EstComp-Tarea13

Bruno Gonzalez

2/12/2019

13-Modelos jerárquicos

En este ejercicio definirás un modelo jerárquico para la incidencia de tumores en grupos de conejos a los que se suministró una medicina. Se realizaron 71 experimentos distintos utilizando la misma medicina.

Considerando que cada conejo proviene de un experimento distinto, se desea estudiar θ_j, la probabilidad de desarrollar un tumor en el j-ésimo grupo, este parámetro variará de grupo a grupo.

Denotaremos y_{ij} la observación en el i-ésimo conejo perteneciente al j-ésimo experimento, y_{ij} puede tomar dos valores: 1 indicando que el conejo desarrolló tumor y 0 en el caso contrario, por tanto la verosimilitud sería:

$$y_{ij} \sim Bernoulli(\theta_j)$$

Adicionalmente se desea estimar el efecto medio de la medicina a lo largo de los grupos μ, por lo que utilizaremos un modelo jerárquico como sigue:

$$\theta_j \sim Beta(a, b)$$

donde

$$a = \mu\kappa$$
$$b = (1 - \mu)\kappa$$

Finalmente asignamos una distribución a los hiperparámetros μ y κ,

$$\mu \sim Beta(A_\mu, B_\mu)$$

$$\kappa \sim Gamma(S_\kappa, R_\kappa)$$

1. Si piensas en este problema como un lanzamiento de monedas, ¿a qué corresponden las monedas y los lanzamientos?

Cada lanzamiento sería equivalente a que cada conejo tenga o no el tumor y cada experimento es una moneda distinta.

2. Los datos en el archivo contienen las observaciones de los 71 experimentos, cada renglón corresponde a una observación.

- Utiliza Stan para ajustar un modelo jerárquico como el descrito arriba y usando una inicial $Beta(1, 1)$ y una $Gamma(1, 0.1)$ para μ y κ respectivamente. Revisa la sección de modelos jerárquicos-Stan, puedes trabajar sobre el modelo que se propone aquí.

Primero definimos el modelo de starn

```
data {
    int N;
    int y[N];
    int nExp;
    int Exp[N];
}
parameters {
    real<lower=0,upper=1> theta[nExp];
    real<lower=0,upper=1> mu;
    real<lower=0>  kappa;
}
transformed parameters {
    real<lower=0>  a;
    real<lower=0>  b;
    a = mu * kappa;
    b = (1-mu) * kappa;
}
model {
    theta ~ beta(a,b);
    y ~ bernoulli(theta[Exp]);
    mu ~ beta(1, 1);
    kappa ~ gamma(1, 0.1);
}
```

A continuación entrenamos el modelo

```
stan_rabbits_fit <- sampling(rabbits,
                             data = list(y = x$tumor, Exp = x$experiment, N = 1810, nExp = 71),
                             chains = 3,
                             iter = 1000,
                             warmup = 500)
```

```
##
## SAMPLING FOR MODEL '6c0641a968847f619b57e8f1c149650e' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.766 seconds (Warm-up)
## Chain 1:                0.657 seconds (Sampling)
```

```
## Chain 1:                1.423 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '6c0641a968847f619b57e8f1c149650e' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.758 seconds (Warm-up)
## Chain 2:                0.635 seconds (Sampling)
## Chain 2:                1.393 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '6c0641a968847f619b57e8f1c149650e' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.749 seconds (Warm-up)
## Chain 3:                0.611 seconds (Sampling)
## Chain 3:                1.36 seconds (Total)
## Chain 3:
```

