Homework 2 Part 1

Pol Ribó León (1840853)

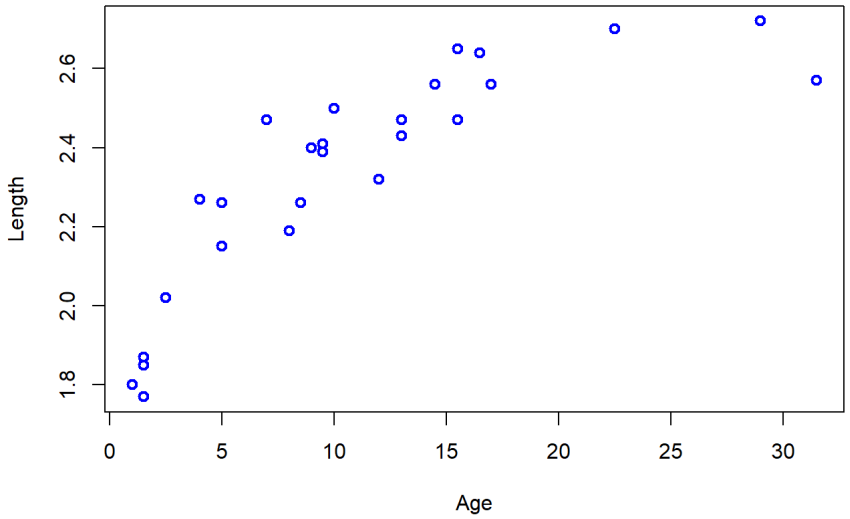
25 de junio de 2019

```
library(invgamma)
library(corrplot)
```

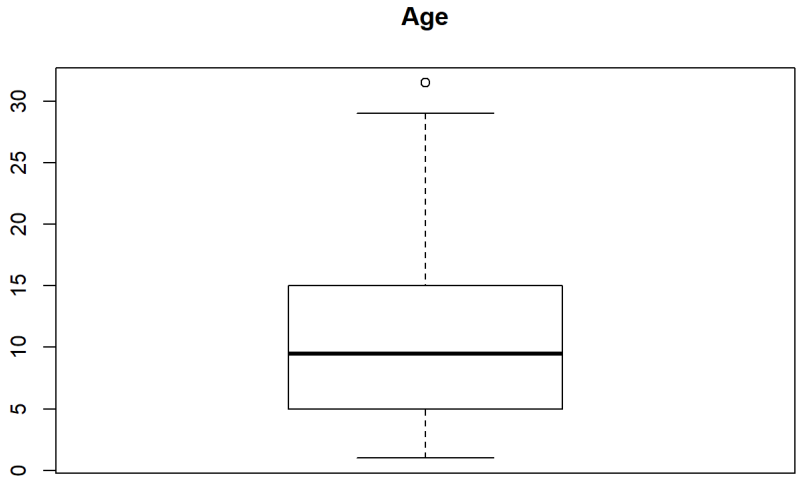
```
## corrplot 0.84 loaded
```

NOTE: Due to problems when knitting, I have been forced to plug-in the variables x and y manually.Nevertheless, the work has been carried out with the standard code for reading files, written below with comments.

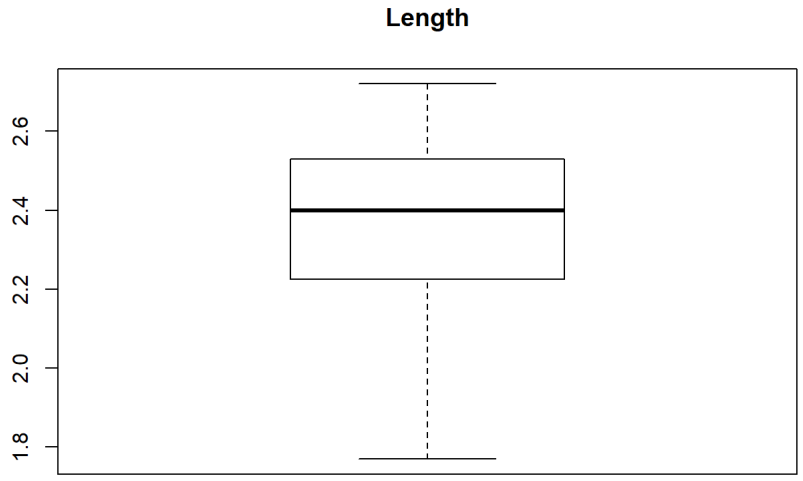
```
#data <- read.table("C:/Users/HP/Documents/dugong-data.txt", header = TRUE)

#x <- my_data$Age
#y <- my_data$Length

#Age
x = c( 1.0, 1.5, 1.5, 1.5, 2.5, 4.0, 5.0, 5.0, 7.0, 8.0, 8.5,
       9.0, 9.5, 9.5, 10.0, 12.0, 12.0, 13.0, 13.0, 14.5, 15.5,
       15.5, 16.5, 17.0, 22.5, 29.0, 31.5)

#Length
y = c(1.80, 1.85, 1.87, 1.77, 2.02, 2.27, 2.15, 2.26, 2.47, 2.19,
      2.26, 2.40, 2.39, 2.41, 2.50, 2.32, 2.32, 2.43, 2.47, 2.56,
      2.65, 2.47, 2.64, 2.56, 2.70, 2.72, 2.57)

#number of observations
n <- length(x)

plot(x,y,lwd=2,col="blue",xlab="Age",ylab="Length")
```




```
boxplot(x, main="Age")
```



```
boxplot(y, main="Length")
```



