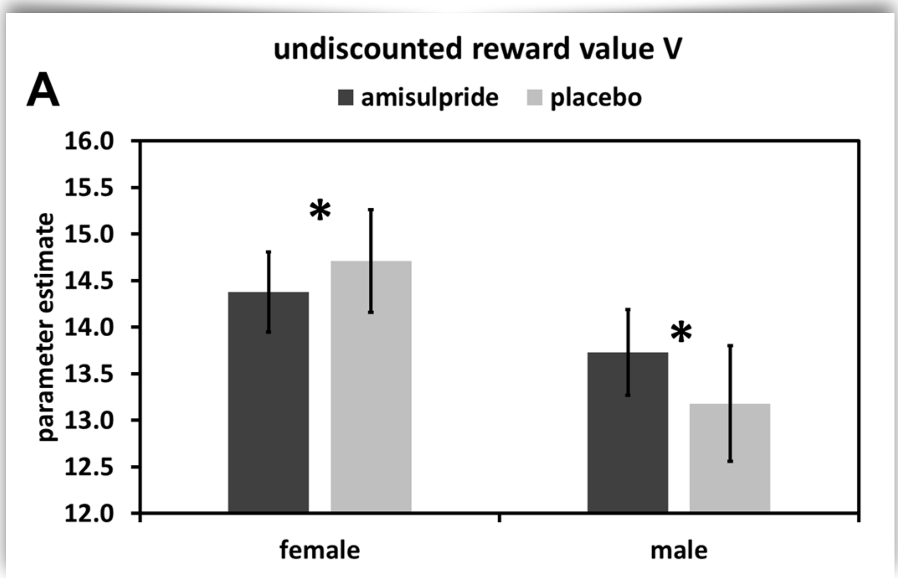
nplot_error - plotting data with the correct error bars

ISN Methods Meeting Dec 4, 2019

Jan Gläscher

A common problem in most papers …



Error bars are SEM

We analysed parameter estimates with a mixed-measures ANOVA including the factors Substance (amisulpride vs. placebo), Parameter (V vs. s), Gender (female vs. male), and Substance Order (amisulpride-placebo vs. placebo-amisulpride). We found significant effects of Parameter, $F(1, 51) = 1341.33$, $p < 0.001$, $\eta^2 = 0.963$, and Substance × Gender, $F(1, 51) = 8.92$, $p < 0.01$, $\eta^2 = 0.149$, which were specified by a three-way Parameter × Substance × Gender interaction, $F(1, 51) = 8.88$, $p < 0.01$, $\eta^2 = 0.148$ (these effects are robust to mean-centering the parameter estimates for V and s and therefore not due to scaling differences between V and s). These results suggest that amisulpride affected parameters V and s differently in the two genders (Figure 3). More specifically, we found that under amisulpride, relative to placebo, the intercept V was significantly reduced in female participants, $t(26) = 2.07$, $p < 0.05$, and significantly increased in male participants, $t(27) = 2.13$, $p < 0.05$, while there were no drug effects on the discount factor s in either gender, both $t < 1$, both $p > 0.50$.

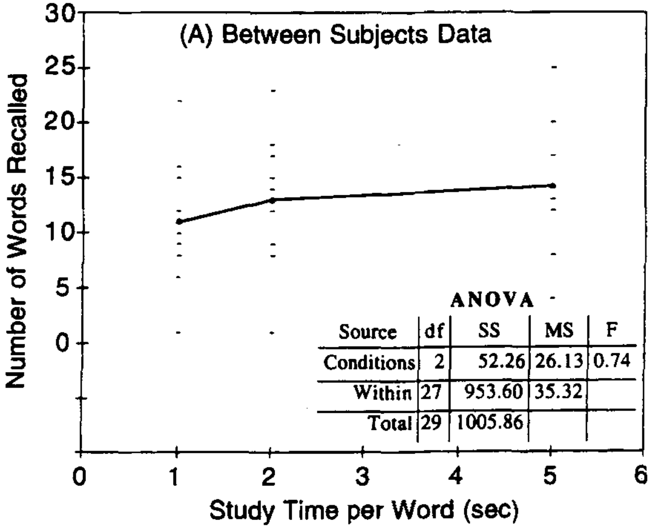
=> most papers only use between-subject error bars for visualization despite appropriate usage of within-subject statistics

A step-by-step example

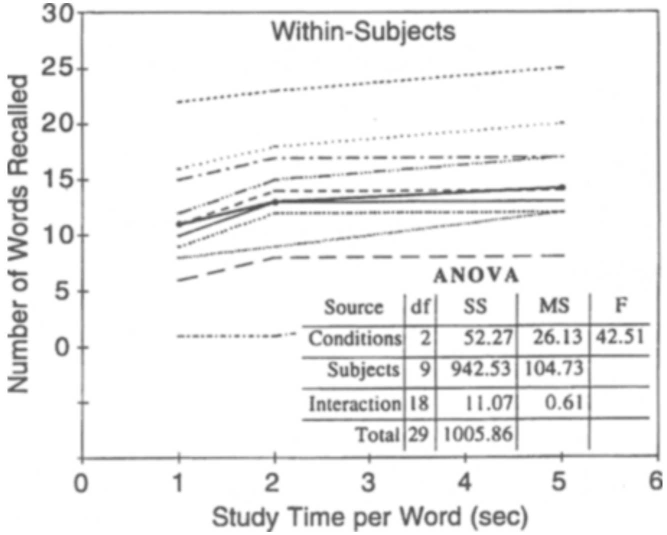
treating repeated measures as a between-subject effect

visualizing and removing between subject variance

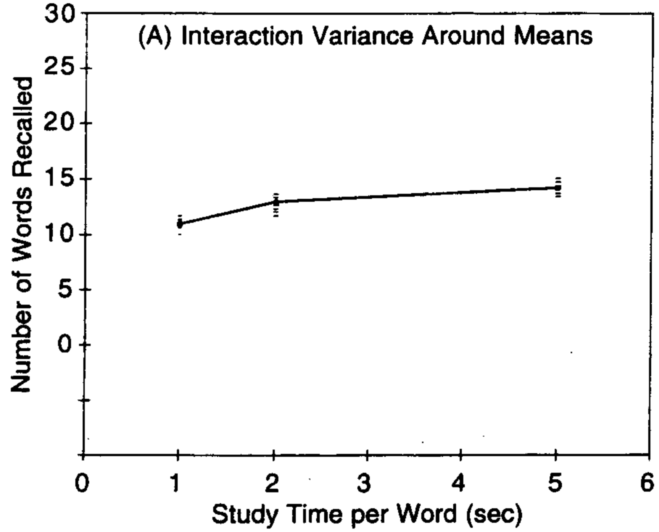
treating repeated measures as a within-subject effect

