Open Peer Review on Qeios

Structural equation modelling with lavaan: a tutorial and an intuitive introduction

Arindam Basu[1]

1 University of Canterbury

Funding: The author(s) received no specific funding for this work.

Potential competing interests: The author(s) declared that no potential competing interests exist.

Abstract

The goal of this paper is to present a tutorial on structural equation modelling (“SEM”). SEM is a combination of multivariate linear regression and path analysis models. We will discuss path analysis, measurement models, measurement invariance and when or how to use them, twin studies, and longitudinal data analysis. In this tutorial, we shall use the free and open source package “lavaan” in R.

Structural equation modelling in health sciences and epidemiology

Structural equation modelling (“SEM”) is a combination of multivariate linear regression and path analysis models. In this brief hands-on tutorial, we will discuss path analysis, measurement models, measurement invariance and when or how to use them, twin studies, and longitudinal data analysis using SEM. We shall use the free and open source package “lavaan” in R - the free and open source statistical programming language

Steps of SEM

As structural equation modelling is graphical, we recommend that you follow the sequence:

1. Draw your model as a system of paths
2. Input data in the form of covariance or correlation matrix
3. Identify the model
4. Specify the model
5. Assess parameter estimates
6. Assess fit measures (chi-square, df, residual matrix, GFI, RMSEA)
7. Check the modification indices
8. Rerun the model till you get the best fit of the data to the model and theory

Absolute first step before you proceed: Load the packages

Assuming that you use R for your analyses, you need to load the following packages in R after installing R:

```
library(lavaan)
library(dagitty)
library(ggdag)
library(tidyverse)
library(DiagrammeR)
library(DiagrammeRsvg)
library(rsvg)
```

Step 1: Notes on the system of paths and directed acyclic graphs (DAGs)

Everything graphical in SEM starts with path analysis. Richard Sewall Wright (1921), a hundred years ago, described a system of finding correlation between two variables, X and Y using a system of paths (Denis 2021). In this approach, he described that if a system of paths exist between two variables X and Y, the multiplication produce of the path coefficents of the sequences of the paths that traverse between the two variables should be added to the path coefficients of the direct paths that exist between X and Y to derive their correlations (Figure 1).

Figure 1. Basic path diagram

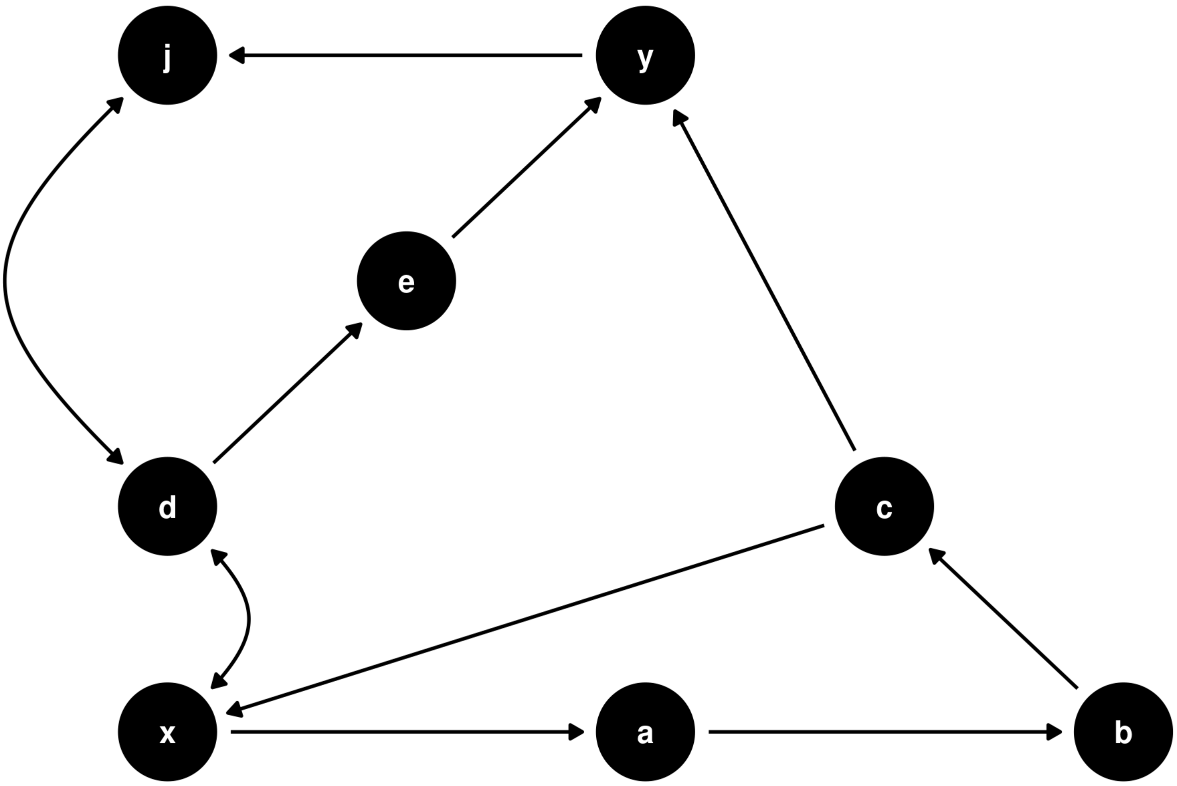


Figure 1. A path diagram

The above figure presents a system of paths that can be connected to derive covariances between pairs fo variables. These paths can be traced from one variable to another according to set rules. Sewall Wright described such a system of path tracing rules as follows:

1. A path can start from one variable to be connected to another variable and can start in either a forward or a reverse direction in the direction of the arrowhead
2. Once started in one direction, the path must continue in the same direction unless it meets with another path in a

reverse direction and at that point can proceed no further

3. A path can only contain ONE curved double headed arrow. A curved double headed arrow signifies either a covariance between two variables or a variance of a single variable
4. A path cannot go through the same variable twice, that is a path can only go through one variable at a time

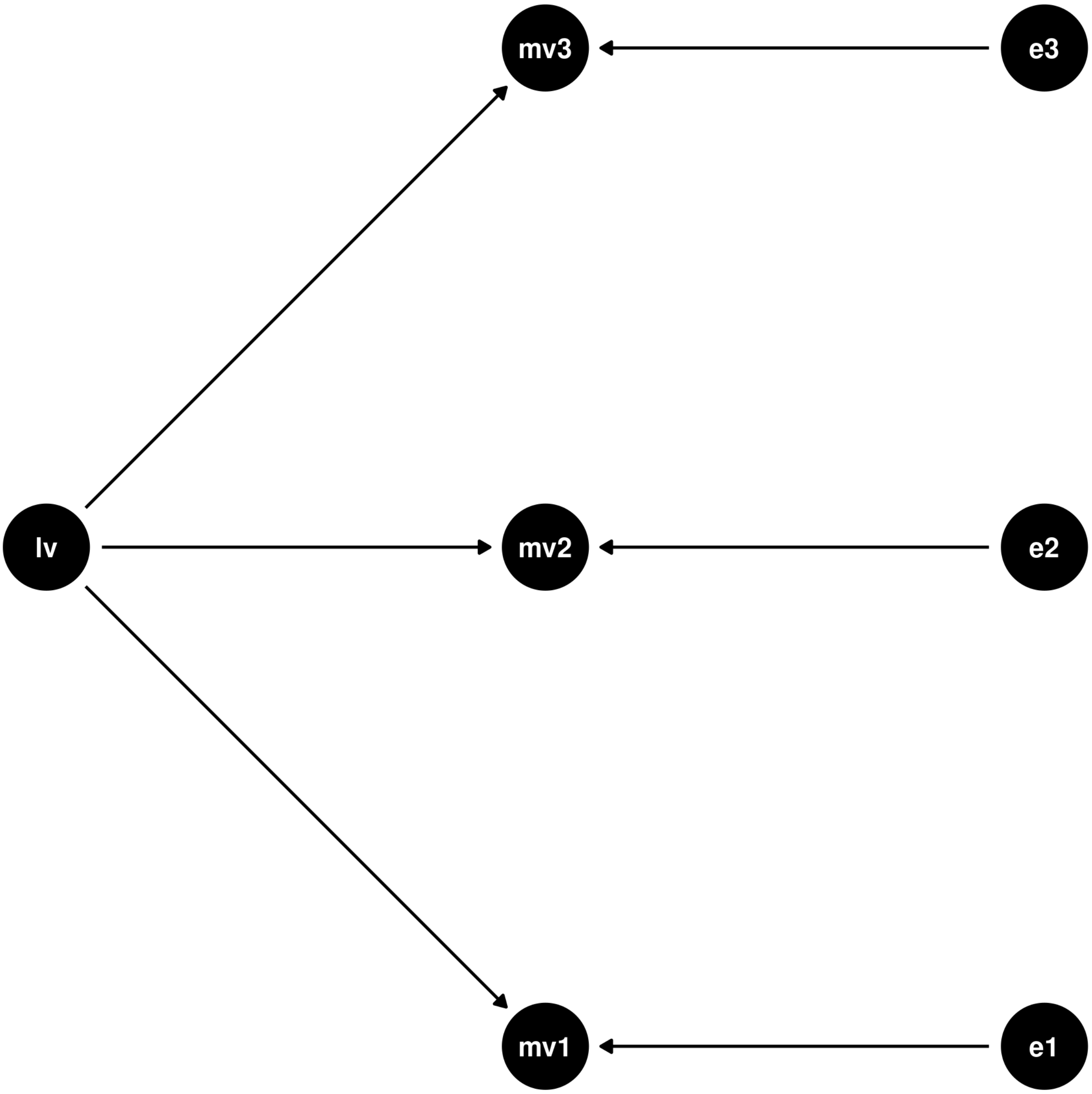
Then, once all the valid paths are identified, their path coefficients are multiplied and added to the direct path coefficient if one exists between the two variables to derive the correlation between these two paths. With these information, we can trace the following valid paths in Figure 1:

- x-a-b-c-y
- x-c-y
- x-d-e-y

These are the only three valid paths. No direct path exists between x and y, and all other paths are either invalid or they are blocked one way or another. In order to derive the correlation between x and y, we will need to add the multiple products of the individual path coefficients as follows:

```
cor(x,y) = (x-a)*(a-b)*(b-c)*(c-y) + (x-c)*(c-y) + (x-d)*(d-e)*(e-y)
```

Figure 2. Another system of paths we use in measurement model or a confirmatory factor analysis model.



Measurement model

Figure 2 above shows a model referred to as measurement model or confirmatory factor analysis model. The following table shows the variables:

Variable Name	What it means	Description
lv	Latent variable	Unobserved variable
mv1 … mv3	manifest variables	Variables that are physically measured
Endogenous variable	Variables that are explained by others	Arrows end in these variables
Exogenous variable	A variable that is used to explain another	Arrows start from these variables
e1 … e3	Error terms	Unexplained variances, the path coefficients are set at 1.0
constant term for mean structure	triangle	circle with 1.0 here

The path model in Figure 2 is a simple measurement model or confirmatory factor analysis model with one latent variable ("lv1"), and three manifest variables (mv1 … mv3). You can see that a set of six paths connect the manifest variables. In confirmatory factor analysis, we estimate or constrain such path coefficients and the path coefficients are used to derive the variances and covariances of these variables. The path coefficients also tell us the effect of one variable over another. For example, the effect of lv on mv1 in Figure 2 will be determined by the path coefficient of the path connecting lv with mv1. Note another feature: all the arrows in these diagrams point in one direction, and the variables are all connected by arrows that move in one direction. Such kind of graphs are referred to as directed acyclic graphs (DAG) as no variable has arrows that eventually return to itself closing any loop. DAGs are visual tools to directly observe the causal relationships between exposures and outcomes, including mediators, confounders, and effect modifiers; this is perhaps as accurate a definition of the role of DAGs as you can get Rohrer (2018).

Path diagrams for structural equation models ("SEM") have symbols that help the readers to understand what these models are doing. However, as our aim is to eventually discuss structural causal models in the light of structural equation models, we will briefly mention them here and continue with the models as we present here for uniformity. Table 2 provides a brief description of the symbols used for structural equation models and the differences we will adopt for our purposes in this tutorial.

Table 2. Symbols used in SEM and what we will do here

Symbol	Description	In our scheme
Square or Rectangle	Manifest variable	Round circle or letter
Circle	Error Term	Round Circle or letter
Oval	Latent Variable	Round or letter
Straight arrow	Path	Straight arrow
long Curved double headed arrow	Covariance	Curved arrow
short curved double headed arrow	Variance	Not shown here

We will assume that all exogenous latent and manifest variables have variances so we do not show them separately and all endogenous latent and manifest variables have exogenous error terms. Hence we suggest the above scheme, besides, using dagitty and ggdag packages in R helps us to draw these graphs in a uniform way. Alternative, publication quality causal and structural equation graphs can be drawn using graphviz and DiagrammeR packages but they are also time consuming. We recommend that for rapid visualisation of the models, use dagitty.

Path analysis in a measurement model: partition of variances

Refer to Figure 2 where we see a measurement model with one latent variable and three manifest variables. Here, we would like to use path tracing rules to derive the variance of manifest variable mv1. Here is the procedure:

- We will trace ALL paths that BEGIN with mv1 and end on mv1.
- We will multiply and add the path coefficients of all such paths
- What paths exist?
  - A path starts from mv1, goes to lv and returns to mv1 (we call this mv1-lv-mv1)
  - Another path is mv1-e1-mv1
  - No other path starts from mv1 and ends in mv1

Hence, - `variance of mv1 = (mv1-lv) * var(lv) * (lv-mv1) + (mv1-e1) * var(e1) * (e1-mv1)` Now, - mv1-lv and lv-mv1 are the same paths, so the path coefficient get squared - mv1-e1 and e1-mv1 are also the same paths, and we set the coefficients of these paths at 1.0 by convention - If we standardise lv, then var(lv) = 1.0

So, from here, we can say that the var(mv1) = square of the path coefficient from latent variable + variance of error term

The term “square of the path coefficient” is referred to as “communality,” because this part of the total variance (or variability, so to say) of mv1 is EXPLAINED by the latent variable that is common to all other manifest variables in the model that receive arrows from the latent variable. The path coefficient is also referred to as **factor loading**.

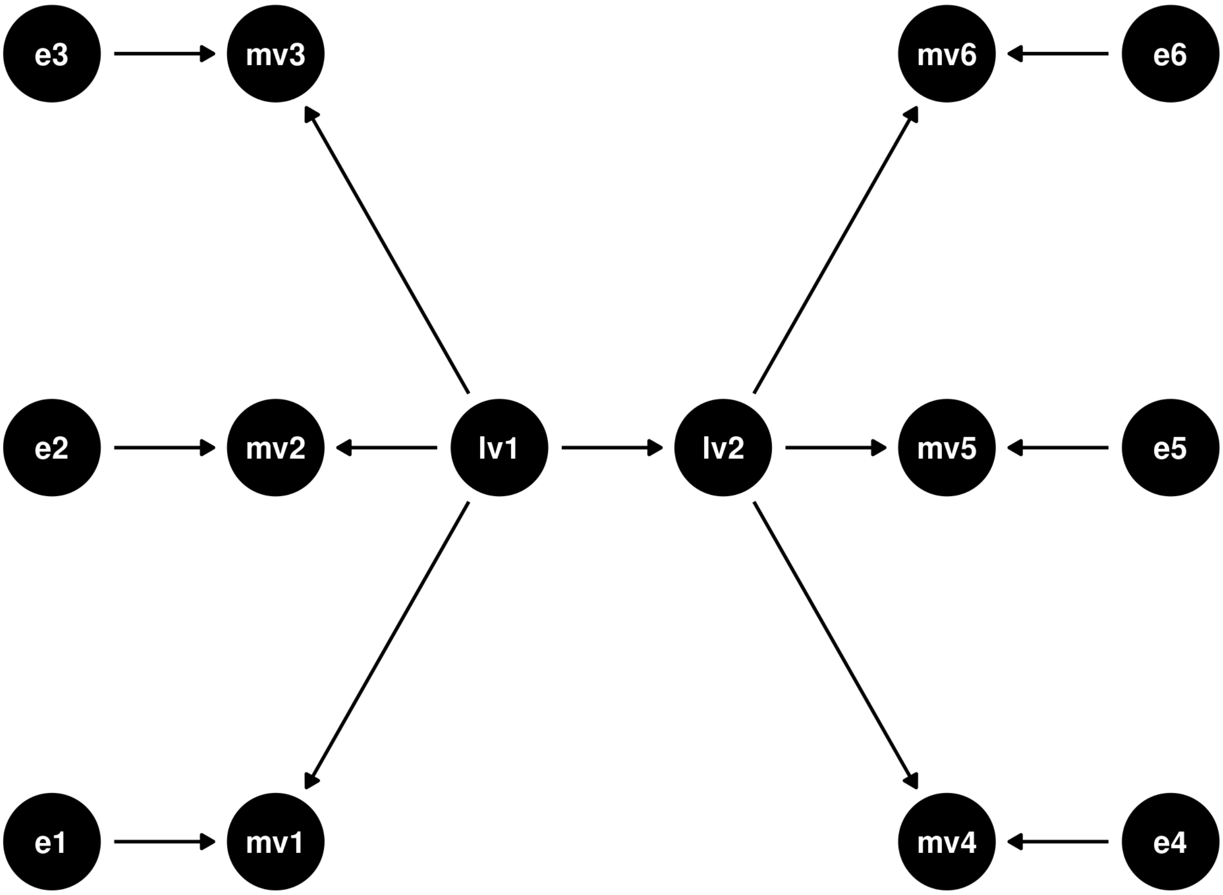
Using the path analysis approach, you will see that as the error terms are uncorrelated, therefore the correlation (or covariance) between mv1 and mv2 is given by

`mv1-lv * var(lv) * lv-mv2`

These concepts are fundamental to understanding what goes in SEMs. We have seen one model, that of a measurement model or confirmatory factor analysis model, where we have one or more latent variables *load* on manifest variables.

