

Experimenting with Markov chain

Markov chain sampling

The purpose of this exercise is to study the properties of Markov chains and how they can be used to produce samples for Monte Carlo estimation.

Consider a Markov chain defined as follows:

- set $\theta^{(0)} = C$, where C is some constant number.
- for $i = 1, \dots$ sample $\theta^{(i)} \sim N(\phi\theta^{(i-1)}, \sigma^2)$ where $\phi \in [0, 1)$ is a parameter controlling the autocorrelation between samples.

Note! This is a Markov chain that is constructed very differently from how Stan constructs the Markov chains to sample from the posterior distributions. However, the properties related to autocorrelation and initial value are analogous.

Result for task 1: the limits

As given in the statement:

As $i \rightarrow \infty$

$$\text{Var}[\theta^{(i)}] = \frac{\sigma^2}{1-\phi^2}$$

$$\text{Corr}([\theta^{(i)}, \theta^{(i+t)}]) = \phi^t$$

Therefore

If $\phi \rightarrow 0$:

$$\text{Var}[\theta^{(i)}] = \frac{\sigma^2}{1-0^2} = \sigma^2$$

$$\text{Corr}([\theta^{(i)}, \theta^{(i+t)}]) = 0^t = 0$$

If $\phi \rightarrow 1$ and since $\phi \in [0, 1)$ we replace ϕ with 1^- :

$$\text{Var}[\theta^{(i)}] = \frac{\sigma^2}{1-1^-^2} = \frac{\sigma^2}{0^+} = +\infty$$

Result for task 2: the table

Given the marginal variance for $\theta^{(i)}$ we can solve for ϕ and σ^2 when the other is given

$\text{Var}[\theta^{(i)}]$	ϕ	σ^2	$\text{Corr}[\theta^{(i)}, \theta^{(i+1)}]$
1	0	1	0
1	0.5	0.75	0.5
1	0.8944272	0.2	0.8944272
1	0.1	0.99	0.1

Result for task 3

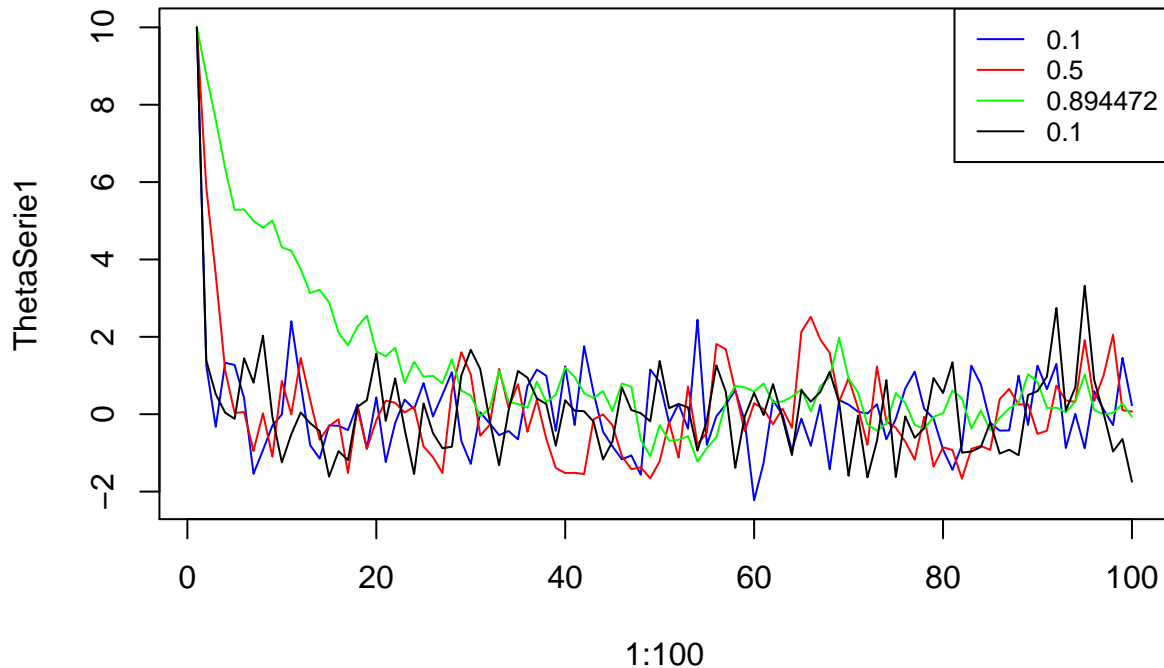
Implement the above Markov chain with R and use it to sample random realizations of $\theta^{(i)}$ where $i = 1, \dots, 100$ with the parameter values given in the above table. As an initial value use $C = 10$. Plot the sample chain and based on the visual inspection, what can you say about the convergence and mixing properties of the chain with the different choices of ϕ ?

```
M = 100

# let's first define a function to conduct the sampling
Sampling = function(initial_value,Phi,Sigma2,m) {
  C=initial_value
  Theta = rep(0,100)
  Theta[1] = C
  for(i in 2:m)
    Theta[i] = rnorm(1,Phi * Theta[i-1] , sqrt(Sigma2))
  return(Theta)
}

# Then we sample from the Markov chain with alternative phi and sigma values and draw them
ThetaSerie1 = Sampling (10,0,1,M)
ThetaSerie2 = Sampling (10,0.5,0.75,M )
ThetaSerie3 = Sampling (10,0.894472,0.2,M )
ThetaSerie4 = Sampling (10,0.1,0.99,M )

plot(1:100,ThetaSerie1,type='l',col="blue")
legend("topright", legend = c("0.1","0.5","0.894472","0.1"),col = c("blue","red","green","black"),lty=1)
lines(1:100,ThetaSerie2,col="red")
lines(1:100,ThetaSerie3,col="green")
lines(1:100,ThetaSerie4,col="black")
```



