Bernoulli model with conjugate priors

Case study: Space shuttle Challenger disaster

Georgios P. Karagiannis @ MATH3341/4031 Bayesian statistics III/IV (practical implementation)

Back to README

```
rm(list=ls())
```

Aim

Students will become able to:

- produce Monte Carlo approximations of posterior quantities required for Bayesian analysis with the RJAGS R package
- implement Bayesian posterior analysis in R with RJAGS package

Students are not required to learn by heart any of the concepts discussed

---

Reference list

The material about RJAGS package is not examinable material, but it is provided for the interested student. It contains references that students can follow if they want to further explore the concepts introdced.

- Lecture notes:
  - the examples and exercises related to the Bernoulli model with conjugate prior
- Application (optional):
  - Dalal, S. R., Fowlkes, E. B., & Hoadley, B. (1989). Risk analysis of the space shuttle: Pre-Challenger prediction of failure. Journal of the American Statistical Association, 84(408), 945-957.
- References for *RJAGS*:
  - JAGS homepage
  - JAGS R CRAN Repository
  - JAGS Reference Manual
  - JAGS user manual
- Reference for *R*:
  - Cheat sheet with basic commands
- Reference of *rmarkdown* (optional):
  - R Markdown cheatsheet
  - R Markdown Reference Guide
  - knitr options
- Reference for *Latex* (optional):
  - Latex Cheat Sheet

New software

- R package `rjags` functions:

- jags.model{rjags}

- jags.samples{rjags}
- update{rjags}

Application: Challenger O-ring

On January 28, 1986, a routine launch was anticipated for the Challenger space shuttle. Seventy-three seconds into the flight, disaster happened: the shuttle broke apart, killing all seven crew members on board. Here is the video. The Rogers commission concluded that the Challenger accident was caused by gas leak through the 6 O-ring joints of the shuttle. Essentially the presence of distressed O-ring joints was the crucial factor that caused the explosion.

Dalal, Fowlkes and Hoadley (1989) analysed a dataset that contains the presence of distressed O-rings, the temperature in the platform, and the leak check pressure for 23 previous shuttle flights. The the data-set is provided below, where in column *Defective.O.rings*, (1) stands for presence of distressed O-rings, and (0) for absence, while the rest of the columns are self explained.

```
# Load R package for printing
library(knitr)
library(kableExtra)
```

```
# load the data
#mydata <- read.csv("./challenger_data.csv")
mydata <- read.csv("https://raw.githubusercontent.com/georgios-stats/Bayesian_Statistics/master/Computer
# print data
## (that's a sophisticated command with fancy output, feel free to ignore it)
kable(mydata)%>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Here,

- we will use only the observations in variable *Defective.O.rings* (so ignore the temperature and pressure measurements) from *04/12/1981* to *01/12/1986* (23 flights), with purpose to:
- learn the (limiting relative) frequency of having defective O-rings?
- predict the outcome in the 24th flight on *1/28/86*, given the information from the previous 23 flights considered.

The model: Bernoulli model

Let y_i denote the presence of a defective O-ring in the ith flight (0 for absence, and 1 for presence).

Regarding the statistical model, we assume that y_i can be modeled as observations generated independently from a Bernoulli distribution with with common parameter p. Here, p denotes the relative frequency of defective O-rings at any flight.

Regarding the prior model, we assign a Beta prior distribution with fixed hyper-parameters $a_0 = 1.0$, $b_0 = 1.0$ on p to account for its uncertainty.

The Bayesian hierarchical model under consideration is:

$$\begin{cases} y_i|p \sim & \text{Bernoulli}(p), \quad \text{for, } i = 1, ..., n \\ p \sim & \text{Beta}(a_0, b_0), \end{cases}$$

with hyper-parameter values $a_0 = 1.0$, $b_0 = 1.0$.

Task

We write a RJAGS program implementing the hierarchical model above, in order to generate a sample of size $n = 100000$ from the posterior distribution

$$p^{(j)} \sim \pi(p|y_{1:n}) \ , \quad j = 1, ..., n.$$

. . . your answer

For now I am doing it for you.

step 1

Load the library

```
# Load rjags
library("rjags")
```

step 2

Create an input script, for rjags, containing the Bayesian hierarchical model