Figure 3 shows the structural and measurement parts of an SEM

Figure 3. SEM with both measurement and structural parts



Structural model

As you can see in Figure 3, on the left and right are two separate measurement models, where lv1 and lv2 are the respective latent variables, mv1 … mv3 are the manifest variables on the left hand side measurement model, and mv4 … mv6 are manifest variables on the right hand side measurement model. We have now added a regression model to the mix where lv2 is regressed on lv1, so the path coefficient of lv1-lv2 can be viewed as a beta coefficient, with something like:

lv2 ~ beta * lv1

This is the path diagram of a full structural equation model. Note that this model incorporates BOTH a set of measurement models (two measurement models here) and a structural model (lv1-lv2 path). You can extend and make these models as complex or as simple as you want. You could probably have only one manifest variable for the structural part where you would regress the manifest variable on the latent variable (simple), or you could have many more latent variable models that you would link up to form complex patterns that you would like to analyse.

Path diagrams of models with meanstructures and group comparisons

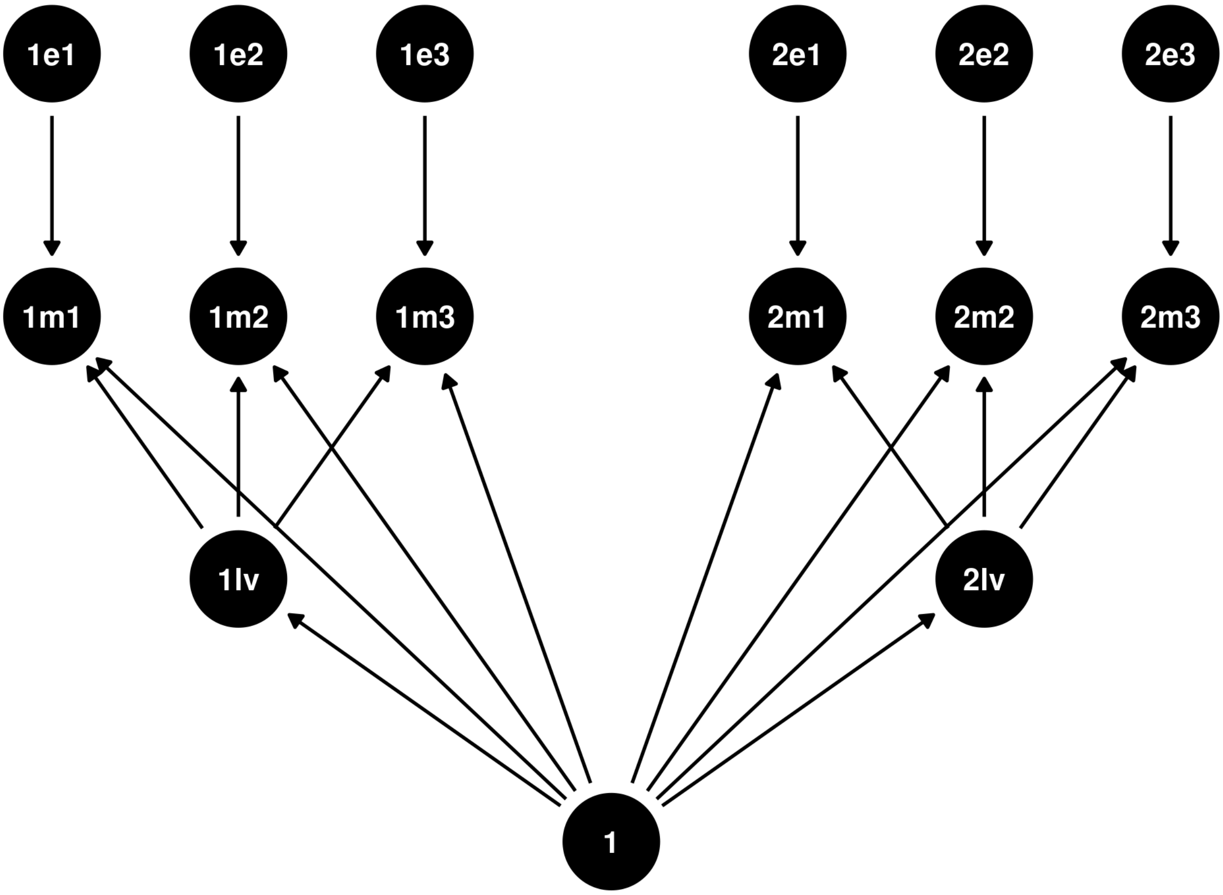
So far, we have confined our discussions to models that have only one group of people and models that only have explained covariances and variances. For example, a measurement model would be well suited to test the validity of the construct of a questionnaire you have set up to investigate some health construct. Say you have developed a questionnaire that aims to tap an individual's concept of "health" and decide to distribute this questionnaire to 200 individuals, and obtain data from them. Each individual is asked five questions, and you could have a measurement model out of these five questions and a latent or unobserved construct of “health” from your research participants. Such a procedure would provide you with an estimate of whether you were able to tap the construct of “health” based on the items you asked your participants.

Now imagine that among your participants there are those with chronic diseases (such as diabetes, high blood pressure, chronic heart disease and so on) and those who are otherwise healthy adults with no evidence of any disease. You may claim that the questions were perceived in the same way among members of both the groups, and that the factor loadings of the latent variable (in this case “health”) would be equal in both groups; So even if the unexplained and explained **variances** of the manifest variables may be similar in both groups, it might still mean that they would have different average values on those scores. In other words, you may want to find out group differences or invariances of your measurements across groups to make sure that your model is a robust model. This is where a mean structure is important (Meredith 1993).

In SEM, the means of the manifest variables are referred to as “intercepts” and the means of the latent variables as “factor mean.” The factor mean is derived from a constant term, represented by a triangle (we will present here in the form of a circle with 1.0 as value). In a standardised solution, the path coefficient from the constant to the latent variable is set at 0 (mean = 0 for a standardised variable). Also note that as this is a constant, it’s variance is 0, and therefore it contributes only to the mean of the latent and manifest variables. When the mean of the latent variable is set to 0, the intercept of the model is same as the mean of the individual manifest variables. Such a structure is used to compare and evaluate that the measurements and the measurement properties (such as factor loadings, and variance of the latent variable) REMAIN THE SAME for each group. The groups themselves can be constituted on the basis of categorical variables such as sex, race, or case-control status, for example.

Figure 4 shows such a scenario

Figure 4. SEM with mean structure and group comparison



Mean structures

As you can see in Figure 4:

- Two arrows head from 1 (the *constant*) to the two latent variables.
- Both latent variables are “lv,” except for group 1 it is 1lv, and for group 2, it is 2lv (similarly for manifest variables mv1, …, mv3 and e1 … e3, they are prefaced with 1 for group 1 and 2 for group 2)
- Three arrows on each side go from 1 to the manifest variables; these are the intercepts
- Then we have the measurement models of the two groups, group 1 and group 2 Now, a few things to consider here:

1. For each manifest variable in each group, we have their intercept, their variance, and their covariance on the basis of which the measurement model is formed
2. For each factor in each group, there are three manifest variables (in this case)
3. Each factor in each group (i.e., latent variable), we have measured their latent mean (the arrows that go from 1 to 1lv and 2lv), we have the factor loadings, and the variances

We believe that the groups will differ in some ways, **but** we must make sure that they were measured in EXACTLY the same way. So, while we study both (or more groups), we can restrict or constrain the group parameters in a number of ways:

- At the least, we say that we only constrain that the **configuration** of the latent and manifest variables remain invariant in between the groups, but everything else can vary. If this is the case, this will be one of configural measurement invariance

