HYPOTHESIS TESTING OF MULTIPLE MARKOV PROCESSES OF ELECTRONIC CIRCUIT COMPONENTS

Andrew Liawan

INTRODUCTION

Markov processes are very widely applicable and can be useful for forecasting purposes in a lot of fields. One of the fields that it might be used in is electric engineering. Specifically, a stationary Markovian model is widely used in electronic circuits to predict the states of electronic components in a circuit. However, this application of stationary Markovian models on the circuits is under the assumption that the state change in the electronic components is appropriately following a stationary Markov process.

Since each electronic circuit might have different patterns of state changes, it is possible that in some circuits, a stationary Markovian model can be used to predict future states, and not in others. One way to differentiate and decide whether a stationary Markovian model can apply to an electronic circuit is to do a hypothesis test on the data of previous states in the circuit with the corresponding test statistic. This way, we can either reject or fail to reject the null hypothesis, which is the assumption that a stationary Markov process takes place in the data, and therefore prediction with a stationary Markovian model is appropriate.

This paper summarizes a paper titled Estimation, Hypothesis Testing and Parameter Correlation for Markov Chains by M. Tainiter, by going through the essential elements and properties of a standard Markov process and simultaneously examine specific conditions allowing the state changes of electronic components in electronic circuits to be framed as a Markov process. Then, a likelihood ratio as a test statistic with a specific distribution is derived to be used in the hypothesis testing. Data and the hypothesis testing of the data will be simulated using R-Code to support the ideas from the paper and apply the whole process to reach a conclusion using numerical data.

It is also worth noting that each electronic circuit has multiple electronic components, and since each component has its own chain of states, there will also be multiple Markov processes taking place at the same time in the same circuit. This fact will be considered when deriving the test statistic for the hypothesis testing.

SUMMARY OF PAPER

Markov Process in Electronic Components

In order to delve more into the Markov process and specific states in this case, we need to clearly define the elements in an electronic circuit.

A single electronic circuit contains multiple electronic components, let's say r components. Each component is able to assume a single value from a possible range of continuous values at a point of time. Each component in the circuit will also continuously change values as time goes on, which is what we call drifting. So, in a single electronic circuit, there will be r components that continuously drift.

In the effort of predicting future states of a component, we are trying to equate the drift of each component as a discrete Markov process. However, one problem is the fact that the drifting of values in a real-life electronic component is continuous with no consistent intervals, which does not fit the discrete Markov process. Therefore, certain reframing of the electronic components is needed.

Firstly, we need to partition the continuous range of possible values to be assumed by a component into disjoint sets. For example, if a component can assume values in the continuous set of [0,10], we can split the range into say 5 disjoint sets (Range 1 = [0,2], Range 2 = (2,4], etc.). Now the possible values that a

component can assume are from a discrete set of possible values of 1,2,3,4,5. This way, we can have discrete values of possible states for the components to assume. Note that the decision of disjoint set sizes from continuous sets is arbitrary as long as it is consistent.

Other than the continuous property of the assumed values, we also need to adjust the continuous property of the drift periods. A component does not drift in consistent time intervals. For example, a component can assume a discrete state of 1 for 3 seconds, then 2 for 7 seconds, then 4 for 1 second, then back to 1 for 5 seconds, etc. This is where a relationship between a discrete and continuous Markov process should be established.

The relationship states that if discrete equidistant time point sample (every t period of time) is drawn from a continuous Markov process, then the resulting chain of states of periods 0, t, 2t, 3t, etc. is indicative of a discrete Markov process. Moreover, the resulting discrete Markov process is also stationary and first order discrete Markov chain.

Markov Process Properties

Definition

In order to examine the properties of a discrete Markov Process, we need to define a discrete Markov process.

Let X(t) where t = 0, 1, 2, … be a sequence of random variables where X(t) is defined as a state of a component at time t. X(t) is considered a finite discrete Markov process of order z if it follows the following conditions.

1) The set of possible values for X(t) (or state space), should be finite.
2) $P(X(t) = m_t \mid X(t-1) = m_{t-1}, X(t-2) = m_{t-2}, \ldots, X(t-z+1) = m_{t-z+1}, X(t-z) = m_{t-z})$,

where m_t is the state of the component at time t. This implies that in a z-order finite discrete Markov process, the probability of the current state of the component (X(t)) is dependent on previous states of the component z time backwards (X(t-1), X(t-2), …, X(t-z+1), X(t-z)).

Order Property

The first property that a discrete Markov process has is the z-order of the process. In this paper, we focus on the first-order finite discrete Markov process (z=1). Therefore, the probability statement would be,

$$P(X(t) = m_t \mid X(t-1) = m_{t-1}),$$

where the probability of the current state of the component (X(t)) is only dependent on the directly previous state of the component (X(t-1)). In terms of predicting states using the model, this implies that prediction of the next state after the current state (X(t+1)) only relies on the current state of the component (X(t)).

The probability statement above is considered as a transition probability, denoted by the small letter p as the following,

$$p_{ij} = P(X(t) = j \mid X(t-1) = i),$$

where i is the state at time t-1 and j is the state at time t. Therefore, p_{ij} represents the transition probability of states changing from i to j from t-1 to t for a first-order discrete Markov process.

Stationarity Property

The second property that is the main focus of this paper is the stationarity property of a discrete Markov process. A Markov process is considered as stationary if,

$$p_{ij}(t) = p_{ij} \text{ for all } t = 1, 2, \ldots$$

where $p_{ij}(t)$ is the transition probability of changing states from i to j at each time t, and p_{ij} is an assumed constant transition probability from i to j throughout all t. This implies that the transition probability from one state to the next of a component remains constant throughout the entire process. In terms of predicting, it would be more reasonable to predict the next state of

the component if we are reasonably sure that the Markov process is stationary and consistent in its transition probabilities throughout the entire process.

Hypothesis Testing

Hypotheses

In Tainiter's paper, three pairs of hypotheses are to be tested, namely:

1) H_0: Transition probabilities are stationary.
   H_A: Transition probabilities are not stationary.
2) H_0: Markov process is first order.
   H_A: Markov process is second order.
3) H_0: Transition probabilities between components are independent.
   H_A: Transition probabilities between components are dependent.

In this paper, we are focusing on the stationarity property of the Markov process to be tested and assume that the components are already known to be first order. Therefore, we are diving deeper into hypotheses (1) from above. A rework of the hypotheses to fit in the case of r components in an electronic circuit would be,

H_0: $p_{ij}(t) = p_{ij}$ for all t = 0, 1, 2, …, r-1, r

H_A: At least one $p_{ij}(t)$ differs

Test Statistic

To conduct a hypothesis test, we need a test statistic to be tested. We can acquire this by the likelihood ratio method. The likelihood ratio is a ratio of the likelihood of the joint probability density function under the null hypothesis ($p_{ij}(t) = p_{ij}$) to the likelihood of the joint probability density function under the extreme case of the alternate hypothesis ($p_{ij}(t)$ is different for each t).

Likelihood & Likelihood Ratio

The likelihood of the joint probability density function, or joint pdf, of r components following a Markov process from t = 0 to t = T has the following form:

$$Likelihood = \prod_{t=1}^{T} \prod_{i,j} p_{ij}(t)^{n_{ij}(t)}$$

where $n_{ij}(t)$ is the number of components observed changing states from i to j from time t-1 to t (or at time t).

Let L be the likelihood of the pdf under the null hypothesis, and L' be the likelihood of the joint pdf under the alternate hypothesis. Then, the test statistic $\bar{\lambda}$ can be defined as:

$$\bar{\lambda} = \frac{L}{L'}$$

$$\bar{\lambda} = \frac{\prod_{t=1}^{T} \prod_{i,j} p_{ij}{}^{n_{ij}(t)}}{\prod_{t=1}^{T} \prod_{i,j} p_{ij}(t)^{n_{ij}(t)}}$$

$$\bar{\lambda} = \prod_{t=1}^{T} \prod_{i,j} \left(\frac{p_{ij}}{p_{ij}(t)} \right)^{n_{ij}(t)}$$

However, the underlying transition probabilities are all unknown. Therefore, we need to estimate both the assumed constant p_{ij} throughout all t, and $p_{ij}(t)$ for each t = 1, …, T, by using the Maximum Likelihood Estimation method calculated from the existing components data in place of the unknown parameters.

