LDA Tutorial

Chris Tufts

Contents

Preface

Chapter 1

What is LDA?

1.1 Animal Generator

1.2 Inference

Inference

Chapter 2

Parameter Estimation

LDA is a generative probabilistic model, so to understand exactly how this works we need to understand the underlying probability distibutions. In this chapter we will focus on the Bernoulli distribution and the Beta distribution. Both of these distributions are very closely related to (and also special cases of) the multinomial and Dirichlet distributions utilized by LDA, but they are a bit easier to comprehend. Once we have made our way through Bernoulli and beta, the following chapter will go into detail about multinomial and Dirichlet distributions and how all these peices are connected.

Throughout the chapter I'm going to build off of a simple example - a single coin flip. Let's begin.

2.1 Distributions

2.1.1 Bernoulli

When you flip a coin you get either heads or tails as an outcome (barring the possibility it lands on it's side). This single coin flip is an example of a Bernoulli trial and we can use the Bernoulli distribution to calculate the probability of either outcome. Any single trial with two possible outcomes can be modeled as a Bernoulli trial: team wins/loses, pitch is a strike/ball, coin comes up heads or tails, etc.

2.1.1.1 Bernoulli: A Special Case of the Binomial Distribution

You will often see Bernoulli distribution mentioned as a special case of the Binomial distribution. The binomial model consists of n Bernoulli trials, where each trial is independent and the probability of success does not change between trials.(Kerns 2010). The Bernoulli distribution is the case of a single trial or n=1.

NOTE: I will use the term success interchangably with the term heads when describing bernoulli distribution. In reality success could be tails if you choose to define it that way.

To clarify, if I want to calculate the probability of getting heads on a single coin flip I will use a bernoulli distribution. However, if I want to know the probability of getting 2 heads (or more) in a row, this is where the binomial distribution comes in. For our purposes we are only concerned about the outcome of a single coin flip and will therefore stick to the Bernoulli distribution.

2.1.1.2 Bernoulli - Distribution Notation

The probability mass function of the bernoulli distribution is shown in Equation 1.

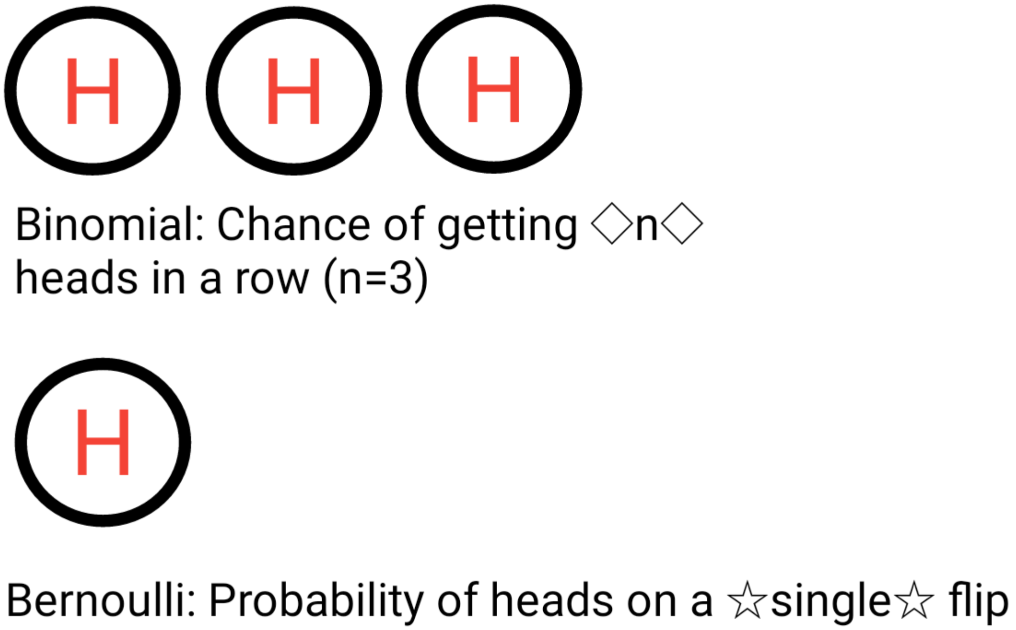
H
H
H
Binomial: Chance of getting ◇n◇ heads in a row (n=3)
H
Bernoulli: Probability of heads on a ☆single☆ flip

