How to Test d Roll

Profit-Maximizing A/B Tests

Elea McDonnell Feit & Ron Berman

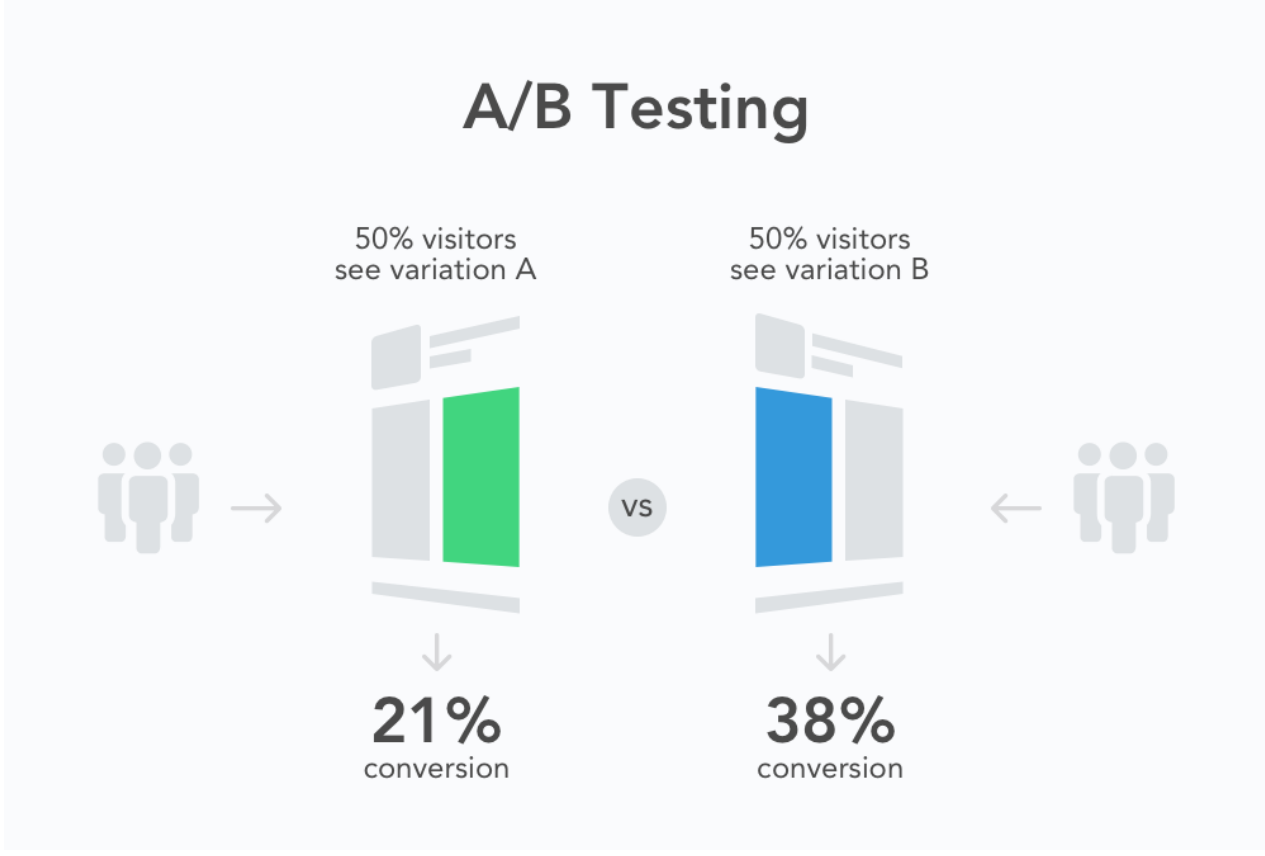
7/16/2020

Thanks to Drexel & Wharton

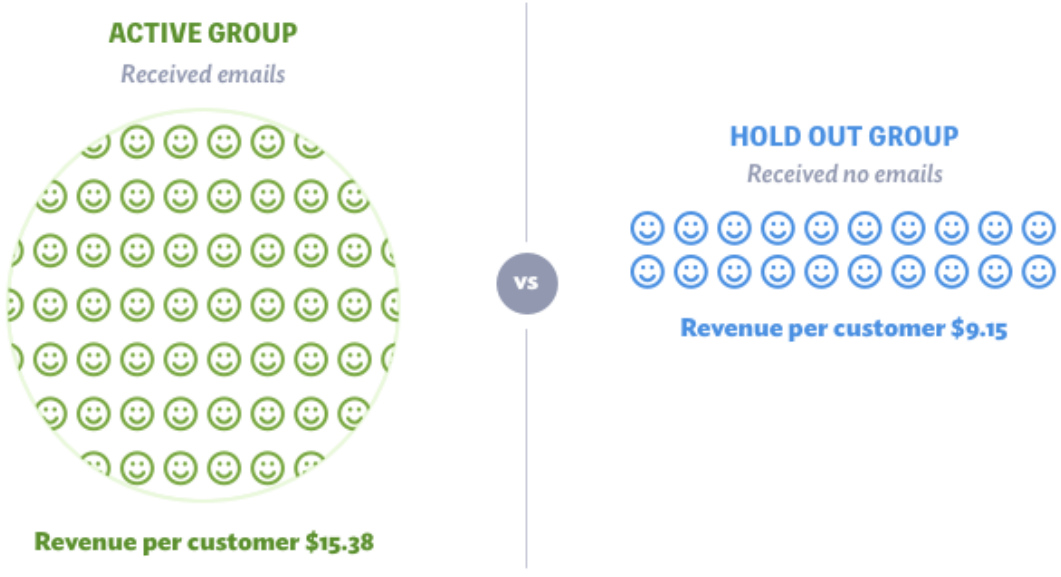
Drexel MS Business Analtyics (Fall applications due 9/15)
Drexel MS Marketing (Fall applications due 9/15)
Drexel Business Analytics Meetup (open to all!)
Drexel Solutions Institute

Wharton Online Certificate in Digital Marketing
Wharton Exec Ed Course in Customer Analtyics
Analytics at Wharton

A/B Testing


A/B Testing
50% visitors
see variation A
50% visitors
see variation B
VS
21%
conversion
38%
conversion

Holdout Testing



Workshop Goals

In this session, I'd like to introduce you to an alternative way of planning and analyzing an A/B test, which is based on *Bayesian decision theory*. We think this approach has a lot of advantages, so Ron and I gave it a name: **Test & Roll**.

We are marketing professors, after all!

Workshop materials

Slides

Those familiar with R Markdown can open the R Markdown file that created the slides in R Studio and run the code as we go along.

- Requires `rstan` and `dplyr` packages
- Requires nn_functions.R with some additional functions I wrote.

Those unfamiliar with R Markdown (or just tired) should follow the slides without trying to run the code.

What I expect participants already know

I just told you about A/B tests, *but ask questions!*

I will assume you know what a probability distribution like the normal distribution is, *but ask questions!*

I expect you are comfortable reading R. I will use mathematical calculations, for loops, and plotting, *but ask questions!

You do not need to know how plan and analyze an A/B test using hypothesis testing, but see previous R Ladies Philly workshop on A/B testing.

How to analyze an A/B test using Bayesian inference

Bayesian Analysis for A/B Tests

The data from an A/B test comparing the time users spend on your website for two versions of the homepage is in the data frame `test_data`. A summary of the data looks like this:

```
test_data %>%
  group_by(group) %>% summarize(mean=mean(time_on_site), sd
```

```
## # A tibble: 2 x 4
##   group  mean    sd     n
##   <chr> <dbl> <dbl> <int>
## 1 A      5.33  1.95   500
## 2 B      5.37  2.28   500
```

It looks like the B version keeps users on the site a bit longer, but how sure are we that B produces longer visits on average? We've only seen 500 visitors in each group.

Prior beliefs

I would like to know what the mean time-on-site is for the A group and the B group.

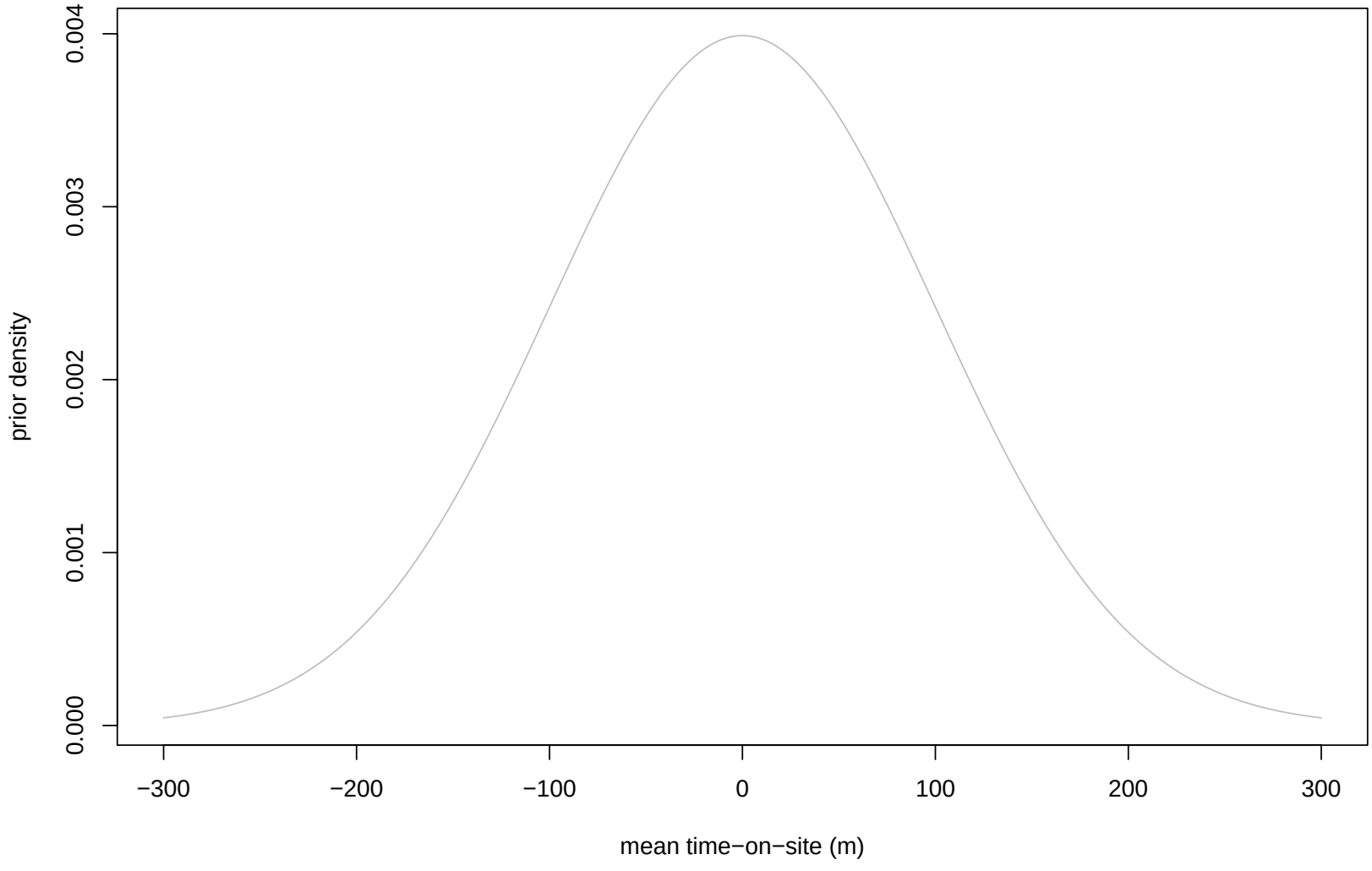
Before I saw this data, I knew nothing about how long people might spend on this website. They might stay for 5 seconds or they might stay for 5 hours.