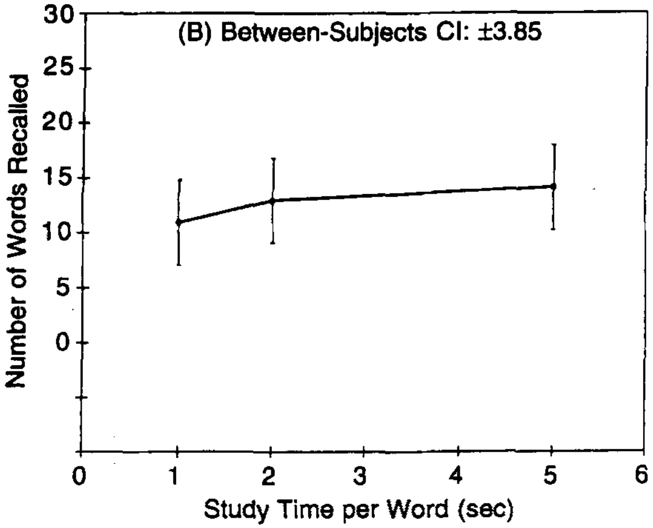


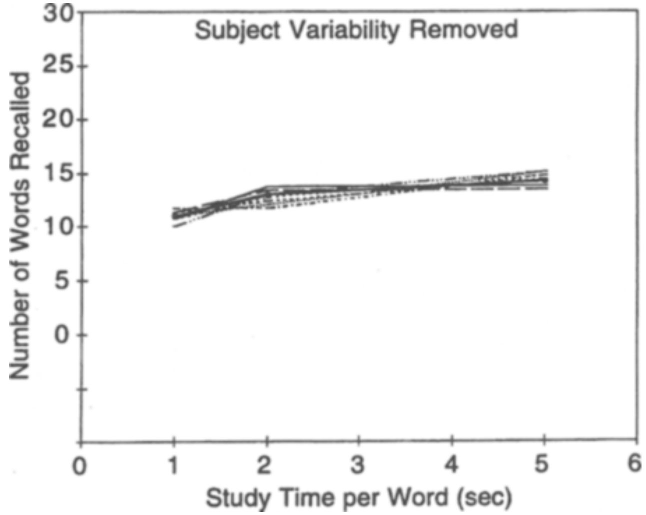
Source	df	SS	MS	F
Conditions	2	52.26	26.13	0.74
Within	27	953.60	35.32	
Total	29	1005.86		

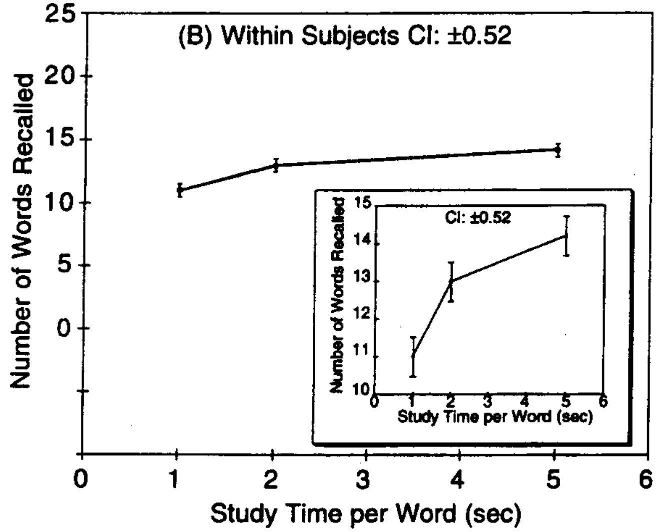


Source	df	SS	MS	F
Conditions	2	52.27	26.13	42.51
Subjects	9	942.53	104.73	
Interaction	18	11.07	0.61	
Total	29	1005.86		









from: Loftus & Masson, Psychon Bull Rev, 1994

An intuitive understanding of within-subject errors

- typical situation for repeated measures designs:
  between-subject variance is much larger than within-subject (condition-specific) variance

- approach: remove between-subject variance and then plot the error bars

How do I compute a within-subject error bar?

- **Loftus & Masson (1994). Using confidence intervals in within-subject designs, PBR, 1(4), 476-490.**
  - construct and CI (or SEM respectively) based on the error variance in an ANOVA
    - statistically sound, but requires running an ANOVA to compute the size of the error
    - multiple sources of errors in an ANOVA design: which one should be picked (MS_C, MS_S or MS_{WxS})?
    - same error bar for all conditions

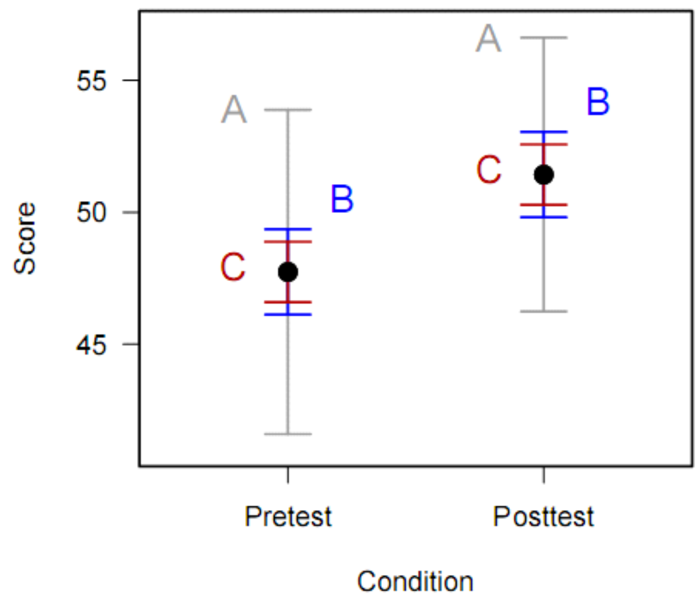
How do I compute a within-subject error bar?

- Cosineau (2005). Confidence intervals in within-subject designs: a simpler solution to Loftus & Masson's method. *Tutorials in Quantative Methods in Psychology,* 1(1), 42-45.
  - normalize within-subject data first -> remove between-subject effect by subtracting subject mean and grand mean, then use between-subject CI or SEM
  - MATLAB:

```
[n,m]     = size(data);
norm_data = data - ( repmat(mean(data,2),1,m) - ...
   repmat(mean(data(:)),n,m) );
norm_sem  = std(norm_data) / sqrt(n);
```

How do I compute a within-subject error bar?