Maximum Likelihood Estimation

Under the null hypothesis, the Maximum Likelihood Estimate $\hat{p}_{ij}$ (assumed constant throughout all t) can be defined as,

$$\hat{p}_{ij} = \frac{\sum_{t=1}^{T} n_{ij}(t)}{\sum_{t=0}^{T-1} n_i(t)}$$

where $n_i(t)$ is the number of components in the data being in state i exactly at time t. While under the alternative hypothesis, the Maximum Likelihood Estimate $\hat{p}_{ij}(t)$ can be defined as,

$$\hat{p}_{ij}(t) = \frac{n_{ij}(t)}{n_i(t-1)}$$

where $n_i(t)$ is the number of components in the data being in state i exactly at time t-1. Using these estimates, we now have the following test statistic from the data.

$$\bar{\lambda} = \prod_{t=1}^{T} \prod_{i,j} \left(\frac{\hat{p}_{ij}}{\hat{p}_{ij}(t)} \right)^{n_{ij}(t)}$$

Test Statistic Distribution and Rejection Region

It is known from Wilk's Theorem that the distribution of -2 log $\bar{\lambda}$ is chi-squared distributed. In this particular case, the degree of freedom (df) of the chi-squared distribution is (T-1)m(m-1), where T is the final time t of the process, and m is the number of possible states that can be assumed by the components.

Setting a value of level of significance (α) against the chi-squared distribution established above, we consider the probability of rejecting the null hypothesis under the assumption that the null hypothesis is true,

$$P(\chi^2(df) > C) = \alpha$$

From this, we set a rejection region as the following. If -2 log $\bar{\lambda}$ > C, where C is a critical value depending on α and df, then we reject the null hypothesis, and conclude that the components are not following a stationary Markov process. If -2 log $\bar{\lambda} \leq$ C, then the calculated -2 log $\bar{\lambda}$ value is not in the rejection region, and therefore we fail to reject the null hypothesis, and conclude that the components are following a stationary Markov process. We illustrate this in the following simulations.

SIMULATION

Methodology

To really illustrate the application of the hypothesis test, we simulate four different scenarios of data, and apply the same hypothesis test to see the difference of results between the simulations. All simulations are done with R-Code, and all codes involved in the simulations and calculations can be referred to in the R-Code Index.

In the first simulation, we assume that the unknown underlying data generating process of 750 components is from a stationary Markov process from t = 1 to t = 100. Each t represents the state of a component at t seconds from 4 possible values of states (1,2,3,4).

The stationary Markov process from t = 1 to t = 100 is simulated from a single unknown underlying transition matrix P,

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ p_{41} & p_{42} & p_{43} & p_{44} \end{bmatrix}$$

This P matrix simulates 750 different sequences of states, which will be the data used for hypothesis testing.

From all the 750 components, the numbers of components changing states from i to j for all combinations of i,j = 1, 2, 3, 4, are recorded at each time t = 1, …, 100. This will create 16 vectors, one for each i,j combination, demonstrated below

$n_{11}(1), \ldots, n_{11}(100)$

$n_{12}(1), \ldots, n_{12}(100)$

…

$n_{43}(1), \ldots, n_{43}(100)$

$n_{44}(1), \ldots, n_{44}(100)$

After calculating the observed $n_{ij}(t)$ counts, we start establishing the hypothesis test. The corresponding hypotheses in this hypothesis test are as follows:

H_0: State drifts of 750 components follow the stationary Markov process, implying $p_{ij}(1) = \ldots = p_{ij}(100) = p_{ij}$

H_A: State drifts of 750 components does not follow the stationary Markov process, implying at least one $p_{ij}(t)$ for t = 1, …, 100 is different from p_{ij}

Under the null hypothesis of stationary Markov process, all the transition probabilities are assumed to be equal all across t = 1 to t = 100, resulting in only one transition probability value p for each i,j combination.

We then calculate the Maximum Likelihood Estimator of $\hat{p}_{ij}$ in transition matrix form $\hat{P}$,

$$\hat{P} = \begin{bmatrix} \hat{p}_{11} & \hat{p}_{12} & \hat{p}_{13} & \hat{p}_{14} \\ \hat{p}_{21} & \hat{p}_{22} & \hat{p}_{23} & \hat{p}_{24} \\ \hat{p}_{31} & \hat{p}_{32} & \hat{p}_{33} & \hat{p}_{34} \\ \hat{p}_{41} & \hat{p}_{42} & \hat{p}_{43} & \hat{p}_{44} \end{bmatrix}$$

Under the alternative hypothesis, we assume that each t has a different transition probability of p_{ij}, and therefore we have different $p_{ij}(t)$ values for each t. We then calculate the Maximum Likelihood Estimators of $\hat{p}_{ij}(t)$ for each t = 1, …, 100 are also calculated, resulting in 16 vectors, one for each i,j combination, demonstrated below

$$\hat{p}_{11}(1), \dots, \hat{p}_{11}(100)$$

$$\hat{p}_{12}(1), \dots, \hat{p}_{12}(100)$$

$$\dots$$

$$\hat{p}_{43}(1), \dots, \hat{p}_{43}(100)$$

$$\hat{p}_{44}(1), \dots, \hat{p}_{44}(100)$$

We then find the test statistic of the hypothesis test by finding the likelihoods of the probability functions under both the null and alternative hypothesis, then finding the Likelihood Ratio, $\bar{\lambda}$ using the formula below,

$$\bar{\lambda} = \prod_{t=1}^{100} \prod_{i,j} \frac{\hat{p}_{ij}}{\hat{p}_{ij}(t)}^{n_{ij}(t)}$$

Using the assumption of -2 log $\bar{\lambda}$ ~ χ^2(df), we set the level of significance α, and get the following,

$$P(\chi^2(df) > C) = \alpha,$$

implying that we reject the null hypothesis is if -2 log $\bar{\lambda}$ > C, where C depends on the degrees of function (df) of the distribution. This is the basis whether we reject or fail to reject our established null hypothesis. Theoretically, it should correctly fail to reject the null hypothesis of the stationarity assumption.

The second simulation follows the same steps and parameters as the first one. The only difference is that we simulate a stationary Markov process from t = 1 to t = 90 using transition matrix P, and randomly sample states from the 4 possible states from t = 91 to t = 90. This scenario is to simulate a process where transition probabilities will not be entirely stationary from start to finish. Theoretically, the hypothesis test should reject the null hypothesis of the stationarity assumption due to the randomness of states from t = 91 to t = 100.

The third simulation is similar to the second simulation, again with the same steps and parameters. However, instead of a random sample of states from t = 91 to t = 100, we used a second transition matrix P_2 different from the transition matrix P_1 used from t = 1 to t = 90. The difference between P_1 and P_2 will be very subtle and small. However, it will still simulate a non-stationary process from t = 1 to t = 100, and therefore theoretically it should also reject the null hypothesis of the stationarity assumption.

The fourth simulation will be the same as the third one. The only difference now is reducing the number of components from 750 to 150. This is to illustrate how the hypothesis test might produce different decisions with the same underlying transition matrices P_1 and P_2 due to the sample size of the data.

