BIM 105
Fall 2024

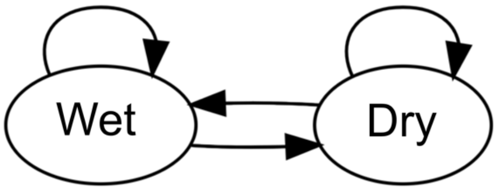
# Lab 7: Limiting Behavior of Markov Chains

Instructions: Read the prompts and use MATLAB to answer the questions. Submit your solution as a single script. A template script is available on Canvas.

## Part 1. Simple Weather System

Weather is a complex physical phenomenon. Nevertheless, some tendencies of weather patterns can be captured by relatively simple models. For example, in 1964, Leonard L. Weiss developed a simple Markov chain probability model of wet and dry days[1]. The model was very effective in predicting the duration of wet and dry sequences across a diverse set of locations. Let's develop a similar Markov chain model of the weather in Davis and use it as an example to learn about the *limiting distribution* of Markov chains.

Consider a simple Markov chain model of the weather with two states: dry and wet.



A student in Davis, California recorded the daily weather in 2018. The first day of record was dry; after that, 270 days were observed to be dry the day after a dry day; 36 days were observed to be dry the day after a wet day; 28 days were observed to be wet the day after a dry day; and 30 days were observed to be wet the day after a wet day (in reality, we know that Davis experiences many other types of weather – cloudy, foggy, etc. – but for this example, any non-rainy day is called "dry" for simplicity).

With this data in hand, we can construct a matrix of counts for the transitions between the two states in our system:

$$\begin{bmatrix} \text{\#of dry days after a dry day} & \text{\#of wet days after a dry day} \\ \text{\#of dry days after a wet day} & \text{\#of wet days ager a wet day} \end{bmatrix} = \begin{bmatrix} 270 & 30 \\ 28 & 30 \end{bmatrix}$$

This matrix informs us how many times each feasible weather transition occurred during 2018, but it's not a transition matrix yet. This is because a transition matrix holds the probability of transitioning from one state to another, not the total counts from a series of observations. So, we need to convert our observed counts into probabilities.

[1] https://doi.org/10.1175/1520-0493(1964)092<0169:SOWODD>2.3.CO;2

To do this, we need to divide each count by the number of times the system was in the previous state. Let's look at an example – the top left element of the transition matrix should give us the probability of the weather staying dry after a dry day. We know that 270 days were dry after a dry day, and 30 days were wet after a dry day. Thus, of the 300 days which were dry, 270/300=0.9 remained dry, and 30/300=0.1 transitioned to rain. These values are the transition probabilities in the first column of our transition matrix! Applying the same procedure to wet days, we arrive at:

$$P = \begin{array}{ccc} & & \text{Second Day} \\ & & \begin{array}{cc}\text{Dry} & \text{Wet}\end{array} \\ \begin{array}{c}\text{First}\\ \text{Day}\end{array} & \begin{array}{c}\text{Dry}\\ \text{Wet}\end{array} & \begin{bmatrix}0.90 & 0.10\\ 0.48 & 0.52\end{bmatrix}\end{array}$$

**<u>Question 1</u>**: Encode this transition matrix in MATLAB and save it into a variable named **`P`**.

Solution:

```
P = [0.9, 0.10; 0.48, 0.52]
```

**<u>Question 2</u>**: Recall that the distribution of states at a point in time can be computed using the **Chapman-Kolmogorov equation**, $\Pi(1) = \Pi(0)P^t$. Compute a large power of our transition matrix, $P$ (a power larger than 100). What does the resulting matrix look like?

Solution:

```
P^100 = [0.8276    0.1724
         0.8276    0.1724]
```

**<u>Question 3</u>**: Assume a weather sequence starts as a dry day, meaning that $\Pi(0) = [1, 0]$. To what distribution does the system converge to as $t$ gets large? If a weather sequence starts as a wet day instead, is the distribution the same?

Solution:

```
pi0 = [1 0]; % Dry day
pi0*P^100
     = [0.8276    0.1724]

pi0 = [0 1]; % Wet day
pi0*P^100
     = [0.8276    0.1724]
```

The same limiting distribution is observed for both initial distributions.

