Introduction to Bayesian Statistics for Demographic Science

Lab 1: Bayesian inference in Demography

Contents

> **Learning Objectives** In this lab, you will learn : - Bayesian inference in practice. - The basics of `CmdStanR`, - Specifying a binomial model in `Stan` and `CmdStanR`.

This Lab is based on workshop designed by Professor Arkadiusz Wiśniowski from the University of Manchester as part of the course on Demographic forecasting of the MSc in Social Research and Statistics.

Preliminary work

First, create a folder for this Lab and change the working directory to this folder in R

Then, install and load the R packages needed for this session.

```
## package 'tidyverse' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\andreaa\AppData\Local\Temp\RtmpCsgImm\downloaded_packages

## package 'ggpubr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\andreaa\AppData\Local\Temp\RtmpCsgImm\downloaded_packages
```

Bayesian inference in practice (basics)

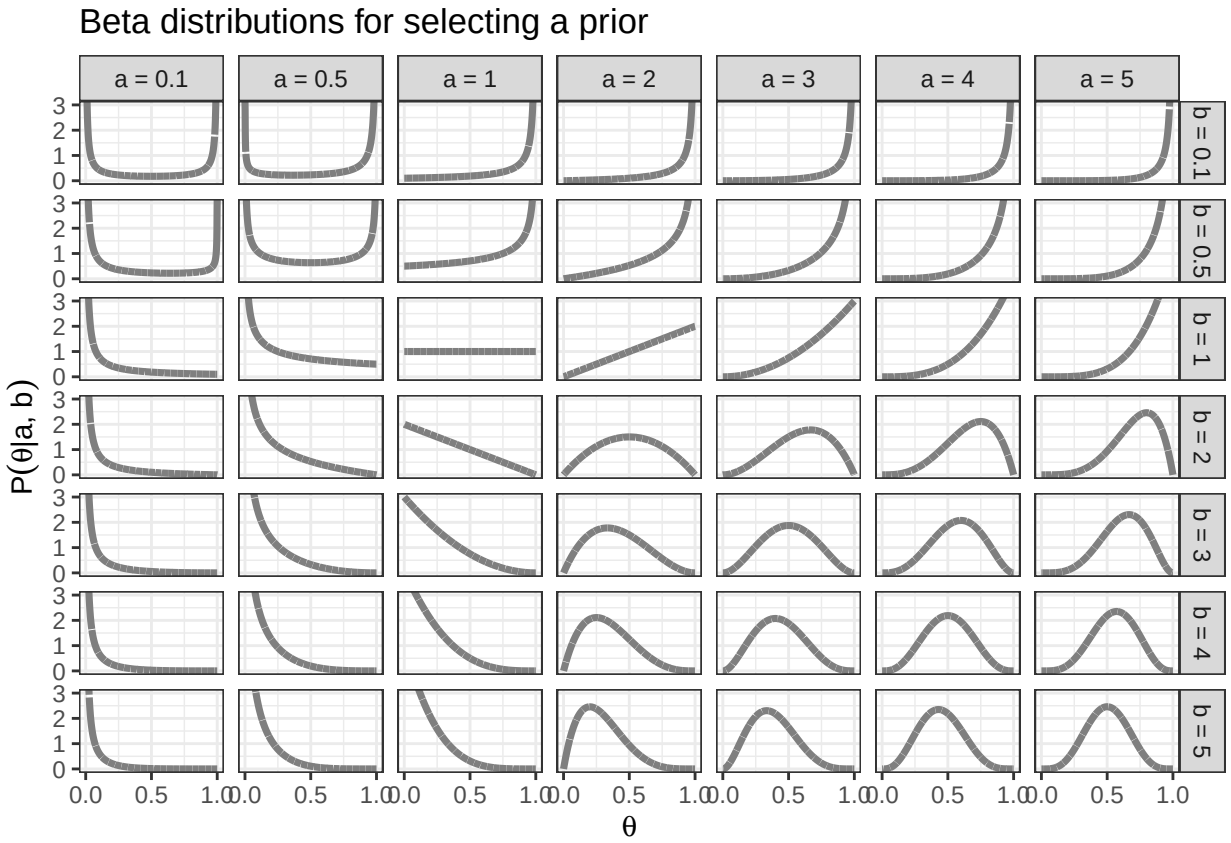
To illustrate how Bayesian inference works in practice, we are going to implement a binomial model which aims at estimating the probability of a baby being born in a given hospital. We want to know what the probability of the baby being a girl is. We can assume (and it is a reasonable assumption) that the outcome variable Y which takes values 0 (boy) or 1 (girl) is generated by Bernoulli trials with some unknown parameter θ - the probability of the baby being a girl. This is well-described by a **binomial model**: $P(girls = k, n|\theta)$.

- It is nice because if we assume the prior to be Beta distribution, we can derive the posterior by hand.

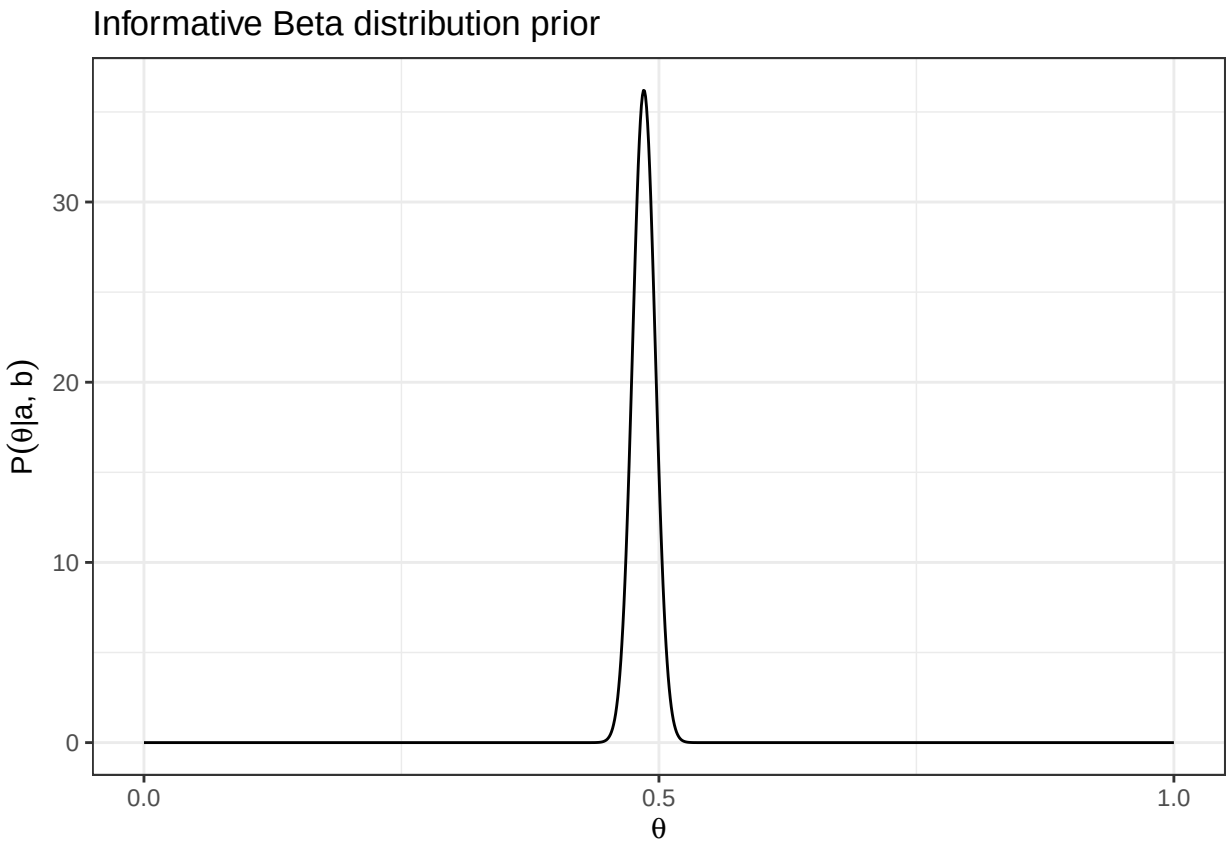
- Prior: $Beta(a, b)$ has expectation $a/(a + b)$
- **posterior** is also Beta!
- Posterior: $Beta(a + k, b + n - k)$, has expectation $(a + k)/(a + b + n)$
- Note: when posterior looks like prior we call the prior **conjugate**

Defining a prior distribution

What do (can) we know about the probability of the baby being a girl? Let's try some distributions. We know that the conjugate prior distribution is beta distribution.