Results and Analysis

Simulation 1. Stationary Process from Start to End (t = 1,2,…,99,100)

We simulate all 100 consecutive state changes of 750 components by using a single transition matrix P, since we are simulating stationary processes for all 750

components. Let the underlying unknown transition matrix P be,

$$P = \begin{bmatrix} 0.35 & 0.55 & 0.06 & 0.04 \\ 0.20 & 0.30 & 0.40 & 0.10 \\ 0.50 & 0.30 & 0.20 & 0 \\ 0 & 0 & 0.70 & 0.30 \end{bmatrix}$$

From the simulated data of 750 processes, we calculate the $n_{ij}(t)$ vectors. For illustration, the vector with $n_{11}(t)$ for all t = 1, …, 100 is,

$n_{11}(1)$, …, $n_{11}(100)$ = 68, 67, 83, …, 90, 74, 79

Using the simulated data of 750 components, the Maximum Likelihood Estimator of $\hat{p}_{ij}$ in transition matrix form $\hat{P}$ is,

$$\hat{P} = \begin{bmatrix} 0.351 & 0.544 & 0.065 & 0.039 \\ 0.202 & 0.300 & 0.397 & 0.101 \\ 0.503 & 0.297 & 0.200 & 0 \\ 0 & 0 & 0.704 & 0.296 \end{bmatrix},$$

and the Maximum Likelihood Estimators of $\hat{p}_{ij}(t)$ are also calculated. For illustration, the Maximum Likelihood Estimators of $\hat{p}_{11}(t)$ for all t = 1, …, 100 is,

$\hat{p}_{11}(1), \dots, \hat{p}_{11}(100)$= 0.364, 0.333, …, 0.315, 0.332

Calculating -2 log $\bar{\lambda}$ we get the value of 939.0055. Using the assumption of -2 log $\bar{\lambda}$ ~ χ^2(df), where df = 99 x 4 x 3 = 1188, we set the level of significance α = 0.05, and get the following,

$P(\chi^2(1188) > C) = 0.05,$

where C = 1269.298. This implies that we reject the null hypothesis is if -2 log $\bar{\lambda}$ > 1269.298.

Since our -2 log $\bar{\lambda}$ test statistic value does not lie in the rejection region, we fail to reject the null hypothesis. This implies that the simulated data of 750 components reasonably follow a stationary Markov process, and the hypothesis test succeeds in detecting a stationary process.

This also means that we can reasonably predict the next states (i.e. t = 101, 102, etc.) using the estimated $\hat{P}$ transition matrix since the stationarity assumption is reasonable.

Simulation 2. Stationary Process from t = 1,…,90, Random sampling of states from t = 91,…,100

We simulate the first 90 consecutive state changes of 750 components by using a transition matrix P. Let the underlying unknown transition matrix P be,

$$P = \begin{bmatrix} 0.35 & 0.55 & 0.06 & 0.04 \\ 0.20 & 0.30 & 0.40 & 0.10 \\ 0.50 & 0.30 & 0.20 & 0 \\ 0 & 0 & 0.70 & 0.30 \end{bmatrix}$$

The states for t = 91, …, 100 are randomly sampled from the 4 possible values of states (1,2,3,4) with equal probabilities using the sample function.

From the simulated data of 750 processes, we calculate the $n_{ij}(t)$ vectors. For illustration, the vectors with $n_{11}(t)$ for t = 1, …, 90 and for t = 91, …, 100 are,

$n_{11}(1)$, …, $n_{11}(90)$ = 68, 67, 81, …, 79, 81, 78

$n_{11}(91)$, …, $n_{11}(100)$ = 55, 41, 41, …, 40, 44, 47

As we can see here, the observed number of components for n_{11} is generally different between the 2 vectors above, due to the different methods of state changes (transition matrix P vs. random sampling). This is also the case with every other n_{ij}, theoretically at first glance making the process non-stationary, and will reflect on the decision by the hypothesis test.

Using the simulated data of 750 components, the Maximum Likelihood Estimator of $\hat{p}_{ij}$ in transition matrix form $\hat{P}$ is,

$$\hat{P} = \begin{bmatrix} 0.339 & 0.524 & 0.080 & 0.057 \\ 0.204 & 0.296 & 0.388 & 0.113 \\ 0.481 & 0.290 & 0.205 & 0.024 \\ 0.066 & 0.071 & 0.578 & 0.285 \end{bmatrix}$$

As we can see here the Maximum Likelihood Estimator of the transition probabilities are slightly different compared to the underlying P matrix. This is due to the random sampling of states at the end of the

processes, which skew the transition probabilities values.

Maximum Likelihood Estimators of $\hat{p}_{ij}(t)$ are also calculated. For illustration, the Maximum Likelihood Estimators of $\hat{p}_{11}(t)$ for t = 1, …, 90 and for t = 91, …, 100 are,

$\hat{p}_{11}(1), \dots, \hat{p}_{11}(90)$= 0.343, 0.332, …, 0.333, 0.332

$\hat{p}_{11}(91), \dots, \hat{p}_{11}(100)$= 0.240, …, 0.246, 0.261

Similar to the component counts, the vectors MLE of $\hat{p}_{ij}(t)$ for the 2 different time ranges are generally different, which will also reflect in the hypothesis test decision.

Calculating -2 log $\bar{\lambda}$ we get the value of Inf, a value too big for the R-code to compute. Using the assumption of -2 log $\bar{\lambda}$ ~ χ^2(df), where df = 99 x 4 x 3 = 1188, we set the level of significance α = 0.05, and get the following,

$$P(\chi^2(1188) > C) = 0.05,$$

where C = 1269.298. This implies that we reject the null hypothesis is if -2 log $\bar{\lambda}$ > 1269.298.

Since our -2 log $\bar{\lambda}$ test statistic value is very big, it lies inside the rejection region, and we reject the null hypothesis. This implies the components do not follow a stationary Markov process, which makes sense since we simulate 2 different processes of state drifts in the components, simulating a non-stationary process. Thus, the hypothesis test works on detecting a non-stationary process.

Simulation 3. Stationary Process from P1 from t = 1,…,90, stationary Process from P2 from t = 91,…,100 (r = 750)

We simulate the first 90 consecutive state changes of 750 components by using a transition matrix P1. Let the underlying unknown transition matrix P1 be,

$$P_1 = \begin{bmatrix} 0.35 & 0.55 & 0.06 & 0.04 \\ 0.20 & 0.30 & 0.40 & 0.10 \\ 0.50 & 0.30 & 0.20 & 0 \\ 0 & 0 & 0.70 & 0.30 \end{bmatrix}$$

The states for t = 91, …, 100 are simulated using a different transition matrix P2. Let the underlying unknown transition matrix P1 be,

$$P_2 = \begin{bmatrix} 0.34 & 0.56 & 0.05 & 0.05 \\ 0.21 & 0.29 & 0.39 & 0.11 \\ 0.49 & 0.31 & 0.19 & 0.01 \\ 0.01 & 0.01 & 0.69 & 0.29 \end{bmatrix}$$

From the simulated data of 750 processes, we calculate the $n_{ij}(t)$ vectors. For illustration, the vectors with $n_{11}(t)$ for t = 1, …, 90 and for t = 91, …, 100 are,

$n_{11}(1)$, …, $n_{11}(90)$ = 72, 74, 91, …, 72, 78, 79

$n_{11}(91)$, …, $n_{11}(100)$ = 72, 59, 51, …, 86, 79, 77

As we can see here, unlike simulation 2, the observed number of components for n_{11} is not visibly and noticeably different between the 2 vectors above, even if we use different transition matrices. This is because the difference between the 2 matrices is very small. This is also the case with every other n_{ij}.

Using the simulated data of 750 components, the Maximum Likelihood Estimator of $\hat{p}_{ij}$ in transition matrix form $\hat{P}$ is,

$$\hat{P} = \begin{bmatrix} 0.351 & 0.542 & 0.064 & 0.042 \\ 0.204 & 0.298 & 0.396 & 0.102 \\ 0.498 & 0.298 & 0.199 & 0.004 \\ 0.004 & 0.004 & 0.692 & 0.301 \end{bmatrix}$$

As we can see here the Maximum Likelihood Estimator of the transition probabilities are very close compared to both underlying P1 and P2 matrices.

