

---

# The Process Explorer

An application for studying issues of causal effects in discrete-time and  
continuous-time stochastic processes

## Master thesis

Master of Science, Psychology

Institute of Psychology

Faculty of Social and Behavioural Sciences

Friedrich-Schiller-University Jena

submitted by

**Julia Gantner**

First Advisor: Prof. Dr. Rolf Steyer

Second Advisor: M. Sc. Andreas Neudecker

Julia Gantner

Matriculation Number: 145115

Wiesenstr. 12a

07749 Jena

July 9, 2018

---

## Contents

<b>1. A misused paradigm</b>	<b>1</b>
<b>2. Mediation models</b>	<b>2</b>
2.1. The classical paradigm . . . . .	2
2.2. Causality in mediation models . . . . .	4
<b>3. Discrete-time stochastic process models</b>	<b>7</b>
3.1. Analysing discrete-time models . . . . .	8
3.2. Causality in discrete-time models . . . . .	11
<b>4. Continuous-time stochastic process models</b>	<b>13</b>
4.1. Analysing continuous-time models . . . . .	18
4.2. Modelling a treatment . . . . .	19
4.3. Causality in continuous-time models . . . . .	20
<b>5. The app</b>	<b>22</b>
5.1. Computation . . . . .	22
5.2. User interface . . . . .	23
<b>6. Discussion</b>	<b>30</b>
<b>7. Appendix</b>	<b>32</b>

## Abstract

It is a classical experimental paradigm to manipulate manifest treatment conditions ( $X$ ) assumed to have an effect on a latent theoretical variable ( $\eta_1$ ) which in turn affects a manifest or latent outcome variable ( $\eta_2$ ). With this paradigm researchers intend to test the theory that  $\eta_1$  has a causal effect on  $\eta_2$ . Therefore they estimate the direct effect of  $X$  on  $\eta_1$  (manipulation check) and make the erroneous conclusion that a successful manipulation check means a significant effect of  $X$  on  $\eta_2$  can be interpreted as evidence for their originally stated hypothesis that  $\eta_1$  affects  $\eta_2$ . Even if the theoretical constructs considered in a case like this can be represented in a model with only the three random variables  $X$ ,  $\eta_1$  and  $\eta_2$ , a causal interpretation of the direct and indirect effects is only possible under very special circumstances, as Mayer, Thoemmes, Rose, Steyer, and West (2014) have shown.

Looking at substantive psychological theories in more detail, however, it seems that the theoretical constructs can often not be defined as a single random variable (such as the latent variable  $\eta$ ). Rather they have to be conceptualised as stochastic processes with latent variables  $\eta_{it}, t \in T$ . These stochastic processes might not even be time-discrete, as assumed in cross-lagged panel models and multivariate autoregressive processes. Instead, for many psychological constructs they have to be conceptualised as continuous-time processes. This rises new questions about causality in discrete-time and continuous-time models, which can be explored by the application presented in this thesis.

## Zusammenfassung

Ein typisches Experimentalparadigma der sozialwissenschaftlichen Forschung besteht darin einen vermuteten Effekt eines latenten Konstruktes  $\eta_1$  auf ein anderes latentes Konstrukt  $\eta_2$  zu untersuchen, indem  $\eta_1$  mit einem manifesten Treatment ( $X$ ) manipuliert und der Unterschied zwischen den Bedingungen in  $\eta_2$  gemessen wird. Häufig wird dann der direkte Effekt von  $X$  auf  $\eta_2$  in einem sog. Manipulation Check untersucht. Fällt dieser positiv aus, wird daraus der fehlerhafte Schluss gezogen ein signifikanter Effekt von  $X$  auf  $\eta_2$  sei ein Beleg für die Theorie, dass  $\eta_1$  einen Effekt auf  $\eta_2$  hat.

Selbst in einem Fall in dem es sinnvoll ist, derartige theoretische Konstrukte mit nur drei Zufallsvariablen  $X$ ,  $\eta_1$  und  $\eta_2$  zu repräsentieren, ist eine kausale Interpretation der direkten und indirekten Effekte nur unter sehr spezifischen Voraussetzungen möglich, wie Mayer et al. (2014) gezeigt haben. Ein genauerer Blick auf übliche psychologische Theorien zeigt allerdings, dass die betrachteten theoretischen Konstrukte oft nicht als einzelne Zufallsvariablen (wie die latente Variable  $\eta$ ) definiert werden können, sondern als stochastische Prozesse mit den latenten Variablen  $\eta_{it}, t \in T$  aufzufassen sind. Diese stochastischen Prozesse müssen dabei nicht einmal zeitdiskret sein. Stattdessen müssen sie für viele psychologische Konstrukte durch zeitkontinuierliche Prozesse charakterisiert werden. Diese Überlegungen werfen neue Fragen bezüglich der Kausalität in zeitdiskreten und zeitkontinuierlichen Modellen auf, die mit der hier vorgestellten Applikation exploriert werden können.

## 1. A misused paradigm

One of the most frequently asked questions in social sciences and psychology in particular is the question of how a certain psychological construct (whatever this means) affects (whatever that means) another construct. Almost any research question in psychology can be rephrased as “How does variable  $X$  affect variable  $Y$ ?”. For example, there are a number of researchers investigating how feelings of loneliness affect depressive mood (e.g. Alpass & Neville, 2003). In this example, as in many other psychological examples, the constructs in question are conceptualised as manifest or latent random variables, in our example the latent variables  $\eta_1$  and  $\eta_2$  that refer to loneliness and depression respectively.

If researchers want to study their question with an experiment they usually come up with a way to manipulate loneliness  $\eta_1$  in an experimental but not in a control condition and then evaluate the difference in depression  $\eta_2$  between the two conditions. Since loneliness cannot be manipulated directly, a manifest treatment ( $X$ ) has to be thought of for the manipulation. Cacioppo et al. (2006) for example treated their subjects with hypnosis to manipulate their loneliness and subsequently measured depression. The usual procedure in these paradigms is to administer the hypnosis in the experimental condition ( $X = 1$ ) but not in the control condition ( $X = 0$ ), to do a manipulation check to confirm  $X$  had the desired effect on  $\eta_1$  and then interpret the difference in depression between conditions ( $E(\eta_2|X = 1) - E(\eta_2|X = 0)$ ) as the effect of loneliness on depression.

However, there are some issues with this method. Although the manipulation check has enjoyed unbroken popularity for more than 20 years (see Fayant, Sigall, Lemonnier, Retsin, & Alexopoulos, 2017) it is often presumed to be more powerful than it actually is. Sigall and Mills (1998) argue that a manipulation check has no use in most applications because it can neither prove nor rule out alternative explanations of the observed effect and thus doesn't “improve construct validity”, no matter which result the manipulation check yields. We would agree insofar as to say that in the above example, the approach relies on the false conclusion that a successful manipulation check means that the difference in depression must be due to difference in loneliness.

To see why this conclusion is not correct, consider the path diagram in Figure 1, where

you can see that we are actually dealing with a mediation paradigm: The manipulation check corresponds to path  $c$  in the figure, the difference in depression  $E(\eta_2|X = 1) - E(\eta_2|X = 0)$  corresponds to the total treatment effect that is path  $e$ . In the following section we examine how these effects are related and how this model can be analysed in terms of a classical mediation model as proposed by Baron and Kenny (1986).

Before we start though, please consider one note regarding all of the models to follow in this thesis: As we are concerned with causality and the question how a misspecification of the structural model for analysis can lead to biased effects, we are always just considering true effects and put no attention to estimation and measurement models. Therefore all of the following path diagrams omit any measurement model or manifest indicators for the latent variables. In fact, all the points we are about to make would hold for manifest and latent variables equally.

## 2. Mediation models

### 2.1. The classical paradigm

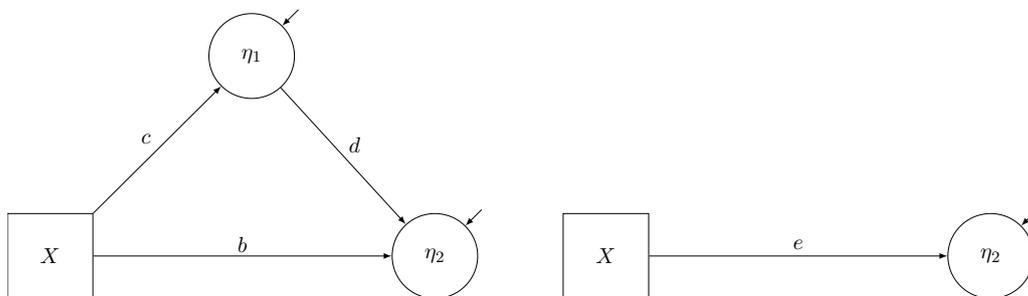


Figure 1: Direct and indirect effects of  $X$  on the left hand side, total treatment effect on  $\eta_2$  the right hand side.

The classical mediation model as proposed by Baron and Kenny (1986) more than thirty years ago is still widely used and has so far been cited more than 70,000 times. It gives a framework to analyse how the effect of one variable on another is mediated through a third variable, the so called mediator. In our example, the total effect  $e$  of the hypnosis ( $X$ ) on depression ( $\eta_2$ ) can stem from two different 'routes': there can be a direct effect ( $b$ ) of

hypnosis on depression and there can be an indirect effect ( $c \cdot d$ ) through loneliness. If we assume linear relationships between the variables, the according conditional expectations can be parametrised as:

$$E(\eta_2|X) = a_0 + e \cdot X \quad (1)$$

$$E(\eta_2|\eta_1, X) = a_1 + d \cdot \eta_1 + b \cdot X \quad (2)$$

$$E(\eta_1|X) = a_2 + c \cdot X. \quad (3)$$

We can show how these effects relate:

$$E(\eta_2|X) = E\left(E(\eta_2|X, \eta_1) \middle| X\right) \quad (4)$$

$$= E(a_1 + d \cdot \eta_1 + b \cdot X|X) \quad (5)$$

$$= a_1 + d \cdot E(\eta_1|X) + b \cdot X \quad (6)$$

$$= a_1 + d \cdot (a_2 + c \cdot X) + b \cdot X \quad (7)$$

$$= (a_1 + d \cdot a_2) + (b + c \cdot d)X \quad (8)$$

$$= a_0 + e \cdot X \quad (9)$$

Which leaves us with the equation (also described by Baron & Kenny, 1986; Wright, 1934)

$$e = b + c \cdot d \quad (10)$$

to describe the relation of direct and indirect effects in the mediation paradigm. To get an understanding of what this equation implies it is easiest to consider two extreme cases: In the first case there is no direct effect of  $X$  on  $\eta_2$ , that is,  $b = 0$ . In this case, we get  $e = c \cdot d$ . In the second case there is no indirect effect of  $X$  on  $\eta_2$ , that is  $c \cdot d = 0$ . In this case we get  $e = b$ .

For our example these cases represent a situation where the entire effect of hypnosis on depression is mediated through loneliness (case 1) and a situation where none of this effect is mediated through loneliness (case 2). To reiterate, this means that the identical total

effect  $e$  can occur in situation were loneliness is the only thing that leads to an effect of  $X$  on  $\eta_2$  and in a situation were loneliness plays no role at all.

It is important to understand that in the example presented in Section 1 only the total effect is evaluated and therefore there is no way for the researcher to know where on the continuum between those extreme cases their model falls. Even the manipulation check does not give any helpful information here since it only addresses the case in which  $c \neq 0$ , which still does not preclude that  $d = 0$ , the effect of loneliness on depression the researcher is actually interested in. The only way to examine an effect of loneliness on depression would be to not only measure  $\eta_1$  for a manipulation check but to include this measure of loneliness in a mediation analysis to get to the indirect effect the researcher is actually interested in.

## 2.2. Causality in mediation models

While the application of a mediation analysis to our example would at first glance solve the problem of which effect precisely we have to analyse, it does introduce some issues regarding causality that have been pointed out by Steyer, Mayer, and Fiege (2014) and Mayer et al. (2014). In our example we would assume that the assignment to treatment conditions ( $X = 1$  for hypnosis,  $X = 0$  for no hypnosis) would be randomized. This satisfies the causality condition CC1 from Steyer et al. (2014) of independence of the treatment variable  $X$  from the (possibly multidimensional) global covariate  $C_X$ :  $P(X=x|C_X) = P(X=x)$  for all values  $x$  of  $X$ , where  $C_X$  represents any covariates that are prior or simultaneous to  $X$ .

