HW6

Rory Michelen

November 9, 2019

Question 1

Model Creation

To answer this question, I will model the lifetimes as a weibull distribution. The model is created in R2Open Bugs, below

```
# Read the input data
input.data<-read_csv('tongue.csv',col_names=FALSE)
colnames(input.data)<-c('dna.profile','duration','observed')
input.data<-input.data%>%
  mutate(lifetime=ifelse(observed==1,duration,NA),
         censored=ifelse(observed==0,duration,0))
```

```
## Warning: package 'bindrcpp' was built under R version 3.5.1
```

```
lifetime<-input.data$lifetime
censored<-input.data$censored
profile<-input.data$dna.profile
N<-nrow(input.data)

# Specify the model
model<-function(){
  for(i in 1 : N){
    lifetime[i]~dweib(v,lambda[i])%_%C(censored[i],)
    lambda[i]<-exp(b0+b1*profile[i])
    median[i]<-pow(log(2)*exp(-b0-b1*profile[i]),1/v)
  }
  b0~dnorm(0,0.0001)
  b1~dnorm(0,0.0001)
  v~dexp(0.001)
}

model.file <- file.path(tempdir(), "model.txt")
write.model(model, model.file)

data<-list("lifetime","censored","profile","N")
inits <- function() {list(b0=0,b1=0,v=1)}
params<-c("b0","b1","v","median")

out <- bugs(data, inits, params,model.file, n.iter=10000)
```

Part A

The 95% Credible Set is provided below for all 3 parameters. For b_1 the credible set is [-1.368,0.370]

```
out$summary%>%
  data.frame()%>%
  rownames_to_column('Parameter')%>%
  select(Parameter,mean,lower.bound='X2.5.',upper.bound='X97.5.')%>%
  dplyr::filter(Parameter %in% c('b0','b1','v'))%>%
  kable()
```

Parameter	mean	lower.bound	upper.bound
b0	-9.7449317	-11.270	-7.7740
b1	-0.2946891	-1.368	0.3698
v	1.9589221	1.597	2.2550

Question 2

```
X<-c(2, 1, 0, 2, NA, 3, 1, 0, 1, 2, 3, 0, 1, NA, NA)
Y<-c(NA, 16, 9, 17, 12, 22, 13, 8, NA, 19, 17, 11, 10, 20, 2)
N<-length(X)


model<-function(){
    for(i in 1:N){
        Y[i]~dpois(lambda[i])
        lambda[i]<-exp(beta[1]+beta[2]*X[i])
        X[i]~dpois(2)
    }
    for(i in 1:2){
        beta[i]~dnorm(0,0.0001)
    }
  mean.w.4routes<-exp(beta[1]+beta[2]*4)
  predict.w.4routes~dpois(mean.w.4routes)
}

model.file <- file.path(tempdir(), "model.txt")
write.model(model, model.file)

data<-list("X","Y","N")
inits <- function() {list(beta=c(0,0),X=c(NA, NA, NA, NA, 2, NA, NA, NA, NA, NA, NA, NA, NA, 2, 2))}
params<-c("beta","X","Y","mean.w.4routes","predict.w.4routes")

out <- bugs(data, inits, params,model.file, n.iter=10000)
```

Part A

The output of the above model is provided below. The deviance of our model is 109.588 with an 95% credible set of [105.5,118.0]

```
out$summary%>%
  data.frame()%>%
  select(mean,lower.bound='X2.5.',upper.bound='X97.5.')%>%
  rownames_to_column("Parameter")%>%
  kable()
```

Parameter	mean	lower.bound	upper.bound
beta[1]	2.1710887	1.8860	2.4370

Parameter	mean	lower.bound	upper.bound
beta[2]	0.2880554	0.1442	0.4373
X[5]	1.2742667	0.0000	3.0000
X[14]	2.5636000	1.0000	4.0000
X[15]	0.1986000	0.0000	1.0000
Y[1]	15.6787333	8.0000	24.0000
Y[9]	11.7696000	5.0000	19.0000
mean.w.4routes	28.2973700	18.7200	40.7900
predict.w.4routes	28.3249333	15.0000	45.0000
deviance	109.5881867	105.5000	118.0000

Part B

The mean expected number of broken packages for a shipment with 4 routes has a credible interval of [18.72,40.79] with a mean of 28.29.

Part C

The predicted number of broken packages for a shipment with 4 routes has a credible interval of [15,45] and a mean of 28.32.

While the mean and the prediction are centered roughly at the same value, the credible set of the prediction is wider than that of the mean. This is because the prediction has an additional source of uncertainty.

There is a shared source of uncertainty: What is the true value of the paramter lambda. However, the prediction has an additional source of uncertainty: given a known distribution with fixed lambda, what will the value of the next observation be?

Part D

The table below provides estimates of all missing data

```r
out$summary%>%
  data.frame()%>%
  select(mean,lower.bound='X2.5.',upper.bound='X97.5.')%>%
  rownames_to_column("Parameter")%>%
  dplyr::filter(substr(Parameter,1,1)%in% c('X','Y'))%>%
  kable()
```

Parameter	mean	lower.bound	upper.bound
X[5]	1.274267	0	3
X[14]	2.563600	1	4
X[15]	0.198600	0	1
Y[1]	15.678733	8	24
Y[9]	11.769600	5	19