Rapport - Statistique bayésienne

Philippe Real

15 mars, 2020

Contents

1 Introduction : Lecture des données - description statistique

On s'intéresse dans cette étude aux mutations des enseignants de collége et lycée de l'académie de Versaille. La variable réponse ou la variable à expliquer est la variable : *Barre*. Qui correspond au barême ou nombre de points nécessaire pour pouvoir obtenir un poste dans un établissement scolaire. Les co-variables sont composées des caractéristiques de l'établissement basées sur les effectifs de 2nd, 1ere et Terminale ainsi que les taux d'accès en 2nd, 1ere, Terminale et de réussites aux examens.

- Renommage des colonnes

On peut vouloir obttenir parfois une notation plus compacte. On utilisera alors le nommage suivant:

N°	Nouveau Nom	Ancien Nom
1	Eff_Prs_l	effectif_presents_serie_l
2	Eff_Prs_es	effectif_presents_serie_es
3	Eff_Prs_s	effectif_presents_serie_s
4	Eff_2e	effectif_de_seconde
5	Eff_1e	effectif_de_premiere
6	Tx_Suc.brt_l	taux_brut_de_reussite_serie_l
7	Tx_Suc.brt_es	taux_brut_de_reussite_serie_es
8	Tx_Suc.brt_s	taux_brut_de_reussite_serie_s
9	Tx_Suc.att_l	taux_reussite_attendu_serie_l
10	Tx_Suc.att_es	taux_reussite_attendu_serie_es
11	Tx_Suc.att_s	taux_reussite_attendu_serie_s
12	Tx_Acc.brt_bac2e	taux_acces_brut_seconde_bac
13	Tx_Acc.att_bac2e	taux_acces_attendu_seconde_bac
14	Tx_Acc.brt_bac1e	taux_acces_brut_premiere_bac
15	Tx_Acc.att_bac1e	taux_acces_attendu_premiere_bac
16	Tx_Suc.brt_Tot	taux_brut_de_reussite_total_series
17	Tx_Suc.att_Tot	taux_reussite_attendu_total_series

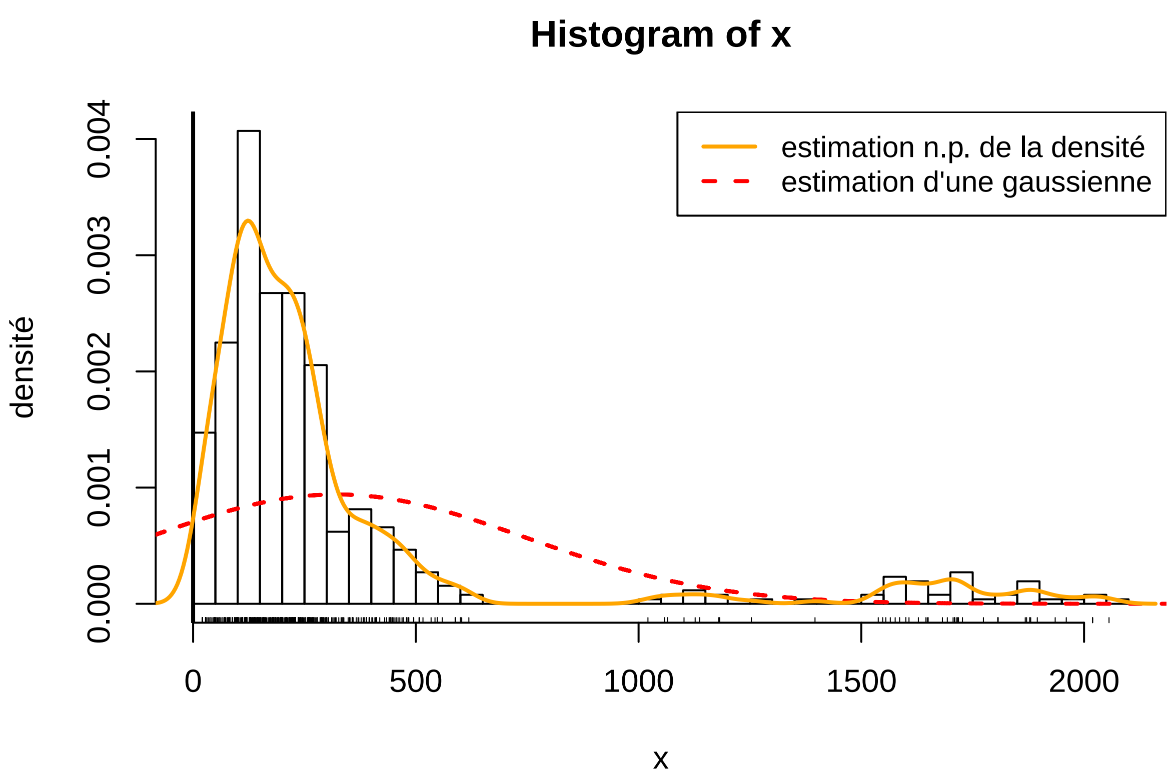
- Résumé des données :

On donne un résumé des variables explicatives utilisées par la suite, qui correspondent aux caractéristiques des établissements scolaire et de la variables explicatives et à la variable Barre qui est la variable à expliquer.

```
##      Barre          Eff_Prs_l        Eff_Prs_es       Eff_Prs_s
##  Min.   :  21.0   Min.   :  6.00   Min.   : 10.00   Min.   : 13.0
##  1st Qu.: 111.0   1st Qu.: 18.00   1st Qu.: 53.00   1st Qu.: 64.0
##  Median : 196.0   Median : 30.00   Median : 69.00   Median :100.0
##  Mean   : 321.9   Mean   : 34.24   Mean   : 74.42   Mean   :106.1
##  3rd Qu.: 292.0   3rd Qu.: 47.00   3rd Qu.: 99.00   3rd Qu.:140.0
##  Max.   :2056.0   Max.   :133.00   Max.   :192.00   Max.   :328.0
##   Tx_Suc.brt_l     Tx_Suc.brt_es    Tx_Suc.brt_s    Tx_Suc.att_l
##  Min.   : 36.00   Min.   : 51.0   Min.   :50.00   Min.   :65.00
##  1st Qu.: 82.00   1st Qu.: 81.0   1st Qu.:81.00   1st Qu.:84.00
##  Median : 89.00   Median : 88.0   Median :88.00   Median :89.00
##  Mean   : 86.35   Mean   : 86.4   Mean   :86.23   Mean   :86.91
##  3rd Qu.: 94.00   3rd Qu.: 94.0   3rd Qu.:93.00   3rd Qu.:92.00
##  Max.   :100.00   Max.   :100.0   Max.   :99.00   Max.   :98.00
##  Tx_Suc.att_es    Tx_Suc.att_s        Eff_2e           Eff_1e
##  Min.   :61.00   Min.   :61.00   Min.   : 36.0   Min.   : 36.0
##  1st Qu.:86.00   1st Qu.:86.00   1st Qu.:268.0   1st Qu.:226.5
##  Median :90.00   Median :89.00   Median :336.0   Median :289.0
```

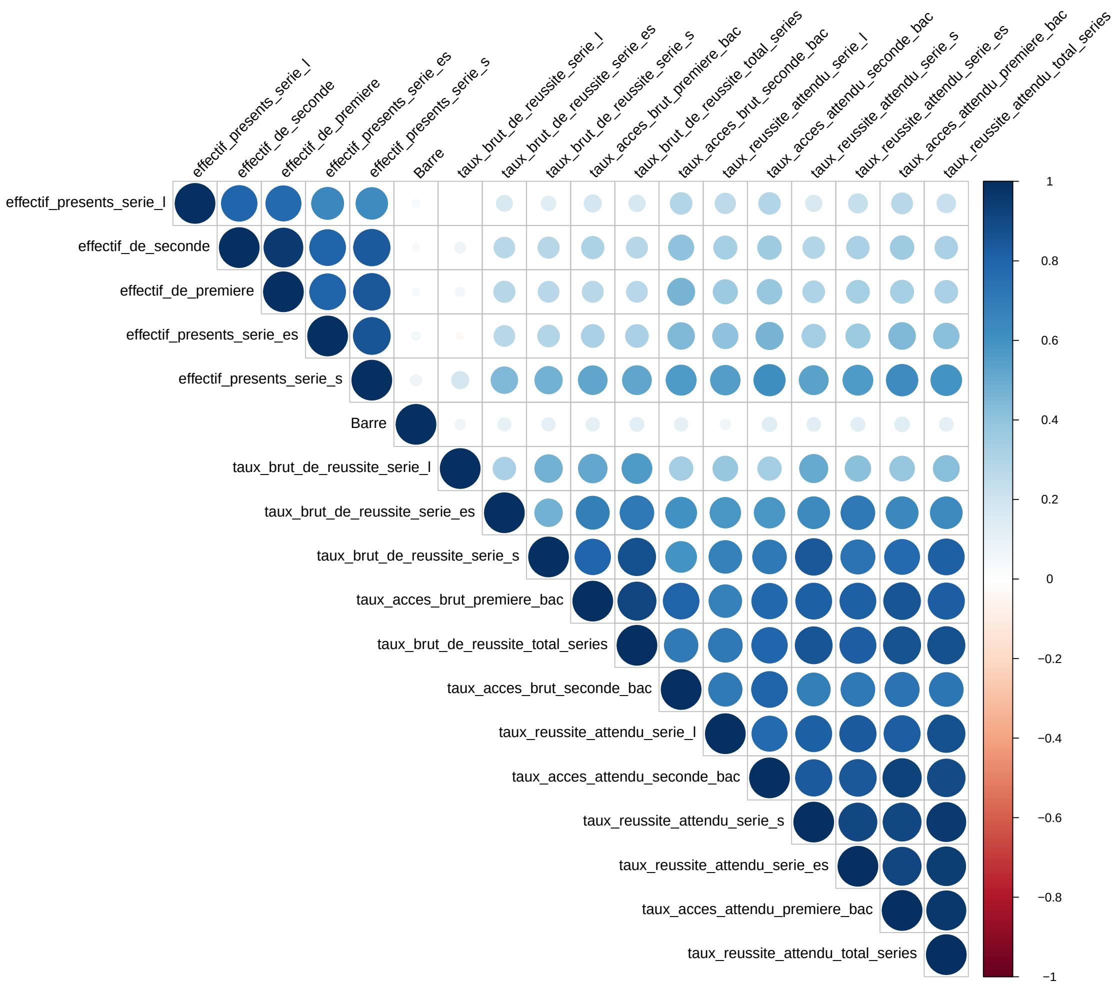
```
##  Mean   :87.97   Mean   :87.39   Mean   :351.6   Mean   :307.7
##  3rd Qu.:94.00   3rd Qu.:94.00   3rd Qu.:415.0   3rd Qu.:364.0
##  Max.   :98.00   Max.   :98.00   Max.   :764.0   Max.   :691.0
##  Tx_Acc.brt_bac2e Tx_Acc.att_bac2e Tx_Acc.brt_bac1e Tx_Acc.att_bac1e
##  Min.   :49.00    Min.   :50.00    Min.   :65.00    Min.   :70.00
##  1st Qu.:64.00    1st Qu.:64.00    1st Qu.:82.00    1st Qu.:81.00
##  Median :71.00    Median :69.00    Median :85.00    Median :85.00
##  Mean   :69.61    Mean   :68.47    Mean   :84.53    Mean   :84.19
##  3rd Qu.:76.00    3rd Qu.:73.00    3rd Qu.:89.25    3rd Qu.:89.00
##  Max.   :87.00    Max.   :83.00    Max.   :97.00    Max.   :94.00
##  Tx_Suc.brt_Tot  Tx_Suc.att_Tot
##  Min.   :64.00   Min.   :67.0
##  1st Qu.:82.00   1st Qu.:84.0
##  Median :86.00   Median :88.0
##  Mean   :85.46   Mean   :86.8
##  3rd Qu.:91.00   3rd Qu.:92.0
##  Max.   :98.00   Max.   :98.0
```

- Histogramme de la variable à expliquer *Barre*



L'allure de la densité représentée par l'histogramme est très assymétrique et comporte une queue qui pourrait être épaisse. En tout cas à ne pas négliger. L'estimation de cette densité par une loi de Pareto proposée en partie II semble justifié.

- Corrélations 2 à 2 entre les variables



La variable a expliquer *Barre* est assez peu corrélée avec les variables constituant les caractéristiques de l'établissement. On remarque aussi deux groupes de variables dinstintcs, avec des corrélations inter-groupe faibles et intra groupes fortes.

- Les variables de type effectifs (Effectifs et Effectifs présents, 5 variables en tout).
- Les variables de *taux* (Taux de réussite et taux Attendu, 12 variables en tout).

Par contre au sein de chacun des groupes, comme on peut s'y attendre les corrélations entre variables (intra-groupe) sont fortes.

On pourrait imaginer de ne conserver que les variables de type effectifs dans le premier groupe. Et de la même manière dans le second groupe, ne garder que les taux de réussite brutes, vaiables qui semblent redondantes avec celles de type taux attendus.

On remarque que le *taux_brute_de_resussite_seriel_l* (Tx_Suc.brt_l) est moins corrélée aux autres variables, et semble avoir une certaine indépendance.

Les variables de type total, réussite et attendu, n'apportent pas vraiment d'information et pourraient être écartées elles aussi si l'on cherchait à réduire le nombre de variables.

2 Partie I - Régression linéaire Bayésienne

On cherche à expliquer le nombre de points nécessaire à une mutation (colonne Barre) par les caractéristiques du lycée. On considère un modéle de régression linéaire gaussien, que l'on rappelle ici.

2.1 Rappels définitions et notations

2.1.1 Modèle linéaire Gaussien

Le modèle linéaire, tente d'expliquer les observations (y_i) (input ici la variable *Barre*) par des covariables $(x_1, ..., x_p)$ (les caractéristues de l'établissement scolaire) à partir du modèle suivant :

$y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} + \epsilon_i$ où $\epsilon_i \sim N(0, \sigma^2)$ et iid.

On note $y = (y_1, .., y_n)$ le vecteur des observations et $X = (x_{ik})_{1 \leq i \leq n, 1 \leq k \leq p}$ la matrice des covaraiables ou de design (predictor).

En notation matricielles le modèle se réécrit de la manière suivante:

$$y \mid \alpha, \beta, \sigma^2 \sim N_n(\alpha 1_n + X\beta, \sigma^2 I_n)$$

où N_n est la distribution de la loi normale en dimension n.

Ainsi les y_i suivent des lois normales indépendantes avec :

$$E(y_i \mid \alpha, \beta, \sigma^2) = \alpha + \sum_{j=1}^{p} \beta_j x_{ij}$$

$$V(y_i \mid \alpha, \beta, \sigma^2) = \sigma^2$$

2.1.2 Contexte bayésien

On rappelle ici la formulation de la régression linaire dans le contexte bayésien.

On se place dans le cadre d'une expérience statistique paramétrique, où le vecteur des observations $Y = (y_1, ..., y_n)$ est iid et les $y_i \sim P_\theta$ une loi de paramètre θ.

Dans le contexte bayésien, on suppose que le paramètre inconnu θ est une v.a dont la loi de probabilité représente notre incertitude sur les valeurs possibles.

