e(giới thiệu chung mô hình hồi quy tuyến tính)

Bắt đầu với bài toán đơn giản sử dụng bộ dữ liệu mẫu có sẵn trên R

|  |
| --- |
| attach(cars); str(cars) |

Bộ dữ liệu này có 2 biến tốc độ xe speed và khoảng cách dist

Nhiệm vụ của chúng ta là phải tìm quan hệ giữa 2 biến này, ta có công thức đường hồi quy như sau:

Yspeed = adist + bdist \* Xdist

Để tìm các hệ số hồi quy adist và bdist ta thực hiện hồi quy tuyến tính trên R:

lm\_cars <- lm(dist ~ speed)

summary(lm\_cars)$r.squared

|  |
| --- |
| [1] 0.6510794 |

|  |
| --- |
| > summary(lm\_cars)  Call:  lm(formula = dist ~ speed)  Residuals:  Min 1Q Median 3Q Max  -29.069 -9.525 -2.272 9.215 43.201  Coefficients:  Estimate Std. Error t value Pr(>|t|)  (Intercept) -17.5791 6.7584 -2.601 0.0123 \*  speed 3.9324 0.4155 9.464 1.49e-12 \*\*\*  ---  Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1  Residual standard error: 15.38 on 48 degrees of freedom  Multiple R-squared: 0.6511, Adjusted R-squared: 0.6438  F-statistic: 89.57 on 1 and 48 DF, p-value: 1.49e-12 |

plot(dist ~ speed); abline(lm\_cars$coefficients, col = "blue")



Hồi quy tuyến tính Bayesian trên bayesvl

dist ~ Normal(mu, sigma)

mu = alpha + beta \* speed

alpha ~ Normal(0, 10)

beta ~ Normal(0,10)

sigma ~ Normal(0,10)

|  |
| --- |
| # Design the model  model <- bayesvl()  model <- bvl\_addNode(model, "speed", "norm")  model <- bvl\_addNode(model, "dist", "norm")  model <- bvl\_addArc(model, "speed", "dist", "slope") |

Trên mô hình này ta định nghĩa 2 biến là dist và speed. Quan hệ giữa speed và dist là hồi quy tuyến tính

|  |
| --- |
| # Fit the model  fit <- bvl\_modelFit(model, cars, warmup = 2000, iter = 4000, chains = 4, cores = 1) |

|  |
| --- |
| > summary(fit)  Model Info:  nodes: 2  arcs: 1  scores: NA  formula: dist ~ a\_dist + b\_speed\_dist \* speed  Estimates:  Inference for Stan model: dc3b164adc4fe47a3b8332c0bfb6a401.  4 chains, each with iter=20000; warmup=2000; thin=1;  post-warmup draws per chain=18000, total post-warmup draws=72000.  mean se\_mean sd 2.5% 25% 50% 75% 97.5% n\_eff Rhat  a\_dist -17.39 0.04 7.00 -31.08 -22.05 -17.40 -12.74 -3.58 24553 1  b\_speed\_dist 3.92 0.00 0.43 3.07 3.64 3.92 4.21 4.76 24316 1  Samples were drawn using NUTS(diag\_e) at Tue Mar 10 18:00:17 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: 76.9043509960175 secs |





Ví dụ đa biến:

Ở trên là 1 ví dụ đơn giản hồi quy với 1 biến độc lập. Nếu ta có nhiều biến độc lập, lúc này mô hình hồi quy có dạng:

burden[i] = alpha + beta\_res \* res[i] + beta\_insured \* insured[i]

|  |
| --- |
| # Design the model  model <- bayesvl()  model <- bvl\_addNode(model, "burden", "norm")  model <- bvl\_addNode(model, "res", "norm")  model <- bvl\_addNode(model, "insured", "norm")  model <- bvl\_addArc(model, "res", "burden", "slope")  model <- bvl\_addArc(model, "insured", "burden", "slope") |

|  |
| --- |
| # Generate the stan code for model  model\_string <- bvl\_model2Stan(model)  cat(model\_string) |