Taking the transition matrix to a large power via the Chapman-Kolmogorov equation is only guaranteed to converge to a unique limiting distribution if the Markov chain is irreducible, aperiodic, and positively recurrent. To arrive at stationary solutions more generally, we can use a different approach leveraging some linear algebra. Remember that a limiting distribution is one which remains constant over time. In other words, if $\Pi_L$ is a limiting distribution, it must be an eigenvector of the matrix $P$, which is defined as:

$$\Pi_L \lambda = \Pi_L P$$

Irreducible: Every state can be reached from any other state, possibly in multiple steps. There's only one equivalent class.

Aperiodic: Returning to any state does not require a fixed number of steps; the greatest common divisor of the number of steps in which a state can be revisited is 1.

Positively recurrent: Every state is expected to be revisited in a finite number of steps.

The corresponding eigenvalue $\lambda$ is 1, because the probability vector must remain unchanged (i.e., not scaled) and total probability must always sum to 1.

? where $\lambda$ is a scalar constant, referred to as the corresponding eigenvalue of the eigenvector. In the case of a transition matrix like the one characterizing the weather in Davis, the eigenvector associated with an eigenvalue of $\lambda = 1$ is the limiting distribution ($\Pi_L = \Pi_L P$).

MATLAB is well-optimized to compute eigenvectors and eigenvalues of a matrix. The **`eig`** command is what will perform the analysis for you. If you want to compute the eigenvalues and eigenvectors of some matrix **`A`**, use the command **`[V,D]=eig(A)`**. The output **`V`** will be a matrix where each column is an eigenvector. The output **`D`** will be a matrix with the eigenvalues on the diagonal.

**Question 4**: Use the **`eig`** function in MATLAB to compute the eigenvalues and eigenvectors of the transition matrix describing weather in Davis. (You'll need to use the transpose of **`P`** when you call the function in order to orient the matrix properly for MATLAB, which by default returns *right* eigenvectors instead of *left* eigenvectors as written above).

Solution: `[V, D] = eig(P');`

**Question 5**: The eigenvector associated with the limiting distribution of our Markov chain will be held in the first column of the eigenvector matrix, **`V`** (in other words, the first eigenvector is **`V(:,1)`**). What is this eigenvector?

Solution:

```
display(V(:,1))
= [0.9790
   0.2040]
```

**Question 6**: A limiting distribution, by definition, must sum to 1. Does the eigenvector from Question 5 sum to 1?

Solution:

```
display(sum(V(:,1)) =  1.1829
```

No, it does not sum to 1.

The MATLAB function 'eig' computes the eigenvectors and eigenvalues of a matrix, but these eigenvectors are not necessarily normalized in the sense that their elements sum to one

**Question 7**: A vector can be scaled to sum to 1 by dividing it by its sum. Divide **`V(:,1)`** by its sum and save the result in a new vector – what result do you get? Compare this result to your answers from Question 3.

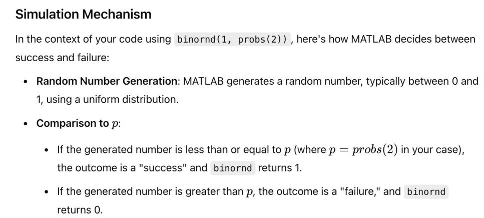
Solution:

```
vec = V(:,1)/sum(V(:,1));
= [0.8276; 0.1724]
```

The result is the same result from Q3.

**Question 8**: The limiting distribution of the Markov chain tells us what proportion of days are expected to be dry and wet, respectively. Do you think these proportions would accurately capture the weather trends in Davis over the course of a full year? How might the seasonality of weather in Davis affect the quality of the model? How might the model be improved?

Solution: Answers will vary; the model may not represent Davis weather during summer when it is nearly 100% dry for several months. Modelling the seasons separately may provide a better characterization of the typical behavior over a year.



### Simulation Mechanism

In the context of your code using `binornd(1, probs(2))`, here's how MATLAB decides between success and failure:

- **Random Number Generation**: MATLAB generates a random number, typically between 0 and 1, using a uniform distribution.
- **Comparison to $p$**:
  - If the generated number is less than or equal to $p$ (where $p = probs(2)$ in your case), the outcome is a "success" and `binornd` returns 1.
  - If the generated number is greater than $p$, the outcome is a "failure," and `binornd` returns 0.

