Project 6

Santiago Castro Dau, June Monge, Rachita Kumar, Sarah Lötscher

2022-04-11

Problem 15: Monte Carlo estimation of an expected value

Proof 1: $\mathbb{E}[\hat{g}(X)] = \mathbb{E}[g(X)]$

Using the properties of the expectation and the fact that $\mathbb{E}[g(X_i)] = \mathbb{E}[g(X)]$:

$$\mathbb{E}[\hat{g}(X)] = \mathbb{E}[\frac{1}{N}\sum_{i=1}^{N} g(X_i)] = \frac{1}{N}\mathbb{E}[\sum_{i=1}^{N} g(X_i)] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[g(X_i)] = \frac{N}{N}\mathbb{E}[g(X)] = \mathbb{E}[g(X)]$$

Proof 2: $Var(\hat{g}(X)) = \frac{Var(g(X))}{N}$

Bienaymé's identity states that:

$$Var\left(\sum_{i=1}^{N} X_i\right) = \sum_{i=1}^{N}\mathrm{Var}(X_i) + \sum_{\substack{i,j=1\\ i\neq j}}^{N}\mathrm{Cov}(X_i, X_j) = \sum_{i,j=1}^{N}\mathrm{Cov}(X_i, X_j)$$

Since the covariance between any pair of independent random variables is zero we get the following.

$$Var\left(\sum_{i=1}^{N} X_i\right) = \sum_{i=1}^{N}\mathrm{Var}(X_i)$$

Specifically for our proof we have that:

$$Var(\hat{g}(X)) = Var(\frac{\sum_{i=1}^{N} g(X_i)}{N})$$

Using one of the properties of the variance we can pull out the constant N and obtain the following:

$$Var(\frac{\sum_{i=1}^{N} g(X_i)}{N}) = \frac{1}{N^2}Var(\sum_{i=1}^{N} g(X_i))$$

Then using Bienaymé's identity we have that:

$$\frac{1}{N^2}Var(\sum_{i=1}^{N} g(X_i)) = \frac{1}{N^2}\sum_{i=1}^{N} Var(g(X_i))$$

Finally since all the X_i's are i.i.d with variance equal to $Var(g(X))$ we arrive at the expression we had seek to proof.

$$\frac{1}{N^2}\sum_{i=1}^{N} Var(g(X_i)) = \frac{N}{N^2} Var(g(X)) = \frac{Var(g(X))}{N}$$

Problem 16: Sampling in the Rain Network

a) Derive the expressions for $P(C = T|R = T, S = T, W = T)$, $P(C = T|R = F, S = T, W = T)$, $P(R = T|C = T, S = T, W = T)$ and $P(R = T|C = F, S = T, W = T)$ and compute their values.

a.i)

$$P(C = T|R = T, S = T, W = T) = P(C = T|R = T, S = T)$$

$$= \frac{P(C = T)P(S = T, R = T|C = T)}{P(S = T, R = T)}$$

$$= \frac{P(C = T)P(S = T|C = T)P(R = T|C = T)}{P(C = T)P(S = T|C = T)P(R = T|C = T) + P(C = F)P(S = T|C = F)P(R = T|C = F)}$$

$$= \frac{0.5 * 0.1 * 0.8}{0.53 * 0.1 * 0.8 + 0.5 * 0.5 * 0.2} = 0.4444$$

a.ii)

$$P(C = T|R = F, S = T, W = T) =$$

$$= \frac{P(C = T)P(S = T|C = T)P(R = F|C = T)}{P(C = T)P(S = T|C = T)P(R = F|C = T) + P(C = F)P(S = T|C = F)P(R = F|C = F)}$$

$$= \frac{0.5 * 0.1 * 0.2}{0.5 * 0.1 * 0.2 + 0.5 * 0.5 * 0.8} = 0.04761905$$

a.iii)

$$P(R = T|C = T, S = T, W = T) =$$

$$\frac{P(R = T|C = T, S = T)P(W = T|R = T, C = T, S = T)}{P(W = T|C = T, S = T)} =$$

$$\frac{P(R = T|C = T)P(W = T|R = T, S = T)}{P(W = T|C = T, S = T)} =$$

$$\frac{P(R = T|C = T)P(W = T|R = T, S = T)}{P(R = T|C = T)P(W = T|R = T, S = T) + P(R = F|C = T)P(W = T|R = F, S = T)}$$

