

Dyadic pattern analysis using longitudinal Actor-Partner Interdependence Model with Markov chains for unique case analysis

Mégane Bollenrucher^a  , Joëlle Darwiche^a  & Jean-Philippe Antonietti^a 

^aUniversity of Lausanne, Switzerland

Abstract ■ Understanding the dynamics of interactions between two individuals requires special conceptual and statistical models. The Actor-Partner Interdependence Model (APIM) is the classic conceptual framework for standard dyadic designs, capturing the interdependence between dyad members by identifying the mechanisms of interaction through actor and partner effects. To analyze the temporal dynamics of dyadic interactions, the longitudinal APIM extends the classic model, often employing categorical variables to capture behavior. To analyze such data considering its categorical nature, specific statistical models are required. Markov chain is a powerful approach considering the longitudinal and categorical aspects of the data. This article describes how to adapt Markov chains in the categorical longitudinal dyadic case. It additionally offers a tutorial to model and identify the pattern of interaction using this method for unique case approach to maintain a simple and focused level of analysis. Codes in R language are provided.

Keywords ■ Dyadic intensive longitudinal categorical data, actor-partner interdependence model, Markov chains. **Tools** ■ R.

Acting Editor ■ Denis Cousineau (Université d'Ottawa)

Reviewers
■ One anonymous reviewer

 megane.bollenrucher@unil.ch

 [10.20982/tqmp.19.3.p230](https://doi.org/10.20982/tqmp.19.3.p230)

Introduction

The Actor-Partner Interdependence Model (APIM) has emerged as a valuable tool for understanding interpersonal interactions (Kenny et al., 2006). The model provides a framework for analyzing the mutual influence between individuals within dyadic relationships. By considering both actor effects, meaning the influence of an individual on their own outcome and partner effects, meaning the influence of one individual on their partner's outcome, the APIM offers insights into the dynamics and complexities of social interactions. Extending the APIM to longitudinal studies provides a more comprehensive understanding of how interactions unfold over time. The classic APIM primarily focuses on cross-sectional data, limiting the ability to capture dynamic processes and changes occurring within dyads. The longitudinal APIM examines how actor and partner effects evolve to identify temporal patterns and gain information about the stability and fluctuations of dynamics (Laurenceau & Bolger, 2011). When dealing with

longitudinal data that involve categorical variables, such as behavioral states or discrete outcomes, employing appropriate statistical techniques becomes crucial. For example, if we want to assess the evolution of the quality of a couple's sexual relationship, we can ask them on a daily basis a question such as *Did you have sexual desire yesterday?* The categorical *yes* or *no* answer of each partner can be followed over time. Markov chains provide a powerful framework for modeling categorical longitudinal data within the APIM. By characterizing transitions between discrete states and capturing the inherent dependencies among observations, Markov chain models offer a flexible and insightful approach to the analysis of longitudinal APIM data. The objective of this article is to provide a tutorial which shows how to use Markov chains in the context of longitudinal APIM for only one dyad to consider the simplest unit of analysis. By incorporating Markov chain modeling into the APIM, we seek to enhance our understanding of the dynamic processes occurring within dyads over time. A theoretical framework of Markov chains is first presented to

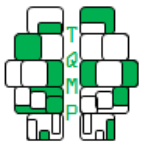
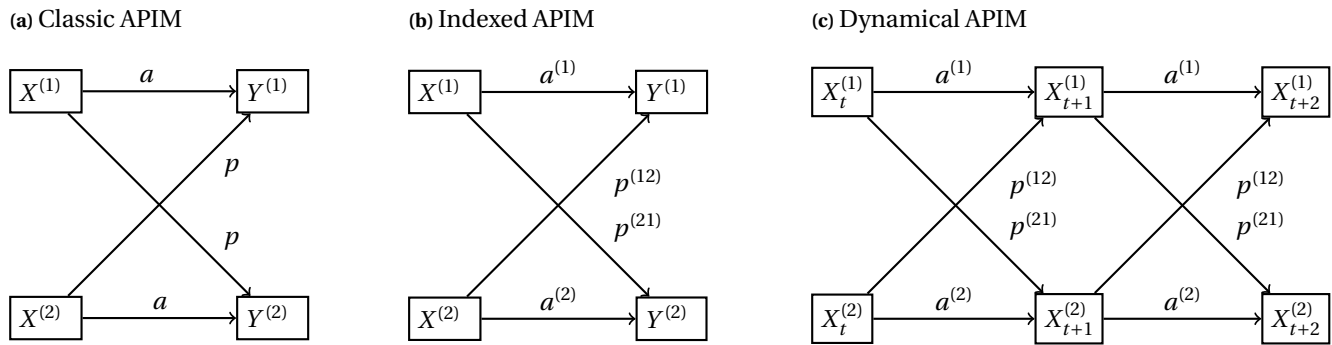


Figure 1 ■ Classic and dynamic actor-partner interdependence model



show how to adapt them in the context of dyad design. Then, the tutorial offers step-by-step guidance on how to apply this novel method to analyze dyadic data conceptually modeled by APIM.

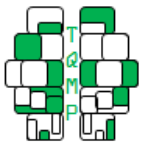
The classic Actor-Partner Interdependence Model

In psychology, understanding interpersonal processes, specifically dyadic interactions, is critical (Kenny & Kashy, 2013). Dyads could be couples, parent-child or siblings. Data about dyadic interaction is collected through questionnaires or observations. Analyzing this data allows researchers to study the multilevel structure of interactions, focusing on individual or dyadic units (Gonzalez & Griffin, 2012). The dyadic data is based on the assumption of shared elements between dyad members. The aim is to account for the interdependence between the members of each pair (Fitzpatrick et al., 2016). This paper uses the APIM to understand these interactions.

