# **Mô hình APS45ID ~ TimeSci + TimeSoc**

## Statement of the problem

Bài toán sử dụng cơ sở dữ liệu “STEM”

Đánh giá ảnh hưởng thời gian đọc sách đến kết quả học tập học sinh

## Cài đặt package bayevl

Package bayesvl có thể được cài đặt trực tiếp trên R từ github

|  |
| --- |
| > install.packages("devtools")  > devtools::install\_github("sshpa/bayesvl") |

Nạp package bayesvl

|  |
| --- |
| > library("bayesvl") |

## Dữ liệu và đánh giá mô hình (dataset and estimations)

## Dữ liệu và xây dựng mô hình

|  |
| --- |
| data1<-read.csv("/Statistics/STEM/5000Toan.csv", header = TRUE)  head(data1) |

Các biến sử dụng:

* TimeSci:
* TimeSoc:
* APS45ID: học lực của học sinh

1: điểm trung bình <5

2: điểm trung bình <7

3: điểm trung bình <=8

4: điểm trung bình <=9

5: điểm trung bình <=10

**Mô hình 1:**

Mục đích đánh giá ảnh hưởng thời gian đọc sách đến học lực của học sinh

Fig. 1

Diễn giải mô hình quan hệ:

Khởi tạo mô hình và các biến dữ liệu quan sát (observation data) trong mô hình bằng bayesvl trên R như sau:

|  |
| --- |
| # Design the model  model <- bayesvl()  model <- bvl\_addNode(model, "APS45ID", "norm")  model <- bvl\_addNode(model, "TimeSci", "cat")  model <- bvl\_addNode(model, "TimeSoc", "cat")  model <- bvl\_addArc(model, "TimeSci", "APS45ID", "slope")  model <- bvl\_addArc(model, "TimeSoc", "APS45ID", "slope")  model <- bvl\_modelFix(model, data1)  model\_string <- bvl\_model2Stan(model) |

Các biến đều được định nghĩa ở phân phối categorical (cat), các biến này thể hiện trên mô hình quan hệ Fig. 1 bằng các nodes có màu blue

Nếu vẽ lại sơ đồ lưới này trên R:

|  |
| --- |
| bvl\_bnPlot(model) |

Fig. 2

Như vậy ta đã dựng lại toàn bộ mô hình quan hệ từ hình Fig. 1 thành mô hình hồi quy bayesvl (Fig. 2). Toàn bộ các công thức toán và mô hình hồi quy Stan sẽ được bayesvl tự động tạo theo mô hình trên.

Ta có thể thấy ở phần formula, công thức tổng quát của mô hình là:

APS45ID ~ b\_TimeSci\_APS45ID \* TimeSci + b\_TimeSoc\_APS45ID \* TimeSoc + a\_Sex[Sex]

Cơ sở toán học:

Oi ~ alpha[xvarint] + betaj \* xji

Trong đó Oi là biến outcome, xvarint là biến varying intercept, xj là biến độc lập thứ j

Biến O là biến binomial, do đó nếu viết dưới dạng thống kê ta có phân phối O có dạng:

O ~ nomial(mu, sigma)

Trong đó:

mui = alpha[xvarint] + betaj \* xji

Code stan:

Để tạo code stan tự động cho mô hình như trên, ta chỉ cần gọi lệnh:

|  |
| --- |
| model\_string <- bvl\_model2Stan(model)  cat(model\_string) |

Toàn bộ code stan được tao ra như sau:

|  |
| --- |
| functions{  int numLevels(int[] m) {  int sorted[num\_elements(m)];  int count = 1;  sorted = sort\_asc(m);  for (i in 2:num\_elements(sorted)) {  if (sorted[i] != sorted[i-1])  count = count + 1;  }  return(count);  }  }  data{  // Define variables in data  int<lower=1> Nobs; // Number of observations (an integer)  real APS45ID[Nobs]; // outcome variable  int NSex;  int<lower=1,upper=NSex> Sex[Nobs];  int NTimeSci;  int<lower=1,upper=NTimeSci> TimeSci[Nobs];  int NTimeSoc;  int<lower=1,upper=NTimeSoc> TimeSoc[Nobs];  }  transformed data{  // Define transformed data  }  parameters{  // Define parameters to estimate  real<lower=0> sigma\_APS45ID;  real b\_TimeSci\_APS45ID;  real b\_TimeSoc\_APS45ID;  real a0\_Sex;  real<lower=0> sigma\_Sex;  vector[NSex] u\_Sex;  }  transformed parameters{  // Transform parameters  real mu\_APS45ID[Nobs];  vector[NSex] a\_Sex;  // Varying intercepts definition  for(k in 1:NSex) {  a\_Sex[k] = a0\_Sex + u\_Sex[k];  }  for (i in 1:Nobs) {  mu\_APS45ID[i] = b\_TimeSci\_APS45ID \* TimeSci[i] + b\_TimeSoc\_APS45ID \* TimeSoc[i] + a\_Sex[Sex[i]];  }  }  model{  // Priors  b\_TimeSci\_APS45ID ~ normal( 0, 10 );  b\_TimeSoc\_APS45ID ~ normal( 0, 10 );  a0\_Sex ~ normal(0,10);  sigma\_Sex ~ normal(0,10);  u\_Sex ~ normal(0, sigma\_Sex);  // Likelihoods  APS45ID ~ normal(mu\_APS45ID, sigma\_APS45ID);  }  generated quantities {  // simulate data from the posterior  real yrep\_APS45ID[Nobs];  // log-likelihood posterior  vector[Nobs] log\_lik\_APS45ID;  for (i in 1:num\_elements(yrep\_APS45ID)) {  yrep\_APS45ID[i] = normal\_rng(mu\_APS45ID[i], sigma\_APS45ID);  }  for (i in 1:Nobs) {  log\_lik\_APS45ID[i] = normal\_lpdf(APS45ID[i] | mu\_APS45ID[i], sigma\_APS45ID);  }  } |

## Thực hiện mô phỏng MCMC

Gọi lệnh thực hiện mô phỏng MCMC mô hình bayesvl:

|  |
| --- |
| model <- bvl\_modelFit(model, data1, warmup = 2000, iter = 5000, chains = 4, cores = 4) |

## Kết quả

|  |
| --- |
| > summary(model)  Model Info:  nodes: 3  arcs: 2  scores: NA  formula: APS45ID ~ a\_APS45ID + b\_TimeSci\_APS45ID \* TimeSci + b\_TimeSoc\_APS45ID \* TimeSoc  Estimates:  Inference for Stan model: ab95566b2110616403bedb2108fa40fe.  4 chains, each with iter=5000; warmup=2000; thin=1;  post-warmup draws per chain=3000, total post-warmup draws=12000.  mean se\_mean sd 2.5% 25% 50% 75% 97.5% n\_eff Rhat  a\_APS45ID 2.33 0 0.05 2.23 2.30 2.33 2.36 2.43 7069 1  b\_TimeSci\_APS45ID 0.05 0 0.03 0.00 0.03 0.05 0.07 0.11 6901 1  b\_TimeSoc\_APS45ID 0.11 0 0.03 0.05 0.09 0.11 0.13 0.17 7641 1  Samples were drawn using NUTS(diag\_e) at Thu Mar 12 22:28:40 2020.  For each parameter, n\_eff is a crude measure of effective sample size,  and Rhat is the potential scale reduction factor on split chains (at  convergence, Rhat=1).  elapsed time: 101.460299968719 secs |

Kết quả trên ta có thể thấy mô hình hội tụ tốt, Rhat ~ 1 (lớn hơn 1.1 là có vấn đề) và n\_eff đều trên 2000 (trên 1000 là dấu hiệu tốt)

Nếu vẽ các chuỗi MCMC:

bvl\_plotTrace(model)



Fig. 3

Kiểm tra Gelman shrink factors:

The results give us the median potential scale reduction factor and its 97.5% quantile.

We also get a multivariate potential scale reduction factor that was proposed by Gelman and Brooks.

Đánh giá tổng quan các hệ số hồi quy:

bvl\_plotIntervals (model)



Fig. 4

Phân phối của các hệ số:

bvl\_plotParams(model, 1, 3)



Fig. 5