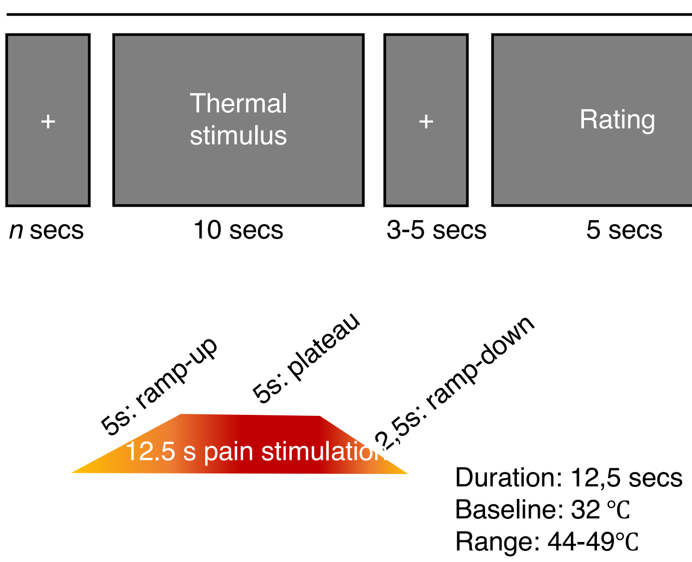
Multilevel model: tutorial

- Let's try to perform multilevel model using SEMIC pain calibration data (*glmfit_multilevel* in CanlabTools)
  - It is exactly same as I described previously (Two-stage model)
- The goal of this analysis is to estimate the temperature effects on pain ratings, considering between-participant variance

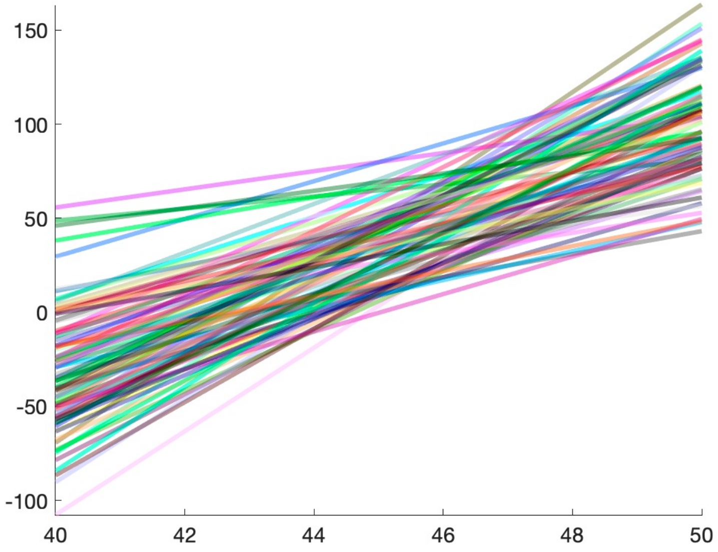


Tasks

- Participants: $N = 84$
- There were 12 trials per each
- We have information about pain ratings and temperature
- First-level: individual level (12 trials per individual
- Second-level: group level (84 participants)

