Cause and Correlation

in Biology

A User's Guide to Path Analysis, Structural Equations and Causal Inference with R

Third Edition

Bill Shipley

Université de Sherbrooke, Canada

À la prochaine génération : Clara, Maëva, Théo, Sophie et la suite

Contents

Preface

This book describes a collection of statistical methods for testing, or developing, causal hypotheses using observational data, but it is not a statistics text. It describes the logical and philosophical relationships between causality and probability distributions, but it is definitely not a book about the philosophy of statistics. Instead, it is a user's guide, written for biologists, whose purpose is to allow practicing biologists to make use of these important developments in statistics when causal questions cannot be answered with randomised experiments.

I have written the book while assuming that you have no previous training in these methods. If you have mastered an introductory university course in statistics and have managed to hold on to the basic notions of sampling, hypothesis testing, and linear models, then you should be able to understand the material in this book. A few sections discuss mixed model regression and, although I give a very basic introduction to this type of regression, you will need some basic knowledge of mixed models to completely follow the text. If you don't have this knowledge, then you can simply skip these sections without preventing you from following the larger discussion. I recommend that you read each chapter through in it entirely, even if you do not feel that you have mastered everything, and then go back if needed. This will at least give you a general feeling for the goals and vocabulary of each chapter before concentrating on the details. A few sections are more theoretical but I always clearly indicate these sections and how to skip them if you do not want this level of detail.

I had two motives, one positive and one more selfish, when writing the first edition of this book, which appeared in the year when the new millennium began. The positive motive was to provide a detailed introduction to structural equations modelling that was specifically aimed at practicing biologists, since structural equations modelling was almost completely unknown in this discipline. The more selfish motive was to provide a detailed *justification* for these methods aimed at biologists. You see, I was frustrated. My research manuscripts in plant ecology that used these methods to test causal hypotheses using only observational data were being rejected by reviewers who viewed these methods as the statistical equivalents of conjurer's tricks. Dispelling such scepticism required explaining all of the logical and mathematical details (many of which were quite new) that link the notion of causality to probability distributions. This was

impossible to do in such empirical research papers in plant ecology. I therefore decided that it was necessary to write a complete book, written for biologists, that lays out the full argument.

The second edition appeared sixteen years later (2016). The situation had changed dramatically during the intervening sixteen years. The use of structural equations modelling in the fields of ecology and evolution had become, if not mainstream, then at least accepted. I hope that the first edition of this book, as well as the very good book by Jim Grace (Grace 2006), contributed to this change in attitude. Although the second edition contained some new developments in the methodology, the main addition of the second edition was the inclusion of the R programming language. The first edition had not included any information on computer programs because the only computer programs for structural equation modelling before 2000 were commercial ones, and I did not want to be a salesman. However, the free R statistical program had become so ubiquitous for statistical analysis by biologists by 2016 that I could include it in the second edition. This was a major improvement since a user's guide without any computer code is clearly deficient.

This third edition now includes extensive use of the R statistical environment and computer package and introduces a new R package (pwSEM) for piecewise structural equation modelling that I have created specifically for this book. It corrects some errors in the existing piecewiseSEM package (Lefcheck 2016) and generalizes piecewise SEM to include implicit latent variables and nonlinear functional relationships. Every chapter in this third edition has been completely rewritten and also includes new chapters and new results. In these respects it is almost a new book rather than simply a new edition.

I would like to acknowledge all of the people who have contributed to the various editions of this book, but I cannot do this. There are so many people that any list will invariably forget someone. I can only name some of the most influential people: Robert van Hulst, the late Paul Keddy and the late Robert Peters, Martin Lechowicz and Clark Glymour. Bob Douma has collaborated with me for several years in extending the generality of piecewise SEM and several of his contributions are found in Chapter 6. Bob also commented on most of the chapters. Thank you all.

1

Cause from correlation?

1.1 The shadow's cause

The *Wayang Kulit* is an ancient theatrical art that is practised in Indonesia and throughout much of southeast Asia. The stories are often about battles between good and evil, as told in the great Hindu epics. The audience does not see actors, or even puppets, on the stage. Instead, they view the shadows of puppets projected onto a canvas screen. Behind the screen is a light. The puppet master creates the action by manipulating the puppets and props so that they will intercept the light and cast shadows. As these shadows dance across the screen the audience must deduce the story from these two-dimensional projections of the hidden three-dimensional objects. Shadows, however, can be ambiguous. In order to infer the hidden three-dimensional action, the shadows must be detailed, with sharp contours and they must be placed in context.

Biologists are unwitting participants in Nature's Shadow Play. These shadows are cast when the hidden causal processes in nature are intercepted by our measurements. Like the audience at the *Wayang Kulit*, the biologist cannot simply peak behind the screen and directly observe the actual causal processes. All that can be directly observed are the consequences of these processes in the form of statistical shadows of association and independence in the data. Like shadows, these correlational patterns are incomplete, and potentially ambiguous, projections of the original causal processes. Like shadows, we can infer much about the underlying causal processes if we can learn to study their details, sharpen their contours, and especially if we can study them in context.

Unfortunately, unlike the Puppet Master in a *Wayang Kulit* who takes care to cast informative shadows, Nature is indifferent to the correlational shadows that it casts. This is the main reason

why researchers go to such extraordinary lengths to randomise treatment allocations and to control variables. These methods, when they can be properly done, simplify the correlational shadows to manageable patterns that can be more easily mapped onto the underlying causal processes.

It is uncomfortably true, although rarely admitted in statistics texts, that many important areas of science are stubbornly impervious to experimental designs based on randomisation of treatments to experimental units . Historically, the response to this embarrassing problem has been to either ignore it or to banish the very notion of causality from the language and to claim that the shadows dancing on the screen are all that exist. Ignoring a problem doesn't make it go away and defining a problem out of existence doesn't make it so. We need to know what we can safely infer about causes from their observational shadows, what we can't infer, and the degree of ambiguity that remains.

I wrote this book to introduce biologists to some recent, and intellectually elegant, methods that help in the difficult task of inferring causes from observational data. Some of these methods, for instance Structural Equation Modelling (SEM), are well known to researchers in other fields although less known to biologists. Other methods, for instance those based on causal graphs, are unknown to almost everyone but a small community of researchers. These methods help both to test pre-specified causal hypotheses and to help discover potentially useful hypotheses concerning causal structures.

This book has three objectives. First, it was written to convince biologists that inferring causes without randomised experiments is possible. If you are a typical reader, then you are already more than a little sceptical. For this reason, I devote the first two chapters to explaining why these methods are justified. The second objective is to produce a user's guide, devoid of as much jargon as possible, which explains how to use and interpret these methods. To do this, I will explain, where appropriate, how to do this using the R[1] open source statistical program. The third objective is to exemplify these methods using biological examples, taken mostly from my own research and from that of my colleagues and students, so that practicing biologists can apply these methods to their own research.

[1] http://www.r-project.org/

I came to these ideas unwillingly. In fact, I find myself in the embarrassing position of having publicly claimed that inferring causes without randomisation and experimental control is probably impossible and, if possible, is not to be recommended (1990, 1991). I had expressed such an opinion in the context of determining how the different traits of an organism interact as a causal system. I will return to this theme repeatedly in this book because it is so basic to biology[2] and yet is completely unamenable to the one method that most modern biologists and statisticians would accept as providing convincing evidence of a causal relationship: the randomised experiment. For instance, imagine that you want to test if an increase in the RUBISCO enzyme in a leaf increases photosynthetic rate. You randomly assign different concentrations of RUBISCO (say, the natural level in the control group and twice the natural level in the treatment group) to each experimental unit (a leaf or a plant). In order to do this, you must directly manipulate only the concentration of RUBISCO; you cannot only induce a change in the concentration of RUBISCO by manipulating something else. If you do, then you cannot know if the result is due to increased RUBISCO or is due to a change in the other traits that also changed in the treatment group. For instance, you cannot use transgenic lines for the treatment group that overexpress the RbcL/RbcS genes because the overexpression of these genes might also affect the expression of some other set of genes that are not related to RUBISCO concentration but that do increase photosynthesis. You must inject the added RUBISCO directly into the aqueous stroma of each of the chloroplasts in the leaf, without damaging the functioning of the chloroplast or the plant cell in which the chloroplast resides. No one has ever done such an experiment. In practice, our knowledge of the causal relationship between RUBISCO and photosynthesis (and the larger Calvin cycle of photosynthesis) has been obtained using "controlled", not randomized, experiments. The distinction between the two will be explained a bit later.

However, even as I advanced the arguments in Shipley and Peters (1990, 1991), I was dissatisfied with the consequences that such arguments entailed. I was also uncomfortably aware of the logical weakness of such arguments: the fact that I did not know of any provably correct way of inferring causation without the randomised experiment does not mean that such a method

[2] This is also the problem that inspired Sewall Wright, one the most influential evolutionary biologists of the twentieth century, the inventor of path analysis, and the intellectual grandparent of the methods described in this book. The history of path analysis is explored in more detail in Chapter 3.

cannot exist. In my defence, and beyond a plea to the folly of youth, I was saying nothing original; such an opinion was (and still is) the position of most statisticians and biologists. This view is summed up in the mantra[3] that is learnt by almost every student who has ever taken an elementary course in statistics: *correlation does not imply causation*.

In fact, with few exceptions[4], correlation does imply causation. If we observe a systematic relationship between two variables, and we have ruled out the likelihood that this is simply due to a random coincidence, then *something* must be causing it. When the audience at a Malay shadow theatre sees a solid round shadow on the screen, they know that some three-dimensional object has cast it although they may not know if the object is a ball or a rice bowl in profile. A more accurate sound bite for introductory statistics would be that a simple correlation implies an *unresolved* causal structure since we cannot know which is the cause, which is the effect, or if both are common effects of other unmeasured variables.

Although correlation implies an unresolved causal structure the reverse is not true: causation implies a completely resolved correlational structure. By this I mean that once a causal structure has been proposed, the complete pattern of correlation and partial correlation is unambiguously fixed. This point is developed more precisely in Chapter 2 but is so central to this book that it deserves repeating: the causal relationships between variables determine the correlational relationships between them. Just as the shape of an object fixes the shape of its shadow, the patterns of direct and indirect causation fix the correlational "shadows" that we observe in observational data. The causal processes generating our observed data impose constraints on the patterns of correlation that such data display. This is the central insight underlying the methods described in this book.

The term "correlation" evokes the notion of a probabilistic association between random variables. One reason why most statisticians rarely speak of causation, except to distance themselves from it, is because there did not exist, until very recently, any rigorous translation between the language of causality (however defined) and the language of probability distributions (Pearl 1988, Verma and Pearl 1988). It is therefore necessary to link causation to

[3] It would be more precise (but a less catchy soundbite) to say that "statistical dependence" does not imply causation, since "correlation" is associated with Pearson's correlation coefficient, which is a more restricted form of dependence.
[4] It could be argued that variables that covary only because they are time-ordered have no causal basis.

probability distributions in a very precise way. Such rigorous logical links have now been forged. It is now possible to give mathematical proofs that specify the correlational pattern that must exist given a causal structure. These proofs also allow us to specify the class of causal structures that must include the causal structure that generates a given correlational pattern. The methods described in this book are justified by these proofs. Since my objective is to describe these methods and show how they can help biologists in practical applications, I won't present these proofs but will direct the interested reader to the relevant primary literature.

Another reason why some prefer to speak of associations rather than causes is perhaps because causation is seen as a metaphysical notion that is best left to philosophers. In fact, even philosophers of science cannot agree on what constitutes a "cause". I have no formal training in the philosophy of science and am neither able nor inclined to advance such a debate. This is not to say that philosophers of science have nothing useful to contribute. Where directly relevant I will outline the development of philosophical investigations into the notion of "causality" and place these ideas into the context of the methods that I will describe. However, I won't insist on any formal definition of "cause" and will even admit that I have never seen anything in the life sciences that resembles the "necessary and sufficient" conditions for causation that are so beloved of logicians.

You probably already have your own intuitive understanding of the term "cause". I won't take it away from you although, I hope, it will be more refined after reading this book. When I first came across the idea that one can study causes without defining them, I almost stopped reading the book (Glymour et al. 1987). I can advance three reasons why you should not follow through on this same impulse. First, and most important, the methods described here are not logically dependent on any particular definition of causality. The most basic assumption that these methods require is that causal relationships exist in relation to the phenomena that are studied by biologists[5]. The second reason why you should continue reading even if you are sceptical is more practical and, admittedly, rhetorical: scientists commonly deal with notions whose meaning is somewhat ambiguous. Biologists are even more promiscuous than most with one notion that can still raise the blood pressure of philosophers and statisticians. This notion is "probability",

[5] Perhaps quantum physics does not need such an assumption. I will leave this question to people better qualified than I. The world of biology does not operate at the quantum level.

for which there are frequentist , objective Bayesian and subjective Bayesian definitions. In the 1920's von Mises is reported to have said: "today, probability theory is not a mathematical science" (Rao 1984). Mayo (1996) gives the following description of the present degree of consensus concerning the meaning of probability: "Not only was there the controversy raging between the Bayesians and the error statisticians, but philosophers of statistics of all stripes were full of criticisms of Neyman-Pearson error statistics…"; here, Mayo uses "error statistics" to mean "frequentist" statistics. The fact that those best in a position to define "probability" cannot agree on one does not prevent biologists from effectively using probabilities, significance levels, confidence intervals, and the other paraphernalia of modern statistics[6]. In fact, insisting of such an agreement would mean that modern statistics could not even have begun.

The third reason why you should continue reading, even if you are sceptical, is eminently practical. Although the randomised experiment is inferentially superior to the methods described in this book when randomisation can be properly applied, it cannot be properly applied to many (perhaps most) research questions asked by biologists. Unless you are willing to simply deny that causality is a meaningful concept then you will need some way of studying causal relationships when randomised experiments cannot be performed. Maintain your scepticism if you wish but grant me the benefit of your doubt. A healthy scepticism while in a car dealership will keep you from being cheated. An unhealthy scepticism might prevent you from obtaining reliable transportation.

I said that the methods in this book are not logically dependent on any particular definition of causality. Rather than *defining* causality, the approach is to *axiomise* causality (Spirtes et al. 1993, Pearl 2009, Pearl and Mackenzie 2018). In other words, one begins by determining those attributes that scientists view as necessary for a relationship to be considered "causal" and then develop a formal mathematical language that is based on such attributes. First, these relationships must be *transitive*: if A causes B and B causes C, then it must also be true that A causes C. Second, such relationships must be "local"; the technical term for this is that the relationships must obey the *Markov condition*, of which there are local and global versions. This is described in more detail in Chapter 2 but can be intuitively understood to mean that events are

[6] The perceptive reader will note that I have now compounded my problems. Not only do I propose to deal with one imperfectly defined notion – causality – but I will do it with reference to another imperfectly defined notion: a probability distribution.

caused only by their proximate causes. Thus, if event A causes event C *only* through its effect of an intermediate event B (A→B→C), then the causal influence of A on C is blocked if event B is prevented from responding to A. Third, these relationships must be *irreflexive*: an event cannot cause itself. This is not to say that every event must be causally explained; to argue in this way would lead us directly into the paradox of infinite regress. Every causal explanation in science includes events that are accepted (measured, observed…) without being derived from previous events[7]. Finally, these relationships must be *asymmetric*: if A is a cause of B, then B cannot simultaneously be a cause of A[8]. In my experience, scientists generally accept these four properties. In fact, so long as I avoid asking for definitions, I find that there is a large degree of agreement between scientists on whether any particular relationship should be considered causal or not. It might be of some comfort to empirically trained biologists that the methods described in this book are based on an almost empirical approach to causality. This is because deductive definitions of philosophers are replaced with attributes that working scientists have historically judged to be necessary for a relationship to be causal. However, this change of emphasis is, by itself, of little use.

Next, we require a new mathematical language that can express and manipulate these causal relationships. This mathematical language is that of directed graphs[9] (Pearl 1988, Spirtes et al. 1993). Even this new mathematical language is not enough to be of practical use. Since, in the end, we wish to infer causal relationships from correlational data, we need a logically rigorous way of translating between the causal relationships encoding in directed graphs and the correlational relationships encoded in probability theory. Each of these requirements can now be fulfilled.

---

[7] The paradox of infinite regress is sometimes "solved" by simply declaring a First Cause: that which causes but which has no cause. This trick is hardly convincing because, if we are allowed to invent such things by fiat, then we can declare them anywhere in the causal chain. The antiquity of this paradox can be seen in the first sentence of the first verse of Genesis: "*In the beginning God created the heavens and the earth*." According to the Confraternity Text of the Holy Bible, the Hebrew word which has been translated as "created" was used only with reference to divine creation and meant "to create out of nothing".

[8] This does not exclude feedback loops so long as we understand these to be dynamic in nature: A causes B at time t, B causes A at time t+Δt, and so on. This is discussed more fully in Chapter 2.

[9] Biologists will find it ironic that this graphical language was actually proposed by Sewall Wright in 1921, one of the most influential evolutionary biologists of the twentieth century, but his insight was largely ignored. This history is explored in Chapter 3.

1.2 Fisher's genius and the randomised experiment

Since this book deals with causal inference from observational data, we should first look more closely at how biologists infer causes from experimental data. What is it about these experimental methods that allow scientists to confidently speak about causes? What is it about inferring causality from non-experimental data that make them squirm in their chairs? I distinguish between two basic types of experiments: the controlled experiment and the randomised experiment. Although the controlled experiment takes historical precedence, the randomised experiment takes precedence in the strength of its causal inferences.

Fisher[10] first laid out the principles of the randomised experiment in (Fisher 1926) and again, more formally, in his classic *Design of Experiments* (Fisher 1935). Since he developed many of his statistical methods in the context of agronomy, let's consider a typical randomised experiment designed to determine if the addition of a nitrogen-based fertiliser can cause an increase in the seed yield of a particular variety of wheat. A field is divided into 30 plots of soil. The treatment variable consists of the fertiliser, which is applied at either 0 or 20 kg/hectare. For each plot we place a small piece of paper in a hat. Fifteen of the pieces of paper have a "0" written on them and the other fifteen have a "20" written on them. After thoroughly mixing the pieces of paper, we randomly draw one for each plot to determine the treatment level that each plot is to receive. In other words, we randomly allocate the value of the experimental manipulation (adding either 0 or 20 kg/hectare) to each of the 30 observational units (the 30 plots). After applying the appropriate level of fertiliser independently to each plot, we plant the wheat seeds and then make no further manipulations until harvest day, at which time we weigh the seed that is harvested from each plot.

The seed weight per plot is normally distributed within each of the two groups (the treatment and control groups). Those plots receiving no fertiliser (the control plots) produce 55 g of seed with a standard error of 6. Those plots receiving 20 kg/hectare of fertiliser (the treatment plots) produce 80 g of seed with a standard error of 6. The increase in seed yield in the treatment plots,

[10] Sir Ronald A. Fisher (1890-1962) was chief statistician at the Rothamsted Agricultural Station. He was later Galton Professor at the University of London and Professor of Genetics at the University of Cambridge.

relative to the control plots, is d=80 – 55 = 25g. Excluding the possibility that a very rare random event has occurred (with a probability of approximately once in $5x10^8$ events), we have very good evidence that an increase in wheat yield is *associated*, or *correlated*, with the increased yield of the wheat.

Where did my claim come from that the results of this experiment would happen by chance approximately once in $5x10^8$ events, assuming that the fertiliser had no effect on seed yield,? If we draw two random samples of size N from a single very large population of elements and measure some variable property (X) of these elements, then we can calculate the difference in the average value of X between the two random samples ($d = \bar{X}_T - \bar{X}_C$). We know, as an empirical fact, that the value of d will change each time we repeat this sampling exercise even though the population from which the samples were drawn doesn't change. We also know, as an empirical fact, that if we repeat this sampling exercise in exactly the same way a large number of times, the relative frequency of these different values of d will become more and more stable as we repeat the sampling exercise more and more times. This stable pattern of relative frequency is called a "sampling distribution" that can often be mathematically described by an equation. If we further assume that the distribution of our variable property (X) follows a normal distribution, then the sampling distribution of our difference (d) follows a mathematical function called Student's t-distribution (Student 1908)[11]. By randomising the treatment allocation to the sample observational units (quadrats), we therefore ensure that our sample follows a sampling distribution. Because the seed yields follow a normal distribution within each group, we know that our sampling distribution is described by Student's equation and this allows us to calculate the probability of observing this result ($\sim 5x10^{-8}$) by chance in a sample of 30 plots if, in reality, there was no real difference in seed yield when wheat plants are growing in soil that have, or haven't, received 20 kg/hectare of this type of fertilizer. This helps us to distinguish between chance associations and systematic ones. Since one error that a researcher can make is to confuse a real difference with a difference due to sampling fluctuations, the sampling distribution allows us to calculate the probability of committing such an error[12].

[11] The actual author was William Sealy Gosset, but he published his result using the pseudonym "Student" because he worked for the Guinness Brewery in Dublin and that company had a strict policy prohibiting employees from publishing research, presumably to protect trade secrets.

[12] It is for this reason that (Mayo 1996) calls such frequency-based statistical tests "error probes".

However, the random allocation of treatments to experimental units does more than simply allow us to use a known sampling distribution to distinguish between a chance association and a real one. Fisher, and many other statisticians[13] (Kempthorpe 1979, Kendall and Stuart 1983) went further by claiming that the process of randomisation allows us to differentiate between associations due to causal effects of the treatment and associations due to some variable that is a common cause both of the treatment and response variables. What allows us to move so confidently from this conclusion about an *association* between fertiliser addition and increased seed yield to the claim that the added fertiliser actually *causes* the increased yield?

Given that two variables (X and Y) are associated, there can be only three elementary, but not mutually exclusive, causal explanations: either X causes Y ($X \rightarrow Y$), Y causes X ($X \leftarrow Y$), or there are some other causes (perhaps unknown to us) that are common to both X and Y ($X \leftarrow ? \rightarrow Y$). Here, I am making no distinctions between "direct" and "indirect" causes. I will argue in Chapter 2 that such terms have no meaning except relative to the other variables in the causal explanation. Remembering that transitivity is a property of causes, to say that X causes Y does not exclude the possibility that there are intervening variables ($X \rightarrow Z_1 \rightarrow Z_2 \rightarrow \ldots \rightarrow Y$) in the causal chain between them. We can confidently exclude the possibility that the seed produced by the wheat at the end of the summer caused the amount of fertiliser that was added in the spring. First, we already know the only cause of the amount of fertiliser that was added to any given plot: the number written on the piece of paper that was drawn from the hat. Second, the fertiliser was added before the wheat plants began to produce seed; unless your meaning of "cause" is very peculiar[14], you will agree that causal relationships cannot travel backwards in time. What allows us to exclude the possibility that the observed association between fertiliser addition and seed yield is due to some unrecognised common cause of both? This was Fisher's genius; the treatments were randomly assigned to the experimental units (i.e. the plots with their associated wheat plants). By definition, such a random process ensures that the order in which the pieces of paper are chosen, and therefore the amount of fertilizer (0 or 20 kg/hectare) that was added to

[13] "Only when the treatments in the experiment are applied by the experimenter using the full randomisation procedure is the chain of inductive inference sound; it is only under these circumstances that the experimenter can attribute whatever effect he observes to the treatment and to the treatment only." (Kempthorpe 1979)

[14] I know that some physicists are considering the possibility of causality that moves backwards in time at the quantum level (Pegg 2008) but it is still just speculation without experimental evidence. Evaluating this literature is above my pay grade. In any case, biology does not operate at the quantum level.

each plot, is causally independent of any attributes of the plot, its soil, or the plant at the moment of randomisation.

Let's retrace the logical steps. We began by asserting that if there was a causal relationship between fertiliser addition and seed yield, then there would also be a systematic relationship between these two variables in our data: *Causation implies correlation*. When we observe a systematic relationship that cannot reasonably be attributed to sampling fluctuations, we conclude that there was some causal mechanism responsible for this association. Correlation does not necessarily imply a causal relationship from the fertiliser addition to the seed yield, but it does imply *some* causal relationship that is responsible for this association. There are only three such elementary causal relationships and the process of randomisation has excluded two of them. We are left with the overwhelming likelihood that the fertiliser addition caused the increased seed yield. We cannot categorically exclude the two alternate causal explanations since it is always possible that we were incredibly unlucky. Perhaps the random allocations resulted, by an incredibly small chance, in those plots that received the 20 kg/hectare of fertiliser having soil with a higher moisture-holding capacity or some other attribute that actually caused the increased seed yield? In any empirical investigation, experimental or observational, we can only advance an argument that is beyond reasonable doubt, not a logical certainty.

The key role played by the process of randomisation seems to be its insurance, up to a probability that can be calculated from the sampling distribution produced by the randomisation, that no uncontrolled common cause of both the treatment and the response variables could produce a spurious association. Fisher said as much himself when he stated that randomisation "relieves the experimenter from the anxiety of considering and estimating the magnitude of the innumerable causes by which his data may be disturbed". Is this strictly true? Consider again the possibility that soil moisture content affects seed yield. By randomly assigning the fertiliser to plots we insure that, *on average*, the treatment and control plots have soil with the same moisture content, therefore removing any chance correlation between the treatment received by the plot and its soil moisture[15]. But the number of attributes of the experimental units (i.e. the plots with their attendant soil and plants) is limited only by our imagination. Let's say that there

[15] More specifically, these two variables, being causally independent, are also probabilistically independent in the statistical population. This is not necessarily true in the sample due to sampling fluctuations.

are 20 different attributes of the experimental units that could cause a difference in seed yield. What is the probability that at least one of these was sufficiently concentrated, by chance, in the treatment plots to produce a significant difference in seed yield even if the fertiliser had no causal effect? If this probability is not large enough for you, then I can easily posit 50 or 100 different attributes that could cause a difference in seed yield. Since there are a large number of potential causes of seed yield, then the likelihood that at least one of them was concentrated, by chance, in the treatment plots is not negligible even if we had used many more than the 30 plots.

Randomisation therefore serves two purposes in causal inference. First, it ensures that there is no causal effect coming from the experimental units to the treatment variable or from a common cause of both. Second, it helps to reduce the likelihood in the sample of a chance correlation between the treatment variable and some other cause of the treatment but doesn't completely remove it. To cite (Howson and Urbach 1989 p. 152): "Whatever the size of the sample, two treatment groups are *absolutely certain* to differ in some respect, indeed, in infinitely many respects, any of which might, unknown to us, be causally implicated in the trial outcome. So, randomisation cannot possibly *guarantee* that the groups will be free from bias by unknown nuisance factors (i.e. variables correlated with the treatment). And since one obviously doesn't know what those unknown factors are, one is in no position to calculate the probability of such a bias developing either." This should not be interpreted as a severe weakness of the randomised experiment in any practical sense but does emphasise that even the randomised experiment does not provide any automatic assurance of causal inference, free of subjective assumptions.

Equally important is what is not required by the randomised experiment. The logic of experimentation up to Fisher's time was that of the controlled experiment, in which it was crucial that all other variables be experimentally fixed to constant values[16]; see, for example, (Feiblman 1972 p. 149). Fisher (1970) explicitly rejected this as an inferior method, pointing out that it is logically impossible to know if "all other variables" have been accounted for. This is not to say that Fisher did not advocate physically controlling for other causes in addition to randomisation. In fact, he explicitly recommends that the researcher do this whenever possible. For instance, in

[16] Clearly, this cannot be literally true. Consider a case in which the causal process is: A→B→C and we want to experimentally test whether A causes C. If we hold constant variable B then we would incorrectly surmise that A has no causal effect on C. It is crucial that common causes of A and C be held constant in order to exclude the possibility of a spurious relationship. It is also a good idea, although not crucial for the causal inference, that causes of C that are independent of A also be held constant in order to reduce the residual variation of C.

discussing the comparison of plant yields of different varieties, he advises that they be planted in soil "that appears to be uniform". In the context of pot experiments he recommends that the soil be thoroughly mixed before putting it in the pots, that the watering be equalised, that they receive the same amount of light and so on. The reason for doing this is not related to causality but, rather, to reduce residual variation and so to increase the statistical power to detect an effect if it exists. The strength of the randomised experiment is in the fact that we do not have to physically control – or even be aware of – other causally relevant variables in order to reduce (but not logically exclude) the possibility that the observed association is due to some unmeasured common cause in our sample.

Yet strength is not the same as omnipotence. Some readers will have noticed that the logic of the randomised experiment has, hidden within it, a weakness not yet discussed that severely restricts its usefulness to biologists; a weakness that is not removed even with an infinite sample size. In order to work, one must be able to randomly assign values of the hypothesised "cause" to the experimental units independently of any attributes of these units. This assignment must be direct and not mediated by other attributes of the experimental units. Yet, a large proportion of biological studies involves relationships between different attributes of such experimental units.

In the experiment described above, the experimental units are the plots of ground with their wheat plants. The attributes of these units include those of the soil, the surrounding environment in the plot and the plants. Imagine that the researcher wants to test the following causal scenario: the added fertiliser increases the amount of nitrogen absorbed by the plant. This increases the amount of nitrogen-based photosynthetic enzymes in the leaves and therefore the net photosynthetic rate. The increased carbon fixation due to photosynthesis causes the increased seed yield (Figure 1.1).

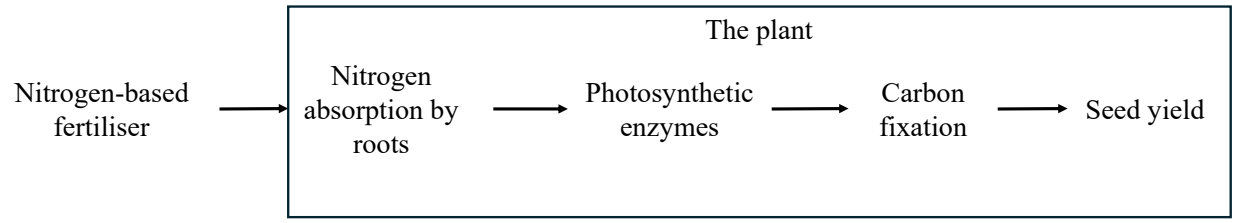


Figure 1.1 A hypothetical causal scenario that is not amenable to a randomised experiment.

The first part of this scenario (nitrogen-based fertiliser→nitrogen absorbed by roots) is perfectly amenable to the randomised experiment since the amount of nitrogen that is absorbed by the plant roots is an attribute of the plant (part of the experimental unit) while the amount of added fertiliser is controlled completely by the researcher independently of any attribute of the plot or its wheat plants. The rest of the hypothesis is impervious to the randomised experiment. For instance, both the rate of nitrogen absorption and the concentration of photosynthetic enzymes are attributes of the plant (the experimental unit). It is impossible to randomly assign rates of nitrogen absorption to each plant independently of any it's other attributes, yet this is the crucial step in the randomised experiment that allows us to distinguish correlation from causation. It is true that the researcher can induce a *change* both in the rate of nitrogen absorption by the plant and in the concentration of photosynthetic enzymes in its leaves but in each case these changes are due to the addition of the fertiliser. After observing an association between the increased nitrogen absorption and the increased enzyme concentration in the leaves, the randomisation of fertiliser addition does not exclude different causal scenarios, only some of which are shown in Figure 1.2.

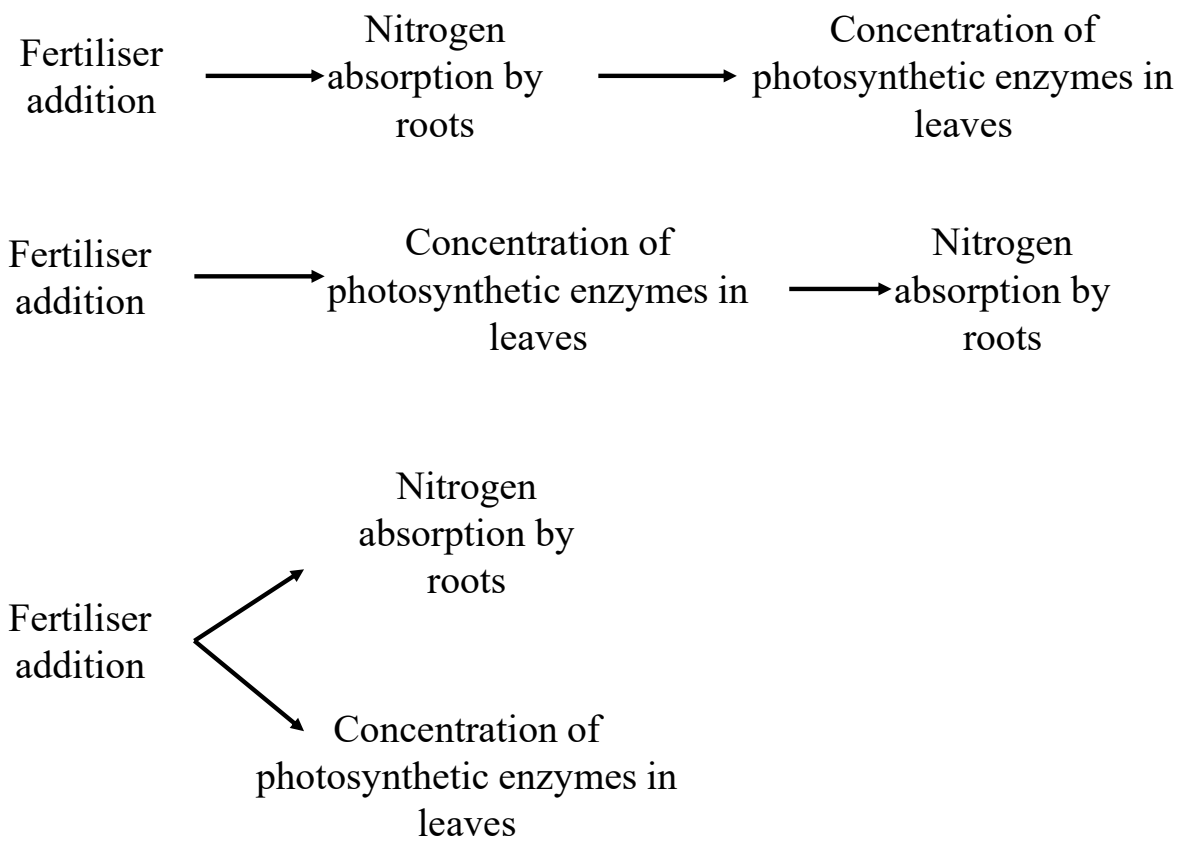


Figure 1.2. Three different causal scenarios that could generate the association between the increased nitrogen absorption by plant roots and the increased concentration of photosynthetic enzymes in leaves, following the addition of a nitrogen-based fertilizer to the soil.

While reading books about experimental design one's eyes often skim across the words "experimental unit" without pausing to consider what these words mean. The experimental unit is the "thing" to which the treatment levels are randomly assigned. The experimental unit is also an experimental *unit*. The causal relationships, if they exist, are between the external treatment variable and each of the attributes of the experimental unit that show a response. In biology the experimental units (for instance plants, leaves or cells) are integrated wholes whose parts cannot be disassembled without affecting the other parts. It is often not possible to randomly "assign" values of one attribute of an experimental unit independently of the behaviour of its other attributes[17]. When such random assignments cannot be done then one cannot infer causality

[17] This is not to say that it is always impossible. For instance, you can randomly manipulate the level of insulin in the blood because you can directly inject functioning insulin molecules into the blood and the only cause of these

from a random experiment. A moment's reflection will show that this problem is very common in biology. Organismal, cell and molecular biology are rife with it. Physiology is hopelessly entangled. Evolution and ecology, dependent as they are on physiology and morphology, are often beyond its reach. If we accept that one cannot study causal relationships without the randomised experiment, then a large proportion of biological research will have been gutted of any demonstrable causal content.

The usefulness of the randomised experiment is also severely reduced because of practical constraints. Remember that the inference is from the randomised treatment allocation to the experimental unit. The experimental unit must be the one that is relevant to the scientific hypothesis of interest. If the hypothesis refers to large-scale units (populations, ecosystems, landscapes) then the experimental unit must consist of such units. There is nothing in the logic of the randomised experiment that allows one to manipulate atmospheric CO_2 in small patches of forest, as done in the well-known FACE (Free Air CO_2 Enrichment) experiments, and then extrapolate to entire forests (Ainsworth and Long 2005). Someone wishing to know if increased atmospheric carbon dioxide concentrations will change the community structure of entire forests will have to use entire forests as the experimental units and assign increased CO_2 to such replicated treatment and control forests. Such experiments are never done in practice. Even when proper randomised experiments can be done in principle, they might not be permitted in practice due to financial or ethical constraints.

The biologist who wishes to study causal relationships using the randomised experiment is therefore severely limited in the questions that can be posed. The philosophically inclined scientist who insists that a positive response from a randomised experiment is an operational *definition* of a causal relationship would have to conclude that causality is irrelevant to much of science.

1.3 The controlled experiment

changes in insulin concentration (given proper controls) is the random numbers assigned to the animal. One cannot randomly add different numbers of functioning chloroplasts to a leaf.

Look again at the date of Fisher's first publication describing the randomised experiment: 1926. The currently prevalent notion that scientists cannot convincingly study causal relationships without the randomised experiment would seem incomprehensible to scientists before the twentieth century. Certainly, scientists thought that they were establishing causal relationships long before 1926, but they did not use randomised experiments. Instead, they used controlled experiments. The controlled experiment consists of proposing a hypothetical structure of cause-effect relationships, deducing what would happen if particular variables are controlled, or "fixed" in a particular state, and then comparing the observed result with its predicted outcome. This earlier method was described by Francis Bacon in his *Novum Organum* in the 17th century, by Hume in the 18th century and especially by J.S. Mill[18] in the 19th century (Boring 1954). Certainly, biologists *thought* that they were demonstrating causal relationships long before the invention of the randomised experiment. A wonderful example of this can be found[19] in *An Introduction to the Study of Experimental Medicine* (Bernard 1865) by the great nineteenth century physiologist, Claude Bernard[20]. I will cite a particularly interesting passage (Rapport and Wright 1963), and I ask that you pay special attention to the ways in which he tries to control variables. I will then develop the connection between the controlled experiment and the statistical methods described in this book.

"*In investigating how the blood, leaving the kidney, eliminated substances that I had injected, I chanced to observe that the blood in the renal vein was crimson, while the blood in the neighbouring veins was dark like ordinary venous blood. This unexpected peculiarity struck me, and I thus made observation of a fresh fact begotten by the experiment, but foreign to the experimental aim pursued at the moment. I therefore gave up my unverified original idea, and directed my attention to the singular colouring of the venous renal blood; and when I had noted it well and assured myself that there was no source of error in my observation, I naturally asked myself what could be its cause. As I examined the urine flowing through the urethra and reflected about it, it occurred to me that the red colouring of the venous blood might well be connected with the secreting or active state of the kidney. On this hypothesis, if the renal*

[18] A System of Logic, Ratiocinative and Inductive, 1843, Bk. III, chap. 8
[19] An English translation can be found in Bernard, C. (1927). *An Introduction to the Study of Experimental Medicine* (H. C. Greene, Trans.). New York: Macmillan.
[20] Rapport and Wright (1963) describe Claude Bernard (1813-1878) as an experimental genius and "a master of the controlled experiment".

secretion was stopped, the venous blood should become dark: that is what happened; when the renal secretion was re-established, the venous blood should become crimson again; this I also succeeded in verifying whenever I excited the secretion of urine. I thus secured experimental proof that there is a connection between the secretion of urine and the colouring of blood in the renal vein."

Our knowledge of human physiology has progressed far from the experiments of Claude Bernard (physiologists might find it strange that he spoke of renal "secretions"), yet his use of the controlled experiment would be immediately recognisable and accepted by modern physiologists. Fisher was correct in describing the controlled experiment as an inferior way of obtaining causal inferences, but the truth is that the randomised experiment is unsuited for much of biological research. In the experiment described by Claude Bernard, the hypothetical causal structure could be conceptualised as shown in Figure 1.3.

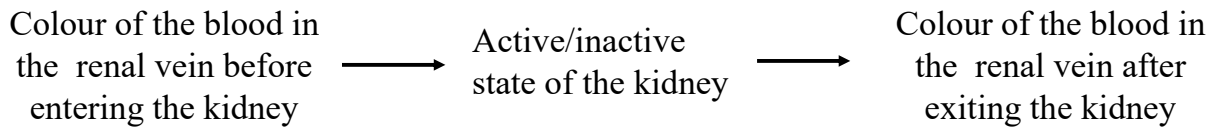


Figure 1.3 Claude Bernard's hypothesised causal explanation for the change in colour of the blood in the renal vein.

The key notion in Bernard's experiment was the realisation that, if his causal explanation was true, then the type of *association* between the colour of the blood in the renal vein as it enters and leaves the kidney would change, depending on the state of the hypothesised cause, i.e. whether the kidney was secreting or not. It is worth returning to his words: "On this hypothesis, if the renal secretion was stopped, the venous blood should become dark: that is what happened; when the renal secretion was re-established, the venous blood should become crimson again; this I also succeeded in verifying whenever I excited the secretion of urine. I thus secured experimental proof that there is a connection between the secretion of urine and the colouring of blood in the renal vein." Since he explicitly stated earlier in the quote that he was inquiring into the "cause" of the phenomenon, it is clear that he viewed the result of his experiments as establishing a *causal connection* between the secretion of urine and the colouring of blood in the renal vein.

Although the controlled experiment is an inferior method of making causal inferences relative to the randomised experiment, it is actually responsible for most of the causal knowledge that science has produced. The method involves two basic parts. First, one must propose a hypothesis stating how the measured variables are linked in the causal process. Second, one must deduce how the associations between the observations must change once particular combinations of variables are controlled so that they can no longer vary naturally; i.e. once particular combinations of variables are "blocked". The final step is to compare the patterns of association after such controls are established with the deductions. Historically, variables have been blocked by physically manipulating them. However (and this is an important point that will be more fully developed and justified in Chapter 2) it is the control of variables, not how they are controlled, that is the crucial step. The weakness of the method, as Fisher pointed out, is that one can never be sure that all relevant variables have been identified and properly controlled. In any field of study, the first causal hypotheses are generally wrong and the process of testing, rejecting, and revising them is what leads to progress in the field.

1.4 Physical controls and observational controls

It is the control of variables, not how they are controlled, that is the crucial step in the controlled experiment. What does it mean to "control" a variable? Can such control be obtained in more than one way? In particular, can one control variables using statistical, rather than experimental, methods? The rigorous links between a physical control through an experimental manipulation and a statistical control through conditioning will be developed in the next chapter, but it is useful to provide an informal demonstration here using an example that should present no metaphysical problems to most biologists.

Body size in large mammals seems to be important in determining much of their ecology. In populations of Bighorn Sheep in the Rocky Mountains, it has been observed that the probability of survival of an individual through the winter is related to the size of the animal in the fall. However, this species has a strong sexual dimorphism with males being up to 60% larger than females. Perhaps the association between body size and survival is because males have a better

probability of survival than females and this is unrelated to their body size? In observing these populations over many years, perhaps the observed association arises because those years showing better survival also have a larger proportion of males? Figure 1.4 shows these two alternative causal hypotheses. I have included variable labelled "other causes" to emphasise that we are not assuming the chosen variables to be the only causes of body size or of survival.

Figure 1.4. Two alternative causal explanations for the relationship between body size and spring survival in Bighorn sheep.

Notice the similarity to Claude Bernard's question concerning the cause of blood colour in the renal vein. The difference between the two alternate causal explanations in Figure 1.4 is that the second explanation assumes that the association between spring survival and autumn body size is due only to the sex ratio of the population. Thus, if the sex ratio could be held constant, then the association would disappear. Since adult males and females of this species live in separate groups for most of the year, it would be possible to physically separate them in their range and, in this way, physically control the sex ratio of each subpopulation. However, it is much easier to simply sort the data according to sex and then look for an association within each homogeneous group. The act of separating the data into two groups such that the variable in question – the sex ratio – is constant within each group represents a *statistical control*. We could imagine a situation in which we instruct one set of researchers to physically separate the original population into two groups based on sex by fencing off their ranges, after which they test for the association within each of their experimental groups and then ask them to combine the data together and give them to a second team of researchers. The second team would analyse the data using the statistical control. Both groups would come to identical conclusions in this case, although it is not true that statistical and physical controls will always give the same conclusion; this is discussed in Chapter 2. In fact, using statistical controls might even be preferable in this situation. Simply observing the population over many years and then statistically controlling for

the sex ratio on paper does not introduce any physical changes in the field population. It is likely that the act of physically separating the sexes in the field might introduce some unwanted, and potentially uncontrolled, change in the behavioural ecology of the animals during the rut that might bias the survival rates during the winter quite independently of body size.

Let's further extend this example to look at a case in which it is not as easy to separate the data into groups that are homogeneous with respect to the control variable. Perhaps the researchers have also noticed an association between the amount and quality of the rangeland vegetation during the early summer and the probability of survival during the next winter. They hypothesise that this pattern is caused by the animals being able to eat more during the summer, which increases their body size in the autumn which then increases their chances of survival during the winter: quantity of summer forage (kg) →body mass in fall (kg) →probability of survival until the spring.

The logic of the controlled experiment requires that we be able to compare the relationship between forage quality and survival until spring after physically preventing body weight from changing, which we cannot do[21]. We could, of course, exert *indirect* experimental control on body weight but this is not what the logic of the controlled experiment requires. For instance, we could restrict food intake in one group (producing lower body weight) and providing unlimited food to the other group (producing increase body weight) but changing food availability could potentially induce many other physiological and behavioural changes besides body weight. We can't even impose a statistical control by sorting the data on body weight and then divide animals into groups that are homogeneous for this variable, since "body weight" is a continuous variable and so each animal will have a different body weight. Nonetheless, there is a way of statistically comparing the relationship between forage quality and winter survival while controlling the body weight of the animals during the comparison. This involves the concept of statistical conditioning, which will be more rigorously developed in Chapters 2 and 3. An intuitive understanding can be had with reference to a simple linear regression (Figure 1.5).

[21] It is actually possible, in principle if not in practice, to conduct a randomised experiment in this case, so long as we are interested only in knowing if summer forage quality causes a change in spring survival without reference to autumn body weight. This is because the hypothetical cause (vegetation quality and quantity) is not an attribute of the unit (the animal) possessing the hypothetical effect (spring survival). Again, it is impossible to use a randomised experiment to determine if body size in the autumn is a cause of increased survival in the spring.

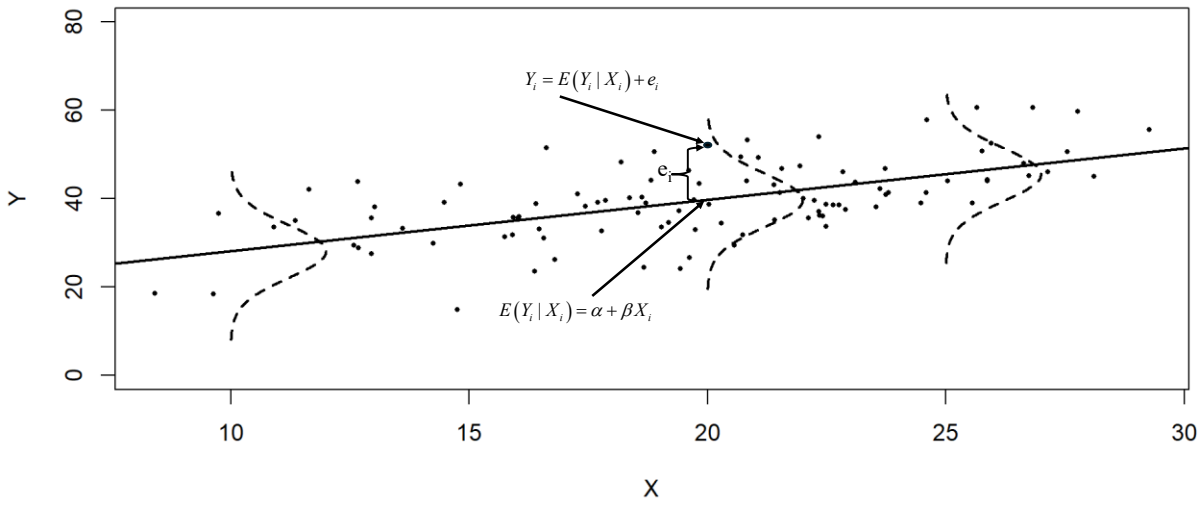


Figure 1.5. The observed value of Y when X=20 is equal to the value predicted by the bivariate regression of Y on X, i.e. $E(Y \mid X_i) = \alpha + \beta X_i$, when X equals 20 plus the residual variation, e_i, of Y_i.

The formula for a linear regression is: $Y_i = \alpha + \beta X_i + e_i$. As the formula makes clear, the observed value of Y consists of two parts. One part, "the expected value of Y_i given the value of X_i", depends on X (i.e. $E(Y_i \mid X_i) = \alpha + \beta X_i$) and the other part that doesn't (i.e. e_i). The expected value of Y given X is the solid line in Figure 1.5. The second part of the value of Y, the part that doesn't depend on X (i.e. e_i), is the part of Y that randomly varies from observation to observation. Because e_i is random, it is sometimes small, in which case Y_i is close to the solid line, and sometimes large, in which case Y_i is much higher or lower that the solid line. In linear regression we must know the distribution of these random deviations from the expected value. Here, the random deviations follow a normal (or Gaussian) probability distribution, and, by assumption, this same normal distribution applies for every possible value of X; this is why the random deviations are independent of X. If we subtract the expected value of each Y_i given X_i, from the value of Y_i itself, then we get the variation in Y_i that is independent of X. This new variable is called the *residual* of Y given X. These are the values of Y that exist for a constant value of x.

If we want to compare the relationship between forage quality and winter survival while controlling the body weight of the animals during the comparison, then we have to remove the effect of body weight on each of the other two variables. We do this by taking each variable in turn, subtracting the expected value of it given body weight, and then see if there is still a relationship between the two sets of residuals. Because each set of residuals are independent of body weight, we are now looking at the variation in that part of summer forage and that part of spring survival that are each independent of body weight. In this way, we can hold constant the effect of body weight in a way that is similar to experimentally holding constant the effect of some variable.

I chose the words "similar to" specifically to emphasise that I am only using an analogy to compare physically controlling a variable and statistically (or observationally) controlling a variable. The analogy is not yet exact. There are situations in which statistically holding constant a variable will produce different patterns of association that those that would occur when physically holding constant the same variable. To know when statistical controls cast the same correlational shadows as experimental controls, and when they differ, we need a provably correct way of translating from the language of causality to the language of probability distributions. This is the topic of the next chapter.

2

From cause to correlation and back

The official language of statistics is the probability calculus, based on the notion of a probability distribution. For instance, if you conduct an ANOVA then the key piece of information is the probability of observing a particular value of Fisher's F statistic in a random sample of data, given a particular hypothesis or model. To obtain this crucial piece of information, you (or your computer) must know the probability density function of the F statistic. Statistics can tolerate certain other mathematical languages but, in the end, you have to convert your biological hypothesis into a probability distribution in order to be understood. If we wish to study causal relationships using statistics then we have to translate, *without error*, from the scientist's language of causality to the language of probability theory.

You might think that this is straightforward. If so, then you are wrong. If I tell you that (1) X causes Y (X→Y), that (2) the relationship between X and Y is linear, and that (3) Y follows a normal distribution with a mean of μ and a standard deviation of σ, then you might be tempted to mathematically represent these facts as a simple linear regression: $Y = \alpha + \beta X + N(\mu=0, \sigma)$. Doing so is an error in translation between the symbol for "cause" (→) and the symbol for "numerical equivalence" (=). The problem is that the phrase "X causes Y" implies several properties of this causal relationship that don't exist in algebra and aren't attached to the "=" symbol. For instance, a causal relationship is asymmetric; if X changes then this will change Y but if Y changes this will not change X. This is why the symbol for cause (→) is a unidirectional arrow. The equality symbol (=) in algebra is symmetric and simply means that the numerical value on its left is the same as the numerical value on its right. Therefore, if X changes then Y must change and if Y changes then X must also change in order to respect this equality. That is why both $Y = \alpha + \beta X + N(\mu=0, \sigma)$ and $X=-(\alpha/\beta)+Y/\beta+N(\mu=0,\sigma/\beta)$ are both true following the

properties of equality in algebra. Yet, it would be profoundly wrong to claim that if X→Y then Y→X following the properties of causal claims.

A rigorous translation device between the scientific language of causality and the mathematical language of probability distributions did not exist until very recently (Pearl 1988, Verma and Pearl 1988, Pearl 1993). It is no wonder that, until recently, statisticians had virtually banished the word "cause" from statistics – such a word has no equivalent in their language[22]. Until recently, within the world of statistics, the scientific notion of causality has been a stranger in a strange land. Posing causal questions in the language of the probability calculus without this rigorous translation device is like a unilingual Englishman asking for directions to the Louvre from a Frenchman who can't speak English. The Frenchman might understand that directions are being requested to the "Louvre", and the Englishman might see fingers pointing in particular directions, but it is not at all sure that works of art will be found. Imperfect translations between the language of causality and the language of probability theory are equally disorienting.

Mistakes in translation come in all kinds. The most dangerous ones are the subtle errors in which a slight change in the inflection or context of a word can change the meaning in disastrous ways. Because the French word "demande" both sounds like the English word "demand" and has approximately – but not exactly – the same meaning (it simply means "to ask for" without any connotation of obligation), I have seen French-speaking people come up to a store clerk and, while speaking English, "demand service". They think that they are politely asking for help while the clerk thinks that they are issuing an ultimatum. Translating from the scientific concept of cause (→) into the algebraic concept of equality (=) is the same sort of subtle error that can lead to misunderstanding and scientific errors.

Mathematical languages are logic machines. They start with initial statements that are true by definition or assumption (axioms) and then derive the logical consequences of these axioms. When a person talks about "X causing Y", they imply, usually implicitly, several properties of such a causal relationship. If we translate the word "cause" into a mathematical language whose axioms don't possess the properties that we imply by the word "cause" then the logical consequences of this improper mathematical language can lead to incorrect or counterintuitive

[22] Fisherian statistics does deal with causal hypotheses, but the causal inferences come from the experimental design, not from the mathematical model; see Chapter 1.

conclusions. So, what is this more rigorous method of translation between the scientific language of causality and the mathematical language of probability distributions that Judea Pearl devised? It actually involves two translation steps. First, we translate our verbal scientific claims about causality into the mathematical language of graph theory. This is because certain types of mathematical graphs share a number of important axiomatic properties with our scientific notion of "cause". Second, we translate from the language of graph theory to the language of probability distributions using provably correct relationships between these mathematical graphs and the properties of probability distributions that are generated from these graphs. This provides a provably correct logical sequence from the scientific notion of cause-and-effect to the official language (probability theory) of statistics.

This might sound intimidating. It is not. This book is not written for mathematicians (they probably wouldn't like it) and you don't need any advanced knowledge of mathematics to understand what follows. The first step – translating scientific causality into mathematical graphs – is quite easy for most biologists. After teaching these ideas to biology students for many years, I have found that the second step – translating mathematical graphs to probability distributions – is usually grasped after only a few minutes of practice.

2.1 Translating from causal claims to directed acyclic graphs

In the mathematical subject of graph theory, a "graph" has nothing to do with scatterplots or histograms. Rather, a graph is a mathematical structure that describes particular defined relationships between pairs of objects and the consequences of these relationships for the properties of networks of such objects. For instance, here is an example of one type of graph: X→Y→Z. The objects in a graph (here, X, Y and Z) are called *vertices* (or nodes) and the relationships joining pairs of objects (here, →) are called *edges*. For our purpose, each vertex represents a variable and so our graphs will consist of variables and the edges between them. Depending on what properties we give to the edges, we end up with different types of graphs. In chapter 6 I will discuss a larger class of edges but, in this chapter, we will use just one type of

edge, an arrow (→), which is also called a *directed* edge. Graphs having only arrows are called *directed graphs*.

An important defined property of a directed edge is that the relationship (or information) between variables can only travel in the direction of the arrow. This means that the asymmetric relationship between variables in a directed graph has the same property as the cause-effect relationship in the scientific notion of causality. As an empirical scientist, you might be tempted to ask, "but how can I know that the relationship only travels in one direction"? This is the wrong question. Directed graphs have this type of relationship because we have defined them this way. We are still in the world of mathematics and haven't yet arrived in the world of empirical reality. If the relationship between your variables doesn't have this property, then you have to choose a different type of graph and shouldn't choose directed graphs! Next, we will restrict our attention to one particular type of directed graph: a directed *acyclic* graph, or DAG. A DAG is a directed graph that has no cycles (i.e. feedback relationships); that is, a directed graph in which you cannot start at any variable and, following the directions of the arrows, cycle back to the same variable. Since many biological systems definitely do have feedback cycles, this is an important assumption. We will deal with this topic later. Figure 2.1 shows three directed graphs. The graph in Figure 2.1a is not a DAG because we can start at X_1 and follow the arrows back to X_1, i.e., $X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_1$. The graph in Figure 2.1b is not a DAG because we can start at X_1, travel to X_2 ($X_1 \rightarrow X_2$) and then go right back to X_1 ($X_1 \leftarrow X_2$). Only Figure 2.1c is a DAG.

> DAG (definition): a graphical object containing only arrows between variables and in which it is impossible to cycle back to the same variable while following the directions of the arrows.

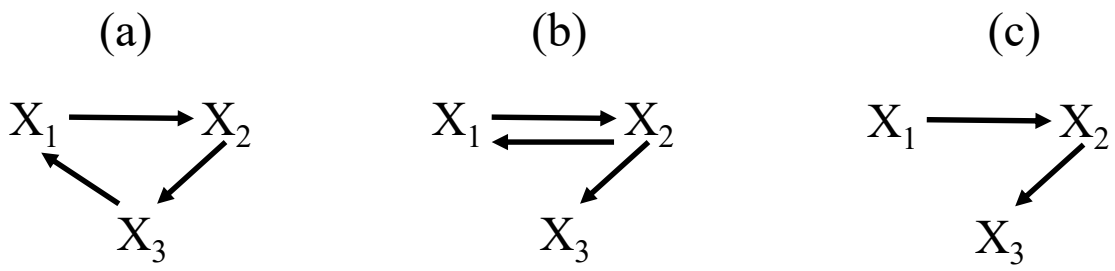


Figure 2.1. Three directed graphs, of which only (c) is a directed acyclic graph (DAG).

I said that the arrow (→) in the DAG is a translation of the scientific concept of "cause" but that is too vague. More precisely, the notation $X_i \rightarrow X_j$ in a DAG is a claim (a hypothesis) that a change in X_i will provoke a change in X_j *even if all other variables in the DAG are prevented from changing*. You have to imagine a hypothetical manipulative experiment in which you physically prevent all of the variables in your DAG except for X_i and X_j from changing and then only change the value of X_i. If you believe that only changing X_i will provoke a change in X_j while preventing all other variables in the DAG from changing in this imaginary experiment, then you write $X_i \rightarrow X_j$ in your DAG. Each arrow in the DAG is a prediction from such an imaginary manipulative experiment. We call the $X_i \rightarrow X_j$ link a *direct* cause because it is a causal link between X_i and X_j that exists independently from any other variable in the DAG. A DAG is composed only of variables and the direct casual links (i.e., arrows) between them. However, the "missing" arrows in a DAG are just as important as the arrows that are present because these, too, are clausal claims. Specifically, a *missing* arrow between X_i and X_j is a claim that (i) if X_i is changed while holding all other variables in the DAG constant except X_j, then X_j will not change, and (ii) if X_j is changed while holding all other variables in the DAG constant except X_i, then X_i will not change.

> Direct cause (definition): Given a DAG G composed of a set of variables **V** containing X_i and X_j, X_i is a direct cause of X_j relative to the remaining variables in **V** if a change in X_i will provoke a change in X_j while holding constant all other variables in **V** but not *vice versa*. A direct cause in a DAG is represented by the arrow symbol: "→".

It doesn't matter if you can't physically carry out such an experiment; it's only a thought experiment[23]. Please keep this image of an imaginary manipulative experiment clearly in your mind when you begin to write down your own DAGs. One of the most dangerous mistakes that you can make is to confuse the causal claim $Xi \rightarrow X_j$ with a claim that X_i is "correlated" with, or "associated" with, or "predicts" X_j. It is also a mistake to confuse the claim $X_i \rightarrow X_j$ with the more general claim that X_i is simply *a cause* of X_j rather than the more precise claim that X_i is a

[23] A philosopher would call this a "counterfactual claim".

direct cause of X_j. This mistake is like confusing the French verb "demander" with the English verb "demand". The two verbs sound similar and both have approximately the same meaning but confusing them could still get you slapped in the face.

One important advantage of translating from the scientific language of causality to the mathematical language of graph theory is that this mathematical language is very precise. Unlike natural languages, where the same word can have different meanings in different contexts[24], nothing in a mathematical language is vague or implicit. Another mistake that beginners make when translating from the scientific language of causality to the mathematical language of graph theory is to think that the claim $X_i \rightarrow X_j$ means that X_i is a direct cause of X_j even if we could hold constant every other variable that might exist in Nature. This is wrong. In fact, it is worse than wrong; it is meaningless. "Variables" are constructs that we create and there are potentially an infinite number of them. That is why, in the definition of a "direct" cause, I stated that "X_i is a direct cause of X_j *relative to the remaining variables in* $\boldsymbol{V}$". In our scientific language of causality, any causal explanation can be modified by adding more or less detail. For instance, an agronomist might first claim that the amount of inorganic nitrogen added to the soil (X_1) is the direct cause of an increase in plant growth rate (X_4): $X_1 \rightarrow X_4$. Later, she might claim that the amount of inorganic nitrogen in the soil (X_1) directly causes an increase in the amount of nitrogen taken up by a plant (X_2) which then causes an increase in plant growth rate (X_4): $X1 \rightarrow X_2 \rightarrow X_4$. In this new DAG, the amount of inorganic nitrogen added to the soil (X_1) is no longer the direct cause of the increase in plant growth rate (X_4). Rather, the amount of nitrogen taken up by the plant (X_2) is a direct cause of the increase in plant growth rate (X_4). Why? Because (according to our hypothesis) if we increase the amount of added inorganic nitrogen to the soil (X_1) but force the plant to keep constant the amount of nitrogen that it takes up (X_2), this would not increase the plant growth rate. What was a direct cause (X_1) in the first DAG is no longer a direct cause in the new DAG. However, an even more detailed causal explanation might introduce a new variable: the quantity of photosynthetic enzymes in the leaves (X_3). In this more detailed explanation, our agronomist might propose a third DAG: $X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4$. Now, the amount of nitrogen taken up by the plant (X_2) is no longer a direct

[24] What does "don't rock the boat" mean? This depends on if we have just complained to our supervisor or if we are in the middle of a lake on a fishing trip.

cause of plant growth rate (X_4). In Chapter 6 you will learn how to modify DAGs, and other graphical objects, when we add or remove variables from them.

DAGs have another important property that aligns closely with scientific notions of causality. Let's go back to the second DAG of our agronomist: $X1 \rightarrow X_2 \rightarrow X_4$ involving only the amount of inorganic nitrogen in the soil (X_1), the amount of nitrogen taken up by a plant (X_2) and plant growth rate (X_4). In our scientific language of causality, we would say that the level of soil inorganic nitrogen is an *indirect* cause of plant growth rate. After all, if we conducted a randomised experiment in which we increase the amount of soil inorganic nitrogen then this would still provoke a change in plant growth rate (according to our causal hypothesis) and so soil nitrogen is still *a cause* of plant growth rate. However, it is an indirect cause because it only occurs if the plant is actually able to take up the additional soil nitrogen; i.e., if X_2 is allowed to respond naturally. This scientific notion of indirect causes can be translated into a precise property of DAGs. If we can trace a path from X_i to X_j that passes through at least one other variable (X_k) in the DAG while respecting the directions of the arrows, then X_i is an indirect cause of X_j. In fact, using this notion of tracing paths in a DAG, we can immediately see that one variable can have more than one indirect causal effect on a second variable. Because we must be precise with our definitions in graph theory, we will need to state clearly what we mean. I will use the DAG in Figure 2.2 to illustrate these definitions.

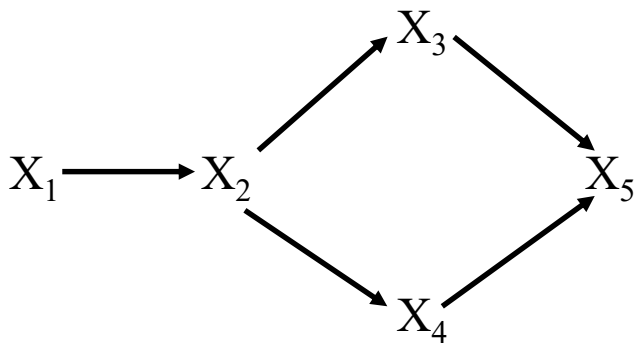


Figure 2.2. A DAG involving five variables

A *path* between any two variables in a graph is simply the sequence of variables that you must traverse in order to get from one to the other. We make a distinction between a directed path and an undirected path. A *directed path* in a DAG is the sequence of variables that you can traverse

to get from one to the other while respecting the directions of the arrows and without visiting the same variable more than once along that particular path. Given our earlier definition of a direct effect, a direct effect is a special type of directed path in a DAG. Specifically, a direct effect is a directed path between two variables that does not involve any other variables. An *undirected path* in a DAG is the sequence of variables that you can traverse to get from one to the other if you ignore (not erase) the directions of the arrows and without visiting the same variable more than once along that particular path. Logically, every directed path is also an undirected path since the arrows are still there even if you ignore their direction. For instance, there are two different directed paths from X_1 to X_5 in Figure 2.2: $X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_5$ and $X_1 \rightarrow X_2 \rightarrow X_4 \rightarrow X_5$. If we ignore the directions of these arrows, then these two directed paths are also undirected paths. However, not every undirected path is also a directed path. There are also two undirected paths between X_3 and X_4: $X_3 \leftarrow X_2 \rightarrow X_4$ and $X_3 \rightarrow X_5 \leftarrow X_4$. Neither of these two undirected paths are also directed paths because, in each case, we must travel against (i.e., ignore) the directions of the arrows. Now we have a definition of an indirect cause in the language of DAGs.

> Indirect cause (definition): Given a DAG $\mathcal{G}$ composed of a set of variables $\mathbf{V}$ including X_i and X_j, X_i is an indirect cause of X_j relative to the remaining variables in $\mathbf{V}$ if there is at least one directed path from X_i to X_j that involves at least one additional variable besides X_i and X_j. It is interpreted causally as a claim that a change in X_i will provoke a change in X_j while holding constant all other variables in $\mathbf{V}$ except those along the directed path.

Note that it is possible for there to exist both a direct cause and an indirect cause between two variables. For instance, if we added the arrow $X_2 \rightarrow X_5$ in the DAG in Figure 2.2, then X_2 would be both a direct cause of X_5, because of $X_2 \rightarrow X_5$, and an indirect cause because of the two paths $X_2 \rightarrow X_3 \rightarrow X_5$ and $X_2 \rightarrow X_4 \rightarrow X_5$. People using DAGs often employ familial terms to describe the relationships between variables. For instance, a variable that is a direct cause of another is also called the *parent* and its direct effect is the *child*. In Figure 2.2, X_1 is the parent of X_2 and X_2 is the child of X_1. An *ancestor* of a variable is any variable in the DAG that has a directed path from it (the ancestor) to the variable in question. Of course, every parent of a variable is also an ancestor of that variable since a direct cause ($X_i \rightarrow X_j$) is a type of directed path. Variables X_1, X_2, X_3 and X_4 are all ancestors of X_5 in Figure 2.2 because there are one or more directed paths from

each to X_5. A *descendent* of a variable (X_i) is any variable in the DAG having a directed path to it from X_i. Every child is also a descendent. Variable X_5 is a descendent of all the other variables in Figure 2.2.

You're probably getting tired of learning definitions. There are a few more definitions to learn before I can explain how we translate from the language of graph theory to the language of probability distributions, but let's pause and look at a biological example in order to apply what you have learnt so far.

2.2 An empirical example of constructing a DAG: Corsican Blue Tits

This example comes from Thomas et al. (2007) who studied certain physiological and environmental causes determining the fledgling success in nestlings of Blue Tits (*Cyanistes caeruleus*), in an woodland in Corsica. In these small birds, the newly hatched chicks must go through a short and intense (~12 day) growth period, leading to an asymptotic body mass by day 25, during which most tissue construction and maturation occurs. By approximately 20 days of age, the nestling then leaves the nest. All birds in this long-term study were banded. A newly fledged bird that returned to the study site the next year was considered to have been recruited into the population. "Recruitment" of a fledglings into the population was therefore a binary (yes/no) variable. Since Blue Tits usually return to the same site each year in the spring, a young bird that failed to recruit most likely died during the intervening year.

The asymptotic mass of the fledgling (g) is very sensitive to its nutrient and energy flux during the short growth period before fledging. The nutrient and energy flux represents the balance between the amount of food provided to the chicks by their parents and the amount of nutrients and energy removed from the chicks by blood-sucking ectoparasites that live in the nest. The food were caterpillars that eat the young Oak leaves during the spring breeding season. Caterpillar abundance was indirectly estimated by placing 15 0.25m^2 collectors in the study site and weighing the daily caterpillar frass (faeces) produced. From this data, the authors were able

to estimate the daily abundance of caterpillars and then caterpillar abundance[25] when each chick reached its asymptotic mass at age 15 days (g m^{-2} d^{-1}). The ectoparasites are the larvae of Blowflies (*Protocalliphora* spp.). The authors manipulated the number of ectoparasites per nest by randomly assigning the nests to either a treatment or a control group. The treatment nests had a nylon fabric installed in the nest cup that blocked access by the ectoparasites to the nestlings and therefore killed them. The control nests did not have the nylon fabric. The number of Blowfly larvae varied naturally in the control nests. The number of ectoparasites in the control nests varied from nest to nest and from year to year (this was a multi-year study). Finally, the authors measured a physiological attribute of each nestling at age 15 days (haematocrit). The haematocrit[26] (the proportion of blood volume occupied by red blood cells, expressed as a percentage). A larger haematocrit increases the transport of oxygen to muscle cells and therefore increases endurance and aerobic capacity of the fledgling during flight. In total, the authors had access to five variables: (1) the number of caterpillars available (food), (2) the number of ectoparasites per nest (parasites), (3) the asymptotic body mass per fledgling (body mass), (4) the volume of the haematocrit (haematocrit) and the recruitment success (yes/no) of the individual. The authors had specific hypotheses concerning how these variables should be related as a causal system but, for this example, I will simplify their causal hypothesis[27] slightly (Figure 2.3a). Let's look at the causal claims and how they map onto the DAG.

[25] The authors reported frass production as mg m^{-2} d^{-1} but I converted this to g m^{-2} to increase the size of the resulting path coefficients.
[26] The data set uses the American spelling of "hematocrit". I have maintained this spelling when referring specifically to the variable in the data set to avoid confusion.
[27] In the original study, the authors created a new variable (massXhematocrit) that was the product of mass and haematocrit in order to introduce an interaction term that was the direct cause of recruitment.

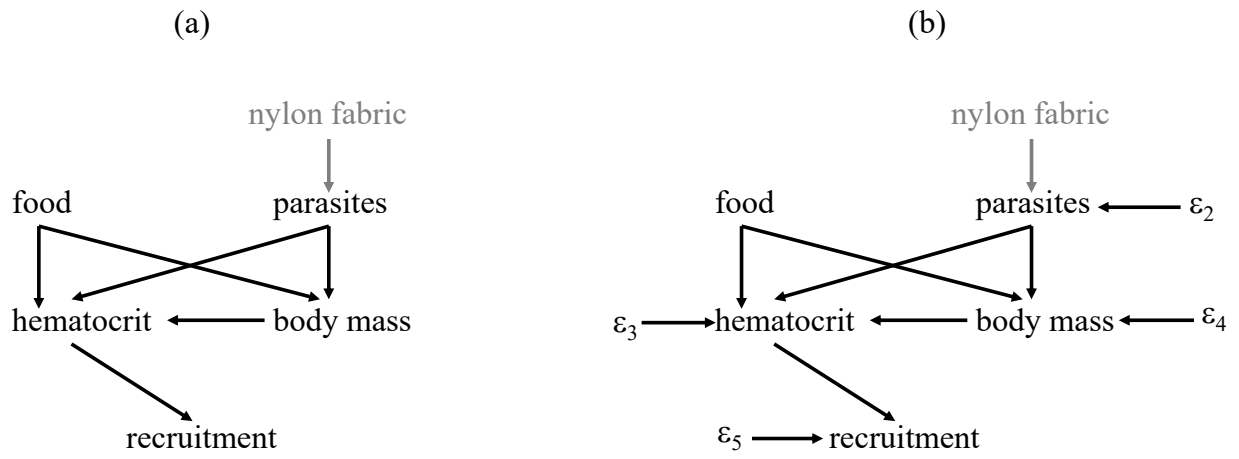


Figure 2.3. (a) The DAG proposed to represent the hypothesized causal structure controlling the recruitment success of Blue Tit fledglings by Thomas et al. (2007). (b) A more explicit DAG of the same hypothesis.

First, notice that there is no directed path from food to parasites. This means that the authors did not believe that there was any way for variation in food availability to cause any increase or decrease in the number of Blowfly larvae in the nest. If you disagree with this causal claim, then you would modify the DAG appropriately. In Chapter 3, you will learn how to test the DAG but, for now, simply be aware that this is a testable prediction. In fact, the missing arrows in a DAG are crucial in allowing us to test causal claims in structural equation modelling. Certainly, the fact that the treatment nests removed all the Blowfly larvae, and that the treatment was randomly assigned, argues in favour of this assumption. However, the control nests did not fix the number of parasites; rather, the number of parasites in these nests varied naturally. I have no specialist knowledge of the biology of Blowflies but a cursory literature search tells me that a series of environmental cues attract Blowflies. These cues include the movement of adult birds around the nest, pheromones produced by the Blue Tits, olfactory cues generated by the presence of feathers or faeces in the nest or even the presence of other Blowflies around the nest. If more food around the nest did cause an increase in any of these attractants, then there would be a directed path from food to parasites in the DAG that does not pass through any other variable in the DAG.

As I said in Chapter 1, although correlation does not imply causation, causation (almost always) does imply correlation. Even if we are willing to assume that there is no causal link between food and parasites, this fact gives us another reason to question the DAG. The study was

conducted over several years. The spring climate in the area (precipitation and temperature) would have varied from year to year and it is reasonable to expect that years that are favourable to the production of Oak leaves would increase the number of caterpillars (i.e., food). If the climate also affects the population of Blowflies from year to year, then there would be an added path food←climate→parasites. Since climate was not measured (at least, not reported) this is an example of a "latent" variable. Latent variables are an important topic that will be dealt with in several chapters in this book, but we won't do it in this chapter.

Next, the DAG in Figure 2.3a has direct paths from each of food and parasites to each of haematocrit and body mass. The direct path food→hematocrit is a claim that more food eaten by a chick would cause it to have a greater proportion of red blood cells per blood volume even if no other variable in the DAG (including its body mass) changed. The direct path food→body mass is a claim that more food eaten by a chick would increase its eventual body mass even if no other variable in the DAG (including its proportion of red blood cells) changed. The direct path parasites→haematocrit is a claim that a greater number of parasites infesting a chick will cause the chick to have (presumably) a lower concentration of red blood cells even if no other variable in the DAG changed. The direct path parasites→body mass is a claim that a greater number of parasites infesting a chick will cause the chick to have (presumably) a smaller body mass even if no other variable in the DAG changed. Because there are directed paths between both food and parasites to recruitment but no arrows from either food or parasites to recruitment, this is a claim that (i) both food and parasites cause a change in recruitment but (ii) that food and parasites *only* cause a change in recruitment by modifying the haematocrit and body size.

It has taken me 667 words to describe the causal claims imbedded in the DAG in Figure 2.3a and this was only a partial[28] list. A DAG is a very compact way of encoding, making explicit, and linking together many causal claims (i.e., causal hypotheses). However, structural equations don't speak the language of graph theory; they speak the language of probability distributions. We must now discuss how to translate from the language of DAGs to the language of probability distributions.

[28] For instance, the DAG claims that larger chicks will have a greater volume of red blood cells (haematocrit) even if the amount of food and the number of parasites infecting it where both held constant.

2.3 Data Generating Mechanisms

Except in the realm of quantum physics, modern science assumes a deterministic universe in which causes and effects are propagated forwards in time. If we knew all the causes of a given effect, and if we knew exactly how to mathematically describe each cause-effect link, then we could perfectly predict each effect. Certainly, perfect predictive ability is almost never the case in the real world except in very simple systems. However, modern science assumes that our lack of perfect predictability arises because we almost never know all the causes of an event and/or we lack perfect knowledge of the mathematical functions and the initial conditions.

Stated slightly differently, modern science assumes that Nature generates the observed values of variables via cause – and – effect mechanisms. If our DAG correctly describes the cause – effect relationships between our variables, then this DAG describes the qualitative links of one part of Nature's data generating mechanism. When we replace the arrows of the DAG (the qualitative links) with actual mathematical equations then we can translate this DAG into a set of equations that describe how the cause – effect relationships are propagated. These equations are called "structural" equations because they are structured according to the cause – effect mechanisms that are represented in the DAG. I will write $Y_i = f_i(X_i)$ to mean "Y_i is linked to X_i according to some unspecified function f_i". For instance, given our Blue Tits DAG (Figure 2.3a), these are the structural equations implied by our DAG:

$$\begin{aligned}
&\text{parasites} = f_2(\text{nylon fabric}) \\
&\text{hematocrit} = f_3(\text{food}, \text{parasites}, \text{body mass}) \\
&\text{body mass} = f_4(\text{food}, \text{parasites}) \\
&\text{recruitment} = f_5(\text{hematocrit})
\end{aligned}$$

Equation 2.1(a-d)

If you have understood the link between causes and DAGs then you might have noticed something very wrong about the DAG in Figure 2.3a, and about the resulting structural equations in Equation 2.1. According to these equations, each of the variables on the left-hand side can be completely determined by the variables on the right-hand side. As an empirical claim, this is highly unlikely. For instance, even if we believe that the body mass of a nestling is caused by the amount of food given to it and the number of parasites feeding on it, we wouldn't believe that its body mass is *only* caused determined by these two variables. More likely, we would expect

that there exist other variables (perhaps unknown to us) that also cause changes in body mass besides the two variables explicitly included in the DAG. To solve this problem, we must be more explicit in both our DAG and in our resulting structural equations. Let ε_i represent all other causes of variable i (known or unknown) besides those explicitly listed in function f_i. Often, the ε_i are omitted from DAGs but they are almost always present even if they are implicit. The ε variables are often called "error" variables. I will follow this terminology even if it is potentially misleading. This potential for confusion exists because beginning users of SEM equate "error variables" with the "residuals" of a regression. Although there is a link between the two under special conditions, they are not the same. The residuals of a regression are simply the difference between the observed and predicted values of the variable. The error variables of a DAG are explicitly causal in nature and represent all those causes of our variable other than the causes made explicit in the DAG. This results in the DAG in Figure 2.3b and in modified structural equations (Equations 2.2):

$$
\begin{aligned}
&\text{nylon fabric} = p_1(\cdot) \\
&\text{food} = p_2(\cdot) \\
&\text{parasites} = f_2(\text{nylon fabric}) + p_2(\varepsilon_2) \\
&\text{hematocrit} = f_3(\text{food, parasites, body mass}) + p_3(\varepsilon_3) \\
&\text{body mass} = f_4(\text{food, parasites}) + p_4(\varepsilon_4) \\
&\text{recruitment} = f_5(\text{hematocrit}) + p_5(\varepsilon_5)
\end{aligned}
\qquad \text{Equations 2.2(a-f)}
$$

Now, there are a few more definitions. All those variables in a DAG that do not have arrows pointing into them (i.e., that do not have explicit causal parents in the DAG) are called *exogenous* variables. The exogenous variables are at the periphery of the DAG. There are six exogenous variables in the DAG in Figure 2.3(b): food, nylon fabric, ε_2, ε_3, ε_4, and ε_5. All those variables in the DAG that have arrows pointing into them (i.e., that have causal parents in the DAG) are called *endogenous* variables. Endogenous variables are inside the DAG. There are four endogenous variables in the DAG in Figure 2.3(b): parasites, haematocrit, body mass and recruitment. Every variable in a DAG, and in the resulting structural equations, can be classified as either an exogenous or an endogenous variable.

All the rules concerning translating from causes to effects in DAGs still apply with respect to the error variables (ε). For instance, the fact that there are no directed paths linking ε_3 (the error

variable of haematocrit volume) and ε_4 (the error variable of body mass) in Figure 2.3b means that none of these unspecified causes (ε_3) of haematocrit (i.e., besides food, parasites) also cause body mass. If any of these unspecified causes did also cause changes in body mass, then we would have to add an arrow from ε_3 to body mass[29]. We could add such an arrow, but this would result in a different type of graph that is discussed in Chapter 6. In this chapter we restrict ourselves to DAGS.

There is an important conceptual difference between exogenous and endogenous variables. The values of endogenous variables are completely determined by their causal parents and the functions linking them. When I say "completely determined", I mean that we can perfectly predict the values of the endogenous variables given the values of their causal parents and the functions linking them. Ultimately, the values of the endogenous variables are completely determined by their exogenous causal ancestors and the system of functions linking them. You might think that this cannot possibly be true in practice since we can almost never perfectly predict the value of any variable in empirical science. The solution to this apparent contradiction is that at least one of the exogenous causal ancestors of any endogenous variable will be an error variable (ε) whose values we don't know. The exogenous variables are different. We can't *predict* the values of the exogenous variables since they have no explicit causes in the structural equations; we can only *observe* them as Nature presents them to us. We can only specify the probability that Nature will choose any particular value. This is what the $p_i(\cdot)$ functions do: they specify the probability distributions[30] of the exogenous variables.

These probability distribution functions , i.e., $p_i(\cdot)$, are mathematical equations that typically require certain parameters. These parameters can themselves be variables that measure the necessary properties of the distribution. There are many different probability distribution functions, and you are probably already familiar with some of them. For instance, the normal probability density function, appropriate for some continuous variables, requires two parameters, the mean (μ) and the standard deviation (σ), and is defined as: $p(X_i \mid \mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(X_i-\mu)^2}{\sigma^2}}$.

[29] Or to the error variable of body mass.
[30] If the variable is continuous then we can only specify the probability of observing a value within some infinitesimal interval around a specific value. We call the function describing this a probability density function.

A Poisson probability function, appropriate for some discrete count variables, requires only one parameter, the mean number of events that occur in each interval of time or space (λ), and is defined as: $p(X_i = k) = \frac{e^{-\lambda}\lambda^k}{k!}$.

Together, the DAG and these two types of functions ($f_i(\cdot)$, $p_i(\cdot)$) determine the structural equations. Together, the DAG and the two types of functions, $f_i(\cdot)$ and $p_i(\cdot)$, describe the *data generating mechanism* that we think exists in Nature. Because the exogenous variables are random variables, and because the endogenous variables are completely determined by the exogenous variables and the $f_i(\cdot)$ functions, the data generating mechanism also specifies the multivariate probability distribution followed by all of the variables in the DAG. At this point, we have translated from the language of graph theory to the language of probability distributions.

2.4 The shadow of a cause

In Chapter 1 I compared the act of inferring causal relationships from observational data to the act of inferring the shape of a three-dimensional object from its two-dimensional shadow. The shadow that a causal process throws onto data is the multivariate probability distribution that the structural equations generate. Let's make this analogy more visual. It is impossible to draw anything more than a bivariate probability distribution on a flat page because we would need to draw an object that has more than three dimensions, but we can draw "slices" through a multivariate object. Imagine that three variables (X, Y, Z) in Nature are causally linked according to the DAG X→Y→Z and that we have observed 1000 observations. The data generating mechanism is given in Equations 2.3(a – c). The resulting distribution is a trivariate normal distribution, with correlations[31] of 0.8 between X and Y and between Y and Z, and a correlation of 0.64 between X and Z.

[31] These are the values of the correlations in the probability distribution. The values of the correlations in our sample of 1000 observations will be slightly different because of sampling variation.

$$X = N(0,1)$$
$$Y = 0.8X + N\left(0, \sqrt{0.2}\right)$$ Equation 2.3(a-c)
$$Z = 0.8Y + N\left(0, \sqrt{0.2}\right)$$

A *marginal* distribution is a distribution of one or more variables that results when we sum over the remaining variables in the distribution. If, in our trivariate probability distribution, we look only at X, then the distribution of X alone (i.e., the marginal distribution of X) is the "shadow" of X that this causal process casts in our data. Figures 2.4(a,b,c) show the marginal distributions of X, Y and Z, which are all standard normal random variables with means of 0 and standard deviations of 1.

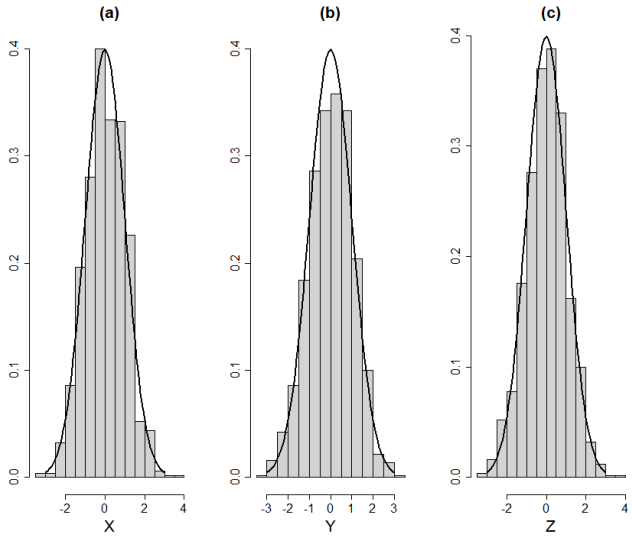


Figure 2.4. The marginal distributions of the three variables (X, Y and Z) in our DAG. The histogram shows the values in the sample of 1000 observations, and the solid line shows the values in the marginal Normal probability distribution.

If we look at the distribution of two variables together while summing over the remaining one, then this is called a joint (here, a bivariate) marginal distribution. In general, the joint distribution of more than one variable at a time is called a multivariate distribution. Figure 2.5 shows the joint marginal distributions of (X, Y), (X, Z) and (Y, Z) as contour plots. The plots of each of these joint marginal distributions is a 3-dimensional "hill" and a contour plot shows this hill as if you were looking down on it from above and plotting the contours of altitude. As expected, the most common combination of values of X and Z are at their respective means (0,

0) which are at the summit of the contour hill. Also as expected, since each of these variables are dependent and highly correlated, the next most common combinations of values are when both variables are higher than average or when both variables are lower than average.

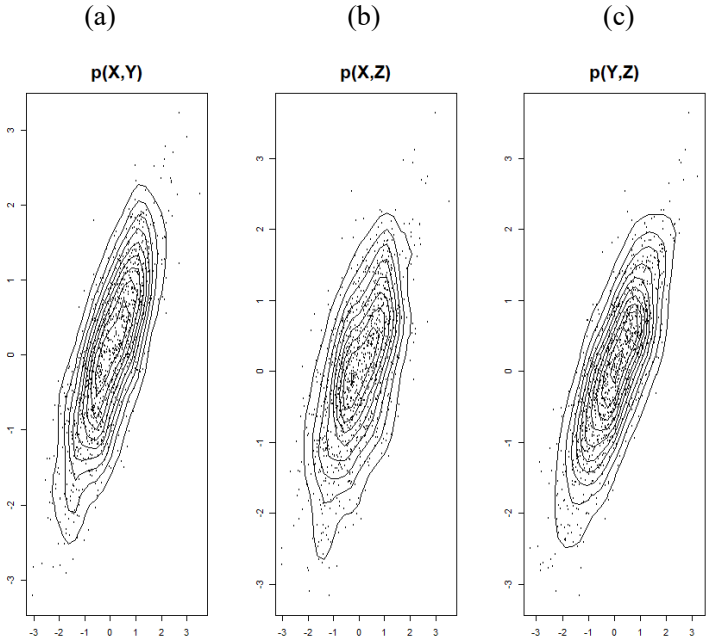


Figure 2.5. The joint (bivariate) marginal distributions of (X,Y), (X,Z) and (Y,Z) in our DAG.

If a marginal distribution is a distribution of a variable that results when we sum over (and, thus, ignore) other variables then a *conditional* distribution is the distribution of a variable that results when we take into account the values of other variables. If "p(Z)" is the marginal probability density of Z then the conditional probability density of Z, given Y, is written "p(Z|Y)"; the vertical line (|) means "*given"* (or "*conditional on"*). What does this mean? We know that the mean (μ_Z) of the marginal distribution of Z is zero: $\mu_Z=0$. Figure 2.5c shows marginal joint distribution of Y and Z. You will notice that the most likely value of Z changes depending on the value of Y. If $Y_i = -2$ then the most likely value of Z_i is approximately -2 but if $Y_i = 2$ then the most likely value of Z_i is approximately 1.5. Let's write the mean of Z *given* the value of Y, as "$\mu_{Z|Y}$". Since Z follows a normal probability distribution, this means that the mean of this normal probability distribution changes with (i.e., is conditional on) the value of Y. If fact, looking at Equation 2.3c, we know that the conditional mean of Z is $\mu_{Z|Y} = 0.8Y$. The distribution of Z conditional on (i.e., given) Y is not the same as the marginal distribution of Z (i.e., ignoring Y)

since the means are different. Going back to Equation 2.3c, we could rewrite Z as $Z = \mu_{Z|Y} + N\left(0, \sqrt{0.2}\right)$. The first part of Z ($\mu_{Z|Y}$) is not random; it is determined by the value of Y. The second part of Z (i.e. $N\left(0, \sqrt{0.2}\right)$) is still random, but it has a normal distribution with a mean of zero and a standard deviation of $\sqrt{2}$. This second part is the distribution of the random part of Z, *conditional on* Y. You might recognize this as the distribution of the residuals of Z given Y in a regression context. Just as you can look at the bivariate distribution of X and Z (Figure 2.5b), you can also look at the bivariate distribution of X and the conditional distribution of Z given Y (i.e., Z|Y) as shown in Figure 2.6. Notice that while X and Z clearly covary, this is not true when we condition Z on Y. Another way of saying this is that while X and Z are dependent, X is independent of Z conditional of Y.

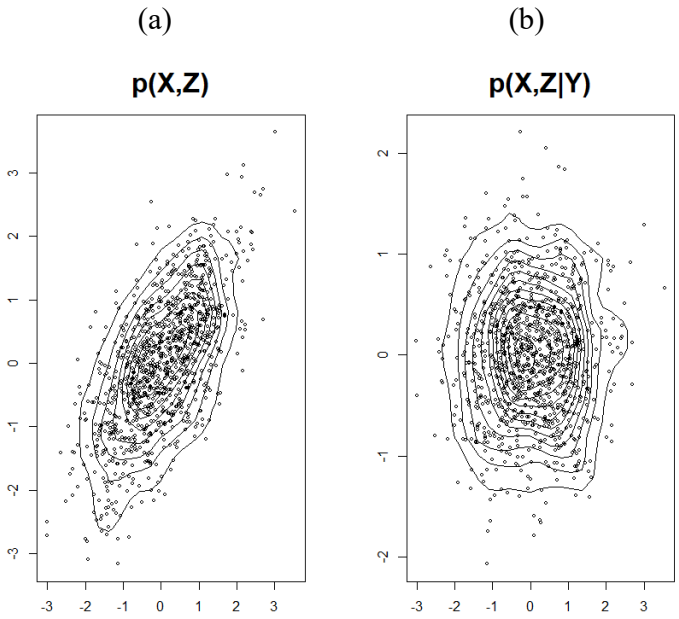


Figure 2.6. The joint marginal distribution of (X,Z) on the left and the joint conditional distribution of (X,Z) given Y on the right. Notice that there is a clear dependency between X and Z in the marginal distribution and no dependency between X and Z in the conditional distribution given Y.

Different DAGs, and thus data generating mechanisms, will cast different[32] shadows of dependence and (conditional) independence in our data. One way of describing structural

[32] This is not completely true due to the existence of "d-separation equivalent" DAGs; this is explained in Chapter 5.

equation modelling is that, given a hypothesized data generating mechanism, we predict how these shadows will change as we condition on different subsets of variables in our data and then test these predictions against our data.

2.5 Independence and (conditional) independence in multivariate probability distributions

Dependence and (conditional) independence are two important properties of multivariate probability distributions that are seen in the shadows generated by DAGs. In fact, the testing of predictions of dependence and (conditional) independence between different subsets of variables in a DAG is the heart of structural equation modelling. As you will see, a special operation on a DAG, called "d-separation", allows us to predict which subsets of variables will be (conditionally) dependent or (conditionally) independent in any multivariate probability distribution generated by that DAG. I will often stop writing "(conditionally)" in the rest of this book to make things simpler but remember that dependence and independence can exist in both the marginal distribution and/or in the conditional distribution. These predictions of dependence and independence are derived from the DAG alone and do not depend on any assumptions concerning the type of function ($f_i(\cdot)$) or the type of probability distribution ($p_i(\cdot)$) in the structural equations. Before explaining the notion of "d-separation", let's first understand what "dependence", "independence" and "conditional independence" means with respect to a multivariate probability distribution.

A probability distribution does not contain notions of "cause". Instead, it contains notions of "information". Imagine that you tell me that the body mass of a Blue Tit nestling follows a normal probability distribution with a given mean (say, 10 g) and standard deviation (say, 1 g). With this information, I won't know the body mass of the next Blue Tit nestling that I see but I will know that there is a probability of approximately 0.0026 of the next nestling will weight 7.2

g or less. We can write[33] this as $p(X_i \leq 7.2 \mid \mu = 10, \sigma = 1) \approx 0.0026$. If we define another random variable attribute of a Blue Tit nestling – say, the day of the week on which it hatches – then we have defined a joint probability distribution. A joint probability distribution is the distribution of observing two events together. In this case, the joint probability distribution p(X,Y) is the probability of observing a nestling that has a specific body mass (X) and that hatches on a specific day (Y). This notion can be extended to any number of variables, defining a multivariate probability distribution. If you then tell me that this nestling hatched on a Wednesday, and *if this added information does not change the probability that this nestling will weight 7.2 g or less*[34], then these two variables (X and Y) are independent. Intuitively, a random variable is independent of another random variable if knowledge about the value of one variable tells us nothing new about the value of the other variable that we didn't already know before we were given this added information. This relationship of independence in a joint probability distribution is symmetric, i.e., if X is independent of Y, then Y is also independent of X. This is true for every probability distribution. If I use the notation "p(X|Y)" to mean the probability of X given (|) the value of Y, then we can give a formal definition of probabilistic independence:

> Independence (definition): Given a joint probability distribution, p(X,Y), variable X is independent of variable Y if $p(X \mid Y) = p(X)$ and $p(Y \mid X) = p(Y)$. From this, it follows that $p(X,Y) = p(X)\,p(Y)$, i.e., the joint probability distribution of a set of independent variables is the product of their univariate marginal distributions.

If two variables are not independent, then they are dependent. Intuitively, this means that knowledge about the value of Y gives more information about X that we would have from X alone. Mathematically, this means that $p(X \mid Y) \neq p(X)$, $p(Y \mid X) \neq p(Y)$ and $p(X,Y) \neq p(X)\,p(Y)$. Consider the two variables "body mass" and "food intake". I said earlier that I won't know the body mass of the next Blue Tit nestling that I see but that I would know that there is only an approximately 0.0026 probability of the next nestling weighting 7.2 g

[33] In words: "the probability of observing a body mass of less than, or equal to, 7.2 g, given a normal distribution having a mean of 10 g and a standard deviation of 1 g, is approximately 0.0026".

[34] $p(X_i \leq 7.2 \,\&\, Y = Wednesday \mid \mu = 10, \sigma = 1) = p(X_i \leq 7.2 \mid \mu = 10, \sigma = 1) \approx 0.0026$

or less. This was from the marginal distribution of body mass. How might this probability change if you then tell me that this next nestling received less food than average? Since food causes an increase in body mass, less food should cause a smaller body mass. Therefore, knowing that this chick received less food should increase the probability that it weighed 7.2 g or less from the previous value of 0.0026. Because this probability changed when I learned that it received very little food, food and body weight are dependent.

Independence, as I have defined it above, is also called "unconditional" independence or "marginal" independence. Now that you have mastered the notion of independence, we next need to talk about the notion of "conditional" independence. In the Blue Tit DAG (Figure 2.3), the amount of food received by a chick causally affects its ability survive and return to the study site the next year (i.e. successful recruitment). Therefore, if I tell you how much food a chick receives, then this will change the probability that it will be successfully recruited into the population, i.e., "food" and "recruitment" are dependent variablesHowever, if I first give you information on its haematocrit volume, and *only then* tell you how much food it received, then knowing how much food it received would not add any new information at all concerning recruitment. Ater all, the causal parent of recruitment is haematocrit and, if we already know its value, then knowing something about food intake won't change these values. In other words, the probability of successful recruitment *given its haematocrit volume* would not change if we were also given its food intake. Written mathematically, p(recruitment | haematocrit, food) = p(recruitment | haematocrit). Because the first probability including information on food intake is the same as the second probability without information on food intake, we say that recruitment is independent of food intake conditional on (or given) haematocrit. This means that two variables can be unconditionally dependent (food intake and recruitment) but conditionally independent just as food and recruitment are unconditionally dependent but are conditionally independent given information on haematocrit volume.

> Conditional independence (definition): . Given a joint probability distribution, p(X,Y,Z), variable X is conditionally independent of variable Y given variable Z if $p(X \mid Y \& Z) = p(X \mid Z)$ and $p(Y \mid X \& Z) = p(Y \mid Z)$. From this, it follows that $p(X, Y \mid Z) = p(X \mid Z)\, p(Y \mid Z)$.

Before leaving the notions of independence and conditional independence, I want to give one more visual illustration of these notions that some people find instructive. Consider again the DAG (X→Y→Z) and the generating equations 2.3(a-c). I have generated 10,000 observations from these equations and the top left panel of Figure 2.7 shows the bivariate relationship between X and Z in all 10,000 observations. The values of Y ranged from -4.2 to 3.6. In the top right panel of Figure 2.7, I have restricted the range of variation of Y to between ±1; i.e., I have excluded all those observations whose value of Y was outside of this range. In the bottom two panels, I have restricted the range of variation of Y to between ±0.5 (bottom left) and to between ±0.1 (bottom right). As I restricted the range of Y more and more, we know more and more about the likely value of Y (because we know that it lies within a more and more restricted range). Simultaneously, as I restricted the range of Y more and more, he correlation between X and Z decreases. In the limit, if we restricted the range of variation of Y to zero (i.e., if we knew the exact value of Y), then the correlation between X and Z would be zero. Stated equivalently, X and Z are strongly correlated in their marginal bivariate distribution but are independent in their bivariate conditional distribution given (i.e., conditional on) Y

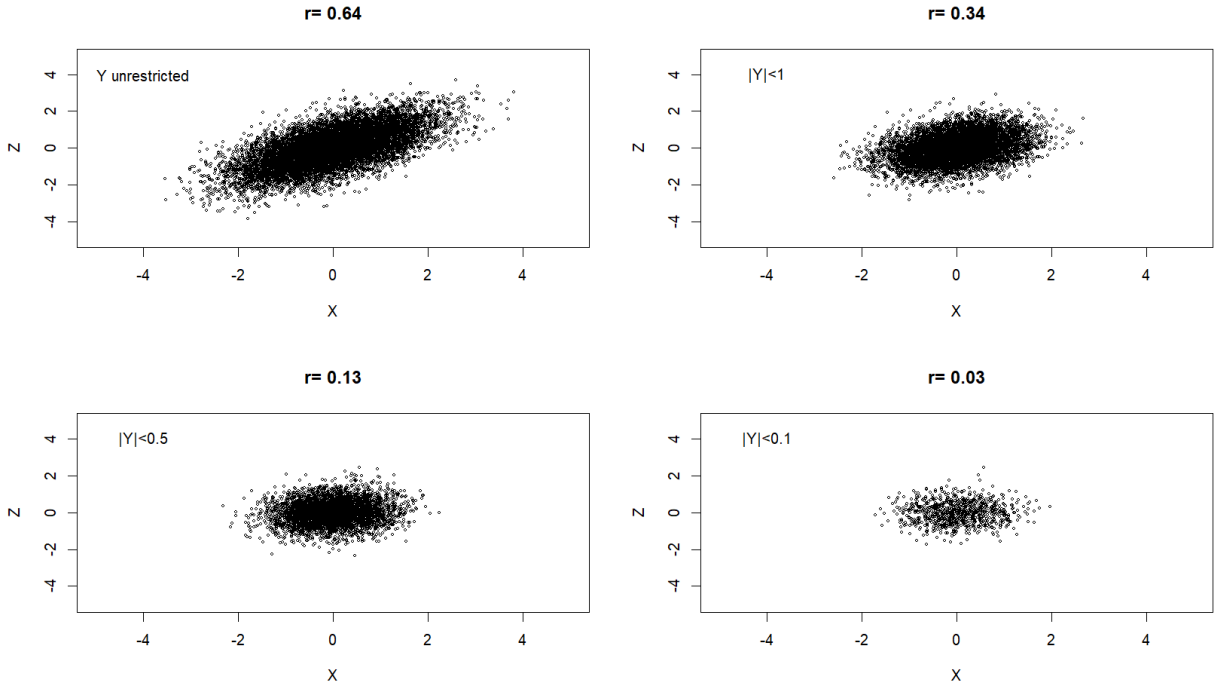


Figure 2.7. The joint distribution of X and Z, and the degree of correlation between them, as we restrict the range of Y more and more.

2.6 D-separation in a DAG: the universal translator from DAGs to probability distributions

Amazingly, it is possible to determine dependence, independence, conditional dependence, and conditional independence. between random variables directly from the DAG without knowing anything about the equations describing the generating mechanism of the data. This is done using a manipulation of the DAG called *d-separation* (short for *separation in a directed acyclic graph*). This was not possible until relatively recently, when it was discovered by Pearl (1988). Before explaining d-separation, there is an important fact about d-separation that you must know:

> *If two variables are d-separated in a DAG, then these two variables will always be independent in the multivariate probability distribution that is generated by the data generating mechanism associated with the DAG. This is true irrespective of the functional forms of the equations in the data generating mechanism (i.e., $f_i(\cdot)$) and irrespective of the probability distributions of the exogenous variables (i.e., $g_i(\cdot)$).*

This is a very powerful and useful fact! Using d-separation, we can predict everything about the independence relationships among every possible subset of variables in our DAG and almost everything about the dependence[35] relationships among every possible subset of variables. That means that we can predict the probability "shadows" that must exist in our data given only the DAG and without knowing anything about the details of the data generating equations.

In order to explain d-separation, we will use the definition of an "undirected" path that you have already learned[36]. We will also need two new definitions. A *non-collider* variable in an undirected path is a variable that does not have arrows pointing into it from both directions; in other words, a pattern of three variables in the undirected path like this: $X_i \rightarrow X_j \rightarrow X_k$, $X_i \leftarrow X_j \rightarrow X_k$, or $X_i \leftarrow X_j \leftarrow X_k$. A *collider* variable in an undirected path is a variable that does have arrows pointing into it from both directions; in other words, a pattern of three variables in

[35] It is possible for subsets of variables to be statistically independent even though they are not d-separated but only in very exceptional situations in which the causal effects along different paths exactly cancel out. When this happens, we say that the probability distribution is *unfaithful* to the DAG. This is explained in more detail in Chapter 9.

[36] An undirected path between any two variables, X_i and X_j, in a DAG is a set of variables that you can traverse when *ignoring* the directions of the arrows in order to pass from X_i to X_j and without visiting the same variable more than once.

the undirected path like this: $X_i \rightarrow X_j \leftarrow X_k$. Note that classification of a variable as a collider or a non-collider only applies for a given undirected path; the same variable can be a collider along one undirected path and a non-collider along a different undirected path . If two variables[37] (X_i, X_j) are d-separated in a DAG given some set **C** of conditioning variables (**C**=$\{X_1,\ldots,X_n\}$), where the conditioning set cannot include either X_i or X_j, then this is written[38] as "$X_i \_||\_ X_j|\mathbf{C}$". The set of conditioning variables can also be "no variables", i.e., the null set; this is written "**C**= $\{\phi\}$".

You can think of d-separation as a device that tells us how to convert the flow of causal information in a DAG into the flow of statistical information. It is a translation device between the language of causality and the language of probability distributions. A non-collider variable in a path allows the flow of causal effects to pass through it. Thus, $X_i \rightarrow X_j \rightarrow X_k$ in a DAG means that the causal effects from X_i can pass through X_j and into X_k. Physically holding constant X_j will block the flow of causal effects from X_i to X_k making X_k causally independent of Xi as long as X_j is prevented from changing. Statistically holding constant X_j, i.e., statistically conditioning on X_j, blocks the flow of *statistical* information through X_j. So, given an undirected path like $X_1 \rightarrow X_2 \rightarrow \cdots X_j \rightarrow \cdots X_n$, statistically conditioning on *any* of the non-colliders will block the flow of statistical (as well as causal) effects from X_1 to X_n.

For a non-collider variable , physical control and statistical control result in the same prediction concerning dependence or independence. As a physical example, let's say that more food will cause an increased body mass of a Blue Tit chick, which will increase its haematocrit, which will cause an increased chance of successful recruitment (food→bodymass→haematocrit→recruitment). Therefore, we will observe both a causal and a statistical dependency between food intake and recruitment success. If we could physically hold constant body mass irrespective of the amount of food received then food intake and recruitment success would become causally independent along this path; in terms of recruitment success, it wouldn't matter how much food the chick received if this didn't change its body mass[39]. The

[37] The operation of d-separation also applies to subsets of variables in the DAG, not just pairs of variables.
[38] There is some confusion in the current literature surrounding this notation. Most authors use "_||_" to mean "d-separation" while "_|_" means probabilistic independence, and this is the notation that I will use.
[39] Of course, given the DAG in Figure 2.1, food also directly causes a change in haematocrit and so the path food→haematocrit→recruitment success would still be open.

same thing would happen if we could physically hold constant the haematocrit volume or if we held constant both body mass and haematocrit.

The statistical analogue to physically holding constant a variable is to statistically "fix" or "hold constant", or "condition on", the variable. The notion of "conditioning on" a random variable in the language of probability is a subtle one but it is important in what follows. To statistically "condition on" a random variable means to analyse the distribution or behaviour of one or more other random variables given that the value of the conditioning variable is no longer random but, instead, is known or fixed at a specific value. Remember that a random variable is one whose precise value we don't know until we observe it. Once we are told its value, the variable stops being random and its value is now "fixed" or "held constant" at a precise value. Look again at the DAG: food→bodymass→haematocrit→recruitment. If, rather than physically holding constant body mass, we were only *told* the body mass of the chick (i.e., if we statistically condition on it so that it is fixed rather than random) then food intake and recruitment success would also become statistically independent along this path. Why? Once we know the value of body mass (i.e. once we have conditioned on body mass), then we can predict haematocrit volume and then recruitment success. Learning how much food the chick received tells us nothing about the value of body mass – and thus haematocrit volume or recruitment success – that we didn't already know. Learning how much food the chick received would be irrelevant to our subsequent prediction of recruitment success. "Irrelevance" of information means "independence" in the language of probability. In the case of a non-collider, physical control and statistical control give the same answer . We can now state the first important rule of d-separation: *Statistically conditioning on a non-collider variable in an undirected path blocks the flow of statistical information through it along the undirected path.* Notice that physically holding constant a non-collider variable in an undirected path also blocks the flow of causal information through it along the undirected path.

Unlike the case of a non-collider, statistically conditioning on a collider variable does *not* result in the same thing as physically controlling it. Given $X_i \rightarrow X_j \leftarrow X_k$, the two variables X_i and X_k are causally independent since changing the value of X_i won't change the value of X_k and *vice versa*. Physically holding constant X_j will not change this causal independence. However, if we know (i.e, are told, or observe) the value of X_j (i.e., statistically holding X_j constant or

conditioning on it) then this will make X_i and X_k statistically dependent even though they are still causally independent. This might seem counterintuitive but remember that d-separation is telling us the consequences of *statistical* control (probabilistic conditioning), not the consequences of *physical* control and probability distributions can't speak the language of causality. To see this, consider a causal scenario in which both food intake and parasite load cause changes in haematocrit volume, and food intake is causally independent of parasite load. The DAG is: food intake→haematocrit←parasite load. To say that food intake is causally independent of parasite load means that if we physically change either one, the other does not change. If we experimentally doubled the number of parasites per nest, then this would not affect the number of caterpillars (the food source) around a nest. Food intake and parasite load are also statistically independent if we condition on nothing: if I tell you how much food a chick receives, this tells you nothing new about its parasite load in this scenario. However, statistically conditioning on haematocrit (i.e., holding haematocrit statistically constant) will make food intake and parasite load statistically dependent. Why? I if I tell you that a chick has a very small haematocrit (now, I have statistically conditioned on haematocrit) and I then tell you that it has no parasites, this information about parasite load will give you new information about how much food the chick likely received. After all, the combination of a chick with a small haematocrit and no parasites would likely only occur if it didn't have much food. In other words, food intake and parasite load are statistically independent if we don't know anything about haematocrit, but they become statistically dependent once we do know about haematocrit. Physically controlling haematocrit and statistically controlling haematocrit (a collider variable along the path food→haematocrit←parasites) do not result in the same prediction. Things get even more complicated. Since haematocrit causes recruitment success (haematocrit→recruitment success) then, if we are told the recruitment success of a chick then this too will generate a statistical dependency between food intake and parasite load! Why? If I tell you that a chick did not return to the population the next year (unsuccessful recruitment) then you have some new information about its haematocrit volume (probably low). If I then tell you that it has no parasites (i.e., I have statistically conditioned on parasite load) then you have new information about how much food the chick likely received (probably little). Therefore, statistically conditioning on the descendent of a collider variable has the same consequence as statistically conditioning on the collider itself. We can now state the second important rule of d-separation,: *Statistically*

conditioning on a collider variable in an undirected path, or on a descendent of this collider variable, allows the flow of statistical information to flow through it along the undirected path.

Once you have learnt the two rules about statistically conditioning on a collider and a non-collider variable, you are now ready to learn the steps to determine if two variables (X_i, X_j) are d-separated in a DAG given a set **C** of conditioning variables . The set **C** can contain no conditioning variables (i.e. the set is empty) or it can contain any number of conditioning variables as long as this set doesn`t also include either X_i or X_j.

1. List all undirected paths between X_i and X_j.

For each undirected path between X_i and X_j in the DAG:

2. Look at the set of variables in **C** and the variables between X_i and X_j along this undirected path. If any non-collider in this undirected path is also in **C**, then stop and conclude that X_i and X_j are d-separated along this undirected path. If not, then go to the next step.

3. Look at the set of variables in **C** and the variables between X_i and X_j along this undirected path. If any collider in this undirected path is not in **C** *nor are its descendants* in **C**, then stop and conclude that X_i and X_j are d-separated along this undirected path.

4. If the undirected path is still not d-separated after step 3 then X_i and X_j are not d-separated along this path.

If every undirected path between X_i and X_j in the DAG is d-separated, then X_i and X_j are d-separated in the DAG. If any undirected path between X_i and X_j in the DAG is not d-separated, then X_i and X_j are not d-separated in the DAG. Remember that if two variables are d-separated in the DAG then these two variables will always be independent in the data generated by the DAG. It is very easy to use the operation of d-separation once you have memorised the four steps. Most students can master this task after a few minutes of practice, even though it is cumbersome to explain in words. We will use the DAG in Figure 2.8 to illustrate d-separation.

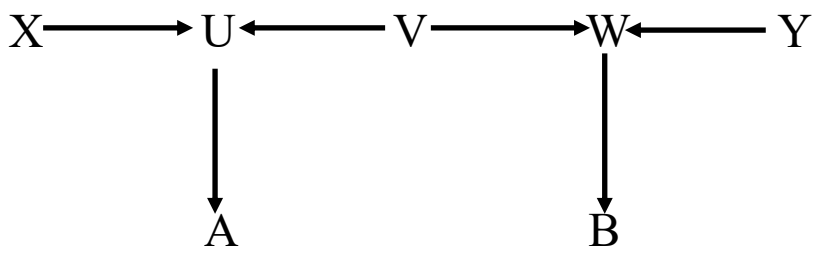


Figure 2.8. A DAG to use in exploring d-separation.

Is it true that V is d-separated from Y if we condition on nothing; i.e. V_||_Y|{ϕ]?

- First, we must get every unique undirected path linking V and Y. There is only one in this DAG: V→W←Y (remember, we are ignoring the directions of the arrows in an undirected path).
- Second, we look at each non-collider variable along this undirected path and see if any are in the conditioning set (which is empty). There are no non-colliders in this undirected path so we go to the next step.
- Third, we look at each collider variable along this undirected path; there is one: W. Now, we must see if this collider variable (i.e., W) is also in the conditioning set. It is not since the conditioning set is empty.
- Finally, we must also see if any descendant of this collider is in the conditioning set. The only descendent of W is B and B isn't in the conditioning set either. Therefore, it is true that V_||_Y|{ϕ}, i.e., V is d-separated from Y given no conditioning variables.

Is it true that U is d-separated from W if we condition on V; i.e. U_||_W|{V}?

- There is only one undirected path between U and W: U←V→W.
- V is a non-collider along this undirected path. V is also in the conditioning set. Therefore, this path is d-separated. As it is the only undirected path, then U is d-separated from W, conditional on V.

We already determined that V_||_Y|{ϕ} is true. Is it true that V_||_Y|{W}?

- There is only one undirected path between V and Y: V→W←Y.
- There are no non-colliders along this path, so we go to the next step.

- There is one collider along this path: W. However, W is also in the conditioning set. Therefore, it is not true that V_||_Y|{W}. We write this as "~V_||_Y|{W}.

We already determined that V_||_Y|{ϕ} is true and that V_||_Y|{W} is not. Is it true that V_||_Y|{B}?

- There is only one undirected path between V and Y: V→W←Y.
- There are no non-colliders along this path so we go to the next step.
- There is one collider along this path: W. W is not in the conditioning set (i.e., B) so we go to the next step.
- We must look at all the descendants of W and there is only one: B. B is in the conditioning set. Therefore, it is not true that V_||_Y|{W}; i.e., ~V_||_Y|{W}.

Once you have practiced a bit, you will be able to quickly read off the d-separation relationships in any DAG, which means that you will be able to determine unconditional and conditional dependence and independence of any pair of variables, conditioned on any possible set of other variables, simply by looking at the DAG. To get some practice, look at the first column of Table 2.1, which lists some d-separation claims derived from Figure 2.8. Try to determine if each d-separation claim is true or false and why. Then see if your answer and explanation agree with the second column of Table 2.1.

