qid: 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 prior 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_Lag + dist1:obs_deTrend_Lag +
dist1:p_obs_deTrend_Lag, 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 level-0 and level-1 of the distinguisher variable provided by the user (for a discussion of this approach see: Kenny, Kashy, & Cook, 2006). The terms "dist0:obs_deTrend_Lag" and "dist1:obs_deTrend_Lag" estimate the inertia parameters for the partners scored 0 and 1 on the distinguishing variable respectively (e.g., "obs_deTrend_Lag" is the person's own de-trended observed state variable lagged by how ever many steps the user specifies). Similarly, the terms "dist0:p_obs_deTrend_Lag" and "dist1:p_obs_deTrend_Lag" estimate the coordination parameter for the partners scored 0 and 1 on the distinguishing variable respectively (e.g., p_obs_deTrend_Lag is the person's partner's de-trended observed state variable lagged by how ever many steps the user specifies). The model is estimated separately for each dyad (e.g., "datai" is the data from couple "i").

Positive inertia estimates imply slower fluctuations of the state variable, while negative inertia estimates imply that the observed variable is oscillating back and forth between each lag (see Figures in overview_data_prep). For the between-person coordination parameters, a positive estimate implies an in-phase pattern, such that when one partner is high on the observed state variable, so is their partner at the specified lag, while a negative parameter implies an anti-phase pattern, such that when one partner is high the other partner is low at the specified lag (Randall, Post, Reed, & Butler, 2013; Reed et al., 2013; Wilson et al., 2018).

Choosing the Lag Length

One complexity in using the Inertia-Coordination model is that the results are highly dependent upon the chosen lag length (for discussion of a similar issue when choosing the measurement interval see: Boker & Nesselroade, 2002). This dependence on the lag makes interpretation problematic unless one has a strong theory about the temporal processes at work. If the state variables are oscillating at all, the inertia and coordination parameter estimates will also oscillate depending on lag. Thus choosing the lag relies on a combination of theory, prior research, and how quickly you expect the phenomenon of interest to be changing (see Thorson, West & Mendes, 2017, for additional discussion). In addition, if the process is fairly stable, inertia will be high across a long lag time and will dominate any results. The stronger the inertia is, the weaker the coordination will be (this results from the behavior of multiple regression models, where the coordination parameters can only account for independent variance in the outcome after accounting for inertia). Thus part of the decision depends upon how much of the observed state variables behavior you want to be explained by within-person inertia processes versus how much you prefer to prioritize between-person coordination. Another consideration is how many observation time points there were. For each lag you lose one observation point (e.g., if the lag is two steps, then the parameter estimates will be based on the total number of observations per dyad minus two). 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 and coordination.

Sample Size Considerations

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, while the second is the number of dyads required. 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 second sample size consideration comes into play when we use the estimated dynamics to either predict the system variable, or be predicted by it. In both cases, the system variable can be either a dyadic variable (e.g., both partners have the same score, as in relationship length) or an individual variable (e.g., partners can have different scores, as in age). **Note that the current version of *rties* takes a different approach at this step** and makes use of latent profile analysis for representing the dynamics, compared to the archival version used in *qid* and reported here. Thus you should consult the *rties* documentation for actual analyses. For the archival *qid* approach, in the case of predicting a dyadic system variable, a multiple regression model is appropriate. In this case, the shared system variable is predicted by both partners' inertia and coordination parameter estimates and you can use your favorite rule of thumb along the lines of "*n* observations for each of 4 predictors" to choose your sample size, or you can conduct a power analysis. The situation is more complicated when the system variable is assessed at the individual level, or when it is the predictor of the dynamics. In the former case, the partner's system variable scores are predicted by actor- and partner- effects of the inertia and coordination parameter estimates using a cross-sectional random-intercept dyadic model. In the latter case, the set of inertia and coordination parameter estimates are predicted by the system variable using a multivariate correlated-residuals dyadic model. If the system variable is at the individual level, both actor- and partner- effects are included in the model. Estimating statistical power for all of these models is non-trivial and would be a good topic for someone to write a paper about (for a great new tool that could be used for this purpose, see the R package "*simr*").