We could define an informative prior based on demographic knowledge with expectation 0.4854369.



Simulating data

Based on demographic knowledge, we simulate data on the first thousand of babies born in some hospital since the beginning of year 2017 (1 denotes a girl, 0 - a boy).

```
set.seed(202)
girl <- rbinom(n = 1000, size = 1, prob = 0.4854369)
girl
```

```
##    [1] 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 1 1 0 0 1
##   [38] 0 1 1 1 1 1 1 0 1 0 1 1 1 0 1 0 1 1 1 0 0 1 1 1 1 1 1 0 0 0 0 1 0 0 1 0 0 1
##   [75] 1 1 0 0 0 0 1 1 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 0 0 0 1 1 0 0 1 1 0 1 0 1
##  [112] 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 1 1 0 0 1 0 1 0 1
##  [149] 0 1 0 1 0 0 0 0 1 0 1 0 1 0 1 1 1 0 0 0 1 0 1 1 0 1 1 1 1 1 1 0 0 0 1 1 0 1
##  [186] 0 0 0 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 1 0 0 1 1 0 0 0 1
##  [223] 0 0 0 1 1 0 1 1 0 1 0 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 1 1
##  [260] 0 0 1 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1 0 1 1 0 0 1 1 0 0 1 1 1 0 0 1 0 1
##  [297] 0 0 1 0 0 0 1 0 1 1 1 0 0 0 1 1 0 0 1 1 1 1 0 0 1 0 1 0 1 0 1 0 1 1 1 1 1 0
##  [334] 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 0 1 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1 0 0 0
##  [371] 0 0 1 0 0 0 1 1 0 1 1 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 1 0 1 0 0 1 1 1 1 1 0 1 1 1
##  [408] 0 1 1 1 1 0 0 0 1 0 0 1 1 1 1 1 1 0 1 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0 1 0 0
##  [445] 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 1 0 1 1 0 1 1 1 0 1 1 1 0 0 1 0 1 0 1 0
##  [482] 1 0 1 1 0 1 1 1 1 1 1 1 0 1 0 0 0 1 0 0 0 1 1 0 1 1 1 0 1 0 1 0 1 0 1 0 0 0
##  [519] 1 0 1 1 0 1 0 0 0 0 0 1 0 0 1 1 1 1 0 1 1 0 0 1 0 1 1 0 0 1 1 0 1 1 0 0 1
```

```
##  [556] 0 1 1 1 0 0 0 0 1 0 0 0 1 0 0 1 0 0 1 1 0 0 0 0 1 0 1 1 1 1 1 0 1 1 1 0 1
##  [593] 1 0 1 1 0 1 1 1 0 1 1 1 0 0 0 0 1 0 0 1 0 1 0 0 0 1 1 0 0 1 1 0 0 0 1 0 1
##  [630] 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 0 1 1 1 1
##  [667] 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 1 1 1 0 1 1 1 1 0 1 0
##  [704] 0 1 0 1 0 0 1 0 0 0 1 1 0 1 1 0 1 1 0 0 0 1 0 0 1 1 1 1 1 1 0 1 0 0 0 0 1 1
##  [741] 1 0 1 0 0 0 1 1 0 0 1 1 1 1 1 1 0 0 1 1 0 0 1 1 1 0 1 1 1 0 1 0 0 1 0 0 1 0
##  [778] 1 0 1 0 1 0 1 0 0 0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 1 0 0 1 0 1 1 1 1 1 0 1 0 0
##  [815] 0 0 0 0 1 1 0 1 1 1 0 0 1 1 1 1 1 1 1 0 1 1 0 0 1 1 0 1 0 0 0 0 0 1 1 0 1 0 0
##  [852] 0 1 1 1 0 1 1 0 0 1 0 0 1 1 0 1 1 1 0 0 1 0 1 0 0 1 0 1 0 0 1 1 0 0 0 1 0
##  [889] 1 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0 1 0 1 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 1 0 0 0
##  [926] 1 1 1 0 1 1 1 0 1 1 1 0 0 1 0 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1
##  [963] 0 0 1 1 1 1 0 1 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 1 1 0
## [1000] 0
```

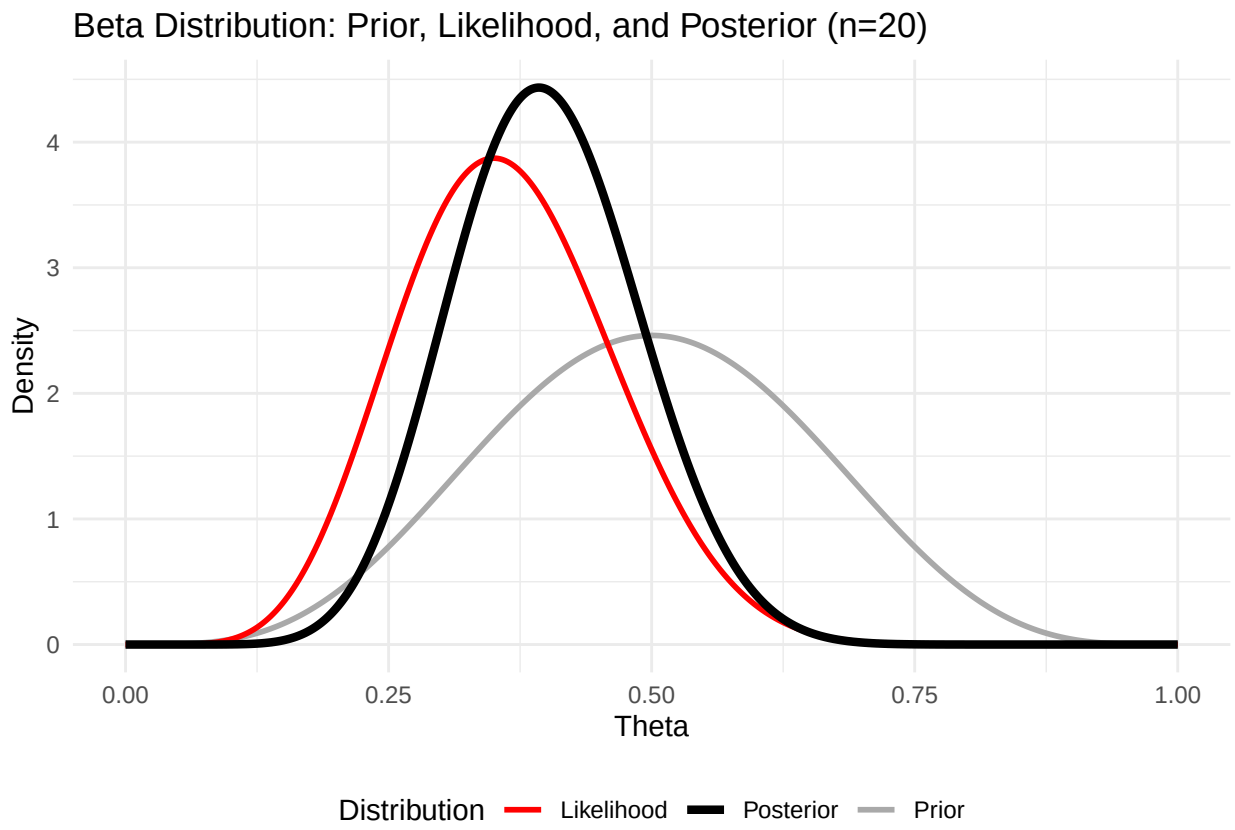
Plotting prior, likelihood and posterior distributions

