Intro to Stan

2017 Oct 19

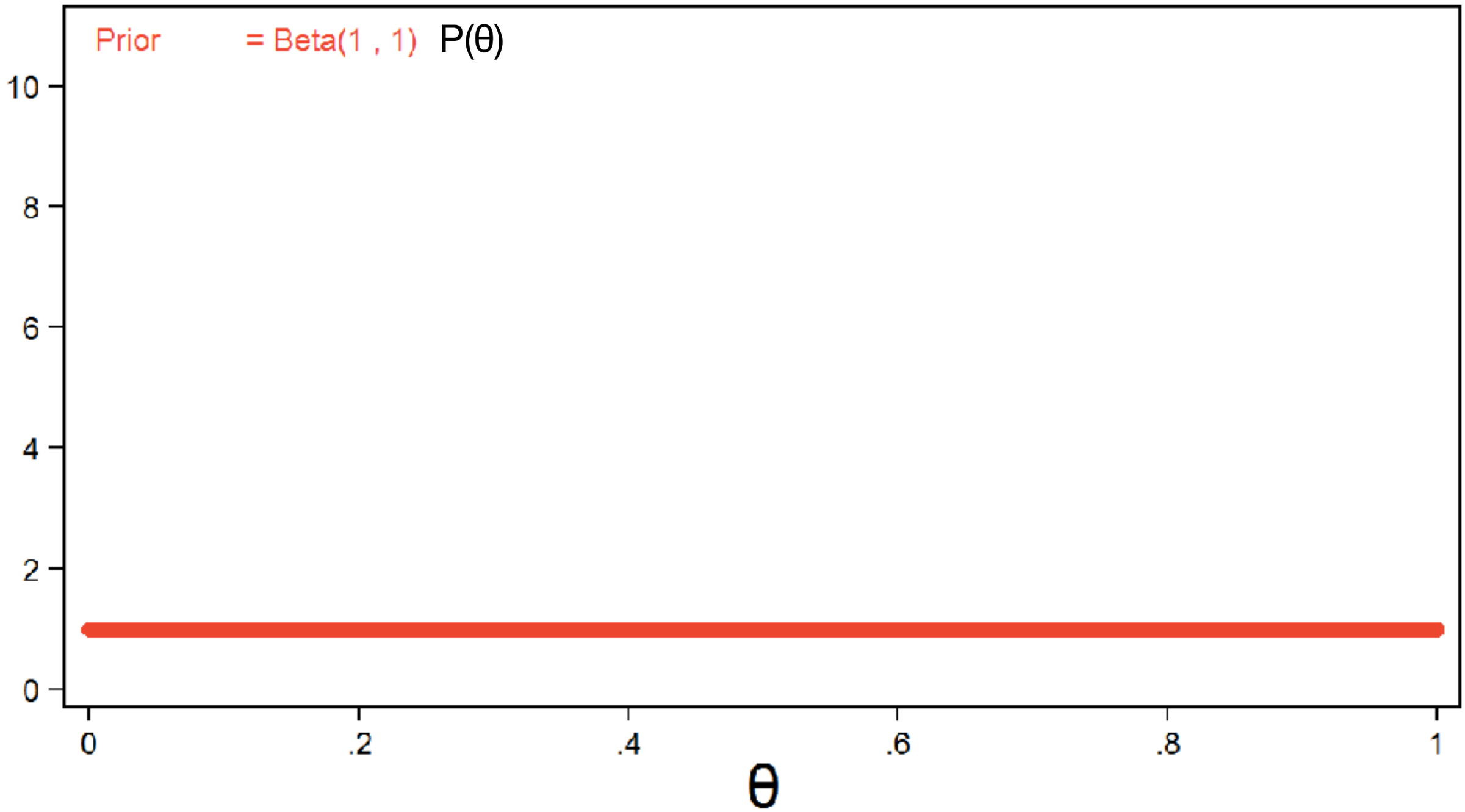
github.com/Sz-Tim/stan_intro

Bayesian Inference

- Frequentist: Parameters are fixed, but unknown
- Bayesian: Parameters are random variables described by probability distributions
- Bayes' Theorem
  - $P(\theta|y) = P(\theta) * P(y|\theta) / P(y)$
  - Posterior = Prior * Likelihood / constant

