<u>ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ</u>
ΤΜΗΜΑ ΣΤΑΤΙΣΤΙΚΗΣ

Μεταπτυχιακό Πρόγραμμα Συμπληρωματικής Ειδίκευσης
Εφαρμοσμένη Στατιστική

Μάθημα: Εφαρμοσμένη Μπεϋζιανή Στατιστική
Applied Bayesian Statistics

Καθηγητής: Ντζούφρας Ιωάννης

Εργασία 1η: MCMC

Ημερομηνία Παράδοσης εργασίας: 11/4/2020

Κωνσταντίνα Τσάμη (p3621817)

ΠΕΡΙΕΧΟΜΕΝΑ

Ερώτημα 1^0 (α)

Στόχος μας στη παρούσα εργασία είναι η προσέγγιση (μοντελοποίηση) της κατανομής των παρακάτω δεδομένων όσο είναι δυνατό καλύτερα.

0.4, 0.01, 0.2, 0.1, 2.1, 0.1, 0.9, 2.4, 0.1, 0.2

Θα προσαρμόσουμε διάφορες κατανομές και θα εκτιμήσουμε τις παραμέτρους (posteriors) μέσω της Μπεϋζιανής θεώρησης και του αλγορίθμου MCMC (*Markov Chain Monte Carlo).* Ξεκινώντας, με την Εκθετική Κατανομή με παράμετρο θ.

$$Y_i \sim Exp(\theta)$$

Κώδικας στο OpenBUGS:

Στην Μπεϋζιανή προσέγγιση και στον αλγόριθμο MCMC (Gibbs Sampling) μας αρκεί να ορίσουμε την πιθανοφάνεια της Y/θ (f(y/θ) = likelihood function) και την prior κατανομή της θ, f(θ). Χτίζουμε λοιπόν το μοντέλο μας στο πρόγραμμα “OpenBUGS” ως εξής.

```
model{
        # Likelihood

        for (i in 1:n){

           Y[i] ~ dexp(theta[i])

           log(theta[i])<-v

           c[i]<- 1/theta[i]

        }

        #prior

        v ~ dnorm(0.0,0.001)

}
```

Εν συνεχεία, θα ορίσουμε την αρχική τιμή του θ ίση με 1, δηλαδή log(θ)=0.

```
# INITIAL VALUES
list( v=0 )
```

Τέλος, περνάμε τα δεδομένα μας με δύο διαφορετικούς τρόπους.
Ο πρώτος τρόπος είναι με τη μορφή λίστας ως εξής.

```
# DATA
list( Y=c(0.4, 0.01, 0.2, 0.1, 2.1, 0.1, 0.9, 2.4, 0.1, 0.2) , n=10)
```

Ο δεύτερος τρόπος είναι σε τετραγωνική μορφή, προσέχοντας να αφήσουμε μία κενή γραμμή κάτω από την εντολή END.

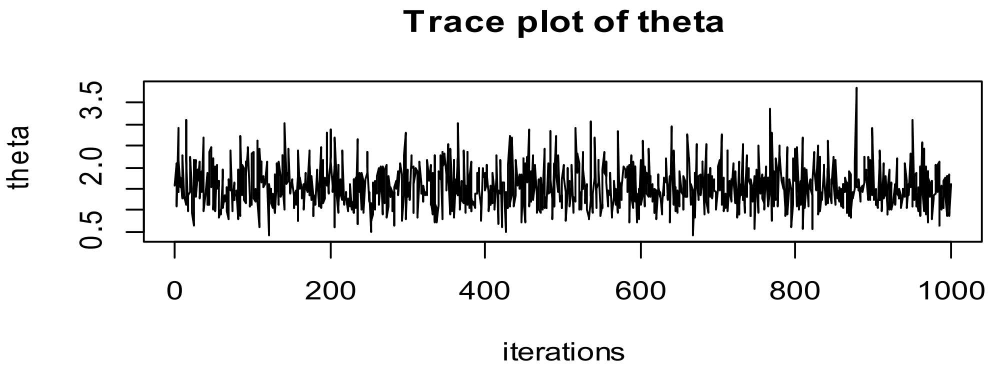
```
list(n=10)
        Y[]

        0.4

        0.01

        0.2

        0.1

        2.1

        0.1

        0.9

        2.4

        0.1

        0.2

        END
```

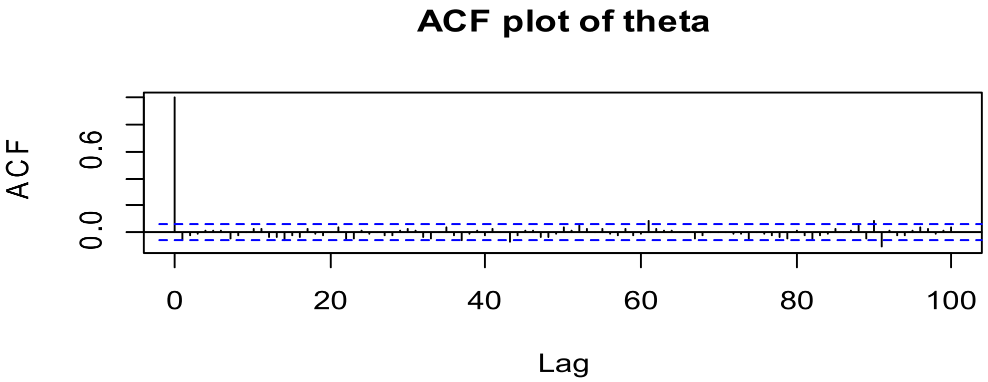
Διαγραμματικός έλεγχος σύγκλισης

Έχοντας κάνει 500 επαναλήψεις για να τρέξει και να σταθεροποιηθεί ο αλγόριθμός μας και άλλες 1000 για να παράξει τιμές, θα κάνουμε τους απαραίτητους διαγνωστικούς ελέγχους μέσω “trace” και “autocorrelation” διαγραμμάτων.

Όπως βλέπουμε στο *Διάγραμμα 1 (‘ trace plot of theta’)* παρακάτω, η διακύμανση της posterior κατανομής της θ φαίνεται σταθερή. Η τιμή της θ κυμαίνεται σε ένα σταθερό εύρος τιμών χωρίς να παρατηρείται κάποια ανοδική ή καθοδική τάση όσο αυξάνεται ο αριθμός επαναλήψεων. Δεν έχουμε καμία ένδειξη ύπαρξης αυτοσυσχέτισης, όπως επιβεβαιώνει και το διάγραμμα ‘*acf plot of theta’*. Παρατηρούμε μόνο 3 peaks σε σύνολο 100 lags, ενώ όλες οι υπόλοιπες τιμές είναι εντός του διαστήματος εμπιστοσύνης. Συμπεραίνουμε λοιπόν, ότι με μεγάλη πιθανότητα ο αλγόριθμός μας συγκλίνει χωρίς πρόβλημα.

Διάγραμμα 1- Διαγράμματα "Trace" και "ACF" της θ





Υπολογισμός σφάλματος Monte-Carlo , περιγραφικά μέτρα και διαγράμματα πυκνότητας πιθανότητας των posterior κατανομών

Εξάγουμε τον συνοπτικό πίνακα των περιγραφικών μέτρων της παραμέτρου θ, 1/θ = c και log(Θ) = v.

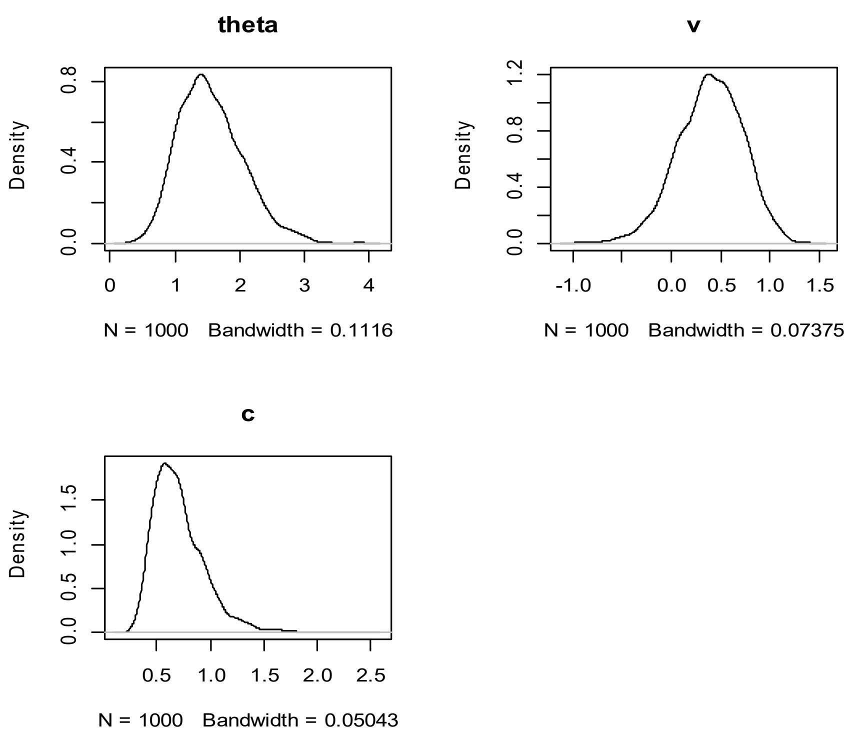
Πίνακας 1- Συνοπτικός πίνακας περιγραφικών μέτρων των posterior κατανομών

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
c	0.7139	0.256	0.007328	0.3719	0.6681	1.347	501	1000
theta	1.558	0.4935	0.01454	0.7503	1.498	2.694	501	1000
v	0.3924	0.3261	0.009414	-0.2872	0.4039	0.9908	501	1000

Όπως παρατηρούμε στον *Πίνακα 1*, το σφάλμα που προκύπτει από την προσομοίωση Monte Carlo (MC_error) είναι πολύ μικρό. Όσο μεγαλύτερο δείγμα (sample) παίρνουμε, τόσο μεγαλύτερη ακρίβεια(μικρότερο σφάλμα) αναμένουμε να έχουμε στην εκτίμηση των παραμέτρων (μέσος posterior κατανομών) . Συγκεκριμένα, για την παράμετρο θ (theta) το σφάλμα εκτίμησης Monte-Carlo υπολογίζεται ίσο με 0.015. Για την παράμετρο c=1/θ το σφάλμα εκτιμάται 0.0073, ενώ για την v=log(θ) είναι ίσο με 0.009.

