Resolución

1.1.

Distribución beta(2,8):

$$E(\pi \mid y) = \frac{\alpha + y}{\alpha + \beta + n} = \frac{2 + 18}{2 + 8 + 50} = 0,33$$

$$\text{Var}(\pi \mid y) = \frac{(\alpha + y)(\beta + n - y)}{(\alpha + \beta + n)^2(\alpha + \beta + n + 1)} = \frac{(2 + 18)(8 + 50 - 18)}{(2 + 8 + 50)^2(2 + 8 + 50 + 1)} = 3,64 \cdot 10^{-3}$$

$$\alpha_{post} = 20$$
$$\beta_{post} = 40$$

```
> cat(sprintf("Li = %6.3f Ls = %6.3f", qbeta(0.025, 20,40), qbeta(0.975, 20,40)))
Li =  0.221 Ls =  0.456
```

Distribución beta(16,64):

$$E(\pi \mid y) = \frac{16 + 18}{16 + 64 + 50} = 0,26$$

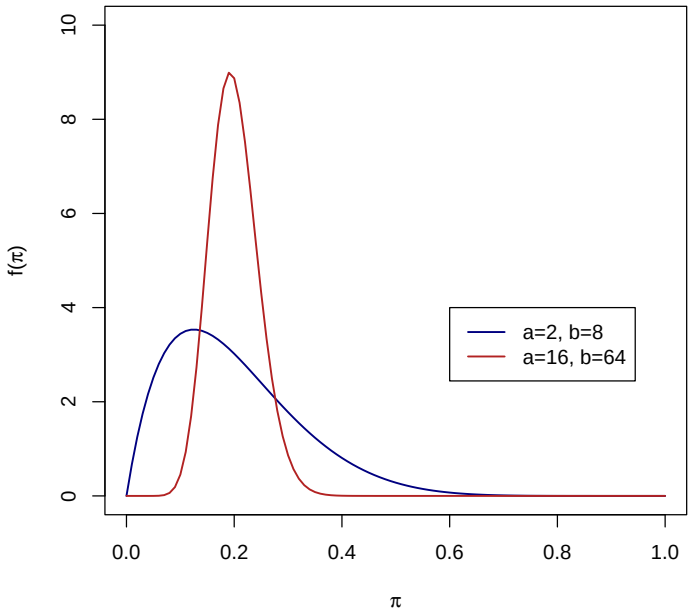
$$\text{Var}(\pi \mid y) = \frac{(\alpha + y)(\beta + n - y)}{(\alpha + \beta + n)^2(\alpha + \beta + n + 1)} = \frac{(16 + 18)(64 + 50 - 18)}{(16 + 64 + 50)^2(16 + 64 + 50 + 1)} = 1,47 \cdot 10^{-3}$$

$$\alpha_{post} = 34$$
$$\beta_{post} = 96$$

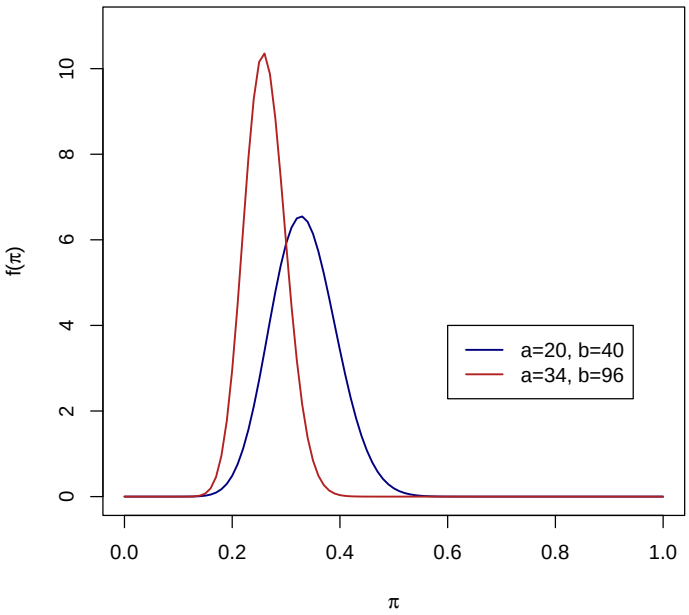
```
> cat(sprintf("Li = %6.3f Ls = %6.3f", qbeta(0.025, 34,96), qbeta(0.975, 34,96)))
Li =  0.190 Ls =  0.340
```