A couple lines of code in MATLAB can simulate sequences according to our Markov model of weather. Starting with an initial state, we just need to sample each next state according to the transition probabilities of the state before it. Setting the initial state to be a dry day, we could produce sequences with **`rand`** and a couple **`if`**-statements. For example, to simulate 365 days of weather from our model:

```
P = [0.9, 0.10; 0.48, 0.52]; % Transitions
T = 365;
X = NaN(1,T);
X(1) = 1;
for i = 2:T
    probs = P(X(i-1),:);
    X(i) = binornd(1, probs(2))+1;
end
display(X)
```

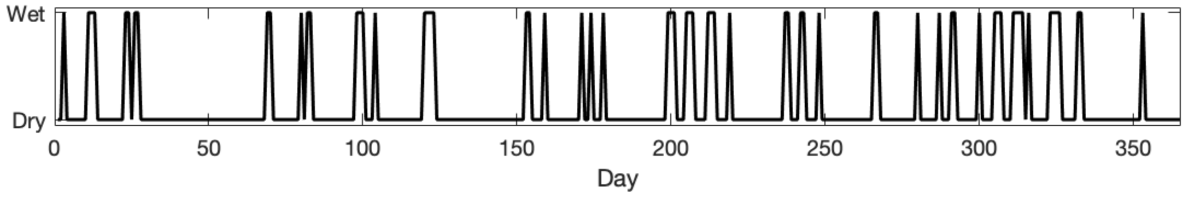
get yesterday's state, thus get corresponding row in transition matrix

Uses the binomial distribution to simulate the next state, based on probability

Which gives a sequence:

**1 1 2 1 1 1 1 1 1 1 2 2 ...**

Plotting the entire sequence of wet and dry days with **`plot`** gives a figure that looks something like this:



**<u>Question 9</u>**: Simulate a sequence of 365 days with the given model of Davis weather.

Solution: Answers will vary, but students should use the code above (or their own code) to simulate one realization of the Markov chain.

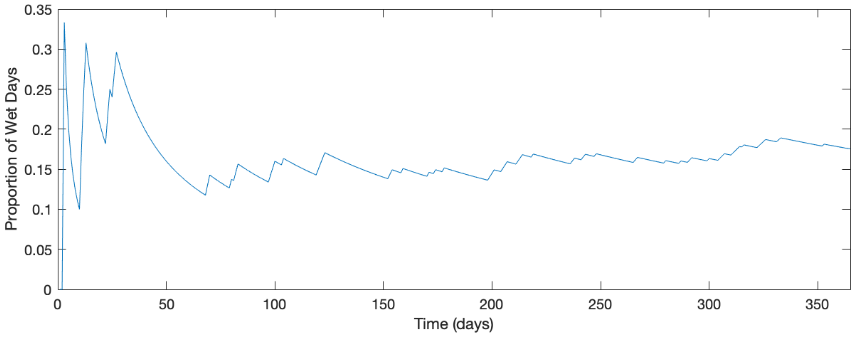
Suppose that we wanted to quantify how the weather *converges* to the limiting distribution of states. Given a sequence of states **`X`**, we can compute the *proportion of wet days* over time with **`cumsum(X-1)./(1:T)`**.

**<u>Question 10</u>**: Compute the proportion of wet days over time for your simulated sequence. Plot the proportion as it changes over time with **`plot`** – does the proportion of wet days converge to the prediction from the limiting distribution?

Solution:

```
plot(1:T, cumsum(X-1)./(1:T))
xlabel('Time (days)')
```

```
xlim([0 T])
ylabel('Proportion of Wet Days')
```



**Question 11**: Consider a different location called "Random Town" where the weather transitions are characterized by a Markov chain with $P = [0.5, 0.5; 0.5, 0.5]$.

