rties: The Patterned-Slopes Model

Emily Butler, eabutler@u.arizona.edu

The Patterned-Slopes model can distinguish between interpersonal escalation (partner's states rising together over time), de-escalation (partner's states decreasing together over time), convergence (partner's states becoming more similar over time) and divergence (partner's states becoming less similar over time). Figure 2 in overview_data_prep.pdf shows what these patterns look like. The Patterned-Slopes model is based on the fact that the four patterns can be distinguished using two linear slopes (one for each partner) and the difference between the partner's at the end of the observation period. Escalation is implied when both partner's have a positive slope and their levels are fairly similar at the end. Likewise, de-escalation is implied when both slopes are negative, but again the difference at the end is small. In contrast, convergence and divergence are implied when partner's have opposite slopes, but for convergence the difference at the end is small, while for divergence it is large. To model this, we start with a standard dyadic growth model in which the time-series state variable is predicted by: 1) separate intercepts for each partner, and 2) separate linear slopes over time, centered at the last observation. The *lm* model used by the *rties* functions is:

```
lm(obs ~ -1 + dist0 + dist1 + dist0:timeEnd + dist1:timeEnd ,
na.action=na.exclude, data=datai)
```

where "obs" is the observed state variable. The "-1" , "dist0" and "dist1" work together to implement a two-intercept model, whereby the overall intercept is omitted and instead separate intercepts are estimated for the 0-level and 1-level of the distinguisher variable provided by the user (for a discussion of this approach see: Kenny, Kashy, & Cook, 2006). The term "timeEnd" is the time variable centered on the last observation point, which can be different for each dyad. For example, if one dyad had 120 observations, their time variable would be centered at time 120, while if another dyad had 135 observations their time variable would be centered at time 135. As a result, the intercepts in the model refer to the predicted level of the state variable at the end of the observation period. Thus the term "dist0" estimates the level of the state variable at the last observation point for the person scored 0 on the distinguishing variable, and "dist1" estimates the same thing for the person scored 1 on the distinguishing variable. The terms "dist0:timeEnd" and "dist1:timeEnd" estimate separate linear slopes over time for each partner.

Note that we estimate this model separately for each dyad (e.g., "datai" is the data from couple "i") and hence it is not a multilevel model, but there are some subtleties here. Due to the nesting of time within people and people in dyads, it is likely that there is autocorrelation and between-partner temporal covariance in the raw observations. The inertia-coordination model directly represents those dependencies. But in choosing a standard dyadic growth model we are making the theoretical claim that the processes that gave rise to the data (and hence the dependencies within- and between- people) can be adequately explained by the two slopes and the two intercepts. If this is the case, the growth model will provide a good description of the data (e.g., we will see relatively high adjusted *R^2s*). In addition, if we are wrong and there are residual dependencies, it would not impact the estimates of the parameters themselves, but only their standard errors (Singer & Willett, 2003). Given that we are not conducting any significance tests on the parameters themselves, this issue is not relevant for our application.

There are two sample size considerations for each of the models implemented in *rties*. The first pertains to the number of observations per dyad that are required, which is largely driven by the complexity of the dynamics to be assessed. The second is the number of dyads required, which is driven by the same issues as in regular multiple regression. The first consideration comes into play when we estimate the dynamics one dyad at a time. Greater complexity requires finer-grained measurement of time and hence more observations per dyad. One advantage of the patterned slopes model is that it is based on the very simple dyadic growth model described above and hence can be estimated with as few as 3 observations per dyad, although having more observations will enhance precision. The second sample size consideration comes into play when we use the estimated dynamics to predict the system variable across dyads using multiple regression models. As described in more detail below, the patterned slopes model for predicting the system variable includes three variables and all their interactions (e.g., a 3-way interaction model), which results in a total of 7 predictors. Thus, to decide the necessary number of dyads you can either apply your favorite rule of thumb along the lines of "*n* observations for each of 7 predictors", consult the literature on sample sizes needed to detect 3-way interactions, or conduct a power analysis.