- What is the posterior distribution of parameter θ if we use conjugate priors?
- Plot prior distributions along the corresponding likelihood and posteriors.

```
a <- 5
b <- 5
th <- seq(0, 1, by = 0.001)
k <- sum(girl[1:20]) # sample size:  20 observations
n <- length(girl[1:20])

distr <- data.frame(th)
distr$prior <- dbeta(th, a, b)
distr$likelihood <- dbeta(th, k + 1, n - k + 1)
distr$posterior <- dbeta(th, a + k, b + n - k)


d1 <- ggplot(data = distr, aes(x = th)) +
  geom_line(aes(y = prior, color = "Prior"), size = 1) +
  geom_line(aes(y = likelihood, color = "Likelihood"), size = 1) +
  geom_line(aes(y = posterior, color = "Posterior"), size = 1.5) +
  labs(title = "Beta Distribution: Prior, Likelihood, and Posterior (n=20)",
       x = "Theta",
       y = "Density") +
  theme_minimal() +
  scale_color_manual(values = c("Prior" = "darkgrey", "Likelihood" = "red",
                                "Posterior" = "black"), name = "Distribution")+
  theme(legend.position = "bottom")
d1
```



Non-informative vs informative priors - How the posterior distribution of getting a girl changes given an informative prior with $a = 1000, b = 1000(1 - 0.4854369)/0.4854369)$?

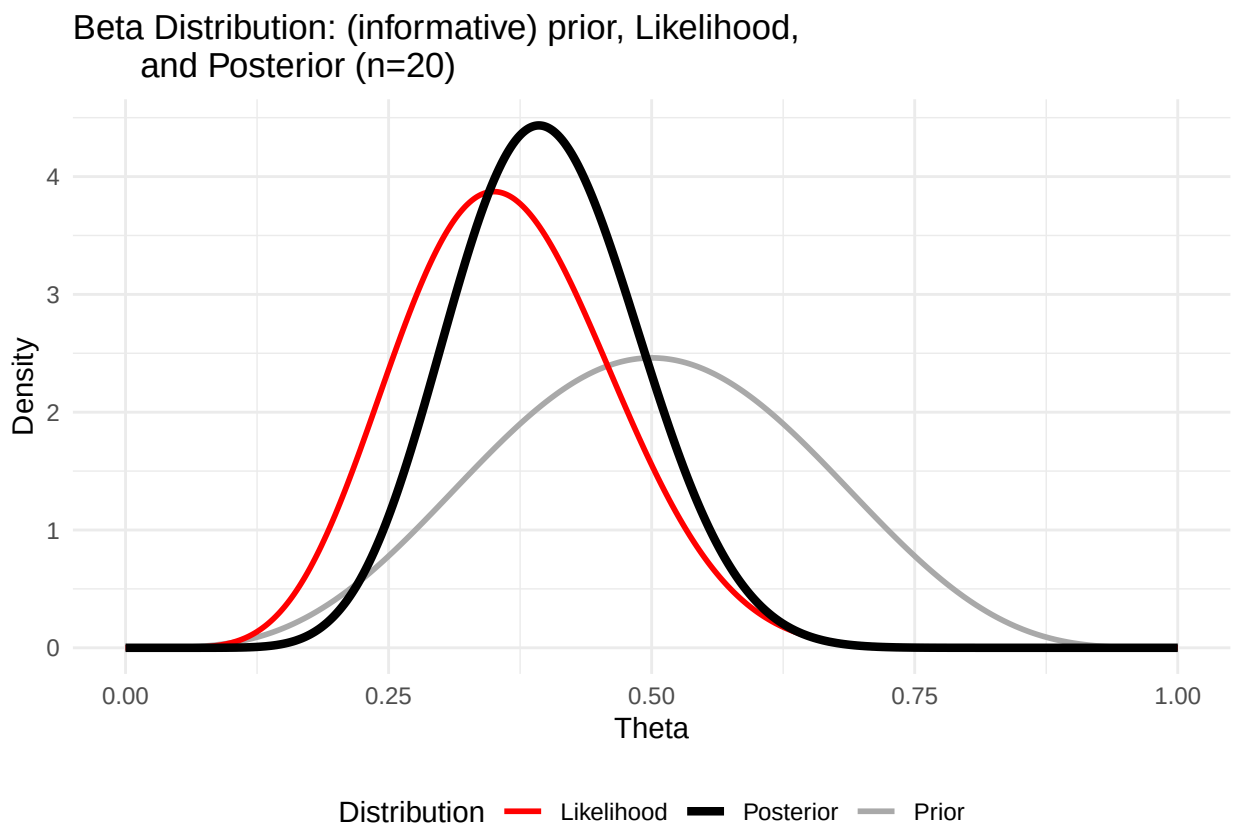
```
th <- seq(0, 1, by = 0.001)
a <- 1000
b <- 1000 * (1 - 0.4854369) / 0.4854369
#informative prior with expectation 0.4854369

a=5
b=5
k <- sum(girl[1:20]) # sample size:  20 observations
n <- length(girl[1:20])

distr <- data.frame(th)
distr$prior <- dbeta(th, a, b)
distr$likelihood <- dbeta(th, k + 1, n - k + 1)
distr$posterior <- dbeta(th, a + k, b + n - k)


d1i <- ggplot(data = distr, aes(x = th)) +
  geom_line(aes(y = prior, color = "Prior"), size = 1) +
  geom_line(aes(y = likelihood, color = "Likelihood"), size = 1) +
  geom_line(aes(y = posterior, color = "Posterior"), size = 1.5) +
  labs(title = "Beta Distribution: (informative) prior, Likelihood,
       and Posterior (n=20)",
       x = "Theta",
       y = "Density") +
```