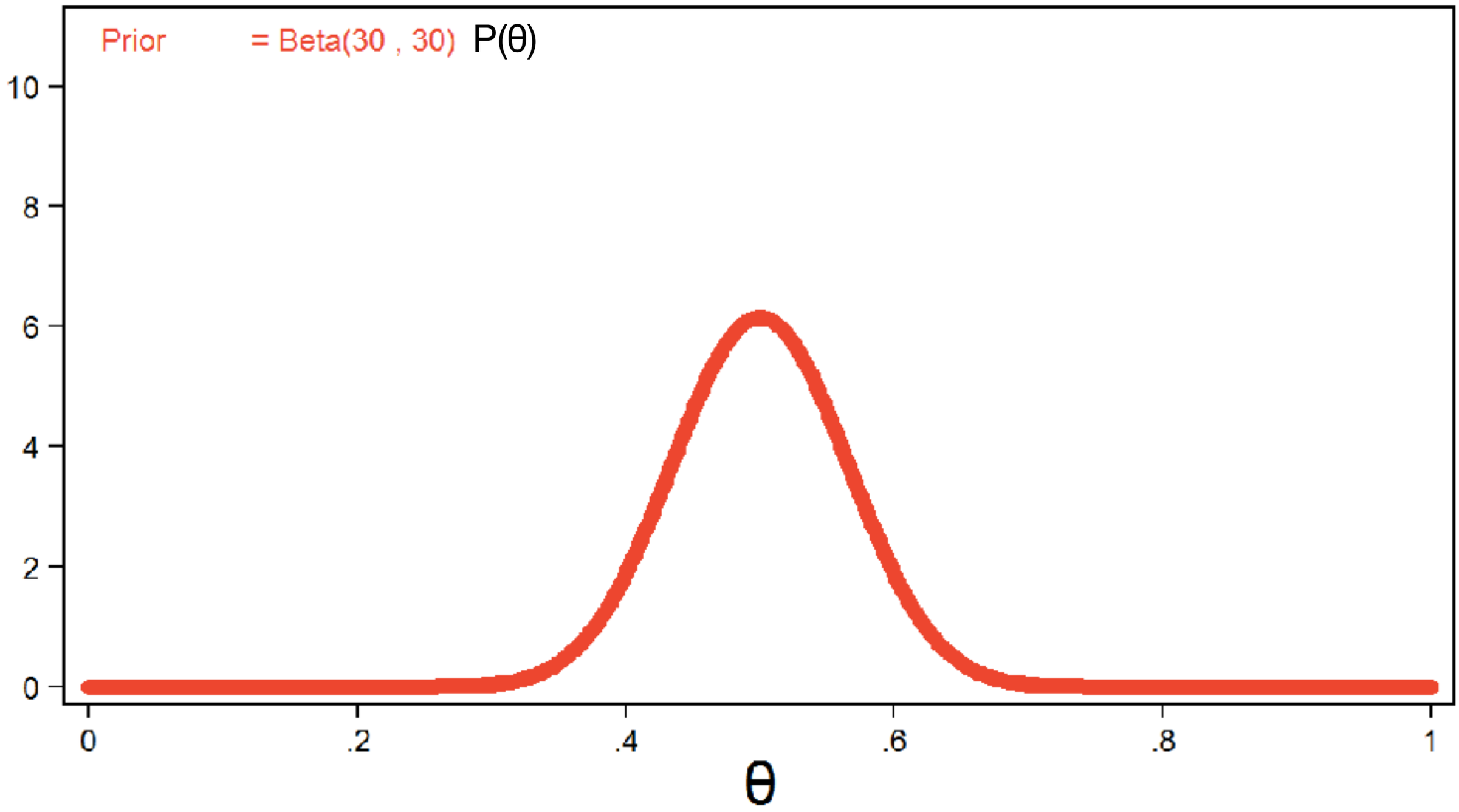
blog.stata.com/2016/11/01/introduction-to-bayesian-statistics-part-1-the-basic-concepts

Prior Distribution: P(θ) (Uninformed)



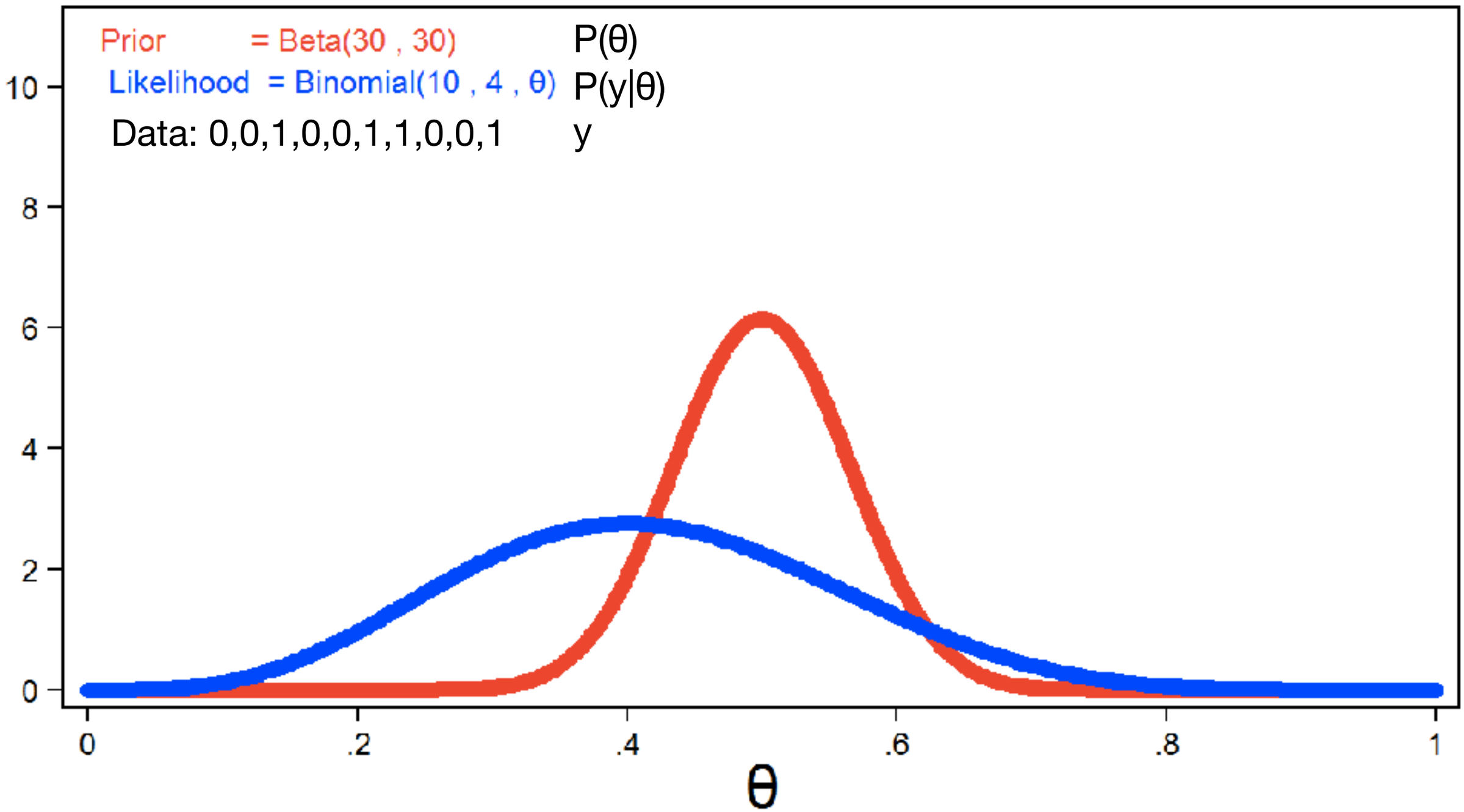
blog.stata.com/2016/11/01/introduction-to-bayesian-statistics-part-1-the-basic-concepts