To highlight, comment that there is only one outlier appearing, in Age, corresponding of an old dugong.

1b) Derive the corresponding likelihood function

$$\prod_{i=1}^{n} \frac{1}{\tau\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{\left(Y_i - \alpha + \beta\gamma^{x_i}\right)^2}{\tau^2}\right)$$

$$= \left(\frac{1}{\tau\sqrt{2\pi}}\right)^n \exp\left(\sum_{i=1}^{n} -\frac{1}{2}\frac{\left(Y_i - \alpha + \beta\gamma^{x_i}\right)^2}{\tau^2}\right)$$

$$= \frac{1}{(\tau\sqrt{2\pi})^n} \exp\left(-\frac{1}{2\tau^2}\sum_{i=1}^{n}\left(Y_i - \alpha + \beta\gamma^{x_i}\right)^2\right) I_{(1,\infty)}(\alpha) I_{(1,\infty)}(\beta) I_{(0,1)}(\gamma) I_{(0,\infty)}(\tau^2)$$

1c) Write down the expression of the joint prior distribution of the parameters at stake and illustrate your suitable choice for the hyperparameters

Assuming that the parameters are independent, the joint prior distribution can be found as:

$$\pi(\theta) = \pi(\alpha)\pi(\beta)\pi(\gamma)\pi\left(\tau^2\right)$$

And the priors for each parameter are:

$$\alpha \quad prior: \pi(\alpha) = \frac{1}{\sigma_\alpha \sqrt{2\pi}} \exp\left(-\frac{\alpha^2}{2\sigma_\alpha^2}\right) I_{(1,\infty)}(\alpha)$$

$$\beta \quad prior: \pi(\beta) = \frac{1}{\sigma_\beta \sqrt{2\pi}} \exp\left(-\frac{\beta^2}{2\sigma_\beta^2}\right) I_{(1,\infty)}(\beta)$$

$$\gamma \quad prior: \pi(\gamma) = I_{(0,1)}(\gamma)$$

$$\tau \quad prior: \pi\left(\tau^2\right) = \frac{b^a}{\Gamma(a)} \tau^{2(-a-1)} \exp\left(\frac{-b}{\tau^2}\right) I_{(0,\infty)}\left(\tau^2\right)$$

Hence, the joint prior distribution is as follows:

$$\pi(\theta) = \frac{1}{\sigma_\alpha \sqrt{2\pi}} \exp\left(-\frac{\alpha^2}{2\sigma_\alpha^2}\right) I_{(1,\infty)}(\alpha) \frac{1}{\sigma_\beta \sqrt{2\pi}} \exp\left(-\frac{\beta^2}{2\sigma_\beta^2}\right) I_{(1,\infty)}(\beta) I_{(0,1)}(\gamma) \frac{b^a}{\Gamma(a)} \tau^{2(-a-1)} \exp\left(\frac{-b}{\tau^2}\right) I_{(0,\infty)}\left(\tau^2\right)$$

The choice for the hyperparameters is:

$$\sigma_\alpha = 10000$$
$$\sigma_\beta = 10000$$
$$a = 0.001$$
$$b = 0.001$$

```
sigma_alpha <- 10000
sigma_beta <- 10000
a <- 0.001
b <- 0.001
```

1d) Derive the functional form (up to proportionality constants) of all full-conditionals

