Special topics: Time series analysis, higher-order models,& oscillations,

Optional: latent change score models, accelerated longitudinal designs

Outline

- Time series (N = 1, T large)
- CARMA(p,q) models (higher order models)
- Oscillations (and a few words on the virtue of varying time intervals)
- Latent change score models
- Accelerated longitudinal designs

Time series (N = 1, T large)

Time series (N = 1, T large)

- Sometimes we are interested in just a single individual (patient, politician, athlete, …)
- Sometimes we are primarily interested in the (within-person) structure and less on the dynamics/development. For example, the personality structure of an individual.
- Sometimes we are interested in modeling complex, but systematic development of a single unit over time (e.g., changes in temperature or the activity of the sun)…
- …and sometimes such systematic developments are of an oscillating nature (e.g., "emotion as a thermostat"; Chow et al. 2005; psychological regulation; Deboeck & Bergeman; 2013; or circadian rhythms?).

Time series (N = 1, T large)

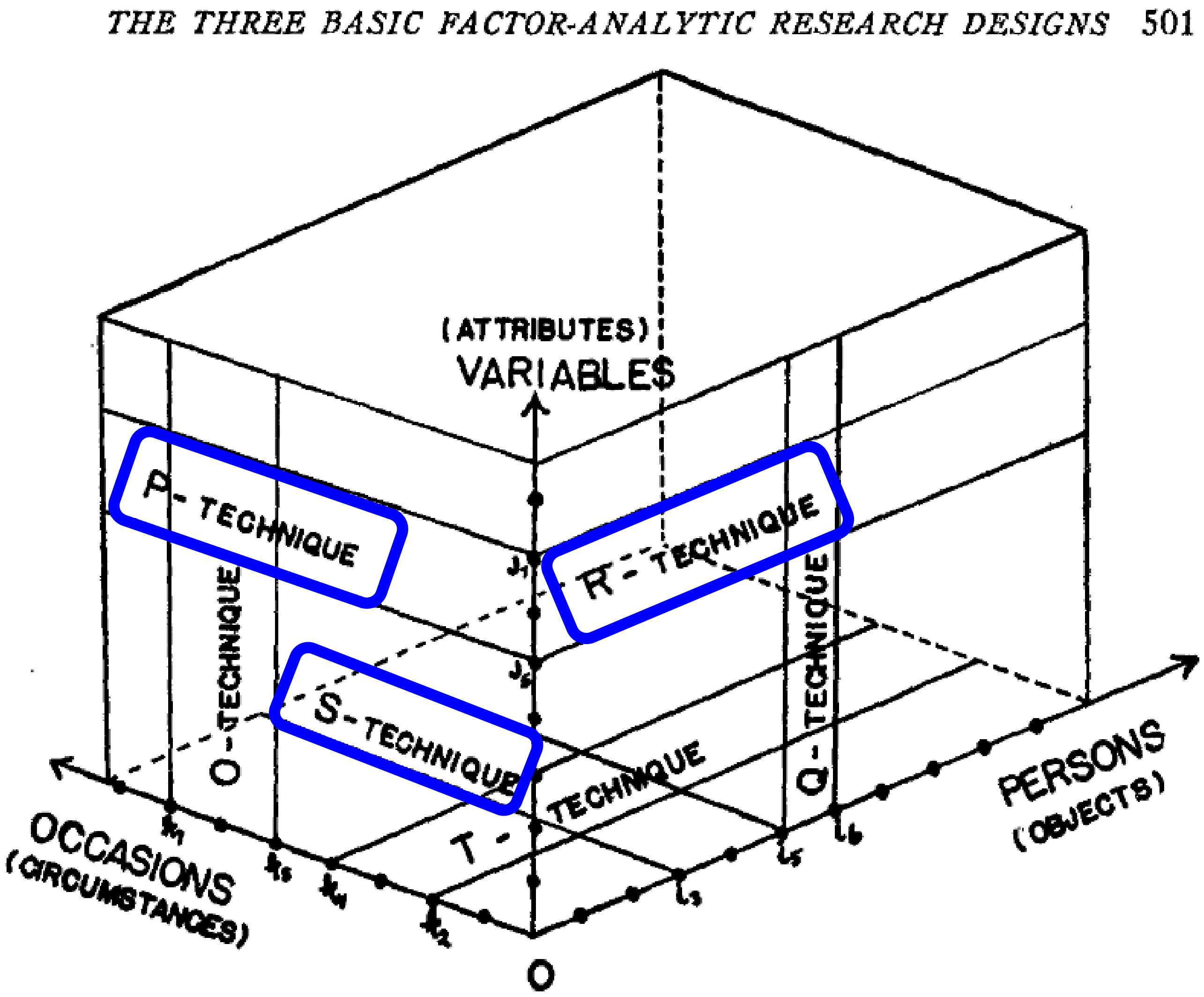


FIG. 1. THE COVARIATION CHART

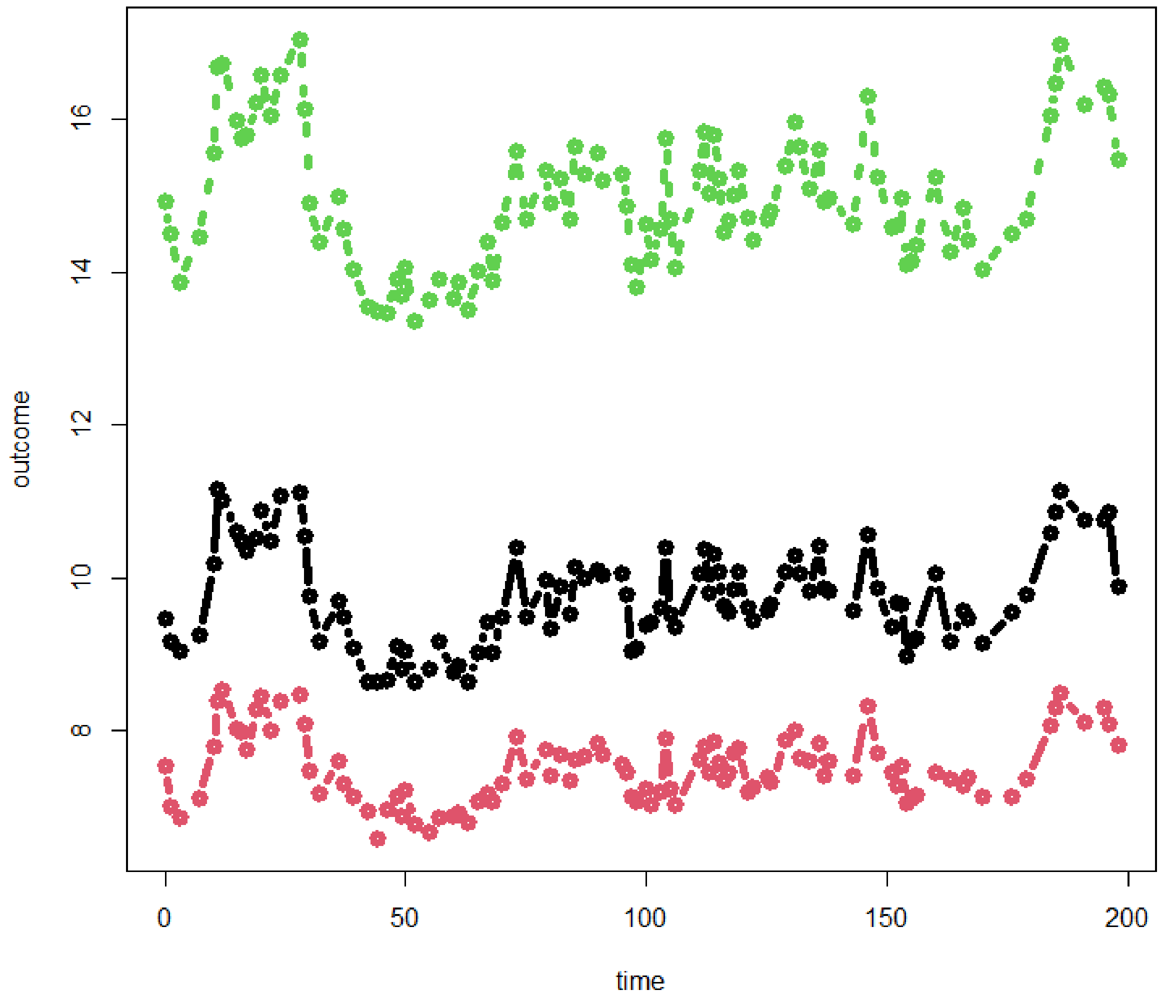
Original figure from Cattell (1952, p. 501)

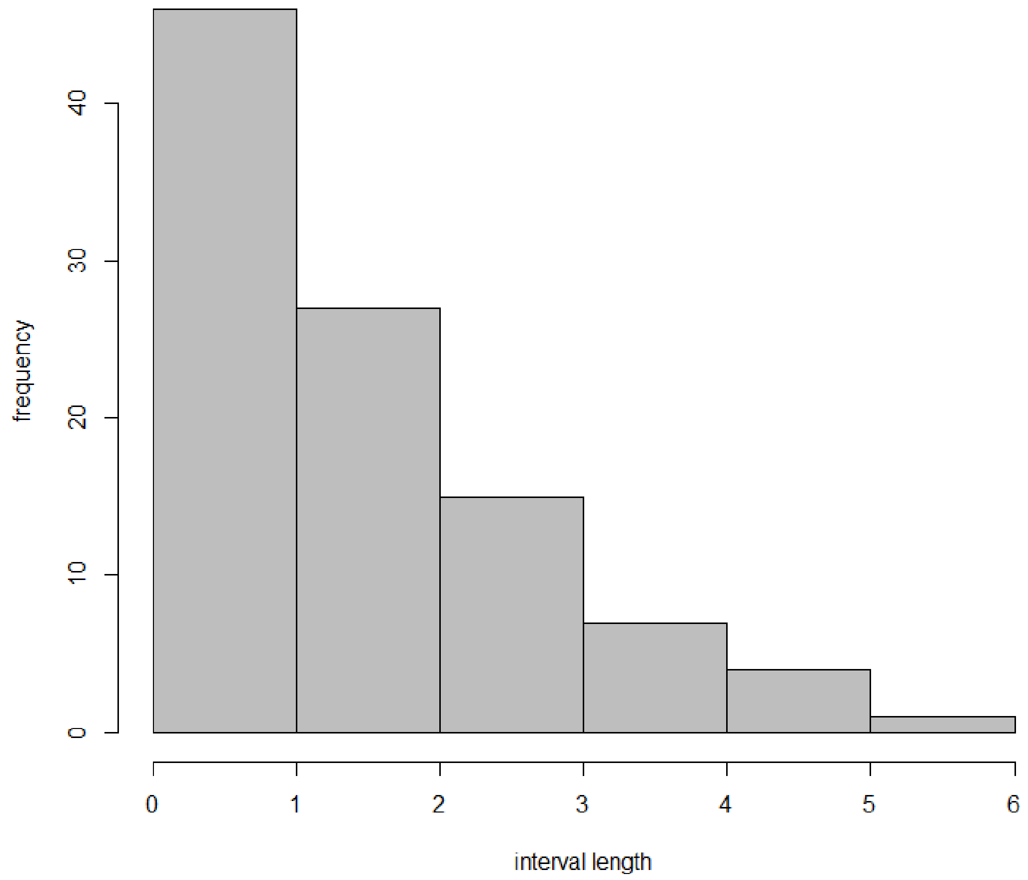
Time series (N = 1, T large)

- Historically, *P*-Technique was proposed to study the factorial structure within a unit/person.
- *P*-Technique was criticized, because it does not adequately account for the lagged covariance structure. Dynamic Factor Analysis (DFA; Molenaar, 1985) was proposed as an alternative that overcomes this limitation.
- Interestingly, in case of equidistant time intervals, *P*-Technique seems to perform reasonably well, despite its known limitations (Molenaar & Nesselroade, 2009).
- However, by design, neither DFA nor *P*-Technique addresses the problem of unequal time intervals. This is resolved in continuous time modeling.

Time series (N = 1, T large)

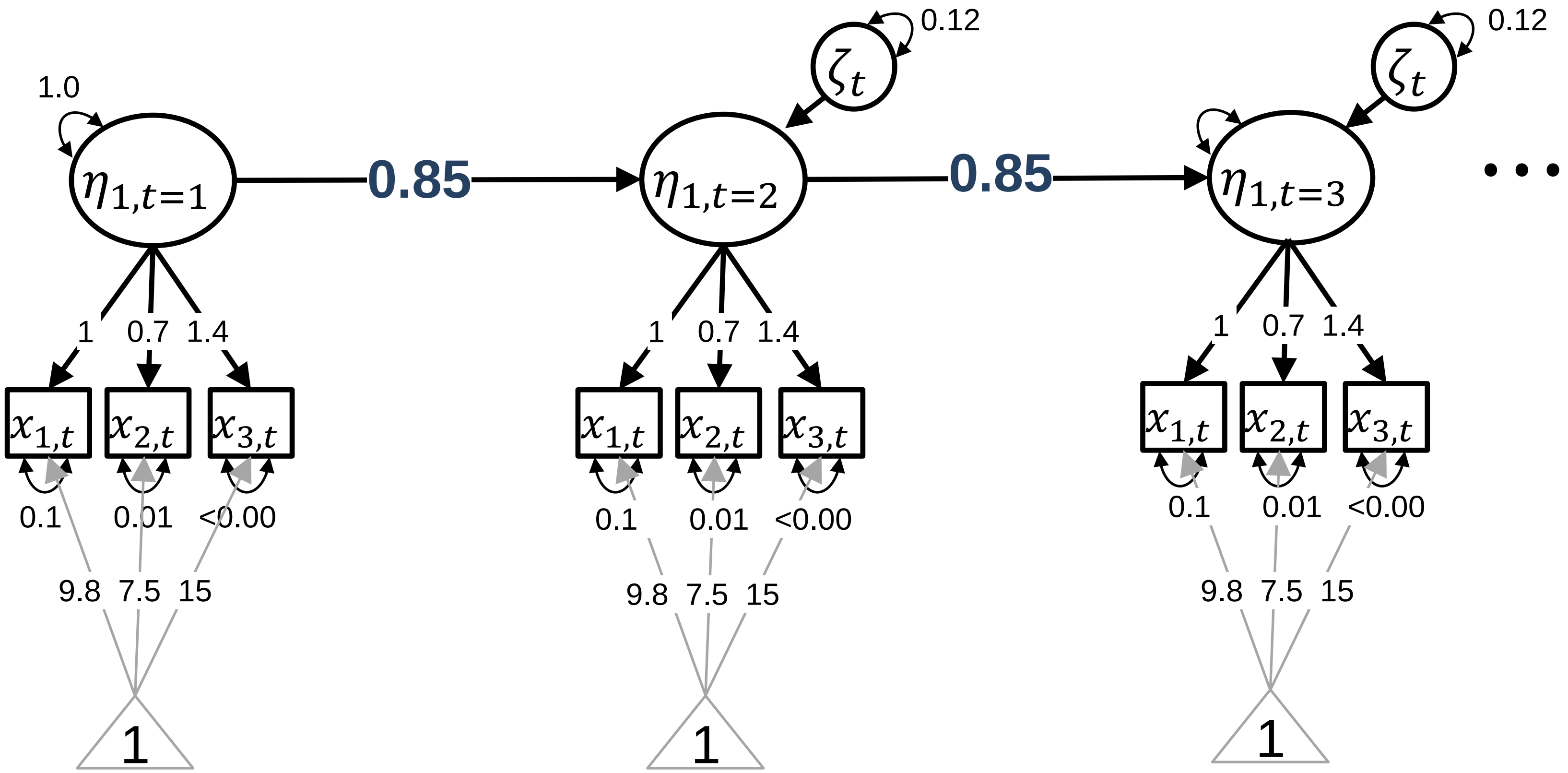
Dynamic factor analysis (U = 100 measurement occasions; N = 1; 3 manifest variables, differing time intervals)





Time series (N = 1, T large)

Dynamic factor analysis (U = 100 measurement occasions; N = 1; 3 manifest variables, differing time intervals)



Time series (N = 1, T large)

From theory to practice…

```
library(ctsem)
library(ctsemOMX)
data(ctExample3)

#interfacing to OpenMx (wide-data format)
DFAmodel <- ctModel(n.latent = 1, n.manifest = 3, Tpoints = 100,
            T0MEANS = matrix(0,1,1),
            LAMBDA = matrix(c(1, 'lambda2', 'lambda3'), nrow = 3, ncol = 1),
            MANIFESTMEANS = matrix(c('manifestmean1', 'manifestmean2', 'manifestmean3'), nrow = 3, ncol = 1),
            T0VAR = diag(1),
            type = "omx")
DFAmodelfit <- ctFit(ctExample3, ctmodelobj = DFAmodel)
summary(DFAmodelfit, verbose=T)
plot(DFAmodelfit)
```

Time series (N = 1, T large)

From theory to practice…

```
#Transforming data from wide to long
ctExample3long_intervals <- ctWideToLong(datawide = ctExample3, Tpoints=100, n.manifest=3) #convert wide to long format
ctExample3long <- ctDeintervalise(datalong = ctExample3long_intervals, id='id', dT='dT')   #convert intervals to absolute time

#interfacing to Stan (long-data format)
DFAmodelv2 <- ctModel(n.latent = 1, n.manifest = 3,
                    manifestNames = c('Y1', 'Y2', 'Y3'),
                    latentNames = "eta",
                    id = "id",
                    time = "time",
                    LAMBDA = matrix(c(1, 'lambda2', 'lambda3'), nrow = 3, ncol = 1),
                    MANIFESTMEANS = matrix(c('manifestmean1', 'manifestmean2', 'manifestmean3'), nrow = 3, ncol = 1),
                    T0VAR = diag(1),
                    T0MEANS = matrix(0,1,1),
                    type = "stanct")

DFAmodelv2fit <- ctStanFit(datalong=ctExample3long, ctstanmodel=DFAmodelv2, indvarying=F, optimize = TRUE, nopriors = TRUE)
summary(DFAmodelv2fit)
plot(DFAmodelv2fit)
```

CARMA(*p*,*q*) models (higher order models)

CARMA(*p,q*) models (higher order models)

- Just like in discrete time series (or in the general cross-lagged panel model), we can move beyond the AR(p = 1) model and move to higher order models of order p (autoregressive part) and q (moving average part).