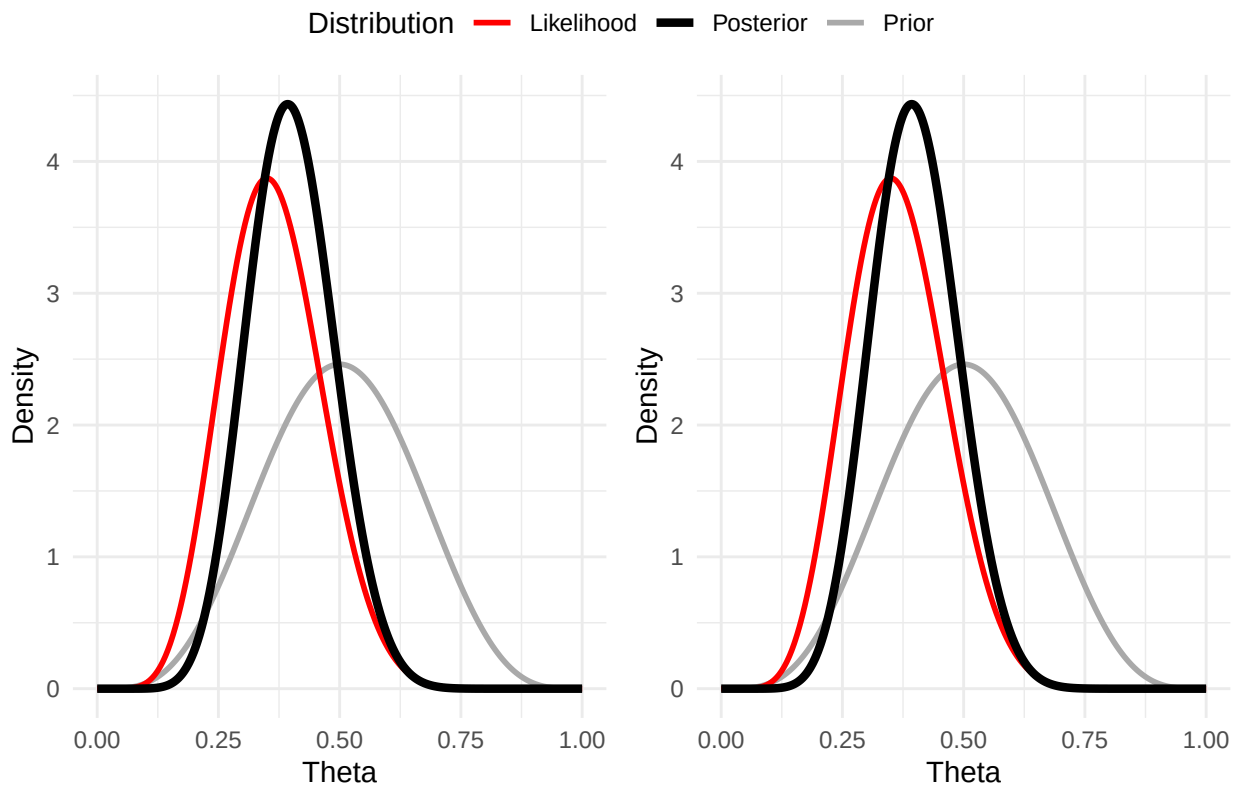
```
  theme_minimal() +
  scale_color_manual(values = c("Prior" = "darkgrey", "Likelihood" = "red",
                                "Posterior" = "black"), name = "Distribution")+
  theme(legend.position = "bottom")
d1i
```



Arrange the plots in a single layout.

```
comb_plot <- ggarrange(d1 + theme(plot.title = element_blank()),
          d1i + theme(plot.title = element_blank()), ncol = 2, nrow = 1,
          common.legend = T)

annotate_figure(
  comb_plot,
  top = text_grob("Beta Distribution: Prior, Likelihood, and Posterior (n=20)",
                  size = 14, face = "bold"),
  right = NULL)
```



Assessing the impact of different sample sizes

Compute the mean for the first 20 observations of the simulated data. Then, calculate the same for $n = 50$, $n = 100$, and $n = 1000$.

What is the frequentist estimator of for each sample size? Compared it with the results obtained in the previous exercise with n=20.

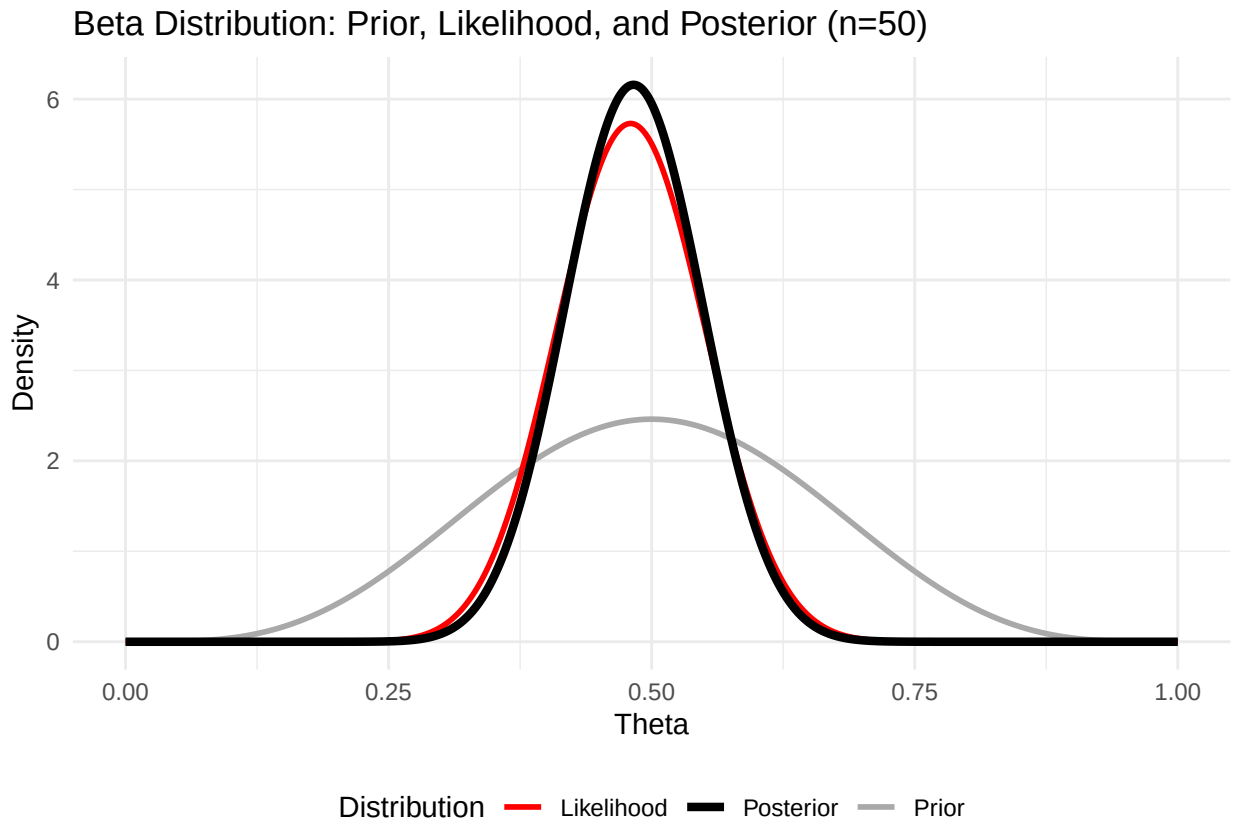
Plot the data to see the distribution of the simulated data of different sample sizes (n=50, 100, 1000). Use the prior if the previous task (i.e. $a = 5, b = 5$).

```
# plotting prior
k <- sum(girl[1:50]) # sample size: 50 observations
n <- length(girl[1:50])


distr <- data.frame(th)
distr$prior <- dbeta(th, a, b)
distr$likelihood <- dbeta(th, k + 1, n - k + 1)
distr$posterior <- dbeta(th, a + k, b + n - k)


d2 <- ggplot(data = distr, aes(x = th)) +
  geom_line(aes(y = prior, color = "Prior"), size = 1) +
  geom_line(aes(y = likelihood, color = "Likelihood"), size = 1) +
  geom_line(aes(y = posterior, color = "Posterior"), size = 1.5) +
```

```
  labs(title = "Beta Distribution: Prior, Likelihood, and Posterior (n=50)",
       x = "Theta",
       y = "Density") +
  theme_minimal() +
  scale_color_manual(values = c("Prior" = "darkgrey", "Likelihood" = "red",
                                "Posterior" = "black"), name = "Distribution") +
  theme(legend.position = "bottom")
d2
```