Table 2.1. An incomplete set of d-separation relationships between pairs of variables in the DAG shown in Figure 2.8 and the explanations for these relationships.

d-separation claim	Explanation
X_||_V| ϕ}	There is one undirected path: X→U←V. U is a collider and is not in the conditioning set.
~ X_||_V|{U}	There is one undirected path: X→U←V. U is a collider, and it is in the conditioning set.
~X_||_V|{A}	There is one undirected path: X→U←V. A is a collider and it is not in the conditioning set, but A is a descendent of A and A is in the conditioning set.

~U_||_W{ϕ}	There is one undirected path: U→V←W. V is a non-collider, and it is not in the conditioning set.
U_||_W|{V}	There is one undirected path: U→V←W. V is a non-collider, and it is in the conditioning set.
X_||_Y{ϕ}	There is one undirected path: X→U←V→W←Y. Both U and W are colliders, and neither is in the conditioning set.
X_||_Y{W}	There is one undirected path: X→U←V→W←Y. Both U and W are colliders, and U is not in the conditioning set.
X_||_Y{U}	There is one undirected path: X→U←V→W←Y. Both U and W are colliders, and W is not in the conditioning set.
~X_||_Y{U,W}	There is one undirected path: X→U←V→W←Y. Both U and W are colliders, and both U and W are in the conditioning set.
~X_||_Y{B,U}	There is one undirected path: X→U←V→W←Y. Both U and W are colliders. U is in the conditioning set and B, which is in the conditioning set, is a descendent of W.
~X_||_Y{B,A}	There is one undirected path: X→U←V→W←Y. Both U and W are colliders. Neither U nor W are in the conditioning set but B is a descendent of W and A is a descendent of U.
X_||_Y|{U,W,V}	There is one undirected path: X→U←V→W←Y. Both U and W are colliders, which opens up these variables, but V is a non-collider and so conditioning on it blocks the undirected path.

The ggm package of R (Marchetti et al. 2024) contains functions that automate the d-separation operation. To enter a DAG, you use the `DAG()` function of the ggm package. You enter each endogenous variable[40], the tilde operator (~), and then the causes of this variable in the DAG. Here is how to enter[41] the DAG shown in Figure 2.8:

[40] If an exogenous variable (X) in a DAG causes nothing, then you would enter this as "X~1".
[41] If you want a graphical display of this DAG, or any DAG that has been created using the DAG() function, you can do this by typing: `drawGraph(Figure2.8_DAG)`; this function is in the ggm package.

```
Figure2.8_DAG<-DAG(U~X+V,A~U,W~V+Y,B~W)
```

To determine if any two variables are d-separated in this DAG, you can use the `dsep ()` function of the ggm package, whose four arguments are (1) amat (the name of the DAG object), (2) first (the name of the first variable of the pair in quotes), (3) second (the name of the second variable of the pair in quotes), and (4) cond (a character vector containing the names of the conditioning variables, each in quotes). For instance, here are the first and last d-separation claims that listed in Table 2.1:

```
dsep (amat=Figure2.8_DAG,first="X",second="V",cond=c())
[1] TRUE
dsep (amat=Figure2.8_DAG,first="X",second="Y",cond=c("U","W","V"))
[1] TRUE
```

Now, using the DAG shown in Figure 2.2, test yourself by answering (TRUE/FALSE) the following d-separation claims and then verify your answer by using the `dsep()` function: X_1_||_X_2|{X_3}, X_1_||_X_3|{X_2}, X_1_||_X_5|{X_3}, X_1_||_X_5|{X_3,X_4}, X1_||_X_5|{X_2}, X_3_||_X_4|{X_2}, X_3_||_X_4|{X_2.X_5}.

What have we accomplished? We have translated from our scientific language of causality into the mathematical language of DAGs. We have then translated from the mathematical language of DAGs into the official language of statistics: probability distributions. Finally, using d-separation, we can accurately predict the statistical shadows (i.e., the patterns of dependence, independence, conditional dependence and conditional independence) of the resulting probability distribution given our causal hypothesis. Now, we can use the same inferential logic as the controlled experiment, described in Chapter 1, to test our causal hypothesis by replacing physical control by statistical control. Given each possible pair of variables (X_i, X_j) in the DAG, d-separation will generate a statistical prediction concerning whether these two variables will be independent or not, given every possible subset of the other variables in the DAG. Each of these predictions can be statistically tested and all of them must be true in your data if your data really were generated by your hypothesized DAG. In fact, the testing of these claims of (conditional) independence is the heart of structural equations modelling.

2.7 The Markov condition

All probability distributions that are generated from DAGs have an important statistical property: they obey the Markov condition (Pearl 2000). The Markov condition states that every variable, X_i, in a DAG is independent of all its non-descendants, given its causal parents. Stated differently, once you know the value of the causal parents of X_i, then knowing anything about its more distant causal ancestors, or of any other variable except its descendants, tells you nothing new. Because the joint distribution of two variables that are independent is the product of each of their univariate distributions, this means that we can replace the complicated multivariate probability distribution that is generated by the DAG into the product of a series of univariate distributions. Univariate distributions are much easier to use than are multivariate distributions. In these univariate distributions, each variable is conditional on its causal parents and the exogenous variables are not conditional on anything (since they have no causal parents). For example, the multivariate probability distribution that is generated by the DAG in Figure 2.7 can be expressed as the product of the series of conditional univariate distributions as shown in Equation 2.4. This is advantageous because it is much easier to work with univariate distributions. As we will see in Chapter 3, this also leads to a way of choosing a small subset of d-separation predictions that, together, imply all the others. This means that we only have to test this small subset of d-separation claims in order to test the whole set of d-separation claims implied by the DAG.

$$p(X,U,A,V,W,B,Y) = p(X)\,p(V)\,p(Y)\,p(U \mid X,U)\,p(A \mid U)\,p(W \mid V,Y)p(B \mid W) \quad \text{Equation 2.4}$$

2.8 Selection bias and conditioning on a collider

Understanding d-separation allows us to explain the phenomenon of selection bias in statistical analysis. Selection bias occurs when the observations that are included and excluded in the data set are influenced (often unknowingly) by some variable that is not explicitly included in the analysis. Imagine that a school decides which applicants can admitted as students based on the

sum of their academic scores and their athletic scores; therefore, students must have some minimum academic score and/or some minimum score of athletic ability. The DAG is: academic score→student←athletic score. Even if academic and athletic abilities are independent in the general population, they will be dependent in the student population of the school. Why? Because to be included in the data, an individual has to be a student. That means that we have implicitly conditioned on the collider variable (student=yes). Any other attributes of students that determine their academic or athletic abilities will also be dependent within the student population even if this is not true in the general population. This phenomenon is probably more common in organismal biology than is recognized. When we measure trait values of organisms and look for correlations between them, we generally only use those organisms that are alive. If these traits also determine survival, and if we are only using individuals that have survived (survival=YES), then this means that we are implicitly conditioning on a collider (survival) in the DAG generating our data: X→survival←Y. In such cases, dependencies between these traits might not be generated by any causal link between them but solely due to a selection bias caused by conditioning on a collider variable (survival).

2.9 The logic of causal inference

Now that we have our translation device (d-separation) and are aware of some of the counterintuitive results that can occur with d-separation (for instance, conditioning on a collider variable), we must be able to infer causal consequences from observational data by using this translation device. The details of how to carry out such inferences will occupy the rest of this book. Before looking at the statistical details, I want you to consider the logic of causal and statistical inferences. This will involve a brief dive into the philosophy of science.

Since we are talking about the logic of inferences from empirical experience, it is useful to briefly look at what philosophers of science have had to say about valid inference. Logical positivism, itself being rooted in the British empiricism of the 19th century that so influenced

people like Pearson[42], was dominant in the 20th century up to the 1960s. This philosophical school was based on the verifiability theory of meaning; to be meaningful, a statement had to be of a kind that could be shown to be either true or false. For logical positivism, there were only two kinds of meaningful statements. The first kind was composed of *analytical* statements (tautologies, mathematical or logical statements) whose truth could be determined by deducing them from axioms or definitions. The links between graph theory and probability theory, that I described in this Chapter, are an example of such analytical statements. The second kind was composed of *empirical* statements that were either self-evident observations ("the water is 23°C") or could be logically deduced from combinations of basic observations whose truth was self-evident[43]. Thus, logical positivists emphasised the hypothetico-deductive method: A hypothesis was formulated to explain some phenomenon by showing that it followed deductively from the hypothesis. The scientist attempted to validate the hypothesis by deducing logical consequences of the hypothesis that were not involved in its formulation and testing these against additional observations. A simplified version of the argument goes like this:

- If my hypothesis is true, then consequence C must also be true. For example, if the data were generated by my DAG, then X is d-separated from Y given **Z**. Therefore, X will be independent of Y given **Z** in my data.
- Consequence C is true.
- Therefore, my hypothesis is true; the data really were generated by my hypothesized DAG.

Readers will immediately recognise that such an argument commits the logical fallacy of affirming the consequent. It is possible for the consequence to be true even though the hypothesis that deduced it is false since there can always be other reasons for the truth of C. Popper (1980) pointed out that, although we cannot use such an argument to verify hypotheses, we can use it to reject them without committing any logical fallacy:

[42] This is explored in more detail in Chapter 3.
[43] That even such simple observational or experiential statements cannot be considered objectively self-evident was shown at the beginning of the twentieth century by (Duhem 1914).

- If my hypothesis is true, then consequence C must also be true; if the data were generated by my DAG, then X is d-separated from Y given **Z**. Therefore, X will be independent of Y given **Z** in my data.
- Consequence C is false. X is not independent of Y given **Z** in my data.
- Therefore, my hypothesis is false; the data were not generated by my hypothesized DAG.

The criterion of falsifiability of a hypothesis is a hallmark of Popper's philosophy of science. Practising scientists would quickly recognise that this argument, although logically acceptable, has important shortcomings when applied to empirical studies. It was recognised as long ago as the turn of the twentieth century (Duhem 1914) that no hypothesis is tested in isolation. Every time that we draw a conclusion from some empirical observation we rely on a whole set of auxiliary hypotheses (A_1, A_2….) as well. Some of these have been repeatedly tested so many times and in so many situations that we scarcely doubt their truth. Other auxiliary assumptions may be less well established. These auxiliary assumptions will typically include ones concerning the experimental or observational background, the statistical properties of the data, and so on. Did the experimental control really prevent the variable from changing? Where the data really normally distributed, as the statistical test assumes? Such auxiliary assumptions are legion in every empirical study including the randomised experiment, the controlled experiment or the methods described in this book involving statistical controls. A large part of every empirical investigation involves checking, as best one can, such auxiliary assumptions so that, once the result is obtained, blame or praise can be directed at the main hypothesis rather that at the auxiliary assumptions.

So, Popper's process of inference might be simplistically paraphrased[44] as:

- If auxiliary hypotheses A_1, A_2, …A_n are true, and
- If my hypothesis is true, then consequence C must be true.
- Consequence C is false.
- Therefore, my hypothesis is false.

[44] *Simplistic* because it is wrong. Popper did not make such a claim.

Unfortunately, to argue in such a manner is also logically fallacious. Consequence C might be false, not because the hypothesis is false, but rather because one or more of the auxiliary hypotheses are false. The empirical researcher is now back where he started: there is no way of determining either the truth or falsity of his hypothesis in any absolute sense from logical deduction. This conclusion applies just as well to the randomised experiment, the controlled experiment or the methods described in this book. Yet, most biologists would recognise the falsifiability criterion as important to science and would probably modify the simplistic paraphrase of Popper's inference by attempting to judge which, the auxiliary hypotheses and background conditions, or the hypothesis under scrutiny, is on firmer empirical ground. If the auxiliary assumptions seem more likely to be true than the hypothesis under scrutiny, yet if the data do not accord with the predicted consequences, then the hypothesis would be tentatively rejected. If there are no reasoned arguments to suggest that the auxiliary assumptions are false, and the data also accord with the predictions of the hypothesis under scrutiny, then the hypothesis would be tentatively accepted. Pollack (1986) calls such reasoning *defeasible* reasoning[45]. Revealingly, practising scientists have explicitly described their inferences in such terms for a long time. At the turn of the 20th century T. H. Huxley likened the decision to accept or reject a scientific hypothesis to a criminal trial in a court of law (Rapport and Wright 1963) in which guilt must be demonstrated beyond reasonable doubt.

Let's apply this reasoning to the examples in Chapter 1 involving the randomised and the controlled experiments. Later, I will apply the same reasoning to the methods involving statistical control. Here is the logic of causal inference with respect to the randomised experiment to test the hypothesis that fertiliser addition increases seed yield:

- If the randomisation procedure was properly done so that the alternate causal explanations were excluded;
- If the experimental treatment was properly applied;
- If the observational data do not violate the assumptions of the statistical test;
- If the observed degree of association was not due to sampling fluctuations;

[45] *Defeasible* because it can be *defeated* with subsequent evidence.

- Then by the causal hypothesis the quantity of seed produced will be associated with the presence of the fertiliser.
- There is/is not an association between the two variables.
- Therefore, the fertiliser addition might have caused/did not cause the increased seed yield.

This list of auxiliary assumptions is only partial. In particular, we still have to make the basic assumption linking causality to observational associations, as described in Chapter 1. At this stage we must either reject one of the auxiliary assumptions or tentatively accept the conclusion concerning the causal hypothesis. If the probability associated with the test for the association is sufficiently large[46], traditionally above 0.05, then we are willing to reject one of the auxiliary assumptions (the observed measure of association was not due to sampling fluctuations) rather than accept the causal hypothesis. Thus, we reject our causal hypothesis. This rejection must remain tentative. This is because another of the auxiliary assumptions (not listed above) is that the sample size is large enough to permit the statistical test to differentiate between sampling fluctuations and systematic differences. Note, however, that it is not enough to propose any old reason to reject one of the auxiliary assumptions; we must propose a reason that has empirical support. We must produce *reasonable* doubt – in the context of the assumption concerning sampling fluctuations scientists generally require a probability above 0.05. Here it is useful to cite from the first edition of Fisher (1925) influential *Statistical methods for research workers*: "Personally, the writer prefers to set a low standard of significance at the 5 per cent point and ignore entirely all results which fail to reach this level. A scientific fact should be regarded as

[46] See Cowles and Davis (1982b) for a history of the 5% significance level. The first edition of Fisher (1925) classic book states on page 47: "It is convenient to take this point as a limit in judging whether a deviation is to be considered significant or not. Deviations exceeding twice the standard deviation are thus formally regarded as significant". The words "convenient" and "formal" emphasize the somewhat arbitrary nature of this value. In fact, this level can be traced back even further to the use of 3 times the probable error (about 2/3 of a standard deviation). Strictly speaking, twice the standard deviation of a normal distribution gives a probability level of 0.0456; perhaps Fisher simply rounded this up to 0.05 for his tables. Pearson and Kendall (1970) records Pearson's reasons at the turn of the century: p=0.5586 "thus we may consider the fit remarkably good"; p=0.28 "fairly represented"; p=0.10 "not very improbable"; p=0.01 "this very improbable result". Note that some doubt began at 0.1 and Pearson was quite convinced at p=0.01. The midpoint between 0.1 and 0.01 is 0.05. Cowles and Davis (1982a) conducted a small psychological experiment by fooling students into believing that they were participating in a real betting game (with money) that was, in reality, fixed. The object was to see how unlikely a result people would accept before they began to doubt the fairness of the game. They found that "on average, people do have doubts about the operation of chance when the odds reach about 9 to 1 [i.e. 0.09], and are pretty well convinced when the odds are 99 to 1 [i.e. ~0.0101]… If these data are accepted, the 5% level would appear to have the appealing merit of having some grounding in common sense".

experimentally established only if a properly designed experiment rarely fails to give this level of significance". Fisher was demanding reasonable doubt concerning the null hypothesis since he asks only that a result "rarely fail" to reject it. What if the probability of the statistical test was sufficiently small, say 0.01, that we do not have reasonable grounds to reject our auxiliary assumption concerning sampling fluctuations? What if we do not have reasonable grounds to reject the other auxiliary assumptions? What if the sampling variation was small compared to a reasonable effect size? Then we must tentatively accept the causal hypothesis. Again, this acceptance must remain tentative since new empirical data might provide such reasonable doubt.

Is there any automatic way of measuring the relative support for or against each of the auxiliary assumptions and of the principal causal hypothesis? No. Although the support (in terms of objective probabilities) for some assumptions can be obtained – for instance, those concerning normality or linearity of the data – there are many other assumptions that deal with experimental procedure or lack of confounding variables for which no objective probability can be calculated. This is one reason why so many contemporary philosophers of science prefer Bayesian methods to frequency-based interpretations of probabilistic inference; see, for example, Howson and Urbach (1989). Such Bayesian methods suffer from their own set of conceptual problems (Mayo 1996). In the end, even the randomised experiment requires subjective decisions on the part of the researcher. This is why the independent replication of experiments in different locations, using slightly different environmental or experimental conditions and therefore having different sets of auxiliary assumptions, is so important. As the causal hypothesis continues to be accepted in these new experiments, it becomes less and less reasonable to suppose that incorrect auxiliary assumptions are conspiring to give the illusion of a correct causal hypothesis.

Here is the logic of our inferences with respect to the controlled experiment to test the hypothesis that renal activity causes the change in the colour of the veinal blood, described in Chapter 1:

- If the activity of the kidney was effectively controlled;
- If the colour of the blood was accurately determined;
- If the experimental manipulation did not change some other uncontrolled attribute besides kidney function that is a common cause of the colour of blood in the renal vein before entering, and after leaving the kidney;

- If there was not some unknown (and therefore uncontrolled) common cause of the colour of blood in the renal vein before entering, and after leaving the kidney;
- If a rare random event did not occur;
- Then by the causal hypothesis, blood will change colour only when the kidney is active.
- The blood did change colour in relation to kidney activity.
- Therefore, kidney activity does cause the change in the colour of blood leaving the renal vein.

Again, this list of auxiliary assumptions is only partial. Again, one must either produce reasonable evidence that one or more of the auxiliary assumptions is false or tentatively accept the hypothesis. In particular, more of these auxiliary assumptions concern properties of the experiment or of the experimental units for which we cannot calculate any objective probability concerning their veracity. This was one of the primary reasons why Fisher considered the controlled experiment as inferior. In the controlled experiment these auxiliary assumptions are more substantial, but it is still not enough to raise any old doubt; there must be some empirical evidence to support the decision to reject one of these assumptions. Since we want the data to cast doubt or praise on the principal causal hypothesis and not on the auxiliary assumptions, we will only ask for evidence that casts reasonable doubt. It is not enough to reject the causal hypothesis simply because "experimental manipulation *might have* changed some other uncontrolled attribute besides kidney function that is a common cause of the colour of blood in the renal vein before entering, and after leaving the kidney". We must advance *some* evidence to support the idea that such an uncontrolled factor in fact exists. For instance, a critic might reasonably point out that some other attribute is also known to be correlated with blood colour and that the experimental manipulation was known to have changed this attribute. Although such evidence would certainly not be sufficient to demonstrate that this other attribute definitely was the cause, it might be enough to cast doubt on the veracity of the principal hypothesis. This is the same criterion as we used before to choose a significance level in our statistical test. Rejecting a statistical hypothesis because the probability associated with it was, say, 0.5 would not be reasonable. Certainly, this gives some doubt about the truth of the hypothesis, but our doubt is not sufficiently strong that we would have a clear preference for the contrary hypothesis.

It is the same defeasible argument that might be raised in a murder trial. If the prosecution has demonstrated that the accused had a strong motive, if it produced several reliable eyewitnesses and if it produced physical evidence implicating the accused then it would not be enough for the defence to simply claim that "maybe someone else did it". If, however, the defence could produce some contrary empirical evidence implicating someone else then reasonable doubt would be cast on the prosecution's argument. In fact, I think that the analogy between testing a scientific hypothesis and testing the innocence of the accused in a criminal trial can be stretched even further. There is no objective definition of reasonable doubt in a criminal trial; what is reasonable is decided by the jury in the context of legal precedence. In the same way, there is no objective definition of reasonable doubt in a scientific claim. In the first instance, reasonable doubt is decided by the peer reviewers of the scientific article and, ultimately, reasonable doubt is decided the entire scientific community. One should not conclude from this that such decisions are purely subjective acts and that scientific claims are therefore simply relativistic stories whose truth is decided by fiat by a power elite. Judgements concerning reasonable doubt and statistical significance are constrained in that they must deliver predictive agreement with the natural world in the long run.

Now let's look at the process of inference with respect to causal graphs.

- If the data were generated according to the causal model;
- If the causal process generating the data does not include non-linear feedback relationships;
- If the statistical test used to test the independence relationships is appropriate for the data;
- If a rare sampling fluctuation did not occur;
- Then each d-separation statement will be mirrored by a probabilistic independence in the data;
- At least one predicted probabilistic independence did not exist.
- Therefore, the causal model is wrong.

By now, you should have recognised the similarity of these inferences. We can prove by logical deduction that d-separation implies probabilistic independence in such directed acyclic graphs.

We can prove that, barring the case of non-linear feedback with non-normal data (an auxiliary assumption), every d-separation statement obtained from any directed graph must be mirrored by a probabilistic independence in any data that were generated according to the causal process that was coded by this directed graph. We can prove that, barring a non-faithful probability distribution (another auxiliary assumption, but one that is only relevant is the causal hypothesis is accepted, not if it is rejected), there can be no independence relation in the data that is not mirrored by d-separation. So, if we have used a statistical test that is appropriate for our data and have obtained a probability that is sufficiently low to reasonably exclude a rare sampling event, then we must tentatively reject our causal model. As in the case of the controlled experiment, if we are led to tentatively accept our causal model, then this will require that we can't reasonably propose an alternative causal explanation that also fits our data as well. As always, it is not sufficient to simply claim that "*maybe* there is such an alternative causal explanation". One must be able to propose an alternative causal explanation that has at least enough empirical support to cast reasonable doubt on the proposed explanation.

In this book, I will describe two different ways of constructing a statistical test that can potentially reject our causal hypothesis. The first method, which I call a "dsep " test, leads to piecewise structural equation models. A dsep test makes explicit use of d-separation. The second method, called "covariance-based structural equation models" leads to the classical version of structural equation models. However, covariance-bases SEM can also be derived from d-separation (Pearl 2009) even though d-separation is not explicitly invoked[47].

[47] In fact, d-separation was not even discovered when covariance-based SEM was developed.

3
Sewall Wright, path analysis and d-separation

3.1 A bit of history

"The ideal method of science is the study of the direct influence of one condition on another in experiments in which all other possible causes of variation are eliminated. Unfortunately, causes of variation often seem to be beyond control. In the biological sciences, especially, one often has to deal with a group of characteristics or conditions which are correlated because of a complex of interacting, uncontrollable, and often obscure causes. The degree of correlation between two variables can be calculated with well-known methods, but when it is found it gives merely the resultant of all connecting paths of influence.

The present paper is an attempt to present a method of measuring the direct influence along each separate path in such a system and thus of finding the degree to which variation of a given effect is determined by each particular cause. The method depends on the combination of knowledge of the degrees of correlation among the variables in a system with such knowledge as may be possessed of the causal relations. In cases in which the causal relations are uncertain the method can be used to find the logical consequences of any particular hypothesis in regard to them."

So begins Sewall Wright's 1921 paper (Wright 1921) in which he describes his "method of path coefficients". In fact, he invented this method while still in graduate school (Provine 1986) and had even used it without presenting its formal description in a paper published the previous year (Wright 1921). The 1920 paper used his new method to describe and measure the direct and indirect causal relationships that he had proposed to explain the patterns of inheritance of different colour patterns in Guinea Pigs. The paper came complete with a path diagram (i.e. a

causal graph) in which actual drawings of the colour patterns of Guinea Pig coats were used instead of variable names.

Wright was one of the most influential evolutionary biologists of the twentieth century, being one of the founders of population genetics and intimately involved in the modern synthesis of evolutionary theory. Despite these other impressive accomplishments, Wright viewed path analysis as one of his more important scientific contributions and continued to publish on the subject right up to his death (Wright 1984). The method was described by his biographer (Provine 1986) as "…the quantitative backbone of his work in evolutionary theory". His method of path coefficients is the intellectual predecessor of all the methods described in this book. It is therefore especially ironic that path analysis – the "backbone" of his work in evolutionary theory – was completely ignored by biologists[48].

This chapter has three goals. First, I want to explore why, despite such an illustrious family pedigree, path analysis and causal modelling was subsequently largely ignored by biologists. To do this I will have to delve into the history of biometry at the turn of the twentieth century, but it is important to understand why path analysis was ignored to appreciate why its modern incarnation does not deserve such a fate. Next, I want to introduce an inferential test (a "dsep " test) that allows one to actually test the causal claims of the path model. Sewall Wright's paper described how one can decompose the overall correlation between pairs of variables into the effects along different paths after assuming a causal structure, i.e., the DAG, but there was no way of testing this causal structure and this (I suspect) was one reason why it was largely ignored. The inferential test described in this chapter is not the first such test. Another inferential test was developed quite independently by statisticians specializing in the social sciences in the early 1970's, based on a statistical technique called maximum likelihood estimation. Since that method forms the basis of modern covariance-based structural equation modelling, I will postpone its explanation until the next chapter. Finally, I will present some published biological examples of path analysis and apply the new inferential test to them.

[48] This was written in the first edition. Path analysis and structural equation modelling are now more common in papers published in ecology and evolution.

3.2 Why Wright's method of path analysis was ignored

I suspect that scientists largely ignored Wright's work on path analysis for two reasons. First, it ran counter to the philosophical and methodological underpinnings of the two main contending schools of statistics at the turn of the 20th century. Second, it was methodologically incomplete in comparison to R. A. Fisher's statistical methods (Fisher 1925), based on the analysis of variance combined with the randomised experiment, which had appeared at about the same time.

Francis Galton invented the method of correlation. Karl Pearson transformed correlation from a formula into a concept of great scientific importance and championed it as a replacement for the "primitive" notion of causality. Despite Pearson's long-term program to provide "mathematical contributions to the theory of evolution" (Aldrich 1995), he had little training in biology, especially in its experimental form. He was educated as a mathematician and became interested in the philosophy of science early in his career (Norton 1975). Presumably, his interest in heredity and genetics came from his interest in Galton's work on regression which was itself applied to heredity and eugenics[49]. In 1892 Pearson published a book entitled *The Grammar of Science* (Pearson 1892). In his chapter entitled "Cause and Effect" he gave the following definition: "Whenever a sequence of perceptions D, E, F, G is invariably preceded by the perception C…, C is said to be the *cause* of D, E, F, G." As will become apparent later, his use of the word "perceptions" rather than "events" or "variables" or "observations" was an important part of his phenomenalist philosophy of science. He viewed the relatively new concept of correlation as having immense importance to science and the old notion of causality as so much metaphysical nonsense. In the third edition of his book (Pearson 1911) he even included a section entitled "The Category of Association, as replacing Causation". On page 166 of this third edition, he had this to say: "The newer and I think truer, view of the universe is that all existences are associated with a corresponding variation among the existences in a second class.

[49] Galton published his *Hereditary Genius* in 1869 (Galton 1869) in which he studied the "natural ability" of men (women were presumably not worth discussing). He was interested in "…those qualities of intellect and disposition, which urge and qualify a man to perform acts that lead to reputation…". He concluded that "…[those] men who achieve eminence, and those who are naturally capable, are, to a large extent, identical". Lest we judge Galton and Pearson too harshly, remember that such views were considered almost self-evident to White Europeans (and North Americans) at the time.

Science has to measure the degree of stringency, or looseness of these concomitant variations. Absolute independence is the conceptual limit at one end to the looseness of the link, absolute dependence is the conceptual limit at the other end to the stringency of the link. The old view of cause and effect tried to subsume the universe under these two conceptual limits to experience – and it could only fail; things are not in our experience either independent or causative. All classes of phenomena are linked together, and the problem in each case is how close is the degree of association."

These words may seem curious to many readers because they express ideas that have mostly disappeared from modern biology. Nonetheless, these ideas dominated the philosophy of science at the beginning of the twentieth century and were at least partially accepted by such eminent scientists as Albert Einstein. Pearson was a convinced phenomenalist and logical positivist[50]. This view of science was expressed by people like the physicist and mathematician Gustav Kirchhoff who held that science can only discover new connections between phenomena, not discover the "underlying reasons". The physicist and philosopher of science Ernst Mach, who dedicated one of his books to Pearson, viewed the only proper goal of science as providing economical descriptions of experience by describing the relationships between diverse experiences in the form of mathematical formulae. To go beyond this and invoke unobserved entities like "atoms" or "causes" or "genes" was not science, and such terms must be removed from its vocabulary. So, Pearson held that a mature science would express its conclusions in the form of functional – i.e. mathematical – relationships that can summarise and predict direct experience, not as causal links that can explain phenomena (Passmore 1966).

Pearson concluded, in accord with British empiricist tradition and the people cited above, that association was all that there was. Causality was an outdated and useless concept. The proper goal of science was simply to measure direct experiences (phenomena) and to economically describe them in the form of mathematical functions. If a scientist could predict the likely values of variable Y after observing the values of variable X, then he would have done his job. The more simply and accurately he could do it, the better his science. Going back to Chapter 2, Pearson did not view the equivalence operator of algebra ("=") as an imperfect *translation* of a causal relationship because he did not recognise "causality" as anything but correlation in the

[50]It is more accurate to say that his ideas were a forerunner to logical positivism.

limit[51]. By the time that Wright published his method of path analysis, Pearson's British school of biometry was dominant. One of its fundamental tenets was that "it is this conception of correlation between two occurrences embracing all relationships from absolute independence to complete dependence, which is the wider category by which we have to replace the old idea of causation" (Pearson 1911 p. 157).

Given these strong philosophical views, imagine what happened when Wright proposed using the biometrists' tools of correlation and regression … to peek beneath direct observation and deduce systems of causation from systems of correlation! In such an intellectual atmosphere, Wright's paper on path analysis was seen as a direct challenge to the Biometrists. One has only to read the title ("Correlation and Causation") and the introduction of Wright's (1921) paper, cited at the beginning of this chapter, to see how infuriating it must have seemed to the Pearson school.

The pagan had entered the temple, and, like the Macabees, someone had to purify it. The reply came the very next year (Niles 1922). Said Niles: "We therefore conclude that philosophically the basis of the method of path coefficients is faulty, while practically the results of applying it were[52] it can be checked prove to be wholly unreliable". Although he found fault in some of Wright's formulae (which were, in fact, correct) the bulk of Niles' scathing criticism was openly philosophical: "Causation' has been popularly used to express the condition of association, when applied to natural phenomena. There is no philosophical basis for giving it a wider meaning than partial or absolute association. In no case has it been proved that there is an inherent necessity in the laws of nature. Causation is correlation…" (Niles 1922).

Any Mendelian geneticist during that time – of whom Wright was one – would have accepted as self-evident that a mere correlation between parent and offspring told nothing about the mechanisms of inheritance. Therefore, concluded these biologists, a series of correlations between traits of an organism told nothing of how these traits interacted biologically or evolutionarily[53]. The Biometricians could never have disentangled the genetic rules determining colour inheritance in Guinea Pigs, which Wright was working on at the time, simply by using

[51] And yet, citing Hume, Pearson did accept that associations could be time-ordered from past to future. Nowhere in his writings did he express unease that such asymmetries could not be expressed by the equivalence operator.
[52] Grammatical error in the original citation.
[53] Karl Pearson was strongly opposed to Mendelism and, according to Norton (1975), this opposition was based on his philosophy of science; Mendelians insisted on using unobserved entities ("genes") and forces ("causation").

correlations or regressions. Even if distinguishing causation from correlation appeared philosophically "faulty" to the Biometricians, Wright and the other Mendelian geneticists were experimentalists for whom statements such as "causation is correlation" would have seemed equally absurd. For Wright, his method of path analysis was not a statistical *test* based on standard formulae such as correlation or regression. Rather, his path coefficients were interpretative parameters for measuring direct and indirect causal effects based on a causal system that had already been determined. His method was a statistical translation, a mathematical analogue, of a biological system obeying asymmetric causal relationships.

As the fates would have it, path analysis soon found itself embroiled in a second heresy. Three years after Wright's *Correlation and Causation* paper, Fisher published his *Statistical Methods for Research Workers* (Fisher 1925). Fisher certainly viewed correlation as distinct from causation. For him the distinction was so profound that he developed an entire theory of experimental design to separate the two. He viewed randomisation and experimental control as the only reliable way of obtaining causal knowledge. Later in his life Fisher even wrote an entire book criticising the research that identified tobacco smoking as a cause of cancer on the basis that such evidence was not based on randomised trials[54] (Fisher 1959). I have already described the assumptions linking causality and probability distributions, unstated by Fisher but needed to infer causation from a randomised experiment, as well as the limitations of these assumptions, when studying different attributes of organisms. Despite these limitations, Fisher's methods had one important advantage over Wright's path analysis: they allowed one to rigorously test causal hypotheses while path analysis could only estimate the direct and indirect causal effects *assuming* that the causal relationships were correct.

Mulaik (1986) has described these two dominant schools of statistics in the twentieth century. His phenomenalist and empiricist school starts with Pearson. Examples of the statistical methods of this school were correlation, regression[55], common-factor and principal component analyses. The purpose of these methods was primarily, as Ernst Mach directed, to provide a description of experience by economically describing a large number of diverse experiences in the form of

[54] Fisher was a smoker. I wonder what he would have thought if, because of a random number, he was assigned to the "non-smoker" group in a clinical trial?

[55] Regression based on least squares was, of course, developed well before Pearson by people like Gauss and had been based on a more explicit causal assumption that the independent variable plus independent measurement errors were the causes of the dependent variable. This distinction lives on under the guise of Type I and Type II regression.

mathematical formulae. The second school was the Realist school begun by Fisher. This second school emphasised the analysis of variance, experimental design based on the randomised experiment and the hypothetico-deductive method. These Fisherian methods were not designed to provide functional relationships, but rather to ensure conditions under which causal relationships could be reliably distinguished from non-causal relationships.

In hindsight then, it seems that path analysis simply appeared at the wrong time. It did not fit into either of the two dominant schools of statistics and it contained elements that were objectionable to each. The phenomenalist school of Pearson disliked Wright's notion that one *should* distinguish "causes" from correlations. The Realist school of Fisher disliked Wright's notion that one *could* study causes without randomised experiments and by only looking at correlations. Professional statisticians therefore ignored it. Biologists found Fisher's methods, complete with inferential tests of significance, more useful and conceptually easier to grasp and so biologists ignored path analysis too. A statistical method, viewed as central to the work of one of the most influential evolutionary biologists of the twentieth century, was largely ignored by biologists until relatively recently.

Wright's method of path analysis was so completely ignored by biologists that most biometry texts published in the twentieth century did not even mention it. Those that did (Li 1975, Sokal and Rohlf 1981) described it as Wright originally presented it without even mentioning that it was re-formulated by others, primarily economists and social scientists, such that it permitted inferential tests of the causal hypothesis and allowed one to include unmeasured (or "latent") variables. The main weakness of Wright's method – that it required one to assume the causal structure rather than being able to test it – had been corrected by 1970 (Jöreskog 1970) but biologists were mostly unaware of this until recently.

Two different ways of testing causal models will be presented in this book. The most common method, which I call covariance-based structural equations modelling, is based on comparing an empirical covariance matrix to one predicted by the causal graph. This method will be described in Chapters 4, 7 and 8 and it does have some advantages when testing models that include variables that cannot be directly observed and measured (so-called *latent* variables) and for which one must rely on observed indicator variables that contain measurement errors. Covariance-based SEM also has some statistical drawbacks. The inferential tests are asymptotic

and can therefore require rather large sample sizes. The functional relationships must be linear. Data that are not multivariate normal and that do not come from mutually independent observations are difficult (sometimes, impossible) to treat. These drawbacks led me to develop an alternative set of methods that can be used for small sample sizes, for non-normally distributed data, for non-linear functional relationships, and for data that have some nesting structure that renders the observations partially dependent (Shipley 2000). I call these "dsep " tests since these methods are derived directly from the notion of d-separation that was described in Chapter 2. More recently, these tests have also been called "piecewise" SEM (Lefcheck 2016).

3.3 Dsep tests

The link between (conditional) probabilistic independence and DAGs, given by d-separation, suggests an intuitive way of testing a causal model: simply list all the d-separation statements that are implied by the causal model and then test each of these using an appropriate test of conditional independence. There are a number of problems with this naïve approach. First, even models with a small number of variables can include a large number of d-separation statements[56], which makes evaluating all of them impractical or impossible in practice. Second, we need some way of combining all of these tests of independence into a single composite test. For instance, if we had a model that implied 100 independent d-separation statements and tested each independently at the traditional 5% significance level then we would expect, on average, that 5 of these tests to reach significance and be rejected simply due to random sampling fluctuations even though these rejected d-separation claims are true. Even worse, the d-separation statements in a causal model are almost never completely independent and so we would not even know what the true overall significance level would be. Each of these problems can be solved.

[56] The number of d-separation claims that exist in a DAG with V variables is $\frac{V!}{(V-2)!2!}2^{V-2}$

Given an acyclic causal graph, we can use d-separation to predict a set of conditional probabilistic independencies that must be true if the causal model is true. However, many of these d-separation statements can be predicted from other d-separation statements and are therefore not independent. Happily, Pearl (1988 section 3.3 Bayesian networks and Theorem 10, corollary 7) describes a simple method of obtaining the minimum number of d-separation statements needed to completely specify the causal graph and proves that this minimum list of d-separation statements is sufficient to predict the entire set of d-separation statements. This minimum set of d-separation statements is called a *basis set*[57]. The basis set is not unique. This method will be illustrated with Figure 3.1.

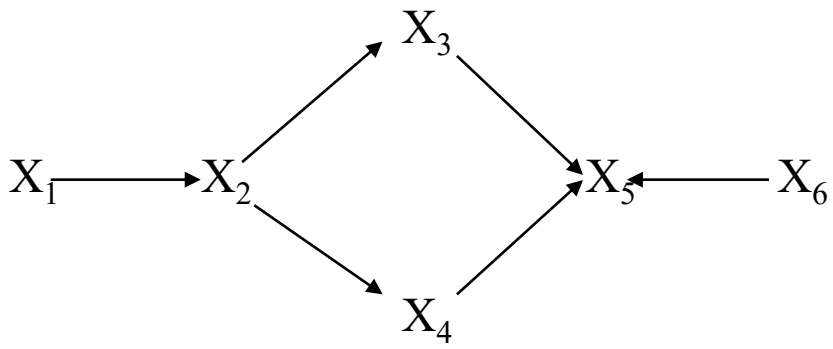


Figure 3.1. A DAG used to illustrate the notion of a basis set of d-separation claim.

To obtain the basis set, the first step is to list each unique pair of non-adjacent vertices. That is, list each pair of variables in the causal model that do not have an arrow between them. So, in Figure 3.1 the list is: {(X_1, X_3), (X_1, X_4), (X_1, X_5), (X_1, X_6), (X_2, X_5), (X_2, X_6), (X_3, X_4), (X_3, X_6), (X_4, X_6)}. Pearl's basis set is given by d-separation statements consisting of each such pair of vertices conditioned on the parents of the vertex having higher causal order. Pearl's basis set[58] is therefore: **B**={X_1_||_X_3|{X_2}, X_1_||_X_4|{X_2}, X_1_||_X_5|{X_3,X_4}, X_1_||_X_6|{ϕ}, X_2_||_X_5|{X_3,X_4}, X_2_||_X_6|{ϕ}, X_3_||_X_4|{X_2}, X_3_||_X_6{ϕ}, X_4_||_X_6|{ϕ}}. Remember that an exogenous variable has no parents (X_1 and X_6 in Figure 3.1), so the set of "parents" of such a variable is

[57] Let **S** be the set of d-separation facts (and therefore the set of conditional independence relationships) that are implied by a directed acyclic graph. A basis set **B** for **S** is a set of d-separation facts that (i) implies, using the axioms of conditional independence (Dawid 1979), all other elements of **S**, and (ii) no proper subset of **B** sustains such implications.

[58] Remember that a d-separation claim like X_1_||_X_3|{X_2} means "X_1 is d-separated from X_3 given (or conditional on) X_2".

empty (such an empty set is written "{ϕ}"). The number of pairs of variables that don't have an arrow between them is always equal to the total number of pairs minus the number of arrows in the causal graph. In general, if there are V variables and A arrows in the causal graph, then the number of elements in the basis set will be: $\frac{V!}{(V-2)!2!} - A$.

For each d-separation claim in the basis set, we must measure the degree of (conditional) independence of the two variables in the pair that we observed in our random sample. Unfortunately, the test statistics, based on such a basis set, that one uses to measure these conditional independencies in a sample, are not necessarily mutually independent (Shipley 2000). A basis set that does have this property[59] is given by the set of unique pairs of non-adjacent vertices, of which each pair is conditioned on the set of causal parents of both (Shipley 2000). I call this the "union" basis set. Therefore, the second step in getting the union basis set that will be used in the inferential test, described next, is to list the causal parents of each vertex in the pair. Using Figure 3.1 and the notation for d-separation introduced in Chapter 2, Table 3.1 summarises the d-separation statements that make up the union basis set.

Table 3.1. The union basis set for the DAG shown in Figure 3.1

Non-adjacent variables	Parent variables of either non-adjacent variable		d-separation claim
X_1, X_3	X_2		X_1_||_X_3|{X_2}
X_1, X_4	X_2		X_1_||_X_4|{X_2}
X_1, X_5	X_6		X_1_||_X_5|{X_6}
X_1, X_6	None		X_1_||_X_6|{ϕ}
X_2, X_5	X_1, X_3, X_4		X_2_||_X_5|{X_1, X_3, X_4}
X_2, X_6	X_1		X_2_||_X_6|{X_1}
X_3, X_4	X_2		X_3_||_X_4|{X_2}
X_3, X_6	X_2		X_3_||_X_6|{X_2}
X_4, X_6	X_2		X_4_||_X_6|{X_2}

[59] This is true for DAGs, but it is not true for mixed acyclic graphs (MAGs) that will be introduced in Chapter 6.

Each of the d-separation statements in Table 3.1 predicts a (conditional) probabilistic independence. How you test each predicted conditional independence depends on the nature of the variables and the nature of the functional relationship linking the d-separated pair. That means that different d-separation statements in your union basis set could be tested with different statistical tests of (conditional) independence. For the moment, assume that you have used tests of independence that are appropriate for the variables involved in each d-separation statement and that you have obtained the exact probability level assuming such independence. By "exact" probability levels, I mean that you cannot simply look at a statistical table and find that the probability is ≤ 0.05; rather, you must obtain the actual probability level – say, p=0.036. Such probabilities refer to each separate hypothesis of independence generated by each separate d-separation claim.

At this point, since we have a union basis set containing k d-separation claims, we have a list of k null probabilities (one for each d-separation claim) based on k separate null hypotheses of (conditional) independence. However, the null hypothesis that we want to test is that all of the d-separation claims are true, i.e. we need a single null probability based on this global null hypothesis. Because the conditional independence tests implied by the union basis set are mutually independent, we can obtain this global composite null probability for the entire set using a test statistic derived by Fisher (1932). Since this test does not seem to have a name, I have called it Fisher's C (for "combined") statistic. If there are a total of k independence tests in the union basis set, and p_i is the exact probability of the i^{th} test assuming independence, then the test statistic is: $C = -2\sum_{i=1}^{k} \ln(p_i)$. If all k independence claims are true, then this statistic will follow a chi-squared distribution with 2k degrees of freedom. This is not an asymptotic test unless you use asymptotic tests for some of the individual independence hypotheses. Furthermore, you can use different statistical tests for different individual independence hypotheses. In this sense, it is a very general test. In fact, it is probably better to describe it as a recipe for constructing your own dsep test.

3.4 Independence of d-separation claims via regression slopes

Statisticians have devised many different inferential tests of independence and conditional independence. These different tests require different assumptions about the nature of the variables in question; for instance, the distributions of these variables and the functional forms linking them. Later in this chapter, I will describe some more specialized tests of independence but will begin with two types of tests that are appropriate for many of the DAGs that biologists are likely to encounter.

The first type of test of independence is based on a zero slope in a regression (Shipley 2009). This is the test of independence used by the piecewiseSEM library (Lefcheck 2016), which we will look at later. Remember that, given $X_i \perp\!\!\!\perp X_j | \{\mathbf{Q}\}$, where **Q** is the set of causal parents of both X_i and X_j, our goal is to test the null hypothesis that X_i and X_j are independent, conditional on the variables in **Q**. If we regress X_j on the variables in **Q** (say, X_1, X_2,…,X_n) plus X_i, then the slope associated with X_i will be zero in the statistical population if X_j is independent of X_i given **Q**. Using the notation in R, we want to perform a regression of $X_j \sim X_1 + X_2 + \ldots + X_n + X_i$. Since the sole purpose of performing this regression is to obtain the null probability associated with the slope for X_i, we are not interested in anything except the sample estimate of this slope and the probability that this slope estimate is different from zero. Why? Because if the slope of X_i in this regression is zero in the statistical population, then this also means that X_j is independent of X_i, given **Q**.

What type of regression should we use? That depends on the distributional assumptions that are appropriate for X_j and on the functional form linking X_j to X_i. If X_j is normally distributed with mutually independent observations and is linearly linked to its predictors, including X_i, then you would perform a linear regression. If there is some nesting structure in the data that renders the observations partly dependent, then you would perform a mixed model regression (Pinheiro and Bates 2000); this type of analysis is also called multilevel or hierarchical regression. If X_j follows a non-normal distribution from a general exponential family, for example, a Poisson or binomial distribution, then you would perform a generalized linear model (McCullagh and Nelder 1989). More complicated regressions are also possible, like a generalized linear mixed

models or generalized additive models (Wood 2017) that allow for quite complicated nonlinear relationships. Many different R packages exist to perform these different types of regressions (examples can be found later in this chapter) but I won't say any more about these different types of regression here. The citations given above should direct you to the appropriate literature if you need more information. However, you do need to understand the basics of these methods if you plan on using them because the statistical assumptions of these models must be correct in order for the null probability associated with the slope of X_i, and therefore the test of independence of X_i and X_j, to be correct.

I have been talking about X_i and X_j, the two variables that are predicted to be d-separated, but which variable should be regressed on the other? After all, conditional independence is a symmetrical relationship, but regression requires that we choose which variable to be the dependent variable and which variables to be the independent variables. For most d-separation claims, the choice of which variable to use as the dependent variable in the regression is specified by the DAG. Given X_i_||_$X_j|\mathbf{Q}$, and if X_j is caused by X_i (i.e. X_j is a descendent of X_i), then X_j is regressed on X_i plus the variables in **Q**. In general, the causal descendent is the dependent variable and the causal ancestor is the independent variable. However, in cases in which neither X_i nor X_j is the causal ancestor, there is some ambiguity. For instance, in the union basis set given in Table 3.1, there is the d-separation claim X_3_||_$X_4|\{X_2\}$. Notice that neither X_3 nor X_4 is the causal ancestor of the other so which variable should be the dependent variable? If both X_i and X_j are normally distributed variables, then it makes no difference which is regressed on the other because the null probability of a zero slope will be the same. However, if either X_i or X_j are not normally distributed, meaning that you would have to perform a generalized linear regression, then the null probability of a zero slope will not be identical depending on which variable is regressed on the other. This occurs because non-normally distributed variables in a generalized linear model require a "link" function, i.e., a transformation of the dependent variable, that linearizes the relationship. For example, consider what happens if X_j is a count variable that follows a Poisson distribution while X_i is normally distributed. A generalized linear regression of X_j on X_i will require a link function of $\ln(X_j)$. That means that the null hypothesis associated with the slope of X_i will be "H_0: $\ln(X_j)$ is independent of X_i". If, on the other hand, we regress X_i on X_j, and since X_i is normally distributed, then the link function is the "identity" link (i.e., $1X_i$). The null hypothesis associated with the slope of X_i will

be "H_0: $1X_j$ is independent of X_i". These are two subtly different hypotheses and so their null probabilities will not be identical. One solution that the piecewiseSEM package proposes is to do both regressions and then use the smaller null probability. I have conducted Monte Carlo simulations of many different DAGs whose variables follow non-normal exponential distributions and have found that this solution does give better rejection rates, but a simulation result is not as reassuring as a mathematical proof and there is a better solution, as will be described later.

Later, I will explain how to use the `psem()` function of the piecewiseSEM package (Lefcheck 2016) to conduct a dsep test via regression slopes but, since not every data set is appropriate for the `psem()` function, it is important to understand how to conduct a dsep test yourself in R. It is also important to know how to do this because, if you need to verify the statistical assumptions of the regression models (and you should), then you will have to understand how this method works. For an initial example of the dsep test, I will use a data set consists of 100 mutually independent observations generated from the DAG in Figure 3.1. Each variable is normally distributed and linearly related.

The first step is to obtain the union basis set. You can easily do this by yourself but you can also do this via the `DAG()` and `basiSet()` functions of the ggm library (Marchetti et al. 2024). These functions are useful for more complicated DAGs. The `DAG()` function stores your DAG as a binary matrix. The notation "X2~X1" means that X_2 is cause by (i.e. "~") X_1. You enter each link in your DAG this way, with all of the direct causes of each endogenous variable[60] entered on the left-hand side of the tilde (~). The `basiSet()` function extracts the union basis set from the DAG.

```
library(ggm)
Fig3_1_DAG<-DAG(X2~X1,X3~X2,X4~X2,X5~X3+X4+X6)
Fig3_1BS<-basiSet(Fig3_1_DAG)
```

The union basis set is now stored in `Fig3_1BS`, which is a list containing the 9 d-separation claims given in Table 3.1. Each list element in `Fig3_1BS` contains the two variables that are d-separated, followed by the causal ancestors of both (which are the conditioning

[60] If your DAG has a variable that neither causes, or is caused by, any other variable, then you would enter it as X_i~1.

variables, if there are any). The first list element tells us that X_6 is d-separated from X_1 without any conditioning variables (because there are no other variables following X_6 and X_1. The second list element tells us that X_6 is d-separated from X_2, conditional on X_1. The other seven list elements give us the other seven d-separation claims from our DAG. Here are the first two list elements:

```
[[1]]
[1] "X6" "X1"

[[2]]
[1] "X6" "X2" "X1"
```

The second step is to obtain the null probability of independence of the first two variables listed in each d-separation claim, conditional of the remaining variables in each d-separation claim. In a complete analysis, you would have to verify, or justify, the probability distribution of your variables but we know that our variables are normally distributed. We can therefore obtain the null probability of our hypotheses of conditional independence by conducting a series of linear regressions using the `lm()` function of R. Here are two general recommendations for using regressions to test for conditional independence :

1. Place the causal descendent of the pair as the dependent variable in the regression. If neither variable of the pair is a causal descendent of the other, and at least one of them has a non-normal distribution[61] (requiring a generalized linear model) then conduct two regressions, one with each of the two variables in the pair as a dependent variable and keep the smaller of the two probabilities of a zero slope for the other.
2. Place the conditioning variables first as predictor variables and the second variable of the d-separated pair as the last predictor variable. If all the predictor variables are continuous then this order doesn't make a difference, but if any of the predictor variables are categorical variables (R calls these "factor" variables) then the order does make a difference, and you want to control the conditioning variables before the variable in the d-separation claim.

Let's get the probability of a zero slope for the first d-separation claim: namely X_6_||_$X_1|\{\phi\}$.

```
summary(lm(X6~X1,data=Fig3_1_data)).
```

[61] If both variables in the d-separation pair are normally distributed, then you don't need to do both regressions for the reason explained earlier in the text.

The relevant output is:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.03769    0.10042  -0.375    0.708
X1           0.11590    0.10796   1.074    0.286
```

The estimated slope of X_1 in this data is 0.11590 with a standard error of this estimate being 0.10796. The Student's t-statistic, given the null hypothesis of a slope of zero in the statistical population[62], is 1.074. The probability of observing a slope of at least |0.11590| given this null hypothesis, is 0.286. Since a slope of zero implies independence between the dependent variable and this independent variable conditional on all of the other independent variables in the regression, this also the probability that X_1 and X_6 are unconditionally independent, which is the probability that we want; thus, p_1=0.286. Since X_1 and X_6 are both exogenous variables and neither is the ancestor of the other, if either X_1 or X_6 was non-normally distributed, then we would redo the regression after making X_1 the dependent variable, but this is not necessary since both are normally distributed.

The probability of a zero slope for the second d-separation claim in the union basis set, X_6_||_X_2|{X_1} is obtained from `summary(lm(X6~X1+X2))`. Note that I have placed the conditioning variable (X_1) before X_2 in the regression. Again, since neither X_6 nor X_1 are causal descendants of the other, we would have to also perform a regression with X_2 as the dependent variable if either was non-normally distributed. The relevant output is:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.03801    0.09986  -0.381    0.704
X1           0.02028    0.12593   0.161    0.872
X2           0.19274    0.13267   1.453    0.149
```

The probability of observing at least as large a slope for X_2 as |0.19274|, given the null hypothesis of a zero slope in the statistical population, is p_2=0.149.

We proceed in the same manner for each of the remaining seven d-separation claims in `Fig3_1BS`, which is our union basis set. The result is a vector of nine null probabilities, one associated with each of the nine d-separation claims: `p<-c(0.286000, 0.149000,`

[62] $t = \frac{|0.1159 - 0|}{0.10796} = 1.074$

```
0.498000, 0.069700, 0.755413, 0.084000, 0.333000, 0.315000,
0.963000).
```

The final step is to combine these nine probabilities, one for each d-separation claim, into a single overall probability for the full DAG. We do this by calculating Fisher's C statistic and getting the probability of observing at least this large a value, assuming that all the d-separation claims are true (i.e., that the C statistic follows a chi-squared distribution[63] with 2k = 2(9) = 18 degrees of freedom):

```
C_stat<- -2*sum(log(p))
1-pchisq(C_stat,df=2*9)
```

The C statistic is 23.13242 and the null probability of observing at least this large a C statistic by chance is 0.186. Therefore, we conclude that we can't reject our null hypothesis that all our d-separation claims are true, i.e. that all the conditional independence claims predicted by the DAG are true. This conclusion is hardly surprising since I generated the data to follow the DAG in Figure 3.1 and to agree with the statistical assumptions of the generating equations (normality, linearity, mutually independent observations).

Since we have not rejected our hypothesized DAG, we can now go on to estimate the generating equations (the structural equations). If we had rejected our hypothesized DAG, then we should not proceed to this step since some of the estimated parameters (slopes, intercepts) will be biased. To estimate the structural equations, we simply regress each endogenous variable on its causal parents by following the arrows in our DAG. Notice that we are no longer doing regressions to test the d-separation claims; rather, we are estimating the structural equations. Therefore, we don't use the d-separation claims in the basis set to structure our regressions but, instead, use the DAG itself to structure our regressions. In other words, given our DAG, we would (i) regress X_2 on X_1, (ii) regress X_3 on X_2, (iii) regress X_4 on X_2, and (iv) regress X_5 on X_6, X_3 and X_4. Here is the output from the first structural equation:

```
lm(formula = X2 ~ X1, data = Fig3_1_data)
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.001648   0.076033   0.022    0.983
```

[63] `pchisq(q=, df= )` is an R function that gives the probability of the chi-squared distribution for a chi-squared statistic of q with df degrees of freedom. Since we want the tail probability, i.e., the probability of observing a value of q or more, we need `1-pchisq()`.

```
X1          0.496103   0.081744   6.069 2.43e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 0.7597 on 98 degrees of freedom
```

Once we have completed our four regressions, here is the result:

$$X_2 = 0.002(\pm 0.076) + 0.496(\pm 0.082) X_1 + N(0, 0.760)$$
$$X_3 = -0.022(\pm 0.092) + 0.497(\pm 0.104) X_2 + N(0, 0.916)$$
$$X_4 = -0.184(\pm 0.084) + 0.440(\pm 0.095) X_2 + N(0, 0.834)$$
$$X_5 = 0.035(\pm 0.046) + 0.511(0.047) X_3 + 0.558(\pm 0.051) X_4 + 0.480(\pm 0.047) X_6 + N(0, 0.452)$$

Next, let’s analyse an empirical data set that has more complications. This data set[64] comes from the study on nestling success in Corsican Blue Tits whose causal hypothesis was described in section 2.2 of Chapter 2. This data set is called BlueTits.txt. Table 3.2 summarizes each of the variables. The hypothesized DAG describing the hypothesized causal links between them is given in Figure 2.3b. Since these data have a two or three level nesting structure, we must use mixed model regression and so we must attach the lme4 package (Bates et al. 2015).

Table 3.2 Summary of the variables contained in the dataset BlueTit.txt

Variable	Description	Nesting structure	Distribution
protos	Number of Blowfly larvae per nest	nest/year	Poisson
frass	Average amount of caterpillar faeces produced (g m^{-2} d^{-1})	Nest/year	Normal
mass	The mass of a single nestling at 15 days (g)	Nest/year/individual	Normal
hemato	The percent of total blood volume occupied by red blood cells (haematocrit, %)	Nest/year/individual	Normal
recruited	1 if the nestling returned and bred in the population the following year; otherwise 0	Nest/year/individual	Binomial

[64] This is a slightly modified version of the original data se: (i) the variable “protos” (i.e., the number of Blowfly larvae per nest) was rounded to a whole number to make it a count variable. The original data set apparently measured the number of Blowfly larvae per chick but only reported the average number of larvae per nest per year; (ii) frass, i.e. the amount of caterpillar faeces, was converted from mg m^{-2} d^{-1} to g m^{-2} d^{-1}.

After attaching the ggm library[65], we enter the hypothesized DAG and obtain the union basis set:

```
BlueTits.DAG<-DAG(mass~protos+frass,hemato~protos+frass+mass,
                  Recruited~hemato)
BS<-basiSet(BlueTits.DAG)
```

The BS list object contains 4 d-separation claims:

```
[[1]]
[1] "frass"  "protos"

[[2]]
[1] "frass"    "recruited" "hemato"

[[3]]
[1] "protos"   "recruited" "hemato"

[[4]]
[1] "mass"     "recruited" "frass"    "protos"   "hemato"
```

The first list element in the union basis set claims that frass and protos will be unconditionally independent. Since neither variable is a causal descendent of the other, the DAG doesn't specify the dependent and independent variable. Since protos is a count variable, we must model it using a Poisson distribution. Because the DAG does not specify the dependent variable, and since one of the variables in non-normally distributed, we must regress each on the other and use the smaller of the two null probabilities Here are the two mixed model regressions[66]:

```
fit1.1<-
glmer(protos~frass+(1|year)+(1|nest),data=BlueTits,family=poisso
n(link="log"))
fit1.2<-lmer(frass~protos+(1|year)+(1|nest),data=BlueTits)
```

The summary of each output gives the null probability of a zero slope, which are 0.915 and 0.551 respectively[67]. We use the smaller of the two values: p_1=0.551.

The second d-separation claim is that recruited is independent of frass, conditional on hemato. The dependent variable is recruited since it is the causal descendent of frass. Since recruited is a binary (yes/no) variable, we must model it using a Binomial distribution. The summary of `fit2`

[65] We will also need to attach the lmerTest package (Kuznetsova et al. 2017) as well to get the null probabilities.
[66] Such mixed regressions might return an error message if convergence doesn't occur. If so, you might have to rescale the variables; for instance, multiplying or dividing by some multiple of 10.
[67] Since these p values are only correct if the statistical assumptions of the regression are correct, you should verify these!

reports that the null probability associated with the hypothesis of a zero slope for frass is 0.659. Thus, p_2=0.659.

Here is the generalized mixed model regression:

```
fit2<-
glmer(recruited~hemato+frass+(1|year)+(1|nest),data=BlueTits,fam
ily=binomial(link="logit"))
```

The third d-separation claim in the union basis set is that recruited is independent of protos, conditional on hemato. The dependent variable is still recruited since it is the causal descendent of protos. Here is the generalized mixed model regression:

```
fit3<-
glmer(recruited~hemato+protos+(1|year)+(1|nest),data=BlueTits,fa
mily=binomial(link="logit"))
```

The summary of fit3 reports that the null probability associated with the hypothesis of a zero slope for protos is 0.217. Thus, p_3=0.217.

The fourth d-separation claim in the union basis set is that mass is independent of recruited given frass, protos, and hemato. The dependent variable is still recruited since it is the causal descendent of mass. Here is the generalized mixed model regression:

```
fit4<-
glmer(recruited~frass+protos+hemato+mass+(1|year)+(1|nest),data=
BlueTits,family=binomial(link="logit"))
```

The summary of fit4 reports that the null probability associated with the hypothesis of a zero slope for mass is 0.538. Thus p_4=0.538. We now have our null probabilities for each of the d-separation claims in the union basis set for our DAG and so we calculate our C statistic, which is 6.322:

```
C.statistic<--2*sum(log(p))
```

The null probability of this C statistic given 2(4)=8 degrees of freedom is 0.611:

```
1-pchisq(C.statistic,8).
```

Since the null probability of this C statistic is not lower than our significance level (0.05), we have no good evidence to reject our DAG. We can therefore proceed to estimate the structural

equations by fitting the appropriate models to each endogenous variable in the DAG and extracting the slopes, intercepts, and their standard errors for each regression:

```
summary(lmer(mass~protos+frass+(1|year)+(1|nest),data=BlueTits))
summary(lmer(hemato~protos+frass+mass+(1|year)+(1|nest),data=Blu
eTits))
summary(glmer(recruited~hemato+(1|year)+(1|nest),data=BlueTits,f
amily=binomial(link="logit")))
```

Here are the structural equations:

$$mass = 9.42(\pm 0.10) - 0.04(\pm 0.003)\, protos + 3.71(\pm 0.57)\, frass$$

$$hematos = 28.60(\pm 2.59) - 0.52(\pm 0.03)\, protos - 1.07(\pm 5.23)\, frass + 1.91(\pm 0.26)\, mass$$

$$\ln\left(\frac{p(recruited)}{1-p(recruited)}\right) = -4.65(\pm 0.78) + 0.04(\pm 0.02)\, hemato$$

Note that the last structural equation uses the logit transformation since this is the default link function for a binomial distribution. The numbers are the estimated values of the regression parameters (slopes, intercepts). In SEM, the slopes are called "path coefficients". The values inside the parentheses are the standard errors of these estimates. Later in this chapter, I will explain how to interpret path coefficients, and how to combine them to give estimates of indirect effects along different directed paths.

Since these structural equations are fit using the `lmer()` and `glmer()` functions via maximum likelihood methods, the sample estimates ($\hat{x}_i$) and standard errors (s_i) of the slopes and intercepts can be compared to any theoretical value (μ_i) using z-statistics: $z_i = \frac{|\hat{x}_i - \mu_i|}{s_i}$. The 95% confidence intervals around these sample estimates are: $\hat{x}_i \pm 1.96 s_i$. Note that the sample estimate is of the transformed value, given the link function, if the variable in question is not normally distributed; for instance, the third structural equation above is a function of the logit-transformed[68] probability of an individual bird being recruited.

[68] $\ln\left(\frac{p(recruited)}{1-p(recruited)}\right)$

It might happen that a path coefficient in a structural equation is not significantly different from zero; equivalently, that the 95% confidence interval for this path coefficient includes zero. What should you do? After all, your causal hypothesis claimed that the independent variable was a direct cause of the dependent variable in your structural equation. Remember that saying that a sample estimate of any parameter is not significantly different from zero is not the same thing as saying that it is equal to zero. There are always two alternative possibilities in such cases: (i) the true value of the parameter is zero (in which case your causal hypothesis has been contradicted) or (ii) the true value of the parameter is not zero but is sufficiently close to zero as to be indistinguishable from zero given the measured level of sampling variation (i.e. the standard error of the parameter estimate). Another way of stating the second possibility is to say that the statistical test lacks sufficient statistical power to exclude the first possibility; the notion of statistical power will be discussed in Chapter 5. There is no statistical method of distinguishing between these two possibilities without increasing your sample size. If you have good non-statistical reasons to believe that the value of the parameter value should be outside of the confidence interval, then you would conclude that your causal hypothesis has been contradicted. If you don't have good non-statistical reasons to believe that the value of the parameter value should be outside of the confidence interval, then you would simply conclude that if your causal hypothesis is correct then the strength of this causal effect is small enough to be within the confidence interval. For instance, the estimated causal effect of a one unit increase in the haematocrit on the log-odds of recruitment in the third structural equation was 0.04 and its 95% confidence interval[69] was between 0.0008 and 0.0792. Although this value is significant[70] at the 5% level (p=0.046), it is a small effect.

3.5 Independence of d-separation claims via regression slopes using the piecewiseSEM package

[69] $0.04 \pm 1.96(0.02)$

[70] $z = \frac{|0.04 - 0|}{0.02} = 2$

Despite the fact that the DAG in Figure 2.3b, for the Blue Tits study, involved only five variables, the above analysis was rather long and involved many separate steps that could lead to errors. The piecewiseSEM package (Lefcheck 2016) automates these steps and allows us to do the entire analysis in just two steps. However, the quality of the analysis output by this package is entirely dependent on the quality of the various regression models used to test the d-separation claims and to estimate the structural equations. It is still your responsibility to verify the appropriateness of the assumptions of these regressions and the piecewiseSEM package does not do this for you. You should go over the underlying regression models and check the model assumptions, including those used to test the d-separation claims, before publishing your results.

To use the piecewiseSEM package, you need only two calls: one to the `psem()` function and one to the `summary()` function. However, whenever your DAG has more than one exogenous variable, you will also need to do a bit more work if you want to include the d-separation claims involving pairs of such exogenous variables. This is because of an unfortunate (in my opinion) choice of defaults involving such pairs of exogenous variables.

The first call is to `psem()`. The main argument of this function is the list of structural equations as specified in your DAG. Each structural equation is a regression model from one of the following model classes: `lm, glm, gls, pgls, Sarlm, lme, glmmPQL, lmerMod, lmerModLmerTest, glmerMod, glmmTMB, gam`. For instance, here is how to use the `psem()` function when analysing the Blue Tits data that we have already studied:

```
fit.Blue.Tits<-psem(
lmer(hemato~protos+frass+mass+ (1|year)+(1|nest),data=BlueTits),
lmer(mass~protos+frass+ (1|year)+(1|nest),data=BlueTits),
glmer(recruited~hemato+(1|year)+(1|nest),family = binomial(link
= "logit"), data=BlueTits)).
```

The second call is to the `summary()` function: `summary( fit.Blue.Tits, conserve=TRUE, conditioning=TRUE)`. You should always include the second argument even though it is not the default. Including `conserve=TRUE` specifies that whenever there is not a natural ordering of dependent and independent variables in the regressions used to test the d-separation claims, i.e., whenever one of the pair of d-separated variables is not a causal descendent of the other, one must do both regressions and keep the smaller null probability of the

pair. The third argument (`conditioning=TRUE`), which isn't required but makes our life easier, specifies that we want to print out all of the conditioning variables for each d-separation claim. We will look at the full output of this summary call in a few paragraphs, but here is the part of the output listing the d-separation claims in the union basis set:

```
Tests of directed separation:

                             Independ.Claim Test.Type   DF Crit.Value P.Value
                recruited ~ protos + hemato      coef 1309    -1.2352  0.2168
                 recruited ~ frass + hemato      coef 1309     0.4409  0.6593
 recruited ~ mass + protos + frass + hemato      coef 1309     0.6154  0.5383
```

You will notice that we had four d-separation claims in the union basis set when we did the analysis step-by-step, but only three of these claims are listed in this output from `psem()`. In particular, the d-separation claim involving the two exogenous variables (protos and frass) is missing. Furthermore, the output describing Fisher's C statistic is as follows: `Fisher's C = 5.13 with P-value = 0.527 and on 6 degrees of freedom`. The six degrees of freedom correspond to the three listed d-separation claims, meaning that the C statistic that `psem()` calculates does not include protos_||_frass|{ϕ}. This differs from our analysis above. This difference requires some explanation and a solution.

An exogenous variable has no explicit causes in the DAG. Therefore, there cannot be any directed paths from one exogenous variable to the other (because this would mean that one of them is not exogenous) nor can there be any directed paths from some other variable to each of these exogenous variables (because this would mean that neither of them is exogenous). Therefore, pairs of such exogenous variables must always be unconditionally d-separated. This is why our DAG for the Blue Tits study (Figure 2.3b) has two exogenous variables (frass and protos) and the union basis set for this DAG includes the claim that frass and protos are d-separated given no other variable (i.e., protos_||_frass|{ϕ}). However, by default, `psem()` ignores all of the d-separation claims involving pairs of exogenous variables. Why? No explanation is given but (presumably) `psem()` implicitly assumes that all such pairs of exogenous variables are both caused by some common unknown variable[71] that is not included in the DAG. Given this assumption, such pairs of exogenous variables are not d-separated by any

[71] Strictly speaking, such a causal graph with an unknown common cause (a "latent" variable) is a mixed acyclic graph (MAG) not a DAG. This is explained in Chapter 6.

of the *explicitly included* variables and so the d-separation claims are removed from the union basis set. If you want to make this assumption, then the default choice of removing all pairs of exogenous variables from the union basis set makes sense, although it is better to deal with this by implicitly including such latent variables as explained in Chapter 6. If you don't want to make this assumption, if the causal independence of a pair if exogenous variables is part of your causal hypothesis, then you will have to do these tests of independence yourself and then include the additional null probabilities from these additional tests into the C statistic. You already know how to do these tests. For each pair of exogenous variables, you must do two regressions[72] by making each of the exogenous variables in the pair as a dependent variable and then using the smaller of the two resulting null probabilities. In the Blue Tits study, we already did this, and the smaller null probability was 0.5508. We then tell `psem()` to include this additional probability using the `add.claims=` argument of the summary object:

```
summary(fit.Blue.Tits, add.claims=c(0.5508),conserve=TRUE,
conditioning=TRUE).
```

Now, let's look at the output from the summary of our analysis. The first part of the output reproduces the structural equations associated with the DAG and gives the maximum likelihood AIC value[73]:

```
Structural Equation Model of fit.Blue.Tits

Call:
  hemato ~ protos + frass + mass
  mass ~ protos + frass
  recruited ~ hemato

    AIC
 12005.758
```

Next, it lists the d-separation claims in the union basis set minus any d-separation claims involving pairs of exogenous variables even though it has now included[74] the extra null probability for the d-separation claim involving protos and frass. It then gives Fisher's C statistic with its associated statistics for the global goodness of fit (ignore the line giving a chi-squared

[72] Unless both variables are normally distributed
[73] We will discuss the AIC statistic in Chapter 5.
[74] The summary now outputs the following message: "Fisher's C has been adjusted to include additional claims not shown in the tests of directed separation."

value, which refers to a different test). The degrees of freedom are now correct (8) for four d-separation claims in the union basis set and the C statistic and its null probability are the same as was obtained in our step-by-step analysis:

```
Global goodness-of-fit:

Chi-Squared = 2.259 with P-value = 0.52 and on 3 degrees of freedom
Fisher's C = 6.323 with P-value = 0.611 and on 8 degrees of freedom
```

Since the DAG has not been rejected, we can go on to look at the rest of the output. The path coefficients (i.e., regression slopes) of the structural equations are given under "Estimate". The standard errors of each path coefficient are given under "Std.Error". Next are the residual degrees of freedom ("DF") associated with each path coefficient, the z-statistic measuring the deviation of each estimated path coefficient from zero ("Crit.Value"), the probability testing the null hypothesis that each path coefficient is equal to zero ("P.Value") and the standardised estimate of each path coefficient (explained later):

```
Coefficients:

   Response Predictor Estimate Std.Error        DF Crit.Value P.Value
     hemato    protos  -0.5194    0.0315 1127.7380   -16.4828  0.0000
     hemato     frass  -1.0735    5.2284  173.5349    -0.2053  0.8376
     hemato      mass   1.9073    0.2588 1273.9885     7.3711  0.0000
       mass    protos  -0.0367    0.0033 1249.8012   -11.2381  0.0000
       mass     frass   3.7048    0.5669  174.6577     6.5349  0.0000
  recruited    hemato   0.0369    0.0166 1309.0000     2.2137  0.0268
  Std.Estimate
       -0.4410 ***
       -0.0085
        0.1907 ***
       -0.3117 ***
        0.2947 ***
        0.1633   *
```

Here are the estimated structural equations[75]:

$$hemato = -0.52\,protos - 1.07\,frass + 1.91 mass$$

$$mass = -0.04\,protos + 3.71\,frass$$

$$\ln\left(\frac{p_{recruited}}{1 - p_{recruited}}\right) = 0.04 hemato$$

[75] The last equation might seem strange. The recruited variable was modelled using a binomial distribution with a logit link function and so the resulting regression, with its associated path coefficient, is given for its logit-transformation.

Each of the estimated path coefficients corresponds to an arrow in the DAG. For instance, hemato had three arrows pointing into it from each of its causal parents (protos, frass and mass) and so there are three estimated path coefficients in the structural equation linking hemato to each of its three direct causes. Since we included each of these arrows in our DAG, this means that we expected each of the path coefficients to be different from zero. Looking at the null probabilities associated with each path coefficient ("P.Value"), we see that this is true for all but frass→hemato, for which the null probability was 0.84. Furthermore, we expected that a nestling having access to more food (i.e., frass) would have a larger haematocrit after holding constant its body mass and the number of ectoparasites that it is exposed to (i.e., protos) but the path coefficient is negative (-1.07). We now have a decision to make. A path coefficient that is not significantly different from zero can mean one of two things: (i) it really is zero or (ii) it is not zero, but the effect is so small relative to its standard error that we don't have enough statistical power[76] to detect such a small difference. There is no statistical method of deciding between these two possibilities and so you must use your biological knowledge. If you decide that the path coefficient associated with frass→hemato really is zero, then you can modify the DAG by removing this arrow and then test the new DAG after explaining why your biological knowledge suggests such a modification. If you decide that the path coefficient is really not zero, then you would not modify the DAG but you should explain why such a weak effect is biologically reasonable.

The last output that is produced by `summary()` are the proportions of the variance of each endogenous variable that is accounted for by its causal parents (i.e., the R^2 values of each regression). Since our structural equations are based on mixed models, there are two types of such estimates: marginal and conditional R^2 values. Conditional R^2 values exclude the variance accounted for by the random part of the model (i.e., the differences in intercepts between years and between nests) while the marginal R^2 values do not exclude this part of the variance. Here is the output:

```
Individual R-squared:
```

[76] You can estimate how big the effect would have to be in order to be detected with your data by calculating a 95% confidence interval around this path coefficient. A z-value that just reaches significance at the two-tailed 5% level (call it "d) is 1.96. Therefore, $z = \frac{|d|}{5.23} = 1.96$ and so d (path coefficient associated with frass→hemato) must be at least 10.25 in order to be detected.

```
   Response      method Marginal Conditional
     hemato        none     0.27        0.43
       mass        none     0.14        0.43
  recruited theoretical     0.03        0.08
```

Notice that, although the path coefficient from hemato to recruited[77] was significantly different from zero, the proportion of the variance that is accounted for by hemato was very low (8%), even if we exclude the variation between years and between nests. In other words, the majority of the causes affecting how likely a nestling will survive after fledging and then return to the same population the next year (which is what recruited is measuring) are not explicitly represented in the DAG. Remember that the goal of SEM is to properly model the causal structure between the variables, not to maximize the variance that is accounted for in any particular variable.

3.6 Independence of d-separation claims via the generalized covariance statistic

As you have seen, there is a weakness when testing the independence of d-separation claims between pairs of variables by testing for a zero slope in a regression: a regression requires choosing which variable in the pair is to be the dependent variable. In cases in which neither variable of the pair is a causal descendent of the other, this choice is arbitrary. Such an arbitrary choice makes no difference when both variables in the pair are normally distributed, but this is not true if either (or both) variables do not follow a normal distribution. If either of the two variables in the pair is not normally distributed, then the null probability that results from the test of a zero slope will differ depending on which variable of the pair is chosen as the dependent variable. This problem can be solved by using a second method of testing the independence of d-separation claims between pairs of variables, based on the generalized covariance statistic (Shah and Peters 2020).

[77] More exactly, $\ln\left(\frac{p_{recruited}}{1-p_{recruited}}\right)$.

A d-separation claim like X_i_||_X_j|{**Q**}, where **Q** is the set of causal parents of either X_i or X_j, generates the hypothesis that X_i is probabilistically independent of X_j after conditioning on the set of variables in **Q**. This means that the residuals of X_i, after conditioning on **Q**, will be independent of the residuals of X_j, after conditioning on **Q**. The residuals of a variable, after conditioning on **Q**, are obtained by subtracting its observed values from its expected values given **Q**. The first step is therefore to get the expected values of both X_i and X_j by treating both X_i and X_j as dependent variables and regressing each of X_i and X_j on the set of variables in **Q**, i.e. the causal parents of both X_i and X_j. The predicted values of X_i and X_j from these two regressions $\left(\hat{X}_i, \hat{X}_j\right)$ are the expected values of each given **Q**. For example, if X_i follows a normal distribution and each of the conditioning variables in **Q** are linked to X_i by a linear function, then the expected value of X_i given **Q** can be obtained by regressing X_i on the variables in **Q** using a linear regression. Different assumptions concerning the distribution of X_i and the functional form of the link between it and each of the variables in **Q** will result in different types of regression: generalized linear models, mixed models, generalized linear mixed models or generalized additive models.

The second step is to obtain the residuals of these two regressions by subtracting the predicted values from the observed values of each of X_i and X_j (Equations 3.1a,b). These residuals are called "response" residuals in R. In the case of mixed model regressions, appropriate for nested data, the residuals are those obtained after taking into account both the fixed and the random components of the model.

$$r_i = X_i - \hat{X}_i$$
$$r_j = X_j - \hat{X}_j$$

Equation 3.1(a, b)

The third step is to measure the degree of association between the two vectors of residuals using the generalized covariance statistic (T, Equation 3.2) irrespective of the type of regressions that have generated these residuals. The generalized covariance statistic[78], and its properties, were derived in (Shah and Peters 2020). Of course, the generalized covariance statistic only correctly measures this association if the correct type of regression has been used relative to the data. The

[78] Shah and Peters (2020) call this the generalized covariance "measure".

generalized covariance statistic is available from the GeneralizedCovarianceMeasure package on CRAN and the pwSEM package. The `pwSEM()` function uses (and outputs) the generalized covariance statistic when testing the d-separation claims in the union basis set and the `generalized.covariance()` function calculates it directly. This statistic is asymptotically distributed as a standard normal variate even when the residual values from Equation 3.1 are not normally distributed when the regressions are based on well-specified parametric or nonparametric models[79] (van der Vaart 1998, Shah and Peters 2020 , theorem 6). In other words, the actual sampling distribution of T becomes closer and closer to a standard normal distribution as the sample size of the data increases. No published simulation studies have yet been published but my simulations suggest that a minimum of about 100 independent observations are required before a standard normal distribution becomes a good approximation. However, it is easy to construct a randomization test that is accurate for small samples and that is quite quick to run on a modern computer; this test will be described later. Therefore, we can get the probability that X_i is independent of X_j, conditional on **Q**, by calculating the generalized covariance statistic and comparing it to a standard normal distribution. There is no need to decide which, X_i or X_j, is the dependent variable in a regression because both are dependent variables in their respective regressions. We can do this for each d-separation claim in the union basis set and then obtain a global test of the DAG by combine these probabilities using Fisher's C statistic.

$$R_k = r_i \cdot r_j$$

$$T = \frac{\frac{1}{\sqrt{n}}\sum_{k=1}^{n} R_k}{\sqrt{\frac{1}{n}\sum_{k=1}^{n} R_k^2 - \left(\frac{1}{n}\sum_{k=1}^{n} R_k\right)^2}} \qquad \text{Equation 3.2(a,b)}$$

3.7 Independence of d-separation claims via the generalized covariance statistic using the pwSEM package

[79] This applies for models that use M-estimators (maximum likelihood type estimators). The difference between the estimator (here, the predicted value, or mean, of the regression) and the true parameter scales as $1/\sqrt{n}$ and the distribution of the scaled difference converges to a normal distribution.

The pwSEM package[80] automates all of these steps using a syntax that is similar to, but not identical with, the piecewiseSEM package. In other words, you input the regression equations based on your DAG and the `pwSEM()` function constructs the DAG, obtains the union basis set of d-separation claims, tests the associated null hypotheses of conditional independence using the generalized covariance statistic, calculates Fisher's C statistic and then estimates the free parameters of the regressions based on your DAG. Compared to `psem()`, one difference in the syntax is that you must explicitly model each exogenous variable as well as the endogenous variables. Since an exogenous variable has no causal parents, this means "regressing" each exogenous variable only on its intercept. The `pwSEM()` function is based on the `gam()` and `gamm4()` functions of the mgcv and gamm packages (Wood 2017).

The `gam()` function fits generalized linear models and generalized additive models (i.e. generalized nonlinear models based on regression smoothers). Its basic syntax is `gam(formula, family= , data= )`. The formula object uses the common R syntax. Thus `gam(Z~X+Y, family= , data= )` is a generalized linear model. A formula object like `gam(Z~s(X)+Y, family= , data= )` will fit a generalized additive regression with a nonlinear function for X using a regression smoother. The pwSEM package only uses the default cross-validation choice for the degree of smoothing. The `gamm4()` function fits generalized linear mixed models and generalized additive mixed models (i.e. generalized nonlinear mixed models based on regression smoothers). Its basic syntax is `gamm(formula, random= , family= , data= )` Its formula object uses the same syntax as `gam()` and its random argument uses the same formula specification as the lmer package that we already used in the piecewiseSEM package.

There are three steps in using the `pwSEM()` function: (i) create a list object containing the regression equations (using gam and/or gamm4) for each variable in your DAG, (ii) call the `pwSEM()` function, and (iii) calling the `summary()`. Here is how to create a list object containg the regression equations to fit the DAG used in the Blue Tits example:

[80] Available on CRAN. The development version is at https://github.com/BillShipley/pwSEM, and updates and bug fixes will be placed there first.

```
my.list<-list(
gamm4(protos~1, random= ~(1|nest)+(1|year), family="poisson",
data=BlueTits),
gamm4(frass~1, random= ~(1|nest)+(1|year), family="gaussian",
data=BlueTits),
gamm4(mass~protos+frass, random= ~(1|nest)+(1|year),
family="gaussian", data=BlueTits),
gamm4(hemato~protos+frass+mass, random= ~(1|nest)+(1|year),
family="gaussian", data=BlueTits),
gamm4(recruited~hemato, family="binomial",
random=~(1|nest)+(1|year), data=BlueTits)
)
```

The first structural equation regression in the list is a mixed model generalized linear regression of the variable protos on none of the other variables in the DAG. As in every regression syntax in the R language, the variable "1" means the intercept. A regression model like "X~1" means that X is to be regressed only on the intercept, which is equivalent to estimating the mean of X. The first argument of the first regression (`gamm4(protos~1,`) specifies that protos is to be regressed[81] on none of the other variables in the DAG since no other variables are included after tilde (~). We do this because `protos` is an exogenous variable. Every exogenous variable must have an explicit regression model in which it is regressed against only the intercept. The second argument (`random= ~(1|nest)+(1|year)`) specifies the nesting structure of the data, thus specifying that mean of `protos` will vary randomly between nests and between years. The third argument (`family="poisson"`) specifies that `protos` is a count variable that follows a Poisson distribution. This means that the transformation of `protos` (the link function) will be `ln(protos)`, which renders the model linear on a natural logarithmic scale. The following four structural equation regressions in this list give the four remaining variables involved in the DAG. There will always be as many structural equation regressions in this list as there are variable, both exogenous and endogenous, in the DAG.

Next, we conduct the dsep test and obtain the structural equations using the `pwSEM(sem.functions=,data=, do.smooth=FALSE,all.grouping.vars=))` function. The `sem.functions` argument requires the list object that we just created. The `do.smooth` argument (TRUE/FALSE) specifies if you want to use nonlinear smoother

[81] Using a generalized additive mixed model, thus the name gamm.

regressions to fit and test each d-separation claim. The `all.grouping.vars` argument has a default value of NULL if there is no nesting structure in any of the structural equations; otherwise, you must supply a character vector giving the names of each of the grouping variables involved in any of the random parts of the structural equations. Here is the call for the Blue Tits example:

```
fit<-pwSEM(sem.functions=my.list,data=BlueTits, do.smooth=FALSE,
     all.grouping.vars=c("nest","year"))
```

Finally, you pass the object created by `pwSEM()` to the `summary()` function to obtain the results. By default, the estimated structural equations are not returned but you can obtain these using the `structural.equations=T` argument:

```
summary(fit,structural.equations=T)
```

The first part of the result prints out the dsep test. It first prints out the hypothesized causal graph (here, a DAG), the d-separation claims in the union basis set and the null probabilities of the generalized covariance statistic for each one, the C-statistic and it associated information, and the maximum likelihood AIC statistic (explained in Chapter 5):

```
Causal graph:
protos ->mass
frass ->mass
mass ->hemato
protos ->hemato
frass ->hemato
hemato ->recruited

Basis Set
( 1 )  mass _||_ recruited | { protos frass hemato }
( 2 )  protos _||_ frass | {  }
( 3 )  protos _||_ recruited | { hemato }
( 4 )  frass _||_ recruited | { hemato }

Null probabilities of d-separation claims in basis set
(1) 0.4816243
(2) 0.7674164
(3) 0.8250628
(4) 0.08063539

C-statistic: 7.41086 , df = 8 , null probability: 0.4930205

AIC statistic: 16462.11
```

Notice that the d-separation claim involving the pair of exogenous variables (protos and frass) is included in the union basis set. By default, `pwSEM()` always includes d-separation claims between pairs of exogenous variables. If you want to exclude such d-separation claims, then you

would have to explicitly include dependent errors (called “correlated errors” or “free covariances” when the variables are normally distributed) between such pairs of exogenous variables; this will be explained in Chapter 6.

Finally, because we added the `structural.equations=T` argument to the summary function, each of the regressions pertaining to the structural equations are output. Since the first two equations are for the two exogenous variables, only the intercepts are estimated; an intercept without any predictor variables is simply the mean. Since protos was modelled following a Poisson distribution, its estimated intercept (i.e. mean) is for ln(protos). The remaining regressions are the same as those output using the piecewiseSEM library.

```
_______Piecewise Structural Equations_________

Structural equation  1 :  protos ~ 1

         Parametric coefficients:
            Estimate Std. Error  z value     Pr(>|z|)
(Intercept) 1.847986   0.108269 17.06847 2.547787e-65
____________________

Structural equation  2 :  frass ~ 1

         Parametric coefficients:
              Estimate Std. Error  t value     Pr(>|t|)
(Intercept) 0.09180031 0.02059695 4.456986 9.022931e-06
____________________

Structural equation  3 :  mass ~ protos + frass

         Parametric coefficients:
               Estimate  Std. Error   t value     Pr(>|t|)
(Intercept)  9.42185545 0.091972215 102.44241 0.000000e+00
protos      -0.03669822 0.003054068 -12.01618 1.269183e-31
frass        3.70479680 0.317245709  11.67800 4.848599e-30
____________________

Structural equation  4 :  hemato ~ protos + frass + mass

         Parametric coefficients:
              Estimate Std. Error     t value     Pr(>|t|)
(Intercept) 28.6017978 2.42468306  11.7960975 1.374977e-30
protos      -0.5194103 0.02969701 -17.4903198 1.059272e-61
frass       -1.0734843 3.08122049  -0.3483958 7.275992e-01
mass         1.9073273 0.24276097   7.8568121 8.182176e-15
____________________

Structural equation  5 :  recruited ~ hemato

         Parametric coefficients:
               Estimate Std. Error   z value     Pr(>|z|)
(Intercept) -4.65096386 0.70550458 -6.592394 4.327913e-11
hemato       0.03685744 0.01626577  2.265951 2.345439e-02
```

3.8 Generalising the dsep test even further

There are five steps in any dsep test:

(1) Write down your causal hypothesis in the form of a DAG;

(2) Obtain the union basis set of dsep claims from this DAG.

(3) Conduct the series of statistical tests of (conditional) independence associated with each dsep claim in the basis set using whatever statistical tests are appropriate for each null hypothesis;

(4) Combine the null probabilities, obtained from step (3), using Fisher's C test.

(5) Reject your causal hypothesis if the null probability from Fisher's C test is below your chosen significance level; otherwise conclude that your data are consistent with your causal hypothesis.

Stated like this, the dsep test is less a statistical "test" than a recipe for creating your own test based on the specific properties of your data. So long as you can (i) specify your causal hypothesis in the form of a DAG[82], (ii) obtain direct measures of each variable in this DAG, and (iii) perform an appropriate test of (conditional) independence for each dsep statement then you can create your very own dsep test. For instance, evolutionary biologists studying the relationships between organismal traits sometimes want to remove any phylogenetic signal before considering any causal hypotheses linking the traits. They do this by using "phylogenetic" regressions (Grafen 1989). At this time, neither piecewiseSEM nor pwSEM can accommodate such regression methods, but you can easily use such phylogenetic regressions in a dsep test by using them in step 3. Even more exotic tests of independence can be used. For instance, Frenette-Dusseault et al. (2013) studied changes in a series of average functional traits[83] of ants and plants along a gradient of increasing aridity in the Moroccan steppes. They wanted to test between two different causal explanations. The first possibility was that changes in environmental conditions along the aridity gradient directly causes changes in the average trait

[82] Or a MAG (mixed acyclic graph); see Chapter 6.
[83] These are called "community-weighted" traits because they are the trait values of each species multiplied by the relative abundance of each species in a given site.

values of both the ant and plant communities: plant community trait composition←environmental changes→ant community trait composition. In this scenario, correlations between the average trait values of plants and ants are "spurious" because each is responding to the same environmental changes. The basis set consists of a single d-separation claim: plant community trait composition _||_ ant community trait composition |{ environmental changes }.The alternative explanation is that changes in the average trait values of the plant communities are directly caused by changes in the environment but the changes in the average ant traits are directly caused by the changes in the average plant traits: environmental changes→plant community trait composition→ant community trait composition. The basis set consists of a single d-separation claim: environmental changes _||_ ant community trait composition | {plant community trait composition}. The data consisted of three dissimilarity matrices: the Bray-Curtis dissimilarities (Shipley 2021) between samples measured on the average trait values of each plant (or ant) community and the Euclidian distance between samples measured on the environmental values of each sample. The basis sets for these two alternative causal explanations consist of a single d-separation claim each but these d-separation claims cannot be tested by any type of regression model because we need to test for conditional independence between pairs of matrices, not vectors of observations. Instead, conditional independence was tested using a partial Mantel test (Legendre and Legendre 2012), which is appropriate for these data[84].

3.9 Interpreting and manipulating path coefficients

Beginning users of dsep tests, when using tests of independence based on zero slopes from regression equations, sometimes get confused because there are actually two different sets of regressions. The first set of regressions are those used to evaluate the d-separation claims in the union basis set. The only useful information in these regressions are the probabilities associated with the null hypothesis of a zero slope that is used in the C statistic; we don't really care about the other parameters in these regressions. The second set of regressions, which are only

[84] They could also have used a partial Procrustes test (Jackson 1995).

performed if the DAG is not rejected, are those that estimate the structural equations themselves. In other words, you fit a series of regressions in which each endogenous variable is regressed only on its causal parents. This second set of regressions form the "piecewise" regressions that follow the DAG itself, which produce the "path coefficients". We call these second set of regressions "structural" equations because they are set up to exactly follow the structure of the DAG, i.e., the way the variables are linked together in the DAG. As you will see, the term "path coefficient" is only strictly appropriate for linear structural equations because only in such linear regressions are the direct causal effects of a parent on its child constant (thus, producing a "coefficient"). For nonlinear structural equations, it is more appropriate to talk about a path effect "function".

Consider a general structural equation like $Y = f_1(X_1) + f_2(X_2) + \varepsilon$. Here, Y is the causal child and X_1, X_2 and ε are its causal parents; as usual, ε represents the other, unknown, causes of Y that are not included in the DAG and that are represented as the residuals of the regression. The path effect function linking Y and X_1 is the partial derivative of Y given X_1, i.e. $\partial Y/\partial X_1$. This function quantifies by how much a unit change in the value of X_1 will change Y, holding constant X_2 and ε. In general, a path effect function quantifies by how much a unit change in the causal parent will change its causal child when holding constant every other variable in the DAG[85]. This is called a "direct" effect and is what is associated with the arrow in the DAG from X_2 to Y. This definition of a path effect function might sound familiar to you because it is also the definition of a partial slope coefficient in a linear regression. For instance, given a regression equation like $Y_j = 1.2 + 0.5X_{1j} - 3.2X_{2j} + \varepsilon_j$, then the partial derivative of Y given X is $\partial Y/\partial X_1 =$ 0 + 0.5 + 0 + 0 = 0.5, which is the same as the partial slope of X_1 in this regression. For linear structural equation models, the partial slopes of the regression are therefore called path "coefficients" because the path effect function (i.e., $\partial Y/\partial X_1$) is a constant. In other words, a unit increase in X_1 will cause a 0.5 unit increase in Y *irrespective* of the value of X_1. However, if the structural equation is nonlinear (say, $Y_j = 0.1e^{0.5X_{1j}} - 3.2X_{2j} + \varepsilon_j$) then the path effect function for

[85] You must not make the mistake of thinking that a path effect function quantifies by how much the causal parent changes the causal child when holding constant variables NOT included in the DAG.

X_1 would be $\partial Y / \partial X_1 = 0.5 \cdot 0.1 e^{0.5 X_{1j}}$. Now, a unit increase in X_1 would change Y by different amounts, depending on the value of X_1.

Given this general definition of a path effect function, and its equivalent path coefficient if the structural equation is linear, we can now begin to combine these path effect functions in order to measure the indirect effects of causal ancestors on causal descendants along different directed paths. Consider the following DAG: $X_1 \rightarrow X_2 \rightarrow X_3$. The amount by which a unit change in X_1 will change the value of X_3 if we hold constant every other variable in the DAG *except* for X_2 (i.e., the indirect effect of X_1 on X_3 via X_2) is:

$$\frac{\partial X_3}{\partial X_1} = \left(\frac{\cancel{\partial X_2}}{\partial X_1} \right) \left(\frac{\partial X_3}{\cancel{\partial X_2}} \right).$$

This leads to the first rule for calculating indirect effects of a causal ancestor on its causal descendant along a single directed path: *you multiply together the path effect functions along that directed path*. If the relationships between X_1, X_2 and X_3 are linear, meaning that the path effect functions are constants, then you multiply together the path coefficients along that directed path. In general, the path effect function of a causal ancestor on its causal descendant along a single path measures by how much a unit change in the causal ancestor will change the causal descendant if you were to hold constant every other variable in the DAG *except* for those variables along that directed path.

Here is the second rule required to calculate indirect effects of a causal ancestor on its causal descendant: *If there is more than one directed path from a causal ancestor to its causal descendant, then the total indirect effect of that causal ancestor on its causal descendent is calculated by summing together the indirect effects along each of the different directed paths linking them.* Such a total indirect effect of a causal ancestor (X_i) on its causal descendant (X_j) measures by how much a unit change in X_i will induce a change in X_j if all variables in the DAG except for those variables along any of the directed paths from X_i to X_j. Combining these two rules gives the total causal effect of a causal ancestor on its causal descendent, which is to the sum together its direct and indirect effects.

Two variables in a DAG can also have a non-causal dependency. This occurs when the two variables share a common causal ancestor. Such a non-causal dependency is sometimes called a

"spurious" association. An example of such a spurious association in the one between variables X_3 and X_4 in Figure 3.1. X_3 is not a causal ancestor of X_4, nor is X_4 a causal ancestor of X_3, since there are no directed paths from one to the other. However, variables X_3 and X_4 will still be associated in any data generated from this DAG because both are caused by changes in variable X_2, which is their common causal parent. *Such a noncausal association can be quantified by multiplying together the direct (or indirect) effects from B (the common causal ancestor) to each of C and D.* As always, if all of the relationships are linear, then this reduces to multiplying together the path coefficients along the two directed paths from B to C and from B to D. *If two non-causally linked variables share more than on common causal ancestor, then the total noncausal association that is generated by all of these common causal ancestors is obtained by summing together each of the individual spurious relationships*. Such noncausal associations are often called "spurious" correlations.

Let's apply these rules to the model fitted using the DAG in Figure 3.1. Since the variables in this DAG are normally distributed and linearly related to one another, the path effect functions are all constants, thus path "coefficients". Figure 3.2 shows the resulting path diagram.

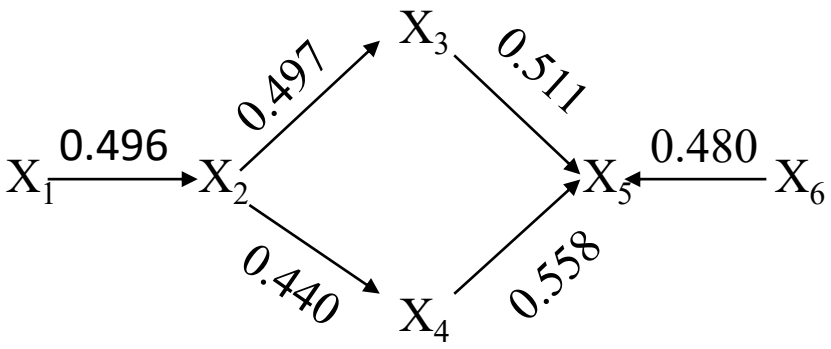


Figure 3.2. A DAG including path coefficients associated with each direct causal effect.

The indirect effect of X_1 on X_5 along the directed path $X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_5$ in Figure 3.2 is (0.496)(0.497)(0.511)=0.126. The indirect effect of X_1 on X_5 along the directed path $X_1 \rightarrow X_2 \rightarrow X_4 \rightarrow X_5$ in Figure 3.2 is (0.496)(0.440)(0.558)=0.122. The total indirect effect of X_1 on X_5 along both directed paths is 0.126+0.122=0.248. The spurious effect between X_3 and X_4 due to their common causal parent (X_2) is (0.497)(0.440)=0.219.

Now, let's apply these rules to the model fit to the Blue Tits data. Figure 3.3 shows the path diagram.

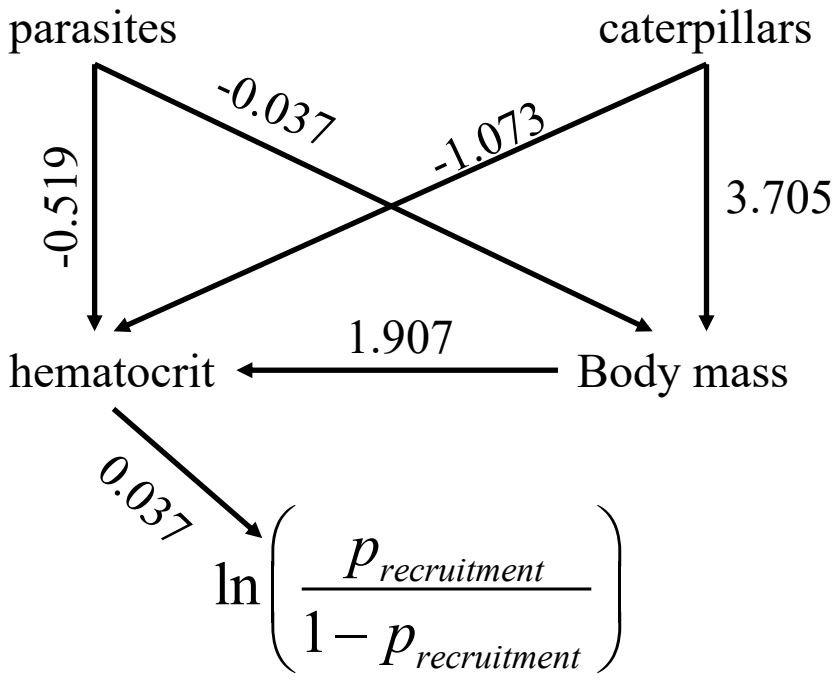


Figure 3.3. The final DAG for the Blue Tits data, with the path coefficients included, based on ln-transformed values of the probability of recruitment.

All the variables in Figure 3.3 are normally distributed except for recruitment, which is binary. For a binary variable, we don't model its actual value (yes/no) but rather the probability that it will be yes (successful recruitment) or no (unsuccessful recruitment). Furthermore, since a binary variable follows a binomial distribution, it is transformed to its log-odds ratio (i.e. ln(p/(1-p)) in the generalized linear model in order to linearize its relationship. This means that, while the path function for the transformed log-odds ratio value is linear, and its path effect function is a constant (here, 0.037), the path function is nonlinear if we back-transform in order to express our variable as the probability of successful recruitment ($p_{recruitment}$):

$$p_{recruitment} = \frac{e^{0.037\,hematocrit}}{1+e^{0.037\,hematocrit}} \ .$$

Taking the derivative of this function gives the path effect function, $\left(\partial p_{recruitment}/\partial hematocrit\right)$, which is not a constant (i.e. a path coefficient):

$$\frac{0.037e^{0.037\,hematocrit}}{1+e^{0.037\,hematocrit}}.$$

Notice that the effect of changing the haematocrit volume by one unit now depends on the value of the haematocrit of the nestling. If the nestling has a low haematocrit, then increasing it by one unit will increase its probability of successfully recruiting into the population more than if the nestling already has a large haematocrit. This results in Figure 3.4.

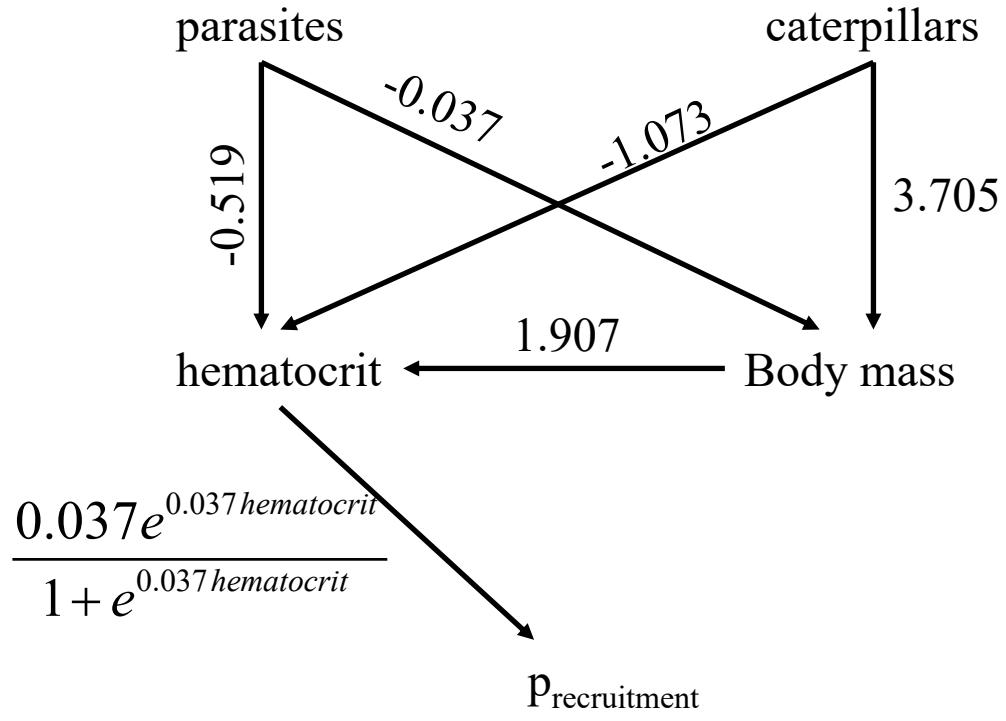


Figure 3.4. The final DAG for the Blue Tits data, with the path coefficients included, based on the original values of the probability of recruitment.

Now we can calculate the indirect effect of (for example), of changing the parasite load of a chick on its probability of being recruited back into the population the next year. There are two such indirect paths: parasites→haematocrit→$p_{recruitment}$ and parasites→body mass→haematocrit→$p_{recruitment}$. Multiplying together the path effect functions for each indirect path gives:

$$(-0.519)\left(\frac{0.037e^{0.037\,hematocrit}}{1+e^{0.037\,hematocrit}}\right)$$
$$(-0.037)(1.907)\left(\frac{0.037e^{0.037\,hematocrit}}{1+e^{0.037\,hematocrit}}\right).$$

The result is shown in Figure 3.5, and we clearly see that the causal effect of a chick having one additional parasite on its probability of successfully returning to the population the next year changes depending on the size of its haematocrit. Adding a parasite to a chick that already has a small haematocrit decreases the probability of successfully recruiting more than does adding a parasite to a chick that has a larger haematocrit. The pwSEM package has a function, `view.paths()`, that calculates and plots the indirect effects such as those shown in Figure 3.5.

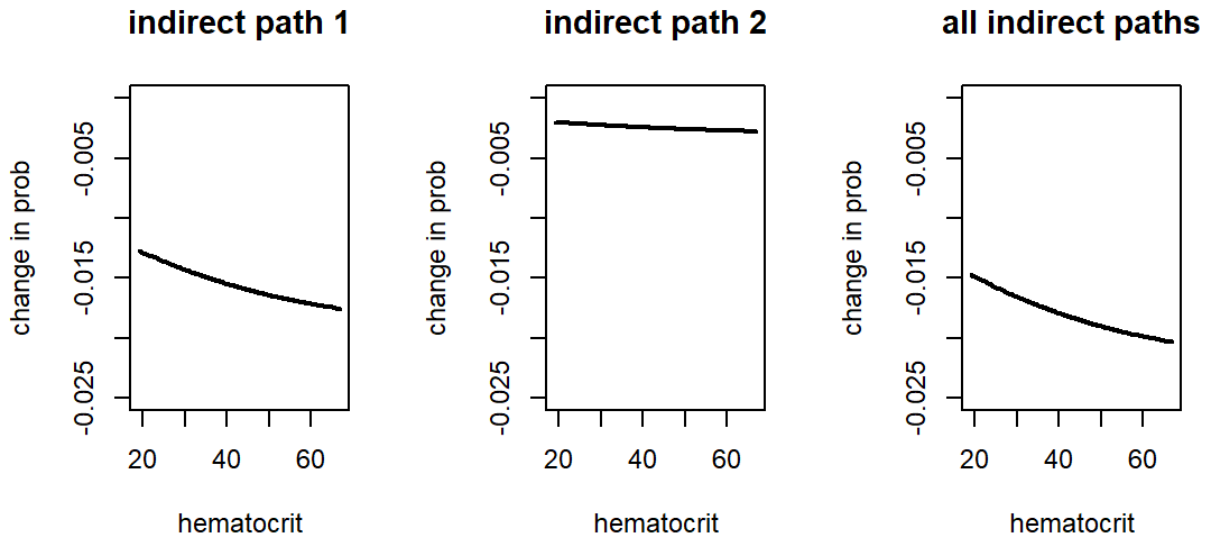


Figure 3.5. The output of the `view.paths()` function, showing by how much the addition of a single nest parasite will indirectly change the probability of a nestling being successfully recruited into the population along each of the two separate indirect paths and the total indirect effect of both indirect paths together.

3.10 Permutation tests of independence

Remember the definition of probabilistic independence given in Chapter 2. We know that if X and Y are independent then the probability of observing any particular value of Y is the same whether or not we know the value of X. In other words, any value of X is just as likely to be paired with any other value of Y as with the particular Y that we happen to observe. The permutation test works by making this true in our data. After calculating our measure of association in our data, we randomly rearrange the values of X and/or Y using a random number generator. In this new randomly mixed "data set" the values of X and Y really are independent because we have forced them to be so; we have literally forced our null hypothesis of independence to be true and the value of the association between X and Y is due only to chance. We do this a very large number of times until we have generated an empirical frequency distribution of our measure of association[86]. This empirical frequency distribution is an estimate of the actual sampling distribution of our measure of association and this estimate becomes more and more accurate as the number of times we randomly permute our values of X and/or Y. Why would we want to do this? We do this when we do not already have a mathematical function describing our expected sampling distribution. In particular, since the sampling distribution of the generalized covariance statistic is only a standard normal distribution at large sample sizes (~100 observations), a permutation version of its sampling distribution can be used when sample size is not large enough to use a standard normal distribution. The exact number of times that we randomly permute our data will depend on the true probability level of our actual data and the accuracy that we want to obtain in our probability estimate. Manly (1997) shows how to determine this number, but it is typically between 1000 and 10000 times. On modern computers this will appear instantaneous unless the intermediate calculations are intensive. The last step is to count the proportion of times that we observe at least as large a value of association within the permuted data sets, or its absolute value for a two-tailed test, as we actually observed in our original data. This proportion is an estimate of the true null probability. As the number of randomly permuted data sets increases, the precision of this estimate increases. You can calculate the 95% confidence interval around this estimate by: $1.95\sqrt{p(1-p)/N}$ where p is the estimated null probability and N is the number of independently generated permuted data sets.

[86] For small samples one can generate all unique permutations of the data. The use of random permutations, described here, is generally applicable and the estimated probabilities converge on the true probabilities as the number of random permutations increase.

Since the covariance statistic is a measure of association, the null probability of this statistic can be estimated by this method when the sample size of your data set is too small[87] to rely on asymptotic results. You can obtain probability estimates based on a permutation distribution in the `pwSEM()` function simply by adding the argument `use.permutations=TRUE`. The default number of permutation runs is 5000, which is sufficient in almost all cases, but you can change this number by also including the argument `n.perms=` in the `pwSEM()` function. You can also get the permutation version of the generalized covariance statistic directly via the `perm.generalized.covariance()` function in the pwSEM package.

[87] Around 100 observations.

4

Covariance-based SEM without explicit latent variables

4.1 Origins and history of covariance-based SEM

James Burke (Burke 1996), in his fascinating book, *The Pinball Effect*, demonstrates the curious and unexpected paths of influence leading to most scientific discoveries. People often speak of the "marriage of ideas". If so then the most prolific intellectual offspring come, not from the arranged marriages preferred by research administrators, but from chance meetings and even illicit unions. The popular view of scientific discoveries as being linear causal chains from idea to solution is profoundly wrong; a better image would be a tangled web with many dead ends and broken strands. If much of present knowledge depends on unlikely chains of events and personalities, then what paths of discovery have been deflected because the right people did not come together at the right time? Which historical developments in science have been changed because two people, each with half of the solution, were prevented from communicating due to linguistic or disciplinary boundaries? The development of modern covariance-based structural equation modelling is a case study in such historical contingencies and interdisciplinary incomprehension. Although I call this method "covariance-based" SEM (for reasons that will become clear in this chapter), it is also know by several other names including "classical" SEM or LISREL modelling[88].

[88] LISREAL is the earliest popular computer program for conducting this type of SEM.

During the First World War, and in connection with the American war effort, Sewall Wright was on a committee allocating pork production to various states based on the availability of corn[89]. He was confronted with a problem that had a familiar feel. Given a whole series of variables related to corn availability and pork production, how do all these variables interact to determine the relationship between supply and demand, and the fluctuations between these two? It occurred to him that his new method of path analysis might help. He calculated the correlation coefficients between each pair of variables over five years, giving 510 separate correlations. After much trial and error, he developed a model involving only four variables (corn price, summer hog price, winter hog price and hog breeding) and only fourteen paths that still gave a "good match" between observed and predicted correlations. He described his results in a manuscript that was submitted as a bulletin of the US Bureau of Animal Industry. It was promptly rejected, perhaps because officials at the Bureau of Agricultural Economics considered it as an intrusion onto their turf. Happily, for Wright, he had also shown it to the son of Henry A. Wallace the secretary of agriculture, who was interested in animal breeding and quantitative modelling. Wallace, using his political influence, intervened to have the manuscript published as a USDA bulletin (Wright 1925).

Although economists and sociologists later developed methods that were very similar to path analysis, Wright's foray into economics does not seem to have been very influential. During the Second World War, Wright presented a seminar on path analysis to the Cowles Commission[90], where economists were developing methods (simultaneous equations) that were the forerunner of structural equations modelling. Neither Wright nor the economists recognised the link between the two approaches or the usefulness of such a marriage (Epstein 1987). Nonetheless, some economists were independently trying to express causal processes in functional form[91] (Haavelmo 1943). In economics, constraints on the covariance matrix (for example, zero partial correlations) were called "overidentifying constraints". Since most work in this area was in parameter estimation, not theory testing, such constraints were mostly avoided because they made consistent estimation difficult.

[89] This next section is based on Wright's biography (Provine 1986).
[90] The Cowles Commission was established in 1932 by the economist Alfred Cowles III. It combined economic theory and statistical methods to lay the foundation for modern econometrics.
[91] Some economists referred to Wright's work in passing (Goldberger 1972; Griliches 1974) but only for historical completeness.

In the 1950's the political scientist Herbert Simon began to derive the causal claims of a statistical model[92]. This led some social scientists to think about expressing causal processes as statistical models that implied certain structural, or "overidentifying", constraints. For such people, overidentifying constraints weren't something to avoid; rather, they were hypotheses linked to causal claims, that could be tested with data. One such person was Hubert M. Blalock Jr., who began deriving overidentifying constraints, in the form of zero partial correlations, that were implied by the structure of the causal process (Blalock 1961, 1964). Indeed, Blalock cited Wright in this 1961 book "Causal inferences in nonexperimental research". Wright's method of path analysis had been rediscovered by social scientists with the important difference that the emphasis shifted from being an *a posteriori* description of an assumed causal process, as Wright viewed his method, to being a test of a hypothesised causal process. The late 1960's and early 1970's saw many applications of path analysis in sociology, political science and related social science disciplines.

The most important next step was the work of people like Jöreskog (1967, 1969, 1970, 1973) and Keesling (1972), who developed ways of combining confirmatory factor analysis (see Chapter 7) and path analysis using maximum likelihood estimation techniques. The advance was not simply in using a new method of estimating the path coefficients. More importantly, the use of maximum likelihood allowed the resulting series of equations describing the hypothesised causal process (a series of *structural equations*) to be tested against data to see if the overidentifying constraints (the zero partial correlation coefficients and other types of constraints) agreed with the observations. This advance solved the main weakness of Wright's original method of path analysis since one did not simply have to *assume* the causal structure, as Wright did. Now, one could test the statistical consequences of the causal structure and therefore potentially falsify the hypothesised causal structure[93]. Unfortunately, by the 1970's most biologists had forgotten about

[92] Summarized in Simon (1977).

[93] The logical and axiomatic relationships between probability distributions and causal properties had not yet been developed. This led to much confusion concerning the causal interpretation of structural equation models (Pearl 1997). One reason why I discuss these points in detail is to prevent the same sterile debates from recurring between biologists.

Wright's method of path analysis and disciplinary boundaries prevented the new covariance-based structural equations modelling (SEM) approach from penetrating into biology[94].

Wright's method was essentially the application of multiple regression based on standardised variables in the order specified by the path diagram (the causal graph). This, along with most other familiar statistical methods, consists of modelling the individual observations. In other words, the path coefficients were obtained using least-square techniques by minimising the squared differences between the observed and predicted values of the individual observations. Piecewise SEM also models the individual observations. Covariance-based SEM, of which covariance-based path analysis is a special case, doesn't model the individual observations. Instead, it concentrates on the pattern of covariation between the variables (the covariance matrix of the observed variables) and minimises the difference between this observed covariance matrix and the predicted pattern of covariation (a predicted model covariance matrix) based on the causal claims of independence and conditional independence encoded in the path diagram.

