

# problema\_\_1.r

*lwiz*

*2020-02-11*

```
## Nesse script vamos gerar dados segundo uma binomial negativa e
## vamos ajustar alguns modelos aos dados e criticá-los

#####
# preparação
library(rstan)

## Loading required package: StanHeaders
## Loading required package: ggplot2
## rstan (Version 2.19.2, GitRev: 2e1f913d3ca3)

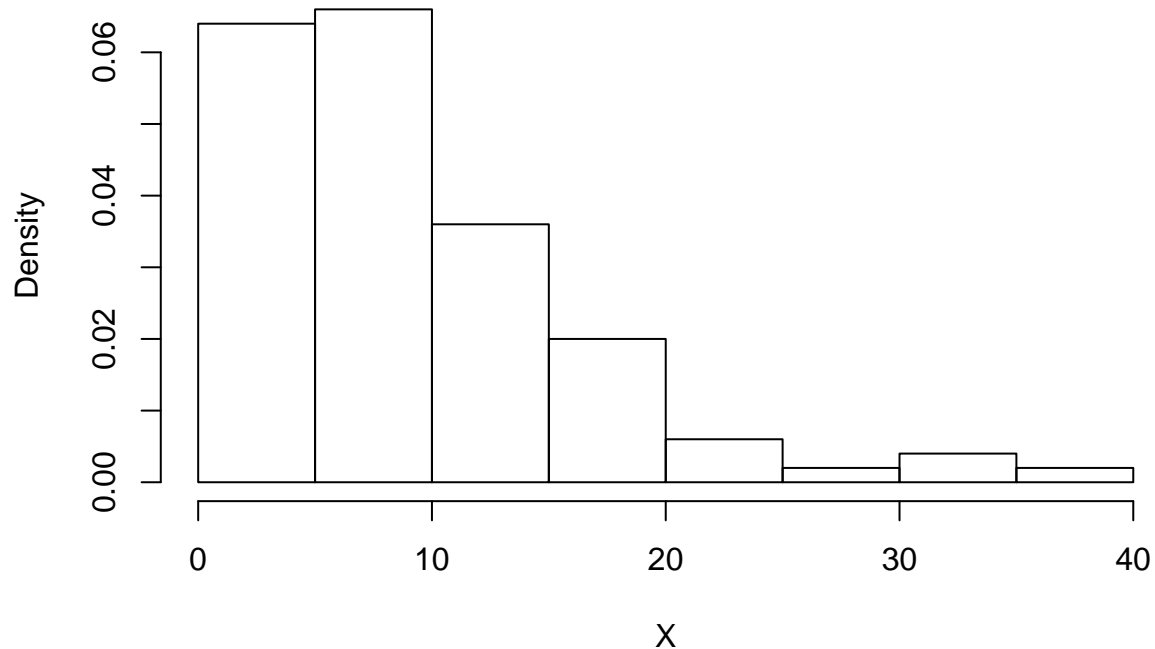
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)

rstan_options(auto_write = TRUE)
options(mc.cores = 4)

set.seed(666)

#####
# Simulando os dados
N <- 100
sfactor <- 5 ##  $Var(X) = sfactor * E[X]$ 
mu <- 10
X <- rnbinom(N, size = mu/(sfactor - 1), mu = mu)
hist(X, probability = TRUE, main = "Ovos de dragão (por macho)")
```

## Ovos de dragão (por macho)



```
mean(X)
```

```
## [1] 9.91
```

```
var(X)
```

```
## [1] 51.29485
```

```
dragonEgg.data <- list(N = N, x = X)
```

```
##
```

```
poisson.code = "
```

```
data{
```

```
  int<lower=1> N;
```

```
  int<lower=0> x[N];
```

```
}
```

```
parameters{
```

```
  real<lower=0> lambda;
```

```
}
```

```
model{
```

```
  target += poisson_lpmf(x | lambda);
```

```
  target += gamma_lpdf(lambda | 1, 1);
```

```
}
```

```
generated quantities{
```

```
  int<lower=0> x_rep[N];
```

```
  for(i in 1:N) x_rep[i] = poisson_rng(lambda);
```

```
}
```

```
"
```

```
fit.poisson <- stan(model_code = poisson.code, data = dragonEgg.data)
```

