Rapport - Statistique bayésienne

Philippe Real

05 mars, 2020

Contents

1 Introduction

1.1 Lecture des données - description statistique

- Rennomage des colonnes

Nouveau Nom	Ancien Nom
Prs_l	effectif_presents_serie_l
prs_es	effectif_presents_serie_es
Prs_s	effectif_presents_serie_s
Eff_2nd	effectif_de_seconde
Eff_1er	effectif_de_premiere
Suc.brt_l	taux_brut_de_reussite_serie_l
Suc.brt_es	taux_brut_de_reussite_serie_es
Suc.brt_s	taux_brut_de_reussite_serie_s
Suc.att_l	taux_reussite_attendu_serie_l
Suc.att_es	taux_reussite_attendu_serie_es
Suc.att_s	taux_reussite_attendu_serie_s
Acc.brt_bac.2	taux_acces_brut_seconde_bac
Acc.brt_bac.1	taux_acces_brut_premiere_bac
Acc.att_bac.1	taux_acces_attendu_premiere_bac)
Acc.att_bac.2	taux_acces_attendu_seconde_bac)
Suc.brt_Tot	taux_brut_de_reussite_total_series)
Suc.att_Tot	taux_reussite_attendu_total_series)

```
##  code_etablissement           ville
##  0950667J: 14       GOUSSAINVILLE: 14
##  0950650R: 12       ARPAJON      : 13
##  0781951X: 10       SARCELLES    : 12
##  0910625K: 10       TAVERNY      : 12
##  0920141D: 10       ARGENTEUIL   : 11
##  0781859X:  9       MAGNANVILLE  : 10
##  (Other) :451       (Other)      :444
##                                          etablissement    commune
##  LYCEE JACQUES PREVERT                          : 16   Min.   :78005
##  LYCEE ROMAIN ROLLAND                           : 14   1st Qu.:91027
##  LYCEE RENE CASSIN                              : 13   Median :92012
##  LYCEE JEAN-JACQUES ROUSSEAU (GENERAL ET TECHNO.): 12   Mean   :89739
##  LYCEE JOLIOT-CURIE                             : 10   3rd Qu.:95018
##  LYCEE LEONARD DE VINCI                         : 10   Max.   :95637
##  (Other)                                        :441
##        Matiere        Barre            Prs_l            Prs_es
##  MATHS     : 59   Min.   :  21.0   Min.   :  6.00   Min.   : 10.00
##  ANGLAIS   : 52   1st Qu.: 111.0   1st Qu.: 18.00   1st Qu.: 53.00
##  HIST. GEO.: 47   Median : 196.0   Median : 30.00   Median : 69.00
##  ESPAGNOL  : 30   Mean   : 321.9   Mean   : 34.24   Mean   : 74.42
##  LET MODERN: 30   3rd Qu.: 292.0   3rd Qu.: 47.00   3rd Qu.: 99.00
##  S. V. T.  : 26   Max.   :2056.0   Max.   :133.00   Max.   :192.00
##  (Other)   :272
##      Prs_s         Suc.brt_l        Suc.brt_es      Suc.brt_s
##  Min.   : 13.0   Min.   : 36.00   Min.   : 51.0   Min.   :50.00
##  1st Qu.: 64.0   1st Qu.: 82.00   1st Qu.: 81.0   1st Qu.:81.00
```

```
##  Median :100.0   Median : 89.00   Median : 88.0   Median :88.00
##  Mean   :106.1   Mean   : 86.35   Mean   : 86.4   Mean   :86.23
##  3rd Qu.:140.0   3rd Qu.: 94.00   3rd Qu.: 94.0   3rd Qu.:93.00
##  Max.   :328.0   Max.   :100.00   Max.   :100.0   Max.   :99.00
##
##    Suc.att_l       Suc.att_es      Suc.att_s        Eff_2nd
##  Min.   :65.00   Min.   :61.00   Min.   :61.00   Min.   : 36.0
##  1st Qu.:84.00   1st Qu.:86.00   1st Qu.:86.00   1st Qu.:268.0
##  Median :89.00   Median :90.00   Median :89.00   Median :336.0
##  Mean   :86.91   Mean   :87.97   Mean   :87.39   Mean   :351.6
##  3rd Qu.:92.00   3rd Qu.:94.00   3rd Qu.:94.00   3rd Qu.:415.0
##  Max.   :98.00   Max.   :98.00   Max.   :98.00   Max.   :764.0
##
##     Eff_1er      Acc.brt_bac.2   Acc.att_bac.2   Acc.brt_bac.1
##  Min.   : 36.0   Min.   :49.00   Min.   :50.00   Min.   :65.00
##  1st Qu.:226.5   1st Qu.:64.00   1st Qu.:64.00   1st Qu.:82.00
##  Median :289.0   Median :71.00   Median :69.00   Median :85.00
##  Mean   :307.7   Mean   :69.61   Mean   :68.47   Mean   :84.53
##  3rd Qu.:364.0   3rd Qu.:76.00   3rd Qu.:73.00   3rd Qu.:89.25
##  Max.   :691.0   Max.   :87.00   Max.   :83.00   Max.   :97.00
##
##  Acc.att_bac.1    Suc.brt_Tot     Suc.att_Tot
##  Min.   :70.00   Min.   :64.00   Min.   :67.0
##  1st Qu.:81.00   1st Qu.:82.00   1st Qu.:84.0
##  Median :85.00   Median :86.00   Median :88.0
##  Mean   :84.19   Mean   :85.46   Mean   :86.8
##  3rd Qu.:89.00   3rd Qu.:91.00   3rd Qu.:92.0
##  Max.   :94.00   Max.   :98.00   Max.   :98.0
##
```

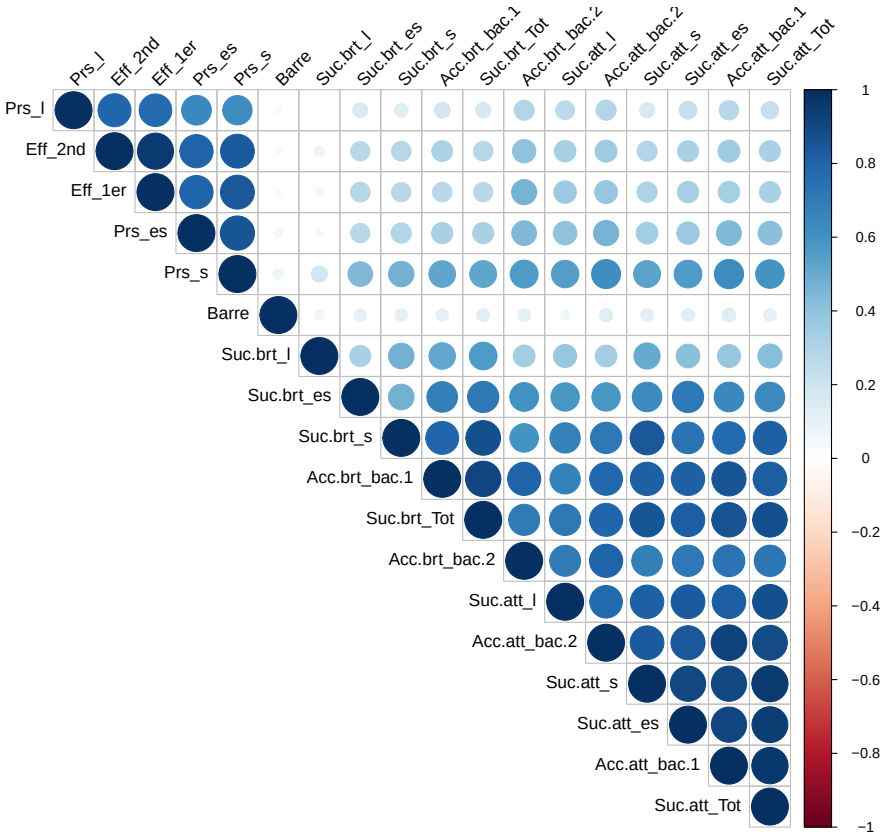
```
## Warning in as.data.frame.integer(length(colnames(data.mutations)),
## colnames(data.mutations)): 'row.names' is not a character vector of length
## 1 -- omitting it. Will be an error!
```

```
## [1] "Barre"
```

```
##   Barre Prs_l Prs_es Prs_s Suc.brt_l Suc.brt_es Suc.brt_s Suc.att_l
## 1 118.0    25     54    97        56         85        80        72
## 2  93.0    25     54    97        56         85        80        72
## 3  38.0    25     54    97        56         85        80        72
## 4 199.0    34     47    47        79         98        85        87
## 5  48.0    34     47    47        79         98        85        87
## 6 256.2    34     47    47        79         98        85        87
##   Suc.att_es Suc.att_s Eff_2nd Eff_1er Acc.brt_bac.2 Acc.att_bac.2
## 1         86        75     304     222            61            64
## 2         86        75     304     222            61            64
## 3         86        75     304     222            61            64
## 4         93        91     194     168            80            69
## 5         93        91     194     168            80            69
## 6         93        91     194     168            80            69
##   Acc.brt_bac.1 Acc.att_bac.1 Suc.brt_Tot Suc.att_Tot
## 1            84            81          81          79
## 2            84            81          81          79
```

```
## 3          84          81          81          79
## 4          92          87          88          89
## 5          92          87          88          89
## 6          92          87          88          89
```

- Corrélations 2 à 2 entre les variables



On a de fortes corrélations entre les groupes de variables. Effectifs (Eff_2nd/Eff_1e) et Effectifs présents (Prs_l/Prs_es/Prs_s) Succés brute (Suc.brt_l/Suc.brt_es/Suc.brt_s) et Succés Attentus (Suc.att_l/Suc.att_es/Suc.att_s) On remarque que le taux de réussite brute série L $Suc.brt_l$ est moins corrélés aux autres variables, et semble avoir une certaine indépendance.

La variable $Acc.brt_bac.2$ est très corrélé avec la variable $Acc.att_bac.2$ et de même pour $Acc.brt_bac.1$ et $Acc.att_bac.1$ On pourrait ne considérer que les variables Accès brute.

les covariables $Suc.brt_Tot$ et $Suc.att_Tot$ sont évidemment fortement corrélés avec les groupes Réussites et Réussites attendus.

La variable a expliquer *Barre* n'est pas corrélés avec les caractéristiques de l'établissement.

On pourrait imaginer, de ne considérer que les variables covaraiables : Effectifs présents: Prs_l/Prs_es/Prs_s Succés brute: Suc.brt_l/Suc.brt_es/Suc.brt_s on garderait aussi Suc.att_l Accès brute: Acc.brt_bac.2/Acc.brt_bac.1

15 taux_acces_attendu_premiere_bac 0.3369
13 taux_acces_attendu_seconde_bac 0.1957
7 taux_reussite_attendu_serie_l 0.1224
17 taux_reussite_attendu_total_series 0.1200
8 taux_reussite_attendu_serie_es 0.1183
16 taux_brut_de_reussite_total_series 0.1161

9 taux_reussite_attendu_serie_s 0.1025
12 taux_acces_brut_seconde_bac 0.0898
5 taux_brut_de_reussite_serie_es 0.0821
6 taux_brut_de_reussite_serie_s 0.0776

Le résultat n'est pas très convaincant, il semble difficile de supprimer des variables.

2 Régression linéaire

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 (input) (y_i) par des covariables $(x_1, ..., x_p)$ 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).

La réponse pour l'individus y_i est donnée par (variable Barre dans notre exemple).

