Using Simulation to Guide Your Research

HARVARD
UNIVERSITY

Roger Strong
Harvard University

Vision Sciences
Laboratory

Why Use Simulation?

Simple answer: often the easiest way to solve a problem

Toy Example: The Birthday Problem

- How many randomly chosen people are needed for there to be > 50% chance that two of them have the same birthday?

The Birthday Problem

No Simulation

Person 1 has a random birthday

Odds person 2 has a different birthday:
    364/365

Odds person 3 has a different birthday:
    363/365

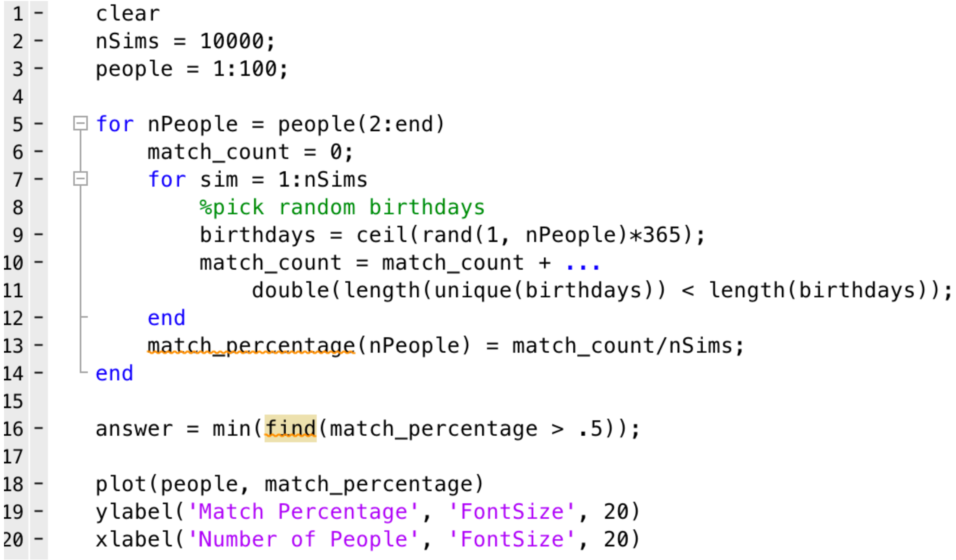
Odds person *n* has a different birthday:
    (365 − *n* + 1) / 365

Odds no matches after 3 people:
    364/365 * 363/365

Odds no matches after n people:

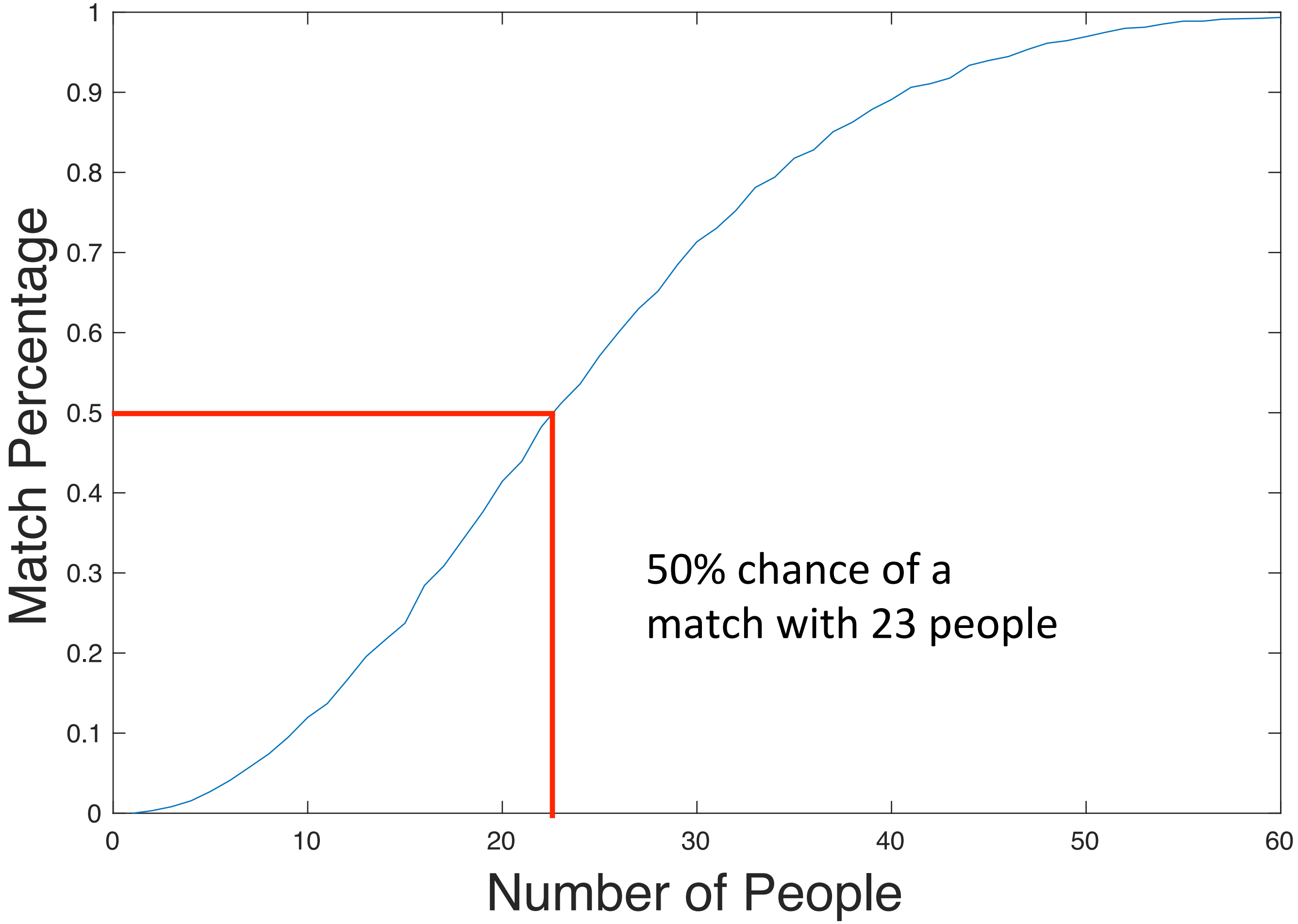
$$\frac{365!}{365^n(365-n)!}$$

Simulation



```
clear
nSims = 10000;
people = 1:100;

for nPeople = people(2:end)
    match_count = 0;
    for sim = 1:nSims
        %pick random birthdays
        birthdays = ceil(rand(1, nPeople)*365);
        match_count = match_count + ...
            double(length(unique(birthdays)) < length(birthdays));
    end
    match_percentage(nPeople) = match_count/nSims;
end

answer = min(find(match_percentage > .5));

plot(people, match_percentage)
ylabel('Match Percentage', 'FontSize', 20)
xlabel('Number of People', 'FontSize', 20)
```

The Birthday Problem



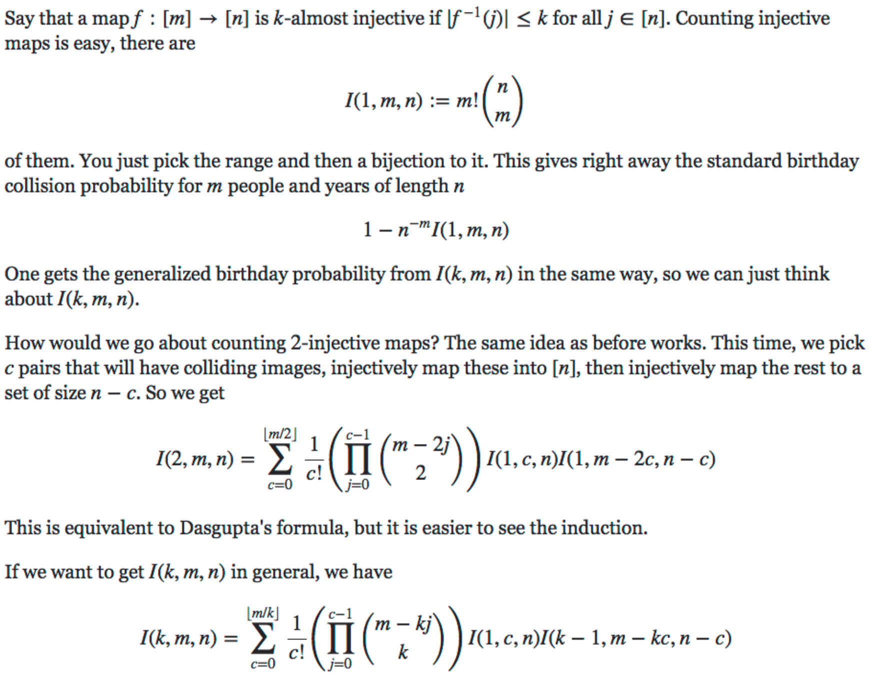
50% chance of a
match with 23 people

The Birthday Problem Extension

- How many randomly chosen people are needed for there to be > 50% chance that THREE of them have the same birthday?

The Birthday Problem Extension

No Simulation



Say that a map $f:[m]\to[n]$ is k-almost injective if $|f^{-1}(j)|\le k$ for all $j\in[n]$. Counting injective maps is easy, there are

$$I(1,m,n) := m!\binom{n}{m}$$

of them. You just pick the range and then a bijection to it. This gives right away the standard birthday collision probability for m people and years of length n

$$1-n^{-m}I(1,m,n)$$

One gets the generalized birthday probability from $I(k,m,n)$ in the same way, so we can just think about $I(k,m,n)$.

How would we go about counting 2-injective maps? The same idea as before works. This time, we pick c pairs that will have colliding images, injectively map these into $[n]$, then injectively map the rest to a set of size $n-c$. So we get

$$I(2,m,n)=\sum_{c=0}^{\lfloor m/2\rfloor}\frac{1}{c!}\left(\prod_{j=0}^{c-1}\binom{m-2j}{2}\right)I(1,c,n)I(1,m-2c,n-c)$$

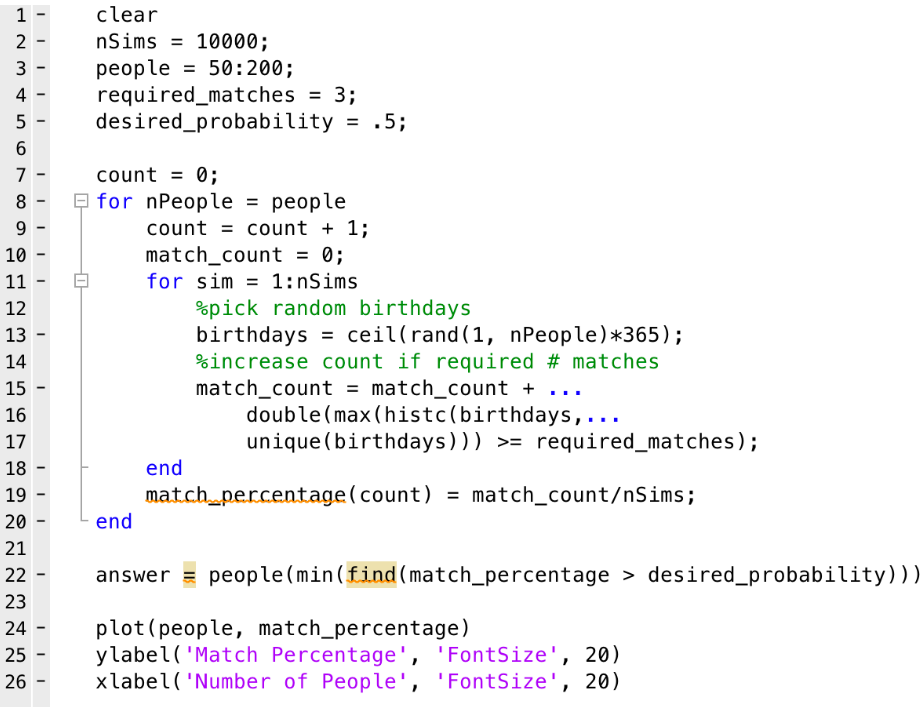
This is equivalent to Dasgupta's formula, but it is easier to see the induction.

If we want to get $I(k,m,n)$ in general, we have

$$I(k,m,n)=\sum_{c=0}^{\lfloor m/k\rfloor}\frac{1}{c!}\left(\prod_{j=0}^{c-1}\binom{m-kj}{k}\right)I(1,c,n)I(k-1,m-kc,n-c)$$

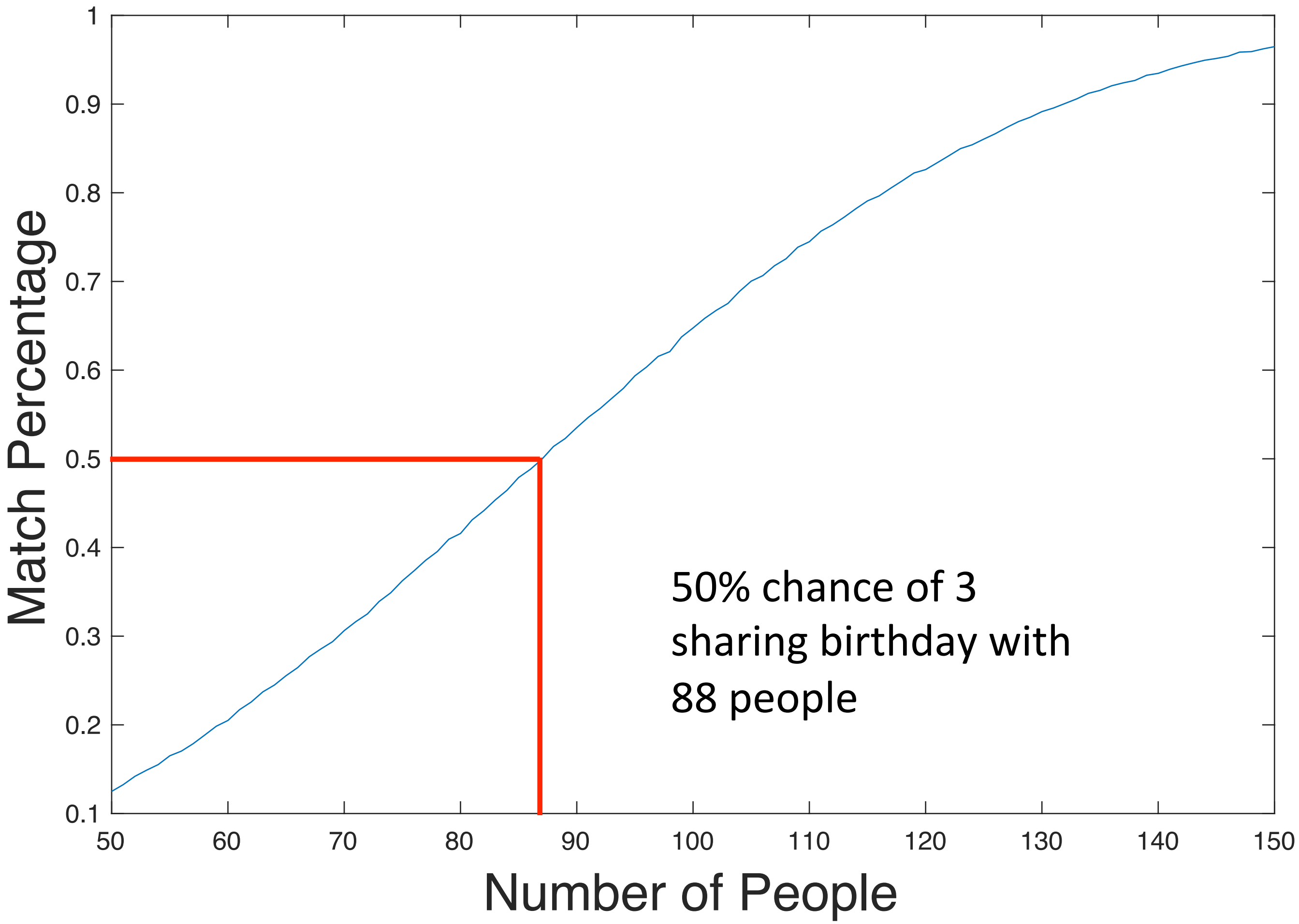
https://math.stackexchange.com/questions/25876/probability-of-3-people-in-a-room-of-30-having-the-same-birthday/25880#25880