|  |
| --- |
| 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 burden[Nobs]; // outcome variable  real res[Nobs];  real insured[Nobs];  }  transformed data{  // Define transformed data  }  parameters{  // Define parameters to estimate  real<lower=0> sigma\_burden;  real a\_burden;  real b\_res\_burden;  real b\_insured\_burden;  }  transformed parameters{  // Transform parameters  real mu\_burden[Nobs];  for (i in 1:Nobs) {  mu\_burden[i] = a\_burden + b\_res\_burden \* res[i] + b\_insured\_burden \* insured[i];  }  }  model{  // Priors  a\_burden ~ normal(0,100);  b\_res\_burden ~ normal( 0, 10 );  b\_insured\_burden ~ normal( 0, 10 );  // Likelihoods  burden ~ normal(mu\_burden, sigma\_burden);  }  generated quantities {  // simulate data from the posterior  real yrep\_burden[Nobs];  // log-likelihood posterior  vector[Nobs] log\_lik\_burden;  for (i in 1:num\_elements(yrep\_burden)) {  yrep\_burden[i] = normal\_rng(mu\_burden[i], sigma\_burden);  }  for (i in 1:Nobs) {  log\_lik\_burden[i] = normal\_lpdf(burden[i] | mu\_burden[i], sigma\_burden);  }  } |

|  |
| --- |
| # Fit the model  fit <- bvl\_modelFit(model, data1, warmup = 2000, iter = 20000, chains = 4, cores = 1) |

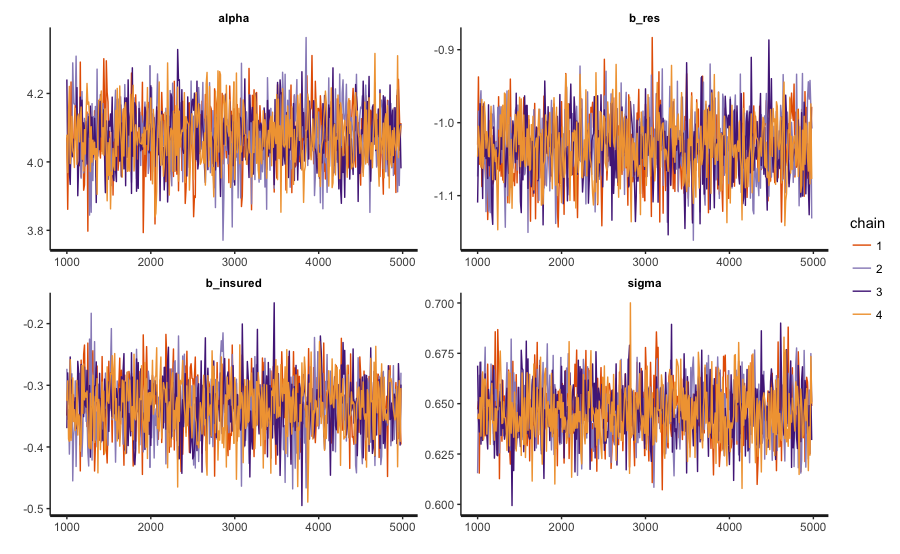
|  |
| --- |
| > summary(model)  4 chains, each with iter=5000; warmup=1000; thin=10;  post-warmup draws per chain=400, total post-warmup draws=1600.  mean se\_mean sd 2.5% 25% 50% 75% 97.5% n\_eff Rhat  alpha 4.08 0 0.09 3.90 4.02 4.08 4.14 4.24 1485 1  b\_res -1.03 0 0.04 -1.12 -1.06 -1.03 -1.01 -0.95 1502 1  b\_insured -0.33 0 0.05 -0.43 -0.37 -0.33 -0.30 -0.24 1610 1  sigma 0.65 0 0.01 0.62 0.64 0.65 0.66 0.67 1763 1 |

Các hệ số hồi quy đều âm, ảnh hưởng burden ngược với res, insured???

