Introduction to Bayesian A/B testing

Johann de Boer

2022-10-22

Table of contents

1 Setting the scene

1.1 Randomised Control Trials (RCTs)

A simplistic example:

- Users are assigned at **random** to two groups, A and B, with equal probability.
- Let A be our **control** group and B be our **treatment** group.

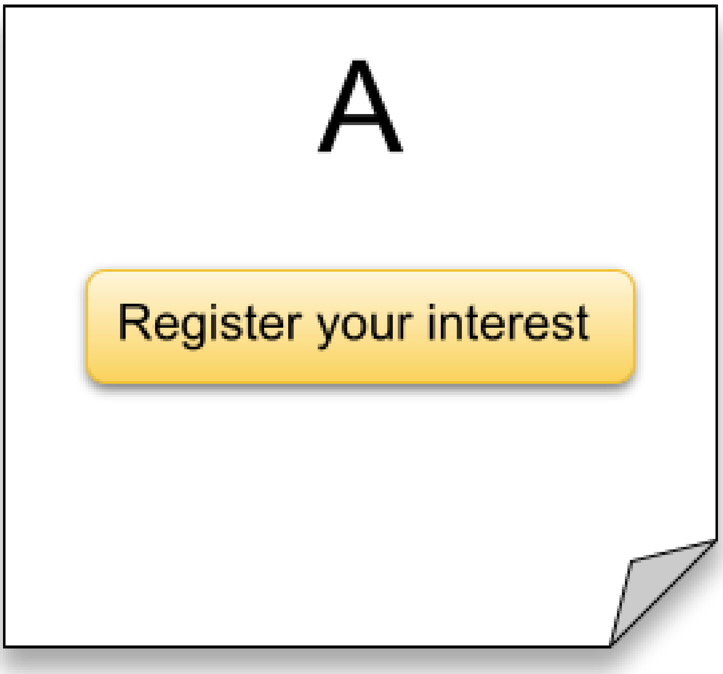
We want to know what effect our treatment has.

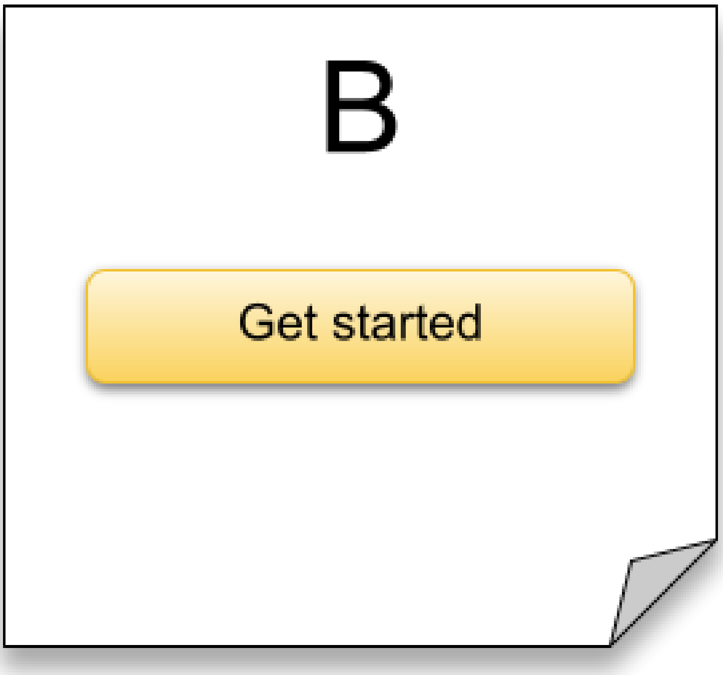
Early on during an experiment, differences between these groups could simply be due to the random allocation of participants. As the groups get larger, these random differences will diminish, bringing us closer to the difference caused by the treatment.

Applying Bayesian inference effectively gives the experiment a guided head start by including more data (probabilistic data, not real data) in the form of **priors**.

1.2 Hypothetical scenario

- A button on a landing page that takes users to a sign up form.
- At present, the button is labelled “Register your interest”.
- Test whether changing it to “Get started” will result in an increased click-through rate (CTR).
- “Get started” was suggested by an experienced and skilled UX designer.


A
Register your interest


B
Get started

2 Priors and probability distributions

The key to speeding up your experiment

2.1 Prior knowledge and beliefs

Before running an experiment, we form opinions and gather evidence such as:

- The baseline click-through rate of the button (with its current label) and knowledge of any outside variables that affects click-through rate, e.g. seasonality
- Effects we have seen from similar previous experiments
- Qualitative research, such as usability tests, focus groups, and surveys that are related to the test
- Opinions (including critical) from stakeholders and experts

2.2 Priors are probability distributions

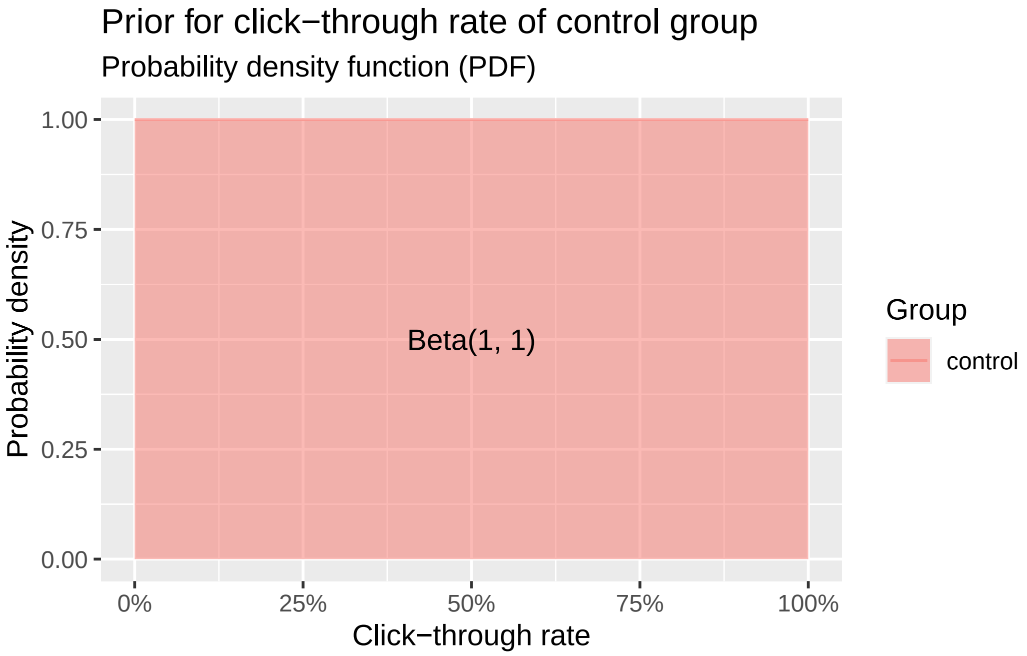
Express prior beliefs about the click-through rate of the control group using a **probability distribution**.