```
# Input parameters  : n, y, a_0, b_0
# output parameters : p
hierarhicalmodel <- "

  model {

  # this is related tot he sampling distribution

    for ( i in 1 : n ) {
      y[ i ] ~ dbern( p )
    }

   # this is related to the prior distributions

    p ~ dbeta( a_0 , b_0 )

  }

"
```

step 3

Create an input list, for jags, containing the data and fixed parameters of the model

```
y_obs <- mydata[ -nrow(mydata) , 4 ] # exclude the last row, and use only the 4th column

y_obs <- as.numeric( y_obs == 1 )     # make it numeric
```

```
n_obs <- length( y_obs )

a_0 <- 1.0

b_0 <- 1.0

data.bayes <- list(y = y_obs,
                    n = n_obs,
                    a_0 = a_0,
                    b_0 = b_0)
```

step 4

Create an input list, for jags, containing the data and fixed parameters of the model

```
model.smpl <- jags.model( file = textConnection(hierarhicalmodel),
                          data = data.bayes)
```

For further reading:

Alternatively we could have used the routine `coda.samples{rjags}` returning the same information but with an object of type `mcmc.list` that can be analysed by the tools provided in the R packages `coda` and `boa`. We do not discuss about these packages as the are not in stable release yet, and they may contain bugs. Pls keep an eye on them.

step 5

Initialize the sampler with $N_{\text{adapt}} = 1000$ iterations.

- This is a warming-up procedure (used as a black-box), where the sampler is automatically tuned and calibated before it starts generatign your samples.
- Regarding $N_{\text{adapt}} = 1000$, the larger the better.

```
adapt( object = model.smpl,
       n.iter = 1000 )
```

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

step 6

Generate a posterior sample of size $N = 10000$.

Use

- `jags.samples{rjags}`

We need to pay attention on two flaqs:

- the `n.iter`: the total size of the total sample sequence.
- the `variable.names`: it specifies the names of the random variables corresponding to the posterior samples I am interested in generating

```
N = 10000      # the size of the sample we ll gonna get
output = jags.samples( model= model.smpl,          # the model
                       variable.names= c("p"),    # names of variables to be sampled
                       n.iter = N                # size of sample
                       )
save.image(file="BernoulliModel.RData")
```

Check the names of the variables sampled

- use `names {base}`

```
names(output)
```

```
## [1] "p"
```

Check the dimensions of each of the variables sampled

- use `dim {base}`

```
dim( output$p )
```

```
##           iteration     chain
##         1     10000         1
```

```
# the first dimension is the numbers of columns of the variable
# the second dimention is the size of the sample drawn
# the third dimension is the number of the sub-samples drawn (in our case it is just 1)
```

Copy the sample of each variable in a vector with a more friendly name…

