This report is automatically generated with the R package **knitr** (version `1.24`) .

```
---
title: "STAT 457 - FINAL"
author: "Martha Eichlersmith"
date: "2019-12-12"
output:
  pdf_document:
    fig_caption: yes
#    number_sections: true
header-includes:
- \usepackage{color}
- \usepackage{mathtools}
- \usepackage{bbm} #for mathbb for numbers
- \usepackage{amsbsy}
- \usepackage{caption} #to remove automatic table name and number - \captionsetup[table]{labelformat=empty}, put code under YAML
- \usepackage{booktabs}
- \usepackage{geometry}
- \usepackage{float} #to hold things in place
- \floatplacement{figure}{H}
- \usepackage{lastpage}
- \usepackage{fancyhdr}
- \pagestyle{fancy}
- \fancyhf{}
- \fancyhead[L]{STAT 457 Fall 2019 \\ Final}
- \fancyhead[R]{Martha Eichlersmith \\ Page \thepage\ of\ \pageref*{LastPage}}
- \setlength{\headheight}{22.5pt} #to remove \fancyhead error for head height
geometry: "left=0.75in,right=0.75in,top=1.1in,bottom=1in"
---

\captionsetup[table]{labelformat=empty}
```{r setup, echo=FALSE, results="hide", warning=FALSE, message=FALSE}
library(ggplot2) #ggplot
library(gridExtra) #organize plots
library(grid) #organize plots
library(latex2exp) #latex in ggplot titles
library(dplyr) #for piping
library(MASS)
library(invgamma)
knitr::opts_chunk$set(fig.width = 10, fig.height = 4)
knitr::opts_chunk$set(echo=FALSE)
decimal <- function(x, k) trimws(format(round(x, k), nsmall=k))
dec <- 5
```

\newpage
Problem 1
Problem 1a
For the data $Y = (125,18,20,34)$, implement the Gibbs sampler algorithm. Use a flat prior on θ.
Plot θ^i versus iteration i.
$Y = (y_1, y_2, y_3, y_4) \propto (2 + \theta, 1 - \theta, 1 - \theta, \theta)$
1. Draw a starting value, $t \sim$Uniform(0,1)
2. Draw a latent value, $Z \sim \text{Binomial}\left(y_1, \ \frac{ \theta}{2 + \theta} \right)$
3. Draw a parameter, $\theta \sim \text{Beta}(Z+y_4 + 1, y_2 + y_3 + 1)$

```{r p1.func_chain}
func_chain <-function(startseed, Y){
  set.seed(startseed)
  t <- runif(1)
  theta <- t
  chain<- c()#rep(NA, 10000)
  chain[1]<-theta
  Z.i<-rbinom(1,Y[1],(theta/(theta+2)))
  theta.i<-rbeta(1,Z.i+Y[4]+1,Y[2]+Y[3]+1)
  chain[2]<-theta.i
  Z.i<-rbinom(1,Y[1],(theta.i/(theta.i+2)))
  theta.i<-rbeta(1,Z.i+Y[1]+1,Y[2]+Y[3]+1)
  chain[3]<-theta.i
  k<-3
    while(abs(chain[k]-chain[k-1]) >= 0.000001) {
      Z.i<-rbinom(1,Y[1], (theta.i/(theta.i+2)))
      theta.i<-rbeta(1,Z.i+Y[4]+1,Y[2]+Y[3]+1)
      k<-k+1
      chain[k]<-theta.i
    }
  chain<-chain[!is.na(chain)]
  chain
}
```

```{r p1.scale_funcs}
func_scalelike<-function(x,y1,y2,y3,y4){
  like <-(2+x)^y1*(1-x)^(y2+y3)*(x)^y4
  like.max<-max(like)
  like/like.max #normalized likelihood (on scale from 0 to 1)
}

func_scalenormal<-function(x, mean, sd){
  scales::rescale(dnorm(x, mean, sd), to=c(0, 1)) #normal (on scale from 0 to 1)
}
```