The goal of this chapter is to describe how covariance-based SEM works. We will use the lavaan package of R (Rosseel 2012) to do our analyses. The advantage of covariance-based SEM over piecewise SEM is that explicit latent variables can be included in the model, given certain conditions. It also has disadvantages relative to piecewise SEM, which will be explained later. Although piecewise SEM can model implicit latent variables (Chapter 6), it cannot model explicit latent variables. However, I want to postpone a discussion of explicit latent variables until Chapter 7. In this chapter, I explain covariance-based SEM without explicit latent variables (i.e. path analysis). Once you have mastered this chapter then the addition of explicit latent variables, in Chapter 7, will be much easier to understand.

There are five basic steps in covariance-based SEM. Only steps three and four are truly different from those used in piecewise SEM.

1. Specify the hypothesised causal structure of the relationships between the variables in the form of a causal diagram (often, but not exclusively, a DAG).

[94] A literature search in Scopus, using the key words "structural equation" or "path analysis" records 4336 publications in biological journals in 2023 but almost none before about 2000.

2. Write down the set of linear equations that follow this causal diagram and specify which parameters (slopes, variances, covariances) are to be estimated from the data (i.e. that are "*free*") and which parameters are not to be changed to accommodate the data (i.e. *"fixed"*) based on the causal hypothesis.

3. Derive the predicted variance and the covariance between each pair of variables in the model using covariance algebra.

4. Estimate these free parameters using maximum likelihood or related methods, while respecting the values of the fixed parameters. This estimation is done by minimising the difference between the observed covariances of the variables in the data and the covariances of the variables that are predicted by the causal model.

5. Calculate the probability of having observed the measured minimum difference between the observed and predicted covariances, assuming that the observed and predicted covariances are identical except for random sampling variation. If the calculated probability that the remaining differences between observed and predicted covariances is due only to sampling variation is sufficiently small (say, below 0.05) then one concludes that the observed data were not generated by the causal process specified by the hypothesis and that the proposed model be rejected. If, on the contrary, the probability is sufficiently large (say, above 0.05) then one concludes that the data are consistent with such a causal process.

4.2 Translating the hypothetical causal system into a path diagram

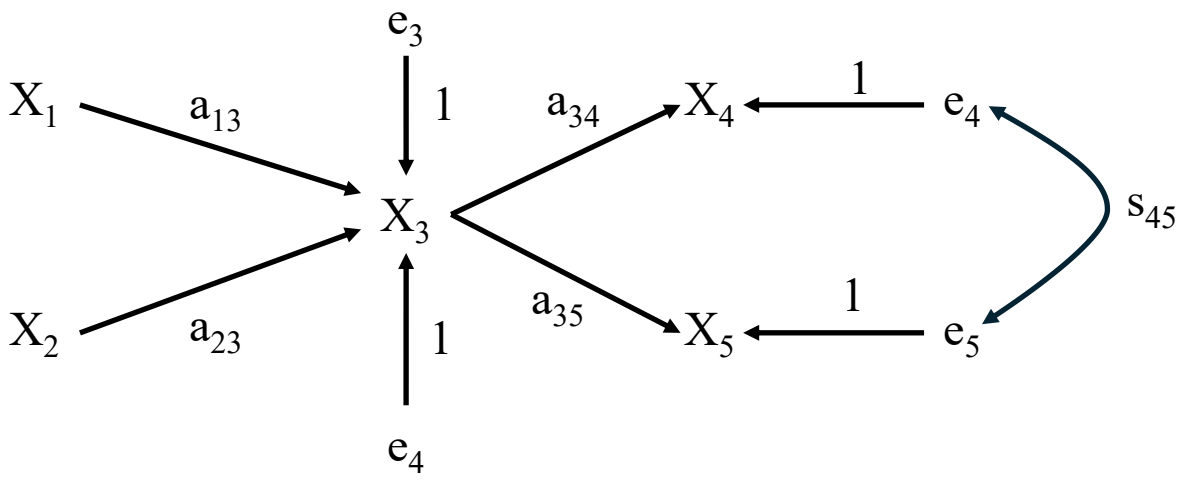


Figure 4.1. A path diagram with two exogenous observed, or "manifest", variables (X_1, X_2), three endogenous observed variables (X_3, X_4, X_5) of which two are terminal endogenous variables (X_4, X_5), and three exogenous error variables (e_3, e_4, e_5). Variables X_4 and X_5 also have correlated errors.

This first step should be almost second nature by now. Everything that you learned about translating causal hypotheses into DAGs applies here. In fact, a path diagram is very similar to a DAG[95] except that it can also include a double-headed arrow (↔) between pairs of variables. Another name for a path diagram is a mixed acyclic graph (MAG). I will explain the meaning of such double-headed arrows in more detail in Chapter 6, where I introduce MAGs and the notion of an implicit latent variable in the context of piecewise SEM.

There are a few notational conventions and jargon terms that must be introduced for you to follow the literature dealing with covariance-based SEM. Path diagrams contain three different types of variables: manifest, latent and error variables. Variables that have been directly observed and measured are called *manifest* variables in SEM jargon, but I will simply call them "observed" variables. Variables that are hypothesised to have a causal role and that are explicitly included in the path diagram, but which have not been directly observed or measured, are enclosed in circles; these variables are called (explicit) *latent* variables in SEM jargon[96]. I will postpone a discussion of such explicit latent variables to Chapter 7. In Figure 4.1 the variables

[95] In Chapter 6 you will learn that a DAG plus a double-headed arrow is called a mixed acyclic graph or MAG.
[96] By convention, a path model is simply a structural equation model that does not involve explicitly unmeasured, or latent, variables.

X_1 to X_5 are observed, or manifest, and there are no latent variables; you know this because none of the variables are enclosed in a circle. The third type of variable is the residual *error* variable. In Figure 4.1 the variables e_3 to e_5 are error variables. This type of variable represents all other remaining causes of the variable into which it points but that are not included in the model. In this sense, error variables are a special type of latent variable. Beginners often confuse these "error" variables in SEM with the error variable of a regression, but they are not the same thing! An error variable in a regression context is simply the residual variation not captured by the predictors and, by definition, such an error variable is uncorrelated with the predictors. This is not necessarily true in covariance-based SEM. The error variables in covariance-based SEM are always assumed to be normally distributed random variables and (usually) with a mean of zero. This assumption is implicit in the name (covariance-based SEM) because a covariance is a parameter in a normal distribution.

As in piecewise SEM, variables are also classified as *exogenous* (a variable that has no causal parents in the model) or *endogenous* (a variable that is caused by some other variable in the model. Variables X_1 and X_2 are observed exogenous variables. The error variables e_3 to e_5 are latent[97] exogenous variables. Variables X_3 to X_5 are observed endogenous variables. Endogenous variables that do not cause any other variable in the model are called *terminal* endogenous variables. Variable X_3 is an observed endogenous variable but not terminal. Variables X_4 and X_5 are observed terminal endogenous variables.

Finally, there are two types of arrows. A straight arrow indicates a causal relationship between two variables just as it does in the DAGs of previous chapters. A double-headed arrow ($X_i \leftrightarrow X_j$) indicates that neither variable causes the other but that both are caused by some unknown common cause; this is an example of an implicit latent variable that will be discussed in more detail in Chapter 6. A double-headed error *does not* mean a feedback relationship. Double-headed arrows can only exist between exogenous variables (either observed or latent, including the error variables). You will sometimes see double-headed arrows pointing to endogenous variables when the error variables are not included but such double-headed arrows are really pointing to the missing error variables associated with the endogenous variables. A double-

[97] Some authors do not classify error variables as "latent", but I will do this because these error variables represent all of the unknown (this *latent*) causes of the associated observed endogenous variable that remain after accounting for the explicit causal parents.

headed arrow between X_i and X_j means that there is a hypothesized statistical association between X_i and X_j due to some missing (perhaps unknown) common cause of both but neither X_i nor X_j is a cause of the other. Since covariance-based SEM assumes multivariate normality and linearity, this statistical association must be a covariance and so it is also called a "free covariance" or a "correlated error". Figure 4.1 shows a path diagram containing both causal claims between pairs of manifest variables (the arrows) and a free covariance between the errors associated with terminal endogenous variables X_4 and X_5. In other words, we are hypothesizing that among the unknown other variables that cause either X_4 or X_5 (i.e. excluding X_3), at least one of these unknown variables is a common cause of both X_4 and X_5, thus generating an association between X_4 and X_5 that is not due to the observed common cause X_3.

4.3 Translating the path diagram into a set of structural equations

This second step should also be familiar to you since it also exists in piecewise SEM, although there are a few differences. You must precisely translate the causal diagram (Figure 4.1) into linear equations describing the functional links between the variables and the multivariate normal probability distribution of the exogenous variables. Much of this second step is done by the lavaan program but it is important that you understand how to do it because lavaan takes many shortcuts (i.e. default values) without telling you and some of these shortcuts might not be what you want. When this is the case, you will have to explicitly tell lavaan what to do. Unlike in piecewise SEM, the functional links between the variables are always linear. Also unlike in piecewise SEM, the variables always follow a multivariate normal distribution[98]. Later I will explain how you can often get around the assumption of multivariate normality, but the assumption of linearity is strict. The only way you can accommodate nonlinear relationships is by transforming your variables beforehand to render the relationships linear; of course, transforming a variable will also change its distribution and this can introduce problems with respect to the assumption of normality.

[98] The assumption of mutually independent observations (for example, no nesting structures in the data) can be partially overcome; this will be explained later in this chapter.

When constructing our structural equations, we are usually (but not always) interested in the relationships between the variables (i.e. the slopes) rather than the mean values of the variables themselves[99] (i.e. the intercepts). For this reason, lavaan "centres" all observed variables by default . This is done by subtracting the mean value of each variable from each observation. For instance, if the mean of X_1 in Figure 4.1 was 6.2, then lavaan will replace each value of X_1 by (X_1-6.2). This trick ensures that the mean of each centred variable is zero and therefore that the intercepts are zero, which frees lavaan from having to estimate them. It is possible to override this default if you do want to estimate intercepts as well. Assuming that all of our variables are already centred, these are the structural equations corresponding to Figure 4.1, where Cov(X_1, X_2) means the population covariance between X_1 and X_2:

$$X_1 = N(0, \underline{\sigma_1})$$
$$X_2 = N(0, \underline{\sigma_2})$$
$$e_3 = N(0, \underline{\sigma_3})$$
$$e_4 = N(0, \underline{\sigma_4})$$
$$e_5 = N(0, \underline{\sigma_5})$$
$$X_3 = \underline{\alpha_{13}} X_1 + \underline{\alpha_{23}} X_2 + e_3$$
$$X_4 = \underline{\alpha_{34}} X_4 + e_4$$
$$X_5 = \underline{\alpha_{35}} X_5 + e_5$$
$$Cov(X_1, X_2) = Cov(X_1, e_3) = Cov(X_1, e_4) = Cov(X_1, e_5) = Cov(X_2, e_3) =$$
$$Cov(X_2, e_4) = Cov(X_2, e_5) = Cov(e_3, e_4) = Cov(e_3, e_5) = 0$$
$$Cov(e_4, e_5) = \underline{\sigma_{45}}$$

As you can see in the first five equations, each of the five exogenous variables are assumed to follow a normal distribution with a mean of zero and a standard deviation (σ_i) whose value is unknown and so must be estimated by the data. Since the exogenous variables are always modelled as normal variates, you don't have to include these first five equations into the call to lavaan. The next three equations describe how each of the endogenous variables are linearly linked to the other variables. Each of the α_{ij} in these equations is a path coefficient and there will always be as many path coefficients as there are arrows (→) in the path diagram. The last ten equations specify if each of the ten possible pairwise covariances between the exogenous

[99] Means can also be modeled but this requires a little bit more work.

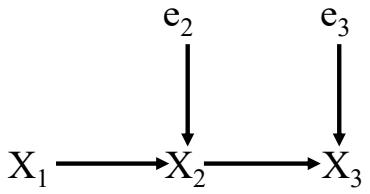
variables (X_1, X_2, e_3, e_4, e_5) are independent or not. You should immediately see why all but the last pair of covariances must be independent: all but the last pair of exogenous variables (e_4, e_5) are unconditionally d-separated given Figure 4.1. Since d-separation implies independence, and since independence of linearly related variables following normal distributions implies a zero covariance, these covariances must be "fixed" to zero. Since there is a double-headed arrow in Figure 4.1, the last covariance, Cov(e_4, e_5), is not fixed to zero; rather, it is "free" to take any value.

Usually, your causal hypothesis will simply state that variable X_i is a direct cause of X_j (thus, $X_i \rightarrow X_j$ in the path diagram) without stating the numerical strength of this causal effect. Your causal hypothesis is predicting that the path coefficient, α_{ij}, will be different from zero but your hypothesis is not sufficiently precise for you to predict the actual value of the path coefficient. In this common situation, α_{ij} is said to be a "free" parameter. If your causal model is sufficiently detailed that you are willing to hypothesise the numerical values of some parameters (path coefficients, variances or covariances) then you can include this information in the model by telling lavaan the value to which this parameter is to be fixed.

Each parameter (a path coefficient, a variance or a covariance) is either "fixed" or "free". "Fixing" a parameter means telling lavaan what this value must be before fitting data to the model and specifying that this value cannot be changed during the fitting of the model via maximum likelihood estimation, which will be described in the next section. "Freeing" a parameter means telling lavaan that it can choose the value for the parameter that best fits the data. I have underlined each of the free parameters in the structural equations listed above to make them obvious to you. Each parameter that is fixed adds one degree of freedom to the inferential test while each parameter that is free subtracts one degree of freedom from the inferential test.

4.4 Deriving the predicted variance and the covariance between each pair of variables in the model using covariance algebra

This step has no equivalent in piecewise SEM. Covariance-based SEM consists of deriving a "model-predicted" covariance matrix, whose elements are functions of the free parameters, and choosing values for these free parameters via maximum likelihood estimation (section 4.4) such that these elements are as close as possible to the empirically estimated covariance matrix obtained from the data. Happily, you do not have to do any of this work because lavaan does it all for you. However, it is important to have at least an intuitive understanding of how this model-predicted covariance is obtained so that you can understand how the model-predicted and empirical covariance matrices allow one to test the causal hypothesis. I will use the very simply path diagram in Figure 4.2 so that we can avoid matrix algebra.



$$X_1 = N(\mu = 0, \underline{\sigma_1})$$
$$e_2 = N(\mu = 0, \underline{\sigma_2})$$
$$e_3 = N(\mu = 0, \underline{\sigma_3})$$
$$X_2 = \underline{\alpha_{12}} X_1 + e_2$$
$$X_3 = \underline{\alpha_{23}} X_2 + e_3$$
$$Cov(X_1, e_2) = Cov(X_1, e_3) = Cov(e_2, e_3) = 0$$

Figure 4.2. A simple path diagram and its structural equations, including the free parameters (underlined).

First, we must convert our structural equations into their "reduced" form. This means replacing each endogenous variable on the right-hand side of each structural equation by a function of exogenous variables. The equation $X_2=\alpha_{12}X_1 + e_2$ is already in its reduced form because all the variables (i.e. causes) on the right-hand side (X_1 and e_2) are exogenous. The equation $X_3=\alpha_{23}X_2 + e_3$ is not in its reduced form because X_2 is endogenous. However, since X_2 can also be written as $X_2=\alpha_{12}X_1 + e_2$, we can replace X_2 in this equation to give $X_3=\alpha_{23}(\alpha_{12}X_1 + e_2) + e_3 = \alpha_{23}\alpha_{12}X_1 + \alpha_{23}e_2 + e_3$. Now, the equation for X_3 is in its reduced form because it is a function only of exogenous variables (X_1, e_2, and e_3). Every structural equation can be expressed in its reduced form. You might also have noticed that these reduced form equations use the rules for path tracing of indirect effects that Sewall Wright developed (Chapter 3).

There are some simple algebraic rules when adding, subtracting, multiplying of dividing covariances. Let's start with the definition of a population covariance that you learned in your first statistics course.

$$Cov(X_i, X_j) = \sigma_{ij} = \sum_{i=1}^{\infty} \frac{(X_i - \mu_i)(X_j - \mu_j)}{n} = E\left[(X_i - \mu_i)(X_j - \mu_j)\right]$$

The notation $E[\bullet]$ is called the expectation operator and is simply a more compact way of writing the equation for an arithmetic mean. Since, by default, we are dealing with variables that have already been centred about their means (i.e. m=0), this reduces to:

$$Cov(X_i, X_j) = \sigma_{ij} = \sum_{i=1}^{\infty} \frac{(X_i - 0)(X_j - 0)}{n} = E\left[(X_i)(X_j)\right] = E\left[X_i X_j\right]$$

A variance is simply the covariance of a variable with itself. Using variables that have already been centred:

$$Cov(X_i, X_i) = \sigma_{ii} = \sum_{i=1}^{\infty} \frac{(X_i)(X_i)}{n} = E\left[X_i X_i\right] = E\left[X_i^2\right]$$

If K_1 and K_2 are constants and X_1, X_2 and X_3 are three normally distributed variables, then it is easy to prove the following four rules:

(1) $Cov(K_1, X_1)=0$

(2) $Cov(K_1X1, X_2)=K_1Cov(X_1, X_2)$

(3) $Cov(K_1X_1,K_2X_2)=K_1K_2Cov(X_1,X_2)$

(4) $Cov(X_1+X_2, X_3)=Cov(X_1, X_3)+Cov(X_2, X_3)$

Using these rules and working with the reduced form of the structural equations, we can write down the predicted covariances between each pair of observed variables (X_1, X_2, X_3) in Figure 4.2. Let's start with the covariance between X_1 and X_2. Since $X_2=\alpha_{12}X_1+e_2$, and applying rule (2):

$$Cov(X_1X_2) = E\left[X_1, X_2\right] = E\left[X_1, \alpha_{12}X_1 + e_2\right] = \alpha_{12}E\left[X_1, X_1\right] + E\left[X_1, e_2\right] .$$

Now, we know from the structural equations (Figure 4.2) that the covariance between X_1 and e_2 is zero (i.e. $E[X_1,e_2]=0$) and, of course, we also know this directly from d-separation. We also know that the covariance of a variable (here, X_1) and itself is its variance. Together, this gives us the equation to predict the covariance between X_1 and X_2, given the path diagram:
$Cov(X_1X_2) = \alpha_{12}\sigma_1^2$.

Now let's get the equation predicting the covariance between X_1 and X_3. We know that $X_3=a_{23}a_{12}X_1 + a_{23}e_2 + e_3$. Applying our rules, we get:
$Cov(X_1, X_3) = Cov(X_1, \alpha_{23}\alpha_{12}X_1 + \alpha_{23}e_2 + e_3) = \alpha_{23}\alpha_{12}E[X_1, X_1] + \alpha_{23}E[X_1, e_2] + E[X_1, e_3]$.
Again, we know from the structural equations that the covariance between X_1 and each of e_2 and e_3 is zero, and so we get the equation predicting the covariance between X_1 and X_3:
$Cov(X_1, X_3) = \alpha_{23}\alpha_{12}\sigma_1^2$. If we had included a double-headed arrow between X_1 and e_3 (a "free" covariance) then $E[X_1,e_3]$ would not be zero and the predicted covariance between X_1 and X_3 would have been $Cov(X_1, X_3) = \alpha_{23}\alpha_{12}\sigma_1^2 + \sigma_{13}$.

What about the model-predicted variance of X_2 (an endogenous variable)? Using the same logic:
$\sigma_2^2 = E[X_2, X_2] = E[\alpha_{12}X_1 + e_2, \alpha_{12}X_1 + e_2] = \alpha_{12}^2E[X_1, X_1] + 2\alpha_{12}E[X_1, e_2] + E[e_2, e_2] = \alpha_{12}^2\sigma_1^2 + \sigma_{e_2}^2$

If we do this for each of the six nonredundant elements of the model-predicted covariance matrix for the observed variables (X_1, X_2, X_3), and we remember that the elements of a covariance matrix are symmetrical about the diagonal (i.e. $\sigma_{ij}^2 = \sigma_{ji}^2$) then we get the model-predicted covariance matrix.

$$\begin{bmatrix} \sigma_1^2 & a_{12}\sigma_1^2 & a_{23}a_{12}\sigma_1^2 \\ a_{12}\sigma_1^2 & a_{12}^2\sigma_1^2 + \sigma_{e_2}^2 & a_{23}\sigma_{e_2}^2 \\ a_{23}a_{12}\sigma_1^2 & a_{23}\sigma_{e_2}^2 & a_{12}^2\sigma_1^2 + a_{23}^2\sigma_{e_2}^2 + \sigma_{e_2}^2 \end{bmatrix}$$

Each element of this matrix is a function of fixed or free parameters involving either path coefficients or the variances or covariances of exogenous variables. Once we obtain estimates for each of the free parameters in this model-predicted covariance matrix (explained in section 4.4), then we can compare them to the actual variances and covariances between the observed variables that we measure in our data. After that, we only have to set up our null hypothesis that

the model-predicted and the observed covariance matrices are the same except for random sampling variation and then calculate the probability of this null hypothesis occurring (explained in section 4.5).

Given that I have chosen the simplest possible path model (Figure 4.2) as an example, this must seem like a lot of work. Don't worry, most SEM programs, including the lavaan package of R, do all this work for you. The important point at this stage is that you have an intuitive understanding of why we can express the covariances between each pair of variables as a function of path coefficients plus variances and covariances of exogenous variables. For those who are used to working with matrix algebra, Box 4.1 gives a more formal derivation of the model-predicted covariance matrix based on the Bentler-Weeks model (Bentler 1995).

If we go back to the analogy of correlations being the shadows that are cast by causal processes, then the model-predicted covariance matrix is a description of the "shape" (the topology of the path diagram), but not the "size" (the numerical values of the free parameters), of the shadow that is cast by the hypothesised causal process shown in Figure 4.2. Imagine that we were describing the shadow cast by a solid square whose size was unknown to us (i.e. the length of whose sides are *free* parameters). We would describe the shadow as having four equal sides (the first constraint) of unknown length with four sides that meet in such a way that they make four corners having 90-degree angles (the second constraint). The general *shape* of the shadow is fixed (a square) by its constraints in the same way that the topology of the causal process is fixed by the path diagram, but the numerical *values* (the lengths of the sides) are free parameters whose values must be estimated such that they agree as closely as possible to the real shadow.

Box 4.1

The Bentler-Weeks Model

Let the endogenous variables in the model be written in a column vector called η and let the exogenous variables (including the error variables) be written in a column vector called ε. Let the coefficients of the effects of endogenous causes to endogenous effects be a matrix called $\boldsymbol{\beta}$ (rows are endogenous effects and columns are endogenous causes) and let the coefficients of the effects of exogenous causes to endogenous effects be a matrix called $\boldsymbol{\gamma}$ (rows are dependent effects and columns are independent causes). Then the system of structural equations can be written as:

$\eta = \boldsymbol{\beta}\eta + \boldsymbol{\gamma}\varepsilon.$

For instance, the path model in Figure 4.1 would be written:

$$\begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ a_{34} & 0 & 0 \\ a_{35} & 0 & 0 \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} a_{13} & a_{23} & b_3 & 0 & 0 \\ 0 & 0 & 0 & b_4 & 0 \\ 0 & 0 & 0 & 0 & b_5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \end{bmatrix}.$$

In reduced form the system of structural equation is: $\eta = (\mathbf{I} - \boldsymbol{\beta})^{-1}\boldsymbol{\gamma}\varepsilon$.

Predicted covariances between exogenous variables: $E[\varepsilon\varepsilon'] = \boldsymbol{\varsigma}$.

Predicted covariances between endogenous and exogenous variables:

$E[\eta\varepsilon'] = (\mathbf{I} - \beta)^{-1}\gamma\varsigma$.

Predicted covariances between endogenous variables:

$$E[\eta\eta'] = (\mathbf{I} - \boldsymbol{\beta})^{-1}\boldsymbol{\gamma}\boldsymbol{\varsigma}\boldsymbol{\gamma}'(\mathbf{I} - \boldsymbol{\beta})^{-1'}.$$

4.5 Estimating the free parameters using maximum likelihood

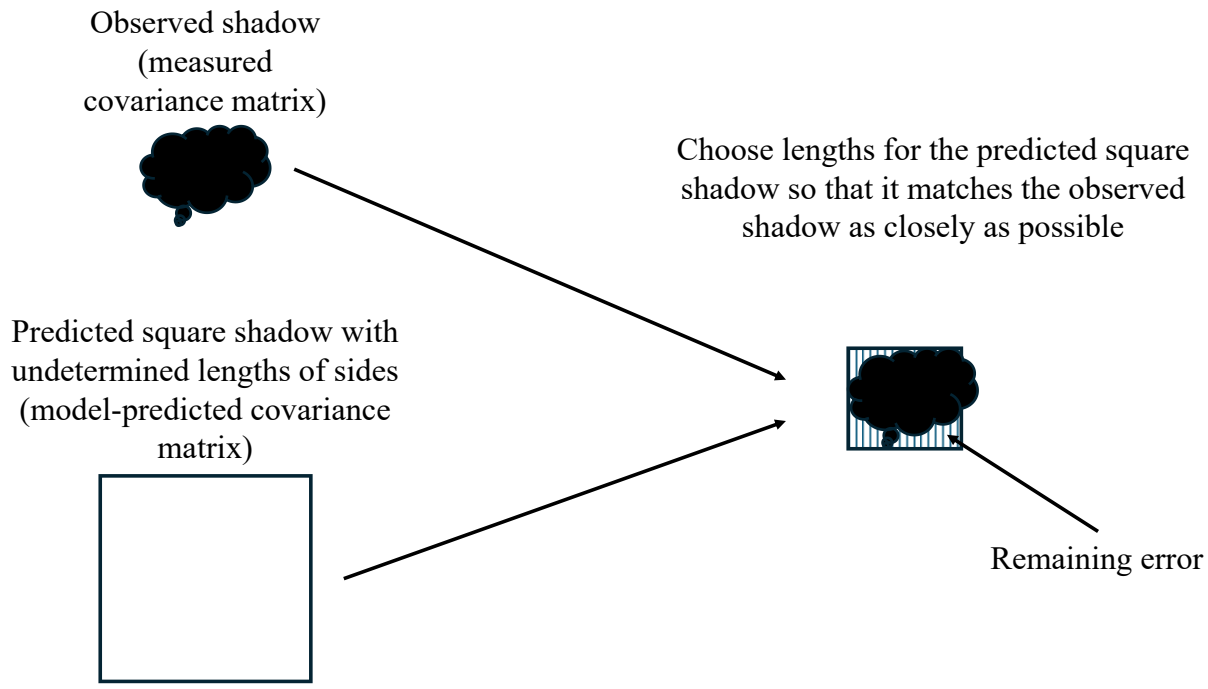


Figure 4.3. Analogy of estimating the values of the free parameters of a model-predicted covariance matrix so that is matches the observed covariance matrix as closely as possible.

The hypothesised object was the solid square and from this we have predicted the shape of the shadow that it would cast (Figure 4.3). Is our hypothesis correct? To decide, we superimpose our hypothesised square shadow (the model-predicted covariance matrix) on top of the actual shadow (the empirically measured covariance matrix). We make an initial guess for the length of the sides (the free parameter) of our hypothesised square matrix and then proceed to sequentially increase or decrease the length of its sides until our hypothesised square is as close to the observed shadow as possible while respecting the constraints (i.e. it must remain a square and so we can't change the angles of the sides or make some sides longer than others). Once we have superimposed the chosen length of the sizes to best match the shadow, we then measure the remaining lack of fit.

Intuitively, you can see that if the real three-dimensional object (the causal process) that cast the shadow (the observed covariance matrix) was square in profile then the predicted square (the model-predicted covariance matrix) and the real shadow of the object (the observed covariance matrix) will closely align. If, however, the real three-dimensional object that cast the shadow was not square in profile then the predicted square and the real shadow of the object will not

closely align, and the remaining lack of fit would be more than expected given measurement error. Section 4.5 will explain how we measure this remaining lack of fit. This is the same basic logic used to fit and test a structural equation model. We first choose values for the free parameters in our model-predicted covariance matrix that make it as numerically close as possible to the observed covariance matrix while respecting the constraints applied to the model-predicted covariance matrix. Then, we see how much difference remains between the observed and predicted covariance matrices.

The general strategy for obtaining the best values for the free parameters is easy enough to grasp: choose values of the free parameters that make the numerical values of the predicted covariance matrix as close as possible to the actual covariances measured in the data. This is done using a method called *maximum likelihood estimation* (Fisher 1922). Since I will also use the method of maximum likelihood elsewhere in this book, I will take some time to give you an intuitive explanation of how it works. In essence, the numerical algorithm used to maximise the likelihood is a bit like playing the child's game of 20 questions (Is it alive? Is it a mammal? Is it a carnivore? Does it live in Africa?) until you arrive at the answer. Another good analogy for maximising a likelihood function might be a person who is blindfolded and finds herself in a landscape with various hills and valleys. Her job is to walk to the top of the highest hill without peeking. Since she can't see the landscape, but she can feel if she is walking uphill or downhill, she begins by taking an initial step in a direction based on her best guess. If she sees that she has moved uphill, then she continues in the same direction with a second step in the same direction. If not, she changes direction and tries again. She continues with this process until she finds herself at a position on the landscape in which every possible change in direction results in movement downhill. She therefore knows that she has reached the top of a hill. Unfortunately, if the landscape is very complicated, she may have found herself on top of a small mound rather than at the top of the highest hill. The only way to find out would be to start over at a different initial position and see if she again ends up in the same place. This is essentially how maximum likelihood estimation works.

Maximum likelihood estimation can be done using any parametric probability distribution or density. In fact, we have been using this method repeatedly in this book already whenever we

have fit generalized linear models[100]. Since covariance-based SEM is based on a multivariate normal probability density, I will use this parametric probability density. To make my job even easier, I will use a univariate normal probability density to illustrate the basic idea.

Common to every probability function is the idea that our observations (X_i) are random values; that is, we can't know or predict their values until we observe them. Common to every *parametric* probability function is the idea that the relative frequency of these random values in the full statistical population can be modelled with a mathematical function. This mathematical function has certain fixed constants, or parameters. Here is the univariate normal probability density function; notice that the notation "$p(X_i|\mu,\sigma)$" means the probability ("p") of observing a random (unknown until observed) value of X given ("|") a specific normal probability density function having a fixed mean of μ and a fixed standard deviation of σ. The parameters are μ and σ (they are "givens" and so are not random) and the unknown random variable is X.

$$p\left(X_i \mid \mu,\sigma\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(X_i-\mu)^2}{2\sigma}} .$$

However, what happens after we have observed and recorded our observed value of X_i? For instance, what if we have taken our measurement and now know that $X_i = 1.96$? The right-hand side of this equation still describes the relative frequency of the random values in the full statistical population, but the left-hand side no longer makes sense since X_i is no longer a random variable; rather, it is fixed at 1.96. After all, if you close the notebook in which you recorded the value of 1.96 and then later open it up, the number written down will not change unpredictably! To account for this difference, we now call the right-hand side a "likelihood" function ($\mathcal{L}$) rather than a "probability" density function (p), and we switch the fixed and random parts around:

$$\mathcal{L}\left(\mu,\sigma \mid X_i\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(X_i-\mu)^2}{2\sigma}} .$$

In this likelihood function, we are now viewing the parameters (μ, σ) as unknowns and the data (X_i) as fixed, i.e. a known value. We are saying that there exists an infinite number of different

[100] Or (generalized) nonlinear models, or (generalized) mixed models, or (generalized) (mixed) additive models. In fact, maximum likelihood estimation is used as a numerical algorithm in the majority of modern statistics.

normal density functions, each having a different value of μ and σ, and we don't know from which of these different normal density functions our known X_i was drawn. We are imagining that we are randomly sampling from this infinite population of normal distributions. Our task is to choose which of these different normal density functions is most likely to have generated X_i = 1.96.

In practice, we usually have a whole data set of values for X; one for each independent[101] observation. Let's say that we have N = 10 independent observations: X={0.0, -0.2, -1.4, -0.6, 0.3, 0.4, -1.2, -0.4, -1.6, -0.3}. Since the probability of independent events is the product of their individual probabilities[102], we can write our likelihood function for the full data set as:

$$\mathcal{L}(\mu,\sigma \mid X_i) = \prod_{i=1}^{N=10} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(X_i-\mu)^2}{2\sigma}} .$$

It is numerically much more difficult to find a solution to a function that is a product of its terms than to find a solution to a function that is a sum of its terms. We therefore take the logarithm of the likelihood function (the log-likelihood):

$$\mathcal{LL}(\mu,\sigma \mid X_i) = -N \ln\left(\sqrt{2\pi}\sigma\right) - \sum_{i=1}^{N=10} \frac{(X_i-\mu)^2}{2\sigma} .$$

In order to maximize the log-likelihood of our data, assuming a normal distribution, we begin with an initial guess for the things we want to know (μ, σ) and calculate the log-likelihood. Let's start with an initial guess of μ=1 and σ=1. If we enter our ten observed values of X into the log-likelihood function, we get a value of -22.619. What happens if we reduce the value of μ a little bit from 1 to μ=0.9? The log-likelihood function is now -21.169. We have increased the log-likelihood value (we have moved uphill), so we know that 0.9 is a better guess for the value of μ than was our initial guess of 1. Figure 4.4 shows how the log-likelihood value changes for different values of μ between -2 and 2 given our ten values of X. You can see that the value of μ that maximizes the log-likelihood value is -0.5. This is the same as the mean of the X values, which shows that the formula that you know for calculating an arithmetic mean is also a

[101] The assumption of mutually independent observations is essential here!
[102] $p(X_i,X_j)=p(X_i)p(X_j)$ if X_i and X_j are independent of each other.

"maximum likelihood" estimate. Some readers might notice that I cheated a bit with this example because I changed the values of μ but kept the value of σ at 1. In fact, one would change both at the same time in order to find the combination of μ and σ that maximises the log-likelihood.

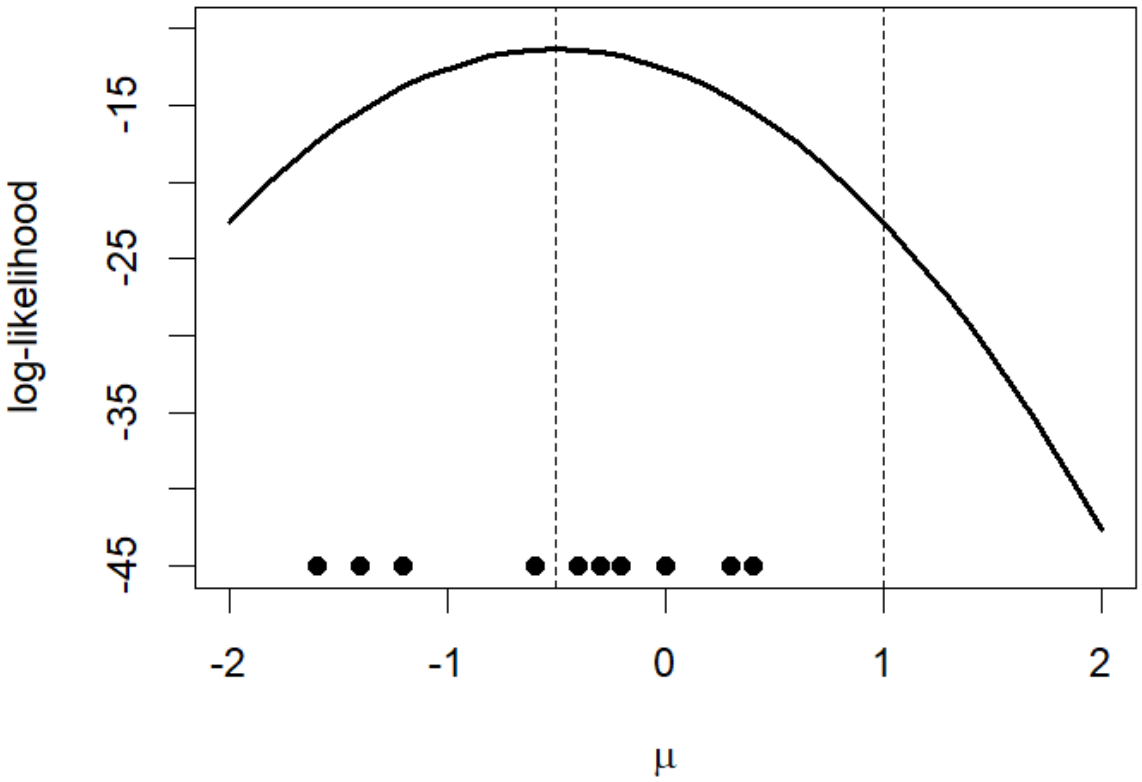


Figure 4.4. The relationship between the log-likelihood value, given a set of 10 observations (the black points) and a normal density function with different values of μ and a fixed value of σ=1. The value of μ = 1 was our initial guess and μ = -0.5 is the value that maximizes the log-likelihood.

In covariance-based SEM, we always have more than one observed variable. Therefore, we must use a multivariate normal (or Gaussian) probability density function, and its likelihood version, rather than the univariate normal function that I gave above. Here is the multivariate likelihood

function, where $\mathbf{X}$ is a matrix holding the observed values of the observed variables in the structural equations, $\boldsymbol{\mu}$ is a vector of the means of each of the observed variables, $\boldsymbol{\Sigma}$ is the model-predicted covariance matrix (section 4.3), $|\boldsymbol{\Sigma}|$ is the determinant of $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}^{-1}$ is the inverse of $\boldsymbol{\Sigma}$, and $(\mathbf{X}-\boldsymbol{\mu})^{T}$ is the transpose of the vector $\mathbf{X}-\boldsymbol{\mu}$:

$$\mathcal{L}(\boldsymbol{\mu}, \boldsymbol{\Sigma} \mid \mathbf{X}) = \frac{1}{(2\pi)^{n/2}\sqrt{|\boldsymbol{\Sigma}|}} e^{-0.5(\mathbf{X}-\boldsymbol{\mu})^{T}\boldsymbol{\Sigma}^{-1}(\mathbf{X}-\boldsymbol{\mu})}$$

.

Those of you who are familiar with matrices will notice that this is simply the generalization of the univariate normal likelihood function. Since covariance-based SEM usually works with observed variables that are centred about their means, we can usually remove the vector of means ($\boldsymbol{\mu}$), meaning that only the model-predicted covariance matrix ($\boldsymbol{\Sigma}$) remains. Taking the logarithm of this function gives the log-likelihood.

Each element in the model-predicted covariance matrix ($\boldsymbol{\Sigma}$) is a function predicting the value of a free parameter in the structural equations (a path coefficient, the variance or covariance of an exogenous variable, or the covariance between a pair of exogenous variables), as explained in section 4.3. To maximise the log-likelihood given the data ($\mathbf{X}$), i.e. to choose the values of the free parameters in $\boldsymbol{\Sigma}$ that best match the data, we do exactly as before. We start with initial values (guesses) for each of the free parameters, calculate the log-likelihood of the data given these initial values, and then incrementally change our values of these free parameters in such a way that each successive step results in a larger log-likelihood value than in the previous step. When we reach a set of values for these free parameters for which any possible change will decrease the log-likelihood, then we stop[103]. The values of the free parameters when we stop are called the "maximum likelihood estimates" for the free parameters.

What can go wrong during maximum likelihood estimation? One potential problem is that, like our woman trying to reach the highest hill while blindfolded and being trapped on the top of a small hill, the maximum likelihood procedure can get trapped in a local maximum and never reach the global maximum. In my toy example involving finding the maximum likelihood

[103] In practice, we must choose some very small value such that, if the change in log-likelihood between two successive steps is less than this value, then we stop. This is called the "convergence tolerance". The default value in lavaan is 10^{-8} but this value can be changed.

estimate for the mean of a single variable, this can never happen because there is only one "hill" to climb. You will always eventually reach the top of the hill no matter where you start (your initial value) and the only consequence of choosing an initial value that is very far away from the maximum likelihood value is that you will have to do more iterations before you reach the top. When dealing with more than one free parameter, the maximum likelihood "landscape" can contain more than one hill and, if you choose the wrong initial values for your free parameters, it is possible that your "maximum" likelihood value will only be the maximum of the nearest hill rather than the maximum of the highest hill. This is because, once you reach a local maximum, any subsequent change in the values of the free parameters will result in a smaller log-likelihood and so the process stops. This is why the woman in our analogy was blindfolded; the maximum likelihood process can't see the entire likelihood landscape, it can only "feel" the likelihood surface in its immediate vicinity. The only way to make sure that you are dealing with a global maximum is by trying different starting values and choosing the result with the highest log-likelihood. In practice, this problem almost never occurs with structural equations that do not include explicit latent variables (the topic of this chapter) but can occasionally occur when models include explicit latent variables.

Another potential problem that can arise with maximum likelihood estimation is that you might run into a "convergence problem". Remember that the process involves successive iterations in which each step yields a larger log-likelihood value until the increase is smaller than the convergence tolerance value (by default 10^{-8}). In order to avoid the algorithm within lavaan running forever, this iterative process is inside a loop that has a maximum number of iterations, after which it gives up, stops, and outputs a warning. The default number of iterations can be changed, but a better way is to use the final estimates of the free parameters that are output when the default number of iterations have been reached and input them again as new starting values.

4.6 Calculating the probability of having observed the measured minimum difference between the observed and predicted covariances, assuming that the observed and predicted covariances are identical except for random sampling variation

At the end of step 4, we have a model-predicted covariance matrix with numerical values that best fits the data while respecting the hypothesized causal generating process captured by the structural equations. Because we have maximum likelihood estimates for all the free parameters, we have parameterized our structural equations. Now, we must compare the values in the model-predicted covariance matrix (**Σ**) that maximize the likelihood to the actual measured covariance matrix (**S**) based on our observed data. If we let p equal the number of free path coefficients, and q equal the number of free variances and covariances of the exogenous variables in the structural equations, then the function that measures the difference between **Σ** and **S** is:

$$F_{ML} = \ln\left(\left|\mathbf{\Sigma}\right|\right) + trace\left(\mathbf{S}\mathbf{\Sigma}^{-1}\right) - \ln\left(\left|\mathbf{S}\right|\right) - \left(p+q\right).$$

This function has an important property[104]. If the structural equations truly model the causal generating process in Nature, then the only remaining differences between the values in **Σ** and **S** will be due to random sampling variation. If this is the case, then $(N)F_{ML}$ will follow a Chi-Squared sampling distribution, where N is the number of mutually independent observations in your data set. $(N)F_{ML}$ is called the "maximum likelihood chi-squared statistic". This allows us to calculate the probability of observing our data assuming our causal hypothesis (i.e., our hypothesized structural equations). In Chapter 3 I said that one probable reason why biologists did not accept Wright's method of path analysis when he first proposed it was that his original method could derive the logical consequences of a causal model but could not test it. The method described above, developed by Jöreskog in 1970 (Jöreskog 1970), was the first to solve this important shortcoming of Wright's original "path analysis". Equally importantly, as will be explained in Chapter 7, this method allows us to include explicit latent variables into our structural equations.

In order to use a Chi-Squared distribution, one needs to know the degrees of freedom. In covariance-based SEM, the observed and predicted "data" that we are comparing are not the N observations in our data set but rather the observed and predicted elements in the covariance matrix. Given V observed variables, there are always V^2 elements in a covariance matrix, but the

[104] Another important property of F_{ML} is that the values of the free parameters that minimizes it also maximizes the log-likelihood.

lower triangle of a covariance matrix is the mirror image of the upper triangle. The number of unique variances and covariances in a covariance matrix is V(V+1)/2. Each time we use the data to estimate the value of a free parameter, we lose one degree of freedom. Therefore, the number of degrees of freedom in our Chi-Squared distribution is $df = V\left(V+1\right)/2-(p+q)$ where (p+q) are the number of free parameters that had to be estimated[105]. Putting all this together, the following test statistic, the *maximum likelihood chi-squared statistic*, will asymptotically follow a central Chi-Squared distribution with the stated degrees of freedom:

$$NF_{ML} \xrightarrow{N\to\infty} \chi^2_{\frac{V(V+1)}{2}-(p+q)}.$$

As explained in Chapter 3, "asymptotically" means that the sample size goes to infinity, and this means that you need a certain minimum sample size. The minimum sample size depends on a number of properties of the data, and there are ways of dealing with small sample sizes, as described later in this chapter.

In most of the statistical tests used by biologists, the biologically interesting hypothesis is the alternative hypothesis. The null hypothesis functions as a strawman ("no effect") that is erected only to see if we have sufficiently strong evidence to knock it down. This is useful because it forces us to have strong evidence (evidence beyond reasonable doubt) before we can accept the biologically interesting alternative hypothesis. In SEM on the other hand, models are constructed based on biological arguments in such a way as to reflect what we hypothesise to be correct. In other words, our model and the resulting predicted covariance matrix embodies what we view to be biologically interesting. The null hypothesis, not the alternative, is therefore the biologically interesting hypothesis. Just as when comparing Fisher's C statistic to a Chi-Squared distribution in piecewise SEM, a probability that is below the chosen significance level means that the predicted model is wrong and should be rejected (i.e. the null hypothesis should be rejected). Although the flipping of the null and alternative hypotheses might seem strange, it is the same logic[106] as testing the null hypothesis that the slope of a simple linear regression equals

[105] p is the number of free path coefficients and q is the number of free variances and covariances of the exogenous variables.

[106] For instance, you might predict that, since surface area (where heat is lost) and mass (where heat is generated) scales as 2/3, the scaling of body size and basal metabolic rate of homeotherms should scale as 0.67. You could test

(say) 0.67. Notice that we are reversing the burden of proof: we are requiring strong evidence, evidence beyond reasonable doubt, before we are willing to reject our preferred hypothesis. This leads naturally to the temptation to conclude that the predicted model is correct simply because we have not obtained strong evidence to the contrary! In fact, all that we can conclude is that we have no good evidence to reject our model, and that the data are consistent with it. The degree to which we have good evidence in favour of our model will depend on how well we can exclude other models that are also consistent with the data. This leads naturally to the subjects of statistical power, AIC statistics and equivalent models (Chapter 5).

4.7 Using lavaan to fit path models

I will illustrate the basic properties of the lavaan package using a simulated data set. I will simulate 500 multivariate "observations" from the path diagram shown in Figure 4.5a and fit these data using both a path diagram representing the correct generating process (Figure 4.4a) and a path diagram representing an incorrect generating process (Figure 4.5b). These data will agree with the assumptions of covariance-based SEM (multivariate normality, linearity and mutually independent observations). Each path coefficient equals 0.5 and each of the "observed" variables follows a standard normal probability density. The "L" variable is unobserved (latent) and is used to generate the free correlation of 0.25 between X_4 and X_5 (explained more fully in Chapter 5). Here is the R code to generate the data:

```
set.seed(10)
N<-500
X1<-rnorm(N)
X2<-rnorm(N)
L<-rnorm(N)
X3<-0.5*X1+0.5*X2+rnorm(N,0,sqrt(1-2*0.5^2))
X4<-0.5*X3+0.5*L+rnorm(N,0,sqrt(1-2*0.5^2))
X5<-0.5*X3+0.5*L+rnorm(N,0,sqrt(1-2*0.5^2))
fig4.4.dat<-data.frame(X1,X2,X3,X4,X5)
```

this with a null hypothesis that the slope of a log-log regression of metabolic rate ~ body size. Failure to reject it would be evidence in favour of your hypothesis.

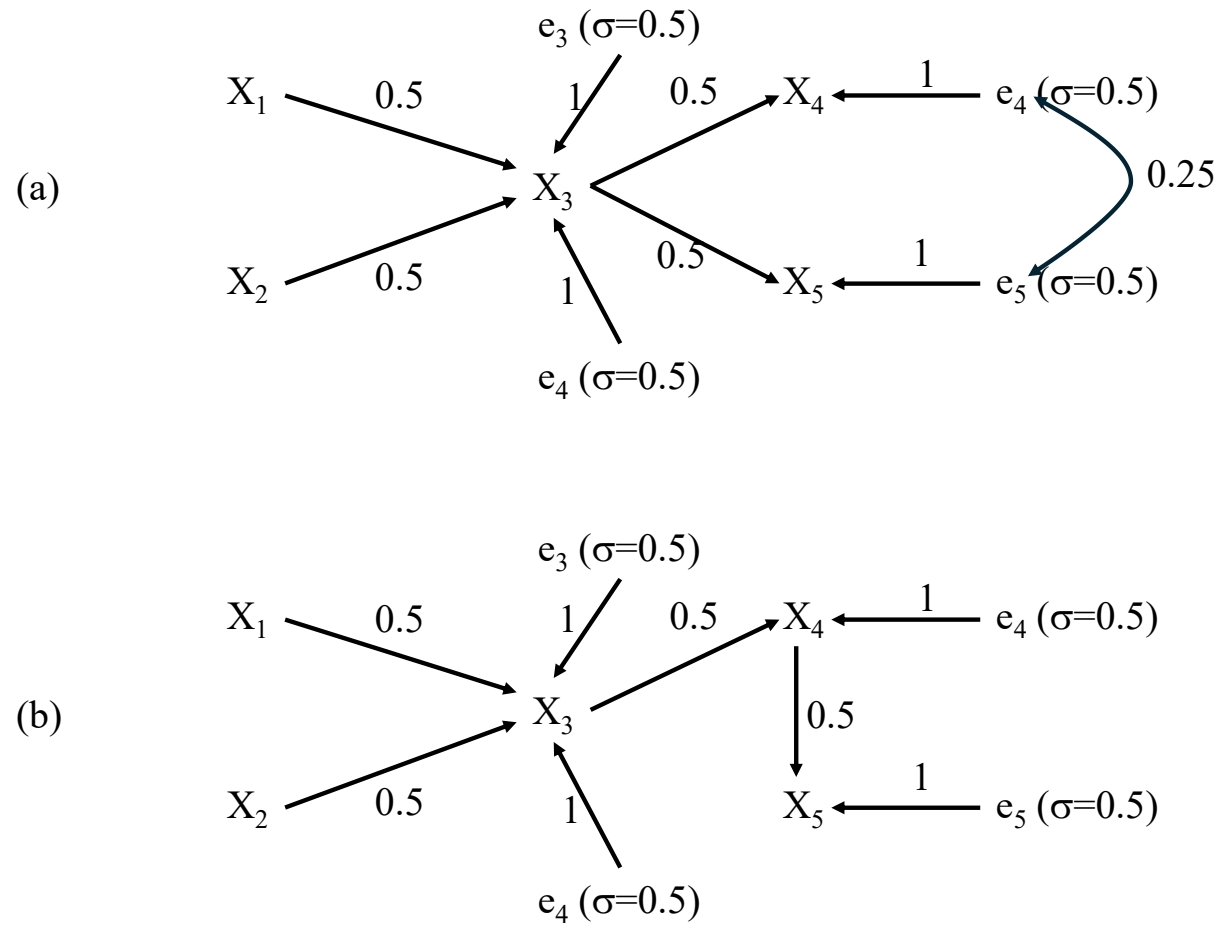


Figure 4.5. The data frame Fig4.5a was produced by the generating process shown in the path diagram (a). The path diagram (b) shows an incorrect causal hypothesis.

I will use the `sem()` function of lavaan throughout this book. There are three basic steps involved in covariance-based SEM in lavaan. The first step is to translate the causal model into R code and save it as an R object. The second step is to fit the data to the causal model using the `sem()` function. The third step is to extract and summarize the output using the `summary()` function. Here is the most basic code for the model object based on the structural equations from the path diagram in Figure 4.5a. This code is enclosed between quotes and saved as `true.model`; I will explain later why this basic code is missing some important components:

```
true.model<-"
X3~X1+X2
```

```
X4~X3
X5~X3
X4~~X5
"
```

It is essential that the names of the variables in this model object agree exactly with the names in the data frame. The single tilde symbol (~) in lavaan should be read as "is caused by". Thus, the line X3~X1+X2 means "X3 is caused by X1 and X2". The double tilde symbol (~~) in lavaan means "covariance" or "variance" and so the line X4~~X5 means that "X_4 covaries with X_5", i.e. that there is a free covariance between X4 and X5.

The `sem()` function of lavaan, which we will use to fit data to a model object, has a number of non-intuitive defaults that convert a model object into the full set of structural equations. For instance, one must specify which covariances between the exogenous variables are fixed at zero and which should be freely estimated. Unless you tell the `sem()` function otherwise, it will allow free covariances between each pair of observed exogenous variables (X_1 and X_2 in Figure 4.5a) and between the error variables of each pair of terminal endogenous variables (X_4 and X_5 in Figure 4.5a). The path diagram in Figure 4.5a clearly states that X_1 and X_2 are independent (they are d-separated) and so we don't want a free covariance between them as the default setting would create. Also, by default, `sem()` doesn't actually estimate the variances of the observed exogenous variables (X_1 and X_2). Instead, it simply uses the measured variances of these exogenous observed variables. There is almost always more than one way to do something in R but the easiest and most obvious way of making sure that `sem()` does what you want rather than what it wants (via defaults) is to be explicit in the coding of the model object. We tell[107] `sem()` to estimate the variances of X_1 and X_2 by adding the lines X1~~X1 and X2~~X2. Why does X1~~X1 mean "freely estimate the variance of X_1" when I earlier used the double tilde symbol to refer to a covariance? Because a variance is simply the covariance of a variable with itself! Finally, we tell `sem()` to fix the covariance between X_1 and X_2 to zero by adding the line X1~~0*X2. Notice that the line X4~~X5 means to freely estimate the covariance between X_4 and X_5 while the line X1~~0*X2 means to *not* freely estimate it but, rather, to fix it to zero. This is a general rule of syntax in lavaan; adding "K*" before a variable name on the righthand side,

[107] You can do the same thing by adding the argument `fixed.x=FALSE` in the `sem()` function.

where K is a number, means to fix the associated parameter (a path coefficient or variance/covariance of an exogenous variable) to the specified number.

It is important to explicitly model the variances/covariances of the exogenous observed variables and the covariances of the terminal observed variables (i.e. their latent exogenous errors) rather than allowing lavaan to do this for you. I have reviewed several manuscripts in which the path diagram does not correspond to the reported degrees of freedom because the authors used the defaults of lavaan without noticing that these default choices did not correspond to what they wanted to model!

Here is the new model object:

```
true.model<-"
X3~X1+X2
X4~X3
X5~X3
X1~~X1
X2~~X2
X1~~0*X2
X4~~X5
"
```

4.8 Fitting the model object to data and outputting the result

Now that we have a model object that encodes the causal claims of our path diagram, we fit the data to this model object using the `sem()` function and use `summary()` to obtain the results:

```
library(lavaan)
fit.true.model<-sem(model=true.model, data=fig4.5.dat)
summary(fit.true.model)
```

Here are the first five line of the summary output:

```
lavaan 0.6.17 ended normally after 12 iterations

  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        10

  Number of observations                           500
```

The first line of the summary output is important because it tells us if we had any convergence problems while maximizing the log likelihood. In this case, it tells us that the numerical algorithm required 12 iterations before converged normally at the maximum likelihood solution. If you don't see the "ended normally" message in this first line, then you should stop because you have run into a problem with the maximization of the log likelihood. The second line tells us that the estimation method (the default) is maximum likelihood ("ML"). The third line tells us that the numerical method used to maximize the log likelihood is "NLMINB"; this is the default algorithm and is called "NonLinear MINimization using Bounds", a type of quasi-Newton method that allows for fixing boundaries on the search for the maximum likelihood parameter estimates. It is unlikely that you will need to change[108] this default setting. The fourth line tells us that there were 10 free parameters ("model parameters") estimated. You can count these directly from Figure 4.5a: four path coefficients, two variances of exogenous observed variables (X_1 and X_2), three residual variances and one covariance (between X_4 and X_5). The fifth line reports the number of complete lines of data that were used. If you have missing values (NA) in any line of the input data for any of the variables in the model object, then the line is ignored.

```
Here are the next four lines of the summary output:
Model Test User Model:

  Test statistic                                 4.771
  Degrees of freedom                                 5
  P-value (Chi-square)                           0.445
```

These lines report the inferential Chi-Squared test. "Test statistic" gives the measured fit between the observed and model-predicted covariance matrix; since we are using maximum likelihood, the test statistic is the maximum likelihood chi-squared test statistic (4.771). The next line gives the residual degrees of freedom (5). It is always good practice to quickly compare these reported degrees of freedom against the equation that I gave earlier in this chapter. We have V=5 observed variables. We have p=4 free path coefficients. We have q=6 free variances and covariances (2 free observed exogenous variances, 3 latent exogenous error variances and 1 free covariance). Putting these values into our equation gives

[108] You can choose another method using the `optim.method=` argument within the `sem()` function.

$df = V(V+1)/2 - (p+q) = 5(6)/2 - (4+6) = 5$. If your number doesn't agree with the reported degrees of freedom, then lavaan has probably estimated some free parameter that you didn't want without telling you. Thus, we are comparing a chi-squared statistic of 4.771 against a Chi-Squared distribution having 5 degrees of freedom. The resulting null probability (P-value) is 0.445. In other words, we would expect at least this level of lack of fit between the model (the model-predicted covariance matrix) and the data (the observed covariance matrix) 44.5% of the time when the lack of fit is due purely to random sampling variation. Therefore, if our chosen significance level was α=0.05, then we cannot reject our null hypothesis that the data were generated by the causal process modelled by the structural equations. Of course, we already knew this because we generated our data using these structural equations. If the reported null probability was lower than our chosen significance level then we would have to conclude that the measured lack of fit (the maximum likelihood chi-squared statistic) is too large, and therefore too unlikely, to have been caused purely by random sampling variation; there is some error in our model and so we have to reject it. If we reject our model, then we should not trust any of the remaining information in the summary output. In our case, we have not rejected our model and so we can proceed.

The rest of the basic summary output reports the maximum likelihood estimates of the free parameters ("Estimate"). It also gives the asymptotic[109] standard errors ("Std.Err") of these maximum likelihood estimates, a z-score comparing the estimate to a null hypothesis of zero ("z-value") and the null probability of the z-score ("P(>|z|)"). It also reports the values of any fixed parameters but, since these have been fixed rather than estimates, there is no standard error associated with these fixed parameters. For example, here is the output for the path coefficients (called "Regressions"):

Regressions:

```
                   Estimate  Std.Err  z-value  P(>|z|)
  X3 ~
    X1                0.520    0.031   16.609    0.000
    X2                0.544    0.033   16.533    0.000
  X4 ~
    X3                0.559    0.040   13.920    0.000
  X5 ~
    X3                0.501    0.038   13.191    0.000
```

[109] This means that if you have a small data set, the reported standard errors of the free parameters will be larger than reported and the null probability will be wrong and usually larger than reported.

The first line is reporting the path coefficient from X_1 to X_3, which is 0.520 (±0.031). According to our causal hypothesis, X_1 is the causal parent of X_3 and so the path coefficient *should* be different from zero; the true value in our simulation was 0.5. The null probability that the estimated value of 0.520 is zero in the statistical population, and that the observed value (0.520) is due only to random sampling variation, is very small (<0.0005). In other words, our causal model *requires* that our path coefficients be different from zero. What if this is not the case? As will be explained in Chapter 5, when we talk about statistical power, there are always two possible explanations. The first possibility is that the path coefficient really is different from zero, but it is close enough to zero that we can't detect the difference given our sample size. If this is the case, then our causal model has not been contradicted[110]. The second possibility is that the path coefficient really is zero, in which case you have just contradicted your causal model.

The remaining lines in the output of the summary list the maximum likelihood estimates of the remaining free parameters (the covariances and variances). The summary output lists the standard errors, z-values and their null probabilities for these as well. We are never interested in testing if the variance of an exogenous observed variable is different from zero; if this ever happened then you would not be able to fit the model. We are almost never interesting in testing if the variance of the latent error associated with an observed endogenous variable is different from zero, but it is possible for this to occur. When it does, it means that the observed parents of that variable have accounted for all of its variation.

An alternative way of getting the maximum likelihood estimates of the free parameters, including confidence intervals, is via the extractor function:

```
parameterEstimates( object=fit.true.model, ci = TRUE, level =
0.95, boot.ci.type = "perc", standardized = FALSE, fmi =
"default").
```

The `level=` argument is the probability level for the confidence interval (the default is 95%) and the `standardized=` argument determines if you also want the confidence intervals for the

[110] Unless our causal model stipulates that the causal effect must be significantly greater than the measured value. If so, then you can test this by substituting this expected effect value into the z-score. Even, better, you would simply fix the path coefficient to the expected value.

standardized values of the free parameters. Of course, you can easily calculate these confidence intervals yourself simply by multiplying[111] the standard error by 1.96. For instance, the first path coefficient was 0.502 with a standard error of 0.031 and so the 95% confidence interval for is $0.502 \pm 1.96(0.031)$.

Unlike with multiple regression, the purpose of SEM is not to produce prediction equations that maximize the explained variation for a target dependent variable. However, after obtaining a causal model that is not rejected, we might also want to know what proportion of the variation in the endogenous variables has been captured by their causal parents. This is given by their Pearson R^2 statistics. There are two ways of obtaining this information. One way is to use the extractor function `inspect(fit.true.model,"r2")`. The second way is to include the argument `rsquare=TRUE` inside the `summary()` function. Before fitting the path model in Figure 4.5b, which we know to be wrong, let me show you another useful function in lavaan that you can use whenever you suspect that lavaan has not correctly translated your path model into structural equations. This is the `parTable()` function, which outputs all of the parameters and states if they are free of fixed. Here is the output from a call to `parTable(fit.true.model)`:

```
   id lhs op rhs user block group free ustart exo label plabel start   est
se
1   1  X3  ~  X1    1     1     1    1     NA   0         .p1. 0.520 0.520
0.031
2   2  X3  ~  X2    1     1     1    2     NA   0         .p2. 0.544 0.544
0.033
3   3  X4  ~  X3    1     1     1    3     NA   0         .p3. 0.559 0.559
0.040
4   4  X5  ~  X3    1     1     1    4     NA   0         .p4. 0.501 0.501
0.038
5   5  X1 ~~  X1    1     1     1    5     NA   0         .p5. 0.516 1.031
0.065
6   6  X2 ~~  X2    1     1     1    6     NA   0         .p6. 0.467 0.934
0.059
7   7  X1 ~~  X2    1     1     1    0      0   0         .p7. 0.000 0.000
0.000
8   8  X4 ~~  X5    1     1     1    7     NA   0         .p8. 0.000 0.290
0.038
9   9  X3 ~~  X3    0     1     1    8     NA   0         .p9. 0.506 0.506
0.032
10 10  X4 ~~  X4    0     1     1    9     NA   0        .p10. 0.855 0.855
0.054
11 11  X5 ~~  X5    0     1     1   10     NA   0        .p11. 0.766 0.766
0.048
```

[111] The value of a standard normal variate, given a two-sided probability of $\alpha=0.05$, is 1.96.

Each parameter is listed. The key column is labelled "free". We see that there were 11 parameters and all but the seventh one, the covariance between X_1 and X_2 (X1 ~~ X2), is listed as free, since fixed parameters are indicated by a "0". If any parameter is listed as free when you wanted it to be fixed, or if any parameter is listed as fixed when you wanted it to be free, then you must go back to your model object and correct the mistake. Often, this mistake is because of the defaults used by lavaan to choose fixed vs. free parameters when you are not explicit.

4.9 What happens if your hypothesized path model is wrong?

We knew that our data were generated using the structural equations associated with the path diagram in Figure 4.5a. What happens if our causal hypothesis, as given in the path diagram, is wrong? To see, I will fit an incorrect path diagram (Figure 4.5b) to the data that was simulated using the path diagram in Figure 4.5a. Here is the code for the model object:

```
wrong.model<-"
X3~X1+X2
X4~X3
X5~X4
X1~~X1
X2~~X2
X1~~0*X2
"
```

Notice what is wrong with this model: X_5 is incorrectly caused directly by X_4 rather than by X_3. The dependency between X_4 and X_5 is therefore incorrectly attributed to this direct causal link rather than to (i) a common effect of X_3 on both and (ii) an unknown latent common cause of both X_4 and X_5. When we look at the part of the summary output giving the test for the goodness of fit, we see:

```
Model Test User Model:

  Test statistic                                56.537
  Degrees of freedom                                 6
  P-value (Chi-square)                           0.000
```

Now, the maximum likelihood chi-squared test statistic is 56.537 with 6 degrees of freedom and a null probability of less than 0.0005. We have 6 degrees of freedom, rather than 5 as in the

correct model, because we have only estimated nine free parameters[112] (4 path coefficients, 2 variances of exogenous observed variables and 3 error variances), thus $V(V+1)/2 - (p+q) = 5(6)/2 - (4+5) = 6$. Since the chance of observing such a large difference between the model-predicted and observed covariance matrices is less than 5 in 10,000, we must conclude that this difference cannot be due only to random sampling variation and that there is some non-random, systematic, error in the model.

Unfortunately, unlike with a dsep test, we can't simply look at the d-separation claims in the union basis set to see where the error lies. The fact that the likelihood maximization is done simultaneously on all of the free parameters means that errors in one part of the model will affect the estimates of the free parameters everywhere. The best that we can do is to look at the differences in the estimates between the model-predicted and observed covariance matrices. We want to base these differences using standardized values so that we can directly compare these differences. Covariances have the same measurement units as the variables to which they refer. If these variables have different measurement units, then we can't compare their numerical values. By standardizing the variables to unit variances, all of the variables will have the same measurement units (standard deviations from the mean) and so can be directly compared. We use the extractor function `residuals()`. Using standardized estimates, these differences will asymptotically follow a standard normal distribution, meaning that only 5% of the differences should be greater than 1.96 in absolute value. Values larger than this indicate unusually large differences between the observed and model-predicted covariance, and this means that the dependence between the two variables involved in the covariance is modelled incorrectly.

Here are the standardized differences in the covariances for the correct model:

```
residuals(fit.true.model,type="standardized")
```

```
$type
[1] "standardized"

$cov
       X3     X4     X5     X1     X2
X3 -1.189
X4 -1.189 -1.189
X5 -1.189 -1.189 -1.189
X1 -1.189  1.015 -0.532  0.000
X2 -1.189 -1.249  0.072 -1.189  0.000
```

[112] We do not estimate a free covariance between X_4 and X_5 as in the correct model.

Notice that none of these residual differences are greater than 1.96 in absolute value. Here are the standardized differences in the covariances for the incorrect model:

```
residuals(fit.wrong.model,type="standardized")

$type
[1] "standardized"

$cov
       x3     x4     x5     x1     x2
x3 -1.189
x4 -1.189 -1.189
x5  6.220 -1.189 -1.189
x1 -1.189  1.015  2.513  0.000
x2 -1.189 -1.249  3.000 -1.189  0.000
```

The covariances between X_3 and X_5 (6.220), between X_1 and X_5 (2.513) and between X_2 and X_5 (3.000) are unusually large in this incorrect model. This is telling us that the modelled causal links between X_5 and these other variables are wrong in some way. However, it doesn't give us any firm clues about why these incorrect causal links are wrong. Importantly, it also doesn't warn us about the incorrect causal link between X_5 and X_4.

4.10 Specifying starting values

As you begin using more complicated models involving explicit latent variables (Chapter 7) you might encounter a frustrating problem in which your model either fails to converge or else reports convergence errors. Unless you have a profound knowledge of the code buried in lavaan then its error messages will probably resemble something written by the Oracle of Apollo at Delphi. Sometimes such errors arise because you have mis-specified your model, in which case you must go back and modify it. However, it some cases the problem is not with your model structure but rather in the starting values used during the process of likelihood maximisation.

In my analogy between maximum likelihood estimation and a blind person trying to find the tallest hill in the maximum likelihood landscape, the more rugged and heterogeneous the landscape, the more difficult this search becomes and the more important it becomes for the

blind person to start their search relatively close to the tallest hill. If poor starting values are used, then they can be so far from the maximum likelihood values that the iterative process never converges. If this happens then the maximum likelihood procedure will terminate and report that convergence has not occurred; this is like our blind person starting so far away from the tallest hill that she gives up before she reaches the top. You can change the default number of iterations in the `sem()` function to (say) 200 by including the following argument: `control = list(iter.max=200)`, but a better solution is to change the default starting values for the free parameters. This is like placing our blind person close enough to the tallest hill that they don't have too walk too far. The syntax for specifying a particular starting value for a parameter in the model object is `start()*`. Here is how to modify our model object by specifying starting values for the free parameters[113] that are equal to the final values that we obtained:

```
true.model2<-"
X3~start(0.52)*X1+start(0.544)*X2
X4~start(0.559)*X3
X5~start(0.501)*X3
X1~~start(1.031)*X1
X2~~start(0.934)*X2
X3~~start(0.506)*X3
X4~~start(0.855)*X4
X5~~start(0.766)*X5
X1~~0*X2
X4~~start(0.29)*X5
"
```

Note that I had to explicitly include the error variances, for example, `X3 ~~ start(0.506) *X3`, in order to change their starting values. The results in the summary output are identical except that it took only 8 iterations to converge rather that the 12 iterations that it took without using these starting values. This is because the iterative process began closer to the actual values that maximise the likelihood. If I were to make these starting values very far from the final ones (say 1000), then I get the following warning from lavaan: "`lavaan WARNING: initial model-implied matrix (Sigma) is not positive definite; check your model and/or starting parameters`". If I had chosen other starting values

[113] You can do this for some, or all, of the free parameters.

(say 1 for the path coefficients and the free covariance and 10 for the variances), then the model would have converged without any errors but after taking 69 iterations.

You might be wondering: how can we choose "better" starting values if we don't know the values that actually maximise the likelihood? There are no fool-proof methods of choosing "better" starting values but there are some good rules of thumb.

1. Try to keep the range of your different variables to within the same order of magnitude. For instance, if you have one variable, measured in cm, which varies from 200 to 2000 and another variable, measured in kg, which varies from 3 to 10, then convert the first variable to metres so that the values only vary from 2 to 20.
2. Propose starting values that are at least within the same order of magnitude as the likely maximum likelihood values and that have the same sign.
3. Propose starting values of exogenous observed variances that are equal to their sample variances.
4. Take an educated guess about the proportion of the variance of each endogenous variable that might remain unexplained by their causal parents and then propose starting values for these endogenous variances equal to the sample variances of these variables multiplied by your guess. For instance, if an endogenous variable has a sample variance of 10.3 and you think that its causal parents might explain 30% of its variance (thus, its unexplained variance is 70%), then propose a starting value of 0.7*10.3 = 7.21. In my experience poor starting variables are particularly troublesome for free variances and free covariances.

4.11 Fixing parameter values, naming free parameters and defining functions of free parameters

Remember that every parameter is either "fixed" or "free". It is very easy to fix a parameter to a specific value. You simply "multiply" the parameter by its fixed value in the model object. Why

might you want to do this? According to metabolic theory, the basal metabolic rate of a homeotherm will increase by 0.75 times its body mass. If this was part of your causal hypothesis, then you would not simply specify that body mass is the causal parent of basal metabolic rate (i.e. body mass → basal metabolic rate) and so treat this path coefficient as a free parameter. Rather, you would want to fix the path coefficient from body mass to basal metabolic rate to 0.75. The first line in the model definition of object `true.model2` is `X3 ~ start(0.52)*X1+start(0.544)*X2`. If, according to our causal hypothesis, we believe that X1 is not only a direct cause of X3, but that its numerical value must be 0.75, then we would modify this line to read `X3 ~ 0.75*X1+start(0.544)*X2`. This line is telling lavaan that (i) both X1 and X2 are direct causes of X3, that (ii) a unit increase in X1 must increase X3 by 0.75 units (i.e. the value of this path coefficient is fixed and can't be changed), and that (iii) X2 will also directly increase X3 but that we don't know by how much.

All parameters in lavaan have a name. By default, the name of each parameter consists of three parts. The first part is the name of the variable that appears on the left-hand side of the formula. The middle part is the operator used in the formula. The last part is the name of the variable that appears on the right-hand part of the formula. For example, the first line in the model definition of object "true.model2" is `X3 ~ start(0.52)*X1+start(0.544)*X2`. Therefore, the name of the first path coefficient is "X3~X1". However, in more complicated models that we will consider in later chapters, we will be specifying various constraints on parameters or sets of parameters, and it is then more convenient to assign our own names to these parameters. There are two cases where naming parameters makes life easier even in the simpler path models considered in this chapter: (i) placing simple logical or hypothesised constraints on parameters and (ii) partitioning total effects into direct and indirect components. The usefulness of naming parameters in multigroup SEM will be explained in Chapter 8. To assign your chosen name to a parameter, simply choose a name that follows the naming conventions of R and that does not already exist in the data file, and then "multiply" this name to the variable. For example, if I want to call the path coefficient measuring the effect of X3 on X1 (X1→X3) as "a31" then I would change the first line in `true.model2` to `X3 ~ a31*start(0.52)*X1+start(0.544)*X2`. Once you have named a parameter then you can define new functions involving this named parameter. Two common reasons why you might

want to do this are to calculate indirect effects along different paths and to place equality or inequality constraints on free parameters.

An example of placing an inequality constraint on a free parameter is when forcing a variance to be non-negative. Of course, no variance can be negative but the maximum likelihood algorithm inside lavaan doesn't know this fact. An estimated residual variance that is negative is usually a sign of some fundamental problem in your model, but not always. If an endogenous variable is very strongly determined by its causal parents, then its unexplained (residual) variance will be very close to zero and it is possible that sampling variation alone could result in the maximum likelihood estimate of this variance "wandering" into the forbidden part of parameter space containing negative variances. To prevent this, we would specify an inequality constraint on a free parameter. For instance, to specify that the residual variance of variable X5 cannot be negative in the `true.model2` object, I could first call this free parameter "var5" (thus, `X5~~var5*start(0.776)*X5`) and then add a new line containing the constraint: `var5>0`. All the usual relational operators in R can be used (=, >, <, <=, >=, !=). Similarly, if I wanted to specify that the path coefficient measuring the effect of X3 on X1 must be negative, then I would give a name to this path coefficient ("`a31`") and then add the line: `a31<0`. Here is the new model object (`true.model3`) that includes these two inequality constraints:

```
true.model3<-"
X3~a31*start(0.52)*X1+start(0.544)*X2
X4~start(0.559)*X3
X5~start(0.501)*X3
X1~~start(1.031)*X1
X2~~start(0.934)*X2
X3~~start(0.506)*X3
X4~~start(0.855)*X4
X5~~var5*start(0.766)*X5
X1~~0*X2
X4~~start(0.29)*X5
var5>0
a31<0
"
```

Remember that the data were generated by the structure specified in `true.model2`, but that the path coefficient linking X3 and X1 was 0.5. The effect of forcing this path coefficient to be negative, when the true value is positive, is to introduce a non-random lack of fit. Here is the part of the summary output giving the fit statistics:

```
Model Test User Model:

  Test statistic                               223.940
  Degrees of freedom                                 5
  P-value (Chi-square)                           0.000
```

A new section is now included in the summary output:

```
Constraints:
                                               |Slack|
    var5 - 0                                     0.766
    0 - (a31)                                    0.000
```

The column called "|Slack|" gives the absolute value of the difference between the actual maximum likelihood value that is estimated and the boundary of the constraint. Thus, the first line tells us that during the maximum likelihood estimation, the algorithm never assigned a value to var5 that went below the lower boundary of zero. The second line tells us that the algorithm "wanted" to assign a value to the path coefficient (`a31`) that was larger than its upper boundary of zero but could not because of the constraint, and so it stopped when it reached this upper bound.

We could force two (or more) free parameters to be equal in the same way. If your causal hypothesis requires that two path coefficients be equal in value, but does not specify this value, then you would name each path coefficient (say `a31` and `a21`) and then add the equality constraint `a31==a32`. Alternatively, if we give two or more free parameters the same name, then we are forcing them to be equal. We will use this trick extensively in Chapter 8 in the context of multigroup models.

You have already learned how to partition the total effect between any two variables into its different components in Chapter 3 (direct effects, indirect effects, spurious associations and total effects). Lavaan doesn't have any function that does this for you, and this is probably a good thing because even moderately complicated models will have many different combinations of direct, indirect and total effects. However, the "`:=`" operator in lavaan allows you to define new parameters that are functions of previously defined parameters and so we can create new composite parameters representing specified effects and then obtain their maximum likelihood estimates, standard errors and so on.

For example, we might want to calculate the indirect effects of X_1 and X_2 on X_4 in Figure 4.1. Using the rules that I gave in Chapter 3, we can do this in lavaan by first naming the path coefficients and then defining the two new composite parameters representing these two indirect effects. Here is the code for the new model object:

```
true.model4<-"
X3~a31*start(0.52)*X1+a32*start(0.544)*X2
X4~a43*start(0.559)*X3
X5~start(0.501)*X3
X1~~start(1.031)*X1
X2~~start(0.934)*X2
X3~~start(0.506)*X3
X4~~start(0.855)*X4
X5~~var5*start(0.766)*X5
X1~~0*X2
X4~~start(0.29)*X5
indirect.X1:=a31*a43
indirect.X2:=a32*a43
"
```

When I fit this new model object using the `sem()` function and then output the summary, the maximum likelihood estimates of these two new parameters (`indirect.X1` and `indirect.X2`), their standard errors, their z-values and the null probabilities that each of these indirect effects are zero are output:

```
Defined Parameters:
                   Estimate  Std.Err  z-value  P(>|z|)
    indirect.X1       0.291    0.027   10.668    0.000
    indirect.X2       0.304    0.029   10.648    0.000
```

Given this information, we could even test the null hypothesis that these two indirect effects are equal by using a z-test:

$$z = \frac{[0.291 - 0.304] - 0}{\sqrt{0.027^2 + 0.029}} = 0.33$$

The null probability of finding at least this large a difference between the two indirect effects simply due to sampling variation is 0.74; i.e. 2*(1-pnorm(z)).

4.12 Dealing with violations of assumptions: small sample sizes

It is not true that the sampling distribution of the maximum likelihood chi-squared statistic (MLX^2) follows a theoretical Chi-Squared distribution (χ^2). There is no known function that describes the sampling distribution of the maximum likelihood chi-squared statistic. However, it is true that the sampling distribution of the maximum likelihood chi-squared statistic *converges* towards a theoretical Chi-Squared distribution as the sample size increases. Stated equivalently, the sampling distribution of the maximum likelihood chi-squared statistic *asymptotically* follows a theoretical Chi-Squared distribution (χ^2) in the limit as the sample size approaches infinity.

The maximum likelihood chi-squared statistic is a bit like home-made wine. When you taste these "wines", you realise that they vary from "gut-rot" to "drinkable" to "divine". At very small sample sizes the MLX^2 statistic is like gut-rot wine; it bears an approximate resemblance to the true χ^2 distribution but there is no confusing the two. At moderate sample sizes the MLX^2 is like "drinkable" home-made wine; it is a reasonable approximation of the real thing unless it is to be used for a special occasion. It is only when sample sizes are very large that one cannot distinguish between the two. So how big is "big enough" and what can be done if one's sample is not big enough? In this section, I will discuss the effects of sample size on the MLX^2 assuming that the data follow a multivariate normal distribution. To explore these questions, I will use simulations drawn from the path model shown in Figure 4.6 with all variables being drawn from a standard normal distribution (i.e. zero mean and unit standard deviation).

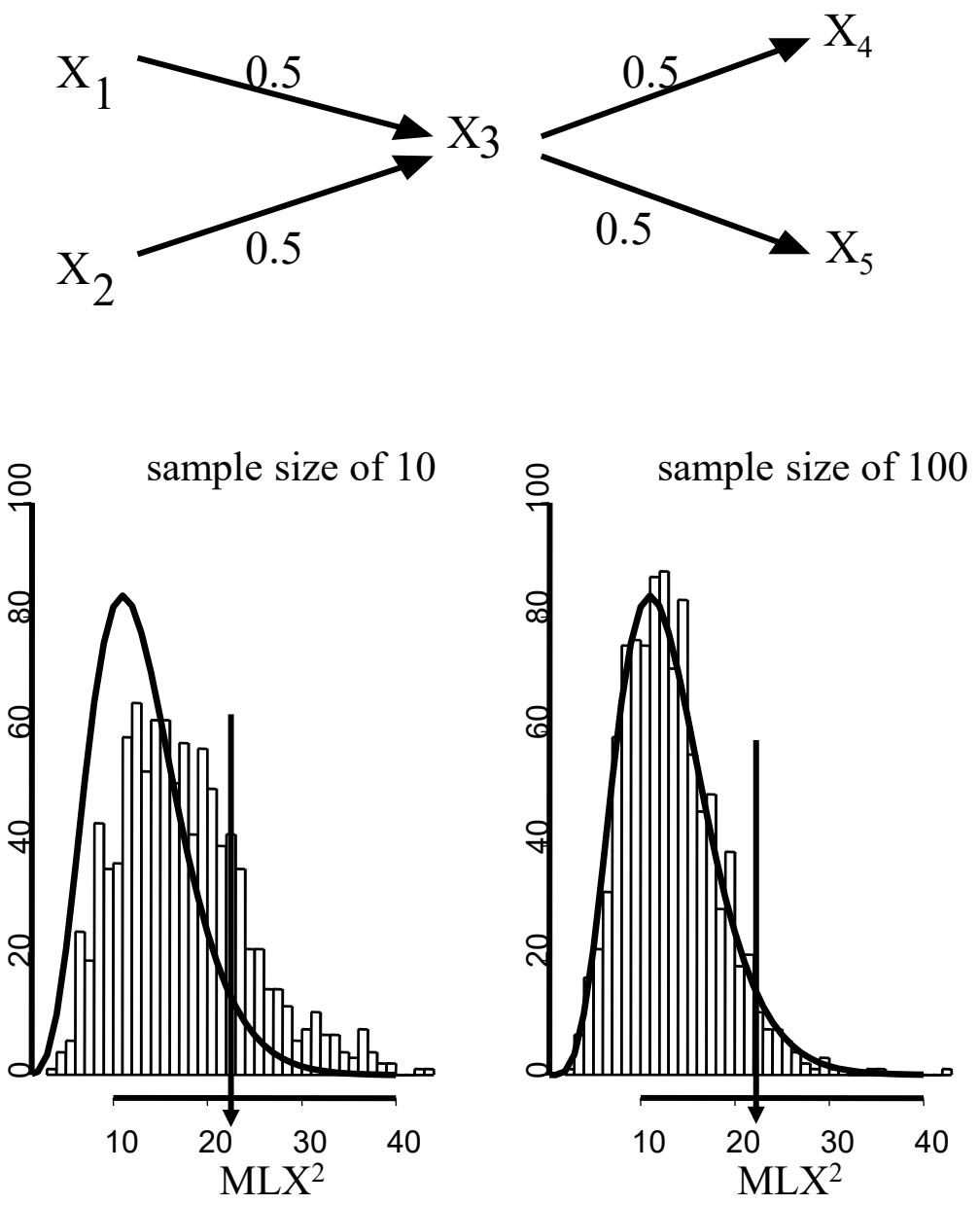


Figure 4.6. The distribution of the maximum likelihood chi-squared statistic (MLX^2) from 1000 independent data sets in which each data set has either 10 or 100 observations that have been generated following the DAG shown at the top. The theoretical χ^2 distribution with 13 degrees of freedom is superimposed as the solid black curve. The arrow shows the value of the theoretical χ^2 value at the 5% significance level.

Figure 4.6 shows the empirical sampling distribution of the MLX^2 statistic, based on 1000 independent data sets. I fixed all path coefficients to their theoretical values (0.5) and all the

error variances to their theoretical values. This way, the only free parameters were the variances of X_1 and X_2, and the model covariance matrix could be determined without iteratively minimising the MLX^2. There were therefore 13 degrees of freedom and the curve shown in Figure 4.6 is the theoretical χ^2 distribution with 13 degrees of freedom. The first histogram shows the distribution of the MLX^2 statistic in the 1000 data sets with 10 observations each. Clearly, this empirical distribution is not well approximated by the theoretical χ^2 distribution. More of the values are larger than is expected given the theoretical χ^2 distribution. As a result, when the real null probabilities were approximately 0.10, 0.05, 0.025 and 0.01, the reported asymptotic probabilities were 0.01, 0.004, 0.001 and 0.0002. In other words, we would reject the model much more often than we should. The second histogram shows the distribution of the MLX^2 statistic in the 1000 data sets with 100 observations each. When the real null probabilities were approximately 0.10, 0.05, 0.025 and 0.01, the reported asymptotic probabilities were 0.103, 0.056, 0.028 and 0.012. Now, the empirical and theoretical distributions are quite close. In this second case, if we assume that the MLX^2 is truly distributed as a χ^2 distribution, we will make only insignificant errors. If we continue to increase the sample size of each data set beyond 100, then the empirical and asymptotic null probabilities become even closer, but the never become identical (unless we had an infinite sample size).

In general, small sample sizes result in conservative probability estimates based on the theoretical χ^2 distribution. In other words, if you have a small data set then the true probability will probably be larger than the value obtained when assuming a χ^2 distribution. In this case, if your model produces a MLX^2 value whose (asymptotic) null probability[114] is below the chosen significance level, then you have an ambiguous result and you will have to use a different method of estimating the true probability level. For the model shown in Figure 4.6, a sample size of at least 30 provides a passable estimate of the tail probabilities but with somewhat conservative probability estimates and a sample size of 50 is quite acceptable. In general, the more the number of free parameters in the model that need to be estimated, the larger the sample size that is required. More complicated models may require sample sizes of 200 or more. One rule of thumb is that there should be at least five times more observations than free parameters in

[114] The values output by lavaan are asymptotic null probabilities, which assume a large sample size.

the model (Bentler 1995), but this rule of thumb assumes multivariate normality; violations of normality will be discussed later.

What can be done if your sample size is too small to confidently assume that the sampling distribution of the MLX^2 statistic is close to the theoretical χ^2 distribution? One way is the estimate the actual sampling distribution of the MLX^2 statistic using Monte Carlo methods (Manly 1997). This method is implemented in the `MCX2()` function in the pwSEM package. Since the MLX^2 statistic requires that we calculate the determinant, and the inverse, of the model covariance matrix it is useful to choose a matrix for which this can be easily done. The determinant of a square matrix whose non-zero values are all on the diagonal is simply the product of these diagonal values. Similarly, the inverse of such a diagonal matrix is simply a diagonal matrix whose diagonal values are the inverse of the original matrix. We therefore simulate data with N observations from a model consisting of v mutually independent normally distributed random variables. The predicted covariance matrix, Σ, of such a model has non-zero values only on its diagonal. There are v(v+1)/2 non-redundant elements. If we estimate the variance of q of the v variables, which will be on the diagonal of Σ, then there will be v(v+1)/2-q degrees of freedom. So, here are the four steps needed to estimate an empirical probability level for a MLX^2 statistic:

(1) Given a desired number of degrees of freedom (df), find the smallest integer value of v such that $v \geq \frac{-1+\sqrt{1+8df}}{2}$. For example, if df=9 then we need the smallest integer value of v such that $v \geq \frac{-1+\sqrt{1+8(9)}}{2} = 3.8$. Thus, v=4. Find the integer value of c such that $c = \frac{v(v+1)}{2} - df$. So, if df=9 and v=4 then c=1.

(2) Construct a model covariance matrix $\boldsymbol{\Sigma}$ with v rows and columns. Leave the first c diagonal elements blank; these will be filled by the sample estimates of the variances of the first c of the mutually independent normally distributed random variables in step 4. Define all other diagonal elements (the remaining variances) to be 1 and all non-diagonal elements to be 0. This is the population covariance matrix for v mutually independent standard normal variates, of which the variances of the first c of

these variables have been estimated from the data. This model covariance matrix will have df degrees of freedom. For instance, if your actual data set has n=30 observations and you have calculated v=4 in step 1, then you would generate 4 mutually independent vectors of 30 standard normally distributed random numbers using rnorm(n=30,mean=0,sd=1), estimate the variance of the first vector, and place this estimate in the first element of the diagonal matrix. The remaining three diagonal elements of this matrix would be equal to 1.

(3) Now, repeat step three a large number of times (say, N=10000). Each time, besides performing step 3 which results in a matrix $\boldsymbol{\Sigma}$, create a second matrix (the sample covariance matrix, **S**), which is also a diagonal matrix whose diagonal elements contain the sample variances of the four vectors of normally distributed random numbers. Now calculate the maximum likelihood chi-square statistic

$$MLX_i^2 = (n-1)(Ln|\Sigma| + tr(S_i\Sigma^{-1}) - Ln|S_i| - c)$$

(4) Count the number (x) of the N MLX^2 values that are greater than the value of the MLX^2 value obtained in your real data, as output in lavaan.

(5) The estimated empirical probability of your data will be p=x/N. The 95% confidence interval of this estimate is $p \pm 1.96\sqrt{\frac{p(1-p)}{N}}$.

For instance, imagine that you have fit a model in lavaan with 13 degrees of freedom to a data set with 15 observations and have obtained a maximum likelihood chi-squared value of 23.10, as output in the `summary()` function. The asymptotic null probability that is output by lavaan would be 0. 040 but you know that this null probability is wrong because of the small sample size of your data. You can obtain the estimated true null probability via Monte Carlo simulation by typing: `MCX2(model.df = 13,n.obs = 15,model.chi.square = 23.1,n.sim=10000, plot.results=F)`. The first argument is the model degrees of freedom (output by lavaan), the second argument is the maximum likelihood chi-squared statistic (output by lavaan), the third argument is the number of independent simulations that you want (defaults to 10000) and the fourth argument is a logical value specifying if you want a graphical summary of the results (defaults to FALSE). Here is the output:

```
$MCprobability
[1] 0.1044

$MLprobability
[1] 0.04049176
```

The output says that the estimated Monte Carlo null probability (MCprobability) is 0.1044. In other words, even though the asymptotic null probability (MLprobability), as output by lavaan is below the 5% significance level (0.040), the better Monte Carlo estimate is well above the 5% significance level (0.104). The larger the value that you specify in the `n.sim=` argument, the more precise the estimated null probability. You could calculate the 95% confidence interval for this estimate as:

$p \pm 1.96\sqrt{\frac{p(1-p)}{n.sim}} = 0.1044 \pm 1.96\sqrt{\frac{0.1044(1-0.1044)}{10000}} = 0.1044 \pm 0.006$. In this case, the 95% confidence interval is far above the 5% significance value and so we would have no good reason to reject our model. There is no reason to choose more that the default value of 10000 unless your chosen significance level is inside the confidence interval of your resulting Monte Carlo estimate.

4.13 Dealing with violations of assumptions: nonnormality

Since non-normality can cause problems with the maximum likelihood chi-square statistic, a number of alternative ways of fitting the model have been devised. Different estimation methods in lavaan are specified via the "estimator" argument in the `sem()` function. Besides the default maximum likelihood estimator (`estimator="ML"`), the lavaan package has a number of different estimators. Most commercial SEM programs will include statistics based on generalised least squares, elliptical estimators and distribution-free estimators, as well as a method of correcting for non-normality that produces "robust" chi-square statistics and confidence intervals. The most popular, and best studied, method that produces estimates that are robust to deviations from non-normality, comes from (Satorra and Bentler 1988), the Satorra-

Bentler (SB) Chi-Square, and is obtained in lavaan by setting the estimator argument of `sem()` to "MLM" (i.e. `estimator="MLM"`).

There now exists an extensive literature that uses Monte Carlo simulations to explore the relative merits of these different solutions for non-normality. Different studies have explored the effects of sample size, the number of free parameters, model type (measurement models, path models, full structural models) and distributional violations (kurtosis, skew and non-independence of errors and their causal non-descendants). (Hoogland and Boomstra 1998) have done a meta-analysis of these studies. Their main recommendations are the following:

1. With respect to sample size, they recommend that there be at least five times as many observations as there are degrees of freedom in the model. If this is not the case, then you can use the MCX2 function.
2. When the observed variables have an average positive kurtosis[115] of 5 or more, the sample size may have to be increased up to 10 times the degrees of freedom.
3. The generalised least squares chi-square statistic has an acceptable performance for a sample size that is two times smaller than the sample size needed for an acceptable performance of the maximum likelihood chi-square statistic. This estimator is obtained in lavaan via `estimator = "GLS"`.
4. With small samples, the standard errors of the estimates of the free parameters are biased. Positive kurtosis results in estimates of the standard errors that are smaller than they should be. Negative kurtosis results in estimates of the standard errors that are larger than they should be.
5. The degree of skew has little effect on the bias of the estimators.
6. The asymptotic distribution-free estimator should not be used except for very large sample sizes (>1000).

[115] Kurtosis refers to the degree to which the distribution is thin with wide tails or fat with short tails. Kurtosis, rather than skew (asymmetry of the distribution about the mean) is a more serious violation of normality with covariance-based SEM because we are modelling variances and covariances rather than means (like with ANOVA etc.).

7. The Satorra-Bentler (SB) robust estimator, upon which is based their robust chi-square statistic and standard errors, largely corrects for excessive kurtosis and for problems in which the errors are not independent of their causal non-descendants. This is particularly important for models that include latent variables and measurement models (Chapter 7), since the SB chi-squared statistic can correct for cases in which the latent variables and the measurement errors are not independent.

Basically, unless your data are extremely kurtotic, you can still perform a reasonable test of your causal model simply by including the argument `estimator="MLM"` within the `sem()` function. If you have a small sample size, then you can use the `MCX2()` function from the pwSEM package while using the Satorra-Bentler (robust) estimate of the maximum likelihood chi-squared statistic rather than the default maximum likelihood chi-squared statistic.

As a last resort, you can use bootstrap methods (Bollen and Stine 1992). The bootstrap is related to Monte Carlo methods except that, rather than sampling from some theoretical distribution (multivariate normal or otherwise), you sample from your own data to build up an empirical sampling distribution. However, bootstrap methods do not perform well with small sample sizes. See (Manly 1991) for a discussion of bootstrap methods in biology. Bootstrapped estimates are obtained in lavaan in one of two ways. While fitting the model in the `sem()` function you can get bootstrapped standard errors of your free parameters by specifying the argument `se="boot"` and you can get a bootstrapped null probability for the chi-square statistic by specifying the argument `test="boot"`. If you are interested in the details, box 4.2 summarises the steps required to generate a bootstrap distribution of the maximum likelihood chi-square statistic.

Box 4.2

Here are the steps to take in order to generate a bootstrap sampling distribution and perform an inferential test.

1. Given your original data set (**Y**) with N rows and p variables centered about their means, calculate the sample covariance matrix (**S**), obtain the predicted model covariance matrix **(Σ)** and the maximum likelihood chi-square statistic, X^2.

2. Calculate the Cholesky factorization of **S** and **Σ** to give $\mathbf{S^{-1/2}}$ and $\mathbf{\Sigma^{1/2}}$.

3. Form a new data set: $\mathbf{Z}=\mathbf{Y}\mathbf{S^{-1/2}}\mathbf{\Sigma^{1/2}}$.

4. Randomly choose N observations from **Z** with replacement to form a bootstrap sample $\mathbf{Z}^*$. Form the covariance matrix from this bootstrap sample, fit the model to these data, and save the bootstrap value of the maximum likelihood chi-square statistic (X^{2*}).

5. Repeat step 4 a large number of times (at least 1000).

6. Count the proportion of times that X^{2*} is greater than X^2. This proportion is the empirical estimate of the probability of observing the data given the model. Note that this probability does not assume any particular sampling distribution.

4.14 Measures of approximate fit

This next section deals with various methods of assessing the degree of "approximate" fit between data and a theoretical model. I don't like these methods, for reasons that I will explain below. However, they are popular with many users of SEM and are produced in lavaan. These measures of approximate fit are generally used once the model has already been rejected and the

purpose of these approximate fit measures are to determine the degree to which the rejected model is “approximately” correct.

The origin and rationale behind the use of these approximate fit indices comes from a consideration of statistical power. Statistical power will be explained in more detail in Chapter 5. The power of a statistical test can be defined as the probability that the test will reject the null hypothesis when it is indeed false. Statistical power increases as the sample size of your data set increases.

Usually, more statistical power is a good thing. Tests of structural equations models also have power properties. The justification for using alternative tests of fit is based on the premise that statistical power is not always such a good thing. If you remember the section in Chapter 2 dealing with the logic of scientific inference, then you will recall that no hypothesis is ever really tested in isolation. Every hypothesis contains within it many other auxiliary hypotheses. In the context of testing covariance-based structural equations models we are really interested in knowing if the causal structure of the model (i.e. the DAG or MAG) is wrong. Unfortunately, when we conduct our statistical test, we are testing all aspects of the model: the causal implications, the distributional properties of the variables, the linearity of the relationships and so on. Now, when we add the notion of statistical power to our argument then we realise that, as sample size increases, we run a greater and greater risk of rejecting our models because of very minor deviations that might not even interest us. This point was raised early in the history of modern SEM by Jöreskog (1969).

What might these uninteresting and minor deviations be? These can’t be minor deviations from multivariate normality since the maximum likelihood chi-square statistic is asymptotically robust against non-normality. In any case, we have already seen ways of dealing with this. Small amounts of non-linearity could be one such minor deviation that would not interest us. If some parameter values (for instance, path coefficients or error variances) are fixed to non-zero values in our model then small deviations from these fixed values might be another minor difference that would not interest us. However, the principal “minor deviation” that is evoked in the justification for measures of approximate fit is a minor deviation in the causal structure of the model (i.e. the DAG or MAG). The theoretical objective of the various indices of approximate fit is therefore to somehow quantify the degree of these deviations in the causal structure of the

model. The various alternative fit indices attempt to quantify the degree of such deviations by measuring the difference between the observed covariance matrix and the predicted (model) covariance matrix. The most popular fit indices do this in a way that standardise for differences in sample size.

At first blush then, these indices of approximate fit have a seductive quality. Would it not be nice, after having found that your preferred causal explanation (as translated by the structural equations model) has been rejected, to be able to say: "but it is almost right! The remaining lack-of-fit is only due to minor errors that are not really very important anyway". This, I suspect, is the real (psychological) objective of these fit indices. Even this weakness of the flesh could be tolerated if there was strong justification for the implicit assumption that minor errors in specifying the causal structure will translate into only minor differences between the observed and predicted covariance matrices. Certainly, there is a relationship between the strength and number of errors in the SEM and the value of these indices of approximate fit, but the relationship is not perfect and there is (to my knowledge) no theoretical result quantifying the relationship. To me, evoking such an argument of approximate fit to justify accepting a causal model is like the old joke about the drunk in the parking lot[116]. The alternative fit indices measure different aspects of the ability of the structural equations to *predict* the observed covariance matrix, not the *explanatory* ability of the *causal* model. As such, the indices of approximate fit commit the sort of subtle error of causal translation that I discussed in Chapter 2: small (but real) differences between the observed and predicted covariances of the observational model do not necessarily mean only small (but real) differences between the actual causal structure and the predicted causal structure. Remember that the goal of SEM is not to "find a model that fits", but rather to "find a causal explanation (causal graph) that is not contradicted by empirical data". If the null probability is low enough that you cannot reasonably ascribe the measured lack of fit, given your causal graph, simply to sampling variation, then you should expend your effort in trying to understand what causes the systematic lack of fit.

Now that I have given my reasons why you should not use these alternative fit indices, you can read the justifications of those who promote them and decide for yourself (Bentler and Bonnett

[116] You enter a parking lot late at night and see a drunk on his knees underneath the only streetlight. He explains that he is looking for his car keys. "Are you sure that you lost your keys here" you ask? "No", he answers. "In fact, I lost them in that dark corner, but at least here I have enough light to see".

1980, Browne and Cudeck 1993, Tanaka 1993) Below, I describe two of the more popular alternative fit indices although there seems to be a growth industry in inventing new ones. The book by Bollen and Long (1993) contains a number of chapters that deal with these alternative indices of approximate fit.

4.15 Bentler's comparative fit index

Let's go back to the maximum likelihood chi-square statistic for a moment. This statistic, and it inferential test, measures exact fit between the observed and predicted covariance matrices. The logic is that if the data are generated by the process specified by the structural equations (and therefore the causal structure of which these equations are a translation) then the observed and predicted covariance matrices will be identical except for random sampling variation. If this assumption is true, then the maximum likelihood chi-square statistic will asymptotically follow a chi-square distribution with the appropriate degrees of freedom (υ). Actually, it is more precise to say that this statistic will asymptotically follow a *central* chi-square distribution (χ^2_ν) with the appropriate degrees of freedom. The central chi-square distribution is a special case of a more general chi-squared distribution called the *non-central* chi-square distribution. The non-central chi-square distribution ($\chi^2_{\nu,\lambda}$) has two parameters: the degrees of freedom (υ) and the non-centrality parameter (λ). A central chi-square distribution is simply a non-central chi-square distribution whose non-centrality parameter (λ) is zero.

Now, if the degree of mis-specification of the model covariance matrix is not zero (as assumed in the test for exact fit) but is small relative to the sampling variation in the observed covariance matrix, then the maximum likelihood chi-square statistic actually asymptotically follows a non-central chi-square ($\chi^2_{\upsilon,\lambda}$) distribution and the non-centrality parameter (λ) measures the degree of mis-specification. The expected value of the non-central chi-square distribution is simply the expected value of the central chi-square distribution plus the non-centrality parameter: $E[\chi^2_{\upsilon,\lambda}] = E[\chi^2_\upsilon] + \lambda = \upsilon + \lambda$. In practice, the non-centrality parameter is estimated as the value of the maximum likelihood chi-square statistic (MLX^2) minus the degrees of freedom of the

model (i.e. the expected value that the maximum likelihood chi-square statistic would have if there were no errors of misspecification). Because the non-centrality parameter cannot be less than zero, negative values are replaced with zero. Therefore $\lambda=\max\{(MLX^2-\nu),0\}$.

The Bentler comparative fit index uses this fact to measure by how much the proposed model has reduced the non-centrality parameter (thus, the degree of misspecification) relative to a baseline model that is definitely wrong. The most common baseline model is one that assumes that the variables are mutually independent[117]. If λ_i is the estimate of the non-centrality parameter for the model of interest and λ_0 is the estimate of the non-centrality parameter for the baseline model that is definitely wrong, then the comparative fit index is defined as: $CFI = \frac{\lambda_0 - \lambda_i}{\lambda_0}$. If the model of interest fits exactly then the expected value of its non-centrality parameter (λ_i) would be zero and the CFI value would be 1.0. Therefore, the CFI index varies from 0 (the proposed model fits no better than a baseline model in which all the variables are independent of each other) to 1. The sampling distribution of this index is unknown. Users of this index consider a value of at least 0.95 as being an acceptable "approximate" fit. This value is not a probability, it is simply an index, and there is no theoretical justification for this value beyond the results of numerical simulations. It is simply a rule of thumb. The extractor function `fitMeasures()` in lavaan prints out a large number of different indices of approximate fit. If you specify the argument "cfi" within this function then only the Comparative Fit Index is output: `fitMeasures(fit,"cfi")` where "fit" is the name of the object created by the `sem()` function. Alternatively, you can include the argument `fit.measures=TRUE` within the `summary()` function but this will output several other measures of approximate fit as well.

4.16 Approximate fit measured by the Root Mean Square Error of Approximation (RMSEA)

[117] i.e. a DAG with no arrows or double-headed arrows between the variables.

Another popular measure of approximate fit was developed in Steiger (1990) and expanded in Browne and Cudeck (1993). This measure also relies on the non-centrality parameter. The root mean square error of approximation (RMSEA, ε) is defined as:

$\varepsilon = \sqrt{\frac{\lambda}{n\upsilon}} = \sqrt{\frac{\max\{MLX^2 - \upsilon), 0\}}{n\upsilon}}$ where λ is the non-centrality parameter and ν is the degrees of freedom of the model. If we propose a null hypothesis for the RMSEA (H_0: $\varepsilon_a \leq a$) then we can test this hypothesis using the non-central chi-square distribution and produce confidence intervals around it. This is obtained in R via `pchisq(q=,df=,ncp=, lower.tail=F)`. Here, q is the MLX^2 statistic, df is the model degrees of freedom, and ncp is the non-centrality parameter. Of course, if your null hypothesis is that $\varepsilon_a=0$ then you are doing a test of exact fit with reference to the central chi-square distribution. Here are the steps:

1. Specify the null hypothesis H_0: $\varepsilon_a \leq a$
2. Obtain the maximum likelihood chi-square statistic (MLX^2) and the non-centrality parameter $\lambda^* = n \cdot \nu \cdot \varepsilon_a^2$. These are printed out by lavaan.
3. Find the probability of having observed MLX^2 given a non-central chi-square distribution with parameters ν, λ^*.
4. If the probability is less than your chosen significance level, reject the null hypothesis and conclude that ε_a is greater than that specified in the null hypothesis.

An obvious problem with this test is in choosing the null hypothesis. Remember that these indices of approximate fit are used when one has already rejected the null hypothesis of exact fit (i.e. $\varepsilon_a=0$). We already know that there is something wrong with the model. Browne and Cudeck (1993) recommend the null hypothesis of $\varepsilon_a \leq 0.05$ but this is only their rule of thumb. In other words, models whose value is 0.05 or less are judged to be "approximately correct". This value is not a probability level and there is no compelling theoretical reason for choosing this value as a reasonable level of "approximate" fit beyond the results of numerical simulations. This index of approximate fit is obtained by specifying "rmsea" in the `fitMeasures()` extractor function: `fitMeasures(fit,"rmsea")`.

Quite apart from using the RMSEA to measure "approximate" fit, there is a very useful property of the inferential test for this fit statistic. If we have not been able to reject our model at our chosen significance level, then it is still important to be able to estimate a confidence interval for the RMSEA. In such a case the confidence interval will have a lower bound of 0. The upper bound will reflect the statistical power of our test. A large upper bound indicates that the test had little statistical power to reject alternative models. A 90% confidence interval for RMSEA would not reject the null hypothesis of exact fit at the 5% level. This interval can be calculated as the values of λ for a non-central chi-square distribution whose 5% and 95% quantiles equals the calculated MLX ^{2} statistic (Browne and Cudeck 1993).

4.17 Missing data

We sometimes have "holes" in our data sets due to missing values in some variables. In fact, we sometimes have rather complicated patterns of "missingness" in which we lack values on different variables for different observations. The pattern of missingness refers to the pattern of "holes" for a given variable or set of variables. For instance, imagine that you have a variable (x) whose values in your data set are x={2.1, 3.2, NA, 5.0, NA, 1.1}, where "NA" is the special code in R for a missing value. If we code a missing value by 0 and an observed value by 1 then the pattern of missingness for variable x is {1, 1, 0, 1, 0, 1}. The mice package in R (van Buuren and Groothuis-Oudshoorn 2011) contains the `md.pattern()` function that takes your data frame or matrix as its argument and outputs this pattern of missingness.

What do we do when this happens? This depends on the type of missing data. Little and Rubin (1987) define three different types of missing data that have acronyms of MCAR ("missing completely at random"), MAR ("missing at random") and NMAR ("not missing at random"). Data for a given variable are missing completely at random (MCAR) when the fact that a value is missing or not is unrelated to the actual values of that variable or to the values of any other variable. Of course, it is not possible to statistically determine if those values of a variable that are missing are systematically different from those that are observed for this same variable because… well… we don't know the values of the missing observations! On the other hand, we

often know why we missed some observations. If the values are missing because we couldn't measure values within certain ranges, then the pattern of missingness is related to its value (NMAR). For instance, if our measuring device cannot measure below a certain range then any value below that range is "missing" because of that limitation, not due to random reasons. On the other hand, if we missed them because we were sick, or the measuring device broke down, or it was raining, then we might be willing to assume MCAR unless the variable we are measuring is related to our health, the state of our measuring device, or precipitation patterns.

A variable is missing at random (MAR), but not completely at random, if its pattern of missingness is unrelated to its own values but is related to (and can therefore be predicted from) other variables in the data set. In other words, a variable is missing at random (MAR) if we can predict the pattern of missingness of this variable given the values of other variables. It is possible to determine this by conducting a logistic regression in which the pattern of missingness (the 1s and 0s) of the variable is regressed on the values of the other variables.

If you have variables that are not missing at random then I don't know of any statistical solution. However, if your variables are MAR then there are solutions. You already understand the notion of maximum likelihood estimation. Maximum likelihood estimation uses information about the covariances (and means if intercepts are included) to obtain parameter estimates for the model covariance matrix. A variant of maximum likelihood, called "full information maximum likelihood" (FIML) finds the parameter estimates for each observation that maximize the likelihood for each observation, after which these are combined to produce the maximum likelihood estimates for the model covariance matrix. Because FIML applies to each observation then, if the patterns of missingness in the data set are MCAR or MAR, we can proceed in the presence of missing values. To do this in lavaan, you simply have to include an argument `sem(…, missing="fiml")` in the `sem()` function. This allows you to include lines in your data set that have missing values, but the result is entirely dependent on whether or not your assumptions about the causes of missing values are good.

4.18 Removing phylogenetic or spatial signals in SEM

Species that experience similar selection pressures from the environment tend to have similar values of traits that are being selected. However, recently diverged species also tend to have trait values that are similar to each other because they have inherited such trait values from their common ancestor (phylogenetic constraints). Evolutionary biologists are often interested in dealing with phylogenetic effects on the covariance between phenotypic traits and of removing such phylogenetic signals. Such methods are complicated by the fact that one must make certain assumptions about the way in which evolution and speciation occurs. For instance, (Felsenstein 1985)'s method of phylogenetically independent contrasts (PICs) assumes a process of Brownian motion and linear relationships between traits while (Martins and Hansen 1997) method of phylogenetic generalised least squares (PGLS) method can incorporate other assumptions as well. These methods allow one to "remove" the effects of a shared evolutionary history from the variables such that the remaining variation in the residuals represents variation that does not incorporate the phylogenetic signal. All such methods require that you already know the phylogenetic relationships between the species that compose your data and any errors in the assumed phylogeny will be carried into the phylogenetic corrections. I won't go into the details of these methods and if you want to use these in the context of causal models then you should master this literature before proceeding.

It is possible to conduct an SEM on data that does not have any phylogenetic signal by working with the residuals of the PICs or the phylogenetic generalised least squares. In the context of piecewise SEM, you could simply test the predicted independencies in the union basis set using the PIC or PGLS regressions directly, although you would have to do this yourself this cannot be done in the pwSEM package; see (von Hardenberg and Gonzalez-Voyer 2012). In the context of covariance-based SEM, you would input the residuals of the PICs or the phylogenetic generalised least squares, rather than the actual values, of the variables, into lavaan.

Ecological samples taken close to one another in space tend to be more similar that samples taken farther away. This is like phylogenetically "close" (i.e. recently diverged) species that tend to have trait values that are more similar to each other than species that are phylogenetically "farther away". Ecologists sometimes wish to remove the geographic signal from their data before continuing analysis (Legendre and Fortin 1989). Lamb et al. (2014) called this "spatially explicit" SEM and reviewed existing methods. The method that they propose is to obtain a

series of variance-covariance matrices calculated over a range of different distances. Since this is a rather specialized topic, I won't go into details here, but the entire method, including R code, is given in Lamb et al. (2014).

5

Statistical power, AIC statistics and equivalent models

Your causal model has not been rejected. What do you do with this conclusion? The purpose of this chapter is to answer this question but, to do this, we need to talk about the concept of statistical power, the use of AIC (Akaike Information Criterion) statistics, and the notion of an "equivalent" model.

Before we can talk about the first topic (statistical power) I need to review some basic notions about hypothesis testing. When I tell this to my students, they roll their eyes and tell me that they already know about this from their introductory statistics course. However, in my experience, even quite advanced users of statistical methods can get confused because the underlying logic of null hypothesis testing is counterintuitive to most people. If you are confident that you have mastered to concept of statistical power, then you can skip to section 5.2.

5.1 The concept of statistical power

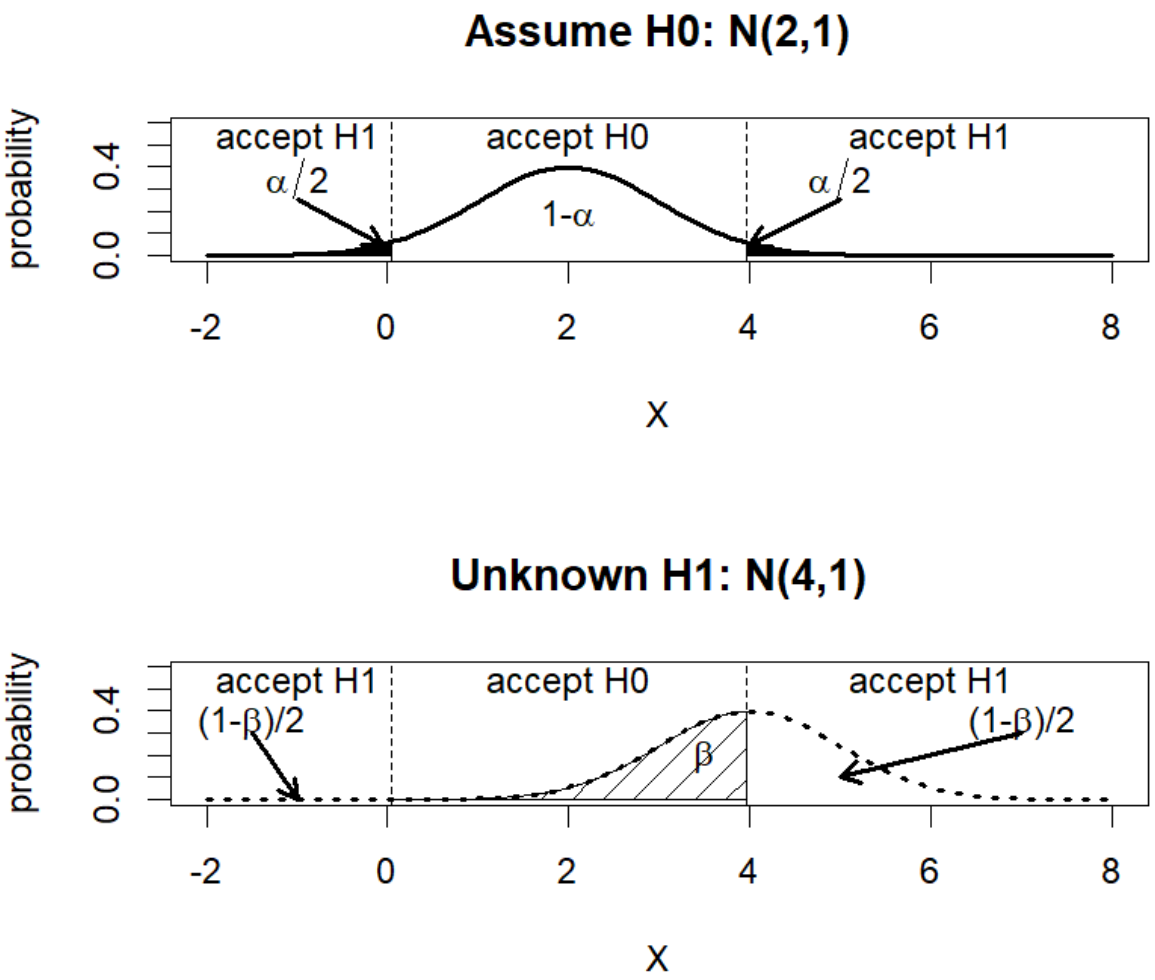


Figure 5.1. The top panel shows a normal probability density assuming the null hypothesis that the data were generated from a normal probability density whose mean is 2 and whose standard deviation is 1. The chosen significance level (α) fixes the range of values of X that would not be rejected. The bottom panel shows the probability of observing different values of X if (unknown to us) the data were actually generated from a normal probability density whose mean is 4 and whose standard deviation is 1. β is the probability according to this alternative hypothesis of observing X within the same range of values that would not result in rejecting the null hypothesis.

