```matlab
%% Project on Opinion Formation for the course
%% "Modelling and Simulating of Social Systems with MATLAB"
%% author: The Opinionators (Elisa Wall, Alexander Stein, Niklas Tidbury)

%% number of time steps
T = 500;

%% number of iterations
Tg = 50;

%% number of society agents
N = 1000;

%% Properties of the SocietyAgents
% The threshold u defines when two agents speak/interact with each other
u = 0.32;


% Mu defines the change of opinion when two agents speak with each other
% mu has to be between 0 and 1 to ensure that all opinions are
% opinions are between 0 and 1.
mu = 0.3;


%% Properties of the extremists
% number of extremists
n0 = 1;
n1 = 1;
% number of agents one extremist can reach
p0 = 500;
p1 = 500;
% An extremist convinces an agent with probability kappa
kappa0 = 0.2;
kappa1 = 0.2;
% an extremist has a range of people he reaches
% The extremist with opinion 0 can reach all agents with opinion in
% [0,infop0], repectively extremists with opinion 1 to [infop1, 1]
infop0 = 0.3;
infop1 = 1-infop0;


%% run the program


%% Figure 1
gen_plot("hist", false, 3, run_simulation("without", create(N), Tg, 30, N, u, mu, n0,↙
p0, kappa0, n1, p1, kappa1, infop0, infop1), "µ = 0.3, w/o E", "Opinion", "Number of↙
Agents", 30, N, false);
gen_plot("hist", false, 3, run_simulation("without", create(N), Tg, 200, N, u, 0.03,↙
n0, p0, kappa0, n1, p1, kappa1, infop0, infop1), "µ = 0.03, w/o E", "Opinion", "Number↙
of Agents", 200, N, false);

%% Figure 2
gen_plot_interval("line", "% of opinion between 0.45 and 0.55, w/o E", "µ",↙
"Percentage", false, "without", "mu", create(N), Tg, 15, N, u, mu, n0, p0, kappa0, n1,↙
p1, kappa1, infop0, infop1);

%% Figure 3
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 30, N, u, mu, n0, 50,↙
kappa0, n1, 50, kappa1, infop0, infop1), "µ = 0.3, w/ E, p = 50", "Opinion", "Number↙
of Agents", 30, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 200, N, u, 0.03, n0,↙
50, kappa0, n1, 50, kappa1, infop0, infop1), "µ = 0.03, w/ E, p = 50", "Opinion",↙
"Number of Agents", 200, N, false);

%% Figure 4
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 5000, N, u, mu, n0,↙
p0, kappa0, n1, p1, kappa1, infop0, infop1), "µ = 0.3, w/ E, p = 500", "Opinion",↙
"Number of Agents", 5000, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 5000, N, u, 0.03, n0,↙
p0, kappa0, n1, p1, kappa1, infop0, infop1), "μ = 0.03, w/ E, p = 500", "Opinion",↙
"Number of Agents", 5000, N, false);

%% Figure 5
gen_plot_interval("line", "% of opinion between 0.45 and 0.55, w/ E", "μ",↙
"Percentage", false, "with", "mu", create(N), Tg, 15, N, u, mu, n0, 50, kappa0, n1,↙
50, kappa1, infop0, infop1);

%% Figure 6
% infop = 0.1
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
5, kappa0, n1, 5, kappa1, 0.1, 0.9), "w/ E, infop = 0.1, p = 5", "Opinion", "Number of↙
Agents", 1000, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
10, kappa0, n1, 10, kappa1, 0.1, 0.9), "w/ E, infop = 0.1, p = 10", "Opinion", "Number↙
of Agents", 1000, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
20, kappa0, n1, 20, kappa1, 0.1, 0.9), "w/ E, infop = 0.1, p = 20", "Opinion", "Number↙
of Agents", 1000, N, false);

% infop = 0.3
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
5, kappa0, n1, 5, kappa1, 0.3, 0.7), "w/ E, infop = 0.3, p = 5", "Opinion", "Number of↙
Agents", 1000, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
10, kappa0, n1, 10, kappa1, 0.3, 0.7), "w/ E, infop = 0.3, p = 10", "Opinion", "Number↙
of Agents", 1000, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
20, kappa0, n1, 20, kappa1, 0.3, 0.7), "w/ E, infop = 0.3, p = 20", "Opinion", "Number↙
of Agents", 1000, N, false);

% infop = 0.5
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
5, kappa0, n1, 5, kappa1, 0.5, 0.5), "w/ E, infop = 0.5, p = 5", "Opinion", "Number of↙
Agents", 1000, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
10, kappa0, n1, 10, kappa1, 0.5, 0.5), "w/ E, infop = 0.5, p = 10", "Opinion", "Number↙