```{r p1.func_p1AB}
func_problem1AB <- function(startseed, Y, number){
```
```

```
chain <- func_chain(startseed, Y)

df.chain <- data.frame(
        "theta.i" = chain,
        "i"=c(1:length(chain))
)

chain.mu <- mean(chain)
chain.sd <- sd(chain)
table <- data.frame(
  "Name" = c("Mean", "SD", "it", "Start"),
  "Value" = c(
    decimal(chain.mu, dec),
    decimal(chain.sd, dec),
    decimal(length(chain), 0),
    decimal(chain[1], dec)
  )
)
tg <- tableGrob(table)

print.Y <- paste(Y, collapse = ",")

#plot theta_i versus i (iterations)
plot.Gibbs <- ggplot(df.chain, aes(x=i, y=theta.i))+
  geom_line(alpha=0.4)+
  ggtitle("Gibbs Sampler")


colors <- c("navy", "maroon")
norm.like <-stat_function(
  fun = func_scalelike
  , args = list(y1=Y[1], y2=Y[2], y3=Y[3], y4=Y[4])
  , lwd = 1
  , linetype="solid"
  , aes(col="Normalized Like.")
  )
normal.approx <-stat_function(
  fun = func_scalenormal
  , args = list(mean=chain.mu, sd=chain.sd)
  , lwd = 1.5
  , linetype="dotted"
  , aes(col="Normal Approx."))

name <- paste("Normal Likelihood and Normal Approx")

#print normalized likelihoods
x <- seq(0, 1, 0.001)
df.x <- data.frame("X"=x)
plot.Like <-  ggplot(data=df.x, aes(x=X))+
  norm.like+normal.approx+
  ggtitle(paste(name))+
  theme(  axis.title.x = element_blank()
         ,axis.title.y = element_blank()
         ,axis.text.x = element_blank()
         ,axis.text.y = element_blank()
        )+
  scale_colour_manual("", values = c(colors[1], colors[2])) +
  theme(legend.position = c(.2,.9))

main <- paste("Chain", number, "for data for Y=(", print.Y, ")")

gs <- list(plot.Gibbs, tg, plot.Like)
grid.arrange(grobs=gs, nrow=1, widths=c(2, 1, 2),
             top = textGrob(main, vjust = .5, gp = gpar(fontface = "bold", cex = 1.2))
            )
}
```

```{r p1aRESULT, fig.height=2}
Y.A <- c(125, 18, 20, 34)

func_problem1AB(111, Y.A, 1)
func_problem1AB(112, Y.A, 2)
func_problem1AB(113, Y.A, 3)
func_problem1AB(114, Y.A, 4)
func_problem1AB(115, Y.A, 5)
```

```
\newpage
## Problem 1b
Repeat 1a for $Y = (14, 0, 1, 5)$.
```{r p1bRESULT, fig.height=2}
Y.B <- c(14,0,1,5)

func_problem1AB(121, Y.B, 1)
func_problem1AB(122, Y.B, 2)
func_problem1AB(123, Y.B, 3)
func_problem1AB(124, Y.B, 4)
func_problem1AB(125, Y.B, 5)
```
There is a lack of fit for the data in 1b, where the fit appears to be better for data in 1a. Convergence was assessed when values were had a difference

\newpage
Problem 1c
```{r p1.func_chaininfo}
func_chaininfo <- function(startseed, Y){
chain <- func_chain(startseed, Y)
```
```

```
it <- length(chain)
chain.mu <- mean(chain)
chain.sd <- sd(chain)
chain.se <- chain.sd/sqrt(it)
vec <-c(chain.mu, chain.sd, chain.se, it)
return(vec)
}
```

```{r p1cRESULT20chains}
Y.C <- c(125, 18, 20, 34)
print.Y.C <- paste(Y.C, collapse=",")
cnames <- c("Mean", "Standard Deviation", "Standard Error", "Iterations")
info.20chain <- mapply(func_chaininfo, startseed=c(1:20), Y=rep(list(Y.C), 20))
info.20chain <- t(info.20chain)
colnames(info.20chain) <- cnames

knitr::kable(info.20chain, align='rrrr', digits=dec, caption=paste("20 Chains for Y=(", print.Y.C, ")"))
```

```{r p1cRESULTavgsd.vs.se}
sd.of.avgs <- sd(info.20chain[,1])
avg.of.se <- mean(info.20chain[,3])

paste(round(sd.of.avgs, dec+1), "is the standard deviation of the 20 averages of theta")
paste(decimal(avg.of.se, dec+1), "is the average of the standard errors of the 20 chains of theta")
```

You would expected these values to be similar but it appears that our standard error average is under-estimating the variation in the chain means of $\t

