Assignment 1

Group 1

November 22th, 2020

Importing libraries

```
library(markovchain)
library(matlib)
```

Functions to solve the problems

```
matrixpower <- function(M,k) {
  if(dim(M)[1]!=dim(M)[2]) return(print("Error: matrix M is not square"))
  if (k == 0) return(diag(dim(M)[1]))
  if (k == 1) return(M)
  if (k > 1)  return(M %*% matrixpower(M, k-1))
}
```

Problem 1

a)

Markov chain criteria:

1- The probability of being in a state only depends on the previous state.

2- It's a stochastic process.

X = The chain hits state j at time n

X_n is the scenario at time n

All states have finite expected return times and are communicated with each other, also the MC is irreducible, therefore its stationary distribution is **unique**.

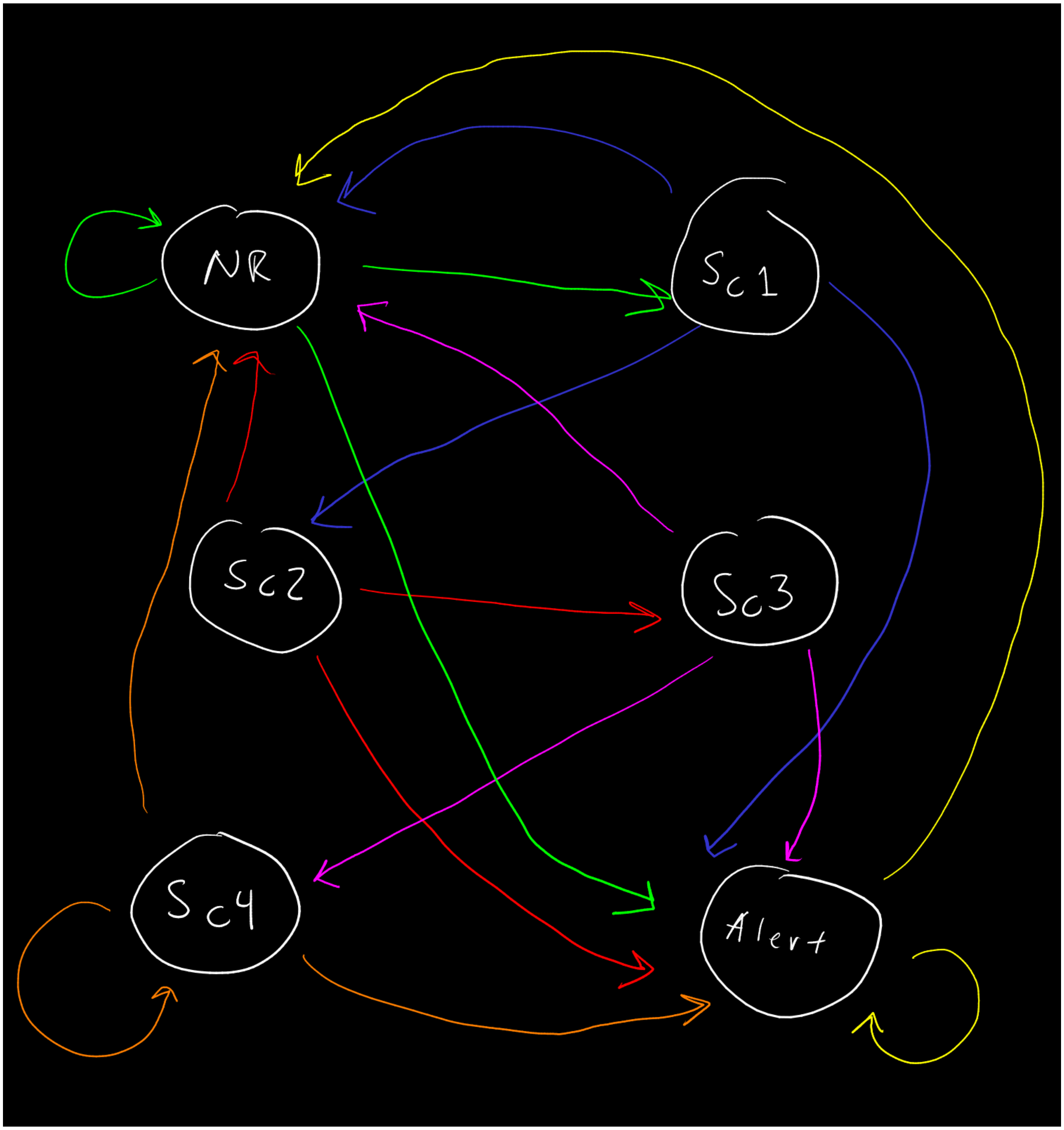

Figure 1: Graph for prob.1

b)

We have first calculated the relative frequencies manually.

```r
load('PollutionMadrid.RData')
data <-  X[1,]
mat <- matrix(rep(0,36), nrow=6, byrow=T)
for (i in 1:length(data)) {
  if (data[i] == "Alert") {
    data[i] = 1
  } else if (data[i] == "NR") {
    data[i] = 2
  } else if (data[i] == "Sc1") {
    data[i] = 3
  } else if (data[i] == "Sc2") {
    data[i] = 4
  } else if (data[i] == "Sc3") {
    data[i] = 5
  } else if (data[i] == "Sc4") {
    data[i] = 6
  }
}
data <- as.numeric(data)
for (i in 1:length(data)) {
  mat[data[i],data[i+1]] = mat[data[i],data[i+1]] + 1
}
mat[data[1460],data[1]] = mat[data[1460],data[1]] + 1

tbl <- table(data)
for (i in 1:length(tbl)) {
  mat[i,] = mat[i,]/tbl[i]
}
mat
```

```