of Agents", 1000, N, false);
gen_plot("hist", false, 3, run_simulation("with", create(N), Tg, 1000, N, u, mu, n0,↙
20, kappa0, n1, 20, kappa1, 0.5, 0.5), "w/ E, infop = 0.5, p = 20", "Opinion", "Number↙
of Agents", 1000, N, false);

%% Figure 7
gen_plot_interval("line", "% of extreme opinions", "p", "Percentage", true, "with",↙
"p", create(N), Tg, 10000, N, u, mu, n0, p0, kappa0, n1, p1, kappa1, infop0, infop1);




%% Functions


%% Function for generating plots
% plot_type = plot type (hist or line)
% slider_bool = when number_of_plots is 1, the option to add a slider is
% given
% number_of_plots = number of data sets in plot
% data = data from simulation (call run_simulation())
% plot_name = name window of plot
% T = entire time (must be same T as passed to run_simulation())
% N = society size (must be same N as passed to run_simulation())
% save = activate / deactivate saving plot as png in folder "exports"
function [] = gen_plot(plot_type, slider_bool, number_of_plots, data, plot_name,↙
x_axis, y_axis, T, N, save)
    figure('name', plot_name);
    disp("Running...");
    if plot_type == "hist"
        edges = linspace(0,1,200);
        if number_of_plots > 1
            for i = 1:number_of_plots
                histdata = data(round(i*T/number_of_plots),:);
                histogram(histdata, edges, 'DisplayName', ['T = ', num2str(round↙
(i*T/number_of_plots))]);
                hold on;
            end
        else
            slmin = 1;
            slmax = T;
            histogram(data(T,:), edges, 'DisplayName', ['T = ', num2str(T)]);
            if slider_bool
                hsl = uicontrol('Style','slider','Min',slmin,'Max',slmax,...
                    'SliderStep',[1 1]./(slmax-slmin),'Value',1,...
                    'Position',[50 10 500 10],'FontSize', 25);
                set(hsl,'Callback',@(hObject,eventdata) histogram(data(round(get↙
(hObject,'Value')),:), edges, 'DisplayName', ['T = ', num2str(round(get↙
(hObject,'Value')))]));
            end
        end
        legend('show');
    elseif plot_type == "line"
        tot_perc = zeros(T);
        for i = 1:T
           tot_perc(i) = countPercentage(0,0.1,data(i,:),N);
           if tot_perc(i) == 100
               disp(["100% at T: ", num2str(i)]);
           end
        end
        plot(tot_perc);
    end
    title({' ', plot_name, ' '}, 'FontSize', 25);
    xlabel(x_axis, 'FontSize', 25);
    ylabel(y_axis, 'FontSize', 25);
    % because data is lost for large Ts and log scaling
    if T < 400
        set(gca,'yscale','log');
    end
    % save to file with unique name
    if save
       format shortg;
       c = clock;
       fix(c);
       filename = sprintf("exports_second/gen_plot_%d%d%d%d%d%d.png",c(1),c(2),c(3),c↙
(4),c(5),c(6));
       saveas(gcf,filename);
    end
    disp("Finished!");
end


%% Function for generating plots over interval of a certain var (can be customized)
% plot_type= plot type (hist or line)
% plot_name = name window of plot
% param = string name of variable to generate and plot over
% other vars same as usual
function [] = gen_plot_interval(plot_type, plot_name, x_axis, y_axis, save, simtype,↙
param, op, Tg, T, N, u, mu, n0, p0, kappa0, n1, p1, kappa1, infop0, infop1)
    FIG = figure('name', plot_name);
    disp("Running...");
    if param == "p"
        if plot_type == "line"
            perc_total = zeros(p0, 1);
            for j = 1:p0
                plot(perc_total);
                arr = run_simulation(simtype, op, Tg, T, N, u, mu, n0, j, kappa0, n1,↙
j, kappa1, infop0, infop1);
                perc_total(j) = countPercentage(0, 0.1, arr(T,:), N);
                if perc_total(j) == 100
                    disp(["100% at p0: ", num2str(j)]);
                end
                pause(0.0001);
                drawnow;
            end
        end
    elseif param == "u"
        if plot_type == "line"
            perc_total = zeros(50, 1);