Figure 2.1: Bernoulli and Binomial Distributions

$$f_x(x) = P(X = x) = \theta^x(1-\theta)^{1-x}, \qquad x = \{0,1\} \tag{1}$$

The only parameter of the bernoulli distribution is θ, which defines the probability of success during a bernoulli trial. The value of x is 0 for a failure and 1 for a success. In a practical example you can think of this as 0 for tails and 1 for heads during a coin flip. In Equation 2 the value of θ is set to 0.7. We can see the probability of getting a success is 0.7, while the probability of failure is 0.3.

$$\begin{aligned}
&P(X = 1) = \theta^1(1-\theta)^{1-1}, \qquad \theta = 0.7\\
&P(X = 1) = 0.7 * 1 = 0.7\\
\\
&P(X = 0) = 0.7^0(1-0.7)^{1-0}\\
&P(X = 0) = 0.3
\end{aligned} \tag{2}$$

2.1.2 Beta Distribution

The beta distribution can be thought of as a probability distribution of distributions(Robinson 2014).

We know the bernoulli distribution has one parameter, θ. We can use the beta distribution to determine the probability of a specific value of θ based on prior information, in our case previous coin flips.

If you flip a coin 2 times resulting in 1 heads and 1 tails how sure are you that the coin is fair? Probably not all that sure, right? But what if you flipped the coin 200 times and it resulted in 100 heads and 100 tails? You would be much more confident that the coin is fair. This is the basis of the beta distribution.

The beta distribution has 2 shape parameters, α and β. These can be though of as the results from the coin flips we just talked about. Below the probability density for different values of θ is displayed based on different values of α and β. In general, the higher the value of α and β the narrower the density curve is. This makes sense with our thought example above, the more information (*coin flip results*) we have, the more confident we are in our coin's bias (i.e. is it fair, head heavy, etc.).

```r
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Warning: package 'tidyr' was built under R version 3.4.2
```

```
## Warning: package 'purrr' was built under R version 3.4.2
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
## Conflicts with tidy packages ------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```r
a <- c(1, 10, 100)
b <- c(1, 10, 100)
params <- cbind(a,b)
ds <- NULL
n <- seq(0,1,0.01)
for(i in 1:nrow(params)){
  ds <- rbind(data.frame(x = n, y = dbeta(n, params[i,1], params[i,2]),
                             parameters = paste0("\U03B1 = ",params[i,1],
                                                 ", \U03B2 = ", params[i,2])), ds)
}

ggplot(ds, aes(x = x, y = y, color=parameters)) + geom_line() +
  labs(x = '\U03B8', y = 'Probability Density') +
  scale_color_discrete(name=NULL) + theme_minimal()
```

```
## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b8

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b8

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b1

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b2

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b1

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b2

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on '  = 100,   = 100' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on '  = 100,   = 100' in 'mbcsToSbcs': dot substituted for <b1>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on '  = 100,   = 100' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on '  = 100,   = 100' in 'mbcsToSbcs': dot substituted for <b2>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b1

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b2

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b1

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b2

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on '  = 10,   = 10' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on '  = 10,   = 10' in 'mbcsToSbcs': dot substituted for <b1>
## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' = 10,   = 10' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' = 10,   = 10' in 'mbcsToSbcs': dot substituted for <b2>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b1

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b2

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b1

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b2

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' = 1,   = 1' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' = 1,   = 1' in 'mbcsToSbcs': dot substituted for <b1>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' = 1,   = 1' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' = 1,   = 1' in 'mbcsToSbcs': dot substituted for <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <b2>
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 10,  = 10' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 10,  = 10' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 10,  = 10' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 10,  = 10' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 10,  = 10' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 10,  = 10' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 10,  = 10' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 10,  = 10' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 1,  = 1' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 1,  = 1' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 1,  = 1' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 1,  = 1' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 1,  = 1' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 1,  = 1' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 1,  = 1' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
```

```
## conversion failure on '  = 1,   = 1' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 100,   = 100' in 'mbcsToSbcs': dot substituted
## for <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 10,   = 10' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 10,   = 10' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 10,   = 10' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 10,   = 10' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 10,   = 10' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 10,   = 10' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 10,   = 10' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 10,   = 10' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 1,   = 1' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 1,   = 1' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 1,   = 1' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 1,   = 1' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 1,   = 1' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 1,   = 1' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 1,   = 1' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 1,   = 1' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on ' = 100,   = 100' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on ' = 100,   = 100' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on ' = 100,   = 100' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on ' = 100,   = 100' in 'mbcsToSbcs': dot
## substituted for <b2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x
## $x, x$y, : conversion failure on ' = 10,   = 10' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x
## $x, x$y, : conversion failure on ' = 10,   = 10' in 'mbcsToSbcs': dot
## substituted for <b1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x
## $x, x$y, : conversion failure on ' = 10,   = 10' in 'mbcsToSbcs': dot
## substituted for <ce>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x
## $x, x$y, : conversion failure on ' = 10,   = 10' in 'mbcsToSbcs': dot
## substituted for <b2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on ' = 1,   = 1' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on ' = 1,   = 1' in 'mbcsToSbcs': dot substituted
```

