Bayesian AMMIT

Alessandra Lemos

```
library(bammit)
library(R2jags)
library(ggplot2)
library(gridExtra)
```

1 - AMMI Model

The model is formulated as

$$y_{ij} = \mu + g_i + e_j + \sum_{q=1}^{Q} \lambda_q \gamma_{iq} \delta_{jq} + \epsilon_{ij},$$

where $\epsilon_{ij} \sim N(0, \sigma_y^2)$. We have two formulations:

Parameter	Josse's	Crossa's
μ	$N(\mu_m u, \sigma_\mu^2)$	$N(\mu_m u, \sigma_\mu^2)$
g_i	$N(0, \sigma_g^2)$	$N(\mu_g, \sigma_g^2)$
e_j	$N(0, \sigma_e^2)$	$N(\mu_e, \sigma_e^2)$
λ_q	$N^+(0, \sigma_\lambda^2)$	$N^+(\mu_\lambda, \sigma_\lambda^2)$
γ_{iq}	$N^+(0,1)$, i = 1; $N(0,1)$, $i > 1$, q = 1,. . . ,Q	spherical uniform
δ_{jq}	$N(0,1)$	spherical uniform
σ_y^2	$U(0, s^2_{\sigma_y^2})$	$Inv - Scaled - \chi^2(a, b)$

For both approaches , $\lambda_q > 0$ and $\lambda_{q-1} \geq \lambda_q$. In our formulation, we consider $\sigma_y^2 \sim \Gamma(a, b)$. In order to meet the model constraints, we perform the following procedure on the bilinear term. Let $\theta^\gamma_{iq} \sim N(0, \sigma^\theta)$, for $i = 1, ..., I-1$, and $\theta^\gamma_{Iq} = -\sum_{i \neq I} \theta^\gamma_{iq}$. Similarly, let $\theta^\delta_{jq} \sim N(0, \sigma^\theta)$, for $j = 1, ..., J-1$, and $\theta^\delta_{Jq} = -\sum_{j \neq J} \theta^\delta_{jq}$. Then,

$$\gamma_{iq} = \frac{\theta^\gamma_{iq}}{\sqrt{\sum_i \theta^{\gamma^2}_{iq}}} \text{ and } \delta_{jq} = \frac{\theta^\delta_{jq}}{\sqrt{\sum_j \theta^{\delta^2}_{jq}}}$$

- Simulation

Simulation scenarios

- Set $I = \{6, 12\}$, $I = \{4, 6\}$, $Q = \{1, 2, 3\}$.
- Fix $\lambda = \{10, 12, 25\}$.
- Fix $\mu = 100$, $\sigma_g = 10$, $\sigma_e = 10$, $\sigma_y = 2$.

In Jags model:

- Fix $\sigma_\theta^2 = 1$.
- Fix $\sigma_\theta^2 = 100$.
- Fix $\mu_\lambda = 10$, $\sigma_\lambda^2 = 1$, $\mu_g = 0$, $\sigma_g^2 = 10$, $\mu_e = 0$, $\sigma_e^2 = 10$, $a = 0.1$, $b = 0.1$.

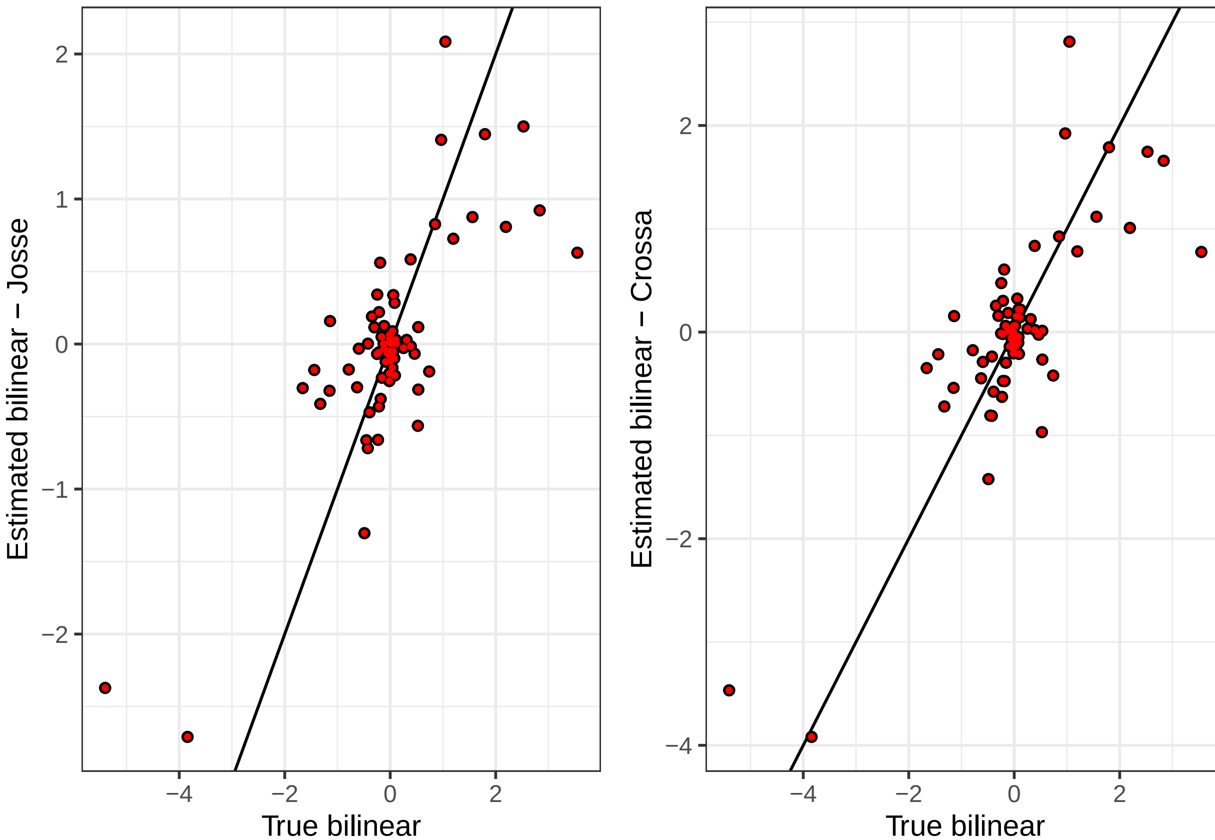
1.1 Jags implementation

```
modelCode <- "
 model
 {
 # Likelihood
  for (k in 1:N) {
   Y[k] ~ dnorm(mu[k], sy^-2)
   mu[k] = muall + g[genotype[k]] + e[environment[k]] + blin[k]
   blin[k] = sum(lambda[1:Q] * gamma[genotype[k],1:Q] * delta[environment[k],1:Q])
  }

 # Priors
 # Prior on grand mean
  muall ~ dnorm(mmu, smu^-2)

 # Prior on genotype effect
 for(i in 1:I) {
 g[i] ~ dnorm(mug, sg^-2) # Prior on genotype effect
 }

 # Prior on environment effect
 for(j in 1:J) {
 e[j] ~ dnorm(mue, se^-2) # Prior on environment effect
 }


 # Priors on gamma
 for(q in 1:Q){
   for(i in 1:(I-1)){
     theta[i,q] ~ dnorm(0,stheta)
   }
   theta[I,q] = -sum(theta[1:(I-1),q])
   thetaSum[q] = sqrt(sum(theta[1:I,q]^2)) + 0.00001
   for(i in 1:I){
     gamma[i,q] = theta[i,q]/thetaSum[q]
   }
 }

 # Priors on delta
  for(q in 1:Q){
   for(j in 1:(J-1)){
     aux[j,q] ~ dnorm(0,stheta)
   }
   aux[J,q] = -sum(aux[1:(J-1),q])
   auxSum[q] = sqrt(sum(aux[1:J,q]^2)) + 0.000001
   for(j in 1:J){
     delta[j,q] = aux[j,q]/auxSum[q]
   }
 }
```

```
# Prior on eigenvalues
for(q in 1:Q) {
  lambda_raw[q] ~ dnorm(mulambda, slambda^-2)T(0,)
}
lambda = sort(lambda_raw)

# Prior on residual standard deviation
 sy ~ dgamma(a, b) # inverse of sy
}
"
```

