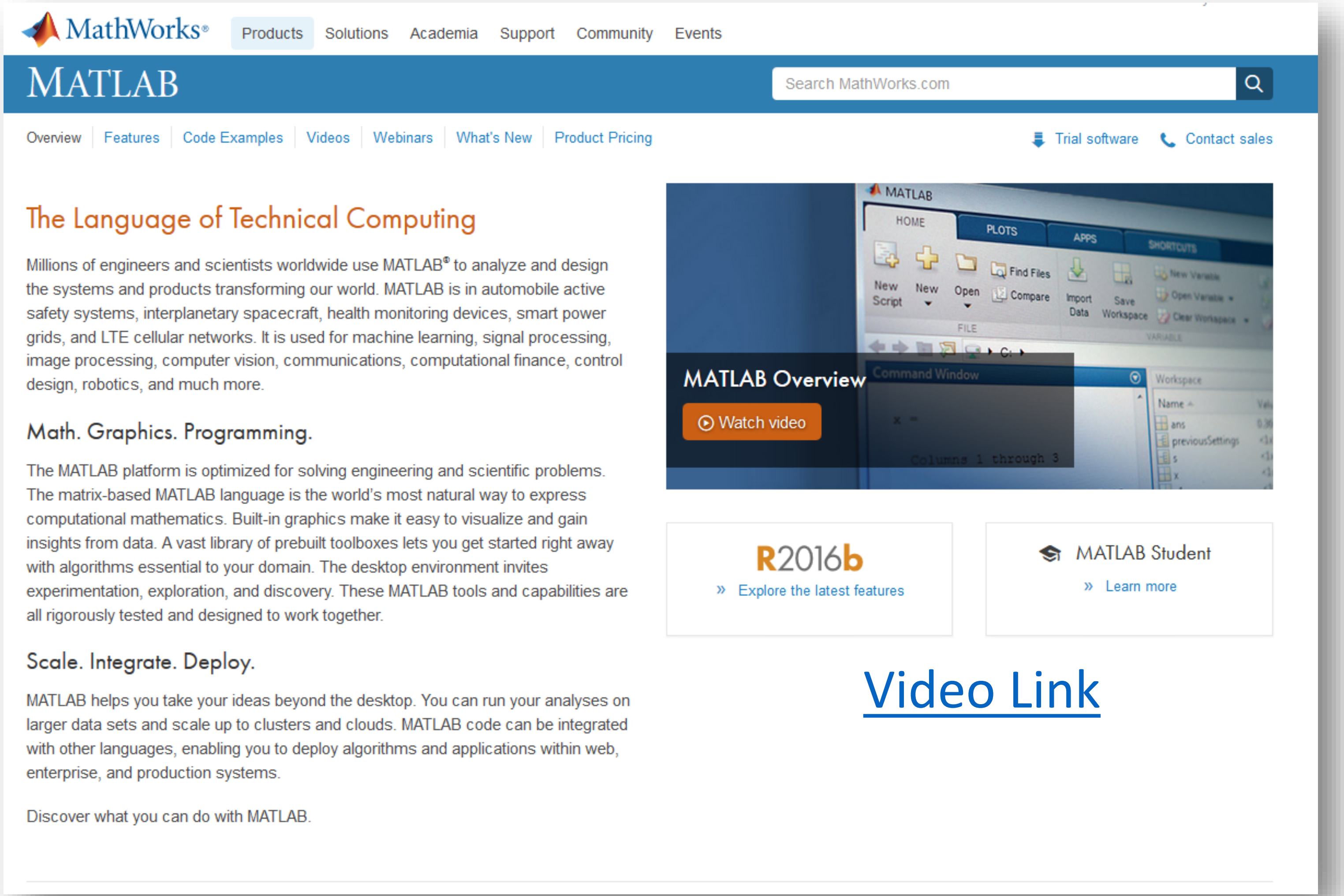
MATLAB



The Language of Technical Computing

Millions of engineers and scientists worldwide use MATLAB® to analyze and design the systems and products transforming our world. MATLAB is in automobile active safety systems, interplanetary spacecraft, health monitoring devices, smart power grids, and LTE cellular networks. It is used for machine learning, signal processing, image processing, computer vision, communications, computational finance, control design, robotics, and much more.

Math. Graphics. Programming.

The MATLAB platform is optimized for solving engineering and scientific problems. The matrix-based MATLAB language is the world's most natural way to express computational mathematics. Built-in graphics make it easy to visualize and gain insights from data. A vast library of prebuilt toolboxes lets you get started right away with algorithms essential to your domain. The desktop environment invites experimentation, exploration, and discovery. These MATLAB tools and capabilities are all rigorously tested and designed to work together.

Scale. Integrate. Deploy.

MATLAB helps you take your ideas beyond the desktop. You can run your analyses on larger data sets and scale up to clusters and clouds. MATLAB code can be integrated with other languages, enabling you to deploy algorithms and applications within web, enterprise, and production systems.

Discover what you can do with MATLAB.