The APIM may be defined as a conceptual model of interpersonal phenomena that accounts for the lack of independence observed with pairs of individuals (Fitzpatrick et al., 2016). The model allows the analysis of interpersonal processes by assuming actor effects and partner effects from the point of view of each member of the dyad. We differentiate between individuals within dyads by randomly assigning them as either the first or second partner. When the first partner's point of view is adopted, their actions are considered as the actor effect, whereas those of the second partner represent the partner effect. The situation is reversed when we examine the second partner's point of view. This distinction allows us to consider the unique roles and contributions of each partner within the dyadic relationship. The original APIM, shown on the left plot of Figure 1(a), was developed from a static point of view using numerical scores for the variables of interest (Kenny & Kashy, 2013). The aim is to analyze the extent to which the independent variable (X) of a person influences their score on the dependent variable (Y), but also

their partner's score. For our example, when analyzing the effect of a woman's lack of sexual desire on the quality of the romantic relationship, the woman's perception of the romantic relationship quality is the actor effect, whereas the man's perception of the romantic relationship quality is the partner effect. Extensions of the classic APIM are possible (Kenny et al., 2006). One adaptation is to allow the actor or partner effect to be different between the members of the dyad. Conceptually, it implies that the strength of the influence of one partner on their variable or on their partner's may be different from one individual to the other. This means indexing the actor effect (1) and the partner effect (2) with respect to the member involved. Figure 1(b) represents this extension. The parameters to be estimated for the APIM are the actor and partner effects, as well as the interaction effects. These parameters are estimated using different statistical methods. The estimation of these parameters allows us to understand how the interaction takes place between the two members of the dyad.

Understanding the link between the answers of both partners is the reason APIM is used to model interaction in the case of dyadic data. Using the combination of actor and partner effects, dyadic patterns are created (Fitzpatrick et al., 2016). The first pattern, called *couple oriented*, is represented by $a = p$. In this pattern, we consider that the score of a dyad member is as much influenced by the actor effect as it is by the partner effect. In other words, we assume that the actor effect is the same as the effect of the partner. The second pattern, called the *contrast pattern* or sometimes the *social comparisons pattern*, is represented by $a + p = 0$ or by $a = -p$. This situation is the opposite of the previous one. We assume that the effect of one member on their partner's score is the opposite of the effect of the other partner on their score. Both patterns can be regrouped under the *actor-partner pattern*. The third pattern is called the *actor pattern*. In this situation, it is assumed that only the individual's score on the independent variable has an influence on their score on the dependent variable. The fourth



pattern is called the *partner pattern*. As before, this is the opposite situation in which one member's score is only associated with the score of their partner on the dependent variable. These situations are summarized as $a \neq 0, p = 0$ and $a = 0, p \neq 0$, respectively. Obviously, it is also possible to have a situation without any pattern with both effects equal to 0. This can be called the *independence pattern*. The patterns allow the analysis of dyad members' respective influence on the dependent variable.

Categorical longitudinal Actor-Partner Interdependence Model

The aim of capturing a time dynamic in dyadic interactions is achieved by using dynamic systems analysis, which considers the temporal evolution of behavioral sequences over discrete time points, in contrast to static models that do not study the temporal dynamics of variables (Estrada et al., 2020; van Rijn, 2008; Kendall & Comer, 2014). The use of sequences is justified to achieve greater ecological validity, which can be defined as a measure of how well a model can predict real-life behavior (Walls et al., 2006). The use of sequences implies another adaptation of the classical APIM model. It transposes the model in the dynamic world. It happens when we consider measurement repeatedly over time (De Haan-Rietdijk et al., 2016). In our example on sexual desire, to understand how it evolves over time, the question needs to be asked over a defined period, which can be relatively short, such as 10 days or long, like 3 months. The answers to this question then form what we refer to as a *sequence*. Longitudinal APIM can be represented as in Figure 1(c). In the case of intensive longitudinal data, the behavior is analyzed over time (t). The score of the variable at a specific time for one member of the dyad may be influenced either by their own past scores on this variable or by their partner's past scores (Fuchs et al., 2017). This longitudinal extension creates a nonindependence between observations (Mair, 2018). The classic APIM and both the indexation and the longitudinal modification can be applied to the usual numerical scores used in this framework, but also to categorical variables.

Observational codes and questionnaires often use response scales, resulting in categorical data. In this article, the focus is on categorical data. An example of answers for both partners is provided in the Appendix to show how the sequences are represented. Analyzing the data as categorical is important to maintain fidelity to the nature of the data and avoid potentially erroneous numerical transformations. The most prevalent statistical model for intensive longitudinal dyadic data is the logistic regression (Kenny et al., 2006). The aim is to use the logit model in the framework of sequential analysis to create a model based on the time variable instead of the dyad. In this model, the answer

at $t - 1$ is used as the independent variable and the answer at t as the dependent variable. The aim is to estimate at the same time the effect of each partner's answer at time $t - 1$ on their answer at time t , but also the effect of interaction. The major limitation is that the logistic regression model is only possible on binary outcomes. For our example, if the possible answers are *yes* or *no*, the logit model can be applied. However, if the answers are *yes*, *no*, *unsure*, the model can no longer be applied because the possible outcomes are no longer dichotomous. Fuchs et al. (2017) mentioned methods other than logistic regression, including multilevel models and Markov chains, for modeling longitudinal APIM. Although they elaborate on the use of logistic regression and multilevel models, the Markovian approach is only mentioned. For this tutorial, we focus specifically on Markov chains to demonstrate its relevance in the context of longitudinal APIM.

Objectives

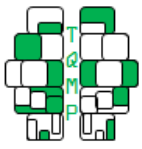
To stay at the simplest modeling level, we propose a unique case analysis. This perspective is often adopted in psychology to develop and evaluate a method with only one unit of analysis (Kazdin, 2012). The objective is to focus on one dyad and to model the interaction and to understand what the pattern of each member is.

The first section of the article details the application of Markov chains to dyadic sequences, whereas the second section concentrates on the identification of patterns. We use the evolution of a couple's sexual desire as the reference example for our work. Each part, modeling, estimation and identification of interaction patterns, is illustrated with R code. We present the case with two possible answers in the text and the case with three possible answers in the Appendix. Moreover, comments about the functions and the code are also available in the Appendix to follow the work discussed.

Actor-Partner Interdependence Model using Markov chains

The use of the APIM allows the identification of interaction pattern, but before this can be done, the interaction must be successfully modeled. To do so, we propose the use of Markov chains, which are a classical approach to model categorical sequences measured with a discrete time. Markov chains represent the sequences as close as possible to the data because no assumption is made about the distribution of the data and no latent variable is used to model the interaction.

In its classic version, Markov chains deal with a categorical temporal sequence. Therefore, to use them in the dyadic case, transformations must be performed. Indeed, the dependence between both members of the dyad must



be considered. To understand how Markov chains integrate the dyadic structure of sequences, we present the classic Markov chains before showing how to transform them for the case in question. For classical Markov chains, we propose a step-by-step example in the Appendix. For dyadic Markov chains, we integrate R code that allows following and reproducing the analysis.