$$= \frac{0.8 * 0.99}{0.8 * 0.99 + 0.2 * 0.9} = 0.8148$$

a.iv)

$$P(R = T|C = F, S = T, W = T) =$$

$$\frac{P(R = T|C = F)P(W = T|R = T, S = T)}{P(R = T|C = F)P(W = T|R = T, S = T) + P(R = F|C = F)P(W = T|R = F, S = T)}$$

$$= \frac{0.2 * 0.99}{0.2 * 0.99 + 0.8 * 0.9} = 0.2156863$$

b) Implement the Gibbs sampler for the Bayesian network

```r
#returns 1 with probability p, and 0 with probability 1-p
rbernoulli=function(p){return(1*runif(1)<p)}



# sample from distribution X given Y above
sample_CgivenR = function(R){
 if(R==0){
   C = rbernoulli(0.0476) # returns 1 with probability 0.2; otherwise 0
 } else {
   C = rbernoulli(0.4444)
 }
  return(C)
}
#' sample from distribution Y given X above
sample_RgivenC = function(C){
  if(C==0){
    R = rbernoulli(0.2157)
  } else {
    R = rbernoulli(0.8148)
  }
  return(R)
}
set.seed(100)
niter = 100
C = rep(0,niter)
R = rep(0,niter)
C[1]=1
R[1]=1 # start from (1,1)
for(i in 2:niter){
  C[i] = sample_CgivenR(R[i-1])
  R[i] = sample_RgivenC(C[i])
}
res = data.frame(C=C,R=R)
```

```r
#Display a 2-by-2 table for the sampled R and C.
#1=True, 0=False
#Row is Cloudy, Column is Rain

table=table(res)/niter
table
```

```
##    R
## C      0    1
##   0 0.64 0.19
##   1 0.04 0.13
```

c) Estimate the marginal probability of rain, given that the sprinkler is on and the grass is wet

You sum over the C’s/ marginalised over C.

```r
colSums(table)
```

```
##    0    1
```

```
## 0.68 0.32
```

d) Draw 50,000 samples instead of 100 using the Gibbs sampler

```
niter = 50000
C = rep(0,niter)
R = rep(0,niter)
C[1]=1
R[1]=1 # start from (1,1)
for(i in 2:niter){
  C[i] = sample_CgivenR(R[i-1])
  R[i] = sample_RgivenC(C[i])
}
res50000run1 = data.frame(C=C,R=R)
head(res50000run1)
```

```
##   C R
## 1 1 1
## 2 1 1
## 3 1 0
## 4 0 0
## 5 0 0
## 6 0 0
```

```
#Print the table
table50000run1=table(res50000run1)/niter
table50000run1
```

```
##    R
## C         0       1
##   0 0.64474 0.17914
##   1 0.03194 0.14418
```