The first step in an *rties* analysis is to follow the instructions in "overview_data_prep.pdf" to visualize and prepare the data. As described there, the end result is a dataframe (called "data3" in our example) that has the processed data ready for *rties* modeling. The next step, which is often neglected in the literature, is to assess how well the dyadic growth model at the core of our analysis fits the observed temporal data. Our ultimate goal is to either predict outcomes of interest from the dynamics assessed by the patterned slopes model, or to test whether other variables moderate those dynamics. Either way, the results are only meaningful if the dyadic growth model does, in fact, realistically represent the dynamics of the system. We therefore provide a function that fits the model to each dyad's data and returns: 1) a named list ("r2") with the adjusted R^2 for each dyad (e.g., how well the model predicts the observed temporal trajectories of the data), 2) a named list ("paramData") with the parameter estimates for the model (for use later in either predicting, or being predicted by, the system variable), and 3) plots of the predicted values superimposed on the observed values for each dyad. The plots can be accessed from the returned named list ("plots") and they are also automatically saved as a .pdf file in the working directory (this process takes awhile and a blank quartz window may appear, depending on your computer). The function takes the name of the processed dataframe, names for the two levels of the distinguishing variable in the correct order (0 first, then 1; these names will be used to label legends on the plots) and a name for the observed state variable (again, this name will appear on the y-axis of the plots).

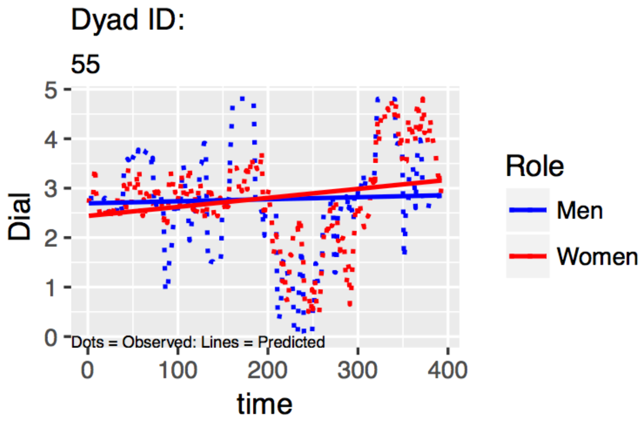
The "indivLinear" function fits the dyadic growth model to each dyad:

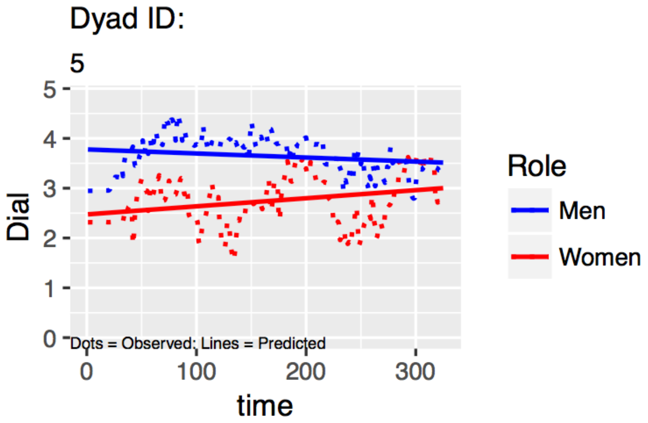
```
indivModels <- indivLinear(data3, "Women", "Men", "Dial")
```

From the results, we can use the "summary" function (or "hist", or any other function) to investigate the adjusted *R^2*s across dyads as an indicator of model fit:

```
summary(indivModels$r2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.7670  0.9030  0.9306  0.9184  0.9499  0.9831
```

Here we see that the dyadic growth model provides an excellent description of the data and accounts on average for about 92% of the variance in the observed state variable (the emotional experience self-reports; see "overview_data_prep.pdf" for a description of the variables in our example). The following figures shows examples of the plots produced, with a poor fit on the left and a good fit on the right. One theoretical point that is made salient by these plots is that the patterned slopes model explicitly ignores the oscillations in the data. If one has a theoretical interest in oscillatory processes, then they should use one of the other models. In contrast, the inertia-coordination and coupled oscillator models explicitly remove information about linear trends, so if that is one's interest then they should use the patterned slopes model.