We can see that around the iteration 20 (this could change when I knit the file but you got the idea), the plots converge and oscillates around 0. As we can see, the convergence is attended faster when phi is smaller. the model with $\phi = 0.89$ takes the most time in order to converge.

Result for task 4

Choose the parameter combination where $\sigma^2 = 0.2$ from the above table. Run three Markov chains with initial values $C_1 = 10$, $C_2 = -10$ and $C_3 = 5$. Find a burn-in value at which the chains have converged according to the PSRF (\hat{R}) statistics. This is implemented in function `Rhat` in RStan. Note, $m = 100$ samples might not be enough here.

```
# Define the sample size
m = 100

# sample from three independent Markov chains as instructed in the exercise
theta1 = Sampling (10,0.894472,0.2,m )
theta2 = Sampling (-10,0.894472,0.2,m )
theta3 = Sampling (5,0.894472,0.2,m )

# examine the chains visually
plot(theta1, type="l", xlab="iteration", ylab=expression(theta), main=sprintf("sigma2=%.1f, phi=%.2f", 0.2, 0.894472),
lines(theta2, col="blue")
lines(theta3, col="red")

# Check visually where the sample chains seem to have converged to the same stationary
#Convergence ~ 20
```

```

# distribution
library(rstan)                                # First, load the necessary tool - that is Rhat function

## Loading required package: StanHeaders

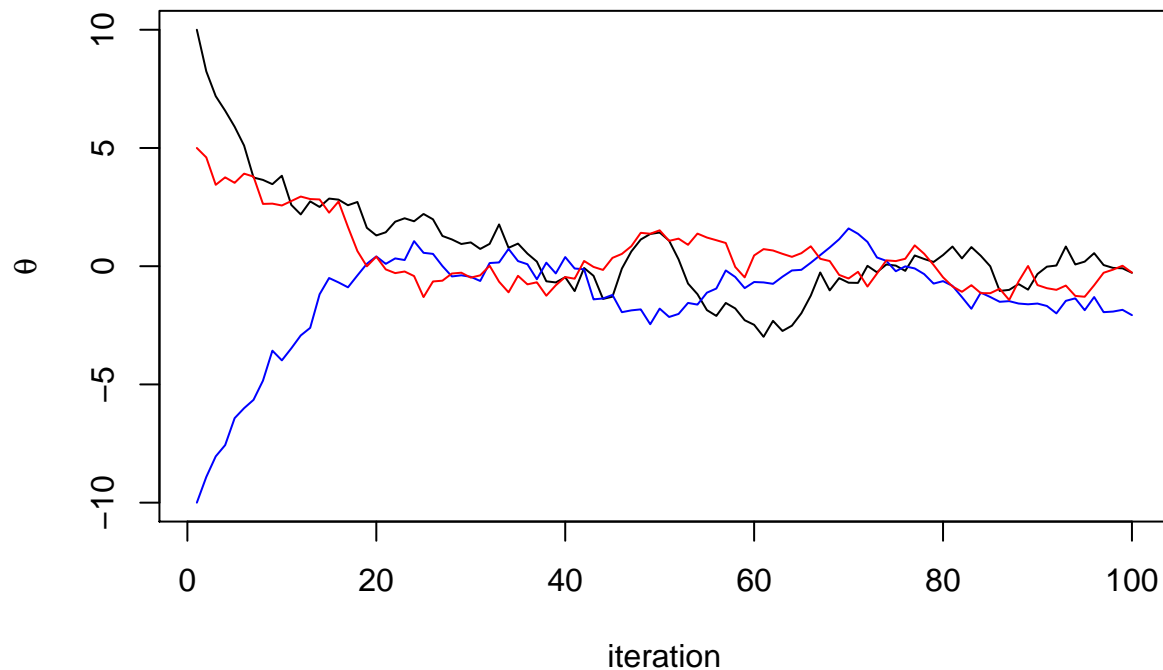
## Loading required package: ggplot2

## rstan (Version 2.19.3, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)

```

sigma2=0.2, phi=0.89



```

# ?Rhat
THETA = rbind(theta1,theta2,theta3) # Put the chains into list where each row is one Markov chain
paste(Rhat(THETA))

## [1] "1.019791227403"

