# 6450 BayesianMethods Project (Driver File)

*TeamPAJ (P.Jordan, A.Jassal, J.Imgrund)*

*25June2018*

## Contents



```
#
#
#
```

### Environment Preparation

```
# Remove any objects in the Environment
rm(list = ls())

# Closes all of R's graphics windows
graphics.off()

# Knitr global options
library(knitr)
opts_chunk$set(eval = TRUE, echo = TRUE, warning = FALSE,
               tidy = TRUE, results = "hold", cache = TRUE)

# Load Necessary Libraries
suppressPackageStartupMessages(library(dplyr))

# Set the overall seed for reproducibility
set.seed(6450)
```

#### Working Directory

```
# initialize the working directory for the final project
jim_dir = "/Users/jimgrund/Documents/GWU/Bayesian_Methods/Final-Project/"
akash_dir = "C:/Users/akash/Desktop/GWU/6450_Bayesian_YHuang/project/project"
patrick_dir = "/Users/pjordan/Documents/GWU/6450/FinalProject"

for (directory in c(akash_dir, jim_dir, patrick_dir)) {
    if (dir.exists(directory)) {
        setwd(directory)
        break
    }
}

# Load the relevant model into R's working memory:
source("model.R")
```

```
## Loading required package: coda
## Linked to JAGS 4.3.0
## Loaded modules: basemod,bugs
source("DBDA2E-utilities.R")
##
## ***********************************************************************
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.
## ***********************************************************************
##
##
## ***********************************************************************
## Kruschke, J. K. (2015). Doing Bayesian Data Analysis, Second Edition:
## A Tutorial with R, JAGS, and Stan. Academic Press / Elsevier.
## ***********************************************************************
```

## Data Load & Tidy

```
filename = "data.csv"
```

### Load Data

```
# Load the Breast Cancer dataset
myData = LoadData(filename)
```

### Feature Importance

```
rf_fi = RFFeatureImportance(myData)  # Run Feature Importance to identify which features to analyze
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:dplyr':
##
##     combine
```

### Bin the Parameters

```
myData = BinData(myData)  # Bin the various columns that we may want to model with
```

### Test Count

```
myData$tests <- 1  # Every patient has performed one test.
```

### Dependent Variable Code

```
myData <- mutate(myData, diagnosis_code = ifelse(diagnosis == "B", 0, 1))  # Construct a column for diag
```

### Interested Parameter

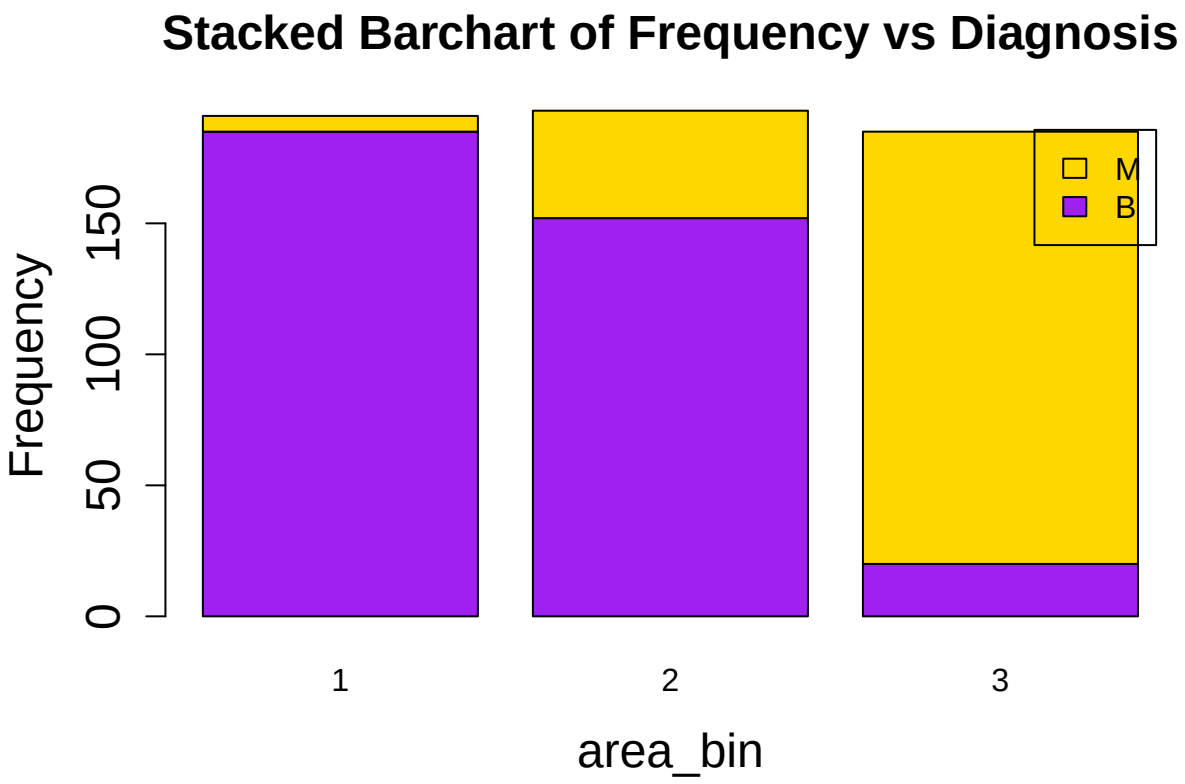
```
interested_parameter = "area_bin"  # Define Interested Parameter
```