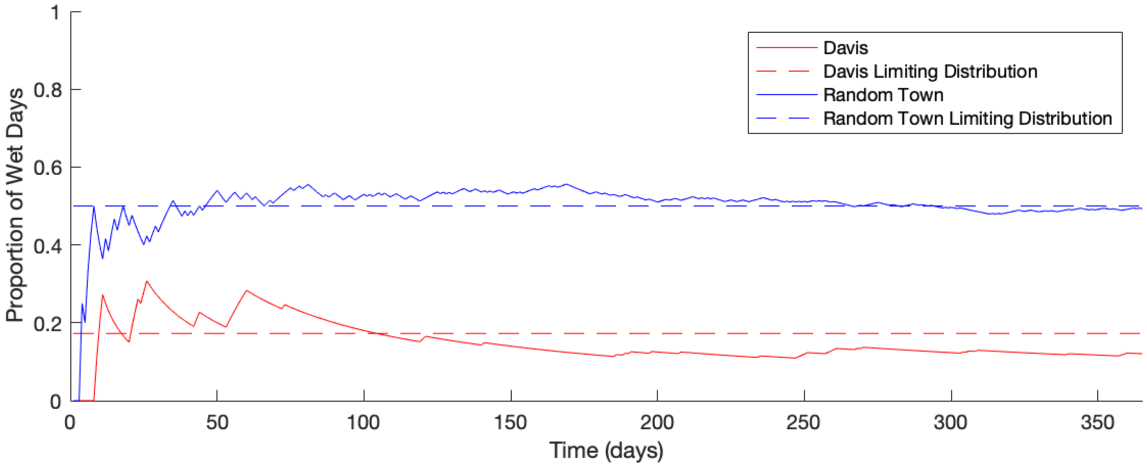
**11a**: What is the limiting distribution of this model?

Solution:
[0.5, 0.5]

**11b**: Simulate a sequence from this model and plot its proportion of wet days over time (as in Question 10). Does this system seem to converge faster or slower than the Davis weather system?

Solution:

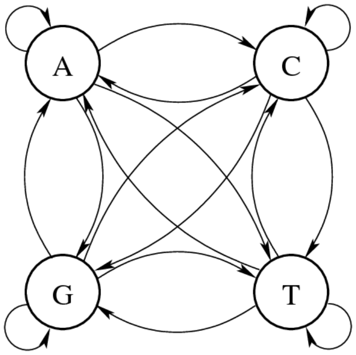
```
P2 = [0.5, 0.5; 0.5, 0.5];
X2 = NaN(1, T);
X2(1) = 1;
for i = 2:T
    probs = P2(X2(i-1),:);
    X2(i) = binornd(1, probs(2))+1;
end

figure;
hold on
plot(1:T, cumsum(X-1)./(1:T), 'r')
plot([1 T], [vec(2) vec(2)], '--r')
plot(1:T, cumsum(X2-1)./(1:T), 'b')
plot([1 T], [0.5 0.5], '--b')
xlabel('Time (days)')
xlim([0 365])
ylim([0 1])
ylabel('Proportion of Wet Days')
legend('Davis', 'Davis Limiting Distribution', ...
    'Random Town', 'Random Town Limiting Distribution')
```



The weather in Random Town seems to converge faster.

# Part 2. DNA Sequence of *E.coli* genome

A slightly more sophisticated application of a Markov chain model is to a sequence of DNA. DNA is comprised by four nucleotides (A, C, T, G), and as such, a Markov chain model can naturally be formulated using these four states. Each state can transition to any of the four states, so the graphical representation of the model might look something like this:



(image from Ohler, 2002)

In order to characterize a DNA sequence using a Markov chain model, one could count how many times each of the feasible state transitions occur in the sequence. This is analogous to counting the transitions in weather from our previous example; however, a Markov chain model of DNA would have 16 possible transitions (4 nucleotides with 4 transitions each) instead of the four transitions that arise from the two-state dry-wet model. These 16 transitions are referred to as "dinucleotide pairs."

For instance, a researcher may wish to characterize the DNA genome of the bacteria *E. coli*. This bacterium has been studied extensively since the late 1890s when it was initially discovered by German-Austrian pediatrician Theodor Escherich. The complete genome of *E. coli* was only sequenced relatively recently, however, in the 1990s. Nowadays, it's easy to download the genome from the Internet. The complete genome of a common *E. coli* strain (K12) is available with this lab as a **`.fasta`** file.

An interesting property of the *E. coli* genome is how uniformly the four nucleotides are utilized in the sequence. If you compute the proportion of the genome that is comprised by A's, by T's, by G's, and by C's, you'll find that each nucleotide is responsible for almost exactly 25% of the sequence. What does this uniformity say about nucleotide transitions, if anything? Is the same uniformity present in dinucleotide pairs?