Στον ίδιο πίνακα επίσης, βλέπουμε τις εκτιμήσεις *(mean)*, την τυπική απόκλιση*(sd)* και τα 95% διαστήματα εμπιστοσύνης των παραμέτρων. Για την θ η εκτίμηση είναι **1.56** με τυπική απόκλιση ίση με 0.49 και 95% διάστημα εμπιστοσύνης (**0.75** – **2.69**). Από το “*density plot*” της θ (*Διάγραμμα 2*) θα λέγαμε ότι η εκτίμηση είναι κεντρική και κοντά στη διάμεσο (*Πίνακας 1, median=1.498*), η posterior κατανομή της θ προσεγγίζει την κανονική κατανομή και ο μέσος είναι ένα καλό μέτρο περιγραφής κεντρικής θέσης. Σε παρόμοια συμπεράσματα καταλήγουμε και για τις υπόλοιπες παραμέτρους. Επίσης, η συμμετρία της κατανομής και το γεγονός ότι είναι μονοκόρυφη μας επιτρέπει να χρησιμοποιήσουμε με ασφάλεια τον κλασσικό τύπο για το κριτήριο DIC ($D + 2p_D$) σε αυτό το μοντέλο.

Διάγραμμα 2- “Density plot” για θ(theta=), log(θ)(=v) και 1/θ(=c)



<u>Ergotic mean plots</u>

Εξάγουμε τις τιμές του MCMC χρησιμοποιώντας την επιλογή “ CODA “ στο OpenBUGS.

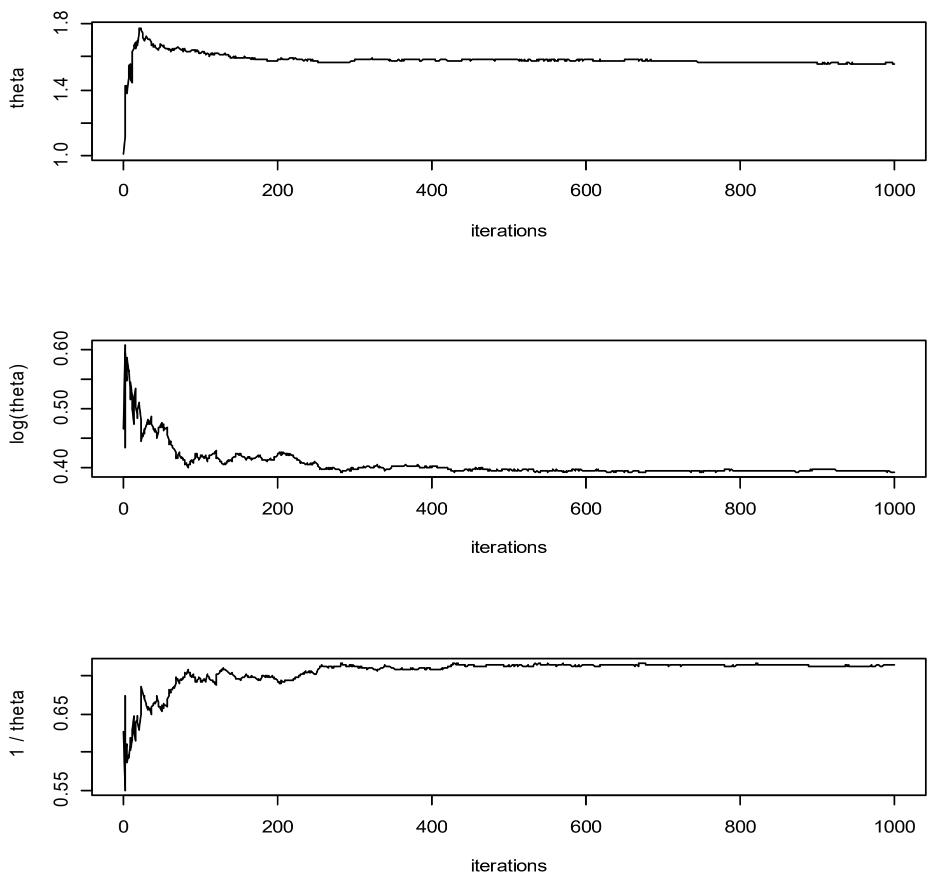
Εισάγουμε τις τιμές στην R.

```
modelexp.sim<-read.coda("output.odcCODAchain1", "output.odcCODAindex",start=501, quiet=FALSE)
```

Δημιουργούμε τα διαγράμματα.

```
par(mfrow=c(3,1))
x<-modelexp.sim[,"theta"]
plot( cumsum(x)/1:length(x), type='l',xlab="iterations",   ylab="theta" )

x<-modelexp.sim[,"v"]
plot( cumsum(x)/1:length(x), type='l',xlab="iterations",   ylab="log(theta)" )

x<-modelexp.sim[,"c"]
plot( cumsum(x)/1:length(x), type='l',xlab="iterations",   ylab="1/theta")
```

<u>Διάγραμμα 3- “Ergotic mean trace plots” των θ, log(θ) και 1/θ</u>



Τα διαγράμματα "ergotic means" *(Διάγραμμα 3)* μας δείχνουν την μέση τιμή της κάθε παραμέτρου σε κάθε βήμα - προσομοιωμένη τιμή, που μας δίνει ο αλγόριθμος MCMC. Στο *Διάγραμμα 3* λοιπόν, βλέπουμε ότι ο μέσος των τιμών σταθεροποιείται σχετικά γρήγορα για όλες τις παραμέτρους και ειδικά μετά το 400^{o} προσομοιωμένο δείγμα φαίνεται να μην υπάρχει σχεδόν καθόλου διακύμανση. Αυτό σημαίνει ότι ο αλγόριθμος συγκλίνει γρήγορα και οι εκτιμήσεις των παραμέτρων είναι αρκετά σταθερές.

Ερώτημα 2^{0} (b)

Σε αυτή την ενότητα, θα ακολουθήσουμε ανάλυση παρόμοια με του πρώτου ερωτήματος, αυτή τη φορά για δύο διαφορετικά μοντέλα.

Στο πρώτο θα προσαρμόσουμε Κατανομή Γάμμα

Yi ~ Gamma(a,b)

Στο δεύτερο θα προσαρμόσουμε log- Normal κατανομή στα δεδομένα.

Yi ~ log-Normal(m,σ^2)

Ξεκινάμε με τη προσαρμογή του πρώτου μοντέλου και την ανάλυσή του που θα ακολουθήσει.

- **Yi ~ Gamma(a,b)**

Κώδικας μοντέλου στο OpenBUGS:

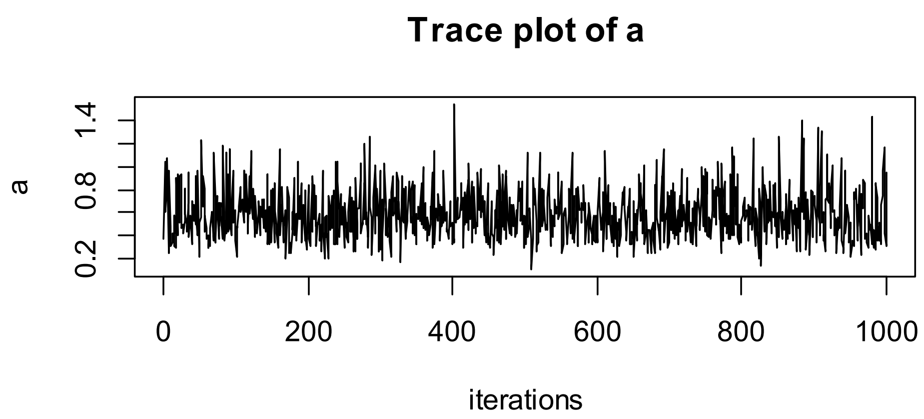
```
model{
        # Likelihood
        for (i in 1:n){
            Y[i] ~ dgamma(a,b)
        }
```

