MSiA400 Lab1

Question 1

```
### Read-in the table
webtraffic <- read.delim("webtraffic.txt")
```

Part 1a

```
Traffic <- matrix(colSums(webtraffic), nrow = 9, ncol = 9, byrow = TRUE)
Traffic[9,1]=1000
Traffic
```

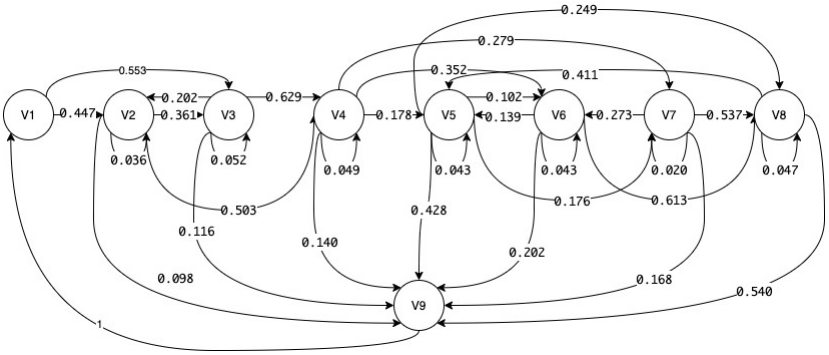
```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
##  [1,]    0  447  553    0    0    0    0    0    0
##  [2,]    0   23  230  321    0    0    0    0   63
##  [3,]    0  167   43  520    0    0    0    0   96
##  [4,]    0    0    0   44  158  312  247    0  124
##  [5,]    0    0    0    0   22   52   90  127  218
##  [6,]    0    0    0    0   67   21    0  294   97
##  [7,]    0    0    0    0    0   94    7  185   58
##  [8,]    0    0    0    0  262    0    0   30  344
##  [9,] 1000    0    0    0    0    0    0    0    0
```

The Traffic matrix is shown as above.

Part 1b

Graph:

```
include_graphics("MCMC.jpg")
```



This markov chain is **irreducible** as all states communicate with each other.

Consider this loop route:

V1 -> V2 -> V3 -> V4 -> V5 -> V6 -> V8 -> V5 -> V7 -> V9 -> V1

As all states were visited at least once and this route is a loop, we can conclude that all states communicate with each other therefore irreducible.

This markov chain is also **ergodic** as it is **recurrent** and **aperiodic**.

Recurrent:

As all states connect with each other, this markov chain is recurrent.

Aperiodic:

State 2-8 are aperiodic because they communicate with themselves within in one step.

State 1 is also aperiodic. Consider these two route below:

1: V1 -> V2 -> V9 -> V1 (step count = 4)

2: V1 -> V2 -> V3 -> V9 -> V1 (step count = 5)

gcd(4,5) = 1

Therefore, state 1 is aperiodic.

State 9 is aperiodic. Consider these two route below:

1: V9 -> V1 -> V2 -> V9 (step count = 4)

2: V9 -> V1 -> V2 -> V3 -> V9 (step count = 5)

gcd(4,5) = 1

Therefore, state 1 is aperiodic.