```
pr.smpl <-output$p
```

Task

Print the trace plot of the sample drawn from the posterior distribution

$$p^{(j)} \sim \pi(p|y_{1:n}) \quad j = 1, ..., N$$

.

Use functions:

- `plot {graphics}`.

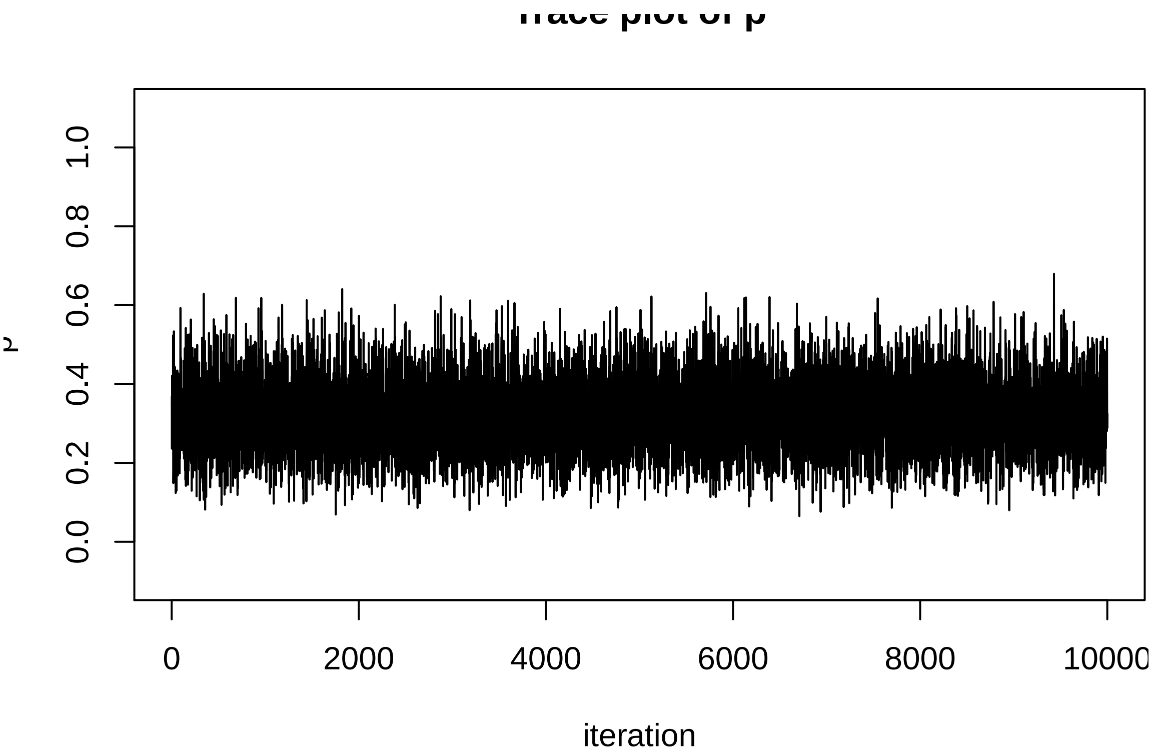
A glimpse forward into Term 2 of Bayesian stats:

- A good quality sample for the purposes of Monte Carlo integration is the one whose trace plot looks completely uncorrelated.
  - e.g. like the independence assumption of the residuals in the simple linear regression in SC2.
- To improve the quality of the sample, you can go back to the sampling stem and play with the flaq values of `n.iter` and `thin` in `jags.samples{rjags}`.

… your answer

```
# extract the sample from the jags object
pr.smpl <-pr.smpl[1,,]
```

```
# draw the trace plots
z <- pr.smpl
plot(z,
     type = "l",
     main = "Trace plot of p",
     xlab = "iteration",
     ylab = "p",
     ylim = c(-0.1,1.1)
     )
```



Well, they all look prety random. That's cool!

Task

- Compute and plot the MC approximation for the posterior distribution cumulative function (CDF) of p, by computing the empirical CDF as

  $$\begin{aligned} F_\pi(p \leq c|y_{1:n}) &= E_\pi(1(p \leq c)|y_{1:n}) \\ &\approx \frac{1}{N}\sum_{i=1}^{N} 1(p^{(i)} \leq c) \end{aligned}$$

  for $c \in (0, 1)$
- Compute and plot the exact posterior CDF of p, which is the CDF of the distribution

  $$p|y_{1:n} \sim \text{Beta}(a_n, b_n)$$

  where

  $$\begin{aligned} a_n =& a_0 + n\bar{y} \\ b_n =& b_0 + n - n\bar{y}. \end{aligned}$$

Print them on the same plot.

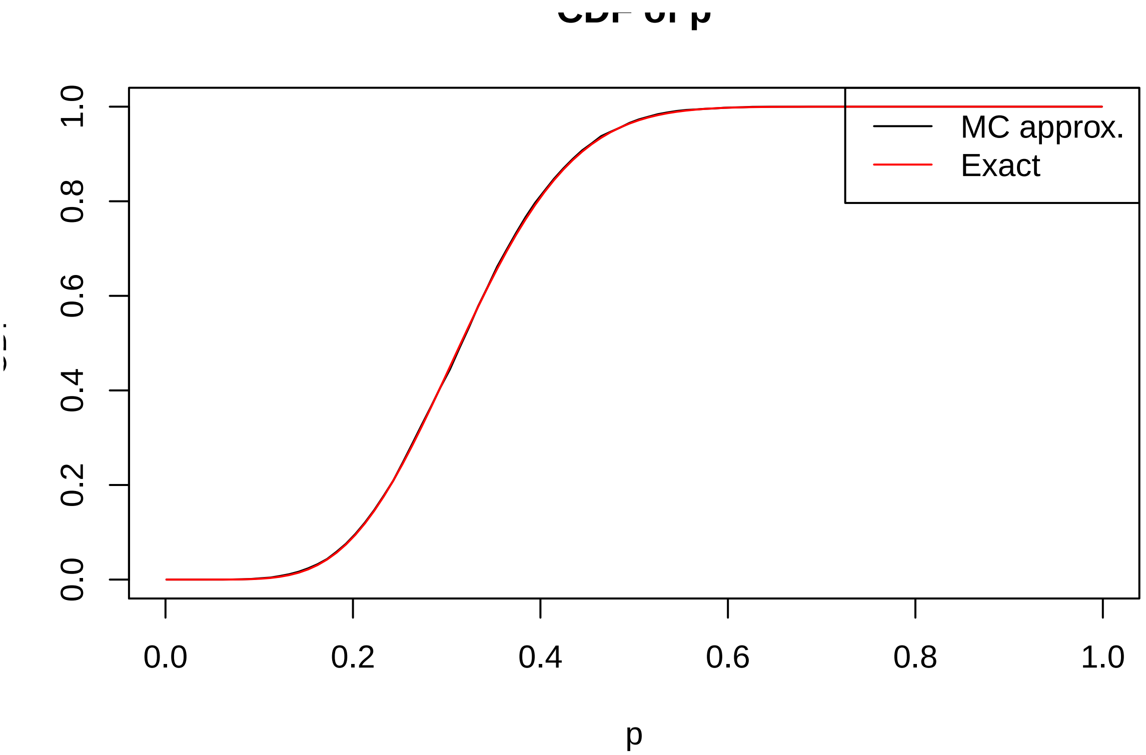
Use functions:

- `lines {graphics}`

… your answer

Regarding the posterior CDF …