- CARMA(p,q) models can be useful when we are interested in modeling more complex, but systematic, development over time (e.g., changes in temperature or the activity of the sun)…

- Given the matrix formulation of the ct model (and the corresponding software implementation), higher order models are easily specified by augmenting the latent state vector (i.e.,

In large parts the following content is based on:

Oud, J. H. L., Voelkle, M. C., & Driver, C. C. (2018). SEM Based CARMA Time Series Modeling for Arbitrary *N*. Multivariate Behavioral Research, 53(1), 36-56. https://doi.org/10.1080/00273171.2017.1383224

CARMA(p,q) models (higher order models)

Remember:

$$\mathrm{d}\boldsymbol{\eta}(t) = \left(\mathbf{A}\boldsymbol{\eta}(t) + \mathbf{b} + \mathbf{M}\boldsymbol{\chi}(t)\right)\mathrm{d}t + \mathbf{G}\mathrm{d}\mathbf{W}(t)\Big)$$

- Symbolically, this can also be written as*

$$\frac{\mathrm{d}\boldsymbol{\eta}(t)}{\mathrm{d}t} = \mathbf{A}\boldsymbol{\eta}(t) + \mathbf{b} + \mathbf{M}\boldsymbol{\chi}(t) + \mathbf{G}\frac{\mathrm{d}\mathbf{W}(t)}{\mathrm{d}t}$$

- With $\mathbf{y}(t) = \boldsymbol{\Lambda}\boldsymbol{\eta}(t) + \boldsymbol{\tau} + \boldsymbol{\varepsilon}(t)$.

- E.g., for a **uni**variate process, without intercepts and without error variance (to keep it as simple as possible), the measurement model would be:

$$y(t) = 1 \cdot \eta(t) + 0 + 0$$

*This is the notation used in Oud et al. (2018) and elsewhere. Don't get confused.

CARMA(*p*,*q*) models (higher order models)

- To incorporate higher order processes, we simply expand the state vector $\boldsymbol{\eta}(t)$, without changing the measurements (because the data do not change).

- E.g., to include a higher order term in a univariate model we write:

$$y(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix} (t)$$

- Note that η_2 is a latent variable without a reflective indicator. Only η_1 is connected to data.

In large parts the following content is based on:
Oud, J. H. L., Voelkle, M. C., & Driver, C. C. (2018). SEM Based CARMA Time Series Modeling for Arbitrary *N*. Multivariate Behavioral Research, 53(1), 36-56. https://doi.org/10.1080/00273171.2017.1383224

CARMA(p,q) models (higher order models)

- Plugging this into the general equation (and ignoring the "irrelevant" parts), yields:

$$\mathrm{d}\begin{pmatrix}\eta_1\\ \eta_2\end{pmatrix}(t) = \left(\mathbf{A}\begin{pmatrix}\eta_1\\ \eta_2\end{pmatrix}(t)\right)\mathrm{d}t + \mathbf{G}\mathrm{d}\mathbf{W}(t)$$

- To incorporate a higher order effect (second order derivative) we define $\eta_2 = \frac{\mathrm{d}\eta_1}{\mathrm{d}t}$, which yields

$$\frac{\mathrm{d}\begin{pmatrix}\eta_1\\ \eta_2\end{pmatrix}(t)}{\mathrm{d}t} = \frac{\mathrm{d}\begin{pmatrix}\eta_1\\ \frac{\mathrm{d}\eta_1}{\mathrm{d}t}\end{pmatrix}(t)}{\mathrm{d}t} = \begin{pmatrix}\frac{\mathrm{d}\eta_1}{\mathrm{d}t}\\ \frac{\mathrm{d}^2\eta_1}{\mathrm{d}t^2}\end{pmatrix}$$

on the left-hand side of the equation.

CARMA(*p*,*q*) models (higher order models)

- With this "dependent vector" we can now set up a second order model:

$$\begin{pmatrix} \frac{\mathrm{d}\eta_1}{\mathrm{d}t} \\ \frac{\mathrm{d}^2\eta_1}{\mathrm{d}t^2} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ a_0 & a_1 \end{pmatrix}}_{\textbf{A}\text{ matrix}} \begin{pmatrix} \eta_1 \\ \frac{\mathrm{d}\eta_1}{\mathrm{d}t} \end{pmatrix}$$

- Without matrix notation:

$$\frac{\mathrm{d}\eta_1}{\mathrm{d}t} = 1\frac{\mathrm{d}\eta_1}{\mathrm{d}t}$$

$$\frac{\mathrm{d}^2\eta_1}{\mathrm{d}t^2} = a_0\eta_1 + a_1\frac{\mathrm{d}\eta_1}{\mathrm{d}t}$$

first order drift effect

second order drift effect

CARMA(p,q) models (higher order models)

- Note that we expressed the univariate second order differential equation as a first order differential equation in matrix notation.

$$\begin{pmatrix} \frac{\mathrm{d}\eta_1}{\mathrm{d}t} \\ \frac{\mathrm{d}^2\eta_1}{\mathrm{d}t^2} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ a_0 & a_1 \end{pmatrix} \begin{pmatrix} \eta_1 \\ \frac{\mathrm{d}\eta_1}{\mathrm{d}t} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & G_0 \end{pmatrix} \frac{\mathrm{d}\mathbf{W}(t)}{\mathrm{d}t}$$

- This is a special case of general equation we are familiar with

$$\frac{\mathrm{d}\boldsymbol{\eta}(t)}{\mathrm{d}t} = \mathbf{A}\boldsymbol{\eta}(t) + \mathbf{0} + \mathbf{0} + \mathbf{G}\frac{\mathrm{d}\mathbf{W}(t)}{\mathrm{d}t}$$

- Thus, the solution of the (matrix) stochastic differential equation remains unchanged by incorporating higher order effects.

CARMA(p,q) models (higher order models)

- The approach can be generalized to any order p, any order q and can be easily extended from univariate to multivariate systems (possibly with a measurement model).

- The math gets a bit tedious, though. For details, see Oud et al. (2018). Switching gears, we can – for example – define a CARMA(2,1) model as:

$$\frac{\mathrm{d}\boldsymbol{\eta}(t)}{\mathrm{d}t} = \mathbf{A}\boldsymbol{\eta}(t) + \mathbf{0} + \mathbf{0} + \mathbf{G}\frac{\mathrm{d}\mathbf{W}(t)}{\mathrm{d}t}$$

with $\boldsymbol{\eta} = \begin{pmatrix} \boldsymbol{\eta}_1 \\ \frac{\mathrm{d}\boldsymbol{\eta}_1}{\mathrm{d}t} \end{pmatrix}$ $\mathbf{A} = \begin{pmatrix} 0 & 1 \\ \mathbf{A}_0 & \mathbf{A}_1 \end{pmatrix}$ $\mathbf{G} = \begin{pmatrix} 0 & 0 \\ \mathbf{G}_0 & \mathbf{G}_1 \end{pmatrix}$

yields $$\begin{pmatrix} \frac{\mathrm{d}\boldsymbol{\eta}_1}{\mathrm{d}t} \\ \frac{\mathrm{d}^2\boldsymbol{\eta}_1}{\mathrm{d}t^2} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \mathbf{A}_0 & \mathbf{A}_1 \end{pmatrix} \begin{pmatrix} \boldsymbol{\eta}_1 \\ \frac{\mathrm{d}\boldsymbol{\eta}_1}{\mathrm{d}t} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \mathbf{G}_0 & \mathbf{G}_1 \end{pmatrix} \begin{pmatrix} \frac{\mathrm{d}\mathbf{W}}{\mathrm{d}t} \\ \frac{\mathrm{d}^2\mathbf{W}}{\mathrm{d}t^2} \end{pmatrix}$$

CARMA(p,q) models (higher order models)

Thus, the CARMA(2,1) model is

$$\frac{\mathrm{d}^2\boldsymbol{\eta}_1(t)}{\mathrm{d}t^2} = \mathbf{A}_0\boldsymbol{\eta}_1(t) + \mathbf{A}_1\frac{\mathrm{d}\boldsymbol{\eta}_1(t)}{\mathrm{d}t} + \mathbf{G}_0\frac{\mathrm{d}\mathbf{W}(t)}{\mathrm{d}t} + \mathbf{G}_1\frac{\mathrm{d}^2\mathbf{W}(t)}{\mathrm{d}t^2}$$

CAR(1)

CAR(2)

MA(0)
(Diffusion)

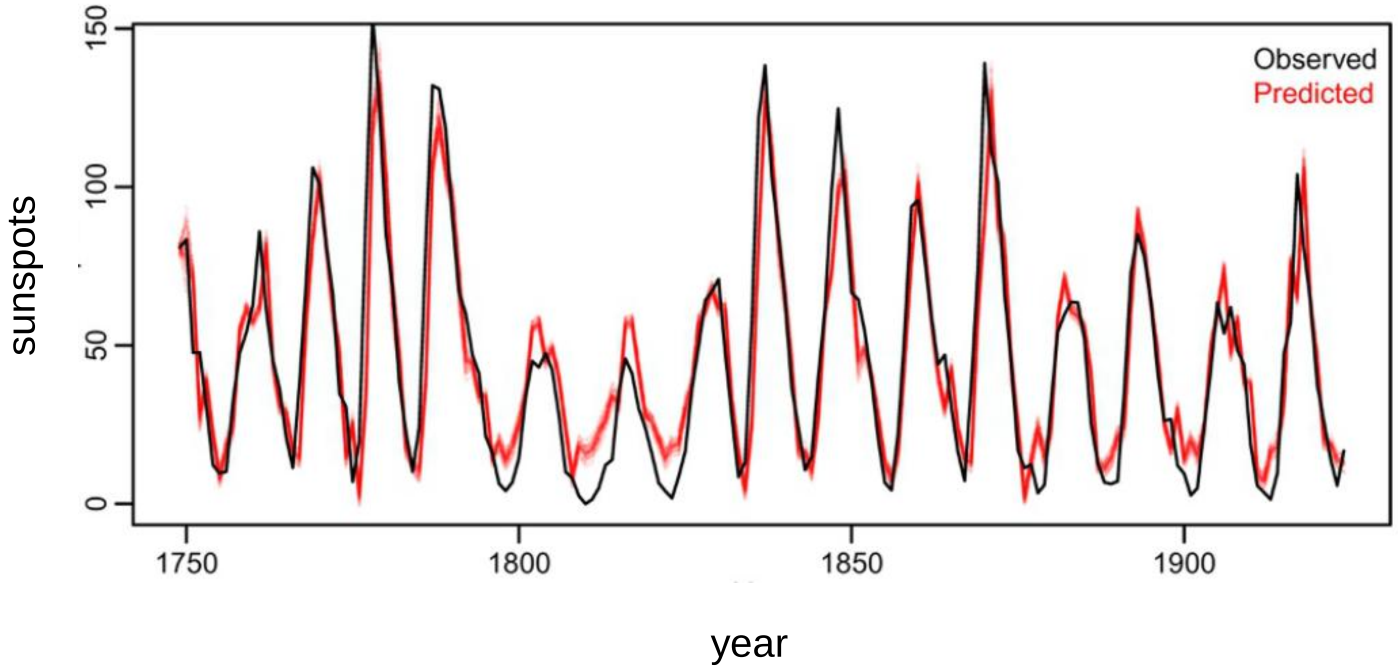
MA(1)

Note that this is only one possible parametrization. If there is no measurement model (as in this example) another way to include MA(q) effects is via the loading matrix.

CARMA(p, q) models

- ➢ From theory to practice

- ➢ A famous example of a real-world process that is well captured by a CARMA(2,1) model is the sunspot activity.



Oud, Voelkle, Driver (2018); Driver & Voelkle (2020)

CARMA(p, q) models (sunspot data)

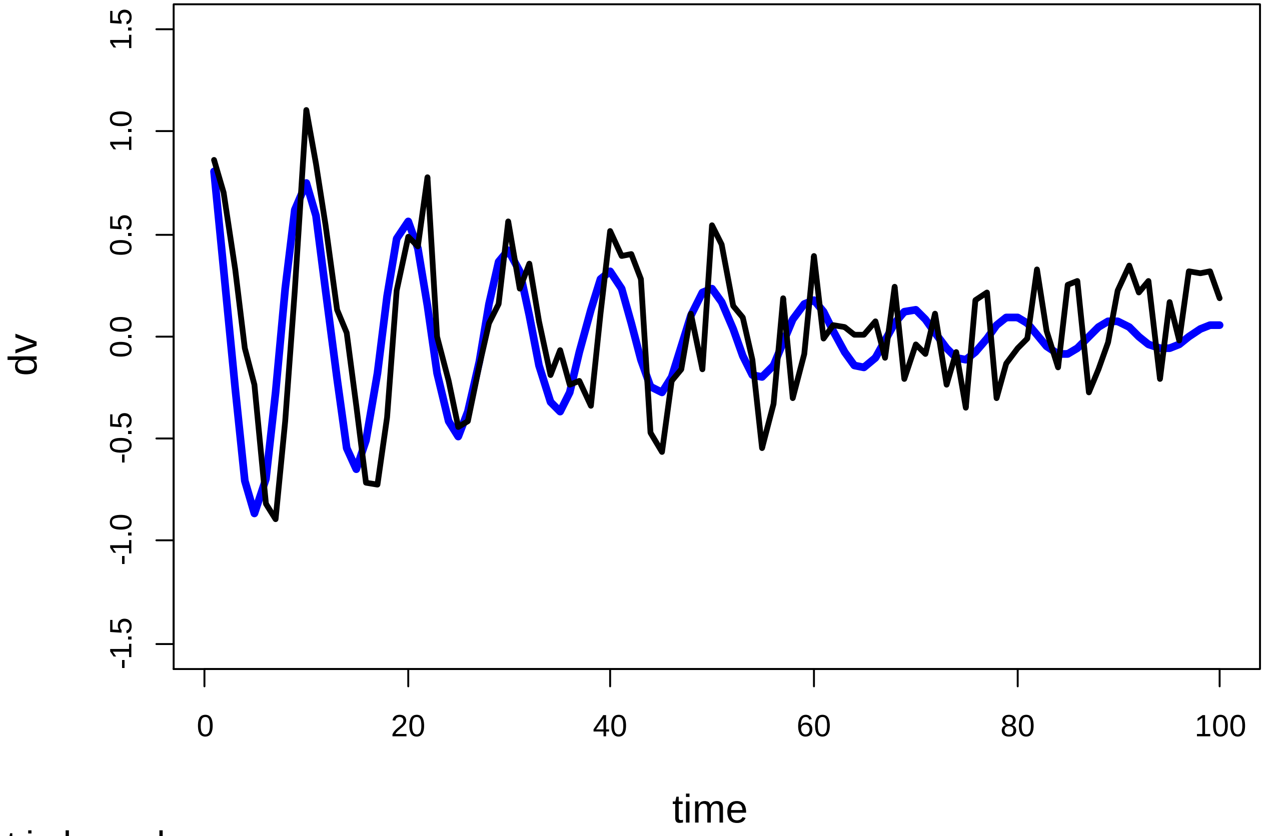
Let's try this:

```
library(ctsem)
sunspots<-sunspot.year #get data
sunspots<-sunspots[50: (length(sunspots) - (1988-1924))]
id <- 1
time <- 1749:1924
datalong <- cbind(id, time, sunspots)
#setup model
ssmodel <- ctModel(type='stanct', n.latent=2, n.manifest=1,
        manifestNames='sunspots',
        latentNames=c('ss_level', 'ss_velocity'),
        LAMBDA=matrix(c( 1, 'ma1' ), nrow=1, ncol=2),
        DRIFT=matrix(c(0, 'a21', 1, 'a22'), nrow=2, ncol=2),
        MANIFESTMEANS=matrix(c('m1'), nrow=1, ncol=1),
        CINT=matrix(c(0, 0), nrow=2, ncol=1),
        T0VAR=matrix(c(1,0,0,1), nrow=2, ncol=2), #Because single subject
        DIFFUSION=matrix(c(0, 0, 0, "diffusion"), ncol=2, nrow=2))
ssmodel$pars$indvarying<-FALSE #Because single subject
ssmodel$pars$transform[14]<- '(param)*5+44 ' #Because not mean centered
ssmodel$pars$transform[4]<-'log(exp(-param*1.5)+1)' #To avoid multimodality
ssfit <- ctStanFit(datalong, ssmodel, iter=1000, chains=4)
summary(ssfit, parmatrices = TRUE)
plot(ssfit)
```

Oscillations
(and a few words on the virtue of varying time intervals)

Damped linear oscillator

- As apparent from the sunspot example, continuous time models are able to capture rhythmic / oscillating patterns.
- This includes standard oscillations, such as damped and/or coupled linear oscillators.



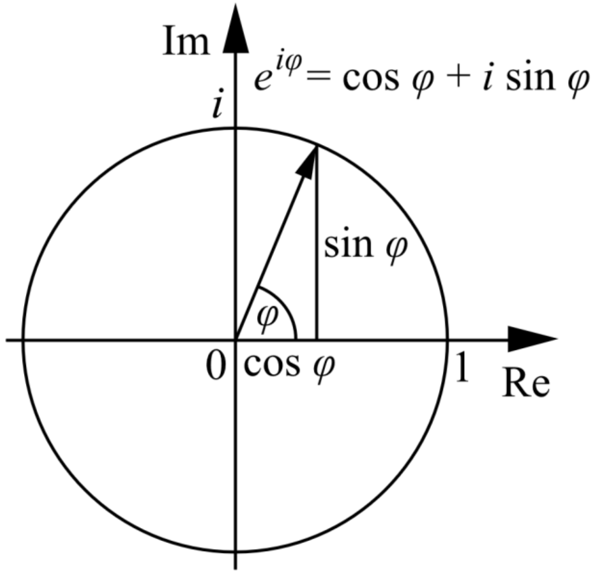
In large parts the following content is based on:
Voelkle, M. C., & Oud, J. H. L. (2013). Continuous time modelling with individually varying time intervals for oscillating and non-oscillating processes. *British Journal of Mathematical and Statistical Psychology*, 103-126. https://doi.org/10.1111/j.2044-8317.2012.02043.x

Damped linear oscillator

Technically, when the eigenvalues of the drift matrix are complex, the process is oscillating.