Simulation



```
clear
nSims = 10000;
people = 50:200;
required_matches = 3;
desired_probability = .5;

count = 0;
for nPeople = people
    count = count + 1;
    match_count = 0;
    for sim = 1:nSims
        %pick random birthdays
        birthdays = ceil(rand(1, nPeople)*365);
        %increase count if required # matches
        match_count = match_count + ...
            double(max(histc(birthdays,...
            unique(birthdays))) >= required_matches);
    end
    match_percentage(count) = match_count/nSims;
end

answer = people(min(find(match_percentage > desired_probability)))

plot(people, match_percentage)
ylabel('Match Percentage', 'FontSize', 20)
xlabel('Number of People', 'FontSize', 20)
```

The Birthday Problem Extension



50% chance of 3 sharing birthday with 88 people

Using Simulation in Research

1. Guiding Experimental Design

2. Understanding Results

3. Your examples?

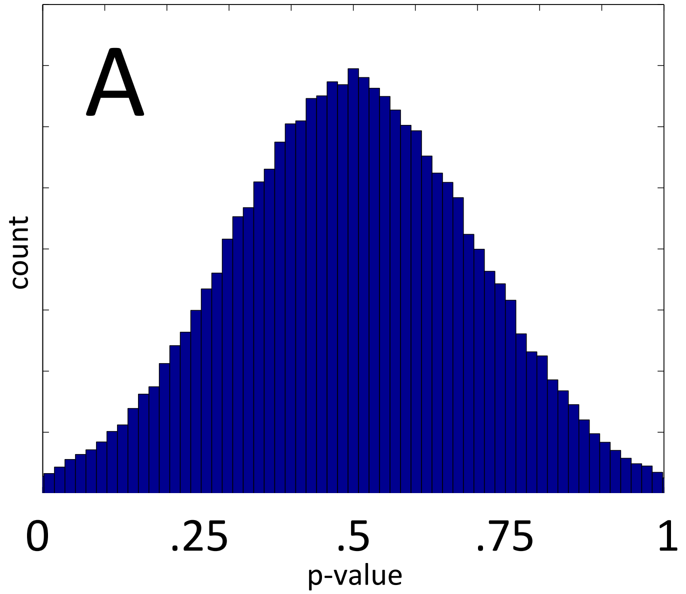
Using Simulation in Research

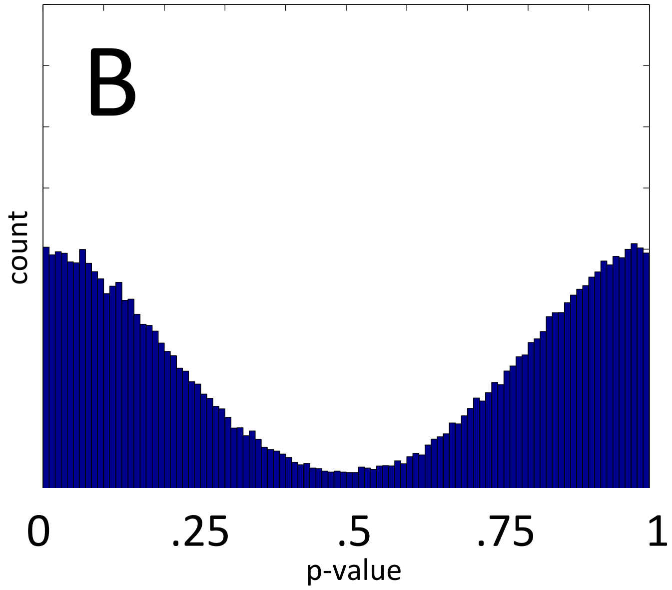
1. Guiding Experimental Design
    - Modeling false positive rates
    - Choosing exclusion criteria
    - Power analysis

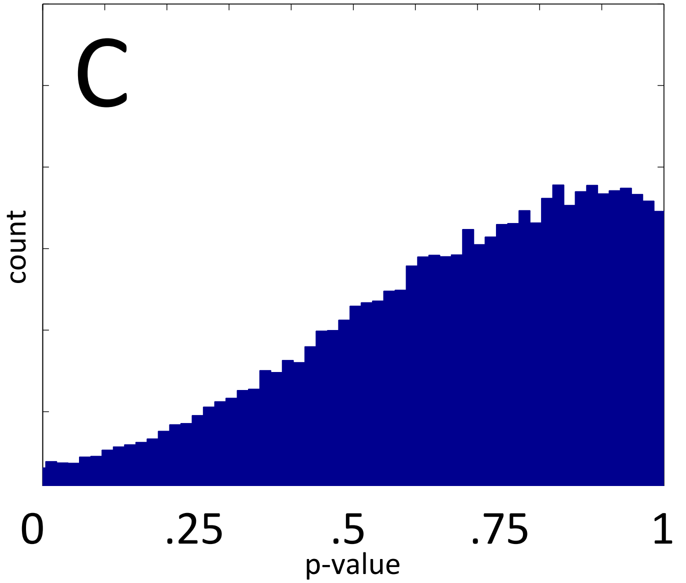
Using Simulation in Research

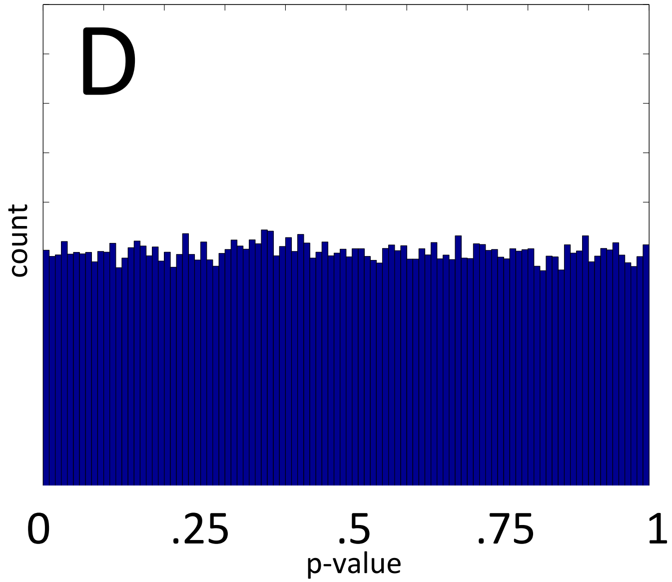
1. Guiding Experimental Design
    - Modeling false positive rates
    - Choosing exclusion criteria
    - Power analysis

If H_0 is true, what is the distribution of p-values?

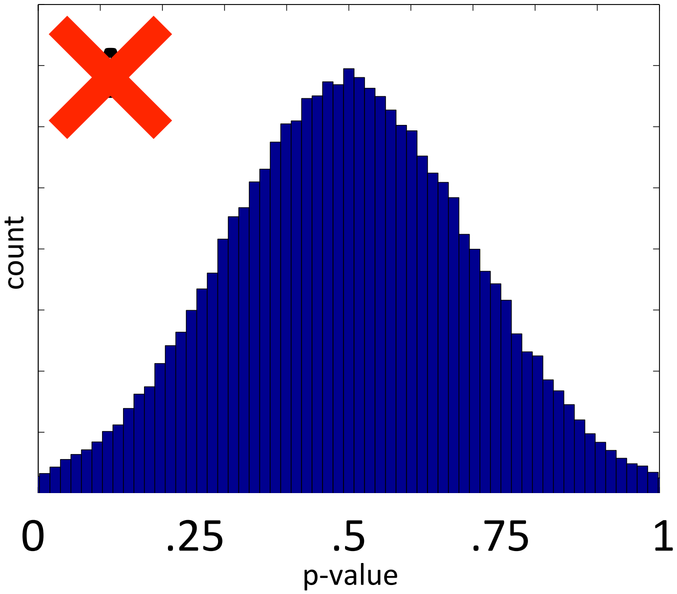


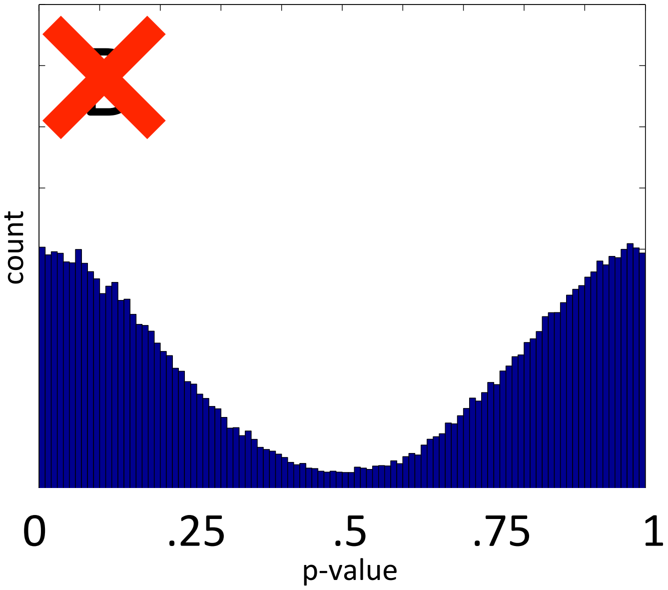


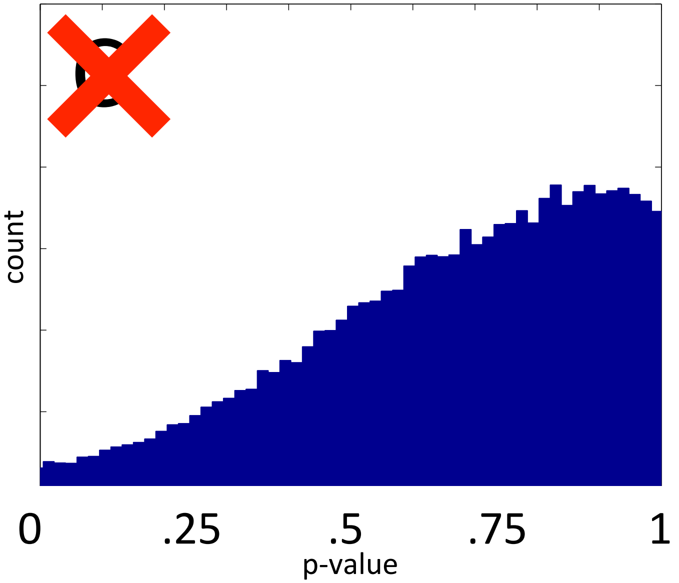


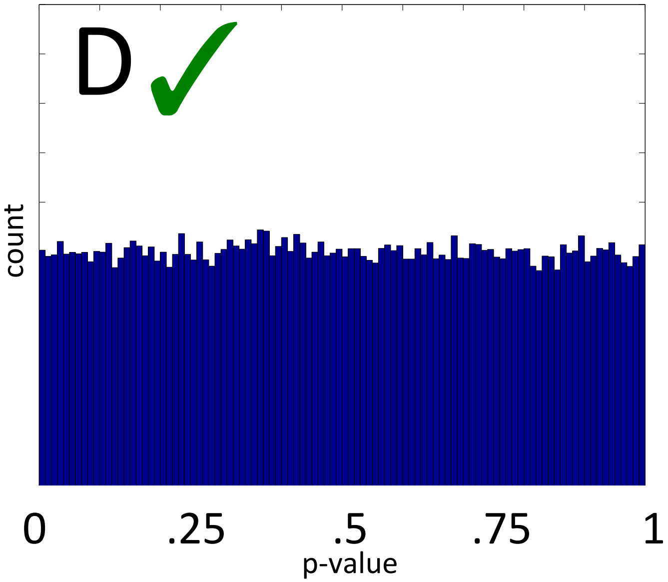


If H_0 is true, what is the distribution of p-values?

