MATH 559: BAYESIAN THEORY AND METHODS
MARKOV CHAIN CALCULATIONS

The generic notation for a discrete-time Markov chain is

$$\{X_t, t = 0, 1, \ldots\}$$

whereas typically in continuous time, we write

$$\{X(t), t \in [0, T)\}.$$

In both cases, the random process (i.e. time-indexed sequence of random variables) takes values on a countable set, $\mathcal{X}$, usually a finite set of integers such as

$$\mathcal{X} = \{1, 2, \ldots, K\}.$$

Markov chain example: A Markov chain is a discrete time model. Let the state space be $\mathcal{X} = \{1, 2\}$ and the transition matrix be

$$P = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} = \begin{pmatrix} 0.3 & 0.7 \\ 0.9 & 0.1 \end{pmatrix}$$

so that for $t = 0, 1, \ldots$,

$$P_{12} = \Pr[X_{t+1} = 2 | X_t = 1] = 0.7 \qquad P_{21} = \Pr[X_{t+1} = 1 | X_t = 2] = 0.9.$$

Let the initial distribution be the row vector

$$p^{(0)} = [\Pr[X_0 = 1] \ \Pr[X_0 = 2]] = [0.25 \ 0.75]$$

and let $p^{(t)}, t = 1, 2, \ldots$ be the distribution vector at step t. For example

$$p^{(0)} = [\Pr[X_1 = 1] \ \Pr[X_1 = 2]]$$

where, by the Theorem of Total Probability, we have that

$$\begin{aligned} \Pr[X_1 = 1] &= \Pr[X_1 = 1 | X_0 = 1] \Pr[X_0 = 1] + \Pr[X_1 = 1 | X_0 = 2] \Pr[X_0 = 2] \\ &= P_{11} p^{(0)}(1) + P_{21} p^{(0)}(2) \end{aligned}$$

and

$$\begin{aligned} \Pr[X_1 = 2] &= \Pr[X_1 = 2 | X_0 = 1] \Pr[X_0 = 1] + \Pr[X_1 = 2 | X_0 = 2] \Pr[X_0 = 2] \\ &= P_{12} p^{(0)}(1) + P_{22} p^{(0)}(2) \end{aligned}$$

so that, in matrix form

$$[\Pr[X_1 = 1] \ \Pr[X_1 = 2]] = [\Pr[X_0 = 1] \ \Pr[X_0 = 2]] \, P$$

or

$$p^{(0)} = p^{(0)} P.$$

By recursion, and the time homogeneity of the chain, we have

$$p^{(2)} = p^{(1)} P = p^{(0)} P^2.$$

and consequently, for $t = 1, 2, \ldots$, we have that

$$p^{(t)} = p^{(0)} P^t.$$

Holding time distribution: Consider the probability that a chain starting in state k stays in state k for t more steps, that is

$$\Pr[X_1 = k, X_2 = k, \ldots, X_t = k | X_0 = k].$$

It is evident that this occurs only if there are no transitions out of k in the t steps, which occurs with probability P_{kk}^t, and corresponds to the event that the first transition out of state k happens after more than t steps. Let $T_1(k)$ denote the holding time for state k, that is, the time of the first transition out of state k; we have that

$$\Pr[T_1(k) > t] = 1 - P_{kk}^t$$

so that $T_1(k) \sim Geometric(1 - P_{kk})$. A similar result holds for any k, and by the Markov property, for transitions out of a state at any point in the chain.

```
set.seed(3242)
K<-2
P<-matrix(c(0.3,0.7,0.9,0.1),K,K,byrow=T);P

+      [,1] [,2]
+ [1,]  0.3  0.7
+ [2,]  0.9  0.1

#Powers of P
Ptmp<-P %*% P
Ptmp

+      [,1] [,2]
+ [1,] 0.72 0.28
+ [2,] 0.36 0.64

for(k in 3:8){
    Ptmp<-Ptmp %*% P
    print(Ptmp)
}

+       [,1]  [,2]
+ [1,] 0.468 0.532
+ [2,] 0.684 0.316
+        [,1]   [,2]
+ [1,] 0.6192 0.3808
+ [2,] 0.4896 0.5104
+         [,1]    [,2]
+ [1,] 0.52848 0.47152
+ [2,] 0.60624 0.39376
+          [,1]     [,2]
+ [1,] 0.582912 0.417088
+ [2,] 0.536256 0.463744
+           [,1]      [,2]
+ [1,] 0.5502528 0.4497472
+ [2,] 0.5782464 0.4217536
+           [,1]      [,2]
+ [1,] 0.5698483 0.4301517
+ [2,] 0.5530522 0.4469478
```

```
Tval<-20
p.mat<-matrix(0,nrow=Tval,ncol=2)
colnames(p.mat)<-c('State 1','State 2')
rownames(p.mat)<-paste('t =',0:(Tval-1))
p.mat[1,]<-c(0.25,0.75)
for(tval in 2:Tval){p.mat[tval,]<- p.mat[tval-1,] %*% P}
p.mat
```

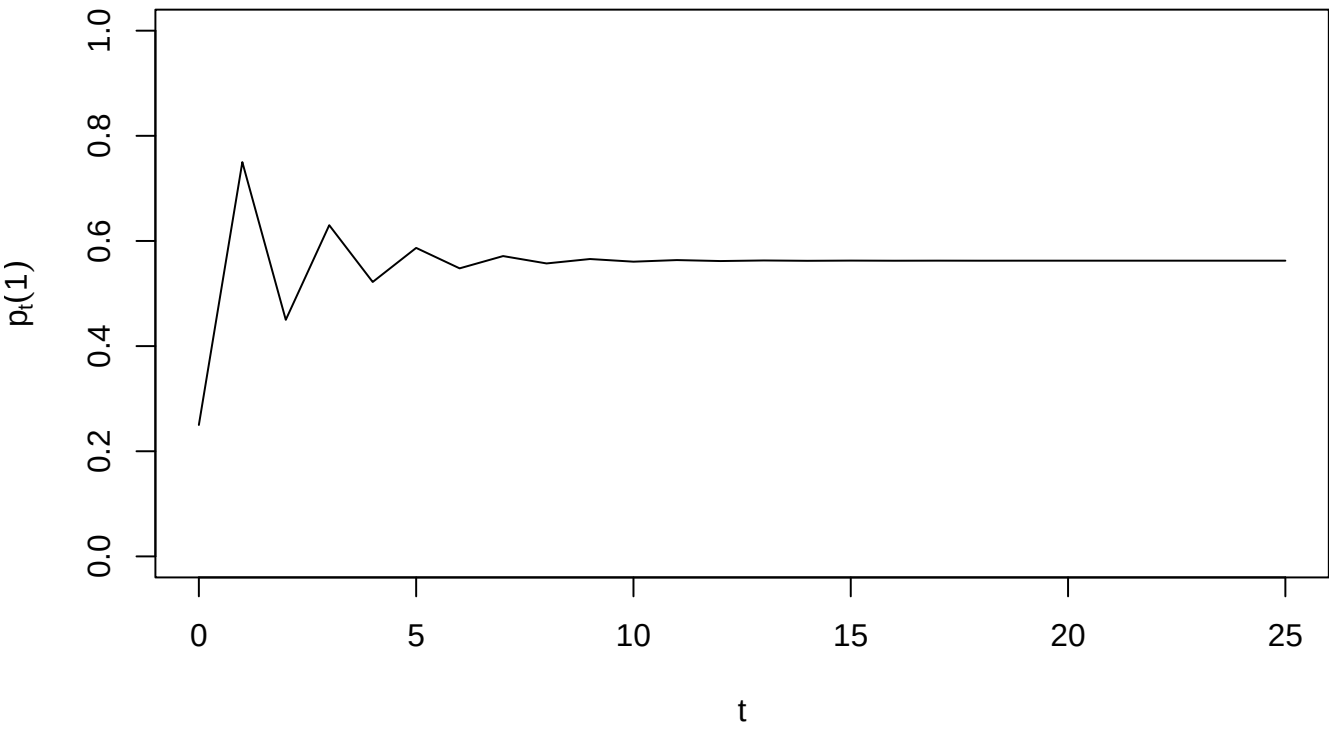
```
+          State 1   State 2
+ t = 0  0.2500000 0.7500000
+ t = 1  0.7500000 0.2500000
+ t = 2  0.4500000 0.5500000
+ t = 3  0.6300000 0.3700000
+ t = 4  0.5220000 0.4780000
+ t = 5  0.5868000 0.4132000
+ t = 6  0.5479200 0.4520800
+ t = 7  0.5712480 0.4287520
+ t = 8  0.5572512 0.4427488
+ t = 9  0.5656493 0.4343507
+ t = 10 0.5606104 0.4393896
+ t = 11 0.5636337 0.4363663
+ t = 12 0.5618198 0.4381802
+ t = 13 0.5629081 0.4370919
+ t = 14 0.5622551 0.4377449
+ t = 15 0.5626469 0.4373531
+ t = 16 0.5624118 0.4375882
+ t = 17 0.5625529 0.4374471
+ t = 18 0.5624683 0.4375317
+ t = 19 0.5625190 0.4374810
```