Prior Distribution: P(θ) (Informed)



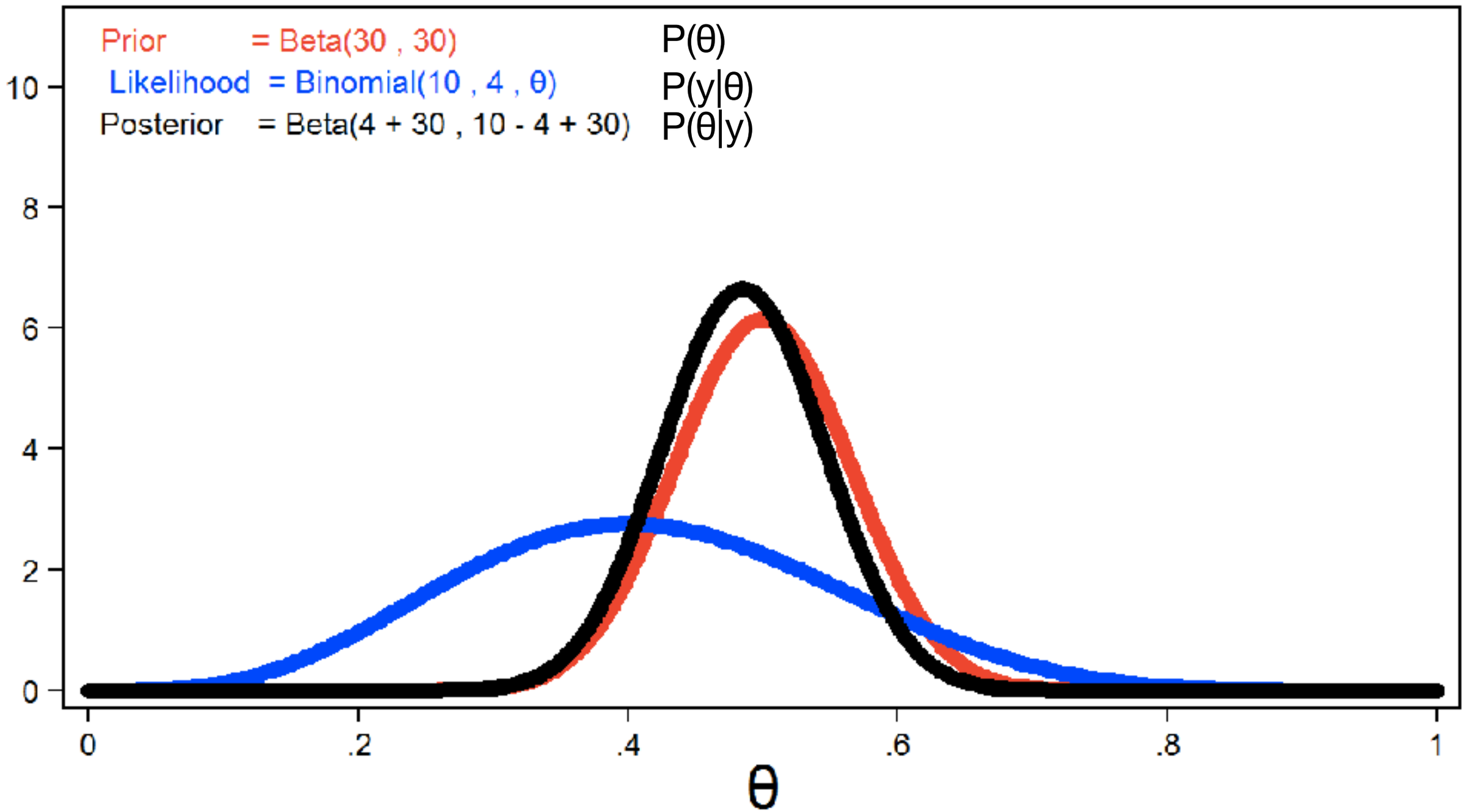
blog.stata.com/2016/11/01/introduction-to-bayesian-statistics-part-1-the-basic-concepts

Likelihood Function: P(y|θ)



