Resolución

1.1.

```
> library(rstan)
> set.seed(2)
> options(mc.cores = parallel::detectCores())
> rstan_options(auto_write = TRUE)
> x1 <- c(4.8, 6.2, 6.1, 5.1, 4.8, 3, 6)
> x2 <- c(4, 4.7, 4.9, 5.7, 4.2, 2.9, 5.6)
> datos <- list(n = length(x1), x1 = x1, x2 = x2)
> codigo <- "
+   data {
+     int<lower=0> n;
+     real<lower=0> x1[n];
+     real<lower=0> x2[n];
+   }
+
+   parameters {
+   real mu1;
+   real mu2;
+   }
+   model{
+   mu1 ~ normal(0,5);
+   mu2 ~ normal(0,5);
+   for(i in 1:n){
+                 x1[i] ~ normal(mu1,1);
+                 }
+         for(i in 1:n){
+                 x2[i] ~ normal(mu2,1);
+                 }
+   }
+   generated quantities{
+   real D;
+   D =  0;
+   for (i in 1:n){
+   D += normal_lpdf(x1[i] | mu1,1);
+   D += normal_lpdf(x2[i] | mu2,1);
+   }
+   D*= -2;
+   }
+   "
> fit <- stan(model_code = codigo, data = datos,iter = 1000)
> 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
mu1   5.12    0.01 0.39   4.37  4.84  5.12  5.39  5.87  1776    1
mu2   4.56    0.01 0.38   3.83  4.30  4.56  4.83  5.30  1709    1
D    41.13    0.07 2.01  39.11 39.68 40.47 41.93 46.57   734    1
lp__ -8.64    0.04 1.01 -11.33 -9.03 -8.31 -7.91 -7.63   736    1

Samples were drawn using NUTS(diag_e) at Wed Apr 19 16:33:20 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).

> D1<- unlist(extract(fit, pars='D'))
> DIC1 <- mean(D1) + 0.5 *var(D1)
> print(DIC1)

[1] 43.15062
```

1.2.

```
> x1 <- c(4.8, 6.2, 6.1, 5.1, 4.8, 3, 6)
> x2 <- c(4, 4.7, 4.9, 5.7, 4.2, 2.9, 5.6)
> datos <- list(n = length(x1), x1 = x1, x2 = x2)
> codigo <- "
+   data {
+     int<lower=0> n;
+     real<lower=0> x1[n];
+     real<lower=0> x2[n];
+   }
+
+   parameters {
+   real mu;
+   }
+   model{
+   mu ~ normal(0,5);
+   for(i in 1:n){
+                 x1[i] ~ normal(mu,1);
+                 }
+         for(i in 1:n){
+                 x2[i] ~ normal(mu,1);
+                 }
+   }
+   generated quantities{
+   real D;
+   D =  0;
+   for (i in 1:n){
+   D += normal_lpdf(x1[i] | mu,1);
+   D += normal_lpdf(x2[i] | mu,1);
+   }
+   D*= -2;
+   }
+   "
> fit <- stan(model_code = codigo, data = datos,iter=1000)
> 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    4.84    0.01 0.27   4.32  4.67  4.84  5.03  5.36   706    1
D    41.18    0.05 1.36  40.19 40.29 40.65 41.52 44.98   834    1
lp__ -8.19    0.02 0.68 -10.11 -8.39 -7.93 -7.75 -7.70   830    1

Samples were drawn using NUTS(diag_e) at Wed Apr 19 16:34:53 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).
```

```
> D2<- unlist(extract(fit, pars='D'))
> DIC2 <- mean(D2) + 0.5 *var(D2)
> DIC2 <- mean(D2) + 0.5 *var(D1)
> print(DIC2)
```

```
[1] 43.20344
```

1.3.

```
> cat(sprintf("\nModelo 1. mean D = %8.2f. pD = %8.2f. DIC = %8.2f\n
+ Modelo 2. mean D = %8.2f. pD = %8.2f. DIC = %8.2f\n",
+ mean(D1), 0.5*var(D1), DIC1, mean(D2), 0.5*var(D2), DIC2))
```

```
Modelo 1. mean D =    41.13. pD =     2.02. DIC =    43.15

Modelo 2. mean D =    41.18. pD =     0.93. DIC =    43.20
```

El modelo 2 presenta peor ajuste, pero es más sencillo. Sin embargo, su IC posterior para mu no incluye el 0 por lo que con un 95% de confianza no podemos asumir que la diferencia entre ambos grupos sea nula. Elijo por lo tanto el modelo 1.
Conclusión: El rendimiento en orientación espacial es mayor en hombres que en mujeres.

