rties: The Inertia-Coordination Model

Emily Butler, eabutler@u.arizona.edu

The Inertia-Coordination model represents the within-person pattern of inertia, which is defined as the extent to which a person's state can be predicted from his or her own state at a prior time point, and the between-person pattern of coordination, which is defined as the extent to which one partner's state can be predicted from their partner's state either concurrently or time-lagged (Reed, Randall, Post, & Butler, 2013). Specifically, the time-series state variable is predicted by the following: 1) separate intercepts for each partner, 2) a person's own state variable at a prior time point, which gives two "inertia" estimates, one for each partner, and 3) the person's partner's state variable at the same time point, which gives two "coordination" estimates, again one for each partner. The *lm* model used by the *rties* functions is:

```
lm(obs_deTrend ~ -1 + dist0 + dist1 + dist0:obs_deTrend_Lag +
dist0:p_obs_deTrend + dist1:obs_deTrend_Lag + dist1:p_obs_deTrend,
na.action=na.exclude, data=datai)
```

where "obs_deTrend" is the observed state variable with individual linear trends removed (e.g., it is the residuals from each person's state variable predicted from time). 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 "dist0:obs_deTrend_Lag" estimates the inertia parameter for the person scored 0 on the distinguishing variable (e.g., "obs_deTrend_Lag" is the person's own de-trended observed state variable lagged by how ever many steps the user specifies), "dist1:obs_deTrend_Lag" estimates the inertia parameter for the person scored 1 on the distinguishing variable, "dist0:p_obs_deTrend" estimates the coordination parameter for the person scored 0 on the distinguishing variable (e.g., p_obs_deTrend is the person's partner's de-trended observed state variable and so indicates the extent to which the 0-level person is predicted by their partner), and "dist1:p_obs_deTrend" estimates the coordination parameter for the person scored 1 on the distinguishing variable. The model is estimated separately for each dyad (e.g., "datai" is the data from couple "i").

The primary difference between our implementation of the model and most other variants is that we only include concurrent coordination, while it is more common to focus on time-lagged coordination. The reason for our choice is that if the state variables are oscillating, then between-partner lagged associations will oscillate as well (for discussion of a similar issue when choosing the measurement interval see: Boker & Nesselroade, 2002). This dependence on the lag length makes interpretation problematic unless one has a strong theory about the temporal processes at work. Technically, the same problem exists with inertia, but pragmatically we have not observed the issue in our own or published research. Specifically, reported inertia estimates are typically positive, while coordination estimates show greater variance. By restricting the model to concurrent coordination, interpretation is simplified (see Figure 1 in overview_data_prep.pdf). Higher positive inertia implies slower fluctuations in emotional responding. If one obtains a negative inertia estimate, however, it implies that the observed variable is oscillating back and forth between each lag. For the between-person coordination parameters, a positive estimate implies an in-phase pattern, such that when one partner is high on some measure of emotion, so is their partner, while a negative parameter implies an anti-phase pattern, such that when one partner is high the other partner is low (Randall, Post, Reed, & Butler, 2013; Reed et al., 2013; Wilson et al., 2018).

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 inertia-coordination model is that it is fairly simple and hence requires relatively few observations per dyad. The exact number will depend on how much variance there is over time, both within-people and between-partners, but it is likely to provide good results with as few as 5 observations per dyad (someone should do a simulation study of this!). The number of observations will impact the choice of lag, however, because for each lag you lose one observation point (e.g., if the lag is two steps, then the inertia estimates will be based on the total number of observations per dyad minus 2). Thus choosing the lag relies on a combination of theory, prior research, how quickly you expect the phenomenon of interest to be changing and how many observation time points there were. One strength of *rties* is that it makes it very easy to alter the lag length and observe the impact on the results (see setting the lag length in “overview_data_prep.pdf”), which is helpful for developing an intuitive understanding of inertia.

