Bayesian Data Analysis

Anthony R. Colombo, Dr. Paul Marjoram

2022-06-26

Contents

About

This is an independent study of the text **Bayesian Data Analysis** by Andrew Gelman and the more introductory text **A First Course In Bayesian Statistical Methods** by Peter D. Hoff. We will read through most of the chapters and typeset the major definitions. The Gelman text can be difficult, and for difficult chapters, we will lean more on the Hoff textbook.

Independent study

We will typeset each chapter of the Gelman text book, unless the chapter is too difficult, then we will use the introductory text **A First Course In Bayesian Statistical Methods** by Hoff. Each chapter will summarize the definitions, and attempt several problems selected.

Supervised learning

Dr. Paul Marjoram will supervise the learning and have a general oversight to the learning process.

Chapter 1

Fundamentals of Bayesian Inference

The first few chapters of Gelman's text are introductory, and we attempt to highlight the key definitions and summarize each chapter. At the end of each chapter we attempt several problems. Probability and inference is defined using three steps

1. setting up the full probability model for a joint distribution for all observable and unobservable quantities.
2. Conditioning on observed data: computing the appropriate *posterior* distribution, the conditional probability distribution of the unobserved quantities of oltimate interest, given the observed data.
3. Evaluating the fit of the model.

1.1 General notation for statistical inference

There are two different kinds of estimands, the first are potentially observable quantities, such as future observations of a process, and the second are quantities that are not directly observable, namely the parameters that govern a process being investigated.

Exchangeability

One key assumption is that the n values y_i are regarded as *exchangeable*, meaning that the uncertainty can be expressed as a joint probability $p(y_1, ..., y_n)$ that is invariant to permutations of indexes. Often times the exchangeable distribution is modeled as *iid*.

Explanatory variables

It is common to have observations on each unit which have non-random variables called *explanatory variables* or *covariates*. The explanatory variables are usually denoted by X. However treating X as random then exchangeability can be extended $(x, y)_i$ which is invariant to permutations of the indexes. Further, it is always appropriate to assume exchangeability of y, conditioned on sufficient information of X, where the indexes can be thought of as randomly assigned. It follows that if two units have the same value of x, then the distributions of y are the same.

Hierarchical modeling

for a model across patients across different cities, we can assume exchangability to patients within a city. Further conditioned on the explanatory variables at the individual, the conditional distribution given these explanatory variables would be exchangeable.

1.2 Bayesian inference

The prior, p(θ), and the sampling distribution, or the *data distribution*, $p(y|\theta)$ is related to the joint distribution by

$$p(\theta, y) = p(\theta)p(y|\theta)$$

Where using Bayes' rule the posterior distribution

$$p(\theta|y) = \frac{p(\theta)p(y|\theta)}{p(y)} \tag{1.1}$$

Where $p(y) = \int p(\theta)p(y|\theta)d\theta$, or a sum in discrete case. An equivalent form of (1.1) is the *unnormalized posterior density* given as

$$p(\theta|y) \propto p(\theta)p(y|\theta) \tag{1.2}$$

Note that $p(y|\theta)$ is taken as a function of θ, not of y.

Prediction

Inferences about an unknown *observable* variable, are called predictive inferences. Before the data y are considered, the distribution of the unknown, observable, y is

$$p(y) = \int p(y,\theta)d\theta = \int p(\theta)p(y|\theta)d\theta$$

this is defined as the marginal distribution of y, and also called *prior predictive distribution*. Prior refers that the data is not conditional on any previous observation, and predictive refers to the data being observable.

The *posterior predictive distribution* is conditional on the observed y, but is predictive because it is predicting observable values.

$$\begin{aligned} p(\hat{y}|y) &= \int p(\hat{y},\theta|y)d\theta \\ &= \int p(\hat{y}|\theta,y)p(\theta|y)d\theta \\ &= \int p(\hat{y}|\theta)p(\theta|y)d\theta \end{aligned} \tag{1.3}$$

Likelihood

The data y affects the posterior inference only through (1.2) likelihood function $p(y|\theta)$ which is regarded as a function of θ for fixed y. The *likelihood function* is defined as $p(y|\theta)$, and the *likelihood principle* is for any given sample, and any two likelihood models $p(y|\theta)$, two models with the same likelihood will have the same inference for θ.

Subjectivity and Objectivity

The frequentist models, MLEs, have subjectivity in their assumptions because they rely on long sequence of identical trials, that are iid. The Bayesian model relies on the prior distribution. If any experiment is repeatable and can replicated, the the prior distribution can be estimated from the data themselves and the analysis is more ‘objective’. Replication increases objectivity of a given model. However the Bayesian approach allows for (1) the ability to combine information from multiple sources (allowing for greater objectivity) and (2) more encompassing by accounting for uncertainity about the unknowns in a statistical problem.

It is important to include as much background information as possible

1.3 Exercises

1. Suppose for $\theta = 1$, then y$\sim$ $N(1,\sigma)$, and if $\theta = 2, y \sim N(2,\sigma)$. Where P$(\theta = 1) = P(\theta = 2) = 0.5$.

   (a) For $\sigma = 2$; we must write the formula for the pdf of y. $p(y) = \sum_\theta p(y|\theta)p(\theta) = (1/2)N(1,\sigma^2) + (1/2)N(2,\sigma^2)$ as the marginal density.

```
 fy<-function(y) {return(0.5*dnorm(y,mean=1,sd=2)+0.5*dnorm(y,2,sd=2))}
```

```
(b) $P(\theta=1 | y=1) = \frac{p(\theta=1)p(y|\theta=1)}{p(y)}= \frac{(1/2)N(1,4)}{(1/
```

```
dy<-function(y){ return( (1/2)*dnorm(y,mean=1,sd=2)/fy(y))}
dy(1)
```

```
## [1] 0.5312094
```

4. twelve games with point spread of 8 points.

   (a) Using relative frequency, P(favorite wins | point spread =8) = 0.67.
   P(favorite wins by at least 8 | point spread =8) = 0.42.
   and P(fav. wins by at least 8 | spread =8, favorite team wins) = 0.62.

```
spread<-8

## outcome of the games favor score - underdog score
 games<-c(-7,-5,-3,-3,1,6,7,13,15,16,20,21)
## frequentist approach
 fav.wins<-  mean(games>0)
 message(paste0("(frequentist): fav wins: ", round(fav.wins,2)))
```

```
## (frequentist): fav wins: 0.67
```

```
 fav.by.8<- mean((games>8))
  message(paste0("(frequentist): fav wins by 8: ", round(fav.by.8,2)))
```

```
## (frequentist): fav wins by 8: 0.42
```

```
  ## P( fav. wins >8 | fav. wins) = P(fav. wins > 8, fav. wins )/ P(fav. wins)
 cond<-sum(games>8 & games>0)/sum(games>0)
  c<-fav.by.8/fav.wins
 message(paste0("(frequentist): fav wins by 8 given fav. wins: ", round(cond,2)))
```

```
## (frequentist): fav wins by 8 given fav. wins: 0.62
```

(b) now we assume a normal distribution with $d|x \sim N(-1.25, 10.10)$. So $P(d > -x) = P(Z\sigma + \mu > -x) = P(Z > -x - \mu/\sigma)$

(c) Probablity fav team wins is 0.75

(ii) fav team wins by 8 (beats the spread) is 0.45, we expect this to be 0.5 (the middle of the normal distribution because we centered on the spread)

(iii) P(wins by 8 | favorite team wins) = P(favorite team wins | wins by 8)P(wins by 8)/P(favorite team wins) = P(wins by 8)/P(fav. team wins) since the conditional prob. =1 given the favorite team wins. The prob. that they win by at least 8 is 0.6.

```
## part b
  d<-games-8
  sample.mean <-mean(d)
  sample.sd<-sd(d)
  ## assume  d|x ~ N(0,10.10)
  fav.wins.norm<- 1-pnorm(-8,mean=sample.mean,sd=sample.sd)
   message(paste0("(normal): fav wins: ", round(fav.wins.norm,2)))
```

```
## (normal): fav wins: 0.75
```

```
  fav.by.8.norm<-1-pnorm(0,mean=sample.mean,sd=sample.sd)
    message(paste0("(normal): fav wins by 8: ", round(fav.by.8.norm,2)))
```

```
## (normal): fav wins by 8: 0.45
```

```
## Pr(Wins by 8 | Fav. wins) = P(Fav. wins | wins by 8)P(wins by 8) / P(fav. wins)
    ## P(Fav. wins | wins by 8) = 1
  cond.norm<-fav.by.8.norm/fav.wins.norm
   message(paste0("(normal): fav wins by 8 given fav. wins: ", round(cond.norm,2)))
```

