Second-level fNIRS analysis with covariates

13rd July 2020

Jessica Gemignani, *PhD*
Integrative Neuroscience and Cognition Center
Université de Paris & CNRS

Outline

- Introduction
  - Why to use covariates?
  - Examples in literature
- How to include covariates in the analysis
  - ANOVA vs Mixed Effects Model
- Walkthrough with the Brain AnalyZIR toolbox
- Conclusions

An experiment is designed to test the effects of some intervention on one or more measures → **outcome/ response / dependent variables** (typically, HbO/HbR activation)

With the statistical analysis, we aim at explaining the DV in terms of **predictors/ independent variables / explanatory variables**

Explanatory variable
= any variable that is thought to possibly affect the outcome

purposely manipulated in the experiment - e.g.: Task

NOT purposely manipulated - e.g.: Age, Gender, IQ, ability to perform certain task (score)…

Covariates
(can be of interest or not)

Examples in literature (1)

Stojanovic-Radic et al 2014:

- Main goal is to determine differences in the neural activation of the orbitofrontal brain region between individuals with multiple sclerosis and healthy individuals
- Task: *n-back*

	MS	HC
N	13	12
% female	92 %	67 %
Age Mean (SD)	45.8 (8.8)	31 (9.6)

Participants info (Stojanovic-Radic et al 2014)

- Only the condition is of interest (0-back vs 1-back vs 2-back in the two group)
- But age is also entered in the model because individuals in one group are much older than the other group

Examples in literature (2)

Vassena et al 2019

- Main goal is to compare activation of the dorsolateral prefrontal cortex (DLPFC) when individuals are mentally preparing for a hard task vs easy task
- Task: mental arithmetic (difficult/easy) + several questionnaires
- Statistics: several covariates (reaction time, performances in executing the task , the difficulty of the task…)
- In this case, the effect of the covariates on the activation was of interest

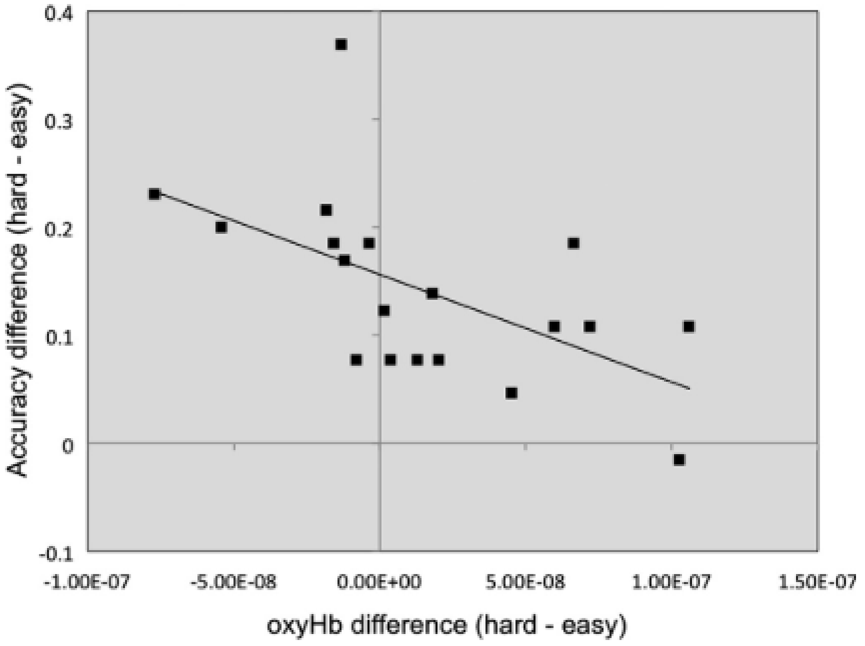


Fig. 4. Correlation between difficulty anticipation in DLPFC and performance accuracy. The x axis shows average differences in oxyHb peak in the hard – easy condition per each participant. The y axis shows differences in performance accuracy (hard – easy). Participants with larger DLPFC difficulty effect showed a smaller difference in accuracy for hard vs. easy trials.

From Vassena et al 2019

Examples in literature (3)

Gemignani et al 2018:

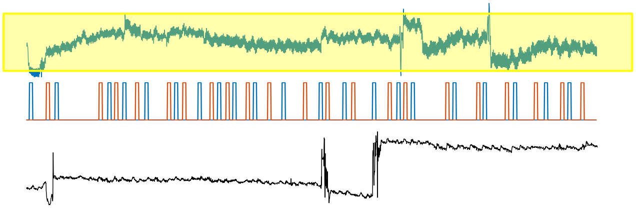
- Main goal was to compare classification accuracies achieved by three different algorithms
- Input to the group-level: subject-level classification accuracies
- Additional covariates: Hair color, time of measurement and gender
- In one case, classification accuracy was lower in brown-haired subjects

	LDA: HbO + HbR		GLM: HbO		GLM: HbR	
	β	p-value	β	p-value	β	p-value
Age	0.0042	0.1762	0.0010	0.8734	0.0068	0.1120
Hair color	0.0361	0.1875	−0.0936	0.1020	*−0.1408*	*0.0052*
Gender	0.0038	0.1476	−0.0077	0.1552	−0.0026	0.4904
Time of measurement	−0.0187	0.5109	0.0573	0.3308	−0.0072	0.7540

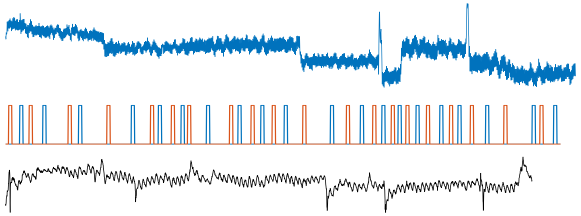
Results table from Gemignani et al 2018

This applies at first and second level

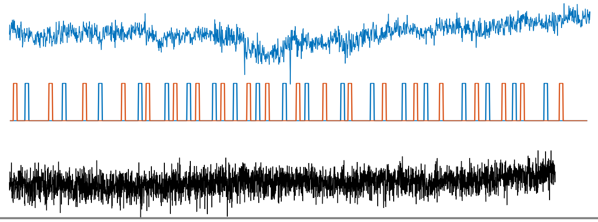
Subject 1



Subject 2

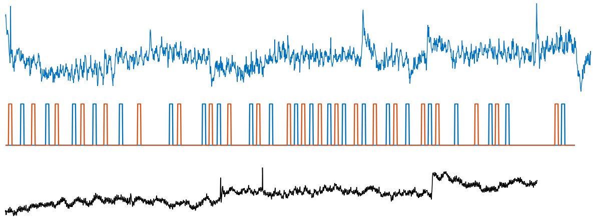


Subject 3



[....]