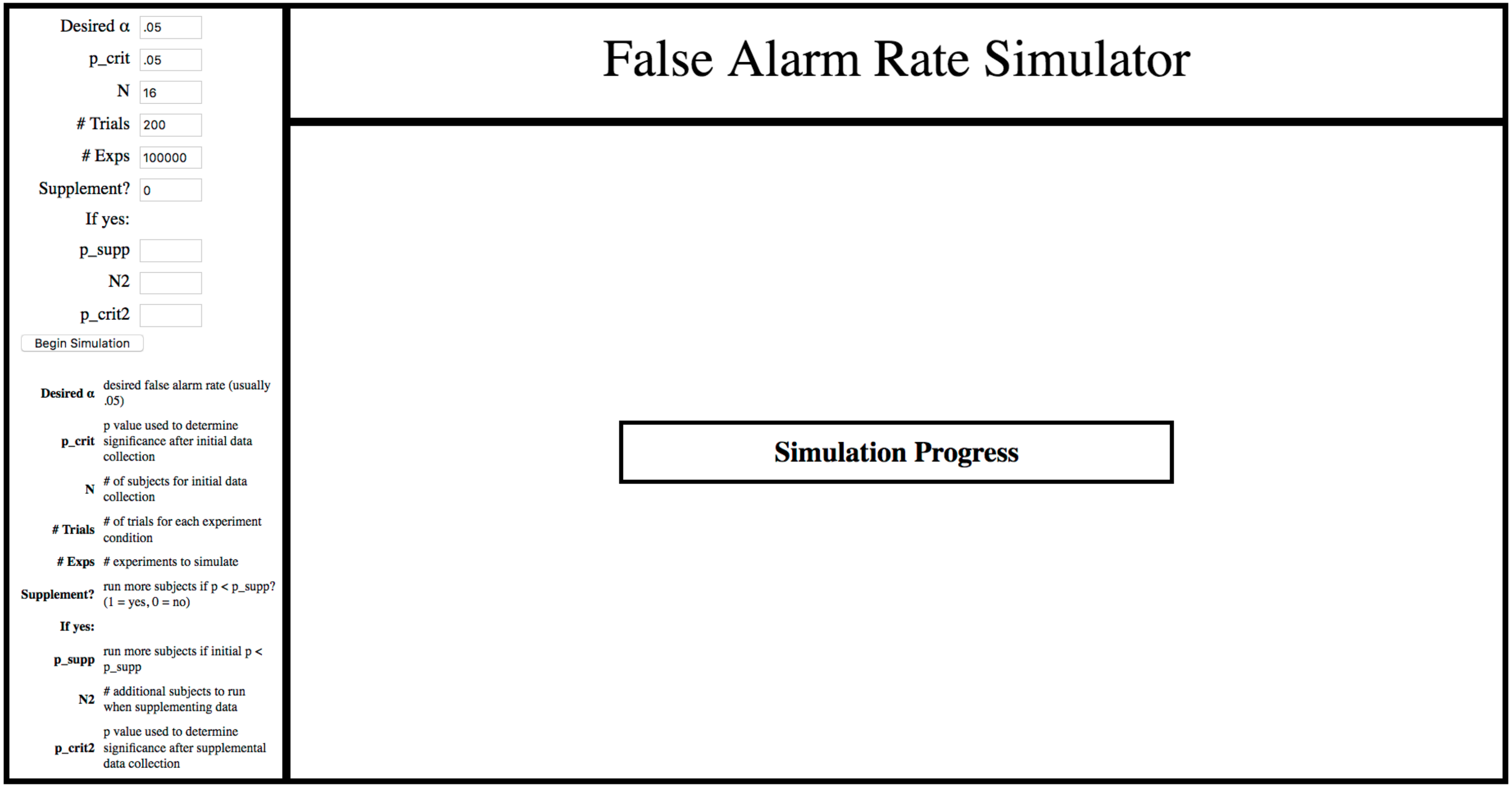








False Alarm Rate Simulator
(bit.do/psim)

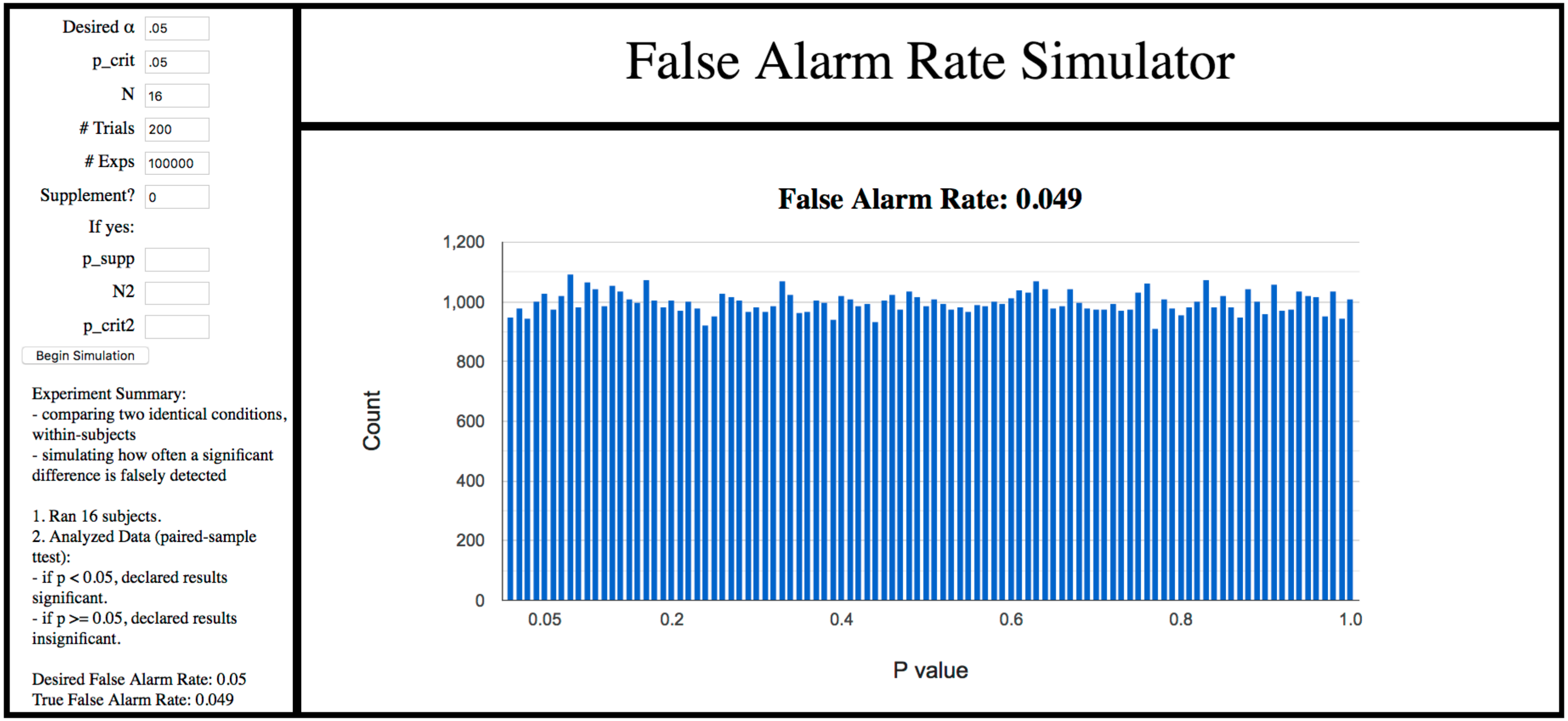


False Alarm Rate Simulator

Desired α	.05
p_crit	.05
N	16
# Trials	200
# Exps	100000
Supplement?	0
If yes:	
p_supp	
N2	
p_crit2	

Desired α	desired false alarm rate (usually .05)
p_crit	p value used to determine significance after initial data collection
N	# of subjects for initial data collection
# Trials	# of trials for each experiment condition
# Exps	# experiments to simulate
Supplement?	run more subjects if p < p_supp? (1 = yes, 0 = no)
If yes:	
p_supp	run more subjects if initial p < p_supp
N2	# additional subjects to run when supplementing data
p_crit2	p value used to determine significance after supplemental data collection

False Alarm Rate Simulator
(bit.do/psim)



Desired α .05
p_crit .05
N 16
Trials 200
Exps 100000
Supplement? 0
If yes:
p_supp
N2
p_crit2

Experiment Summary:
- comparing two identical conditions, within-subjects
- simulating how often a significant difference is falsely detected

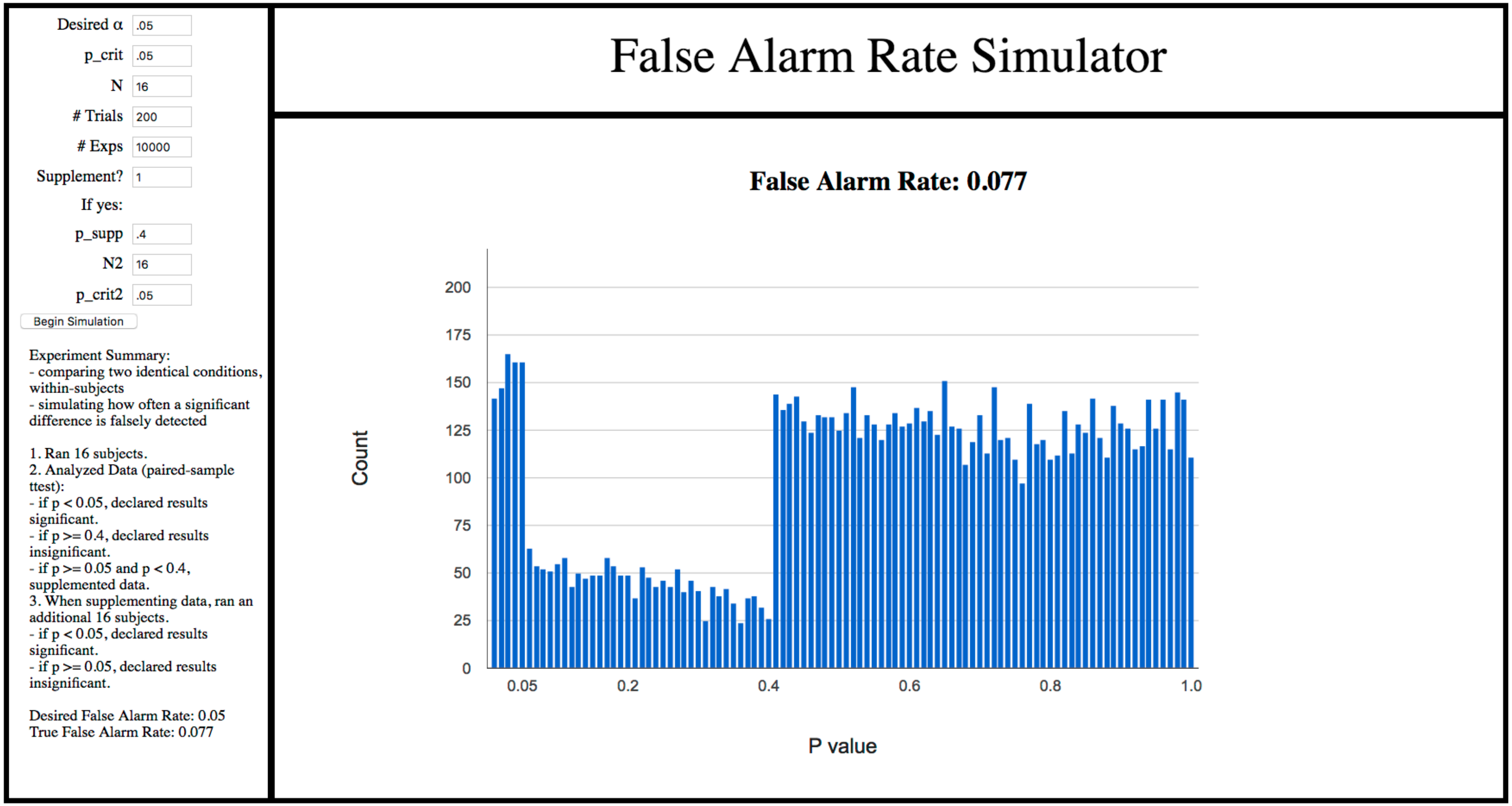
1. Ran 16 subjects.
2. Analyzed Data (paired-sample ttest):
- if p < 0.05, declared results significant.
- if p >= 0.05, declared results insignificant.

Desired False Alarm Rate: 0.05
True False Alarm Rate: 0.049

False Alarm Rate Simulator

False Alarm Rate: 0.049

False Alarm Rate Simulator
(bit.do/psim)



Desired α .05
p_crit .05
N 16
Trials 200
Exps 10000
Supplement? 1
If yes:
p_supp .4
N2 16
p_crit2 .05

Experiment Summary:
- comparing two identical conditions, within-subjects
- simulating how often a significant difference is falsely detected

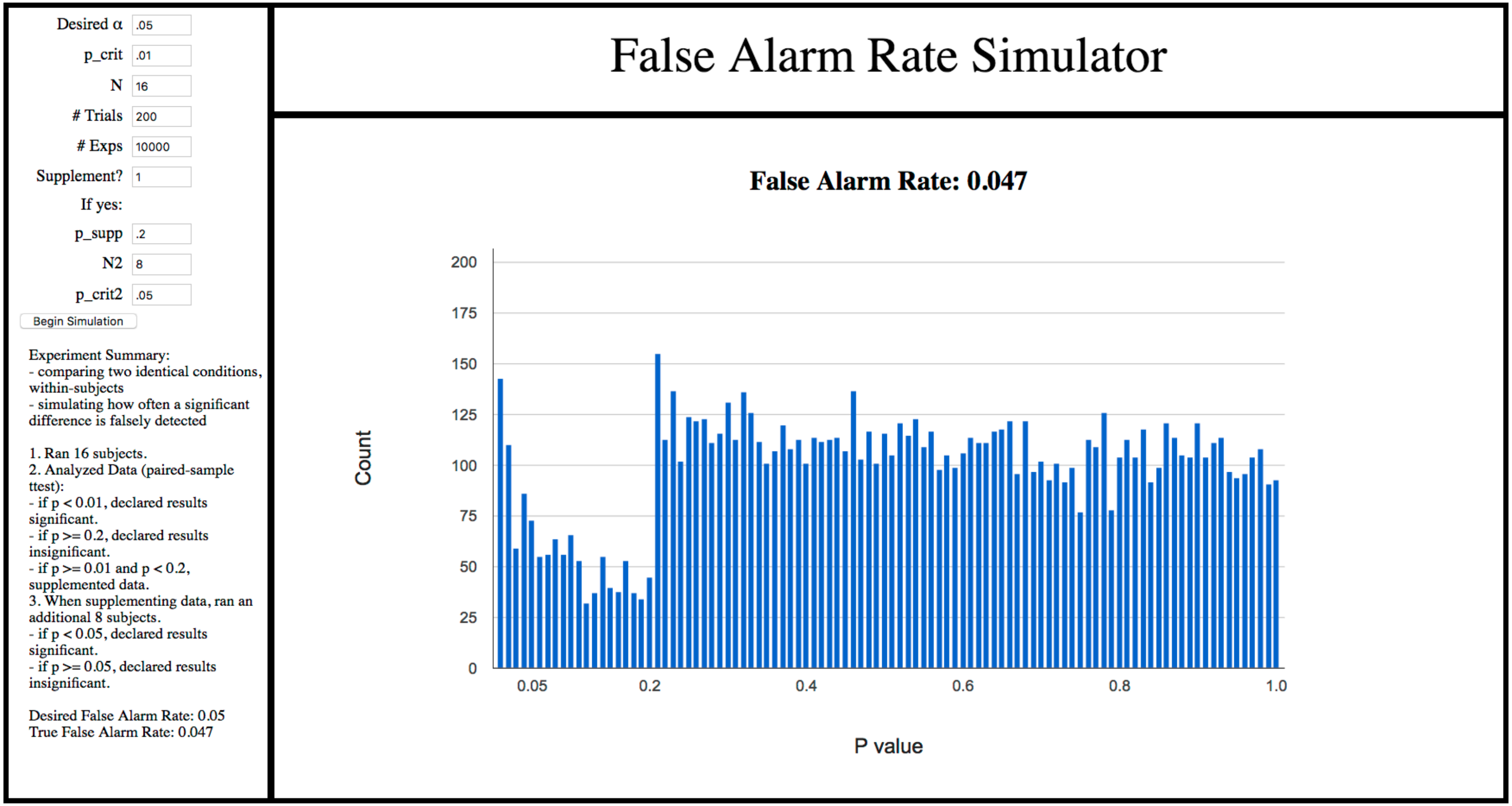
1. Ran 16 subjects.
2. Analyzed Data (paired-sample ttest):
- if p < 0.05, declared results significant.
- if p >= 0.4, declared results insignificant.
- if p >= 0.05 and p < 0.4, supplemented data.
3. When supplementing data, ran an additional 16 subjects.
- if p < 0.05, declared results significant.
- if p >= 0.05, declared results insignificant.