Video Link

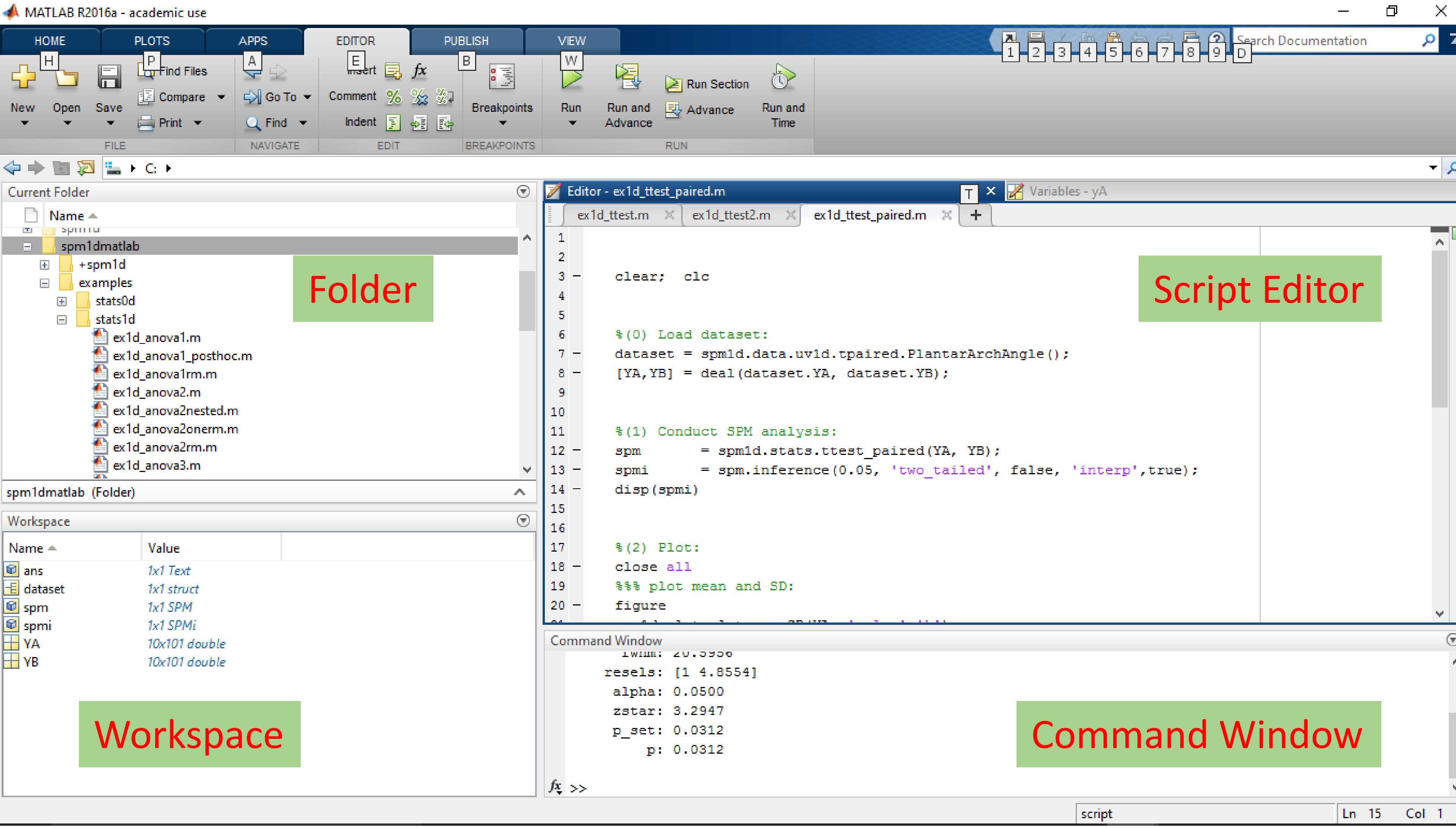
m.a.robinson@ljmu.ac.uk

Mark Robinson, Snr Lecturer in Biomechanics

MATLAB

- “MATrix LABoratory”
- Numerical array computations
- Useful for data analysis
- Broad functionality:
  - Statistics, optimization, signal processing, dynamic simulation, etc.

MATLAB



```
clear;  clc


%(0) Load dataset:
dataset = spm1d.data.uv1d.tpaired.PlantarArchAngle();
[YA,YB] = deal(dataset.YA, dataset.YB);


%(1) Conduct SPM analysis:
spm       = spm1d.stats.ttest_paired(YA, YB);
spmi      = spm.inference(0.05, 'two_tailed', false, 'interp',true);
disp(spmi)


%(2) Plot:
close all
%%% plot mean and SD:
figure
```

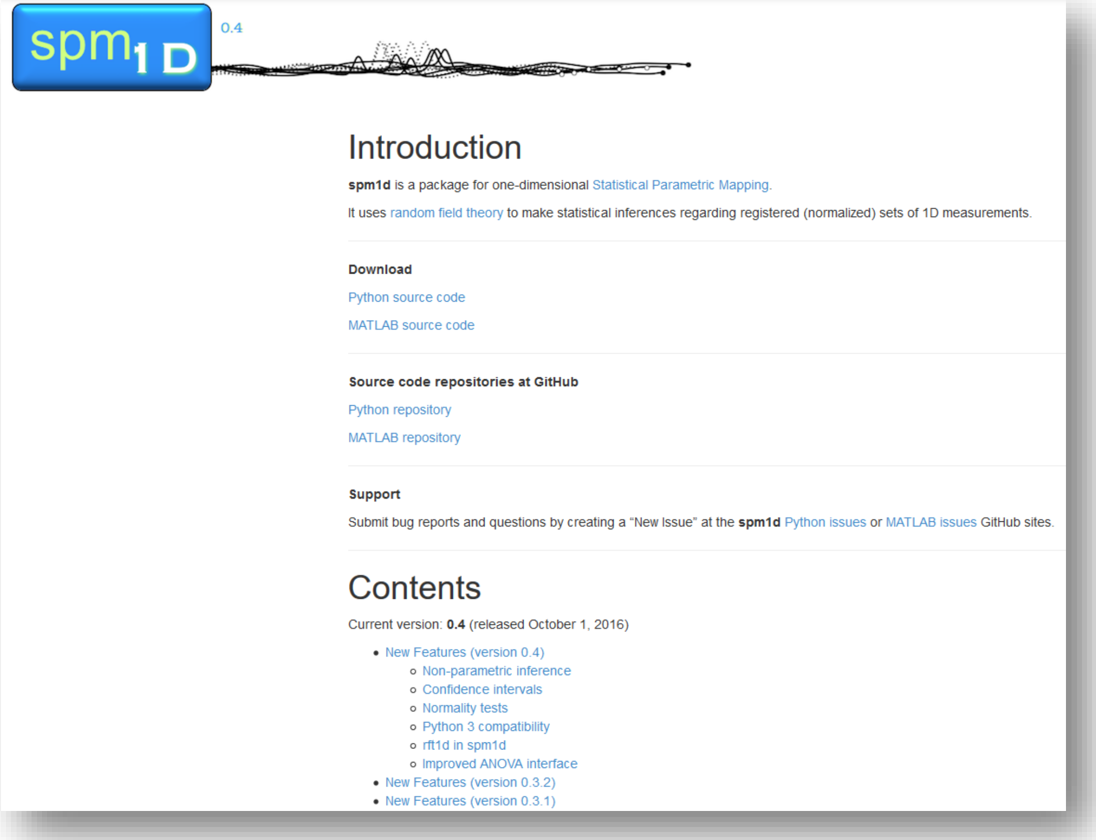
Name	Value
ans	1x1 Text
dataset	1x1 struct
spm	1x1 SPM
spmi	1x1 SPMi
YA	10x101 double
YB	10x101 double

```
    FWHM: 20.3956
  resels: [1 4.8554]
   alpha: 0.0500
   zstar: 3.2947
   p_set: 0.0312
       p: 0.0312

>>
```

3 examples

1. Workshops & tutorials – Statistical Parametric Mapping
2. Student tracking – Level 4 Year Tutor
3. Developing GUIs to assist student data analysis
    - a) Strength data analysis
    - b) Accelerometry analysis

1. Statistical parametric mapping



Introduction

spm1d is a package for one-dimensional Statistical Parametric Mapping.