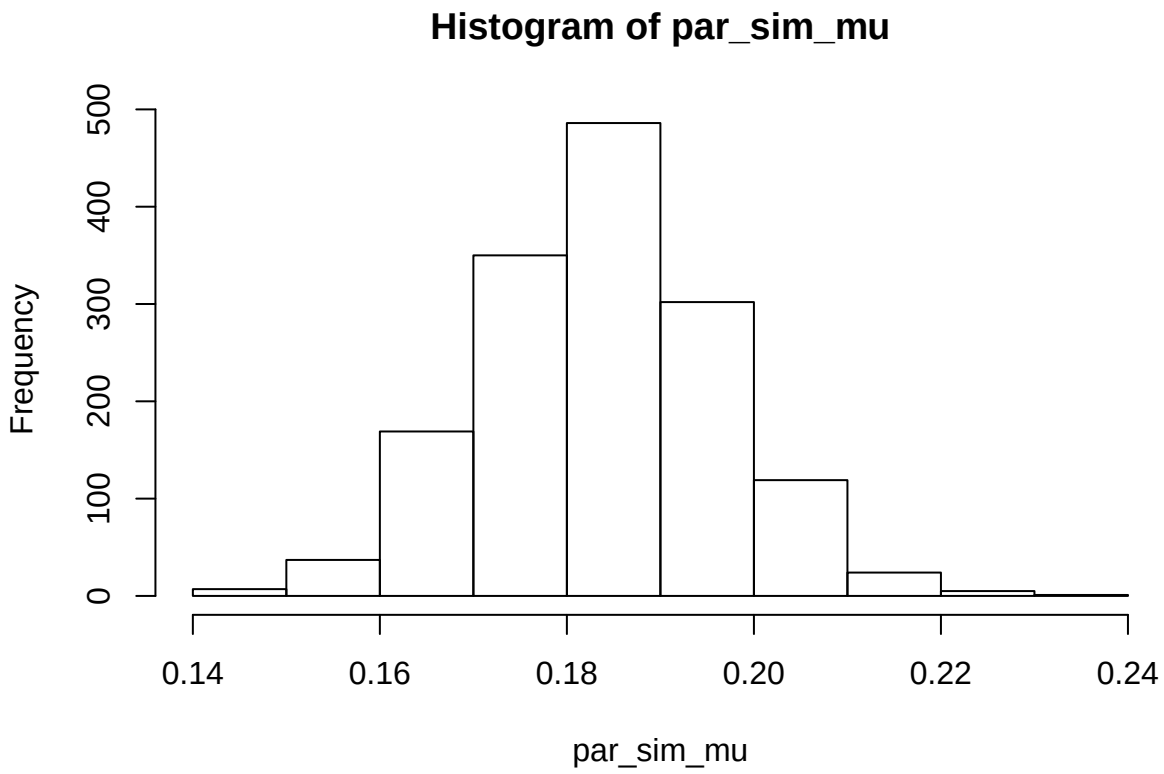
- Revisa la salida de Stan para diagnosticar convergencia y para asegurar un tamaño efectivo de muestra

razonable.

- Realiza un histograma de la distribución posterior de μ, κ. Comenta tus resultados.

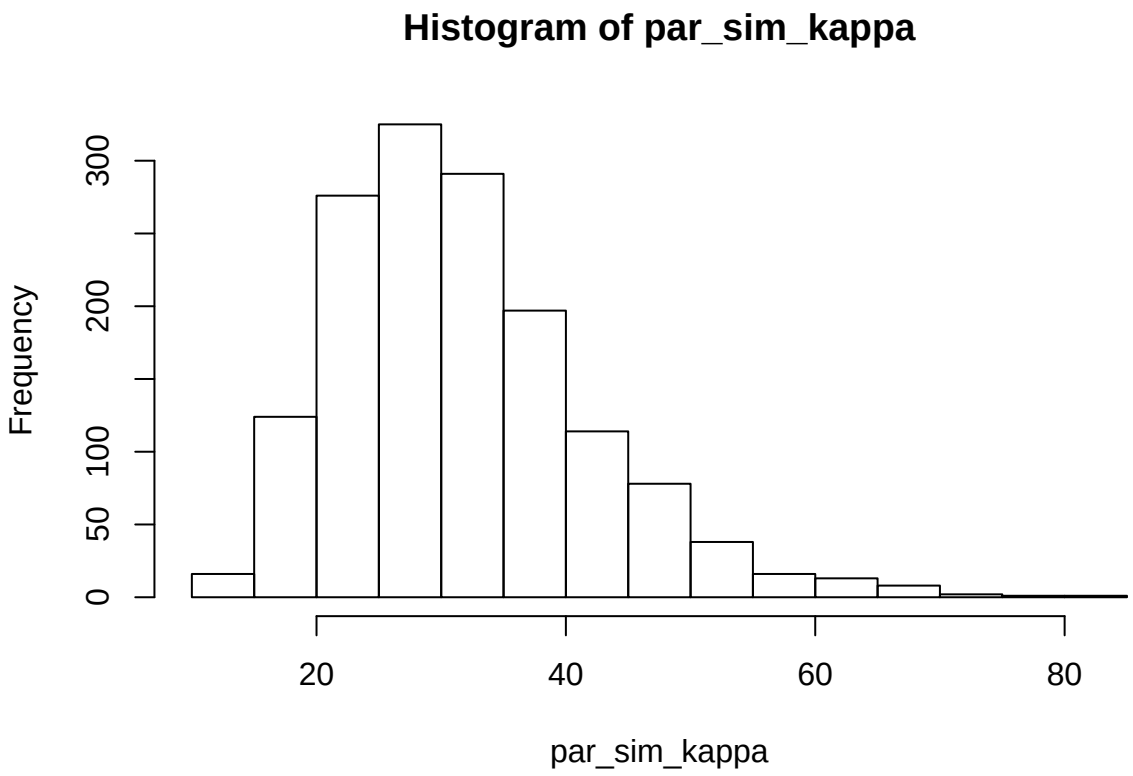
El histograma para μ es

```
par_sim <- extract(stan_rabbits_fit)
par_sim_mu <- par_sim$mu

hist(par_sim_mu)
```