However, it is important to note that this causality condition only implies  $C_X$ -unbiasedness of  $E(\eta_2|X)$ , that is, unbiasedness of the total effect of  $X$  on  $\eta_2$ . Mayer et al. (2014) have shown that  $P(X=x|C_X) = P(X=x)$  does not imply unbiasedness of  $E(\eta_2|X, \eta_1)$ , that is, unbiasedness of the direct and indirect effects of  $X$  on  $\eta_2$ . We can show that ignoring pretests pertaining to  $\eta_1$  and  $\eta_2$  (for notation we use  $\eta_{1pre}$  and  $\eta_{2pre}$  for pretests and  $\eta_1$  and  $\eta_2$  for posttests), can lead to strongly biased estimations of direct and indirect effects, even if randomisation guarantees  $X$  is (unconditionally) independent of  $\eta_{1pre}$  and  $\eta_{2pre}$ ,

that is  $P(X=x|\eta_{1pre}, \eta_{2pre}) = P(X=x)$ . Consider the following conditional expectations:

$$\begin{aligned} E(\eta_2|X) &= \alpha_0 + \alpha_1 X \\ E(\eta_2|X, \eta_{1pre}, \eta_{2pre}) &= \beta_0 + \beta_1 X + \beta_2 \eta_{1pre} + \beta_3 \eta_{2pre} \\ E(\eta_2|X, \eta_1) &= \gamma_0 + \gamma_1 X + \gamma_2 \eta_1 \\ E(\eta_2|X, \eta_1, \eta_{1pre}, \eta_{2pre}) &= \delta_0 + \delta_1 X + \delta_2 \eta_1 + \delta_3 \eta_{1pre} + \delta_4 \eta_{2pre}, \end{aligned}$$

We can show that the total effect of  $X$  is the same, whether we include pretreatment measures or not, because randomisation of the treatment guarantees independence of  $X$  and  $\eta_{1pre}$  and  $\eta_{2pre}$ :

$$\begin{aligned} E(\eta_2|X) &= E\left(E(\eta_2|X, \eta_{1pre}, \eta_{2pre}) \middle| X\right) \\ &= E(\beta_0 + \beta_1 \cdot X + \beta_2 \cdot \eta_{1pre} + \beta_3 \cdot \eta_{2pre} | X) \\ &= \beta_0 + \beta_1 \cdot X + \beta_2 \cdot E(\eta_{1pre}|X) + \beta_3 \cdot E(\eta_{2pre}|X) \\ &= \beta_0 + \beta_1 \cdot X + \beta_2 \cdot E(\eta_{1pre}) + \beta_3 \cdot E(\eta_{2pre}) \\ &= \left(\beta_0 + \beta_2 \cdot E(\eta_{1pre}) + \beta_3 \cdot E(\eta_{2pre})\right) + \beta_1 \cdot X \\ &= \alpha_0 + \alpha_1 \cdot X, \end{aligned}$$

which means that  $\alpha_1 = \beta_1$ .

This does however not work if we consider direct and indirect effects. Since we condition on the mediating variable  $\eta_1$  here and, conditional on  $\eta_1$ ,  $X$  can be dependent on  $\eta_{1pre}$  and  $\eta_{2pre}$ , this means that  $P(X=x|\eta_1, \eta_{1pre}, \eta_{2pre}) \neq P(X=x|\eta_1)$  for all values  $x$  of  $X$ .

$$\begin{aligned} E(\eta_2|X, \eta_1) &= E\left(E(\eta_2|X, \eta_1, \eta_{1pre}, \eta_{2pre}) \middle| X, \eta_1\right) \\ &= E(\delta_0 + \delta_1 \cdot X + \delta_2 \cdot \eta_1 + \delta_3 \cdot \eta_{1pre} + \delta_4 \cdot \eta_{2pre} | X, \eta_1) \\ &= \delta_0 + \delta_1 \cdot E(X|\eta_1) + \delta_2 \cdot E(\eta_1|X) + \delta_3 \cdot E(\eta_{1pre}|X, \eta_1) + \delta_4 \cdot E(\eta_{2pre}|X, \eta_1) \\ &\neq \gamma_0 + \gamma_1 X + \gamma_2 \eta_1 \end{aligned}$$

which means that  $\gamma_1 \neq \delta_1$ . This is plausible in intuitive terms if we look at Fig. 2: Both  $\eta_{1pre}$  and  $X$  are correlated with  $\eta_1$  because loneliness depends both on hypnosis and previous loneliness. Although the (unconditional!) correlation of hypnosis and previous loneliness ( $\eta_{1pre}$ ) is zero due to randomisation, the partial correlation  $Corr(X, \eta_{1pre}; \eta_1)$  is not zero as can be shown by a thought experiment: Posttreatment loneliness is likely to be very high if pretreatment loneliness was high and the hypnosis induced more loneliness. Posttreatment loneliness is likely to be very low if pretreatment loneliness was low and hypnosis didn't induce more loneliness. This means that high values on  $X$  and  $\eta_{1pre}$  imply high values on  $\eta_1$  and low values on  $X$  and  $\eta_{1pre}$  imply low values on  $\eta_1$ .

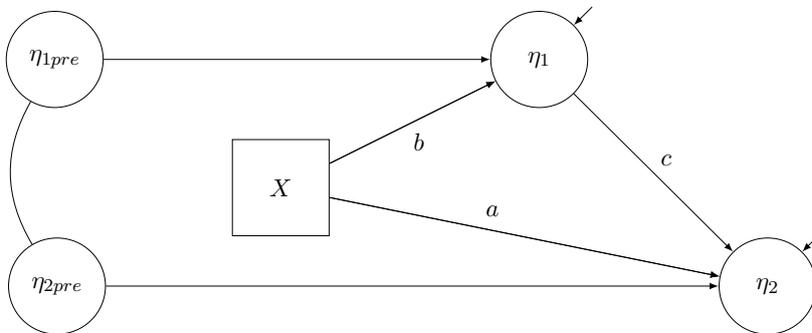


Figure 2: Mediation model with pretreatment measures

In empirical applications this can of course be true for other variables that are prior to the treatment too, but careful selection of pretreatment measures will often reduce most of the bias that could stem from unobserved covariates, as has also been argued by Steiner, Cook, Shadish, and Clark (2010). In our simple example, were there are no covariates but the pretreatment measures, we will be able to identify the causal effect of loneliness ( $\eta_1$ ) on depression ( $\eta_2$ ) as the coefficient  $\beta_2$  in

$$E(\eta_2|X, \eta_1, \eta_{1pre}, \eta_{2pre}) = \beta_0 + \beta_1 X + \beta_2 \eta_1 + \beta_3 \eta_{1pre} + \beta_4 \eta_{2pre}.$$

Once we have acknowledged the fact that neither loneliness nor depression can be represented by single variables that occur only once, it is natural to admit that representing them with only one pretreatment and one posttreatment measure still oversimplifies mat-

ters when we look at how we imagine those constructs in our theory. The next section therefore discusses how the exemplary scenario should rather be represented as a stochastic process model.

### 3. Discrete-time stochastic process models

A stochastic process is defined as a family  $(\eta_t, t \in T)$  of random variables  $\eta_t$ , where  $t \in T$  often indicates a time point. In the sequel we will also call these variables observations. We start simple, considering a discrete-time stochastic process for our example, that is, we assume there is a finite sequence of random variables  $\eta_{1t}$  and  $\eta_{2t}$  representing states of loneliness and depression over time. According to this line of thinking, depression and loneliness exist before and after our experimental intervention and are likely to have some kind of reciprocal interaction over time. Our hypnosis ( $X$ ) then occurs once at some time point  $t_X$  and can affect the following states  $\eta_{1s}$  and  $\eta_{2s}$ , where  $s > t_X$ . Note that as long as we are talking about discrete-time models, the set  $T$  is countable.

Though there are a large number of possible process models in this kind of setting, we will go for a rather simple version and confine our discussion to so called  $AR_1$ -processes. This is a short cut for *autoregressive process of order 1* and describes a process where each variable only depends on the immediately preceding variables but not on the prior variables. If we look at one of these  $AR_1$ -processes in isolation, we can write:

$$E(\eta_t | \eta_1, \dots, \eta_{t-1}) = \alpha_0 + \alpha_1 \cdot \eta_{t-1} \quad (11)$$

$$\zeta_t = \eta_t - E(\eta_t | \eta_1, \dots, \eta_{t-1}). \quad (12)$$

$\zeta$  is the error term that is sometimes referred to as innovation. In our case we have two  $AR_1$ -processes that interact with each other, resulting in the process depicted in Figure 3. The highlighted variables are the variables of Figure 1, which are now embedded in the stochastic process. In this process, depression at any point in time depends on previous

depression and previous loneliness:

$$E(\eta_{2t}|\eta_{11}, \dots, \eta_{1(t-1)}, \eta_{21}, \dots, \eta_{2(t-1)}) = \beta_0 + \beta_1 \cdot \eta_{1(t-1)} + \beta_2 \cdot \eta_{2(t-1)} \quad (13)$$

For all  $t$  posterior to the treatment we also have to include the intervention variable  $X$ , resulting in:

$$E(\eta_{2t}|\eta_{11}, \dots, \eta_{1(t-1)}, \eta_{21}, \dots, \eta_{2(t-1)}, X) = \gamma_0 + \gamma_1 \cdot \eta_{1(t-1)} + \gamma_2 \cdot \eta_{2(t-1)} + \gamma_3 \cdot X \quad (14)$$

Of course processes can get much more complex than this. The variables could for example have an effect on more than just the immediately following variable and the conditional expectations don't have to be parametrisable as linear combinations of the preceding variables. Nevertheless, this model allows us to examine some issues with the interpretation of the effects we are measuring in our experiment.

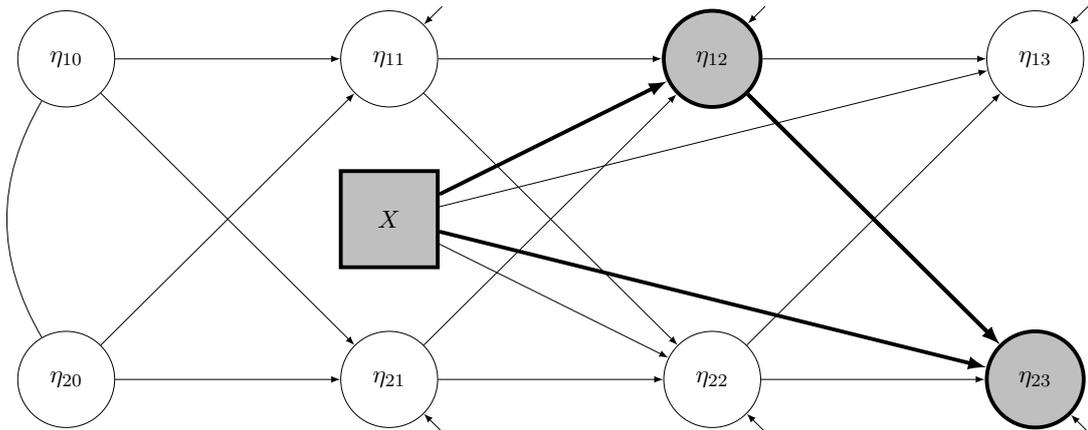


Figure 3: Simple stochastic process with four time points

### 3.1. Analysing discrete-time models

There are different ways to analyse these kinds of structural equation models (SEM). As we mentioned before, processes can be latent or manifest. If they are manifest, we can use multiple regression analysis to estimate the effects we are interested in. If we want to construct them as latent processes, we have to use manifest variables and a measurement

model that defines the latent variables and use a program for SEM-modelling such as the R package lavaan (Rosseel, 2012) or the statistical program Mplus (“MPLUS (Version 8).[Computer Software]”, 2012). Since this thesis is not concerned with the measurement model or estimation, all diagrams only show the structural equations. The focus of this thesis is on identification and the misspecification of the causal model. Hence, we simulate data and analyse it with different models to explore the bias if the model for analysis doesn’t match the generating (true) process. However, since we are interested in identification and not in estimation, we do not simulate empirical data. Instead, we will calculate true regression coefficients from true variance-covariance matrices of the variables in the structural equation model (see Jöreskog & Sörbom, 1988):

$$\boldsymbol{\eta} = \boldsymbol{\alpha} + \mathbf{B}\boldsymbol{\eta} + \boldsymbol{\Gamma}\boldsymbol{\xi} + \boldsymbol{\zeta}, \quad (15)$$

An overview over all the symbols used in the following notation can be found in Table 1. The covariance matrix  $\boldsymbol{\Sigma}(\boldsymbol{\xi}, \boldsymbol{\eta})$  can be composed as:

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Phi} & \boldsymbol{\Phi}\boldsymbol{\Gamma}'(\mathbf{I} - \mathbf{B}')^{-1} \\ (\boldsymbol{\Phi}\boldsymbol{\Gamma}'(\mathbf{I} - \mathbf{B})^{-1})' & (\mathbf{I} - \mathbf{B})^{-1}(\boldsymbol{\Gamma}\boldsymbol{\Phi}\boldsymbol{\Gamma}' + \boldsymbol{\Psi})(\mathbf{I} - \mathbf{B}')^{-1} \end{pmatrix} \quad (16)$$