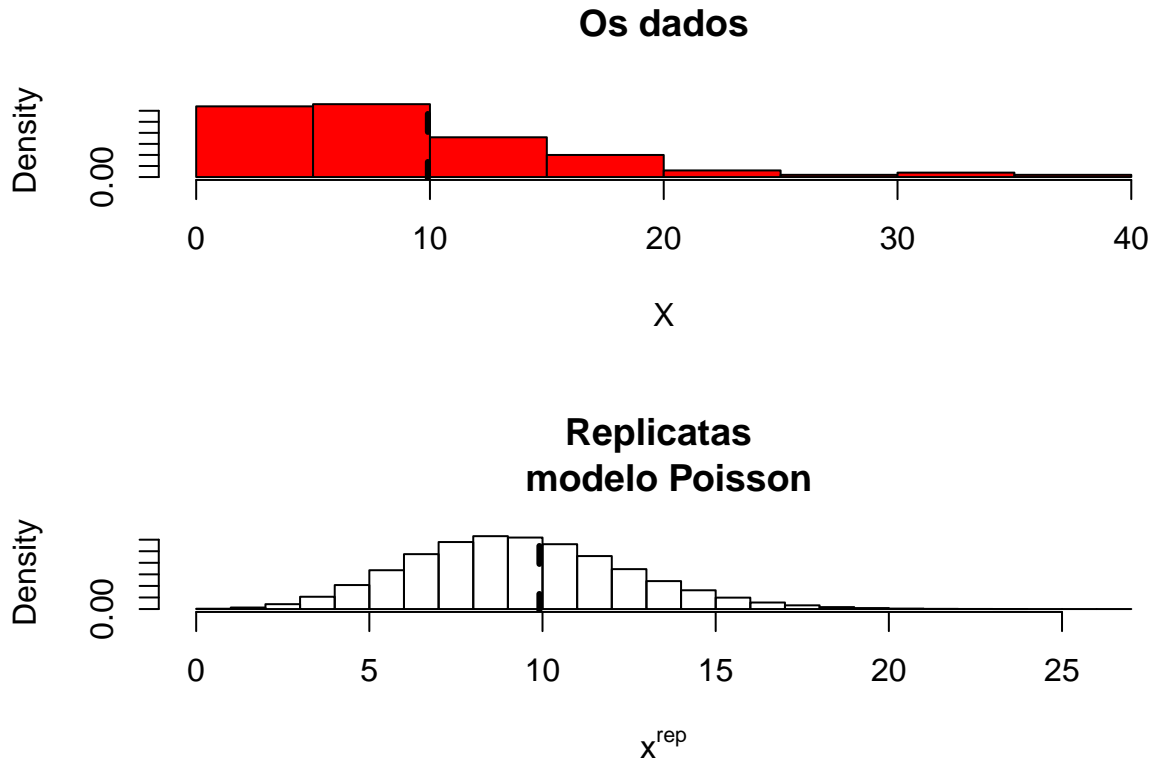
```
poisson.replicas <- extract(fit.poisson, 'x_rep')$x_rep
```

```

par(mfrow = c(2, 1))

hist(X, probability = TRUE, col = "red", main = "Os dados")
abline(v = mean(X), lwd = 3, lty = 2)
hist(as.vector(poisson.replicas), probability = TRUE,
     main = "Replicatas \n modelo Poisson", xlab = expression(x^{rep}))
abline(v = mean(X), lwd = 3, lty = 2)