El histograma para κ es

```
par_sim_kappa <- par_sim$kappa

hist(par_sim_kappa)
```



3. Ajusta un nuevo modelo utilizando una iniciales $Beta(10, 10)$ y $Gamma(0.51, 0.01)$ para μ y κ (lo demás quedará igual).

```
data {
    int N;
    int y[N];
    int nExp;
    int Exp[N];
}
parameters {
    real<lower=0,upper=1> theta[nExp];
    real<lower=0,upper=1> mu;
    real<lower=0>  kappa;
}
transformed parameters {
    real<lower=0>  a;
    real<lower=0>  b;
    a = mu * kappa;
    b = (1-mu) * kappa;
}
model {
    theta ~ beta(a,b);
    y ~ bernoulli(theta[Exp]);
    mu ~ beta(1, 1);
    kappa ~ gamma(1, 0.1);
}
```

```r
stan_rabbits_fit2 <- sampling(rabbits2,
                              data = list(y = x$tumor, Exp = x$experiment, N = 1810, nExp = 71),
                              chains = 3,
                              iter = 1000,
                              warmup = 500)
```

```
##
## SAMPLING FOR MODEL '6c0641a968847f619b57e8f1c149650e' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.848 seconds (Warm-up)
## Chain 1:                0.623 seconds (Sampling)
## Chain 1:                1.471 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '6c0641a968847f619b57e8f1c149650e' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.795 seconds (Warm-up)
## Chain 2:                0.603 seconds (Sampling)
```

```
## Chain 2:                1.398 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '6c0641a968847f619b57e8f1c149650e' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 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: 0.782 seconds (Warm-up)
## Chain 3:                0.654 seconds (Sampling)
## Chain 3:                1.436 seconds (Total)
## Chain 3:
```

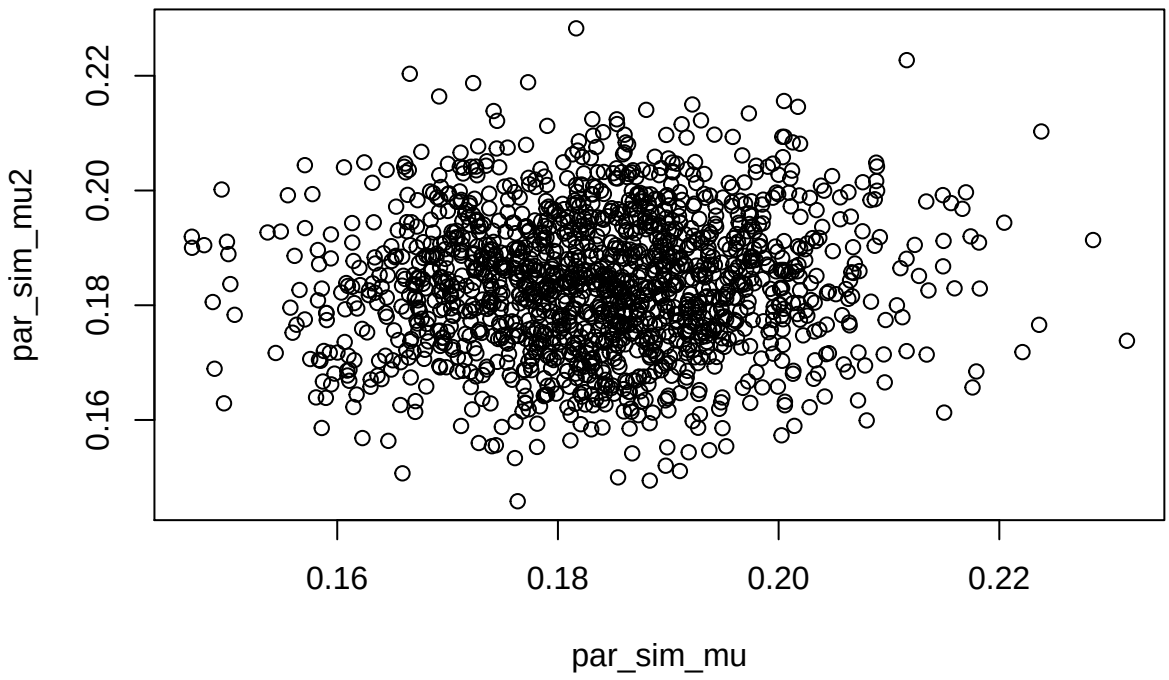
```
## Warning: Bulk Effective Samples Size (ESS) is too low, indicating posterior means and medians may be
## Running the chains for more iterations may help. See
## http://mc-stan.org/misc/warnings.html#bulk-ess
```

```
par_sim2 <- extract(stan_rabbits_fit2)
par_sim_mu2 <- par_sim2$mu
par_sim_kappa2 <- par_sim2$kappa
```

