```
*******************************************************************************************
*** Replication materials for:                                                          ***
*** Estimating Social Preferences and Kantian Morality in Strategic Interactions        ***
*** Boris van Leeuwen & Ingela Alger                                                    ***
*******************************************************************************************

Version September 2024
Analyses were done using Stata 17.0/Stata 18.5 and Matlab R2022a/R2023a/R2023b/R2024a

*******************************************************************************************
0. Raw experimental data ******************************************************************
*******************************************************************************************
All data is contained in two files: "allresults.dta" and "allsubjects.dta". All other data files
are derived from these files, the steps are described below.

These two files contain the following:

"allresults.dta"
        Contains all decisions by subjects, each row is a decision screen. It contains the
following variables:
        ppnr                    subject id (unique within a session)
        round                   # round in the experiment
        game_no                 # decision situation
        game_type               game protocol type: SPD, TG or UG
        decision                actions or beliefs
        t                       T payoff in the current game protocol
        r                       R payoff in the current game protocol
        p                       P payoff in the current game protocol
        s                       S payoff in the current game protocol
        tijd                    decision time (in seconds)
        keuzea                  decision as a first mover (LEFT or RIGHT for actions, % choosing
RIGHT for beliefs). In the SPDs, LEFT = D, RIGHT = C; In the TGs, LEFT = N, RIGHT = I; In the
UGs, LEFT = E, RIGHT = U.
        keuzebl                 decision as a second mover if the first mover choses LEFT (SOUTH
or WEST for actions, % choosing SOUTH for beliefs). In the SPDs, SOUTH = D, WEST = C; In the TGs
and UGs this choice is not available.
        keuzebr                 decision as a second mover if the first mover choses RIGHT
(NORTH or EAST for actions, % choosing EAST for beliefs). In the SPDs, NORTH = C, EAST = D, In
the TGs, NORTH = G, EAST = K; In the UGs, NORTH = F, EAST = A.
        id                      subject id (unique across all sessions)

"allsubjects.dta"
        Contains all subjects specific data, each row is a subject. It contains the following
variables:
        ppnr                    subject id (unique within a session)
        session                 session number
        studie                  major field of study (econ = Economics or Business, law = Law,
psy = Psychology, anders = Other)
        sexe                    sex (male/female)
        age                     age (in years)
        datum                   date of the session
        tijdstempel             timestamp at the end of the experiment
        risk                    lottery choice in the Eckel-Grossman task
        loterijtijd             time (in seconds) spent on the Eckel-Grossman task
        attempts1               Number of attempts to successfully complete the first set of
quiz questions
        attempts2               Number of attempts to successfully complete the second set of
quiz questions
        totverdeuros            earnings (in euros)
        id                      subject id (unique across all sessions)


*******************************************************************************************
1. Combine all raw experimental data ******************************************************
*******************************************************************************************
Run the stata do-file "data_prep_hm.do". This will take the raw data from "data/allresults.dta"
and "data/allsubjects.dta" and produce the following output:

        "hm_all.dta"
        This file contains all experimental decisions, with one row for each subject/game
protocol combination

        "mnl_data.csv"
        This file contains the data used in the maximum likelihood estimations with subjective
beliefs run in Matlab (complete sample)
```

"mnl_data_re.csv"
This file contains the data used in the maximum likelihood estimations with rational expectations run in Matlab (complete sample)

"risk.csv"
This file contains the lottery choices in the Eckel-Grossman task, used in the individual estimations in Matlab when we allow for risk aversion

The do-file will also identify the "core sample" (i.e. alpha, beta, kappa, all smaller than 2 in absolute value). For this, one should first run the individual estimations in Matlab (see step 2 below). This will produce the following output:

"core.dta"
This file contains an indicator whether a participant is in the "core sample".

"mnl_data_core.csv"
This file contains the data used in the maximum likelihood estimations with subjective beliefs run in Matlab (core sample)

"mnl_data_re_core.csv"
This file contains the data used in the maximum likelihood estimations with rational expectations run in Matlab (core sample)

****
2. Run the individual estimations in Matlab **********
****
Run "hm_mnl_indiv.m". For the main analyses, enter the following settings at the top of the file.

```
%set a name for the estimation output
estimation_name='abc_rn';

%set risk neutral or not
risk_neutral=1;         %if set to zero, takes the individual lottery choices for crra utility
                        %if set to one, assumes risk neutrality
                        %if set to two, estimates r as a free parameter
                        %if set to three, assumes log-utility (r=1)

%set constraints
alpha_zero=0;           %sets alpha to zero
beta_zero=0;            %sets beta to zero
kappa_zero=0;           %sets kappa to zero
delta_zero=1;           %sets delta to zero
gamma_zero=1;           %sets gamma to zero

alpha_negative=0;       %choose -1 to set alpha positive, 1 to set alpha negative, 0 for no constraint on the sign
beta_negative=0;        %choose -1 to set beta positive, 1 to set beta negative, 0 for no constraint on the sign
kappa_negative=0;       %choose -1 to set kappa positive, 1 to set kappa negative, 0 for no constraint on the sign
delta_negative=0;       %choose -1 to set delta positive, 1 to set delta negative, 0 for no constraint on the sign
gamma_negative=0;       %choose -1 to set gamma positive, 1 to set gamma negative, 0 for no constraint on the sign
lambda_negative=-1;     %choose -1 to set lambda positive, 1 to set lambda negative, 0 for no constraint on the sign


%import choices from experiment
filename='mnl_data.csv';
dt=csvread(filename);
file2='risk.csv';       % contains the Eckel-Grossman lottery choices
risk=csvread(file2);

no_trials=18;
n = length(dt)/no_trials; % n is the number of subjects in the data

options = optimoptions(@fmincon,'MaxFunEvals',6000);
n_steps = 6;           % number of starting points used per parameter
```

