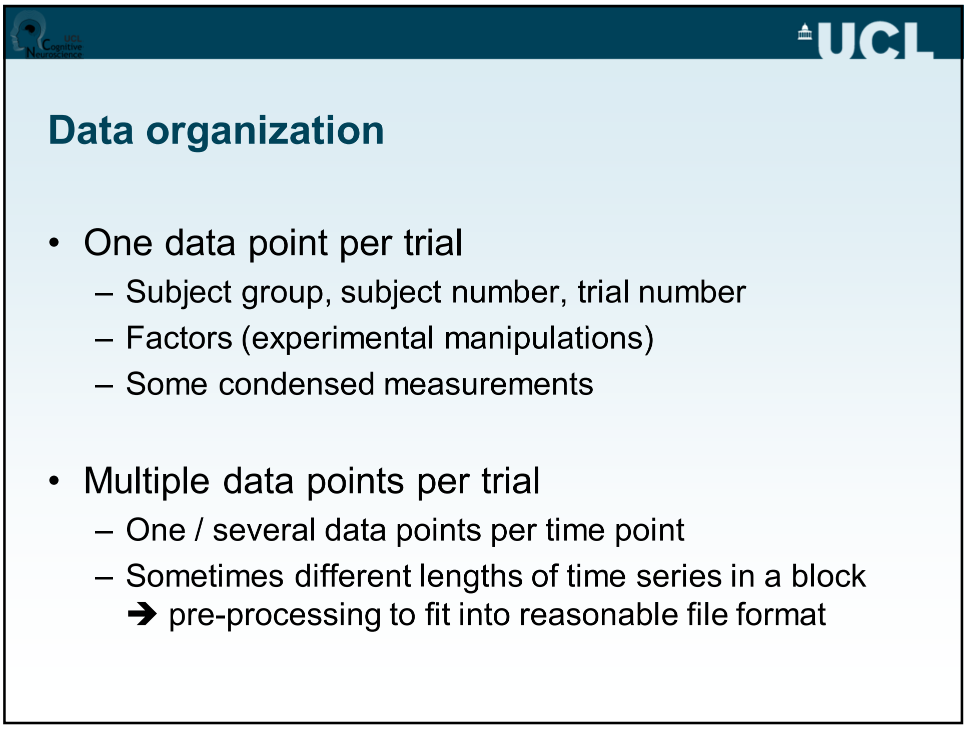
➔ Condense data to one or multiple measures

- Onset time (saccade, MEP, movement, etc)
- Maximum amplitude (velocity, position, MEP, etc)
- Mean over a certain period
- Measure at a characteristic event / time point (e.g. position at max. acceleration)

➔ Directly compare the data series over a longer period of time



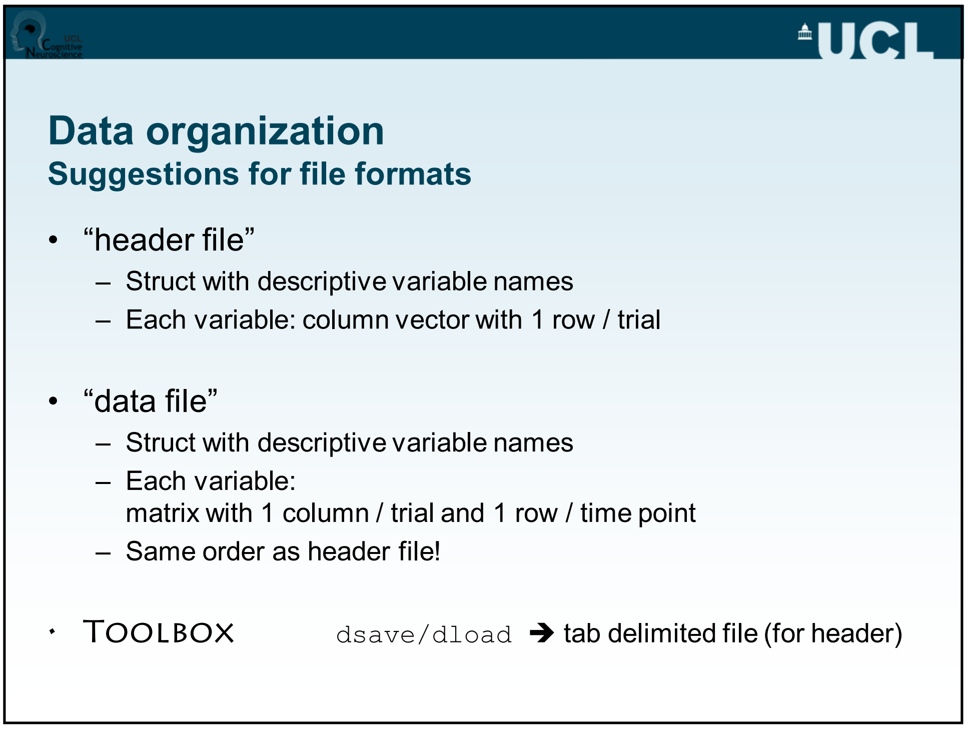
My demo example: reaching data



UCL Cognitive Neuroscience
UCL

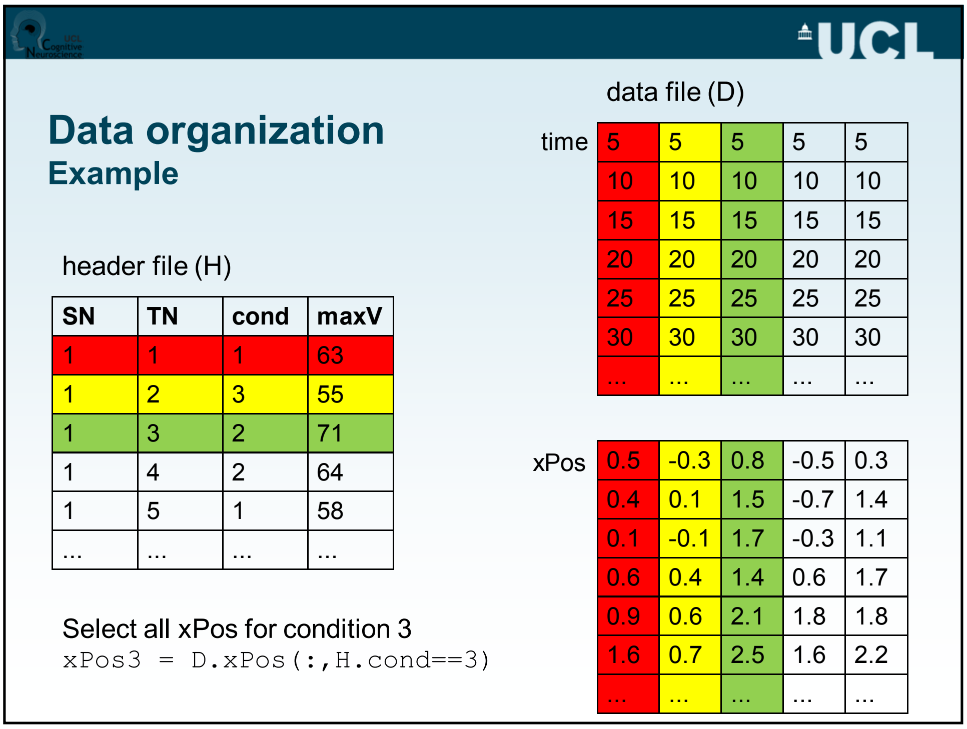
Data organization

- One data point per trial
  - Subject group, subject number, trial number
  - Factors (experimental manipulations)
  - Some condensed measurements
- Multiple data points per trial
  - One / several data points per time point
  - Sometimes different lengths of time series in a block
    ➔ pre-processing to fit into reasonable file format