- Loi à priori $\pi(\theta)$

Cette loi du paramètre θ est la loi à priori, notée: $\pi(\theta)$. Elle représente "l'appriori" ou la croyance du statisticien avant le début de l'expérience. Sont choix est important, et on doit la choisir de manière à obtenir: une loi conjuguée pour faciliter les calculs, ou bien non informative (à priori de Jeffreys), fournit par un expert…

- Loi à postériori $\pi(\theta, y)$

On appelle la loi à postériori de θ sachant $y_1, y_2, ..., y_n$ la loi de distribution $\pi(\theta \mid Y) \propto \pi(\theta) L(\theta \mid Y)$

Cette définition découle de la formule de Bayes: $\pi(\theta \mid y) = \frac{\pi(\theta) f_{Y|\theta}(y|\theta)}{f_Y(y)}$

On retrouve l'équivalence des écritures avec $f_{Y|\theta}(y \mid \theta) = L(\theta \mid Y)$ Et $f_Y(y)$ ne dépend pas du paramètre θ, c'est une constante de normalisation qui est unique et que l'on peut retrouver une fois la loi à postériori déterminer analytiquement, qui doit s'intégrer à 1.

2.1.3 Régression linaire Bayésienne - Inférence bayésienne à l'aide de la loi a priori g de Zellner

On reprend les hypothèses et le contexte de définition du modèle linéaire gaussien, que l'on réinterprète avec l'approche Bayésienne. On considère la loi à priori $\pi(\theta)$ définit à partir des deux lois suivantes :

$$\beta \mid \sigma^2, X \sim N_{k+1}(\tilde{\beta}, \sigma^2 M^{-1})$$
$$\sigma^2 \mid X \sim IG(a, b)$$

L'idée principale de la modélisation de la G-prior de Zellner est de permettre d'introduire des informations (éventuellement faibles) sur le paramétre de localisation de la régression (commandé par le paramètre g) et surtout de contourner les aspects les plus difficiles de la définition de la prior, à savoir la structure de la corrélation (p76 Marin et Robert - bayesian essential with R).

En fixant la matrice M de la manière suivante dans l'approche de Zellner, on obtient la g-prior ou loi informative de Zellner :

$$\beta \mid \sigma^2, X \sim N_{k+1}(\tilde{\beta}, g\sigma^2({}^tXX)^{-1})$$
$$\sigma^2 \sim \pi(\sigma^2 \mid X) \propto \sigma^{-2}$$

Il reste à choisir le paramètre g, souvent g=1 ou g=n en fonction du poids que l'on veut accorder à la prior. Si g=2 celà revient à donner à la prior le même poids que 50% de l'échantilon. Avec g=n on donne à la loi à priori le même poids que 1-observation.

Pour l'espérance à priori $\tilde{\beta}$ ou pourra la prendre = 0 si l'on n'a pas d'information à priori.

La loi à priori $\pi(\theta)$ se déduit simplement à partir des deux lois précédentes:

$$\pi(\theta) = \pi(\beta, \sigma^2 \mid X) = \pi(\beta \mid \sigma^2, X)\pi(\sigma^2 \mid X)$$

Cette loi à la propriété remarquable d'être une loi conjugué et sa loi à postériori associée a l'expression analytique suivante:

$$\beta \mid \sigma^2, y, X \sim N_{k+1}(\frac{g}{g+1}\hat{\beta}, \frac{\sigma^2 g}{g+1}({}^tXX)^{-1})$$
$$\sigma^2 \mid y, X \sim IG(\frac{n}{2}\hat{\beta}, \frac{s^2}{2} + \frac{1}{2(g+1)}({}^t\hat{\beta}^t XX\hat{\beta})$$

donc :

$$\beta \mid y, X \sim Student_{k+1}(n, \frac{g}{g+1}\hat{\beta}, \frac{g(s^2 + ({}^t\hat{\beta}^t XX\hat{\beta})/(g+1))}{n(g+1)}({}^tXX)^{-1})$$

2.2 Résultats et interprétation des coefficients

Pour cette étude, on va s'appuyer sur les éléments du cours et les fonctions utilisées en TP et plus particulièrement du TP-N°4. On utilisera aussi des fonctions du package R *Bayess* ainsi que le livre associé: *"Bayesian essential with R" ou "Bayesian Core"* de Marin et Robert. Comme suggéré en page 69 de cet ouvrage, on va centrer et réduire les éléments de la matrice de design X. Comme dans l'exemple du livre de Marin et Robert, on prendra aussi le log de la variable *y=Barre* à expliquer pour opérer une forme de linéarisation ou atténuation des écarts. Dans ce qui suit on va confronter les résultats obtenus à partir des fonctions pour l'essentiel vu ou adaptées du cours et des fonctions du package Bayess, plus particulièrement les fonctions: *BayesReg* et *ModChoBayesReg*.

2.2.1 Calcul explicite des coefficients

On se place dans le contexte Bayésien avec pour loi à prioiri $\pi(\theta) = \pi(\beta, \sigma^2 \mid X)$ la G-prior de Zellner :

$$\beta \mid \sigma^2, X \sim N_{k+1}(\tilde{\beta}, g\sigma^2({}^tXX)^{-1})$$
$$\sigma^2 \sim \pi(\sigma^2 \mid X) \propto \sigma^{-2}$$

On cherche à calculer la moyenne à priori, à partir de la formule suivante:

$$E^\pi(\beta \mid y) = \frac{g}{g+1}\left(\hat{\beta} + \tilde{\beta}/g\right)$$

Où $\hat{\beta}$ est le vecteur des coefficients du modèle linéaire classique obtenu par maximum de vraissemblance ou moindre carré ordinaire.

On peut justifier cette expression de la manière suivante, comme par définition de la prior on a :

$$E^\pi(\beta \mid \sigma^2, y) = \frac{g}{g+1}\left(\hat{\beta} + \tilde{\beta}/g\right)$$

Puis en prenant l'espérance et en conditionnant par rapport à y, on obtient :

$$E^\pi(E^\pi(\beta \mid \sigma^2, y) \mid y) = E^\pi\left(\frac{g}{g+1}\left(\hat{\beta} + \tilde{\beta}/g\right)\right) = \frac{g}{g+1}\left(\hat{\beta} + \tilde{\beta}/g\right)$$

Et comme par définition β ne dépend pas de σ on a :

$$E^\pi(E^\pi(\beta \mid \sigma^2, y) \mid y) = E^\pi(\beta \mid y)$$

On va maintenat calculer explicitement la quantité : $E^\pi(\beta \mid y)$

- calcul de $\hat{\beta}$ coefficient du modèle linéaire

On sait que $\hat{\beta}$ s'obtient comme solution du problème : $\hat{\beta} = (X^TX)^{-1}X^Ty$

```
beta0.lm=mean(y)
beta.lm=solve(t(X)%*%X,t(X)%*%y)
betahat=beta.lm
betahat
```

```
##                          [,1]
## Eff_Prs_l          0.053332691
## Eff_Prs_es        -0.026963434
## Eff_Prs_s         -0.020843975
## Tx_Suc.brt_l       0.007732727
## Tx_Suc.brt_es      0.100143233
## Tx_Suc.brt_s       0.170613928
## Tx_Suc.att_l      -0.150586117
## Tx_Suc.att_es      0.013747137
## Tx_Suc.att_s      -0.144943119
## Eff_2e             0.086840268
## Eff_1e            -0.118486206
## Tx_Acc.brt_bac2e   0.161617462
## Tx_Acc.att_bac2e  -0.228735899
## Tx_Acc.brt_bac1e  -0.311734958
## Tx_Acc.att_bac1e   0.591735214
## Tx_Suc.brt_Tot    -0.031198324
## Tx_Suc.att_Tot     0.109418375
```

On peut aussi retrouver les coefficients $\hat{\beta}$ à partir de la fonction lm. On obtient quasiment les mêmes résultats:

```r
reg.lm=lm(y~X)
summary(reg.lm)
```

```
##
## Call:
## lm(formula = y ~ X)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.44317 -0.50721 -0.05068  0.39502  2.67793
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        5.265726   0.041149 127.966   <2e-16 ***
## XEff_Prs_l         0.053333   0.076316   0.699   0.4850
## XEff_Prs_es       -0.026963   0.093958  -0.287   0.7743
## XEff_Prs_s        -0.020844   0.130785  -0.159   0.8734
## XTx_Suc.brt_l      0.007733   0.065536   0.118   0.9061
## XTx_Suc.brt_es     0.100143   0.091782   1.091   0.2758
## XTx_Suc.brt_s      0.170614   0.128597   1.327   0.1852
## XTx_Suc.att_l     -0.150586   0.113031  -1.332   0.1834
## XTx_Suc.att_es     0.013747   0.155069   0.089   0.9294
## XTx_Suc.att_s     -0.144943   0.199290  -0.727   0.4674
## XEff_2e            0.086840   0.186957   0.464   0.6425
## XEff_1e           -0.118486   0.201057  -0.589   0.5559
## XTx_Acc.brt_bac2e  0.161617   0.113712   1.421   0.1559
## XTx_Acc.att_bac2e -0.228736   0.144497  -1.583   0.1141
## XTx_Acc.brt_bac1e -0.311735   0.163039  -1.912   0.0564 .
## XTx_Acc.att_bac1e  0.591735   0.253937   2.330   0.0202 *
## XTx_Suc.brt_Tot   -0.031198   0.210838  -0.148   0.8824
## XTx_Suc.att_Tot    0.109418   0.376114   0.291   0.7712
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9347 on 498 degrees of freedom
## Multiple R-squared:  0.08332,    Adjusted R-squared:  0.05202
## F-statistic: 2.663 on 17 and 498 DF,  p-value: 0.0003462
```

On a "éliminé" l'intercept en centrant ou sinon avec on aurait dû utiliser la formule: y~X-1 pour pouvoir comparer.

- Calcul de $E^{\pi}(\beta \mid y, X) = \frac{g}{g+1}(\hat{\beta} + \frac{\tilde{\beta}}{g})$ G-prior informative de Zellner

Avec comme Hypothèses Zellner G-prior: g=1 et $\tilde{\beta} = 0$

```r
g=1
betatilde=rep(0,dim(X)[2])

mbetabayes=g/(g+1)*(beta.lm+betatilde/g)
postmean=rbind(Intercept=beta0.lm,mbetabayes)
postmean
```

```
##                          [,1]
## Intercept         5.265726370
## Eff_Prs_l         0.026666346
```

```
## Eff_Prs_es       -0.013481717
## Eff_Prs_s        -0.010421988
## Tx_Suc.brt_l      0.003866363
## Tx_Suc.brt_es     0.050071616
## Tx_Suc.brt_s      0.085306964
## Tx_Suc.att_l     -0.075293059
## Tx_Suc.att_es     0.006873569
## Tx_Suc.att_s     -0.072471559
## Eff_2e            0.043420134
## Eff_1e           -0.059243103
## Tx_Acc.brt_bac2e  0.080808731
## Tx_Acc.att_bac2e -0.114367949
## Tx_Acc.brt_bac1e -0.155867479
## Tx_Acc.att_bac1e  0.295867607
## Tx_Suc.brt_Tot   -0.015599162
## Tx_Suc.att_Tot    0.054709188
```

Avec comme Hypothèses Zellner G-prior: g=n et $\tilde{\beta} = 0$ On accorde ici moins d'importance à la prior et on se retrouve plus proche des coefficients obtenus à partir d'une régression classique.

```r
g=length(y)
betatilde=rep(0,dim(X)[2])

mbetabayes=g/(g+1)*(beta.lm+betatilde/g)
postmean=rbind(Intercept=beta0.lm,mbetabayes)
postmean
```

```
##                         [,1]
## Intercept         5.26572637
## Eff_Prs_l         0.05322953
## Eff_Prs_es       -0.02691128
## Eff_Prs_s        -0.02080366
## Tx_Suc.brt_l      0.00771777
## Tx_Suc.brt_es     0.09994953
## Tx_Suc.brt_s      0.17028392
## Tx_Suc.att_l     -0.15029485
## Tx_Suc.att_es     0.01372055
## Tx_Suc.att_s     -0.14466276
## Eff_2e            0.08667230
## Eff_1e           -0.11825703
## Tx_Acc.brt_bac2e  0.16130486
## Tx_Acc.att_bac2e -0.22829347
## Tx_Acc.brt_bac1e -0.31113199
## Tx_Acc.att_bac1e  0.59059066
## Tx_Suc.brt_Tot   -0.03113798
## Tx_Suc.att_Tot    0.10920673
```

C'est cette dernière hypothèse que l'on va conserver.

2.3 Choix des covariables et comparaison au résultat obtenu par une analyse fréquentiste.

Pour choisir les covariables significatives, on peut se baser sur les facteurs de Bayes. Ils donnent une idée de l'importance d'une variable. En effet on peut tester l'hypothèse H_0 ={Modèle sans la variable i} contre {Modèle complet}. Ceci pour chacune des variables.

2.3.1 Choix des covariables avec les Bayes factors

Pour comparer les modèles on peut utiliser les facteurs de Bayes: Test d'hypothèse $H_0 : \beta_i = 0$
On test l'hypothèse H_0, $\forall i = 1, ..., 17$ et on calcul le Bayes Factor. C'est ce que propose la fonction *BayesReg* du package Bayess. Ce qui donne une indication de la pertinance de la variable, un peu à la manière de la fonction lm.

On va calculer tout d'abord les Bayes Factor à partir de la formule et de la fonction vue en du cours qui est reprise dans la fonction: $CalcBayesFactor$.

- A partir de la fonction $CalcBayesFactor$:

Avec $g = n$ on obtient :

```
##          colnames(X) bfactor
## 1          Eff_Prs_l -1.3535
## 2         Eff_Prs_es -1.3562
## 3          Eff_Prs_s -1.3566
## 4       Tx_Suc.brt_l -1.3567
## 5      Tx_Suc.brt_es -1.3489
## 6       Tx_Suc.brt_s -1.3451
## 7       Tx_Suc.att_l -1.3450
## 8      Tx_Suc.att_es -1.3567
## 9       Tx_Suc.att_s -1.3532
## 10            Eff_2e -1.3553
## 11            Eff_1e -1.3544
## 12 Tx_Acc.brt_bac2e -1.3434
## 13 Tx_Acc.att_bac2e -1.3401
## 14 Tx_Acc.brt_bac1e -1.3325
## 15 Tx_Acc.att_bac1e -1.3208
## 16   Tx_Suc.brt_Tot -1.3566
## 17   Tx_Suc.att_Tot -1.3562
```

Avec $g = 1$ on obtient :

```
##          colnames(X) bfactor
## 1          Eff_Prs_l -0.1489
## 2         Eff_Prs_es -0.1502
## 3          Eff_Prs_s -0.1504
## 4       Tx_Suc.brt_l -0.1505
## 5      Tx_Suc.brt_es -0.1466
## 6       Tx_Suc.brt_s -0.1447
## 7       Tx_Suc.att_l -0.1446
## 8      Tx_Suc.att_es -0.1505
## 9       Tx_Suc.att_s -0.1488
## 10            Eff_2e -0.1498
## 11            Eff_1e -0.1494
## 12 Tx_Acc.brt_bac2e -0.1438
## 13 Tx_Acc.att_bac2e -0.1422
## 14 Tx_Acc.brt_bac1e -0.1384
## 15 Tx_Acc.att_bac1e -0.1325
## 16   Tx_Suc.brt_Tot -0.1504
## 17   Tx_Suc.att_Tot -0.1502
```

En donnant plus de poids à la prior certains coefficients commencent à être significatifs au sens de $Jeffrey$: 7, 12 14 et 15éme variable.