Here's an example of an extremely uninformative prior – a uniform prior that says any range of click-through rate is as probable as any other equally wide range, i.e. naive.

```r
plot_beta_pdf <- function(shape1, shape2, group) {
    p <- seq(0, 100, by = 0.1) / 100 # 0 to 100 percent
    df <- pmap_dfr(
        list(shape1, shape2, group),
        function(shape1, shape2, group) {
            tibble(
                p = p,
                d = dbeta(p, shape1, shape2),
                group = group
            )
        }
    )
    labels_df <- tibble(
        group = group,
        p = shape1 / (shape1 + shape2),
        d = dbeta(p, shape1, shape2) / 2,
        label = glue("Beta({shape1}, {shape2})")
    )
    ggplot(df) +
        aes(x = p, y = d, fill = group) +
        geom_area(alpha = 0.5, position = position_identity()) +
        geom_line(aes(colour = group), alpha = 0.5) +
        geom_text(aes(label = label), data = labels_df) +
        scale_x_continuous(labels = scales::percent) +
        labs(
            subtitle = "Probability density function (PDF)",
            x = "Click-through rate",
            y = "Probability density",
            colour = "Group",
            fill = "Group"
        )
}

experiment_groups <- c("control", "treatment")

plot_beta_pdf(1, 1, group = factor("control", levels = experiment_groups)) +
    labs(
        title = "Prior for click-through rate of control group"
    )
```



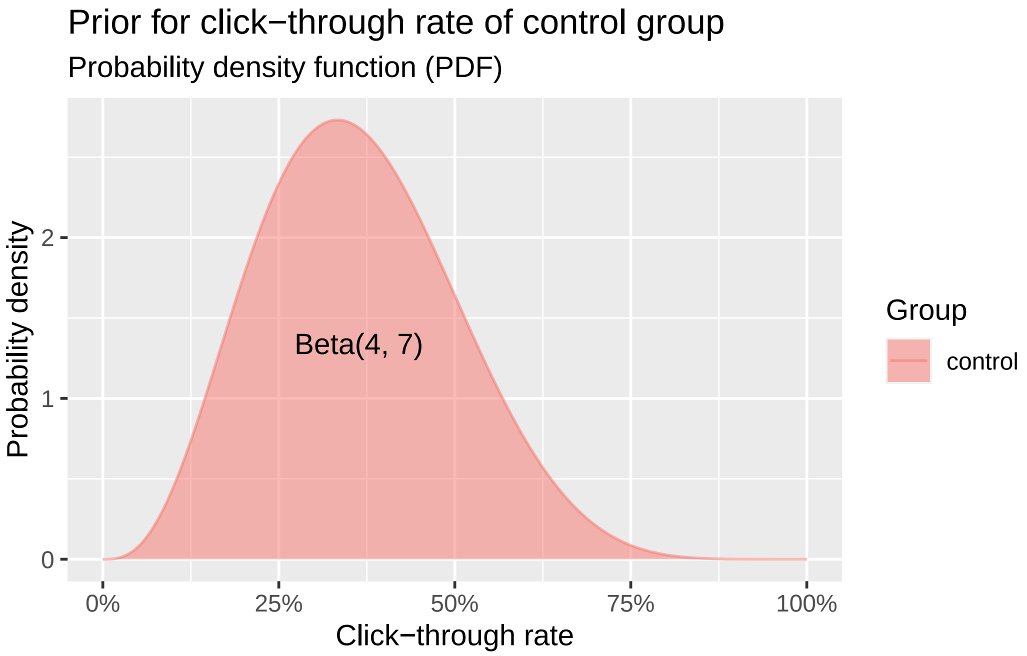
Tip

The **Beta distribution** is a **probability density function (PDF)** with two **shape parameters**: $B(shape1, shape2)$. It's used to describe proportions, such as click-through rate.

The total area under the curve will always add to 100%. That is, the curve represents all possibilities regardless of what shape parameters are used.

2.3 Something a little more informative

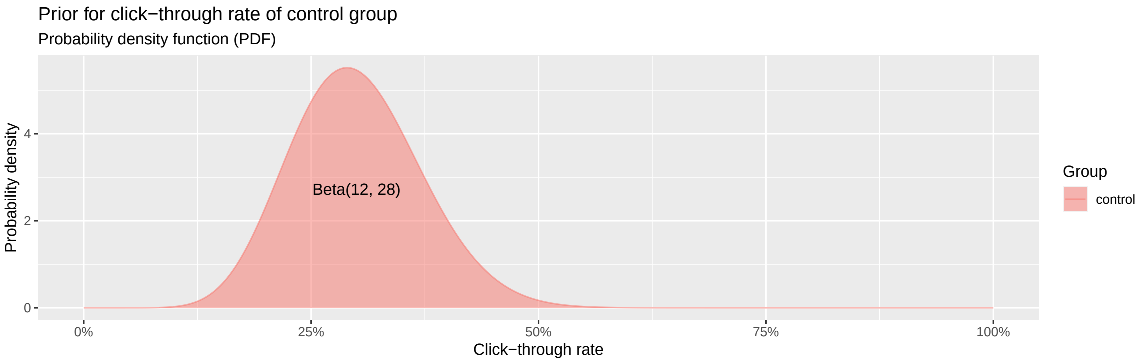
```
plot_beta_pdf(4, 7, group = factor("control", levels = experiment_groups)) +
    labs(
        title = "Prior for click-through rate of control group"
    )
```



As the curve narrows, notice that the shape parameters of the Beta distribution increase.

