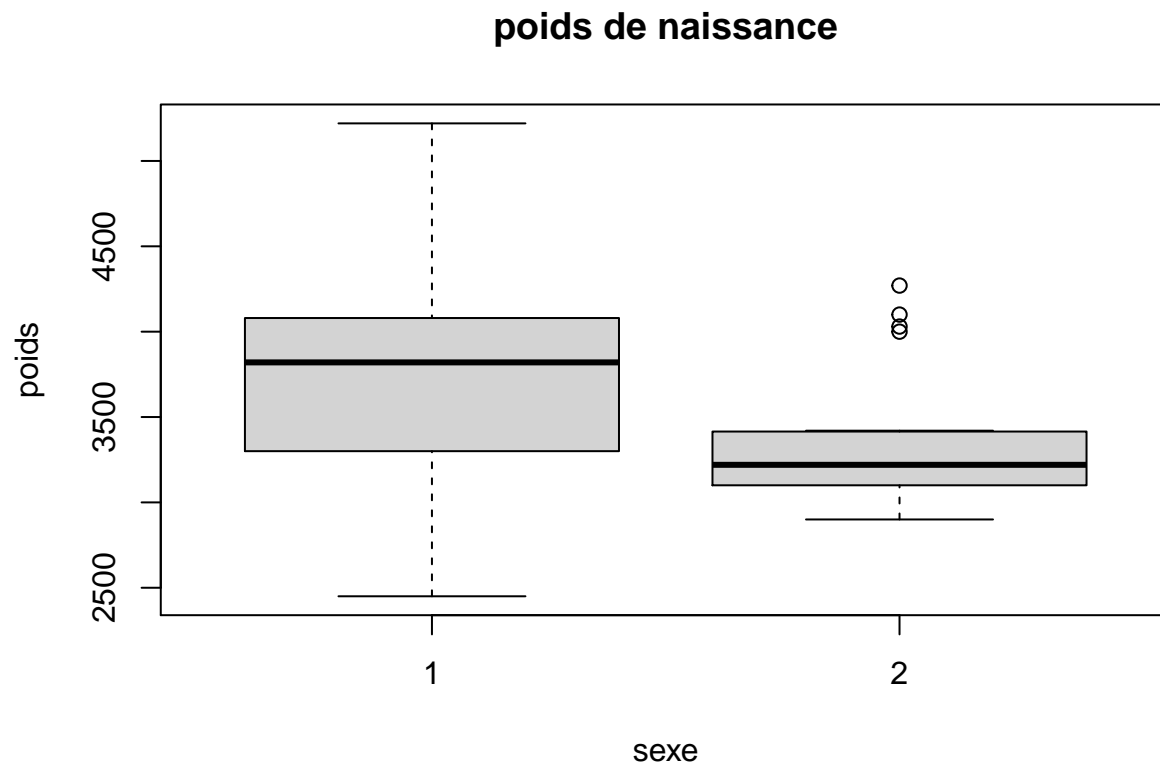


# Statistique bayésienne avec R

## Exercice sur les poids de naissance

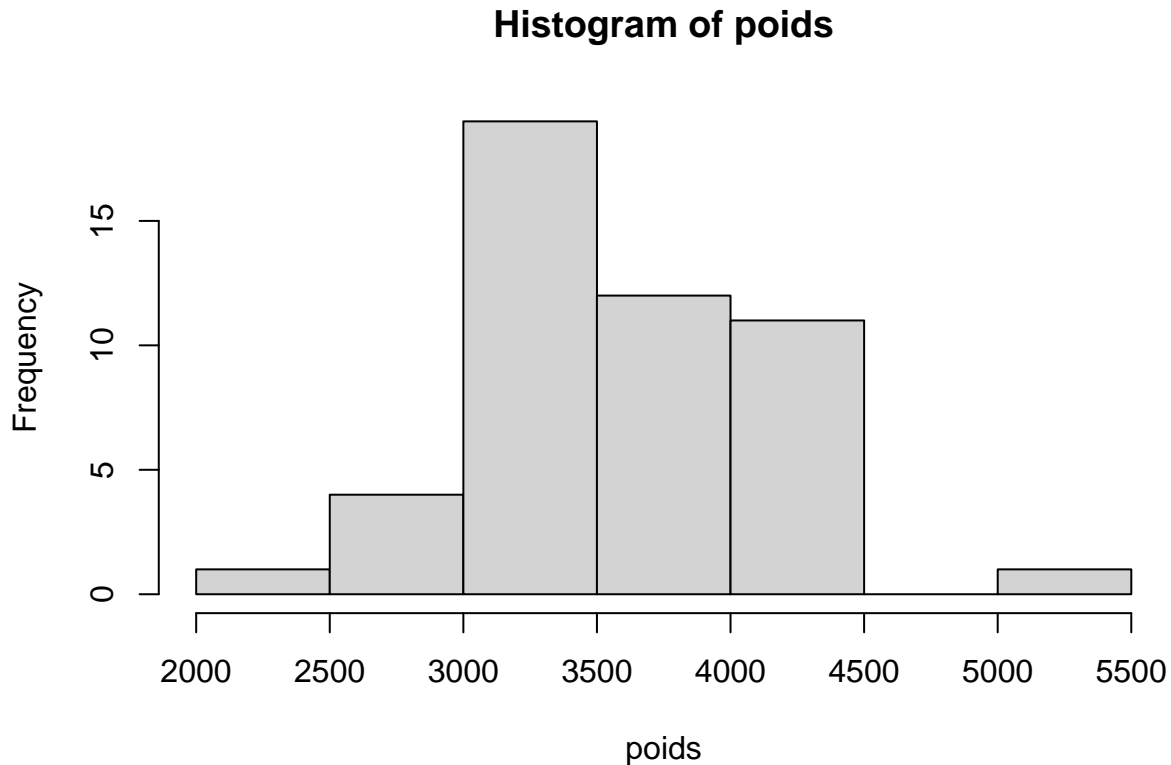
Julien JACQUES

```
data=read.table('Rcode/poidsnaissance.txt',header = T,sep=',',row.names = 1)
data$OBS=NULL
sexe=data$SEXE+1
poids=data$POIDNAIS
boxplot(poids~sexe,main="poids de naissance")
```



L'histogramme du poids de naissance ressemble à peu près à une loi gaussienne, ce qui est confirmé par le test de Shapiro

```
hist(poids)
```



```
shapiro.test(poids)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  poids
## W = 0.96218, p-value = 0.1237
```

## Estimation bayésienne

### Estimer le poids de naissance moyen

De façon fréquentiste :

```
mean(poids)
```

```
## [1] 3590
```

Les données étant distribuées suivant une loi gaussienne, nous allons choisir un a priori conjugué gaussien. Il nous reste à déterminer les moyennes et variances a priori.

L'histogramme nous donne l'idée d'un a priori gaussien centré en 3250g. Pour l'écart-type, il va traduire la confiance que l'on a dans notre a priori.

Nous avons vu que le MAP est alors donné par :

$$\hat{\theta} = E[\theta|\underline{x}] = \frac{\tau^2 \frac{\sigma^2}{n}}{\tau^2 + \frac{\sigma^2}{n}} \left( \frac{\bar{x}}{\frac{\sigma^2}{n}} + \frac{\mu}{\tau^2} \right)$$

dont on peut coder le calcul. En jouant sur la valeur de l'écart-type a priori  $\tau$ , on pourra jouer sur la confiance en notre a priori et examiner son influence sur le MAP

```
s=sd(poids)
tau=100
n=length(poids)
MAP=(tau^2*s^2/n)/(tau^2+s^2/n)*(mean(poids)/(s^2/n)+3250/(tau^2))
print(MAP)
```

```
## [1] 3461.82
```

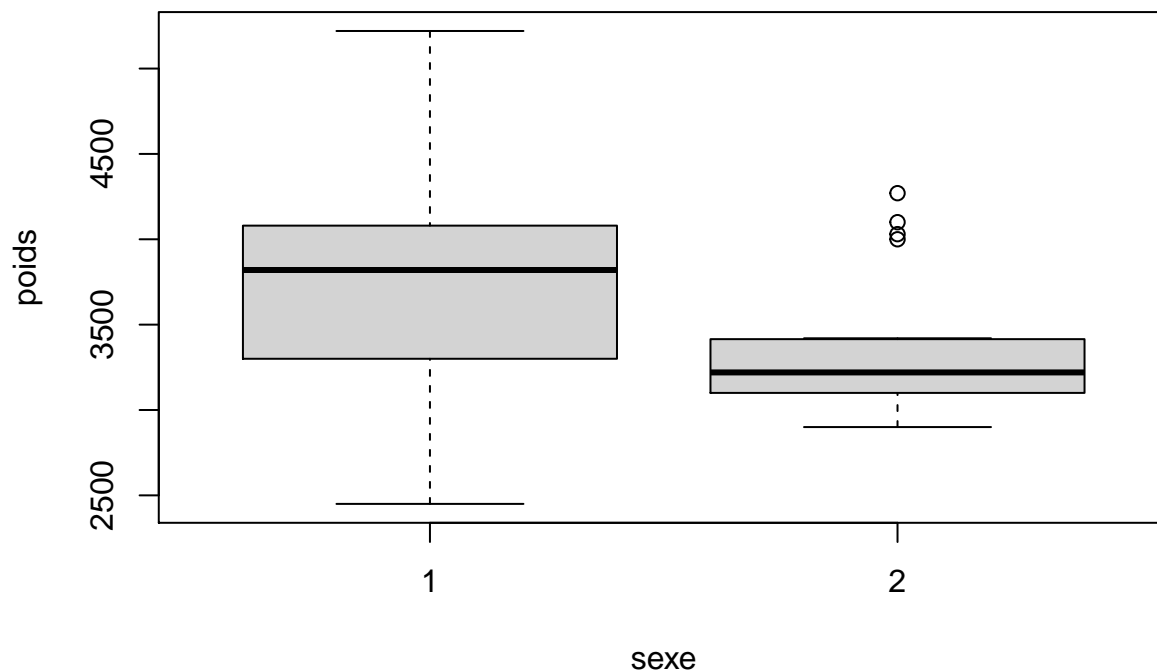
Les calculs sont faits ici *à la main*, nous verrons plus tard comment les faire sous R.

