# Appendix A

## Pre-registered Matlab Script for Confirmatory Analyses

```
% This code is based on code written by Edward Vul,
% except for the computation of the confidence intervals for the
% effect sizes, which is based on code provided by Simonsohn  (2013).
data;
% The data are 2 nx16 matrices (the first matrix with data from
% immediate condition; the second matrix with data from the the ...
    delayed
% condition), where n is the sample size of the corresponding ...
    condition.
% Each row corresponds to the answers from one participant. The ...
    first 8
% columns correspond to the first guesses; the final 8 colums ...
    correspond to
% the second guesses.

answers = [6.3 43.3 32.3 13.4 53.6 54.8 26.4 22.4];
% These answers are derived from The World Factbook (Central ...
    Intelligence
% Agency, 2013)

%% sets are 1: immediate, 2: delayed

for set = [1:2]
    [si sj] = size(data{set});
    grp(set).n=si;
    % guess 1
    grp(set).g{1} = data{set}(:,[1:8]);
    % guess 2
```

```
    grp(set).g{2} = data{set}(:,[9:16]);
    % average of guesess
    grp(set).g{3} = (grp(set).g{1}+grp(set).g{2})./2;

    for g = [1:3]
        %[n k] = size(grp(set).g{1});
        bigans = repmat(answers, grp(set).n, 1);
        % mean squared error
        grp(set).d{g} = mean((grp(set).g{g}-bigans).^2,2);
        grp(set).mumse(g) = mean(grp(set).d{g});
        grp(set).semse(g) = std(grp(set).d{g})./sqrt(grp(set).n);
    end
end

%% graph of mean MSE guess 1, guess2, average guess

figure();
barweb([grp(1).mumse; grp(2).mumse], [grp(1).semse; grp(2).semse]);
ylim([400 700]);

%% comparisons between guess and average

for set = [1:2]
    for g = [1:2]
    % guess 1 or guess 2 compared to average.
    % ttest
    [h p ci stats] = ttest(grp(set).d{g} - grp(set).d{3});
    grp(set).t{g} = stats.tstat;
    grp(set).p{g} = p;
    % effect size
    grp(set).dz{g} = grp(set).t{g}/sqrt(grp(set).n);
    % confidence interval (see Cumming and Finch (2001, pp. 544-545))
    alpha=.05;
    df = grp(set).n-1;
```

```
    tnonct = inline('nctcdf(x,df,delta) - pr');
    ncp_low = fzero(@(delta) tnonct(delta, df, 1-alpha/2, ...
        grp(set).t{g}), [-20,20]);
    ncp_high = fzero(@(delta) tnonct(delta, df, alpha/2, ...
        grp(set).t{g}), [-20,20]);
    grp(set).dzlow{g} = ncp_low/sqrt(grp(set).n);
    grp(set).dzhigh{g} = ncp_high/sqrt(grp(set).n);
    end

    % guess 1 compared to guess 2.
    [h3 p3 ci3 stats3] = ttest(grp(set).d{1} - grp(set).d{2});
end


%% comparison between magnitude of averaging benefit over guess 1.
[h4 p4 ci4 stats4] = ttest2(grp(2).d{1} - grp(2).d{3}, ...
    grp(1).d{1} - grp(1).d{3});
```