It uses random field theory to make statistical inferences regarding registered (normalized) sets of 1D measurements.

Download

Python source code

MATLAB source code

Source code repositories at GitHub

Python repository

MATLAB repository

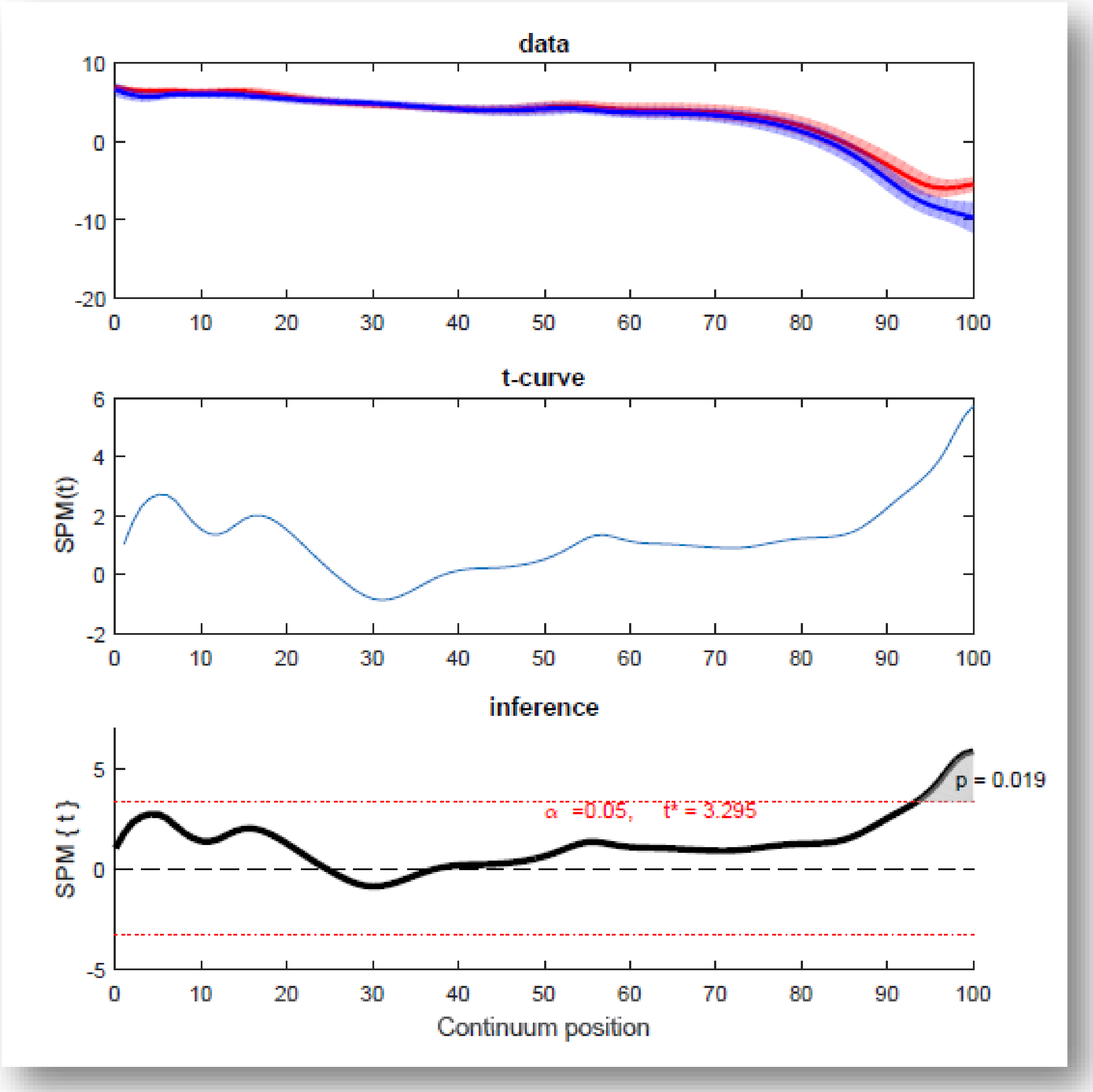
Support

Submit bug reports and questions by creating a “New Issue” at the **spm1d** Python issues or MATLAB issues GitHub sites.

Contents

Current version: **0.4** (released October 1, 2016)

- New Features (version 0.4)
  - Non-parametric inference
  - Confidence intervals
  - Normality tests
  - Python 3 compatibility
  - rft1d in spm1d
  - Improved ANOVA interface
- New Features (version 0.3.2)
- New Features (version 0.3.1)


data
10
0
-10
-20
0
10
20
30
40
50
60
70
80
90
100
t-curve
SPM(t)
6
4
2
0
-2
inference
SPM { t }
5
0
-5
α =0.05, t* = 3.295
p = 0.019
Continuum position

SPM1D
t-test Matlab function names

spm1d.stats.ttest(…)

One-sample t test

spm1d.stats.ttest_paired(…)

Paired t test

spm1d.stats_ttest2(….)

