# Structural Equation Modeling

P.04 - Estimation Methods in SEM

November 07, 2022

---

## Lab Description

For this practical you will need the following packages:

- `lavaan`
- `semPlot`
- `psych`
- `ggplot2`

You can install and load these packages using the following code:

```
# Install packages.
install.packages(c("lavaan", "semPlot", "psych", "ggplot2"))

# Load the packages.
library(lavaan)
library(semPlot)
library(psych)
library(ggplot2)
```

### Exercise 1

Upon installing the `R` packages mentioned above perform the following:

b. Import the dataset `ELEMM1.csv` that is available in the folder for this practical on Canvas.

Set the working directory to the location where your data file has been downloaded and load the data.

```
# For example.
setwd("/Users/mihai/Downloads")

# Load data.
data_ex_1 <- read.csv("ELEMM1.csv")

# Inspect the data.
View(data_ex_1)
```

Quickly list the variables and their names.

```r
# List the variables.
str(data_ex_1)
```

```
## 'data.frame':    372 obs. of  22 variables:
##  $ ITEM1 : int  4 2 6 7 6 2 6 4 6 4 ...
##  $ ITEM2 : int  4 2 6 7 6 2 6 6 5 6 ...
##  $ ITEM3 : int  5 1 7 7 6 2 6 3 5 2 ...
##  $ ITEM4 : int  4 7 7 7 6 7 7 7 7 4 ...
##  $ ITEM5 : int  4 2 3 1 4 6 2 2 4 2 ...
##  $ ITEM6 : int  2 1 5 1 2 2 7 3 3 2 ...
##  $ ITEM7 : int  7 7 6 7 6 7 7 7 7 6 ...
##  $ ITEM8 : int  2 2 6 7 2 1 7 3 4 1 ...
##  $ ITEM9 : int  7 7 5 7 7 7 4 7 6 7 ...
##  $ ITEM10: int  2 1 4 1 6 1 1 3 4 2 ...
##  $ ITEM11: int  2 1 5 1 6 1 1 3 4 2 ...
##  $ ITEM12: int  6 4 4 1 7 7 1 6 6 6 ...
##  $ ITEM13: int  3 2 4 1 4 3 6 3 3 3 ...
##  $ ITEM14: int  4 2 4 4 7 2 6 3 6 5 ...
##  $ ITEM15: int  1 1 2 1 5 1 1 3 4 1 ...
##  $ ITEM16: int  1 1 5 1 2 2 7 3 4 2 ...
##  $ ITEM17: int  7 7 5 7 7 7 6 7 6 6 ...
##  $ ITEM18: int  6 4 6 5 7 6 2 6 6 6 ...
##  $ ITEM19: int  6 6 6 7 7 5 4 6 7 6 ...
##  $ ITEM20: int  2 1 2 4 3 2 2 2 3 2 ...
##  $ ITEM21: int  6 6 4 7 5 6 6 6 6 2 ...
##  $ ITEM22: int  2 1 3 1 2 2 1 6 3 2 ...
```

c. Inspect the *skewness* and *kurtosis* of `ITEM1` to `ITEM22` using the `psych` package. Do you see indications of severe deviations from normality?

Check out the documentation for the package `psych` on how to compute descriptive measures for your variables by running `??psych`. We can use the function `psych::describe`.

```r
# Describe the data using `psych`.
describe(data_ex_1)
```