- Morey (2005). Confidence Intervals from Normalized Data: A correction to Cousineau (2005), *Tutorials in Quantative Methods for Psychology*, 4(20, 61-64.
  - normalization in Cosineau, 2005 induces a positive bias in the variance estimates -> error bars are too small (compared to the statistically correct ones from L&M, 1994)
  - correction factor for sample variance after normalization: M / (M-1), where M is the number of within-subject conditions
  - in case of multi-factor designs, replace M with $\prod_{p=1}^{P} M_p$, where P is the number of fixed factors, and M_p is the number of levels in factor P



- MATLAB:

```
[n,m] = size(data);
norm_data = data - ( repmat(mean(data,2),1,m) - ...
   repmat(mean(data(:)),n,m) );
norm_sem  = std(norm_data) / sqrt(n) * sqrt(m/(m-1))
```

Which error bar is appropriate?

- between-subject effects -> use between-subject error
- within-subject effects -> use within-subject error
- But what if I have a mixed design?
  -> it depends on what you which effect you want to show in the figure
  -> **IMPORTANT:** state the type of error bar in the figure legend!

`nplot_error` - a tool for plotting data
(with the correct error bars :-)

- originated from an old MATLAB plotting utilities by CB (plot_error)
- complete re-write to confirm to MATLAB’s parameter-value input scheme
- many new features … (including within-subject error bars)

List of features

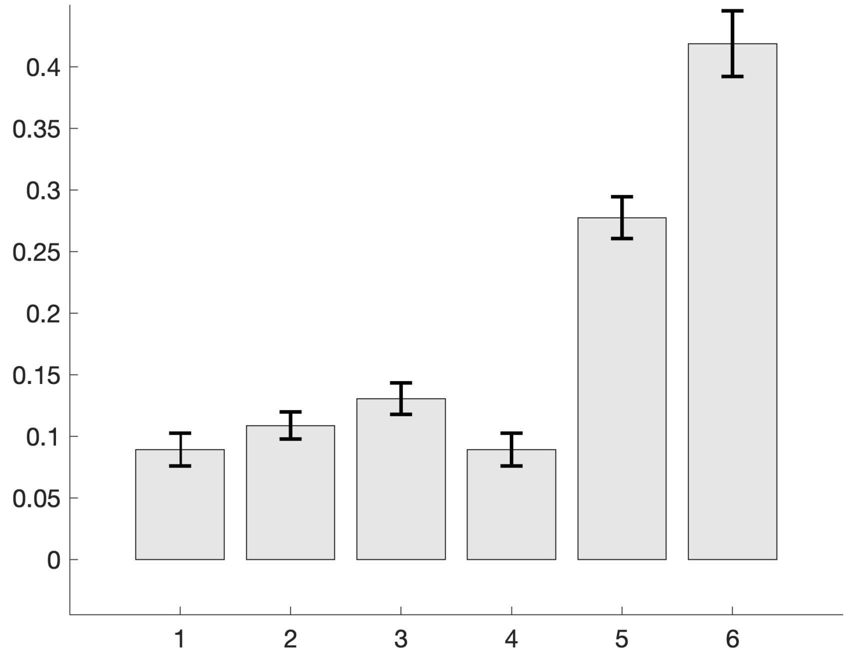
```
% INPUTS
% OPTION             (POSSIBLE) VALUES
% 'config'           struct with config options (from previous nplot_error)
%
% 'type'             'bar','dot','line','fact'
%
% 'data'             nSubjects*nConditions data matrix
% 'mean'             vector of (manually) computed means
% 'error'            vector of (manually) computed errors
%
% 'errorclass'       'bw' (between-subject) or 'ws' (within-subject)
% 'errortype'        'sem','sd','ci' (90%), 'none'
% 'color'            cell array of color specs (1*nCondition) [bar,dot,line] OR
%                    vector of dimension 1*nCondition with values between
%                    1:8 referring to the specialized color palette defined in
%                    cfg.colorder (1=blue, 2=orange, 3=red, 4=green, 5=violet,
%                    6=yellow, 7=magenta, 8=brown)
%                    Colors can be also specified using standard MATLAB characters
%                    (rgbcmykw) or as RGB vectors with values between 0 and 1
% 'errorcolor'       string or numerical color spec for errorbars
%                    special string 's' for "same as bar/dot"
%
% 'indivdata'        plot individual data points as little dots ('on','off')
% 'subjectnum'       plot subject numbers next to dots for individual data
%                    ('on','off')
%
```

List of features

```
% 'group'          cell array of vectors of subject numbers (rows in data
%                  matrix), specify xpos, xtick, and xlabel for one group
%                  only, it will be automatically replicated for the remaining
%                  groups
% 'gnames'         cell array of names for groups
%
% 'factor'         vector of number of level per factor, left-most factor
%                  rotates slowest [type=fact,prod(factor)==size(data,2)]
% 'factornames'    cell array of factor names [type=fact]
% 'levelnames'     cell array of cell array of levelnames (for each factor)
%                  [type=fact]
%
% 'xpos'           positions on x-axis for plotting the data, use this for
%                  visual grouping
% 'xtick'          ticks on x-axis for adding labels for data
% 'xlabel'         cell array of labels for xtick [type=bar/dot/line]
%                  (must correspond to length(xtick)
% 'xaxislabel'     label for x-axis (type=bar/dot/line]
% 'yaxislabel'     label for y-axis
% 'title'          plot title
%
% 'legend'         flag to turn on legend ('on','off')
% 'legnames'       cell array for different colors [type=bar/dot/line]
% 'leglocation'    location for legend (e.g. 'nw','ne','se','sw')
%
% 'figcmd'         command for figure/axes/subplots
```

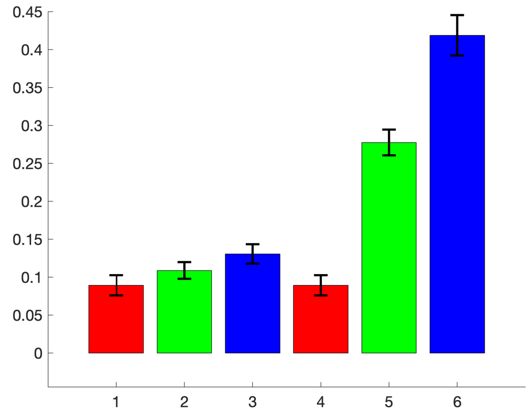
Simple example ...

```
nplot_error('type','bar',...
  'data',data)
```

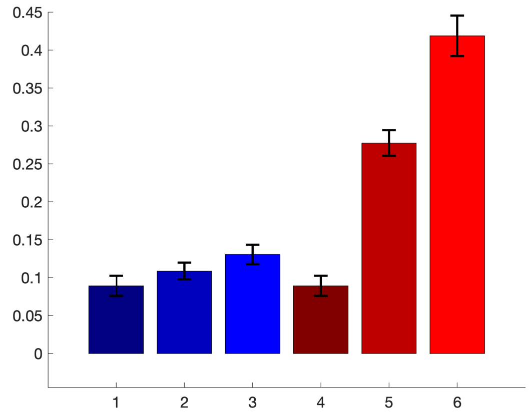


adding colors ...