Two-sample test

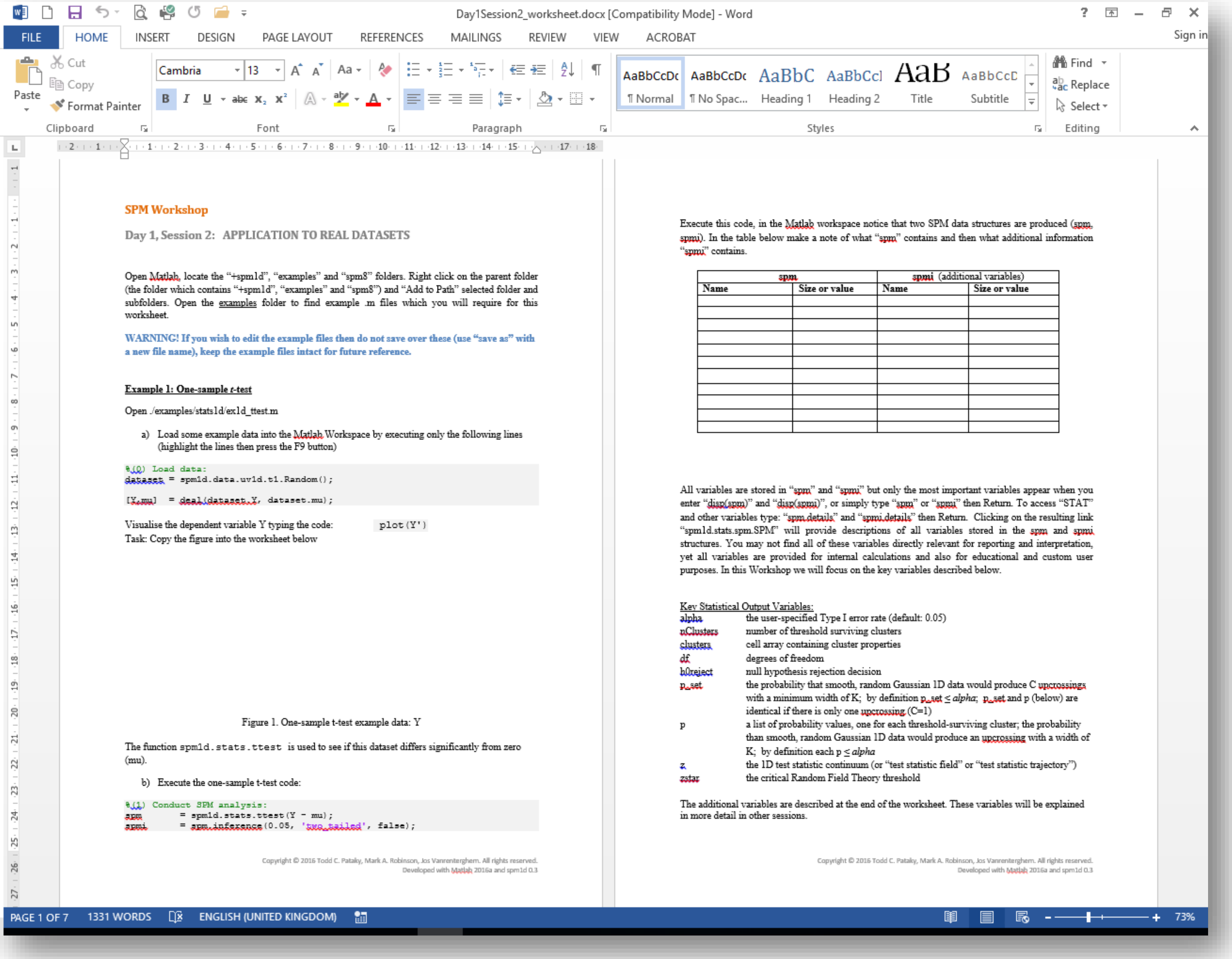
SPM1D
ex1d_ttest2.m

```
%(1) Conduct SPM analysis:
spm  = spm1d.stats.ttest2(YA, YB);
spmi = spm.inference(0.05,'two_tailed',true);
```

yA = 10 x 101 double

yB = 10 x 101 double

Workshops & tutorials



SPM Workshop

Day 1, Session 2: APPLICATION TO REAL DATASETS

Open Matlab, locate the "+spm1d", "examples" and "spm8" folders. Right click on the parent folder (the folder which contains "+spm1d", "examples" and "spm8") and "Add to Path" selected folder and subfolders. Open the examples folder to find example .m files which you will require for this worksheet.

WARNING! If you wish to edit the example files then do not save over these (use "save as" with a new file name), keep the example files intact for future reference.

Example 1: One-sample *t*-test

Open ./examples/stats1d/ex1d_ttest.m

a) Load some example data into the Matlab Workspace by executing only the following lines (highlight the lines then press the F9 button)

```
%(0) Load data:
dataset = spm1d.data.uv1d.t1.Random();

[Y,mu]  = deal(dataset.Y, dataset.mu);
```

Visualise the dependent variable Y typing the code: `plot(Y')`

Task: Copy the figure into the worksheet below

Figure 1. One-sample t-test example data: Y

The function `spm1d.stats.ttest` is used to see if this dataset differs significantly from zero (mu).

b) Execute the one-sample t-test code:

```
%(1) Conduct SPM analysis:
spm       = spm1d.stats.ttest(Y - mu);
spmi      = spm.inference(0.05, 'two_tailed', false);
```

Execute this code, in the Matlab workspace notice that two SPM data structures are produced (spm, spmi). In the table below make a note of what "spm" contains and then what additional information "spmi" contains.

spm		spmi (additional variables)	
Name	**Size or value**	**Name**	**Size or value**

All variables are stored in "spm" and "spmi" but only the most important variables appear when you enter "disp(spm)" and "disp(spmi)", or simply type "spm" or "spmi" then Return. To access "STAT" and other variables type: "spm.details" and "spmi.details" then Return. Clicking on the resulting link "spm1d.stats.spm.SPM" will provide descriptions of all variables stored in the spm and spmi structures. You may not find all of these variables directly relevant for reporting and interpretation, yet all variables are provided for internal calculations and also for educational and custom user purposes. In this Workshop we will focus on the key variables described below.

Key Statistical Output Variables:

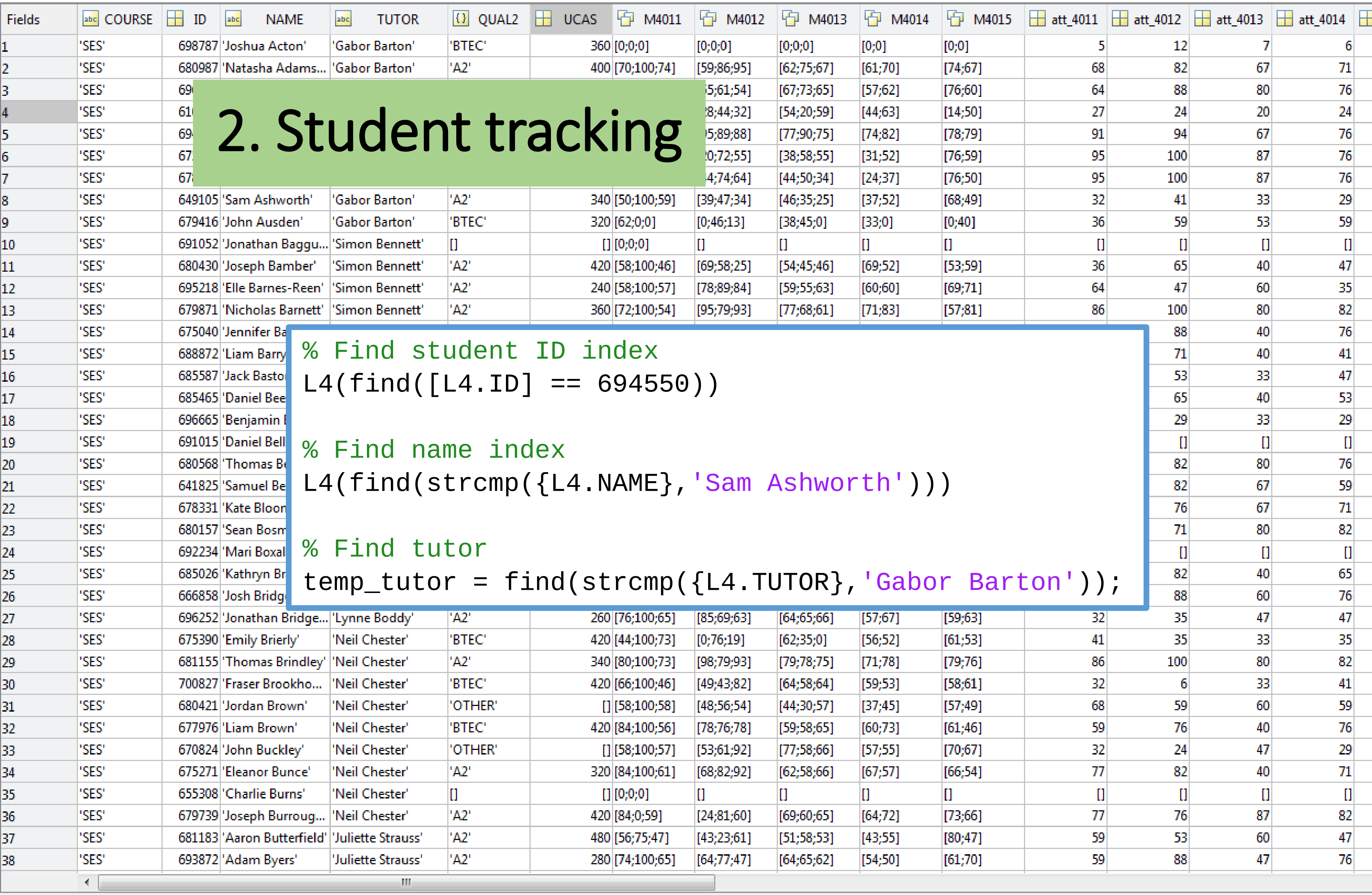
alpha	the user-specified Type I error rate (default: 0.05)
nClusters	number of threshold surviving clusters
clusters	cell array containing cluster properties
df	degrees of freedom
h0reject	null hypothesis rejection decision
p_set	the probability that smooth, random Gaussian 1D data would produce C upcrossings with a minimum width of K; by definition p_set ≤ *alpha*; p_set and p (below) are identical if there is only one upcrossing (C=1)
p	a list of probability values, one for each threshold-surviving cluster; the probability than smooth, random Gaussian 1D data would produce an upcrossing with a width of K; by definition each p ≤ *alpha*
z	the 1D test statistic continuum (or "test statistic field" or "test statistic trajectory")
zstar	the critical Random Field Theory threshold

The additional variables are described at the end of the worksheet. These variables will be explained in more detail in other sessions.

DEMO 1

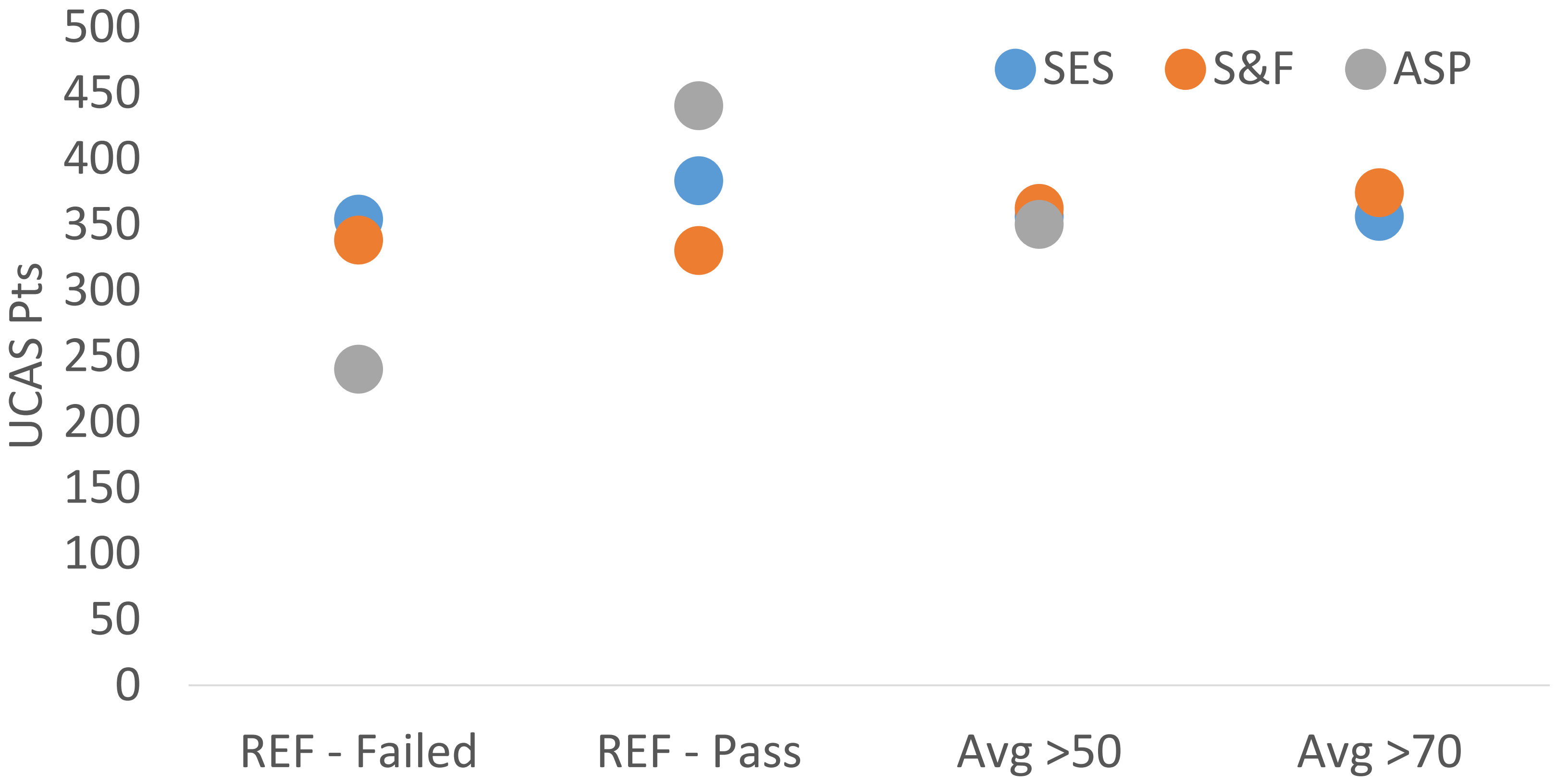
2. Student tracking

```
% Find student ID index
L4(find([L4.ID] == 694550))

% Find name index
L4(find(strcmp({L4.NAME},'Sam Ashworth')))

% Find tutor
temp_tutor = find(strcmp({L4.TUTOR},'Gabor Barton'));
```