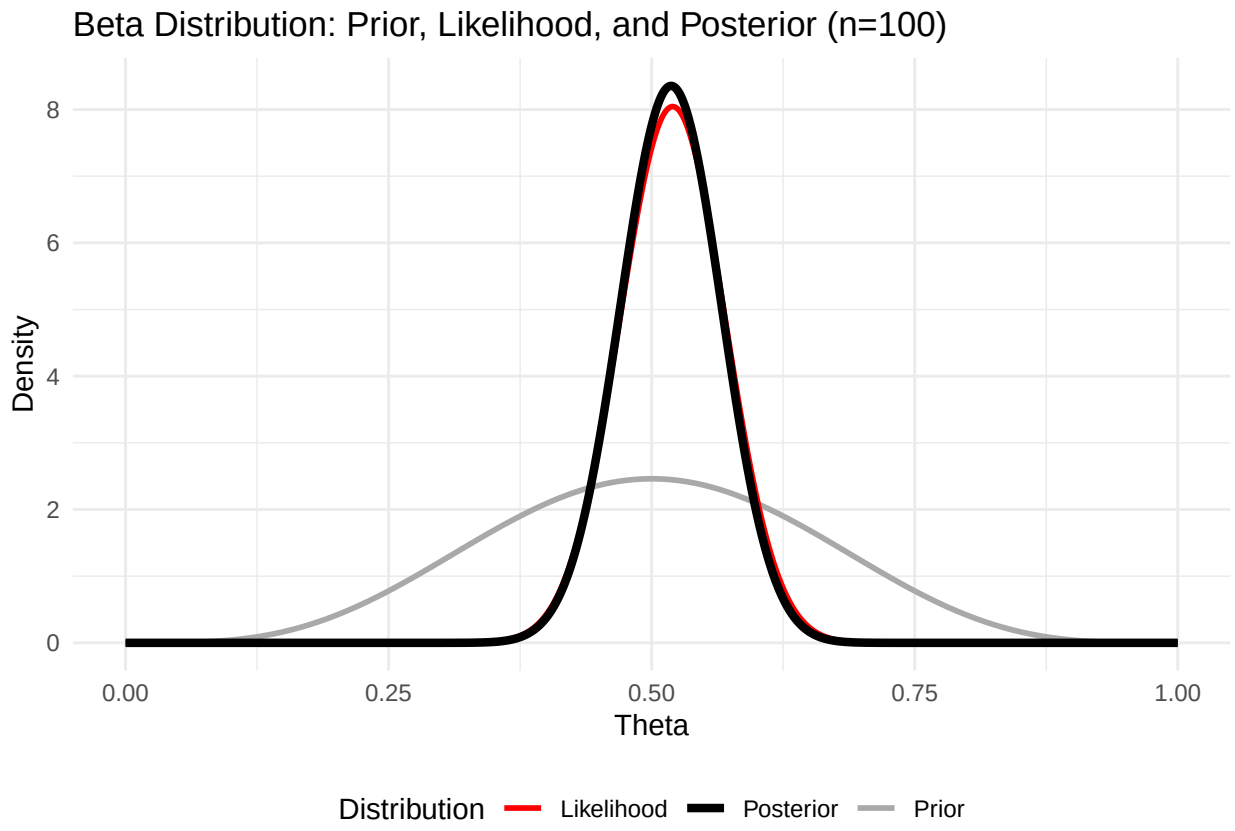
Keeping the same prior, increase sample size to $n = 100$

```
# plotting prior
k <- sum(girl[1:100]) # sample size:  100 observations
n <- length(girl[1:100])


distr <- data.frame(th)
distr$prior <- dbeta(th, a, b)
distr$likelihood <- dbeta(th, k + 1, n - k + 1)
distr$posterior <- dbeta(th, a + k, b + n - k)


d3 <- ggplot(data = distr, aes(x = th)) +
  geom_line(aes(y = prior, color = "Prior"), size = 1) +
  geom_line(aes(y = likelihood, color = "Likelihood"), size = 1) +
  geom_line(aes(y = posterior, color = "Posterior"), size = 1.5) +
```

```
  labs(title = "Beta Distribution: Prior, Likelihood, and Posterior (n=100)",
       x = "Theta",
       y = "Density") +
  theme_minimal() +
  scale_color_manual(values = c("Prior" = "darkgrey", "Likelihood" = "red",
                                "Posterior" = "black"), name = "Distribution") +
  theme(legend.position = "bottom")
d3
```



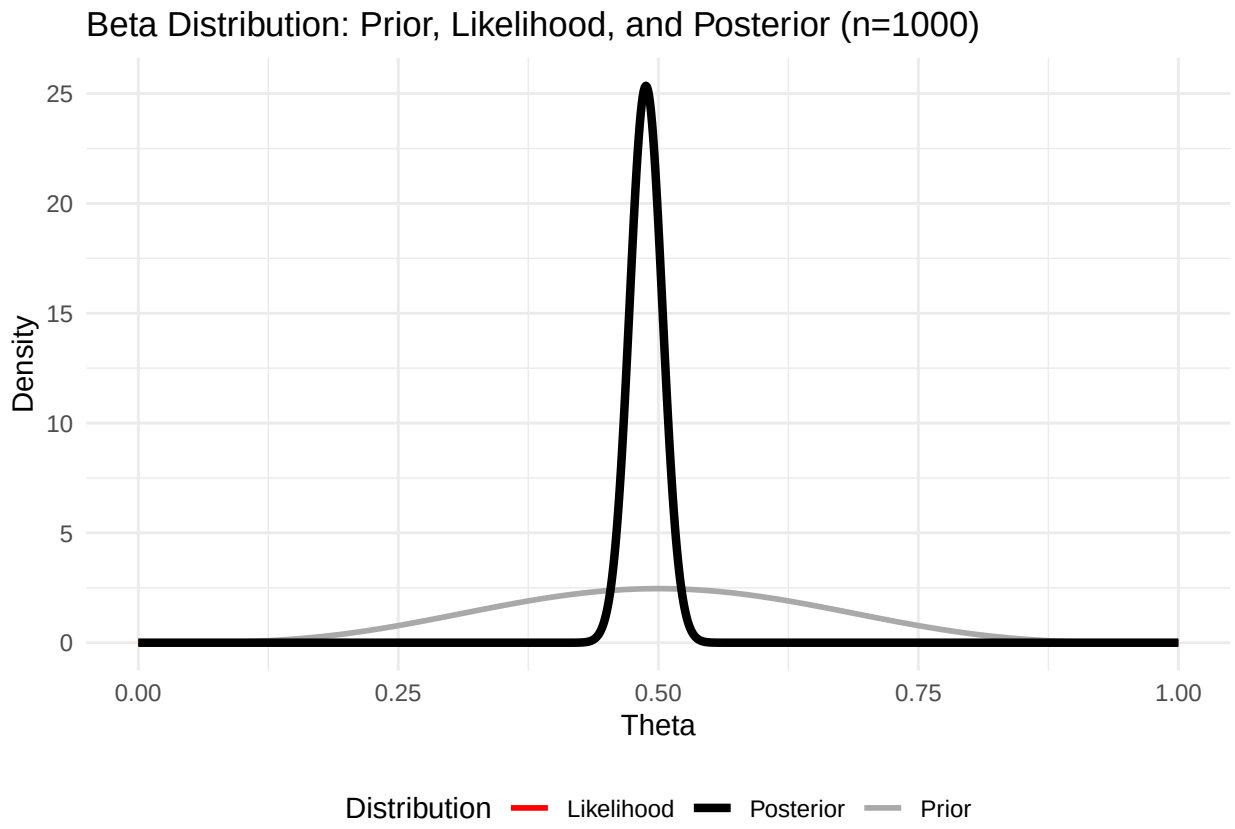
Now, use the whole sample (i.e. $n = 1000$).

```
# plotting prior
k <- sum(girl) # sample size:  1000 observations
n <- length(girl)


distr <- data.frame(th)
distr$prior <- dbeta(th, a, b)
distr$likelihood <- dbeta(th, k + 1, n - k + 1)
distr$posterior <- dbeta(th, a + k, b + n - k)


d4 <- ggplot(data = distr, aes(x = th)) +
  geom_line(aes(y = prior, color = "Prior"), size = 1) +
  geom_line(aes(y = likelihood, color = "Likelihood"), size = 1) +
  geom_line(aes(y = posterior, color = "Posterior"), size = 1.5) +
```