```
# Draw the histogram as the MC approximate of the CDF
z <- pr.smpl
x_plot <- seq( from = 0.001, to = 0.999, length.out = 100)
y_plot <- rep(NaN, 100)
for (i in 1:100) y_plot[i] <- mean(z<=x_plot[i])
plot(x_plot,
     y_plot,
     type = "l",
     main = "CDF of p",
     xlab = "p",
     ylab = "CDF")
# Draw the Exact CDF
a_n = a_0+n_obs*mean(y_obs)
b_n = b_0+n_obs-n_obs*mean(y_obs)
x_plot <- seq( from = 0.001, to = 0.999, length.out = 100)
y_plot <- pbeta(x_plot, a_n, b_n )
lines( x_plot,
       y_plot,
       col = 'red'
       )
# Create a legend
legend("topright",
       legend=c("MC approx.", "Exact"),
       lty = c(1,1),
       col=c("black", "red"))
```



Well, we can observe a perfect match!!!

Task

- Compute and plot the MC approximation for the posterior distribution density of $p|y_{1:n}$ as a histogram
  - use the function `hist {graphics}` provided from R.
  - Just for your information, the mathematical formula of a histogram estimator is

    $$\pi(p|y_{1:n}) \approx \frac{1}{2\epsilon}\frac{1}{N}\sum_{j=1}^{N} 1\left(p^{(j)} \in (p-\epsilon, p+\epsilon]\right) \text{ for small } \epsilon$$

    for $p \in (0,1)$, however, you do not need to code it. Just use the function `hist {graphics}`
- Compute and plot the exact the posterior distribution density of $p|y_{1:n}$ which is the PDF of

  $$p|y_{1:n} \sim \text{Beta}(a_n, b_n)$$

  where

  $$\begin{aligned} a_n =& a_0 + n\bar{y} \\ b_n =& b_0 + n - n\bar{y}. \end{aligned}$$

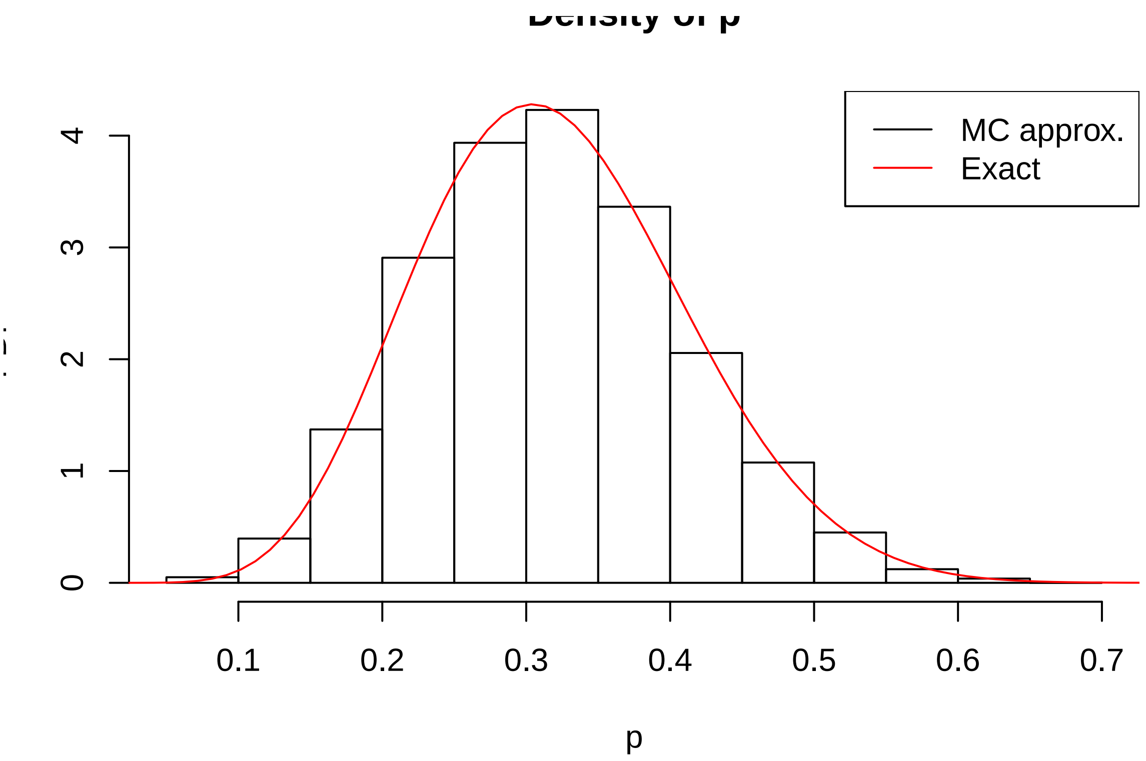
Use functions:

- `hist {graphics}`

... your answer

Regarding the posterior PDF...

