Tiny Data, Approximate Bayesian Computation and the Socks of Karl Broman

Rasmus Bååth, Lund University
@rabaath || rasmus.baath@gmail.com
http://www.sumsar.net

Karl Broman
@kwbroman

Following

That the 1st 11 socks in the laundry are each distinct suggests there are a lot more socks.


THE
DAWN
OF
BIG DATA


STEREO
jackson 5
ABC
Approximate
Bayesian
Computation
MOTOWN

Approximate Bayesian Computation

- A method of figuring out *unknowns* that requires:
  - ✓ Data
  - → A *generative* model
  - *Priors*. What information the model has before seeing the data.
  - A *criterion* for when simulated data *matches* the actual data.

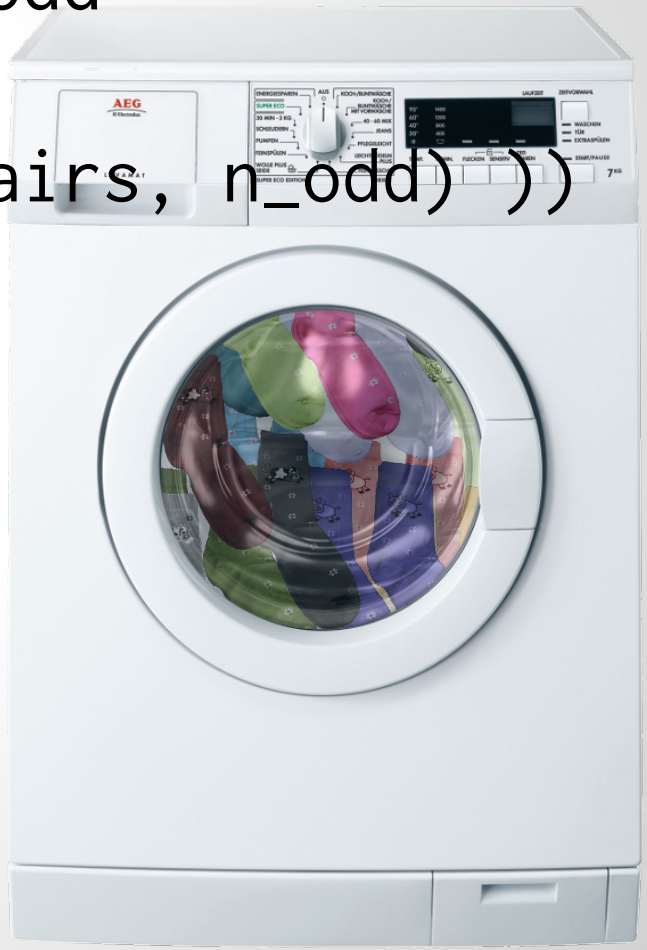
A Model of Picking out Socks from Your Washing Machine

```
n_pairs <- 9
```


ONE DOES NOT SIMPLY
MATCH ALL THE SOCKS

```
n_pairs <- 9
n_odd   <- 5
```

```
> socks
 [1]  1  1  2  2  3  3  4  4  5  5  6  6  7  7
8  8  9  9 10 11 12 13 14
n_sock_types <- n_pairs + n_odd
socks <- rep(1:n_sock_types,
             rep( 2:1, c(n_pairs, n_odd) ))
```



```
n_sock_types <- n_pairs + n_odd
socks <- rep(1:n_sock_types,
             rep( 2:1, c(n_pairs, n_odd) ))
```

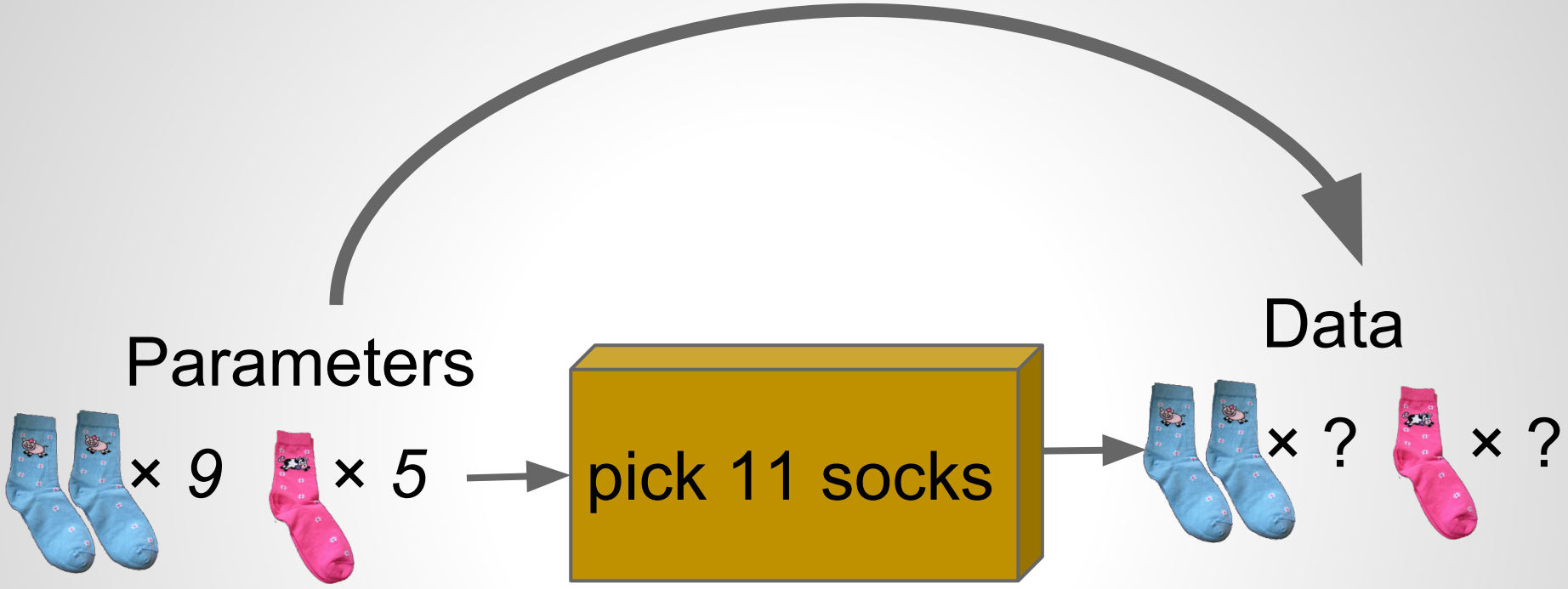
```
n_sock_types <- n_pairs + n_odd
socks <- rep(1:n_sock_types,
             rep( 2:1, c(n_pairs, n_odd) ))
```

```
n_sock_types <- n_pairs + n_odd
socks <- rep(1:n_sock_types,
             rep( 2:1, c(n_pairs, n_odd) ))
picked_socks <- sample(socks, 11)
sock_counts <- table(picked_socks)
> sock_counts
picked_socks
 1  3  4  5  7  8  9 10 11
 1  2  2  1  1  1  1  1  1
```