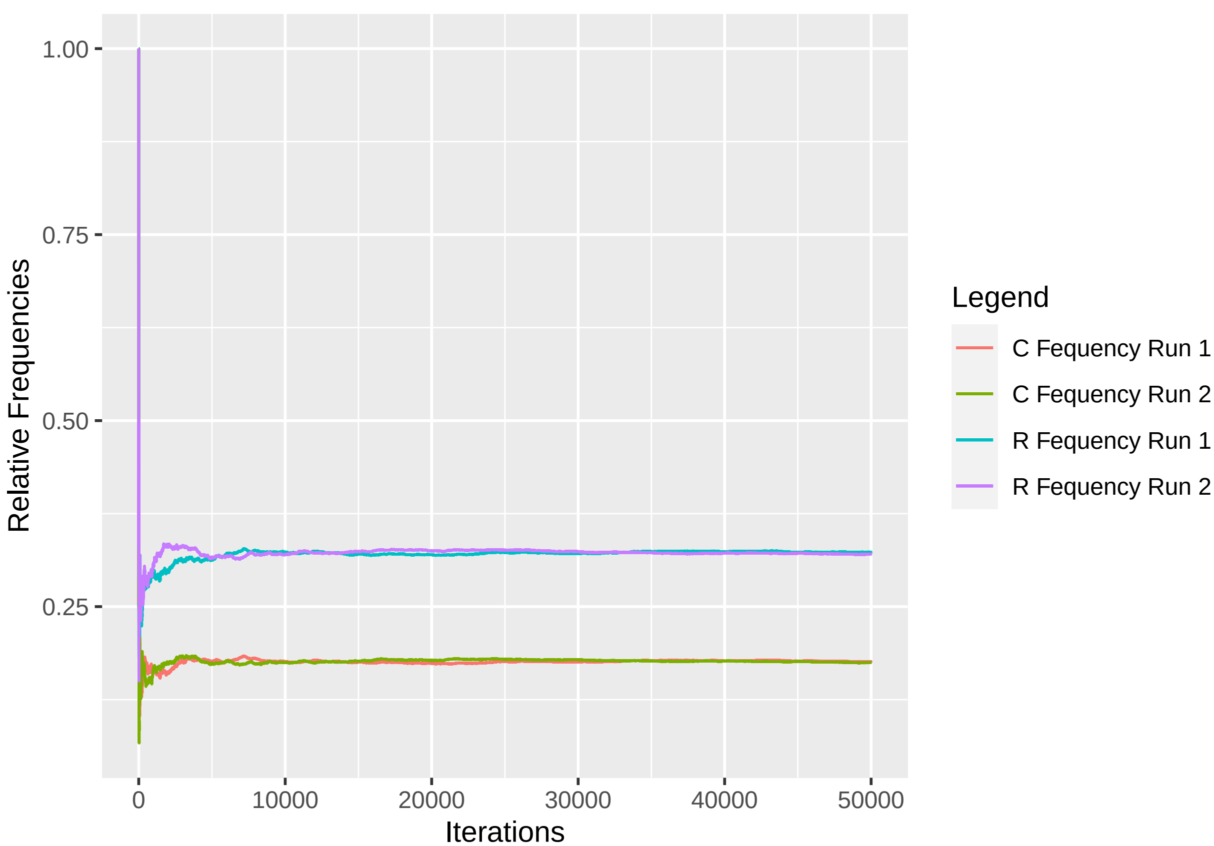
```
colSums(table50000run1)
```

```
##       0       1
## 0.67668 0.32332
```

e) Plot the relative frequencies of R = T and C = T up to each iteration t against t, for two independent runs of the sampler. Suggest a burn-in time based on this plot.

```
#First Run of Gibbs sampler
freqC<-c()
freqR<-c()
for(i in 1:niter){
  freqC<-append(freqC,sum(res50000run1$C[1:i])/i)
  freqR<-append(freqR,sum(res50000run1$R[1:i])/i)
}

#Second Run of Gibbs sampler
niter = 50000
C = rep(0,niter)
R = rep(0,niter)
C[1]=1
R[1]=1 # start from (1,1)
for(i in 2:niter){
```

```
  C[i] = sample_CgivenR(R[i-1])
  R[i] = sample_RgivenC(C[i])
}
res50000run2 = data.frame(C=C,R=R)


freqC2<-c()
freqR2<-c()
for(i in 1:niter){
  freqC2<-append(freqC2,sum(res50000run2$C[1:i])/i)
  freqR2<-append(freqR2,sum(res50000run2$R[1:i])/i)
}



df <- data.frame((1:niter),freqC,freqR,freqC2,freqR2)
ggplot(df, aes(1:niter)) +
  geom_line(aes(y=freqC, colour="C Fequency Run 1")) +
  geom_line(aes(y=freqR, colour="R Fequency Run 1")) +
  geom_line(aes(y=freqC2, colour="C Fequency Run 2")) +
  geom_line(aes(y=freqR2, colour="R Fequency Run 2")) + ylab("Relative Frequencies") + xlab("Iterations
```



