````
---
title: STAT 457 Homework 04
author: Martha Eichlersmith
date: 2019-11-05
output:
  pdf_document:
    fig_caption: yes
header-includes:
  - \usepackage{color}
  - \usepackage{mathtools}
  - \usepackage{amsbsy} #bold in mathmode
  - \usepackage{nicefrac} # for nice fracs
---
```{r, echo=FALSE, results="hide", warning=FALSE, message=FALSE}
library(ggplot2) #ggplot
library(readr) #import CSV
library(gridExtra) #organize plots
library(grid) #organize plots
library(latex2exp) #latex in ggplot titles
library(knitr) #to help make tables
library(matlib) #A = matrix, inv(A) = A^{-1}
library(numDeriv) #calculate numerical first and second order
derivatives
decimal <- function(x, k) trimws(format(round(x, k), nsmall=k))
dec <- 5
```

## Problem 1
Let $y_1, \cdots, y_n$ be an iid sample from the Poisson
distribution with parameter $\lambda$.  Derive Jeffery's
(noninformative) prior.  This prior corresponds to the gamma
distribution with which parameters?

$$
\begin{aligned}
L(\lambda \mid Y) & = \frac{ \lambda^y e^{-\lambda}}{y!}
\propto \lambda^y e^{-\lambda}
\\[.5ex]
\ell(\lambda \mid Y) & = y \log \lambda - \lambda
\\[.5ex]
\frac{\partial \ell(\lambda \mid Y)}{\partial \lambda }
& = \frac{y}{\lambda} - 1
\\[.5ex]
\frac{\partial^2 \ell(\lambda \mid Y)}{\partial \lambda^2}
& = - \frac{y}{\lambda^2}
````

```
\right]
$$
and $|X|$ is the determinant of the matrix $X$.

Let $y_1, \cdots, y_n$ be an iid sample from the $\mathcal{N}
(\mu, \sigma^2)$ distribution, where $\mu$ and $\sigma$ are both
unknown.  Derive the invariant prior.  How does it compare with
the prior $p(\theta, \sigma^2) \propto \nicefrac{1}{\sigma^2}$?

$$
\begin{aligned}
L(\mu, \sigma^2 \mid \pmb{Y}) & = \prod_{i=1}^n \frac{ 1}{\sqrt{
2 \pi \sigma^2}} \exp \left\{ \frac{ (y_i - \mu)^2}{2 \sigma^2}
\right\}
\\[.5ex]
\ell (\mu, \sigma^2 \mid \pmb{Y}) & = \log \left( \prod_{i=1}^n
\frac{ 1}{\sqrt{ 2 \pi \sigma^2}} \exp \left\{ \frac{ (y_i -
\mu)^2}{2 \sigma^2}  \right\} \right)
\\[.5ex]
& = - \frac{n}{2} \log (2 \pi) - \frac{n}{2} \log (\sigma^2) -
\frac{ \sum_i (y_i - \mu)^2}{2 \sigma^2}
\\[.5ex]
\frac{\partial^2 \ell}{\partial \mu^2} & = - \frac{n}{\sigma^2}
\\[.5ex]
\frac{\partial^2 \ell}{\partial \mu \partial \sigma^2} & = -
\frac{ \sum_i (y_i - \mu)^2}{\sigma^4} = \frac{ \partial^2 \ell}
{\partial \sigma^2 \partial \mu}
\\[.5ex]
\frac{ \partial^2 \ell}{\partial (\sigma^2)^2 } & = \frac{n}
{\sigma^2} = \frac{n}{2 \sigma^4} - \frac{ \sum_i (y_i - \mu)^2}
{\sigma^6}
\\[1ex]
p(\mu, \sigma^2) & \propto | J(\mu, \sigma^2) |^{\frac{1}{2}}
\\[.5ex] & = \left(
- \frac{1}{n} \text{det} \begin{bmatrix}
-\nicefrac{n}{\sigma^2} & 0 \\
0 & - \nicefrac{n}{2 \sigma^4}
\end{bmatrix}
\right)^{\frac{1}{2}}
\\[.5ex]
& = \frac{1}{\sigma^3}
\end{aligned}
$$
This is different than commonly used prior of $\nicefrac{1}
{\sigma^2}$.  This shows that the Jeffery's prior does not work
```

````
for (i in 1:3) tg$grobs[[font.vec[i]]] <-
editGrob(tg$grobs[[font.vec[i]]], gp=gpar(col=colors[i]))
for (i in 1:3) tg$grobs[[bg.vec[i]]] <-
editGrob(tg$grobs[[bg.vec[i]]], gp=gpar(fill=colors[i]))
x <- seq(0,1, len=100)
qplot(x, geom="blank")+
  annotation_custom(tg, xmin=0.4, ymin=1)+
  Uniform+Beta1+Beta2+
  ggtitle("Graph of Priors")+
  theme(axis.title.x = element_blank())+
  scale_colour_manual("", values = c(colors[3], colors[2],
colors[1])) +
  scale_linetype_manual("", values=c(line[3], line[2], line[1]))
+
  theme(legend.position = "bottom")
```