\newpage
Problem 2

Problem 2a
For the genetic linkage model applied to $Y = (125, 18, 20, 34)$, implement the Metropolis algorithm.
(use a flat prior on $\theta)$. Use one long chain and plot θ^i versus i.
Try several driver functions:

```{r p2.func_pi,func_Metropolis,func_drivers}
func_pi <- function(theta, Y){
  (2+theta)^(Y[1]) * (1 - theta)^(Y[2]+Y[3]) * (theta)^(Y[4])
}

m <- 10

#theta.i = X (old value)
#theta.j = Y (new value)
theta.i <- 0.1
Y <- c(125, 18, 20, 34)

func_MetroFix <- function(startseed, m, Y, driver){
  #fixed driver
  set.seed(startseed)
  theta.i <- runif(1)
  chain <- c()
  alpha <- c()
  chain[1] <- theta.i
  for (k in 2:m){
    theta.j <- driver(1)
    alpha.ij <- min(c(1, func_pi(theta.j, Y)/func_pi(theta.i, Y)))
    if (theta.j >1 | theta.j <0){alpha.ij <- 0}
    u <- runif(1)
    if( u < alpha.ij){theta.i <- theta.j}
    chain[k] <- theta.i
  }
  chain
}

func_MetroDyn <- function(startseed, m, Y, driver){
  #dynamic driver
  set.seed(startseed)
  theta.i <- runif(1)
  chain <- c()
  alpha <- c()
  chain[1] <- theta.i
  for (k in 2:m){
    theta.j <- driver(1, theta.i)
    alpha.ij <- min(c(1, func_pi(theta.j, Y)/func_pi(theta.i, Y)))
    if (theta.j >1 | theta.j <0){alpha.ij <- 0}
    u <- runif(1)
    if( u < alpha.ij){theta.i <- theta.j}
    chain[k] <- theta.i
  }
  chain
}

func_driver1.Uniform <-    function(n){runif(n, min=0, max=1) }
func_driver2.Norm.sd.01 <- function(n, mu){rnorm(n, mean=mu, sd=0.01)}
func_driver3.Norm.sd.1 <-  function(n, mu){rnorm(n, mean=mu, sd=0.1)}
func_driver4.Norm.sd.5 <-  function(n, mu){rnorm(n, mean=mu, sd=0.5)}
func_driver5.Norm.mu.4sd.5 <-  function(n){rnorm(n, mean=0.4, sd=0.5)}
```

```{r p2.func_p2ABC}
func_problem2ABC <- function(chain, driver.name){
#main title
print.Y <- paste(Y, collapse=",")
```

```
main <- paste("Metropolis for data Y=(", print.Y, ") using the driver", driver.name)

df <- data.frame(
         "theta.i"=chain
        ,"i"=c(1:length(chain))
)


#table grob
chain.mu <- mean(chain)
chain.sd <- sd(chain)
table <- data.frame(
  "Name" = c("Mean", "SD", "it", "Start"),
  "Value" = c(
    decimal(chain.mu, dec),
    decimal(chain.sd, dec),
    decimal(length(chain), 0),
    decimal(chain[1], dec)
  )
)
tg <- tableGrob(table)

#plot theta_i versus i (iterations)
plot <- ggplot(df, aes(x=i, y=theta.i))+
  geom_line(alpha=0.4)

grid.arrange(plot, tg, widths=c(4,1)
             ,top = textGrob(main, vjust = .5, gp = gpar(fontface = "bold", cex = 1.1))
                            )
}
```

```{r p2aRESULT, fig.height=2}
Y.A <- c(125, 18, 20, 34)

func_problem2ABC(func_MetroFix(211, 10000, Y.A, func_driver1.Uniform), "Uniform(0,1)")
func_problem2ABC(func_MetroDyn(212, 10000, Y.A, func_driver2.Norm.sd.01),   "Normal(theta.i, 0.01)")
func_problem2ABC(func_MetroDyn(213, 10000, Y.A, func_driver3.Norm.sd.1 ),   "Normal(theta.i, 0.10)")
func_problem2ABC(func_MetroDyn(214, 10000, Y.A, func_driver4.Norm.sd.5 ),   "Normal(theta.i, 0.50)")
func_problem2ABC(func_MetroFix(215, 10000, Y.A, func_driver5.Norm.mu.4sd.5),"Normal(0.40, 0.10)")
```

\newpage
Problem 2b
Repeat 2a for $Y = (14, 0, 1, 5)$
```{r p2bRESULT, fig.height=2}
Y.B <- c(14, 0, 1, 5)

func_problem2ABC(func_MetroFix(221, 10000, Y.B, func_driver1.Uniform), "Uniform(0,1)")
func_problem2ABC(func_MetroDyn(222, 10000, Y.B, func_driver2.Norm.sd.01),   "Normal(theta.i, 0.01)")
func_problem2ABC(func_MetroDyn(223, 10000, Y.B, func_driver3.Norm.sd.1 ),   "Normal(theta.i, 0.10)")
func_problem2ABC(func_MetroDyn(224, 10000, Y.B, func_driver4.Norm.sd.5 ),   "Normal(theta.i, 0.50)")
func_problem2ABC(func_MetroFix(225, 10000, Y.B, func_driver5.Norm.mu.4sd.5),"Normal(0.40, 0.10)")
```

Problem 2c
Compute both the posterior mean and standard deviation for both data sets.
Compare to results from the previous problem.

In problem 1 the means were similar, but in problem 2 the means vary depending on the driver. Some of the means in problem 2 are close to the means in