The eigenvalues of the drift matrix

$$\mathbf{A} = \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & -\gamma \end{pmatrix}$$



are

$$\lambda_{1,2} = -\frac{\gamma}{2} \pm \mathrm{i}\sqrt{-\frac{\gamma^2}{4} + \omega_0^2}$$ with $\mathrm{i} = \sqrt{-1}$ yielding

$$\mathrm{e}^{\mathbf{A}\cdot\Delta t_i} = \mathrm{e}^{-\frac{\gamma}{2}\Delta t_i} \times \begin{bmatrix} \frac{\gamma}{2\omega}\sin(\omega\Delta t_i) + \cos(\omega\Delta t_i) & \frac{1}{\omega}\sin(\omega\Delta t_i) \\ -\frac{\omega_0^2}{\omega}\sin(\omega\Delta t_i) & \cos(\omega\Delta t_i) - \frac{\gamma}{2\omega}\sin(\omega\Delta t_i) \end{bmatrix}$$

Damped linear oscillator

Just like before, the “trick” is to define a second order differential equation as a first order differential equation by expanding the state vector

$$\begin{pmatrix} \frac{d\eta(t)}{dt} \\ \frac{d^2\eta(t)}{dt^2} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & -\gamma \end{pmatrix} \begin{pmatrix} \eta(t) \\ \frac{d\eta(t)}{dt} \end{pmatrix} + \begin{pmatrix} 0 \\ b \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & g \end{pmatrix} \begin{pmatrix} \frac{dW_1(t)}{dt} \\ \frac{dW_2(t)}{dt} \end{pmatrix}$$
$$= \frac{d\eta(t)}{dt} = \mathbf{A}\eta(t) + \mathbf{b} + \mathbf{G}\frac{d\mathbf{W}(t)}{dt}.$$

with the first order derivative as a latent variable without an underlying manifest variable.

$$y(t) = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \eta(t) \\ \frac{d\eta(t)}{dt} \end{pmatrix}$$

Voelkle & Oud (2013)

Damped linear oscillator

Let’s try this:

```
library(ctsem)
data(Oscillating)

#interfacing to OpenMx
inits <- c(-39, -.3, 1.01, 10.01, .1, 10.01, 0.05, .9, 0)
names(inits) <- c("crosseffect","autoeffect", "diffusion",
        "T0var11", "T0var21", "T0var22","m1", "m2", 'manifestmean')

oscillatingm <- ctModel(n.latent = 2, n.manifest = 1, Tpoints = 11,
            MANIFESTVAR = matrix(c(0), nrow = 1, ncol = 1),
            LAMBDA = matrix(c(1, 0), nrow = 1, ncol = 2),
            T0MEANS = matrix(c('m1', 'm2'), nrow = 2, ncol = 1),
            T0VAR = matrix(c("T0var11", "T0var21", 0, "T0var22"), nrow = 2, ncol = 2),
            DRIFT = matrix(c(0, "crosseffect", 1, "autoeffect"), nrow = 2, ncol = 2),
            CINT = matrix(0, ncol = 1, nrow = 2),
            MANIFESTMEANS = matrix('manifestmean', nrow = 1, ncol = 1),
            DIFFUSION = matrix(c(0, 0, 0, "diffusion"), nrow = 2, ncol = 2),
            startValues=inits, type="omx")

oscillating_fit <- ctFit(Oscillating, oscillatingm)
summary(oscillating_fit, verbose=T)
plot(oscillating_fit)
```

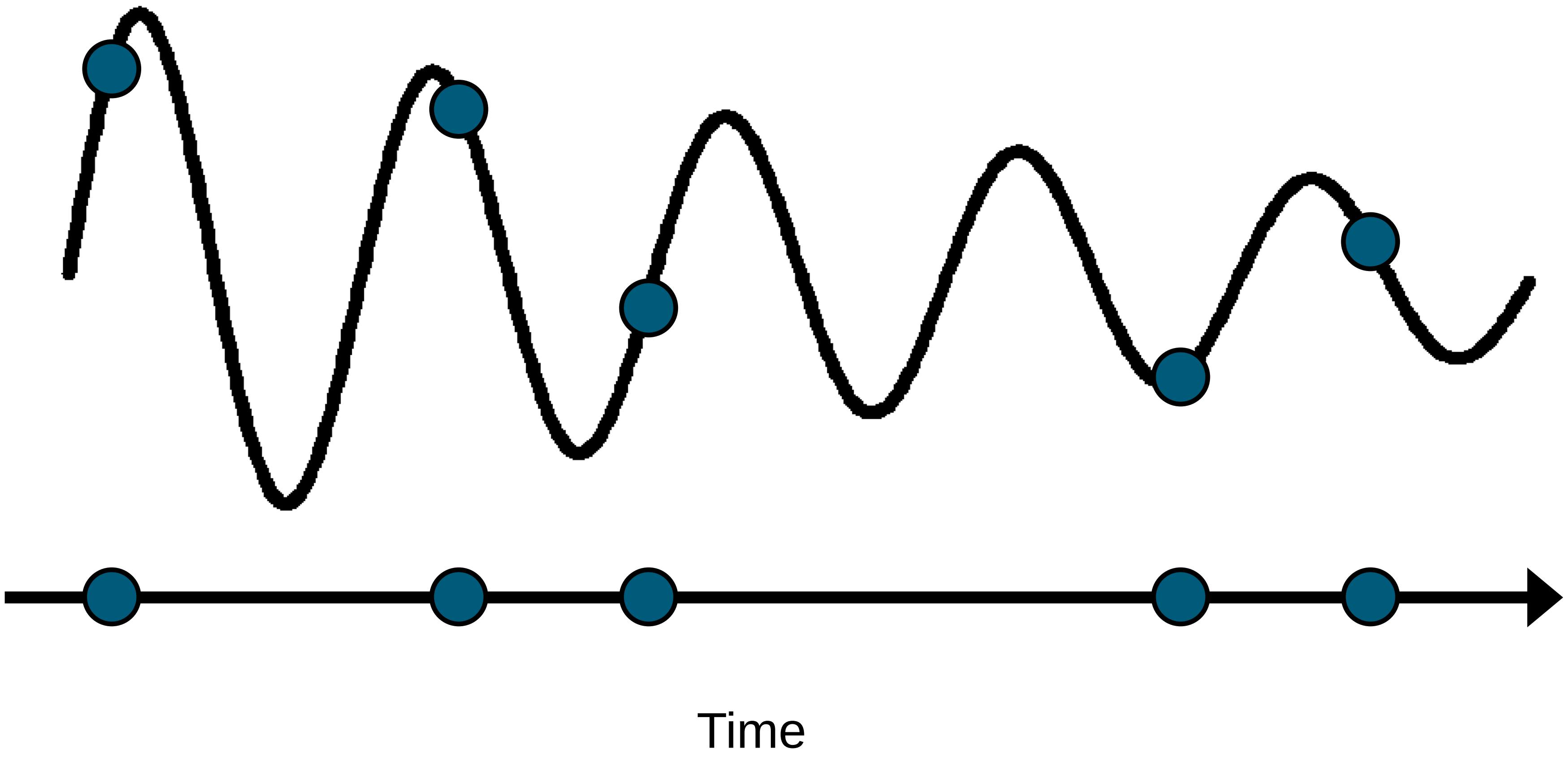
Damped linear oscillator

Let's try this:

```
#interfacing to Stan (long-data format)
oscillationlong_intervals <- ctWideToLong(datawide = Oscillating, Tpoints=11, n.manifest=1) #convert wide to long format
oscillationlong <- ctDeintervalise(datalong = oscillationlong_intervals, id='id', dT='dT')   #convert intervals to absolute time
#hist(oscillationlong[,2], ylab = "frequency", xlab="interval length", main="", col="gray")


oscillatingmodel_stan <-ctModel(n.latent = 2, n.manifest=1, Tpoints=11,
                    MANIFESTVAR=matrix(c(0), nrow=1, ncol=1),
                    LAMBDA=matrix(c(1, 0), nrow=1, ncol=2),
                    DRIFT=matrix(c(0, "cross", 1, "auto"), nrow=2, ncol=2),
                    CINT=matrix(c(0,0), ncol=1, nrow=2, ),
                    DIFFUSION=matrix(c(0, 0, 0, "diffusion22"), nrow=2, ncol=2),
                    #startValues = inits,
                    type="stanct")
oscillatingmodel_stan$pars$indvarying <- FALSE # no individual differences
oscillatingfit_stan <-ctStanFit(oscillationlong, oscillatingmodel_stan)
summary(oscillatingfit_stan, verbose=T)
plot(oscillatingfit_stan)
```

Damped linear oscillator

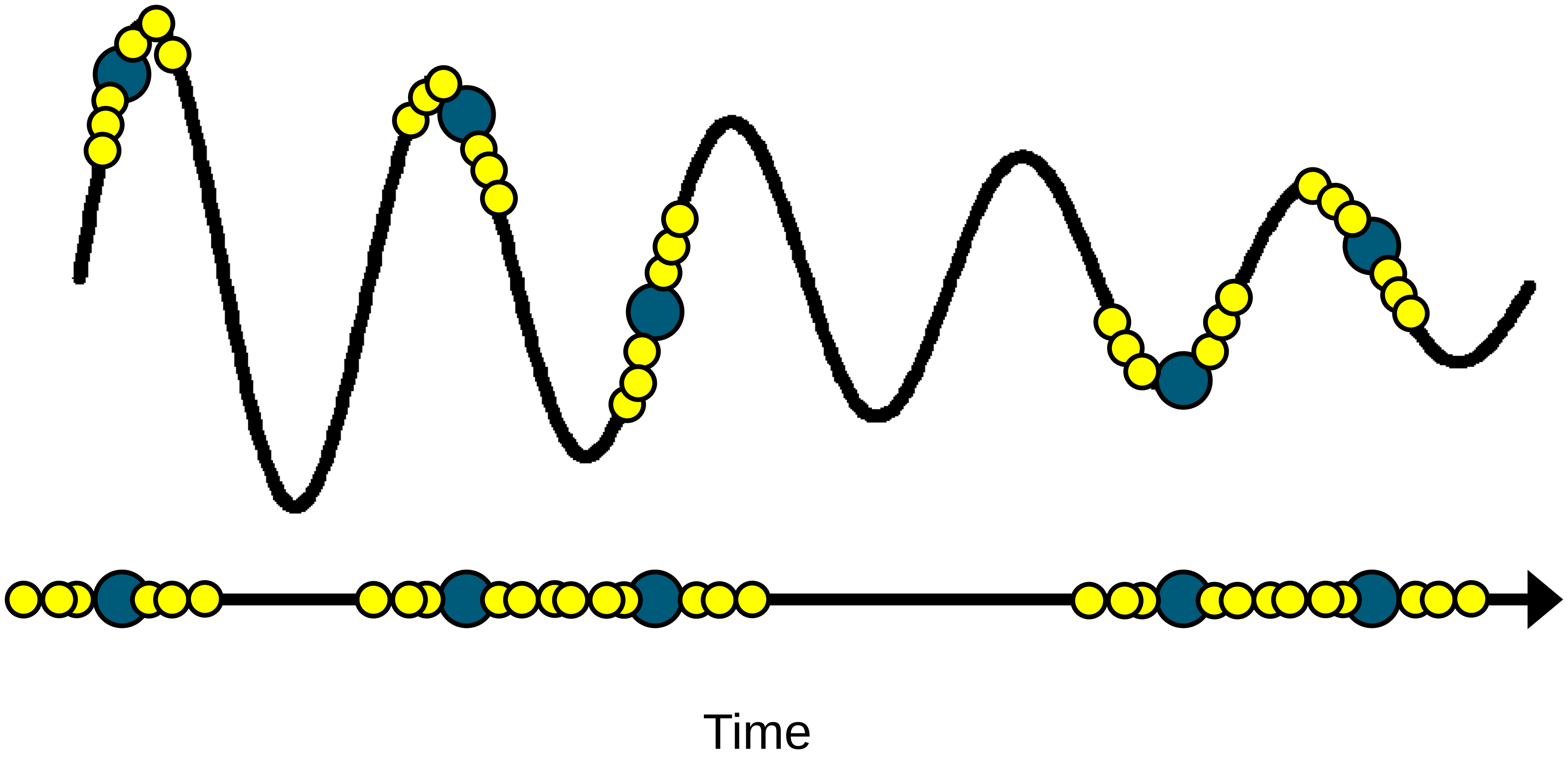
Unequal time intervals may help to detect oscillations with high(er) frequency, even if the sampling in time is low(er).



Voelkle & Oud (2013)

Damped linear oscillator

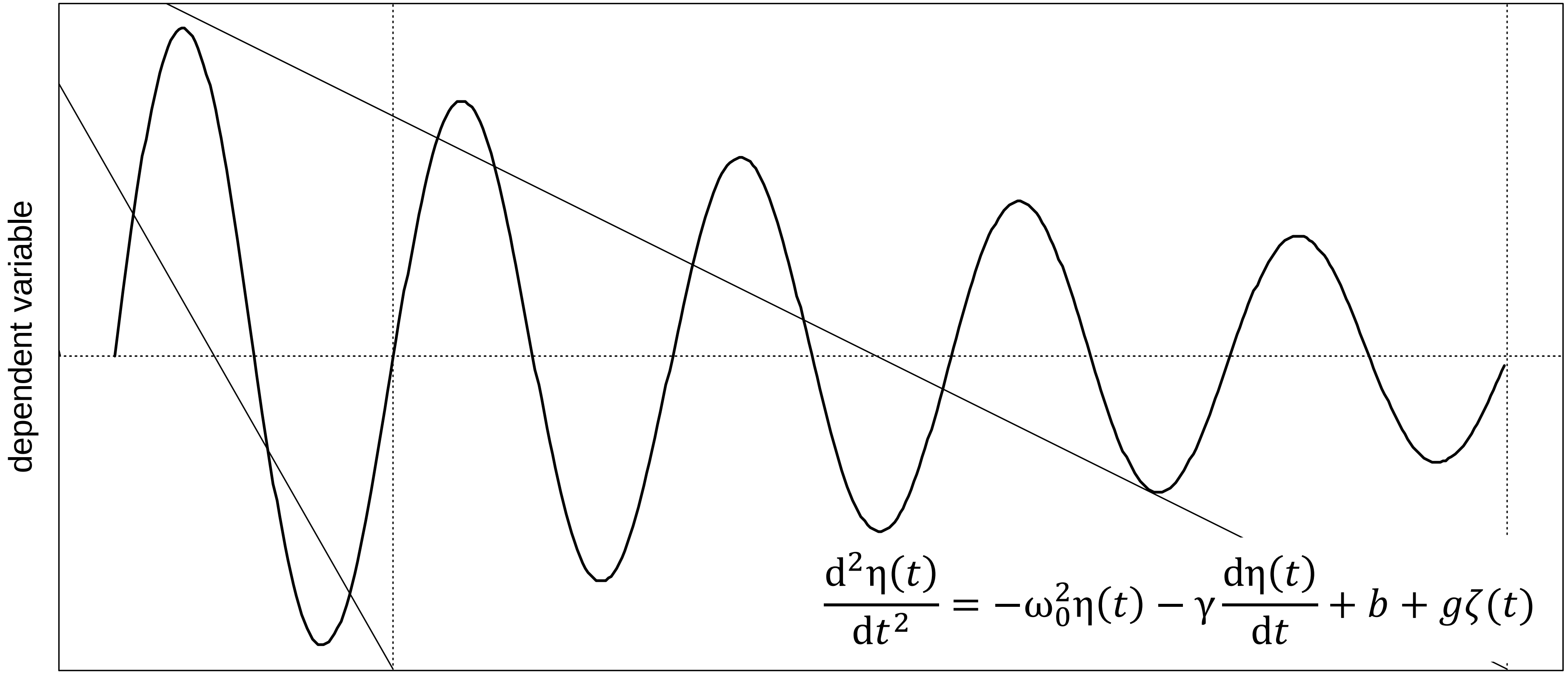
Unequal time intervals may help to detect oscillations with high(er) frequency, even if the sampling in time is low(er).