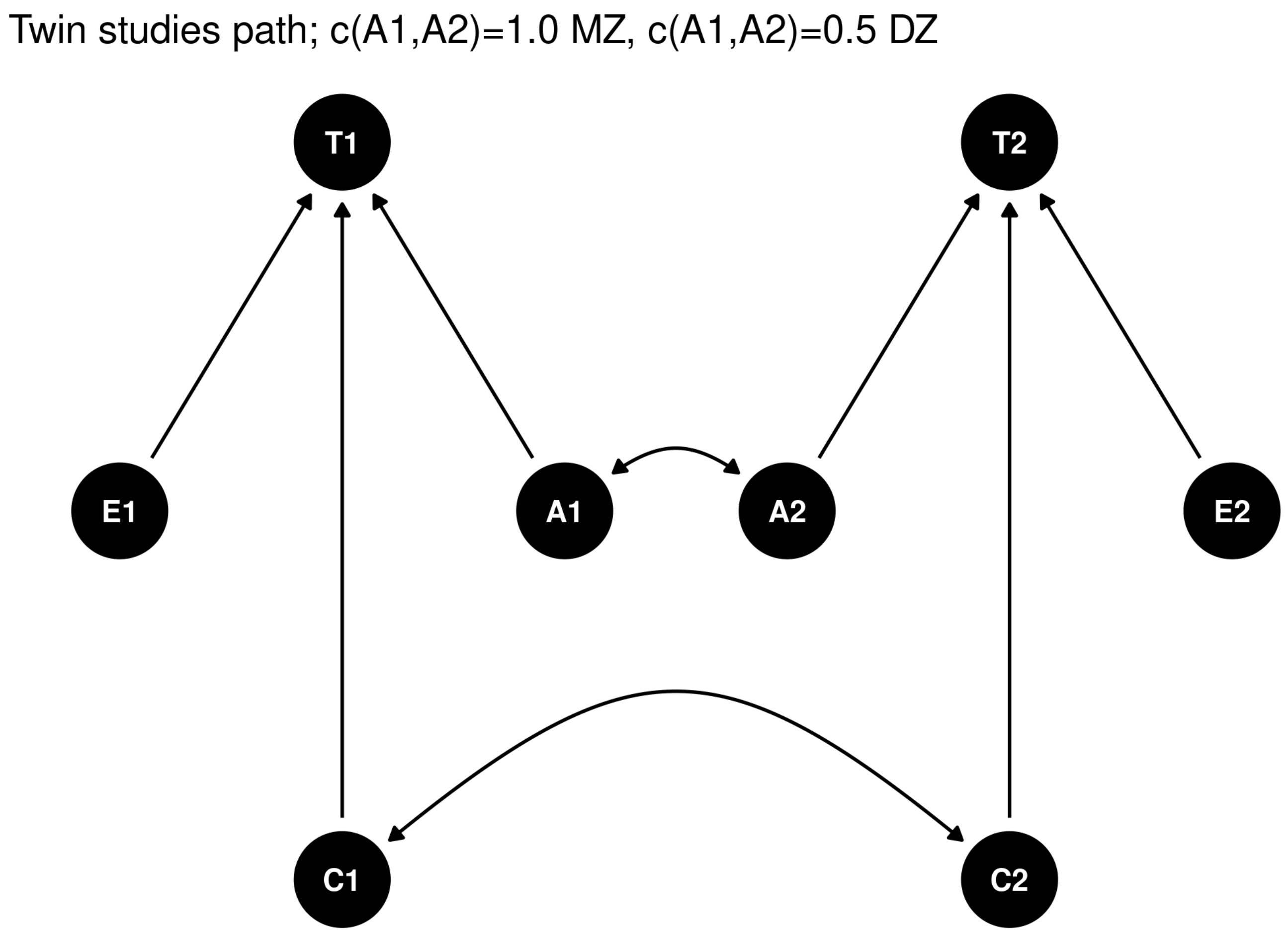
- Then we can say that our **factor loadings** MUST be constrained in both the groups so as to ensure that both the groups had similar measurements. This is a weak measurement invariance because we still leave the possibiity that the variance of the latent variables themselves could differ between the groups
- Then we add the constraints of both **factor loadings** as well as **latent variable variances** to remain identical across the groups, but the latent means could still vary. This is *strong invariance* and is most practical assumption
- Finally, we can add the constraint that everything will be identical between the groups and see what factor structure will satisfy this condition and how would the model hold. This is referred to as **strict measurement invariance**.

Using strong invariance to examine the group differences or groups is important to ascertain as to what differences exist even as the measurement of the constructs were conducted identically. Such groups can be two periods of time for the same population, or same sample, or two conditions (cases and controls or exposed and non-exposed, or treatment and control conditions), or gender, or indeed any characteristic of the individuals. It might be useful to investigate situations over two periods of time (as in baseline and final time point).

Group comparison in the context of twin studies

Group comparison is particularly useful in the context of twin studies (Neale and Cardon 2013). The problem and the solution are as follows.

Figure 6. Twin study design



Twin study design

Say you want to test the hypothesis that BMI is determined by genes. BMI is a quantitative trait, so rather than one gene, we hypothesise that several genes **add up** to exert their effects on BMI. Yet we may also argue that in addition to genes, there are environmental variables that also determine individual BMI scores. To study such effects, you have collected data from 400 twins (200 monozygotic twins and 200 dizygotic twins), these twins were all raised in the same household and shared the same environment; however, even then, they also had unique environmental factors other than their shared households and rearing (such as different friends, different universities where they studied, different occupations, etc). So, if you take PAIRS of monzygotic and dizygotic twins and examine these TWO groups using a path analysis model, you will have the following:

- For each of the 200 pairs of Monozygotic (and it is true for dizygotic twins), you will have data on their BMI
- Twin1 and Twin2 (T1 and T2) will have on BMI a variance-covariance matrix whose patterns can be explained by three variables:
    - their additive genetic effects (As)
    - their common environmental effects (Cs)
    - their dominant genetic effects (Ds): this is the case where one gene is dominant over another
    - their unique environmental effects (Es), this is basically their error variance
- For MONOZYGOTIC twins, as they have EXACTLY the same complement of genes, their As will perfectly vary, i.e.,

cor(A1, A2) = 1.0

- For DIZYGOTIC twins, as they share 50% of their genes, their As will have cor(A1, A2) = 0.5
- For DIZYGOTIC twins, as they share 25% of their dominant genes (heterozygosity), this will be cor(D1, D2) = 0.25
- For BOTH monzygotic and dizygotic twins, their shared environments (Cs) will be perfectly correlated, so cor(C1, C2) = 1.0
- for BOTH mono- and dizygotic twins, their unique environments are uncorrelated, so we will see cor(E1, E2) = 0
- a$ ^2$ + e = 1.00 (assuming standardised coefficients)
- a is also referred to as “narrow heritability”

Figure 5 shows the twin studies path diagram

Figure 5. Path diagram for twin studies (same structure for BOTH MZ and DZ twins, the correlations will differ, that’s all, the path coefficients are identical)

While analysing these paths, we have to be careful that while Cs are common to both MZ and DZ twins, the path coefficient that explains the part of the variation in the phenotype (say BMI in the example), will be very low if not negligible. This needs to be taken into account as you assess these models, so setting or constraining the path coefficients C1A1 and C2A2 to close to 0 (something like 0.001) is a useful strategy.

Path analysis of longitudinal data or latent growth curve models

So far, we have discussed path models that are all assessed over a single period of time, and even when we discussed measurement invariance and discussed two groups in two periods of time, the scope was limited. We now turn to the problem of what happens when you have correlated measures over repeated measurements taken over three or more periods. These models are referred to as latent growth curve models.