Maximum Likelihood Estimators of $\hat{p}_{ij}(t)$ are also calculated. For illustration, the Maximum Likelihood Estimators of $\hat{p}_{11}(t)$ for t = 1, …, 90 and for t = 91, …, 100 are,

$\hat{p}_{11}(1), \dots, \hat{p}_{11}(90)$= 0.371, 0.372, …, 0.339, 0.372

$\hat{p}_{11}(91), \dots, \hat{p}_{11}(100)$= 0.306, …, 0.348, 0.342

Again, unlike simulation 2, the Maximum Likelihood Estimator of the transition probabilities are not noticeably different between the two time ranges,

as well as the underlying P matrix, even if we use different transition matrices, due to the very small difference. All the illustrations above point to an initial assumption that the components follow a stationary process. The hypothesis test will statistically confirm or deny that initial assumption.

Calculating -2 log $\bar{\lambda}$ we get the value of Inf, a value too big for the R-code to compute. Using the assumption of -2 log $\bar{\lambda} \sim \chi^2(df)$, where df = 99 x 4 x 3 = 1188, we set the level of significance $\alpha = 0.05$, and get the following,

$$P(\chi^2(1188) > C) = 0.05,$$

where C = 1269.298. This implies that we reject the null hypothesis is if -2 log $\bar{\lambda} > 1269.298$.

Since our -2 log $\bar{\lambda}$ test statistic value is very big, it lies inside the rejection region, and we reject the null hypothesis. This implies that the process of the components does not follow a stationary Markov process, which makes sense since we simulate 2 different processes with 2 different transition matrices, even if the difference is very small. This also proves that the hypothesis test is able to pick up small differences in different transition matrices to be able to conclude that the process is non-stationary, as long as sample size is big enough. We will demonstrate this in simulation 4.

Simulation 4. Stationary Process from P1 from t = 1,…,90, stationary Process from P2 from t = 91,…,100 (r = 150)

We repeat simulation 3 for simulation 4, with the only difference being the considerably smaller number of components in simulation 4. This implies a considerably smaller sample size of data.

Calculating -2 log $\bar{\lambda}$ we get the value of 1187.706. Using the assumption of -2 log $\bar{\lambda} \sim \chi^2(df)$, where df = 99 x 4 x 3 = 1188, we set the level of significance $\alpha = 0.05$, and get the following,

$$P(\chi^2(1188) > C) = 0.05,$$

where C = 1269.298. This implies that we reject the null hypothesis is if -2 log $\bar{\lambda} > 1269.298$.

Since our -2 log $\bar{\lambda}$ test statistic value is not inside the rejection region, we fail to reject the null hypothesis, concluding a stationary process. This implies that the hypothesis test fails to detect the difference between the 2 different transition matrices, and considers whole process stationary, even though in simulation 3 the test did detect the difference. This can be explained by the smaller sample size of data in simulation 4, which usually tends to decrease the power of a test in detecting subtle differences.

Here are the results of all 4 simulations in a table.

	Sim 1	Sim 2	Sim 3	Sim 4
r	750	750	750	150
-2 log $\bar{\lambda}$	939.0	Inf	Inf	1187.7
C	1269.3	1269.3	1269.3	1269.3
Decision	Fail to reject H_0	Reject H_0	Reject H_0	Fail to reject H_0
Conclusion	St	Non-St	Non-St	St

*St = Stationary, Non-St = Non-Stationary

Conclusion

From all the simulations, we conclude that the hypothesis test formulated above does work when it comes to identifying the stationarity of a component in a process, for the purpose of predicting future states of the components. However, it is also worth noting that a difference of sample size of the number of components also affects the effectiveness of the hypothesis test in detecting whether a process is stationary, shown by the difference of results of simulations 3 and 4.

REFERENCES

[1] M. Tainiter, "Estimation, Hypothesis Testing and Parameter Correlation for Markov Chains," in IEEE Transactions on Reliability, vol. R-12, no. 4, pp. 26-35, Dec. 1963, doi: 10.1109/TR.1963.5218225.

CODE/DATA APPENDIX

Creating a Function That Simulates Discrete Markov Chain of r Component Parameters with m Possible States Observed over t time periods/iterations

```
# simulate discrete Markov chains according to transition matrix P
run.mc.sim <- function( P, num.iters,num.states) {

  # number of possible states
  num.states <- nrow(P)

  # stores the states X_t through time
  states     <- numeric(num.iters)

  # initialize variable for first state
  states[1]    <- sample(seq(1,num.states,1),1)

  for(t in 2:num.iters) {

    # probability vector to simulate next state X_{t+1}
    p  <- P[states[t-1],]

    ## draw from multinomial and determine state
    states[t] <-  which(rmultinom(1, 1, p) == 1)
  }
  return(states)
}
```