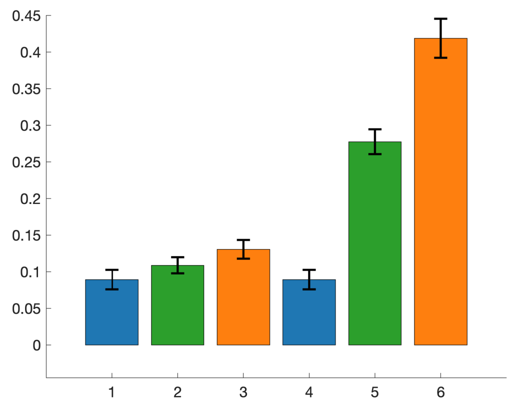
```
nplot_error('type','bar',...
   'data',data,...
   'color',{'r','g','b','r','g','b'})
```




```
nplot_error('type','bar',...
   'data',data,...
   'color',...
   {[0 0 .5],[0 0 .75],[0 0 1],[.5 0 0],[.75 0 0],[1 0 0]})
```

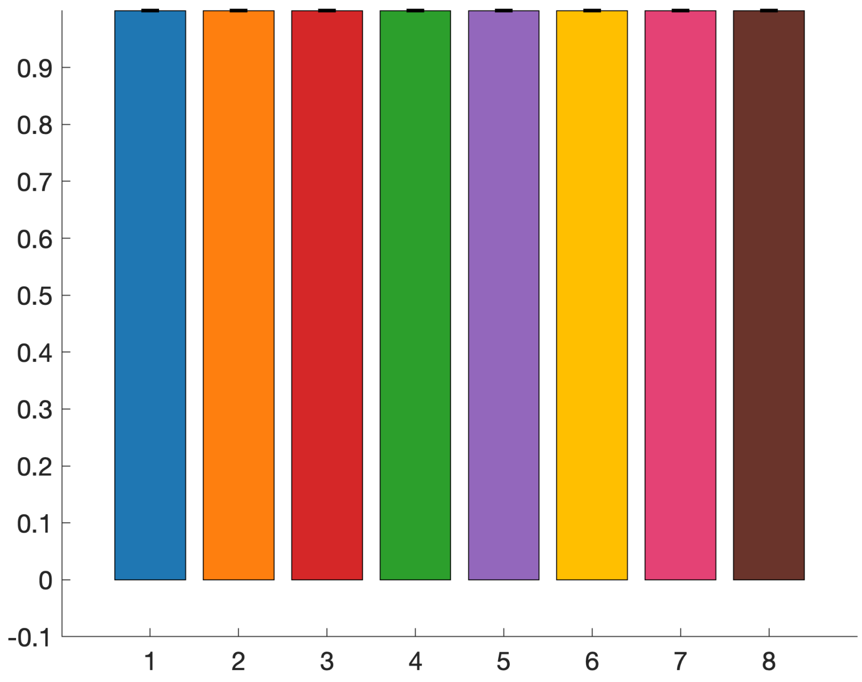



```
nplot_error('type','bar',...
   'data',data,...
   'color',[1 4 2 1 4 2])
```



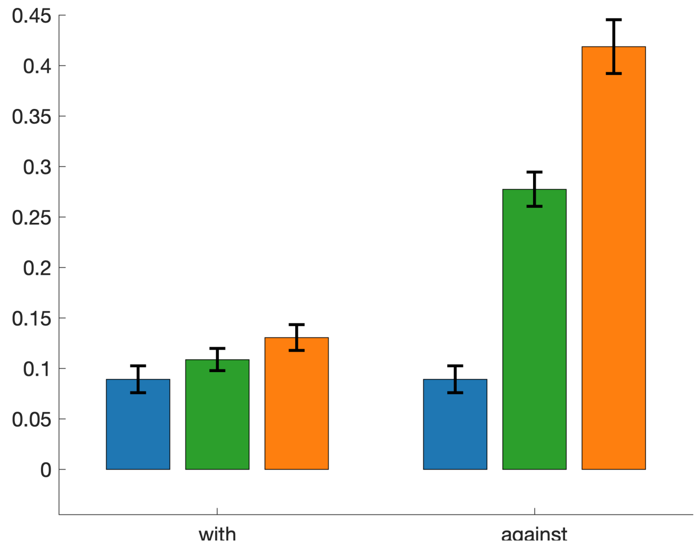
to view all 8 pre-specified colors ....

```
nplot_error('type','bar',...
   'mean',ones(1,8),...
   'error',zeros(1,8),...
   'color',1:8)
```

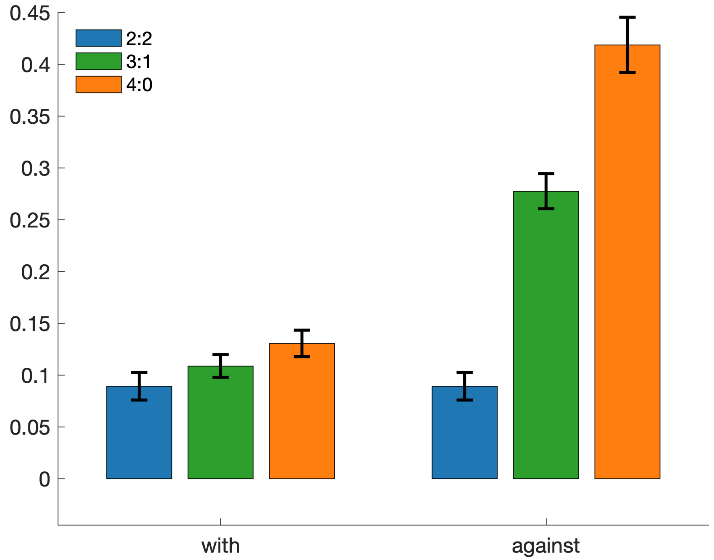


Visual grouping and labels

```
nplot_error('type','bar',...
   'data',data,...
   'color',[1 4 2 1 4 2],...
   'xpos',[1 2 3 5 6 7],...
   'xtick',[2 6],...
   'xlabel',{'with','against'})
```