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 above, the inertia-coordination model includes four parameters, which become the predictors of the system variable. 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 4 predictors”, or you can 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 different variants of the inertia-coordination model fit the observed temporal data. Our ultimate goal is to either predict outcomes of interest from the dynamics assessed by the inertia-coordination model, or to test whether other variables moderate those dynamics. Either way, the results are only meaningful if the inertia-coordination model does, in fact, realistically represent the dynamics of the system. We therefore provide a set of functions that fit three versions of the model to each dyad's data and return: 1) named lists (“r2”) with the adjusted R^2 for each dyad for each model (e.g., how well each model predicts the observed temporal trajectories of the data), 2) named lists (“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 three models are: 1) an inertia-only model, 2) a coordination-only model, and 3) the full inertia-coordination model. Each 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 first model, inertia-only, has only the inertia parameters as predictors. The *lm* model used is:

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

The “indivInert” function fits the inertia-only model to each dyad:

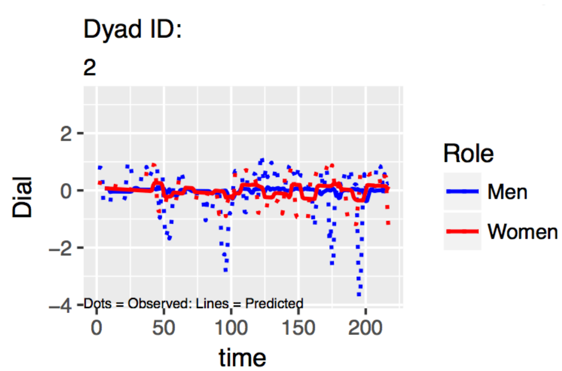
```
indivModelsInert <- indivInert(data3, "Women", "Men", "Dial")
```

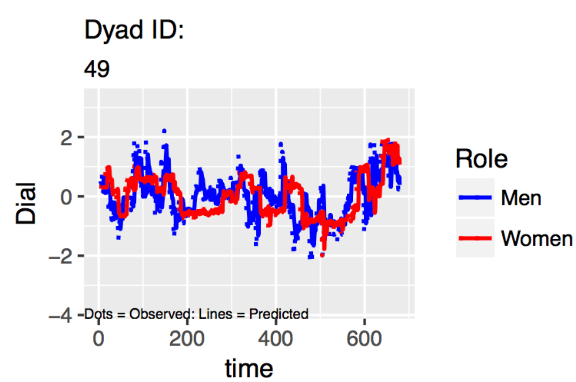
From the results, we can use the “summary” function (or “hist”, or any other function) to investigate the adjusted *R^2*s across dyads as an indicator of model fit. The results for the inertia-only model are:

```
summary(indivModelsInert$r2)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.02383 0.25125 0.35658 0.36360 0.47824 0.73168
```

Here we see that the inertia-only model accounts on average for about 36% 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), and has a fairly large range, with some dyads very well described and others quite poorly.

The plots of predicted values overlaid on observed values will be automatically written to the working directory in a file called “inertPlots.pdf” or they can be accessed as a named list called “plots” from the object created by “indivInert” (e.g., in our example indivModelsInert$plots holds the plots). The following figures shows examples, with a poor fit on the left and a good fit on the right.





The second model, coordination-only, has only the coordination parameters as predictors. The *lm* model used is:

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

The “indivCoord” function fits the coordination-only model to each dyad:

```
indivModelsCoord <- indivCoord(data3, "Women", "Men", "Dial")
```

Again, we can investigate the adjusted *R^2*s across dyads as an indicator of model fit. The results show that the coordination-only model does not fit as well as the inertia-only model, with on average only about 12% of the variance in the observed state variable explained. This is perhaps not surprising, since in the extreme (lag = 0) the inertia estimates are identical to the raw data, while the coordination estimates do not have that dependence. The results are:

```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-0.007517  0.028145  0.070321  0.122174  0.185345  0.480332
```

Despite the generally poor fit of the coordination-only model, as shown by the plot on the right below, some dyads are quite well described, which leads us to wonder what factors determine which couples are characterized by coordination versus which are not. We see this as an example of how the *rties* package can help to generate new research questions by making it easy to visualize the data and model results.