Desired False Alarm Rate: 0.05
True False Alarm Rate: 0.077

False Alarm Rate Simulator

False Alarm Rate: 0.077

False Alarm Rate Simulator
(bit.do/psim)



Desired α: .05
p_crit: .01
N: 16
Trials: 200
Exps: 10000
Supplement?: 1
If yes:
p_supp: .2
N2: 8
p_crit2: .05

Experiment Summary:
- comparing two identical conditions, within-subjects
- simulating how often a significant difference is falsely detected

1. Ran 16 subjects.
2. Analyzed Data (paired-sample ttest):
- if $p < 0.01$, declared results significant.
- if $p >= 0.2$, declared results insignificant.
- if $p >= 0.01$ and $p < 0.2$, supplemented data.
3. When supplementing data, ran an additional 8 subjects.
- if $p < 0.05$, declared results significant.
- if $p >= 0.05$, declared results insignificant.

Desired False Alarm Rate: 0.05
True False Alarm Rate: 0.047

False Alarm Rate Simulator

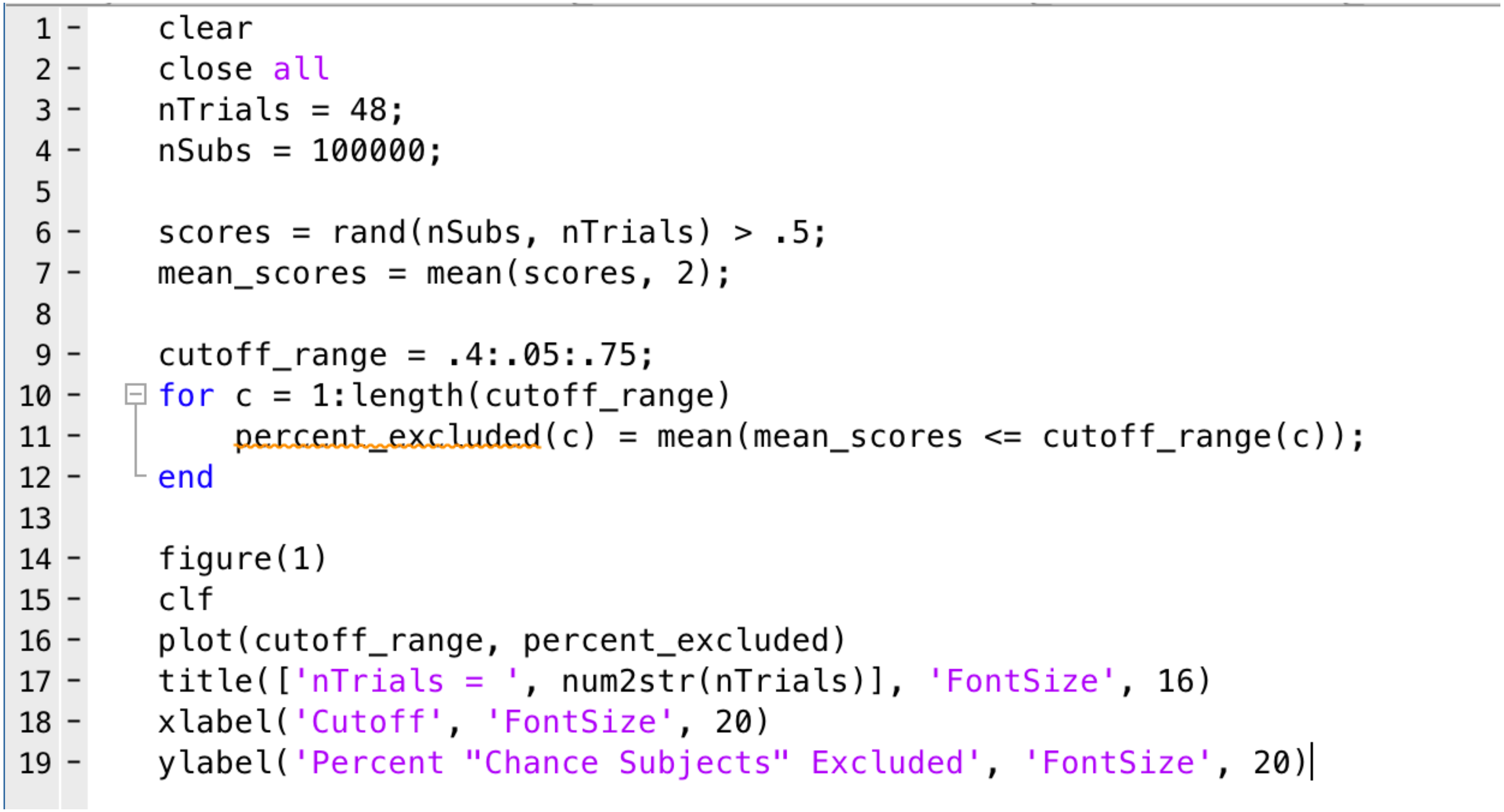
False Alarm Rate: 0.047

Using Simulation in Research

1. Guiding Experimental Design
    - Modeling false positive rates
    - Choosing exclusion criteria
    - Power analysis

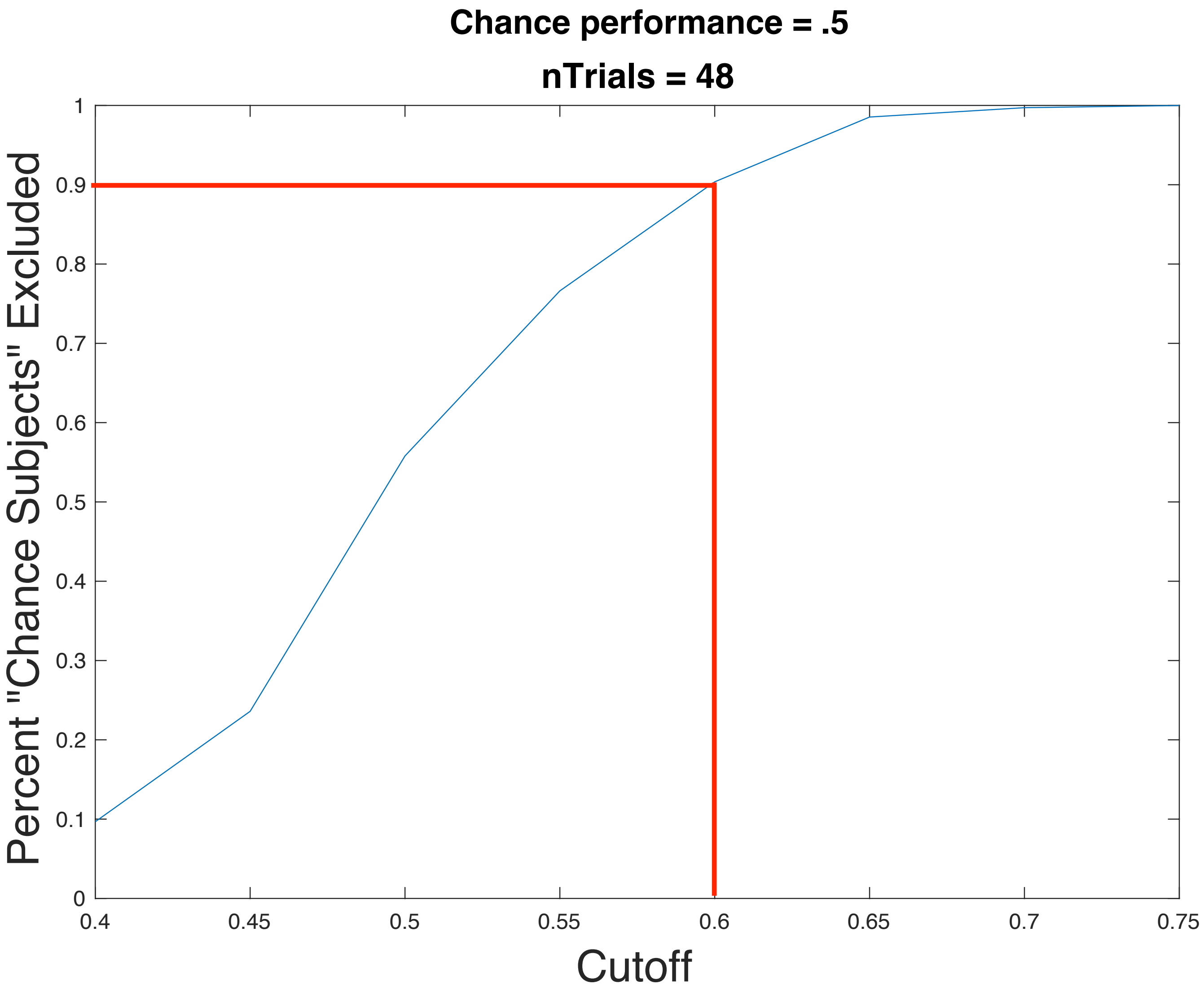
Choosing Exclusion Criteria

Choosing Exclusion Criteria



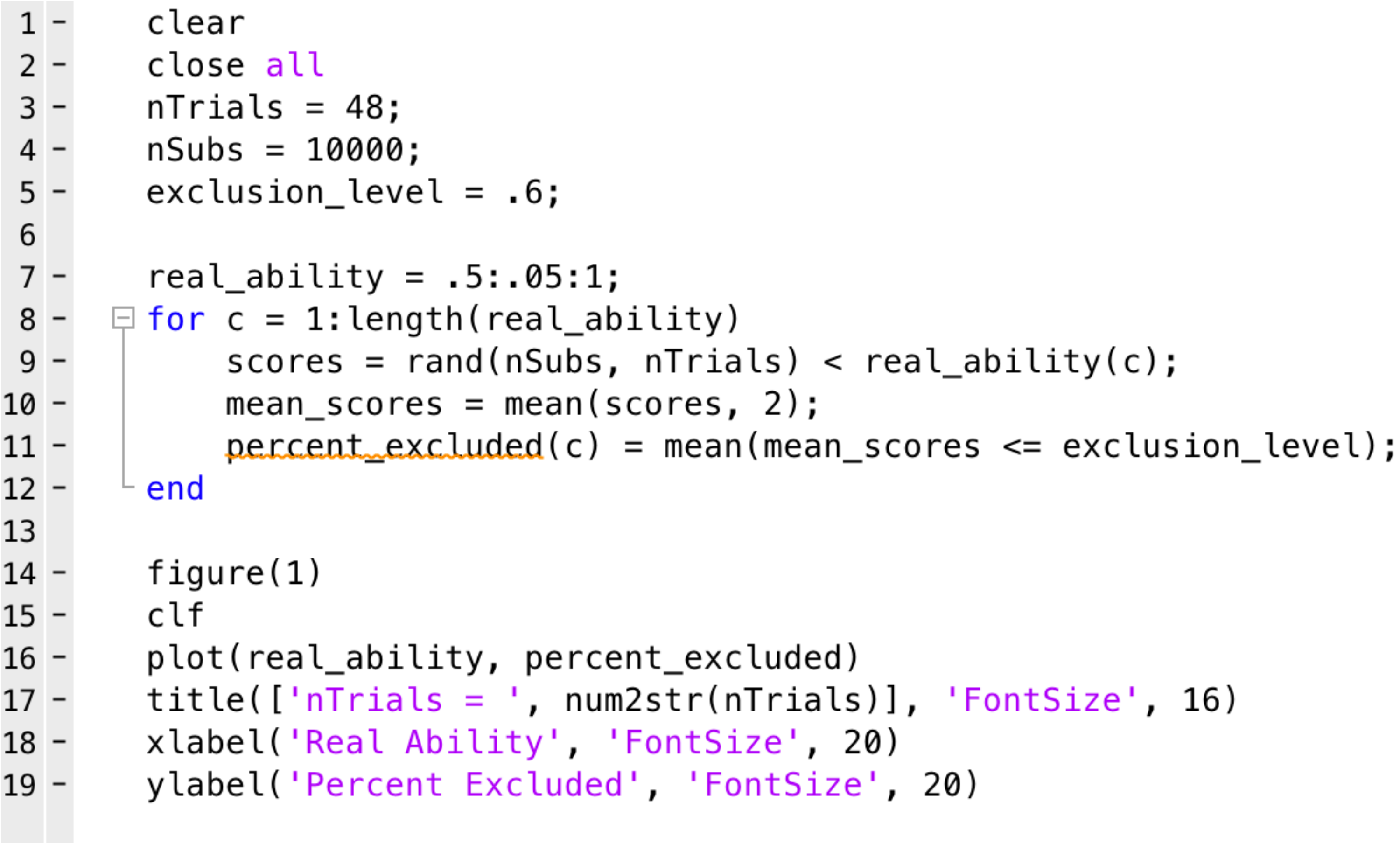
```matlab
clear
close all
nTrials = 48;
nSubs = 100000;

scores = rand(nSubs, nTrials) > .5;
mean_scores = mean(scores, 2);

cutoff_range = .4:.05:.75;
for c = 1:length(cutoff_range)
    percent_excluded(c) = mean(mean_scores <= cutoff_range(c));
end

figure(1)
clf
plot(cutoff_range, percent_excluded)
title(['nTrials = ', num2str(nTrials)], 'FontSize', 16)
xlabel('Cutoff', 'FontSize', 20)
ylabel('Percent "Chance Subjects" Excluded', 'FontSize', 20)
```