Symbol	Meaning	Dimensions
$\boldsymbol{\eta}$	Vector of the endogenous variables	$n_{\boldsymbol{\eta}} \times 1$
$\boldsymbol{\xi}$	Vector of the exogenous variables	$n_{\boldsymbol{\xi}} \times 1$
$\boldsymbol{\Phi}$	Covariance matrix of the exogenous variables	$n_{\boldsymbol{\xi}} \times n_{\boldsymbol{\xi}}$
$\boldsymbol{\Psi}$	Covariance matrix of the residuals of the endogenous variables	$n_{\boldsymbol{\eta}} \times n_{\boldsymbol{\eta}}$
$\boldsymbol{\Gamma}$	Effects of exogenous variables on endogenous variables	$n_{\boldsymbol{\eta}} \times n_{\boldsymbol{\xi}}$
$\mathbf{B}$	Effects of endogenous variables on each other	$n_{\boldsymbol{\eta}} \times n_{\boldsymbol{\eta}}$
$\mathbf{I}$	Identity matrix	$n_{\boldsymbol{\eta}} \times n_{\boldsymbol{\eta}}$

Table 1: Notation used for the true parameters in the structural equations

If we want to compute any regression coefficients within this set of variables, we first have to choose one outcome variable from the endogenous variables ( $\boldsymbol{\eta}$ ) and one or more regressors from the exogenous variables ( $\boldsymbol{\xi}$ ) and endogenous variables ( $\boldsymbol{\eta}$ ). We call the outcome variable  $Y$  and the (column-) vector containing the regressors  $\mathbf{X}$ . For the highlighted model in Figure 3 this is for example:

$$Y = \eta_{23}$$

$$\mathbf{X} = \begin{pmatrix} X \\ \eta_{12} \end{pmatrix}$$

From the  $\boldsymbol{\Sigma}$ -matrix we can extract the two matrices we need to compute the regression coefficients, namely  $\boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}}$ , the covariance matrix of  $\mathbf{X}$ , and  $\boldsymbol{\Sigma}_{\mathbf{X}Y}$ , the covariance matrix of  $\mathbf{X}$  and  $Y$ . The formula to compute the true regression coefficients in this regression  $E(Y|\mathbf{X}) = \boldsymbol{\beta}_0 + \boldsymbol{\beta}\mathbf{X}$  is then:

$$\boldsymbol{\beta}_0 = E(Y) - \boldsymbol{\beta}' \begin{pmatrix} E(x_1) \\ \vdots \\ E(x_n) \end{pmatrix} = E(\eta_{23}) - \boldsymbol{\beta}' \begin{pmatrix} E(X) \\ E(\eta_{12}) \end{pmatrix} \quad (17)$$

$$\boldsymbol{\beta} = \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}}^{-1} \boldsymbol{\Sigma}_{\mathbf{X}Y} \quad (18)$$

This procedure allows us to come up with any discrete-time model by specifying the true effects and covariance structure and to explore how the choice of different response variables and different regressors determines whether the calculated regression coefficients match the true effects or not, regardless of any issues concerning estimation.

### 3.2. Causality in discrete-time models

If we assume to have a discrete-time process as depicted in Figure 3 it makes sense to look at what happens if we don't measure every occurrence of the  $\eta_{it}$ -processes for our analysis. Even though it is not at all realistic for our specific example of depression and loneliness, let us for the sake of the argument assume that the time interval between  $t$  and  $t + 1$  in Figure 3 is exactly 1 minute and let us consider a process without a treatment variable. To represent this (true) process perfectly, we would need to measure it exactly every minute, thereby capturing every single state of the  $\eta_{it}$ . If we then consider  $E(\eta_{23}|\eta_{10}, \eta_{20}, \eta_{11}, \eta_{21}, \eta_{12}, \eta_{22}, X)$ , we would be sure all our effects are unbiased, because we included all variables determining the conditional expectation of  $\eta_{23}$  in the analysis. What this means in terms of causality is that we control for all possible confounders that exist, that is, for the global covariate  $C_X$ .

In many cases this is not necessary for all possible confounders. However, if the outcome variable depends on unobserved covariates, the effects we observe may be biased. To overcome this problem, we can for example try to find a (possibly multidimensional) variable  $Z_t$  that implies  $(C_X, Z_t)$ -unbiasedness of  $E(\eta_{23}|X, Z_t)$ , by satisfying causality condition  $CC2Z$  from Steyer et al. (2014), which is  $E(Y|X, C_X) = E(Y|X, Z_t)$ .

The more crucial problem is though, that to measure the process exactly we would have to be aware that the 'natural pulse' of this particular process is one minute. If for some reason we were assuming it to be 2 or 3 minutes and timing our measurement accordingly instead, we would end up observing total effects while believing that we are observing direct effects.

This scenario is visualised in Figure 4: On the top we can see the direct effects in the model, on the bottom we can see the total effects getting more and more biased with every additional time point we skip. This can even get to the point were some of the effects

change sign. It is important to emphasize the relevance of this observation: If no one knows the natural pulse of the system and two different researchers do an experiment on the exact same process but with different time intervals between their measurements, they can come to completely opposing conclusions about the effects, for example regarding the question whether current loneliness depends positively or negatively on previous loneliness.

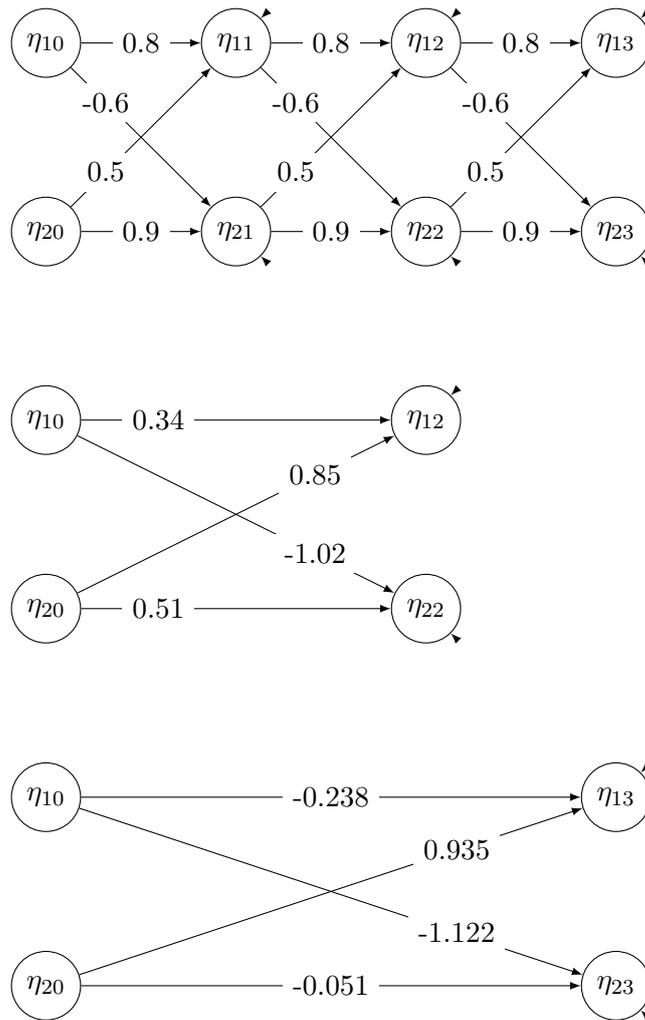


Figure 4: Direct effects on top, total effects with lag 2 and and lag 3 below.

By now it has become clear that our example of loneliness and depression is not at all a good example for a discrete-time stochastic process. If we are looking at adequate psychological examples for discrete-time processes there don't seem to be a lot of realistic scenarios.

One we came up with is a situation where two subjects are communicating via computers that send and receive messages with a certain pulse, where action and reaction of the two subjects will occur exactly at the time when the message arrives and there is nothing to react to in between two messages. For the vast majority of psychological applications however, it is really hard to figure out the natural pulse at which the processes occur and that is because they are not discrete-time processes. Instead, they have to be conceptualised as continuous-time processes.

#### 4. Continuous-time stochastic process models

To figure out why depression and loneliness (and most other psychological constructs for that matter) have to be conceptualised as continuous-time processes, it is useful to review the assumptions you would make when you claim they are discrete-time processes: Defining depression as a discrete-time process that assumes there is depression at time point  $t$ , there is depression at  $t + 1$  and *there is no depression in between those time points*. Generalising this principle to other affective and cognitive processes would mean to assume that there are points in time where the individual is in some kind of a cognitive-affective vacuum, with feelings of loneliness and depression turning on and off at the pulse of the discrete-time process.

That is of course not how we imagine things to happen. When we think about loneliness and depression, we rather assume that we could measure it at any given point in time and that even though it might not be possible from a technical perspective, we should in theory be able to observe our constructs between any two time points, no matter how close together these points are. This means that we think of two processes  $\eta_1(t)$  and  $\eta_2(t)$  depending on  $t$ , where  $t$  is the value of a continuous variable  $T$ , representing the continuous time line of reality.

Trying to extend our path diagram from the previous section for continuous-time leads us to Figure 5. The fact that there still are discrete instances of  $\eta_i(t)$  is due to drawing possibilities when actually there is an infinite number of  $\eta_i(t)$  between any of those depicted instances. What this diagram shows however, is that depression and loneliness go on as

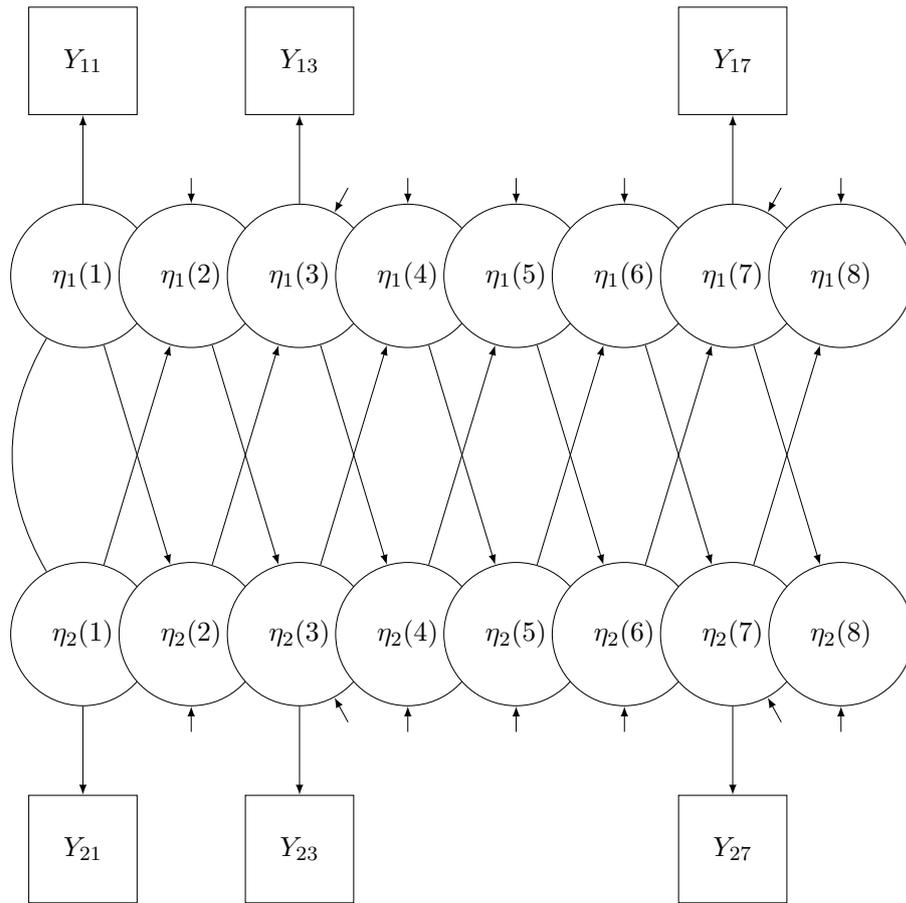


Figure 5: A continuous-time process

a continuous stream and our observations  $Y_{it}$  only take a snapshot of the situation at the time when the measurement takes place.

A short version of this to depict only the structural model without any measurement model is shown in Figure 6. Here,  $a$  and  $b$  represent the continuous-time auto effects and  $c$  and  $d$  represent the continuous-time cross effects.