2.1.

```
> y <- c(93, 66, 42, 45, 38, 32, 36, 43, 40, 53)
> x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
> datos <- list(n = length(x), x = x, y = y)
> codigo <- "
+   data {
+     int<lower=0> n;
+     real<lower=0> x[n];
+     real<lower=0> y[n];
+   }
+
+   parameters {
+   real phi;
+   real tau;
+   real omega;
+   }
+   model{
+   real mu;
+   real alpha;
+   real beta;
+
+
+   tau ~ normal(0,5);
+   omega ~ normal(0,5);
+         for(i in 1:n){
+                 mu = exp(tau+omega*x[i]);
+                 alpha = mu*phi;
+                 beta = phi;
+                 y[i] ~ gamma(alpha,beta);
+         }
+   }
+   generated quantities{
+   real D;
+   D =  0;
+   for (i in 1:n){
+   D += gamma_lpdf(y[i] | exp(tau+omega*x[i])*phi,phi);
+   }
+   D*= -2;
+   }
+   "
> fit <- stan(model_code = codigo, data = datos, iter = 2000)
> 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
phi     0.31    0.00 0.14   0.10   0.21   0.29   0.40   0.62   925    1
tau     4.24    0.01 0.19   3.86   4.12   4.24   4.36   4.61  1105    1
omega  -0.07    0.00 0.03  -0.14  -0.09  -0.07  -0.05   0.00  1069    1
D      81.53    0.07 2.40  78.72  79.78  80.92  82.66  87.69  1347    1
lp__  -41.12    0.03 1.20 -44.21 -41.69 -40.83 -40.25 -39.72  1342    1

Samples were drawn using NUTS(diag_e) at Wed Apr 19 16:36:36 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).
```

```
> D1<- unlist(extract(fit, pars='D'))
> DIC1 <- mean(D1) + 0.5 *var(D1)
> print(DIC1)
```

```
[1] 84.39532
```

2.2.

```
> y <- c(93, 66, 42, 45, 38, 32, 36, 43, 40, 53)
> x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
> datos <- list(n = length(x), x = x, y = y)
> codigo <- "
+   data {
+     int<lower=0> n;
+     real<lower=0> x[n];
+     real<lower=0> y[n];
+   }
+
+   parameters {
+   real phi;
+   real tau;
+   real omega1;
+   real omega2;
+   }
+   model{
+   real mu;
+   real alpha;
+   real beta;
+
+
+   tau ~ normal(0,5);
+   omega1 ~ normal(0,5);
+   omega2 ~ normal(0,5);
+         for(i in 1:n){
+                 mu = exp(tau+omega1*x[i]+omega2*(x[i])^2);
+                 alpha = mu*phi;
+                 beta = phi;
+                 y[i] ~ gamma(alpha,beta);
+         }
+   }
+   generated quantities{
+   real D;
+   D =  0;
+   for (i in 1:n){
+   D += gamma_lpdf(y[i] | exp(tau+omega1*x[i]+omega2*(x[i])^2)*phi,phi);
+   }
+   D*= -2;
+   }
+   "
> fit <- stan(model_code = codigo, data = datos, iter = 2000)
> 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
phi      2.69    0.04 1.25   0.82   1.77   2.46   3.45   5.57   840 1.00
tau      4.89    0.00 0.09   4.71   4.84   4.89   4.95   5.07   995 1.00
omega1  -0.43    0.00 0.04  -0.51  -0.45  -0.43  -0.40  -0.34   909 1.00
omega2   0.03    0.00 0.00   0.03   0.03   0.03   0.04   0.04   976 1.00
D       59.49    0.09 2.91  55.86  57.36  58.84  60.90  67.22  1071 1.01
lp__   -30.23    0.04 1.45 -34.09 -30.93 -29.91 -29.16 -28.42  1071 1.01

Samples were drawn using NUTS(diag_e) at Wed Apr 19 16:38:14 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).
```

```
> D2<- unlist(extract(fit, pars='D'))
> DIC2 <- mean(D1) + 0.5 *var(D2)
> print(DIC2)
```

```
[1] 85.75326
```