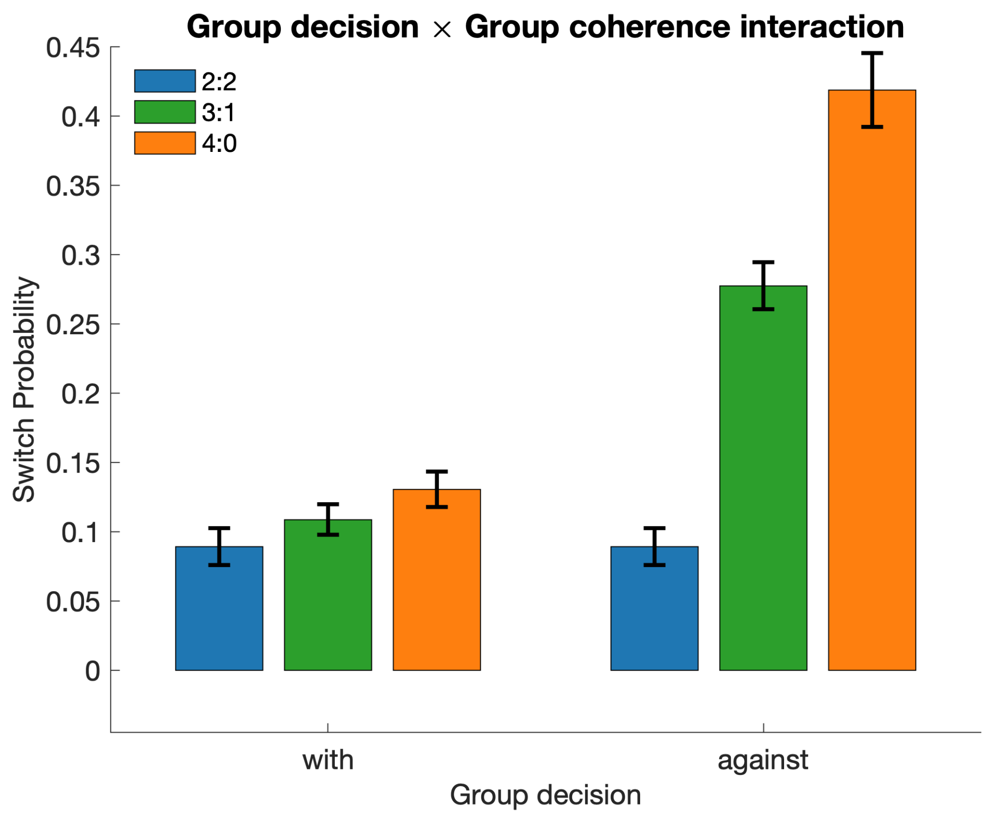



```
nplot_error('type','bar',...
   'data',data,...
   'color',[1 4 2 1 4 2],...
   'xpos',[1 2 3 5 6 7],...
   'xtick',[2 6],...
   'xlabel',{'with','against'},...
   'legend','on',...
   'legnames',{'2:2','3:1','4:0'},...
   'leglocation','nw')
```



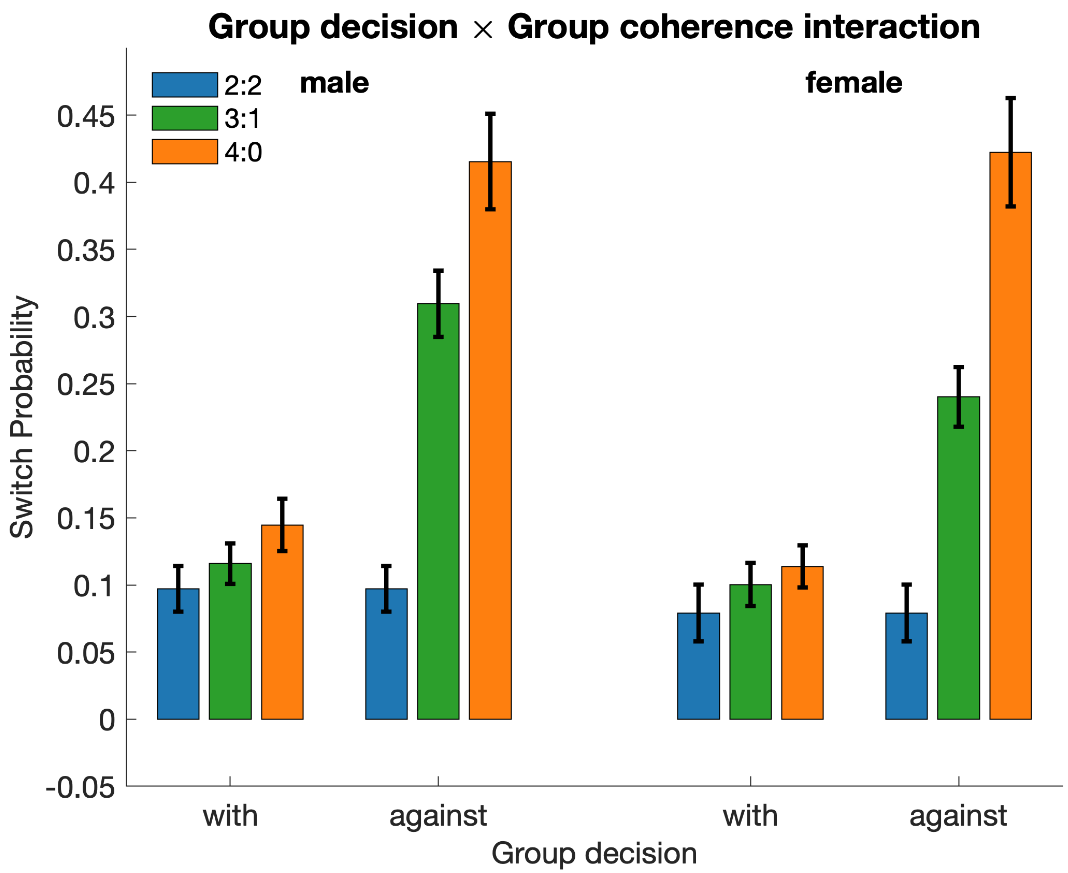
Axis labels and title

```
nplot_error('type','bar',...
  'data',data,...
  'color',[1 4 2 1 4 2],...
  'xpos',[1 2 3 5 6 7],...
  'xtick',[2 6],...
  'xlabel',{'with','against'},...
  'legend','on',...
  'legnames',{'2:2','3:1','4:0'},...
  'leglocation','nw',...
  'xaxislabel','Group decision',...
  'yaxislabel','Switch Probability',...
  'title','Group decision \times Group coherence interaction')
```