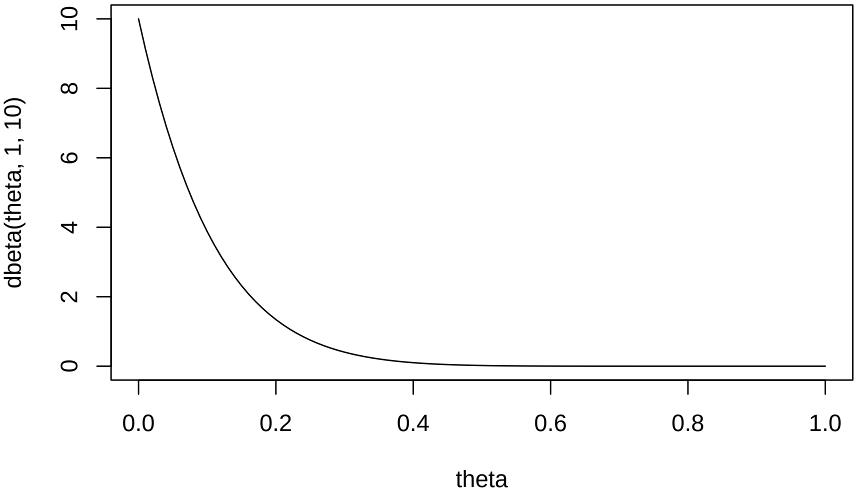
```
## (normal): fav wins by 8 given fav. wins: 0.6
```

5. We need to estimate the probability that there is at least one congressional election that is tied in the next U.S. election. There are 435 senate elections.

   (a) The parameters of interest are θ_i the true probability that the election is tied. We can let the *prior* $\theta \sim Beta(\alpha, \beta)$. The *likelihood* is $y|\theta_i \sim Binomial(435, \theta_i) = \theta^{\sum y_i}(1-\theta)^{435-\sum y_i}$ follows a Binomial distribution (ignoring the binomial coefficient) where we assume each election is independent. Hence the posterior for theta

$f(\theta|y) \sim Beta(\sum y_i + \alpha, n - \sum y_i + \beta)$. where α, β are set to 1 for the uniform prior. For this case we set α, β equal to 1, 10 which has a prior mean of 0.09.

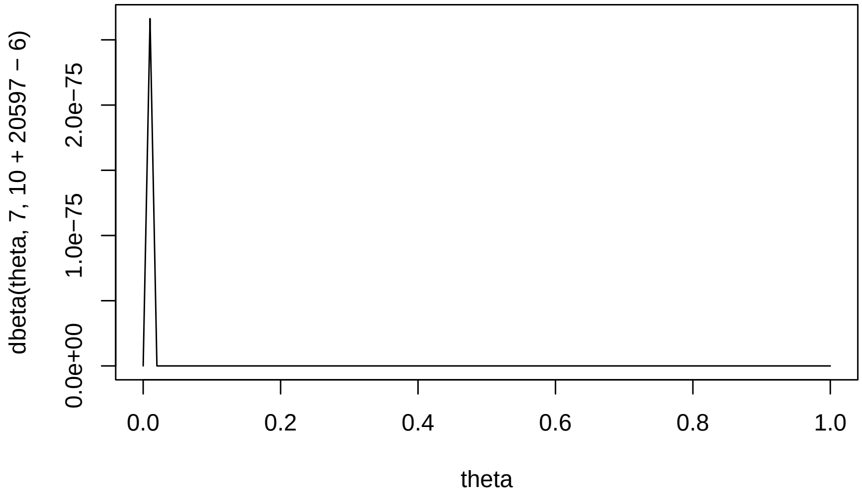
```
theta=seq(from=0,to=1,by=.01)
plot(theta,dbeta(theta,1,10),type='l')
```



(b) In the period of 1900-1992, there were 20,597 elections, out of which 6 were decided by less than 10 votes, and 49 were decided by less than 100 votes.
we can estimate the probability of a tie to be less than 6/20,597 and bounded by 49/20,597. So for the Binomial trials the sum of the successes is 6, and n=20,597, so the posterior could be $\theta|y \sim Beta(1+6, 10+20,597-6)$ is the posterior for θ. This assumes that 10 votes is within the neighborhood of an election tie.
The question asks to compute at least one election tie, from a total of 435 elections. This follows a Binomial(435, $\hat{\theta}$). Where we use the posterior mean to estimate θ. The posterior mean using the Beta(7,20601) yields a mean of $\hat{\theta} = \frac{7}{20608} = 3.4e-04$ as the posterior mean.

Then the probability that at least 1 election is tied, from 435 total elections will follow a Binomial(435, $\hat{\theta}$), where we can use the posterior distribution for $\theta|y$ in the Binomial likelihood $P(X \geq 1|\hat{\theta}) = 1 - P(X \leq 0|\hat{\theta})$ which has a probability of 0.14 of at least 1 election tie.

```
# the posterior for theta is Beta(1+6,10+20597-6)
plot(theta,dbeta(theta,7,10+20597-6),type='l')
```



```
 ## posterior mean is 7/(20601)
## then P(X>=1) = 1-P(X<=0 | p)
  1-pbinom(0,435,prob=7/20608)
```

```
## [1] 0.1373819
```

9. A clinic has three doctors. Patients come into the clinic at random, starting at 9 a.m. according to a Poisson process, with a time parameter, t, of 10 minutes; that is after opening the first patient appears follows an exponential distribution with average waiting time of 10 minutes. Then the next patient arrives with a waiting time of an expected 10 minutes as iid exponential distribution. After a patient arrives, the patient waits until a doctor is available, and the doctor visits a patitient uniformly between 5-20 minutes. The clinic stops admitting patients at 4 pm, and closes after the last patient is completed with the visit.

   (a) Simulate this process once. how many patients visited the office? how many had to wait for a doctor? what was the average wait? when did office close?

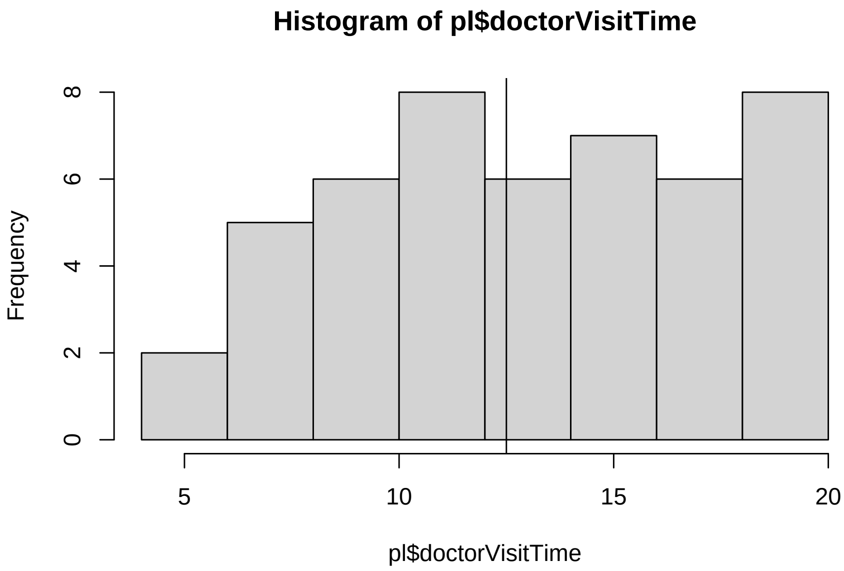
```
### waiting time for a new patient to arrive in the clinic
###########################################################
 # patientList is the data frame of all patients
 # closeTime is the time to stop admitting (420 minutes)
 # currentPatient Number
 # current time is the running total of time
  newPatientArrival<-function(patientList,
                             closeTime=timeToClose,
                              waitTime,
                             visitTime,
                             currentPatientNumber=0,
                             currentTime,
                             assignedDoctor="none",
                             completionTime=0){
    # waiting time for next patient
     patientTime<-round(rexp(1,rate=1/10),2)
    # current time of existing patients
     current<-max(patientList$currentTime)
   ## the clinic stops admitting patients at 4pm
   if( (current+patientTime)<=closeTime){
     ## in minutes
   newPatient<-createPatientChart(currentPatientNumber,patientTime,waitTime,visitTime,c
    }else{
    newPatient<-createPatientChart(currentPatientNumber,patientTime,waitTime,visitTime
   }
   return(newPatient)
 }
####################



 computeWaitTime<-function(doctors=NULL,
                           patientList=NULL,
                           patientID=1){
   ## need to compute visiting time (booked)
   ## next time available
   ## required input current time for a specific doctor/patient ?
   # patient time (minutes)

   ## FIX ME: it is grabbing 2 patient IDs?
   currentTime<-patientList$currentTime[which(patientList$patient==patientID)]
   visitTime<-runif(1,min=5,max=20) ## minutes


   if(any(doctors$nextTimeAvail<currentTime)){
```

```
  waitTime=0
  assignedDr<-sample(doctors$dr[which(doctors$nextTimeAvail<currentTime)],1)
  ### current time + visitTime
  nextAvailTime<- visitTime+currentTime+waitTime
  ## completion time for patient exit (closing time).
 }else if(any(doctors$nextTimeAvail<currentTime)==FALSE){
  # all doctors are booked, no available doctors.
  # wait time is the difference between next available time (assuming all times are greater th
  waitTime<-min(doctors$nextTimeAvail-currentTime)
  assignedDr<-doctors$dr[which( (doctors$nextTimeAvail-currentTime)==min(doctors$nextTimeAvail
   if(length(assignedDr)>1){
    assignedDr<-assignedDr[1]
   }
  nextAvailTime<- visitTime+currentTime+waitTime  ## completion time for patient to exit
  }## if all doctors unavail
  #print(assignedDr)
  #print(currentTime)
  ##  update doctor list
  doctors[which(doctors$dr==assignedDr),'visitingPatient']<-patientID
  doctors[which(doctors$dr==assignedDr),'nextTimeAvail']<-nextAvailTime
  doctors[which(doctors$dr==assignedDr),'currentTime']<-currentTime ## patient time
  doctors[which(doctors$dr==assignedDr),'visitTimeLength']<-visitTime
  # flag avail to no.
  doctors[which(doctors$dr==assignedDr),'avail']<-'no'
  ## update patient list
  patientList[which(patientList$patient==patientID),'doctorWaitTime']<-waitTime
  patientList[which(patientList$patient==patientID),'doctorVisitTime']<-visitTime
  patientList[which(patientList$patient==patientID),'assignedDoctor']<-assignedDr
  patientList[which(patientList$patient==patientID),'completionTime']<-nextAvailTime
 return(list(patient=patientList,doctor=doctors))
}


## creates a patient object
createPatientChart<-function(currentPatientNumber,arrivalTime,waitTime,visitTime,currentTime,ass
 patientID<-data.frame(patient=currentPatientNumber+1,
                       arrivalTime=arrivalTime,
                       doctorWaitTime=waitTime,
                       doctorVisitTime=visitTime,
                       currentTime=currentTime,
                       assignedDoctor=assignedDoctor,
                       completionTime=0)
 return(patientID)
}
```