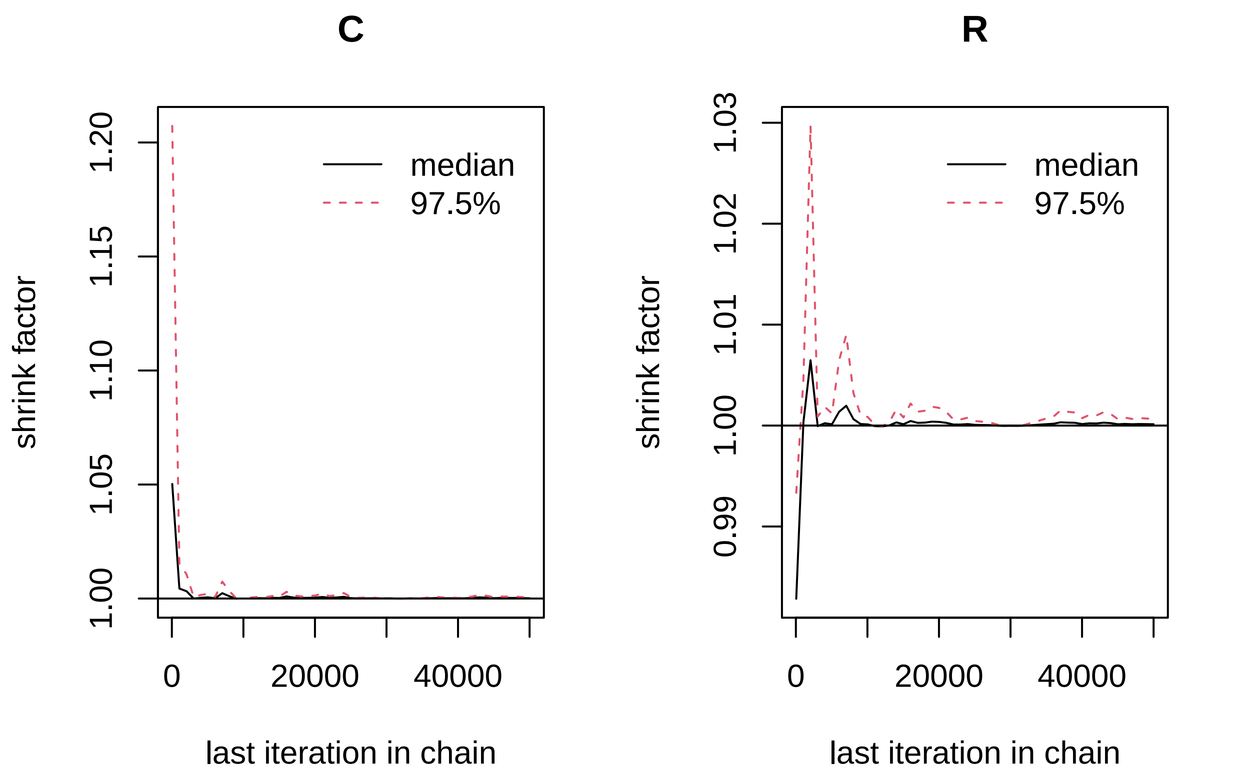
Suggested burn-in time based on this plot is about ~5000 iterations.

f) Apply the Gelman and Rubin test

```r
mcmc1<-mcmc(data=res50000run1 , start = 1, end = niter)
mcmc2<-mcmc(data=res50000run2 , start = 1, end = niter)

combinedchains = mcmc.list(mcmc1, mcmc2)

gelman.diag(combinedchains)
```

```
## Potential scale reduction factors:
##
##   Point est. Upper C.I.
## C          1          1
## R          1          1
##
## Multivariate psrf
##
## 1
```