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 demanière 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 :

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

$\sigma^2 \mid X \sim IG(a, b)$

En fixant la matrice M de la manière suivante, 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 suivnate:

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

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

donc :

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

2.2 Résultats et interprétation des coéfficients

2.2.1 Calcul explicite des coefficients

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

$\hat{\beta} = (X^TX)^{-1}X^Ty$

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

```
##                                       [,1]
## effectif_presents_serie_l          16.3770102
## effectif_presents_serie_es         10.0578749
## effectif_presents_serie_s           0.5621583
## taux_brut_de_reussite_serie_l      36.1191826
## taux_brut_de_reussite_serie_es     47.4496652
## taux_brut_de_reussite_serie_s      85.4422916
## taux_reussite_attendu_serie_l    -106.0647897
## taux_reussite_attendu_serie_es     32.3521086
## taux_reussite_attendu_serie_s     -40.3864199
## effectif_de_seconde                 5.8396882
## effectif_de_premiere              -44.5331083
## taux_acces_brut_seconde_bac        97.6265317
## taux_acces_attendu_seconde_bac    -51.1283516
## taux_acces_brut_premiere_bac     -140.2871142
## taux_acces_attendu_premiere_bac   206.2261510
## taux_brut_de_reussite_total_series -39.8727718
## taux_reussite_attendu_total_series -31.4216860
```

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

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

```
##
## Call:
## lm(formula = y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -429.72 -205.90 -122.25   -8.55 1645.96
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        321.9155    18.5937  17.313   <2e-16
## Xeffectif_presents_serie_l          16.3770    34.4842   0.475   0.6351
## Xeffectif_presents_serie_es         10.0579    42.4558   0.237   0.8128
## Xeffectif_presents_serie_s           0.5622    59.0966   0.010   0.9924
## Xtaux_brut_de_reussite_serie_l      36.1192    29.6131   1.220   0.2232
## Xtaux_brut_de_reussite_serie_es     47.4497    41.4726   1.144   0.2531
## Xtaux_brut_de_reussite_serie_s      85.4423    58.1080   1.470   0.1421
## Xtaux_reussite_attendu_serie_l    -106.0648    51.0743  -2.077   0.0383
## Xtaux_reussite_attendu_serie_es     32.3521    70.0697   0.462   0.6445
## Xtaux_reussite_attendu_serie_s     -40.3864    90.0514  -0.448   0.6540
## Xeffectif_de_seconde                5.8397    84.4786   0.069   0.9449
```

```
## Xeffectif_de_premiere                -44.5331    90.8498  -0.490   0.6242
## Xtaux_acces_brut_seconde_bac          97.6265    51.3820   1.900   0.0580
## Xtaux_acces_attendu_seconde_bac      -51.1284    65.2923  -0.783   0.4340
## Xtaux_acces_brut_premiere_bac       -140.2871    73.6707  -1.904   0.0575
## Xtaux_acces_attendu_premiere_bac     206.2262   114.7440   1.797   0.0729
## Xtaux_brut_de_reussite_total_series  -39.8728    95.2695  -0.419   0.6757
## Xtaux_reussite_attendu_total_series  -31.4217   169.9511  -0.185   0.8534
##
## (Intercept)                         ***
## Xeffectif_presents_serie_l
## Xeffectif_presents_serie_es
## Xeffectif_presents_serie_s
## Xtaux_brut_de_reussite_serie_l
## Xtaux_brut_de_reussite_serie_es
## Xtaux_brut_de_reussite_serie_s
## Xtaux_reussite_attendu_serie_l      *
## Xtaux_reussite_attendu_serie_es
## Xtaux_reussite_attendu_serie_s
## Xeffectif_de_seconde
## Xeffectif_de_premiere
## Xtaux_acces_brut_seconde_bac        .
## Xtaux_acces_attendu_seconde_bac
## Xtaux_acces_brut_premiere_bac       .
## Xtaux_acces_attendu_premiere_bac    .
## Xtaux_brut_de_reussite_total_series
## Xtaux_reussite_attendu_total_series
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 422.4 on 498 degrees of freedom
## Multiple R-squared:  0.04068,    Adjusted R-squared:  0.007931
## F-statistic: 1.242 on 17 and 498 DF,  p-value: 0.2267
```

On a "éliminé" l'intercept en centrant ou sinon avec la formule: y~X-1

- 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 $\tilde{\beta} = 0$

```
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                       321.9155039
## effectif_presents_serie_l         8.1885051
## effectif_presents_serie_es        5.0289374
## effectif_presents_serie_s         0.2810791
## taux_brut_de_reussite_serie_l    18.0595913
```

```
## taux_brut_de_reussite_serie_es      23.7248326
## taux_brut_de_reussite_serie_s       42.7211458
## taux_reussite_attendu_serie_l      -53.0323949
## taux_reussite_attendu_serie_es      16.1760543
## taux_reussite_attendu_serie_s      -20.1932099
## effectif_de_seconde                  2.9198441
## effectif_de_premiere               -22.2665542
## taux_acces_brut_seconde_bac         48.8132658
## taux_acces_attendu_seconde_bac     -25.5641758
## taux_acces_brut_premiere_bac       -70.1435571
## taux_acces_attendu_premiere_bac    103.1130755
## taux_brut_de_reussite_total_series -19.9363859
## taux_reussite_attendu_total_series -15.7108430
```

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

```
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                        321.9155039
## effectif_presents_serie_l         16.3453332
## effectif_presents_serie_es        10.0384206
## effectif_presents_serie_s          0.5610709
## taux_brut_de_reussite_serie_l     36.0493196
## taux_brut_de_reussite_serie_es    47.3578863
## taux_brut_de_reussite_serie_s     85.2770261
## taux_reussite_attendu_serie_l   -105.8596354
## taux_reussite_attendu_serie_es    32.2895320
## taux_reussite_attendu_serie_s    -40.3083030
## effectif_de_seconde                5.8283928
## effectif_de_premiere             -44.4469708
## taux_acces_brut_seconde_bac       97.4376989
## taux_acces_attendu_seconde_bac   -51.0294573
## taux_acces_brut_premiere_bac    -140.0157659
## taux_acces_attendu_premiere_bac  205.8272610
## taux_brut_de_reussite_total_series -39.7956484
## taux_reussite_attendu_total_series -31.3609090
```

C'est cette dernière hypothèse que l'on conserve.

2.2.2 Autre Calcul de $\hat{\beta}$

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 renvoyé par la fonction du livre de P. Hoff: A First Course in Bayesian Statistical Methods.

- Bayes Regression : *FonctionBayesReg*

```
##
##            PostMean PostStError Log10bf EvidAgaH0
## Intercept  321.9155     18.3206
## x1          16.3295     33.9119 -1.3062
## x2          10.0287     41.7513 -1.3442
## x3           0.5605     58.1159 -1.3567
## x4          36.0144     29.1217 -1.0238
## x5          47.3120     40.7843 -1.0638
## x6          85.1944     57.1437 -0.8733
## x7        -105.7570     50.2267 -0.3944
## x8          32.2582     68.9069  -1.309
## x9         -40.2692     88.5569 -1.3117
## x10          5.8227     83.0766 -1.3557
## x11        -44.4039     89.3421 -1.3029
## x12         97.3432     50.5293 -0.5506
## x13        -50.9800     64.2088 -1.2194
## x14       -139.8800     72.4481  -0.547
## x15        205.6277    112.8398 -0.6352
## x16        -39.7571     93.6884 -1.3175
## x17        -31.3305    167.1307 -1.3491
##
##
## Posterior Mean of Sigma2: 173193.2688
## Posterior StError of Sigma2: 245171.3446

## $postmeancoeff
##  [1]  321.915504   16.329487   10.028689    0.560527   36.014371
##  [6]   47.311975   85.194353 -105.757008   32.258228  -40.269226
## [11]    5.822742  -44.403881   97.343237  -50.979986 -139.880026
## [16]  205.627719  -39.757068  -31.330506
##
## $postsqrtcoeff
##                                          effectif_presents_serie_l
##                            18.32064                           33.91195
##         effectif_presents_serie_es          effectif_presents_serie_s
##                            41.75126                           58.11585
##      taux_brut_de_reussite_serie_l     taux_brut_de_reussite_serie_es
##                            29.12169                           40.78434
##      taux_brut_de_reussite_serie_s      taux_reussite_attendu_serie_l
##                            57.14370                           50.22669
##     taux_reussite_attendu_serie_es      taux_reussite_attendu_serie_s
##                            68.90688                           88.55694
##                effectif_de_seconde               effectif_de_premiere
##                            83.07664                           89.34214
##        taux_acces_brut_seconde_bac     taux_acces_attendu_seconde_bac
##                            50.52925                           64.20878
##       taux_acces_brut_premiere_bac    taux_acces_attendu_premiere_bac
##                            72.44811                          112.83976
## taux_brut_de_reussite_total_series taux_reussite_attendu_total_series
##                            93.68842                          167.13066
##
## $log10bf
```

```
##  [1] -1.3062110 -1.3441685 -1.3567250 -1.0238434 -1.0637704 -0.8732526
##  [7] -0.3944166 -1.3089805 -1.3116783 -1.3556744 -1.3029098 -0.5506177
## [13] -1.2194081 -0.5470375 -0.6351706 -1.3174966 -1.3490849
##
## $postmeansigma2
## [1] 173193.3
##
## $postvarsigma2
## [1] 60108988208
```

Les Log10 bayes factors sont tous négatifs, aucunes des variables ne se dégage véritablement.

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

Choisir les covariables significatives. Comparer au résultat obtenu par une analyse fréquentiste. Afin de réduire le coût computationnel, il peut être intéressant d'éffectuer une présélection des covariables considérées.

2.3.1 Choix des covariables avec les Bayes factors

Bayes Factors et comparaison de modèles Pour comparer les modèles on peut utiliser les facteurs de Bayes

- Test d'hypothèse $H_0 : \beta_i = 0i$

On test l'hypothèse H_0, $\forall i = 1, ..., 17$ et on calcul le Bayes Factor à partir de la formule du cours (TP4)

- A partir de la fonction $CalcBayesFactor$ pour $g = n$

```
##                            colnames(X) bfactor
## 7       taux_reussite_attendu_serie_l  1.9596
## 14       taux_acces_brut_premiere_bac  1.8639
## 12        taux_acces_brut_seconde_bac  1.8617
## 15    taux_acces_attendu_premiere_bac  1.8086
## 6       taux_brut_de_reussite_serie_s  1.6594
## 4       taux_brut_de_reussite_serie_l  1.5651
## 5      taux_brut_de_reussite_serie_es  1.5401
## 13     taux_acces_attendu_seconde_bac  1.4427
## 11               effectif_de_premiere  1.3904
## 1           effectif_presents_serie_l  1.3884
## 8      taux_reussite_attendu_serie_es  1.3866
## 9       taux_reussite_attendu_serie_s  1.3849
## 16 taux_brut_de_reussite_total_series  1.3813
## 2          effectif_presents_serie_es  1.3646
## 17 taux_reussite_attendu_total_series  1.3615
## 10                effectif_de_seconde  1.3574
## 3           effectif_presents_serie_s  1.3568
```

- A partir de la fonction $BayesReg2$ pour $g = n$

```
##
##            PostMean PostStError Log10bf EvidAgaH0
## Intercept  321.9155     18.3206
## x1          16.3295     33.9119 -1.3062
## x2          10.0287     41.7513 -1.3442
## x3           0.5605     58.1159 -1.3567
## x4          36.0144     29.1217 -1.0238
## x5          47.3120     40.7843 -1.0638
## x6          85.1944     57.1437 -0.8733
## x7        -105.7570     50.2267 -0.3944
## x8          32.2582     68.9069  -1.309
## x9         -40.2692     88.5569 -1.3117
## x10          5.8227     83.0766 -1.3557
## x11        -44.4039     89.3421 -1.3029
## x12         97.3432     50.5293 -0.5506
## x13        -50.9800     64.2088 -1.2194
## x14       -139.8800     72.4481  -0.547
## x15        205.6277    112.8398 -0.6352
## x16        -39.7571     93.6884 -1.3175
## x17        -31.3305    167.1307 -1.3491
##
##
## Posterior Mean of Sigma2: 173193.2688
## Posterior StError of Sigma2: 245171.3446

## $postmeancoeff
##  [1]  321.915504   16.329487   10.028689    0.560527   36.014371
##  [6]   47.311975   85.194353 -105.757008   32.258228  -40.269226
## [11]    5.822742  -44.403881   97.343237  -50.979986 -139.880026
## [16]  205.627719  -39.757068  -31.330506
##
## $postsqrtcoeff
##                                          effectif_presents_serie_l
##                             18.32064                       33.91195
##         effectif_presents_serie_es          effectif_presents_serie_s
##                             41.75126                       58.11585
##      taux_brut_de_reussite_serie_l     taux_brut_de_reussite_serie_es
##                             29.12169                       40.78434
##      taux_brut_de_reussite_serie_s      taux_reussite_attendu_serie_l
##                             57.14370                       50.22669
##     taux_reussite_attendu_serie_es      taux_reussite_attendu_serie_s
##                             68.90688                       88.55694
##                effectif_de_seconde               effectif_de_premiere
##                             83.07664                       89.34214
##        taux_acces_brut_seconde_bac     taux_acces_attendu_seconde_bac
##                             50.52925                       64.20878
##       taux_acces_brut_premiere_bac    taux_acces_attendu_premiere_bac
##                             72.44811                      112.83976
## taux_brut_de_reussite_total_series taux_reussite_attendu_total_series
##                             93.68842                      167.13066
##
## $log10bf
##  [1] -1.3062110 -1.3441685 -1.3567250 -1.0238434 -1.0637704 -0.8732526
##  [7] -0.3944166 -1.3089805 -1.3116783 -1.3556744 -1.3029098 -0.5506177
```

```
## [13] -1.2194081 -0.5470375 -0.6351706 -1.3174966 -1.3490849
##
## $postmeansigma2
## [1] 173193.3
##
## $postvarsigma2
## [1] 60108988208
```