##            [,1]      [,2]       [,3]      [,4]      [,5]      [,6]
## [1,] 0.00000000 1.0000000 0.00000000 0.0000000 0.0000000 0.0000000
## [2,] 0.00000000 0.9529851 0.04701493 0.0000000 0.0000000 0.0000000
## [3,] 0.01587302 0.4920635 0.00000000 0.4920635 0.0000000 0.0000000
## [4,] 0.00000000 0.5806452 0.00000000 0.0000000 0.4193548 0.0000000
## [5,] 0.23076923 0.3846154 0.00000000 0.0000000 0.0000000 0.3846154
## [6,] 0.00000000 0.5555556 0.00000000 0.0000000 0.0000000 0.4444444
```

We then tested using the *markovchain* package in order to confirm our results.

```
data <- X[1,]
markovchainFit(data)$estimate
```

```
## MLE Fit
##  A  6 - dimensional discrete Markov Chain defined by the following states:
##  Alert, NR, Sc1, Sc2, Sc3, Sc4
##  The transition matrix  (by rows)  is defined as follows:
##            Alert        NR        Sc1 Sc2       Sc3       Sc4
## Alert 0.00000000 1.0000000 0.00000000 0.0 0.0000000 0.0000000
## NR    0.00000000 0.9529851 0.04701493 0.0 0.0000000 0.0000000
## Sc1   0.01612903 0.4838710 0.00000000 0.5 0.0000000 0.0000000
## Sc2   0.00000000 0.5806452 0.00000000 0.0 0.4193548 0.0000000
## Sc3   0.23076923 0.3846154 0.00000000 0.0 0.0000000 0.3846154
## Sc4   0.00000000 0.5555556 0.00000000 0.0 0.0000000 0.4444444
```

What can you say of the comparison of your estimates andthe possible transitions between states that you had argued in part a

According to our probabilities shown in the graph. There are 3 arrows with probability 0. This is due to the fact that in the data there are zero transitions from $Sc2 \to Alert$, $Sc4 \to Alert$, $NR \to Alert$, $Alert \to Alert$.

This is logical given that it is very unlikely to hit an alert state. Unlike the rest of the states.

Later it will be shown that there's a unique stationary distribution (see 1d).