```



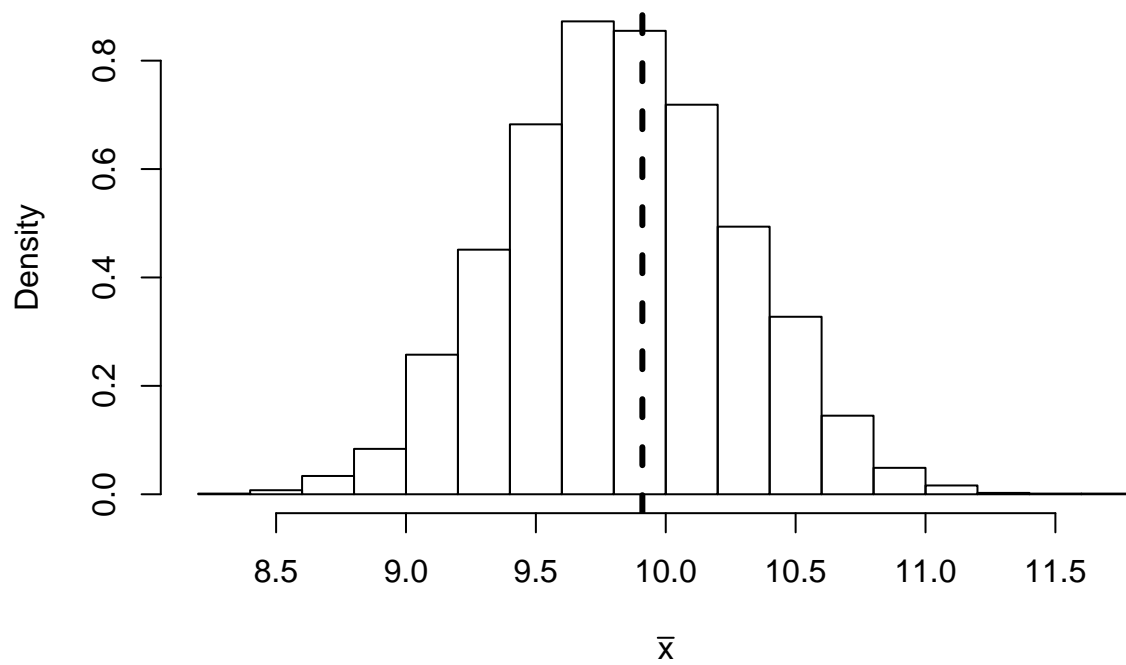
```

par(mfrow = c(1, 1))

# Como nosso modelo "prevê" a média?
medias.poisson <- rowMeans(poisson.replicas)
hist(medias.poisson, probability = TRUE, xlab = expression(bar(x)), main = "Médias \n modelo Poisson")
abline(v = mean(X), lwd = 3, lty = 2)

```

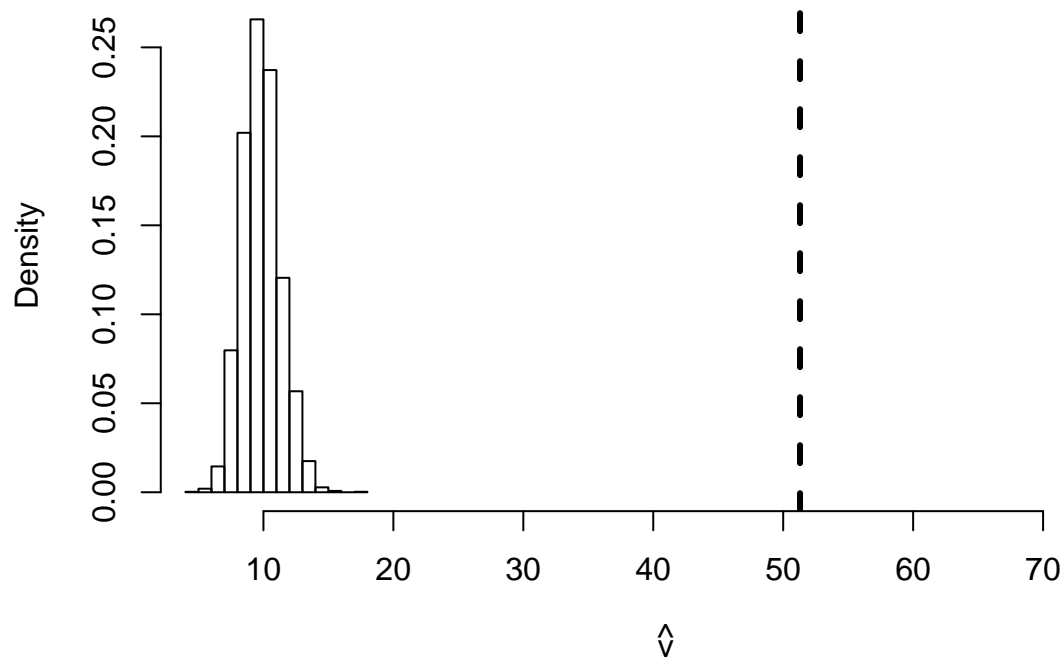
## Médias modelo Poisson



*# E a variância?*

```
variancias.poisson <- apply(poisson.replicas, 1, var)
hist(variancias.poisson, probability = TRUE,
     main = "Variâncias \n modelo Poisson", xlab = expression(hat(v)),
     xlim = c(min(variancias.poisson), 1.5*var(X)))
abline(v = var(X), lwd = 3, lty = 2)
```