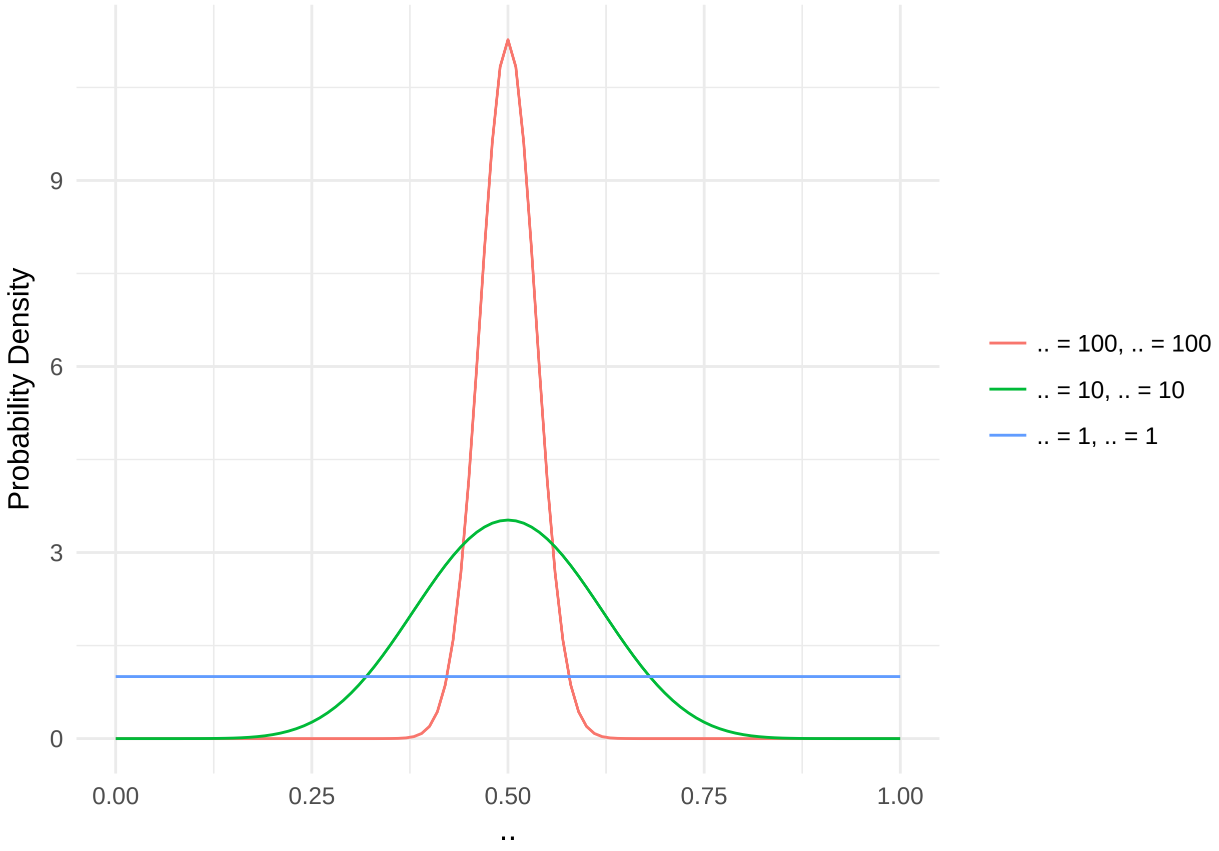

Figure 2.2: Beta Distribution

```
## for <b1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '  = 1,   = 1' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '  = 1,   = 1' in 'mbcsToSbcs': dot substituted
## for <b2>
```