- A partir de la fonction $CalcBayesFactor$ pour $g = 1$

```
##                            colnames(X) bfactor
## 7       taux_reussite_attendu_serie_l  0.4490
## 14       taux_acces_brut_premiere_bac  0.4016
## 12        taux_acces_brut_seconde_bac  0.4005
## 15    taux_acces_attendu_premiere_bac  0.3742
## 6       taux_brut_de_reussite_serie_s  0.3003
## 4       taux_brut_de_reussite_serie_l  0.2536
## 5      taux_brut_de_reussite_serie_es  0.2412
## 13     taux_acces_attendu_seconde_bac  0.1930
## 11               effectif_de_premiere  0.1672
## 1           effectif_presents_serie_l  0.1661
## 8      taux_reussite_attendu_serie_es  0.1653
## 9       taux_reussite_attendu_serie_s  0.1645
## 16 taux_brut_de_reussite_total_series  0.1627
## 2          effectif_presents_serie_es  0.1544
## 17 taux_reussite_attendu_total_series  0.1529
## 10                effectif_de_seconde  0.1508
## 3           effectif_presents_serie_s  0.1505
```

- A partir de la fonction $BayesReg2$ pour $g = 1$

```
##
##           PostMean PostStError Log10bf EvidAgaH0
## Intercept 321.9155     18.5131
## x1          8.1806     24.2547 -0.1257
## x2          5.0241     29.8616 -0.1443
## x3          0.2808     41.5660 -0.1505
## x4         18.0421     20.8286  0.0129       (*)
## x5         23.7018     29.1700 -0.0067
## x6         42.6797     40.8707   0.087       (*)
## x7        -52.9810     35.9234  0.3226       (*)
## x8         16.1604     49.2840 -0.1271
## x9        -20.1736     63.3382 -0.1284
## x10         2.9170     59.4186   -0.15
## x11       -22.2450     63.8998 -0.1241
## x12        48.7659     36.1398  0.2457       (*)
## x13       -25.5394     45.9238 -0.0831
## x14       -70.0756     51.8168  0.2475       (*)
## x15       103.0131     80.7060  0.2041       (*)
## x16       -19.9171     67.0084 -0.1313
## x17       -15.6956    119.5362 -0.1468
##
##
```

```
## Posterior Mean of Sigma2: 176850.8134
## Posterior StError of Sigma2: 250348.9427

## $postmeancoeff
##  [1] 321.9155039   8.1805667   5.0240621   0.2808066  18.0420832
##  [6]  23.7018323  42.6797293 -52.9809820  16.1603722 -20.1736334
## [11]   2.9170134 -22.2449676  48.7659432 -25.5393923 -70.0755556
## [16] 103.0131113 -19.9170583 -15.6956119
##
## $postsqrtcoeff
##                                             effectif_presents_serie_l
##                             18.51308                             24.25472
##         effectif_presents_serie_es          effectif_presents_serie_s
##                             29.86160                             41.56598
##      taux_brut_de_reussite_serie_l     taux_brut_de_reussite_serie_es
##                             20.82860                             29.17003
##      taux_brut_de_reussite_serie_s      taux_reussite_attendu_serie_l
##                             40.87067                             35.92345
##     taux_reussite_attendu_serie_es      taux_reussite_attendu_serie_s
##                             49.28401                             63.33825
##                effectif_de_seconde               effectif_de_premiere
##                             59.41859                             63.89984
##        taux_acces_brut_seconde_bac     taux_acces_attendu_seconde_bac
##                             36.13985                             45.92380
##       taux_acces_brut_premiere_bac    taux_acces_attendu_premiere_bac
##                             51.81679                             80.70596
## taux_brut_de_reussite_total_series taux_reussite_attendu_total_series
##                             67.00842                            119.53624
##
## $log10bf
##  [1] -0.125719637 -0.144344559 -0.150505049  0.012933199 -0.006683201
##  [6]  0.086951691  0.322646154 -0.127078682 -0.128402522 -0.149989616
## [11] -0.124099616  0.245703916 -0.083115184  0.247466860  0.204077270
## [16] -0.131257564 -0.146756674
##
## $postmeansigma2
## [1] 176850.8
##
## $postvarsigma2
## [1] 62674593126
```

- Conclusion les 7ème (Suc.att_l), 12ème (Acc.brt_bac.2) et 14ème (Acc.brt_bac.1) variables sont les plus significatives.

2.3.2 Choix de modèle : calcul exact

- A partir de la méthode vue en TP, on va considérer les 4 variables les plus significatives

```
## [1] 0.000 0.138 0.652 0.032 0.015 0.007 0.147 0.010
```

c'est le modèle (F,T,F) qui est de loin le plus probable a posteriori. Le modèle avec la covariable: taux_reussite_attendu_serie_l (Suc.att_l)

- 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. 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 -2050.608
## 2           13 -2050.789
## 3           16 -2051.045
## 4            8 -2051.065
## 5            9 -2051.135
## 6           17 -2051.175
## 7         7 15 -2051.256
## 8           12 -2051.283
## 9           14 -2051.317
## 10           6 -2051.371
```

```
## $top10models
##  [1] "15"   "13"   "16"   "8"    "9"    "17"   "7 15" "12"   "14"   "6"
##
## $postprobtop10
##  [1] -2050.608 -2050.789 -2051.045 -2051.065 -2051.135 -2051.175 -2051.256
##  [8] -2051.283 -2051.317 -2051.371
```

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   0.1241
## 2           13   0.0729
## 3           16   0.0476
## 4            8   0.0412
## 5            9   0.0409
## 6         7 15   0.0328
## 7           17   0.0309
## 8           14   0.0283
## 9           12   0.0258
## 10           6   0.0239
```

```
## $top10models
##  [1] "15"   "13"   "16"   "8"    "9"    "7 15" "17"   "14"   "12"   "6"
##
```

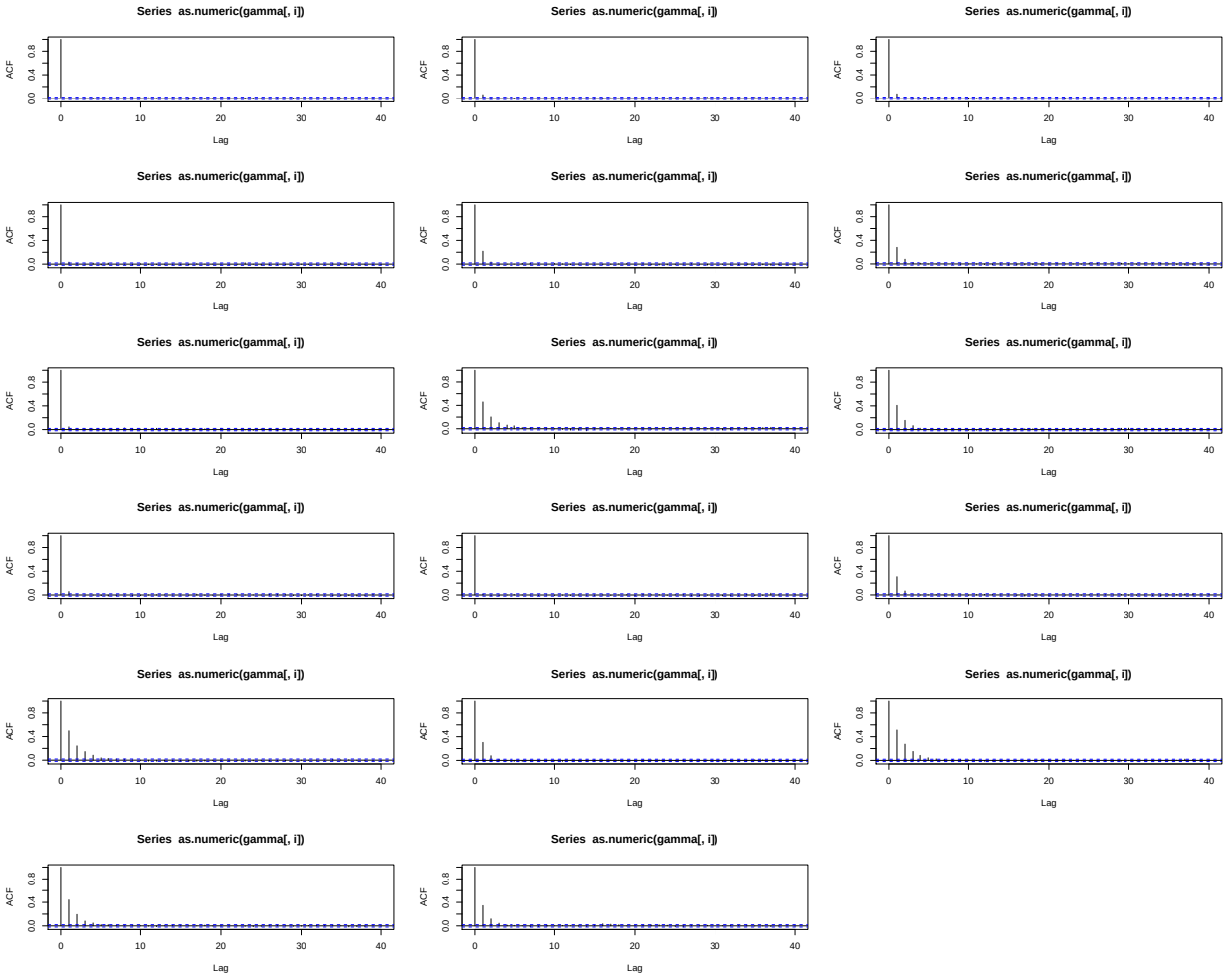
```
## $postprobtop10
##  [1] 0.1241375 0.0729375 0.0476250 0.0412250 0.0408625 0.0327625 0.0309250
##  [8] 0.0282625 0.0258125 0.0239000
```