Data organization
Suggestions for file formats

- “header file”
  - Struct with descriptive variable names
  - Each variable: column vector with 1 row / trial
- “data file”
  - Struct with descriptive variable names
  - Each variable:
    matrix with 1 column / trial and 1 row / time point
  - Same order as header file!
- TOOLBOX `dsave/dload` ➔ tab delimited file (for header)



Data organization
Example

header file (H)

SN	TN	cond	maxV
1	1	1	63
1	2	3	55
1	3	2	71
1	4	2	64
1	5	1	58
…	…	…	…

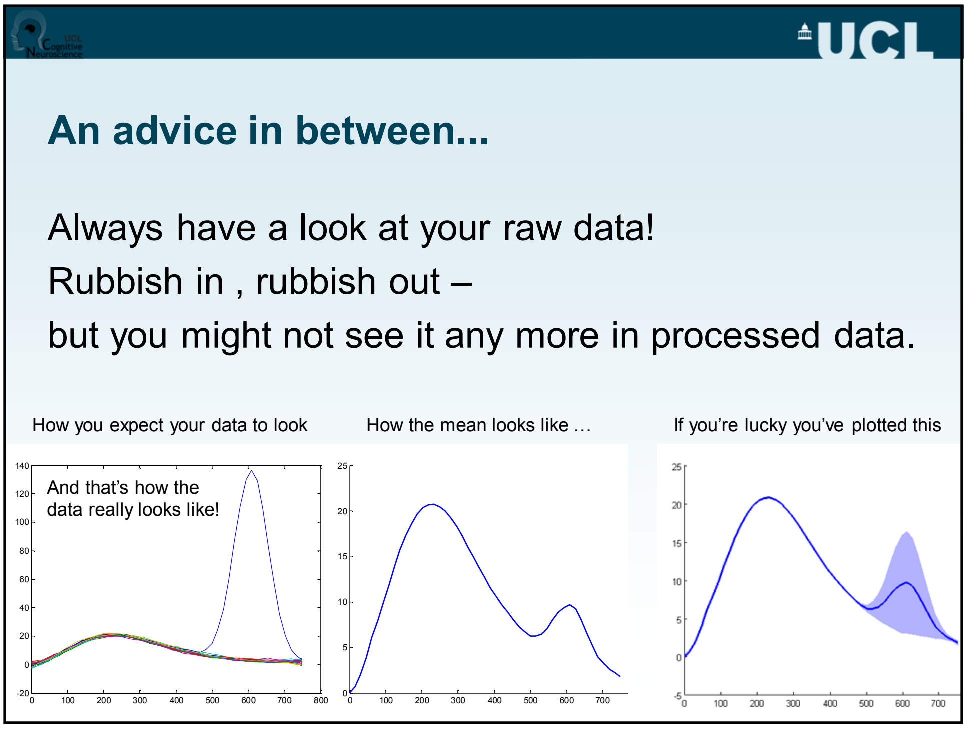
Select all xPos for condition 3

```
xPos3 = D.xPos(:,H.cond==3)
```

data file (D)

time	5	5	5	5	5
	10	10	10	10	10
	15	15	15	15	15
	20	20	20	20	20
	25	25	25	25	25
	30	30	30	30	30
	…	…	…	…	…

xPos	0.5	-0.3	0.8	-0.5	0.3
	0.4	0.1	1.5	-0.7	1.4
	0.1	-0.1	1.7	-0.3	1.1
	0.6	0.4	1.4	0.6	1.7
	0.9	0.6	2.1	1.8	1.8
	1.6	0.7	2.5	1.6	2.2
	…	…	…	…	…



An advice in between...

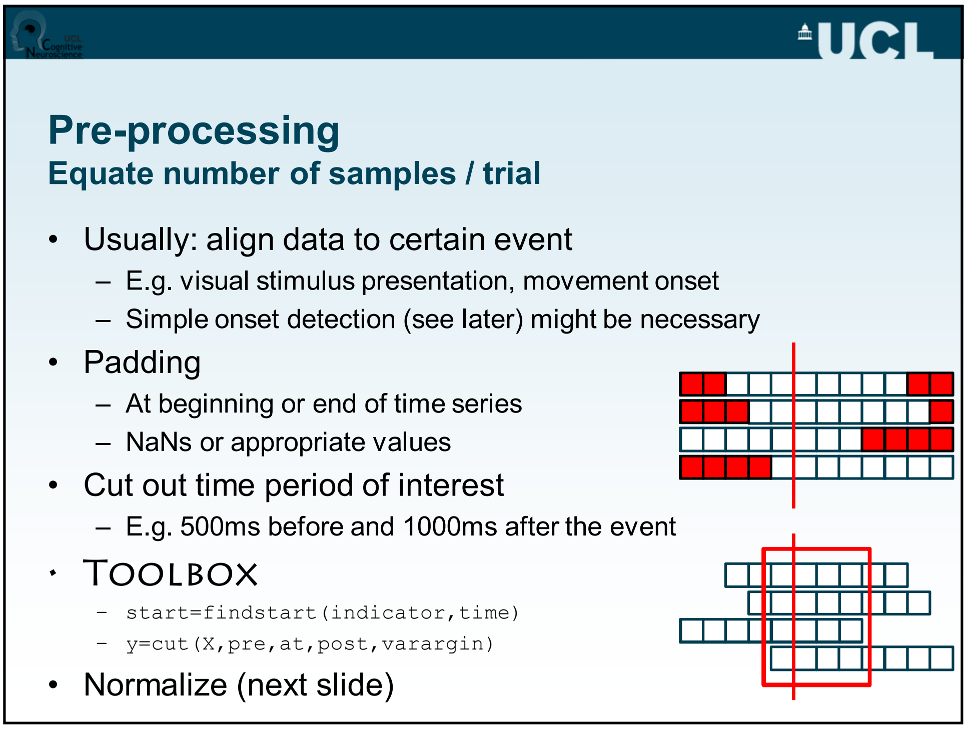
Always have a look at your raw data!
Rubbish in , rubbish out –
but you might not see it any more in processed data.