Given

$$\pi\left(\alpha, \beta, \gamma, \tau^2 \mid Y\right) \propto \pi\left(Y, \alpha, \beta, \gamma, \tau^2\right)$$

$$** Full \quad conditional \quad for \quad \alpha ** : \pi\left(\alpha \mid \beta, \gamma, \tau^2, x, y\right) = \frac{1}{\sigma_\alpha \sqrt{2\pi}} \exp\left(-\frac{\alpha^2}{2\sigma_\alpha^2}\right) \frac{1}{(\tau\sqrt{2\pi})^n} \exp\left(-\frac{1}{2\tau^2} \sum_{i=1}^{n} \left(Y_i - \alpha + \beta\gamma^{x_i}\right)^2\right) = \frac{1}{\sigma_\alpha \sqrt{2\pi}(\tau\sqrt{2\pi})^n} \exp\left(-\frac{\alpha^2}{2\sigma_\alpha^2} - \frac{1}{2\tau^2} \sum_{i=1}^{n} Y_i^2 - 2Y_i\alpha - 2\alpha\beta\gamma^{x_i} + \alpha^2 - 2Y_i\beta\gamma^{x_i} + \beta^2\gamma^{2x_i}\right) \propto \exp\left(-\frac{\alpha^2}{2\sigma_\alpha^2} -$$

Note: TN=Truncated Normal distribution as its domain is “truncated”. (1,+ inf)

$$** Full \quad conditional \quad for \quad \beta ** : \pi\left(\beta \mid \alpha, \gamma, \tau^2, x, y\right) = \frac{1}{\sigma_\beta \sqrt{2\pi}} \exp\left(-\frac{\beta^2}{2\sigma_\beta^2}\right) \frac{1}{(\tau\sqrt{2\pi})^n} \exp\left(-\frac{1}{2\tau^2} \sum_{i=1}^{n} \left(Y_i - \alpha + \beta\gamma^{x_i}\right)^2\right) = \frac{1}{\sigma_\beta \sqrt{2\pi}(\tau\sqrt{2\pi})^n} \exp\left(-\frac{\beta^2}{2\sigma_\beta^2} - \frac{1}{2\tau^2} \sum_{i=1}^{n} Y_i^2 - 2Y_i\alpha - 2\alpha\beta\gamma^{x_i} + \alpha - 2Y_i\beta\gamma^{x_i} + \beta^2\gamma^{2x_i}\right) \propto \exp\left(-\frac{\beta^2}{2\sigma_\beta^2} -\right.$$

Note: TN=Truncated Normal distribution as its domain is “truncated”. (1,+ inf)\

$$**Full \quad conditional \quad for \quad \gamma** : \pi\left(\gamma \mid \alpha, \beta, \tau^2, x, y\right) = \frac{1}{(\tau\sqrt{2\pi})^n} \exp\left(-\frac{1}{2\tau^2}\sum_{i=1}^{n}\left(Y_i - \alpha + \beta\gamma^{x_i}\right)^2\right) I_{(0,1)}(\gamma)$$

$$**Full \quad conditional \quad for \quad \tau^2** : \pi\left(\tau^2 \mid \alpha, \beta, \gamma, x, y\right) = \frac{b^a}{\Gamma(a)} \tau^{2^{(a-1)}} \exp\left(\frac{-b}{\tau^2}\right) \frac{1}{(\tau\sqrt{2\pi})^n} \exp\left(-\frac{1}{2\tau^2}\sum_{i=1}^{n}\left(Y_i - \alpha + \beta\gamma^{x_i}\right)^2\right) \propto \frac{1}{\tau^2\left(a+1+\frac{n}{2}\right)} \exp\left(-\frac{b + \frac{1}{2}\sum_{i=1}^{n}\left(Y_i - \alpha + \beta\gamma^{x_i}\right)^2}{\tau^2}\right) I_{(0,\infty)}(\tau) \pi\left(\tau^2 \mid \alpha, \beta, \gamma, x, y\right) \sim IGamma\left(a +$$

1e) Which distribution can you recognize within standard parametric families so that direct simulation from full conditional can be easily implemented ?

Except for γ, the other parameters α, β and τ^2 can be recognized as Truncated Normal, Truncated Normal and Inverse Gamma respectively.