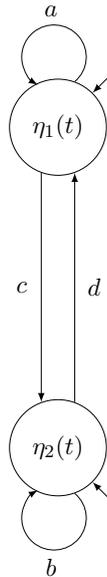


Figure 6: A continuous-time process without manifest indicators

This kind of continuous-time process is often described in form of differential equation models (see e.g. Voelkle, Oud, Davidov, & Schmidt, 2012; Deboeck & Preacher, 2015). Differential equation models are models that contain at least one derivative that describes the change in one variable (e.g.,  $\eta_1$ ) with respect to another variable (e.g.,  $T$ ) in the form of  $\frac{d\eta_1(t)}{dt}$  which would describe the change in loneliness with respect to time. Models that contain such expressions represent moment-to-moment changes in a system of variables. Through integration these moment-to-moment changes can be accumulated to compute the change in the system for a given time interval  $\Delta t_i$ . The equation used to describe a differential equation model is:

$$d\boldsymbol{\eta}(t) = (\mathbf{A}\boldsymbol{\eta}(t) + \mathbf{b})dt + Gd\mathbf{W}(t), \quad (19)$$

where  $\boldsymbol{\eta}(t)$  is a column-vector of the processes (e.g.,  $\boldsymbol{\eta}(t) = \begin{pmatrix} \eta_1(t) \\ \eta_2(t) \end{pmatrix}$ ),  $\mathbf{A}$  is the so called drift matrix that describes the reciprocal dependencies of those processes,  $\mathbf{b}$  is an intercept vector that provides a constant fixed input to the processes,  $\mathbf{W}(t)$  is the Wiener process (also called Brownian motion) that is used to model the error term and  $G$  is a scaling

factor for scaling this error term. This equation describes the moment-to-moment changes in our two processes contained in  $\boldsymbol{\eta}(t)$ . For our purposes the drift matrix is the most interesting part of Equation 19 as it contains the information of how the processes depend on themselves (auto-effects) and each other (cross-effects). For our example with the two processes loneliness ( $\eta_1(t)$ ) and depression ( $\eta_2(t)$ ) this matrix takes the form of a two-by-two matrix such as

$$\mathbf{A} = \begin{pmatrix} -0.5 & 0.3 \\ 0.2 & -0.4 \end{pmatrix}$$

where  $-0.5$  and  $-0.4$  are the auto-effects that describe how a process depends itself and  $0.3$  and  $0.2$  are the cross-effects that describe how the processes depend on each other.

However, when we observe this process in real life we can only observe changes that accumulated over some time interval  $\Delta t_i = t_i - t_{i-1}$ :

$$\boldsymbol{\eta}(t_i) = \mathbf{A}(\Delta t_i)\boldsymbol{\eta}(t_i - \Delta t_i) + \mathbf{w}(\Delta t_i). \tag{20}$$

In this notation, we use the index  $i$  for both  $t$  and  $\Delta t$ , because it can denote the rank of the (hypothetical) measurement occasion and the rank of the time interval between two measurement occasions at the same time:

If we consider  $i = 0, \dots, N$  measurement occasions, there are exactly  $i = 1, \dots, N$  intervals in between. Thus,  $t_i$  denotes the  $(i+1)$ th measurement occasion and  $\Delta t_i$  denotes the  $i$ th interval between measurements.  $t_3$  for example denotes the fourth measurement occasion (because we start counting at 0), and the  $\Delta t_3$  denotes the third interval, that is, the interval between measurement occasions  $t_2$  and  $t_3$ .

The matrix  $\mathbf{A}(\Delta t_i)$  describes the discrete autoregressive and cross-lag effects between observations of the processes at time  $t_i$  ( $\boldsymbol{\eta}(t_i)$ ) and observations of the processes  $\Delta t_i$  earlier ( $\boldsymbol{\eta}(t_i - \Delta t_i)$ ). Stochastic integration reveals that the formula that relates the drift parameters and their discrete effects observed for a given time interval  $\Delta t_i$  is:

$$\mathbf{A}(\Delta t_i) = e^{\mathbf{A} \cdot \Delta t_i}. \quad (21)$$

According to Deboeck and Preacher (2015) and Driver and Voelkle (2017) we can learn more about the interpretation of the drift parameters if we calculate the corresponding discrete-time effects for different hypothetical drift parameter constellations. The authors show that if  $\Delta t_i$  is one, as we let the continuous-time auto effects approach zero, the discrete-time autoregressive effects approach one and as the autoregressive effects approach zero, the auto effects approach negative infinity. This means that a negative continuous-time auto effect represents a non-explosive process that – in the absence of any other influences on the process – tends to approach to the baseline, because the negative sign means that the more positive the latent state, the stronger is its negative influence on the expected change of the process.

For the cross effects the situation is a bit more complicated as the discrete-time cross-lag effects depend both on the auto effects and on the cross effects. However, one can say that a positive cross effect for both processes means that as one of the processes gets more positive, the expected change in the other process also becomes more positive.

To get a better feeling for how the drift parameters translate into discrete effects, we will use our example matrix to compute the discrete effects for the time intervals  $\Delta t_1 = 1$ ,  $\Delta t_2 = 3$  and  $\Delta t_3 = 10$ :

$$\mathbf{A}(1) = e^{\mathbf{A}\cdot 1} = \begin{pmatrix} 0.625 & 0.193 \\ 0.129 & 0.690 \end{pmatrix}$$

$$\mathbf{A}(3) = e^{\mathbf{A}\cdot 3} = \begin{pmatrix} 0.293 & 0.256 \\ 0.171 & 0.378 \end{pmatrix}$$

$$\mathbf{A}(10) = e^{\mathbf{A}\cdot 10} = \begin{pmatrix} 0.055 & 0.081 \\ 0.054 & 0.082 \end{pmatrix}$$

The first thing we can see is that indeed a negative continuous-time auto effect translates to positive discrete-time autoregressive effects and that positive continuous-time cross effects also translate to positive discrete-time cross-lag effects. Furthermore, what we can see is that the discrete effects in this example get smaller, the greater the interval  $\Delta t_i$  gets. The discrete effects will take different values for any given time interval  $\Delta t_i$ , which already points to what they have to be thought of: Since our processes are oscillating on a continuous time scale, any effect we can actually observe will always be a total effect. This is due to the fact that we cannot have two observations where there is nothing in between, that is, we cannot have two discrete random variables representing the process at two consecutive time points where there isn't another random variable representing the state of the process in between those two. Instead we can only observe the total effects of those processes in our time interval, which are essentially accumulating every interaction that happened in between.

#### 4.1. Analysing continuous-time models

Since we can only make discrete observations of the continuous process we need a way to estimate the continuous parameters from those discrete observations. This is possible with the R package `ctsem` by Driver, Oud, and Voelkle (2017). Their method of analysis takes

a data frame with discrete observations of the process and information about the time interval between those observations and computes, amongst others, the drift matrix. To help understand this kind of models better the package also has a data generating function which takes a model specification and a number of observations and simulates a data frame with discrete observations of the process accordingly.

However, if we want to take a similar approach as we did for discrete-time models and look at how misspecification of the model leads to bias on the level of identification, we have to find a workaround for the fact that it isn't quite as easy to write down the identification formula for the drift-matrix as it is for regression coefficients (see Equation 18). The solution is to simulate empirical data that doesn't contain any measurement errors but adheres perfectly to the expected variances and covariances of the variables in the true model.

To achieve this, we specify a model with the `ctsem` package, extract the expected variances, covariances and means and use this information to generate a multivariate dataset that perfectly follows this expected structure for the true model. Having found a way to eliminate estimation issues we can now see how the drift-matrix gets estimated perfectly if we analyse the data with the correct model, as opposed to the biased drift-parameters if we misspecify the model for analysis, for example, by leaving out one of the true processes for the analysis.

## 4.2. Modelling a treatment

To introduce the idea of continuous-time processes we have so far left out the treatment variable from the introductory example. It is of course still possible to include a treatment in these kinds of processes. Driver and Voelkle (2017) describe several ways a treatment can be conceptualised in a continuous-time process. The general equation to describe these processes extends Equation 19 with a vector of time-dependent predictors  $\chi(t)$ :

$$d\boldsymbol{\eta}(t) = (\mathbf{A}\boldsymbol{\eta}(t) + \mathbf{b} + \mathbf{M}\boldsymbol{\chi}(t))dt + Gd\mathbf{W}(t), \quad (22)$$

where  $\mathbf{M}$  contains the effects the time-dependent predictors have on  $\boldsymbol{\eta}(t)$ . The authors describe a number of shapes that these time-dependent predictor effects can take, but since it is the simplest, the one we will consider here is the 'basic impulse effect'. When we think about how we imagine what happens in our experiment we would assume that loneliness and depression are continuous processes interacting over time and that the hypnosis comes in as an impulse to the system at some point, directly affecting the two processes only at the time it occurs but its effects being carried on through the processes for a while after the treatment has already stopped.

This kind of impulse can be modelled as a basic impulse in ctsem. The predictor  $X(t)$  is technically still modelled as a process, but as having an expectation  $E(X(t_i)) = 0$  for most of the time points  $i$  and only changing to a value  $E(X(t_X)) \neq 0$  at the point in time  $t_X$  when the treatment occurs. If we specify the predictor accordingly, we get an estimation of the effects of the treatment on the two processes  $\eta_i(t)$ . Note that it is advisable to have at least one observation before and one observation after the treatment, that is, a total of at least three observation points for a proper calculation of the treatment effects on the processes.

### 4.3. Causality in continuous-time models

While we can figure out the causality conditions in any discrete-time process by examining which conditions will imply unbiasedness of certain conditional expectations like  $E(\eta_{2t}|\eta_{1t}, X)$ , unbiasedness is not defined in the context of continuous-time processes. In Section 3 we were interested in causal interpretability of direct and indirect effects of  $X$ . In continuous-time processes the whole concept of a direct effect doesn't seem to make any sense since there will never be two observations of  $\eta_i(t)$  that do not have another unobserved state of  $\eta_i(t)$  in between. All conditional expectations of our discrete observations of  $\eta_i(t)$  will therefore just contain total effects whose size will depend on the time interval in between observations.

How can we approach causal interpretability of this theoretically infinite number of effects? So far we haven't been able to fully understand all aspects of a continuous-time model as necessary to confidently state a way in which causality has to be approached

in continuous-time models. The first step towards this goal is to get better acquainted with these models and one way to achieve that is the procedure that has been mentioned in the previous sections: By repeatedly specifying generating processes and models for analysis that do or don't match we can get a feeling for which misspecification results in which bias and by analysing the data with discrete-time regression models we can explore how continuous-time parameters translate to discrete-time parameters for different time intervals between observations.

## 5. The app

The application “Process Explorer” (PE) is a program to explore biases in discrete-time and continuous-time processes that may or may not occur depending on the true (“data-generating”) process and the model that is used for analysis. The app has been built in the programming language R (R Core Team, 2018) with version 1.1.0 of the package shiny (Chang, Cheng, Allaire, Xie, & McPherson, 2018). A list of all required packages can be found in appendix A, the code can be found in Section B . The app can be installed on any computer with an internet connection regardless of whether R is installed on it or not. To do that, the PE-installer can be downloaded from the authors dropbox. After the download the installer has to be executed once to install the app and create a desktop icon.

### 5.1. Computation

The PE allows the user to specify a true, data-generating process and a model for analysis. These can be specified *independently of each other*. It is even possible to have different temporal structures for the two, for example specifying a continuous-time generating process but analysing it with a discrete-time analysis model. The computation of the effects for both versions of the analysis model will be explained shortly in the following.

**Discrete-time analysis model** If this model is chosen, the results are the linear regressions coefficients one obtains if a linear regression is applied to the generating process, regardless of whether this process is a discrete-time process or a continuous-time process. If the generating process is discrete-time, the procedure described in Section 3 is used to compute the true regression coefficients from the covariance matrix resulting from the generating process. If the generating process is continuous, the regression coefficients are computed from the expected covariance matrices for the specified number of discrete observations points corresponding to the continuous-time process that has been specified.

**Continuous-time analysis model** If a continuous-time analysis model is chosen, the package ctsem is used to compute the drift matrix and, if applicable, the treatment effects in the form of time-dependent predictor effects. Since the package requires a dataframe of

discrete observations of the processes rather than their (true) covariance matrices, the app internally creates such a dataframe that reflects the specified generating process without any measurement error. For a discrete-time generating process this is possible by simulating a dataframe that perfectly adheres to the true covariance matrix of the variables in the process. Similarly, for a continuous-time generating process a dataframe is simulated that adheres perfectly to the expected covariance matrices of this process. Then a ctsem analysis is applied to these error-free dataframes.

Note that due to the form of the treatment in the continuous-time process (see Section 4.2) it is very hard to translate a continuous-time process into a form that is analysable with linear regression. Therefore the analysis model for continuous-time processes with a treatment variable is restricted to continuous-time processes. Furthermore, it is necessary to specify at least three observation points when computing a treatment effect in a continuous-time process.

For the performance of the continuous-time analysis there are some caveats. The first one concerns computation time, which in general is longer than the time needed for regression analysis anyway and will increase drastically with the number of variables. The computation time also heavily depends on the fit of the model. The worse the analysis model fits the generating process, the longer the computation will last. Especially with a discrete-time process it is vital to be very aware that some discrete-time effects we may specify just do not correspond to any realistic continuous-time process and will therefore not yield interpretable drift parameters. If the estimating algorithm doesn't converge because of this, the user gets an accordant notification.