- Bayes Regression fonction *BayesReg* :

Pour estimer les β à postériori, on va utiliser la fonction (modifiée) *BayesReg* du package *Bayess* issue du livre de *Marin et Robert : Bayesian Essentials with R.* Le calcul détaillé a été exposé au § précédent. Comme on l'a vu ce calcul peut aussi être obtenu directement à partir de la fonction lm (residuals). On comparera le résultat obtenu avec le résultat précédent renvoyé par la fonction reprise

Avec $g = n$ on obtient :

```
##
##             PostMean PostStError Log10bf EvidAgaH0
## Intercept    5.2657      0.0405
## x1           0.0532      0.0751 -1.2474
## x2          -0.0269      0.0925 -1.3383
## x3          -0.0208      0.1287 -1.3511
## x4           0.0077      0.0645 -1.3536
## x5           0.0999      0.0904 -1.0903
## x6           0.1703      0.1266  -0.963
## x7          -0.1503      0.1113 -0.9597
## x8           0.0137      0.1527  -1.355
## x9          -0.1447      0.1962 -1.2383
## x10          0.0867      0.1840 -1.3084
## x11         -0.1183      0.1979 -1.2789
## x12          0.1613      0.1119  -0.905
## x13         -0.2283      0.1422 -0.7966
## x14         -0.3111      0.1605 -0.5405
## x15          0.5906      0.2500 -0.1465
## x16         -0.0311      0.2076 -1.3518
## x17          0.1092      0.3702 -1.3378
##
##
## Posterior Mean of Sigma2: 0.8483
## Posterior StError of Sigma2: 1.2009
```

```
## $postmeancoeff
##  [1]  5.26572637  0.05322953 -0.02691128 -0.02080366  0.00771777
##  [6]  0.09994953  0.17028392 -0.15029485  0.01372055 -0.14466276
## [11]  0.08667230 -0.11825703  0.16130486 -0.22829347 -0.31113199
## [16]  0.59059066 -0.03113798  0.10920673
##
## $postsqrtcoeff
##                         Eff_Prs_l       Eff_Prs_es        Eff_Prs_s
##       0.04054683       0.07512600       0.09249263       0.12874552
##     Tx_Suc.brt_l    Tx_Suc.brt_es     Tx_Suc.brt_s     Tx_Suc.att_l
##       0.06451402       0.09035059       0.12659189       0.11126847
##    Tx_Suc.att_es     Tx_Suc.att_s           Eff_2e           Eff_1e
##       0.15265116       0.19618245       0.18404180       0.19792191
## Tx_Acc.brt_bac2e Tx_Acc.att_bac2e Tx_Acc.brt_bac1e Tx_Acc.att_bac1e
##       0.11193874       0.14224334       0.16049614       0.24997680
##   Tx_Suc.brt_Tot   Tx_Suc.att_Tot
##       0.20755035       0.37024882
##
## $log10bf
##  [1] -1.2473606 -1.3382924 -1.3510535 -1.3536256 -1.0902888 -0.9630004
##  [7] -0.9597210 -1.3549842 -1.2382756 -1.3084085 -1.2789489 -0.9049926
## [13] -0.7966277 -0.5405042 -0.1465210 -1.3518388 -1.3377818
```

```
##
## $postmeansigma2
## [1] 0.8483276
##
## $postvarsigma2
## [1] 1.442131
```

Les facteurs de bayes sont négatifs, et leur interprétation au sens de *Jeffrey* montre qu'ils ne sont pas significatifs.

Avec $g = 1$ on obtient :

```
##
##            PostMean PostStError Log10bf EvidAgaH0
## Intercept    5.2657      0.0415
## x1           0.0267      0.0544 -0.0981
## x2          -0.0135      0.0669 -0.1417
## x3          -0.0104      0.0932 -0.1478
## x4           0.0039      0.0467  -0.149
## x5           0.0501      0.0654 -0.0227
## x6           0.0853      0.0916  0.0384       (*)
## x7          -0.0753      0.0805  0.0399       (*)
## x8           0.0069      0.1105 -0.1497
## x9          -0.0725      0.1420 -0.0937
## x10          0.0434      0.1332 -0.1273
## x11         -0.0592      0.1432 -0.1132
## x12          0.0808      0.0810  0.0662       (*)
## x13         -0.1144      0.1029  0.1183       (*)
## x14         -0.1559      0.1161  0.2414       (*)
## x15          0.2959      0.1809  0.4312       (*)
## x16         -0.0156      0.1502 -0.1482
## x17          0.0547      0.2679 -0.1414
##
##
## Posterior Mean of Sigma2: 0.8867
## Posterior StError of Sigma2: 1.2552

## $postmeancoeff
##  [1]  5.265726370  0.026666346 -0.013481717 -0.010421988  0.003866363
##  [6]  0.050071616  0.085306964 -0.075293059  0.006873569 -0.072471559
## [11]  0.043420134 -0.059243103  0.080808731 -0.114367949 -0.155867479
## [16]  0.295867607 -0.015599162  0.054709188
##
## $postsqrtcoeff
##                        Eff_Prs_l       Eff_Prs_es        Eff_Prs_s
##       0.04145427      0.05436358       0.06693063       0.09316438
##     Tx_Suc.brt_l    Tx_Suc.brt_es     Tx_Suc.brt_s     Tx_Suc.att_l
##       0.04668441      0.06538058       0.09160595       0.08051742
##    Tx_Suc.att_es     Tx_Suc.att_s           Eff_2e           Eff_1e
##       0.11046327      0.14196390       0.13317853       0.14322263
## Tx_Acc.brt_bac2e Tx_Acc.att_bac2e Tx_Acc.brt_bac1e Tx_Acc.att_bac1e
##       0.08100246      0.10293184       0.11614014       0.18089121
##   Tx_Suc.brt_Tot   Tx_Suc.att_Tot
##       0.15019008      0.26792390
##
## $log10bf
##  [1] -0.09807614 -0.14167061 -0.14778702 -0.14901979 -0.02272945
```

```
##  [6]  0.03837066  0.03994529 -0.14967094 -0.09371958 -0.12734569
## [11] -0.11322228  0.06622719  0.11828629  0.24143367  0.43115161
## [16] -0.14816342 -0.14142583
##
## $postmeansigma2
## [1] 0.8867237
##
## $postvarsigma2
## [1] 1.575629
```

En donnant plus d'importance à la prior, on voit que certaines variables se dégagent: les 6, 7, 12, 13, 14 et 15éme.

- Conclusion

On obtient des résultats comparable avec les deux implémentations des facteurs de Bayes (Fonction du cours: *CalcBayesFactor* et Bayess: *BayesReg*) Les 6ème (taux_brut_de_reussite_serie_l), 7ème (taux_brut_de_reussite_serie_es), 12ème (taux_acces_brut_seconde_bac), 13ème (taux_acces_attendu_seconde_bac), 14ème (taux_acces_brut_premiere_bac) et 15ème (taux_acces_attendu_premiere_bac) variables semblent être les plus significatives.

2.3.2 Choix de modèle : par calcul exact

On considère ici encore une implémentation de calcul exact vue en cours: *BayesModelChoice_Exact* que l'on rapproche de la fonction *ModChoBayesReg* du package Bayess.

- A partir de la méthode vue en cours qui est recodée ici dans la fonction : *BayesModelChoice_Exact*

```
##       model.name model.prob
## 16385        _15 0.36233676
## 65537        _17 0.04923733
## 20481     _13_15 0.03527423
## 24577     _14_15 0.02734963
## 16449      _7_15 0.02362729
## 18433     _12_15 0.02206402
## 16393      _4_15 0.02107504
## 81921     _15_17 0.01872976
## 16641      _9_15 0.01765499
## 16401      _5_15 0.01724574
```

Le modèle n'incluant que la variable N°15 *taux_acces_attendu_premiere_bac* a une proba beaucoup plus importante (facteur 10). Par ailleurs elle est aussi présente dans presque tous les modèles alternatifs. On remarque aussi la variable N°17: *taux_reussite_attendu_total_series*. La variable N°13: *taux_acces_attendu_seconde_bac* la variable N°14: *taux_acces_brut_premiere_bac*. On retrouve les variables qui se détachaient dans l'analyse des Bayes factor avec g=1.

- A partir de la fonction (modifée) - ModChoBayesReg du package Bayess

Remarque: la valeur de la PostProb a été transformée aussi et n'est pas une plus une proba. Par contre le classement à partir de cette valeur reste valable (cf. annexe). On a ajouté un paramètre *bCalcul*=TRUE par défaut, qui impose le calcul exact et par échantillonage de Gibbs sinon.

```
##
## bCalc = TRUE
## Model posterior probabilities are calculated exactly
```

```
##
##    Top10Models  PostProb
## 1           15 -682.8192
## 2           17 -683.7381
## 3        13 15 -683.8099
## 4        14 15 -683.9271
## 5         7 15 -683.9945
## 6        12 15 -684.0261
## 7         4 15 -684.0472
## 8        15 17 -684.1015
## 9         9 15 -684.1287
## 10        5 15 -684.1395

## $top10models
##  [1] "15"    "17"    "13 15" "14 15" "7 15"  "12 15" "4 15"  "15 17"
##  [9] "9 15"  "5 15"
##
## $postprobtop10
##  [1] -682.8192 -683.7381 -683.8099 -683.9271 -683.9945 -684.0261 -684.0472
##  [8] -684.1015 -684.1287 -684.1395
```

On retrouve exactement les mêmes 10 meilleurs modèles. Maintenant plutôt que de faire un calcul exact on va maintenant utiliser l'algoritme d'echantillonnage de Gibbs. L'idée est d'obtenir la distribution d'intérêt à partir des lois conditionnelles, plus facile à calculer. Cette algorithme est surtout intéressant lorsque le paramètre du modèle θ est de dimension >2. Ici $\theta = (\beta, \sigma)$

2.3.3 Choix de modèle : par échantillonnage de Gibbs

- Méthode N°1 - A partir de la fonction (modifée) *ModChoBayesReg* du package Bayess

```
##
## bCalc + false
## Model posterior probabilities are calculated by Gibbs
##
##    Top10Models PostProb
## 1           15   0.3557
## 2           17   0.0562
## 3        13 15   0.0338
## 4        14 15   0.0303
## 5         7 15   0.0256
## 6        12 15   0.0237
## 7        15 17   0.0224
## 8         2 15   0.0205
## 9         9 15   0.0196
## 10        4 15   0.0186

## $top10models
##  [1] "15"    "17"    "13 15" "14 15" "7 15"  "12 15" "15 17" "2 15"
##  [9] "9 15"  "4 15"
##
## $postprobtop10
##  [1] 0.3556750 0.0561875 0.0337500 0.0303375 0.0255750 0.0237125 0.0224000
##  [8] 0.0204875 0.0196000 0.0186000
```

Cette fois-ci la probabilité de chacun des modèles a pu être calculée. On retrouve des résultats très proches de ceux renvoyés par la fonction de calcul exact vue en cours: *BayesModelChoice_Exact*. Le classement des modèles est le même quelque soit les méthodes utilisées.

- Méthode N°2 - A partir de la méthode vue en cours *BayesModelChoice_Gibbs*

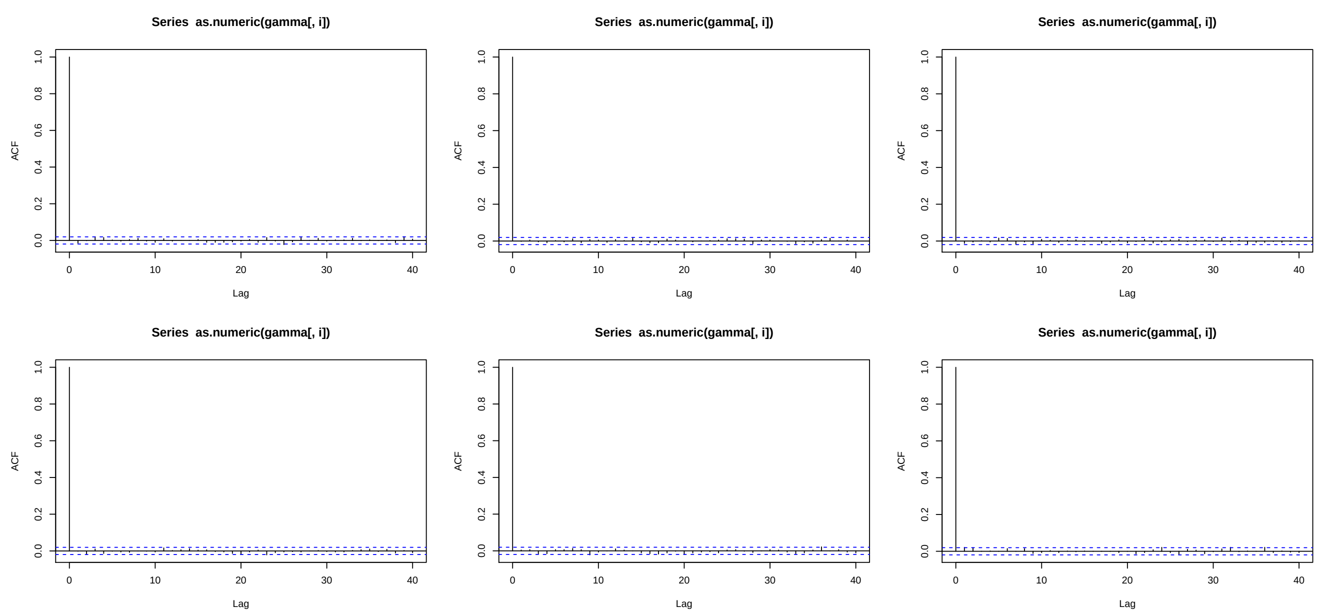
On va maintenant utiliser la fonction implémentée en cours: *BayesModelChoice_Gibbs* et comparer les résultats obtenus.