1.2.

```
> ## Priori
> pr <- seq(0, 1, by=0.01)
> f1 <- dbeta(pr, 2, 8)
> f2 <- dbeta(pr, 16, 64)
> plot(pr, f1, type="l", lwd=1.5,
+ xlab=expression(pi),
+ ylab=expression(paste("f(",pi,")")),
+ ylim=c(0, 10), col="navyblue")
> lines(pr, f2, type="l", lwd=1.5, col="firebrick")
> legend(0.6, 4, c("a=2, b=8", "a=16, b=64"), col=c("navyblue", "firebrick"), lwd=1.5)
```




```
> ## Posteriori
> pr <- seq(0, 1, by=0.01)
> f1 <- dbeta(pr, 20, 40)
> f2 <- dbeta(pr, 34, 96)
> plot(pr, f1, type="l", lwd=1.5,
+ xlab=expression(pi),
+ ylab=expression(paste("f(",pi,")")),
+ ylim=c(0, 11), col="navyblue")
> lines(pr, f2, type="l", lwd=1.5, col="firebrick")
> legend(0.6, 4, c("a=20, b=40", "a=34, b=96"), col=c("navyblue", "firebrick"), lwd=1.5)
```




1.3.

Los investigadores deberían rechazar la hipótesis nula, dado que, ambas distribuciones a posteriori presentan un valor esperado superior a 0,2.
Beta(16,64) es más informativa, por lo que las dispersiones son inferiores en la distribución a posteriori.

1.4.

```
> ## beta (2,8)
> library(rstan)
> set.seed(1)
> y <- 18
> n <- 50
> alpha <- 2
> beta <- 8
> datos <- list(N=n, exitos=y, a=alpha, b=beta)
> codigo <- '
+ data {
+ int<lower=0> N;
+ int exitos;
+ real a;
+ real b;
```

```
+ }
+
+ parameters {
+ real<lower=0, upper=1> pi;
+ }
+
+ model {
+ pi ~ beta(a,b);
+ exitos ~ binomial(N, pi);
+ }
+ '
> fit <- stan(model_code = codigo, data = datos)

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 1.6e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.16 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0.013 seconds (Warm-up)
Chain 1:                0.016 seconds (Sampling)
Chain 1:                0.029 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 4e-06 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.04 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
```

```
Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 2:
Chain 2:  Elapsed Time: 0.015 seconds (Warm-up)
Chain 2:                0.012 seconds (Sampling)
Chain 2:                0.027 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 3e-06 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.03 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 3:
Chain 3:  Elapsed Time: 0.011 seconds (Warm-up)
Chain 3:                0.01 seconds (Sampling)
Chain 3:                0.021 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 5e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.05 seconds.
```

```
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 4:
Chain 4:  Elapsed Time: 0.012 seconds (Warm-up)
Chain 4:                0.013 seconds (Sampling)
Chain 4:                0.025 seconds (Total)
Chain 4:
```

```
> print(fit)
```

```
Inference for Stan model: anon_model.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

       mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
pi     0.33    0.00 0.06   0.22   0.29   0.33   0.37   0.46  1465    1
lp__ -38.69    0.02 0.71 -40.67 -38.85 -38.41 -38.24 -38.19  1528    1

Samples were drawn using NUTS(diag_e) at Mon Mar 20 17:09:22 2023.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