Voelkle & Oud (2013)

Damped linear oscillator

To support this claim, consider the following situation: $\omega_0^2 = 39.48\ [(2\pi)^2]$ and $\gamma = 0.5$:

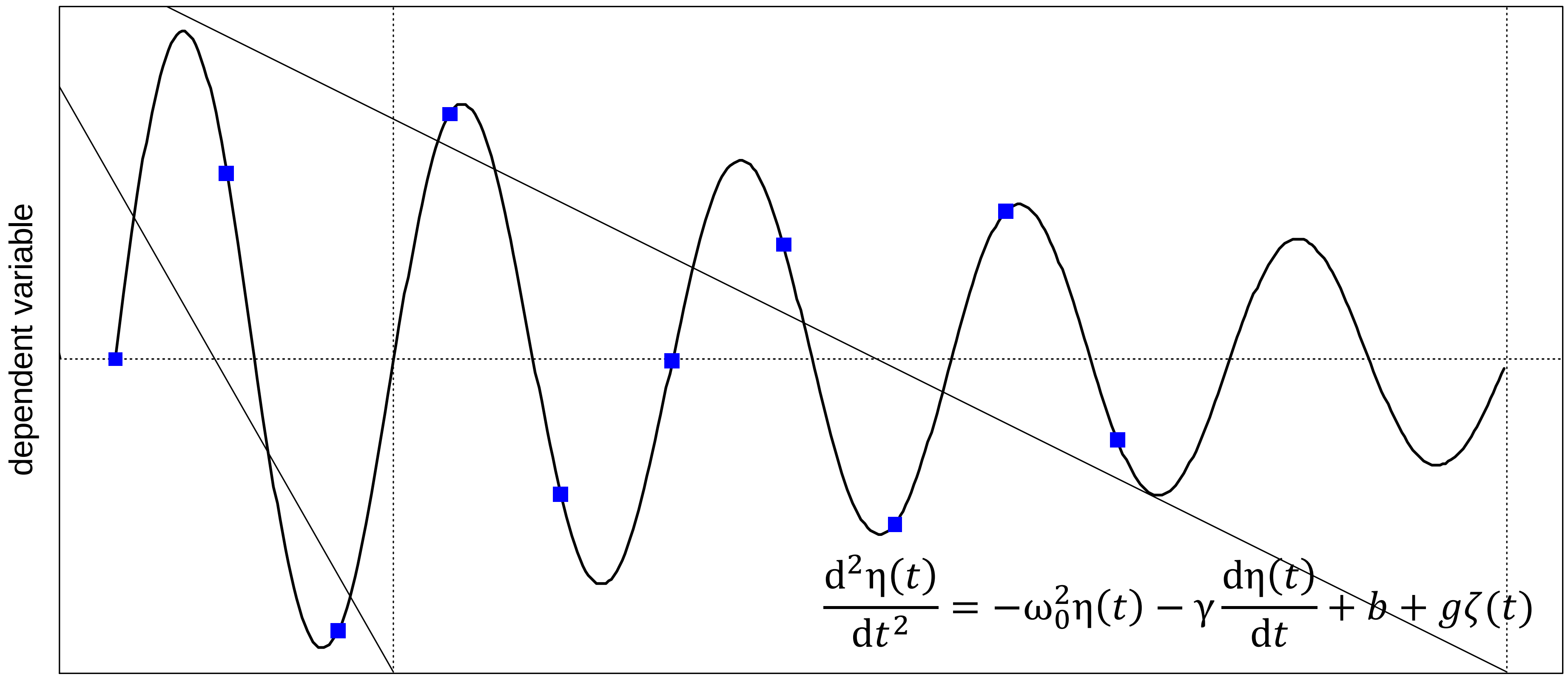


$$\frac{\mathrm{d}^2\eta(t)}{\mathrm{d}t^2} = -\omega_0^2\eta(t) - \gamma\frac{\mathrm{d}\eta(t)}{\mathrm{d}t} + b + g\zeta(t)$$

Voelkle & Oud (2013)

Damped linear oscillator

Generate data for N = 200 and T = 11 with Δt = 0.4:



Voelkle & Oud (2013)

Damped linear oscillator

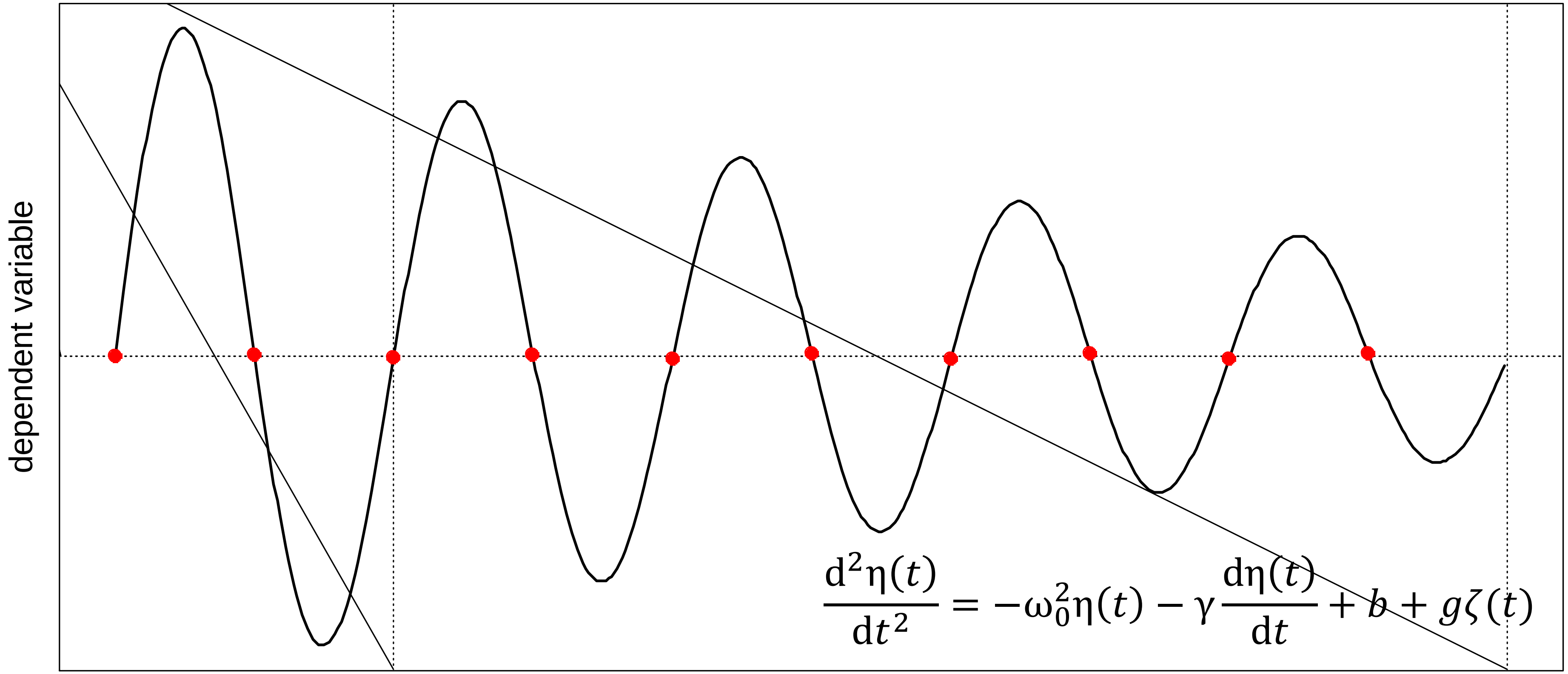
Fixed intervals:

Δt	0.4
Valid solutions	100%
Mean(ω_0^2)	39.521
SD(ω_0^2)	0.088
Mean(γ)	0.501
SD(γ)	0.012

Voelkle & Oud (2013)

Damped linear oscillator

Generate data for N = 200 and T = 11 with Δt = 0.5:



$$\frac{d^2\eta(t)}{dt^2} = -\omega_0^2\eta(t) - \gamma\frac{d\eta(t)}{dt} + b + g\zeta(t)$$

Voelkle & Oud (2013)

Damped linear oscillator

Fixed intervals:

Δt	0.4	0.5
Valid solutions	100%	20% [a)]
Mean(ω_0^2)	39.521	39.43
SD(ω_0^2)	0.088	0.389
Mean(γ)	0.501	0.517
SD(γ)	0.012	0.068

Voelkle & Oud (2013)

Damped linear oscillator

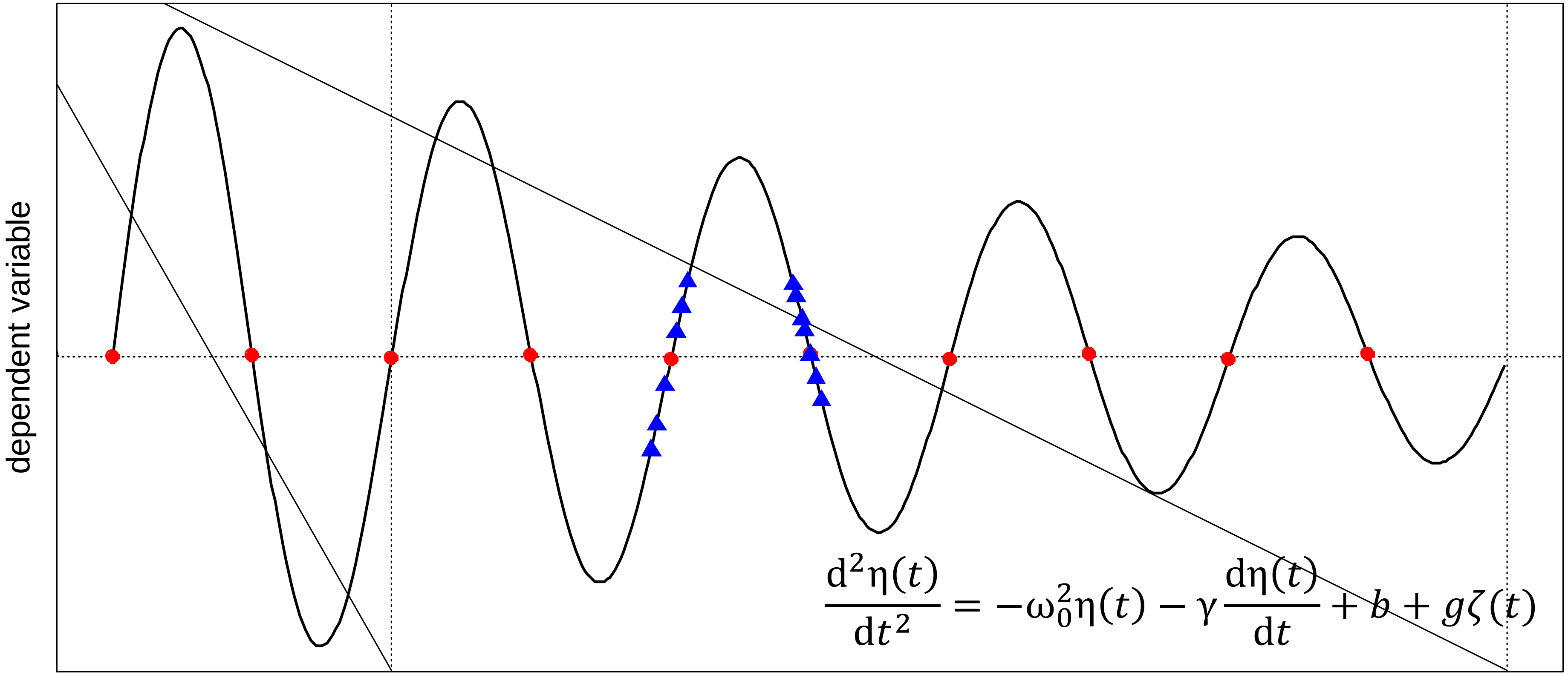
Fixed intervals:

Δt	0.4	0.5	1.0	1.0 no dampening	1.0 inferior SV
			NS-criterion not met		
Valid solutions	100%	20% [a)]	23% [b)]	19% [c)]	49% [d)]
Mean(ω_0^2)	39.521	39.43	39.433	39.787	230.14
$SD(\omega_0^2)$	0.088	0.389	0.409	0.292	630.547
Mean(γ)	0.501	0.517	0.511	0	886.22
$SD(\gamma)$	0.012	0.068	0.04	0.024	2381.932

Voelkle & Oud (2013)

Damped linear oscillator

Generate data for N = 200 and T = 11 with Δt_{ij} ~ N(0.4 or 0.5 or 1.0; 1/12):

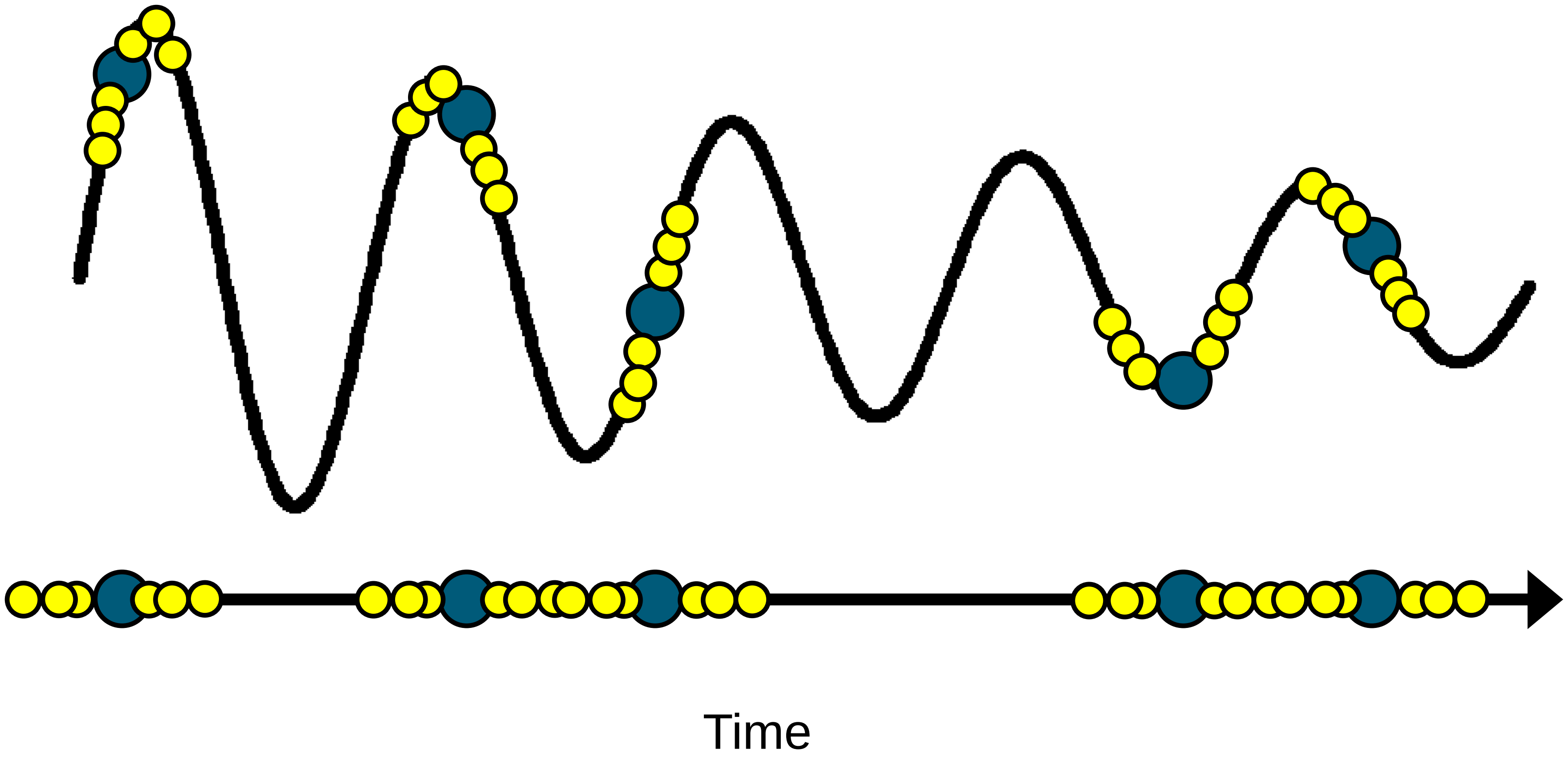


$$\frac{d^2\eta(t)}{dt^2} = -\omega_0^2\eta(t) - \gamma\frac{d\eta(t)}{dt} + b + g\zeta(t)$$

Voelkle & Oud (2013)

Damped linear oscillator

Individually varying intervals:



Voelkle & Oud (2013)

Damped linear oscillator

Individually varying intervals:

Δt	$N\left(0.4, \frac{1}{12}\right)$
Valid solutions	100%
Mean(ω_0^2)	39.529
$SD(\omega_0^2)$	0.091
Mean(γ)	0.501
$SD(\gamma)$	0.013

Voelkle & Oud (2013)

Damped linear oscillator

Individually varying intervals:

Δt	$N\left(0.4, \frac{1}{12}\right)$	$N\left(0.5, \frac{1}{12}\right)$
Valid solutions	100%	100%
Mean(ω_0^2)	39.529	39.558
$SD(\omega_0^2)$	0.091	0.097
Mean(γ)	0.501	0.502
$SD(\gamma)$	0.013	0.012

Voelkle & Oud (2013)

Damped linear oscillator

Individually varying intervals:

Δt	$N\left(0.4, \frac{1}{12}\right)$	$N\left(0.5, \frac{1}{12}\right)$	$N\left(1, \frac{1}{2}\right)$	$N\left(1, \frac{1}{2}\right)$ no dampening	$N\left(1, \frac{1}{2}\right)$ inferior SV
			NS-criterion not met		
Valid solutions	100%	100%	99% [e)]	56% [f)]	7% [g)]
Mean(ω_0^2)	39.529	39.558	39.969	39.993	39.956
SD(ω_0^2)	0.091	0.097	0.091	0.038	0.109
Mean(γ)	0.501	0.502	0.51	0	0.512
SD(γ)	0.013	0.012	0.012	0.006	0.011

Voelkle & Oud (2013)

Latent change score models

Latent Change Score models

- Latent Change Score (LCS) models are rather popular in some fields of psychology
- LCS are particularly useful for pre-post data (two wave) data
- For more waves, standard models, such as the

  - Proportional Change Score Model, or

  - Dual Change Score Model

  are (usually) just restricted approximations to true ct models
- In simple cases (of equal time intervals) ct and LCS models yield identical results, for more complex situations ct models are superior
- …what follows is a quick demonstration to support this argument.

