Assignment 11

Shinjon Ghosh

2025-04-18

Loading Libraries

```
library(rstan)
```

```
## Warning: package 'rstan' was built under R version 4.4.3
```

```
## Loading required package: StanHeaders
```

```
## Warning: package 'StanHeaders' was built under R version 4.4.3
```

```
##
## rstan version 2.32.7 (Stan version 2.32.2)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calli
ng
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
## For within-chain threading using `reduce_sum()` or `map_rect()` Stan funct
ions,
## change `threads_per_chain` option:
## rstan_options(threads_per_chain = 1)
```

```
## Do not specify '-march=native' in 'LOCAL_CPPFLAGS' or a Makevars file
```

```
library(brms)
```

```
## Warning: package 'brms' was built under R version 4.4.3
```

```
## Loading required package: Rcpp
```

```
## Loading 'brms' package (version 2.22.0). Useful instructions
## can be found by typing help('brms'). A more detailed introduction
## to the package is available through vignette('brms_overview').
```

```
##
## Attaching package: 'brms'
```

```
## The following object is masked from 'package:rstan':
##
##     loo
```

```
## The following object is masked from 'package:stats':
##
##     ar
```

```r
# Simulate 100 variables from standard normal distributions
set.seed(123)
n <- 100
X <- matrix(rnorm(n * 100), nrow = n, ncol = 100)

# Calculate the mean parameter μ_i
mu <- 2 * X[, 1] + X[, 2] + 0.5 * X[, 5] + 1.5 * X[, 10]

# Generate the count response Y_i from Poisson(μ_i)
Y <- rpois(n, mu)
```

```
## Warning in rpois(n, mu): NAs produced
```

1.Stan Bayesian Model with Normal Priors

```r
stan_model_normal <- "
data {
  int<lower=0> N;  // number of observations
  int<lower=0> Y[N];  // count response
  matrix[N, 4] X;  // predictor matrix
}
parameters {
  vector[4] beta;  // regression coefficients
}
model {
  beta ~ normal(0, 10);  // Normal prior
  Y ~ poisson_log(X * beta);  // Poisson likelihood
}
"

# Remove any NA values from Y
valid_indices <- !is.na(Y)
X <- X[valid_indices, ]
Y <- Y[valid_indices]
n <- length(Y)

# Prepare data for Stan
stan_data <- list(N = n, Y = Y, X = X[, c(1, 2, 5, 10)])


# Fit the model
fit_normal <- stan(model_code = stan_model_normal, data = stan_data, iter = 2
000, chains = 2)
```

```
## WARNING: Rtools is required to build R packages, but is not currently inst
alled.
##
## Please download and install the appropriate version of Rtools for 4.4.2 fr
```

```
om
## https://cran.r-project.org/bin/windows/Rtools/.

## Trying to compile a simple C file

## Running "D:/R-4.4.2/bin/x64/Rcmd.exe" SHLIB foo.c
## using C compiler: 'gcc.exe (GCC) 13.3.0'
## gcc  -I"D:/R-4.4.2/include" -DNDEBUG   -I"D:/R-4.4.2/library/Rcpp/include/
"  -I"D:/R-4.4.2/library/RcppEigen/include/"  -I"D:/R-4.4.2/library/RcppEigen
/include/unsupported"  -I"D:/R-4.4.2/library/BH/include" -I"D:/R-4.4.2/librar
y/StanHeaders/include/src/"  -I"D:/R-4.4.2/library/StanHeaders/include/"  -I"
D:/R-4.4.2/library/RcppParallel/include/" -DRCPP_PARALLEL_USE_TBB=1 -I"D:/R-4
.4.2/library/rstan/include" -DEIGEN_NO_DEBUG  -DBOOST_DISABLE_ASSERTS  -DBOOS
T_PENDING_INTEGER_LOG2_HPP  -DSTAN_THREADS  -DUSE_STANC3 -DSTRICT_R_HEADERS
-DBOOST_PHOENIX_NO_VARIADIC_EXPRESSION  -D_HAS_AUTO_PTR_ETC=0  -include "D:/R
-4.4.2/library/StanHeaders/include/stan/math/prim/fun/Eigen.hpp"  -std=c++1y
-I"C:/RBuildTools/4.4/x86_64-w64-mingw32.static.posix/include"     -O2 -Wall
-mfpmath=sse -msse2 -mstackrealign  -c foo.c -o foo.o
## cc1.exe: warning: command-line option '-std=c++14' is valid for C++/ObjC++
but not for C
## In file included from D:/R-4.4.2/library/RcppEigen/include/Eigen/Core:19,
##                  from D:/R-4.4.2/library/RcppEigen/include/Eigen/Dense:1,
##                  from D:/R-4.4.2/library/StanHeaders/include/stan/math/pri
m/fun/Eigen.hpp:22,
##                  from <command-line>:
## D:/R-4.4.2/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:679:10:
fatal error: cmath: No such file or directory
##   679 | #include <cmath>
##       |          ^~~~~~~
## compilation terminated.
## make: *** [D:/R-4.4.2/etc/x64/Makeconf:289: foo.o] Error 1

## WARNING: Rtools is required to build R packages, but is not currently inst
alled.
##
## Please download and install the appropriate version of Rtools for 4.4.2 fr
om
## https://cran.r-project.org/bin/windows/Rtools/.

##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 3.1e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would tak
e 0.31 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
```

```
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.016 seconds (Warm-up)
## Chain 1:                0.016 seconds (Sampling)
## Chain 1:                0.032 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 3e-06 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would tak
e 0.03 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.016 seconds (Warm-up)
## Chain 2:                0.015 seconds (Sampling)
## Chain 2:                0.031 seconds (Total)
## Chain 2:
```

```
print(fit_normal)
```

```
## Inference for Stan model: anon_model.
## 2 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=2000.
##
##          mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## beta[1]  0.73    0.00 0.09  0.55  0.67  0.73  0.79  0.88  1600    1
## beta[2]  0.25    0.00 0.11  0.02  0.17  0.25  0.32  0.47  1752    1
```

```
## beta[3]  0.12    0.00 0.09 -0.07  0.06  0.12  0.19  0.31  1662    1
## beta[4]  0.45    0.00 0.09  0.27  0.39  0.45  0.51  0.61  1626    1
## lp__    -4.89    0.05 1.43 -8.51 -5.56 -4.55 -3.87 -3.16   958    1
##
## Samples were drawn using NUTS(diag_e) at Fri Apr 18 11:11:49 2025.
## 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).
```

2.Stan Bayesian Poisson regression model with Laplace Prior

```
stan_model_laplace <- "
data {
  int<lower=0> N;  // number of observations
  int<lower=0> Y[N];  // count response
  matrix[N, 4] X;  // predictor matrix
}
parameters {
  vector[4] beta;  // regression coefficients
}
model {
  beta ~ double_exponential(0, 1);  // Laplace prior
  Y ~ poisson_log(X * beta);  // Poisson likelihood
}
"

# Fit the model
fit_laplace <- stan(model_code = stan_model_laplace, data = stan_data, iter =
2000, chains = 2)

## WARNING: Rtools is required to build R packages, but is not currently inst
alled.
##
## Please download and install the appropriate version of Rtools for 4.4.2 fr
om
## https://cran.r-project.org/bin/windows/Rtools/.

## Trying to compile a simple C file

## Running "D:/R-4.4.2/bin/x64/Rcmd.exe" SHLIB foo.c
## using C compiler: 'gcc.exe (GCC) 13.3.0'
## gcc  -I"D:/R-4.4.2/include" -DNDEBUG   -I"D:/R-4.4.2/library/Rcpp/include/
"  -I"D:/R-4.4.2/library/RcppEigen/include/"  -I"D:/R-4.4.2/library/RcppEigen
/include/unsupported"  -I"D:/R-4.4.2/library/BH/include" -I"D:/R-4.4.2/librar
y/StanHeaders/include/src/"  -I"D:/R-4.4.2/library/StanHeaders/include/"  -I"
D:/R-4.4.2/library/RcppParallel/include/" -DRCPP_PARALLEL_USE_TBB=1 -I"D:/R-4
.4.2/library/rstan/include" -DEIGEN_NO_DEBUG  -DBOOST_DISABLE_ASSERTS  -DBOOS
T_PENDING_INTEGER_LOG2_HPP  -DSTAN_THREADS  -DUSE_STANC3 -DSTRICT_R_HEADERS
-DBOOST_PHOENIX_NO_VARIADIC_EXPRESSION  -D_HAS_AUTO_PTR_ETC=0  -include "D:/R
-4.4.2/library/StanHeaders/include/stan/math/prim/fun/Eigen.hpp"  -std=c++1y
```

```
-I"C:/RBuildTools/4.4/x86_64-w64-mingw32.static.posix/include"     -O2 -Wall
-mfpmath=sse -msse2 -mstackrealign  -c foo.c -o foo.o
## cc1.exe: warning: command-line option '-std=c++14' is valid for C++/ObjC++
but not for C
## In file included from D:/R-4.4.2/library/RcppEigen/include/Eigen/Core:19,
##                  from D:/R-4.4.2/library/RcppEigen/include/Eigen/Dense:1,
##                  from D:/R-4.4.2/library/StanHeaders/include/stan/math/pri
m/fun/Eigen.hpp:22,
##                  from <command-line>:
## D:/R-4.4.2/library/RcppEigen/include/Eigen/src/Core/util/Macros.h:679:10:
fatal error: cmath: No such file or directory
##   679 | #include <cmath>
##       |          ^~~~~~~
## compilation terminated.
## make: *** [D:/R-4.4.2/etc/x64/Makeconf:289: foo.o] Error 1

## WARNING: Rtools is required to build R packages, but is not currently inst
alled.
##
## Please download and install the appropriate version of Rtools for 4.4.2 fr
om
## https://cran.r-project.org/bin/windows/Rtools/.

##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 2.8e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would tak
e 0.28 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.016 seconds (Warm-up)
## Chain 1:                0.016 seconds (Sampling)
## Chain 1:                0.032 seconds (Total)
## Chain 1:
##
```

```
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 3e-06 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would tak
e 0.03 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.014 seconds (Warm-up)
## Chain 2:                0.016 seconds (Sampling)
## Chain 2:                0.03 seconds (Total)
## Chain 2:
```

```
print(fit_laplace)
```

```
## Inference for Stan model: anon_model.
## 2 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=2000.
##
##          mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## beta[1]  0.72    0.00 0.08  0.56  0.66  0.72  0.78  0.88  1904    1
## beta[2]  0.24    0.00 0.12  0.01  0.16  0.23  0.32  0.47  1824    1
## beta[3]  0.12    0.00 0.10 -0.06  0.05  0.12  0.18  0.30  1905    1
## beta[4]  0.44    0.00 0.09  0.26  0.38  0.44  0.50  0.60  2112    1
## lp__    -6.47    0.05 1.39 -9.93 -7.19 -6.16 -5.42 -4.72   946    1
##
## Samples were drawn using NUTS(diag_e) at Fri Apr 18 11:12:23 2025.
## 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).
```

3.Comparison of coefficients in both models

```
# Extract coefficients
beta_normal <- extract(fit_normal)$beta
beta_laplace <- extract(fit_laplace)$beta
```

```r
# Summary statistics
summary_normal <- apply(beta_normal, 2, mean)
summary_laplace <- apply(beta_laplace, 2, mean)

# Print the comparison
comparison <- data.frame(
  Predictor = c("X1", "X2", "X5", "X10"),
  Normal_Prior = summary_normal,
  Laplace_Prior = summary_laplace
)
print(comparison)
```

```
##   Predictor Normal_Prior Laplace_Prior
## 1        X1    0.7259523     0.7213073
## 2        X2    0.2469757     0.2361619
## 3        X5    0.1214497     0.1169451
## 4       X10    0.4475080     0.4384686
```

From the comparison table, it can be seen that coefficient of the Normal Prior and the Laplace Prior model are approximately the same and comparing them to our specified linear equation they have significant differences in their coefficients.