```
> ## beta(16,64)
> set.seed(1)
> y <- 18
> n <- 50
> alpha <- 16
> beta <- 64
> datos <- list(N=n, exitos=y, a=alpha, b=beta)
> codigo <- '
+ data {
+ int<lower=0> N;
+ int exitos;
```

```
+ real a;
+ real b;
+ }
+
+ parameters {
+ real<lower=0, upper=1> pi;
+ }
+
+ model {
+ pi ~ beta(a,b);
+ exitos ~ binomial(N, pi);
+ }
+ '
> fit <- stan(model_code = codigo, data = datos)

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 5e-06 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.05 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0.012 seconds (Warm-up)
Chain 1:                0.013 seconds (Sampling)
Chain 1:                0.025 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 1.7e-05 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.17 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
```

```
Chain 2:
Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 2:
Chain 2:  Elapsed Time: 0.012 seconds (Warm-up)
Chain 2:                0.013 seconds (Sampling)
Chain 2:                0.025 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 4e-06 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.04 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 3:
Chain 3:  Elapsed Time: 0.013 seconds (Warm-up)
Chain 3:                0.01 seconds (Sampling)
Chain 3:                0.023 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
```

```
Chain 4: Gradient evaluation took 4e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.04 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
Chain 4:
Chain 4:  Elapsed Time: 0.017 seconds (Warm-up)
Chain 4:                0.012 seconds (Sampling)
Chain 4:                0.029 seconds (Total)
Chain 4:
```

```
> print(fit)
```

```
Inference for Stan model: anon_model.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

       mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
pi     0.26    0.00 0.04   0.19   0.24   0.26   0.29   0.34  1588    1
lp__ -75.21    0.02 0.72 -77.31 -75.36 -74.93 -74.76 -74.71  1892    1

Samples were drawn using NUTS(diag_e) at Mon Mar 20 17:09:23 2023.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

2.1.

```
> b <- c(-1.20, -0.65, 0.20, 0.80, 1.15 )
> muestras <- 1000
> theta <- rnorm(muestras)
> ## Prob de éxito de cada item en cada muestra
> p_mat <- matrix(nrow = 1000,ncol =  5)
> for (i in 1:length(b)) {
```

```
+         for (j in 1:muestras) {
+                 p_mat[j,i] <- exp(theta[j] - b[i]) / (1 + exp(theta[j] - b[i]))
+         }
+ }
> ## Promedio de las probs de cada item
> Pr_v <- apply(p_mat, mean, MARGIN = 2)
> ## Calculamos la intersección
> P_3prim = Pr_v[1] * Pr_v[2] * Pr_v[3] * (1-Pr_v[4]) * (1-Pr_v[5])
> print (P_3prim)

[1] 0.1012611
```

2.2.

```
> library(rstan)
> set.seed(1)
> x <- c(1, 1, 1, 0, 0)
> b <- c(-1.20, -0.65, 0.20, 0.80, 1.15)
> datos <- list(n_items = length(x), x = x, b = b)
> codigo <- "
+ data {
+   int  n_items;
+   int x [n_items];
+   real b[n_items];
+ }
+
+ parameters {
+   real theta;
+ }
+
+ model {
+   real p;
+   theta ~ normal(0, 1);
+   for (i in 1:n_items) {
+     p = exp(theta - b[i]) / (1 + exp(theta - b[i]));
+     x[i] ~ bernoulli(p);
+   }
+ }
+ "
> fit <- stan(model_code = codigo, data = datos, iter = 1000)

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 2.9e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.29 seconds.
Chain 1: Adjust your expectations accordingly!
```

```
Chain 1:
Chain 1:
Chain 1: Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 1: Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 1: Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 1: Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 1: Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 1: Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 1: Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 1: Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 1: Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 1: Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 1: Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 1: Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0.011 seconds (Warm-up)
Chain 1:                0.01 seconds (Sampling)
Chain 1:                0.021 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 5e-06 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.05 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 2: Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 2: Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 2: Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 2: Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 2: Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 2: Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 2: Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 2: Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 2: Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 2: Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 2: Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 2:
Chain 2:  Elapsed Time: 0.012 seconds (Warm-up)
Chain 2:                0.009 seconds (Sampling)
Chain 2:                0.021 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
```

```
Chain 3:
Chain 3: Gradient evaluation took 2e-05 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 3: Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 3: Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 3: Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 3: Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 3: Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 3: Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 3: Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 3: Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 3: Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 3: Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 3: Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 3:
Chain 3:  Elapsed Time: 0.012 seconds (Warm-up)
Chain 3:                0.009 seconds (Sampling)
Chain 3:                0.021 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 5e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.05 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 4: Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 4: Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 4: Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 4: Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 4: Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 4: Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 4: Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 4: Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 4: Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 4: Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 4: Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 4:
Chain 4:  Elapsed Time: 0.008 seconds (Warm-up)
Chain 4:                0.01 seconds (Sampling)
```

```
Chain 4:                0.018 seconds (Total)
Chain 4:
```