\newpage
Problem 2d

```{r p2.func_Metroinfo}
func_Metroinfo <- function(startseed, m, Y, driver, metro.func){
chain <- metro.func(startseed, m, Y, driver)
it <- length(chain)
chain.mu <- mean(chain)
chain.sd <- sd(chain)
chain.se <- chain.sd/sqrt(it)
vec <-c(chain.mu, chain.sd, chain.se)
return(vec)
}
```

```{r p2.func_20chains}
func_20metrochains <- function(Y, driver, metro.func){
print.Y <- paste(Y, collapse=",")
cnames <- c("Mean", "SD", "SE")
info.20chain <- mapply( func_Metroinfo
                       , startseed=c(1:20)
                       , m =rep(10000, 20)
                       , Y=rep(list(Y), 20)
                       , driver=rep(list(driver), 20)
                       , metro.func=rep(list(metro.func), 20)
                      )
info.20chain <- t(info.20chain)
colnames(info.20chain) <- cnames
info.20chain
}
```

```{r p2dRESULT20chains}
Y.D <- c(125, 18, 20, 34)

d1 <- func_20metrochains(Y.D, func_driver1.Uniform,      func_MetroFix)
d2 <- func_20metrochains(Y.D, func_driver2.Norm.sd.01,    func_MetroDyn)
```

```
d3 <- func_20metrochains(Y.D, func_driver3.Norm.sd.1,     func_MetroDyn)
d4 <- func_20metrochains(Y.D, func_driver4.Norm.sd.5,     func_MetroDyn)
d5 <- func_20metrochains(Y.D, func_driver5.Norm.mu.4sd.5, func_MetroFix)

table2d <- cbind(d1, d2, d3, d4, d5)

print.Y.D <- paste(Y.D, collapse=",")

knitr::kable(table2d, digits=4, booktabs=TRUE, 'latex'
             , caption=paste("20 Chains for Y=(",print.Y.D, ") with Different Drivers")
             ) %>%
  kableExtra::kable_styling(latex_options=c("hold_position", "scale_down") ) %>%
  kableExtra::add_header_above(c( "Uniform(0, 1)" =3
                                ,"Normal(theta.i, 0.01)"=3
                                ,"Normal(theta.i, 0.10)"=3
                                ,"Normal(theta.i, 0.50"=3
                                ,"Normal(0.40, 0.10)"=3
                               ))
```

```{r p2dRESULTavgsd.vs.se}
sd.of.avgs <- c(
   sd(table2d[,1])
  ,sd(table2d[,4])
  ,sd(table2d[,7])
  ,sd(table2d[,10])
  ,sd(table2d[,13])
  )

avg.of.se <- c(
   mean(table2d[,3])
  ,mean(table2d[,6])
  ,mean(table2d[,9])
  ,mean(table2d[,12])
  ,mean(table2d[,15])
  )

driver.name <- c(
  "Uniform(0,1)"
  ,"Normal(theta.i, 0.01)"
  ,"Normal(theta.i, 0.10)"
  ,"Normal(theta.i, 0.50)"
  ,"Normal(0.40, 0.10)"
)


table2d.sdse <- cbind(sd.of.avgs, avg.of.se)
rownames(table2d.sdse) <- driver.name

knitr::kable(table2d.sdse, 'latex', booktabs=TRUE
             , caption=paste("SD of Average theta's versus Average SE")
             )  %>%
    kableExtra::kable_styling(latex_options="hold_position")
```

```
Similar to problem 1, it appears that our estimation of the variation in the mean of theta is lower than the actual variation between the means.


\newpage
# Problem 3

## Problem 3a
Consider the 1-way variance components model
$$Y_{ij} = \theta_i + \epsilon_{ij}$$
where $Y_{ij}$ is the $j$th observation from the $i$th group, $\theta_i$ is the effect, $\epsilon_{ij}$=error,
$i=1, ..., K$ and $j=1, ..., J$.
It is assume that $\epsilon_{ij} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2_{\epsilon})$
and $\theta_i \stackrel{\text{iid}}{\sim} \mathcal{N}(\mu, \sigma^2_{\theta})$.
Under the prior specification $p(\sigma^2_{\epsilon}, \sigma^2_{\theta}, \mu) =$
$p(\sigma^2_{\epsilon})p(\sigma^2_{\theta})p(\mu)$, with
$p(\sigma_\theta^2) = \text{InverseGamma}(a_1, b_1)$,
$p(\sigma_\epsilon^2) = \text{InverseGamma}(a_2, b_2)$, and
$p(\mu) = \mathcal{N}(\mu_0, \sigma^2_0)$.
Let $\overline{Y}_i = \frac{1}{J} \sum_{j=1}^J Y_{ij}$ and $\theta = (theta_1, \cdots, \theta_k)$.
Show the following:

$$
\begin{aligned}
p(\mu, \sigma^2_\epsilon, \sigma^2_\theta) & = p(\mu) p(\sigma^2_\theta) p(\sigma^2_\epsilon)
\\
& = \mathcal{N}(\mu_0, \sigma^2_0) \text{IG}(a_1, b_1) \text{IG}(a_2, b_2)
\\
&= \left[
\frac{1}{\sqrt{2 \pi} \  \sigma_0} \exp \left\{ - \frac{1}{2} \frac{ (\mu - \mu_0)^2}{\sigma_0^2} \right\}