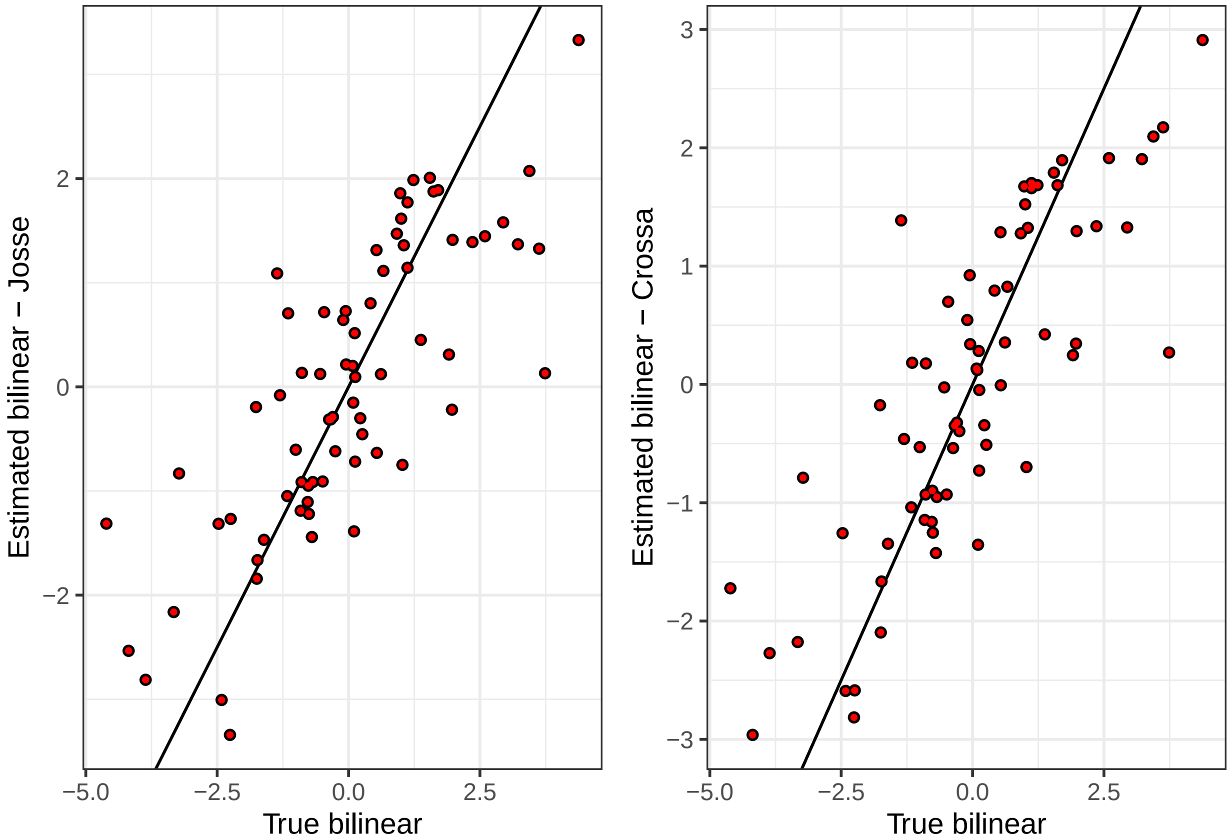
1.2 - Results

1.2.1 $\sigma^2_\theta = 1$

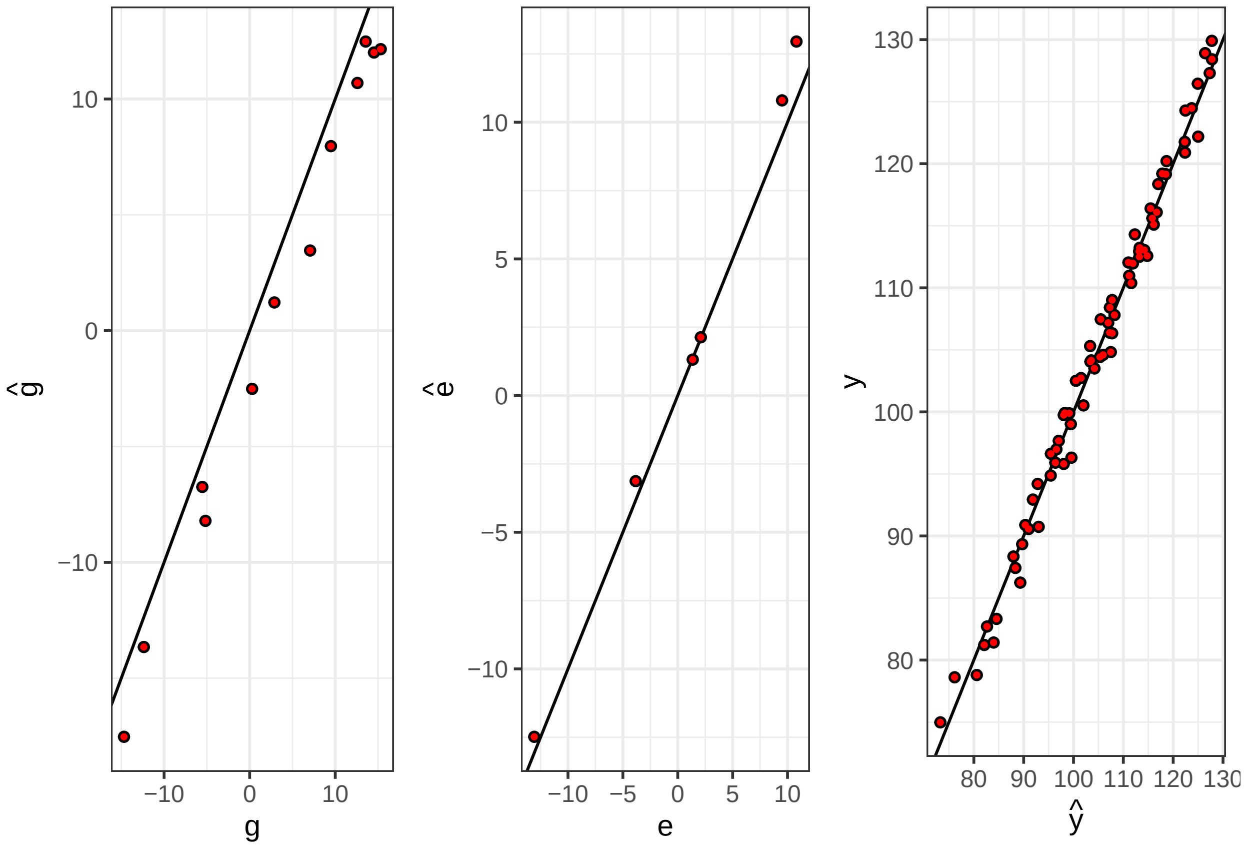


1.2.1.1 $\mathbf{Q = 1}$

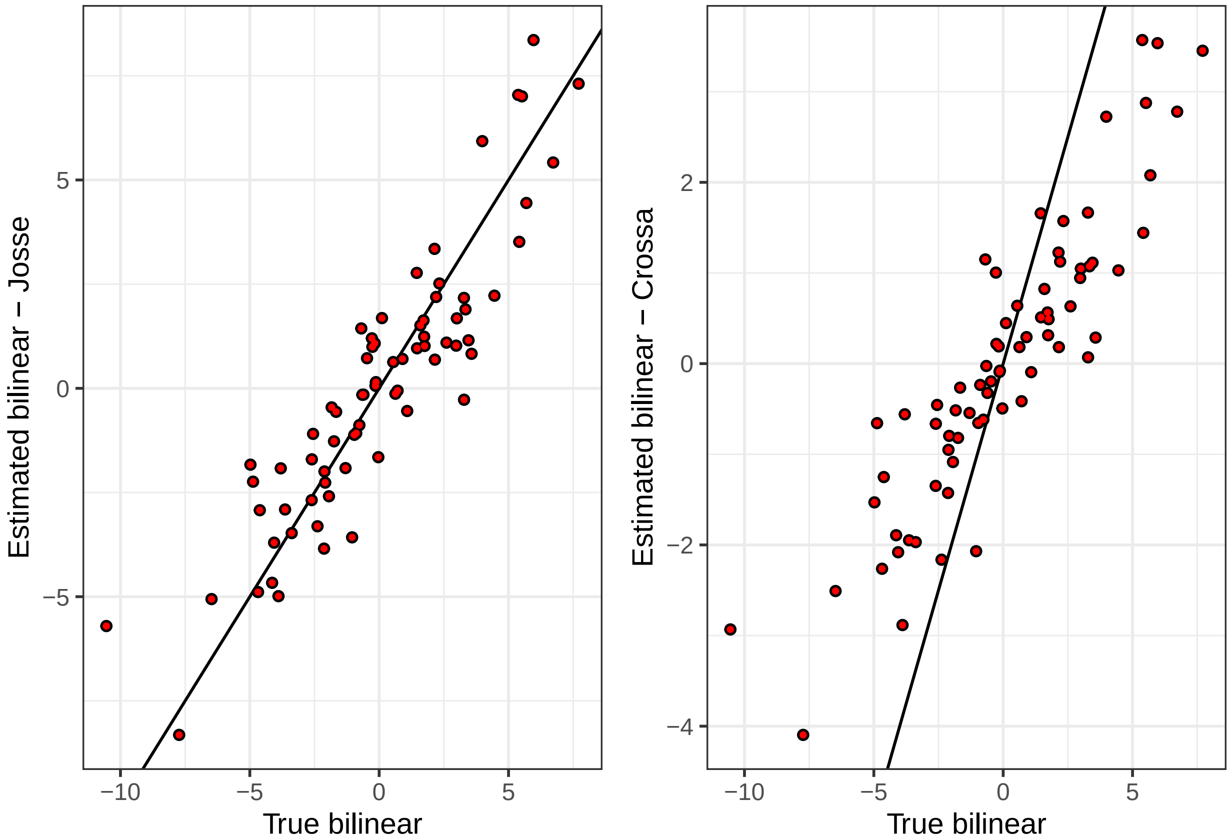