            for j = 1:100
                plot(perc_total);
                arr = run_simulation(simtype, op, Tg, T, N, j/100, mu, n0, p0, kappa0,↙
n1, p1, kappa1, infop0, infop1);
                perc_total(j) = countPercentage(0.45, 0.55, arr(T,:), N);
                if perc_total(j) == 100
                    disp(["100% at p0: ", num2str(j)]);
                end
                pause(0.0001);
                drawnow;
            end
        end

    elseif param == "mu"
        if plot_type == "line"
            perc_total = zeros(50,1);
            for j = 1:50
                plot(perc_total);
                arr = run_simulation(simtype, op, Tg, T, N, u, j/100, n0, p0, kappa0,↙
n1, p1, kappa1, infop0, infop1);
                perc_total(j) = countPercentage(0.45, 0.55, arr(T,:), N);
                if perc_total(j) == 100
                    disp(["100% at mu: ", num2str(j)]);
                end
                pause(0.0001);
                drawnow;
            end
        end
    end
    title({' ', plot_name, ' '}, 'FontSize', 25);
    xlabel(x_axis, 'FontSize', 25);
    ylabel(y_axis, 'FontSize', 25);
    if param == "u"
        % set x axis labelling for u
        xticks([0 10 20 30 40 50 60 70 80 90 100]);
        xticklabels({'0','0.1','0.2','0.3','0.4','0.5','0.6', '0.7', '0.8', '0.9',↙
'1'});
    elseif param == "mu"
        % set x axis labelling for mu
        xticks([0 10 20 30 40 50]);
        xticklabels({'0','0.1','0.2','0.3','0.4','0.5'});
    end
    disp("Finished!");
    % save file with unique name
    if save
       format shortg;
       c = clock;
       fix(c);
       filename = sprintf("exports_second/gen_plot_intervall_%d%d%d%d%d%d.png",c(1),c↙
(2),c(3),c(4),c(5),c(6));
       saveas(gcf,filename);
    end
end


%% Function for calculating average over several simulated societies and plotting them↙
```

```
accordingly
% param = with / without extremists
% number_of_plots: number of data sets in plot
% data = data from simulation (call run_simulation())
% plot_name = name window of plot
% plot_type = type of plot
% x_axis / y_axis = axis labelling
% other vars = as usual
% save = activate / deactivate saving plot as png in folder "exports"
function [] = gen_av_plot(param, number_of_plots, plot_type, plot_name, x_axis,↙
y_axis, Tg, T, N, u, mu, n0, p0, kappa0, n1, p1, kappa1, infop0, infop1, save)
    av_matrix = zeros(T, N, Tg);
    for j = 1:Tg
        av_matrix(:,:,j) = run_simulation(param, create(N), Tg, T, N, u, mu, n0, p0,↙
kappa0, n1, p1, kappa1, infop0, infop1);
    end
    size(av_matrix)
    av = mean(av_matrix,3);
    gen_plot(plot_type, number_of_plots, av, plot_name, x_axis, y_axis, T, N, save);
end


%% Running a round of simulation
% returns an T x N matrix of data from the simulation, so we have the data
% from each given time step for analysis
function [data] = run_simulation(param, op, Tg, T, N, u, mu, n0, p0, kappa0, n1, p1,↙
kappa1, infop0, infop1)
    if param == "without"
        data = without(op, T, N, u, mu);
    elseif param == "with"
        data = with(op, T, N, u, mu, n0, p0, kappa0, n1, p1, kappa1, infop0, infop1);
    end
end


%% Creating the society
% Input: number of extremists
% Output: (1xN) opinion matrix of an arbitrary random distribution
%        (gaussian or uniform distribution)
function [op] = create(N)
    %% Creating the society
    % A society of N SocietyAgents with opinions op in [0,1] that are randomly
    %        distributed

    % uniform distribution
    op = rand(1,N);
end


%% The program without extremists as given in the paper
% Input: T, N, u, mu
% Output: updated opinion
function [simulation] = without(op, T, N, u, mu)
    %%  A world without extrimists
    %   The influence of a single agent in a singlte timestep t is defined in the
    %   function SocietyAgent. We raise up the time steps to T. In every
    %   time step t, every agent has the chance to speak with another.