Classic Markov chains

Behavioral sequences collected at fixed intervals using categorical scales can be modeled using Markov chains (Ibe, 2013a). They are dependent of three elements: the states space, the transition matrix and the initial state (Ibe, 2013c). The states space corresponds to the number of possible levels of the scales used to collect the data. In the example proposed before, there are two states, namely *yes* and *no*. In the case of a Likert scale with the following possible answers: *strongly agree, agree, neutral, disagree, strongly disagree*, there are five states. The initial state is the state at the first measurement point. The sequence has a fixed length. At each measuring point, the state in which the individual is found is noted. The changes of state as the chain progresses are called transitions. The total number of transitions is the length of the chain minus 1. The number of transitions between each state can be summarized in a count matrix. Each transition happens with a certain probability, called transition probability, which is also expressed in a matrix. In the next part, the following abbreviations hold: *S* for the state space, *s* for any state, *s₀* for the initial state, *T* for the length of the chain, *t* for any time measurement, *t₀* for the first time measurement, *C* for the count matrix and *TP* for the transition probabilities matrix.

Markov chains are based on the assumption that we can determine the number of steps that we need to do in the past to model the future (Ibe, 2013b). The number of steps that must be done give the order of the chain. For example, if the chain is of order 2, the last two states must be considered to model the next future state. In this work, we assume that we can work with an order 1 chain, meaning that we use the state at time *t* to model the state at time *t* + 1. Therefore, the transition probability may be written as a conditional probability: *P*(state at time *t* + 1 | state at time *t*). In a simplified way, we write it as *p_{ji}* with *i* being the state at time *t* and *j* the state at time *t* + 1. Because at any time measurement the individual may be in any state, the transition matrix is of size (*S* × *S*). For our example, *S* = 2 and the size of *TP* is (2 × 2). Given that the states are mutually exclusive and collectively exhaustive, meaning that the individual necessarily ends up in one and only one state, each row of *TP* sums to 1. *TP* is a squared matrix with the same number of states at time *t* than at time *t* + 1. Recoding *yes* as 1 and *no* as 2, the matrix of the example may be

represented as follows:

$$TP = \begin{matrix} & \begin{matrix} 1 & 2 \end{matrix} \\ \begin{matrix} 1 \\ 2 \end{matrix} & \begin{pmatrix} p_{1|1} & p_{2|1} \\ p_{1|2} & p_{2|2} \end{pmatrix} \end{matrix}$$

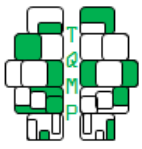
The matrix can be read either by rows or by columns. Each line of the matrix represents the probabilities of ending in the different possible states at time *t* + 1 knowing that we are in a state determined at time *t*. In each column, we find the likelihood of ending in a fixed state at time *t* + 1 knowing that we were in the different possible states at time *t* + 1. Usually, the matrix is read row by row.

The parameters to be estimated are the transition probabilities. To estimate them, we construct the count matrix, *C*. The elements of this matrix are *c_{ji}*, which denotes the number of times the sequence goes from any state *i* to *j*. We use the maximum likelihood to estimate the transition probabilities, *p̂_{ji}*. Once we have estimated all the transitions, we can construct the estimated transition probabilities matrix, which summarizes the behavior of the individual. An illustration of the construction of the count matrix and the transition probabilities matrix is done in the Appendix for the sequence of the woman.

Dyadic Markov chains

The classic Markov chain approach allows to model one sequence at a time. In the case of a dyadic design, two sequences must be modeled at the same time considering that they are dependent. Therefore, the aim is to synthesize the dynamics of change and the dependence between both members of the dyad. As presented for the case of an individual, we model the Markov chains with one matrix per individual and one matrix per couple. Modeling at the level of the couple allows a summarized vision of the couple's dynamic, but it does not allow an analysis of the individual behaviors, which is why modeling at the individual level is more interesting and allows us to understand the effect of the behavior of both the individual and their partner.

At the individual level, in the case of a dyadic design, the possible states to model the state in which an individual will end up are all possible combinations of states of this individual and their partner. In other words, if we model the behavior of only one individual considering their partner, we must consider their state and their partner's state at time *t* to model only their state at time *t* + 1. In the case of our example, the state space has length 4: (*yes, yes*), (*yes, no*), (*no, yes*), (*no, no*). In this notation, the first element is the state of the individual and the second is their partner's state. Considering that the dyad has two members, each member is successively the first member. In other words, when constructing the woman's matrix, the first element of the parenthesis is the woman's state and the second is the



Listing 1 ■ Libraries, functions and data

```
source("FunctionsDyadicMarkov.R")
load("APM.RData")
str(APM)
## 'data.frame': 90 obs. of 2 variables:
## $ chain1: num 2 2 1 1 2 1 1 1 2 2 ...
## $ chain2: num 2 2 2 1 2 2 1 1 1 1 ...

S <- 2
apmFM <- APM$chain1
apmSM <- APM$chain2
```

man's. The opposite situation arises when we model the man's chain. Consequently, two matrices exist, one per individual. Obviously, the individual will end up in only one of the states. Therefore, the state space at time $t + 1$ has length 2: *yes, no*. This set up changes the structure and the size of the count and the transition probabilities matrices. Indeed, it should consider all possible combinations of input states. Consequently, the matrix becomes rectangular and the size of TP is $(S^2 \times S)$, where S^2 is the combinations. For the example, the matrix size is $(2^2 \times 2)$ and appears as follows:

$$TP = \begin{matrix} & \begin{matrix} 1 & 2 \end{matrix} \\ \begin{matrix} (1, 1) \\ (1, 2) \\ (2, 1) \\ (2, 2) \end{matrix} & \begin{pmatrix} p_{1|11} & p_{2|11} \\ p_{1|12} & p_{2|12} \\ p_{1|21} & p_{2|21} \\ p_{1|22} & p_{2|22} \end{pmatrix} \end{matrix}$$

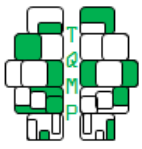
This represents the matrix for one individual. Each element of the matrix may be written as $p_{j|m}$, with m being any possible combination of states. As for classic Markov chains, the parameters to be estimated are the transition probabilities. The procedure is the same as described previously: Construct the count matrix for each individual considering the dyadic structure of the data and estimate the transition probabilities using the maximum likelihood estimation.