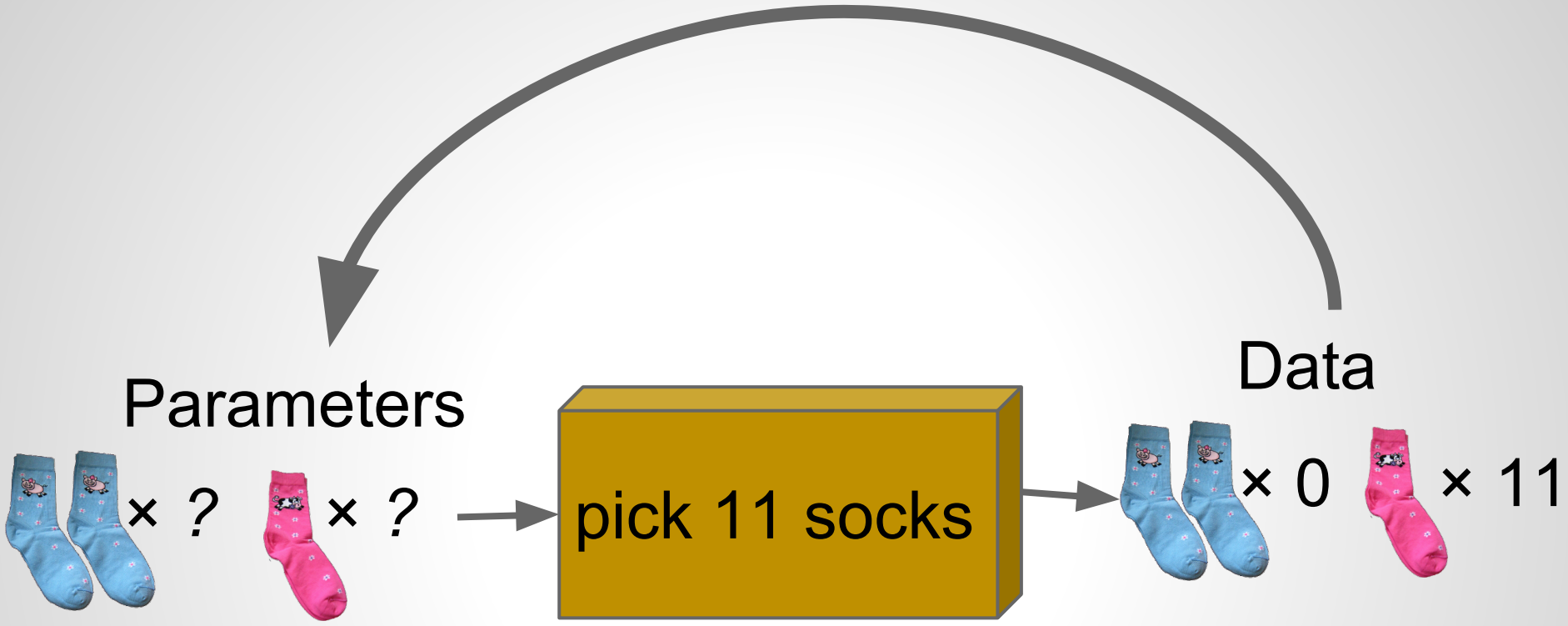
```
n_sock_types <- n_pairs + n_odd
socks <- rep(1:n_sock_types,
             rep( 2:1, c(n_pairs, n_odd) ))
picked_socks <- sample(socks, 11)
sock_counts <- table(picked_socks)

unique <- sum(sock_counts == 1)
pairs  <- sum(sock_counts == 2)
```

× 2

× 7




```
pick_socks(pairs = 9, odds = 5, n_pick = 11)
```




```
prob_socks(pairs = 0, odds = 11)
```

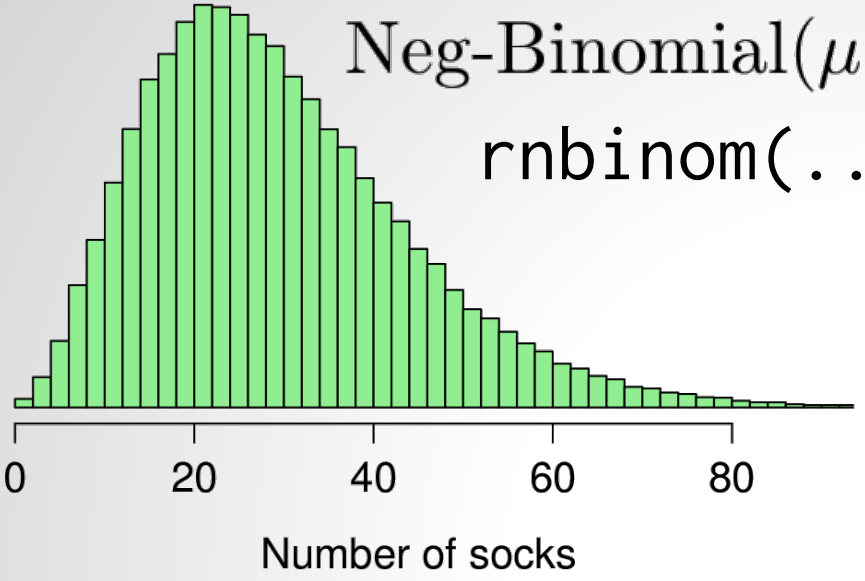
Approximate Bayesian Computation

- A method of figuring out *unknowns* that requires:
  - ✓ Data
  - ✓ A *generative* model
  - → *Priors*. What information the model has before seeing the data.
  - A *criterion* for when simulated data *matches* the actual data.