```
# Draw the histogram as the MC approximate of the PDF
z <- pr.smpl
hist(z,
     probability = TRUE,
     main = "Density of p",
     xlab = "p",
     ylab = "PDF")
# Draw the Exact PDF
a_n = a_0+n_obs*mean(y_obs)
b_n = b_0+n_obs-n_obs*mean(y_obs)
x_plot <- seq( from = 0.001, to = 0.999, length.out = 100)
y_plot <- dbeta(x_plot, a_n, b_n )
lines( x_plot,
       y_plot,
       col = 'red'
       )
# Create a legend
legend("topright",
       legend=c("MC approx.", "Exact"),
       lty = c(1,1),
       col=c("black", "red"))
```



Well, we can observe a perfect match!!!

Task

- Compute the MC approximate of the posterior probability that the frequence parameter p is greater than or equal to 0.5 as

$$\begin{aligned}
\Pr_{\pi}(p \geq 0.5|y_{1:n}) =& 1 - \Pr_{\pi}(p < 0.5|y_{1:n}) \qquad (1)\\
=& 1 - \mathrm{E}_{\pi}(1(p < 0.5)|y_{1:n}) \qquad (2)\\
\approx& \frac{1}{N}\sum_{i=1}^{N}(1(p^{(i)} < 0.5)) \qquad (3)
\end{aligned}$$

- Compute its exact value of as

$$\begin{aligned}
\Pr_{\pi}(p \geq 0.5|y_{1:n}) =& 1 - \Pr_{\pi}(p < 0.5|y_{1:n}) \qquad (4)\\
& 1 - \Pr_{\mathrm{Beta}(a_n,b_n)}(p < 0.5|y_{1:n}) \qquad (5)\\
& 1 - \int_{-\infty}^{0.5}\mathrm{Beta}(p|a_n, b_n)\mathrm{d}p \qquad (6)
\end{aligned}$$

  where

$$\begin{aligned}
a_n =& a_0 + n\bar{y}\\
b_n =& b_0 + n - n\bar{y}
\end{aligned}$$

. . . your answer

The MC approximate is