Subject N

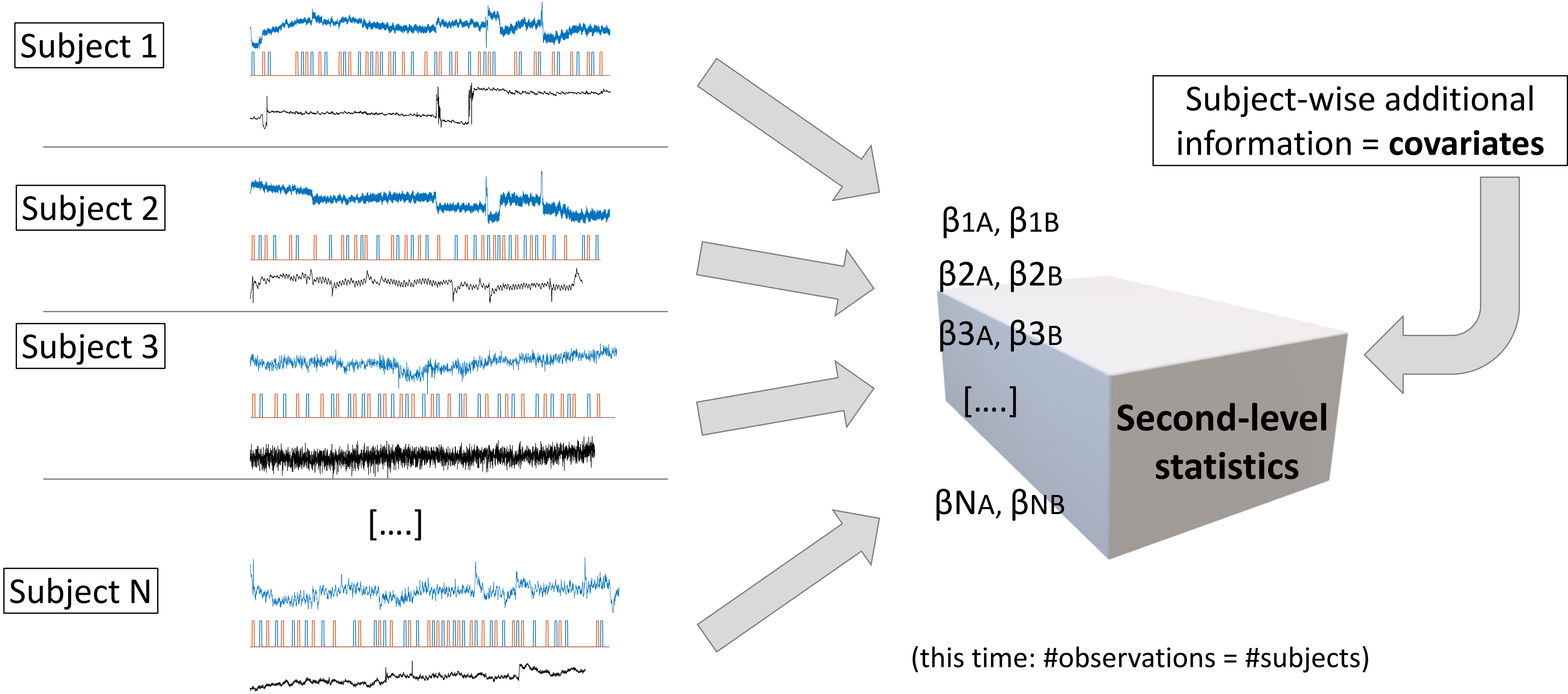


First-level statistics:

→ β for each condition
→ β for each additional covariate

The covariate must have as many observations as the timeseries to be modeled (at first level: #observations = #time samples)

This applies at first and second level



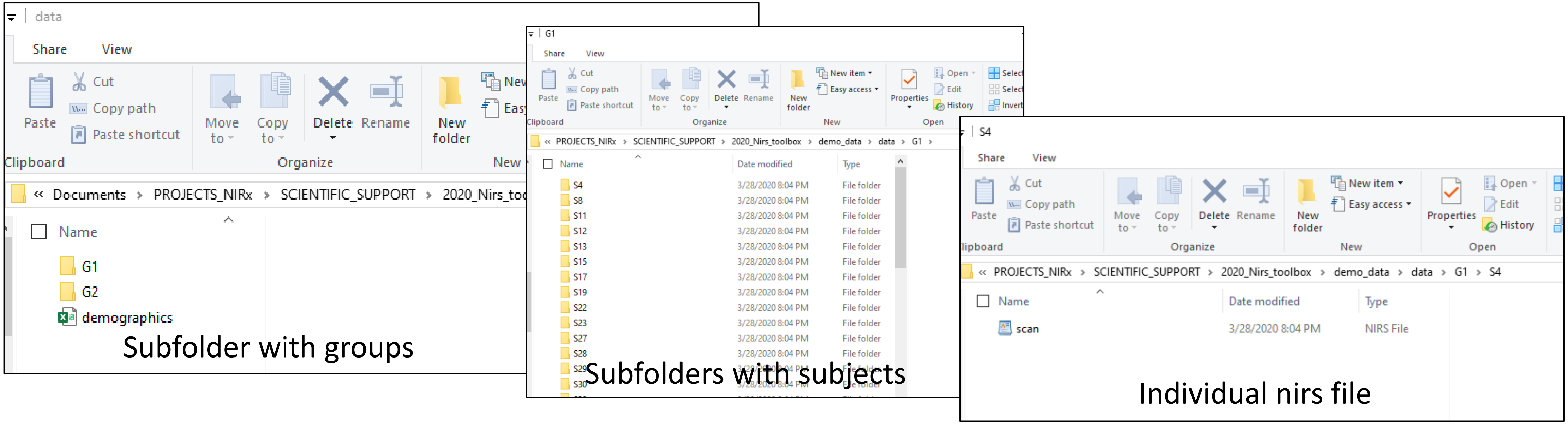
(this time: #observations = #subjects)

Group-level analysis with covariate

- ANOVA with covariates: ANCOVA *(Analysis of Covariance)*
- ANCOVA vs Mixed-Effects Model:
  - Ideally, they should yield the same answer
  - But Mixed-Effects Models are to be preferred:
    - Better account of imbalance in the design (different number of participants in the groups, different number of tasks within each participant, missing data as a result of discarding bad quality channels)
    - Possibility of including **random effects,** accounting for variability in outcome across participants

Walkthrough with Brain AnalyzIR toolbox

- Download at https://github.com/huppertt/nirs-toolbox/ , unzip and add to the matlab path
- Many webinars available in the NIRx Help Center
- Structure of the folder containing the data:



Subfolder with groups

Subfolders with subjects

Individual nirs file

```
%% 2020, July 13rd - Webinar on Statistics with covariates
%% Jessica Gemignani

% This demo is largely based on the demo scripts available in the nirs
% toolbox folder, have a look!

dataFolder= 'C:\Users\Jessica\Documents\PROJECTS_NIRx\202007_StatisticsWebinar\demo_data';
```

```
%% load data
% this function loads a whole directory of .nirs files. The second argument
% tells the function to use the first level of folder names to specify
% group id and to use the second for subject id.
raw = nirs.io.loadDirectory(dataFolder, {'group', 'subject'});

% View the demographics information:
demographics = nirs.createDemographicsTable(raw);
```

← The second option specifies the hierarchy of the root folder

```
>> raw

raw =

  68×1 Data array with properties:

    description
    data
    probe
    time
    Fm
    auxillary
    stimulus
    demographics
    Fs
```

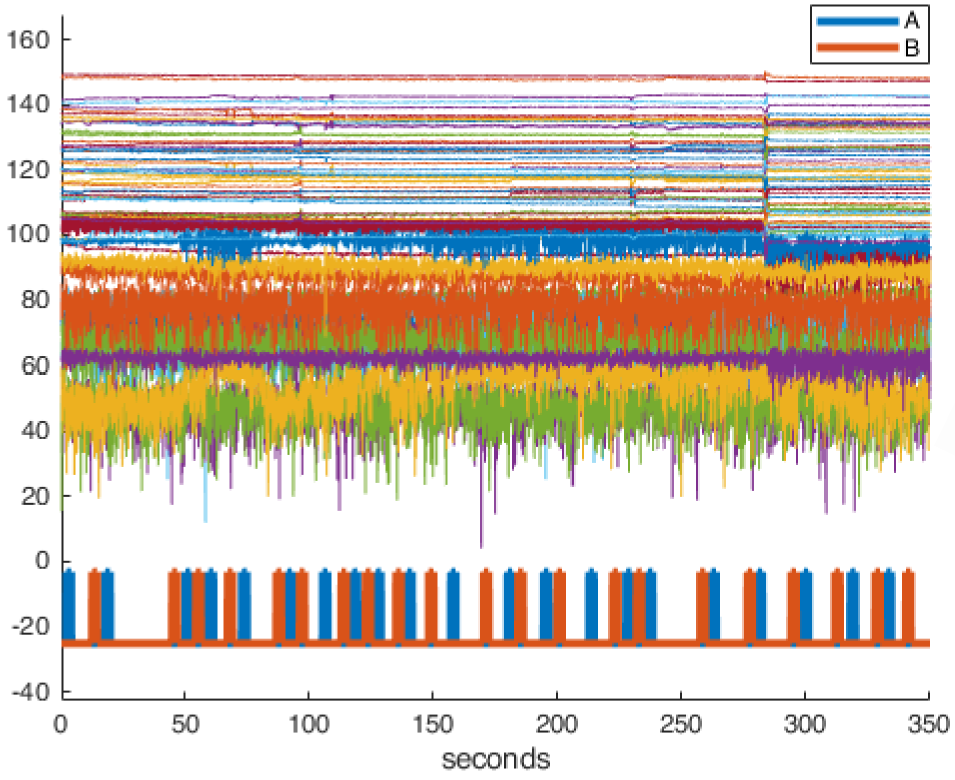
```
>> demographics

demographics =

  68×2 table

    group    subject
    _____    _______

    'G1'     'S11'
    'G1'     'S12'
    'G1'     'S13'
    'G1'     'S15'
    'G1'     'S17'
    'G1'     'S19'
    'G1'     'S22'
```