## 5.2. User interface

In this section the structure of the user interface will be explained. Whenever the name of an element that can be found in the corresponding screen shot of the app is mentioned for the first time, it is written in italics. The app is generally divided in two equally sized parts: the *input* on the left hand side and the *output* on the right hand side. Everything the user can manipulate is listed on the input side whereas everything the user gets as a result of his input will appear on the output side. Both sides will now be explained.

## Input

The input is divided in two parts, represented by two tabs: specification of the *generating process* and specification of the *model for analysis*.

**Generating process** The landing page for the generating process allows the user to specify the parameters that define the basic structure of the data-generating process. Those parameters are the *temporal structure* of the process (discrete-time vs. continuous-time), the *number of processes*, the specification of an (only once occurring) *treatment variable*, the *number of time points* for a discrete-time process or *number of observations points* for a continuous-time process, respectively, as well as *initial variances* and *initial correlations* of the processes.

### Input

Generating process
Model for analysis

Process parameters
Effects
Let me download data

Temporal structure in the generating process

Discrete-time processes
▼

Number of processes

2

Treatment variable

No Treatment|
▲

No Treatment  
Treatment at t=1  
Treatment at t=2

Number of time points

2

3

8

Initial variances

**eta11:**

0

100

200

**eta21:**

0

100

200

Initial correlation

**eta11 and eta21**

-1

0.5

1

Figure 7: Process parameters

Depending on the parameters chosen, the *auto-effects*, *cross-effects* and *treatment effects* can be specified under the tab *Effects*. The set-up of this input already hints at an assumption the app makes: For reasons of simplicity it is assumed that in a discrete-time process the effects of one  $\eta_{it}$  on the next  $\eta_{j(t+1)}$  are fixed over time, that is, all autoregressive and cross-lag effects are the same for all of the time points.

## Input

The screenshot shows the 'Effects' tab in the app interface. It is divided into three main sections: 'Auto-effects', 'Cross-effects', and 'Treatment effects'. Each section contains sliders for different parameters, with numerical values displayed above the sliders.

Section	Parameter	Value
Auto-effects	eta1t	-0.5
	eta2t:	-0.2
Cross-effects	eta1t on eta2t:	0.5
	eta2t on eta1t:	0.7
Treatment effects	Effect on eta12	0
	Effect on eta22	0
	Effect on eta13	0
	Effect on eta23	0

Figure 8: Effects

If the user wishes to download a simulated dataset of empirical data based on this generating process, he or she can do so under the tab *Let me download data*. Here one can specify the *number of observations*, and the *residual variance* that is added for every instance of  $\eta_{it}$ . Note that we assume that this residual variance is the same for all time points  $t \in T$ . The csv-file that is produced will be named *DataForRegression.csv* for a discrete-time pro-

cess and *DataForCtsem.csv* for a continuous-time process. Both files already have a format that is suitable for an immediate analysis with the corresponding package in R. That is, the discrete-time data comes in a format that is usable in linear regression or lavaan, the continuous-time data comes in a format that is usable for a ctsem analysis.

### Input

Generating process
Model for analysis

Process parameters
Effects
Let me download data

Here you can download a dataset generated from the specified parameters.

**Number of observations**

100

10,000

Specify the residual terms to add for the simulation of empirical data

**Residual variance for eta1t**

0.1

10

**Residual variance for eta2t**

0.1

10

Download

Click to download data as csv-file.

Figure 9: Downloading data

**Model for analysis** For the analysis model, the user can specify the *temporal structure* as being either a discrete-time or a continuous-time. For a discrete-time model the user can pick one *outcome variable* and as many *regressors* as wanted. For a continuous-time model, the user can chose the processes for analysis. As mentioned before in a continuous-time process with a treatment variable, the analysis model is restricted to be continuous-time.

## Input

Generating process
  Model for analysis

Temporal structure of the analysis model

Discrete-time processes ▼

Choose the outcome:

eta23 ▼

Choose the regressors:

eta11

X

eta21

eta12

eta22

Figure 10: Discrete model for analysis

## Input

Generating process
  Model for analysis

Temporal structure of the analysis model

Continuous-time processes ▼

Which processes shall be included in the analysis?

eta1t

eta2t

X

Figure 11: Continuous model for analysis

## Output

On the output side there are two tabs, *Display the generating process* and *Display analysis results*. The first one displays a path diagram of the data-generating process. Due to

drawing possibilities all variables appear in circles, however, that does not reflect whether they are manifest or latent. As the main concern of this app is identification, we make no assumptions for the measurement model and for whether the variables and processes explored are latent or manifest.

## Output

Display the generating process

Display analysis results

Update diagram

### Path diagram of the generating process

This diagram displays the process you have specified.

All manifest indicators and residual terms are omitted.

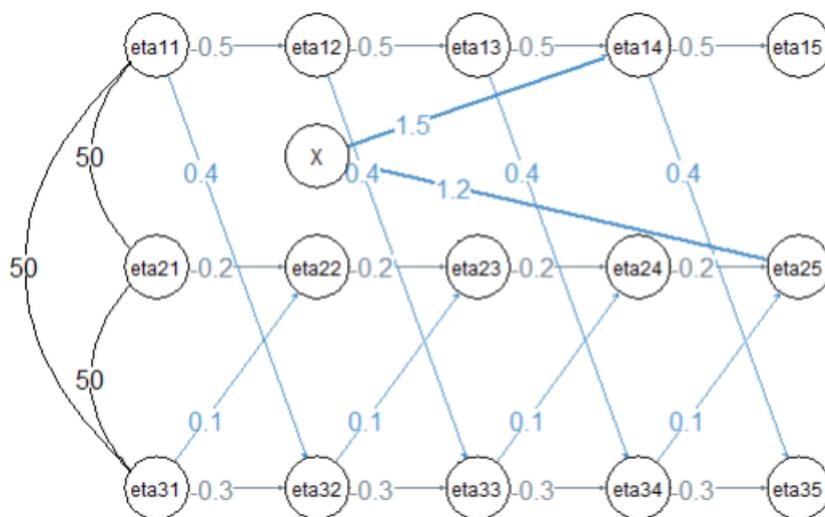


Figure 12: Path diagram of the generating process

The second tab displays the results of the analysis with the analysis model that has been specified. For a discrete-time analysis it shows a table comparing the parameters calculated for the analysis model with the parameters from the generating process. For a continuous-time analysis it shows the computed drift matrix and, if applicable, the treatment effects. This tab also shows the path diagrams of the generating process and the analysis model next to each other to allow for a quicker overview over potential biases.

## Output

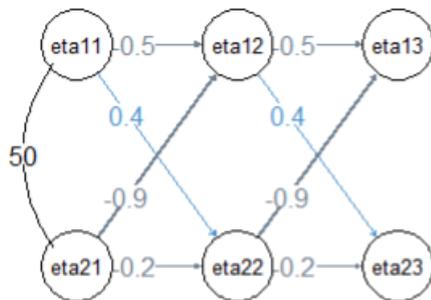


## Analysis results

## Effects of the chosen regressors on eta23

	Parameters in the analysis model	Parameters in the generating process
eta11	-0.10	0.00
eta12	0.36	0.40

## The generating process



## The analysis model

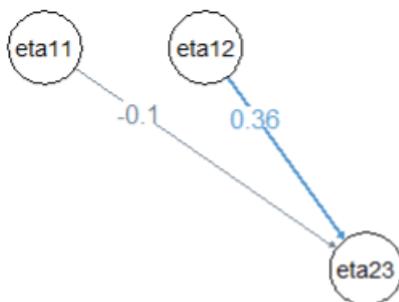


Figure 13: Display of the analysis model

## 6. Discussion

Considering the many unresolved questions concerning causality in stochastic processes the app presented in this thesis will hopefully prove useful as it can offer a rather intuitive and explorative way to approach causality. Though building an actual theory of course needs a bottom up approach of defining the relevant concepts and building up logically from the more abstract to the more concrete, the top-down approach of this app can serve as a first entering point.

Having the ability to track how biases in analyses are introduced at the level of identification already gives the user the opportunity to develop a deeper understanding of how the principles of causality conditions work. The fact that one can find extensive literature on causal effects in discrete-time processes allows the user to retrace the claims made within the app and furthermore allows to see how some of the causality conditions from the literature on discrete-time processes may apply to continuous-time processes too.

A particular strength of this app is that it allows the user to keep apart questions of identification and estimation. For many practical users of statistical software the lines between those two concepts often seem to blur. However it is very important to keep these two things apart in order to adjust the method and statistical analysis at the correct points if the results are biased. If the causal effect one is looking for is not identifiable with the experimental design that was chosen, it doesn't help to increase the number of observations or to use a more sophisticated algorithm for estimation. Likewise it doesn't help to use a model that should be able to identify causal effects when there are other problems that bias the estimation of those effects.

Looking 'under the hood' of the app there are still some aspects that could be optimized in the future. The most prominent is the fact that I didn't find a clean way of identifying the continuous-time drift parameters from the true covariance structure of a process. Though I found a workaround to compute the true (estimation error free) parameters, this way still technically involves an estimation algorithm that is time consuming and does not converge if the model for analysis fits the true model too poorly.

Another point is of course that the app makes several restrictive assumptions, for exam-

ple concerning the maximum number of processes and time points or the equality of the autoregressive and cross-lag effects over time. However, those assumptions are necessary to prevent the app from getting overloaded and confusing. All in all one can conclude that this app contributes its own part to a better understanding of causality in stochastic processes and all the exciting innovations that we encounter in this area of research at the moment.

## 7. Appendix

### A. Required R packages

Name	Version	Authors
shiny	1.1.0	Chang, Cheng, Allaire, Xie, and McPherson (2018)
shinycssloaders	0.2.0	Sali (2017)
ctsem	2.5.0	Driver, Oud, and Voelkle (2017)
MASS	7.3-49	Venables and Ripley (2002)
stringr	1.3.1	Wickham (2018)
qgraph	1.5	Epskamp, Cramer, Waldorp, Schmittmann, and Borsboom (2012)

---

## B. Code

### ui.R

```
library(shinycssloaders)

shinyUI(fluidPage(

  fluidRow(
    #####Input#####
    column(6,
      h3("Input"),
    #####Generating model#####
    tabsetPanel(type = "tabs",
      tabPanel("Generating process",
    #####Modelparameters#####
      tabsetPanel(type="pills",
        tabPanel("Process parameters",
          fluidRow(
            column(6,
              wellPanel(h5("Temporal structure in the generating
                process"),
                selectInput("true.time",
                  "",
                  choices = c("Discrete-time processes", "
                    Continuous-time processes"),
                  selected = "Discrete-time processes")),
            wellPanel(h5("Treatment variable"),
              uiOutput("treatment"),
              uiOutput("treatmentNote")),
            wellPanel(h5("Initial variances"),
```

```
sliderInput("var1",
            "eta11:",
            min=0,
            max=200,
            step=1,
            value=100,
            ticks=F),

sliderInput("var2",
            "eta21:",
            min=0,
            max=200,
            step=1,
            value=100,
            ticks=F),

conditionalPanel(
  condition= "input.NumberProcessesInp == 3",
  sliderInput("var3",
              "eta31:",
              min=0,
              max=200,
              step=1,
              value=100,
              ticks=F)
)
),
column(6,
  wellPanel(h5("Number of processes"),
            numericInput("NumberProcessesInp",
                          "",
                          min=2,
                          max=3,
```

```
value=2)),  
  
wellPanel(uiOutput("timeTitle"),  
  sliderInput("NumberOfPoints",  
    "",  
    min=2,  
    max=6,  
    value = 3,  
    step = 1,  
    ticks=F),  
  uiOutput("timeNote")  
) ,  
  
wellPanel(h5("Initial correlation"),  
  sliderInput("cov12",  
    "eta11 and eta21",  
    min=-1,  
    max=1,  
    step=0.1,  
    value=0.5,  
    ticks=F),  
  
  conditionalPanel(  
    condition= "input.NumberProcessesInp == 3",  
    sliderInput("cov13",  
      "eta11 and eta31:",  
      min=-1,  
      max=1,  
      step=0.1,  
      value=0.5,  
      ticks=F),  
    sliderInput("cov23",  
      "eta21 and eta31:",
```

```

                                min=-1,
                                max=1,
                                step=0.1,
                                value=0.5,
                                ticks=F)
                                )
                                )
                                )
                                ),
#####Effects#####
    tabPanel("Effects",
      fluidRow(
        div(uiOutput("effectsNote"), style='padding:20px;'),
        column(6,
#####Autoeffects & treatment#####
          wellPanel(h5("Auto-effects"),
            sliderInput("Eff11",
              "eta1t",
              min=-3,
              max=3,
              step=0.1,
              value=-0.5,
              ticks=F),

            sliderInput("Eff22",
              "eta2t:",
              min=-3,
              max=3,
              step=0.1,
              value=-0.2,
              ticks=F),

          conditionalPanel(