Let's review again the basic logic used in rejecting a two-tailed null hypothesis. This is illustrated in the top panel of Figure 5.1. For example, we could define some variable attribute, X, and then hypothesize that our data follow a normal distribution whose mean value (μ) is 2.0 and whose standard deviation (σ) is 1.0. When we do this, we are imagining a hypothetical

statistical population, written $N(\mu=2.0, \sigma=1.0)$, from which a random observation of X will be taken. This hypothetical statistical population defines our null hypothesis (H_0), which we are assuming is true until we see convincing evidence otherwise. We then observe a value of $X=x_i$ in Nature. If X was not generated according to our null hypothesis, then x_i will be rather different from the typical values (here, around 2) in our imagined statistical population. On the other hand, we also know that the value of x_i will be different in each random sample that we could draw even if it does come from our assumed statistical population. We therefore expect our value of x_i to be different from 2 because of random sampling variation. Therefore, there are always two competing explanations for why a sampled value is different from the typical value in the statistical population:

i. The sample did come from the hypothesized statistical population (i.e. our null hypothesis is correct), but sampling variation caused the difference; call this the "null hypothesis" H_0.
ii. The sample did not come from the hypothesized statistical population (i.e. our null hypothesis is wrong); call this the "alternative hypothesis" H_1.

Any value of X that is different from 2.0 should make us doubt H_0 to some degree – and therefore cause us to consider our alternative hypothesis (H_1) to some degree – but our amount of doubt[118] will increase as the first explanation above becomes less and less likely. In order to quantify this level of doubt, we need to know how likely we will observe a given difference between x_i and μ when this difference really is only caused by natural sampling variation, i.e. when our null hypothesis is correct. As the probability of a particular difference between x_i and the expected value (μ) from the null hypothesis gets smaller, two things happen: (a) the greater our doubt that this amount of difference is due only to sampling variation and (b) the more we are willing to reject this explanation in favour of its contrary; namely that our null hypothesis is wrong and that the sample didn't come from the hypothesized statistical population. In the end, we need to make a decision concerning the statistical population that generated our data (which we can never observe) based on the limited and uncertain information provided by our random sample. We need to decide how small the probability (p) of a particular difference needs to be

[118] People who are familiar with Bayesian statistics will know that the above explanation in incomplete. In order to quantify the level of doubt, we also need to take into account the prior probabilities that we assign to each competing hypothesis.

when assuming that our null hypothesis is true before our level of doubt is large enough to make us reject our null hypothesis. The value of p (the null probability) that makes us reject the null hypothesis is called the significance level (α). Thus, we provisionally accept our null hypothesis if $p>\alpha$ and reject our null hypothesis (i.e. accept the alternative hypothesis) if $p \leq \alpha$.

If we decide to reject our null hypothesis whenever the null probability is less than our significance level (α), then we have also accepted that we will incorrectly reject our null hypothesis 100α% of the time. After all, as the top panel of Figure 5.1 shows, it is still possible to observe a value of $X=x_i$ even if the probability of this occurring is less than α. In fact, it is still possible to observe *any* value of x_i no matter how far x_i is from μ – it is just very unlikely. Statistics texts call this a "type I" error: the error of incorrectly rejecting the null hypothesis when it is true. So, what value should we choose for our significance level? The traditional value of α in pure science is one error in twenty (i.e. 0.05) and a brief history of this value was given in section 2.9. Naturally, we want to reduce our chances of making type I errors. Why not choose a very small value of α; perhaps $\alpha=0.000001$? Isn't an error rate of one in a million (0.000001) better than an error rate of (say) one in twenty (0.05)?

The problem with choosing a very small significance level is that a type I error (rejecting H_0 when it is true) is not the only mistake that we can make. We can also fail to reject our null hypothesis even when it is false; that is, when the alternative hypothesis is actually the correct one. This second type of error is called a type II error, whose probability is β. Remember: we can never actually know for certain if our null hypothesis, some other hypothesis, is correct; if we did know this then we wouldn't need any statistical test! The type II error is shown in the bottom panel of Figure 5.1. In this scenario, and unknown to us (remember, we are assuming the null hypothesis), the value of X was not generated by the distribution assumed by our null hypothesis ($\mu=2$) but, rather, was generated from a normal distribution whose mean is 4. If we had known about this alternative hypothesis then we would have found that the probability of observing X within the range of values that caused us to incorrectly accept the null hypothesis (shown by the vertical dotted lines) was equal to β. The values of X that mark the boundary between those values that cause us to accept the null hypothesis and those than cause us to reject it (the confidence intervals) are 0.04 and 3.96 in Figure 5.1 because $\alpha=0.05$. In Figure 5.1 we

see that the probability of X being within these confidence intervals is 0.95 (i.e. 1-α) according to the null hypothesis (top panel) but the probability that we would observe the same thing according to the alternative hypothesis (β) is 0.5 (bottom panel). To see how changing α (the chance of making a type I error) will affect β (the chance of making a type II error), imagine changing the positions of the two vertical dotted lines in Figure 5.1. If we choose a larger value for α (say, α=0.2) then we bring these two dotted lines closer to X=2. This simultaneously increases to 20% (i.e. α=0.2) the chance of falsely rejecting the null hypothesis even if it is true (the type I error) and decreases to about 23% the chance of falsely failing to reject the null hypothesis even if it is false (type II error). We have reduced our Type II error rate by increasing our Type I error rate! If we choose a smaller value for α (say, α=0.01) then we move the two dotted lines in Figure 5.1 further from X=2. This simultaneously decreases the probability of falsely rejecting the null hypothesis even if it is true (type I error) to 0.01 and increases the probability of falsely failing to reject the null hypothesis even if it is false (type II error) to 69%. We have reduced our type I error rate by increasing our type II errorrate! There is no free lunch[119]. We can only reduce our chance of making a type I error by increasing our chance of making a type II error, and *vice vera*.

The problem of balancing type I and type II errors is even more complicated than is shown in Figure 5.1. If Figure 5.1 was the end of the story, and if we had no *a priori* preference for H_0 over H_1, then we could simply choose a value of α so that we have an equal chance of committing either a type I or type II error. Unfortunately, and unlike in Figure 5.1, we almost never have only two competing hypotheses (H_0 and H_1). After all, if the null hypothesis is wrong, then our value of X could have come from an infinite number of different Normal distributions – all of them except for N(2,1). The alternative "hypothesis" is actually a composite of all of these possible alternatives. For instance, if the alternative hypothesis in the second panel of Figure 5.1 was N(8,1) (i.e. a theoretical mean of 8) then the curve would be twice as far to the right and the probability of X being within our confidence interval (i.e. β) would be very small; the type II error rate would be very small. However, if the alternative

[119] At a fixed sample size!

hypothesis was N(2.1,1) then the second curve would be almost superimposed on our null hypothesis and the type II error (β) would be almost equal to 1-α.

The scenarios that we have been imagining so far in Figure 5.1 are not typical of most statistical analyses in one important way: we rarely take only one observation of X for our random sample. Typically, we take a sample of N observations and then compare some estimate of a parameter averaged over these observations. Whenever you estimate a sample mean, or a sample correlation, or a sample regression slope, you are doing this. The variation of such average estimates (the standard error) decreases as the sample size (N) increases. In this case it is the standard error of the estimate, not the standard deviation of the individual observations, that is important in determining the spread of the curves in Figure 5.1. The larger the sample size, the smaller the standard errors, the narrower the spread of the curves and the less overlap between the curves in the top and bottom panels of Figure 5.1. Therefore, the best way[120] of reducing both our type I and type II error rates is to increase the sample size. This statement, found in most statistics texts, is rather less useful than it appears. I have rarely come across a researcher who purposely chose to take fewer samples than their resources permitted! Sample sizes are usually fixed at the largest value that the researcher can practically obtain. Given a fixed sample size, it is important to consider the "power" of a statistical test.

The power of a statistical test is the probability of correctly rejecting the null hypothesis when it is false, and correctly not rejecting it when it is true (i.e. when the alternative hypothesis is false). The power of a statistical test is 1-β in Figure 5.1. Given a fixed sample size, each possible alternative hypothesis has its own particular level of power. Graphing the relationship between different possible alternative hypotheses against 1-β, and how this relationship changes for different possible sample sizes, is called a power curve. Figure 5.2 shows a graph of power curves for a simple DAG. Here, I have assumed that the DAG H_0 is true and that we have tested it using a dsep test. H_0 assumes that there is no direct effect of X_1 on X_3. If we were to test this DAG, then the alternative hypothesis is simply that the data weren't generated according to the DAG in H_0. In fact, unknown to us, the data were generated by the H_1 DAG in which there is a direct effect of X_1 on X_3, but the strength of this effect differs from 0 (i.e. identical to H_0) to 0.5

[120] Different statistical tests can have different power properties. By choosing tests having more statistical power, you can also reduce type II error rates as well. For instance, non-parametric tests often have less power than parametric tests when the assumptions of the parametric tests are valid.

(the missing effect is equally strong as the assumed direct effects in H_0). The results shown in Figure 5.2 are based on 1000 simulations and, in each simulation, I have generated a data set of either 20, 50, 100, 500 or 1000 observations.

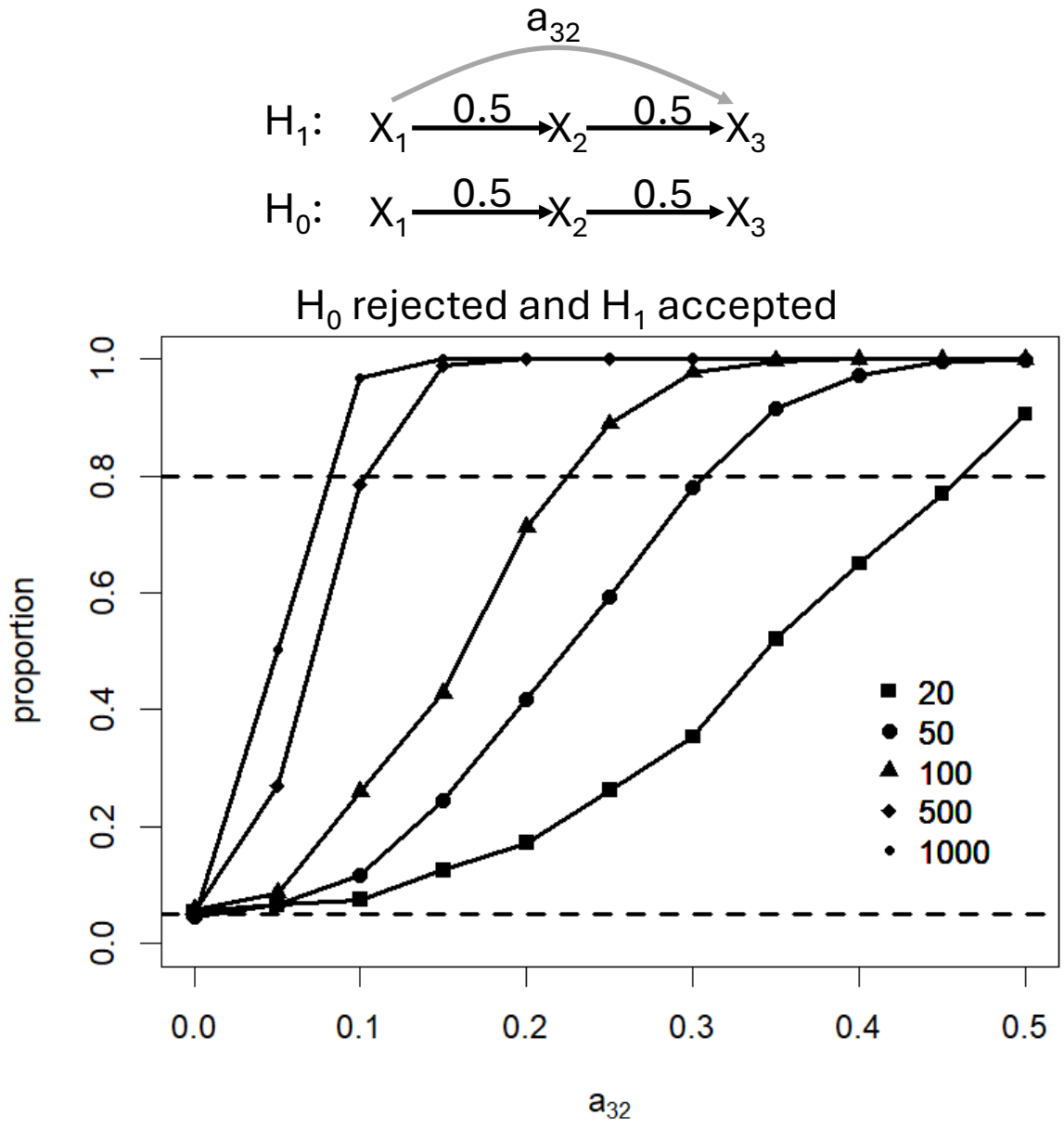


Figure 5.2. Power curves for the DAG assumed by the null hypothesis (H_0) against the alternative DAG (H_1) for different sample sizes. The lower dotted horizontal line shows the 5% significance level (α=0.05).

Figure 5.2 shows three important results. Look first at the case in which the value of the missing path coefficient (a_{32}) is zero, as the null hypothesis assumes. A proportion of ~5% of the 1000 simulations incorrectly rejects the null hypothesis (the type I error). This is because I chose an α=0.05 significance level for the dsep test. Notice too, that this type I error rate does not depend on the sample size. This is the first important point: *type I error rates are independent of sample size*. If you have reject your null DAG then you can be confident, with a probability of α, of making a type I error.

In an ideal world, as soon as the null hypothesis is wrong (here, as soon as a_{32} is not equal to zero), then we would always reject the H_0 DAG and always accept the alternative H_1 DAG. In the real world, this never happens. Just as there is always a certain risk of making a type I error when the null hypothesis is true, there is always a risk of making a type II error when the null hypothesis is false. This type II risk decreases as the null DAG becomes further from the truth – here, when the value of the missing a_{32} path coefficient becomes further from a_{32}=0 – but the type II risk also decreases as the sample size increases. For instance, using a very small sample size (N=20), the incorrect null DAG is rejected, and the correct alternative DAG is accepted, only about 8% of the time when the missing path coefficient is weak (a_{32}=0.1). In fact, the missing path coefficient needs to be almost 0.5 (the same strength as the path coefficients in the null DAG) before we can correctly reject H_0 and accept H_1 at least 80% of the time given only 20 observations! This is the second important point: *the power to correctly reject H_0 and to correctly accept H_1 depends on how "wrong" H_0 is relative to H_1*. If you have not rejected the H_0 DAG then you can be sure that there are other alternative DAGs that would also not be rejected if only you had tested them as long as these alternative DAGs are not "too" wrong.

The third important point that is shown in Figure 5.2 is that *the risk of committing a type II error decreases with increasing sample size*. With a sample size of 50, there is an 80% chance of rejecting the null DAG and accepting the correct alternative DAG as soon as a_{32} equals about 0.28 (i.e. there is 80% power when a_{32}=0.28). With a sample size of 100, there is 80% power as soon as $a_{32} \approx 0.23$ and with a sample size of 1000 there is 80% power as soon as $a_{32} \approx 0.05$.

5.2 AIC statistics in SEM

I began this chapter by asking what to conclude if your causal model has not been rejected. Because the statistical power to reject a causal graph when it is false depends on both sample size and how "wrong" the alternative graph is, we know that if we fail to reject a hypothesized causal graph, there will always be other causal graphs that would also not have been rejected if only we had tested them. Indeed, because our causal understanding of biological phenomena is usually incomplete, there are often different causal explanations (causal graphs) that could be proposed based on different biological hypotheses. You should always be on the lookout for such alternative causal explanations because they represent the best practice of "strong inference" (Platt 1964). If you find more than one causal graph that is not rejected, then it is possible to rank these competing causal graphs using Akaike's Information Criterion (AIC). Given the omnipresence of type II errors, it is virtually assured that such alternative non-rejected causal models exist! It may be that some of these competing causal models, although not rejected based on your chosen significance level can be much less supported by the data than others; if so, then we can confidently exclude them. AIC statistics were popularized for biologists by Burnham and Anderson (2002), who give a good explanation of the theoretical justification and practical use of these statistical methods.

Imagine, in an ideal world, that we knew the true process in nature that generates our observations. This true generating process produces the true probability distribution (call it f(x)) over the variables involved in the generating process. We also have a probability distribution generated by our statistical model (call in g(x)). In the context of Information Theory, a probability distribution encodes the information content of the generating process. The Kullback-Leibler information[121] (or distance[122]) function (Kullback 1959) measures the amount of information that is lost when approximating f(x) by g(x). If we had different causal models

[121] $I(f,g) = \int f(x) \ln\left(\frac{f(x)}{g(x \mid \theta)}\right) dx$ for continuous distributions and $I(f,g) = \sum_{i=1}^{k} p_i \ln \frac{p_i}{g_i}$ for discrete distributions. This is sometimes called the relative entropy of g.

[122] The term "distance" is not strictly correct in a mathematical sense because the Kullback-Leibler function only measures the distance from f(x) to g(x) but not the contrary; a true distance measure is symmetrical. It is sometimes called a directed distance.

(i.e. different causal graphs generating different probability distributions, $g_i(x)$) then we would choose the model that is the closest to the true probability distribution.

Of course, we never know the true probability distribution. If we did, then we wouldn't need statistical tests. However, Hirotugu AkaikeAkaike (1973) did the next best thing. He showed that, when comparing between different statistical models, we can obtain the *relative* Kullback-Leibler distance of each model to the truth without knowing[123] the true probability distribution (i.e. f(x)). Using the relative Kullback-Leibler distance, he proposed the AIC[124] statistic (Equation 5.1a,b). The AIC statistic is based on the maximum likelihood of the statistical model, $\mathcal{L}\left(\hat{\theta} \mid x, g_i\right)$, and on the number of free parameters (K) that must be estimated by the data in order to maximize the likelihood. Maximum likelihood was explained in Chapter 4. Equation 5.1b (Sugiura 1978) is called the second-order (or bias-corrected) AIC statistic, which corrects for bias in the original AIC when the model is based on relatively few observations. "Relatively few" observations (N) means N<40-K, where K is the number of free parameters to be estimated in the model (Burnham and Anderson 2002)). Since AIC_C approaches AIC as the sample size increases, it is best to always use the bias-corrected version.

$$AIC = -2\ln \mathcal{L}\left(\hat{\theta} \mid x, g_i\right) + 2K$$

$$AIC_C = -2\ln \mathcal{L}\left(\hat{\theta} \mid x, g_i\right) + 2\frac{N}{N-K-1} \qquad \text{Equation 5.1a,b,c}$$

$$AIC_c = AIC + \frac{2K(K+1)}{N-K-1}$$

The absolute value of an AIC statistic has no meaning[125]. Whether an AIC value is -123 or +123 tells us nothing. However, given a set of competing models, smaller AIC values identify those models in the set that are closer to the probability distribution generated by the true, but unknown, causal generating process. AIC statistics are therefore used to compare between different hypothesized models and only the difference in AIC values between the models has

[123] This is because the integral $\int f(x)\ln f(x)dx$ in footnote 4 appears in both $g_i(x)$ and $g_j(x)$ (the two competing approximating models) and so drops out.

[124] Akaike called this the AIC statistic as an acronym for An Information Criterion (Burnham and Anderson 2002) but many people use AIC as an acronym for Akaike's Information Criterion.

[125] Because the AIC is related to the *relative*, not the *absolute*, Kullback-Leibler distance of different approximating models (g_i) to the true (unknown) probability distribution.

meaning. Given a set of competing models, the difference between the AIC of model i (g_i) and the model having the smallest AIC value (g_*) in the set is $\Delta AIC_i = AIC_i - AIC_{min}$, where AIC_{min} is the model in the set (g_*) having the smallest AIC value. The bias-corrected version (AIC_C) is used in the same way, but you cannot mix together AIC and AIC_C statistics in the comparison.

Note that, since the absolute value of AIC has no meaning, there is no way of knowing how close *any* of your competing causal models are to the true generating process in Nature. We can only know how much *better* one causal model is relative to another[126]. If all of the competing causal models do a poor job of representing the true unknown generating process in Nature, then ΔAIC is only telling you which of these poor models is less poor than the others! The usefulness of AIC statistics therefore depends entirely on *how you choose your competing models*. While Burnham and Anderson (2002) point this out, they are very unhelpful in guiding us about how to decide which models to include in our set of competing models. On page 2 they state: "A philosophy of thoughtful, science-based *a priori* modeling is advocated... Science and biology play a lead role in this *a priori* model building and careful consideration of the problem." Later, on page 17, they state that the candidate models should reflect "…causal mechanisms thought likely, based on *the science of the situation*". In this book, I am advocating for only using AIC statistics to compare between different causal models that have not already been rejected based on whatever level of significance you have chosen. After all, if a causal graph has a null probability that is less than your significance level (say, $p<0.05$), then you have already decided that this causal graph is too unlikely to reflect the "causal mechanisms thought likely, *based on the science of the situation*". At the very least, one of the competing causal models should not have already been rejected based on your chosen significance level; otherwise, you are trying to choose between a set of competing causal models that you have already decided are all too unlikely to have generated your observed data.

5.3 Calculating AIC statistics in SEM

[126] Since the AIC statistic is based on sample data, the AIC statistic is only a sample estimate. This means that if we took another random sample of data and fitted our competing models, the AIC statistics of these competing models would also randomly differ from those obtained from the first set of data (Preacher and Merkle 2012)

Calculating an AIC statistic in covariance-based SEM is quite straightforward. Covariance-based SEM assumes a multivariate normal distribution and this distribution is fixed by choosing values for the K free parameters in the structural equations by maximizing the log-likelihood (Chapter 4). Therefore, we have only to get the log-likelihood values given this multivariate normal distribution and then sum them. Once you fit a model with the `sem()` function of lavaan and save the object (let's call it `fit`), you can extract the AIC values using `AIC(fit)`. There is no function in lavaan to calculate the bias-corrected AIC but you can easily get this using Equation 5.1c.

Piecewise SEM requires only a bit more work. The original piecewise SEM AIC statistic, proposed by Shipley (2013), was based on Fisher's C statistic and the number of free parameters that must be estimated to fit the structural equations (Equation 5.2). Because it is based on the C statistic, which is based on the d-separation claims in the union basis set, this means that this version of AIC only considers the causal topology of the model (how the variables link together) but completely ignores the other statistical assumptions included in the actual structural equations.

$$AIC = C + 2K$$

$$AIC_C = C + 2K\frac{N}{N-K-1} \qquad \text{Equation 5.2a,b}$$

However, in 2020 Bob Douma and I derived a simpler AIC statistic for piecewise SEM (Shipley and Douma 2020b) that incorporates both the causal topology of the model and the statistical fit of the structural equations (Equation 5.3). Moreover, it easily generalizes to any structural equation, in the form of a regression, that is obtained from maximum likelihood. Such regressions include linear, generalized linear, mixed, generalized mixed, generalized additive or generalized mixed additive regressions. Therefore, I recommend that you use Equation 5.3 rather than Equation 5.2. Given V variables, it turns out that the AIC for the full set of structural equations has a particularly simple a pleasing form: it is equal to the sum of the AIC values for each of the structural equations (i.e. regression equations) associated with each of the V

variables. Note that the exogenous variables also have to be included in this calculation in the form of regressions on the intercept only[127].

$$AIC = \sum_{i=1}^{V} AIC_i$$
$$AIC_C = \sum_{i=1}^{V} AIC_{Ci}$$

Equation 5.3a,b

Equation 5.3 is all you need to use AIC statistics when doing piecewise SEM. If you aren't interested in why Equation 5.3 is valid then move on the section 5.4. If you want to understand why Equation 5.3 is valid, then read on. What follows is valid for DAGs. In Chapter 6 I will introduce mixed acyclic graphs (MAGs); MAGs are appropriate for path models that include implicit latent variables that are represented as correlated errors. I will postpone how to calculate AIC statistics for MAGs in Chapter 6 and you will see that the AIC statistic for MAGs is a generalization of what follows.

You will remember, from Chapter 2 (for example, Equation 2.4), that a DAG generates the multivariate probability distribution of observations that come from this DAG. Importantly, this multivariate probability distribution can be expressed as the product of a series of univariate conditional probability distributions in which each univariate probability distribution is conditioned on its parents. This is important because we can easily get each conditional univariate probability directly from the regressions making up the piecewise structural equations. Given a DAG with variables X_1, X_2 … X_n, and letting **pa**(X_i) denoting the causal parents of variable X_i in the DAG, the decomposition of the multivariate probability distribution is:

$$p(X_1, X_2, \ldots, X_n) = p(X_1 \mid \mathbf{pa}(X_1))\, p(X_2 \mid \mathbf{pa}(X_2) \cdots p(X_n \mid \mathbf{pa}(X_n))$$

You will also remember, from Chapter 4, that a probability density function (p) and a likelihood function ($\mathfrak{L}$) are really the same function except that the random and fixed variables in this common function are switched. Thus, if we know that our next observation comes from (say) a normal probability density whose mean and standard deviation are t μ=2 and σ=1, but we don't yet know what value this observation (X) will take, then μ and σ are "fixed" rather than random

[127] If X_i is an exogenous variable, then the regression is X_i~1

(because we already know their values) but X is random, not fixed (because we don't yet know its value); that is why we write the probability density as p(X|μ,σ) where (|) means "given". If, on the other hand, we have already observed the value of X and know that it follows a normal probability, then X is "fixed" rather than random[128]. However, we don't know *which* normal probability X has generated this value of X (i.e. we don't know the values of μ and σ), and so μ and σ are now "random" rather than "fixed". In this case, our function is called a likelihood function (ℒ) and so we write it as ℒ(μ,σ|X).

$$p(X \mid \mu,\sigma) = \frac{1}{2\pi\sigma} e^{\frac{-(X-\mu)^2}{2\sigma}}$$

$$\mathcal{L}(\mu,\sigma \mid X) = \frac{1}{2\pi\sigma} e^{\frac{-(X-\mu)^2}{2\sigma}}$$

Since the multivariate probability density function and the multivariate likelihood function are really the same function with the random and fixed components switched then, if the data were generated by our DAG, we can also decompose the multivariate likelihood function as the product of a series of univariate conditional likelihood functions in which each univariate likelihood distribution is conditioned on its parents. Taking logarithms, we get the expression for the log-likelihood (ℓℓ):

$$\mathcal{L}(X_1, X_2, \ldots, X_n) = \mathcal{L}(X_1 \mid \mathbf{pa}(X_1))\,\mathcal{L}(X_2 \mid \mathbf{pa}(X_2) \cdots \mathcal{L}(X_n \mid \mathbf{pa}(X_n))$$

$$\ell\ell(X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} \ell\ell(X_i \mid \mathbf{pa}(X_1))$$

Notice that the expression $\ell\ell(X_i \mid \mathbf{pa}(X_1))$ is simply the log-likelihood of variable X_i given its parents, which is obtained from a regression[129] of X_i on its parents (one of the piecewise structural equations). This regression will involve K_i free parameters that have to be estimated. Therefore, the AIC statistic for this particular regression is $-2\ell\ell(X_i \mid \mathbf{pa}(X_1)) + 2K_i$ and the AIC statistic for all of the structural equations (thus piecewise regressions) is:

[128] After all, the value of X is now written down in your notebook or in your spreadsheet, and it won't change.
[129] In R, you can get the log-likelihood of any regression using `logLik(fit)`, where "fit" is the object generated by a regression.

$$AIC = -2\sum_{i=1}^{n} \mathcal{LL}\left(X_i \mid \mathbf{pa}(X_1)\right) + 2\sum_{i=1}^{n} K_i = \sum_{i=1}^{n} AIC_i$$

5.4 Interpreting AIC statistics

We have found a set of competing causal models, we have identified the best model in our set of competing models (the one with the smallest AIC value), and we have calculated by how much larger each of the other competing models to this best model is (the ΔAIC values). Which of these competing models do we keep as viable contenders, and which models can we confidently reject ignore?

The magnitude of ΔAIC_i measures by how much further model g_i is from the true (unknown) probability distribution than is the model having AIC_{min}, i.e. g_*. There is a relationship between the magnitude of ΔAIC_i and the likelihood of the model g_i given the observed data (Akaike 1983). $\mathcal{L}(g_i \mid x)$ (Equation 5.4) is the likelihood of model g_i in a set of competing models, given the observations (x), and it represents the relative strength of evidence for this model relative to the best model in the set of competing models (i.e. the one with the lowest AIC value). The larger the ΔAIC of model g_i, the smaller its likelihood, and the less evidence that g_i is actually closer to probability distribution of the true (unknown) generating process in Nature. Just like ΔAIC, $\mathcal{L}(g_i \mid x)$ does *not* measure the strength of the evidence provided by the data for the true unknown process that actually generated the data; it only measures the strength of the evidence in favour of model g_i relative to g_*.

$$\mathcal{L}(g_i \mid x) \propto e^{\frac{-\Delta_i}{2}} \qquad \text{Equation 5.4}$$

In order to interpret $\mathcal{L}(g_i \mid x)$ (Equation 5.4), I find it useful to use likelihood[130] ratios. Royall (1997) provides a complete explanation and justification for likelihood ratios in statistical inference. A likelihood ratio between any two models is the ratio of the likelihoods of each model. The best model in the set (the one with the smallest AIC value) always has ΔAIC=0. The evidence in favour of this best model relative to itself (Equation 5.2) is always $e^0 = 1$. The evidence in favour of this best model relative to model g_i in our candidate set is the likelihood of model g_i relative to the best model. For instance, if model i has a value of Δ_i=2 then the likelihood ratio in favour of the best model is $e^{\frac{-0}{2}} / e^{\frac{-2}{2}} \approx 2.7$. That means it is 2.7 times more likely that the best model in the set is closer to the true unknown generating process than is model g_i. If we let g_* be the best model in our set (i.e. the one with the smallest AIC value), the likelihood ratio (LR) between this best model and any other model in the set is given in Equation 5.5.

$$LR = \frac{\mathcal{L}(g_* \mid x)}{\mathcal{L}(g_i \mid x)} = e^{0.5\Delta_i} \qquad \text{Equation 5.5}$$

We are now talking about making inferences concerning competing hypotheses (i.e. causal models) given the data rather than about inferences concerning the data, given the hypotheses. Some readers might wonder if these ideas can be cast in a Bayesian context. Bayes' Theorem is a fundamental concept in probability theory that describes how to update the probability of a given hypothesis (H_i) based on new evidence or data (x) and Bayes' Theorem makes use of the likelihood function. In the context of SEM, a "hypothesis" is synonymous with a causal model. Bayes' Theorem is a function of a "prior" probability, a likelihood and a posterior probability. The prior probability is the probability that we ascribe to the hypothesis before we observe the data; this is denoted $p(H_i)$. In the context of SEM and specifically with reference to the ΔAIC statistic, we are comparing the "best model" (g_*, i.e. the one with the smallest AIC value) to the competing model g_i. The prior probability of a causal model is the probability that you assign to a causal model before fitting it, and this based only on what you know from previous biological knowledge. Again, in the context of SEM and specifically with reference to the ΔAIC statistic, you would assign this prior probability to each of the two competing models (g_* and g_i) such that

[130] Also called evidence ratios.

the two prior probabilities sum to 1. This is because there are only two possible alternative hypotheses: either g_* is closer to the true (unknown) generating probability distribution or else g_i is closer to the true (unknown) generating probability distribution. Assigning prior probabilities to hypotheses is a complicated, and somewhat controversial, topic that I don't want to discuss here. However, if you don't have any *a priori* reason to favour g_* over g_i before you have looked at the data, then the maximally uninformative prior for each causal model is 1/2. The probability that we ascribe to the hypothesis after we have observed the data (after we have fit the causal model to the data) is called the posterior probability and is denoted $p(H_i|x)$. Note the difference in notation: $p(H_i)$ is the probability that you assign to the truth of hypothesis i before you see the data (i.e. the prior) while $p(H_i|x)$ is the probability that you assign to the truth of hypothesis i after you see the data (i.e. x), which is the posterior. The likelihood (written $p(x|H_i)$ or $\mathcal{L}(x|H_i)$), as you already know, is the probability of observing the data assuming that the hypothesis (the causal model) is true. As Royall (1997, pages 10-11) points out, algebraic manipulation[131] of Bayes' Theorem shows that observations with a likelihood ratio of LR are evidence strong enough to increase or decrease the prior probabilities by LR times. The actual values of the prior probabilities don't matter, nor does their ratio. It doesn't even matter if we know the values of the prior probabilities. Given Equation 5.5, a likelihood ratio of LR means it is LR times more likely that the best model in the set is closer to the true unknown generating process than is model g_i. By combining Equation 5.5 with Bayes' Theorem, and the fact that we are in a situation in which there are only two mutually exclusive hypotheses (either g_* is closer to the true distribution or else g_i is so that $p(g_*|x) + p(g_i|x) = 1$), gives Equation 5.6. Equation 5.6 gives the posterior probability that the causal model g_* is closer to the true (unknown) generating process than is the causal model g_i. Figure 5.3 shows the relationship between the posterior probability of g_* after testing g_* and g_i (with a value of ΔAIC_i) against the data if we thought that both models (g_* and g_i) were equally likely to be the best one before we tested them g_i.

$$p(H_i|x) = \frac{p(x|H_i)\,p(H_i)}{\sum_i p(x|H_i)\,p(H_i)} = \frac{\mathcal{L}(x|H_i)\,p(H_i)}{\sum_i \mathcal{L}(x|H_i)\,p(H_i)}$$

[131]

$$\frac{p(H_i|x)}{p(H_j|x)} = \left[\frac{\mathcal{L}(x|H_i)}{\mathcal{L}(x|H_j)}\right]\left[\frac{p(H_i)}{p(H_j)}\right]$$

$$p(g_* \mid x) = \frac{p(g_*)e^{\frac{\Delta_i}{2}}}{1 - p(g_*)\left(1 - e^{\frac{\Delta_i}{2}}\right)}$$

Equation 5.6

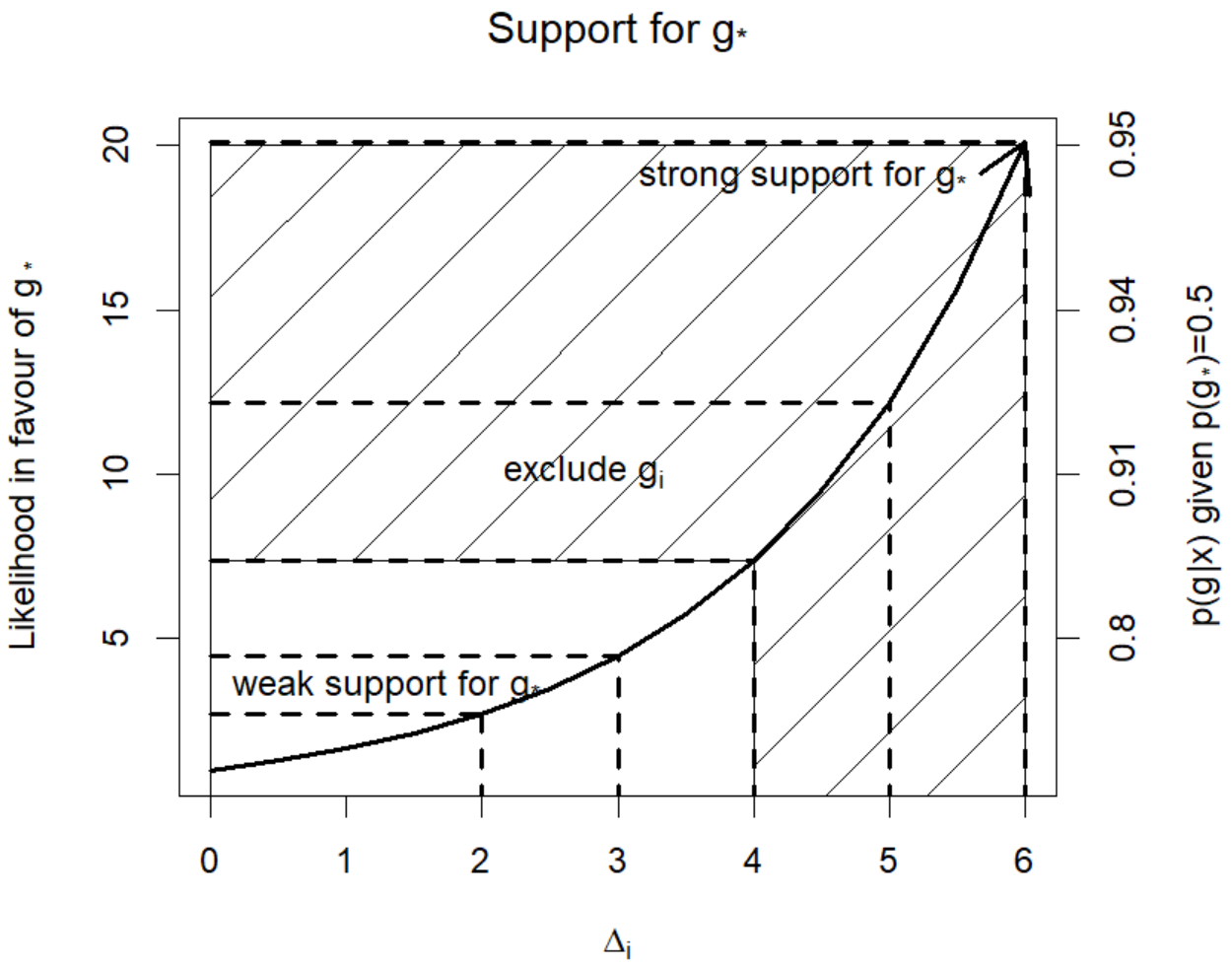


Figure 5.3 The relationship between the increase in AIC (Δ_i) between model i and the model having the lowest AIC value (g_*) and the likelihood in favour of g_*, given the data, being the model closer to the true (unknown) process generating the data. The right-hand axis shows the posterior probability that g_* is closer to the true (unknown) process generating the data assuming that the prior probabilities of g_* and g_i were equal. The hatched area shows values of ΔAIC_i that would normally result in the rejection of g_i.

How much more likely does g_* have to be relative to g_i before we can exclude g_i as a valid candidate? This is like deciding at which age a person is sufficiently mature to vote or drive a car. There are ages at which everyone agrees that the average person is *not* sufficiently mature. There are ages at which everyone agrees that the average person *is* sufficiently mature. There are ages in between in which we are not sure. Both maturity and "strength of evidence" are continuous variables. The ΔAIC statistic is used to compare two models: the one with the lowest AIC value (g_*) and another one in the set of competing models (g_i). We have to decide if g_* is so much more likely to be closer to the true (unknown) generating process in Nature than g_i that we can exclude g_i from our list of competing models.

Equation 5.6 can help guide us in this decision. To see how, consider the simplest situation in which we have these two competing causal models (g_i and g_*) and, *before we test them against the data*, we have no reason to prefer one over the other. That means their prior Bayesian probabilities are each ½ (i.e. $p(g_*)=p(g_i)=0.5$). Now, we fit these two models to the data and find that g_i is 5 AIC units larger than g_* (ΔAIC=5). From Equation 5.5, the likelihood ratio is about 12.2; i.e. the probability that g_* is closest to the true (unknown) generating process is 12.2 times the probability that g_i is closest. This means (Equation 5.6) that $p(g_*|x)\sim0.924$ and $p(g_i|x)\sim0.076$. There is a 92% chance that g_* is the model closest to the true (unknown) causal process and there is only an 8% chance that g_i is closest to the true (unknown) causal process. Are these odds[132] convincing enough for you to choose the first model? If you want g_* to be twenty times more likely than g_i then you would choose $\Delta AIC_i=5.99$, in which case $p(g_*|x)\sim0.95$ and so $p(g_i|x)\sim0.05$.

Burnham and Anderson (2002) propose that a model, g_i, whose Δ_i is less than 4 (a likelihood ratio of less than ~7.4) should be viewed as having sufficient support relative to g_* that it should remain a viable contender along with g_*. A model whose Δ_i is between 4 and 7 (LR between 7.4 and 33.1) has "considerably less support" that g_*; presumably it should be excluded unless you have some good non-statistical reason to keep it. A model whose Δ_i is greater than 10 (LR greater than 148.4) has "essentially no support". Deeks and Altman (2004), in the context of clinical tests in medicine, propose similar, but slightly different rules. Likelihood ratios of

[132] These odds would change if we changed the prior probabilities of the two models but the likelihood ratios (i.e. by how much the data have supported the first model over the second) would remain the same.

between 2 and 5 (therefore Δ_i between 1.4 and 3.2) provide "weak evidence" in favour of g_* and against g_i. Likelihood ratios of between 5 and 10 (Δ_i between 3.2 and 4.6) as providing "moderate evidence". Likelihood rations of greater than 10 (Δ_i of more than 4.6) provide "strong evidence". Similar proposed ranges can be found in other references. Combining these proposed rules, the consensus (such as it is) seems to be that if Δ_i is less than 4 (LR less than 7.4) then the data do not give enough evidence against g_i and in favour of g_* to exclude g_i while Δ_i values of greater than 4 (LR greater than 7.4) do provide enough evidence exclude model g_i. Figure 5.3 summarizes these rules.

5.5 Empirical example

A practical example will show both how to obtain AIC statistics in R and how to interpret them. Shipley and Tardif (2021) studied how certain chemical properties of dead leaves affect how rapidly they decompose. To do this, we measured the exponential decay constants (k values) of four chemical fractions of mixed-species leaf litter under controlled conditions. These chemical fractions were water-soluble carbon, cellulose, hemicellulose and lignin. The exponential rates of decay of these four fractions were represented by the variables k_S, k_C, k_H and k_L. The data set consisted of these four variables measured on 48 different mixtures of dead leaves of five species of trees typical of Southeastern Canada. Notice that 48 lines of data do not provide much statistical power against type II errors and yet these 48 observations represented almost two years of work. This is a good practical example of how resources can place strong limits on statistical power. Antoine Tardif and I originally planned the project to test between two alternative hypotheses that had been proposed in the literature concerning how these different chemical fractions of plant tissues interact to control the decomposition of dead leaves on the forest floor. One of these proposed models is shown in Figure 5.4a. This model was rejected, based on the maximum likelihood Chi-Squared statistic. The lack of fit was entirely due to the claim by this model that k_H and k_C be independent conditional on k_S. We therefore proposed a series of alternative models that were both consistent with our biological knowledge of leaf litter

decomposition and that removed this claim. Three of these alternative models are also shown in Figure 5.4.

I will calculate the AIC statistics of these models using the lavaan and pwSEM packages. Be careful when using AIC statistics from the piecewiseSEM package. Only the likelihood version of the AIC statistic (Equation 5.X) is printed out[133] and the printed value is wrong if there are correlated errors, because the likelihood of the free covariance is not included in the calculation.

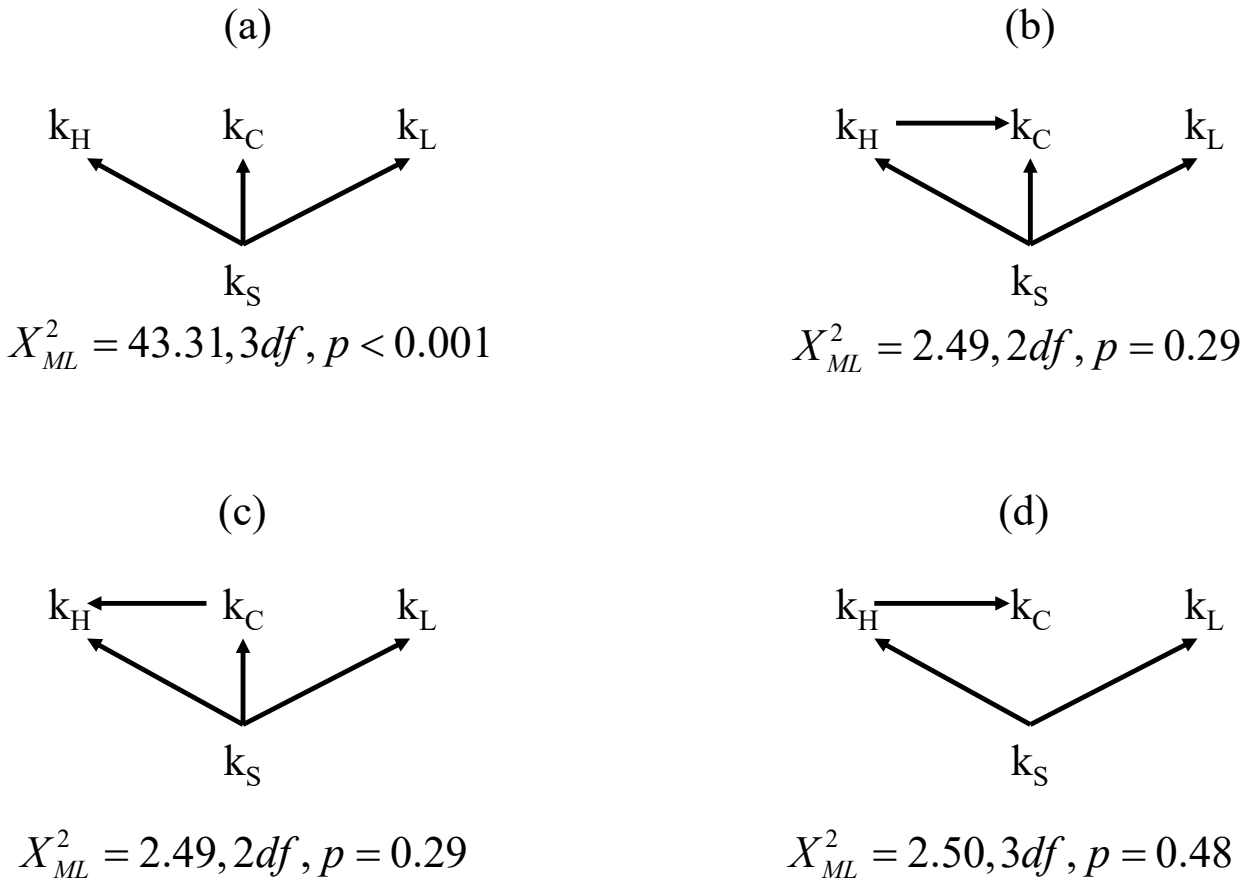


Figure 5.4. Four alternative DAGs describing hypothesized causal links between the decomposition rates in dead leaves for soluble sugars (k_S), lignin (k_L), cellulose (k_C) and hemicellulose (k_H).

Let's begin with the model in Figure 5.4a and use the lavaan package. Here is the code to fit this model; the last three lines in the model object (`Fig5.4a`) are needed to force the free

[133] The `summary()` of `psem()` has the `AIC.type=` argument with choices of "loglik" or "dsep", but choosing "dsep" returns an error message.

covariances between the terminal endogenous variables to zero, as required by the DAG, since these are freely estimated by default in lavaan:

```
Fig5.4a<-"
k.lignin~k.labile
k.cellulose~k.labile
k.hemicellulose~k.labile
k.lignin~~0*k.cellulose
k.lignin~~0*k.hemicellulose
k.hemicellulose~~0*k.cellulose
"
fit<-sem(Fig5.4a,data=Tardif,fixed.x=FALSE, meanstructure=TRUE)
AIC(fit)
```

The `fixed.x=FALSE` argument in the `sem()` function was included so that the exogenous variable is explicitly modelled and included in the AIC statistic. The exogenous variances are *not* estimated by maximum likelihood using the default configuration of lavaan and failing to do this will exclude the likelihood of the exogenous variables in the calculation of the AIC statistic. Excluding the exogenous variables during the calculation of the AIC statistic (by not including[134] the `fixed.x=FALSE` argument in the `sem()` function) is not an error but I recommend that you don't do this. The `fixed.x=FALSE` argument tells the `sem()` function to estimate the variance of the exogenous variables via maximum likelihood while `fixed.x=TRUE` (the default) argument tells the `sem()` function not to estimate the variance of the exogenous variables via maximum likelihood but rather, to simply use the sample variances of these exogenous variables. Since the AIC statistic is calculated using the log-likelihoods, the effect of `fixed.x=TRUE` (the default) is to exclude the exogenous variables when calculating the AIC values. You should always include the `fixed.x=FALSE` argument when comparing models with AIC statistics is because the variables in the alternative models might have different combinations of exogenous and endogenous variables. If this is the case, then the resulting AIC values cannot be compared if the default `fixed.x=TRUE` argument is used. By always including both exogenous and endogenous variables when using the `sem()` function by including the `fixed.x=FALSE` argument, you will avoid this mistake. Of course, you must always have the same variables in all of the competing models.

[134] The fixed.x=TRUE is the default value in `sem()`.

I also included the `meanstructure=TRUE` argument in the `sem()` function. This is not necessary, but I have included it here so that the resulting AIC statistic will be comparable with the output of the pwSEM package. The `meanstructure=TRUE` argument tells the `sem()` function to also estimate the intercepts[135] of each endogenous variable via maximum likelihood and so these estimated intercepts are included in the number of free parameters that are estimated (the K values in Equation 5.1). If you include the `meanstructure=TRUE` argument, then it must be in all of the competing models.

The AIC statistic that is output by the lavaan package is the original AIC (Equation 5.2a) not the bias-corrected version for small sample sizes. In our example, this is questionable because of the small sample size (48) relative to the number of free parameters (7). The following R function will calculate the bias-corrected AIC value (Equation 5.2b).

```
AIC.C<-function(sem.object,data){
  LL<-logLik(sem.object)
  K<-(AIC(sem.object)+2*LL)/2
  N<-dim(data)[1]
  AIC.C<--2*LL+2*K*(N/(N-K-1))
  data.frame(LL=LL,K=K,AIC.C=AIC.C)
}
```

Doing the same thing for the other three competing models in Figure 5.4 give the results shown in Table 5.1. The model with the smallest AIC is DAG (d) and so this is the DAG that is best supported amongst the competing models by the data. The DAG (a) that had been derived from pre-existing theory has a ΔAIC_C of 40.82, and so it should definitely be excluded. Since this DAG was also clearly rejected by the maximum likelihood Chi-squared statistic (X^2_{ML}=43.31, 3 degrees of freedom, p<0.0005), I would not normally even include it in the set of competing models but have done this here as an example. The other two DAGs had AIC_C values that were only 3.89 units larger than DAG (d), resulting in a likelihood ratio of only 6.0. In other words, DAG (d) is 6 times more likely than DAGs (b) or (c) to be closer to the true unknown generating process. If we accept the recommendation of Burnham and Anderson (2002) for rejecting models based on AIC values, then we would conclude that the two other DAGs cannot reasonably be excluded but that DAG (d) has moderate support relative to the other two DAGs.

[135] Remember that, by default, covariance-based SEM centres each variable around its mean before estimating the structural equations. Therefore, the intercepts are not estimated via maximum likelihood.

Some readers might notice that the second and third DAGs have the same AIC_C values[136]; I will explain why this occurs later in this chapter.

Table 5.1. The bias-corrected AIC statistics for the four alternative DAGs shown in Figure 5.4, along with the ΔAIC_C values relative to the best model, which is DAG (d).

DAG	AIC_C	ΔAIC_C
(a)	109.24	40.82
(b)	72.00	3.58
(c)	72.00	3.58
(d)	68.42	0

Next, we do the same analysis using the pwSEM package. Here is the script to fit the model in Figure 5.1a; I included the `use.permutations=TRUE` argument because of the small sample size.

```
library(pwSEM)
Fig5.1a<-list(
  gam(k.labile~1,data=Tardif),
  gam(k.lignin~k.labile,data=Tardif),
  gam(k.cellulose~k.labile,data=Tardif),
  gam(k.hemicellulose~k.labile,data=Tardif)
)
fit<-pwSEM(sem.functions=Fig5.1a,data=Tardif,use.permutations =
TRUE)
summary(fit)
```

As expected, the model is clearly rejected (C=18.53, 6 df, p=0.005). The AIC statistic is 101.90 and the bias-corrected version is 109.24. These are the same values[137] as output by lavaan because the assumptions of linearity and normality (which is what lavaan requires) are also implicit in the `gam()` regressions that I have specified. We can verify this by calculating the AIC statistic directly using Equation 5.1, since it is simply the sum of the AIC values of each of

[136] And exactly the same maximum likelihood Chi-Squared statistics and null probabilities.
[137] This is only true if you use the `fixed.x=TRUE and meanstructures=TRUE` arguments in the `sem()` function.

the four regressions given above; note that if your model has dependent errors (free covariances) then you must calculate the AIC value differently as explained in Chapter 6.

```
AIC(gam(k.labile~1,data=Tardif)) +
AIC(gam(k.lignin~k.labile,data=Tardif)) +
AIC(gam(k.cellulose~k.labile,data=Tardif)) +
AIC(gam(k.hemicellulose~k.labile,data=Tardif))
```

The pwSEM package has an advantage over lavaan, when dealing with models without explicitly modelled latent variables, when a probability distribution other than a normal distribution in more appropriate for any of the variables, or when nonlinear relationships between the variables are more appropriate. This is because the pwSEM package can accommodate these more appropriate assumptions,[138] which is reflected in the AIC statistic while the AIC output by lavaan always assumes normality and linearity.

5.6 Equivalent models

I asked, at the beginning of this chapter, what you should conclude if a DAG has not been rejected. The first part of the answer to this question is that not rejecting a model only tells you that the data have not given you sufficient reason to reject it given the statistical power available in your data. It is possible that a larger sample size might provide sufficient reason to reject it. Equally, given your current data, there are likely other DAGs that, if tested, would not be rejected either. This emphasizes the importance of looking for alternative biologically reasonable DAGs. However, given your non-rejected DAG, there are also (almost always) alternative DAGs that can never be statistically differentiated from your preferred DAG at any sample size. These are called d-separation equivalent DAGs. Such d-separation equivalent DAGs will always produce the same value for the fit statistic (i.e. the C statistic or the maximum likelihood Chi-Squared statistic), the same value for the resulting null probability, and the same value for the AIC statistic. In other words, no purely statistical evidence can differentiate

[138] With respect to the AIC statistic, the current version (1.0.0) of `pwSEM()` can only accommodate Normal, Poisson, Binomial or negative Binomial distributions.

between equivalent DAGs although they can often be differentiated based on biological knowledge.

Happily, once you have a DAG that you have provisionally accepted, there is a very simple way of finding the equivalent DAGs. Before giving you the steps to do this, you only need to understand one additional concept: the notion of an "unshielded" collider variable. You are already familiar with a collider variable along a path; it is a variable (Y) that has arrows pointing into it from both directions along the path (X→Y←Z). The collider variable Y along this path is "unshielded" if there is no arrow between X and Z. Given this definition, and given any DAG $\mathcal{G}$, here are the steps in obtaining all of the equivalent DAGs of $\mathcal{G}$. These steps are shown in Figure 5.5.

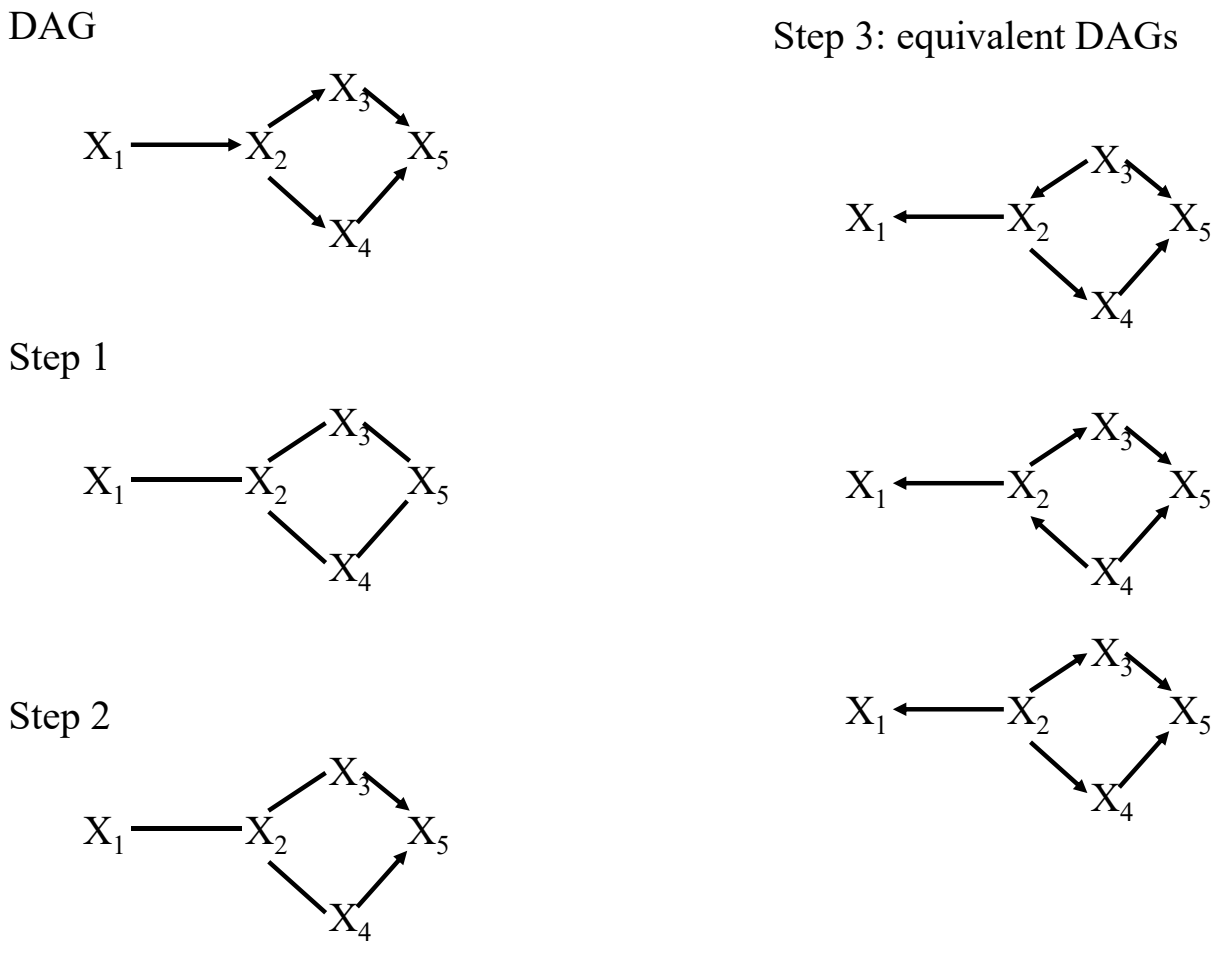


Figure 5.5. The three steps involved in obtaining d-separation equivalent DAGs.

- Step 1: Change all of the directed edges (i.e. arrows) in the DAG into lines. This is called the "skeleton" of the DAG.
- Step 2: If any triplet of variables in the original DAG forms an unshielded collider, add these to the skeleton of the DAG. This is called a partially oriented graph.
- Step 3: You can now convert the remaining lines in the partially oriented graph into arrows in any way that you like so long as (i) you do not create any new unshielded colliders and (ii) you do not create any cycles (feedback loops). The result is a new DAG that is d-separation equivalent to the original DAG.

At the end of step 3, you will have all of the d-separation equivalent DAGs to your original DAG. If you test these equivalent DAGs to the same data set then you will get exactly the same fit statistic (a C statistic for dsep tests or a maximum likelihood Chi-Squared statistic for covariance-based SEM), exactly the same null probability and exactly the same AIC statistic. If you write down the union basis sets of these DAGs, then you will see that the basis sets are identical. That is why they are called "d-separation equivalent". In other words, when you test a DAG, you are simultaneously testing all the d-separation equivalent DAGs as well.

Although there is no way of statistically differentiating equivalent DAGs (and the resulting structural equations), there are often non-statistical ways of doing this based on your biological knowledge. If there is a pattern like X_i─X_j in the partially oriented graph, then biological knowledge might tell you that the causal direction must be from X_i to X_j. Once you orient this edge as $X_i \rightarrow X_j$ at the end of step 2 then this will severely reduce the ways that the remaining lines can be oriented in step 3. For instance, if X_1 in Figure 5.5 is a variable that occurs before X_2 then you would know that a viable DAG must have $X_1 \rightarrow X_2$. Once this edge is oriented, then step 3 excludes the three equivalent DAGs.

Go back to Figure 5.4, which showed four alternative DAGs. Each is a different causal hypothesis concerning how soluble sugars, lignin, cellulose and hemicellulose causally interact during decomposition of dead leaves. If you apply the above rules to get the d-separation equivalent DAGs to DAG (b), then you will see that DAG (c) is actually equivalent to DAG (b). This is because, although there are colliders in both DAG (b) and DAG (c), they are both

shielded[139]. That is why the C statistic, null probability and AIC statistic was identical for these two DAGs.

[139] In DAG (b) there is a collision along $K_H \rightarrow K_C \leftarrow K_S$ but it is shielded by $K_S \rightarrow K_H$. In DAG (c) there is a collision along $K_C \rightarrow K_H \leftarrow K_S$ but it is shielded by $K_S \rightarrow K_C$.

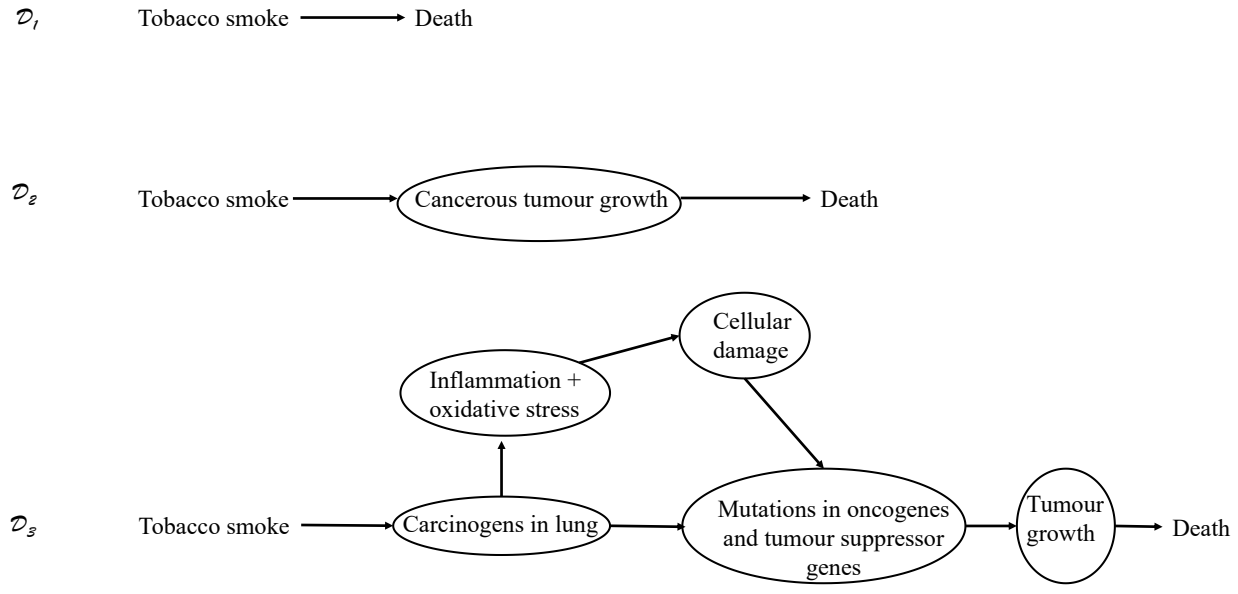
6
Piecewise SEM with implicit latent variables

You learned about "correlated errors" in Chapter 4. Correlated errors, in covariance-based path analysis, represent dependencies between pairs of observed variables (X_i, X_j) that are generated by an implicit common cause (L) of the pair of observed variables (i.e., $X_i \leftarrow L \rightarrow X_j$). The latent variable (L) is implicit because it is "missing" from the DAG. We could not model such missing common causes in piecewise SEM (Chapter 3) because dependencies generated by such implicit latents cannot be represented in a DAG. One goal of this chapter is to remove this restriction in piecewise SEM. We will do this by including the missing, or *latent*, variables into a DAG and then removing these latent variables to produce a new type of causal graph, called a mixed acyclic graph (MAG), that maintains all of the dependence and independence relationships that were implied in the original DAG. A MAG can include correlated errors without the assumptions of normality and linearity that are required in covariance-based SEM. As an added bonus, a MAG can also include another type of noncausal dependency called "selection bias" that cannot always be modelled in covariance-based SEM. However, in order to empirically test this MAG, we will have to convert it into a (possibly) different MAG, called an "m-equivalent" MAG.

This chapter leans heavily on the terminology, definitions and notions in Chapter 2. You should not read this chapter unless you have mastered Chapter 2. Much of this chapter explains the logic and justification of extending dsep tests to causal hypotheses that include correlated errors and/or selection bias. If you just want to know how to do this, without understanding why you can do this, then you only need sections 6.1, 6.2 and 6.7.

6.1 Latent variables: observable or unobservable in practice

According to the U.S. National Institutes of Health[140], tobacco use is a leading cause of death. This leads to a simple DAG, $\mathcal{D}_1$: tobacco use→death. The variable "tobacco use" is a direct cause of death in the DAG $\mathcal{D}_1$ (Figure 6.1). The U.S. National Institutes of Health is more specific: it says that tobacco use is a leading cause of many types of cancer and that cancer is a leading cause of death. Now we have a new DAG $\mathcal{D}_2$: tobacco use→cancer→death. "Tobacco use" in the $\mathcal{D}_2$ DAG is an indirect cause of death, not its direct cause. If "tobacco use" is a direct cause in $\mathcal{D}_1$ but and indirect cause in $\mathcal{D}_2$ then is $\mathcal{D}_2$ contradicting $\mathcal{D}_1$? No. "Direct" and "indirect" are descriptions about the relationships between variables in the DAG (i.e. your causal hypothesis), not about the relationships between variables in Nature. The adjectives "direct" or "indirect" are always relative descriptions; more precisely, they are relative to the other variables that are explicitly included in the DAG. To say that tobacco use is a direct cause of death (i.e. $\mathcal{D}_1$) doesn't mean that no other intervening variables exist in Nature along the directed path between them. Rather, tobacco use is a direct cause of death in DAG $\mathcal{D}_1$ because no other intervening variables exist in the causal claim (i.e., in $\mathcal{D}_1$). In fact, a more detailed causal description of the relationships between tobacco smoking, lung cancer and death is shown in the DAG $\mathcal{D}_3$, although even this DAG could be made more complex by adding additional levels of causal detail.



[140] https://www.cancer.gov/about-cancer/causes-prevention/risk/tobacco

Figure 6.1. Three different DAGs describing the hypothesized causes linking smoking and premature death. Variables within circles represent latent, or "hidden" variables relative to DAG $\mathcal{D}_1$.

The Meriam-Webster dictionary[141] defines the word "latent" as something that is "present and capable of emerging or developing but not now visible, obvious, active, or symptomatic". If you include a variable in a DAG that you have not observed or measured, then such a variable is called a "latent" variable because it is present in the DAG but is not "visible" in your data. By convention, latent variables are identified in the DAG by enclosing them in circles although I will also identify them by enclosing them in triangles; the exact meaning of the circles and triangles will be explained later in this chapter. Each of the variables enclosed in circles in Figure 6.1 are latent variables relative to the DAG in $\mathcal{D}_1$ because they are implicit in $\mathcal{D}_1$ even though they are missing from it, i.e., not "visible".

There are two different reasons why a variable might be latent in a DAG. The first reason is because you didn't directly observe the variable, but you could have observed and measured it in principle. There are many reasons for why this might occur. Perhaps you simply didn't require the level of causal detail that the latent variable provides? For instance, if the causal description in the $\mathcal{D}_1$ DAG of Figure 6.1 was sufficient for your purposes then you wouldn't need to go into the detail provided by the $\mathcal{D}_3$ DAG. Every causal explanation can be rendered more complex by adding more causal detail[142]. Perhaps you didn't have the time or resources required to measure the variable? Perhaps you didn't have the required equipment to measure it or the required knowledge to use the equipment? Perhaps you didn't know that the missing variable was important until after you had taken your measurements? Perhaps you didn't even know that such a variable exists? In each of these cases, *you* did not directly observe the latent variable but someone with the requisite time, resources, skill or knowledge *could* have directly observed it. The existence of the latent variable, and the capacity to accurately measure it, is not in question although the actual measurement values are missing. I will call this first type of latent variable one that is *latent but observable in practice*.

[141] https://www.merriam-webster.com/dictionary/latent accessed 4 August, 2024.

[142] The standard model of physics tells us that everything in the known universe is composed of 16 fundamental particles (quarks, leptons, gauge bosons and the Higgs boson) that are governed by four fundamental forces (the electromagnetic, weak, and strong nuclear and gravitational forces). In principle, every causal explanation could be reduced to this level but in practise no causal explanation in chemistry or biology is actually reduced to this level.

A second reason why a variable might be latent in a DAG is because no one can directly observe and accurately measure it even they wanted to. For instance, a psychologist might want to study the causal relationships between intelligence, academic performance and later career choices in teenagers, but no one can directly observe or measure "intelligence"; all that we can do is to infer it from performance on a series of standardized tests which, themselves, can be directly observed. Indeed, other psychologists might even deny that "intelligence" has any objective reality beyond the tautological claim that it is the ability to correctly answer the questions on a particular series of standardized tests. Gardner's theory of multiple intelligences (Gardner 1987, Gardner et al. 2018) claims that intelligence exists along several different dimensions and so there is no such thing as "intelligence" as a general property in humans. This more complicated situation requires that we infer its existence and measure it via other variables that we can directly observe. I will call this second type of latent variable one that is *latent and unobservable in practice*. Another name that is often used for this second type of latent variable is a *theoretical construct* (Bollen 1989). As the history of science has repeated shown, latent variables that are unobservable in practice at one point in history can change their meaning and become observable in practice once their definition has been fixed and sufficiently precise measuring devices have been devised. Discussion of theoretical constructs, and measurement models of such theoretical constructs, will be postponed until Chapter 7. In this chapter, we will concern ourselves with DAGs containing latent variables of the first type: variables in a DAG you didn't directly observe, but you could have observed in practice.

6.2 Latent variables: implicitly marginalized or implicitly conditioned

The distinction between latent variables that are observable or unobservable in practice relates to our ability to measure them. This distinction affects how we model such latent variables in structural equation modelling. There is a second distinction that is also important, but which has not received much attention in the SEM literature: whether the latent variable has been implicitly marginalized or implicitly conditioned in your data. This has nothing to do with how you measure latent variables. Instead, it refers to how the latent variable has affected the way that

you have collected your data. When we conduct any statistical analysis, we always work with a sample of observations taken from a much larger (possibly infinite) statistical population. A statistical population is the complete collection of observational units (individuals, items or data points) that possess a specific set of variable characteristics of interest. The sample is the subset of observational units of this statistical population that you have actually observed and included in your data set. You must always choose which observations you will include[143] in your data even if that choice is random and even if the choice is implicit. If the selection (or not) of an observation to be included in the sample date is independent of the values of a latent variable, then the relative frequency of different values of that latent variable in your data will be the same (except for random sampling variation) as the relative frequency of different values of the latent variable in Nature[144]. This is called implicitly *marginalizing* over the latent variable. In a DAG containing latent variables, I will indicate a marginalized latent variable by enclosing it inside a circle. If, on the other hand, you collect data in which the inclusion (or not) of an observation is *dependent* on the values of a latent variable, then the relative frequency of different values of that latent variable in your data will not be the same as the relative frequency of different values of the latent variable in Nature. This is called implicitly *conditioning* on the latent variable. In a DAG, I will indicate an implicitly conditioned latent variable, L_i, due to a selection process, S, by including $S \rightarrow L_i$ in the DAG and enclosing L_i inside a triangle. To summarize, a variable enclosed in a circle means that the inclusion or exclusion of an observation in the data has been done independently of the value of that latent variable while a variable enclosed in a triangle means that the inclusion or exclusion of an observation in the data is dependent on the value of that variable.

The following examples will make this distinction clearer. Imagine that you have collected data on two traits (X_1 and X_2) of a random sample of 100 individuals of some animal species. Your data set therefore contains 100 lines (individuals) and two columns; each column records the value of each of the two traits. Your data set does not include information on whether the

[143] This means that you also implicitly choose which observations you don't include in your data.

[144] This can be stated mathematically. The marginal probability of X_i (i.e. the probability of X_i if we don't know if selection has occurred or not) is $p(X_i)$. The joint probability that selection has occurred (S) and of observing a particular value of a variable X_i is $p(S,X_i)$. This joint probability can also be written, using the chain rule of conditional probability, as $p(S)p(X_i|S)$. So, when will the conditional probability of X_i, once we know that selection has occurred, $p(X_i|S)$, be the same as the marginal probability of X_i when we know nothing about S? When $p(S,X_i)=p(S)p(X_i)$; that is when X_i is independent of S.

individual was alive or dead when you sampled it. If you randomly sample the individuals independently of whether they are alive or dead, and therefore include both living and dead individuals in your data set with the same frequency that they occur in Nature, then you have implicitly marginalized over the latent variable "survival". "Survival" is latent in your data because you have not actually recorded the survival status of each individual in your data set. The relative frequency of living or dead individuals in your data will be the same (except for random sampling variation) as the relative frequency of living or dead individuals in Nature. If, however, you only sample those individuals that are alive (maybe because the dead ones no longer exist) then you have implicitly conditioned on the latent variable "survival" because your choice of inclusion of an individual in your data was dependent on it being alive. Even if you only bias your sampling in favour of living individuals rather than completely exclude dead individuals, you have still implicitly conditioned on survival. The relative frequency of living or dead individuals in your data will not be the same as the relative frequency of living or dead individuals in Nature, even after accounting for random sampling variation. Implicitly conditioning on a latent variable has the same effect as explicitly conditioning on an observed variable. If the two traits each affect the chances of an individual being alive then the DAG with the latent "survival" variable is $X_1 \rightarrow \text{survival} \leftarrow X_2$. In this DAG, survival is a latent collider and the effect of only including living individuals in your data set (i.e., implicitly conditioning on them) is to generate a dependency between the two measured traits X_1 and X_2 even if they are unconditionally independent in Nature. After all, conditioning on the collider (survival) opens up the undirected path between X_1 and X_2, which generates a dependency between them. This is called "selection bias" or Berkson's paradox in statistics (Berkson 1946).

If you are not familiar with the phenomenon of selection bias, then you can observe it in the following simple simulation in R. In this numerical simulation, the two traits of a series of individuals, x1 and x2, are independent of each other but each trait partially determines if the individual will survive or not until we are ready to begin sampling. The DAG is $x1 \rightarrow \text{survival} \leftarrow x2$ and survival is latent since we have only measured the two traits. We generate 10000 observations to represent the statistical population. There is a 50% chance of survival being positive (the individual is alive when we begin sampling) or negative (the individual is dead when we begin sampling). Survival is a standard normal variable, and the individual is alive if survival is positive and deal if survival is negative. We randomly choose 500 of the

10000 observations and this occurs independently of whether the individual is alive or dead. That is, we have implicitly marginalized over the latent survival. The resulting correlation between x1 and x2 in our data is -0.035 (p=0.34). This is what d-separation predicts since survival is a collider, and we have not conditioned on it. Next, we again randomly choose 500 observations but only from those individuals that are alive; that is, we have *implicitly* conditioned on the latent survival by restricting our data to only include individuals whose survival is positive. The correlation between x1 and x2 in this second data set is -0.21 ($p=2e^{-16}$). This is still what d-separation predicts since we have now conditioned on the collider (survival).

```
set.seed(10)
x1<-rnorm(10000)
x2<-rnorm(10000)
survival<-0.5*x1+0.5*x2 +rnorm(1000,0,sqrt(1-2*0.5^2))
marginalized.dat<-data.frame(x1,x2)
N<-dim(marginalized.dat)[1]

#select 500 observations independently of survival
sel<-sample(1:N,size=500)

#correlation between x1 and x2 when not conditioning on survival
cor.test(marginalized.dat$x1[sel],marginalized.dat$x2[sel])

#select 500 observations conditional of survival>0
conditioned.dat<-data.frame(x1[survival>0],x2[survival>0])
N<-dim(conditioned.dat)[1]

#select 500 observations independently of survival
sel<-sample(1:N,size=500)

#correlation between x1 and x2 when using only living
#individuals i.e. (survival>0)
cor.test(conditioned.dat$x1[survival>0],conditioned.dat$x2[survi
val>0])
```

The basic rule in determining how a latent variable affects the dependency between pairs of observed variables (X_i, X_j) in a DAG is this:

- Implicitly marginalizing over a latent variable is equivalent to excluding this latent variable in the conditioning set during d-separation.
- Implicitly conditioning on a latent variable is equivalent to including this latent variable in the conditioning set during d-separation.

So, if you want to know if two observed variables (X_i, X_j) in a DAG with latents are d-separated given some set **C** of other observed variables then you simply apply the above rule. You include the set of implicitly conditioned latents ($\mathbf{L_C}$) into your conditioning set but exclude the implicitly marginalized latents ($\mathbf{L_M}$): $X_i \_ || \_ X_j \,|\, \{\mathbf{C} \cup \mathbf{L_C}\}$ [145]; notice that the conditioning set contains both **C** and the implicitly conditioned latents ($\mathbf{L_C}$) but does not include the implicitly marginalized latents ($\mathbf{L_M}$). That is why, given our DAG $X_1 \rightarrow \text{survival} \leftarrow X_2$, X_1 was independent of X_2 in the data set containing both living and dead individuals (implicitly marginalized survival) but X_1 was correlated with X_2 in the data set containing only living individuals (implicitly conditioned survival). The latent "survival" is a collider along the undirected path from X_1 to X_2 in the DAG. The d-separation relation was $X_1 \_ || \_ X_2 | \{\phi\}$ in the first case and therefore X_1 is d-separated from X_2 by the collider "survival". The d-separation relation was $X_1 \_ || \_ X_2 | \{\phi \cup \text{survival}\}$ in the second case, thus opening up the path by conditioning on the collider "survival".

6.3 Converting a DAG with explicit latents into a MAG without explicit latents

In practice, rather than beginning with a DAG that has latents, we often begin directly with a different type of acyclic causal graph that does not explicitly include these latents but that implicitly includes these latents using different types of edges than just the arrows (→) of a DAG. This different type of causal graph contains the direct effects ($X_i \rightarrow X_j$) and then shows the noncausal dependency (dependent errors) that is generated by an implicitly marginalized latent by a double-headed arrow ($X_i \leftrightarrow X_j$) and the noncausal dependency (selection bias) that is generated by an implicitly conditioned latent by a line ($X_i - X_j$). This new type of causal graph is called a mixed acyclic graph (MAG). In a MAG, the marginalised latent is "hidden" but is implicit in the double-headed arrow ($X_i \leftrightarrow X_j$) while the conditioned latent is "hidden" but is implicit in the line ($X_i - X_j$).

Our goal in this section is to convert a DAG that explicitly contains latent variables into a MAG that does not include these latent variables but that maintains the same causal claims between the

[145] The $\cup$ symbol is the union operator for sets; i.e. the combination of the set **C** and the set $\mathbf{L_C}$.

observed variables as specified in the original DAG with latents. We will do this by making use of the distinction between implicitly marginalized and implicitly conditioned latent variables and by applying the notion of d-separation. We then apply the following steps:

1. Start with the full DAG including latents and identify each variable in the DAG as either (i) observed, (ii) an implicitly marginalized latent (enclosed in circles) or (iii) an implicitly conditioned latent (enclosed in triangles). Construct the initial MAG by listing all (and only) the same observed variables as in the full DAG. At this step, the initial MAG has no edges between any of the variables.
2. If two observed variables (X_i, X_j) are causal parent and child in the DAG then add $X_i \rightarrow X_j$ in the MAG.
3. Identify each pair of observed variables (X_i, X_j) that are not causal parent and child in the full DAG, but that have an undirected path between them consisting *only of latent variables*.
    3.1 Construct the conditioning set ($\mathbf{L_C}$) by adding only the implicitly conditioned variables *in this undirected path* to the conditioning set and excluding all of the implicitly marginalized variables *in this undirected path* from the conditioning set.
    3.2 If X_i is d-separated from X_j in the full DAG conditional on $\mathbf{L_C}$, then there is no edge between them in the MAG.
    3.3 If X_i is not d-separated from X_j in the full DAG conditional on $\mathbf{L_C}$ and there is at least one implicitly conditioned latent in the undirected path (i.e. $\mathbf{L_C}$ is not an empty set) then X_i is dependent of X_j due to selection bias; add an ─ edge between them in the MAG: X_i─X_j.
    3.4 If X_i is not d-separated from X_j in the full DAG conditional on $\mathbf{L_C}$ and there are no implicitly conditioned latents in the undirected path (i.e. $\mathbf{L_C}$ is an empty set) then X_i is dependent of X_j. If X_i is an ancestor of X_j along this undirected path then add $X_i \rightarrow X_j$ to the MAG. If X_i is a descendent of X_j along this undirected path, then add $X_i \leftarrow X_j$ to the MAG. If neither X_i nor X_j are

descendants of the other along this undirected path, then add $X_i \leftrightarrow X_j$ to the MAG[146].

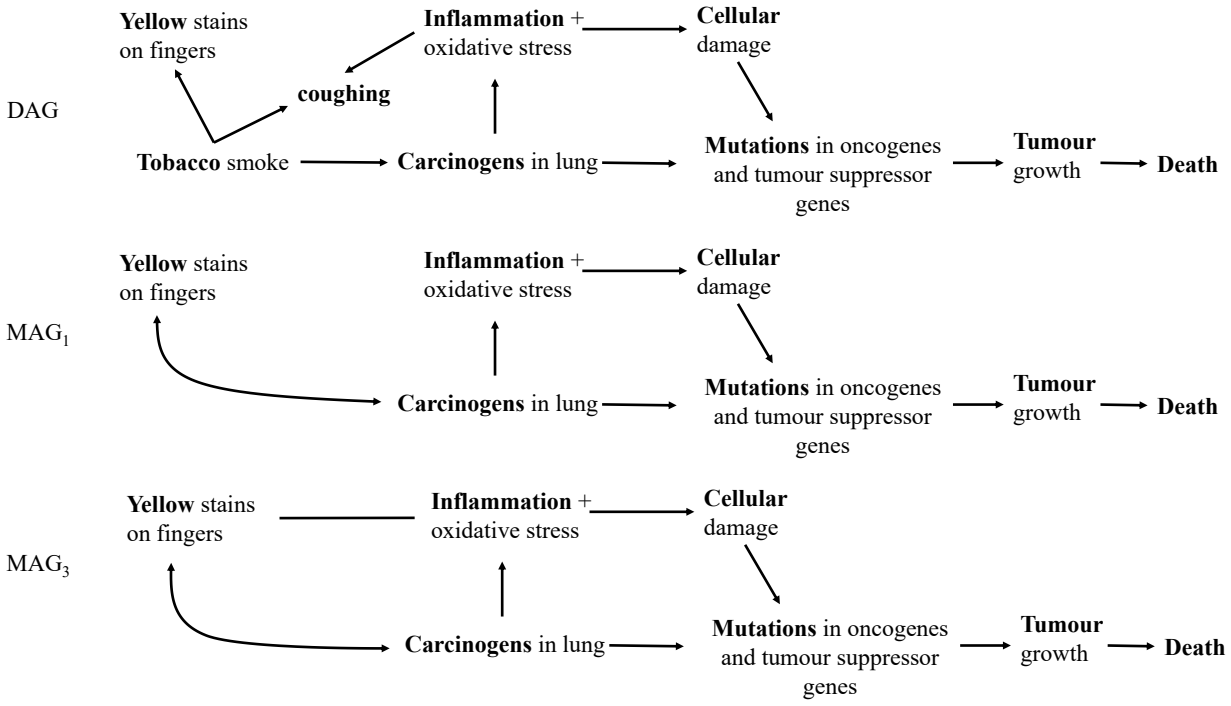


Figure 6.2. A DAG describing the hypothesized causal relationships linking tobacco smoking to premature death. MAG_1 is the mixed acyclic graph, derived from the DAG, when "tobacco smoke", "coughing" and "mutations in oncogenes and tumour suppressor genes" are implicitly marginalized latents. MAG_2 is the mixed acyclic graph, derived from the DAG, when "tobacco smoke" is an implicitly marginalized latent but "coughing" in an implicitly conditioned latent due to selection bias.

Imagine that we propose the DAG shown in Figure 6.2, and we have information in our data set on all of the variables except for (i) whether the patients in the study smoked tobacco or not, (ii) whether they had a persistent cough and (iii) whether they had mutations in the oncogenes or tumour suppressor genes. Therefore, these three variables (tobacco, coughing, mutations) are latent. How would we replace this DAG with a MAG, assuming that patients were not chosen for the study based on whether they smoked, had persistent coughs or had tumors; in other words, assuming that the three latents were implicitly marginalized in the data? MAG_1 in Figure 6.2 shows the answer and is produced by the application of the above rules. The path

[146] It is not possible for X_i to be an ancestor along one undirected path and a descendant along another undirected path since this would result in a cyclic relationship.

yellow←tobacco→carcinogens in the DAG is replaced by yellow↔carcinogens because tobacco is a marginalized latent (therefore the path remains open) and neither "yellow" or "carcinogens" are descendants of the other, generating a noncausal association between the two. The path yellow→tobacco→coughing←inflammation is replaced in the MAG by "yellow" and "inflammation" with no edge between them because both tobacco and coughing are marginalized latents, and coughing is a collider along this path, thus blocking it. The path carcinogens→mutations→tumour is replaced with carginogens→tumour because "mutations" is a marginalized latent and "carcinogens" is an ancestor of "tumour" along this path. Finally, the path cellular→mutations→tumour is replaced with cellular→tumour because, again, "mutations" is a marginalized latent and "cellular" is an ancestor of "tumour" along this path.

How would we replace this DAG with a MAG, assuming that patients were more likely to included in the study if they had persistent coughs; in other words, assuming that coughing was an implicitly conditioned latent while tobacco and mutations were implicitly marginalized? The application of the above rules results in MAG_2 in Figure 6.2. Now, the path yellow→tobacco→coughing←inflammation is replaced with yellow─inflammation because there is now a noncausal association between yellow and inflammation generated by selection bias. This occurs because "coughing" is a collider along this path and "coughing" has been implicitly conditioned in the data by only including patients who did have persistent coughs.

We will not often have to specify MAGs directly when using R because the `pwSEM()` function will do this for us, but we can do this using the `makeMG()` function[147] of the ggm library. The `makeMG()` function has three arguments. The first argument (dg) specifies the arrows in the MAG, and you already know how to enter a DAG; to specify $X_i \rightarrow X_j$, you would enter `dg=DAG(Xj~Xi)`; the "dg" stands for "directed graph". The second argument (ug) specifies the line edges (─) in the MAG; the "ug" stands for "undirected graph". To specify a line edge between variables X_i and X_j you would enter `ug=UG(~Xi*Xj)`. The third argument (bg) specifies the bidirected arrows (↔) in the MAG; the "bg" stands for "bidirected graph". To specify a bidirected arrow between variable X_2 and X_4 you would enter `bg=UG(~X2*X4)`. To specify MAG_2 of Figure 6.2, we would do the following:

[147] makeMG means "make a mixed graph"

```
library(ggm)
my.mag<-makeMG(dg=DAG(inflammation~carcinogens,
cellular~inflammation, tumour~carcinogens+cellular,
death~tumour),
               ug=UG(~yellow*inflammation),
bg=UG(~yellow*carcinogens))
```

6.4 Converting a MAG to an m-equivalent MAG

You already know how to conduct a dsep test to compare a DAG against empirical data when the DAG only involves observed variables (Chapter 3). You can't use a dsep test on a DAG with latent variables because you can't statistically condition on variables that you haven't measured! However, you now know (section 6.3) how to convert a DAG that includes latent variables into a MAG that does not include latent variables but that makes the same causal claims among the observed variables as the original DAG. Since the MAG only involves observed variables, can you use these observed variables to test the MAG against the data by using the same logic as in a dsep test? Yes, but only after we have slightly modified the dsep test. This is because, although a MAG maintains the causal relationships among the observed variables that existed in the full DAG with latent variables, we cannot obtain the smallest set of independence relationships that, together, imply all of the others (i.e. the union basis set) directly from this MAG. In order to obtain the union basis set of the MAG we must first convert this MAG into a (possibly different) type of MAG called an "m-equivalent" MAG that maintains all (and only) the dependence and independence relationships among the observed variables in the original DAG with latent variables. You mustn't confuse MAGs and m-equivalent MAGs; sometimes they are the same and sometimes they are different. We only use m-equivalent MAGs as a statistical "trick" to obtain the union basis set of the MAG.

How can we determine the dependence and independence relationships among the observed variables in the original DAG with latent variables? Richardson and Spirtes (2002) provided the proofs required to answer this question and Bob Douma and I (Douma and Shipley 2021) used these results to extend the dsep test of piecewise SEM to the case of MAGs that include implicit latent variables. The steps are almost identical to those of a dsep test: (i) obtain the union basis

set of independence claims made by the m-equivalent MAG by listing the set of pairs of variables that do not have an edge between them conditional on their parents; (ii) obtain the null probabilities of independence for each of the members of the union basis set; (iii) collect these null probabilities together into Fisher's C statistic; (iv) obtain the null probability of the Fisher's C statistic. There are only two modifications required when testing a MAG rather than a DAG. The first modification is in the first step; we must modify the MAG into an m-equivalent MAG. Often, but not always, the m-equivalent MAG is the same as the MAG. The second modification is in the last step; we must modify the chi-squared distribution to take into account possible non-independence between the null probabilities obtained in the second step.

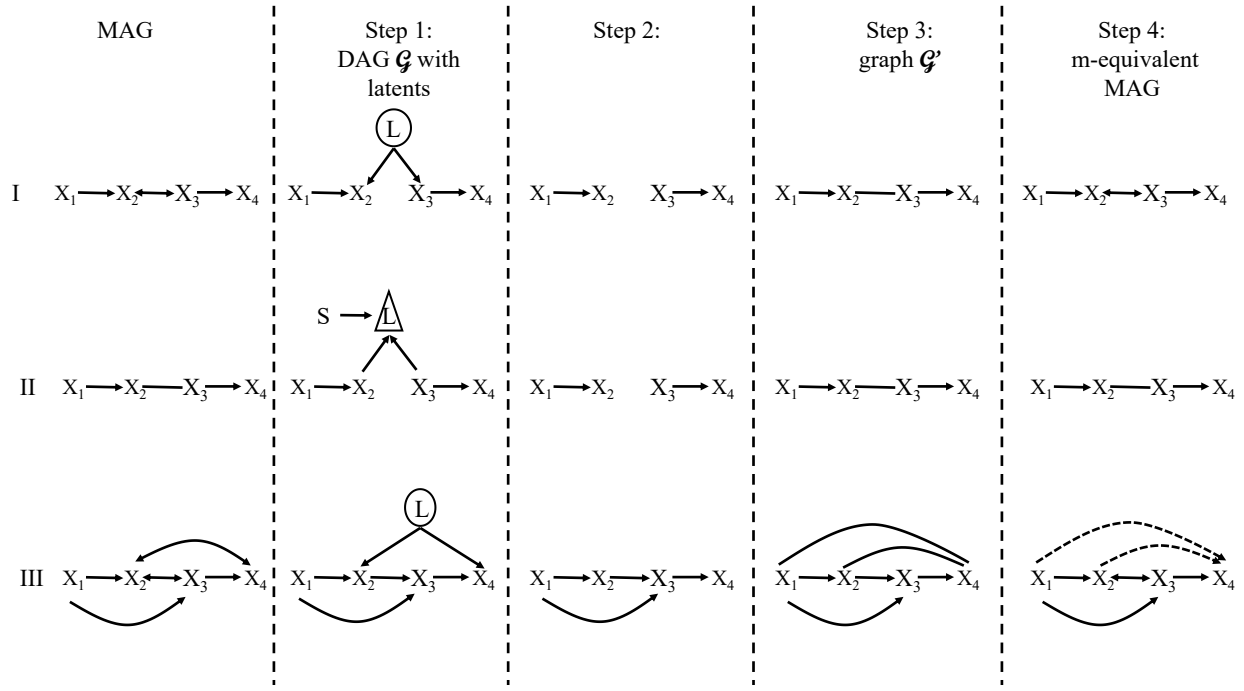


Figure 6.3. Illustration of the four steps required to convert a MAG into an m-equivalent MAG. Variables within circles are implicitly marginalized latents and variables within triangles are implicitly conditioned latents induced by a selection intervention (S) on the latent that excludes certain observations.

There are four steps involved in converting a MAG into its m-equivalent MAG[148] (Douma and Shipley 2021). These four steps are illustrated in Figure 6.3.

Step 1: Convert the MAG into its equivalent DAG containing latent variables by replacing each $X_i \leftrightarrow X_j$ by a marginalized latent: $X_i \leftarrow L_M \rightarrow X_j$ and by replacing each $X_i - X_j$ by a conditioned latent ($X_i \rightarrow L_C \leftarrow X_j$) undergoing selection; call this DAG $\mathcal{G}$. Separate the variables in the DAG into two subsets: those variables that are observed (**O**) and those variables that are latent (**L**). The subset of latent variables is further divided into the subset of latents that are implicitly marginalized ($\mathbf{L_M}$) and those latents that are implicitly conditioned ($\mathbf{L_C}$).

Step 2: Create a new graph ($\mathcal{G}'$) containing only the observed variables that are found in the DAG $\mathcal{G}$ obtained at the end of step 1. Do this by removing all latent variables from $\mathcal{G}$ (including any selection variables, S) and by removing all arrows pointing into, or out of, each latent variable while keeping all of the causal parent → causal child pairs and the arrows between them for the observed variables.

Step 3: Identify each pair of variables (X_i, X_j) in $\mathcal{G}'$ (i.e., produced in step 2), that are not adjacent in $\mathcal{G}'$, i.e. that don't have an arrow between them in $\mathcal{G}'$. If these two variables (X_i, X_j) are not d-separated in the original DAG $\mathcal{G}$ given any possible subset of remaining variables including the latents in $\mathbf{L_C}$ but excluding the latents in $\mathbf{L_M}$, then add a line (–) between X_i and X_j in $\mathcal{G}'$. This line simply indicates that these two observed variables are dependent conditional on all possible subsets as described in the previous sentence. Another way of saying the same thing is that the line indicates that there is an open path between X_i and X_j that cannot be blocked by any subset of other observed variables. Note that the line added in this step does not mean the same thing as the line (—) in the MAG! Before continuing to step 4, let's look in more detail at the first three steps, using each of the three DAGs in Figure 6.3.

In DAG (I), after removing the marginalized latent and its associated arrows at the end of step 2, the pairs of variables that are not adjacent are (X_1, X_3), (X_1, X_4), (X_2, X_3) and (X_2, X_4). We must

[148] We call this an m-equivalent MAG because it is a MAG that maintains all (and only) the independence claims (via m-separation) involving the observed variables in the DAG with latents.

now see if we can d-separate these pairs of variables in the original DAG using any subset (including the null subset, ϕ) of the remaining observed variables plus any implicitly conditioned latents (of which there are none in this example) but excluding the single implicitly marginalized latent (L). If we cannot do this then we must add a line between the pair of variables. Since X_1 and X_3 are d-separated in $\mathcal{G}$ if conditioned on nothing (X_1_||_$X_3|\{\phi\}$), we don't add a line between X_1 and X_3. Since X_1 and X_4 are d-separated by X_3 in $\mathcal{G}$ (X_1_||_$X_4|\{X_3\}$), we don't add a line between X_1 and 4_3. Finally, X_2 and X_4 are also d-separated in $\mathcal{G}$ by X_3 (X_2_||_$X_4|\{X_3\}$) so we don't add a line between X_2 and X_4. However, X_2 and X_3 cannot be d-separated by subset of other variables in $\mathcal{G}$ except for the marginalized latent, which we are not allowed to use in the conditioning set. Therefore, we add a line between X_2 and X_3 ($X_2 - X_3$).

In DAG (II), we have the same set of four pairs of observed variables that are not adjacent at the end of step 2 but the latent (L) is implicitly conditioned, not implicitly marginalized as in DAG (I). This implicitly conditioned latent is also a collider, which opens up the latent since it is always in the conditioning set. Since we must include this implicitly conditioned latent in our conditioning set, X_2 and X_3 (one of the four pairs of variables that are not adjacent at the end of step 2) is not d-separated in $\mathcal{G}$ given L and so we add a line between X_2 an X_3 ($X_2 - X_3$).

DAG (III) is more complicated. There are only two non-adjacent pairs in this example at the end of step 2: (X_1, X_4) and (X_2, X_4). X_2 in not d-separated from X_4 in $\mathcal{G}$ given any subset of remaining variables (excluding the implicitly marginalized latent, which we can't use) because of the open path $X_2 \leftarrow L \rightarrow X_4$, so we must add a line between them ($X_2 - X_4$). What about the pair (X_1, X_4)? If we condition on nothing, then there is an open path: $X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4$ in $\mathcal{G}$. If we condition on X_2 then we open up the undirected path $X_1 \rightarrow X_2 \leftarrow L \rightarrow X_4$ in $\mathcal{G}$ since X_2 is a collider along this path. If we condition on X_3 then we block the undirected path $X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow X_4$ in $\mathcal{G}$ but, since X_3 is a descendant of X_2, and since X_2 is a collider along the path $X_1 \rightarrow X_2 \leftarrow L \rightarrow X_4$, we again open up this undirected path. If we condition on both X_2 and X_3 then we again open up the undirected path $X_1 \rightarrow X_2 \leftarrow L \rightarrow X_4$ in $\mathcal{G}$. Therefore, there is no subset of remaining variables in $\mathcal{G}$ (excluding the implicitly marginalized latent) that d-separates X_1 from X_4 and so we must add a line between them: ($X_2 - X_4$).

Before describing step 4, we need some additional information. (Richardson and Spirtes 2002) define a transformation of a DAG[149] into the graph $\mathcal{G}'$ that is produced at the end of step 3. They prove that the undirected edges in $\mathcal{G}'$ (i.e. the lines that are added at the end of step 3) represent the result of implicitly conditioning on, or marginalizing over, latent variables in the DAG. They distinguish between "ancestral" variables and "anterior" variables in such graphs. X_i is an ancestor[150] of X_j if there is at least one directed path from X_i to X_j in $\mathcal{G}'$ $\left(X_i \rightarrow \cdots \rightarrow X_j\right)$ or if X_i and X_j are the same variable. X_i is *anterior to* X_j in $\mathcal{G}'$ if there is at least one undirected path between X_i to X_j in $\mathcal{G}'$ in which every edge is either a line (—) or an arrow pointing away from X_i. In other words, there can be no bidirected edge (↔) nor any arrow pointing towards X_i, such as $\left(X_i \rightarrow \cdots \leftarrow \cdots \rightarrow X_j\right)$. Of course, in a DAG, a variable that is an ancestor along such a path is also always anterior to X_j, but this is not always true in graphs like those that result after step 3. Lemma 3.9 of Richardson and Spirtes (2002) proves the following orientation rule for orienting a line (—) in the graphs ($\mathcal{G}'$) that result after step 3:

(i) Orient $X_i - X_j$ in $\mathcal{G}'$ as $X_i \rightarrow X_j$ if X_i is an ancestor of X_j in the original DAG with latents. Note that it is therefore impossible for X_i to both be an ancestor of X_j and for X_i and X_j to be joined by either an ↔ or an ─ edge.

(ii) Orient $X_i - X_j$ in $\mathcal{G}'$ as $X_i \leftarrow X_j$ if X_j is an ancestor of X_i in the original DAG with latents. Note that it is therefore impossible for X_j to both be an ancestor of X_i and for X_i and X_j to be joined by either an ↔ or an ─ edge.

(iii) Keep $X_i - X_j$ in $\mathcal{G}'$ as $X_i \text{─} X_j$ if neither X_i nor X_j are ancestors of the other in the original DAG with latents and both X_i and X_j are ancestral to the same implicitly conditioned latent descendant in the original DAG (thus $X_i \rightarrow \cdots \rightarrow L_C \leftarrow \cdots \leftarrow X_j$).

(iv) Orient $X_i - X_j$ in $\mathcal{G}'$ as $X_i \leftrightarrow X_j$ if neither X_i nor X_j are ancestors of the other in the original DAG with latents and condition (iii) doesn't hold.

[149] Actually, they begin with a more general type of graph, which they call an "ancestral" graph. A DAG is a type of ancestral graph.

[150] Note that this definition differs slightly from the definition used Pearl (2009) since, in Richardson and Spirtes Richardson and Spirtes (2002), a variable is ancestral only with respect to a given path. I will use the expressions "ancestor in the graph" and "ancestor along the path" to differentiate between these two meanings.

The resulting graph is an m-equivalent MAG to the original MAG.

Step 4 (Figure 6.3) consists of applying the above orientation rules. Looking at DAG (I) in Figure 6.3, we had a line between X_2 and X_3 at the end of step 3. Since neither X_2 nor X_3 is an ancestor of the other in the original DAG, we apply orientation rule (iv) to give $X_2 \leftrightarrow X_3$. In DAG (II) in Figure 6.1, we also had a line between X_2 and X_3 at the end of step 3. Since neither X_2 nor X_3 is an ancestor of the other in the original DAG, but both X_2 and X_3 are ancestors of the implicitly conditioned latent (L), we apply orientation rule (iii) to give $X_2 — X_3$. In DAG (III), we had two pairs of variables joined by a line: (X_1, X_4) and (X_2, X_4). Since X_1 is an ancestor of X_4 we apply orientation rule (i) to give $X_1 \rightarrow X_4$. Since X_2 is also an ancestor of X4 we again apply orientation rule (i) to give $X_2 \rightarrow X$.

The additional arrows that result in the m-equivalent MAG from step 4 of DAG (III) require more comment. These two arrows cannot mean that X_1 and X_2 are direct causes of X_4 with respect to the other observed variables since this is clearly not the case in the original DAG nor in the original MAG. What do these two arrows mean? They mean that (i) X_2 and X_3 are each causes (but not necessarily *direct* causes) of X_4 in the original MAG and (ii) that the dependence between each and X_4 cannot be removed by conditioning on any possible subset of other *observed* variables. This might seem strange, but it is exactly the same as the example that we used at the beginning of this chapter in which we reduced (tobacco smoke → cancerous tumour growth → death) to (tobacco smoke → death). In the second DAG, tobacco smoke is a cause of death but not a direct cause relative to the first DAG and the dependence between tobacco smoke and death in the second DAG cannot be removed by any set of other observed variables (since there are none in this example). Spirtes et al. (1993) call such arrows *inducing paths*. Any arrow that exists at the end of step 4, but that was not present in the original DAG with its latents, is an inducing path arrow and I will indicate such inducing path arrows by using a broken line for the stem of the arrow.

The `DAG.to.MAG.in.pwSEM()` function of the pwSEM package implements these rules and produces an m-equivalent MAG of a DAG with latent variables. There are three arguments to this function. The argument "`full.DAG=`" takes a named binary matrix encoding the DAG with latents. This is typically provided by the DAG function of the ggm package. The second argument ("`marginalized.latents=`") takes a character vector giving the names of the

implicitly marginalized latent variables in the DAG. The default for this argument is NULL. The third argument ("`conditioning.latents=`") is a character vector giving the names of the implicitly conditioned latents. The default for this argument is NULL. Here is the code to obtain the m-equivalent MAG to DAGs I, II and III in Figure 6.3:

```
library(pwSEM)
library(ggm)
dag<-DAG(X2~X1+L,X3~L,X4~X3)
#DAG (I)
DAG.to.MAG.in.pwSEM(full.DAG=dag,latents="L",conditioning.latent
s=NULL)
#DAG (II)
DAG.to.MAG.in.pwSEM(full.DAG=dag,latents="L",conditioning.latent
s="L")
#DAG (III)
dag<-DAG(X2~X1+L,X3~X2,X4~X1+X3+L)
DAG.to.MAG.in.pwSEM(full.DAG=dag,latents="L",conditioning.latent
s=NULL)
```

6.5 M-separation, the union basis set of a MAG, and Fisher's C statistic

A MAG gives us an acyclic causal graph that contains only observed variables but that differentiates between dependencies generated by causal effects and dependencies generated by two types of noncausal effects. Causal relationships are encoded by the → edge. Noncausal relationships that are generated by common implicitly marginalized latent causes are encoded by the ↔ edge. In Chapter 4, in the context of covariance-based SEM, we called this type of relationship (i.e ↔) a "correlated error" or a "free covariance". More generally, since we don't want to be limited to normally distributed variables and linear relationships, we will call these "dependent errors" when working with MAGs. Noncausal relationships that are generated by common implicitly conditioned latent children through a selection process (i.e. selection bias) are encoded by the ─ edge.

However, simply reducing a DAG with latents into an m-equivalent MAG is not enough. A MAG, just like a DAG, is a multivariate causal hypothesis. We can develop an empirical test of a MAG that is almost identical to the dsep test of DAGs that we discussed in Chapter 3. To do

this, we need to generalize the notion of d-separation (which is limited to DAGs) to apply to MAGs, and then obtain the union basis set of the m-equivalent MAG.

The notion of d-separation in a DAG is a special case of a more general notion of separation in a MAG. Appropriately, this generalization in MAGs is called "m-separation" (Richardson and Spirtes 2002). To use m-separation, we must generalize the notion of a collider variable along a path. In a DAG, a collider variable (X_k) along an undirected path between X_i and X_j has an arrow pointing into it from both directions: ($\rightarrow X_k \leftarrow$). In a MAG, a collider variable (X_k) along an undirected path between X_i and X_j can have any of four orientations: (i) $\rightarrow X_k \leftarrow$, (ii) $\rightarrow X_k \leftrightarrow$, (iii) $\leftrightarrow X_k \leftarrow$ or (iv) $\leftrightarrow X_k \leftrightarrow$. The key point is that there must be an arrowhead pointing into the collider from both directions. This makes perfect sense if you remember that the $\leftrightarrow$ edge means that there is a marginalized latent variable (L_M) "hidden" in the generating process in Nature; so, $\rightarrow X_k \leftrightarrow$ in the MAG means $\rightarrow X_k \leftarrow L_M \rightarrow$ in the DAG with latents and X_k is a collider in the DAG. You can see why d-separation in a DAG is a special case of m-separation in a MAG: in a DAG, only the first orientation ($\rightarrow X_k \leftarrow$) can exist.

We must also generalize the notion of the descendant of a variable when working with MAGs. In a DAG, a variable (X_j) is a descendant of X_i if there is a directed path from X_i to X_j; that is, an undirected path from X_i to X_j in which all of the arrows point towards X_j ($X_i \rightarrow \cdots \rightarrow X_j$). In a MAG, a variable (X_j) is a *quasi-descendant* of X_i if there is an undirected path between X_i and X_j that involves only lines (—) or directed arrows pointing away from X_i and towards X_j ($X_i \rightarrow \cdots$ —$\cdots \rightarrow X_j$). Stated equivalently, there cannot be any double-headed edges ($\leftrightarrow$) along this path nor any arrows pointing towards X_i ($\leftarrow$). Another way to say the same thing is that X_i is *anterior* to X_j. You can again see why d-separation in a DAG is a special case of m-separation in a MAG since, in a DAG, the — edge cannot exist. For m-separation, as in d-separation, conditioning on a collider variable opens up that collider variable along the undirected path. In d-separation, conditioning on the descendant of a collider variable also opens up that collider variable along the undirected path. In m-separation, conditioning on the quasi-descendant of a collider variable opens up that collider variable along the undirected path. Given these two generalizations (the definition of a collider and the definition of a descendant), everything else stays the same when going from d-separation of a DAG to m-separation of a MAG.

Richardson and Spirtes (2002) prove that the independence claims (the m-separation claims) of an m-equivalent MAG are a subset of the independence claims (the d-separation claims) of the original DAG with latents. There can be no m-separation claims in the m-equivalent MAG that are not mirrored by the equivalent d-separation claim in the DAG. In other words, we can test all (and only) the independence claims involving observed variables that are implied by the full DAG with latents by testing the independence claims of the m-equivalent MAG[151]. Therefore, the union basis set of the m-equivalent MAG is a subset of the union basis set of the original DAG. How do you obtain the union basis set of an m-equivalent MAG? As before, you first list each pair of variables (X_i, X_j) that are not adjacent, i.e. that aren't joined by either a unidirectional arrow (→), a bidirectional arrow (↔) or a line (—). Next, you construct the conditioning set (**Z**) by listing the causal parents of each. Each m-separation claim is then $X_i \perp\!\!\!\perp X_j | \{\mathbf{Z}\}$.

If you understood the dsep test in Chapter 3 then you already know how to test a MAG given empirical data except for one detail. As before, you evaluate each of the k independence claims in the union basis set of the MAG, obtain its null probability, and calculate Fisher's C statistic ($C = -2\sum_{i=1}^{k} \ln(p_i)$). In a DAG, the null probabilities associated with the union basis set are mutually independent (Shipley 2000) and this fact allows us to compare the value of C to a chi-squared distribution with 2k degrees of freedom. However, in a MAG, the null probabilities associated with the union basis set (or any other basis set) are not necessarily mutually independent. When this happens, Fisher's C statistic is not distributed as a chi-squared distribution with 2k degrees of freedom. Instead, it is distributed as a gamma distribution, of which the chi-squared distribution is a special case. Brown (1975) showed that if the variables used to obtain the null probabilities of each of the separate tests of independence are normally distributed, then one can approximate the distribution of Fisher's C statistic by taking into account the covariances between the null probabilities[152]. In practice, you use the empirically

[151] However, you must not make the mistake of thinking that by testing the MAG, you are also testing the full DAG with its latents. When testing the MAG, you are only testing the independence claims of the full DAG involving the observed variables.

[152] More specifically, one considers the covariance between the values of $-2\ln(p_i)$, where p_i is the null probability of the i^{th} independence claim. The `mvnconv()` function in the poolr package calculates this.

estimated covariances between the variables to derive the covariances between the null probabilities.

Brown's approximation assumes that the variables used to obtain the null probabilities of each of the separate tests of independence are normally distributed. The generalized covariance statistic (Shah and Peters 2020) is asymptotically normally distributed, as explained in Chapter 3 even when the variables whose residual values are used in this statistic are not normally distributed. In fact, given k independence claims, the resulting k generalized covariance statistics are distributed as a multivariate normal distribution whose correlation matrix is approximated by the sample correlation matrix between the k generalized covariance statistics. Therefore, we can use Brown's correction to get the composite null probability associated with our C statistic by using the generalized covariance statistic as our measure of dependence when we calculate each of the null probabilities and by using the correlations between the generalized covariance statistics to measure the dependencies between the null probabilities.

The `fisher(p,adjust="generalized",R=mvnconv(cor(R)))` function of the poolr package of R (Cinar and Viechtbauer 2022) calculates Fisher's C statistic and determines its null probability by implementing Brown's method[153]. The first argument (p) is the vector of null probabilities associated with each of the generalized covariance statistics, which are obtained from each of the m-separation claims in the union basis set. The third argument (R) is the covariance matrix between null probabilities, which are functions of the matrix of Pearson correlation coefficients between the product of residuals (R) associated with each of the m-separation claims. Remember how the generalized covariance statistic is calculated (section 3.2): given an m-separation claim, X_i_||_$X_j|\{\mathbf{C}\}$, with **C** being the set of causal parents of each of X_i and X_j, each generalized covariance statistic is based on the product of the residuals ($R_{ij} = r_i \cdot r_j$), where r_i is the vector of residuals of X_i regressed on **C** and r_j is the vector of residuals of X_j regressed on **C**. Each m-separation claim in the union basis set therefore generates a vector (R_{ij}) of these products of residuals. You simply collect the k vectors of R_{ij} in a data frame and obtain the matrix of Pearson correlations between these k values using `cor(R)`; this is done

[153] You can get Fisher's C statistic and its null probability assuming mutually independent tests (as done in Chapter 3) using `fisher(p, adjust="none")`, where p is the vector of null probabilities associated with each of the independence tests specified in the union basis set.

automatically in the `pwSEM()` function. The `mvnconv()` function in the poolr package converts the correlations among the generalized covariance statistics into covariances among the null probabilities associated with these statistics.

6.6 Unbiased estimates of path coefficients in MAGs

If you use the dsep test for DAGs (Chapter 3) and conclude that the data do not contradict the DAG (because the null probability of the C statistic is above the chosen significance level), then you proceed to estimate the path coefficients of the structural equations by conducting a series of regressions in which each dependent variable is regressed on its causal parents. This results in unbiased estimates of the path coefficients, assuming that the distributional and functional assumptions of these regressions are correct. However, this is not necessarily true in MAGs. To explain how to get unbiased estimates of causal effects in a MAG, I need to explain the notions of "causal identifiability" and Pearl's "back-door" adjustment.

Parameter identifiability in classical statistics and causal identifiability in Judea Pearl's theory of causal inference are related but distinct concepts. A parameter is statistically identified if its value can be uniquely determined from the probability distribution of the observed data. Statistical identifiability is defined only with reference to a multivariate probability distribution and without any reference to causality. A parameter is statistically identifiable if different values of the parameter lead to different probability distributions of the data. However, you already know about the existence of equivalent DAGS (Chapter 5). Equivalent DAGs are different causal graphs that fit the sample data equally well irrespective of the sample size, because they make the same d-separation claims. That means that a parameter can be statistically identified given a multivariate probability distribution but still result in different estimated values for causal effects between a pair of variables in the different equivalent DAGs. For instance, you know that $X_1 \rightarrow X_2 \rightarrow X_3$ and $X_1 \leftarrow X_2 \leftarrow X_3$ are equivalent DAGs and will produce different estimates for the path coefficients. We must therefore augment the notion of statistical identifiability to that of causal identifiability. Causal identifiability ensures that the added assumptions conveyed by the causal graph, in terms only of its causal topology, are sufficient to insure statistical

identifiability[154] Pearl (2009, p. 77). Theorem 3.2.5 of Pearl (2009, p.78) gives the necessary and sufficient conditions for the causal effect of $X_i \rightarrow X_j$ to be causally identifiable, which I will paraphrase as follows: given a MAG, the causal effect of $X_i \rightarrow X_j$ is causally identifiable if X_i, X_j and the causal parents of X_i are all observed variables in this MAG. Since a DAG is simply a MAG in which all of the variables are observed, this is why all of the causal effects in a DAG are causally identified.