The Uniform has a mean of 0.27 and both beta distributions have a
mean of 0.30.  The uniform distribution has the lowest variance
and a block shape.  The Beta(10.2,23.8) has the highest variance
and a lower peak than the Beta(20.4,47.6).

\newpage
### Problem 3b
Suppose a player gets 5 hits in 40 at-bats.  For each of the
above priors: plot the likelihood, posterior and prior; compute
the probability that he player is better than a .200 hitter;
compute your best guess as to the batting average of the player;
compute a 95% credible interval for $p$.

```{r, echo=FALSE}
dist <- c("Likelihood", "Posterior", "Prior")
colors <- c("darkgrey", "navy", "maroon")

p <- 5/40
n <- 40
it <- 10000
Binomial.Likelihood <- data.frame("L"=rbinom(it, 1, p))

func_distplot <- function(Likelihood, prior, posterior, a, b,
name){
plot1 <- ggplot(Likelihood, aes(L))
+geom_density(aes(y=..density..,  col=dist[1]), lwd=1.5)+
  xlim(0, 1) + ylim(0, 8)+
  prior+posterior+
````

````
n <- 40
y <- 5
a0 <- 0.19
b0 <- 0.35
a <- y + a0
b <- n - y + b0

func_postdist <- function(x){dunif(x, .19, .35)*dbeta(x,
shape1=a, shape2=b)}
prior <-    stat_function(fun = dunif, args =
list(min=0.19,max=0.35), lwd = 1.5, linetype="dashed",
aes(col=dist[3]))
posterior <-stat_function(fun = func_postdist,  lwd = 1.5,
linetype="dotted", aes(col=dist[2]))

set.seed(040302)
x.sim <- seq(0, 1, length=1000)
y.sim <- func_postdist(x.sim)
df <- data.frame("Xsim" = x.sim, "Ysim"=y.sim)


plot1 <- ggplot(Binomial.Likelihood, aes(L))
+geom_density(aes(y=..density..,  col=dist[1]), lwd=1.5)+
  xlim(0, 1) + ylim(0, 20)+
  prior+posterior+
  ggtitle(paste("Unifrom(.19, .35) Prior, Binomial Likelihood,
Beta Posterior"))+
  theme(axis.title.x = element_blank())+
  scale_colour_manual("", values = c(colors[1], colors[2],
colors[3])) +
  theme(legend.position = "bottom")

#For the probability of > 0.2, we are essentially finding P(X>=
8), because 0.2 is **8**/40. We do this analytically with
integral, for each value of y and then add those probabilites up
lower <- 0.19
upper <- 0.35
f <- function(theta){dbinom(y, n, theta)/(upper - lower)}
num_integral <- integrate(f, .20, .35)
den_integral <- integrate(f, .19, .35)

pval <- num_integral$value/den_integral$value
```

```{r, echo=FALSE, eval=FALSE }
````

````
```{r, echo=FALSE, warning=FALSE, fig.width=10, fig.height=4}

CI.up <- 0.3057802  #run beforehand (above)
CI.low <- 0.1910938 #run beforehand (above)



plot2 <- ggplot(df, aes(Xsim, Ysim))+ geom_line(lwd = 1.5,
linetype="dotted", aes(col=dist[2]))+xlim(0, 1)+
  geom_ribbon(data=subset(df, Xsim>0.2), aes(ymax=Ysim), ymin=0,
fill=colors[2], color=NA, alpha=0.4)+
  annotate("text", x=0.3, y=5, label=paste("P(X > 0.200 | Y):",
decimal(pval, dec)), hjust=0, size=5)+
  annotate("text", x=0.3, y=8, label=paste("CI: (",
decimal(CI.low, dec), ",", decimal(CI.up, dec), ")"), size=5,
hjust=0)+
  ggtitle("Probability X > 0.200, Credible Interval")+
  theme(axis.title.x = element_blank())+
  scale_colour_manual("", values = c(colors[2])) +
  theme(legend.position = "bottom")