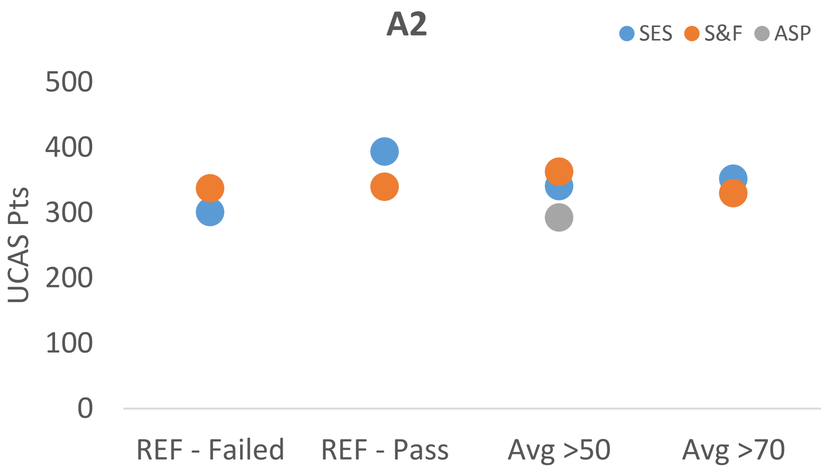


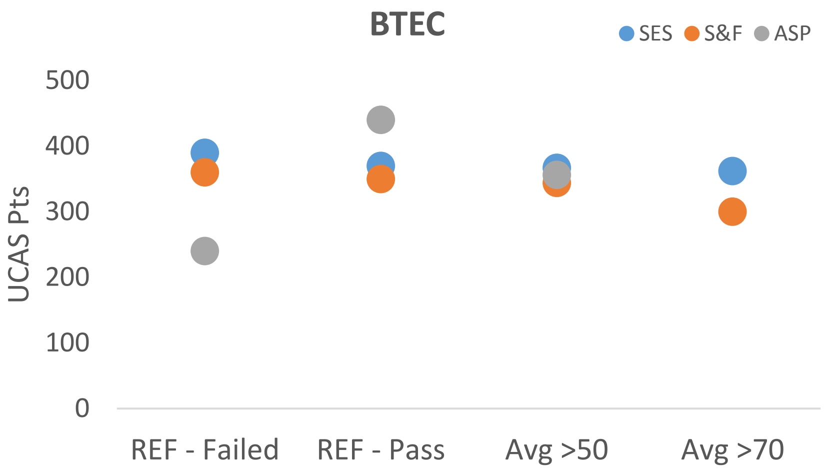
Fields	COURSE	ID	NAME	TUTOR	QUAL2	UCAS	M4011	M4012	M4013	M4014	M4015	att_4011	att_4012	att_4013	att_4014
1	'SES'	698787	'Joshua Acton'	'Gabor Barton'	'BTEC'	360	[0;0;0]	[0;0;0]	[0;0;0]	[0;0]	[0;0]	5	12	7	6
2	'SES'	680987	'Natasha Adams...	'Gabor Barton'	'A2'	400	[70;100;74]	[59;86;95]	[62;75;67]	[61;70]	[74;67]	68	82	67	71
3	'SES'	69						5;61;54]	[67;73;65]	[57;62]	[76;60]	64	88	80	76
4	'SES'	61						8;44;32]	[54;20;59]	[44;63]	[14;50]	27	24	20	24
5	'SES'	69						5;89;88]	[77;90;75]	[74;82]	[78;79]	91	94	67	76
6	'SES'	67						0;72;55]	[38;58;55]	[31;52]	[76;59]	95	100	87	76
7	'SES'	67						4;74;64]	[44;50;34]	[24;37]	[76;50]	95	100	87	76
8	'SES'	649105	'Sam Ashworth'	'Gabor Barton'	'A2'	340	[50;100;59]	[39;47;34]	[46;35;25]	[37;52]	[68;49]	32	41	33	29
9	'SES'	679416	'John Ausden'	'Gabor Barton'	'BTEC'	320	[62;0;0]	[0;46;13]	[38;45;0]	[33;0]	[0;40]	36	59	53	59
10	'SES'	691052	'Jonathan Baggu...	'Simon Bennett'	[]	[]	[0;0;0]	[]	[]	[]	[]	[]	[]	[]	[]
11	'SES'	680430	'Joseph Bamber'	'Simon Bennett'	'A2'	420	[58;100;46]	[69;58;25]	[54;45;46]	[69;52]	[53;59]	36	65	40	47
12	'SES'	695218	'Elle Barnes-Reen'	'Simon Bennett'	'A2'	240	[58;100;57]	[78;89;84]	[59;55;63]	[60;60]	[69;71]	64	47	60	35
13	'SES'	679871	'Nicholas Barnett'	'Simon Bennett'	'A2'	360	[72;100;54]	[95;79;93]	[77;68;61]	[71;83]	[57;81]	86	100	80	82
14	'SES'	675040	'Jennifer Ba										88	40	76
15	'SES'	688872	'Liam Barry										71	40	41
16	'SES'	685587	'Jack Basto										53	33	47
17	'SES'	685465	'Daniel Bee										65	40	53
18	'SES'	696665	'Benjamin										29	33	29
19	'SES'	691015	'Daniel Bell										[]	[]	[]
20	'SES'	680568	'Thomas Be										82	80	76
21	'SES'	641825	'Samuel Be										82	67	59
22	'SES'	678331	'Kate Bloon										76	67	71
23	'SES'	680157	'Sean Bosm										71	80	82
24	'SES'	692234	'Mari Boxal										[]	[]	[]
25	'SES'	685026	'Kathryn Br										82	40	65
26	'SES'	666858	'Josh Bridg										88	60	76
27	'SES'	696252	'Jonathan Bridge...	'Lynne Boddy'	'A2'	260	[76;100;65]	[85;69;63]	[64;65;66]	[57;67]	[59;63]	32	35	47	47
28	'SES'	675390	'Emily Brierly'	'Neil Chester'	'BTEC'	420	[44;100;73]	[0;76;19]	[62;35;0]	[56;52]	[61;53]	41	35	33	35
29	'SES'	681155	'Thomas Brindley'	'Neil Chester'	'A2'	340	[80;100;73]	[98;79;93]	[79;78;75]	[71;78]	[79;76]	86	100	80	82
30	'SES'	700827	'Fraser Brookho...	'Neil Chester'	'BTEC'	420	[66;100;46]	[49;43;82]	[64;58;64]	[59;53]	[58;61]	32	6	33	41
31	'SES'	680421	'Jordan Brown'	'Neil Chester'	'OTHER'	[]	[58;100;58]	[48;56;54]	[44;30;57]	[37;45]	[57;49]	68	59	60	59
32	'SES'	677976	'Liam Brown'	'Neil Chester'	'BTEC'	420	[84;100;56]	[78;76;78]	[59;58;65]	[60;73]	[61;46]	59	76	40	76
33	'SES'	670824	'John Buckley'	'Neil Chester'	'OTHER'	[]	[58;100;57]	[53;61;92]	[77;58;66]	[57;55]	[70;67]	32	24	47	29
34	'SES'	675271	'Eleanor Bunce'	'Neil Chester'	'A2'	320	[84;100;61]	[68;82;92]	[62;58;66]	[67;57]	[66;54]	77	82	40	71
35	'SES'	655308	'Charlie Burns'	'Neil Chester'	[]	[]	[0;0;0]	[]	[]	[]	[]	[]	[]	[]	[]
36	'SES'	679739	'Joseph Burroug...	'Neil Chester'	'A2'	420	[84;0;59]	[24;81;60]	[69;60;65]	[64;72]	[73;66]	77	76	87	82
37	'SES'	681183	'Aaron Butterfield'	'Juliette Strauss'	'A2'	480	[56;75;47]	[43;23;61]	[51;58;53]	[43;55]	[80;47]	59	53	60	47
38	'SES'	693872	'Adam Byers'	'Juliette Strauss'	'A2'	280	[74;100;65]	[64;77;47]	[64;65;62]	[54;50]	[61;70]	59	88	47	76