## Test d'hypothèse et comparaison bayésienne de modèles

les garçons sont-ils plus lourds que les filles à la naissance ?

De façon fréquentiste c'est significatif,

```
boxplot(poids~sexe)
```



```
t.test(poids~sexe,alternative='greater')
```

```
##
## Welch Two Sample t-test
##
## data:  poids by sexe
## t = 2.4522, df = 45.553, p-value = 0.009047
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  110.0071      Inf
## sample estimates:
## mean in group 1 mean in group 2
##      3728.103      3379.211
```

Pour la version bayésienne, on indiquera l'unilatéralité de l'hypothèse alternative en spécifiant `nullInterval=c(0, Inf)`

```
library(BayesFactor)
ttestBF(poids[sexe==1],poids[sexe==2],nullInterval=c(0, Inf))
```

```
## Bayes factor analysis
## -----
## [1] Alt., r=0.707 0<d<Inf      : 4.525689 ±0%
## [2] Alt., r=0.707 !(0<d<Inf) : 0.1017375 ±0%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

Le facteur de Bayes faut 4.525689

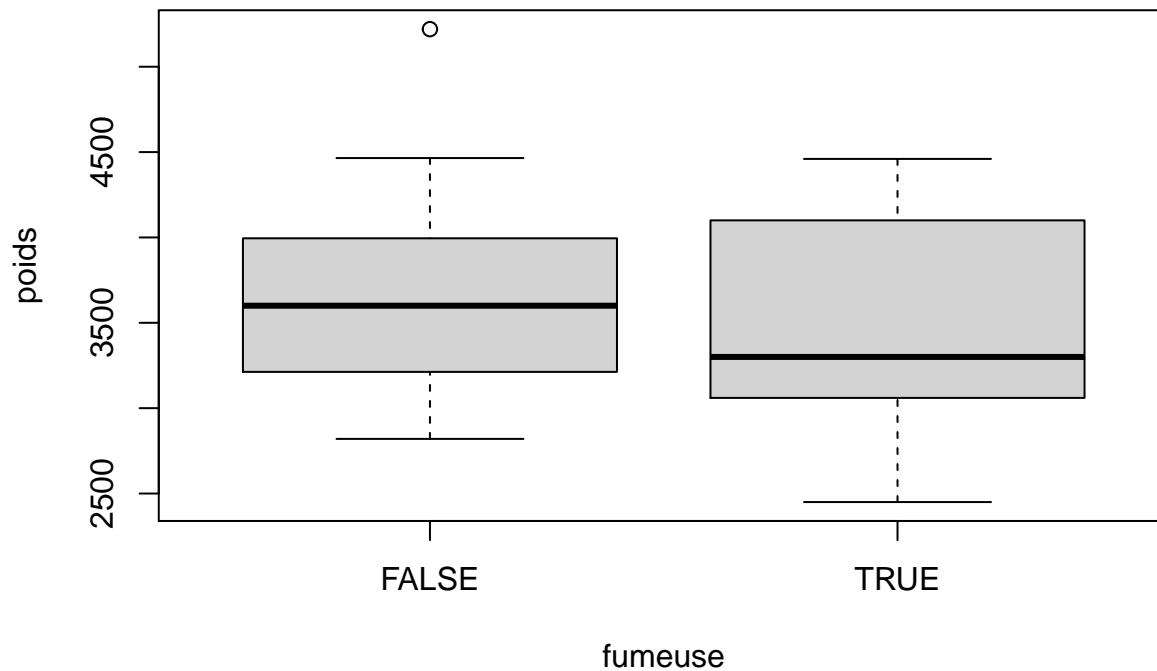
```
log10(4.525689)
```

```
## [1] 0.6556847
```

De façon bayésienne, c'est *substantielle* (!), il est difficile de conclure à la supériorité de poids des garçons par rapport aux filles.

**le poids de naissance dépend-il du fait que la mère soit fumeuse ?**

```
fumeuse=data$CIGJOUR>0
boxplot(poids~fumeuse)
```



```
t.test(poids~fumeuse,alternative='greater')
```

```
##
## Welch Two Sample t-test
##
## data:  poids by fumeuse
## t = 0.53947, df = 10.414, p-value = 0.3005
```

```
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
## -295.1788      Inf
## sample estimates:
## mean in group FALSE  mean in group TRUE
##      3613.590      3487.778
```

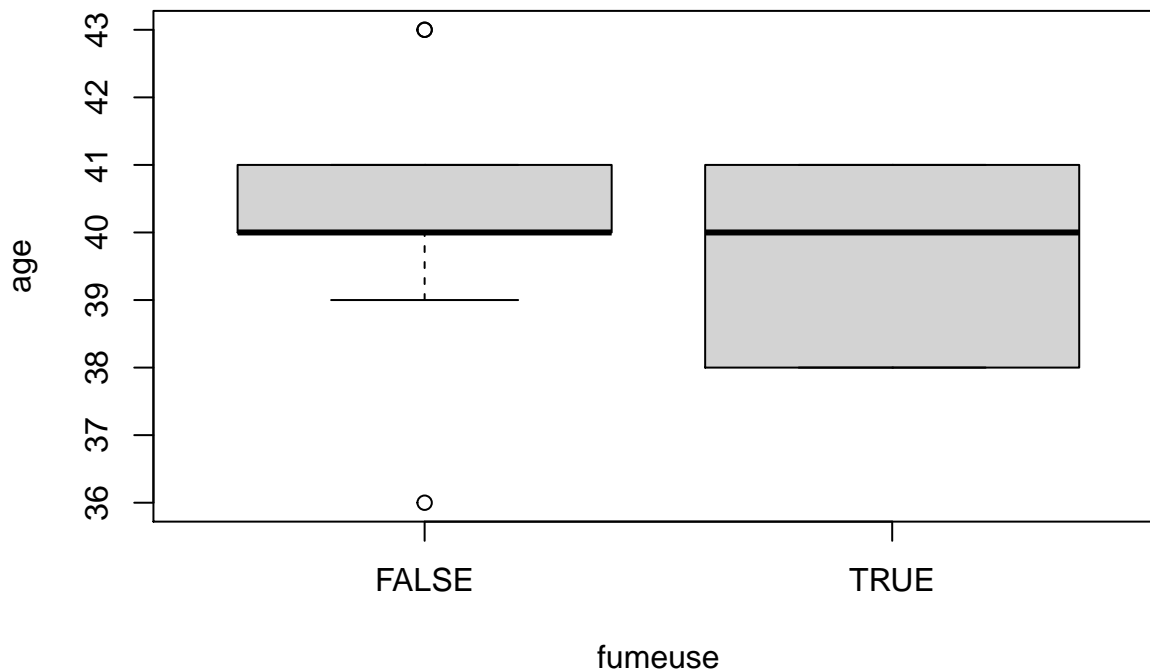
```
ttestBF(poids[fumeuse],poids[!fumeuse],nullInterval=c(-Inf,0))
```

```
## Bayes factor analysis
## -----
## [1] Alt., r=0.707 -Inf<d<0      : 0.5668229 ±0.02%
## [2] Alt., r=0.707 !(-Inf<d<0) : 0.2385683 ±0%
##
## Against denominator:
##   Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

Non significatif.

l'âge gestationnel dépend-il du fait que la mère soit fumeuse ?

```
age=data$AGEGEST
boxplot(age~fumeuse)
```



```
t.test(age~fumeuse,alternative='greater')
```

```
##
## Welch Two Sample t-test
##
## data: age by fumeuse
## t = 0.96285, df = 11.094, p-value = 0.1781
## alternative hypothesis: true difference in means is greater than 0
```

```
## 95 percent confidence interval:
## -0.3986471      Inf
## sample estimates:
## mean in group FALSE mean in group TRUE
##      40.12821      39.66667
ttestBF(age[fumeuse],age[!fumeuse],nullInterval=c(-Inf,0))
```

```
## Bayes factor analysis
## -----
## [1] Alt., r=0.707 -Inf<d<0 : 0.84529 ±0%
## [2] Alt., r=0.707 !(-Inf<d<0) : 0.1953459 ±0%
##
## Against denominator:
## Null, mu1-mu2 = 0
## ---
## Bayes factor type: BFindepSample, JZS
```

Non significatif.