Part 1c

```
P <- matrix(NA, nrow = 9, ncol =9)
Traffic_rowsum = rowSums(Traffic)

for(i in 1:nrow(P)){
  for (j in 1:ncol(P)){
    P[i,j] = Traffic[i,j]/ Traffic_rowsum[i]
  }
}
P
```

```
##       [,1]       [,2]       [,3]       [,4]      [,5]       [,6]       [,7]
##  [1,]    0 0.44700000 0.55300000 0.00000000 0.0000000 0.00000000 0.00000000
##  [2,]    0 0.03610675 0.36106750 0.50392465 0.0000000 0.00000000 0.00000000
##  [3,]    0 0.20217918 0.05205811 0.62953995 0.0000000 0.00000000 0.00000000
##  [4,]    0 0.00000000 0.00000000 0.04971751 0.1785311 0.35254237 0.27909605
##  [5,]    0 0.00000000 0.00000000 0.00000000 0.0432220 0.10216110 0.17681729
##  [6,]    0 0.00000000 0.00000000 0.00000000 0.1398747 0.04384134 0.00000000
##  [7,]    0 0.00000000 0.00000000 0.00000000 0.0000000 0.27325581 0.02034884
##  [8,]    0 0.00000000 0.00000000 0.00000000 0.4119497 0.00000000 0.00000000
##  [9,]    1 0.00000000 0.00000000 0.00000000 0.0000000 0.00000000 0.00000000
##             [,8]      [,9]
##  [1,] 0.00000000 0.0000000
##  [2,] 0.00000000 0.0989011
##  [3,] 0.00000000 0.1162228
##  [4,] 0.00000000 0.1401130
##  [5,] 0.24950884 0.4282908
##  [6,] 0.61377871 0.2025052
##  [7,] 0.53779070 0.1686047
##  [8,] 0.04716981 0.5408805
##  [9,] 0.00000000 0.0000000
```

Part 1d

```
start = c(1,rep(0,8))
prob5 = start %*% P %*% P %*% P %*% P %*% P
prob5[5]
```

```
## [1] 0.1315178
```

The probability of a vistor being on Page 5 after 5 clicks is 0.1315178.

Part 1e

```
# make the network irreducible
Q = t(P) - diag(9)
Q[9,] = rep(1,9)
rhs = c(rep(0,8),1)

Pi = solve(Q,rhs)
Pi
```

```
## [1] 0.15832806 0.10085497 0.13077897 0.14012033 0.08058898 0.07583914 0.05446485
## [8] 0.10069664 0.15832806
```

The steady-state matrix vector is 0.1583281, 0.100855, 0.130779, 0.1401203, 0.080589, 0.0758391, 0.0544649, 0.1006966, 0.1583281.

Part 1f

```
B = P[1:8,1:8]
Q = diag(8)-B
rhs = c(0.1, 2, 3, 5, 5, 3, 3,  2)
m = solve(Q,rhs)
m[1]
```

```
## [1] 14.563
```

Average time a visitor spends on the website is 14.563 seconds.

Question 2

Part a

Determine the number of samples required to achieve an error tolerance of 10^{-3} with 99% confidence.

$$n \geq \frac{\frac{1}{\lambda^2}}{(10^{-3})^2 0.01}$$

$$n \geq \frac{10^8}{\lambda^2}$$

part b

lambda = 1

```
## lambda = 1
set.seed(1009)
lambda = 1
n = 10^8
x = runif(n,0,1)
y = -ln(x)
g = sin(y)
result_1 = sum(g)/n
result_1_compare = 1 / (1+lambda^2)
```

The result using MCMC is 0.5000296. The result using the true value is 0.5. The difference is 2.957232^{-5} which is smaller than the tolerance

lambda = 2

```
## lambda = 2
lambda = 2
n = (10^8)/(lambda^2)
x = runif(n,0,1)
y = (-ln(x))/lambda
g = (sin(y))/lambda
result_2 = sum(g)/n
result_2_compare = 1 / (1+lambda^2)
```

The result using MCMC is 0.199968. The result using the true value is 0.2. The difference is -3.2030119^{-5} which is smaller than the tolerance

lambda = 4

```
## lambda = 2
lambda = 4
n = (10^8)/(lambda^2)
x = runif(n,0,1)
y = (-ln(x))/lambda
g = (sin(y))/lambda
result_4 = sum(g)/n
result_4_compare = 1 / (1+lambda^2)
```

The result using MCMC is 0.0588133. The result using the true value is 0.0588235. The difference is -1.0195265^{-5} which is smaller than the tolerance

Question 3

part a

The exponential distribution is not symmetric. Therefore, we can not use Metropolis Algorithm.