34 subjects per group

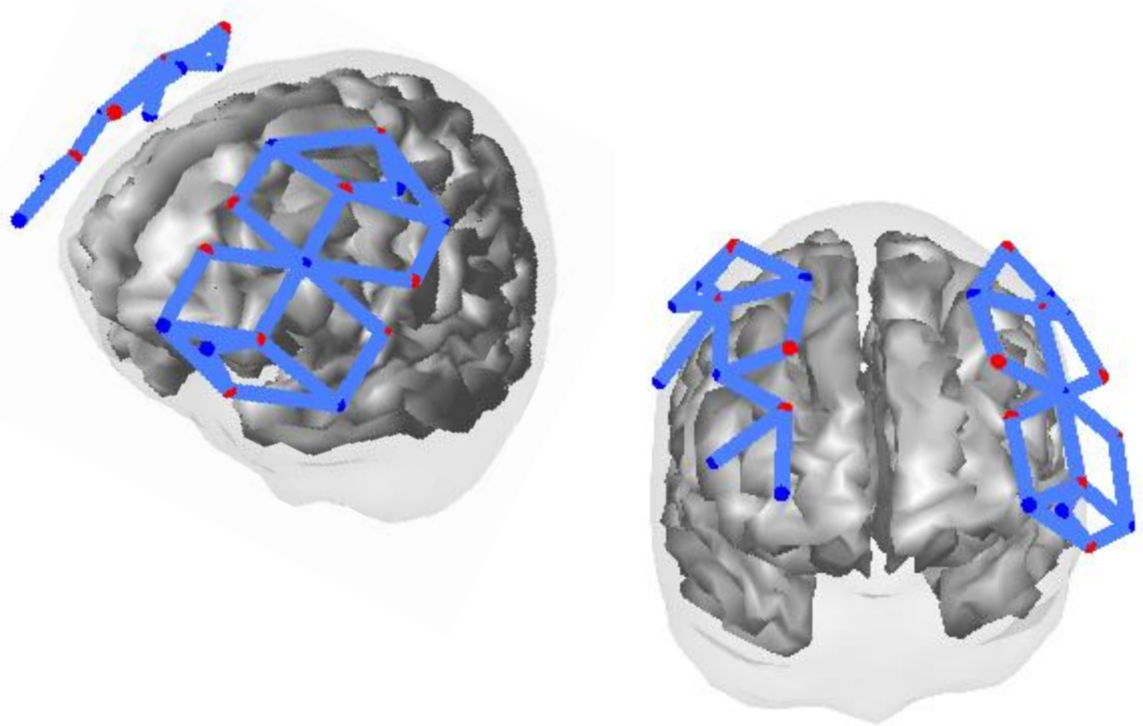
Experimental design

One condition, two different levels A and B

raw(1).draw



raw(1).probe.defaultdrawfcn='3D mesh';
raw(1).probe.draw;



Two ways to add further demographic information

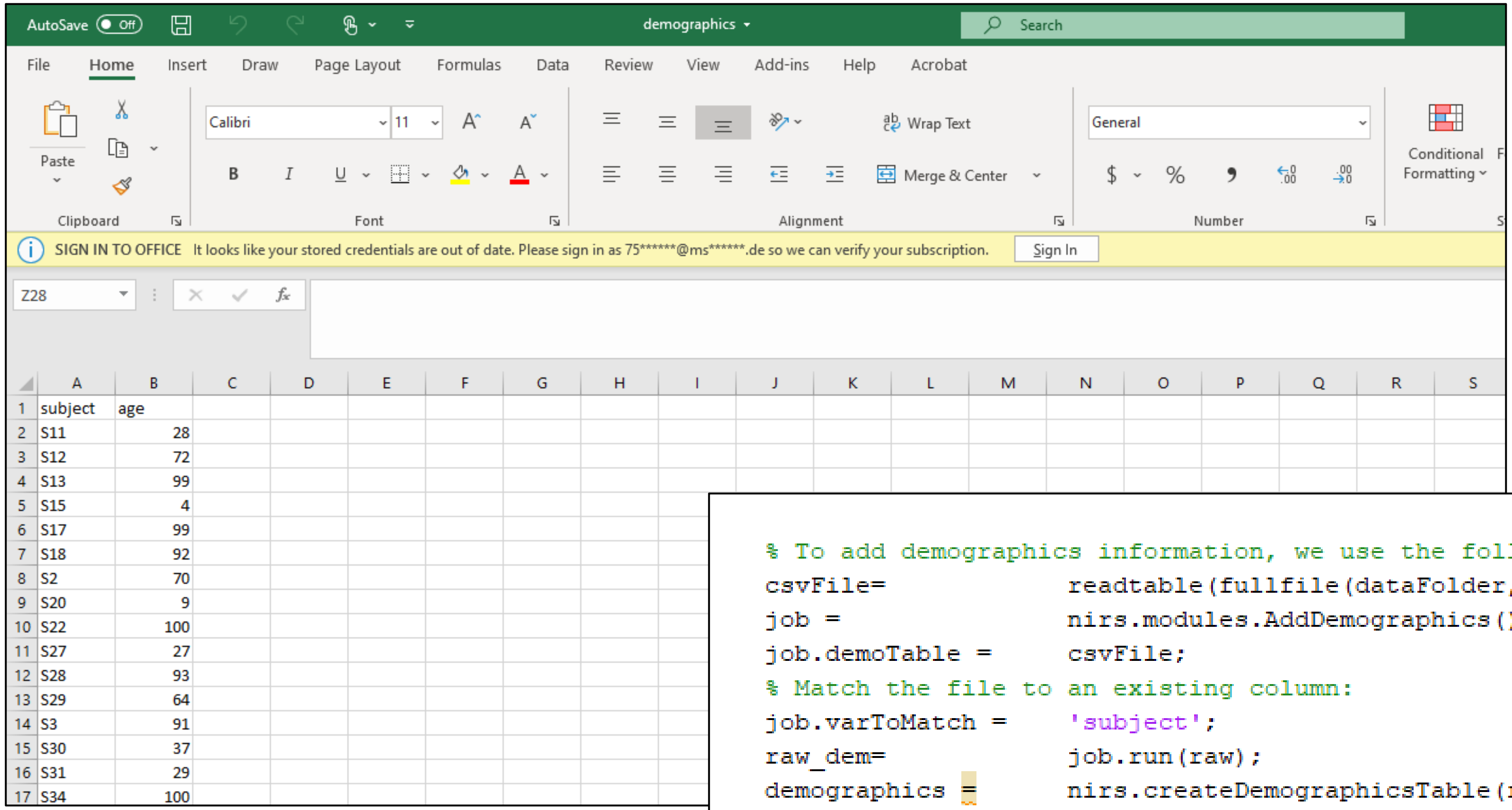
1 Hard-code it in the script:

```
nsubjects= numel(raw);

% Simulate a vector with age ranging from 4 to 10 years and append to
% demographics
age_min=4;
age_max=10;

age =age_min+rand(1,nsubjects)*(age_max-age_min);

for idx=1:nsubjects
    raw(idx).demographics('age')=age(idx);
end
```

```
demographics =

  68×3 table

    group    subject     age
    _____    _______    ______

    'G1'      'S11'     7.3416
    'G1'      'S12'     5.6436
    'G1'      'S13'     4.7923
    'G1'      'S15'     8.1979
    'G1'      'S17'     6.9154
    'G1'      'S19'     5.0963
    'G1'      'S22'     4.6073
```