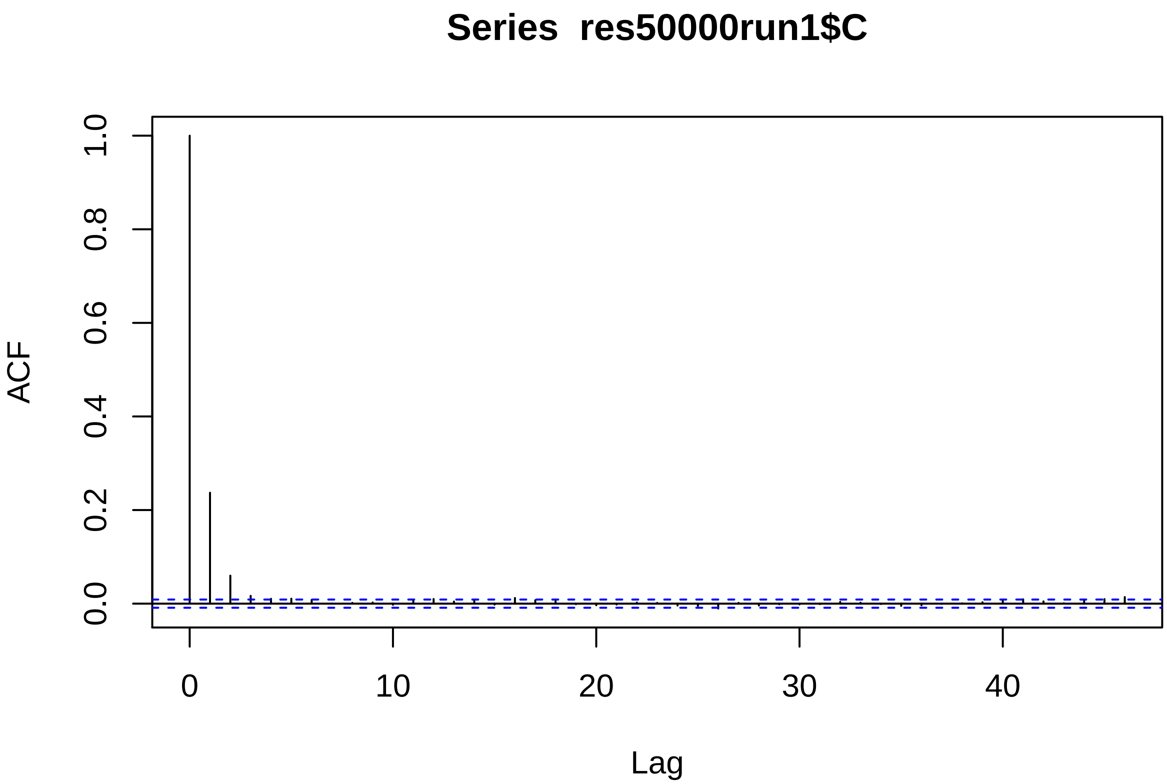
```r
gelman.plot(combinedchains)
```



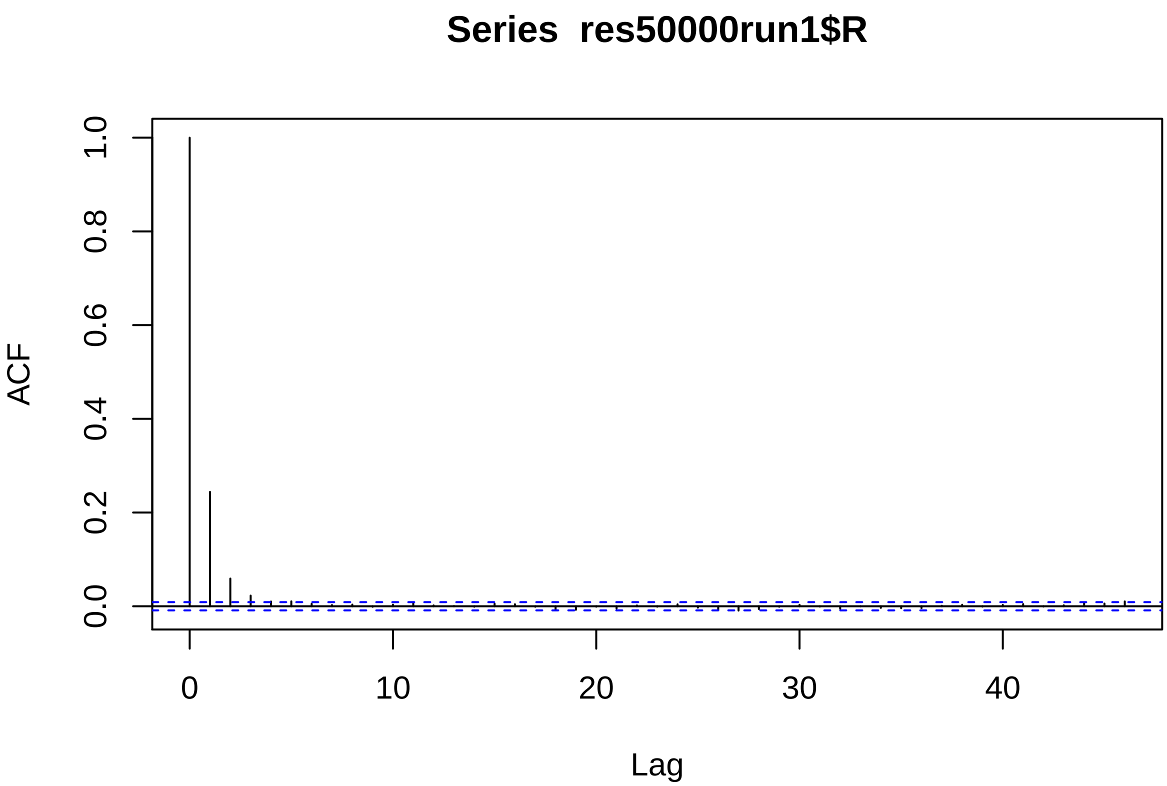
Suggested burn-in time based on this plot is about ~5000 iterations or also ~20000 iterations for R.