\right]
\cdot
\left[
(\sigma^2_\theta)^{a_1 - 1} \exp \left\{ -b_1 \sigma^2_\theta \right\}
\right]^{-1}
\cdot
\left[
(\sigma^2_\epsilon)^{a_2 - 1} \exp \left\{ -b_2 \sigma^2_\epsilon \right\}
\right]^{-1}
\\[0.5ex]
& \propto
\sigma_\theta^{-2(a_1 - 1)} \cdot
\sigma_\epsilon^{-2(a_2 - 1)} \cdot
\exp \left\{
```

```
-\frac{1}{2} \frac{(\mu - \mu_0)^2}{\sigma_0^2} + b_1 \sigma^2_\theta + b_2 \sigma^2_\epsilon
\right\}
\\[2ex]
p(\mu, \theta, \sigma^2_\theta, \sigma^2_\epsilon \mid Y) & \propto p(Y \mid \mu, \theta, \sigma^2_\theta, \sigma^2_\epsilon ) \cdot p(\theta \mid \mu,
\\
& = \left[ \prod_{i=1}^K \prod_{j=1}^J \left( p(y_{ij}) \sim \mathcal{N}(\theta_i, \sigma^2_\epsilon) \right) \right]  \cdot
\left[ \prod_{i=1}^K \left( p(\theta_i) \sim \mathcal{N}(\mu, \sigma^2_\theta) \right) \right]
 \cdot \prod_{i=1}^K p( \mu, \sigma^2_\theta, \sigma^2_\epsilon)
\\[0.5ex]
& = \left[
\sigma_\epsilon^{-(KJ)} \cdot \exp \left\{ - \frac{1}{2} \frac{\sum_{i=1}^K \sum_{j=1}^J (y_{ij}  - \theta_i)^2}
{\sigma^2_\epsilon} \right\} \right]
\\[0.5ex]
& \quad \cdot \left[\sigma_\theta^{-K} \exp \left\{ - \frac{1}{2} \frac{\sum_{i=1}^K (\theta_i - \mu)^2}
{\sigma^2_\theta } \right\} \right]
\\[0.5ex]
& \quad \cdot \left[
\sigma_\theta^{-2K(a_1 - 1)} \cdot
\sigma_\epsilon^{-2K(a_2 - 1)} \cdot
\exp \left\{ K \left(
-\frac{1}{2} \frac{(\mu - \mu_0)^2}{\sigma_0^2} + b_1 \sigma^2_\theta + b_2 \sigma^2_\epsilon
\right) \right\} \right]
\end{aligned}
$$

\newpage
### Problem 3a(1)
$$
\begin{aligned}
p(\mu \mid \theta, \sigma^2_{\epsilon}, \sigma^2_{\theta}, Y)
&\propto  =  \exp \left\{ - \frac{1}{2} \frac{\sum_{i=1}^K (\theta_i - \mu)^2}
{\sigma^2_\theta } \right\}  \cdot
\exp \left\{ K \left(
-\frac{1}{2} \frac{(\mu - \mu_0)^2}{\sigma_0^2}
\right) \right\}
\\[0.5ex]
& =  \mathcal{N} \left(
\frac{ \sigma_\theta^2 \mu_0 + \sigma_0^2 \sum \theta_i}{\sigma^2_\theta + K \sigma^2_0}
, \
\frac{ \sigma^2_\theta \sigma^2_0}{\sigma^2_\theta + K \sigma^2_0}
\right)
\end{aligned}
$$

### Problem 3a(2)
$$
\begin{aligned}
p(\theta_i \mid \mu, \sigma^2_\epsilon, \sigma^2_\theta, Y) & \propto
\cdot \exp \left\{ - \frac{1}{2} \frac{\sum_{i=1}^K \sum_{j=1}^J (y_{ij}  - \theta_i)^2}
{\sigma^2_\epsilon} \right\}  \cdot
\exp \left\{ - \frac{1}{2} \frac{\sum_{i=1}^K (\theta_i - \mu)^2}
{\sigma^2_\theta } \right\}
\\[0.5ex]
& =  \mathcal{N} \left(
\frac{J\sigma^2_\theta}{J\sigma^2_\theta + \sigma_\epsilon^2} \cdot \overline{Y}_i
+
\frac{\sigma^2_\epsilon}{J\sigma^2_\theta + \sigma_\epsilon^2} \cdot \mu
, \ \ \
\frac{\sigma^2_\theta \sigma^2_\epsilon}{J\sigma^2_\theta + \sigma_\epsilon^2}
\right)
\end{aligned}
$$

### Problem 3a(3)
$$
\begin{aligned}
p(\sigma^2_\epsilon \mid \mu, \theta, \sigma^2_\theta, Y)   & \propto
 =
\sigma_\epsilon^{-(KJ)} \cdot \exp \left\{ - \frac{1}{2} \frac{\sum_{i=1}^K \sum_{j=1}^J (y_{ij}  - \theta_i)^2}
{\sigma^2_\epsilon} \right\}
 \cdot
\sigma_\epsilon^{-2K(a_2 - 1)} \cdot
\exp \left\{ K \left(
 b_2 \sigma^2_\epsilon
\right) \right\}
\\[0.5ex]
& =\text{IG}
\left(
a_2 + \frac{KJ}{2}, \
b_2 + \frac{1}{2} \sum_{i=1}^K \sum_{j=1}^J (Y_{ij} - \theta_i)^2
\right)
\end{aligned}
$$

### Problem 3a(4)
$$
\begin{aligned}
p(\sigma^2_\theta \mid \mu, \theta, \sigma^2_\epsilon, Y)  & \propto
\sigma_\theta^{-K} \exp \left\{ - \frac{1}{2} \frac{\sum_{i=1}^K (\theta_i - \mu)^2}
{\sigma^2_\theta } \right\}
 \cdot
\sigma_\theta^{-2K(a_1 - 1)} \cdot
\exp \left\{ K \left(
 b_1 \sigma^2_\theta
\right) \right\}
\\[0.5ex]
& = \text{IG}
\left(
```

```
a_1 + \frac{K}{2}
, \
b_1 + \frac{1}{2} \sum_{i=1}^K  (\theta_i - \mu)^2
\right)
\end{aligned}
$$

\newpage
## Problem 3b
Run the Gibbs sampler for the data below.
```{r p3b.data, eval=TRUE}
j1 <- c( 7.298,  3.846,  2.434,  9.566,  7.990)
j2 <- c( 5.220,  6.556,  0.608, 11.788, -0.892)
j3 <- c( 0.110, 10.386, 13.434,  5.510,  8.166)
j4 <- c( 2.212,  4.852,  7.092,  9.288,  4.980)
j5 <- c( 0.282,  9.014,  4.458,  9.446,  7.198)
j6 <- c( 1.722,  4.782,  8.106,  0.758,  3.758)