How you expect your data to look

How the mean looks like …

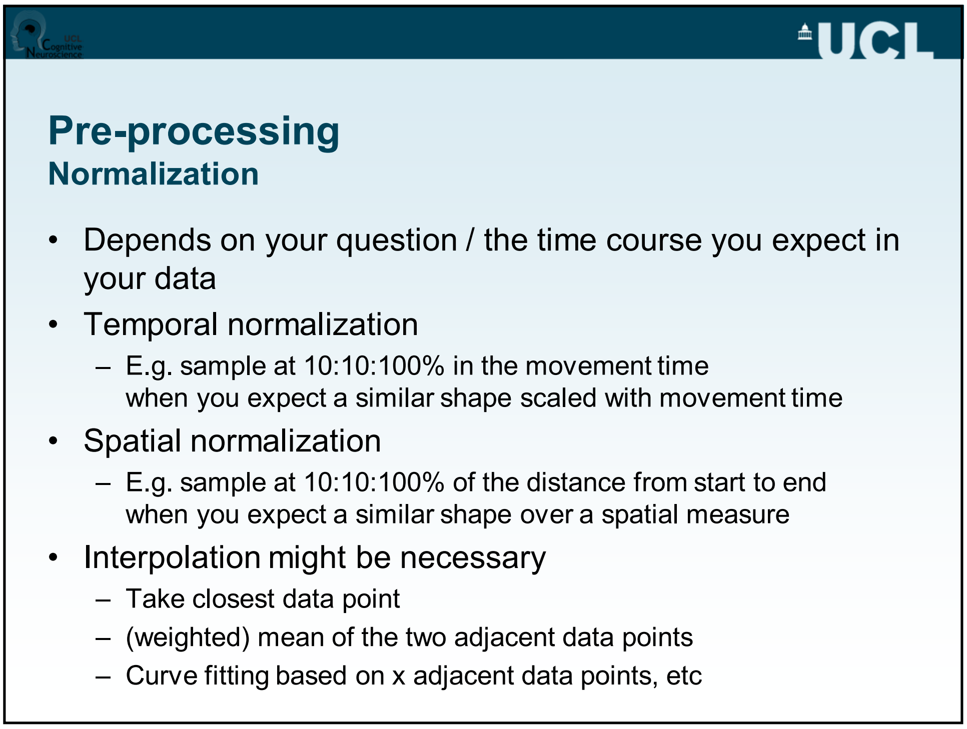
If you're lucky you've plotted this

And that's how the data really looks like!



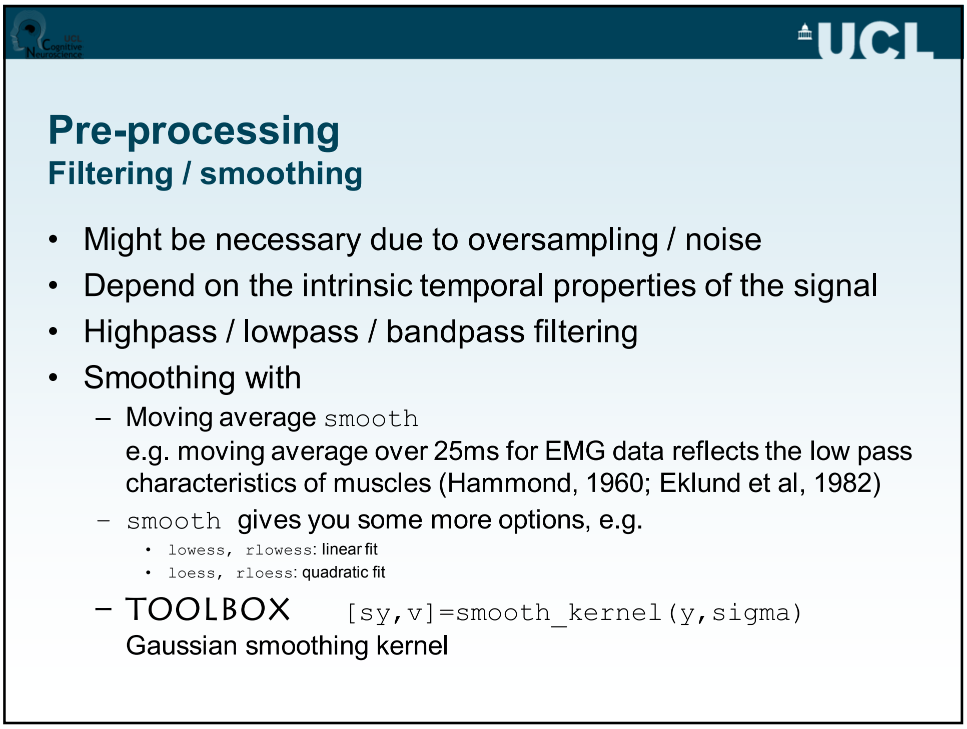
Pre-processing
Equate number of samples / trial

- Usually: align data to certain event
  - E.g. visual stimulus presentation, movement onset
  - Simple onset detection (see later) might be necessary
- Padding
  - At beginning or end of time series
  - NaNs or appropriate values
- Cut out time period of interest
  - E.g. 500ms before and 1000ms after the event
- TOOLBOX
  - `start=findstart(indicator,time)`
  - `y=cut(X,pre,at,post,varargin)`
- Normalize (next slide)



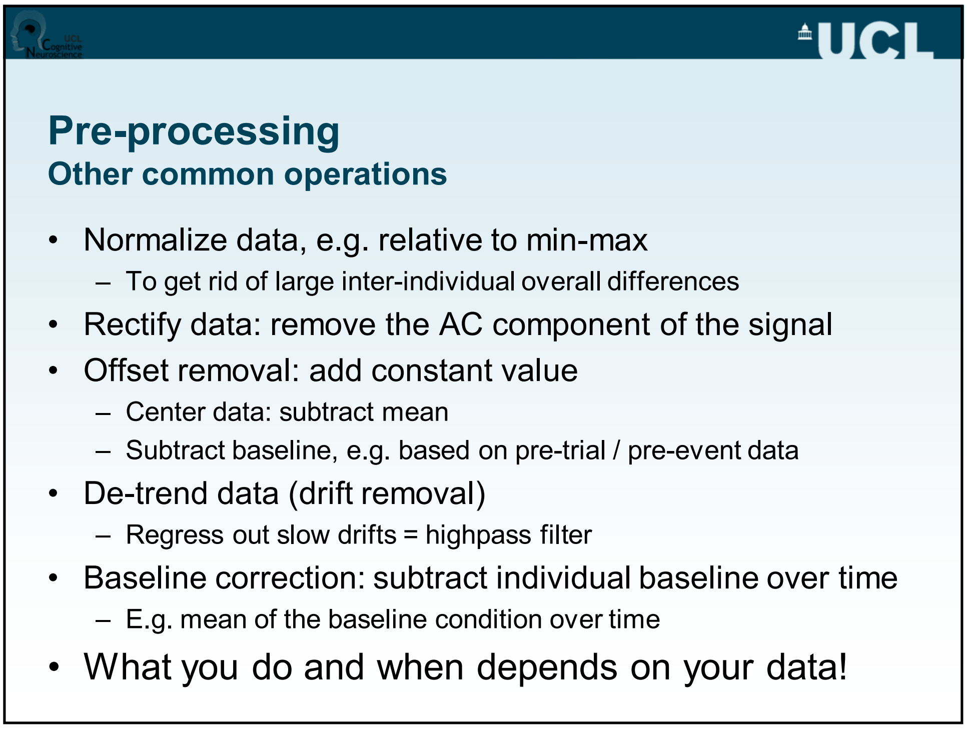
Pre-processing
Normalization

- Depends on your question / the time course you expect in your data
- Temporal normalization
  - E.g. sample at 10:10:100% in the movement time when you expect a similar shape scaled with movement time
- Spatial normalization
  - E.g. sample at 10:10:100% of the distance from start to end when you expect a similar shape over a spatial measure
- Interpolation might be necessary
  - Take closest data point
  - (weighted) mean of the two adjacent data points
  - Curve fitting based on x adjacent data points, etc