2.4 Something even more informative

```
plot_beta_pdf(12, 28, group = factor("control", levels = experiment_groups)) +
    labs(
        title = "Prior for click-through rate of control group"
    )
```



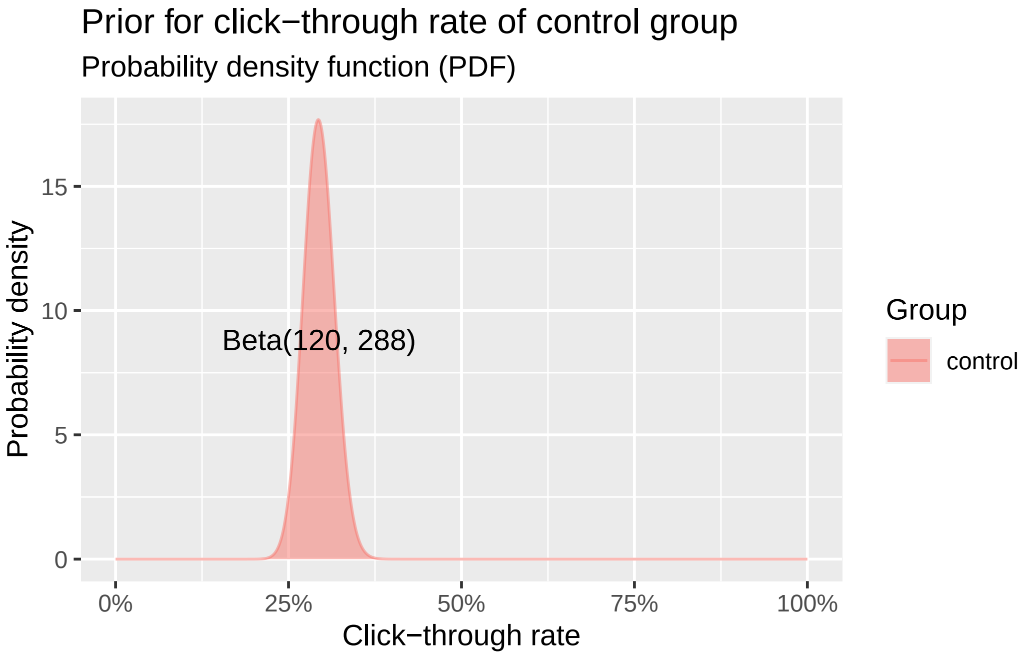
The shape parameters (`shape1` and `shape2`) of the Beta distribution can be considered counts of **successes** and **failures**, respectively. The mean probability of success (i.e. average click-through rate) can be calculated by this formula:

$$\frac{shape1}{shape1 + shape2}$$

The shape parameters are actually slightly more than the count of successes and failures, i.e. $successes = \alpha - 1$ and $failures = \beta - 1$, or $successes = \alpha - 0.5$ and $failures = \beta - 0.5$ if using Jeffreys prior.

2.5 Let's say we've settled on this:

```
plot_beta_pdf(
    shape1 = 120, shape2 = 288,
    group = factor("control", levels = experiment_groups)
) +
    labs(
        title = "Prior for click-through rate of control group"
    )
```



The more confident we are about our beliefs, the narrower the curve.

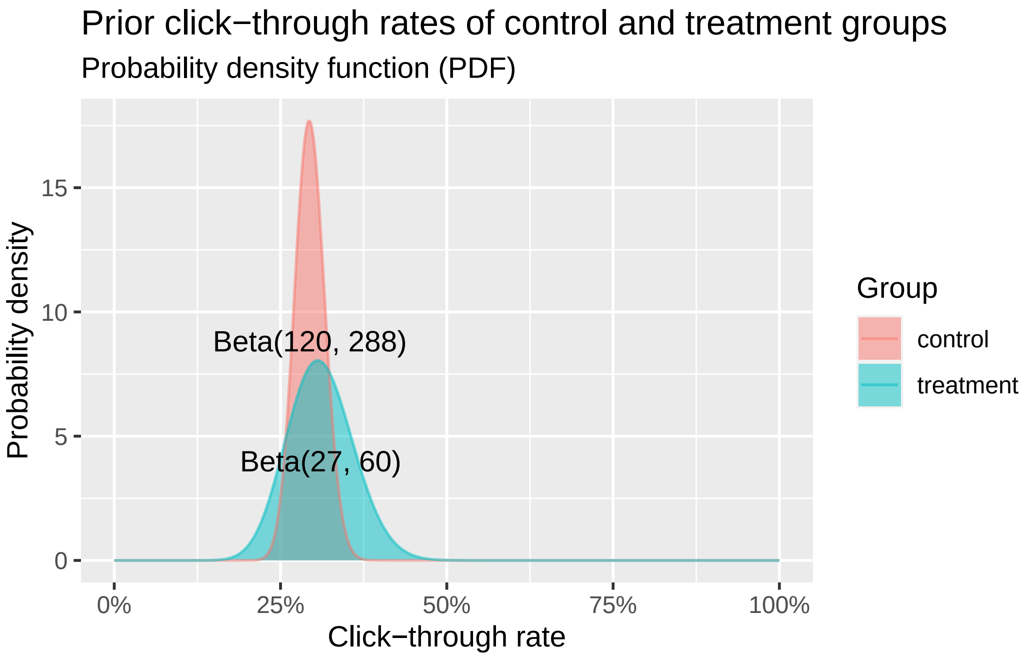
2.6 What about the treatment group?

- We expect that the click-through rates of the treatment and control groups will be correlated.
- We're unsure about how correlated they will be, but we're not expecting a dramatic difference.
- We're more confident than not that the treatment will be an improvement, but we're open to other possibilities.
- We don't want to bias the experiment results in favour of treatment or control, or towards a conclusion of there being a difference or no difference.

2.7 We've settled on these priors:

```
priors <- tibble(
    group = factor(c("control", "treatment"), levels = experiment_groups),
    shape1 = c(120, 27),
    shape2 = c(288, 60)
)

do.call(plot_beta_pdf, priors) +
    labs(
        title = "Prior click-through rates of control and treatment groups"
    )
```



3 Running a Bayesian A/B test

3.1 Prior agreement

- Agreement must be reached on the priors before collecting and analysing data from the experiment.
- Once the priors are agreed to and locked in, we can start the experiment.

Here's a summary of the priors we have chosen:

```
priors %>%
    mutate(
        mean = (shape1 / (shape1 + shape2)) %>%
            scales::percent(),
        sd = sqrt(
            shape1 * shape2 /
                ((shape1 + shape2) ^ 2 * (shape1 + shape2 + 1))
        ) %>%
            scales::percent(suffix = " p.p.")
    )
```

group	shape1	shape2	mean	sd
control	120	288	29.4%	2.3 p.p.
treatment	27	60	31.0%	4.9 p.p.

It's important to not change your original priors after seeing data collected from the experiment. Doing so is effectively double-dipping, whereby your priors are being influenced by the data you have collected.

3.2 Let's run an experiment

```r
true_ctr <- c(control = 32, treatment = 35) / 100
```

We'll generate some fake data to mimic a real experiment.

It'll be rigged though, as we'll already know the click-through rates for control and treatment, which are:

- Control: 32%
- Treatment: 35%

That's a relative uplift of 9.4%.