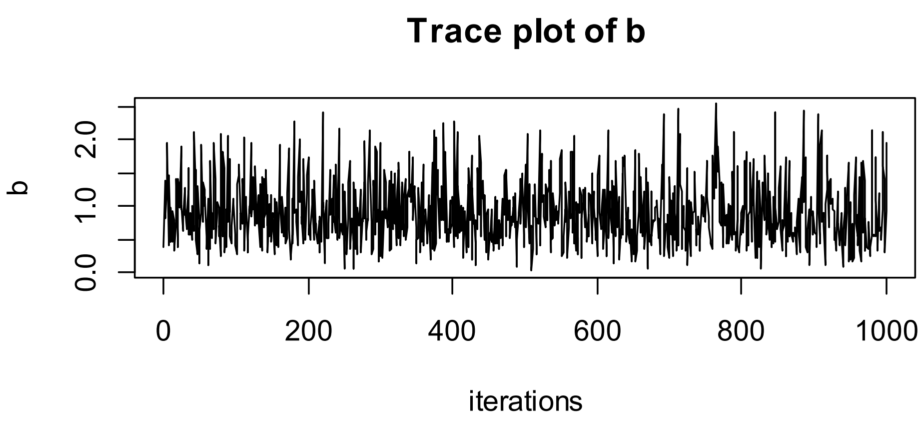
```
        #prior
        a~dgamma(0.001,0.001)
        b~dgamma(0.001,0.001)
}
```

Αρχικές τιμές.
list(a=1 , b=1)

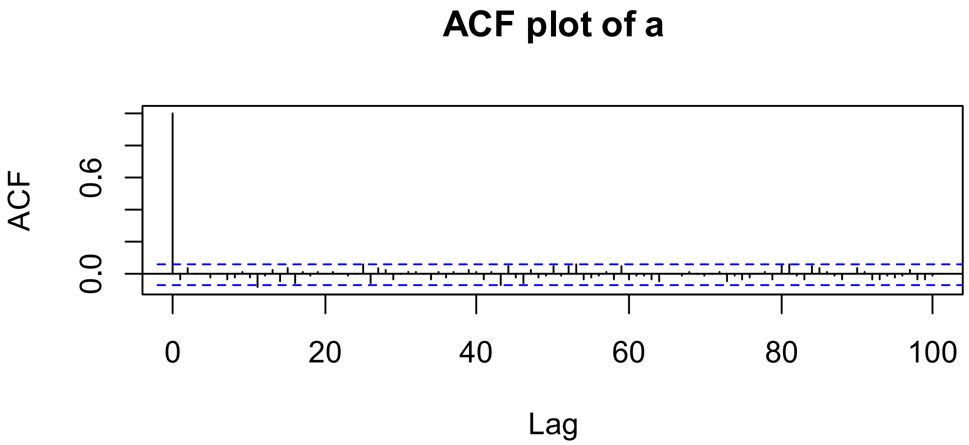
Διαγραμματικός έλεγχος σύγκλισης

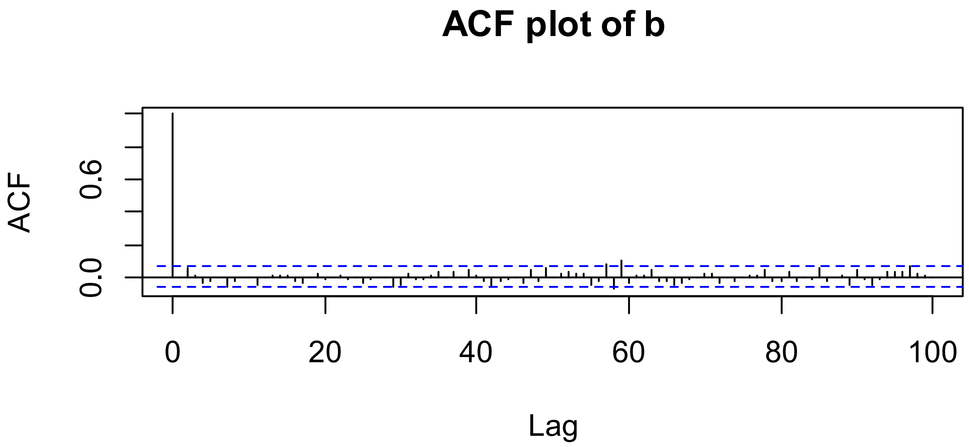
Διάγραμμα 4 - Διαγράμματα "Trace plots" των a,b





Διάγραμμα 5 - Διαγράμματα "ACF plots" των a,b





Από τα *Διαγράμματα 4* και *5*, μπορούμε να πούμε ότι ο αλγόριθμος MCMC συγκλίνει χωρίς κανένα πρόβλημα. Τα "trace plots" *(Διάγραμμα 4)* των παραμέτρων a και b εμφανίζονται με σταθερή διακύμανση. Επιπλέον, στα διαγράμματα των συντελεστών αυτοσυσχέτισης *(Διάγραμμα 5)* δεν παρατηρούμε ιδιαίτερα 'peaks' και φθίνουν γρήγορα που σημαίνει ότι δεν υπάρχει πρόβλημα αυτοσυσχέτισης στις τιμές που παρήγαγε ο αλγόριθμος.

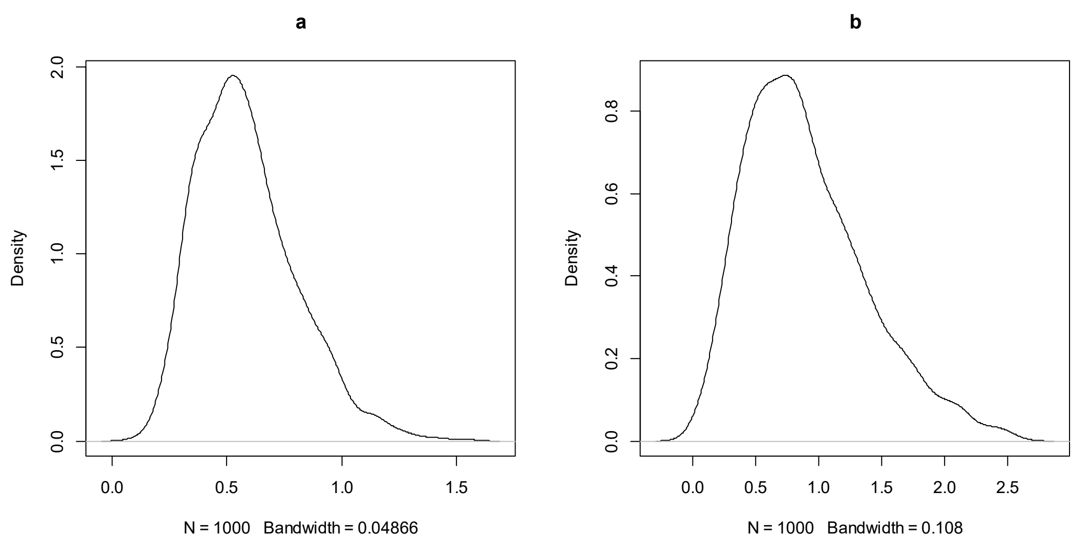
<u>Υπολογισμός σφάλματος Monte-Carlo , περιγραφικά μέτρα και διαγράμματα πυκνότητας πιθανότητας των posterior κατανομών</u>

Εξάγουμε τον συνοπτικό πίνακα των περιγραφικών μέτρων των παραμέτρων a,b του μοντέλου. Στον πίνακα υπάρχει και το Monte-Carlo error *(Πίνακας 2 , MC_error).* Παρατηρούμε λοιπόν ότι το σφάλμα της προσομοίωσης Monte Carlo είναι αρκετά μικρό και οι εκτιμήσεις που έχουμε είναι ότι το a είναι ίσο με **0.59** (*Πίνακας 2 , mean)* και το b ίσο με **0.90**. Επίσης, στα " density plots" *(Διάγραμμα 6)* των a, b παρατηρούμε ότι οι εκτιμήσεις αυτές είναι κεντρικές και κοντά στη διάμεσο (*Πίνακας 2 , median)* και οι posterior κατανομές εμφανίζονται με μία μικρή δεξιά ασυμμετρία.

