

Exercise 1

question 1

(a)

```
dinvgamma <- function(x, alpha, beta){ return((beta^alpha / gamma(alpha)) * x^(-alpha - 1) * exp(-beta / x)) * (x >= 0)) }
```

(b)

```
pinvgamma <- function(x, alpha, beta){ return(ifelse(x > 0, 1 - pgamma(1/x, alpha, beta), 0)) }
```

(c)

```
qinvgamma <- function(p, alpha, beta){ return(1 / qgamma(1-p, alpha, beta)) }
```

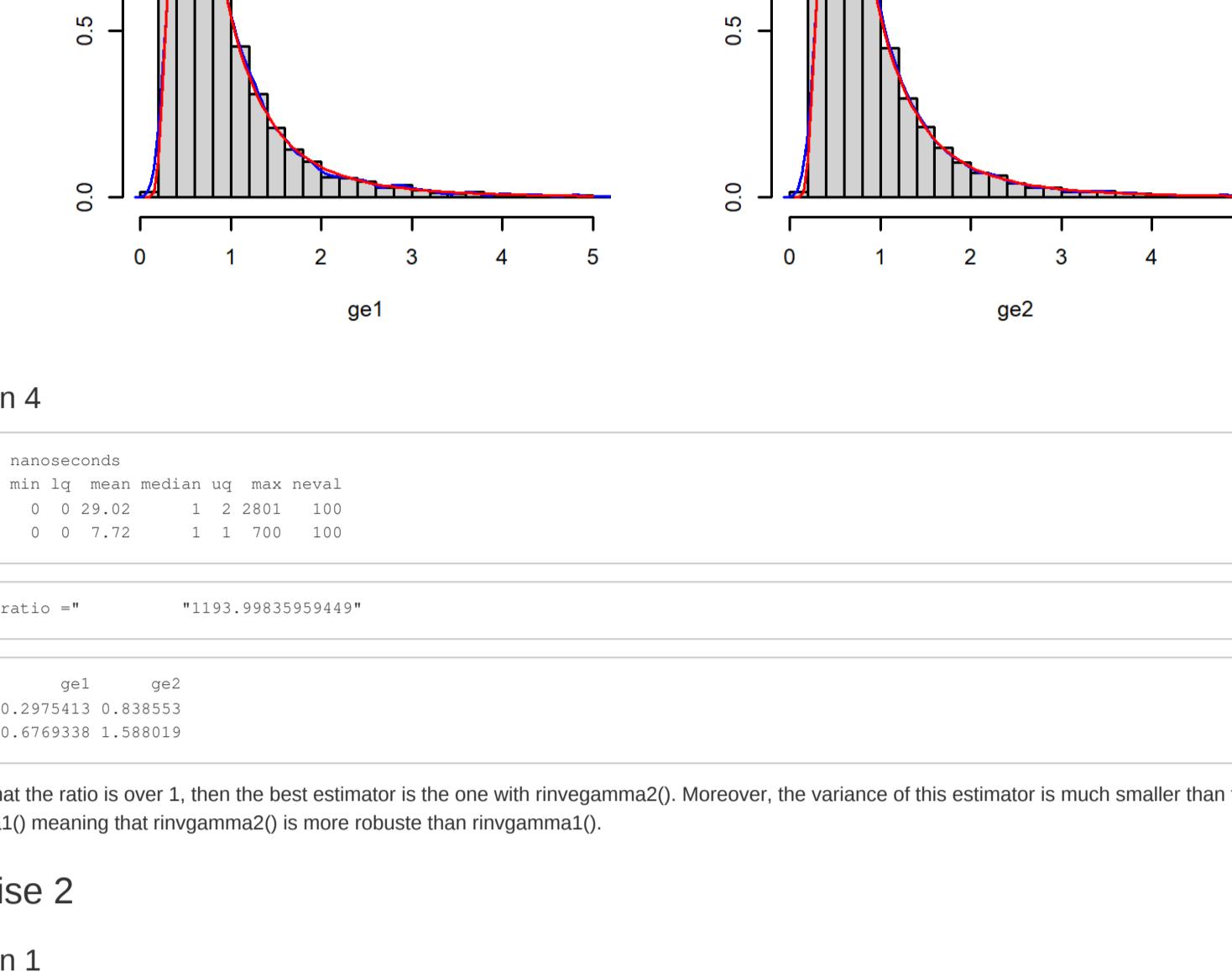
(d)

```
rinvgamma <- function(n, alpha, beta){ return(1 / rgamma(n, alpha, beta)) }
```