grid.arrange(plot1, plot2, nrow=1)


```
````

````
```{r, echo=FALSE, warning=FALSE, fig.width=10, fig.height=4}
#BETA PRIOR Beta(10.2,23.8)
n <- 40
y <- 5
a0 <- 10.2
b0 <- 23.8
a <- y + a0
b <- n - y + b0
prior <-    stat_function(fun = dbeta, args = list(shape1=a0,
shape2=b0), lwd = 1.5, linetype="dashed",  aes(col=dist[3]))
posterior <-stat_function(fun = dbeta, args = list(shape1=a,
shape2=b), lwd = 1.5, linetype="dotted",  aes(col=dist[2]))
func_distplot(Binomial.Likelihood, prior, posterior, a, b,
"Beta(10.2, 23.8) Prior, Binomial Likelihood, Beta Posterior")
```
````

```
g0 <- 89 #number of games played
n0 <- 47 #number of games won
p0 <- n0/g0
g1 <- 73 #remaining games
it <- 1000
Binomial.Likelihood <- data.frame("L"=rbinom(it, 162, p0))/162
#random gen for est games won in next 73 games

func_distplot3c <- function(Likelihood, a0, b0, name){

a <- n0 + a0
b <- g0 - n0 + b0
prior <-    stat_function(fun = dbeta, args = list(shape1=a0,
shape2=b0), lwd = 1.5, linetype="dashed", aes(col=dist[3]))
posterior <-stat_function(fun = dbeta, args = list(shape1=a,
shape2=b), lwd = 1.5, linetype="dotted",  aes(col=dist[2]))

mean.post <- a/(a + b)
mode.post <- (a - 1)/(a + b - 2)
med.post <- (a - (1/3))/(a + b - 2/3)

stat <- c("Mean", "Mode", "Est. Median")
stat.val <- c(mean.post, mode.post, med.post)
est.games <- stat.val*162 #total games played is 162
actual.games <- rep(84, 3) #actual games won = 89
difference <- actual.games - est.games
labels <- c("Posterior Value", "Est Games Won", "Actual Games
Won", "Difference")
vec.blank <- rep(NA, length(labels))
compare <- data.frame("Mean"=vec.blank, "Mode"=vec.blank, "Est.
Median" = vec.blank)
for (i in 1:3){
compare[,i] <- c(
                 decimal(stat.val[i], dec),
                 decimal(est.games[i], 3),
                 actual.games[i],
                 decimal(difference[i], 3)
                 )
}

rownames(compare) <- labels

plot <- ggplot(Likelihood, aes(L))
+geom_density(aes(y=..density..,  col=dist[1]), lwd=1.5)+
  xlim(0, 1) +
```

````
```{r, echo=FALSE, warning=FALSE, fig.width=10, fig.height=4}
a0 <- 0
b0 <- 0
func_distplot3c(Binomial.Likelihood, a0, b0, "Haldane's Prior:
Beta(0, 0)")
```




\newpage
## Problem 4
The following data represents the number of arrivals for 45 time
intervals of length 2 minutes at a cashier's desk at a
supermarket and are taken from Andersen (1980):
```{r}
Arrival <- c(rep(0,6), rep(1,18), rep(2,9), rep(3,7), rep(4, 4),
rep(5, 1))
```

### Problem 4a
For a Gamma(2,1) prior, obtain the posterior distribution under a
Poisson ($\lambda$) model for the data.  Draw the prior and the
posterior.  Note on your plot the mean, variance and mode of the
posterior.
```{r, echo=FALSE, fig.width=10, fig.height=4}
dist <- c("Likelihood", "Posterior", "Prior")
colors <- c("darkgrey", "navy", "maroon")
n <- length(Arrival)
a0 <- 2
b0 <- 1
a <- sum(Arrival) + a0
b <- n + b0
lam <- mean(Arrival)
prior <- stat_function(fun = dgamma, args =
list(shape=a0,rate=b0), lwd = 1.5, aes(col=dist[3],
linetype=dist[3]))
posterior <-   stat_function(fun = dgamma, args = list(shape=a,
rate=b), lwd = 1.5, aes(col=dist[2], linetype=dist[2]))