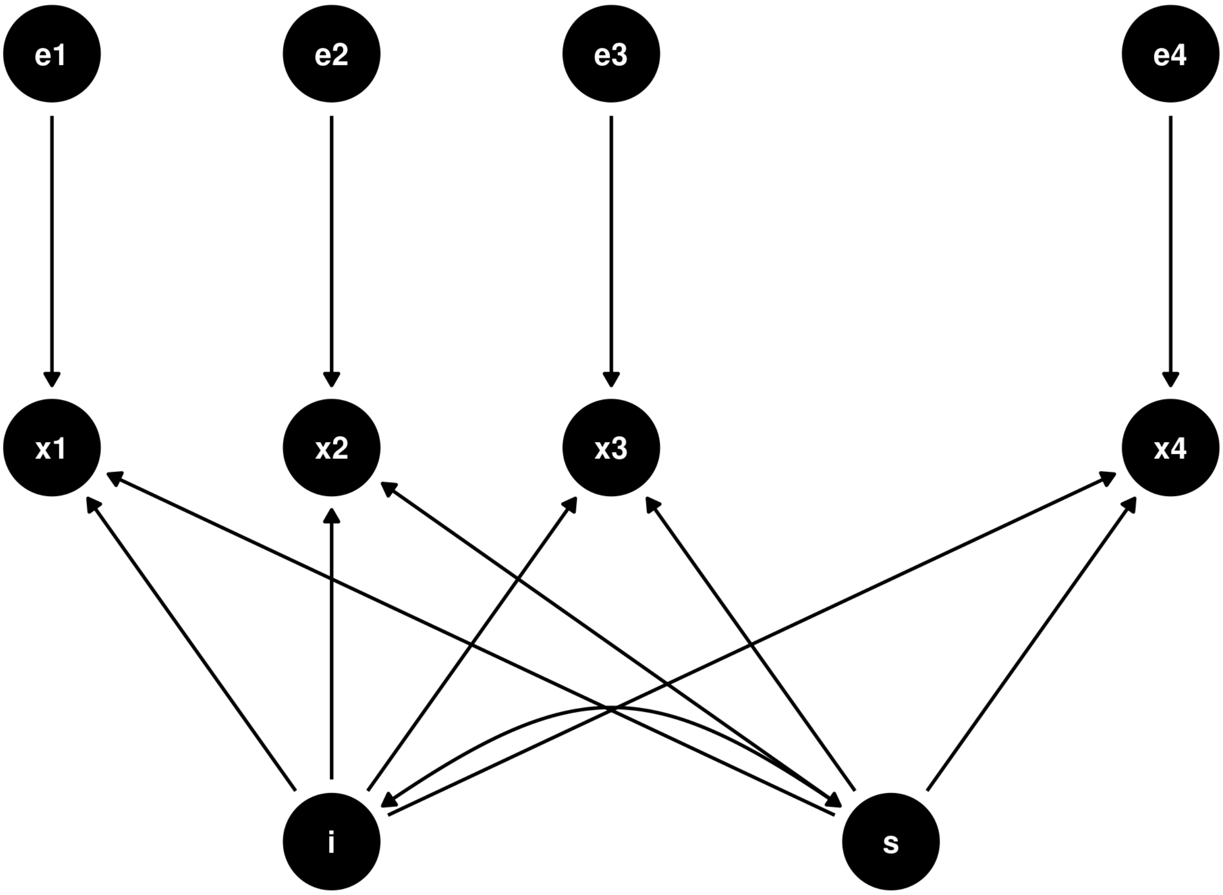
There are two major differences between the models we have studied so far and latent growth curve models. First, so far we have assumed that the data for all our models came from studies that were conducted over a specified period of time, rather than longitudinally. However, while this may be so for majority of the studies for which we would use SEM, for clinical and public health studies, large longitudinal studies for which data collection happens over repeated measurements over long periods of time are important. Longitudinal ageing studies are important study designs that allow studying change over time. Second, we have considered how latent variables would **explain** variability in the manifest variables, so we have considered path coefficients that we’d like to measure as influences. When we analyse longitudinal studies, we *fix* the parameters. Instead we study two issues with longitudinal studies:

1. We study change. – Now, change is unobserved and therefore change is inherently a latent construct.
2. Change has two parameters: it has to have an intercept, and it has to have a slope. It has to have an intercept for the latent or unobserved construct of change, as change has to happen from a baseline, such as change from value X to value Y over a period of time. X in our case, is a baseline intercept value for the latent variable. We have met intercept before, but that was intercept for the manifest variable, which, in this case, are time-bound value of some trait repeated over time. We are not talking about that, instead, we mean what would be the latent value of baseline measure at ZERO TIME? Is that time constant, or is that time variable over the number of people or the context of the study? Is that level a sample from a distribution of infinite other values, i.e., a random effect?

3. Change must also have some kind of rate of change. This could be linear, this could be quadratic. Equally, as we factor in time, we factor in what intervals over which such change occur? Are the data collected linearly, as in every n number of years? Or are data collected every two years for the first two waves and then every five years for the next few waves and so on?

Figure 6 shows a simple latent growth curve model

Figure 6. A simple latent growth curve model



Latent growth curve model

As you can see, x1 … x4 shows the measured X variables that we want to model over 4 periods of time. These are our manifest variables. The arrows from i, the latent intercept is fixed at 1.0 to indicate that the latent intercept has the *same influence* over the time bound values. The arrows from latent slope, s, to x1 … x4 is fixed as follows: if we want to model them as linear, we start with 0, so in this case this will be 0,1,2,3. A 0th time point is the beginning or the first measurement. Depending on the frequency over which data were collected, we adjust the value of the path coefficients of the paths that run from s to individual xs. Finally the error terms for each time point observation. Here we have shown them to be uncorrelated, but these can be correlated errors.

Code of all the graphs and diagrams we have presented so far:

1. Basic path diagram

```
mypaths <- dagitty('dag{
      x [pos = "1,1"]
      a [pos = "2,1"]
      b [pos = "3,1"]
      c [pos = "2.5,2"]
      d [pos = "1,2"]
      e [pos = "1.5,3"]
      j [pos = "1,4"]
      y [pos = "2,4"]

      x -> a
      a -> b
      b -> c
      c -> y
      c -> x
      x <-> d
      d <-> j
      y -> j
      d -> e -> y


}')

plot(mypaths)

mypaths %>%
 ggdag() +
 theme_dag_blank() +
 ggsave("path1.png")
```

Draw measurement model

```
path_mm = dagitty('dag{
    lv [pos = "1,2"]
    mv1 [pos = "2,1"]
    mv2 [pos = "2,2"]
    mv3 [pos = "2,3"]
    e1 [pos = "3,1"]
    e2 [pos = "3,2"]
    e3 [pos = "3,3"]

    lv <-> lv
    lv -> mv1
    lv -> mv2
    lv -> mv3
    e1 -> mv1
    e1 <-> e1
    e2 -> mv2
    e2 <-> e2
    e3 -> mv3
    e3 <-> e3



}')
```

```
path_mm %>%
 ggdag() +
 theme_dag_blank() +
 ggsave("path_mm.png"
```

Code for structural equation modelling

```
semdag <- dagitty('dag{
   e1 [pos = "1,1"]
   mv1 [pos = "2,1"]
   e2 [pos = "1,2"]
   mv2 [pos = "2,2"]
   e3 [pos = "1,3"]
   mv3 [pos = "2,3"]
   lv1 [pos = "3,2"]
   lv2 [pos = "4,2"]
   mv4 [pos = "5,1"]
   e4 [pos = "6,1"]
```

```
  mv5 [pos = "5,2"]
  e5 [pos = "6,2"]
  mv6 [pos = "5,3"]
  e6 [pos = "6,3"]

  e1 -> mv1
  e2 -> mv2
  e3 -> mv3
  lv1 -> mv1
  lv1 -> mv2
  lv1 -> mv3
  lv1 -> lv2
  lv2 -> mv4
  e4 -> mv4
  lv2 -> mv5
  e5 -> mv5
  lv2 -> mv6
  e6 -> mv6


}')
```

```
semdag %>%
 ggdag() +
 theme_dag_blank() +
 ggsave("semfull.png")
```

Code for Mean structures

```
meanstr <- dagitty('dag{
  1 [pos = "4,1"]
  1lv [pos = "2,2"]
  2lv [pos = "6,2"]
  1m1 [pos = "1,3"]
  1m2 [pos = "2,3"]
  1m3 [pos = "3,3"]
  2m1 [pos = "5,3"]
  2m2 [pos = "6,3"]
```

```
    2m3 [pos = "7,3"]
    1e1 [pos = "1,4"]
    1e2 [pos = "2,4"]
    1e3 [pos = "3,4"]
    2e1 [pos = "5,4"]
    2e2 [pos = "6,4"]
    2e3 [pos = "7,4"]

    1 -> 1lv
    1 -> 2lv
    1lv -> {1m1 1m2 1m3}
    2lv -> {2m1 2m2 2m3}
    1 -> {1m1 1m2 1m3}
    1 -> {2m1 2m2 2m3}
    1e1 -> 1m1
    1e2 -> 1m2
    1e3 -> 1m3
    2e1 -> 2m1
    2e2 -> 2m2
    2e3 -> 2m3



}')
```

```
meanstr %>%
  ggdag() +
  theme_dag_blank() +
  ggsave("meanstr.png")
```

Diagram code for twin studies

```
twin_studies <- dagitty('dag{

    E1 [pos = "1,2"]
    C1 [pos = "2,1"]
    A1 [pos = "3,2"]
    A2 [pos = "4,2"]
    C2 [pos = "5,1"]
    E2 [pos = "6,2"]
    T1 [pos = "2,3"]
    T2 [pos = "5,3"]

    E1 -> T1
    C1 -> T1
    A1 -> T1
    A1 <-> A2
    A2 -> T2
    C2 -> T2
    E2 -> T2
    C1 <-> C2

}')
```

```
twin_studies %>%
 ggdag() +
 theme_dag_blank() +
 ggtitle("Twin studies path; c(A1,A2)=1.0 MZ, c(A1,A2)=0.5 DZ") +
 ggsave("twins.png")
```