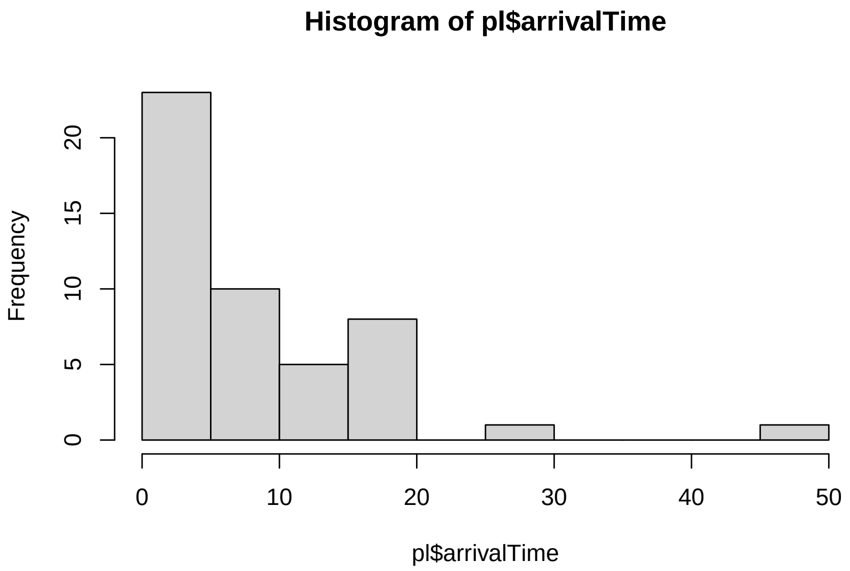
```
 updatePatientList<-function(patientList,patientID){
   patientList<-rbind(patientList,patientID)
   return(patientList)
 }

updateTime<-function(currentTime,newTime=NULL,p1){
  p1$currentTime<-currentTime+newTime
  return(p1)
}
 totalPatients<-0
## this is the simulation
 ## first task : loop through the time update for patients
 ## second task : include the doctor assignment query.
simulateProcess<-function(doctors=NULL,
                          totalWait=NULL,
                          totalPatients=0,
                          timeToClose=420,
                          currentTime=NULL){
  ## initiate Patient List
 patientList<-data.frame(patient=0,
                         arrivalTime=0,
                         doctorWaitTime=0,
                         doctorVisitTime=0,
                         currentTime=0,
                         assignedDoctor='none',
                         completionTime=0)
 ## not sure what to put here.
 currentTime<-patientList$currentTime[which(patientList$patient==max(patientList$patien
 currentPatientNumber<-0

 ## timeToClose (minutes) is stopping to admit patients
  while(currentTime<timeToClose){
    ## patient enters after the (i-1) patient enters.
    p1<-newPatientArrival(patientList,
                             closeTime=timeToClose,
                          waitTime=0,
                          visitTime=0,
                             currentPatientNumber=currentPatientNumber,
                             currentTime)
    ## update time
    p1<-updateTime(p1$currentTime,newTime=p1$arrivalTime,p1)

    # given a patient time, switch the availability of any doctor
    # if a doctors next available time is less than the current time, switch him to av
    ## FIX ME: need to ensure this flag is correct.
    if(any(doctors$nextTimeAvail<p1$currentTime)){
      doctors$avail[which(doctors$nextTimeAvail<p1$currentTime)]<-'yes'
    }

    ## create a patient list
    if(currentPatientNumber==0){
      patientList<-p1
     # update patient number
      currentPatientNumber<-currentPatientNumber+1
    }else{
      patientList<-rbind(patientList,p1)
     # update patient number
      currentPatientNumber<-currentPatientNumber+1
    }

    ## task 2 assign a doctor
     ### check for doctor availability
     ## compute wait time, and/or compute the next available time
     ## returns a list object.
    clinicList<-computeWaitTime(doctors,patientList,patientID=patientList$patient[currentPatientN

    doctors<-clinicList[["doctor"]]
   patientList<-clinicList[["patient"]]
   ## update flags
     # update currentTime
    ## current time is cumulative sum of the arrival times.
   currentTime<-patientList$currentTime[which(patientList$patient==max(patientList$patient))] ##

   ## fix me:
   ## reset doctor availability based on current patient time.
   upID<-which(doctors$nextTimeAvail<currentTime)
   doctors$nextTimeAvail[upID]<-currentTime
   doctors$currentTime[upID]<-currentTime
   doctors$visitTimeLength[upID]<-0
  }## while loop
  return(list(patient=patientList,doctors=doctors))
}
```

```
 doctors<-data.frame(dr=c('a','b','c'),
                     visitingPatient=c(0,0,0), ## who is doctor seeing (patient ID)
                     visitTimeLength=c(0,0,0), # length of doctor visit U(5,20)
                     currentTime=c(0,0,0),    ## current Time
                     nextTimeAvail=c(0,0,0),  ## current time + visitTimeLength = next avail time
                     avail=c("yes","yes","yes"))
## initiate times
```

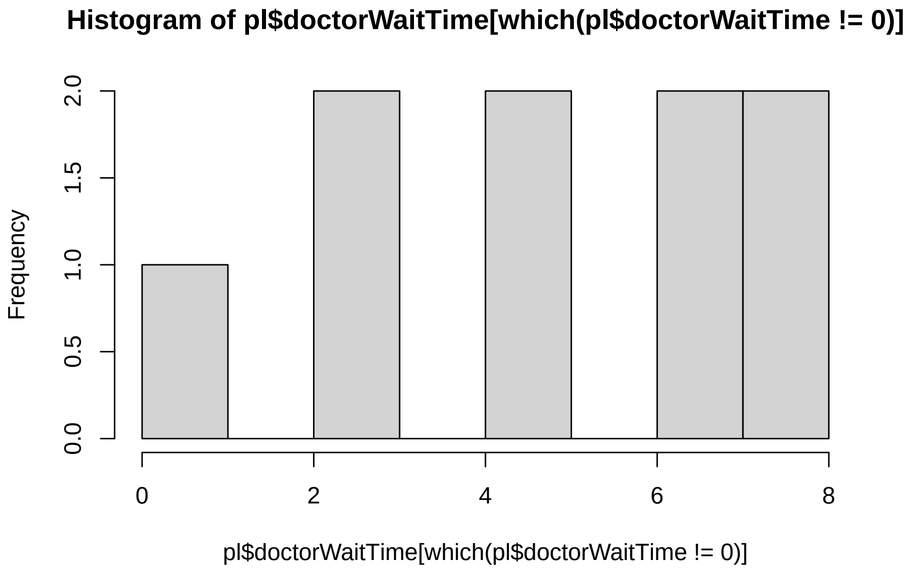
```
 totalWait<-0
 currentPatientNumber<-0
 ## clinic opens at 9am -4pm that is 7 hours (420 min.)
 timeToClose<-7*60 ## stops admiting patienets in 420 minutes
 ## current time is 0
 ## this will be the running total of minutes.
 currentTime<-0



res<-simulateProcess(doctors,
                          totalWait,
                          totalPatients,
                          timeToClose,
                          currentTime)

pl<-res$patient[which(res$patient$currentTime<=420),]

 hist(pl$doctorVisitTime)
 abline(v=(20+5)/2) ## should be ~12
```



```
 hist(pl$arrivalTime) ## should be close to 10  exp(1/10) has mean 10
```



```
hist(pl$doctorWaitTime[which(pl$doctorWaitTime!=0)]) ## about 2.41
```



```
  print(max(pl$completionTime)-420) ## closing time
```

```
## [1] 10.1471
```

```
  print(max(pl$patient)) ## total patient  should be 42
```