We can see that the probability of each state is converging: if we run this to 200 steps, we see the convergence to limiting value

```
set.seed(3242)
Tval<-200
p.mat<-matrix(0,nrow=Tval,ncol=2)
colnames(p.mat)<-c('State 1','State 2')
rownames(p.mat)<-paste('t =',0:(Tval-1))
p.mat[1,]<-c(0.25,0.75)
for(tval in 2:Tval){p.mat[tval,]<- p.mat[tval-1,] %*% P}
p.mat[(Tval-10):Tval,]
```

```
+         State 1 State 2
+ t = 189  0.5625  0.4375
+ t = 190  0.5625  0.4375
+ t = 191  0.5625  0.4375
+ t = 192  0.5625  0.4375
+ t = 193  0.5625  0.4375
+ t = 194  0.5625  0.4375
+ t = 195  0.5625  0.4375
+ t = 196  0.5625  0.4375
+ t = 197  0.5625  0.4375
+ t = 198  0.5625  0.4375
+ t = 199  0.5625  0.4375
```

```
par(mar=c(4,4,1,0))
plot(0:25,p.mat[1:26,1],type='l',ylim=range(0,1),xlab='t',ylab=expression(p[t](1)))
```



This limiting vector is the stationary distribution, π, which solves

$$\pi = \pi P$$

that is, it is probability distribution that is left unchanged by the application of the transition mapping by P

```
pivec<-p.mat[Tval,]
pivec

+ State 1 State 2
+  0.5625  0.4375

pivec %*% P

+        [,1]   [,2]
+ [1,] 0.5625 0.4375
```

Notice also that we can compute π as the identical rows of the matrix P^t as $t \longrightarrow \infty$

$$\mathbf{1}\pi = \lim_{t \longrightarrow \infty} P^t.$$

```
Ptmp<-P
for(tval in 1:200){Ptmp<-P%*% Ptmp  }
Ptmp

+        [,1]   [,2]
+ [1,] 0.5625 0.4375
+ [2,] 0.5625 0.4375
```

Diagonalization can help facilitate this calculation; attempt to write $P = ADA^{-1}$ where D is a diagonal matrix, so that

$$P^2 = PP = (ADA^{-1})(ADA^{-1}) = AD^2A^{-1}$$

so therefore

$$P^t = AD^tA^{-1}.$$

We have that the entries of D are the eigenvalues of P, and the columns of A are the (right) eigenvectors. We can use `eigen` in `R` to compute this.

```
eigen(P)

+ eigen() decomposition
+ $values
+ [1]  1.0 -0.6
+
+ $vectors
+           [,1]       [,2]
+ [1,] 0.7071068 -0.6139406
+ [2,] 0.7071068  0.7893522

D<-diag(eigen(P)$values)
A<-eigen(P)$vectors
A %*% D %*% solve(A)

+      [,1] [,2]
+ [1,]  0.3  0.7
+ [2,]  0.9  0.1
```

Therefore

$$P^t = AD^tA^{-1} = A\begin{bmatrix}1.00 & 0\\ 0 & -0.6000\end{bmatrix}^t A^{-1} = A\begin{bmatrix}1.00 & 0\\ 0 & (-0.6000)^t\end{bmatrix}A^{-1}$$

and so

$$P^t \longrightarrow A\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}A^{-1}$$

```
A %*% diag(c(1,0)) %*% solve(A)

+        [,1]   [,2]
+ [1,] 0.5625 0.4375
+ [2,] 0.5625 0.4375
```

Reversibility: This Markov chain is *reversible* as it obeys *detailed balance*, that is