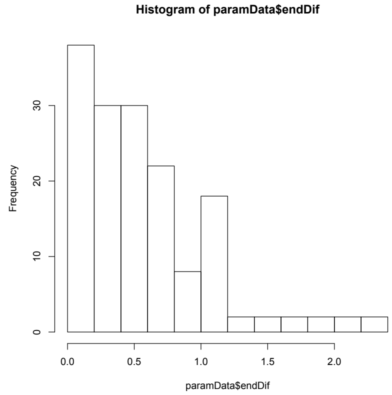


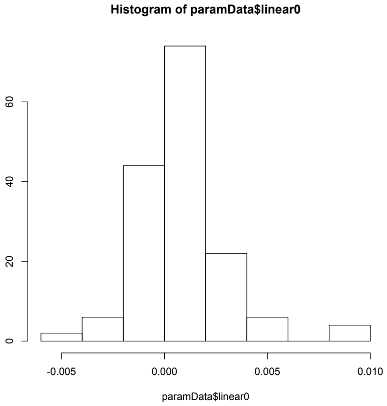
The next step in the analysis is to use combinations of the parameter estimates generated by "indivLinear" to either predict, or be predicted by, the system variable (which is shared unhealthy behavior in our example, called "sub"). At the time of writing, we have only implemented the functions needed to predict the system variable. Future versions of *rties* will also have functions to use it as the predictor, with the parameters from the patterned slopes model as the outcomes. Either way, we start by making the parameter estimates into a stand-alone object for convenience:

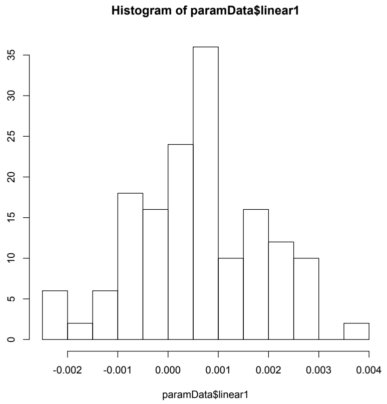
```
paramData <- indivModels$paramData
```

The variables in "paramData" that are relevant for the analysis are: linear0 = the slope estimate for the person scored 0 (partner-0) on the distinguishing variable, linear1 = the slope estimate for the person scored 1 (partner-1) on the distinguishing variable, and endDif = the absolute difference between the partner's estimated level of the state variable at the last observation point. It is a good idea to look at histograms of these to check that there is adequate variance across dyads to make them meaningful as predictors or outcomes of the system variable. Further, if they are to be used as outcomes, they should be fairly normally distributed (although in later versions of *rties* we intend to implement non-Gaussian options). In this example, we see they all have adequate variance, but the endDif is fairly skewed, which is not surprising given that it is an absolute difference.

```
hist(paramData$endDif)
hist(paramData$linear0)
hist(paramData$linear1)
```





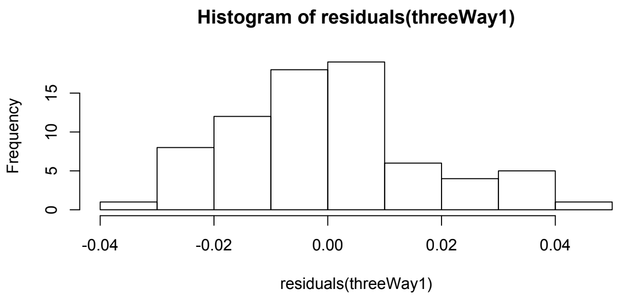


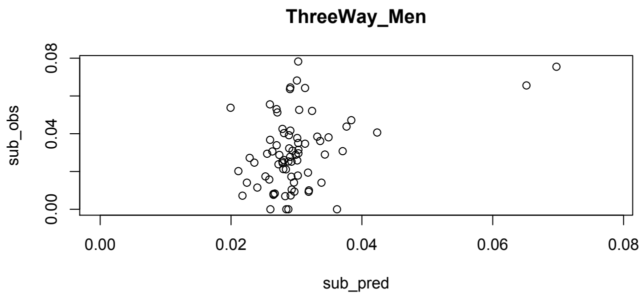
The "linearSysVarOutCompare" function uses the dyadic growth parameter estimates to predict the system variable across dyads using multiple regression models. It does so for three sets of predictors: 1) main effects only (endDif + linear0 + linear1), 2) a two-way slope interaction, controlling for the end-difference (endDif + linear0 * linear1), and 3) a three-way interaction (endDif * linear0 * linear1). The models are currently estimated separately for the two levels of the distinguishing variable, but in future versions of *rties* we will combine them and provide significance tests and model comparisons to assess whether the results differ between the two types of partner. In the present example, the system variable (shared unhealthy behavior) is assessed at the dyad level (e.g., both partners have the same score) and the predictor sets for the patterned slopes model include parameters from both partners. In this situation the results are identical for both types of partner. If, however, the system variable was assessed at the individual level (e.g., a person's depression score) or the predictor sets did not contain both partner's parameters (as is the case for the coupled oscillator model), then results can vary by partner type.