Prior on Number of Socks

$\text{Neg-Binomial}(\mu: 30, \sigma: 15)$

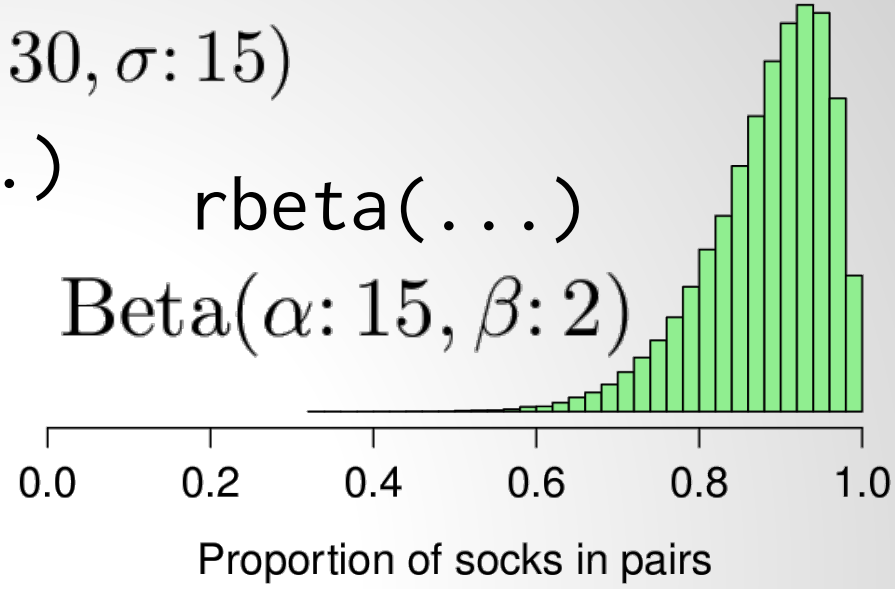
`rnbinom(...)`



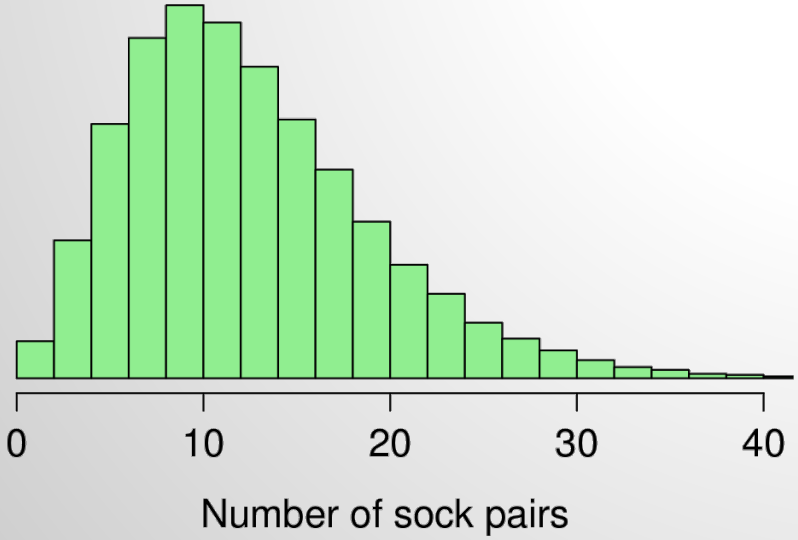
Prior on Proportion of Pairs

`rbeta(...)`

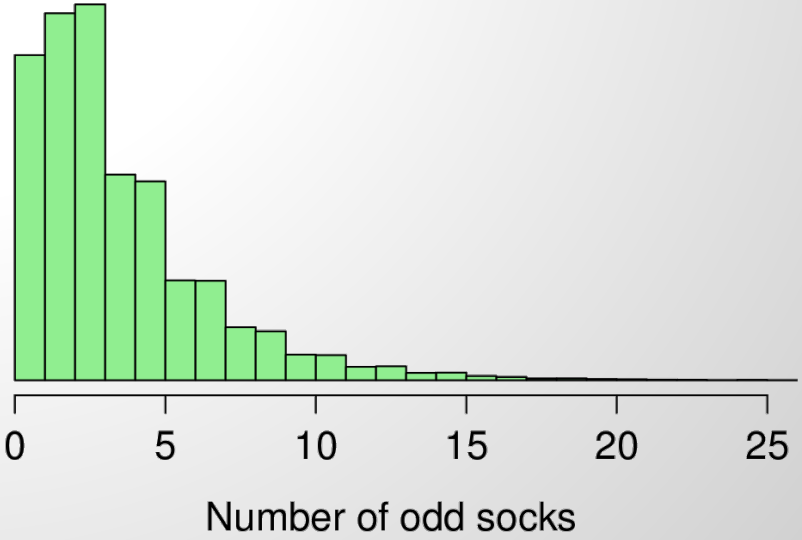
$\text{Beta}(\alpha: 15, \beta: 2)$



Resulting prior on Number of Pairs



Resulting prior on Number of Odd Socks

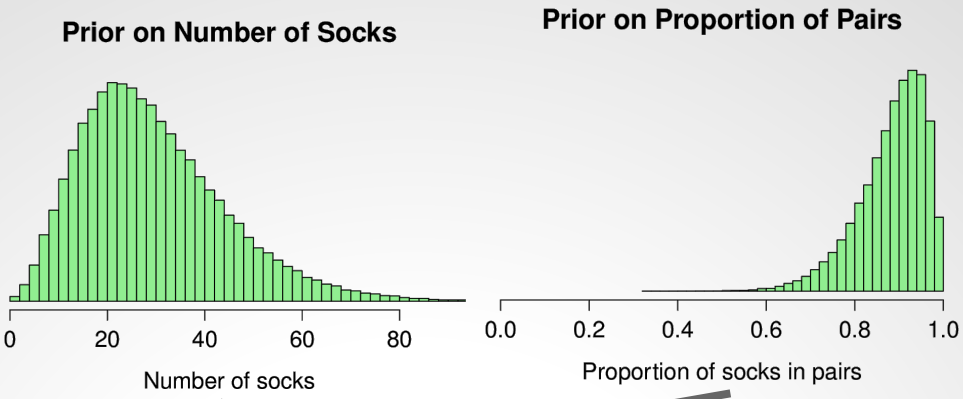


Approximate Bayesian Computation

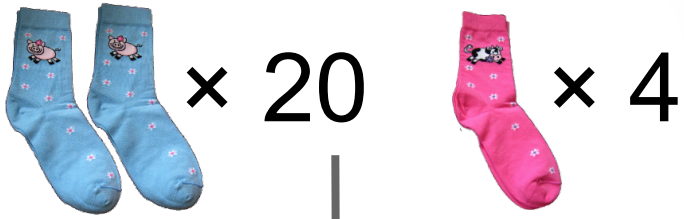
- A method of figuring out *unknowns* that requires:
  - ✓ Data
  - ✓ A *generative* model
  - ✓ *Priors*. What information the model has before seeing the data.
  - ✓ A *criterion* for when simulated data *matches* the actual data.

Let's do the ABC!