### Graph Options

```
# Optional: Specify filename root and graphical format for saving output.
# Otherwise specify as NULL or leave saveName and saveType arguments out of
# function calls. fileNameRoot = interested_perimeter
fileNameRoot = paste(interested_parameter, "_")
graphFileType = "png"
```

### Plot Stacked Histogram

```
# plot the distribution of the interested_parameter to show the distribution
# of B vs M in each of the constructed bins
PlotHistogram(myData, interested_parameter)
```




```
## pdf
##   2
```

## MCMC Chain

### Generate

```
# Generate the MCMC chain:
startTime = proc.time()
mcmcCoda = genMCMC(data = myData, zName = "diagnosis_code", NName = "tests",
    sName = "id", cName = interested_parameter, numSavedSteps = 11000, saveName = fileNameRoot,
    thinSteps = 20)
stopTime = proc.time()
elapsedTime = stopTime - startTime
show(elapsedTime)
```

```
## Calling 4 simulations using the parallel method...
## Following the progress of chain 1 (the program will wait for all
## chains to finish before continuing):
## Welcome to JAGS 4.3.0 on Sun Jun 24 21:42:33 2018
## JAGS is free software and comes with ABSOLUTELY NO WARRANTY
## Loading module: basemod: ok
## Loading module: bugs: ok
## . . Reading data file data.txt
## . Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 569
##    Unobserved stochastic nodes: 577
##    Total graph size: 2317
## . Reading parameter file inits1.txt
## . Initializing model
## . Adapting 500
## -------------------------------------------------| 500
## +++++++++++++++++++++++++++++++++++++++++++++++++ 100%
## Adaptation successful
## . Updating 500
## -------------------------------------------------| 500
## ************************************************* 100%
## . . . . . . Updating 55000
## -------------------------------------------------| 55000
## ************************************************* 100%
## . . . . Updating 0
## . Deleting model
## .
## All chains have finished
## Simulation complete.  Reading coda files...
## Coda files loaded successfully
## Finished running the simulation
##    user  system elapsed
##   38.90    8.67  910.95
```

### Diagnostics

```
# Display diagnostics of chain, for specified parameters:
parameterNames = varnames(mcmcCoda)  # get all parameter names for reference
```

```
for (parName in c("omega[1]", "omega0", "kappa[1]", "kappa0", "theta[1]")) {
    diagMCMC(codaObject = mcmcCoda, parName = parName, saveName = fileNameRoot,
        saveType = graphFileType)
}
```

### Summary Statistics

```
# Get summary statistics of chain: summaryInfo = smryMCMC( mcmcCoda ,
# compVal=NULL , diffSVec=c(75,156, 159) , diffCVec=c(1,2,3) ,
# compValDiff=0.0 , saveName=fileNameRoot )
```

### Plot Results

```
# Display posterior information:
plotMCMC(mcmcCoda, data = myData, zName = "diagnosis_code", NName = "tests",
    sName = "id", cName = interested_parameter, compVal = NULL, diffCList = list(c(1,
        2), c(2, 3), c(1, 3)), diffSList = list(c("842302", "926682"), c("8510653",
        "84501001"), c("855563", "91376702")), compValDiff = 0, ropeDiff = c(-0.05,
        0.05), saveName = fileNameRoot, saveType = graphFileType)
```