Formally, I can describe my prior beliefs with a *prior distribution*:

$$\text{mean time-on-site for group} \sim N(0, 100^2)$$

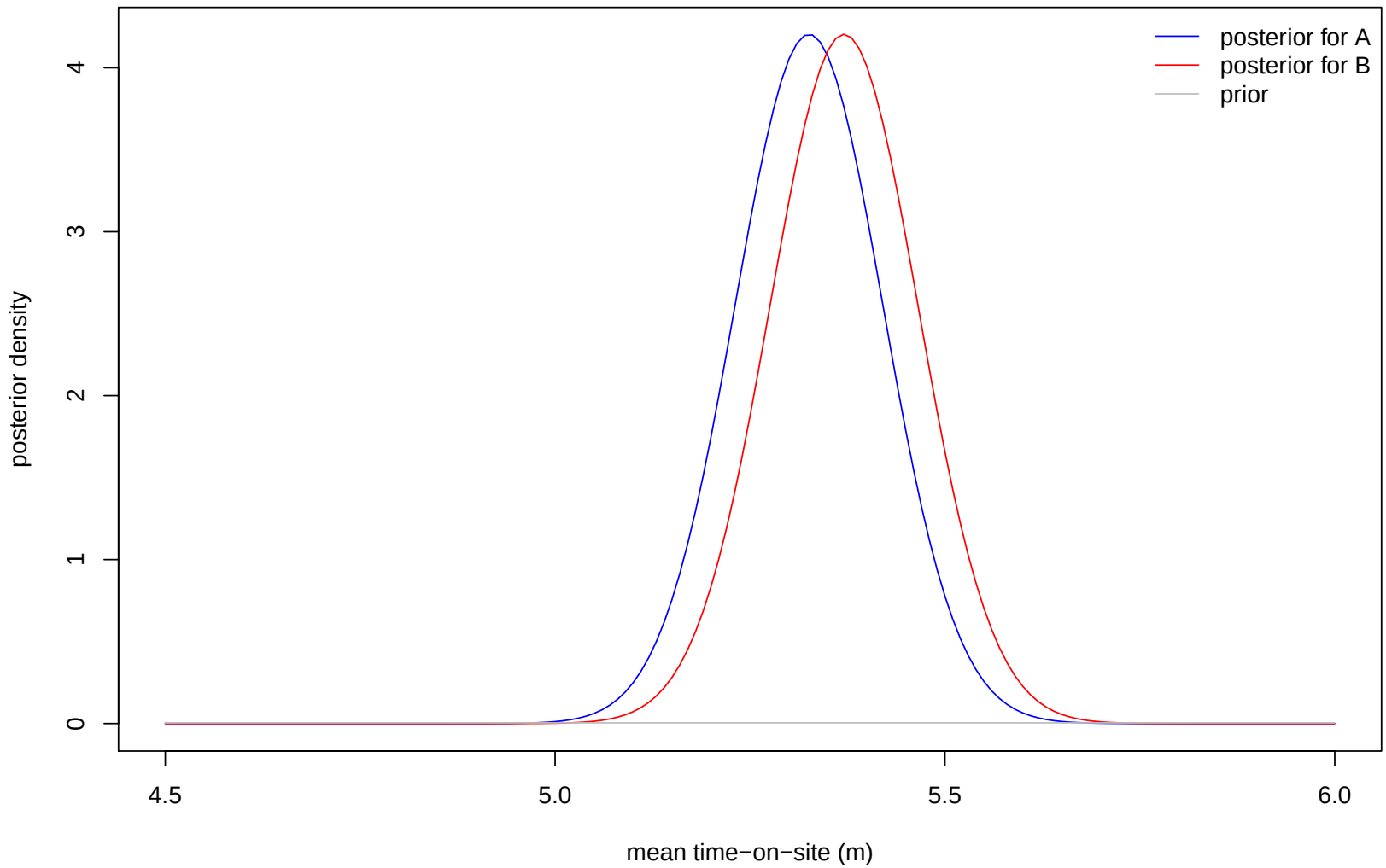
Prior beliefs

I find it easier to draw a picture of my prior.



Posterior beliefs

Bayes rule tells you how you should update your beliefs after you see some data. This is also easier to see with a picture.



Model details (mathematically)

If we assume that the time-on-site for each customer is distributed normally:

$$\text{time-on-site} \sim \mathcal{N}(\text{mean time-on-site for group}, s^2)$$

Then Bayes rule tells us that the posterior distribution for mean time-on-site for each group should be:

$$\text{mean time-on-site for group given data} \sim \mathcal{N}\left(\overline{y}, \left(\frac{1}{100^2} + \frac{n}{s^2}\right)^{-1}\right)$$

I'm skipping the derivation. Hope you don't mind!

Code for posterior updating

```
n_A <- sum(test_data$group=="A") # obs for A
n_B <- sum(test_data$group=="B") # obs for B
s <- sd(test_data$time_on_site) # standard deviation of da

# Posterior mean is just the mean for each group
post_mean_A <- mean(test_data[test_data$group=="A", "time_
post_mean_B <- mean(test_data[test_data$group=="B", "time_

# Posterior standard deviation follows this formula
post_sd_A <- (1/100^2 + n_A/s^2)^-(1/2)
post_sd_B <- (1/100^2 + n_B/s^2)^-(1/2)
```

Credible intervals for groups

If you like intervals, we can compute a 95% credible intervals for each group, by cutting off the left and right 2.5% of the posterior distribution. In this case, our posterior is normal, so we use the `qnorm()` function, which computes quantiles of the normal distribution.

There is a 95% probability that the average time-on-site for treatment A is:

```
qnorm(c(0.025, 0.975), mean=post_mean_A, sd=post_sd_A) # C
```

```
## [1] 5.139694 5.511620
```

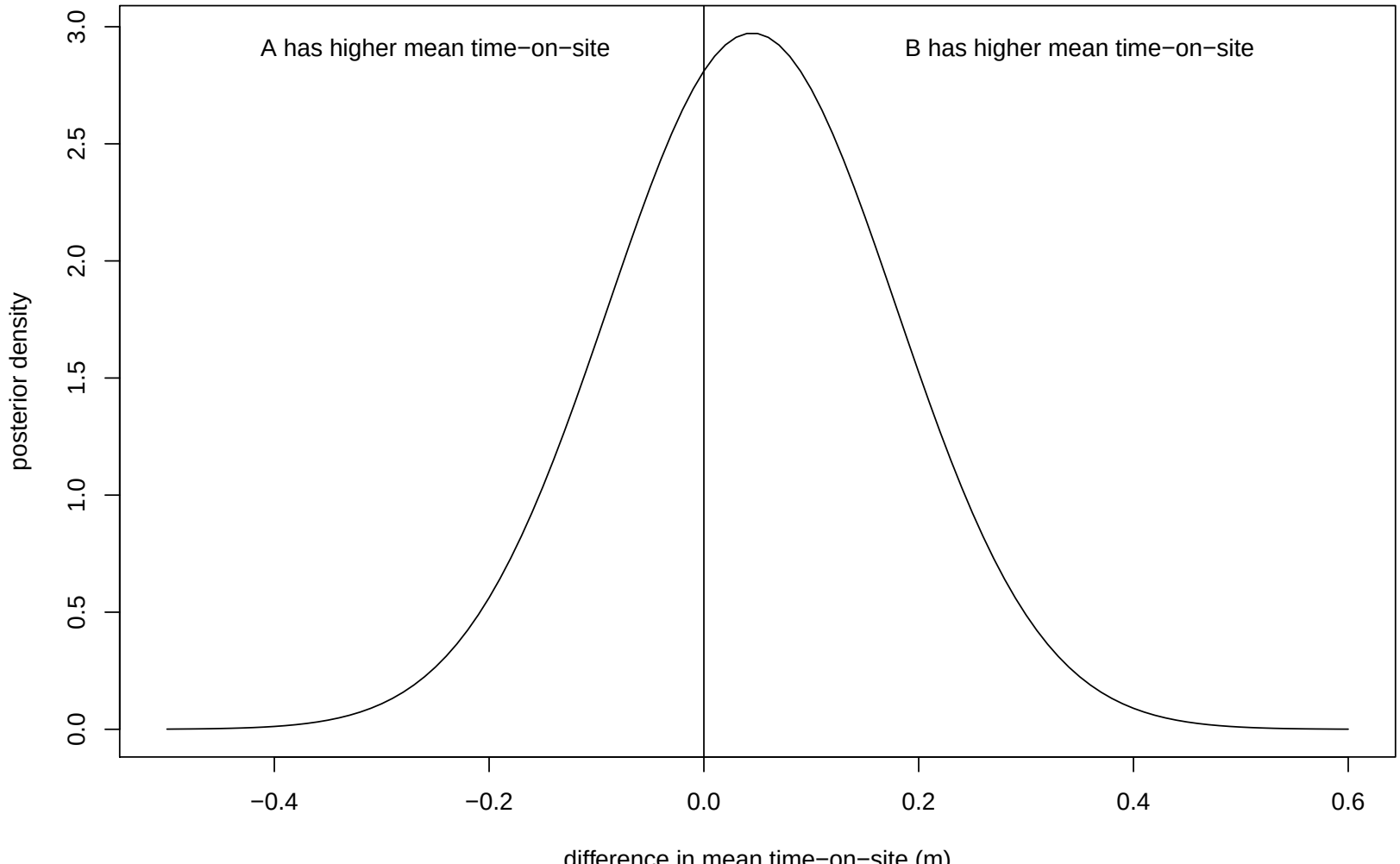
There is a 95% probability that the average time-on-site for treatment B is:

```
qnorm(c(0.025, 0.975), mean=post_mean_B, sd=post_sd_B) # C
```

```
## [1] 5.184622 5.556548
```