The third model is the full inertia-coordination model. The *lm* model used is:

```
lm(obs_deTrend ~ -1 + dist0 + dist1 + dist0:obs_deTrend_Lag +
dist0:p_obs_deTrend + dist1:obs_deTrend_Lag + dist1:p_obs_deTrend,
na.action=na.exclude, data=datai)
```

The “indivInertCoord” function fits the full inertia-coordination model to each dyad:

```
indivModelsInertCoord <- indivIndivCoord(data3, "Women", "Men",
 "Dial")
```

The *R^2* estimates show that the full model fits very similarly to the inertia only model:

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0228  0.2737  0.3914  0.3952  0.5185  0.7401
```

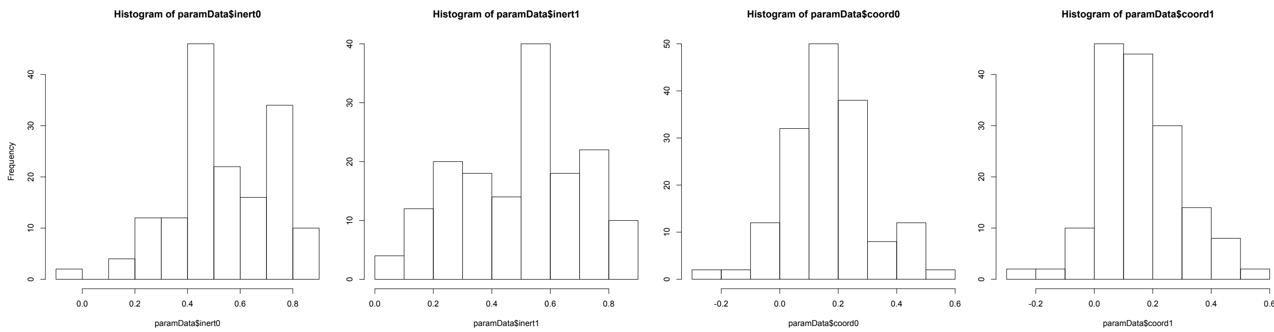
The comparative assessment of model fit we have just done provides information somewhat similar to a consideration of measurement error (we say “somewhat similar” because the similarity is at the conceptual level, not the mathematical or formal level). In subsequent analyses we will be using the inertia and coordination parameter estimates as predictors or outcomes of the system variable in multiple regression models. The quality of fit for the inertia-coordination model gives some indication of how accurately those parameter estimates are assessing the underlying dynamics. We have seen that the inertia estimates appear to be capturing more systematic variance in the observed time-series data than the coordination estimates. Thus, we have more confidence that any results using inertia will be robust than we do for coordination. Nevertheless, the observed time-series for some couples were well characterized by coordination alone, so we have faith that there is some signal in these estimates. Furthermore, they are of central theoretical interest and so we believe they should not be completely ignored (as we might if the *R^2* for the coordination-only model were uniformly very low or they were not theoretically important).

The next step in the analysis is to use the parameter estimates generated by “indivInertCoord” 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 inertia and coordination parameters as the outcomes. Either way, we start by making the parameter estimates into a stand-alone object for convenience:

```
paramData <- indivModelsInertCoord$paramData
```

The variables in “paramData” that are relevant for the analysis are: inert0 = the inertia estimate for the person scored 0 (partner-0) on the distinguishing variable, inert1 = the inertia estimate for the person scored 1 (partner-1) on the distinguishing variable, coord0 = the coordination estimate for partner-0, and coord1 = the coordination estimate for partner-1. 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 are fairly normally distributed with adequate variance.

```
hist(paramData$inert0)
hist(paramData$inert1)
hist(paramData$coord0)
hist(paramData$coord1)
```

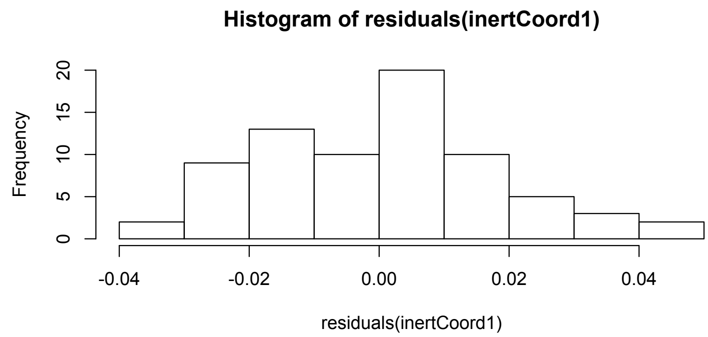


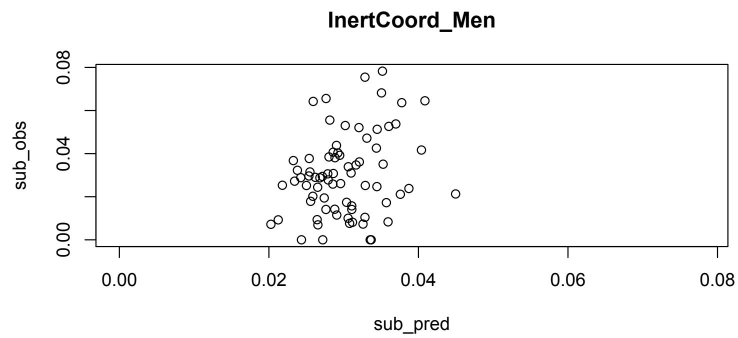
The “inertCoordSysVarOutCompare” function uses the inertia-coordination parameter estimates to predict the system variable across dyads using multiple regression models. It does so for three sets of predictors: 1) inertia-only, 2) coordination-only, and 3) the full model. The models are currently estimated 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 inertia-coordination 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 do 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 <- inertCoordSysVarOutCompare(paramData, "Women", "Men",
"sub")
```