## Variâncias modelo Poisson



*## Veredito: o modelo não captura corretamente a variabilidade dos dados*

*## Vamos tentar um modelo hierárquico, mais complexo*

```
poissonBinomial.code = "
data{
  int<lower=1> N;
  int<lower=0> x[N];
}
parameters{
  real<lower=0, upper=1> p;
  real<lower=0> lambda;
}
model{
  target += poisson_lpmf(x | lambda * p );
  target += beta_lpdf(p | 2, 2);
  target += gamma_lpdf(lambda | 1, 1);
}
generated quantities{
  int<lower=0> M_rep[N];
  int<lower=0> x_rep[N];
  for(i in 1:N){
    M_rep[i] = poisson_rng(lambda);
  }
  for(i in 1:N){
    x_rep[i] = binomial_rng(M_rep[i], p);
  }
}
"
```

```

fit.poissonBinomial <- stan(model_code = poissonBinomial.code, data = dragonEgg.data)

## Warning: There were 11 divergent transitions after warmup. Increasing adapt_delta above 0.8 may help
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: Examine the pairs() plot to diagnose sampling problems

print(fit.poissonBinomial, pars = c('p', 'lambda') )

## Inference for Stan model: 75082da6a3409d30038b5fcd936403f3.
## 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
## p             0.87    0.00 0.07 0.72  0.83  0.89  0.93  0.98   534 1.01
## lambda 11.29     0.05 1.04 9.83 10.55 11.09 11.83 13.76   500 1.01
##
## Samples were drawn using NUTS(diag_e) at Tue Feb 11 15:09:20 2020.
## 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).

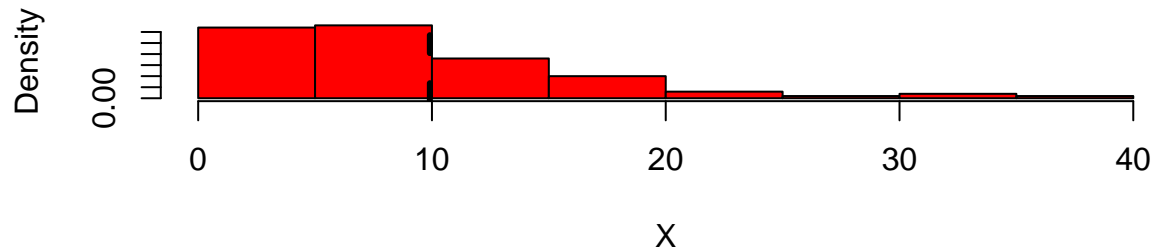
poissonBinomial.replicas <- extract(fit.poissonBinomial, 'x_rep')$x_rep

par(mfrow = c(2, 1))

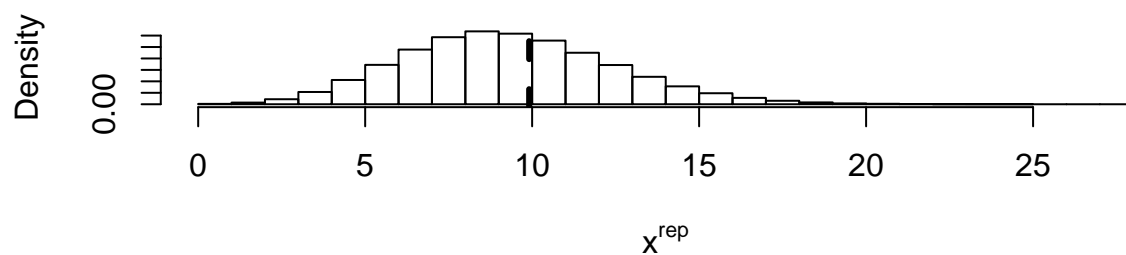
hist(X, probability = TRUE, col = "red", main = "Os dados")
abline(v = mean(X), lwd = 3, lty = 2)
hist(as.vector(poissonBinomial.replicas), xlab = expression(x^{rep}),
     probability = TRUE, main = "Replicatas \n modelo Poisson-binomial")
abline(v = mean(X), lwd = 3, lty = 2)