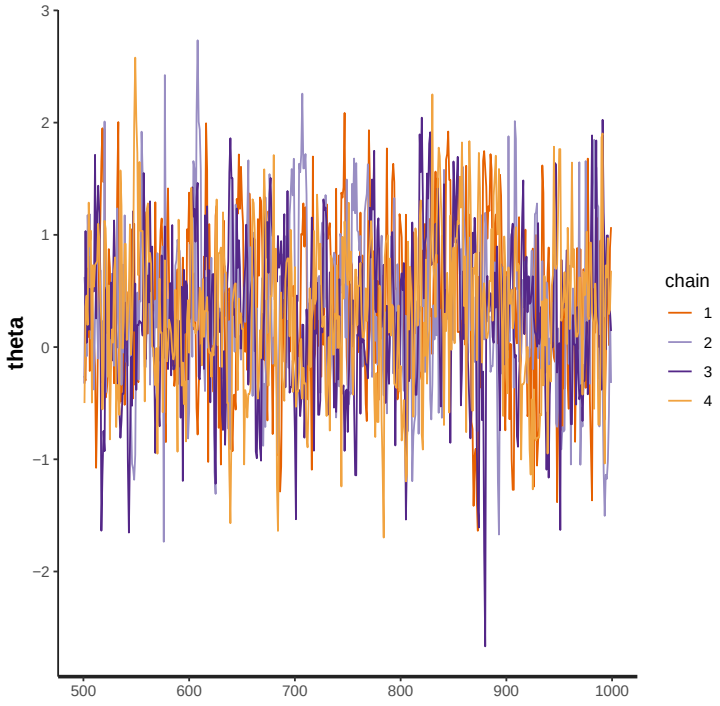
```
> print(fit)
```

```
Inference for Stan model: anon_model.
4 chains, each with iter=1000; warmup=500; thin=1;
post-warmup draws per chain=500, total post-warmup draws=2000.

       mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
theta  0.31    0.03 0.71 -1.07 -0.15  0.30  0.76  1.71   554    1
lp__  -2.55    0.02 0.71 -4.50 -2.71 -2.26 -2.10 -2.05   979    1

Samples were drawn using NUTS(diag_e) at Mon Mar 20 17:10:24 2023.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

2.3.

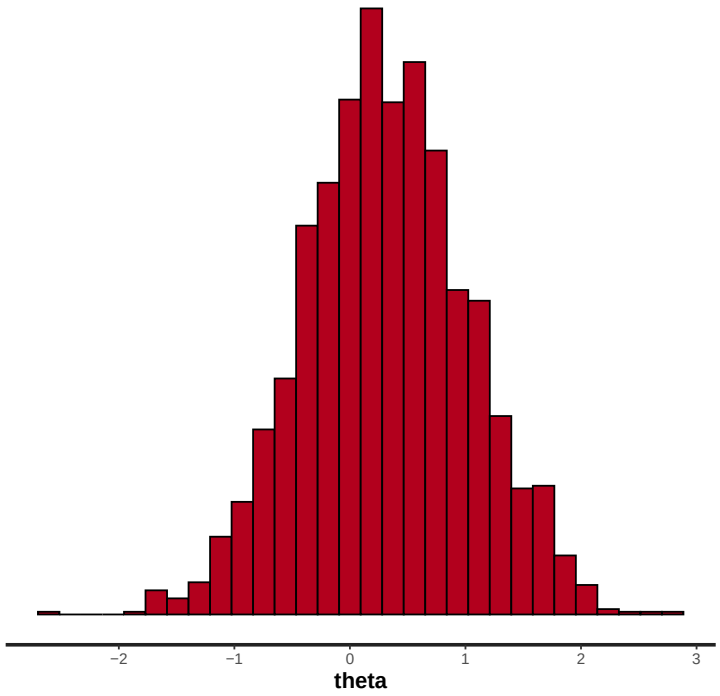
```
> library(rstan)
> library(ggplot2)
```