Simulation of Component Data

Component Parameters (r) = 1, 2, 3, … , 749, 750

Time periods (t) = 1, 2, 3, … , 99, 100

States (m) = 1, 2, 3, 4

Stationary Process from beginning to end

```
set.seed(403-560)

# P matrix of pij transition probabilities to simulate chains
num.states = 4
P <- t(matrix(c(0.35,0.55,0.06,0.04,
                0.20,0.30,0.40,0.10,
                0.50,0.30,0.20,0.00,
                0.00,0.00,0.70,0.30),nrow=num.states,ncol=num.states))

# Number of Electronic Components (r)
num.comp = 750

# Number of Iterations/Time periods (t)
time = 101 ## t+1 to take account of initial state of component
```

```r
# Simulating r Markov Processes
chain <- matrix(NA,ncol=num.comp,nrow=time)
for (c in seq_len(num.comp)){
  chain[,c] <- run.mc.sim(P,time,num.states)
}

# Markov Chain of the First Component
chain[,1]
```

```
##   [1] 2 3 1 2 4 4 3 3 2 4 3 2 2 2 3 1 2 3 1 2 4 3 2 3 1 1 1 1 2 4 3 2 1 2 2 2 2
##  [38] 2 3 1 1 1 4 4 3 1 2 1 1 1 2 3 1 2 3 2 2 3 1 1 3 1 2 1 3 1 2 4 4 3 2 3 1 1
##  [75] 2 3 1 1 2 3 2 2 4 4 3 3 3 1 1 2 4 3 2 3 1 1 1 1 1 1 2
```

```r
# Number of Observations of each transition (nij)
N <- list()
for (t in 1:(time-1)){
  N[[t]] <- paste(chain[t,],chain[t+1,],sep="")
}
n <- list("n11"=NULL,"n12"=NULL,"n13"=NULL,"n14"=NULL,
          "n21"=NULL,"n22"=NULL,"n23"=NULL,"n24"=NULL,
          "n31"=NULL,"n32"=NULL,"n33"=NULL,"n34"=NULL,
          "n41"=NULL,"n42"=NULL,"n43"=NULL,"n44"=NULL)
for (i in 1:(time-1)){
  n$n11[i] <- length(N[[i]][N[[i]]=="11"])
  n$n12[i] <- length(N[[i]][N[[i]]=="12"])
  n$n13[i] <- length(N[[i]][N[[i]]=="13"])
  n$n14[i] <- length(N[[i]][N[[i]]=="14"])
  n$n21[i] <- length(N[[i]][N[[i]]=="21"])
  n$n22[i] <- length(N[[i]][N[[i]]=="22"])
  n$n23[i] <- length(N[[i]][N[[i]]=="23"])
  n$n24[i] <- length(N[[i]][N[[i]]=="24"])
  n$n31[i] <- length(N[[i]][N[[i]]=="31"])
  n$n32[i] <- length(N[[i]][N[[i]]=="32"])
  n$n33[i] <- length(N[[i]][N[[i]]=="33"])
  n$n34[i] <- length(N[[i]][N[[i]]=="34"])
  n$n41[i] <- length(N[[i]][N[[i]]=="41"])
  n$n42[i] <- length(N[[i]][N[[i]]=="42"])
  n$n43[i] <- length(N[[i]][N[[i]]=="43"])
  n$n44[i] <- length(N[[i]][N[[i]]=="44"])
}

n$n11
```

```
##   [1] 68 67 83 86 84 82 80 68 59 77 76 93 78 84 84 95 97 74 87 71 53 78 81 83 90
##  [26] 97 95 87 83 75 98 85 77 99 70 73 80 65 71 92 87 83 88 97 87 93 91 78 86 86
##  [51] 77 86 97 96 87 84 86 63 70 73 82 90 83 86 83 82 97 86 89 76 83 82 70 76 76
##  [76] 81 97 87 84 74 72 90 90 73 72 75 71 83 94 98 75 73 89 94 87 92 80 90 74 79
```

```r
# Estimates of Transition Probabilities (MLE)

## Under Ho : Stationary
phat_ij <- NULL
for (i in 1:16){
 phat_ij[i] <- sum(n[[i]])/length(chain[1:(time-1),][chain[1:(time-1),]==ceiling(i/4)])
}
```

```
round(phat_ij,3)

##  [1] 0.351 0.544 0.065 0.039 0.202 0.300 0.397 0.101 0.503 0.297 0.200 0.000
## [13] 0.000 0.000 0.704 0.296

## Under Ha : Not Stationary
phat_ijt <- list("phat11"=NULL,"phat12"=NULL,"phat13"=NULL,"phat14"=NULL,
          "phat21"=NULL,"phat22"=NULL,"phat23"=NULL,"phat24"=NULL,
          "phat31"=NULL,"phat32"=NULL,"phat33"=NULL,"phat34"=NULL,
          "phat41"=NULL,"phat42"=NULL,"phat43"=NULL,"phat44"=NULL)
for (t in 1:(time-1)){
  phat_ijt$phat11[t] <- n[[1]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat12[t] <- n[[2]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat13[t] <- n[[3]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat14[t] <- n[[4]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat21[t] <- n[[5]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat22[t] <- n[[6]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat23[t] <- n[[7]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat24[t] <- n[[8]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat31[t] <- n[[9]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat32[t] <- n[[10]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat33[t] <- n[[11]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat34[t] <- n[[12]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat41[t] <- n[[13]][t]/length(chain[t,][chain[t,]==4])
  phat_ijt$phat42[t] <- n[[14]][t]/length(chain[t,][chain[t,]==4])
  phat_ijt$phat43[t] <- n[[15]][t]/length(chain[t,][chain[t,]==4])
  phat_ijt$phat44[t] <- n[[16]][t]/length(chain[t,][chain[t,]==4])
}

round(phat_ijt$phat11,3)

##   [1] 0.364 0.333 0.362 0.357 0.362 0.337 0.340 0.292 0.276 0.355 0.390 0.355
##  [13] 0.344 0.364 0.332 0.408 0.406 0.319 0.367 0.285 0.275 0.375 0.339 0.353
##  [25] 0.357 0.404 0.375 0.366 0.336 0.341 0.407 0.331 0.316 0.409 0.278 0.346
##  [37] 0.374 0.273 0.360 0.388 0.348 0.342 0.378 0.390 0.370 0.388 0.357 0.338
##  [49] 0.379 0.352 0.318 0.376 0.377 0.375 0.355 0.357 0.343 0.272 0.337 0.315
##  [61] 0.374 0.381 0.362 0.374 0.324 0.380 0.377 0.355 0.346 0.323 0.347 0.366
##  [73] 0.287 0.333 0.313 0.372 0.418 0.355 0.349 0.336 0.329 0.385 0.363 0.313
##  [85] 0.317 0.357 0.316 0.362 0.390 0.397 0.287 0.348 0.389 0.376 0.331 0.398
##  [97] 0.329 0.391 0.315 0.332

# Likelihood Ratio
L <- prod(c((phat_ij[1]/phat_ijt$phat11)^n$n11,
       (phat_ij[2]/phat_ijt$phat12)^n$n12,
       (phat_ij[3]/phat_ijt$phat13)^n$n13,
       (phat_ij[4]/phat_ijt$phat14)^n$n14,
       (phat_ij[5]/phat_ijt$phat21)^n$n21,
       (phat_ij[6]/phat_ijt$phat22)^n$n22,
       (phat_ij[7]/phat_ijt$phat23)^n$n23,
       (phat_ij[8]/phat_ijt$phat24)^n$n24,
       (phat_ij[9]/phat_ijt$phat31)^n$n31,
       (phat_ij[10]/phat_ijt$phat32)^n$n32,
       (phat_ij[11]/phat_ijt$phat33)^n$n33,
       (phat_ij[12]/phat_ijt$phat34)^n$n34,
       (phat_ij[13]/phat_ijt$phat41)^n$n41,
```

```
       (phat_ij[14]/phat_ijt$phat42)^n$n42,
       (phat_ij[15]/phat_ijt$phat43)^n$n43,
       (phat_ij[16]/phat_ijt$phat44)^n$n44))

#Chi-Squared Test
chisq <- -2*log(L)
df <- (time-2)*num.states*(num.states-1)
chisq

## [1] 939.0055

qchisq(0.05,df,lower.tail=F)

## [1] 1269.298
```