The function takes the name of the dataframe containing the parameter estimates ("paramData" in our example), names for the two levels of the distinguishing variable in the correct order (0 first, then 1; these names will be used to label legends on the plots) and a name for the observed system variable (again, this name will appear on the y-axis of the plots). The function returns a named list including: 1) the lm objects containing the full results for each model for each level of the distinguishing variable (called "models0" and "models1"), 2) anova output for each model for each level of the distinguishing variable (called "anovas0" and "anovas1"), 3) summary output for each model for each level of the distinguishing variable (called "summaries0" and "summaries1") and 4) adjusted *R^2* information for each model for each level of the distinguishing variable (called "adjustR20" and "adjustR21"). The function also displays histograms of the residuals and plots of the predicted values against observed values for each model for each level of the distinguishing variable.

```
output <- linearSysVarOutCompare(paramData, "Women", "Men", "sub")
```

Here is an example of the plots produced.





We can see that the residuals of the three-way interaction model predicting the system variable for men are fairly normally distributed around zero, suggesting that model assumptions have been met. We also see that the predicted "sub" scores appear to have a weak positive association with the observed "sub" scores, with 2 clear outliers that may influence results. We can formalize this by looking at the adjusted

R^2 results for each of the three models (main effects, 2-way interaction, 3-way interaction) predicting “sub” :

```
output$adjustR20

$main0R2
[1] 0.03885873

$twoWay0R2
[1] 0.03667398

$threeWay0R2
[1] 0.06344231
```

