rties: Overview and Data Preparation

Emily Butler, eabutler@u.arizona.edu

The *rties* package streamlines and simplifies testing whether a set of bivariate temporal patterns either predict, or are predicted by, global variables of interest. Although we focus on time-series data from both people in dyads, the models are appropriate for any bivariate time-series data. The analysis starts with an observed variable that is assumed to be a valid indicator of some underlying phenomenon (e.g., emotion), assessed repeatedly over time for both partners in a sample of dyads. We provide a set of models that represent different aspects of the bivariate dynamics of the observed variable. The parameter estimates from those models are then used either to predict, or be predicted by, cross-sectional variables of interest (e.g., each person's global health). Currently the package includes 3 models: 1) the "Inertia-Coordination" model, which can represent within-person inertia and between-person coordination (Figure 1), 2) the "Patterned-Slopes" model, which can represent between-person escalation, de-escalation, convergence and divergence (Figure 2), and 3) a "Coupled-Oscillator" model, which can represent between-person amplification and damping of coupled oscillations (Figure 3).

The first step in our method is to choose the observed variable, which must take the form of a bivariate time-series. In the language of dynamic systems, the pair of observed responses are indicative of the underlying (latent) states of the system across time. In *rties* we refer to the two time-series of observed responses as the "state" variables, and seek to represent their dynamics with a mathematical model. The temporal unit of observation (e.g., seconds, minutes, days) and minimum number of observations per dyad required for these analyses depend on theory about the process of interest, in combination with the model chosen. The minimum number of dyads follows guidelines for the number of observational units needed for multiple regression, given a set of *n* predictors. More information is given in the documentation for each model.

The second step is to choose which of the model(s) include parameters representing relevant aspects of the system that you think characterizes the observed state variables. For example, imagine a very simple model that only has intercepts and linear slopes for each partner (e.g., we theorize that the system can be characterized by two independent lines, which is clearly over-simplified and not "interpersonal" at all, but useful for understanding our general approach). We then estimate the parameters for that model separately for each dyad. In our example, we would use our data to estimate the 4 parameters of the model (2 intercepts and 2 slopes) for each dyad.

In the final step we use the set of parameter estimates (e.g., each dyad's intercept and slope estimates) either as predictors, or outcomes, of some variable of interest across all dyads. We refer to these global variables as "system" variables because they are theoretically related to the system being studied, but change slower than the state variables used to assess the dynamics of the system (for examples of other research groups using similar approaches see: Felmlee, 2007; Ferrer, Chen, Chow, & Hsieh, 2010; Ferrer & Steele, 2014; Hollenstein, 2013; Steele, Ferrer, & Nesselroade, 2014; Steenbeek & van Geert, 2005).

Getting Started

You can install *rties* from GitHub and load it in R with the following syntax:

```
devtools::install_github("ebmtnprof/rties")
library(rties)
```

You can then load the example data, assign it to a dataframe and list the contents:

```
data(rties_ExampleData_1)
data1 <- rties_ExampleData_1
str(data1)
```

This produces the following output:

```
'data.frame':  78858 obs. of  6 variables:
 $ couple: int  2 2 2 2 2 2 2 2 2 2 ...
 $ person: int  2 2 2 2 2 2 2 2 2 2 ...
 $ time  : int  1 2 3 4 5 6 7 8 9 10 ...
 $ dial  : num  2.51 2.51 2.51 2.51 2.51 ...
 $ sub   : num  0.0755 0.0755 0.0755 0.0755 0.0755 ...
 $ sexm  : int  0 0 0 0 0 0 0 0 0 0 ...
```

All of the examples in the *rties* documentation use this data. The variables "couple" and "person" are identification variables for the dyads and individuals within dyads respectively. The variable "time" indicates the temporal sequence of the observations. The variable called "dial" is the state variable, which in this example is the emotional experience of romantic partners, self-reported in 2 second units using a rating dial following a conversation while watching the video of their conversation as a memory prompt. The variable called "sub" is the system variable to be predicted from (or used to predict) the dynamics of the partner's emotional experience. In this example it is an observational measure representing the proportion of time during the conversation that a couple showed evidence of shared unhealthy behaviors. Finally, "sexm" is a numeric variable, scored as zero or one, which distinguishes between the types of partners, which in our example are females (scored 0) and males (scored 1).