mean.post <- a/b
var.post <- a/b^2
mode.post <- (a-1)/b

it <- 10000
````
````

```
a <- sum(Arrival) + a0
b <- n + b0
lam <- mean(Arrival)
prior <- stat_function(fun = dgamma, args =
list(shape=a0,rate=b0), lwd = 1.5, aes(col=dist[3],
linetype=dist[3]))
posterior <-   stat_function(fun = dgamma, args = list(shape=a,
rate=b), lwd = 1.5, aes(col=dist[2], linetype=dist[2]))

mean.post <- a/b
var.post <- a/b^2
mode.post <- (a-1)/b

 ggplot(Poisson.Likelihood, aes(L))+
  prior+posterior+
  ggtitle("Gamma(.5, .00001) Prior, Poisson Likelihood, Gamma
Posterior")+
  geom_point(aes(x=mode.post, y=dgamma(mode.post, a, b)), size=2,
shape=21, color="grey45", stroke=1.5)+
   annotate("text", x=mode.post, y=dgamma(mode.post, a, b),
label=paste("Mode:", decimal(mode.post, dec)), hjust=1.1, size=5,
color="grey45", fontface="bold")+
  geom_point(aes(x=mean.post, y=dgamma(mean.post, a, b)), size=2,
shape=22, color="grey35", stroke=1.5)+
   annotate("text", x=mean.post, y=dgamma(mean.post, a, b),
label=paste("Mean:", decimal(mean.post, dec)), hjust=-.1,
size=5.5, color="grey35", fontface="bold")+
   annotate("text", x=2, y=dgamma(2, a, b),
label=paste("Variance:", decimal(var.post, dec)), hjust=-.1,
size=5, fontface="bold")+
  theme(axis.title.x = element_blank())+
  scale_colour_manual("", values = c(colors[2], colors[3])) +
scale_linetype_manual("", values = c("dotted", "dashed"))+
  theme(legend.position = "bottom")
```

\newpage
Problem 5
197 animals are distributed into four categories: $Y = (y_1, y_2, y_3, y_4)$ according ot the genetic linkage model $\left(\frac{2 + \theta}{4}, \frac{1 - \theta}{4}, \frac{ 1 - \theta}{4}, \frac{\theta}{4} \right)$

$$

````
    root.approx <- tail(k, n=1)
    it.completed <- length(k)
    # Once the difference between x0 and x1 becomes sufficiently
small, output the results.
    if (abs(x1 - x0) < tol & !is.na(abs(x1- x0)))
      {print(paste("Start at", start, ": Root approximation is",
root.approx, "with", it.completed, "iterations"))
      break}
    else if( it.completed == it){print(paste("Start at", start,
": diverges"))}
    else{ x0 <- x1}
  }
}
tol <- 1e-5
it <- 1000
```


### Problem 5a
$$
L(\theta \mid \pmb{Y} = (125, 18, 20 , 34)) \propto (2+
\theta)^{125} \cdot (1 - \theta)^{38} \cdot (\theta)^{34}
\\[.5ex]
$$


### Problem 5b
$$
L(\theta \mid \pmb{Y} = (14, 0, 1, 5)) \propto (2+ \theta)^{14}
\cdot (1 - \theta)^{1} \cdot (\theta)^{5}
$$


### Problem 5c
Use th Newton-Raphson to obtain the MLE $(\hat{\theta})$ for $Y =
(125, 18, 20, 34)$.  Start the algorithm at $\theta = .1, .2, .3,
.4, .6, .8$.
```{r, echo=FALSE, eval=FALSE}
#How do you assess convergence of the algorithm.
```


```{r, echo=FALSE}
y1 <- 125
y2 <- 18
````

````
```{r, echo=FALSE}
y1 <- 14
y2 <- 0
y3 <- 1
y4 <- 5
func_b<-function(x){
 (
  y1/(2+x)-(y2+y3)/(1-x)+y4/x
  ) / (
  -y1/(2+x)^2-(y2+y3)/(1-x)^2-y4/x^2
  )
}
 #-(14/(2+x)-1/(1-x)+5/x)/(14/(2+x)^2+1/(1-x)^2+5/x^2)

func_newton.raphson(func_b, 0.1, it, tol)
func_newton.raphson(func_b, 0.2, it, tol)
func_newton.raphson(func_b, 0.3, it, tol)
func_newton.raphson(func_b, 0.4, it, tol)
func_newton.raphson(func_b, 0.6, it, tol)
func_newton.raphson(func_b, 0.8, it, tol)
```
````