We exemplify the modeling of dyadic sequences by Markov chains using simulated data for the four main patterns of interaction, namely the actor-partner model (APM), actor-only model (AM), partner-only model (PM) and independence model (IM). We detail in the text the general case of the actor-partner for $S = 2$. The other three patterns of interaction are developed in the Appendix. R code vignettes are used to expose the steps of the modeling. We comment on the lines of code and outputs. In addition, more specific comments on the functions used are provided in the Appendix. The sequences were simulated with $S = 2$ and $T = 90$, which correspond to 3 months. Details on the simulation are also available in the comment

of Listing 1 in the Appendix. `FunctionsDyadicMarkov.R` is a collection of functions available on the journal's web site; `APM.RData` is a simulated database containing chains for two individuals in columns. We take `chain1` as the sequence of the first partner and `chain2` as the sequence of the second partner. For our example, the first partner is the woman and the second is the man. As explained before, each element of the step-by-step procedure is done from the point of view of each partner successively.

To obtain the transition probabilities matrix, we must first extract the transition count matrix and estimate the probabilities matrix using the maximum likelihood. We show how to obtain the count and the transition probabilities matrix in Listing 2. The count individual matrices are called `empMatFM` and `empMatSM` for the first and the second member, respectively. The lines of the matrices correspond to the S^2 possible combinations, namely $(1, 1), (1, 2), (2, 1), (2, 2)$. To check whether the count matrix is correct, we can sum all the transitions. We obtain the number of measurement points minus 1, $T - 1$, which is correct. By briefly analysing the count matrices, we find that the first member of this dyad finishes almost the same number of times in both states, unlike the second member who finishes more often in state (1). For the first member of the dyad, we see that the weakest transition is between the combination $(1, 2)$ and state (2). However, we find the maximum transition at the level of the combination $(1, 1)$ and state (2). For the second member of the dyad, the weakest transition is between the combination $(2, 2)$ and the state (1) and the most important number of transitions is between the combination $(1, 2)$ and state (1). Both matrices `estimationFM.APM` and `estimationSM.APM` are the estimated transition probabilities matrices for both members of the dyad. The matrices represent the probabilities of transition between the S^2 combination and the S states. The sum of each row of these matrices is 1, which is correct. Details on both `countEmp` and `mleEstimation` are available in the comment of Listing 2 in the Appendix.

As shown, when the unit of analysis is the individual,



Listing 2 ■ Count matrix and probabilities transition matrix

```

empMatFM.APM <- countEmp(states = S,
  chainFM = apmFM, chainSM = apmSM)
empMatSM.APM <- countEmp(states = S,
  chainFM = apmSM, chainSM = apmFM)

empMatFM.APM
## [,1] [,2]
## [1,] 8 17
## [2,] 15 3
## [3,] 14 14
## [4,] 6 12

empMatSM.APM
## [,1] [,2]
## [1,] 15 10
## [2,] 21 7
## [3,] 12 6
## [4,] 5 13

sum(empMatFM.APM)
## [1] 89

sum(empMatSM.APM)
## [1] 89

estimationFM.APM <- mleEstimation(countMat =
  empMatFM.APM)
estimationSM.APM <- mleEstimation(countMat =
  empMatSM.APM)

estimationFM.APM
## [,1] [,2]
## [1,] 0.3200000 0.6800000
## [2,] 0.8333333 0.1666667
## [3,] 0.5000000 0.5000000
## [4,] 0.3333333 0.6666667

estimationSM.APM
## [,1] [,2]
## [1,] 0.6000000 0.4000000
## [2,] 0.7500000 0.2500000
## [3,] 0.6666667 0.3333333
## [4,] 0.2777778 0.7222222

rowSums(estimationFM.APM)
## [1] 1 1 1 1

rowSums(estimationSM.APM)
## [1] 1 1 1 1

```

there are two matrices. At the couple level, the possible states are the same at time t and $t + 1$. Indeed, because we are interested in the behavior of the dyad, the state space

is the combination of all possible states. This leads to a squared matrix of size $S^2 \times S^2$. Using the previous example, the matrix at the couple level appears as follows:

$$TP = \begin{matrix} & \begin{matrix} (1,1) & (1,2) & (2,1) & (2,2) \end{matrix} \\ \begin{matrix} (1,1) \\ (1,2) \\ (2,1) \\ (2,2) \end{matrix} & \begin{pmatrix} p_{1|11}^{FM} p_{1|11}^{SM} & p_{1|11}^{FM} p_{2|11}^{SM} & p_{2|11}^{FM} p_{1|11}^{SM} & p_{2|11}^{FM} p_{2|11}^{SM} \\ p_{1|12}^{FM} p_{1|21}^{SM} & p_{1|12}^{FM} p_{2|21}^{SM} & p_{2|12}^{FM} p_{1|21}^{SM} & p_{2|12}^{FM} p_{2|21}^{SM} \\ p_{1|12}^{FM} p_{1|21}^{SM} & p_{1|12}^{FM} p_{2|21}^{SM} & p_{2|12}^{FM} p_{1|21}^{SM} & p_{2|12}^{FM} p_{2|21}^{SM} \\ p_{1|22}^{FM} p_{1|22}^{SM} & p_{1|22}^{FM} p_{2|22}^{SM} & p_{2|22}^{FM} p_{1|22}^{SM} & p_{2|22}^{FM} p_{2|22}^{SM} \end{pmatrix} \end{matrix}$$

The matrix summarizes both sequences at once. The probabilities constituting it are a composite score of the individual transition probabilities. Therefore, although this matrix represents the overall dynamics of the couple, it remains less interesting than the individual matrices, which makes it possible to understand the behavior of each individual towards the behavior of their partner. Dyadic data analysis at the dyadic level loses individual information.