<u>Πίνακας 2- Συνοπτικός πίνακας περιγραφικών μέτρων των posterior κατανομών</u>

	mean	**sd**	**MC_error**	**val2.5pc**	**median**	**val97.5pc**	**start**	**sample**
a	0.5868	0.2169	0.01246	0.2561	0.5574	1.119	501	1000
b	0.9017	0.4796	0.02556	0.1928	0.8202	2.085	501	1000

<u>Διάγραμμα 6 "Density plots" για a, b</u>

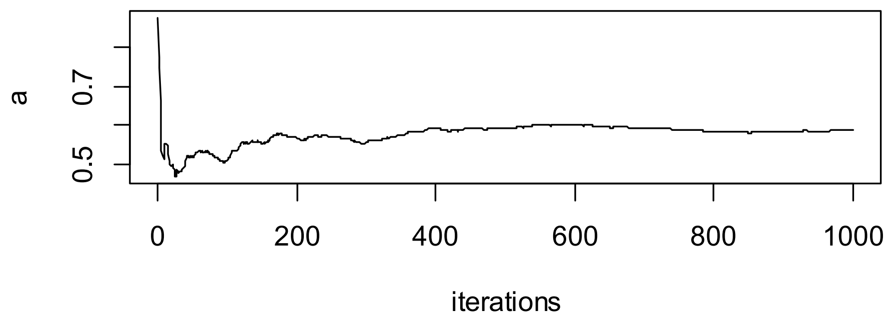


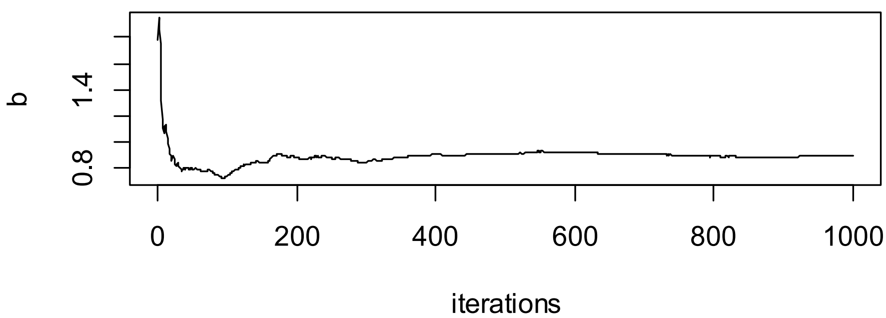
Ergotic mean plots

Εκτελούμε τις σχετικές εντολές στην R για το διάβασμα του αρχείου που εξάγαμε με την επιλογή CODA και στη συνέχεια την παραγωγή των διαγραμμάτων.

```
modelgamma.sim←read.coda("output.odcCODAchain1",
 "output.odcCODAindex",start=501, quiet=FALSE)

par(mfrow=c(2,1))
x<-modelgamma.sim[,"a"]
plot( cumsum(x)/1:length(x), type='l',xlab="iterations",ylab=" a " )

x<-modelgamma.sim[,"b"]
plot( cumsum(x)/1:length(x), type='l',xlab="iterations",   ylab=" b " )
```

Διάγραμμα 7 - "Ergotic mean trace plots" των a, b





Στο Διάγραμμα 7 των “ ergotic mean plots “ ο μέσος των τιμών φαίνεται να σταθεροποιείται σχετικά γρήγορα, μετά τις 400 επαναλήψεις, με μία πολύ μικρή σχεδόν μηδενική διακύμαση. Οι εκτιμήσεις φαίνονται σταθερές.

Σειρά έχει τώρα η προσαρμογή και η ανάλυση του δεύτερου μοντέλου της log-Normal κατανομής.

- **Yi ~ log-Normal(m,σ^2)**

Κώδικας μοντέλου στο OpenBUGS:

Στο OpenBUGS αντί για τη διακύμανση έχουμε σαν παράμετρο την ακρίβεια στο μοντέλο μας (precision =$1/\sigma^2$) . Αν μας ενδιαφέρει να εκτιμήσουμε τη διακύμανση της log-Normal κατανομής για το δείγμα μας, αρκεί να ορίσουμε στον κωδικά μας το s^2=1/tau.

Οπότε, γράφουμε το μοντέλο μας στο OpenBUGS όπως ακολουθεί.

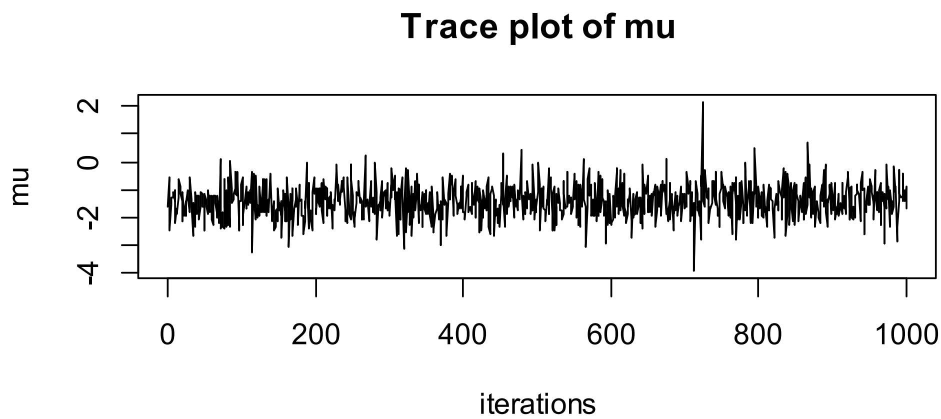
```
model{

   # Likelihood

   for (i in 1:n){

      Y[i] ~ dlnorm(mu,tau)

   }

   #prior

  mu~dnorm(0.0,0.001)

   tau~dgamma(0.001,0.001)

   s2<-1/tau

}
```

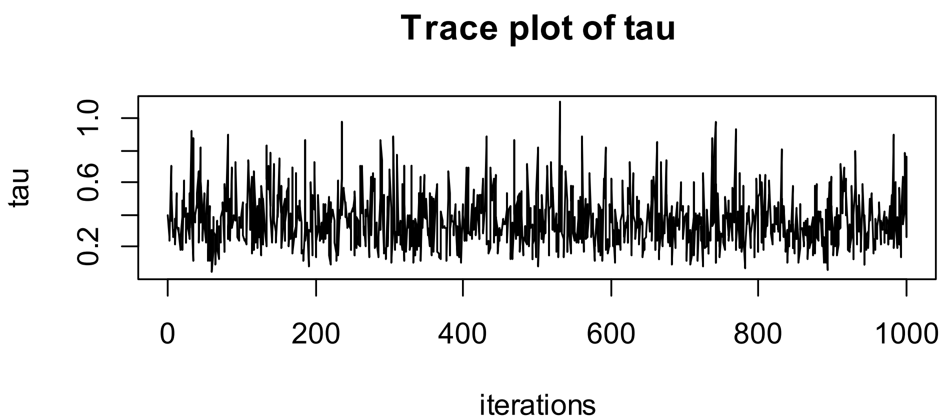
Οι αρχικές τιμές που θα ορίσουμε είναι

```
list( mu=0 , tau=1)
```

<u>Διαγραμματικός έλεγχος σύγκλισης</u>

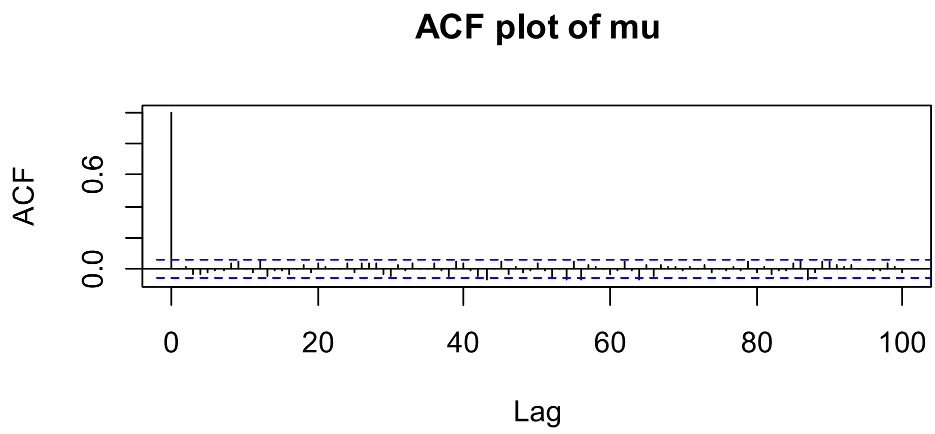
Διάγραμμα 8 - Διαγράμματα "Trace plots" των mu, tau

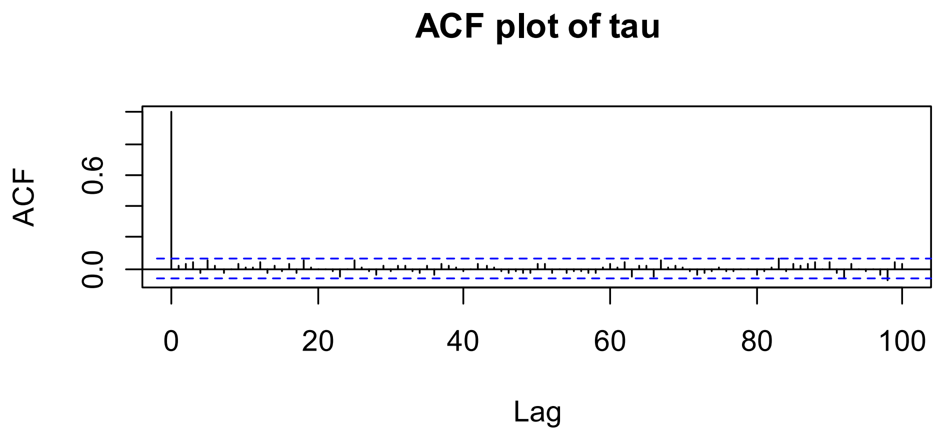




Από τα διαγράμματα "trace plots" (Διάγραμμα 8) και "autocorrelation plots" (Διαγράμμα 9) μπορούμε να συμπεράνουμε ότι ο αλγόριθμος συγκλίνει χωρίς να υπάρχει πρόβλημα αυτοσυσχετίσεων. Η διακύμανση στα "trace plots" είναι σταθερή και στις δύο παραμέτρους αφού ο μέσος των τιμών είναι σταθερός στον αριθμό των επαναλήψεων. Δεν παρατηρούμε έντονες αυξομειώσεις επομένως ο MCMC συγκλίνει επιτυχώς ούτε έχουμε προβλήματα αυτοσυσχετίσεων όπως μας επιβεβαιώνουν και τα "autocorrelation plots" στα οποία οι συντελεστές αυτοσυσχέτισης είναι όλοι εντός των διαστημάτων εμπιστοσύνης.