Two ways to add further demographic information

2 Use an external csv file

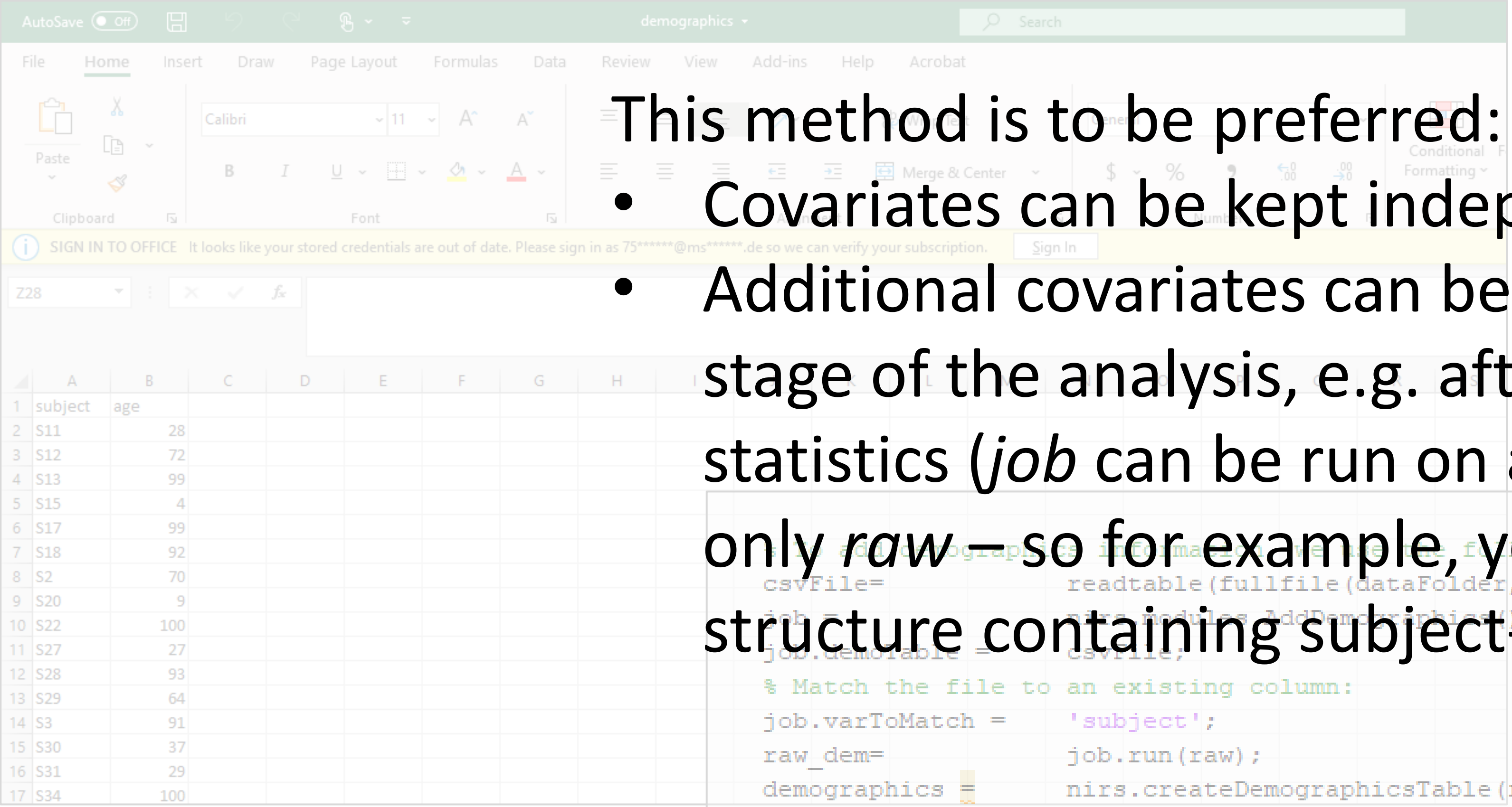


subject	age
S11	28
S12	72
S13	99
S15	4
S17	99
S18	92
S2	70
S20	9
S22	100
S27	27
S28	93
S29	64
S3	91
S30	37
S31	29
S34	100

```
% To add demographics information, we use the following job:
csvFile=            readtable(fullfile(dataFolder, 'demographics.csv'));
job =               nirs.modules.AddDemographics();
job.demoTable =     csvFile;
% Match the file to an existing column:
job.varToMatch =    'subject';
raw_dem=            job.run(raw);
demographics =      nirs.createDemographicsTable(raw_dem)
```

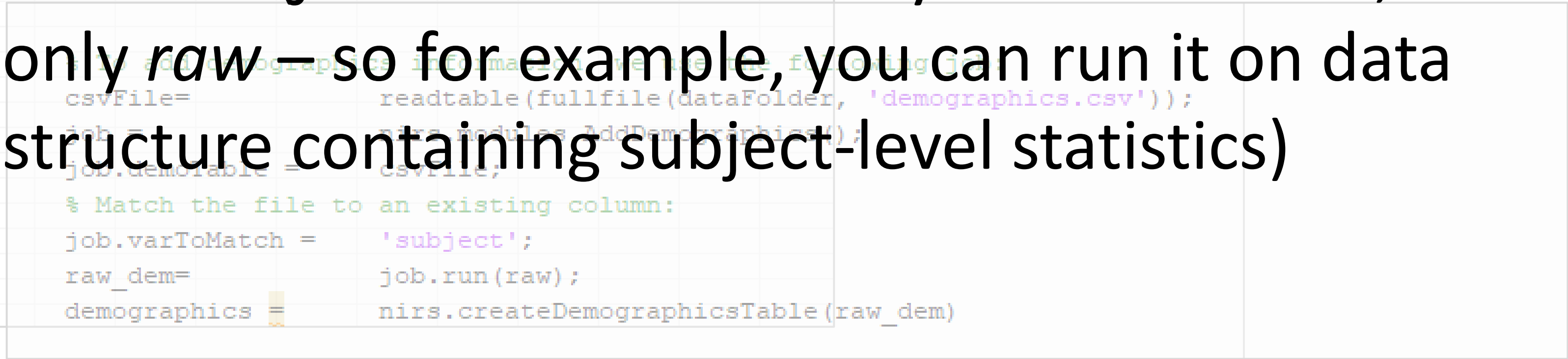
Two ways to add further demographic information

2 Use an external csv file



This method is to be preferred:

- Covariates can be kept independent of the script
- Additional covariates can be added also at a later stage of the analysis, e.g. after subject-level statistics (*job* can be run on any data structure, not only *raw* – so for example, you can run it on data structure containing subject-level statistics)



Pre-processing and first-level stats

```
%% Set up basic pre-processing and first-level statistics
job = nirs.modules.default_modules.single_subject;

% run the analysis (Hb and SubjStats are automatically saved to the
% workspace, it's defined within the pipeline)
job.run(raw)
```

Customize your pipeline and carefully check the default options

```
function jobs = single_subject
jobs=nirs.modules.ImportData();
jobs.Input='raw';                          % --> specify name of the workspace variable that undergoes this pipeline
jobs=nirs.modules.RemoveStimless(jobs); % --> remove files without stimuli (delete this if you're doing resting state)

jobs = nirs.modules.FixNaNs(jobs);        % -->  Attempts to fix NaN values by interpolation.
jobs = nirs.modules.Resample(jobs);       % --> resamples to lower Fs (saves computing time)
jobs.Fs = 5; % resample to 5 Hz

jobs = nirs.modules.OpticalDensity( jobs );
jobs = nirs.modules.BeerLambertLaw( jobs );
jobs.PPF= 6;                               % --> Default DPF is set to 5/50, change if needed
jobs = nirs.modules.ExportData(jobs);
jobs.Output='Hb';                          % after Beer-Lambert, export variable and call it 'Hb'
jobs = nirs.modules.TrimBaseline( jobs );
jobs.preBaseline   = 30;                   % Keep 30 seconds before start of the block and 30 seconds after the end
jobs.postBaseline  = 30;
jobs = nirs.modules.GLM(jobs );            % Subject-level GLM (important: check defaults, e.g. basis function, peak time, algorithm...)
jobs = nirs.modules.ExportData(jobs);
jobs.Output='SubjStats';
```

Pre-processing and first-level stats

```
>> SubjStats(1)

ans =

  ChannelStats with properties:

     description: 'C:\Users\Jessica\Documents\PROJECTS_NIRx\202007_StatisticsWebinar\demo_data\data\G1\S11\scan.nirs'
       variables: [140×4 table]
            beta: [140×1 double]
            covb: [140×140 double]
             dfe: 1.6103e+03
            tail: 'two-sided'
           probe: [1×1 nirs.core.Probe]
    demographics: [1×1 Dictionary]
           basis: [1×1 struct]
      conditions: {2×1 cell}
           tstat: [140×1 double]
               p: [140×1 double]
               q: [140×1 double]
```

```
>> SubjStats(1).variables

ans =

  140×4 table

    source    detector    type     cond
    ______    ________    _____    ____

      1          1        'hbo'    'A'
      1          1        'hbr'    'A'
      1          2        'hbo'    'A'
      1          2        'hbr'    'A'
      1          6        'hbo'    'A'
      1          6        'hbr'    'A'
      2          1        'hbo'    'A'
      2          1        'hbr'    'A'
      2          2        'hbo'    'A'
```

GLM is *mass-univariate* analysis

one coefficient (β) for each regressor (two levels of condition), for each channel and for each component (HbO, HbR):