Code for drawing diagrams for latent growth curve models

```
lgcm <- dagitty('dag{
  x1 [pos = "1,2"]
  i [pos = "2,1"]
  x2 [pos = "2,2"]
  x3 [pos = "3,2"]
  s [pos = "4,1"]
  x4 [pos = "5,2"]
  e1 [pos = "1,3"]
  e2 [pos = "2,3"]
  e3 [pos = "3,3"]
  e4 [pos = "5,3"]

  i <-> s
  i -> {x1 x2 x3 x4}
  s -> {x1 x2 x3 x4}
  e1 -> x1
  e2 -> x2
  e3 -> x3
  e4 -> x4


}')

lgcm %>%
  ggdag() +
  theme_dag_blank() +
  ggsave("lgcm.png")
```

Step 2: Input data in the form of a covariance matrix

In order to analyse data in SEM, you will need the following:

- A correlation or a covariance matrix
- A set of standard deviation (if you use correlation matrix)
- a set of means for the individual manifest variables
- Sample size (the more the better)

In our examples of SEM here, we will use lavaan package in R. You can download and install lavaan from lavaan’s website. There are several ways of getting covariance or correlation matrices:

Direct input of numbers

You can input a raw set of numbers as variance-covariance or correlation matrix (lower or upper) and using the function lav_matrix_lower2full() you can obtain a full matrix in lavaan. ## Convert between correlation and covariance matrices Besides, if you have a correlation matrix and a set of standard deviation, you can convert a correlation matrix to a covariance matrix using the function cor2cov(matrix, sd) ## Directly compute the matrix from the variables in your data set Using the function cov(c(v1, ..., vn), na.rm = T), remember to set na.rm = T in R

Examples of each one with illustrations

Confirmatory factor analysis

We will analyse better life index data. You can find more about the project here, and you can find the data in our github archive here:

Code:

```
# Input the data and clean the data
bli <- read_csv("https://raw.githubusercontent.com/arinbasu/2021_04_sem_datasets/main/bti-scores-regional.csv")
##
## ── Column specification ────────────────────────────────────────────────────────
## cols(
##   Country = col_character(),
##   Region = col_character(),
##   `Education and skills` = col_double(),
##   Jobs = col_double(),
##   Income = col_double(),
##   Safety = col_double(),
##   Health = col_double(),
##   Environment = col_double(),
##   `Civic engagement` = col_double(),
##   `Accessiblity to services` = col_double(),
##   Housing = col_double(),
##   Community = col_double(),
##   `Life satisfaction` = col_double(),
##   Overall = col_double()
## )
bli %>% names()
##  [1] "Country"                  "Region"
##  [3] "Education and skills"     "Jobs"
##  [5] "Income"                   "Safety"
##  [7] "Health"                   "Environment"
##  [9] "Civic engagement"         "Accessiblity to services"
## [11] "Housing"                  "Community"
## [13] "Life satisfaction"        "Overall"
bli <- bli %>%
  rename(life_satisfaction = `Life satisfaction`,
         education = `Education and skills`,
         civic = `Civic engagement`,
         accessibility = `Accessiblity to services`)
bli %>% head()
## # A tibble: 6 x 14
##   Country Region        education  Jobs Income Safety Health Environment civic
##   <chr>   <chr>             <dbl> <dbl>  <dbl>  <dbl>  <dbl>       <dbl> <dbl>
## 1 Austria Burgenland         8.65  8.08   5.44  10      6.85        3.26  8.77
## 2 Austria Lower Austria      8.38  8.7    5.53   9.89   6.94        3.49  8.83
## 3 Austria Vienna             8.4   5.28   5      9.43   6.12        1.79  7.04
## 4 Austria Carinthia          8.95  7.98   5.14  10      7.35        5.18  7.54
## 5 Austria Styria             8.52  8.28   5.21  10      7.26        4.59  7.8
## 6 Austria Upper Austria     8.04  9.1    5.33   9.89   7.52       4.22  8.21
## # … with 5 more variables: accessibility <dbl>, Housing <dbl>, Community <dbl>,
## #   life_satisfaction <dbl>, Overall <dbl>
# select the columns to work with

bli_cfa = bli %>%
  select(education, Income, Health, Environment,
         Housing, civic, accessibility)
bli_cov = cov(bli_cfa, use = "complete.obs")
```

So now we will set up the model and examine it further.

```
# Model
bli_1_model = '
better_life =~ education + Income + Health + Environment + Housing + civic + accessibility
'
# Fit the model
bli_1_fit = cfa(bli_1_model,
              sample.cov = bli_cov,
              sample.nobs = 400)

## summary
model_summary = summary(bli_1_fit)

# we will analyse fit measures and the parameter estimates separately.

blifitm = fitMeasures(bli_1_fit, fit.measure = c("chisq", "pvalue", "cfi", "rmsea", "gfi", "aic"))
# you can see that these measures indicate a poor fit of the model
# when we have one latent variable

bli_1_params = parameterEstimates(bli_1_fit, standardized = T) %>%
  select(lhs, rhs, est, ci.lower, ci.upper, std.lv, std.all)



# modification
modified_m = modificationIndices(bli_1_fit)
changes = modified_m %>%
  arrange(desc(abs(epc)))
```

```
blifitm
# we will now create two different latent variables
# one will load on safety, health, environment, housing, community, we will call it safe_community
# the other will load on education, jobs, income, we will call it secure_self

# so our new model

bli_2_model = '
   personal =~ d * Health + a * education + b * civic  + c * accessibility
   social =~   e * Environment + f * Housing + g * Income
   personal ~~ h * social
   
'
# fit the model
bli_2_fit = cfa(bli_2_model,
             sample.cov = bli_cov,
             sample.nobs = 400)
#  new summary
bli_2_summary = summary(bli_2_fit)

# fit
bli2_fit_m = fitMeasures(bli_2_fit, fit.measure = c("chisq", "pvalue", "cfi", "rmsea", "gfi", "aic"))
bli_2_params = parameterEstimates(bli_2_fit, standardized = T) %>%
 select(lhs, rhs, est, ci.lower, ci.upper, std.lv, std.all)
```

Now we will study a structural equation model. Here, we will regress personal well-being on social well-being and will test the hypothesis that social well-being influence a person’s personal sense of well being.

```
# Using the above model, we will examine the association between social satisfaction
# is personal satisfaction with life
bli_3_model = '
   personal =~ d * Health + a * education + b * civic  + c * accessibility
   social =~   e * Environment + f * Housing + g * Income
   personal ~ h * social
   
'
```

```
# fit the model
bli_3_fit = sem(bli_3_model,
        sample.cov = bli_cov,
        sample.nobs = 400)
#  new summary
bli_3_summary = summary(bli_3_fit)
## lavaan 0.6-8 ended normally after 47 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        15
##
##   Number of observations                           400
##
## Model Test User Model:
##
##   Test statistic                               479.474
##   Degrees of freedom                                13
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   personal =~
##     Health     (d)    1.000
##     education  (a)    2.816    0.413    6.817    0.000
##     civic      (b)    1.361    0.250    5.436    0.000
##     accessblty (c)    2.294    0.336    6.817    0.000
##   social =~
##     Environmnt (e)    1.000
##     Housing    (f)    2.081    0.242    8.594    0.000
##     Income     (g)    2.144    0.246    8.708    0.000
```

```
##
## Regressions:
##                Estimate  Std.Err  z-value  P(>|z|)
##   personal ~
##     social    (h)   0.628    0.116    5.417    0.000
##
## Variances:
##                Estimate  Std.Err  z-value  P(>|z|)
##    .Health          6.066    0.439   13.802    0.000
##    .education       3.269    0.376    8.684    0.000
##    .civic           7.920    0.580   13.658    0.000
##    .accessibility   2.159    0.249    8.661    0.000
##    .Environment     6.656    0.478   13.934    0.000
##    .Housing         2.835    0.286    9.929    0.000
##    .Income          0.495    0.217    2.285    0.022
##    .personal        0.330    0.100    3.281    0.001
##     social          1.435    0.335    4.280    0.000
# fit
bli3_fit_m = fitMeasures(bli_3_fit, fit.measure = c("chisq", "pvalue", "cfi", "rmsea", "gfi", "aic"))
bli_3_params = parameterEstimates(bli_3_fit, standardized = T) %>%
  select(lhs, rhs, est, ci.lower, ci.upper, std.lv, std.all)
```

Are personal growth a matter of nature or nurture?

