semopy

Structural equation models: latent variables, measured by indicators, linked by paths

Python cheatsheet · MATH60033A · Quantitative Methods in IB · used in session 9

```
import semopy
```

pip install semopy · the Python counterpart of R's lavaan

1 · Write the model as text

A model is a string: one line per relation, using three operators.

`=~`	**is measured by**	visual =~ x1 + x2 + x3
`~`	**is regressed on**	textual ~ visual
`~~`	**covaries with**	x1 ~~ x4

`desc = """ … """`

the **measurement** part: each latent variable on the left, its indicators on the right.

```
desc = """
visual  =~ x1 + x2 + x3
textual =~ x4 + x5 + x6
speed   =~ x7 + x8 + x9
"""
```

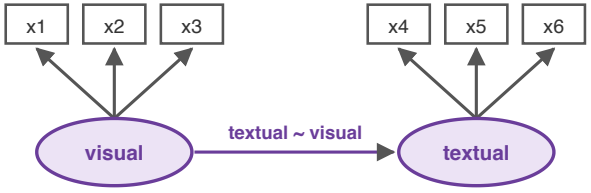
`textual ~ visual`

the **structural** part: a regression between latent variables.

`x1 ~~ x4`

a covariance between two residuals, when the questions share something the factors do not.

2 · Read it as a picture



visual =~ x1 + x2 + x3 textual =~ x4 + x5 + x6

oval = latent (not observed) · box = measured · arrow = effect

> Draw the diagram **before** writing the string: every arrow is one term in the model.

3 · Fit

`from semopy.examples import holzinger39`

the classic data set: 301 children, nine ability tests — ships with semopy.

```
data = holzinger39.get_data()
```

`fit = semopy.Model(desc)`

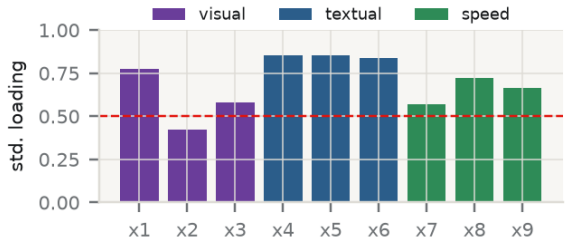
build the model from the string.

`fit.fit(data)`

estimate it, by maximum likelihood; the column names must match the string.

> Each latent variable needs a scale: semopy fixes the **first** loading of each factor to 1. That row has no standard error.

4 · Read the estimates



`print(fit.inspect(std_est=True))`

every parameter: estimate, **standardised** estimate (`Est. Std`), standard error, z, p.

`lval op rval`

how to read a row: `x2 ~ visual` is the loading of x2 on visual (semopy prints =~ lines as ~).

`ins[ins["op"] == "~~"]`

the variances and covariances, including those of the latent variables.

> Standardised loadings above about 0.5 say the indicator measures its factor well; x2 here is weak (0.42).

5 · Judge the fit

CFI	**0.931**	≥ 0.95 good · ≥ 0.90 acceptable
TLI	**0.896**	≥ 0.95 good · ≥ 0.90 acceptable
RMSEA	**0.092**	≤ 0.06 good · ≤ 0.08 acceptable

the three-factor model: acceptable at best

`print(semopy.calc_stats(fit).T)`

χ^2 and its p-value, CFI, TLI, RMSEA, AIC, BIC, … in one table.

`chi2 p-value`

almost always significant with real data and a decent sample: read it, but do not decide on it alone.

> Report **all** the indices, not the one that passes. A good fit does not make the model true — other models fit too.

6 · A full model: measurement and paths

Session 9's second example: 75 countries.

`from semopy.examples import political_democracy`

the data: x1–x3 measure industrialisation in 1960, y1–y8 democracy in 1960 and 1965.

`ind60 =~ x1 + x2 + x3`

measurement, as before — three latent variables.

```
model = """
  ind60 =~ x1 + x2 + x3
  dem60 =~ y1 + y2 + y3 + y4
  dem65 =~ y5 + y6 + y7 + y8
  dem60 ~ ind60
  dem65 ~ ind60 + dem60
"""
```

`dem65 ~ ind60 + dem60`

the structural paths: a regression whose variables are all latent.

7 · Pitfalls

`KeyError / variable not found`

a name in the string is not a column of the data: check spelling and case.

`three indicators per factor`

fewer than three and the factor is hard to identify; one indicator cannot carry a latent variable.