35 channels → 140 coefficients

Pre-processing and first-level stats

```
>> SubjStats(1)

ans =

  ChannelStats with properties:

     description: 'C:\Users\Jessica\Documents\PROJECTS_NIRx\202007_StatisticsWebinar\demo_data\data\G1\S11\scan.nirs'
       variables: [140×4 table]
            beta: [140×1 double]
            covb: [140×140 double]
             dfe: 1.6103e+03
            tail: 'two-sided'
           probe: [1×1 nirs.core.Probe]
    demographics: [1×1 Dictionary]
           basis: [1×1 struct]
      conditions: {2×1 cell}
           tstat: [140×1 double]
               p: [140×1 double]
               q: [140×1 double]
```

```
>> SubjStats(1).p

ans =

    0.0000
    0.0943
    0.4037
    0.2920
    0.0032
    0.4734
    0.6755
    0.2824
    0.4915
    0.7015
    0.6673
    0.3705
    0.0317
    0.7988
    0.0000
    0.5150
    0.1150
```

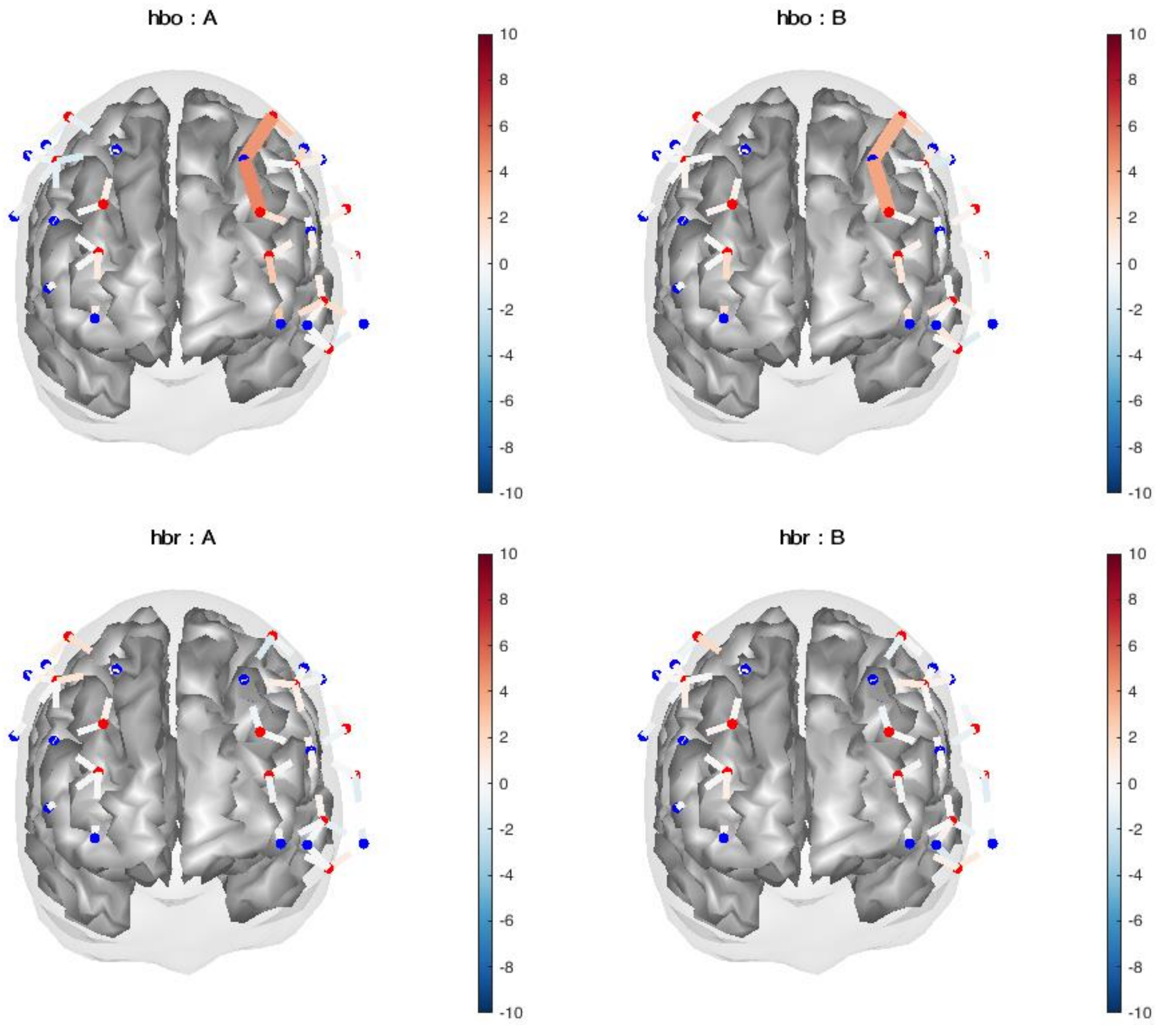
Benjamini-Hochberg (1995) procedure for multiple comparisons correction (*FDR correction)*

```
>> SubjStats(1).q

ans =

    0.0007
    0.9596
    0.9918
    0.9918
    0.0914
    0.9918
    1.0000
    0.9918
    0.9918
    1.0000
    1.0000
    0.9918
    0.5726
    1.0000
    0.0002
    0.9918
    0.9767
```

First-level stats

SubjStats(1).draw('tstat', [-10 10], 'q < 0.05')

- t values (altern: beta)
- In the range [-10 10]
- Only for p_{FDR}<0.05



The Mixed-Effects module

nirs.modules.MixedEffects()

- Can use any info already stored in the SubjStats variable to build the model (both categorical and continuous)
- Model must be defined in Wilkinson-Rogers notation
- Performs mixed-effects analysis (fixed + random factors), based on the matlab built-in function *fitlme* (https://www.mathworks.com/help/stats/fitlme.html):

Response variable (βs from first-level)

$$\beta = \boxed{L \cdot F} + \boxed{Z \cdot R} + \epsilon$$

FIXED FACTORS RANDOM FACTORS

Model for a single channel

The Mixed-Effects module

nirs.modules.MixedEffects()

- In the toolbox, it is implemented as to model all channels simultaneously

$$\beta = \boxed{L \otimes Isr_{cdet}} + \boxed{Z \otimes Isr_{cdet}} + \epsilon$$

FIXED FACTORS FOR ALL CHANNELS

RANDOM FACTORS FOR ALL CHANNELS

- FDR-corrected *p* values
- Has options for robust fit (to downweigh the outliers) , to center the variables (especially when they have a very different scale) and to apply pre-whitening also at second-level (for formulation and details: Santosa et al 2018)

The Mixed-Effects module

```
classdef MixedEffects < nirs.modules.AbstractModule
    %% MixedEffect - Performs group level mixed effects analysis.
    %
    % Options:
    %     formula     - string specifiying regression formula (see Wilkinson notation)
    %     dummyCoding - dummyCoding format for categorical variables (full, reference, effects)
    %     centerVars  - (true or false) flag for whether or not to center numerical variables
    %
    % Example Formula:
    %     % this will calculate the group average for each condition
    %     j = nirs.modules.MixedEffects();
    %     j.formula = 'beta ~ -1 + group:cond + (1|subject)';
    %     j.dummyCoding = 'full';

    properties
        formula = 'beta ~ -1 + group:cond + (1|subject)';
        dummyCoding = 'full';
        centerVars = true;
        include_diagnostics=false;
        robust=false;
        weighted=true;
        verbose=false;
    end
```

Included in: nirs.modules.Anova() → performs Mixed-Effects + ANOVA in the same run