Choosing Exclusion Criteria



Choosing Exclusion Criteria

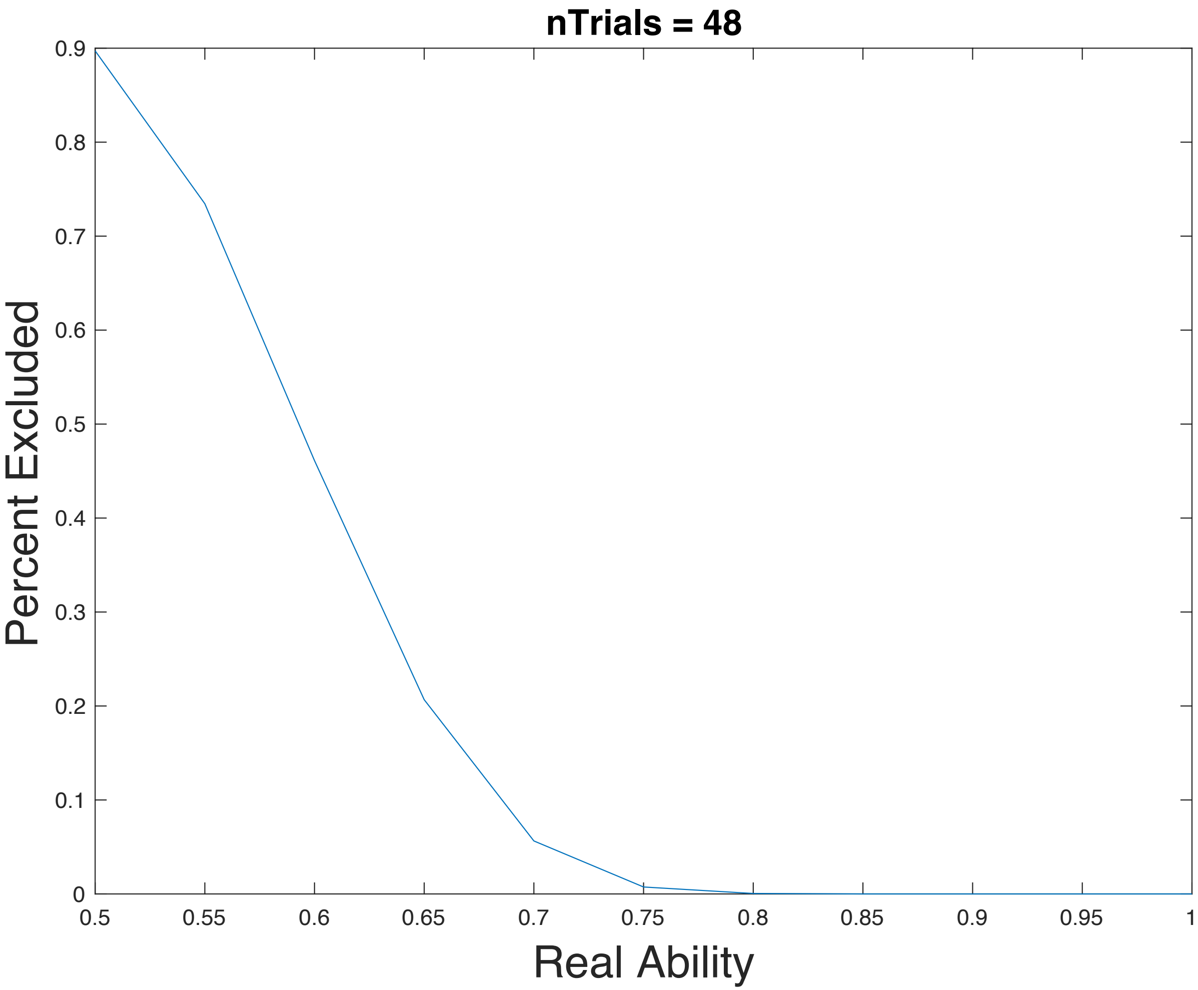
What about subjects not at chance levels?



```matlab
clear
close all
nTrials = 48;
nSubs = 10000;
exclusion_level = .6;

real_ability = .5:.05:1;
for c = 1:length(real_ability)
    scores = rand(nSubs, nTrials) < real_ability(c);
    mean_scores = mean(scores, 2);
    percent_excluded(c) = mean(mean_scores <= exclusion_level);
end

figure(1)
clf
plot(real_ability, percent_excluded)
title(['nTrials = ', num2str(nTrials)], 'FontSize', 16)
xlabel('Real Ability', 'FontSize', 20)
ylabel('Percent Excluded', 'FontSize', 20)
```

Choosing Exclusion Criteria

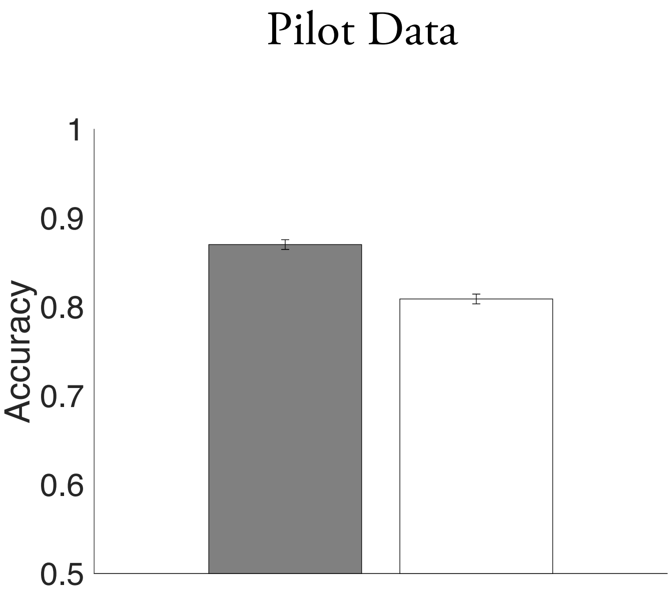
What about subjects not at chance levels?



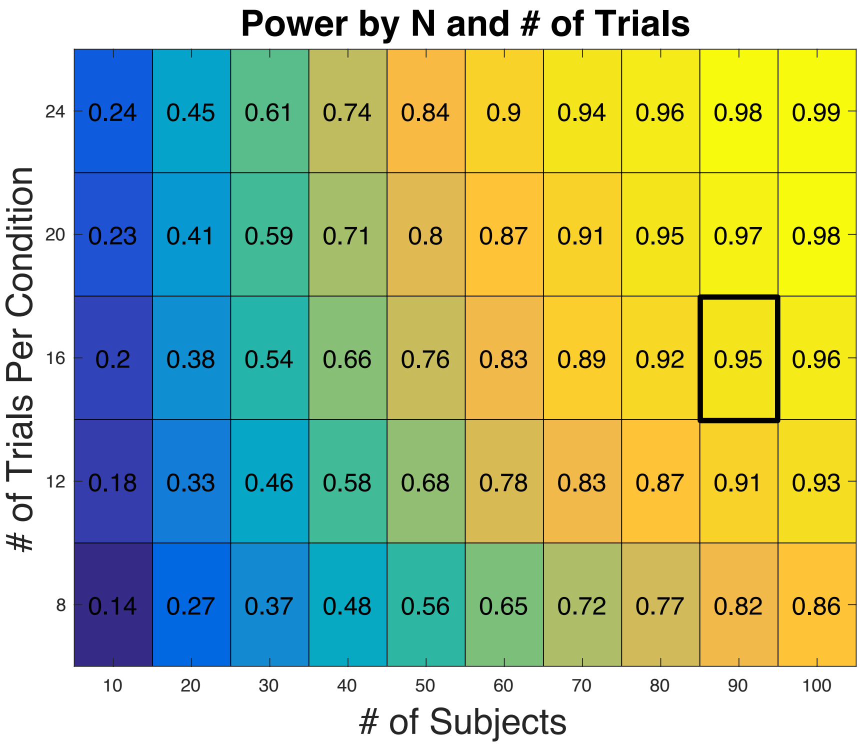
Using Simulation in Research

1. Guiding Experimental Design
    - Modeling false positive rates
    - Choosing exclusion criteria
    - Power analysis

Maximizing Power



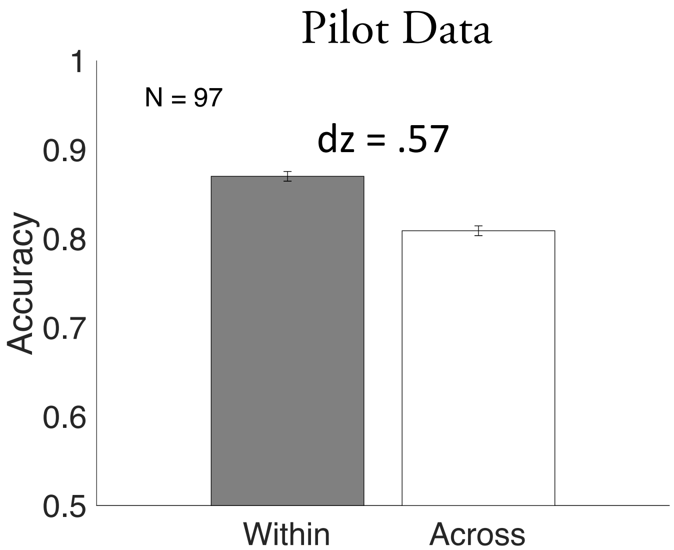
Pilot Data

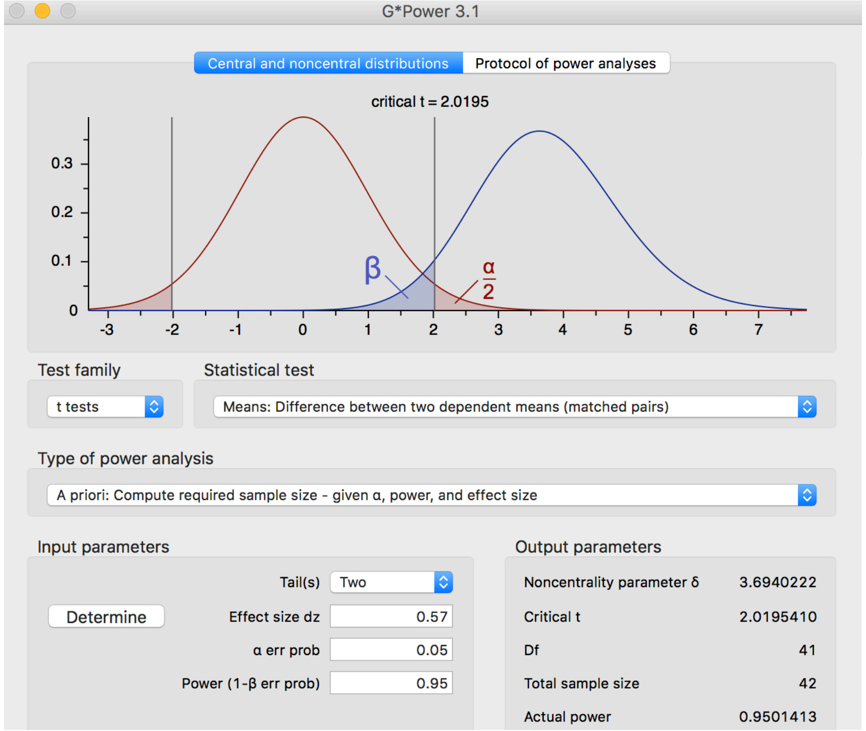


Power by N and # of Trials

# of Trials Per Condition \ # of Subjects	10	20	30	40	50	60	70	80	90	100
24	0.24	0.45	0.61	0.74	0.84	0.9	0.94	0.96	0.98	0.99
20	0.23	0.41	0.59	0.71	0.8	0.87	0.91	0.95	0.97	0.98
16	0.2	0.38	0.54	0.66	0.76	0.83	0.89	0.92	0.95	0.96
12	0.18	0.33	0.46	0.58	0.68	0.78	0.83	0.87	0.91	0.93
8	0.14	0.27	0.37	0.48	0.56	0.65	0.72	0.77	0.82	0.86

For more info: rogerstrong.weebly.com/resources.html





Test family: t tests

Statistical test: Means: Difference between two dependent means (matched pairs)

Type of power analysis: A priori: Compute required sample size - given α, power, and effect size

Input parameters

Tail(s): Two

Effect size dz: 0.57

α err prob: 0.05

Power (1-β err prob): 0.95

Output parameters

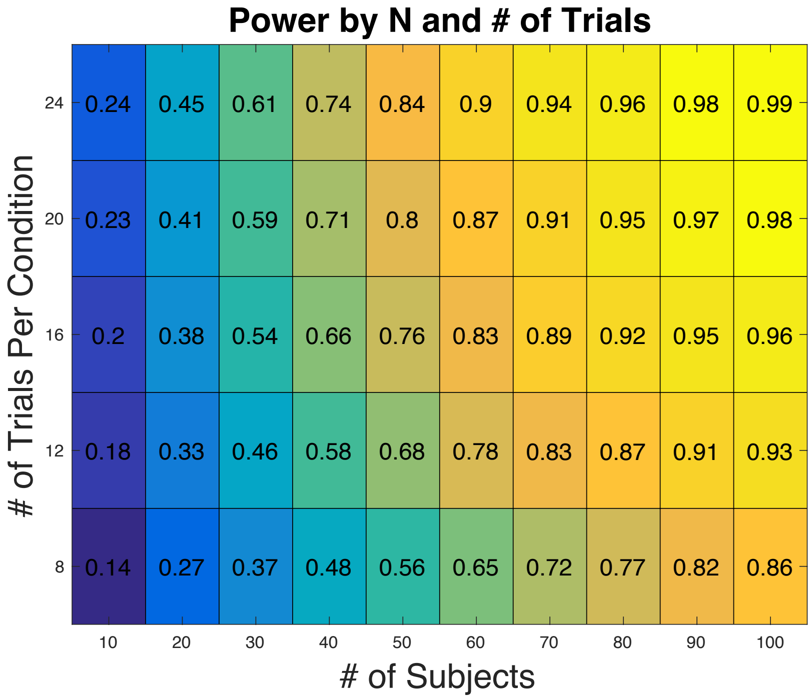
Noncentrality parameter δ: 3.6940222

Critical t: 2.0195410

Df: 41

Total sample size: 42

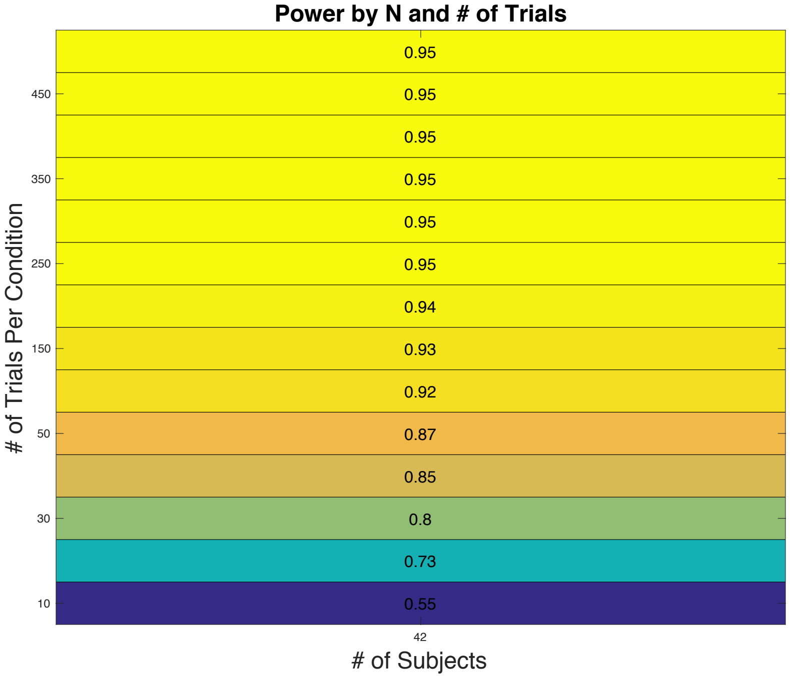
Actual power: 0.9501413



Power by N and # of Trials

# of Trials Per Condition	10	20	30	40	50	60	70	80	90	100
24	0.24	0.45	0.61	0.74	0.84	0.9	0.94	0.96	0.98	0.99
20	0.23	0.41	0.59	0.71	0.8	0.87	0.91	0.95	0.97	0.98
16	0.2	0.38	0.54	0.66	0.76	0.83	0.89	0.92	0.95	0.96
12	0.18	0.33	0.46	0.58	0.68	0.78	0.83	0.87	0.91	0.93
8	0.14	0.27	0.37	0.48	0.56	0.65	0.72	0.77	0.82	0.86

