````
---
title: "STT465_Hw6"
author: "Sam Isken"
date: "November 24, 2019"
output:
  word_document: default
  html_document: default
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
```

STT 465  Fall 2019

Homework 6  Due 12/04/2019 (In Class)
Instruction:
When using R in any problem, copy the code and results onto your word document under that question number and add any required comments. You will lose points if I do not see your codes. You should present a stapled document when multiple pages are used. The grader will not be held responsible for any loss of pages.

Logistic Regression
(1) Maximum Likelihood Estimation(Frequentist) Analysis

Using the titanic data set (in D2L) fit a logistic regression with survived as response, sex, class and age as predictors using glm. Note: sex and class are factors, while age is a continuous predictor. Note: Some entries have missing values. Be sure to remove all the rows of the data set that contain missing values @ sex, class, age or survived. Hint: you can find missing values using is.na(DATA$survivied) or non-missing using !is.na(DATA$survived).

# Ask questions and finish data cleaning

```{r}
titanic <- read.csv("titanic.csv",header=TRUE)
titanic_limit <- titanic %>% dplyr::select(survived,sex,pclass, age)
titanic_rm_na <- na.omit(titanic_limit)
titanic_rm_na$sex <- factor(titanic_rm_na$sex, levels=c("male","female"), labels=c(0,1))
head(titanic_rm_na)
```

```{r}
titanic.logit <- glm(survived ~ sex+pclass+age, data = titanic_rm_na, family = "binomial")
```


a. Report parameter estimates, SEs and p-values

```{r}
summary(titanic.logit)
```

b. Summarize your findings

```{r}
logit2prob <- function(logit){
  odds <- exp(logit)
  prob <- odds / (1 + odds)
  return(prob)
}

coef(titanic.logit)

print(paste(logit2prob(2.49737591),"in probability of living given a Male"))
print(paste(logit2prob(1.13324383),"in probability of living given a unit increase in pclass"))
print(paste(logit2prob(0.03388497),"in probability of living given a unit increase in age"))

```

c. Report estimated probabilities for male and female in each class set age to be :

(i) 35

(Intercept)  2.09189
sex1         2.49738
pclass      -1.13324
age         -0.03388


```{r}
# Female output of predicted probabilities
newdata=titanic_rm_na
newdata$sex <- "0"

logit2prob(predict(titanic.logit,newdata = newdata))

# Male output of predicted probabilities
newdata=titanic_rm_na
newdata$sex <- "0"

logit2prob(predict(titanic.logit,newdata = newdata))

```


(ii) 55

Compare the two age groups by sex.

(2) Bayesian Analysis

Use the Metropolis sampler developed in class logisticRegressionBayes to fit the logistic regression. Collect 55,000 samples, discard the frist 5,000 for burn in. Note: to avoid confusion when comparing results from the Bayesian and ML analysis, do not center the predictors.

```{r}
# A function to evaluate the log of the posterior density
logP=function(y,X,b,b0,varB){
````
````

```
  Xb=X%*%b
  theta=exp(Xb)/(1+exp(Xb))
  logLik=sum( dbinom(x=y,p=theta,size=1,log=T)  )
  logPrior=sum(  dnorm(x=b,sd=sqrt(varB),mean=b0,log=T))
  return(logLik+logPrior)
}
```

```{r}
logisticRegressionBayes=function(y,X,nIter=100000,V=.02,varB=rep(10000,ncol(X)),b0=rep(0,ncol(X))){

  ####### Arguments #######################
  # y  a vector with 0/1 values
  # X  incidence matrix of effects
  # b0,varB, the prior mean and prior variance bj~N(b0[j],varB[j])
  # V the variance of the normal distribution used to generate candidates~N(b[i-1],V)
  # nIter: number of iterations of the sampler
  # Details: generates samples from the posterior distribution of a logistic regression using a Metropolis algorithm
  #########################################