```

```
condition= "input.NumberProcessesInp == 3",
sliderInput("Eff33",
            "eta3t:",
            min=-3,
            max=3,
            step=0.1,
            value=-0.3,
            ticks=F)
    )
),
####Crosseffects####
column(6,
       wellPanel(h5("Cross-effects"),
                 sliderInput("Eff12",
                             "etalt on eta2t:",
                             min=-3,
                             max=3,
                             step=0.1,
                             value=0.5,
                             ticks=F),

                 sliderInput("Eff21",
                             "eta2t on etalt:",
                             min=-3,
                             max=3,
                             step=0.1,
                             value=0.7,
                             ticks=F),

                 conditionalPanel(
                   condition= "input.NumberProcessesInp > 2",
                   sliderInput("Eff13",
                               "etalt on eta3t:",
                               min=-3,
```

```
max=3,
step=0.1,
value=0.4,
ticks=F),

sliderInput("Eff31",
  "eta3t on etalt:",
  min=-3,
  max=3,
  step=0.1,
  value=0.7,
  ticks=F),

sliderInput("Eff23",
  "eta2t on eta3t:",
  min=-3,
  max=3,
  step=0.1,
  value=0.3,
  ticks=F),

sliderInput("Eff32",
  "eta3t on eta2t:",
  min=-3,
  max=3,
  step=0.1,
  value=0.1,
  ticks=F)
)
)
)),

####Treatment effects####
conditionalPanel(condition="input.treatment != 'No Treatment'",
```

```

wellPanel(fluidRow(
  div("Treatment effects", style='padding:17px;'),
  column(4,
    uiOutput("treatEff1")),
  column(4,
    uiOutput("treatEff2")),
  column(4, conditionalPanel(condition="input.
    NumberProcessesInp>2",
    uiOutput("treatEff3")))))
)),
#####Download Data#####
tabPanel("Let me download data",br(),br(),
  div("Here you can download a dataset generated from the
    specified parameters.", style='padding:20px;') ,br(),
column(8,
wellPanel(
  sliderInput("ObsNum",
    "Number of observations",
    min=1,
    max=10000,
    step=1,
    value=100,
    ticks=F)
),
wellPanel(
  "Specify the residual terms to add for the simulation of
    empirical data",br(),br(),
  sliderInput("res1",
    "Residual variance for etalt",
    min=0,
    max=10,
    step=0.1,
    value=0.1,
    ticks=F),

```

```
sliderInput("res2",
            "Residual variance for eta2t",
            min=0,
            max=10,
            step=0.1,
            value=0.1,
            ticks=F),

conditionalPanel(condition= "input.NumberProcessesInp > 2",
                 sliderInput("res3",
                             "Residual variance for eta3t",
                             min=0,
                             max=10,
                             step=0.1,
                             value=0.1,
                             ticks=F))),

wellPanel(uiOutput("downloadNote"),
          downloadButton("downloadData", "Download"), "Click to download
          data as csv-file.")
))
)),
####Analysemodell####
tabPanel("Model for analysis",br(),br(),
         uiOutput("analyse.time"),
         uiOutput("analysed.model"),
         uiOutput("regressors")
        ))
),
####Output (right)####
column(6,
       h3("Output"),
       tabsetPanel(type = "tabs",
```

```

tabPanel("Display the generating process",
  br(),br(),
  actionButton("updateDiagram", "Update diagram"),
  br(),br(),
  h4("Path diagram of the generating process"),
  "This diagram displays the process you have specified.",
  br(),br(),
  helpText("All manifest indicators and
  residual terms are omitted."),
  withSpinner(plotOutput("plot", width="auto"))
),
tabPanel("Display analysis results",
  h4("Analysis results"),
  actionButton("updateAnalysis", "Update analysis/diagrams"), br
  (), br(),
  uiOutput("analysisNote"),br(),br(),
  uiOutput("results.title"),br(),
  textOutput("ctsemMessage"),br(),
  withSpinner(tableOutput("analysisResult")),br(),
  uiOutput("TDpredTitle"),
  uiOutput("TDpred"),
  strong("The generating process"),
  plotOutput("plot1"),
  strong("The analysis model"),
  plotOutput("plot2")
)
)
)
)
))

```

server.R

```
library(stringr)
library(ctsem)
library(MASS)
library(qgraph)
library(shiny)

shinyServer(function(session, input, output) {

####reactive ui elements####
output$analyse.time<-renderUI({
  if(is.null(true.time())|is.null(treat())){return()}

  if(true.time()==1 & treat()==1){
    selectInput("analyseTime",
                "Temporal structure of the analysis model has to be
                continuous if treatment is chosen",
                choices = c("Continuous-time processes"),
                selected = "Continuous-time processes")
  }else{
    selectInput("analyseTime",
                "Temporal structure of the analysis model",
                choices = c("Discrete-time processes", "Continuous-time
                processes"),
                selected = "Discrete-time processes")
  }
})

output$treatment<-renderUI({
  if(is.null(tim())){return()}
  treat.choice<-if(true.time()==0){
    c("No Treatment",paste0("Treatment at t=", 1:(tim
    ()-1)))
  }else{
    c("No Treatment", "Treatment")}
  selectInput("treatment","", choices=treat.choice, selected="No
```

```
Treatment")
}))

#####Reactive Treatment effects#####

treat.time<-reactive({as.numeric(str_extract(input$treatment, "[1-9]"))
})

output$treatEff1<-renderUI({if(treat()==1)
  if(true.time()==1){
    sliderInput("treatEff1",
               "Effect on etalt",
               min=-3,
               max=3,
               step=0.1,
               value=0.5,
               ticks=F)
  }else{
    lapply((treat.time()+1):tim(), function(i){
      sliderInput(paste0("treatEff1",i),
                 paste0("Effect on eta1",i),
                 min=-3,
                 max=3,
                 step=0.1,
                 value=0,
                 ticks=F)})
  })
})

output$treatEff2<-renderUI({if(treat()==1)
  if(true.time()==1){
    sliderInput("treatEff2",
               "Effect on eta2t",
               min=-3,
```

```
        max=3,
        step=0.1,
        value=0.3,
        ticks=F)
    }else{
      lapply((treat.time()+1):tim(), function(i){
        sliderInput(paste0("treatEff2",i),
                    paste0("Effect on eta2",i),
                    min=-3,
                    max=3,
                    step=0.1,
                    value=0,
                    ticks=F)}})
    })
```

```
output$treatEff3<-renderUI({if(treat()==1)
  if(true.time()==1){
    sliderInput("treatEff3",
                "Effect on eta3t",
                min=-3,
                max=3,
                step=0.1,
                value=0.6,
                ticks=F)
  }else{
    lapply((treat.time()+1):tim(), function(i){
      sliderInput(paste0("treatEff3",i),
                  paste0("Effect on eta3",i),
                  min=-3,
                  max=3,
                  step=0.1,
                  value=0,
                  ticks=F)}})
  })
```

```
}}
```

```
output$analysed.model<-renderUI({
  if(is.null(analyse.time())){return()}
  if(analyse.time()==0){
    selectInput("regressand",
               "Choose the outcome:",
               choices = variables()[(n()+1):(length(variables()))],
               selected="eta23")
  }else{
    selectInput("analysed.processes",
               "Which processes shall be included in the analysis?",
               choices = if(treat()==0){paste0(processes(),"t")}else{c(
                 paste0(processes(),"t"),"X")},
               selected="etalt",
               multiple = T)
  }
})
```

```
output$regressors<-renderUI({
  if(is.null(input$regressand)|is.null(variables())){return()}
  }else if(analyse.time()==1){
    return()}else{
    vars<-isolate(variables())
    y<-input$regressand
    wahl<-vars[substring(vars,5) < substring(y,5)]
    if(treat()==1) {wahl<-c("X",wahl)}

    selectInput("regressoren",
               "Choose the regressors:",
               choices = wahl,
               selected = "etall",
               multiple = T)}
```

```

}))

#####reactive headings and lines of text#####
output$analysisNote<-renderUI({ifelse(analyse.time()==1,"Note that a
  continuous-time model can take up to two minutes of computation
  time.", return())})

output$treatmentNote<-renderUI({helpText(ifelse(true.time()==1, "For
  continuous-time models we assume the treatment takes place in the
  middle observation.", return()))})

output$effectsNote<-renderUI({helpText(ifelse(true.time()==1, "Note
  that for continuous-time models the effects can not be intepreted
  in the same way as discrete effects, for an explanation see the
  manual of this app.",return()))})

output$downloadNote<-renderUI(helpText(ifelse(true.time()==1, "
  Simulation of large datasets according to a continuous model can
  take some time.",return()))})

output$timeTitle<-renderUI({ifelse(true.time()==0,"Number of time
  points","Number of measurement occasions")})

output$timeNote<-renderUI({helpText(ifelse(true.time()==1&treat()==1,"A
  minimum of 3 observation points is advisable for correct
  calculation of treatment effects.",return()))})

results.title<-eventReactive(input$updateAnalysis,{ifelse(analyse.time
  ())==0,paste("Effects of the chosen regressors on",as.character(
  input$regressand)), "Calculated drift matrix"},ignoreNULL = F)
output$results.title<-renderUI(strong({results.title()}))

TDpredTitle<-eventReactive(input$updateAnalysis,"Calculated treatment
  effects")

```

```

output$TDpredTitle<-renderUI({
  if(analyse.time()==1 & "X" %in% analysed.processes())
    {strong(TDpredTitle())}else{return()}
})

#####Read in data#####
tim<-reactive({input$NumberOfPoints})
n<-reactive({input$NumberProcessesInp})

true.time<-reactive({ifelse(input$true.time=="Discrete-time processes"
  ,0,1)})
analyse.time<-reactive({if(is.null(input$analyseTime)){return()}else if
  (input$analyseTime=="Discrete-time processes"){0}else{1}})

treat<-reactive({if(is.null(input$treatment)){0}else if(input$treatment
  == "No Treatment"){0}else{1}})

regressors<-reactive({input$regressoren})

analysed.processes<-reactive({input$analysed.processes})

##Variablennamen
processes<-reactive({
  paste0("eta",1:input$NumberProcessesInp)})

variables<- reactive({if(is.null(tim())){return()}
  paste0(processes(), rep(1:tim(), each=n()))})
exos<- reactive({
  if(treat()==0){paste0(processes(),1)
  } else {c("X",paste0(processes(),1))}})

endos<-reactive({if(is.null(tim())){return()}
  paste0(processes(), rep(2:tim(), each=n()))})

```

```

###Varianzen/Kovarianzen
varcov<-reactive({
  varcov<-matrix(c(input$var1,input$cov12*sqrt(input$var1)*sqrt(input$
    var2), input$cov13*sqrt(input$var1)*sqrt(input$var3),
      input$cov12*sqrt(input$var1)*sqrt(input$var2), input$
        var2, input$cov23*sqrt(input$var2)*sqrt(input$var3),
          input$cov13*sqrt(input$var1)*sqrt(input$var3), input$
            cov23*sqrt(input$var2)*sqrt(input$var3), input$var3)
        ,3, byrow=T,
        dimnames = list(c("eta11","eta21","eta31"),c("eta11","
          eta21","eta31"))[1:n(),1:n()])
})

###Effekte
effects<-reactive({
  m<-matrix(c(input$Eff11, input$Eff21, input$Eff31,
    input$Eff12, input$Eff22, input$Eff32,
    input$Eff13, input$Eff23, input$Eff33), 3,byrow=T,
    dimnames = list(c("eta1","eta2","eta3"),c("eta1","eta2","
      eta3"))[1:n(),1:n()])

  if(true.time()==1){
    validate(
      need(det(m) != 0 & input$Eff11!=input$Eff22, "The effect
        matrix you chose cannot be solved. Please choose different
        effects (e.g. auto-effects shouldn't add up to zero)")})
    m
  })

  treat.effects<-reactive({if(treat()==1)
    if(true.time()==0){

```

---

```

lhs1 <- paste0("e1", (treat.time()+1):tim())
rhs1 <- paste0("input$treatEff1", (treat.time()+1):tim())
l1 <- paste(paste(lhs1, rhs1, sep="<-"), collapse=";")

lhs2 <- paste0("e2", (treat.time()+1):tim())
rhs2 <- paste0("input$treatEff2", (treat.time()+1):tim())
l2 <- paste(paste(lhs2, rhs2, sep="<-"), collapse=";")

lhs3 <- paste0("e3", (treat.time()+1):tim())
rhs3 <- paste0("input$treatEff3", (treat.time()+1):tim())
l3 <- paste(paste(lhs3, rhs3, sep="<-"), collapse=";")

if(n()>2){eval(parse(text=c(l1,l2,l3)))}else{eval(parse(text=c(l1,l2
)))}

ls<-ls()[grepl("^e",ls())]
ls<-mysort(ls,3)
l4<-paste0("c(",paste(ls,collapse=","),")")
matrix(eval(parse(text=l4)), ncol=1)
}else{
matrix(c(input$treatEff1, input$treatEff2, input$treatEff3),ncol=1,
nrow = 3)[1:n(),]
}
})

#####Process data#####

####True model discrete####
#Effects of endogenous variables
B<-reactive({
if(is.null(tim())){return()}
B<-matrix(0,nrow=n()* (tim()-1), ncol=n()* (tim()-1), dimnames = list(

