HW7\_my\_R\_code.R

AM

2020-11-10

setwd('/Users/AM/Documents/\_CU Masters/2020 fall Bayesian\_7393/HW')  
load(file = "bank\_salary.rda")  
options(digits=4)  
library(rstan)

## Loading required package: StanHeaders

## Loading required package: ggplot2

## rstan (Version 2.21.2, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend calling  
## options(mc.cores = parallel::detectCores()).  
## To avoid recompilation of unchanged Stan programs, we recommend calling  
## rstan\_options(auto\_write = TRUE)

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(tidyr)

##   
## Attaching package: 'tidyr'

## The following object is masked from 'package:rstan':  
##   
## extract

library(ggplot2)  
library(mvtnorm)  
library(bayesplot)

## This is bayesplot version 1.7.2

## - Online documentation and vignettes at mc-stan.org/bayesplot

## - bayesplot theme set to bayesplot::theme\_default()

## \* Does \_not\_ affect other ggplot2 plots

## \* See ?bayesplot\_theme\_set for details on theme setting

# obtain the number of observations  
n = length(bank\_salary$bsalary)  
# obtain the sample variance of the data  
v = var(bank\_salary$bsalary)  
N=10^5 # number of iterations  
X = cbind(1, bank\_salary$male, bank\_salary$education, bank\_salary$experience, bank\_salary$time)  
  
# Create model.   
  
## Model\_1  
# Prior distributions:  
# beta\_j ~ N(0, 10^4) (sigmasq = 10^4)  
# tau ~ Gamma(0.01,0.01).  
stanmod\_1 = "  
data {  
 int<lower=1> n; // number of observations  
 vector[n] y; // data  
 vector[n] male; //covariate  
 vector[n] edu; //covariate  
 vector[n] exper; //covariate  
 vector[n] t; //covariate  
 real<lower=0> v; // sample variance of y   
}  
parameters {  
 real<lower=0> prec; // tau  
 real beta0;  
 real beta1;  
 real beta2;  
 real beta3;  
 real beta4;   
}  
  
transformed parameters{  
 real<lower=0> sigma; //get sigma from the precision  
 sigma = sqrt(1/prec);  
}  
model {  
 //specify priors  
 beta0 ~ normal(0.0, 100);  
 beta1 ~ normal(0.0, 100);  
 beta2 ~ normal(0.0, 100);  
 beta3 ~ normal(0.0, 100);  
 beta4 ~ normal(0.0, 100);  
 prec ~ gamma(0.01, 0.01);  
  
 // data distribution  
 for(i in 1:n){  
 y[i] ~ normal(beta0 + beta1\*male[i] + beta2\*edu[i] + beta3\*exper[i] + beta4\*t[i], sigma);  
 }  
}  
generated quantities {  
 real<lower=0> sigmasq; //get sigmasq from the precision  
 real Rbsq;  
 sigmasq = 1/prec;  
 Rbsq = 1 - sigmasq / v;  
}  
"  
## Model\_2  
# Prior distributions:  
# beta ~ N(0, sigma^2 \* c^2 \* (X'X)^(-1)) with c^2 = n  
# tau ~ Gamma(0.01,0.01).  
stanmod\_2 = "  
data {  
 int<lower=1> n; // number of observations  
 vector[n] y; // data  
 matrix[n, 5] X; //covariates  
 vector[5] mu0; //prior mean for beta  
 cov\_matrix[5] V; //V part of Zellner's g-prior  
 ////cov\_matrix[n] I; //nxn identity matrix  
 real<lower=0> csq; //constant for Zellner's g-prior  
 real<lower=0> v; //sample variance  
}  
parameters {  
 real<lower=0> prec;  
 vector[5] beta;  
}  
transformed parameters{  
 real<lower=0> sigmasq; //get sigmasq from the precision  
 ////vector[n] mu; // mean of responses  
 sigmasq = 1/prec;  
 ////mu = X \* beta;  
}  
model {  
 vector[n] mu; // mean of responses. Temporary  
 mu = X \* beta;  
 // specify priors  
 beta ~ multi\_normal(mu0, sigmasq\*csq\*V);  
 prec ~ gamma(0.01, 0.01);  
  
 // specify data distribution  
 // y ~ multi\_normal(mu, sigmasq \* I);  
 // The statement below is equivalent to what is above,  
 // but faster since we don't sample from a multivariate  
 // normal distribution internally.  
 for(i in 1:n) {  
 y[i] ~ normal(mu[i], sqrt(sigmasq));  
 }  
}  
generated quantities {  
 real Rbsq;  
 Rbsq = 1 - sigmasq / v;  
}  
"  
  
## Model\_3  
# Create model. Notice the quotes  
stanmod\_3 = "  
data {  
 int<lower=1> n; // number of observations  
 vector[n] y; // data  
 matrix[n, 5] X; //covariates  
 vector[5] mu0; //prior mean for beta  
 cov\_matrix[5] V; //V part of Zellner's g-prior  
 cov\_matrix[n] I; //nxn identity matrix  
 real<lower=0> csq; //constant for Zellner's g-prior  
 real<lower=0> v; //sample variance  
}  
parameters {  
 real<lower=0> prec;  
 vector[5] beta;  
}  
transformed parameters{  
 real<lower=0> sigmasq; //get sigmasq from the precision  
 sigmasq = 1/prec;  
}  
model {  
 vector[n] mu; // mean of responses. Temporary  
 mu = X \* beta;  
 //prior distributions  
 beta ~ multi\_normal(mu0, sigmasq\*csq\*V);  
 prec ~ gamma(0.01, 0.01);  
 // data distribution  
 //y ~ multi\_normal(X\*beta, sigmasq\*I);  
 for(i in 1:n) {  
 y[i] ~ normal(mu[i], sqrt(sigmasq));  
 }  
}  
generated quantities {  
 real Rbsq;  
 Rbsq = 1 - sigmasq / v;  
}  
"  
  
# fit models using stan with 4 chains  
stan\_dat\_1 = list(n = n, y = bank\_salary$bsalary, male = bank\_salary$male,  
 edu = bank\_salary$education, exper = bank\_salary$experience,  
 t = bank\_salary$time, v = v)  
salary\_fit\_1 = stan(model\_code = stanmod\_1, data = stan\_dat\_1, iter = N, chains = 4)

## Trying to compile a simple C file

## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c  
## clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/Rcpp/include/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/unsupported" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/BH/include" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/src/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppParallel/include/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/rstan/include" -DEIGEN\_NO\_DEBUG -DBOOST\_DISABLE\_ASSERTS -DBOOST\_PENDING\_INTEGER\_LOG2\_HPP -DSTAN\_THREADS -DBOOST\_NO\_AUTO\_PTR -include '/Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp' -D\_REENTRANT -DRCPP\_PARALLEL\_USE\_TBB=1 -I/usr/local/include -fPIC -Wall -g -O2 -c foo.c -o foo.o  
## In file included from <built-in>:1:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp:13:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Dense:1:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:88:  
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:1: error: unknown type name 'namespace'  
## namespace Eigen {  
## ^  
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:16: error: expected ';' after top level declarator  
## namespace Eigen {  
## ^  
## ;  
## In file included from <built-in>:1:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp:13:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Dense:1:  
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:96:10: fatal error: 'complex' file not found  
## #include <complex>  
## ^~~~~~~~~  
## 3 errors generated.  
## make: \*\*\* [foo.o] Error 1  
##   
## SAMPLING FOR MODEL '1bd27150f17f78596f2c8f567e1ea6d8' NOW (CHAIN 1).  
## Chain 1:   
## Chain 1: Gradient evaluation took 5.7e-05 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.57 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## Chain 1:   
## Chain 1: Iteration: 1 / 100000 [ 0%] (Warmup)  
## Chain 1: Iteration: 10000 / 100000 [ 10%] (Warmup)  
## Chain 1: Iteration: 20000 / 100000 [ 20%] (Warmup)  
## Chain 1: Iteration: 30000 / 100000 [ 30%] (Warmup)  
## Chain 1: Iteration: 40000 / 100000 [ 40%] (Warmup)  
## Chain 1: Iteration: 50000 / 100000 [ 50%] (Warmup)  
## Chain 1: Iteration: 50001 / 100000 [ 50%] (Sampling)  
## Chain 1: Iteration: 60000 / 100000 [ 60%] (Sampling)  
## Chain 1: Iteration: 70000 / 100000 [ 70%] (Sampling)  
## Chain 1: Iteration: 80000 / 100000 [ 80%] (Sampling)  
## Chain 1: Iteration: 90000 / 100000 [ 90%] (Sampling)  
## Chain 1: Iteration: 100000 / 100000 [100%] (Sampling)  
## Chain 1:   
## Chain 1: Elapsed Time: 13.0007 seconds (Warm-up)  
## Chain 1: 19.4299 seconds (Sampling)  
## Chain 1: 32.4306 seconds (Total)  
## Chain 1:   
##   
## SAMPLING FOR MODEL '1bd27150f17f78596f2c8f567e1ea6d8' NOW (CHAIN 2).  
## Chain 2:   
## Chain 2: Gradient evaluation took 2e-05 seconds  
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.  
## Chain 2: Adjust your expectations accordingly!  
## Chain 2:   
## Chain 2:   
## Chain 2: Iteration: 1 / 100000 [ 0%] (Warmup)  
## Chain 2: Iteration: 10000 / 100000 [ 10%] (Warmup)  
## Chain 2: Iteration: 20000 / 100000 [ 20%] (Warmup)  
## Chain 2: Iteration: 30000 / 100000 [ 30%] (Warmup)  
## Chain 2: Iteration: 40000 / 100000 [ 40%] (Warmup)  
## Chain 2: Iteration: 50000 / 100000 [ 50%] (Warmup)  
## Chain 2: Iteration: 50001 / 100000 [ 50%] (Sampling)  
## Chain 2: Iteration: 60000 / 100000 [ 60%] (Sampling)  
## Chain 2: Iteration: 70000 / 100000 [ 70%] (Sampling)  
## Chain 2: Iteration: 80000 / 100000 [ 80%] (Sampling)  
## Chain 2: Iteration: 90000 / 100000 [ 90%] (Sampling)  
## Chain 2: Iteration: 100000 / 100000 [100%] (Sampling)  
## Chain 2:   
## Chain 2: Elapsed Time: 14.2447 seconds (Warm-up)  
## Chain 2: 17.7012 seconds (Sampling)  
## Chain 2: 31.9459 seconds (Total)  
## Chain 2:   
##   
## SAMPLING FOR MODEL '1bd27150f17f78596f2c8f567e1ea6d8' NOW (CHAIN 3).  
## Chain 3:   
## Chain 3: Gradient evaluation took 2.1e-05 seconds  
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.21 seconds.  
## Chain 3: Adjust your expectations accordingly!  
## Chain 3:   
## Chain 3:   
## Chain 3: Iteration: 1 / 100000 [ 0%] (Warmup)  
## Chain 3: Iteration: 10000 / 100000 [ 10%] (Warmup)  
## Chain 3: Iteration: 20000 / 100000 [ 20%] (Warmup)  
## Chain 3: Iteration: 30000 / 100000 [ 30%] (Warmup)  
## Chain 3: Iteration: 40000 / 100000 [ 40%] (Warmup)  
## Chain 3: Iteration: 50000 / 100000 [ 50%] (Warmup)  
## Chain 3: Iteration: 50001 / 100000 [ 50%] (Sampling)  
## Chain 3: Iteration: 60000 / 100000 [ 60%] (Sampling)  
## Chain 3: Iteration: 70000 / 100000 [ 70%] (Sampling)  
## Chain 3: Iteration: 80000 / 100000 [ 80%] (Sampling)  
## Chain 3: Iteration: 90000 / 100000 [ 90%] (Sampling)  
## Chain 3: Iteration: 100000 / 100000 [100%] (Sampling)  
## Chain 3:   
## Chain 3: Elapsed Time: 14.4575 seconds (Warm-up)  
## Chain 3: 17.4898 seconds (Sampling)  
## Chain 3: 31.9472 seconds (Total)  
## Chain 3:   
##   
## SAMPLING FOR MODEL '1bd27150f17f78596f2c8f567e1ea6d8' NOW (CHAIN 4).  
## Chain 4:   
## Chain 4: Gradient evaluation took 1.4e-05 seconds  
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.14 seconds.  
## Chain 4: Adjust your expectations accordingly!  
## Chain 4:   
## Chain 4:   
## Chain 4: Iteration: 1 / 100000 [ 0%] (Warmup)  
## Chain 4: Iteration: 10000 / 100000 [ 10%] (Warmup)  
## Chain 4: Iteration: 20000 / 100000 [ 20%] (Warmup)  
## Chain 4: Iteration: 30000 / 100000 [ 30%] (Warmup)  
## Chain 4: Iteration: 40000 / 100000 [ 40%] (Warmup)  
## Chain 4: Iteration: 50000 / 100000 [ 50%] (Warmup)  
## Chain 4: Iteration: 50001 / 100000 [ 50%] (Sampling)  
## Chain 4: Iteration: 60000 / 100000 [ 60%] (Sampling)  
## Chain 4: Iteration: 70000 / 100000 [ 70%] (Sampling)  
## Chain 4: Iteration: 80000 / 100000 [ 80%] (Sampling)  
## Chain 4: Iteration: 90000 / 100000 [ 90%] (Sampling)  
## Chain 4: Iteration: 100000 / 100000 [100%] (Sampling)  
## Chain 4:   
## Chain 4: Elapsed Time: 13.1969 seconds (Warm-up)  
## Chain 4: 15.7592 seconds (Sampling)  
## Chain 4: 28.9561 seconds (Total)  
## Chain 4:

stan\_dat\_2 = list(n = n, y = bank\_salary$bsalary,  
 X = X, mu0 = c(0, 0, 0, 0, 0), V = solve(crossprod(X)),  
 I = diag(n), csq = n, v = v)  
salary\_fit\_2 = stan(model\_code = stanmod\_2, data = stan\_dat\_2, iter = N, chains = 4)

## Trying to compile a simple C file

## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c  
## clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/Rcpp/include/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/unsupported" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/BH/include" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/src/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppParallel/include/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/rstan/include" -DEIGEN\_NO\_DEBUG -DBOOST\_DISABLE\_ASSERTS -DBOOST\_PENDING\_INTEGER\_LOG2\_HPP -DSTAN\_THREADS -DBOOST\_NO\_AUTO\_PTR -include '/Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp' -D\_REENTRANT -DRCPP\_PARALLEL\_USE\_TBB=1 -I/usr/local/include -fPIC -Wall -g -O2 -c foo.c -o foo.o  
## In file included from <built-in>:1:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp:13:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Dense:1:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:88:  
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:1: error: unknown type name 'namespace'  
## namespace Eigen {  
## ^  
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:16: error: expected ';' after top level declarator  
## namespace Eigen {  
## ^  
## ;  
## In file included from <built-in>:1:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp:13:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Dense:1:  
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:96:10: fatal error: 'complex' file not found  
## #include <complex>  
## ^~~~~~~~~  
## 3 errors generated.  
## make: \*\*\* [foo.o] Error 1  
##   
## SAMPLING FOR MODEL 'eb7f6ca93339c7efedf67171e3e8b097' NOW (CHAIN 1).  
## Chain 1:   
## Chain 1: Gradient evaluation took 6.3e-05 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.63 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## Chain 1:   
## Chain 1: Iteration: 1 / 100000 [ 0%] (Warmup)  
## Chain 1: Iteration: 10000 / 100000 [ 10%] (Warmup)  
## Chain 1: Iteration: 20000 / 100000 [ 20%] (Warmup)  
## Chain 1: Iteration: 30000 / 100000 [ 30%] (Warmup)  
## Chain 1: Iteration: 40000 / 100000 [ 40%] (Warmup)  
## Chain 1: Iteration: 50000 / 100000 [ 50%] (Warmup)  
## Chain 1: Iteration: 50001 / 100000 [ 50%] (Sampling)  
## Chain 1: Iteration: 60000 / 100000 [ 60%] (Sampling)  
## Chain 1: Iteration: 70000 / 100000 [ 70%] (Sampling)  
## Chain 1: Iteration: 80000 / 100000 [ 80%] (Sampling)  
## Chain 1: Iteration: 90000 / 100000 [ 90%] (Sampling)  
## Chain 1: Iteration: 100000 / 100000 [100%] (Sampling)  
## Chain 1:   
## Chain 1: Elapsed Time: 14.195 seconds (Warm-up)  
## Chain 1: 17.6589 seconds (Sampling)  
## Chain 1: 31.8539 seconds (Total)  
## Chain 1:   
##   
## SAMPLING FOR MODEL 'eb7f6ca93339c7efedf67171e3e8b097' NOW (CHAIN 2).  
## Chain 2:   
## Chain 2: Gradient evaluation took 2.6e-05 seconds  
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.26 seconds.  
## Chain 2: Adjust your expectations accordingly!  
## Chain 2:   
## Chain 2:   
## Chain 2: Iteration: 1 / 100000 [ 0%] (Warmup)  
## Chain 2: Iteration: 10000 / 100000 [ 10%] (Warmup)  
## Chain 2: Iteration: 20000 / 100000 [ 20%] (Warmup)  
## Chain 2: Iteration: 30000 / 100000 [ 30%] (Warmup)  
## Chain 2: Iteration: 40000 / 100000 [ 40%] (Warmup)  
## Chain 2: Iteration: 50000 / 100000 [ 50%] (Warmup)  
## Chain 2: Iteration: 50001 / 100000 [ 50%] (Sampling)  
## Chain 2: Iteration: 60000 / 100000 [ 60%] (Sampling)  
## Chain 2: Iteration: 70000 / 100000 [ 70%] (Sampling)  
## Chain 2: Iteration: 80000 / 100000 [ 80%] (Sampling)  
## Chain 2: Iteration: 90000 / 100000 [ 90%] (Sampling)  
## Chain 2: Iteration: 100000 / 100000 [100%] (Sampling)  
## Chain 2:   
## Chain 2: Elapsed Time: 13.5577 seconds (Warm-up)  
## Chain 2: 17.3469 seconds (Sampling)  
## Chain 2: 30.9046 seconds (Total)  
## Chain 2:   
##   
## SAMPLING FOR MODEL 'eb7f6ca93339c7efedf67171e3e8b097' NOW (CHAIN 3).  
## Chain 3:   
## Chain 3: Gradient evaluation took 2e-05 seconds  
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.  
## Chain 3: Adjust your expectations accordingly!  
## Chain 3:   
## Chain 3:   
## Chain 3: Iteration: 1 / 100000 [ 0%] (Warmup)  
## Chain 3: Iteration: 10000 / 100000 [ 10%] (Warmup)  
## Chain 3: Iteration: 20000 / 100000 [ 20%] (Warmup)  
## Chain 3: Iteration: 30000 / 100000 [ 30%] (Warmup)  
## Chain 3: Iteration: 40000 / 100000 [ 40%] (Warmup)  
## Chain 3: Iteration: 50000 / 100000 [ 50%] (Warmup)  
## Chain 3: Iteration: 50001 / 100000 [ 50%] (Sampling)  
## Chain 3: Iteration: 60000 / 100000 [ 60%] (Sampling)  
## Chain 3: Iteration: 70000 / 100000 [ 70%] (Sampling)  
## Chain 3: Iteration: 80000 / 100000 [ 80%] (Sampling)  
## Chain 3: Iteration: 90000 / 100000 [ 90%] (Sampling)  
## Chain 3: Iteration: 100000 / 100000 [100%] (Sampling)  
## Chain 3:   
## Chain 3: Elapsed Time: 13.2769 seconds (Warm-up)  
## Chain 3: 19.5474 seconds (Sampling)  
## Chain 3: 32.8243 seconds (Total)  
## Chain 3:   
##   
## SAMPLING FOR MODEL 'eb7f6ca93339c7efedf67171e3e8b097' NOW (CHAIN 4).  
## Chain 4:   
## Chain 4: Gradient evaluation took 2.2e-05 seconds  
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.22 seconds.  
## Chain 4: Adjust your expectations accordingly!  
## Chain 4:   
## Chain 4:   
## Chain 4: Iteration: 1 / 100000 [ 0%] (Warmup)  
## Chain 4: Iteration: 10000 / 100000 [ 10%] (Warmup)  
## Chain 4: Iteration: 20000 / 100000 [ 20%] (Warmup)  
## Chain 4: Iteration: 30000 / 100000 [ 30%] (Warmup)  
## Chain 4: Iteration: 40000 / 100000 [ 40%] (Warmup)  
## Chain 4: Iteration: 50000 / 100000 [ 50%] (Warmup)  
## Chain 4: Iteration: 50001 / 100000 [ 50%] (Sampling)  
## Chain 4: Iteration: 60000 / 100000 [ 60%] (Sampling)  
## Chain 4: Iteration: 70000 / 100000 [ 70%] (Sampling)  
## Chain 4: Iteration: 80000 / 100000 [ 80%] (Sampling)  
## Chain 4: Iteration: 90000 / 100000 [ 90%] (Sampling)  
## Chain 4: Iteration: 100000 / 100000 [100%] (Sampling)  
## Chain 4:   
## Chain 4: Elapsed Time: 13.5377 seconds (Warm-up)  
## Chain 4: 19.2035 seconds (Sampling)  
## Chain 4: 32.7412 seconds (Total)  
## Chain 4:

stan\_dat\_3 = list(n = n, y = bank\_salary$bsalary,  
 X = X, mu0 = c(0, 0, 0, 0, 0), V = solve(crossprod(X)),  
 I = diag(n), csq = 10^3, v = v)  
salary\_fit\_3 = stan(model\_code = stanmod\_3, data = stan\_dat\_3, iter = N, chains = 4)

## Trying to compile a simple C file

## Running /Library/Frameworks/R.framework/Resources/bin/R CMD SHLIB foo.c  
## clang -mmacosx-version-min=10.13 -I"/Library/Frameworks/R.framework/Resources/include" -DNDEBUG -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/Rcpp/include/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/unsupported" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/BH/include" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/src/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppParallel/include/" -I"/Library/Frameworks/R.framework/Versions/4.0/Resources/library/rstan/include" -DEIGEN\_NO\_DEBUG -DBOOST\_DISABLE\_ASSERTS -DBOOST\_PENDING\_INTEGER\_LOG2\_HPP -DSTAN\_THREADS -DBOOST\_NO\_AUTO\_PTR -include '/Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp' -D\_REENTRANT -DRCPP\_PARALLEL\_USE\_TBB=1 -I/usr/local/include -fPIC -Wall -g -O2 -c foo.c -o foo.o  
## In file included from <built-in>:1:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp:13:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Dense:1:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:88:  
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:1: error: unknown type name 'namespace'  
## namespace Eigen {  
## ^  
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:16: error: expected ';' after top level declarator  
## namespace Eigen {  
## ^  
## ;  
## In file included from <built-in>:1:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp:13:  
## In file included from /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Dense:1:  
## /Library/Frameworks/R.framework/Versions/4.0/Resources/library/RcppEigen/include/Eigen/Core:96:10: fatal error: 'complex' file not found  
## #include <complex>  
## ^~~~~~~~~  
## 3 errors generated.  
## make: \*\*\* [foo.o] Error 1  
##   
## SAMPLING FOR MODEL 'a90792f942d252328a1a9f9f341082d2' NOW (CHAIN 1).  
## Chain 1:   
## Chain 1: Gradient evaluation took 5.4e-05 seconds  
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.54 seconds.  
## Chain 1: Adjust your expectations accordingly!  
## Chain 1:   
## Chain 1:   
## Chain 1: Iteration: 1 / 100000 [ 0%] (Warmup)  
## Chain 1: Iteration: 10000 / 100000 [ 10%] (Warmup)  
## Chain 1: Iteration: 20000 / 100000 [ 20%] (Warmup)  
## Chain 1: Iteration: 30000 / 100000 [ 30%] (Warmup)  
## Chain 1: Iteration: 40000 / 100000 [ 40%] (Warmup)  
## Chain 1: Iteration: 50000 / 100000 [ 50%] (Warmup)  
## Chain 1: Iteration: 50001 / 100000 [ 50%] (Sampling)  
## Chain 1: Iteration: 60000 / 100000 [ 60%] (Sampling)  
## Chain 1: Iteration: 70000 / 100000 [ 70%] (Sampling)  
## Chain 1: Iteration: 80000 / 100000 [ 80%] (Sampling)  
## Chain 1: Iteration: 90000 / 100000 [ 90%] (Sampling)  
## Chain 1: Iteration: 100000 / 100000 [100%] (Sampling)  
## Chain 1:   
## Chain 1: Elapsed Time: 12.0559 seconds (Warm-up)  
## Chain 1: 18.7176 seconds (Sampling)  
## Chain 1: 30.7735 seconds (Total)  
## Chain 1:   
##   
## SAMPLING FOR MODEL 'a90792f942d252328a1a9f9f341082d2' NOW (CHAIN 2).  
## Chain 2:   
## Chain 2: Gradient evaluation took 2e-05 seconds  
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.  
## Chain 2: Adjust your expectations accordingly!  
## Chain 2:   
## Chain 2:   
## Chain 2: Iteration: 1 / 100000 [ 0%] (Warmup)  
## Chain 2: Iteration: 10000 / 100000 [ 10%] (Warmup)  
## Chain 2: Iteration: 20000 / 100000 [ 20%] (Warmup)  
## Chain 2: Iteration: 30000 / 100000 [ 30%] (Warmup)  
## Chain 2: Iteration: 40000 / 100000 [ 40%] (Warmup)  
## Chain 2: Iteration: 50000 / 100000 [ 50%] (Warmup)  
## Chain 2: Iteration: 50001 / 100000 [ 50%] (Sampling)  
## Chain 2: Iteration: 60000 / 100000 [ 60%] (Sampling)  
## Chain 2: Iteration: 70000 / 100000 [ 70%] (Sampling)  
## Chain 2: Iteration: 80000 / 100000 [ 80%] (Sampling)  
## Chain 2: Iteration: 90000 / 100000 [ 90%] (Sampling)  
## Chain 2: Iteration: 100000 / 100000 [100%] (Sampling)  
## Chain 2:   
## Chain 2: Elapsed Time: 12.3582 seconds (Warm-up)  
## Chain 2: 14.8846 seconds (Sampling)  
## Chain 2: 27.2428 seconds (Total)  
## Chain 2:   
##   
## SAMPLING FOR MODEL 'a90792f942d252328a1a9f9f341082d2' NOW (CHAIN 3).  
## Chain 3:   
## Chain 3: Gradient evaluation took 2e-05 seconds  
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.  
## Chain 3: Adjust your expectations accordingly!  
## Chain 3:   
## Chain 3:   
## Chain 3: Iteration: 1 / 100000 [ 0%] (Warmup)  
## Chain 3: Iteration: 10000 / 100000 [ 10%] (Warmup)  
## Chain 3: Iteration: 20000 / 100000 [ 20%] (Warmup)  
## Chain 3: Iteration: 30000 / 100000 [ 30%] (Warmup)  
## Chain 3: Iteration: 40000 / 100000 [ 40%] (Warmup)  
## Chain 3: Iteration: 50000 / 100000 [ 50%] (Warmup)  
## Chain 3: Iteration: 50001 / 100000 [ 50%] (Sampling)  
## Chain 3: Iteration: 60000 / 100000 [ 60%] (Sampling)  
## Chain 3: Iteration: 70000 / 100000 [ 70%] (Sampling)  
## Chain 3: Iteration: 80000 / 100000 [ 80%] (Sampling)  
## Chain 3: Iteration: 90000 / 100000 [ 90%] (Sampling)  
## Chain 3: Iteration: 100000 / 100000 [100%] (Sampling)  
## Chain 3:   
## Chain 3: Elapsed Time: 12.0903 seconds (Warm-up)  
## Chain 3: 15.6898 seconds (Sampling)  
## Chain 3: 27.7801 seconds (Total)  
## Chain 3:   
##   
## SAMPLING FOR MODEL 'a90792f942d252328a1a9f9f341082d2' NOW (CHAIN 4).  
## Chain 4:   
## Chain 4: Gradient evaluation took 2e-05 seconds  
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 0.2 seconds.  
## Chain 4: Adjust your expectations accordingly!  
## Chain 4:   
## Chain 4:   
## Chain 4: Iteration: 1 / 100000 [ 0%] (Warmup)  
## Chain 4: Iteration: 10000 / 100000 [ 10%] (Warmup)  
## Chain 4: Iteration: 20000 / 100000 [ 20%] (Warmup)  
## Chain 4: Iteration: 30000 / 100000 [ 30%] (Warmup)  
## Chain 4: Iteration: 40000 / 100000 [ 40%] (Warmup)  
## Chain 4: Iteration: 50000 / 100000 [ 50%] (Warmup)  
## Chain 4: Iteration: 50001 / 100000 [ 50%] (Sampling)  
## Chain 4: Iteration: 60000 / 100000 [ 60%] (Sampling)  
## Chain 4: Iteration: 70000 / 100000 [ 70%] (Sampling)  
## Chain 4: Iteration: 80000 / 100000 [ 80%] (Sampling)  
## Chain 4: Iteration: 90000 / 100000 [ 90%] (Sampling)  
## Chain 4: Iteration: 100000 / 100000 [100%] (Sampling)  
## Chain 4:   
## Chain 4: Elapsed Time: 12.0295 seconds (Warm-up)  
## Chain 4: 15.997 seconds (Sampling)  
## Chain 4: 28.0265 seconds (Total)  
## Chain 4:

dim(salary\_fit\_3)# i.e. a warmup of 50,000 iterations and 50,000 saved values

## [1] 50000 4 9

# calculate the Gelman-Rubin statistic  
summary(salary\_fit\_1)$summary[,"Rhat"]

## prec beta0 beta1 beta2 beta3 beta4 sigma sigmasq Rbsq lp\_\_   
## 1 1 1 1 1 1 1 1 1 1

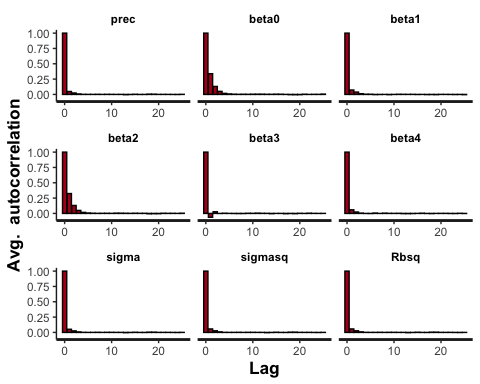
summary(salary\_fit\_2)$summary[,"Rhat"]

## prec beta[1] beta[2] beta[3] beta[4] beta[5] sigmasq Rbsq lp\_\_   
## 1 1 1 1 1 1 1 1 1

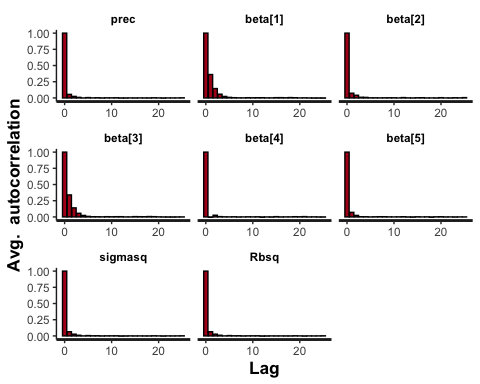
summary(salary\_fit\_3)$summary[,"Rhat"]

## prec beta[1] beta[2] beta[3] beta[4] beta[5] sigmasq Rbsq lp\_\_   
## 1 1 1 1 1 1 1 1 1

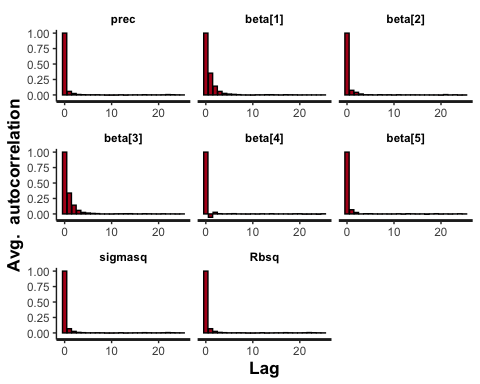
# check the ACF of draws from the MCMC chain  
stan\_ac(salary\_fit\_1)



stan\_ac(salary\_fit\_2)



stan\_ac(salary\_fit\_3)



#effectiv sample size  
summary(salary\_fit\_1)$summary[, "n\_eff"]

## prec beta0 beta1 beta2 beta3 beta4 sigma sigmasq Rbsq lp\_\_   
## 170295 93142 158903 95008 214435 170279 168941 166912 166912 81461

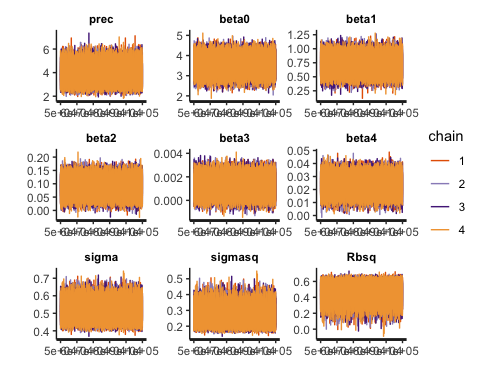
summary(salary\_fit\_2)$summary[, "n\_eff"]

## prec beta[1] beta[2] beta[3] beta[4] beta[5] sigmasq Rbsq lp\_\_   
## 167992 88502 155073 90990 189283 164772 163238 163238 83168

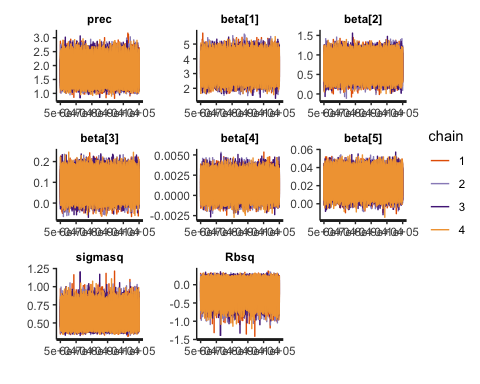
summary(salary\_fit\_3)$summary[, "n\_eff"]

## prec beta[1] beta[2] beta[3] beta[4] beta[5] sigmasq Rbsq lp\_\_   
## 167273 90501 154499 91775 207937 167419 163603 163603 80256

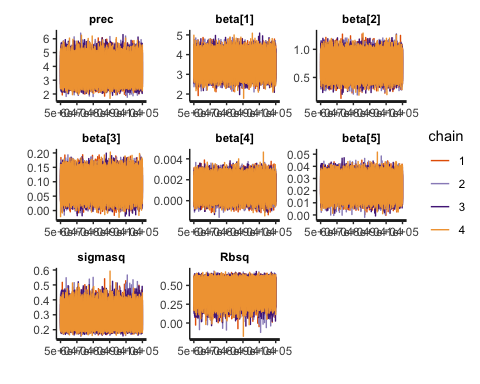
# traceplots  
traceplot(salary\_fit\_1)



traceplot(salary\_fit\_2)



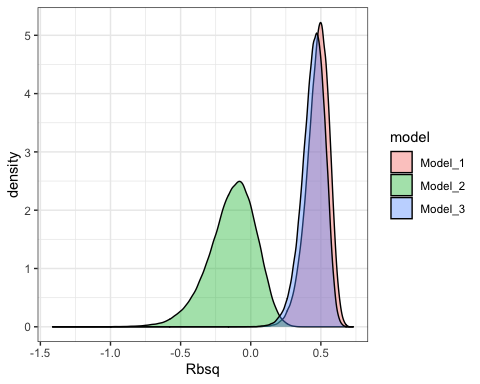
traceplot(salary\_fit\_3)



# output  
a = array(0, dim = c(3,6,4), dimnames = list(c("Model\_1", "Model\_2", "Model\_3"),  
 c("beta\_0", "beta\_1","beta\_2", "beta\_3", "beta\_4", "sigmasq"),  
 c("Mean", "Median","2.5%", "97.5%")))  
a[1,,] = summary(salary\_fit\_1)$summary[,c("mean", "50%","2.5%", "97.5%")][c(2,3,4,5,6,8),]  
a[2,,] = summary(salary\_fit\_2)$summary[,c("mean", "50%","2.5%", "97.5%")][c(2,3,4,5,6,7),]  
a[3,,] = summary(salary\_fit\_3)$summary[,c("mean", "50%","2.5%", "97.5%")][c(2,3,4,5,6,7),]  
  
for (i in dimnames(a)[[2]]) {  
 cat('\n',i,'\n')  
 print(a[,i,])  
}

##   
## beta\_0   
## Mean Median 2.5% 97.5%  
## Model\_1 3.526 3.526 2.874 4.178  
## Model\_2 3.488 3.487 2.533 4.444  
## Model\_3 3.522 3.522 2.850 4.195  
##   
## beta\_1   
## Mean Median 2.5% 97.5%  
## Model\_1 0.7219 0.7218 0.4887 0.9558  
## Model\_2 0.7148 0.7150 0.3731 1.0572  
## Model\_3 0.7216 0.7219 0.4798 0.9618  
##   
## beta\_2   
## Mean Median 2.5% 97.5%  
## Model\_1 0.09005 0.09011 0.04088 0.1392  
## Model\_2 0.08914 0.08925 0.01686 0.1612  
## Model\_3 0.08994 0.09000 0.03928 0.1407  
##   
## beta\_3   
## Mean Median 2.5% 97.5%  
## Model\_1 0.001267 0.001265 1.020e-04 0.002429  
## Model\_2 0.001256 0.001257 -4.478e-04 0.002959  
## Model\_3 0.001267 0.001266 6.568e-05 0.002473  
##   
## beta\_4   
## Mean Median 2.5% 97.5%  
## Model\_1 0.02344 0.02344 0.013081 0.03375  
## Model\_2 0.02317 0.02319 0.008012 0.03824  
## Model\_3 0.02341 0.02341 0.012809 0.03402  
##   
## sigmasq   
## Mean Median 2.5% 97.5%  
## Model\_1 0.2638 0.2597 0.1957 0.3552  
## Model\_2 0.5710 0.5630 0.4277 0.7612  
## Model\_3 0.2794 0.2754 0.2096 0.3727

###\_2\_distribution of Rb^2  
fitss = dplyr::bind\_rows(cbind(as.data.frame(salary\_fit\_1), model = "Model\_1"),  
 cbind(as.data.frame(salary\_fit\_2), model = "Model\_2"),  
 cbind(as.data.frame(salary\_fit\_3), model = "Model\_3"))  
fitss\_df = as.data.frame(fitss)  
# comparison of Rbsq  
ggplot(fitss, aes(x = Rbsq)) + theme\_bw() +  
 geom\_density(aes(fill = model), alpha = 0.4)



###\_3\_Provide a point estimate  
x\_names = names(bank\_salary)  
for (m in 1:3){  
 cat("bsalary\_hut = ", a[m,1,1], "+ ")  
 for (i in 2:4) {  
 cat(a[m,i,1], "\*", x\_names[i], "+ ")  
 }  
 cat(a[m,5,1], "\*", x\_names[5])  
 cat("\n")  
}

## bsalary\_hut = 3.526 + 0.7219 \* male + 0.09005 \* education + 0.001267 \* experience + 0.02344 \* time  
## bsalary\_hut = 3.488 + 0.7148 \* male + 0.08914 \* education + 0.001256 \* experience + 0.02317 \* time  
## bsalary\_hut = 3.522 + 0.7216 \* male + 0.08994 \* education + 0.001267 \* experience + 0.02341 \* time