

Project 6

Contents

Problem 16	1
Gibbs sampler (b)	1
Marginal probability (c)	2
Auto-correlation and effective sample size (ESS) (d)	2
Gibbs sampling of 50000 samples (e)	4
Relative frequencies to detect burn-in phase (f)	5
Potential scale reduction factor (g)	6
Re-estimation of marginal probability (h)	6
Analytical solution of marginal probability (i)	7

Problem 16

Gibbs sampler (b)

```
# Encoding conditional probabilities
p <- list(C=0.5, S.C1=0.1, S.C0=0.5, R.C1=0.8, R.C0=0.2,
          W.S1_R1=0.99, W.S1_R0=0.9, W.S0_R1=0.9, W.S0_R0=0.01)
p$C.R1_S1_W1 <- p$R.C1*p$S.C1*p$C/(p$R.C1*p$S.C1*p$C+p$R.C0*p$S.C0*(1-p$C))
p$C.R0_S1_W1 <- (1-p$R.C1)*p$S.C1*p$C/((1-p$R.C1)*p$S.C1*p$C+(1-p$R.C0)*p$S.C0*(1-p$C))
p$R.C1_S1_W1 <- p$W.S1_R1*p$R.C1/(p$W.S1_R1*p$R.C1+p$W.S1_R0*(1-p$R.C1))
p$R.C0_S1_W1 <- p$W.S1_R1*p$R.C0/(p$W.S1_R1*p$R.C0+p$W.S1_R0*(1-p$R.C0))

Gibbs_sampler <- function(N=100) {
  samples <- matrix(NA, ncol=2, nrow=N, dimnames=list(NULL, c("R", "C")))
  # Initialisation (beginning state)
  samples[1,] <- c(TRUE, TRUE)

  for(n in 2:N) {
    # Random sampling to choose which variable (R or C) is updated
    indicator <- sample(c("R", "C"), size=1, prob=c(0.5, 0.5))
    if(indicator == "R") {
      # Sampling from  $P(R_{\{n+1\}}|C_{\{n\}}, S=T, W=T)$ 
      if(samples[(n-1), "C"]) sampling_prob <- p$R.C1_S1_W1 else sampling_prob <- p$R.C0_S1_W1

      sampled_R <- sample(c(TRUE, FALSE), size=1, prob=c(sampling_prob, (1-sampling_prob)))
      samples[n,] <- c(sampled_R, samples[(n-1), "C"])
    } else {
      # Sampling from  $P(C_{\{n+1\}}|R_{\{n\}}, S=T, W=T)$ 
      if(samples[(n-1), "R"]) sampling_prob <- p$C.R1_S1_W1 else sampling_prob <- p$C.R0_S1_W1

      sampled_C <- sample(c(TRUE, FALSE), size=1, prob=c(sampling_prob, (1-sampling_prob)))
      samples[n,] <- c(samples[(n-1), "R"], sampled_C)
    }
  }
}
```

```

    }

    return(samples)
}

set.seed(42)
N <- 100
samples <- Gibbs_sampler(N)

result <- table(samples[, "R"], samples[, "C"])/N
rownames(result) <- c("R0", "R1")
colnames(result) <- c("C0", "C1")
result

##
##          C0   C1
##   R0 0.59 0.05
##   R1 0.18 0.18

```

Marginal probability (c)

Computing marginal probability $P(R=T|S=T, W=T)$.

```

marginal.R.S1_W1 <- sum(result[2,])
marginal.R.S1_W1