Here is an example of the plots produced. We can see that the residuals of the full inertia-coordination 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. We can formalize this by looking at the adjusted *R^2* results for each of the three models (inertia only, coordination only, full model) predicting "sub" :






```
output$adjustR20

$inert0R2
[1] 4.714654e-05

$coord0R2
[1] 0.03450754

$inertCoord0R2
[1] 0.01214933
```

We see that the coordination only model accounts for about 3% of the variance in the shared unhealthy behaviors, followed by the full model, which accounts for about 1%, and the inertia only model, which is essentially non-explanatory with an adjusted *R^2* of zero.

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

$inert0Anova
Anova Table (Type III tests)

Response: sysVar
             Sum Sq Df F value Pr(>F)
inert0    0.0000236  1  0.0666 0.7971
inert1    0.0006866  1  1.9386 0.1682
Residuals 0.0251457 71

$coord0Anova
Anova Table (Type III tests)

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

```
coord0    0.0002525  1  0.7385 0.39303
coord1    0.0015449  1  4.5178 0.03702 *
Residuals 0.0242791 71
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

$inertCoord0Anova
Anova Table (Type III tests)

Response: sysVar
             Sum Sq Df F value Pr(>F)
inert0    0.0000034  1  0.0097 0.9220
coord0    0.0000947  1  0.2706 0.6046
inert1    0.0001372  1  0.3921 0.5333
coord1    0.0008878  1  2.5373 0.1158
Residuals 0.0241416 69
```

In keeping with the fact that the coordination-only model had the highest adjusted *R^2*, we also see it produces the only significant effect, with the coordination parameter for the men (coord1: recall the men were the 1's on the distinguishing variable) being a significant predictor of the unhealthy behaviors. To understand the direction of this effect we look at the summary output for the coordination-only model (similar summary results are produced for each of the three models):

```
output$summaries0

$coord0Summary

Call:
lm(formula = sysVar ~ coord0 + coord1, data = basedata0)

Residuals:
      Min        1Q    Median        3Q       Max
-0.032967 -0.014282  0.001419  0.010242  0.044563

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
intercept    0.026658   0.003508   7.599 9.24e-11 ***
coord_Women -0.014501   0.016874  -0.859    0.393
coord_Men    0.038160   0.017953   2.126    0.037 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01849 on 71 degrees of freedom
Multiple R-squared:  0.06096, Adjusted R-squared:  0.03451
F-statistic: 2.305 on 2 and 71 DF,  p-value: 0.1072
```

These results show that higher men's coordination (e.g., how well they are predicted by their partners) predicts higher levels of shared unhealthy behaviors. Specifically, for each 1 unit increase in coordination, "sub" is predicted to increase by .038 units. However, in keeping with the low *R^2* we saw before, the overall model does not account for a significant amount of variance in the outcome (see F-statistic at the bottom of the summary output above).