Let's investigate. The code below (1) reads in the *E. coli* genome found in **`Ecoli-k12-genome.fasta`**, (2) counts the number of times each dinucleotide pair occurs, and (3) normalizes the dinucleotide counts to produce a transition matrix, based on the DNA sequence. Note that we have mapped the letter "A" to the first row/column of the transition matrix, "T" to the second row/column, "G" to the third row/column, and "C" to the fourth.

```
file = fastaread('Ecoli-k12-genome.fasta');
seq = file.Sequence;

di_nuc_counts = zeros(4);

for i = 1:length(seq)-1

    di_nuc = seq(i:i+1); % Read dinucleotide

    row = nuc_to_index(di_nuc(1)); % Convert letters to index for matrix
    col = nuc_to_index(di_nuc(2)); % Convert letters to index for matrix

    di_nuc_counts(row, col) = di_nuc_counts(row, col) + 1;

end

transition_matrix = di_nuc_counts./sum(di_nuc_counts, 2);
display(transition_matrix);

function index = nuc_to_index(nuc)

switch nuc
    case {'A', 'a'}
        index = 1;
        return
    case {'T', 't'}
        index = 2;
        return
    case {'G', 'g'}
        index = 3;
        return
    case {'C', 'c'}
        index = 4;
        return
end
end
```

**Question 12**: Download the *E. coli* genome file and run the code above to obtain the transition matrix. Are the dinucleotide pairs used uniformly? If not, which dinucleotide pairs are used the most?

Solution: `transition_matrix =`

```
A  [0.2958    0.2712    0.2083    0.2247
T   0.1858    0.2975    0.2824    0.2343
G   0.2271    0.2172    0.2295    0.3262
C   0.2757    0.2001    0.2939    0.2303]
```

Dinucleotide pairs are not used uniformly. We see that G’s often transition to C, T’s often transition to T’s, and C’s often transition to G’s.

**Question 13**: Find the limiting distribution of this Markov chain model. Do you observe the nucleotides are used about equally?

Solution:

```
[V, D] = eig(transition_matrix);
vec = V(:,1)/sum(V(:,1));
```

= [0.2500; 0.2500; 0.2500; 0.2500]

The use of nucleotides appears to be about uniform.

Assume that a researcher is placed in a certain predicament. The researcher has two DNA sequences, but they've been mis-labeled, and they are unsure which sequence belongs to which organism. Initially, they characterize the sequences by their nucleotide composition to distinguish them, but it is determined that both sequences have the same composition, which is 14.3% A, 14.3% T, 35.7% G, and 35.7% C.

Having been unable to distinguish the sequences by their composition, the researcher instead decides to check for differences in their nucleotide transitions. One of the organisms, *E. atrepeatium*, is known to contain some regions where many A's appear one after another and other regions where many T's appear one after another. The other organism, *E. norepeatium*, does not have the same propensity 倾向 to have A's and T's appear in such long stretches, respectively. The researcher believes this difference may assist in distinguishing the genomes.

The two mis-labeled genomes were posted with this lab as "**`Organism_A.fasta`**" and "**`Organism_B.fasta`**." Download their genomes and answer the questions below.

**Question 14**: Use the code provided for the *E. coli* example to analyze the nucleotide transitions that comprise these two genomes. What does the transition matrix look like for each genome?

Solution:

Organism A: `trans_A =`

```
    0.4513    0.0503    0.2490    0.2494
    0.0525    0.4392    0.2525    0.2558
    0.0981    0.1008    0.3996    0.4015
    0.1003    0.0991    0.3987    0.4019
```

Organism B: `trans_B =`

```
    0.2459    0.2506    0.2550    0.2486
    0.2528    0.2527    0.2457    0.2488
    0.0988    0.1008    0.3998    0.4005
    0.0984    0.1014    0.3990    0.4011
```

**Question 15**: For each genome's transition matrix, confirm that the overall composition tends to the values given in the text above by computing their limiting distribution.

overall composition tends to be similar in two organisms

Solution:

Organism A: `[VA, ~] = eig(trans_A'); display(VA(:,1)/sum(VA(:,1)));`

```
= [0.1430; 0.1405; 0.3571; 0.3594]
```

Organism B: `[VB, ~] = eig(trans_B'); display(VB(:,1)/sum(VB(:,1)));`

```
= [0.1417; 0.1441; 0.3568; 0.3573]
```

**Question 16**: Does analyzing the dinucleotide counts for these genomes allow us to distinguish which genome belongs to which organism? Which genome do you believe belongs to *E. atrepeatium*?

Solution: The Markov properties of **Organism A** are in agreement with the genome being *E. atrepretium*. The transition probabilities of A and T to remain in their respective states are much higher than in Organism B.