# of Subjects



Power by N and # of Trials

# of Trials Per Condition	42
	0.95
450	0.95
	0.95
350	0.95
	0.95
250	0.95
	0.94
150	0.93
	0.92
50	0.87
	0.85
30	0.8
	0.73
10	0.55

# of Subjects

G power doesn't know how many trials you are using, likely overestimating your power

General Notes

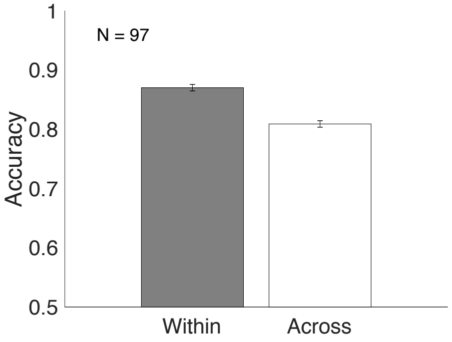
- PowerAnalysis.m does most the work, and is called in the example scripts
- NOTE: This version only simulates t-tests between within subject conditions

- Key Components:
  - prefs.data:
    - either a #subjects (rows) x #conditions (columns) array, or a string file name of an excel or .csv file with data listed as #subjects x #conditions.
    - Data can be listed as either decimal (.5) or percentage (50), although you will get a warning for the later (as data will be converted to decimal)
    - If using excel or csv file, there should NOT be a header row
  - prefs.N_range
    - Range of number of participants to simulate. E.g., 10:10:50 will simulate with 10, 20, 30, 40, and 50 participants
  - prefs.trial_range
    - Range of number of trials per condition to simulate. E.g., 8:4:24 will simulate with 8, 12, 16, 20, and 24 trials per condition
  - prefs.alpha
    - p-value to use in power simulations
  - prefs.nSims
    - How many simulations to use for every particpant/trial number combination. 10,000 is a decent estimate and runs pretty quickly, 100,000 is slower but a more stable estimate.
  - prefs.comps
    - Which comparisons to test for significance. Each row is a comparison, with the condition expected to be higher magnitude listed in the first column, and the condition expected to have lower magnitude in the second column. A study will be classified as "successful" only if all listed comparisons are significant (see examples).

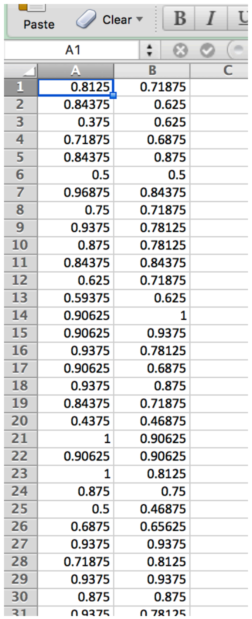
Example 1

Pilot Data

- 97 subjects, 2 conditions
- Excel file is 97 rows x 2 columns

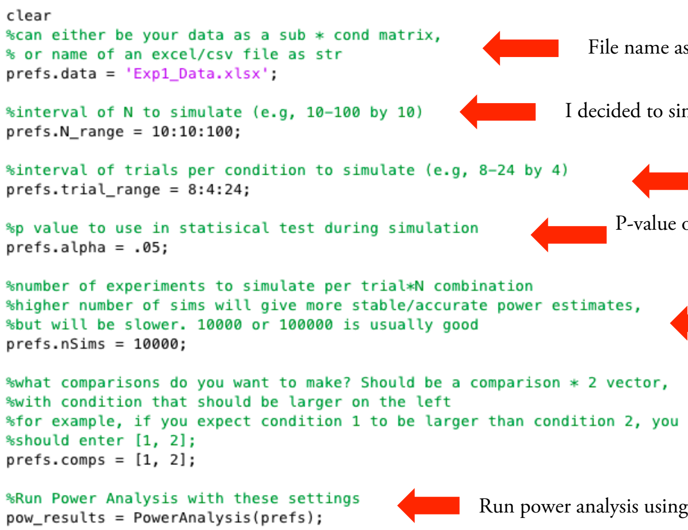


Exp1_Data.xlsx



	A	B	C
1	0.8125	0.71875	
2	0.84375	0.625	
3	0.375	0.625	
4	0.71875	0.6875	
5	0.84375	0.875	
6	0.5	0.5	
7	0.96875	0.84375	
8	0.75	0.71875	
9	0.9375	0.78125	
10	0.875	0.78125	
11	0.84375	0.84375	
12	0.625	0.71875	
13	0.59375	0.625	
14	0.90625	1	
15	0.90625	0.9375	
16	0.9375	0.78125	
17	0.90625	0.6875	
18	0.9375	0.875	
19	0.84375	0.71875	
20	0.4375	0.46875	
21	1	0.90625	
22	0.90625	0.90625	
23	1	0.8125	
24	0.875	0.75	
25	0.5	0.46875	
26	0.6875	0.65625	
27	0.9375	0.9375	
28	0.71875	0.8125	
29	0.9375	0.9375	
30	0.875	0.875	
31	0.9375	0.78125	

Power Analysis Settings



```
clear
%can either be your data as a sub * cond matrix,
% or name of an excel/csv file as str
prefs.data = 'Exp1_Data.xlsx';

%interval of N to simulate (e.g, 10-100 by 10)
prefs.N_range = 10:10:100;

%interval of trials per condition to simulate (e.g, 8-24 by 4)
prefs.trial_range = 8:4:24;

%p value to use in statisical test during simulation
prefs.alpha = .05;

%number of experiments to simulate per trial*N combination
%higher number of sims will give more stable/accurate power estimates,
%but will be slower. 10000 or 100000 is usually good
prefs.nSims = 10000;

%what comparisons do you want to make? Should be a comparison * 2 vector,
%with condition that should be larger on the left
%for example, if you expect condition 1 to be larger than condition 2, you
%should enter [1, 2];
prefs.comps = [1, 2];

%Run Power Analysis with these settings
pow_results = PowerAnalysis(prefs);
```

File name as string (can also do data directly in matlab)

I decided to simulate N from 10-100 by 10

I decided to simulate trial number per condition from 8-24 by 4

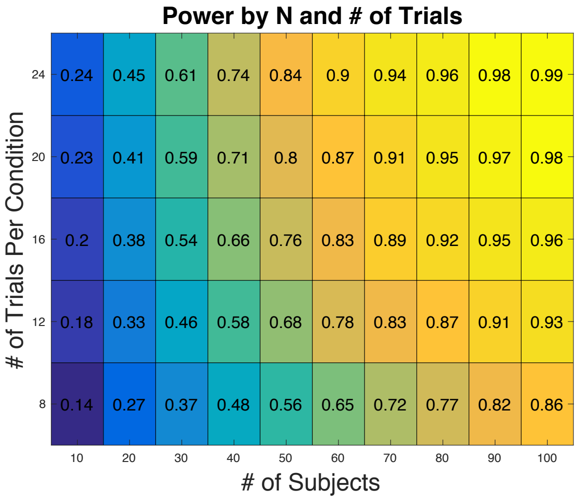
P-value of .05 used in simulation

10,000 sims per N x num_trials combo (sims per cell in output graph)

Only comparison I was interested in was condition 1 being larger than condition 2

Run power analysis using these settings

Power Analysis Output



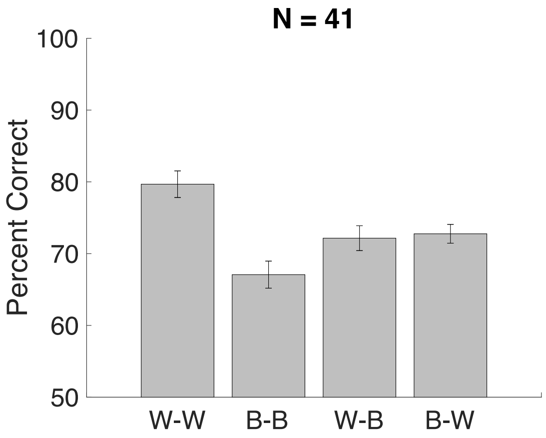
# of Trials Per Condition	10	20	30	40	50	60	70	80	90	100
24	0.24	0.45	0.61	0.74	0.84	0.9	0.94	0.96	0.98	0.99
20	0.23	0.41	0.59	0.71	0.8	0.87	0.91	0.95	0.97	0.98
16	0.2	0.38	0.54	0.66	0.76	0.83	0.89	0.92	0.95	0.96
12	0.18	0.33	0.46	0.58	0.68	0.78	0.83	0.87	0.91	0.93
8	0.14	0.27	0.37	0.48	0.56	0.65	0.72	0.77	0.82	0.86

Simulated power for each N X number or trials per condition combo we specified in settings. Looking at this, I know I could achieve > 90% power by running 90 subjects with 12 trials per condition, for example

Example 2

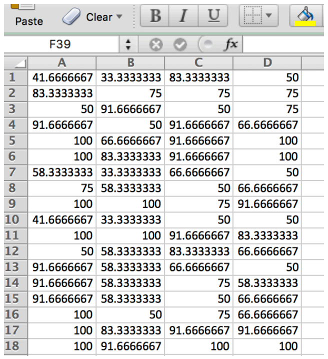
Pilot Data

- 41 subjects, 4 conditions
- Excel file is 41 rows x 4 columns



Exp2_Data.xlsx

Data is in percent, so script will convert to decimal and give a warning that this has occurred.



	A	B	C	D
1	41.6666667	33.3333333	83.3333333	50
2	83.3333333	75	75	75
3	50	91.6666667	50	75
4	91.6666667	50	91.6666667	66.6666667
5	100	66.6666667	91.6666667	100
6	100	83.3333333	91.6666667	100
7	58.3333333	33.3333333	66.6666667	50
8	75	58.3333333	50	66.6666667
9	100	100	75	91.6666667
10	41.6666667	33.3333333	50	50
11	100	100	91.6666667	83.3333333
12	50	58.3333333	83.3333333	66.6666667
13	91.6666667	58.3333333	66.6666667	50
14	91.6666667	58.3333333	75	58.3333333
15	91.6666667	58.3333333	50	66.6666667
16	100	50	75	66.6666667
17	100	83.3333333	91.6666667	91.6666667
18	100	91.6666667	100	100

Power Analysis Settings

```
clear
%can either be your data as a sub * cond matrix,
% or name of an excel/csv file as str
prefs.data = 'Exp2_Data.xlsx';

%interval of N to simulate (e.g, 50-300 by 25)
prefs.N_range = 50:25:300;

%interval of trials per condition to simulate (e.g, 8-20 by 4)
prefs.trial_range = 8:4:20;

%p value to use in statisical test during simulation
prefs.alpha = .05;

%number of experiments to simulate per trial*N combination
%higher number of sims will give more stable/accurate power estimates,
%but will be slower. 10000 or 100000 is usually good
prefs.nSims = 10000;

%what comparisons do you want to make? Should be a comparison * 2 vector,
%with condition that should be larger on the left
%for example, if you expect condition 1 to be larger than condition 2, you
%should enter [1, 2];
prefs.comps = [1, 2
    1 3
    1 4
    3 2
    4 2];

%Run Power Analysis with these settings
pow_results = PowerAnalysis(prefs);
```

File name as string (can also do data directly in matlab).

I decided to simulate N from 50-300 by 25

I decided to simulate trial number per condition from 8-20 by 4

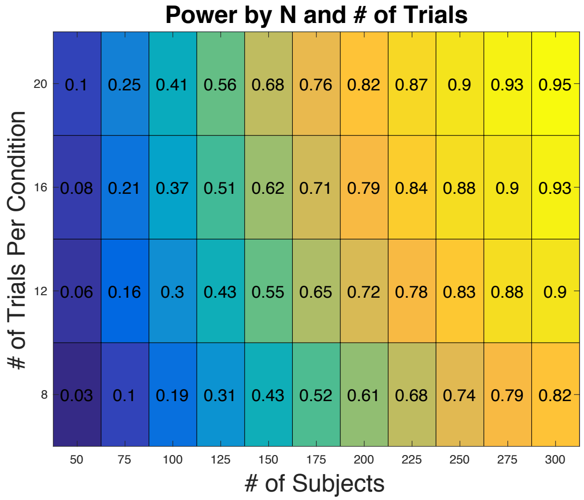
P-value of .05 used in simulation

10,000 sims per N x num_trials combo (sims per cell in output graph)

This time, I had 5 comparisons I am interested in. Specifically, I only want to call the study a "success" if condition 1 >2, 1 >3, 1>4, 3>2, and 4>2. Each comparison specified as a separate row.