  # A matrix to store samples
  p=ncol(X)
  B=matrix(nrow=nIter,ncol=p)
  colnames(B)=colnames(X)

  # A vector to trace acceptance
  accept=matrix(nrow=nIter,ncol=p,NA)
  accept[1,]=TRUE

  # Initialize
  B[1,]=0
  B[1,1]=log(mean(y)/(1-mean(y)))
  b=B[1,]
  #print(b) # Test
  for(i in 2:nIter){

    for(j in 1:p){
      candidate=b
      candidate[j]=rnorm(mean=b[j],sd=sqrt(V),n=1)

      logP_current=logP(y,X,b0=b0,varB=varB,b=b)
      logP_candidate=logP(y,X,b0=b0,varB=varB,b=candidate)
      r=min(1,exp(logP_candidate-logP_current))
      delta=rbinom(n=1,size=1,p=r)

      accept[i,j]=delta

      if(delta==1){ b[j]=candidate[j] }
    }
    B[i,]=b
    if(i%%1000==0){
      message(" Iteration ",i)
    }

  }

  return(list(B=B,accept=accept))
}
```

```{r}
#y: titanic_rm_na$survived
#X: cbind(as.matrix(model.matrix(~survived+sex+pclass+age,data=titanic_rm_na)))

Z=as.matrix(model.matrix(~sex+pclass+age,data=titanic_rm_na))#[,-1]
samples=logisticRegressionBayes(y=titanic_rm_na$survived,X=cbind(Z),nIter=55000)
samples_df <- as.data.frame(samples)
head(samples_df,10)
burn_in <- 5000
samples_post_burn_in <- tail(samples_df, -burn_in)
head(samples_post_burn_in)
nrow(samples_post_burn_in)
```

```{r}
library(coda)
samples_mcmc <- as.mcmc(samples_post_burn_in)
# Clearly we have stationarity
autocorr.plot(samples_mcmc,lag.max = 100)
```

a. Report parameter estimates, posterior standard deviation and 95% posterior credibility regions for each of the regression coefficients.

```{r}
# HPD Interval
HPDinterval(samples_mcmc,prob = .95)
summary(samples_mcmc)
```

b. Report, for each coefficient, the trace plot and estimates of the number of effective samples and the MC standard error.

```{r}
traceplot(samples_mcmc)
effectiveSize(samples_mcmc)
summary(samples_mcmc)
```

c. Use the samples collected to estimate the posterior distribution of the survival probability for male and female in each of the classes (set age to be 35). For each of the groups report a histogram of the posterior density of the survival probability with vertical read lines indicating 95% posterior credibility regions.

```{r}
densplot(samples_mcmc)
```

(3) The function logisticRegressionBayes implements a Metropolis algorithm. With candidates generated from normal distribution with mean equal to

the current sample and variance V. Small values of lambda lead to high rates of acceptance but high correlation between samples. Fit the logistic regression of question 2 using V=.5, V=.1,V=.001,V=.0001, and V=.00005.

```{r}
# V=.5, V=.1,V=.001,V=.0001, and V=.00005
samples1=logisticRegressionBayes(y=titanic_rm_na$survived,X=cbind(Z),nIter=55000,V=.5)
samples2=logisticRegressionBayes(y=titanic_rm_na$survived,X=cbind(Z),nIter=55000,V=.1)
samples3=logisticRegressionBayes(y=titanic_rm_na$survived,X=cbind(Z),nIter=55000,V=.001)
samples4=logisticRegressionBayes(y=titanic_rm_na$survived,X=cbind(Z),nIter=55000,V=.0001)
samples5=logisticRegressionBayes(y=titanic_rm_na$survived,X=cbind(Z),nIter=55000,V=.00005)
```

(a) Report the average acceptance rate and the lag-50 correlation and effective number of samples for the effect of age.

```{r}
samples1df=as.data.frame(samples1)
samples2df=as.data.frame(samples2)
samples3df=as.data.frame(samples3)
samples4df=as.data.frame(samples4)
samples5df=as.data.frame(samples5)

autocorr(as.mcmc(as.data.frame(samples1)),lag=50)
autocorr(as.mcmc(as.data.frame(samples2)),lag=50)
autocorr(as.mcmc(as.data.frame(samples3)),lag=50)
autocorr(as.mcmc(as.data.frame(samples4)),lag=50)
autocorr(as.mcmc(as.data.frame(samples5)),lag=50)

effectiveSize(as.mcmc(as.data.frame(samples1)))
effectiveSize(as.mcmc(as.data.frame(samples2)))
effectiveSize(as.mcmc(as.data.frame(samples3)))
effectiveSize(as.mcmc(as.data.frame(samples4)))
effectiveSize(as.mcmc(as.data.frame(samples5)))
```

(b) What value of V would you recommend? Why?