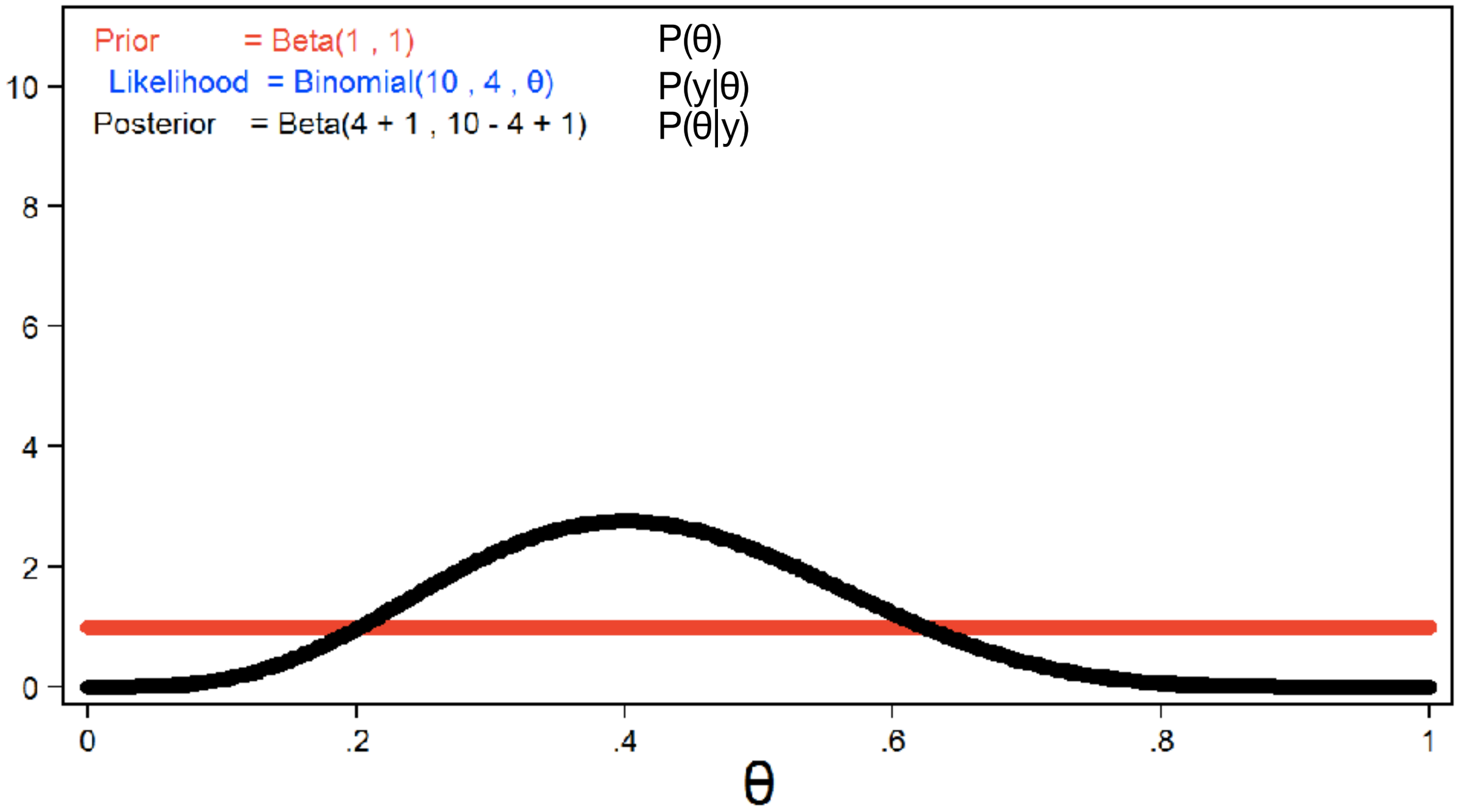
blog.stata.com/2016/11/01/introduction-to-bayesian-statistics-part-1-the-basic-concepts

Posterior Distribution: P(θ|y)



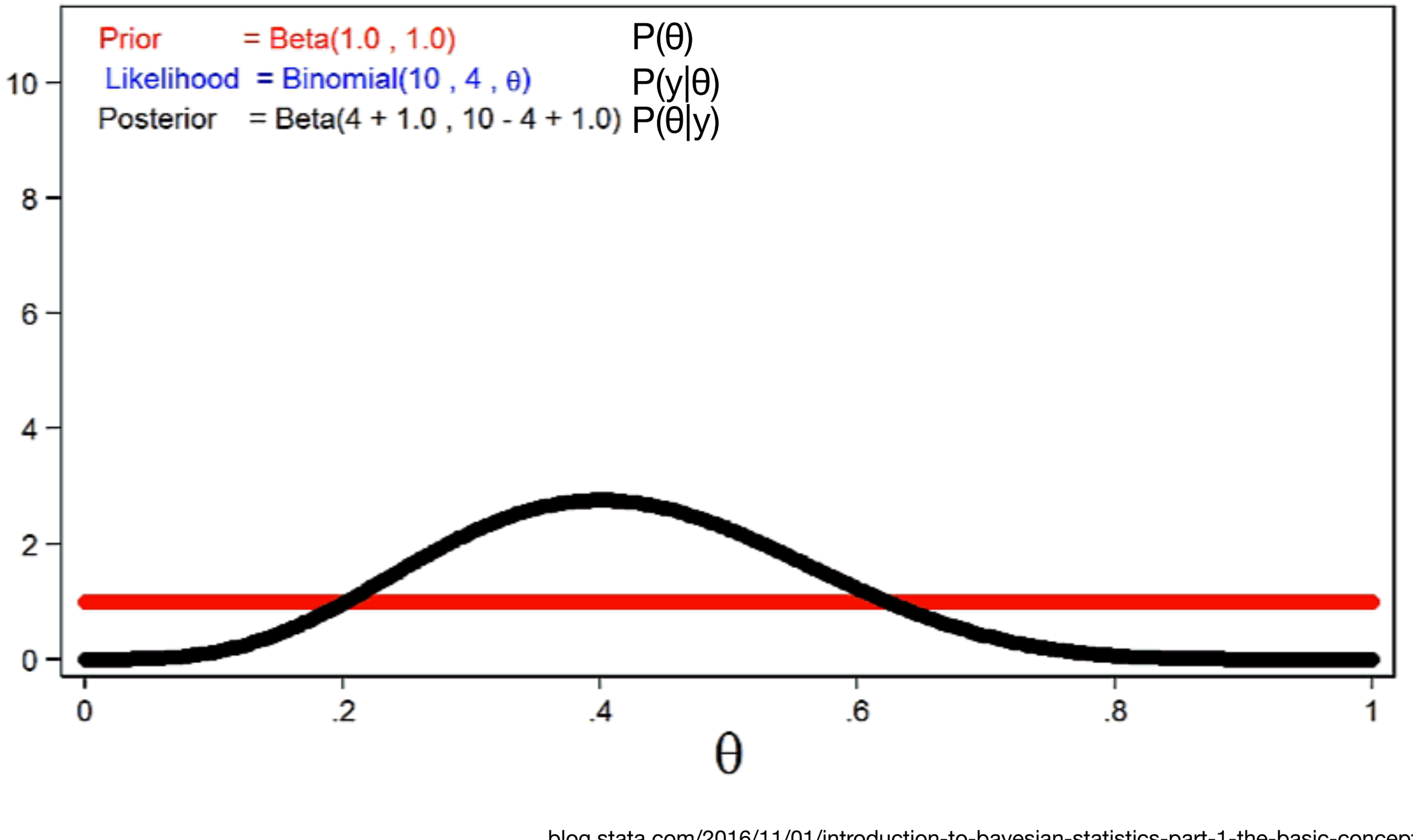
blog.stata.com/2016/11/01/introduction-to-bayesian-statistics-part-1-the-basic-concepts