Data Visualization and Preparation

The *rties* package includes functions to do all the data processing necessary for any of the models, but you must provide it with a dataframe that meets a number of criteria. At a minimum, the dataframe must include: 1) a person-level ID, 2) a dyad-level ID, 3) a time-varying observable (the state variable), 4) a time-constant system variable, e.g., something that will be either predicted from, or act as a predictor of, the dynamics of the observed state variables, 5) a variable that distinguishes the partners (e.g., sex, mother/daughter, etc) that is numeric and scored 0/1, and 6) a variable that indicates sequential temporal observations. The dataframe can include additional variables and they will be ignored by most functions in the package. In a few cases, however, such as some of the visualization functions, all variables in the dataframe will be processed, making it advisable to work with a smaller set of relevant variables, rather than every variable measured in a study. The system variable must be cross-sectional, but can be measured at either the individual or the dyad level. In our example, it is at the dyad-level, since it was the couple as a whole who received a score for shared unhealthy behaviors. In this situation, all results apply to both partners equally. In contrast, the system variable could be something like individual's depression scores, in which case model results are provided separately for each level of the distinguishing variable (women and men in our examples).

The partners within each dyad must have the same number of observations (e.g. rows of data), although those can include rows that have missing values (NAs). Each dyad, however, can have it's own unique number of observations. The dyad ID and person ID must be the same for the 0-level of the distinguishing variable. Then you must choose a large number to add to the 0-level person ID to get the 1-level person ID. For example, if you have 200 couples, you might choose 500 as the value to add. Then if the dyad ID and 0-level person IDs for 4 participants were 1, 3, 4, 7 the 1-level partners would be 501, 503, 504, 507. Some functions will need that number as an argument called "idConvention".

Bivariate dynamic models are complex and there are a lot of ways to fool yourself about the

quality and interpretation of the results. To help prevent this, we strongly advocate looking at your data and model results at every step of the analysis and we provide visualization tools to make it easy. As an initial check, we recommend looking at histograms of all of the numeric variables in your initial dataframe. This should uncover any seriously ill-behaved data (extreme outliers, zero variance, etc). In addition, if you intend to use the system variable as an outcome of the dynamics, it needs to be reasonably normally distributed (although we intend to implement non-Gaussian options in the future). The following function produces histograms for all numeric variables in a dataframe:

```
histAll(data1)
```

A second highly recommended check is to plot the raw time-series data for each dyad, which can be accomplished with the "plotRaw" function. It takes as arguments the names of the dataframe, the dyad ID, the state variable, the distinguishing variable, and the variable that indicates sequential temporal observations. It also takes strings for the labels of the levels of the distinguishing variable in the correct order (e.g., a name for 0-level first, in this case "Women", then a name for 1-level, in this case "Men"). Running the following code we see that one person is completely missing data for dyad 19, which will cause problems if we don't fix it. This visualization is also useful for spotting outliers, noise in the data, etc.

```
plotRaw(data1, "couple", "dial", "sexm", "time", "Women", "Men")
```

At this point, it would be worthwhile to see if the missing data can be accounted for. In our case, we discovered there had been recording problems for the one person and so the only thing we can do about the missing data for dyad 19 is to remove that dyad from the analysis (several steps in the modeling require at least partial data from both partners and will generate error messages if that is violated). The following syntax accomplishes that. First, the dyad to remove is assigned to a variable called "dyads" (if more than one dyad is to be removed, this could be a vector, e.g., dyads <- c(3, 5, 19) would remove three dyads). Then the "removeDyads" function takes as arguments the full dataframe, the variable indicating which dyads to remove, and the name of the dataframe column that includes dyad membership. Having removed the dyad we then re-plot the raw data to ensure everything worked correctly.

```
dyads <- c(19)
data2 <- removeDyads(data1, dyads, data1$couple)
plotRaw(data2, "couple", "dial", "sexm", "time", "Women", "Men")
```

The "dataPrep" function does all the work of processing the data into a generic form that is appropriate for any of the *rties* models. Function arguments are: basedata (the cleaned up dataframe; "data2" in this example), id (the person ID), dyad (the dyad ID), obs (the observed state variable), sysVar (the cross-sectional system variable), dist (the distinguishing variable), time_name (the variable indicating temporal sequence), and time_lag (an optional argument for the number of lags to use for the lagged observable).The function returns a dataframe that has all the variables needed for TIES modeling, each renamed to a generic variable name, which are:

- id = person id
- dyad = dyad id
- obs = observable state variable
- sysVar = system variable
- dist1 = 0/1 variable where the 1's indicate the 1's in the original distinguishing variable

• dist0 = 0/1 variable where the 1's indicate the 0's in the original distinguishing variable
• time = the variable indicating temporal sequence
• obs_deTrend = the observed state variable with each person's linear trend removed
• p_ = all the same variables, but for a person's partner rather than themselves

A note about choosing the lag: You can lag the observed variable by any number of steps and that decision should be driven by theory and prior research. In this case, emotional experience was assessed in 2 second intervals and so a 1-step lag is very small and doesn't leave time for meaningful change between observations. We chose a lag of 5 steps, which is equivalent to 10 seconds, because that is a very common unit to aggregate over in the emotional experience literature and if emotions unfold over several minutes (as suggested by many theories), then 10-second steps are large enough to allow meaningful change between each step, but small enough to not miss important changes. The usage of "dataPrep" follows, where "data2" is the original data set with any necessary cleaning done and "data3" will be a dataframe holding the processed data.

```
data3 <- dataPrep(data2, "person", "couple", "dial", "sub", "sexm",
 "time", time_lag=5)
```

At this point you are ready to begin using one or more of the models to test whether any of the dynamic patterns can predict, or be predicted by, your system variable(s) of interest. The procedures for each model are documented in the various examples provided in the "documentation" folder.