Formulation

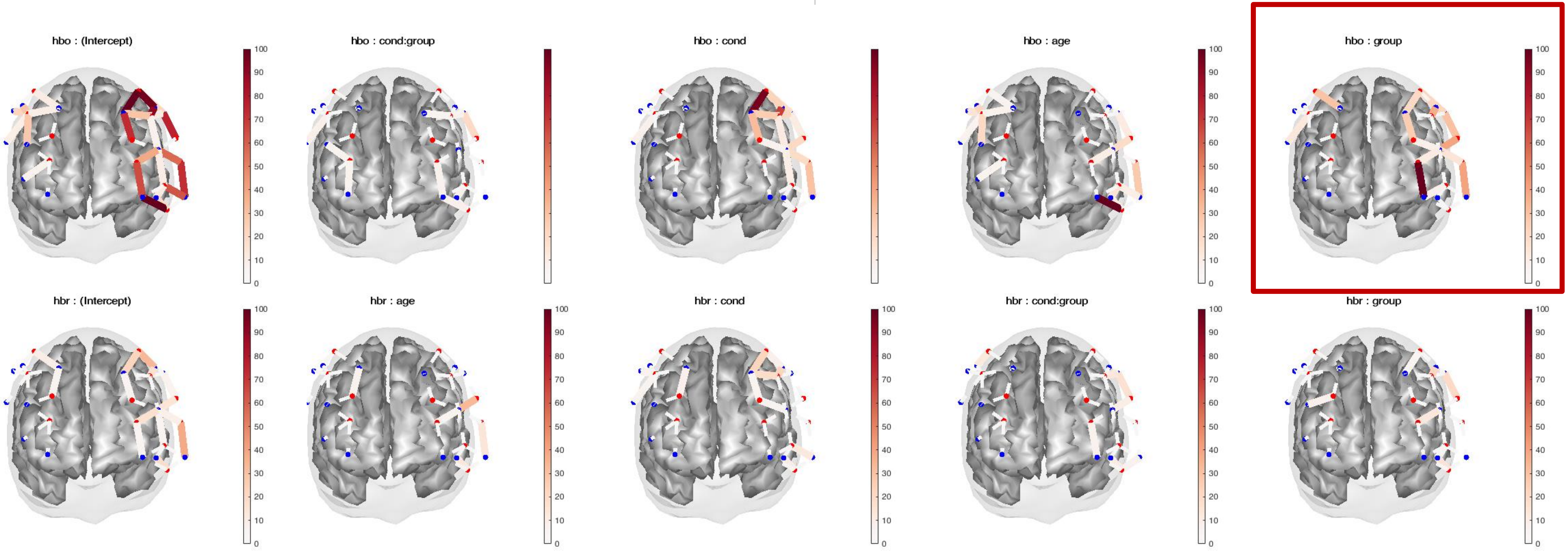
```
%% Group-level analysis (regressors: group and condition, random intercept: subject)
% Step 1) : LMM + ANOVA to get main effects and interaction

job= nirs.modules.Anova();

job.formula=           'beta ~ group*cond + age + (1|subject)';
job.dummyCoding=       'effects';

GroupStats_ANOVA = job.run(SubjStats);
```

350×9 table

source	detector	type	cond	F	df1	df2	p	q
1	1	'hbo'	'(Intercept)'	155.26	1	9170	2.3738e-35	2.4524e-33
1	1	'hbo'	'age'	2.2655	1	9170	0.13232	0.16274
1	1	'hbo'	'cond'	95.986	1	9170	1.4923e-22	5.1391e-21
1	1	'hbo'	'cond:group'	1.4066	1	9170	0.23565	0.23985
1	1	'hbo'	'group'	20.676	1	9170	5.5104e-06	2.6478e-05
1	1	'hbr'	'(Intercept)'	22.559	1	9170	2.0686e-06	1.1873e-05
1	1	'hbr'	'age'	1.2737	1	9170	0.25911	0.25694
1	1	'hbr'	'cond'	22.635	1	9170	1.989e-06	1.1873e-05
1	1	'hbr'	'cond:group'	0.0172	1	9170	0.89566	0.55778
1	1	'hbr'	'group'	5.4164	1	9170	0.019971	0.035572
1	2	'hbo'	'(Intercept)'	21.754	1	9170	3.1433e-06	1.6237e-05
1	2	'hbo'	'age'	0.1338	1	9170	0.71454	0.49213
1	2	'hbo'	'cond'	17.549	1	9170	2.8263e-05	0.0001123
1	2	'hbo'	'cond:group'	0.01326	1	9170	0.90833	0.56023

Formulation

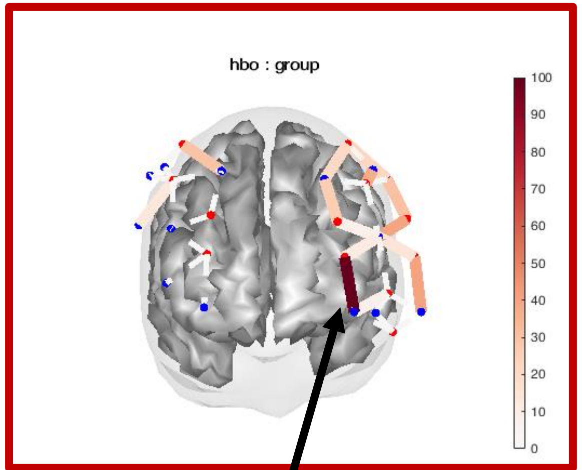
```
% ######################################################
% see results in plots
figure;
GroupStats_ANOVA.probe.defaultdrawfcn='3D mesh (frontal)';
GroupStats_ANOVA.draw(100, 'q<0.05')
```



Formulation

```
% #####################################################
% See results in table
idx= find(GroupStats_ANOVA.q < 0.05);
T=  GroupStats_ANOVA.table; T= T(idx, :);
groupHbO = find(strcmpi(T.cond,'group') & strcmpi(T.type,'hbo'));
T(groupHbO, :)
% #####################################################
```

source	detector	type	cond	F	df1	df2	p	q
1	1	'hbo'	'group'	20.676	1	9170	5.5104e-06	2.6478e-05
1	2	'hbo'	'group'	22.079	1	9170	2.655e-06	1.4258e-05
2	6	'hbo'	'group'	37.122	1	9170	1.1545e-09	1.2555e-08
3	1	'hbo'	'group'	25.815	1	9170	3.8323e-07	2.6395e-06
3	3	'hbo'	'group'	8.76	1	9170	0.0030869	0.0074164
4	2	'hbo'	'group'	30.431	1	9170	3.5542e-08	3.1929e-07
4	3	'hbo'	'group'	39.858	1	9170	2.8574e-10	3.69e-09
5	3	'hbo'	'group'	10.764	1	9170	0.001039	0.0029407
5	4	'hbo'	'group'	214.57	1	9170	4.8063e-48	9.9308e-46
6	3	'hbo'	'group'	14.872	1	9170	0.00011584	0.00041992
6	5	'hbo'	'group'	40.371	1	9170	2.2005e-10	3.0312e-09
7	4	'hbo'	'group'	8.3838	1	9170	0.0037947	0.0088096
9	9	'hbo'	'group'	29.582	1	9170	5.4967e-08	4.5429e-07
10	12	'hbo'	'group'	13.185	1	9170	0.00028374	0.00091603



S5-D4

$F(1, 9170) = 214.57$, $p<0.001$, $p_{FDR} <0.001$

Formulation

```
%% Follow-up
job= nirs.modules.MixedEffects();

job.formula=              'beta ~ group*cond + age + (1|subject)';
job.dummyCoding=          'effects';
job.weighted=             1;
job.include_diagnostics=  1;

GroupStats_ME_effects= job.run(SubjStats);
```

```
>> GroupStats_ME_effects.variables(GroupStats_ME_effects.variables.source==5 & GroupStats_ME_effects.variables.detector==4, :)

ans =

  10×5 table

    source    detector    type        cond               model
    ______    ________    _____    ______________    __________________

      5          4        'hbo'    '(Intercept)'     [1×1 LinearModel]
      5          4        'hbo'    'A'               [1×1 LinearModel]
      5          4        'hbo'    'G1'              [1×1 LinearModel]
      5          4        'hbo'    'age'             [1×1 LinearModel]
      5          4
      5          4
      5          4        'hbr'    'A'               [1×1 LinearModel]
      5          4        'hbr'    'G1'              [1×1 LinearModel]
      5          4        'hbr'    'age'             [1×1 LinearModel]
      5          4        'hbr'    'A:group_G1'      [1×1 LinearModel]
```