1f) Using a suitable Metropolis-within-Gibbs algorithm simulate a Markov chain (T = 10000) to approximate the posterior distribution for the above model

```
# Full conditional for alpha
fc_alpha= function(beta, gamma,tau_2){
  mu=sigma_alpha*sum(y+beta*gamma^x)/(n*sigma_alpha+tau_2)
  sd=(tau_2*sigma_alpha)/(n*sigma_alpha + tau_2)
  alpha1=0
  if(alpha1 < 1) {
    alpha1=rnorm(1,mu,sqrt(sd))
  }else{
  return(alpha1)
  }
}


# Full conditional for beta
fc_beta= function(alpha,gamma,tau_2){
  mub=(sigma_beta*sum((alpha-y)*gamma^x))/(sigma_beta*sum(alpha- y)*gamma^(2*x) + tau_2)
  sdb=sqrt((tau_2*sigma_beta)/(sigma_beta*sum(gamma^(2*x)) +tau_2))
  beta1=0
  if(beta1< 1){
    beta1=rnorm(1,mub,sdb)
  }
    else{
      return(beta1)
  }

}

# Full conditional for gamma
fc_gamma <- function(gamma, alpha, beta, tau_2){
   return(exp(-1/(2*tau_2)*sum((y-alpha+beta*gamma^x)^2)))
}

# Full conditional for tau^2
fc_tau_2 <- function(alpha, beta, gamma){

  mut <- n/2 + a
  mut2=b+1/2*sum((y-alpha+beta*gamma^x)^2)
```

```
  tau_21 <- rinvgamma(1, mut, mut2)

  return(tau_21)
}
```

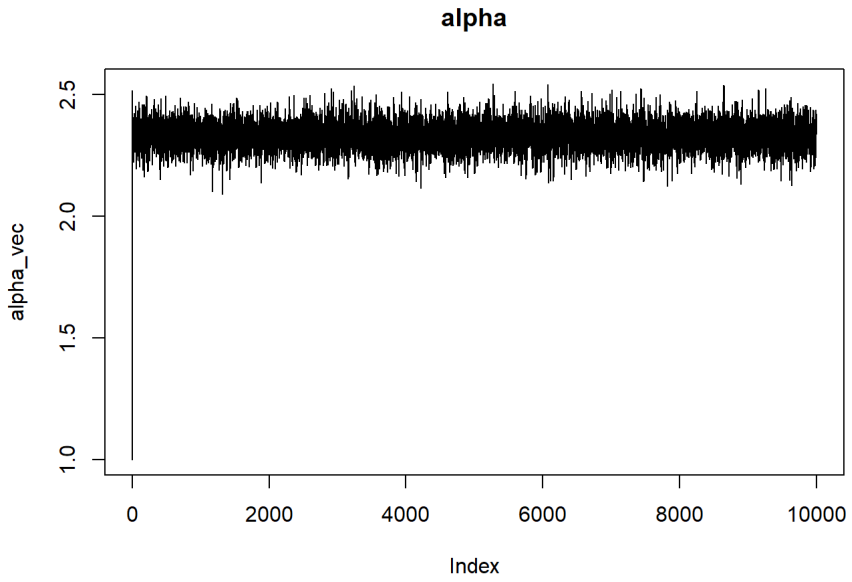
```
#SIMULATION


M = 10000

# parameters
alpha_vec  <-rep(NA,M+1)
beta_vec   <- rep(NA,M+1)
gamma_vec  <- rep(NA,M+1)
tau_2_vec  <- rep(NA,M+1)

alpha_vec[1]  <- 1
beta_vec[1] <- 1
gamma_vec[1]  <- 0.5
tau_2_vec[1]  <- 0.5

#hyperparameters
sigma_alpha <- 10000
sigma_beta <- 10000
a <- 0.001
b <- 0.001

for (i in 1:M){
  alpha_vec[i+1] <- fc_alpha(beta = beta_vec[i],
                             gamma = gamma_vec[i],
                             tau_2 = tau_2_vec[i])



  beta_vec[i+1] <- fc_beta(alpha = alpha_vec[i+1],
                           gamma = gamma_vec[i],
                           tau_2 = tau_2_vec[i])

  gamma_vec[i+1] <- fc_gamma(gamma=gamma_vec[i],
                             alpha = alpha_vec[i+1],
                            beta = beta_vec[i+1],
                             tau_2 = tau_2_vec[i])


  tau_2_vec[i+1] <- fc_tau_2(alpha = alpha_vec[i+1],
                             beta = beta_vec[i+1],
                            gamma = gamma_vec[i+1])



    }
```

```
head(cbind(alpha_vec, beta_vec, gamma_vec, tau_2_vec),10)
```

