HW14

```
rm(list = ls())
P <- t(matrix(c(0.180,0.274,0.426,0.120,
                0.171,0.367,0.274,0.188,
                0.161,0.339,0.375,0.125,
                0.079,0.355,0.384,0.182),nrow=4, ncol=4))
```

1. Compute P^5, P^50 and P^500

```
library(expm)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'expm'
```

```
## The following object is masked from 'package:Matrix':
##
##     expm
```

```
P%^%5
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.1546840 0.3409589 0.3498497 0.1545074
## [2,] 0.1546760 0.3409679 0.3498378 0.1545183
## [3,] 0.1546805 0.3409631 0.3498445 0.1545119
## [4,] 0.1546730 0.3409700 0.3498363 0.1545207
```

```
P%^%50
```

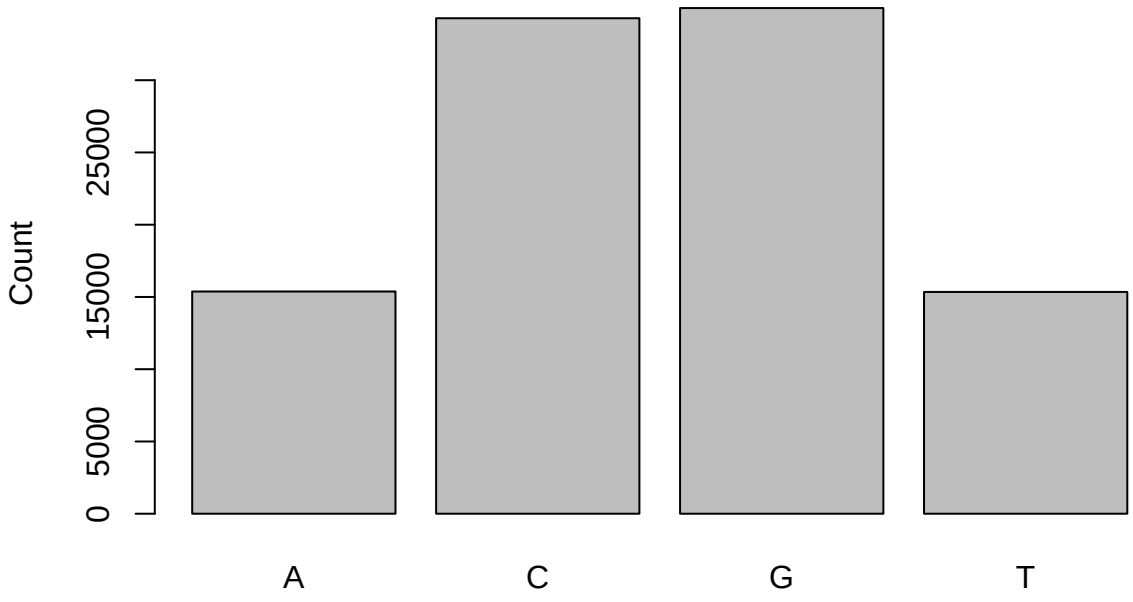
```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.1546783 0.3409652 0.3498417 0.1545148
## [2,] 0.1546783 0.3409652 0.3498417 0.1545148
## [3,] 0.1546783 0.3409652 0.3498417 0.1545148
## [4,] 0.1546783 0.3409652 0.3498417 0.1545148
```

```
P%^%500
```