Identifiability is a desirable property of a parameter estimate but, to get a causally identifiable parameter estimate, we must also avoid "confounding bias". A confounder variable in a regression context (Rothman et al. 2018) is a variable that is missing from the regression but that influences both the dependent variable and one or more independent variables. This means that the estimated regression slope will not correctly reflect the true causal relationship between the dependent and independent variables. We want to obtain parameter estimates of our path coefficients for each $X_i \rightarrow X_j$ (or the parameters describing the path function linking X_i and X_j if the linking function is nonlinear) that reflects a true causal effect given our causal hypothesis (our causal graph) rather than a biased estimate due to confounding bias. In other words, we want parameter estimates that tell us by how much X_j will change if we (or Nature) change X_i by one unit while holding constant all other variables in the causal graph. Other than via a randomized experiment, statisticians have devised certain methods to answer this question, such as instrumental variables or the potential-outcome approach (Rosenbaum and Rubin 1983). These methods are subsumed into Pearl's Pearl (2009, p.79) "back-door" criterion. Given $X_i \rightarrow X_j$, a back-door path between X_i and X_j is an open path between them that passes through X_i. Given a child – parent pair in a causal graph, $X_i \rightarrow X_j$, a set of conditioning variables **C** satisfies the back-door criterion relative to this pair if:

(i) No variable in **C** is a descendant of X_i; and

(ii) **C** blocks every path between X_i and X_j that contains an arrow into X_j.

The name "back-door" comes from condition (ii) which requires that only paths with arrows pointing into the causal parent (X_i) be blocked since these paths are entering the causal parent

[154] In other words, to supply the missing information required to statistically identify the parameter values of the structural equations without giving the distributional and functional details of the full structural equations.

"through the back door". However, Pearl's definition did not consider causal graphs in which dependencies can be generated by selection bias. If we include such dependencies, then part (ii) of the definition must be generalized, using m-separation, to read: "**C** blocks every path between X_i and X_j that contains an arrow *or a line* into X_i". If the conditioning set **C** satisfies the back-door criterion then Theorem 3.2.2 of Pearl (2009) assures us that the causal effect of X_i on X_j is causally identifiable and is an unbiased estimate of how much X_j will change if we (or Nature) change X_i by one unit while holding constant all other variables in the causal graph. Of course, this back-door adjustment requires that Nature really has generated the data as specified by the causal graph; this is why it makes no sense to estimate the structural equations before we have assessed that data against our causal hypothesis (i.e., our MAG) and not falsified it. We need sufficient evidence to accept the MAG before Pearl's Theorem 3.2.2 can ensure that our path coefficients[155] measure the causal effects.

As an example, consider the third MAG (III) in Figure 6.3. Variable X_4 in this MAG has only one observed causal parent (X_3). If we were to simply regress X_4 on X_3 then this would result in a biased estimate of the function linking this parent – child pair. We know this because X_4 still has a dependency with each of X_2 and X_1, as shown by the fact that the m-equivalent MAG has $X_2 \rightarrow X_4$ and $X_1 \rightarrow X_4$. There is a back-door path between X_3 and X_4 via $X_3 \leftarrow X_2 \leftarrow L \rightarrow X_4$ because this open path between X_3 and X_4 has an arrow pointing into X_3. If we include X_2 in the conditioning set, then this opens another back-door path $X_3 \leftarrow X_1 \rightarrow X_2 \leftarrow L \rightarrow X_4$ since X_2 is a conditioned collider along this path. We must include both X_2 and X_1 (the two other parents of X_4 in the m-equivalent MAG) in the conditioning set, $\mathbf{C}=\{X_2, X_1\}$, in order to block both of these back-door paths so that we have an unbiased estimate of the causal effect (the path coefficient) of X_3 on X_4.

These two arrows in the m-equivalent MAG ($X_1 \rightarrow X_4$ and $X_2 \rightarrow X_4$) represent inducing paths; they are not direct causal claims made by the original MAG, but they are still dependencies. After all, neither the MAG nor the equivalent DAG (III) with its latent in Figure 6.3 claims that X_4 is a causal child of either X_2 or X_1. However, it is still true that X_4 would change if (i) we manipulated X_1 while holding constant all other observed variables (X_2 and X_3) and (ii) if we manipulated X_2 while holding constant all other observed variables (X_1 and X_3) *as long as we*

[155] Or whatever parameter estimates we have obtained to quantify the functional form of the $X_i \rightarrow X_j$ function.

don't restrict our data to observations that are dependent on the latent variable (L); i.e. as long as we marginalize, rather than condition, on the latent variable. This means that we would obtain a biased estimate of the path coefficient from X_3 to X_4 if we simply regressed X_4 on X_3. In order to get an unbiased estimate of this path coefficient, we must regress X_4 simultaneously on X_1, X_2 and X_3 as the m-equivalent MAG states. As a general rule, you must use the m-equivalent MAG, not the MAG itself, when setting up your structured regression equations in order to obtain unbiased path coefficients, even if some of these path coefficients are not found in the MAG.

The pwSEM package includes all of the details discussed in this chapter for DAGs with latent variables. Given a MAG, it obtains the m-equivalent MAG and its union basis set, the C statistic after modifying the degrees of freedom of the associated chi-squared distribution according to Brown's correction and produces unbiased estimates[156] of the resulting path coefficients.

6.7 piecewise SEM of a MAG

We now have all of the components needed to test MAGs using piecewise SEM. Let's first look at some simulated data based on the MAG in Figure 6.4. I generated 1000 observations using standard normal distributions for each of the five observed variables[157] based on the DAG with its latent and with path coefficients equal to 0.5.

[156] Assuming, of course, that you have chosen a regression model that is appropriate for the distributional and functional assumptions of the dependent variable.

[157]
```
set.seed(101)
X1<-rnorm(1000
L1<-rnorm(1000)
X2<-0.5*X1+0.5*L1+rnorm(1000,0,sqrt(1-2*0.5^2))
X3<-0.5*X2+rnorm(1000,0,sqrt(1-0.5^2))
X4<-0.5*X3+0.5*L1+rnorm(1000,0,sqrt(1-2*0.5^2))
X5<-0.5*X4+rnorm(1000,0,sqrt(1-0.5^2))
my.dat<-data.frame(X1,X2,X3,X4,X5)
```

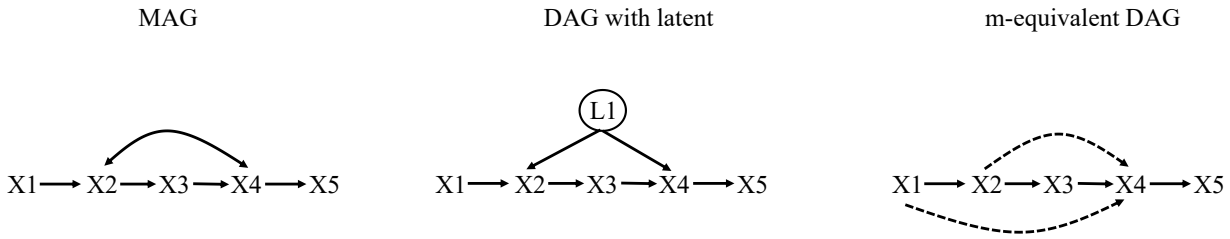


Figure 6.4. A MAG with correlated errors between X_2 and X_3 due to an implicitly marginalized latent that is a common cause of both, the full DAG including this latent, and the m-equivalent MAG.

We first create the MAG object (my.mag):

```
my.mag<-makeMG(dg=DAG(X2~X1,X3~X2,X4~X3,X5~X4),bg=UG(~X2*X4)).
```

We then convert the MAG into a DAG with an implicitly marginalized latent[158] by replacing $X_2 \leftrightarrow X_4$ with an implicitly marginalized latent common cause of X_2 and X_4 (i.e. $X_2 \leftarrow L1 \rightarrow X_4$). We can easily do this via the `DAG()` function for this simple MAG[159] but the `pwSEM()` function uses the `MAG.to.DAG.in.pwSEM()` function of the pwSEM package:

```
DAG.with.latent<-DAG(X2~X1+L1,X3~X2,X4~X3+L1,X5~X4).
```

We then obtain the m-equivalent MAG, using the `DAG.to.MAG.in.pwSEM()` function of the pwSEM package:

```
equivalent.mag<- DAG.to.MAG.in.pwSEM(
full.DAG=DAG.with.latent,latents="L1",conditioning.latents=NULL)
```

We then get the union basis set of this m-equivalent MAG using the `basiSet.mag()` function of the pwSEM package:

```
basiSet.MAG(equivalent.mag)
[[1]]
[1] "X2" "X5" "X1" "X4"

[[2]]
[1] "X1" "X3" "X2"

[[3]]
```

[158] If you have an implicitly conditioned latent (X─Y) then you would add X→L←Y.
[159] `DAG.with.latent<-DAG(X2~X1+L1,X3~X2,X4~X3+L1)`

```
[1] "X1" "X5" "X4"

[[4]]
[1] "X3" "X5" "X2" "X4"
```

We then get the null probability of the generalized covariance statistic for each of these four independence claims as well as the product of residuals. For the first independence claim, we must regress each of X_2 and X_5 on X_1 and X_4, collect the response residuals from these two regressions (r1, r2), get the product of the residuals (R12) and the null probability (p1).

```
r1<-residuals(lm(X2~X1+X4,my.dat))
r2<-residuals(lm(X5~X1+X4,my.dat))
R12<-r1*r2
p1<-generalized.covariance(r1,r2)$prob
```

We repeat this for the other three independence claims in our union basis set to get the vector of null probabilities (p.vec) and the data frame containing the four vectors of products of residuals (R).

```
p.vec<-c(p1,p2,p3,p4)
P.of.R<-data.frame(R12,R13,R15,R35)
```

The vector of null probabilities is (0.546, 0.631, 0.013, 0.932). The Pearson correlations between the four vectors of products of residuals is:

```
> round(cor(P.of.R),3)
       R12    R13   R15    R35
R12  1.000 -0.064 0.038 -0.015
R13 -0.064  1.000 0.034 -0.012
R15  0.038  0.034 1.000  0.081
R35 -0.015 -0.012 0.081  1.000
```

Given 1000 observations, a Pearson correlation becomes significant at the 5% level at approximately 0.06 and so we see that the first and second m-separation claims (r=-0.06) and the fourth and fifth m-separation claims (r=0.08) are dependent even though these correlations are very weak. Fisher's C statistic, calculated without Brown's correction, is 10.916 with 8 degrees of freedom. Using Brown's correction[160], we get the following output:

[160] The line stating "test statistic: 10.844 ~ chi-square(df=7.947)" means that Fisher's C statistic relative to a gamma distribution with a null probability of 0.207 is equivalent to a chi-squared value of 10.844 with 7.947 degrees of freedom.

```
fisher(p=p.vec,adjust="generalized",R=mvnconv(cor(P.of.R)))

combined p-values with:      Fisher's method
number of p-values combined: 4
test statistic:              10.844 ~ chi-square(df = 7.947)
adjustment:                  Brown's method
combined p-value:            0.207
```

The correction to the null probability is very small because the correlations between the generalized covariance statistics are very weak. However, this is not always the case[161].

Since we have no strong evidence to reject our MAG, we can proceed to estimate the path coefficients. To do this, we must perform the set of four structured regressions by following the m-equivalent MAG in Figure 6.4, not our original MAG, even though we only want the path coefficients that are in our original MAG. Note that, in the third regression, we regress X_4 on $X_1 + X_2 + X_3$ even though we only need the path coefficient for X_3.

```
summary(lm(X2~X1))
summary(lm(X3~X2))
summary(lm(X4~X1+X2+X3))
summary(lm(X5~X4))
```

Here are resulting structured regressions for our MAG, with the values inside the box being the inducing paths in the m-equivalent MAG that we would ignore for our hypothesised MAG. The path coefficients are all within ±1 standard error of their population values (0.5).

$$X_2 = 0.046 + 0.530(\pm 0.029) X_1$$
$$X_3 = 0.012 + 0.508(\pm 0.027) X_2$$
$$X_4 = -0.003 \boxed{-0.145 X_1 + 0.323 X_2} + 0.528(\pm 0.030) X_3$$
$$X_5 = 0 + 0.488(\pm 0.025) X_4$$

6.8 Piecewise SEM of a MAG using pwSEM

[161] I have done several numerical simulations based on different MAGs and with varying degrees of association between the free covariances and even quite strong correlations generated by free covariances only bias the null probability of Fisher's C statistic downwards slightly. This suggests that comparing Fisher's C statistic to a chi-squared distribution without Brown's correction will probably not change the decision to reject the MAG except if the null probability is close to the 0.05 significance level. However, numerical simulations are not the same as a mathematical proof and so it is always better to use Brown's correction with MAGs!

You already learnt how to use the `pwSEM()` function in Chapter 3. To compare data to a MAG that includes dependent errors, you enter the list of structured regressions[162], representing the directed edges (arrows) in your MAG, just as you did in Chapter 3. You enter the dependent errors that are generated by marginalized latents (the ↔ edges) as a list in the `marginalized.latents=list()` argument in the `pwSEM()` function in which each element in this list is a $X_i \leftrightarrow X_j$ pair in the MAG written as $X_i \sim\sim X_j$, separated by a comma. You enter the dependent errors that are generated by conditioned latents (the ─ edges) as a list in the `conditioned.latents=list()` argument in which each element in this list is an $X_i \text{─} X_j$ pair in the MAG written as $X_i \sim\sim X_j$, separated by a comma. Everything else in the `pwSEM()` function is the same as described in Chapter 3.

Here is how to do the analysis[163] of the MAG in Figure 6.4 using `pwSEM()`.

```
library(pwSEM)
structured.regressions<-list(
  gam(X1~1,data=my.dat,family=gaussian),
  gam(X2~X1,data=my.dat,family=gaussian),
  gam(X3~X2,data=my.dat,family=gaussian),
  gam(X4~X3,data=my.dat,family=gaussian),
  gam(X5~X4,data=my.dat,family=gaussian)
)
out<-pwSEM(sem.functions=structured.regressions,
marginalized.latents=list(X2~~X4),data=my.dat)
summary(out,structural.equations=TRUE)
```

Here is the first part of the summary output:

```
 Causal graph:
X1 ->X2
X2 ->X3
X2<->X4
X3 ->X4
X4 ->X5
d-separation equivalent DAG or MAG
X1 ->X2
X2 ->X3
X2 ->X4
X1 ->X4
X3 ->X4
X4 ->X5
```

[162] Exogenous variables in this list are regressed only on their mean: lm(X~1)

[163] Of course, you can change the distributional family, use a mixed model or even a generalized additive model as appropriate for the data.

```
Basis Set
( 1 )  X2 _||_ X5 | { X1 X4 }
( 2 )  X1 _||_ X3 | { X2 }
( 3 )  X1 _||_ X5 | { X4 }
( 4 )  X3 _||_ X5 | { X2 X4 }

Null probabilities of independence claims in basis set
(1) 0.5462247
(2) 0.6310464
(3) 0.01327271
(4) 0.9316496
Number of observations in data set: 1000


C-statistic: 10.91589 , df = 8 , null probability: 0.2065108

Brown correction to null probability for correlated tests: 0.2069729


Correlations between the tests of independence
(product of residuals):
       1      2     3      4
1  1.000 -0.064 0.038 -0.015
2 -0.064  1.000 0.034 -0.012
3  0.038  0.034 1.000  0.081
4 -0.015 -0.012 0.081  1.000


AIC statistic: 12939.87
```

The second part of the summary output (if you include the `structural.equations=TRUE` argument in the summary function) outputs each regression for the MAG. If the variables in the regression are normally distributed, then the standardized values are also output. The values of the dependent errors (specified in the lists given in the `marginalized.latents=` and the `conditioned.latents=` arguments in the pwSEM function) are printed as the covariances and Pearson correlations between the residuals (if both variables in the X_i~~X_j pair are normally distributed) or, otherwise, as Spearman correlations between the residuals.

6.9 Piecewise SEM of a MAG using piecewiseSEM

The `psem()` function of the piecewiseSEM package includes a special operator (`%~~%`) to represent a dependency between two variables arising from a marginalised latent that is a common cause of two observed variables. For instance, including `X%~~%Y` in a call to `psem()` is supposed to include a dependency (a free covariance or a correlated error) between X and Y.

However, the `psem()` function incorrectly obtains the union basis set from the underlying DAG and then simply removes the d-separation claim between the pair involved in the dependency. This is *not* how MAGs with double headed arrows (i.e. X↔Y) should be modelled, as explained above. As a result, the resulting union basis set, the C statistic, and its null probability are wrong, and the path coefficients are possibly biased as well[164]. Furthermore, the AIC statistic that is output in the `summary()` is also wrong (the correct way is explained below) because it does not include the likelihood value of this dependency. You should not use this package to model causal graphs that are not true DAGs.

6.10 Parameter estimation in the presence of selection bias

Most readers will not need this section but, if you want to understand when selection bias affects the estimation of causal effects, read on.

In general, we can write the nonparametric structural equation for any variable, x_i, as $x_i = f_i\left(pa_i, \varepsilon_i\right)$ where pa_i are the observed parents of x_i and ε_i are the latent causes of x_i.

Consider the simple causal process shown in Figure 6.5 (i, left). Selection bias on X_i can be conceptualized as an intervention, S_i (by us or by Nature) that alters the function f_i between X_i and its parents (here, ε_i). Graphically, this is represented by adding S_i as a parent of X_i and the effect of this intervention can be analysed by standard conditionalization (Pearl 2009, p.71); i.e. by conditioning our probability on the event that variable S_i takes the value s_i. For example, S_i could be a binary variable taking values of {include, exclude} or S_i could be a function specifying the probability that an observation is included. Selection bias based on the values of X_i can therefore be written graphically as in Figure 6.5 (i, right). The conditional probability of X_j given X_i, $p(X_j|X_i)$, remains invariant to changes in S_i for any set of descendants of X_i if X_i d-separates S_i from these descendants. This is true for the same reason as explained earlier in this chapter. The marginal probability of X_i, $p(X_i)$, is the probability of X_i when we don't know if selection has occurred or not. The joint probability that selection has occurred (S) and of

[164] This applies to version 2.3.0

observing a particular value of a variable X_i is $p(S,X_i)$. This joint probability can also be written, using the chain rule of conditional probability, as $p(S)p(X_i|S)$. So, when will the conditional probability of X_i, once we know that selection has occurred, i.e. $p(X_i|S)$, be the same as the marginal probability of X_i when we know nothing about S? When $p(S,X_i)=p(S)p(X_i)$; that is when X_i is independent of S.

If the conditional probability of X_j given X_i, $p(X_j|X_i)$, remains invariant after selection (i.e. to changes in S_i) then the function describing the causal effect of X_i on X_j (for instance, a path coefficient) doesn't change (Pearl 2009, p. 72). If the conditional probability of X_j given X_i, $p(X_j|X_i)$, does change after selection (i.e., it does not remain invariant to changes in S_i) then the function describing the causal effect of X_i on X_j (for instance, a path coefficient) does change. This rule is formalized by Theorem 3.2.2 ("Adjustment for direct causes") in Pearl (2009). This allows us to predict when a path coefficient, or a path function, will change after selection bias. The rule is: $p(X_j,X_i)$ is invariant, and the causal effect of X_i on X_j doesn't change after an intervention (S_i) in which selection occurs on X_j, if X_j is d-separated from S_i, given X_i, or $X_j \perp\!\!\!\perp S_i | \{X_i\}$. For example, Figure 6.5 (i) shows selection on values of X_i in which S_i which takes values of "select observation i" whenever (say) X_i is positive. Since X_j is d-separated from S_i given X_i, we know that selection on X_i will not change the causal effect of X_i on X_j. For example, if you regressed X_j on X_i then the slope would be the same in the presence or absence of selection on X_i and numerical simulation confirms this. Figure 6.5 (ii) shows selection on values of X_j. Since X_j is not d-separated from S_i given X_i, we know that selection on X_j will change the causal effect of X_i on X_j. Therefore, if you regressed X_j on X_i then the slope would change depending on whether this was done using data resulting from the presence or absence of selection on X_j and numerical simulation confirms this. The fact that the regression slope changes is not a "bias" in the estimation of the slope. It is still an accurate description of the causal effect of X_i on X_j because the presence or absence of selection changes the causal effect.

Figure 6.5 illustrates this fact. It shows five different DAGs, the MAGs that result from them, and the m-equivalent MAGs. In each case, the true values of all path coefficients are 0.5, including the one associated with $X_i \rightarrow X_j$. I simulated very large (10,000) data sets according to these DAGs, and either regressed X_j on X_i without blocking the back-door path through X_h (the values below the $X_i \rightarrow X_j$ arrow in the m-equivalent MAG) and after blocking the back-door through X_h by regressing X_j on both of its parents in the m-equivalent MAG (i.e., X_h and X_i).

The estimated slope after blocking the back-door path is shown above the $X_i \rightarrow X_j$ arrow in the m-equivalent MAG and the estimated slope without blocking the back-door path is shown below the $X_i \rightarrow X_j$ arrow in the m-equivalent MAG. The estimated slope is not significantly different from 0.5 in the first three m-equivalent MAGs when blocking the back door path even though the slope is always significantly different from 0.5 when not blocking the back door path. The reason why the estimated slope for $X_i \rightarrow X_j$ in (iii) is not affected by the selection process is because X_j is d-separated from S_1 given X_i ($X_j \perp\!\!\!\perp S_i | \{X_i\}$.). The last two DAGs show selection bias that will change the slope after selection. This is because the second selection process, S_2, is not d-separated from X_j given X_i.

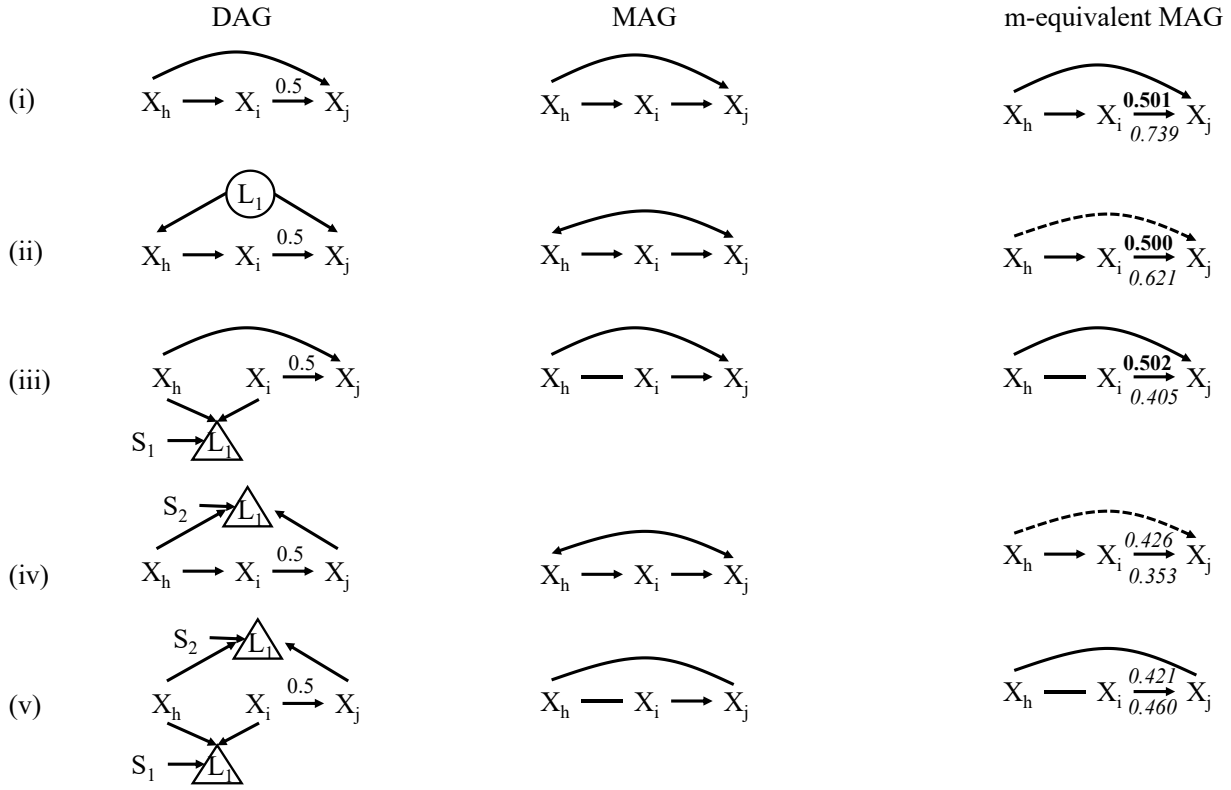


Figure 6.5. Variables within circles are implicitly marginalized latents. Variables within triangles are implicitly conditioned latents given selection (S) on these latents. The path coefficients for $X_i \rightarrow X_j$ are all equal to 0.5 in the pre-selected population. Values above the arrow in the m-equivalent MAG give the estimated path coefficient after blocking the back-door path by regressing X_j on both X_h and X_i and values below the arrow in the m-equivalent MAG give the estimated path coefficient when not blocking the back door path; i.e. when regressing X_j only on X_i. Values in **bold** are not significantly different from 0.5 and values in *italics* are significantly different from 0.5.

6.11 AIC statistics and MAGs

The AIC statistic, applied to MAGs, is used in exactly the same way as described in Chapter 5. However, the calculation of the AIC statistic for DAGs that was described in Chapter 5, in the case of piecewise SEM, requires some modification before we can extend it to MAGs. If you only want to use the AIC statistic, and don't care about how it is calculated, then you can skip this section after knowing one thing: the AIC statistic that is output by the piecewiseSEM package (Lefcheck 2016) is wrong because it does not include the maximum likelihood values of the correlated pairs of variables joined by the double-headed arrows (↔).

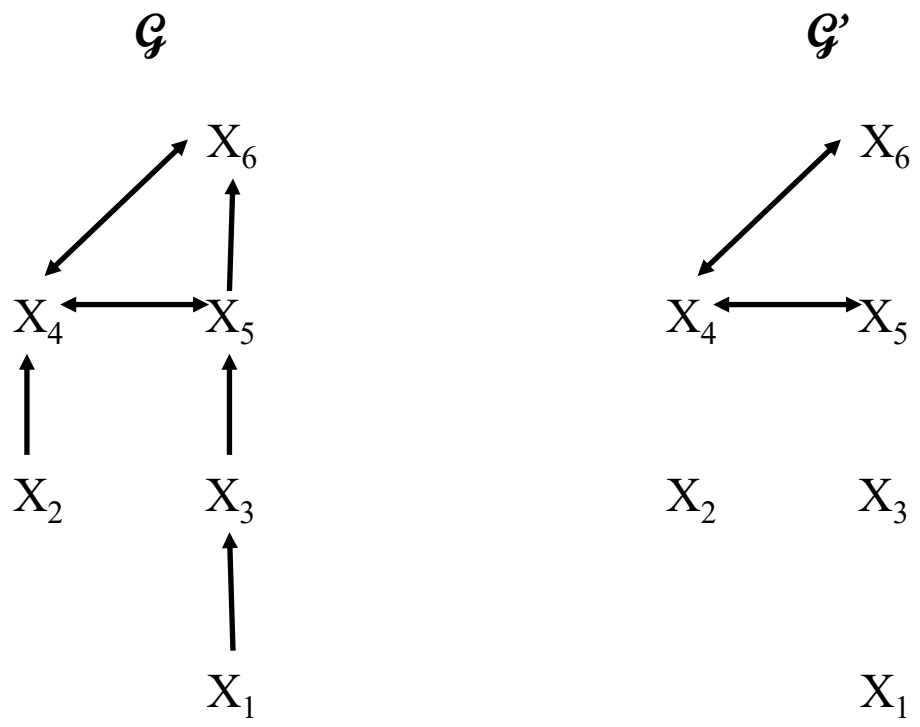


Figure 6.6. A MAG $\mathcal{G}$ and its induced bidirected graph $\mathcal{G}'$. There are six variables in $\mathcal{G}$ but only 4 districts: X_1, X_2, X_3 and $\{X_4, X_5, X_6\}$.

In Chapter 5 (Equation 5.3) I explained that the AIC statistic for data generated by a DAG (i.e. a MAG without any double-headed arrows) is simply the sum of the AIC statistics associated with each separate regression in the set of structural equations. We have to generalize Equation 5.3 when dealing with a MAG containing any double-headed arrows. This generalization requires the notion of a "district" in a causal graph (Evans and Richardson 2014). In order to identify the districts of a MAG $\mathcal{G}$, (Figure 6.6) you first remove all of the arrows from $\mathcal{G}$ while keeping all of

the double-headed arrows, to produce an "induced bidirected graph" $\mathcal{G}'$. All unique sets of variables in $\mathcal{G}'$ that are joined when ignoring the directions of the arrowheads define a single district of $\mathcal{G}$. There are therefore only four districts in $\mathcal{G}$. Variables X_1, X_2 and X_3 each form their own districts but variables X_4, X_5 and X_6 are all in a single district because each can be reached from the other when ignoring the directions of the arrowheads in the induced bidirected graph ($\mathcal{G}'$). Notice that every variable in a DAG (i.e. a MAG without double-headed arrows) is in its own unique district.

Given a DAG with n variables, in Chapter 5 I said that the multivariate probability distribution that is generated by this DAG ($p(X_1,\ldots,X_n)$) can always be decomposed into the product of each the conditional probability of each variable, conditioned on its parents:

$p(X_1,\ldots X_n)=\prod_{i=1}^{n} p(X_i \mid \mathbf{pa}(X_i))$. Given a MAG with n variables, the multivariate probability distribution that is generated by this MAG can always be decomposed into the product of each the conditional probability of each district D, conditioned on its "external" parents:

$p(X_1,\ldots X_n)=\prod_{i=1}^{n} p(D_i \mid \mathbf{epa}(D_i))$. An external parent (epa) of a district is the parent of any variable in the district that is not, itself, a member of that district. The proof is given in Appendix S1 of (Douma and Shipley 2021). The external parents of the district {X_4, X_5, X_6} in Figure 6.6 are X_2 (the parent of X_4) and X_3 (the parent of X_5). Even though X_6 is in this district, and even though X_5 is the parent of X_6, X_5 is not an external parent of the district because X_5 is a member of the district. So, the multivariate probability density that is generated by the MAG $\mathcal{G}$ in Figure 6.6 is $p(X_1,\ldots X_n)=p(X_1)\,p(X_2)\,p(X_3 \mid X_1)\,p(X_4,X_5,X_6 \mid X_2,X_3)$.

Because of the symmetry of a probability density function and a likelihood function (Chapter 5), this means that the maximum likelihood function for a MAG is

$\mathcal{L}(X_1,\ldots X_n)=\prod_{i=1}^{n} \mathcal{L}(D_i \mid \mathbf{epa}(D_i))$ and the log-likelihood function for a MAG is

$\mathcal{LL}(X_1,\ldots X_n)=\sum_i \mathcal{LL}(D_i \mid \mathbf{epa}(D_i))$. So, the log-likelihood function for the data generated by the MAG $\mathcal{G}$ in Figure 6.6 is

$\mathcal{LL}(X_1,\ldots X_n) = \mathcal{LL}(X_1) + \mathcal{LL}(X_2) + \mathcal{LL}(X_3 \mid X_1) + \mathcal{LL}(X_4, X_5, X_6 \mid X_2, X_3)$. It also follows, for the same reasons as given in Chapter 5, that the AIC statistic for a MAG is the sum of the AIC statistics of each district. Since each variable in a DAG defines its own district, this is why the equation for the AIC statistic of a DAG (Equation 5.3) is a special case of the AIC statistic of a MAG.

Let's take a closer look at how to calculate the log-likelihood values for the MAG in Figure 6.6. You already know how do to this for the first three terms because it is the same as for DAGs. For the first term ($\mathcal{LL}(X_1)$), and using R, you would regress X_1 only on its intercept[165] (call the output "fit"), specifying the probability distribution of the residuals, and extract the log-likelihood using the `logLik(fit)` function of R. You would do the same thing for $\mathcal{LL}(X_2)$. For $\mathcal{LL}(X_3 \mid X_1)$, you would regress X_3 on X_1 and extract the log-likelihood from this regression. These first three values of log-likelihood are straightforward because, in each case, we are performing a regression with only one dependent variable. What about the last term $\mathcal{LL}(X_4, X_5, X_6 \mid X_2, X_3)$? Here, we have three dependent variables (X_4, X_5 and X_6) conditioned on both X_2 and X_3. This requires us to get the log-likelihood of a trivariate conditional probability density. As an added challenge, we have to do this involving variables having different distributions and potentially nonlinear relationships.

When estimating the log-likelihood of parametric multivariate conditional probability distributions, like the one for the {X_4, X_5, X_6} district in the MAG $\mathcal{G}$, we can use copulas (Douma and Shipley 2021). A copula is a statistical method that allows us to model how random variables are related to one another, even if their individual (marginal) distributions are different[166]; see Appendix S2 of Douma and Shipley (2021) for a more detailed explanation. Sklar's Theorem (Sklar 1959) states that any multivariate joint probability distribution can be decomposed into is marginal distributions and a copula. The pwSEM package uses this fact[167] to obtain the maximum likelihood values of each multivariate district in a MAG.

---

[165] $X_1 \sim 1$

[166] Assuming that these individual distributions can be modelled using parametric probability distributions.

[167] There are different types of copulas (Gaussian copulas, t-copulas, Archimedean copulas). The current version of pwSEM only supports a Gaussian copula which can convert between Normal, Poisson, Binomial and negative Binomial distributions for the individual variables in the district.

In particular, the pwSEM package uses a "two-stage" maximum likelihood process. First, the structural equations are estimated using regressions using unbiased path coefficients, as explained in section 6.6. The maximum likelihood estimates of the correlations between the variables in the multivariate district are then conditioned on predicted values of the variables in the multivariate district using the copula package of R (Kojadinovic and Yan 2010). Variables within a single district that are d-separated will have their bivariate correlation fixed at zero when maximizing the likelihood. As a result, these two-stage maximum likelihood estimates from the full MAG can differ slightly from those produced from simultaneous maximum likelihood estimates produced in covariance-based SEM when all variables are normally distributed and linearly related. Of course, if the variables are not normally distributed and linearly related, then simultaneous maximum likelihood estimates produced in covariance-based SEM will be wrong anyway since these assume multivariate normality!

7

Modelling explicit latent variables in covariance-based SEM

7.1 Explicit vs. implicit latent variables

If you include a variable in a DAG or a MAG that you have not directly observed or measured, then such a variable is called a "latent" variable. It is latent because it is part of your causal hypothesis (your DAG or MAG) but it is not present in the empirical data. In this book, latent variables are identified in the causal graph by enclosing them in circles (if they have been implicitly marginalized by your sampling method) or by enclosing them in triangles (if they have been implicitly conditioned by your sampling method). If you have forgotten what it means to implicitly marginalize or condition a latent variable, then go back to Chapter 6.

The strategy of dealing with latent variables that was employed in Chapter 6 was to convert the *explicit* latent variables in the DAG into *implicit* latent variables. Each implicit latent variable is hidden inside a double-headed arrow (↔) or a line (—) in a MAG. The advantage of modelling latent variables as implicit is that we can use the statistical flexibility of piecewise SEM. Certainly, double-headed arrows can also be modelled using covariance-based SEM but only by imposing the stricter statistical assumptions that come along with it. When we convert an explicit latent variable into an implicit one then we are simultaneously doing two things. First, we are taking into account the existence of this latent variable as part of the hypothesised data generating process. Second, we are hiding it as a variable in its own right. This is a reasonable strategy when we are primarily interested in the causal relationships between the observed variables and when the implicit latents are only causing a few dependencies. However, there are often situations in which we are interested in the latent variable itself and, perhaps, the causal relationships between different latent variables. Similarly, there are often situations in which the

hypothesised latent variables cause dependencies between several of the observed variables. When this occurs then treating the latent variables as implicit results in an equivalent MAG that makes very few testable causal claims. We don't want to hide the latents in such situations by making them implicit.

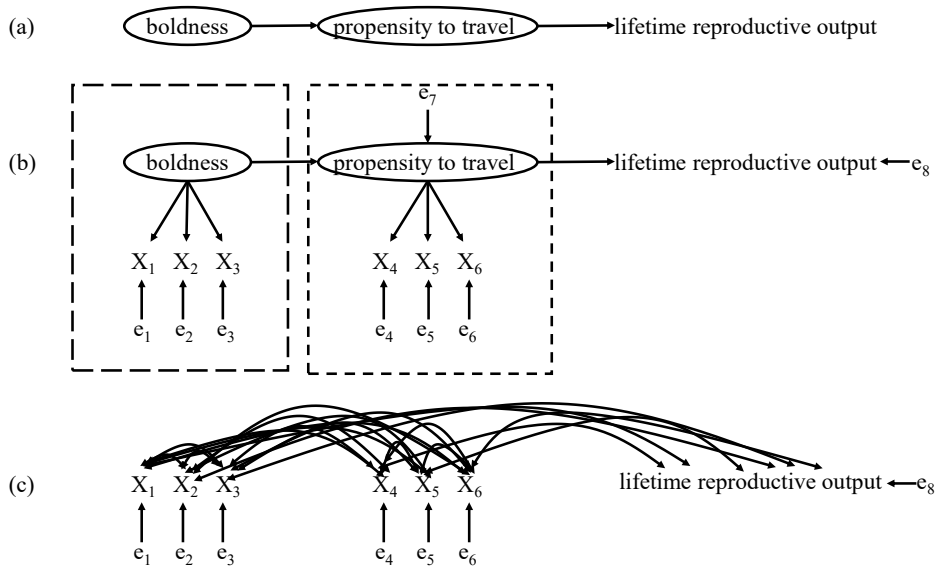


Figure 7.1. (a) The causal hypothesis of interest. (b) the two measurement models (inside the boxes) used to test the causal hypothesis of interest. (c) The mixed acyclic graph (MAG) that results if we treat the two latent variables (boldness, propensity to travel) as implicit.

For instance, ethologists define "boldness" as a personality trait that differs between individuals and that describes an animal's tendency to take risks or engage in potentially dangerous situations (Réale et al. 2007). It reflects how an animal behaves in uncertain or threatening contexts, such as encountering predators, exploring novel environments, or interacting with unfamiliar objects or individuals. Bold animals are generally more willing to approach or investigate potential risks, while shy or cautious animals avoid them. Boldness is hypothesised to influence various aspects of an animal's ecology, such as its foraging strategy, territory selection, social dynamics within its group, and affect survival and reproduction. It is observed across many animal species, including fish, birds and mammals. Imagine that an ethologist hypothesises that individuals who are "bolder" will have a propensity to travel further distances from their nests and that these greater distances will result in a greater number of offspring

during its life (Figure 7.1a). There does not exist any scientific equipment that can accurately measure the degree of "boldness" of an individual. You cannot directly observe the "boldness" of an animal in the same way that you can observe its length. Boldness is a latent variable that is unobservable even in principle. Typically, ethologists will devise a series of different standardized experimental setups that are designed to measure certain aspects of "boldness" but with substantial measurement error. For instance, they might place an individual in a large box containing a novel object and measure how long the individual takes before approaching the object. The results of each of these experiments are directly observed and so are not latent but the thing they are supposed to measure – boldness – cannot be directly observed. Similarly, "propensity to travel" might be measured by a series of capture-recapture events in a grid of traps. At each recapture, the biologist would record the distance from the nest. The actual propensity to travel is not directly observed; only the results from the series of capture-recapture events are observed variables and these observations are only imperfect indicators of this latent propensity. This full causal hypothesis is shown in Figure 7.1b in which the observed variables X_1 to X_3 are the results from three experiments to measure boldness and the observed variables X_4 to X_6 are the results from three capture-recapture events. We would expect that all six observed variables are simply imperfect measures of the latents to which they are responding and that each observed variable has substantial measurement error.

The variables of real interest to our ethologist are boldness and propensity to travel but he cannot directly observe them. The variables that our ethologist can actually observe (X_1 to X_6) are not really of interest except to the extent that they provide information about the latent variables. Figure 7.1c shows the MAG that results from making these latent variables implicit. Although correct, this MAG is useless to our ethologist for at least two reasons. First, every observed variable is joined to every other observed variable by a double-headed arrow and so there are no causal hypotheses to test; there would be no d-separation claims in the union basis set and no degrees of freedom in the covariance-based SEM. Second, our ethologist is primarily interested in the causal relationships between the two latents (boldness, propensity to travel) and lifetime reproductive output (Figure 7.1a); the act of making the two latents implicit means that we have lost all information about them. Making latents implicit, as described in Chapter 6, is reasonable as long as the observed variables, not the latents, are not of primary concern and as long as the act of marginalizing the latents does not remove too many of the testable causal implications

among the observed variables. When this is not the case then we can keep the latent variables explicit in covariance-based SEM under certain conditions. We do this by linking each latent to each of the observed variables that are its direct effect via a "measurement model". The dashed boxes in Figure 7.1b isolate the two measurement models for the two the latents.

7.2 Developing a measurement model using a theoretical cause construct and its observed effect indicators

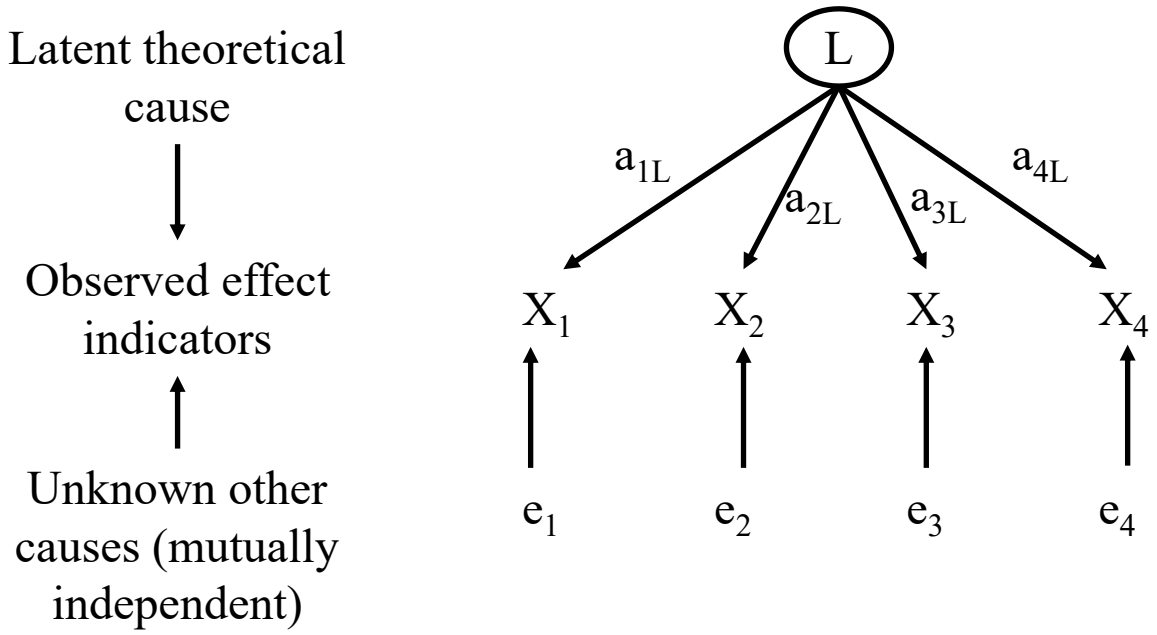


Figure 7.2. The general structure of a measurement model.

A measurement model is a type of confirmatory factor analysis[168]. A measurement model uses a DAG that has the general structure shown in Figure 7.2. The model has three parts. The first

[168] There are two main types of factor analysis: confirmatory and exploratory. In confirmatory factor analysis, one specifies the number of latent factors and then tests this hypothesized structure against the data. In exploratory factor analysis, one allows the data to suggest the number of latent factors. Factor analysis is statistically like principal components analysis except that PCA attempts to capture as much of the total variance as possible in the first few axes and doesn't differentiate between the shared variance due to the underlying factors and the unique (error) variance. while CFA attempts to separate the shared variance from the unique (error) variance and forces all of the variance to exist within the chosen number of axes.

part is the explicit latent theoretical cause; other names for such a latent variable in SEM jargon are "theoretical constructs" or "factors". A latent theoretical cause is a property of our experimental units that we hypothesize to exist (thus, a "construct") but that we cannot directly observe, like the "boldness" of an individual. In fact, a major reason for including a measurement model in our causal graph is so that we can test this hypothesis and, therefore, obtain support for the existence of this hypothesized property. The second part involves a series of observed variables, called observed effect indicators[169]. We choose these effect indicators in order to measure aspects of our theoretical cause construct. We hypothesise, based on our understanding of the nature of the theoretical construct, that the effect indicators are each direct observable effects (causal children) of this unobservable latent cause. Normally[170], we choose the effect indicators to be (i) caused only by this single latent theoretical cause and (ii) to be the ones that are most strongly correlated with the latent theoretical cause. A wise choice of these effect indicator variables is essential to a successful measurement model and this choice is guided by a clear definition of the latent theoretical cause; I will give an example later. The third part of the measurement model involves the error variables (the e_i in Figure 7.2). Each error variable represents all of the unknown or unmodelled causes of its associated observed effect indicator besides the latent theoretical cause. Normally[171], we want the error variables to be mutually independent, meaning that the unknown causes of each observed indicator are unique to it and do not also cause changes in the other observed indicator variables. I will use the hypothetical example of an ethologist who wishes to develop a measurement model of "boldness" for individuals of a particular species.

The first step is conceptual. One must define, as clearly as possible, what it means for an individual to be "bold". Let's start with the definition that I gave previously: *"boldness" is a personality trait that differs between individuals and that describes an animal's tendency to take risks or engage in potentially dangerous situations when in uncertain or threatening contexts*. It is equally important to clearly define what the theoretical cause construct is not. For instance, we would only want to include behaviours that are repeatable, not ones that are transient or that

[169] Most biologists are more familiar with the term "proxy" variable.
[170] If this is not the case, then you would have an arrow from each latent theoretical cause to this observed effect indicator, but this reduces the degrees of freedom.
[171] If this is not the case, then you can add a free covariance (i.e. a double-headed arrow) between the error variables, but each such free covariance reduces the degrees of freedom.

change over time[172]. This is implicit in the phrase "personality trait". This definition uses the word "*describes*", which is ambiguous. If we are making a claim about causality then we should say that this personality trait "…*causes an animal's tendency*…" and this implies that this tendency is not simply our perception of how the animal behaves but has some physical basis in the animal's brain, even if we don't know the mechanism by which this physical property causes behaviour. A more explicit definition might be: *"boldness" is a stable characteristic or disposition that influences the animal's behaviour across a wide range of situations over time, that differs between individuals, and that is a cause of an animal's tendency to take risks or engage in potentially dangerous situations when in uncertain or threatening contexts*. It is possible – and maybe likely – that the theoretical definition will change as researchers further develop the notion of animal "boldness".

The next step is to use the definition of this theoretical cause construct to devise measurement "instruments", i.e. the observable effect indicators. The goal is to choose variables (effect indicators) that are direct causes of this unobservable latent variable but that are not also being caused by any other unobserved variable. If this is not true then our measurement model will be rejected when we test it against empirical data, providing that we have sufficient statistical power (Chapter 5). For example, imagine that we place each individual in the same unfamiliar space, a box, whose roof has a window in the middle so that the animal can see out. If we place our animal in in a corner of the box, then, according to our definition of "boldness", bolder animals will explore this space more quickly. The first effect indicator variable (X_1), which is an observed variable, could therefore be the time, in seconds, until the individual begins to move. Bolder animals should begin to move more quickly. Bolder animals should spend more time in the open area under the window compared to shyer animals and so this could be the second effect indicator variable (X_2). If an unfamiliar object is introduced into the animal's environment, a bolder animal should approach or interact with the object more quickly. A third effect indicator variable (X_3) would be the time, in seconds, until the animal interacts with the object. If a model predator (an image or a statue) is introduced and slowly moved toward the animal then the flight initiation distance, in centimetres, can be measured as the distance at which the animal begins to display predator avoidance behaviours (freezing, fleeing or hiding). A bolder animal would have

[172] A further refinement might want to clearly specify how much variation in the behaviour is acceptable and for what time span.

a shorter flight initiation distance, defining the fourth effect indicator variable (X_4). None of these effect indicators, used separately, can completely capture the notion of "boldness" but each would capture aspects of it. Each effect indicator alone is likely to be only partially correlated with "boldness" (i.e. each has significant measurement error), but several of them together might better capture the notion of "boldness" and, together, would provide a better estimate of the position of an individual along the one-dimensional axis between shy and bold. All these effect indicators will be correlated if they are all being caused by the same thing, this "stable characteristic or disposition that influences the animal's behaviour", but each will become independent of the others once we hold constant (i.e. condition on) this latent "boldness". Note also that these different effect indicators of the latent "boldness" have different measurement units and can have different degrees of correlation with the latent "boldness".

7.3 Translating the measurement model DAG into a covariance-based SEM

We already have measurement units for our observed effect indicators. What units do we want to assign to "boldness"? Do we measure boldness in seconds? Centimetres? Do we invent a new scale? Before we derive the model-predicted covariance matrix, we must decide which units we want to use when measuring the theoretical cause latent. We have two choices. These two choices are exactly equivalent in terms of the test of model fit, but they will result in different values for the free parameters[173]. The free parameters in Figure 7.2 are the four path coefficients plus the variances of the latent and the four error variables. The first choice is to assign the same measurement unit to the latent cause as used in one of the observed effect indicators. You would do this by fixing the path coefficient from the latent to that effect indicator to unity; in other words, a unit increase in the effect indicator is caused by the same unit increase in the latent. This reduces the number of free parameters by one and this choice makes sense if your definition of the theoretical cause construct implies this measurement unit. The second choice is to treat the theoretical cause latent as a standardised normal variable; that is, a variable with a variance

[173] This is only to be expected since, even in a multiple regression, any change in measurement units will change the slopes.

of unity and with measurement units of "standard deviations from the mean". This second choice also reduces the number of free parameters by one and makes sense if your definition of the theoretical cause construct does not imply any logical measurement unit or, at least, not a measurement unit that is used in any of the effect indicators. Since our definition of "boldness" does not imply any particular measurement units, and units of time or distance do not make sense, I will use this second option and fix the variance of the latent "boldness" to 1. Here are the structural equations associated with our measurement model in Figure 7.2:

$$L_i = N(0, \sigma_L = 1)$$
$$e_1 = N(0, \sigma_1); e_2 = N(0, \sigma_2); e_3 = N(0, \sigma_3); e_4 = N(0, \sigma_4)$$
$$X_{1i} = a_{1L} L_i + e_{1i}$$
$$X_{2i} = a_{2L} L_i + e_{2i}$$
$$X_{3i} = a_{3L} L_i + e_{3i}$$
$$X_{4i} = a_{4L} L_i + e_{4i}$$
$$Cov(e_{1i}, e_{2i}) = Cov(e_{1i}, e_{3i}) = Cov(e_{1i}, e_{4i}) = Cov(e_{2i}, e_{3i}) =$$
$$Cov(e_{2i}, e_{4i}) = Cov(e_{3i}, e_{4i}) = Cov(L_i, e_{1i}) = Cov(L_i, e_{2i}) =$$
$$Cov(L_i, e_{3i}) = Cov(L_i, e_{4i}) = 0$$

Using the logic of covariance-based SEM that I described in Chapter 4, we can now obtain the model-predicted covariance matrix of our observed effect indicators. Each element of the model-predicted covariance matrix is a function of variances/covariances of exogenous variables and path coefficients. There are five exogenous variances in Figure 7.2: the variance of the latent (which we have fixed at 1) and the error variances of each of the observed effect indicators (e_i). There are four path coefficients in Figure 7.2 (a_{iL}). In SEM jargon, these path coefficients from a latent cause to an observed effect indicator variable are also called "factor loadings". If you use the rules of covariance algebra that I explained in Chapter 4, then you can obtain the model-predicted covariance matrix:

$$\begin{bmatrix} 1a_{1L}^2 + e_1^2 & 1a_{1L}a_{2L} & 1a_{1L}a_{2L} & 1a_{1L}a_{4L} \\ 1a_{1L}a_{2L} & 1a_{2L}^2 + e_2^2 & 1a_{2L}a_{3L} & 1a_{2L}a_{4L} \\ 1a_{1L}a_{2L} & 1a_{2L}a_{3L} & 1a_{3L}^2 + e_3^2 & 1a_{3L}a_{4L} \\ 1a_{1L}a_{4L} & 1a_{2L}a_{4L} & 1a_{3L}a_{4L} & 1a_{4L}^2 + e_4^2 \end{bmatrix}.$$

Don't worry if you couldn't derive this matrix! You don't have to know this since lavaan does the work for you. I simply want you to look at that the functions in the diagonal elements

of this model-predicted covariance matrix, i.e. the predicted variances of each of the observed effect indicators. Notice that each is composed of two parts. For instance, the variance of the first observed effect indicator is $1a_{1L}^2 + e_1^2$. The first part ($1a_{1L}^2$) is the variance of the latent variable (here, fixed at 1) times the square of the causal effect of the latent on the effect indicator (a_{1L}^2). This is called the *common* variance of the first effect indicator because it is the variance of X_1 that is due to the latent cause, which is the common cause of all of the effect indicators. The second part (e_1^2), the error variance of X_1, is the variance of X_1 that is due to the other unknown causes of X_1 but *not* due to the common latent cause and also *not* due to the other unknown causes of the other effect indicators. We know this because there are no arrows or double-headed arrows joining e_1^2 to any of the other error variances[174]. This second source of variation is called the *unique* variance of the first effect indicator. If $1a_{1L}^2$ gives the variance of this first effect indicator that is due to the common latent variable and the total variance of X_1 is $1a_{1L}^2 + e_1^2$, then the ratio of the two gives the proportion of the total variance of X_1 that is due to the latent variable; i.e. the R^2 value for X_1. Therefore, the Pearson correlation coefficient between an observed effect indicator variable and the common latent cause is $R = \frac{\sigma_L a_{1L}}{\sqrt{\sigma_L^2 a_{1L}^2 + e_L^2}}$. In our example, in which I have fixed the latent variance to unity, this reduces to $R = \frac{a_{1L}}{\sqrt{a_{1L}^2 + e_L^2}}$.

Therefore, we can measure how strongly each observed effect indicator is correlated with the unmeasured (latent) common cause. The square of this value (R^2) is called the "reliability" of the effect indicator. Values closer to unity mean that the effect indicator is more tightly correlated to the latent variable that is the common cause of all of the effect indicators; therefore, the effect indictor is a more reliable proxy for the unmeasured latent variable. Practitioners of factor analysis have devised various ways of summarizing the combined reliability provided by the full set of effect indicators and these are available using the `comprelSEM()` function of the semTools package[175] on CRAN. One widely used index in sociology is called Cronbach's

[174] It is possible to add double-headed arrows between the error variances, but each such double-headed cause reduces the degrees of freedom of the model by 1.
[175] Lavaan does not output values of these combined reliability indices.

alpha[176], and it measures the degree to which responses are consistent across the different effect indicators for a given measurement model (Kline 2016). Cronbach's alpha varies between 0 and 1 and increases as (i) the average correlation between the effect indicators and the common latent cause increases and (ii) as the total number of effect indicators used to measure the common latent cause increases. Values of 0.7 or more are generally viewed as necessary to reliably capture the latent cause.

In other words, this measurement model separates the total variance of each observed effect indicator into parts: (i) the variance that is common to all of them due to the common theoretical cause and (ii) the variance that due to the different (unknown) causes of each and which is unique to each. The off-diagonal elements of our model-predicted covariance matrix (i.e. the covariances between the observed effect indicators) give the predicted covariances between each pair of observed effect indicators. These covariances between each pair of observed effect indicators are entirely due to the common effect of the latent cause on the two effect indicators. For example, the predicted covariance between the first two effect indicators ($1a_{1L}a_{2L}$) is a function only of the variance of the common latent cause (here, fixed at 1) and the product of the path coefficients from the common latent cause to each. This covariance is only generated by the causal effect of the common latent cause on each of the observed effect indicators.

Now, following the logic explained in Chapter 4, we use maximum likelihood estimation to choose values for the free parameters so that the predicted covariances in this model-predicted covariance matrix are as close as possible to the observed covariance matrix. We have eight free parameters in the model-predicted covariance matrix (four path coefficients and four error variances) and we have V(V+1)/2 = 4(5)/2 = 10 unique values in the observed covariance matrix. We have two remaining degrees of freedom[177]. We can therefore test if the hypothesized causal structure shown in Figure 7.2 is contradicted by the observed data. If the resulting null probability is lower than our chosen significance level, then we must reject our causal structure

[176] Cronbach's alpha is not calculated in lavaan but this can be obtained from the semTools library on CRAN via the `compRelSEM(fit,tau.eq=TRUE)` function, where fit is the object output from the `sem()` function. Alternatively, this can be closely approximated by the formula $\alpha_C = \frac{n\bar{r}}{1+(n-1)\bar{r}}$ were n is the number of effect indicators for the common latent cause and $\bar{r}$ is the average correlation between an effect indicator and the common latent cause, as explained in the main text.

[177] As explained in Chapter 4, df=V(V+1)/2 – (p + q) = 10-(4+4) = 2.

and conclude that the chosen effect indicators are not caused by a single latent variable. This means that our chosen effect indicators are not really responding only to the same single latent cause. Since we have chosen these effect indicators based on our definition of this latent cause, then the rejection of this measurement model means that our chosen effect indicator variables are not really measuring the latent variable as it has been defined. If the resulting null probability is higher than our chosen significance level, then we can provisionally accept that these chosen effect indicators are measuring our theoretical cause latent.

If we have a well-fitting measurement model (and since we have separated the error variances of each effect indicator from the variance that is due to the same common cause) then can we obtain the actual values of the underlying latent cause? No[178]. However, we can obtain more accurate estimates of the values of the latent common cause than are provided by each effect indicator alone by combining the estimates together. These estimates are called "factor scores" in SEM jargon. To do this, we calculate a weighted sum of the estimate provided by each effect indicator. The weight assigned to each effect indicator (i) increases as the strength of the association between the effect indicator and the theoretical latent increases (we know this from the R^2 values) and (ii) the weight decreases as the magnitude of the error variance of the effect indicator increases (we know this from our estimated error variances). Several weighting methods have been proposed but the most common one, and the one implemented in lavaan, is the regression method[179] (Thurstone 1935). Assuming that the measurement model fits the data, then increasing the number of effect indicators will improve the accuracy of the resulting factor scores, as will choosing effect indicators with stronger correlations with the latent cause, i.e. choosing effect indicators with smaller unique error variances. However, don't make the mistake of thinking that these factor scores are the same as the latent cause itself; the factor scores are simply better estimates of it than are provided by any of the effect indicators alone.

7.4 Fitting the measurement model using lavaan

[178] This is called factor indeterminacy (Bollen 1989).
[179] These various methods are all highly correlated in practice (Bollen 1989).

We need some empirical data in order to fit our measurement model using lavaan. It makes no sense to demonstrate this using empirical data because, by definition, we don't have direct measurements of the latent variable. Instead, I will simulate data using the DAG in Figure 7.2 and the example of estimating "boldness". Here is the code to simulate values of our four observed effect indicators (X_1 to X_4) given the latent variable "boldness":

```
N<-500
set.seed(101)
#latent "boldness"
boldness<-rnorm(N)
#time (seconds) until the individual begins to move.
X1<-0.4*boldness+rnorm(N,0,sqrt(1-0.4^2))
X1<-abs(min(X1))+X1
#time (seconds) spent in the open area
X2<-0.5*boldness+rnorm(N,0,sqrt(1-0.5^2))
X2<-abs(min(X2))+X2
#time (seconds) until the individual interacts with object
X3<-0.6*boldness+rnorm(N,0,sqrt(1-0.6^2))
X3<-abs(min(X3))+X3
#flight initiation distance (cm)
X4<-0.7*boldness+rnorm(N,0,sqrt(1-0.7^2))
X4<-abs(min(X4))+X4
dat<-data.frame(X1,X2,X3,X4)
```

Everything you learned about lavaan in Chapter 4 applies when you include explicit latent variables in your model. In fact, you already know almost everything you need to include latent variables into the model object of lavaan. You only need a few new pieces of information. First, the name for the latent variable in the model object can be any valid name in R that does not already exist in the object containing the empirical data (our simulated data, called `dat`). I will call the latent variable `bold` in the model object since this name is not found in `dat`. Second, lavaan uses the "=~" operator to define a latent cause variable. The variable on the left of this operator is the latent variable and the variables on the right are the causal children of the latent cause. We say that this latent cause variable is "measured by" the effect indicators on the right of the operator. Thus, a line like `bold=~X1+X2+X3+X4` in a model object tells lavaan that `bold` "is measured by" – or, equivalently, "is causing" – the effect indicator variables X_1, X_2, X_3 and X_4. Since `bold` is not a name in the data frame (`dat`), lavaan knows that `bold` is latent. Be careful; this formulation is confusing since the tilde operator (Y~X) in lavaan means that "Y is

caused by X" while the "Y =~X" operator in lavaan means "Y is a latent cause of X". By default, if lavaan sees a line like `bold=~X1+X2+X3+X4` then it assumes that the path coefficient associated with the first variable on the right (X_1) is fixed to unity (to fix the measurement scale of the latent `bold`) and the other path coefficients are free parameters. To be more explicit, you could equivalently write this as `bold=~1*X1+X2+X3+X4`. If you write the line as `bold=~NA*X1+X2+X3+X4` then this tells lavaan *not* to fix the path coefficient associated with X_1 to unity but to treat it, along with the path coefficients for X_2, X_3 and X_4 as free parameters. If you do this then you will have to fix the measurement scale of `bold` by fixing its variance to unity (`bold~~1*bold`).

I said that you can fix the scale of the latent variable either by fixing the path coefficient from the latent to one of the effect indicator variables to unity (`mod1`) or by fixing the variance of the latent to unity (`mod2`). Since we have chosen to fix the scale of the latent to unity in `mod2` we must add the line `bold~~1*bold`. Here is how you would do each of these options:

```
mod1<-"
bold=~X1+X2+X3+X4
"

mod2<-"
bold=~NA*X1+X2+X3+X4
bold~~1*bold
"
```

We fit the data to the model using the `sem()` function and obtain the result using the `summary()` function in exactly the same way as you learned in Chapter 4: `fit2<-sem(mod2, data=dat)`. The model fit statistics that are output from the `summary()` function are identical using either `mod1` or `mod2` because these two model formulations differ only in the measurement units associated with the latent variable `bold`; `mod1` scales the latent `bold` in seconds, since these are the measurement units of X_1, while `mod2` scales the latent `bold` in standard deviations from the mean. Since our theoretical definition of "boldness" does not make reference to time (thus measurement units of seconds), I will use the `mod2` output. Here are the model fit statistics from `summary(fit2)` which tell us what we already know; namely, that the data agree with the causal structure of Figure 7.2:

```
Model Test User Model:
```

```
  Test statistic                                0.745
  Degrees of freedom                                2
  P-value (Chi-square)                          0.689
```

Here are the estimates of the free parameters:

```
  bold =~
    X1                0.433    0.052    8.265    0.000
    X2                0.497    0.055    9.025    0.000
    X3                0.550    0.054   10.105    0.000
    X4                0.676    0.057   11.789    0.000

Variances:
                  Estimate  Std.Err  z-value  P(>|z|)
    bold             1.000
   .X1               0.745    0.055   13.559    0.000
   .X2               0.782    0.060   12.935    0.000
   .X3               0.677    0.058   11.599    0.000
   .X4               0.536    0.066    8.122    0.000
```

If you compare the estimates of the path coefficients (the factor loadings) to the values that I used to simulate the data, you will see that the true values of all of these estimates are within their 95% confidence intervals (i.e. estimate±1.96SE). For instance, the 95% confidence intervals of the first path coefficient[180] are (0.331, 0.535) while the true value is 0.5. In other words, if the data were really generated according to our hypothesised measurement model, then these path coefficients are unbiased estimates. Later, when we allow latent variables to cause other latent variables, or to allow observed variables to cause the latent, then we can use these unbiased estimates and the rules for calculating direct, indirect and total causal effects (Chapter 3) to get unbiased estimates of these causal effects.

The estimates of the unique variances of the observed effect indicator variables are the residual variances (i.e. measurement errors); i.e. the variance of each effect indicator that is not caused by the latent "boldness". All of these residual variances are significantly different from zero, as shown by the column named "P(>|z|)", meaning that there is significant measurement error associated with each. We expect these residual variances to be significantly greater than zero. If any are not different from zero, then this means that the effect indicator might be perfectly correlated with the theoretical latent. If this were the case, then we could simply replace the latent with this effect indicator and forget about the measurement model! To see the proportion

[180] 0.433±1.96(0.052)

of the variance of each effect indicator that is caused by the common latent "boldness", we can add the `rsquare=T` argument to the `summary(fit2, rsquare=uniqueT)` function. The result is:

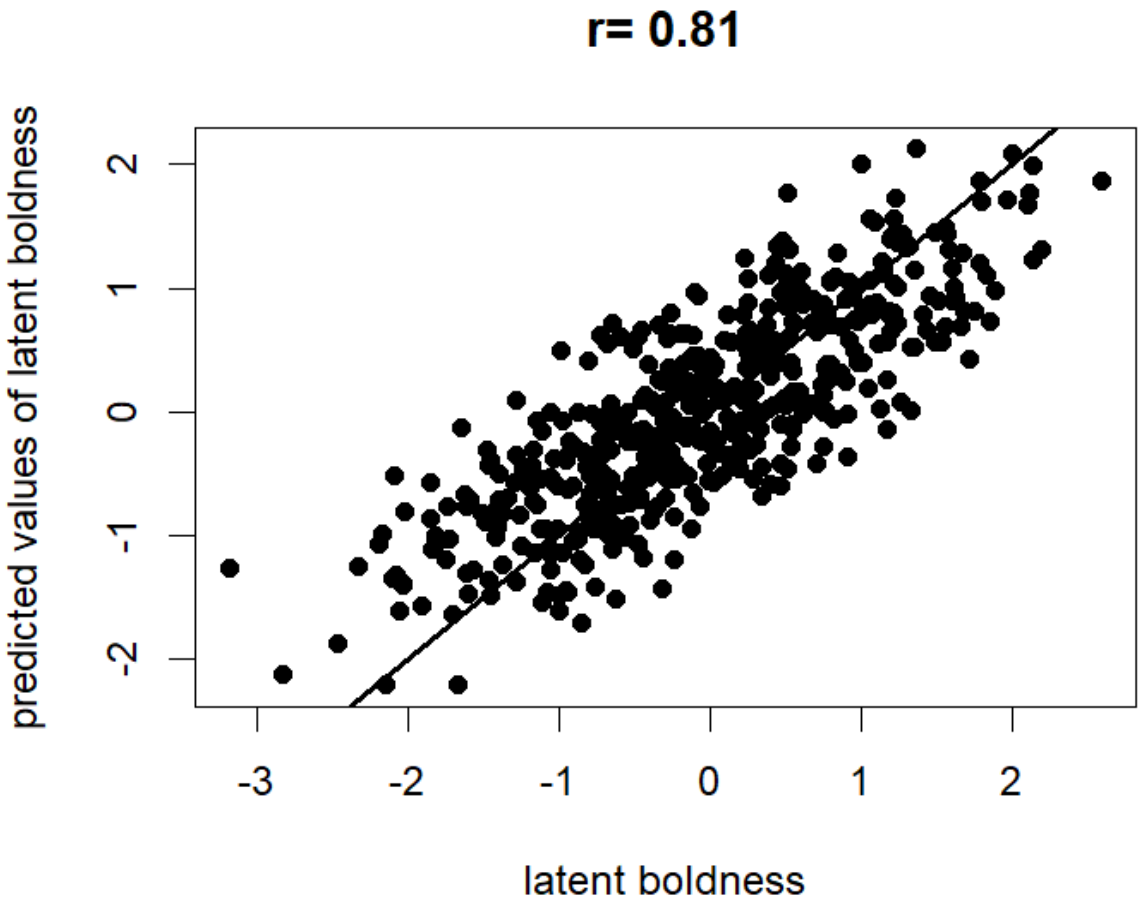
```
R-Square:
                 Estimate
    X1              0.201
    X2              0.240
    X3              0.309
    X4              0.460
```

The square root of these R^2 values gives the Pearson correlation coefficient between each effect indicator and the latent "boldness". Clearly, none of our observed effect indicators is a good proxy for the latent "boldness" since between only 20% to 46% of the total variance of any of them is associated with the latent. The highest correlation ($\sqrt{0.460} = 0.678$) is with X_4 (the minimum flight distance). If we had found that one effect indicator was very strongly correlated, then we could have simply replaced the latent "boldness" by this effect indicator, but this is not the case in our example.

Since none of our effect indicators is very strongly correlated with the latent "boldness", we can combine these four effect indicators together to get a better estimate of the latent "boldness" than is provided by any one of them alone. If we calculate Cronbach's alpha, explained above, then we get a value of 0.63. Since this value is less than 0.7, it tells us that our latent "boldness" is not particularly well captured by our four chosen latents. This does not mean that our model is wrong but if we want good estimates of the latent "boldness" then we would either have to include more effect indicators or else identify better ones.

Lavaan uses the most common "regression" method to estimate the values of the latent cause (i.e. factor scores) given your sample data. These estimates are produced by the `predict(fit2)` function of lavaan, which returns a matrix with as many columns as there are explicit cause latents in the model; here `fit2` is the name of the object returned when fitting our model using `sem()`. Remember that these predicted values are simply sample estimates of the true latent variable. You can get the standard error of these estimates as the square root of the variance of the latent variable, which is output in the `summary()` function. You can get a

prediction equation that reproduces these predicted values by regressing[181] these predicted values against the values of the observed indicators. Since we have used simulated data, we actually know the "true" values of our latent variable (`boldness`) and we also know the estimated factor scores for the modelled latent variable in the model (`bold`). Figure 7. 3 plots the true values of our latent "boldness" to the estimated values produced by `predict(fit2)`. The correlations between the true values of our latent "boldness" and each of the observed effect indicators (X_1, X_2, X_3, X_4) are 0.431, 0.447, 0.553, and 0.715. The correlation between the true values of our latent "boldness" and our estimated latent `bold` is 0.81. This shows that our predicted value of the latent variable "boldness" is not the same as the true latent "boldness" but that it is a better estimate than any of the effect indicators used singly.



[181] the R^2 of this multiple regression will always be 1.0

Figure 7.3. The true values of the latent "boldness" of an animal plotted against the estimated values of the latent variable bold produced by the measurement model.

7.5 When to use (and when not to use) a measurement model

In Chapter 6, I distinguished between variables that are latent but observable in practice and variables that are latent and unobservable in practice. The variable "boldness", as an attribute of animal behaviour, is clearly unobservable in practice since we don't have any way of accurately measuring it and must rely on imperfect effect indicators. In truth, the phrase "latent but observable in practice" is too imprecise and "air temperature" is a good example of why this is so. A better phrase might be "latent but replaceable by an accurate effect indicator". It seems obviously true that we can directly observe, and accurately measure, air temperature simply by looking at a mercury thermometer or the output of a thermocouple of thermistor. That this claim is obviously true is an illusion. The current physical definition of temperature is that "temperature" the average kinetic energy of the molecules in a substance. However, no one can directly observe or measure the kinetic energy of each of the molecules in the air. "Air temperature" is a theoretical concept[182] that is unobservable even in practice but is replaceable by an accurate effect indicator. When we read a mercury thermometer, we are not observing or measuring the "average kinetic energy of the molecules in the air". Rather, we are measuring the height of a column of mercury inside a closed glass tube that is in a partial vacuum. This height is then converted to an arbitrary[183] scale of degrees Celsius or Fahrenheit. Thermometers can also be constructed using alcohol or even water rather than mercury, although mercury has certain properties that make it less prone to error (McCaskey 2020). When we measure temperature using the output of a thermocouple, we are measuring the voltage generated at the junction of two dissimilar metals which is then converted to degrees Celsius or Fahrenheit.

[182] This is often called a latent "construct" in the SEM literature.

[183] 1°C is simply the difference in height of the column of mercury between the temperature when freshwater freezes (an arbitrary zero) and when it begins to boil at sea level, divided by 100. °F is equally arbitrary since 0°F is the temperature at which sea water freezes. Degrees Kelin is not completely arbitrary because 0°K is the temperature at which the total kinetic energy of the substance is zero although the choice of units of each °K is arbitrary.

When we measure temperature using the output of a thermistor, we are measuring the resistance of electrons in the thermistor material which is then converted to degrees Celsius or Fahrenheit. The outputs of thermometers, thermocouples and thermistors are all observable effect indicators of the underlying latent concept of "temperature".

So why do we commonly treat "temperature" to be an observed variable rather than an unobservable theoretical construct? It is mental slight-of-hand. Many independent studies have shown that these effect indicators only respond to this one theoretical concept[184] and to have very small measurement errors relative to the natural variation of the unobservable "average kinetic energy of the molecules in the air". Because of this, we take a mental shortcut and equate the unobservable theoretical concept ("temperature") with the measurements of these reliable effect indicators. (McCaskey 2020) has described the long scientific journey, and the continuous interplay between theoretical conception and practical measurement, that led to the modern theoretical concept of "temperature". Phrased in terms of a measurement model, we have simply found effect indicators that are so highly correlated with the theoretical cause latent that we choose to ignore the minor measurement error and mentally equate the latent with the indicator. The fact that one would not use a mercury thermometer to measure temperature differences of (say) $1/100^{th}$ of a degree C, where measurement error would overwhelm the true differences in temperature, reveals this mental slight-of-hand.

When is this mental slight-of-hand justified? When is it not? When must we include measurement models in our causal models and when can we simply choose one good effect indicator and mentally equate the cause latent and its observable effect indicator? Biologists perform this mental slight-of-hand all the time, even though they are often unaware of doing this. Consider the well-known process of maintaining body temperature in homeotherms. Ambient air temperature affects the metabolic rate of animals. When it is cold, a homeothermic animal must burn stored energy reserves, first glycogen and fat and then, when these are exhausted, protein, in order to generate heat and maintain its body temperature. The scaling of surface area (the site of heat loss to the atmosphere) to body volume (where the heat is generated) means that small homeothermic animals like songbirds can lose up to 15% of their body fat in one cold night. To burn this fat the bird must increase its metabolic rate, which generates heat. Imagine that we

[184] Assuming that they have been correctly constructed and calibrated.

conduct an experiment in which we place a small bird inside a metabolic chamber overnight and decrease the air temperature sequentially from 25°C to 0°C. We note the air temperature after each decrease, measure the metabolic rate of the bird, and weight it. The hypothesised causal process is: decrease in air temperature → increase in metabolic rate → decrease in fat reserves (Figure 7.4). There are no unobserved variables in this causal explanation, right?

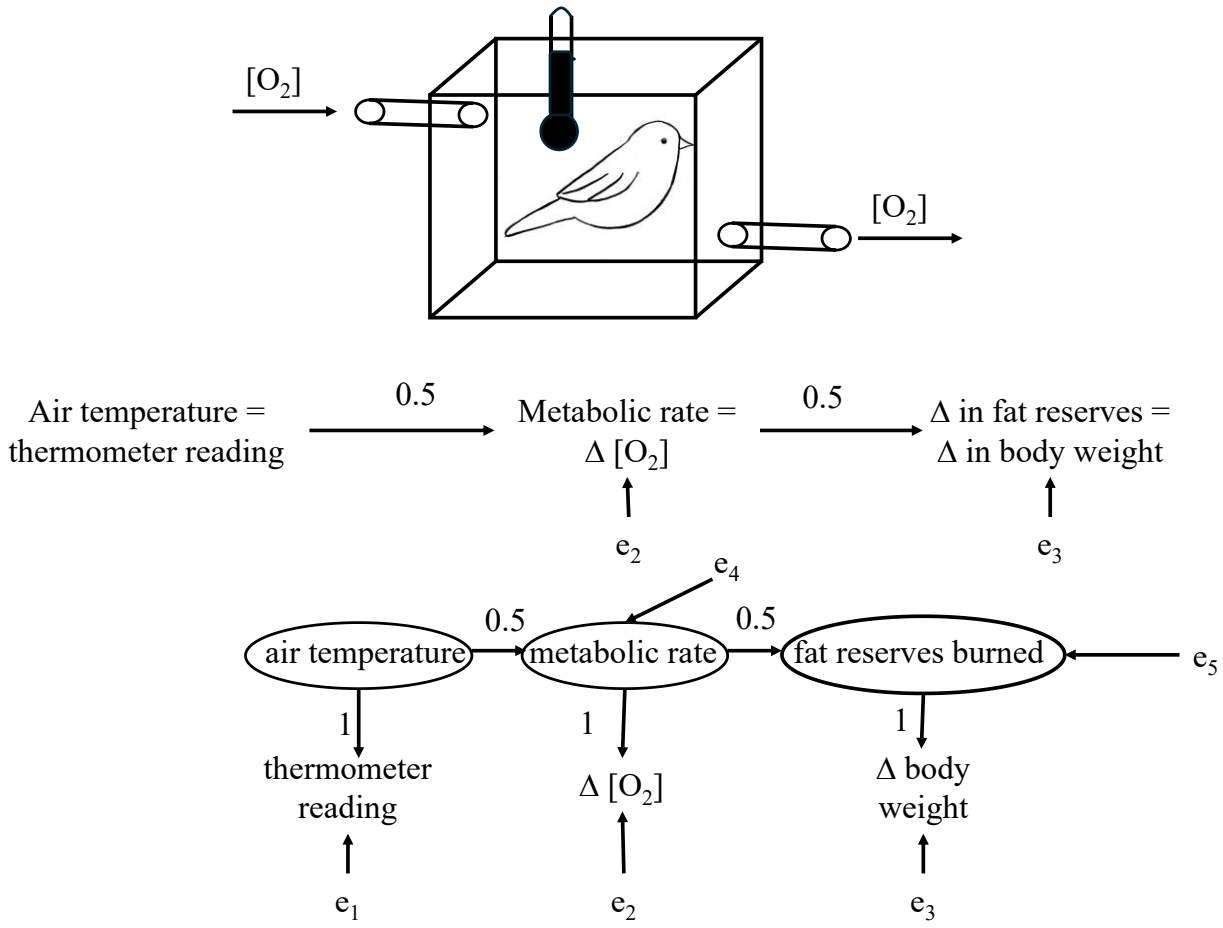


Figure 7.4. An experiment to study how changes in air temperature cause changes in fat reserved in the body of a small passerine bird by increasing its metabolic rate. The DAG at the bottom shows the actual causal structure. The DAG in the middle is the DAG assuming that each effect indicator is equivalent to its latent cause.

Wrong. Not only is air temperature an unobservable variable, but so is metabolic rate and the decrease in fat reserves. Temperature is the average kinetic energy of the molecules in the air, but we can't see the molecules banging into each other, and this is not what we are actually measuring, as explained earlier. We can't directly measure metabolic rate either. Metabolic rate is the rate at which an organism uses energy to maintain physiological processes; that is, the number of ATP molecules that are converted to ADP molecules in each cell of the body per unit

time, with the resulting release of energy. We have no way of directly observing and counting these biochemical reactions. Typically, one measures the rate of gas exchange (oxygen decrease or carbon dioxide increase) between the air entering and leaving the metabolic chamber[185]. If we measure oxygen consumption using an infrared gas analyser (IRGA), then we aren't even directly measuring *oxygen* consumption. Instead, we are measuring differences in the amount of light of particular wavelengths that is absorbed by the oxygen molecules as the light passes through the air. When we measure the fat reserves that are burned by the birds, we might actually be measuring the difference in body weight over the course of the experiment using a digital scale, and this too will include measurement error. A digital scale is measuring the change in electrical resistance of a load cell caused by the force of gravity.

In this causal scenario, the variables that we can observe and measure are not the variables that we hypothesise to form the causal chain of interest. The causal process involves latent variables that we cannot directly observe even we wanted to. The variables that form the causal chain of interest are theoretical constructs. We are not really interested in the variables that we can directly measure (the height of the column of mercury, the decrease in the concentration of O_2 between the air entering and leaving the chamber, the compression of the load cell in the balance) except to the extent that they help us to quantify the unobservable theoretical constructs. However, the animal ecophysiologist who is doing this experiment will mentally equate these effect indicators with their underlying theoretical constructs and treat the two as the same. In other words, he would mentally use the DAG in the middle of Figure 7.4 rather than the more complete DAG at the bottom of Figure 7.4.

When is this reasonable? What happens when we mentally equate the three observable effect indicators with each of their latent theoretical concepts? What happens as the observable effect indicators become less and less strongly correlated with their associated latent variables? You can glean a first indication by asking what would happen if the error variances (e_i) in the DAG at the bottom of Figure 7.4 that are associated with the effect indicators were actually zero? In other words, when there is no residual variation in the effect indicator after accounting for the

[185] The conversion of ATP to ADP involves the release of a phosphate group, which releases the chemical energy in the bond. This process does not directly consume oxygen. However, during the final stage of oxidative phosphorylation, oxygen acts as the terminal electron acceptor in the electron transport chain, helping to drive the rejuvenation of the ADP to ATP.

variation in the latent cause. In this case, the latent variable is perfectly correlated with its effect indicator and the values of the two will be identical up to a scaling constant. Applying the concept of d-separation, holding constant the effect indicator in this case also means holding constant its theoretical latent cause. In this case it is reasonable to mentally equate the observable effect indicator and its latent cause because the two are numerically equivalent. What happens as we increase the ratio of the residual variation to the total variation of the effect indicator? We will consider the two main consequences of this increase. The first consequence is on the fit statistics of the causal model, i.e. the null probability of the maximum likelihood X^2 statistic for covariance-based SEM[186]. The second consequence is on the estimation of the path coefficients.

I will present the results of a simple numerical simulation. I generated 500 independent data sets, each having 500 observations, in which the data follow the correct causal structure (the DAG at the bottom of Figure 7.4). The two path coefficients linking the three latent causes are equal to 0.5. I tested each of these 500 data sets at the 5% significance level using the `sem()` function of lavaan. However, rather than giving this correct causal structure to lavaan, I gave the incorrect DAG (Figure 7.4, middle) in which each effect indicator is treated as if it was the same as the latent theoretical cause. In other words, I tested the incorrect DAG: height of mercury in thermometer → Δ O_2 concentration → g measured on the digital balance. I did this after assigning error variances that are either 1% or 50% of the variance of the latent theoretical construct. The result is shown in Table 7.1.

Table 7.1. Simulation results from 500 independent data sets (each with 500 observations) when generating data follow the true DAG at the bottom of Figure 7.4 but tested in lavaan using the DAG in the middle of Figure 7.4, in which the observed indicator variables are assumed to be the same as the latent theoretical constructs. Each line shows the results when the error variance associated with each observed indicator variable is either 1% or 50% of the variance of the associated latent theoretical construct. Values in bold show where the errors occur.

measurement error (% of latent cause variance)		path coefficient	path coefficient

[186] The consequence is identical if we had used the C-statistic of piecewise SEM.

error variance in observed thermometer reading (Y_1) (σ of e_1)	error variance in observed Δ $[O_2]$ (Y_2) (σ of e_2)	error variance in observed Δ weight (Y_3) (σ of e_3)	rejection rate at α=0.05	$Y_1 \rightarrow Y_2$ (true value is 0.500)	$Y_2 \rightarrow Y3$ (true value is 0.500)
0.01 (1%)	0.01 (1%)	0.01 (1%)	0.056	0.500	0.507
0.50 (50%)	0.01 (1%)	0.01 (1%)	0.052	**0.400**	0.507
0.01 (1%)	0.50 (50%)	0.01 (1%)	**0.178**	0.498	**0.396**
0.01 (1%)	0.01 (1%)	0.50 (50%)	0.056	0.499	0.507

The first line of Table 7.1 shows what happens when the residual variances of the effect indicators (mercury thermometer, Δ $[O_2]$ and Δ weight) are only 1% of the variances of the theoretical constructs (air temperature, metabolic rate and fat loss). This means that the residual (measurement) error of the effect indicators is very small (1%) relative to the total variance of the theoretical constructs. Lavaan correctly rejects the hypothesized DAG in 5.6% of the 500 data sets at the 5% level even though the effect indicators are not *really* identical to their latent causes (since they have a small amount of measurement error). This might seem strange. For instance, the effect indicator "observed thermometer reading" in the correct DAG at the bottom of Figure 7.4 is not d-separated from the effect indicator "observed Δ weight" conditional on the effect indicator "Δ $[O_2]$" as the incorrect DAG requires[187]. Rather, the two are d-separated conditional on the latent variable "metabolic rate". However, because the residual measurement error of "Δ $[O_2]$" is so small relative to the variance of its latent cause ("metabolic rate"), the two are almost numerically identical up to a scaling constant and so, conditioning on (holding constant) "Δ $[O_2]$" is almost the same thing as conditioning on "metabolic rate". This is why, when I assigned a large residual measurement error (0.50 or 50% of the variance of its latent cause) to "Δ $[O_2]$", the model was rejected almost 18% of the time at a significance level of α=0.05. This is a general conclusion: increasing levels of measurement error in effect indicators whose underlying latent causes are conditioning variables in the true DAG will cause you to

[187] This single (incorrect) d-separation claim of conditional independence results in the single constraint in the model-predicted covariance matrix of a zero partial correlation between "thermometer reading" and "Δ weight", conditional on "Δ $[O_2]$".

reject the model more often than you should if you pretend that the effect indicator is equivalent to its latent cause. This phenomenon increases as the sample size of your data set, and therefore your statistical power to detect such errors, increases (Chapter 5).

The other error that arises when you pretend that the effect indicator is equivalent to its latent cause in the presence of measurement error is to bias the estimation of the path coefficients. This arises when the variables on the right-hand side of the structural equation (the direct causes) contain measurement error[188]. This is why, when we incorrectly equated "air temperature" with "thermometer reading" in the presence of substantial measurement error (50%) in "thermometer reading", the path coefficient was lower than its true value (second row of Table 7.1). The same thing happened to the second path coefficient when we incorrectly equated "Δ [CO_2]" with "metabolic rate" in the presence of substantial measurement error in "Δ [CO_2]" (Table 7.1, third row). No bias in the path coefficients occurred when the substantial measurement error was in "Δ weight" since its latent cause ("fat reserved") is not a cause in any of the structural equations in the true DAG. The amount of bias in the estimation of the path coefficients is not affected by sample size, but your ability to statistically detect this bias increases with sample size. This is because the increased statistical power results in narrower standard errors of the estimated path coefficients. In more complicated models, bias in the path coefficients in the presence of measurement error is also more complicated; see Chapter 5 of (Bollen 1989) for a more complete exposition.

Let's go back to the motivating question of this section: when should you use a measurement model and when can you mentally replace the theoretical latent cause with a single effect indicator? After all, it takes a lot of work to properly develop and test a measurement model. Even when you have convinced yourself that the measurement model is valid, you still have to expend the money, time and effort in measuring the different effect indicators. So far, the results of this section suggest that we can avoid using the full measurement model and mentally substitute the latent cause with our best effect indicator, when the residual measurement variance of this effect indicator are "small" relative to the variance of the underlying latent cause. We can pretend that the reading on our mercury thermometer is the same thing as "air temperature" when

[188] Remember that all of the residual variation is supposed to reside in the dependent variable (the causal child) in a regression, not in the independent variables (the causal parents).

the two are highly correlated, i.e. when “air temperature” varies very much more that the variation in the readings of the thermometer when “air temperature” is constant. This minimum correlation must increase, and the residual measurement error must decrease, with increasing sample size because of increased statistical power.

I know what you are thinking: how much measurement error is allowable before I can no longer pretend that my best effect indicator is the same thing as the latent cause? How much measurement error must exist so that I am required to explicitly model it using a measurement model? There is no definitive answer to your question because it depends on so many aspects of your data and the true causal structure that generated it, but you can explore the answer by simulating data as I have done. To get an answer for the true DAG at the bottom of Figure 7.4, I simulated the data for a data size of 500 observations while sequentially increasing the residual variance of “Δ [CO_2]”, since this residual variance affects both the rejection rate and the bias in the second path coefficient (Table 7.1, row 3). The result is shown in Figure 7.5. Given this level of statistical power, you can pretend that “Δ [CO_2]” is the same thing as “metabolic rate” until the residual measurement error in “Δ [CO_2]” that is about 20% of the total variance in the latent “metabolic rate”. A 20% measurement error corresponds to a correlation coefficient between the latent “metabolic rate” and the effect indicator “Δ [CO_2]” of about 0.91 or an R^2 value (which is what lavaan outputs) of about 0.83. If the measurement error is greater that about 20% then you begin to reject your model more often than you should and the bias in the estimated path coefficient associated with “metabolic rate” → “fat loss” becomes evident. Note that this recommendation will change as your sample size increases since you will have increasing power to detect even very small deviations in the measurement model.

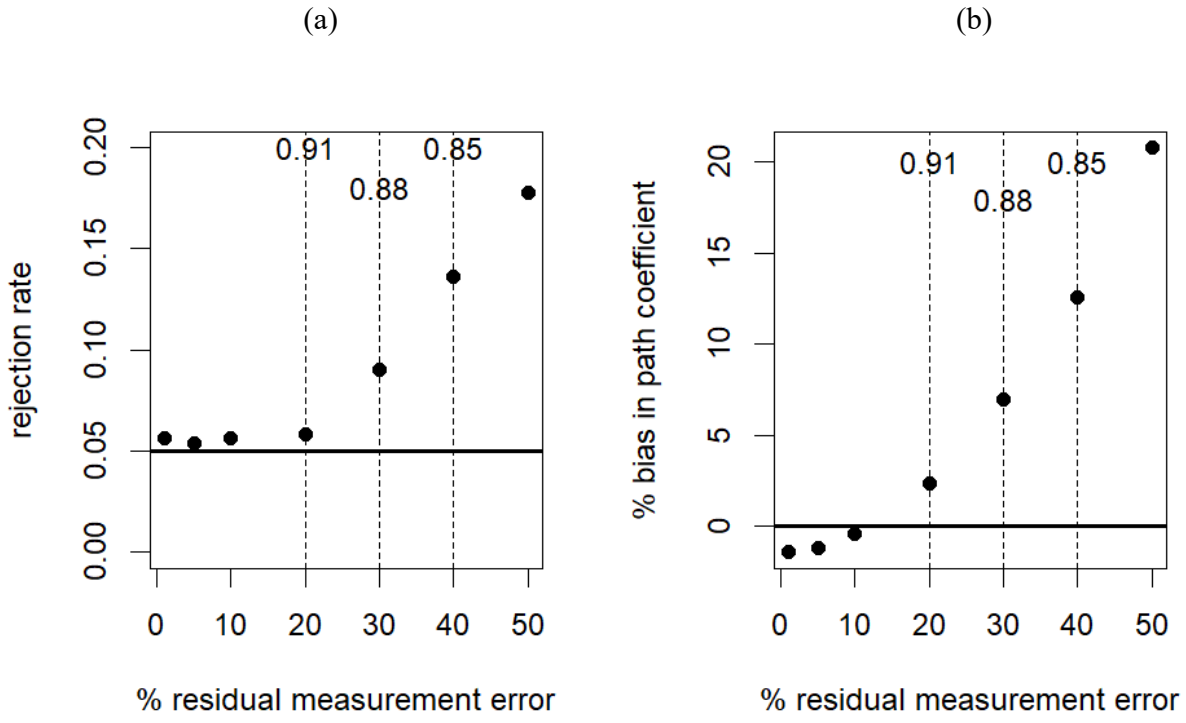


Figure 7.5. (a) The rejection rate of the DAG of Figure 7.4, in which the effect indicators are incorrectly assumed to be equivalent to their underlying latent causes when, in fact, the residual error of "Δ [O_2]" is between 1% and 50% of the variation in the latent cause "metabolic rate". The true rejection rate should be α=0.05 (solid black line). (b) The % downward bias in the estimated path coefficient associated with "metabolic rate" → "fat loss" from the true value of 0.5 as a function of increasing residual measurement error of "Δ [O_2]". The values on the vertical broken lines are the correlation coefficients (R) between the latent "metabolic rate" and the effect indicator "Δ [O_2]".

The measurement error of mercury thermometers, infrared gas analysers and digital balances (the effect indicators) are minor relative to the natural variation in air temperature, metabolic rate and fat loss (the underlying latent causes of the effect indicators) in our imaginary experiment. Furthermore, lots of experimental work has shown that these measurements have been isolated from any other causes besides the theoretical construct for which they are designed. Therefore, our ecophysiologist can safely pretend that his readings on the thermometer, infrared gas analyser and digital balance are the same thing as "air temperature", "metabolic rate" and "fat loss[189]". However, the measurement errors of the effect indicators of "boldness" are too great for

[189] Actually, fat loss would probably have more measurement error, since the change in body weight (the effect indicator) might be also indicating loss of protein, carbohydrates, water etc. These additional unknown causes of body weight would contribute to the measurement error.

the ethologist to simply choose the best one (the flight initiation distance, X_4) and pretend that this is the same thing as "boldness", since the correlation between the latent `bold` and the observed flight initiation distance was $\sqrt{0.46}=0.68$.

7.6 Combining several measurement models in a causal hypothesis and the concept of structural identification

Let's go back to our example of the ethologist who is measuring the "boldness" of individuals and further specify that he is studying Eastern Chipmunks (*Tamias striatus*). Adult males and females maintain their own burrows and territory, come together only during the breeding season, and actively defend their territories. If boldness is "a stable characteristic or disposition that influences the animal's behaviour across a wide range of situations over time, that differs between individuals, and that causes an animal's tendency to take risks or engage in potentially dangerous situations when in uncertain or threatening contexts", then we could hypothesise that bolder female chipmunks will travel longer distances in the landscape and that this will allow them to increase the amount of nuts and seeds stored in their cache. The increased food stores will allow bolder females to successfully produce more offspring during their lifetime. However, bolder females will experience greater predation risks, and this increased predation risk would reduce their average age at death, which would reduce the average number of offspring produced.

Our ethologist cannot accurately measure the number and distance of each trip outside of the nest, but he maintains a grid of traps. He can therefore measure the average number of times the individual has been trapped and the average linear distance from the trap to the nest. Bolder individuals, who travel longer and more often, would be trapped more often and at greater distances from the nest. Our ethologist cannot accurately measure the amount of food that the individual has stored, but he can count the average number of caches per year within the territory of each individual and he can estimate of the volume of the main cache. He can accurately count the number of offspring in the nest for each female. Figure 7.6 shows the DAG for his hypothesis.

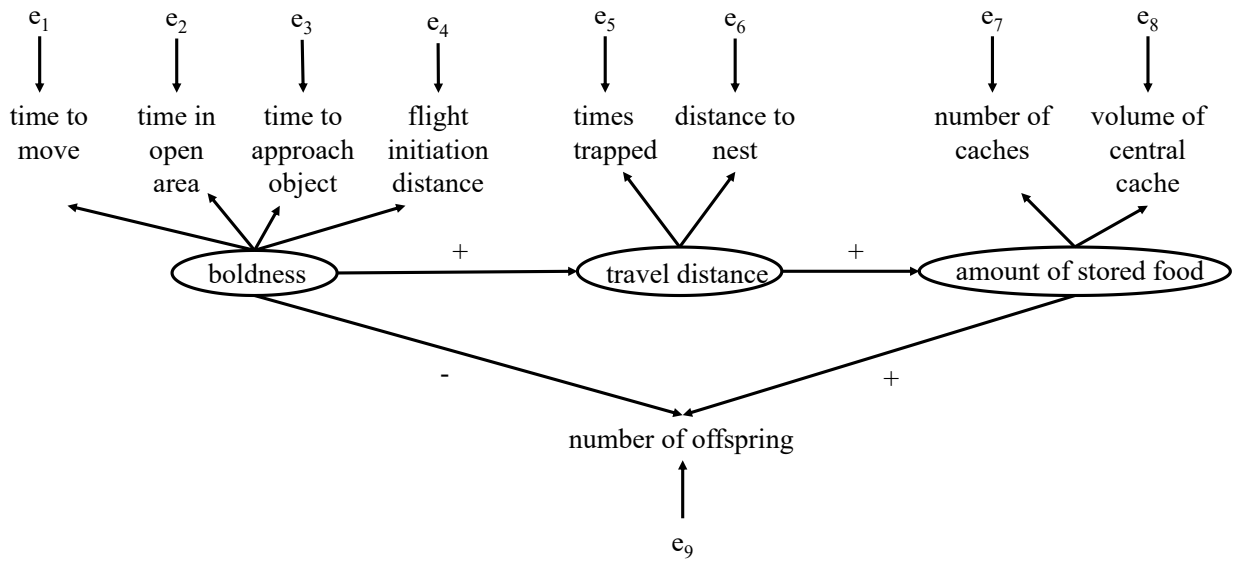


Figure 7.6. A more complicated structural equation model that contains three latent cause variables and nine observed variables, of which eight are effect indicators.

Notice that there are three latent cause variables in Figure 7.6, since they are enclosed by circles. One of these (total travel distance of an individual during her lifetime) is latent but clearly defined and observable in practice. The concept of "distance" presents no theoretical difficulty. In principle, a radio collar could have been placed on each individual and a sufficiently sophisticated radio telemetry system could have measured the total distance travelled over its lifetime. In practice, such a radio telemetry system would be so expensive that it could not be used on more than a few individuals. One of these (total amount of food stored in caches) is latent, slightly less well defined and observable in principle but perhaps not in practice. The food is cached in underground chambers, and one cannot disturb these without affecting the behaviour of the chipmunk. Furthermore, only seeds and nuts that are still edible should be counted. Finally, "boldness" is latent, its theoretical definition is still contested, and it is not observable even in principle. However, this model is still testable using covariance-based SEM, and you know everything needed to test it in lavaan. Because each measurement model has separated out the residual unique variances of the effect indicators from the common variance due to the latent causes, then the estimates of the path coefficients will be unbiased and our ethologist can still estimate the direct and indirect effects of each variable, including the latent variables, on each other variable.

I said that the model in Figure 7.6 is testable. More specifically, I should have said that it is testable because it has positive degrees of freedom and because it is *structurally identified*. This leads to the notion of structural identification. Structural identification, in the context of SEM, refers to the ability to obtain unique estimates for all of the free parameters during likelihood estimation. As a simple example, imagine that I give you the equation X+Y=2 and I tell you that X=1. With this information, you can give me the single value of Y that satisfies this equation (Y=2-1=1) and so this equation is *identified*. If I don't tell you the value for X, then you can't give me the single value of Y that satisfies the equation. Since you don't know the value of X, all you will know is that Y=2-X and there are an infinite number of possible values of Y that can satisfy the equation. The equation is under-identified[190].

A structural equations model that is under-identified cannot be fit or tested and you will be wasting your time if you collect data to test it. If you give such an under-identified model to lavaan then lavaan will usually give a warning such as "`Could not compute standard errors! The information matrix could not be inverted. This may be a symptom that the model is not identified`". This is a rather cryptic message because there are other reasons why the combination of a properly identified model and particular data might prevent lavaan from computing standard errors or inverting the information matrix! Another such cryptic warning due to under-identification is "`the optimizer warns that a solution has NOT been found!`". There are rules for determining if a model is structurally identified, described next.

The following rules are from Kenny et al. (1998), Kenny and Milan (2012), Kline (2016). The first rule is that the measurement scales of all of the latent variables must be fixed, either by fixing the path coefficient from it to one of the effect indicators or by fixing its variance to unity. They apply to a causal graph with a single measurement model or to a causal graph with several measurement models that are correlated together, via free covariances between them (i.e. double-headed arrows between them) or via directed paths between then (i.e. connected by single-

[190] This equation would be *over-identified* if I gave you more than one value for X (say, 1 and 3). All that you could do is choose a value for Y that is "best" in some way; for example, the value that minimizes the error. You might do this by setting X at the mean of 1 and 3 (i.e. 2) and then solving Y=2-2=0. This is what statistical methods like least-squares regression or maximum likelihood estimation do.

headed arrows between them). All four of these rules must hold for a model to be structurally identified.

Rule 2: There must be positive degrees of freedom. Actually, the requirement for structural identification is that there cannot be negative degrees of freedom. A model with zero degrees of freedom is structurally identified, but you cannot test it against empirical data, so you can never know if it has any empirical validity. You already know how to calculate the model degrees of freedom in covariance-based SEM, namely $df = V/(V+1) - P_{free}$, where P_{free} is the number of parameters that must be estimated (thus "free"). Normally, $P_{free}=(p+q)$ where p is the number of free path coefficients in the model and q is the number of free variances and covariances of exogenous variables (including the residual error variances) in the model. Of course, if you fix a path coefficient, a variance or a covariance to some numerical value, as explained in Chapter 4, then this parameter is no longer free. I say that P_{free} is "normally" equal to (p+q) because it is also possible to place equality constraints on two or more free parameters, as explained in Chapter 4, and this will decrease the number of free parameters. For instance, if I have two free parameters (a_{ij} and a_{ik}) and I force these two free parameters to be equal during likelihood maximization, then there is only one, not two, values that can be assigned to them and so there is really only one "free" parameter to be estimated.

Rule 3: for each cause latent variable, at least one of the following conditions must hold

(i) There are at least three effect indicators whose errors are uncorrelated with each other (i.e. no double-headed arrows between them), or

(ii) There are at least two effect indicators whose errors are uncorrelated with each other and either

   a) The errors of both effect indicators are not correlated with the error term of an effect indicator of another latent variable; or

   b) An equality constraint is imposed on the path coefficients (loadings) of the two indicators.

Rule 4: For every pair of cause latents, there are at least two effect indicators, one from each cause latent, whose error terms are uncorrelated.

Rule 5: For every effect indicator, there is at least one other effect indicator (not necessarily of the same latent cause), with which its error term is not correlated.

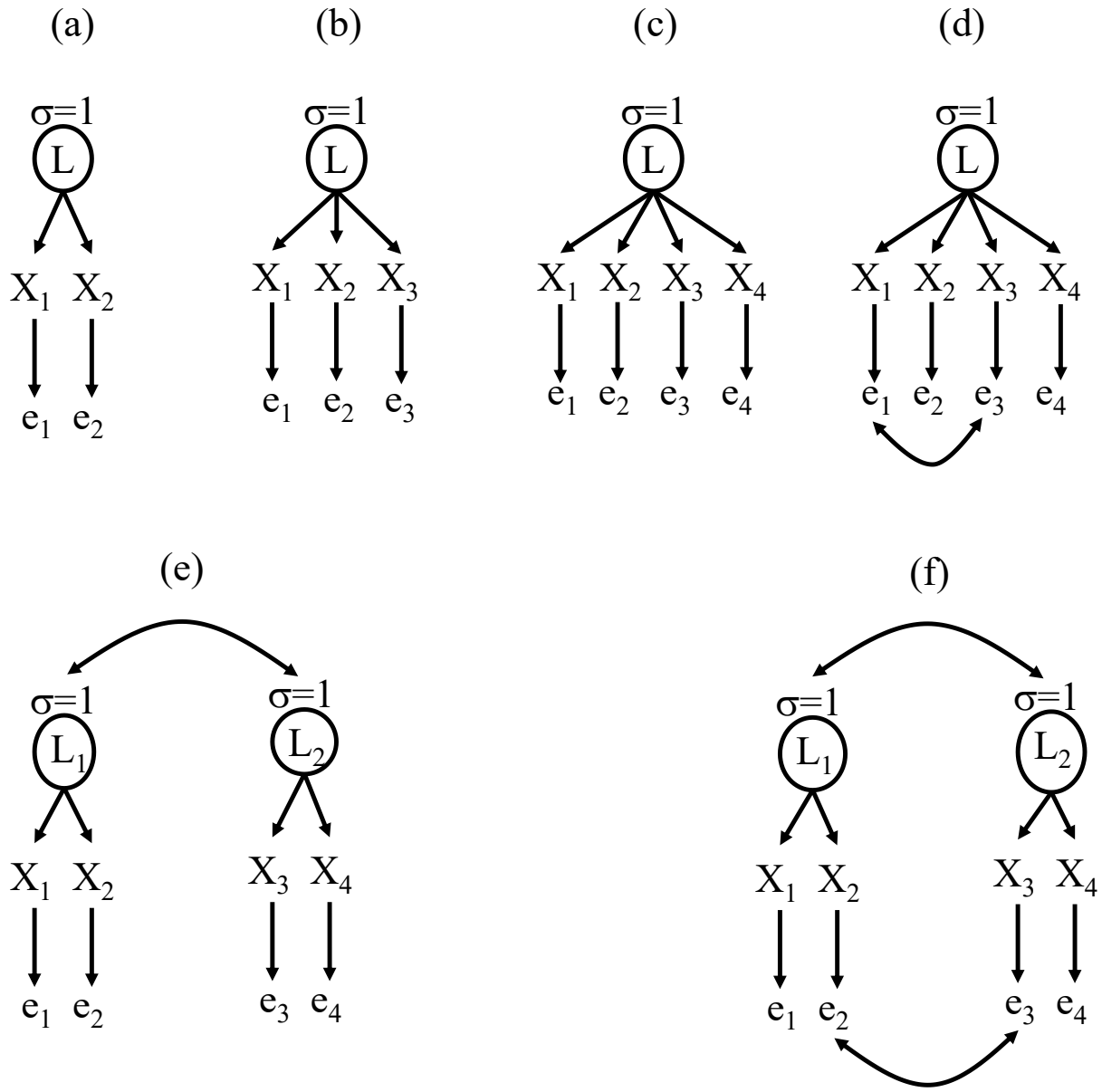


Figure 7.7. Six causal graphs involving causal latents (inside circles) and their effect indicators. The measurement scale of each latent is determined by fixing its variance to unity.

To see how to apply these rules, look at Figure 7.7. Is the causal graph (a) structurally identified? No. There are only V=2 observed variables (X_1, X_2) but there are P_{free}=4 free parameters: two path coefficients (p) and two residual error variances (q). Therefore, there are

2(3)/2 – 4 = -1 degrees of freedom, violating the second rule. Is the causal graph (b) structurally identified? Yes, but there are no degrees of freedom to test it. There are V=3 observed variables (X_1, X_2, X_3) and there are P_{free}=6 free parameters: three path coefficients (p) and three residual error variances (q). Therefore, there are 3(4)/2 – 6 = 0 degrees of freedom. Is the causal graph (c) structurally identified? Yes. There are V=4 observed variables. There are four path coefficients (p) and four residual variances (q) and so there are P_{free}=8 free parameters. Therefore, there are 4(5)/2 – 8 = 2 degrees of freedom. Is the causal graph (d) structurally identified? Yes. It is the same as graph (c) except for an added free covariance between the residual errors e_3 and e_4, which is an additional free parameter, and so there are 4(5)/2 – 9 = 1 degree of freedom. How about the causal graph (e)? Since V=4, p=4 and q=5, there is still 4(5)/2 – 9 = 1 degree of freedom. Furthermore, rules 3, 4 and 5 hold. Is the causal graph (f) structurally identified? No. It is the same as the causal graph (e) except that there is a free covariance between e_2 and e_3. Since X_2 is an effect indicator for the first latent (L_1) and X_3 is an effect indicator for the other latent variable (L_2), this violates rule 3 since this requires that the errors of both effect indicators are not correlated with the error term of an effect indicator of another latent variable. Finally, go back to the causal graph in Figure 7.6 and verify that it doesn't violate any of the rules.

These rules apply to the measurement models that are embedded in the overall structural equation model, but they don't cover all possible cases. For instance, you might only have a single effect indicator for a latent variable, but you are confident that this effect indicator is a good proxy for its latent cause, and you have some external information about its reliability and therefore of the proportion of its total variance that is due to its unique (error) variance. I will give an empirical example later in this chapter. In this case, you can still structurally identify the model by fixing the error variance of this single effect indicator to the known (or estimated) value. Even if you do not know the proportion of its total variance of this single effect indicator that is due to its unique (error) variance, you run the analysis using different reasonable guesses and compare the results. If, for example, the model is not rejected until the assumed error variance is (say) 30% of total variance of this single effect indicator, then you can assess if this level or measurement error is reasonable. Remember what this measurement error represents: it is the amount of variation in this effect indicator that would occur if you held constant its single latent cause. Figure 7.6 has an observed variable "number of offspring produced by a female

over her lifetime". We might want to replace this observed variable by a latent "true number of offspring" that is measured by the single observed variable "counted number of offspring". After all, it is possible for a baby to be born and die before the researcher visits the nest or for the researcher to make a mistake when the young chipmunks are being counted. What is a likely probability of these errors occurring? Let's say that the researcher believes that there is a 5% chance of a baby being born and die between surveys. Then a reasonable estimate of the error variance would be 5% of the total variance in the latent variable "true number of offspring" produced. After telling lavaan to estimate the latent variance and giving it a name (Chapter 4), you would then tell lavaan to fix the residual variance of the observed "counted number of offspring" to 5% of the total variance of the latent. Furthermore, if it is reasonable to assume that an increase of 1 offspring produced would result, on average, with an increase of 1 offspring actually counted, then the path coefficient from the latent "true number of offspring" would be fixed to 1.0.

Before leaving the subject of model identification, there is one more complication that can cause problems. The above rules apply to "structural" identification; that is, identification based on the structure of the causal graph. However, this doesn't insure "empirical" identification. To understand the difference, consider the causal graph in Figure 7.7(e). It was structurally identified according to our rules but remember that when there is more than one latent cause, the above rules assume that these latent causes are correlated – either because there is a free covariance between them or because there is a directed path between them. What would happen if, as an empirical fact, the two latent variables were uncorrelated? In that case, we would really have two independent measurement models (one for each latent). Each separate measurement model only has two effect indicators, resulting in "empirical" under-identification.

Because these rules can become complicated, one way of determining if a proposed model is structurally identified before you begin expending time, effort and resources to collecting data is to generate simulated data from the causal graph as I have done several times already in this book, and then "test" these data against the proposed model in lavaan. If lavaan can't properly fit the data to the model, then the model is under-identified, and you would be wasting your time collecting data to test it.

7.7 Composite latent variables

So far, I have modelled latent variables using a measurement model, in which the unobserved latent variable is the common *cause* of one or more observed effect indicators; other names for such latents are "reflective" or "cause" latents. However, a measurement model is only one way of including explicit latents into a covariance-based SEM. Another way of including an explicit latent is via a "composite" latent. A composite latent is a latent variable that is a common *effect* of a set of observed causes rather than the common *cause* of a set of observed effects (indicator variables) (Grace and Bollen 2008). Because of this, another name for a composite latent is an "effect" latent. Therefore, the causal claims of a composite latent variable are exactly opposite of the causal claims of a measurement model.

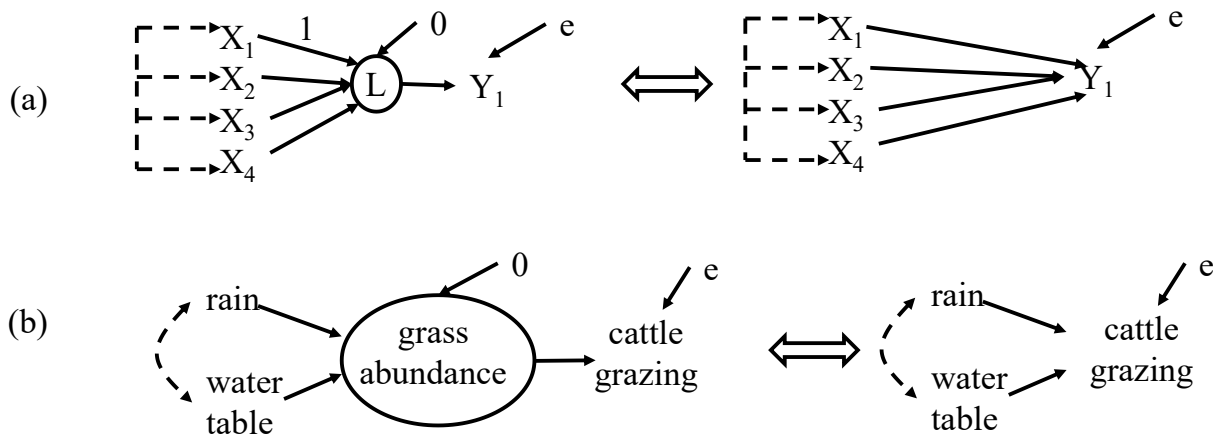


Figure 7.8. Examples of composite latent variables. (a) Four observed variables (X_1 to X_4) that combine to cause the latent variable (L); in this case, the four observed variables completely determine L but this is not necessarily the case. Removing the latent (L) results in the m-separation equivalent DAG, which is a multiple regression. (b) Two correlated observed variables that combine to determine the latent variable (grass abundance), which then causes an observed variable (cattle grazing). Removing the latent (grass abundance) results in the m-separation equivalent DAG, which is a multiple regression.

I begin with the simplest composite latent model, shown in Figure 7.8a (left). The composite latent is variable "L" because it is latent (indicated by putting it inside a circle) and it is composite because it is a common effect of (i.e. composed of) the observed variables (X_i). I have added broken double-headed arrows to signify that the observed causes of the latent composite variable can have any patterns of dependency between them, from a case in which the

observed causes are mutually independent to the case in which they are all correlated in some way. In Chapter 5 I explained how to modify a DAG by removing an explicit latent variable while maintaining the same causal claims among the remaining variables. If you apply these rules to the causal graph in Figure 7.7a (left) then you will notice that the causal graph at the right in Figure 7.7a is equivalent to the one on the left. However, the causal graph on the right is simply a multiple regression of the observed causes (X_i) on Y_1! This is a general property of composite latents with a single effect indicator. They are simply an alternative way of representing the joint direct effects of several observed causes on an observed effect that places the emphasis on their joint effects on the observed effect (i.e. the composite latent) rather than on the unique effects of each observed cause on the observed effect.

The scale of the composite latent must be defined by fixing the path coefficient from one of its observed causes, similar to what you learned with the measurement model. This value can be whatever you like but, since you are really modelling a linear regression, I suggest that you simply run the regression (`lm(Y1~X1+X2+X3+X4)` for Figure 7.8a) and then fix the path coefficient of one of the observed causes to the value obtained in the multiple regression. Alternatively, you can fix the path coefficient of one of the observed causes to 1.0. Either way, the indirect effect of each observed cause (X_1) on the observed effect (Y_1) is (as always) the product of the path coefficients along that directed path. As an aside, models that include composite latents can sometimes run into convergence problems. If this happens, then using the partial slopes output by the multiple regression as starting values will often solve the problem. Finally, the variance of the composite latent must be set to zero[191] because there is only one effect indicator of this latent; lavaan does this by default. Just as lavaan has a special operator for modelling an effect latent in a measurement model ("~="), it also has a special operator to model a composite latent ("<~"). The variable on the left of this operator is assumed by lavaan to be a composite latent and can have any valid name as long as the name doesn't already exist in the input data. Here is how to specify the model object for the causal graph in Figure 7.8a:

```
mod<-"L<~0.272*X1+X2+X3+X4
Y1~L
```

[191] If the causal graph contains several observed effect indicators for the composite latent then you can estimate the residual variance of the composite because, in effect, this composite latent is also part of a measurement model.