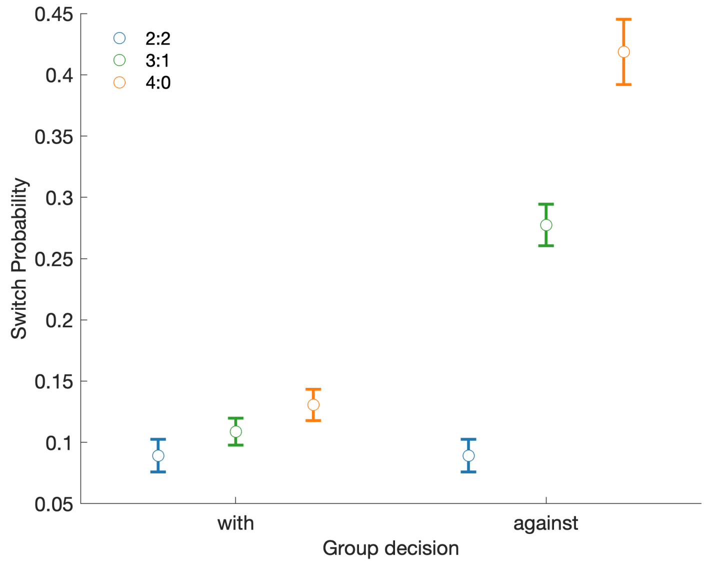
Splitting data into groups

```
nplot_error('type','bar',...
   'data',data,...
   'color',[1 4 2 1 4 2],...
   'xpos',[1 2 3 5 6 7],...
   'xtick',[2 6],...
   'xlabel',{'with','against'},...
   'legend','on',...
   'legnames',{'2:2','3:1','4:0'},...
   'leglocation','nw',...
   'xaxislabel','Group decision',...
   'yaxislabel','Switch Probability',...
   'title','Group decision \times Group coherence interaction',...
   'group',{1:100,101:185},...
   'gnames',{'male','female'})
```

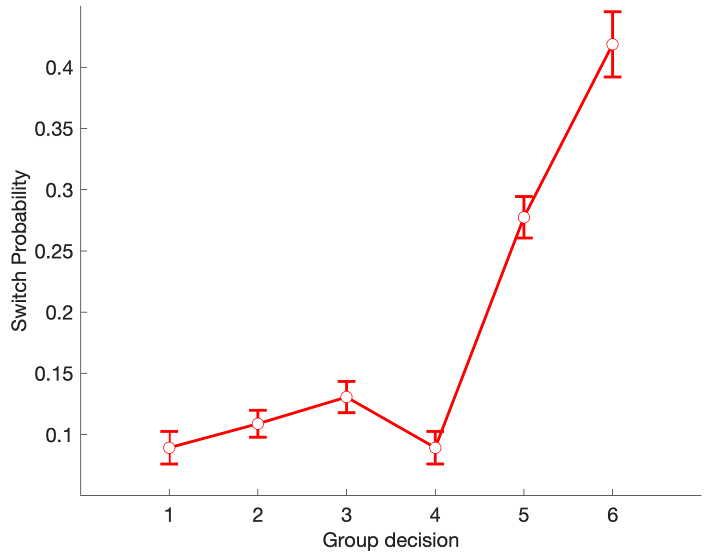


Different plot types

```
nplot_error('type','dot',...
   'data',data,...
   'color',[1 4 2 1 4 2],...
   'errorcolor','s',...
   'xpos',[1 2 3 5 6 7],...
   'xtick',[2 6],...
   'xlabel',{'with','against'},...
   'legend','on',...
   'legnames',{'2:2','3:1','4:0'},...
   'leglocation','nw',...
   'xaxislabel','Group decision',...
   'yaxislabel','Switch Probability')
```




```
nplot_error('type','line',...
   'data',data,...
   'color',{'r'},...
   'errorcolor','s',...
   'xaxislabel','Group decision',...
   'yaxislabel','Switch Probability')
```



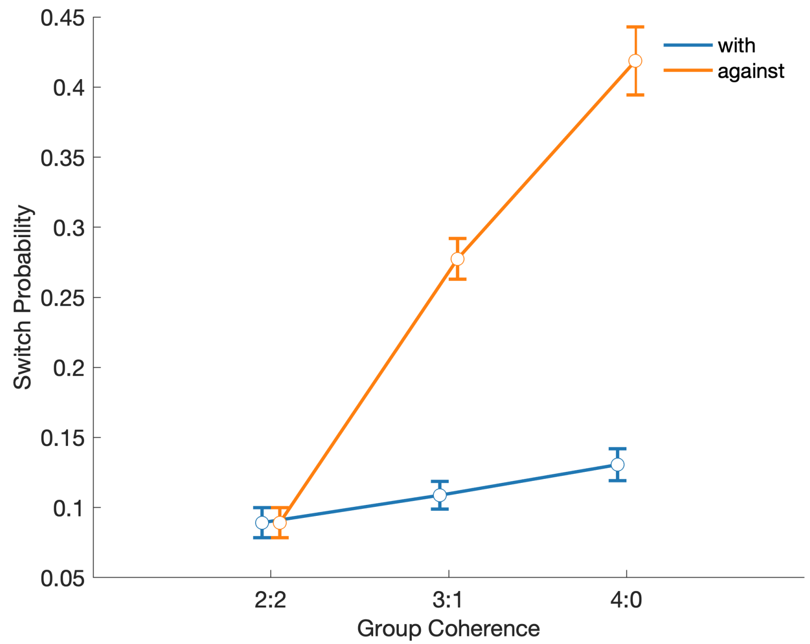
Special plot type: `factorial`

Factorial plots require a specific organization of the data matrix:

- each column is a factor level in ascending order
- left-most factor rotates slowest

Example: 2 factors:

- Group Decision (2 levels) -> Factor A
- Group Coherence (3 levels) -> Factor B
- column in data matrix
  A1B1 A1B2 A1B3 A2B1 A2B2 A2B3