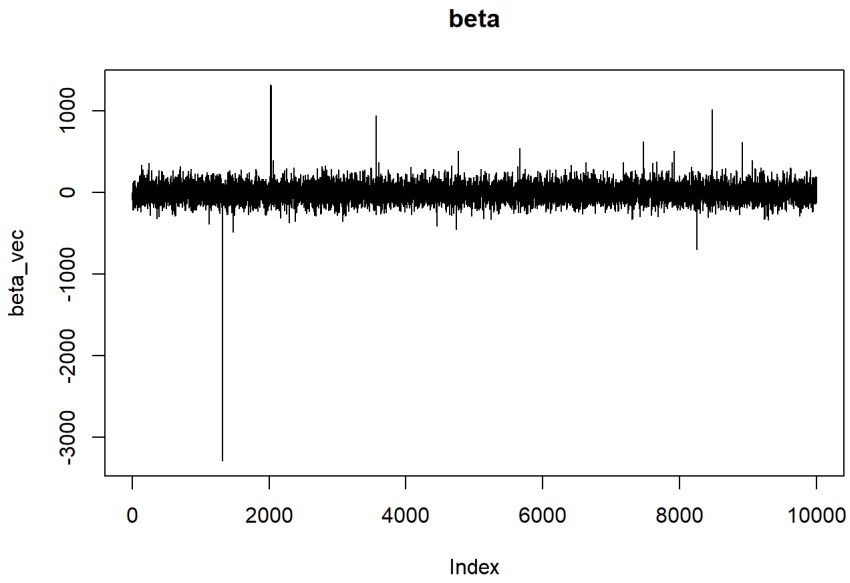
```
##       alpha_vec   beta_vec    gamma_vec  tau_2_vec
##  [1,]  1.000000   1.000000 5.000000e-01 0.50000000
##  [2,]  2.446167   1.960584 5.418937e-01 0.01353411
##  [3,]  2.517268   1.093578 2.121823e-14 0.09452051
##  [4,]  2.289226 -24.340732 2.341773e-05 0.10024101
##  [5,]  2.247841 -85.916464 2.053712e-05 0.09010948
##  [6,]  2.306040 -12.757816 1.667101e-05 0.06496615
##  [7,]  2.363764  70.703401 2.341625e-07 0.13809719
##  [8,]  2.369346  12.576015 7.299096e-04 0.07580333
##  [9,]  2.260514  -1.628439 9.117585e-07 0.08540734
## [10,]  2.348044  20.050970 1.000444e-05 0.07590853
```

1g) Show the 4 univariate trace-plots of the simulations of each parameter

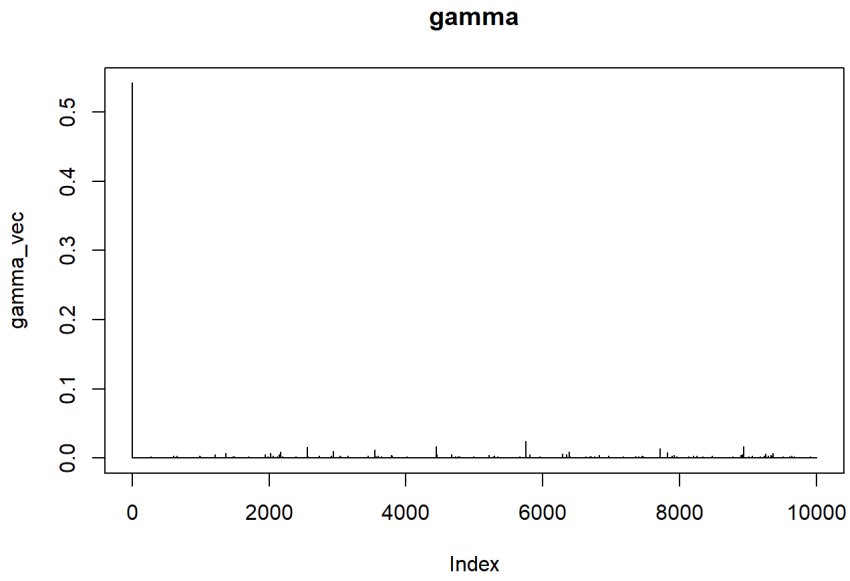
```
plot(alpha_vec,type='l',main='alpha')
```