Y <- cbind(j1, j2, j3, j4, j5, j6)
y_j. <- apply(Y, 2, mean)

rbind(Y, c(rep(" ", 6)), y_j.)
paste("y_..=", mean(Y))

```

Use one chain of length 75,000.
Take $p(\mu) =\mathcal{N}(0, 10^{12})$, $p(\sigma^2_\epsilon) = IG(0, 0)$, and $p(\sigma^2_\theta) = IG(1, 1)$.
For each θ_i, for σ_ϵ, and for θ_θ, plot the simulated value at iteration j versus j.
Summarize each posterior marginal.

```{r p3b.func_gibbsdf}
func_gibbsdf <- function(startseed, it, mu.o, sig2.o, a1, b1, a2, b2, Y){
set.seed(startseed)
J <- dim(Y)[2] #ROW, observations
K <- dim(Y)[1] #COLUMN, group
Y.col <- apply(Y, 2, mean) #col means
sig.o <- sqrt(sig2.o)
chain.mu <- c()
chain.sig2.0 <- c()
chain.sig2.e <- c()
chain.theta <- matrix(0, nrow=it, ncol=K)

#Iteration ONE - STARTING VALUES
mu <- rnorm(1, mu.o, sig.o) #generate \mu
sig2.0 <- rinvgamma(1, a1, b1) #generate \sigma_\theta
sig2.e <- rinvgamma(1, a2, b2) #generate \sigma_\epsilon
theta <- rnorm(K, mu, sqrt(sig2.0)) #generate \theta = (\theta_1, ..., theta_K)
chain.mu[1] <- mu
chain.sig2.0[1] <- sig2.0
chain.sig2.e[1] <- sig2.e
chain.theta[1,] <- theta

#ITERATION TWO+
for (l in 2:it){
  #values for respective distributions
  mean.mu <- (sig2.0 * mu.o + sig2.o *sum(theta)) / (sig2.0 + K * sig2.o)
  sig2.mu <- (sig2.0*sig2.o) / (sig2.0 + K * sig2.o)
  A.sig2.0 <- a1 + K/2
  B.sig2.0 <- b1 + .5 * sum((theta - mu)^2)
  A.sig2.e <- a2 + (K*J)/2
  B.sig2.e <- b2 + .5 *sum( (Y - matrix(theta, K, J, byrow=TRUE))^2 )
  mean.theta <- Y.col * (J * sig2.0 )/(J * sig2.0 + sig2.e) + mu * (sig2.e)/(J * sig2.0 + sig2.e)
  sig2.theta <- (sig2.0 * sig2.e)/(J * sig2.0 + sig2.e)

  #new values
  mu <- rnorm(1, mean.mu, sqrt(sig2.mu))
  sig2.0 <- rinvgamma(1, A.sig2.0, B.sig2.0)
  sig2.e <- rinvgamma(1, A.sig2.e, B.sig2.e)
  theta <- rnorm(K, mean.theta, sqrt(sig2.theta))

  #put new values into chains
  chain.mu[l] <- mu
  chain.sig2.0[l] <- sig2.0
  chain.sig2.e[l] <- sig2.e
  chain.theta[l,] <- theta
}

df.chain <- data.frame(
   "iteration"= c(1:it)
  ,"mu" = chain.mu
  ,"sig2.theta" = chain.sig2.0
  ,"sig2.epsilon" = chain.sig2.e
  ,"theta.1" = chain.theta[,1]
  ,"theta.2" = chain.theta[,2]
  ,"theta.3" = chain.theta[,3]
  ,"theta.4" = chain.theta[,4]
  ,"theta.5" = chain.theta[,5]
  ,"theta.6" = chain.theta[,5]
)

return(df.chain)
}

```

```{r 3b.func_plotchain}
func_plotchain <- function(df.chain,throw.out,  c, yval.tex){
```
```

```
df.chain <- df.chain[throw.out:nrow(df.chain), ] #throw out first _ values
yval <- df.chain[, c]
post.mean <- mean(yval)
xname <- colnames(df.chain)[1]
ggplot(df.chain, aes(x=iteration, y=df.chain[,c])) + geom_line(alpha=0.4)+
  ggtitle(TeX(paste(yval.tex ,"versus iteration")))+ ylab(TeX(yval.tex)) + xlab("Iteration")+
  annotate('text', x=mean(df.chain$iteration), y=Inf,
           label=paste("Mean=",decimal(post.mean, dec)), vjust=2, fontface="bold")
}
```

```{r p3b.func_problem3b}

func_problem3b <- function(startseed, it, mu.o, sig2.o, a1, b1, a2, b2, Y, throw.out){
df.chain <- func_gibbsdf(startseed, it, mu.o, sig2.o, a1, b1, a2, b2, Y)
plot.mu <-          func_plotchain(df.chain, throw.out, 2, "$\\mu$")
plot.sig2theta <-   func_plotchain(df.chain, throw.out, 3, "$\\sigma^2_{\\theta}$")
plot.sig2epsilon <- func_plotchain(df.chain, throw.out, 4, "$\\sigma^2_{\\epsilon}$")
plot.theta1 <-      func_plotchain(df.chain, throw.out, 5, "$\\theta_1$")
plot.theta2 <-      func_plotchain(df.chain, throw.out, 6, "$\\theta_2$")
plot.theta3 <-      func_plotchain(df.chain, throw.out, 7, "$\\theta_3$")
plot.theta4 <-      func_plotchain(df.chain, throw.out, 8, "$\\theta_4$")
plot.theta5 <-      func_plotchain(df.chain, throw.out, 9, "$\\theta_5$")
plot.theta6 <-      func_plotchain(df.chain, throw.out, 10, "$\\theta_6$")

gs <- list(  plot.mu
            , plot.sig2theta
            , plot.sig2epsilon
            , plot.theta1
            , plot.theta2
            , plot.theta3
            , plot.theta4
            , plot.theta5
            , plot.theta6)

#gs <- list(plot.theta1, plot.theta2, plot.theta3)
grid.arrange(grobs=gs, nrow=3, ncol=3
            , top = textGrob(paste("Chains versus Iterations, throwing out first", throw.out,"values on", it, "chain"), vjust = .5, gp = gpar(fontface
           )
}
```