$$\pi_j P_{jk} = \pi_k P_{kj} \qquad j,k = 1,2, j \neq k.$$

For detailed balance, we require that

$$\pi_1 P_{12} = \pi_2 P_{21}$$

but as $\pi_2 = 1 - \pi_1$ we must have

$$\frac{\pi_1}{1-\pi_1} = \frac{P_{21}}{P_{12}} = \frac{0.9}{0.7}$$

so therefore have that

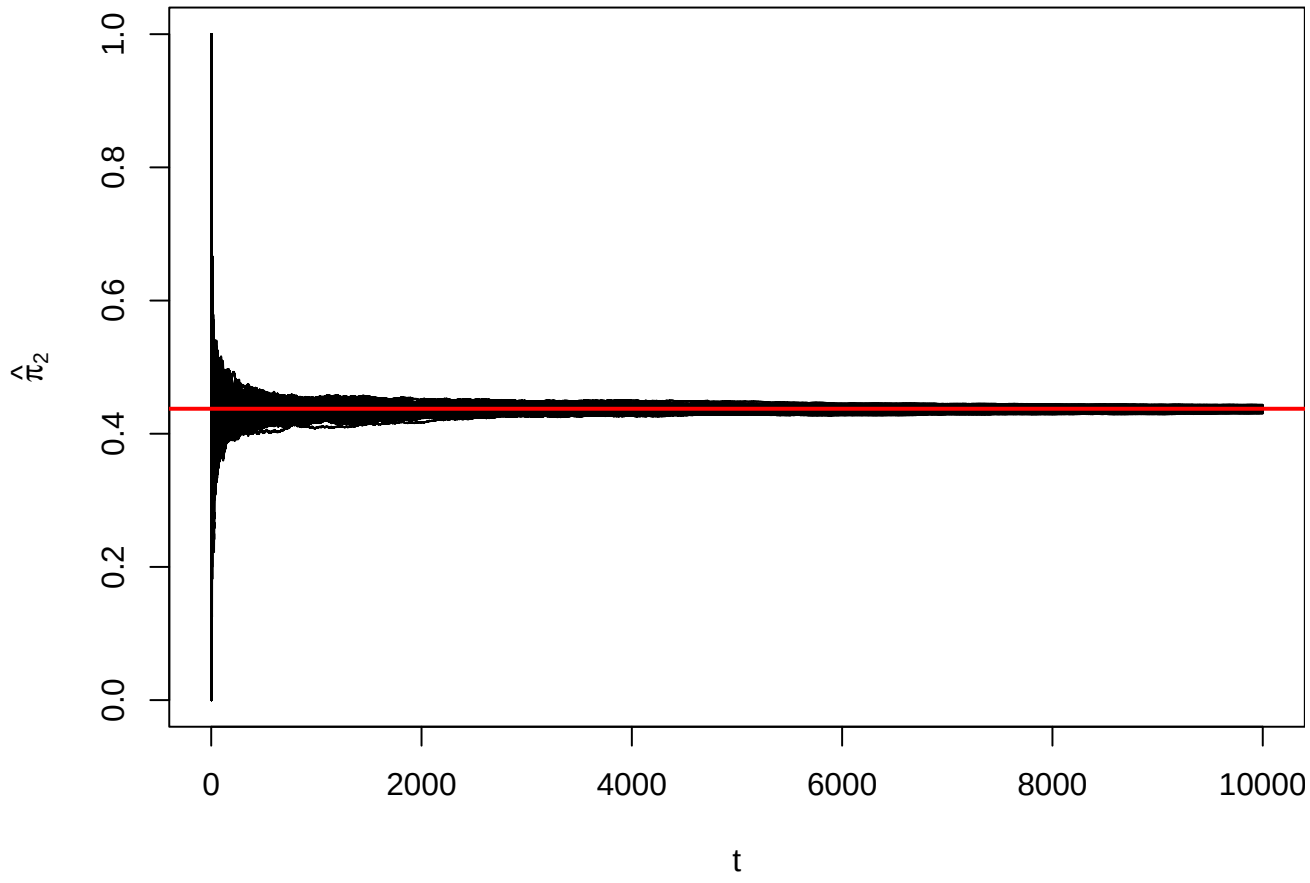
$$\pi_1 = \frac{0.9}{0.7+0.9} = 0.5625 \qquad \pi_2 = 1 - \pi_1 = 0.4375.$$

as before. Finally, note that realizations of the chain (that is, replicated sampled runs of the chain) reflect this stationary behaviour. We have that, after T steps

$$\widehat{\pi}_j = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}\{X_t = j\}$$

and observe that $\widehat{\pi}_j \xrightarrow{p} \pi_j$ as $T \longrightarrow \infty$. The following simulation shows 200 realizations of the chain run up to $T = 10000$ steps, with the estimate $\widehat{\pi}_2$ shown converging to the true value $\pi_2 = 0.4375$.

```
set.seed(3242)
Tval<-10000
nreps<-200
Xmat<-matrix(0,nrow=nreps,ncol=Tval)
Xmat[,1]<-rbinom(nreps,1,0.5)+1
pi.est<-Xmat*0
for(irep in 1:nreps){
    for(tval in 2:Tval){
        Xmat[irep,tval]<-sample(c(1:K),size=1,prob=P[Xmat[irep,tval-1],])
    }
    pi.est[irep,]<-cumsum(Xmat[irep,]-1)/c(1:Tval)
}
par(mar=c(4,5,2,0))
plot(c(1:Tval),xlab='t',pi.est[1,],ylim=range(0,1),type="l",ylab=expression(hat(pi)[2]))
for(irep in 2:nreps){lines(c(1:Tval),pi.est[irep,])}
abline(h=pivec[2],col="red",lwd=2)
```



Example: 4 state chain. Let the state space be $\mathcal{X} = \{1, 2, 3, 4\}$. First, we will randomly generate a P matrix, and set the initial distribution $p^{(0)}$ to be uniform on the 4 states.

```
set.seed(3242)
K<-4
P<-matrix(rgamma(K*K,2),K,K)
P<-t(apply(P,1,function(x){return(x/sum(x))}))
P

+           [,1]      [,2]       [,3]      [,4]
+ [1,] 0.1541770 0.6289847 0.06026378 0.1565745
+ [2,] 0.3565287 0.1108510 0.34098328 0.1916370
+ [3,] 0.3595891 0.1965477 0.20142193 0.2424413
+ [4,] 0.3586037 0.1002481 0.14334158 0.3978066

p.mat<-matrix(0,nrow=Tval,ncol=K)
colnames(p.mat)<-c('State 1','State 2','State 3','State 4')
rownames(p.mat)<-paste('t =',0:(Tval-1))
p.mat[1,]<-rep(1,K)/K
```

Here is an example simulation:

```
Tval<-25
X<-rep(0,Tval)
X[1]<-sample(1:K,size=1,prob=p.mat[1,])
for(tval in 2:Tval){
    X[tval]<-sample(c(1:K),size=1,prob=P[X[tval-1],])
}
X

+  [1] 1 2 1 2 3 1 4 1 4 1 1 2 1 2 1 2 1 4 4 4 4 4 1 1 2
```

Here is the equilibrium distribution calculations:

```
for(tval in 2:Tval){p.mat[tval,]<- p.mat[tval-1,] %*% P}
p.mat[1:Tval,]

+          State 1   State 2   State 3   State 4
+ t = 0  0.2500000 0.2500000 0.2500000 0.2500000
+ t = 1  0.3072246 0.2591579 0.1865026 0.2471149
+ t = 2  0.2954448 0.2833970 0.1798706 0.2412876
+ t = 3  0.2977961 0.2767869 0.1852547 0.2401623
+ t = 4  0.2973345 0.2784785 0.1840656 0.2401214
+ t = 5  0.2974242 0.2781378 0.1843692 0.2400688
+ t = 6  0.2974068 0.2782109 0.1843121 0.2400702
+ t = 7  0.2974102 0.2781970 0.1843247 0.2400682
+ t = 8  0.2974095 0.2781998 0.1843224 0.2400683
+ t = 9  0.2974096 0.2781993 0.1843228 0.2400682
+ t = 10 0.2974096 0.2781994 0.1843228 0.2400682
+ t = 11 0.2974096 0.2781994 0.1843228 0.2400682
+ t = 12 0.2974096 0.2781994 0.1843228 0.2400682
+ t = 13 0.2974096 0.2781994 0.1843228 0.2400682
+ t = 14 0.2974096 0.2781994 0.1843228 0.2400682
+ t = 15 0.2974096 0.2781994 0.1843228 0.2400682
```

```
+ t = 16 0.2974096 0.2781994 0.1843228 0.2400682
+ t = 17 0.2974096 0.2781994 0.1843228 0.2400682
+ t = 18 0.2974096 0.2781994 0.1843228 0.2400682
+ t = 19 0.2974096 0.2781994 0.1843228 0.2400682
+ t = 20 0.2974096 0.2781994 0.1843228 0.2400682
+ t = 21 0.2974096 0.2781994 0.1843228 0.2400682
+ t = 22 0.2974096 0.2781994 0.1843228 0.2400682
+ t = 23 0.2974096 0.2781994 0.1843228 0.2400682
+ t = 24 0.2974096 0.2781994 0.1843228 0.2400682

Ptmp<-P
for(tval in 1:Tval){Ptmp<-P%*% Ptmp  }
Ptmp

+           [,1]      [,2]      [,3]      [,4]
+ [1,] 0.2974096 0.2781994 0.1843228 0.2400682
+ [2,] 0.2974096 0.2781994 0.1843228 0.2400682
+ [3,] 0.2974096 0.2781994 0.1843228 0.2400682
+ [4,] 0.2974096 0.2781994 0.1843228 0.2400682
```

```
eigen(P)

+ eigen() decomposition
+ $values
+ [1]  1.0000000 -0.1913527  0.1741290 -0.1185198
+
+ $vectors
+      [,1]       [,2]        [,3]       [,4]
+ [1,]  0.5  0.8039007 -0.45420131  0.7763593
+ [2,]  0.5 -0.3206560 -0.22386538 -0.1994677
+ [3,]  0.5 -0.3614515 -0.05043728 -0.4695669
+ [4,]  0.5 -0.3468087  0.86083770 -0.3701159

D<-diag(eigen(P)$values)
A<-eigen(P)$vectors
A %*% D %*% solve(A)

+           [,1]      [,2]       [,3]      [,4]
+ [1,] 0.1541770 0.6289847 0.06026378 0.1565745
+ [2,] 0.3565287 0.1108510 0.34098328 0.1916370
+ [3,] 0.3595891 0.1965477 0.20142193 0.2424413
+ [4,] 0.3586037 0.1002481 0.14334158 0.3978066

A %*% diag(c(1,0,0,0)) %*% solve(A)

+           [,1]      [,2]      [,3]      [,4]
+ [1,] 0.2974096 0.2781994 0.1843228 0.2400682
+ [2,] 0.2974096 0.2781994 0.1843228 0.2400682
+ [3,] 0.2974096 0.2781994 0.1843228 0.2400682
+ [4,] 0.2974096 0.2781994 0.1843228 0.2400682
```