Posterior Distribution: P(θ|y)



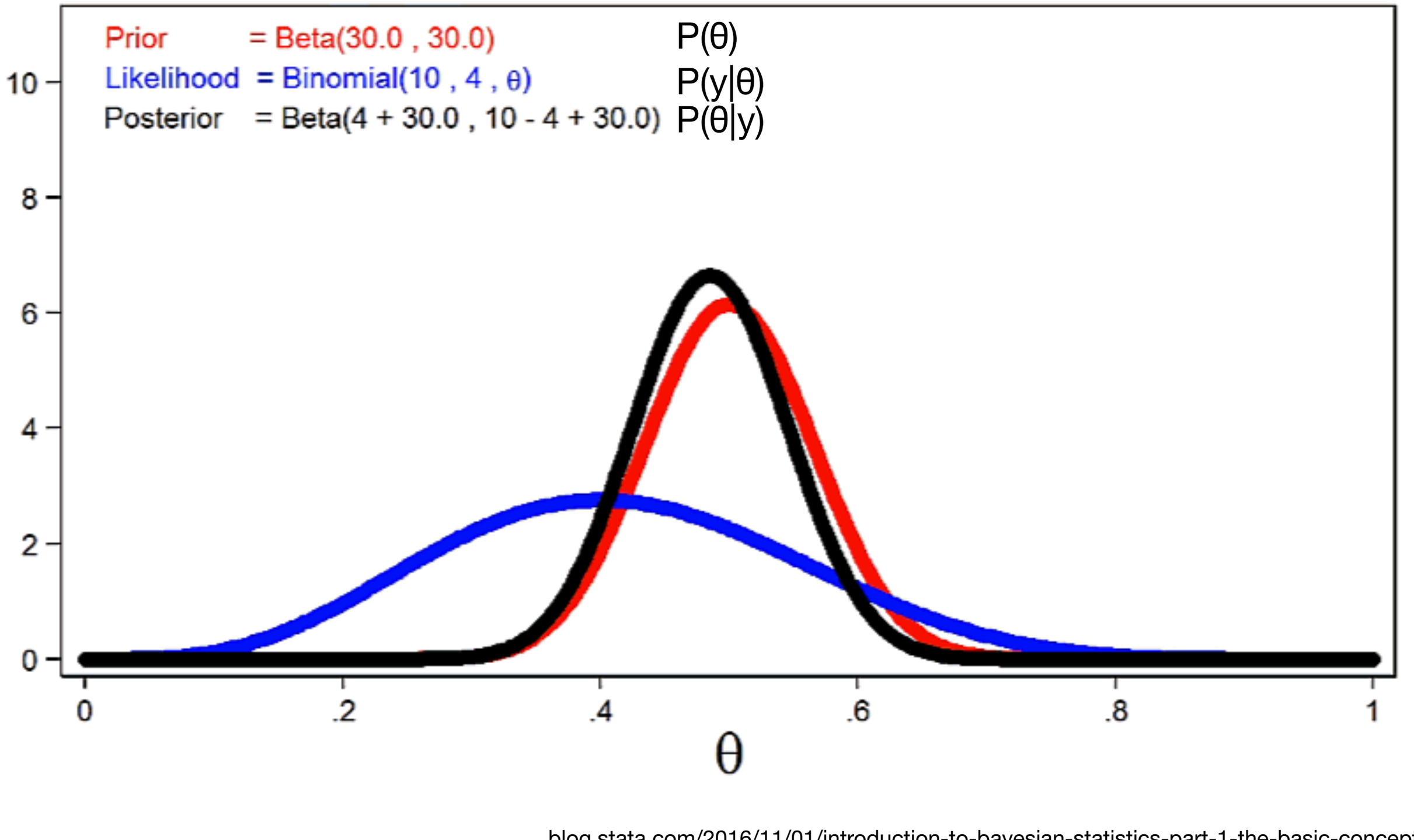
blog.stata.com/2016/11/01/introduction-to-bayesian-statistics-part-1-the-basic-concepts

Posterior Distribution: P(θ|y)



blog.stata.com/2016/11/01/introduction-to-bayesian-statistics-part-1-the-basic-concepts

Posterior Distribution: P(θ|y)