burden ~ 4.08 -1.03 \* res -0.33\*insured

res có ảnh hưởng đến finacial burden nhiều hơn insured

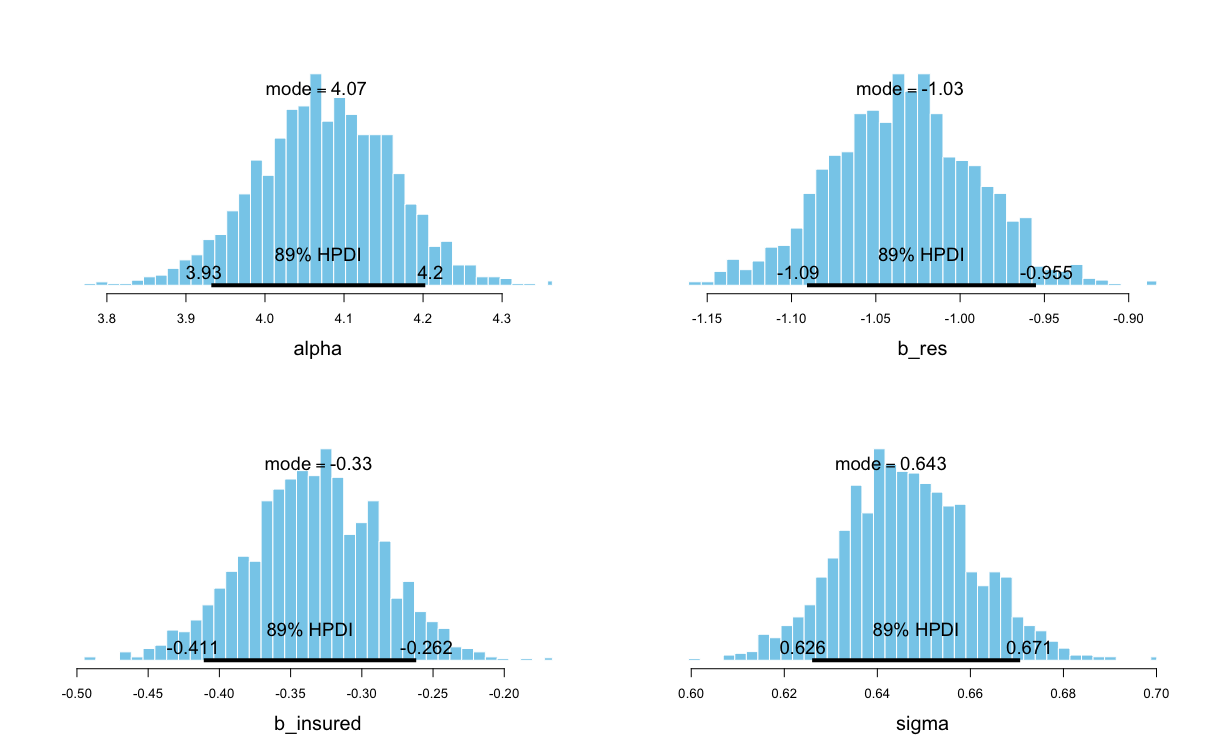
traceplot(fit,pars = c("alpha","b\_res","b\_insured","sigma"))



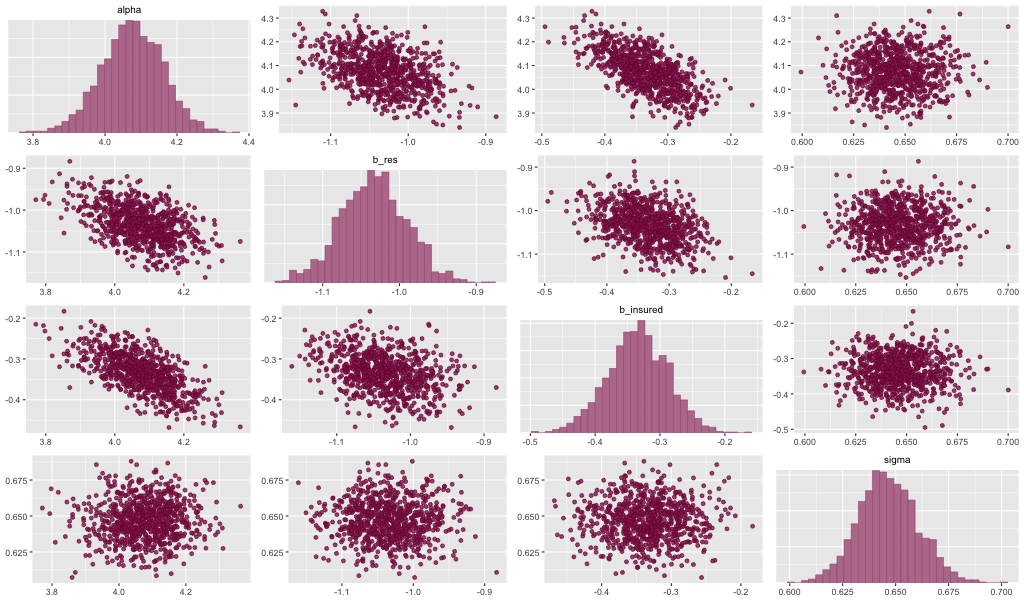
MCMC chain

Các hệ số posteriors:

stan\_hist(fit,pars = params, bins = 50)



Phân phối các hệ số hồi quy



Tương quan các hệ số trong mô hình

Pair tham số:

