Latent Variable Modelling Workflow Reference

Alex Rand

10/25/22

Table of contents

Preface

What is this?

This is a book full of code to use when you want to do latent variable modelling. It gives suggested workflows I've cobbled together from a few different textbooks, and has worked examples with data from those textbooks or from open datasets I found online. When you need to do latent variable modelling for your research, you can use these workflows as a place to start.

Specifically, it seems like these are the sub-areas of latent variable modelling to know how to do:

- Exploratory Factor Analysis;
- Confirmatory Factor Analysis;
- Item Response Theory;
- Full SEM;
- Longitudinal SEM

Maybe I'll discover some other types of things along the way. It's a lifelong journey haha.

What am I referencing?

The first book on latent variable modelling I read was Gorsuch (1983). This was a nice conceptual introduction, but the applied examples were pretty whack. I've since found a few sources with data and R code to work with:

- *Latent Variable Modelling with R*, by Finch (2015). They helpfully provide all of the datasets here.
- *Principles and Practice of Structural Equation Modeling*, by Kline (2011). The publisher provides data and code here.
- The *lavaan* documentation has some nice worked examples too.

I'll mostly be using **lavaan** and **tidyverse**, but maybe also some **brms** at some point.

1 Introduction

This is a book created from markdown and executable code.

```
1 + 1
```

```
[1] 2
```

2 CFA

Let's load the packages we'll need for what is to come in this chapter:

```
library(tidyverse)
library(lavaan)
```

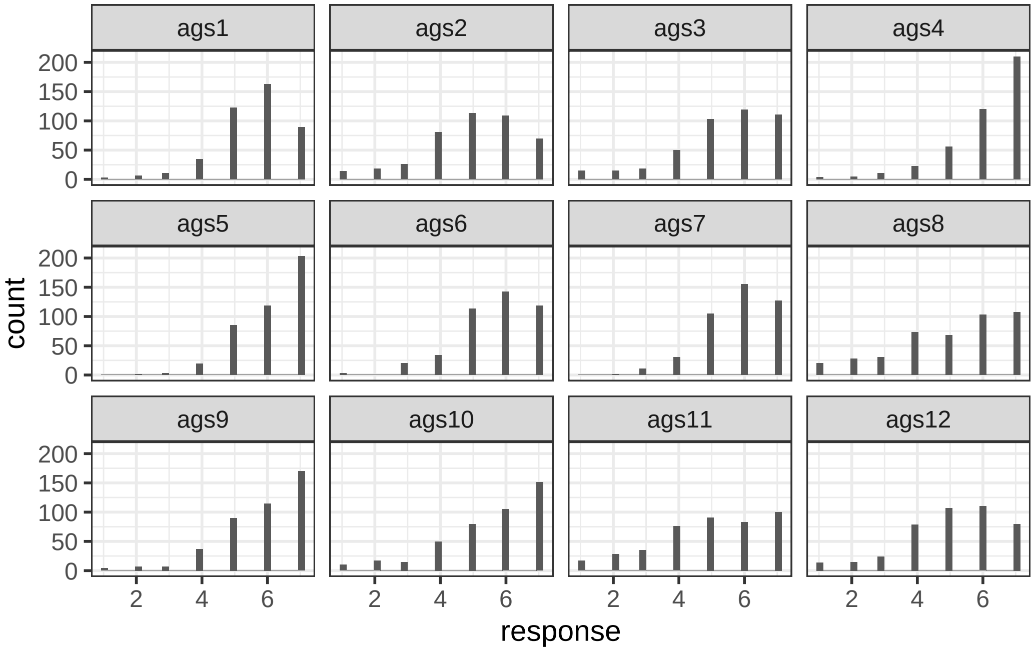
2.1 Example 1: Toxic Striving Energy

The first example we'll look at is from Finch (2015). chapter 3. The practice dataset is introduced on page 10. It is from a study about human motivation. The dataset is a weird questionnaire called the 'Achievement Goal Scale' (AGS), which asks people 12 questions about how much toxic striving energy they have. The dataset provided seems to have lots of mysterious columns in it, but we're probably good to just keep the columns that mention AGS in the name:

```
### Load the data
dat_raw <- foreign::read.spss('data/finch-and-french/edps744.sav')

### Clean the data
dat_ags <- dat_raw %>%

  # Convert to a data frame for ease of use
  as.data.frame() %>%

  # Keep only columns that start with the prefix 'ags' followed by a question number
  select(matches("ags\\d"))
```

We don't want to do too much exploration before fitting our factor models, because the whole game of CFA is to commit to our hypotheses before checking what the data looks like, so we don't mislead ourselves with [forking paths](http://www.stat.columbia.edu/~gelman/research/unpublished/p_h
But just for fun, we can explore the distributions of the answers to each of the 12 questions:

```
dat_ags %>%

  # Pivot to prepare the data for visualization
  pivot_longer(
    cols      = everything(),
    names_to  = "question",
    values_to = "response",
    names_transform = list(question = fct_inorder)
  ) %>%

  # Plot
  ggplot() +
  geom_histogram(aes(x = response)) +
  theme_bw() +
  facet_wrap(~question)
```



Seems like some questions have different means and variances from each other. For example, the answers to `ags11` and `ags12` are relatively flat, while the answers to `ags4` and `ags5` are more bunched up around the highest values.

We can also do some healthy exploration of missingness in the dataset. For starters: what proportion of values are missing in each row?

```
dat_ags %>%

  # Calculate the proportion of missing values
  summarise_all(~ sum(is.na(.)) / (sum(is.na(.) + sum(!is.na(.))))) %>%

  mutate(across(everything(), round, 6)) %>%

  knitr::kable(title = "Proportion of Missing Responses in Each Column")
```

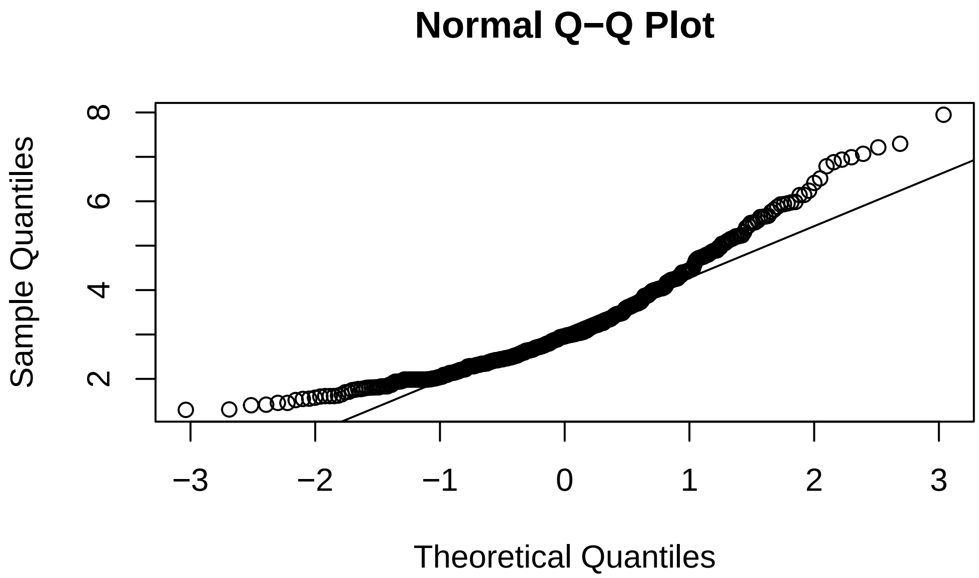
ags1	ags2	ags3	ags4	ags5	ags6	ags7	ags8	ags9	ags10	ags11	ags12
1.1e-05	5e-06	5e-06	1.6e-05	1.6e-05	1.1e-05	1.6e-05	1.6e-05	1.1e-05	2.2e-05	1.1e-05	1.6e-05

That's very little missingness. Probably no need to do multiple imputation here.

The authors also do a preliminary test of whether the responses are normally distributed, since this is one of the fundamental assumptions of maximum likelihood estimation. Kristoffer Magnusson has created a cool interactive teaching tool that nicely illustrates this point. It is worth remembering that we *do not* make this type of assumption for linear regression in general – only for maximum likelihood estimates. All we need assume for linear regression is that the *residuals* are normally distributed, as opposed to the data themselves. This common misunderstanding leads to what Richard McElreath has called 'histomancy'.

To evaluate the assumption of normalness underlying maximum likelihood estimation, the authors do what seems to be a multivariate version of a classic 'normal probability plot'. These are explained nicely in this stack exchange thread. They also produce some of the classic tests of skew and kurtosis, which I don't want to get into here. This youtuber has nice introductory videos about these topics.

```
# Run the Mardia tests for normalness
mardia.object <- psych::mardia(dat_ags)
```



```r
# Plot the multivariate version of the normal probability plot
plot(mardia.object)

# Present the outputs we're interested in
tibble(
  "Skew" = mardia.object$skew,
  "Skew p-value" = mardia.object$p.skew,
  "Kurtosis" = mardia.object$kurtosis,
  "Kurtosis p-value" = mardia.object$p.kurt
) %>%

  knitr::kable()
```

Skew	Skew p-value	Kurtosis	Kurtosis p-value
2359.475	0	40.52999	0

The plotted points don't seem to fit the straight line super well, which suggests that the normalness assumption may not hold here. Also, the hypothesis tests for skew and kurtosis return some mighty low p-values, suggesting that we've got lots of each of them. So maybe maximum likelihood estimation isn't such a good idea here?

The authors proceed with it anyway for pedogogical reasons, because they want to illustrate how the maximum likelihood estimates differ from estimates arrived at using other methods,

The researchers who collected the data do what good factor analysts do: they look to the literature to set up some clear and specific candidate hypotheses, and see the degree to which this new data is compatible with each of them.

One of the candidate hypotheses is that a person's toxic striving energy ('achievement goal orientedness'?) is secretly driven by four platonic unobservable things, namely:

1. Mastery Approach 'MAP' (eg. *"I want to learn as much as possible")*;
2. Mastery Avoidant 'MAV' (eg. *"I want to avoid learning less than I possibly could"*);
3. Performance Approach 'PAP' (eg. *"I want to do well compared to other students");*
4. Performance Avoidant 'PAV' (eg. *"It is important for me to avoid doing poorly compared to other students"*)

But another theory says that actually the 'Mastery' variables are just one monolithic thing, so really there are only 3 factors, namely 'Mastery', 'PAP', and 'PAV'.

These will be the two candidate hypotheses we're gonna test via factor analysis.

```
dat_ff <- foreign::read.spss('data/finch-and-french/performance.data.sav')

# Seems like I need to only use the first 12 columns I think?
dat_ff <- dat_ff %>%

  as_tibble() %>%

  select(1:12)
```

2.2 Example 2:

```
dat_ff <- foreign::read.spss('data/finch-and-french/performance.data.sav')

# Seems like I need to only use the first 12 columns I think?
dat_ff <- dat_ff %>%

  as_tibble() %>%

  select(1:12)
```

Next we'll do another example from Finch (2015), from the SEM chapter on page 62. They say it is important to CFA first before testing the relationships between the latent variables a la SEM, because we first want to make sure those latent variables actually seem good. This example uses a different dataset than the previous one.

> [W]e must ascertain whether the proposed model structure is supported by our data, which we will do by fitting a CFA to each of the latent variables...

The authors fit a CFA for each of the latent variables in isolation, namely Mastery, Self-Oriented Perfectionism, and 'ATTC', which seems to mean 'Attention Control'

2.3 Example 3:

Now we'll look at an example from Kline (2011), chapter 13.

Load the data:

```
dat_kline <- read_csv('data/kline/kabc-amos.csv')
```

```
Warning: One or more parsing issues, call `problems()` on your data frame for details,
e.g.:
  dat <- vroom(...)
  problems(dat)
```

```
Rows: 11 Columns: 10
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): rowtype_, varname_
dbl (8): HM, NR, WO, GC, Tr, SM, MA, PS

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

2.4 Example 4:

Lastly, let's walk through an example from the lavaan documentation

References

Finch, French, W. Holmes. 2015. *Latent Variable Modeling with r.*

Gorsuch, Richard L. 1983. *Factor Analysis, 2nd Edition.*

Kline, Rex B. 2011. *Principles and Practice of Structural Equation Modeling.*