We can not use gibbs sampling because it is for joint distribution.

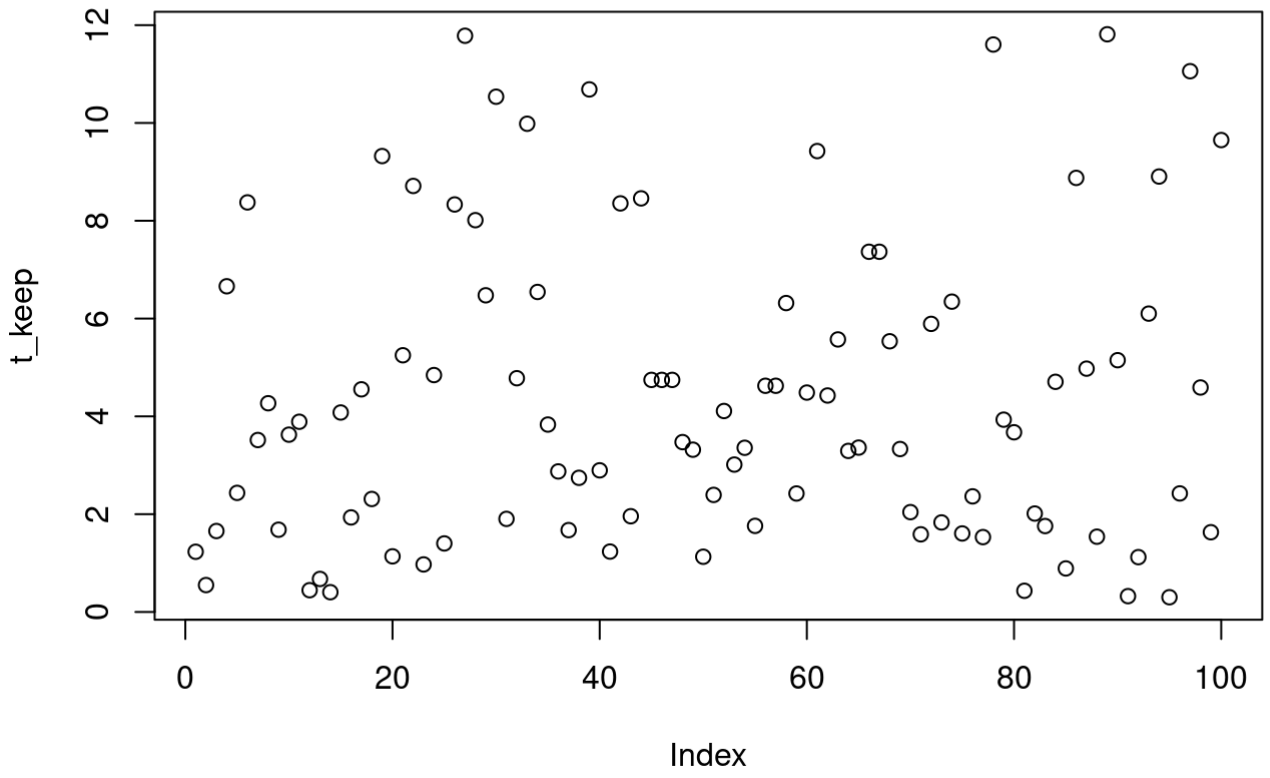
part b

```
x = 1 ### x_t
n = 15000
t = rep(NA, n)

for (i in 1:n){
  candidate = rexp(n = 1, rate = x)
  alpha_top = dexp(x, rate = candidate) * dgamma(candidate, shape = 2, scale = 2)
  alpha_bottom = dgamma(x, shape = 2, scale = 2) * dexp(candidate, rate = x)
  alpha = alpha_top/alpha_bottom

  u = runif(1, min = 0, max = 1)

  if (u <= alpha){
    x = candidate
  }
  t[i] = x
}
```

