$x \rightarrow y$

Regression recap II

Revisiting aspects of regression models that we will need for CFA and SEM

Regression recap II

Today's goal:

Go over regression assumptions, and how they apply to CFA and SEM.

Outline:

- Positive definiteness
- Near-perfect correlations
- Outliers
- Normality
- Missing data

Positive definiteness

An important assumption in CFA and SEM

Positive definiteness

- CFA and SEM use the covariance matrix as their basis
  - In fact, you can run some CFA and SEM analyses without using any raw data!
  - However, this is not true if variables are ordinal (e.g. 5-point items)
- The covariance matrix needs to be **positive definite**
  - Technically speaking, it needs to have an inverse and positive eigenvalues

Positive definiteness

What can cause nonpositive definiteness?

- Perfect or near-perfect correlations (multicollinearity; between two or more variables)
- Outliers (or data entry errors)
- Missing data
- Non-continuous items (e.g. 5-point items, binary items)

In these cases, nonpositive definiteness is a possibility (not a given)

Also, problems may occur even with positive definiteness

Multicollinearity

Remember VIFs?

Multicollinearity

Both X_1 and X_2 are predictors of Y, but highly correlated with each other

- Correlation of X_1 with Y is .4 but controlling for X_2 it is .2
- Correlation of X_2 with Y is .4, but controlling for X_1 it is .2

Two possibilities:

- X_1 has a high b (e.g. $b_1 = .6$) and X_2 has a low b (e.g. $b_2 = .3$)
- X_1 has a low b (e.g. $b_1 = .3$) and X_2 has a high b (e.g. $b_2 = .6$)

Which one is correct?

Multicollinearity

In regression:

Problem: The wizard is having a hard time deciding on b_1 and b_2!

Consequence: b_1 and b_2 are untrustworthy

In CFA/SEM:

Problem: Some of the eigenvalues become very small

Consequences: analysis may fail to converge, or give nonsensical loadings

Multicollinearity

Tests for multicollinearity:

- High correlation between Xes
- Variance inflation factor (VIF), should be lower than 10 (or 5), and lower than 1 on average

VIF = 1 / (1 – R^2)

Where R^2 is the R^2 of the regression of this X with all other Xes

Multicollinearity

Multicollinearity is more likely to happen for 5-point scales, and even worse for binary (0/1) variables

- Fewer values = higher chance of perfect correlation

Note: We will also get this at the latent level, when two measurement scales are too highly correlated to be considered separate

- In this case we call the problem a lack of "discriminant validity"

Outliers

Remember Cook's distances etc.?

Outliers

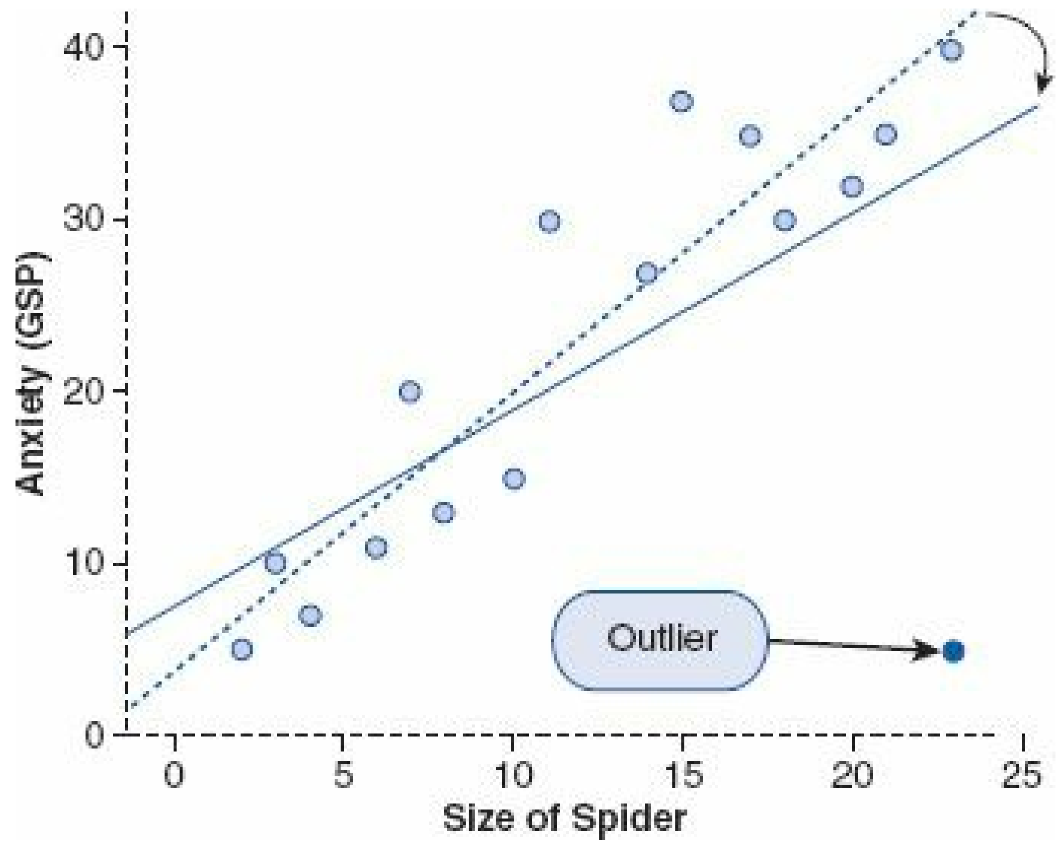
A data point that differs substantially from the model

- They have a very large residual (error)

Outliers can bias your regression coefficients

- How can we detect them?