If we're successful at applying Bayesian inference then we should hope (but can't guarantee due to randomness) that the results somewhat match with these expected CTRs.

3.3 The next day

```r
avg_daily_users <- 150

experiment_simulator <- function(rate_of_users, true_ctr) {
    experiment_groups <- names(true_ctr)
    n_users <- rpois(1, lambda = rate_of_users)
    group_assignment <- sample(
        seq_along(experiment_groups),
        size = n_users, replace = TRUE
    ) %>%
        factor(labels = experiment_groups)
    group_sizes <- table(group_assignment)
    clicks_by_group <- map2(group_sizes, true_ctr, rbernoulli)
    map_dfr(clicks_by_group, function(clicks) {
        list(
            click_data = list(clicks),
            clicked = sum(clicks),
            not_clicked = sum(!clicks)
        )
    }, .id = "group") %>%
        mutate(group = factor(group, levels = experiment_groups))
}
```

Let's pretend that on average 150 users enter our experiment each day, and we've received the following data from day 1:

```
experiment_data <- list(
    batch1 = experiment_simulator(avg_daily_users, true_ctr)
)

experiment_data[['batch1']] %>%
    mutate(
        CTR = scales::percent(clicked / (clicked + not_clicked)),
        total_users = clicked + not_clicked
    ) %>%
    select(group, total_users, clicked, not_clicked, CTR)
```

group	total_users	clicked	not_clicked	CTR
control	61	20	41	33%
treatment	63	29	34	46%

Our experiment simulator randomly selects users and assigns them to each group using a Poisson process. It then randomly chooses which users had clicked using Bernoulli trials (e.g. coin flips).

3.4 Let's now incorporate our priors

Posteriors represent your updated beliefs once you've incorporated experiment data with your priors. Like priors, posteriors represent your beliefs about the metric of interest, which in our case is click-through rate.

For each experiment group, we derive our posterior shape parameters through simple arithmetic addition:

- Increment the first shape parameter by the count users who had **clicked**
- Increment the second shape parameter by the count users who **didn't click**

```
posterior_update <- function(priors, experiment_data) {
    experiment_data %>%
        left_join(priors, by = "group") %>%
        rename(prior_shape1 = shape1, prior_shape2 = shape2) %>%
        select(-click_data) %>%
        mutate(
            posterior_shape1 = prior_shape1 + clicked,
            posterior_shape2 = prior_shape2 + not_clicked
        )
}

posteriors <- posterior_update(priors, experiment_data[['batch1']])

posteriors_long <- posteriors %>%
    gather(key = "count", value = "value", -group, factor_key = TRUE) %>%
    spread(group, value) %>%
    mutate(count = count %>% fct_relevel(
        c(
            "prior_shape1", "clicked", "posterior_shape1",
            "prior_shape2", "not_clicked", "posterior_shape2"
        )
    )) %>%
    arrange(count)
```

```
posteriors_long %>%
    filter(count %in% c("prior_shape1", "clicked", "posterior_shape1"))
```

count	control	treatment
prior_shape1	120	27
clicked	20	29
posterior_shape1	140	56

```
posteriors_long %>%
    filter(count %in% c("prior_shape2", "not_clicked", "posterior_shape2"))
```

count	control	treatment
prior_shape2	288	60
not_clicked	41	34
posterior_shape2	329	94

The process of incorporating data with priors is called Bayesian updating. The data generated follows a Bernoulli distribution (Binomial with 1 trial). The prior follows a Beta distribution, which is conjugate to the Binomial distribution.

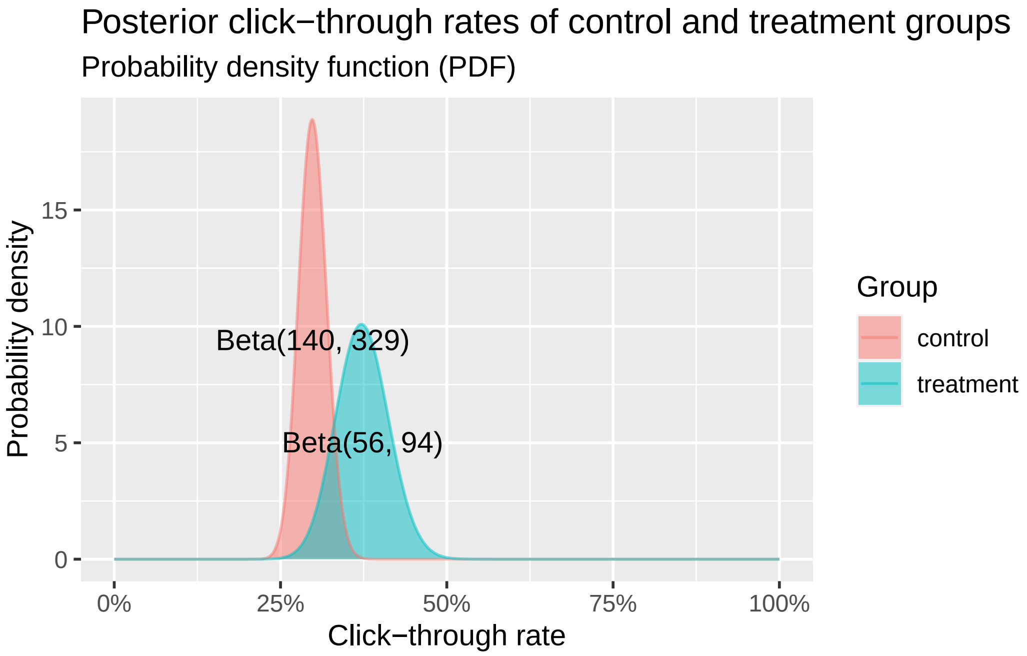
3.5 Posterior distribution of each group

We have now updated our beliefs. These posteriors can now be thought of as our new updated priors.

```
priors_initial <- priors

priors <- posteriors %>%
    select(
        group = group,
        shape1 = posterior_shape1,
        shape2 = posterior_shape2
    )

do.call(plot_beta_pdf, priors) +
    labs(
        title = "Posterior click-through rates of control and treatment groups"
    )
```