1.2.2 Q = 2

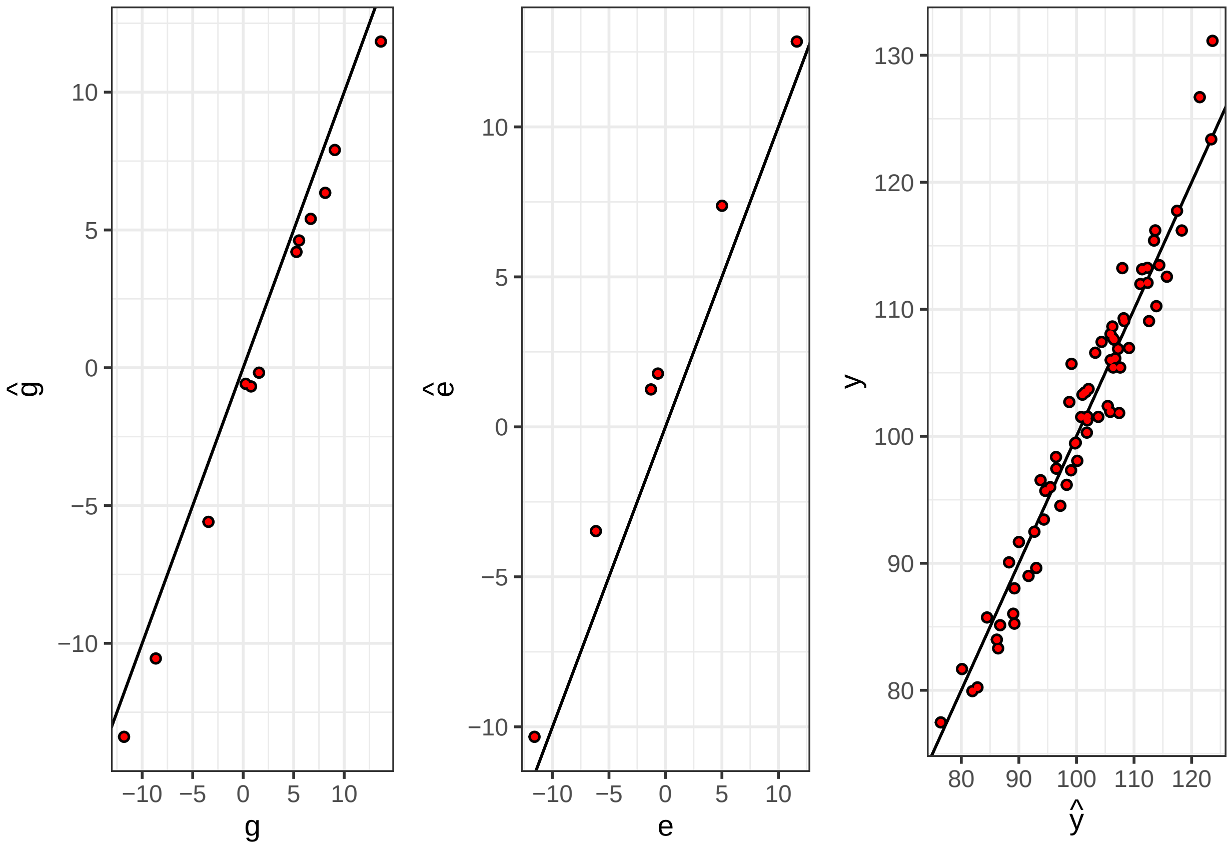


Estimated bilinear – Josse
True bilinear
Estimated bilinear – Crossa
True bilinear
−5.0
−2.5
0.0
2.5
−2
0
2
−3
−1
1
3