Run power analysis using these settings

Power Analysis Output



# of Trials Per Condition	50	75	100	125	150	175	200	225	250	275	300
20	0.1	0.25	0.41	0.56	0.68	0.76	0.82	0.87	0.9	0.93	0.95
16	0.08	0.21	0.37	0.51	0.62	0.71	0.79	0.84	0.88	0.9	0.93
12	0.06	0.16	0.3	0.43	0.55	0.65	0.72	0.78	0.83	0.88	0.9
8	0.03	0.1	0.19	0.31	0.43	0.52	0.61	0.68	0.74	0.79	0.82

Simulated power for each N X number or trials per condition combo we specified in settings. Looking at this, I know I could achieve > 90% power by running 300 subjects with 12 trials per condition, for example. Note that this is power for ALL 5 comparisons of interest being significant

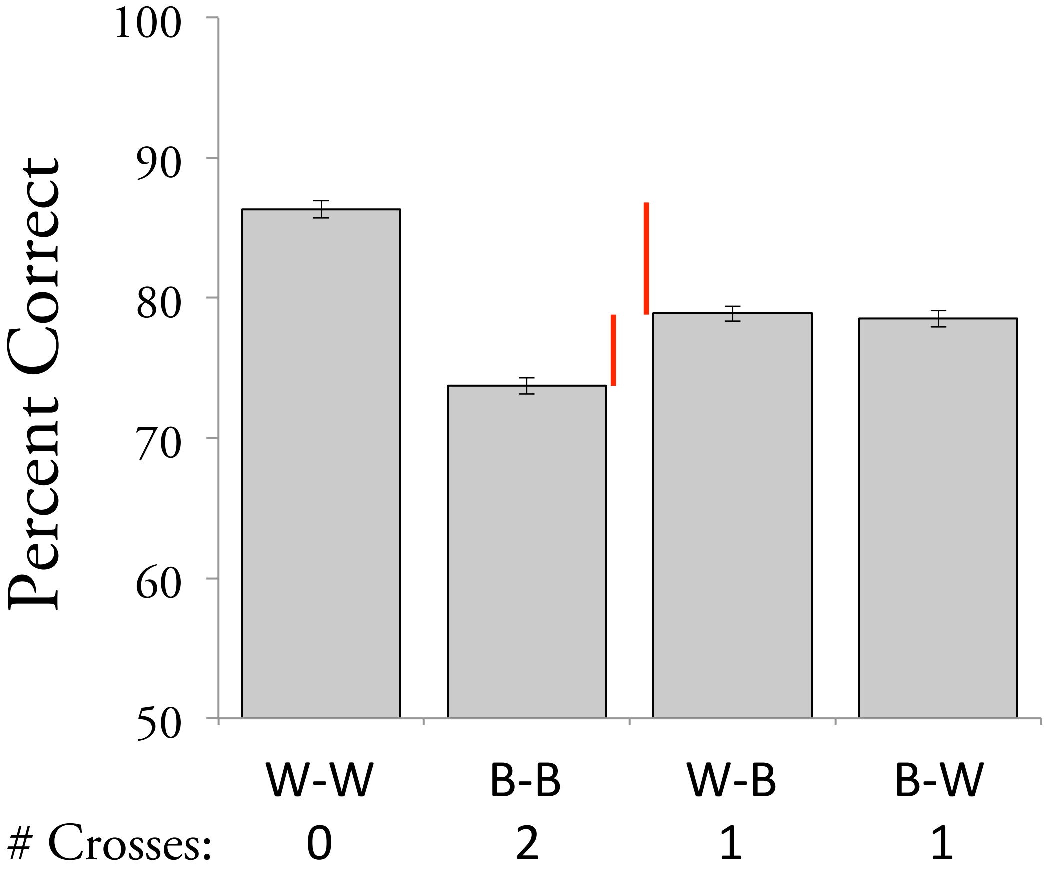
Note: for my actual power analysis, applied exclusion criteria as well (not currently implemented)