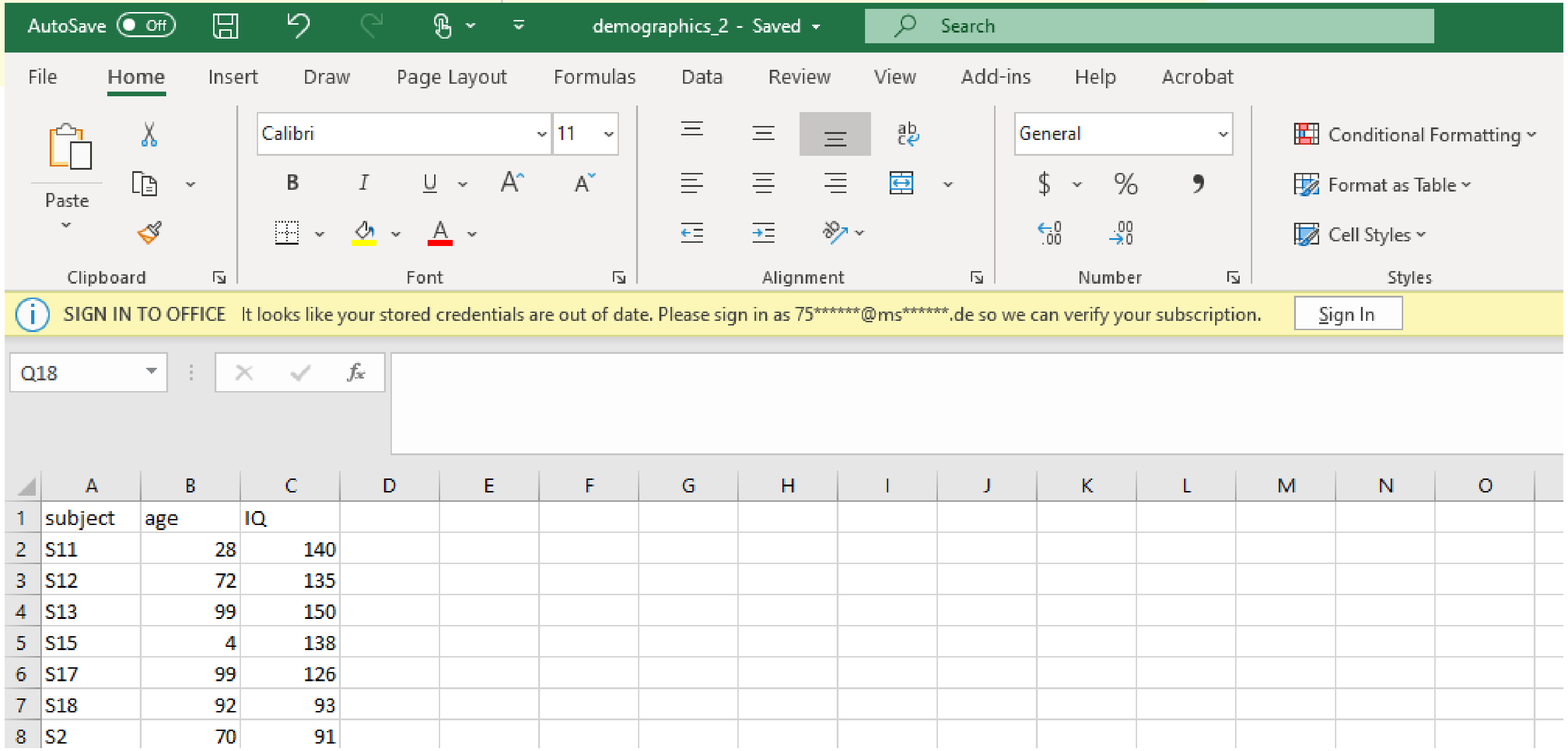
Access single models (resulting from *fitlme*)

```
>> s5d4_effects.model{1}

ans =


Linear regression model:
    beta ~ x_Intercept_ + A + G1 + age + A_group_G1

Estimated Coefficients:
                    Estimate         SE         tStat       pValue
                   ___________    __________    _______    __________

    x_Intercept_      0.025452    0.00097915     25.994    5.5998e-144
    A                0.0032255    0.00072651     4.4396     9.1121e-06
    G1                0.013197     0.0010424      12.66     1.9247e-36
    age            -0.00012909    3.8714e-05    -3.3344     0.00085807
    A_group_G1      -0.0014188    0.00072653    -1.9529        0.05086


Number of observations: 9520, Error degrees of freedom: 9515
Root Mean Squared Error: 0.967
```

Add new demo: IQ

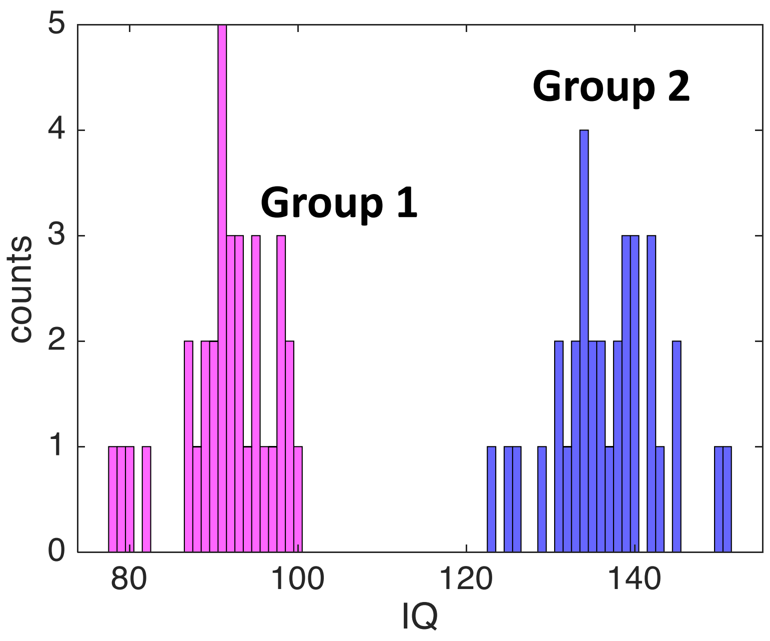
```
%% Add new demo:
csvFile=            readtable(fullfile(dataFolder, 'demographics_2.csv'));
job =               nirs.modules.AddDemographics();
job.demoTable =     csvFile;
% Match the file to an existing column:
job.varToMatch =    'subject';
SubjStats_newDemo=  job.run(SubjStats);
newDemo =           nirs.createDemographicsTable(SubjStats_newDemo);
```

No need to start over, you can append new info to the subject-level statistical results



subject	age	IQ
S11	28	140
S12	72	135
S13	99	150
S15	4	138
S17	99	126
S18	92	93
S2	70	91

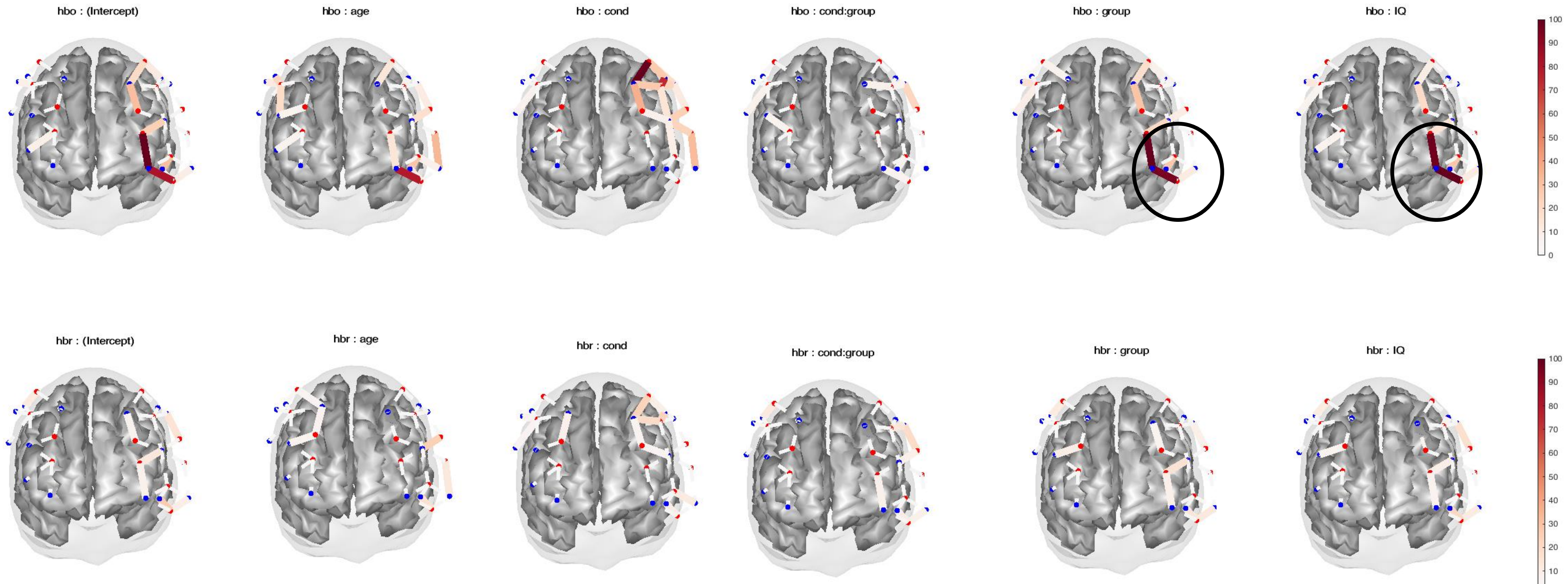
Add new demo: IQ



IQ very different between the two groups

```
% Redo exactly same analysis
%% Group-level analysis (regressors: group and condition, random intercept: subject)
% Step 1) : LMM + ANOVA to get main effects and interaction

job= nirs.modules.Anova();

job.formula=          'beta ~ group*cond + age + IQ + (1|subject)';
job.dummyCoding=      'effects';
job.weighted=          1;

GroupStats_ANOVA_IQ = job.run(SubjStats_newDemo);
```