- Realiza una gráfica con las medias posteriores de los parámetros θ_j bajo los dos escenarios de distribuciones iniciales: en el eje horizontal grafica las medias posteriores del modelo ajustado en el paso anterior y en el eje vertical las medias posteriores del segundo modelo . ¿Cómo se comparan? ¿A qué se deben las diferencias?

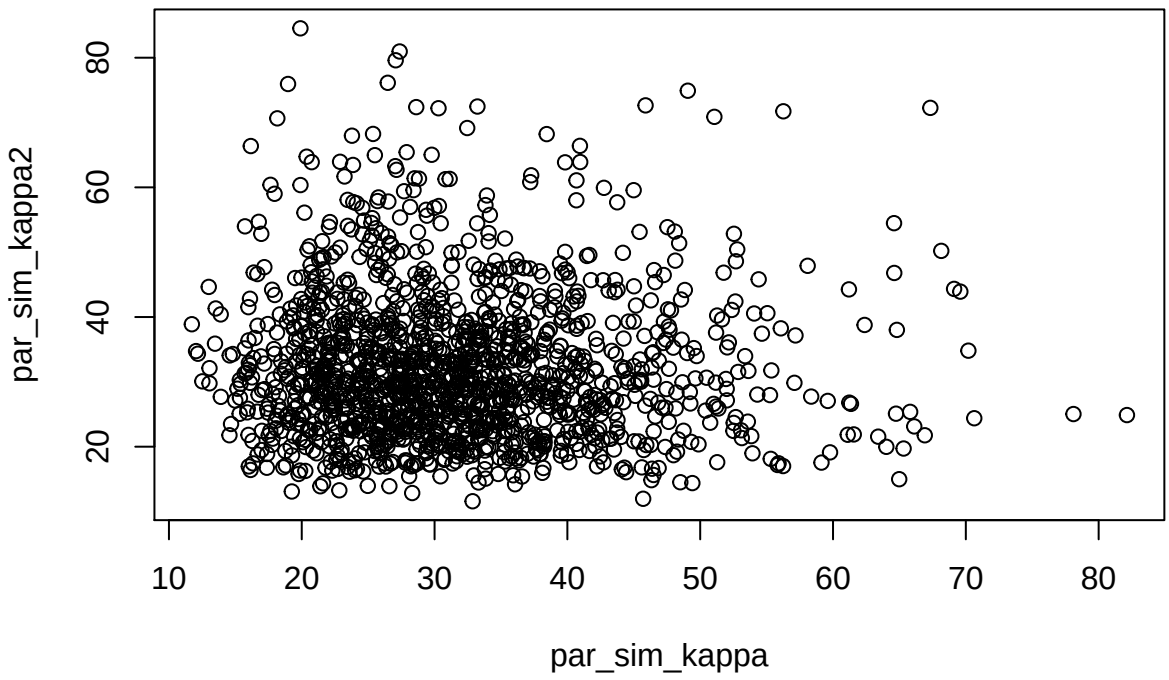
Primero graficamos los valores para μ.

```
plot(par_sim_mu,par_sim_mu2)
```



Y posteriormente graficamos los valores para κ.

```
plot(par_sim_kappa,par_sim_kappa2)
```


par_sim_kappa2
20
40
60
80
10
20
30
40
50
60
70
80
par_sim_kappa