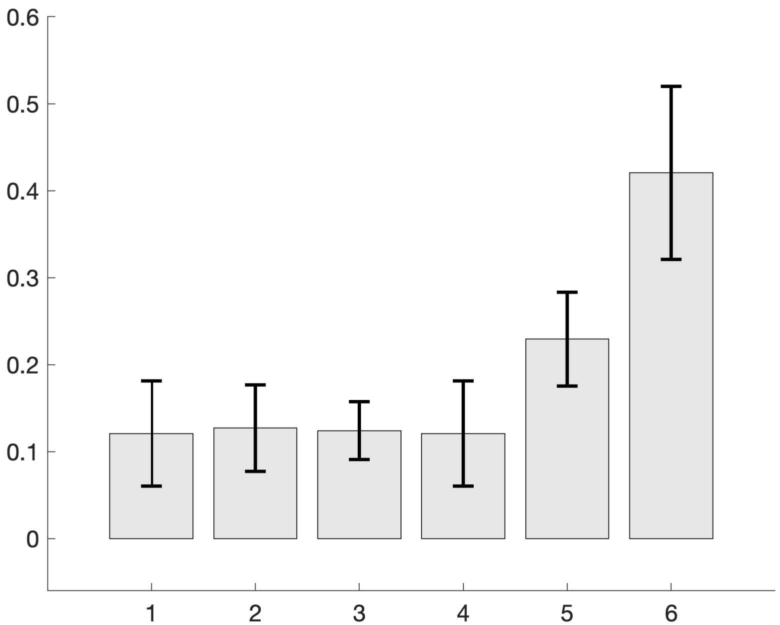
Factorial Plot uses within-subject errors by default!
(unless specified otherwise)

```
nplot_error('type','fact',...
   'data',data,...
   'factor',[2 3],...
   'factornames',{'Group decision','Group Coherence'},...
   'levelnames',{{'with','against'},{'2:2','3:1','4:0'}},...
   'errorcolor','s',...
   'legend','on',...
   'yaxislabel','Switch Probability')
```

Comparing different error types

Between-subject error

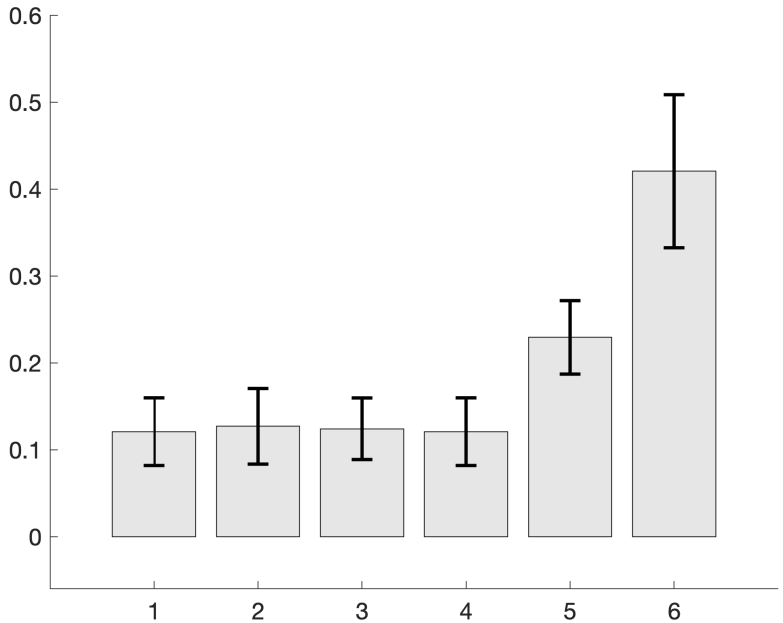
```
[hdl,cfg1] =
nplot_error('type','bar',...
   'data',data(170:181,:),...
   'errorclass','bw')
```




```
>> cfg1.error
ans =
 0.0605 0.0497 0.0333 0.0605 0.0539 0.0995
```

Within-subject error

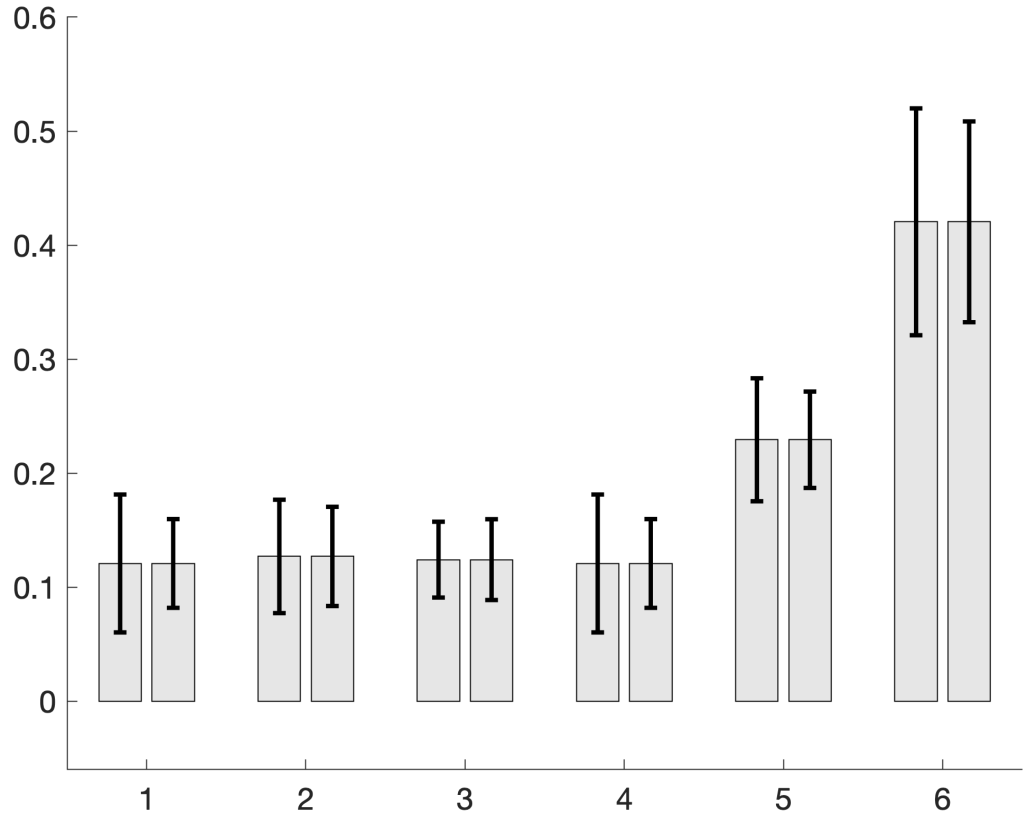
```
[hdl,cfg2] =
nplot_error('type','bar',...
   'data',data(170:181,:),...
   'errorclass','ws')
```




```
>> cfg2.error
ans =
 0.0389 0.0435 0.0354 0.0389 0.0423 0.0880
```

Comparing different error types

```
nplot_error('type','bar',...
  'mean',splice(cfg1.mean,cfg2.mean),...
  'error',splice(cfg1.error,cfg2.error),...
  'xpos',[1 2 4 5 7 8 10 11 13 14 16 17],...
  'xtick',[1.5 4.5 7.5 10.5 13.5 16.5],...
  'xlabel',{'1','2','3','4','5','6'})
```



Making small changes …

```
[hdl,cfg] = nplot_error('type','bar',...
  'data',data,...
  'color',[1 4 2 1 4 2],...
  'xpos',[1 2 3 5 6 7],...
  'xtick',[2 6],...
  'xlabel',{'with','against'},...
  'legend','on',...
  'legnames',{'2:2','3:1','4:0'},...
  'leglocation','nw',...
  'xaxislabel','Group decision',...
  'yaxislabel','Switch Probability',...
  'title','Group decision \times Group coherence interaction',...
  'group',{1:100,101:185},...
  'gnames',{'male','female'})
```

Making small changes …

```
>> cfg =
  struct with fields:

           defs: [1×1 struct]
           type: 'bar'
           data: [185×6 double]
          color: {1×6 cell}
           xpos: [2×6 double]
          xtick: [2×2 double]
         xlabel: {2×2 cell}
         legend: 'on'
       legnames: {'2:2'  '3:1'  '4:0'}
    leglocation: 'nw'
     xaxislabel: 'Group decision'
     yaxislabel: 'Switch Probability'
          title: 'Group decision \times Group coherence interaction'
          group: {[1×100 double]  [1×85 double]}
         gnames: {'male'  'female'}
           mean: [2×6 double]
          error: [2×6 double]
     errorclass: 'bw'
      errortype: 'sem'
     errorcolor: {'k'  'k'  'k'  'k'  'k'  'k'}
      indivdata: 0
     subjectnum: 0

>> nplot_error(‘config’,cfg)
```

Where can I get it? What’s still missing?

`https://github.com/GlascherLab/nplot_error`

Missing features

- time-series plot (e.g. BOLD, SCR, PDR, etc.) across subjects (between-subject errors) with shaded error “area”
- box plots
- violin plots, split violin plots
- bee swarm plots of individual data points (“data cloud”)

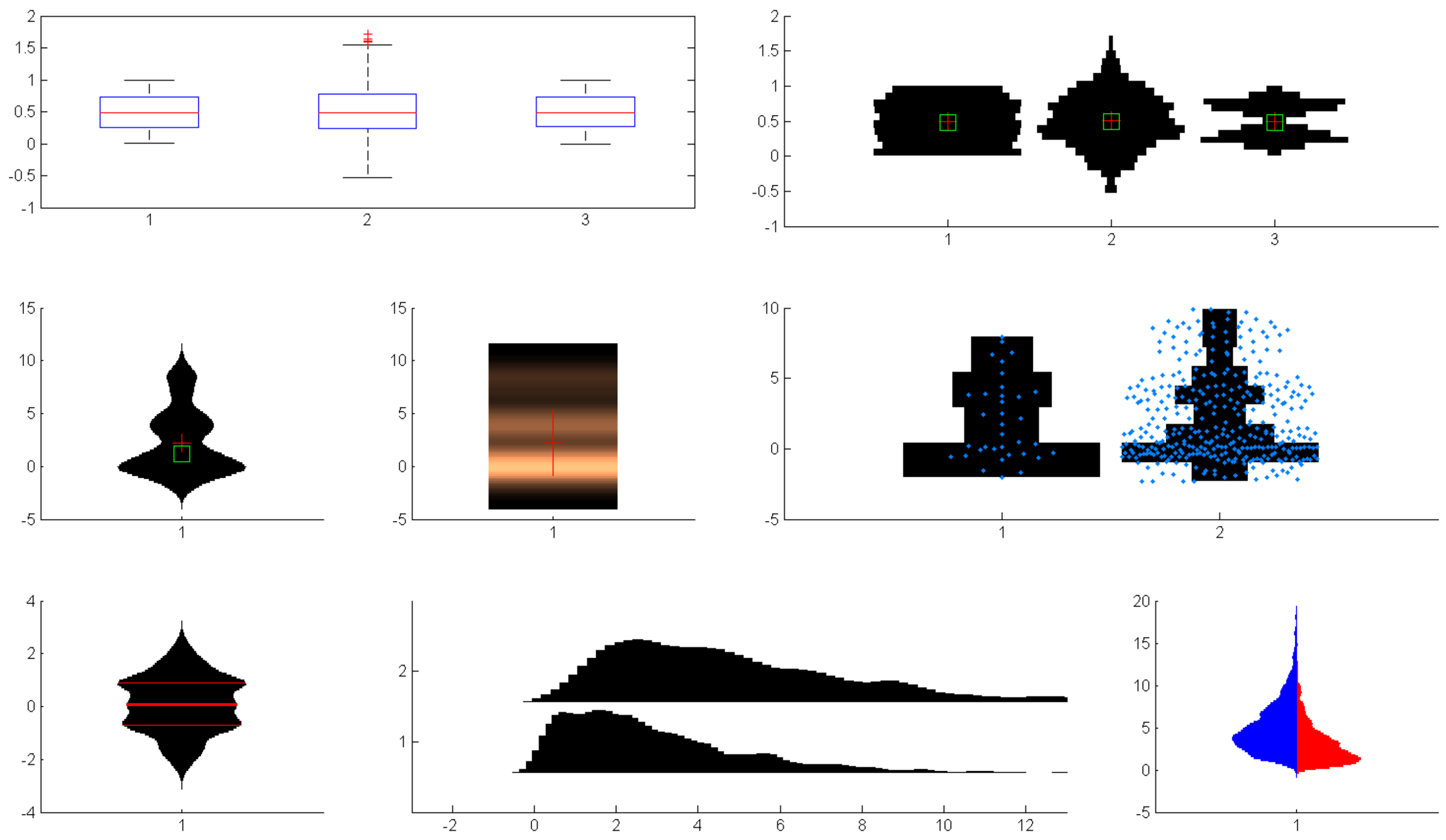
Violin and Bee swarm Plots in MATLAB

distributionPlot.m

https://de.mathworks.com/matlabcentral/fileexchange/23661-violin-plots-for-plotting-multiple-distributions-distributionplot-m

plotSpread.m

https://de.mathworks.com/matlabcentral/fileexchange/37105-plot-spread-points-beeswarm-plot?focused=9cb303a9-261e-a18d-4f33-7ffbdcebc416&tab=function



Raincloud Plot in MATLAB and R

https://github.com/RainCloudPlots/RainCloudPlots