```

## Os dados



## Replicatas modelo Poisson-binomial



*# "Previsão" da média pelo modelo Poisson-binomial*

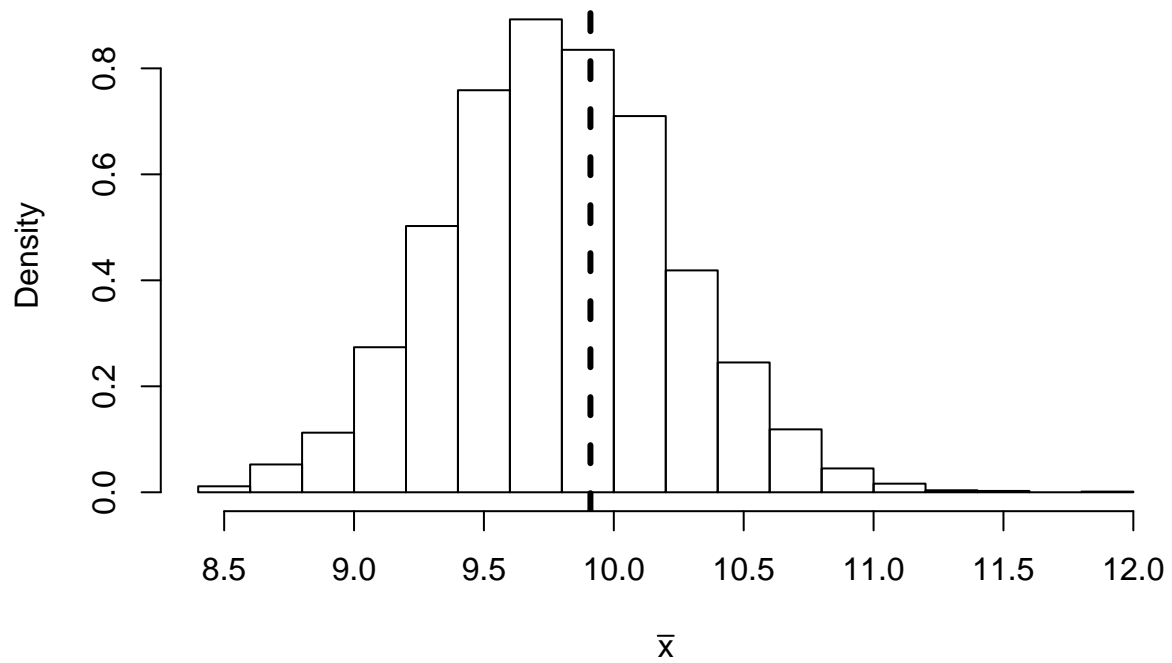
```
par(mfrow = c(1, 1))
```

```
medias.poissonBinomial <- rowMeans(poissonBinomial.replicas)
```

```
hist(medias.poissonBinomial, probability = TRUE,  
     xlab = expression(bar(x)), main = "Médias \n modelo Poisson-binomial")
```

```
abline(v = mean(X), lwd = 3, lty = 2)
```