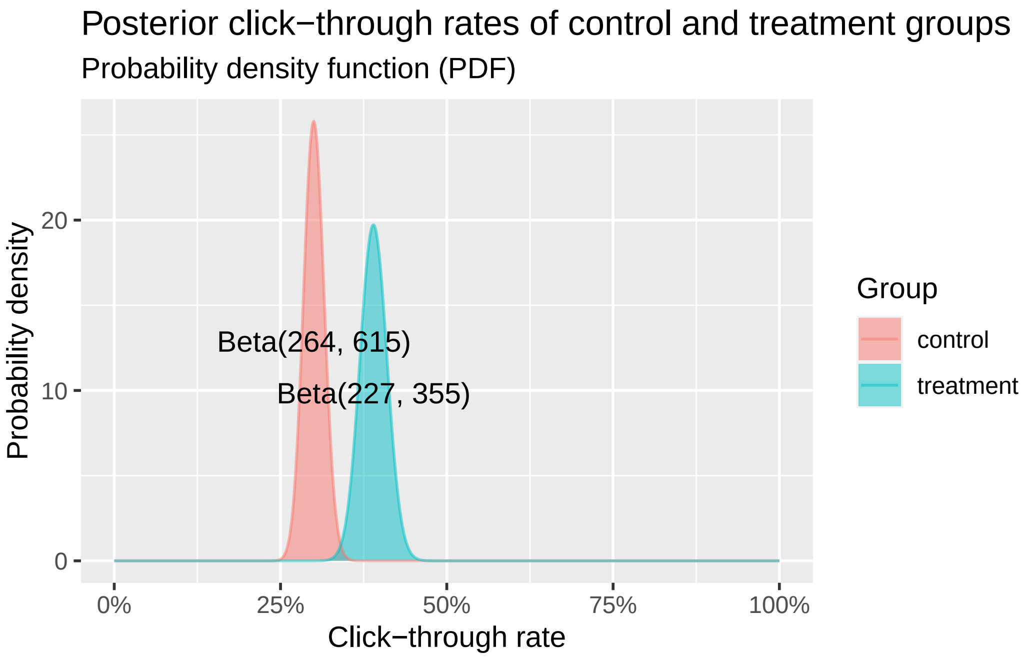
3.6 Another six days later...

We've collected more data, so let's again update our priors to form new posteriors for the click-through rates of each group.

```
experiment_data[['batch2']] <- experiment_simulator(
    avg_daily_users * 6, true_ctr
)
posteriors <- posterior_update(priors, experiment_data[['batch2']])
```

```
priors <- posteriors %>%
    select(
        group = group,
        shape1 = posterior_shape1,
        shape2 = posterior_shape2
    )

do.call(plot_beta_pdf, priors) +
    labs(
        title = "Posterior click-through rates of control and treatment groups"
    )
```



3.7 Another three weeks later...

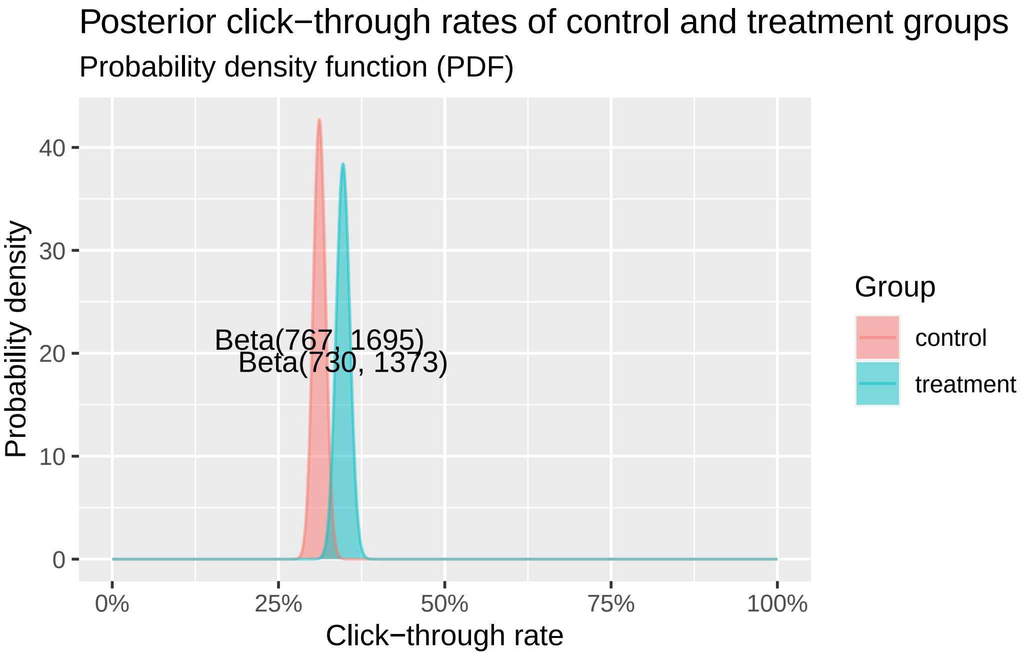
```
experiment_data[['batch3']] <- experiment_simulator(
    avg_daily_users * 7 * 3, true_ctr
)
posteriors <- posteriors %>%
    select(
        group = group,
        shape1 = posterior_shape1,
        shape2 = posterior_shape2
    ) %>%
    posterior_update(experiment_data[['batch3']])

experiment_data_df <- experiment_data %>% map_dfr(~., .id = "batch")
```

We've now observed a total sample of 4,070 users and a decision is made to end the experiment.

```
posteriors <- posteriors %>%
    select(
        group = group,
        shape1 = posterior_shape1,
        shape2 = posterior_shape2
    )

do.call(plot_beta_pdf, posteriors) +
    labs(
        title = "Posterior click-through rates of control and treatment groups"
    )
```



4 Posterior analysis

Statistical inferences using the posterior distributions

4.1 Monte Carlo simulation

Let's draw a very large quantity of random samples from our posterior distributions to make inferences about the experiment.

This is called Monte Carlo simulation – named after a casino.