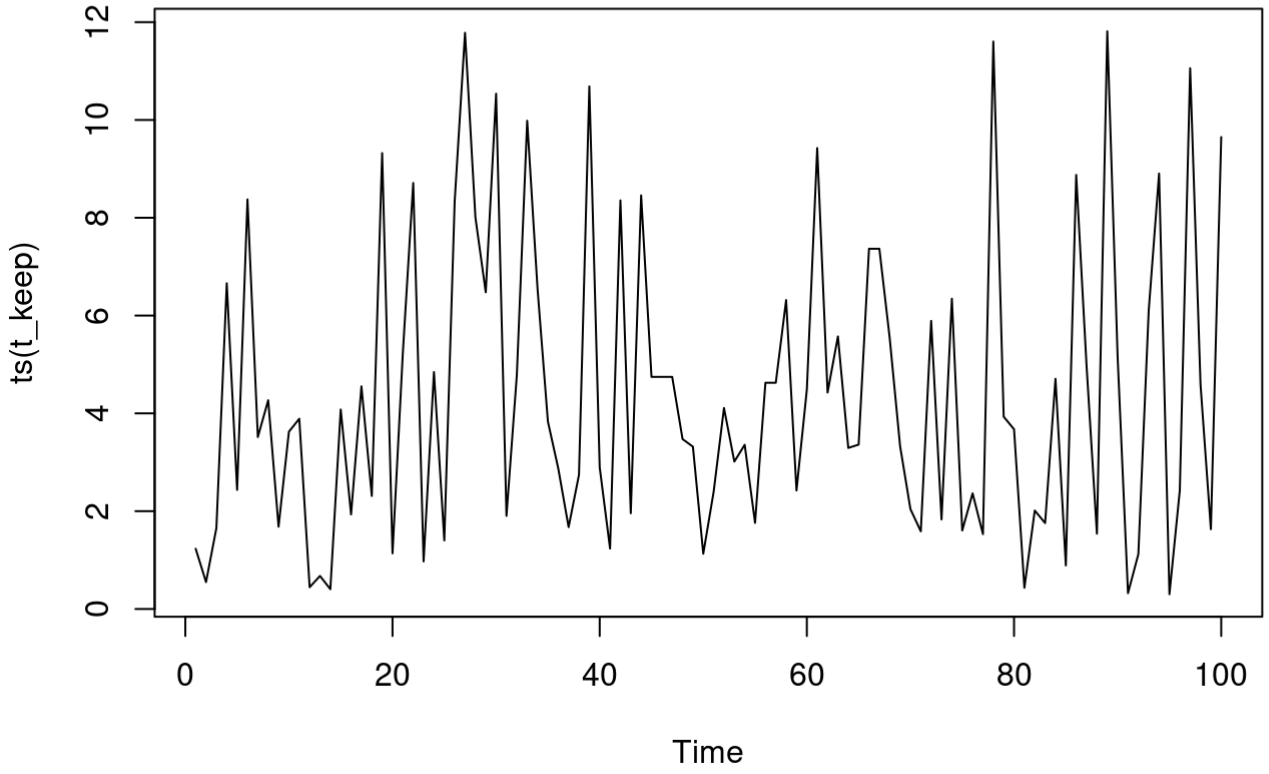
```
#how to consider what to keep
t_remove_burnin = t[5001 : n]
#t_keep = rep(NA, 100)
#for (i in 1:100){
 # t_keep[i] = t_remove_burnin[100*(i-1) + 1]
#}
t_keep <- t_remove_burnin[seq(1, 9901, 100)]
```

part c

```
plot(t_keep)
```




```
plot(ts(t_keep))
```



First, I plot the samples we kept over their index. We see that in the plot, the dots randomly place. Therefore, this process is random.

I also plot the samples we got over time. The plot is also completely random. Hence, I conclude that this process is random.