Priors



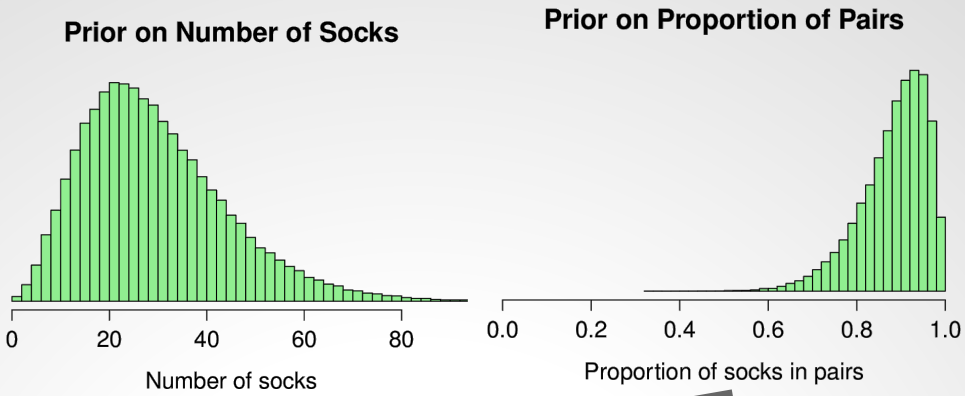
Parameters



Generative Model

```
pick_socks(20, 4, n_pick = 11)
```

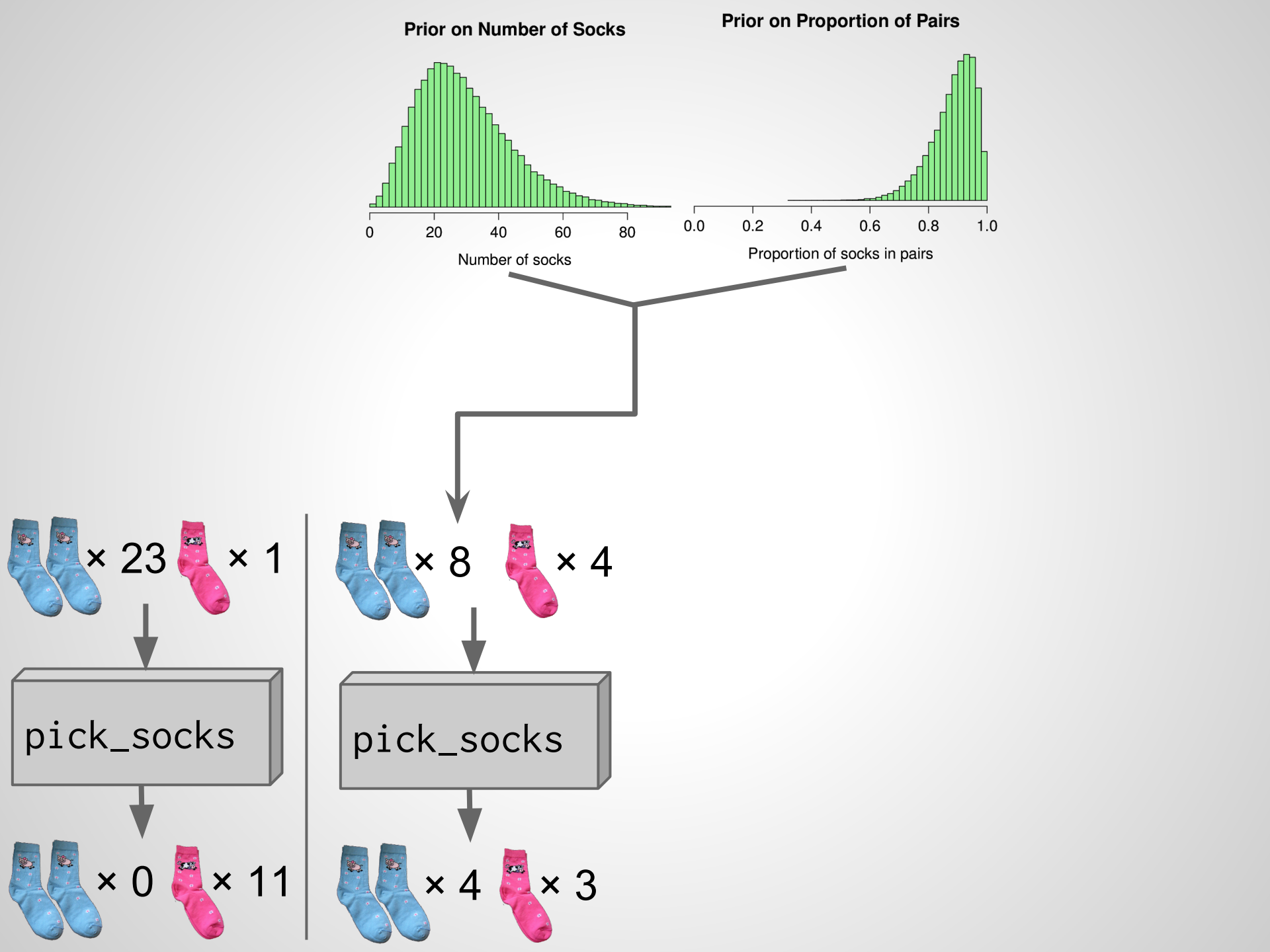
Simulated data

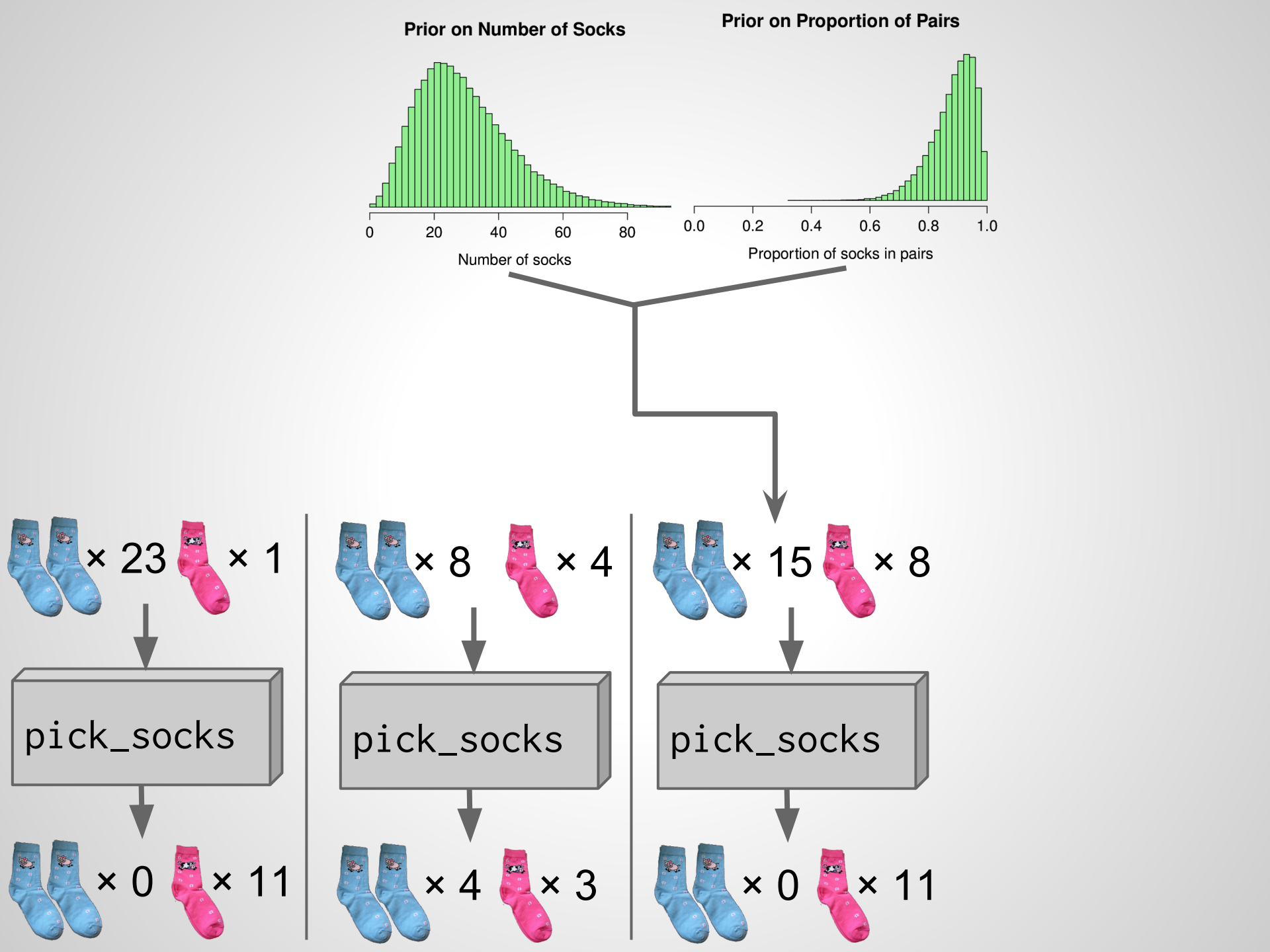

Prior on Number of Socks
0
20
40
60
80
Number of socks
Prior on Proportion of Pairs
0.0
0.2
0.4
0.6
0.8
1.0
Proportion of socks in pairs