2.3.

```
> y <- c(93, 66, 42, 45, 38, 32, 36, 43, 40, 53)
> x <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
> datos1 <- list(n = length(x), x = x, y = y, modelo=1)
> datos2 <- list(n = length(x), x = x, y = y, modelo=2)
> codigo <- "
+   data {
+     int<lower=0> n;
+     real<lower=0> x[n];
+     real<lower=0> y[n];
+     int modelo;
+   }
+
+   parameters {
+   real phi;
+   real tau;
+   real omega1;
+   real omega2;
+   }
+   model{
+   real mu;
+   real alpha;
+   real beta;
+
+
+   tau ~ normal(0,5);
+   omega1 ~ normal(0,5);
+   omega2 ~ normal(0,5);
+
+         for(i in 1:n){
+                 if(modelo==1){
+                 mu = exp(tau+omega1*x[i]);
+                 }else{
+                 mu = exp(tau+omega1*x[i]+omega2*(x[i])^2);
+                 }
+                 alpha = mu*phi;
+                 beta = phi;
+                 y[i] ~ gamma(alpha,beta);
+         }
+   }
+   generated quantities{
+   real D;
+   real alpha_D;
+   real beta_D;
+   beta_D = phi;
+   real yPred[n];
+   D =  0;
+   for (i in 1:n){
+           if(modelo == 1){
+           alpha_D = exp(tau+omega1*x[i])*phi;
+           } else {
+           alpha_D = exp(tau+omega1*x[i]+omega2*(x[i])^2)*phi;;
+           }
+   yPred[i] = gamma_rng(alpha_D, beta_D);
+   D += gamma_lpdf(y[i] | alpha_D,beta_D);
+   }
+   D*= -2;
+   }
+   "
> fit1 <- stan(model_code = codigo, data = datos1, iter = 2000)
> fit2 <- stan(model_code = codigo, data = datos2, iter = 2000)
> D1 <- unlist(extract(fit1, pars='D'))
> D2 <- unlist(extract(fit2, pars='D'))
> DIC1 <- mean(D1) + 0.5 * var(D1)
> DIC2 <- mean(D2) + 0.5 * var(D2)
> cat(sprintf("\nModelo 1. mean D = %8.2f. pD = %8.2f. DIC = %8.2f\n
+ Modelo 2. mean D = %8.2f. pD = %8.2f. DIC = %8.2f\n",
+                                                     mean(D1), 0.5*var(D1), DIC1, mean(D2), 0.5*var(D2), DIC2))
```

```
Modelo 1. mean D =    81.40. pD =     2.81. DIC =    84.21

Modelo 2. mean D =    59.52. pD =     4.35. DIC =    63.87
```

```
> print(fit1)
```