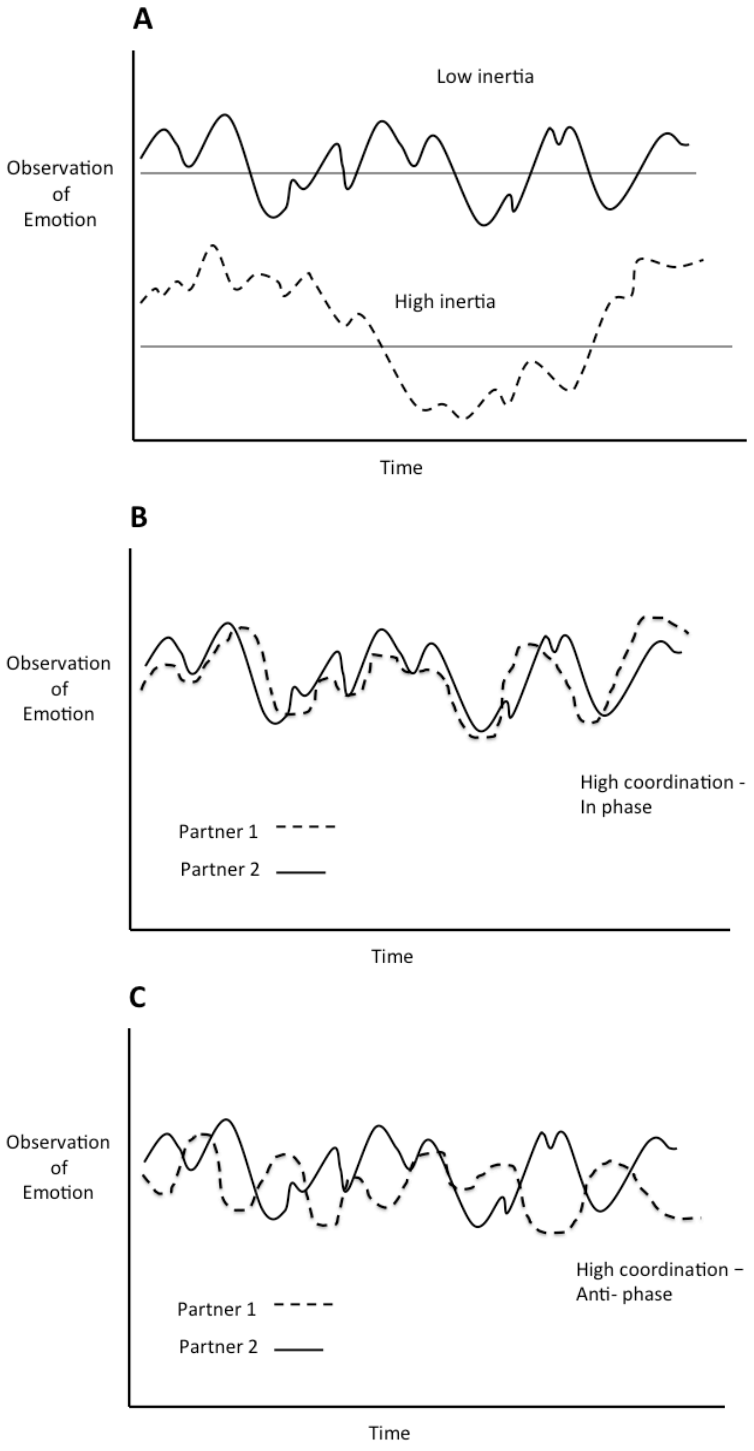

Figure 1. Patterns represented by the Inertia-Coordination model. Panel A shows the interpretation of low versus high positive inertia parameter estimates. Panels B and C show the interpretation of positive and negative coordination parameter estimates respectively.