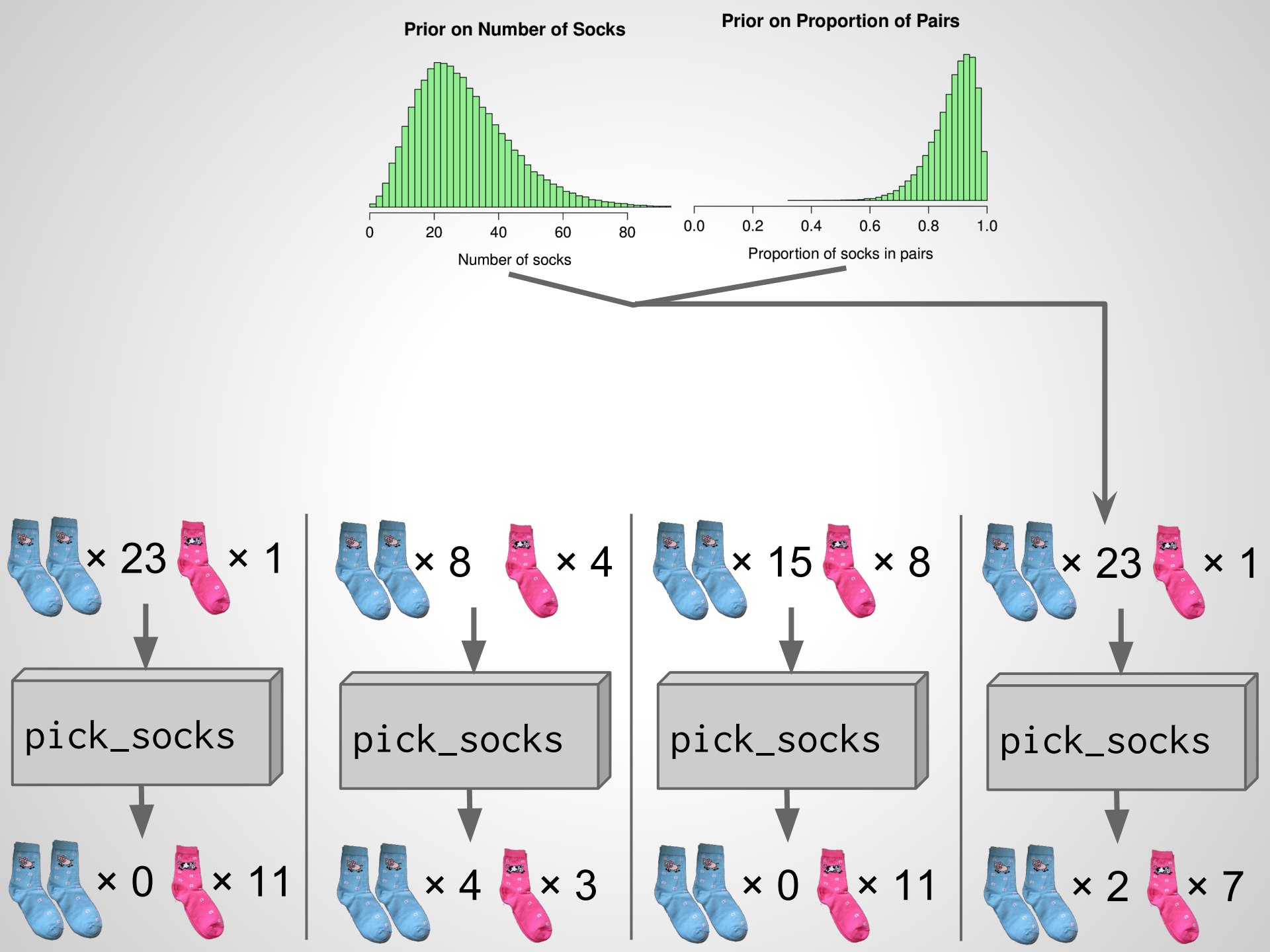

× 23
× 1

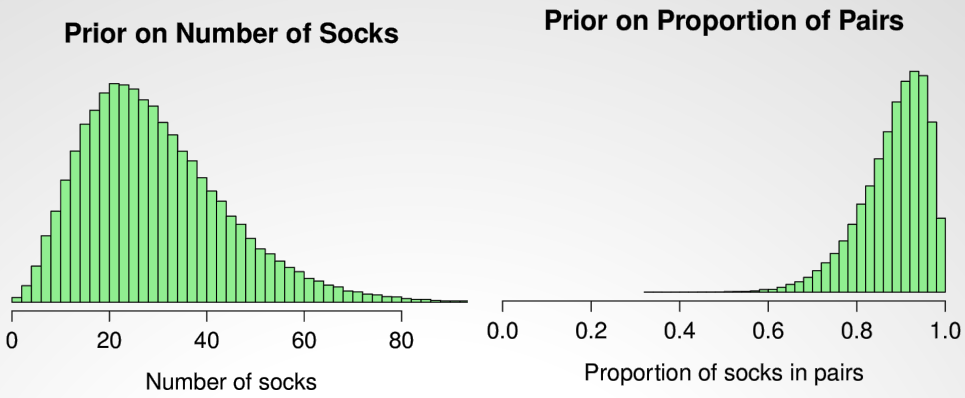
pick_socks

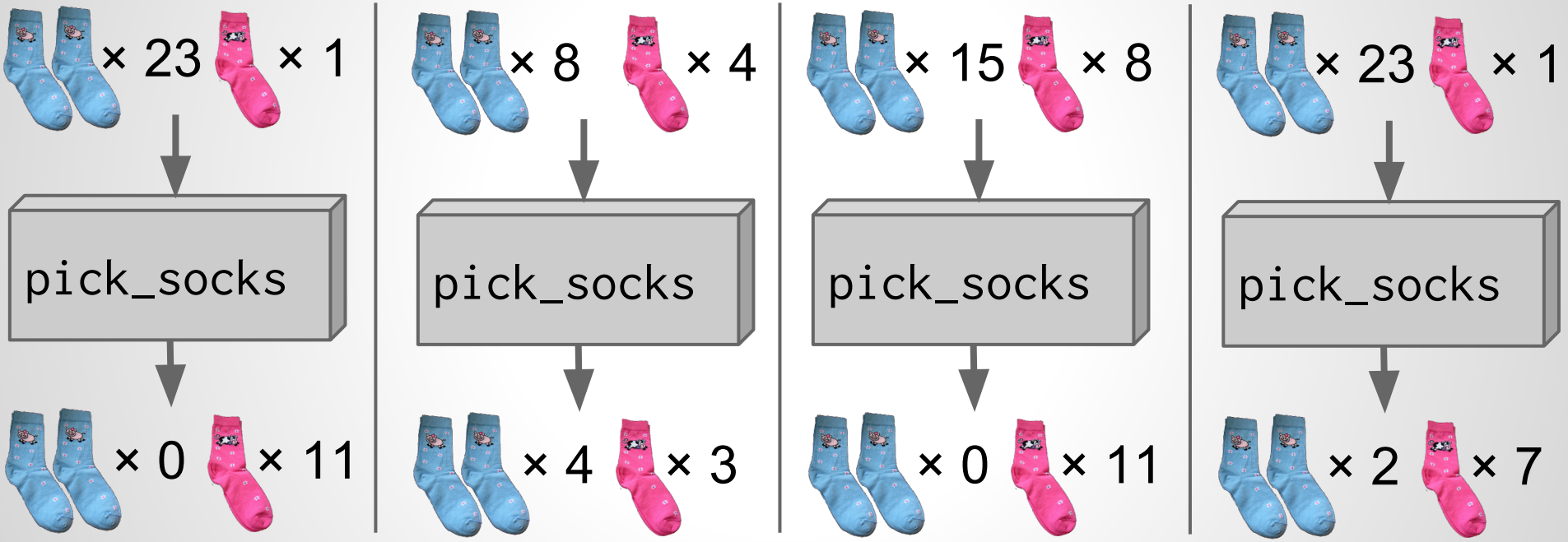

× 0
× 11


Prior on Number of Socks
0
20
40
60
80
Number of socks
Prior on Proportion of Pairs
0.0
0.2
0.4
0.6
0.8
1.0
Proportion of socks in pairs
× 23
× 1
pick_socks
× 0
× 11
× 8
× 4
pick_socks
× 4
× 3

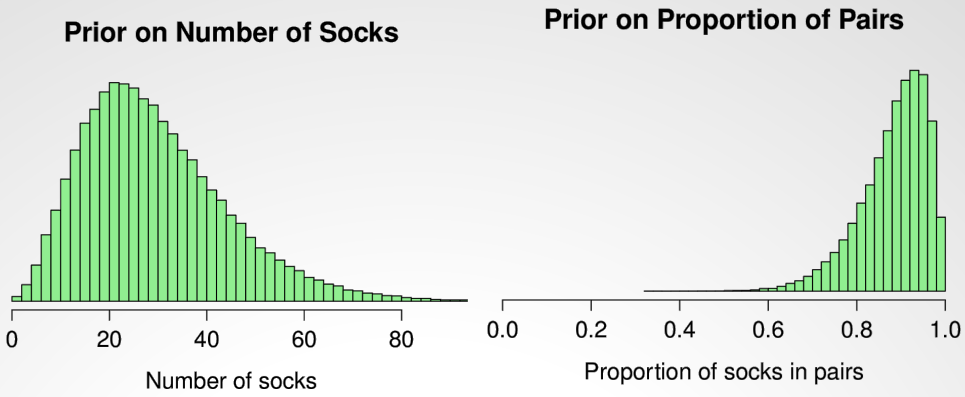

Prior on Number of Socks
0
20
40
60
80
Number of socks