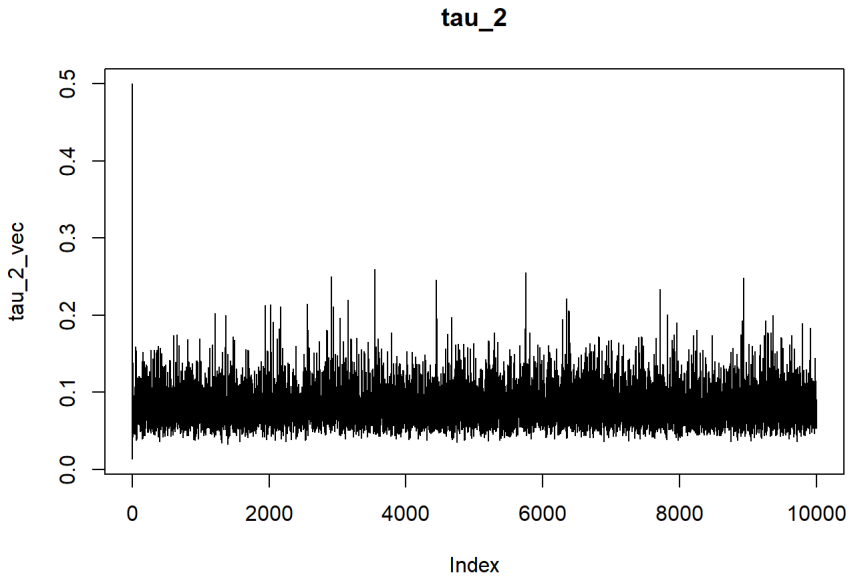

```
plot(beta_vec,type='l',main='beta')
```




```
plot(gamma_vec,type='l',main='gamma')
```

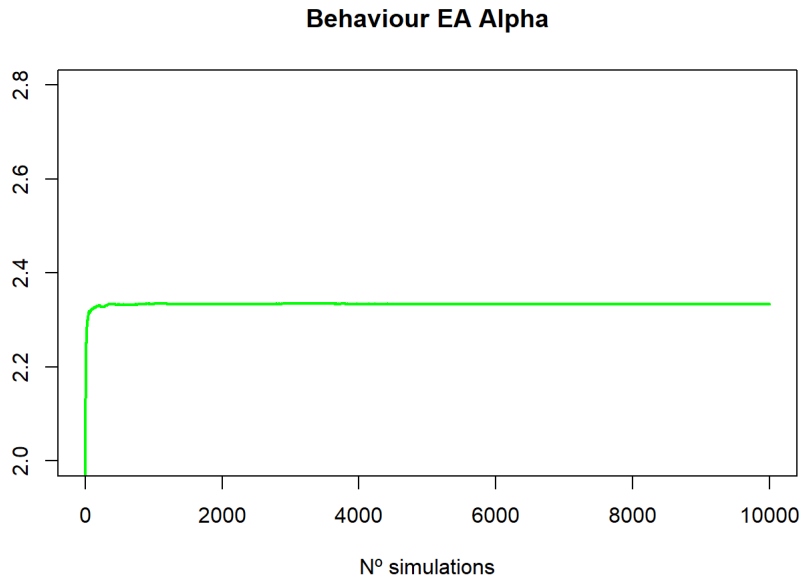


```
plot(tau_2_vec,type='l',main='tau_2')
```

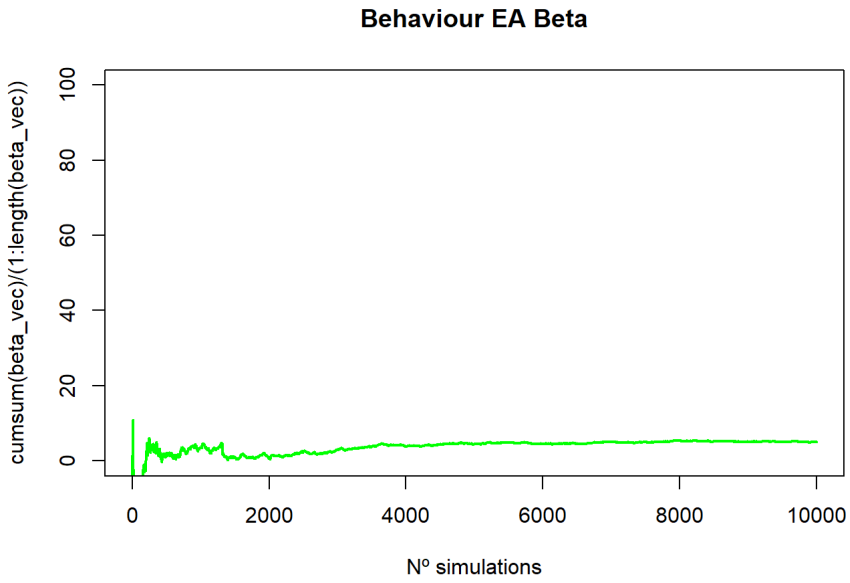


1h) Evaluate graphically the behaviour of the empirical averages $\hat{I}_t$ with growing $t = 1, \ldots, T$

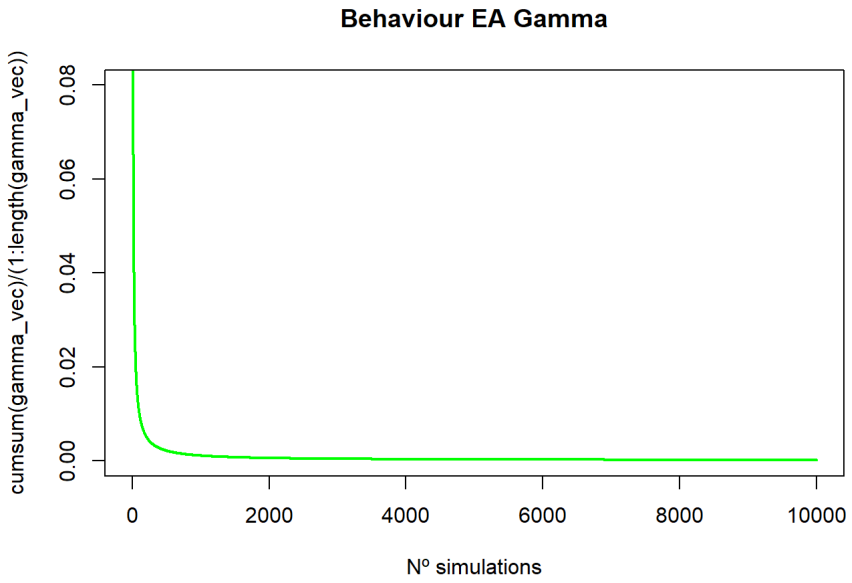
```
# alpha
plot(cumsum(alpha_vec)/(1:length(alpha_vec)), type="l",ylim = c(2, 2.8), ylab="", main="Behaviour EA Alpha", xlab
="Nº simulations",col='green',lwd=2)
```



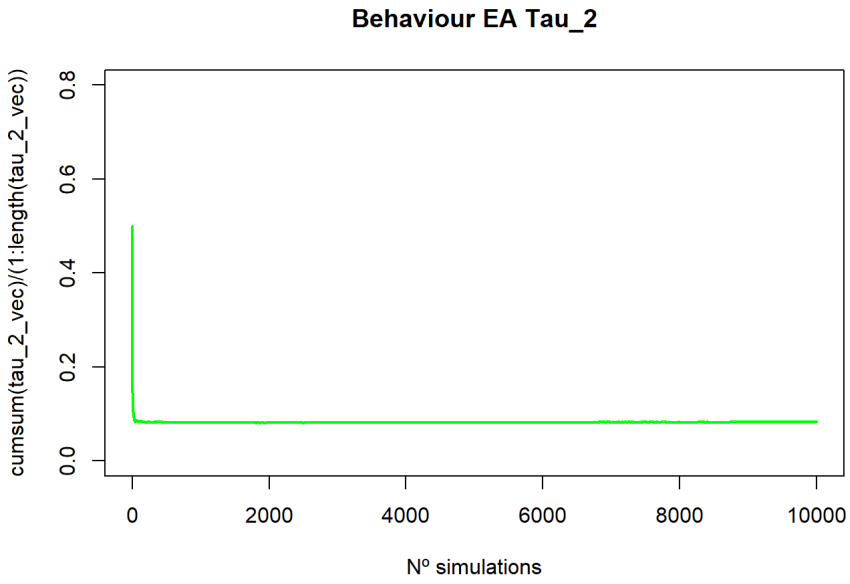

```
# beta
plot(cumsum(beta_vec)/(1:length(beta_vec)), type="l",ylim = c(0,100), main="Behaviour EA Beta", xlab="Nº simulati
ons",col='green',lwd=2)
```




```
# gamma
plot(cumsum(gamma_vec)/(1:length(gamma_vec)), type="l",ylim = c(0,.08), main="Behaviour EA Gamma", xlab="Nº simul
ations",col='green',lwd=2)
```




```
#tau_2
plot(cumsum(tau_2_vec)/(1:length(tau_2_vec)), type="l",ylim = c(0,.8), main="Behaviour EA Tau_2", xlab="Nº simula
tions",col='green',lwd=2)
```



1i) Provide estimates for each parameter together with the approximation error and explain how you have evaluated such error