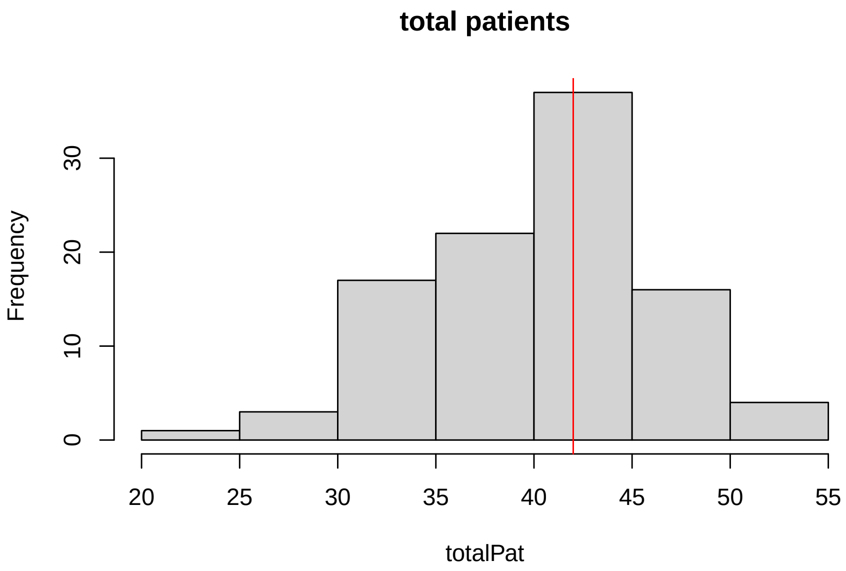
```
## [1] 48
```

```
## (20-5)/6 + 10 this is about 12.5 minutes of arrival + visit time. which is approxima
## the arrival time is about 10 minutes.

## we should expect 42 patients
 #420/10

  ## sanity check
  #all(pl$currentTime+pl$doctorWaitTime+pl$doctorVisitTime-pl$completionTime==0)
```

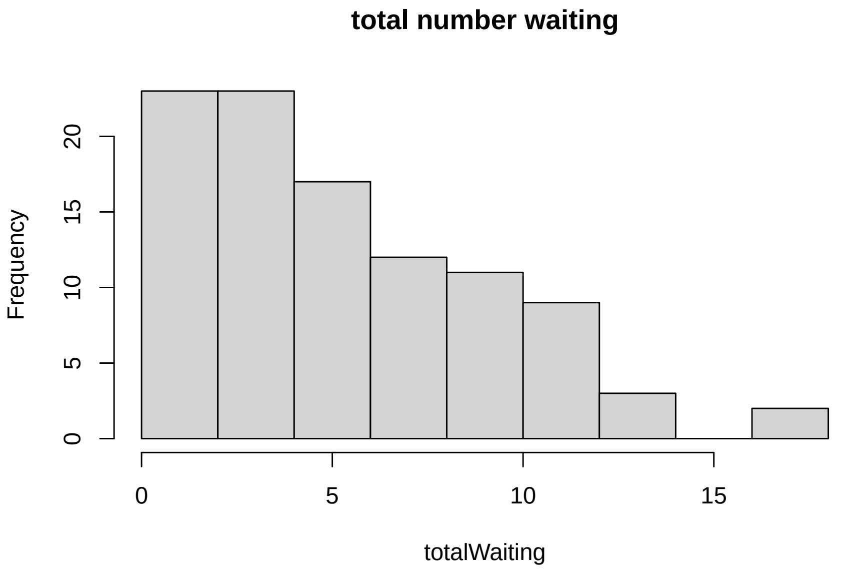
Simulation 100 times

total number of patients was approximately 42, which we expect since the total 420/10. The total number waiting with 3 doctors is 6.61 for 1 day. the average waiting time was about 4-5 minutes. For 1 day, the average closing time was 5.32 minutes after 4 pm

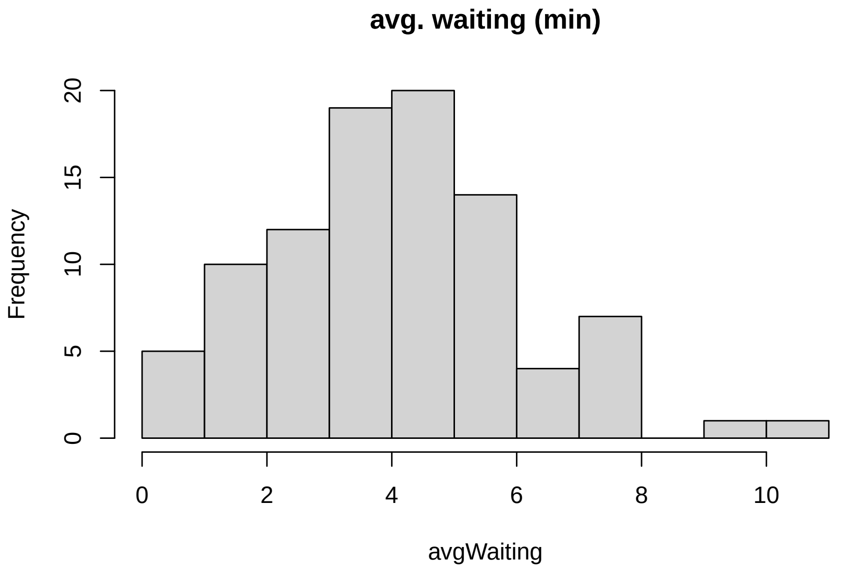
```
 totalPat<-NULL
 totalWaiting<-NULL
 avgWaiting<-NULL
 closing<-NULL
 patientList<-NULL
 p1<-NULL

for(i in 1:100){

 doctors<-data.frame(dr=c('a','b','c'),
                     visitingPatient=c(0,0,0), ## who is doctor seeing (patient ID)
                     visitTimeLength=c(0,0,0), # length of doctor visit U(5,20)
                     currentTime=c(0,0,0),    ## current Time
                     nextTimeAvail=c(0,0,0),  ## current time + visitTimeLength = next
                     avail=c("yes","yes","yes"))
## initiate times
 totalWait<-0
 totalPatients<-0
 currentPatientNumber<-0
 ## clinic opens at 9am -4pm that is 7 hours (420 min.)
 timeToClose<-7*60 ## stops admiting patienets in 420 minutes
 ## current time is 0
 ## this will be the running total of minutes.
 currentTime<-0


res<-simulateProcess(doctors,
                          totalWait,
                          totalPatients,
                          timeToClose,
                          currentTime)

pl<-res$patient[which(res$patient$currentTime<=420),]


 totalPat<-c(totalPat,max(pl$patient))
  totalWaiting<-c(totalWaiting,nrow(pl[which(pl$doctorWaitTime!=0),]))
 avgWaiting<-c(avgWaiting,mean(pl[which(pl$doctorWaitTime!=0),"doctorWaitTime"]))
 closing<-c(closing,max(pl$completionTime))

}

 hist(totalPat,main="total patients")
 abline(v=420/10,col='red')
```