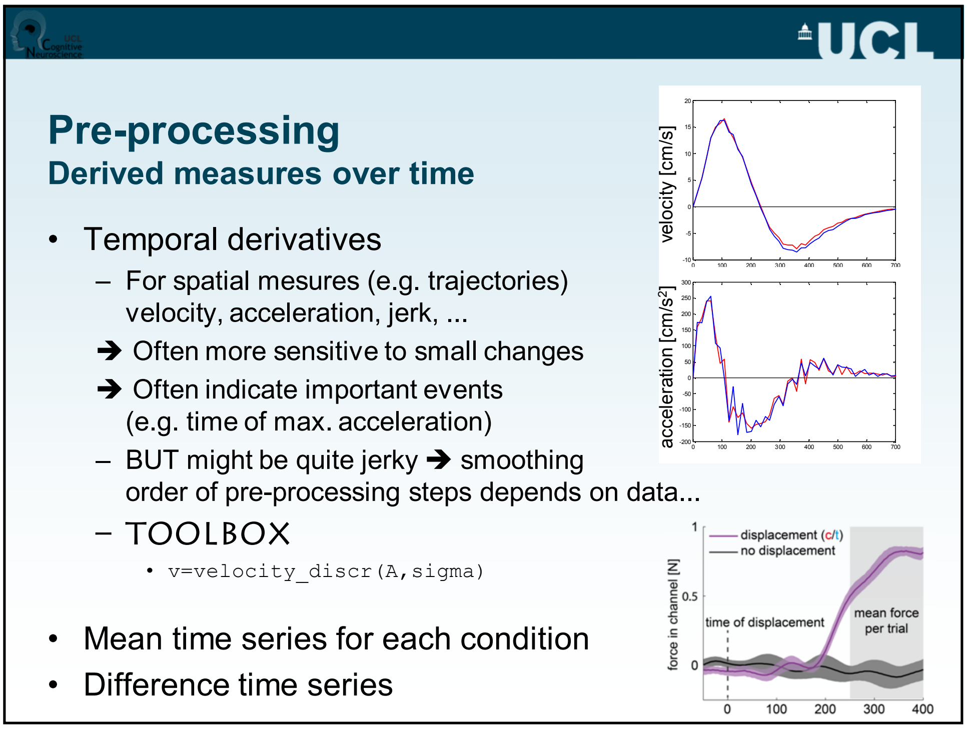
Pre-processing
Filtering / smoothing

- Might be necessary due to oversampling / noise
- Depend on the intrinsic temporal properties of the signal
- Highpass / lowpass / bandpass filtering
- Smoothing with
  - Moving average `smooth`
    e.g. moving average over 25ms for EMG data reflects the low pass characteristics of muscles (Hammond, 1960; Eklund et al, 1982)
  - `smooth` gives you some more options, e.g.
    - `lowess, rlowess`: linear fit
    - `loess, rloess`: quadratic fit
  - TOOLBOX `[sy,v]=smooth_kernel(y,sigma)`
    Gaussian smoothing kernel



Pre-processing
Other common operations

- Normalize data, e.g. relative to min-max
  - To get rid of large inter-individual overall differences
- Rectify data: remove the AC component of the signal
- Offset removal: add constant value
  - Center data: subtract mean
  - Subtract baseline, e.g. based on pre-trial / pre-event data
- De-trend data (drift removal)
  - Regress out slow drifts = highpass filter
- Baseline correction: subtract individual baseline over time
  - E.g. mean of the baseline condition over time
- What you do and when depends on your data!

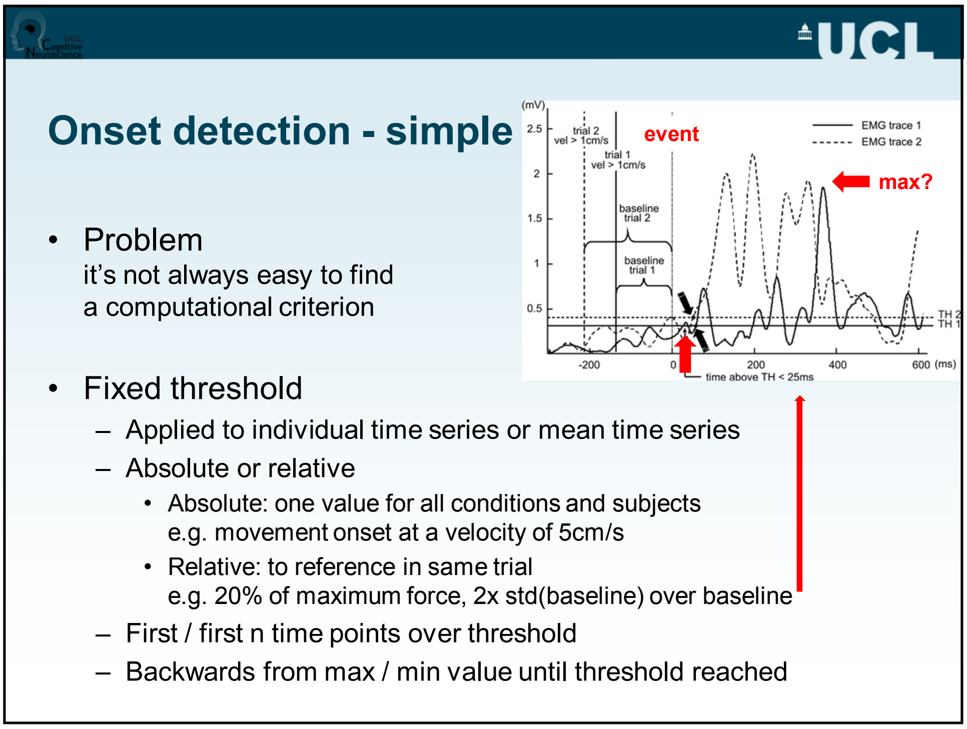


Pre-processing
Derived measures over time

- Temporal derivatives
  - For spatial mesures (e.g. trajectories) velocity, acceleration, jerk, ...

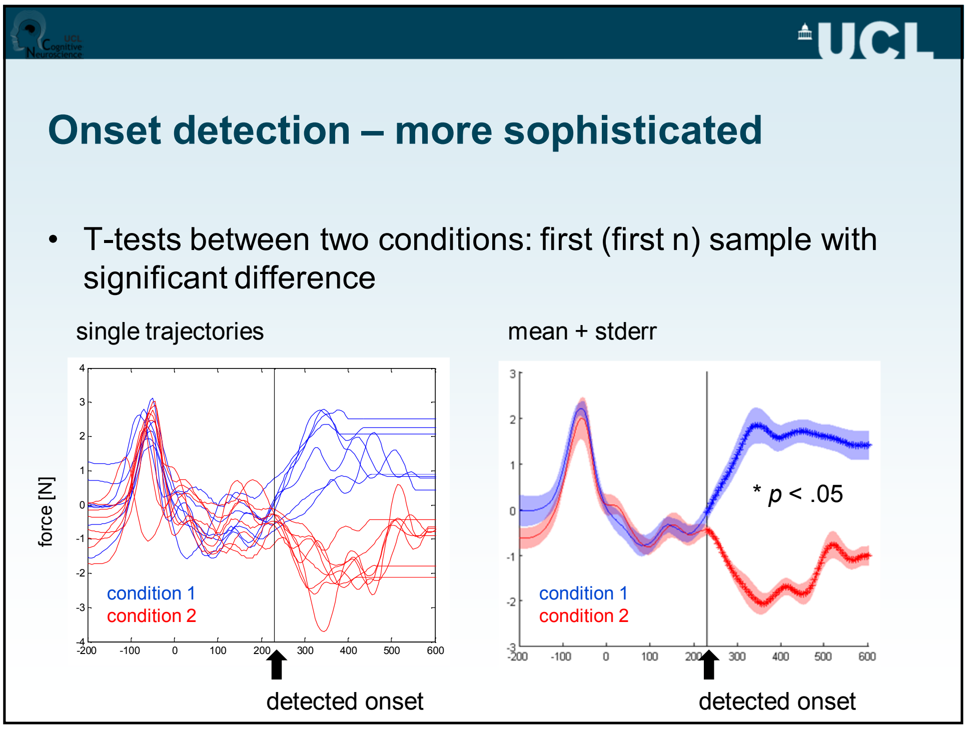
    ➔ Often more sensitive to small changes