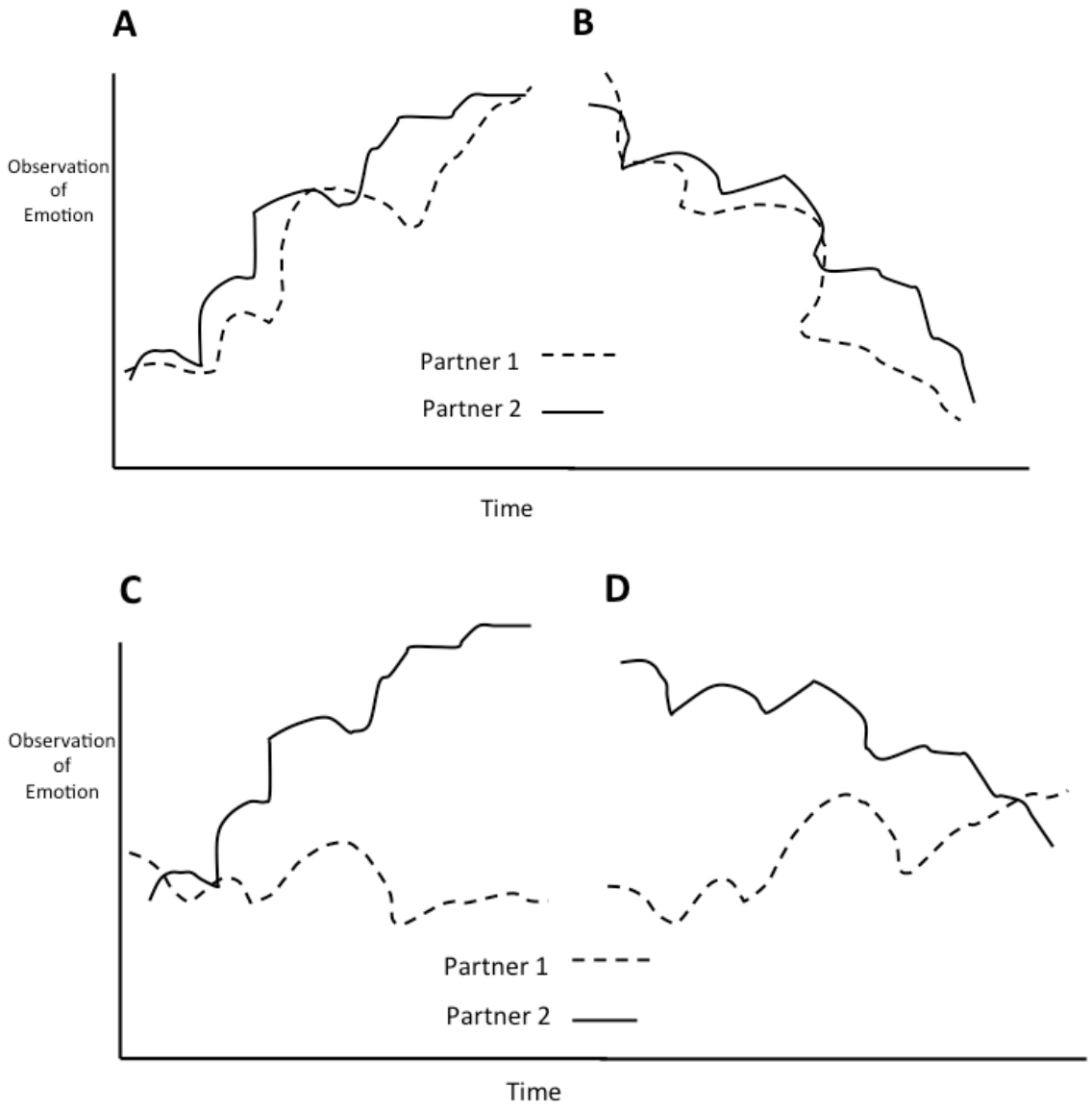

Figure 2. Patterns represented by the Patterned-Slopes model. Panel A shows escalation, Panel B shows de-escalation, Panel C shows divergence and Panel D shows convergence. The four patterns can be distinguished based on whether the two partner's slopes are the same (escalation, de-escalation) or different (divergence, convergence), combined with whether their difference in levels at the end are small (escalation, de-escalation, convergence) or large (divergence).

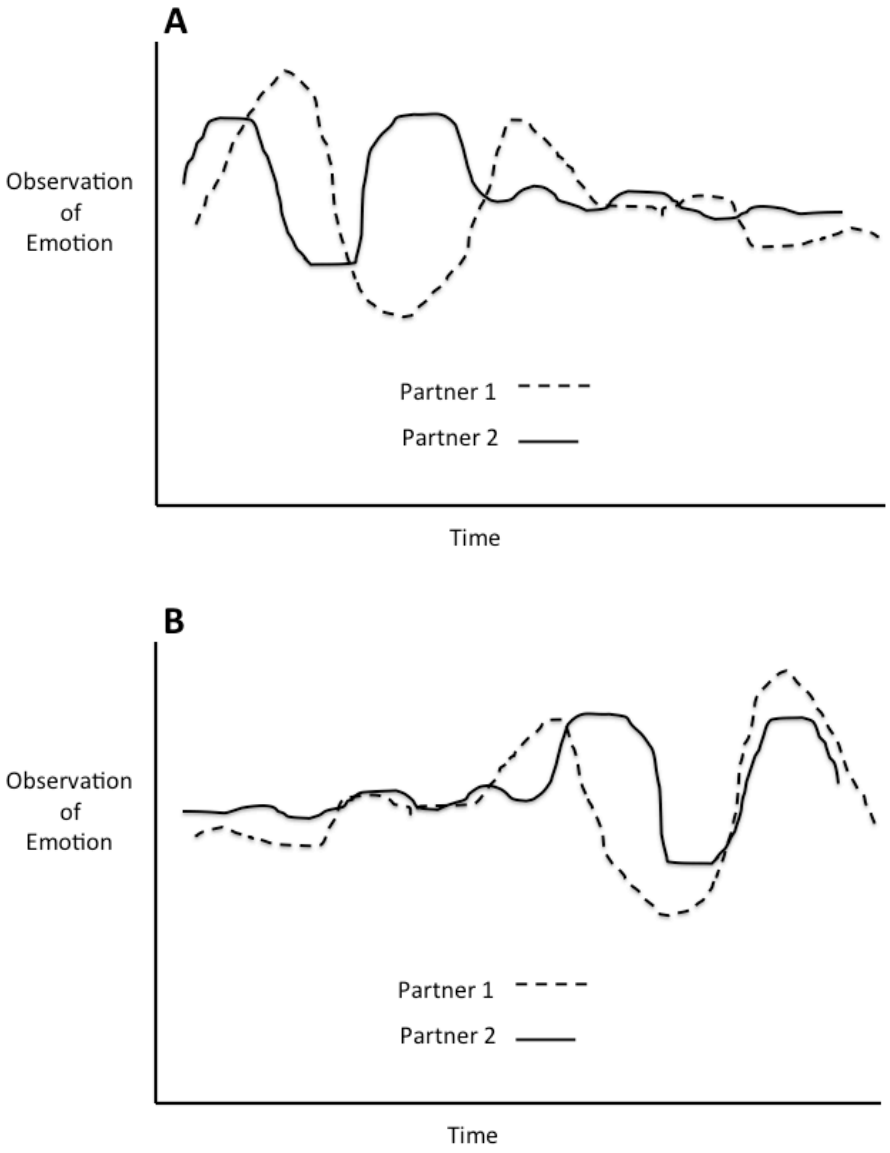

Figure 3. Patterns represented by the Coupled-Oscillator model. Panel A shows mutual partner damping, which suggests co-regulation. Panel B shows mutual partner amplification, which suggests co-dysregulation.