```
> traceplot(fit, pars="theta")
```



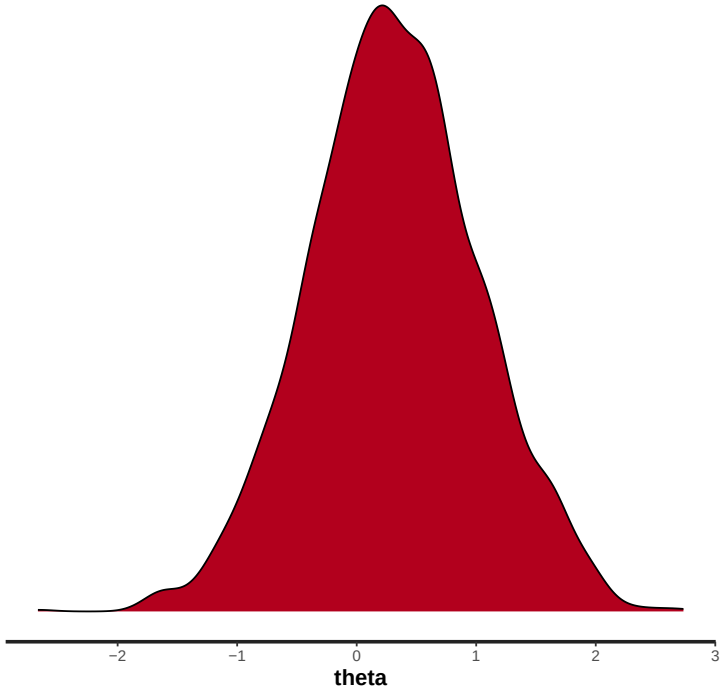
Cadenas de Markov: Podemos observar una convergencia aceptable dada la estabilidad de los valores simulados. Los valores muestreados son representativos de la distribución a posteriori.

```
> stan_hist(fit, pars="theta")
```



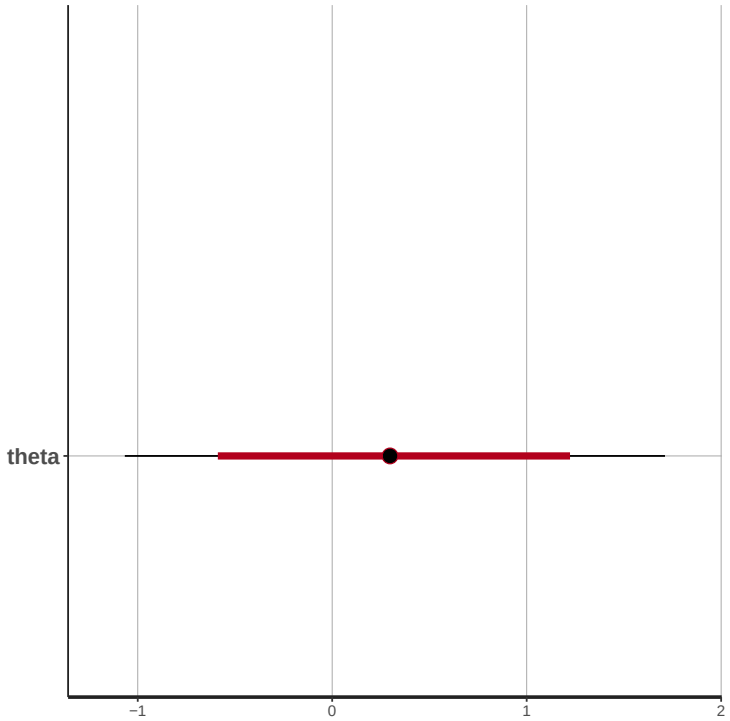

La distribución $f(\theta \mid y)$, con media aproximada de 0,31.