Posterior for the difference between A and B

We can also compute the posterior distribution for the difference between the mean time-on-site for the B group and the mean time-on-site for the A group.



Posterior for the difference between A and B

Since the posterior for A and the posterior for B are both normal, the posterior for the difference is also normal with mean and standard deviation:

```
post_mean_diff <- post_mean_B - post_mean_A
post_sd_diff <- sqrt(post_sd_B^2 + post_sd_A^2)
```

Once we have the distribution for the difference in the mean time-on-site, we can compute the probability that the mean of B is greater than the mean of A.

```
1-pnorm(0, mean=post_mean_diff, sd=post_sd_diff)
```

```
## [1] 0.6311218
```

It is up to the decision maker to decide whether they would like to use version B knowing that there is a 63% change that B is better than A. This depends on how costly it is to deploy B.

More on prior beliefs

Many people get hung up on priors. **Don't!**

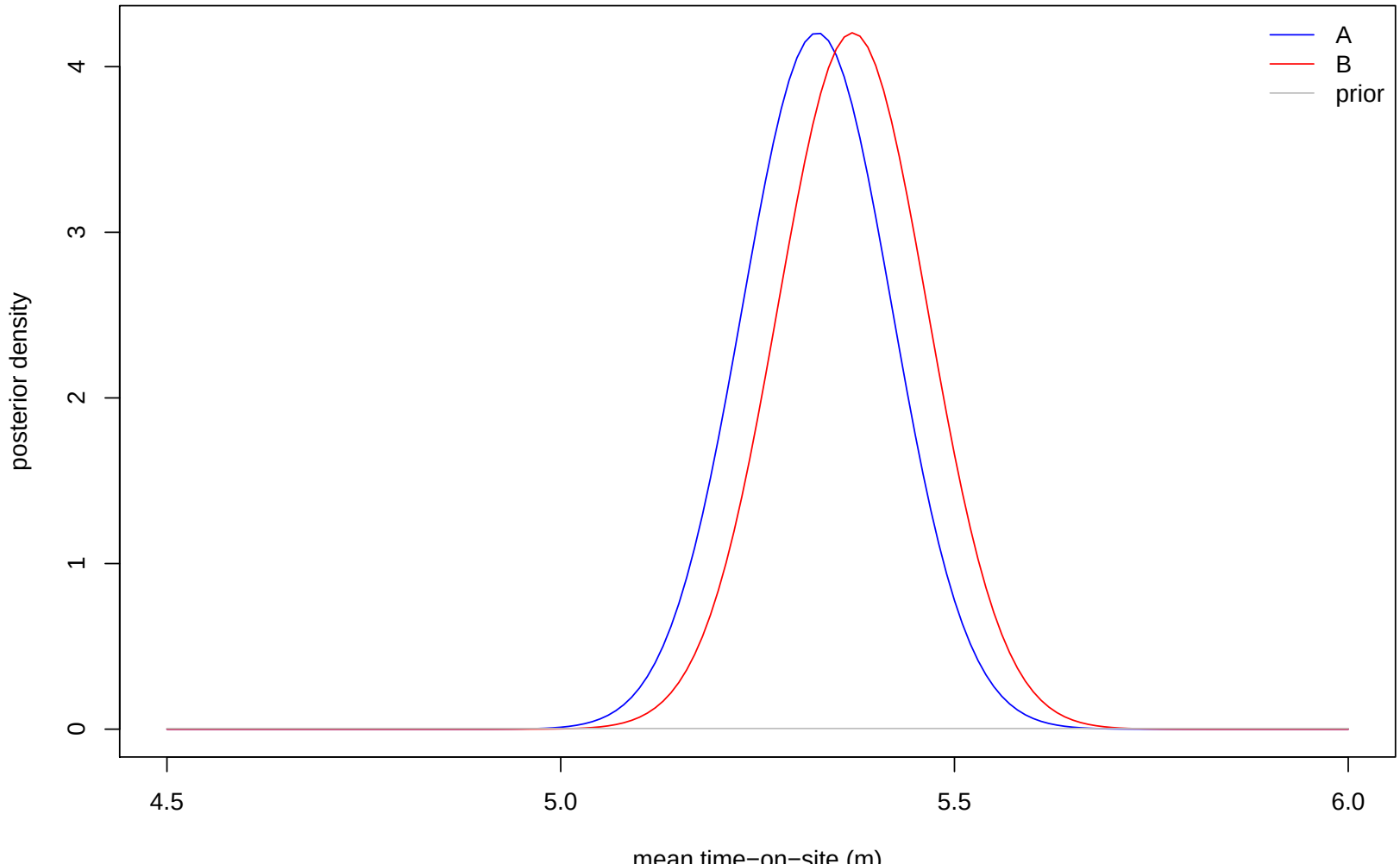
As you get more data, the posterior becomes more and more influenced by the data. In most practical situations you have enough data that the priors do not affect the analysis much.

If you have prior information, priors are a principled way to bring it in.

Priors are also useful when planning an A/B test (more later).

More on prior beliefs

When we don't want our priors to influence the outcome, we use "flat" priors. The prior we jused puts nearly equal weight on 4.5 minutes versus 6 minutes, so it is pretty flat.



More on prior beliefs

In this analysis I used the same prior for both A and B because I know nothing about A and B.

Important: using the same prior for A and B is *not the same* as assuming A and B have the same mean time-on-site. Priors reflect our uncertainty about A and B. Becuase we are uncertain, A might be better than B or B better than A.

You can use priors that reflect your (justifiable) prior beliefs. For instance, if A is a discount and B is no discount and our outcome is how much you purchase, then I'm pretty sure A will be as good as or better than B.

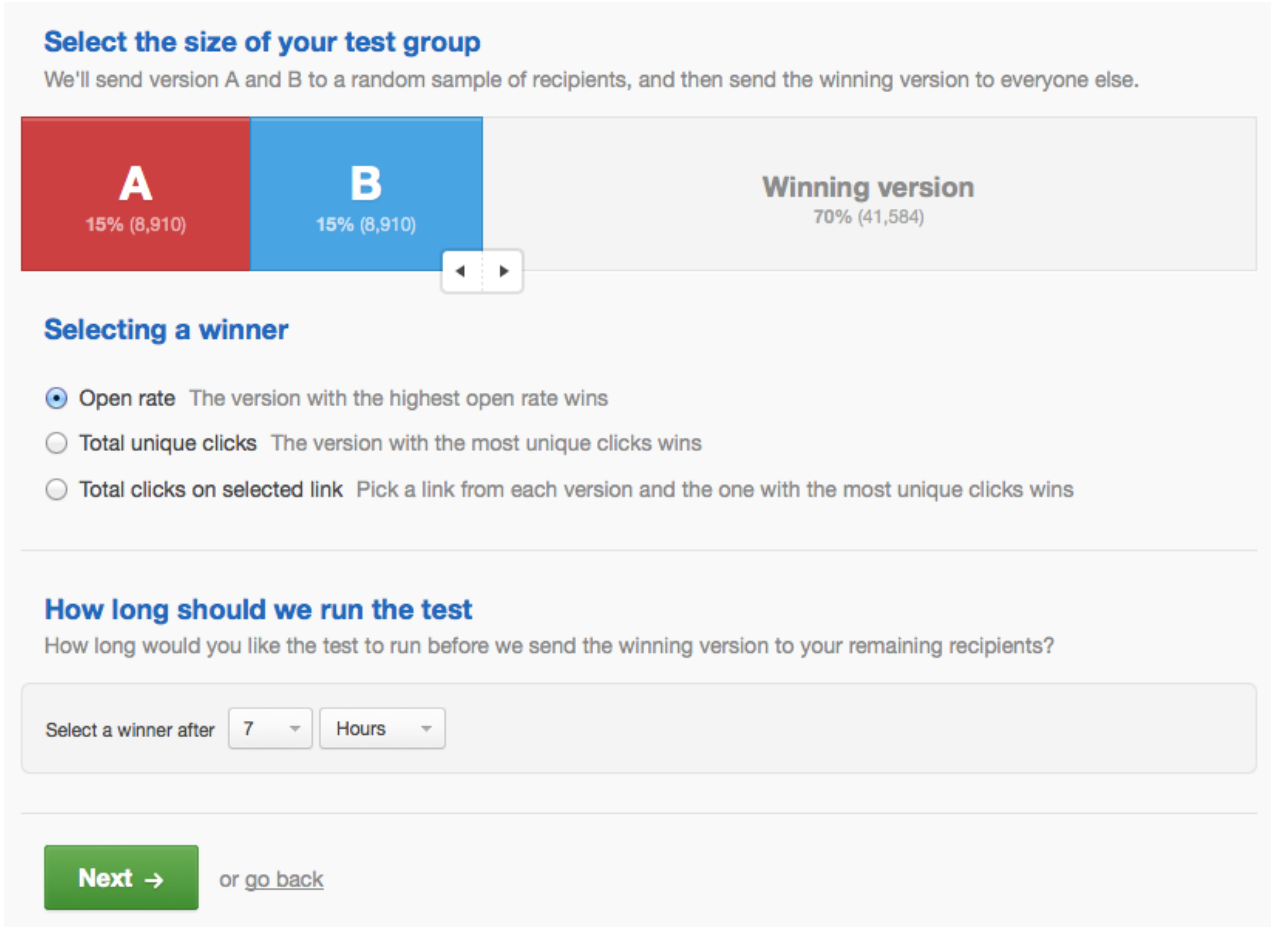
How to analyze an A/B test using Bayesian analysis