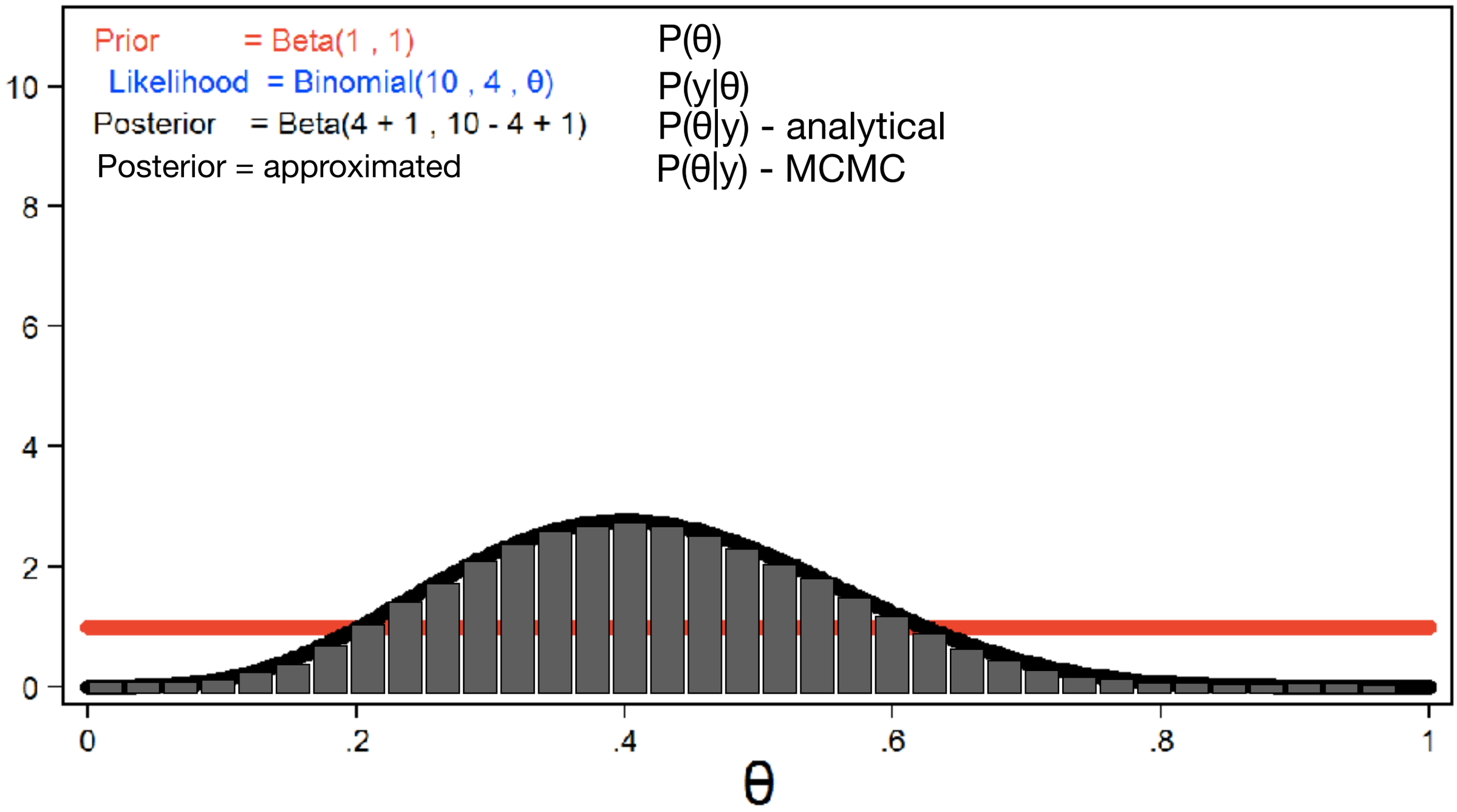


blog.stata.com/2016/11/01/introduction-to-bayesian-statistics-part-1-the-basic-concepts

Markov Chain Monte Carlo

- Estimate posterior probability distributions
- Avoids difficult/impossible analytical solutions
- In each iteration, draw parameter values & calculate likelihood
- Several common algorithms for draws

Posterior Distribution: P(θ|y)



blog.stata.com/2016/11/01/introduction-to-bayesian-statistics-part-1-the-basic-concepts

stan

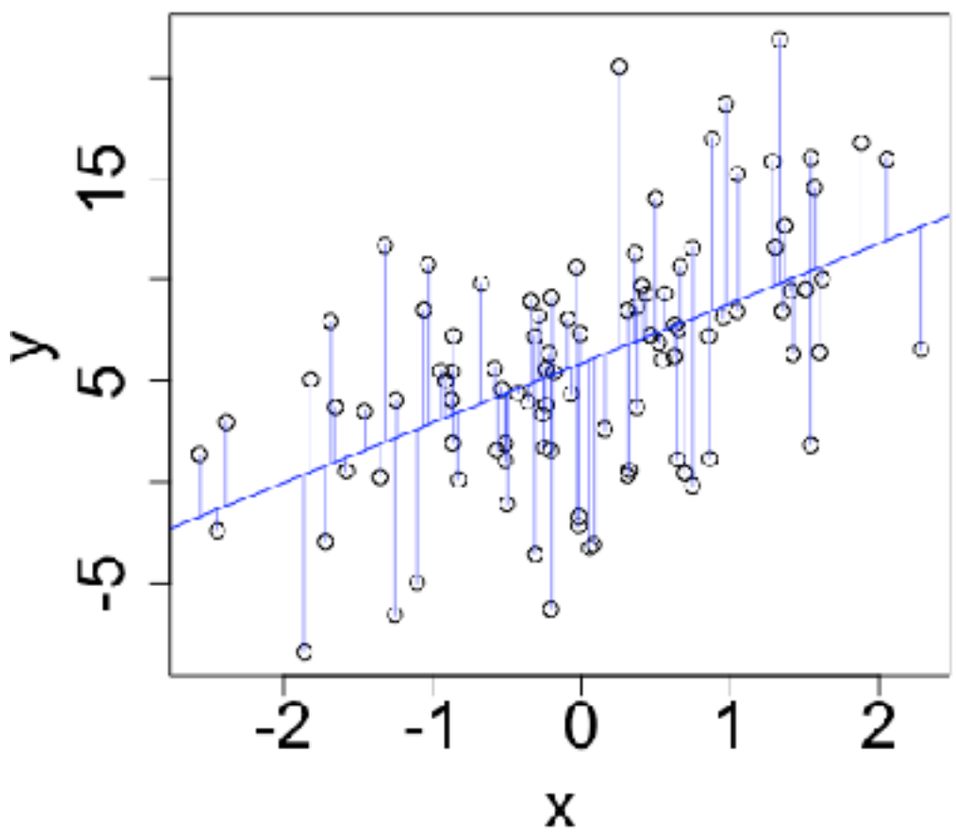
- Language/platform (2011)
- Open-source
- Many interfaces (R, python, Matlab, etc)
- Compiles model to C++
- Hamiltonian Monte Carlo (HMC) combined with a No U-Turn Sampler (NUTS)