- Méthode N°2 - A partir de la méthode vue en TP

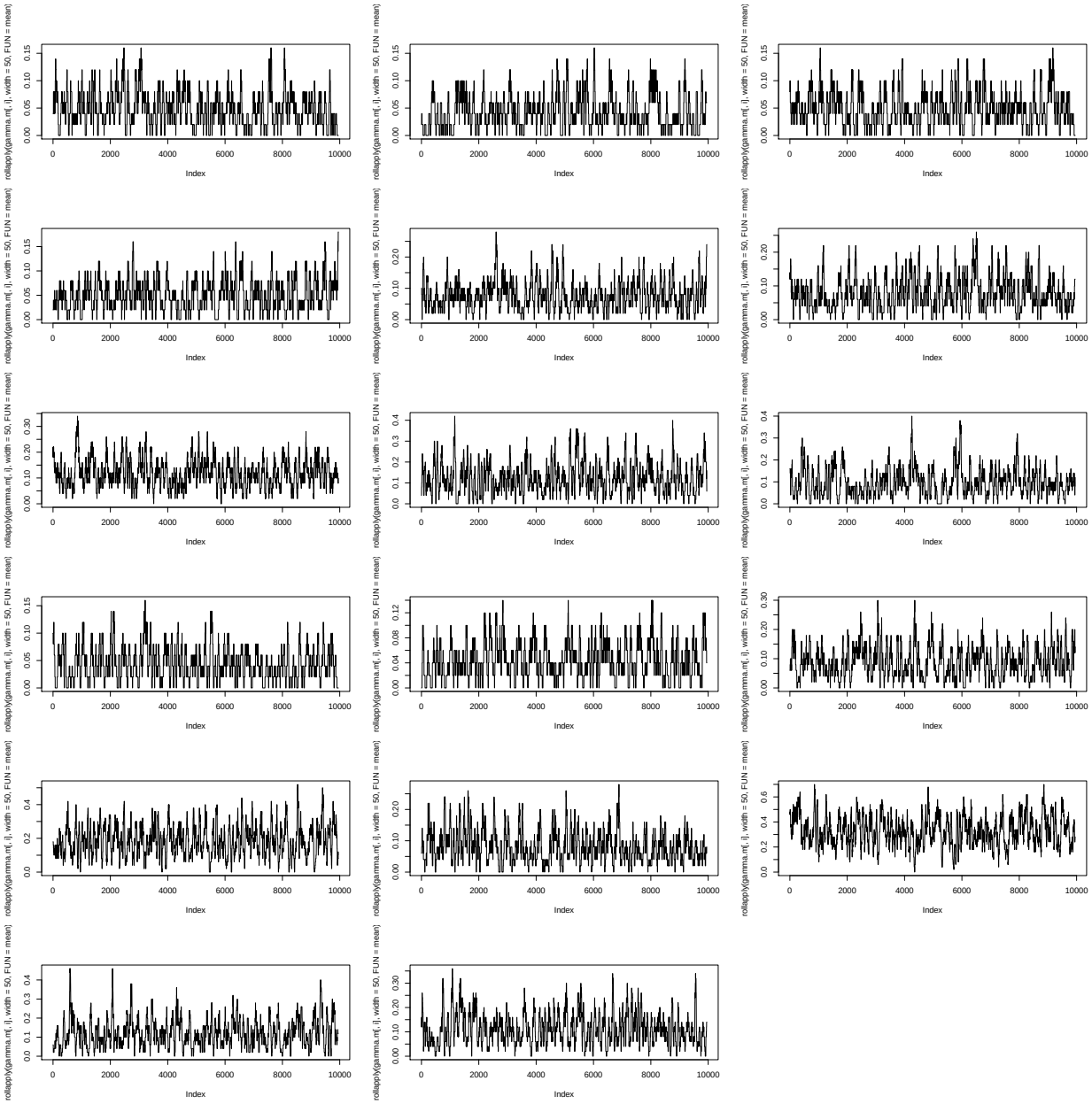
On va maintenant utiliser la fonction implémentée en TP, et comparer les résultats.

```
##                                     X gamma.mean
## 15    taux_acces_attendu_premiere_bac     0.3296
## 13     taux_acces_attendu_seconde_bac     0.1878
## 8      taux_reussite_attendu_serie_es     0.1248
## 16 taux_brut_de_reussite_total_series     0.1224
## 7       taux_reussite_attendu_serie_l     0.1198
## 17 taux_reussite_attendu_total_series     0.1141
## 9       taux_reussite_attendu_serie_s     0.0992
## 12        taux_acces_brut_seconde_bac     0.0893
## 14       taux_acces_brut_premiere_bac     0.0838
## 6       taux_brut_de_reussite_serie_s     0.0805
## 5      taux_brut_de_reussite_serie_es     0.0793
## 4       taux_brut_de_reussite_serie_l     0.0514
## 3           effectif_presents_serie_s     0.0500
## 1           effectif_presents_serie_l     0.0491
## 11               effectif_de_premiere     0.0448
## 10                effectif_de_seconde     0.0444
## 2          effectif_presents_serie_es     0.0432
```

On regarde maintenant, la convergence de la méthode :

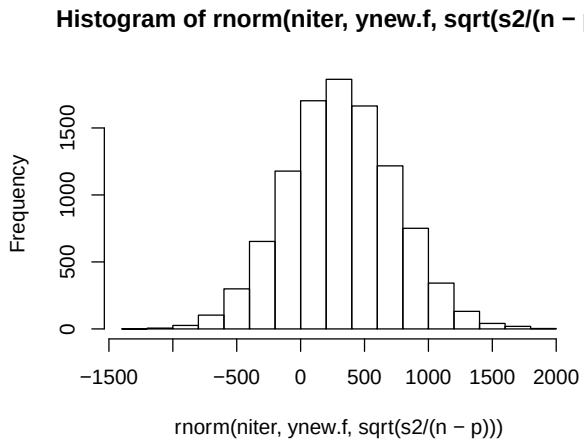


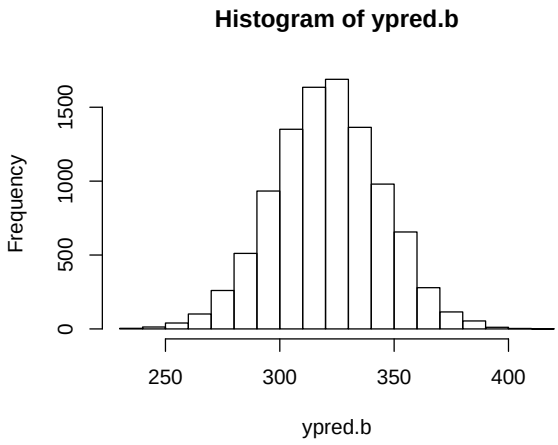
- Vérifions la convergence + le mélange à l'aide de la trace (on utilise une moyenne glissante puisque les valeurs sont binaires).




```
##  probtop20          1          2          3          4          5
## 0.02825263 0.00000000 0.05000000 0.00000000 0.05000000 0.10000000
##          6          7          8          9         10         11
## 0.05000000 0.20000000 0.15000000 0.05000000 0.05000000 0.00000000
##         12         13         14         15         16         17
## 0.05000000 0.15000000 0.05000000 0.35000000 0.05000000 0.15000000
```

- Prédiction





2.3.4 Comparaison au résultat obtenu par une analyse fréquentiste

- Analyse fréquentiste

On considère un modéle de régression linéaire gaussiennne 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.

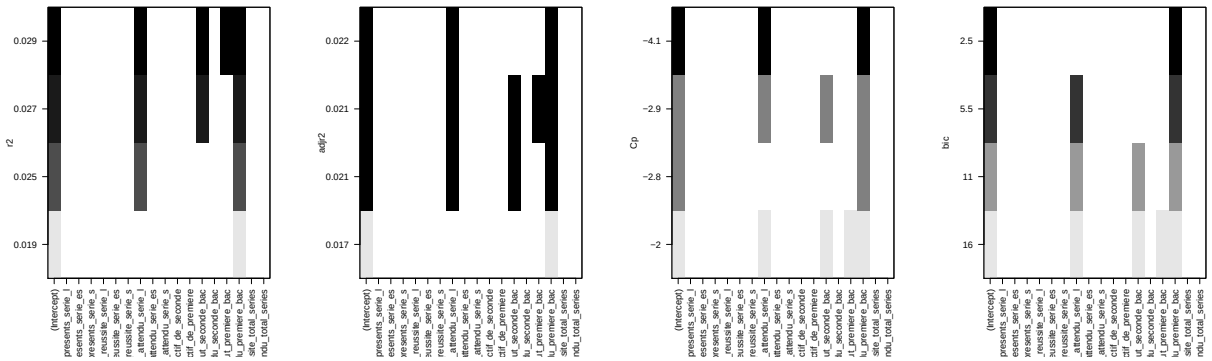
Ainsi les y_i suivent des lois normales indépendantes avec :

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

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

```
##
## Call:
## lm(formula = Barre ~ ., data = data.mutations)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -429.72 -205.90 -122.25   -8.55 1645.96
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -4.725e+02  5.586e+02  -0.846   0.3980
## effectif_presents_serie_l        7.781e-01  1.638e+00   0.475   0.6351
## effectif_presents_serie_es       2.924e-01  1.234e+00   0.237   0.8128
## effectif_presents_serie_s        9.694e-03  1.019e+00   0.010   0.9924
## taux_brut_de_reussite_serie_l    3.122e+00  2.559e+00   1.220   0.2232
## taux_brut_de_reussite_serie_es   4.811e+00  4.205e+00   1.144   0.2531
## taux_brut_de_reussite_serie_s    9.385e+00  6.383e+00   1.470   0.1421
## taux_reussite_attendu_serie_l   -1.428e+01  6.879e+00  -2.077   0.0383
## taux_reussite_attendu_serie_es   3.814e+00  8.261e+00   0.462   0.6445
## taux_reussite_attendu_serie_s   -4.299e+00  9.586e+00  -0.448   0.6540
```

```
## effectif_de_seconde                 4.306e-02  6.229e-01   0.069   0.9449
## effectif_de_premiere               -3.521e-01  7.182e-01  -0.490   0.6242
## taux_acces_brut_seconde_bac         1.074e+01  5.655e+00   1.900   0.0580
## taux_acces_attendu_seconde_bac     -7.077e+00  9.038e+00  -0.783   0.4340
## taux_acces_brut_premiere_bac       -2.039e+01  1.071e+01  -1.904   0.0575
## taux_acces_attendu_premiere_bac     3.444e+01  1.916e+01   1.797   0.0729
## taux_brut_de_reussite_total_series -5.392e+00  1.288e+01  -0.419   0.6757
## taux_reussite_attendu_total_series -4.072e+00  2.202e+01  -0.185   0.8534
##
## (Intercept)
## effectif_presents_serie_l
## effectif_presents_serie_es
## effectif_presents_serie_s
## taux_brut_de_reussite_serie_l
## taux_brut_de_reussite_serie_es
## taux_brut_de_reussite_serie_s
## taux_reussite_attendu_serie_l      *
## taux_reussite_attendu_serie_es
## taux_reussite_attendu_serie_s
## effectif_de_seconde
## effectif_de_premiere
## taux_acces_brut_seconde_bac        .
## taux_acces_attendu_seconde_bac
## taux_acces_brut_premiere_bac       .
## taux_acces_attendu_premiere_bac    .
## taux_brut_de_reussite_total_series
## taux_reussite_attendu_total_series
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 422.4 on 498 degrees of freedom
## Multiple R-squared:  0.04068,    Adjusted R-squared:  0.007931
## F-statistic: 1.242 on 17 and 498 DF,  p-value: 0.2267
```




```
summary(step_mod)
```

```
##
## Call:
## lm(formula = Barre ~ taux_reussite_attendu_serie_l + taux_acces_attendu_premiere_bac,
```

```
##     data = data.mutations)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -387.32 -196.56 -130.83  -14.95 1696.20
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -494.324    260.593  -1.897  0.05840 .
## taux_reussite_attendu_serie_l    -7.882      4.360  -1.808  0.07124 .
## taux_acces_attendu_premiere_bac  17.833      5.407   3.298  0.00104 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 419.5 on 513 degrees of freedom
## Multiple R-squared:  0.02539,    Adjusted R-squared:  0.02159
## F-statistic: 6.681 on 2 and 513 DF,  p-value: 0.001366
```

Les 3 covariables qui se dégagent :

- taux_reussite_attendu_serie_l
- taux_acces_attendu_premiere_bac
- taux_acces_brut_seconde_bac

nettement - “taux_acces_brut_brute_bac”