Prior on Proportion of Pairs
0.0
0.2
0.4
0.6
0.8
1.0
Proportion of socks in pairs
× 23
× 1
pick_socks
× 0
× 11
× 8
× 4
pick_socks
× 4
× 3
× 15
× 8
pick_socks
× 0
× 11

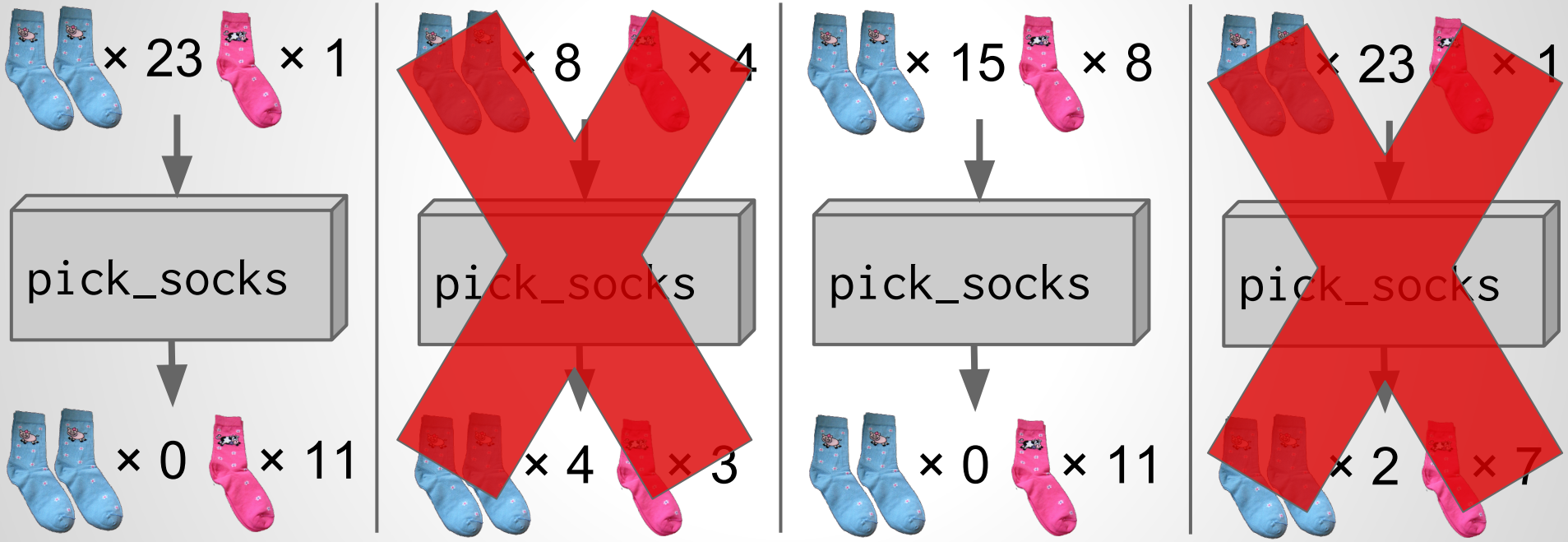

Prior on Number of Socks
0
20
40
60
80
Number of socks
Prior on Proportion of Pairs
0.0
0.2
0.4
0.6
0.8
1.0
Proportion of socks in pairs
× 23
× 1
pick_socks
× 0
× 11
× 8
× 4
pick_socks
× 4
× 3
× 15
× 8
pick_socks
× 0
× 11
× 23
× 1
pick_socks
× 2
× 7