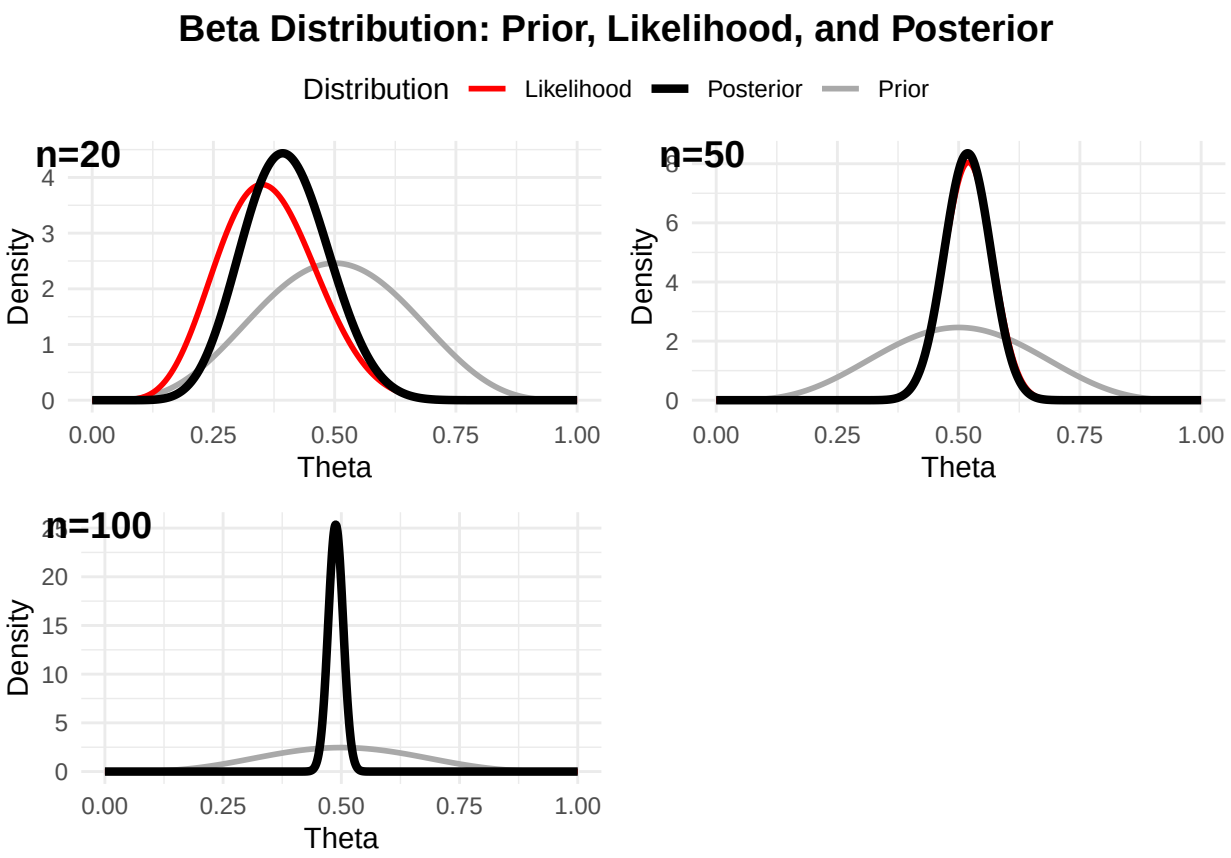
```
  labs(title = "Beta Distribution: Prior, Likelihood, and Posterior (n=1000)",
       x = "Theta",
       y = "Density") +
  theme_minimal() +
  scale_color_manual(values = c("Prior" = "darkgrey", "Likelihood" = "red",
                                "Posterior" = "black"), name = "Distribution") +
  theme(legend.position = "bottom")
d4
```



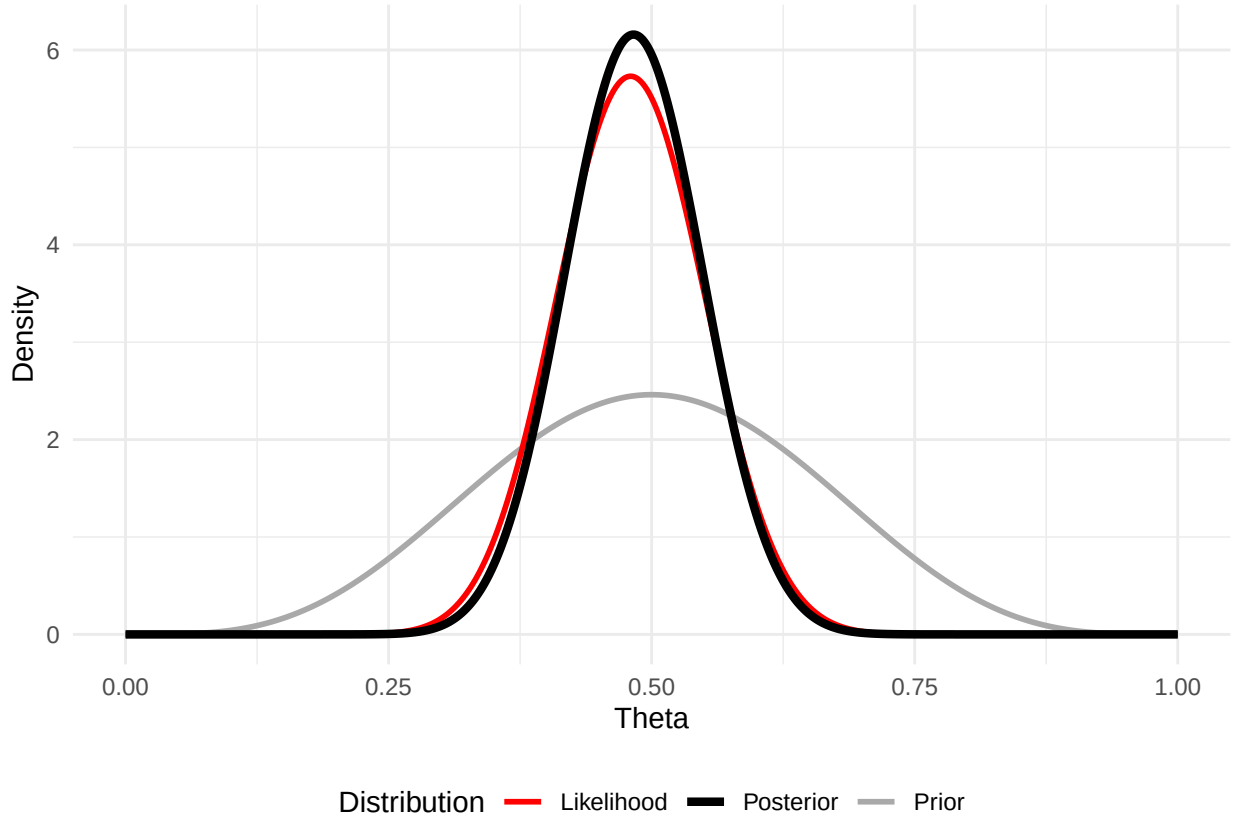
Arrange the plots in a single layout.

```
comb_plot <- ggarrange(d1 + theme(plot.title = element_blank()),
          d3 + theme(plot.title = element_blank()),
          d4 + theme(plot.title = element_blank()), ncol = 2, nrow = 2,
          labels = c("n=20", "n=50", "n=100", "n=1000"), common.legend = T)

annotate_figure(
  comb_plot,
  top = text_grob("Beta Distribution: Prior, Likelihood, and Posterior",
                  size = 14, face = "bold"),
  right = NULL)
```




```
d2 + theme(plot.title = element_blank())
```



- How does our inference change when we take sample size $n = 50$, $n = 100$, $n = 1000$?