    ➔ Often indicate important events (e.g. time of max. acceleration)
  - BUT might be quite jerky ➔ smoothing order of pre-processing steps depends on data...
  - TOOLBOX
    - `v=velocity_discr(A,sigma)`
- Mean time series for each condition
- Difference time series



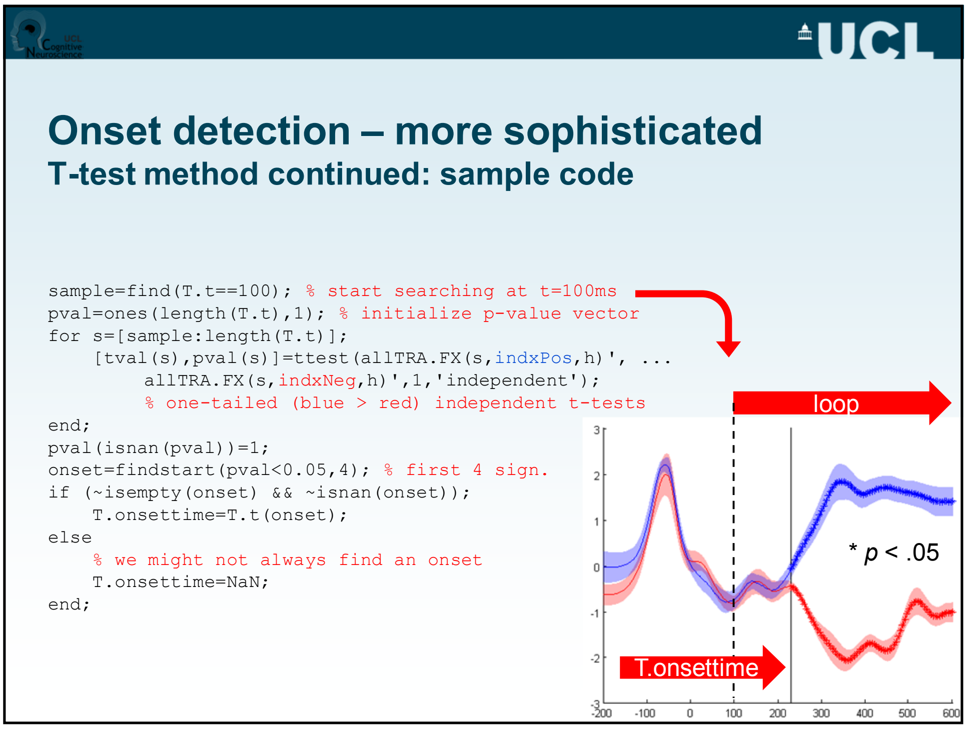
Onset detection - simple

- Problem
  it's not always easy to find a computational criterion
- Fixed threshold
  - Applied to individual time series or mean time series
  - Absolute or relative
    - Absolute: one value for all conditions and subjects
      e.g. movement onset at a velocity of 5cm/s
    - Relative: to reference in same trial
      e.g. 20% of maximum force, 2x std(baseline) over baseline
  - First / first n time points over threshold
  - Backwards from max / min value until threshold reached



Onset detection – more sophisticated

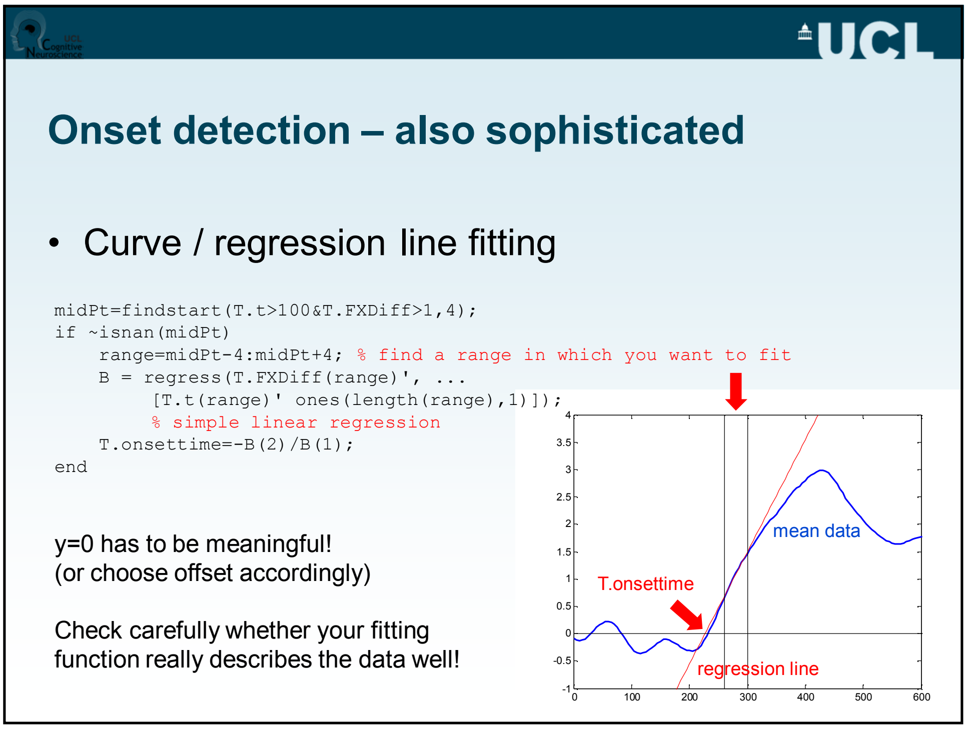
- T-tests between two conditions: first (first n) sample with significant difference



Onset detection – more sophisticated

T-test method continued: sample code

```
sample=find(T.t==100); % start searching at t=100ms
pval=ones(length(T.t),1); % initialize p-value vector
for s=[sample:length(T.t)];
    [tval(s),pval(s)]=ttest(allTRA.FX(s,indxPos,h)', ...
         allTRA.FX(s,indxNeg,h)',1,'independent');
         % one-tailed (blue > red) independent t-tests
end;
pval(isnan(pval))=1;
onset=findstart(pval<0.05,4); % first 4 sign.
if (~isempty(onset) && ~isnan(onset));
    T.onsettime=T.t(onset);
else
    % we might not always find an onset
    T.onsettime=NaN;
end;
```