1. Pick treatments and an outcome to measure
2. Randomly assign A and B to some users and record outcomes
3. Quantify your beliefs about the outcome with a prior distribution
4. Update your beliefs according to Bayes rule (posterior distribution)
5. Plot your updated belief distribution; compute intervals and probabilities
6. Make a decision or **go to 3**

Questions?

How to Test and Roll

Typical A/B email test setup screen



A/B test planning as a decision problem

Test

Choose n_1^* and n_2^* customers to send the treatments.
Collect data on profit for both treatments.

Roll

Choose a treatment to deploy to the remaining $N - n_1^* - n_2^*$ customers.

Objective

Maximize combined profit for test stage and the roll stage.

Profit-maximizing sample size

If you have a test where the **profit** earned for each customer is:
$y \sim \mathcal{N}(m_1, s)$ for group 1
$y \sim \mathcal{N}(m_2, s)$ for group 2

and your priors are
$(m_1, m_1 \sim N(\mu, \sigma))$

the profit maximizing sample size is:

$$n_1 = n_2 = \sqrt{\frac{N}{4}\left(\frac{s}{\sigma}\right)^2 + \left(\frac{3}{4}\left(\frac{s}{\sigma}\right)^2\right)^2} - \frac{3}{4}\left(\frac{s}{\sigma}\right)^2$$

This new sample size formula is derived in Feit and Berman (2019) *Marketing Science*.

Compute the sample size in R

```
source("S:/testandroll/nn_functions.R") # some functions I
N <- 100000 # available population
mu <- 0.68  # average conversion rate across previous trea
sigma <- 0.03 # range of expected conversation rates acros
s <- mu*(1-mu) # binomial approximation
test_size_nn(N=N, s=s, mu=mu, sigma=sigma) # compute the o
```

```
## [1] 1108.073 1108.073
```

Why is this the profit-maximizing test size?

`test_eval_nn()` computes the profit of a Test & Roll.

```
# Optimal test size
n_star <- test_size_nn(N=N, s=s, mu=mu, sigma=sigma)
test_eval_nn(n=n_star, N=N, s=s, mu=mu, sigma=sigma)
```

```
##          n1       n2 profit_per_cust   profit profit_test
## 1 1108.073 1108.073       0.6961711 69617.11    1506.979
##   profit_rand profit_perfect profit_gain      regret err
## 1       68000       69692.57   0.9554201 0.001082677 0.0
##   tie_rate
## 1        0
```

Why is this the profit-maximizing test size?

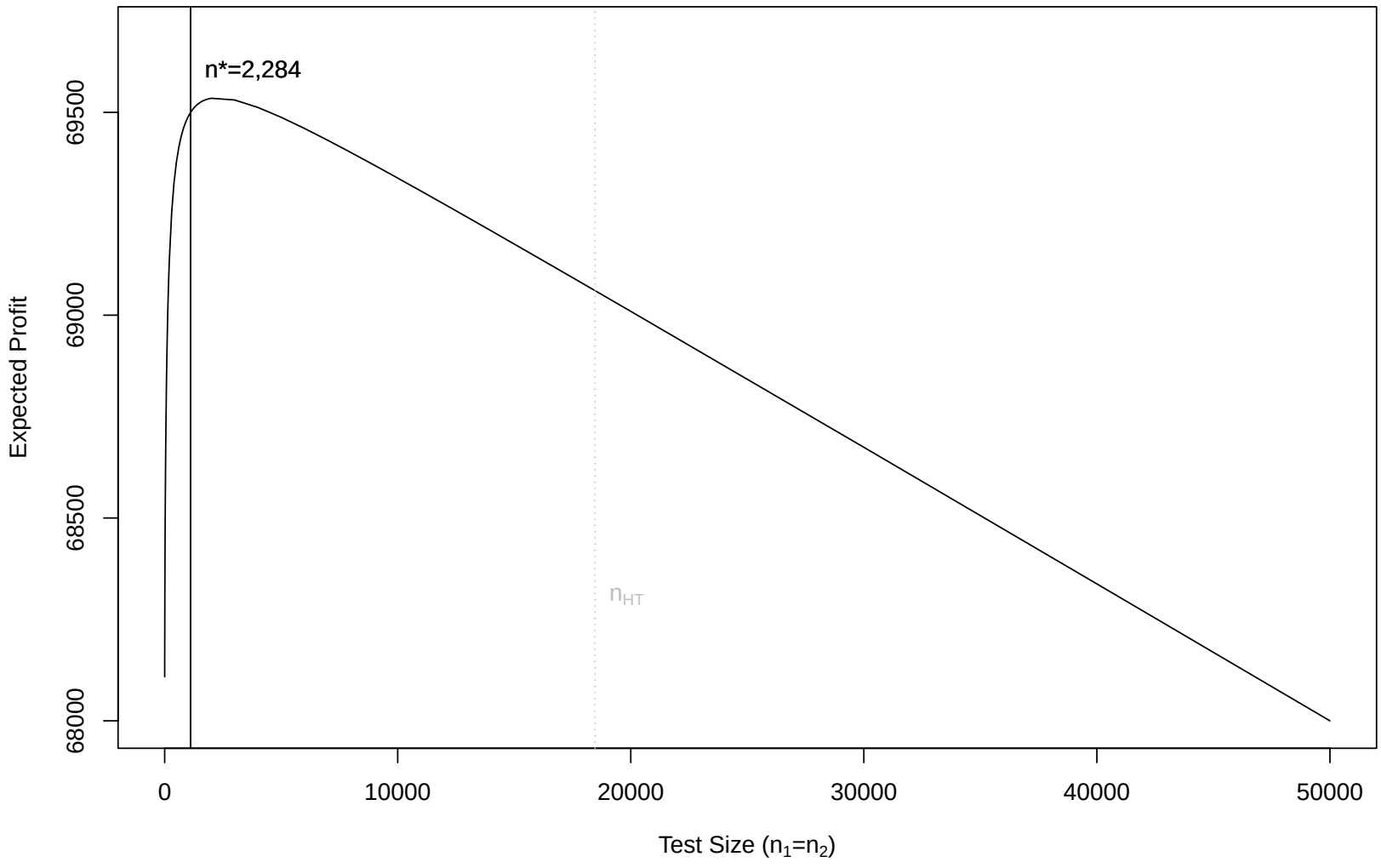
```
# Bigger test
test_eval_nn(n=c(10000, 10000), N=N, s=s, mu=mu, sigma=sigm
```

```
##      n1    n2 profit_per_cust   profit profit_test profi
## 1 10000 10000       0.6935051 69350.51       13600
##   profit_perfect profit_gain      regret error_rate depl
## 1       69692.57   0.7979038 0.004908151 0.02304771
```