```

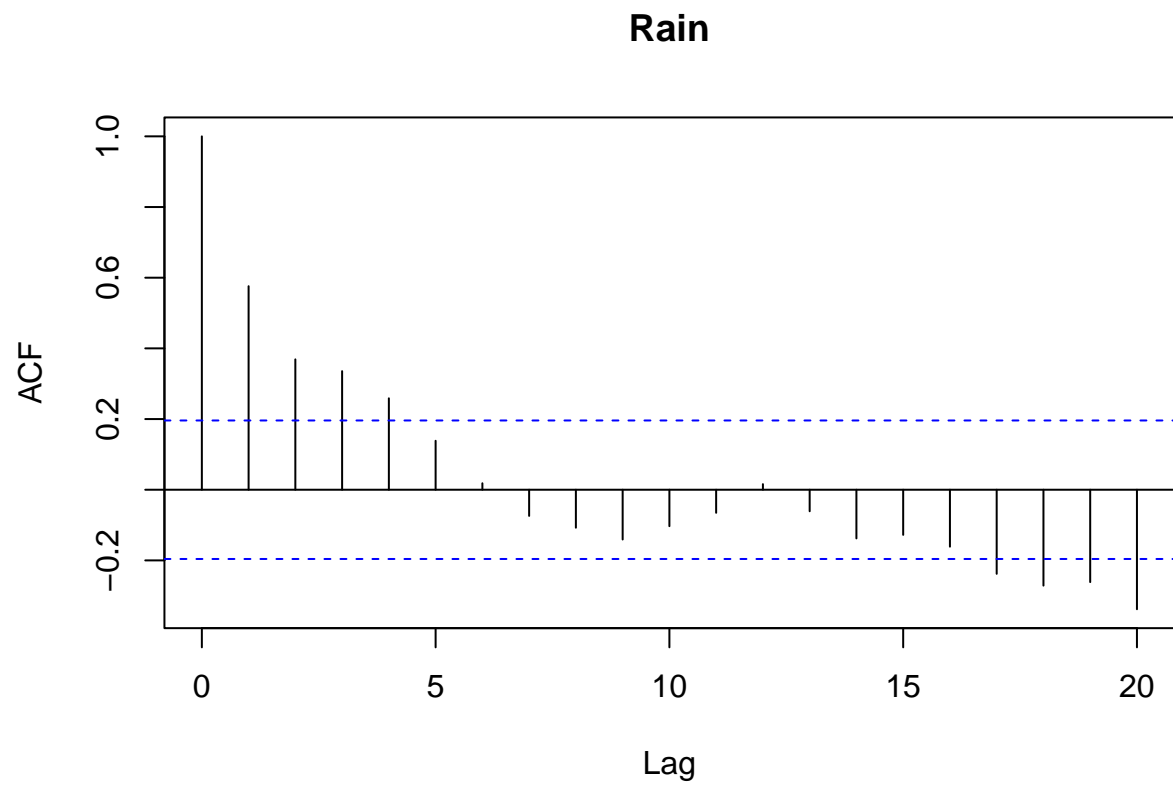
```
## [1] 0.36
```

Auto-correlation and effective sample size (ESS) (d)

```

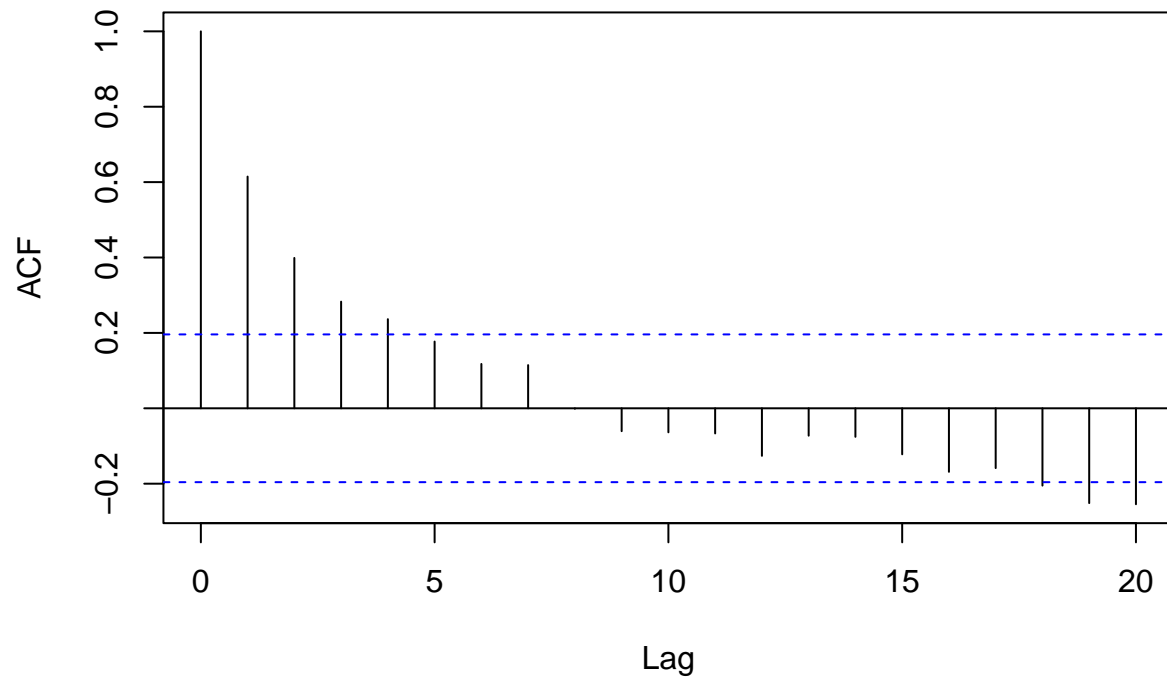
ro_R <- acf(as.numeric(samples[, "R"]), main="Rain")