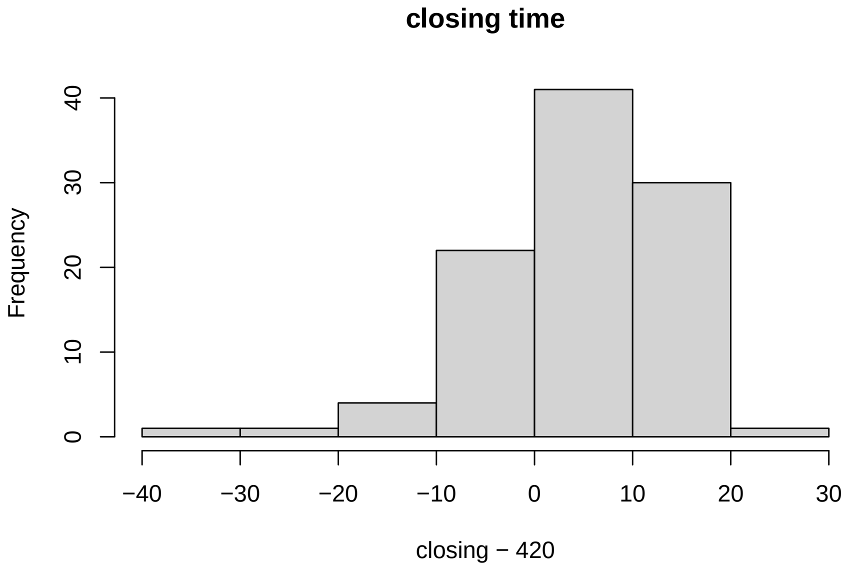
```
hist(totalWaiting,main="total number waiting")
```



```
hist(avgWaiting,main="avg. waiting (min)")
```



```
hist(closing-420,main="closing time")
```



Chapter 2

Single parameter models

2.1 Estimating a probability from binomial data

$$p(y|\theta) = \binom{n}{y}\theta^y(1-\theta)^{n-y} \tag{2.1}$$

To perform Bayesian inference we assume $\theta \sim U(0,1)$ where the posterior is

$$p(\theta|y) \propto \theta^y(1-\theta)^{n-y} \tag{2.2}$$

which is the form of a `beta` distribution $\theta|y \sim Beta(y+1, n-y+1)$

2.2 Posterior as a compromise between data and prior information

The posterior is less variable than the prior because it incorporates the information from the data.

$$E(\theta) = E(E(\theta|y)) \tag{2.3}$$

$$V(\theta) = E(V(\theta|y)) + V(E(\theta|y)) \tag{2.4}$$

where $\theta|y$ is the posterior. So the average of the prior, is the average of the posterior means over the distribution of possible data. The variance of the prior (2.4) says the posterior variance is on average smaller than the prior variance.

2.3 Summarizing the posterior inference

The mean, median, mode, and standard deviation of the posterior distribution summarize the all the current information about a model.

Posterior quantiles and intervals

The posterior uncertainty can be reported by presenting the quantiles of the posterior distribution. The interval, a *central interval of posterior probability* corresponds to the case of $100(1-\alpha)\%$, to the range of values above and below which lies exactly $100(\alpha/2)\%$ of the posterior probability. The interval estimates are *posterior intervals.* This differences from the confidence interval because the confidence interval is not a probability interval, because either the parameter is within the region or it is not, but the confidence interval provides information in the long run over repeated experimentation as to how many experiments would contain the true parameter.

There is also the *highest posterior interval* which is a probabilistic interval that is not less than any region outside of the interval.

2.4 Informative prior distributions

the property that the posterior distribution follows the same parametric form as the prior distribution is called *conjugacy.* Where the beta prior distribution is a *conjugate family* for the binomial likelihood.

so given the binomial likelihood $p(y|\theta) \propto \theta^a(1-\theta)^b$, and a prior density $p(\theta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$ the posterior is of the beta family.

$$\begin{aligned} p(\theta|y) &\propto \theta^y(1-\theta)^{n-y}\theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &= \theta^{y+\alpha-1}(1-\theta)^{n-y+\beta-1} \\ &= Beta(\theta|\alpha+y, \beta+n-y) \end{aligned}$$

Conjugate prior distributions

Conjugacy is formally defined as if F is a class of sampling distributions $p(y|\theta)$, and P is a class of prior distributions for θ, then the class P is conjugate for F if $p(\theta|y) \in P$ for all $p(.|\theta) \in F$ and $p(.) \in P$.

Conjugate prior, distributions, exponential families, and sufficient statistics

Posterior distributions can be derived using sufficient statistics from exponential families. The exponential family is defined as

$$p(y_i|\theta) = f(y_i)g(\theta)e^{\phi(\theta)^T u(y_i)}$$

Where $\phi(\theta), u(y_i)$ are vectors of equal dimension to that of θ. The $\phi(\theta)$ is called the *natural parameter* for the family (F). The likelihood of a sequence $y = (y_1, ..., y_n)$ iid is

$$\begin{aligned} p(y|\theta) &= (\prod_{i=1}^{n} f(y_i))g(\theta)^n exp(\phi(\theta)^T \sum_{i=1}^{n} u(y_i)) \\ &\propto g(\theta)^n e^{\phi(\theta)^T t(y)}, t(y) = \sum_{y=1}^{n} u(y_i) \end{aligned}$$

The *sufficient statistic* for θ is $t(y)$ because the likelihood for $ depends on the data, y, only through the value of t(y).

Sufficient statistics benefit posterior distributions because if the prior density is specified as

$$p(\theta) \propto g(\theta)^\eta e^{\phi(\theta)^T \nu}$$

Then the posterior density using sufficient statistics is

$$p(\theta|y) \propto g(\theta)^{\eta+n} e^{\phi(\theta)^T (\nu + t(y))}$$

Exponential families are the only classes of distributions that have natural conjugate prior distributions.

2.5 Normal distribution with known variance

The normal distribution is foundational to statistical modeling, with the central limit theorem (CLT) allowing for the use of normal likelihood in many statistical problems which can approximate complex likelihoods. If the normal distribution does not provide a good model fit, finite mixtures of distributions can identify useful solutions.

Likelihood of one data point

With mean θ and known variance σ^2 the sampling distribution of a given point is defined

$p(y|\theta) = \frac{1}{\sqrt{2*\pi*\sigma^2}} e^{-\frac{1}{2\sigma^2}(y-\theta)^2}$

2.5.1 Conjugate prior and posterior distributions

The prior has the exponential family form given as $\theta \sim N(\mu_0, \tau_0^2)$

$$p(\theta) \propto exp(\frac{1}{2\tau_0^2}(\theta - \mu_0)^2)$$

WHere completing the square can find the posterior distribution

$$\begin{aligned} p(\theta|y) &\propto exp(-1/2(\frac{(y-\theta)^2}{\sigma^2} + \frac{(\theta-\mu_0)^2}{\tau_0^2})) \\ p(\theta|y) &\propto exp(\frac{1}{2\tau_1^2}(\theta-\mu_1)^2), \theta|y \sim N(\mu_1, \tau_1^2) \end{aligned}$$

where $\mu_1 = \frac{1/\tau_0^2\mu_0 + 1/\sigma^2 y}{1/\tau_0^2 + 1/\sigma^2}$ and $1/\tau_1^2 = 1/\tau_0^2 + 1/\sigma^2$

In manipulating the distributions the inverse of the variance is defined as the *precision*. The posterior precision is equal to the prior precision plus the data precision. And the posterior mean is a weighted average of the prior mean and the observed value, y, proportional to the total precision.

Posterior predictive distribution

the posterior predictive distribution of a future observation, x, p(x|y) can be calculated

$$p(x|y) = \int p(x|\theta)p(\theta|y)d\theta \quad \propto \int exp(-1/2\sigma^2(x-\theta)^2)exp(-1/2\tau_1^2(\theta-\mu_1)^2)d\theta$$

the future observations, x, does not depend on the past observations y given θ.

Normal model with multiple observations

For multiple observations, y, the posterior density is formulated as:

$$\begin{aligned} p(\theta|y) &\propto p(\theta)p(y|\theta) \\ &= p(\theta)\prod_i p(y_i|\theta) \\ &\propto exp(-1/2\tau_0^2(\theta-\mu_0)^2)\prod_i exp(-1/2\sigma^2(y_i-\theta)^2) \\ &\propto exp(-1/2(1/\tau_0^2(\theta-\mu_0)^2 + 1/\sigma^2\sum_i(y_i-\theta)^2)) \end{aligned}$$

Simplfying the algebra shows the posterior depends on y only through the sample mean (sufficient statistic), $\bar{y}$ is the sufficient statistic for θ, and the final model is $\bar{y}|\theta, \sigma^2 \sim N(\theta, \sigma^2/n)$

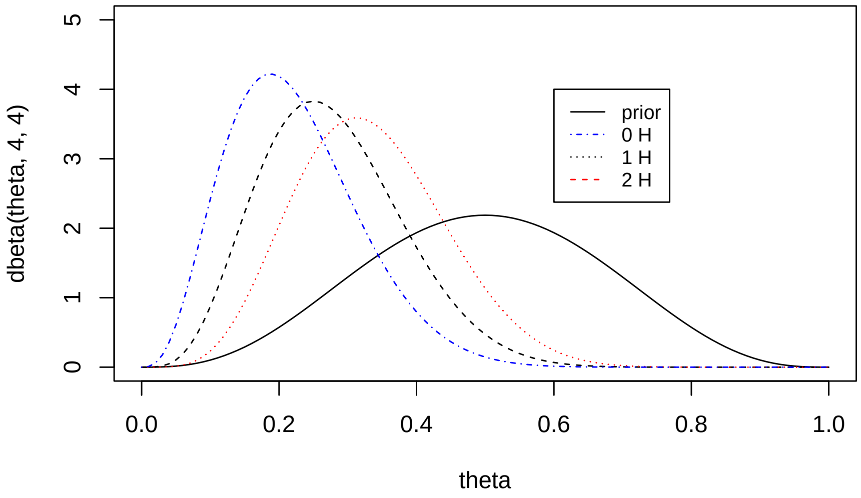
2.6 Other standard single-parameter models

2.7 Exercises

Question 1

prior Beta(4,4), where a coin is tossed 10 times and heads appears fewer than 3 times. the exact posterior is Beta(4+y, 4+10-y) for y=0,1,2. Since we don't know the observed heads, but that $y < 3$ we plot the posterior distributions for each possibility. For 2 heads it is closer to the prior, with posterior mean of 0.33, which is closest to the prior mean of 1/2.