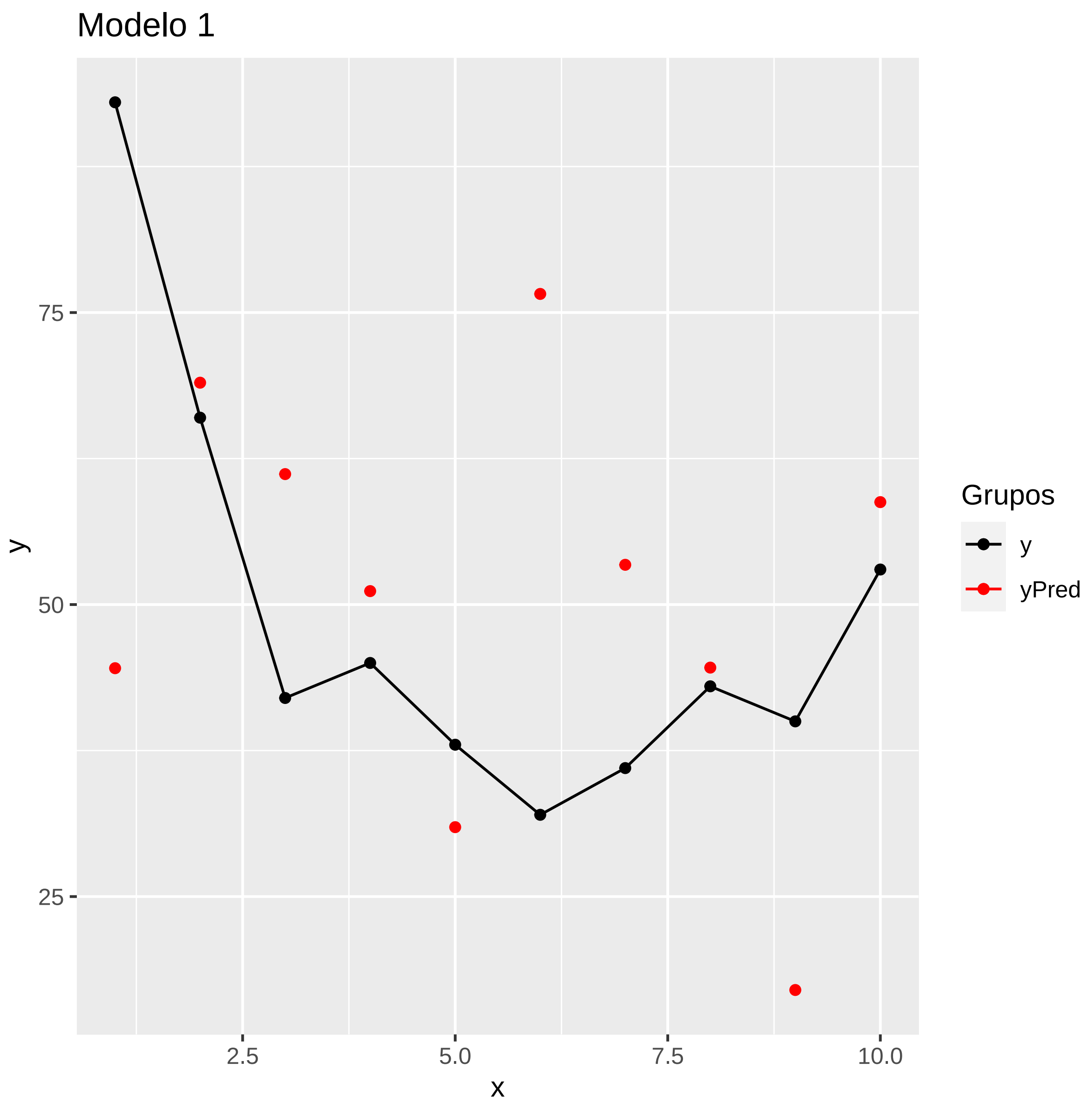
```
Inference for Stan model: anon_model.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

            mean se_mean    sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
phi         0.31    0.00  0.13   0.10   0.21   0.29   0.38   0.63  1354    1
tau         4.23    0.00  0.19   3.84   4.12   4.24   4.35   4.59  1848    1
omega1     -0.07    0.00  0.03  -0.13  -0.09  -0.07  -0.05   0.00  1872    1
omega2     -0.03    0.10  5.07  -9.88  -3.36  -0.05   3.36  10.25  2712    1
D          81.40    0.06  2.37  78.70  79.63  80.77  82.49  87.80  1689    1
alpha_D    10.83    0.13  4.85   3.49   7.30  10.15  13.66  22.44  1497    1
beta_D      0.31    0.00  0.13   0.10   0.21   0.29   0.38   0.63  1354    1
yPred[1]   65.61    0.35 19.45  32.23  52.71  63.86  76.21 108.00  3032    1
yPred[2]   60.91    0.30 17.91  30.90  48.86  59.09  70.80 100.03  3499    1
yPred[3]   56.32    0.26 15.85  27.99  45.86  55.02  65.14  91.51  3683    1
yPred[4]   52.92    0.25 15.89  26.08  42.27  51.56  61.50  88.89  3959    1
yPred[5]   49.48    0.24 15.21  24.53  39.37  48.12  57.77  82.09  3964    1
yPred[6]   46.08    0.22 14.07  22.26  36.45  44.71  54.14  77.20  4059    1
yPred[7]   43.19    0.23 14.26  19.50  33.61  41.69  51.00  76.08  3970    1
yPred[8]   40.40    0.23 13.71  17.56  31.07  38.81  48.09  71.71  3540    1
yPred[9]   38.05    0.24 14.27  15.35  28.37  36.40  45.82  69.79  3579    1
yPred[10]  35.50    0.26 13.99  12.93  26.24  34.00  43.14  67.83  2924    1
lp__      -41.57    0.03  1.39 -45.14 -42.30 -41.24 -40.54 -39.86  1672    1

Samples were drawn using NUTS(diag_e) at Wed Apr 19 16:39:53 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).
```

```
> print(fit2)
```