Multilevel model: tutorial

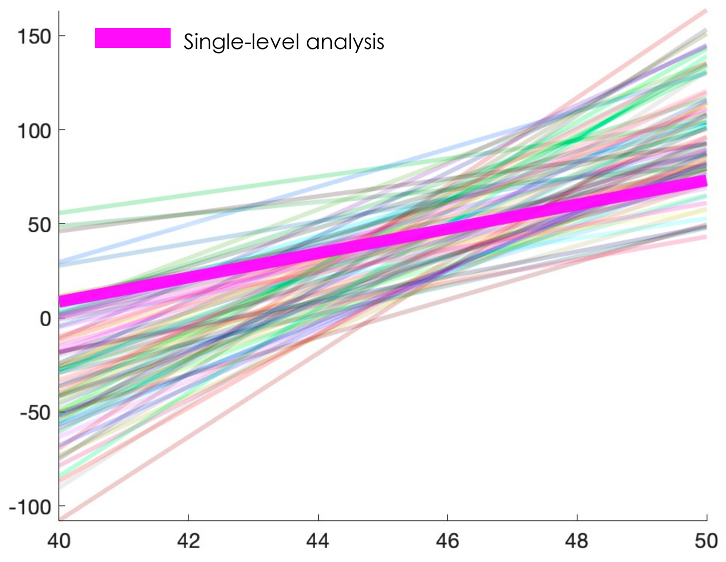
- Let's fit each first-level regression model
- Although all slopes show similar patterns, but there are not exactly same
  → between-subject variability




```
%% SET PATH
% calibration datasets
basedir = '/Users/suhwan/Dropbox/Projects/SEMIC/data_NAS/behavioral/raw'
datdir = fullfile(basedir,'CALI_SEMIC_data'); % sein
matlist = filenames(fullfile(datdir,'*.mat'));

%% LOAD DATA
cal_dat = [];
for i = 1:length(matlist)
    cal_dat{i}=load(matlist{i},'reg');
end

%% Comparsion between single-level linear regression and first-level regression
% 1. Estimating each participant's first-level regression model
create_figure;
set(gca,'Ylim',[0 100],'Xlim', [40 50]);
clear h1line;
col = colorcube(length(cal_dat));
col = col(shuffles(1:length(cal_dat)),:);

for i = 1:length(cal_dat)
    % refline function is to repersent regression line on a plot
    slope = cal_dat{i}.reg.total_fit.Coefficients.Estimate(2);
    intercep = cal_dat{i}.reg.total_fit.Coefficients.Estimate(1); %intercept
    h1line(i) = refline([slope intercep]);
    h1line(i).LineWidth = 3;
    h1line(i).Color = [col(i,:) 0.2];

    hold on;
end

hold off;
```

Multilevel model: tutorial

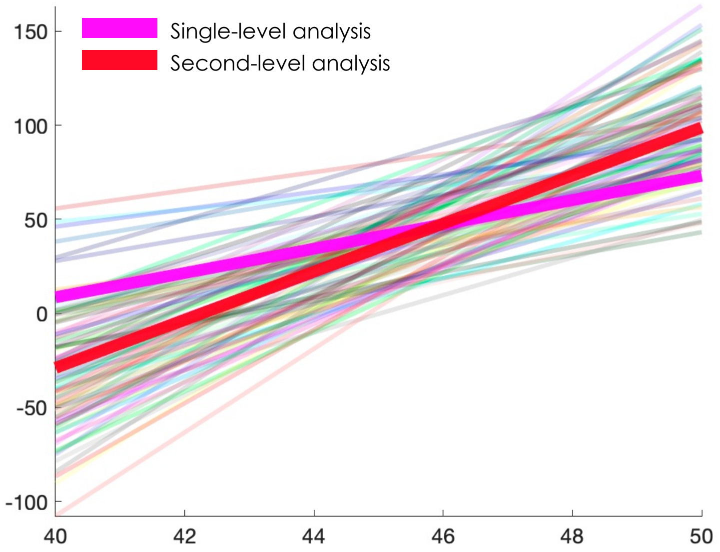
- Let's compare between single-level analysis results and the trends of first-level analysis results
- In this case, all data were aggregated without considering the participants
- The estimated regression line quite differ compared to trends of the first-level analysis results




```
% 2. single-level regression model
rating = [];
degree = [];
% aggregate all variables
for i = 1:length(cal_dat)
    degree = [degree cal_dat{i}.reg.stim_degree];

    rating = [rating cal_dat{i}.reg.stim_rating];
end
res_lm = fitlm(degree, rating); % fit linear model

clear h2line
slope = res_lm.Coefficients.Estimate(2);
intercep = res_lm.Coefficients.Estimate(1)]; %intercept
h2line = refline([slope intercep]);
h2line.Color = [1 0 1 0.9];
h2line.LineWidth = 8;
```

Multilevel model: tutorial

- The next is to consider hierarchical data structures
- Using glmfit_multilevel in CanlabCORE, we can estimate the second-level analysis
  - **help glmfit_multilevel**




```
%
stats = glmfit_multilevel(y,x1,[],'names',{'intercept','temperature'},'verbose');
%%
```

```
%% Multilevel linear regression
x1 = [];
y = [];
% making data structure depends on participants
for i = 1:length(cal_dat)
    x1{i} = cal_dat{i}.reg.stim_degree';
    y{i}  = cal_dat{i}.reg.stim_rating';
end
%
stats = glmfit_multilevel(y,x1,[],'names',{'intercept','temperature'},'verbose');
%%
clear h3line
h2line = refline([stats.beta(2) stats.beta(1)]);
h2line.Color = [1 0 0.1 0.9];
h2line.LineWidth= 8;
```