What about the cases where α and β are not equal or close to equal? Well in those cases you would probably assume a bit of skew in the distribution, i.e. your coin may be biased toward head or tails as shown below.

```
a <- c(8, 2)
b <- c(2, 8)
params <- cbind(a,b)
ds <- NULL
n <- seq(0,1,0.01)
for(i in 1:nrow(params)){
  ds <- rbind(data.frame(x = n, y = dbeta(n, params[i,1], params[i,2]),
                             parameters = paste0("\U03B1 = ",params[i,1],
                                                 ", \U03B2 = ", params[i,2])), ds)
}

ggplot(ds, aes(x = x, y = y, color=parameters)) + geom_line() +
  labs(x = '\U03B8', y = 'Probability Density') +
  scale_color_manual(name=NULL, values = c("#7A99AC", "#E4002B")) + theme_minimal()
```

```
## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b8

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b8

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b1

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b2

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b1

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b2

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' = 2,   = 8' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' = 2,   = 8' in 'mbcsToSbcs': dot substituted for <b1>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' = 2,   = 8' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' = 2,   = 8' in 'mbcsToSbcs': dot substituted for <b2>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b1

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b2

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b1

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for Unicode character U+03b2

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' = 8,   = 2' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' = 8,   = 2' in 'mbcsToSbcs': dot substituted for <b1>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' = 8,   = 2' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on ' = 8,   = 2' in 'mbcsToSbcs': dot substituted for <b2>
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
```

```
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 2,  = 8' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <b1>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' = 8,  = 2' in 'mbcsToSbcs': dot substituted for
## <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '  = 8,   = 2' in 'mbcsToSbcs': dot substituted for
## <b2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <ce>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on ' ' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '  = 2,   = 8' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '  = 2,   = 8' in 'mbcsToSbcs': dot substituted
## for <b1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '  = 2,   = 8' in 'mbcsToSbcs': dot substituted
```