Dyadic pattern identification

The transition matrix is constructed with conditional probabilities related to the state the two members of the dyad are in at the previous time. In the case of the AM, the state of the partner does not have to be considered. There-

fore, we can perform a restriction on the transition matrix at the level of the partner state. Similarly, in the case of the PM, the state of the actor does not matter and therefore a restriction can also be made. In the case of the IM, time t and $t + 1$ are independent. Therefore, the restrictions can be done both on the actor and on the partner effects. In the matrices, each restriction is done with \cdot at the place of the state. Using the same notation and example used previously, the case where the partner has no effect implies that rows (1, 1) and (1, 2) and rows (2, 1) and (2, 2) are the same. In a similar way, the case where the actor has no effect implies that rows (1, 1) and (2, 1) and rows (1, 2) and (2, 2) are the same. For the independence model, we do both restrictions. For the example with $S = 2$, the restrictions can be

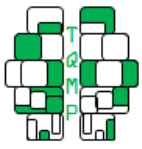


Table 1 ■ Likelihood ratio test: Identification of the interaction

APM vs. AM	APM vs. PM	
	$> \alpha$	$\leq \alpha$
$> \alpha$	Independence model	Actor-only model
$\leq \alpha$	Partner-only model	Actor-partner model

done as follows:

$$TP^{AM} = \begin{matrix} & & 1 & 2 \\ \begin{matrix} (1,1) \\ (1,2) \\ (2,1) \\ (2,2) \end{matrix} & \left(\begin{matrix} p_{1|1\cdot} & p_{2|1\cdot} \\ p_{1|1\cdot} & p_{2|1\cdot} \\ p_{1|2\cdot} & p_{2|2\cdot} \\ p_{1|2\cdot} & p_{2|2\cdot} \end{matrix} \right) \end{matrix}$$

$$TP^{PM} = \begin{matrix} & & 1 & 2 \\ \begin{matrix} (1,1) \\ (1,2) \\ (2,1) \\ (2,2) \end{matrix} & \left(\begin{matrix} p_{1|\cdot 1} & p_{2|\cdot 1} \\ p_{1|\cdot 2} & p_{2|\cdot 2} \\ p_{1|\cdot 1} & p_{2|\cdot 1} \\ p_{1|\cdot 2} & p_{2|\cdot 2} \end{matrix} \right) \end{matrix}$$

$$TP^{IM} = \begin{matrix} & & 1 & 2 \\ \begin{matrix} (1,1) \\ (1,2) \\ (2,1) \\ (2,2) \end{matrix} & \left(\begin{matrix} p_{1|\cdot\cdot} & p_{2|\cdot\cdot} \\ p_{1|\cdot\cdot} & p_{2|\cdot\cdot} \\ p_{1|\cdot\cdot} & p_{2|\cdot\cdot} \\ p_{1|\cdot\cdot} & p_{2|\cdot\cdot} \end{matrix} \right) \end{matrix}$$

The type of interaction is therefore completely defined by the structure of the matrix. To identify whether restrictions are possible, we can use the likelihood ratio test (LRT). This test is used between two nested models. In the LRT, the hypotheses are as follow:

- Null hypothesis (H_0): The reduced model is the model for the data.
- Alternative hypothesis (H_1): The full model is the model for the data.

The null hypothesis assumes that a restriction on parameters is possible. If H_0 is rejected, the APM is the right model to build the structure of the matrix. Two tests can be performed. The first one tests the full APM against the AM and the second one tests the full APM against the PM. In each case, if H_0 is not rejected, the reduced model is kept. For each dyadic sequence, both tests are performed at the same time. If restrictions are possible for both models, the IM is kept. Table 1 represents the four possible cases when testing both restrictions. The α can be any threshold chosen between 0 and 1, although it is generally fixed at 0.05.

The test is done at the individual level because each individual can adopt one type of interaction. This test is performed using the function `intType`. As for the count matrix, each member of the dyad is successively the first and the second member. The function performs both tests at the same time. Consequently, the interpretation of the test

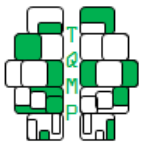
directly shows the type of interaction that exists. Listing 3 performs the test for the APM. The `apm.resFM` is done from the point of view of the first member of the dyad and the `apm.resSM` is done from the point of view of the second member. Referring to our example on sexual desire, the first test identifies the interaction pattern of the woman, and the second test identifies that of the man. In each case, the test identifies the type of interaction as being an actor-partner case. Knowing that our data was simulated using the same pattern, the test correctly identifies the type of interaction that exists between both individuals.

Our example focuses on the question "Did you have sexual desire yesterday?". The possible answers are *yes* and *no*, which implies two states, $S = 2$. If the possible answers had been *yes*, *no* and *unsure*, S would have been 3. A complete example of this case for the three kinds of pattern of interaction is provided in Listing 7 with a comment.

Conclusion and further directions

To analyze categorical dyadic sequences correctly within the conceptual framework of APIM, it is necessary to use specialized statistical models that account for both the categorical nature of the data and the longitudinal design. The models should accurately identify the actor and partner effects inherent in APIM. Markov chains provide a suitable approach to fulfill these requirements. Our article provides a comprehensive explanation of how Markov chains can be adapted to the dyadic context. In addition, we include a step-by-step tutorial in R language to explain the practical application of this approach. We show how this method can be used to identify interaction patterns within dyadic data. This work shows that Markov chains provide a valuable framework that can handle not only cases similar to logistic regression, which is commonly used for situations with two states, but also cases with a greater number of states. Moreover, we have demonstrated that by employing Markov chains, we can effectively identify the mechanism of interactions without the need for additional analytical methods.

Logistic regression and Markov chains can handle a case with two states. Conducting a comparative analysis between these two approaches would be valuable to discern the crucial distinctions and identify the strengths and limitations of each method. This paper adopts a unique case perspective to remain at the most basic level of anal-

**Listing 3 ■ Identification of the interaction**

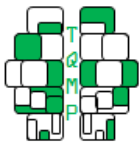
```
apm.resFM <- intType(states = S, FM = apmFM, SM = apmSM, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 12.323, p-value = 0.002109
## alternative hypothesis: The full model fits the data better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 11.02, p-value = 0.004047
## alternative hypothesis: The full model fits the data better
##
## [1] "The type of the model is APM"

apm.resSM <- intType(states = S, FM = apmSM, SM = apmFM, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 6.8253, p-value = 0.03295
## alternative hypothesis: The full model fits the data better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 10.141, p-value = 0.006279
## alternative hypothesis: The full model fits the data better
##
## [1] "The type of the model is APM"
```

ysis. However, multiple dyad samples are also a reality in psychological research, and it is therefore necessary to work not only with a unique case dyad approach, but also with a group approach. Consequently, our approach needs to be further developed to examine similarities and differences between a larger number of individuals within a group of dyads.