Second Level of Multilevel Model

	Outcome variables:	
	intercept	temperature
Adj. mean	-539.95	12.77

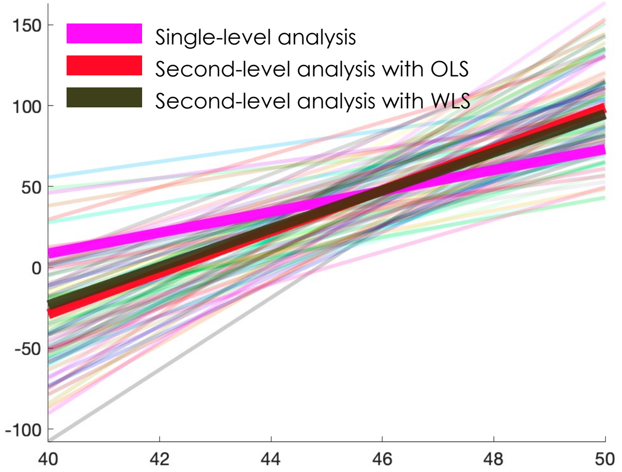
2nd-level B01

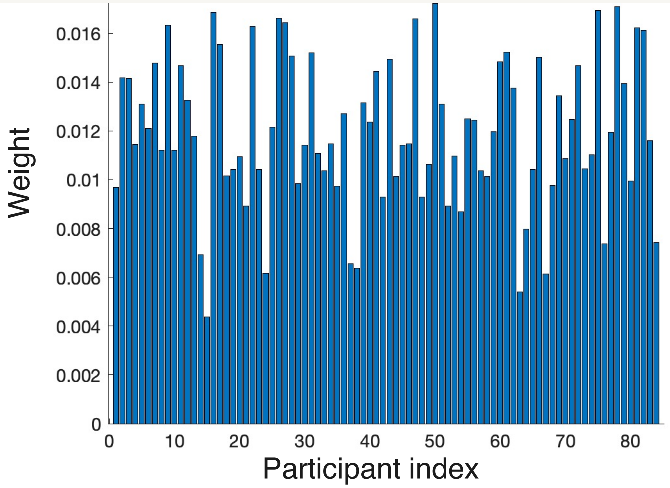
	intercept	temperature
Coeff	-539.95	12.77
STE	23.88	0.52
t	-22.61	24.56
Z	-8.21	8.21
p	0.00000	0.00000

Multilevel model: tutorial

- Finally, there are several useful options to estimate multilevel regression models in **glmfit_multilevel**
  - **'weighted' :** Using empirical Bayesian estimation, each first-level results will be reweighted. In other words, if the first-level results is unstable, their weight is down-weighted when a second-level beta estimated
  - **'boot':** One of the non-parametric significance inference options with random sampling

```
stats2 = glmfit_multilevel(y, x1, [], ...
    'names', {'intercep','temp'}, 'verbose', 'boot', 'nresample', 10000,'weighted');
```





Bootstrapped statistics. **Bootstrap** inference

	Outcome variables:	
	intercep	temp
Adj. mean	-496.60	11.83
2nd-level B01		
	intercep	temp
Coeff	-496.42	11.82
STE	16.60	0.35
t	-20.98	23.05
Z	-3.72	3.81
p	0.00020	0.00014

Multilevel model: Additional issues

- To estimate more accurately, usually the multilevel analysis is useful methods because of decomposition of variance
- In our examples, or tutorial, the OLS (or WLS) was used to estimate the beta
- However, the other packages such as **LMER4 packages** in R language employ maximum likelihood estimation or restricted maximum likelihood to estimate beta coefficients

https://cran.r-project.org/web/packages/**lme4**/vignettes/lmer.pdf

Multilevel model: Additional issues

- In psychology, there are consensus the multilevel approach can be helpful to make inference population level results
    - Because it can decompose between-each level variance by building each level regression models
- However, someone raise a question of this approach. So, if you have interest, I will recommend you the article (Yarkoni 2021)

The generalizability crisis

Tal Yarkoni[1*]

[1]Department of Psychology, University of Texas at Austin
[*]Email: tyarkoni@gmail.com

Yarkoni, T. (2020). The generalizability crisis. Behavioral and Brain Sciences, 1-37. doi:10.1017/S0140525X20001685