## Régression linéaire

Effectuer une régression du poids de naissance en fonction des autres variables disponibles

De façon fréquentiste

```
m1=lm(POIDNAIS~.,data=data)
summary(m1)
```

```
##
## Call:
## lm(formula = POIDNAIS ~ ., data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1039.65  -192.17   -33.47   187.49  1277.90
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  621.107    3050.952   0.204  0.83969
## AGEGEST      164.131     57.489   2.855  0.00672 **
## SEXE        -422.069    145.419  -2.902  0.00593 **
## CIGJOUR       -2.552     5.493  -0.465  0.64472
## TAILMERE     -28.560     14.524  -1.966  0.05605 .
## POIDAVG      -17.229     17.021  -1.012  0.31737
## POIDFING      30.330     13.433   2.258  0.02934 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 459.7 on 41 degrees of freedom
## Multiple R-squared:  0.3654, Adjusted R-squared:  0.2725
## F-statistic: 3.935 on 6 and 41 DF, p-value: 0.003388
m2=step(m1)
```

```

## Start: AIC=594.96
## POIDNAIS ~ AGEGEST + SEXE + CIGJOUR + TAILMERE + POIDAVG + POIDFING
##
##           Df Sum of Sq      RSS      AIC
## - CIGJOUR   1     45599  8708906  593.22
## - POIDAVG   1     216495  8879803  594.15
## <none>                                8663308  594.96
## - TAILMERE   1     817055  9480362  597.29
## - POIDFING   1    1077169  9740477  598.59
## - AGEGEST    1    1722285 10385593  601.67
## - SEXE       1    1780026 10443334  601.93
##
## Step: AIC=593.22
## POIDNAIS ~ AGEGEST + SEXE + TAILMERE + POIDAVG + POIDFING
##
##           Df Sum of Sq      RSS      AIC
## - POIDAVG   1     194586  8903492  592.28
## <none>                                8708906  593.22
## - TAILMERE   1     820823  9529730  595.54
## - POIDFING   1    1031572  9740478  596.59
## - SEXE       1    1821002 10529908  600.33
## - AGEGEST    1    1864248 10573155  600.53
##
## Step: AIC=592.28
## POIDNAIS ~ AGEGEST + SEXE + TAILMERE + POIDFING
##
##           Df Sum of Sq      RSS      AIC
## <none>                                8903492  592.28
## - TAILMERE   1     846954  9750446  594.64
## - POIDFING   1    1429451 10332943  597.42
## - AGEGEST    1    1929166 10832659  599.69
## - SEXE       1    2367275 11270767  601.59
summary(m2)

##
## Call:
## lm(formula = POIDNAIS ~ AGEGEST + SEXE + TAILMERE + POIDFING,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1168.11  -184.86   -20.26   186.56  1216.36
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   306.933    3000.300   0.102  0.91899
## AGEGEST       171.179     56.080   3.052  0.00388 **
## SEXE        -465.769    137.750  -3.381  0.00155 **
## TAILMERE      -29.063     14.370  -2.022  0.04937 *
## POIDFING       18.413      7.008   2.627  0.01188 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 455 on 43 degrees of freedom

```

```
## Multiple R-squared:  0.3478, Adjusted R-squared:  0.2871
## F-statistic: 5.733 on 4 and 43 DF,  p-value: 0.0008655
```

De façon bayésienne

```
y=data$POIDNAIS
x=as.matrix(data[, -3])
summary(lm(y~x))
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1039.65  -192.17   -33.47   187.49  1277.90
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   621.107    3050.952   0.204  0.83969
## xAGEGEST      164.131     57.489   2.855  0.00672 **
## xSEXE        -422.069    145.419  -2.902  0.00593 **
## xCIGJOUR       -2.552     5.493  -0.465  0.64472
## xTAILMERE     -28.560     14.524  -1.966  0.05605 .
## xPOIDAVG      -17.229     17.021  -1.012  0.31737
## xPOIDFING      30.330     13.433   2.258  0.02934 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 459.7 on 41 degrees of freedom
## Multiple R-squared:  0.3654, Adjusted R-squared:  0.2725
## F-statistic: 3.935 on 6 and 41 DF,  p-value: 0.003388
```

Je vais introduire un a priori sur le fait que fumer une 10 cigarettes par jour fait perdre 100g au poids de naissance (donc -10g par cigarette) et que les garçons sont aussi lourds que les filles. Comme je ne veux pas toucher aux autres variables, je donne comme a priori l'estimation du max de vraisemblance. Et comme je suis sûr de moi, je mets  $g = 1$

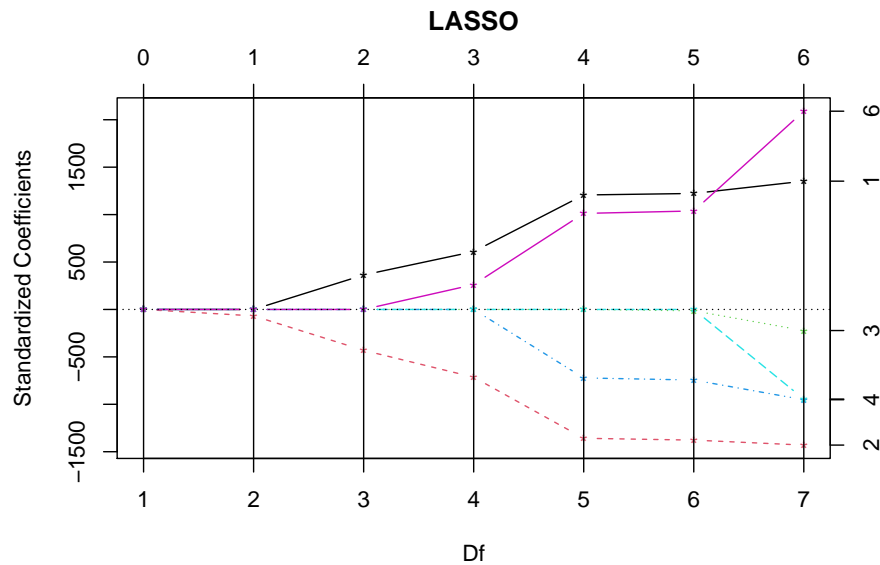
```
library(bayess)
res1=BayesReg(y,x,betatilde = c(164,-422,-10,-28,-17,30),g=1)
```

```
##
##              PostMean PostStError Log10bf EvidAgaH0
## Intercept 3590.0000      70.5637
## x1        179.6175      51.4271  1.3474      (***)
## x2       -314.2020      53.4801  0.5897      (**)
## x3       -21.0516      51.9705 -0.1176
## x4       -82.7004      52.5473  0.3658      (*)
## x5       -76.6759     101.3032 -0.0246
## x6       165.8403     100.4829  0.4398      (*)
##
##
## Posterior Mean of Sigma2: 239003.0691
## Posterior StError of Sigma2: 341820.7276
```

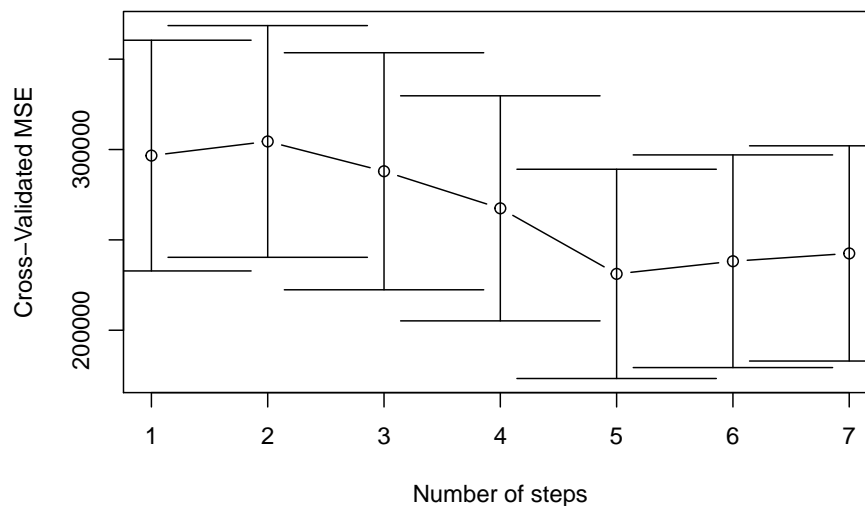
Avec une régression LASSO



```
library('lars')
model_lasso=lars(x,y,type="lasso",trace=F,normalize=TRUE)
plot(model_lasso,xvar='df', plottype='coeff')
```



```
cv=cv.lars(x,y,K=48,
           trace=F,plot.it=T,se=T,type=c("lasso"),
           mode='step',normalize=TRUE)
```



Le minimum est à l'étape 5, néanmoins l'étape 4 donne une erreur pas significativement plus grande que celle de l'étape 5...

```
print(model_lasso$lambda[4])
```

```
## [1] 581.9842
```

```
print(model_lasso$beta[4,])
```

```
##      AGEGEST      SEXE      CIGJOUR      TAILMERE      POIDAVG      POIDFING
## 73.993249 -210.877346  0.000000    0.000000    0.000000    3.797626
```

Seules 3 variables sont sélectionnées (agegest, sexe, poidfing). Pour terminer, nous ré-estimons un modèle de régression classique sur ces 3 variables :

```

m3=lm(POIDNAIS~AGEGEST+SEXE+POIDFING,data=data)
summary(m3)

##
## Call:
## lm(formula = POIDNAIS ~ AGEGEST + SEXE + POIDFING, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1244.88  -228.25   -32.43   213.42  1429.99
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3601.977   2374.076  -1.517  0.13637
## AGEGEST      156.539     57.531   2.721  0.00929 **
## SEXE        -441.818    141.978  -3.112  0.00326 **
## POIDFING      14.280      6.935   2.059  0.04542 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 470.7 on 44 degrees of freedom
## Multiple R-squared:  0.2858, Adjusted R-squared:  0.2371
## F-statistic: 5.868 on 3 and 44 DF,  p-value: 0.001842

```

## Modélisation hiérarchique

On commence par définir les données

```
dat <- list(poids = poids, sexe = sexe, N = length(poids))
```

Puis 3 initialisations différentes

```
inits <- list( list(moyennes = c(2600, 4000), sigma = 500), list(moyennes = c(4500, 2700), sigma = 700),
```

On définit le modèle

```
library(rjags)
```

```
## Linked to JAGS 4.3.0
```

```
## Loaded modules: basemod,bugs
```

```
m1 <- jags.model('Rcode/modelepoidsnaissance.txt', data = dat, inits = inits, n.chains = 3, quiet=TRUE)
```

Puis on lance les itérations MCMC

```
update(m1, 3000, progress.bar="none")
```

```
mcmc1 <- coda.samples(m1, variable.names = c("moyennes", "sigma"), n.iter = 2000, progress.bar="none")
```