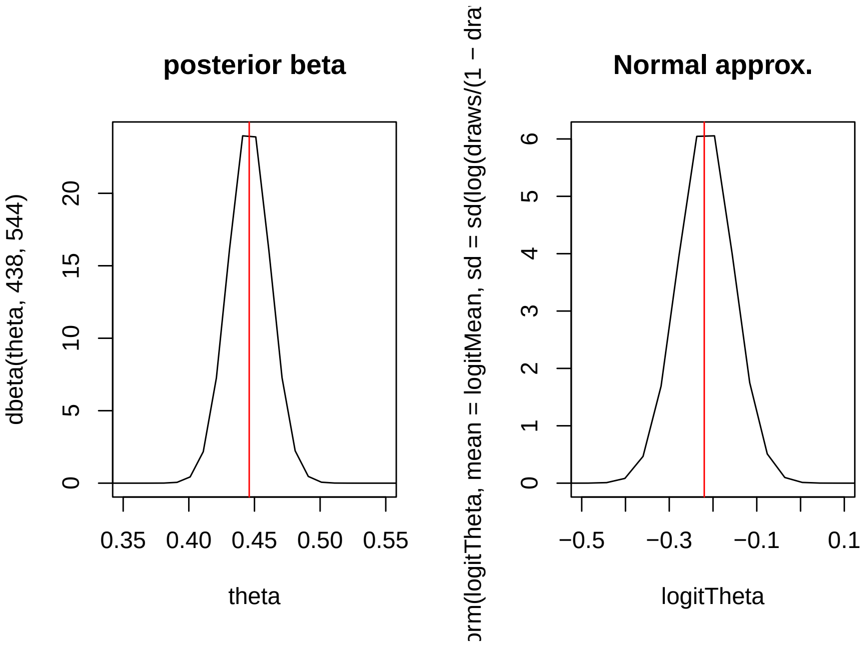
```
theta<-seq(from=0,to=1,by=0.01)

plot(theta,dbeta(theta,4,4),type='l',ylim=c(0,5)) ## prior
lines(theta,dbeta(theta,4+1,4+10-1),lty=2) ## 1 success
  lines(theta,dbeta(theta,4+2,4+10-2),lty=3,col='red') ## 2 succe
  lines(theta,dbeta(theta,4,4+10),lty=4,col='blue')  ## 0 successes
 legend(0.6, 4,
        legend=c("prior", "0 H", "1 H", "2 H"),
      col=c("black","blue","black", "red"), lty=c(1,4,3,2), cex=0.8)
```



Normal approximation example

For female births we have beta(438,544) we use the normal approximation. This replicates Gelman's Figure 2.3 (a, b)

```
theta<-seq(from=0.001,to=1,by=0.01)
## example births
 postMean <-function(alpha,beta,y,n){
   return( (alpha+y)/(alpha+beta+n))
 }
 postVar<-function(alpha,beta,y,n){
   return( ((alpha+y)*(beta+n-y))/( (alpha+beta+n)^2*(alpha+beta+n+1)) )
 }
 sdnorm<-sqrt(postVar(438,544,0,0))
 logitMean<-log( postMean(438,544,0,0)/(1-postMean(438,544,0,0)))

logitTheta<-log(theta/(1-theta))

par(mfrow=c(1,2))
  plot(theta,dbeta(theta,438,544),type='l',xlim=c(0.35,0.55),main="posterior beta") #
  abline(v=0.446,col='red')
draws<-rbeta(1000,438,544)
plot(logitTheta,dnorm(logitTheta,mean =logitMean, sd=sd(log(draws/(1-draws))) ),type=
  abline(v=-0.22,col='red')
```



Question 3

The prior predictive distributions for the number of 6's in a fair roll, tossed 1,000 times will follow a beta distribution. Let y be the number of 6's in 1000 rolls of fair die, the probability for a 6 is 1/6, so the number of 6's in this trial is approximatley 167, and 833 failures as the prior prediction.

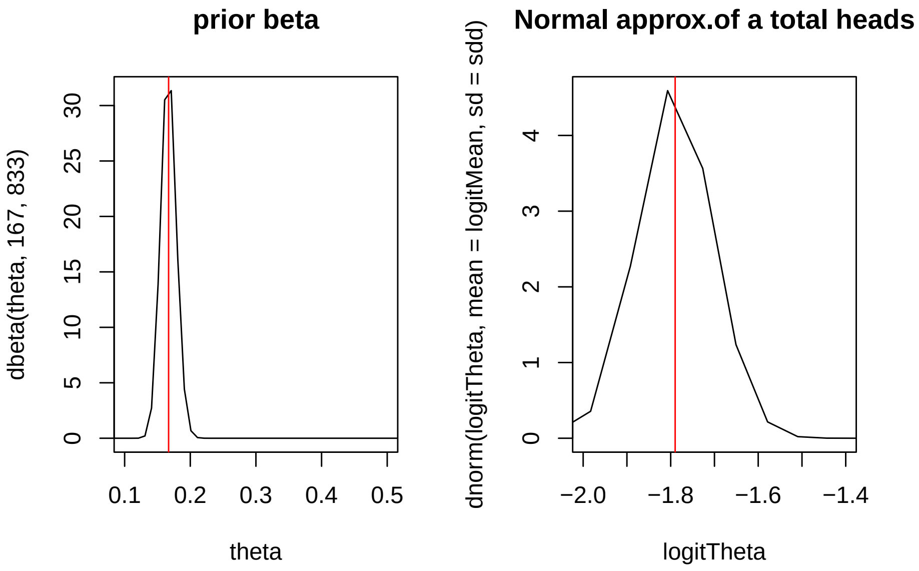
We plot the beta distribution of the expected number of heads in 1000 tosses.

The normal approximation shows the probability of heads in a given 1000 tosses, using a non-informative prior beta(167,833) which has a prior predictive mean of $exp(-1.79)/(1 + exp(-1.79)) = 0.143$. This is not the same for number of success in 1000 tosses.

We find the probability distribution of a given success and the prior probability predictive interval follows a beta with 95% (0.12,0.17) for hte probability of rolling a 6

```
 ## based on normal approximation sketch the distribution of y
 ## for normal we use the logit transform
 ## n = 1,000
 ## first lets construct a beta distribution.
 ## let the prior be beta(4,4) or even beta(1,1)
par(mfrow=c(1,2))
```

```
  plot(theta,dbeta(theta,167,833),type='l',xlim=c(0.1,0.5),main="prior beta") ## prio
 # plot(theta,dbeta(theta,1,5),type='l',xlim=c(0.1,0.5), main='prior beta')
  abline(v=0.167,col='red') ##
  #  qbeta(c(0.05,0.25,0.5,0.75,0.95),167,833)
   ## predictive on normal scale.
    draws<-rbeta(1000,167,833)
    logitMean<-log( 167/1000)
 sdd<-sd(log(draws/(1-draws)))


 plot(logitTheta,dnorm(logitTheta,mean =logitMean, sd=sdd ),type='l',main="Normal app
   abline(v=logitMean,col='red')
```



```
   ### the number of heads.
 message("prior predictive mean:" ,round( exp(logitMean)/(1+exp(logitMean)),2))
```

```
## prior predictive mean:0.14
```

```
qx<-(qnorm(c(0.05,0.25,0.5,0.75,0.95),mean =logitMean, sd=sdd ))
## this is the probability of landing 6 for theta
exp(qx)/(1+exp(qx))
```

```
## [1] 0.1267783 0.1362061 0.1431020 0.1502862 0.1611396
## wald 95% prior interval
c( exp(logitMean-1.96*sdd)/(1+exp(logitMean-1.96*sdd)),exp(logitMean+1.96*sdd)/(1+exp(logitMean+1
```

```
## [1] 0.1238386 0.1647981
```