In large parts the following content is based on:
Voelkle, M. C., & Oud, J. H. L. (2015). Relating latent change score and continuous time models. *Structural Equation Modeling: A Multidisciplinary Journal*, 22(3), 366-381. doi:10.1080/10705511.2014.935918

Latent Change Score models

- Latent change score model for *T* = 2

$$\Delta\eta(t_u - t_{u-1}) = \eta(t_u) - \eta(t_{u-1}) = \Delta\eta(\Delta_u)$$

- Advantage: Separation of true variance and error variance

- For $\Delta_u = t_u - t_{u-1} = 1$, the (latent) difference between two time points is equal to the rate of change

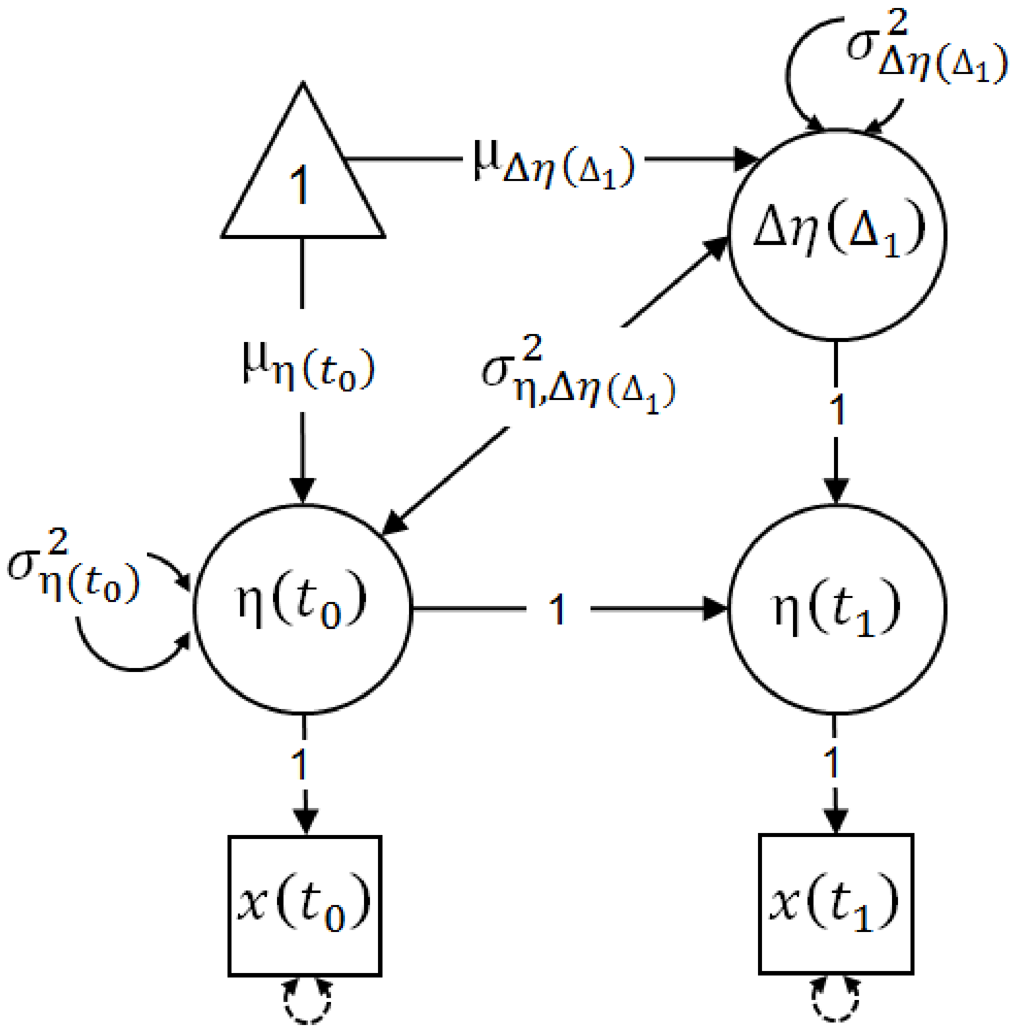
!

$$\frac{\Delta\eta(\Delta_u)}{\Delta_u}$$

- So that

$$\eta(t_u) = 1 \cdot \eta(t_{u-1}) + 1 \cdot \Delta\eta$$

Latent Change Score models



Latent Change Score models

- For $T > 2$ we can not only separate true from error variance but begin to impose a structure among the latent variables

- The proportional change score (PCS) model is one example

$$\Delta\eta = \beta \cdot \eta(t_{u-1})$$

- The dual change score (DCS) model adds an additional „slope“

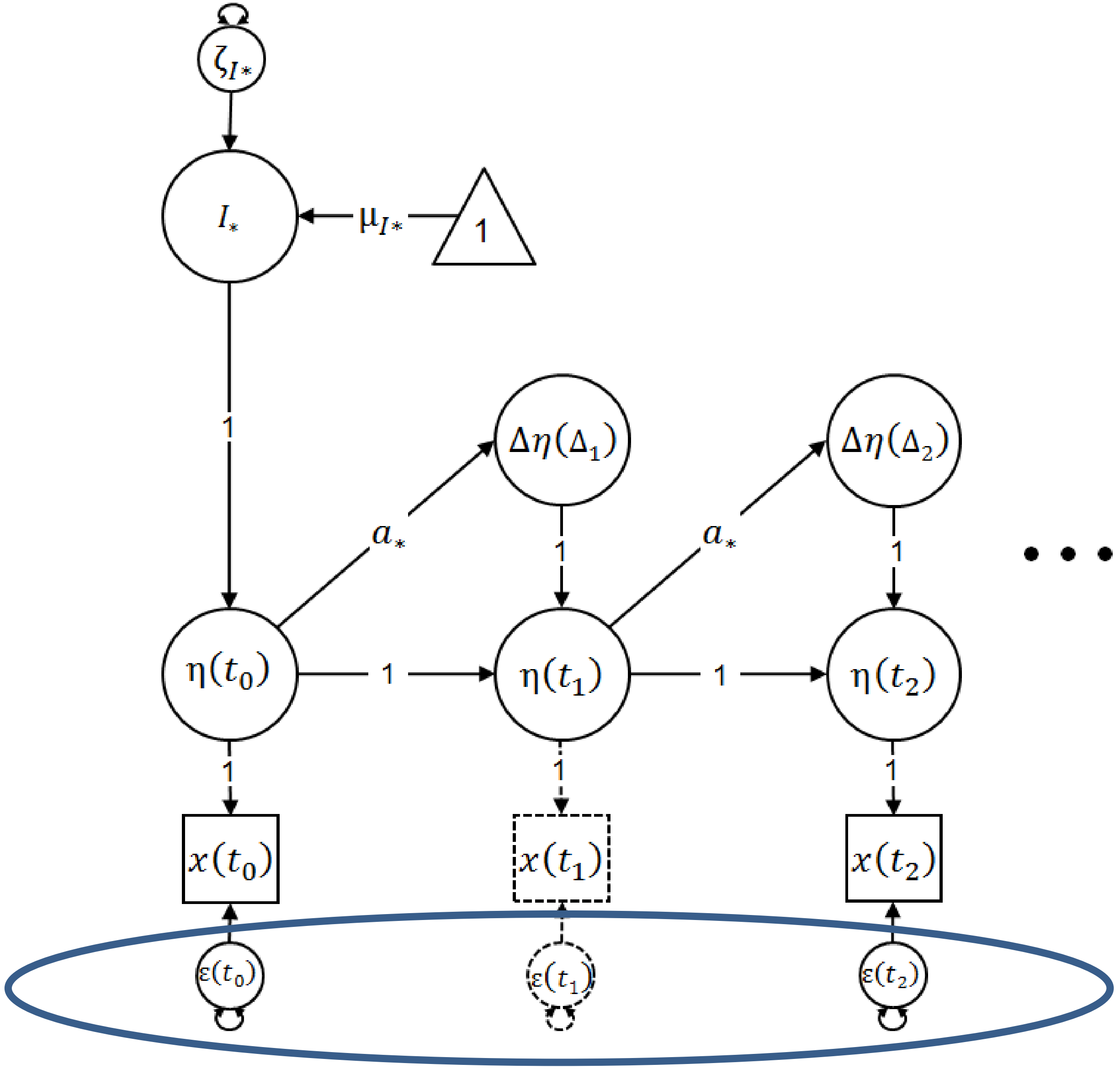
$$\Delta\eta = \beta \cdot \eta(t_{u-1}) + S_*$$

- Further distinction between measurement and dynamic error yields:

$$\Delta\boldsymbol{\eta} = \mathbf{A}_* \cdot \boldsymbol{\eta}(t_{u-1}) + \mathbf{S}_* + \boldsymbol{\zeta}_*$$

$\beta = a_*$

Latent Change Score models (PCS model)



Latent Change Score models (PCS model)

- Proportional change score model with measurement error but without dynamic error for $T = 5$

$$
\mathbf{A} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

Latent Change Score models (PCS model)

- Proportional change score model with measurement error but without dynamic error for T = 5

$$
\mathbf{S} = \begin{bmatrix}
\sigma_{I_*}^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\varepsilon^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\varepsilon^2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\varepsilon^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\varepsilon^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\varepsilon^2
\end{bmatrix}
$$

Latent Change Score models (PCS model)

- Proportional change score model with measurement error but without dynamic error for T = 5

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{M}' = \begin{bmatrix} \mu_{I_*} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Latent Change Score models (PCS model)

- Proportional change score model with measurement error but without dynamic error for $T = 5$

R script (with OpenMx and Lavaan Code):

1_PCS_model_OpenMx_Lavaan.R

Data:

1_data_PCSmodel.dat

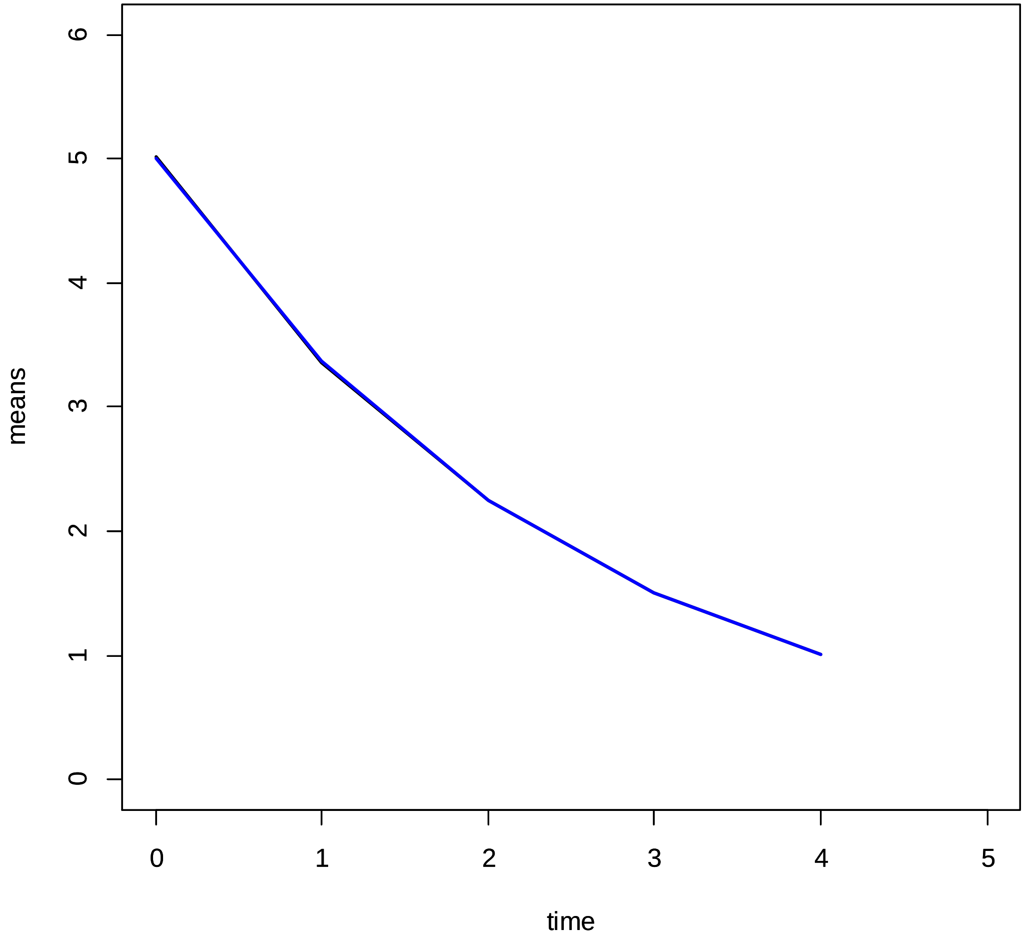
Latent Change Score models (PCS model)

- Proportional change score model with measurement error but without dynamic error for $T = 5$

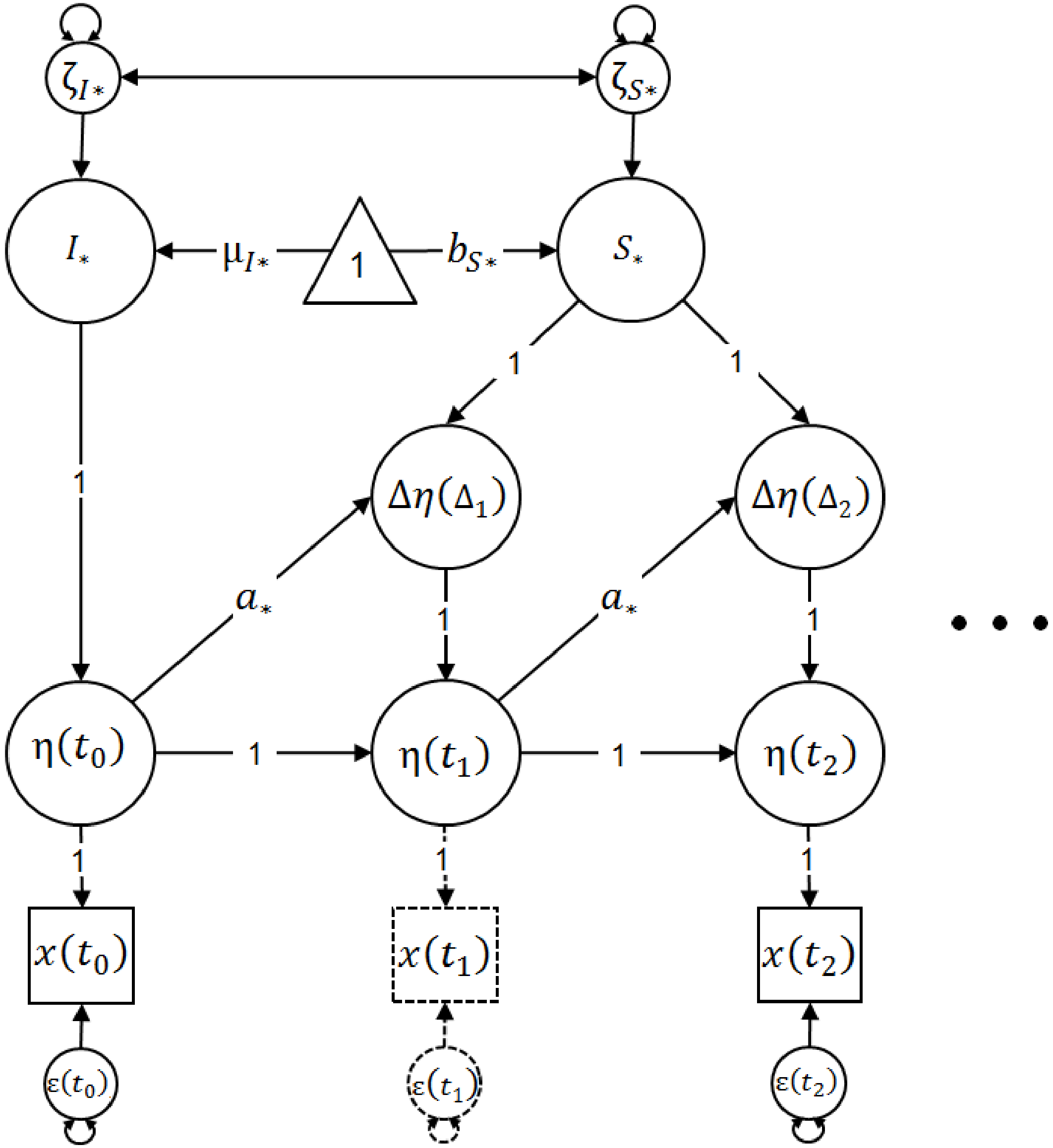
```
free parameters:
    name matrix row col   Estimate   Std.Error A
1      b      A   3   2 -0.3304290 0.001382494
2   var1      S   1   1  1.0028719 0.049993041
3 var(e)      S  11  11  0.1985027 0.004438655
4  meta1      M   1   I  5.0146682 0.034204305

Model Statistics:
               |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
       Model:             4                   4996             8406.41
   Saturated:            20                   4980                  NA
Independence:            10                   4990                  NA
Number of observations/statistics: 1000/5000

Information Criteria:
      |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
AIC:       -1585.59              8414.410                 8414.450
BIC:      -26104.74              8434.041                 8421.337
```

Latent Change Score models (PCS model)

- Proportional change score model with measurement error but without dynamic error for $T = 5$



Latent Change Score models (DCS model)



Latent Change Score models (DCS model)

- Dual change score model with measurement error but without dynamic error for T = 5

$$
\mathbf{A} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

Latent Change Score models (DCS model)

- Dual change score model with measurement error but without dynamic error for T = 5

$$\mathbf{S} = \begin{bmatrix} \sigma^2_{S_*} & \sigma_{S_*,I_*} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sigma_{S_*,I_*} & \sigma^2_{I_*} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{\varepsilon} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{\varepsilon} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{\varepsilon} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{\varepsilon} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^2_{\varepsilon} \end{bmatrix}$$

Latent Change Score models (DCS model)

- Dual change score model with measurement error but without dynamic error for *T* = 5

$$\mathbf{F} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{M}' = [\mu_{S_*} \quad \mu_{I_*} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

Latent Change Score models (DCS model)

- Dual change score model with measurement error but without dynamic error for $T = 5$

```
R script (with OpenMx and Lavaan Code):

          2_DCS_model_OpenMx_Lavaan.R

Data:

              2_data_DCSmodel.dat
```

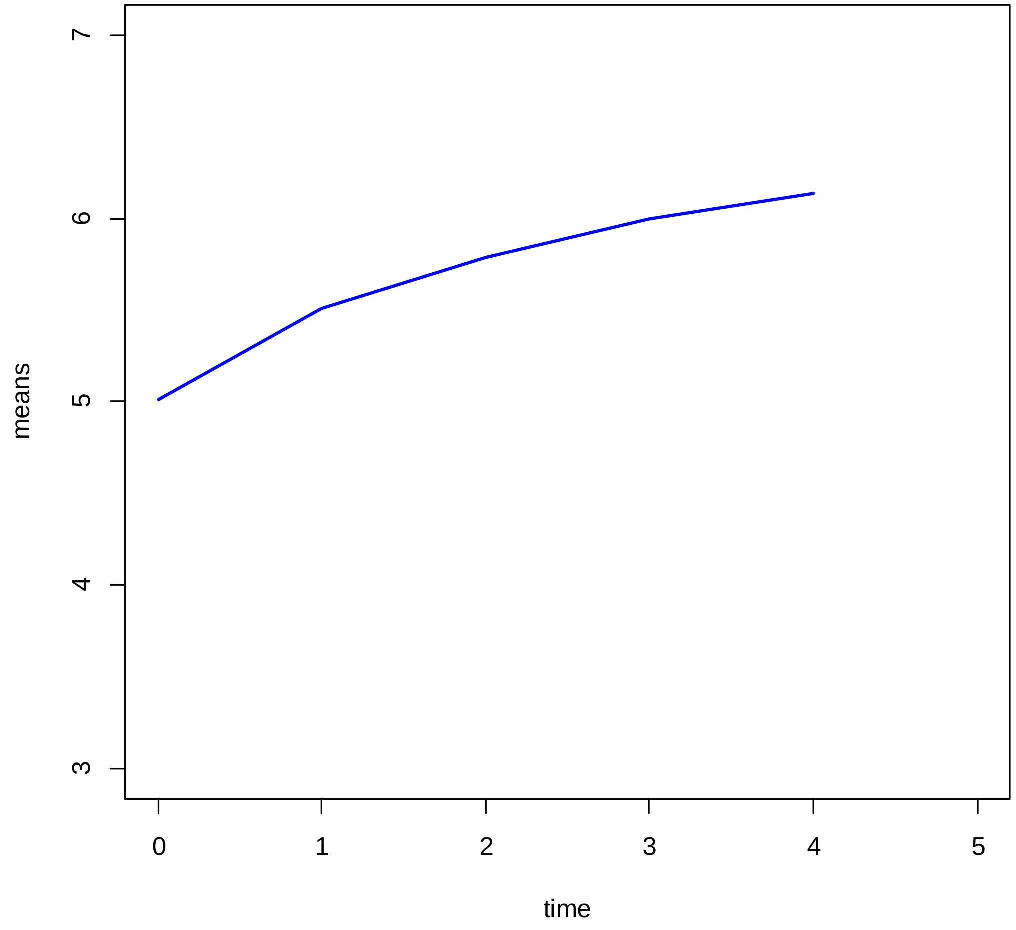
Latent Change Score models (DCS model)