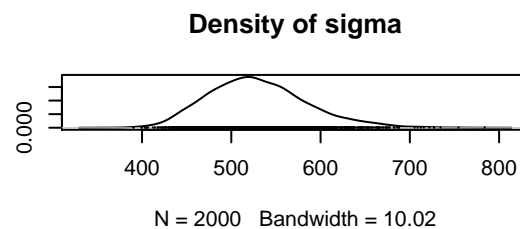
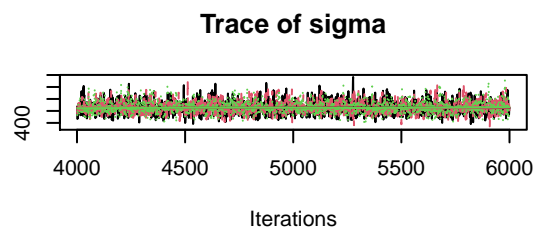
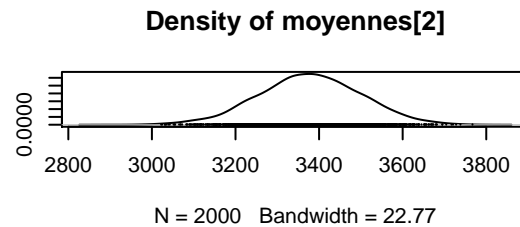
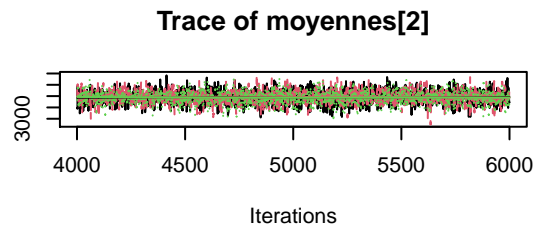
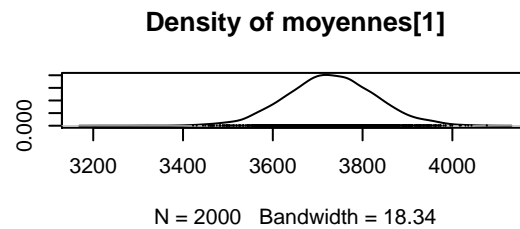
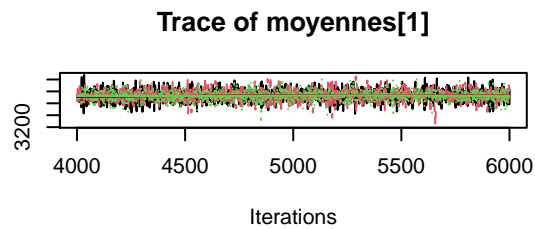
Comme résultat on peut regarder par exemple la moyenne du poids moyen des garçons :

```
mean(mcmc1[[1]][, "moyennes[1]"])
```

```
## [1] 3731.819
```

On peut représenter les chaînes MCMC

```
plot(mcmc1)
```



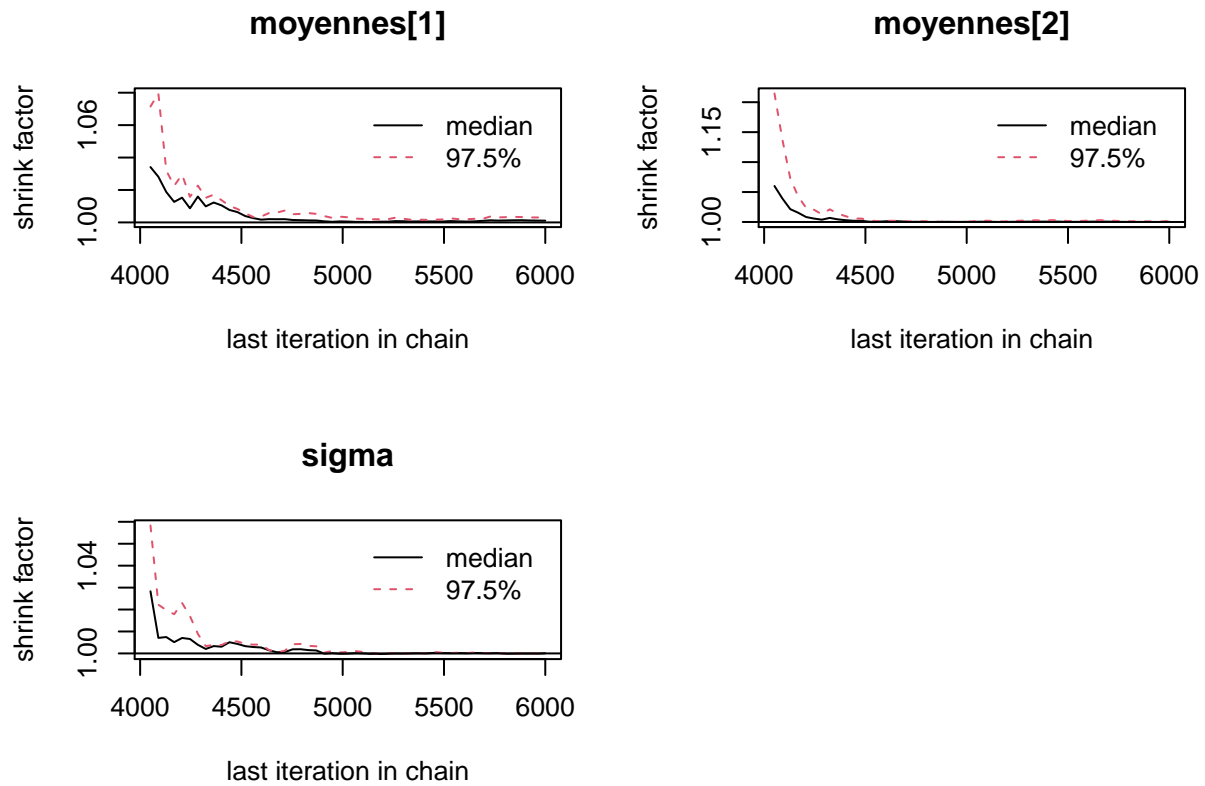
Les diag-

nostiques de convergence permettent de vérifier que la période de chauffe était suffisamment longue

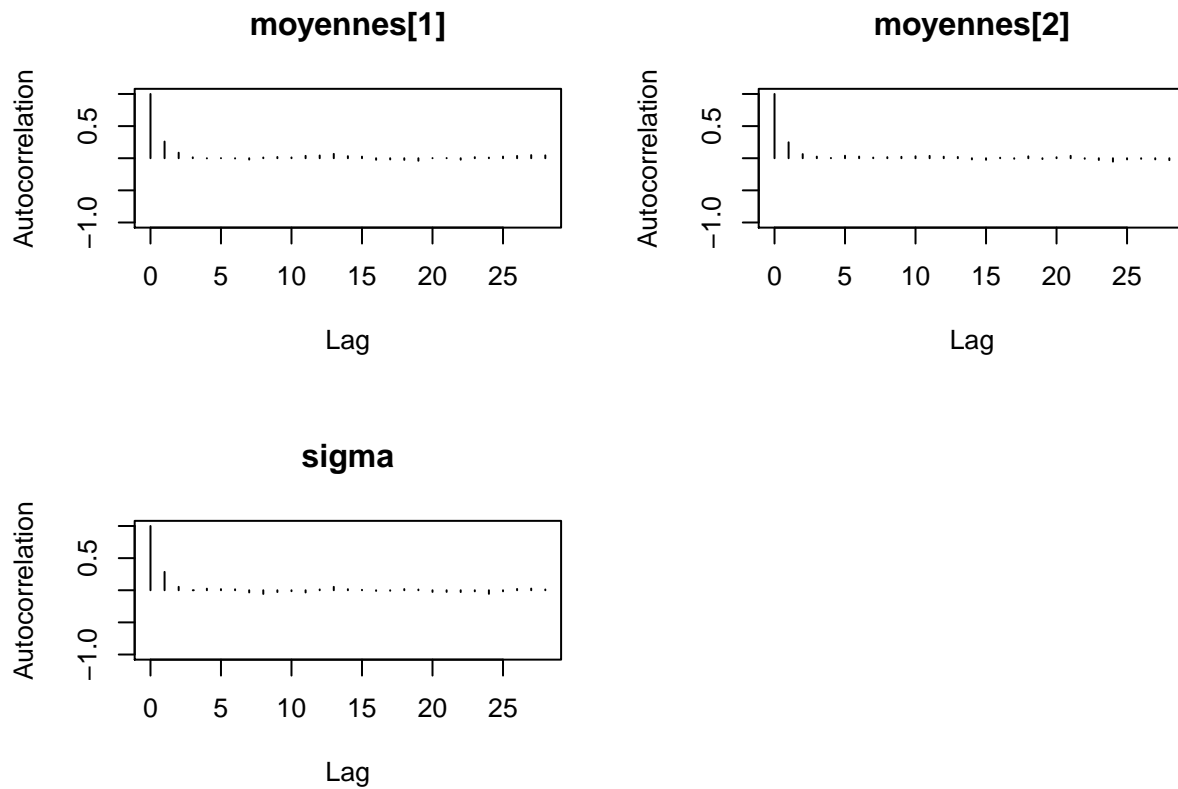
```
gelman.diag(mcmc1)
```

```
## Potential scale reduction factors:
##
##           Point est. Upper C.I.
## moyennes[1]           1           1
## moyennes[2]           1           1
## sigma                 1           1
##
## Multivariate psrf
##
## 1
```

```
gelman.plot(mcmc1)
```



```
autocorr.plot(mcmc1[[1]])
```



peut finalement examiner les résultats (loi a posteriori des paramètres) :