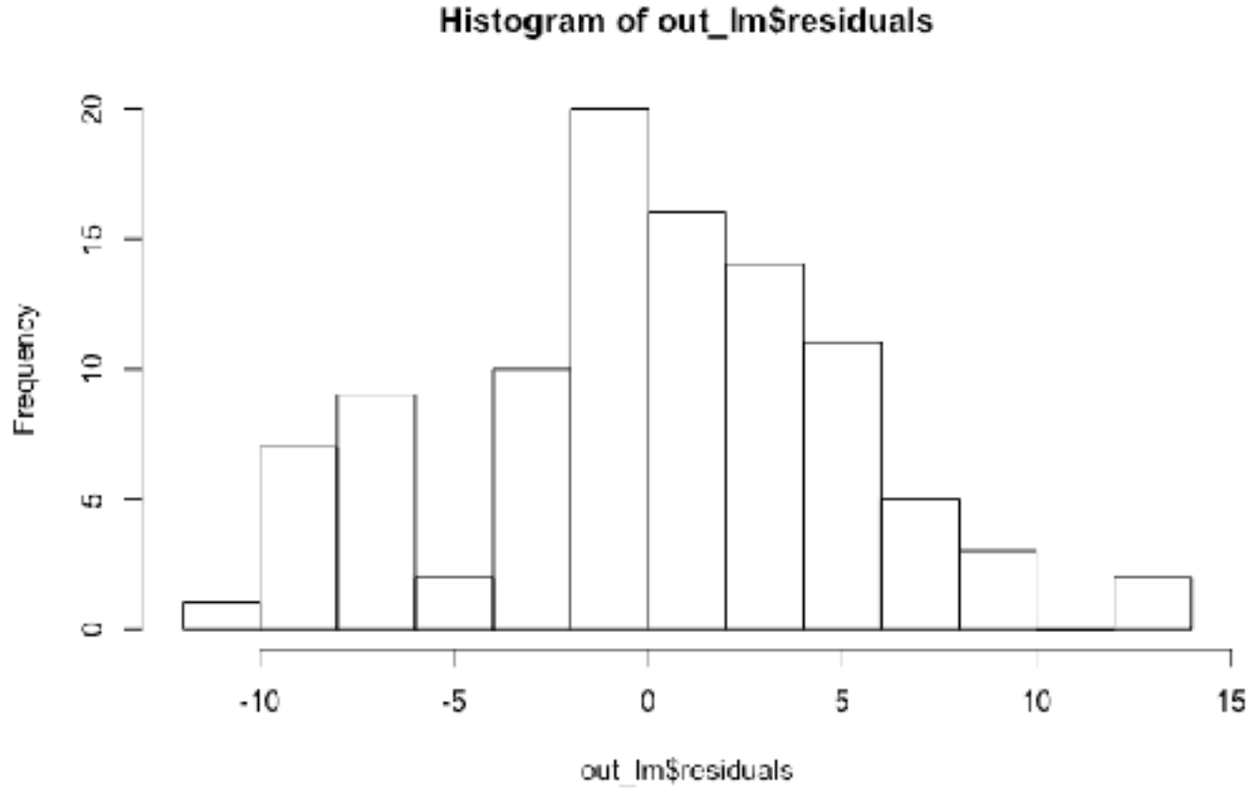
An example: Linear regression

$$y = bx + a\ (+ \varepsilon)$$

in R:

```
mod1 <- lm(y ~ x)
summary(mod1)
```



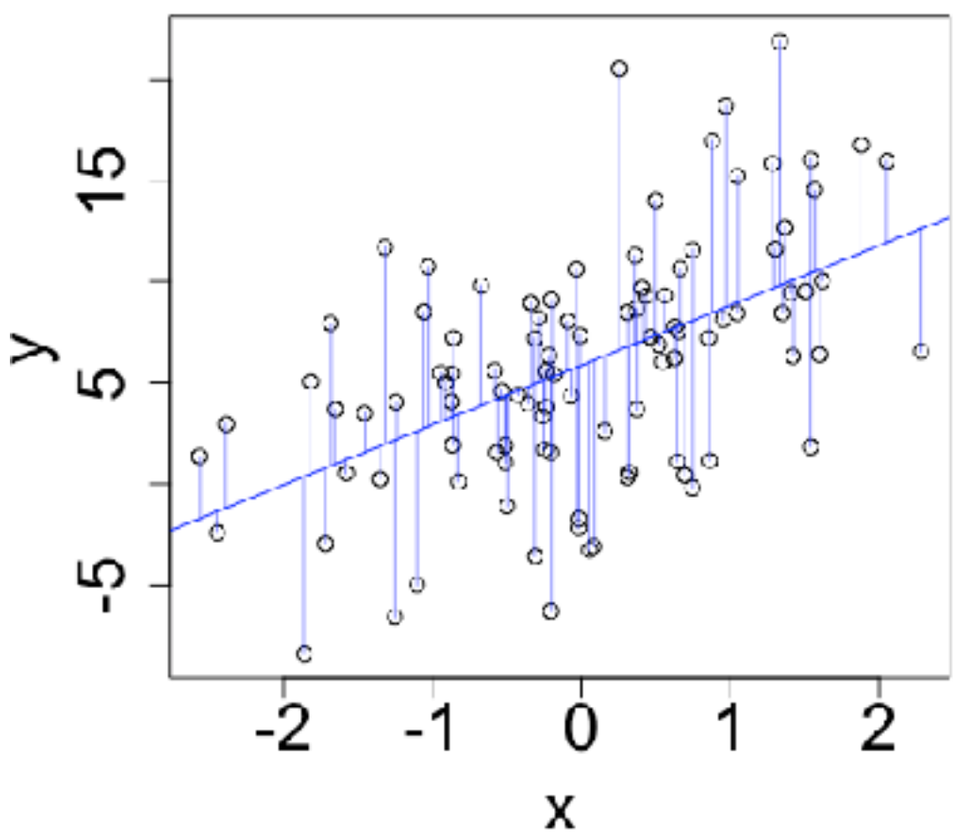


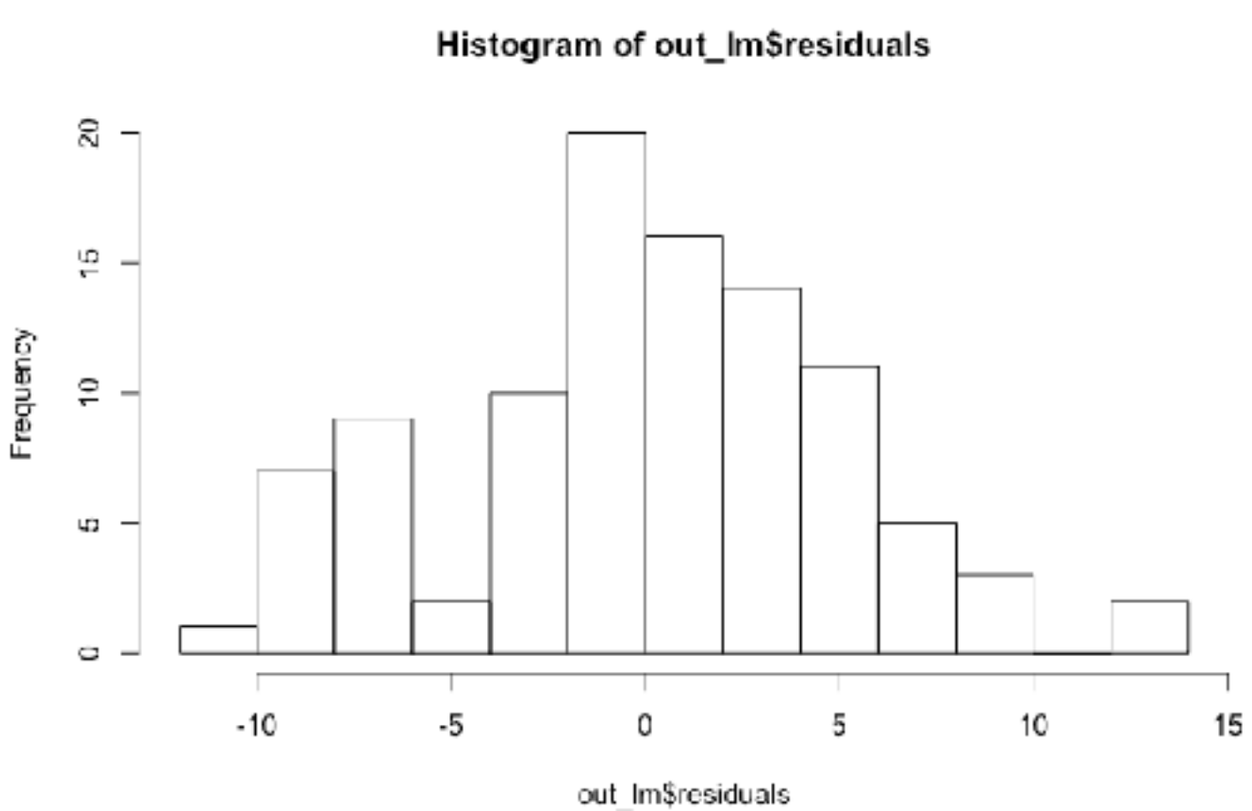
An example: Linear regression

$$y = bx + a\ (+ \varepsilon)$$

$$\varepsilon \sim \mathrm{Norm}(0, \sigma)$$

$$y \sim \mathrm{Norm}(bx + a, \sigma)$$





$$y \sim \text{Norm}(bx + a, \sigma)$$

in stan:

- Program blocks
- Explicitly declare variable type, size, & constraints
- End lines with ";"
- As in R, order matters