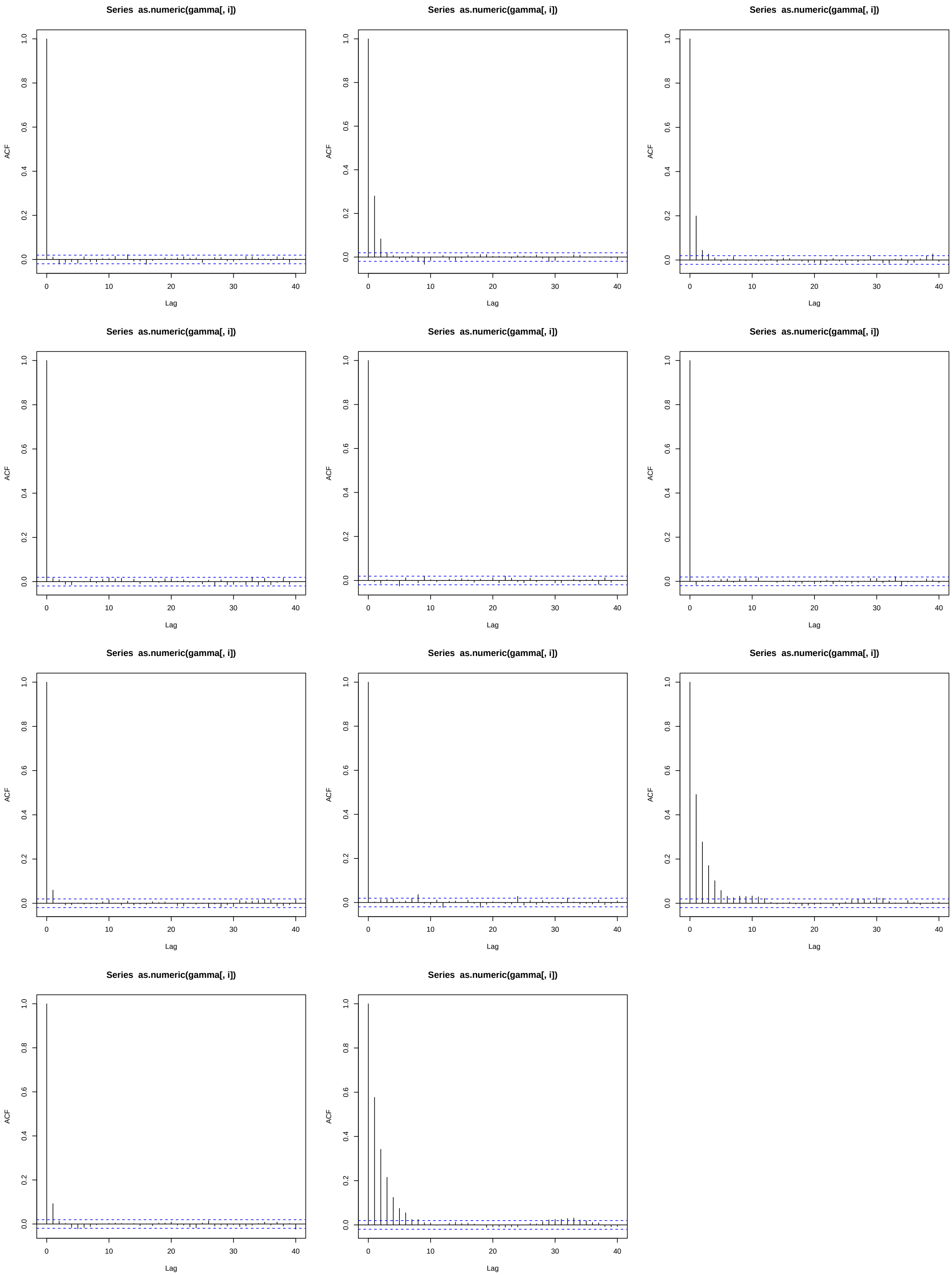
```
##                    X gamma.mean
## 15 Tx_Acc.att_bac1e     0.8431
## 17   Tx_Suc.att_Tot     0.1418
## 13 Tx_Acc.att_bac2e     0.0879
## 14 Tx_Acc.brt_bac1e     0.0667
## 8     Tx_Suc.att_es     0.0660
## 7      Tx_Suc.att_l     0.0651
## 9      Tx_Suc.att_s     0.0626
## 4      Tx_Suc.brt_l     0.0527
## 12 Tx_Acc.brt_bac2e     0.0524
## 16   Tx_Suc.brt_Tot     0.0518
## 10           Eff_2e     0.0498
## 1         Eff_Prs_l     0.0487
## 6      Tx_Suc.brt_s     0.0482
## 5     Tx_Suc.brt_es     0.0480
## 11           Eff_1e     0.0443
## 3         Eff_Prs_s     0.0439
## 2        Eff_Prs_es     0.0434
```

On retrouve le même classement pour les 2 premières variables. Et un classement assez voisin pour les suivantes. On regarde maintenant, la convergence de la méthode.

- Vérication de la convergence et du mélange - autocorrélations:

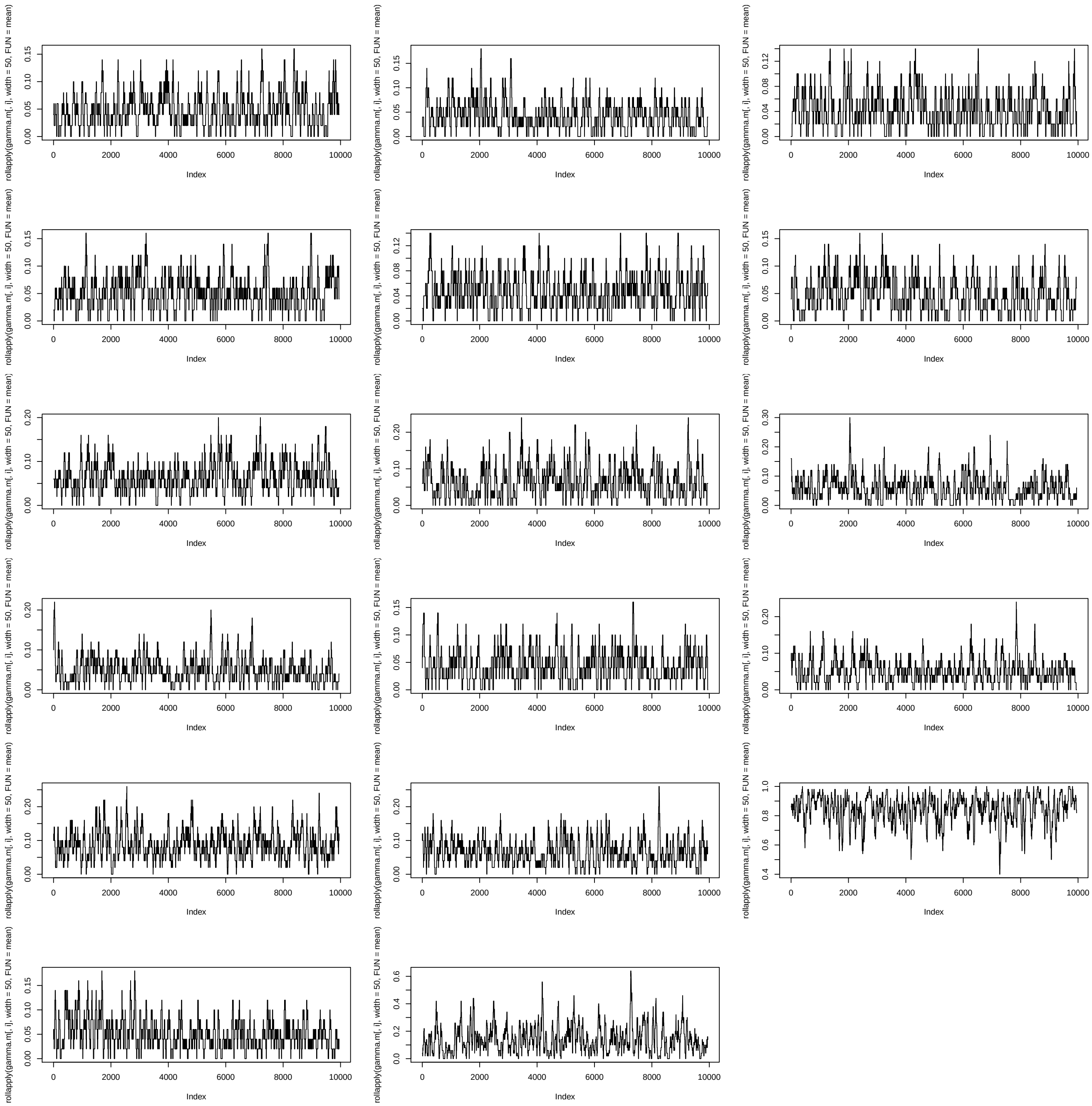
On vérifie le mélange de la chaine de Markov à l'aide des autocorrélations. Dans tous les cas les autocorrélations décroissent rapidement. On n'a pas besoin de sous-échantillonner.




Series as.numeric(gamma[, i])
ACF
Lag
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40

- Vérication de la convergence et du mélange - trace:

A l'aide de la trace (on utilise une moyenne glissante puisque les valeurs sont binaires).



Pour chacune des variables, si on regarde précisemment on peut voir que souvent la chaine reste bloquée au même endroit pendant plusieurs itérations. Il est possible que l'algorithme ne soit pas correctment dimensionné. Par la suite on utilisera plutôt la fonction *ModChoBayesReg2* avec le paramètre (bCalc=FALSE) qui a donné de bons résultats, comme on l'a vu comparables au calcul exact. Que ce soit avec la méthode vue en cours et codée ici avec la fonction *BayesModelChoice_Exact* ou bien avec la fonction reprise du package Bayess: *ModChoBayesReg2* cette fois avec le paramètre (bCalc=TRUE).

Pour comparaison, on va maintenant reprendre l'analyse et effectuer une analyse fréquentiste classique.

2.3.4 Comparaison au résultat obtenu par une analyse fréquentiste

- Analyse fréquentiste

On considère un modéle de régression linéaire gaussien i.e

$$y \mid \alpha, \beta, \sigma^2 \sim N_n(\alpha 1_n + X\beta, \sigma^2 I_n)$$

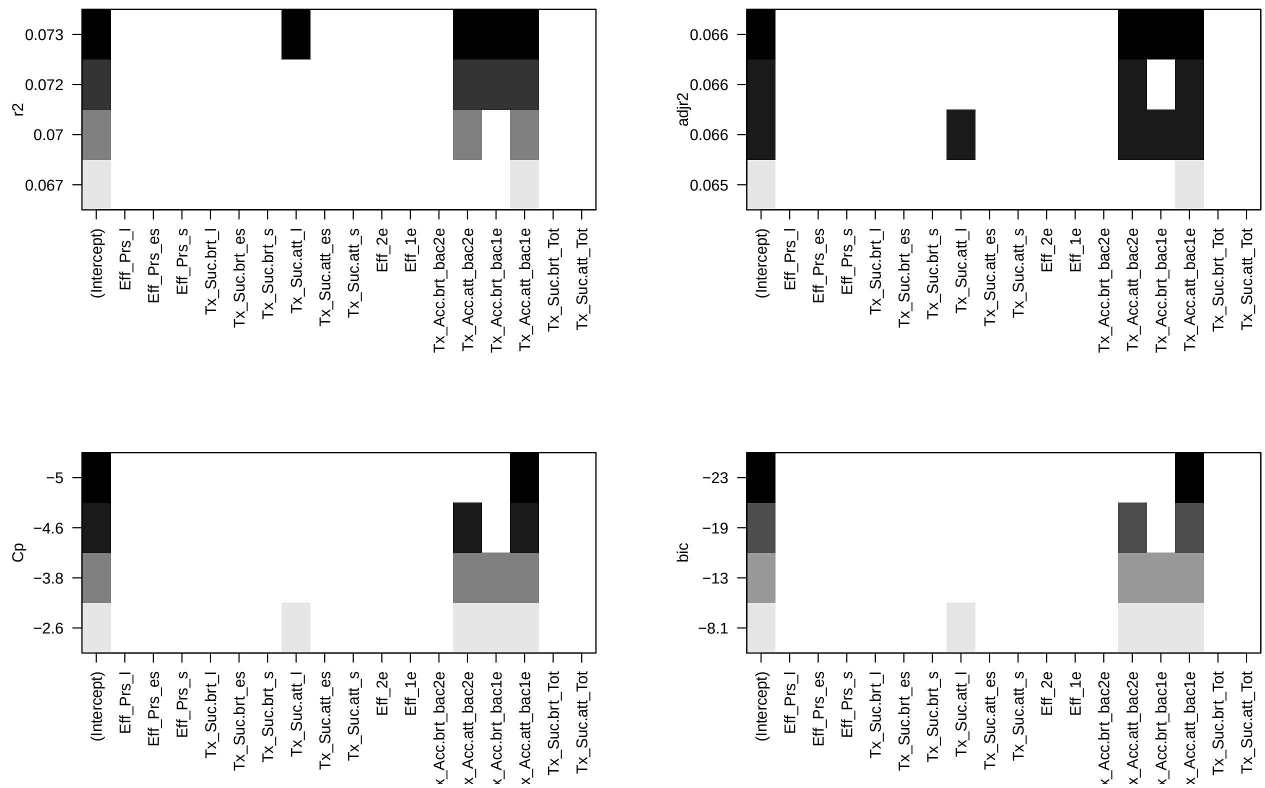
où N_n est la distribution de la loi normale en dimension n.

Ainsi les y_i suivent des lois normales indépendantes avec :

$$E(y_i \mid \alpha, \beta, \sigma^2) = \alpha + \sum_{j=1}^{p} \beta_j x_{ij}$$

$$V(y_i \mid \alpha, \beta, \sigma^2) = \sigma^2$$

```
##
## Call:
## lm(formula = Barre ~ ., data = d.reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.44317 -0.50721 -0.05068  0.39502  2.67793
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.265726   0.041149 127.966   <2e-16 ***
## Eff_Prs_l         0.053333   0.076316   0.699   0.4850
## Eff_Prs_es       -0.026963   0.093958  -0.287   0.7743
## Eff_Prs_s        -0.020844   0.130785  -0.159   0.8734
## Tx_Suc.brt_l      0.007733   0.065536   0.118   0.9061
## Tx_Suc.brt_es     0.100143   0.091782   1.091   0.2758
## Tx_Suc.brt_s      0.170614   0.128597   1.327   0.1852
## Tx_Suc.att_l     -0.150586   0.113031  -1.332   0.1834
## Tx_Suc.att_es     0.013747   0.155069   0.089   0.9294
## Tx_Suc.att_s     -0.144943   0.199290  -0.727   0.4674
## Eff_2e            0.086840   0.186957   0.464   0.6425
## Eff_1e           -0.118486   0.201057  -0.589   0.5559
## Tx_Acc.brt_bac2e  0.161617   0.113712   1.421   0.1559
## Tx_Acc.att_bac2e -0.228736   0.144497  -1.583   0.1141
## Tx_Acc.brt_bac1e -0.311735   0.163039  -1.912   0.0564 .
## Tx_Acc.att_bac1e  0.591735   0.253937   2.330   0.0202 *
## Tx_Suc.brt_Tot   -0.031198   0.210838  -0.148   0.8824
## Tx_Suc.att_Tot    0.109418   0.376114   0.291   0.7712
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9347 on 498 degrees of freedom
## Multiple R-squared:  0.08332,    Adjusted R-squared:  0.05202
## F-statistic: 2.663 on 17 and 498 DF,  p-value: 0.0003462
```



Dans l'ordre croissant de significativité les variables N°7, 13, 14 et 15 ressortent, quelque soit le critère: R2, R2 ajusté, cp de Mallow et Bic. Cependant comme dans la regréssion linéaire Bayésienne, la 15ème variable semble prépondérante: *taux_acces_attendu_premiere_bac*.

```
summary(step_mod)
```

```
##
## Call:
## lm(formula = Barre ~ Tx_Acc.att_bac1e, data = d.reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.50342 -0.52743 -0.06094  0.42639  2.66655
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.26573    0.04087 128.845  < 2e-16 ***
## Tx_Acc.att_bac1e  0.24799    0.04091   6.062  2.6e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9284 on 514 degrees of freedom
## Multiple R-squared:  0.06672,    Adjusted R-squared:  0.06491
## F-statistic: 36.75 on 1 and 514 DF,  p-value: 2.605e-09
```

3 covariables se dégagent : - taux_reussite_attendu_serie_l - taux_acces_attendu_premiere_bac - taux_acces_brut_seconde_

Au vu des p-valeurs des tests de Fisher, renvoyées par un test anova (cf. annexe) on peut envisager de se passer des variables : *taux_acces_brut_premiere_bac* et *taux_acces_brut_seconde_bac* on conserve donc le plus petit modèle *step_mod* composé de la variable *taux_acces_attendu_premiere_bac* qui est la plus significative et de la variable *taux_reussite_attendu_serie_l* qui est assez particulière du fait qu'elle est beaucoup moins corrélée que les autres (c. Introduction).