On

```
summary(mcmc1)
```

```
##
## Iterations = 4001:6000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean      SD Naive SE Time-series SE
## moyennes[1] 3729.2 100.27  1.2945      1.6936
## moyennes[2] 3378.0 122.42  1.5804      2.0321
## sigma       529.6  55.32  0.7142      0.9666
##
## 2. Quantiles for each variable:
##
##              2.5%   25%  50%  75%  97.5%
## moyennes[1] 3533 3664.1 3728 3796 3931.0
## moyennes[2] 3133 3297.0 3379 3461 3614.7
## sigma       436  490.8  525  563  652.8
```

## Modèle avec le sexe et la taille de la mère

On commence par définir les données

```
dat <- list(poids = poids, sexe = sexe, N = length(poids), taille=data$TAILMERE)
```

Puis 3 initialisations différentes

```
inits <- list( list(moyennes = c(2600, 4000), sigma = 500, beta=10),list(moyennes = c(4500, 2700), sigma = 500, beta=10),list(moyennes = c(3000, 3000), sigma = 500, beta=10))
```

On définit le modèle

```
library(rjags)
m1b <- jags.model('Rcode/modelepoidsnaissance4.txt', data = dat, inits = inits, n.chains = 3, quiet=TRUE)
```

Puis on lance les itérations MCMC

```
update(m1b, 3000, progress.bar="none")
mcmc1b <- coda.samples(m1b, variable.names = c("moyennes", "sigma", "beta"), n.iter = 2000, progress.bar="none")
```

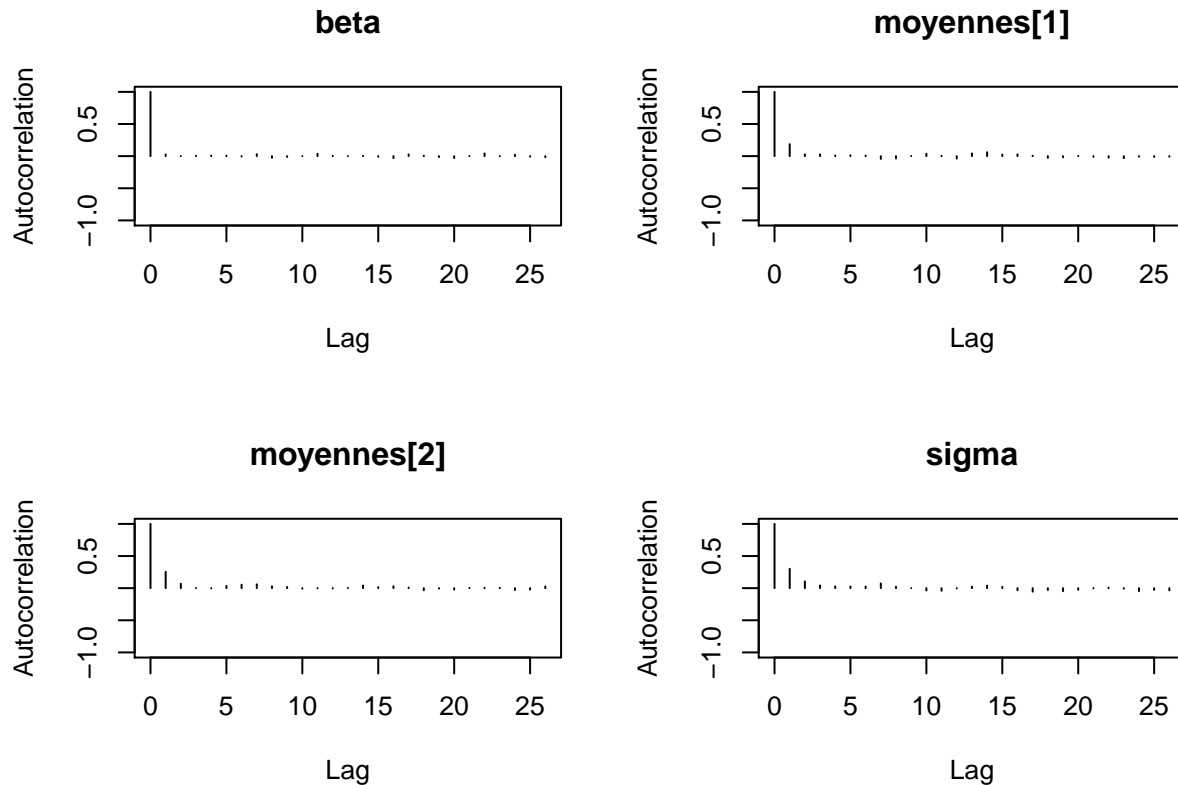
On vérifie que les chaînes ont bien convergées.

```
gelman.diag(mcmc1b)
```

```
## Potential scale reduction factors:
##
##              Point est. Upper C.I.
## beta              1              1
## moyennes[1]       1              1
## moyennes[2]       1              1
## sigma             1              1
##
## Multivariate psrf
##
```

```
## 1
```

```
autocorr.plot(mcmc1b[[1]])
```



Regardons les estimations

```
summary(mcmc1b)
```

```
##
## Iterations = 4001:6000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean      SD Naive SE Time-series SE
## beta           3.303    8.60  0.1110      0.1136
## moyennes[1] 3726.086  99.45  1.2838      1.6517
## moyennes[2] 3378.630 125.02  1.6140      2.0490
## sigma        534.292  57.14  0.7377      1.0201
##
## 2. Quantiles for each variable:
##
##              2.5%      25%      50%      75%     97.5%
## beta          -13.32   -2.704    3.203    9.241   20.06
## moyennes[1] 3528.81 3660.196 3727.332 3792.232 3919.58
## moyennes[2] 3133.36 3293.574 3378.172 3464.288 3625.05
## sigma         434.39  494.259  530.296  569.598  657.12
```

```
#plot(mcmc1b)
```

Le beta ne semble pas significatif.

Calculons le DIC pour comparer à celui du modèle sans la taille

```
dic.samples(m1,n.iter=1000)
```

```
## Mean deviance: 737.1
## penalty 3.237
## Penalized deviance: 740.3
```

```
dic.samples(m1b,n.iter=1000)
```

```
## Mean deviance: 737.8
## penalty 3.597
## Penalized deviance: 741.3
```

## Modèle avec le sexe, l'âge gestationnel et le nombre de cigarettes

On va faire de même mais en intégrant la variable agegest, supposé avoir un effet sur le poids  $\mu_{i,j}$ , ainsi que la variable cigjour supposé avec un effet elle sur la variable agegest

```
data=read.table('Rcode/poidsnaissance.txt',header = T,sep=',')
sexe=data$SEXE+1
poids=data$POIDNAIS
nbsemaines=data$AGEGEST
nbcig=data$CIGJOUR
```

definition des donnees

```
dat <- list(poids = poids, sexe = sexe, nbsemaines = nbsemaines, nbcig=nbcig, N = length(poids))
```

initialisation

```
inits <- list(
  list(moyennes = c(2600, 4000), sigma = 500, b=150, b2 = -1,b3=0),
  list(moyennes = c(4500, 2700), sigma = 700, b=100, b2 = 0,b3=0),
  list(moyennes = c(4000, 4000), sigma = 300, b=200, b2 = 1,b3=0))
```

definition modele

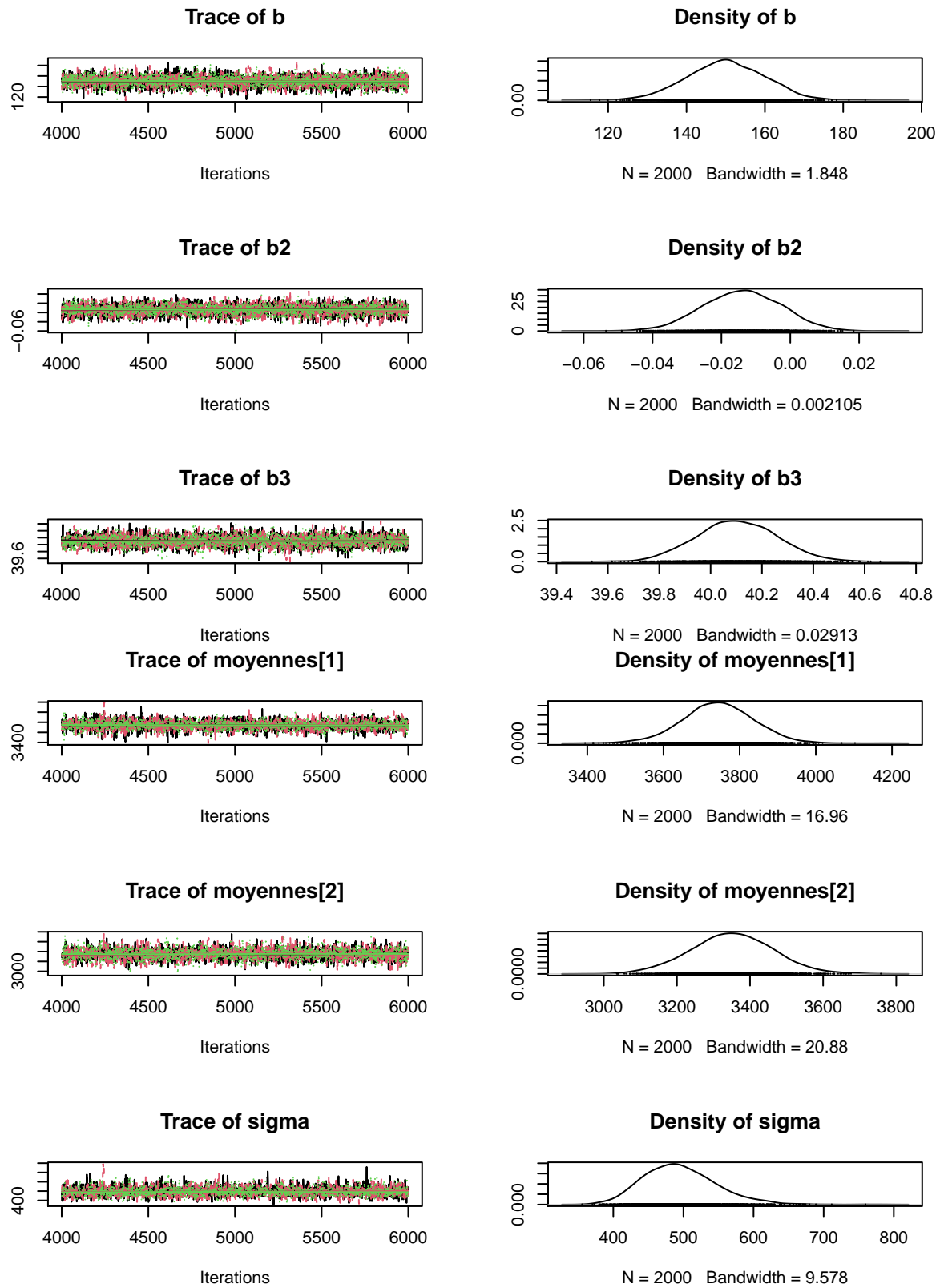
```
m2 <- jags.model('Rcode/modelepoidsnaissance2.txt', data = dat, inits = inits, n.chains = 3, quiet=TRUE)
```

