HW 8

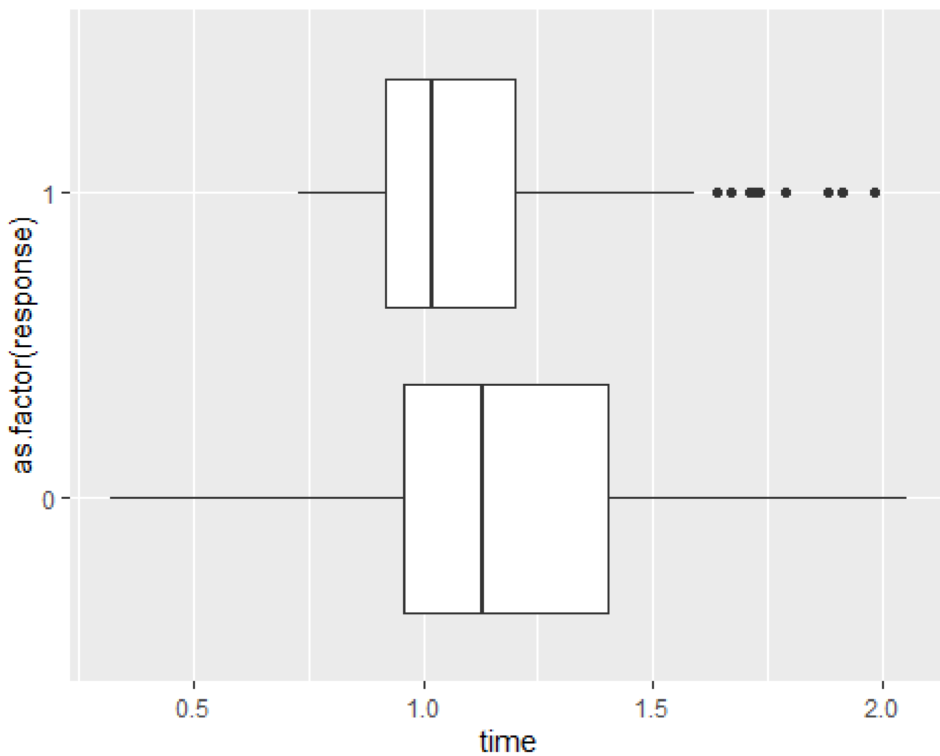
Stat 451

3/22/2021

For this assignment you will use the data file 'outside.dat'. In this file there are two columns, time and response. The response is a binary variable indicating whether or not a set in volleyball resulted in a kill. The time is the elapsed time in seconds the ball was in the air after leaving the setter's hands until being struck by the outside hitter. The question is whether there is a relationship between set speed (i.e. time the ball was in the air) and probability of the set resulting in a kill. The problem should be done as a linear logistic regression.

Likelihood will be bernouli

```
outside <- read.table("outside.dat", header = TRUE)
dim(outside)

## [1] 718   2

head(outside)

##   time response
## 1 1.25        1
## 2 1.20        1
## 3 0.74        0
## 4 1.43        1
## 5 0.89        1
## 6 1.04        1

library(R2jags)

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.0.4

ggplot(data = outside, mapping = aes(x= time, y = as.factor(response))) +
  geom_boxplot()
```



1. Write the code to analyze the data in JAGS.

```
mdl <- "
  model {

    for(i in 1:718){
      response[i] ~ dbern(p[i])
      logit(p[i]) <- b0 + b1*time[i]
    }
     # Prior Values
      b1 ~ dnorm(0, 0.001)
      b0 ~ dnorm(0, 0.001)
  }
"

# Reponse is binomial
# The effects are on the logit scale

writeLines(mdl, 'logistic.txt')
response <- outside$response
time <- outside$time

data.jags <- c('time', "response")
parms <- c('b0','b1')
logistic.sim <- jags(data = data.jags, inits = NULL, parameters.to.save =
parms,
                     model.file = 'logistic.txt', n.iter = 12000, n.burnin =
```

```
2000,
                     n.chains = 8, n.thin = 10)
```

```
## module glm loaded
```

```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 718
##    Unobserved stochastic nodes: 2
##    Total graph size: 1791
##
## Initializing model
```

```
logistic.sim
```

```
## Inference for Bugs model at "logistic.txt", fit using jags,
##  8 chains, each with 12000 iterations (first 2000 discarded), n.thin = 10
##  n.sims = 8000 iterations saved
##          mu.vect sd.vect    2.5%     25%     50%     75%   97.5%  Rhat
n.eff
## b0         1.795   0.355   1.100   1.550   1.795   2.029   2.517 1.001
5300
## b1        -1.699   0.310  -2.330  -1.904  -1.697  -1.485  -1.095 1.001
4500
## deviance 961.872   2.011 959.912 960.431 961.228 962.673 967.271 1.001
8000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 2.0 and DIC = 963.9
## DIC is an estimate of expected predictive error (lower deviance is
better).
```

2. Show conversion diagnostics. At the very least the Raftery-Lewis and the effective sample size.

Diagnostics are good.

```
sims <- as.mcmc(logistic.sim)
chains <- as.matrix(sims)
raftery.diag(chains)
```

```
##
## Quantile (q) = 0.025
## Accuracy (r) = +/- 0.005
## Probability (s) = 0.95
##
```

```
##             Burn-in  Total Lower bound  Dependence
##             (M)      (N)   (Nmin)       factor (I)
##   b0        2        3693  3746         0.986
##   b1        2        3710  3746         0.990
##   deviance 1         3747  3746         1.000
```

```
effectiveSize(chains)
```

```
##       b0       b1 deviance
## 7729.763 7693.461 8000.000
```

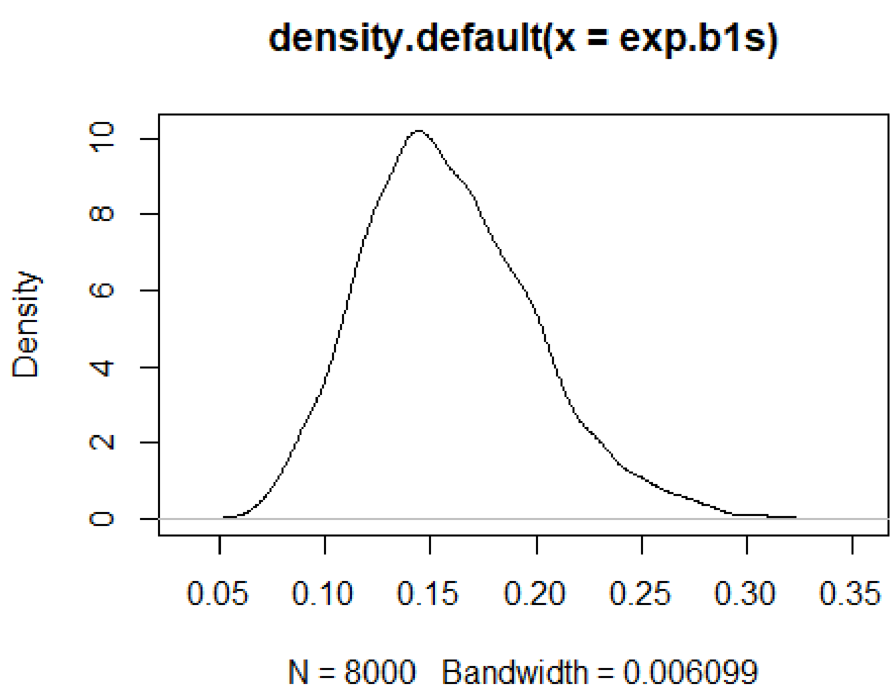
3. Is there a significant relationship between time and the probability of a kill? How do you know? What are the $\hat{\beta}$'s?

Since the 95% posterior credible interval does not span 0 then there is a significant effect. Yes there is a significant relationship between time and probability of a kill. A density plot of the changes in probability associated with a 1 second time change does not contain 0. So I conclude that it is significant.

```
exp(-1.665)/(1+exp(-1.665))
```

```
## [1] 0.1590919
```

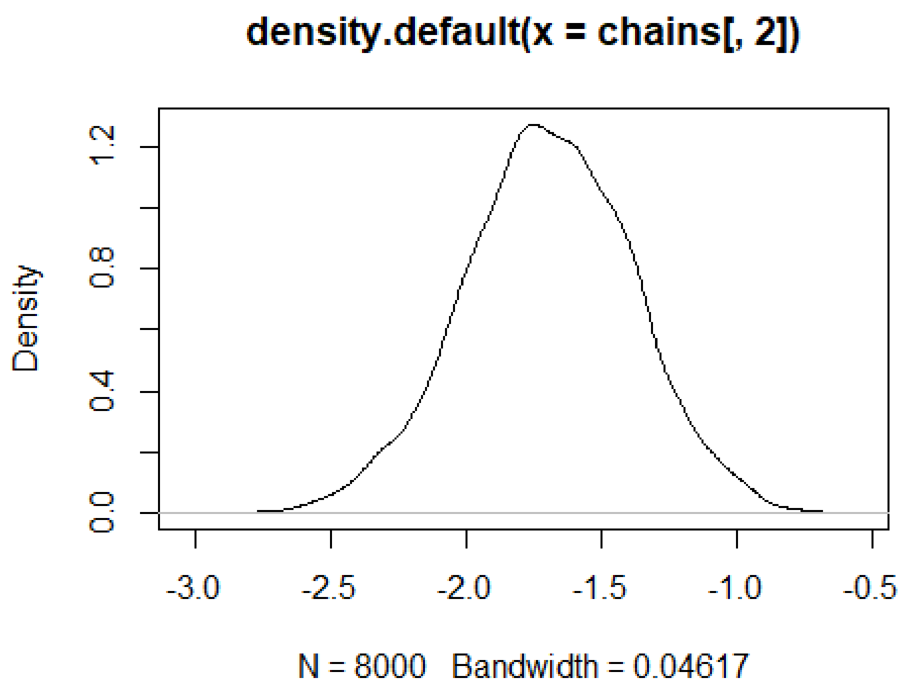
```
exp.b1s <- exp(chains[,2])/(1+exp(chains[,2]))

plot(density(exp.b1s))
```




```
colnames(chains)
```

```
## [1] "b0"       "b1"       "deviance"
```

```
plot(density(chains[,2]))
```



4. Produce a plot with probability of a kill on the y-axis and time on the x-axis. Show the best fit line.

```
min(outside$time)
```

```
## [1] 0.32
```

```
max(outside$time)
```

```
## [1] 2.05
```

```
plot(outside$time, outside$time*-0.159)
```

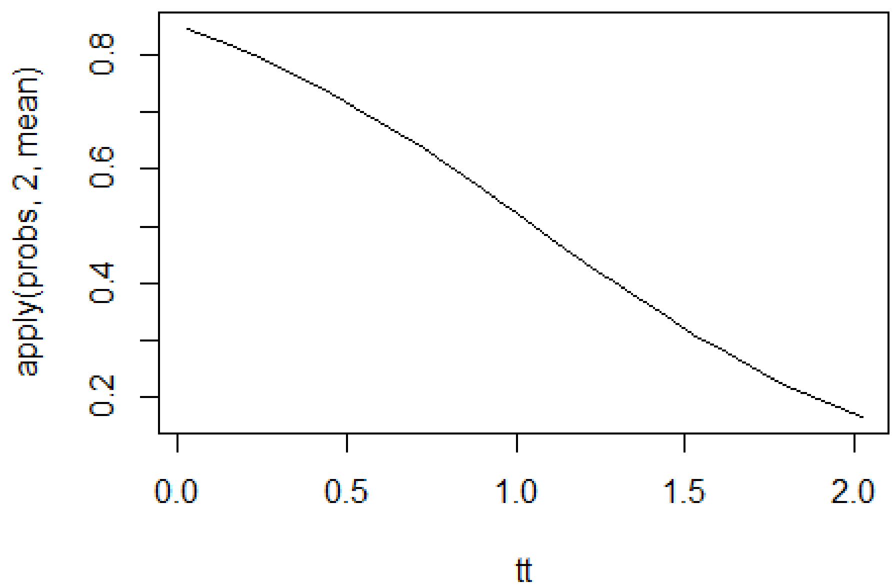
```
p1 <- exp(chains[,1] + chains[,2])/(exp(1+chains[,1]+chains[,2]))
p1 <- chains[,1] + chains[,2]
prob1 <- exp(p1)/(1+exp(p1))
quantile(p1,c(0.25,0.975))
```

```
##        25%      97.5%
## 0.03966186 0.26594924
```

```
length(prob1)
```

```
## [1] 8000
```

```r
tt <- seq(0.025, 2.1, by = .1)
length(tt)

## [1] 21

probs <- matrix(0,8000,21)
for(i in 1:21){
  probs[,i] <- exp(chains[,1]+chains[,2]*
tt[i])/(1+exp(chains[,1]+chains[,2]*tt[i]))
}

length(tt)

## [1] 21

plot(tt, apply(probs,2,mean), type = 'l')
```



5. Add the 95% posterior probability intervals to the plot in the previous problem.

```r
exp.b1s <- exp(chains[,2])/(1+exp(chains[,2]))
min(outside$time)

## [1] 0.32

max(outside$time)

## [1] 2.05

plot(outside$time, outside$time*-0.159)
```

```r
p1 <- exp(chains[,1] + chains[,2])/(exp(1+chains[,1]+chains[,2]))
p1 <- chains[,1] + chains[,2]
prob1 <- exp(p1)/(1+exp(p1))
quantile(p1,c(0.25,0.975))
```

```
##        25%      97.5%
## 0.03966186 0.26594924
```

```r
length(prob1)
```

```
## [1] 8000
```

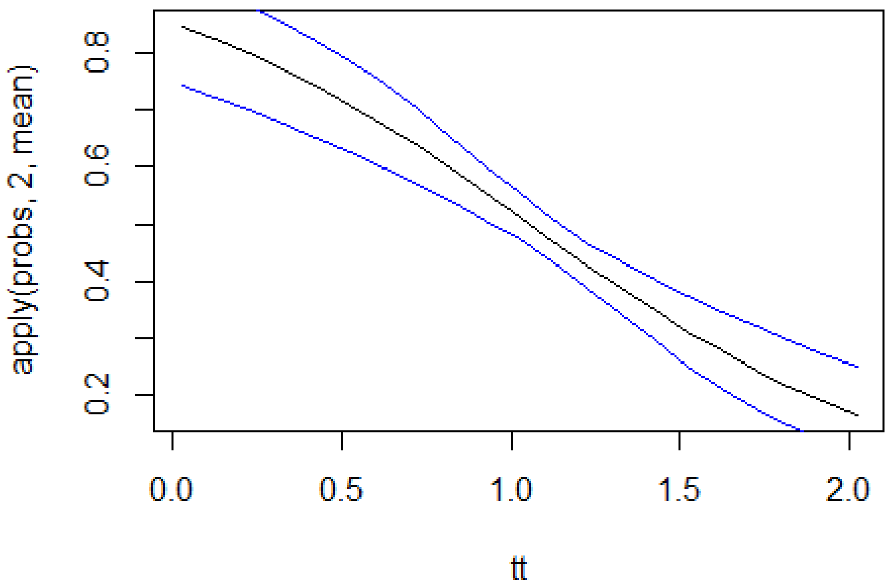
```r
tt <- seq(0.025, 2.1, by = .1)
length(tt)
```

```
## [1] 21
```

```r
probs <- matrix(0,8000,21)
for(i in 1:21){
  probs[,i] <- exp(chains[,1]+chains[,2]*
tt[i])/(1+exp(chains[,1]+chains[,2]*tt[i]))
}

length(tt)
```

```
## [1] 21
```

```r
plot(tt, apply(probs,2,mean), type = 'l')
lines(tt,apply(probs,2,quantile,0.025), col = "blue")
lines(tt,apply(probs,2,quantile,0.975), col = "blue")
```



6. Now write the code to analyze the problem in Stan. Compare the β's you got in JAGS to the β's you got in Stan. Provide plots of the posteriors for the β_0's on one set of axes. Provide plots of the posteriors for the β_1's on one set of axes.

The effects for intercept and slope are almost the same as the model in jags.

```
head(outside)

##   time response
## 1 1.25        1
## 2 1.20        1
## 3 0.74        0
## 4 1.43        1
## 5 0.89        1
## 6 1.04        1

dim(outside)

## [1] 718   2

N <- nrow(outside)
x <- outside$time
y <- outside$response
out_dat <- list(N=N, y = y, x = x)
library(rstan)

## Warning: package 'rstan' was built under R version 4.0.3
```

```
## Loading required package: StanHeaders

## Warning: package 'StanHeaders' was built under R version 4.0.3

## 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)

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

##
## Attaching package: 'rstan'

## The following object is masked from 'package:R2jags':
##
##     traceplot

## The following object is masked from 'package:coda':
##
##     traceplot
```

```
logistic_stan.sim <- stan(file = "outside.stan", data = out_dat, iter = 5000,
              warmup = 1000, chains = 1, thin = 2)
```

```
##
## SAMPLING FOR MODEL 'outside' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would
take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 5000 [  0%]  (Warmup)
## Chain 1: Iteration:  500 / 5000 [ 10%]  (Warmup)
## Chain 1: Iteration: 1000 / 5000 [ 20%]  (Warmup)
## Chain 1: Iteration: 1001 / 5000 [ 20%]  (Sampling)
## Chain 1: Iteration: 1500 / 5000 [ 30%]  (Sampling)
## Chain 1: Iteration: 2000 / 5000 [ 40%]  (Sampling)
## Chain 1: Iteration: 2500 / 5000 [ 50%]  (Sampling)
## Chain 1: Iteration: 3000 / 5000 [ 60%]  (Sampling)
## Chain 1: Iteration: 3500 / 5000 [ 70%]  (Sampling)
## Chain 1: Iteration: 4000 / 5000 [ 80%]  (Sampling)
## Chain 1: Iteration: 4500 / 5000 [ 90%]  (Sampling)
## Chain 1: Iteration: 5000 / 5000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 1.207 seconds (Warm-up)
## Chain 1:                3.588 seconds (Sampling)
```

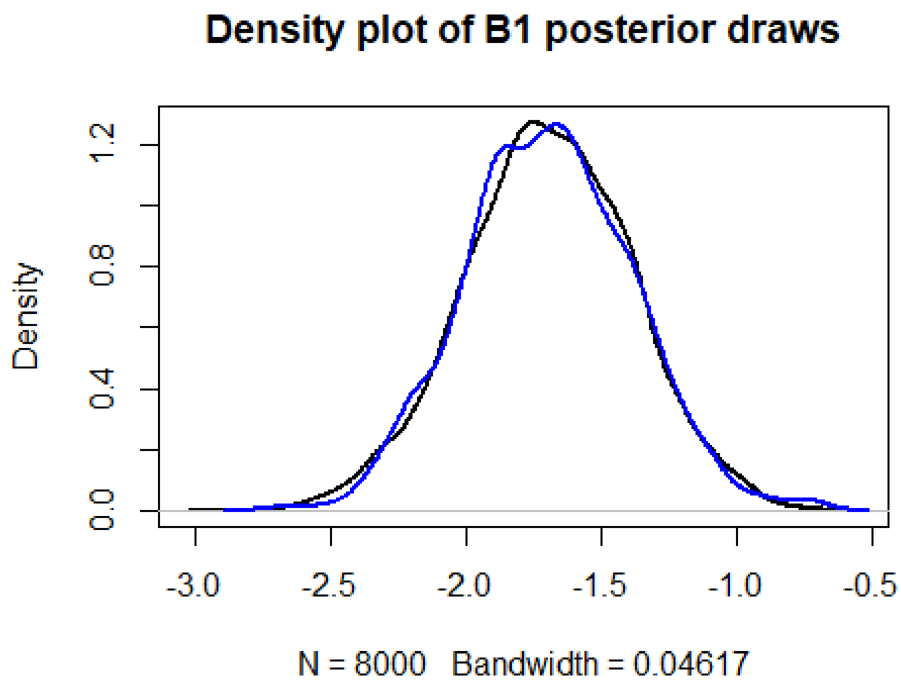
```
## Chain 1:                4.795 seconds (Total)
## Chain 1:
```

```
logistic_stan.sim
```

```
## Inference for Stan model: outside.
## 1 chains, each with iter=5000; warmup=1000; thin=2;
## post-warmup draws per chain=2000, total post-warmup draws=2000.
##
##          mean se_mean   sd    2.5%     25%     50%     75%   97.5% n_eff
Rhat
## alpha    1.79    0.01 0.35    1.12    1.55    1.79    2.03    2.44   622
1
## beta    -1.70    0.01 0.31   -2.28   -1.90   -1.70   -1.49   -1.10   620
1
## lp__  -480.96    0.05 1.02 -483.79 -481.33 -480.69 -480.24 -479.96   489
1
##
## Samples were drawn using NUTS(diag_e) at Mon Apr 05 15:00:58 2021.
## 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).
```

Density Plots

```
sims2 <- as.matrix(logistic_stan.sim)
chains2 <- as.mcmc(sims2)


plot(density(chains[,2]), main = "Density plot of B1 posterior draws", lwd =
2)
lines(density(chains2[,2]), col = "blue", lwd = 2)
```




```r
plot(density(chains[,1]), main = "Density plot of B0 posterior draws", lwd =
2)
lines(density(chains2[,1]), col = "blue", lwd = 2)
```