Notice that 1 is again an eigenvalue, all the other values are less than one in absolute value, so in the diagonalization calculation D^t converges to the matrix with value 1 in the (1,1) position, and zeros elsewhere.

Continuous time model: For a Markov chain operating in continuous time, we have that $\{X(t), t \geq 0\}$ comprises an uncountable collection of random variables that take values on $\mathcal{X}$, subject to the Markov assumption that says that for all $s, t > 0$, and for all j and k

$$\Pr[X(t+s) = k | X(s) = j, \mathcal{H}_s] = \Pr[X(t+s) = k | X(s) = j]$$

where $\mathcal{H}_s$ is (informally) the history of the chain up to time s. Typically the chain is presumed to be time homogeneous, so that

$$\Pr[X(t+s) = k | X(s) = j] \equiv \Pr[X(t) = k | X(0) = j] \qquad \forall s, t, j, k.$$

We can summarize the chain by listing the times at which the state value changes, and the state value after the change; that is we consider the sequence of transition times

$$0 = T_0 < T_1 < \cdots < T_n < \cdots$$

which is a countable collection of non-negative, increasing random variables, along with the countable collection of random variables

$$X(0), X(T_1), X(T_2), \ldots, X(T_n), \ldots$$

where we might write $X_{T_n} \equiv X(T_n) \equiv X_n$.

It remains to specify the form of

$$\Pr[X(t) = k | X(0) = j] = P_{jk}(t)$$

say. First, note that for $s, t > 0$

$$\begin{aligned}
\Pr[X(t+s) = k | X(0) = j] &= \sum_{i=1}^{K} \Pr[X(t+s) = k | X(s) = i, X(0) = j] \Pr[X(s) = i | X(0) = j] && \text{Total Prob.} \\
&= \sum_{i=1}^{K} \Pr[X(t+s) = k | X(s) = i] \Pr[X(s) = i | X(0) = j] && \text{Markov assumption} \\
&= \sum_{i=1}^{K} P_{ik}(t) P_{ji}(s) && \text{Time homog.}
\end{aligned}$$

and arranging the entries (j, k) into $K \times K$ matrix $P(t)$, we have

$$P(t+s) = P(s)P(t) = P(t)P(s) \tag{1}$$

as s and t are arbitrary. These identities are the *Chapman-Kolmogorov equations* in continuous time. Note that $P(0)$ is the identity matrix.

We consider the rate (per unit time) of transitions *out* of each state, that is, define

$$q_{jk} = \lim_{t \longrightarrow 0} \frac{P_{jk}(t)}{t} \qquad j \neq k.$$

as the rate of transition from j to k. The quantity q_{jk} is not a probability, but the *net* (or *aggregate*) rate of transitions out of state j can be written

$$\lambda_j = \sum_{k \neq j} q_{jk}$$

so we may also consider the *instantaneous* probability of transition by defining

$$P_{jk} = \frac{q_{jk}}{\sum\limits_{i \neq j} q_{ji}}$$

which is does not depend on t, and satisfies

$$0 \leq P_{jk} \leq 1 \qquad \sum_{k \neq j} P_{jk} = 1.$$

The $K \times K$ matrix of P_{jk} values has zero on the main diagonal, that is, $P_{jj} = 0, j = 1, \ldots, K$. We may write

$$q_{jk} = \lambda_j P_{jk}.$$

Note finally that

$$\lambda_j = \sum_{k \neq j} q_{jk} = \sum_{k \neq j} \lim_{t \longrightarrow 0} \frac{P_{jk}(t)}{t} = \lim_{t \longrightarrow 0} \sum_{k \neq j} \frac{P_{jk}(t)}{t} = \lim_{t \longrightarrow 0} \frac{(1 - P_{jj}(t))}{t}.$$

The model essentially assumes that for h small and for all $t > 0$

$$\Pr[X(t+h) = k | X(t) = j] = q_{jk}h + \mathrm{o}(h) = \lambda_j P_{jk} h + \mathrm{o}(h) \tag{2}$$

so that for all $k \neq j$,

$$P_{jk}(h) = \lambda_j P_{jk} h + \mathrm{o}(h).$$

Now the Chapman-Kolmogorov equations imply that as $P(0)$ is the identity

$$\frac{P(t+h) - P(t)}{h} = \frac{P(t)P(h) - P(t)}{h} = P(t)\frac{P(h) - P(0)}{h}$$

and taking the limit as $h \longrightarrow 0$ we have

$$P'(t) = P(t)P'(0) = P'(0)P(t) \tag{3}$$

where the prime indicates time derivative. All of these quantities are matrices, and the derivative is interpreted componentwise. Taking the second version

$$P'(t) = P'(0)P(t)$$

evaluating componentwise we have

$$P'_{jk}(t) = \sum_{i=1}^{K} P'_{ji}(0) P_{ik}(t).$$

Recall that for $i \neq j$

$$P'_{ji}(0) = q_{ji} = \lambda_j P_{ji}.$$

Now consider $P'_{jj}(0)$, that is

$$P'_{jj}(0) = \lim_{h \longrightarrow 0} \frac{(P_{jj}(h) - P_{jj}(0))}{h} = \lim_{h \longrightarrow 0} \frac{(P_{jj}(h) - 1)}{h} = -\lambda_j$$

by the earlier definition. The matrix $P'(0)$ therefore takes the form

$$P'_{jk}(0) = \begin{cases} q_{jk} & k \neq j \\ -\lambda_j & k = j \end{cases}$$

and is termed the *infinitesimal generator* matrix. Note that

$$\sum_{k=1}^{K} P'_{jk}(0) = -\lambda_j + \sum_{k \neq j} q_{jk} = -\lambda_j + \lambda_j = 0$$

by the earlier definitions, so the rows of $P'(0)$ sum to zero.