## Médias modelo Poisson-binomial



*# Mais uma vez, vamos olhar a variância*

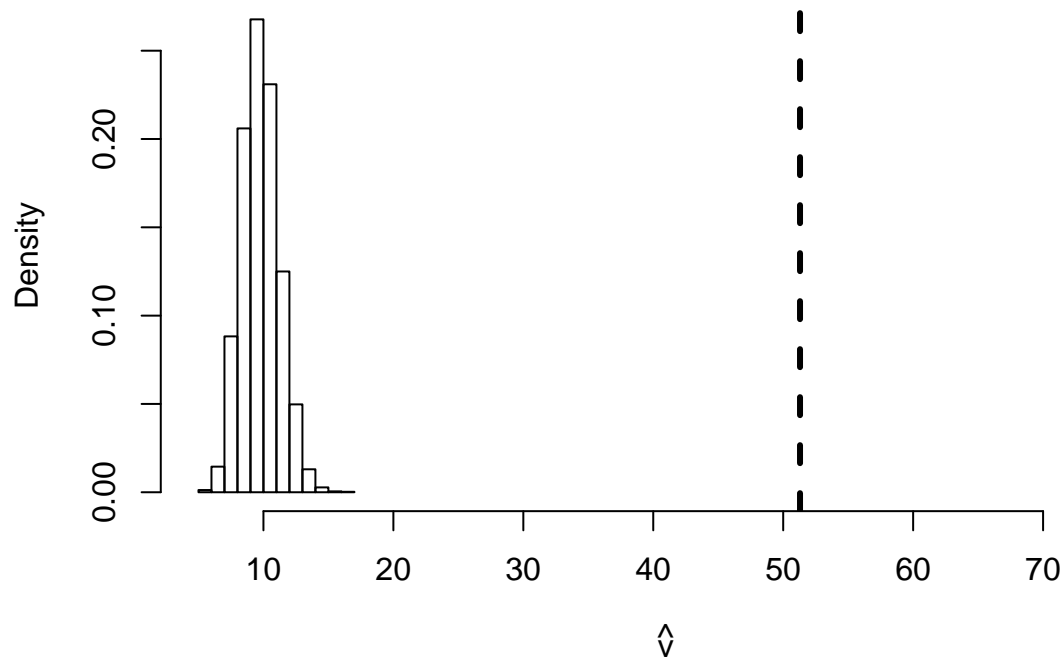
```

variancias.poissonBinomial <- apply(poissonBinomial.replicas, 1, var)
hist(variancias.poissonBinomial, probability = TRUE,
     main = "Variâncias \n modelo Poisson-binomial", xlab = expression(hat(v)),
     xlim = c(min(variancias.poisson), 1.5*var(X)))
abline(v = var(X), lwd = 3, lty = 2)

```



## Variâncias modelo Poisson-binomial



```
## Resultado: o modelo ainda é ruim.

# Vamos tentar agora um modelo com uma estrutura hierárquica diferente

poissonGamma.code = "
data{
  int<lower=1> N;
  int<lower=0> x[N];
}
parameters{
  real<lower=0> alpha;
  real<lower=0> beta;
  real<lower=0> lambda[N];
}
model{
  target += poisson_lpmf(x | lambda);
  target += gamma_lpdf(lambda | alpha, beta);
  target += exponential_lpdf(alpha | 1);
  target += exponential_lpdf(beta | 1);
}
generated quantities{
  int<lower=0> x_rep[N];
  for(i in 1:N){
    x_rep[i] = poisson_rng(lambda[i]);
  }
}
"

fit.poissonGamma <- stan(model_code = poissonGamma.code, data = dragonEgg.data)
```

```

print(fit.poissonGamma, pars = c('alpha', 'beta'))

## Inference for Stan model: 3fbe14727025396d8c9001d7e92d432d.
## 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
## alpha 2.70      0.01 0.48 1.91 2.37 2.65 2.98 3.79 1352    1
## beta  0.27      0.00 0.05 0.19 0.24 0.27 0.30 0.39 1436    1
##
## Samples were drawn using NUTS(diag_e) at Tue Feb 11 15:10:25 2020.
## 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).

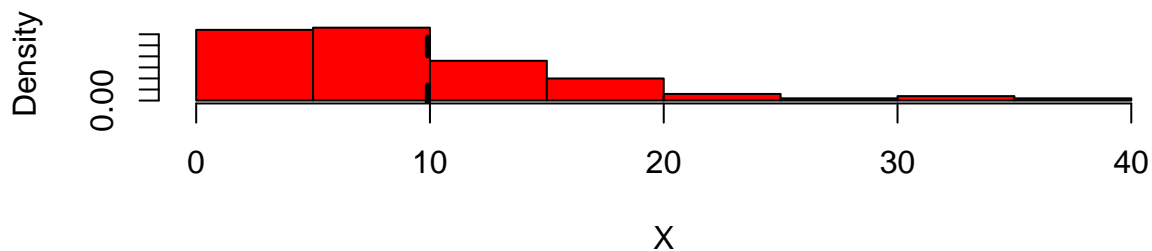
poissonGamma.replicas <- extract(fit.poissonGamma, 'x_rep')$x_rep

par(mfrow = c(2, 1))

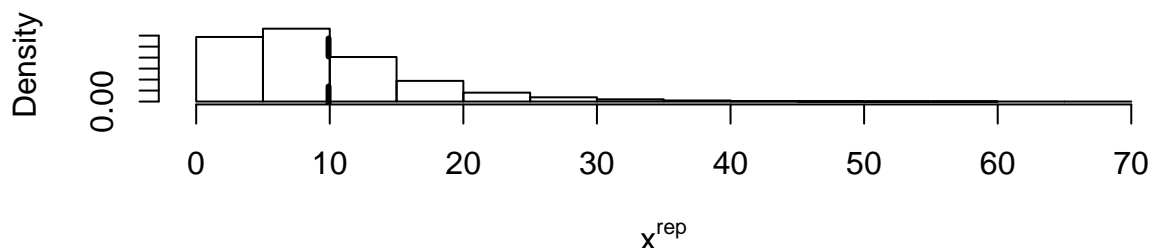
hist(X, probability = TRUE, col = "red", main = "Os dados")
abline(v = mean(X), lwd = 3, lty = 2)
hist(as.vector(poissonGamma.replicas), xlab = expression(x^{rep}),
     probability = TRUE, main = "Replicatas \n modelo Poisson-gama")
abline(v = mean(X), lwd = 3, lty = 2)