Avec la régression linéaire classique on trouve la même variable significative qu'avec l'approche Bayésienne, à savoir la 15ème variable *taux_acces_attendu_premiere_bac* qui est prépondérante.

2.3.5 Préselection des covariables

On pourrait utiliser l'échantilloneur de Gibbs pour effectuer une préselection des variables ou bien les Bayes factor et ensuite faire un calcul exact de modèle. Mais ici ce n'est pas encore obligatoire, et on peut se passer de cette préselection. Le calcul exact incluant tous les modèles est encore rapide.

2.4 Mutations en mathématiques et anglais

2.4.1 Régression linéaire bayésienne et choix des covariables à l'aide des Bayes factors

Pour comparer les modèles on peut utiliser les facteurs de Bayes. On test l'hypothèse H_0, $\forall i = 1, ..., 17$ et on calcul le Bayes Factor à partir de la fonction *BayesReg*, pour g=n et g=1.

- Mutations en mathématiques - A partir de la fonction *BayesReg pour g=n*

```
##
##             PostMean PostStError Log10bf EvidAgaH0
## Intercept   4.4485      0.0024
## x1          0.0009      0.0043  -0.879
## x2          0.0005      0.0056 -0.8874
## x3         -0.0073      0.0084  -0.719
## x4         -0.0029      0.0036 -0.7414
## x5         -0.0099      0.0058 -0.2424
## x6          0.1126      0.0083  17.403    (****)
## x7         -0.0036      0.0065 -0.8201
## x8          0.0019      0.0094 -0.8803
## x9          0.0152      0.0127 -0.5694
## x10         0.0135      0.0113 -0.5696
## x11        -0.0091      0.0129 -0.7772
## x12         0.0061      0.0085 -0.7739
## x13        -0.0032      0.0092 -0.8616
## x14        -0.0133      0.0114 -0.5872
## x15         0.0067      0.0150 -0.8438
## x16         0.0223      0.0138 -0.3114
## x17        -0.0150      0.0221 -0.7865
##
##
## Posterior Mean of Sigma2: 3e-04
## Posterior StError of Sigma2: 5e-04
```

```
## $postmeancoeff
##  [1]  4.4484932055  0.0009181915  0.0004816153 -0.0073257345 -0.0029096829
##  [6] -0.0099061914  0.1125856153 -0.0035783593  0.0018547708  0.0152106026
## [11]  0.0135052792 -0.0091018424  0.0060679499 -0.0032334392 -0.0132804478
## [16]  0.0067330257  0.0222971004 -0.0149702309
##
## $postsqrtcoeff
##                          Eff_Prs_l       Eff_Prs_es        Eff_Prs_s
```

```
##      0.002373760      0.004341124      0.005550524      0.008394782
##     Tx_Suc.brt_l    Tx_Suc.brt_es     Tx_Suc.brt_s     Tx_Suc.att_l
##      0.003580189      0.005767218      0.008315673      0.006453335
##    Tx_Suc.att_es     Tx_Suc.att_s           Eff_2e           Eff_1e
##      0.009384830      0.012677773      0.011260430      0.012874452
## Tx_Acc.brt_bac2e Tx_Acc.att_bac2e Tx_Acc.brt_bac1e Tx_Acc.att_bac1e
##      0.008458629      0.009244208      0.011393519      0.015001147
##   Tx_Suc.brt_Tot   Tx_Suc.att_Tot
##      0.013752783      0.022120718
##
## $log10bf
##  [1] -0.8790183 -0.8873825 -0.7189611 -0.7413945 -0.2424129 17.4029995
##  [7] -0.8201145 -0.8802941 -0.5694230 -0.5696480 -0.7771668 -0.7738656
## [13] -0.8615897 -0.5871582 -0.8438499 -0.3113638 -0.7864908
##
## $postmeansigma2
## [1] 0.0003324494
##
## $postvarsigma2
## [1] 2.250642e-07
```

- Mutations en mathématiques - A partir de la fonction *BayesReg pour g=1*

```
##
##           PostMean PostStError Log10bf EvidAgaH0
## Intercept   4.4485      0.0116
## x1          0.0005      0.0152 -0.1503
## x2          0.0002      0.0194 -0.1505
## x3         -0.0037      0.0293 -0.1469
## x4         -0.0015      0.0125 -0.1474
## x5         -0.0050      0.0202 -0.1365
## x6          0.0572      0.0291  0.6928      (**)
## x7         -0.0018      0.0226 -0.1491
## x8          0.0009      0.0328 -0.1503
## x9          0.0077      0.0443 -0.1437
## x10         0.0069      0.0394 -0.1437
## x11        -0.0046      0.0450 -0.1481
## x12         0.0031      0.0296 -0.1481
## x13        -0.0016      0.0323 -0.1499
## x14        -0.0068      0.0398 -0.1441
## x15         0.0034      0.0524 -0.1496
## x16         0.0113      0.0481  -0.138
## x17        -0.0076      0.0773 -0.1483
##
##
## Posterior Mean of Sigma2: 0.008
## Posterior StError of Sigma2: 0.0114
```

```
## $postmeancoeff
##  [1]  4.4484932055  0.0004668770  0.0002448891 -0.0037249497 -0.0014794998
##  [6] -0.0050370465  0.0572469230 -0.0018195047  0.0009431038  0.0077342047
## [11]  0.0068670911 -0.0046280555  0.0030853983 -0.0016441216 -0.0067527701
## [16]  0.0034235724  0.0113375087 -0.0076119818
##
## $postsqrtcoeff
##                         Eff_Prs_l       Eff_Prs_es        Eff_Prs_s
```

```
##       0.01163727       0.01517579       0.01940363       0.02934664
##     Tx_Suc.brt_l    Tx_Suc.brt_es     Tx_Suc.brt_s     Tx_Suc.att_l
##       0.01251569       0.02016115       0.02907009       0.02255969
##    Tx_Suc.att_es     Tx_Suc.att_s           Eff_2e           Eff_1e
##       0.03280767       0.04431921       0.03936443       0.04500676
## Tx_Acc.brt_bac2e Tx_Acc.att_bac2e Tx_Acc.brt_bac1e Tx_Acc.att_bac1e
##       0.02956984       0.03231608       0.03982968       0.05244130
##   Tx_Suc.brt_Tot   Tx_Suc.att_Tot
##       0.04807725       0.07733004
##
## $log10bf
##  [1] -0.1503021 -0.1504792 -0.1468921 -0.1473726 -0.1364845  0.6927904
##  [7] -0.1490521 -0.1503291 -0.1436676 -0.1436725 -0.1481371 -0.1480666
## [13] -0.1499329 -0.1440520 -0.1495565 -0.1380143 -0.1483360
##
## $postmeansigma2
## [1] 0.007990139
##
## $postvarsigma2
## [1] 0.0001300062
```

- Mutations en anglais - A partir de la fonction *BayesReg* pour $g = n$

```
##
##           PostMean PostStError Log10bf EvidAgaH0
## Intercept   4.4372      0.0026
## x1          0.0037      0.0067 -0.7934
## x2         -0.0018      0.0063 -0.8442
## x3         -0.0077      0.0090 -0.6964
## x4         -0.0034      0.0045 -0.7355
## x5         -0.0071      0.0060 -0.5435
## x6          0.1127      0.0092 14.7148    (****)
## x7         -0.0047      0.0093 -0.8058
## x8         -0.0005      0.0093 -0.8614
## x9          0.0220      0.0123 -0.1653
## x10         0.0098      0.0133 -0.7408
## x11        -0.0060      0.0156 -0.8285
## x12         0.0089      0.0083 -0.6046
## x13         0.0006      0.0111 -0.8614
## x14        -0.0144      0.0108 -0.4687
## x15         0.0018      0.0208 -0.8605
## x16         0.0135      0.0150 -0.6824
## x17        -0.0123      0.0273 -0.8162
##
##
## Posterior Mean of Sigma2: 4e-04
## Posterior StError of Sigma2: 5e-04
```

```
## $postmeancoeff
##  [1]  4.4372369919  0.0036867304 -0.0017669259 -0.0077041284 -0.0034116450
##  [6] -0.0071453420  0.1127051681 -0.0046728986 -0.0005197538  0.0219900010
## [11]  0.0097957223 -0.0060288205  0.0089323959  0.0006433518 -0.0144371450
## [16]  0.0017860707  0.0134655936 -0.0123362118
##
## $postsqrtcoeff
##                         Eff_Prs_l       Eff_Prs_es        Eff_Prs_s
```

```
##       0.002647624       0.006675923       0.006266908       0.008962346
##      Tx_Suc.brt_l     Tx_Suc.brt_es      Tx_Suc.brt_s      Tx_Suc.att_l
##       0.004544736       0.005974895       0.009171423       0.009344576
##     Tx_Suc.att_es      Tx_Suc.att_s            Eff_2e            Eff_1e
##       0.009297754       0.012326963       0.013330346       0.015625451
## Tx_Acc.brt_bac2e Tx_Acc.att_bac2e Tx_Acc.brt_bac1e Tx_Acc.att_bac1e
##       0.008319783       0.011117558       0.010845486       0.020843560
##     Tx_Suc.brt_Tot    Tx_Suc.att_Tot
##       0.015037474       0.027340169
##
## $log10bf
##  [1] -0.7934247 -0.8441862 -0.6963787 -0.7355031 -0.5435336 14.7148447
##  [7] -0.8057643 -0.8614317 -0.1653003 -0.7407609 -0.8285434 -0.6046354
## [13] -0.8613811 -0.4687182 -0.8604785 -0.6823750 -0.8162193
##
## $postmeansigma2
## [1] 0.0003645155
##
## $postvarsigma2
## [1] 2.712794e-07
```

- Mutations en anglais - A partir de la fonction *BayesReg* pour $g = 1$

```
##
##             PostMean PostStError Log10bf EvidAgaH0
## Intercept     4.4372      0.0124
## x1            0.0019      0.0223 -0.1489
## x2           -0.0009      0.0209 -0.1501
## x3           -0.0039      0.0300 -0.1466
## x4           -0.0017      0.0152 -0.1476
## x5           -0.0036      0.0200  -0.143
## x6            0.0574      0.0307  0.6156      (**)
## x7           -0.0024      0.0312 -0.1492
## x8           -0.0003      0.0311 -0.1505
## x9            0.0112      0.0412 -0.1338
## x10           0.0050      0.0446 -0.1477
## x11          -0.0031      0.0522 -0.1497
## x12           0.0046      0.0278 -0.1445
## x13           0.0003      0.0372 -0.1505
## x14          -0.0074      0.0363 -0.1412
## x15           0.0009      0.0697 -0.1505
## x16           0.0069      0.0503 -0.1463
## x17          -0.0063      0.0914 -0.1494
##
##
## Posterior Mean of Sigma2: 0.008
## Posterior StError of Sigma2: 0.0114

## $postmeancoeff
##  [1]  4.4372369919  0.0018788145 -0.0009004526 -0.0039261424 -0.0017386268
##  [6] -0.0036413762  0.0574362876 -0.0023813810 -0.0002648745  0.0112064428
## [11]  0.0049920508 -0.0030723797  0.0045520864  0.0003278620 -0.0073573912
## [16]  0.0009102091  0.0068622737 -0.0062867233
##
## $postsqrtcoeff
##                          Eff_Prs_l        Eff_Prs_es         Eff_Prs_s
```

```
##       0.01239754       0.02231577       0.02094854       0.02995865
##      Tx_Suc.brt_l    Tx_Suc.brt_es     Tx_Suc.brt_s     Tx_Suc.att_l
##       0.01519180       0.01997243       0.03065754       0.03123634
##     Tx_Suc.att_es     Tx_Suc.att_s           Eff_2e           Eff_1e
##       0.03107983       0.04120564       0.04455967       0.05223158
## Tx_Acc.brt_bac2e Tx_Acc.att_bac2e Tx_Acc.brt_bac1e Tx_Acc.att_bac1e
##       0.02781074       0.03716293       0.03625347       0.06967428
##    Tx_Suc.brt_Tot   Tx_Suc.att_Tot
##       0.05026613       0.09139065
##
## $log10bf
##  [1] -0.1489131 -0.1500974 -0.1466340 -0.1475552 -0.1430048  0.6156412
##  [7] -0.1492015 -0.1504986 -0.1338109 -0.1476787 -0.1497330 -0.1444615
## [13] -0.1504974 -0.1412105 -0.1504764 -0.1463036 -0.1494456
##
## $postmeansigma2
## [1] 0.007992351
##
## $postvarsigma2
## [1] 0.0001304169
```

- Conclusion

La 6ème variable: *taux_brut_de_reussite_serie_s* est prépondérante dans tous les cas.

2.4.2 Choix de modèles par test de tous les modèles ou Gibbs-sampler

On utilise la fonction *ModChoBayesReg* du package *Bayess*

- Mutations en Math

```
##
## Number of variables greather than 15
## Model posterior probabilities are estimated by using an MCMC algorithm
##
##    Top10Models PostProb
## 1          5 6   0.0530
## 2            6   0.0225
## 3          6 8   0.0153
## 4        5 6 9   0.0145
## 5          6 7   0.0124
## 6        5 6 7   0.0104
## 7         6 14   0.0097
## 8       5 6 11   0.0094
## 9       5 6 16   0.0086
## 10      5 6 10   0.0085
```

```
## $top10models
##  [1] "5 6"    "6"      "6 8"    "5 6 9"  "6 7"    "5 6 7"  "6 14"
##  [8] "5 6 11" "5 6 16" "5 6 10"
##
## $postprobtop10
##  [1] 0.0529500 0.0225125 0.0152625 0.0145375 0.0124375 0.0104250 0.0096750
##  [8] 0.0093500 0.0085750 0.0085125
```

La 6ème covariable est omniprésente dans tous les modèles. La probabilité à piriori du modèle constitué de cette seule variable est écrasante.

- Mutations en Anglais

```
##
## Number of variables greather than 15
## Model posterior probabilities are estimated by using an MCMC algorithm
##
##    Top10Models PostProb
## 1            6   0.0378
## 2          5 6   0.0178
## 3          6 8   0.0150
## 4        6 8 9   0.0127
## 5         6 14   0.0122
## 6          6 7   0.0110
## 7       3 6 11   0.0106
## 8       3 6 10   0.0092
## 9          3 6   0.0085
## 10        6 16   0.0079

## $top10models
##  [1] "6"      "5 6"    "6 8"    "6 8 9"  "6 14"   "6 7"    "3 6 11"
##  [8] "3 6 10" "3 6"    "6 16"
##
## $postprobtop10
##  [1] 0.0377750 0.0178375 0.0150500 0.0127000 0.0121750 0.0109500 0.0105500
##  [8] 0.0091500 0.0085375 0.0079250
```

On retrouve la encore la prédominance de la 6ème variable : Suc.brt_s soit le *taux_brut_de_reussite_serie_s*.