Διάγραμμα 9 - Διαγράμματα "ACF plots" των mu, tau





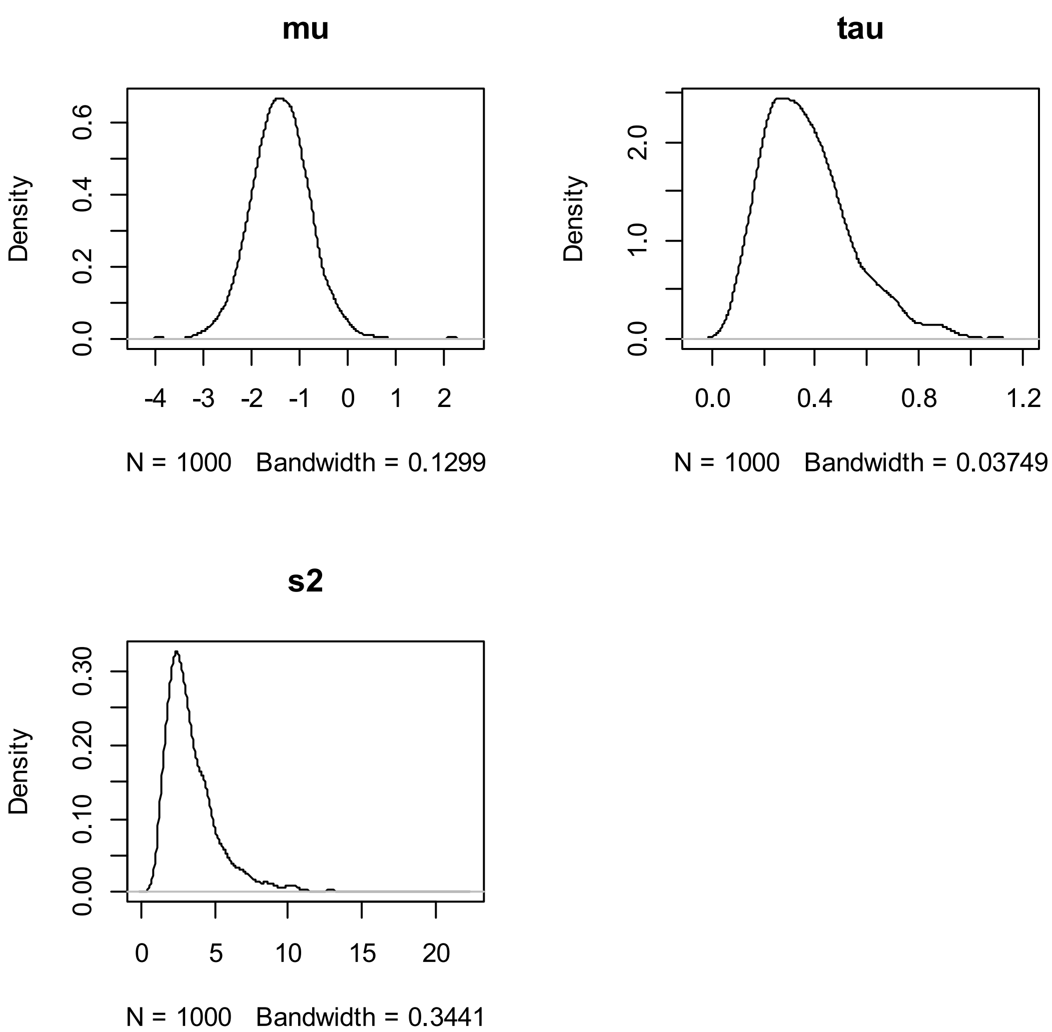
Υπολογισμός σφάλματος Monte-Carlo , περιγραφικά μέτρα και διαγράμματα πυκνότητας πιθανότητας των posterior κατανομών

Ο συνοπτικό πίνακας των περιγραφικών μέτρων των παραμέτρων mu, tau και του s^2 φαίνεται παρακάτω (*Πίνακας3),* επίσης και τα διαγράμματα πυκνότητας πιθανότητας *(Διάγραμμα 10).* Στον πίνακα υπάρχει και το Monte-Carlo error *(Πίνακας 3 , MC_error)* το οποίο είναι πολύ μικρό. Οι εκτιμήσεις που έχουμε για τις παραμέτρους(*Πίνακας 3 , mean)* είναι**: mu=-1.42, tau=0.37, s^2=3.51** . Επίσης, στα "density plots" *(Διάγραμμα 10)* των posterior κατανομών, η κατανομή της mu είναι κανονική (συμμετρική) κατανομή και ο μέσος είναι κατάλληλο μέτρο περιγραφής κεντρικής θέσης. Στις κατανομές των tau και s^2 παρατηρείται μικρή θετική ασυμμετρία.

Πίνακας 3- Συνοπτικός πίνακας περιγραφικών μέτρων των posterior κατανομών

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
mu	-1.422	0.592	0.01277	-2.574	-1.429	-0.2507	501	1000
s^2	3.511	2.064	0.0529	1.248	2.971	9.169	501	1000
tau	0.366	0.1765	0.004132	0.1092	0.3373	0.8032	501	1000

Διάγραμμα 10 "Density plots" των mu, tau , s^2

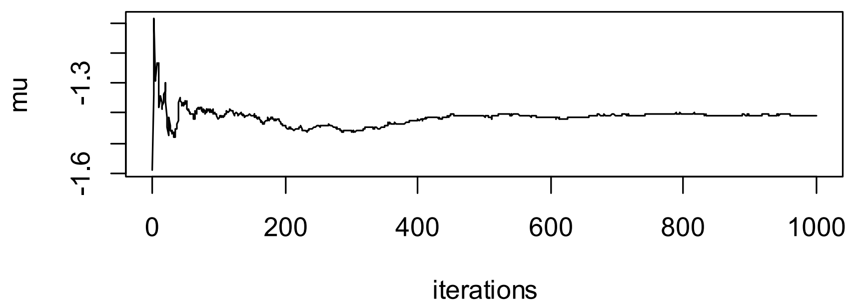


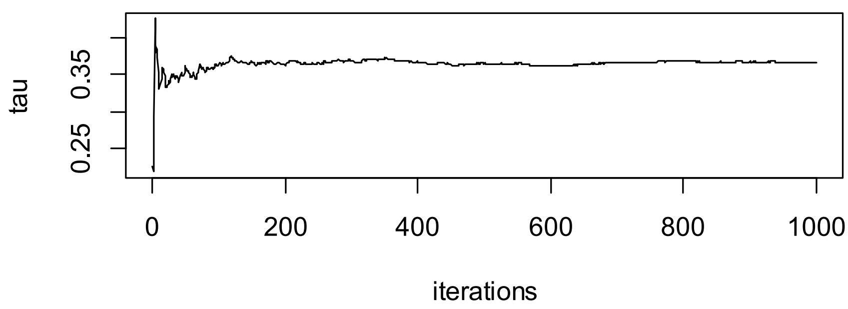
Ergotic mean plots

Εκτελούμε τις σχετικές εντολές στην R από το αρχείο που εξήγαμε με την επιλογή CODA.

```
modellnorm.sim<-read.coda("output.odcCODAchain1","output.odcCODAindex",
quiet=FALSE)

par(mfrow=c(3,1))
x<-modellnorm.sim[,"mu"]
plot( cumsum(x)/1:length(x), type='l',xlab="iterations",   ylab=" mu " )

x<-modellnorm.sim[,"tau"]
plot( cumsum(x)/1:length(x), type='l',xlab="iterations",   ylab=" tau " )

x<-modellnorm.sim[,"s2"]
plot( cumsum(x)/1:length(x), type='l',xlab="iterations",   ylab=" s2 " )
```

Διάγραμμα 11 - "Ergotic mean trace plots" των mu, tau,





Ο μέσος των τιμών όπως βλέπουμε σταθεροποιείται γρήγορα σε μία τιμή και για τις δύο παραμέτρους του μοντέλου, mu και tau (*Διάγραμμα 11*).

Ερώτημα 3^0 (c)

Κώδικας στο OpenBUGS:

Σε αυτό το σημείο καλούμαστε να προσαρμόσουμε μία Κανονική Κατανομή στους λογαρίθμους των τιμών των δεδομένων. Καλούμαστε δηλαδή να εφαρμόσουμε ένα μετασχηματισμό στα δεδομένα μας όπου:

$$\log(Yi) \sim Normal(mu, tau) , tau=1/\sigma^2$$

Το σημαντικό σε αυτή τη περίπτωση είναι ότι δε μπορούμε να ορίσουμε μία μεταβλητή δύο φορές παρά μόνο στη περίπτωση ενός μόνο μετασχηματισμού. Αυτό σημαίνει ότι αν μετασχηματίσουμε τη μεταβλητή μας, μετά δεν θα έχουμε τη δυνατότητα να επιστρέψουμε πάλι στην αρχική κλίμακα των δεδομένων μας. Είναι κάτι λογικό, αφού από τη στιγμή που θα προσαρμόσουμε μία posterior κατανομή στους λογαρίθμους των τιμών της μεταβλητής μας f(θ/log(y)), δεν είναι εφικτό να έχουμε posterior κατανομή στις αρχικές τιμές μας (f(θ/y)).