```

### Os dados



### Replicatas modelo Poisson-gama



```

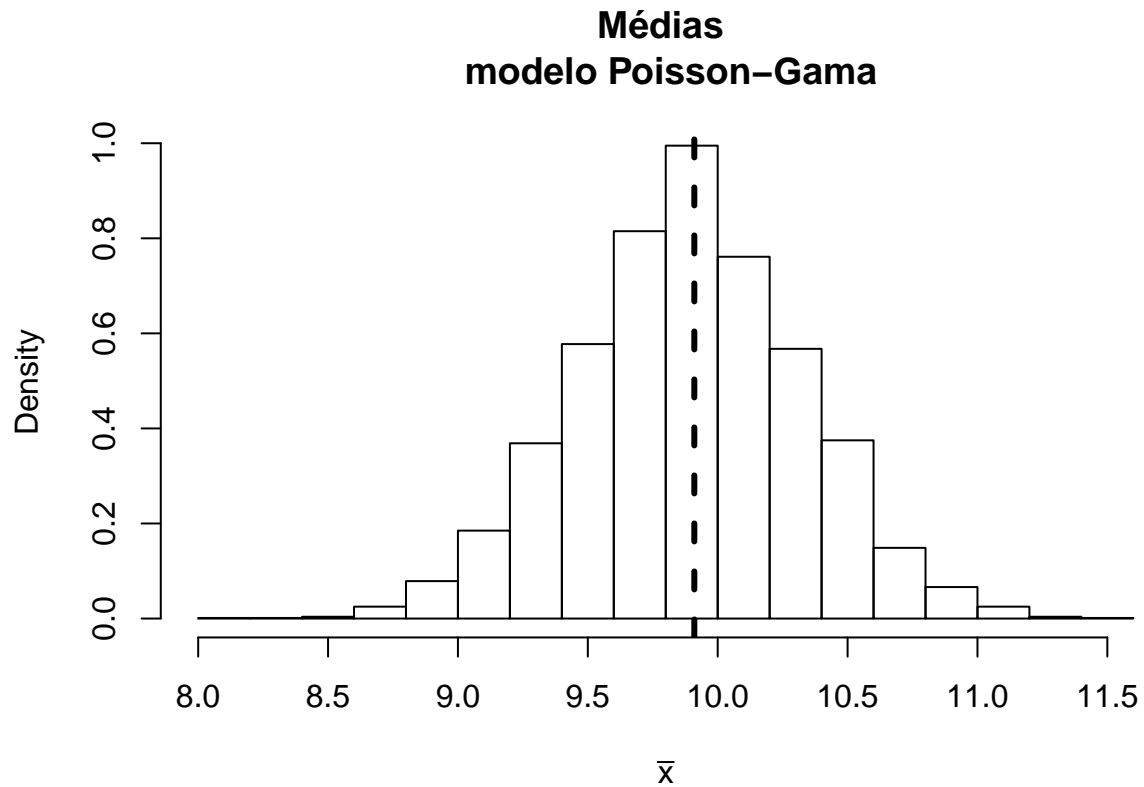
# "Previsão" da média pelo modelo Poisson-Gamma

par(mfrow = c(1, 1))

medias.poissonGamma <- rowMeans(poissonGamma.replicas)