10
0
−10
ĝ
−10
0
10
g
10
5
0
−5
−10
ê
−10
−5
0
5
10
e
130
120
110
100
90
80
y
80
90
100
110
120
130
ŷ

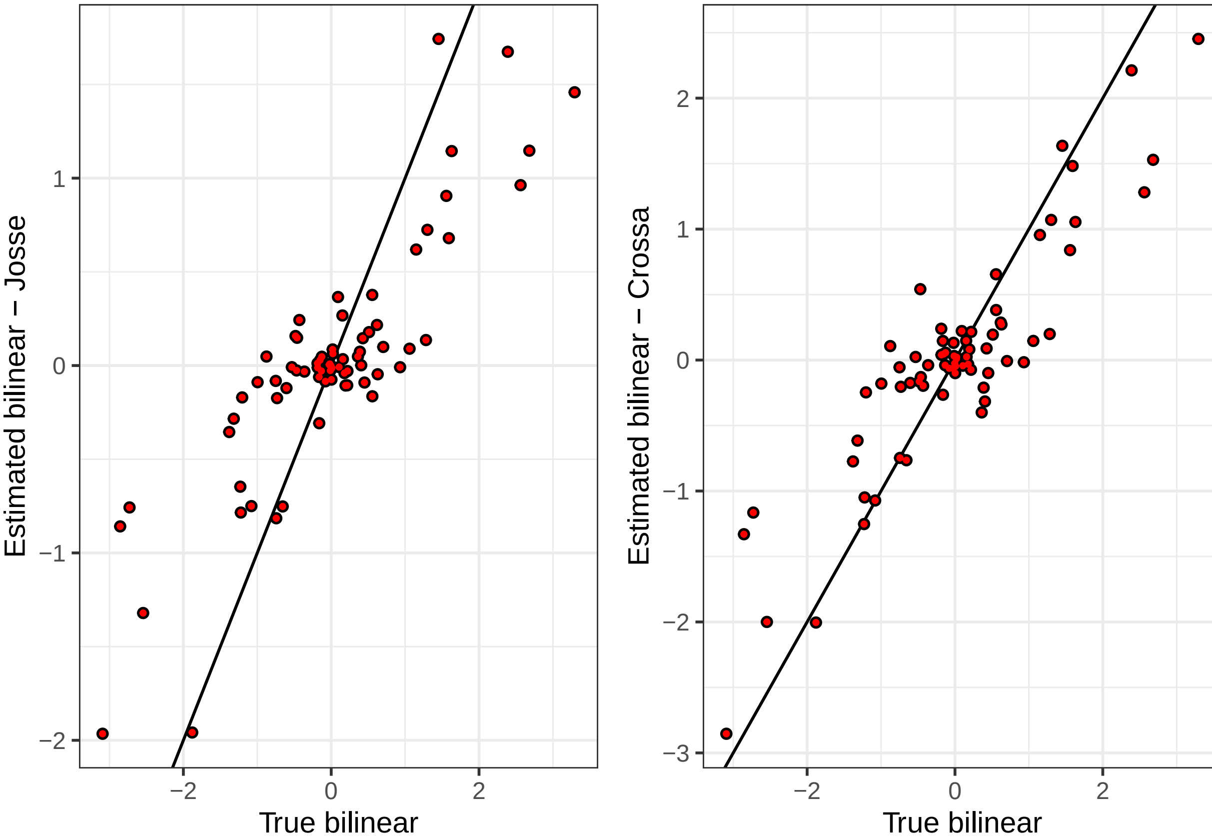
1.2.3 Q = 3


Estimated bilinear − Josse
5
0
−5
−10
−5
0
5
True bilinear
Estimated bilinear − Crossa
2
0
−2
−4
−10
−5
0
5
True bilinear



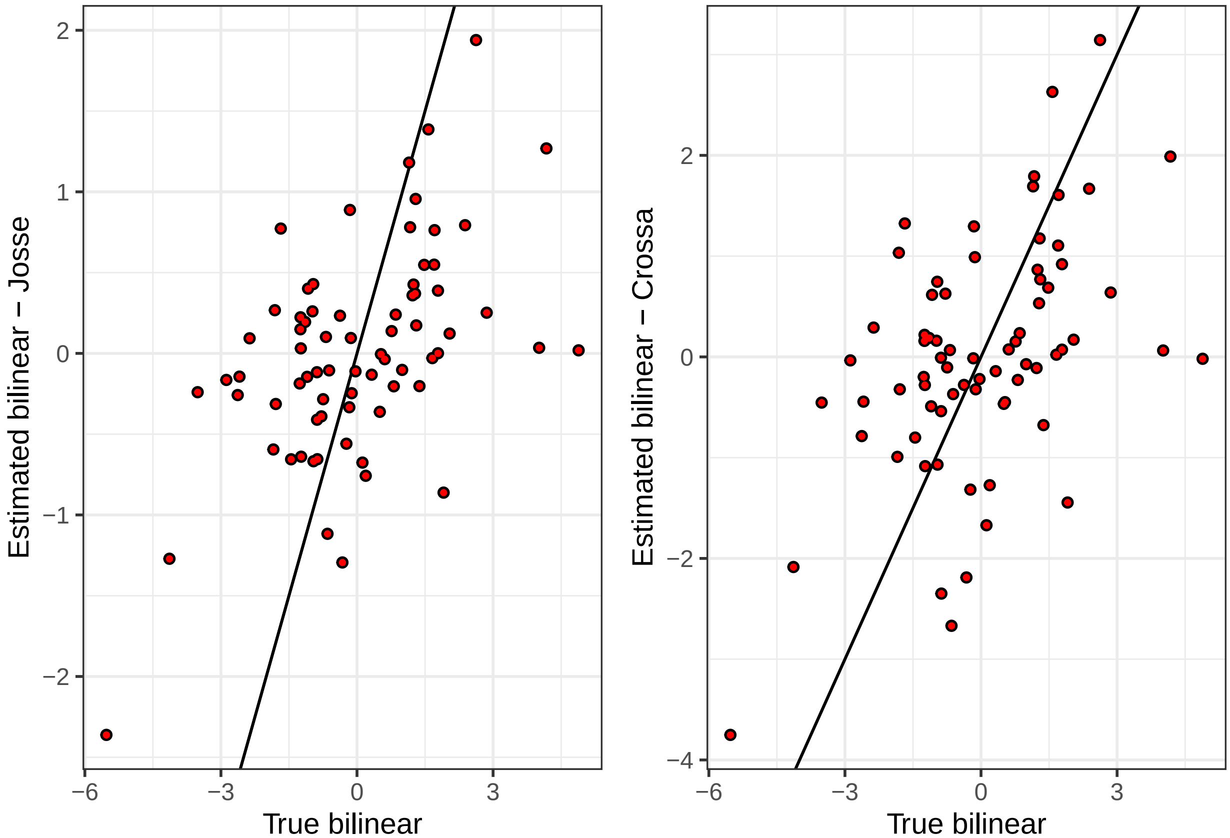
10
5
0
−5
−10
ĝ
−10
−5
0
5
10
g
10
5
0
−5
−10
ê
−10
−5
0
5
10
e
130
120
110
100
90
80
y
80
90
100
110
120
ŷ

