We will use R to do some basic calculations on discrete space Markov chains. We will first calculate the n step probability matrix for our taxi driver model. Then we will calculate the stationary distribution for this Markov chain.

First, we need to enter the probability transition matrix into R. We may use the following command to do that:

```
p=matrix(c(0, 0.5, 0.5, 0.75, 0, 0.25, 0.75, 0.25, 0), nrow=3, ncol=3, byrow=TRUE)
```

1 N-step Probability Matrix

Now, let's calculate the 3 step probability matrix. There is no explicit function in R to raise a matrix to a power. You may copy and paste the following function into R (and then press return).

```
matrixpower=function(p,n){
temp=p
for (i in 1:(n-1)){
temp=temp %*% p}
temp}
```

Then, we may call this function to raise our probability transition matrix to the third power giving us the 3 step probability transition matrix.

```
matrixpower(p,3)
```

2 Inverting matrices and products of matrices.

Note that you can find the inverse of a matrix by using the following command

```
solve(p)
```

If you try this with the matrix defined above, this will not work. Probability transition matrices are singular. (Think about the constraint that the rows must sum to one to see why.) By changing one or two entries in the matrix p, you should be able to find an invertible matrix.

Once you have done this, try checking the inverse by multiplication.

```
solve(p) %*% p
```

3 Probability mass function at time N.

Now, let's calculate the $P(X_3 = i)$ when the initial distribution for the Markov chain is uniform. In other words, $\alpha = (1/3, 1/3, 1/3)$. In our notation, we are going to do the following calculation

$$(P(X_3 = 1), P(X_3 = 2), P(X_3 = 3)) = \alpha P^3$$

We do this in R with the following command.

```
c(1/3, 1/3, 1/3) %*% matrixpower(p,3)
```

4 Stationary Distribution of a Markov Chain

Now, we would like to calculate the stationary distribution for the taxi driver model. We want to solve the equations

$$\pi P = \pi$$

where

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

(using the notation $\pi = (\pi_1, \pi_2, ..., \pi_s)$).

The first thing we need to do is to arrange the equation in the traditional form for matrix equations ($Ax = b$). We do this by moving π to one side.

$$\pi(I - P) = (0, ..., 0)$$

The zero vector on the right hand side has as many entries as π. The matrix I is the identity matrix of the same dimensions as P. We are going to save the new matrix, $I - P$, into a new variable in R.

```
new=diag(rep(1,3))-p
```

We now need to incorporate the constraint into the equations. The equations as they are to this point look like the following

$$\pi \begin{pmatrix} 1 - p_{11} & p_{12} & p_{13} \\ p_{21} & 1 - p_{22} & p_{23} \\ p_{31} & p_{32} & 1 - p_{33} \end{pmatrix} = (0, ..., 0)$$

To incorporate the constraint, we replace one column of the new matrix with ones, and then replace the corresponding entry on the right hand side with a one. This will yield the following equations

$$\pi \begin{pmatrix} 1 & p_{12} & p_{13} \\ 1 & 1-p_{22} & p_{23} \\ 1 & p_{32} & 1-p_{33} \end{pmatrix} = (1, 0, ..., 0)$$

We can change the matrix in R with the following command

```
new[,1]=rep(1,3)
```

This replaces the first column of `new` to all ones. We can solve the equations now using the following command

```
pi=solve(t(new),c(1,0,0))
```

This R command solves $Ax = b$ when it is given A and b.

Note that we have used the R command $t()$ to transform our matrix into the transpose. We have done this to convert the problem into a problem involving column vectors. In other words R really knows how to solve this equation

$$\begin{pmatrix} 1 & 1 & 1 \\ p_{12} & 1-p_{22} & p_{32} \\ p_{13} & p_{23} & 1-p_{33} \end{pmatrix} \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$