- 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

```
#reg.mod2 = lm(Barre ~ Suc.att_l + Acc.att_bac.1  + Acc.brt_bac.1 +  Acc.brt_bac.2, data=dataMutations_
reg.mod2 = lm(Barre ~ taux_reussite_attendu_serie_l + taux_acces_attendu_premiere_bac + taux_acces_brut_
summary(reg.mod2)
```

```
##
## Call:
## lm(formula = Barre ~ taux_reussite_attendu_serie_l + taux_acces_attendu_premiere_bac +
##     taux_acces_brut_seconde_bac + taux_acces_brut_premiere_bac,
##     data = dataMutations_d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -410.82 -203.23 -128.06   -4.57 1670.03
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -356.286    279.244  -1.276  0.20257
## taux_reussite_attendu_serie_l   -10.207      4.671  -2.185  0.02934 *
## taux_acces_attendu_premiere_bac  20.776      7.694   2.700  0.00716 **
## taux_acces_brut_seconde_bac       4.986      3.708   1.345  0.17930
```

```
## taux_acces_brut_premiere_bac      -6.280      6.129  -1.025  0.30600
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 419.5 on 511 degrees of freedom
## Multiple R-squared:  0.02905,    Adjusted R-squared:  0.02145
## F-statistic: 3.822 on 4 and 511 DF,  p-value: 0.004514
```

taux_reussite_attendu_serie_l + taux_acces_attendu_premiere_bac + taux_acces_brut_seconde_bac

```
reg.mod1 = lm(Barre ~ taux_reussite_attendu_serie_l
              + taux_acces_attendu_premiere_bac
              +  taux_acces_brut_seconde_bac, data=dataMutations_d)
summary(reg.mod1)
```

```
##
## Call:
## lm(formula = Barre ~ taux_reussite_attendu_serie_l + taux_acces_attendu_premiere_bac +
##     taux_acces_brut_seconde_bac, data = dataMutations_d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -379.54 -206.00 -132.06   -2.57 1674.19
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -422.564    271.663  -1.555  0.12045
## taux_reussite_attendu_serie_l    -8.952      4.508  -1.986  0.04759 *
## taux_acces_attendu_premiere_bac  15.685      5.875   2.670  0.00783 **
## taux_acces_brut_seconde_bac       2.903      3.101   0.936  0.34963
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 419.5 on 512 degrees of freedom
## Multiple R-squared:  0.02705,    Adjusted R-squared:  0.02135
## F-statistic: 4.745 on 3 and 512 DF,  p-value: 0.002831
```

- On réalise maintenant des tests entre modèles emboîtés :

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

```
## Analysis of Variance Table
##
## Model 1: Barre ~ taux_reussite_attendu_serie_l + taux_acces_attendu_premiere_bac +
##     taux_acces_brut_seconde_bac + taux_acces_brut_premiere_bac
## Model 2: Barre ~ taux_reussite_attendu_serie_l + taux_acces_attendu_premiere_bac +
##     taux_acces_brut_seconde_bac
##   Res.Df      RSS Df Sum of Sq    F Pr(>F)
## 1    511 89918104
## 2    512 90102863 -1   -184758 1.05  0.306
```

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.

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

```
## Analysis of Variance Table
##
## Model 1: Barre ~ taux_reussite_attendu_serie_l + taux_acces_attendu_premiere_bac
## Model 2: Barre ~ taux_reussite_attendu_serie_l + taux_acces_attendu_premiere_bac +
##     taux_acces_brut_seconde_bac
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1    513 90257096
## 2    512 90102863  1    154234 0.8764 0.3496
```

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

Un estimateur sans biais de σ^2 est donnée par la formule suivante:

$$\hat{\sigma}^2 = \frac{1}{n-p-1}(y - \hat{\alpha}\mathbb{1}_n - X\hat{\beta})^T(y - \hat{\alpha}\mathbb{1}_n - X\hat{\beta}) = \frac{s^2}{n-p-1}$$

on obtient σ^2

```
##          [,1]
## [1,] 181239.1
```

et les estimations par les moindres carrés des coéfficients de régression :

```
##
## Call:
## lm(formula = Barre ~ taux_reussite_attendu_serie_l + taux_acces_attendu_premiere_bac,
##     data = data.mutations)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -387.32 -196.56 -130.83  -14.95 1696.20
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -494.324    260.593  -1.897  0.05840 .
## taux_reussite_attendu_serie_l    -7.882      4.360  -1.808  0.07124 .
## taux_acces_attendu_premiere_bac  17.833      5.407   3.298  0.00104 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 419.5 on 513 degrees of freedom
## Multiple R-squared:  0.02539,    Adjusted R-squared:  0.02159
## F-statistic: 6.681 on 2 and 513 DF,  p-value: 0.001366
```

effectif_presents_serie_l
effectif_presents_serie_es taux_reussite_attendu_serie_l
taux_brut_de_reussite_total_series

2.3.5 Préselection des covariables

2.3.6 Conclusion

2.4 Mutations en mathématiques et anglais

2.4.1 Calcul explicite des coefficients

- G-prior informative de Zellner Hypothèses Zellner G-prior calcul de $\hat{\beta}$ coefficient du modèle linéaire: $\hat{\beta} = (X^T X)^{-1} X^T y$ Calcul de $E^{\pi}(\beta \mid y, X) = \frac{g}{g+1}(\hat{\beta} + \frac{\tilde{\beta}}{g})$
- Mutations - Mathématiques

```
y<-y.math
X<-X.math

#X=scale(X)
g=length(y)
betatilde=rep(0,dim(X)[2])
beta0.lm=mean(y)
beta.lm=(solve(t(X)%*%X)%*%t(X))%*%(y)
betahat=rbind(Intercept=beta0.lm,beta.lm)
#betahat
mbetabayes=g/(g+1)*(beta.lm+betatilde/g)
postmean=rbind(Intercept=beta0.lm,mbetabayes)
postmean
```

```
##                         [,1]
## Intercept      8.610169e+01
## Prs_l         -4.890902e-14
## Prs_es        -1.720550e-13
## Prs_s          3.349395e-13
## Suc.brt_l     -4.531727e-13
## Suc.brt_es    -6.899666e-13
## Suc.brt_s      9.496442e+00
## Suc.att_l      1.310937e-12
## Suc.att_es     1.729338e-12
## Suc.att_s      1.753738e-12
## Eff_2nd       -3.794816e-13
## Eff_1er        3.559005e-13
## Acc.brt_bac.2  4.503888e-13
## Acc.att_bac.2 -1.517490e-12
## Acc.brt_bac.1 -2.140643e-12
## Acc.att_bac.1  3.080402e-12
## Suc.brt_Tot    3.513153e-12
## Suc.att_Tot   -6.225420e-12
```

On pourrait aussi retrouver les coéfficients $\hat{\beta}$ à partir de la fonction lm.
On remarque cependant une différence assez significative entre les deux appproches, bien que l'odre de grandeur des coefficients est comparable.

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

```
## Warning in summary.lm(reg.lm): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = y ~ X)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -6.973e-14 -1.009e-14 -3.688e-15  1.135e-14  1.451e-13
##
## Coefficients:
##                  Estimate Std. Error    t value Pr(>|t|)
## (Intercept)     8.610e+01  4.159e-15  2.070e+16  < 2e-16 ***
## XPrs_l         -6.827e-15  7.736e-15 -8.820e-01  0.38266
## XPrs_es        -2.159e-15  9.892e-15 -2.180e-01  0.82828
## XPrs_s          2.502e-15  1.496e-14  1.670e-01  0.86799
## XSuc.brt_l     -1.630e-14  6.380e-15 -2.555e+00  0.01444 *
## XSuc.brt_es    -1.644e-14  1.028e-14 -1.600e+00  0.11728
## XSuc.brt_s      9.657e+00  1.482e-14  6.517e+14  < 2e-16 ***
## XSuc.att_l     -1.440e-14  1.150e-14 -1.252e+00  0.21775
## XSuc.att_es     4.666e-14  1.672e-14  2.790e+00  0.00797 **
## XSuc.att_s     -2.169e-14  2.259e-14 -9.600e-01  0.34276
## XEff_2nd        2.930e-15  2.007e-14  1.460e-01  0.88464
## XEff_1er        4.728e-15  2.294e-14  2.060e-01  0.83776
## XAcc.brt_bac.2 -3.088e-14  1.507e-14 -2.048e+00  0.04698 *
## XAcc.att_bac.2  2.235e-14  1.647e-14  1.357e+00  0.18229
## XAcc.brt_bac.1  3.000e-14  2.030e-14  1.477e+00  0.14723
## XAcc.att_bac.1  1.093e-14  2.673e-14  4.090e-01  0.68485
## XSuc.brt_Tot    1.198e-14  2.451e-14  4.890e-01  0.62750
## XSuc.att_Tot   -4.326e-14  3.942e-14 -1.097e+00  0.27894
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.195e-14 on 41 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 3.118e+29 on 17 and 41 DF,  p-value: < 2.2e-16
```

- Mutations - Anglais

```
y<-y.en
X<-X.en

#X=scale(X)
g=length(y)
betatilde=rep(0,dim(X)[2])
beta0.lm=mean(y)
beta.lm=(solve(t(X)%*%X)%*%t(X))%*%(y)
betahat=rbind(Intercept=beta0.lm,beta.lm)
```

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

```
##                         [,1]
## Intercept      8.513462e+01
## Prs_l          4.731812e-13
## Prs_es        -2.695412e-13
## Prs_s          9.280627e-13
## Suc.brt_l     -3.790678e-14
## Suc.brt_es    -7.407073e-14
## Suc.brt_s      9.370547e+00
## Suc.att_l     -5.008488e-13
## Suc.att_es    -6.012800e-13
## Suc.att_s      1.187310e-12
## Eff_2nd        8.276315e-13
## Eff_1er       -1.791640e-12
## Acc.brt_bac.2  4.670813e-13
## Acc.att_bac.2  7.738213e-13
## Acc.brt_bac.1 -1.917125e-13
## Acc.att_bac.1 -3.281769e-12
## Suc.brt_Tot   -2.056552e-13
## Suc.att_Tot    2.589861e-12
```

On pourrait aussi retrouver les coéfficients $\hat{\beta}$ à partir de la fonction lm.
On remarque cependant une différence assez significative entre les deux appproches, bien que l'odre de grandeur des coefficients est comparable.

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

```
## Warning in summary.lm(reg.lm): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = y ~ X)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.381e-14 -9.472e-15 -4.810e-16  9.431e-15  6.801e-14
##
## Coefficients:
##                  Estimate Std. Error    t value Pr(>|t|)
## (Intercept)     8.513e+01  3.003e-15  2.835e+16   <2e-16 ***
## XPrs_l         -4.546e-15  7.719e-15 -5.890e-01   0.5598
## XPrs_es         4.722e-15  7.246e-15  6.520e-01   0.5190
## XPrs_s          6.061e-15  1.036e-14  5.850e-01   0.5625
## XSuc.brt_l      8.442e-15  5.255e-15  1.606e+00   0.1174
## XSuc.brt_es     9.616e-15  6.909e-15  1.392e+00   0.1730
## XSuc.brt_s      9.551e+00  1.060e-14  9.006e+14   <2e-16 ***
## XSuc.att_l      1.628e-14  1.080e-14  1.507e+00   0.1411
```

```
## XSuc.att_es     9.921e-15  1.075e-14  9.230e-01   0.3626
## XSuc.att_s     -3.051e-14  1.425e-14 -2.141e+00   0.0395 *
## XEff_2nd       -2.187e-14  1.541e-14 -1.419e+00   0.1650
## XEff_1er        1.497e-14  1.807e-14  8.290e-01   0.4131
## XAcc.brt_bac.2 -1.775e-14  9.620e-15 -1.845e+00   0.0738 .
## XAcc.att_bac.2 -1.336e-15  1.285e-14 -1.040e-01   0.9178
## XAcc.brt_bac.1  2.892e-14  1.254e-14  2.306e+00   0.0273 *
## XAcc.att_bac.1 -4.673e-15  2.410e-14 -1.940e-01   0.8474
## XSuc.brt_Tot   -2.876e-14  1.739e-14 -1.654e+00   0.1073
## XSuc.att_Tot    9.699e-15  3.161e-14  3.070e-01   0.7609
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.166e-14 on 34 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 5.835e+29 on 17 and 34 DF,  p-value: < 2.2e-16
```

2.4.2 Choix des covariables à l'aide des Bayes factor

Bayes Factors et comparaison de modèles 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 $BayesReg2$ pour $g = n$