```



```
ro_C <- acf(as.numeric(samples[, "C"]), main="Cloudy")
```

Cloudy



```
ESS_R <- ro_R$n.used/(1+2*sum(ro_R$acf))
ESS_C <- ro_C$n.used/(1+2*sum(ro_C$acf))
print(paste("Effective sample size for chain 'rain':", ESS_R))

## [1] "Effective sample size for chain 'rain': 44.4718962322423"
print(paste("Effective sample size for chain 'cloudy':", ESS_C))

## [1] "Effective sample size for chain 'cloudy': 27.5547672392332"
```

Gibbs sampling of 50000 samples (e)

```
set.seed(42)
N <- 50000
samples1 <- Gibbs_sampler(N)

result1 <- table(samples1[, "R"], samples1[, "C"])/N
rownames(result1) <- c("R0", "R1")
colnames(result1) <- c("C0", "C1")

samples2 <- Gibbs_sampler(N)

result2 <- table(samples2[, "R"], samples2[, "C"])/N
rownames(result2) <- c("R0", "R1")
colnames(result2) <- c("C0", "C1")
```

```
result1
```

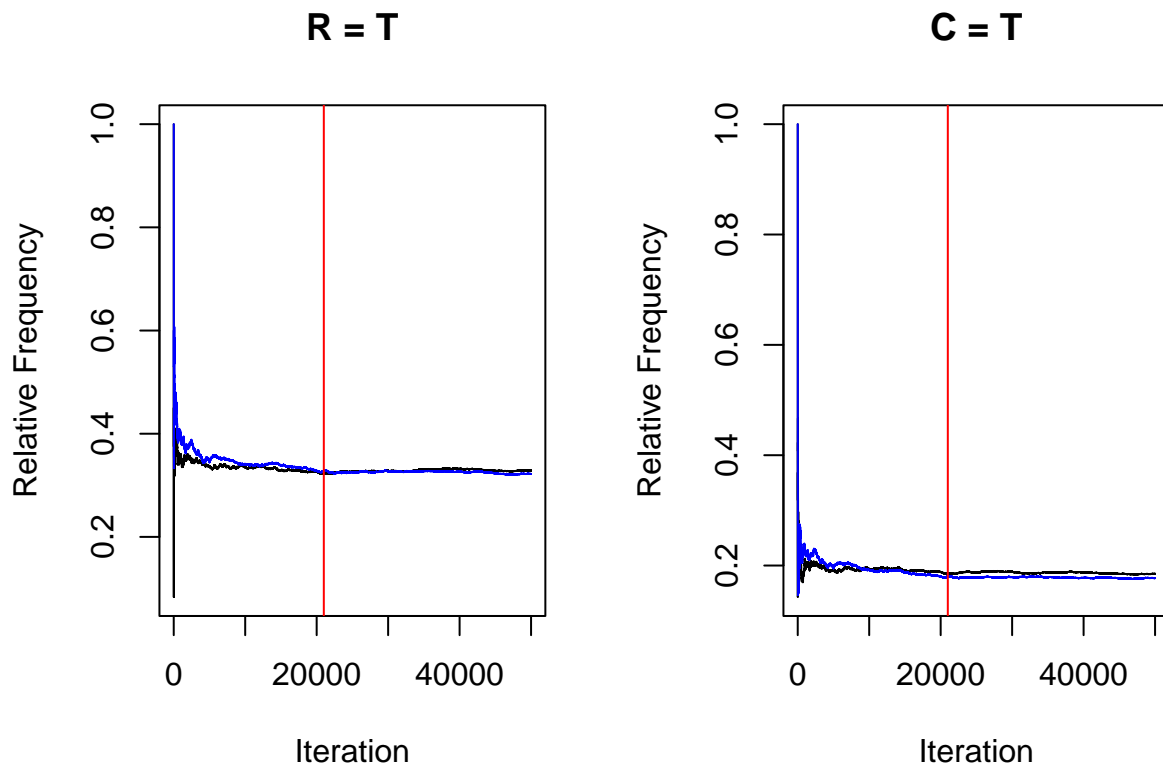
```
##
##           C0      C1
##    R0 0.63588 0.03570
##    R1 0.17904 0.14938
```

```
result2
```

```
##
##           C0      C1
##    R0 0.64450 0.03334
##    R1 0.17826 0.14390
```