- **Backward equation:** We have from $P'(t) = P'(0)P(t)$ that for any j, k,

$$P'_{jk}(t) = \sum_{i=1}^{K} P'_{ji}(0) P_{ik}(t).$$

Note that for $k \neq j$,

$$P'_{jk}(0) = \lim_{h \longrightarrow 0} \frac{P_{jk}(h) - P_{jk}(0)}{h} = \lim_{h \longrightarrow 0} \frac{P_{jk}(h)}{h} = q_{jk}$$

as $P(0)$ has zero as the (j,k)th entry. In fact

$$\frac{P_{jk}(t+h) - P_{jk}(t)}{h} = \frac{1}{h}\left(\sum_{i=1}^{K} P_{ji}(h)P_{ik}(t) - P_{jk}(t)\right) \qquad \text{C-K eqn. (1)}$$

$$= \left(\sum_{i \neq j} \frac{P_{ji}(h)}{h} P_{ik}(t) - \frac{(1 - P_{jj}(h))}{h} P_{jk}(t)\right)$$

and taking limits as $h \longrightarrow 0$, we have

$$P'_{jk}(t) = \sum_{i \neq j} q_{ji} P_{ik}(t) - \lambda_j P_{jk}(t)$$

which is termed the *backward* Kolmogorov equation; note that as $t \longrightarrow 0$

$$P'_{jk}(0) = \lim_{t \longrightarrow 0} P'_{jk}(t) = \sum_{i \neq j} q_{ji} P_{ik}(0) - \lambda_j P_{jk}(0) = q_{jk}$$

as before, as only the term with $i = k$ contributes to the sum.

- **Forward equation:** We could have equivalently written $P'(t) = P(t)P'(0)$ so that

$$P'_{jk}(t) = \sum_{i=1}^{K} P_{ji}(t) P'_{ik}(0)$$

and hence

$$\frac{P_{jk}(t+h) - P_{jk}(t)}{h} = \frac{1}{h}\left(\sum_{i=1}^{K} P_{ji}(t)P_{ik}(h) - P_{jk}(t)\right) \qquad \text{C-K eqn. (1)}$$

$$= \left(\sum_{i \neq k} \frac{P_{ik}(h)}{h} P_{ji}(t) - \frac{(1 - P_{kk}(h))}{h} P_{jk}(t)\right)$$

and taking limits as $h \longrightarrow 0$, we have

$$P'_{jk}(t) = \sum_{i \neq j} q_{ik} P_{ji}(t) - \lambda_k P_{jk}(t)$$

which is termed the *forward* Kolmogorov equation.

Matrix exponentials: We seek to solve (3) to get $P(t)$ explicitly. Recall first that the solution to the ordinary differential equation

$$\frac{dy(t)}{dt} = \mu y(t)$$

is

$$y(t) = y(0)\exp\{\mu t\}.$$

The equivalent for (3) would be

$$P(t) = P(0)\exp\{P'(0)t\} = \exp\{P'(0)t\}$$

as $P(0)$ is the identity matrix, but we need to take care with this calculation as $P'(0)$ is also a matrix – the exponentiation does not operate componentwise. If we use the usual definition of the exponential function, we may write for matrix A

$$\exp\{At\} = \sum_{n=0}^{\infty} A^n \frac{t^n}{n!}$$

implying the solution

$$P(t) = \sum_{n=0}^{\infty} \{P'(0)\}^n \frac{t^n}{n!}.$$

We confirm this is a solution by checking directly:

$$P'(t) = \sum_{n=1}^{\infty} \{P'(0)\}^n \frac{t^{n-1}}{(n-1)!} = P'(0) \sum_{n=1}^{\infty} \{P'(0))\}^{n-1} \frac{t^{n-1}}{(n-1)!} \equiv P'(0)P(t).$$

Thus we have that

$$P(t) = \exp\{P'(0)t\}. \tag{4}$$

Holding times: Recall that the net rate of transition out of state j is λ_j. Suppose that $X_{n-1} \equiv X(T_{n-1}) = j$. The next transition is observed at T_n. Let

$$L_n = T_n - T_{n-1} \qquad n = 1, 2, \ldots$$

We have by the Markov property that $L_1, L_2, \ldots, L_n, \ldots$ are independent. We have also that

$$\Pr[L_n > l | X_{n-1} = j] = \exp\{-\lambda_j l\}$$

that is, given $X_{n-1} = j$, $L_n \sim Exponential(\lambda_j)$. To see this, consider the interval $(0, l]$ broken into N equal intervals,

$$\left(0, \frac{l}{N}\right], \left(\frac{l}{N}, \frac{2l}{N}\right], \ldots, \left(\frac{(N-1)l}{N}, l\right].$$

with N large. Then by assumption, for $i = 1, \ldots, N$,

$$P\left[\text{`One event in } \left(\frac{(i-1)l}{N}, \frac{il}{N}\right]', \middle| X_{n-1} = j\right] = \frac{\lambda_j l}{N} + \mathrm{o}(N^{-1})$$

$$\begin{aligned}
\Pr[\text{`No event in } (0, l]' | X_{n-1} = j] &= P\left[\bigcap_{i=1}^{N} \text{`No event in } \left(\frac{(i-1)l}{N}, \frac{il}{N}\right]', \middle| X_{n-1} = j\right] \\
&= \left(1 - \frac{\lambda_j l}{N} + \mathrm{o}(N^{-1})\right)^N \qquad \text{independence} \\
&\longrightarrow \exp\{-\lambda_j l\}
\end{aligned}$$

as $N \longrightarrow \infty$. By using the same argument, transitions out from state j occur at rates $q_{jk}, k \neq j$. Suppose, without loss of generality, that $j = 1$, and consider random variables $Z_2, \ldots, Z_K$ where $Z_k \sim Exponential(q_{1k}), k = 2, \ldots, K$. These variables are the hypothetical exit times into states $2, \ldots, K$, conditional on $X_{n-1} = 1$. The observed transition occurs after length of time L_n since entering state 1 at time T_{n-1}. Then

$$L_n = \min\{Z_2, \ldots, Z_K\}$$

and by standard theory for minimum order statistics

$$L_n \sim Exponential\left(\sum_{k=2}^{K} q_{1k}\right) \equiv Exponential(\lambda_1).$$

The equivalent result holds for whatever state the chain enters at time T_{n-1}, that is, whatever the value of X_{n-1}. Note also that

$$P_[X_n = k | X_{n-1} = j] = \frac{q_{jk}}{\lambda_j} = P_{jk}$$

which tells us that the random variables $X_0, X_1, \ldots, X_n, \ldots$ observed at the transition times form a discrete time Markov chain with transition matrix formed by the instantaneous probabilities P_{jk}.