Specifying a binomial model using `Stan` and `CmdStanR`.

There is different software to carry out Bayesian inference. One of the most recent and popular software is `Stan`, which we are using in this lab for estimating the probability that aa newborn in a given hospital is a girl as in the previous section.

Basics of `Stan` and `CmdStanR`

To use `Stan` in `R`, we need the to install and load the `CmdStanR` package. Depending on your computer, you will be asked to install `Rtools`. Install it and be patient, given that it takes a little bit of time.

```
install.packages("cmdstanr", repos = c('https://stan-dev.r-universe.dev',
                                       getOption("repos")))
```

```
## package 'cmdstanr' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\andreaa\AppData\Local\Temp\RtmpCsgImm\downloaded_packages
```

```
library("cmdstanr")
library("posterior")
```

```
library("bayesplot")
color_scheme_set("brightblue")
```

You can write a `Stan` program in a text editor, saving them with the extension `.stan` at the end of the file name (e.g. `model.stan`). You can also write a `Stan` program in RStudio as we are going to do in this lab.

Writing a model in `Stan`

Stan works in six blocks of code, which correspond to `data`, `parameters`, `model`, `transformed data`, `transformed parameters`, and `generated quantities`. You need to define at least the three first blocks.

- The `data` block comprehends the input data. You need to define all the elements that your model requires. For our case, the elements are the sample size and the data denoting whether a girl or boy is born.
- The `parameters` block includes the parameters of our interest. In out case is θ, which is bounded between 0 and 1 due to being a probability.
- In the `model` block, you need to specify not only the likelihood for a model but also its priors.

```
## In file included from stan/src/stan/model/model_header.hpp:11:
## stan/src/stan/model/model_base_crtp.hpp:205: warning: 'void stan::model::model_base_crtp<M>::write_ar
##   205 |   void write_array(stan::rng_t& rng, std::vector<double>& theta,

## C:/Users/andreaa/AppData/Local/Temp/RtmpCsgImm/model-31c048f84df7.hpp:301: note:   by 'stan_model_mod
##   301 |   write_array(RNG& base_rng, std::vector<double>& params_r, std::vector<int>&
## stan/src/stan/model/model_base_crtp.hpp:136: warning: 'void stan::model::model_base_crtp<M>::write_ar
##   136 |   void write_array(stan::rng_t& rng, Eigen::VectorXd& theta,
## C:/Users/andreaa/AppData/Local/Temp/RtmpCsgImm/model-31c048f84df7.hpp:301: note:   by 'stan_model_mod
##   301 |   write_array(RNG& base_rng, std::vector<double>& params_r, std::vector<int>&
```

There are three blocks which are empty in the `Stan` model above:

- The `transformed data` block, in which you can specify any transformation of the input data.
- The `transformed parameters` block which enables the definition of intermediate parameters when the likelihood requires to be specified in a different ways.
- The `generated quantities` block, from which you can make predictions based on the resulting posterior.

Formatting input data and initial values for `Stan` model

Once the model is specified, you need to format the input data to be used. `Stan` accepts data in a `list` format as follows

```
data.inp <- list(k = sum(girl[1:20]),
                 N = 20)
```

To obtain easily convergence of the model, you can also define initial values. In our case, it is sensible to include 0.5 as an initial value of our parameter θ.

```r
inits00 <- list(theta=0.5)
```

Model estimation

```r
fit.bin <- mod0$sample(
  data = data.inp,
  chains = 2,
  parallel_chains = 2,
  iter_warmup = 500,
  iter_sampling = 1000,
  seed = 26,
  init = list(inits00, inits00)  # A list of initial values for each chain
)
```

```
## Running MCMC with 2 parallel chains...
##
## Chain 1 Iteration:    1 / 1500 [  0%]  (Warmup)
## Chain 1 Iteration:  100 / 1500 [  6%]  (Warmup)
## Chain 1 Iteration:  200 / 1500 [ 13%]  (Warmup)
## Chain 1 Iteration:  300 / 1500 [ 20%]  (Warmup)
## Chain 1 Iteration:  400 / 1500 [ 26%]  (Warmup)
## Chain 1 Iteration:  500 / 1500 [ 33%]  (Warmup)
## Chain 1 Iteration:  501 / 1500 [ 33%]  (Sampling)
## Chain 1 Iteration:  600 / 1500 [ 40%]  (Sampling)
## Chain 1 Iteration:  700 / 1500 [ 46%]  (Sampling)
## Chain 1 Iteration:  800 / 1500 [ 53%]  (Sampling)
## Chain 1 Iteration:  900 / 1500 [ 60%]  (Sampling)
## Chain 1 Iteration: 1000 / 1500 [ 66%]  (Sampling)
## Chain 1 Iteration: 1100 / 1500 [ 73%]  (Sampling)
## Chain 1 Iteration: 1200 / 1500 [ 80%]  (Sampling)
## Chain 1 Iteration: 1300 / 1500 [ 86%]  (Sampling)
## Chain 1 Iteration: 1400 / 1500 [ 93%]  (Sampling)
## Chain 1 Iteration: 1500 / 1500 [100%]  (Sampling)
## Chain 2 Iteration:    1 / 1500 [  0%]  (Warmup)
## Chain 2 Iteration:  100 / 1500 [  6%]  (Warmup)
## Chain 2 Iteration:  200 / 1500 [ 13%]  (Warmup)
## Chain 2 Iteration:  300 / 1500 [ 20%]  (Warmup)
## Chain 2 Iteration:  400 / 1500 [ 26%]  (Warmup)
## Chain 2 Iteration:  500 / 1500 [ 33%]  (Warmup)
## Chain 2 Iteration:  501 / 1500 [ 33%]  (Sampling)
## Chain 2 Iteration:  600 / 1500 [ 40%]  (Sampling)
## Chain 2 Iteration:  700 / 1500 [ 46%]  (Sampling)
## Chain 2 Iteration:  800 / 1500 [ 53%]  (Sampling)
## Chain 2 Iteration:  900 / 1500 [ 60%]  (Sampling)
## Chain 2 Iteration: 1000 / 1500 [ 66%]  (Sampling)
## Chain 2 Iteration: 1100 / 1500 [ 73%]  (Sampling)
## Chain 2 Iteration: 1200 / 1500 [ 80%]  (Sampling)
## Chain 2 Iteration: 1300 / 1500 [ 86%]  (Sampling)
## Chain 2 Iteration: 1400 / 1500 [ 93%]  (Sampling)
## Chain 2 Iteration: 1500 / 1500 [100%]  (Sampling)
## Chain 1 finished in 0.0 seconds.
```