Effect of including IQ



Effect of including IQ

```
% ####################################################
% See results in table
idx= find(GroupStats_ANOVA_IQ.q < 0.05);
T=  GroupStats_ANOVA_IQ.table; T= T(idx, :);
groupIQHbO = find(strcmpi(T.cond,'group') | strcmpi(T.cond,'IQ') & strcmpi(T.type,'hbo'));
T(groupIQHbO, :)
```

31×9 table

source	detector	type	cond	F	df1	df2	p	q
1	1	'hbo'	'IQ'	9.7669	1	9100	0.0017824	0.0062961
1	1	'hbo'	'group'	17.95	1	9100	2.2903e-05	0.00014935
1	2	'hbo'	'group'	5.5715	1	9100	0.018276	0.041874
1	6	'hbo'	'group'	7.5703	1	9100	0.0059454	0.016732
3	1	'hbo'	'IQ'	19.292	1	9100	1.1343e-05	8.1194e-05
3	1	'hbo'	'group'	29.481	1	9100	5.7931e-08	9.8222e-07
3	1	'hbr'	'group'	5.5135	1	9100	0.018891	0.042898
4	2	'hbr'	'group'	8.5293	1	9100	0.0035032	0.010735
4	3	'hbo'	'group'	10.209	1	9100	0.0014026	0.0051697
5	3	'hbo'	'IQ'	15.584	1	9100	7.9514e-05	0.00044939
5	3	'hbo'	'group'	20.721	1	9100	5.3816e-06	4.2771e-05
5	3	'hbr'	'group'	15.651	1	9100	7.6757e-05	0.00044367
5	4	'hbo'	'IQ'	97.956	1	9100	5.5886e-23	2.8427e-21
5	4	'hbo'	'group'	144.08	1	9100	6.0593e-33	1.541e-30
5	4	'hbr'	'group'	8.5494	1	9100	0.0034648	0.010735
7	7	'hbo'	'IQ'	27.563	1	9100	1.5548e-07	2.326e-06
7	7	'hbo'	'group'	27.337	1	9100	1.747e-07	2.4684e-06
8	4	'hbo'	'IQ'	110.07	1	9100	1.3213e-25	1.1201e-23

Before including IQ, effect of group for channel S5-D4 was:

$F(1, 9170)= 214.57$, $p<0.001$, $p_{FDR} <0.001$

Effect of including IQ

```
%% Follow-up
job= nirs.modules.MixedEffects();

job.formula=            'beta ~ group*cond + age + IQ + (1|subject)';
job.dummyCoding=        'effects';
job.weighted=           1;
job.include_diagnostics=    1;

GroupStats_ME_IQ_effects= job.run(SubjStats_newDemo);
```

```
s5d4_effects_IQ =

  12×5 table

    source    detector    type        cond             model
    ______    ________    _____    _____________    __________________

      5          4        'hbo'    '(Intercept)'    [1×1 LinearModel]
      5          4        'hbo'    'A'              [1×1 LinearModel]
      5          4        'hbo'    'G1'             [1×1 LinearModel]
      5          4        'hbo'    'age'            [1×1 LinearModel]
      5          4        'hbo'    'IQ'             [1×1 LinearModel]
      5          4        'hbo'    'A:group_G1'     [1×1 LinearModel]
      5          4        'hbr'    '(Intercept)'    [1×1 LinearModel]
      5          4        'hbr'    'A'              [1×1 LinearModel]
      5          4        'hbr'    'G1'             [1×1 LinearModel]
      5          4        'hbr'    'age'            [1×1 LinearModel]
      5          4        'hbr'    'IQ'             [1×1 LinearModel]
      5          4        'hbr'    'A:group_G1'     [1×1 LinearModel]
```

```
>> s5d4_effects_IQ.model{1}

ans =


Linear regression model:
    beta ~ x_Intercept_ + A + G1 + age + IQ + A_group_G1

Estimated Coefficients:
                     Estimate          SE         tStat       pValue
                    ___________    __________    _______    __________

    x_Intercept_       0.023509     0.0010531     22.324    1.2197e-107
    A                 0.0032674    0.00072932     4.4801     7.5474e-06
    G1                 0.037899     0.0042713     8.8729     8.4122e-19
    age             -0.00019766    3.9731e-05    -4.9749     6.6408e-07
    IQ                -0.001159    0.00020713    -5.5954     2.2619e-08
    A_group_G1       -0.0013639    0.00072915    -1.8705       0.061448


Number of observations: 9520, Error degrees of freedom: 9514
Root Mean Squared Error: 0.97
```

Effect of including IQ

- Access to single models is very useful to look up any other info about the fit (goodness of fit measures, model criteria, ..) → **model selection**

```
s5d4_effects_IQ =

  12×5 table
```

source	detector	type	cond	model
5	4	'hbo'	'(Intercept)'	[1×1 LinearModel]
5	4	'hbo'	'A'	[1×1 LinearModel]
5	4	'hbo'	'G1'	[1×1 LinearModel]
5	4	'hbo'	'age'	[1×1 LinearModel]
5	4	'hbo'	'IQ'	[1×1 LinearModel]
5	4	'hbo'	'A:group_G1'	[1×1 LinearModel]
5	4	'hbr'	'(Intercept)'	[1×1 LinearModel]
5	4	'hbr'	'A'	[1×1 LinearModel]
5	4	'hbr'	'G1'	[1×1 LinearModel]
5	4	'hbr'	'age'	[1×1 LinearModel]
5	4	'hbr'	'IQ'	[1×1 LinearModel]
5	4	'hbr'	'A:group_G1'	[1×1 LinearModel]

Property	Value
Residuals	9520x4 table
Fitted	9520x1 double
Diagnostics	9520x7 table
MSE	0.9401
Robust	[ ]
RMSE	0.9696
Formula	1x1 LinearFormula
LogLikelihood	-1.3211e+04
DFE	9514
SSE	8.9440e+03
SST	1.0319e+04
SSR	1.3749e+03
CoefficientCovariance	6x6 double
CoefficientNames	1x6 cell
NumCoefficients	6
NumEstimatedCoefficients	6
Coefficients	6x4 table
Rsquared	1x1 struct
ModelCriterion	1x1 struct
VariableInfo	7x4 table
NumVariables	7
VariableNames	7x1 cell
NumPredictors	6
PredictorNames	6x1 cell
ResponseName	'beta'
NumObservations	9520
Steps	[ ]
ObservationInfo	9520x4 table
Variables	9520x7 table
ObservationNames	0x0 cell

Effect of including IQ

- Access to single models is very useful to look up any other info about the fit (goodness of fit measures, model criteria, ..)
  → **model selection**

Akaike information criterion (AIC):
without IQ : 26376
with IQ: 26434

Bayesian information criterion (BIC) :
without IQ:26412
with IQ : 26477

- Including the IQ did not improve the model under neither of the criteria

Summary

- Always enter covariates that are known to have correlatation to the outcome variable
- Enter covariates if they are not homogeneously distributed across groups of interest
- Brain AnalyZIR toolbox has methods for Mixed Effects + ANOVA at group level
- **Model selection:** check the model fit and the model criteria to see if it improves the fit (→ then, it is explanatory of the data)

Recommended resources

- Statistics without math for psychology, by Dancey and Reidy
- Discovering statistics using SPSS/R, by Andy Field
- Multilevel modelling free online course offered by University of Bristol (http://www.bristol.ac.uk/cmm/learning/online-course/)
- Howard J. Seltman's book *"Experimental design and analysis"* (freely available at http://www.stat.cmu.edu/~hseltman/309/Book/Book.pdf), and in general his whole website – it has tons of resources