    simulation = zeros(T, N);
    simulation(1,:) = op;
    for t = 1:T+1
        for i = 1:N
            [op0, op1, k] = SingleAgent(op(i), op, u, N, mu);
            op(i) = op0;
            op(k) = op1;
        end
        simulation(t+1, :) = op;
    end
```

```matlab
end


%% The program with extremists
% Input: T, N, mu, n0, n1, p0, p1, kappa0, kappa1, infop0, infop1
% Output: histograms
function [simulation] = with(op, T, N, u, mu, n0, p0, kappa0, n1, p1, kappa1, infop0,↙
infop1)
    %% Effective number of influenced people by the extremists
    %   All agents have the same behavior, so we can sum up the influence of all
    %   agents in the number of people that get influenced

    simulation = zeros(T, N);
    neff0 = p0 * n0;
    neff1 = p1 * n1;

    % add start state to data matrix
    simulation(1,:) = op;

    %% A world with extremists
    for t = 1:T+1
        % For timestep t; the SocietyAgents play their game
        for i = 1:N
            [op0, op1, k] = SingleAgent(op(i), op, u, N, mu);
            op(i) = op0;
            op(k) = op1;
        end
        % For timestep t; the extremists with opinion 0 play their game
        for i = 1:neff0
            r = rand;
            k = randi(N);
            counter = 0;
            % extremists with opinion 0 only reach agents with similar opinion
            while op(k) > infop0 && counter < N
                k = randi(N);
                counter = counter + 1;
            end
            if r < kappa0
                op(k) = 0;
            end
        end
        % For timestep t; the extremists with opinion 1 play their game
        for i = 1:neff1
            r = rand;
            k = randi(N);
            % extremists with opinion 1 only reach agents with similar opinion
            while op(k) < infop1 && counter < N
                k = randi(N);
                counter = counter + 1;
            end
            if r < kappa1
                op(k) = 1;
            end
        end

        % add current state to data matrix
        simulation(t+1, :) = op;
    end
end


%% Defining the influence of a single SocietyAgent during one timestep t
% Input: op0 = opinion of a single agent, op = opinion of the society,
%       mu, u and N as described above
% Output: new opinion of op0 (opnew0),
%       new opinion of a randomly chosen op1 (opnew1) that interacted with
%       op0 and the position of op1 (pos)
function [opnew0, opnew1, pos] = SingleAgent(op0, op, u, N, mu)
```

```matlab
    % op0 meets a randomly chosen agent in the society op, called op1
    pos = randi(N);
    op1 = op(pos);
    if abs(op1 - op0) < u
        % weighted difference of opinions, weigh is given by mu
        opnew0 = op0 + mu*(op1-op0);
        opnew1 = op1 + mu*(op0-op1);
    else
        opnew0 = op0;
        opnew1 = op1;
    end
end


%% Calculate percentage of opinions in certain interval (extremist, 0-0.1, 0.9-1)
function [perc] = countPercentage(lower, upper, op, N)
  counter = 0;
  for i = 1:N
      % add to counter if opinion in interval
       if (op(i) >= lower && op(i) <= upper) || (op(i) >= 1-upper && op(i) <= 1-lower)
           counter = counter + 1;
       end
  end
  % calculate percentage
  perc = counter/N*100;
end

%% Create movie over T from histogram data
function [] = createMovie(data, title, T)
    edges = linspace(0,1,200);
    vidObj = VideoWriter(title,'MPEG-4');
    vidObj.FrameRate = 20;
    open(vidObj);
    % iterate through data along T
    for t = 1:T
        histogram(data(t,:), edges, 'DisplayName', ['T = ', num2str(T)]);
        title({' ', plot_name, ' '}, 'FontSize', 25);
        xlabel(x_axis, 'FontSize', 25);
        ylabel(y_axis, 'FontSize', 25);
        pause(0.005);
        % draw plot and frame
        drawnow;
        F = getframe(FIG);
        writeVideo(vidObj,F);
    end
    close(vidObj);
end
```