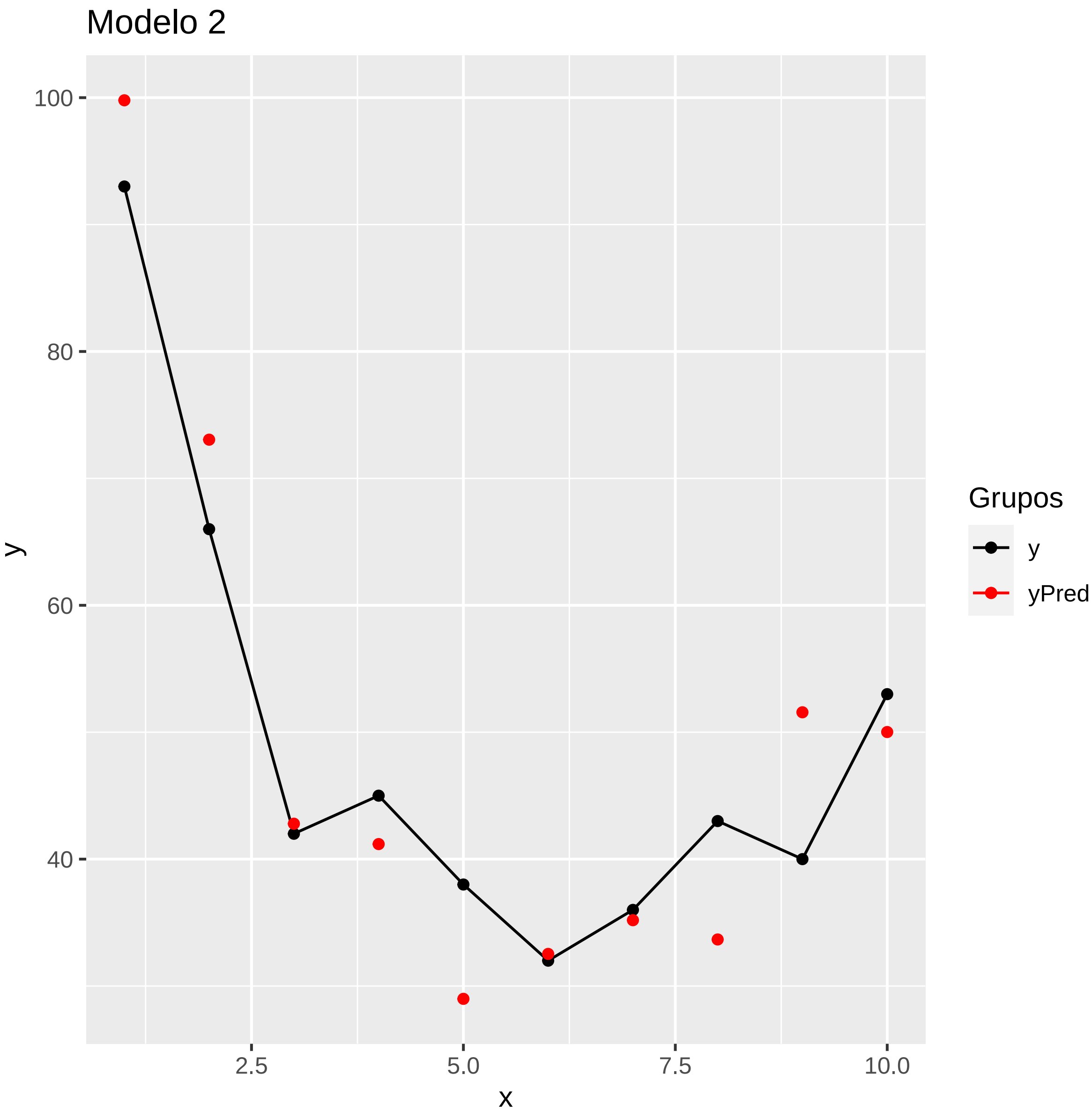
```
Inference for Stan model: anon_model.
4 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=4000.

            mean se_mean    sd   2.5%    25%    50%    75%  97.5% n_eff Rhat
phi         2.88    0.08  1.37   0.89   1.86   2.68   3.68   6.22   281 1.01
tau         4.90    0.00  0.10   4.70   4.84   4.90   4.96   5.09  1004 1.00
omega1     -0.43    0.00  0.05  -0.52  -0.46  -0.43  -0.40  -0.33   972 1.00
omega2      0.03    0.00  0.00   0.02   0.03   0.03   0.04   0.04  1034 1.00
D          59.52    0.09  2.95  55.80  57.32  58.88  60.99  66.79  1044 1.00
alpha_D   154.75    4.42 74.20  46.34  99.31 142.88 197.93 329.07   282 1.01
beta_D      2.88    0.08  1.37   0.89   1.86   2.68   3.68   6.22   281 1.01
yPred[1]   90.71    0.20  8.45  74.27  85.28  90.50  96.00 107.84  1701 1.00
yPred[2]   65.15    0.10  5.84  53.89  61.46  64.90  68.63  77.10  3526 1.00
yPred[3]   50.16    0.09  5.23  40.44  46.68  49.97  53.39  60.82  3706 1.00
yPred[4]   41.57    0.08  4.70  32.56  38.44  41.47  44.39  51.51  3139 1.00
yPred[5]   36.58    0.07  4.40  28.44  33.69  36.42  39.18  45.61  3510 1.00
yPred[6]   34.58    0.08  4.22  26.97  31.83  34.44  37.16  42.92  3148 1.00
yPred[7]   34.90    0.08  4.32  26.88  32.03  34.75  37.44  44.13  3126 1.00
yPred[8]   37.73    0.07  4.45  29.51  34.87  37.61  40.48  46.97  3842 1.00
yPred[9]   43.37    0.08  4.88  33.86  40.18  43.29  46.38  53.16  4101 1.00
yPred[10]  53.59    0.11  6.11  41.98  49.61  53.38  57.39  66.11  3171 1.00
lp__      -30.24    0.05  1.47 -33.92 -30.98 -29.93 -29.14 -28.38  1044 1.00

Samples were drawn using NUTS(diag_e) at Wed Apr 19 16:40:12 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).
```