- Mutations en mathématiques - A partir de la fonction $BayesReg2$ pour $g = n$

```
##
##           PostMean PostStError Log10bf EvidAgaH0
## Intercept  86.1017      0.1652
## x1          0.0000      0.3021 -0.8891
## x2          0.0000      0.3863 -0.8891
## x3          0.0000      0.5842 -0.8891
## x4          0.0000      0.2491 -0.8891
## x5          0.0000      0.4013 -0.8891
## x6          9.4156      0.5787 21.0919    (****)
## x7          0.0000      0.4491 -0.8891
## x8          0.0000      0.6531 -0.8891
## x9          0.0000      0.8822 -0.8891
## x10         0.0000      0.7836 -0.8891
## x11         0.0000      0.8959 -0.8891
## x12         0.0000      0.5886 -0.8891
## x13         0.0000      0.6433 -0.8891
## x14         0.0000      0.7929 -0.8891
## x15         0.0000      1.0439 -0.8891
## x16         0.0000      0.9570 -0.8891
## x17         0.0000      1.5394 -0.8891
##
##
## Posterior Mean of Sigma2: 1.6099
## Posterior StError of Sigma2: 2.2974
```

```
## $postmeancoeff
##  [1]  8.610169e+01  3.318827e-14  2.910524e-13 -4.445481e-13 -3.286075e-14
##  [6]  2.620126e-14  9.415619e+00 -4.366877e-14  7.696621e-14  1.607011e-13
```

```
## [11] -1.021849e-13  3.777349e-14 -9.694467e-14  1.703082e-14  1.484738e-13
## [16]  1.050098e-14 -1.838455e-13 -3.283892e-13
##
## $postsqrtcoeff
##                      Prs_l        Prs_es         Prs_s     Suc.brt_l
##     0.1651880     0.3020953     0.3862564     0.5841860     0.2491424
##    Suc.brt_es     Suc.brt_s     Suc.att_l    Suc.att_es     Suc.att_s
##     0.4013360     0.5786809     0.4490823     0.6530826     0.8822358
##       Eff_2nd       Eff_1er Acc.brt_bac.2 Acc.att_bac.2 Acc.brt_bac.1
##     0.7836040     0.8959226     0.5886290     0.6432968     0.7928656
## Acc.att_bac.1   Suc.brt_Tot   Suc.att_Tot
##     1.0439175     0.9570449     1.5393626
##
## $log10bf
##  [1] -0.8890756 -0.8890756 -0.8890756 -0.8890756 -0.8890756 21.0919120
##  [7] -0.8890756 -0.8890756 -0.8890756 -0.8890756 -0.8890756 -0.8890756
## [13] -0.8890756 -0.8890756 -0.8890756 -0.8890756 -0.8890756
##
## $postmeansigma2
## [1] 1.609937
##
## $postvarsigma2
## [1] 5.278048
```

- Mutations en mathématiques - A partir de la fonction $BayesReg2$ pour $g = 1$

```
##
##           PostMean PostStError Log10bf EvidAgaH0
## Intercept  86.1017      0.9048
## x1          0.0000      1.1799 -0.1505
## x2          0.0000      1.5086 -0.1505
## x3          0.0000      2.2816 -0.1505
## x4          0.0000      0.9731 -0.1505
## x5          0.0000      1.5675 -0.1505
## x6          4.7876      2.2601  0.8203      (**)
## x7          0.0000      1.7540 -0.1505
## x8          0.0000      2.5507 -0.1505
## x9          0.0000      3.4457 -0.1505
## x10         0.0000      3.0605 -0.1505
## x11         0.0000      3.4992 -0.1505
## x12         0.0000      2.2990 -0.1505
## x13         0.0000      2.5125 -0.1505
## x14         0.0000      3.0967 -0.1505
## x15         0.0000      4.0772 -0.1505
## x16         0.0000      3.7379 -0.1505
## x17         0.0000      6.0122 -0.1505
##
##
## Posterior Mean of Sigma2: 48.2981
## Posterior StError of Sigma2: 68.922
```

```
## $postmeancoeff
##  [1]  8.610169e+01  1.687539e-14  1.479927e-13 -2.260414e-13 -1.670886e-14
```

```
##  [6]  1.332268e-14  4.787603e+00 -2.220446e-14  3.913536e-14  8.171241e-14
## [11] -5.195844e-14  1.920686e-14 -4.929390e-14  8.659740e-15  7.549517e-14
## [16]  5.339479e-15 -9.348078e-14 -1.669775e-13
##
## $postsqrtcoeff
##                      Prs_l        Prs_es         Prs_s     Suc.brt_l
##     0.9047719     1.1798837     1.5085890     2.2816360     0.9730674
##    Suc.brt_es     Suc.brt_s     Suc.att_l    Suc.att_es     Suc.att_s
##     1.5674849     2.2601349     1.7539660     2.5507234     3.4457197
##       Eff_2nd       Eff_1er Acc.brt_bac.2 Acc.att_bac.2 Acc.brt_bac.1
##     3.0604967     3.4991755     2.2989892     2.5125034     3.0966693
## Acc.att_bac.1   Suc.brt_Tot   Suc.att_Tot
##     4.0771944     3.7378988     6.0122382
##
## $log10bf
##  [1] -0.1505150 -0.1505150 -0.1505150 -0.1505150 -0.1505150  0.8202562
##  [7] -0.1505150 -0.1505150 -0.1505150 -0.1505150 -0.1505150 -0.1505150
## [13] -0.1505150 -0.1505150 -0.1505150 -0.1505150 -0.1505150
##
## $postmeansigma2
## [1] 48.29812
##
## $postvarsigma2
## [1] 4750.243
```

- Mutations en anglais - A partir de la fonction $BayesReg2$ pour $g = n$

```
##
##           PostMean PostStError Log10bf EvidAgaH0
## Intercept  85.1346      0.1856
## x1          0.0000      0.4680 -0.8621
## x2          0.0000      0.4393 -0.8621
## x3          0.0000      0.6283 -0.8621
## x4          0.0000      0.3186 -0.8621
## x5          0.0000      0.4189 -0.8621
## x6          9.2800      0.6429 17.5057    (****)
## x7          0.0000      0.6551 -0.8621
## x8          0.0000      0.6518 -0.8621
## x9          0.0000      0.8641 -0.8621
## x10         0.0000      0.9345 -0.8621
## x11         0.0000      1.0954 -0.8621
## x12         0.0000      0.5832 -0.8621
## x13         0.0000      0.7794 -0.8621
## x14         0.0000      0.7603 -0.8621
## x15         0.0000      1.4612 -0.8621
## x16         0.0000      1.0542 -0.8621
## x17         0.0000      1.9166 -0.8621
##
##
## Posterior Mean of Sigma2: 1.7913
## Posterior StError of Sigma2: 2.5596
```

```
## $postmeancoeff
```

```
##  [1]  8.513462e+01 -6.448510e-14 -1.039713e-13  1.498843e-13  2.614261e-15
##  [6] -1.115418e-13  9.280008e+00  7.516000e-14  2.039124e-13  1.172060e-13
## [11] -8.539919e-14  1.254845e-13 -1.045704e-14 -1.080561e-13  2.439977e-14
## [16]  1.036990e-13  2.439977e-14 -4.147961e-13
##
## $postsqrtcoeff
##                      Prs_l        Prs_es         Prs_s     Suc.brt_l
##     0.1856030     0.4679936     0.4393209     0.6282758     0.3185938
##    Suc.brt_es     Suc.brt_s     Suc.att_l    Suc.att_es     Suc.att_s
##     0.4188504     0.6429324     0.6550707     0.6517884     0.8641411
##       Eff_2nd       Eff_1er Acc.brt_bac.2 Acc.att_bac.2 Acc.brt_bac.1
##     0.9344800     1.0953708     0.5832310     0.7793598     0.7602870
## Acc.att_bac.1   Suc.brt_Tot   Suc.att_Tot
##     1.4611691     1.0541526     1.9165925
##
## $log10bf
##  [1] -0.8621379 -0.8621379 -0.8621379 -0.8621379 -0.8621379 17.5056625
##  [7] -0.8621379 -0.8621379 -0.8621379 -0.8621379 -0.8621379 -0.8621379
## [13] -0.8621379 -0.8621379 -0.8621379 -0.8621379 -0.8621379
##
## $postmeansigma2
## [1] 1.79132
##
## $postvarsigma2
## [1] 6.551355
```

- Mutations en anglais - A partir de la fonction $BayesReg2$ pour $g = 1$

```
##
##           PostMean PostStError Log10bf EvidAgaH0
## Intercept  85.1346      0.9554
## x1          0.0000      1.7198 -0.1505
## x2          0.0000      1.6145 -0.1505
## x3          0.0000      2.3088 -0.1505
## x4          0.0000      1.1708 -0.1505
## x5          0.0000      1.5392 -0.1505
## x6          4.7292      2.3627  0.7199      (**)
## x7          0.0000      2.4073 -0.1505
## x8          0.0000      2.3952 -0.1505
## x9          0.0000      3.1756 -0.1505
## x10         0.0000      3.4341 -0.1505
## x11         0.0000      4.0254 -0.1505
## x12         0.0000      2.1433 -0.1505
## x13         0.0000      2.8641 -0.1505
## x14         0.0000      2.7940 -0.1505
## x15         0.0000      5.3696 -0.1505
## x16         0.0000      3.8739 -0.1505
## x17         0.0000      7.0433 -0.1505
##
##
## Posterior Mean of Sigma2: 47.47
## Posterior StError of Sigma2: 67.8284

## $postmeancoeff
```

```
##  [1]  8.513462e+01 -3.286260e-14 -5.298539e-14  7.638334e-14  1.332268e-15
##  [6] -5.684342e-14  4.729235e+00  3.830269e-14  1.039169e-13  5.973000e-14
## [11] -4.352074e-14  6.394885e-14 -5.329071e-15 -5.506706e-14  1.243450e-14
## [16]  5.284662e-14  1.243450e-14 -2.113865e-13
##
## $postsqrtcoeff
##                      Prs_l        Prs_es         Prs_s     Suc.brt_l
##     0.9554497     1.7198243     1.6144555     2.3088435     1.1707966
##    Suc.brt_es     Suc.brt_s     Suc.att_l    Suc.att_es     Suc.att_s
##     1.5392285     2.3627050     2.4073121     2.3952498     3.1756225
##       Eff_2nd       Eff_1er Acc.brt_bac.2 Acc.att_bac.2 Acc.brt_bac.1
##     3.4341100     4.0253659     2.1433090     2.8640606     2.7939703
## Acc.att_bac.1   Suc.brt_Tot   Suc.att_Tot
##     5.3696341     3.8738935     7.0432642
##
## $log10bf
##  [1] -0.1505150 -0.1505150 -0.1505150 -0.1505150 -0.1505150  0.7198731
##  [7] -0.1505150 -0.1505150 -0.1505150 -0.1505150 -0.1505150 -0.1505150
## [13] -0.1505150 -0.1505150 -0.1505150 -0.1505150 -0.1505150
##
## $postmeansigma2
## [1] 47.46998
##
## $postvarsigma2
## [1] 4600.689
```

- Conclusion

Critère de choix : Succés brute S $XSuc.brt_s$ comme pour le modèle linéaire.

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

On utilise la fonctionModChoBayesReg 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            6   0.1399
## 2          6 7   0.0212
## 3         6 11   0.0211
## 4         6 17   0.0205
## 5         6 15   0.0197
## 6         6 13   0.0195
## 7          6 9   0.0194
## 8          6 8   0.0186
## 9          3 6   0.0181
## 10        6 14   0.0178
```

```
## $top10models
##  [1] "6"    "6 7"  "6 11" "6 17" "6 15" "6 13" "6 9"  "6 8"  "3 6"  "6 14"
##
## $postprobtop10
##  [1] 0.1398625 0.0212125 0.0211125 0.0205000 0.0196750 0.0195000 0.0194250
##  [8] 0.0186000 0.0180625 0.0177625
```

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   0.1226
## 2          6 9   0.0198
## 3          5 6   0.0195
## 4         6 12   0.0194
## 5         6 17   0.0179
## 6         6 15   0.0177
## 7         6 13   0.0176
## 8          6 8   0.0175
## 9          3 6   0.0174
## 10         4 6   0.0171
```

```
## $top10models
##  [1] "6"    "6 9"  "5 6"  "6 12" "6 17" "6 15" "6 13" "6 8"  "3 6"  "4 6"
##
## $postprobtop10
##  [1] 0.1226125 0.0198000 0.0195250 0.0194500 0.0179375 0.0177125 0.0176375
##  [8] 0.0175125 0.0174375 0.0171125
```

On retrouve la encore la prédominance de la 6ème variable : $Suc.brt_s$ = Réussite brute terminale s.