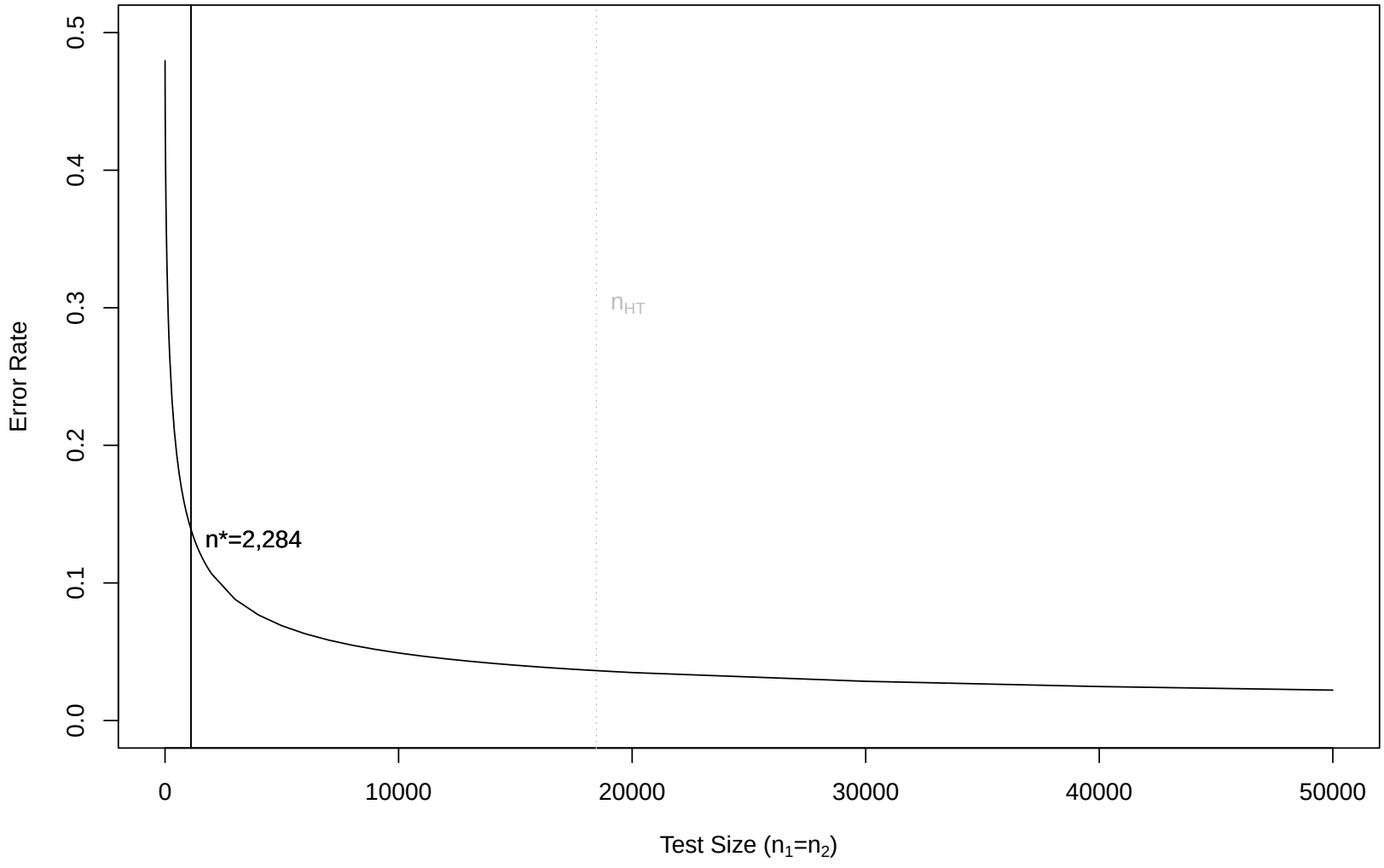
```
# Smaller test
test_eval_nn(n=c(100, 100), N=N, s=s, mu=mu, sigma=sigma)
```

```
##    n1  n2 profit_per_cust   profit profit_test profit_de
## 1 100 100       0.6936736 69367.36         136      6923
##   profit_perfect profit_gain      regret error_rate depl
## 1       69692.57   0.8078632 0.004666275  0.1997479
```

Why is this the profit-maximizing test size?



How often do you deploy the wrong treatment?



Profit-maximizing sample size

The `test_size_nn()` function is making the optimal trade-off between:

- the opportunity cost of the test (You are sending the wrong treatment to half your customers!)

and

- the risk of deploying the wrong treatment

How to choose `N`, `mu`, `sigma` and `s`

What is N?

N is the total number of customers you have available, i.e. the size of your email mailing list or the the number of visits that might visit a webpage in the next month.

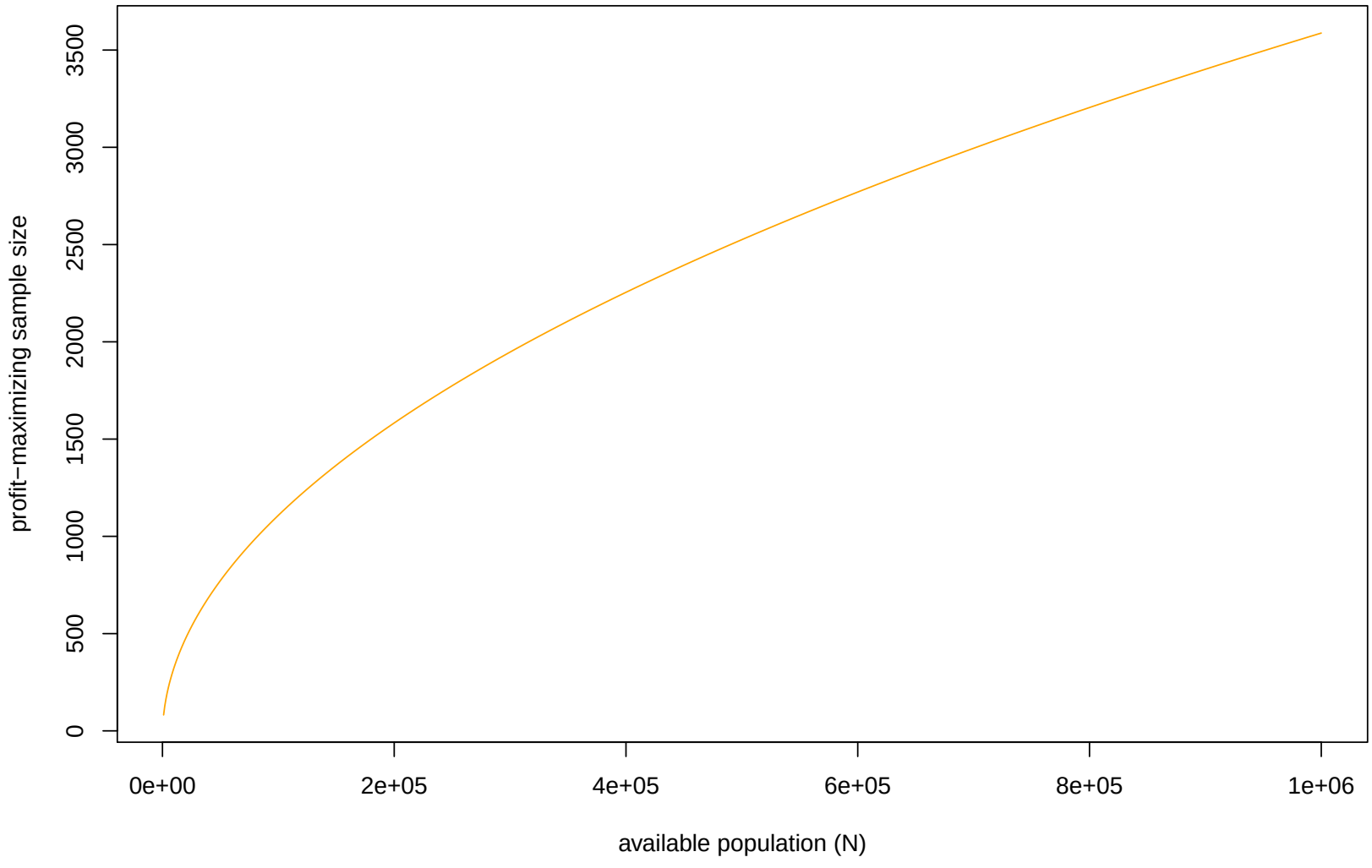
Let's vary N and look at how the profit-maximizing test size changes:

```
Ns <- (1:1000)*1000
test_sizes <- rep(NA, length(Ns))
for (i in 1:length(Ns))
  test_sizes[i] <- test_size_nn(N=Ns[i], s=s, mu=mu, sigma=
```

```
plot(x=Ns, y=test_sizes, type="l", col="orange",
     xlab="available population (N)", ylab="profit-maximiz
```

What is N?

Bigger N $\rightarrow$ bigger test



What is mu?

mu is the average profit you expect across treatments you might test.

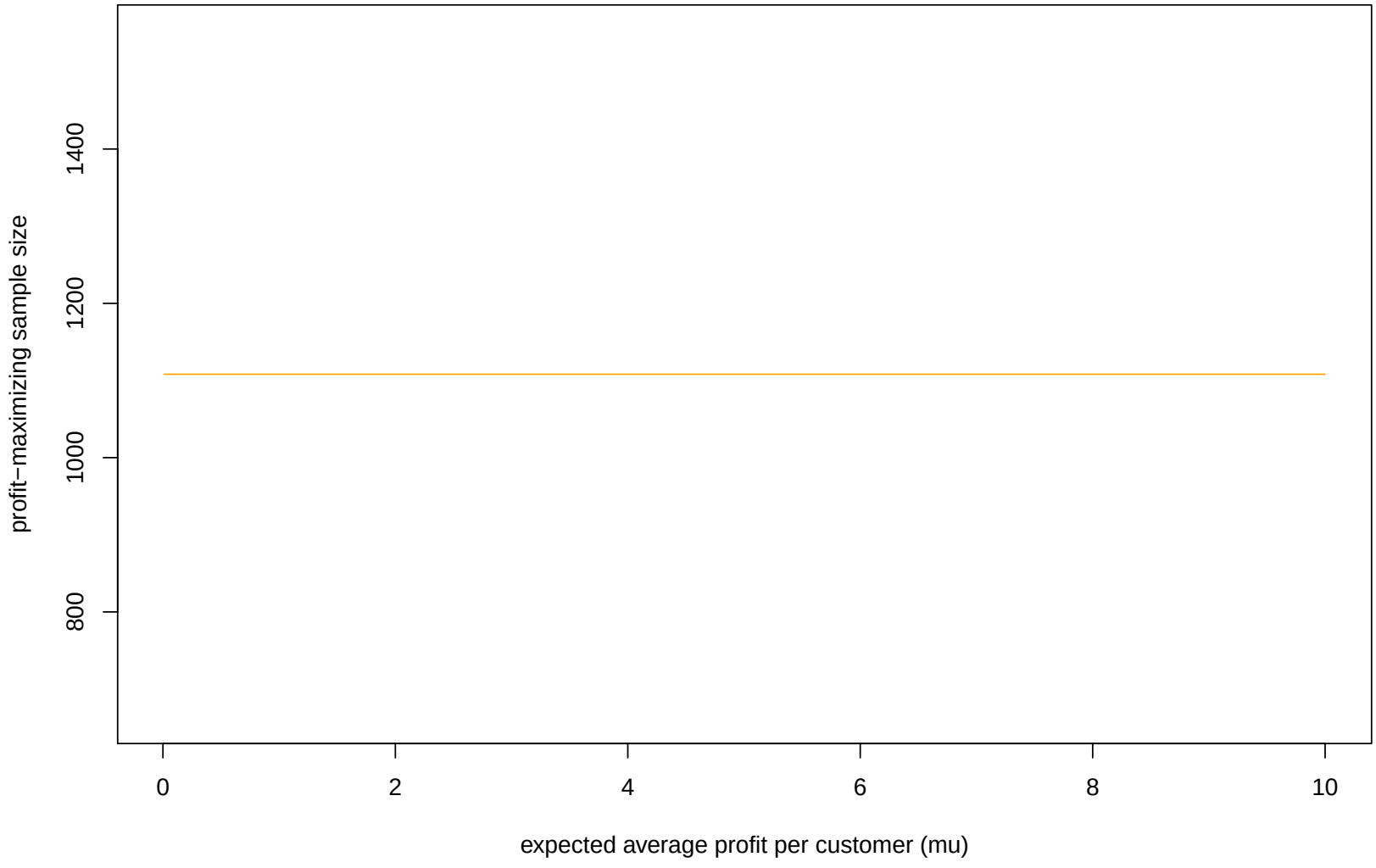
Let's vary mu:

```
mus <- (1:1000)/100
test_sizes <- rep(NA, length(mus))
for (i in 1:length(mus))
  test_sizes[i] <- test_size_nn(N=N, s=s, mu=mus[i], sigma=
```

```
plot(x=mus, y=test_sizes, type="l", col="orange",
     xlab="expected average profit per customer (mu)", ylab
```

What is `mu`?