Now that we have seen that evidence from 400 countries in Europe, Australia, and other parts of the world suggest that several aspects of personal well-being and better living are related to the social well-being, we can ask another related question: what are the relative genetic and environmental contributions to sense of psychological well-being. We will use data from Archontaki et.al. (2012) on the twin study of psychological well-being; for the purpose of demonstration, we will only use a small set of twin correlations, but you can read the entire study here (Archontaki, Lewis, and Bates 2013)

We will replicate a small subscale from the data presented in the article, Table 3,

Figure 7. Table data for the twin study we evaluate here

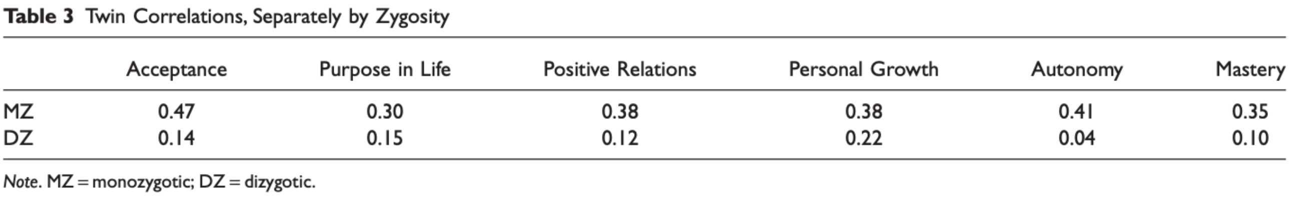


Table 3 Twin Correlations, Separately by Zygosity

	Acceptance	Purpose in Life	Positive Relations	Personal Growth	Autonomy	Mastery
MZ	0.47	0.30	0.38	0.38	0.41	0.35
DZ	0.14	0.15	0.12	0.22	0.04	0.10

Note. MZ = monozygotic; DZ = dizygotic.

From the paper we know that there were 240 monozygotic twin pairs (MZ pairs), and 597 dizgyotic twin pairs (DZ pairs).

We will analyse here the scale “personal growth” for this paper. We will set up the data for analysis as follows:

```
# for mz twins, we do:
mz = lav_matrix_lower2full(c(1.00,
                             0.38, 1.00))
rownames(mz) = colnames(mz) = c("T1", "T2")

# Explanation:
# lav_matrix_lower2full() converts a lower matrix to full matrix
# we are using here a correlation matrix
# 1.00 is the standardised variance, hence 1.00
# 0.38 is the correlation
# T1 and T2 are the twin pairs

# for dz twins
dz = lav_matrix_lower2full(c(1.00,
                             0.12, 1.00))
rownames(dz) = colnames(dz) = c("T1", "T2")

# we will now combine the sample size and correlation matrices

cormat = list(mz = mz, dz = dz)
sampsize = list(mz = 240, dz = 597)

cormat
## $mz
##      T1   T2
## T1 1.00 0.38
## T2 0.38 1.00
##
## $dz
##      T1   T2
## T1 1.00 0.12
## T2 0.12 1.00
```

Now that we have set up the data for this study, we will run the models and evaluate them as follows. Here we will evaluate an ACE model, where we will test the path coefficients and heritability estimates for a model where we will test additive genetic effecs (As), common environmental effects (Cs), and unique environmental effects (Es). Note that as correlation of Cs on both MZ and DZ twins is 1.0 (that is they share the SAME environment), the impact of a shared or common environment on their phenotype is likely to be very low, so we will set it at more than 0.001

```
ace_model <- '
  A1 =~ NA*T1 + c(a,a)*T1 + c(0.5, 0.5)*T1
  A2 =~ NA*T2 + c(a,a)*T2 + c(0.5, 0.5)*T2
  C1 =~ NA*T1 + c(c,c)*T1
  C2 =~ NA*T2 + c(c,c)*T2

  # Variances

  A1 ~~ 1*A1
  A2 ~~ 1*A2
  C1 ~~ 1*C1
  C2 ~~ 1*C2
  T1 ~~ c(e,e)*T1
  T2 ~~ c(e,e)*T2

  # Covariances

  A1 ~~ c(1, 0.5)*A2
  A1 ~~ 0 * C1 + 0 * C2
  A2 ~~ 0 * C1 + 0 * C2
  C1 ~~ c(1, 1) * C2

  c > 0.001
'
# Now let's run the model

ace_fit = cfa(ace_model,
          sample.cov = cormat,
          sample.nobs = sampsize)

# summary
model_sum = summary(ace_fit)

# parameter estimates

model_est = parameterEstimates(ace_fit)
```

```
# fit measures
ace_measures = fitMeasures(ace_fit,
                           fit.measures = c("chisq", "gfi", "rmsea"))



a_squared = 0.578 * 0.578
# a_squared = 33.4% of heritability is explained by genes
e_squared = 0.664 * 0.664
# e_squared = 44% of heritability explained by unique environmental factors


model_est
```

Are people getting happier over time?

Our final analysis will be based on data where Gallup World Poll measured the proportion of people in various countries where they said they were happy over time, see for more information here. We obtained the data and you can download a copy of the data from here. The data were collected between 1984 through 2014. We will study the longitudinal pattern using SEM (growth model) and study the baseline percentage from where it began and slope of the growth. The data were gathered every five years.

```
# get the data

happiness_data = read_csv("https://raw.githubusercontent.com/arinbasu/2021_04_sem_datasets/main/share_happy.csv")

# Preprocess the data to rename variables, etc

happiness_data %>%
  head(5)
## # A tibble: 5 x 4
##   Entity  Code   Year `Share of people who are happy (World Value Survey 2014)`
##   <chr>   <chr> <dbl>                                                     <dbl>
## 1 Albania ALB    1998                                                      33.4
## 2 Albania ALB    2004                                                      58.8
## 3 Algeria DZA    2004                                                      80.7
## 4 Algeria DZA    2014                                                      79.9
## 5 Andorra AND    2009                                                      92.9
number_of_records = length(happiness_data$Code)
number_of_records # 237 countries/regions
## [1] 237
```

```
# let's clean up the data
happiness = happiness_data %>%
  rename(country = 'Entity',
         year = 'Year',
         pct_happy = 'Share of people who are happy (World Value Survey 2014)') %>%
  select(country, year, pct_happy )

# let's get a sense of years

years = happiness %>%
  count(year)

happiness %>%
  head()
## # A tibble: 6 x 3
##   country    year pct_happy
##   <chr>     <dbl>     <dbl>
## 1 Albania    1998      33.4
## 2 Albania    2004      58.8
## 3 Algeria    2004      80.7
## 4 Algeria    2014      79.9
## 5 Andorra    2009      92.9
## 6 Argentina  1984      78.6
# years # shows only 7 records in 1984 so we will select years 1993 - 2014
# they were recorded every 5 years
# we will remove 1984 from the data
# then we will pivot the data wider

happiness1 = happiness %>%
  filter(year > 1984) %>%
  pivot_wider(names_from = year,
              values_from = pct_happy) %>%
  rename(wave1 = `1993`,
         wave2 = `1998`,
         wave3 = `2004`,
         wave4 = `2009`,
         wave5 = `2014`) %>%
  select(country, wave1, wave2, wave3, wave4, wave5)




# happiness1 %>% head() shows us the data in desired format
# Now we prepare the means and covariance matrix
happy_mean = c(mean(happiness1$wave1, na.rm = T),
               mean(happiness1$wave2, na.rm = T),
               mean(happiness1$wave3, na.rm = T),
               mean(happiness1$wave4, na.rm = T),
               mean(happiness1$wave5, na.rm = T))

# happy_mean

# covariance matrix of the five time points


happy_cov = happiness1 %>%
  select(wave1, wave2, wave3, wave4, wave5) %>%
  cov(use = "complete.obs")
```