2.4.

```
> library(ggplot2)
> yPred_1 <- extract(fit1,"yPred")
> ## Empleo los últimos 10 valores de la última iteración
> yPred_1 <- yPred_1[[1]]
> yPred_1 <- yPred_1[4000,1:10]
> yPred_2 <- extract(fit2,"yPred")
> yPred_2 <- yPred_2[[1]]
> yPred_2 <- yPred_2[4000,1:10]
> data_1 <- as.data.frame( cbind(x,y,yPred_1))
> data_2 <- as.data.frame(cbind(x,y,yPred_2))
> ggplot(data_1, aes(x, y)) +
+         geom_point(aes(color = "y")) +
+         geom_line(aes(color = "y")) +
+         geom_point(aes(x, yPred_1, color = "yPred")) +
+         scale_color_manual(name = "Grupos",
+                 values = c("y" = "black", "yPred" = "red")) +
+         labs(color = "Leyenda") +
+         ggtitle("Modelo 1")
```




```
> ggplot(data_2, aes(x, y)) +
+         geom_point(aes(color = "y")) +
+         geom_line(aes(color = "y")) +
+         geom_point(aes(x, yPred_2, color = "yPred")) +
+         scale_color_manual(name = "Grupos",
+                 values = c("y" = "black", "yPred" = "red")) +
+         labs(color = "Leyenda") +
+         ggtitle("Modelo 2")
>
```



2.5.
El modelo 1 es más sencillo, pero presenta un ajuste considerablemente menor.
Elijo el modelo 2 pues.