This will produce an output file with parameter estimates for each individual (".../output/indiv/estimates/abc_rn.txt") and a MAT file (".../output/indiv/mat_files/abc_rn.mat"). Note that one should create these subfolders.

The settings above will produce the main estimates used in Figure 4 and Table 3 (for the core sample). All other models can be estimated by changing the above parameters accordingly (using subjective expectations). To run the estimations with rational expectations, change:

```
        %import choices from experiment
        filename='mnl_data.csv';
```

to

```
        %import choices from experiment
        filename='mnl_data_re.csv';
```

The file "hm_estimates.dta" contains all individual estimates reported in the paper. In the names of these models, we use "a" for models including alpha, "b" for "beta", "c" for kappa, "d" for delta, "e" for gamma. Models assuming risk neutrality are denoted by "_rn", models allowing for risk aversion are denoted using "_crra". For example, the variable "a_abc_rn" contains the individual estimate of alpha in the model allowing for alpha, beta, and kappa under risk neutrality. The variable "core" indicates whether the individual is part of the "core sample".

****
3. Run the aggregate estimations in Matlab **********
****
Run "hm_mixture_em.m". For the main analyses, enter the following settings (change the number of types accordingly):

```
        %set a name for the estimation output
        estimation_name='2t_abc_rn';

        %set number of types
        n_types=2;

        %set risk neutral or not
        risk_neutral=1;         %if set to zero, takes r=1 for crra utility
                                %if set to one, assumes risk neutrality
                                %if set to two, estimates r as a free parameter

        %set constraints
        alpha_zero=0;           % sets alpha to zero
        beta_zero=0;            % sets beta to zero
        kappa_zero=0;           % sets kappa to zero
        delta_zero=1;           % sets delta to zero
        gamma_zero=1;           % sets gamma to zero

        %settings for the EM-algorithm
        n_steps = 24;           % number of starting points used
        n_iterations = 50;      % maximal number of iterations
        ll_diff_stop = .01;     % difference in log-likelihood for convergence
        get_se = 1;             % set to 1 to get standard errors (long computing time)
        nr = 60;                % number of replications for bootstrapping standard errors

        %import choices from experiment
        filename='mnl_data_core.csv';
        dt=readmatrix(filename);
```

This will produce the following output files:

"../output/mixtures/estimates/2t_abc_rn.txt"
Contains the parameter estimates and standard errors for each type. One column for each type. Rows indicate estimates for (respectively) alpha, beta, kappa, delta, gamma, lambda, phi, and then standard errors for alpha, beta, kappa, delta, gamma, lambda, phi. If crra parameter r is estimated alongside the other parameters (set "risk_neutral=2"), the order is alpha, beta, kappa, delta, gamma, lambda, r, phi.

"../output/mixtures/fits/2t_abc_rn.txt"
Contains model fit measures. fit_results_1 = log likelihood, fit_results_2 = ICL, fit_results_3 = entropy measure, fit_results_4 = "Q-value" (used in the M-step).

"../output/mixtures/taus/2t_abc_rn.txt"
Contains the estimated "taus", for each individual (rows) and type (columns).

"../output/mixtures/mat_files/2t_abc_rn.mat"
All the output in a MAT file.

The settings above will produce the main estimates for the two-types model reported in Table 4. For reference, the files "Tab4_est_1t_abc_rn", "Tab4_est_2t_abc_rn", and "Tab4_est_3t_abc_rn", and "Tab4_fit_1t_abc_rn", "Tab4_fit_2t_abc_rn", and "Tab4_fit_3t_abc_rn" contain the parameter estimates and model fit measures of the models reported in Table 4. The underlying files for Tables 5 and 6 are also included. All "tau" estimates are included in the file "taus.dta". All

other finite mixture models can be estimated by changing the above parameters (using subjective expectations).