1.2.4 $\sigma^2_\theta = 100$

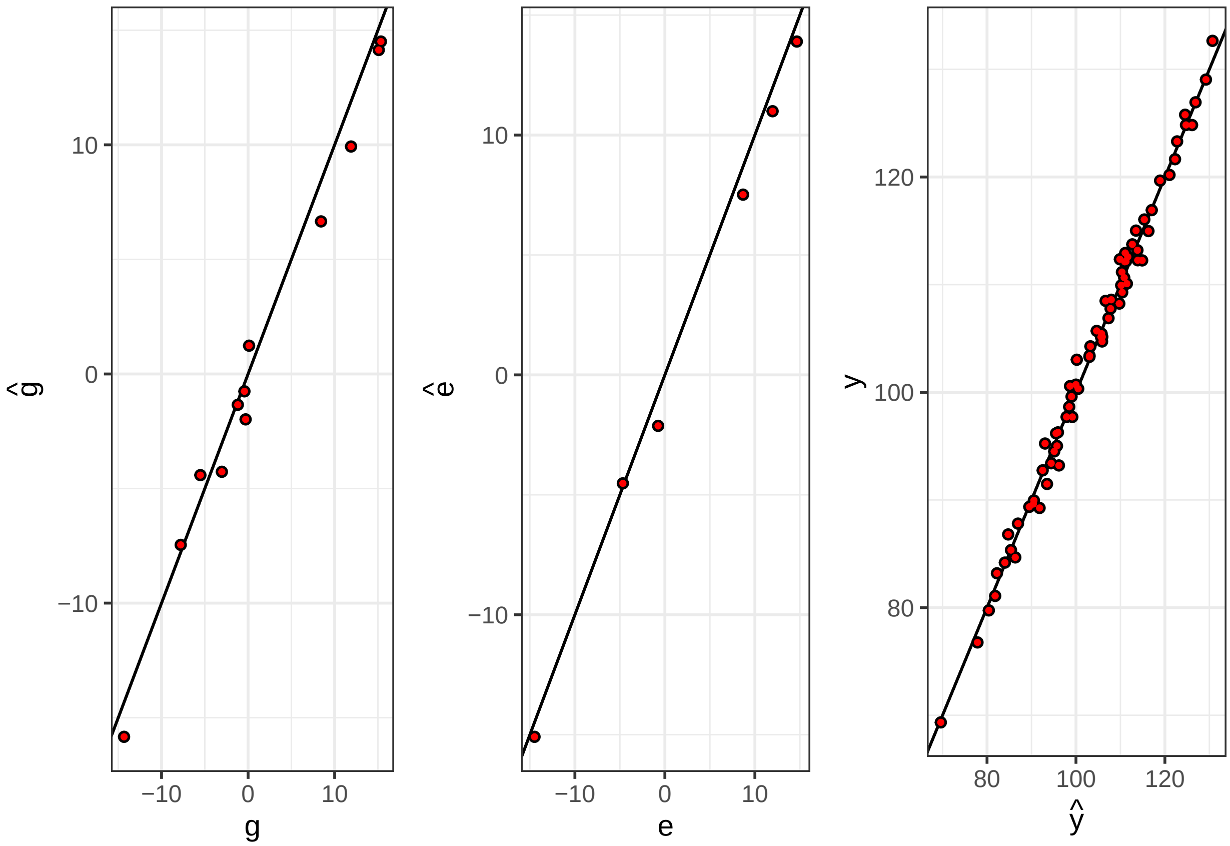


1.2.4.1 $\mathbf{Q = 1}$

1.2.5 Q = 2

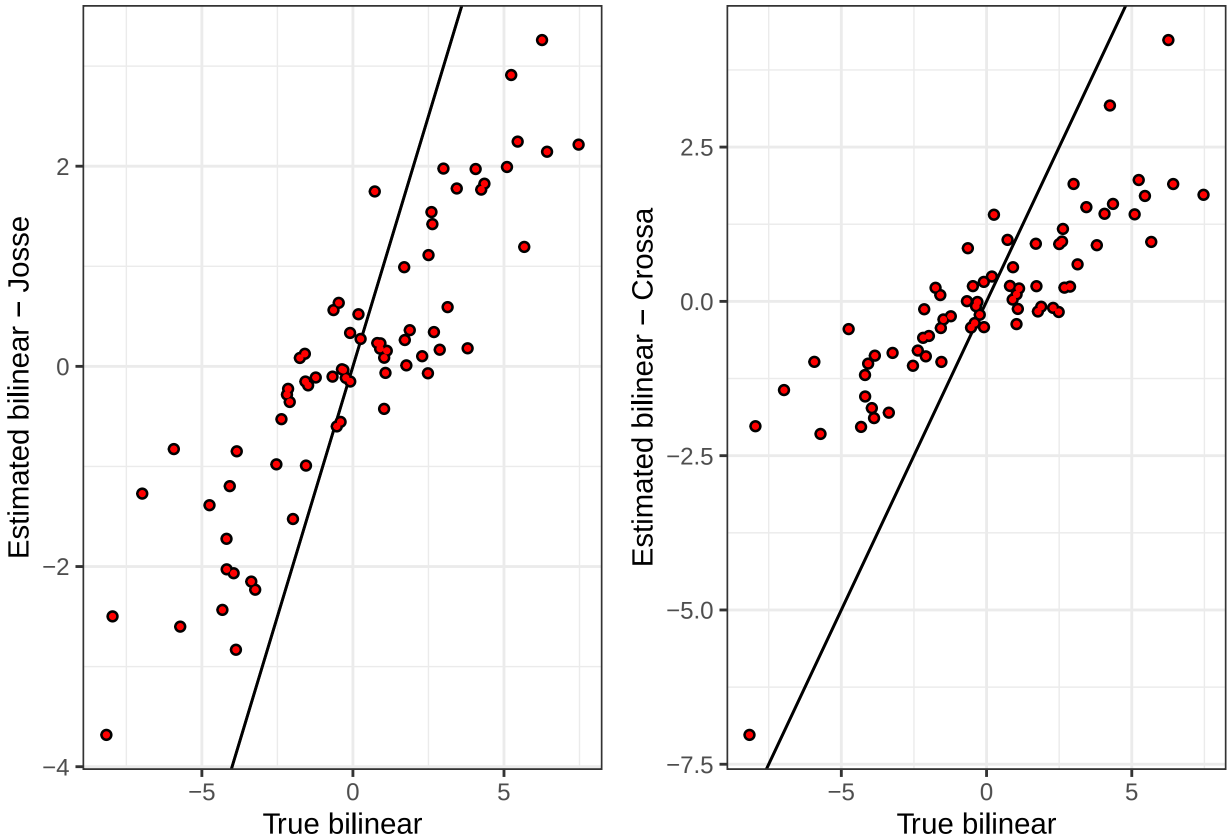


Estimated bilinear – Josse
True bilinear
Estimated bilinear – Crossa
True bilinear