The more samples drawn, the greater the resolution of the inferences you make, but this comes at the cost of computational time and memory. Nowadays, computer processing speed and memory are more than adequate for what we need. Analytical solutions, providing the greatest level of precision, are also sometimes possible.

```
simulation_size <- 100
```

4.2 100 simulations

Let's start slowly by drawing 100 random samples from our distributions and plot them using histograms...

```r
draw_posterior_samples <- function(posteriors, simulation_size) {
    posteriors %>%
        rowwise() %>%
        mutate(
            sim_id = list(1:simulation_size),
            ctr = list(
                rbeta(
                    n = simulation_size,
                    shape1 = shape1,
                    shape2 = shape2
                )
            )
        ) %>%
        select(group, sim_id, ctr) %>%
        unnest(cols = c(sim_id, ctr))
}

posterior_samples <- draw_posterior_samples(posteriors, simulation_size)
```

Here’s some of our Monte Carlo samples:

```
posterior_samples %>%
    spread(group, ctr) %>%
    mutate(
        uplift = treatment / control - 1,
        beats_control = treatment > control
    ) %>%
    select(-sim_id) %>%
    head(n = 7) %>%
    knitr::kable(digits = 3)
```

control	treatment	uplift	beats_control
0.326	0.334	0.024	TRUE
0.301	0.367	0.216	TRUE
0.294	0.357	0.212	TRUE
0.315	0.340	0.077	TRUE
0.322	0.332	0.031	TRUE
0.294	0.348	0.184	TRUE
0.316	0.346	0.094	TRUE

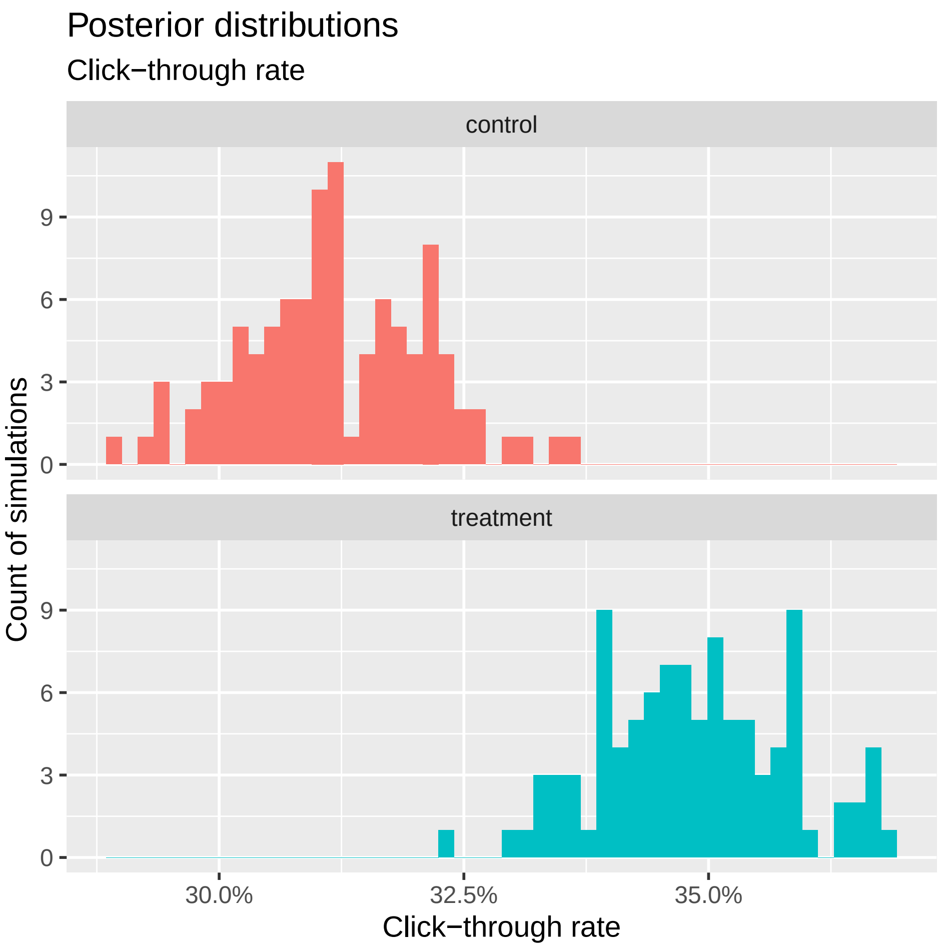
```
plot_posterior_samples <- function(
        posterior_samples, bins = 100, animate = FALSE
) {
    posterior_samples <- posterior_samples %>%
        mutate(
            frame = cut(log(sim_id), breaks = 50, labels = FALSE)
        )
    p <- ggplot(posterior_samples) +
        facet_wrap(~group, ncol = 1) +
        aes(ctr, fill = group) +
        geom_histogram(bins = bins) +
        scale_x_continuous(labels = scales::percent) +
        labs(
            title = "Posterior distributions",
            subtitle = "Click-through rate",
            x = "Click-through rate",
            fill = "Group",
            y = "Count of simulations"
        ) +
        guides(fill = guide_none())

    if(animate) {
        p <- p +
            transition_manual(frame, cumulative = TRUE) +
            view_follow()
        p <- p %>%
            animate(duration = 10, renderer = gifski_renderer(loop = FALSE))
    }

    p

}

posterior_samples %>%
    plot_posterior_samples(animate = FALSE, bins = 50)
```



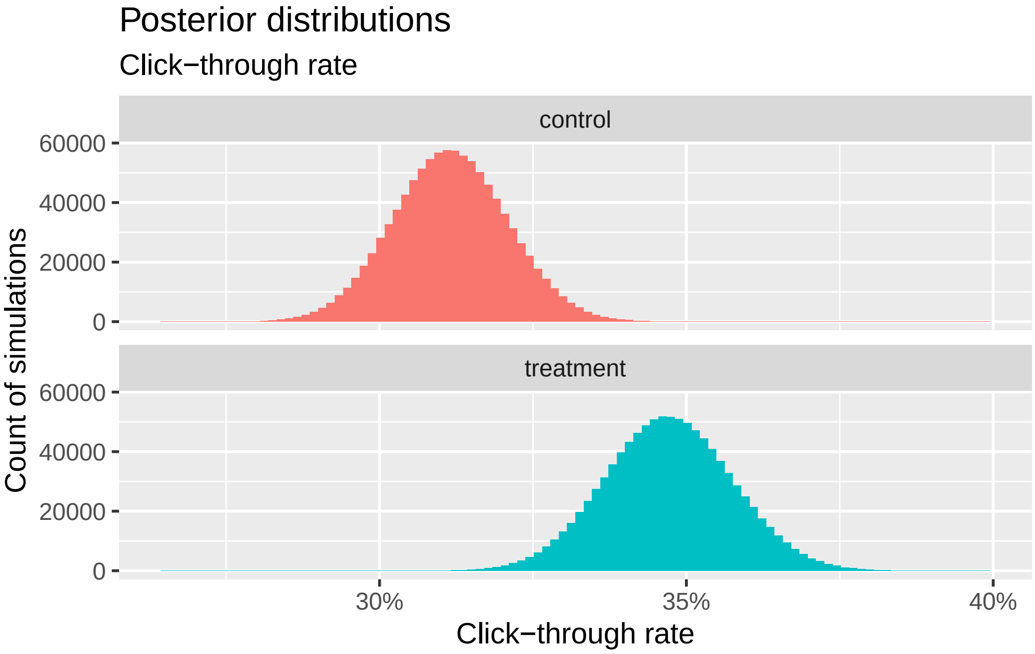
4.3 Let's now beef it up a bit...

```
simulation_size <- 1000000
```

We'll now draw 1,000,000 samples...

```
posterior_samples <- draw_posterior_samples(posteriors, simulation_size)

posterior_samples %>%
    plot_posterior_samples(animate = render_animations)
```



Notice how these histograms follow the same distribution as our posteriors. That is because these samples have been drawn at random according to those posterior distributions.