```
> stan_dens(fit, pars="theta")
```



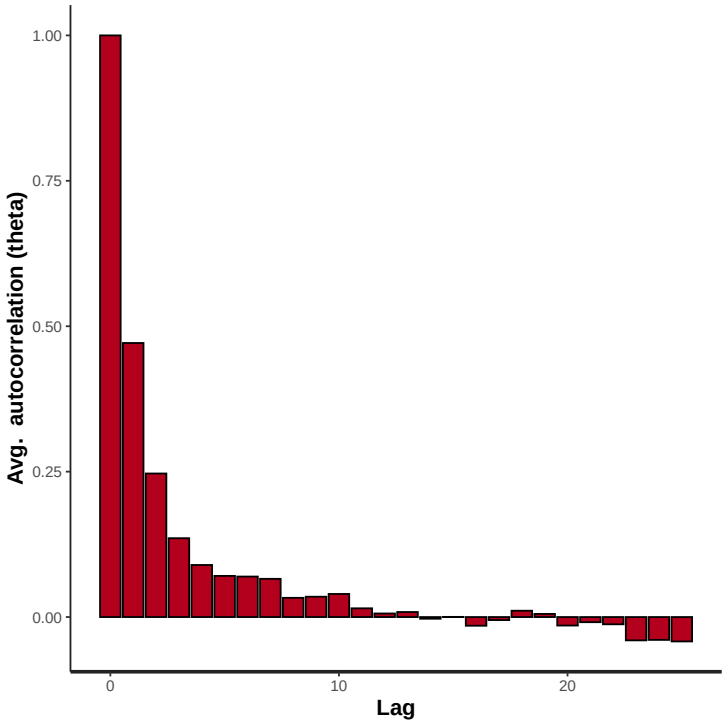

La distribución (suavizada) $f(\theta \mid y)$, con media aproximada de 0,31.

```
> plot(fit)
```




Con un 95% de confianza, el verdadero valor de $\theta$ se encuentra entre -1,07 y 1,71.

```
> stan_ac(fit, pars="theta")
```




La autocorrelación disminuye al aumentar la distancia entre valores, es decir, las observaciones de la serie están correlacionadas positivamente con las observaciones anteriores.

3.

```
> library(rstan)
> set.seed(1)
> x <- rnorm(25,3,2)
> datos <- list(N = length(x), x = x)
> codigo <- "
+ data {
+   int<lower=0> N;
+   real x[N];
+ }
+
+ parameters {
+   real mu;
+ }
+
+ model {
+   mu ~ normal(0,10);
```

```
+   x ~ normal(mu,2);
+ }
+ "
> fit <- stan(model_code = codigo, data = datos, iter = 1000)