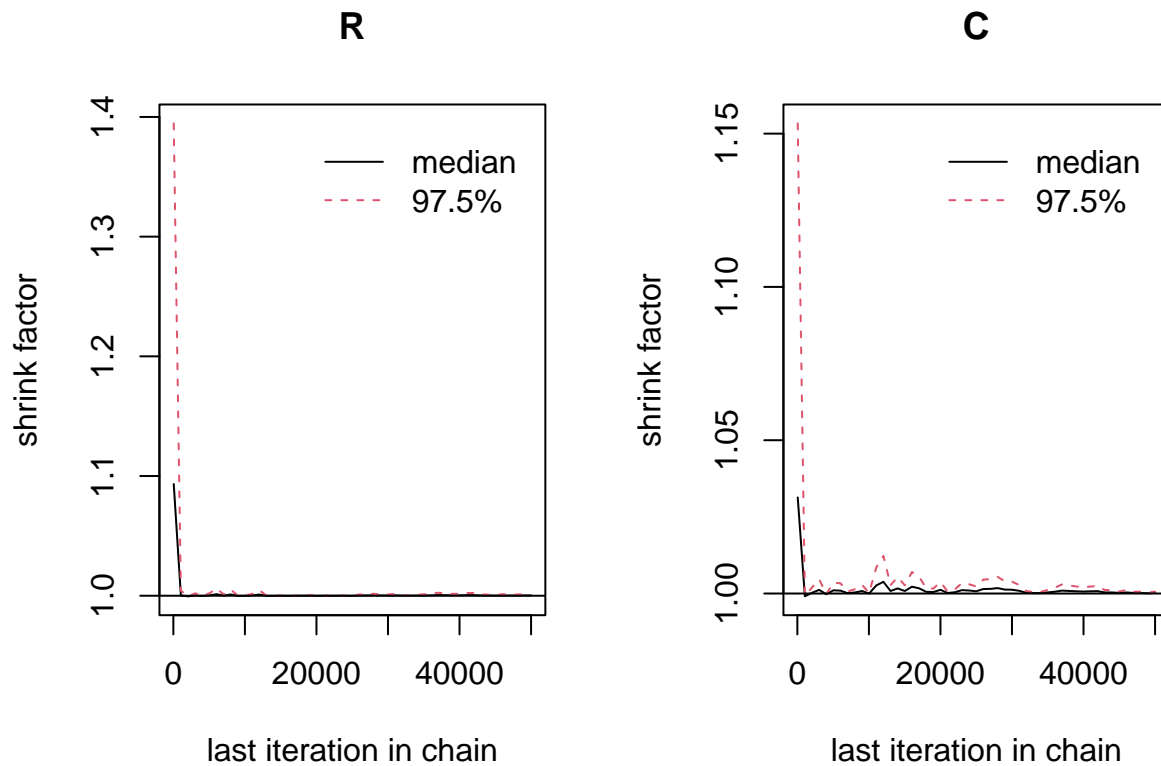
Relative frequencies to detect burn-in phase (f)

```
# Plotting relative frequencies of R = T and C = T
burn_in_phase <- 21000
par(mfrow=c(1,2))
plot(1:N, cumsum(samples1[, "R"])/seq_len(N), type="l",
     xlab="Iteration", ylab="Relative Frequency", main="R = T")
lines(cumsum(samples2[, "R"])/seq_len(N), type='l', col='blue')
abline(v=burn_in_phase, col="red")
plot(1:N, cumsum(samples1[, "C"])/seq_len(N), type="l",
     xlab="Iteration", ylab="Relative Frequency", main="C = T")
lines(cumsum(samples2[, "C"])/seq_len(N), type='l', col='blue')
abline(v=burn_in_phase, col="red")
```