4.4 We can now make some inferences

A summary of our 1,000,000 posterior samples for click-through rate:

```
posterior_comparison <- posterior_samples %>%
    spread(group, ctr) %>%
    mutate(
        uplift = treatment / control - 1,
        beats_control = treatment > control
    )

posterior_comparison %>%
    select(-sim_id) %>%
    summary()
```

```
   control          treatment          uplift          beats_control 
 Min.   :0.2648   Min.   :0.3009   Min.   :-0.08501   Mode :logical  
 1st Qu.:0.3052   1st Qu.:0.3401   1st Qu.: 0.08278   FALSE:5365     
 Median :0.3115   Median :0.3471   Median : 0.11421   TRUE :994635   
 Mean   :0.3115   Mean   :0.3471   Mean   : 0.11525                  
 3rd Qu.:0.3178   3rd Qu.:0.3541   3rd Qu.: 0.14658                  
 Max.   :0.3570   Max.   :0.3986   Max.   : 0.35880                  
```

- How do these compare to our theoretical CTRs of 32% for control and 35% for treatment, and uplift of 9.4%?

```
posterior_uplift <- with(
    posterior_comparison,
    mean(beats_control)
)
```

- What is the posterior probability that the CTR of the treatment is greater than that of control? Answer: 99.46%

Out of our 1 000 000 simulations, we can see how often the treatment bet control. This tells us the probability that treatment is the winner.

If we filtered our simulations to those where control won and calculated the median CTR uplift, and did the same for cases where treatment won, we can determine the expected losses of choosing either variant as the winner. We should prefer the variant with the lowest expected loss or continue to run the experiment longer to improve our confidence.

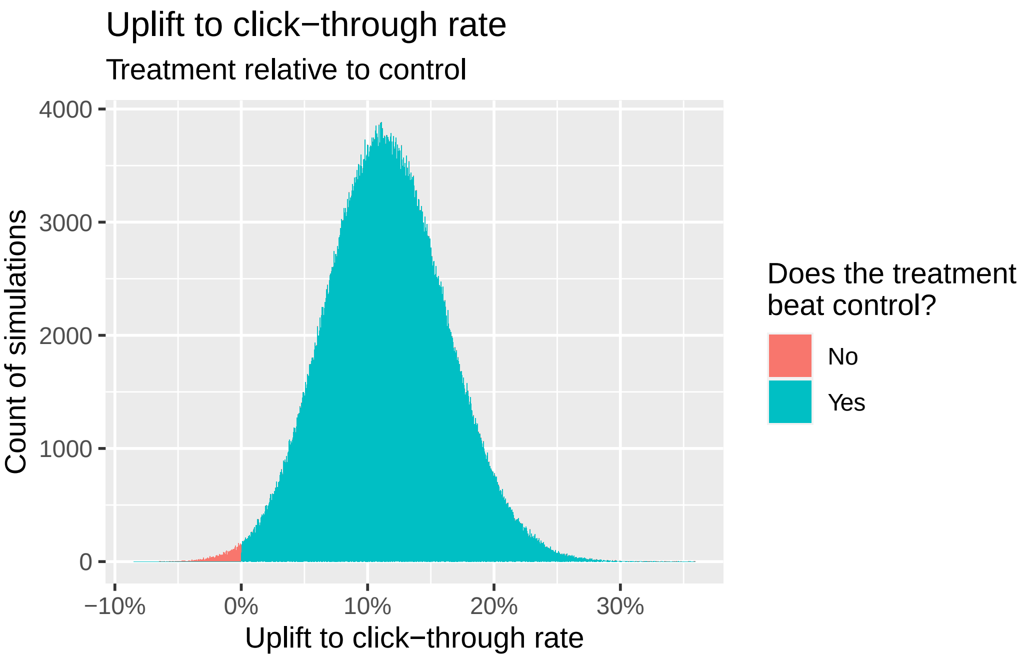
4.5 Posterior distribution of the CTR uplift

```
p <- ggplot(
    posterior_comparison %>%
        mutate(
            frame = cut(log(sim_id), breaks = 10, labels = FALSE)
        )
) +
    aes(
        uplift,
        fill = factor(
            if_else(beats_control, "Yes", "No"),
            levels = c("No", "Yes")
        )
    ) +
    geom_histogram(bins = 1000) +
    scale_x_continuous(labels = scales::percent) +
    scale_fill_discrete(drop = FALSE) +
    labs(
        title = "Uplift to click-through rate",
        subtitle = "Treatment relative to control",
        x = "Uplift to click-through rate",
        y = "Count of simulations",
        fill = "Does the treatment\nbeat control?"
    )

if (render_animations & FALSE) {
    p <- p +
        transition_manual(frame, cumulative = TRUE) +
        view_follow()

    p <- p %>%
        animate(duration = 5, renderer = gifski_renderer(loop = FALSE))
}

p
```



5 Summary and some final remarks

5.1 Before starting an experiment

Gather prior knowledge and articulate beliefs:

- Establish a baseline - what do you know about the control group?
- What do you expect the effect of the treatment to be? How sure are you?

Express those beliefs and knowledge as distributions - these are your priors for your control and treatment groups.

Ensure that the priors encapsulate the collective knowledge and beliefs of all interested parties so that there is agreement. This is to avoid the results from being challenged later.

5.2 Running the experiment

- Start the experiment, gather data, and update your priors to form posteriors about the metric of interest
- Draw inferences by running a large number of Monte Carlo simulations using the posterior distributions
- Know when to end the experiment – try to plan for this ahead of running the experiment

Null-hypothesis significance testing (NHST) is not what Bayesian is for:

- Bayesian tells you the probability of some effect being within some range, given the data. I.e. Given everything we know so far, what are the risks associated with the choices we have?
- NHST tells you the probability of data at least as extreme as what has been observed, given there is no real effect. I.e. How ridiculous would this outcome be if it were due to chance alone?

NHST is often referred to as the frequentist approach, where decisions are made using p-values and some arbitrary threshold α (i.e. false positive rate).

Unlike NHST, Bayesian A/B testing doesn't give you a yes/no answer – it instead informs you about the probabilities and risks associated with the choices you have.