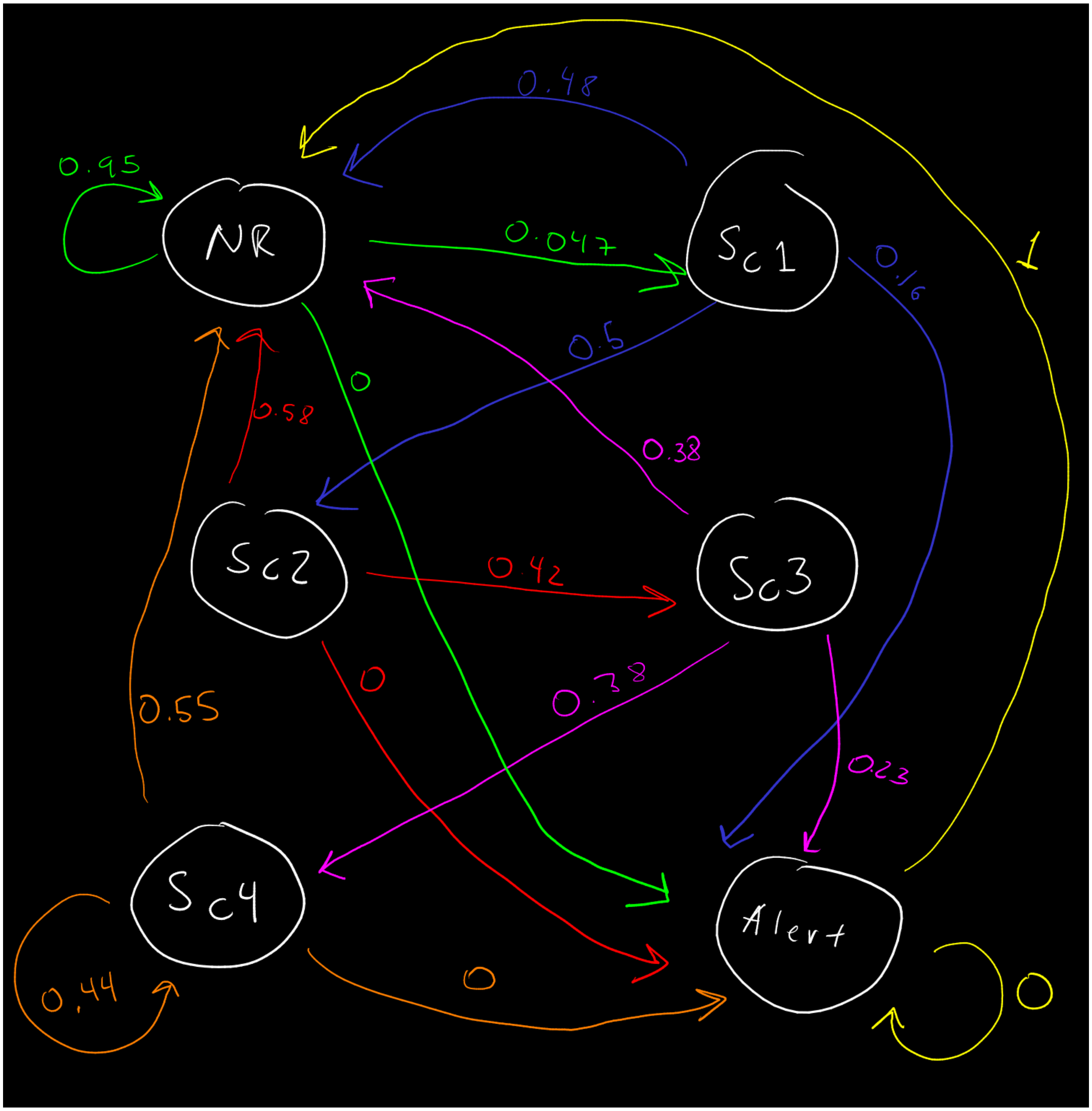

Figure 2: Graph with probabilities for problem 1 (b)

c)

Given that the first state of the chain is NR. We see the following 7 states:

```
data[1:7]
```

```
## [1] "NR" "NR" "NR" "NR" "NR" "NR" "NR"
```

And we calculate the probability as follows:

```
mat[2,2]^7
```

```
## [1] 0.713843
```