```

```

        endos(), endos()))
  if(tim()>2) {
    for(i in (1:(tim()-2))){
      B[(i*n()+(1:n())),((i-1)*n()+(1:n()))] <- effects()
    }
  }
  B
})

#Effects of exogenous variables on endogenous variables
Gamma<-reactive({
  if(is.null(tim())){return()}
  treat.time<-if(treat()==0){NULL} else {as.numeric(str_extract(input$
    treatment, "[1-9]")+1}
  Gamma<-matrix(0,nrow = n()*(tim()-1), ncol = n()+treat(), dimnames =
    list(endos(), exos()))
  if(treat()==1) Gamma[(n()*(treat.time()-1)+1):(n()*(tim()-1)),1]<-
    treat.effects()
  Gamma[(1:(n())),((1+treat()):n()+treat())]<-effects()
  Gamma
})

#Complete effects matrix
all.effects<-reactive({
  if(is.null(tim())){return()}
  temp<-cbind(Gamma(), B())
  temp2<-matrix(0, nrow = n()+treat(), ncol=n()*tim()+treat())
  rownames(temp2)<-exos()
  rbind(temp2, temp)
})

#Variance & covariance of exogenous variables
Phi<-reactive({
  Phi<-matrix(0,nrow =n()+treat(), ncol=n()+treat())

```

```

Phi[1,1]<-100
Phi[(1+treat()):n()+treat()),(1+treat()):n()+treat())<-varcov()
dimnames(Phi)<-list(exos(), exos())
Phi
})

#Residual variances and covariances
Psi<-reactive({
  Psi<-diag(c(input$res1, input$res2, input$res3)[1:n()], nrow= n()*(
    tim()-1))
  rownames(Psi)<-endos()
  colnames(Psi)<-endos()
  Psi
})

#Identity matrix
I<-reactive({
  diag(nrow = n()*(tim()-1))
})

#Variances and covariances of endogenous variables
Erg1<-reactive({solve(I()-B())%*% (Gamma() %*% Phi() %*% t(Gamma())) +
  Psi() %*% solve(I()-t(B()))})

#Covariance of endogenous and exogenous variables
Erg2<-reactive({Phi() %*% t(Gamma()) %*% solve(I()-t(B()))})

#Complete variances and covariances
gesamt<-reactive({
  if(true.time()==1){
    validate(need(treat()==0, ""))
    erg<-cov()
    colnames(erg)<-variables()
    rownames(erg)<-variables()
  }
})

```

```

    erg
  }else{
    oben<-cbind(Phi(),Erg2())
    unten<-cbind(t(Erg2()), Erg1())
    rbind(oben,unten)}
  })

####True model continuous####
fit1<-reactive({
  model<-ctModel(
    Tpoints=tim(),
    n.latent=n(),
    n.manifest=n(),
    MANIFESTVAR=t(chol(diag(.001,n()))),
    LAMBDA=diag(1,n()),
    DRIFT=effects(),
    DIFFUSION=t(chol(matrix(rep(0.01,n()*n()),n()))),
    TOVAR=t(chol(unnname(varcov()))))

  if(treat()==1){
    TDPREDMEANS=matrix(0,tim(),1)
    TDPREDMEANS[ceiling(tim()/2)]=1
    TDPREDEFFECT<-treat.effects()
    model<-ctModel(
      Tpoints=tim(),
      n.latent=n(),
      n.manifest=n(),
      n.TDpred=treat(),
      TDPREDMEANS=TDPREDMEANS,
      TDPREDEFFECT = matrix(TDPREDEFFECT,nrow = n(),ncol = 1),
      MANIFESTVAR=t(chol(diag(.001,n()))),
      LAMBDA=diag(1,n()),
      DRIFT=effects(),
      DIFFUSION=t(chol(matrix(rep(0.01,n()*n()),n()))),

```

```

    TOVAR=t(chol(unname(varcov()))))
  }

  d<-ctGenerate(model, 10)
  ctFit(dat=d, ctmodelobj=model, nofit = T)
})

cov<-reactive({
  if(true.time()==1){
    mxGetExpected(fit1()$mxobj, "covariance")
  }else{
    gesamt()
  })
})

mean<-reactive({
  if(true.time()==1){
    mxGetExpected(fit1()$mxobj, "means")
  }else{
    rep(1,n()*tim())
  })
})

data<-reactive({
  cov<-if(treat()==0|input$true.time=="Continuous-time processes"){cov
    ()}else{cov()[2:nrow(cov()),2:ncol(cov())]}
  data<-mvrnorm(
    n=500,
    mu=mean(),
    Sigma = cov,
    empirical = T
  )
  data<-as.data.frame(data)
  names<-paste0(paste0("Y",1:n()),rep(paste0("_T",0:(tim()-1)),each=n
    ()))
  colnames(data)<-names

```

```

colnames(data)[is.na(colnames(data))]<-paste0("TD1_T",0:(tim()-1))
if(true.time()==0 & treat()==1){
  TD<-matrix(0,ncol = tim(),nrow=500)
  TD[,ceiling(tim()/2)]<-1
  colnames(TD)<-paste0("TD1_T",0:(tim()-1))
  data<-cbind(data,TD)}
m<-matrix(1,ncol=tim()-1,nrow=500)
colnames(m)<-paste0("dT",1:(tim()-1))
data<-cbind(data,m)
data
})

####Download data according to model####
downData<-reactive({
  if(true.time()==1){
    model<-ctModel(
      Tpoints=tim(),
      n.latent=n(),
      n.manifest=n(),
      MANIFESTVAR=t(chol(diag(.001,n()))),
      LAMBDA=diag(1,n()),
      DRIFT=effects(),
      DIFFUSION = t(chol(diag(c(input$res1, input$res2, input$res3)
        [1:n()],n()))),
      # DIFFUSION=t(chol(matrix(rep(0.01,n()*n()),n()))),
      TOVAR=t(chol(unname(varcov()))))

    if(treat()==1){
      TDPREDMEANS=matrix(0,tim(),1)
      TDPREDMEANS[ceiling(tim()/2)]=1
      TDPREDEFFECT<-treat.effects()
      model<-ctModel(
        Tpoints=tim(),
        n.latent=n(),

```

```

n.manifest=n(),
n.TDpred=treat(),
TDPREDMEANS=TDPREDMEANS,
TDPREDEFFECT = matrix(TDPREDEFFECT,nrow = n(),ncol = 1),
MANIFESTVAR=t(chol(diag(.001,n()))),
LAMBDA=diag(1,n()),
DRIFT=effects(),
DIFFUSION = t(chol(diag(c(input$res1, input$res2, input$
      res3)[1:n()],n()))),
# DIFFUSION=t(chol(matrix(rep(0.01,n()*n()),n()))),
TOVAR=t(chol(unnname(varcov()))))
}
ctGenerate(model, input$ObsNum)
}else{
mean<-if(treat()==0){mean()}else{c(1,mean())}
mvrnorm(n=input$ObsNum, mu=mean, Sigma = gesamt())
}
})

output$downloadData <- downloadHandler(
  filename = function() {
    ifelse(true.time()==0,"DataForRegression.csv", "DataForCtsem.csv"
  )
},
content = function(file) {
  write.csv(downData(), file, row.names = FALSE)
}
)

#### Model for analysis ####
####Discrete####
#Indicators of regressors

```

---

```

uv.num<-reactive({sort(charmatch(input$regressoren, colnames(gesamt()))
)})
av.num<-reactive({charmatch(input$regressand, colnames(gesamt()))})

Regressorenmatrix<-reactive({gesamt()[uv.num(), uv.num()]})

KovMitAv<-reactive({as.matrix(gesamt()[uv.num(), av.num()])}) #Kovarianz
    Regressoren mit Regressand und Entfernung irrelevanter Variablen

##sorting names of regressors
mysort<-function(x, last){
  temp<-x[order(as.numeric(substring(x, last-1)), na.last = F)]
  erg<-temp[order(as.numeric(substring(temp, last)), na.last = F)]
  erg
}

Koeffizienten<-reactive({
  estimated<-round(solve(Regressorenmatrix())%*%KovMitAv(), 2)
  namen<-mysort(input$regressoren, 5)
  if(true.time()==0){
    true<-all.effects()[av.num(), uv.num()]
    d<-data.frame(estimated, true, row.names=namen)
    names(d)<-c("Parameters in the analysis model", "Parameters in the
        generating process")
    d
  }else{
    d<-data.frame(estimated, row.names=namen)
    names(d)<-"Parameters in the analysis model"
    d
  }
})

####Continuous####

```

---

```

Drift<- eventReactive(input$updateAnalysis, {
  if(is.null(analysed.processes())) {return()}

  n<-if("X" %in% analysed.processes()) {length(analysed.processes())-1}
  else{length(analysed.processes())}
  numbers<-na.exclude(str_extract(analysed.processes(), "[1-9]"))
  chosen<-paste0(paste0("Y", numbers), rep(paste0("_T", 0:(tim()-1)), each
    =n))
  if("X" %in% analysed.processes()) {chosen<-c(chosen, paste0("TD1_T"
    , 0:(tim()-1)))}
  chosen<-c(chosen, paste0("dT", 1:(tim()-1)))
  data<-data()[, names(data)%in%chosen]
  names(data)[1:(n*tim())]<-paste0(paste0("Y", (1:n)), rep(paste0("_T"
    , 0:(tim()-1)), each=n))
  TDPREDMEANS=matrix(0, 1*tim(), 1)
  TDPREDMEANS[ceiling(tim()/2)]=1
  TDPREDEFFECT<-treat.effects()

  model<-ctModel(
    Tpoints=tim(),
    n.latent=n,
    n.manifest=n,
    MANIFESTVAR=t(chol(diag(.001, n))),
    LAMBDA=diag(1, n),
    #DIFFUSION = t(chol(diag(c(input$res1, input$res2, input$res3)[1:
      n()], n()))),
    DIFFUSION=t(chol(matrix(rep(0.01, n*n), n))),
    T0VAR=t(chol(unnname(varcov()[as.numeric(numbers), as.numeric(
      numbers)])))

  if("X" %in% analysed.processes()) {
  model<-ctModel(
    Tpoints=tim(),
    n.latent=n,

```

```

n.manifest=n,
n.TDpred=1,
TDPREDMEANS=TDPREDMEANS,
MANIFESTVAR=t(chol(diag(.001,n))),
LAMBDA=diag(1,n),
# DIFFUSION = t(chol(diag(c(input$res1, input$res2, input$res3)[1:
  n()],n()))),
DIFFUSION=t(chol(matrix(rep(0.01,n*n),n))),
TOVAR=t(chol(varcov()[as.numeric(numbers),as.numeric(numbers)]))
}

ctsemMessage<-capture.output(fit2<-ctFit(dat = data, ctmodelobj =
  model), type="message")
drift<-round(summary(fit2)$DRIFT,2)
tref<-summary(fit2)$TDPREDEFFECT
colnames(drift)<-paste0("eta",numbers)
rownames(drift)<-colnames(drift)
list(t(drift),tref,ctsemMessage)
}, ignoreNULL = F)

#####Results for output#####

###Complete path diagram###
graph<-eventReactive(c(input$updateDiagram,input$updateAnalysis),{
  m<- if(true.time()==0){
    t(all.effects())
  }else{
    if(treat()==0){
      t(effects())
    }else{
      rbind(c(0,t(treat.effects())),(cbind(0,t(effects()))))
    }
  }
})

```

```

lay<- if(true.time()==0){
  if(treat()==1){
    matrix(as.numeric(c(c(as.numeric(str_extract(input$treatment,
      "[1-9]"))-2, rep(-1:(tim()-2),each=n()), 0.5,rep(1:(2-n()),
      ,tim()))),ncol=2)
  }else{
    matrix(as.numeric(c(rep(-1:(tim()-2),each=n()),rep(1:(2-n()),
      tim()))),ncol=2)
  }
}else{
  if(treat()==1){
    matrix(c(c(0,c(0,1.5,3)[1:n()]),c(0.5,c(-0.5,0.5,-0.5)[1:n()]
      ), ncol=2)
  }else{
    matrix(c(c(0,1.5,3)[1:n()],c(-0.5,0.5,-0.5)[1:n()]), ncol=2)
  }
}

lab<-if(true.time()==0){if(treat()==1){c("X",variables())}else{
  variables()}}else{if(treat()==1){c("X","eta1t","eta2t","eta3t")[1:(
  n()+1)]}else{c("eta1t","eta2t","eta3t")[1:n()]}}

shape<-if(true.time()==0){
  if(treat()==1){c("square",rep("circle",length(variables()))}else{"
  circle"}
}else{
  if(treat()==1){c("square",rep("circle",n()))}else{"circle"}
}

##ignoring shape at the moment, to use, set shape=graph()$shape in the
  qqgraph command

list(m=m, lay=lay, lab=lab, shape=shape)
}, ignoreNULL = F)