References

- De Haan-Rietdijk, S., Kuppens, P., & Hamaker, E. (2016). What's a day? A guide to decomposing the variance in intensive longitudinal data. *Frontiers in Psychology*, 7(891), 1–16. doi: [10.3389/fpsy.2016.00891](https://doi.org/10.3389/fpsy.2016.00891).
- Estrada, E., Sbarra, D., & Ferrer, E. (2020). Models for dyadic data. *Handbook of Research Methods in Clinical Psychology*, 350–368.
- Fitzpatrick, J., Gareau, A., Lafontaine, M.-F., & Gaudreau, P. (2016). How to use the actor-partner interdependence model (APIM) to estimate different dyadic patterns in MPLUS: A step-by-step tutorial. *The Quantitative Methods for Psychology*, 12(1), 74–86. doi: [10.20982/tqmp.12.1.p074](https://doi.org/10.20982/tqmp.12.1.p074).
- Fuchs, P., Nussbeck, F., Meuwly, N., & Bodenmann, G. (2017). Analyzing dyadic sequence data - research questions and applied statistical models. *Frontiers in Psychology*, 8(429), 1–14. doi: [10.3389/fpsyg.2017.00429](https://doi.org/10.3389/fpsyg.2017.00429).
- Gonzalez, R., & Griffin, D. (2012). Dyadic data analysis. In H. Cooper, P. M. Camic, D. L. Long, A. T. Panter, D. Rindskopf, & K. J. Sher (Eds.), *APA handbook of research methods in psychology* (pp. 439–450, Vol. 3). American Psychological Association. doi: [10.1037/13620-000](https://doi.org/10.1037/13620-000).
- Ibe, O. (2013a). Hidden markov models. In O. Ibe (Ed.), *Markov processes for stochastic modeling* (pp. 341–376). Elsevier.
- Ibe, O. (2013b). Introduction to markov processes. In O. Ibe (Ed.), *Markov processes for stochastic modeling* (pp. 45–53). Elsevier.
- Ibe, O. (2013c). Markov chain Monte Carlo. In O. Ibe (Ed.), *Markov processes for stochastic modeling* (pp. 423–450). Elsevier.
- Kazdin, A. E. (2012). Single-case research designs, second edition. *Child & Family Behavior Therapy*, 34, 76–79. doi: [10.1080/07317107.2012.654458](https://doi.org/10.1080/07317107.2012.654458).
- Kendall, P. C., & Comer, J. S. (2014). Research methods in clinical psychology. In D. Barlow (Ed.), *The Oxford handbook of clinical psychology: Upyear edition* (pp. 23–33). Oxford University Press. doi: [10.1093/oxfordhb/9780199328710.001.0001](https://doi.org/10.1093/oxfordhb/9780199328710.001.0001).
- Kenny, D., & Kashy, D. (2013). The design and analysis of data from dyads and groups. In H. Reis & C. Judd (Eds.), *Handbook of research methods in social and personality psychology* (pp. 587–605). Cambridge University Press.
- Kenny, D., Kashy, D., & Cook, W. (2006). *Dyadic data analysis*. Guilford.
- Laurenceau, J.-P., & Bolger, N. (2011). Analyzing diary and intensive longitudinal data from dyads. In M. R. Mehl & T. Conner (Eds.), *Handbook of research methods for studying daily life* (pp. 63–92). Guilford.



Mair, P. (2018). Modeling trajectories and time series. In P. Mair (Ed.), *Modern psychometrics with R* (pp. 365–407). Springer International Publishing. doi: [10.1007/978-3-319-93177-7](https://doi.org/10.1007/978-3-319-93177-7).

van Rijn, P. (2008). *Categorical time series in psychological measurement* [Doctoral dissertation, Faculty of social and behavioral sciences, University of Amsterdam].

Walls, T., Jung, H., & Schwartz, J. (2006). Multilevel models for intensive longitudinal data. In T. Walls & J. Schafer (Eds.), *Models for intensive longitudinal data* (pp. 3–37). Oxford University Press. doi: [10.1093/acprof:oso/9780195173444.001.0001](https://doi.org/10.1093/acprof:oso/9780195173444.001.0001).

Appendix

Dyadic answers for a couple. Example of potential answers for both partners of a couple to the question *Did you have sexual desire yesterday?* The answers are reported for 10 days. Tables 2 and 3 report the answers for the woman and the man, respectively.

Table 2 ■ Answers for the woman

t	0	1	2	3	4	5	6	7	8	9
s	yes	yes	yes	no	no	yes	no	yes	yes	no

Table 3 ■ Answers for the man

t	0	1	2	3	4	5	6	7	8	9
s	yes	yes	no	no	yes	yes	yes	no	no	no

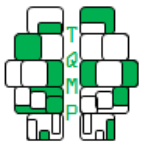
Step-by-step example for classic Markov chains. Classic Markov chains model one sequence only. Let us take the sequence of the woman in the previous example and recode *yes* as 1 and *no* as 2. Table 4 shows the answers to this question over 10 days. It can be seen as a classic Markov chain with 10 time measurements. We can write this in the following way: $T = 10$, $S = 2$. The initial state is the state at time $t = 0$. The chain begins in state $s = 1$. If we want to analyze the chain, we can see that the woman is more likely to be in state 1 than in state 2. We can also see that the woman can go from state 1 to state 2 and vice versa, therefore, there is no absorbing state from which the woman does not emerge. We can make these few observations because the chain is short, but to have a deeper analysis, we must obtain the transition probabilities between each state. The count matrix C represents the transition as the number of times the chain changes from one state to another. The transition probabilities matrix TP represents the estimates’ transition probabilities between each state at a time. They are condition probabilities. Looking at the transition probabilities matrix, we observe that when the woman is in state 1, she has the same chance to end up in the same state or in state 2 at the next measurement point. We also see that when she is in state 2, she has more chance to end up in state 1 than in state 2. Therefore, we see that she has a greater tendency to be in state 1 than in state 2, which translates into the fact that the woman answers *yes* more often than *no* to the question she is asked.