6 Thank you!

Further topics that might interest you:

- **Bayesian Generalised Linear Models** to better isolate the effect of the treatment from other predictors.
- **Survival Analysis**, such as **Kaplan Meier**, to analyse lagged conversion outcomes.

> These slides and simulations were produced in RStudio using Quarto. Download the source code and slides at: https://github.com/jdeboer/measurecamp2022

7 When to stop a Bayesian A/B test?

7.1 If using *uninformative* priors…

If your original priors are uninformative or too weak, then you face the same risks as with frequentist experiments.

Perform **power analysis** ahead of running the experiment. This is to determine the required sample size before any inferences are made.

Before commencing the experiment, decide on:

- The **minimum detectable effect** size
- The accepted **false positive rate**
- The accepted **false negative rate**

7.2 If using *informative* priors...

If your priors are relatively informative and chosen carefully, then this can reduce the chances of false positives and negatives. But:

- Be careful not to bias the results of the experiment.
- Power analysis is still recommended in order to gauge the worse case scenario for how long the experiment might run.

Bayesian inference, with informative priors, can make it **possible to end an experiment early**.

7.3 If deciding to end early...

Ask yourself:

- Has the experiment run for at least a couple of cycles? (e.g. at least two full weeks)
- Have the results stabilised? Is there a clear winner?
- Could it be worth running longer to learn more?
- What are the **risks** of continuing or ending now? What if the results you see are just a fluke and are therefore misguiding you? What is the impact of making the wrong choice? What are the chances?

8 Extras

8.1 Bayes theorem

$$P(B \cap A) = P(A \cap B)$$

$$P(B) \times P(A|B) = P(A) \times P(B|A)$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$f(\theta|data) = \frac{f(data|\theta)f(\theta)}{f(data)}$$

$$Posterior \propto \mathcal{L}(\theta|data) \times prior$$

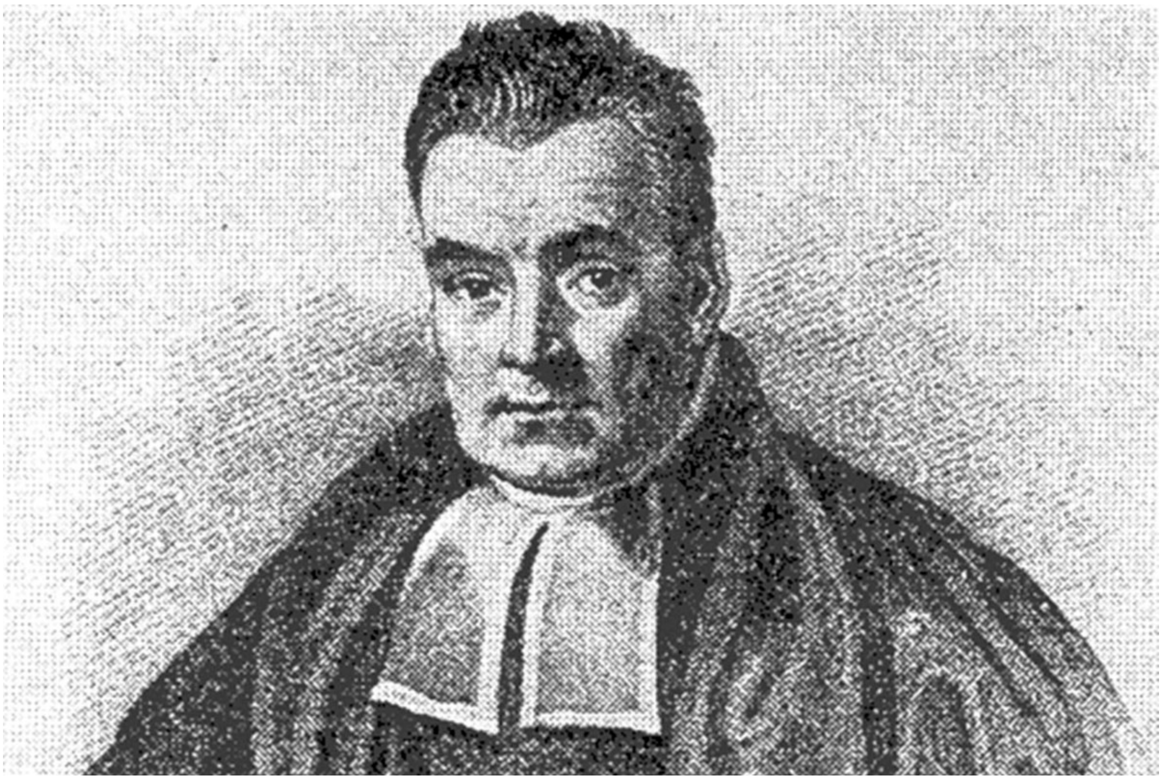


Figure 1: Thomas Bayes - 1701 – 1761

8.2 Some useful formulas

Let α and β represent the first and second shape parameters of the Beta distribution, respectively.

The mean of this distribution is: $\mu = \frac{\alpha}{\alpha+\beta}$

The standard deviation is: $\sigma = \sqrt{\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}}$

Through substitution and rearrangement, you can determine α and β from μ and σ.

$$v = \frac{\mu(1-\mu)}{\sigma^2} - 1$$

$$\alpha = \mu v$$

$$\beta = (1 - \mu)v$$

This way, you can determine the shape parameters based on centrality and spread.

9 Game of chances

9.1 To play this game:

- There is one host and at least 2 contestants.
- The host will need a uniform random number generator.

9.2 Instructions

1. The host secretly picks a number, Y, between 0% and 100% and writes it down.
2. The host will then secretly generate a random number, X, again between 0% and 100%:
   - If X is less than Y then the host will mark it as a 'success', otherwise as a 'failure', using a tally board that's visible to all contestants.
3. The objective of the game is for contestants to estimate Y by asking the host to perform step 2 as many times as they need (to a reasonable limit). The closer their guess is, the better. However, each contestant can only call out their guess once.
4. Once two contestants have called out what they believe Y is, then the game ends and the host reveals the true answer for Y.
   - The first contestant to call out their guess wins the game if they are within 5 points of Y. If not, then the contestant whose guess is closest to Y wins.

9.3 Lessons of the game

- As the number of successes and failures increases, you get closer to knowing what Y is.
- The compromise each contestant makes between speed and certainty will influence who wins.

> Tip
>
> Try a modification to the game where Y is a number related to a topic that the audience will have some prior knowledge about. This way they can incorporate their prior expectations when making a guess.