`L~~0*L"`

I fixed the path coefficient for X_1 to 0.272 because this was the partial slope of X_1 that was obtained when I did a multiple regression (`lm(Y1~X1+X2+X3+X4)`), but I could have fixed it to 1.0. The advantage of fixing it to 0.272 is that the measurement scale of Y_1 relative to its causes will remain the same as in the multiple regression. By default, lavaan fixes the residual variance of the composite latent to zero and so the last (`L~~0*L`) is not needed. In practice, you would specify the causal relationships between the observed causes (X_1 to X_4) within the model object or else explicitly model the free covariances between them, setting pairs that should be independent to zero.

What is the causal interpretation of the model in Figure 7.8a (left)? Because we must fix the residual variance of the composite latent to zero in order to identify the model, we are saying that this composite latent is completely caused by these direct causes. The composite latent is *defined as* the combined effects of these direct causes and nothing else. Because of this, the composite latent is simply a statistical device that allows us to focus to the combined effects of these direct causes on Y_1 (Figure 7.8a, left) rather than focusing on the separate effects of each direct cause on Y_1 (Figure 7.8a, right). For example, imagine that the survival of some insect species (Y_1) is affected by the amount of litter (i.e. dead vegetation) on the forest floor (litter→survival). This dead vegetation consists of newly fallen dead tree leaves (X_1), newly fallen dead branches (X_2), newly produced dead understory herbs (X_3) and the amount of dead organic matter from previous years (X_4). You could model this as in Figure 7.8a (right), in which case the emphasis would be on the unique effects of each component (X_i) on survival. Alternatively, you could introduce the latent composite variable ("litter"), as in Figure 7.8a (left), in which case the path coefficient from "litter" (L) to survival (Y_1) would quantify the combined effect of each of these four components of "litter" on survival.

Because a composite latent is defined as the combined effects of its direct causes, then it can also combine different, but related, variables. For instance, perhaps you have the following observed variables: average weekly rainfall amount ("rain"), average water table depth ("water table") and grazing rate of cows ("grazing"). You could model this as shown if Figure 7.8b (left). However, it is not immediately obvious how either rainfall or the water table could directly affect how much grass cows would eat. It is much more likely that the two variables relating to water

availability indirectly cause cow grazing rate by affecting the abundance of grass, but you haven't measured the grass abundance. You could therefore introduce a composite latent (Figure 7.8b centre) which is statistically identical to Figure 7.8 (left). What does this composite latent represent? What should you call it? Strictly speaking, it is simply the weighted sum of average weekly rain fall and average weekly water table depth. If these two variables are the major determinants of soil water availability, then you could interpret the composite latent as "water availability". If water availability is the major determinant of grass abundance, then you could equally interpret the composite latent as "grass abundance". Either of these interpretations are possible if you can provide external evidence[192] to support them but neither interpretation can be justified solely from the model fit since they are equivalent (Chapter 4). However, if water availability is not the major determinant of grass abundance (perhaps soil properties or temperature are equally important) then you should not call the composite latent "grass availability". This is because you must set the error variance of the composite latent to zero, i.e. that there are no other causes of "grass availability" except for rainfall and the water table depth. In such a case, ff you did want to identify the latent as "grass availability", then you would need some observed indicator variables of this latent variable.

[192] Of course, you would have to provide this external evidence and justify your interpretation to a skeptical scientific audience.

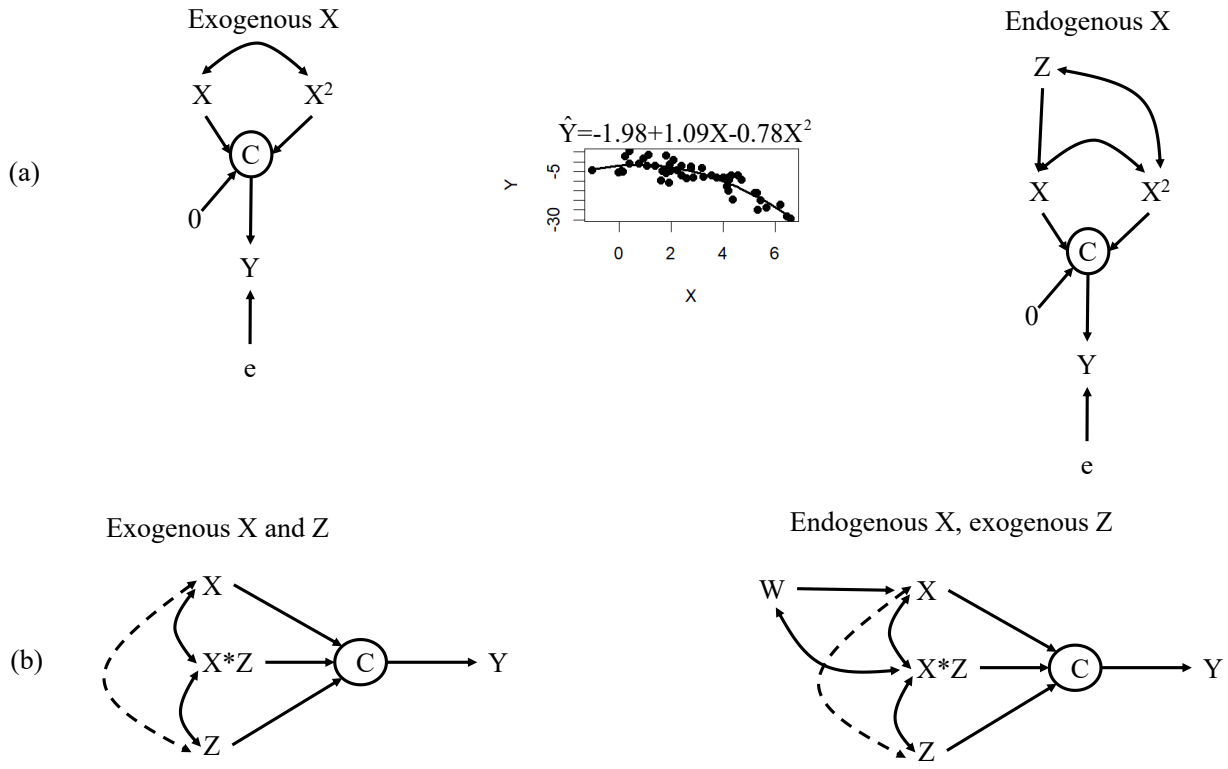


Figure 7.9. Representing nonlinear (polynomial) relationships (a) and interactions between variables as composite latents (b) when dealing with exogenous or endogenous variables.

Another example of when a composite latent improves interpretation is when you want to include nonlinear effects or interactions between variables (Grace and Bollen 2008). Since covariance-based SEM only allows linear relationships, the only way to model nonlinear relationships or interaction terms is to use the same statistical trick as done with multiple regression. If you want to model a quadratic relationship between X and Y, you would centre X about its mean[193], include a "new" variable in the data set that is the square of the centred X (`X2<-(X-mean(X))^2`), and then regress Y on both X and X^2: `lm(Y~X+X2)` (Figure 7.9a); the free covariance between X and X^2 is mandatory in this case because X^2 will always be associated with X. You can certainly model this in SEM without introducing a composite latent but, if you want to emphasise the overall nonlinear relationship rather than the separate contributions of X and X^2, then you can combine the two in a composite latent. If X is endogenous, then you must also allow a free covariance between X^2 and the direct cause of X

[193] This is done to reduce the autocorrelation between the polynomial terms.

since, if this variable directly causes X then it will necessarily be correlated with X^2 as well. If you want to regress Y on two predictor variables (X_1, X_2) plus their interaction then you would create a "new" variable that is the product of the centred values of X_1 and X_2 (Figure 7.9b); the free covariances between $X_1 \bullet X_2$ and each of X_1 and X_2 are again mandatory and the broken double-headed arrow means that X_1 and X_2 can (or might not) have some dependency that would have to be modelled.

7.8 Composite latent? Measurement model? How to decide

It is important not to get hung up on the classification of latent variables. The distinction between a cause latent (leading to a measurement model) and an effect latent (leading to a composite latent) is somewhat arbitrary because a latent can have one or more variables (observed or latent) causing it and also have one of more variables (observed or latent) that it causes. When choosing how to explicitly model latent variables, besides the necessity of model identification, the key is to conduct the same thought experiments on your causal graph as described in Chapter 2, irrespective of whether you can conduct these manipulations in practice. You must conduct a thought experiment based on your causal hypothesis and ask two questions.

1. If I could manipulate the latent variable L and then observe variable X, *while preventing all other variables in the causal graph (observed or latent) from responding*, would I expect variable X to respond?
2. If I could hypothetically manipulate observed variable X and then observe latent variable L, *while preventing all other variables in the causal graph (observed or latent) from responding*, would I expect latent variable L to respond?

If you answer "yes" to question 1 then your causal graph must include L→X. If you answer "yes" to several observed variables then you would include a measurement model in your causal graph with this causal latent directly causing a series of effect indicators. If you answer "yes" to question 2, then your causal graph must include X→L. If you answer "yes" to several observed variables then you would include a composite latent in your causal graph with a series of effect

indicators all causing the common cause latent L. If you answer "yes" to both questions for different sets[194] of observed variables, then your latent is both a cause and an effect latent.

I will illustrate such a thought experiment using the paper by Verheyen et al. (2003). My purpose is not to present the actual analysis by these authors, or the analysis by Grace and Bollen (2008). Instead, I want to go through the process of conducting such a thought experiment in order to best define the latent variables and decide how to model them, given my understanding of the theoretical and empirical arguments presented in Verheyen et al. (2003).

These authors studied the factors determining the success of different species of forest understory herbs to disperse into, and then grow and reproduce, in forest stands that had formed via secondary succession within a 360-ha matrix of grassland and arable land in Belgium. These understory species had presumably colonized these regenerating forest stands from propagules arriving from the remaining old-growth forested areas and then expanded (or not) their populations depending on their ability to grow and compete within a given forest stand. The forest stands differed in landscape properties like their age since regeneration (1 – 224 years) and their distance to the nearest source population (defined as an old-growth forest patch). These two variables ("age", "distance") were considered by the authors to affect the rate at which propagules of each understory species could immigrate into a patch from the old-growth forest. The forest stands also differed in internal environmental properties related to the soil and the abundance of potential competitors. The authors measured three soil properties in each forest patch: soil moisture, soil texture and soil pH. Soil pH was measured after placing soil samples in a 1:5 suspension of water to 1N KCl. Soil moisture was based on an ordinal scale (1=relatively well drained, 2=imperfectly drained, 3=poorly drained, 4=very poorly drained), as was soil texture (1=loamy sand, 2=loam, 3=clay). The authors also measured two biotic properties in each forest patch related to the abundance of potential competitors of the target understory herb species: the basal area of shrubs and the % cover of tall herbs of other species within a 5 m radius of each sampling point.

The first step is to carefully define and name the theoretical constructs involved in the causal hypothesis. One theoretical construct is the population size of the target understory herb

[194] The same observed variable cannot be both a direct cause and a direct effect of the latent variable since this would create a cycle in the causal graph.

species[195]. Ideally, one would define what "size" means (number of individuals, total biomass…) but, for lack of information, I will assume that this is the total biomass of this species in a given forest patch at the time of sampling. A second theoretical construct is the summed population sizes of potential competitors of the target understory species within a forest stand, whose effect is to supress the growth or reproduction (thus reduce the population size) of the target species. The third theoretical construct is slightly ambiguous. I have called it "propagule immigration" based on my understanding of the text in Verheyen et al. (2003). This is because they link the age of a forest patch and its distance from the old-growth forest to dispersal of propagules. However, Verheyen et al. (2003) called it "land use" and Grace and Bollen (2008) called it "landscape properties". The final theoretical construct is even more ambiguous and is related to the three soil properties that they measured. However, it is clear in the text that this theoretical construct is related to the ability of different soils to promote plant growth and so I am calling it "soil properties affecting growth".

The next step is to include the directed causal relationships between the theoretical constructs, as dictated by the causal hypothesis. According to the hypothesis proposed in Verheyen et al. (2003), increased immigration of propagules will increase the target population size even if we held constant all of the other variables and will also increase the population sizes of the potential competitors even if we held constant all of the other variables. Therefore, the causal graph includes propagule immigration→target population size and immigration→total population of competitors. Increasing population sizes of competitors will decrease the population size of the target species even if we held constant all of the other variables, thus we include the arrow total population of competitors→target population size. Soil properties promoting plant growth will increase both the population sizes of the target population and of the other potential competitors even if we held constant all of the other variables, thus we include soil properties affecting growth→target population size and soil properties affecting growth→total population of competitors. Figure 7.10a shows our causal graph so far.

[195] The authors called this latent "colonization", but the context makes it clear that they are referring to the success of colonization, i.e. the population size of the target species.

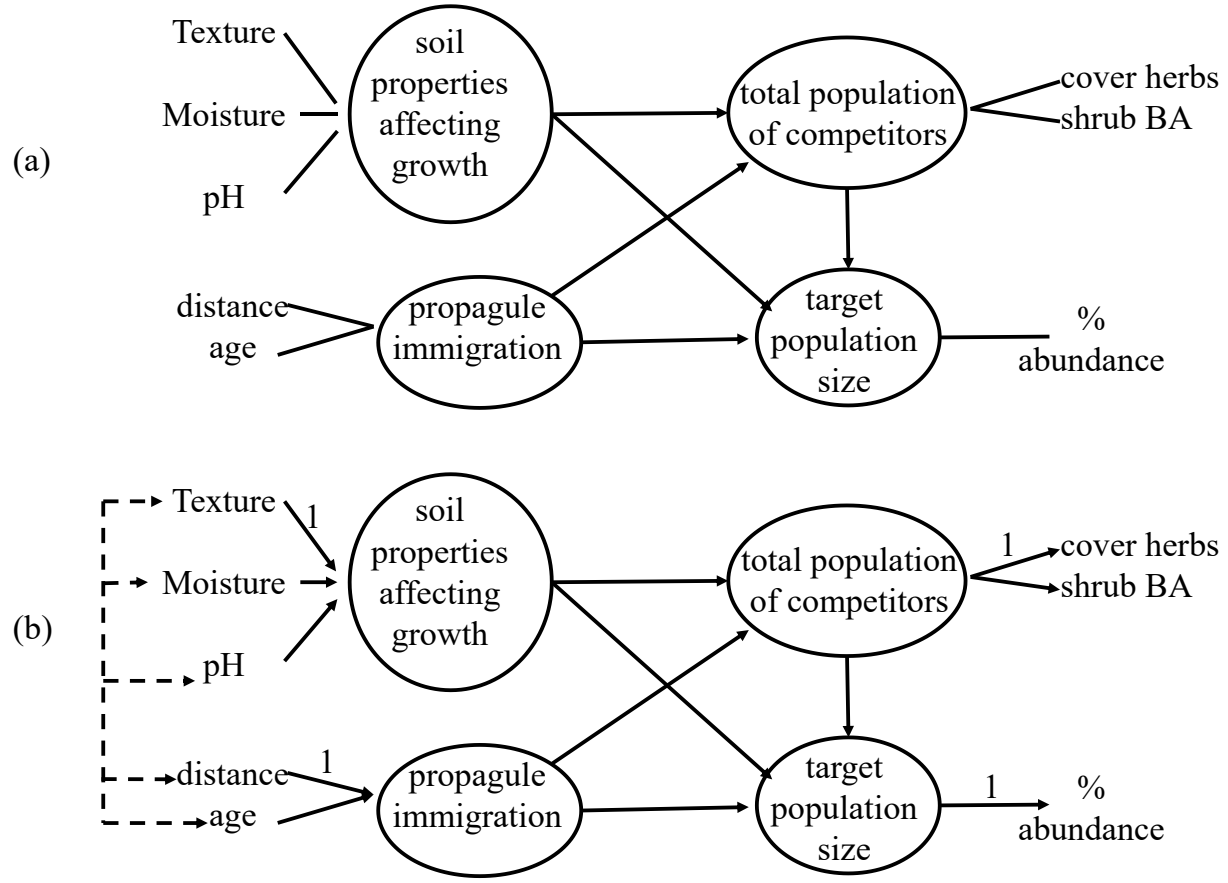


Figure 7.10. The causal relationships between the theoretical constructs (latent variables), but not between the latents and the observed variables, are shown in (a). The relationships between the latents and the observed variables are shown in (b).

Since I have not yet decided how to link these latent variables to the observed variables, I have not yet oriented the lines joining the observed variables to the latent variables in Figure 7.10a. Should the arrows go from the latent to the observed variable (i.e. a causal latent) or from the observed variable to the latent (i.e. an effect latent)? Answering this question will define whether we model these latents using a measurement model (a cause latent) or a composite latent (an effect latent). If I could increase the true population size of the target population then I would expect that the proportion of the subsamples in a forest patch that had the target species present ("% abundance") at the moment of sampling to increase. However, if I redid our sampling and saw a change in % of the subsamples in a forest patch that had the target species present, I wouldn't expect this to cause an increase in the true population size of the target species. Therefore, I must orient our arrow as target population size→% abundance. A similar argument

leads me to add arrows from “total population of competitors” to each of “cover of herbs” and “shrub BA” (shrub basal area).

What about the latent “propagule immigration”? Clearly, adding propagules to a forest patch would not cause either the distance to an old-growth forest (“distance”) or the age of the forest patch (“age”) to change. On the other hand, according to the ecological theory being tested, changing either of these two observed variables would change the number of propagules immigrating into the patch even if I could keep constant all of the other variables in the causal graph. Therefore, I must orient the arrows from each of these observed variables to the latent “propagule immigration”.

How should I model the latent “soil properties affecting plant growth”? Are texture, moisture and pH of the soil imperfect indicators of this soil property or are they causes of this soil property? If I choose the former, then I am claiming that a change in this latent soil property will simultaneously change soil texture, soil moisture and soil pH. However, I know that it is possible to change soil pH (by adding lime) without changing soil texture (the proportions of sand, silt and clay) or soil moisture content. The same thought experiment leads to the same conclusion for each of the other two observed soil variables. This means that the arrows must go from each of the observed soil variables to the latent “soil properties affecting growth”. The result is shown in Figure 7.10b. To complete the causal graph, I must also decide how to model the exogenous observed variables. Nothing in the biological hypothesis specifies the causal relationships between them and so I have allowed free covariances between them. Finally, to ensure structural identification, I could define the scale of the four latents by fixing one of the path coefficients to each of the latents to 1.0, as shown in Figure 7.10b.

7.9 Empirical example: Measuring “soil fertility”

Now, let’s go through all of the steps of defining, testing, and revising a measurement model using the empirical example of measuring “soil fertility” at the level of plant communities. The key references are Daou and Shipley (2019, 2020), Lamontagne and Shipley (2022). The first step is to define the meaning of this concept as clearly as possible in relation to existing theory.

Some of my students view this first step as too "philosophical" for empirically minded biologists. It is not. It is essential to clearly establish in your own mind (and in the mind of critical readers) exactly what you are trying to measure before you can properly choose good candidate observed effect variables in your measurement model.

A first attempt at defining the meaning of "soil fertility" might be "*the propensity of a given soil to promote plant growth*". Note that this this formulation defines soil fertility with reference to the hypothesized effect of the soil on a biological response (soil fertility → plant growth) and is silent with respect to the physical properties of a soil that cause soil fertility. This is important because a measurement model requires the observed variables to be effects, not causes, of the latent variable. Of course, you can, and should, try to identify the causes of your latent variable but only after you have convinced yourself (and critical readers of your model) that you have correctly measured the latent variable! I will give an example later. Since plant growth is also affected by environmental properties that are not properties of soils, we must modify the definition to "*the propensity of a given soil to promote plant growth when all non-soil related environmental properties affecting plant growth are held constant*". In other words, we require that our observed response (plant growth) not be affected by environmental variables that are not properties of the soil (temperature, water availability, light intensity and duration, CO_2 concentration in the air, the presence of other plants). This is analogous to requiring that a column of mercury in a thermometer be in a closed glass tube in a partial vacuum so that it responds only to changes in air temperature and not to changes in air pressure. Since soil biologists generally include soil microbes (bacterial and fungal) as a property of a soil, I will count this as a soil property, but this is at least open to debate. If you don't want to count soil microbes as a property of soils, then you would have to hold constant[196] the abundance of soil microbes as well. However, our definition is still too ambiguous because we have not clearly defined what we mean by "plant growth". Which plants? Since the goal in the above-cited papers is to produce a measure of soil fertility that is applicable at the level of entire plant communities, we must not limit our definition to any particular species or subset of species. Here is a third attempt at defining soil fertility at the level of plant communities: "*soil fertility is*

[196] For instance, by killing all microbes and fungi so that their abundance is zero across all soils in your measurements.

the propensity, generalizable across plant species, of a given soil to promote plant growth when all non-soil related environmental properties affecting plant growth are held constant".

This definition of soil fertility makes a strong biological claim. Imagine that we grew a single plant species in different soils while holding constant all non-soil related environmental properties affecting plant growth. We then rank these soils based on the growth rate of this single species. Now, imagine that we did it for many other plant species. Assume that we could do this with such large sample sizes that we would not have to deal with random sampling fluctuations in our measurements of plant growth. Our definition of soil fertility requires that the ranking of soils would remain the same irrespective of the plant species. In other words, if soil B produces a higher growth rate than soil A for one species then it does this for all species. If this were not the case, then the propensity of a soil to promote plant growth (when all non-soil related environmental properties affecting plant growth are held constant) would differ between species; the "fertility" of a soil would depend on which plant species was being queried and so would not apply at the level of plant communities. If this occurred, then this propensity of soils would not apply to "any plant species". In the jargon of confirmatory factor analysis, we are claiming that "soil fertility" is a unidimensional factor.

Before beginning the empirical analysis, there is another technical detail to specify. Plant growth is the rate of biomass[197] accumulation per unit time but this rate is not linear over time. Plants initially grow at an exponential rate and then slow down – even when resource supply rates are not limiting – when they begin to allocate biomass to reproduction. Therefore, plant biologists measure the "relative growth rate" as the rate of the log-transformed biomass accumulation per unit time, which is measured as the slope of a regression of ln(biomass)~time giving units of the mass of new biomass produced per mass of existing mass per unit time. Often, this is expressed in units[198] of mg of new biomass produced per g of existing biomass per unit time. Given these details, and the definition of "soil fertility" that we have developed, we can now develop a method of measuring "soil fertility". First, collect a wide range of different soils and store these soils in such a way that they are undisturbed and biologically quiescent until the experiment

[197] This is usually specified as plant dry mass.

[198] $\frac{d\ln(biomass)}{dt} = \frac{d(biomass)}{biomass \cdot dt} = \frac{g}{g \cdot d}$

begins. Collect soils that span a wide range of soil properties that might affect plant growth. Second, choose a set of plant species that are biologically diverse with respect to their responses to these soils. We do this so that, if our biological hypothesis is wrong (i.e. if the propensity of a soil to promote plant growth when all non-soil related environmental properties affecting plant growth are held constant differs between species), then we can reject our measurement model. Third, grow each species in each of the different soils in the absence of herbivores or pathogens that are not considered part of the "soil". Fourth, ensure that interactions between individual plants (a non-soil property that can affect growth) will not occur. Finally, hold constant all other environmental properties that can affect plant growth (temperature, water availability, relative humidity of the air, light intensity and duration, CO_2 concentration).

Laurent Daou (Daou and Shipley 2019) conducted such an experiment whose results are in the data frame `Daou2019`. He collected intact 8X8X9 cm soil cubes from 76 uncultivated[199] grassland sites in southern Quebec (Canada), removed all above-ground plant mass without disturbing the soil, allowed the soils to dry and then stored the soil cubes in the dark at 4°C. He then chose four indicator species. Red Clover (*Trifolium pretense* L.) can access atmospheric nitrogen in nitrogen deficient soils via its symbiotic association with *Rhizobium* bacteria. Most plant species have mycorrhizal fungi that provide access to phosphate in phosphate deficient soils, but Thale Cress (*Arabidopsis thaliana* (L.) Heynh.) does not have such mycorrhizal fungi. Red Fescue (*Festuca rubra* L.) is a grass that often dominates low productivity grasslands but is largely absent from high productivity grasslands. Hard Red Wheat (*Triticum aestivum* L.) has been artificially selected to grow in highly productive soils. These four species were chosen because, based on their different biological responses, he expected that they would display different responses to soils if our biological hypothesis was wrong, i.e. if soil fertility is a propensity that is not generalizable across plant species. Seeds of each species were planted in four replicate pots per soil containing each of the 76 different soils in identical environmental conditions of light intensity and duration, air temperature, atmospheric CO_2 concentration and amounts of added water to the soil. Once the seeds germinated, excess seedlings were removed so that a single pot only contained four widely spaced plants. The eight seedlings of each species in two pots were harvested at each of two harvest dates for each soil before any reproductive

[199] All were originally forested sites within the last three hundred years, and some may have previously been cultivated fields, but the history of land use is unknown.

tissues developed, the aboveground dry weight of each plant was measured, and the average relative growth rate (RGR) of each species in each soil was measured[200].

The scale of the latent "soil fertility" was determined by fixing the path coefficient from it to the RGR of wheat. In other words, a 1-unit increase in "soil fertility" is one that will increase the relative growth rate of wheat by 1 mg/g/d. By default, lavaan centres each variable, meaning that a soil fertility of zero would be a soil producing the average RGR of wheat. I want a measurement scale of soil fertility whose zero value is a soil in which wheat has a relative growth rate of zero. I get this by telling lavaan to also include the intercepts (`meanstructure=T`) and then adding the intercept for wheat to the predicted values of the "soil fertility" latent variable. In order to get the strength of the correlations between the observed RGR values of each species and the latent "soil fertility", I include the `rsquare=T` argument in the `summary()` call. In order to get an overall estimate of composite reliability, I use the `compRelSEM` function of the semTools library. The measurement model for soil fertility is shown in Figure 7.11a. Here is the R code:

```
Daou.mod1<-"
soil.fertility=~1*RGR.Triticum+RGR.Festuca+RGR.Trifolium+RGR.Ara
bidopsis
"
fit.Daou1<-
sem(Daou.mod1,data=Daou2019,meanstructure=T,estimator="MLM")
summary(fit.Daou1,rsquare=T)
compRelSEM(fit.Daou1)
```

[200] RGR was measured using a separate mixed-model regression for each species. The model was ln(biomass, mg) ~ age (fixed effect, days) + soil (random effect). RGR in each soil was the slope of this regression estimated for each of the 76 soils. These mixed-model regressions were based on 923, 1198, 1191 and 1195 individual plants of *F. rubra*, *T. pretense*, *T. aestivum* and *A. thaliana*.

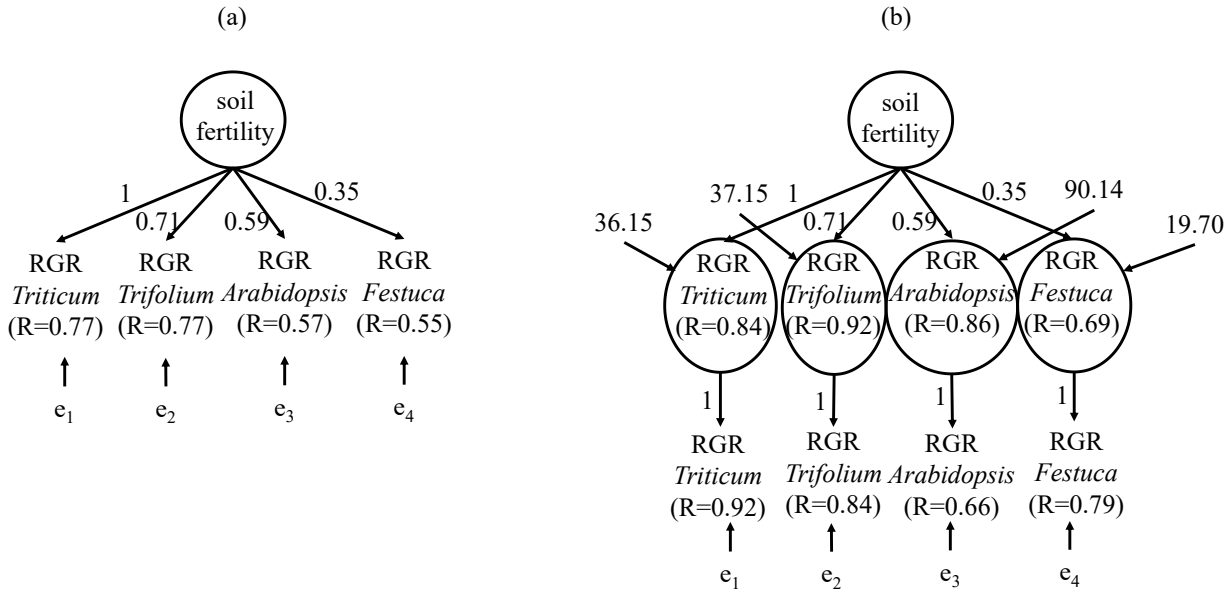


Figure 7.11. Left: the measurement model of Daou and Shipley (2019). Right: the same measurement model with the measurement error associated with the mean values of RGR of each species removed. The Pearson R values are the correlations between each effect variable and its direct cause.

This measurement model is not rejected (X^2_{ML}=0.105, 2 df, p=0.95) and so we can provisionally accept the hypothesis that the relative growth rates of these four species are all responding to a single latent cause and that once this latent cause is kept constant, the growth rates of the four species are mutually independent of each other. On the other hand, the reliability of our chosen indicator variables (the proportion of the total variance in each RGR value that is due to this common latent variable), are not particularly good, since the R^2 values are 0.60 (*Triticum aestivum*), 0.59 (*Trifolium pratense*), 0.32 (*Arabidopsis thaliana*) and 0.30 (*Festuca rubra*). The overall reliability of the measurement model is 0.78, which is acceptable.

Since Laurent Daou kept constant all possible causes of plant growth except those related to the soil, we can also provisionally conclude that this common latent cause of the growth rates of the four species is a property of soils. Since we have already defined soil "fertility" as "the propensity of a given soil to promote plant growth when all non-soil related environmental properties affecting plant growth are held constant", we are justified in calling this latent property of soils "soil fertility" although we still don't know what properties of the soil cause this latent "soil fertility". Since the measurement model in not rejected, we can provisionally

conclude that this propensity is generalizable at least to the four chosen species. We can get the estimated value of "soil fertility" (the factor scores) for each of the 76 soils as follows: `latent.fertility<-predict(fit.Daou1)+118.414`. The first part (`predict(fit.Daou1)`) is the predicted value of the latent variable and the value of 118.414 is the intercept of *Triticum aestivum*, as output from `summary(fit.Daou1)`, which we need to add so that a zero value of the measured value of "soil fertility" represents the soil fertility at which wheat has a zero growth rate. Figure 7.12 plots the estimated "soil fertility" of each soil against the observed relative growth rates of each species.

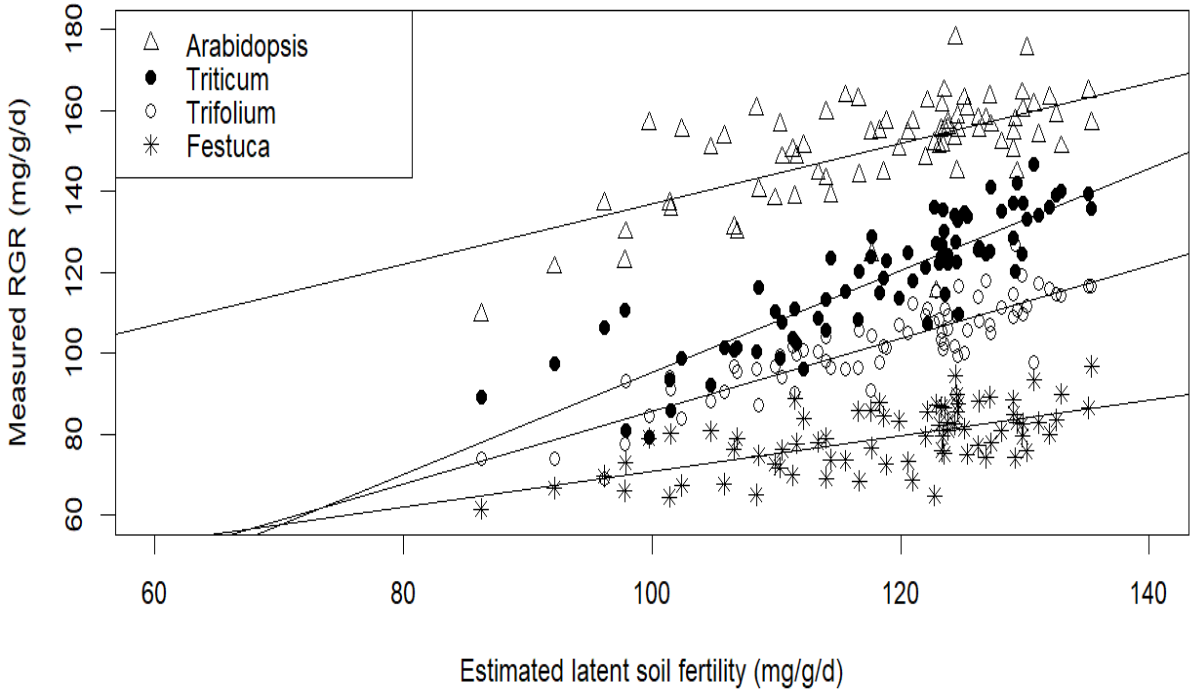


Figure 7.12. The estimated latent "soil fertility" (the factor scores) of 76 different soils is plotted against the measured relative growth rates (RGR) of each of the four chosen indicator species growth in these 76 soils. One unit of soil fertility produce 1 mg of new biomass of wheat (Triticum aestivum) per day for a plant whose aboveground biomass is 1 g.

As the "soil fertility" of a soil increases in Figure 7.12, the average RGR of each species increases, as required by our definition and by the measurement model with one common cause.

The remaining residual variation of the observed RGR values of each species around the average (the lines) represents the unique variation of each species. This residual variation is due to all those causes of differences in RGR that are unique to each species. What might these unique causes be? One such difference is simply sampling variation. Daou and Shipley (2019) had independent estimates of this sampling variation in the form of the residual variation from the regressions of ln(biomass) ~ time for each species in each soil. We can therefore remove this sampling variation by (1) creating four new latent variables representing the true values of RGR for the four indicator species, (2) fixing the path from these true latent RGR values to the sample estimates at 1.0, and (3) fixing the residual variances of the true latent RGR values to the values of the residual variation from the regressions of ln(biomass) ~ time for each species in each soil. For example, `RGR.Triticum~~36.15*RGR.Triticum` fixes the residual variance of the RGR of wheat to 36.15. The value of 36.15 is the residual variation of the regression of ln(biomass of wheat)~time. The line `true.RGR.Triticum = ~1*RGR.Triticum` creates a new latent variable linking the measured RGR of wheat (`RGR.Triticum`) to its true latent value (`true.RGR.Triticum`) with a slope of 1. This model is shown in Figure 7.11b and you can see that the reliability of the estimation of the latent "soil fertility" is much higher since the correlation coefficients between latent "soil fertility" and the true latent RGR values are much higher. The R code for the model in Figure 7.11b is:

```
Daou.mod2<-"
true.RGR.Triticum=~1*RGR.Triticum
RGR.Triticum~~36.15*RGR.Triticum
true.RGR.Festuca=~1*RGR.Festuca
RGR.Festuca~~22.12*RGR.Festuca
true.RGR.Trifolium=~1*RGR.Trifolium
RGR.Trifolium~~37.15*RGR.Trifolium
true.RGR.Arabidopsis=~1*RGR.Arabidopsis
RGR.Arabidopsis~~90.14*RGR.Arabidopsis
soil.fertility=~1*true.RGR.Triticum+true.RGR.Festuca+true.RGR.Tr
ifolium+
true.RGR.Arabidopsis
"
```

We have identified a single latent property of these soils that is the only common cause of the growth responses of our four indicator species. What is this property? Our definition of "soil fertility" is based on its effects on plant growth, not on the properties of soils that cause these effects and so we are now leaving the realm of hypothesis testing and entering the realm of

exploratory analysis. One obvious way of exploring the data is to calculate the correlations between the predicted values (the factor scores) of the latent "soil fertility" and certain measured attributes of the soils. In fact, Daou and Shipley (2019) did measure a series of chemical and structural properties of these soils that we thought might affect plant growth. You can see these correlations by typing[201] `round(cor(predict(fit.Daou2)[,5], Daou2019[,6:14]),2)`. The soil variables having significant correlations with the predicted values of the latent "soil fertility" were nitrate (NO_3) concentration (r=0.61), % silt (r=0.42) and soil water holding capacity (r=0.31). The resulting model, if we add soil nitrate as the cause of the latent "soil fertility", is not rejected (X^2=6.364, 5 df, p=0.2720. The output of `summary()` shows that soil nitrate concentration accounts for 52% of the variance in the latent "soil fertility" but the remaining residual variation in this latent variable is still significantly greater than zero. If these results are confirmed in an independent study, then we can conclude that (i) soil fertility applies at the level of plant communities, that (ii) it consists of a single property of soils, that (iii) a major cause of soil fertility is the available amount of nitrate ions in the soil, but that (iv) there are other unknown causes as well.

The process of defining, testing, and refining a measurement model is ongoing and requires independent testing with new data. For instance, our definition of "soil fertility" requires that it apply to all soils and all plant species but the study by Daou and Shipley (2019) only used four species. What if there are aspects of soils to which some species respond differently such that a soil could be "more fertile" for a subset of species and "less fertile" for others? If so, then we would have to modify our definition and its measurement. Xavier Lamontagne (Lamontagne and Shipley 2022) therefore decided to repeat the experiment of Daou and Shipley (2019) but with different species. To expand the range of possible soils, he therefore chose 23 of the natural sites used in Daou and Shipley (2019) that covered the range of values of the latent "soil fertility" and then added two additional soils: a commercial horticultural soil that is designed to maximize plant growth and non-acidic mine tailings (essentially crushed bedrock) from an abandoned mine. To expand the range[202] of plant species, he chose 9 herbaceous species typical of

[201] Since `fit.Daou2` has five latent variables, and we only want the correlations with the latent "soil fertility", we only use the fifth column in `predict(fit.Daou2)[,5]`.

[202] Besides their wide range of ecologies, the four species chosen in Daou and Shipley (2019) were chosen because their seeds are widely available to plant biologists worldwide.

grasslands but having as wide a range of ecologies as possible; one of the species was wheat (*Triticum aestivum* L.) so that he could fix the scale of the latent "soil fertility" to be comparable to the one in Daou and Shipley (2019). These data are in the data frame `Lamontagne2021`.

The measurement model was soundly rejected (X^2_{ML} =49.19, df=20, p=0.0003). Since there are only 25 observational units (soils), this sample size is certainly too small to trust the asymptotic null probability and so I calculate the Monte Carlo estimate for the null probability (Chapter 4) using the MLX2 function in the pwSEM library; this null probability is 0.001. The fact that this model is clearly rejected, even though it has quite low statistical power, means that we can confidently conclude that there is not a single common latent property of soils (what I have been calling "soil fertility") that causes differences in the growth of these nine species.

7.10 Refining the definition and measurement of "soil fertility": Exploratory SEM

After rejecting our measurement model with a single common cause of differences in growth rate across soils, we could just stop here. In fact, from the perspective of hypothesis testing, we do stop at this point. However, the purpose of this statistical analysis is not only to know if our hypothesis is false, but also to help us learn about the response of plants to soils. Therefore, we must now try to formulate a new hypothesis that accounts both for the fact that our measurement model was not rejected given the data in Daou and Shipley (2019) and that it was rejected given the data in Lamontagne and Shipley (2022). At this point, we leave the realm of hypothesis testing and enter the realm of hypothesis generation. Since we will be using these same data to help us formulate the new hypothesis, we are doing exploratory analysis. Cycling between confirmatory and exploratory statistical analysis is a normal, and necessary, part of science as long as you clearly let the reader know which activity you are doing. Chapter 9 will introduce you to some clever algorithms for exploratory SEM that can help to guide you when you don't know where to start. In our case, since our measurement model was not rejected given the data in Daou and Shipley (2019), we will start with this model and try and modify it.

The simplest explanation is that at least some of the new species in Lamontagne and Shipley (2022) are responding to an additional latent property of soils that was not experienced by the four species in Daou and Shipley (2019). We can model this by adding a second latent variable that is an additional common cause of the observed RGR values of the species. Here is the R code:

```
Lamontagne.mod2<-"
latent1 =~ 1*RGR_Ta+RGR_Cb+RGR_Fo+RGR_Ma+RGR_Ca+RGR_Cn+
RGR_Pa+RGR_Sa
latent1~~latent1
latent2 =~ 1*RGR_Ta+RGR_Cb+RGR_Fo+RGR_Ma+RGR_Ca+RGR_Cn+
RGR_Pa+RGR_Sa
latent2~~latent2
latent1~~0*latent2
"
fit.Lamontange2<-sem(Lamontagne.mod2,data=Lamontagne2021,
meanstructure=T, fixed.x=FALSE)
```

Since we don't have theory to guide us in interpreting the latent variables, I have called the two latent variables by the neutral names of "latent1" and "latent2". This new model, with two independent[203] latent causes of the differences in RGR between the eight species, was not rejected (X^2=13.04, 12 df, p=0.366); the Monte Carlo null probability was 0.457. This result is telling us that there is not one common cause of differences in RGR that is a property of soils. Rather, there are two different and independent properties of soils that are each common causes of differences in RGR. What might be the nature of the second latent variable? Since, we are in an exploratory mode, we can simply calculate the correlation coefficient between the estimated factor scores of this second latent and a series of soil properties measured in Lamontagne and Shipley (2022) and which are also in the data frame `Lamontagne2021`. Since there are two latent variables produced in `Lamontagne.mod2`, lavaan calculates the factor scores for each of these two latents and `predict(fit.Lamontange2)[,2]` produces these predicted values for the second latent. Most of these correlations were very weak but two soil properties, soil pH (r=0.68, p=0.0001) and the concentration of ammonium (NH_4^+) (r=-0.69, p=0.0001), were strongly correlated with this second latent variable but in opposite directions. Furthermore, pH and ammonium were very strongly negatively correlated (-0.75). It is well known that soil

[203] They are independent because I added `latent1~~0*latent2`. I will change this later.

pH changes the activity and composition of nitrifying and denitrifying bacteria in the soil. More acidic soils suppress nitrification, favouring ammonium accumulation, while neutral or more alkaline soils promote nitrification, increasing nitrate (NO_3^{-1}) accumulation (Brady and Weil 2017). Furthermore, although most plant species can take up both ammonium and nitrate in solution, they differ in their abilities to use these two forms of nitrogen (Marschner 2012). The pH of the soils in Lamontagne and Shipley (2022) varied from 4.96 (alkaline) to 8.10 (acidic) with a mean of 7.31 (neutral). The soils in Daou and Shipley (2019) varied from 6.03 to 8.2 with a mean of 7.03. In other words, some of the soils in Lamontagne and Shipley (2022) were more alkaline.

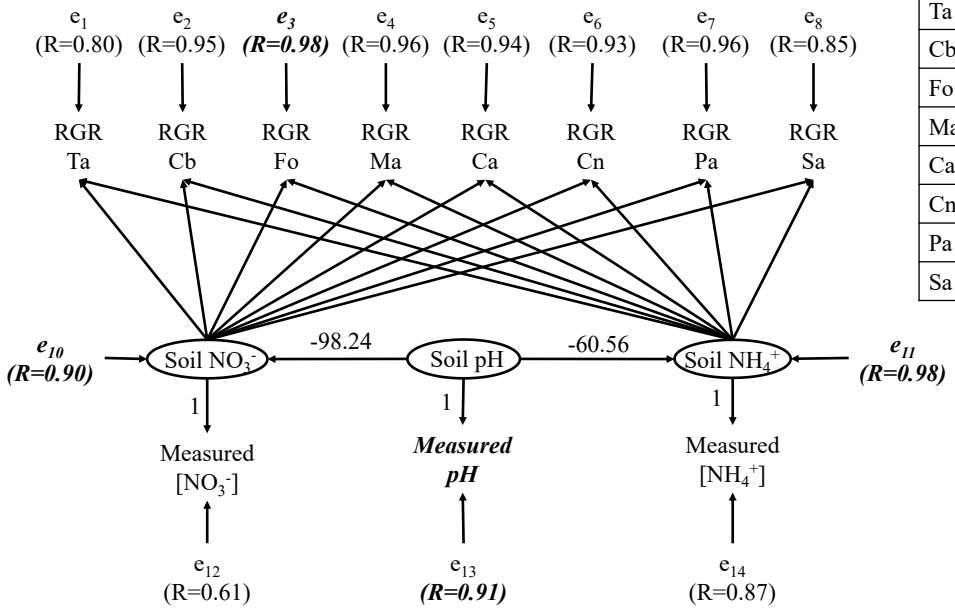
If we follow the argument presented above, concerning how soil pH changes the relative concentrations of nitrate and ammonium in soils, and how different species respond differently to the concentrations of these two alternate sources of available nitrogen in soils, then we are led to an alternative model with three latents. The two latent variables (latent1 and latent2) would be the actual availabilities of nitrate and ammonium in the soil and a third latent variable would be the actual pH of the soils (called `true.pH` in the code below). The concentrations of nitrate and ammonium and pH, were measured in the laboratory at one time, would be the effect indicators that define the scale of `latent1` and `latent2`, respectively:

```
Lamontagne.mod3<-
latent1 =~ 1*NO3.N+RGR_Ta+RGR_Cb+RGR_Fo+RGR_Ma+RGR_Ca+RGR_Cn+
RGR_Pa+RGR_Sa
latent2 =~ 1*NH4.N+RGR_Ta+RGR_Cb+RGR_Fo+RGR_Ma+RGR_Ca+RGR_Cn+
RGR_Pa+RGR_Sa
latent.pH=~1*pH
latent1~latent.pH
latent2~latent.pH
latent1~~0*latent2
pH~~pH"
```

This final model[204] is shown in Figure 7.13, and it is not rejected by the data (X^2=36.702, 34 df, p=0.345). The null probability after correcting for the small sample size is 0.568. The residual

[204] This model cannot be tested against the data from Daou and Shipley (2019) because soil ammonium concentrations were not measured in that study. Furthermore, it is not possible to introduce a second common latent cause because there were only four species, meaning that a model with two latent common causes would not be identified and could not be fit. However, adding the measured soil nitrate concentration, which was measured, as another indicator variable of the single common cause does not produce any significant lack of fit and the measured

variances of the two latent soil variables, labelled "soil NH_3^- "and "soil NH_4^+", are not significantly different from zero, meaning that this model may have captured all of their sources of systematic variance. The residual variance of the soil pH that was measured in the laboratory might[205] not significantly different from zero and this indicates that we could actually remove the latent "soil pH" and replace it directly with the measured soil pH.



Species	Soil NO_3	Soil NH_4
Ta	0.38	-0.71
Cb	0.84	-1.53
Fo	0.27	-0.32
Ma	0.37	-0.67
Ca	0.73	-1.22
Cn	0.41	-0.64
Pa	0.47	-0.63
Sa	0.38	-0.76

Figure 7.13. The exploratory model that fits the data from Lamontagne and Shipley (2022). R values in bold indicate that the residual variance of the variable is not significantly different from zero. The "e_i" represent the residual error variances of each endogenous variable. The values in the table are the path coefficients (factor loadings) from the two latent variables to the observed RGR values. Species codes: Ta (*Triticum aestivum*), Cb (*Capsella bursa-pastoris*), Fo (*Festuca ovina*), Ma (*Melilotus albus*), Ca (*Chenopodium album*), Cn (*Centaurea nigra*), Pa (*Pilosella aurantiaca*), Sa (*Sedum acre*).

The process of finding a new model that fits the data in such an exploratory manner is not as linear or straightforward as I have presented it here. In practice, one will usually try a number of different models, almost all of which will present significant lack of fit, before (maybe) finding a

soil nitrate concentration is significantly associated with the single latent cause. This suggests that the single common latent cause in Figure 7.11 and labelled "soil fertility" actually represents soil nitrate levels.

[205] The output from lavaan is 0.162±0.096 (p=0.093) but this is an asymptotic probability and the actual sample size was quite small and so the real significance level would probably be somewhat smaller than this.

model that both fits and makes biological sense. The models that are tried and rejected help to guide us. The fact that we will usually try many different models when in the exploratory phase means that we still need independent data before we can have good confidence in this new model. Remember that we are in the realm of hypothesis generation (exploratory analysis). There is nothing wrong with such an exploratory approach as long as you clearly state what you are doing. However, it is wrong to go through a long exploratory phase and then, once you have settled on a model that fits the data while making biological sense, to then present your analysis as if it were an independent test of this hypothesis. After all, the whole point of this statistical analysis is not to "find a model that fits" but rather to help us learn about how plants and soils interact as a causal system. This requires both exploratory and confirmatory phases. We must both find promising hypotheses (causal models that are not contradicted by either pre-existing causal knowledge or by the data that we used to generate them) and also independently test them.

8
Multigroup and multilevel structural equation models

Like successful politicians, good statistical models must be able to lie without getting caught. For instance, few empirical observations from nature are *really* normally distributed. The normal distribution is just a useful abstraction – a myth – that makes life bearable. In constructing statistical models, we pretend that our data follow a normal distribution and then check to ensure that our data do not deviate from it so much that the myth becomes a fairy tale. In previous chapters we saw how far we could stretch the truth about the distributional properties of our data before our data called us a liar. The goal of this chapter is to describe how SEM can deal with two other statistical myths that people often tell with respect to their data. These two myths are that (1) all of the observations in our data sets are generated by the same causal process (causal homogeneity) and (2) that these observations are mutually independent draws from this single causal process.

8.1 Causal heterogeneity and multigroup SEM

This chapter deals with multigroup and multilevel SEM. Multigroup SEM is designed to deal with situations in which different subsets of observations are generated by different causal processes, i.e. violations of causal homogeneity. Multilevel SEM is designed to deal with situations in which subsets of observations are not independent. Everything that you have learned in this book up to now applies to multilevel and multigroup models as well. If you have understood this content, then you should have no difficulty in using and understanding multigroup SEM once you have mastered a few new concepts. If you are arriving at this chapter without having read the previous chapters, then you will have difficulty unless you are already an advanced user of both piecewise and covariance-based SEM. The topic of multilevel, or mixed model, SEM requires that you already have a basic understanding of mixed model regression.

An implicit assumption of the structural equation models that have been described so far in this book, both for covariance-based SEM and piecewise SEM, is that all of the observations in your data set come from the same statistical population. We have been assuming that the same causal graph (the qualitative way in which the variables interact as a causal system) has generated all of the observations in our data set. We have also been assuming that the same structural equations (the quantitative way in which the variables in the causal graph interact as a causal system) has generated all of the observations in the data set. This is the assumption of causal homogeneity.

There are many cases in which we know (or suspect) that this is not the case. For instance, if we are studying attributes related to reproductive success in a sexually dimorphic species, then we might suspect that different causal processes are at work for males and females. Even if we believe that the causal graph is the same in males and females, is possible that the two sexes differ in the numerical strength of the causal relationships. This is an example of causal heterogeneity. Causal heterogeneity occurs when we mix together different groups of observations having different causal generating mechanisms. Imagine that an increase of one nanomole of testosterone per litre (X_1) directly doubles the level of aggressivity (X_2) in male Bighorn sheep (i.e. the path coefficient from $X_1 \rightarrow X_2$) but only increases the level of aggressivity in females by 0.2. If we were to combine observations taken from both males and females into one data set, then we would obtain an incorrect estimate for this path coefficient (somewhere between 0.2 and 2 depending on the relative frequency of males and females in the combined data set). Worse, we might incorrectly reject[206] the model even though the qualitative structure (the causal graph) of the model is correct in both groups. Perhaps our data set contains observations from several individuals of several different species. Even if the same causal graph describes the causal relationships between the variables, the quantitative strength of the path coefficients might differ between species; this too will result in causal heterogeneity if we mix together the observations from different species. Perhaps our data come from three different geographical regions, and we are not willing to assume that the same causal forces (with the same numerical strengths) apply to the observations in these different regions. Perhaps our data come from groups that we have subjected to different experimental treatments. All of these are examples of causal heterogeneity. They require that we explicitly model the causal structure (the

[206] This problem is more severe in covariance-based SEM because the estimation of free parameters and the calculation of the maximum likelihood chi-square statistic are done simultaneously.

causal graph), and properly estimate the free parameters, for each subgroup in our data. Such analyses are called "multigroup SEM".

An absolute requirement of multigroup SEM is that each observation belong to only one group. This is because the observations between groups must be independent. For instance, imagine that the individuals of some species inhabit different geographical locations and so you define each group as a "population". This if acceptable if individuals remain in the same geographical location during their lives. However, if individuals can move between geographical locations, then the same individual can belong to more than one "population" during its lifetime and so can potentially be responding to the causal processes occurring in both populations. If this occurs, then a multigroup SEM grouping individuals by population would not be appropriate.

Another absolute requirement of multigroup SEM is that the variable describing to which group an individual belongs not be an explicit part of the structural equation model. If you have a variable in the data set called "population" that describes to which population each individual belongs, then you cannot include "population" as a variable in the structural equations. This is because a multigroup SEM is modelling the causal relationships between variables (i.e. attribute values that vary between observations) within each group. By definition, the value of "population" would be a constant attribute, not a variable attribute, within each group. However, as you will learn later, we can still determine how the causal structure within each population changes from one population to another.

The first impulse is to simply conduct a separate SEM analysis for each group. In fact, a multigroup SEM in which we have allowed all free parameters to be potentially different across groups results in the same estimated free parameter values as when simply fitting the data in each group separately[207]. However, an important objective of multigroup SEM is the ability to statistically compare between groups in order to determine which parts of the models in each group (i.e. which parameters) are the same and which parts differ. You can do this using either piecewise or covariance-based SEM but the mechanics of doing this differ between the two. I

[207] However, the overall fit chi-square fit statistic is different in a multigroup SEM because it combines the fit statistics of each group.

will explain later how to determine which parts of the models in each group (i.e. which parameters) are the same and which parts differ.

I will use the data from Meziane (1998) to illustrate multigroup SEM using the lavaan and pwSEM packages. This data set is called `meziane1998.txt`. The study by Meziane (1998) consisted of individuals of 22 species of herbaceous plants grown under controlled conditions in four different experimental environments: high (N) and low (n) nutrient concentrations in hydroponic culture crossed with high (L) and low (l) light intensities. This gave four different groups of data corresponding to the four different experimental environments. The variable in the data set that describes to which of the four experimental groups each observation belongs is called `lightXnutrients` and contains the characters NL, Nl, nL, nl to encode the four combinations[208] of light intensity and nutrient concentrations. Four morphological attributes of leaves were measured: the water content of the leaf, the thickness of the lamella, the thickness of the midvein and the specific leaf area (the ratio between the projected leaf area and its dry weight). Because Meziane (1998) was only interested in interspecific differences, he averaged the values of individual plants belonging to the same species in each environment. Each line of the data therefore consisted of the mean[209] value of a species for each of the four attributes of leaves when grown in one of the four environments. Due to a few missing values, there were a total of 80 lines in the final data set. A previously published study (Shipley 1995) had described a path model relating these variables, and one objective of Meziane (1998) was to see if the previous path model could be applied under different environmental conditions. If Meziane had simply tested his path model using these 80 lines of data without distinguishing between the four different environmental treatments, then he would have implicitly assumed that the different environments had no effect of the relationships between the four variables. By "no effect" I mean both that the qualitative causal relationships linking the four leaf variables (the causal graph) are the same and also that the path coefficients, intercepts[210] and residual variances do not

[208] Capital letters refer to the "high" level and lower-case letters refer to the "low" levels of light and nutrients.
[209] If the actual values for each individual had been reported, rather than the mean values per species, then we could combine a multigroup model and a multilevel model together. This would be rather straightforward using pwSEM but somewhat more complicated using covariance-based SEM.
[210] If you remember that each variable is centred around its mean in the data set in lavaan, then combining the data from all four environments would also implicitly require that the treatments did not affect the mean values of the variables either. By separating the data into the four groups, the variables are centred around their respective group

differ between environments. Using a multigroup analysis, he specified four sets of structural equations (one for each of the four groups representing the four experimental treatments). The structural equations in each group followed the same causal structure (i.e. the same DAG) but each potentially differed in the numerical strengths of the free parameters (Figure 8.1). In this model there are five free path coefficients and four free error variances (ε) in each of the four experimental environments.

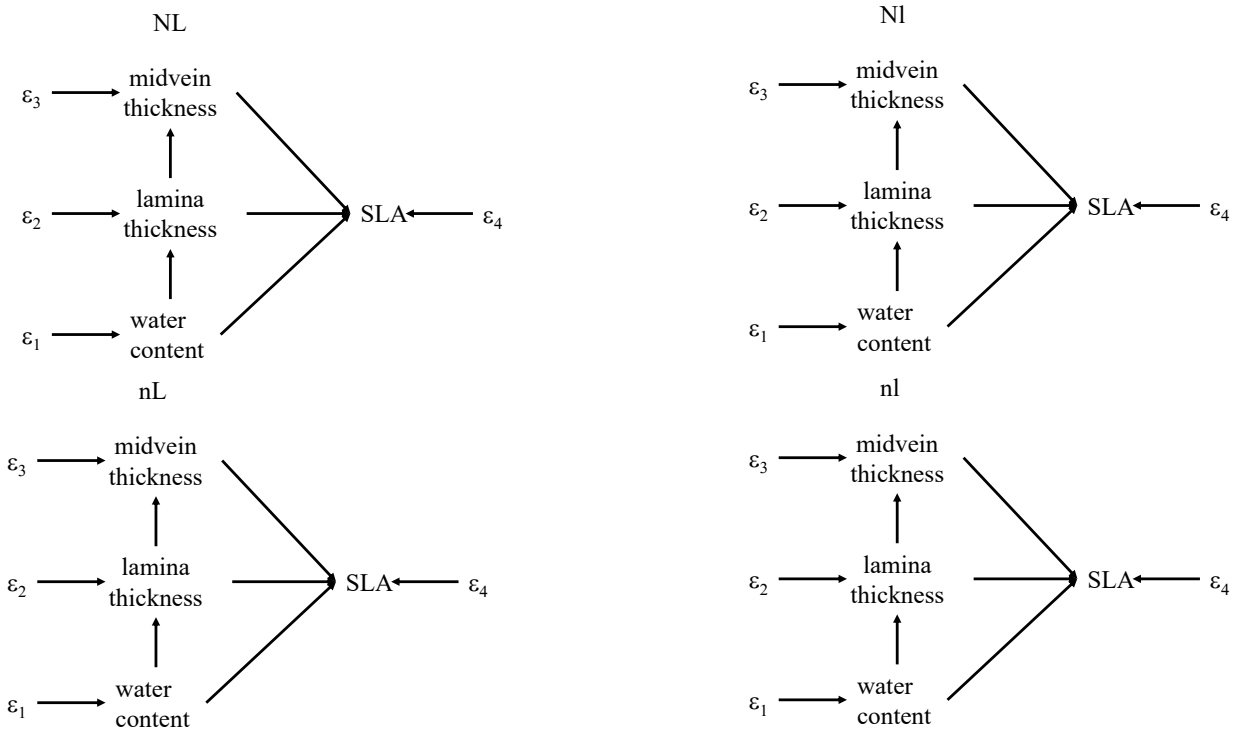


Figure 8.1. The same causal graph (a DAG) is hypothesized to structure the causal relationships between the four observed attributes of leaves in each of the four experimental conditions (NL: high nutrients, high light, Nl: high nutrients, low light, nL: low nutrients high light, nl: low nutrients, low light). "SLA" is the acronym for "specific leaf area", which is the ratio of the projected surface area of a leaf divided by its dry mass.

8.2 The chi-squared distribution in multigroup SEM

means. In this way, the treatment effects on the means are removed and only the relationships between the variables are analysed.

A multigroup model can be fit using covariance-based SEM with a minor modification of the method that you already know. In fact, the covariance-based SEM that you already know is simply a multigroup model with only one "group". With only one group we have only one observed covariance matrix ($\mathbf{S_1}$). Remember the steps in Chapter 4. We set up the model covariance matrix ($\mathbf{\Sigma_1}$) using covariance algebra and iteratively find values of the free parameters of $\mathbf{\Sigma_1}$ that minimise the maximum likelihood chi-square statistic:

$X^2 = (N_1)\left(Ln\left|\mathbf{\Sigma}_1(\boldsymbol{\theta}_1)\right| + trace\left(\mathbf{S_1}\mathbf{\Sigma_1}^{-1}(\boldsymbol{\theta}_1)\right) - Ln\left|\mathbf{S_1}\right| - p_1\right)$. This is the same formula that you saw in Chapter 4 except that I have added the subscript "1" to emphasise that we are referring to this single group. When our data are divided into g different groups with N_1, N_2, …, N_g observations in the different groups, then we have g sample covariance matrices ($\mathbf{S_1}$, $\mathbf{S_2}$, …, $\mathbf{S_g}$) and g population covariance matrices ($\mathbf{\Sigma_1}$, $\mathbf{\Sigma_2}$, …, $\mathbf{\Sigma_g}$). Each population covariance matrix can potentially have different sets of free and fixed parameters or even different sets of variables. We iteratively choose values of these free parameters simultaneously to minimise:

$X^2 = \left[(N_1)\left(Ln\left|\mathbf{\Sigma_1}(\boldsymbol{\theta}_1)\right| + trace\left(\mathbf{S_1}\mathbf{\Sigma_1}^{-1}(\boldsymbol{\theta}_1)\right) - Ln\left|\mathbf{S}\right| - p_1\right)\right] + \cdots +$

$\left[(N_g)\left(Ln\left|\mathbf{\Sigma_g}(\boldsymbol{\theta}_g)\right| + trace\left(\mathbf{S_g}\mathbf{\Sigma_g}^{-1}(\boldsymbol{\theta}_g)\right) - Ln\left|\mathbf{S_g}\right| - p_g\right)\right]$. Although this equation looks intimidating, it is simply the sum of the maximum likelihood chi-square statistics for each group.

The value of this multigroup maximum likelihood chi-square statistic (X^2) also asymptotically follows a chi-squared distribution. The maximum likelihood chi-squared statistic over all groups is the sum of the maximum likelihood chi-squared statistics of each group: $X^2 = \sum_{i=1}^{g} X_i^2$. The total degrees of freedom over all groups is the sum of the degrees of freedom of each group: $df = \sum_{1=1}^{g} df_i$.

The same property applies to Fisher's C statistic (which also follows a chi-squared distribution given the null hypothesis) in piecewise SEM. Fisher's C statistic is defined as $C = -2\sum_{i=1}^{k} \ln(p_i)$, where p_i is the null probability associated with the i^{th} independence claim in the union basis set

and k is the number of claims of independence in the union basis set (Chapter 3). The degrees of freedom are equal to 2k. If the data have been divided into g groups, each with its DAG or MAG (or m-equivalent MAG), then the C statistic over all groups is $C = \sum_{j=1}^{g} C_j = -2\sum_{j=1}^{g}\sum_{i=1}^{k} \ln\left(p_{ij}\right)$. The total degrees of freedom over all groups is the sum of the degrees of freedom of each group: $df = \sum_{j=1}^{g} df_j$.

8.3 The basic lavaan syntax to fit a multigroup model

Everything that you have learned about covariance-based SEM with lavaan applies to multigroup models as well, including the inclusion of explicit latent variables. There are two basic different ways fitting a multigroup model in lavaan. Here is the simplest way:

```
my.mod1<-"
lamina.thickness~percent.water.content
midvein.thickness~lamina.thickness
specific.leaf.area~ percent.water.content +
lamina.thickness + midvein.thickness
"
fit<-sem(model=my.mod1,data=meziane,group="lightXnutrients")
summary(fit)
```

This syntax is almost identical to what you have already learned. The only difference is the `group="lightXnutrients"` argument in the `sem()` function. This argument tells lavaan that there is more than one group and that the variable specifying the group membership of each line in the data set (`meziane`) is called `lightXnutrients`. Since the variable `lightXnutrients` defines the group membership, it cannot also appear in the structural equations. Since there is only one set of structural equations given in the model object (`my.mod1`), lavaan applies the same set of structural equations to each group. By default, since the free parameters are not explicitly named, lavaan does not place any between-group equality constraints when estimating the values of the free parameters across groups (I will explain what between-group equality constraints are, and why you will want them, later). Although the variances (exogenous and residual) are not explicitly modelled in the model object, remember that lavaan will include these by default and (since they are not given explicit names), no

between-group equality constraints will be included on them. The resulting chi-squared statistic is 6.477, the degrees of freedom are 4, and the null probability is 0.166. We cannot reject this multigroup model.

A slightly more complicated syntax is needed if you want to specify different sets of structural equations, i.e. different causal graphs, for different groups. The following code will specify exactly the same model as above:

```
my.mod<-"
group:NL
lamina.thickness~percent.water.content
midvein.thickness~lamina.thickness
specific.leaf.area~percent.water.content+lamina.thickness+midvei
n.thickness
group:Nl
lamina.thickness~percent.water.content
midvein.thickness~lamina.thickness
specific.leaf.area~percent.water.content+lamina.thickness+midvei
n.thickness
group:nL
lamina.thickness~percent.water.content
midvein.thickness~lamina.thickness
specific.leaf.area~percent.water.content+lamina.thickness+midvei
n.thickness
group:nl
lamina.thickness~percent.water.content
midvein.thickness~lamina.thickness
specific.leaf.area~percent.water.content+lamina.thickness+midvei
n.thickness
     "
```

The syntax `group:NL` tells lavaan that the structural equations that follow apply only to the group called "`NL`". `NL` is one of the values in the grouping variable `lightXnutrients`. There are four sets of structural equations, each following a line that begins with `group:`, because there are four unique values (`NL, Nl, nL, nl`) in the grouping variable `lightXnutrients`. In this case, since exactly the same set of structural equations are specified in all four groups, the result is identical to the more simplified model object presented previously, but you can change the structural equations within a given group if you believe that a different causal graph is appropriate for that group. We will add more complexity to the model object later when we discuss between-group equality constraints.

The summary output is similar, but not identical, to the output to which you are already familiar. Here is the output relating to the maximum likelihood chi-square statistic:

```
Test statistic                                6.477
Degrees of freedom                                4
P-value (Chi-square)                          0.166
Test statistic for each group:
  NL                                          0.527
  Nl                                          0.010
  nL                                          0.606
  nl                                          5.333
```

The overall maximum likelihood chi-squared statistic (6.477) is not rejected at the 5% significance level (p-value = 0.166). The only constraint that we have placed on the model covariance matrices of the four groups is that the causal graph be the same. We have not required that the numerical values of any of the free parameters be the same across groups. Therefore, this first step is only testing the hypothesized qualitative causal links (the causal graph). Notice that there is both a value for the global maximum likelihood chi-square statistic (6.477) and global degrees of freedom (4) and also chi-square values for each of the four groups. The degrees of freedom for the test statistic within each group are not printed but you can easily calculate that there is 1 degree of freedom (Chapter 3) for each group; this is why the global degrees of freedom[211] are 4. The sum of the chi-square values over the four groups equals the global chi-squared value[212]. The other difference in the summary output is that parameter estimates are output for each of the four groups. By default, as done here, lavaan will estimate different values for each free parameter across the groups and each different value will use up one degree of freedom. Later, I will explain how to force lavaan to estimate the same value for a set of free parameters across groups. We will want to do this in order to determine which values of free parameters differ across groups and which values of free parameters are the same across groups.

8.4 The basic pwSEM syntax to fit a piecewise multigroup model

[211] $\sum_{i=1}^{g=4} df_i = 4$

[212] The slight difference is simply rounding error.

The pwSEM package does not have any explicit way of modelling multigroup SEM. However, it is quite straightforward to fit the basic multigroup model in pwSEM without between-group equality constraints. Everything that you have learned about modelling piecewise SEM of causal graphs that do not include explicit latent variables using pwSEM applies with multigroup piecewise SEM as well. You simply create different data frames to hold the data in each group, fit the piecewise model to the data in each group, and then sum the C-statistics and degrees of freedom across the groups. For instance, to fit the model in Figure 8.1 to the first group using pwSEM you would use the following code:

```
NL<-meziane[meziane$lightXnutrients=="NL",]
my.mod<-list(gam(percent.water.content~1,data=NL),

gam(lamina.thickness~percent.water.content,data=NL),
             gam(midvein.thickness~lamina.thickness,data=NL),

gam(specific.leaf.area~percent.water.content+lamina.thickness+
                    midvein.thickness,data=NL))
fit<-pwSEM(sem.functions=my.mod,data=NL)
summary(fit)
C1<-fit$C.statistic
df1<-fit$df
```

The first line creates a new data frame called `NL`. The same data frame must be specified both in the model object and in the `data=` argument of the `pwSEM()` function. The model object (`my.mod`) specifies the structural equations that apply to this group. You then collect the resulting value of the C-statistic and the degrees of freedom. In our example, you would do this for each of the four groups, giving four values of the C-statistic (C1, C2, C3, C4) and four values of the degrees of freedom (df1, df2, df3, df4). Finally, you sum the C-statistics and the degrees of freedom, and obtain the null probability from the Chi-Squared distribution:

```
C.value<-C1+C2+C3+C4
df<-df1+df2+df3+df4
1-pchisq(C.value,df)
```

Table 8.1 shows the result after repeating this for the remaining three groups. Summing the C statistics over the four groups gives C=12.007 and summing the degrees of freedom over the four groups gives df=8. The resulting null probability is 1-pchisq(12.007,8) = 0.159. The model is

not rejected at the 5% significance level and so we can provisionally accept the hypothesised DAG for all four experimental environments.

Group	C statistic	Degrees of freedom
NL	1.622	2
Nl	0.127	2
nL	1.640	2
nl	8.619	2

Table 8.1. The C statistics and degrees of freedom for the dsep tests conducted on each group, given the DAGs in Figure 8.1.

8.5 *A priori* tests of significance in multigroup SEM

Sections 8.3 and 8.4 described how to test the most basic causal hypothesis in multigroup SEM: whether or not the qualitative causal relationships (the way the variables are linked in a causal system as specified in the hypothesized causal graph) is supported by the data. Once you have tested the hypothesized causal graph in your multigroup model, and have not rejected it, you can proceed to determine which (if any) free parameters might differ across groups. These free parameters can be path coefficients, intercepts or free covariances. How you proceed depends on whether or not you have established specific *a priori* hypotheses concerning which values of which free parameters you expected to differ across groups. The most common situation is that you do not have *a priori* expectations and simply want to know which free parameters are likely to be different. This is a *post hoc* question, not an *a priori* question, and will be dealt with in section 8.6.

Piecewise SEM involves two separate steps. The first step, the dsep test, only tests the causal claims implied by the causal graph and these causal claims are captured by the union basis set. The only *a priori* hypothesis that can be tested in this first step is whether or not the data were generated by the causal graph and this hypothesis is automatically tested as described in section 8.4. The second step, which is only done if the null probability of the dsep test does not result in rejection of the causal graph, involves the estimation of the free parameters in the structural

equations. The dsep test is not affected by the estimation of the free parameters. Douma and Shipley (2021) do describe an alternative type of piecewise SEM based on maximum likelihood[213] rather than d-separation, whose result simultaneously incorporates both the causal graph and the estimation of free parameters, and that test can test *a priori* hypotheses involving which free parameters might differ across groups, but that test is not implemented in the pwSEM package or in the piecewiseSEM package. You can test *a priori* hypotheses in multigroup piecewise SEM concerning equality of the values of the free parameters across groups and this is essentially an application of analysis of covariance to the structural equations. This method is described in Douma and Shipley (2021), along with R scripts, but it is not incorporated into the pwSEM package, and so you will have to do it step by step.

You can test *a priori* hypotheses in multigroup covariance-based SEM concerning which free parameters might differ across groups. Such *a priori* hypotheses involve the comparison of a reference model and a nested model. What is a nested model? Given two SEM models with the same set of variables, one model is *nested* within a second one if (i) all of the fixed parameters in the nested model are also fixed to the same values in the reference model, but (ii) some of the free parameters in the nested model are still fixed in the reference model. In other words, the fixed parameters in the nested model are a subset of the fixed parameters in the reference model. The notion of nesting can be grasped most easily by comparing some path diagrams (Figure 8.2).

[213] It involves setting up a "saturated" set of structural equations and calculating the change in likelihood between the hypothesized model and the saturated model. Research is ongoing concerning the best way to fit the saturated model, especially when the data have a nested structure.

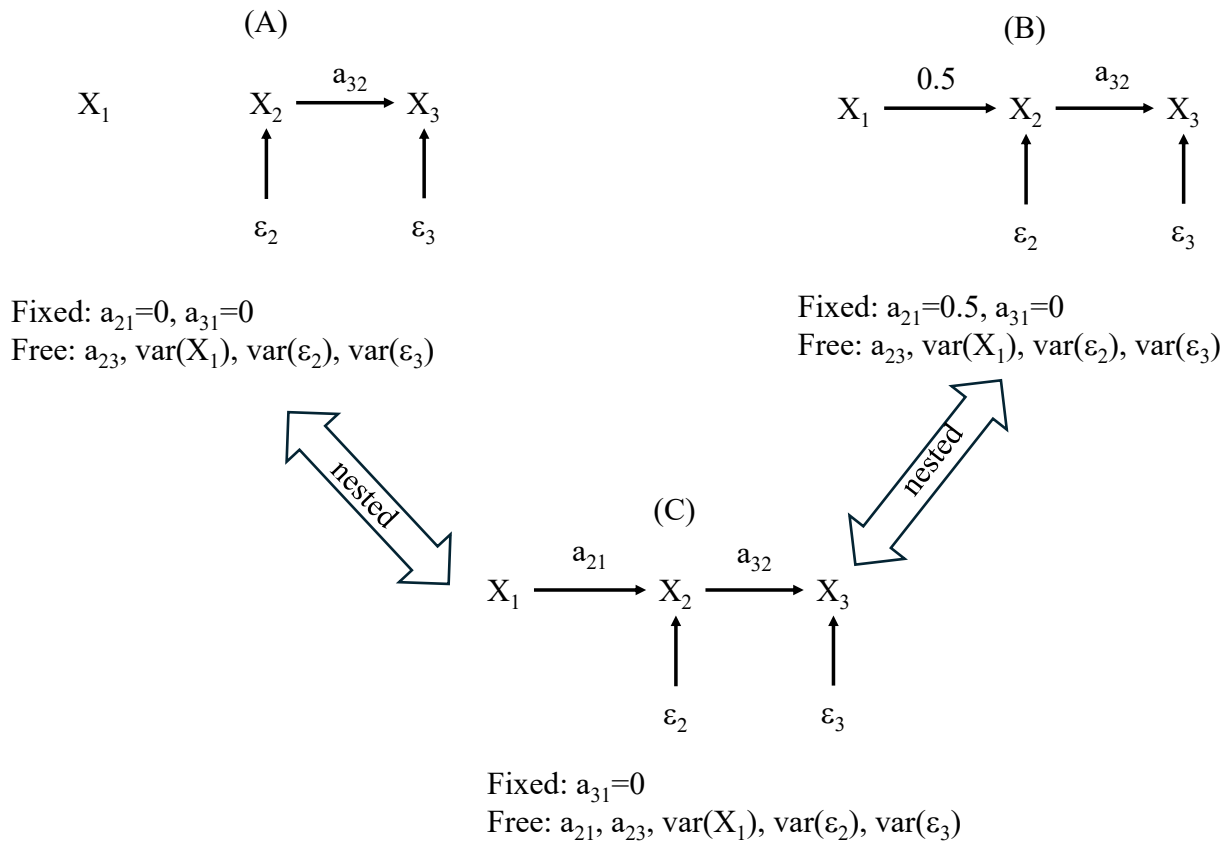


Figure 8.2 Illustration of the notion of a nested model. Model C is nested within both models A and B, but neither models A or B are nested within the other.

Model C in Figure 8.2 is nested within both models A and B. I will call models A and B (the ones into which model C is nested) the "reference" models. Reference model A has two fixed parameters, since the path coefficients for the edges between X_1 to X_2 (i.e. a_{21}=0) and between X_1 to X_3 (i.e. a_{31}=0) have each been fixed to zero. Remember that a path coefficient of zero means that there is no direct effect. Therefore, a_{21}=0 means that there is no edge between X_1 and X_2 and a_{31}=0 means that there is no edge between X_1 and X_3. The nested model C has only one fixed parameter (a_{31}=0) since the $X_1 \rightarrow X_3$ edge is still missing. The set of fixed parameters in the nested model C (a_{31}=0) is therefore a subset of the set of fixed parameters in the reference model A (a_{21}=0, a_{31}=0) and so model C is nested within model A. Reference model B also has two fixed parameters (a_{21}=0.5, a_{31}=0). A "fixed" parameter doesn't necessarily mean fixed at zero, only that the parameter value cannot change during maximum likelihood estimation. The set of fixed parameters in the nested model C (a_{31}=0) is also a subset of the set of fixed parameters in

the reference model B. However, even though models A and model B both have the path coefficients a_{21} and a_{31} as fixed parameters, a_{21} is not fixed to the same values in models A and B. Therefore, models A and B are not nested within each other.

Nested models are useful because the difference in the values of the maximum likelihood chi-square statistics between nested models is, itself, asymptotically distributed as a Chi-Squared variate if the freely estimated parameters are equal to their associated fixed parameters. The degrees of freedom of this change in the chi-square values equals the number of parameters that have been freed in the nested model but that remained fixed in the reference model, which is the same as the change in the degrees of freedom between the nested models.

Intuitively, the testing of a nested model uses the following logic: One starts with a reference model (call it model 1) in which a set of parameters are fixed to particular values (zero or otherwise). Now, we define a nested model (call it model 2) by freeing some previously fixed parameters but without changing anything else relative to reference model 1. If we allow some of these previously fixed parameters to be freely estimated, and if the data generating process in nature really does use the value to which these newly freed parameters had previously been fixed, then the only difference in the estimated covariance matrices between models 1 and 2 will be due to random sampling variation. If this is true, then the difference between the maximum likelihood chi-square statistics will also follow a chi-square distribution with degrees of freedom equal to the number of previously fixed parameters that have been freed in the nested model 2. Here are the steps:

1. Fit the reference model, obtain its chi-square value (X_1^2) and its degrees of freedom (df_1).

2. Fit the nested model, obtain its chi-square value (X_2^2) and its degrees of freedom (df_2).

3. Calculate the change in the chi-square value and the change in the degrees of freedom: $\Delta \mathrm{X}^2 = \mathrm{X}_1^2 - \mathrm{X}_2^2$ and $\Delta df = df_1 - df_2$.

4. Determine the probability of having observed this change in the chi-squared value $\left(\Delta X^2\right)$ assuming that the freed parameters in the second (nested) model are equal to those in the first

model, except for random sampling variation. You will have to use the `pchisq()` function of R to get this null probability.

5. If this probability is less than the chosen significance level, conclude that the freed parameters were not the same as those fixed in the first model.

There are typically many different possible nested models within a given reference model. For instance, in Figure 8.1, there are five path coefficients, four intercepts, and four residual variances that could potentially be different across the four groups and you could allow these free parameters to vary across all four groups or any combination of these four groups. That makes a lot of different possible nested models relative to a reference model! If you test N different nested models, then you are sequentially testing N null hypotheses and must adjust the significance level using a Bonferroni adjustment, in which the overall significance level (typically α=0.05) is divided by the number of tests that you have conducted (N), or the false discovery rate (Benjamini and Hochberg 1995). For instance, if you test 5 different nested models then the significance level of each test would be 0.05/5=0.01 using the Bonferroni adjustment. You would reject the null hypothesis that a particular nested model gave the same level of fit as the reference model only if the resulting null probability was less than 0.01. However, a Bonferroni adjustment can result in a conservative rejection rate. Therefore, I recommend that you not use this method of nested models as a *post hoc* method of determining which free parameters differ across groups[214]. A better method is to use AIC statistics for such *post hoc* exploratory questions, as described in section 8.3. AIC statistics can be used for both covariance-based SEM and piecewise SEM. However, this *a priori* method is still useful to ask the most basic question: do any of a set of free parameters vary across groups? If the answer is "no", then you would not proceed to the *post hoc* AIC analysis. If the answer is "yes" then you can proceed to the *post hoc* AIC analysis.

As an example, consider the null hypothesis that the values of all of the path coefficients and intercepts are equal within the four experimental groups in Meziane (1998). In other words, this *a priori* hypothesis states that although the values of the different free parameters (path coefficients, intercepts, variances or free covariances) can differ within a single experimental

[214] Despite the fact that I did recommend this in the first two editions of this book!

group, the value of the same free parameter (for example a path coefficient or an intercept) is the same in each of the four experimental groups. What is the appropriate reference model? Since the null hypothesis does not concern the equality of residual variances across groups, we do not force these residual variances to be the same across groups[215].

The first step is to fit a reference model in which both the intercepts and the path coefficients are forced to be equal in all groups. Lavaan has a number of different ways of forcing free parameters to be equal across groups, but I will show you the most flexible way. In Chapter 3, I explained how one can assign names to the free parameters. When more than one free parameter has the same name, lavaan will force those free parameters having the same name to have the same value during maximum likelihood estimation. In other words, we are telling lavaan: "you can choose the best value for the free parameters having the same name during maximum likelihood estimation, but you must choose the *same* value for all of them". By using this trick, you can specify any pattern of constraints on the free parameters between groups. Here is how to specify the reference multigroup model[216] while forcing all of the intercepts and all of the path coefficients to be equal across groups:

```
my.mod1<-"
percent.water.content~c(a,a,a,a)*1
lamina.thickness~c(b,b,b,b)*1+c(c,c,c,c)*percent.water.content
midvein.thickness~c(d,d,d,d)*1+c(e,e,e,e)*lamina.thickness
specific.leaf.area~c(f,f,f,f)*1 +
c(g,g,g,g)*percent.water.content +
c(h,h,h,h)*lamina.thickness + c(i,i,i,i)*midvein.thickness
"
fit<-sem(model=my.mod1,data=meziane,group="lightXnutrients")
summary(fit)
```

The "1" indicates an intercept, just like in other packages that perform regression-type analyses in R. There are four such intercepts, one for each of the variables in the model. The line "`percent.water.content ~ c(a,a,a,a)*1`" tells lavaan to estimate the intercept of the variable "percent.water.content" and, because the same name ("a") has been assigned to the intercept in all the four groups, lavaan must estimate the same value for this intercept in each of

[215] It is, of course, possible to include the equality of the residual variances if this is of biological interest.
[216] This reference multigroup model has the same structural equations for all groups. If we wanted to have different structural equations (causal graphs) for different groups, then we would use the "`group:`" line.

the four groups. Since percent water content is an exogenous variable, its intercept is also its mean value and so lavaan will estimate the same value for the intercept (i.e. mean) of the percent water content in all four experimental groups. Because each of the nine free parameters (four intercepts and five path coefficients) in a given group are assigned different names ("a" to "i"), lavaan will estimate different values for them. Because the *same* names have been assigned to the same free parameter in each of the four groups, lavaan will estimate the same value for that particular free parameter in each of the four groups. You are free to choose whatever names you want as long as they agree with the naming conventions of R and those names are not already used for variables in the data set. If you do this for each of the intercepts and path coefficients in the reference model, then you are defining a reference model in which all of the path coefficients and intercepts are equal across groups. The variances of the exogenous variables (including the residual variances) have not been given the same names across groups and so these free parameters are not constrained to be equal across groups. In summary, the above code instructs lavaan to fit a multigroup model based on the group names found in the "lightXnutrients" variable of the data frame called "meziane", to use the same causal graph for all four groups, and to constrain the values of the same intercept or path coefficient to be equal across groups. The resulting global chi-squared statistic is 86.790, the global degrees of freedom are 31, and the null probability is less than 0.0005. The reference model, in which all groups have the same causal graph and the same values for all of the intercepts and path coefficients, is clearly rejected. This does not mean that the underlying causal graph has been rejected. In fact, since we already tested this causal graph before, and failed to reject it, when allowing all of the free parameters to differ across groups, we know that the rejection is not related to the causal graph. Rather, the rejection is due to the constraint the values of the equivalent free parameters be identical in all groups.

Of particular interest is whether the *change* in the chi-squared statistic is significant when we compare the fit of the reference model to the fit of the nested model. To do this, we need to fit a second model, nested within the first model, in which the values of each of the four intercepts and each of the five path coefficients are allowed to differ across the four groups[217]. In fact, we could simply not assign names to any of the free parameters because, by default, lavaan will

[217] And, by default, all of the residual variances.

estimate different values for free parameters if they do not have the same explicit names, but I include these different names to make the comparison clear.

```
my.mod2<-"
percent.water.content~c(a1,a2,a3,a4)*1
lamina.thickness~c(b1,b2,b3,b4)*1+c(c1,c2,c3,c4)*percent.water.c
ontent
midvein.thickness~c(d1,d2,d3,d4)*1+c(e1,e2,e3,e4)*lamina.thickne
ss
specific.leaf.area~c(f1,f2,f3,f4)*1+c(g1,g2,g3,g4)*percent.water
.content+
c(h1,h2,h3,h4)*lamina.thickness+c(i1,i2,i3,i4)*midvein.thickness
"
fit<-sem(model=my.mod2,data=meziane,group="lightXnutrients")
summary(fit)
```

Notice that the name of a given free parameter is now different across the four groups. For example, the first line of the script is `percent.water.content ~ c(a1,a2,a3,a4)*1`. Because the names of the intercept of `percent.water.content` are now different from one another, `c(a1,a2,a3,a4)`, rather than being the same , `c(a,a,a,a),` as in the reference model, lavaan will estimate different values for this intercept in each group. The global chi-squared statistic for `my.mod2` is 6.477 with 4 degrees of freedom. The change in the chi-squared statistic between the reference model and this nested model is `delta.X2<-86.790-6.477` and the change in the degrees of freedom are `delta.df<-31-4`, giving a null probability for the change in the chi-squared statistic of `1-pchisq(delta.X2,delta.df)`; we use `1-pchisq()` because we want the probability of observing at least this amount of the maximum likelihood chi-squared statistic, i.e. the tail probability. The resulting null probability ($3.39X10^{-7}$) is highly significant, meaning that there are almost certainly some differences in the intercepts and path coefficients across the groups.

When you use the Satorra-Bentler correction to the chi-squared statistic (in order to deal with non-normality) by including the argument `estimator="MLM"` in the `sem()` function, you must calculate the difference in chi-squared values slightly differently (Kline 2016). Here are the steps:

1. Use the unscaled maximum likelihood chi-squared statistics of the reference (X^2_R) and nested (X^2_N) models to calculate the change in the chi-squared statistic ($\Delta X^2 = X^2_R - X^2_N$). The change in the degrees of freedom remain the same ($\Delta df = df_R - df_N$).
2. Get the scaling correction factors for the reference (c_R) and nested (c_N) models. Each is printed in the `summary()` output under the heading "Scaling correction factor".
3. The bias-corrected (or scaled) difference in the chi-squared statistic ($\Delta X^2_{S\text{-}B}$) is

$$\Delta X^2_{S\text{-}B} = \frac{\Delta X^2}{\left(c_R df_R - c_2 df_N\right) / \Delta df}$$

4. Get the null probability as $1\text{-pchisq}\left(\Delta X^2_{S\text{-}B}, \Delta df\right)$.

8.6 *Post hoc* analysis in covariance-based multigroup SEM

We have decided to reject the reference model, with the values of all of the intercepts and path coefficients forced to be equal across the four experimental groups, because its null probability was only 0.0005. We have decided that the nested model, in which the values of all of the intercepts and path coefficients are allowed to be different across the four experimental groups, cannot be rejected because its null probability was 0.166. We have also decided, based on the change in the global chi-squared statistic between the reference and nested model, that at least one of the intercepts or path coefficients differs from the others within some of the experimental groups because the null probability was 3.39×10^{-7}. Which of these intercepts or path coefficients differ between groups?

If you have an *a priori* hypothesis of which specific free parameters should be equal across groups then you would test this hypothesis using the change in the chi-squared value, as explained above. Often, however, we lack such an *a priori* hypothesis. Instead, we want to explore different models with different combinations of equality constraints across groups, in order to see which free parameters differ in which groups in a *post hoc* manner. The most appropriate way to do this is by using the AIC statistic, as explained in Chapter 5. I used the AIC statistic in Chapter 5 to choose between a set of models having different causal graphs that

had not been rejected. Here, I will explain how to use AIC statistics to choose between a set of models having the same causal graph but differing in the number of equality constraints across groups that these models impose. This can be done in both covariance-based and piecewise SEM. However, even small causal graphs like the one in Figure 8.1 can generate a large number of possible comparisons of equality constraints since each possible combination of free parameters can be compared for each possible combination of groups. You should decide which combinations of free parameters and groups are of interest. Remember that you are no longer conducting *a priori* tests of significance but are instead asking *post hoc* questions.

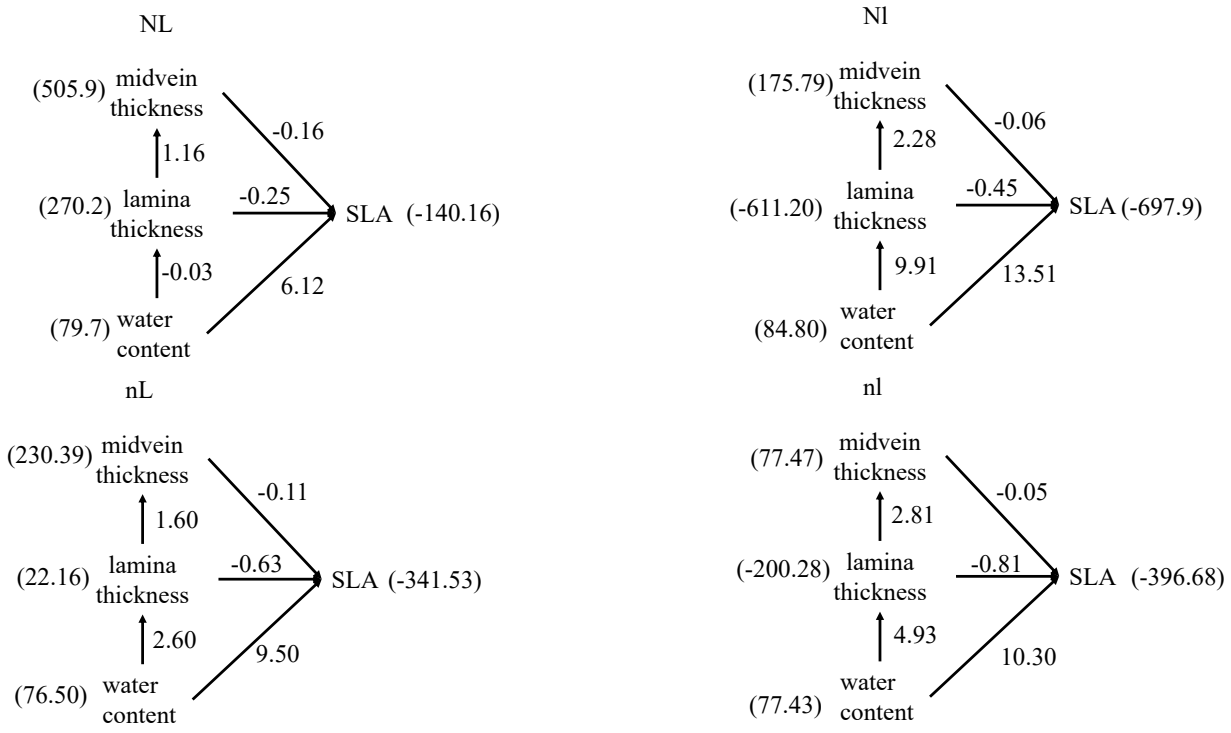


Figure 8.3. The maximum likelihood estimates of the path coefficients and intercepts (in brackets) of the multigroup model shown in Figure 8.1 when no equality constraints were applied across groups. The values of the variances of water content and the residual variances are not shown for simplicity.

Figure 8.3 shows the values of the path coefficients and intercepts (shown within brackets) in each of the four experimental groups when the data were fit to the nested model in which no equality constraints involving path coefficients or intercepts across groups were placed on the data. We have already decided that at least one of these values differ across the four experimental treatments. Let's first see which intercepts are likely to be different. We start with

the reference model, in which all of the path coefficients and intercepts are forced to be equal across groups, and obtain the AIC value, which is 3503.399.

```
my.mod<-"
percent.water.content~c(a,a,a,a)*1
lamina.thickness~c(b,b,b,b)*1+c(c,c,c,c)*percent.water.content
midvein.thickness~c(d,d,d,d)*1+c(e,e,e,e)*lamina.thickness
specific.leaf.area~c(f,f,f,f)*1+c(g,g,g,g)*percent.water.content
+
c(h,h,h,h)*lamina.thickness+c(i,i,i,i)*midvein.thickness
"
fit<-sem(model=my.mod,data=meziane,group="lightXnutrients")
AIC(fit)
```

Next, we sequentially fit a series of nested models in which the value of only one path coefficient or intercept at a time is allowed to freely vary across groups. Here is the code to allow only the value of the intercept of water content to differ across groups:

```
my.mod<-"
percent.water.content~c(a1,a2,a3,a4)*1
lamina.thickness~c(b,b,b,b)*1+c(c,c,c,c)*percent.water.content
midvein.thickness~c(d,d,d,d)*1+c(e,e,e,e)*lamina.thickness
specific.leaf.area~c(f,f,f,f)*1+c(g,g,g,g)*percent.water.content
+
c(h,h,h,h)*lamina.thickness+c(i,i,i,i)*midvein.thickness
"
fit<-sem(model=my.mod,data=meziane,group="lightXnutrients")
AIC(fit)
```

The resulting AIC value is 3484.706, which is lower than the AIC of the reference model by 18.693 units. Since 18.693 units is much more than the typical cutoff of 4 units, we would definitely prefer this second model over the reference model. We repeat this for each of the three remaining intercepts and each of the five path coefficients. Table 8.2 shows the result.

Table 8.2 A series of alternative structural equation models that all use the same causal graph as shown in Figure 8.1, but that differ in the number and type of equality constraints that are imposed across the four experimental groups. Model 1 is the reference model in which the values of each intercept and path coefficient are equal across groups. Models 2 – 10 are models in which the values of only one intercept of path coefficient are allowed to be different across groups. Also shown are the AIC values for each of the models, and the change in the AIC relative to the reference model (ΔAIC_1) and the change in the AIC values for each model relative

to the best model (ΔAIC_{best}). AIC values with an asterisk (*) indicate models that are within 4 AIC units of the best model.

Model	Equality constraints across groups	AIC	ΔAIC_1	ΔAIC_{best}
1	Reference model: All path coefficients and intercepts equal	3503.399	0	33.658
2	Intercept of percent water content free	3484.706	18.693 (1)	16.965
3	Intercept of lamina thickness free	3499.066	4.333 (6)	31.325
4	Intercept of midvein thickness free	3506.909	-3.51 (8.5)	39.168
5	Intercept of specific leaf area free	3498.037	5.362 (5)	30.296
6	Percent water content→lamina thickness free	3499.716	3.683 (7)	31.975
7	Percent water content→specific leaf area free	3497.546	5.853 (4)	29.805
8	Lamina thickness→midvein thickness free	3506.909	-3.51 (8.5)	39.168
9	Lamina thickness→specific leaf area free	3491.185	12.214 (2)	23.444
10	Midvein thickness→specific leaf area free	3494.29	9.109 (3)	26.549
	Models 2+9	3472.536	30.863	4.795
	Models 2+9+10	3470.746	32.653	3.005*
	Models 2+7+9+10	3471.451	31.948	3.71*
	Models 2+5+7+9+10	3472.074	31.325	4.333
	Models 2+3+5+7+9+10	**3467.741**	**35.658**	**0**
	Models 2+3+5+6+7+9+10	3469.852	33.547	2.111*
	Models 2+3+5+6+7+8+9+10	3473.168	30.231	5.419
	Models 2+3+4+5+6+7+8+9+10	3477.086	26.313	9.345

	(All path coefficients and intercepts free)			

The strategy for identifying the best model, i.e. the model with the lowest AIC value, is similar to that used in stepwise regression, except that we are applying it to the full set of structural equations rather than to a single equation. Models 2 to 10 in Table 8.1 show the AIC values that result from removing only one between-group equality constraint. We start with model 2, in which the values of only the intercept of percent water content are allowed to vary across groups, because it is the single change that results in the largest reduction of the AIC value. We then subsequently remove additional between-group equality constraints, each time adding a single additional removal of a between-group equality constraint that results in the greatest decrease in the AIC value.

The model that has the lowest AIC value of those considered in Table 8.1 is the one in which the intercept of midvein thickness, the path coefficient of the direct effect percent water content→lamina thickness and the path coefficient of the direct effect of lamina thickness→midvein thickness are forced to be equal across groups while all other intercepts and path coefficients are allowed to freely vary across groups. This is therefore our current "best" model. There are three other models, identified by asterisks in Table 8.1, whose AIC values are within 4 units of this best model. We should therefore consider these three other models as well.

Remember that AIC statistics only help us to choose between a set of alternative models that we have decided are viable. If we enlarge the set of alternative models, then a different "best" model might emerge. For example, when looking at the values of the path coefficients of the best model so far for midvein thickness → sla in the four experimental groups that are given the summary object of this model, the values are -0.111 (±0.023) in the NL group, -0.061 (±0.027) in the Nl group, -0.108 (±0.030) in the nL group and -0.051 (±0.023) in the nl group. It looks like the NL and nL groups have very similar values, as does the Nl and nl groups. If so, then only the light treatments (L vs. l) are affecting the values of this path coefficient. To see if this might be true, we can modify the "best" model so that the values of this path coefficient are constrained to be equal in the NL and nL groups and equal in the Nl and nl groups. To do this, we only have to give the same name to the groups that are to be equal (`c(i1,i2,i1,i2)*midvein.thickness`). Since the first group (NL) and the third group

(nL) are given the same name (`i1`), the value of this path coefficient in these two groups must be the same. Since the second (Nl) and fourth (nl) groups are given the same name (`i2`), the value of this path coefficient in these two groups must also be the same.

```
my.mod2<-"
percent.water.content~c(a1,a2,a3,a4)*1
lamina.thickness~c(b1,b2,b3,b4)*1+c(c,c,c,c)*percent.water.conte
nt
midvein.thickness~c(d,d,d,d)*1+c(e,e,e,e)*lamina.thickness
specific.leaf.area~c(f1,f2,f3,f4)*1+c(g1,g2,g3,g4)*percent.water
.content+
c(h1,h2,h3,h4)*lamina.thickness+c(i1,i2,i1,i2)*midvein.thickness
"
fit<-sem(model=my.mod2,data=meziane,group="lightXnutrients")
summary(fit)
AIC(fit)
```

The resulting model is not rejected (X^2=15.21, 15 df, p=0.436) and the AIC value is 3463.823, which is 3.918 AIC units lower than the previous "best" model. Figure 8.4 shows the resulting values of each of the intercepts and path coefficients. The causal relationship leaf water content → leaf lamina thickness → leaf midvein thickness appears to be the same in all four experimental environments. The causal relationship leaf water content → leaf lamina thickness → leaf midvein thickness → SLA appears to be the same in the same light environment irrespective of nutrient status.

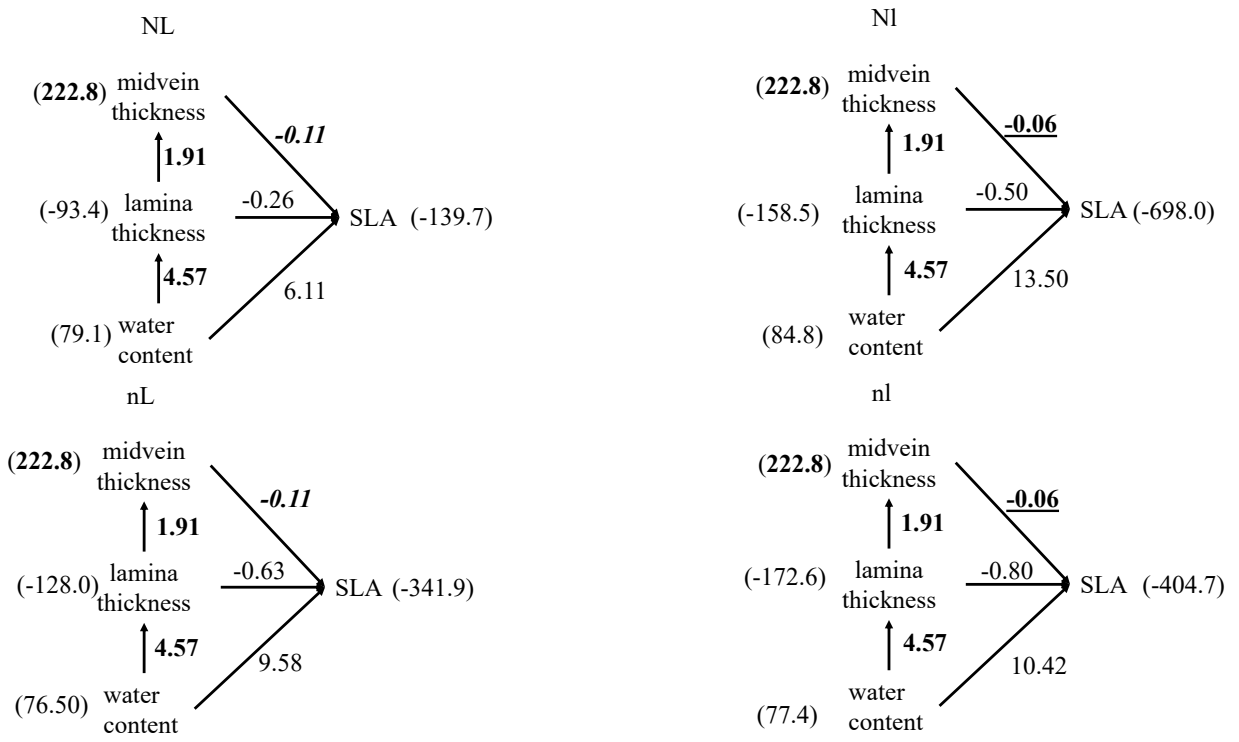


Figure 8.4 The final "best" model showing the values of the intercepts and path coefficients in each of the four experimental groups (NL: high nutrients, high light, Nl: high nutrients, low light, nL: low nutrients, high light, nl: low nutrients, low light). Values in **bold** type have been constrained to be equal across all four groups, values in ***bold italic*** type have been constrained to be equal in high light environments and values in **bold underline** type have been constrained to be equal in the low light environments.

How do you interpret these values across groups? The glib answer is that you interpret them exactly as you have done throughout this book. The intercepts tell you something about the average values of a variable and the path coefficient tells you by how much a unit change in the direct cause will change the value of the direct effect when holding constant all other variables. A more useful answer is to show you how the means and slopes (i.e. path coefficients) change in the simplest structural equation: when X is the only explicit cause of Y. Let's write out this pair of equations. There are two observed variables (X, Y). Each variable has observations in each of i=1,2 groups and has observations (indexed by j) within each group. The mean[218] value of X in the group i is $\bar{X}_i$ and the value of the j^{th} observation in group i varies according to a normal

[218] Strictly speaking, I should use Greek letters to represent the means because these are population values, not sample values, but I use Latin script here to make the meaning clearer to people who are not used to mathematical notation.

distribution with standard deviation σ_i. The mean value of Y, when X=0, in the group i is $\bar{Y}_i$ (i.e. the intercept of Y). A unit increase in X_{ij} increases the value of Y_{ij} by β units. The remaining variation of Y_{ij}, caused by the other independent and unknown causes of Y, follows a normal distribution with a mean of zero and a standard deviation of $\sigma_i^{'}$.

$$X_{ij} = \bar{X}_i + N\left(0, \sigma_{ij}\right)$$
$$Y_{ij} = \bar{Y}_i + \beta_i X_{ij} + N\left(0, \sigma_{ij}^{'}\right)$$

Figure 8.5 shows what happens if (i) neither the intercepts (means) or the path coefficient is different between the two groups, (ii) if only the mean (intercept) of Y varies between the two groups ($\bar{Y}_1 = -1, \bar{Y}_2 = +1$), (iii) if only the mean (intercept) of X varies between the two groups ($\bar{X}_1 = -1, \bar{X}_2 = +1$) and (iv) if only the path coefficient linking X→Y varies between the two groups (β_1=1, β_2=-1).

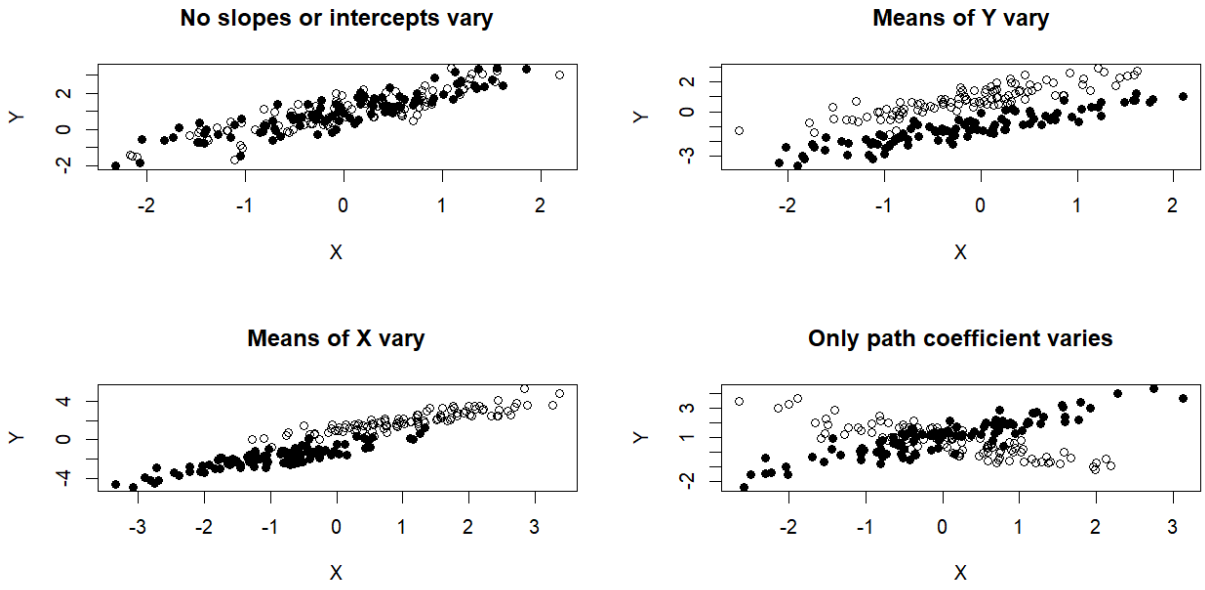


Figure 8.5. Visualising the consequences of varying the means (intercepts) or path coefficient in a simply bivariate relationship with two groups.

Given this, we can now interpret how the causal relationships between the four leaf attributes appear to change across the four experimental groups. Since leaf water content is exogenous in Figure 8.4, and its intercept varies across the groups, this is like changing the mean of X in

Figure 8.5 (bottom left). High light inputs (Nl and Nl) caused larger leaf water contents than did low light inputs[219]. The causal effect of leaf water content on leaf lamina thickness does not vary across the four environments, but the mean of leaf lamina thickness does; this is analogous to changing the means of both X and Y, while keeping the path coefficient the same. The causal effect of leaf lamina thickness on leaf midvein thickness does not vary across the four environments, nor does the mean of leaf midvein thickness; this is analogous to only changing the mean of X. Finally, changes in each of leaf water content, leaf lamina thickness and leaf midvein thickness cause changes of different magnitudes in specific leaf area across the four environments, although this is only true when changing light intensity for the relationship between midvein thickness and specific leaf area.

8.7 *Post hoc* analysis in piecewise multigroup SEM

The logic of using AIC statistics for *post hoc* analyses in piecewise multigroup SEM is exactly the same as with covariance-based SEM. I fit a multigroup piecewise SEM in section 8.4 by constructing four data frames, each holding the observations in one of the four experimental groups, fitting the piecewise SEM using the `pwSEM()` function, and then summing the C statistics and degrees of freedom. I did this so that all of the free parameters, including the residual variances, would not be constrained across groups in the least constrained covariance-based model. We can get the global AIC statistic simply by summing the AIC statistics produced in each separate call to `pwSEM()` (Chapter 5). Doing so produces an AIC statistic of 3477.086. This is the same value as produced by lavaan (last line of Table 8.2) because the model object passed to `pwSEM()` assumed (by default) linearity and normality. The strength of piecewise multigroup SEM is that we can easily change these assumptions simply by choosing different probability distributions or by using regression smoothers for nonlinear fits (via the `s()` argument), as explained in Chapter 3.

You can also obtain AIC statistics for models with various equality constraints, except that these will also have equality constraints on the residual variances. Since this will be true for all of the

[219] Although we haven't tested to see if this variable only responds to light, we could specify such a model and get its AIC value, as explained already for the path coefficient of midvein thickness→specific leaf area.

models that you compare using the AIC statistic, this will not affect your ability to identify differences in the other free parameters across groups. The key idea is that the AIC of the full set of structural equations is the sum of the AIC value of each structural equation plus any free covariances that might exist if you are working with MAGs that include them (Chapter 6). The pwSEM package includes a function called "get.AIC" that extracts the AIC values of each part of the SEM (including free covariances) and combines them. The general form is `get.AIC(sem.model, MAG, data)`. The first argument is the model object, as a list, that would be given to `pwSEM()`. The second argument is a binary matrix that encodes the causal graph in the form of a DAG or a MAG. The third argument is the data frame containing the data and this must be the same data frame that is passed to the `gam()` or `gamm4()` functions in the model object. You already learnt how to create a DAG object using the ggm library in Chapters 2 and 3. You also learnt how to create a MAG object with free covariances using the `DAG.to.MAG.in.pwSEM()` function in the pwSEM package in Chapter 6. In the present case, there are no free covariances, but I will go though all of the steps to illustrate. First, I place between-group constraints on all of the free parameters, including on the residual variances, by fitting a model using the full data set (`Meziane`) rather than using the four separate data sets, as I did above.

```
my.mod<-list(
gam(percent.water.content~1,data=Meziane),

gam(lamina.thickness~percent.water.content,data=meziane),

gam(midvein.thickness~lamina.thickness,data=meziane),

gam(specific.leaf.area~percent.water.content+lamina.thickness+
                    midvein.thickness,data=meziane))
my.dag<-DAG(lamina.thickness~percent.water.content,
                        midvein.thickness~lamina.thickness,

specific.leaf.area~percent.water.content+lamina.thickness+
                           midvein.thickness)
my.mag<-
DAG.to.MAG.in.pwSEM(full.DAG=my.dag,latents=NA,conditioning.late
nts=NULL)
get.AIC(sem.model=my.mod,MAG=my.mag,data=meziane)
```

The first call creates a list (`my.mod`) of the structural equations. If you to get the summary information (values of intercepts, slopes etc.) for each of the structural equations then you will have to fit each regression separately and extract this information via `summary()`. Notice that the first line (`gam(percent.water.content~1,data=meziane)`) is a model in which percent water content is regressed only on a single intercept. This specifies that the same intercept be estimated for each of the factor levels (i.e. NL, Nl, lN and nl) of this factor. In other words, this first line creates the between-group equality constraint on this intercept! The second call, `DAG()`, creates the DAG object (`my.dag`). If there had been latent variables in our causal graph, then these latent variables would be included in the call to `DAG()` although there are no latent variables in the present case. Since this is the case, we don't actually need to call the `DAG.to.MAG.in.pwSEM()` function and could pass the `my.dag` object directly to the `get.AIC()` function. However, for completeness, I call the `DAG.to.MAG.in.pwSEM()` function (Chapter 6) in order to convert the DAG object (which potentially includes latent variables) into the MAG object that outputs the m-equivalent MAG. In the present case, `my.dag` is identical to `my.mag` because there are no latent variables in the causal graph. Finally, the call to `get.AIC()` produces the global AIC value for the full SEM. The resulting global AIC is 3507.659, which is again slightly higher that the equivalent AIC value using lavaan (3503.399) because this model includes an equality constraint across groups for the residual variances. If your m-equivalent MAG is different from the original MAG with implicit latents, then you must use the m-equivalent MAG when setting up your structural equations, as explained in Chapter 6.

In order to allow the values of an intercept to vary across groups, you simply have to add the factor variable (`lightXnutrients`) in the formula of the appropriate `gam()`. For instance, to remove the between-group equality constraint on the intercept of percent water content, you would modify the model object as shown below and the resulting global AIC value is 3489.945:

```
my.mod<-
list(gam(percent.water.content~lightXnutrients,data=meziane),

gam(lamina.thickness~percent.water.content,data=meziane),

gam(midvein.thickness~lamina.thickness,data=meziane),
```

```
gam(specific.leaf.area~percent.water.content+lamina.thickness+
                    midvein.thickness,data=meziane))
get.AIC(my.mod,MAG=my.mag,data=meziane)
```

If you want to free the between-group equality constraint on the intercept of the equation for lamina thickness, then you would modify it to read: `lamina.thickness ~ lightXnutrient+percent.water.content`, and so on.

The easiest way of removing between-group equality constraints on path coefficients, for models that do not involve nonlinear smoother functions, is to use the FACTOR/X syntax that is common to most linear model packages in R, where FACTOR is the name of the factor variable for the groups (here, called `lightXnutrients`) and X is the name of the continuous variable whose path coefficient is being allowed to vary across groups. For instance, to remove the between-group equality constraint on the path coefficient of the equation for lamina thickness, then you would modify it to read: `lamina.thickness ~ lightXnutrient/percent.water.content`, and so on. Here is the model object that removes the between-group constraints on all of the intercepts and path coefficients:

```
my.mod<-
list(gam(percent.water.content~lightXnutrients,data=meziane),
             gam(lamina.thickness~lightXnutrients+
                lightXnutrients/percent.water.content,
           data=meziane),
             gam(midvein.thickness~lightXnutrients+
                lightXnutrients/lamina.thickness,data=meziane),
             gam(specific.leaf.area~lightXnutrients+
                lightXnutrients/percent.water.content+
                lightXnutrients/lamina.thickness+
                lightXnutrients/midvein.thickness,data=meziane))
get.AIC(my.mod,MAG=my.mag,data=meziane)
```

Of course, you can change the distributional family from the default “gaussian” (normal) distribution inside the `gam()` or `gamm4()` functions by including the `family=` argument. You can include mixed model data structures by using the `gamm4()` function as long as the random structure does not include the groups that define the multigroup structure. If you want to use nonlinear smoother regression (by using the `s()` argument), then the model syntax is

different when removing between-group constraints on the path "coefficients"[220]. Instead, you use the `by=` argument; note that the variable name given to the `by` argument must be explicitly defined as a factor. For instance, if I want to allow the direct effect linking percent water content to lamina thickness to vary between the groups in a nonlinear way, then I would change the line

```
gam(lamina.thickness~lightXnutrients +
lightXnutrients/percent.water.content, data=meziane)
```

to

```
gam(lamina.thickness~lightXnutrients+
s(percent.water.content,by=as.factor(lightXnutrients)),
data=meziane)
```

This line tells the gam function to separately estimate a separate nonlinear smoother regression for each group. In this way, you can use the logic of AIC statistics to identify the model that is most likely, the models that should also be considered (if the change in the AIC value is less than ~4), and the models that you can exclude.