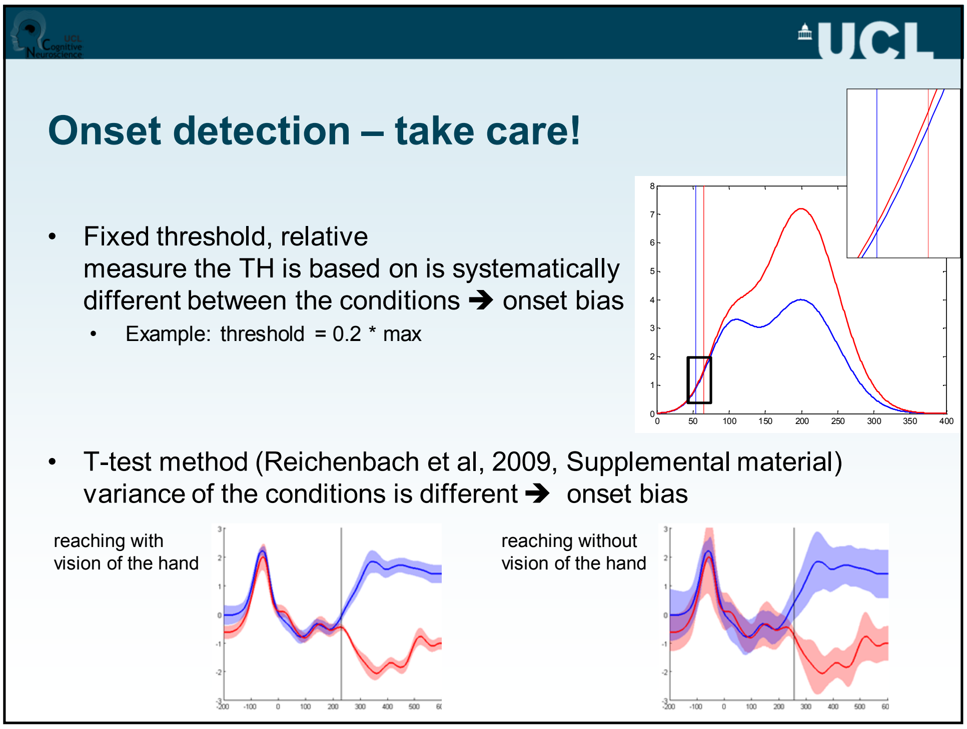
Onset detection – also sophisticated

- Curve / regression line fitting

```
midPt=findstart(T.t>100&T.FXDiff>1,4);
if ~isnan(midPt)
    range=midPt-4:midPt+4; % find a range in which you want to fit
    B = regress(T.FXDiff(range)', ...
         [T.t(range)' ones(length(range),1)]);
         % simple linear regression
    T.onsettime=-B(2)/B(1);
end
```

y=0 has to be meaningful!
(or choose offset accordingly)

Check carefully whether your fitting function really describes the data well!


UCL Cognitive Neuroscience
UCL
8
7
6
5
4
3
2
1
0
0
50
100
150
200
250
300
350
400
3
2
1
0
-1
-2
-3
-200
-100
0
100
200
300
400
500
600

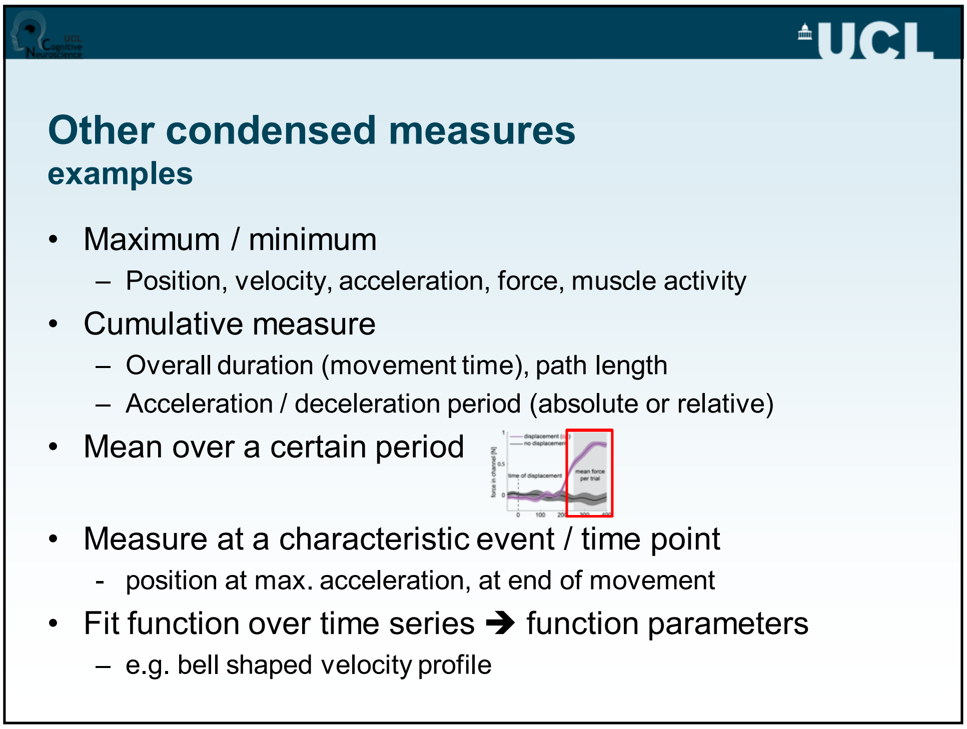
Onset detection – take care!

- Fixed threshold, relative
  measure the TH is based on is systematically different between the conditions ➔ onset bias
  - Example: threshold = 0.2 * max

- T-test method (Reichenbach et al, 2009, Supplemental material)
  variance of the conditions is different ➔ onset bias

reaching with vision of the hand

reaching without vision of the hand

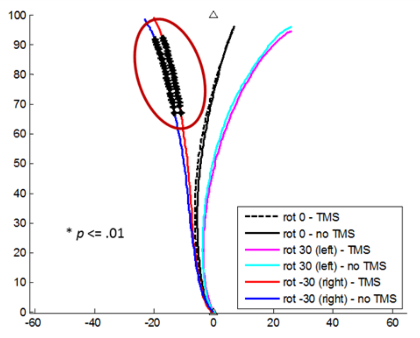

UCL Cognitive Neuroscience
UCL
1
0.5
0
force in channel [N]
displacement (c)
no displacement
time of displacement
mean force per trial
0
100
200
300
400

Other condensed measures
examples

- Maximum / minimum
  – Position, velocity, acceleration, force, muscle activity
- Cumulative measure
  – Overall duration (movement time), path length
  – Acceleration / deceleration period (absolute or relative)
- Mean over a certain period
- Measure at a characteristic event / time point
  - position at max. acceleration, at end of movement
- Fit function over time series ➔ function parameters
  – e.g. bell shaped velocity profile

Comparing whole time series

- Another t-test method
  - n t-tests in a row significant
  - Which n is sufficient?
    ➔ multiple comparison problem



➔Permutation test

- “clusterwise” correction for false positives,
- similar to fMRI analyses, only that the clusters are 1-dimensional (along the trajectories)

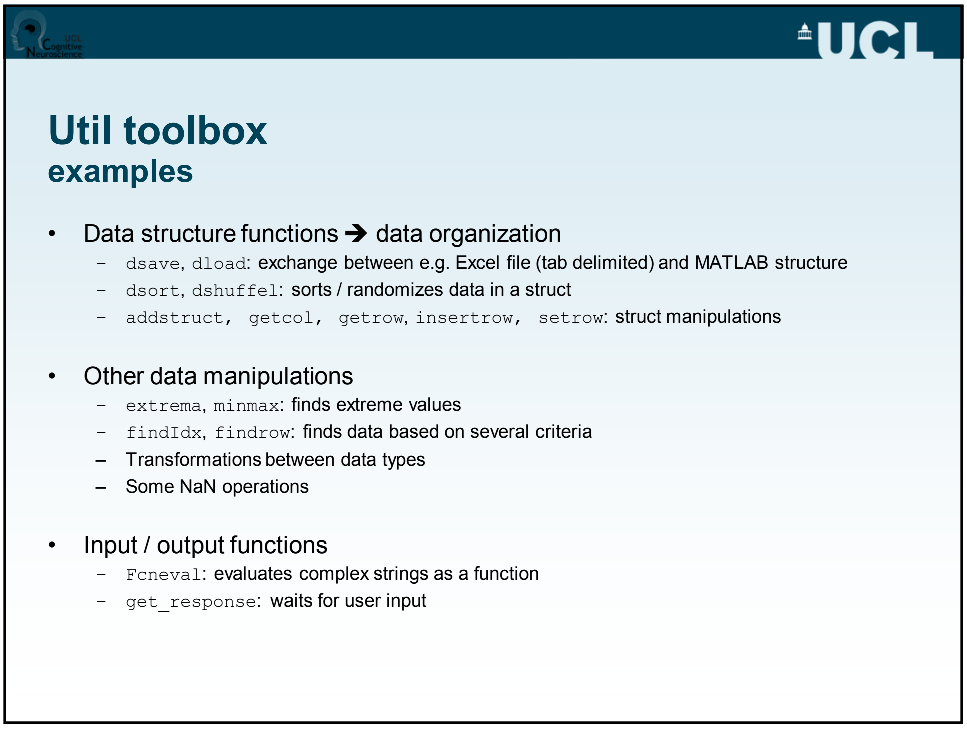
1. Permute the labels for the conditions for each subject, randomly assign them to the trajectories
2. t-tests over the trajectories on group level ➔ cluster size for each condition
3. 1000-10000 repetition ➔ chance level for each cluster size