Assessing Model Fit

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 "data2" 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 predict 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 named lists that include: 1) the adjusted R^2 for each dyad for each model ("R2" e.g., how well each model predicts the observed temporal trajectories of the data), 2) the parameter estimates for the model ("paramData", 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 list entry ("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:

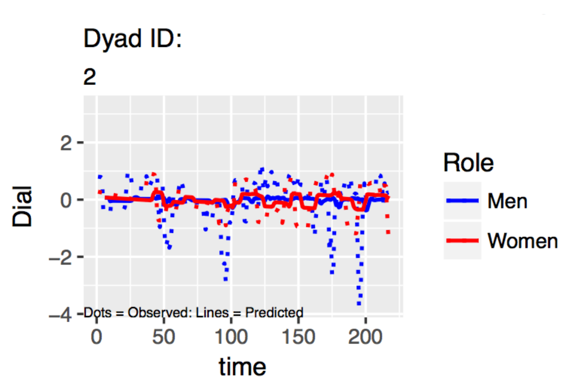
```
indivModelsInert <- indivInert(data2, "Women", "Men", "Dial")
```

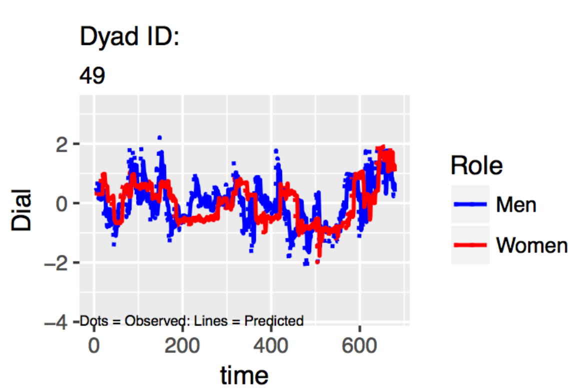
From the results, we can use the "summary" function (or "hist", or any other function) to investigate the adjusted *R^2*s across dyads as an indicator of model fit. The results for the inertia-only model are:

```
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 from the returned list (called "plots" 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_Lag +
dist1:p_obs_deTrend_Lag, na.action=na.exclude, data=datai)
```

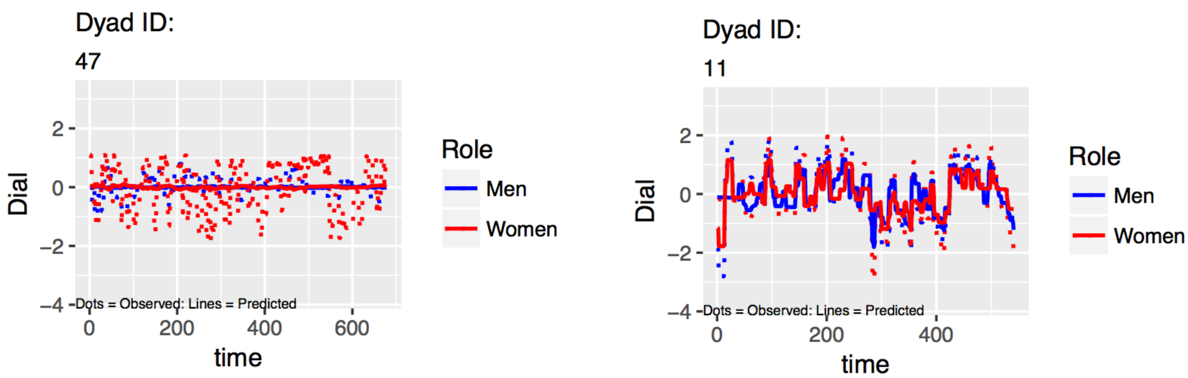
The "indivCoord" function fits the coordination-only model to each dyad:

```
indivModelsCoord <- indivCoord(data2, "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 13% 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  0.028022  0.081781  0.126853  0.204078  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_Lag + dist1:obs_deTrend_Lag +
dist1:p_obs_deTrend_Lag, na.action=na.exclude, data=datai)
```

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

```
indivModelsInertCoord <- indivIndivCoord(data2, "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  0.2710  0.3977  0.3970  0.5186  0.7401
```

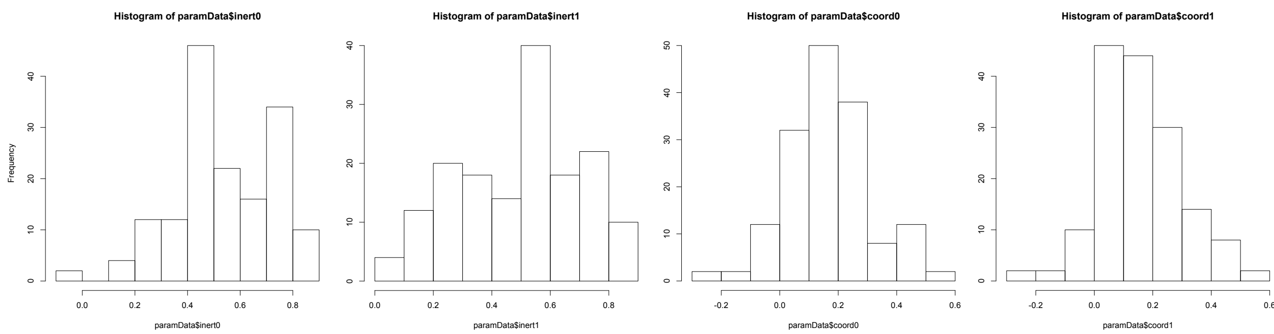
Associations Between the Dynamics and the System Variable