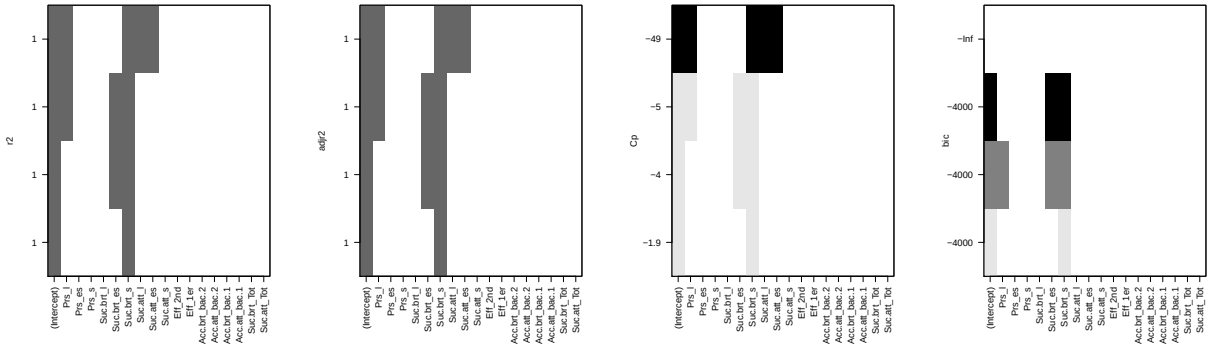
2.4.4 Comparaison au résultat obtenu par une analyse fréquentiste

- Analyse fréquentiste - Mutations en mathématiques

```
## Warning in summary.lm(reg.f1): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = Barre ~ ., data = d.math.reg)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -6.973e-14 -1.009e-14 -3.688e-15  1.135e-14  1.451e-13
```

```
##
## Coefficients:
##                  Estimate Std. Error    t value Pr(>|t|)
## (Intercept)    8.610e+01  4.159e-15  2.070e+16  < 2e-16 ***
## Prs_l         -6.827e-15  7.736e-15 -8.820e-01  0.38266
## Prs_es        -2.159e-15  9.892e-15 -2.180e-01  0.82828
## Prs_s          2.502e-15  1.496e-14  1.670e-01  0.86799
## Suc.brt_l     -1.630e-14  6.380e-15 -2.555e+00  0.01444 *
## Suc.brt_es    -1.644e-14  1.028e-14 -1.600e+00  0.11728
## Suc.brt_s      9.657e+00  1.482e-14  6.517e+14  < 2e-16 ***
## Suc.att_l     -1.440e-14  1.150e-14 -1.252e+00  0.21775
## Suc.att_es     4.666e-14  1.672e-14  2.790e+00  0.00797 **
## Suc.att_s     -2.169e-14  2.259e-14 -9.600e-01  0.34276
## Eff_2nd        2.930e-15  2.007e-14  1.460e-01  0.88464
## Eff_1er        4.728e-15  2.294e-14  2.060e-01  0.83776
## Acc.brt_bac.2 -3.088e-14  1.507e-14 -2.048e+00  0.04698 *
## Acc.att_bac.2  2.235e-14  1.647e-14  1.357e+00  0.18229
## Acc.brt_bac.1  3.000e-14  2.030e-14  1.477e+00  0.14723
## Acc.att_bac.1  1.093e-14  2.673e-14  4.090e-01  0.68485
## Suc.brt_Tot    1.198e-14  2.451e-14  4.890e-01  0.62750
## Suc.att_Tot   -4.326e-14  3.942e-14 -1.097e+00  0.27894
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.195e-14 on 41 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 3.118e+29 on 17 and 41 DF,  p-value: < 2.2e-16
```




```
summary(step_mod)
```

```
## Warning in summary.lm(step_mod): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = Barre ~ Prs_es + Prs_s + Suc.brt_l + Suc.brt_es +
##     Suc.brt_s + Suc.att_es + Acc.brt_bac.2 + Acc.att_bac.2 +
##     Acc.brt_bac.1 + Suc.att_Tot, data = d.math.reg)
##
```

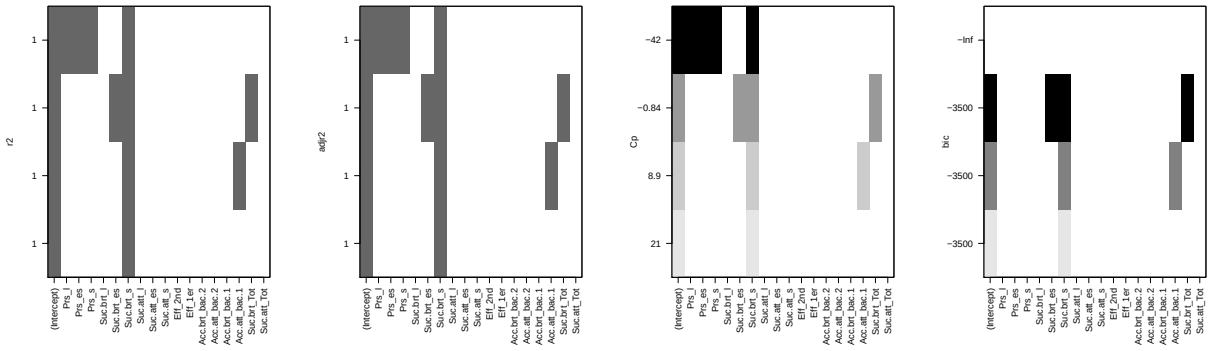
```
## Residuals:
##        Min         1Q     Median         3Q        Max
## -7.245e-14 -1.177e-14 -1.458e-15  8.307e-15  1.522e-13
##
## Coefficients:
##                 Estimate Std. Error    t value Pr(>|t|)
## (Intercept)    8.610e+01  3.917e-15  2.198e+16  < 2e-16 ***
## Prs_es        -1.281e-14  8.705e-15 -1.471e+00 0.147802
## Prs_s          7.699e-15  9.517e-15  8.090e-01 0.422533
## Suc.brt_l     -1.862e-14  5.056e-15 -3.683e+00 0.000585 ***
## Suc.brt_es    -1.533e-14  7.571e-15 -2.025e+00 0.048456 *
## Suc.brt_s      9.657e+00  8.354e-15  1.156e+15  < 2e-16 ***
## Suc.att_es     4.588e-14  1.372e-14  3.345e+00 0.001605 **
## Acc.brt_bac.2 -3.152e-14  1.078e-14 -2.925e+00 0.005247 **
## Acc.att_bac.2  2.271e-14  1.241e-14  1.831e+00 0.073336 .
## Acc.brt_bac.1  4.647e-14  1.195e-14  3.888e+00 0.000310 ***
## Suc.att_Tot   -6.595e-14  1.793e-14 -3.679e+00 0.000591 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.009e-14 on 48 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 5.976e+29 on 10 and 48 DF,  p-value: < 2.2e-16
```

- Analyse fréquentiste - Mutations en Anglais

```
## Warning in summary.lm(reg.f1): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = Barre ~ ., data = d.en.reg)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.381e-14 -9.472e-15 -4.810e-16  9.431e-15  6.801e-14
##
## Coefficients:
##                 Estimate Std. Error    t value Pr(>|t|)
## (Intercept)    8.513e+01  3.003e-15  2.835e+16   <2e-16 ***
## Prs_l         -4.546e-15  7.719e-15 -5.890e-01   0.5598
## Prs_es         4.722e-15  7.246e-15  6.520e-01   0.5190
## Prs_s          6.061e-15  1.036e-14  5.850e-01   0.5625
## Suc.brt_l      8.442e-15  5.255e-15  1.606e+00   0.1174
## Suc.brt_es     9.616e-15  6.909e-15  1.392e+00   0.1730
## Suc.brt_s      9.551e+00  1.060e-14  9.006e+14   <2e-16 ***
## Suc.att_l      1.628e-14  1.080e-14  1.507e+00   0.1411
## Suc.att_es     9.921e-15  1.075e-14  9.230e-01   0.3626
## Suc.att_s     -3.051e-14  1.425e-14 -2.141e+00   0.0395 *
## Eff_2nd       -2.187e-14  1.541e-14 -1.419e+00   0.1650
## Eff_1er        1.497e-14  1.807e-14  8.290e-01   0.4131
## Acc.brt_bac.2 -1.775e-14  9.620e-15 -1.845e+00   0.0738 .
## Acc.att_bac.2 -1.336e-15  1.285e-14 -1.040e-01   0.9178
```

```
## Acc.brt_bac.1  2.892e-14  1.254e-14  2.306e+00   0.0273 *
## Acc.att_bac.1 -4.673e-15  2.410e-14 -1.940e-01   0.8474
## Suc.brt_Tot   -2.876e-14  1.739e-14 -1.654e+00   0.1073
## Suc.att_Tot    9.699e-15  3.161e-14  3.070e-01   0.7609
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.166e-14 on 34 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 5.835e+29 on 17 and 34 DF,  p-value: < 2.2e-16
```




```
summary(step_mod)
```

```
## Warning in summary.lm(step_mod): essentially perfect fit: summary may be
## unreliable

##
## Call:
## lm(formula = Barre ~ Prs_s + Suc.brt_l + Suc.brt_es + Suc.brt_s +
##     Suc.att_l + Suc.att_es + Suc.att_s + Eff_2nd + Acc.brt_bac.2 +
##     Acc.brt_bac.1 + Suc.brt_Tot, data = d.en.reg)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.892e-14 -8.816e-15 -1.148e-15  9.297e-15  7.071e-14
##
## Coefficients:
##                 Estimate Std. Error    t value Pr(>|t|)
## (Intercept)    8.513e+01  2.804e-15  3.036e+16  < 2e-16 ***
## Prs_s          1.187e-14  7.162e-15  1.657e+00  0.10535
## Suc.brt_l      9.788e-15  3.930e-15  2.490e+00  0.01702 *
## Suc.brt_es     1.156e-14  4.849e-15  2.384e+00  0.02195 *
## Suc.brt_s      9.551e+00  8.100e-15  1.179e+15  < 2e-16 ***
## Suc.att_l      1.511e-14  6.277e-15  2.407e+00  0.02081 *
## Suc.att_es     1.177e-14  8.624e-15  1.365e+00  0.17989
## Suc.att_s     -2.374e-14  9.918e-15 -2.394e+00  0.02145 *
## Eff_2nd       -1.371e-14  5.594e-15 -2.451e+00  0.01869 *
## Acc.brt_bac.2 -1.339e-14  5.940e-15 -2.254e+00  0.02977 *
## Acc.brt_bac.1  2.127e-14  9.059e-15  2.348e+00  0.02390 *
```

```
## Suc.brt_Tot   -3.420e-14  1.226e-14 -2.791e+00  0.00802 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.022e-14 on 40 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 1.034e+30 on 11 and 40 DF,  p-value: < 2.2e-16
```

2.5 Conclusion

Pour les mutations en Math et en Anglais, on a plus de difficulté à sélectionner les variables dans le cas fréquentiste, alors que dans le cas bayésien une covariable ressort très nettement.