```r
# Draw the histogram as the MC approximate of the PDF
z <- pr.smpl
Pr.p.mc <- 1-mean(z<=0.5)
Pr.p.mc
```

```
## [1] 0.0306
```

The exact value is

```r
a_n = a_0+n_obs*mean(y_obs)
b_n = b_0+n_obs-n_obs*mean(y_obs)
Pr.p.ex <- 1- pbeta(0.5, a_n, b_n)
Pr.p.ex
```

```
## [1] 0.03195733
```

The MC approximate is close to the exact values.

Task

- compute the MC approximate of the 95% posterior equal tail credible interval of the frequency parameter p is

$$[Q_{0.025}(p|y_{1:n})\ ,\ Q_{0.975}(p|y_{1:n})]$$

where $Q_{\alpha}(p|y_{1:n})$ is the α-th quantile of the posterior distribution of p

- compute the exact 95% posterior equal tail credible interval of the frequency parameter p is

$$[Q_{0.025}(p|y_{1:n}) \ , \ Q_{0.975}(p|y_{1:n})]$$

  where $Q_{\alpha}(p|y_{1:n})$ is the α-th quantile of the posterior distribution of p which is

$$p \sim \text{Beta}(a_n, b_n)$$

  where

$$\begin{aligned} a_n =& a_0 + n\bar{y} \\ b_n =& b_0 + n - n\bar{y} \end{aligned}$$

Use:

- quantile{stats}
- qbeta{stats}

... your answer

The MC approximate 95% credible interval for p is

```r
z <- pr.smpl
CI.mc <- quantile(z, probs = c(0.025, 0.0975))
CI.mc
```

```
##      2.5%     9.75%
## 0.1535355 0.2031534
```

and the exact
95% credible interval for p is

```r
a_n <- a_0+n_obs*mean(y_obs)
b_n <- b_0+n_obs-n_obs*mean(y_obs)
CI.exact <- qbeta(c(0.025, 0.0975),
                  shape1 = a_n,
                  shape2 = b_n)
CI.exact
```

```
## [1] 0.1563023 0.2038193
```

Task

- Compute the MC approximate of the posterior expected value of p, $\text{E}_{\pi}(p|y_{1:n})$, as

$$\text{E}_{\pi}(p|y_{1:n}) \approx \frac{1}{N}\sum_{i=1}^{N} p^{(i)}$$

- Compute exact value of $\text{E}(p|y_{1:n})$ which is

$$\text{E}_{\pi}(p|y_{1:n}) = \frac{a_n}{a_n + b_n}$$

where

$$\begin{aligned} a_n =& a_0 + n\bar{y} \\ b_n =& b_0 + n - n\bar{y} \end{aligned}$$

. . . your answer

Regarding the MC approximate of $\text{E}(p|y_{1:n})$ it is

```r
# Draw the histogram as the MC approximate of the PDF
z <- pr.smpl
E_mc <- mean(z)
print(E_mc)
```

```
## [1] 0.3192045
```

Regarding the exact value of $\text{E}(p|y_{1:n})$ it is

```r
a_n <- a_0+n_obs*mean(y_obs)
b_n <- b_0+n_obs-n_obs*mean(y_obs)
E_exact <- a_n / (a_n+b_n)
print(E_exact)
```