iterations de MCMC

```
update(m2, 3000,progress.bar="none")
mcmc1 <- coda.samples(m2, variable.names = c("moyennes", "sigma","b","b2","b3"), n.iter = 2000,progress
```

représentation de la chaine

```
plot(mcmc1)
```



nostique de convergence

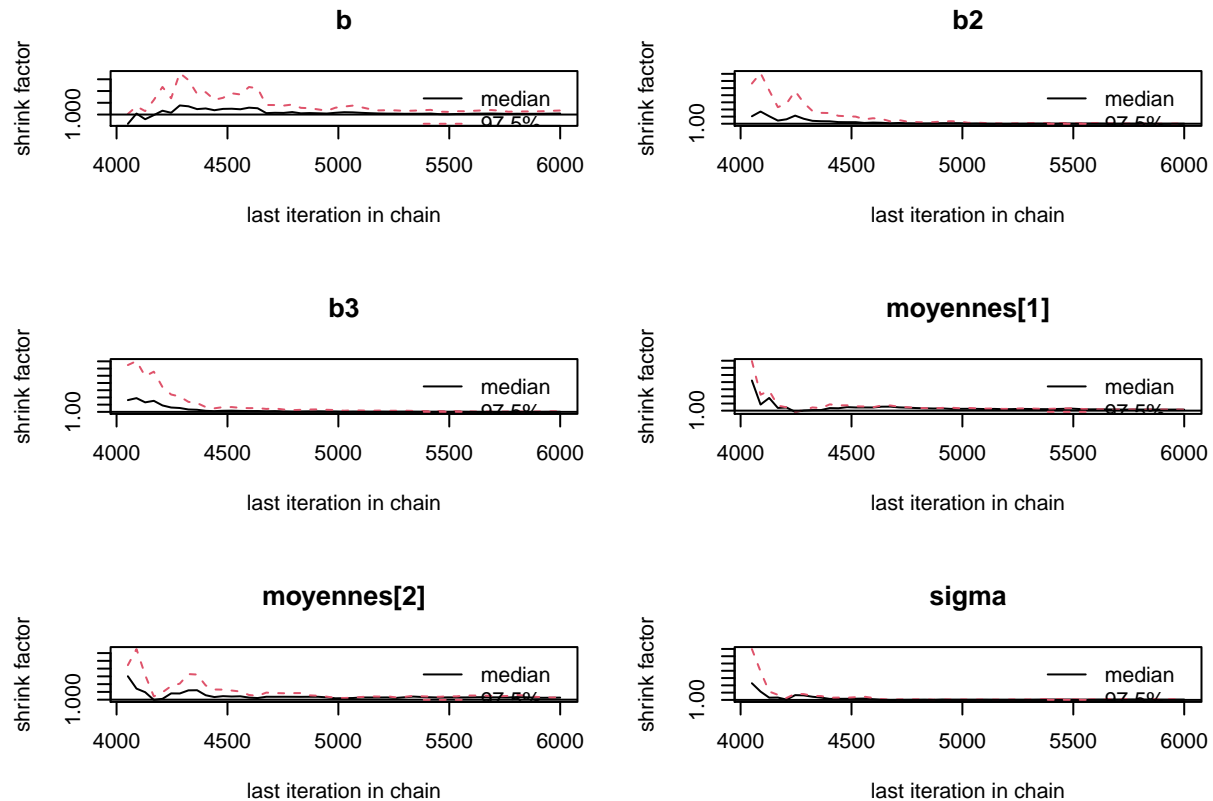
diag-



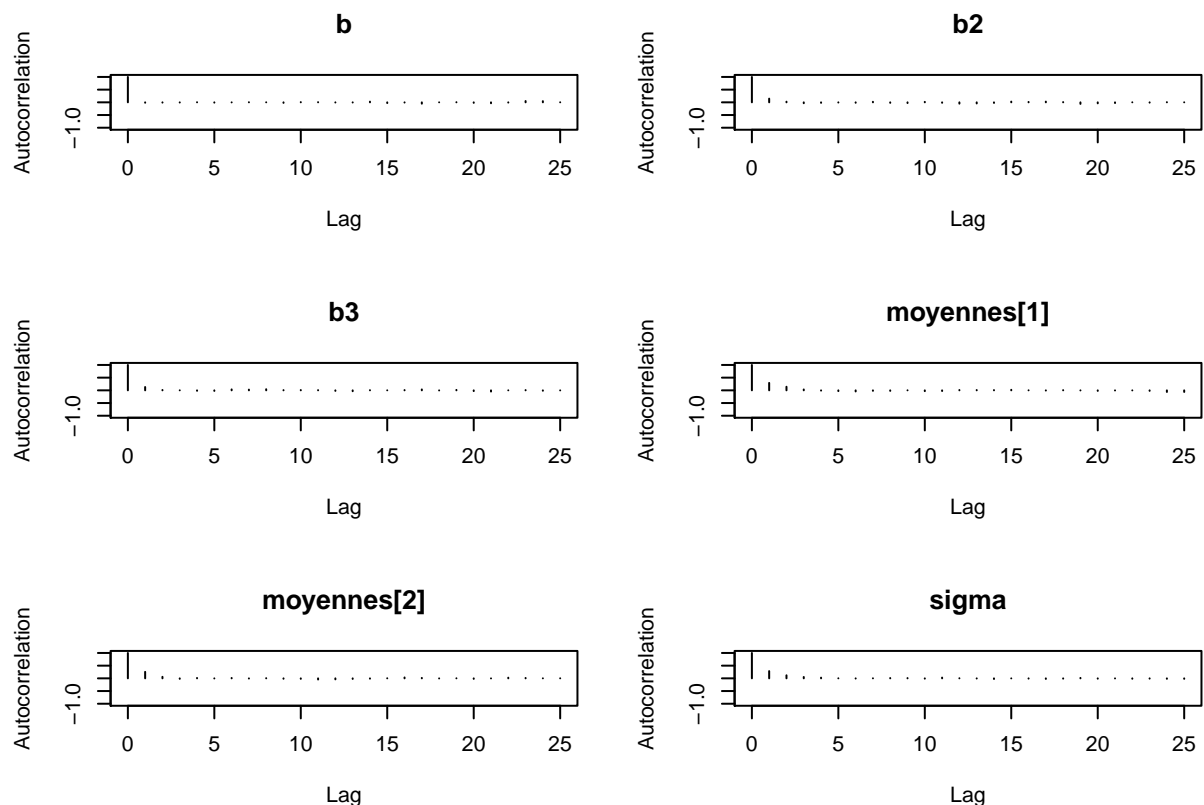
```
gelman.diag(mcmc1)
```

```
## Potential scale reduction factors:
##
##           Point est. Upper C.I.
## b                1          1
## b2               1          1
## b3               1          1
## moyennes[1]      1          1
## moyennes[2]      1          1
## sigma            1          1
##
## Multivariate psrf
##
## 1
```

```
gelman.plot(mcmc1)
```



```
autocorr.plot(mcmc1[[1]])
```



resultats

```
summary(mcmc1)
```

```
##
## Iterations = 4001:6000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean          SD Naive SE Time-series SE
## b             150.10317    9.93044 0.128201    0.1281922
## b2            -0.01401    0.01131 0.000146    0.0001695
## b3             40.10307    0.15655 0.002021    0.0022449
## moyennes[1]  3743.43314   93.47129 1.206709    1.5988232
## moyennes[2]  3350.74366  113.55212 1.465952    1.9022186
## sigma        495.17740   52.35218 0.675864    1.0259530
##
## 2. Quantiles for each variable:
##
##              2.5%        25%        50%        75%       97.5%
## b             130.62362   143.44033   150.11689   1.569e+02  1.693e+02
## b2            -0.03629   -0.02176   -0.01383   -6.115e-03  7.852e-03
## b3             39.80591    39.99855    40.10118    4.021e+01  4.041e+01
## moyennes[1]  3557.71174  3683.24001  3743.46462   3.805e+03  3.926e+03
## moyennes[2]  3126.30492  3276.33799  3350.29172   3.427e+03  3.577e+03
```

```
## sigma          406.28484  458.20451  491.07008  5.272e+02  6.097e+02
```

On peut comparer les modèle à l'aide du critère DIC. Le critère DIC est un critère de déviance ( $-2 \loglik + \text{cste}$ ) pénalisée par la complexité du modèle. Le critère DIC doit être le plus petit possible.

```
dic.samples(m1,n.iter=1000)
```

```
## Mean deviance:  736.9
## penalty 3.059
## Penalized deviance: 740
```

```
dic.samples(m2,n.iter=1000)
```

```
## Mean deviance:  887
## penalty 5.225
## Penalized deviance: 892.3
```