Stationary process from beginning to t = 90, then random samling of states from t = 91 to end

```
set.seed(403-560)

# P matrix of pij transition probabilities to simulate chains
num.states = 4
P <- t(matrix(c(0.35,0.55,0.06,0.04,
                0.20,0.30,0.40,0.10,
                0.50,0.30,0.20,0.00,
                0.00,0.00,0.70,0.30),nrow=num.states,ncol=num.states))

# Number of Electronic Components (r)
num.comp = 750

# Number of Iterations/Time periods (t)
time = 101 ## t+1 to take account of initial state of component
time_change = 91

# Simulating r Markov Processes
chain <- matrix(NA,ncol=num.comp,nrow=time)
for (c in seq_len(num.comp)){
  chain[,c] <- c(run.mc.sim(P,time_change,num.states),
                 sample(1:4,time-time_change,replace = T))
}

# Markov Chain of the First Component
chain[,1]

##   [1] 2 3 1 2 4 4 3 3 2 4 3 2 2 2 3 1 2 3 1 2 4 3 2 3 1 1 1 1 2 4 3 2 1 2 2 2 2
##  [38] 2 3 1 1 1 4 4 3 1 2 1 1 1 2 3 1 2 3 2 2 3 1 1 3 1 2 1 3 1 2 4 4 3 2 3 1 1
##  [75] 2 3 1 1 2 3 2 2 4 4 3 3 3 1 1 2 4 1 3 2 3 2 2 1 3 4 1

# Number of Observations of each transition (nij)
N <- list()
for (t in 1:(time-1)){
  N[[t]] <- paste(chain[t,],chain[t+1,],sep="")
}
n <- list("n11"=NULL,"n12"=NULL,"n13"=NULL,"n14"=NULL,
          "n21"=NULL,"n22"=NULL,"n23"=NULL,"n24"=NULL,
          "n31"=NULL,"n32"=NULL,"n33"=NULL,"n34"=NULL,
          "n41"=NULL,"n42"=NULL,"n43"=NULL,"n44"=NULL)
```

```r
for (i in 1:(time-1)){
  n$n11[i] <- length(N[[i]][N[[i]]=="11"])
  n$n12[i] <- length(N[[i]][N[[i]]=="12"])
  n$n13[i] <- length(N[[i]][N[[i]]=="13"])
  n$n14[i] <- length(N[[i]][N[[i]]=="14"])
  n$n21[i] <- length(N[[i]][N[[i]]=="21"])
  n$n22[i] <- length(N[[i]][N[[i]]=="22"])
  n$n23[i] <- length(N[[i]][N[[i]]=="23"])
  n$n24[i] <- length(N[[i]][N[[i]]=="24"])
  n$n31[i] <- length(N[[i]][N[[i]]=="31"])
  n$n32[i] <- length(N[[i]][N[[i]]=="32"])
  n$n33[i] <- length(N[[i]][N[[i]]=="33"])
  n$n34[i] <- length(N[[i]][N[[i]]=="34"])
  n$n41[i] <- length(N[[i]][N[[i]]=="41"])
  n$n42[i] <- length(N[[i]][N[[i]]=="42"])
  n$n43[i] <- length(N[[i]][N[[i]]=="43"])
  n$n44[i] <- length(N[[i]][N[[i]]=="44"])
}

n$n11

##   [1]  68  67  81  80  88  92  79  75  64  68  69  79  90  94  91  65  66  76
##  [19]  86  72  65  83  69  71  74  91  89  87  78  69  90  87  74  72  76  81
##  [37]  84  76  74  71  77  75  83  81 105  97  99  88  89  87  83  82  92  87
##  [55] 103  95  88  88  97  84  69  71  83  72  62  93  89  86  91  69  80  90
##  [73]  82  91  84  74  74  67  74  80  87  97  97  85  78  81  80  79  81  78
##  [91]  55  41  41  49  49  51  57  40  44  47

# Estimates of Transition Probabilities (MLE)

## Under Ho : Stationary
phat_ij <- NULL
for (i in 1:16){
 phat_ij[i] <- sum(n[[i]])/length(chain[1:(time-1),][chain[1:(time-1),]==ceiling(i/4)])
}

round(phat_ij,3)

##  [1] 0.339 0.524 0.080 0.057 0.204 0.296 0.388 0.113 0.481 0.290 0.205 0.024
## [13] 0.066 0.071 0.578 0.285

## Under Ha : Not Stationary
phat_ijt <- list("phat11"=NULL,"phat12"=NULL,"phat13"=NULL,"phat14"=NULL,
          "phat21"=NULL,"phat22"=NULL,"phat23"=NULL,"phat24"=NULL,
          "phat31"=NULL,"phat32"=NULL,"phat33"=NULL,"phat34"=NULL,
          "phat41"=NULL,"phat42"=NULL,"phat43"=NULL,"phat44"=NULL)
for (t in 1:(time-1)){
  phat_ijt$phat11[t] <- n[[1]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat12[t] <- n[[2]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat13[t] <- n[[3]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat14[t] <- n[[4]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat21[t] <- n[[5]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat22[t] <- n[[6]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat23[t] <- n[[7]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat24[t] <- n[[8]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat31[t] <- n[[9]][t]/length(chain[t,][chain[t,]==3])
```

```
  phat_ijt$phat32[t] <- n[[10]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat33[t] <- n[[11]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat34[t] <- n[[12]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat41[t] <- n[[13]][t]/length(chain[t,][chain[t,]==4])
  phat_ijt$phat42[t] <- n[[14]][t]/length(chain[t,][chain[t,]==4])
  phat_ijt$phat43[t] <- n[[15]][t]/length(chain[t,][chain[t,]==4])
  phat_ijt$phat44[t] <- n[[16]][t]/length(chain[t,][chain[t,]==4])
}

round(phat_ijt$phat11,3)
```

```
##   [1] 0.343 0.332 0.342 0.339 0.370 0.387 0.326 0.338 0.294 0.322 0.300 0.350
##  [13] 0.385 0.388 0.355 0.278 0.351 0.336 0.368 0.283 0.323 0.358 0.309 0.317
##  [25] 0.354 0.397 0.365 0.347 0.338 0.325 0.396 0.355 0.302 0.348 0.329 0.354
##  [37] 0.354 0.303 0.346 0.332 0.332 0.350 0.353 0.338 0.410 0.385 0.406 0.337
##  [49] 0.359 0.355 0.339 0.343 0.385 0.331 0.435 0.352 0.367 0.386 0.375 0.335
##  [61] 0.319 0.305 0.342 0.305 0.288 0.408 0.366 0.364 0.367 0.297 0.365 0.370
##  [73] 0.324 0.401 0.346 0.315 0.319 0.299 0.368 0.343 0.380 0.404 0.382 0.357
##  [85] 0.321 0.342 0.351 0.338 0.333 0.332 0.240 0.230 0.225 0.272 0.255 0.255
##  [97] 0.282 0.222 0.246 0.261
```

```
# Likelihood Ratio
L <- prod(c((phat_ij[1]/phat_ijt$phat11)^n$n11,
       (phat_ij[2]/phat_ijt$phat12)^n$n12,
       (phat_ij[3]/phat_ijt$phat13)^n$n13,
       (phat_ij[4]/phat_ijt$phat14)^n$n14,
       (phat_ij[5]/phat_ijt$phat21)^n$n21,
       (phat_ij[6]/phat_ijt$phat22)^n$n22,
       (phat_ij[7]/phat_ijt$phat23)^n$n23,
       (phat_ij[8]/phat_ijt$phat24)^n$n24,
       (phat_ij[9]/phat_ijt$phat31)^n$n31,
       (phat_ij[10]/phat_ijt$phat32)^n$n32,
       (phat_ij[11]/phat_ijt$phat33)^n$n33,
       (phat_ij[12]/phat_ijt$phat34)^n$n34,
       (phat_ij[13]/phat_ijt$phat41)^n$n41,
       (phat_ij[14]/phat_ijt$phat42)^n$n42,
       (phat_ij[15]/phat_ijt$phat43)^n$n43,
       (phat_ij[16]/phat_ijt$phat44)^n$n44))

#Chi-Squared Test
chisq <- -2*log(L)
df <- (time-2)*num.states*(num.states-1)
chisq
```

```
## [1] Inf
```

```
qchisq(0.05,df,lower.tail=F)
```

```
## [1] 1269.298
```