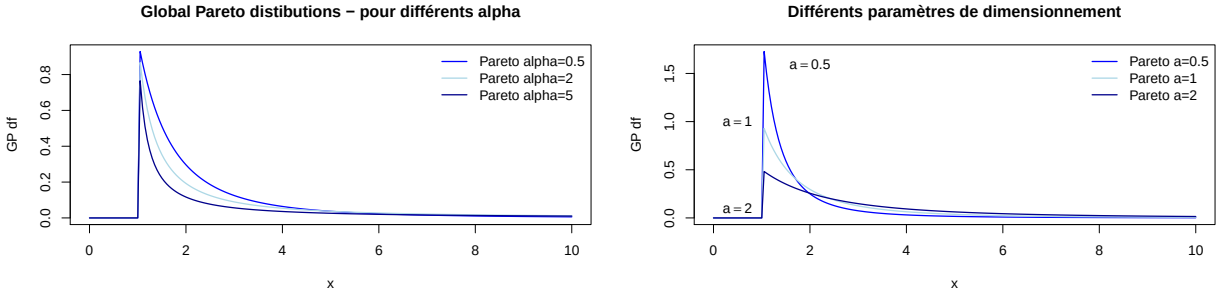
3 Loi de Pareto

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:

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

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

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



3.2 Choix d'une loi à priori pour α

- Loi de paréto :

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

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

Au vu des données on prend : 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
```

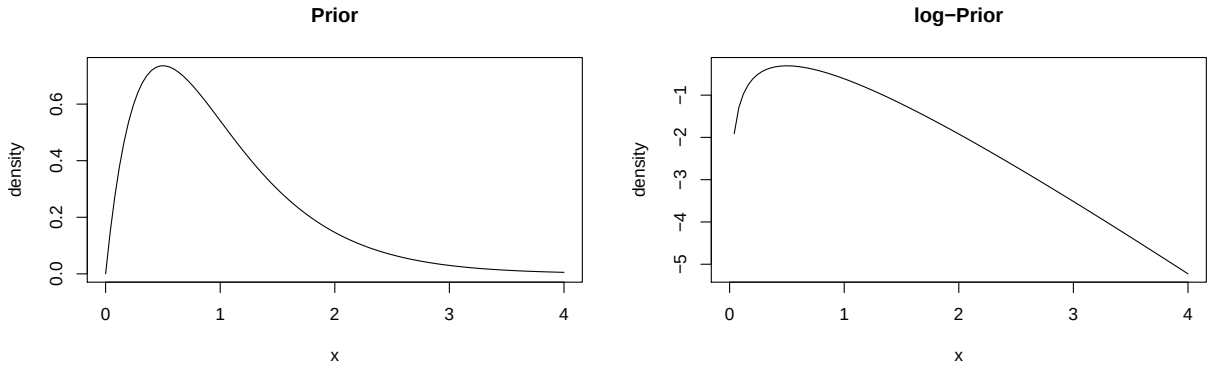
```
## Fitting of the distribution ' exp ' by maximum likelihood
## Parameters:
##       estimate Std. Error
## rate 0.4502063 0.01981913
```

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")
```



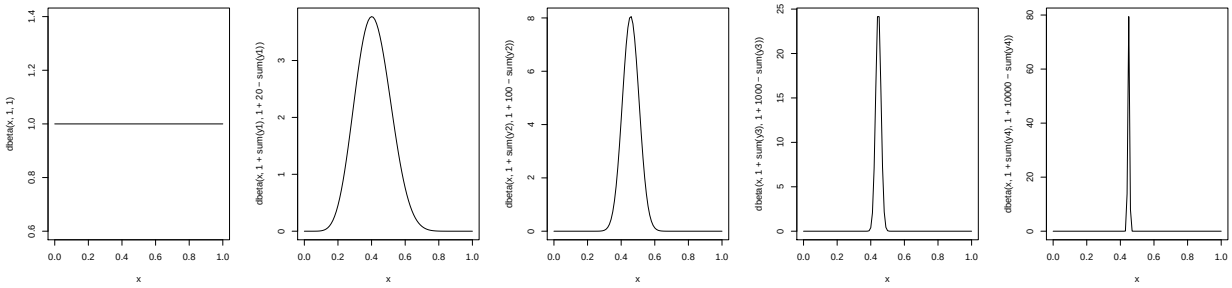
- EMV de *alpha*

Llog = nlog alpha + alpha n log m-(alpha+1)Somme des Xi EMV(alpha) = n/(Somme (log (Zi) + nlog m)

```
m = 21
n=length(y.tot)

EMV_alpha = n/(sum(log(y.tot)) + n*log(m))
EMV_alpha
```

```
## [1] 0.1203333
```



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%.

```
m<-21
MH <- function(Y,alpha0, 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)
}
```

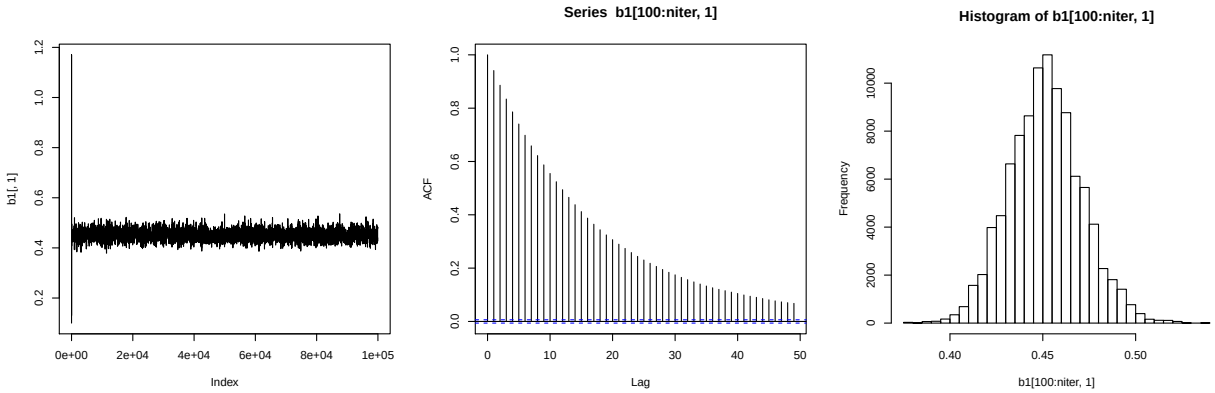
```
niter <- 1e5
b1 <- MH(y.tot, .1, niter)
```

```
# Ã©tudions la sortie de l'algorithme
par(mfcol=c(1,3))
i = 1 # Changer en i=2 pour l'autre paramÃ¨tre
# trace
```

```r
plot(b1[, 1], type="l")
#plot(b2[, i], type="l")
#plot(b3[, i], type="l")

# autocorrÃ©lations
acf(b1[100:niter, 1])
#acf(b2[100:niter, i])
#acf(b3[100:niter, i])

# histogrammes
hist(b1[100:niter, 1], breaks=50)
```




```r
#hist(b2[100:niter, i], breaks=50)
#hist(b3[100:niter, i], breaks=50)
```

Intervalle de confiance à 95% :

```
##      2.5%     97.5%
## 0.4140977 0.4916154
```

```r
# Effective Sample Size
niter/(2*sum(acf(b1[100:niter, 1], plot=F)$acf) - 1)
```

```
## [1] 3085.79
```

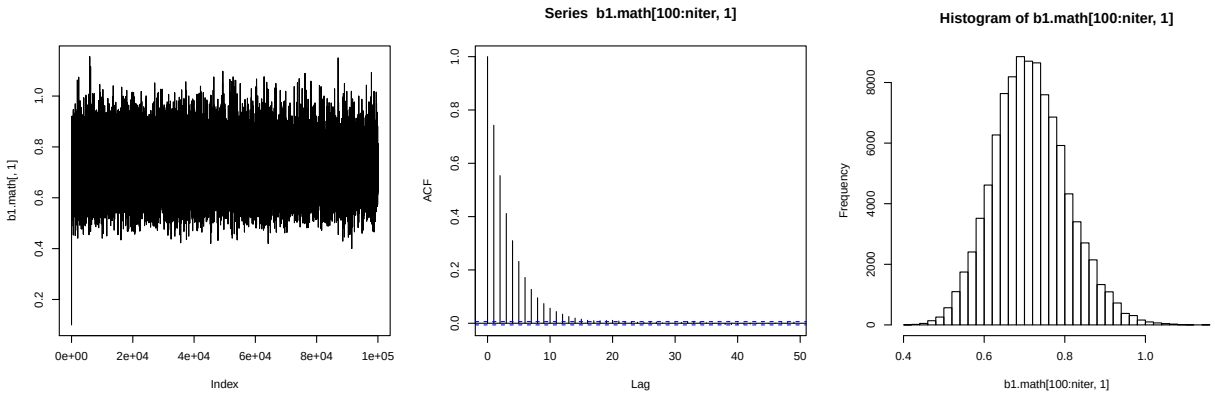
3.5 Analyse pour les mutation en anglais et en math

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

```r
niter <- 1e5
b1.math <- MH(y.math, .1, niter)
b1.en <- MH(y.en, .1, niter)
```

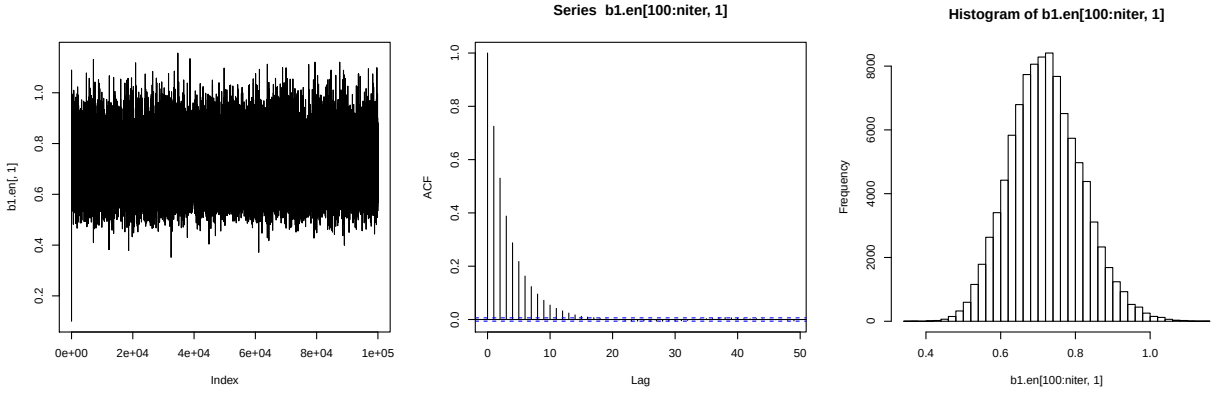
3.5.2 Convergence de l'algorithme de Metropolois-Hastings: mutations en mathématiques

```r
# Etudions la sortie de l'algorithme
par(mfcol=c(1,3))
# trace
plot(b1.math[, 1], type="l")
# autocorrélations
acf(b1.math[100:niter, 1])
# histogrammes
hist(b1.math[100:niter, 1], breaks=50)
```



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

```r
# Etudions la sortie de l'algorithme
par(mfcol=c(1,3))
# trace
plot(b1.en[, 1], type="l")
# autocorrélations
acf(b1.en[100:niter, 1])
# histogrammes
hist(b1.en[100:niter,1], breaks=50)
```



Intervalle de confiance à 95% math et anglais

```r
quantile(b1.math , c(.025,.975))
```

```
##      2.5%     97.5%
## 0.5452695 0.9063543
```

```r
quantile(b1.en , c(.025,.975))
```

```
##      2.5%     97.5%
## 0.5416378 0.9212812
```

4 Annexes

4.1 Test des méthodes BayesReg du package Bayess et BayesReg2 version modifiée

```r
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
```

```r
BayesReg2(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
```

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

- Fonction BayesReg

```
BayesReg(y.cat, 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, 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
```

- Fonction *ModChoBayesReg* pour le choix de modèle

```
ModChoBayesReg(y.cat,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
```

```
ModChoBayesReg2(y.cat,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
```

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

```
##
## bCalc + false
## Model posterior probabilities are calculated by Gibbs
##
##    Top10Models PostProb
## 1        1 2 7   0.0776
## 2          1 7   0.0709
## 3      1 2 3 7   0.0687
## 4        1 3 7   0.0372
## 5    1 2 3 5 7   0.0347
## 6      1 2 5 7   0.0325
## 7        1 2 6   0.0311
## 8      1 2 4 7   0.0217
## 9            7   0.0203
## 10       1 5 7   0.0186

## $top10models
##  [1] "1 2 7"     "1 7"       "1 2 3 7"   "1 3 7"     "1 2 3 5 7"
##  [6] "1 2 5 7"   "1 2 6"     "1 2 4 7"   "7"         "1 5 7"
##
## $postprobtop10
##  [1] 0.0775750 0.0709250 0.0686875 0.0372375 0.0347250 0.0325000 0.0310875
##  [8] 0.0217375 0.0203125 0.0186250
```