Actual data: × 0 × 11



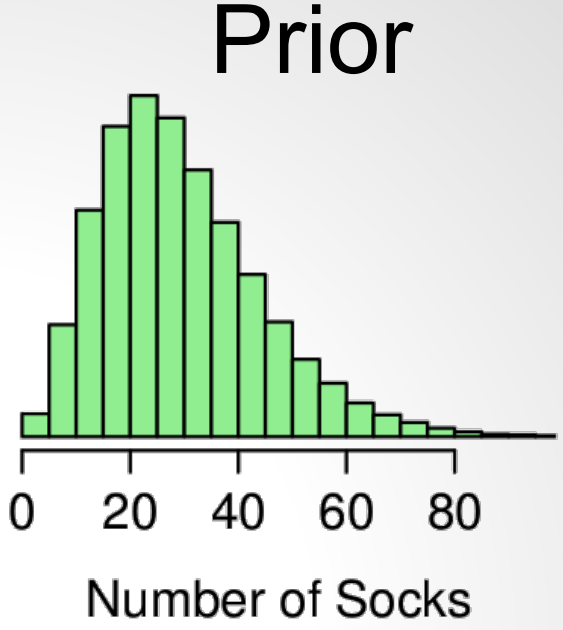


Actual data: × 0 × 11

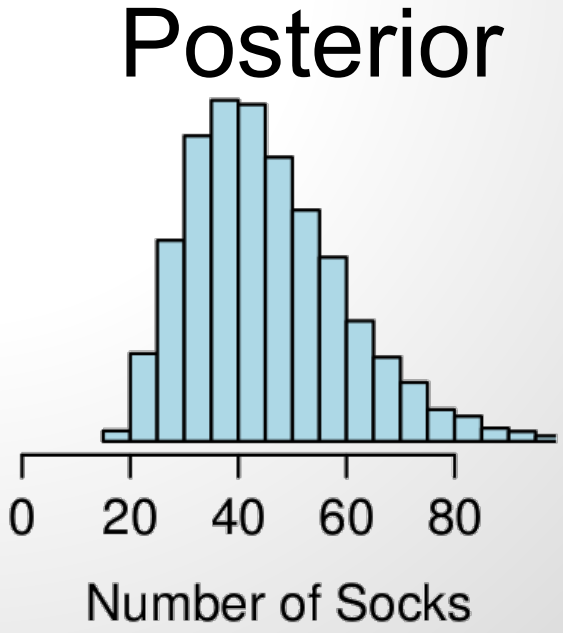
```
> head(sock_sim)
   n_pairs n_odd n_socks prop_pairs pairs unique
        12     4      28     0.8391     1      9
        11     4      26     0.8802     0     11
        19     2      40     0.9604     1      9
        11     4      26     0.8699     3      5
        24     5      53     0.9192     0     11
```

```
posterior <- subset(sock_sim, unique == 11)
> head(posterior)
   n_pairs n_odd n_socks prop_pairs pairs unique
        25     9      59     0.8626     0     11
        24    21      69     0.6980     0     11
        20    20      60     0.6580     0     11
        11     4      26     0.8802     0     11
```

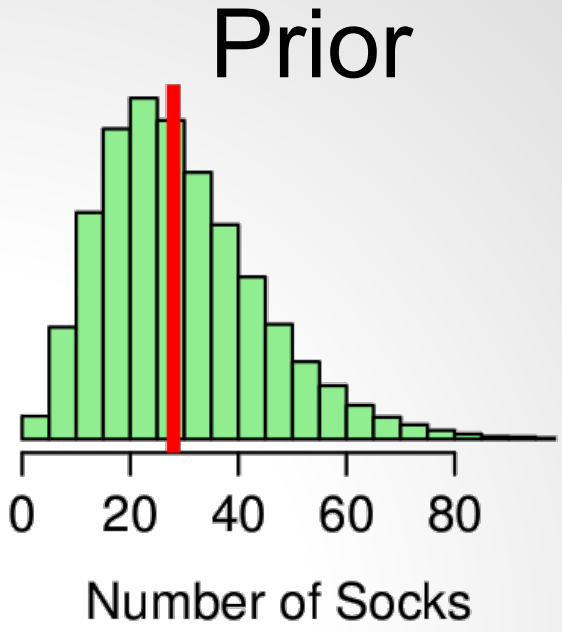
```
hist(sock_sim$n_socks)
```



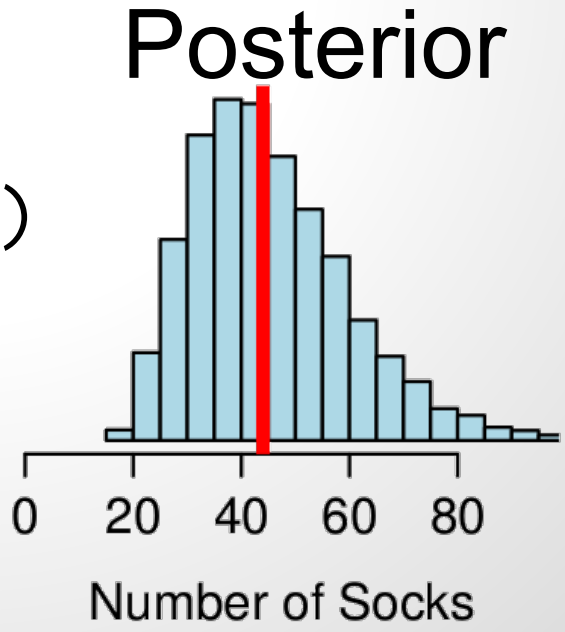

```
hist(posterior$n_socks)
```



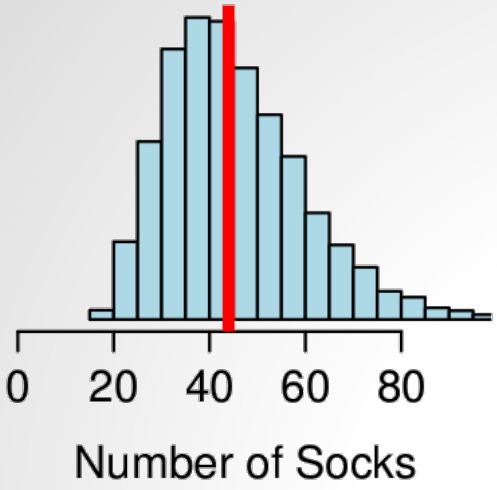
```
hist(sock_sim$n_socks)
median(sock_sim$n_socks)
```




```
hist(posterior$n_socks)
median(posterior$n_socks)
```