Stationary distribution: As in the discrete time case, we can consider the stationary distribution of the continuous time Markov chain. Let $p(j,t)$ denote the probability of being in state j at time t after the initial state is chosen from initial distribution $p(j,0)$. We may define the stationary distribution π by

$$\pi_k = \lim_{t \longrightarrow \infty} P_{jk}(t) \qquad k = 1, \ldots, K$$

assuming the limit exists and does not depend on j. In vector form we have as before

$$\mathbf{1}\pi = \lim_{t \longrightarrow \infty} P(t).$$

Recall from the forward Kolmogorov equation

$$P'_{jk}(t) = \sum_{i \neq k} q_{ik} P_{ji}(t) - \lambda_k P_{jk}(t).$$

Letting $t \longrightarrow \infty$, we obtain

$$\lim_{t \longrightarrow \infty} P'_{jk}(t) = \lim_{t \longrightarrow \infty} \left\{ \sum_{i \neq k} q_{ik} P_{ji}(t) - \lambda_k P_{jk}(t) \right\}$$

$$= \sum_{i \neq k} q_{ik} \pi_i - \lambda_k \pi_k.$$

Now, as it assumed the limits exist in the interval $[0, 1]$ for the $P_{jk}(t)$, we must have that

$$\lim_{t \longrightarrow \infty} P'_{jk}(t) = 0$$

implying that π must satisfy

$$\sum_{i \neq k} q_{ik} \pi_i - \lambda_k \pi_k = 0 \qquad k = 1, \ldots, K$$

which defines a set of linear equations which can be solved to find π, that is

$$\begin{bmatrix}\pi_1 & \pi_2 & \cdots & \pi(K)\end{bmatrix}\begin{bmatrix}-\lambda_1 & q_{12} & q_{13} & \cdots & q_{1K}\\ q_{21} & -\lambda_2 & q_{23} & \cdots & q_{2K}\\ q_{31} & q_{32} & -\lambda_3 & \cdots & q_{3K}\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ q_{K1} & q_{K2} & q_{K3} & \cdots & -\lambda_K\end{bmatrix} = \mathbf{0}$$

that is, in the previous notation

$$\pi P'(0) = \mathbf{0}.$$

Note that π satisfies

$$\lambda_k \pi_k = \sum_{i \neq k} q_{ik}\pi_i \qquad k = 1, \ldots, K$$

and the left hand side of this equation measures the long run rate at which the chain leaves state k, whereas the right hand side measures the net log run rate at which the chain enters state k. That is, the stationary distribution results when these to quantities balance. We may also write this

$$\pi_k = \sum_{i \neq k} P_{ik}\pi_i$$

using the instantaneous probability matrix.

Example: Two state chain. Let $K = 2$, and suppose that the infinitesimal generator is

$$P'(0) = \begin{bmatrix}-\lambda_1 & q_{12}\\ q_{21} & -\lambda_2\end{bmatrix} = \begin{bmatrix}-1 & 1\\ 1.5 & -1.5\end{bmatrix}$$

(recall that the row sums must be zero).

```
set.seed(3242)
K<-2
Q<-matrix(c(-1,1.5,1,-1.5),2,2)
Q

+      [,1] [,2]
+ [1,] -1.0  1.0
+ [2,]  1.5 -1.5

lambda<--diag(Q)
P.inst<-Q/lambda+diag(rep(1,K))
P.inst

+      [,1] [,2]
+ [1,]    0    1
+ [2,]    1    0
```

We can compute the $P(t)$ matrix by using (4) and the `expm` function from the `Matrix` package.

```
library(Matrix)
Pt.mat<-array(0,c(10,2,2))
for(tval in 1:10){
    Pt.mat[tval,,]<-as.matrix(expm(tval*Q))
}
Pt.mat[2,,]
```

```
+           [,1]      [,2]
+ [1,] 0.6026952 0.3973048
+ [2,] 0.5959572 0.4040428

Pt.mat[5,,]

+           [,1]      [,2]
+ [1,] 0.6000015 0.3999985
+ [2,] 0.5999978 0.4000022

Pt.mat[10,,]

+      [,1] [,2]
+ [1,]  0.6  0.4
+ [2,]  0.6  0.4
```

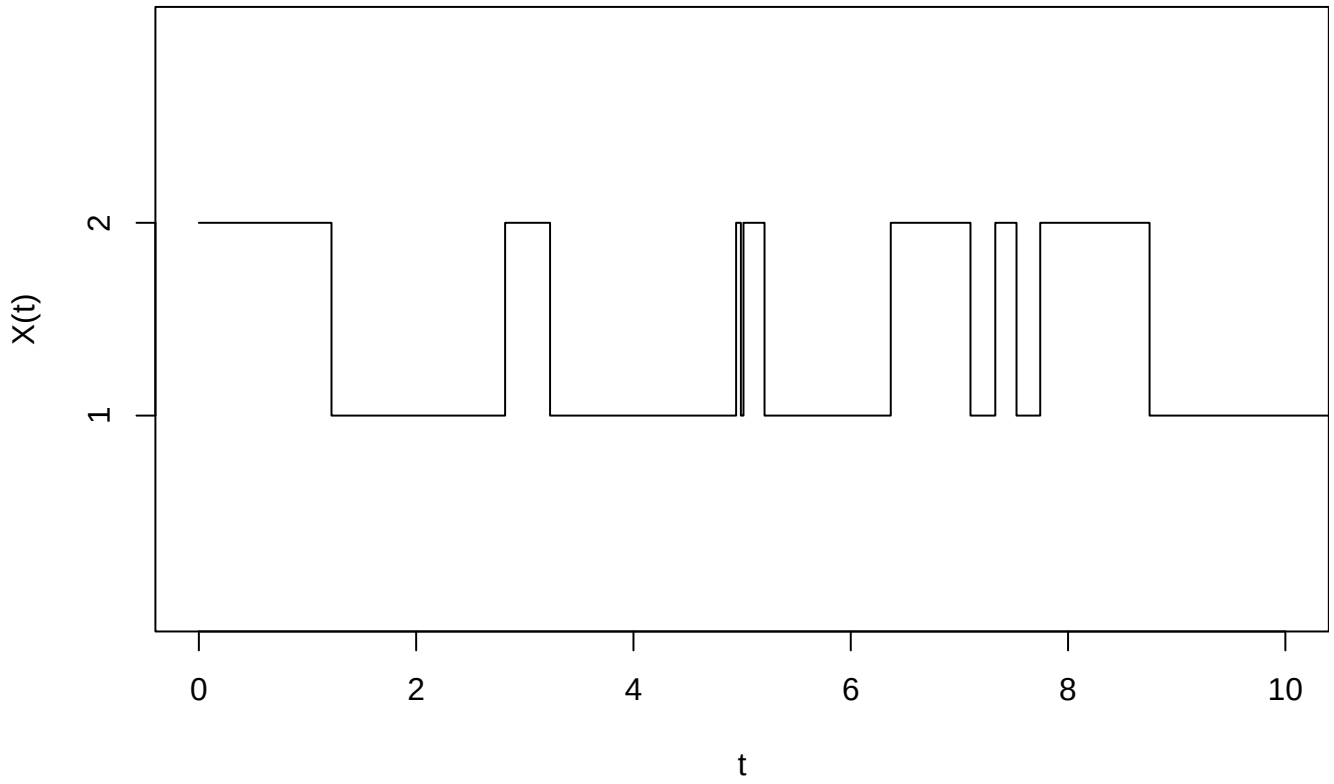
This computation gives the stationary distribution by computing for large t. Here is an example simulation on the interval $[0, 10]$

```
p0<-c(1:K)/K
X<-Tvec<-rep(0,1000)
Tval<-10
X[1]<-sample(1:K,size=1,prob=p0)
Ttot<-0
ico<-1
Tvec[1]<-0
while(Ttot < Tval){
    L<-rexp(1,lambda[X[ico]])
    Ttot<-Ttot+L
    ico<-ico+1
    Tvec[ico]<-Ttot
    X[ico]<-ifelse(X[ico-1]==1,2,1)
}
X<-X[1:ico]
Tvec<-Tvec[1:ico]
cbind(Tvec,X)

+            Tvec X
+  [1,]  0.000000 2
+  [2,]  1.220719 1
+  [3,]  2.818590 2
+  [4,]  3.232399 1
+  [5,]  4.944818 2
+  [6,]  4.988645 1
+  [7,]  5.012669 2
+  [8,]  5.206310 1
+  [9,]  6.368221 2
+ [10,]  7.102039 1
+ [11,]  7.330438 2
+ [12,]  7.525635 1
+ [13,]  7.743816 2
+ [14,]  8.750173 1
+ [15,] 11.418153 2

par(mar=c(4,4,2,0))
```

```
plot(Tvec,X,xlim=range(0,10),ylim=range(0,3),type='s',yaxt='n',xlab='t',ylab='X(t)')
axis(2,at=c(1,2))
```