Question 4

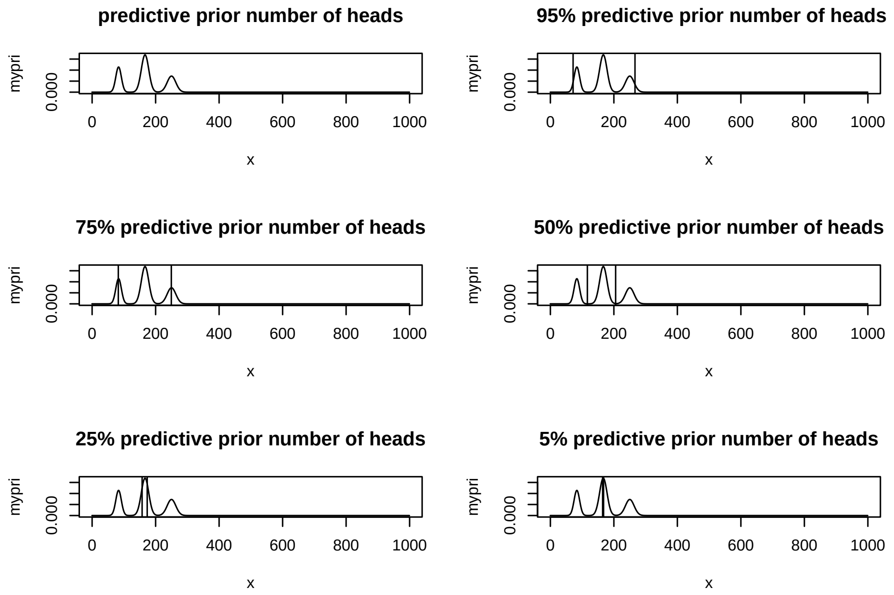
We have a mixture of 3 normal distributions, and show the central intervals for 5,25,50,75, and 95% predictive probabilities. The question gives θ as the probability of a 6 on a die, possibly unfair, in 1,000 tosses. we have $\theta = 1/12, 1/6, 1/4$ types of biased die. Using the normal approximation, the predictive prior probability is $\sum_{\theta} p(\theta)p(y|\theta)$ where the likelihood is approximated using a normal distribution $\mu = n * \theta_i, \sigma = n * \theta_i(1 - \theta_i)$

```
x<-seq(0,1000)-0.5 # continuity correction.
theta<-c(1/12,1/6,1/4)
n=1000
normLikelihood<-function(x,n,p){
  mean<-n*p
  varz<-n*p*(1-p)
  return(dnorm(x,mean=mean,sd=sqrt(varz)))

}
a<-dnorm(x,mean=n*theta[1],sd=sqrt(n*theta[1]*(1-theta[1]))  )
b<-dnorm(x,mean=n*theta[2],sd=sqrt(n*theta[2]*(1-theta[2]))  )
c<-dnorm(x,mean=n*theta[3],sd=sqrt(n*theta[3]*(1-theta[3]))  )



mypri<-a*0.25+b*0.5+c*0.25

sum(mypri) ## sums to 1 it is a distribution
```

```
## [1] 1
```

```
 par(mfrow=c(3,2))
plot(x,mypri,type='l',main='predictive prior number of heads')




data<-data.frame(x=x,p=mypri)

## highest probability interval
# 95%
```

```
plot(x,mypri,type='l',main=' 95% predictive prior number of heads')
abline(v=max(data[which(cumsum(data$p)<0.025),1])
)
abline(v=max(data[which(1-cumsum(data$p)>0.025),1]))

plot(x,mypri,type='l',main='75% predictive prior number of heads')
abline(v=max(data[which(cumsum(data$p)<0.125),1]))
abline(v=max(data[which(1-cumsum(data$p)>0.125),1])
)

plot(x,mypri,type='l',main='50% predictive prior number of heads')
abline(v=max(data[which(cumsum(data$p)<0.25),1]))
abline(v=max(data[which(1-cumsum(data$p)>0.25),1])
)

plot(x,mypri,type='l',main='25% predictive prior number of heads')
abline(v=max(data[which(cumsum(data$p)<0.375),1]))
abline(v=max(data[which(1-cumsum(data$p)>0.375),1])
)

plot(x,mypri,type='l',main='5% predictive prior number of heads')
abline(v=max(data[which(cumsum(data$p)<0.475),1]))
abline(v=max(data[which(1-cumsum(data$p)>0.475),1])
)
```



Question 8 (Normal distribution with unknown mean)

A random sample of n students is drawn from a large population, and weights are measured. The average height of the n sampled students is $\bar{y} = 150$ lbs. Assume the weights in the population are normally distribution with unknown mean, θ, and known standard deviation 20 lbs. Suppose the prior for $\theta \sim N(180, 40^2)$

For known variance, the limit of the posterior is $p(\theta, y) \approx N(\theta|\bar{y}, \sigma^2/n)$. And the direct formulation is $p(\theta|\bar{y}) = N(\theta|\mu_n, \tau_n^2)$. using equations 2.12.

For a posterior predictive interval, the marginal distribution for new data $p(\tilde{y}|y) \sim N(\mu_n, \sigma^2 + \tau_n^2)$

```
mu_n<-function(mu0,ybar,n,tau02,sigma2){
  mun<- (mu0/tau02 + (ybar*n)/sigma2)/(1/tau02 + n/sigma2)
  return(mun)
}
taun2<-function(tau02,n,sigma2){
  inv.taun2<- 1/tau02 +n/sigma2
  return(1/inv.taun2)
}

ybar=150
sigma2=20^2
mu0=180
  tau02=40^2
  n=10

## posterior interval n=10
  #lower<-round(qnorm(0.025,mu_n(mu0,ybar,tau02,10,sigma2),sd=sqrt(taun2(tau02,10,sigma2))),2)
    upper<-round(qnorm(c(0.025,0.975),mu_n(mu0,ybar,tau02,10,sigma2),sd=sqrt(taun2(tau02,10,sigma

message("posterior interval n=10: ",upper[1]," ",upper[2])
```

```
## posterior interval n=10: 138.49 162.98
```

```
## posterior predictive interval
  upper2<-round(qnorm(c(0.025,0.975),mu_n(mu0,ybar,tau02,10,sigma2),sd=sqrt(taun2(tau02,10,sigma

message("posterior predictive interval n=10: ",upper2[1]," ",upper2[2])
```

```
## posterior predictive interval n=10: 109.66 191.8
```

```
 lower<-round(qnorm(0.025,mu_n(mu0,ybar,tau02,100,sigma2),sd=sqrt(taun2(tau02,100,sigma
    upper<-round(qnorm(0.975,mu_n(mu0,ybar,tau02,100,sigma2),sd=sqrt(taun2(tau02,100,si

message("posterior interval n=100: ",lower," ",upper)
```

```
## posterior interval n=100: 146.16 153.99
```

```
 ## this approximately equals the limit
print("The asymptotic approximation")
```

```
## [1] "The asymptotic approximation"
```

```
 qnorm(c(0.025,0.975),mean=150, sigma2/100)
```

```
## [1] 142.1601 157.8399
```

```
## posterior predictive interval
 lower2<-round(qnorm(0.025,mu_n(mu0,ybar,tau02,100,sigma2),sd=sqrt(taun2(tau02,100,sign
    upper2<-round(qnorm(0.975,mu_n(mu0,ybar,tau02,100,sigma2),sd=sqrt(taun2(tau02,100,s

message("posterior predictive interval n=100: ",lower2," ",upper2)
```

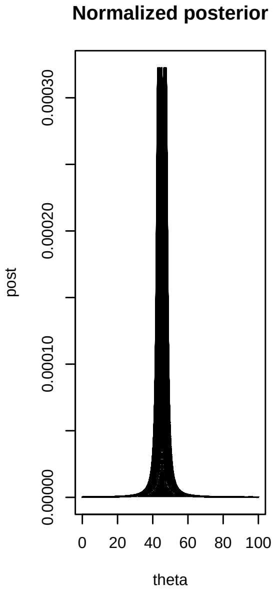
```
## posterior predictive interval n=100: 110.68 189.47
```

Question 10

suppose y1,...,y5 are iid Cauchy$(\theta, 1)$ r.vs. and the prior distribution for$\theta \sim U[0, 100]$. the given observations are y=43,44,45,46.5,47.5.