8.8 Multilevel and mixed model SEM

Multigroup SEM is designed to deal with violations of the assumption of causal homogeneity: when different subsets of the data are being generated by different quantitative and/or qualitative causal structures. Multilevel and mixed model SEM is designed to deal with violations of the statistical assumption of independence of the observations.

The notion of the independence of observations is not self-evident for most people. Let's say that the probability of observing some random value of X (say, X_i=1.2) is $p(X_i)$ and that the probability of observing some other random value of X (say, X_j=-0.2) is $p(X_j)$. If these two observations of X are independent then this means that the probability of observing both of these two values – $p(X_i$=1.2 & X_j=-0.2) – is equal to the product of their individual probabilities; i.e. $p(X_i,X_j)=p(X_i)p(X_j)$. This simple formula is the statistical definition of independence and is the basis for calculating degrees of freedom. Each independent observation gives one "bit" of

[220] Because in the case on nonlinear functions, there are no "coefficients" (i.e. constants) to describe the slope.

information and so N independent observations gives N "bits" of information, i.e. the total degrees of freedom. If, once we observe a value of X_i=1.2, we were absolutely certain that another observation would be X_j=-0.2, then the probability of X_i=1.2 and X_j=-0.2 would be the same as the probability of X_i=1.2; after all, if X_i=1.2 then it is a sure thing that X_j=-0.2. In mathematical notation, $p(X_i,X_j)=p(X_i)$. These two observations would only give us one "bit" of information, not two "bits" of information.

Imagine that you have randomly chosen N individual plants of a single species. For each individual plant, you have randomly chosen a single leaf and have measured the projected surface area (X) of this leaf. Let's say that the surface area of the single leaf of individual i, Xi, is 23.2 cm^2 and so the probability of observing such a value is $p(X_i=23.2)$. Now we look at the single leaf of a new individual j and see that its surface area is 19.8 and so the probability of observing this new value is $p(X_j=19.8)$. If these two observations are independent then, by definition, the joint probability of observing these two values in the same data set, $p(X_i=23.2, X_j=19.8)$, is equal to $p(X_i=23.2)p(X_j=19.8)$.

Now, we choose two leaves from each randomly chosen plant and so have 2N observations in our data set. Imagine further that these plants are quite unusual in that every leaf of a given plant is exactly the same size. In that case, once we have chosen plant i and observed a leaf with a surface area of 23.2 cm^2, then we know that there is a second leaf in our data set that is also exactly 23.2 cm^2. The value of the second leaf of each plant is perfectly correlated with the value of the first leaf of each plant. Even though we have 2N observations in our data set, we still only have N independent "bits" of information, and we still only have N, not 2N, total degrees of freedom. What happens if the surface areas of the leaves on the same plant are not identical, but are more similar to each other than are the surface areas of the leaves of the different plants? The values of the surface areas of the two leaves on the same plant are still not independent, since knowing the value of the first leaf will give us some additional information about the likely surface area of the second leaf. However, since the two leaves on the same plant are not identical, we still have some new information when we observe the second leaf. Now, we have more than N "bits" of information (total degrees of freedom), but still less than 2N "bits" of information (total degrees of freedom). If the correlation between the surface areas of the two leaves on each plant is strong then we will have closer to N total degrees of freedom and if the

correlation is weak then we will have closer to 2N total degrees of freedom. In the limit, if the correlation is zero, then the leaves on the same plant resemble each other no more that do the leaves on different plants; all of the observations are independent of each other, and we have 2N total degrees of freedom. If we do not take such partial dependencies in our data into account, then the degrees of freedom will be wrong. Since variances, covariances, means and path coefficients are functions of the degrees of freedom assigned to them, these will also be wrong.

Multilevel and mixed-model SEM is designed to account for partial dependence between observations in a data set that are due to these observations having some nested (or partially nested) structure. Such nesting is common in biological data. In the fictional example above, we expect that the trait values of leaves of the same plant will be somewhat correlated because these leaves have the same genotype and have developed in similar environmental conditions. If we sample a single leaf from each individual but measure it at different times during the growing season (repeated measures sampling), then we expect the same thing. If we sample several individuals from different species, then we expect the traits of individuals of the same species to be somewhat correlated for the same reason. If we sample individuals of the same species in different populations, then we expect the same thing.

I referred above to observations having some nested, or *partially* nested, structure. A data set is completely nested if each lower-level unit can only belong to a single higher-level unit. For instance, a single leaf can only belong to a single plant and a single plant can only belong to a single species. A data set is partially nested, or "cross-classified" if this is not strictly true. For instance, if you sample several individuals from different sites and from different species, then each individual can only belong to a single site (if all of your samples are taken at the same time) and to a single species, but the same species can be found in different sites. This is a cross-classified sample. It can get even more complicated! If you sample different sites at different times, and if it is possible for the same individual to move between sites, then the same individual can also be found in more than one site.

A number of similar, but not exactly equivalent methods have been devised to deal with this problem that are variously called hierarchical, multilevel, or mixed models. Hierarchical or multilevel models assume a truly nested design and (as best as I can determine) they are equivalent names for the same thing. Mixed models include multilevel or hierarchical models as

special cases but can also handle cross-classified data. This distinction between nested and cross-classified data is important because, as you will learn, lavaan can only deal with truly nested designs (multilevel SEM) while pwSEM can deal with both (mixed model SEM).

The goal of this section is not to teach you about mixed models. Entire books are devoted to this topic (Pinheiro and Bates 2000, Hox 2002, Gelman and Hill 2007). Wood (2017) gives an overview of generalized additive mixed models in the mgcv package, available via the `gamm4()` function, that is used by the pwSEM package. The syntax used by `gamm4()`, when specifying mixed models, is the same[221] as the popular lme4 package (Bates et al. 2015). Here, I only want to give you a flavour of how these models work but you must have a basic understanding of mixed-model regression to properly follow this section.

Let's start with the simplest two-level model: a simple regression of Y on X assuming linearity and normality. We randomly choose n_1 individual plants of a single species and, for each plant, we randomly choose n_2 leaves. We measure the values of traits X and Y on each leaf. Our data set consists of $N = n_1 \cdot n_2$ observations[222]. Figure 8.6 shows these observations. If we did a separate regression for each plant then we would have n_1 different regressions of which the i^{th} regression, based on the j=1 to n_2 leaves of the i^{th} plant has the form $Y_{ij} = \alpha_i + \beta_i X_{ij} + \varepsilon_{ij}$. The intercept of this i^{th} regression is α_i, the slope is β_i and the residual value of the j^{th} leaf from this i^{th} plant is ε_{ij}; these residuals follow a normal distribution whose mean is zero and whose standard deviation is σ_i. To make things even simpler, assume that the slope linking X and Y is the same for all of our plants (β) so that $\beta_i=\beta$ but the intercepts (α_i) can be different. Now, we have a regression of the form $Y_{ij} = \alpha_i + \beta X_{ij} + \varepsilon_{ij}$. However, since we have randomly chosen our n_1 individual plants, the values of the intercepts that are associated with each of these individual plants are also random values from a normal distribution! This second normal distribution (describing the distribution of the intercepts, not the distribution of trait) will have a mean intercept (α). Let δ_i represent the difference (the residual) between the i^{th} intercept (α_i) and the

[221] Except that it additionally allows for smoother terms, via the `s()` function, which allows it to generalize to nonlinear smoother regression.
[222] In general, we could randomly choose different number of leaves for each plant, but we are simplifying things to the maximum.

mean intercept (α). Then each of these residual values (δ_i) will randomly vary around the mean value (α) with a standard deviation of $\sigma_i^{'}$. Putting this all together:

$$Y_{ij} = \alpha_i + \beta X_{ij} + e_{ij}$$
$$\alpha_i = \alpha + \delta_i$$
$$e_{ij} \sim N(0, \sigma_i)$$
$$\delta_i \sim N(0, \sigma_i^{'})$$

An entirely equivalent way of writing the regression equation, by replacing α_i by $\alpha + \delta_i$, is $Y_{ij} = \underline{\alpha + \beta X_{ij}} + \delta_i + \varepsilon_{ij}$. The first two terms (underlined) are called the "fixed" part because the intercept (α) and slope (β) are fixed at the same value for all of the individual regressions. The last two terms ($\delta_i, \varepsilon_{ij}$) are called the random part because they randomly vary from plant to plant (δ_i) and from leaf to leaf within the same plant (ε_{ij}). Together, the fixed and random parts form a "mixed" model regression[223].

[223] Since every regression, even one with only one level, contains both a fixed and random part, all statistical models are "mixed", but the qualifier "mixed" is reserved for models with more than one source of random variation.

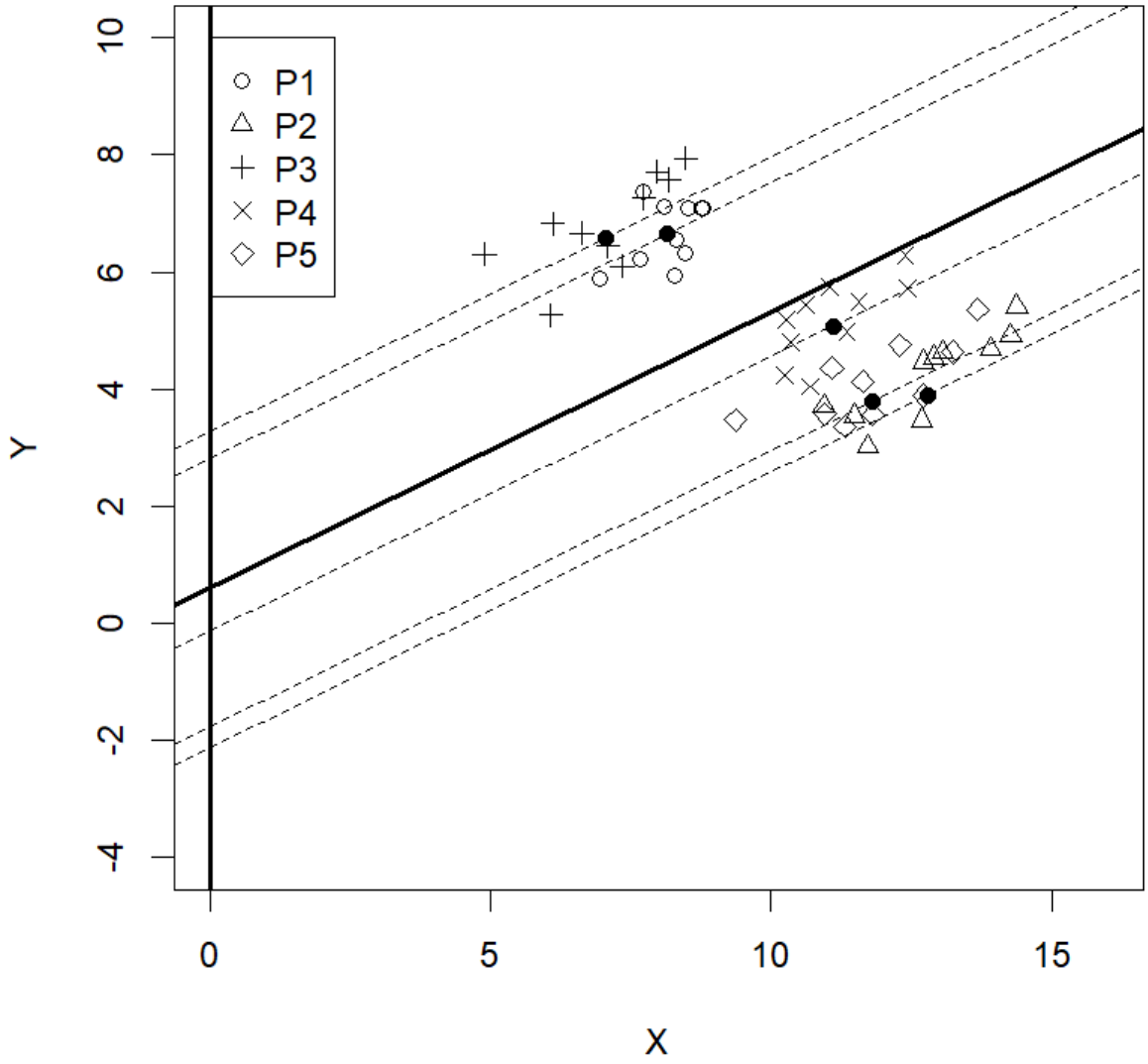


Figure 8.6. 50 (fictitious) observations of values of two traits (X, Y) taken on 10 randomly chosen leaves from five randomly chosen individual plants (P1 to P5). The broken lines show the relationship between X and Y within each individual plant. The solid line shows the relationship between X and Y within an "average" plant. The solid points show the mean values of X and of the predicted values of Y for each individual plant. Notice that the intercepts tend to get lower as the average value of X increases.

If you fit a mixed model regression to the data in Figure 8.6 then you get the following results:

```
Random effects:
 Groups   Name        Variance Std.Dev.
```

```
 ind      (Intercept) 5.9283   2.4348
 Residual             0.2512   0.5012
Number of obs: 50, groups:  ind, 5

Fixed effects:
              Estimate Std. Error t value
X.0(Intercept)  0.60175    1.31763   0.457
X.0X            0.47103    0.07249   6.498

Correlation of Fixed Effects:
     X.0(I)
X.0X -0.561
```

The first part of the output ("Random effects") gives the estimated variances and standard deviations of the two sources of random variation. The first source is the variation in the intercepts of the regressions for each individual (Groups=ind and Name=(Intercept)). If you look at Figure 8.6 and extend each of the separate regressions to the vertical line at X=0, then you can see the five different intercepts. The variation of these intercepts is 5.93. The second source is the variation of each leaf from the predicted value of the regression specific to the individual plant to which it belongs (Groups=Residual). By assumption[224], the residuals of all of these individual regressions have the same variation, and it is 0.25. Notice that the variation of the intercepts of each plant is almost 24 times[225] greater than the variation of individual leaves within the same plant. The second part ("Fixed effects") gives the relationship between X and Y of the average leaf in the average individual: Y_i=0.60+0.47X. This is the thick solid regression line in Figure 8.6.

The purpose of showing you Figure 8.6, and the resulting output, is simply to illustrate what would happen if we ignored the nested structure of these data. The overall pattern between X and Y in Figure 8.6 is negative. If we fit[226] a simple regression of Y on X, the slope is -0.34. Yet we clearly see that the relationship between X and Y for each individual plant is positive with a slope of 0.47. The overall negative pattern arises because those individuals who have the largest values of X on average (the solid black dots in Figure 8.6) have the lowest values of Y on average and so tend to have the lowest intercepts (where the individual regressions cross X=0). Not only would we completely misunderstand the relationship between X and Y but, if we used

[224] Depending on the R package that you use, this assumption can be modified.
[225] 5.93/0.25
[226] `summary(lm(Y~X, data=dat))`

the residuals from the simple (non-mixed) regression when conditioning during d-separation, we would get the wrong answers concerning conditional independence as well.

This is perhaps the simplest non-trivial mixed model regression. A slightly more complicated model would allow the slopes (β_i) to also randomly vary across individuals. The complications beyond that could include more levels of random variation (perhaps, individuals varying across randomly chosen species), cross-classification of observations (perhaps individuals in different populations, in which individuals that can belong to more than one population) more predictor variables than simply X_{ij}, error distributions that are not normally distributed (thus, generalized linear mixed models), and finally relationships between predictor and response variables that are nonlinear smoother functions (thus generalized additive mixed models). I will not talk about any of these complications in this book, but you can learn about these from the references that I gave previously.

8.9 Multilevel and mixed model piecewise SEM

If you understood the logic and practice of the d-separation test that was presented in Chapter 3, its extension to include implicit latent variables that was presented in Chapter 6, and how to fit and evaluate mixed model regression, then you already know everything that you need to use mixed model piecewise SEM. Whenever you encounter a d-separation claim like X_||_Y|**Z**, and if either variables X and/or Y involve data that have a nested or cross-classified structure, then you simply use the appropriate mixed models when obtaining the residuals of X given **Z** or the residuals of Y given **Z**. Of course, choosing an *appropriate* mixed model requires that you understand how to fit and evaluate mixed models, including knowing when it is appropriate to include intercepts and slopes in the random part of the model. These details can quickly become complicated, and I will not deal with this topic.

Because working with mixed models involving both random intercepts and slopes often requires evaluating different combinations of random slopes and intercepts, the pwSEM package does not allow for random slopes. If you do require random slopes, then you must go through the steps of a dsep test (Chapter 3) yourself while carefully deciding which slopes should randomly vary

between levels. However, it is often sufficient to allow only intercepts to randomly vary between levels and the pwSEM package can easily accommodate random intercepts. Since the pwSEM package is built using the mgcv package (Wood 2017) which includes the `gamm4()` function (generalized additive mixed models), you can fit models involving both nested and cross-classified data structures, linear or nonlinear smoother functions (via the `s()` function), and normal or generalized exponential error distributions (binomial, Poisson, Gamma etc.). You simply call the `gamm4()` function, rather than the `gam()` function, when constructing your model object that is passed to the `pwSEM()` function. The following are the most important arguments of the `gamm4()` function that can be using with `pwSEM()`.

formula=: this is simply a gam formula for the fixed part of the mixed model, which is like the formula for a glm except that smooth terms can be included using the `s()` function. Examples are Y~X+Z, Y~s(X)+Z, Y~as.factor(X)+Z

random=: this is the formula specifying the random effects of the mixed model. It follows the popular lmer style (Bates et al. 2015). Examples are ~(1|species) or~(1|species)+(1|sites) or ~(1|site/species) which specifies random intercepts across species and random intercepts across both species and sites; in the latter case this will be cross classified if you have the same name for the variable "species"

family=: the name of the distributional family to be used (binomial, gaussian, Gamma, poisson)

data=: a data frame containing the data, including the variables defining the nesting structure of the data.

As an example, let's look again at the Blue Tits example that I presented in Chapter 3. In those data, 125 nests were monitored in eight different years (1994, 1996, 1998,1999, 2000, 2001, 2002 and 2004) and up to 10 chicks were found in a single nest in a single year. In the data set, the numeric variable "nest" contains numbers assigned to each nest to identify it and the numeric variable "year" contains the year number. Another numeric variable, "ind", contains unique numbers assigned to each chick. Each of these three variables could equally be a character variable. The remaining variables contain the measured trait values of each chick. These are all numeric, but the variable "recruited", which only contains values of 0 or 1 and indicates if the chick had been successfully recruited to the population, could also have been encoded as

characters, for instance “yes” and “no”, since it is binary. Here is an example of the entries for nest number 1 in the years 1994 and 1996:

```
Year nest     ind recruited mass hemato protos   frass dateweighted
1994    1 4087518         0  8.6   51.5      0 0.03460          112
1994    1 4087519         0  9.2   52.0      0 0.03460          112
1994    1 4087520         0  8.8   51.2      0 0.03460          112
1994    1 4087521         0  9.0   55.5      0 0.03460          112
1994    1 4087522         0  8.3   52.5      0 0.03460          112
1996    1 4151118         1 11.3   42.0      0 0.09075          108
```

Notice that the value of “frass” is identical for all five lines taken in the year 1994. This is because this variable (an indication of caterpillar abundance around that nest in that year) is the same from all of the chicks in that nest in that year. The variable “frass” varies between nests and between years, but not between chicks in the same nest and year. However, the value of “mass” differed for each chick in that nest in that year. There were five chicks in nest 1 in 1994 but only one chick in 1996. Since there can be more than one individual chick within a single nest, the variable “ind” is nested within the variable “nest”. Since there can be more than one individual chick in a single year, the variable “ind” is also nested within the variable “year”. However, since the same nest occurs in more than one year, and since the same year will have more than one nest, neither “year” nor “nest” are nested within the other. This is a case of cross-classification because the value nest=1 occurs in more than one year and the value year=1994 occurs in more than one nest. The `gamm4()` function will automatically detect these patterns of dependence. For instance, imagine[227] that different nests were monitored each year so that the same number for “nest” did not occur in more than one year. In that case, `gamm4()` would automatically recognise that “nest” was nested within “year”. Here is the model object that I created in Chapter 3:

```
my.list<-
list(gamm4::gamm4(formula=protos~1,random=~(1|nest)+(1|year),fam
ily="poisson",data=BlueTits),

gamm4::gamm4(formula=frass~1,random=~(1|nest)+(1|year),family="g
aussian",data=BlueTits),

gamm4::gamm4(formula=mass~protos+frass,random=~(1|nest)+(1|year)
,family="gaussian",data=BlueTits),
```

[227] In that case, the value (1) of “nest” in the year 1994 would be different from the value of “nest” in the year 1996

```
gamm4::gamm4(formula=hemato~protos+frass+mass,random=~(1|nest)+(
1|year),family="gaussian",data=BlueTits),

gamm4::gamm4(formula=recruited~hemato,family="binomial",random=~
(1|nest)+(1|year),data=BlueTits))
```

The first call to `gamm4()` states that the fixed part is "protos" regressed against its intercept (`formula=protos~1`), that the intercepts[228] randomly vary between nests and between years (`random=~(1|nest)+(1|year)`) and that "protos" followed a Poisson distribution (`family="poisson"`).

To fit the mixed model piecewise SEM, you simply call the `pwSEM()` function:

```
fit<-pwSEM(sem.functions=my.list,data=BlueTits,
           use.permutations=FALSE,do.smooth=FALSE,
           all.grouping.vars=c("nest","year"))
```

Notice that the `pwSEM()` function has a new argument: `all.grouping.vars = c("nest","year")`. This argument is needed when fitting mixed model piecewise SEM, even if not all variables have a nesting structure, and it lists the variables in the data set that specify the nesting structure of the data. The values must be identical to the values in the `random=` argument of the `gamm4()` call in the model object.

8.10 Multilevel covariance-based SEM

Multilevel (or hierarchical) SEM in the covariance-based framework is a topic of active research but is rather limited in lavaan[229]. However, you can perform a two-level SEM with random intercepts in lavaan. A two-level SEM is one that has only two levels; examples would be many

[228] In this case, the means, since the intercept is the mean value.

[229] I am told that the commercial SEM package MPLUS allows for random slopes as well as intercepts, but I can't vouch for this since I don't own this package, and I don't want to sell any commercial software! There is also an add-one package for lavaan, called lavaan.survey, which interfaces with an R package called survey, that can do multilevel SEM among other things. This is a more advanced topic that I won't discuss in this book.

different individuals with repeated measurements on each individual, or many different species with several individuals in each species.

There are two basic approaches to estimating multilevel covariance-based SEM. The (relatively) older approach is to treat the problem as a two-group SEM (between-groups and within groups) like I have described in the first part of this chapter. The lower within-group level consists of the values of each variable after centring each around its group mean. The group means are treated as latent variables that are each modelled as direct causes of the (centred) within-group variables. The path coefficients of links are fixed at a scaling factor which is the square root of the number of observations per group, or an approximation due to Muthén (1994b, 1994a) when the number of observations per group varies across groups. I explained how to do this in the first two editions of this book since lavaan did not then include a multilevel option. Now, it is possible to fit a multilevel (2-level) SEM with random intercepts in lavaan. However, lavaan uses a different approach to fitting such two-level models from the one sketched above and described in the first two editions of this book. The approach that lavaan now uses is called a "complex survey data" approach by adjusting standard errors using a cluster-robust sandwich estimator (Hox et al. 2017).

To illustrate a two-level SEM using lavaan, I will use the `Demo.twolevel` data set that is included in the lavaan package. Since these data are simulated values, not real empirical data, I will give the observed variables names that simulate a study in which a behavioural ecologist is studying the "boldness" personality trait behaviour in Chipmunks that was discussed in Chapter 7. Imagine that our behavioural ecologist has sampled 200 individual Chipmunks and has trapped each individual either 5, 10, 15 or 20 times during their life. This is an example of repeated measures and so has a two-level nested (multilevel) data structure. Each time an individual is captured, it is placed into the experimental box that was described in Chapter 7 and three behavioural traits are measured: the time until the individual begins to move (y1), the time until the individual approaches an unfamiliar object (y2) and the flight initiation distance (y3) when a model predator is introduced. The number of repeats of a gene coding for cortisol production (w1), and the birth order of each individual (w2) are measured for each individual; the values of these two variables vary between individuals but are constant for each of the repeated measurements. Finally, three additional variables are measured on each individual at

each capture: the number of predator incursions into its territory in the last 24 hours (x1), the number of other Chipmunks that have entered its territory in the last 24 hours (x2) and its age (x3). The behavioural ecologist hypothesizes that its genetic disposition to produce cortisol (w1) and its birth order (w2) are direct causes of its latent "boldness" (fb), which it turn causes the three behavioural traits (y1, y2 and y3). However, these three behavioural traits are also caused by (and measure) the latent short-term stress (fw) experienced during the last 24 hours, which is caused by (and is measured by the three short-term experiences (x1, x2 and x3). Figure 8.7 shows the causal graph.

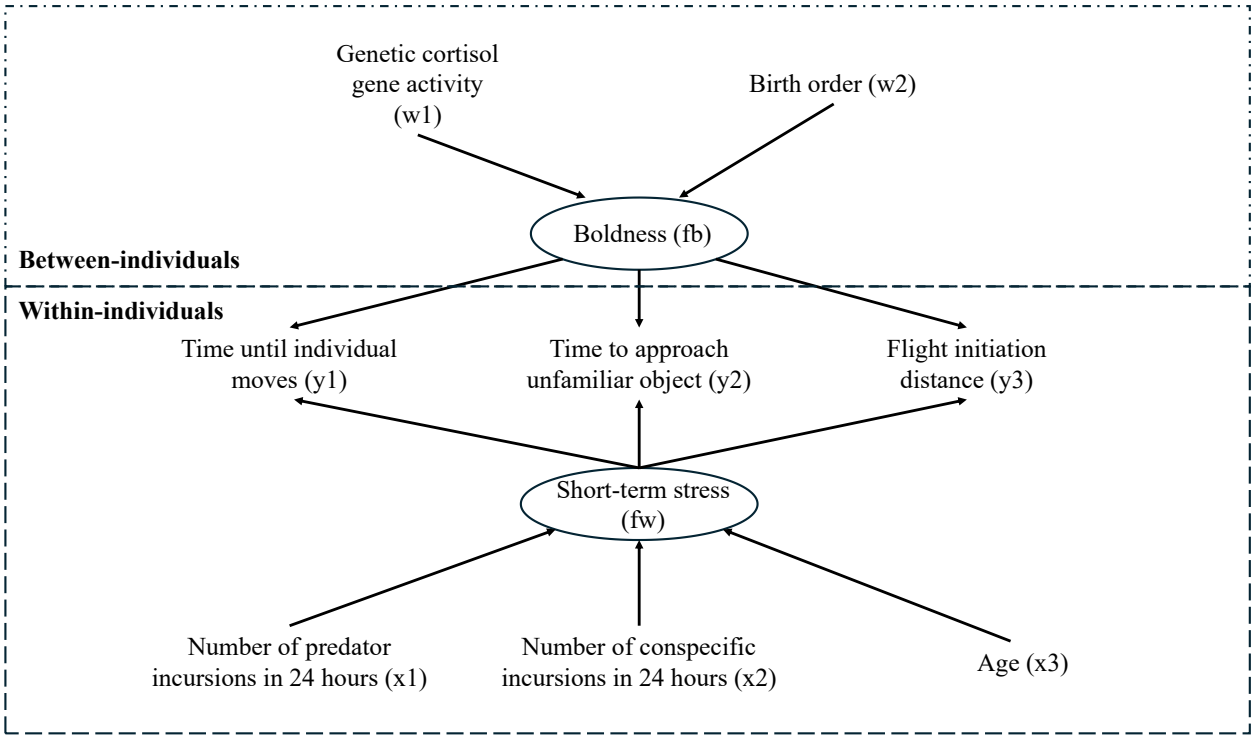


Figure 8.7. The hypothesized causal graph relating latent "boldness" in Chipmunks.

To fit such a multilevel SEM in lavaan, you must include the "`level: 1`" (i.e. the within-individual level) and "`level: 2`" (i.e. the between-individual level) lines in your model object. Following each of these lines is the code specifying the hypothesised causal structure at that level. Here is the model object:

```
my.mod<-"
level: 1
  fw=~y1+y2+y3
  fw~x1+x2+x3
```

```
level: 2
  fb=~y1+y2+y3
  fb~w1+x2
"
```

The data frame holding the data must have a variable specifying to which level 2 (i.e. which individual) each observation belongs. This variable is called "cluster" in the `Demo.twolevel` data frame. The values of the cluster variable in the first six lines of `Demo.twolevel$cluster` are:

```
> Demo.twolevel$cluster[1:6]
[1] 1 1 1 1 1 2
```

In other words, the first five lines are the results of the same individual Chipmunk that has been captured and measured at five different times.

You fit the multilevel SEM by including the `cluster=` argument in the `sem()` function (you can also include any of the other arguments that you have already learnt).

```
fit<-sem(model=my.mod,data=Demo.twolevel,cluster="cluster")
```

Typing summary(fit) will give the typical summary output.

9

Exploratory structural equations modelling

9.1 Hypothesis generation

Modern statistics is mostly concerned with *testing* hypotheses, not *developing* them. Such a bureaucratic approach views science as a compartmentalised activity in which hypotheses are constructed by one group, data are collected by another group and then the statistician confronts the hypothesis with the data. Since this book is a user's guide to causal modelling, such a compartmentalised approach will not do. One of the main challenges faced by the practising biologist is not in testing causal hypotheses but in developing causal hypotheses that are worth testing.

The philosophy of science mostly deals with questions like: How can we know if a scientific hypothesis is true or not? What demarcates a scientific hypothesis from a non-scientific hypothesis? For most philosophers of science, just as with most modern statisticians, the question of how one looks for a useful scientific hypothesis is someone else's problem. For instance, on page 32 of Popper's (1980) influential *Logic of Scientific Discovery* he says that "…there is no such thing as a logical method of having new ideas, or a logical reconstruction of this process. My view may be expressed by saying that every discovery contains 'an irrational element', or 'a creative intuition'…". Later, he says that "[scientific laws] can only be reached by intuition, based on something like an intellectual love of the objects of experience." Again, one gets the impression that science consists to two hermetically sealed compartments. One compartment, labelled "*hypothesis generation"*, consists of an irrational fog of thoughts and ideas, devoid of method, out of which a few gifted people are able to extract brilliant insights. The other compartment, labelled "*hypothesis testing*", is the public face of science. Here, one

finds method and logic, in which established rules govern how observations are to be taken, statistically manipulated, and interpreted.

At a purely analytic level there is much to be gained by taking this schizophrenic view of the scientific process. After all, how a scientific idea is developed is irrelevant to its truth. The history of science documents many important ideas whose genesis was bizarre[230]. Archimedes reportedly discovered the laws of hydrostatics after jumping into a bathtub full of water. Kukulé discovered the ring structure of benzene after falling asleep before a fire and dreaming of snakes biting their tails. These curious stories are entertaining, but we remember them only because the laws of hydrostatics hold, and benzene really does have a ring structure. As a public activity, science is interested in the result of the creation, not in the creative act itself.

The day-to-day world of the biologist does not exist at such a purely analytic level. Although it is possible to conceptually divide science into distinct hypothesis-generating and hypothesis-testing phases, the two are often intimately intertwined in practice. When the two are not intertwined the science can even suffer. Peters (1991), in his "*A Critique For Ecology*", points out that because empirical and theoretical ecology are often done by different people, the result is that much ecological theory is crafted in such a way that it cannot be tested in practice and much of field ecology cannot be generalised because it is not placed into a proper theoretical perspective. In this context I like the citation, attributed to W. H. George, given at the beginning of Beveridge's (1957) *The Art of Scientific Investigation*: "Scientific research is not itself a science; it is still an art or craft". Unlike the assembly-line worker who receives a partly finished object, adds to it, and then passes it along to someone else, the craftsman must construct the object from start to finish. In the same way, the craft of causal modelling consists as much of the generation of useful hypotheses as of their testing. Certainly, hypothesis generation is more art than method, and hypothesis testing is more method than art, but this does not mean that we must relegate hypothesis generation to a mystical world of creative intuition in which there are no rules. As you use the methods in this book to conduct research and to learn about the natural world, you will constantly move between the testing of causal hypotheses, their rejection, and then the search for new or modified causal hypotheses. There is nothing wrong, and much that is

[230] The appendix of *The Art of Scientific Investigation* (Beveridge 1957) lists 19 cases in which the origin of important scientific ideas arose from bizarre or haphazard situations. In fact, Beveridge devotes an entire chapter dealing with the importance of chance in scientific discovery.

right, about bouncing between hypothesis generation and testing. There is nothing wrong, and much that is right, about bouncing between exploratory and confirmatory statistical analysis, with one important proviso: you must be honest with yourself and with your audience about when you switch between the two. The purpose of this chapter is to introduce some exploratory methods in structural equation modelling in order to generate or modify causal hypotheses.

9.2 Exploring hypothesis space

How does one go about developing promising hypotheses concerning causal processes? To place the problem in context, imagine that you have collected data on N variables and at least some of these variables are not amenable to controlled randomised experiments. Why you suspect that these N variables possess interesting or important causal relationships may well be due to the irrational creative intuition to which Popper refers, but you are still left with the problem of forming a multivariate hypothesis specifying the causal connections linking these variables.

To simplify things, let's assume that all of the data are generated by the same unknown causal process (i.e. causal homogeneity), that there are no latent variables responsible for some observed associations (i.e. causal sufficiency) and that the data are faithful[231] to the causal process. How many different causal graphs could exist under these conditions? Each pair of variables (X and Y) can have one of three different causal relationships: either X directly causes Y (X→Y), Y directly causes X (X←Y), or the two have no direct causal links (X Y). We now must count up the number of different pairs of variables, which is just the number of combinations of 2 objects out of N. The combinatorial formula is therefore $3^{\frac{N!}{2!(N-2)!}}$. Table 9.1 gives the number of different potential causal graphs of this type that can exist given N variables.

Table 9.1. The number of different cyclic causal graphs without latent variables that can be constructed give N variables

[231] See Chapter 2 for the definition of faithfulness. In fact, much of this chapter makes use of notions introduced in Chapter 2, and the reader might want to re-read that chapter before continuing.

N	Number of graphs
2	3
3	27
4	729
5	59,049
6	14,348,907
7	10,460,353,203

If we think of the full set of potential causal graphs having N variables as forming an "hypothesis space", and your research program as a search through this space to find the correct causal graph, then Table 9.1 is bad news. Even if we could test one potential graph per second it would take us over 166 days to test every potential graph containing only six variables! If we add just one more variable, then we would be dead long before we even explored a tiny fraction of the hypothesis space. If we were to restrict our problem to acyclic graphs that the numbers would be smaller, but still astronomical (Glymour et al. 1987). On the other hand, if we accept the likelihood that we have not measured all of the causally relevant variables (i.e. if there are latent variables), then the size of the hypothesis space explodes to unbelievable sizes. If it is true that the process of hypothesis generation (in this case, proposing one casual graph out of all those in the hypothesis space) is pure intuition, devoid of method, then it is a wonder that science has made any progress at all. That science *has* made progress shows that efficient methods of hypothesis generation, although perhaps largely unstated, do exist.

So how should we go about efficiently exploring this hypothesis space? To go back to my previous question: How does one go about generating promising hypotheses concerning causal processes? One way would be to choose a graph at random and then collect data to test it. With six variables there is a bit less than one chance in 14 million of hitting on the correct structure. There is nothing logically wrong with such a search strategy; we will have proposed a falsifiable hypothesis and tested it. However, no thinking person would ever attempt such a search strategy because it is incredibly inefficient. We need search strategies that have a good chance of quickly finding those regions of hypothesis space that are likely to contain the correct answer. What would be our chances of hitting on the correct structure if we were to appeal only to "pre-

existing theory"? Clearly that would depend on the quality of the pre-existing theory. If you think that your pre-existing theory is mostly correct but that it has to be slightly modified (since your model, based on this theory, has been rejected), then Section 9.3 might be of some use. Section 9.3 discusses some exploratory methods that can be used if you think that your existing (falsified) causal hypothesis is mostly correct but only needs some minor modifications. I caution you that we are all susceptible to confirmation bias, especially if the causal hypothesis that you think is "mostly right" is one that you developed yourself! I urge you therefore to use the methods in section 9.3 sceptically and always ask the question: "does this make biological sense"?

What if you want to reconsider your understanding of a causal system without being constrained by your pre-existing theory? Sometimes biologists find themselves in the awkward position of straddling the "hypothesis generation" and "hypothesis testing" compartments. Often, we have some background knowledge that excludes certain causal relationships and suggests others, but not enough firmly established background knowledge to specify the full causal structure without ambiguity. In such situations the goal is not to test a pre-existing theory, but rather to develop a more complete causal hypothesis that would be worth testing with independent data. We need search strategies that can be proven to be efficient at exploring hypothesis space, at least given explicitly stated assumptions. Until very recently such search strategies, which are described in this chapter after section 9.3, did not exist. You will see that these search strategies rely heavily on the notion of d-separation and on how this notion allows a translation from causal graphs to probability distributions.

9.3 Modifying a pre-existing causal model

Imagine that you have tested a causal model that you thought was correct but that has been rejected by the data. You are confident that most of the direct effects in the model (the X→Y links) are correct but suspect that a few are wrong. Which ones? As an illustration, Figure 9.1a shows the true generating process for the empirical data while Figure 9.1b shows your rejected model. This rejected model is correct about most of the causal claims. It correctly states that X1 is a direct cause of X2. It correctly states that X1 is an indirect cause of X3, X4 and X5 through

the joint effect on X2. It correctly states that neither X3 nor X4 are causes of the other, but both are common direct effects of X2. However, it mistakenly claims that neither X3 nor X4 are direct causes of X5 and mistakenly claims that X2 is a direct cause of X5. Clearly, if we were to add a direct effect of X3 on X5 (X3→X5) to our rejected model, then the fit would improve substantially, meaning that the maximum likelihood chi-squared statistic would decrease substantially. Note that the original incorrect model has implicitly fixed the path coefficient from X3 to X5 at zero (remember that a missing arrow in the causal graph is equivalent to adding this missing arrow but fixing its path coefficient to zero). Adding the arrow X3→X5 to the structural equation means changing its fixed zero path coefficient to a free parameter. Similarly, if we were to add a direct effect of X4 on X5 (X4→X5), then the fit would again improve substantially, and the maximum likelihood chi-squared statistic would again decrease substantially.

Figure 9.1. The true generating process (a) and the incorrect causal graph (b).

The lavaan package has a function that does what I have just described. It identifies each fixed parameter in a specified covariance-based SEM and then calculates by how much the maximum-likelihood chi-squared statistic would decrease if that single fixed parameter was freed. This is essentially a set of nested models in which each reference model is your original rejected model, and each nested model is your original model plus this one extra free parameter. The change in the maximum likelihood chi-squared statistic between these two models is called the "modification index" and (as you know from Chapter 8) this change in the chi-squared statistic, with just one extra free parameter, has 1 degree of freedom. Be clear about what you are now doing! You are now exploring the data to modify your rejected model, not testing a preconceived hypothesis, so you should not be too hung up on null probabilities but remember that a change in the chi-squared statistic (i.e. the modification index) would have to be at least ~4 to reach

significance at the 5% level. Here is the basic call to the modification index function: `modindices(object, sort.=TRUE)`. The first argument (`object`) is the fitted object returned from the `sem()` function that holds the rejected model. The second argument (`sort.=TRUE`) ensures that the modification indices for each newly freed parameter are sorted in decreasing order. This second argument is optional, but I recommend that you include it since we are only interested in the freed parameters that have the highest modification indices. One final point: this method of exploring the data and of modifying a rejected model has a statistical justification – the notion of a nested model – but it does not have any provably correct causal justification since it is not based on d-separation (Chapter 2). It is easy to create cases in which this method will lead to better fitting models (in terms of the chi-squared statistic) that are very wrong in terms of identifying the true underlying generating structure of the data. This is especially true if the original rejected model contains several causal errors. The objective is to provide you with suggestions that you must always evaluate in light of your biological knowledge. Never blindly follow the method that I will describe next.

At this point, let's fit data to the incorrect model in Figure 9.1b, using 500 lines of data, contained in the data frame `dat`, that were generated from the true generating process shown in Figure 9.1a.

```
wrong.mod1<-"
X2~X1
X3~X2
X4~X2
X5~X2
X3~~0*X5
X4~~0*X5
X3~~0*X4"
wrong.fit<-sem(wrong.mod1,data=dat)
summary(wrong.fit)
```

Notice that I had to explicitly fix the covariances between the terminal endogenous variables (X3, X4 and X5) to zero since, by default, the `sem()` function allows these to be free by default. As expected, the model is strongly rejected since the chi-squared statistic is 316.148, with 6 degrees of freedom and with a null probability of less than 0.0005. Here is how to use the `modindices()` function; I also show the first 8 out of 21 lines of output:

```
modindices(wrong.fit,sort.=TRUE)
    lhs op rhs      mi    epc sepc.lv sepc.all sepc.nox
25  X5  ~  X3 130.283  0.530   0.530    0.515    0.515
5   X3 ~~  X5 130.283  0.437   0.437    0.510    0.510
20  X3  ~  X5 130.283  0.491   0.491    0.506    0.506
26  X5  ~  X4 104.707  0.483   0.483    0.462    0.462
6   X4 ~~  X5 104.707  0.385   0.385    0.458    0.458
23  X4  ~  X5 104.707  0.434   0.434    0.454    0.454
17  X2  ~  X4   1.331  0.090   0.090    0.095    0.095
14  X2 ~~  X4   1.331  0.072   0.072    0.098    0.098
```

The first three columns (`lhs op rhs`) give the left-hand side, the operator and the right-hand side of the newly freed parameter. The first line therefore refers to a new model in which the path coefficient from X3 to X5 (X3→X5, i.e. X5~X3) is freed and is no longer fixed to zero[232]. The fourth column (`mi`) gives the modification index. This means that adding X3→X5 to our rejected model would decrease the chi-squared statistic by 130.284. Note that the modification indices for adding a free covariance between X3 and X5 (X3~~X5) and for making X5 a cause of X3 (X3~X5) are equal to the first modification index (because they are d-separation equivalent) and so we have to make a non-statistical choice here. The fifth column (epc) gives the expected parameter change; that is by how much the value of the newly freed parameter (the path coefficient from X3 to X5) would change from its current value of zero. The final three columns give the same information on the expected parameter change based on standardized values for any latents in the model (`sepc.lv`), based on standardized values for all variables in the model (`sepc.all`) or based on standardized values of all non-exogenous variables (`sepc.nox`).

The various proposed changes, in terms of freeing fixed parameters, that are listed by `modindices()` are not independent of one another. As soon as you allow one previously fixed parameter to be free, this can change the values of the other proposed changes. That means that you should only make one change at a time, refit this nested model and, if needed, use the `modindices()` function again on the nested model to see what other modifications might be considered. Here is the model with the suggested change (X5~X3) added, as well as the first 8 lines of the output of the `modindices()` function.

```
#Modify to allow X5~X3
wrong.mod2<-"
```

[232] remember that in our original rejected model, there was no direct effect from X3 to X5 and so the path coefficient from X3 to X5 was fixed to zero

```
X2~X1
X3~X2
X4~X2
X5~X2+X4
X3~~0*X5
#X4~~0*X5
X3~~0*X4"
wrong.fit2<-sem(wrong.mod2,data=dat)
modindices(wrong.fit2,sort.=TRUE)
   lhs op rhs      mi    epc sepc.lv sepc.all sepc.nox
6   X3 ~~  X5 161.080  0.432   0.432    0.568    0.568
26  X5  ~  X3 161.080  0.524   0.524    0.509    0.509
21  X3  ~  X5 130.283  0.491   0.491    0.506    0.506
18  X2  ~  X4   1.331  0.090   0.090    0.095    0.095
14  X2 ~~  X4   1.331  0.072   0.072    0.098    0.098
30  X1  ~  X4   1.331 -0.066  -0.066   -0.069   -0.069
25  X4  ~  X1   1.331 -0.056  -0.056   -0.053   -0.055
17  X2  ~  X3   1.143  0.082   0.082    0.088    0.088
13  X2 ~~  X3   1.143  0.068   0.068    0.091    0.091
```

The model `wrong.fit2` had a chi-squared statistic of 198.658 (5 df, $p<0.0005$) and so, although it was a big improvement over the original rejected model, it is still rejected. The highest modification indices (161.08) suggest that we either add an arrow from X3 to X5 (X5~X3), add an arrow from X5 to X3 (X3~X5), or else add a free covariance between them (X3~~X5). I will add an arrow from X3 to X5:

```
#Modify to allow X5~X3
mod3<-"
X2~X1
X3~X2
X4~X2
X5~X2+X4+X3
#X3~~0*X5
#X4~~0*X5
X3~~0*X4"
fit<-sem(mod3,data=dat)
summary(fit)
modindices(wrong.fit,sort.=TRUE)
   lhs op rhs    mi    epc sepc.lv sepc.all sepc.nox
15  X2 ~~  X5 1.658 -0.062  -0.062   -0.109   -0.109
27  X5  ~  X1 1.658  0.048   0.048    0.044    0.046
19  X2  ~  X4 1.331  0.090   0.090    0.095    0.095
14  X2 ~~  X4 1.331  0.072   0.072    0.098    0.098
26  X4  ~  X1 1.331 -0.056  -0.056   -0.053   -0.055
30  X1  ~  X4 1.331 -0.066  -0.066   -0.069   -0.069
29  X1  ~  X3 1.143 -0.060  -0.060   -0.064   -0.064
23  X3  ~  X1 1.143 -0.053  -0.053   -0.049   -0.051
```

This third model (`mod3`) has a chi-squared statistic of 4.198 (4 df, p=0.38). The remaining modification indices are all 1.658 or less (i.e. less than ~4) and so including any of these would not improve this model significantly. At this point, we have almost completely recreated the true generating structure (Figure 9.1a) except that this model incorrectly includes a direct effect from X2 to X5. This suggested change will not appear in the output of `modindices()` since it involves fixing a free parameter (X5~X2) to zero (i.e. removing the arrow from X2 to X5), not freeing a fixed parameter. However, when we look at the output of the summary object, i.e. `summary(fit)`, we see that the estimated path coefficient linking X2 to X5 is only 0.004 and the null probability of this path coefficient being equal to zero is 0.930. Fixing this path coefficient to zero recovers exactly the true causal structure generating the data.

If you have been carefully reading my description of the series of modifications of the original rejected model, you will realize that I have been cheating! At each step, whenever `modindices()` made suggestions for modification, I chose the correct change because I knew the true causal structure. For instance, the first application of `modindices()` to the original rejected model in Figure 9.1a suggested 6 possible changes that would have each significantly improved the model (those with modification indices of greater than ~4). If I had incorrectly chosen the proposed change of adding a free covariance between X3 and X5 (X3~~X5) and then rerun the `modindices()` function with this incorrect change, these are the best changes that are proposed:

```
   lhs op rhs      mi    epc sepc.lv sepc.all sepc.nox
26  X5  ~  X4 137.640  0.476   0.476    0.455    0.455
6   X4 ~~  X5 137.640  0.380   0.380    0.451    0.451
23  X4  ~  X5 104.707  0.434   0.434    0.454    0.454
19  X3  ~  X4  33.013 -0.224  -0.224   -0.221   -0.221
7   X3 ~~  X4  33.013 -0.179  -0.179   -0.221   -0.221
```

If I had then incorrectly chosen the proposed change of adding a free covariance between X4 and X5 (X4~~X5), the resulting model (Figure 9.2) would not be rejected (X^2=4.198, 4 df, p=0.38).

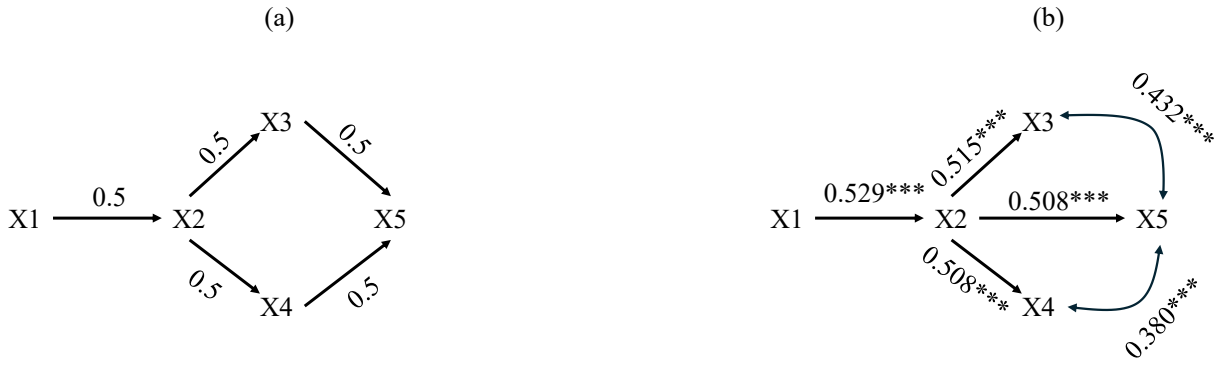


Figure 9.2. The true causal structure generating the data is shown in causal graph (a) along with the true values of the path coefficients. A model (b), derived using modification indices, is shown in causal graph (b) that is not rejected but that is wrong in terms of its causal claims, along with the estimated path coefficients and their null probabilities (*** indicates p<0.0005).

Even though this modified model is not rejected, it is even worse than the original rejected model in Figure 9.1a in terms of its causal claims. It incorrectly claims that X3 is not a cause of X5 but that both X3 and X5 are common direct effects of an unknown latent variable. It incorrectly claims that X4 is not a cause of X5 but that both X4 and X5 are common direct effects of a second unknown latent variable. Finally, it incorrectly claims that X2 is a direct cause of X5, and this incorrect path coefficient is highly significant. This reinforces my earlier message: the suggestions of `modindices()` are based only on statistical arguments with no theoretical link, based on d-separation, to the underlying causal structure of the data. You must still make non-statistical decisions about which suggestions to follow (or not), and incorrect decisions can lead to models that are wrong in terms of their causal claims.

Although there is no function in the pwSEM package to estimate such modification indices, you could calculate these yourself based on the change in the C statistic of the dsep test, since they are simply the result of a series of nested models resulting from freeing a single fixed zero parameter. A better way, besides the method described in section 9.4, is to look at the null probabilities associated with the d-separation claims:

```
library(pwSEM)
library(mgcv)
wrong.mod<-list(
  gam(X1~1,data=dat),
  gam(X2~X1,data=dat),
  gam(X3~X2,data=dat),
```

```
  gam(X4~X2,data=dat),
  gam(X5~X2,data=dat)
)
summary(pwSEM(sem.functions=wrong.mod,data=dat))
```

The resulting output gives the following:

```
Basis Set
( 1 )  X1 _||_ X3 | { X2 }
( 2 )  X1 _||_ X4 | { X2 }
( 3 )  X1 _||_ X5 | { X2 }
( 4 )  X3 _||_ X4 | { X2 }
( 5 )  X3 _||_ X5 | { X2 }
( 6 )  X4 _||_ X5 | { X2 }

Null probabilities of independence claims in basis set
(1) 0.2915921
(2) 0.2770074
(3) 0.9063379
(4) 0.7751565
(5) 0
(6) 0
```

The last two d-separation claims are clearly wrong. The first incorrect d-separation claim is that X3 is d-separated from X5, conditional on X2. Therefore, either X3 causes X5 (X3→X5), X5 causes X3 (X3←X5), or both X3 and X5 are caused by some unknown latent variable L1 (X3←L1→X5). These are the same suggestions made by `modindices()`. The second incorrect d-separation claim is that X4 is d-separated from X5, conditional on X2. Therefore, either X4 causes X5 (X4→X5), X5 causes X4 (X4←X5), or both X4 and X5 are caused by some unknown latent variable L2 (X4←L1→X5). These are again the same suggestions made by `modindices()`.

9.4 The shadow's cause re-visited

I have repeatedly compared the relationship between cause and correlation to the relationship of an object and its shadow. There is something missing in this analogy when applied to actual research projects. When we measure a correlation in a sample of data we are almost never interested in the value of the correlation in that particular sample. Rather, we use the sample

value to infer what the correlation might be in the full population. It is as if, in Nature's Shadow Play, the causal processes not only cast potentially ambiguous correlational shadows in our data, but these shadows are randomly blurred as well. We therefore have two problems. First, we have to find a way to provably deduce causal processes from correlational shadows. Second, we have to account for the inaccuracies caused by using sample correlations to infer population correlations. It is important to keep these two problems distinct. The second problem, that of dealing with sampling variation, is a typical problem of mainstream statistics. For this reason, we will first see how to go from correlations to causes when there is no sampling variation. In other words, we will consider asymptotic methods.

The history of the development of these exploratory methods, or "search" algorithms, is fascinating. The word "history" has connotations of age but, in fact, these methods date to after 1990. The mathematical relationships between graphs, d-separation and probability distributions were worked out in the mid 1980's by Judea Pearl and his students at UCLA (Pearl 1988). This was the translation device between the language of causality and the language of probability distributions that had been missing for so long. As soon as it became possible to convert causal claims into probability distributions the dam broke, and the conceptual flood came pouring out. It became immediately obvious that one could also convert statements concerning probabilistic independencies into causal claims. Pearl and his team at UCLA developed a series of algorithms to extract causal information from observational data during the period 1988-1992[233]. Interestingly, a group of people at the Philosophy department at Carnegie-Mellon University (Clark Glymour, Peter Spirtes, Richard Scheines and their students) had also been working on the same goal. In 1987 they had published a book (Glymour et al. 1987) in which zero partial correlations and "vanishing tetrad differences" were used to infer causal structure, but without the benefit of d-separation or the mathematical link between causal graphs and probability distributions. As soon as the Carnegie-Mellon group encountered Pearl's work on d-separation (they didn't know about the discovery algorithms that Pearl and his students were working on) they immediately began to independently derive and prove almost identical search algorithms. These algorithms (and much more) were proven and published in Spirtes, Glymour and Scheines (Spirtes et al. 1993) and incorporated into their TETRAD II program[234]. An algorithm called the

[233] This brief history, and the algorithms of Pearl and his students, are given in Chapter 2 of Pearl (2000).
[234] https://www.cmu.edu/dietrich/philosophy/tetrad/

Inductive Causation (IC) algorithm was proven and published by Verma and Pearl (Pearl and T.S.Verma 1991) and is very similar to the Causal Inference (CI) algorithm of the Carnegie-Mellon group. I will leave it to the people involved to sort out questions of priority. It is fair to say that once the d-separation criterion was developed, the various algorithms were "in the air" and had only to be brought down to earth by those with the requisite knowledge. The philosopher's dream of inferring (partial knowledge of) causation from observational data had been realised.

I explained, in Chapter 2, how to translate from the language of causality, with its inherently asymmetric relationships, to the language of probability distributions with its inherently symmetric relationships. The Rosetta Stone allowing this translation was the notion of d-separation. Using d-separation, we can reliably convert the causal statements expressed in a directed acyclic graph into probabilistic statements of dependence or independence that are expressed as (conditional) associations. When attempting to discover, rather than test, causal relationships, the problem is turned on its head. Now, we must start with probabilistic statements of (conditional) dependence or independence and somehow back-translate into the language of causality. As you will see, this back-translation is almost always incomplete. There is almost always more than one acyclic causal graph that implies the same set of probabilistic statements of (conditional) dependence or independence. In other words, there are almost always different acyclic causal graphs that make different causal predictions but exactly the same predictions concerning probabilistic dependence or independence. You already know this since you already know about equivalent models (section 5.6 of Chapter 5), a topic that has been recognised in SEM for a long time and generally ignored for just as long.

The methods that I describe in this section are based on the strategy of back-translation that I described above. The first step is to obtain a list of probabilistic statements of (conditional) dependence or independence involving the variables in question. This list will have statements like: "X1 is unconditionally independent from X2", "X1 is not independent from X2 when conditioned on X3", and so on. In practice[235], this list will have statements like "the probability that X1 is unconditionally independent fromX2 is 0.002" but at this stage we are assuming that are samples are so large, and our statistical power so great, that we can confidently ignore the

[235] Since we will have to rely on random samples

probability and simply claim certainty. From this list, we construct an *undirected dependency graph*[236]. An undirected dependency graph looks like a causal graph except that all of the arrows have been converted into lines (—) without arrowheads. However, the lines in the undirected dependency graph have a very different meaning. Two variables in this graph have a line between them if they are probabilistically dependent conditional on every possible subset of other variables in the graph. The lines in the undirected dependency graph express symmetric associations, not asymmetric causal relationships. Since we cannot measure associations involving variables that we have not measured, the undirected dependency graph cannot have latent variables. The next step is to convert as many of the symmetric relationships in the undirected dependency graph as possible into asymmetric causal relationships. This is called *orienting* the edges and uses the notion of d-separation. Generally, not all of the undirected lines can be converted into directed arrows and so we do not end up with a completely directed acyclic graph. Rather, we end up with a partially oriented acyclic graph. If you have forgotten about d-separation and m-separation then you should probably go back to Chapter 2 (section 2.6) and to Chapter 6 (sections 6.3 and 6.4).

Let's begin with the following assumptions:

1. For each possible association, or partial association, we definitely know if the association or partial association involving the measured variables in the data exists (i.e., is different from zero) or doesn't exist (is equal to zero). This is simply the assumption that there is no sampling variation. In a practical sense, we are assuming that we have sufficiently large sample sizes that statistical power is not an issue. Later, I will discuss what to do when this assumption doesn't hold.

2. Every unit in the population is governed by the same causal process (i.e. causal homogeneity, as defined in Chapter 8). This assumption can be relaxed if we know the actual nesting structure of the data.

3. The probability distribution of the observed variables measured on each unit is *faithful* to some (possibly unknown) cyclic[237] or acyclic causal graph. "Faithfulness" means that there

[236] Also sometimes called a skeleton graph.
[237] The subsequent orientation phases will differ depending on whether or not we assume an acyclic structure.

are no (conditional) independence relationships in the data that are not predicted by d-separation in the causal graph. Remember (Chapter 2) that d-separation always entails (conditional) independence, but (conditional) independence does not always entail d-separation since opposing causal effects along different paths can theoretically cancel each other out completely. So, we are assuming that such perfectly balanced opposing causal effects don't exist. We don't have to assume that there are no unmeasured variables generating some associations (this assumption is called *causal sufficiency*) or that the variables follow any particular probability distribution, or that the causal relationships between the variables take any particular functional form. We will assume (for now) that the true causal graph is acyclic since the algorithms for cyclic structures require linearity in the functional relationships between variables.

Given these assumptions, Pearl (1988) proved that there will be an edge (a line) in our undirected dependency graph between a pair of variables (X and Y) if X and Y are dependent conditional on every subset[238] of variables in the graph that does not include X or Y. We can therefore discover the undirected dependency graph of the causal process that generated our data by applying the algorithm described in section 9.5. Let's define the *conditioning order* of an association as the number of variables in the conditioning set. So, a zero-order association is an association between two variables without conditioning (an empty conditioning set), a first-order association is an association between two variables conditioned on one other variable, and so on. How you measure these associations will depend on the nature of the data; a natural way to do this is by using the generalize covariance statistic (section 3.6 of Chapter 3).

9.5 The undirected dependency graph algorithm[239]

The first step is to form the *complete*, or *saturated*, undirected graph involving the V observed variables. In other words, add a line between each observed variable and every other observed variable in the data. Since latent variables are, by definition, unmeasured, we cannot include

[238] Including the empty subset.
[239] This algorithm is included in the SGS algorithm of Spirtes, Glymour and Scheines (Spirtes et al. 1993).

them in our complete undirected graph. Now, for each unique pair of observed variables (X, Y) that have a line between them in the undirected dependency graph at any stage during the implementation of the algorithm, do the following:

Let the conditioning order of the association be zero.

> 1.1 Form every possible set of conditioning variables, containing the number of variables specified by the conditioning order, out of the remaining observed variables in the graph.
> 1.2 If the association between the pair of variables (X, Y) is zero when conditioned on any of these sets, then remove the line between X and Y from the undirected dependency graph, move on to a new pair of variables that still have a line between them, and then go to step 1.1.
> 1.3 If the association between the pair of variables (X, Y) is not zero when conditioned on all of these sets, then increase the conditioning order of the association by one and go to back to step 1.1. If you cannot increase the conditioning order (because you have used up all of the remaining variables), then the line between your 2 variables is kept. Move on to a new pair of variables.

Once you have applied this algorithm to every set of observed variables, the result is the undirected dependency graph. Given the assumptions listed above, you are guaranteed to obtain the correct undirected dependency graph of the causal process that generated the data if the algorithm is properly implemented. To illustrate this algorithm, let's imagine that we have been given data (lots of it so that we do not have to worry about sampling variation or statistical power[240]) that, unknown to us, was generated by the causal graph shown in Figure 9.3a.

[240] This is simply the first of our three assumptions: For each possible association, or partial association, we definitely know if the association or partial association involving the measured variables in the data exists (i.e., is different from zero) or doesn't exist (is equal to zero).

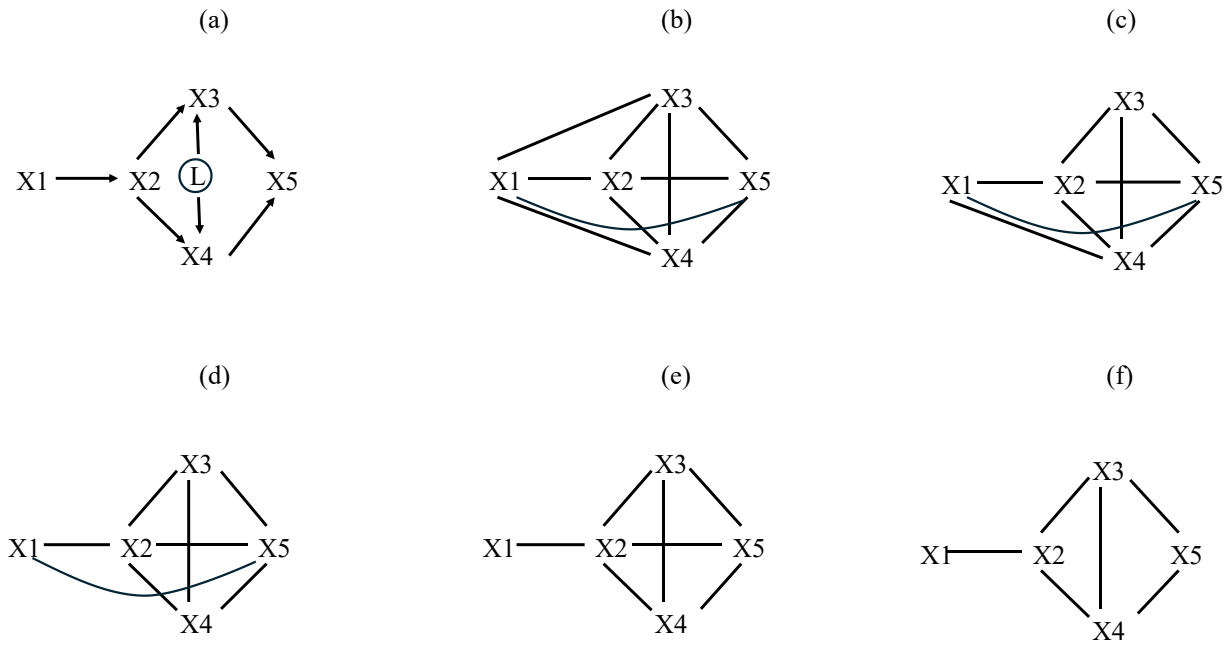


Figure 9.3. The true causal structure generating the data is in (a), including a latent variable (L) that is not in the data set. The undirected dependency graphs that result from different stages of the algorithm are shown in graphs (b) to (f), with (f) being the final undirected dependency graph.

The true causal generating structure is as shown in Figure 9.3a but we don't know this! In other words, this causal structure is hidden behind the screen of Nature's Shadow Play. In fact, we don't even know of the existence of the latent variable (L). All that we have is a (very large) data set containing observations on the variables X1 to X5 and a series of statements of association and partial association between them. These statements of association and partial association are the shadows that we can observe on the screen. Our task is to infer as much about the structure of Figure 9.3a as we can.

To begin, we create the complete undirected dependency graph of these five variables (Figure 9.3b), in which each variable is joined to each other variable by a line. Notice that the latent variable (L) doesn't appear in Figure 9.3b because we are only dealing with observed variables. Let's begin with the pair (X1, X2) and apply the algorithm. The starting conditioning order is zero (i.e. no conditioning variables and an empty conditioning set). Since X1 and X2 are adjacent in the true causal structure (Figure 9.3a) then these two variables are not unconditionally d-separated. Because d-separation implies independence, we will find that the pair is associated in our data when we test for a zero association (independence). Therefore, the

line X1─X2 in Figure 9.3b remains after the zero-order step. We then increase the conditioning order to 1 and see if X1 and X2 become independent upon conditioning on any of the possible first-order conditioning sets: {X3}, {X4}, {X5}. These are the only first-order conditioning sets that we can form from five variables while excluding variables X1 and X2. From Figure 9.3a we know that X1 and X2 are not d-separated given any of these sets. Therefore, they will not be independent in our data upon first-order conditioning and the line between them in Figure 9.3b remains after this step. We continue by increasing the conditioning order to 2 and test for independence relative to the following sets: {X3, X4}, {X3, X5}, {X4, X5}. These are the only second-order conditioning sets that we can form given these variables. Given the true causal structure in Figure 9.3a, we will find that the second-order association between X1 and X2 remains. We increase the conditioning order to 3. There is only one possible set of the remaining three variables taken three at a time and so we test for independence relative to the following conditioning set: {X3, X4, X5} but still the association between X1 and X2 will remain. Since we cannot increase the conditioning order further, we conclude that there is a line between X1 and X2 in the final undirected dependency graph. The lines X2─X3, X2─X4, X3─X5, X4─X5 and X3─X4 will also remain when we repeat this process for these pairs of variables, and for the same reason.

We then go on to a new pair of variables that are still joined by a line; in this case, X1 and X3. When we apply the algorithm to the pair (X1, X3) we will find that X1 and X3 are still zero-order associated since they are d-connected in Figure 9.3a. When we increase to order 1 and form the sets {X2}, {X4} and {X5} we will find that X1 and X3 become independent upon conditioning on X2. This is because, in Figure 9.3a (the true graph) X1 and X3 are d-separated given X2, and d-separation implies probabilistic independence. So, we remove the line between X1 and X3, giving Figure 9.3c. Since we have removed this line, we don't have to go any further with this pair.

When we apply the algorithm to the pair (X1, X4) we find that X1 and X4 also become independent upon conditioning on X2, and so we would remove the line from X1 and X4 (Figure 9.3d). This is because X1 and X4 are d-separated given X2. X1 and X5 would also become independent either upon conditioning on X2 or on the sets {X2, X3}, {X2,X4} or {X2, X3, X4},

since X1 and X5 are d-separated given any of these three sets of conditioning variables. We therefore remove the line X1─X5 (Figure 9.3e).

When we apply the algorithm to the pair (X2, X5), these two variables will never become independent conditional on only a single other variable because X2 and X5 are only d-separated conditional on both X3 and X5. Therefore, when we increase the conditioning order to 2, we will find that they become independent conditional of the pair (X3, X4). We therefore remove the line between X2 and X5 (Figure 9.3e). The undirected dependency graph that results after applying the algorithm to every possible pair is shown in Figure 9.3f; this is the correct undirected dependency graph given the causal process shown in Figure 9.3a.

9.6 Interpreting the undirected dependency graph

The undirected dependency graph informs us of the pattern of associations in our data. It doesn't inform us of the pattern of *causes* in our data. For instance, there is a line between X3 and X4 in Figure 9.3f (the final undirected dependency graph) even though, by peaking at the causal process that generated the data (Figure 9.3a) we know that the line X3─X4 is due only to the effect of the latent variable (L). Just as the term "direct" cause can only have meaning relative to the other variables in the causal explanation, a "direct" association can only have meaning relative to the other variables that have been measured. However, we can infer from the undirected dependency graph that if two variables have a line between them then there is either:

(i) a direct causal relationship between the two and/or:

(ii) there is a latent variable that is a common cause of the two and/or:

(iii) there is a more complicated type of path between the two, called an inducing path; an inducing path was explained in section 6.4 of Chapter 6, and I will remind you of it soon.

At the same time, we can exclude other types of latent variables. For instance, we know that there is no latent variable that is a common cause of X1, X2 and X3 in Figure 9.3f. If there

where, then X1 and X3 would not be d-separated given any set of other observed variables and there would therefore be a line between X1 and X3 in the undirected dependency graph.

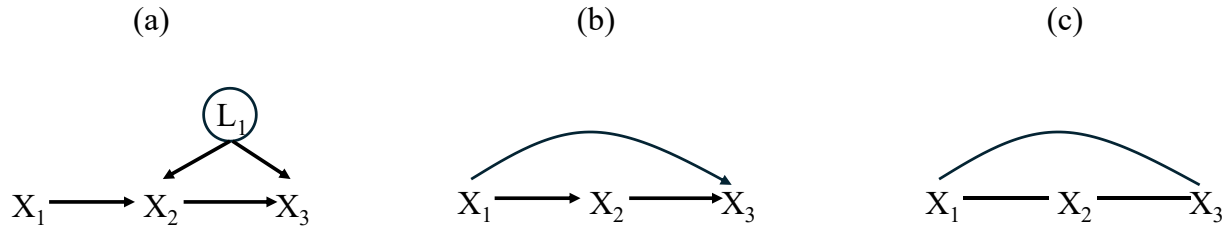


Figure 9.4. The true causal generating process is shown in (a). The m-equivalent mixed acyclic graph (MAG) is shown in (b), in which $X_1 \rightarrow X_3$ is an inducing path. The undirected dependency graph is shown in (c).

Let's look at the third possibility more closely; namely, that the line between two variables in the undirected dependency graph (X_i—X_j) is generated by an *inducing* path from one to the other. In section 6.4 I defined an inducing path from X_i to X_j as one in which (i) X_i is a cause (but not necessarily a *direct* cause) of X_j in the causal structure generating the data and (ii) the dependence between X_i and X_j cannot be removed by conditioning on any possible subset of other *observed* variables. This occurs if there is at least one directed path from X_i to X_j in the true causal structure that includes[241] at least one latent variable and in which each observed variable along that path is a collider (excluding the endpoints X_i and X_j) and is an ancestor of either X_i or X_j , while every latent variable along that path is a non-collider (Verma & Pearl 1991). To illustrate this, look at Figure 9.4, in which X_1, X_2 and X_3 are the observed variables (and so are in the data set) while L_1 is a latent variable (and so is not in the data set). The true causal generating structure is shown in Figure 9.4a. There are two paths from X_1 to X_3: $X_1 \rightarrow X_2 \rightarrow X_3$ and $X_1 \rightarrow X_2 \leftarrow L_1 \rightarrow X_3$. The second path ($X_1 \rightarrow X_2 \leftarrow L_1 \rightarrow X_3$) is an inducing path from X_1 to X_3 because all of its observed variables (excluding the endpoints X_1 and X_3, thus X_2) are colliders along this path and X_2 is an ancestor of X3. Notice that X_1 is never d-separated from X_3 given any possible set of conditioning variables involving only observed variables. X_1 is not d-separated from X_3 given the empty conditioning set ($X_1 \_||\_ X_3|\phi$) because of the directed path $X_1 \rightarrow X_2 \rightarrow X_3$. They are not d-separated given X_2 because of the path $X_1 \rightarrow X_2 \leftarrow L_1 \rightarrow X_3$, since conditioning on X_2 opens up this path even as it closes the other one ($X_1 \rightarrow X_2 \rightarrow X_3$).

[241] This requirement is not included in Verma and Pearl's original definition since they include direct effects ($X_i \rightarrow X_j$) as a special case of an inducing path. I include it here to differentiate the two.

If you have completely mastered the concepts in Chapter 6, you might be wondering about the relationship between an m-equivalent MAG and the undirected dependency graph. If fact, the undirected dependency graph is simply an m-equivalent MAG with all of the directed edges (i.e. $\rightarrow$, $\leftarrow$ and $\leftrightarrow$) replaced with lines! You could say that the undirected dependency graph is the "skeleton" of the m-equivalent MAG. The algorithm, given above, that produces the undirected dependency graph from the patterns of (conditional) independence in the data also automatically produces the skeleton of the m-equivalent MAG.

One practical problem with the algorithm that I have presented for obtaining the undirected dependency graph is that, as the number of observed variables increases, the number of sets of conditioning variables increases geometrically. When faced with large numbers (say, 50) observed variables, even fast personal computers might take a long time to construct the undirected graph if the topology of the true causal graph is uncooperative. The `CI.algorithm()` function in the pwSEM package, that I will explain later, is even slower and you would have time to get a coffee while it is running given only 10 variables! A slightly modified version of the algorithm[242] is presented in Spirtes, Glymour and Scheines (Spirtes et al. 1993) and it is more efficient when dealing with many observed variables. The two algorithms are equivalent given population measures of association, but the more efficient algorithm can make more mistakes in small data sets. The TETRAD II program[243] is faster (and more complete) but is a stand-alone program. The R package pcalg also has functions implementing both the CI and FCI algorithms but are limited to non-nested data and to variables that are either gaussian, discrete or binomial.

We sometimes have independent information about some of the causal relationships governing our data. In such cases it is straightforward to modify the algorithm for the undirected dependency graph to incorporate such information. If we know that the association between two observed variables is due only to the fact that another measured variable, or set of measured variables, is a common cause of both, then we simply remove that edge before applying the algorithm. Similarly, if we know that two observed variables either have a direct causal

[242] This modified algorithm is incorporated in their PC algorithm. The algorithm that I have described forms part of their SGS algorithm.
[243] www.cmu.edu/dietrich/philosophy/tetrad/. The authors of the TETRAD II program are working on an R interface, but it is not yet user-friendly in 2025.

relationship, or share at least one common latent cause, then we simply forbid the algorithm from removing the line joining that pair. Note that it is not enough to know (say from a randomised experiment) that one measured variable is *a* cause of another; we must know that it is (or is not) a *direct* cause relative to all of the other observed variables. A randomised experiment will not be able to tell us this if some of the observed variables are attributes of the experimental units, as explained in Chapter 1.

9.7 Orienting edges in the undirected dependency graph using unshielded colliders assuming an acyclic causal structure

You already know that if a causal graph has a pattern like X→Z←Y, then X and Y will not be d-separated given any conditioning set that contains Z. In general[244], if we have two variables (X and Y) and condition on some set of variables **Q** that contains at least one common causal descendent of both X and Y, then X and Y will not be d-separated. Because of this X and Y will not be probabilistically independent upon conditioning on **Q** even if X and Y are causally independent. This fact allows us to determine the causal direction of some lines in the undirected dependency graph.

In Chapter 2 I defined an *unshielded collider* as a causal relationship between three variables (X, Y and Z) in a DAG such that both X and Y are direct causes of Z (X→Z←Y) but there is no direct causal relationship between X and Y (i.e. there is no arrow going from one to the other[245]). Let's now define an *unshielded "pattern"* in an undirected dependency graph as one in which we have three variables (X, Y and Z) such that there is a line between X and Y, a line between Y and Z (X—Y—Z), but no line between X and Z. Since there is no line between X and Z, we know that X and Z are d-separated given some subset of other variables in the undirected dependency graph since this is what the lack of a line means. Given an unshielded pattern X—Y—Z, we can decide if there are arrowheads pointing into Y from both directions in the causal graph that

[244] This is true for acyclic causal structures but not for cyclic causal structures. This is discussed in more detail later.
[245] If we were dealing with a mixed acyclic graph rather than a directed acyclic graph then there must not be any edge at all, either an arrow or any double headed arrows, between X and Y.

generated the data. If there are arrowheads pointing into Y from both directions in the actual causal process generating the data (i.e. Xo→Y←oZ), then X and Z will never by probabilistically independent conditional on any set of other observed variables that includes Y. The "o" symbol (Spirtes et al. 1993) is simply a placeholder to tell us that we don't yet know if we should add an arrowhead or not in that position. Therefore, Xo→Y←oZ means that there are definitely arrowheads pointing into Y from both directions, but we don't yet know if there are also arrowheads pointing into X and/or into Y. Thus, the actual orientation could be X→Y←Z, X↔Y←Z, X→Y↔Z or X↔Y↔Z but we don't yet know which one is correct.

Therefore, for each triplet of variables in the undirected dependency graph that form an unshielded pattern (X_i─X_j─X_k), i.e. with no line between X_i and X_k, determine if X_i and X_k is independent conditional on X_j plus every other possible set of remaining variables. If X_i and X_k are independent in any of these conditioning sets, then X_j cannot be a collider variable and so there cannot be an arrowhead pointing into X_k from both directions. If X_i and X_k are not independent in anyof these conditioning sets, then X_j must be a collider variable and so there must be an arrowhead pointing into X_k from both directions. Here is the orientation phase of the algorithm[246] to do this:

Let the unshielded pattern be X_i─X_j─X_k. Let the conditioning order be O=1. Let **Q** be the set of variables in the undirected dependency graph except for X_i, X_j or X_k.

> 1. Form all possible sets of O variables in **Q**, excluding the empty set. Let one of these possible sets be $\mathbf{Q_c}$. Let the conditioning set **C** be be X_j plus any of the $\mathbf{Q_c}$ sets: **C**={X_j, $\mathbf{Q_c}$}
>
> 2.1 If X_i and X_k, conditioned on any set **C**, is independent then stop and conclude that the three variables forming the unshielded pattern do not form an unshielded collider in the true causal graph (i.e. not X_io→X_j←oX_k). We can call such a pattern a *definite non-collider* and write it as Xo─oZo─oY or Xo─<u>oZo</u>─oY with the underline emphasizing that there cannot be arrows pointing into Z from both directions. The fact that there are placeholder "o" symbols at both sides of the middle variable (Z) rather than arrowheads

[246] This algorithm is used in Pearl's IC (Inductive Causation) algorithm. The related algorithm in Spirtes, Glymour and Scheines (Spirtes et al. 1993) uses a set called **Sepset**(X,Y) that reduces the computational burden. The output is identical in acyclic causal structures but can be different in cyclic causal structures.

means that there cannot be arrowheads pointing into the middle variable (Z) from both sides. This means that the underline is redundant, and it is sometimes omitted.

2.2 If X_i and X_k, conditioned on any set **C**, is independent, then increase the conditioning number (O) by one and go to back to step 1.

After cycling through all possible orders of O, if we have not declared the unshielded pattern to be a definite non-collider, then it is a collider. Orient the pattern as: $X_i o\!\rightarrow X_j \leftarrow\! o X_k$. Again, the "o" symbol is simply a placeholder to tell us that we don't yet know if there is an arrowhead or not.

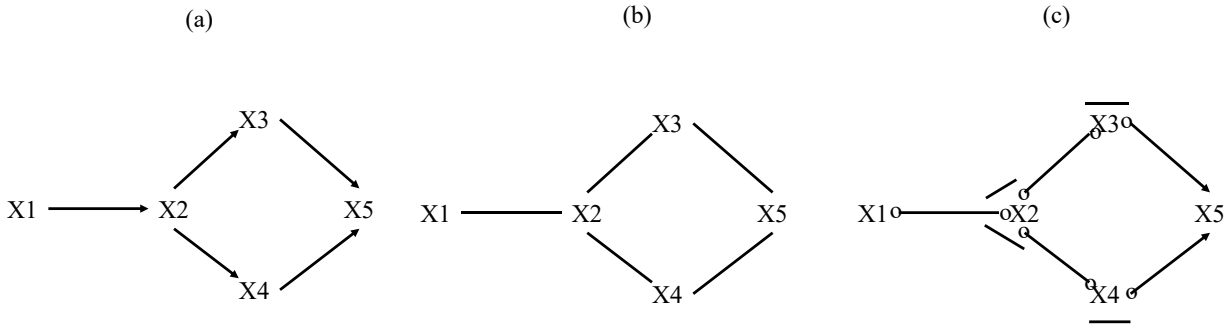


Figure 9.5. The true causal generating process is shown in (a). The resulting undirected dependency graph is shown in (b). The partially oriented dependency graph is shown in (c).

To illustrate this method of orienting our undirected edges in the undirected dependency graph, imagine that the unknown causal process generating our observed data is as shown in Figure 9.5a. Even though the causal process, and therefore the true DAG, is hidden from us, we will obtain[247] the undirected dependency graph shown in Figure 9.5b once we apply the algorithm to our data. The undirected dependency graph in Figure 9.6b has 6 unshielded patterns: X1─X2─X3, X3─X2─X4, X1─X2─X4, X2─X3─X5, X2─X4─X5 and X3─X5─X4. Since, in this example, we can peak at the true causal graph (top of Figure 9.5a), we can use d-separation to predict what would happen if we applied the above algorithm to each of the six unshielded patterns that we found in our partially oriented graph. For instance, when we test the unshielded pattern X1─X2─X3, we would begin the orientation phase of the algorithm by testing for a dependency between X1 and X3 conditional only on X2. The remaining variables are **Q**={X4,

[247] Given the assumptions stated earlier.

X5}. The possible subsets of **Q** are{X4}, {X5} and {X4, X5}. We start with O=1 and so form the conditioning sets as $X2 \cup X4 =$ {X2, X4} and $X2 \cup X5 =$ {X2, X5}. Since X1 and X3 are d-separated given either {X2, X4} and {X2, X5}, they will be conditionally independent given either of these conditioning sets. We therefore stop right away, conclude that the causal effects do not collide at X2 and declare that X1─X2─X3 is a definite non-collider: X1o─oX2o─oX3. In other words, we know that the following orientations are excluded: X2→X2←X3, X2↔X2←X3, X2→X2↔X3 and X2→X2↔X3 because X2 is a collider in all of them.

Now, consider the triplet X3─X5─X4. **Q**={X1, X2}. The possible subsets of **Q** are{X1}, {X2} and {X1, X2}. We start with O=1 (thus, only one additional conditioning variable besides X5) and so form the conditioning sets as $X5 \cup X1 =$ {X5, X1} and $X5 \cup X2 =$ {X5, X2}. Since X5 is a collider in the true generating process, then X3 and X4 will not be d-separated by either of these conditioning sets and so will not be conditionally independent given either of these conditioning sets. We then increase the conditioning order to O=2 (thus, two additional conditioning variables besides X5) and so form the single conditioning set $X5 \cup X1 \cup X2 =$ {X5, X1, X2}. Since X5 is a collider in the true generating process, X3 and X4 will still not be d-separated by this conditioning set and so X3 and X4 will not be conditionally independent given this single conditioning set. Therefore, X3 and X4 are never conditionally independent given any possible conditioning set containing X5. We conclude that the triplet X3─X5─X4 collides on X5: X3o→X5←oX4. The possible orientations are therefore X3→X5←X4, X3↔X5←X4, X3→X5↔X4 or X3↔X5↔X4. The full results, and their explanations, are given in Table 9.1.

Table 9.1. Applying the orientation algorithm using unshielded colliders to the undirected dependency graph in Figure 9.5b.

Unshielded pattern	Partially oriented pattern	Explanation
X1─X2─X3	X1o─oX2o─oX3	X1 and X3 are d-separated given X2 and so they will be conditionally independent given **C**={X2,**Q**} where **Q** is any subset of {ϕ,X4,X5}. Therefore, X2 is a non-collider along this path.
X3─X2─X4	X3o─oX2o─oX4	X3 and X4 are d-separated given X2 and so they will be conditionally independent given **C**={X2,**Q**} where **Q** is

		any subset of {ϕ,X3,X5}. Therefore, X2 is a non-collider along this path.
X1─X2─X4	X1o─oX2o─oX4	X1 and X4 are d-separated given X2 and so they will be conditionally independent given **C**={X2,**Q**} where **Q** is any subset of {ϕ,X1,X3}. Therefore, X2 is a non-collider along this path.
X2─X3─X5	X2o─oX3o─oX5	X2 and X5 are d-separated given {X3, X4} and so they will be conditionally independent given **C**={X3,**Q**} where **Q** is any subset of {ϕ,X1,X4}; specifically **C**={X3,X4}. Therefore, X3 is a non-collider along this path.
X2─X4─X5	X2o─oX4o─oX5	X2 and X5 are d-separated given {X3, X4} and so they will be conditionally independent given **C**={X4,**Q**} where **Q** is any subset of {ϕ,X1,X3}; specifically, **C**={X3,X4}. Therefore, X4 is a non-collider along this path.
X3─X5─X4	X3o→X5←oX4	X3 and X4 are never d-separated given X5 because X5 is a collider along (X3→X5←X4). Therefore, X3 and X4 will never be independent given **C**={X5,**Q**} where **Q** is any subset of {ϕ,X1,X2}. Therefore, X5 is a collider along this path.

The combination of the algorithm to produce the undirected dependency graph and the algorithm to orient some edges using the notion of an unshielded collider produces the IC algorithm of Verma and Pearl (Verma and Pearl 1990). Spirtes, Glymour and Scheines (Spirtes et al. 1993) independently published and proved an algorithm called the SGS algorithm[248] that includes the IC algorithm plus an additional orientation step involving a "definite discriminating path". I explained definite discrimination paths in the second edition of this book, but the notion is quite involved and (in my experience) is even more sensitive to sample size limitations than are the

[248] SGS simply means Spirtes, Glymour and Sheines.

other steps, and so I skip it here. If you want to learn about definite discriminating paths, go to the original publication (p. 181) or to the second edition of this book.

The final partially oriented dependency graph is shown in Figure 9.5c. A line over (or under) the middle variable of an unshielded pattern, such as $X1o-o\overline{X2}o-oX3$ or X1o─oX2o─oX3, means that X2 is a definite non-collider in this triplet. In fact, since the partially-oriented edges indicate inducing paths, Spirtes, Glymour and Scheines (Spirtes et al. 1993) call this final partially oriented dependency graph a *partially-oriented inducing path graph* or POIPG (try pronouncing that one fast!).

There is a simple rule when completing the orientation of a partially oriented dependency graph: you can replace the the "o" symbols (i.e. the undetermined orientations) with arrowheads as long as you do not create any new unshielded colliders or any feedback cycles (i.e. a directed path that starts and ends at the same variable). If this sounds familiar, it is because these are the same rules for obtaining d-separation (or m-separation) equivalent DAGs (or MAGs) that I gave in section 5.6 of Chapter 5. It is important that you clearly make the distinction between an unshielded collider and a shielded collider because you can create shielded colliders when completing the orientation, just not unshielded colliders. For example, Figure 9.6a shows the same true causal generating process as in Figure 9.2 and Figure 9.6b shows the resulting partially oriented dependency graph. Notice that there are only two unshielded patterns: X1o─oX2o─oX3 and X1o─X2o─oX4. Because the only constraints are to not have collisions on X2 involving these two unshielded patterns, there are more ways of completing the orientation, of which the MAGs in Figure 9.6c,d are only two.

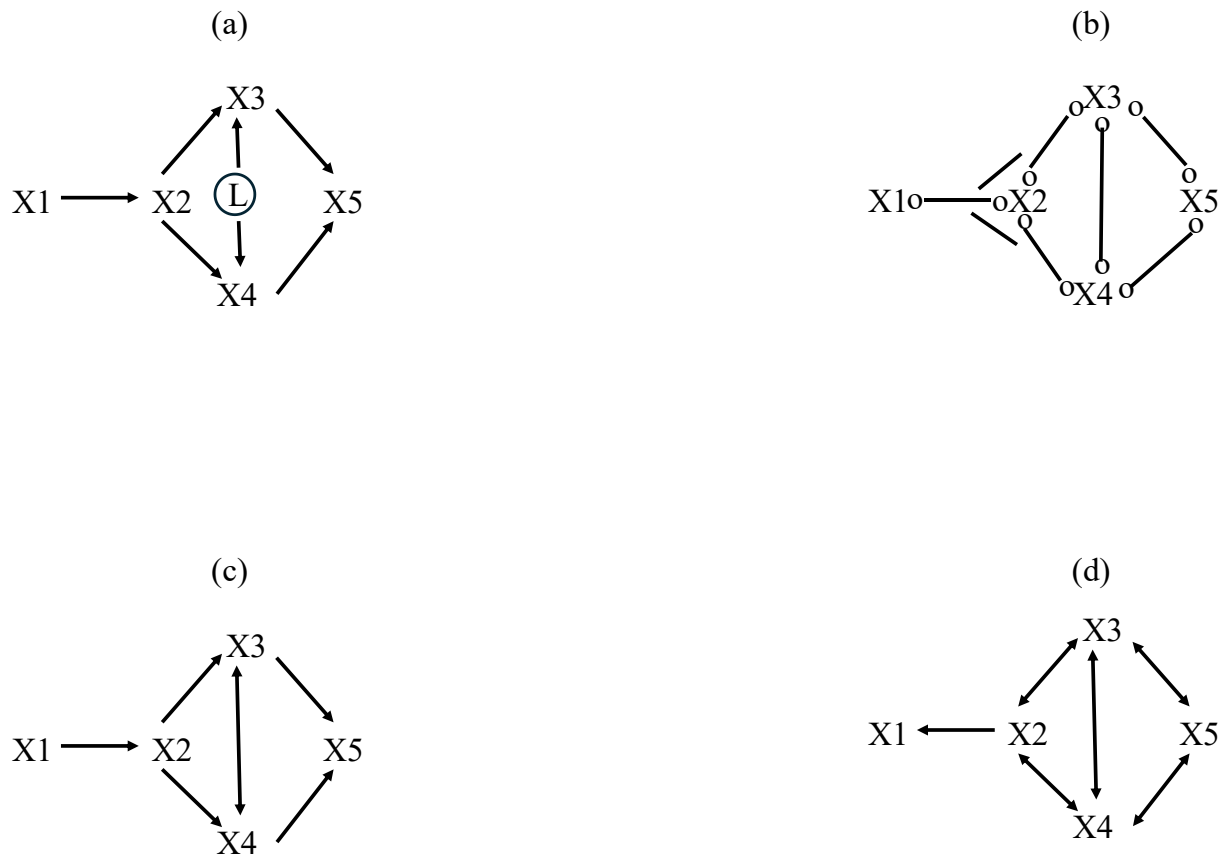


Figure 9.6. The true causal generating process is shown in (a). The resulting partially oriented dependency graph is shown in (b). Two possible complete orientations, out of many, are shown in (c) and (d).

9.8 The Causal Inference (CI) algorithm when the assumptions don't hold

The CI algorithm is provably correct given the assumptions. In other words, you are guaranteed to produce the correct partially oriented dependency graph, given information on the patterns of (conditional) independence among the variables and given these three assumptions: (1) for each possible association, or partial association, we definitely know if the association or partial association involving the measured variables in the data exists or doesn't exist, (2) every unit in the population is governed by the same causal process, and (3) the probability distribution of the observed variables measured on each unit is faithful to some (possibly unknown) cyclic or acyclic causal graph. The CI algorithm goes a long way towards realizing the philosopher's

dream of inferring causation from correlation. However, these three assumptions are big ones and, unfortunately, they are almost never met in real applications! Let's look at them in more detail so that we can see what can go wrong when they are not met and what we can do about it.

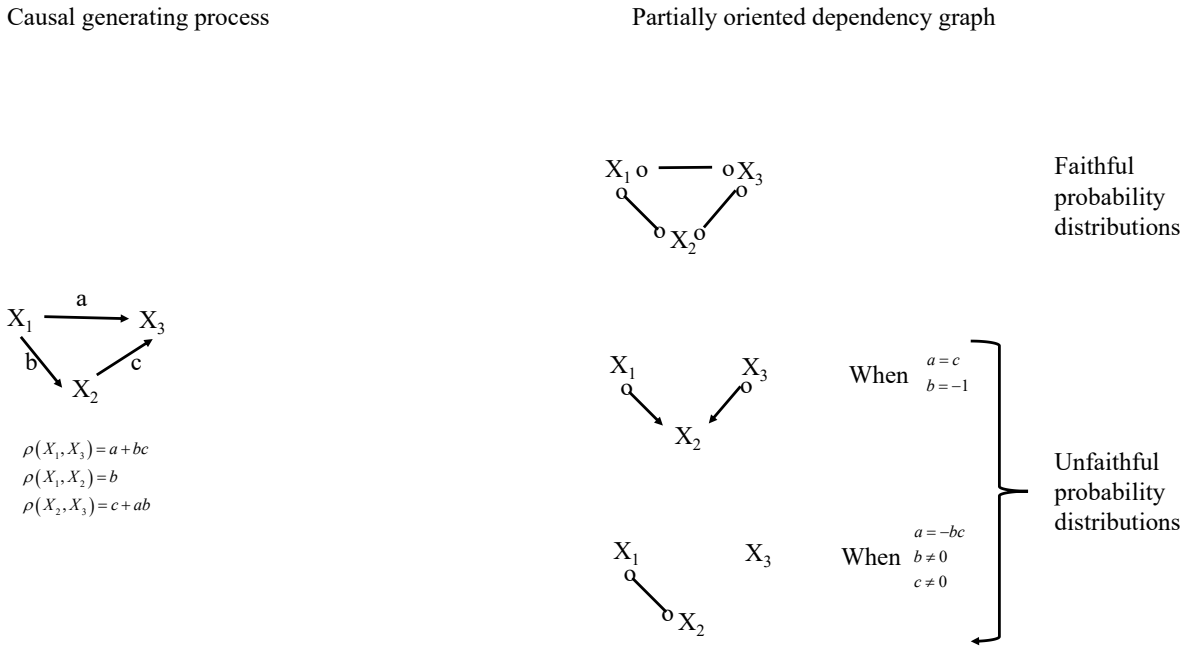


Figure 9.7. The true causal generating process (left) assuming linearity and normality. The three possible partially oriented dependency graphs (right), of which two result from unfaithful probability distributions generated by special cases.

I start with the third assumption (faithfulness) because it is the least dangerous one in practice. Remember (Chapter 2) that d-separation[249] in the causal graph that generates the data always implies (conditional) independence in the resulting probability distribution describing the data. However, the converse is not always true: it is possible to have (conditional) independence in the probability distribution without d-separation. When there exists a (conditional) independence in the probability distribution without the causal graph having a d-separation claim, we say that the probability distribution is "unfaithful" to the causal graph. This occurs when different causal paths perfectly cancel each other. Consider Figure 9.7, which shows a causal generating process (left) having[250] linear relationships and multinormal variables. Note that none of the variables in the DAG of Figure 9.7 are d-separated given any possible conditioning sets. Except under special conditions, this means that none of the variables in the resulting probability distribution

[249] This also applies to the more general notion of m-separation (Chapter 6) in a mixed acyclic graph.
[250] In order to use the rules for calculating total and indirect effects

should be (conditionally) independent of any other variable unless the probability distribution is unfaithful to the DAG. One special condition in which the probability distribution is unfaithful to the DAG is the set of path coefficients in which a=c and b=-1. Given this special case, then the total effect of X_1 on X_3 is a+bc = a+-1a=0, meaning that X_1 and X_3 are independent, while the effect of X_1 on X_3, conditional on X_2 is a≠0. The fact that the total effect of X_1 on X_3 is zero means that the Pearson correlation between them is zero, and so X_1 and X_3 are independent even though they are not d-separated. The fact that X_1 and X_3 are independent even though they are not d-separated makes the probability distribution unfaithful to the true generating process. There is no line between X_1 and X_3 in the partially oriented dependency graph (because of the independence between them) and the arrows point into X_2 (because X_1 and X_3 are not independent conditional on X_2). This results in an incorrect partially oriented dependency graph. Another special case is when a=-bc and b≠0 and c≠0. Now, the total effect of X_1 on X_3 is zero (a+bc = a=-1a = 0) and the total effect of X_2 on X_3 is also zero (c+ab = a-1a=0). The fact that both X_1 and X_3 and X_2 and X_3 are independent in the probability distribution without them being d-separated is another case of unfaithfulness, resulting in an incorrect partially oriented dependency graph. I said that a violation of the assumption of faithfulness is the least dangerous one in practice. This is because the conditions required for unfaithfulness are very specific, unusual, and finely balanced. In the example of Figure 9.7, any combination of values for the path coefficients other than for these two special cases[251], independence in the data will be mirrored by a d-separation claim. In other words, even a tiny change those special conditions will make the resulting probability distribution faithful. By assuming faithfulness, we are assuming that our data did not come from such special cases in which the effects along different paths are perfectly balanced such that they perfectly cancel each other out.

Now consider the second assumption: every unit in the population is governed by the same causal process. If this is not true, if there are subsets of observations that are generated by different causal processes, then the result of combining them together into a single data set will result in a mixture of true causal graphs, not a single causal graph (see Chapter 8). The CI algorithm will be trying to identify a single partially oriented dependency graph when it doesn't exist. If you have subsets of observations taken from different situations, for instance, in

[251] Excluding the trivial case in which a=b=c=0.

different sites or years, then you can reduce the chances of violating this second assumption by applying the CI algorithm separately to each subset in the data (different sites, different times, different species etc.). I suspect that violations of this second assumption will become more likely as you increase the sample size of your data by including observations from more different situations since you might not even be aware of the different cases that represent different causal generating processes.

Finally, consider the first assumption: for each possible association, or partial association, we definitely know if the association or partial association involving the measured variables in the data exists (i.e., is different from zero) or doesn't exist (is equal to zero). This assumption is the most difficult one to justify because, strictly speaking, it is always false unless you have access to the entire statistical population. At every step in the CI algorithm, we are required to decide if a pair of variables are (conditionally) independent or dependent. However, we only ever have access to a random sample of the full statistical population. When we are dealing with a random sample rather than the full statistical population, we must choose a null probability when deciding if independence in the sample also exists in the full statistical population and then accept the resulting Type I error (section 5.1 of Chapter 5) of being wrong. The smaller the sample size, the greater the chance of mistaking a small, but real, dependency for independence at a given significance level. Making this mistake in the first part of the CI algorithm (when constructing the undirected dependency graph) will result in incorrectly removing lines from the dependency graph. In the second part of the CI algorithm (to orient the lines based unshielded patterns of triplets), we must constantly decide on the conditional dependence between the exterior variables of an unshielded collider given the middle variable plus every possible subset of remaining variables. That means accepting the resulting Type II errors (section 5.1 of Chapter 5). Making this mistake will result in incorrectly declaring a triplet to be an unshielded collider. Even worse, since the second part of the CI algorithm is based on the dependency graph produced in the first part, any errors in the first part will leak into the second part.

One partial solution to this problem is to start with a low rejection level (say, 0.01) when judging the null probability of (conditional) independence. You obtain the partially oriented dependency graph, complete the orientation to get an m-equivalent MAG, and then test one of the m-equivalent MAGs using the dsep or msep test using your normal significance level (typically

0.05). It doesn't matter which m-equivalent MAG that you choose since all of them will have the same probability. If the m-equivalent MAG is rejected, then increase the rejection level and try again using your normal significance level. Keep doing this until you find an m-equivalent MAG that is not rejected. Of course, you can continue to increase the rejection level used to generate the partially oriented dependency graph and see how the resulting m-equivalent MAGs change. Remember: you are only exploring the data to generate ideas for credible causal hypotheses; you are not testing an *a priori* causal hypothesis. You can use such exploratory methods to propose causal hypotheses worth testing, but you cannot simultaneously claim to have conducted an independent test of these hypotheses!

I will use the `CI.algorithm()` function of the pwSEM package to illustrate this method:

```
CI.algorithm(dat, family=NA,nesting=NA,smooth=TRUE,alpha.reject
= 0.05, constrained.edges=NA, write.result = TRUE)
```

The first argument (`dat=`) inputs your data set as a data frame. All variables in this data frame, except for nesting variables (if present) that are given in the `nesting=` argument, are used to construct the partially oriented dependency graph. If you want to exclude certain variables then you must exclude them by subsetting the data frame[252]. The second argument (`family=`) specifies the distributional type of each variable (gaussian, poisson, binomial, gamma). It is a data frame having the same number of elements as there are variables in the data frame (excepting, if present, nesting variables). A variable is assumed to be normally distributed unless you explicitly assign it a different distribution, so you only need to explicitly include those variables in the data frame that are not normally distributed. If your data has some nested structure (Chapter 8, sections 8.8 and 8.9), then you give this as a list (an example will follow). The `smooth=` argument specifies if you want to assume linear (or generalized linear) relationships (`smooth=FALSE`) or non-linear relationships via smoother functions (`smooth=TRUE`). The `alpha.reject=` argument specifies the rejection level to be used in each of the many tests of (conditional) independence done in the function. This is *not* a significance level for judging the correctness of the resulting partially oriented dependency graph. Rather, it is a significance level for judging if each possible pair of variables is

[252] For instance, if you want to exclude a variable named "Var1" from your data frame called my.dat then you would use `dat=my.dat[,-my.dat$Var1]`.

(conditionally) independent in the CI algorithm. You could begin at `alpha.reject=0.01`, choose an m-equivalent MAG from the resulting partially oriented dependency graph, and test it using piecewise or covariance-based SEM. If the m-equivalent MAG is rejected, then increase the value of `alpha.reject=` and repeat the process. In practice, the smaller the number of observations in your data, the greater the chances of the algorithm making mistakes (especially at low levels of `alpha.reject`), and the larger the value of `alpha.reject` that you will probably need. The `write.result=TRUE` argument causes the resulting MAG to be written to the screen while `write.result=FALSE` simply returns the MAG in the form of a matrix. The `constrained.edges=` argument is an optional character object that you would normally create before calling `CI.algorithm()`. This character object gives the edges that you want to force the CI algorithm to either remove or to impose. For instance, if you know that there cannot[253] be an edge between X and Y then you would specify "X|Y". If you know that X must be a direct cause of Y, then you would specify "X→Y". If you know that neither X nor Y are direct causes of the other, but that some latent variable is a direct cause of both, then you would specify "X<->Y". If you want to impose more than one constrained edge, then you would create a text object (I call this `my.constraints` below) which would include all of your constrained edges within quotes, and with a RETURN between each one, and then include the argument `constrained.edges=my.constraints` in the CI.algorithm function.

```
my.constraints<-"
X|Y
Z->W
W<->X
"
```

To illustrate, I generate simulated data sets following the DAG in Figure 9.5a having either 25, 100 or 500 observations. Each variable follows a standard normal distribution, and each path coefficient is 0.5 (thus, moderately strong effects). The result is shown in Figure 9.8 using α rejection values of α=0.05, 0.1 and 0.3. Here is the call[254] when using a rejection level of 0.05 for each possible (conditional) independence test:

```
CI.algorithm(dat=sim.dat,alpha.reject=0.05)
```

[253] In other words, you know that X is not a direct cause of Y, Y is not a direct cause of X, and neither X nor Y have a common latent cause.
[254] All of the optional arguments are left at their default values.

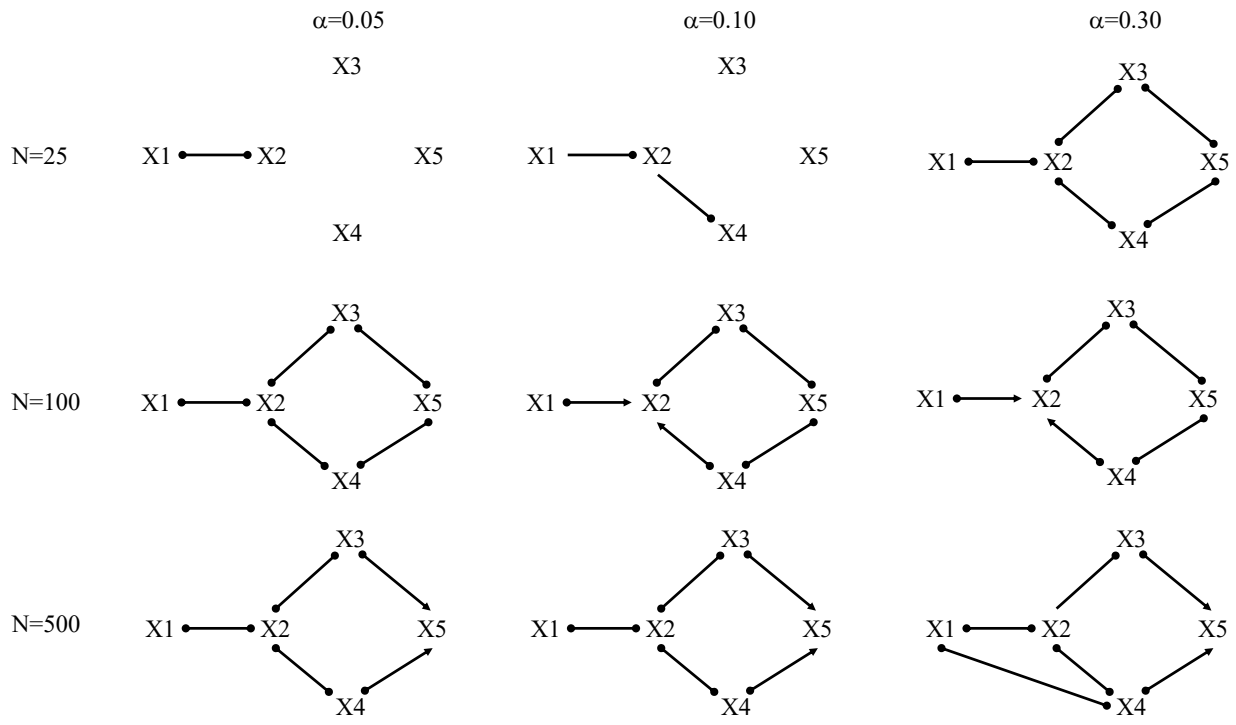


Figure 9.8. Output of the CI algorithm at three different sample sizes (N=25, 100, 500) and at three different rejection levels (α=0.05, 0.10, 0.30).

With a very small data set (N=25) and a strict rejection level (`alpha.reject=0.05`), the CI algorithm makes many mistakes. For instance, it mistakenly declares X3, X4 and X5 to be mutually independent. These mistakes arise because the small sample size and the strict rejection level results in very low statistical power to detect real, but only moderately strong correlations. Increasing the rejection level allows the algorithm to correctly interpret weak dependencies as real and so, at `alpha.reject=0.30`, it correctly recovers the undirected dependency graph but still fails to detect the unshielded collider at variable X5. At a moderate sample size (N=100), the CI algorithm correctly recovers the undirected dependency graph even at `alpha.reject=0.05` but, at `alpha.reject=0.30`, it mistakenly declares X2 to be an unshielded collider for the triplet X1o→X2←oX3 and misses the unshielded collider for the triplet X3o─oX5o─oX4. At a large sample size (N=500), the CI algorithm correctly produces the complete partially oriented dependency graph even at `alpha.reject=0.05`. If I increase

the rejection level to `alpha.reject=0.05`, then it incorrectly adds a line between X1 and X3.

Of course, the exact pattern of mistakes will differ for each data set, but the general pattern is the same. Rejection levels (i.e. `alpha.reject=`) that are too low will result in missing edges that should be present. Rejection levels that are too high will result in adding edges that are not really present. The number of these errors increase as the sample size decreases. Smaller sample sizes will increase the number of errors concerning unshielded colliders, especially because unshielded colliders are based on unshielded triplets in the undirected dependency graph and small sample sizes will increase the chances of incorrectly adding or removing edges. However, if you are aware of the types of errors that can occur at different sample sizes then the CI algorithm, in conjunction with any biological knowledge that you possess, can still guide you when you are simply exploring your data. Remember that the purpose of the CI algorithm is not to test pre-existing causal hypotheses but to suggest causal hypotheses that are testing.

What about a real biological data set? The obvious problem with using the CI algorithm on an empirical data set is that we can't know what the "true" causal graph (i.e. the causal data-generating mechanism) looks like. However, we can at least apply the algorithm to empirical data for which we have some independent evidence concerning the possible causal generating mechanism. For instance, in Chapter 3 I presented a hypothesised DAG for the Blue Tits data that had the twin virtues of (i) being consistent with the (partial) biological knowledge of the system and (ii) of not being rejected by the data. What happens if I apply the CI algorithm to these data without imposing any pre-existing constraints on the algorithm?

Since these data have a nesting structure, I must first create a list (`blue.tits.nesting`) describing this nesting structure for each variable. In this case, the observations are nested within years and within nests. This list is not needed if the data do not have a nesting structure, in which case you would also omit the `nesting=` argument in the call to `CI.algorithm()`. Next, I call `CI.algorithm()`, inputting only[255] those columns of the data set (`blue.tits`) that hold the variables to include in the causal graph and those describing the nesting structure. I have excluded the columns 3 (`ind`), 9 (`dateweighted`) and 10 (`massXhemato`) because

[255] If you don't do this then these additional variables will be included in the partially oriented dependency graph.

these variables will not be included in the partially oriented dependency graph, and they do not describe any nesting structure in the data. I excluded `massXhemato` because this was simply a variable created by the authors to encode the interaction term between body mass (`mass`) and haematocrit volume (`hemato`). You must always exclude any variable (except for ones encoding the nesting structure) that is not part of the causal graph that you are exploring. Since all of the variables in the causal graph are normally distributed except for the binary `recruited` variable, I only have to include this variable in the `family=` argument. If all of the variables in the partially oriented dependency graph are normally distributed, then you can ignore the `family=` argument. I must also define the nesting structure via `nesting = blue.tits.nesting.` By default (i.e. `smooth=TRUE`), the tests of conditional independence (which use the generalized covariance function explained in section 3.6 of Chapter 3) are based on regression smoothers rather than on linear relationships. If you want to use (generalized) linear models rather than regression smoothers, then you would specify `smooth=FALSE`. Here is the call:

```
blue.tits.nesting<-
list(recruited=c("year","nest"),mass=c("year","nest"),hemato=c("
year","nest"),protos=c("year","nest"),frass=c("year","nest"))

CI.algorithm(dat=blue.tits[,c(3,9,10)],family=data.frame(recruit
ed="binomial"),nesting=blue.tits.nesting)
```

The first step is to see how the partially oriented dependency graph changes as we increase the rejection level (`alpha.reject`); remember that this is the null probability below which the null hypothesis of (conditional) independence between each pair of variables is rejected and so the CI algorithm considers the pair to be dependent (Figure 9.9). Figure 9.9(a) shows the result when `alpha.reject`=0.01. Figure 9.9(b) shows the result when `alpha.reject` is between 0.05 and 0.1. Figure 9.9(c) shows the result when `alpha.reject` is between 0.2 and 0.4. When we require very strong evidence for dependence before rejecting our null hypothesis of independence (i.e., when `alpha.reject`=0.01), the algorithm declares that "recruited" is mutually independent from all of the other variables. When we weaken our requirement a bit (when `alpha.reject` is between 0.05 and 0.1), the algorithm adds the edge hemato o─o recruited. When we further weaken our requirement (when `alpha.reject` is between 0.2 and

0.4), it adds an additional edge between “frass” and “recruited” and also detects a collider in the unshielded triplet frass o→mass←o protos. Remember: these results are entirely based on the empirical patterns of (conditional) independence in the data and without any biological knowledge.

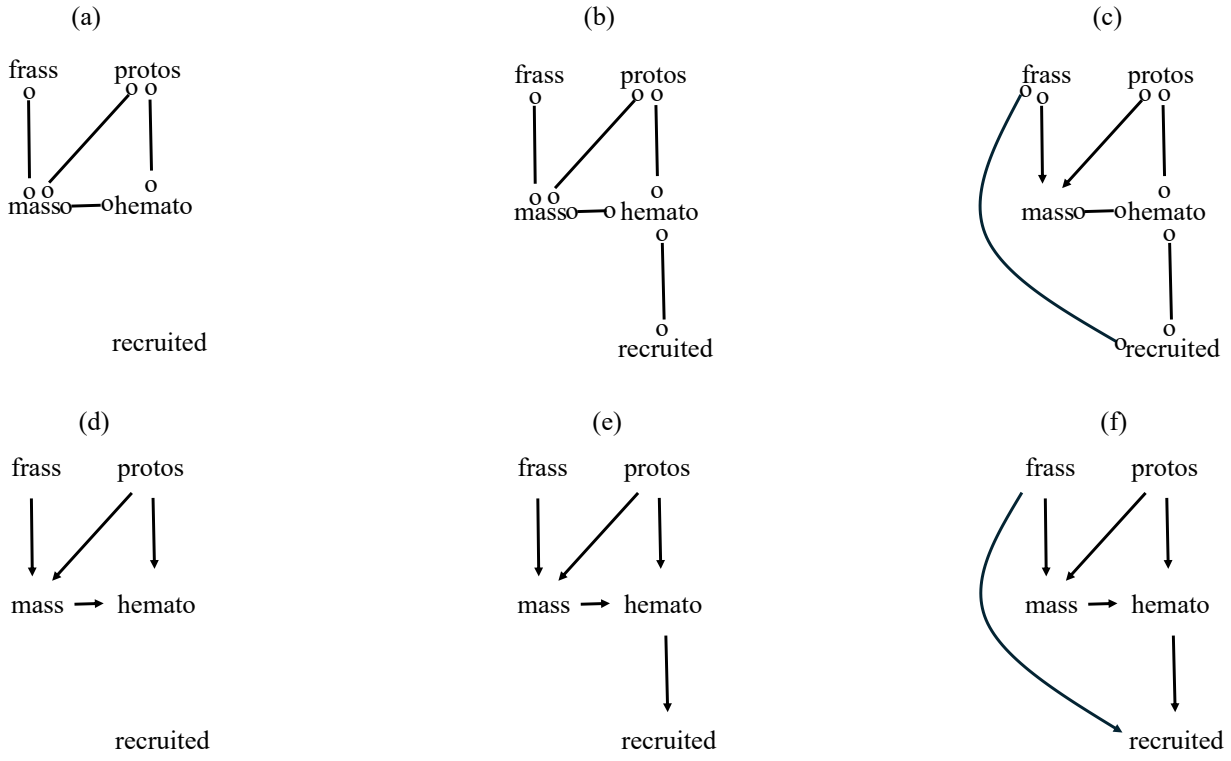


Figure 9.9. Output of the CI algorithm using the empirical Blue Tits data. (a) The partially oriented dependency graph using a rejection level of 0.01. (b) The partially oriented dependency graph using a rejection levels of between 0.05 and 0.10. (c) The partially oriented dependency graph using a rejection level of between 0.2 and 0.4. DAGs (d), (e) and (f) result from applying the orientation rules plus some biologically plausible arguments.

