Problem Set 1

Sungsik Hwang

February 10, 2025

1. It's only three days into the Major League Blernsball season, but fans are already excited about rookie sensation Casey Lee of the Boston Poindexters. Lee hit a grand slam blern in the first game of the Poindexters' season, none in the second game, and three in the third game. Statistician Jack Johnson sets out to estimate the rate at which Lee hits grand slam blerns. Johnson assumes that the number of gland slam blerns that Johnson hits in a game follows a Poisson distribution with unknown rate parameter lambda and that these observations are exchangeable. Since no one had seen Lee play before the start of the season, Johnson uses a flat prior on the rate parameter. Using the results of Lee's first three games, estimate this model in Stan with a flat prior on lambda. Please run the model with 2 chains and at least 10,000 iterations per chain (i.e., use the options chains=2, iter=10000 when running sampling). Note: To implement a flat prior in Stan, you can simply omit the prior from the model section entirely. This implicitly gives a prior density proportional to one (or log prior density of zero), which is a flat prior

```r
library(rstan)

y <- c(1, 0, 3)
n <- length(y)

Model <- "
data {
  int<lower=0> n;
  int<lower=0> y[n];
}
parameters {
  real<lower=0> lambda;
}
model {
    y ~ poisson(lambda);
}
"
fit <- stan(
  model_code = Model,
  data       = list(n = n, y = y),
  chains     = 2,
  iter       = 10000,
  seed       = 12345
)
```

```
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 3.6e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.36 seconds.
## Chain 1: Adjust your expectations accordingly!
```

```
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 1: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 1: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 1: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 1: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 1: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 1: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 1: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 1: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 1: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 1: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 1: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.045 seconds (Warm-up)
## Chain 1:                0.045 seconds (Sampling)
## Chain 1:                0.09 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 4e-06 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.04 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 2: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 2: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 2: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 2: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 2: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 2: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 2: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 2: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 2: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 2: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 2: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.045 seconds (Warm-up)
## Chain 2:                0.048 seconds (Sampling)
## Chain 2:                0.093 seconds (Total)
## Chain 2:
```

##(a) Give an estimate of the rate at which Lee hits grand slam blerns using the Bayes estimator of lambda under quadratic loss.

```r
results <- rstan::extract(fit, "lambda")$lambda
Mean<-mean(results)
Mean
```

```
## [1] 1.652879
```

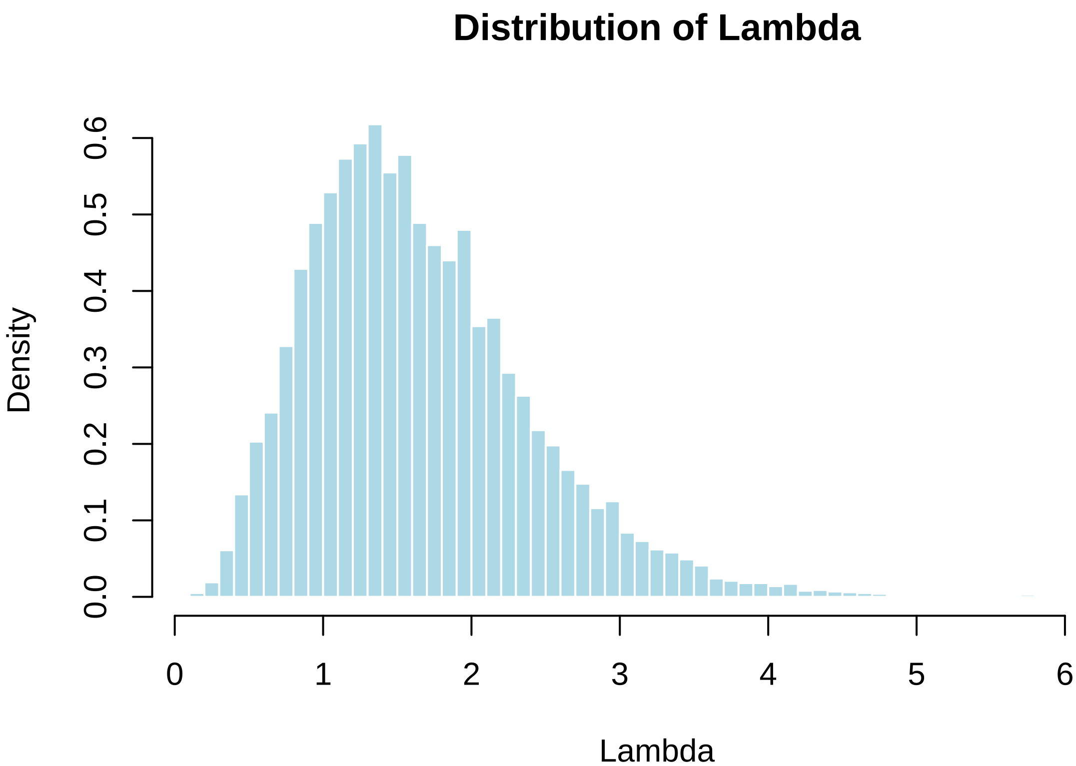
##(b) Give a 95% credible interval for lambda.

```r
CI <- quantile(results, probs = c(0.025, 0.975))
CI
```

```
##      2.5%     97.5%
## 0.5236603 3.4076202
```

##(c) Plot the posterior distribution of lambda (as approximated by a histogram or kernel density plot of the samples)

```r
hist(
  results,
  breaks = 50,
  main = "Distribution of Lambda",
  xlab = "Lambda",
  col = "lightblue",
  border = "white",
  freq = FALSE
)
```



##(d) The Poindexters are hoping that Lee is that rare blernsball player that averages a rate of one grand slam blerns per game or more. Give the posterior probability that this is the case (i.e., lambda > 1).

```r
prob <- mean(results > 1)
prob
```

```
## [1] 0.8091
```

2. Johnson's bitter rival, John Jackson, criticizes Johnson's choice of prior. While Jackson agrees with Johnson's model, Jackson argues that the appropriate prior on lambda should be based on the distribution of the grand-slam-blerns rate exhibited by other blernsball players. This distribution is similar to a Weibull distribution with a shape parameter of 1.6 and a scale parameter of 0.3. Estimate this same model in Stan using Jackson's Weibull in place of the Johnson's uniform prior.

```r
Model2 <- "
data {
  int<lower=0> n;
  int<lower=0> y[n];
}
parameters {
  real<lower=0> lambda;
}
model {
  lambda ~ weibull(1.6, 0.3);
   y ~ poisson(lambda);
}
"

fit2 <- stan(
  model_code = Model2,
  data       = list(n = n, y = y),
  chains     = 2,
  iter       = 10000,
  seed       = 12345
)
```

```
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 3.4e-05 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0.34 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 1: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 1: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 1: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 1: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 1: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 1: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 1: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 1: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 1: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 1: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 1: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 0.05 seconds (Warm-up)
## Chain 1:                0.049 seconds (Sampling)
## Chain 1:                0.099 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'anon_model' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 4e-06 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 0.04 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
```

```
## Chain 2:
## Chain 2: Iteration:    1 / 10000 [  0%]  (Warmup)
## Chain 2: Iteration: 1000 / 10000 [ 10%]  (Warmup)
## Chain 2: Iteration: 2000 / 10000 [ 20%]  (Warmup)
## Chain 2: Iteration: 3000 / 10000 [ 30%]  (Warmup)
## Chain 2: Iteration: 4000 / 10000 [ 40%]  (Warmup)
## Chain 2: Iteration: 5000 / 10000 [ 50%]  (Warmup)
## Chain 2: Iteration: 5001 / 10000 [ 50%]  (Sampling)
## Chain 2: Iteration: 6000 / 10000 [ 60%]  (Sampling)
## Chain 2: Iteration: 7000 / 10000 [ 70%]  (Sampling)
## Chain 2: Iteration: 8000 / 10000 [ 80%]  (Sampling)
## Chain 2: Iteration: 9000 / 10000 [ 90%]  (Sampling)
## Chain 2: Iteration: 10000 / 10000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 0.049 seconds (Warm-up)
## Chain 2:                0.045 seconds (Sampling)
## Chain 2:                0.094 seconds (Total)
## Chain 2:
```

```
results2 <- rstan::extract(fit2, "lambda")$lambda
```

##(a) Give an estimate of the rate at which Lee hits grand slam blerns using the Bayes estimator of lambda under quadratic loss.

```
Mean<-mean(results2)
Mean
```

```
## [1] 0.5160787
```

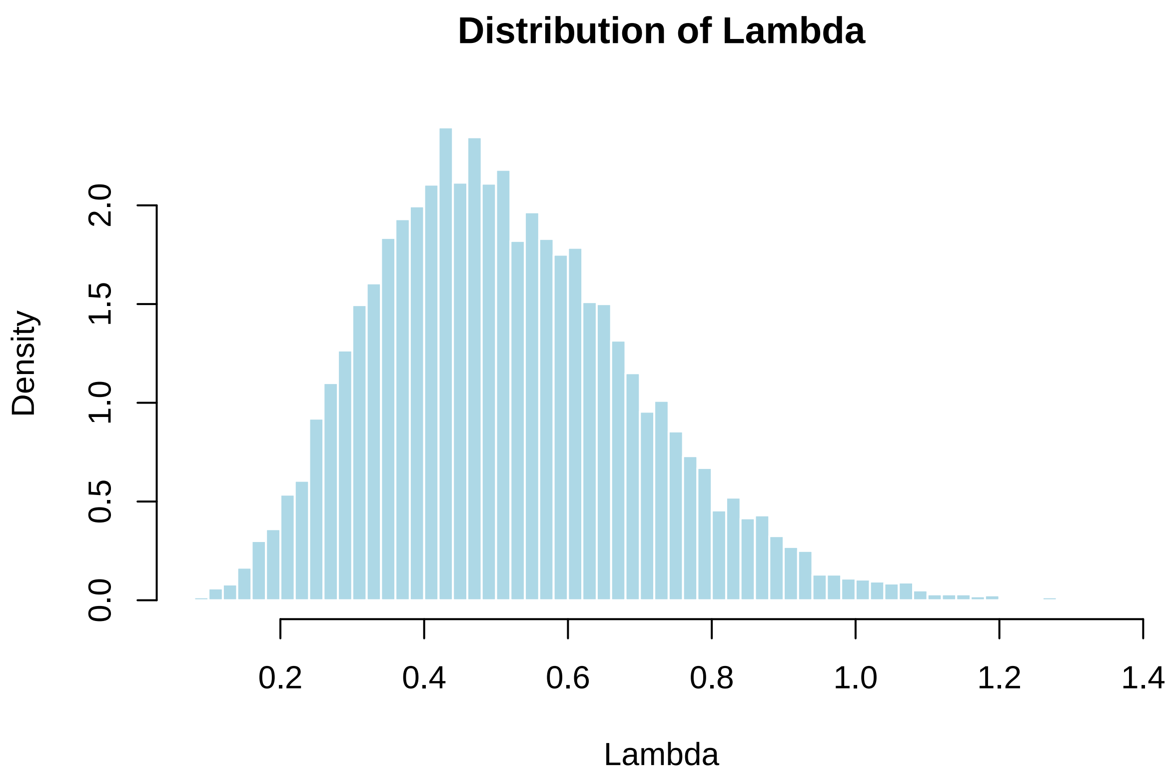
##(b) Give a 95% credible interval for lambda.

```
CI <- quantile(results2, probs = c(0.025, 0.975))
CI
```

```
##      2.5%     97.5%
## 0.2117950 0.9183176
```

##(c) Plot the posterior distribution of lambda (as approximated by a histogram or kernel density plot of the samples).

```
hist(
  results2,
  breaks = 50,
  main = "Distribution of Lambda",
  xlab = "Lambda",
  col = "lightblue",
  border = "white",
  freq = FALSE
)
```



##(d)

Give the posterior probability that Lee hits grand slam blerns at a rate of one per game or more.

```
prob <- mean(results2 > 1)
prob
```

```
## [1] 0.0121
```