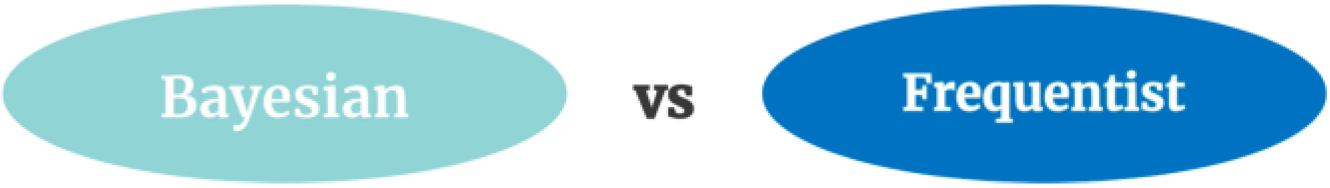
Source: https://towardsdatascience.com/bayesian-a-b-testing-and-its-benefits-a7bbe5cb5103

What is Bayesian A/B testing and when should you use it?



The A/B testing dilemma. Image by author.

Recently, Bayesian A/B testing has gotten lots of publicity because its methods are easy to understand and allow useful calculations, such as the probability that a treatment is better than the control. Bayesian inference also performs much better on small sample sizes; according to a 2019 medium post, Bayesian A/B testing can reduce required sample size by 75%.

While these methods are more computationally expensive than traditional frequentist approaches, they are computed offline, which reduces performance requirements. The main challenge is choosing effective distributions to support inference.

Anyone with an experimentation pipeline and access to a computer can leverage Bayesian A/B testing techniques. Here's how…

Steps of Bayesian A/B Testing

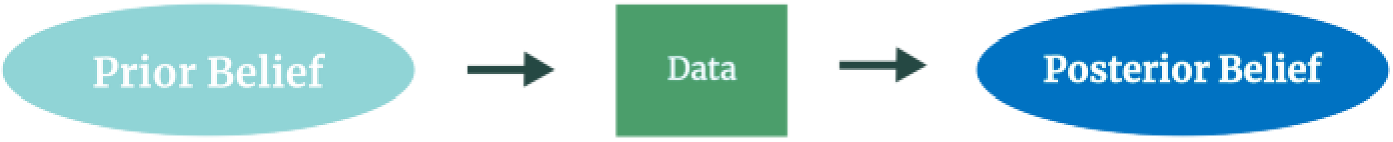
1. **Select your distribution based on your metric of interest.** Here, we discuss the binomial, multinomial, and exponential distributions. They cover most business use cases.
2. **Calculate your prior.** Based on the distribution selected above, we next select a conjugate prior and choose distribution parameters that best reflect our pre-experiment data. Distribution parameters can be chosen manually or using a library.
3. **Run the experiment.**
4. **Calculate three key metrics using Monte Carlo simulations.** These metrics are percent lift, probability of being best, and expected loss.

But, what’s actually going on?

Ok, let’s slow down and understand what’s actually going on.

Bayesian Statistics

Starting at square 1, let’s talk about what Bayesian inference is. In one sentence, Bayesian inference leverages conditional probability to help us understand how data impacts our beliefs.



The Bayesian Approach. Image by author.

Let’s say we start with a prior belief that the sky is red. After looking at some data, we would soon realize that this prior belief is wrong. So, we perform *Bayesian updating* to improve our incorrect model about the color of the sky, ending up with a more accurate posterior belief.

Likelihood Distributions and Conjugate Priors

One key component of our belief is the structure of our data. We often describe this structure through distributions. For instance, in our sky example, assuming that there are categories of color (red, white, blue, etc.), our distribution would be multinomial. If we were looking at a set of numbers, our distribution might be normal. If the data were true/false values, it would be binomial, and so on.

These distributions are called likelihood distributions because they show the likelihood that our data will take on a certain value.

For our sky example, we’re working with a multinomial distribution, but there’s one more distribution we need to think about. When performing Bayesian updates, we must consider the distribution that’s the *conjugate prior* of our likelihood distribution. Conjugates are distributions that come from the same family. For our case, the conjugate prior of a multinomial distribution is the Dirichlet distribution.

Conjugate priors are the source of our data's likelihood distribution. For example, if we're flipping a coin, the binomial distribution shows what *n* number of coinflips would look like with probability *p* of being heads. However, often *p* itself has a distribution. The distribution of *p* is the conjugate prior distribution.