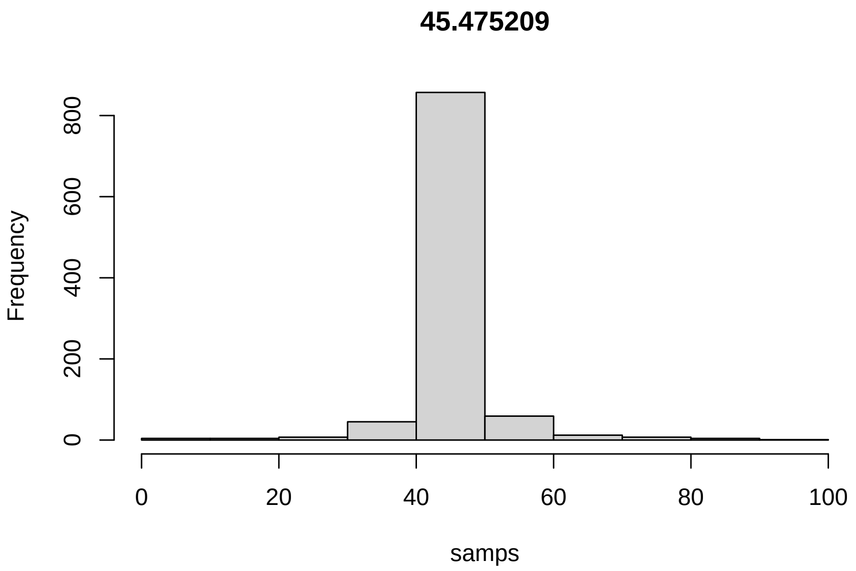
- (a) compute the unnormalized posterior density on a grid of points $\theta = 01/m, 2/m, ...100$. using the grid approximation, compute and plot the posterior density as a function of θ

```
 y= c(43,44,45,46.5,47.5)
 m=1000
theta<-seq(from=0,to=100000)/m
 ## p(theta | y) ~ p(y|theta)*p(theta)
 unnorm.post<-dcauchy(y,location=theta,scale=1)*dunif(theta,min=0,max=100) ## un norm p

 post<-unnorm.post/sum(unnorm.post)
 par(mfrow=c(1,3))
 plot(theta,post,type='l',main='Normalized posterior')
```



- (b) Sample 1000 draws from theta from posterior density and plot histogram we sample from theta [0,100] using the grid approximation, and the probability is from the posterior distribution.

```
samps<-(sample(theta,1000,prob=post,replace=T))
hist(samps,main=mean(samps))
```

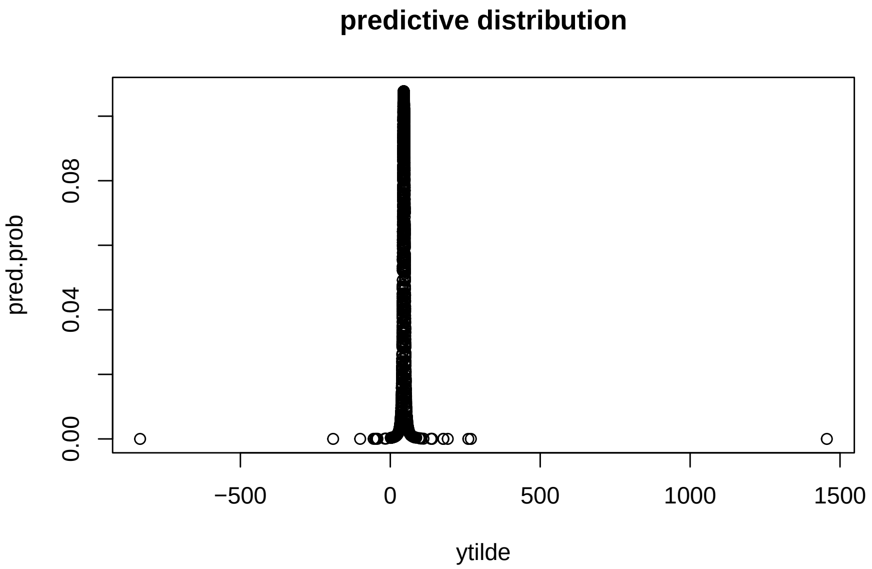


```
##sample mean is close to the mean of the observed.
```

- (c) Using the previous 1000 samples of θ to obtain 1000 samples from the predictive distribution of a future observation y_6, and plot the predictive draws. we use the sampled thetas from the posterior to sample from the Cauchy distribution. The predictive probability follows $p(x|y) = \int p(x|\theta)p(\theta|y)d\theta$ where $p(x|\theta)$ follows from the Cauchy distribution, and we have the posterior values for each theta (given the uniform grid of thetas). then for each predictive value, we sum the total probability across all thetas to compute the predictive probability. The maximum predictive value probability 44.745 with probability of 0.10.

```
## predictive distribution ? ??
 # p(x | y) = int p(x|theta)*p(theta|y ) dtheta
## the posterior is p(theta|y)
 # the likelihood p(x|theta)  ## we use the sampled thetas using the posterior
 ytilde<-rcauchy(1000,location=samps,scale=1)
 ## probability of the samples
 #prob_samp<-post[match(samps,theta)]
  summary(ytilde) ## we have a wide distribution of predictive values.
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -834.59   42.87   45.28   46.31   48.02 1456.07
## for all predictive values,  find the total probability
   ## int p(x|theta)*p(theta|y) d\theta  note that p(x|theta) is a function of theta, so we input
 pred.prob<-sapply(ytilde,function(x) sum(dcauchy(x,location=theta,scale=1)*post))

  plot(ytilde,pred.prob, main='predictive distribution')
```



```
  ## maximum predictive probability
  message('max pred. prob ', round(ytilde[which(pred.prob==max(pred.prob))],3))
```

```
## max pred. prob 44.754
```

Chapter 3

Parts

You can add parts to organize one or more book chapters together. Parts can be inserted at the top of an .Rmd file, before the first-level chapter heading in that same file.

Add a numbered part: `# (PART) Act one {-}` (followed by `# A chapter`)

Add an unnumbered part: `# (PART\*) Act one {-}` (followed by `# A chapter`)

Add an appendix as a special kind of un-numbered part: `# (APPENDIX) Other stuff {-}` (followed by `# A chapter`). Chapters in an appendix are prepended with letters instead of numbers.

Chapter 4

Footnotes and citations

4.1 Footnotes

Footnotes are put inside the square brackets after a caret `^[]`. Like this one [1].

4.2 Citations

Reference items in your bibliography file(s) using `@key`.

For example, we are using the **bookdown** package [Xie, 2022] (check out the last code chunk in index.Rmd to see how this citation key was added) in this sample book, which was built on top of R Markdown and **knitr** [Xie, 2015] (this citation was added manually in an external file book.bib). Note that the `.bib` files need to be listed in the index.Rmd with the YAML `bibliography` key.

The RStudio Visual Markdown Editor can also make it easier to insert citations: https://rstudio.github.io/visual-markdown-editing/#/citations

[1] This is a footnote.

Chapter 5

Blocks

5.1 Equations

Here is an equation.

$$f\left(k\right) = \binom{n}{k} p^k \left(1 - p\right)^{n-k} \tag{5.1}$$

You may refer to using `\@ref(eq:binom)`, like see Equation (5.1).

5.2 Theorems and proofs

Labeled theorems can be referenced in text using `\@ref(thm:tri)`, for example, check out this smart theorem 5.1.

Theorem 5.1. *For a right triangle, if c denotes the* length *of the hypotenuse and a and b denote the lengths of the* ***other*** *two sides, we have*

$$a^2 + b^2 = c^2$$

Read more here https://bookdown.org/yihui/bookdown/markdown-extensions-by-bookdown.html.

5.3 Callout blocks

The R Markdown Cookbook provides more help on how to use custom blocks to design your own callouts: https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html

Chapter 6

Sharing your book

6.1 Publishing

HTML books can be published online, see: https://bookdown.org/yihui/bookdown/publishing.html

6.2 404 pages

By default, users will be directed to a 404 page if they try to access a webpage that cannot be found. If you'd like to customize your 404 page instead of using the default, you may add either a `_404.Rmd` or `_404.md` file to your project root and use code and/or Markdown syntax.

6.3 Metadata for sharing

Bookdown HTML books will provide HTML metadata for social sharing on platforms like Twitter, Facebook, and LinkedIn, using information you provide in the `index.Rmd` YAML. To setup, set the `url` for your book and the path to your `cover-image` file. Your book's `title` and `description` are also used.

This `gitbook` uses the same social sharing data across all chapters in your book-all links shared will look the same.

Specify your book's source repository on GitHub using the `edit` key under the configuration options in the `_output.yml` file, which allows users to suggest an edit by linking to a chapter's source file.

Read more about the features of this output format here:

https://pkgs.rstudio.com/bookdown/reference/gitbook.html

Or use:

```
?bookdown::gitbook
```

Bibliography

Yihui Xie. *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition, 2015. URL http://yihui.org/knitr/. ISBN 978-1498716963.

Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*, 2022. URL https://CRAN.R-project.org/package=bookdown. R package version 0.26.