We see that the 3-way interaction model accounts for about 6% of the variance in the shared unhealthy behaviors, followed by the other two models which both account for about 4% of the variance. We next consider the results of the anovas for the 0 level of the distinguishing variable for each model (recall that in this situation the results are identical for both types of partners):

```
output$anovas0

$main0Anova
Anova Table (Type III tests)

Response: sysVar
             Sum Sq Df F value  Pr(>F)
endDif    0.0000045  1  0.0131 0.90914
linear0   0.0017037  1  5.0049 0.02846 *
linear1   0.0009899  1  2.9078 0.09259 .
Residuals 0.0238293 70
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

$twoWay0Anova
Anova Table (Type III tests)

Response: sysVar
                   Sum Sq Df F value  Pr(>F)
endDif          0.0000029  1  0.0084 0.92716
linear0         0.0010563  1  3.0958 0.08293 .
linear1         0.0012610  1  3.6960 0.05867 .
linear0:linear1 0.0002870  1  0.8412 0.36224
Residuals       0.0235422 69
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

$threeWay0Anova
Anova Table (Type III tests)

Response: sysVar
                          Sum Sq Df F value  Pr(>F)
```

```
endDif                    0.0000142  1  0.0428 0.83665  
linear0                   0.0000288  1  0.0867 0.76928  
linear1                   0.0007167  1  2.1607 0.14633  
endDif:linear0            0.0000002  1  0.0006 0.98103  
endDif:linear1            0.0002390  1  0.7204 0.39909  
linear0:linear1           0.0018976  1  5.7206 0.01962 *
endDif:linear0:linear1    0.0014020  1  4.2267 0.04375 *
Residuals                 0.0218929 66                  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

In keeping with the fact that the 3-way interaction model had the highest adjusted *R^2*, we also see it has a significant 3-way interaction effect. The next step for interpreting these effects is to plot the model predicted means and standard errors for the system variable, for one of the partner types, at user-specified low and high levels of the predictors (e.g., low and high centering values for the end difference and two slopes estimates). The goal in choosing centering values is that they should indicate levels of the predictor(s) that are of interest due to being representative of meaningful values in the population. If the predictor(s) are normally distributed, then minus and plus one standard deviation, or the 25th and 75th percentile, make good centering values indicative of typical cases at the low and high ends of the distribution. One might also choose theoretically or practically meaningful values, however, such as 25% below and above some clinical cut off. Or, if the predictor(s) are not normally distributed, then histograms can be used to select reasonable values. For example, when a variable is zero-inflated (e.g., there are a large number of zero observations), then zero becomes a good choice for one of the centering values since it is highly representative of part of the population.

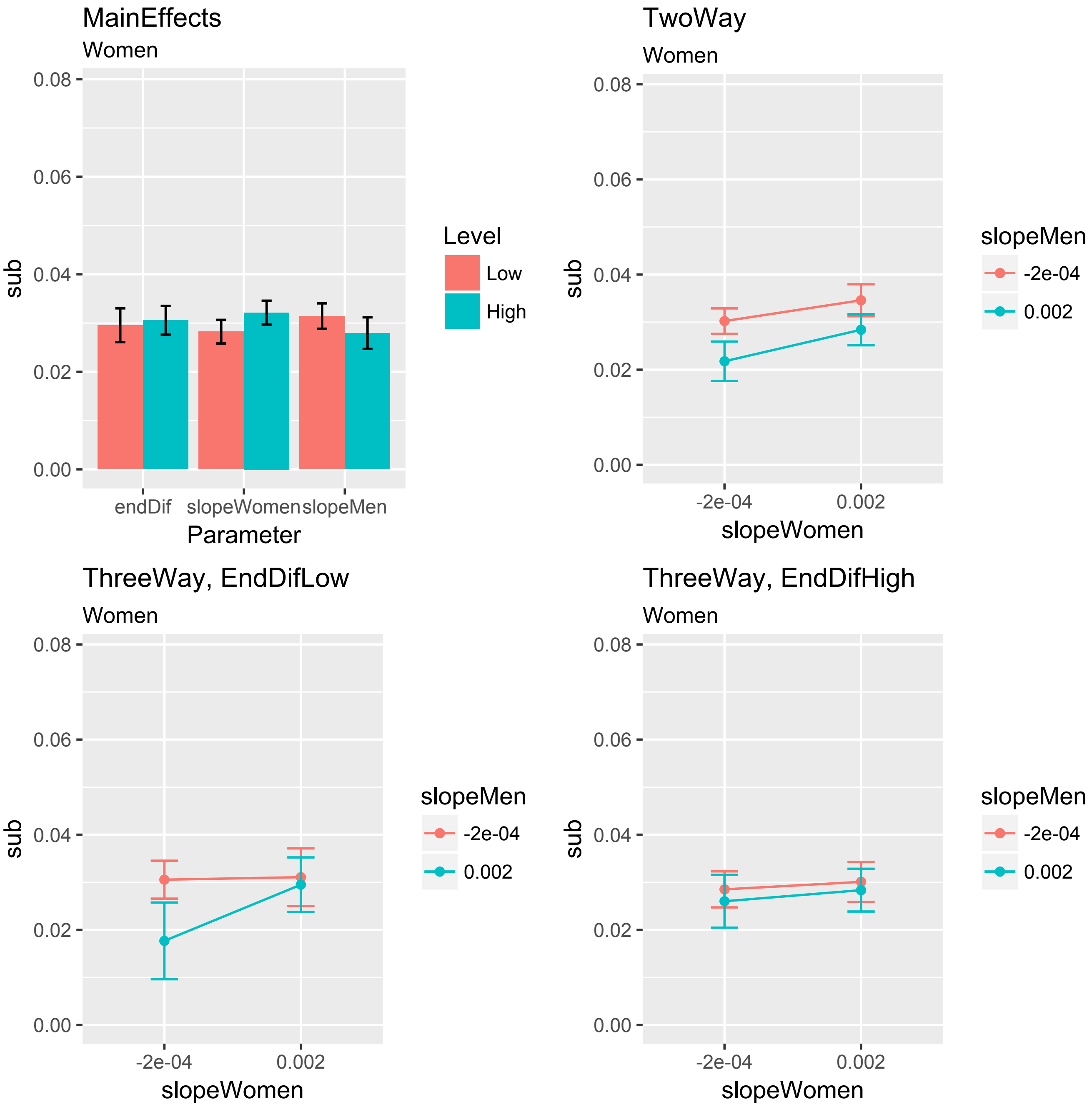
The “linearSysVarOutPlots” function produces plots for each of the 3 models (main effects, 2-way interaction, 3-way interaction). It takes as arguments the dataframe containing the parameter estimates (“paramData” in our example), several vectors of centering values indicating prototypical low, medium and high levels of the end difference and the two slope parameter estimates (the centering values for “medium” are needed as control variables in the analysis), a name to appear on the y-axis indicating the system variable, a 0 or 1 to indicate which level of the distinguishing variable to plot the results for, and names for the two levels of the distinguishing variable in the correct order (0 first, then 1; these names will be used to label legends on the plots). As we saw in the histograms of the predictors earlier (e.g., the end difference and two slope parameter estimates) that the slope estimates were normally distributed, but the end difference estimate was skewed towards zero. We therefore chose to use the 25th, 50th and 75th percentiles as our centering values for the slopes, but for the end difference we chose 0, the median (.45) and 1 based on visual inspection of its distribution. The following syntax creates the vectors with the centering values and then creates the plots:

```
centEnd <- c(0, .45, 1)
centLine0 <- c(-.0002, 0, .002)
centLine1 <- c(-.0002, 0, .002)

modelPlots <- linearSysVarOutPlots(paramData, centEnd, centLine0,
 centLine1, "sub", 0, "Women","Men")
```

There are two choices for viewing the plots. When the function is run, it automatically displays the plots for all 3 models in one window (see below for what this looks like). It is also possible to view the plots for one model at a time, because they are stored in the “modelPlots$grobs” named list (the names are “mainEffects”, “twoWay”, “threeWayL” and “threeWayH”).

In the following figure, the upper left panel shows the model predicted means and standard errors of the system variable (“sub”) predicted by the main effects only model. The upper right shows the results from the 2-way interaction model. The bottom 2 panels show the results from the 3-way interaction model, with the end difference at either the low (left) or high (right) centering value.



The results from the 3-way interaction model show that the lowest predicted levels of the shared unhealthy behaviors (“sub”) occur in couples who have fairly similar levels of emotional experience at the end of the interaction (e.g., the end difference is low) and where the woman’s slope is flat or slightly negative, but the man’s slope is positive. This combination implies a pattern of convergence, since the partner’s emotional experiences were changing in opposite directions over time, but ended up at a similar level.

Finally, full information for each of the models can be obtained from the "summary" output of the "linearSysVarOutCompare" function that we used earlier. We focus here on the results of the 3-way interaction for the 0 level of the distinguishing variable (recall that in this situation the results are identical for both types of partners):

```
output$summaries0

$threeWay0Summary

Call:
lm(formula = sysVar ~ endDif * linear0 * linear1, data = basedata0)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.036199 -0.012254 -0.000975  0.006729  0.047998 

Coefficients:
                               Estimate Std. Error t value Pr(>|t|)
Intercept                     2.952e-02  3.744e-03   7.884 4.26e-11
***
endDif                       -1.088e-03  5.255e-03  -0.207   0.8366
slope_Women                   7.028e-01  2.386e+00   0.295   0.7693
slope_Men                    -5.382e+00  3.662e+00  -1.470   0.1463
endDif:slope_Women            4.534e-02  1.899e+00   0.024   0.9810
endDif:slope_Men              4.279e+00  5.041e+00   0.849   0.3991
slope_Women:slope_Men         2.333e+03  9.754e+02   2.392   0.0196 *
endDif:slope_Women:slope_Men -2.177e+03  1.059e+03  -2.056   0.0438 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01821 on 66 degrees of freedom
Multiple R-squared:  0.1532,  Adjusted R-squared:  0.06344 
F-statistic: 1.706 on 7 and 66 DF,  p-value: 0.1227
```

The most notable result here is that, despite the significant interaction, the overall model does not account for a significant amount of variance in the shared unhealthy behaviors (see F test at the bottom), suggesting that we must interpret our results as representing a very small and potentially unreliable effect.