```

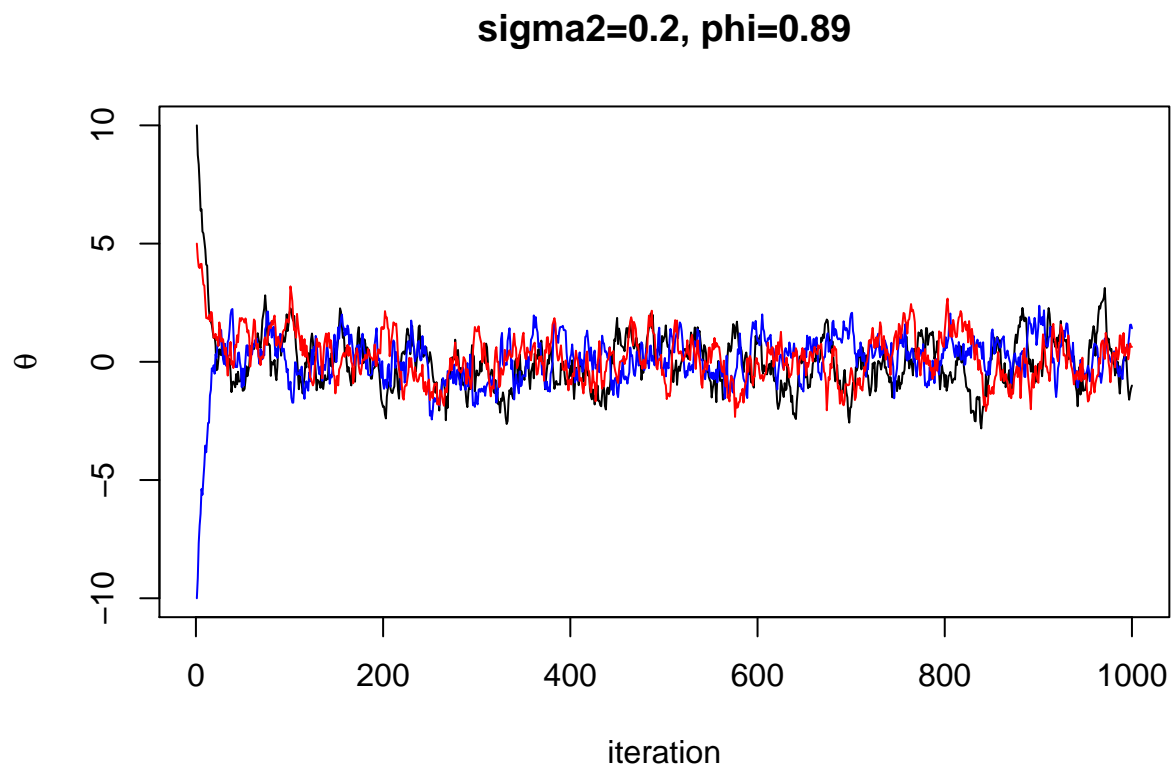
```
# if Rhat value is greater than 1.05 try to remove more samples from the beginning
# However, note that you may need to increase the sample size in order to get reliable estimate for PSR.
# Note also that the Rhat values change each time you rerun the Markov chains. Hence, even if with one
# realization the PSRF looked fine it might not be so in another even with same number of samples.
```

Visually the Convergence is around 20 (May change after knitting the document therefore the answer isn't very precise) Seen that Rhat value is less than 1.05 the model converges. However, after running the code many times, sometimes I get a $\text{Psrf} > 1.05$. Therefore, I will try again with a sample of 1000 iterations so I don't get this issue anymore.

```
# Define the sample size
m = 1000

# sample from three independent Markov chains as instructed in the exercise
theta1 = Sampling (10,0.894472,0.2,m )
theta2 = Sampling (-10,0.894472,0.2,m )
theta3 = Sampling (5,0.894472,0.2,m )

# examine the chains visually
plot(theta1, type="l", xlab="iteration", ylab=expression(theta), main=sprintf("sigma2=%.1f, phi=%.2f", 0.2, 0.89),
lines(theta2, col="blue")
lines(theta3, col="red")
```



```

# Check visually where the sample chains seem to have converged to the same stationary
#Convergence ~ 20

# distribution

library(rstan) # First, load the necessary tool - that is Rhat function
# ?Rhat
THETA = rbind(theta1,theta2,theta3) # Put the chains into list where each row is one Markov chain
paste(Rhat(THETA))

```

```
## [1] "1.01292131581124"
```

```

# if Rhat value is greater than 1.05 try to remove more samples from the beginning
# However, note that you may need to increase the sample size in order to get reliable estimate for PSR
# Note also that the Rhat values change each time you rerun the Markov chains. Hence, even if with one
# realization the PSRF looked fine it might not be so in another even with same number of samples.

```

In this case the coefficient is always < 1.05 and we always have a good convergence

Burn in value ?

```

for ( i in 2:m){
  THETA = rbind(theta1[1:i],theta2[1:i],theta3[1:i]) # Put the chains into list where each row is one
  r = Rhat(THETA)

  if (r < 1.05) {
    print("The burn in value is: ")
    print(i)
    print(". In this case Rhat =")
    print(r)
    break
  }
}

```

```

## [1] "The burn in value is: "
## [1] 33
## [1] ". In this case Rhat ="
## [1] 1.040816

```