```
## [1] 0.32
```

We observe that the two values are prety close each other:

- I observe that the MC approximation is 0.3192045, while the Exact value is 0.32.
- So their absolute different is 7.9551068×10^{-4} .
- So the MC approximation provides a good approximation!!!

Task

- Compute the MC approximate of the posterior expected value of the odds parameter $\theta = \frac{p}{1-p}$ which is denoted as $\text{E}_{\pi}(\theta|y_{1:n})$.

$$\text{E}_{\pi}(\theta|y_{1:n}) = \text{E}_{\pi}(\frac{p}{1-p}|y_{1:n}) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{p^{(i)}}{1-p^{(i)}}$$

- Compute the MC approximation of $\text{E}_{\pi}(\theta|y_{1:n})$ with its exact value which is

$$\text{E}_{\pi}(\theta|y_{1:n}) = \text{E}_{\pi}(\frac{p}{1-p}|y_{1:n}) = \frac{a_n}{b_n - 1}$$

where

$$\begin{aligned} a_n =& a_0 + n\bar{y} \\ b_n =& b_0 + n - n\bar{y} \end{aligned}$$

... your answer

Regarding the MC approximate of $\mathrm{E}(\theta|y_{1:n})$ it is

```
# Draw the histogram as the MC approximate of the PDF
z <- pr.smpl
theta <- z/(1-z)
E_mc <- mean(z)
print(E_mc)
```

```
## [1] 0.3192045
```

Regarding the exact value of $\mathrm{E}(\theta|y_{1:n})$ it is

```
a_n <- a_0+n_obs*mean(y_obs)
b_n <- b_0+n_obs-n_obs*mean(y_obs)
E_exact <-a_n/(b_n-1)
print(E_exact)
```

```
## [1] 0.5
```

We observe that the two values are prety close each other:

- I observe that the MC approximation is 0.3192045, while the Exact value is 0.5.
- So their absolute different is 0.1807955.
- So the MC approximation provides a good approximation!!!

Task

- Compute the MC approximation of the predictive distribution mass function of $y_{n+1}|y_{1:n}$, as

$$
\begin{aligned}
f_\pi(y_{n+1}=c|y_{1:n}) &= \int_0^1 f(y_{n+1}=c|p)\pi(p|y_{1:n})\mathrm{d}p, \quad c\in\{0,1\} \\
&= \int_0^1 p^c(1-p)^{1-c}\pi(p|y_{1:n})\mathrm{d}p, \quad c\in\{0,1\} \\
&= \begin{cases} \int_0^1 (1-p)\pi(p|y_{1:n})\mathrm{d}p & \text{, c = 0} \\ \int_0^1 p\pi(p|y_{1:n})\mathrm{d}p & \text{, c = 1} \end{cases} \\
&= \begin{cases} 1-\mathrm{E}(p|y_{1:n}) & \text{, c = 0} \\ \mathrm{E}(p|y_{1:n})\mathrm{d}p & \text{, c = 1} \end{cases} \\
&\approx \begin{cases} 1-\frac{1}{N}\sum_{j=1}^N p^{(j)} & \text{, c = 0} \\ \frac{1}{N}\sum_{j=1}^N p^{(j)} & \text{, c = 1} \end{cases}
\end{aligned}
$$

by using a barplot.

- Compute the exact predictive distribution mass function of $y_{n+1}|y_{1:n}$, as

$$
\begin{aligned}
f(y_{n+1}=c|y_{1:n}) =& \frac{B(a_n+c, b_n+1-c)}{B(a_n,b_n)} 1(c\in\{0,1\}) \\
=& \begin{cases} \frac{B(a_n,b_n+1)}{B(a_n,b_n)} & \text{, c} = 0 \\ \\ \frac{B(a_n+1,b_n)}{B(a_n,b_n)} & \text{, c} = 1 \end{cases}
\end{aligned}
$$

where

$$
\begin{aligned}
a_n =& a_0 + n\bar{y} \\
b_n =& b_0 + n - n\bar{y}
\end{aligned}
$$

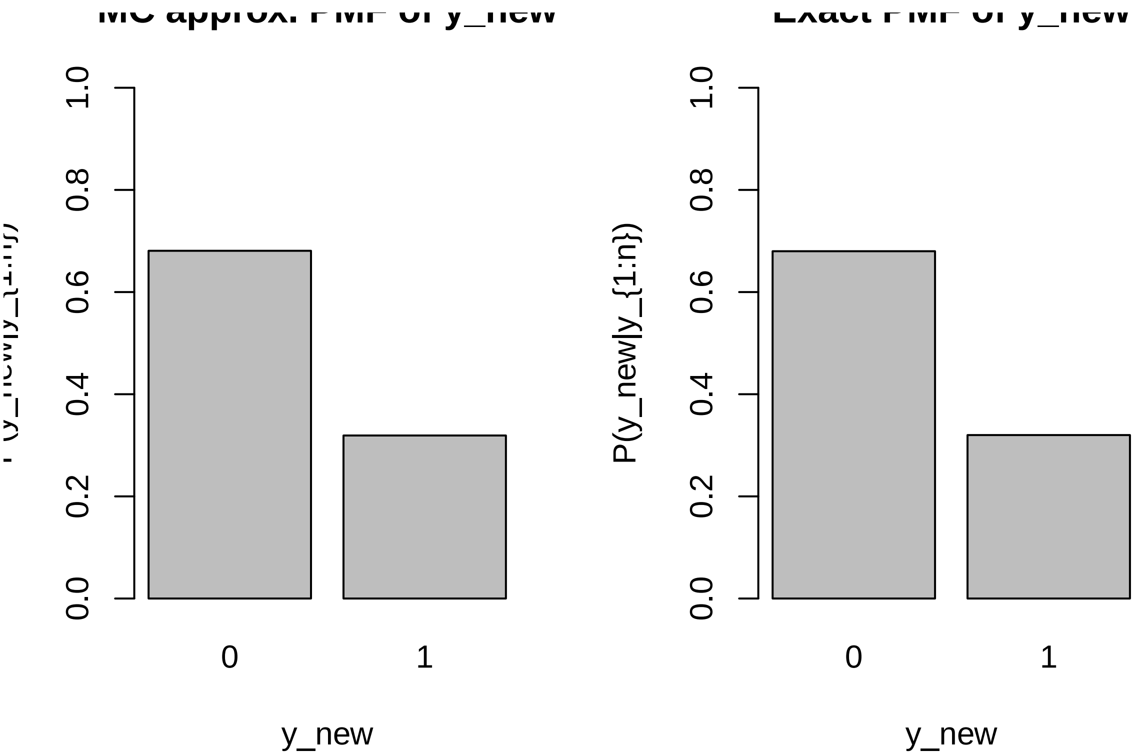
by using a barplot.

Use functions:

- `barplot {graphics}`,

. . . your answer