The `CI.algorithm()` function simply outputs the partially oriented dependency graph that results from the chosen value given to `alpha.reject`. You must still test the proposed causal graph against the empirical data using either piecewise or covariance-based SEM, and this requires completing the orientation of the edges following the rules given in section 5.6 of Chapter 5 as well as any biological information that you possess. For instance, the binary (yes/no) “recruited” variable refers to be ability of the young Blue Tit fledgling to leave the nest to then to return to the population the next year. The other variables measure what happened the year before the fledgling bird left the nest. Since causes cannot travel backwards in time, that means that any edge between any of these other variables and “recruited” must have an

arrowhead into it (o→recruited). Next, it seems biologically implausible that the number of caterpillars ("frass") or the number of nest parasites ("protos") could be caused by the mass or haematocrit volumes of the nestling birds[256]; indeed, this is what the partially oriented graph states when `alpha.reject` reaches a value of 0.2, although not below that level. Similarly, is seems more biologically plausible that the number of nest parasites ("protos"), which feed on the blood of the nestling, would cause changes in the haematocrit volume of a nestling ("hemato"), than the contrary. The orientation of mass o─o hemato seems the least obvious[257], but the requirement to not create a new unshielded collider at frass o→mass o─o hemato requires that we orient it as mass→hemato. This then forces the unshielded pattern mass→hemato o→recruited to be oriented as mass→hemato→recruited, again, in order to again avoid creating an unshielded collider. What should we do with the remaining unoriented parts of these alternative causal graphs (i.e., the "o" marks)? This depends on if we think that (for instance) the number of caterpillars (the food source) cause the changes is nestling masses (thus, frass→mass) or if neither are causes of the other but are associated due to some unknown common latent cause. It is certainly more biologically plausible that changes in the amount of food ("frass") and in the number of blood parasites ("protos") cause the changes in nestling mass and haematocrit volume than that these variables are only associated due to some unknown common causes.

These arguments lead to the DAGS d, e and f in Figure 9.9, and illustrate how the CI algorithm, in conjunction with incomplete biological knowledge, can generate causal hypotheses. Now, we can test the resulting DAGs using the `pwSEM()` function. DAG (d) results in a null probability of 0.053 and an AIC of 16468.2; it is not rejected but it is very close to being rejected. DAG (e) results in a null probability of 0.587 and an AIC of 16465.3; furthermore, the null probability of the path coefficient in the newly added edge (hemato→recruited) is 0.023, which is significantly different from zero. DAG (f) results in a null probability of 0.905 and an AIC of 16467.1; the null probability of the path coefficient in the newly added edge (frass→recruited) is 0.61, which is not significantly different from zero. DAG (e) is the preferred model based on the AIC value but the other two cannot be clearly excluded because their AIC values are not 4 units larger than DAG (e).

[256] If you disagree with this, then you could orient the edges otherwise and then test the resulting model.
[257] To me… maybe someone with more knowledge in animal physiology might disagree.

I presented the (slightly modified) causal model that had originally been proposed by the original authors (Thomas et al. 2007) in Figure 2.3 of Chapter 2. The original authors developed their causal model based only on their biological knowledge, with no input from the CI algorithm. You will notice that it is identical to DAG (e) in Figure 9.9. In other words, the CI algorithm correctly recreated the partially oriented dependency graph implied by the causal graph of (Thomas et al. 2007). The additional orientation of just a few edges that were unspecified by the CI algorithm, but that were supported by biological knowledge, completely recovered the original DAG that had been proposed by Thomas et al. (2007).

It is possible to experimentally remove blood from chicks in a randomized controlled experiment. Imagine that we had experimental evidence that a change in haematocrit volume causes a decrease in nestling body mass. In that case, then we could impose this constraint on the CI algorithm as follows:

```
force.edges<-"
XH->XM
"
CI.algorithm(dat=nested_data[,-
c(3,9,10)],family=data.frame(XR="binomial",
      XP="poisson"),nesting=blue.tits.nesting,alpha.reject=0.1,
      constrained.edges=force.edges,smooth=T)
```

The CI algorithm is an intellectually elegant creation that goes part way towards realizing the philosopher's dream of inferring causation from correlation. It is a tool to help you with the process of generating causal hypotheses. However, the output of the CI algorithm can certainly make mistakes, and such mistakes will increase as sample size decreases. It also becomes increasingly less informative when the patterns of dependence and independence among the variables becomes increasingly generated by latent variables that are common causes of several observed variables. You saw this in Figure 7.1, where a DAG with two latents that were each common causes of three observed variables resulted in a mixed acyclic graph in which each observed variable was joined to each of the others. This is a general result: when several observed variables are each causal children of the same latent parent, the partially oriented dependency graph involving these observed variables will have each variable joined to each other variable.

A set of observed variables in which each one is joined to each other one in a partially oriented dependency graph is called a "saturated" pattern. For instance, imagine that data involving variables X1, X2, X3 and X4 are generated by the DAG in Figure 9.10a, where "L" is a latent variable. If you generate data from this causal structure and give them to the CI algorithm, then Figure 9.10c is the result. Since variables X1, X2 and X3 are each joined to the other two, these variables are "saturated" in the partially oriented dependency graph. A saturated pattern is an imperfect clue that a latent variable might be lurking in the background. You should consider the possibility of a common latent cause whenever you see such a saturated pattern. However, a saturated pattern is not proof of the existence of such a latent variable. After all, the DAG in Figure 9.10b would also result in the saturated pattern of Figure 9.10c. As more and more variables become saturated, the resulting partially oriented dependency graph becomes less and less informative because more and more of it contains saturated patterns.

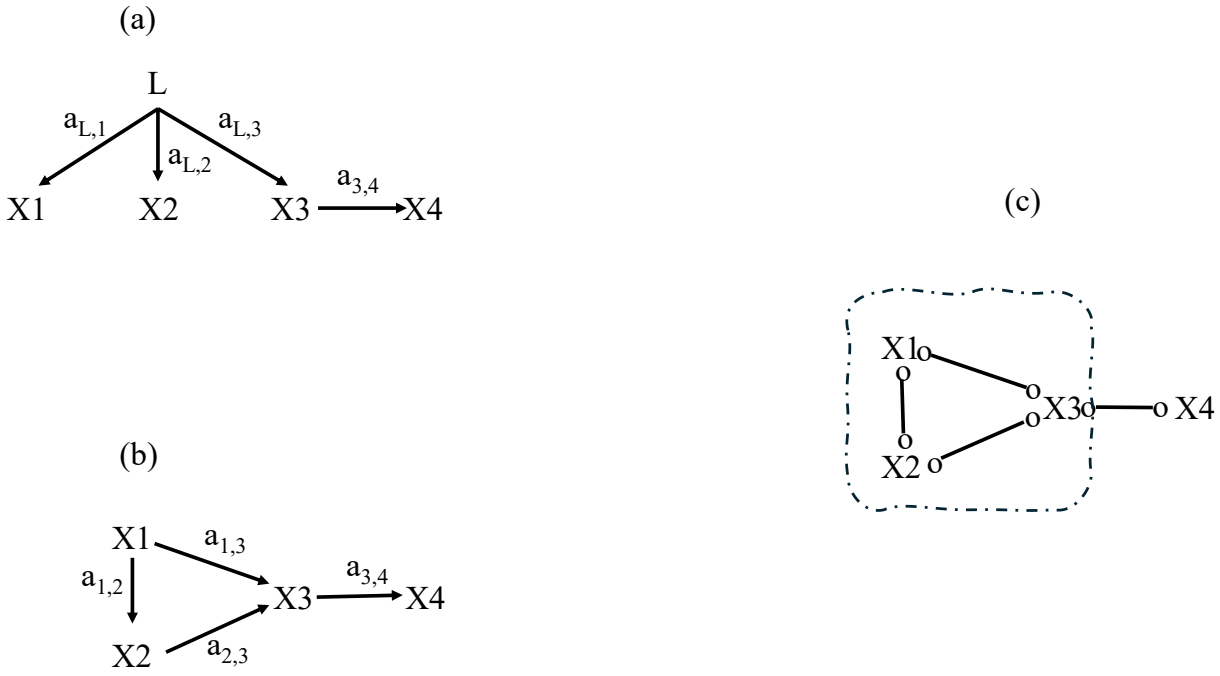


Figure 9.10. Two different DAGs (a, b) that generate the same partially oriented dependency graph (c). Variable L is latent while X1, X2, X3 and X4 are observed. The path coefficients are also shown for the two DAGs. The three observed variables inside the square form a saturated pattern, in which each variable is joined to each other variable.

9.9 Detecting latent variables

Whenever you see a set of observed variables that form a saturated pattern in a partially oriented dependency graph, you should at least consider the possibility that the variables in the saturated pattern are each causal children of one or more latent parents. However, as Figure 9.10(b) illustrates, it is possible for such saturated patterns to arise even without latent variables. Is there any way of differentiating between the two? Yes, in some cases. Theorems 6.10 and 6.11 of Spirtes, Glymour and Scheines (Spirtes et al. 1993) state the conditions, given certain statistical assumptions, under which a saturated pattern in a partially oriented dependency graph necessarily implies the existence of latent variables. These theorems lead to an algorithm based on vanishing tetrad differences that can identify the existence of latent variables. These assumptions[258] are (i) linearity in the functional links between each observed variable in the saturated pattern and the latent variables, (ii) multivariate normality of these observed variables, (iii) large sample size and (iv) causal homogeneity in the generating process. Furthermore, the saturated pattern must involve at least four observed variables. If you only want to use vanishing tetrad differences as a guide when considering latent variables, then go to section 9.10. If you want to understand why vanishing tetrad differences can, in certain situations, identify the existence of latent variables, then read on.

Spearman (Spearman 1904) derived a set of equations called "vanishing tetrads". A vanishing tetrad equation is a function of four correlation (or covariance) coefficients[259] (thus, "tetrads") such that the product of the first two correlation coefficients exactly equals the product of the second two correlation coefficients (thus "vanishing" tetrads). Therefore, the difference between the two products equals zero, or "vanishes". Here are the three possible tetrad equations (t_1, t_2, t_3) for the four observed variables in the DAGs of Figure 9.10, where ρ_{ij} is the Pearson correlation coefficient between variables i and j in the statistical population:

[258] I conjecture that the first three assumptions could be relaxed by using the generalized covariance statistic, rather than Pearson correlations of covariances, in the tetrad equations. Certainly, you can transform your variables to approximate normality, or you can transform them to ranks and use Spearman correlations instead of Pearson correlations.

[259] The requirement of linearity comes from the fact that the tetrad equations involve covariances.

$$\rho_{12} \cdot \rho_{34} - \rho_{14} \cdot \rho_{23} = t_1$$
$$\rho_{13} \cdot \rho_{24} - \rho_{14} \cdot \rho_{23} = t_2$$
$$\rho_{13} \cdot \rho_{24} - \rho_{12} \cdot \rho_{34} = t_3$$

If the four variables that are generated by the DAGs in Figure 9.10 are standardised to each have a mean of zero and a standard deviation of unity, then the total effect between any two variables is the Pearson correlation coefficient[260] between them. You can therefore use the rules for tracing the total effects between variables that were given in section 3.9 of Chapter 3 to get the predicted Pearson correlations. For instance, the correlation between X1 and X2 in Figure 9.10a is $\rho_{12} = a_{L1}a_{L2}$. If we use these rules to get the predicted Pearson correlations between each pair of variables in our tetrad equations, then we get the following values for the DAG in Figure 9.10a; notice that the topology of the DAG with its latent variable requires that the second tetrad equation equal zero, but not the first or third tetrad equations, irrespective of the actual non-zero values of the path coefficients:

$$\left(a_{l1}a_{L2}\right)\left(a_{23} - a_{L2}a_{l3}\right) = t_1$$
$$\left(a_{L2}a_{L3}^2a_{L1}\right)\left(a_{34} - a_{L3}\right) = 0 = t_2$$
$$\left(a_{L1}a_{L2}a_{34}\right)\left(a_{L3}^2 - 1\right) = t_3$$

Given a set of four observed variables (I, J, K, L), theorem 6.11 of Spirtes, Glymour and Scheines (Spirtes et al. 1993)(p. 197) states that the causal generating process in nature linearly implies a tetrad equation of the form $\rho_{IJ}\rho_{KL} - \rho_{IL}\rho_{JK} = 0$ if (i) ρ_{IJ} or $\rho_{KL} = 0$ and ρ_{IL} or $\rho_{JK} = 0$ or (ii) if there is a set **Q** of random variables in the causal generating process such that each pair of variables in the tetrad equation is d-separated by **Q**. Since this theorem only applies to linear relationships, this means that all the following Pearson partial correlations must be zero: $\rho_{IJ.\mathbf{Q}} = \rho_{KL.\mathbf{Q}} = \rho_{IL.\mathbf{Q}} = \rho_{JK.\mathbf{Q}} = 0$.

This theorem allows us to identify the presence of latent variables using these tetrad equations. How? If a set of four or more variables form a saturated pattern in a partially oriented

[260] If the variables are not standardised then the total effect, multiplied by a variance, is the covariance between them.

dependency graph, then we know two things. First, we know that none of the correlations ρ_{IJ}, ρ_{KL}, ρ_{IL}, or ρ_{JK} are zero since, if any were zero, then they would not be joined and would not form a saturated pattern. Second, we know that there is no combination of other variables in the generating DAG that can d-separate any of the four pairs (IJ, KL, IL or JK) since, if any did, then that pair would not be joined and would not form a saturated pattern. However, since the tetrad equation vanishes (i.e. equals zero), the theorem requires that there is some set of variables (**Q**) that d-separates each pair. Since **Q** does not include any of the observed variables, it must include latent variables.

A vanishing tetrad equation can be given a graphical interpretation that helps us to see where latent variables might be lurking. This graphical interpretation also helps us to understand Theorem 6.10, called the Tetrad Representation Theorem of Spirtes, Glymour and Scheines. To understand this, we will need some new definitions. I will use the DAG in Figure 9.11 to explain them.

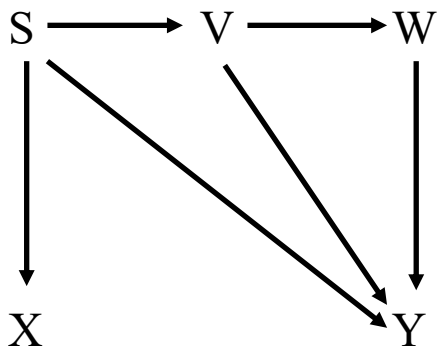


Figure 9.11. A DAG used to explain the concepts of a trek and a choke point.

A *trek* between two variables (X, Y) is a pair of directed paths such that one directed path goes from a *source* variable (S) to X and the other directed path goes from the same source variable S to Y. One of the two directed paths can be of length 0 (i.e. S=X or S=Y). Graphically, a trek has the form: X←…←S→…→Y and if S=X then the trek is X→…→Y. For instance, in Figure 9.11 we see that X←S→V→W→Y is a trek *between* X and Y *from* the source variable S. I will write "T(X, Y)" to mean a trek between X and Y. So, the trek T(X, Y) = X←S→V→W→Y is one of the treks *between* X and Y *from* the source variable S in Figure 9.11 and this trek is composed of two directed paths: X←S and S→V→W→Y. I will write "X(T(X, Y))" to mean the

directed path in a trek between X and Y that goes into X; in Figure 9.11, X(T(X, Y)) = S→X. I will write "**T**(X, Y)" to mean the set of all treks between X and Y and I will write X(**T**(X, Y)) to mean all of the directed paths into X in all of the treks between X and Y. In Figure 9.11 there are 3 different treks between X and Y: X←S→Y, X←S→V→Y and X←S→V→W→Y and so:

T(X, Y) = {X←S→Y, X←S→V→Y, X←S→V→W→Y}

X(**T**(X, Y)) = {X←S}

Y(**T**(X, Y)) = {S→Y, S→V→Y, S→V→W→Y}.

Notice that all the directed paths in all these treks leading into X and into Y pass through S. When this occurs, we say that S is a *choke point* or *choke variable*, for the set of treks, XY(**T**(X, Y)), between X and Y that have a directed path both into X and into Y. Given these definitions, here[261] is the Tetrad Representation Theorem:

Given a set of four variables (X_I, X_J, X_K, X_L), there are six possible unique pairs: (X_I, X_J), (X_I, X_K), (X_I, X_L), (X_J, X_K), (X_J, X_L) and (X_K, X_L). The tetrad equation $\rho_{IJ}\rho_{JK}$-$\rho_{IL}\rho_{JK}$=0 involves four of these six pairs and does not involve two of these six pairs. Identify the two pairs of variables: (X_I, X_K) and (X_J, X_L). Then

(i) every trek between the four pairs of variables in the tetrad equation that have a directed path into X_I and into X_K in these treks passes through the same choke variable

OR

(ii) every trek between the four pairs of variables in the tetrad equation that have a directed path into X_J and into X_L in these treks passes through the same choke variable

The OR is inclusive, not exclusive. In other words, it means that at least one of (i) and (ii) must be true but both can also be true. To see what all this has to do with vanishing tetrads, look again at the DAG in Figure 9.10a, which has a set of four variables (X_1, X_2, X_3, X_4). We can always construct 6 possible pairs from these four variables: (X_1, X_2), (X_1, X_3), (X_1, X_4), (X_2, X_3), (X_2,

[261] This is not a direct quote from page 196 of Spirtes et al. (1993) but is a rewording to make it easier to understand.

X_4) and (X_3, X_4). We already determined that the second tetrad equation must be zero given the topology of the DAG: $\rho_{13} \cdot \rho_{24} - \rho_{14} \cdot \rho_{23} = t_2$. The second tetrad equation only involves (X_1, X_3), (X_2, X_4), (X_1, X_4) and (X_2, X_3), and doesn't involve (X_1, X_2) or (X_3, X_4). Now, find all treks between each pair of variables in this tetrad equation. Then

(i) all directed paths into each of X_1 and X_2 in all of these treks must pass through the same choke variable, OR

(ii) all directed paths into each of X_3 and X_4 in all of these must pass through the same choke variable.

Looking at Figure 9.10a, we see that

(i) all directed paths into X_1 and into X_2 in all treks between X_1 and X_3 (i.e. $X_1 \leftarrow L \rightarrow X_3$), between X_2 and X_4 (i.e. $X_2 \leftarrow L \rightarrow X_4$), between X_1 and X_4 (i.e. $X_1 \leftarrow L \rightarrow X_4$) and between X_2 and X_3 (i.e. $X_2 \leftarrow L \rightarrow X_3$) do indeed pass through the same choke variable (L). As well,

(ii) all directed paths into X_3 and into X_4 in all treks between X_1 and X_3 (i.e. $X_1 \leftarrow L \rightarrow X_3$), between X_2 and X_4 (i.e. $X_2 \leftarrow L \rightarrow X_3 \rightarrow X_4$), between X_1 and X_4 (i.e. $X_1 \leftarrow L \rightarrow X_3 \rightarrow X_4$) and between X_2 and X_3 (i.e. $X_2 \leftarrow L \rightarrow X_3$) also pass through the same choke variable (L).

I have underlined the directed paths in order to make them more obvious to you. In this example, both conditions (i) and (ii) apply, although only one of the two necessarily applies in general. How can vanishing tetrads help to detect the presence of latent variables? If you see a saturated pattern in your undirected dependency graph involving at least three of the four variables in the tetrad then test to see if there are any vanishing tetrads between these four variables. If vanishing tetrads exist, then this is evidence for a latent variable. Why? If the variable that is the choke point implied by this vanishing tetrad was an observed variable, then all directed paths into at least one of the two pairs of variables that are not in the tetrad equation would be d-separated by this choke point and therefore could not form part of the saturated pattern. In Figure 9.10a, variables X_1, X_2 and X_3 only form a saturated pattern in Figure 9.10c because L is latent. This fact provides a simple algorithm, the vanishing tetrad algorithm, to test

whether the observed correlations among a set of four observed variables is due to a common latent cause. These are the four steps.

1. Identify a set of at least[262] four observed variables that either form a saturated pattern or else at least three of them form a saturated pattern in the undirected dependency graph and the fourth is joined to at least one of the other three. The undirected dependency graph in Figure 9.10c is an example.

2. Choose one of the three tetrad equations that are possible given the four chosen variables. This tetrad equation involves Pearson correlation correlations or covariances for four of the six possible pairs that exist given these four chosen variables. Identify the two pairs of variables that are not included in the tetrad equation.

3. If the tetrad equation doesn't equal zero, go to step 1. If you have tried all three then stop.

4. If the tetrad equation does equal zero, then identify the two pairs of variables that are not included in the tetrad equation. There is a latent variable that forms the choke point for all directed paths into both variables in the first pair and/or for all directed paths into both variables in the second pair of variables.

Wishart (1928) derived the asymptotic sampling variance of this statistic in the first part of the twentieth century. The asymptotic sampling variance[263] is:

$$\left(\frac{D_{IK}D_{JL}\left(N+1\right)}{\left(N-1\right)}-D\right)\left(\frac{1}{N-2}\right)$$

Here, D is the determinant of the population correlation or covariance matrix of the four variables, D_{IK} is the determinant of the 2X2 matrix consisting of the population correlation or covariance matrix of variables I and K , D_{JL} is the determinant of the 2X2 matrix consisting of the population correlation or covariance matrix of variables J and L, N is the sample size and the four variables follow a multivariate normal distribution. . If the null hypothesis is true, then the test statistic (τ) of a tetrad equation is asymptotically distributed as a normal variate with a zero

[262] If there are more than four variables forming a saturated pattern, then take each unique set of four variables.
[263] The formula given in Spirtes et al. (1993) is incorrect, but these authors give the correct formula in their earlier book (Glymour et al. 1987).

mean and the given variance. Therefore, the value $z = \tau/\text{var}(\tau)$ asymptotically follows a standard normal distribution.

9.10 Detecting latent variables using the pwSEM package

The `vanishing.tetrads()` function in the pwSEM package implements the Tetrad Representation Theorem. There are four arguments. The first argument, `dat=`, gives the data frame containing only the four or more observed variables satisfying the first step, i.e. that form a saturated pattern. The second argument, `sig=` specifies the significance level to be used when testing the null hypothesis that the tetrad equation equals zero, and defaults to 0.05. If you use only these two arguments, then the resulting null probability (i.e. H_0:τ=0, where τ is the value of the tetrad equation) is an asymptotic probability assuming multivariate normality and a large sample size. The last two arguments are required if you do not want to make these assumptions, although linearity is still required, since these generate an empirical bootstrap sampling distribution (Manly 1997) (`bootstrap=TRUE`) using the number of bootstrap runs given by the argument `B=`. The defaults are `bootstrap=TRUE` and `B=1000`. Increasing the value of B will increase the precision of the bootstrap null probability estimate but will (slightly) increase computing time. I recommend that you always use the bootstrap option.

I generated a large data set (N=500), called `dat`, using the DAG in Figure 9.10a. Imagine that we don't know that the data were generated according to this DAG. What can we learn about the generating process using only the CI algorithm and vanishing tetrads? The first step is to apply the Causal Inference algorithm using `CI.algorithm(dat=dat)` in order to insure that the variables form a saturated pattern. Here is the output:

```
Partially oriented dependency graph:
X1 o--o X2
X1 o--o X3
X2 o--o X3
X3 o--o X4
```

The resulting partially oriented dependency graph is the same as in Figure 9.10c. Since X1, X2 and X3 form a saturated pattern, and since X4 is joined to X3, we next check for vanishing tetrad differences using `vanishing.tetrads(dat,sig=0.05,bootstrap=TRUE,B=1000)`. Here is the output:

```
tetrad: (X1,X2)*(X3,X4)-(X1,X4)*(X2,X3)
Bootstrap probability< 0.001

tetrad: (X1,X3)*(X2,X4)-(X1,X4)*(X2,X3)
All directed paths going into X1 and into X2 OR/AND into X3 and into X4
 in all treks between the variable pairs listed in the tetrad equation pass
through the same (possibly latent) choke variable
Bootstrap probability= 0.108

tetrad: (X1,X3)*(X2,X4)-(X1,X2)*(X3,X4)
Bootstrap probability< 0.001
```

Now, we know the partially oriented dependency graph (Figure 9.10a) and we know that only the second tetrad equation vanishes. Therefore, we know that all of the directed paths going into X1 and into X2 pass through the same latent choke point and/or that all of the directed paths going into X3 and into X4 also pass through this same latent choke point. This information is enough to deduce the DAG in Figure 9.10a. However, this is not always (or even commonly) the case.

One exception is when all three tetrad equations vanish because, when this happens, this implies a DAG that takes the form of a measurement model (section 7.2 of Chapter 7) in which none of the four observed variables are causes of any other, but all are direct common effects of a single latent cause. When a saturated pattern involving four observed variables occurs, but not all three of the tetrad equations vanish, then the only thing that can be deduced is that there are one or more latent variables lurking as choke variables in the set of four observed variables, but that the causal generating process is not in the form of a measurement model with a single latent. For example, Figure 9.12 shows three different generating DAGs that all result in a saturated partially oriented dependency graph and in which only the second tetrad equation vanishes.

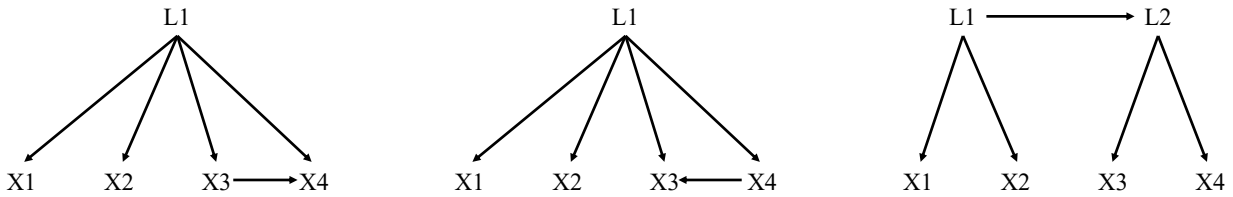


Figure 9.12. The different DAGs involving latent variables (L1 and L2) that each have the same partially oriented dependency graph, in which the four observed variables (X1, X2, X3, X4) form a saturated pattern, and in which only the second tetrad equation vanishes.

I am aware of only one empirical example of the use of vanishing tetrads in biology as an aide in exploratory causal analysis. One of the most influential papers in plant functional ecology (Wright et al. 2004) assembled all available empirical data[264] on a set of interspecific leaf traits related to the morphology and physiology of leaves and conducted a principal components analysis. Four key traits lay at the heart of this data set. Leaf mass per area (LMA) is the ratio of the dry mass of the leaf divided by the projected area of a leaf. This morphological variable is mostly determined by the shape of the leaf, the proportions of different tissue types and the pattern and thickness of the veins. Leaf nitrogen content (N_{mass}) per mass is the amount of leaf organic nitrogen divided by its dry mass. Nitrogen is a key component of proteins, including the RUBISCO[265] enzyme that powers photosynthesis. Instantaneous leaf photosynthetic rate per mass (or "assimilation rate", A_{mass}) is the net amount of atmospheric CO_2 that is fixed per unit time and per dry mass of the leaf. Finally, leaf lifespan (LL) is the average lifespan of a leaf from the time it is first fully formed until it is dropped by the plant. The data set consisted of 492 complete lines involving these four variables and each line represented one species. The key result of Wright et al. (2004) was that a single principal component could capture a large (~75%) percentage of the total variation across vascular plant species worldwide, and that this pattern did not seem to be greatly affected by climate, geographical location or plant type (angiosperms/gymnosperms, deciduous/evergreen, trees/shrubs/herbs). Although the Wright et al. (2004) paper was primarily an empirical description of covariation, the authors speculated about the causes of this pattern in the Discussion section of the paper.

[264] At the time of publication. An order of magnitude more data on several of these traits can now be found in the TRY data base (TRY ref).
[265] Ribulose-1,5-bisphosphate carboxylase/oxygenase

Shipley et al. (2006) converted these speculated causes into a DAG and tested it. The DAG was clearly rejected. A few years earlier, Martin Lechowicz and I had published an alternative DAG involving these same variables[266] (Shipley and Lechowicz 2000) and, confident that our DAG would be supported, I also tested this DAG against the data in Wright et al. (2004). It was just as clearly rejected. Since we had now run out of causal hypotheses, we turned to the Causal Inference algorithm. Here is a recreation of that analysis using the data in a data frame[267] called "wright":

```
CI.algorithm(wright,alpha.reject=0.05)
Partially oriented dependency graph:
logLL o--o logLMA
logLL o--o logNmass
logLL o--o logAmass
logLMA o--o logNmass
logLMA o--o logAmass
logNmass o--o logAmass
```

The result was a partially oriented dependency graph with a saturated pattern involving all four variables. In fact, this saturated pattern persisted until the rejection level (`alpha.reject`) was reduced to 0.001. Such a saturated pattern is an imperfect clue that latent variables might by common causes of these observed variables, and so we applied the method of vanishing tetrad differences. Here is the result, using 10000 bootstrap runs:

```
vanishing.tetrads(dat=wright,sig=0.05,bootstrap=TRUE,B=10000)
tetrad: (logLL,logLMA)*(logNmass,logAmass)-(logLL,logAmass)*(logLMA,logNmass)
Bootstrap probability= 2e-04

tetrad: (logLL,logNmass)*(logLMA,logAmass)-(logLL,logAmass)*(logLMA,logNmass)
Bootstrap probability= 0.0268

tetrad: (logLL,logNmass)*(logLMA,logAmass)-(logLL,logLMA)*(logNmass,logAmass)
All directed paths going into logLL and into logAmass OR/AND into logNmass
and into logLMA
 in all treks between the variable pairs listed in the tetrad equation pass
through
 the same (possibly latent) choke variable
Bootstrap probability= 0.1598
```

Only one the three tetrad equations vanish. This means that there is not a single latent common direct cause of all four observed variables (i.e. a measurement model in which the four observed

[266] The analysis was based on only a small data set (40 species) involving only wetland herbs.
[267] The data set of Wright et al 2004 was published as an online appendix.

variables are the observed indicator variables of this common latent cause), but there is a latent variable involved. This information alone is not sufficient to deduce the likely DAG with its latent variable(s) and so we can start with the measurement model and then use the method of modification indices in lavaan to help us in exploring these data.

```
mesurement.mod<-"
L=~logLMA+logLL+logAmass+logNmass
"
fit<-sem(mesurement.mod,data=wright)
summary(fit)
modindices(fit,sort.=T)
       lhs op      rhs     mi    epc sepc.lv sepc.all sepc.nox
13    logLL ~~ logAmass 13.519 -0.013  -0.013   -0.565   -0.565
12   logLMA ~~ logNmass 13.519 -0.004  -0.004   -0.204   -0.204
10   logLMA ~~    logLL  8.113 -0.007  -0.007   -0.215   -0.215
15 logAmass ~~ logNmass  8.113 -0.004  -0.004   -0.256   -0.256
14    logLL ~~ logNmass  0.191  0.001   0.001    0.029    0.029
11   logLMA ~~ logAmass  0.191  0.001   0.001    0.046    0.046
```

As expected, given the vanishing tetrads, the measurement model version (measurement.mod) is rejected (X^2=15.112, 2 df, p=0.001). There are two modifications of this basic measurement model that would greatly improve the statistical fit. First (logLL~~logAmass), we can introduce a direct dependency between the leaf lifespan (logLL) and the photosynthetic rate per mass (logAmass). Second (logLMA~~logNmass), we can introduce a direct dependency between the leaf mass per area (logLMA) and leaf nitrogen content per mass (logNmass). The first suggested modification has a biological justification. Kikuzawa (1991) published a theoretical model predicting the optimal leaf lifespan of a leaf that links a decreasing leaf lifespan with an increasing maximum photosynthetic rate; thus Amass→LL. The second suggested modification also has a biological justification. One way to decrease leaf mass per area is to increase the thickness of the leaf lamina by increasing the number of layers of spongy and palisade mesophyll cells. These mesophyll cells are like water-filled balloons with relatively little dry mass and these cells are the main ones that contain the chloroplasts and photosynthetic enzymes (and thus leaf nitrogen). Therefore, decreasing leaf mass per area (LMA) means increasing the number of mesophyll cells, and so the number of nitrogen-rich photosynthetic enzymes per unit leaf area. Thus, LMA→Nmass. If we test each of these suggested modifications, both result in well-fitting models with exactly the same null probability (X^2=1.987, 1df, p=0.159).

The result is two alternative models with a single latent variable plus a direct causal effect that cannot be rejected. There are plausible biological justifications for each of the direct causal effects. What about the latent variable itself? Although the SEM requires this latent variable, nothing in the statistics, or in the Causal Inference and Vanishing Tetrad algorithms, tells us what this latent variable might be. In Shipley et al. (2006), we hypothesised that this latent variable represents the ratio of the volume of the leaf cells occupied by cytoplasm (the liquid in the cells in which the photosynthetic machinery resides) to the volume of the leaf cells occupied by cell walls (where most of the dry mass resides). We favoured the added $A_{mass} \rightarrow LL$ link in Shipley et al (2006) because it had a pre-existing theoretical justification, but the alternative model should also be considered a viable contender.

Note the exploratory status of this analysis. Its purpose was to identify causal hypotheses that make biological sense, and which are supported by the available data, not to test pre-existing causal hypotheses. This is the real strength of these discovery algorithms. Unless pre-existing theory is already quite solid, then proposing a complete causal model from such theory often degenerates into asking: "If I were God, and the world was a machine, then how would I construct it"? Since none of us are gods and the world is not really a machine, such "hypothesis generation" can easily mask unbridled speculation. The discovery algorithms first show us the correlational shadows in our data in the form of a partially oriented dependency graph. This constrains our speculation, forces us to consider different alternative models, and also forces us to explicitly justify any causal process that appears to contradict what the data seem to say.

The development of discovery algorithms for causal analysis is an active field of research. For instance Richardson (1996) has developed an extension of the CI algorithm that can discover causal graphs that include cyclic relationships, but only given the assumptions that all variables are continuous and that all relationships are linear[268]. There are also score-based algorithms for causal discovery. Score-based methods use score criteria that capture the degree to which the causal graph fits the patterns of conditional independence in the data, rather than using significance levels to distinguish between conditional independence and dependence. One early method is called "greedy equivalence search" (Chickering 2002) but there have been many

[268] The operation of d-separation applies to causal graph when all variables are continuous and all relationships between the variables are linear. However, there can exist conditional independencies in cyclic graphs that are not captured by d-separation when nonlinear relationships or discrete variables are involved.

additional advances since that first publication. Since this book is meant to be a user's guide rather than a comprehensive review, and since I have already exceeded the page length set by my editor, I won't go into these other methods here, but a good review is Glymour et al. (2019).

9.11 In conclusion…

U.S. Supreme Court Justice Potter Stewart was charged with ruling on whether a particular film was obscene and thus not protected by the First Amendment of the US constitution in the 1964 case *Jacobellis v. Ohio*. He is reported to have said that "*I shall not today attempt further to define the kinds of material I understand to be embraced within that shorthand description [i.e. hard-core pornography]; and perhaps I could never succeed in intelligibly doing so. But I know it when I see it, and the motion picture involved in this case is not that*." Causal analysis is the statistical analogue of hard-core pornography. Philosophers have argued over definitions of "causality" without resolution and many statistics texts implicitly (often explicitly) warn the reader against attaching conclusions of cause-and-effect to the results of statistical analysis. As summarised in Chapter 3, influential statisticians like Pearson and Fisher explicitly denied the possibility of inferring causality without evidence from randomized experiments (Pearson went further by denying the concept of causality completely). Scientists cannot yet convincingly and intelligibly define "causality". Yet, I find that if I don't ask for a definition of causality, most scientists "know it when they see it". Given the dim view of causality that is adopted by most empiricists, it is ironic that the approach to causality taken in this book is almost… *empirical*. Rather than defining causality, one looks for those properties of relationships that scientists have deemed to call "causal" and then develop a mathematical language that possesses such properties. In time perhaps this will lead to a comprehensive definition that can be accepted by everyone. For myself, I view "causality" as a relationship between events or classes of events (i.e. variables) that possesses the properties of asymmetry, transitivity and the Markovian condition[269]. I expect that as our mathematical language of

[269] I know that in Chapter 1 I promised not to give a definition of causality. I couldn't resist the temptation; I'm an academic and academics are drawn to definitions like young children are drawn to puddles. We like to jump in and stir up the mud.

causality improves, we will be able to better express our scientific notions of causality using mathematics and this should lead to both better tests of causal hypotheses and better discovery algorithms.

The various methods in this book all attempt to test, or detect, causal relationships using observational data. I have not intended my book to be an encyclopaedic treatment of the relationship between cause and correlation – there is certainly much more to be said – but I hope that it will be useful as you watch the correlational shadows dance across the screen of Nature's Shadow Play.

Enjoy the show.

10

A cheat sheet of important R functions

10.1 The ggm package

The ggm package on CRAN (Marchetti et al. 2024) requires the graph package. However, the graph package is now on BiocManager, rather than on CRAN. You must install it first before installing the ggm library. Here is how to install the graph package:

```
install.packages("devtools")
library(devtools)
install.packages("BiocManager")
BiocManager::install("graph")
```

`drawGraph(amat)`

is a simple function that draws the graph that is specified by `amat`, where `amat` is a matrix. Another function, `plotGraph(amat, tcltk=TRUE)`, allows for a more interactive plotting of graphs.

.

`DAG(…, order=FALSE)`

Constructs a directed acyclic graph. This then can be plotted using `drawGraph(amat)` or `plotGraph(amat)`. An example is `my.dag<-DAG(X2~X1,X3~X2, order=FALSE)`. The first argument is a sequence of model formulae in which the left hand side (eg. X2) is the causal child and the right hand side are the causal parents in the DAG. An exogenous variable that does not cause anything (X) is entered as `X~1`. The second argument is optional, and the default is FALSE. Specifying `order=TRUE` then the nodes (variables) are permuted according to topological order. The output is a binary matrix in which element a_{ij} is 1 if variable i is the causal parent of variable j.

```
my.mixed.graph<-makeMG(dg= ,ug=, bd= )
```

Constructs a mixed acyclic graph that is more general than the mixed graphs in this book. These can then be plotted using the `drawGraph()` or `plotGraph()` functions. An example is `my.mixed.graph<-makeMG(dg=DAG(Y~X),bg=UG(~Y*Z+Z*W))`. The first argument defines the directed part (Y←X) of the graph, using the `DAG()` function. The second argument defines the undirected part of the graph (X─Y); undirected edges define symmetrical links between variables. The third argument defines the bidirected (X↔Y). The second and third arguments are specified using the `UG()` function. The arguments in the `UG()` function specify which pairs of variables have a bidirected or undirected edge between them. For example, creates bidirected edges Y↔Z and Z↔W.

```
basiSet(amat)
```

Give the union basis set of the DAG that is specified in the `amat` binary matrix (usually created using the `DAG()` function).

```
dSep(amat=, first=, second=, cond=)
```

Returns a logical value stating if the first and second variables are d-separated given the set of variables in the vector of conditioning variables, given the DAG specified in the `amat` binary matrix (usually created using the `DAG()` function). The arguments `first`, `second` and `cond` are character names that must be row and column names in `amat`. An example is `dSep(my.dag, first= "X", second= "Z", cond= "c("Z")).`

10.2 The piecewiseSEM package

This library is on the CRAN site of R. There are two main functions in the piecewiseSEM library (Lefcheck 2016): `psem()` and `summary.psem()`. The first step is to create an object containing the structural equations, although these structural equations can also be directly input into `psem()`. These can be of classes `lm, glm, gls, pgls, Sarlm, lme, glmmPQL, lmerMod, lmerModLmerTest, glmerMod, glmmTMB,` and `gam`. You can include whatever arguments are needed within each class of models. The second step is to

pass this list to the function `psem()`. The third step is to extract the results using the `summary()` function. Here is an example:

Step 1: call `psem()` using the `lm()` function to fit normally distributed linear regressions for all of the structural equations, using the data in the data frame `dat`.

```
fit<-psem(
    lm(X2~X1,data=dat),
    lm(X3~X2,data=dat),
    lm(X4~X2,data=dat),
    lm(X5~X3+X4,data=dat)
)
```

Note that the `psem()` function includes a special operator (`%~~%`) that is supposed to model dependencies arising from a marginalized implicit latent common cause of two observed variables. However, the way that `psem()` deals with such dependencies is wrong (see Chapter 6) and results in an incorrect union basis set, C-statistic and its null probability[270]. Only true DAGs, not causal graphs that include correlated errors or free covariances, should be modelled using the piecewiseSEM package.

Step 2: extract the results using the `summary()` function. I recommend using the conserve=TRUE argument when generalised linear or mixed model regressions are used. The argument conditioning=TRUE will print out the full set of conditioning variables in the d-separation claims. If there is more than one exogenous variable, and you want to include independence of any (or all) pairs of exogenous variables, then you must calculate the p-value of each such independence claim, combine these together as a numerical vector (e.g. `exogenous=c(0.12, 0.31, 0.044)`), and add the argument `add.claims=exogenous` to the summary function.

```
summary(fit,conserve=TRUE,conditioning=TRUE)
```

Other useful optional arguments to the `summary.psem()` function are `standardize=`, which specifies if standardised coefficients are to be output, `standardize.type=` that specifies the type of standardisation, and `intercepts=`, which specifies if intercepts should be

[270] Version 2.3.0

output. Note that the `AIC.type=` argument does not work[271]; only the loglikelihood AIC is calculated.

10.3 The pwSEM package

The pwSEM package is first documented in this third edition. It is available on CRAN. Note that this package uses the ggm package, and the ggm package requires that you install the graph package from the BiocManager site (see the section on the ggm package). The main function is `pwSEM()`, which performs piecewise SEM based on either DAGs or MAGs. Additional functions are included to allow you to do some of the steps involved in piecewise SEM. The function `DAG.to.MAG.in.pwSEM()` converts a DAG involving latents (either marginalised or conditioned) into a d-separation equivalent MAG. The function `MAG.to.DAG.in.pwSEM()` converts a MAG into a DAG involving latents. The function `generalized.covariance()` calculates the generalised covariance function and its asymptotic null probability while the `perm.generalized.covarince()` function does the same thing while calculating the non-parametric permutation probability. The `view.paths()` function plots direct and indirect path effects between pairs of variables in their original scales; this is useful when nonlinear structural relationships are modelled. The `basiSet.MAG()` function outputs the union basis set given a mixed acyclic graph (MAG). The `get.AIC()` function outputs the Akaike Information Criterion (AIC) value of a set of structural equations based on either a DAG or a MAG with correlated errors arising from implicit marginalized latent variables. The `MCX2()` function calculates a Monte Carlo null probability for the maximum likelihood chi-squared statistic. Finally, there are two functions for performing exploratory analyses: `CI.algorithm()` and `vanishing.tetrads()`.
Some technical details:

The null probabilities associated with each of the d-separation claims in the union basis set of your causal model are based on the generalized covariance statistic. Given a d-separation claim: X1 _||_X2|{**C**}, where **C** is the set of causal parents of either X1 or X2, the generalized

[271] Version 2.3.0

covariance statistic consists of conducting two regressions, X1~C and X2~C, getting the "response" residuals (which are not the default type of residuals for gam or gamm4) from these two regressions, and then inputting these two vectors of residuals to calculate the generalized covariance statistic. These regressions can be of any type that are appropriate given the nature of the dependent variables (X1, X2) and the assumed functional form linking them to the causal parents, including mixed, generalized, generalized mixed, additive, generalized additive or generalized additive mixed regressions. Note that mixed model regressions can only include random intercepts in this version of the pwSEM package. You should not use `pwSEM()`, `get.AIC()`,`CI.algorithm()` or `vanishing.tetrads()` in a mixed model context if you think that the values of the path coefficients (i.e. partial slopes) differ greatly between levels of the random components. The only solution right now is to do the d-separation tests via the `generalized.covariance()` function and then estimate the path coefficients yourself.

Although `pwSEM()` allows for generalized linear or mixed generalized linear fits when testing the d-separation claims, it can also accommodate generalized additive or mixed generalized additive fits when testing the d-separation claims via the "`do.smooth=TRUE`" argument. You should do this when you believe that the relationships between the variables are nonlinear beyond what is expected (i.e. beyond an exponential for Poisson variables or a logistic for Binomial variables). However, `pwSEM()` uses the default values for smoother terms and it is possible that the default maximum number of knots (i.e. 10) might not be high enough for very nonlinear functions or not enough for very small data sets; in this case an error message will be produced. The only solution right now is to do the d-separation tests outside of pwSEM via the `generalized.covariance()` function.

If all of the variables in your SEM are normally distributed, then you can also obtain the fits based on standardized variables, resulting in standardized path coefficients. If any of your variables assume a non-normal distribution, then standardized fits are not returned. Optionally, the dependent errors ("free covariances") between pairs of observed variables that are the causal children of implicitly marginalised latents or causal parents of implicitly conditioned latents are calculated. This is either a covariance and Pearson correlation for normally distributed variables, or a Spearman correlation otherwise; these are based on the "response" residuals' i.e. based on

the actual (observed - predicted) values of the original variables. The maximum likelihood AIC statistic is calculated using the structural equations specified in the equivalent MAG if the equivalent MAG differs from your original causal model. The AIC statistic is given in (Shipley and Douma 2020a). If there are correlated errors in your model, then the likelihood of the correlated errors is based on a normal copula function. Mixed models involving beta-distributed variables in the structural equations are not currently supported

There are three main steps required to fit and output the results of a piecewise SEM using `pwSEM()`:

1. Create a list containing all of the structural equations, following your causal hypothesis (DAG or MAG). Any dependent errors or selection bias are not included in this first step. Note that you must also include the structural equations for all of the exogenous variables, i.e. X~1. People often forget this and, if you do, it will result in an error message**.** These structural equations are constructed using either the `gam()` or the `gamm4()` functions of the mgcv package; see (Wood 2017).

2. Create output from the `pwSEM()` function by inputting (1) your list (from step 1), (2) a list of pairs of variables that are common causal children of implicitly marginalised latents (i.e. dependent errors, if any), (3) a list of pairs of variables that are common causal parents of implicitly conditioned latents (i.e. selection bias, if any), (4) the data set that will be used to fit the data (this must be the same as given inside the `gam` or `gamm4` calls from step 1), (5) whether you want to use asymptotic probabilities of the d-separation claims or permutation probabilities (for small sample sizes of approximately <100), and (6) how many permutations you require (defaults to 5000). Note that permutation probabilities will slow down the `pwSEM()` function.

3. The output from step 2 contains many objects resulting from the fit, but it is more convenient to use the `summary()` function to output the results. Use the argument "`structural.equations=TRUE`" if you also want the fits of the structural equations to be output.

There is also one optional step if you want to visually view the effects of variables along different paths. If all of your variables are normally distributed (i.e. family=gaussian) and the relationships between the variables are all linear, then you can easily obtain the effects of variables along different paths using the basic rules of combining path coefficients: multiply the path coefficients (i.e. slopes of the regressions) along a directed path. For instance, if you have X-(a)->Y-(b)->Z, where a and b are the slopes of the two regressions (Y~X and Z~Y) then the effect of X on Z along this path is a*b. However, if you have allowed the relationships between the variables to be potentially nonlinear via a smoother term in `gam` or `gamm4` (for example, Y~s(X)) then the relationship between the variables is nonlinear and the effect (i.e. the 1st derivative of the function) is not constant. You can't talk about a path "coefficient" because the slope (the 1st derivative) changes with the values of the independent variable. Note also that if any of the endogenous variables along the path are non-normal (for example, using `family=poisson`) then the effect (the slope) is constant when the dependent variable is transformed by its link function (a ln(Y) transform if family=poisson), the effect is not constant if you consider the dependent variable in its original scale. Therefore, to view the relationship between two variables along any path in the DAG, you can use the `view.paths()` function in this package, which will produce a graph showing the relationship between the two variables and a graph showing the (approximate) effect for different values of the independent variable.

Modelling the structural equations in the pwSEM package uses the mgcv package; see (Wood 2017) for complete details of the practice and theory of this package. The use of the mgcv package, and the two model functions "`gam`" and "`gamm4`", allow for a very wide range of models. If you want linear models, then use "gam" or "gamm4" without any smoother terms. For example, `gam(Y~X,…)` fits a linear relationship between Y and X while `gam(Y~s(X),…)` will fit a smoother function that is not linear unless the underlying relationship is truly linear (and you have lots of data). The smoother operator "`s()`" can also be used in `gamm4`. If you specify distributions other than "gaussian" (i.e. normal) in the `family=` argument, then you will model non-normal data. If you want to include a random component to the model (for example, to account for nested data), then use the `gamm4` function. Note that not all of the functionality of these two functions is accepted in this version of pwSEM. For instance, you can somewhat control the degree of nonlinearity ("wigglyness") of smoother splines via `s(X, k=)` or other

types of smoother splines, but `pwSEM` cannot accommodate this; instead, it uses the default choice. If your SEMs are sufficiently complicated to require this, then it is best to do this individually rather than via `pwSEM`. Certainly, more complicated modelling of generalized linear and generalized additive models, with or without a random component (i.e. mixed models) requires that you have a good knowledge of this field! For instance, fitting models with a Poisson distribution (i.e. `family=poisson`) can run into problems when there are lots of zero values even though the model might converge without throwing out any error or warning messages. You could try a zero-inflated version, but this is not something for beginners.

Here is the pwSEM function:

```
pwSEM(sem.functions,marginalized.latents = NULL,
conditioned.latents = NULL, data,use.permutations =
FALSE,n.perms = 5000,do.smooth = FALSE,all.grouping.vars = NULL)
```

Arguments

`sem.functions`: A list giving the gamm4 (gamm4 package) or gam (mgcv package) models associated with each variable in the structural equation model, INCLUDING exogenous variables.

`marginalized.latents`: A list giving any dependent errors between pairs of observed variables that are generated by common marginalised latents ("free covariance" in covariance-based SEM). Each element of this list is a pair of variables whose errors are hypothesized to be dependent (i.e. they have some unknown common cause) separated by two tildes (~). For example: `list(X~~Y)`.

`conditioned.latents`: A list giving any dependent errors between pairs of observed variables that are generated by being common causal parents of common conditioned latents ("selection bias"). Each element of this list is a pair of variables whose errors are hypothesised to be dependent (i.e. they have some unknown common cause) separated by two tildes (~). For example: `list(X~~Y)`.

`data`: A data frame containing the empirical data.

`use.permutations`: A logical value (TRUE, FALSE) indicating if you want to use permutation probabilities for the d-separation tests. Defaults to FALSE. You should use TRUE for smaller data sets.

`n.perms`: The number of permutation runs to use for permutation probabilities. Defaults to 5000.

`do.smooth`: A logical value indicating if you want to use regression smoothers (generalized additive models) for the dsep tests. Defaults to FALSE. TRUE will fit nonlinear (regression smoothers) when evaluating the d-separation claims, but this will slow down the function.

`all.grouping.vars`: A character vector giving the names of all variables involved in the SEM functions that define groups for random effects. NULL if there is no random component to any of the variables.

This function returns a list containing the following elements: causal.graph, dsep.equivalent.causal.graph, basis.set, dsep.probs, sem.functions,C.statistic, prob.C.statistic, AIC, n.data.lines, use.permutations, n.perms.

Consider a causal hypothesis in which X1–>X2–>X3–>X4, with dependent errors between X2 and X4 due to a common implicitly marginalised latent cause, where X2, X3 and X4 are distributed following a Poisson distribution, and we want to obtain a permutation probability for the C-statistic:

```
my.list<-
list(mgcv::gam(X1~1,data=sim_poisson.no.nesting,family=gaussian)
,
mgcv::gam(X2~X1,data=sim_poisson.no.nesting,family=poisson),
mgcv::gam(X3~X2,data=sim_poisson.no.nesting,family=poisson),
mgcv::gam(X4~X3,data=sim_poisson.no.nesting,family=poisson))
out<-pwSEM(sem.functions=my.list,
```

```
marginalized.latents=list(X4~~X2),data=sim_poisson.no.nesting,us
e.permutations = TRUE,n.perms=10000)
summary(out,structural.equations=TRUE)
```

This example uses generalized linear mixed models via the `gamm4()` function of the mgcv package. In this case, your data set must include variables that give the random components of the model, thus the nesting structure. In these data, there are two random components, between groups and within groups, and there is a variable in the data set called "groups" that gives the group to which each observation belongs. You must also now include the "`all.grouping.vars=`" argument, giving all of the variables that define the random component (since different variables might have different random components). Furthermore, in this example, we allow potentially nonlinear functions, via generalized additive mixed models, to test the d-separation claims via the "`do.smooth=TRUE`" argument.

```
my.list<-
list(gamm4::gamm4(X1~1,random=~(1|group),data=sim_normal.with.ne
sting,family=gaussian),
gamm4::gamm4(X2~X1,random=~(1|group),data=sim_normal.with.nestin
g,family=gaussian),gamm4::gamm4(X3~X2,random=~(1|group),data=sim
_normal.with.nesting,family=gaussian),
gamm4::gamm4(X4~X3,random=~(1|group),data=sim_normal.with.nestin
g,family=gaussian))

out<-pwSEM(sem.functions=my.list,
marginalized.latents=list(X4~~X2),
data=sim_normal.with.nesting,use.permutations = TRUE,
do.smooth=TRUE,all.grouping.vars=c("group"))
summary(out,structural.equations=TRUE)
```

The generalized covariance function and its permutation version

The pwSEM package also includes two functions to calculate the generalized covariance statistic of and its null probability. The first is called "generalized.covariance" and the second is called "perm.generalized covariance". The first one, `generalized.covariance()`, produces asymptotic null probabilities based on a standard normal distribution, and is appropriate for larger sample sizes. How large? Simulations suggest that you need at least 100 observations, but these simulations are not exhaustive. However, the permutation version is quite fast and so you

can instead use the permutation version if you are in doubt. The second one, `perm.generalized.covariance()`, produces an empirical permutation distribution of the generalized covariance statistic rather than assuming a standard normal distribution. The default number of permutations is 5000, and this should be fine for most situations, but you can change this via the `nperm=` argument. The larger the number of permutations, the more precise the probability estimate, but also the longer it takes. If we again use the sim_poisson.no.nesting data set, and want to test the conditional independence of X1 and X3, given X2 then here is how to do it.

1. Get the two vectors of residuals from X1~X2 and X3~X2.
2. Call the generalized covariance function.

```
R1<-residuals(mgcv::gam(X1~X2,data=sim_poisson.no.nesting,
family=gaussian))
R2<-residuals(mgcv::gam(X3~X2,data=sim_poisson.no.nesting,
family=poisson))
generalized.covariance(R1,R2)
```

If you want to use the permutation version:

```
R1<-residuals(mgcv::gam(X1~X2,data=sim_poisson.no.nesting,
family=gaussian))
R2<-residuals(mgcv::gam(X3~X2,data=sim_poisson.no.nesting,
family=poisson))
perm.generalized.covariance(R1,R2,nperm=5000)
```

The view.paths function to graphically visualize how indirect effects

This function, usually called after pwSEM, allows you to visually see how one variable causes another in the DAG along all directed paths from the first to the second and to see how the first derivative (a path "coefficient" even though it is not necessarily constant) of this relationship changes as the "from" variable changes. For linear relationships, this is a constant (the path coefficient). Here is the `view.paths()` function:

```
view.paths(from,to,sem.functions,data,minimum.x=NULL,maximum.x=
NULL,scale="response",return.values=FALSE,dag)
```

You would normally call this function after testing, and fitting, your structural equations model using `pwSEM()`. Here is an example to graph the indirect effect of X1 on X4. The graph shows the entire range of the cause if the `minimum.x` and `maximum.x` arguments are omitted; otherwise, these arguments specify the minimum and maximum values that you want to see.

```
# DAG: X1->X2->X3->X4 and X2<->X4
my.list<-
list(mgcv::gam(X1~1,data=sim_poisson.no.nesting,family=gaussian)
,
mgcv::gam(X2~X1,data=sim_poisson.no.nesting,family=poisson),
mgcv::gam(X3~X2,data=sim_poisson.no.nesting,family=poisson),
mgcv::gam(X4~X3,data=sim_poisson.no.nesting,family=poisson))
 out<-
pwSEM(sem.functions=my.list,marginalized.latents=list(X4~~X2),
data=sim_poisson.no.nesting,use.permutations = TRUE,
n.perms=10000)

view.paths(from="X1",to="X4",sem.functions=out$sem.functions,dat
a=sim_poisson.no.nesting,scale="response",dag=out$causal.graph)
```

The MCX2 function

The maximum likelihood chi-square statistic that is commonly calculated in covariance-based structural equations modelling only asymptotically follows a theoretical chi-squared distribution. With small sample sizes it can deviate enough from the theoretical distribution to make it problematic. This function estimates the empirical probability distribution of the maximum likelihood chi-square statistic (output, for instance, from the lavaan package), given a fixed sample size and degrees of freedom (output, for instance, from the lavaan package), and outputs the estimated null probability given this sample size and degrees of freedom. Here is the `MCX2()` function:

```
MCX2(model.df, n.obs, model.chi.square, n.sim =
10000,plot.result=FALSE)
```

The `n.sim` argument gives the number of Monte Carlo simulations requested, and 10000 are usually enough. The `plot.result=TRUE` argument will produce a plot showing the Monte Carlo estimate of the sampling distribution along with the theoreticalcChi-squared distribution. For instance, if lavaan outputs a X2 value of 9.42 with 4 degrees of freedom, and you only have 15 observations in your data set, then lavaan will output an asymptotic null probability of 0.049.

However, this null probability is biased with small sample sizes. Here is how to get a better estimate of the null probability, where MCprobability is the Monte Carlo probability estimate:

```
MCX2(model.df=4,n.obs=15,model.chi.square=9.42)
```

The DAG.to.MAG.in.pwSEM function

Given a DAG containing either marginalised or conditioned latents, this function outputs the d-separation[272] equivalent DAG. Here is the function:

```
DAG.to.MAG.in.pwSEM(full.DAG, latents = NA,
conditioning.latents=NULL)
```

The first argument (`full.DAG`) is a matrix holding the DAG; this is usually produced via the `DAG()` function of the ggm package. The second argument (`latents`) is a character vector holding the variable names in the full DAG that are latent. The third argument (`conditioning.latents`) is a character vector holing the variable names of those latents that are conditioned, rather than marginalised. Here is an example:

```
full.dag<-DAG(X2~X1+L,X3~X2,X4~X3+L)
DAG.to.MAG.in.pwSEM(full.DAG=full.dag,latents=c("L"),conditionin
g.latents = NULL)
```

The MAG.to.DAG.in.pwSEM function

Given a MAG, this function produces a DAG. Here is the basic call structure:

```
MAG.to.DAG.in.pwSEM(MAG,marginalized.latents,conditioned.latents
)
```

MAG is a matrix, usually produced using the makeMG function of the ggm package, that potentially includes implicit marginalised (↔) or conditioned (—) latents. The `marginalized.latents` argument is a list, for example `list(X2~~X3,X4~~X5)` that lists the pairs of observed variables in the MAG that have a marginalised latent as a common causal parent. The `conditioned.latents` argument is a list, for example

[272] Or m-separation equivalent

`list(X2~~X3,X4~~X5)` that lists the pairs of observed variables in the MAG that have a conditioned latent as a common causal child.

The basiSet.MAG function

This function outputs the union basis set of a MAG (mixed acyclic graph) involving either directed edges (X->Y) if X is a direct cause of Y or bi-directed edges (X<->Y) if X is not a cause of Y, Y is not a cause of X, but both X and Y share a common latent cause. It is easiest to create the MAG using the `DAG()` function of the ggm library and then modifying the binary output matrix by adding a value of 100 for each pair (row & column) of variables with a bi-directed edge. Alternatively, a MAG with bi-directed edges can be created using the `makeMG()` function of the ggm library. Here is the function: `basiSet.MAG(cgraph)`. Here, cgraph is a matrix object, created using `DAG()` or `makeMG()`.

The get.AIC function

The `get.AIC()` function outputs the log-likelihood (LL), the number of free parameters that were estimated (K), the AIC value and the bias-corrected AIC (AICc). You must input (1) a list (`sem.model`) containing the structural equations, each created using either the `gam()` or the `gamm()` functions of the mgcv package, (2) the DAG or MAG (`MAG`) and (3) the data frame (`data`) containing the observed data used in the calls to the models in the `sem.model` object. Here is the function: `get.AIC(sem.model, MAG, data)`.

The CI.algorithm function

This function implements the exploratory method of causal discovery called the IC (Inductive causality) algorithm in Pearl (2009) and the CI (Causal Inference) algorithm of Spirtes et al. (2000). It uses the patterns of independence and conditional independence in an empirical data set to determine the partially oriented dependency graph that is implied by these patterns of (conditional) independence. The patterns of independence and conditional independence in an

empirical data are obtained using the generalized covariance function, based on either (generalized) linear (mixed) models or (generalized) additive (mixed) models (the default). Note that if any nesting structure is declared, the random terms in the model are based only on random intercepts; if substantial variation in random slopes do exist, this could result in incorrect output. You can include prior knowledge concerning (i) the absence of an undirected edge in the final partially oriented dependency graph ("X|Y"), (ii) the presence of a direct parent-child link ("X->Y") or (iii) the presence of a common latent direct cause (X<->Y") using the `constrained.edges` argument. Note that "Y<-X" is not permitted and will result in an error. Here is the CI.algorithm function:

```
CI.algorithm(dat, family=NA,nesting=NA,smooth=TRUE,alpha.reject
= 0.05, constrained.edges=NA,write.result = T)
```

Arguments

`dat`: a named data frame containing only the variables for which the partially oriented dependency graph is sought as well as (if nesting is present in the data) the variables describing the random terms. This is identical to the data structure of the `pwSEM()` function.

`family`: a named data frame giving the type of distribution to assume for each variable that is not normally distributed. This argument is not needed if all variables all normally distributed.

`nesting`: a named list giving the random terms describing the nesting structure of each variable. This argument is not needed if no nesting structure exists for any of these variables.

`smooth`: a logical value stating if (generalized) linear relationships are assumed (smooth=FALSE) or if (generalized) additive (i.e. potentially nonlinear) relationships are assumed.

`alpha.reject`: a numerical value between 0 and 1 giving the probability below which each of the null hypotheses of (conditional) independence are rejected. Small data sets will likely require larger values than 0.05. Often, the `CI.algorithm()` function is first run using a low value (e.g. 0.01) and then sequentially re-run with increasingly higher values in order to see how the resulting partially oriented dependency graph changes.

`constrained.edges`: a character object specifying which pairs of variables have *a priori* known edges in the partially oriented dependency graph: `constrained.edges="X2|X1"` means that you know that there cannot be an edge between X1 and X2 in this graph, `"X2->X1"` means that X2 is a direct cause of X1 and `"X2<->X1"` means that neither X1 nor X2 cause the other, but both share a common latent direct cause. For more than one constrained edge, insert a RETURN after each pair in the character object with the full set of pairs enclosed in quotes. The default is `constrained.edges=NA`, meaning that no edges are constrained by *a priori* knowledge.

`write.result`: a logical value. If `write.result=T` then the partially oriented dependency graph is written to the screen. If `write.result=F` then only a matrix is returned with (i,j)=1 meaning a line joins variables i and j, (i,j)=2 means a line with an arrowhead pointing into j joins variables i and j.

Here is an example, using the nested_data data set, which includes a nesting structure and a binomial distribution for the XR variable. The first line creates a named list in which two variables in the data frame (year, nest) define the nesting structure of each variable (XR, XM, XH, XP, XF) in the partially oriented dependency graph. The second line calls the `CI.algorithm()` function. Note that column 3 of the data set is excluded because it is not to be included in the partially oriented dependency graph. Variable XR is defined as a non-normally distributed (binomial) variable and XP is defined as a Poisson distributed variable.

Only variables that are not normally (gaussian) distributed must be explicitly declared in the `family=` argument. The partially oriented dependency graph is obtained for the case in which every test of (conditional) dependency whose null probability is less than 0.3 is assumed to be dependent and nonlinear smoother functions are used.

```
nesting.structure<-
list(XR=c("year","nest"),XM=c("year","nest"),XH=c("year","nest")
,XP=c("year","nest"),XF=c("year","nest"))
CI.algorithm(dat=nested_data[,3],family=data.frame(XR="binomial"
,XP="poisson"),
nesting=nesting.structure,alpha.reject=0.3,smooth=T)
```

The resulting partially oriented dependency graph defines a set of equivalent causal graphs. An "o" means that there could be either an arrowhead (> or <) or nothing at this end of the edge. The rules for orienting these edges in equivalent causal graphs apply, as explained in the book.

If you wanted to forbid edges between XF and XM (XF|XM) and also force XH to directly cause XR, you would do this:

```
nesting.structure<-
list(XR=c("year","nest"),XM=c("year","nest"),XH=c("year","nest")
,XP=c("year","nest"),XF=c("year","nest"))
con.edges<-"
XF|XM
XH->XR
"
CI.algorithm(dat=nested_data[,3],family=data.frame(XR="binomial"
,XP="poisson"),nesting=nesting.structure,alpha.reject=0.3,constr
ained.edges=con.edges,smooth=T)
```

Note that forbidding edges is a strong causal claim! In this example, you are claiming to know, for example, that XF cannot be a direct cause of XM, XM cannot be a direct cause of MF, and that there is no latent variable that is a direct cause of both XM and XF.

The vanishing.tetrads function

This function tests for tetrad equations that are not significantly different from zero, and implements the vanishing tetrad algorithm of (Spirtes et al. 1993).

If a set of four observed variables has a saturated partially oriented dependency graph, as determined by the CI algorithm, and if all three tetrad equations involving these four variables is zero, then this implies a measurement model in which each of the four observed variables is the causal child of only a single common latent cause.

Note that this algorithm (and the function) assumes multivariate normality, mutually independent observations, sufficient sample size and linearity between the latent and each observed variable. Depending on the numerical strengths of the path coefficients linking each observed variable to the latent cause, the sample size can be quite large (several hundreds). You should use the bootstrap version of the test for data that are not normally distributed.

Here is the `vanishing.tetrads()` function:

```
vanishing.tetrads(dat, sig = 0.05, bootstrap=FALSE, B=1000)
```

arguments

`dat`: a data frame or matrix having at least four columns, and containing only numerical values

`sig`: the significance level to be used in the inferential test.

`bootstrap`: a logical value specifying if you want bootstrap null probabilities. This defaults to zero, meaning that you will get asymptotic values assuming multivariate normality. Bootstrap probabilities do not assume multivariate normality and are not asymptotic, but a minimum sample size of ~30 is often recommended.

Here is an example using simulated data (`sim_tetrads`) given the DAG L->Z1, L->X2, L->X3 and L->X4. In other words, there is a latent variable (L) that is a common cause of X1, X2 and X3, but not of X4. First, determine if the partially oriented dependency graph is saturated, using the `CI.algorithm()` function. A saturated graph is one in which every variable is joined by a line to every other variable.

```
CI.algorithm(sim_tetrads)
```

Since this graph is saturated, we then test the three possible tetrad equations, given four variables:

```
vanishing.tetrads(dat=sim_tetrads,sig=0.05)
```

If you do not want to assume multivariate normality, you can use the bootstrap probabilities, although this takes (slightly) longer to run. Here is the call with the default 1000 bootstrap runs:

```
vanishing.tetrads(dat=sim_tetrads,sig=0.05,bootstrap=TRUE,B=1000
)
```

All three tetrad equations vanish (i.e. are not significantly different from zero), and this result implies a single common latent variable as the causal parent of all four observed variables.

10.4 The lavaan package

The lavaan package (Rosseel 2012) (version 0.6-17) is on CRAN. There are three main functions in this package: `lavaan()`, `cfa()` and `sem()`. The cfa function is optimised for confirmatory factor analysis, the sem function is optimised for structural equations modelling and the lavaan function is the most general function, upon which the other two are special cases. Only `sem()` is discussed in this book. The general structure of an R session using the lavaan library, as presented in this book, has three parts:

1. The creation of a model object that specifies the model structure. The model structure is enclosed in quotes.
2. A call to the `sem()` function, which inputs the model object, the data object, and other arguments. Using this information, it fits the model to the data and calculates the various output statistics.
3. A call to `summary()` or to one of several extractor functions in order to obtain various types of information about the model fit.

A simple example of this sequence is:

```
#input this simple model: X→Y→Z and save it as an object called
"my.model"
my.model<-"Y~X
Z~Y"
# fit "my.model" to a data set called "input.data" using the
sem() function
model.fit<-sem(model=my.model,data=input.data)
# obtain a summary of the model fit
summary(model.fit)
```

The next section summarizes the various details in lavaan related to specifying the model (i.e. model syntax), choosing values for the arguments of the `sem()` function that control the fitting of the model to the data, and the various extractor functions that allow you to see various details of the resulting fit.

Specifying the model structure: model syntax

Formula type	Operator	Example	Causal graph	Meaning
Latent variable definition	=~	`L =~ x+y`	L→y L→x	Latent variable "L" causes and is measured by observed variables x and y.
regression	~	`y ~ x`	x→y	Observed variable y is caused by, and is regressed on, x.
(residual) (co)variance	~~	`x ~~ y, x~~x`	x↔y, x↔x	free covariance, free variance.
intercept		`x ~ 1`		estimate the intercept of x.
New parameter	:=	`Total:=a+b`		Create a new free parameter called "Total" which is constrained to be the sum of old free parameters a and b.

Fixing a parameter value

Whenever you specify a structural equation via the model syntax of lavaan you implicitly define free parameters. One exception is when you use the "=~" operator since the path coefficient of the first observed variable on the right-hand side of the operator is fixed to unity by default; see "allowing the first indicator of a latent variable to be free" to change this default choice. Thus, a model statement like "y~x+z" implicitly defines two free parameters which are the path coefficients associated with x and z. In fact, depending on what other command lines you include

in the model object, this statement could also implicitly define free error variances and covariances.

In order to force a parameter to take a specific value ("fixing it") rather than allowing it to be estimated from the data, you "multiply" the variable by the desired fixed value. Thus, "y~x+6*z" means that the path coefficient associated with the variable "z" is fixed to a value of 6 and can't be changed during the process of parameter estimation.

Examples:

`y~1.5*x` The path coefficient associated with x is fixed at a value of 1.5.

`y~~1*y` The (residual) variance of y is fixed at a value of 1.

`Y~~0*x` The covariance between (the residuals of) y and x is fixed at zero.

Specifying starting values

The iterative process of estimating the values of free parameters requires specifying the initial (starting) values of these free parameters. By default, lavaan sets all starting values to unity. However, more complicated models can fail to converge and one reason for this is that the starting values were simply too far away from the final values. In such cases, one must supply better initial values. This is done by via the `start()` argument, which is "multiplied" to the variable.

Example:

`y~start(0.1)*x` The path coefficient associated with the variable "x" is free but its starting value during the iterative fitting process is equal to 0.1.

`y~~start(10)*y` The starting value of the free (residual) variance of y is equal to 10.

`L=~start(0.001)*x+y` The starting value of the free path coefficient associated with x is 0.001 (see also "allowing the first indicator of a latent to be free".

Specifying starting values in multigroup models

In order to fit a model involving more than one group, you need to have at least one grouping variable in the data frame. If you want the starting value of a parameter to be the same in all groups, then simply give a single start value. To specify different start values for each group, you specify these as a vector: `start(c(0.1,0.2,...))*x`

Preventing exogenous variables from freely covarying

By default, all exogenous variables in lavaan are assumed to have non-zero covariances. The default occurs because the default value of the argument `fixed.x` in the `sem()` function is `fixed.x=TRUE`. This is a poor default choice and so you should always explicitly specify the state of these exogenous covariances based on your conception of the causal process. To do this, you must specify `sem(...,fixed.x=FALSE)` in the `sem()` function and then explicitly specify these covariances in the model syntax as either fixed (to zero or some other value) or free; see "fixing a parameter value".

Example:

```
my.model<-"z~x+y
# the next line fixes the covariance between the two exogenous
variables to #zero
x~~0*y "
sem(model.syntax=my.model, data=, fixed.x=FALSE)
```

Specifying parameter labels

Every parameter in your model syntax has a name. For example, these are the parameter names that you see when you use the `summary()` function. The default name for a parameter is simply a concatenation of variable name 1 + operator + variable name 2. In other words if, in your model syntax, you have a line like `y~x+z` then the first path coefficient is named "y~x" and the second path coefficient is named "y~z". However, you can specify your own names for

these parameters. To do this, simply "multiply" the variable name, using the usual naming conventions of R, by the label that you want to use for its associated parameter.

Examples:

`y ~ x + z` The (free) path coefficient associated with variable x is called "y~x" and the (free) path coefficient associated with variable z is called "y~z".

`y ~ a*x + b*z` The (free) path coefficient associated with variable x is called "a" and the (free) path coefficient associated with variable z is called "b".

You can combine these "multiplication" conventions. For instance: `y ~ start(2)*a*x` both specifies a starting value and a label name for the free path coefficient associated with x.

Specifying parameter labels in multiple groups

In order to fit a multigroup model, you need to have a grouping variable in the data frame. If you want different labels for each group then specify these different labels as a vector whose length is equal to the number of groups in the model: `c(ag1,ag2,...)*x`. BE CAREFUL; if you give the same label to more than one group then this will force the parameter estimation to be equal across these groups sharing the same label name.

Specifying equality constraints

You can force combinations of free parameters to be equal during model fitting. In such cases, the fitted values of the parameters are still chosen so as to minimize the difference between the observed and model covariance matrices, but the chosen values are constrained to be equal. This is most often done when fitting multigroup models but can be done whenever your causal hypothesis requires it. There are different but equivalent ways of doing this.

1: Simply give the same parameter label to the parameters whose values are to be equal. For example, `y ~ a*x + a*z`, will force the values of the estimated path coefficients associated with both variables x and z to be equal since the labels of both parameters to be equal.

2. Use the `equal()` function. For example, `y ~ x + equal("y~x")*z` forces the fitted value of the path coefficient associated with variable z to equal the value associated with variable x.

3. Use the "==" operator:

```
y ~ a*x + b*z
a == b
```

Specifying nonlinear equality or inequality constraints

1. Give explicit labels to the parameters in question.
2. Specify the desired constraint using logical operators (e.g. "==", "<" or ">") and the parameter labels. For example:
3.

```
# give names (a1, a2, a3) to the three free path
coefficients
y ~ a1*x + a2*z +a3*e
# here is a nonlinear equality constraint
a1 == (a2 + a3)^2
# here is a nonlinear inequality constraint
a1 > exp(a2 + a3)
```

Preventing a free variance from being negative

By definition, a variance cannot be negative, but the various algorithms used by lavaan to estimate parameter values don't know this fact. As a result, it sometimes happens that the estimated value of free residual variance is negative. This is usually a sign of a poorly fitting model or some problem that has occurred during the iterative process of estimation. However, it is possible that the model fits the data well but that the true value of the residual variance is very close to zero. If this happens then the estimate can become negative because of sampling fluctuations. If you think that this is the case then you force lavaan to maintain non-negative variance estimates by specifying a nonlinear constraint on this residual variance as follows:

```
# name the parameter label for the variance
x ~~ varx*x
# force this parameter value to remain non-negative
varx >= 0
```

Specifying more than one causal model in multigroup or multilevel models

Inside the model object, you must name each group and then enter the model specifics for each group. For instance, to have x→y→z in group 1 and x→y←z in group two, with no cross-group equality constraints (thus, different labels for the free parameters), you would specify:

```
"
Group 1:
y~a1*x
z~b1*y
x~~vx1*x
y~~vy1*y
z~~vz1*z
Group 2:
y~a2*x +b2*z
x~~vx2*x
y~~vy2*y
z~~vz2*z
"
```

Calculating compound (indirect, total) effects

To calculate compound effects, such as indirect or total effects, and their standard errors, you must first give labels to the coefficients in question and then use the := operator to calculate the desired compound effects. For instance, consider the simple path model: x─(a)→y, y─(b)→z, x─(c)→z where a, b and c are the label names for the three direct effects. There is both a direct effect of x on z (x→z) and an indirect effect (x→y→z). Only the direct effect is calculated by default in lavaan (i.e. the value of the path coefficient associated with variable z). To calculate the direct, indirect and total effects of x on z you would do:

```
my.model<-"
# give label names for the path coefficients
y ~ a*x
z ~c*x  + b*y
# define the parameter measuring the indirect effect (a*b)
indirect.effect := a*b
# define the parameter measuring the total effect
total.effect := c + (a*b)"
```

When you fit this model then the values and standard errors of the two new defined parameters (`indirect.effect` and `total.effect`) will be output.

Allowing the first indicator of a latent to be free

When specifying a latent variable via the =~ operator, the default in lavaan is to fix the path coefficient of the first indicator variable on the right-hand side of the operator to unity in order to fix the scale of the latent. If you want to fix the scale of the latent in this way, then you don't have to do anything except to make sure that the first observed variable on the right-hand side is the variable whose scale you want to use. However, if you don't want to do this (for instance, you want to identify the latent by fixing its variance to unity) then the usual (and easiest) way is to explicitly fix the latent variance to unity via the `std.lv` argument in the `sem()` function: `sem(...,std.lv=TRUE)`. This fixes the standard deviation of all of the latent variables in the model to unity.

However, there are times in which this method is not appropriate. For instance, you might have more than one latent in your model whose scales you want to fix in different ways. Alternatively, you might want to fix the scale of the latent using some value of an observed scale other than unity. In such instances, you can explicitly force that the path coefficient of the first observed variable on the right-hand side of the =~ operator be free by "multiplying" it by NA (to free it) or by a number (to fix its scale to a value other than unity). Thus, `L =~ NA*x + ...` tells lavaan that the path coefficient associated with x (the first observed variable on the right-hand side) is NOT fixed. Similarly, `L=~2.5x+ ...` tells lavaan that the path coefficient associated with x is fixed at 2.5.

Arguments used when fitting the model via `sem()`

The `sem()` function is actually a wrapper for another, more general function, called `lavaan()`. There are two other wrapper functions, called `cfa()` and `growth()`, that I won't discuss here. The `sem()` function contains very many arguments and several of these arguments deal with advanced topics that are not discussed in this book. Most of these

arguments have default values and there are complicated interactions between these default values affecting things like the method of parameter estimation, the types of test statistics that are calculated, and so on. Which of these arguments can be safely kept at their default values, and which need to be specified will depend on the complexity of your model. I have indicated those arguments that refer to topics that are discussed in this book with an asterisk.

Here is the full function. You will see most of this if you use type help(sem) in R but I have added some further details:

```
sem(model = NULL, data = NULL,
   meanstructure = "default", fixed.x = "default",
   orthogonal = FALSE, std.lv = FALSE,
   parameterization = "default", std.ov = FALSE,
   missing = "default", ordered = NULL,
   sample.cov = NULL, sample.cov.rescale = "default",
   sample.mean = NULL, sample.nobs = NULL,
   ridge = 1e-05, group = NULL,
   group.label = NULL, group.equal = "", group.partial = "",
   group.w.free = FALSE, cluster = NULL, constraints = '',
   estimator = "default", likelihood = "default", link = "default",
   information = "default", se = "default", test = "default",
   bootstrap = 1000L, mimic = "default", representation = "default",
   do.fit = TRUE, control = list(), WLS.V = NULL, NACOV = NULL,
   zero.add = "default", zero.keep.margins = "default",
   zero.cell.warn = TRUE,
   start = "default", verbose = FALSE, warn = TRUE, debug = FALSE)
```

`model*` = the name of the object holding the model description. This must always be specified.

`data*` = the name of the data frame holding the observations, including (if applicable) the grouping structure. You must always either provide this data frame or else provide (1) the covariance matrix via the `sample.cov` argument, (2) the vector of sample means for each variable via the `sample.mean` argument and (3) the number of observations used to calculate the sample covariance matrix via the `sample.obs` argument. These last three arguments are described below.

`meanstructure*` = `FALSE` by default; if TRUE then the means (intercepts) are also modelled.

`fixed.x*` = If TRUE, the exogenous variables are not considered random variables and so the means, variances and covariances of these variables are not estimated but are rather fixed to their sample values. This is different from the way these variables are treated in this book. If FALSE (which is the choice for the way they are treated in this book), they are considered random, and the means, variances and covariances are free parameters. If `default`, the value is set depending on the mimic option (see below).

`orthogonal* = If TRUE`, the exogenous latent variables are assumed to be uncorrelated; i.e. the covariances between them are fixed at zero.

`std.ov*` = (the default is FALSE) only if you want all observed variables to be standardized (unit variance, zero mean) before the analysis. This would give standardized coefficients.

`parameterization` = an argument used to treat categorical data and not discussed in this book.

`missing` = If "`listwise`", cases with missing values are removed listwise from the data frame before analysis. If "`direct`" or "`ml`" or "`fiml`" and the estimator (see below) is maximum likelihood, Full Information Maximum Likelihood (FIML) estimation is used using all available data in the data frame. This is only valid if the data are *missing completely at random* (MCAR) or *missing at random* (MAR). If "`default`", the value is set depending on the estimator and the mimic option (see below). This is only justified if the missing values are given the precise definitions of the terms "*missing at random"* or *"completely at random"*. A value is missing *completely at random* if its probability of being missing is unrelated to any other variable – observed or unobserved. For example, if you missed certain values because your measuring devise broke down one day then the pattern of missed values is probably missing completely at random. Missing values of a variable are said to be *missing at random* if the values of the other variables in the data set can predict the pattern of missingness. If you are missing data on seed output because some plants already shed their seeds before you started measurements, but you also have information on (say) the date of flowering, then this would probably be a case of *missing at random*. If, however, you have no other information in your data set that is related to the phenology of reproduction, then your missing values would not accord with this necessary assumption.

`ordered` = character vector and only used if the data is in a data frame. Treat these variables as ordered (ordinal) variables, if they are endogenous in the model. Importantly, all other variables will be treated as numeric (unless they are declared as ordered in the original data frame).

`sample.cov` = a sample covariance if you want to input this instead of the actual data set.

`sample.cov.rescale` = If `TRUE`, the sample covariance matrix provided by the user is internally rescaled by multiplying it with a factor (N-1)/N. If "default", the value is set depending on the estimator and the likelihood option: it is set to `TRUE` if maximum likelihood estimation is used and `likelihood="normal"`, and `FALSE` otherwise.

`sample.mean` = a sample mean vector if you want to model means, and you have input the sample covariance instead of the actual data set.

`sample.nobs`= the number of observations used to calculate the sample covariance matrix if you have input this instead of the actual data set.

`ridge` = a small numeric constant used for ridging. This is only used if the sample covariance matrix is non positive definite.

`group` = the name of the variable in your data frame that codes the group names (only if you are doing a multigroup model). If you do not specify this argument, then lavaan assumes that you have only one group.

`group.label` = a character vector. You can use this to specify which group (or factor) levels need to be selected from the grouping variable, and in which order. If `NULL` (the default), all grouping levels are selected, in the order as they appear in the data.

`group.equal` = a vector of character strings. This is only used in a multigroup analysis and is used to specify the pattern of equality constraints across multiple groups. Choices can be one or more of the following:

> ="`loadings`" means that the path coefficients from latents to indicators are equal across groups, as specified by the "=~" operator in the model syntax

="regressions" means that all regression coefficients are equal across groups, as specified by the "~" operator in the model syntax

="residuals" means that the residual variances of the observed variables are equal across groups

="residual.covariances" means that the covariances of the observed variables are equal

="lv.variances" means that the (residual) variances of the latents are equal

="lv.covariances" means that the (residual) covariances of the latent variables are equal

="means" means that the intercepts/means of the latent variables are equal

="intercepts" means that the intercepts of the observed variables are equal

="threshholds" refers to categorical variables and this is not discussed in this book.

group.partial* = a vector of character strings containing the labels of the parameters which should be free in all groups; this is used to override the group.equal argument for some specific parameters.

group.w.free = If TRUE, the group frequencies are considered to be free parameters in the model. In this case, a Poisson model is fitted to estimate the group frequencies. If FALSE (the default), the group frequencies are fixed to their observed values. This is not discussed in this book.

constraints* = additional (in)equality constraints not yet included in the model syntax. This is an alternative to including such constraints in the model syntax.

estimator* = The estimator to be used; the default is "ML" for maximum likelihood. There are many choices here, only some of which are discussed in this book. The options are: "ML" for maximum likelihood, "GLS" for generalized least squares, "WLS" for weighted least squares (sometimes called ADF estimation), "ULS" for unweighted least squares and "DWLS" for diagonally weighted least squares. These are the main options that affect the estimation. For

convenience, the "ML" option can be extended as "MLM", "MLMV", "MLMVS", "MLF", and "MLR". The estimation will still be plain "ML", but now with robust standard errors and a robust (scaled) test statistic. For "MLM", "MLMV", "MLMVS", classic robust standard errors are used (se="robust.sem"); for "MLF", standard errors are based on first-order derivatives (se="first.order"); for "MLR", 'Huber-White' robust standard errors are used (se="robust.huber.white"). In addition, "MLM" will compute a Satorra-Bentler scaled (mean adjusted) test statistic (test="satorra.bentler") , "MLMVS" will compute a mean and variance adjusted test statistic (Satterthwaite style) (test="mean.var.adjusted"), "MLMV" will compute a mean and variance adjusted test statistic (scaled and shifted) (test="scaled.shifted"), and "MLR" will compute a test statistic which is asymptotically equivalent to the Yuan-Bentler T2-star test statistic. Analogously, the estimators "WLSM" and "WLSMV" imply the "DWLS" estimator (not the "WLS" estimator) with robust standard errors and a mean or mean and variance adjusted test statistic. Estimators "ULSM" and "ULSMV" imply the "ULS" estimator with robust standard errors and a mean or mean and variance adjusted test statistic.

likelihood = Only relevant for ML estimation. If "wishart", the Wishart likelihood approach is used. In this approach, the covariance matrix has been divided by N-1, and both standard errors and test statistics are based on N-1. If "normal", the normal likelihood approach is used. Here, the covariance matrix has been divided by N, and both standard errors and test statistics are based on N. If "default", it depends on the mimic option: if mimic="lavaan" or mimic="Mplus", normal likelihood is used; otherwise, Wishart likelihood is used.

link = Currently only used if the chosen estimator is MML and you have binary or ordered observed endogenous variables; this topic has not been discussed in this book. If "logit", a logit link is used for binary and ordered observed variables. If "probit", a probit link is used. If "default", it is currently set to "probit".

information = If "expected", the expected information matrix is used (to compute the standard errors). If "observed", the observed information matrix is used. If "default", the value is set depending on the estimator and the mimic option.

`se` = specifies how the standard errors of the parameter estimates are to be calculated. If `"standard"` (the default), conventional standard errors are computed based on inverting the (expected or observed) information matrix. If `"first.order"`, standard errors are computed based on first-order derivatives. If `"robust.sem"`, conventional robust standard errors are computed. If `"robust.huber.white"`, standard errors are computed based on the 'mlr' (aka pseudo ML, Huber-White) approach. If `"robust"`, either `"robust.sem"` or `"robust.huber.white"` is used depending on the estimator, the mimic option, and whether the data are complete or not. If `"boot"` or `"bootstrap"`, bootstrap standard errors are computed using standard bootstrapping (unless Bollen-Stine bootstrapping is requested for the test statistic; in this case bootstrap standard errors are computed using model-based bootstrapping). If `"none"`, no standard errors are computed.

`test` = If `"standard",` a conventional chi-square test is computed. If `"Satorra.Bentler"`, a Satorra-Bentler scaled test statistic is computed. If `"Yuan.Bentler"`, a Yuan-Bentler scaled test statistic is computed. If `"mean.var.adjusted"` or `"Satterthwaite"`, a mean and variance adjusted test statistic is compute. If `"scaled.shifted"`, an alternative mean and variance adjusted test statistic is computed (as in Mplus version 6 or higher). If `"boot"` or `"bootstrap"` or `"Bollen.Stine"`, the Bollen-Stine bootstrap is used to compute the bootstrap probability value of the test statistic. If `"default"`, the value depends on the values of other arguments.

`bootstrap` = Number of bootstrap draws, if bootstrapping is used.

`mimic` = If `"Mplus"`, an attempt is made to mimic the Mplus program. If `"EQS"`, an attempt is made to mimic the EQS program. If `"default"`, the value is (currently) set to to `"lavaan"`, which is very close to `"Mplus"`.

`representation` = If `"LISREL"` the classical LISREL matrix representation is used to represent the model (using the all-y variant).

`do.fit` = If `FALSE`, the model is not fit, and the current starting values of the model parameters are preserved. Defaults to `TRUE`.

control = A list containing control parameters passed to the optimizer. By default, lavaan uses "nlminb". See the R help file of nlminb for an overview of the control parameters. A different optimizer can be chosen by setting the value of optim.method. For unconstrained optimization (the model syntax does not include any "==", ">" or "<" operators), the available options are "nlminb" (the default), "BFGS" and "L-BFGS-B". See the help page of the optim function for the control parameters of the latter two options. For constrained optimization, the only available option is "nlminb.constr".

WLS.V = A user provided weight matrix to be used by estimator "WLS"; if the estimator is "DWLS", only the diagonal of this matrix will be used. For a multiple group analysis, a list with a weight matrix for each group. The elements of the weight matrix should be in the following order (if all data is continuous): first the means (if a meanstructure is involved), then the lower triangular elements of the covariance matrix including the diagonal, ordered column by column. In the categorical case: first the thresholds (including the means for continuous variables), then the slopes (if any), the variances of continuous variables (if any), and finally the lower triangular elements of the correlation/covariance matrix excluding the diagonal, ordered column by column.

NACOV = A user provided matrix containing the elements of (N times) the asymptotic variance-covariance matrix of the sample statistics. For a multiple group analysis, a list with an asymptotic variance-covariance matrix for each group. See the WLS.V argument for information about the order of the elements.

zero.add = A numeric vector containing two values. These values affect the calculation of polychoric correlations when some frequencies in the bivariate table are zero. The first value only applies for 2x2 tables. The second value for larger tables. This value is added to the zero frequency in the bivariate table. If "default", the value is set depending on the "mimic" option. By default, lavaan uses zero.add = c(0.5. 0.0).

zero.keep.margins = This logical argument only affects the computation of polychoric correlations for 2x2 tables with an empty cell, and where a value is added to the empty cell. If TRUE, the other values of the frequency table are adjusted so that all margins are unaffected. If "default", the value is set depending on the "mimic". The default is TRUE.

`zero.cell.warn` = Only used if some observed endogenous variables are categorical. If `TRUE`, give a warning if one or more cells of a bivariate frequency table are empty.

`start` = If it is a character string, the two options are currently "`simple`" and "`Mplus`". In the first case, all parameter values are set to zero, except the factor loadings (set to one), the variances of latent variables (set to 0.05), and the residual variances of observed variables (set to half the observed variance). If "`Mplus`", we use a similar scheme, but the factor loadings are estimated using the fabin3 estimator (tsls) per factor. If `start` is a fitted object of class `lavaan`, the estimated values of the corresponding parameters will be extracted. If it is a model list, for example the output of the `paramaterEstimates()` function, the values of the `est` or `start` or `ustart` column (whichever is found first) will be extracted.

`verbose` = If `TRUE`, the function value is printed out during each iteration.

`warn` = If `TRUE`, some (possibly harmless) warnings are printed out during the iterations.

`debug` = If `TRUE`, debugging information is printed out.

Extractor functions

Once you have fit the model to the data using the `sem()` function and saved it as an object (we'll call it "fit"), you can extract different types of information about the resulting fit using different extractor functions.

```
summary(fit, standardized=FALSE, fit.measures=FALSE, rsquare=FALSE,
modindices=FALSE).
```

This is the basic extractor function that outputs (as defaults) information on convergence, the basic test statistics (which depend on which estimator you specified in the `sem()` function) and the parameter estimates and their standard errors. The additional arguments allow you to also obtain standardized parameter estimates, additional fit statistics, the proportion of the total variance (R^2) of the endogenous variables of the model that are explained, and modification indices (Lagrange multipliers).

`coef(fit)` This outputs the fitted coefficients only.

`parameterEstimates(fit)` This outputs the parameter estimates, their standard errors and confidence intervals.

`standardizedSolution(fit)` This outputs the standardized estimates of the parameters.

`residuals(fit, type="standardized")` This outputs the standardized differences between the observed and predicted covariance matrices.

`AIC(fit), BIC(fit)` These output the AIC or BIC values of the model.

`fitMeasures(fit)` This outputs all of the various fit measures.

`inspect(fit,"r2")` This outputs the R^2 (proportion of variance explained) associated with each endogenous variable.

`parameterTable(fit)` This outputs each of the parameters, their estimated values, and whether they are fixed or free.

`modindices(fit)` This outputs the modification indices for a given fitted model. You should also include the optional argument `sort.=TRUE` in order to have the modification indices sorted from highest to lowest.

References

Ainsworth, E. A., and S. P. Long. 2005. What have we learned from 15 years of Free-Air CO_2 Enrichment (FACE)? A meta-analytic review of the responses of photosynthesis, canopy properties and plant production to rising CO_2. New Phytologist **165**:351–372.

Akaike, H. 1973. Information theory and an extension of the maximum likelihood principle.*in* B. N. Petrov and F. Csaki, editors. Proceedings of the 2nd Intenational Symposium on Information Theory. Akademiai Kiado, Budapest.

Akaike, H. 1983. Information measures and model selection. International Statistical Institute **44**:277-291.

Aldrich, J. 1995. Correlations genuine and spurious in Pearson and Yule. Statistical Science **10**:364-376.

Bates, D. M., M. Machler, B. Bolker, and S. Walker. 2015. Fitting linear mixed-effects models using lme4. Journal of Statistical Software **67**:1-48.

Benjamini, Y., and Y. Hochberg. 1995. Controlling the false discovery rate : a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society Series B **57**:289-300.

Bentler, P. M. 1995. EQS structural equations program manual, version 3.0. BMDP Statistical Software, Los Angeles.

Bentler, P. M., and D. G. Bonnett. 1980. Significance tests and goodness of fit in the analysis of covariance structures. Psychological Bulletin **88**:588-606.

Berkson, J. 1946. Limitations of the application of fourfold table analysis to hospital data. Biometrics Bulletin **2**:47-53.

Bernard, C. 1865. Introduction à l'étude de la médecine expérimentale. J.-B. Ballière et fils, Paris.

Blalock, H. M. 1961. Correlation and causality: The multivariate case. Social Forces **39**:246-251.

Blalock, H. M. 1964. Causal inferences in nonexperimental research. University of North Carolina, Chapel Hill.

Bollen, A. K., and J. S. Long, editors. 1993. Testing structural equation models. Sage Publications edition, Newbury Park.

Bollen, K. A. 1989. Structural equations with latent variables. John Wiley and Sons, New York.

Bollen, K. A., and R. A. Stine. 1992. Bootstrapping goodness-of-fit measures in structural equation models.*in* This work was presented at the social Science Methodology Conference in Trento, Italy.

Boring, E. G. 1954. The Nature and History of Experimental Control. The American Journal of Psychology **67**:573-589.

Brady, N. C., and R. R. Weil. 2017. The nature and properties of soils. 15th edition. Pearson, Hoboken, NJ, USA.

Brown, M. B. 1975. A method for combining non-independent, on-sided test of significance. Biometrics **31**:987-992.

Browne, M. W., and R. Cudeck. 1993. Alternative ways of assessing model fit. Pages 136-162 *in* K. A. Bollen and J. S. Long, editors. Testing structural equation models. Sage, Newbury Park.

Burke, J. 1996. The pinball effect and other journeys through knowledge. Little, Brown and Company, Boston.
Burnham, K. P., and D. R. Anderson. 2002. Model selection and multimodel inference: A practical information-theoretic approach. Springer-Verlag, New York, NY.
Chickering, D. M. 2002. Optimal Structure Identification with Greedy Search. Journal of Machine Learning Research **3**:507-554.
Cinar, O., and W. Viechtbauer. 2022. The poolr package for combining independent and dependent p values. Journal of Statistical Software **101**:1-42.
Cowles, M., and C. Davis. 1982a. Is the .05 level subjectively reasonable? Canadian Journal of Behavioural Sciences **14**:248-232.
Cowles, M., and C. Davis. 1982b. On the origins of the .05 level of statistical significance. American Psychologist **37**:553-558.
Daou, L., and B. Shipley. 2019. The measurement and quantification of generalized gradients of soil fertility relevant to plant community ecology. Ecology **100**:e02549.
Daou, L., and B. Shipley. 2020. Simplifying the protocol for the quantification of generalized soil fertility gradients in grassland community ecology. Plant and Soil **457**:457-468.
Dawid, A. P. 1979. Conditional independence in statistical theory. Journal of the Royal Statistical Society, Series A **41**:1-31.
Deeks, J. J., and D. G. Altman. 2004. Diagnostic tests 3: likelihood ratios. British Medical Journal **329**:168-169.
Douma, J. C., and B. Shipley. 2021. Testing piecewise structural equations models in the presence of latent variables and including correlated errors. Structural Equation Modeling **28**: 582–589.
Duhem, P. 1914. La théorie physique: Son objet, sa structure. Rivière, Paris.
Epstein, R. J. 1987. A history of econometrics. Elsevier Science Publishing, New York.
Evans, R. J., and T. S. Richardson. 2014. Markovian acyclic directed mixed graphs for discrete data. The Annals of Statistics **42**:1452-1482.
Feiblman, J. K. 1972. Scientific method. Martinus Nijhoff, The Hague.
Felsenstein, J. 1985. Phylogenies and the comparative method. American Naturalist **125**:1-15.
Fisher, F. M. 1970. A correspondence principle for simultaneous equation models. Econometrica **38**:73-92.
Fisher, R. A. 1922. On the Mathematical Foundations of Theoretical Statistics. Philosophical Transactions of the Royal Society of London. Series A **222**:309-368.
Fisher, R. A. 1925. Statistical methods for research workers. 1st edition. Oliver & Boyd, Edinburgh.
Fisher, R. A. 1926. The arrangement of field experiments. Journal of the Ministry of Agriculture of Great Britain **33**:503-513.
Fisher, R. A. 1932. Statistical Methods for Research Workers. 9th edition. Oliver and Boyd, Edinburgh.
Fisher, R. A. 1935. The design of experiments. Oliver and Boyd, Edinburgh.
Fisher, R. A. 1959. Smoking. The cancer controversy. Oliver & Boyd, Edinburgh.
Frenette Dussault, C., B. Shipley, and Y. Hingrat. 2013. Linking plant and insect traits to understand multitrophic community structure in arid steppes. Functional Ecology **27**:786-792.
Galton, F. 1869. Hereditary genius: An inquiry into its laws and consequences. Macmillan, London.

Gardner, H. 1987. The theory of multiple intelligences. Annals of Dyslexia **37**:19-35.
Gardner, H., M. Kornhaber, and J.-Q. Chen. 2018. The theory of multiple intelligences: Psychological and educational perspectives. Pages 116-129 *in* R. J. Sternberg, editor. The Nature of Human Intelligence. Cambridge University Press, Cambridge.
Gelman, A., and J. Hill. 2007. Data analysis using regression and multilevel/hierarchical models. Cambridge University Press, Cambridge.
Glymour, C., K. Zhang, and P. Spirtes. 2019. Review of causal discovery methods based on graphical models. Frontiers in Genetics **10**:524 https://doi.org/510.3389/fgene.2019.00524.
Glymour, G., R. Scheines, R. Spirtes, and K. Kelly. 1987. Discovering causal structure. Artificial intelligence, philosophy of science, and statistical modeling. Academic Press, Orlando.
Grace, J. B. 2006. Structural equation modeling and natural systems. Cambridge University Press, Cambridge.
Grace, J. B., and K. A. Bollen. 2008. Representing general theoretical concepts in structural equation models: The role of composite variables. . Environmental & Ecological Statistics **15**:191-213.
Grafen, A. 1989. The phylogenetic regression. Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences **326**:119-157.
Haavelmo, T. 1943. The statistical implications of a system of simultaneous equations. Econometrica **11**:1-12.
Hoogland, J. J., and A. Boomstra. 1998. Robustness studies in covariance structure modeling. An overview and a meta-analysis. Sociological Methods and Research **26**:239-367.
Howson, C., and P. Urbach. 1989. Scientific reasoning. The Bayesian approach. Open Court, LaSalle, Illinois.
Hox, J. J. 2002. Multilevel analysis: techniques and applications. Lawrence Erlbaum, Mahwah, NJ.
Hox, J. J., M. Moerbeek, and R. van de Schoot. 2017. Multilevel Analysis: Techniques and Applications. 3rd edition edition. Routledge, New York.
Jackson, D. A. 1995. Protest - a Procrustean Randomization Test of Community Environment Concordance. Ecoscience **2**:297-303.
Jöreskog, K. G. 1967. Some contributions to maximum likelihood factor analysis. Psychometrika **32**:443-482.
Jöreskog, K. G. 1969. A general approach to confirmatory maximum likelihood factor analysis. Psychometrika **34**:183-202.
Jöreskog, K. G. 1970. A general method for analysis of covariance structures. Biometrika **57**:239-251.
Jöreskog, K. G. 1973. A general method for estimating a linear structural equation system. Pages 85-112 *in* A. S. Goldberger and O. D. Duncan, editors. Structural equation models in the social sciences. Academic Press, New York.
Keesling, J. W. 1972. Maximum likelihood approaches to causal analysis. Ph.D. University of Chicago, Chicago.
Kempthorpe, O. 1979. The design and analysis of experiments. Robert E. Krieger, Huntington, NY.
Kendall, M. G., and A. Stuart. 1983. The advanced theory of statistics. 4 edition. Charles Griffin & Company, London.

Kenny, D. A., D. A. Kashy, and N. Bolger. 1998. Data analysis in social psychology. Pages 233-265 *in* D. Gilbert, S. Fiske, and G. Lindzey, editors. The handbook of social psychology. McGraw-Hill, Boston.
Kenny, D. A., and S. Milan. 2012. Identification: A nontechnical discussion of a technical issue. Pages 145-163 *in* R. H. Hoyle, editor. Handbook of structual equation modeling. Guilford Press, New York.
Kikuzawa, K. 1991. A cost-benefit analysis of leaf habit and leaf longevity of trees and their geographical pattern. American Naturalist **138**:1250-1260.
Kline, R. B. 2016. Principles and practice of structural equation modeling. Guilford Press, New York.
Kojadinovic, I., and J. Yan. 2010. Modeling Multivariate Distributions with Continuous Margins Using the copula R Package. Journal of Statistical Software **34**:1-20.
Kullback, S. 1959. Information theory and statistics. Wiley, New York.
Kuznetsova, A., P. B. Brockhoff, and R. H. B. Christensen. 2017. lmerTest package: Tests in linear mixed effects models. Journal of Statistical Software **82**:1-26.
Lamb, E. G., K. L. Mengersen, K. J. Stewart, U. Attanayake, and S. D. Siciliano. 2014. Spatially explicit structural equation modeling. Ecology **95**:2434-2442.
Lamontagne, X., and B. Shipley. 2022. A measure of generalized soil fertility that is largely independent of species identity. Annals of Botany **129**:29-36.
Lefcheck, J. S. 2016. piecewiseSEM: Piecewise structural equation modelling in r for ecology, evolution, and systematics. Methods in Ecology and Evolution **7**:573-579.
Legendre, P., and M. J. Fortin. 1989. Spatial Pattern and Ecological Analysis. Vegetatio **80**:107-138.
Legendre, P., and L. Legendre. 2012. Numerical Ecology. Elsevier, Amsterdam.
Li, C. C. 1975. Path analysis - a primer. Boxwood Press, Pacific Grove.
Little, R. J. A., and D. B. Rubin. 1987. Statistical analysis with missing data. Wiley & Sons., NY.
Manly, B. F. J. 1991. Randomization and Monte Carlo methods in Biology. Chapman and Hall.
Manly, B. F. J. 1997. Randomization, Bootstrap and Monte Carlo Methods in Biology, second edition. Chapman and Hall, London.
Marchetti, G., M. Drton, and K. Sadeghi. 2024. ggm: Graphical Markov Models with Mixed Graphs. R package version 2.5.1.
Marschner, P. 2012. Marschner's mineral nutrition of higher plants. 3rd edition. Academic Press, London, UK.
Martins, E. P., and T. F. Hansen. 1997. Phylogenies and the comparative method: A general approach to incorporating phylogenetic information into the analysis of interspecific data. American Naturalist **149**:646-667.
Mayo, D. G. 1996. Error and the growth of experimental knowledge. Chicago University Press, Chicago.
McCaskey, J. P. 2020. History of "temperature": maturation of a measurement concept. Annals of Science **77**:399-444.
McCullagh, P., and J. A. Nelder. 1989. Generalized Linear Models. 2nd edition. Chapman and Hall, London.
Meziane, D. 1998. Étude de la variation interspécifique de la vitesse spécifique de croissance et modélisation de l'effet des attributs morphologiques, physiologiques et d'allocation de biomasse. Ph.D. Université de Sherbrooke, Sherbrooke.

Mulaik, S. A. 1986. Toward a synthesis of deterministic and probabilistic formulations of causal relations by the functional relation concept. Philosophy of Science **53**:313-332.
Muthén, B. 1994a. Latent variable modeling of longitudinal and multilevel data. Pages 453-481 *in* American Sociological Association, Section on Methodology, Showcase Session. American Sociological Association, Los Angeles.
Muthén, B. 1994b. Multilevel covariance structure analysis. Sociological Methods and Research **22**:376-398.
Niles, H. E. 1922. Correlation, causation and Wright's theory of "path coefficients". Genetics **7**:258-273.
Norton, B. J. 1975. Biology and philosophy: The methodological foundations of biometry. Journal of the History of Biology **8**:85-93.
Passmore, J. 1966. A hundred years of philosophy. 2 edition. Penguin, Harmondsworth.
Pearl, J. 1988. Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann, San Mateo, CA.
Pearl, J. 1993. Graphical models, causality, and intervention. Statistical Science **8**:266-269.
Pearl, J. 1997. The new challenge: From a century of statistics to an age of causation. Computing Science and Statistics **29**:415-423.
Pearl, J. 2000. Causality. Cambridge University Press, Cambridge.
Pearl, J. 2009. Causality: Models, Reasoning, and Inference. 2nd edition. Cambridge University Press, Cambridge.
Pearl, J., and D. Mackenzie. 2018. The book of why. The new science of cause and effect. Basic Books, NY.
Pearl, J., and T.S.Verma. 1991. A Statistical Semantics for Causation. Pages 2-5 *in* Technical Report (R-155). In Proceeding 3rd International Workshop on AI & Statistics Fort Lauderdale Fl.
Pearson, E. S., and M. G. Kendall. 1970. Studies in the history of statistics and probability. Griffin, London.
Pearson, K. 1892. The Grammar of Science. 1 edition. Adam & Charles Black, London.
Pearson, K. 1911. The Grammar of Science. 3 edition. Adam & Charles Black, London.
Pegg, D. T. 2008. Retrocausality and Quantum Measurement. Foundations of Physics **38**:648-658.
Pinheiro, J. C., and D. M. Bates. 2000. Mixed-effects models in S and S-PLUS. Springer, New York.
Platt, J. R. 1964. Strong inference. Science **146**:347-353.
Pollack, J. L. 1986. Contemporary theories of knowledge. Rowman & Littlefield, Totowa.
Popper, K. 1980. The logic of scientific discovery. 10th edition. Hutchinson, London.
Preacher, K. J., and E. C. Merkle. 2012. The problem of model selection uncertianty in structural equation modeling. Psychological Methods **17**:1-14.
Provine, W. B. 1986. Sewall Wright and evolutionary biology. University of Chicago Press, Chicago.
Rao, M. M. 1984. Probability theory with applications. Academic Press, Orlando.
Rapport, S., and T. Wright. 1963. Science: method and meaning. New York University Press, New York.
Réale, D., S. M. Reader, D. Sol, P. T. McDougall, and N. J. Dingemanse. 2007. Integrating animal temperament within ecology and evolution. Biological Reviews **82**:291-318.

Richardson, T. 1996. A discovery algorithm for directed cyclic graphs. Pages 454-461 *in* Proceedings of the 12th Conference of Uncertainty in Artificial Intelligence. Morgan Kaufmann, Portland, OR.
Richardson, T., and P. Spirtes. 2002. Ancestral graph Markov models. The Annals of Statistics **30**:962-1030.
Rosenbaum, P., and D. Rubin. 1983. The central role of propensity score in observational studies for causal effects. Biometrika **70**:41-55.
Rosseel, Y. 2012. lavaan: An R Package for Structural Equation Modeling. Journal of Statistical Software **48**:1-36.
Rothman, K. J., S. Greenland, and T. L. Lash. 2018. Modern epidemiology. 3rd edition. Lippincott, Williams & Wilkins, Philadelphia.
Royall, R. 1997. Statistical evidence. A likelihood paradigm. Chapman & Hall, London.
Satorra, A., and P. M. Bentler. 1988. Scaling corrections for chi-square statistics in covariance structure analysis. Pages 308-313 *in* Proceedings of the American Statistics Association. American Statistics Association, Alexandria, Va.
Shah, R. D., and J. Peters. 2020. The hardness of conditional independence testing and the generalised covariance measure. The Annals of Statistics **48**:1514-1538.
Shipley, B. 1995. Structured interspecific determinants of specific leaf area in 34 species of herbaceous angiosperms. Functional Ecology **9**:312-319.
Shipley, B. 2000. A new inferential test for path models based on directed acyclic graphs. Structural Equation Modeling **7**:206-218.
Shipley, B. 2009. Confimatory path analysis in a gereralized multilevel context. Ecology **90**:363-368.
Shipley, B. 2013. The AIC model selection method applied to path analytic models compared using a d-separation test. Ecology **94**:560-564.
Shipley, B. 2021. Ordination methods for biologists: a non-mathematical introduction using R. BS Publishing, Sherbrooke (QC).
Shipley, B., and J. C. Douma. 2020a. Generalized AIC and chi-squared statistics for path models consistent with directed acyclic graphs. Ecology **101**:e02960.
Shipley, B., and J. C. Douma. 2020b. Generalized AIC and chi-squared statistics for path models consistent with directed acyclic graphs. Ecology **101**:e02960.
Shipley, B., and M. J. Lechowicz. 2000. The functional co-ordination of leaf morphology, nitrogen concentration, and gas exchange in 40 wetland species. Ecoscience **7**:183-194.
Shipley, B., M. J. Lechowicz, I. Wright, and P. B. Reich. 2006. Fundamental trade-offs generating the worldwide leaf economics spectrum. Ecology **87**:535-541.
Shipley, B., and R. H. Peters. 1990. A test of the Tilman model of plant strategies: relative growth rate and biomass partitioning. American Naturalist **136**:139-153.
Shipley, B., and R. H. Peters. 1991. The Seduction by Mechanism - a Reply to Tilman. American Naturalist **138**:1276-1282.
Shipley, B., and A. Tardif. 2021. Causal hypotheses accounting for correlations between decomposition rates of different mass fractions of leaf litter. Ecology **102**:e03196.
Sklar, A. 1959. Fonctions de répartition à n dimensions et leurs marges. **8**:229-231.
Sokal, R. R., and F. J. Rohlf. 1981. Biometry. 2 edition. Freeman, New York.
Spearman, C. 1904. General intelligence objectively determined and measured. American Journal of Psychology **15**:201-293.

Spirtes, P., C. Glymour, and R. Scheines. 1993. Causation, Prediction, and Search. Springer-Verlag, New York.
Spirtes, P., C. Glymour, and R. Scheines. 2000. Causation, prediction, and search. 2nd Edition edition. MIT Press, Cambridge, Mass.
Steiger, J. H. 1990. Structural model evaluation and modification: An interval estimation approach. Multivariate Behavioral Research **25**:173-180.
Student. 1908. The probable error of a mean. Biometrika **6**:1-25.
Sugiura, N. 1978. Further analysis of the data by Akaike's information criterion and the finite corrections. Communications in Statistics, Theory and methods **A7**:13-26.
Tanaka, J. S. 1993. Multifaceted conceptions of fit in structural equation models. Pages 10-39 *in* K. A. Bollen and J. S. Long, editors. Testing structural equation models. Sage, Newbury Park.
Thomas, D. W., B. Shipley, J. Blondel, P. Perret, A. Simon, and M. M. Lambrechts. 2007. Common paths link food abundance and ectoparasite loads to physiological performance and recruitment in nestling blue tits. Functional Ecology **21**:947-955.
Thurstone, L. L. 1935. The vectors of mind: Multiple-factor analysis for the isolation of primary traits. University of Chicago Press, Chicago.
van Buuren, S., and K. Groothuis-Oudshoorn. 2011. mice: Multivariate imputation by chained equations in R. Journal of Statistical Software **45**:1-67.
van der Vaart, A. J. 1998. Asymptotic statistics. Cambridge University Press, Cambridge.
Verheyen, K., G. R. Guntenspergen, B. Beisbrouck, and M. Hermy. 2003. An integrated analysis of the effects of past land use on forest herb colonization at the landscape scale. Journal of Ecology **91**:731-742.
Verma, T., and J. Pearl. 1988. Causal networks: Semantics and expressiveness. Pages 352-359 *in* Proceedings of the 4th Workshop on Uncertainty in Artificial Intelligence, Mountain View.
Verma, T., and J. Pearl. 1990. Causal networks: Semantics and expressiveness. Pages 69-76 *in* R. Shachter, T. S. Levitt, and L. N. Kanal, editors. Uncertainty in AI 4. Elsevier Science Publishers.
von Hardenberg, A., and A. Gonzalez-Voyer. 2012. Disentagling evolutionary cause-effect relationships with phylogenetic confirmatory path analysis. Evolution **67**:378-387.
Wishart, J. 1928. Sampling errors in the theory of two factors. British journal of psychology **19**:180-187.
Wood, S. N. 2017. Generalized additive models: An introduction with R. CRC Press, Boca Raton, FL.
Wright, I. J., P. B. Reich, M. Westoby, D. D. Ackerly, Z. Baruch, F. Bongers, J. Cavender-Bares, T. Chapin, J. H. C. Cornelissen, M. Diemer, J. Flexas, E. Garnier, P. K. Groom, J. Gulias, K. Hikosaka, B. B. Lamont, T. Lee, W. Lee, C. Lusk, J. J. Midgley, M. L. Navas, U. Niinemets, J. Oleksyn, N. Osada, H. Poorter, P. Poot, L. Prior, V. I. Pyankov, C. Roumet, S. C. Thomas, M. G. Tjoelker, E. J. Veneklaas, and R. Villar. 2004. The worldwide leaf economics spectrum. Nature **428**:821-827.
Wright, S. 1921. Correlation and causation. Journal of Agricultural Research **10**:557-585.
Wright, S. 1925. Corn and hog correlations. 1300, U.S. Deparment of Agriculture.
Wright, S. 1984. Diverse uses of path analysis. Pages 1-34 *in* A. Chakravarti, editor. Human population genetics. Van Nostrand Reinhold, New York.

Index