question 2

```
rinvgamma2 <- function(n, alpha, beta){ return(qinvgamma(runif(n, 0, 1), alpha, beta)) }
```

question 3



question 4

```
## Unit: nanoseconds  
## expr min lq mean median uq max neval  
## ge1 0 0 23.02 1 2 2801 100  
## ge2 0 0 7.72 1 1 700 100
```

```
## [1] "ratio =" *1193.99835959449"
```

```
## gel ge2  
## mean 0.2975413 0.838553  
## var 0.6769338 1.588019
```

We have that the ratio is over 1, then the best estimator is the one with rinvgamma2(). Moreover, the variance of this estimator is much smaller than the variance of rinvgamma1() meaning that rinvgamma2() is more robust than rinvgamma1().

Exercise 2

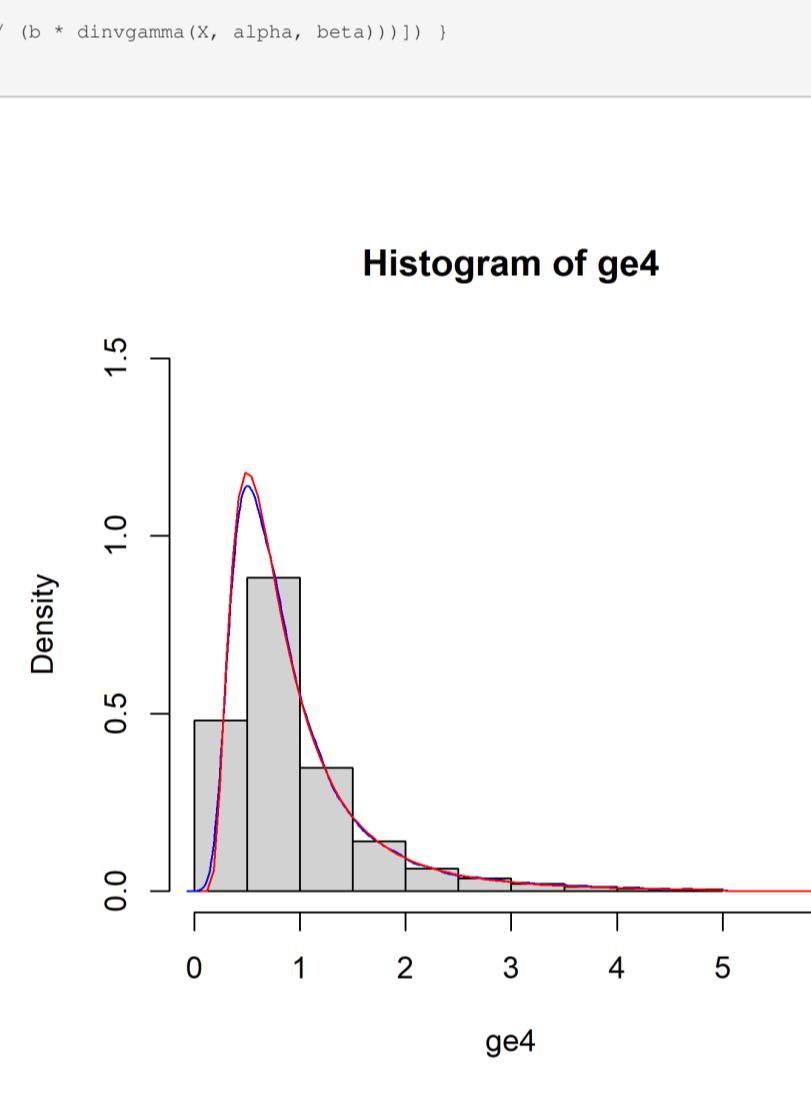
question 1

```
dinvgamma_trunc <- function(x, alpha, beta, b){ return(dinvgamma(x, alpha, beta) * (0 <= x & x <= b) / (pinvgamma(b, alpha, beta) - pinvgamma(0, alpha, beta)) * (0 <= x & x <= b)) }
```

question 2

```
rinvgamma_trunc1 <- function(n, alpha, beta, b){ return(qinvgamma(pinvgamma(0, alpha, beta) + runif(n, 0, 1) * (pinvgamma(b, alpha, beta) - pinvgamma(0, alpha, beta)), alpha, beta)) }
```

question 3



question 4

(a)