```
par(mfrow=c(1,2))
# Draw the histogram as the MC approximate of the PDF
z <- pr.smpl
pmf_y_new_0.mc <- 1-mean(z)
pmf_y_new_1.mc <- mean(z)
## draw
barplot( c(pmf_y_new_0.mc , pmf_y_new_1.mc),
         names.arg= c('0','1'),
         main = "MC approx. PMF of y_new",
         xlab = "y_new",
         ylab = "P(y_new|y_{1:n})",
         ylim = c(0,1))
# Draw the histogram as the Exact of the PDF
## compute
a_n = a_0+n_obs*mean(y_obs)
b_n = b_0+n_obs-n_obs*mean(y_obs)
pmf_y_new_0 <- beta(a_n+0,b_n+1-0) / beta(a_n,b_n)
pmf_y_new_1 <- beta(a_n+1,b_n+1-1) / beta(a_n,b_n)
## draw
barplot( c(pmf_y_new_0,pmf_y_new_1),
         names.arg= c('0','1'),
         main = "Exact PMF of y_new",
         xlab = "y_new",
         ylab = "P(y_new|y_{1:n})",
         ylim = c(0,1))
```



We observe that the two plots are close each other, and the MC approximation is good!

Discussion

Now, it is January 27, 1986, and you take part in the 3-hour teleconference with people from Morton Thiokol, Marshall space flight center, and Kennedy space center.

The forcast says that tommorrow the temperature too frosty. This is a bit unusual in the area and NASA people wonder if temperature can cause problems.

You perform the aforesaid statistical analysis and you find that the predictive probability in the next flight that there will be at least a defective O ring is less than 0.5.

You report your findinds, and the next day, January 28, 1986 explodes due to the presence of a defective O-ring component.

What did it go wrong with your aforesaid analysis?

How could you possibly do your analysis so that it can succesfully sugest you that the next flight (the one on January 28, 1986) will be a disaster if it is preformed?

If you have finished this handout, you can proceed tot he next [Bernoulli regression model].