Cannot use $a_1, b_1 = 0$, otherwise $\sigma^2_\theta = \infty$, resulting in N/A so will use 0.01 instead.
```{r p3bRESULT, fig.height=8}
it <-75000
#prior for mu
mu.o <- 0
sig2.o <- 10^12
#prior for sig2.theta
a1 <- .01
b1 <- .01
#prior for sig2.epsilon
a2 <- 1
b2 <- 1

Y <- cbind(j1, j2, j3, j4, j5, j6)
startseed <- 32

func_problem3b(startseed, it, mu.o, sig2.o, a1, b1, a2, b2, Y, 1000)
```

\newpage
Problem 3c
Repeat 3b using the prior specification $p(\mu) = \mathcal{N}(0, 10^{12})$, $p(\sigma_\epsilon^2) = IG(0, 0)$, and
$p(\sigma^2_\theta) = IG(0, 0)$. Does this specification violate the Hobart-Casella conditions?
Describe what happens to the Gibbs sampler chain in this case.

This violates the Hobart-Casella conditions. If left at zero, the result will "blow up" (i.e., the chains are full of NA).
Cannot use $a_1, b_1, a_2, b_2 = 0$, otherwise $\sigma^2_\theta = \sigma^2_\epsilon = \infty$, resulting in N/A so will use 0.01 instead.
```{r p3cRESULT, fig.height=8}
it <-75000
#prior for mu
mu.o <- 0
sig2.o <- 10^12
#prior for sig2.theta
a1 <- .01
b1 <- .01
#prior for sig2.epsilon
a2 <- .01
b2 <- .01

Y <- cbind(j1, j2, j3, j4, j5, j6)
startseed <- 32

func_problem3b(startseed, it, mu.o, sig2.o, a1, b1, a2, b2, Y, 1000)
```

\newpage
Problem 5
Suppose that X and Y have exponential conditional distributions restricted over the interval $(0, B)$, i.e.
$p(x \mid y) \propto y \exp \left\{ -yx \right\}$ for $0 < x < B < \infty$ and
$p(y \mid x) \propto x \exp \left\{ -xy \right\}$ for $0 < y < B < \infty$, where B is known constant.

Problem 5a
Take $m=1$ and $B=3$. Run the data augmentation algorithm using these conditionals.
(Hist: Reject the exponential deviates that lie outside $(0, B)$).
Obtain the marginal for x using the mixture of conditionals $p(x \mid y)$, mixed over the simulated y deviates in your chain.
```
```

```
```{r p5.func_gibbs.prob5A}
func_gibbs.p5 <- function(B, it){
  x <- c(rep(B+1, it))
  y <- c(rep(B+1, it))
  x[1] <- runif(1, 0, B)
  y[1] <- runif(1, 0, B)
  for(k in 2:it) {
    while(x[k] > B){ x[k]<-rexp(1,y[k-1]) }
    while(y[k] > B){ y[k]<-rexp(1,x[k-1]) }
  }
  df <- as.data.frame(cbind(x, y))
return(df)
}

marginal <- function(k, rate){
  (1-exp(-rate*k))/(k)
}

func_marginal <- function(k, rate){
  marginal(k, rate)/rate #to normalize to 0 to 1
}

func_problem5A <- function(startseed, B, it){
  set.seed(startseed)
  df <- func_gibbs.p5(B, it)

true.cdf <-stat_function(fun = func_marginal, args = B, lwd = 1, linetype="solid", col="maroon")

plot.x <- ggplot(df, aes(x)) + xlim(0, B)+
  geom_histogram(aes(y=..density..), alpha=0.4) +
  true.cdf +xlab("X Chain Values")+
  ggtitle("Marginal of X: Histogram and True Curve")

plot.y <- ggplot(df, aes(y)) + xlim(0, B)+
  geom_histogram(aes(y=..density..), alpha=0.4) +
  true.cdf +xlab("Y Chain Values")+
  ggtitle("Marginal of Y: Histogram and True Curve")

grid.arrange(plot.x, plot.y, nrow=1)