`mu` doesn't effect the test size (when `mu` is the same for both A and B)



What is s?

s is how variable the profit is from one customer to another.

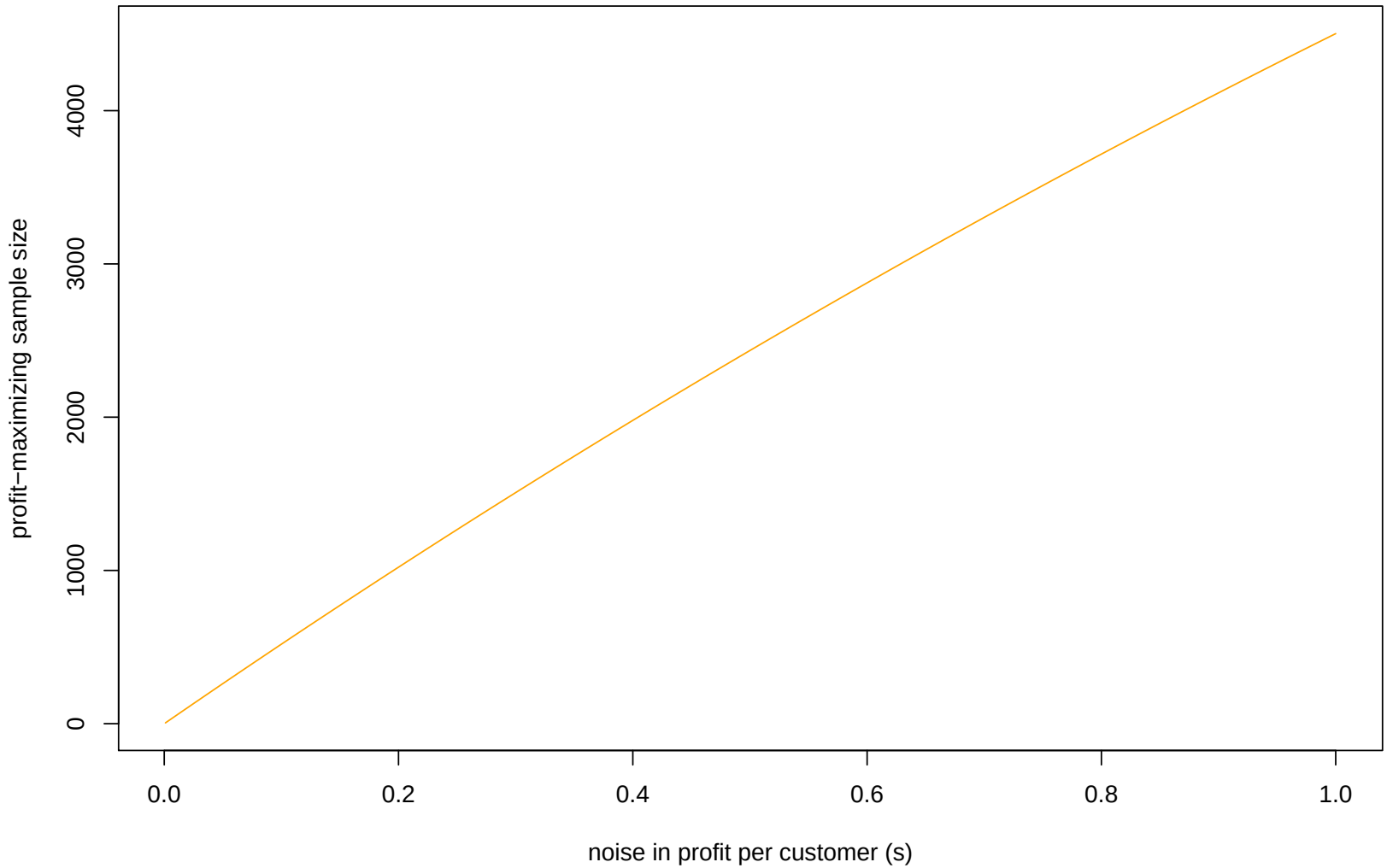
Let's vary s:

```
ss <- (1:1000)/1000
test_sizes <- rep(NA, length(ss))
for (i in 1:length(ss))
  test_sizes[i] <- test_size_nn(N=N, s=ss[i], mu=mu, sigma=
```

```
plot(x=ss, y=test_sizes, type="l", col="orange",
     xlab="noise in profit per customer (s)", ylab="profit-
```

What is s?

Bigger s $\rightarrow$ harder to see what is going on $\rightarrow$ bigger test

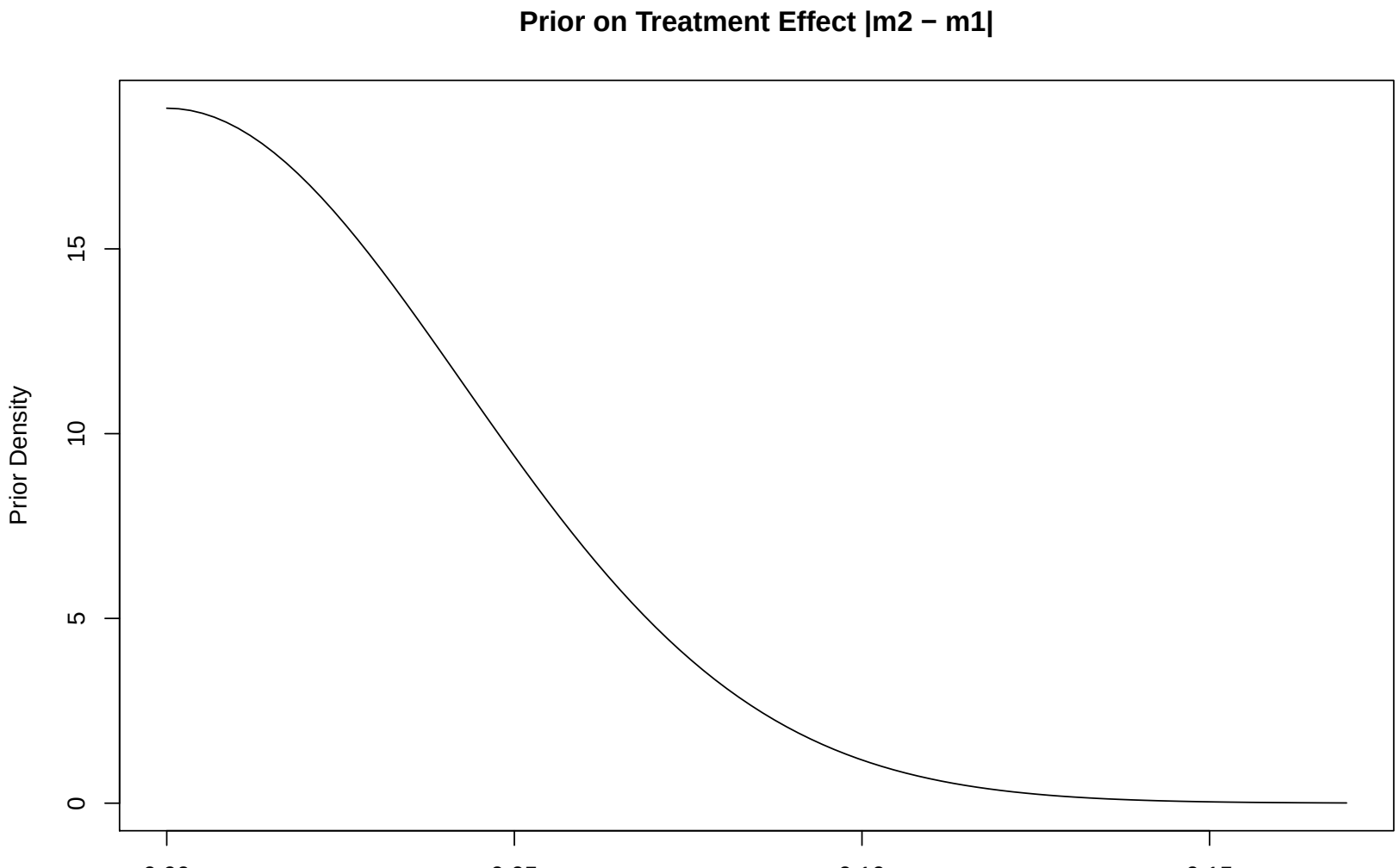


Hypothesis testing requires much bigger sample sizes (proportional

What is sigma?

sigma defines the difference we expect between average profit for the two treatments.

```
plot_prior_effect_nn(mu, sigma, abs=TRUE)
```



What is sigma?