Outliers rarely happen with 5- or 7-point items!



Univariate outliers

Standardize the variable

- Divide it by the standard deviation (this creates a z-score)

Assess the situation:

- $z > 3$: clear outliers

May need to use robust methods

- (see book for example)

Multivariate outliers

When the combination of values of a case is unusual

Can be detected using:

- Standardized residuals
- Cook's distances
- Hat values
- DFBeta and covariance ratio
- Mahalanobis distances (see book)

Solutions

What to do with outliers?

- Remove the score (only if you have good reasons to)
- Transform the data (see later)
- Replace the score (with the next highest score + 1, or mean + 3.29*sd)
- Use a robust test

Outliers

Note: In Rasch modeling we will pay special attention to outliers at the latent level

Under-fit:

- A person has several “easy” items wrong, but “hard” items correct
- An item is answered correctly by several “weak” persons, but incorrectly by several “strong” persons

Overfit:

- Answers of a person / to an item are too deterministic

Outliers

Scale mismatch:

- Some items are answered correctly by everyone or by nobody (hard to determine its difficulty!)
- Some persons have all items correct or all items incorrect (hard to determine their ability!)

$x \to y$

Normality

Skewness, kurtosis, etc.

$x \rightarrow y$ Normality

The **error distribution** of a model should be normal

- In most linear models, this means that the sampling distribution of our outcome value should be normal
- Why? Because outcome = (model) + error; model is fixed, so if the sample is normal, then the error is normal

We don't know the sampling distribution, so we look at the sample itself

- If a value is normally distributed within the sample, then the statistic (e.g. mean) is normal between samples as well

Normality

Typical deviations:

Skewness: the data is slanted to the left or to the right

- Kline: skewness > 3 is bad

Kurtosis: the data is “peaked” or “fat-tailed”

- Kline: kurtosis > 10 is bad

Outliers and limits can also cause non-normality

Detection

- Visually inspect
  - Use ggplot to create a histogram with normal curve
  - Check whether the QQ-plot (qplot) is a straight line
- Numerical check
  - Use stat.desc with norm=T (in the pastecs package) to get skewness, kurtosis, and the Kolmogorov-Smirnoff test
- If there are multiple conditions, also do this per condition
  - See cheat sheets for M&E I

Normality

Consequences: When your test assumes normality, deviations can result in biased test statistics

Overestimated or underestimated SEs and p-values

Solution: transform the data

This also helps with outliers, non-linear relationships, and heteroscedasticity

However, transformed results are harder to interpret!

Transformations

log transform:

```
transformed <- log(original + 1)
```

(we use +1, because log(0) does not exist!)

square root transform:

```
transformed <- sqrt(original)
```

reciprocal transform:

```
transformed <-1/(original + 1)
```

How can we interpret these?

Other solutions

Most modern CFA/SEM packages use robust estimation methods by default!

- But note that the complexity of these methods sometimes causes them to not converge

When data is binary, ordered, or count, we can use logistic, ordered logistic, and poisson models

- MPlus and lavaan have excellent functionality for this

Note: at the latent level, normality is not a problem, because CFA factors are approx. normally distributed by definition

Missing data

How to deal with it

Missing data

Types of missing data:

Missing Completely at Random (MCAR):

- Missing entries are unrelated to X and Y
- This is usually not a problem (it's just like having less data)

Missing at Random (MAR):

- Missing entries are unrelated to Y (not necessarily to X)
- Can result in some biases
- Values can be imputed, if you have auxiliary variables

Missing data

Types of missing data:

Missing not at Random (MNAR):

- Missing entries may be related to X or Y
- Example: dropout due to discomfort in HCI studies
- You can remove the cases with missing data, but even then it can be a problem due to sampling bias

Solutions

Listwise deletion

- Just delete all cases with missing data
- Okay for everything except MNAR
- Severely reduces power if there is a lot of missing data

Pairwise deletion

- E.g. if var A is missing, then remove case from A —> B, but not from B —> C
- Can result in non-positive definiteness!

Solutions

Substitution

Replace data by the overall mean, group-mean, or a prediction (e.g. based on regression with auxiliary vars)

This tends to results in underestimated SEs

Imputation

Use stochastic regression, pattern matching, or random hot-deck imputation to come up with the missing value

These methods try to get closest to the missing value as possible, and keep the SE unbiased

Solutions

FIML

- Split the data by "missingness pattern", fit a model on each subset, then combine the models

Multiple imputation

- Iteratively impute, fit the model, impute based on the model, refit the model, etc.
- Imputations are sampled stochastically

When available, FIML is the most reliable method

Final note

Scaling relative variances

CFA and SEM are based on the covariance matrix

CFA and SEM use iterative methods to create the best-fitting model

- At each step, it will look at how much improvement has been made

Final note

Let's say one variable ranges from 0 to 1000, and another from 0 to 10

- A small improvement on predicting one variable may look like a large improvement on predicting another!

This messes with the iterative improvement method

- This method tries to always get better, but with unbalanced variances, it sometimes gets worse instead

Solution: rescale variables to balance variances

- Manually, or using standardized algorithms (e.g. WLSMV)

"It is the mark of a truly intelligent person to be moved by statistics."

T H A N K S !

George Bernard Shaw