MATLAB TOOLBOXES - overview

http://www.icn.ucl.ac.uk/motorcontrol/toolboxes/matlab_toolboxes.htm

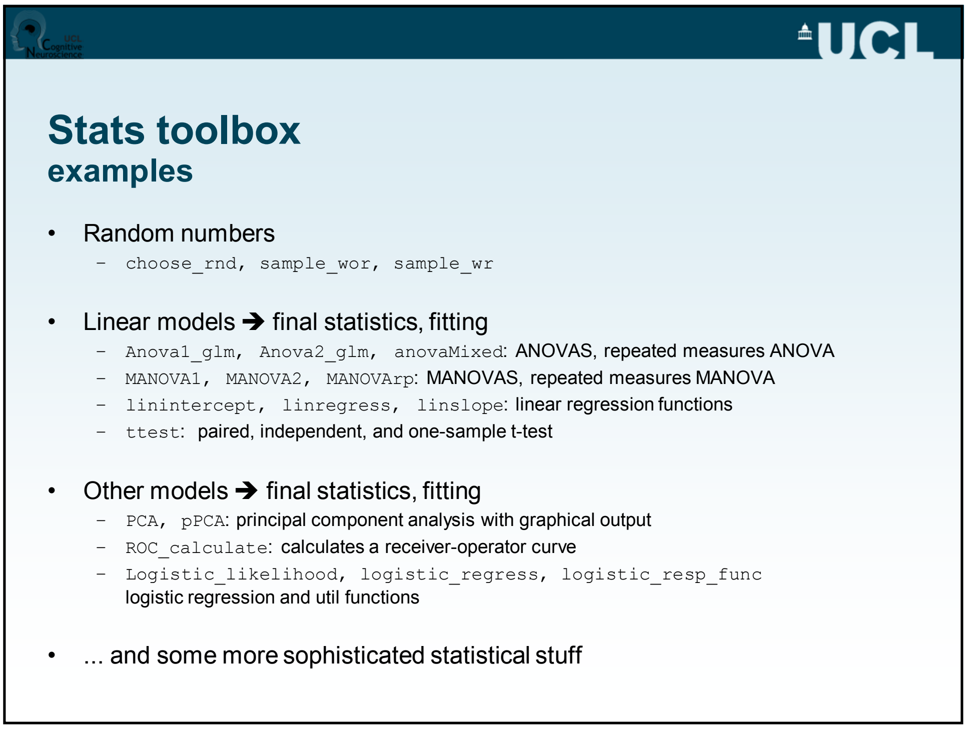
- **Util**
  - A lot of useful utility functions, esp. for structs
- **(Timeseries)**
  - Fourier transformation, frequency, autocovariance function, etc
- **Stats**
  - All kind of classical and advanced statistics
- **Kinematics**
  - Handling trajectories, velocity profiles, etc
- **Graph**
  - Very nice and highly flexible plotting functions
- **Pivot**
  - For quickly getting an overview over data
- **(Circ)**
  - Handling circular data

J. Diedrichsen



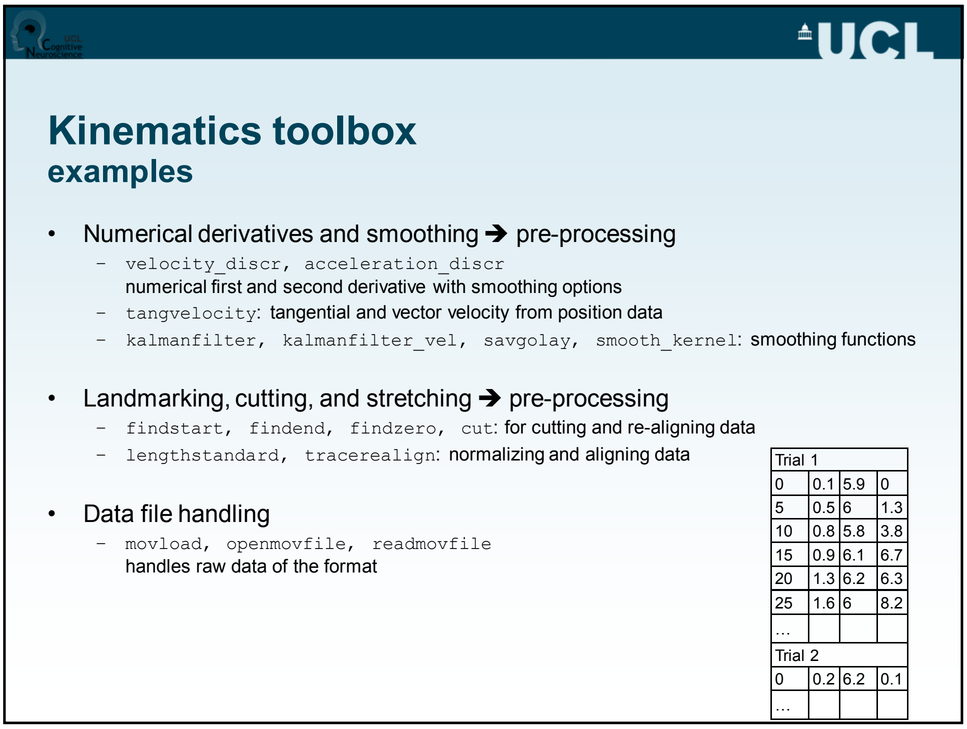
Util toolbox
examples

- Data structure functions ➔ data organization
  - `dsave`, `dload`: exchange between e.g. Excel file (tab delimited) and MATLAB structure
  - `dsort`, `dshuffel`: sorts / randomizes data in a struct
  - `addstruct`, `getcol`, `getrow`, `insertrow`, `setrow`: struct manipulations
- Other data manipulations
  - `extrema`, `minmax`: finds extreme values
  - `findIdx`, `findrow`: finds data based on several criteria
  - Transformations between data types
  - Some NaN operations
- Input / output functions
  - `Fcneval`: evaluates complex strings as a function
  - `get_response`: waits for user input