Μπορούμε ωστόσο, να παράγουμε προβλέψεις για τα log(Yi) και με έναν απλό μετασχηματισμό (exp(log(Yi)) να έχουμε προβλέψεις για τη μεταβλητή μας. Το μοναδικό μειονέκτημα λοιπόν, αυτής της προσέγγισης είναι ότι δε μπορούμε να εφαρμόσουμε το κριτήριο του DIC, όπως και οποιοδήποτε άλλο κριτήριο από τη στιγμή που το μοντέλο μας έχει προσαρμοστεί σε μετασχηματισμένα δεδομένα και επομένως η response μεταβλητή μας είναι διαφορετική από την Y.

Χτίζουμε κανονικά το μοντέλο μας στο “OpenBUGS” ορίζοντας την likelihood κατανομή f(X=/mu,tau) και τον μετασχηματισμό X=log(Y). Ορίζουμε επίσης και τις prior των mu και tau και μία επιπλέον παράμετρο s2 για το s^2.

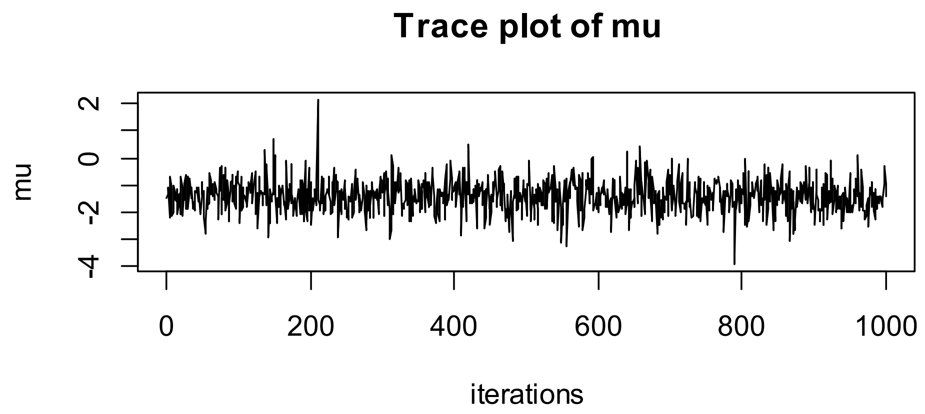
```
model{
   # Likelihood
   for (i in 1:n){
      X[i] ← log(Y[i])
      X[i] ~ dnorm(mu, tau)
      }
   #prior
  mu ~ dnorm(0.0, 0.001)
   tau ~ dgamma(0.001, 0.001)
   s2<-1/tau
}
```

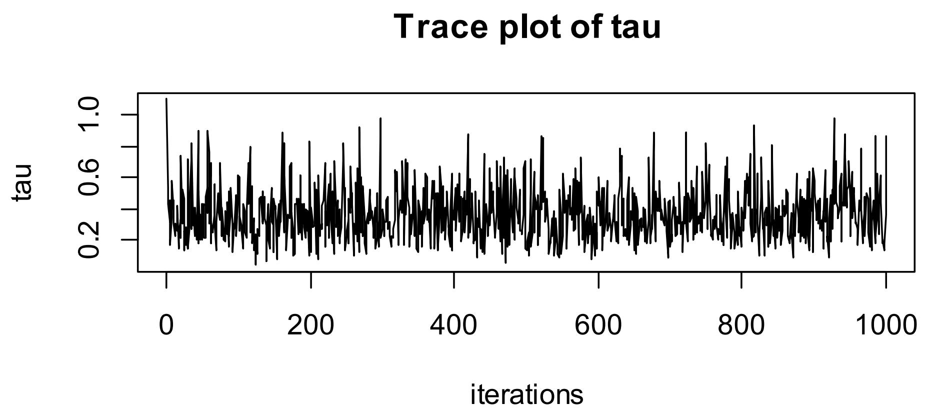
Ορίζουμε τις αρχικές τιμές των mu, tau.

```
list( mu=0 , tau=1)
```

<u>Διαγραμματικός έλεγχος σύγκλισης</u>

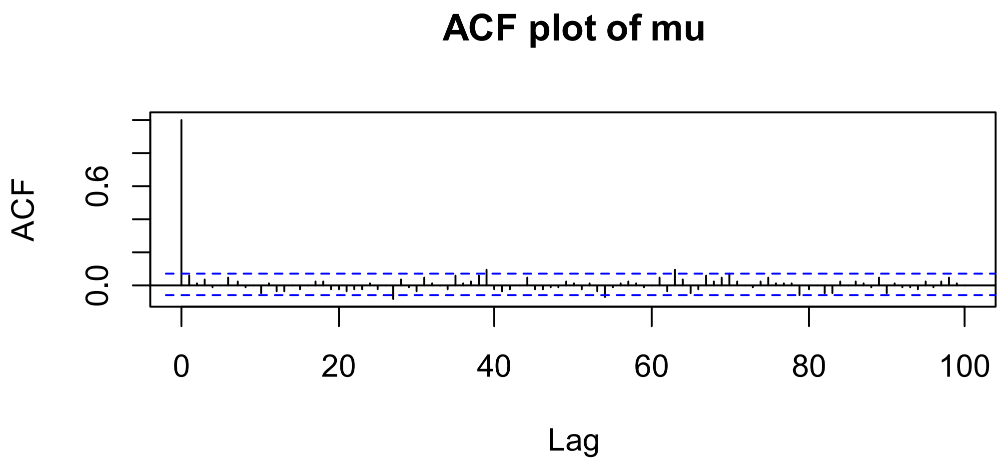
<u>Διάγραμμα 12 - Διαγράμματα "Trace plots" των mu, tau</u>

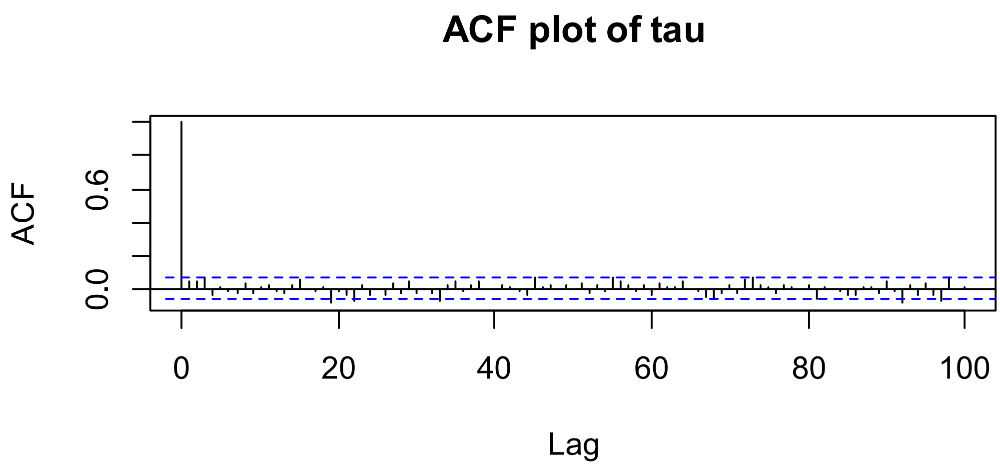




Εξετάζοντας τα Διαγράμματα 12 και 13, δεν παρατηρούμε αυτοσυσχετίσεις και ο αλγόριθμος μπορούμε να πούμε πως συγκλίνει ομαλά. Πιο συγκεκριμένα, οι συντελεστές αυτοσυσχετίσεων φθίνουν γρήγορα προς το 0 και σχεδόν όλοι είναι εντός διαστήματος εμπιστοσύνης *(Διάγραμμα 13)*, ενώ οι τιμές των παραμέτρων στα "trace plots" *(Διάγραμμα 12)* παραμένουν εντός ενός σταθερού εύρους.