At this point, we would probably forsake this analysis with the conclusion that inertia is unrelated to shared unhealthy behavior and coordination is at best very weakly associated with it. But as a demonstration, we continue with the additional analyses available in *rties*. If we had better evidence for coordination as a meaningful predictor, it would be useful to compare the predictive ability of the coordination-only model to the full model, which can be accomplished with the following syntax (recall here we are focused on the results for the person scored 0 on the distinguishing variable, but in this situation the results for both partner types is identical):

```
anova(output$models0$coord0, output$models0$inertCoord0)
```

The results below show that the more complex full model (Model 2, which has fewer residual degrees of freedom) does not provide a significant reduction in the residual sums of squares when compared to the simpler coordination-only model (Model 1), a result that would lead us to prefer the simpler model.

```
Analysis of Variance Table

Model 1: sysVar ~ coord0 + coord1
Model 2: sysVar ~ inert0 + coord0 + inert1 + coord1
  Res.Df      RSS Df  Sum of Sq      F Pr(>F)
1     71 0.024279
2     69 0.024142  2 0.00013752 0.1965  0.822
```

Finally, 3 functions are provided to plot the results for each of the 3 models (inertia-only: "inertSysVarOutPlots", coordination-only: "coordSysVarOutPlots", and the full model: "inertCoordSysVarOutPlots") for one or the other type of partner. Specifically these plots show 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 inertia and coordination parameter 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.

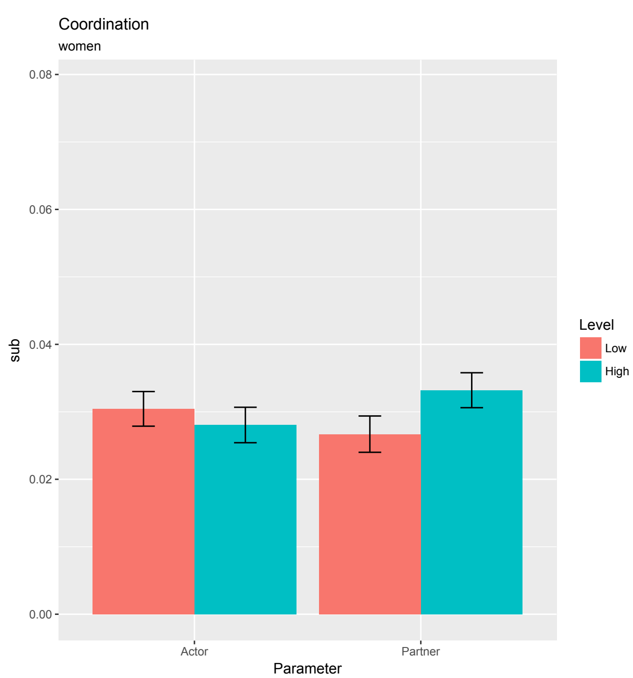
One other feature of these plots is that they show the model predictions with respect to both the person's own parameter estimates (actor effects) and their partner's parameter estimates (partner effects). For example, if we consider the results of the coordination-only model for the women, then the significant effect of men's coordination that we observed earlier would take the form of a partner effect. In contrast, if we considered the plots for the men, it would take the form of an actor effect. In the present example, the two are symmetric, but if the outcome was assessed at the level of the individual that would not necessarily by the case.

Each of the 3 functions 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 inertia and coordination 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 when we looked at the histograms of the predictors (e.g., the inertia and coordination parameter estimates) earlier, they were all fairly normally distributed, so we decided to use the 25th, 50th and 75th percentiles as our centering values. The following syntax calculates those values and assigns them to vectors:

```
centInert0 <- as.numeric(quantile(paramData$inert0, probs= c(.25, .5,
 .75), na.rm=T))
centInert1 <- as.numeric(quantile(paramData$inert1, probs= c(.25, .5,
 .75), na.rm=T))
centCoord0 <- as.numeric(quantile(paramData$coord0, probs= c(.25, .5,
 .75), na.rm=T))
centCoord1 <- as.numeric(quantile(paramData$coord1, probs= c(.25, .5,
 .75), na.rm=T))
```

We provide the plots for the coordination-only model for women as an example, but the process would be similar for the other two models, or for men:

```
cPlots <- coordSysVarOutPlots(paramData, centCoord0, centCoord1,
 "sub", 0, "women", "men")
```



The plot makes the men's coordination effect clear, such that men at the 75th percentile of the distribution of coordination are predicted to be in relationships with the highest shared unhealthy behaviors. Further, since the 75th percentile of coordination was a positive value (see histograms earlier) we know that this is an example of in-phase coordination.