Stats toolbox
examples

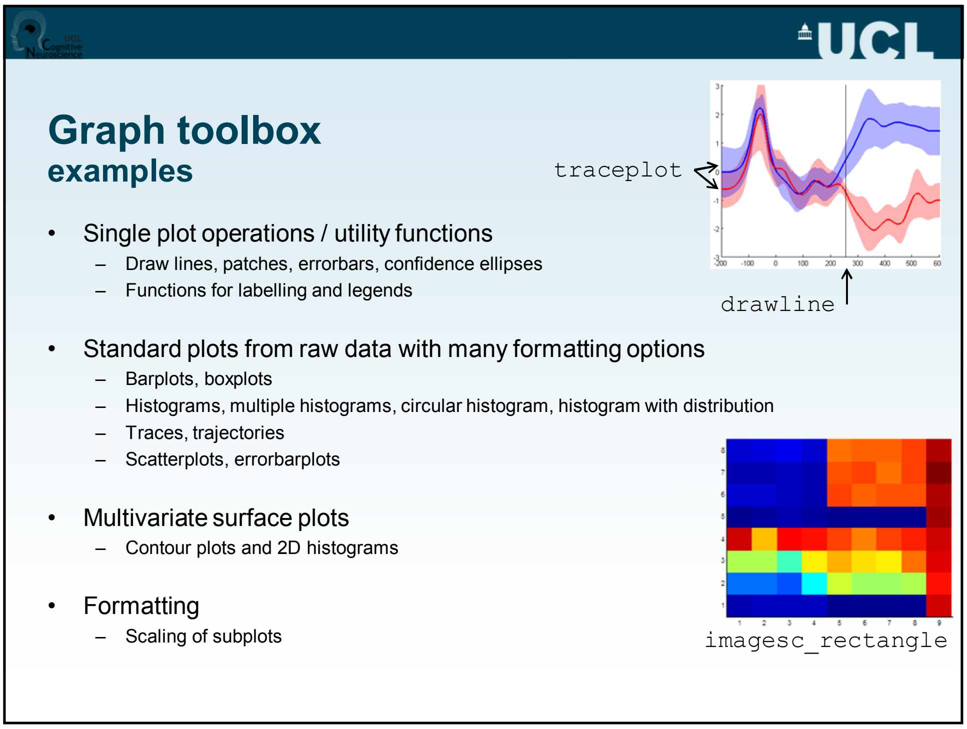
- Random numbers
  - `choose_rnd`, `sample_wor`, `sample_wr`
- Linear models ➔ final statistics, fitting
  - `Anova1_glm`, `Anova2_glm`, `anovaMixed`: ANOVAS, repeated measures ANOVA
  - `MANOVA1`, `MANOVA2`, `MANOVArp`: MANOVAS, repeated measures MANOVA
  - `linintercept`, `linregress`, `linslope`: linear regression functions
  - `ttest`: paired, independent, and one-sample t-test
- Other models ➔ final statistics, fitting
  - `PCA`, `pPCA`: principal component analysis with graphical output
  - `ROC_calculate`: calculates a receiver-operator curve
  - `Logistic_likelihood`, `logistic_regress`, `logistic_resp_func` logistic regression and util functions
- ... and some more sophisticated statistical stuff



Kinematics toolbox
examples

- Numerical derivatives and smoothing ➔ pre-processing
  - `velocity_discr, acceleration_discr`
    numerical first and second derivative with smoothing options
  - `tangvelocity`: tangential and vector velocity from position data
  - `kalmanfilter, kalmanfilter_vel, savgolay, smooth_kernel`: smoothing functions
- Landmarking, cutting, and stretching ➔ pre-processing
  - `findstart, findend, findzero, cut`: for cutting and re-aligning data
  - `lengthstandard, tracerealign`: normalizing and aligning data
- Data file handling
  - `movload, openmovfile, readmovfile`
    handles raw data of the format

Trial 1			
0	0.1	5.9	0
5	0.5	6	1.3
10	0.8	5.8	3.8
15	0.9	6.1	6.7
20	1.3	6.2	6.3
25	1.6	6	8.2
…			
Trial 2			
0	0.2	6.2	0.1
…			



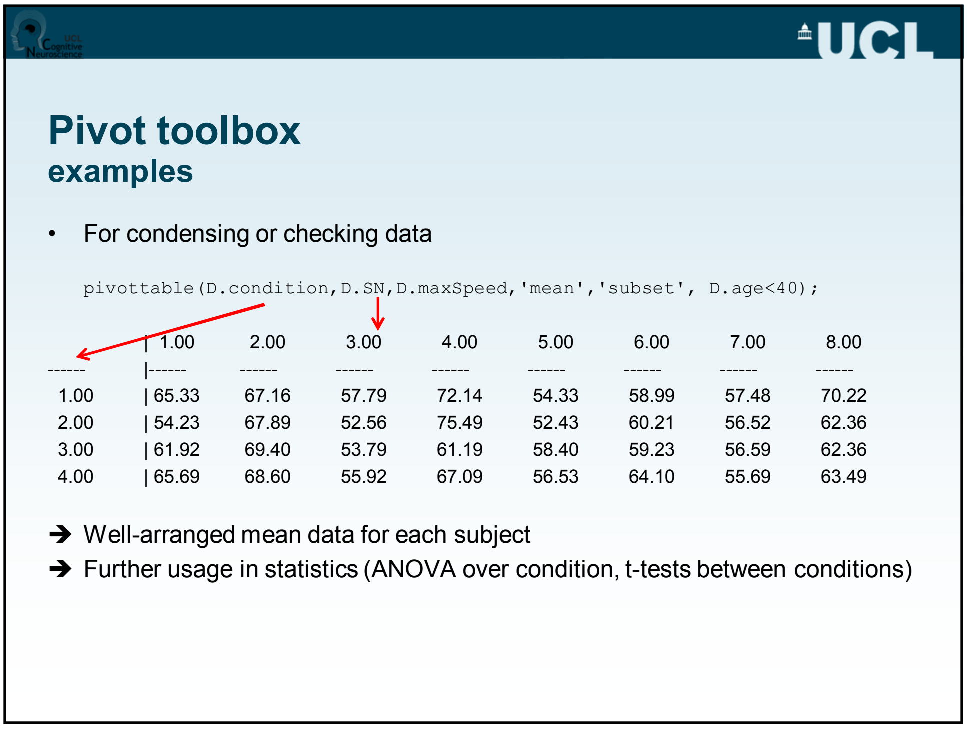
Graph toolbox
examples

`traceplot`

`drawline`

- Single plot operations / utility functions
  - Draw lines, patches, errorbars, confidence ellipses
  - Functions for labelling and legends
- Standard plots from raw data with many formatting options
  - Barplots, boxplots
  - Histograms, multiple histograms, circular histogram, histogram with distribution
  - Traces, trajectories
  - Scatterplots, errorbarplots
- Multivariate surface plots
  - Contour plots and 2D histograms
- Formatting
  - Scaling of subplots

`imagesc_rectangle`



Pivot toolbox
examples

- For condensing or checking data

```
pivottable(D.condition,D.SN,D.maxSpeed,'mean','subset', D.age<40);
```

```
          | 1.00     2.00     3.00     4.00     5.00     6.00     7.00     8.00
------    |------   ------   ------   ------   ------   ------   ------   ------
 1.00     | 65.33   67.16    57.79    72.14    54.33    58.99    57.48    70.22
 2.00     | 54.23   67.89    52.56    75.49    52.43    60.21    56.52    62.36
 3.00     | 61.92   69.40    53.79    61.19    58.40    59.23    56.59    62.36
 4.00     | 65.69   68.60    55.92    67.09    56.53    64.10    55.69    63.49
```

➔ Well-arranged mean data for each subject

➔ Further usage in statistics (ANOVA over condition, t-tests between conditions)