```

```

output$plot<-renderPlot({
  g<-graph()
  vars<-matrix(c(0, isolate(input$cov12*sqrt(input$var1)*sqrt(input$
    var2)), isolate(input$cov13*sqrt(input$var1)*sqrt(input$var3)),
    isolate(input$cov12*sqrt(input$var1)*sqrt(input$var2)),
    0, isolate(input$cov23*sqrt(input$var2)*sqrt(input$
    var3)),
    isolate(input$cov13*sqrt(input$var1)*sqrt(input$var3)),
    isolate(input$cov23*sqrt(input$var2)*sqrt(input$
    var3)), 0), 3, byrow=T)[1:isolate(n()), 1:isolate(n())
  ]
  qqgraph(g$m, layout=g$lay, directed=T, rescale=F, mar=c(3, 10, 3, 35),
    aspect=T, parallelEdge=T, labels=g$lab, posCol="steelblue3", negCol=
    "slategray", edge.label.position=0.25, edge.labels=T, edge.width
    =0.3, edge.label.cex=1.25, fade=F)
  if(isolate(true.time()==0)){qqgraph(vars, labels=F, color="transparent"
    , layout=g$lay[(1+isolate(treat())):isolate(n()+treat())], plot=F
    , rescale=F, borders=F, curve=-5.5, curveAll=T, edge.width=0.1, edge
    .color="black", edge.labels=T, edge.label.cex=1.5, fade=F)}
})

```

```

output$plot1<-renderPlot({
  g<-graph()
  vars<-matrix(c(0, isolate(input$cov12*sqrt(input$var1)*sqrt(input$
    var2)), isolate(input$cov13*sqrt(input$var1)*sqrt(input$var3)),
    isolate(input$cov12*sqrt(input$var1)*sqrt(input$var2)),
    0, isolate(input$cov23*sqrt(input$var2)*sqrt(input$
    var3)),
    isolate(input$cov13*sqrt(input$var1)*sqrt(input$var3)),
    isolate(input$cov23*sqrt(input$var2)*sqrt(input$var3)
    ), 0), 3, byrow=T)[1:isolate(n()), 1:isolate(n())]
  qqgraph(g$m, layout=g$lay, directed=T, rescale=F, mar=c(3, 10, 3, 35),
    aspect=T, parallelEdge=T, labels=g$lab, posCol="steelblue3", negCol=
    "slategray", edge.label.position=0.25, edge.labels=T, edge.width

```

```

    =0.3,edge.label.cex=1.25, fade=F)
  if(isolate(true.time()==0)){qgraph(isolate(varcov()),labels=F,color=
    "transparent",layout=g$lay[(1+isolate(treat())):isolate(n()+
    treat())],plot=F, rescale=F,borders=F,curve=-5.5, curveAll=T,
    edge.width=0.1,edge.color="black",edge.labels=T,edge.label.cex
    =1.5, fade=F)}
})

###Analysed path diagram###
graph_part<-eventReactive(input$updateAnalysis,{
  complete<-graph()

  m<- if(analyse.time()==0){
    size<-length(input$regressoren)+1
    temp<-matrix(0,nrow =size,ncol =size)
    temp[size,]<-c(round(solve(Regressorenmatrix())%*%KovMitAv(),2)
      ,0)
    t(temp)
  }else{
    dr<-Drift()
    if(!"X" %in% analysed.processes()){
      erg<-dr[[1]]
    }else{
      erg<-rbind(c(0,dr[[2]]),(cbind(0,dr[[1]])))
    }
    erg
  }

  lay<- if(analyse.time()==0){
    temp<-matrix(as.numeric(c(rep(-1:(tim()-2),each=n()),rep(1:(2-n)
      ),tim()))),ncol=2)
    if(treat()==0){
      temp[c(uv.num(),av.num()),]
    }else{

```

```

        if("X" %in% regressors()){
            complete$lay[c(uv.num(), av.num()), ]
        }else{
            temp[c(uv.num()-1, av.num()-1), ]
        }
    }

}else{
    n<-if("X" %in% analysed.processes()){length(analysed.processes())
        -1}else{length(analysed.processes())}
    if("X" %in% analysed.processes()){
        matrix(c(c(0, c(0, 1.5, 3) [1:n]), c(0.5, c(-0.5, 0.5, -0.5) [1:n])),
            ncol=2)
    }else{
        matrix(c(c(0, 1.5, 3) [1:n], c(-0.5, 0.5, -0.5) [1:n]), ncol=2)
    }
}

lab<-if(analyse.time()==0){c(mysort(regressors(), 5), input$regressand
    )}else{analysed.processes()[order(as.numeric(substring(analysed.
    processes(), 4, 4)), na.last = F)]}#mysort(analysed.processes(), 5)
}

shape<-{"circle"}
# if("X" %in% regressors()){c("square", rep("circle", (length(
    regressors())-1))}else{"circle"}
# }else{
# if("X" %in% analysed.processes()){c("square", rep("circle", n))}
    else{"circle"}

    list(m=m, lay=lay, lab=lab, shape=shape)
}, ignoreNULL = F)

output$plot2<-renderPlot({

```

```

g<-graph_part()
qgraph(g$m, layout=g$lay, directed=T, rescale=F, mar=c(3,10,3,35),
  aspect=T, parallelEdge=T, labels=g$lab, posCol="steelblue3", negCol=
  "slategray", edge.label.position=0.25, edge.labels=T, edge.width
  =0.3, edge.label.cex=1.25, fade=F)
))

analysisResult<-eventReactive(input$updateAnalysis,
  {validate(need(!is.null(input$regressoren), "Please choose a
    regressor"))
  if(analyse.time()==1){
    erg<-Drift()[[1]]
    colnames(erg)<-analysed.processes()[1:ncol(erg)][order(as.numeric
      (substring(analysed.processes()[1:ncol(erg)],4,4)), na.last =
      F)]
    rownames(erg)<-analysed.processes()[1:ncol(erg)][order(as.numeric
      (substring(analysed.processes()[1:ncol(erg)],4,4)), na.last =
      F)]
    return(erg)
  }else{
    Koeffizienten()}}, ignoreNULL = F
)
#####
tdpredeffect<-eventReactive(input$updateAnalysis,
  {
    erg<-Drift()[[2]]
    rownames(erg)<-isolate(analysed.processes())[1:
      nrow(erg)][order(as.numeric(substring(
        analysed.processes()[1:nrow(erg)],4,4)), na
        .last = F)]
    return(erg)
  })

```

---

```

#####Objects to render#####

###Drift/Regression coefficients
output$analysisResult<-renderTable({
  analysisResult()},rownames=T,colnames=T, striped = T, bordered = T)

###Treatment effects
output$tdpredeffect<-renderTable(
  {tdpredeffect()},rownames=T,colnames=F, striped = T, bordered = T)

output$TDpred<-renderUI({
  req(input$updateAnalysis)
  if(is.null(analyse.time())||"X" %in% analysed.processes()){return()}
  if(analyse.time()==1)
    tableOutput("tdpredeffect")
})

output$ctsemMessage<-renderText({if("Retry limit reached" %in% Drift()
  [[3]]){
  "The algorithm didn't converge. If the calculated effects are very
  big, consider specifying a more realistic continuous process."
}
})

####Run even when hidden
outputOptions(output, "analysed.model", suspendWhenHidden = FALSE)
outputOptions(output, "regressors", suspendWhenHidden = FALSE)
outputOptions(output, "analyse.time", suspendWhenHidden = FALSE)
outputOptions(output, "treatEff1", suspendWhenHidden = FALSE)
outputOptions(output, "treatEff2", suspendWhenHidden = FALSE)
outputOptions(output, "treatEff3", suspendWhenHidden = FALSE)

```

```
#For Rinno
session$onSessionEnded(function() {
  stopApp()
  q("no")
})

})
```

---

## References

- Alpass, F. M. & Neville, S. (2003). Loneliness, health and depression in older males. *Aging & Mental Health*, 7(3), 212–216.
- Baron, R. M. & Kenny, D. A. (1986). The moderator mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of Personality and Social Psychology*, 51(6), 1173–1182.
- Cacioppo, J. T., Hawkley, L. C., Ernst, J. M., Burleson, M., Berntson, G. G., Nouriani, B., & Spiegel, D. (2006). Loneliness within a nomological net: An evolutionary perspective. *Journal of Research in Personality*, 40, 1054–1085.
- Chang, W., Cheng, J., Allaire, J., Xie, Y., & McPherson, J. (2018). *Shiny: web application framework for r*. R package version 1.1.0. Retrieved from <https://CRAN.R-project.org/package=shiny>
- Deboeck, P. R. & Preacher, K. J. (2015). No need to be discrete: a method for continuous time mediation analysis. *Structural Equation Modeling: A Multidisciplinary Journal*, 23(1), 61–75. doi:10.1080/10705511.2014.973960
- Driver, C., Oud, J. H. L., & Voelkle, M. C. (2017). Continuous time structural equation modeling with r package ctsem. *Journal of Statistical Software*, 77(5). doi:10.18637/jss.v077.i05
- Driver, C. & Voelkle, M. (2017). *Understanding the time course of interventions with continuous time dynamic models*.
- Epskamp, S., Cramer, A. O. J., Waldorp, L. J., Schmittmann, V. D., & Borsboom, D. (2012). qgraph: network visualizations of relationships in psychometric data. *Journal of Statistical Software*, 48(4), 1–18. Retrieved from <http://www.jstatsoft.org/v48/i04/>
- Fayant, M.-P., Sigall, H., Lemonnier, A., Retsin, E., & Alexopoulos, T. (2017). On the limitations of manipulation checks: an obstacle toward cumulative science. *International Review of Social Psychology*, 30(1), 125–130. doi:10.5334/irsp.102
- Jöreskog, K. G. & Sörbom, D. (1988). *Lisrel 7: a guide to the program and applications*. Spss.

- Mayer, A., Thoemmes, F., Rose, N., Steyer, R., & West, S. G. (2014). Theory and analysis of total, direct, and indirect causal effects. *Multivariate Behavioral Research*, *49*(5), 425–442.
- MPLUS (Version 8).[Computer Software].
- R Core Team. (2018). *R: a language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. Retrieved from <https://www.R-project.org/>
- Rosseel, Y. (2012). lavaan: an R package for structural equation modeling. *Journal of Statistical Software*, *48*(2), 1–36. Retrieved from <http://www.jstatsoft.org/v48/i02/>
- Sali, A. (2017). *Shinycssloaders: add css loading animations to 'shiny' outputs*. R package version 0.2.0. Retrieved from <https://CRAN.R-project.org/package=shinycssloaders>
- Sigall, H. & Mills, J. (1998). Measures of independent variables and mediators are useful in social psychology experiments: but are they necessary? *Personality and Social Psychology Review*, *2*(3), 218–226.
- Steiner, P. M., Cook, T. D., Shadish, W. R., & Clark, M. H. (2010). The importance of covariate selection in controlling for selection bias in observational studies. *Psychological Methods*, *15*(3), 250–267. doi:10.1037/a0018719
- Steyer, R., Mayer, A., & Fiege, C. (2014). Causal inference on total, direct, and indirect effects. In *Encyclopedia of quality of life and well-being research* (pp. 606–630). Springer Netherlands. doi:10.1007/978-94-007-0753-5\_295
- Venables, W. N. & Ripley, B. D. (2002). *Modern applied statistics with s* (Fourth). ISBN 0-387-95457-0. New York: Springer. Retrieved from <http://www.stats.ox.ac.uk/pub/MASS4>
- Voelkle, M. C., Oud, J. H. L., Davidov, E., & Schmidt, P. (2012). An SEM approach to continuous time modeling of panel data: relating authoritarianism and anomia. *Psychological Methods*, *17*(2), 176–192. doi:10.1037/a0027543
- Wickham, H. (2018). *Stringr: simple, consistent wrappers for common string operations*. R package version 1.3.1. Retrieved from <https://CRAN.R-project.org/package=stringr>
- Wright, S. (1934). The method of path coefficients. *The annals of mathematical statistics*, *5*(3), 161–215.

# Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources and that I have explicitly marked all material which has been quoted either literally or by content from the used sources. The work was not submitted or published in the same or a similar form in another test authority so far.

Jena, July 9, 2018

Julia Gantner