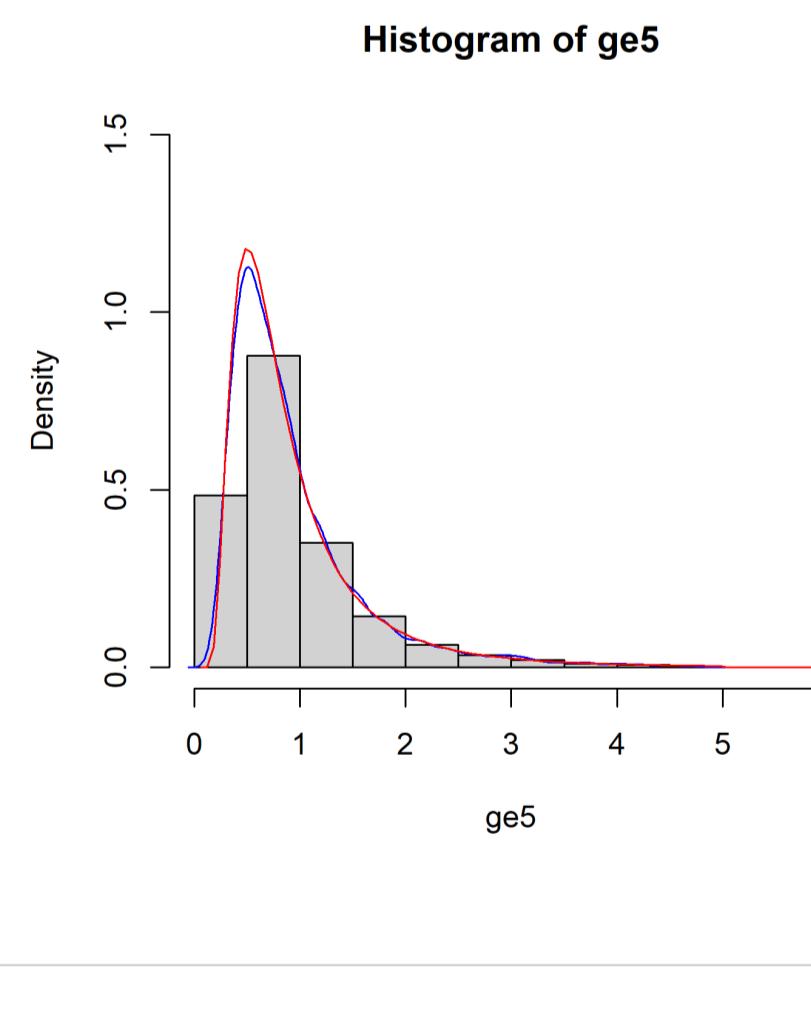
If $f(x) = f_{\alpha,\beta}(x)1_{(0,b)}(x)$ and $g(x) = f_{\alpha,\beta}(x)$. Thus $M = \sup_x \frac{f(x)}{g(x)} = b$

(b)

```
rinvgamma_trunc2 <- function(n, alpha, beta, b){  
  x <- numeric(0)  
  f <- function(x,b){ return(dinvgamma(x, alpha, beta) * (x <= b)) }  
  while(length(x) < n){  
    U <- runif(1)  
    X <- rinvgammln(1, alpha, beta)  
    x <- append(x, X[U <= (f(X,b) / (b * dinvgamma(X, alpha, beta)))])  
  }  
  return(x)}
```

(c)

Histogram of ge4



question 5

(a)