## Modèle avec l'âge de gestation et le sexe :

```
dat <- list(poids = poids, sexe = sexe, nbsemaines = nbsemaines, N = length(poids))
inits <- list(
  list(moyennes = c(2600, 4000), sigma = 500, b=150),
  list(moyennes = c(4500, 2700), sigma = 700, b=100),
  list(moyennes = c(4000, 4000), sigma = 300, b=200))
m3 <- jags.model('Rcode/modelepoidsnaissance3.txt', data = dat, inits = inits, n.chains = 3, quiet=TRUE)
update(m3, 3000,progress.bar="none")
mcmc1 <- coda.samples(m3, variable.names = c("moyennes", "sigma","b"), n.iter = 2000,progress.bar="none")
dic.samples(m3,n.iter=1000)
```

```
## Mean deviance:  731.6
## penalty 4.213
## Penalized deviance: 735.8
```

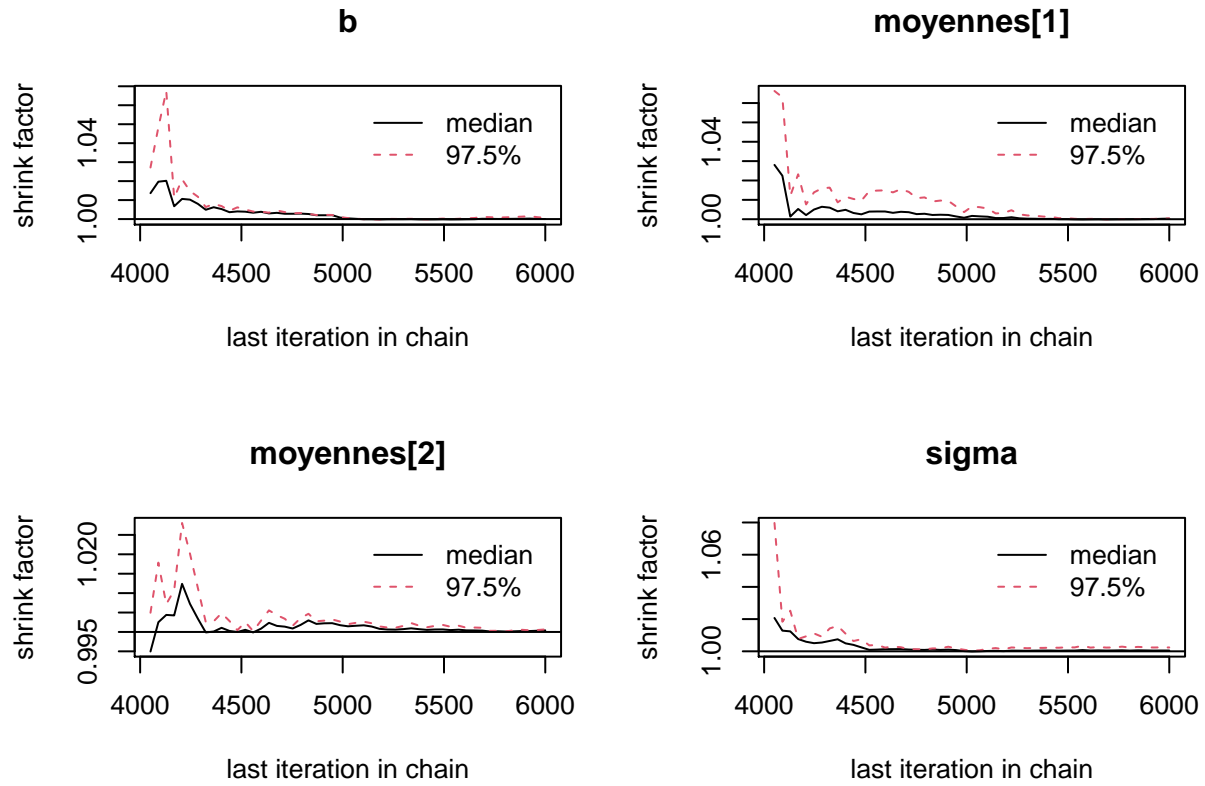
Le critère DIC est meilleur.

On vérifie la convergence

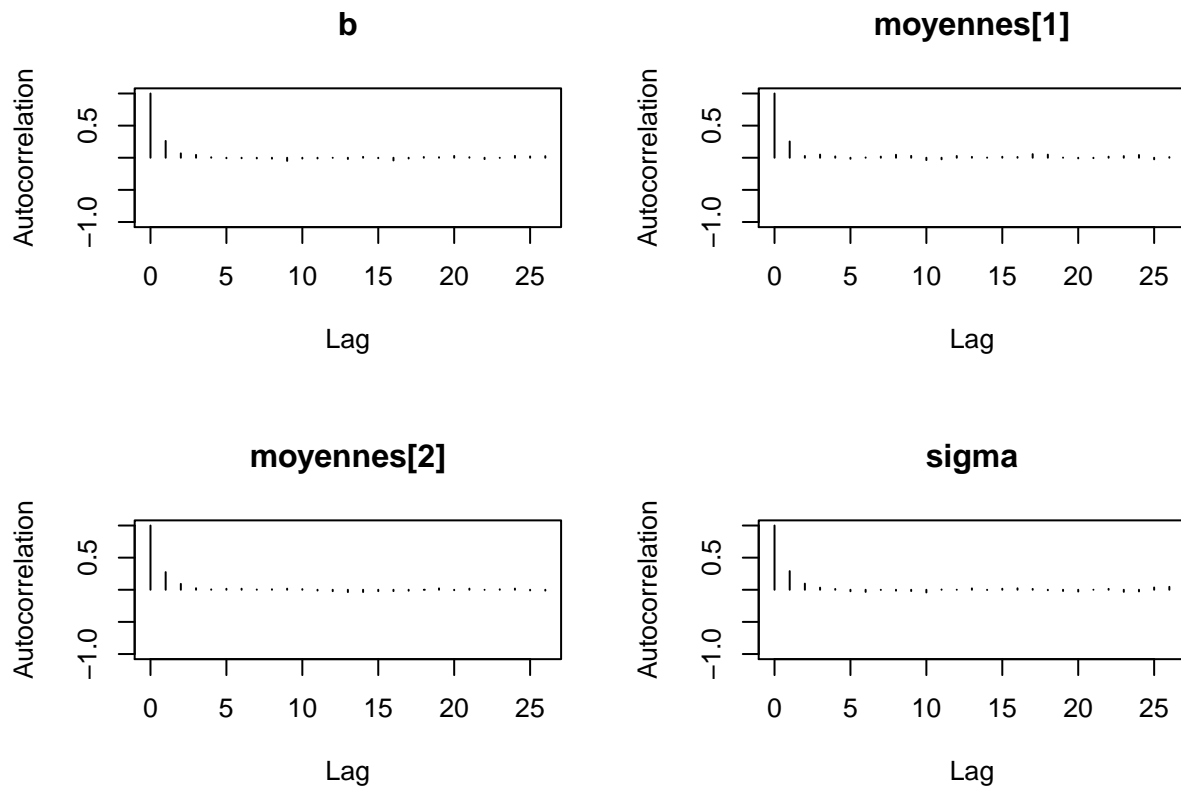
```
gelman.diag(mcmc1)
```

```
## Potential scale reduction factors:
##
##           Point est. Upper C.I.
## b           1           1
## moyennes[1] 1           1
## moyennes[2] 1           1
## sigma       1           1
##
## Multivariate psrf
##
## 1
```

```
gelman.plot(mcmc1)
```



```
autocorr.plot(mcmc1[[1]])
```



On compare le critère DIC :

```
dic.samples(m3,n.iter=1000)
```

```
## Mean deviance: 731.5
## penalty 4.062
## Penalized deviance: 735.5
```

Examinons les estimations obtenues

```
summary(mcmc1)
```

```
##
## Iterations = 4001:6000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean      SD Naive SE Time-series SE
## b              154.4  59.59  0.7693          0.9781
## moyennes[1]  3744.4  93.43  1.2062          1.5358
## moyennes[2]  3353.9 115.02  1.4849          1.9312
## sigma         500.5  54.38  0.7020          1.0214
##
## 2. Quantiles for each variable:
##
##              2.5%    25%    50%    75%   97.5%
## b              41.01  113.3  154.4  193.6  275.2
## moyennes[1]  3558.19 3682.7 3744.8 3807.9 3926.2
## moyennes[2]  3128.02 3277.2 3351.6 3430.1 3582.3
## sigma         406.45  462.0  495.4  534.8  620.1
```

Modèle avec l'âge de gestation, le sexe et la prise de poids de la mère :

```
prisepoids=data$POIDFING-data$POIDAVG
dat <- list(poids = poids, sexe = sexe, nbsemaines = nbsemaines, N = length(poids), prisepoids=prisepoids)
inits <- list(
  list(moyennes = c(2600, 4000), sigma = 500, b=150,c=-100),
  list(moyennes = c(4500, 2700), sigma = 700, b=100,c=0),
  list(moyennes = c(4000, 4000), sigma = 300, b=200,c=100))
m5 <- jags.model('Rcode/modelepoidsnaissance5.txt', data = dat, inits = inits, n.chains = 3, quiet=TRUE)
update(m5, 3000,progress.bar="none")
mcmc1 <- coda.samples(m5, variable.names = c("moyennes", "sigma","b","c"), n.iter = 2000,progress.bar="none")
```