Stationary process of P1 Transition Matrix from beginning to t = 90, then stationary process of P2 Transition Matrix from t = 91 to end

```
set.seed(403-560)

# P matrix of pij transition probabilities to simulate chains
```

```r
num.states = 4
P1 <- t(matrix(c(0.35,0.55,0.06,0.04,
                 0.20,0.30,0.40,0.10,
                 0.50,0.30,0.20,0.00,
                 0.00,0.00,0.70,0.30),nrow=num.states,ncol=num.states))
P2 <- t(matrix(c(0.34,0.56,0.05,0.05,
                 0.21,0.29,0.39,0.11,
                 0.49,0.31,0.19,0.01,
                 0.01,0.01,0.69,0.29),nrow=num.states,ncol=num.states))

# Number of Electronic Components (r)
num.comp = 750

# Number of Iterations/Time periods (t)
time = 101 ## t+1 to take account of initial state of component
time_change = 91

# Simulating r Markov Processes
chain <- matrix(NA,ncol=num.comp,nrow=time)
for (c in seq_len(num.comp)){
  chain[,c] <- c(run.mc.sim(P1,time_change,num.states),
                 run.mc.sim(P2,time-time_change,num.states))
}

# Markov Chain of the First Component
chain[,1]
```

```
##   [1] 2 3 1 2 4 4 3 3 2 4 3 2 2 2 3 1 2 3 1 2 4 3 2 3 1 1 1 1 2 4 3 2 1 2 2 2 2
##  [38] 2 3 1 1 1 4 4 3 1 2 1 1 1 2 3 1 2 3 2 2 3 1 1 3 1 2 1 3 1 2 4 4 3 2 3 1 1
##  [75] 2 3 1 1 2 3 2 2 4 4 3 3 3 1 1 2 4 1 2 3 1 1 1 1 1 1 2
```

```r
# Number of Observations of each transition (nij)
N <- list()
for (t in 1:(time-1)){
  N[[t]] <- paste(chain[t,],chain[t+1,],sep="")
}
n <- list("n11"=NULL,"n12"=NULL,"n13"=NULL,"n14"=NULL,
          "n21"=NULL,"n22"=NULL,"n23"=NULL,"n24"=NULL,
          "n31"=NULL,"n32"=NULL,"n33"=NULL,"n34"=NULL,
          "n41"=NULL,"n42"=NULL,"n43"=NULL,"n44"=NULL)
for (i in 1:(time-1)){
  n$n11[i] <- length(N[[i]][N[[i]]=="11"])
  n$n12[i] <- length(N[[i]][N[[i]]=="12"])
  n$n13[i] <- length(N[[i]][N[[i]]=="13"])
  n$n14[i] <- length(N[[i]][N[[i]]=="14"])
  n$n21[i] <- length(N[[i]][N[[i]]=="21"])
  n$n22[i] <- length(N[[i]][N[[i]]=="22"])
  n$n23[i] <- length(N[[i]][N[[i]]=="23"])
  n$n24[i] <- length(N[[i]][N[[i]]=="24"])
  n$n31[i] <- length(N[[i]][N[[i]]=="31"])
  n$n32[i] <- length(N[[i]][N[[i]]=="32"])
  n$n33[i] <- length(N[[i]][N[[i]]=="33"])
  n$n34[i] <- length(N[[i]][N[[i]]=="34"])
  n$n41[i] <- length(N[[i]][N[[i]]=="41"])
  n$n42[i] <- length(N[[i]][N[[i]]=="42"])
```

```
  n$n43[i] <- length(N[[i]][N[[i]]=="43"])
  n$n44[i] <- length(N[[i]][N[[i]]=="44"])
}

n$n11
```

```
##   [1]  72  74  91  75  74  88  72  77  71  75  87  87  76  80  87  85  78  85
##  [19]  85 102  91  77  92  92  80  76  85  90  83 102  92  99  78  82  87 100
##  [37]  95  89  87  86  84 101  79  89  75  92  90  80  86  83  87  79  78  75
##  [55]  64  74  74  78  85  76  85 101  85  92  78  72  82  87  88  76  80  85
##  [73]  78  81  73  83  72  75  85  96  84  85  80  81  81  79  72  78  79  89
##  [91]  72  59  51  68  79  84  86  86  79  77
```

```
# Estimates of Transition Probabilities (MLE)

## Under Ho : Stationary
phat_ij <- NULL
for (i in 1:16){
 phat_ij[i] <- sum(n[[i]])/length(chain[1:(time-1),][chain[1:(time-1),]==ceiling(i/4)])
}

round(phat_ij,3)
```

```
##  [1] 0.351 0.542 0.064 0.042 0.204 0.298 0.396 0.102 0.498 0.298 0.199 0.004
## [13] 0.004 0.004 0.692 0.301
```

```
## Under Ha : Not Stationary
phat_ijt <- list("phat11"=NULL,"phat12"=NULL,"phat13"=NULL,"phat14"=NULL,
          "phat21"=NULL,"phat22"=NULL,"phat23"=NULL,"phat24"=NULL,
          "phat31"=NULL,"phat32"=NULL,"phat33"=NULL,"phat34"=NULL,
          "phat41"=NULL,"phat42"=NULL,"phat43"=NULL,"phat44"=NULL)
for (t in 1:(time-1)){
  phat_ijt$phat11[t] <- n[[1]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat12[t] <- n[[2]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat13[t] <- n[[3]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat14[t] <- n[[4]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat21[t] <- n[[5]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat22[t] <- n[[6]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat23[t] <- n[[7]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat24[t] <- n[[8]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat31[t] <- n[[9]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat32[t] <- n[[10]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat33[t] <- n[[11]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat34[t] <- n[[12]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat41[t] <- n[[13]][t]/length(chain[t,][chain[t,]==4])
  phat_ijt$phat42[t] <- n[[14]][t]/length(chain[t,][chain[t,]==4])
  phat_ijt$phat43[t] <- n[[15]][t]/length(chain[t,][chain[t,]==4])
  phat_ijt$phat44[t] <- n[[16]][t]/length(chain[t,][chain[t,]==4])
}

round(phat_ijt$phat11,3)
```

```
##   [1] 0.371 0.372 0.370 0.304 0.326 0.393 0.301 0.317 0.327 0.315 0.358 0.331
##  [13] 0.326 0.359 0.349 0.376 0.348 0.360 0.341 0.392 0.373 0.330 0.400 0.364
##  [25] 0.357 0.347 0.363 0.370 0.356 0.410 0.352 0.414 0.307 0.383 0.385 0.402
##  [37] 0.377 0.363 0.373 0.360 0.368 0.401 0.310 0.371 0.316 0.391 0.366 0.327
```

```
##  [49] 0.389 0.343 0.337 0.366 0.325 0.338 0.294 0.333 0.338 0.348 0.354 0.353
##  [61] 0.357 0.409 0.335 0.390 0.345 0.319 0.361 0.378 0.355 0.321 0.364 0.340
##  [73] 0.344 0.343 0.327 0.356 0.309 0.328 0.365 0.398 0.351 0.348 0.372 0.335
##  [85] 0.346 0.341 0.338 0.351 0.339 0.372 0.306 0.295 0.300 0.289 0.359 0.337
##  [97] 0.363 0.354 0.348 0.342
```

```
# Likelihood Ratio
L <- prod(c((phat_ij[1]/phat_ijt$phat11)^n$n11,
       (phat_ij[2]/phat_ijt$phat12)^n$n12,
       (phat_ij[3]/phat_ijt$phat13)^n$n13,
       (phat_ij[4]/phat_ijt$phat14)^n$n14,
       (phat_ij[5]/phat_ijt$phat21)^n$n21,
       (phat_ij[6]/phat_ijt$phat22)^n$n22,
       (phat_ij[7]/phat_ijt$phat23)^n$n23,
       (phat_ij[8]/phat_ijt$phat24)^n$n24,
       (phat_ij[9]/phat_ijt$phat31)^n$n31,
       (phat_ij[10]/phat_ijt$phat32)^n$n32,
       (phat_ij[11]/phat_ijt$phat33)^n$n33,
       (phat_ij[12]/phat_ijt$phat34)^n$n34,
       (phat_ij[13]/phat_ijt$phat41)^n$n41,
       (phat_ij[14]/phat_ijt$phat42)^n$n42,
       (phat_ij[15]/phat_ijt$phat43)^n$n43,
       (phat_ij[16]/phat_ijt$phat44)^n$n44))

#Chi-Squared Test
chisq <- -2*log(L)
df <- (time-2)*num.states*(num.states-1)
chisq
```

```
## [1] Inf
```

```
qchisq(0.05,df,lower.tail=F)
```

```
## [1] 1269.298
```