- Dual change score model with measurement error but without dynamic error for T = 5

```
free parameters:
     name matrix row col    Estimate   Std.Error A
1       b      A   4   3 -0.33426495 0.006656825
2  var(S)      S   1   1  1.73772734 0.083646333
3 cov(1S)      S   2   1  0.05610959 0.043449833
4    var1      S   2   2  0.91148597 0.047891566
5  var(e)      S  12  12  0.20355572 0.005255795
6      mS      M   1   S  2.14360572 0.056047387
7   meta1      M   1   I  5.01742658 0.032716777

Model Statistics:
               |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
       Model:             7                   4993             12957.27
   Saturated:            20                   4980                   NA
Independence:            10                   4990                   NA
Number of observations/statistics: 1000/5000

Information Criteria:
      |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
AIC:      2971.272              12971.27               12971.38
BIC:    -21533.150              13005.63               12983.39
```

Latent Change Score models (DCS model)

- Dual change score model with measurement error but without dynamic error for T = 5



Latent Change Score models (DCS model)

- Dual change score model with measurement error **and** dynamic error for *T* = 5

```
R script (with OpenMx and Lavaan Code):

         3_DCS_model_OpenMx_Lavaan.R

Data:

              3_data_DCSmodel.dat
```

Latent Change Score models (DCS model)

- Dual change score model with measurement error **and** dynamic error for T = 5

```
free parameters:
     name matrix row col    Estimate  Std.Error A
1       b      A   4   3 -0.31873809 0.02453309
2  var(S)      S   1   1  1.68148341 0.16702023
3 cov(1S)      S   2   1  0.06780329 0.05194576
4    var1      S   2   2  0.93882555 0.08146233
5  var(z)      S   4   4  2.01040764 0.12547098
6  var(e)      S  12  12  0.20174170 0.06882644
7      mS      M   1   S  2.02171459 0.14367425
8   meta1      M   1   I  5.01101731 0.03371202

Model Statistics:
               |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
       Model:             8                   4992              19082.67
   Saturated:            20                   4980                    NA
Independence:            10                   4990                    NA
Number of observations/statistics: 1000/5000

Information Criteria:
      |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
AIC:       9098.666               19098.67                 19098.81
BIC:     -15400.848               19137.93                 19112.52
```

Relating LCS to CT Models

- Dual change score model reformulated as a latent AR model

For $\Delta\boldsymbol{\eta}_i(\Delta_{j,i}) = \boldsymbol{\eta}_i(t_{j,i}) - \boldsymbol{\eta}_i(t_{j-1,i})$, the dual change score model is:

$$\boldsymbol{\eta}_i(t_{j,i}) = (\mathbf{A}_* \cdot \Delta_{j,i} + \mathbf{I}) \cdot \boldsymbol{\eta}_i(t_{j-1,i}) + \Delta_{j,i} \cdot \mathbf{S}_{*i} + \boldsymbol{\zeta}_{*i} \ ,$$

which corresponds to a vector autoregressive and cross-lagged (ARCL) panel model of the general form

$$\boldsymbol{\eta}_i(t_{j,i}) = \mathbf{A}(\Delta_{j,i}) \cdot \boldsymbol{\eta}_i(t_{j-1,i}) + \mathbf{S}_i(\Delta_{j,i}) + \boldsymbol{\zeta}_i(\Delta_{j,i}) \ ,$$

with

$\mathbf{A}(\Delta_{j,i}) = (\mathbf{A}_* \cdot \Delta_{j,i} + \mathbf{I})$,

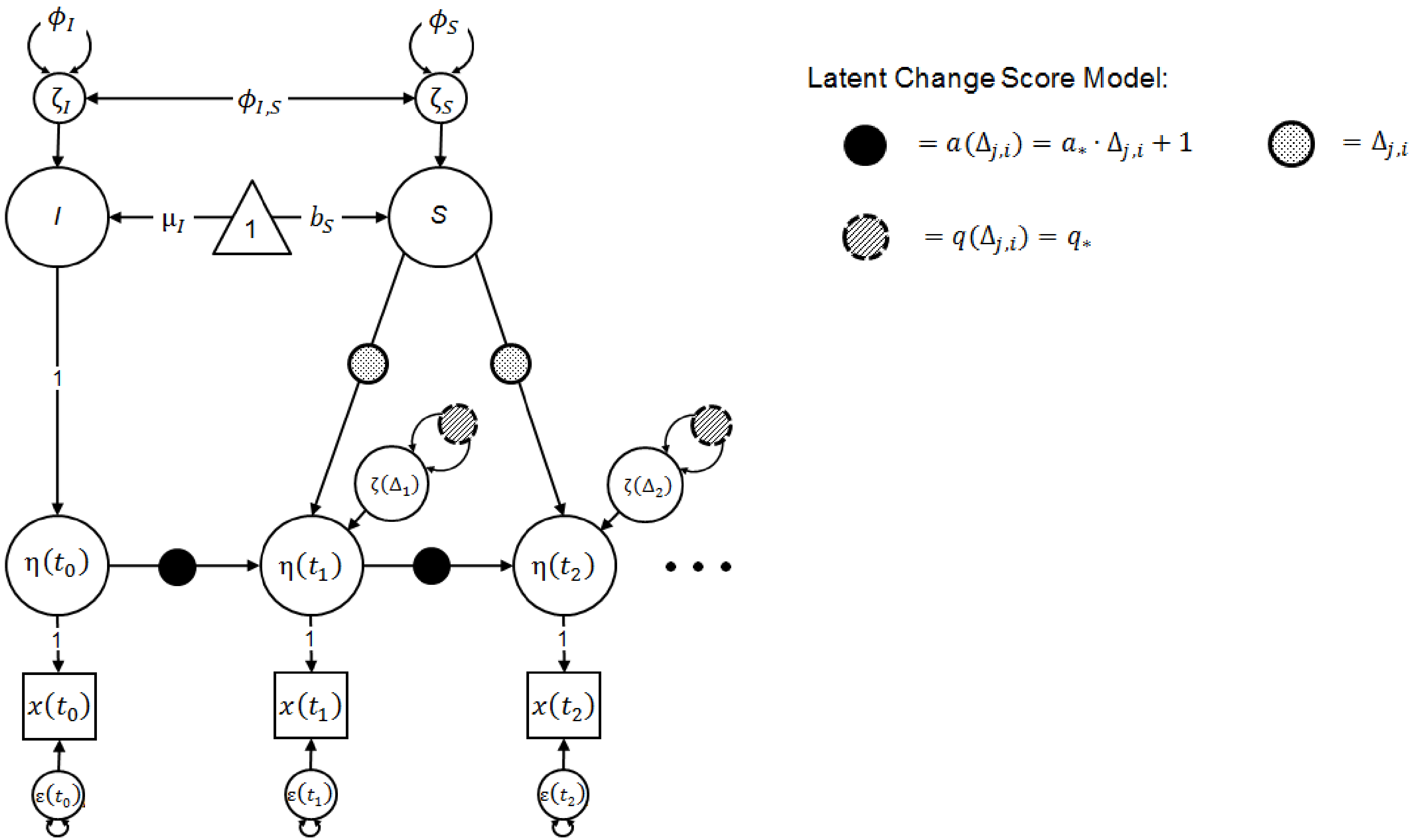
$\mathbf{S}_i(\Delta_{j,i}) = \Delta_{j,i} \cdot \mathbf{S}_{*i}$, and

$\boldsymbol{\zeta}_i(\Delta_{j,i}) = \boldsymbol{\zeta}_{*i}$.

$\beta = a_*; j = u$

Relating LCS to CT Models

- Dual change score model reformulated as a latent AR model



$\beta = a_*; j = u$

Relating LCS to CT Models

- We have already seen how to formulate a continuous time model as a latent AR model

$$\boldsymbol{\eta}_i(t_{j,i}) = \mathrm{e}^{\mathbf{A}\cdot\Delta_{j,i}}\boldsymbol{\eta}_i(t_{0,i}) + \mathbf{A}^{-1}[\mathrm{e}^{\mathbf{A}\cdot\Delta_{j,i}} - \mathbf{I}]\mathbf{S}_i + \int_{t_0}^{t_{j,i}} \mathrm{e}^{\mathbf{A}\cdot(t_{j,i}-s)}\mathbf{G}\mathrm{d}\mathbf{W}_i(s)$$

This corresponds to a vector autoregressive and cross-lagged panel model with

$\mathbf{A}(\Delta_{j,i}) = \mathrm{e}^{\mathbf{A}\cdot\Delta_{j,i}}$,

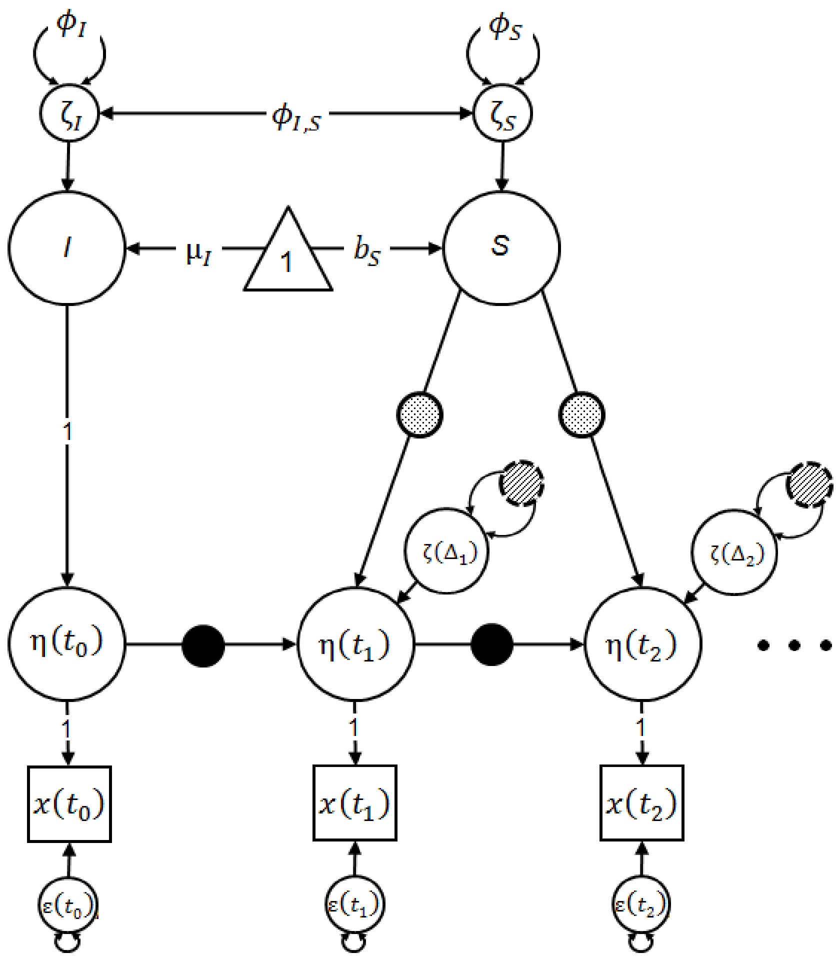
$\mathbf{S}_i(\Delta_{j,i}) = \mathbf{A}^{-1}[\mathrm{e}^{\mathbf{A}\cdot\Delta_{j,i}} - \mathbf{I}]\mathbf{S}_i$, and

$\boldsymbol{\zeta}_i(\Delta_{j,i}) = \int_{t_0}^{t_{j,i}} \mathrm{e}^{\mathbf{A}\cdot(t_{j,i}-s)}\mathbf{G}\mathrm{d}\mathbf{W}_i(s)$.

$\beta = a_*; j = u$

Relating LCS to CT Models

- Continuous time model reformulated as a latent AR model



Continuous Time Model:

$= a(\Delta_{j,i}) = e^{a\cdot\Delta_j}$

$= a^{-1}(e^{a\cdot\Delta_{j,i}} - 1)$

$= q(\Delta_{j,i}) = (a + a)^{-1}(e^{(a+a)\cdot\Delta_{j,i}} - 1) \cdot q$

$\beta = a_*; j = u$

Relating LCS to CT Models

- Thus, there is a direct relationship between latent change score and continuous time models:

Parameter	Latent Change Score Model	Continuous Time Model
$\mathbf{A}(\boldsymbol{\Delta}_{j,i})$	$= (\mathbf{A}_* \cdot \Delta_{j,i} + \mathbf{I})$	$= \mathrm{e}^{\mathbf{A} \cdot \Delta_{j,i}}$
$\mathbf{S}_i(\boldsymbol{\Delta}_{j,i})$	$= \Delta_{j,i} \cdot \mathbf{S}_{*i},$	$= \mathbf{A}^{-1}\left[\mathrm{e}^{\mathbf{A} \cdot \Delta_{j,i}} - \mathbf{I}\right]\mathbf{S}_i$
$\boldsymbol{\zeta}_i(\boldsymbol{\Delta}_{j,i})$	$= \boldsymbol{\zeta}_{*i}$	$= \int_{t_0}^{t_{j,i}} \mathrm{e}^{\mathbf{A} \cdot (t_{j,i} - s)} \mathbf{G} \mathrm{d}\mathbf{W}_i(s)$

$\beta = a_*; j = u$

Relating LCS to CT Models

- The relation between LCS and CT models is best understood when putting the matrix exponential constraint of the CT model in power series expansion:

$$\mathbf{A}(\Delta_{j,i}) = \mathrm{e}^{\mathbf{A}\cdot\Delta_{j,i}} = \mathbf{I} + \mathbf{A}\cdot\Delta_{j,i} + \frac{1}{2!}\mathbf{A}^2\cdot\Delta_{j,i}^2 + \frac{1}{3!}\mathbf{A}^3\cdot\Delta_{j,i}^3 \cdots$$

$\beta = a_*; j = u$

Relating LCS to CT Models

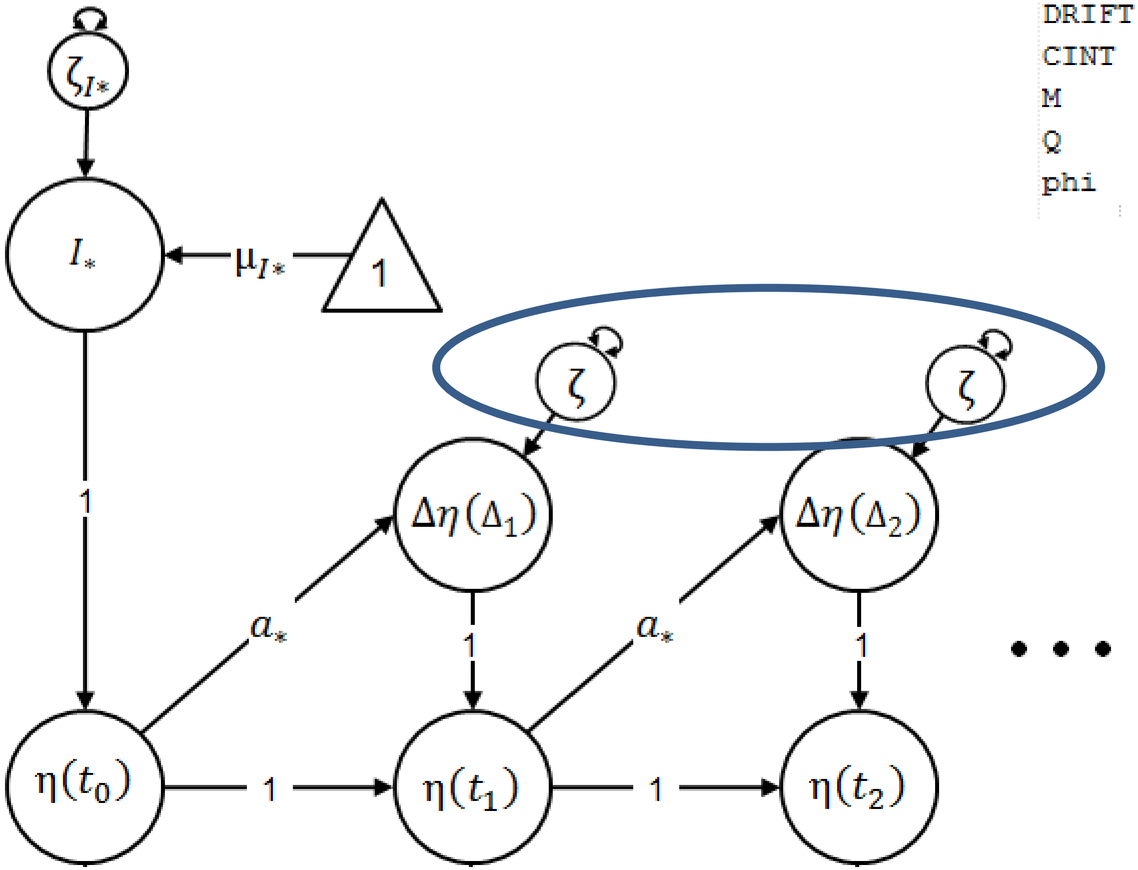
From theory to practice…

…let's have another example:

1. Generate data according to a known dynamic model with equal time intervals (T = 5)
2. Fit a latent change score model to the data
3. Replicate results using a CT model
4. Generate data according to a known dynamic model with individually varying time intervals
5. Fit a CT model to the data
6. Try to fit a latent change score model to the data

Relating LCS to CT Models

1. Generate data according to a known dynamic model with equal time intervals (T = 5)




```
N          <- 500
DRIFT      <- (matrix(c(-0.4), nrow= 1, ncol = 1, byrow = TRUE))
CINT       <- (matrix(c(0), nrow= 1, ncol = 1, byrow = TRUE))
M          <- (matrix(c(2), nrow= 1, ncol = 1, byrow = TRUE))
Q          <- matrix(c(1), nrow=1, ncol=1, byrow=TRUE)
phi        <- matrix(c(2), nrow=1, ncol=1, byrow=TRUE)
```

Relating LCS to CT Models

2. Fit a latent change score model to the data

```
R script (with OpenMx and Lavaan Code):

        4a_PCS_model_OpenMx_Lavaan.R

Data:

        4_sim_univariate_delta_1.dat
```

Relating LCS to CT Models

2. Fit a latent change score model to the data

```
free parameters:
    name matrix row col   Estimate  Std.Error A
1      b      A   3   2 -0.3368296 0.01020483
2   var1      S   1   1  1.9845720 0.12551366
3 var(z)      S   3   3  0.6729965 0.02128202
4  meta1      M   1   I  1.9915945 0.06300088

Model Statistics:
               |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
       Model:             4                   2496              6645.364
   Saturated:            20                   2480                    NA
Independence:            10                   2490                    NA
Number of observations/statistics: 500/2500

Information Criteria:
      |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
AIC:      1653.364               6653.364                 6653.445
BIC:     -8866.298               6670.222                 6657.526
```

Relating LCS to CT Models

3. Replicate results using a CT model

R script:

4b_CT_model_OpenMx.R

Data:

4_sim_univariate_delta_1.dat

Relating LCS to CT Models

3. Replicate results using a CT model

```
> summary(ct_model4b)$omxsummary$Minus2LogLikelihood
[1] 6645.364
> summary(ct_model4b)$DRIFT
           eta1
eta1 -0.4107233
> summary(ct_model4b)$DIFFUSION
          eta1
eta1 0.9868363
> summary(ct_model4b)$T0VAR
         eta1
eta1 1.984572
> summary(ct_model4b)$T0MEANS
         [,1]
eta1 1.991595
```

Remember the relationship between the continuous and discrete time parameters:

exp(-0.4107233) = 0.6631704 = 1-0.3368297 and

((a+a)**-1)*(exp(a+a)-1)*(g**2) = 0.6729962

Relating LCS to CT Models

4. Generate data according to a known dynamic model with individually varying time intervals.
   We use the same model as before.

$$\Delta_{u,i} \sim \mathrm{U}(0,4]$$

for all *u* and *i*.