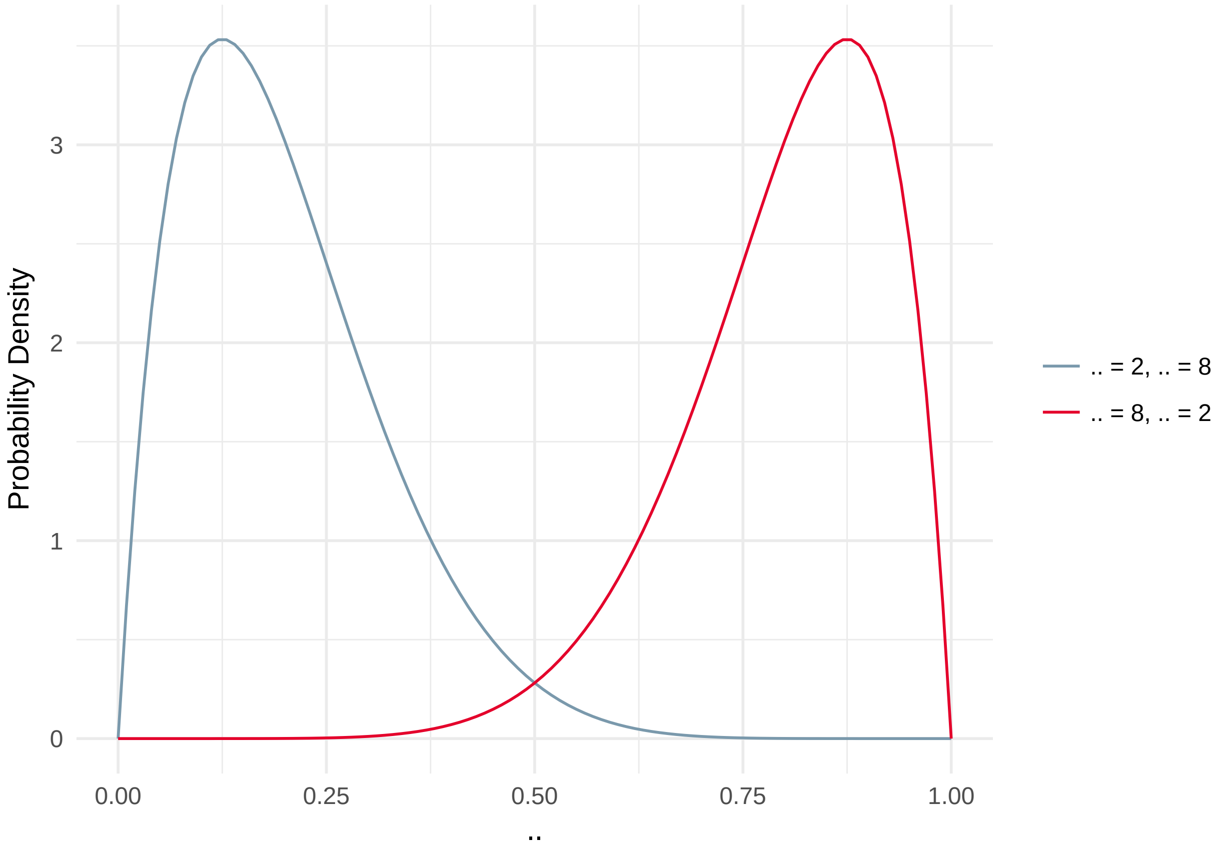

Figure 2.3: Beta Distribution - Skewed

```
## for <ce>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '  = 2,   = 8' in 'mbcsToSbcs': dot substituted
## for <b2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '  = 8,   = 2' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '  = 8,   = 2' in 'mbcsToSbcs': dot substituted
## for <b1>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '  = 8,   = 2' in 'mbcsToSbcs': dot substituted
## for <ce>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '  = 8,   = 2' in 'mbcsToSbcs': dot substituted
## for <b2>
```

The probability distribution function for the beta distribution can be found below.

$$f(\theta;\alpha,\beta) = \frac{\theta^{(\alpha-1)}(1-\theta)^{(\beta-1)}}{B(\alpha,\beta)}$$

Quick Note:

- The *Beta* function, B is the ratio of the product of the *Gamma* function, Γ, of each parameter divided by the *Gamma* function of the sum of the parameters. The *Beta* function is **not** the same as the beta distribution. The *Beta* function is shown below along with the *Gamma* function, which is used in the *Beta* function.

$$\beta(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$$

- The *Gamma* function is the factorial of the parameter minus 1.

$$\Gamma(a) = (a-1)!$$

2.2 Inference: The Building Blocks

The equation below is a fundamental to understanding parameter estimation and inference.

$$\underbrace{p(\theta|D)}_{posterior} = \frac{\overbrace{p(D|\theta)}^{likelihood}\ \overbrace{p(\theta)}^{prior}}{\underbrace{p(D)}_{evidence}} \tag{3}$$

The 4 components are:

- **Prior**: The probability of the parameter(s). Defines our prior beliefs of the parameter. Do we *believe* to a good degree of certainty that the coin is fair? Maybe take a step back and ask yourself ‘do I trust the manufacturer of this coin?’. If this manufacturer has always had great quality (i.e. fair coins) then you would have some confidence that your case is no different.
- **Posterior**: The probability of the parameter **given** the evidence. The only way to know this value is to already have the evidence. However we can estimate the posterior in various ways. Think of it this way, given 100 coin flips with 47 heads and 53 tails what is the probability that theta is 0.5 (coin is fair)?
- **Likelihood**: The probability of the evidence **given** the parameter. Given that we know the coin is fair (theta = 0.5) what is the probability of having 47 heads out of 100 flips?
- **Evidence**: The probability of all possible outcomes. Probability of 1/100 heads, 2/100 heads, ..., 100/100 heads.

Conditioning your brain for LDA : We are starting with a coin flip, but the eventual goal is to link this back to words appearing in a document. Try to keep in mind that we think of a word similar to the outcome of a coin: a word exists in the document (heads!) or a word doesn’t exist in the document (tails!).

2.3 Maximum Likelihood

The simplest method of parameter estimation is the maximum likelihood method. Effectively we calculate the parameter that maximizes the likelihood.

$$\underbrace{p(\theta|D)}_{posterior} = \frac{\overbrace{\mathbf{p(D|\theta)}}^{\textbf{LIKELIHOOD}}\quad \overbrace{p(\theta)}^{prior}}{\underbrace{p(D)}_{evidence}} \tag{4}$$