Here the uniform distribution is of parameters $(0, b)$ as we $x \in (0, b)$

(b)

we have $f(x) = f_{\alpha,\beta}(x)1_{(0,b)}(x)$ and $g(x) = \frac{1}{b}1_{(0,b)}(x)$. Thus $M = \sup_x \frac{f(x)}{g(x)} = b \max_x f_{\alpha,\beta}(x) = b f_{\alpha,\beta}\left(\frac{\beta}{\alpha+1}\right)$

(c)

```
rinvgamma_trunc3 <- function(n, alpha, beta, b){  
  x_max <- beta / (alpha + 1)  
  M <- b * dinvgamma(x_max, alpha, beta)  
  x <- numeric(0)  
  f <- function(x,b){ return(dinvgamma(x, alpha, beta) * (x <= b)) }  
  g <- function(x,b){ return((1/b) * (0 <= x & x <= b)) }  
  while(length(x) < n){  
    U <- runif(1)  
    X <- runif(1, 0, b)  
    x <- append(x, X[U <= (f(X,b) / (M * g(X,b))))})  
  }  
  return(x)}
```

(d)

Histogram of ge5



(e)

```
## ratio  
## b = 5 1.01237  
## b = 0,5 NA
```

```
## mean 0.9439541 0.9398790 0.384243658 0.385317849  
## var 0.4447986 0.4393638 0.005902655 0.005860143
```

Exercise 3

question 1

```
estim_mc_invgamma <- function(n, sample_y, alpha, beta, b){  
  hs <- (1 - sample_y^2) * as.numeric(sample_y <= b)  
  return(c(mean(hs), var(hs)))}
```

question 2

```
estim_mc_invgamma_trunc <- function(n, sample_y, alpha, beta){  
  hs <- 1 - sample_y^2  
  return(c(mean(hs), var(hs)))}
```

question 3

```
## mc_1 mc_2 mc_trunc_1 mc_trunc_2 mc_trunc_3  
## mean -0.3613066 -0.3567347 -0.318916 -0.3496139 -0.3715746  
## var 6.3090497 5.9026857 5.708490 6.0551452 6.3617217
```

Unit: nanoseconds

expr min lq mean median uq max neval

mc_1 0 1 34.15 1.0 100 801 100

mc_2 0 1 59.97 1.0 100 3001 100

mc_trunc_1 0 1 43.00 1.5 101 501 100

mc_trunc_2 0 1 47.04 1.0 51 1901 100

mc_trunc_3 0 1 36.96 1.0 51 1201 100

mc_1 mc_2 mc_trunc_1 mc_trunc_2 mc_trunc_3

mean 2.670775 NA NA NA

mc_2 6.271119 3.096899 NA NA

mc_trunc_1 3126.837996 1170.760630 378.0429 NA

mc_trunc_2 2976.153136 1114.340725 359.8247 0.9518092

We can see that the better estimator is mc_trunc_2() the Monte-Carlo estimator base on a $InvGamma_{[0,b]}(\alpha, \beta)$ or mc_trunc_3() the Monte-Carlo estimator base on a $\mathcal{U}([0, b])$ depending on the randomness of the microbenchmark function. This estimator has the smallest variance with a ratio with the other estimator always letting us choose it (greater or equal than 1).

mc_1 mc_2 mc_trunc_1 mc_trunc_2 mc_trunc_3

mean 0.2018811 0.2009180 0.846717131 0.846873132 0.845478987

var 0.0310126 0.1303823 0.003160534 0.003200629 0.003223051

Unit: nanoseconds

expr min lq mean median uq max neval

mc_1 0 1 10.97 1 1.5 2101 100

mc_2 0 1 25.04 1 1.5 2101 100

mc_trunc_1 0 0 8.88 1 1.0 701 100

mc_trunc_2 0 0 27.83 1 1.0 2201 100

mc_trunc_3 0 0 27.83 1 1.0 2201 100

Here the variance of our estimator are so close to 0 that the ratio will not be relevant, as we divide something really close to 0 by something really close to 0 it's undetermined. But we can argue with the matrix displaying all the mean-var that the best estimator is also mc_trunc_2() or mc_trunc_3() with the same argument as previously.