Using Simulation in Research

1. Guiding Experimental Design

2. Understanding Results

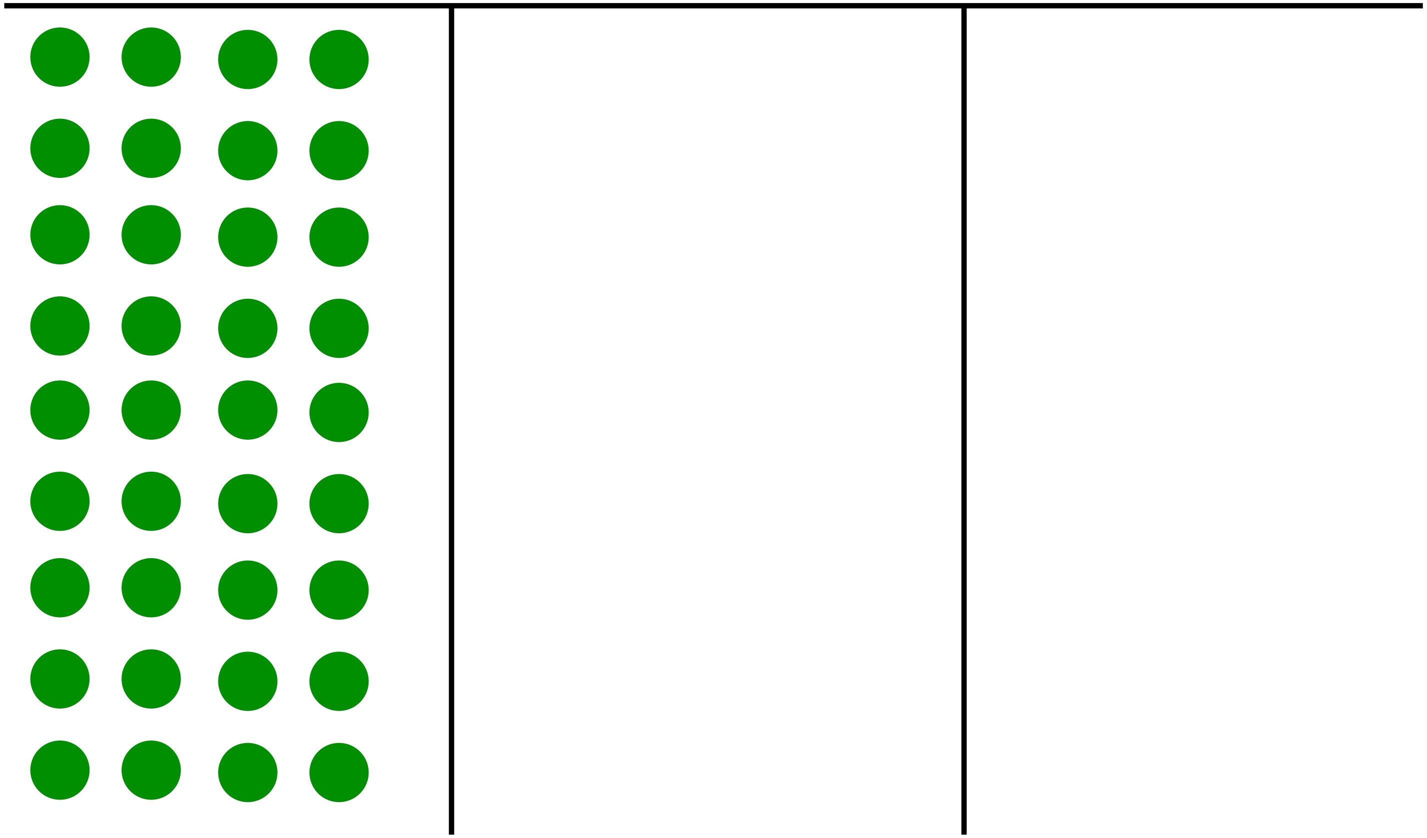
Modeling the Exchange



Modeling the Exchange

Before 1st Cross

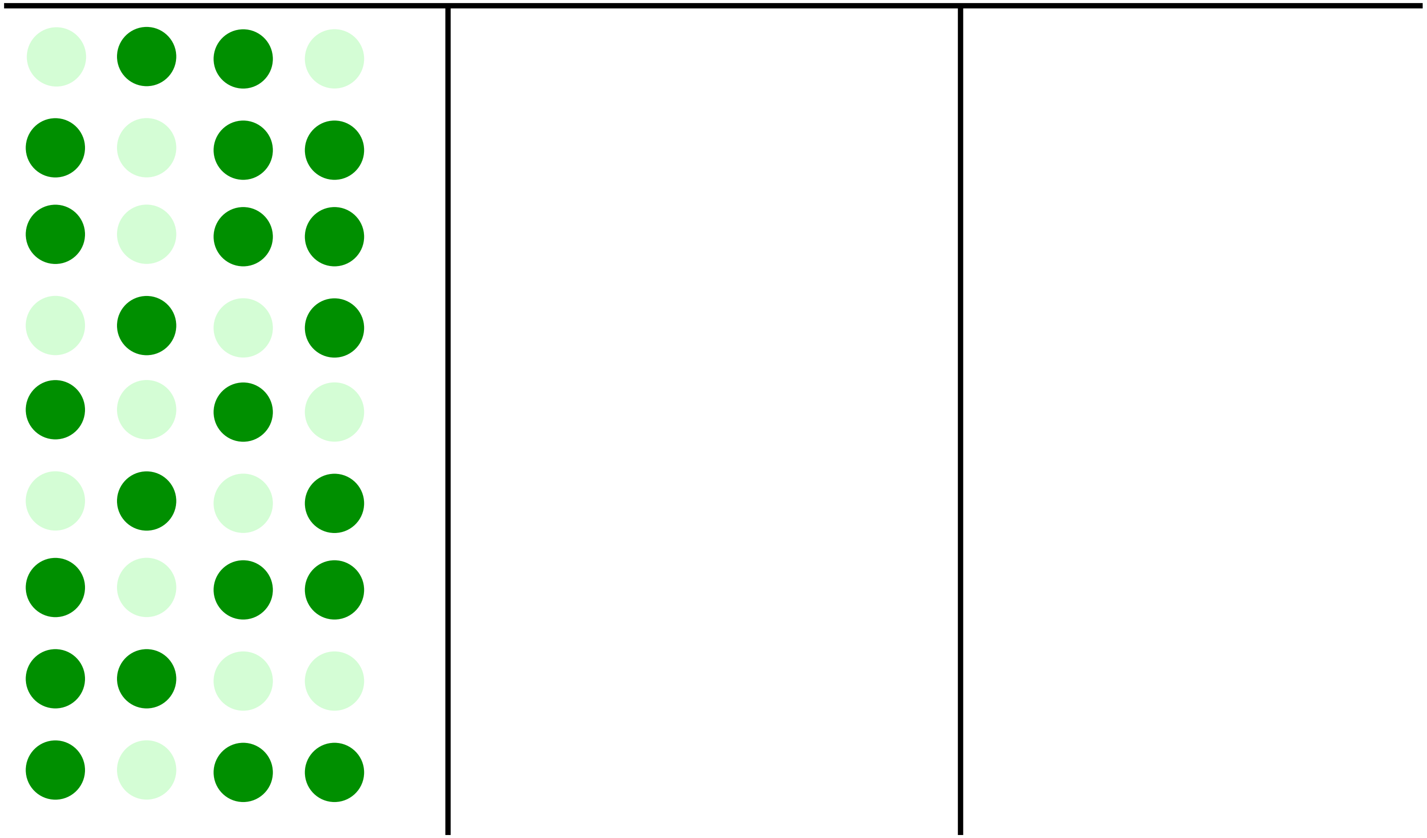
0 Wrong



Modeling the Exchange

Before 1st Cross

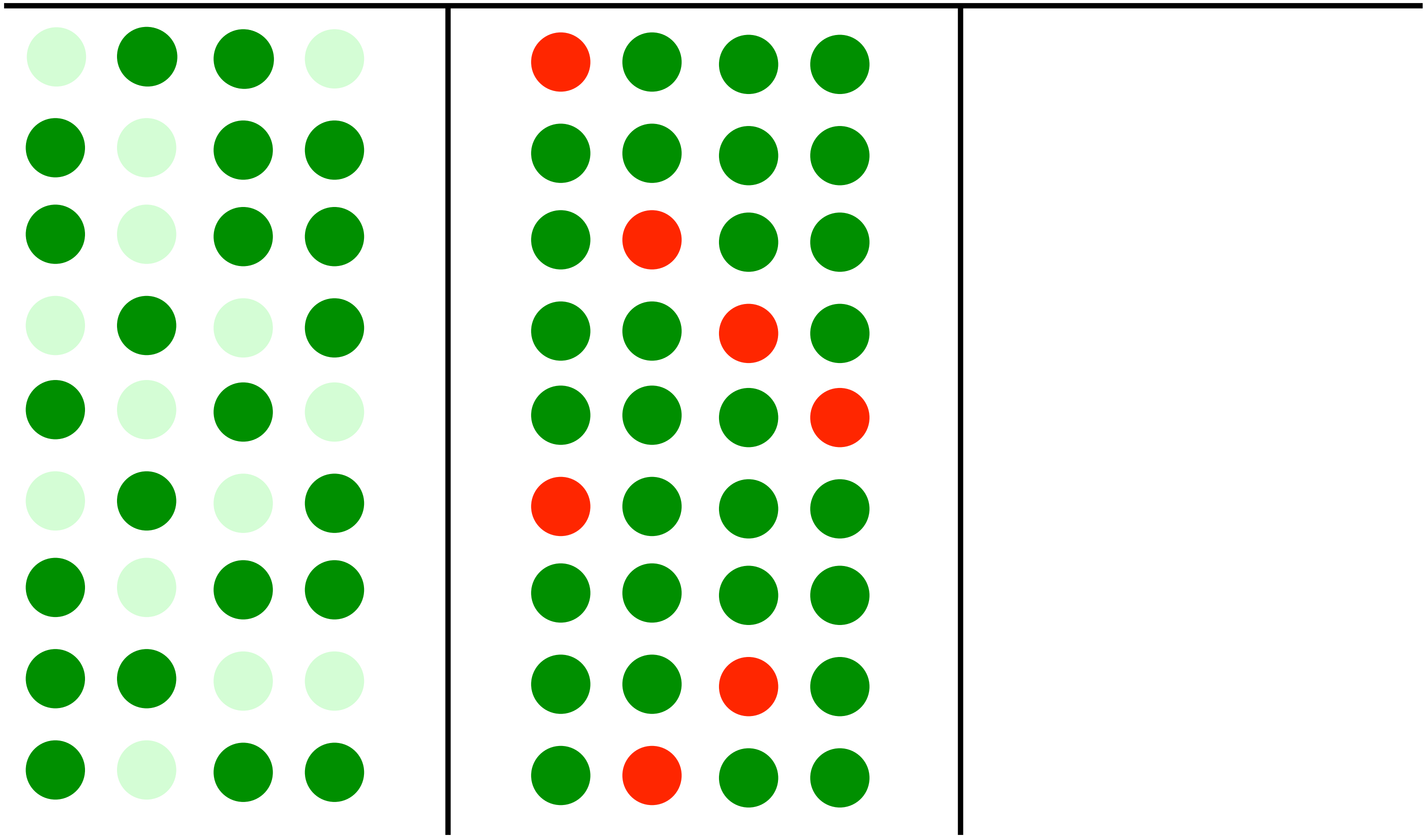
0 Wrong



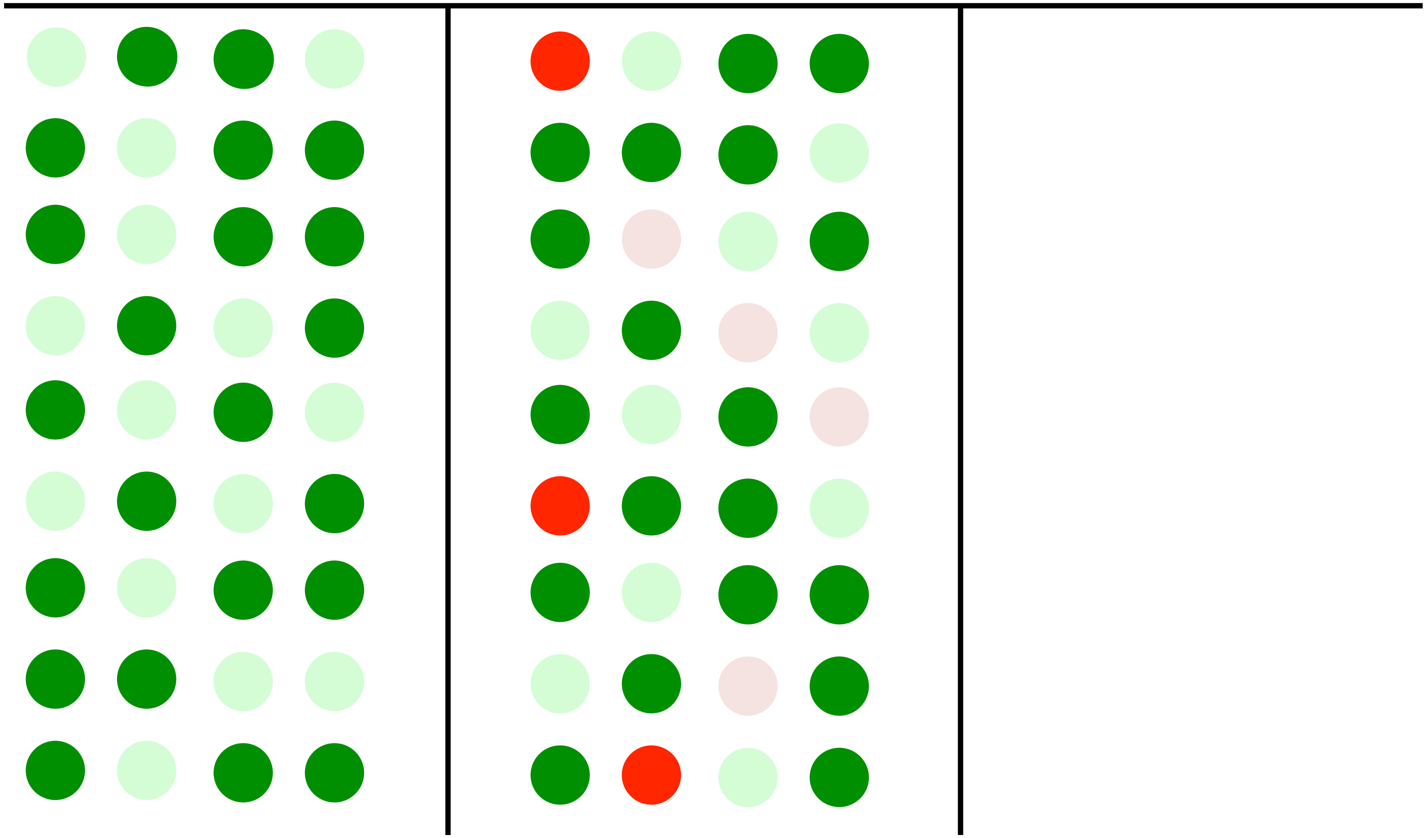
Modeling the Exchange

Before 1st Cross
0 Wrong

After 1st Cross
7 Wrong



Modeling the Exchange



Before 1st Cross
0 Wrong

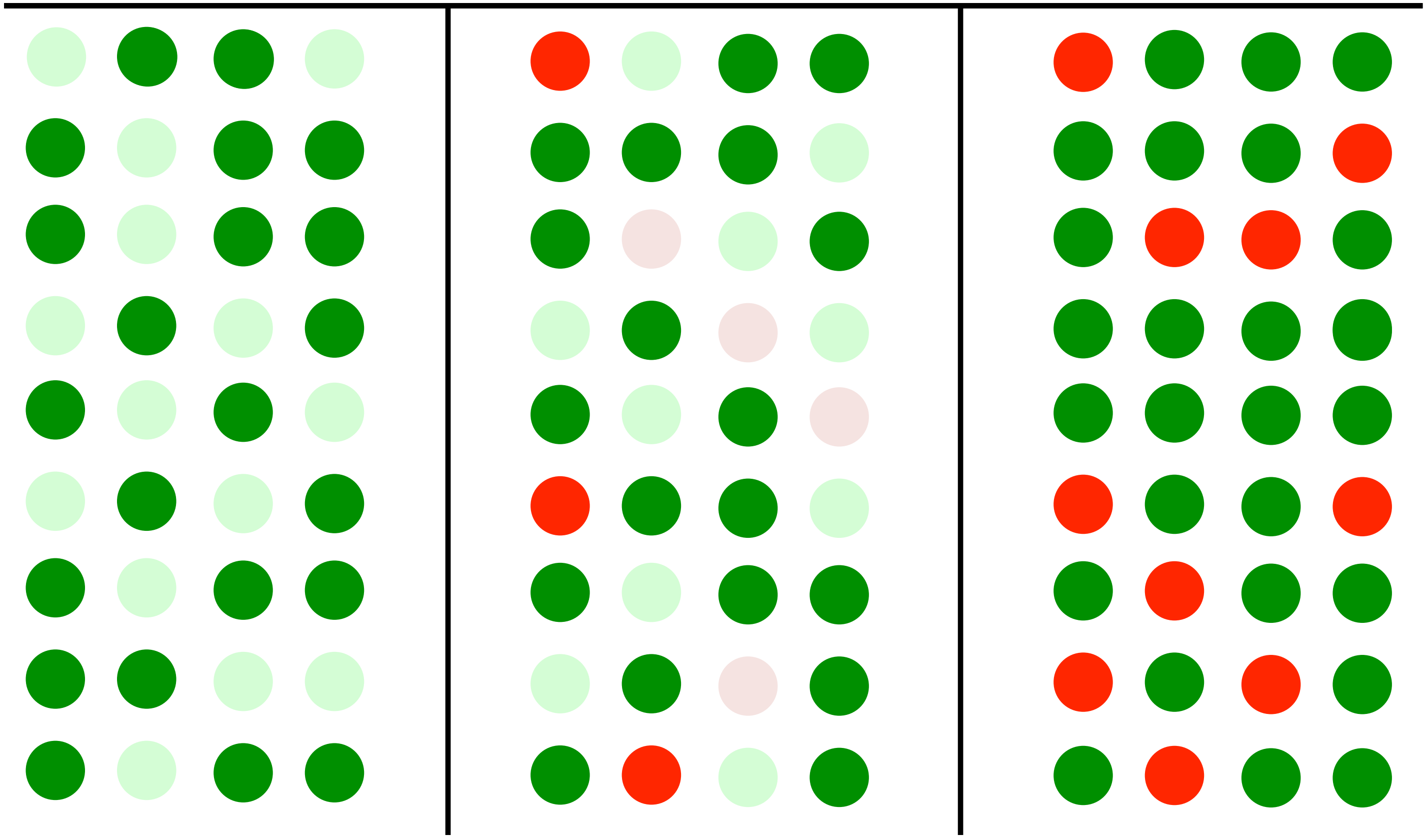
After 1st Cross
7 Wrong

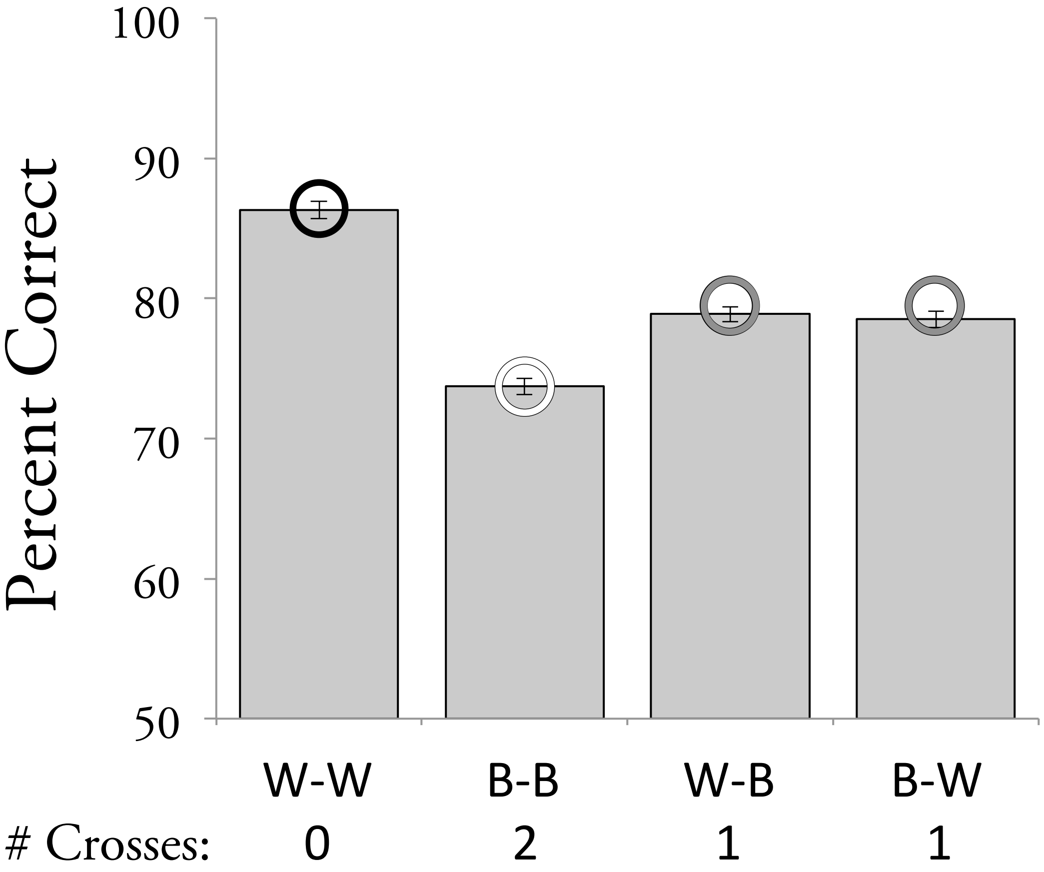
Modeling the Exchange

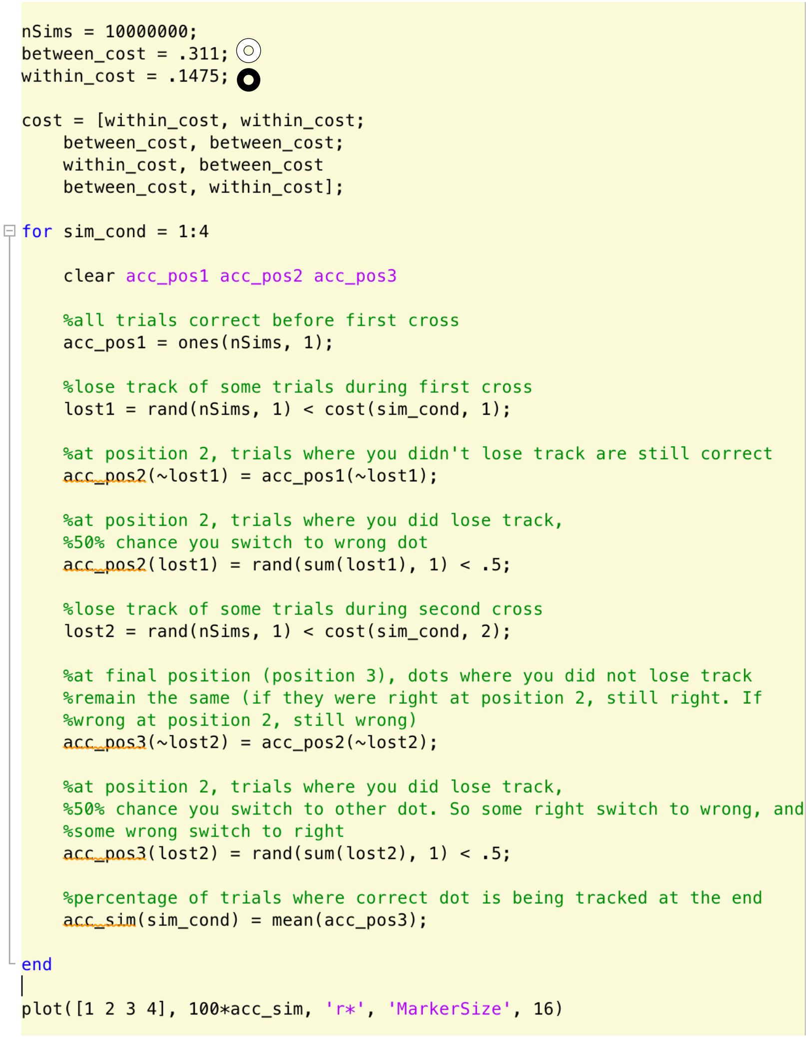
Before 1st Cross
0 Wrong

After 1st Cross
7 Wrong

After 2nd Cross
10 Wrong







```
nSims = 10000000;
between_cost = .311; ○
within_cost = .1475; ●

cost = [within_cost, within_cost;
    between_cost, between_cost;
    within_cost, between_cost
    between_cost, within_cost];

for sim_cond = 1:4

    clear acc_pos1 acc_pos2 acc_pos3

    %all trials correct before first cross
    acc_pos1 = ones(nSims, 1);

    %lose track of some trials during first cross
    lost1 = rand(nSims, 1) < cost(sim_cond, 1);

    %at position 2, trials where you didn't lose track are still correct
    acc_pos2(~lost1) = acc_pos1(~lost1);

    %at position 2, trials where you did lose track,
    %50% chance you switch to wrong dot
    acc_pos2(lost1) = rand(sum(lost1), 1) < .5;

    %lose track of some trials during second cross
    lost2 = rand(nSims, 1) < cost(sim_cond, 2);

    %at final position (position 3), dots where you did not lose track
    %remain the same (if they were right at position 2, still right. If
    %wrong at position 2, still wrong)
    acc_pos3(~lost2) = acc_pos2(~lost2);

    %at position 2, trials where you did lose track,
    %50% chance you switch to other dot. So some right switch to wrong, and
    %some wrong switch to right
    acc_pos3(lost2) = rand(sum(lost2), 1) < .5;

    %percentage of trials where correct dot is being tracked at the end
    acc_sim(sim_cond) = mean(acc_pos3);

end

plot([1 2 3 4], 100*acc_sim, 'r*', 'MarkerSize', 16)
```

Using Simulation in Research

1. Guiding Experimental Design

2. Understanding Results

3. Your examples?