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"). 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. In this example, we see they are fairly normally distributed with adequate variance.

```
histAll(paramData)
```

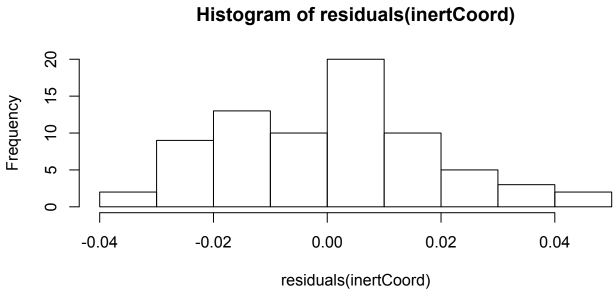


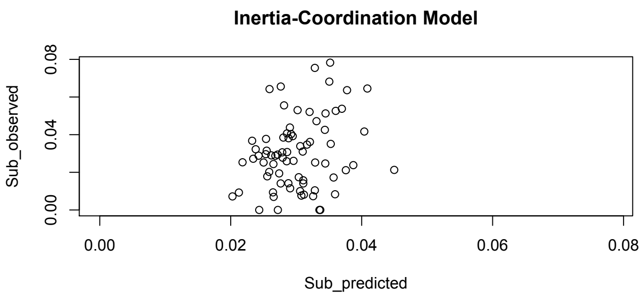
The "inertCoordSysVarOut" function uses the inertia-coordination parameter estimates to predict the system variable across dyads. It does so for 4 models with different sets of predictors: 1) intercept only (provided as a sort of null model for comparison), 2) inertia-only, 3) coordination-only, and 4) the full model. The system variable can be either a dyadic variable (e.g., both partners have the same score, as in relationship length) or an individual variable (e.g., partners can have different scores, as in age). In the present example, the system variable (shared unhealthy behavior) is assessed at the dyad level. In this case, the shared system variable score is predicted from both partner's inertia and coordination estimates using regular regression models. When the system variable is at the individual level, both partner's system variable scores are predicted from actor- and partner- effects of the inertia and coordination parameter estimates using cross-sectional random-intercept dyadic models.

The function takes the name of the dataframe containing the parameter estimates ("paramData" in our example), an argument indicating whether the system variable is dyadic or individual (called "dyad" or "indiv"), names for the two levels of the distinguishing variable in the correct order (0 first, then 1; these names will be used to label output) 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 or lme objects containing the full results for each model, and 2) adjusted *R^2* information for each model (called "R2"). The function also displays histograms of the residuals and plots of the predicted values against observed values for each model.

```
sysVarOut <- inertCoordSysVarOut(paramData, "dyad","Women", "Men",
"sub")
```

An example of the plots is shown below. We can see that the residuals of the full inertia-coordination model predicting the system variable 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 4 models (intercept only, inertia only, coordination only, full model) predicting "sub" :






```
sysVarOut$R2

$baseR2
[1] 0

$inertR2
[1] 4.714654e-05

$coordR2
[1] 0.03450754

$inertCoordR2
[1] 0.01214933
```

In this case (a dyadic system variable) the intercept-only baseline model has an R^2 of zero, because there is no variance in the predictor (e.g., the intercept). We see that the R^2 for inertia-only model is also essentially zero. The coordination- only model accounts for the most variance in the system variable, but it is still about 3%, followed by the full model, which accounts for about 1% of the variance in shared unhealthy behavior.

We next consider the full model results, using the summary function and focusing on the coordination-only model as an example:

```
summary(sysVarOut$models$coord)

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 *  

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
```

In keeping with the fact that the coordination-only model had the highest adjusted *R^2*, we also see it produces a significant effect, with the coordination parameter for the men being a predictor of the unhealthy behaviors. This result suggests that higher men's in-phase coordination 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 accord 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), telling us that this is a very small effect.