Table 4 ■ Classic Markov chains with two states: $yes = 1$, $no = 2$

t	0	1	2	3	4	5	6	7	8	9
s	1	1	1	2	2	1	2	1	1	2

$$C = \begin{matrix} & & 1 & 2 \\ 1 & \left(\begin{matrix} 3 & 3 \\ 2 & 1 \end{matrix} \right) \\ 2 & & & \end{matrix}$$

$$TP = \begin{matrix} & & 1 & 2 \\ 1 & \left(\begin{matrix} 0.5 & 0.5 \\ 0.667 & 0.333 \end{matrix} \right) \\ 2 & & & \end{matrix}$$

**Listing 4** ■ Actor-only sequences

```
load("AM.RData")

amFM <- AM$chain1
amSM <- AM$chain2

empMatFM.AM <- countEmp(states = S, chainFM = amFM, chainSM = amSM)
empMatSM.AM <- countEmp(states = S, chainFM = amSM, chainSM = amFM)

estimationFM.AM <- mleEstimation(countMat = empMatFM.AM)
estimationSM.AM <- mleEstimation(countMat = empMatSM.AM)

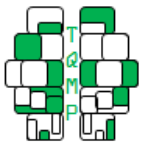
am.resFM <- intType(states = S, FM = amFM, SM = amSM, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 0.48677, p-value = 0.784
## alternative hypothesis: The full model fits the data better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 12.183, p-value = 0.002262
## alternative hypothesis: The full model fits the data better
##
## [1] "The type of the model is AM"

am.resSM <- intType(states = S, FM = amSM, SM = amFM, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 2.9648, p-value = 0.2271
## alternative hypothesis: The full model fits the data better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 11.625, p-value = 0.00299
## alternative hypothesis: The full model fits the data better
##
## [1] "The type of the model is AM"
```

Comment on the simulation method from Listing 1. We simulated dyadic sequences using Markov chains. The first simulation concerns the transition probabilities matrix. To simulate an individual transition matrix, it is necessary to simulate S vectors according to a standardized uniform distribution using random numbers. Because the transition matrix is a stochastic matrix, each vector must sum to 1. To simulate a dyadic transition probabilities matrix, the same procedure is used considering that the state space is the combination of specific states for each individual. Sequences generation depends on three elements: the initial state, the transition probabilities matrix and the desired length of the chain. Once the transition matrix is simulated according the type of interaction involved, the sequence is generated using the multinomial distribution.

Comment on functions used in Listing 2. Both `countEmp` and `mleEstimation` functions have been coded to adapt classic Markov chains to the dyadic case. The `countEmp` function computes the count matrix for both individual and dyadic cases. The input elements of this function are the number of states involved in the sequence, the sequence and optionally the sequence of the partner of the individual considered in the case of dyadic data. The function is structured such that the first chain concerns the first member. Consequently, in the case of dyadic data, each member of the dyad is successively the first member. The `mleEstimation` function takes a count matrix for the input element and returns the estimated probabilities transition. It computes these probabilities according to maximum likelihood. Because each row of the count matrix is independent, the estimation is done on each row separately.

Comment on Listing 4. The first step is to extract the chains for both members. They are the `amFM` and `amSM` elements. Once the chains are extracted, the likelihood ratio test can be run to identify the kind of interaction the members of the dyad use. For both members, we observe that the actor-only model is the model of the interaction. Indeed, for Actor-only model, H_0 is not rejected, meaning that the reduced model must be kept. The behavior of the partner does

**Listing 5** ■ Partner-only sequences

```
load("PM.RData")

pmFM <- PM$chain1
pmSM <- PM$chain2

pm.resFM <- intType(states = S, FM = pmFM, SM = pmSM, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 7.9844, p-value = 0.01846
## alternative hypothesis: The full model fits the data better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 1.1539, p-value = 0.5616
## alternative hypothesis: The full model fits the data better
##
## [1] "The type of the model is PM"

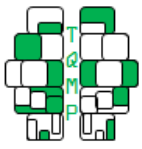
pm.resFM <- intType(states = S, FM = pmSM, SM = pmFM, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 9.1778, p-value = 0.01016
## alternative hypothesis: The full model fits the data better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 2.3873, p-value = 0.3031
## alternative hypothesis: The full model fits the data better
##
## [1] "The type of the model is PM"
```

not influence the behavior of the actor. For Partner-only model, H_0 is rejected, meaning that the full model must be kept. Consequently, the behavior of the partner does not influence the behavior of the actor, but their own behavior matters for the modeling of the sequence. Therefore, the pattern of interaction is actor-only. The count matrix and the estimated transition probabilities matrix could be extracted in a way that is similar to that of actor-partner sequences.

Comment on Listing 5. The first step is to extract the chains for both members. They are the pmFM and pmSM elements. Once the chains are extracted, the likelihood ratio test can be run to identify the kind of interaction the members of the dyad use. For both members, we observe that the partner-only model is the model of the interaction. Indeed, for Actor-only model, H_0 is rejected, meaning that the full model must be kept. For Partner-only model, H_0 is not rejected, meaning that the reduced model must be kept. Consequently, the behavior of the partner fully defined the behavior of the actor. Therefore, the pattern of interaction is partner-only. The count matrix and the estimated transition probabilities matrix could be extracted in a similar way than for actor-partner sequences.

Comment on Listing 6. The first step is to extract the chains for both members. They are the imFM and imSM elements. Once the chains are extracted, the likelihood ratio test can be run to identify the kind of interaction the members of the dyad use. For both members, we observe that the independence model is the model of the interaction. Indeed, for Actor-only model, H_0 is not rejected, meaning that the reduced model must be kept. For Partner-only model, H_0 is also not rejected, meaning that the reduced model must be kept. Consequently, both actor and partner restrictions can be made, leading to an independent pattern of interaction. The count matrix and the estimated transition probabilities matrix could be extracted in a way that is similar to that of actor-partner sequences.

Comment on Listing 7. For case $S = 3$, the procedure is similar to the one for case $S = 2$. First, we have to import the data, set the parameter and export the chains. Then, the sequences can be estimated using Markov chains as in the case $S = 2$. We observe that the size of the matrix is not the same, because it contains the following combinations: (1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3). Concerning the tests for the identification of interaction patterns, the procedure is similar. We see that it works and identifies the patterns correctly.



Listing 6 ■ Independent sequences

```
load("IM.RData")

imFM <- PM$chain1
imSM <- PM$chain2

im.resFM <- intType(states = S, FM = imFM, SM = imSM, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 7.9844, p-value = 0.01846
## alternative hypothesis: The full model fits the data better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 1.1539, p-value = 0.5616
## alternative hypothesis: The full model fits the data better
##
## [1] "The type of the model is PM"

im.resFM <- intType(states = S, FM = imSM, SM = imFM, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 9.1778, p-value = 0.01016
## alternative hypothesis: The full model fits the data better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 2.3873, p-value = 0.3031
## alternative hypothesis: The full model fits the data better
##
## [1] "The type of the model is PM"
```

Open practices

- The *Open Data* badge was earned because the data of the experiment(s) are available on [the journal's web site](#).
- The *Open Material* badge was earned because supplementary material(s) are available on [the journal's web site](#).