The estimate that it will be used it will be the mean of each parameter.

```
#Estimates

cat("Estimate alpha:", mean(alpha_vec),"\nEstimate beta :", mean(beta_vec),"\nEstimate gamma", mean(gamma_vec),
"\nEstimate tau^2", mean(tau_2_vec))
```

```
## Estimate alpha: 2.333593
## Estimate beta : 4.990535
## Estimate gamma 0.0001942675
## Estimate tau^2 0.08223282
```

To find the approximation error, we will compute the variance over the length of the Markov chain.

```
cat("Approximation error alpha:", var(alpha_vec)/length(alpha_vec),"\nApproximation error beta :", var(beta_vec)/
length(beta_vec),"\nEstimatApproximation error gamma", var(gamma_vec)/length(gamma_vec), "\nApproximation error t
au^2", var(tau_2_vec)/length(tau_2_vec))
```

```
## Approximation error alpha: 3.272156e-07
## Approximation error beta : 1.193219
## EstimatApproximation error gamma 5.467289e-09
## Approximation error tau^2 6.301069e-08
```

1l) Which parameter has the largest posterior uncertainty? How did you measure it?

The largest posterior uncertainty can be measured checking the standard error of each parameter towards its estimate(the mean of the parameter).Hence,

```
cat("Approximation error alpha:", sd(alpha_vec)/mean(alpha_vec),"\nApproximation error beta :", sd(beta_vec)/mean
(beta_vec),"\nEstimatApproximation error gamma", sd(gamma_vec)/mean(gamma_vec), "\nApproximation error tau^2", sd
(tau_2_vec)/mean(tau_2_vec))
```

```
## Approximation error alpha: 0.02451397
## Approximation error beta : 21.88944
## EstimatApproximation error gamma 38.0634
## Approximation error tau^2 0.3052697
```

Beta has the largest uncertainty, most likely because at the early stage of the simulation its value fluctuates too much.

1m) Which couple of parameters has the largest correlation (in absolute value)?

```
cor(cbind(alpha_vec, beta_vec, gamma_vec, tau_2_vec))
```

```
##             alpha_vec      beta_vec     gamma_vec   tau_2_vec
## alpha_vec  1.00000000  0.0105414763 -0.1434847429 -0.04026259
## beta_vec   0.01054148  1.0000000000  0.0007943756 -0.01283943
## gamma_vec -0.14348474  0.0007943756  1.0000000000  0.09530053
## tau_2_vec -0.04026259 -0.0128394286  0.0953005253  1.00000000
```

```
corr = cor(data.frame(alpha_vec, beta_vec, gamma_vec, tau_2_vec))
corrplot.mixed(corr)
```

As it can be appreciated, though there aren't high coefficient values, alpha and gamma are the most correlated

1n) Use the Markov chain to approximate the posterior predictive distribution of the length of a dugong with age of 20 years.

```
dug_20 = rep(NA, 10000)

for(i in 1:M){
  mu=alpha_vec[i]+beta_vec[i]*gamma_vec[i]^20
  sd=sqrt(tau_2_vec[i])
  dug_20[i]= rnorm(1, mu,sd)

}

cat("prediction", mean(dug_20))
```

```
## prediction 2.332552
```

1o) Provide the prediction of a different dugong with age 30

```
dug_30 = rep(NA, 10000)

for(i in 1:M){
  mu=alpha_vec[i]+beta_vec[i]*gamma_vec[i]^30
  sd=sqrt(tau_2_vec[i])
  dug_30[i]= rnorm(1, mu,sd)

}

cat("prediction", mean(dug_30))
```

```
## prediction 2.332499
```

CONCLUSION: this incredible little difference between predictions could be due to the fact that once a dugong reaches an age close to 20, they don't grow more.

1p) Which prediction is less precise?

We will take a look at the approximation error and the uncertainty.

```
#Approximation error
err_dug_20 = var(dug_20)/length(dug_20)
err_dug_30 = var(dug_30)/length(dug_30)

err_dug_20
```

```
## [1] 8.475135e-06
```

```
err_dug_30
```

```
## [1] 8.562057e-06
```

```
#Uncertainty
un_dug_20 = sd(dug_20)/mean(dug_20)
un_dug_30 = sd(dug_30)/mean(dug_30)

un_dug_20
```

```
## [1] 0.1248079
```

```
un_dug_30
```

```
## [1] 0.1254491
```

As it can be seen,althoug for a very little values, both error and uncertainty of dugongs of age 30 are higher, so we can conclude that the prediction for age 20 is a bit less precise.