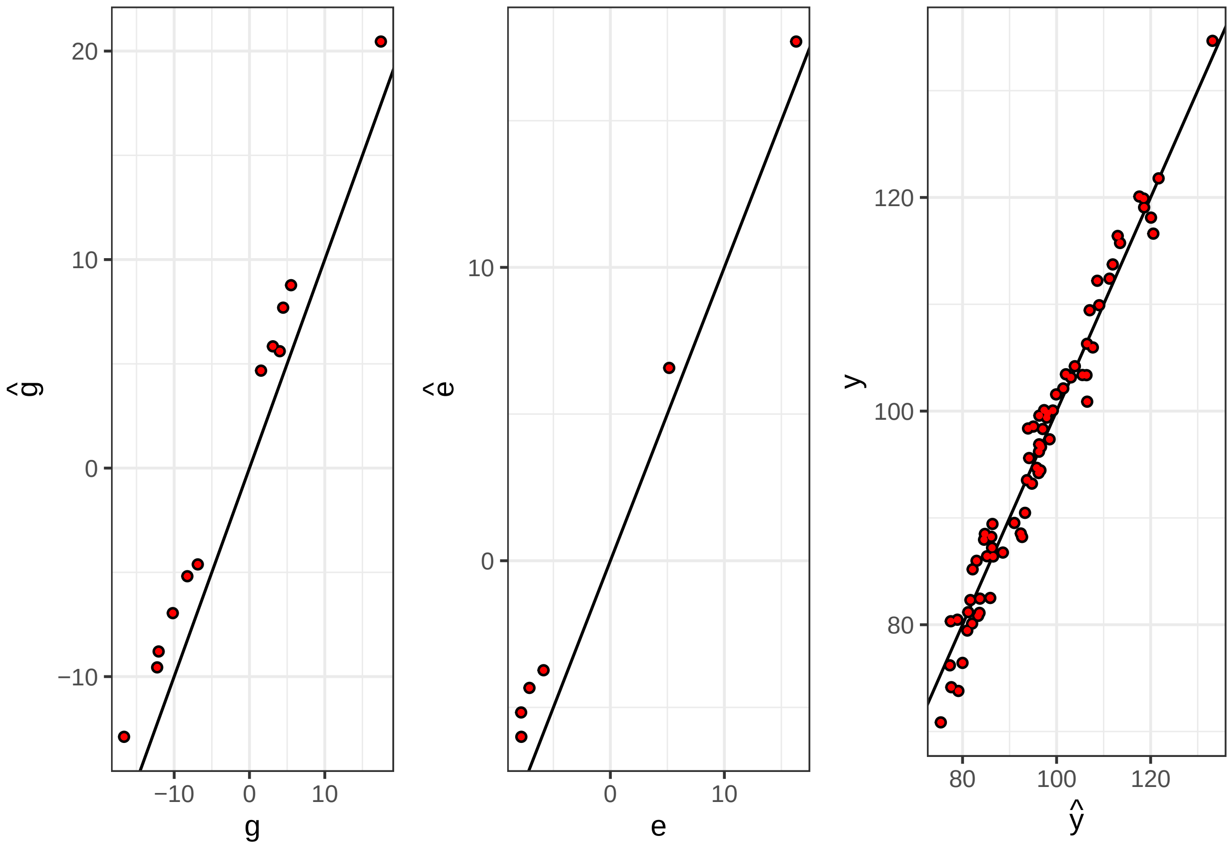


10
0
−10
−10
0
10
g
ĝ
10
0
−10
−10
0
10
e
ê
120
100
80
80
100
120
ŷ
y

1.2.6 Q = 3



Estimated bilinear − Josse
True bilinear
Estimated bilinear − Crossa
True bilinear



2 - BAMMIT Model

The BAMMIT model is formulated as

$$y_{ijk} = \mu + g_i + e_j + t_k + \sum_{q=1}^{Q} \lambda_q \gamma_{iq} \delta_{jq} \kappa_{kq} + \epsilon_{ijk},$$

where $\epsilon_{ijk} \sim N(0, \sigma_y^2)$, $i = 1, ..., I$, $j = 1, ..., J$ and $k = 1, ..., K$. We Assume that

