Bayes Main Assignment 3 of 2022

Sean van der Merwe

2022/05/23

library(knitr)  
opts\_chunk$set(echo = TRUE, dev='win.metafile', fig.width=7, fig.height=4, out.width = "7in", out.height = "4in")  
st <- 'St2012345678'  
options(scipen = 9)

# Currently marking student St2012345678

# Instructions

Two sets of patients (individual subjects) are asked to rate their pain levels following a routine operation. They record the levels daily for 16 days. The first set receives a painkiller currently on the market, the second set receives a pain treatment that should be equivalent but cheaper and needs to be tested. Neither patients nor doctors know which patient receives which. Your goal is to determine whether the new treatment is as least 90% as effective as the one currently on the market.

The measurements are perceptions rated on a scale of 0 to 10.

Prior studies have suggested that the pain is sharp at first, dropping quickly to a more shallow pain that lingers and fades slowly. Suggested models include an exponential slope with priors and .

## Part 1

Read in and visualise your data. Is there an observable difference between the two treatments in the patterns observed?

*HINT* Convert the data to long format before removing the missing values: Create a new data set where the columns for the 16 days are converted to 2 columns: *Day* and *Score*. The other needed columns will have to be stretched to accommodate the longer form of the data. Consider using the *pivot\_longer* command.

## Part 2

A Binomial GLM formulation is recommended as a starting point. Ignore the time and subject effects for now, just assume i.i.d. observations first (conditional on treatment of course). What is the modelled treatment effect? Can you give a value for or something similar?

## Part 3

Bring in the subject effect. Consider each subject to have a random effect, but with constant variances throughout. How does this change your previous conclusions and results?

## Part 4

Bring in the time effect neatly, ideally assuming an exponential decay (fast drop then slow drop in pain levels) although a linear decay will work nearly as well. Which treatment is better after 1 day and which treatment is better after 10 days?

Incorporating the subject effect is not required for full marks, but will earn bonus marks, especially if you can answer the above question with uncertainty for a random future person.

## Part 5

Compare your previous models using at least one criterion and say which model seems the most parsimonious. The options (in order of preference) include LOOIC, WAIC, mDIC, cDIC (standard DIC). Properly calculated Bayes Factors would also get full credit.

**[Each of the 5 parts above is worth 20 marks. Total is out of 100.]**

# Generate samples that are different for each student

This is the code used to generate the data, for interest only.

library(openxlsx)  
students <- c('Sipho','Nhlanhla','Malethena','Tebello','Sandisile','Mari','Palesa','Edrich','Sihle','Lwazilwenkosi','Katlego','Walena','Jan','Bereng','St2012345678','St2123456789','St9876543210')  
nn <- length(students)  
n <- 120  
datasets <- vector('list',nn)  
Treatments <- c('T1', 'T2')  
muvec <- c(0.7, 0.5)  
svec <- c(0.2, 0.14)  
nmeasurements <- 16  
  
genperson <- function(mu, s, nt=nmeasurements) {  
 runif(1,mu,mu+0.2)\*exp(-runif(1,s-0.02, s+0.02)\*(seq\_len(nt)))  
}  
  
for (i in 1:nn) {  
 Tr <- sample(1:2, n, T)  
 Gr <- Treatments[Tr]  
 mus <- mapply(genperson, muvec[Tr], svec[Tr])  
 obs <- apply(mus, 1, \(props) rbinom(n, 10, props))  
 colnames(obs) <- paste0('Day', 1:nmeasurements)  
 IDnum <- cumsum(rbinom(n,6,0.5)+1)+100  
 ID <- gsub(' ','0',paste0('PID',format(IDnum,width = 6)), fixed = TRUE)  
 wmiss <- rbinom(n\*nmeasurements, 1, seq(0, 0.1, l=n\*nmeasurements))  
 obsholey <- obs  
 obsholey[wmiss==1] <- NA\_integer\_  
 datasets[[i]] <- data.frame(ID=ID, Treatment=Gr, TrNum=Tr, obsholey)  
}  
names(datasets) <- students  
write.xlsx(datasets, file = "BayesAssignmentSlopes.xlsx")  
# rm('d', 'students','nn','datasets','i','n','Treatments','muvec','svec','Tr','Gr','nmeasurements','mus','IDnum','ID','obs','wmiss','obsholey')