Relating LCS to CT Models

5. Fit a CT model to the data

```
R script:

            5a_CT_model_OpenMx.R

Data:

      5_sim_univariate_deltavarying.dat
```

Relating LCS to CT Models

5. Fit a CT model to the data

```
> summary(ct_model5afit)$omxsummary$Minus2LogLikelihood
[1] 7543.671
> summary(ct_model5afit)$DRIFT
           eta1
eta1 -0.4083977
> summary(ct_model5afit)$DIFFUSION
         eta1
eta1 1.030807
> summary(ct_model5afit)$T0VAR
         eta1
eta1 2.078926
> summary(ct_model5afit)$T0MEANS
         [,1]
eta1 2.030593
```

Relating LCS to CT Models

6. Try to fit a latent change score model to the data

```
R script (with OpenMx and Lavaan Code):

        5b_PCS_model_OpenMx_Lavaan.R

Data:

      5_sim_univariate_deltavarying.dat
```

Relating LCS to CT Models

6. Try to fit a latent change score model to the data

```
free parameters:
    name matrix row col   Estimate  Std.Error A
1      b      A   3   2 -0.6226777 0.01420925
2   var1      S   1   1  2.0789260 0.13148419
3 var(z)      S   3   3  1.1104069 0.03511424
4  meta1      M   1   I  2.0305927 0.06448176

Model Statistics:
               |  Parameters  |  Degrees of Freedom  |  Fit (-2lnL units)
       Model:             4                   2496              7670.072
   Saturated:            20                   2480                    NA
Independence:            10                   2490                    NA
Number of observations/statistics: 500/2500

Information Criteria:
      |  df Penalty  |  Parameters Penalty  |  Sample-Size Adjusted
AIC:       2678.072               7678.072                 7678.152
BIC:      -7841.590               7694.930                 7682.234
```

Also:

exp(-0.4083978) = 0.6647144 IS NOT IDENTICAL TO

1-0.6226777 = 0.3773223

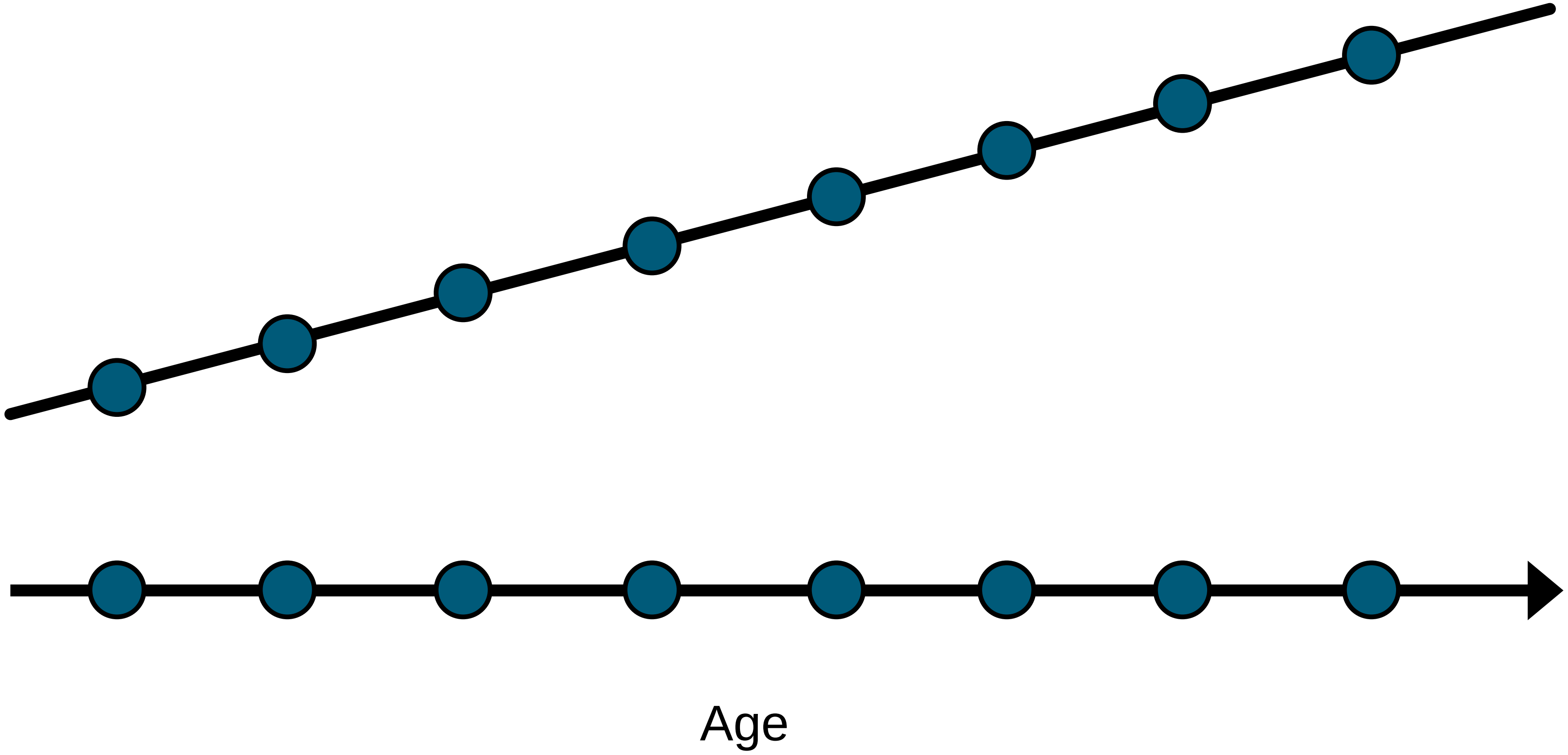
Summary Latent Change Score models

- As dynamic models, LCS models are powerful models for the analysis of change.
- For Δt = 1, the rate of change equals the latent difference.
- LCS models can be reformulated as (multivariate) autoregressive models.
- CT models manifest themselves as autoregressive models in discrete time.
- LCS models are special cases of the more general class of CT models. In case of Δt = 1, discrete time parameter estimates of CT and LCS models are “equivalent”. For unequal Δt, CT models should be used.

Accelerated longitudinal designs

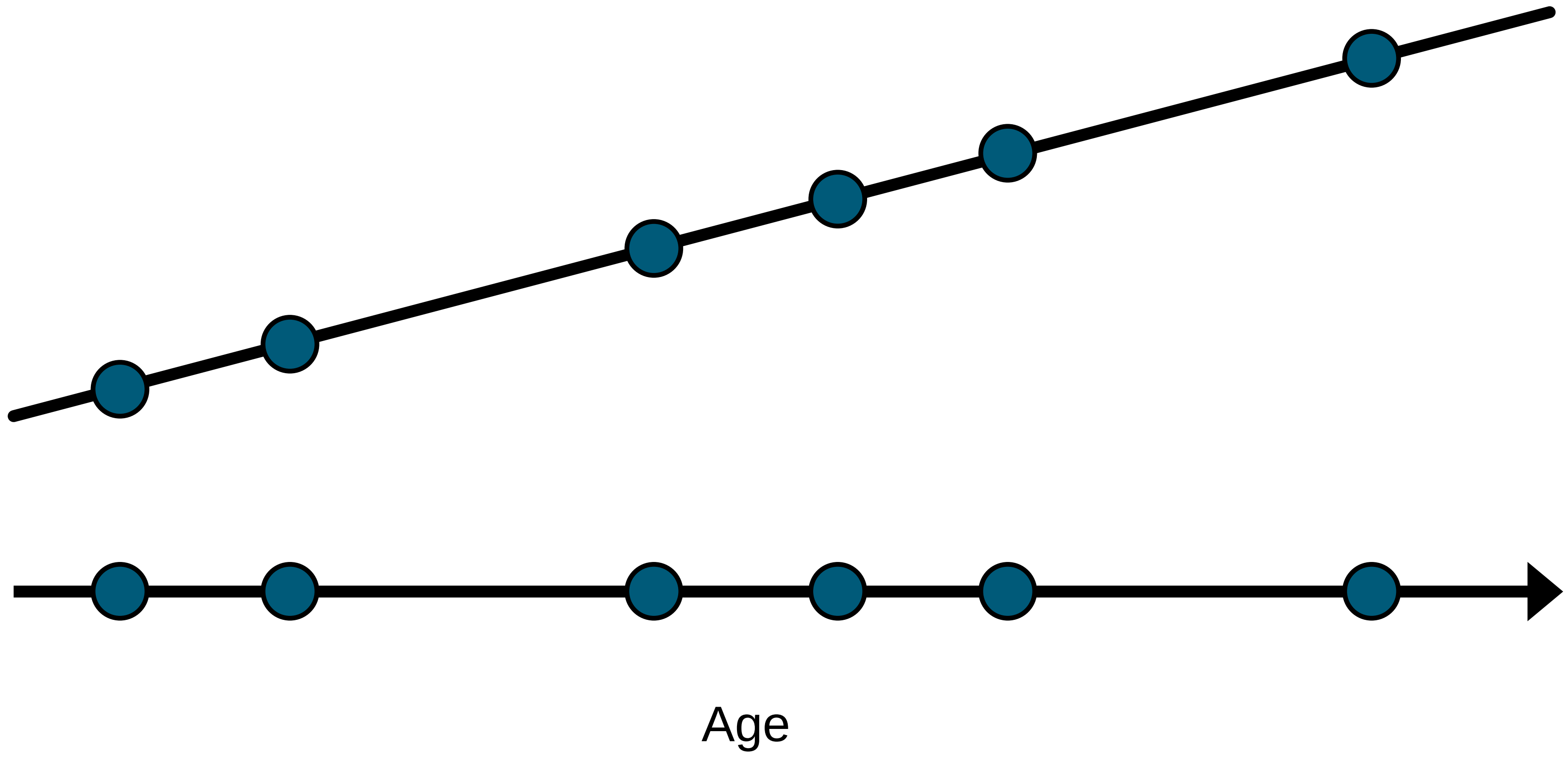
Accelerated Longitudinal Designs

Single cohort equally spaced:



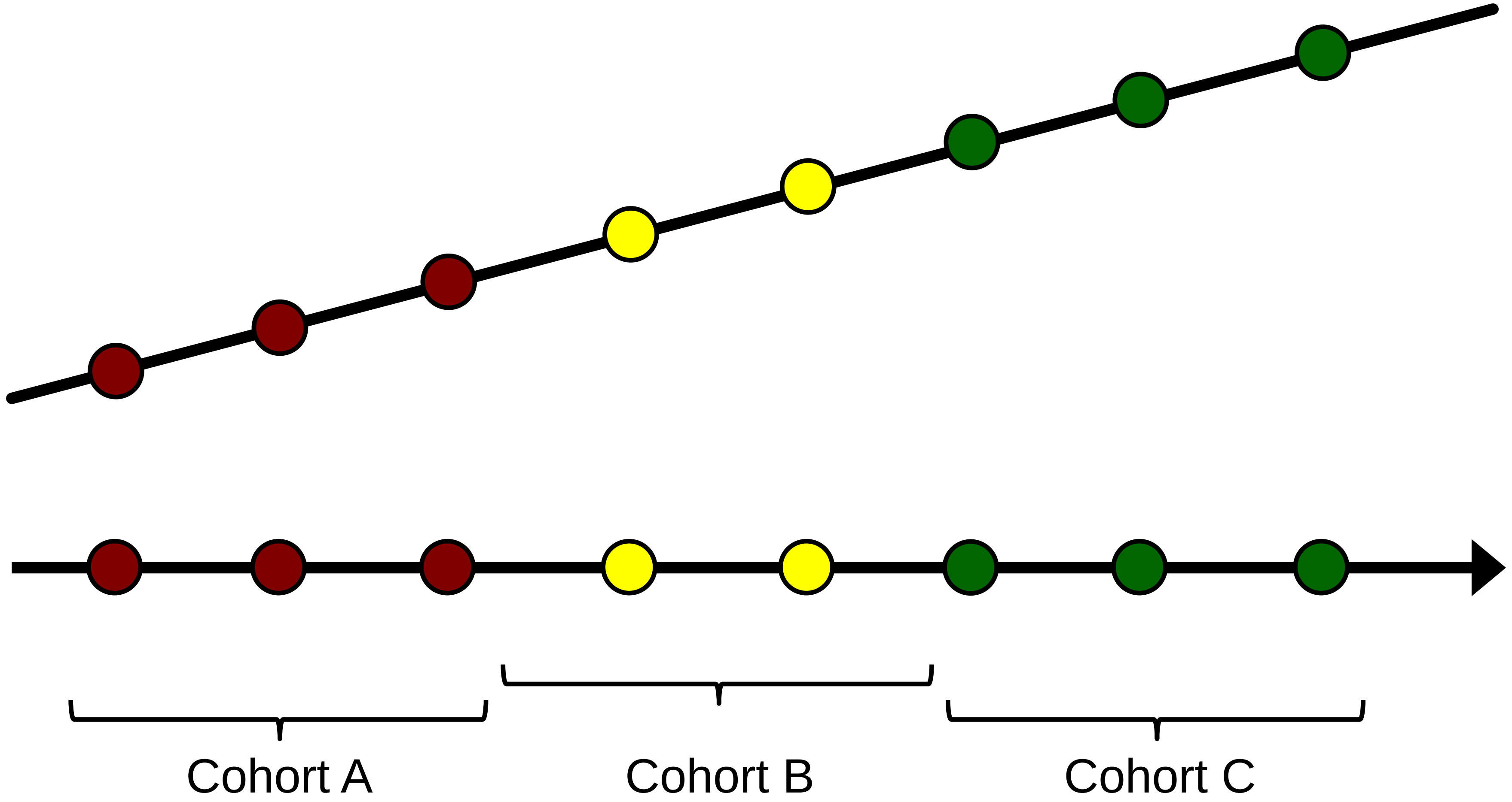
Accelerated Longitudinal Designs

Single cohort unequally spaced:



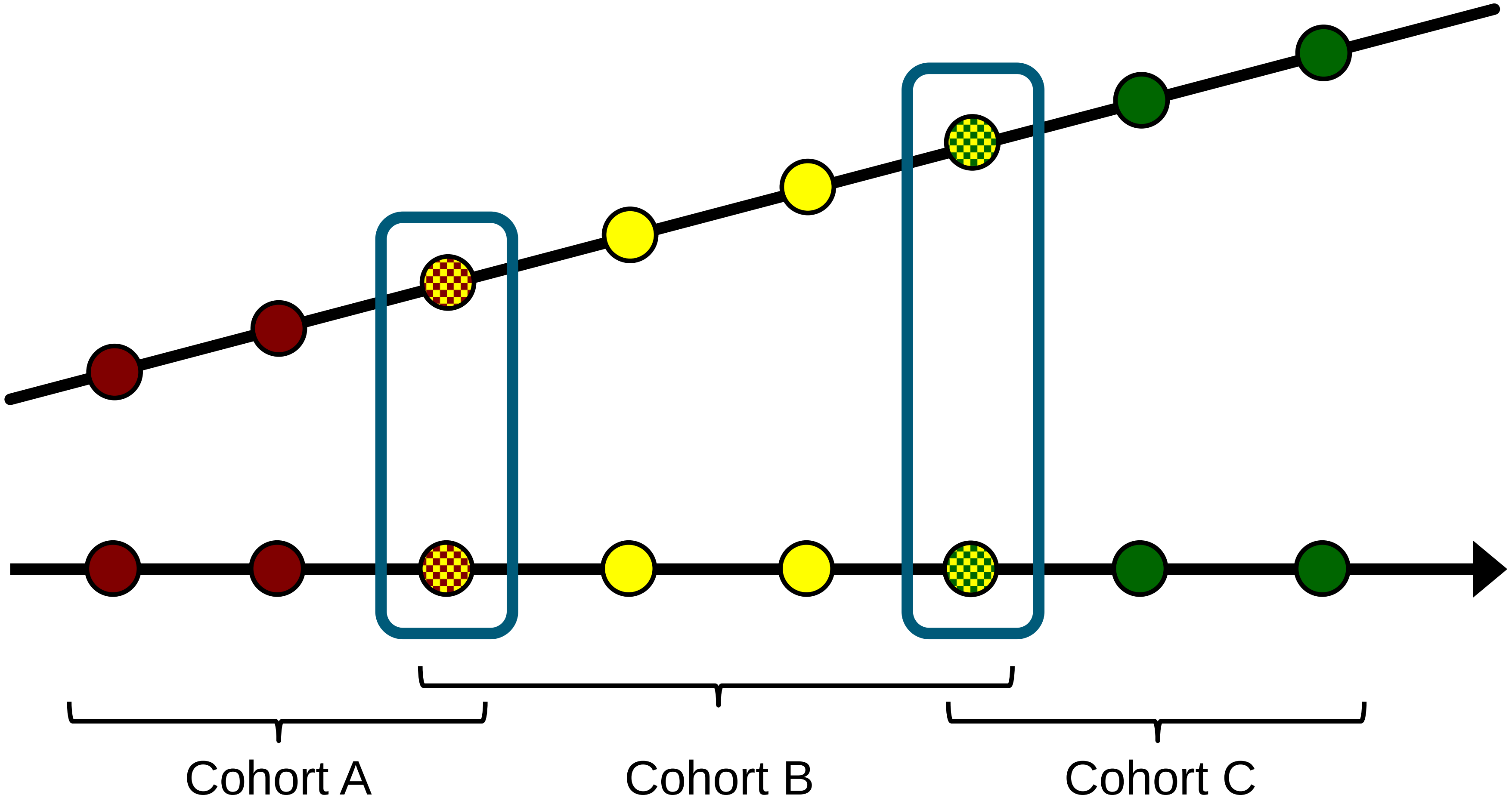
Accelerated Longitudinal Designs

Multiple cohorts (equivalent developmental process):



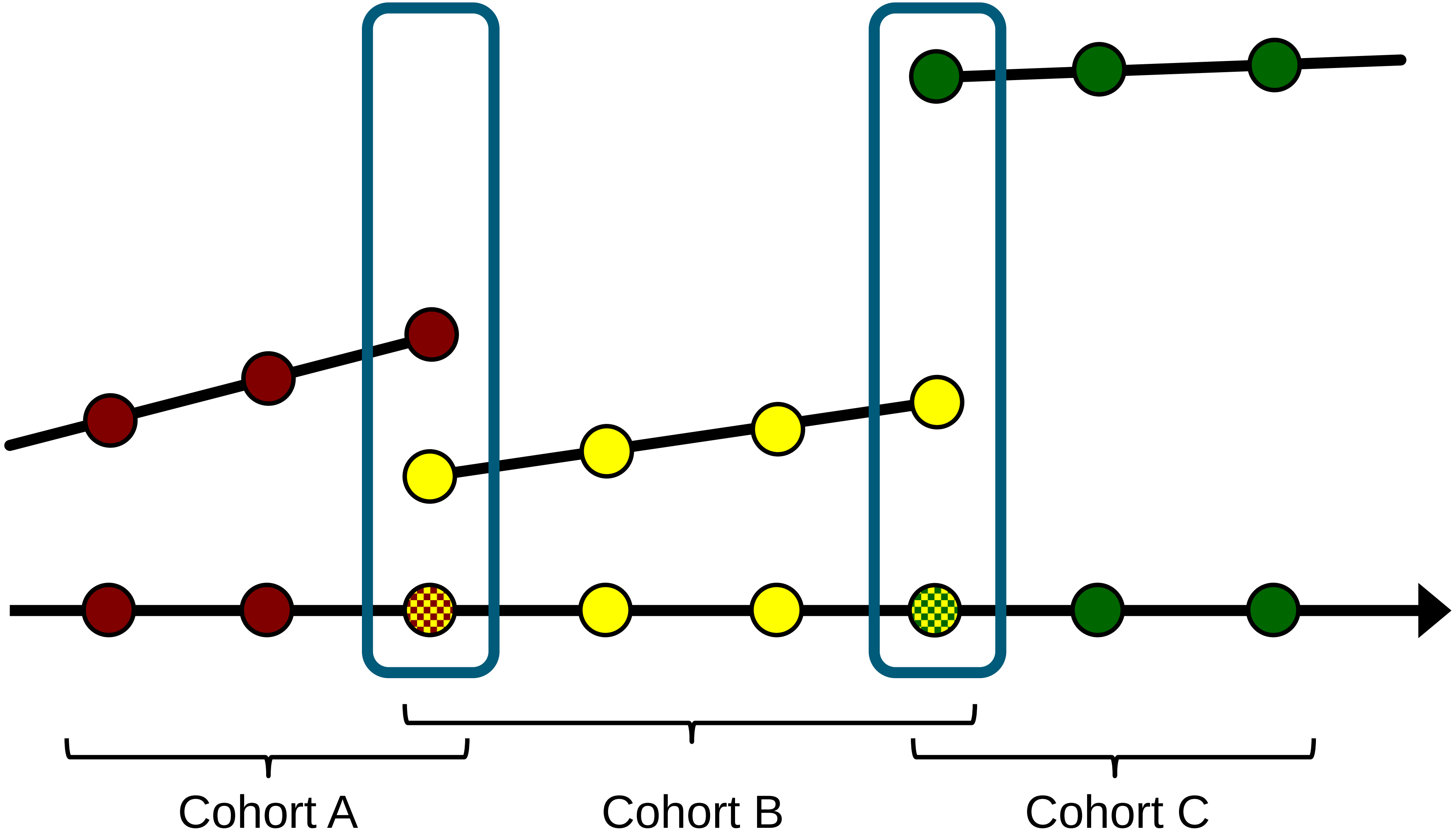
Accelerated Longitudinal Designs

Multiple cohorts (testing for equivalence):



Accelerated Longitudinal Designs

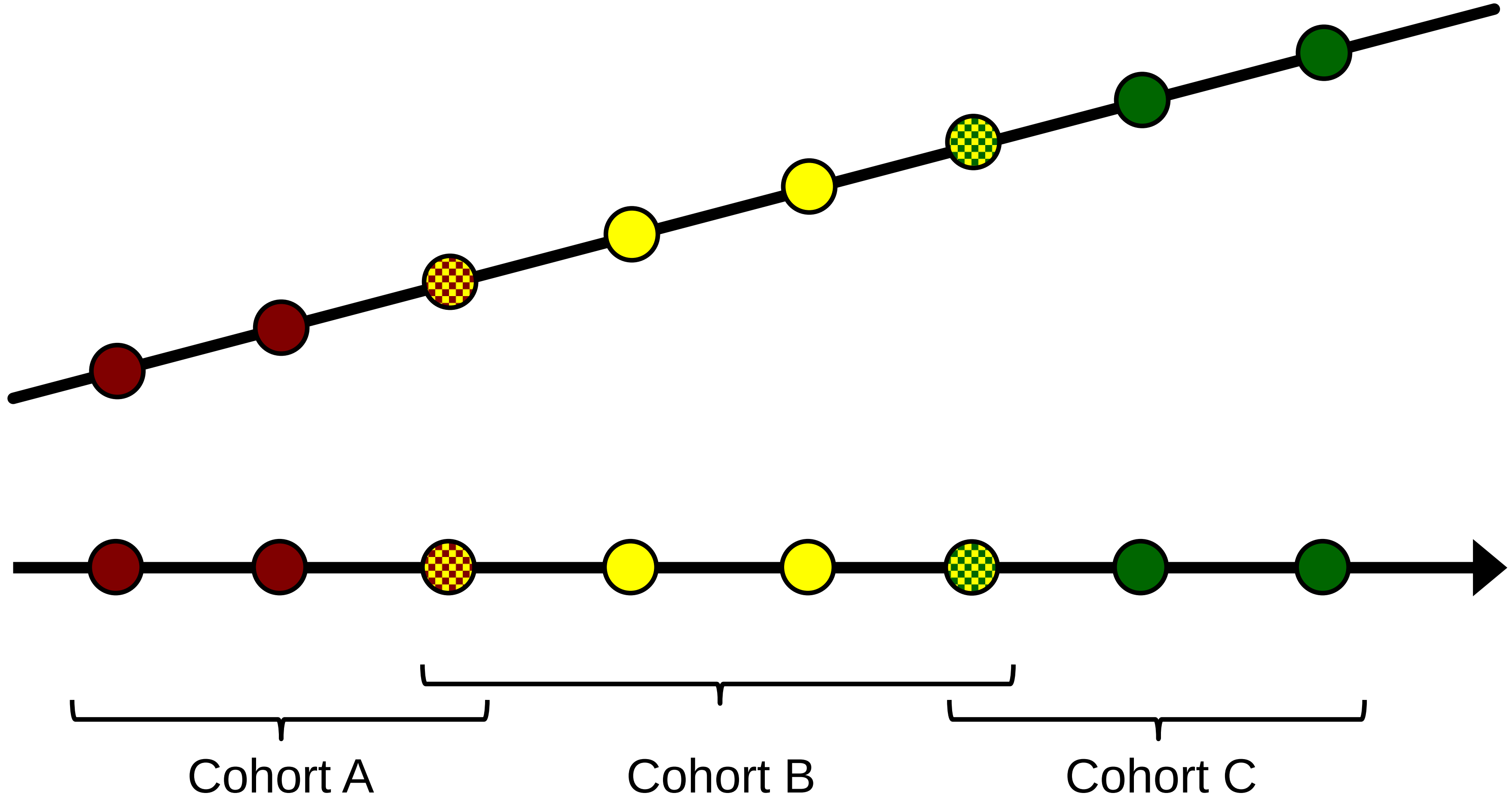
Multiple cohorts (nonequivalent developmental processes):



Accelerated Longitudinal Designs

Cross-sequential designs ≠ longitudinal designs

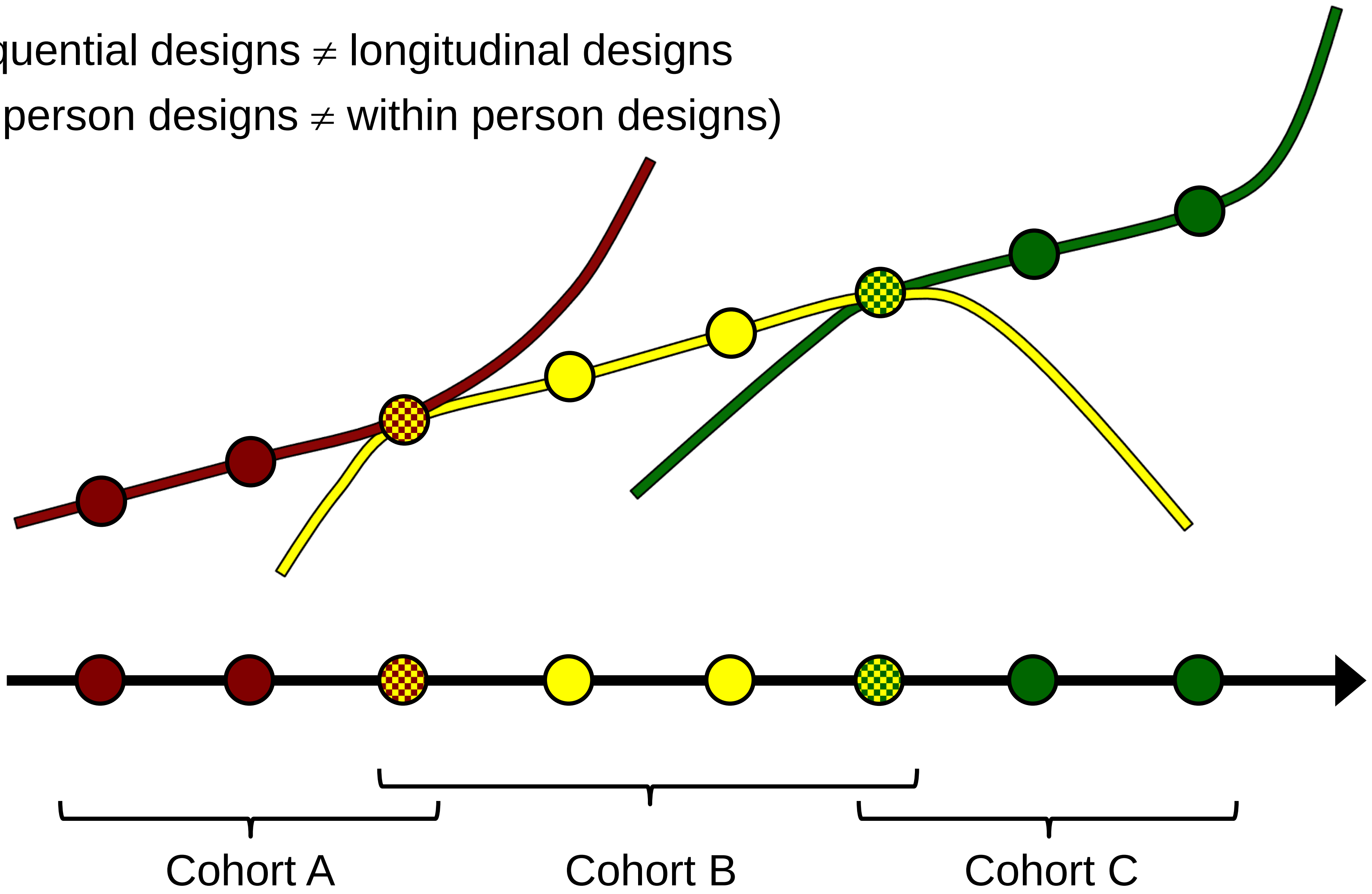
(between person designs ≠ within person designs)



Accelerated Longitudinal Designs

Cross-sequential designs ≠ longitudinal designs

(between person designs ≠ within person designs)



Oud & Voelkle (2013) *Quality & Quantity*

Accelerated Longitudinal Designs

Renewed interest in accelerated longitudinal designs (in continuous time), e.g.,

Cáncer, P. F., Estrada, E., & Ferrer, E. (2023). A Dynamic Approach to Control for Cohort Differences in Maturation Speed Using Accelerated Longitudinal Designs. *Structural Equation Modeling: A Multidisciplinary Journal*, 1-17. https://doi.org/10.1080/10705511.2022.2163647

Estrada, E., Bunge, S. A., & Ferrer, E. (2023). Controlling for cohort effects in accelerated longitudinal designs using continuous- and discrete-time dynamic models. *Psychological Methods, 28*(2), 359–378. https://doi.org/10.1037/met0000427

Estrada, E., & Ferrer, E. (2019). Studying developmental processes in accelerated cohort-sequential designs with discrete- and continuous-time latent change score models. *Psychol Methods, 24*(6), 708-734. https://doi.org/10.1037/met0000215

Accelerated Longitudinal Designs

Let's implement a continuous time cross-sequential design using *ctsem* along a blog-post by Charlie.

For scripts and details, see:

https://cdriver.netlify.app/post/accelerated/

09a_CT_cross_sequential.R

For an R-script using the multi-group functionality in ctsemOMX see :

09b_CT_cross_sequential.R

References

- Cáncer, P. F., Estrada, E., & Ferrer, E. (2023). A Dynamic Approach to Control for Cohort Differences in Maturation Speed Using Accelerated Longitudinal Designs. *Structural Equation Modeling: A Multidisciplinary Journal*, 1-17. https://doi.org/10.1080/10705511.2022.2163647
- Cattell, R. B. (1952, Sep). The three basic factor-analytic research designs-their interrelations and derivatives. Psychol Bull, 49(5), 499-520. https://doi.org/10.1037/h0054245
- Chow, S.-M., Ram, N., Boker, S. M., Fujita, F., & Clore, G. (2005). Emotion as a Thermostat: Representing Emotion Regulation Using a Damped Oscillator Model. Emotion, 5(2), 208-225. https://doi.org/10.1037/1528-3542.5.2.208
- Deboeck, P. R., & Bergeman, C. S. (2013). The Reservoir Model: A Differential Equation Model of Psychological Regulation. Psychological Methods, No Pagination Specified. https://doi.org/10.1037/a0031603
- Estrada, E., Bunge, S. A., & Ferrer, E. (2023). Controlling for cohort effects in accelerated longitudinal designs using continuous- and discrete-time dynamic models. *Psychological Methods, 28*(2), 359–378. https://doi.org/10.1037/met0000427
- Estrada, E., & Ferrer, E. (2019). Studying developmental processes in accelerated cohort-sequential designs with discrete- and continuous-time latent change score models. *Psychol Methods, 24*(6), 708-734. https://doi.org/10.1037/met0000215
- Molenaar, P. C. M. (1985). A dynamic factor model for the analysis of multivariate time series. Psychometrika, 50(2), 181-202. https://doi.org/10.1007/BF02294246
- Molenaar, P. C. M., & Nesselroade, J. R. (2009). The recoverability of P-technique factor analysis. Multivariate Behavioral Research, 44(1), 130-141. https://doi.org/10.1080/00273170802620204
- Oud, J. H. L., Voelkle, M. C., & Driver, C. C. (2018). SEM Based CARMA Time Series Modeling for Arbitrary *N*. Multivariate Behavioral Research, 53(1), 36-56. https://doi.org/10.1080/00273171.2017.1383224
- Voelkle, M. C., & Oud, J. H. L. (2013). Continuous time modelling with individually varying time intervals for oscillating and non-oscillating processes. British Journal of Mathematical and Statistical Psychology, 103-126. https://doi.org/10.1111/j.2044-8317.2012.02043.x

Study Questions

Question 1:

Compare P-technique and Dynamic Factor Analysis for time series data from a single subject. Why might one be preferred over the other, and under what conditions?

Question 2:

Explain the structure and utility of CARMA(p, q) models. What psychological or real-world phenomena might benefit from emplying CARMA(2,1) models?

Question 3:

Describe a damped linear oscillator in the context of ctsem. How are they detected and what psychological constructs might show this behavior?

Study Questions

Question 4:

Compare latent change score (LCS) and continuous-time models. Under what conditions do they yield the same results, and when should a CT model be preferred?

Question 2:

Explain the structure of accelerated longitudinal designs. How can ctsem be used to disentangle cohort effects from developmental processes?