Example: Four state chain. Let $K = 4$, and suppose that the infinitesimal generator is

$$P'(0) = \begin{bmatrix} -\lambda_1 & q_{12} & q_{13} & q_{14} \\ q_{21} & -\lambda_2 & q_{23} & q_{24} \\ q_{31} & q_{32} & -\lambda_3 & q_{34} \\ q_{41} & q_{42} & q_{43} & -\lambda_4 \end{bmatrix} = \begin{bmatrix} -3 & 1 & 2 & 0 \\ 2 & -3 & 0 & 1 \\ 0.5 & 0.5 & -2 & 1 \\ 0 & 0 & 2 & -2 \end{bmatrix}$$

Note that the net flow out of states 1 and 2 is higher than the net flow out of states 3 and 4. Note also that the instantaneous transition matrix is

$$\begin{bmatrix} 0 & 1/3 & 2/3 & 0 \\ 2/3 & 0 & 0 & 1/3 \\ 1/4 & 1/4 & 0 & 1/2 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

so, for example, transitions out of state 4 must be into state 3.

```
set.seed(3242)
K<-4
qvec<-c(-3,2,0.5,0,1,-3,0.5,0,2,0,-2,2,0,1,1,-2)
Q<-matrix(qvec,4,4)
Q

+      [,1] [,2] [,3] [,4]
+ [1,] -3.0  1.0    2    0
```

```
+ [2,]  2.0 -3.0    0    1
+ [3,]  0.5  0.5   -2    1
+ [4,]  0.0  0.0    2   -2

lambda<--diag(Q)
P.inst<-Q/lambda+diag(rep(1,K))
P.inst

+           [,1]      [,2]      [,3]      [,4]
+ [1,] 0.0000000 0.3333333 0.6666667 0.0000000
+ [2,] 0.6666667 0.0000000 0.0000000 0.3333333
+ [3,] 0.2500000 0.2500000 0.0000000 0.5000000
+ [4,] 0.0000000 0.0000000 1.0000000 0.0000000
```

We can compute the $P(t)$ matrix by using (4) and the `expm` function from the `Matrix` package.

```
library(Matrix)
Pt.mat<-array(0,c(10,4,4))
for(tval in 1:10){
    Pt.mat[tval,,]<-as.matrix(expm(tval*Q))
}
Pt.mat[2,,]

+           [,1]      [,2]      [,3]      [,4]
+ [1,] 0.1630240 0.1294531 0.4333614 0.2741615
+ [2,] 0.1633595 0.1297886 0.4320195 0.2748324
+ [3,] 0.1561137 0.1249581 0.4378564 0.2810718
+ [4,] 0.1495389 0.1204630 0.4416805 0.2883175

Pt.mat[5,,]

+          [,1]      [,2]      [,3]     [,4]
+ [1,] 0.156267 0.1250113 0.4374887 0.281233
+ [2,] 0.156267 0.1250114 0.4374886 0.281233
+ [3,] 0.156250 0.1250000 0.4375000 0.281250
+ [4,] 0.156233 0.1249887 0.4375113 0.281267

Pt.mat[10,,]

+         [,1]  [,2]   [,3]    [,4]
+ [1,] 0.15625 0.125 0.4375 0.28125
+ [2,] 0.15625 0.125 0.4375 0.28125
+ [3,] 0.15625 0.125 0.4375 0.28125
+ [4,] 0.15625 0.125 0.4375 0.28125
```

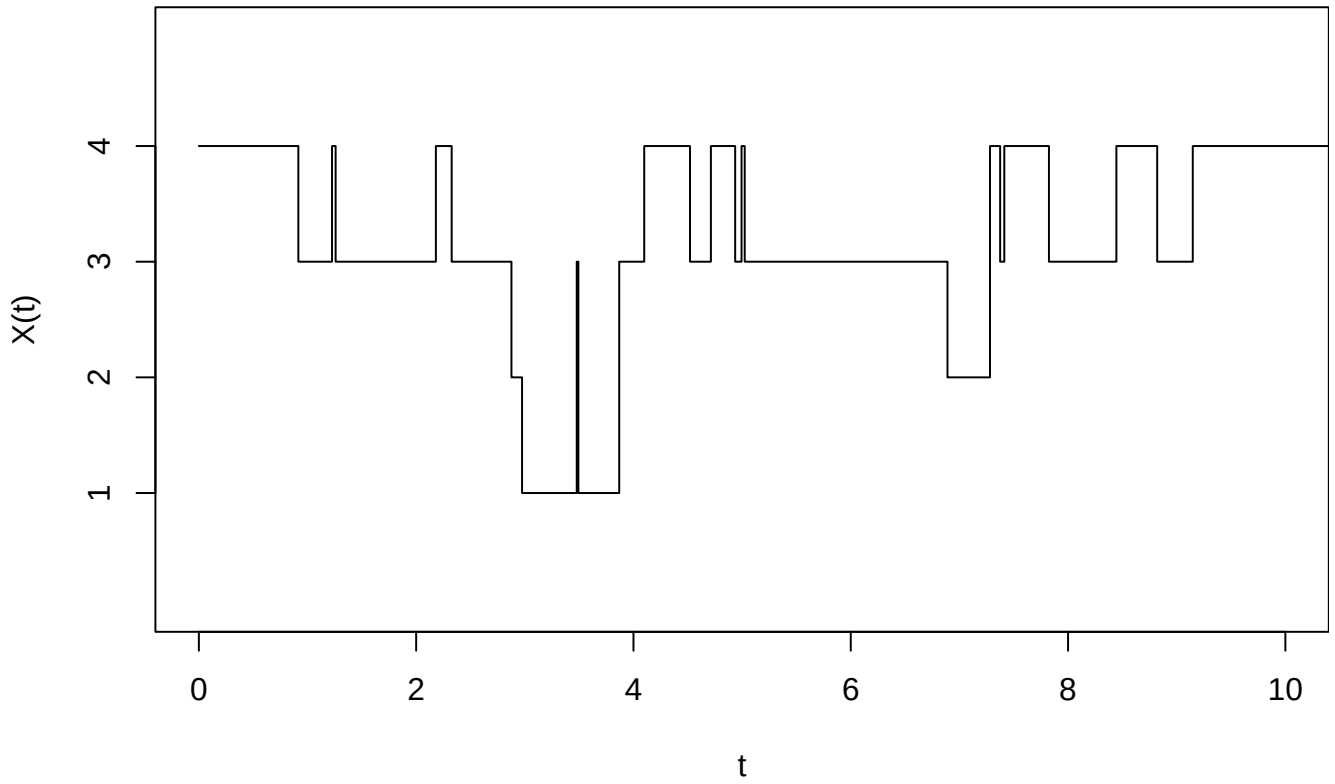
Here are two example simulations on the interval $[0, 10]$: simulation 1

```
p0<-c(1:K)/K
X<-Tvec<-rep(0,1000)
Tval<-10
X[1]<-sample(1:K,size=1,prob=p0)
Ttot<-0
ico<-1
Tvec[1]<-0
```