We can see the probability is 0.713843

d)

We can see that because we have a unique solution to the system, we have a unique stationary distribution.

```
stationary_dist <- function(P) {
    dim = sqrt(length(P))
    A = P - diag(dim)
    b = matrix(c(1,rep(0,dim-1)),nrow=dim,byrow=T)
    A[,1] <- rep(1,dim)
    print("The solution is the following:")
    return(matlib::Solve(t(A), b))
}
stat_dist <- stationary_dist(mat)
```

```
## [1] "The solution is the following:"
## x1            =  0.00273973
##   x2          =  0.91780822
##     x3        =  0.04315068
##       x4      =  0.02123288
##         x5    =  0.00890411
##           x6  =  0.00616438
```

```
stat_dist
```

```
## [1] "x1            =  0.00273973" "  x2          =  0.91780822"
## [3] "    x3        =  0.04315068" "      x4      =  0.02123288"
## [5] "        x5    =  0.00890411" "          x6  =  0.00616438"
```

This is our stationary distribution:

$\pi_1 = 0.00273973$

$\pi_2 = 0.91780822$

$\pi_3 = 0.04315068$

$\pi_4 = 0.02123288$

$\pi_5 = 0.00890411$

$\pi_6 = 0.00616438$

Comparing with the proportions we get from our data:

```
rel_error = c()
props = table(data)/length(data)
results <- c(0.00273973,0.91780822,0.04315068,0.02123288,0.00890411,0.00616438)
for (i in 1:length(props)) {
    rel_error[i] <- abs(props[i]-results[i])/results[i]
}
```

We can see our relative errors are all quite low ($<1 * 10^{-5}$)

e)

Taking the 120th power of our transition matrix we get the following:

```
matrixpower(mat,120)
```

```
##             [,1]      [,2]       [,3]       [,4]       [,5]        [,6]
## [1,] 0.002739726 0.9178082 0.04315068 0.02123288 0.00890411 0.006164384
## [2,] 0.002739726 0.9178082 0.04315068 0.02123288 0.00890411 0.006164384
## [3,] 0.002739726 0.9178082 0.04315068 0.02123288 0.00890411 0.006164384
## [4,] 0.002739726 0.9178082 0.04315068 0.02123288 0.00890411 0.006164384
## [5,] 0.002739726 0.9178082 0.04315068 0.02123288 0.00890411 0.006164384
## [6,] 0.002739726 0.9178082 0.04315068 0.02123288 0.00890411 0.006164384
```

Problem 2

a)

We set up the following system of equations:

$\sum_{i=0} \pi_i P_{i,0} = \pi_1$

$\sum_{i=1} \pi_i = 1$

$(1-p)\pi_1 = \pi_2 \; \ldots \; (1-p)\pi_{n-2} = \pi_{n-1} \; \ldots$

For the first equation, each $P_{i,0} = p$, therefore:

$\sum_{i=0} P_{i,0}\pi_i = \pi_1 \Rightarrow p \sum_{i=1} \pi_i = \pi_1$

$p = \pi_1$

$(1-p)p = \pi_2 \; (1-p)^2 p = \pi_3 \; \ldots \; (1-p)^{n-1} p = \pi_n \; \ldots$

Then, we get:

Because our MC is an irreducible infinite state MC, we have a unique stationary distribution π, $\pi_i = \frac{1}{\mu_i}$ and all states have expected finite return times then we have:

$E[T_i|X_0 = i] = \mu_i = \frac{1}{\pi_i}$

b)

Because it has a unique stationary distribution, it can only have one communication class (it is irreducible), all states are recurring states and there is no transient state.

c)

```
mc <- function(p, sequences ,steps) {
    n <- 100
    MarkovChain <- matrix(rep(0,sequences^2), nrow=sequences, byrow=TRUE)
    MarkovChain[,1] <- p
    for (i in 1:sequences) {
        if (i == sequences) {
            MarkovChain[i,i] <- 0
        } else {
            MarkovChain[i,i+1] <- 1-p
        }
    }
    return(MarkovChain)
}
```