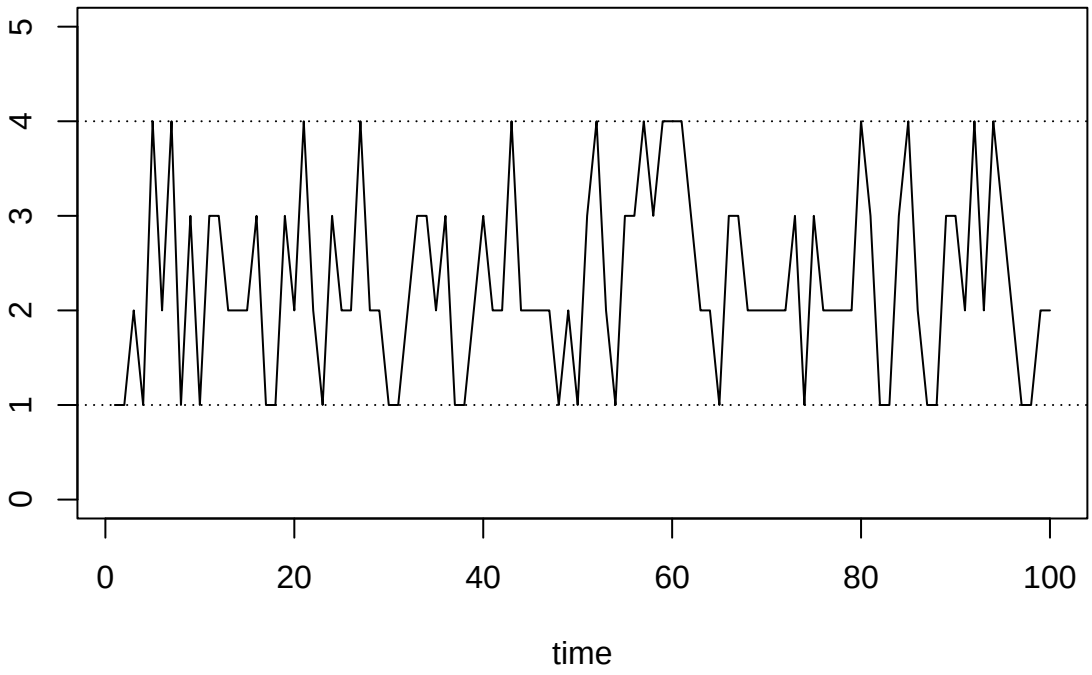
```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 0.1546783 0.3409652 0.3498417 0.1545148
## [2,] 0.1546783 0.3409652 0.3498417 0.1545148
## [3,] 0.1546783 0.3409652 0.3498417 0.1545148
## [4,] 0.1546783 0.3409652 0.3498417 0.1545148
```

2. Simulate one sequence of 100,000 nucleotides from this 4-state Markov chain and then visualize this sequence. Use set.seed(440) in your simulations

```
run.mc.sim <- function(P, num.iters) {
  set.seed(440)
  # number of possible states
  num.states <- nrow(P)
  # stores the states X_t through time
  states <- numeric(num.iters)
  # initialize variable for first state
  states[1] <- 1
  for(t in 2:num.iters) {
    # probability vector to simulate next state X_{t+1}
    p <- P[states[t-1], ]
    # draw from multinomial and determine state
    states[t] <- which(rmultinom(1, 1, p) == 1)
  }
  return(states)
}
num.iterations <- 100000
chain.states <- run.mc.sim(P, num.iterations)
counts <- table(chain.states)
barplot(counts,
        names.arg=c("A", "C", "G", "T"),
        ylab="Count")
```




```
matplot(chain.states[99900:99999], type='l', lty=1, col=1:5, ylim=c(0,5), ylab='state
', xlab='time')
abline(h=1, lty=3)
abline(h=4, lty=3)
```



3. Based on the sequence of 100,000 nucleotides simulated from the previous part,compute the proportions of four types of nucleotides in this sequence.

```
proportions <- counts/100000
proportions
```

```
## chain.states
##       1       2       3       4
## 0.15376 0.34285 0.34991 0.15348
```

4. Based on the sequence of 100,000 nucleotides simulated above, compute the the proportion of CG occurrence in this sequence. For example, if your simulated sequence is GCTAATCCGGCGAAT, then the number of CG occurrence in this sequence is 2 and the proportion of CG occurrence is 2/(15-1)=1/7

```
string.chain.states <- toString(chain.states)
library(stringr)
str_count(string.chain.states, pattern = "2, 3")/100000
```

```
## [1] 0.09528
```

5. Now use the one-step transition probability matrix P to compute the expected proportion of CG occurrence simulated from this 4-state Markov chain

```
P[2,3] * (P%^%500)[1,2]
```

```
## [1] 0.09342446
```

Consider a new sequence of i.i.d. nucleotides with C and G proportions matching your simulated sequence above. What is the expected proportion of CG occurrence in this new sequence?

```
P[2,3] * proportions[2]
```

```
##         2
## 0.0939409
```