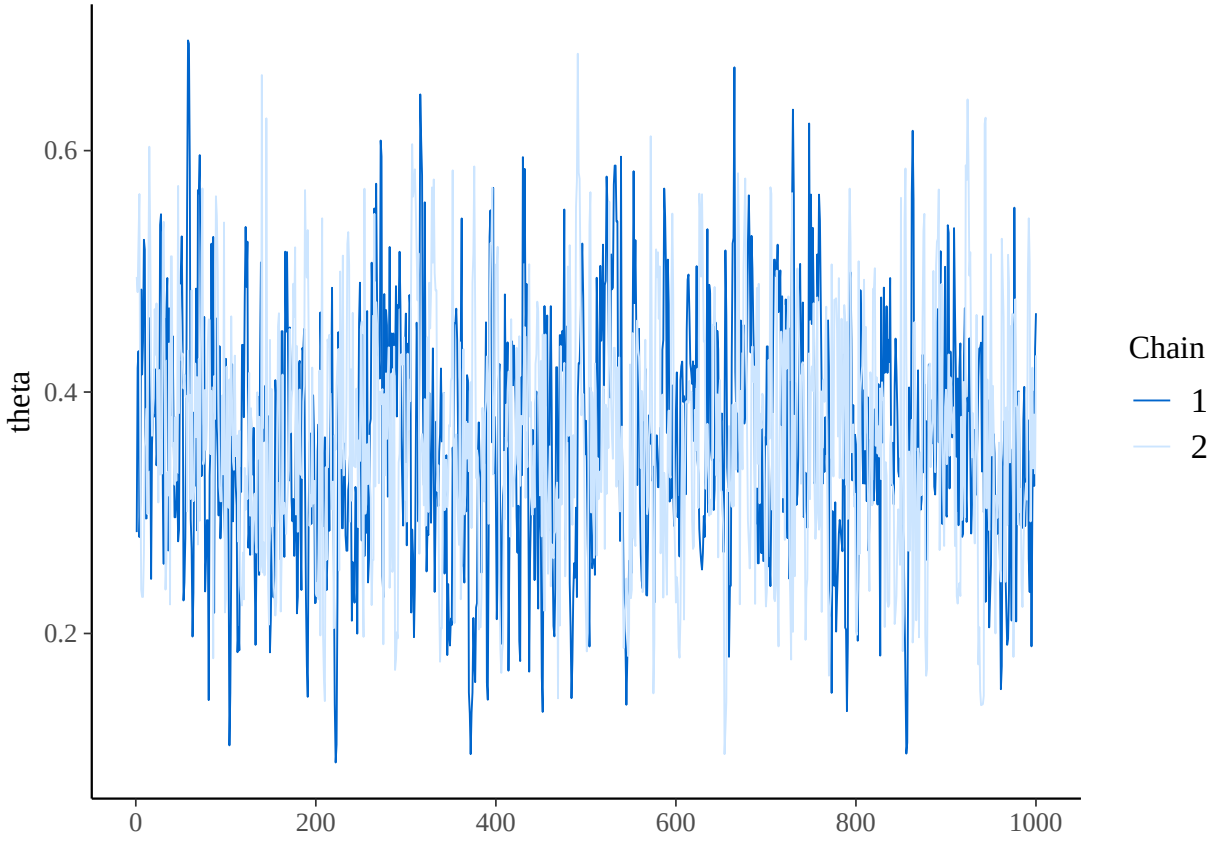
```
## Chain 2 finished in 0.0 seconds.
##
## Both chains finished successfully.
## Mean chain execution time: 0.0 seconds.
## Total execution time: 0.6 seconds.
```

Basic diagnostics

Before looking at the posterior distribution, you need to perform basic diagnostics.

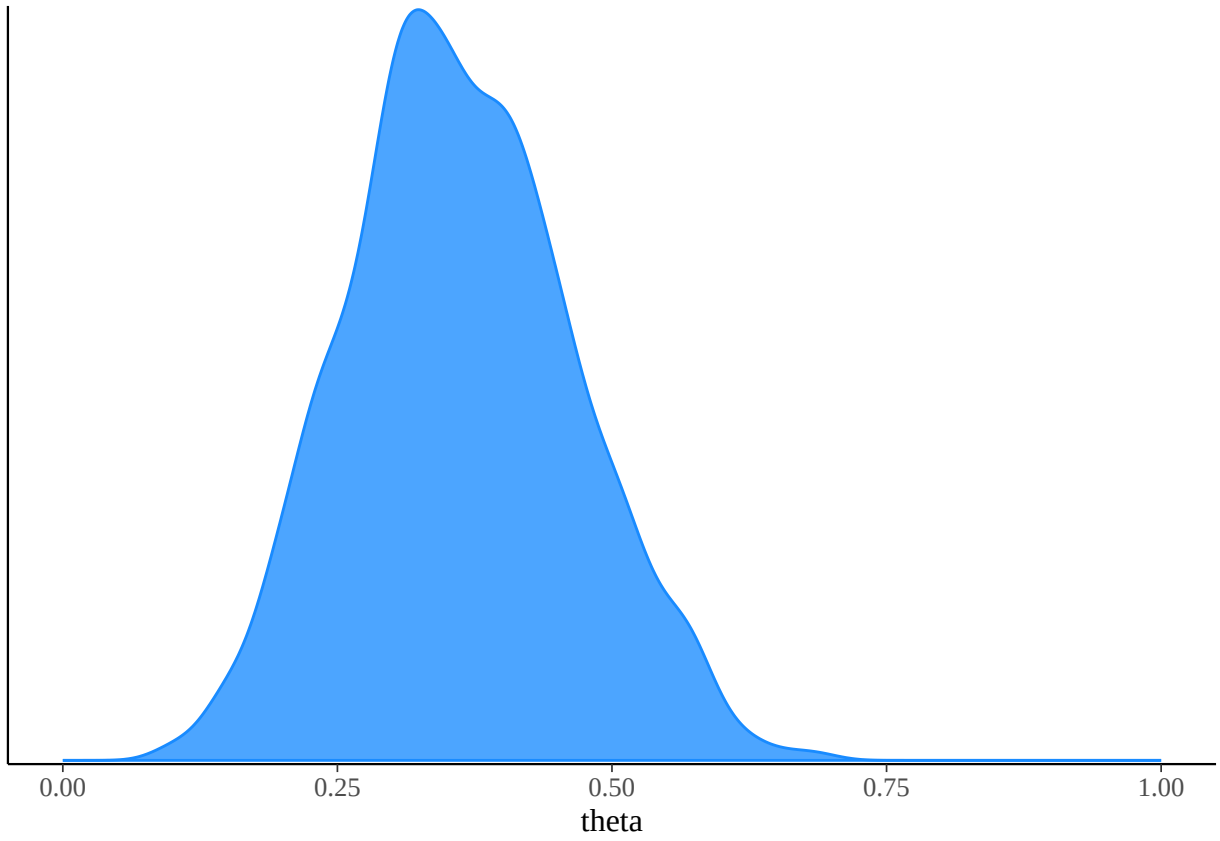
```
# Extract the posterior draws for the parameter `theta`
posterior_draws <- fit.bin$draws(variables = "theta")

# Trace plot for `theta`
mcmc_trace(posterior_draws, pars = "theta", inc_warmup = FALSE)
```



Now, look at the posterior.

```
# Density plot for `theta` with limits on the x-axis
mcmc_dens(posterior_draws, pars = "theta") +
  scale_x_continuous(limits = c(0, 1))
```


0.00
0.25
0.50
0.75
1.00
theta