# Memorandum

## Part 1

library(openxlsx)  
mydata <- read.xlsx('BayesAssignmentSlopes.xlsx', st)  
names(mydata)

## [1] "ID" "Treatment" "TrNum" "Day1" "Day2" "Day3"   
## [7] "Day4" "Day5" "Day6" "Day7" "Day8" "Day9"   
## [13] "Day10" "Day11" "Day12" "Day13" "Day14" "Day15"   
## [19] "Day16"

nwide <- nrow(mydata)  
plot(c(1,16), c(0,10), type='n', xlab='Day 📅', ylab='⬅😀 Score 😣➡', main='') # Emojis only work on R version 4.2.0+

# plot(c(1,16), c(0,10), type='n', xlab='Day', ylab='Score', main='') # Use this line on earlier versions  
cols <- c('red', 'blue')  
for (i in seq\_len(nwide)) {  
 lines(1:16, mydata[i,4:19], col=cols[mydata$TrNum[i]], lty=mydata$TrNum[i])  
}



library(tidyverse)  
mydata |> pivot\_longer(starts\_with("Day"), names\_to = "Day", values\_to = "Score") |>  
 na.omit() |> mutate(DayNum = as.numeric(gsub("Day", "", Day))) -> d

d |> ggplot(aes(x = DayNum, y = Score, colour = Treatment, group = ID)) + geom\_line()



In this case the first treatment appears to have lower scores in the long run (Good), but not so much on Day 1.

**|| 5 Marks for reading in the data correctly, 5 for visualising it, 5 for saying what is visible, 5 for transforming the shape. ||**

## Part 2

library(parallel)  
library(rstan)  
mycores <- max(1,floor(detectCores(logical = FALSE)\*0.75))  
options(mc.cores = mycores)  
rstan\_options(auto\_write = TRUE)

// This Stan block defines a Binomial model, by Sean van der Merwe, UFS  
data {  
 int<lower=1> n; // number of observations  
 int<lower=1> mx; // Binomial upper limit  
 int<lower=0, upper=mx> y[n]; // observations  
 int<lower=1> ng; // number of groups  
 int<lower=1, upper=ng> g[n]; // group membership  
}  
// The parameters of the model  
parameters {  
 real alpha[ng]; // group intercepts  
}  
model {  
for (i in 1:n) {  
 y[i] ~ binomial\_logit(mx, alpha[g[i]]); // fit the data pattern  
}  
}  
generated quantities {  
 vector[n] log\_lik;  
 for (i in 1:n) {  
 log\_lik[i] = binomial\_logit\_lpmf(y[i] | mx, alpha[g[i]]);  
 }  
}

saveRDS(Part2, file = 'Part2.Rds')

list(n=nrow(d), y=d$Score, ng=max(d$TrNum), g=d$TrNum, mx=10) -> stan\_data  
Part2 |> sampling(stan\_data, pars=c('alpha', 'log\_lik'), iter = 20000, chains = mycores) -> ModelFitP2  
ModelFitP2 |> extract() -> draws  
(ModelFitP2 |> summary(pars = 'alpha'))$summary |> kable(digits = 3)

|  | mean | se\_mean | sd | 2.5% | 25% | 50% | 75% | 97.5% | n\_eff | Rhat |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| alpha[1] | -1.270 | 0 | 0.027 | -1.323 | -1.288 | -1.270 | -1.252 | -1.218 | 32093.42 | 1 |
| alpha[2] | -1.229 | 0 | 0.023 | -1.275 | -1.245 | -1.229 | -1.213 | -1.184 | 32602.35 | 1 |

(draws$alpha |> plogis())\*10 -> scores  
1 - (draws$alpha |> plogis()) -> effectiveness  
targetprob <- mean((effectiveness[,1]/effectiveness[,2] > 0.9) & (effectiveness[,1]/effectiveness[,2] < 1.1))

The probability of Treatment 2 being as effective as Treatment 1, with 10% margin, is about 1.

Summary of expected scores over the two treatments:

kable(summary(data.frame(Treatment1ave=scores[,1], Treatment2ave=scores[,2])))

|  | Treatment1ave | Treatment2ave |
| --- | --- | --- |
|  | Min. :1.998 | Min. :2.106 |
|  | 1st Qu.:2.162 | 1st Qu.:2.236 |
|  | Median :2.192 | Median :2.264 |
|  | Mean :2.193 | Mean :2.264 |
|  | 3rd Qu.:2.224 | 3rd Qu.:2.291 |
|  | Max. :2.364 | Max. :2.427 |

**|| 10 marks for the model, 5 for running it correctly, and 5 for answering the question. ||**

## Part 3

// This Stan block defines a Binomial model, by Sean van der Merwe, UFS  
data {  
 int<lower=1> n; // number of observations  
 int<lower=1> mx; // Binomial upper limit  
 int<lower=0, upper=mx> y[n]; // observations  
 int<lower=1> ng; // number of groups  
 int<lower=1, upper=ng> g[n]; // group membership  
 int<lower=1> ns; // number of subjects  
 int<lower=1, upper=ns> s[n]; // subject membership  
}  
// The parameters of the model  
parameters {  
 real alpha[ng]; // group intercepts  
 real sbj[ns]; // subject effects  
 real<lower = 0> sbjsd; // between subject variation  
}  
model {  
 for (i in 1:n) {  
 y[i] ~ binomial\_logit(mx, alpha[g[i]] + sbj[s[i]]); // fit the data pattern  
 }  
 sbj ~ normal(0, sbjsd);  
 sbjsd ~ cauchy(0, 2.5);  
}  
generated quantities {  
 vector[n] log\_lik;  
 for (i in 1:n) {  
 log\_lik[i] = binomial\_logit\_lpmf(y[i] | mx, alpha[g[i]] + sbj[s[i]]);  
 }  
}

saveRDS(Part3, file = 'Part3.Rds')

d$IDnum <- as.numeric(factor(d$ID, levels = unique(d$ID)))  
list(n=nrow(d), y=d$Score, ng=max(d$TrNum), g=d$TrNum, mx=10, ns = max(d$IDnum), s = d$IDnum) -> stan\_data  
Part3 |> sampling(stan\_data, pars=c('alpha', 'log\_lik'), iter = 20000, chains = mycores) -> ModelFitP3

## Warning: There were 4 chains where the estimated Bayesian Fraction of Missing Information was low. See  
## https://mc-stan.org/misc/warnings.html#bfmi-low

## Warning: Examine the pairs() plot to diagnose sampling problems

## Warning: Tail Effective Samples Size (ESS) is too low, indicating posterior variances and tail quantiles may be unreliable.  
## Running the chains for more iterations may help. See  
## https://mc-stan.org/misc/warnings.html#tail-ess

ModelFitP3 |> extract() -> draws  
(ModelFitP3 |> summary(pars = 'alpha'))$summary |> kable(digits = 3)

|  | mean | se\_mean | sd | 2.5% | 25% | 50% | 75% | 97.5% | n\_eff | Rhat |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| alpha[1] | -1.273 | 0 | 0.031 | -1.333 | -1.293 | -1.272 | -1.252 | -1.213 | 34549.00 | 1 |
| alpha[2] | -1.231 | 0 | 0.027 | -1.285 | -1.249 | -1.231 | -1.213 | -1.179 | 27569.02 | 1 |

(draws$alpha |> plogis())\*10 -> scores  
1 - (draws$alpha |> plogis()) -> effectiveness  
targetprob <- mean((effectiveness[,1]/effectiveness[,2] > 0.9) & (effectiveness[,1]/effectiveness[,2] < 1.1))

The probability of Treatment 2 being as effective as Treatment 1, with 10% margin, is about 1.

Summary of expected scores over the two treatments:

kable(summary(data.frame(Treatment1ave=scores[,1], Treatment2ave=scores[,2])))

|  | Treatment1ave | Treatment2ave |
| --- | --- | --- |
|  | Min. :1.984 | Min. :2.075 |
|  | 1st Qu.:2.154 | 1st Qu.:2.228 |
|  | Median :2.189 | Median :2.260 |
|  | Mean :2.189 | Mean :2.260 |
|  | 3rd Qu.:2.224 | 3rd Qu.:2.291 |
|  | Max. :2.386 | Max. :2.468 |

**|| 10 marks for the model, 5 for running it correctly, and 5 for answering the question. ||**

## Part 4

// This Stan block defines a Binomial model, by Sean van der Merwe, UFS  
data {  
 int<lower=1> n; // number of observations  
 int<lower=1> mx; // Binomial upper limit  
 int<lower=0, upper=mx> y[n]; // observations  
 int<lower=1> ng; // number of groups  
 int<lower=1, upper=ng> g[n]; // group membership  
 real day[n]; // time points  
}  
// The parameters of the model  
parameters {  
 real<lower=0, upper=1> mu[ng]; // group intercepts  
 real<lower=0, upper=1> lambda[ng]; // group slopes  
}  
transformed parameters {  
 real<lower=0, upper=1> alpha[n]; // expected values  
 for (i in 1:n) {  
 alpha[i] = exp(log(mu[g[i]]) - day[i]\*lambda[g[i]]);  
 }  
}  
model {  
 mu ~ beta(5.5, 2.5);  
 lambda ~ beta(1.5, 5.5);  
 for (i in 1:n) {  
 y[i] ~ binomial(mx, alpha[i]); // fit the data pattern  
 }  
}  
generated quantities {  
 vector[n] log\_lik;  
 for (i in 1:n) {  
 log\_lik[i] = binomial\_lpmf(y[i] | mx, alpha[i]);  
 }  
}

// This Stan block defines a Binomial model, by Sean van der Merwe, UFS  
data {  
 int<lower=1> n; // number of observations  
 int<lower=1> mx; // Binomial upper limit  
 int<lower=0, upper=mx> y[n]; // observations  
 int<lower=1> ng; // number of groups  
 int<lower=1, upper=ng> g[n]; // group membership  
 int<lower=1> ns; // number of subjects  
 int<lower=1, upper=ns> s[n]; // subject membership  
 real day[n]; // time points  
}  
// The parameters of the model  
parameters {  
 real<lower=0, upper=1> mu[ng]; // group intercepts  
 real<lower=0, upper=1> lambda[ng]; // group slopes  
 vector[2] sbj[ns]; // subject effects  
 corr\_matrix[2] sigmamat; // between subject correlation  
 vector<lower=0>[2] sbjsd; // between subject variation  
}  
transformed parameters {  
 real<lower=0, upper=1> alpha[n]; // expected values  
 for (i in 1:n) {  
 alpha[i] = exp(log(mu[g[i]]) + sbj[1, s[i]] - day[i]\*lambda[g[i]]\*exp(sbj[2, s[i]]));  
 }  
}  
model {  
 mu ~ beta(5.5, 2.5);  
 lambda ~ beta(1.5, 5.5);  
 sbjsd ~ uniform(0, 0.001);  
 sigmamat ~ lkj\_corr(2);  
 sbj ~ multi\_normal(zeros\_vector(2), quad\_form\_diag(sigmamat, sbjsd));  
 for (i in 1:n) {  
 y[i] ~ binomial(mx, alpha[i]); // fit the data pattern  
 }  
}  
generated quantities {  
 vector[n] log\_lik;  
 for (i in 1:n) {  
 log\_lik[i] = binomial\_lpmf(y[i] | mx, alpha[i]);  
 }  
}

saveRDS(Part4simple, file = 'Part4simple.Rds')  
saveRDS(Part4, file = 'Part4.Rds')

d$IDnum <- as.numeric(factor(d$ID, levels = unique(d$ID)))  
list(n=nrow(d), y=d$Score, ng=max(d$TrNum), g=d$TrNum, mx=10, ns = max(d$IDnum), s = d$IDnum, day = d$DayNum) -> stan\_data  
Part4 |> sampling(stan\_data, iter = 20000, chains = mycores) -> ModelFitP4  
ModelFitP4 |> extract() -> draws  
(ModelFitP4 |> summary(pars = c('mu', 'lambda')))$summary |> kable(digits = 3)

|  | mean | se\_mean | sd | 2.5% | 25% | 50% | 75% | 97.5% | n\_eff | Rhat |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| mu[1] | 0.820 | 0 | 0.021 | 0.778 | 0.806 | 0.820 | 0.834 | 0.862 | 21678.90 | 1 |
| mu[2] | 0.597 | 0 | 0.016 | 0.566 | 0.587 | 0.597 | 0.608 | 0.630 | 20675.78 | 1 |
| lambda[1] | 0.209 | 0 | 0.006 | 0.198 | 0.205 | 0.208 | 0.212 | 0.219 | 22297.89 | 1 |
| lambda[2] | 0.139 | 0 | 0.004 | 0.131 | 0.137 | 0.139 | 0.142 | 0.148 | 20296.58 | 1 |

day\_pred <- function(mu, lambda, daynum) {  
 mu\*exp(-daynum\*lambda)  
}  
nsims <- nrow(draws$mu)  
preds\_day1 <- data.frame(Treatment = rep(c('T1', 'T2'), each = nsims), Prediction = c(day\_pred(draws$mu[,1], draws$lambda[,1], 1), day\_pred(draws$mu[,2], draws$lambda[,2], 1)))  
preds\_day10 <- data.frame(Treatment = rep(c('T1', 'T2'), each = nsims), Prediction = c(day\_pred(draws$mu[,1], draws$lambda[,1], 10), day\_pred(draws$mu[,2], draws$lambda[,2], 10)))

Predictions at Day 1:

preds\_day1 |> ggplot(aes(x = Treatment, y = Prediction, colour = Treatment)) + geom\_boxplot()



Predictions at Day 10:

preds\_day10 |> ggplot(aes(x = Treatment, y = Prediction, colour = Treatment)) + geom\_boxplot()



Clearly Treatment 1 is worse at Day 1 and better at Day 10.

**|| 10 marks for the model, 10 for answering the question. ||**

## Part 5

fits <- list(IIDmodel = ModelFitP2, REmodel = ModelFitP3, EDmodel = ModelFitP4)

Here the models are compared using cross-validated information criteria.

library(loo)  
fits |> lapply(\(fit) {extract\_log\_lik(fit, merge\_chains = FALSE)}) -> log\_lik  
log\_lik |> lapply(\(ll) {relative\_eff(exp(ll), cores = mycores)}) -> r\_eff  
fits |> length() |> seq\_len() |>   
 lapply(\(i) {loo(log\_lik[[i]], r\_eff = r\_eff[[i]], cores = mycores)}) |>   
 loo\_compare() -> comparison  
rownames(comparison) <- names(fits)[order(rownames(comparison))]  
comparison |> kable(digits = 1)

|  | elpd\_diff | se\_diff | elpd\_loo | se\_elpd\_loo | p\_loo | se\_p\_loo | looic | se\_looic |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| REmodel | 0.0 | 0.0 | -2764.3 | 34.0 | 4.2 | 0.2 | 5528.5 | 68.0 |
| EDmodel | -1393.9 | 62.9 | -4158.2 | 62.2 | 5.0 | 0.2 | 8316.4 | 124.5 |
| IIDmodel | -1411.2 | 63.5 | -4175.4 | 62.7 | 69.2 | 2.4 | 8350.9 | 125.4 |

The models with the subject random effect are better. Philosophically it better matches what we are trying to achieve in terms of a random future patient. Introducing the slopes correctly should also help.

**|| 10 marks for correctly bringing in the log likelihood generation and export OR 10 marks for implementing DIC calculations for both models. 10 marks for doing and explaining the comparison. ||**