To run the estimations with rational expectations, change:

```
        %import choices from experiment
        filename='mnl_data_core.csv';
to
        %import choices from experiment
        filename='mnl_data_re_core.csv';
```

To include the complete sample (and not just the "core" sample), change:

```
        %import choices from experiment
        filename='mnl_data_core.csv';
to
        %import choices from experiment
        filename='mnl_data.csv';
```

****
4. Run the out-of-sample predictions in Matlab **********
****
For out-of-sample predictions at the individual level, run "hm_out_of_sample.m". Enter the following settings (change the desired parameters restrictions accordingly):

```
        %set a name for the estimation output
        estimation_name='abc_rn';

        %%%%%%%%%%%%%%%%%%%%%%%%
        %estimation settings
        %%%%%%%%%%%%%%%%%%%%%%%%

        %set risk neutral or not
        risk_neutral=1;         %if set to zero, takes the individual lottery choices for crra
utility
                                %if set to one, assumes risk neutrality
                                %if set to two, estimates rho as a free parameter

        %set constraints
        alpha_zero=0;           %sets alpha to zero
        beta_zero=0;            %sets beta to zero
        kappa_zero=0;           %sets kappa to zero
        delta_zero=1;           %sets delta to zero
        gamma_zero=1;           %sets gamma to zero
        pms_zero = [alpha_zero beta_zero kappa_zero delta_zero gamma_zero];

        alpha_negative=0;       %choose -1 to set alpha positive, 1 to set alpha negative, 0 for
no constraint on the sign
        beta_negative=0;        %choose -1 to set beta positive, 1 to set beta negative, 0 for
no constraint on the sign
        kappa_negative=0;       %choose -1 to set kappa positive, 1 to set kappa negative, 0 for
no constraint on the sign
        delta_negative=0;       %choose -1 to set delta positive, 1 to set delta negative, 0 for
no constraint on the sign
        gamma_negative=0;       %choose -1 to set gamma positive, 1 to set gamma negative, 0 for
no constraint on the sign
        lambda_negative=-1;     %choose -1 to set lambda positive, 1 to set lambda negative, 0
for no constraint on the sign
        pms_pos = [alpha_negative beta_negative kappa_negative delta_negative gamma_negative
lambda_negative];

        n_steps = 2;           % number of starting points used per parameter

        sts = [risk_neutral pms_zero pms_pos n_steps];

        %%%%
        % import choices from experiment
        filename='mnl_data.csv';
        dt=csvread(filename);
        file2='risk.csv';       % contains the Eckel-Grossman lottery choices
        risk=csvread(file2);
```

This will produce the following output files:

```
        ".../output/out_of_sample/predictions/abc_rn.txt"
        Each row gives the result for each individual and each game protocol.
```

```
        ".../output/out_of_sample/mat_files/abc_rn.mat"
        All the output in a MAT file.
```

For out-of-sample predictions at the aggregate level, run "hm_mixture_out_of_sample.m". Enter the following settings (change the desired parameters restrictions accordingly). As starting values, we used the mixture estimates based on all 18 game protocols:

```
        %set a name for the estimation output
        estimation_name='abc_2t_rn';

        %%%%%%%%%%%%%%%%%%%%%%%%%
        %estimation settings
        %%%%%%%%%%%%%%%%%%%%%%%%%

        %set number of types
        n_types=2;

        %set risk neutral or not
        risk_neutral=1;         %if set to zero, takes r=1 for crra utility
                                %if set to one, assumes risk neutrality
                                %if set to two, estimates r as a free parameter

        %set constraints
        alpha_zero=0;           %sets alpha to zero
        beta_zero=0;            %sets beta to zero
        kappa_zero=0;           %sets kappa to zero
        delta_zero=1;           %sets delta to zero
        gamma_zero=1;           %sets gamma to zero
        pms_zero = [alpha_zero beta_zero kappa_zero delta_zero gamma_zero];

        %set starting values
        starting_values_type1 = [0.12 0.37 0.10 0.00 0.00 8.45];
        starting_values_type2 = [0.18 0.01 0.10 0.00 0.00 4.25];
        starting_values_type3 = [0.00 0.00 0.00 0.00 0.00 0.00];

        %settings for the EM-algorithm
        n_steps = 1;            % number of starting points used
        n_iterations = 50;      % maximal number of iterations
        ll_diff_stop = .01;     % difference in log-likelihood for convergence

        sts = [n_types risk_neutral pms_zero n_steps n_iterations ll_diff_stop];

        if n_types == 1
            starting_values = starting_values_type1;
        elseif n_types == 2
            starting_values = [starting_values_type1 starting_values_type2];
        elseif n_types == 3
            starting_values = [starting_values_type1 starting_values_type2
starting_values_type3];
        end

        %import choices from experiment
        filename='mnl_data_core.csv';
        dt=readmatrix(filename);
```

This will produce the following output files:

```
        ".../output/out_of_sample/predictions/abc_2t_rn.txt"
        Each row gives the result for each individual and each game protocol.

        ".../output/out_of_sample/mat_files/abc_2t_rn.mat"
        All the output in a MAT file.
```

The file "predictions.dta" contains all out-of-sample predictions reported in the paper.

```
*******************************************************************************
5. Prepare all tables, figures and tests ******************************************
*******************************************************************************
```

Run the Stata do-file "analysis_hm.do". This will produce all tables, figures and tests reported in the paper, except the tables reporting the estimates of the finite mixture models. To create these tables, copy-paste the estimates and model fit criteria from the respective TXT files (see step 3 above).