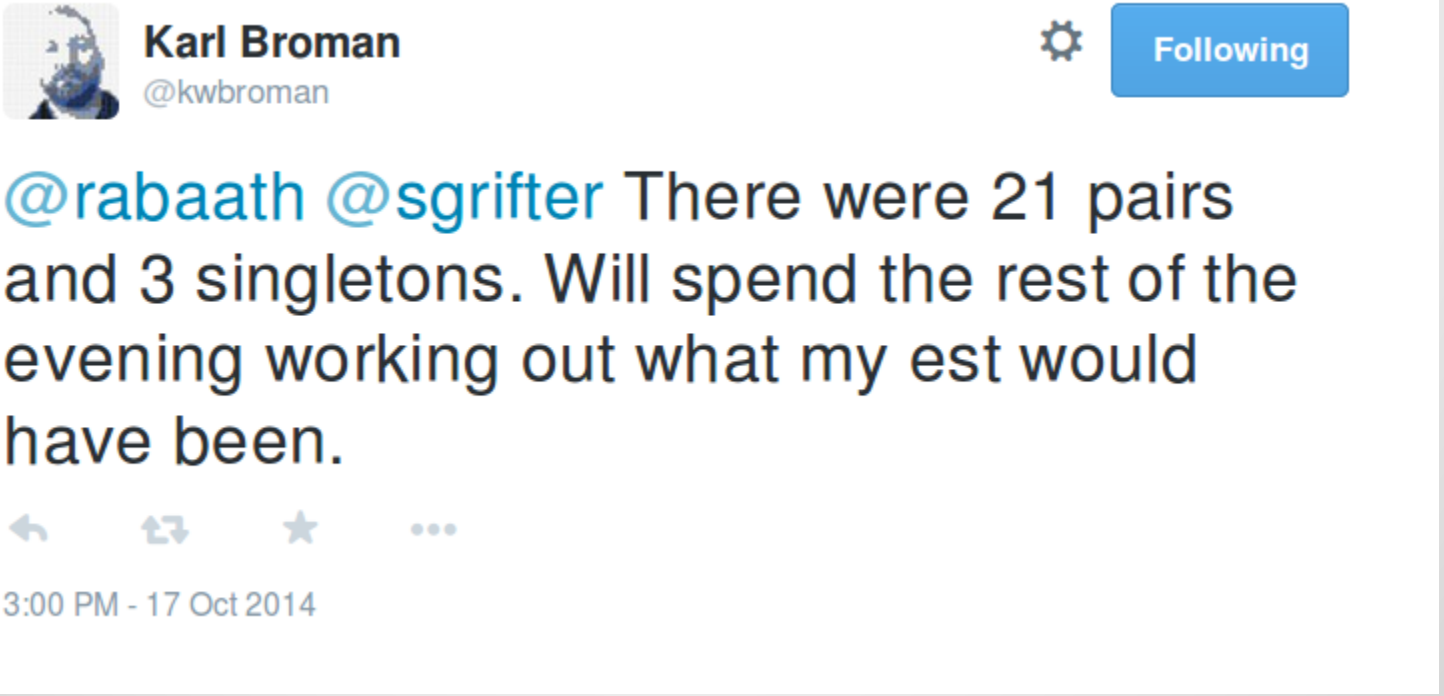


Our best guess: 44



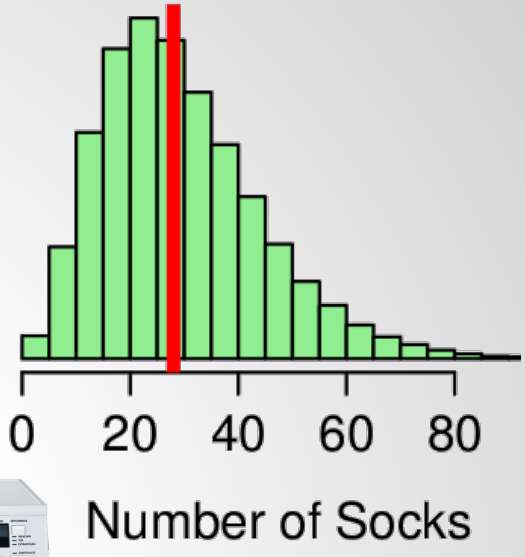
Actual number of socks:
21 x 2 + 3 = 45

Error:



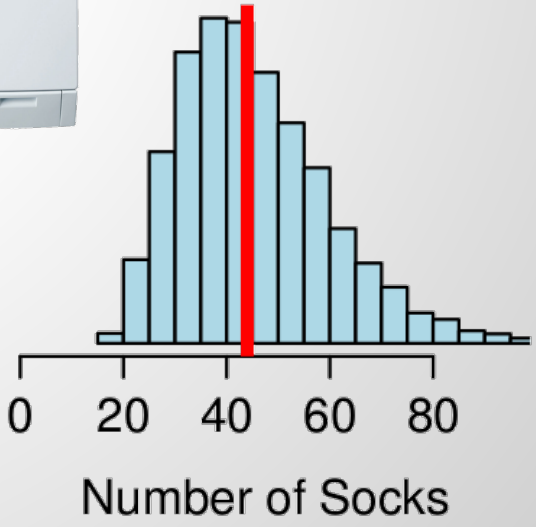
So, what have we done?

- We have specified prior information



- A generative model

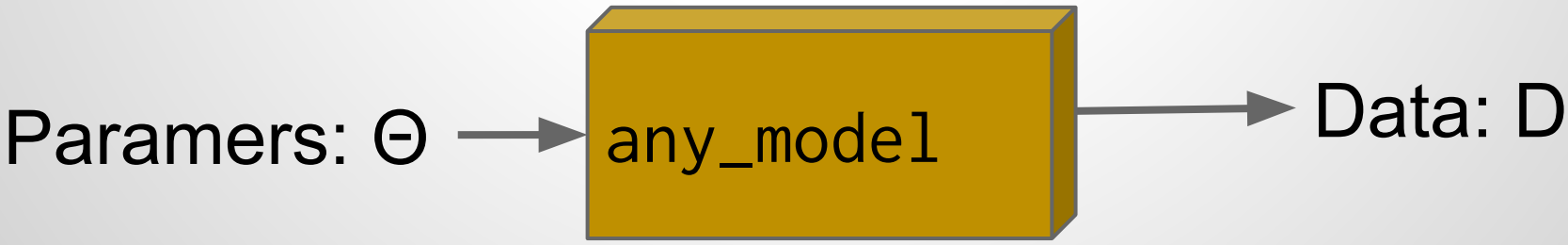
- And got out the probability of different parameter values using ABC.



So, what have we done?

- The example we used was about socks in Karl Broman's washing machine.

- But the general method works on *any* generative model.



In Conclusion

Approximate Bayesian Computation is

+ Information efficient
+ Principled
+ Very easy to code up in R

```
sock_sim <- t(replicate(100000, {
  n_socks <- rnbinom(1, mu = 30, size = -30^2 / (30 - 15^2) )
  prop_pairs <- rbeta(1, shape1 = 15, shape2 = 2)
  n_pairs <- round(floor(n_socks / 2) * prop_pairs)
  n_odd <- n_socks - n_pairs * 2

  n_sock_types <- n_pairs + n_odd
  socks <- rep(seq_len(n_sock_types), rep( 2:1, c(n_pairs, n_odd) ))
  picked_socks <- sample(socks, size =  min(11, n_socks))
  sock_counts <- table(picked_socks)

  c(unique = sum(sock_counts == 1), pairs = sum(sock_counts == 2),
    n_socks = n_socks, prop_pairs = prop_pairs)
}))

post_samples <- sock_sim[sock_sim[, "unique"] == 11 &
                         sock_sim[, "pairs" ] == 0 , ]
```

```
library(abc)
```

```
library(EasyABC)
```

In Conclusion

Approximate Bayesian Computation is

+ Information efficient
+ Principled
+ Very easy to code up in R
- So very slooow.

JAGS

STAN

What's wrong with the model?

✉ : rasmus.baath@gmail.com
⌂ : http://www.sumsar.net
: @rabaath

This talk was sponsored by: