Reliability

Josh Christensen, Mitchell Wassom, Ryan Whetten

April 2021

1 Introduction and Problem Background

Learning Suite is BYU's course management system which supports online learning and teaching. It allows professors to post grades, information, and assignments online for students to view. If a professor permits, students can use Learning Suite to view performance statistics with a histogram showing the class's grades on any given assignment.

Learning Suite is currently testing a beta version with improvements in its software. While using the beta version of Learning Suite, we noticed that sometimes statistics that displayed properly in the original version of Learning Suite failed to display in the beta version. The goal of this project is to develop a model that could lead to some insights as to why this feature of the beta version is not functioning properly as well as estimate the overall reliability of this feature within the beta version.

2 Data Collection

For convenience and due to time constraints, we collected data using our own Learning Suite accounts and those of a few close friends. We used the following process:

- Switch to the beta version of Learning Suite
- Select class
- Check the statistics feature on a particular assignment
- Record results as well as other relevant data in a shared google sheet at the time of the attempt
- Repeat for different classes and assignments at various times of day

We defined a successes and failures in the following way: A success is clicking on the statistics button and receiving the expected histogram display in a reasonable amount of time for your internet speed.

A failure would be incorrect results display or failure to display results after a much longer loading time than expected (compared to your experience on the original version Learning Suite on your system). Assignments where stats are blocked by the professor should not be reported. Visuals for both of the outcomes can be found in Appendix A.

3 Statistical Models

Because we have pass/failure data, we decided to model the failures using the binomial regression. This also allows us to explore the relationships between the failures and the covariates we included. We used a logit link function to move our parameter space from the closed interval [0,1] to the real number line. Within our models we included covariates for time of day and user. Time of day was discretized into morning (2:01am-12pm), afternoon (12:01pm-5pm) and evening (5:01pm-2am). Ryan was used as the baseline user and morning was used as the baseline time of day when calculating the intercept.

We looked at three different models:

1. A binomial model with somewhat informative BCJ priors,

2. A binomial model with hyperpriors on the BCJ priors, and

3. A binomial model with much less informative BCJ priors.

All 3 models had the same Bernoulli likelihood for each observation which is equivalent to a binomial likelihood for the experiment:

$$y \sim Bernoulli(p_i) \tag{1}$$

$$logit(p_i) = \beta_0 + \beta_1 * afternoon_i + \beta_2 * evening_i + \beta_3 * Josh_i + \beta_4 * Riley_i + \beta_5 * Cameron_i \tag{2}$$

Since the β coefficients are less intuitive, instead of putting priors directly on the β's, we put priors on probabilities that were more intuitive for us, and then back-transformed to give us priors for the β coefficients. This type of prior is called a Bedrick-Christensen-Johnson (BCJ) prior.

3.1 Model 1

For the first model, we used previous experiences with Learning Suite to set the priors on the individual failure probabilities. We set lightly informative priors for all BCJ parameters, but slightly weighted towards not seeing problems with the statistics, to coincide with Josh's experience. Since experience from being a teaching assistant has shown us that students trying to submit an exam or assignment right before a deadline can lead to Learning Suite slow downs or crashes, we decided to have the prior for the probability of failure in the evening be higher than other priors. We had no reason to believe that any individual would experience more failures than another and therefore assigned the same prior failure probability to each individual.

$$P_{Ryan/Morning} \sim Beta(1, 1/2)$$

$$P_{Ryan/Afternoon} \sim Beta(1, 1/2)$$

$$P_{Ryan/Evening} \sim Beta(2/3, 2/3)$$

$$P_{Josh/Morning} \sim Beta(1, 1/2)$$

$$P_{Riley/Morning} \sim Beta(1, 1/2)$$

$$P_{Cameron/Morning} \sim Beta(1, 1/2)$$

3.2 Model 2

After this initial model we defined two others using different priors. For model 2 we added a hierarchical element to our BCJ priors by setting hyperpriors on the parameters for the user effects. These priors are defined below.

$$P_{Ryan/Morning} \sim Beta(\alpha, \beta)$$
$$P_{Ryan/Afternoon} \sim Beta(1, 1/2)$$
$$P_{Ryan/Evening} \sim Beta(2/3, 2/3)$$
$$P_{Josh/Morning} \sim Beta(\alpha, \beta)$$
$$P_{Riley/Morning} \sim Beta(\alpha, \beta)$$
$$P_{Cameron/Morning} \sim Beta(\alpha, \beta)$$
$$\alpha \sim Exponential(1)$$
$$\beta \sim Exponential(2)$$

3.3 Model 3

For model 3 we used diffuse BCJ priors in order to see our model's sensitivity to our priors. These priors are once again listed below.

$$P_{Ryan/Morning} \sim Beta(.1, .05)$$
$$P_{Ryan/Afternoon} \sim Beta(.1, .05)$$
$$P_{Ryan/Evening} \sim Beta(.067, .067)$$
$$P_{Josh/Morning} \sim Beta(.1, .05)$$
$$P_{Riley/Morning} \sim Beta(.1, .05)$$
$$P_{Cameron/Morning} \sim Beta(.1, .05)$$

After these priors were set up, for each model we back-transformed them as such:

$$\beta_0 = logit(P_{Ryan/Morning})$$
$$\beta_1 = logit(P_{Ryan/Afternoon}) - logit(P_{Ryan/Morning})$$
$$\beta_2 = logit(P_{Ryan/Evening}) - logit(P_{Ryan/Morning})$$
$$\beta_3 = logit(P_{Josh/Morning}) - logit(P_{Ryan/Morning})$$
$$\beta_4 = logit(P_{Riley/Morning}) - logit(P_{Ryan/Morning})$$
$$\beta_5 = logit(P_{Cameron/Morning}) - logit(P_{Ryan/Morning})$$

.

4 Model Selection and Validation

Using 2 chains, and thinning every other observation, we generated 10,000 observations per model after an initial burn-in of 2000 observations. We looked at the following trace plots to verify convergence:

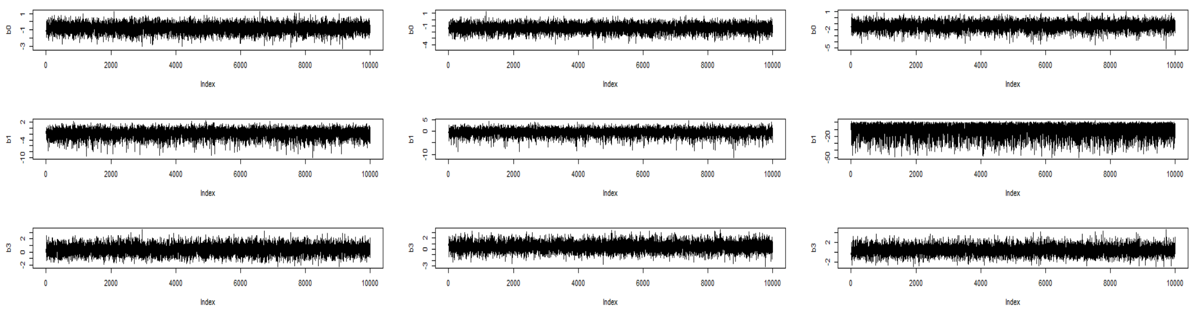

Figure 1: Trace plots for β_0, β_1, and β_2 from Models 1-3, respectively

Because there are no evident trends in the trace plots we can see that these models have converged, and that no additional burn-in is required. The Gelman diagnostic for these MCMC samples showed convergence since the diagnostics for all parameters are close to 1.

Because at least one of our models included some form of hierarchy, it's probably best to only use DIC. The DIC of the 3 models are as follows:

Model 1	Model 2	Model 3
51.8	42.9	40.5

This shows that the Model 1 conforms least to the data, followed by Models 2 and 3, respectively.

We also inspect the appropriateness of the 3 models using a Bayesian χ^2 goodness-of-fit test. The proportion of p-values corresponding to this test that are under .05 are as follows:

Model 1	Model 2	Model 3
.078	.047	.048

The histograms showing the distribution of p-values are seen in Fig. 2.

Notice that none of the histograms look particularly *bad*, and neither do the proportions of p-values under 0.05. However, we do begin to express concern at Model 1, whose p-values are lower on average than a uniform distribution, resulting in a higher proportion of p-values under .05. This pattern indicates Model 1 is somewhat inappropriate, and therefore, we should probably select a more appropriate model. Because Model 3 has the lowest DIC, its Bayesian χ^2 values are phenomenal, and because it holds the same information as the priors we set in Model 1 (just less informative), we decide to select Model 3 as our model of choice.

5 Results

Summary of the covariates found in Table 1. The three most significant covariates were β_1, β_4, and β_5. These represent the log odds ratio of failure in the afternoon and morning, the log odds ratio of failure

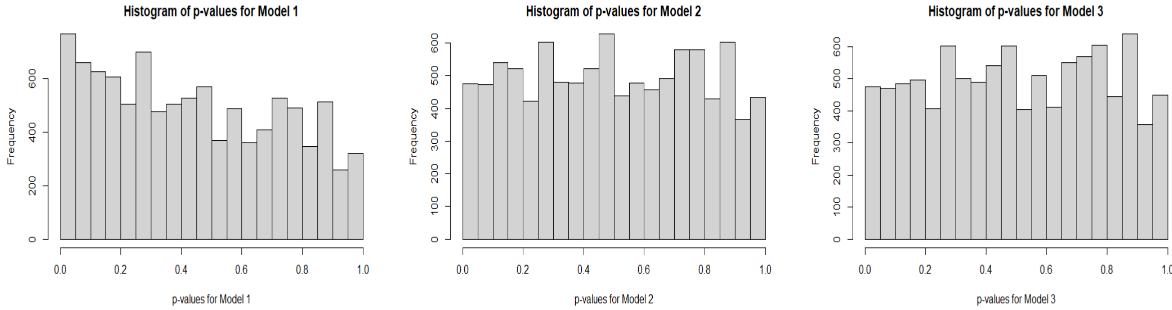

Figure 2: p-values for the Bayesian χ^2 test

between Riley and Ryan, and log odds ratio of failure between Cameron and Ryan, respectively. In short, people are significantly less likely to fail in the afternoon than in the morning, and Riley and Cameron are significantly less likely to fail than Ryan. Josh's probability of failure does not appear to be significantly different than Ryan's. This is noteworthy, showing that the beta version of Learning Suite seems to work better for some people than other people, which might be attributed to certain classes, browsers, operating systems, etc. One thing that surprised us is that there were significantly fewer failures in the afternoon than in the morning.

Estimate	Mean	Median	Std. Err.	95% C.I.	p-val
β_0	-1.284	-1.251	0.697	(-2.746, -0.070)	.0474
β_1	-14.97	-14.41	9.909	(-44.99, -0.340)	.0206
β_2	1.593	1.538	1.474	(-1.079, 4.633)	.2598
β_3	0.480	0.468	0.921	(-1.318, 2.313)	.6126
β_4	-15.01	-11.65	10.92	(-36.97, -4.688)	.0000
β_5	-13.63	-12.53	7.558	(-35.75, -2.662)	.0002

Table 1: Summary of Covariates

6 Conclusions

From our analysis we observed that person had the largest impact on failure probability. This fits our initial exploration of the data which showed that Riley and Cameron had no failures. Other factors seemed to have minimal impact and we observed that the overall reliability of the learning suite beta statistics was very good.

Upon further inquiry into the details on when a failure occurred, we noticed that all of them occurred for Josh and Ryan. After reviewing our methods we realized that these failures were all occurring with assignments in the same class (Statistics 466). This could lead into further questions pertaining to the class size, the professor's settings on the class, and missing values from missing assignments. While we did not include these as covariates in our study, we strongly suspect that there is a relationship between failure and one of these class-specific factors.

We suggest that the Learning Suite team use Statistics 466 as a case study to try and isolate the problem that prevents statistics from displaying properly and minimize future failures. Alternatively, a study comparable to ours that randomly collected data and included information on class-specific covariates could help identify the true cause of the failures.

7 Appendix A: Reference Images

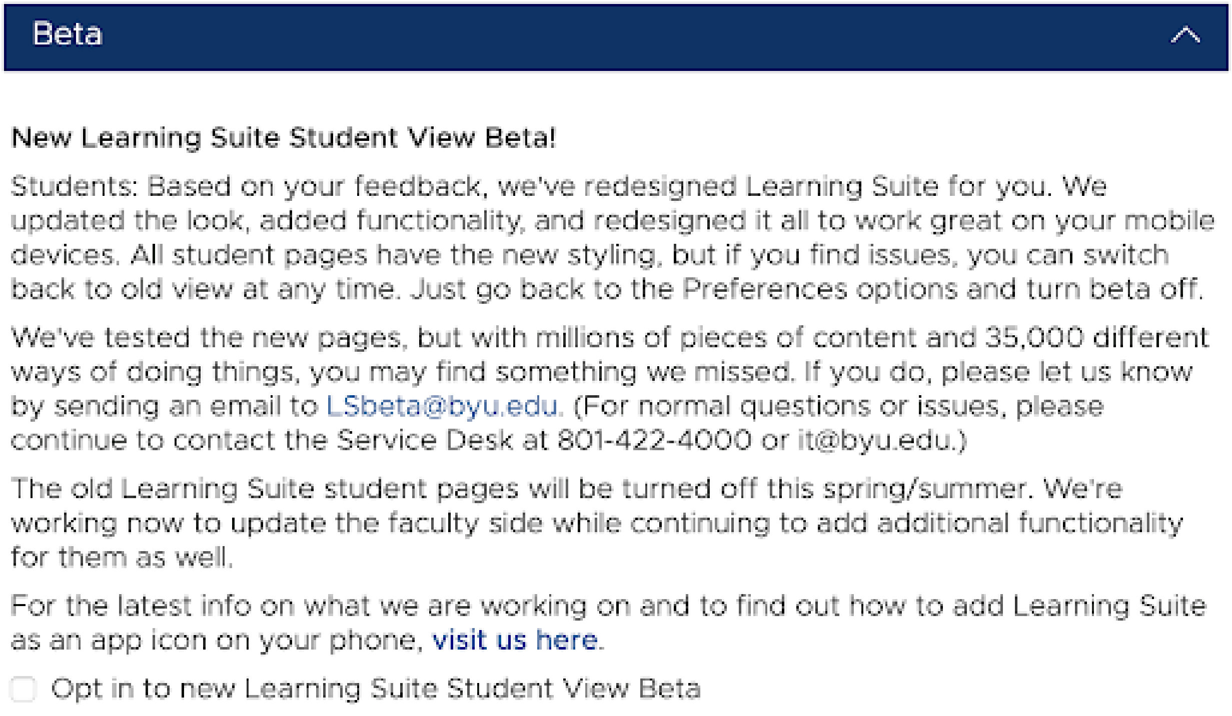

Figure 3: Opt in to Beta Version of LS

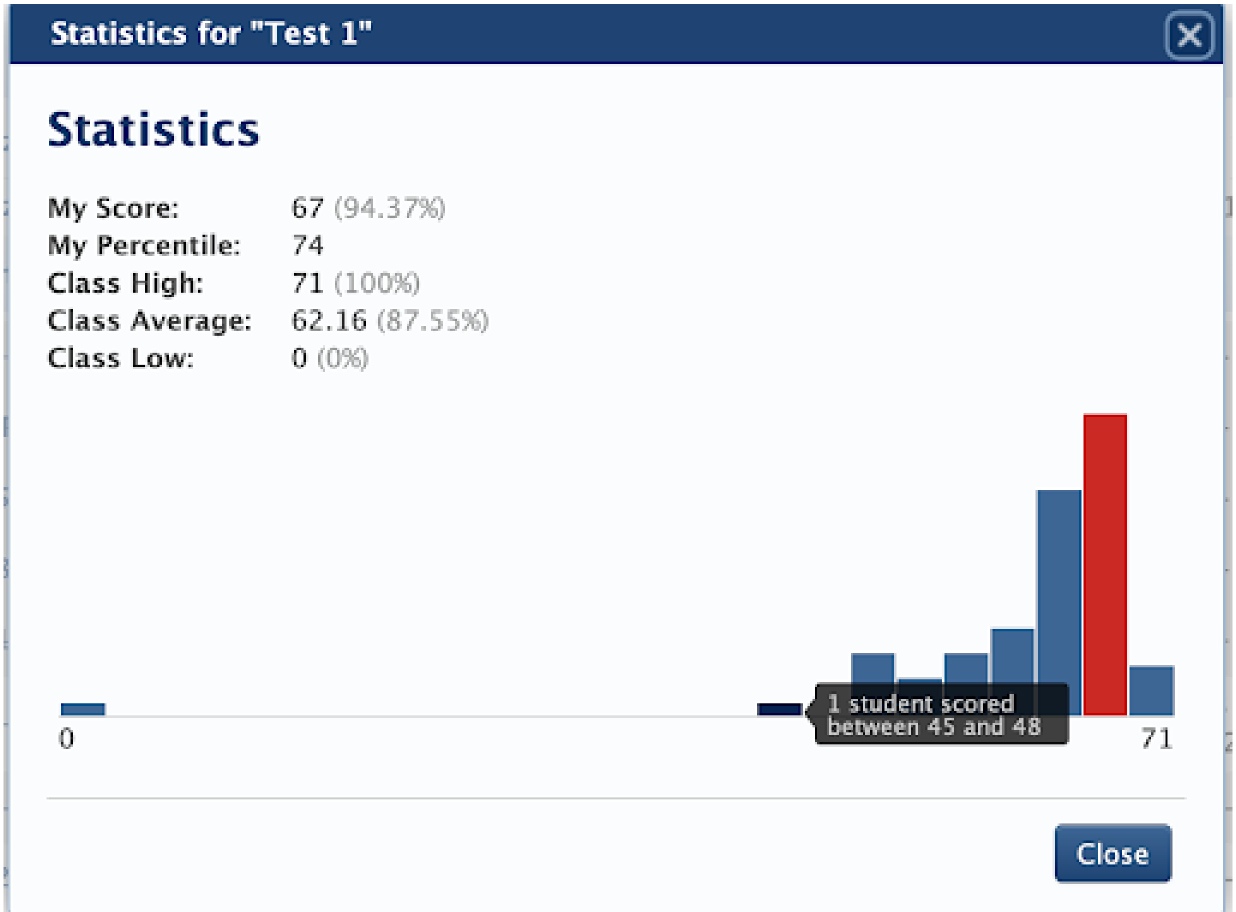

Figure 4: Original LS

Figure 5: Success in Beta Version of LS

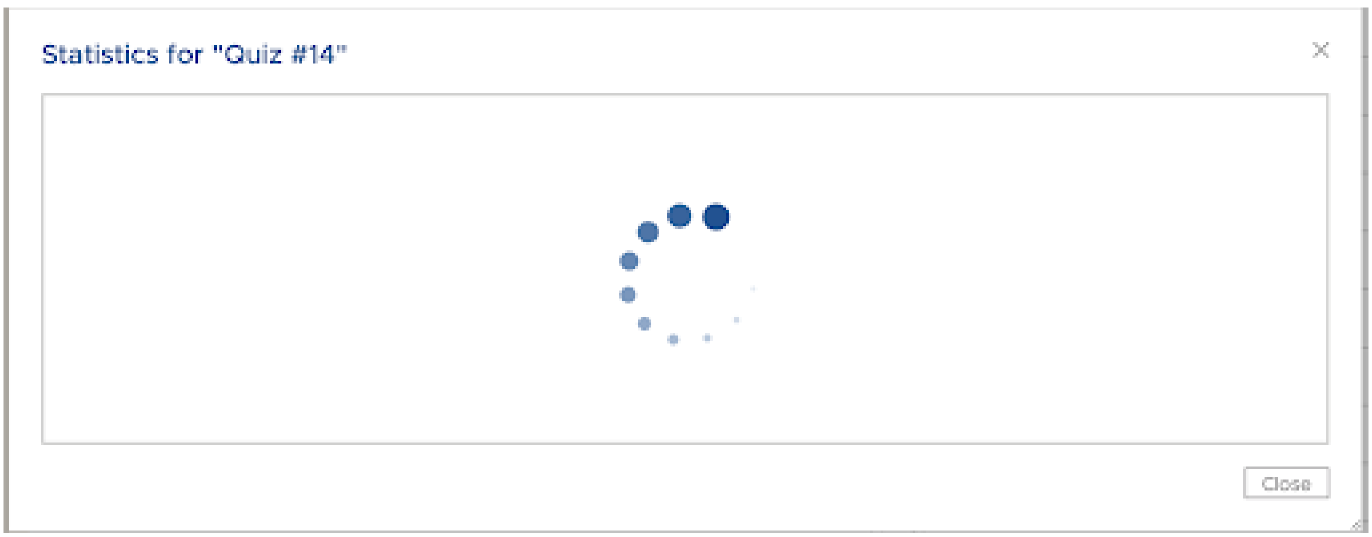

Figure 6: Failure in Beta Version LS

8 Appendix B: Analysis Code

```
library(R2jags)
dat <- read.csv("final_data.csv")
names(dat)[1] <- "fail"
attach(dat)

#CODE (main)
mod1 <-
"model {
for (i in 1:length(fail)){
  fail[i] ~ dbern(p[i])
  p[i] = ilogit(eta[i])
  eta[i] = b0 + b1*aft[i] + b2*eve[i] + b3*josh[i] + b4*riley[i] + b5*cameron[i]
}
pwm ~ dbeta(1,1/2)
pwa ~ dbeta(1,1/2)
pwe ~ dbeta(2/3,2/3)
pjm ~ dbeta(1,1/2)
prm ~ dbeta(1,1/2)
pcm ~ dbeta(1,1/2)
b0 = logit(pwm)
b1 = logit(pwa) - logit(pwm)
b2 = logit(pwe) - logit(pwm)
b3 = logit(pjm) - logit(pwm)
b4 = logit(prm) - logit(pwm)
b5 = logit(pcm) - logit(pwm)
}"

#CODE (hierarchical model)
mod2 <-
"model {
for (i in 1:length(fail)){
  fail[i] ~ dbern(p[i])
  p[i] = ilogit(eta[i])
  eta[i] = b0 + b1*aft[i] + b2*eve[i] + b3*josh[i] + b4*riley[i] + b5*cameron[i]
}
pwm ~ dbeta(a,b)
pwa ~ dbeta(1,1/2)
pwe ~ dbeta(2/3,2/3)
pjm ~ dbeta(a,b)
prm ~ dbeta(a,b)
pcm ~ dbeta(a,b)
```

```
a ~ dexp(1)
b ~ dexp(2) #mean of 1/2
b0 = logit(pwm)
b1 = logit(pwa) - logit(pwm)
b2 = logit(pwe) - logit(pwm)
b3 = logit(pjm) - logit(pwm)
b4 = logit(pmm) - logit(pwm)
b5 = logit(pcm) - logit(pwm)
}"

#CODE (flat priors)
mod3 <-
"model {
for (i in 1:length(fail)){
  fail[i] ~ dbern(p[i])
  p[i] = ilogit(eta[i])
  eta[i] = b0 + b1*aft[i] + b2*eve[i] + b3*josh[i] + b4*riley[i] + b5*cameron[i]
}
pwm ~ dbeta(.1,.05)
pwa ~ dbeta(.1,.05)
pwe ~ dbeta(.067,.067)
pjm ~ dbeta(.1,.05)
pmm ~ dbeta(.1,.05)
pcm ~ dbeta(.1,.05)
b0 = logit(pwm)
b1 = logit(pwa) - logit(pwm)
b2 = logit(pwe) - logit(pwm)
b3 = logit(pjm) - logit(pwm)
b4 = logit(pmm) - logit(pwm)
b5 = logit(pcm) - logit(pwm)
}"

Sim1 <- jags( data=c("fail","aft","eve","josh","riley","cameron"),
              parameters.to.save=c("b0","b1","b2","b3","b4","b5"),
              model.file=textConnection(mod1), n.iter=12000, n.burnin=2000,
              n.chains=2, n.thin=2)

Sim2 <- jags( data=c("fail","aft","eve","josh","riley","cameron"),
              parameters.to.save=c("b0","b1","b2","b3","b4","b5"),
              model.file=textConnection(mod2), n.iter=12000, n.burnin=2000,
              n.chains=2, n.thin=2)

Sim3 <- jags( data=c("fail","aft","eve","josh","riley","cameron"),
```

```
            parameters.to.save=c("b0","b1","b2","b3","b4","b5"),
            model.file=textConnection(mod3), n.iter=12000, n.burnin=2000,
            n.chains=2, n.thin=2)

par(mfrow=c(3,1))

plot(Sim1$BUGSoutput$sims.matrix[,"b0"],type = "l", ylab="b0")
plot(Sim1$BUGSoutput$sims.matrix[,"b1"],type = "l", ylab="b1")
plot(Sim1$BUGSoutput$sims.matrix[,"b3"],type = "l", ylab="b3")

plot(Sim2$BUGSoutput$sims.matrix[,"b0"],type = "l", ylab="b0")
plot(Sim2$BUGSoutput$sims.matrix[,"b1"],type = "l", ylab="b1")
plot(Sim2$BUGSoutput$sims.matrix[,"b3"],type = "l", ylab="b3")

plot(Sim3$BUGSoutput$sims.matrix[,"b0"],type = "l", ylab="b0")
plot(Sim3$BUGSoutput$sims.matrix[,"b1"],type = "l", ylab="b1")
plot(Sim3$BUGSoutput$sims.matrix[,"b3"],type = "l", ylab="b3")

gelman.diag(as.mcmc(Sim1))
gelman.diag(as.mcmc(Sim2))
gelman.diag(as.mcmc(Sim3))

Sim1 <- jags( data=c("fail","aft","eve","josh","riley","cameron"),
            parameters.to.save=c("b0","b1","b2","b3","b4","b5","p"),
            model.file=textConnection(mod1), n.iter=12000, n.burnin=2000,
            n.chains=2, n.thin=2)

Sim2 <- jags( data=c("fail","aft","eve","josh","riley","cameron"),
            parameters.to.save=c("b0","b1","b2","b3","b4","b5","p"),
            model.file=textConnection(mod2), n.iter=12000, n.burnin=2000,
            n.chains=2, n.thin=2)

Sim3 <- jags( data=c("fail","aft","eve","josh","riley","cameron"),
            parameters.to.save=c("b0","b1","b2","b3","b4","b5","p"),
            model.file=textConnection(mod3), n.iter=12000, n.burnin=2000,
            n.chains=2, n.thin=2)

Sim1$BUGSoutput$DIC
Sim2$BUGSoutput$DIC
Sim3$BUGSoutput$DIC

GoF_Test <- function(fitted_quantiles) {
  n <- length(fitted_quantiles)
```

```
  K <- round((n)^(0.4))
  mK <- table(cut(fitted_quantiles,(0:K)/K))
  np <- n/K
  RB <- sum(((mK-np)^2)/np)
  return(1-pchisq(RB,K-1))
}

ps1 <- Sim1$BUGSoutput$sims.matrix[,-c(1:7)]
ps2 <- Sim2$BUGSoutput$sims.matrix[,-c(1:7)]
ps3 <- Sim3$BUGSoutput$sims.matrix[,-c(1:7)]

GoF1 <- matrix(NA,ncol=length(fail),nrow=nrow(ps1))
for (i in 1:nrow(ps1)) {
  GoF1[i,] <- runif(length(fail),
                    pbinom(fail-1,1,ps1[i,]),
                    pbinom(fail,1,ps1[i,]))  #ppois(fails,lam1[i])
}
# Calculating the p-values for each posterior model
GoF_Summary1 <- apply(GoF1,1,GoF_Test)
# Histogram of posterior model p-values
hist(GoF_Summary1, xlim=c(0,1), main = "Histogram of p-values for Model 1",
     xlab = "p-values for Model 1")
# Percent of posterior models with p-value less than 0.05
mean(GoF_Summary1 < 0.05)

GoF2 <- matrix(NA,ncol=length(fail),nrow=nrow(ps2))
for (i in 1:nrow(ps2)) {
  GoF2[i,] <- runif(length(fail),
                    pbinom(fail-1,1,ps2[i,]),
                    pbinom(fail,1,ps2[i,]))  #ppois(fails,lam1[i])
}
# Calculating the p-values for each posterior model
GoF_Summary2 <- apply(GoF2,1,GoF_Test)
# Histogram of posterior model p-values
hist(GoF_Summary2, xlim=c(0,1), main = "Histogram of p-values for Model 2",
     xlab = "p-values for Model 2")
# Percent of posterior models with p-value less than 0.05
mean(GoF_Summary2 < 0.05)

GoF3 <- matrix(NA,ncol=length(fail),nrow=nrow(ps3))
for (i in 1:nrow(ps3)) {
  GoF3[i,] <- runif(length(fail),
                    pbinom(fail-1,1,ps3[i,]),
```

```
                    pbinom(fail,1,ps3[i,]))   #ppois(fails,lam1[i])
}
# Calculating the p-values for each posterior model
GoF_Summary3 <- apply(GoF3,1,GoF_Test)
# Histogram of posterior model p-values
hist(GoF_Summary3, xlim=c(0,1), main = "Histogram of p-values for Model 3",
     xlab = "p-values for Model 3")
# Percent of posterior models with p-value less than 0.05
mean(GoF_Summary3 < 0.05)

options(digits = 4)

# Model 3
b0 <- Sim3$BUGSoutput$sims.matrix[,"b0"]
b1 <- Sim3$BUGSoutput$sims.matrix[,"b1"]
b2 <- Sim3$BUGSoutput$sims.matrix[,"b2"]
b3 <- Sim3$BUGSoutput$sims.matrix[,"b3"]
b4 <- Sim3$BUGSoutput$sims.matrix[,"b4"]
b5 <- Sim3$BUGSoutput$sims.matrix[,"b5"]

mean(b0)
mean(b1)
mean(b2)
mean(b3)
mean(b4)
mean(b5)

median(b0)
median(b1)
median(b2)
median(b3)
median(b4)
median(b5)

sd(b0)
sd(b1)
sd(b2)
sd(b3)
sd(b4)
sd(b5)

quantile(b0,c(.025,.975))
quantile(b1,c(.025,.975))
```

```
quantile(b2,c(.025,.975))
quantile(b3,c(.025,.975))
quantile(b4,c(.025,.975))
quantile(b5,c(.025,.975))

2*min(mean(b0>0),mean(b0<0))
2*min(mean(b1>0),mean(b1<0))
2*min(mean(b2>0),mean(b2<0))
2*min(mean(b3>0),mean(b3<0))
2*min(mean(b4>0),mean(b4<0))
2*min(mean(b5>0),mean(b5<0))
```

9 Appendix C: Analysis Data

Result	aft	eve	josh	riley	cameron
0	1	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
1	0	1	1	0	0
1	0	1	1	0	0
0	0	1	1	0	0
0	0	1	0	0	1
0	0	1	0	0	1
0	0	1	0	0	1
0	0	1	0	0	1
0	0	1	0	0	1
0	1	0	0	1	0
0	1	0	0	1	0
0	1	0	0	1	0
0	1	0	0	1	0
0	1	0	0	1	0
0	0	0	0	0	0
1	0	0	1	0	0
1	0	0	1	0	0
1	0	0	1	0	0
0	0	0	1	0	0
0	0	0	1	0	0
1	0	0	1	0	0
1	0	0	0	0	0
1	0	0	0	0	0
1	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	1	0	0	0	0
0	1	0	0	0	1
0	1	0	0	0	1
0	1	0	0	0	1
0	1	0	0	0	1
0	1	0	0	0	1

0	1	0	0	0	1
0	1	0	0	0	1
0	1	0	0	0	1
0	1	0	0	0	1
0	0	1	0	1	0
0	0	1	0	1	0
0	0	1	0	1	0
0	0	1	0	1	0
0	0	1	0	1	0
0	0	1	0	1	0
0	0	1	0	1	0
0	0	1	0	1	0
0	0	1	0	1	0
0	0	1	0	1	0
0	0	1	0	1	0
0	0	1	0	1	0
0	0	1	0	1	0
0	0	1	0	1	0
0	0	0	1	0	0
0	0	0	1	0	0
0	0	0	1	0	0
0	0	0	1	0	0
0	0	0	1	0	0
0	0	0	1	0	0
0	0	0	1	0	0