Potential scale reduction factor (g)

```
library(coda)
mcmc$ <- mcmc.list(mcmc(data=samples1, start=2, end=N),
                  mcmc(data=samples2, start=2, end=N))
gelman.plot(mcmc$)
```



```
gelman.diag(mcmc$)
```

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

After around 20000 iterations, the fluctuations around the potential scale reduction factor are negligible, therefore a suggested burn-in time is 20000.

Re-estimation of marginal probability (h)

Computing marginal probability $P(R=T|S=T,W=T)$ excluding the samples from the burn-in phase.

```
burn_in_phase <- 10000
result1 <- table(samples1[(burn_in_phase+1):N, "R"],
```

```

    samples1[(burn_in_phase+1):N, "C"])/(N-burn_in_phase)
rownames(result1) <- c("R0", "R1")
colnames(result1) <- c("C0", "C1")
result1

##
##           C0           C1
##  R0 0.637625 0.035600
##  R1 0.179275 0.147500

result2 <- table(samples2[(burn_in_phase+1):N, "R"],
    samples2[(burn_in_phase+1):N, "C"])/(N-burn_in_phase)
rownames(result2) <- c("R0", "R1")
colnames(result2) <- c("C0", "C1")
result2

##
##           C0           C1
##  R0 0.649850 0.032700
##  R1 0.176575 0.140875

marginal1.R.S1_W1 <- sum(result1[2,])
marginal2.R.S1_W1 <- sum(result2[2,])
print("Marginal probability using the first run of 50000 iterations and excluded burn-in phase:")

## [1] "Marginal probability using the first run of 50000 iterations and excluded burn-in phase:"
print(marginal1.R.S1_W1)

## [1] 0.326775
print("Marginal probability using the second run of 50000 iterations and excluded burn-in phase:")

## [1] "Marginal probability using the second run of 50000 iterations and excluded burn-in phase:"
print(marginal2.R.S1_W1)

## [1] 0.31745

```

Analytical solution of marginal probability (i)

```

S1_W1 <- p$W.S1_R0*p$S.CO*(1-p$R.CO)*(1-p$C) +
    p$W.S1_R0*p$S.C1*(1-p$R.C1)*p$C +
    p$W.S1_R1*p$S.CO*p$R.CO*(1-p$C) +
    p$W.S1_R1*p$S.C1*p$R.C1*p$C

R1_S1_W1 <- p$W.S1_R1*p$S.CO*p$R.CO*(1-p$C) +
    p$W.S1_R1*p$S.C1*p$R.C1*p$C

analytical_marginal.R.S1_W1 <- R1_S1_W1 / S1_W1

print("Analytical solution for the marginal probability:")

## [1] "Analytical solution for the marginal probability:"
print(analytical_marginal.R.S1_W1)

## [1] 0.3203883

```

The marginal probabilities of the (h) part are closer to the analytical solution, meanwhile probability of (c) part deviates more. It is intuitive, since the sample size in (c) was small (100).

```
library(rmarkdown)
render("Gibbs.Rmd", pdf_document(TRUE), "Indilewitsch_Toidze_Houhamdi_Pudziuvelyte_Project6.pdf")
```