$$\mu \sim N(\mu_\mu, \sigma_\mu^2)$$

$$g_i \sim N(\mu_g, \sigma_g^2)$$

$$e_j \sim N(\mu_e, \sigma_e^2)$$

$$t_k \sim N(\mu_t, \sigma_t^2)$$

$$\lambda_q \sim N^+(\mu_\lambda, \sigma_\lambda^2)$$

$$\sigma_y^2 \sim \Gamma(a, b)$$

For the parameters γ_{iq}, δ_{jq} and κ_{kq} we follow as before. Let

$$\theta_{iq}^{\gamma} \sim N(0, \sigma^{\theta}), i = 1, ..., I-1,$$

$$\theta_{Iq}^{\gamma} = -\sum_{i \neq I} \theta_{iq}^{\gamma}$$

$$\theta_{jq}^{\delta} \sim N(0, \sigma^{\theta}), j = 1, ..., J-1,$$

$$\theta_{Jq}^{\delta} = -\sum_{j \neq J} \theta_{jq}^{\delta};$$

$$\theta_{kq}^{\kappa} \sim N(0, \sigma^{\theta}), k = 1, ..., K-1,$$

$$\theta_{Kq}^{\kappa} = -\sum_{k \neq K} \theta_{kq}^{\kappa};$$

2.1 - Simulation

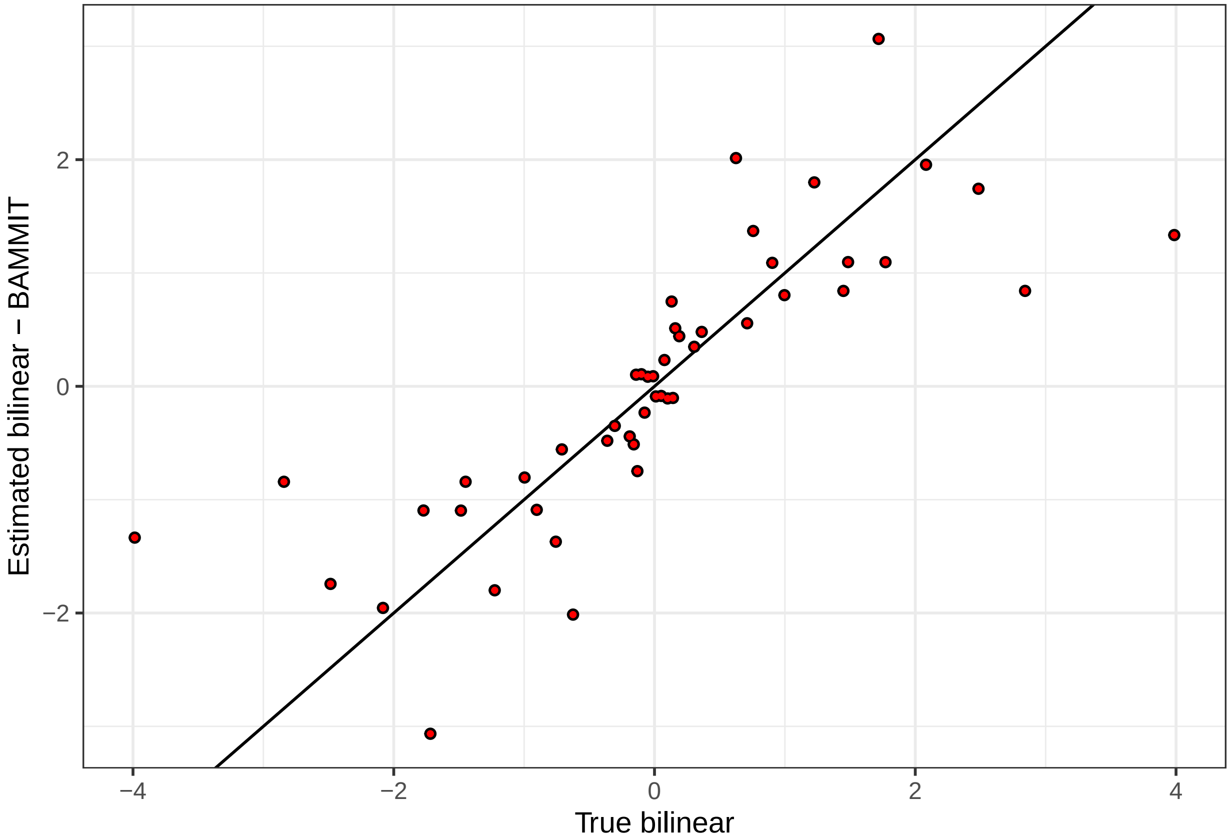
Simulation scenarios

- Set $I = \{6, 12\}$, $I = \{4, 6\}$, $K = \{2\}$ $Q = \{1\}$.
- Fix $\lambda = \{10\}$.
- Fix $\mu = 100$, $\sigma_g = 10$, $\sigma_e = 10$, $\sigma_t = 10$, $\sigma_y = 2$.

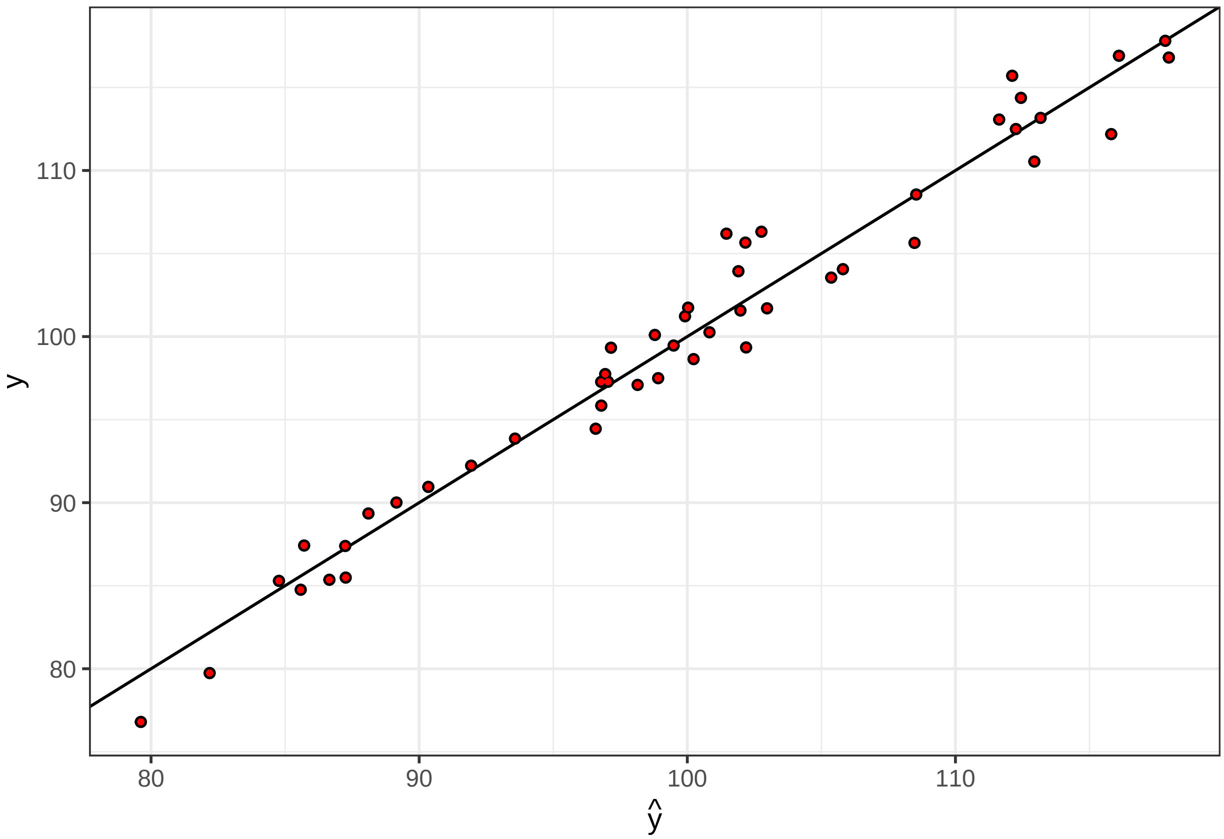
In Jags model:

- Fix $\sigma_\theta^2 = 1$.
- Fix $\sigma_\theta^2 = 100$.
- Fix $\mu_\lambda = 10$, $\sigma_\lambda^2 = 1$, $\mu_g = 0$, $\sigma_g^2 = 10$, $\mu_e = 0$, $\sigma_e^2 = 10$, $\mu_t = 0$, $\sigma_t^2 = 10$, $a = 0.1$, $b = 0.1$.

