Bayesian Data Analysis Exam 1

Your Name Here

APRIL 7, 2022

1 Exam 1 Instructions and information about the exam

1. You have 24 hours to complete this exam.
2. All working should be placed in an Rmd document, calculations done in R using R functions etc
3. Answer all questions!
4. All code should be placed into r chunks and Mathematics should be written in Latex.
5. Please upload to CANVAS the RMD and HTML as any other Assignment.
6. Treat this document as a template for the exam and fill out the answers below each question.
7. This is an open book exam and any literature can be consulted.
8. You cannot consult with people concerning the exam once the exam begins.
9. This exam is not proctored.
10. You will need to have viable R, RStudio Jags and OpenBUGS installations available to use on your machine.
11. CANVAS will give the details of when the exam is due.

2 Questions

2.1 Basic Distributional Properties

Suppose that $X \sim Pois(\lambda = 4)$ where λ is the average number of particles leaving an ustable nucleus per second.

2.1.1 Find $P(X \geq 8)$ using an R function

2.1.2 Find $P(X = 2)$

2.1.3 Make a plot of the distribution of X for X=0 up to X=30

The following is for the case where $Y \sim N(\mu = 6, \sigma = 10)$

2.1.4 Find $P(Y \leq 3)$ using an R function

2.1.5 Find $P(5 \leq Y < 8)$ using R

2.1.6 Make an R function `mynorm(a,b, mu)` that will plot the area between `a` and `b` where `b>a` and below the density curve when Y is distributed as follows, $Y \sim N(\mu, \sigma = 10)$ and display a 4 dec estimate of the area. Make sure your name appears on the plot.

2.1.6.1 Call your function with the following parameters

a. mynorm(3,25,5)

b. mynorm(0,20,5)

c. mynorm(-6,10, 5)

2.2 The following relates to Bayes' rule

2.2.1 If $\theta \sim Beta(\alpha, \beta)$ and $X \sim Bin(n, \theta)$ prove that $\theta|X \sim Beta(x + \alpha, n - x + \beta)$ by completing the proof below:

$$
\begin{aligned}
p(\theta|x) &\propto p(\theta)p(x|\theta) \\
&= \frac{1}{B(\alpha, \beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}\binom{n}{x}\theta^x(1-\theta)^{n-x}
\end{aligned}
$$

2.2.2 Prove Bayes' rule for the case where A and B are two discrete events. That is, show:

$$p(A|B) = \frac{p(A)p(B|A)}{p(B)}$$

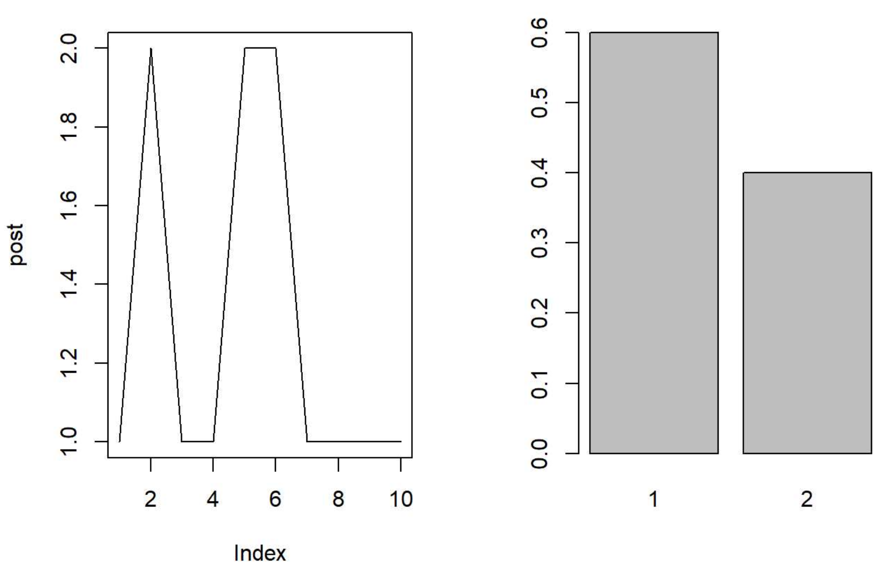
You may assume:

$$p(A|B) = \frac{p(A \cap B)}{p(B)}$$

2.3 The following is some code for performing a coin die simulation

```r
coindie<-function(n=100, h=c(1/4,3/4),E2=c(5,6),init=1,...){
if(!require(xtable)) install.packages(xtable)
dieset<-c()
dieset[1]<-"E1"
die<-function(n=1){
sample(1:6,size=n,replace=TRUE)
}

coin<-function(n=1){
sample(1:2,size=n,replace=TRUE)
}
face<-c()
alpha<-c() # holds acceptance probs
alpha[1]<-1
post<-c()# post sample
prop<-c() # vec of proposed states 1s and 2s
prop[1]=init # initial state
post[1]=prop[1]
dice<-c()
dice[1]<-die()

for(i in 2:n){ # starts at 2 because initial value given above
prop[i]<-coin()
alpha[i]=min(1,h[prop[i]]/h[post[i-1]])

dice[i]<-die()
ifelse(alpha[i]==1,dieset[i]<-"E1",dieset[i]<-"E2")
# is x an element of set y
if(alpha[i]==1 | (is.element(dice[i],E2) & alpha[i]!=1)){post[i]<-prop[i]}
else{post[i]<-post[i-1]}
 }
res<-matrix(c(prop,round(alpha,2),dieset,dice,post ),nc=5,nr=n,byrow=FALSE,dimnames=list(1:n,c("proposal","alpha", "E","dic
e","post")))
sim<-table(post)/n
postexact<-h/sum(h)
#dev.new(noRStudioGD = TRUE)
layout(matrix(1:2, nr=1,nc=2))
plot(post,type ="l",...)
names(h)=c("1", "2")
barplot(h)
return(list(iter=res,sim=sim,postexact=postexact,post=post,xtable=xtable(res,dig=1)) )
}

set.seed(25)#24
coindie(n=10,h=c(0.6,0.4),E2=c(3,4,5,6)) ->ans
```

```
## Loading required package: xtable
```



```
ans$it
```

```
##    proposal alpha  E    dice post
## 1  "1"      "1"    "E1" "5"  "1"
## 2  "2"      "0.67" "E2" "4"  "2"
## 3  "1"      "1"    "E1" "1"  "1"
## 4  "2"      "0.67" "E2" "1"  "1"
## 5  "2"      "0.67" "E2" "3"  "2"
## 6  "2"      "1"    "E1" "4"  "2"
## 7  "1"      "1"    "E1" "3"  "1"
## 8  "1"      "1"    "E1" "4"  "1"
## 9  "2"      "0.67" "E2" "2"  "1"
## 10 "1"      "1"    "E1" "5"  "1"
```

2.3.1 When the coindie function is called above what is the acceptance set?

2.3.2 There are 10 iterations above. From the table of output `ans$it` how many times does the sampler move from state 1 to 2?

2.3.3 Write down the formula for the acceptance probability.

2.4 This relates to Jags and OpenBUGS

2.4.1 Make a DAG using the doodle bug editor in OpenBUGS for the simple linear regression model. Use low impact priors, the model is $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$ where $\epsilon_i \sim N(0, \sigma^2)$ Assume `n` data from an experiment. (You have done this in lab 5) Check that the model compiles. Record the DAG from the doodle editor here: use `![](jpeg){ width=60% }`

2.4.2 Once you have checked the model in OpenBUGS use `pretty print` and place the code in the appropriate place in the R code below – you may need to adjust a few things to make it work – remove the option `eval = FALSE` from the code chunk. Be sure to check that it works!

```r
require(rjags)               # Must have previously installed package rjags.

fileNameRoot="Exam1" # For output file names.
x=-20:19
set.seed(33)
y=10+20*x + rnorm(40,0,15)

Ntotal = length(y)  # Compute the total number of x,y pairs.
dataList = list(    # Put the information into a list.
  x = x,
  y = y ,
  Ntotal = Ntotal
)

Define the model:
modelString = "







" # close quote for modelString
writeLines( modelString , con="TEMPmodel.txt" )

# Initialize the chains based on MLE of data.
# Option: Use single initial value for all chains:
#  thetaInit = sum(y)/length(y)
#  initsList = list( theta=thetaInit )

initsList = list(beta0 = 0, beta1 = 0, beta2=0, sigma =10)

# Run the chains:
jagsModel = jags.model( file="TEMPmodel.txt" , data=dataList , inits=initsList ,
                        n.chains=3 , n.adapt=500 )
update( jagsModel , n.iter=500 )
codaSamples = coda.samples( jagsModel , variable.names=c("beta0", "beta1", "beta2" ,"sigma") ,
                            n.iter=33340 )
save( codaSamples , file=paste0(fileNameRoot,"Mcmc.Rdata") )
```

2.4.3 Summarize the MCMC chains by using the appropriate code in the chunk below - remove `eval = FALSE` when ready:

```r
summary(...)
```

2.4.4 Give a 95% Bayesian credible interval for β_1 and interpret it.

The following is the data used in the above SLR analysis.

```r
x=-20:19
set.seed(33)
y=10+20*x + rnorm(40,0,15)
y=round(y,1)
df = data.frame(x=x,y=y)
head(df)
```

```
##     x      y
## 1 -20 -392.0
## 2 -19 -370.6
## 3 -18 -334.8
## 4 -17 -332.4
## 5 -16 -342.3
## 6 -15 -282.5
```

2.4.5 Using the simulated data in the above code chunk and the `ggplot2` package make a plot of the points `y vs x` and add the estimating line from the point estimates you found from the Bayesian model.

```
library(ggplot2)
# This code needs to be changed to take the estimating intercept and slope
g = ggplot(df, aes(x=x,y=y)) + geom_point() + geom_abline(intercept = 10, slope= 20)
g = g + ggtitle("Your Name")
g
```