}
```


```{r p5aRESULTS, message=FALSE, warning=FALSE}
func_problem5A(510, 3, 100000)
```


Problem 5b
Show that the marginal for x is proportional to $(1 - \exp\left\{ -Bx \right\})/x$.
Compare your results in 5a to this curve.

$$
\begin{aligned}
p_{X \mid Y}(x \mid y) & = ye^{-yx}
\\
p_{X}(x) & \propto \int_{0}^B e^{-yx} dy
\\[0.5ex]
& = \frac{1}{x} \int_{0}^B xe^{-xy} dy
\\[0.5ex]
& = \frac{1}{x} \Big[- e^{-xy}
\Big]_{y=0}^{y=B}
\\[0.5ex]
& = \frac{1}{x} \Big(
\left[-e^{-Bx} \right] - \left[-e^{-0 \cdot x} \right]
\Big)
\\[0.5ex]
& = \frac{1}{x} \Big(- e^{-Bx} + 1 \Big)
\\[0.5ex]
& = \frac{ 1 - e^{-Bx} }{x}
\\[1.5ex]
p_{X}(x) & \propto \frac{ 1 - e^{-Bx} }{x}
\end{aligned}
$$

See above for the histogram of the marginals with the curves. Note the curves are normalized by dividing by B so that both the histogram density and

\newpage

Problem 5c
Repeat 5a and 5b using $B= \infty$. Describe what happens. Is the marginal for x a proper density in this case?

$$
\lim_{B \to \infty} \frac{ 1 - e^{-Bx} }{x} = \frac{ 1 - \lim_{B \to \infty} e^{-Bx}}{x} = \frac{1}{x}
$$

This is not a proper marginal.
$$
\int_{0}^B \frac{1}{x} dx = \ln(x) \Big|_{0}^B = \ln(B) - \ln(0) = \ln(B) + \infty = \infty
$$


```{r p5.func_prob5B}

func_margInfty <- function(k, B){
```
```

````
  scales::rescale(marginal(k, B), to=c(0, 1/B))
}

func_problem5C <- function(startseed, B, it){
  set.seed(startseed)
  df <- func_gibbs.p5(B, it)

true.cdf <-stat_function(fun = func_margInfty, args = B, lwd = 1, linetype="solid", col="maroon")

plot.x <- ggplot(df, aes(x)) +
  xlim(0, B)+
  ylim(0, 1/B)+
  geom_histogram(aes(y=..density..), alpha=0.4) +
  true.cdf +
  xlab("X Chain Values")+
  ggtitle("Marginal of X: Histogram and True Curve")

plot.y <- ggplot(df, aes(y)) +
  xlim(0, B)+
  ylim(0, 1/B)+
  geom_histogram(aes(y=..density..), alpha=0.4) +
  true.cdf +
  xlab("Y Chain Values")+
  ggtitle("Marginal of Y: Histogram and True Curve")

grid.arrange(plot.x, plot.y, nrow=1)

}
```


```{r p5RESULTS, message=FALSE, warning=FALSE}
func_problem5C(530, 1e08, 100000)
```

```{r PRINTCODE}
#PRINTING THE CODE
#knitr::stitch("HW06.Rmd") to go to latex
#knitr::stitch(   script="STAT457-FINAL.Rmd"   , system.file("misc", "knitr-template.Rhtml", package="knitr")) #code to HTML
```
````

````
## Error: <text>:10:3: unexpected input
## 9: header-includes:
## 10: - \
##       ^
````

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 17763)
## 
## Matrix products: default
## 
## locale:
## [1] LC_COLLATE=English_United States.1252  LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252 LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
## 
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods   base
## 
## other attached packages:
##  [1] invgamma_1.1        MCMCpack_1.4-4      MASS_7.3-51.4       coda_0.19-3
##  [5] gtable_0.3.0        numDeriv_2016.8-1.1 matlib_0.9.2        readr_1.3.1
##  [9] dplyr_0.8.3         latex2exp_0.4.0     gridExtra_2.3       ggplot2_3.2.1
## [13] knitr_1.24
## 
## loaded via a namespace (and not attached):
##  [1] httr_1.4.1              jsonlite_1.6            viridisLite_0.3.0
##  [4] carData_3.0-2           shiny_1.3.2             assertthat_0.2.1
##  [7] highr_0.8               cellranger_1.1.0        yaml_2.2.0
## [10] pillar_1.4.2            backports_1.1.4         lattice_0.20-38
## [13] quantreg_5.51           glue_1.3.1              digest_0.6.20
## [16] manipulateWidget_0.10.0 promises_1.0.1          rvest_0.3.4
## [19] colorspace_1.4-1        htmltools_0.3.6         httpuv_1.5.1
## [22] Matrix_1.2-17           pkgconfig_2.0.2         SparseM_1.77
## [25] haven_2.1.1             purrr_0.3.2             xtable_1.8-4
## [28] scales_1.0.0            webshot_0.5.1           openxlsx_4.1.0.1
## [31] later_0.8.0             rio_0.5.16              MatrixModels_0.4-1
## [34] tibble_2.1.3            car_3.0-3               withr_2.1.2
## [37] lazyeval_0.2.2          magrittr_1.5            crayon_1.3.4
## [40] readxl_1.3.1            mime_0.7                mcmc_0.9-6
## [43] evaluate_0.14           forcats_0.4.0           xml2_1.2.2
## [46] foreign_0.8-71          tools_3.6.1             data.table_1.12.4
## [49] hms_0.5.0               stringr_1.4.0           munsell_0.5.0
## [52] zip_2.0.4               kableExtra_1.1.0        compiler_3.6.1
## [55] rlang_0.4.0             rstudioapi_0.10         htmlwidgets_1.5.1
## [58] crosstalk_1.0.0         miniUI_0.1.1.1          labeling_0.3
## [61] rmarkdown_1.14          abind_1.4-5             curl_4.2
## [64] R6_2.4.0                zeallot_0.1.0           stringi_1.4.3
```
```

```
## [67] Rcpp_1.0.2             vctrs_0.2.0            rgl_0.100.30
## [70] tidyselect_0.2.5       xfun_0.8
```

```r
Sys.time()
```

```
## [1] "2019-12-11 12:08:17 CST"
```