2.1.1 $\sigma_\theta^2 = 1$

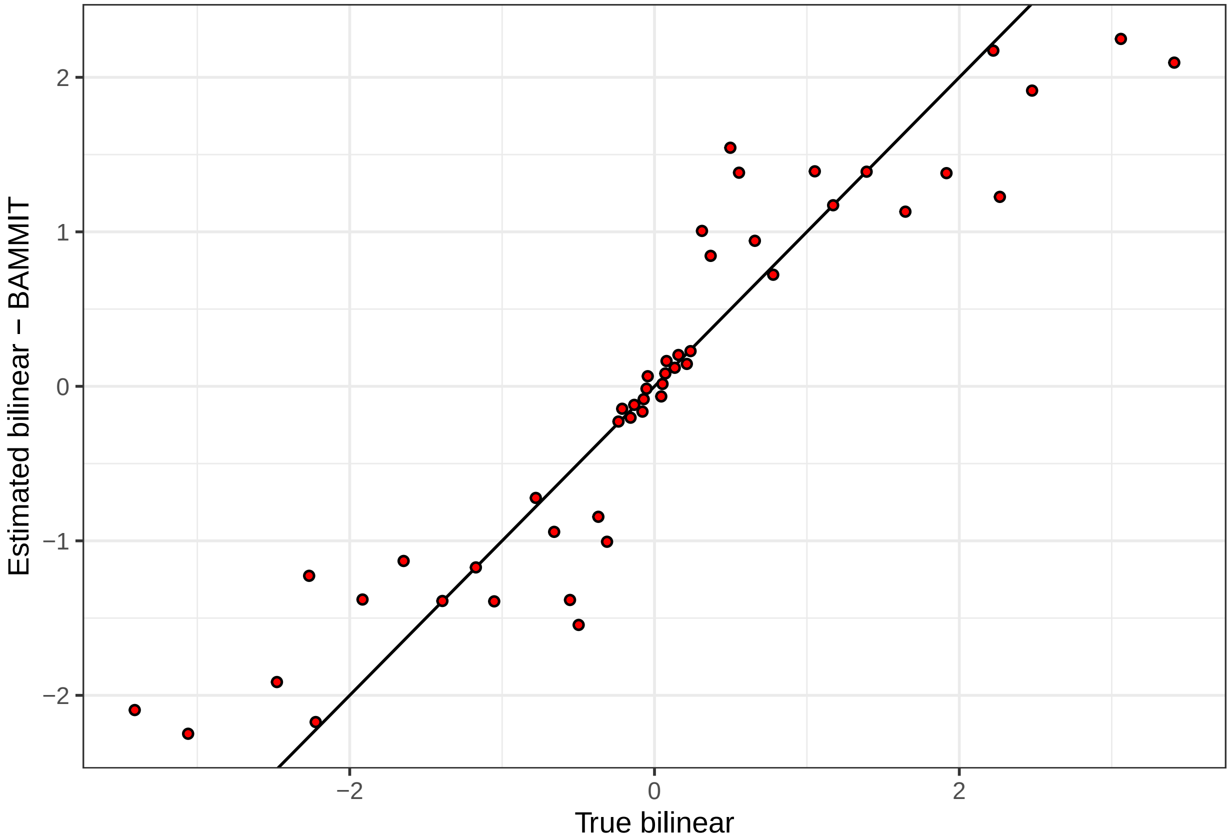



```
yhatB <- predictionBAMMIT(bammitModel, dataT)
qplot(yhatB, dataT$y, xlab = expression(hat(y)), ylab = "y") + geom_abline() +
  geom_abline() + theme_bw() + geom_point(colour = "red", size = 0.6)
```

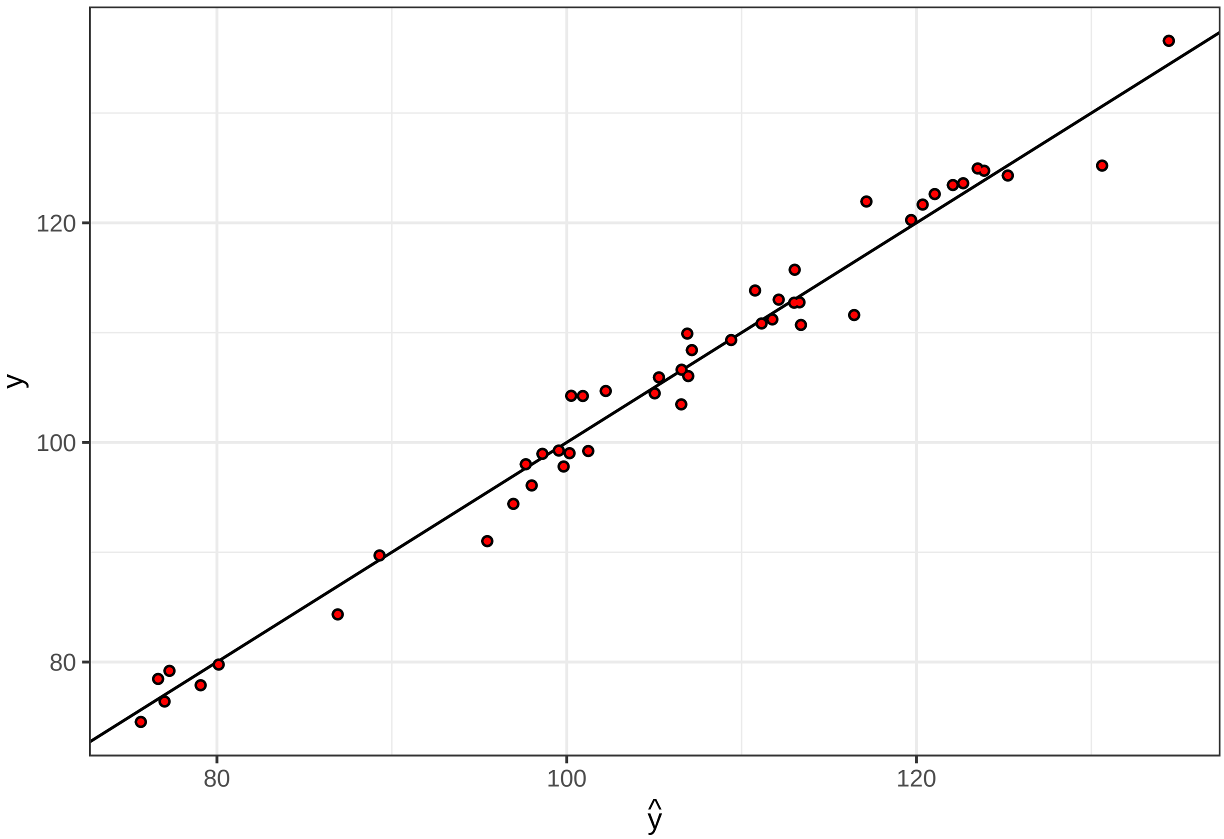


110
100
90
80
y
80
90
100
110
ŷ

2.1.2 $\sigma_\theta^2 = 100$




```
yhatB <- predictionBAMMIT(bammitModel, dataT)
qplot(yhatB, dataT$y, xlab = expression(hat(y)), ylab = "y") + geom_abline() +
  geom_abline() + theme_bw() + geom_point(colour = "red", size = 0.6)
```


120
100
80
y
80
100
120
ŷ