# Appendix B

## Used Matlab Script for Confirmatory Analyses

```
% This is the code used for the confirmatory analyses.
% Apart from minor syntactic clean-up, this code differs from the
% pre-registered code in Appendix A in that we added code for
% 1) descriptive statistics
% 2) the scatter histogram plots in Figure 2
% 3) some effect sizes
% 4) some confidence intervals for effect sizes.
% Further, the appearance of the bar chart (Figure 1) is changed.

% The data file (data_steegenetal2014.mat) is made from several ...
    .txt files,
% using the code from steegenetal2014_postprocessing.m

% This code is based on code written by Ed Vul,
% except for the computation of the confidence intervals for the
% effect sizes, which is based on code provided by Simonsohn  (2013).
clear all
close all
load data_steegenetal2014;
% The data are 2 nx16 matrices (the first matrix with data from
% immediate condition; the second matrix with data from the the ...
    delayed
% condition), where n is the sample size of the corresponding ...
    condition.
% Each row corresponds to the answers from one participant. The ...
    first 8
% columns correspond to the first guesses; the final 8 colums ...
    correspond to
% the second guesses.
```

```

% The code relies on barweb.m, which is available on matlabcentral

% Steegen, S., Dewitte, L., Tuerlinckx, F., & Vanpaemel, W. ...
    (2014). Measuring the crowd within again: A pre-registered ...
    replication study. Frontiers in Psychology

answers = [6.3 43.3 32.3 13.4 53.6 54.8 26.4 22.4];
% These answers are derived from The World Factbook (Central ...
    Intelligence
% Agency, 2013)

% answers = [6.3 44.4 30.3 10.5 58 72.4 18.9 20.3]; these are the ...
    answers
% used in Vul & Pashler (2008) (see first colomn Table 3)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% compute MSE, descriptive statistics and plot data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% sets are 1: immediate, 2: delayed

for set = [1:2]
    [si sj] = size(data{set});
    grp(set).n=si;
    % guess 1
    grp(set).g{1} = data{set}(:,[1:8]);
    % guess 2
    grp(set).g{2} = data{set}(:,[9:16]);
    % average of guesses
    grp(set).g{3} = (grp(set).g{1}+grp(set).g{2})./2;

    for g = [1:3]
        bigans = repmat(answers, grp(set).n, 1);
```

```
        % mean squared error
        grp(set).d{g} = nanmean((grp(set).g{g}-bigans).^2,2);
        grp(set).mumse(g) = nanmean(grp(set).d{g});
        grp(set).semse(g) = nanstd(grp(set).d{g})./sqrt(grp(set).n);
    end
end

% descriptive statistics
for set = [1:2]
grp(set).sdmse = grp(set).semse.*sqrt(grp(set).n);
    for g=[1:2]
        grp(set).corr{g}=corr(grp(set).d{g}, grp(set).d{3});
        grp(set).mudiff{g} = mean(grp(set).d{g} - grp(set).d{3});
        grp(set).sddiff{g} = std(grp(set).d{g} - grp(set).d{3});
    end
end

%% graph of mean MSE guess 1, guess2, average guess
figure();
barweb([grp(1).mumse; grp(2).mumse], [grp(1).semse; ...
    grp(2).semse], [],{'Immediate'; 'Delayed '},[],[],'Mean ...
    Squared Error',[0 0 0; 1 1 1; .5 .5 .5],[],{'Guess 1'; 'Guess ...
    2'; 'Average'},'WestEast');
ylim([0 850]);
legend({'Guess 1'; 'Guess 2'; 'Average'},'Location','Northeast')

% scatter histogram plots
figure();
scatterhist(grp(1).d{1},grp(1).d{2}, 'Location', 'NorthEast', ...
    'Direction', 'Out', 'Color', 'k')
xlabel('MSE guess 1', 'FontSize', 12)
ylabel('MSE guess 2', 'FontSize', 12)
figure
```

```
scatterhist(grp(2).d{1}, grp(2).d{2}, 'Location', 'NorthEast', ...
    'Direction', 'Out', 'Color', 'k')
xlabel('MSE guess 1', 'FontSize', 12)
ylabel('MSE guess 2', 'FontSize', 12)


%%%%%%%%%%%%%%%%%%%
% inferential tests
%%%%%%%%%%%%%%%%%%%

alpha=.05;

%% comparisons between guess and average and between guess 1 and ...
    guess 2

for set = [1:2]
    % guess 1 or guess 2 compared to average.
    for g = [1:2]
    % ttest
    [h p ci stats] = ttest(grp(set).d{g} - grp(set).d{3});
    grp(set).t{g} = stats.tstat;
    grp(set).df{g} = stats.df;
    grp(set).p{g} = p;
    % effect size
    grp(set).dz{g} = grp(set).t{g}/sqrt(grp(set).n);
    % confidence interval (see Cumming and Finch (2001, pp. 549-550))
    df = grp(set).df{g};
    tnonct = inline('nctcdf(x,df,delta) - pr');
    ncp_low = fzero(@(delta) tnonct(delta, df, 1-alpha/2, ...
        grp(set).t{g}), [-20,20]);
    ncp_high = fzero(@(delta) tnonct(delta, df, alpha/2, ...
        grp(set).t{g}), [-20,20]);
    grp(set).dzlow{g} = ncp_low/sqrt(grp(set).n);
    grp(set).dzhigh{g} = ncp_high/sqrt(grp(set).n);
```

```
    end

    % guess 1 compared to guess 2.
    % ttest
    [h p ci stats] = ttest(grp(set).d{1} - grp(set).d{2});
    grp(set).t{3} = stats.tstat;
    grp(set).df{3} = stats.df;
    grp(set).p{3} = p;
    % effect size
    grp(set).dz{3} = grp(set).t{3}/sqrt(grp(set).n);
    % confidence interval
    df = grp(set).df{3};
    tnonct = inline('nctcdf(x,df,delta) - pr');
    ncp_low = fzero(@(delta) tnonct(delta, df, 1-alpha/2, ...
        grp(set).t{3}), [-20,20]);
    ncp_high = fzero(@(delta) tnonct(delta, df, alpha/2, ...
        grp(set).t{3}), [-20,20]);
    grp(set).dzlow{3} = ncp_low/sqrt(grp(set).n);
    grp(set).dzhigh{3} = ncp_high/sqrt(grp(set).n);
end


%% comparison of magnitude of averaging benefit over guess 1 ...
    between immediate and delayed condtion.
% ttest
[h p ci stats] = ttest2(grp(2).d{1} - grp(2).d{3}, grp(1).d{1} - ...
    grp(1).d{3});
grpcmp.t{1} = stats.tstat;
grpcmp.df{1} = stats.df;
grpcmp.p{1} = p;
% effect size (cohen's standardized mean difference d for ...
    independent groups; see Cumming and Finch (2001, pp. 567)
grpcmp.d{1} = grpcmp.t{1}.*sqrt((1/grp(1).n)+(1/grp(2).n));
% confidence interval (see Cumming and Finch (2001, pp. 567))
```

```
df = grpcmp.df{1};
tnonct = inline('nctcdf(x,df,delta) - pr');
ncp_low = fzero(@(delta) tnonct(delta, df, 1-alpha/2, ...
    grpcmp.t{1}), [-20,20]);
ncp_high = fzero(@(delta) tnonct(delta, df, alpha/2, ...
    grpcmp.t{1}), [-20,20]);
grpcmp.dlow{1} = ncp_low.*sqrt((1/grp(1).n)+(1/grp(2).n));
grpcmp.dhigh{1} = ncp_high.*sqrt((1/grp(1).n)+(1/grp(2).n));
```