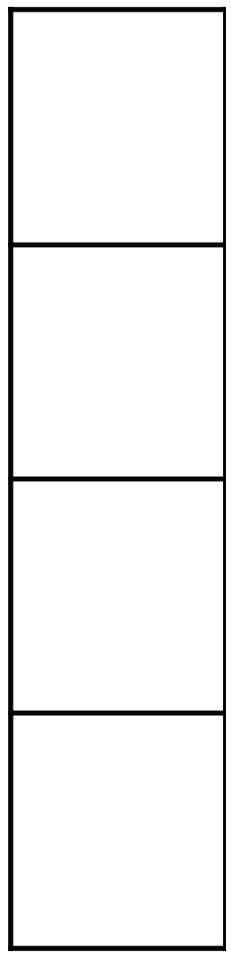
```
data {
 int n;
 vector[n] x;
 vector[n] y;
}
parameters {
 real a;
 real b;
 real<lower=0> sigma;
}
model {
 a ~ normal(0, 1);
 b ~ normal(0, 1);
 sigma ~ cauchy(0, 2.5);
 y ~ normal(b*x + a, sigma);
}
```

Data Types: R

- vector (1D)
- matrix (2D)
- array (nD)
- dataframe (2D)
- list

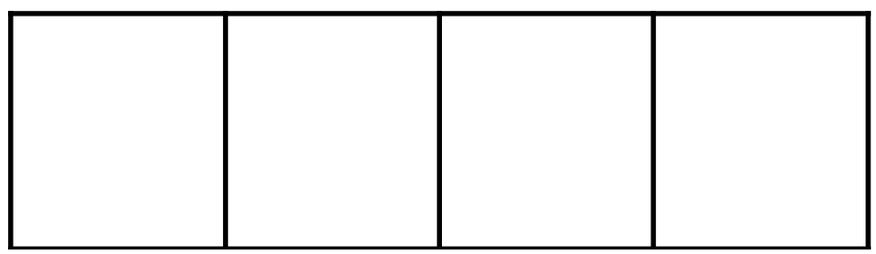
Data Types

- int
- real
- vector (1D)



Data Types

- int
- real
- vector (1D)
- row_vector (1D)



Data Types

- int
- real
- vector (1D)
- row_vector (1D)
- matrix (2D)
- arrays (nD)