2.4.3 Comparaison au résultat obtenu par une analyse fréquentiste

Analyse fréquentiste - Mutations en Mathématiques et en Anglais

```
##
## Call:
## lm(formula = Barre ~ ., data = d.math.reg)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0281099 -0.0041035  0.0004529  0.0052922  0.0148572
##
## Coefficients:
##                    Estimate Std. Error  t value Pr(>|t|)
## (Intercept)       4.4484932  0.0012583 3535.386  < 2e-16 ***
## Eff_Prs_l         0.0009418  0.0023405    0.402  0.68949
## Eff_Prs_es        0.0004940  0.0029925    0.165  0.86970
## Eff_Prs_s        -0.0075138  0.0045260   -1.660  0.10451
## Tx_Suc.brt_l     -0.0029844  0.0019302   -1.546  0.12975
## Tx_Suc.brt_es    -0.0101606  0.0031093   -3.268  0.00220 **
## Tx_Suc.brt_s      0.1154766  0.0044833   25.757  < 2e-16 ***
## Tx_Suc.att_l     -0.0036702  0.0034792   -1.055  0.29765
## Tx_Suc.att_es     0.0019024  0.0050597    0.376  0.70886
## Tx_Suc.att_s      0.0156012  0.0068351    2.283  0.02772 *
```

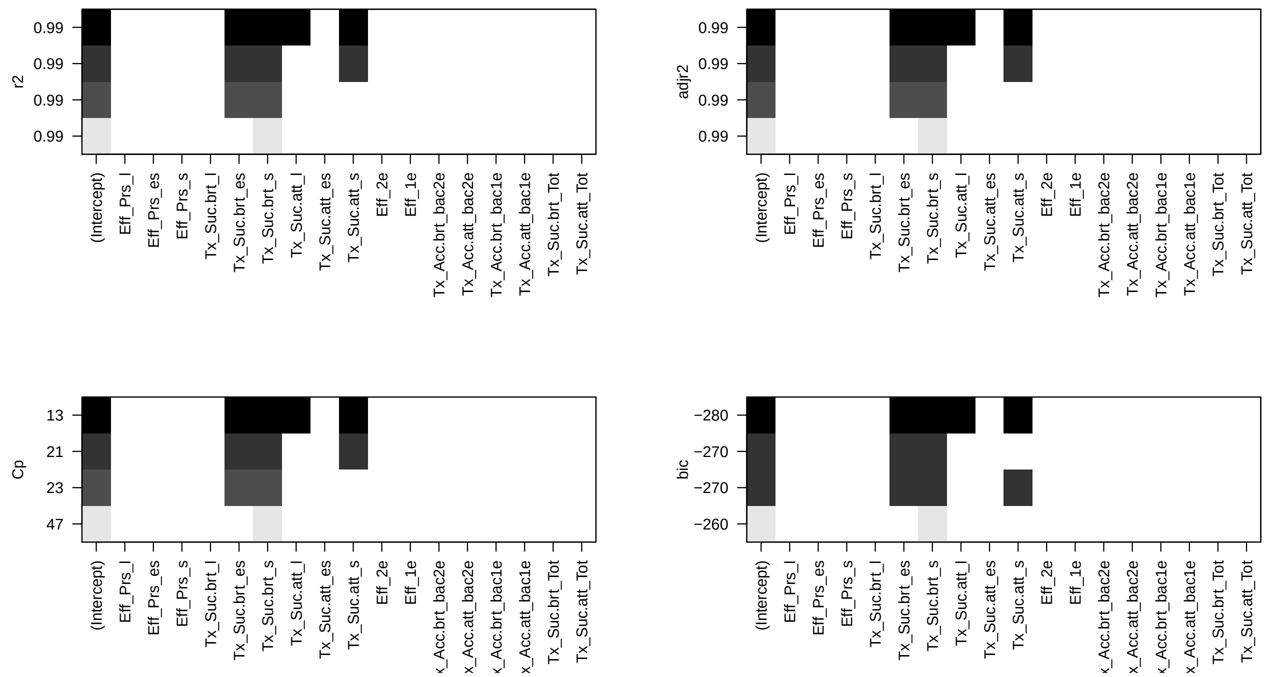
```
## Eff_2e            0.0138521  0.0060709    2.282  0.02777 *
## Eff_1e           -0.0093356  0.0069411   -1.345  0.18603
## Tx_Acc.brt_bac2e  0.0062238  0.0045604    1.365  0.17978
## Tx_Acc.att_bac2e -0.0033165  0.0049839   -0.665  0.50950
## Tx_Acc.brt_bac1e -0.0136215  0.0061427   -2.218  0.03219 *
## Tx_Acc.att_bac1e  0.0069059  0.0080877    0.854  0.39813
## Tx_Suc.brt_Tot    0.0228697  0.0074147    3.084  0.00364 **
## Tx_Suc.att_Tot   -0.0153546  0.0119261   -1.287  0.20515
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009665 on 41 degrees of freedom
## Multiple R-squared:  0.9957, Adjusted R-squared:  0.9939
## F-statistic: 558.7 on 17 and 41 DF,  p-value: < 2.2e-16
```

Dans le cas des mutations en mathématiques ci-dessus on retrouve en gros les mêmes variables significatives que dans le modèle de regression Bayésienne. La variable N°6 et dans une moindre mesure la variable N°5.

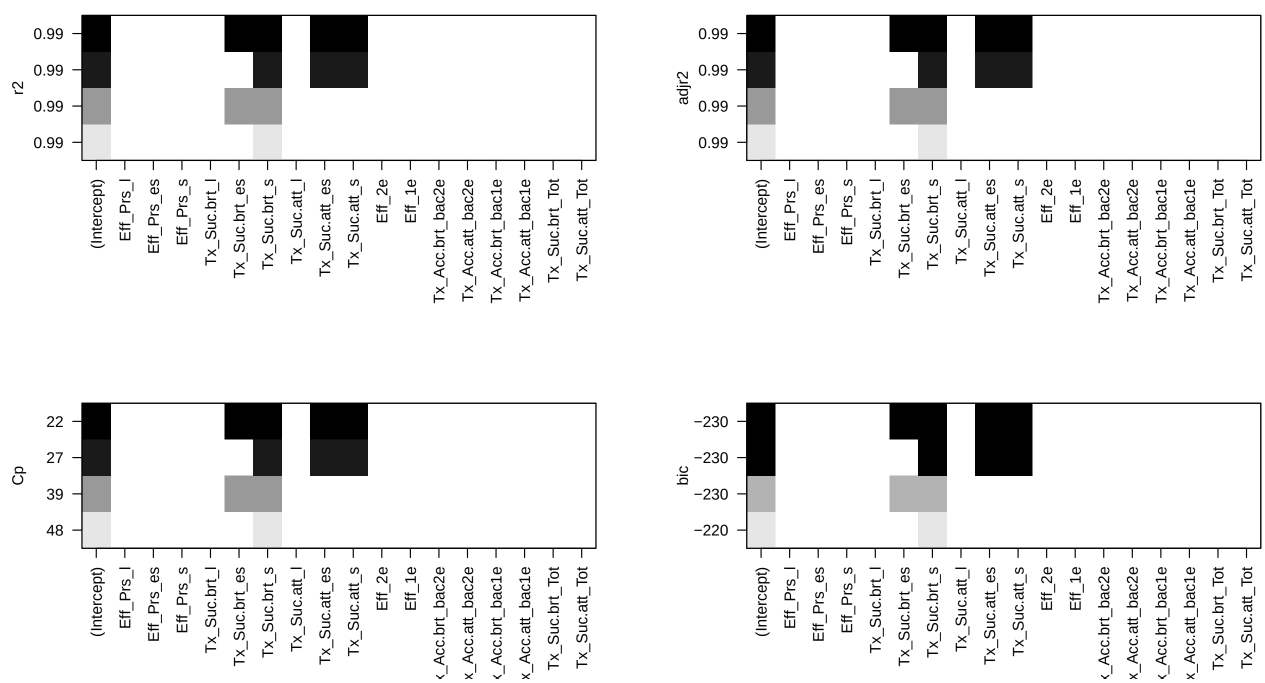
```
##
## Call:
## lm(formula = Barre ~ ., data = d.en.reg)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0273618 -0.0051516  0.0003793  0.0049311  0.0160208
##
## Coefficients:
##                    Estimate Std. Error  t value Pr(>|t|)
## (Intercept)       4.4372370  0.0013462 3296.232  < 2e-16 ***
## Eff_Prs_l         0.0037943  0.0034602    1.097  0.28055
## Eff_Prs_es       -0.0018185  0.0032482   -0.560  0.57926
## Eff_Prs_s        -0.0079289  0.0046453   -1.707  0.09697 .
## Tx_Suc.brt_l     -0.0035112  0.0023556   -1.491  0.14529
## Tx_Suc.brt_es    -0.0073538  0.0030969   -2.375  0.02335 *
## Tx_Suc.brt_s      0.1159933  0.0047537   24.401  < 2e-16 ***
## Tx_Suc.att_l     -0.0048092  0.0048434   -0.993  0.32775
## Tx_Suc.att_es    -0.0005349  0.0048191   -0.111  0.91227
## Tx_Suc.att_s      0.0226316  0.0063892    3.542  0.00118 **
## Eff_2e            0.0100815  0.0069093    1.459  0.15371
## Eff_1e           -0.0062047  0.0080989   -0.766  0.44889
## Tx_Acc.brt_bac2e  0.0091930  0.0043122    2.132  0.04033 *
## Tx_Acc.att_bac2e  0.0006621  0.0057624    0.115  0.90920
## Tx_Acc.brt_bac1e -0.0148583  0.0056213   -2.643  0.01233 *
## Tx_Acc.att_bac1e  0.0018382  0.0108035    0.170  0.86590
## Tx_Suc.brt_Tot    0.0138584  0.0077941    1.778  0.08434 .
## Tx_Suc.att_Tot   -0.0126961  0.0141707   -0.896  0.37659
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009707 on 34 degrees of freedom
## Multiple R-squared:  0.9959, Adjusted R-squared:  0.9938
## F-statistic: 484.9 on 17 and 34 DF,  p-value: < 2.2e-16
```

ans le cas des mutations en anglais ci-dessus on trouve des résultats comparable en particulier pour la significativité de la variable N°6: (Tx_Suc.brt_s) *taux_reussite_attendu_serie_s*.

Choix de modèles - méthode *regsubsets*: cas des mutations en mathématiques



Choix de modèles - méthode *regsubsets*: Cas des mutations en anglais



Choix de modèles - méthode *step*: cas des mutations en mathématiques

```r
summary(step_mod.math)
```

```
##
## Call:
## lm(formula = Barre ~ Eff_Prs_s + Tx_Suc.brt_l + Tx_Suc.brt_es +
##     Tx_Suc.brt_s + Tx_Suc.att_l + Tx_Suc.att_s + Eff_2e + Eff_1e +
##     Tx_Acc.brt_bac2e + Tx_Acc.brt_bac1e + Tx_Suc.brt_Tot, data = d.math.reg)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0294954 -0.0036786 -0.0004132  0.0050052  0.0167705
##
## Coefficients:
##                   Estimate Std. Error  t value Pr(>|t|)
## (Intercept)       4.448493   0.001207 3685.525  < 2e-16 ***
## Eff_Prs_s        -0.008526   0.003107   -2.744 0.008576 **
## Tx_Suc.brt_l     -0.002155   0.001588   -1.357 0.181231
## Tx_Suc.brt_es    -0.009207   0.002199   -4.186 0.000123 ***
## Tx_Suc.brt_s      0.115641   0.003715   31.126  < 2e-16 ***
## Tx_Suc.att_l     -0.006721   0.002352   -2.858 0.006337 **
## Tx_Suc.att_s      0.010065   0.003398    2.962 0.004788 **
## Eff_2e            0.016038   0.005499    2.917 0.005409 **
## Eff_1e           -0.009384   0.005941   -1.579 0.120942
## Tx_Acc.brt_bac2e  0.005260   0.003420    1.538 0.130747
## Tx_Acc.brt_bac1e -0.012361   0.005393   -2.292 0.026433 *
## Tx_Suc.brt_Tot    0.019706   0.005949    3.313 0.001783 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009271 on 47 degrees of freedom
## Multiple R-squared:  0.9955, Adjusted R-squared:  0.9944
## F-statistic: 938.1 on 11 and 47 DF,  p-value: < 2.2e-16
```

Choix de modèles - méthode *step*: cas des mutations en anglais

```r
summary(step_mod.en)
```

```
##
## Call:
## lm(formula = Barre ~ Eff_Prs_l + Eff_Prs_s + Tx_Suc.brt_es +
##     Tx_Suc.brt_s + Tx_Suc.att_l + Tx_Suc.att_s + Tx_Acc.brt_bac2e +
##     Tx_Acc.brt_bac1e, data = d.en.reg)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.033886 -0.004001 -0.000235  0.003696  0.018414
##
## Coefficients:
##                   Estimate Std. Error  t value Pr(>|t|)
## (Intercept)       4.437237   0.001304 3402.294  < 2e-16 ***
## Eff_Prs_l         0.006868   0.001844    3.724 0.000565 ***
## Eff_Prs_s        -0.007810   0.002434   -3.209 0.002520 **
## Tx_Suc.brt_es    -0.003258   0.001901   -1.714 0.093792 .
```

```
## Tx_Suc.brt_s       0.123228   0.002901   42.483  < 2e-16 ***
## Tx_Suc.att_l      -0.010269   0.002552   -4.025 0.000227 ***
## Tx_Suc.att_s       0.018634   0.003763    4.952 1.19e-05 ***
## Tx_Acc.brt_bac2e  0.008924   0.002564    3.481 0.001160 **
## Tx_Acc.brt_bac1e -0.014253   0.003323   -4.289 9.96e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009405 on 43 degrees of freedom
## Multiple R-squared:  0.9951, Adjusted R-squared:  0.9942
## F-statistic:  1097 on 8 and 43 DF,  p-value: < 2.2e-16
```

2.5 Conclusion

Pour les mutations en Math et en Anglais, dans l'approche bayésiennne une covariable ressort très nettement: *taux_brut_de_reussite_serie_s*. On retouve la significativité de cette variable, quelque soit la matière math ou anglais. Les résultats obtenus pour l'une ou l'autre des matière sont très proches. Dans l'approche cas fréquentiste (modèle linéaire classique) on trouve encore que cette variable: *taux_brut_de_reussite_serie_s* est très significative. Par contre de nombreuses autres variables sont elles aussi significatives. On a plus de difficulté à sélectionner les variables qui comme ont la vue sont très corrélées. Le modèle de régression Bayésienne apparaît plus parcimonieux. Dans le cadre bayésien la loi à priori choisie G-prior de Zellner a la particularité d'éliminer les corrélations entre covariables. Ceci pouvant peut-être expliquer la différence entre les deux approches. Si on regarde maintenant la valeur des coefficients du $\hat{\beta}$ celui correspondant à la vaériable N°6 est bien plus élévé et prédomine, quelque soit l'approche Bayésienne ou classique.p

Dans le cas général, sans distinction des matières c'est la variable N°15 *taux_acces_attendu_premiere_bac* qui est significative. Et les variables de type plus général (taux d'accès brute / attendu en 1er et 2nd au bac). De la même manière on retrouve des résultats comparables avec la régression classique et Bayésienne. Il faut noter que l'on a choisit une hypothèse de modèle Bayésien pour la prior qui est peut informative avec g=dim(y) et $\beta_0 = 0$. Ce qui explique certainement la similarité des résultats obtenus avec les deux approches.