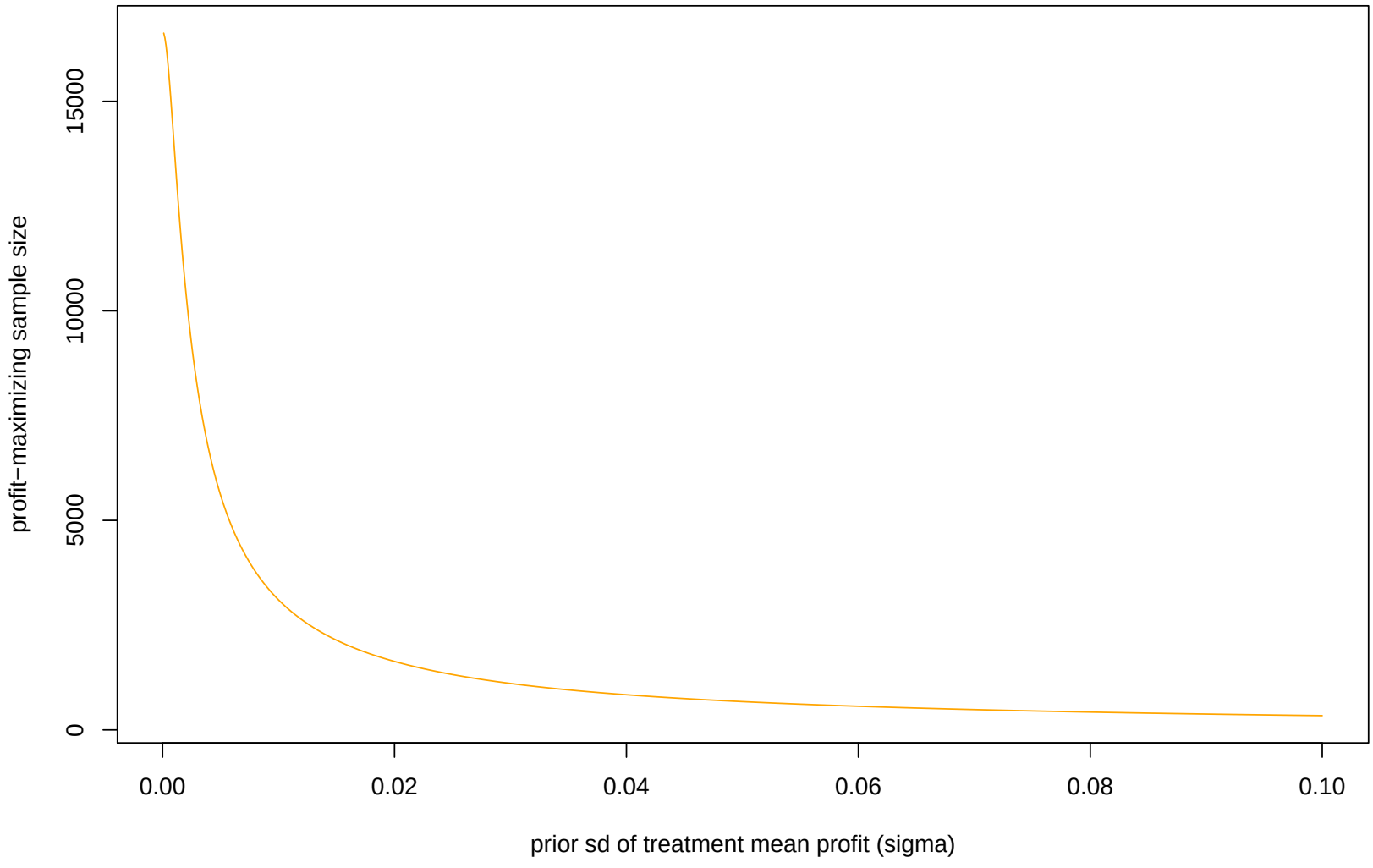
Let's vary sigma:

```
sigmas <- (1:1000)/10000
test_sizes <- rep(NA, length(sigmas))
for (i in 1:length(sigmas))
  test_sizes[i] <- test_size_nn(N=N, s=s, mu=mu, sigma=sigm
```

```
plot(x=sigmas, y=test_sizes, type="l", col="orange",
     xlab="prior sd of treatment mean profit (sigma)", ylab
```

What is `sigma`?

Bigger `sigma` $\rightarrow$ bigger difference between A and B $\rightarrow$ smaller test



How to come up with `s` and `sigma` for your own test

Data on previous treatments

Suppose you have data on some previous A/B tests that represent the range of results you might expect from the treatments in the test you are planning.

```
head(expts$y)
```

```
##      A_conv_rate B_conv_rate
## [1,]   0.3071216   0.3239707
## [2,]   0.4095521   0.4645631
## [3,]   0.5314887   0.5666940
## [4,]   0.4139385   0.4467867
## [5,]   0.8332743   0.8229312
## [6,]   0.8431361   0.8289914
```

This data is synthetic data that is similar to data we have from a large A/B testing platform. In the data we have `conv_rate` is the number of users who click anywhere on a website.

Meta-analyze past experiments

Assume that this data represents the range of performance that we might see in the test we are planning.

We can fit a model to this data to estiamte `mu` and `sigma` for these types of tests.

We'll use Stan, which is a flexible, open-source tool for Bayesian modeling. You can access Stan by installing the `rstan` package.

Hierarchical Stan model for past experiments

```
data {
  int<lower=1> nexpt; // number of experiments
  real<lower=0,upper=1> y[nexpt,2]; // observed mean respon
  int nobs[nexpt,2]; // sample size for each arm (1 and 2)
}
parameters {
  real<lower=0, upper=1> m[nexpt,2];
  real<lower=0, upper=1> t[nexpt];
  real<lower=0, upper=1> mu;
  real<lower=0> sigma;
  real<lower=0> omega;
}
model {
  // priors
  mu ~ normal(0.5, 0.1);
  omega ~ normal(0, 0.1);
  sigma ~ normal(0, 0.1);
  // likelihood
```

Fit Stan model to past experiments

```
m1 <- sampling(stan_model, data=expts, seed=20030601, iter=
```

```
## Warning: Bulk Effective Samples Size (ESS) is too low, i
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

Fitted parameters

We're interested in `mu`, which is the average profit per customer across all the tests and `sigma`, which is the variation between the A and B groups *within* a test.

```
summary(m1, pars=c("sigma", "mu"))$summary[,c(1,3,5,8)]
```

```
##                mean          sd         25%       97.5%
## sigma 0.02847655 0.00205630 0.02704171 0.03281609
## mu    0.68087114 0.01653565 0.66970172 0.71340606
```

Compute optimal sample size

```
(n_star <- test_size_nn(N=100000, mu=0.68, sigma=0.029, s=0
```

```
## [1] 1144.924 1144.924
```

```
test_eval_nn(n_star, N=100000, mu=0.68, sigma=0.029, s=0.68
```

```
##          n1       n2 profit_per_cust   profit profit_test
## 1 1144.924 1144.924       0.6956077 69560.77    1557.097
##   profit_rand profit_perfect profit_gain      regret err
## 1       68000       69636.15   0.9539282 0.001082488 0.0
##   tie_rate
## 1        0
```

Then we run the test somewhere out in the world and get some data...

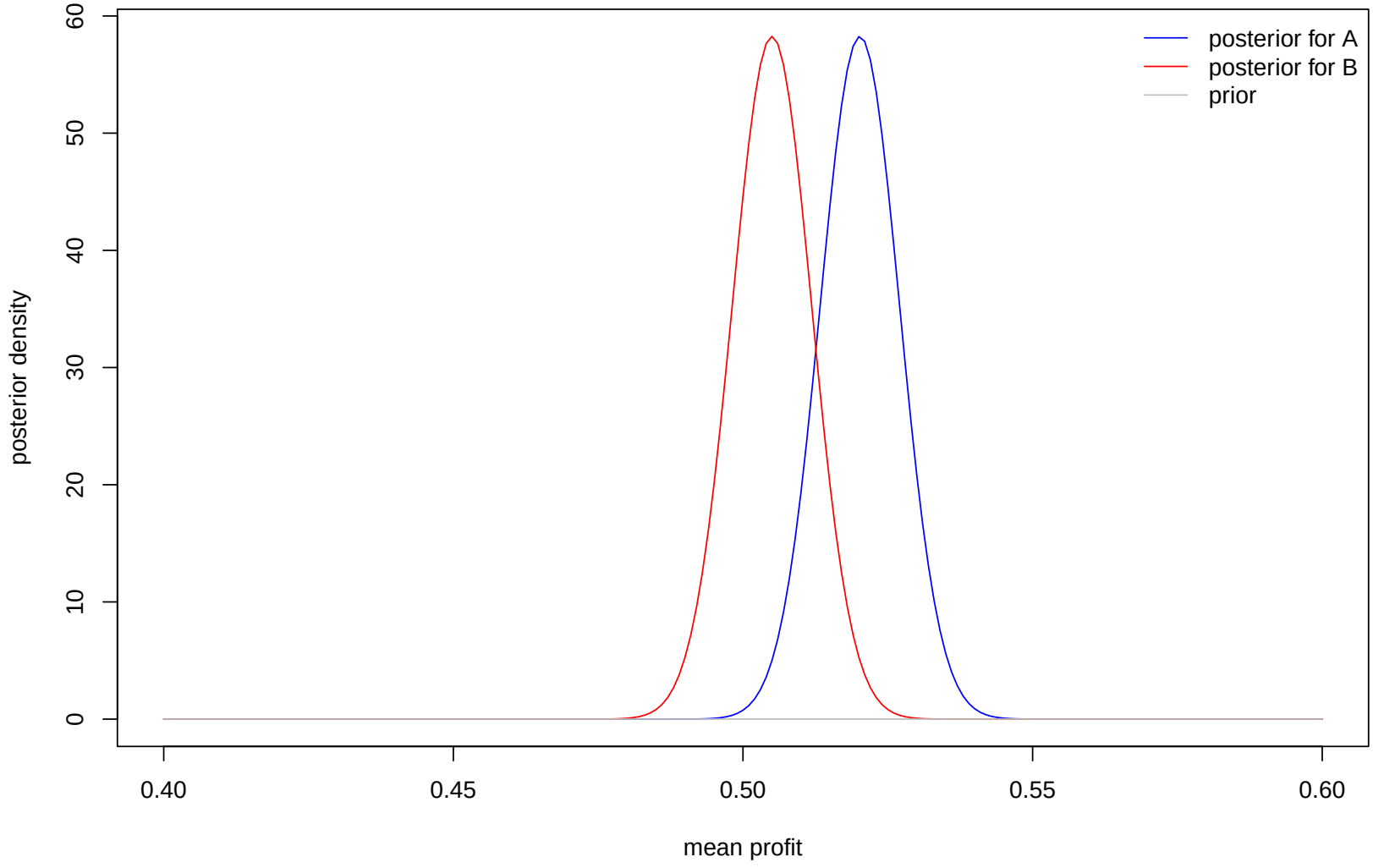
```
test_data %>%
  group_by(group) %>% summarize(mean=mean(profit), sd=sd(pr

## `summarise()` ungrouping output (override with `.groups`

## # A tibble: 2 x 4
##   group  mean    sd     n
##   <chr> <dbl> <dbl> <int>
## 1 A     0.520 0.244  1272
## 2 B     0.505 0.245  1272
```

How to decide what to roll

To decide which treatment to delpoy, we compute the posteriors for each group.