Διάγραμμα 13 - Διαγράμματα "ACF plots" των mu, tau





Υπολογισμός σφάλματος Monte-Carlo , περιγραφικά μέτρα και διαγράμματα πυκνότητας πιθανότητας των posterior κατανομών

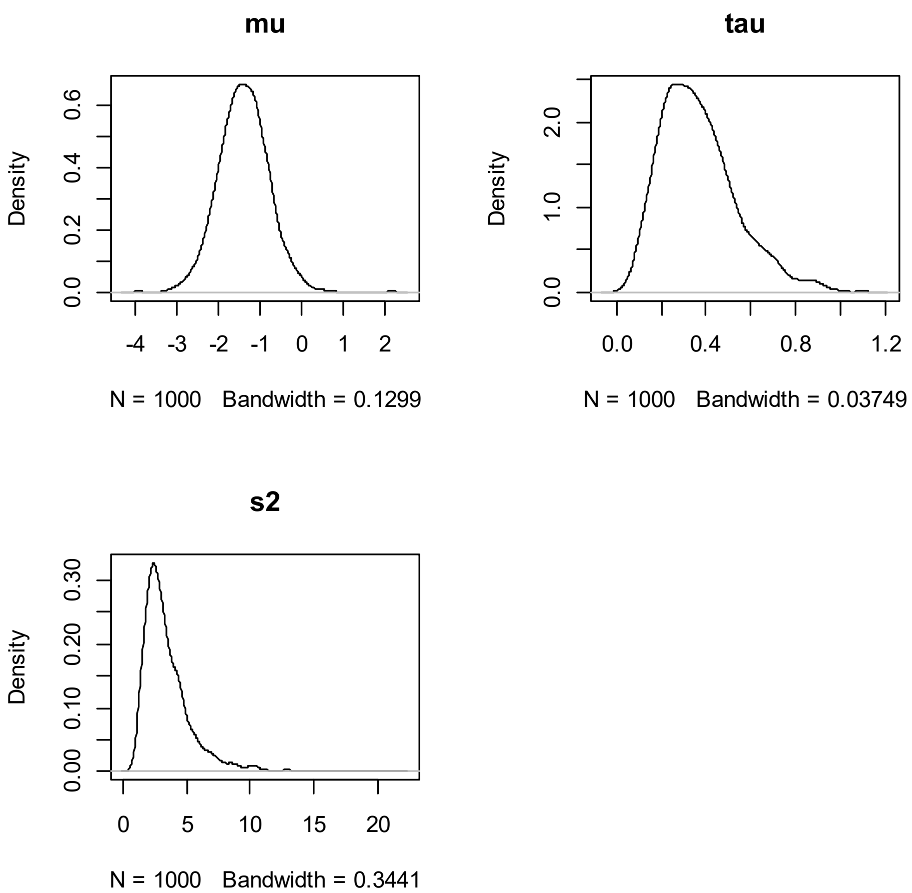
Με μια προσεκτική ματιά στον συνοπτικό πίνακας των περιγραφικών μέτρων των posterior κατανομών(*Πίνακας4)* και στα διαγράμματα πυκνότητας πιθανότητας *(Διάγραμμα 14),* διαπιστώνουμε ότι δε διαφέρουν και πολύ από εκείνα του προηγούμενου μοντέλου της log-Normal κατανομής. Εδώ οι εκτιμήσεις των παραμέτρων(*Πίνακας 4 , mean)* είναι**: mu= -1.42 tau=0.37, s^2=3.51** , ενώ στο προηγούμενο μοντέλο που προσαρμόσαμε είναι: **mu= -1.41, tau=0.37, s^2=3.46** (*Πίνακας 3 , mean)***.** Η απόκλιση μεταξύ των εκτιμήσεων στα δύο μοντέλα είναι ελάχιστη και το σφάλμα Monte Carlo είναι και εδώ πολύ μικρό οδηγώντας σε υψηλή ακρίβεια.

Γενικότερα, θα λέγαμε ότι αναμέναμε αυτή την ταύτιση των αποτελεσμάτων καθώς η προσαρμογή μιας κανονικής κατανομής στους λογαρίθμους των δεδομένων δεν διαφέρει από την προσαρμογή μιας log-Normal κατανομής απευθείας στα δεδομένα. Η τελευταία προσέγγιση μάλιστα, θα λέγαμε ότι είναι πιο ευθύβολη καθώς μας δίνει άμεσα τη δυνατότητα υπολογισμού του DIC και της σύγκρισης με άλλα μοντέλα, καθώς επίσης και την άμεση δημιουργία προβλέψεων ή υπολογισμού ελλιπών τιμών.

Πίνακας 4- Συνοπτικός πίνακας περιγραφικών μέτρων των posterior κατανομών

	mean	**sd**	**MC_error**	**val2.5pc**	**median**	**val97.5pc**	**start**	**sample**
mu	-1.41	0.6021	0.01713	-2.605	-1.419	-0.2301	501	1000
s^2	3.46	2.016	0.06024	1.28	2.948	8.693	501	1000
tau	0.3668	0.1702	0.004956	0.1153	0.3395	0.7833	501	1000

Διάγραμμα 14 "Density plots" των mu, tau , s^2

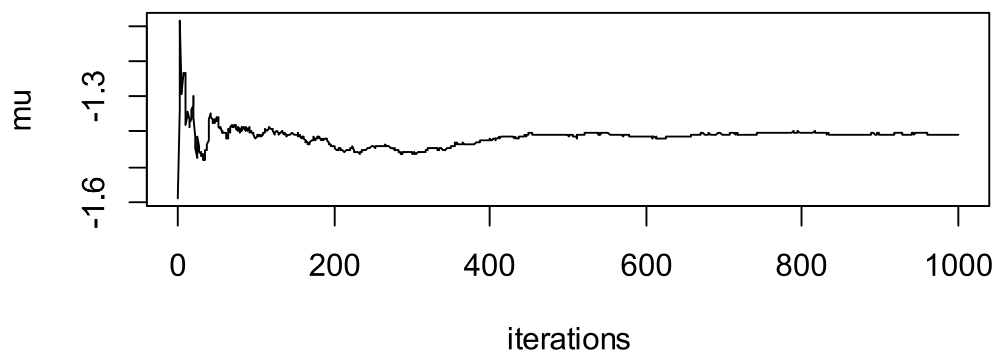


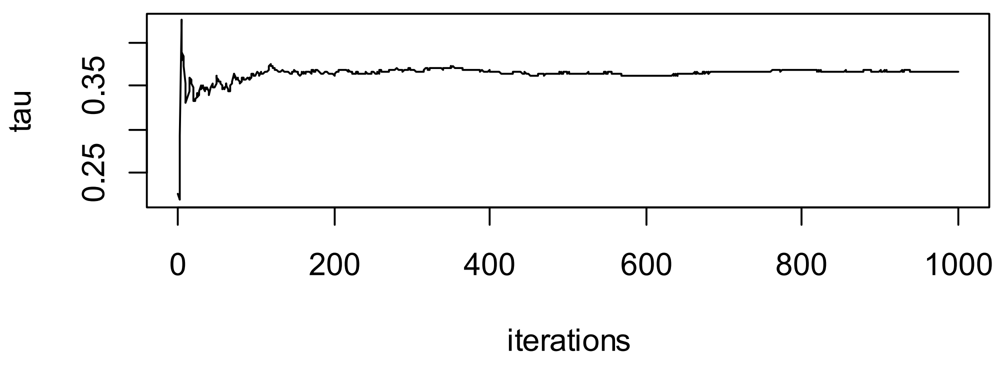
Ergotic mean plots

Εκτελούμε τις σχετικές εντολές στην R από το αρχείο που εξήγαμε με την επιλογή CODA.

```
modellog.sim<-read.coda("output.odcCODAchain1","output.odcCODAindex",quiet=FALSE)

par(mfrow=c(2,1))

x<-modellog.sim[,"mu"]
plot( cumsum(x)/1:length(x), type='l',xlab="iterations",  ylab=" mu " )

x<-modellog.sim[,"tau"]
plot( cumsum(x)/1:length(x), type='l',xlab="iterations",  ylab=" tau " )
```

Διάγραμμα 15 - "Ergotic mean trace plots" των mu, tau.





Στο Διάγραμμα 15 παρατηρούμε ότι οι εκτιμήσεις (μέσος των τιμών των posterior κατανομών) των δύο παραμέτρων που έχουμε σταθεροποιούνται γρήγορα.

Ερώτημα 4^0 (d)

Στο τελευταίο μέρος της ανάλυσής μας, θα συγκρίνουμε όλα τα μοντέλα που έχουμε προσαρμόσει μέχρι στιγμής. Για αυτό το σκοπό θα συμβουλευτούμε το κριτήριο του DIC (*Deviance Information Criterion*) . Όπως και με τα υπόλοιπα κριτήρια, μικρότερες τιμές του DIC φανερώνουν και καλύτερη προσαρμογή και προβλεπτική ικανότητα.

Όπως προαναφέραμε, το τελευταίο μοντέλο που προσαρμόσαμε δεν έχει νόημα να συγκριθεί με οποιοδήποτε κριτήριο από τη στιγμή που η εξαρτημένη μεταβλητή είναι διαφορετική. Θα χρησιμοποιήσουμε ωστόσο το κριτήριο στη log-Normal κατανομή που προσαρμόσαμε και επί της ουσίας δε διαφέρει.

Λόγω του ότι παρατηρήσαμε ασυμμετρία σε κάποιες από της posterior κατανομές, θα υπολογίσουμε την τιμή του DIC χρησιμοποιώντας τον τύπο των *Gelman et al. (2004).*

Αυτό μπορεί να γίνει εύκολα στην R, με τον παρακάτω υπολογισμό:

0.5*var(D) + mean(D), όπου D η Deviance του μοντέλου.

Έχουμε ήδη τρέξει τα διαφορετικά μοντέλα και στην R, μέσω του πακέτου ‘R2WinBUGS’ *(για λεπτομέρειες για τις εντολές στην R, βλ. Παράρτημα, Εντολές για την R)*.