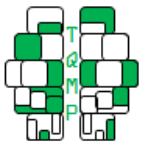
Citation

Bollenr ucher, M., Darwiche, J., & Antonietti, J.-P. (2023). Dyadic pattern analysis using longitudinal Actor-Partner Interdependence Model with Markov chains for unique case analysis. *The Quantitative Methods for Psychology*, 19(3), 230–243. doi: [10.20982/tqmp.19.3.p230](https://doi.org/10.20982/tqmp.19.3.p230).

Copyright © 2023, *Bollenr ucher et al.* This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) or licensor are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Received: 30/11/2022 ~ Accepted: 22/07/2023

Listing 7 follows.



Listing 7 ■ Full code analysis for S = 3

```

load("APM3.RData")
load("AM3.RData")
load("PM3.RData")
load("IM3.RData")

S <- 3

apmFM3 <- APM3$chain1
apmSM3 <- APM3$chain2
amFM3 <- AM3$chain1
amSM3 <- AM3$chain2
pmFM3 <- PM3$chain1
pmSM3 <- PM3$chain2
imFM3 <- IM3$chain1
imSM3 <- IM3$chain2

empMatFM3.APM <- countEmp(states = S, chainFM = apmFM3,
  chainSM = apmSM3)
empMatSM3.APM <- countEmp(states = S, chainFM = apmSM3,
  chainSM = apmFM3)

empMatFM3.APM
## [,1] [,2] [,3]
## [1,] 35 0 1
## [2,] 0 0 2
## [3,] 1 3 0
## [4,] 3 1 0
## [5,] 0 0 3
## [6,] 0 0 0
## [7,] 1 3 2
## [8,] 0 0 2
## [9,] 2 0 30

empMatSM3.APM
## [,1] [,2] [,3]
## [1,] 32 1 3
## [2,] 4 0 0
## [3,] 4 2 0
## [4,] 1 0 1
## [5,] 3 0 0
## [6,] 0 1 1
## [7,] 1 3 0
## [8,] 0 0 0
## [9,] 0 0 32

estimationFM3.APM <- mleEstimation(countMat =
  empMatFM3.APM)
estimationSM3.APM <- mleEstimation(countMat =
  empMatSM3.APM)

estimationFM3.APM
## [,1] [,2] [,3]
## [1,] 0.9722222 0.00 0.02777778
## [2,] 0.0000000 0.00 1.00000000
## [3,] 0.2500000 0.75 0.00000000
## [4,] 0.7500000 0.25 0.00000000
## [5,] 0.0000000 0.00 1.00000000
## [6,] 0.0000000 0.00 0.00000000
## [7,] 0.1666667 0.50 0.33333333
## [8,] 0.0000000 0.00 1.00000000
## [9,] 0.0625000 0.00 0.93750000

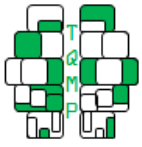
estimationSM3.APM
## [,1] [,2] [,3]
## [1,] 0.8888889 0.02777778 0.08333333
## [2,] 1.0000000 0.00000000 0.00000000
## [3,] 0.6666667 0.33333333 0.00000000
## [4,] 0.5000000 0.00000000 0.50000000
## [5,] 1.0000000 0.00000000 0.00000000
## [6,] 0.0000000 0.50000000 0.50000000
## [7,] 0.2500000 0.75000000 0.00000000
## [8,] 0.0000000 0.00000000 0.00000000
## [9,] 0.0000000 0.00000000 1.00000000

apm3.resFM <- intType(states = S, FM = apmFM3,
  SM = apmSM3, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 84.994, p-value = 4.564e-13
## alternative hypothesis: The full model fits the data
  better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 57.009, p-value = 7.875e-08
## alternative hypothesis: The full model fits the data
  better
##
## [1] "The type of the model is APM"

apm3.resSM <- intType(states = S, FM = apmSM3,
  SM = apmFM3, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 50.984, p-value = 9.375e-07
## alternative hypothesis: The full model fits the data
  better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 67.472, p-value = 9.49e-10
## alternative hypothesis: The full model fits the data
  better
##
## [1] "The type of the model is APM"

am3.resFM <- intType(states = S, FM = amFM3,
  SM = amSM3, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 8.5686, p-value = 0.7393
## alternative hypothesis: The full model fits the data
  better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 37.139, p-value = 0.0002121
## alternative hypothesis: The full model fits the data
  better
##
## [1] "The type of the model is AM"

```

**Listing 8 ■ Listing 7 (followed)**

```
am3.resSM <- intType(states = S, FM = amSM3,
  SM = amFM3, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 11.859, p-value = 0.4571
## alternative hypothesis: The full model fits the data
  better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 26.249, p-value = 0.009897
## alternative hypothesis: The full model fits the data
  better
##
## [1] "The type of the model is AM"

pm3.resFM <- intType(states = S, FM = pmFM3,
  SM = pmSM3, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 52.241, p-value = 5.618e-07
## alternative hypothesis: The full model fits the data
  better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 16.198, p-value = 0.1823
## alternative hypothesis: The full model fits the data
  better
##
## [1] "The type of the model is PM"

pm3.resSM <- intType(states = S, FM = pmSM3,
  SM = pmFM3, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 36.736, p-value = 0.0002465
## alternative hypothesis: The full model fits the data
  better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 5.3098, p-value = 0.9468
## alternative hypothesis: The full model fits the data
  better
##
## [1] "The type of the model is PM"

im3.resFM <- intType(states = S, FM = imFM3,
  SM = imSM3, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 7.8812, p-value = 0.7943
## alternative hypothesis: The full model fits the data
  better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 9.2959, p-value = 0.6775
## alternative hypothesis: The full model fits the data
  better
##
## [1] "The type of the model is IM"

im3.resSM <- intType(states = S, FM = imSM3,
  SM = imFM3, alpha = 0.05)
## Likelihood ratio test, Actor-only model
## data: Dyadic sequences
## X-squared = 15.391, p-value = 0.2207
## alternative hypothesis: The full model fits the data
  better
##
## Likelihood ratio test, Partner-only model
## data: Dyadic sequences
## X-squared = 14.379, p-value = 0.2772
## alternative hypothesis: The full model fits the data
  better
##
## [1] "The type of the model is IM"
```