Stationary process of P1 Transition Matrix from beginning to t = 90, then stationary process of P2 Transition Matrix from t = 91 to end (r = 150)

```
set.seed(403-560)

# P matrix of pij transition probabilities to simulate chains
num.states = 4
P1 <- t(matrix(c(0.35,0.55,0.06,0.04,
                 0.20,0.30,0.40,0.10,
                 0.50,0.30,0.20,0.00,
                 0.00,0.00,0.70,0.30),nrow=num.states,ncol=num.states))
P2 <- t(matrix(c(0.34,0.56,0.05,0.05,
                 0.21,0.29,0.39,0.11,
                 0.49,0.31,0.19,0.01,
                 0.01,0.01,0.69,0.29),nrow=num.states,ncol=num.states))

# Number of Electronic Components (r)
num.comp = 150

# Number of Iterations/Time periods (t)
time = 101 ## t+1 to take account of initial state of component
```

```r
time_change = 91

# Simulating r Markov Processes
chain <- matrix(NA,ncol=num.comp,nrow=time)
for (c in seq_len(num.comp)){
  chain[,c] <- c(run.mc.sim(P1,time_change,num.states),
                 run.mc.sim(P2,time-time_change,num.states))
}

# Markov Chain of the First Component
chain[,1]
```

```
##   [1] 2 3 1 2 4 4 3 3 2 4 3 2 2 2 3 1 2 3 1 2 4 3 2 3 1 1 1 1 2 4 3 2 1 2 2 2 2
##  [38] 2 3 1 1 1 4 4 3 1 2 1 1 1 2 3 1 2 3 2 2 3 1 1 3 1 2 1 3 1 2 4 4 3 2 3 1 1
##  [75] 2 3 1 1 2 3 2 2 4 4 3 3 3 1 1 2 4 1 2 3 1 1 1 1 1 1 2
```

```r
# Number of Observations of each transition (nij)
N <- list()
for (t in 1:(time-1)){
  N[[t]] <- paste(chain[t,],chain[t+1,],sep="")
}
n <- list("n11"=NULL,"n12"=NULL,"n13"=NULL,"n14"=NULL,
          "n21"=NULL,"n22"=NULL,"n23"=NULL,"n24"=NULL,
          "n31"=NULL,"n32"=NULL,"n33"=NULL,"n34"=NULL,
          "n41"=NULL,"n42"=NULL,"n43"=NULL,"n44"=NULL)
for (i in 1:(time-1)){
  n$n11[i] <- length(N[[i]][N[[i]]=="11"])
  n$n12[i] <- length(N[[i]][N[[i]]=="12"])
  n$n13[i] <- length(N[[i]][N[[i]]=="13"])
  n$n14[i] <- length(N[[i]][N[[i]]=="14"])
  n$n21[i] <- length(N[[i]][N[[i]]=="21"])
  n$n22[i] <- length(N[[i]][N[[i]]=="22"])
  n$n23[i] <- length(N[[i]][N[[i]]=="23"])
  n$n24[i] <- length(N[[i]][N[[i]]=="24"])
  n$n31[i] <- length(N[[i]][N[[i]]=="31"])
  n$n32[i] <- length(N[[i]][N[[i]]=="32"])
  n$n33[i] <- length(N[[i]][N[[i]]=="33"])
  n$n34[i] <- length(N[[i]][N[[i]]=="34"])
  n$n41[i] <- length(N[[i]][N[[i]]=="41"])
  n$n42[i] <- length(N[[i]][N[[i]]=="42"])
  n$n43[i] <- length(N[[i]][N[[i]]=="43"])
  n$n44[i] <- length(N[[i]][N[[i]]=="44"])
}

n$n11
```

```
##   [1] 15 11 11 13 13 15 16 15 16 13 14 18 12 15 20 16 11 14 18 21 17 21 23 16 16
##  [26] 16 18 23 17 23 16 21 13 17 19 20 21 16 12 17 20 28 10 14 14 21 17 17 17 15
##  [51] 20 21 19 10 11 17 12 13 17 18 16 23 19 15 12 12 15 18 16 13 13 18 17 18 11
##  [76] 21 17 21 27 20 16 13 15 19 18 17 12 13 16 28 19 12 11 15 12 18 20 27 24 16
```

```r
# Estimates of Transition Probabilities (MLE)

## Under Ho : Stationary
phat_ij <- NULL
for (i in 1:16){
```

```
 phat_ij[i] <- sum(n[[i]])/length(chain[1:(time-1),][chain[1:(time-1),]==ceiling(i/4)])
}

round(phat_ij,3)

##  [1] 0.357 0.538 0.066 0.039 0.196 0.294 0.405 0.105 0.494 0.291 0.210 0.005
## [13] 0.005 0.005 0.687 0.304

## Under Ha : Not Stationary
phat_ijt <- list("phat11"=NULL,"phat12"=NULL,"phat13"=NULL,"phat14"=NULL,
          "phat21"=NULL,"phat22"=NULL,"phat23"=NULL,"phat24"=NULL,
          "phat31"=NULL,"phat32"=NULL,"phat33"=NULL,"phat34"=NULL,
          "phat41"=NULL,"phat42"=NULL,"phat43"=NULL,"phat44"=NULL)
for (t in 1:(time-1)){
  phat_ijt$phat11[t] <- n[[1]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat12[t] <- n[[2]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat13[t] <- n[[3]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat14[t] <- n[[4]][t]/length(chain[t,][chain[t,]==1])
  phat_ijt$phat21[t] <- n[[5]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat22[t] <- n[[6]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat23[t] <- n[[7]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat24[t] <- n[[8]][t]/length(chain[t,][chain[t,]==2])
  phat_ijt$phat31[t] <- n[[9]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat32[t] <- n[[10]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat33[t] <- n[[11]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat34[t] <- n[[12]][t]/length(chain[t,][chain[t,]==3])
  phat_ijt$phat41[t] <- n[[13]][t]/length(chain[t,][chain[t,]==4])
  phat_ijt$phat42[t] <- n[[14]][t]/length(chain[t,][chain[t,]==4])
  phat_ijt$phat43[t] <- n[[15]][t]/length(chain[t,][chain[t,]==4])
  phat_ijt$phat44[t] <- n[[16]][t]/length(chain[t,][chain[t,]==4])
}

round(phat_ijt$phat11,3)

##   [1] 0.366 0.262 0.268 0.289 0.302 0.385 0.340 0.366 0.320 0.228 0.298 0.327
##  [13] 0.240 0.455 0.400 0.302 0.289 0.341 0.360 0.389 0.370 0.512 0.426 0.296
##  [25] 0.372 0.327 0.367 0.469 0.378 0.442 0.327 0.375 0.265 0.362 0.463 0.426
##  [37] 0.420 0.327 0.245 0.447 0.385 0.519 0.169 0.378 0.341 0.420 0.405 0.315
##  [49] 0.370 0.333 0.370 0.438 0.365 0.256 0.282 0.405 0.261 0.333 0.405 0.391
##  [61] 0.327 0.442 0.365 0.300 0.324 0.293 0.312 0.429 0.314 0.325 0.317 0.367
##  [73] 0.354 0.375 0.268 0.438 0.327 0.438 0.482 0.370 0.348 0.295 0.326 0.432
##  [85] 0.316 0.447 0.267 0.310 0.372 0.500 0.333 0.279 0.314 0.349 0.240 0.474
##  [97] 0.392 0.519 0.453 0.314

# Likelihood Ratio
L <- prod(c((phat_ij[1]/phat_ijt$phat11)^n$n11,
       (phat_ij[2]/phat_ijt$phat12)^n$n12,
       (phat_ij[3]/phat_ijt$phat13)^n$n13,
       (phat_ij[4]/phat_ijt$phat14)^n$n14,
       (phat_ij[5]/phat_ijt$phat21)^n$n21,
       (phat_ij[6]/phat_ijt$phat22)^n$n22,
       (phat_ij[7]/phat_ijt$phat23)^n$n23,
       (phat_ij[8]/phat_ijt$phat24)^n$n24,
       (phat_ij[9]/phat_ijt$phat31)^n$n31,
       (phat_ij[10]/phat_ijt$phat32)^n$n32,
       (phat_ij[11]/phat_ijt$phat33)^n$n33,
```

```
       (phat_ij[12]/phat_ijt$phat34)^n$n34,
       (phat_ij[13]/phat_ijt$phat41)^n$n41,
       (phat_ij[14]/phat_ijt$phat42)^n$n42,
       (phat_ij[15]/phat_ijt$phat43)^n$n43,
       (phat_ij[16]/phat_ijt$phat44)^n$n44))

#Chi-Squared Test
chisq <- -2*log(L)
df <- (time-2)*num.states*(num.states-1)
chisq

## [1] 1187.706

qchisq(0.05,df,lower.tail=F)

## [1] 1269.298
```