Η R υπολογίζει και αποθηκεύει τα αποτελέσματα της Deviance στο “sims.matrix” που επιστρέφει η συνάρτηση “bugs”.

```
#dic model1.exp

0.5*var(model1.exp$sims.matrix[,"deviance"])
+mean(model1.exp$sims.matrix[,"deviance"])
```

```
#dic model2.gamma

0.5*var(model2.gamma$sims.matrix[,"deviance"])
+mean(model2.gamma$sims.matrix[,"deviance"])

#dic model3.lnorm

0.5*var(model3.lnorm$sims.matrix[,"deviance"])
+mean(model3.lnorm$sims.matrix[,"deviance"])
```

Τα αποτελέσματα που έχουμε φαίνονται παρακάτω.

Πίνακας 5- Τιμές του DIC για τα διαφορετικά μοντέλα

Κατανομή	**Εκθετική**	**Γάμμα**	**log-Normal**
DIC	13.49714	13.51169	13.48527

Παρατηρούμε ότι το μοντέλο με τη μικρότερη τιμή του DIC είναι το log-Normal μοντέλο *(Πίνακας 5, DIC=13.48527)*. Ωστόσο, η διαφορά μεταξύ των τιμών του DIC και στα τρία μοντέλα είναι πάρα πολύ μικρή, κατά πολύ μικρότερη του γενικού κανόνα των 5 μονάδων. Αυτό σημαίνει ότι και τα τρία μοντέλα είναι παρόμοια και έχουν την ίδια προβλεπτική ικανότητα. Μπορούμε δηλαδή, να χρησιμοποιήσουμε οποιοδήποτε από τα μοντέλα που προσαρμόσαμε (συμπεριλαμβανομένου και του μοντέλου της Κανονικής Κατανομής των λογαρίθμων το οποίο δίνει την ίδια σχεδόν πληροφορία με το log-Normal), για την εκτίμηση μελλοντικών/άγνωστων τιμών.

ΠΑΡΑΡΤΗΜΑ

Εντολές για την R

Για την προσαρμογή των μοντέλων μέσω της R, ακολουθήσαμε την παρακάτω διαδικασία.

Αρχικά διαβάσαμε τα αντίστοιχα πακέτα.

library(R2WinBUGS)

library(BRugs)

Περάσαμε τα δεδομένα μας.

data<-list(Y=c(0.4, 0.01, 0.2, 0.1, 2.1, 0.1, 0.9, 2.4, 0.1, 0.2),n=10)

Αποθηκεύσαμε το “directory” του προγράμματος “OpenBUGS” στον υπολογιστή μας.

openbugs.dir <- "C:/Program Files (x86)/OpenBUGS"

Για το μοντέλο της Εκθετικής Κατανομής, εφαρμόσαμε τις εξής εντολές.

Ορίσαμε τις αρχικές τιμές.

Inits1← list(list(v=0.0)   )

Στη συνέχεια, ορίσαμε τα ονόματα των παραμέτρων για τις οποίες θέλουμε να παράξουμε εκτιμήσεις (‘monitor’).

parameter.names <- c('theta', 'v', 'c')

Τρέξαμε το μοντέλο μέσω της συνάρτησης “bugs” και έχοντας πρώτα αποθηκεύσει το μοντέλο σε κώδικα “OpenBUGS” σε ένα txt αρχείο.

```
model1.exp <- bugs( data, inits1, model.file = "modelexp.txt",
        parameters = parameter.names,n.chains = 1,
      n.iter = 1500, n.burnin=500, n.thin=1,
      bugs.directory = openbugs.dir, debug=F, program="OpenBUGS")
```

Για την δημιουργία των γραφημάτων “trace plots” και “acf plots”, εκτελέσαμε τις εντολές που ακολουθούν.

```
par(mfrow=c(2,1))

# Trace plot of  "theta"

plot(model1.exp$sims.matrix[,"theta"],type='l',main="Trace plot of theta"

        ,xlab="iterations",ylab="theta")


#acf plot

# Autocorrelation of theta

acf(model1.exp$sims.matrix[,"theta"],main="ACF plot of theta",lag=100)
```

Για τα “density plots” των παραμέτρων.

```
par(mfrow=c(2,2))

# Posterior density of "theta"

plot(density(model1.exp$sims.matrix[,"theta"]),main="theta")


# Posterior density of "v"

plot(density(model1.exp$sims.matrix[,"v"]),main="v")


# Posterior density of "c"

plot(density(model1.exp$sims.matrix[,"c"]),,main="c")
```

Ομοίως για τα υπόλοιπα μοντέλα, ακολουθήθηκαν οι ίδιες διαδικασίες.

Για το μοντέλο της Γάμμα Κατανομής εκτελέσαμε τις εντολές που ακολουθούν.

```
# initial values

inits1<-list(

        list(a=1.0, b=1.0) )
```

```
# defining the names of the parameters we wish to monitor
parameter.names <- c( 'a', 'b')

# generating random samples using OpenBUGS
model2.gamma <- bugs( data, inits1, model.file = "modelgamma.txt",
              parameters = parameter.names,n.chains = 1,
          n.iter = 1500, n.burnin=500, n.thin=1,
          bugs.directory = openbugs.dir, debug=F, program="OpenBUGS")

#trace plots
par(mfrow=c(2,1))
# Trace plot of  "a"
plot(model2.gamma$sims.matrix[,"a"],type='l',main="Trace plot of a"
      ,xlab="iterations",ylab="a")
# Trace plot of  "b"
plot(model2.gamma$sims.matrix[,"b"],type='l',main="Trace plot of b"
      ,xlab="iterations",ylab="b")

#acf plots
par(mfrow=c(2,1))
acf(model2.gamma$sims.matrix[,"a"],main="ACF plot of a",lag=100)
acf(model2.gamma$sims.matrix[,"b"],main="ACF plot of b",lag=100)

#density plots

par(mfrow=c(1,2))
# Posterior density of "a"
plot(density(model2.gamma$sims.matrix[,"a"]),main="a")
```

```
# Posterior density of "b"
plot(density(model2.gamma$sims.matrix[,"b"]),main="b")
```

Ομοίως, για το μοντέλο της log-Normal κατανομής.

```
# initial values
inits1<-list(
        list(mu=0.0, tau=1.0) )

# defining the names of the parameters we wish to monitor
parameter.names <- c( 'mu', 'tau' , 's2', 's')

model3.lnorm <- bugs( data, inits1, model.file = "modellnorm.txt",
            parameters = parameter.names,n.chains = 1,
          n.iter = 1500, n.burnin=500, n.thin=1,
          bugs.directory = openbugs.dir, debug=F, program="OpenBUGS")

#trace plots
par(mfrow=c(2,1))
# Trace plot of  "mu"
plot(model3.lnorm$sims.matrix[,"mu"],type='l',main="Trace plot of mu"
      ,xlab="iterations",ylab="mu")

# Trace plot of  "tau"
plot(model3.lnorm$sims.matrix[,"tau"],type='l',main="Trace plot of tau"
      ,xlab="iterations",ylab="tau")
```

```
#acf plots
par(mfrow=c(2,1))
acf(model3.lnorm$sims.matrix[,"mu"],main="ACF plot of mu",lag=100)
acf(model3.lnorm$sims.matrix[,"tau"],main="ACF plot of tau",lag=100)

#density plots
par(mfrow=c(1,2))
# Posterior density of "a"
plot(density(model2.gamma$sims.matrix[,"a"]),main="a")

# Posterior density of "b"
plot(density(model2.gamma$sims.matrix[,"b"]),main="b")
```

Τέλος, για το μοντέλο της Κανονικής Κατανομής στους λογαρίθμους των δεδομένων εκτελέσαμε τις παρακάτω εντολές.

```
# initial values
inits1<-list(
     list(mu=0.0, tau=1.0) )

# defining the names of the parameters we wish to monitor
parameter.names <- c( 'mu', 'tau' , 's2')

model4.log <- bugs( data, inits1, model.file = "modellog.txt",
        parameters = parameter.names,n.chains = 1,
      n.iter = 1500, n.burnin=500, n.thin=1,
      bugs.directory = openbugs.dir, debug=F, program="OpenBUGS")
```

```
#trace plot

par(mfrow=c(2,1))

# Trace plot of  "mu"

plot(model4.log$sims.matrix[,"mu"],type='l',main="Trace plot of mu"

       ,xlab="iterations",ylab="mu")


# Trace plot of  "tau"

plot(model4.log$sims.matrix[,"tau"],type='l',main="Trace plot of tau"

       ,xlab="iterations",ylab="tau")


par(mfrow=c(2,1))

#acf plots

acf(model4.log$sims.matrix[,"mu"],main="ACF plot of mu",lag=100)

acf(model4.log$sims.matrix[,"tau"],main="ACF plot of tau",lag=100)



#density plots


par(mfrow=c(2,2))

# Posterior density of "mu"

plot(density(model4.log$sims.matrix[,"mu"]),main="mu")


# Posterior density of "tau"

plot(density(model4.log$sims.matrix[,"tau"]),main="tau")


# Posterior density of "s2"

plot(density(model4.log$sims.matrix[,"s2"]),main="s2")
```