```
n_A <- sum(test_data$group=="A")
n_B <- sum(test_data$group=="B")
s <- sd(test_data$profit)
post_mean_A <- mean(test_data[test_data$group=="A", "profit
post_mean_B <- mean(test_data[test_data$group=="B", "profit
post_sd_A <- (1/100^2 + n_A/s^2)^-(1/2)
post_sd_B <- (1/100^2 + n_B/s^2)^-(1/2)
```

How to decide what to roll



How to decide what to roll

The posterior mean is our best guess at how well the treatment will perform. Our expected profit is maximized when we choose the treatment with the higher posterior mean (cf. Berger).

But, when the priors for A and B are the same, then the posterior is just the average profit in the test.

```
post_mean_A <- mean(test_data[test_data$group=="A", "profit
post_mean_B <- mean(test_data[test_data$group=="B", "profit
```

So, you can skip the Bayesian analysis! Just compute the average profit for each group and deploy the treatment with the higher profit.

That's why we call it Test & **Roll**.

Summary: How to Test and Roll

1. Come up with priors

If you have data on past treatments that are similar to the ones you plan to test, fit a hierarchcial model.

```
# define a stan model (not shown)
# estimate the model with your prior data
m1 <- sampling(stan_model, data=expts, seed=20030601, iter=
```

```
summary(m1, pars=c("sigma", "mu"))$summary[,c(1,3,5,8)]
```

```
##                mean          sd         25%       97.5%
## sigma 0.02847655 0.00205630 0.02704171 0.03281609
## mu    0.68087114 0.01653565 0.66970172 0.71340606
```

The data doesn't have to be A/B tests. For example, you could use past response rates on regular email campaigns.

2. Use priors to compute sample size

```
(n_star <- test_size_nn(N=100000, mu=0.68, sigma=0.026, s=0
```

```
## [1] 1271.801 1271.801
```

```
test_eval_nn(n_star, N=100000, mu=0.68, sigma=0.026, s=0.68
```

```
##          n1       n2 profit_per_cust   profit profit_test
## 1 1271.801 1271.801       0.6939177 69391.77    1729.649
##   profit_rand profit_perfect profit_gain      regret err
## 1       68000       69466.89   0.9487871 0.001081434 0.0
##   tie_rate
## 1        0
```

3. Run the test & collect data

This can be done using any A/B testing tool that works with the medium you are testing.

4. Deploy treatment with higher average profit

```
test_data %>%
  group_by(group) %>% summarize(mean=mean(profit), sd=sd(pr
```

```
## `summarise()` ungrouping output (override with `.groups`
```

```
## # A tibble: 2 x 4
##   group  mean    sd     n
##   <chr> <dbl> <dbl> <int>
## 1 A     0.520 0.244  1272
## 2 B     0.505 0.245  1272
```

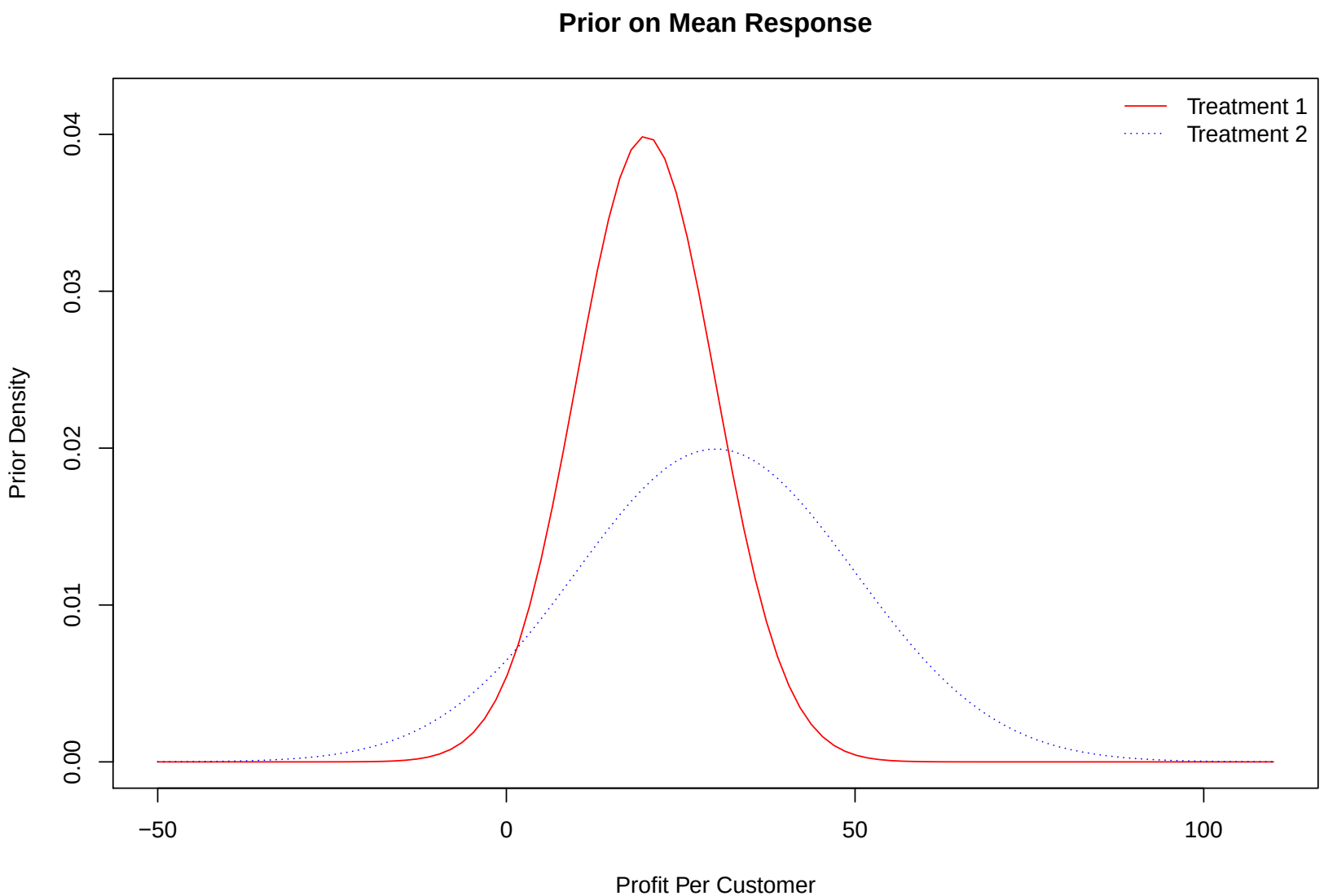
That was just five lines of code

```
m1 <- sampling(stan_model, data=expts, seed=20030601, iter=
summary(m1, pars=c("omega", "sigma", "mu"))$summary[,c(1,3,
(n_star <- test_size_nn(N=100000, mu=0.68, sigma=0.026, s=0
test_eval_nn(n_star, N=100000, mu=0.68, sigma=0.026, s=0.68
test_data %>%
  group_by(group) %>% summarize(mean=mean(profit), sd=sd(pr
```

A few more things you should know how to do

You can have different priors for A and B

```
mu <-  c(20, 30); sigma <-  c(10, 20); s <- c(100, 200)
plot_prior_mean_resp_nn(mu=mu, sigma=sigma)
```



And different priors lead to different sample sizes

```
test_size_nn(N=100000, mu=mu, sigma=sigma, s=s)
```

```
## [1]  641.7535 2549.7398
```

Why not Hypothesis Testing?

What if recommended sample size is larger than available population?

Which treatment should be deployed if the difference is non-significant?

False positives do not reduce profit in this setting. Why control them?

Can't rationalize unequal test group sizes

Hypothesis Test versus Test & Roll

Hypothesis tests are for science

- Want to know which treatment is better with high confidence
- Don't explicitly consider the cost of the experiment
- Generally requires much larger sample sizes

Test & Roll is for marketing

- Want to select treatments that maximize profits
- Considers the cost of the test relative to the error rate
- Generally requires small sample sizes that are scaled

Learning more

If you don't like R code check out the Test & Roll Shiny App

If you want to dive into the details of Test & Roll, check out our paper Feit and Berman (2019).

There is complete replication code for the paper

Chris Said (Stitch Fix) wrote a blog post outlining a similar approach framed as discounting.

Coming soon

R package to make the functions in `nn_functions.R` more accessible.

Improvements to the Shiny App

A new paper on latent stratification

Thanks!

Elea McDonnell Feit
Associate Professor of Marketing
LeBow College of Business
Drexel University
@eleafeit
eleafeit.com

Ron Berman
Assistant Professor of Marketing
The Wharton School
University of Pennsylvania
@marketsensi
ron-berman.com