Now that we have the means and the covariance matrix, we are ready to code and find out whether over time, the proportion of people who report they are happy in the happiness surveys are increasing. Besides, we would learn the baseline levels of proportions of people who were happy and the rate of change of that state. Here are the findings and the code:

```
# build the model (unconstrained, simple model)
happiness_model = '
    i =~ 1 * wave1 + 1 * wave2 + 1 * wave3 + 1 * wave4 + 1 * wave5
    s =~ 0 * wave1 + 5 * wave2 + 10 * wave3 + 15 * wave4 + 20 * wave5
    wave1 ~~ r * wave1
    wave2 ~~ r * wave2
    wave3 ~~ r * wave3
    wave4 ~~ r * wave4
    wave5 ~~ r * wave5
'

# then we fit the model
```

```
happiness_fit = growth(happiness_model,
                       sample.cov = happy_cov,
                       sample.mean = happy_mean,
                       sample.nobs = 300)

sum_happy = summary(happiness_fit)
## lavaan 0.6-8 ended normally after 89 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        10
##   Number of equality constraints                     4
##
##   Number of observations                           300
##
## Model Test User Model:
##
##   Test statistic                              1439.300
##   Degrees of freedom                                14
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   i =~
##     wave1             1.000
##     wave2             1.000
##     wave3             1.000
##     wave4             1.000
##     wave5             1.000
##   s =~
##     wave1             0.000
##    wave2          5.000
##    wave3         10.000
##    wave4         15.000
##    wave5         20.000
##
## Covariances:
##                 Estimate  Std.Err  z-value  P(>|z|)
##   i ~~
##    s              -0.938    0.172   -5.469    0.000
##
## Intercepts:
##                 Estimate  Std.Err  z-value  P(>|z|)
##   .wave1          0.000
##   .wave2          0.000
##   .wave3          0.000
##   .wave4          0.000
##   .wave5          0.000
##    i             72.631    0.328  221.480    0.000
##    s              0.614    0.024   25.847    0.000
##
## Variances:
##                 Estimate  Std.Err  z-value  P(>|z|)
##   .wave1    (r)   19.522    0.920   21.213    0.000
##   .wave2    (r)   19.522    0.920   21.213    0.000
##   .wave3    (r)   19.522    0.920   21.213    0.000
##   .wave4    (r)   19.522    0.920   21.213    0.000
##   .wave5    (r)   19.522    0.920   21.213    0.000
##    i             20.549    2.691    7.635    0.000
##    s              0.091    0.014    6.380    0.000
param_happy = parameterEstimates(happiness_fit)
fit_happy = fitMeasures(happiness_fit)

#modificationIndices(happiness_fit)
```

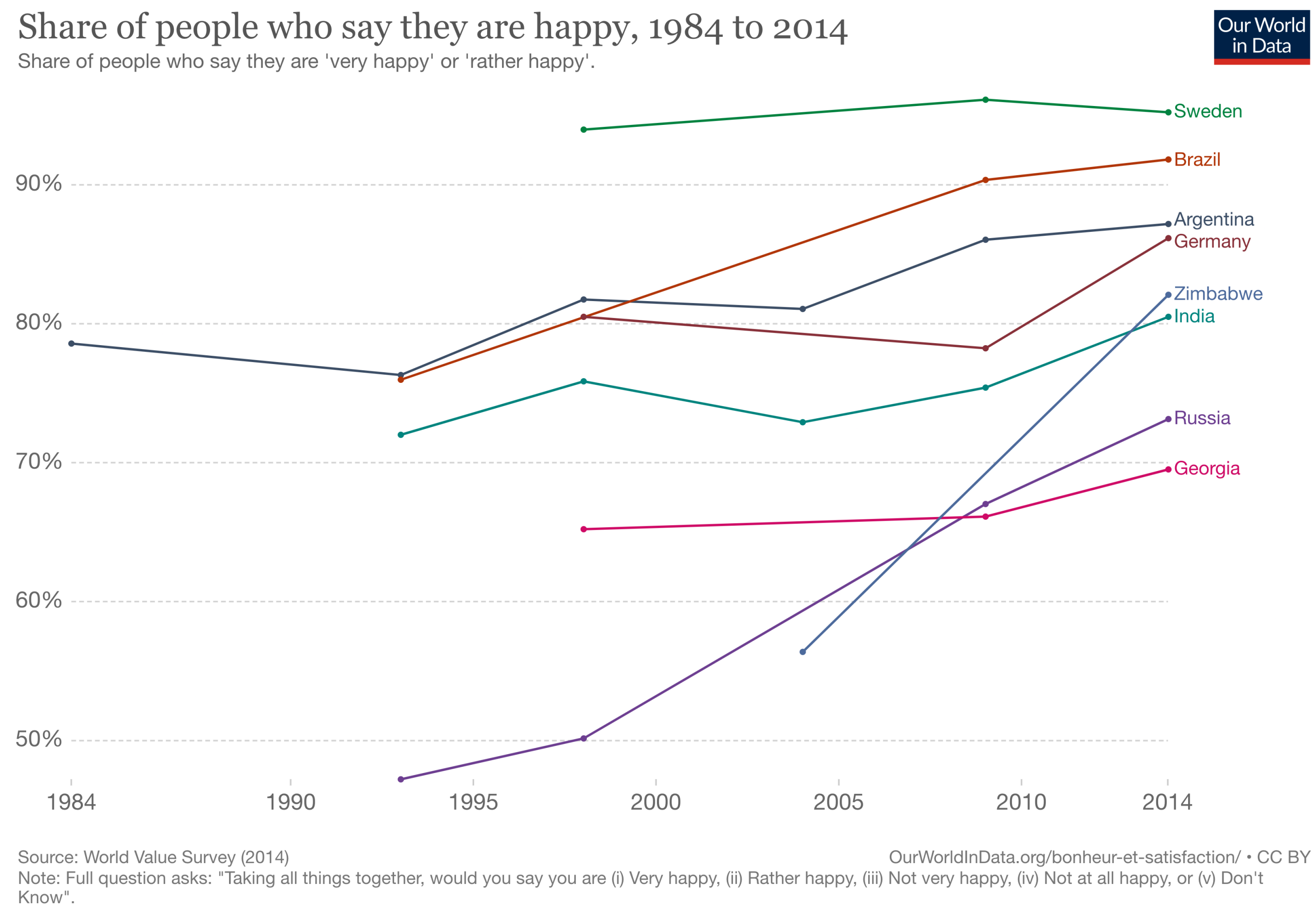
As this analysis suggests:

- About 72% people over the world population reported that they were happy or satisfied in 1993
- Over time, more people tended to report they were happy over subsequent surveys

- Countries that started with lower percentage of their people reporting in surveys they were happy, had a faster rate of growth over subsequent waves

While our analysis borne this, you can also view the results reported graphically on the ourworldindata website as shown in the following figure:

Figure 8. Trend in reporting of happiness


Share of people who say they are happy, 1984 to 2014

Share of people who say they are 'very happy' or 'rather happy'.

Source: World Value Survey (2014)

OurWorldInData.org/bonheur-et-satisfaction/ • CC BY

Note: Full question asks: "Taking all things together, would you say you are (i) Very happy, (ii) Rather happy, (iii) Not very happy, (iv) Not at all happy, or (v) Don't Know".

Shared they were happy across time span, Ourworld in data as the source

As you can see in this figure, countries such as Russia and Zimbabwe, that started at the lower end of the graphs had steeper upward slopes reporting happiness over time, while countries that started with higher proportion of people as happy tended to have slower trajectory of growth reporting happy people in subsequent waves. Overall, people do seem to get happier over time, at least till 2014 starting from 1993.

Summary

In this tutorial, we provided a brief roundup of using structural equation modelling as an analytical strategy. Structural equation modelling is a mix of regression and factor analysis, and as you may have seen, this can be used for validation of surveys and questionnaires, reducing data, linear regression modelling, analysis of group differences, twin data analyses, and growth curve modelling. We used worked out examples to show where you can start with simple measurement models, then transform or modify them to test how well they fit the data, and in the same modelling strategy, you can test regression as in testing structural models. Longitudinal studies and twin studies are different from measurement models and structural models in the sense that they need theoretical understandings to fix and constrain parameter estimates, and also for longitudinal studies, our goal is to find the parameters for latent intercepts and slopes, as we constrain the parameters that explain variance of the time bound values of variables. While SEMs are powerful strategies, one needs to be careful to deal with missing values, extreme outliers, non-normal distribution of the variables, and categorical variables. Lately, SEM strategies are being used to model non-parametric equations as in (Structural Causal Models) [https://www.causalflows.com/structural-causal-models/]. We have also not touched the issues of model specification, interpretation of modification indices, and sample size estimation. In subsequent extensions to this inflammation, we will discuss these issues.

References

Archontaki, Despina, Gary J Lewis, and Timothy C Bates. 2013. “Genetic Influences on p Sychological w Ell-b Eing: A Nationally Representative Twin Study.” *Journal of Personality* 81 (2): 221–30.

Denis, D. 2021. “ORIGINS OF PATH ANALYSIS: Causal Modeling and the Origins of Path Analysis.” https://theoryandscience.icaap.org/content/vol7.1/denis.html.

Meredith, William. 1993. “Measurement Invariance, Factor Analysis and Factorial Invariance.” *Psychometrika* 58 (4): 525–43.

Neale, MCCL, and Lon R Cardon. 2013. *Methodology for Genetic Studies of Twins and Families*. Vol. 67. Springer Science & Business Media.

Rohrer, Julia M. 2018. “Thinking Clearly about Correlations and Causation: Graphical Causal Models for Observational Data.” *Advances in Methods and Practices in Psychological Science* 1 (1): 27–42.

Williams, Thomas C, Cathrine C Bach, Niels B Matthiesen, Tine B Henriksen, and Luigi Gagliardi. 2018. “Directed Acyclic Graphs: A Tool for Causal Studies in Paediatrics.” *Pediatric Research* 84 (4): 487–93.