Finally, we can investigate the inertia and coordination parameter estimates as outcomes of the system variable, rather than predictors of it. The function "inertCoordSysVarIn" uses multivariate correlated-residuals dyadic models to compare an intercept-only baseline model to one that includes the system variable as a predictor of the dynamic parameters. If the system variable is at the individual level, then both actor and partner effects are included in the model. The function takes the name of the dataframe containing the parameter estimates ("paramData" in our example), an argument indicating whether the system variable is dyadic or individual (called "dyad" or "indiv"), 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 output). The function returns a named list including: 1) the gls objects containing the full results for each model (called "models"), and 2) adjusted *R^2* information for each model (called "R2").

```
sysVarIn <- inertCoordSysVarIn(paramData, "dyad", "Women",
"Men")
```

We first consider the R^2 output:

```
sysVarIn$R2
$baseR2
[1] 0.4956769

$sysVarInR2
[1] 0.5047864
```

We see that the intercept-only model explains about half the variance in the inertia and coordination parameter estimates, and including the system variable as a predictor does not explain any additional variance. We can also obtain the full model results using the summary function:

```
summary(sysVarIn$models$sysVarIn)
```

We interpret the output one section at a time, beginning with the correlation structure:

```
Correlation Structure: General
 Formula: ~1 | dyad
 Parameter estimate(s):
 Correlation:
  1      2      3
2  0.007
3 -0.228  0.149
4 -0.007 -0.259  0.531
```

This shows the correlation at the dyad level of the 4 inertia-coordination parameters, numbered in the order they went into the model (inert0, inert1, coord0, coord1). We see, for example, that partner's inertia is uncorrelated (r = .01), but their coordination is strongly correlated (r = .53).

The next output shows the relative residual variances for the 4 parameters. The first parameter, inert0 acts as the reference and the residual standard error at the end of the output is for that variable. The residual variance for the others are then multiplied by it. For example, here the residual variance for inert1 is 1.06 times larger than for inert0.

```
Variance function:
 Structure: Different standard deviations per stratum
 Formula: ~1 | parameter
 Parameter estimates:
   inert0    inert1    coord0    coord1
1.0000000 1.0601399 0.7723448 0.7075846

Residual standard error: 0.1937055
```

Finally, the following output gives the fixed effects estimates for the intercepts of the dynamic parameters and the regression parameters for the system variable as a predictor of the dynamics. For example, we see that the estimate of the mean for inertia for women is .53 and that the system variable has a non-significant parameter estimate of .31 for predicting women's inertia.

```
Coefficients:
                        Value Std.Error   t-value p-value
inert_Women         0.5260387 0.0433159 12.144233  0.0000
inert_Men           0.5406162 0.0459209 11.772758  0.0000
coord_Women         0.1624812 0.0334548  4.856733  0.0000
coord_Men           0.1048954 0.0306497  3.422399  0.0007
inert_SysVar_Women  0.3089701 1.2212865  0.252987  0.8005
inert_SysVar_Men   -1.8136872 1.2947345 -1.400818  0.1623
coord_SysVar_Women  0.2782415 0.9432543  0.294980  0.7682
coord_SysVar_Men    1.7031884 0.8641635  1.970910  0.0497
```

In summary, from this analysis we conclude that the inertia-coordination model fits the observed trajectories of the state variables quite well. There is also some evidence for a very small effect of men's coordination on shared health behaviors. In contrast there is no evidence that those behaviors predict inertia or coordination in turn.

Controlling for Covariates

In the present example, we do not have any evidence that the system dynamics are associated meaningfully with the system variable. If we did have such evidence, however, we may want to ask whether the variance in the system variable associated with the dynamics is unique, or could be explained by other cross-sectional variables such as relationship quality, personality, etc. To allow for this, the central functions allow for an optional argument that provides the name of a covariate to be included in the analysis. The following syntax repeats the core of the analysis, this time including a specified covariate (in this example, "stress", which is the sum of relational and general stress during the last week, which was correlated with shared unhealthy behaviors, $r = .17$).

```
coVar <- "stress"

data2 <- dataPrep(data1, "person","couple",
 "dial","sub","male","time", time_lag=5, coVar= TRUE)

inertCoord <- indivInertCoord(data2, "Women", "Men", "Dial", coVar=
 TRUE)

paramData <- inertCoord$paramData
```

```
sysVarOut <- inertCoordSysVarOut(paramData, "dyad", "Women", "Men",
 "Sub", coVar= TRUE)
```

The following results show that including the covariate reduces the adjusted R^2 for the full model and increases the p-value to non-significance for men's coordination, although the covariate is not a significant predictor of the unhealthy behaviors either.

```
sysVarOut$R2

$baseR2
[1] 0

$inertR2
[1] 4.714654e-05

$coordR2
[1] 0.03450754

$inertCoordR2
[1] 0.01214933

$inertCoordCovarR2
[1] 0.0253705

summary(sysVarOut$models$inertCoordCovar)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
intercept    0.0239891  0.0100221   2.394   0.0195 *
covariate    0.0043423  0.0028546   1.521   0.1329
inert_Women  0.0005256  0.0115976   0.045   0.9640
inert_Men   -0.0068460  0.0116283  -0.589   0.5580
coord_Women -0.0141690  0.0192148  -0.737   0.4635
coord_Men    0.0381193  0.0208722   1.826   0.0723 .
```