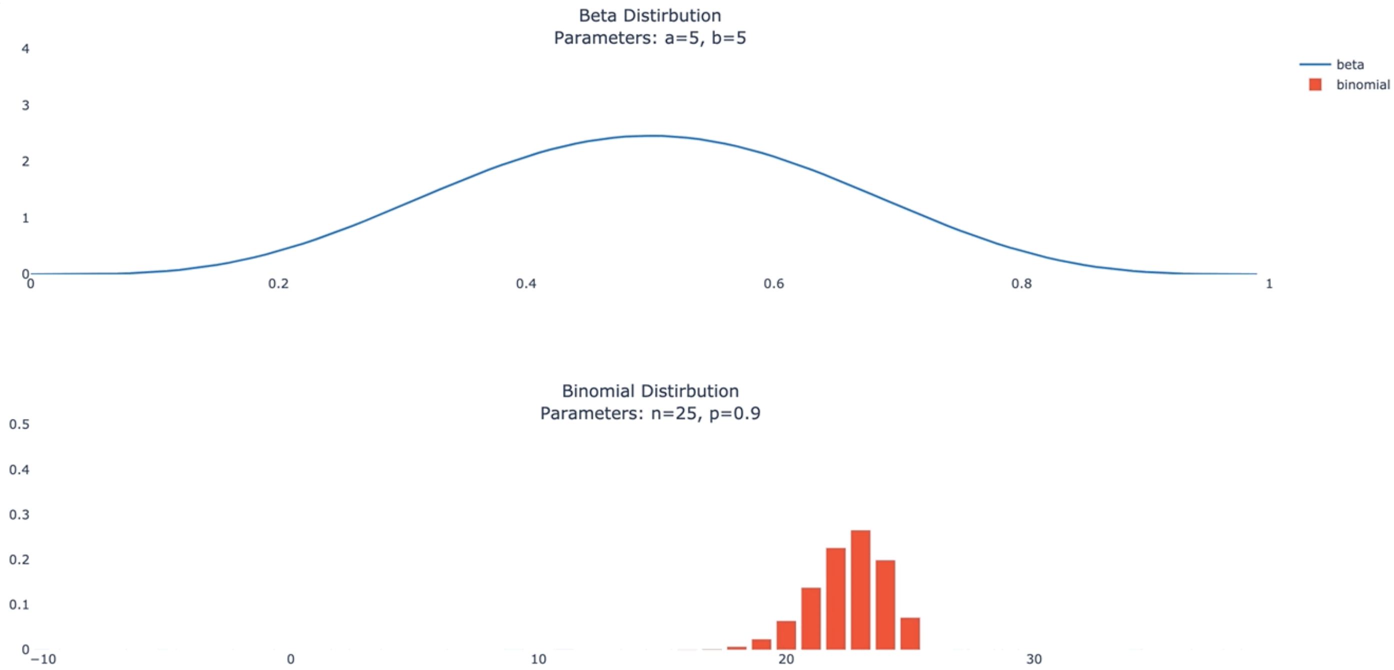
Common Distributions in Business

Typically, there are three main types of data that we could observe in a business setting (although other distributions are useful):

1. **Binary**: data where the user has one option to choose between.
2. **Categorical**: data where the user has a set of options to choose from.
3. **Continuous**: data where the user has a set of options to choose from, but we only observe those choices in aggregate.

For binary data, let's take a concrete example of conversion for an online ML textbook store. *Conversion=1* indicates a user purchased a book and *conversion=0* indicates they did not.

In this scenario, the binomial distribution describes our data. The conjugate prior of the binomial is the beta distribution, both of which can be seen below.



Beta (top) and binomial (bottom) with different hyperparameter values. Image by author.

Moving on to our second type of data, categorical data’s likelihood distribution is multinomial and its conjugate prior is Dirichlet. Because these plots are often high dimensional, we unfortunately aren’t going to show a visualization.

Finally, for continuous data, we use the exponential distribution with a gamma conjugate prior. If you’re curious what they look like, check out those links.

Pretty simple, right? Depending on the data type, just apply those distributions and you’re good to go.

Bayesian Statistics Calculations

Now that we understand how to select a distribution for our experiment, let’s learn how to determine experimental impact. We do this by calculating three statistics:

1. Treatment lift
2. Probability of being the best
3. Expected loss

For our first calculation, treatment lift is simply the percent change between our treatment and control. This is our treatment impact and can be calculated with the formula below. And, for simplicity, let’s assume there’s just one treatment and one control.

```
treatment_lift = (treatment - control) / control
```

For our second and third calculations, we need to develop a Monte Carlo simulation. Why you might ask? Well, because we only have one sample of data, we can’t compute any probabilities — we don’t know how the data would look in alternative samples. So, to generate more data, we leverage knowledge about our data generating mechanism (the posterior distribution) and samples from that. 10,000 samples is a good rule of thumb.

For the *probability of being best*, our second statistic, we simply look at all of our simulated samples and calculate the percentage of the time that our treatment is better than our control. That proportion becomes our *probability of being best* for the treatment. Check out the pythonic pseudo code below.

```
# probability best of treatment
samp_treatment = sample_from_distribution(treatment_dist,
n=10000)
samp_control = sample_from_distribution(control_dist,
n=10000)probability_best = mean(int(samp_treatment >
samp_control))
```

Finally, for our third statistic, we look to calculate the *expected loss* i.e. the price we pay for implementing an incorrect treatment. To calculate expected loss, we iterate over our samples and calculate *max(treat - control, 0)*. Then, we take the mean of those zero-bounded values to determine our *expected loss*.

```
# pythonic pseudo code - expected lossloss =
mean(max(samp_control - samp_treatment, 0))
```

Expected loss and *probability of being best* are two of the main selling points for Bayesian experimentation. Frequentist methods don’t provide either value.

For details on these calculations, check out this post.

And there you have it!

Implementation Notes

- If you choose an effective prior, Bayesian A/B testing requires a smaller sample size so you can get results faster.
- Bayesian methods are more computationally intensive than frequentist methods.
- When choosing a prior, err on the side of a weak prior i.e. smaller hyperparameter values.