UCAS pts vs Progress

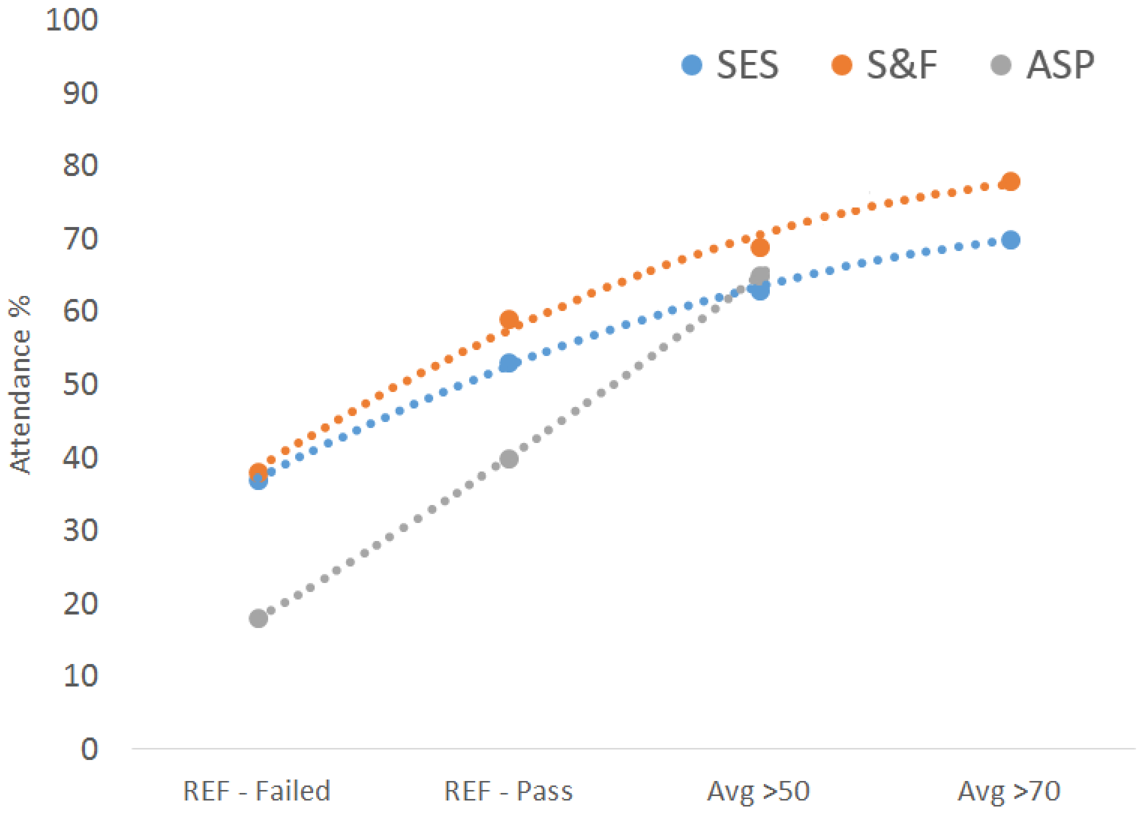


UCAS pts vs Progress



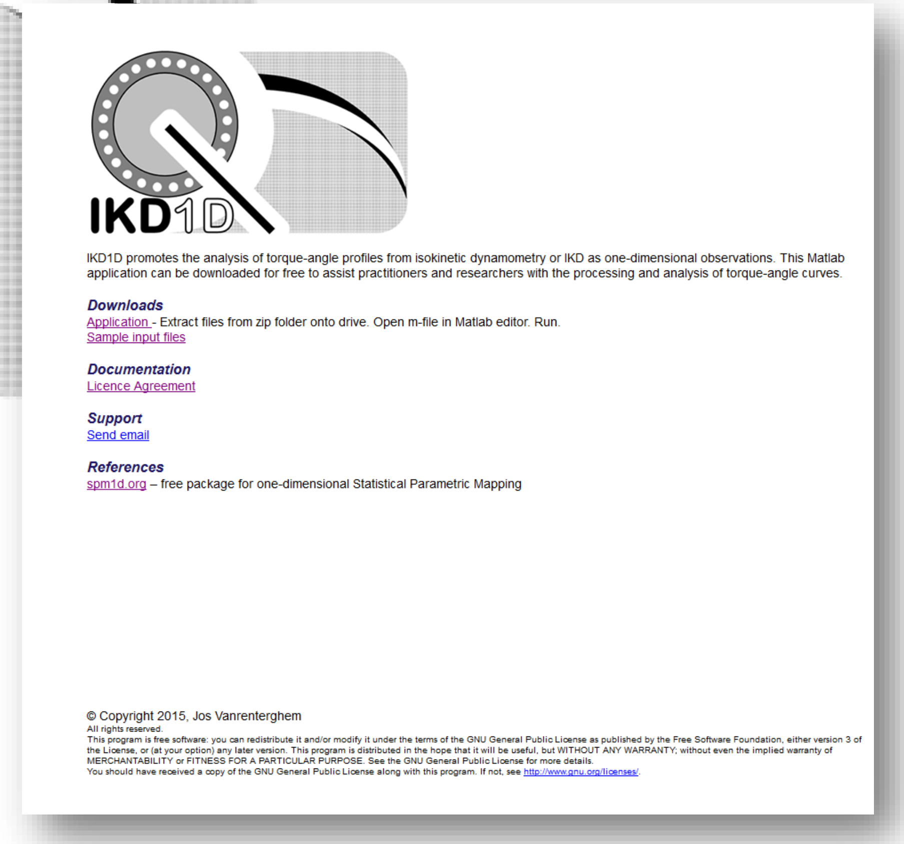


Attendance vs Progress



DEMO 2

1. Graphical User Interface – IKD1D



IKD1D promotes the analysis of torque-angle profiles from isokinetic dynamometry or IKD as one-dimensional observations. This Matlab application can be downloaded for free to assist practitioners and researchers with the processing and analysis of torque-angle curves.

Downloads

Application - Extract files from zip folder onto drive. Open m-file in Matlab editor. Run.

Sample input files

Documentation

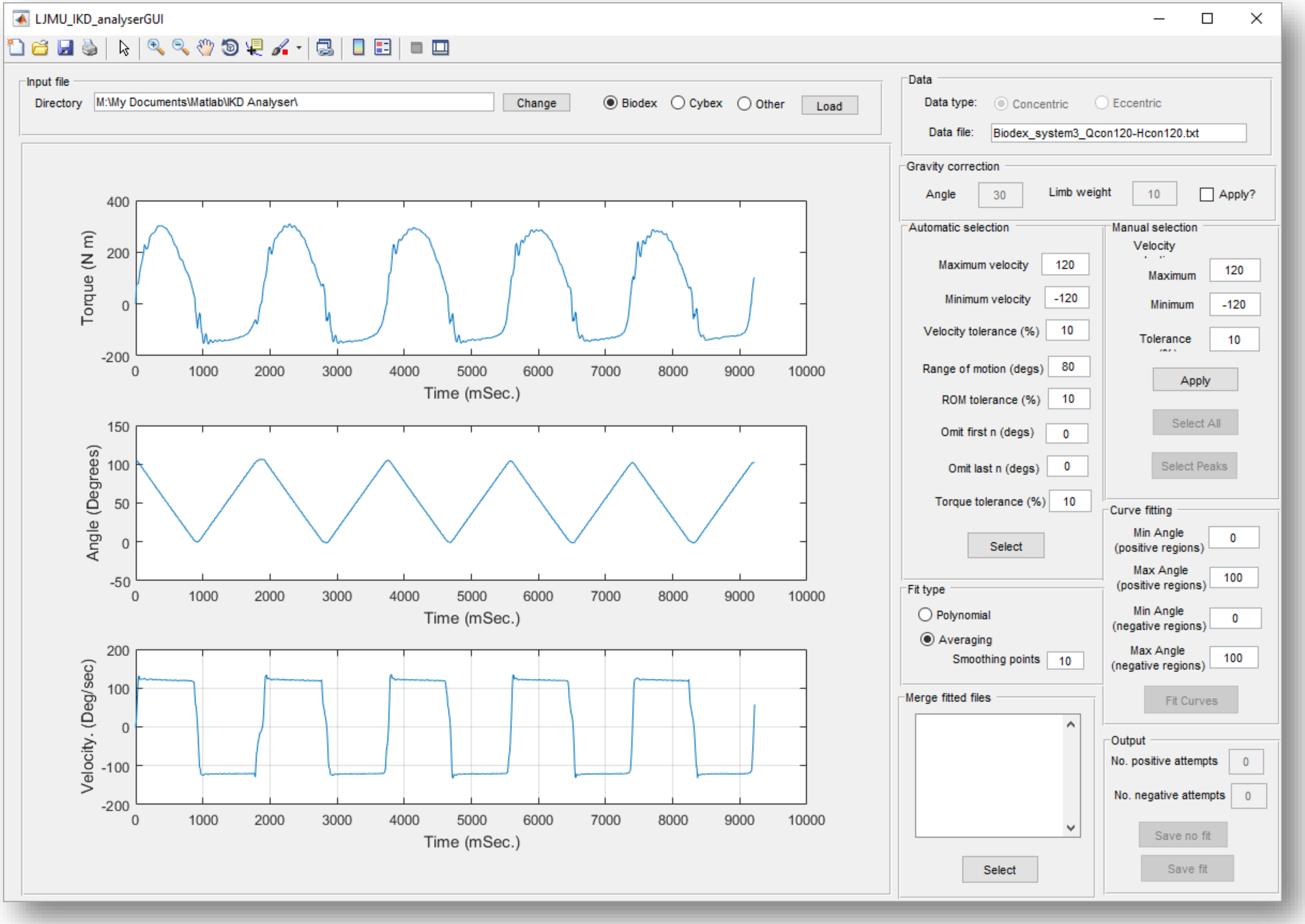
Licence Agreement

Support

Send email

References

spm1d.org – free package for one-dimensional Statistical Parametric Mapping



LJMU_IKD_analyserGUI

Input file

Directory: M:\My Documents\Matlab\IKD Analyser\

● Biodex ○ Cybex ○ Other

Data

Data type: ○ Concentric ○ Eccentric

Data file: Biodex_system3_Qcon120-Hcon120.txt

Gravity correction

Angle: 30

Limb weight: 10

□ Apply?

Automatic selection

Maximum velocity: 120

Minimum velocity: -120

Velocity tolerance (%): 10

Range of motion (degs): 80

ROM tolerance (%): 10

Omit first n (degs): 0

Omit last n (degs): 0

Torque tolerance (%): 10

Manual selection

Velocity

Maximum: 120

Minimum: -120

Tolerance: 10

Fit type

○ Polynomial

● Averaging

Smoothing points: 10

Curve fitting

Min Angle (positive regions): 0

Max Angle (positive regions): 100

Min Angle (negative regions): 0

Max Angle (negative regions): 100

Merge fitted files

Output

No. positive attempts: 0

No. negative attempts: 0

DEMO 3

MATLAB

Three quick demos

Available on the app player and to download for home use for teaching and research

m.a.robinson@ljmu.ac.uk
Mark Robinson, Snr Lecturer in Biomechanics