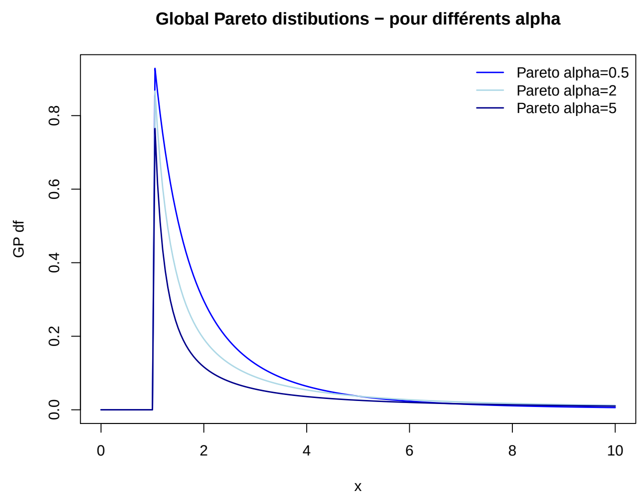
3 Partie II - Loi de Pareto

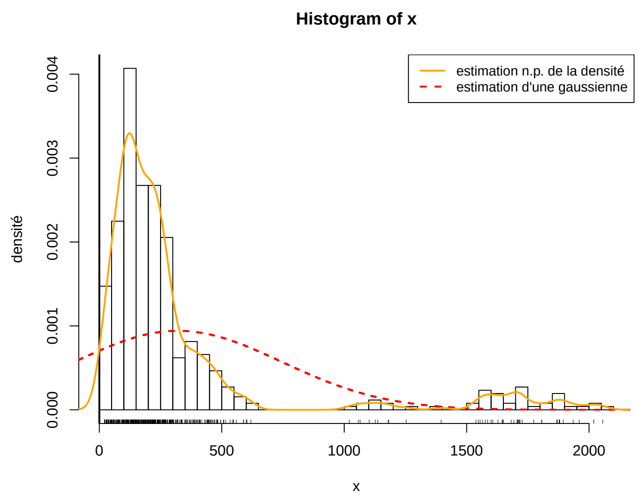
On ignore maintenant les covariables, et on s'intéresse uniquement à la loi du nombre de points nécessaire (colonne Barre). La loi gaussienne peut paraître peu pertinente pour ces données : on va plutôt proposer une loi de Pareto. Pour $m > 0$ et $\alpha > 0$, on dit que $ZPareto(m; \alpha)$ si Z est à valeurs dans $[m; +1[$ de densité:

$f(z \mid \alpha, m) = \alpha \frac{m^\alpha}{z^{\alpha+1}} 1_{[m,+\infty[}$

3.1 Package R pour générer des réalisations d'une loi de Paréto

On peut utiliser le package *extRemes* et la fonction *devd*





En comparant avec l'histogramme de la variable Barre, on voit que la loi de Pareto semble un être un bon choix.

3.2 Choix d'une loi à priori pour α

- Loi de paréto :

$$f(z \mid \alpha, m) = \alpha \frac{m^\alpha}{z^{\alpha+1}} 1_{[m,+\infty[}$$

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    21.0   111.0   196.0   321.9   292.0  2056.0
```

le résumé des données nous ammène à choisir: m=21

A une constante multiplicative près et après transformation en log, on reconnaît une loi exponentielle de paramètre α.

$$f(z \mid \alpha, m) \propto \alpha e^{\alpha log(m/z)}$$

En applicant la transformation : $z \to ln(\frac{z}{m})$ a notre échantillon (Z_i), on a que $ln(\frac{Z}{m}) \sim Exp(\alpha)$

On peut alors estimer le paramètre α par mle à partir de la fonction R: *fitdist* du package *fitdistrplus*.

```
m=21
y.exp<-log(y.tot/m)
fit.exp <- fitdist(y.exp, "exp", method="mle")
fit.exp
```

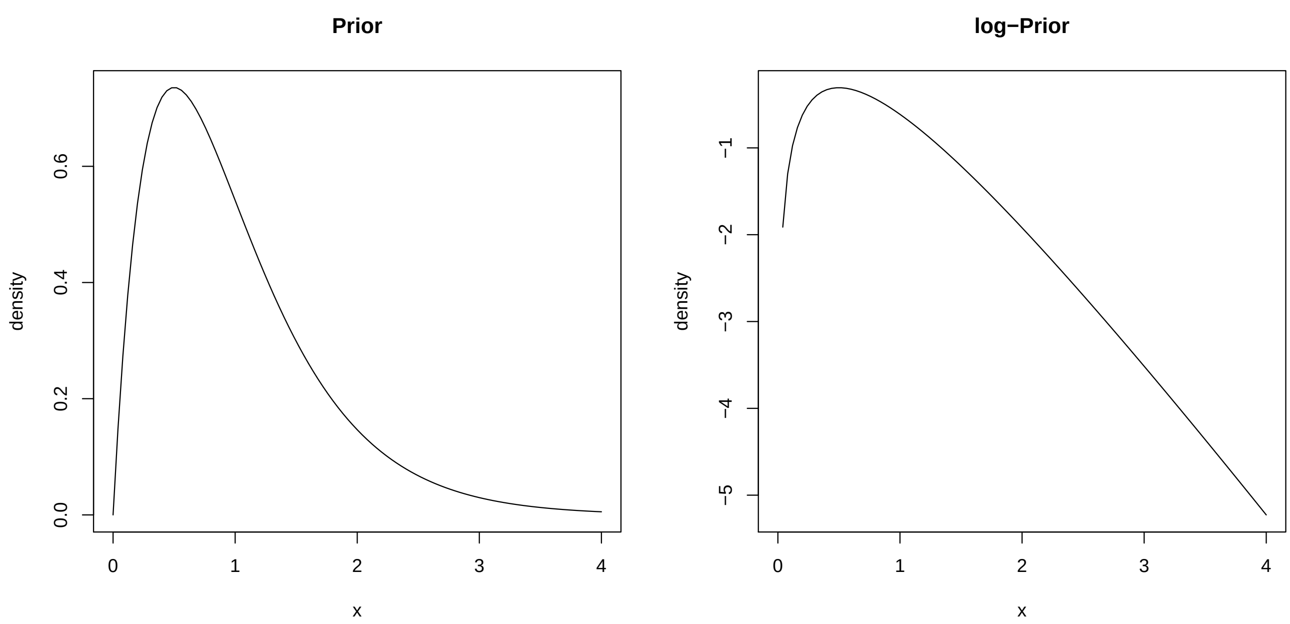
```
## Fitting of the distribution ' exp ' by maximum likelihood
## Parameters:
##       estimate Std. Error
## rate 0.4502063 0.01981913
```

On peut prendre pour loi à priori la loi $\Gamma(a, b)$ de manière à avoir une loi conjuguée. Nous allons tester une loi a priori avec un paramètre shape = 2 et scale = 2.

```
prior = function(alpha){
return(dgamma(alpha, 2, 2))}

logprior = function(alpha){
return(dgamma(alpha, 2, 2, log = T))}
```

```
par(mfrow = c(1, 2))
curve(dgamma(x, 2, 2), xlim=c(0, 4), main="Prior", ylab="density")
curve(dgamma(x, 2, 2, log = T), xlim=c(0, 4), main="log-Prior", ylab="density")
```



3.3 Loi à postériori de α

La loi à postériori correspondante est la loi : $\Gamma(a + n, b + \sum_{i=1}^{n} ln(\frac{Z_i}{m}))$

```
logposterior <- function(m,alpha,y){
n<-length(y)
loglkd <- n*log(alpha) + alpha*n*log(m)-(alpha+1)*sum(log(y))
if(!is.finite(loglkd)) return(-Inf)
return(loglkd+logprior(alpha))
}
```

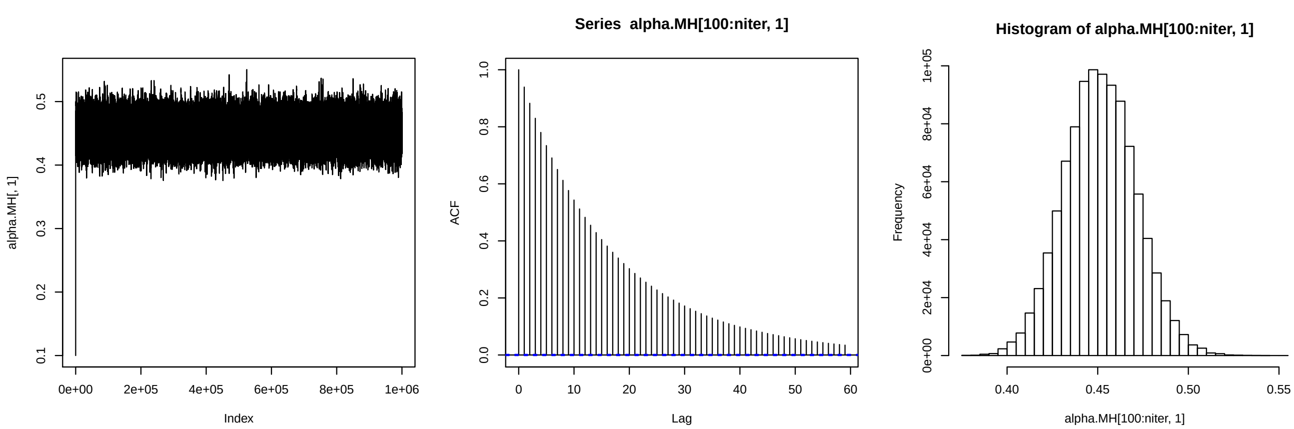
3.4 Echantillon de la loi à postériori de α

Par la méthode de votre choix, tirer un échantillon de la loi a posteriori de α. Donner un intervalle de crédibilité à 95%.

```r
MH <- function(Y,alpha0, m, niter){
  alpha <- matrix(NA, nrow=niter, ncol=1)
  alpha[1] <- alpha0
  for(i in 2:niter){
    proposal <- rgamma(1, 2, 2)
    logalpha <- logposterior(m, proposal, Y)-logposterior(m, alpha[i-1,], Y)
    if(log(runif(1)) < logalpha){alpha[i] <- proposal}
    else{ alpha[i] <- alpha[i-1]}
  }
  return(alpha)}
```

```r
alpha.MH <- MH(Y=y.tot, alpha0=0.1, m=21, niter=1e6)
```

```r
niter=1e6
# Etudions la sortie de l'algorithme
par(mfcol=c(1,3))
# trace
plot(alpha.MH[, 1], type="l")
# autocorrélations
acf(alpha.MH[100:niter, 1])
# histogrammes
hist(alpha.MH[100:niter, 1], breaks=50)
```



L'algorithme est bien dimensionné, la chaîne comme le montre le premier graphique explore bien toute la loi. L'autocorrélation décroit assez rapidement. L'histogramme des valeurs obtenues est régulier.

Intervalle de confiance à 95% et estimation de $\hat{\alpha}$:

```
## [1] 0.4513569
```

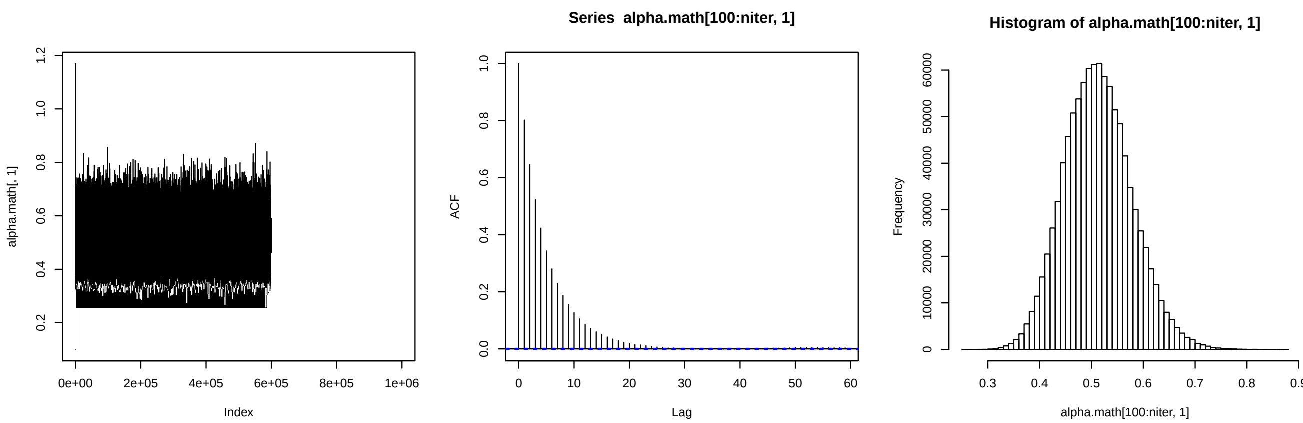
```
##      2.5%     97.5%
## 0.4132579 0.4907452
```

3.5 Analyse pour les mutation en anglais et en math

3.5.1 Calcul du *alpha* par l'alogorithme de Métropolis-Hastigs

```
niter <- 1e6
alpha.math <- MH(y.math, .1, 21, niter)
alpha.en <- MH(y.en, .1, 21, niter)
```

3.5.2 Convergence de l'algorithme de Metropolis-Hastings: mutations en mathématiques



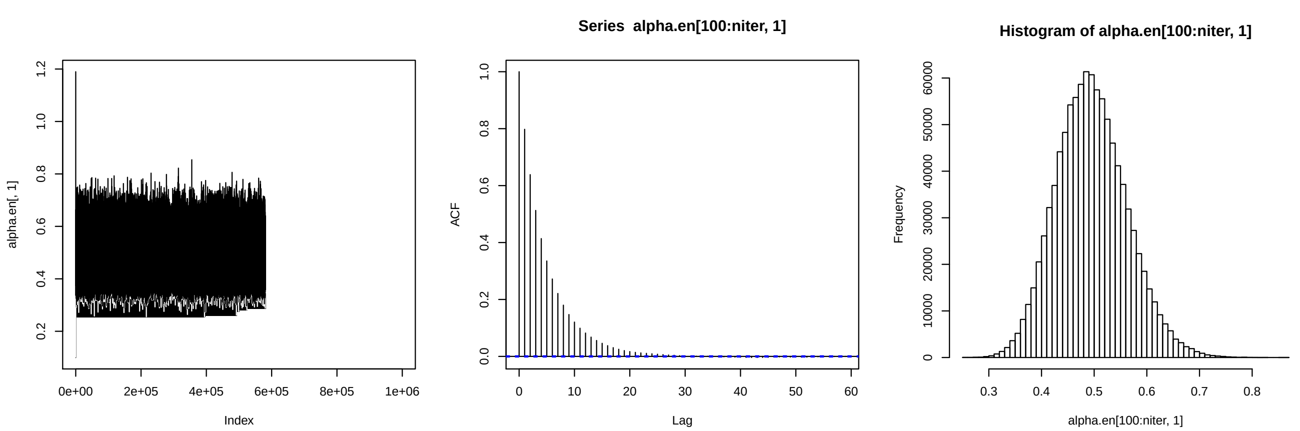
L'algorithme est bien dimensionné, la chaîne comme le montre le premier graphique explore bien toute la loi. L'autocorrélation décroit assez rapidement. L'histogramme des valeurs obtenues est régulier.

Intervalle de confiance à 95% et estimation de $\hat{\alpha}_{math}$:

```
## [1] 0.5129885

##      2.5%     97.5%
## 0.3931538 0.6481325
```

3.5.3 Convergence de l'algorithme de Metropolis-Hastings: mutations en anglais



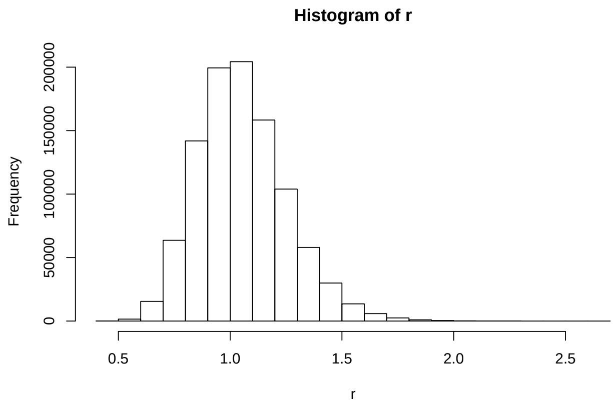
Dans ce cas aussi, l'algorithme est bien dimensionné, la chaîne comme le montre le premier graphique explore bien toute la loi. L'autocorrélation décroit assez rapidement. L'histogramme des valeurs obtenues est régulier.

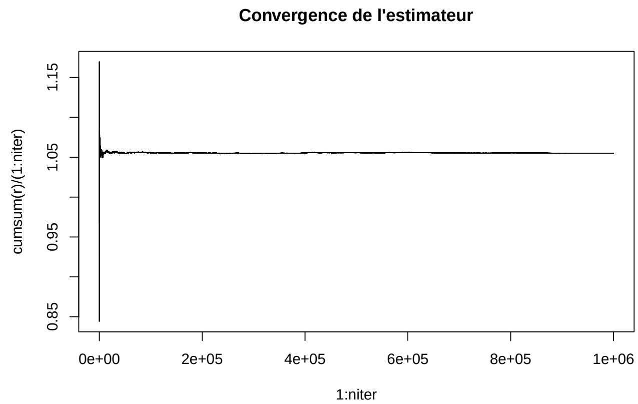
Intervalle de confiance à 95% et estimation de $\hat{\alpha}_{anglais}$:

```
## [1] 0.4951641
```

```
##       2.5%     97.5%
## 0.3731709 0.6345194
```

On va tester l'hypothèse $\alpha_{math} = \alpha_{anglais}$. Pour celà on va estimer l'espérance à postériori du quotient $r_\alpha = \frac{\alpha_{math}}{\alpha_{anglais}}$. On utilise les approximations obtenues par Métropolis-Hastings précédemment pour chacun des α. On regarde la convergence de l'estimateur, qui d'après les graphiques est obtenue à partir de 10000 path au moins.






```
## [1] 1.055071
```

```
## [1] 0.1976557
```

```
##       2.5%     97.5%
## 0.7214509 1.4925029
```

A la vue des résultats on peut conclure à l'égalité des paramètres α pour les mutations en math et en anglais.

4 Annexes

Test des méthodes BayesReg du package Bayess et BayesReg2 version modifiée

```
data(faithful)
BayesReg(faithful[,1],faithful[,2])
```

```
##
##           PostMean PostStError Log10bf EvidAgaH0
## Intercept   3.4878      0.0304
## x1          1.0225      0.0303     Inf    (****)
##
##
## Posterior Mean of Sigma2: 0.2513
## Posterior StError of Sigma2: 0.3561

## $postmeancoeff
## [1] 3.487783 1.022509
##
## $postsqrtcoeff
## [1] 0.03039825 0.03034252
##
## $log10bf
##      [,1]
## [1,]  Inf
##
## $postmeansigma2
## [1] 0.2513425
##
## $postvarsigma2
## [1] 0.1268176
```

```
BayesReg2(faithful[,1],faithful[,2])
```

```
##
##           PostMean PostStError Log10bf EvidAgaH0
## Intercept   3.4878      0.0304
## x1          1.0244      0.0304     Inf    (****)
##
##
## Posterior Mean of Sigma2: 0.2513
## Posterior StError of Sigma2: 0.3561

## $postmeancoeff
## [1] 3.487783 1.024394
##
## $postsqrtcoeff
## [1] 0.03039825 0.03039845
##
## $log10bf
##      [,1]
## [1,]  Inf
##
## $postmeansigma2
```

```
## [1] 0.2513425
##
## $postvarsigma2
## [1] 0.1268176
```

```
data("caterpillar")
y.cat=log(caterpillar$y)
X.cat=as.matrix(caterpillar[,1:8])
```

- Fonction BayesReg

```
BayesReg(y.cat, scale(X.cat))
```

```
##
##           PostMean PostStError Log10bf EvidAgaH0
## Intercept  -0.8133      0.1407
## x1         -0.5039      0.1883  0.7224      (**)
## x2         -0.3755      0.1508  0.5392      (**)
## x3          0.6225      0.3436 -0.0443
## x4         -0.2776      0.2804 -0.5422
## x5         -0.2069      0.1499 -0.3378
## x6          0.2806      0.4760 -0.6857
## x7         -1.0420      0.4178  0.5435      (**)
## x8         -0.0221      0.1531 -0.7609
##
##
## Posterior Mean of Sigma2: 0.6528
## Posterior StError of Sigma2: 0.939
```

```
## $postmeancoeff
## [1] -0.81328069 -0.50390377 -0.37548142  0.62252447 -0.27762947 -0.20688023
## [7]  0.28061938 -1.04204277 -0.02209411
##
## $postsqrtcoeff
##                  x1        x2        x3        x4        x5        x6
## 0.1406514 0.1882559 0.1508271 0.3436217 0.2803657 0.1498641 0.4759505
##        x7        x8
## 0.4178148 0.1530573
##
## $log10bf
## [1]  0.72241000  0.53918250 -0.04430805 -0.54224765 -0.33779821 -0.68568404
## [7]  0.54353138 -0.76091468
##
## $postmeansigma2
## [1] 0.6528327
##
## $postvarsigma2
## [1] 0.8817734
```

```
BayesReg2(y.cat, scale(X.cat))
```

```
##
##           PostMean PostStError Log10bf EvidAgaH0
## Intercept  -0.8133      0.1407
```

```
## x1         -0.5117      0.1912  0.7224      (**)
## x2         -0.3813      0.1532  0.5392      (**)
## x3          0.6322      0.3489 -0.0443
## x4         -0.2819      0.2847 -0.5422
## x5         -0.2101      0.1522 -0.3378
## x6          0.2850      0.4833 -0.6857
## x7         -1.0582      0.4243  0.5435      (**)
## x8         -0.0224      0.1554 -0.7609
##
##
## Posterior Mean of Sigma2: 0.6528
## Posterior StError of Sigma2: 0.939
```

```
## $postmeancoeff
## [1] -0.81328069 -0.51171670 -0.38130319  0.63217659 -0.28193406 -0.21008787
## [7]  0.28497033 -1.05819944 -0.02243668
##
## $postsqrtcoeff
##                  x1        x2        x3        x4        x5        x6
## 0.1406514 0.1911748 0.1531656 0.3489495 0.2847127 0.1521877 0.4833300
##        x7        x8
## 0.4242930 0.1554305
##
## $log10bf
## [1]  0.72241000  0.53918250 -0.04430805 -0.54224765 -0.33779821 -0.68568404
## [7]  0.54353138 -0.76091468
##
## $postmeansigma2
## [1] 0.6528327
##
## $postvarsigma2
## [1] 0.8817734
```

Les légères différences sur les coefficients s'expliquent par la fonction utilisée pour centrer et réduire qui est différente dans l'une et l'autre des implémentations.

Test des méthodes ModChoBayesReg du package Bayess et ModChoBayesReg2 version modifiée

```r
ModChoBayesReg(y.cat,scale(X.cat))
```

```
##
## Number of variables less than 15
## Model posterior probabilities are calculated exactly
##
##    Top10Models PostProb
## 1        1 2 7   0.0767
## 2          1 7   0.0689
## 3      1 2 3 7   0.0686
## 4        1 3 7   0.0376
## 5        1 2 6   0.0369
## 6    1 2 3 5 7   0.0326
## 7      1 2 5 7   0.0294
## 8          1 6   0.0205
## 9      1 2 4 7   0.0201
## 10           7   0.0198
```

```
## $top10models
##  [1] "1 2 7"     "1 7"       "1 2 3 7"   "1 3 7"     "1 2 6"
##  [6] "1 2 3 5 7" "1 2 5 7"   "1 6"       "1 2 4 7"   "7"
##
## $postprobtop10
##  [1] 0.07670048 0.06894313 0.06855427 0.03759751 0.03688912 0.03262797
##  [7] 0.02941759 0.02050185 0.02006371 0.01979095
```

```r
ModChoBayesReg2(y.cat,scale(X.cat),bCalc=FALSE)
```

```
##
## bCalc + false
## Model posterior probabilities are calculated by Gibbs
##
##    Top10Models PostProb
## 1        1 2 7   0.0726
## 2          1 7   0.0663
## 3      1 2 3 7   0.0662
## 4        1 2 6   0.0391
## 5        1 3 7   0.0364
## 6    1 2 3 5 7   0.0312
## 7      1 2 5 7   0.0286
## 8            7   0.0208
## 9          1 6   0.0202
## 10     1 2 4 7   0.0197
```

```
## $top10models
##  [1] "1 2 7"     "1 7"       "1 2 3 7"   "1 2 6"     "1 3 7"
##  [6] "1 2 3 5 7" "1 2 5 7"   "7"         "1 6"       "1 2 4 7"
##
## $postprobtop10
##  [1] 0.0725625 0.0663000 0.0662375 0.0391250 0.0363500 0.0312000 0.0286375
##  [8] 0.0207750 0.0202375 0.0196875
```

La version par Gibbs utilsée ici, renvoi des résultats proches du calcul exact avec ModChoBayesReg.

```r
ModChoBayesReg2(y.cat,scale(X.cat),bCalc=TRUE)
```

```
##
## bCalc = TRUE
## Model posterior probabilities are calculated exactly
##
##    Top10Models PostProb
## 1        1 2 7 -24.3915
## 2          1 7 -24.4378
## 3      1 2 3 7 -24.4402
## 4        1 3 7 -24.7011
## 5        1 2 6 -24.7094
## 6    1 2 3 5 7 -24.7627
## 7      1 2 5 7 -24.8076
## 8          1 6 -24.9645
## 9      1 2 4 7 -24.9738
## 10           7 -24.9798
```

```
## $top10models
##  [1] "1 2 7"     "1 7"       "1 2 3 7"   "1 3 7"     "1 2 6"
##  [6] "1 2 3 5 7" "1 2 5 7"   "1 6"       "1 2 4 7"   "7"
##
## $postprobtop10
##  [1] -24.39145 -24.43776 -24.44021 -24.70109 -24.70935 -24.76266 -24.80764
##  [8] -24.96446 -24.97384 -24.97978
```

On retrouve le même classement qu'avec ModChoBayesReg. Mais la PostProb devra être modifiée.

Test anova des modèles linéaire du cas général

- On considère les 2 modèles suivants :

taux_reussite_attendu_serie_l + taux_acces_attendu_premiere_bac + taux_acces_brut_seconde_bac + taux_acces_brut_premiere_bac

```r
reg.mod2 = lm(Barre ~ Tx_Suc.att_l + Tx_Acc.att_bac1e + Tx_Acc.brt_bac2e + Tx_Acc.brt_bac1e, data=d.reg)
summary(reg.mod2)
```

```
##
## Call:
## lm(formula = Barre ~ Tx_Suc.att_l + Tx_Acc.att_bac1e + Tx_Acc.brt_bac2e +
##     Tx_Acc.brt_bac1e, data = d.reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.46829 -0.50180 -0.04623  0.42214  2.70901
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.265726   0.040902 128.741  < 2e-16 ***
## Tx_Suc.att_l     -0.073523   0.076821  -0.957  0.33899
## Tx_Acc.att_bac1e  0.384969   0.102035   3.773  0.00018 ***
## Tx_Acc.brt_bac2e  0.001383   0.074624   0.019  0.98523
## Tx_Acc.brt_bac1e -0.091302   0.093411  -0.977  0.32882
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9291 on 511 degrees of freedom
## Multiple R-squared:  0.07067,    Adjusted R-squared:  0.0634
## F-statistic: 9.715 on 4 and 511 DF,  p-value: 1.404e-07
```

taux_reussite_attendu_serie_l + taux_acces_attendu_premiere_bac + taux_acces_brut_seconde_bac

```r
reg.mod1 = lm(Barre ~ Tx_Suc.att_l + Tx_Acc.att_bac1e + Tx_Acc.brt_bac2e , data=d.reg)
summary(reg.mod1)
```

```
##
## Call:
## lm(formula = Barre ~ Tx_Suc.att_l + Tx_Acc.att_bac1e + Tx_Acc.brt_bac2e,
##     data = d.reg)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.48751 -0.51692 -0.05349  0.41624  2.61157
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       5.26573    0.04090 128.746  < 2e-16 ***
## Tx_Suc.att_l     -0.05384    0.07413  -0.726    0.468
## Tx_Acc.att_bac1e  0.32056    0.07790   4.115 4.51e-05 ***
## Tx_Acc.brt_bac2e -0.03861    0.06241  -0.619    0.536
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9291 on 512 degrees of freedom
## Multiple R-squared:  0.06893,    Adjusted R-squared:  0.06348
## F-statistic: 12.64 on 3 and 512 DF,  p-value: 5.577e-08
```

- On réalise maintenant des tests entre modèles emboîtés :

```r
anova(reg.mod2,reg.mod1)
```

```
## Analysis of Variance Table
##
## Model 1: Barre ~ Tx_Suc.att_l + Tx_Acc.att_bac1e + Tx_Acc.brt_bac2e +
##     Tx_Acc.brt_bac1e
## Model 2: Barre ~ Tx_Suc.att_l + Tx_Acc.att_bac1e + Tx_Acc.brt_bac2e
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    511 441.12
## 2    512 441.94 -1   -0.8247 0.9553 0.3288
```

Au vu des p-valeurs des tests de Fisher, on peut envisager de se passer de la variable : taux_acces_brut_premiere_bac On conserve le plus petit modèle : reg.mod1

On réalise à nouveaux un test anova, maintenant entre reg.mod1 et step_mod.

```r
anova(step_mod,reg.mod1)
```

```
## Analysis of Variance Table
##
## Model 1: Barre ~ Tx_Acc.att_bac1e
## Model 2: Barre ~ Tx_Suc.att_l + Tx_Acc.att_bac1e + Tx_Acc.brt_bac2e
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    514 442.99
## 2    512 441.94  2    1.0498 0.6081 0.5448
```

Au vu des p-valeurs des tests de Fisher, on peut envisager de se passer des variables : taux_acces_brut_premiere_bac et taux_acces_brut_seconde_bac. On conserve le plus petit modèle : step_mod