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
Chain 1:
Chain 1: Gradient evaluation took 2e-05 seconds
Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.
Chain 1: Adjust your expectations accordingly!
Chain 1:
Chain 1:
Chain 1: Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 1: Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 1: Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 1: Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 1: Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 1: Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 1: Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 1: Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 1: Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 1: Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 1: Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 1: Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 1:
Chain 1:  Elapsed Time: 0.009 seconds (Warm-up)
Chain 1:                0.006 seconds (Sampling)
Chain 1:                0.015 seconds (Total)
Chain 1:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
Chain 2:
Chain 2: Gradient evaluation took 2e-06 seconds
Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.02 seconds.
Chain 2: Adjust your expectations accordingly!
Chain 2:
Chain 2:
Chain 2: Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 2: Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 2: Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 2: Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 2: Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 2: Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 2: Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 2: Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 2: Iteration: 700 / 1000 [ 70%]  (Sampling)
```

```
Chain 2: Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 2: Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 2: Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 2:
Chain 2:  Elapsed Time: 0.005 seconds (Warm-up)
Chain 2:                0.005 seconds (Sampling)
Chain 2:                0.01 seconds (Total)
Chain 2:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 3).
Chain 3:
Chain 3: Gradient evaluation took 4e-06 seconds
Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.04 seconds.
Chain 3: Adjust your expectations accordingly!
Chain 3:
Chain 3:
Chain 3: Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 3: Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 3: Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 3: Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 3: Iteration: 400 / 1000 [ 40%]  (Warmup)
Chain 3: Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 3: Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 3: Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 3: Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 3: Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 3: Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 3: Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 3:
Chain 3:  Elapsed Time: 0.005 seconds (Warm-up)
Chain 3:                0.007 seconds (Sampling)
Chain 3:                0.012 seconds (Total)
Chain 3:

SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 4).
Chain 4:
Chain 4: Gradient evaluation took 4e-06 seconds
Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.04 seconds.
Chain 4: Adjust your expectations accordingly!
Chain 4:
Chain 4:
Chain 4: Iteration:   1 / 1000 [  0%]  (Warmup)
Chain 4: Iteration: 100 / 1000 [ 10%]  (Warmup)
Chain 4: Iteration: 200 / 1000 [ 20%]  (Warmup)
Chain 4: Iteration: 300 / 1000 [ 30%]  (Warmup)
Chain 4: Iteration: 400 / 1000 [ 40%]  (Warmup)
```

```
Chain 4: Iteration: 500 / 1000 [ 50%]  (Warmup)
Chain 4: Iteration: 501 / 1000 [ 50%]  (Sampling)
Chain 4: Iteration: 600 / 1000 [ 60%]  (Sampling)
Chain 4: Iteration: 700 / 1000 [ 70%]  (Sampling)
Chain 4: Iteration: 800 / 1000 [ 80%]  (Sampling)
Chain 4: Iteration: 900 / 1000 [ 90%]  (Sampling)
Chain 4: Iteration: 1000 / 1000 [100%]  (Sampling)
Chain 4:
Chain 4:  Elapsed Time: 0.006 seconds (Warm-up)
Chain 4:                0.006 seconds (Sampling)
Chain 4:                0.012 seconds (Total)
Chain 4:
```

```
> print(fit)

Inference for Stan model: anon_model.
4 chains, each with iter=1000; warmup=500; thin=1;
post-warmup draws per chain=500, total post-warmup draws=2000.

       mean se_mean   sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
mu     3.31    0.01 0.41   2.52   3.03   3.31   3.59   4.12   795    1
lp__ -11.41    0.02 0.73 -13.46 -11.58 -11.14 -10.95 -10.89  1074    1

Samples were drawn using NUTS(diag_e) at Mon Mar 20 17:11:32 2023.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

```
> cat(sprintf("EAP = %6.3f, Var_post = %6.3f, Li_post = %6.3f,  Ls_post = %6.3f",
+                                                 3.31, 0.41^2,qnorm(0.04, 3.31,0.41), qnorm

EAP =  3.310, Var_post =  0.168, Li_post =  2.592,  Ls_post =  4.028
```

Con un 96% de confianza, el parámetro poblacional $\mu$, tras observar los datos muestreados, se encontrará entre los valores 2.592 y 4.028.