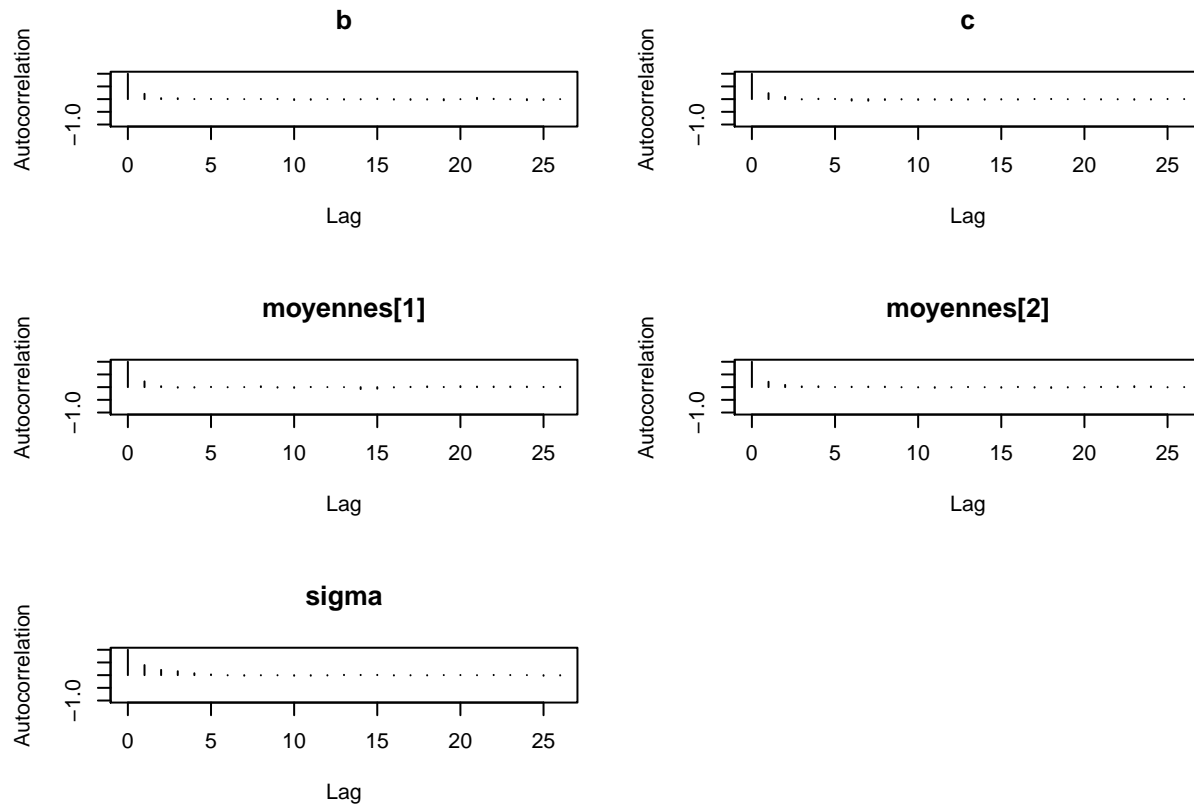
On vérifie la convergence

```
gelman.diag(mcmc1)
```

```
## Potential scale reduction factors:
##
##              Point est. Upper C.I.
## b              1          1.01
## c              1          1.00
## moyennes[1]    1          1.00
```

```
## moyennes[2]      1      1.00
## sigma            1      1.01
##
## Multivariate psrf
##
## 1
```

```
autocorr.plot(mcmc1[[1]])
```



On compare le critère DIC :

```
dic.samples(m5,n.iter=1000)
```

```
## Mean deviance: 728.1
## penalty 5.221
## Penalized deviance: 733.3
```

Examinons les estimations obtenues

```
summary(mcmc1)
```

```
##
## Iterations = 4001:6000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##              Mean      SD Naive SE Time-series SE
```

```
## b          152.07  56.97  0.7355      0.9323
## c           27.55  13.76  0.1776      0.2203
## moyennes[1] 3732.16  89.44  1.1547      1.5554
## moyennes[2] 3372.57 112.57  1.4533      1.9101
## sigma       483.87  53.92  0.6962      1.0775
##
## 2. Quantiles for each variable:
##
##           2.5%    25%    50%    75%    97.5%
## b          42.0373 112.72 151.19 190.75 266.03
## c          -0.2022  18.61  27.63  36.69  54.46
## moyennes[1] 3557.7066 3673.08 3730.98 3791.84 3910.22
## moyennes[2] 3149.5448 3299.77 3371.69 3445.75 3592.71
## sigma      392.7700  446.47  478.71  516.47  600.79
```

Modèle avec le sexe et un impact du fait d'être fumeur sur le poids moyen :

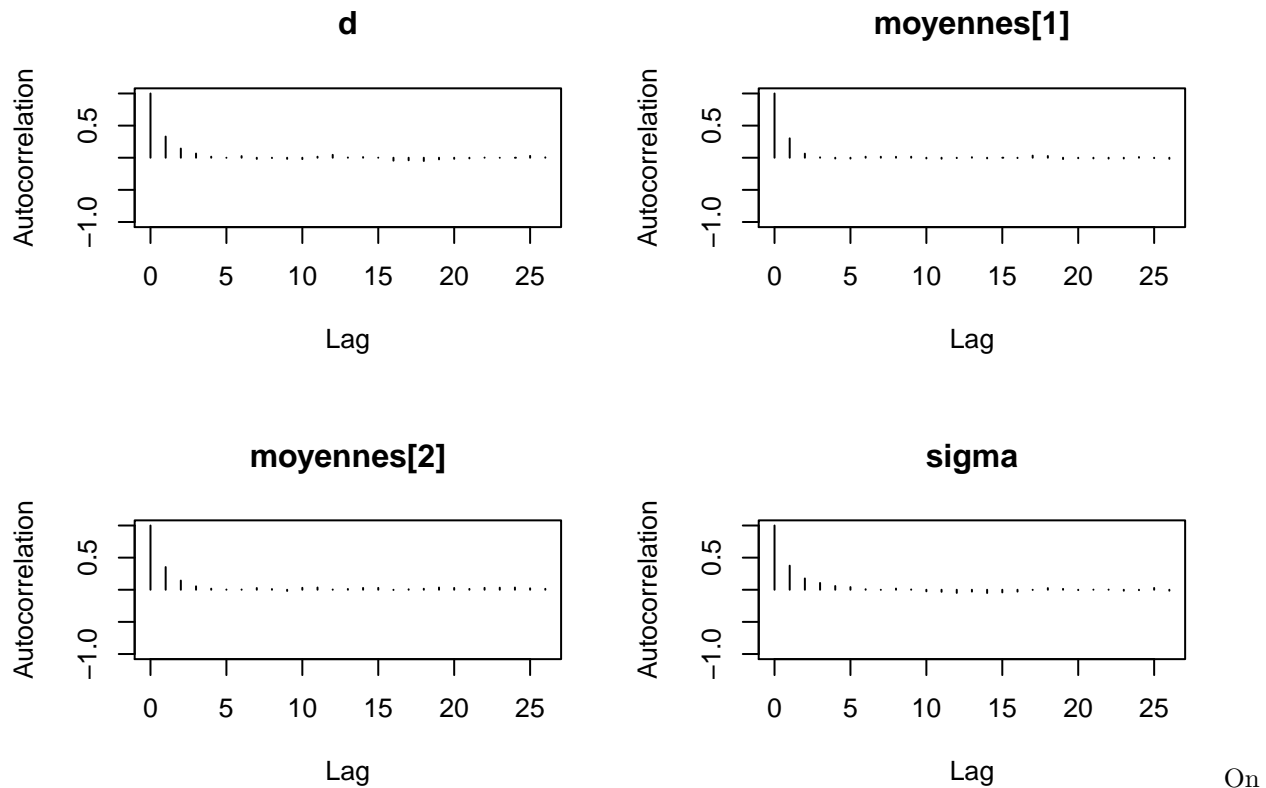
```
fumeur=data$CIGJOUR>0
dat <- list(poids = poids, sexe = sexe, fumeur = fumeur, N = length(poids))
inits <- list(
  list(moyennes = c(2600, 4000), sigma = 500, d=100),
  list(moyennes = c(4500, 2700), sigma = 700, d=0),
  list(moyennes = c(4000, 4000), sigma = 300, d=-100))
m6 <- jags.model('Rcode/modelepoidsnaissance6.txt', data = dat, inits = inits, n.chains = 3, quiet=TRUE)
update(m6, 3000, progress.bar="none")
mcmc1 <- coda.samples(m6, variable.names = c("moyennes", "sigma", "d"), n.iter = 2000, progress.bar="none")
```

On vérifie la convergence

```
gelman.diag(mcmc1)
```

```
## Potential scale reduction factors:
##
##           Point est. Upper C.I.
## d              1          1.00
## moyennes[1]    1          1.00
## moyennes[2]    1          1.00
## sigma          1          1.01
##
## Multivariate psrf
##
## 1
```

```
autocorr.plot(mcmc1[[1]])
```



compare le critère DIC :

```
dic.samples(m6,n.iter=1000)
```

```
## Mean deviance: 737.7
## penalty 4.122
## Penalized deviance: 741.8
```

Examinons les estimations obtenues

```
summary(mcmc1)
```

```
##
## Iterations = 4001:6000
## Thinning interval = 1
## Number of chains = 3
## Sample size per chain = 2000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##           Mean      SD Naive SE Time-series SE
## d          -97.72 182.2   2.3527          3.392
## moyennes[1] 3745.45 106.7   1.3775          1.889
## moyennes[2] 3400.28 130.5   1.6847          2.316
## sigma       534.27  59.4   0.7669          1.123
##
## 2. Quantiles for each variable:
##
##           2.5%    25%    50%    75%   97.5%
## d          -435.8 -229.1 -105.3  26.47 280.2
```



```
## moyennes[1] 3532.3 3676.1 3748.6 3818.64 3945.5
## moyennes[2] 3146.6 3311.5 3399.6 3484.53 3658.1
## sigma      434.2  492.4  528.4  570.05  668.6
```