```

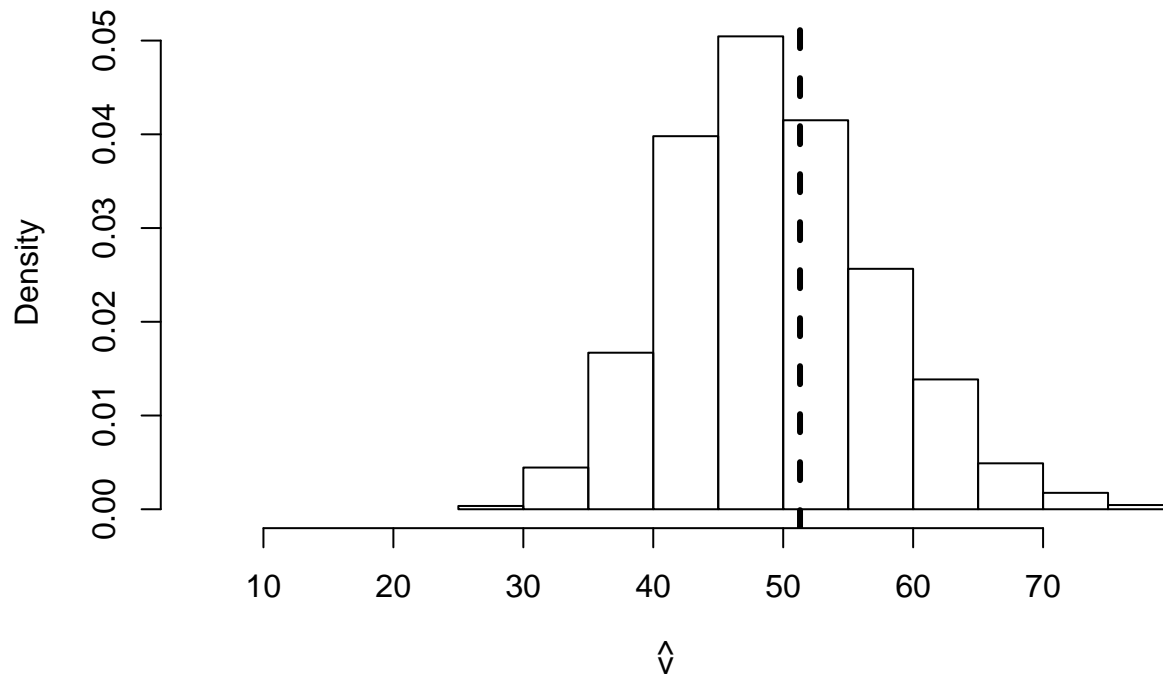
```
hist(medias.poissonGamma, probability = TRUE,
     xlab = expression(bar(x)), main = "Médias \n modelo Poisson-Gama")
abline(v = mean(X), lwd = 3, lty = 2)
```



```
# Mais uma vez, vamos olhar a variância

varancias.poissonGamma <- apply(poissonGamma.replicas, 1, var)
hist(varancias.poissonGamma, probability = TRUE, xlab = expression(hat(v)) ,
     main = "Variâncias \n modelo Poisson-Gama", xlim = c(min(varancias.poisson), 1.5*var(X)))
abline(v = var(X), lwd = 3, lty = 2)
```

## Variâncias modelo Poisson-Gama



```
# Intervalos
quantile(as.vector(poisson.replicas), probs = c(.025, .975))

## 2.5% 97.5%
##      4      16

quantile(as.vector(poissonBinomial.replicas), probs = c(.025, .975))

## 2.5% 97.5%
##      4      16

quantile(as.vector(poissonGamma.replicas), probs = c(.025, .975))

## 2.5% 97.5%
##      1      28

qnbinom(p = c(.025, .975), size = mu/(sfactor - 1), mu = mu)

## [1] 1 28
```