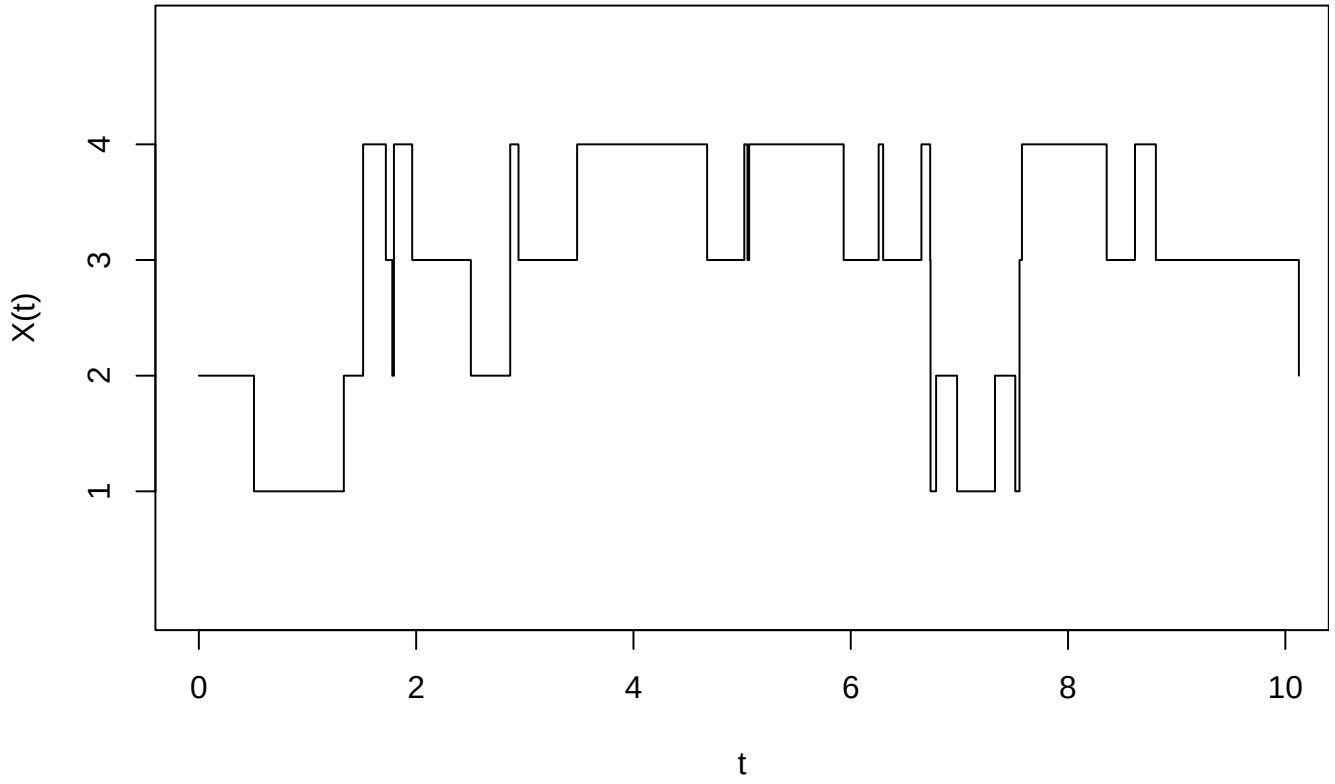
```
Svec<-c(1:K)
while(Ttot < Tval){
    L<-rexp(1,lambda[X[ico]])
    Ttot<-Ttot+L
    ico<-ico+1
    Tvec[ico]<-Ttot
    X[ico]<-sample(Svec[-X[ico-1]],size=1,prob=P.inst[X[ico-1],-X[ico-1]])
}
X<-X[1:ico]
Tvec<-Tvec[1:ico]
cbind(Tvec,X)

+            Tvec X
+  [1,]  0.000000 4
+  [2,]  0.915539 3
+  [3,]  1.225896 4
+  [4,]  1.258766 3
+  [5,]  2.181519 4
+  [6,]  2.326750 3
+  [7,]  2.877113 2
+  [8,]  2.974711 1
+  [9,]  3.477890 3
+ [10,]  3.492936 1
+ [11,]  3.868864 3
+ [12,]  4.098939 4
+ [13,]  4.520274 3
+ [14,]  4.712620 4
+ [15,]  4.936157 3
+ [16,]  4.994593 4
+ [17,]  5.024069 3
+ [18,]  6.890159 2
+ [19,]  7.281558 4
+ [20,]  7.374734 3
+ [21,]  7.413246 4
+ [22,]  7.824173 3
+ [23,]  8.444625 4
+ [24,]  8.820432 3
+ [25,]  9.147661 4
+ [26,] 10.478040 3

par(mar=c(4,4,2,0))
plot(Tvec,X,xlim=range(0,10),ylim=range(0,5),type='s',yaxt='n',xlab='t',ylab='X(t)')
axis(2,at=c(1:K))
```



Simulation 2:



Diagonalization: Recalling that we have the explicit form

$$P(t) = \exp\{P'(0)t\} = \sum_{n=0}^{\infty} \{P'(0)\}^n \frac{t^n}{n!}.$$

so if we can diagonalize $P'(0)$ as

$$P'(0) = ADA^{-1}$$

then

$$\left\{P'(0)\right\}^n = AD^nA^{-1}$$

and so

$$P(t) = \exp\{P'(0)t\} = \sum_{n=0}^{\infty} AD^nA^{-1}\frac{t^n}{n!} = A\left(\sum_{n=0}^{\infty} D^n\frac{t^n}{n!}\right)A^{-1} = A\exp\{Dt\}A^{-1}$$

If D is the diagonal matrix with diagonal entries $(d_1, \ldots, d_K)$, then $\exp\{Dt\}$ is also a diagonal matrix with diagonal entries $(\exp\{d_1t\}, \ldots, \exp\{d_Kt\})$, so $P(t)$ is easily computed.

```
eigen(Q)

+ eigen() decomposition
+ $values
+ [1] -4.000000032151140416658 -3.999999967848864912412
+ [3] -2.000000000000001776357  0.000000000000000222045
+
+ $vectors
+            [,1]       [,2]                        [,3] [,4]
+ [1,]  0.5000000 -0.5000000 -0.577350269189625620036 -0.5
+ [2,] -0.8333333  0.8333333 -0.577350269189625397992 -0.5
+ [3,]  0.1666667 -0.1666667  0.000000000000000047833 -0.5
+ [4,] -0.1666667  0.1666667  0.577350269189626286170 -0.5

D<-diag(eigen(Q)$values)
A<-eigen(Q)$vectors
round(A %*% D %*% solve(A),6)

+      [,1] [,2] [,3] [,4]
+ [1,] -3.0  1.0    2    0
+ [2,]  2.0 -3.0    0    1
+ [3,]  0.5  0.5   -2    1
+ [4,]  0.0  0.0    2   -2

round(A %*% diag(diag(exp(D*10))) %*% solve(A),6) #Computing P(10).

+         [,1]  [,2]   [,3]    [,4]
+ [1,] 0.15625 0.125 0.4375 0.28125
+ [2,] 0.15625 0.125 0.4375 0.28125
+ [3,] 0.15625 0.125 0.4375 0.28125
+ [4,] 0.15625 0.125 0.4375 0.28125
```