g) Provide plots for both variables Rain and Cloudy and suggest an interval for drawing approximately independent samples.

```r
acf(res50000run1$C,type="correlation")
```




```
acf(res50000run1$R,type="correlation")
```



The safe interval for taking independent samples is from 4 on.

h)Re-estimate P (R = T | S = T, W = T) based on samples obtained after the suggested burn-in time and thinning.

```
dd<-res50000run1[5000:50000,]





tabledd=table(dd)/niter
colSums(tabledd)
```

```
##       0       1
## 0.60794 0.29208
```

i) Compute the probability P (R = T | S = T, W = T) analytically. Compare with (c) and (h)

$$P(R=T|S=T,W=T) = \frac{P(R=T,S=T,W=T)}{P(S=T,W=T)}$$

$$= \frac{P(W=T|R=T,S=T)P(R=T|S=T)P(S=T)}{\sum_{c\in\{F,T\}} P(C=c)P(S=T,W=T|C=c)}$$

$$= \frac{P(W=T|R=T,S=T)P(R=T|S=T)P(S=T)}{\sum_{c\in\{F,T\}} P(C=c)P(W=T|S=T,C=c)P(S=T|C=c)}$$

Here we compute the quantities we dont know inmediatly from the table:

$$P(R=T|S=T) = \sum_{c\in\{F,T\}} P(C=c)P(R=T|S=T,C=c)$$

$$= \sum_{c\in\{F,T\}} P(C=c)P(R=T|C=c) = P(R=T) = 0.5\times 0.8 + 0.5\times 0.2 = 0.5$$

$$P(R=F) = 1 - P(R=T) = 0.5$$

$$P(S=T) = \sum_{c\in\{F,T\}} P(C=c)P(S=T|C=c) = 0.5\times 0.1 + 0.5\times 0.5 = 0.3$$

$$P(W=T|S=T,C=T) = \sum_{r\in\{F,T\}} P(R=r)P(W=T|S=T,R=r,C=T)$$

$$= \sum_{r\in\{F,T\}} P(R=r)P(W=T|S=T,R=r) = 0.5\times 0.99 + 0.5\times 0.9 = 0.945$$

Then putting it all together:

$$\frac{P(W=T|R=T,S=T)P(R=T|S=T)P(S=T)}{\sum_{c\in\{F,T\}} P(C=c)P(W=T|S=T)P(S=T|C=c)}$$

$$= \frac{0.99\times 0.5\times 0.3}{0.5\times 0.945\times 0.1 + 0.5\times 0.945\times 0.5} = 0.5238095$$