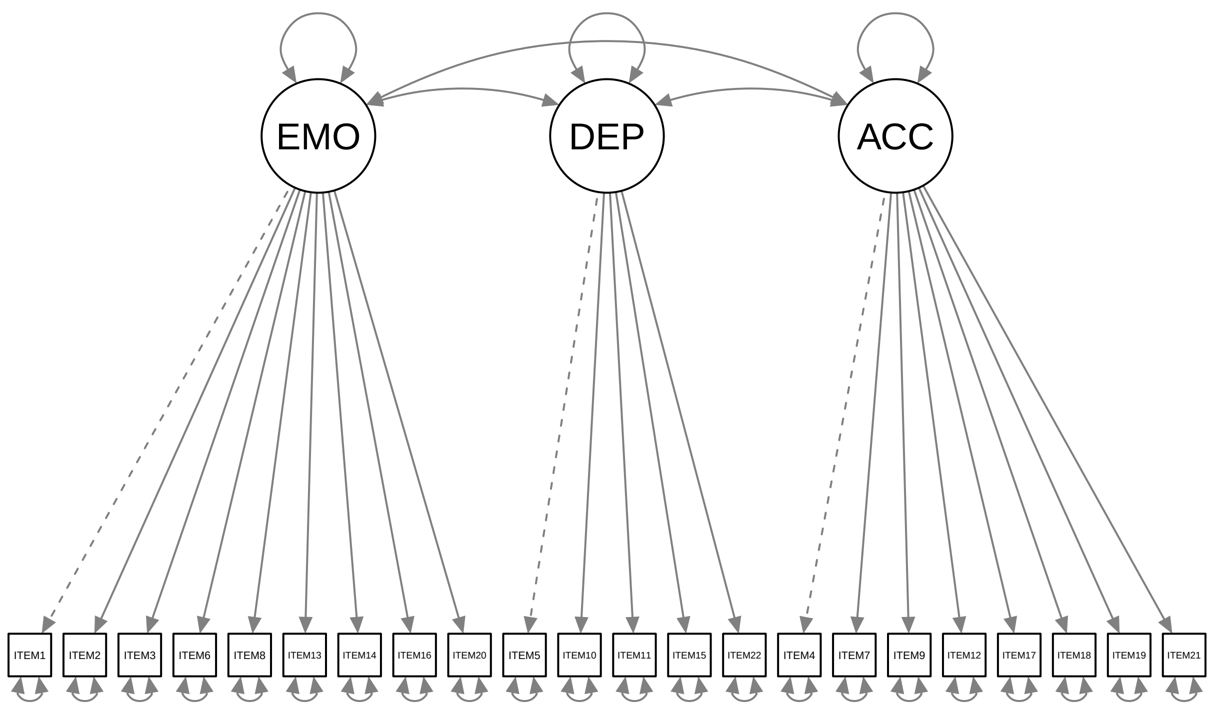
```
##        vars   n mean   sd median trimmed  mad min max range  skew kurtosis   se
## ITEM1     1 372 4.37 1.66    4.0    4.36 2.97   1   7     6 -0.11    -1.17 0.09
## ITEM2     2 372 4.87 1.55    5.0    4.97 1.48   1   7     6 -0.50    -0.71 0.08
## ITEM3     3 372 3.53 1.73    3.0    3.49 1.48   1   7     6  0.32    -1.11 0.09
## ITEM4     4 372 6.30 1.00    7.0    6.50 0.00   2   7     5 -1.80     3.63 0.05
## ITEM5     5 372 2.20 1.49    2.0    1.92 1.48   1   7     6  1.32     0.91 0.08
## ITEM6     6 372 2.71 1.58    2.0    2.50 1.48   1   7     6  0.92    -0.01 0.08
## ITEM7     7 372 6.31 0.84    6.0    6.46 1.48   2   7     5 -1.64     3.77 0.04
## ITEM8     8 372 3.04 1.73    2.0    2.89 1.48   1   7     6  0.74    -0.61 0.09
## ITEM9     9 372 6.03 1.32    7.0    6.29 0.00   1   7     6 -1.54     1.84 0.07
## ITEM10   10 372 2.20 1.45    2.0    1.96 1.48   1   7     6  1.20     0.56 0.08
## ITEM11   11 372 2.24 1.53    2.0    1.97 1.48   1   7     6  1.27     0.80 0.08
## ITEM12   12 372 5.70 1.19    6.0    5.86 1.48   1   7     6 -1.31     1.84 0.06
## ITEM13   13 372 3.59 1.68    3.5    3.52 2.22   1   7     6  0.35    -0.79 0.09
## ITEM14   14 372 4.03 1.73    4.0    4.01 1.48   1   7     6  0.03    -0.94 0.09
```

```
## ITEM15   15 372 1.77 1.30    1.0    1.47 0.00   1   7     6  2.09     4.24 0.07
## ITEM16   16 372 2.47 1.44    2.0    2.28 1.48   1   7     6  0.97     0.16 0.07
## ITEM17   17 372 6.41 0.85    7.0    6.58 0.00   2   7     5 -1.97     5.06 0.04
## ITEM18   18 372 5.70 1.27    6.0    5.87 1.48   1   7     6 -1.23     1.34 0.07
## ITEM19   19 372 5.95 1.19    6.0    6.15 1.48   1   7     6 -1.48     2.21 0.06
## ITEM20   20 372 2.24 1.41    2.0    2.01 1.48   1   7     6  1.29     1.17 0.07
## ITEM21   21 372 5.85 1.27    6.0    6.06 1.48   2   7     5 -1.29     1.16 0.07
## ITEM22   22 372 2.58 1.58    2.0    2.35 1.48   1   7     6  1.06     0.18 0.08
```

There seems to be some indication of non-normality, but not too severe. This may warrant using a robust estimator.

d. Estimate the model in Figure 1 using the default Maximum Likelihood method.

```r
# Model syntax.
model_ex_1 <- "
    EMO =~ ITEM1 + ITEM2  + ITEM3  + ITEM6  + ITEM8  + ITEM13 + ITEM14 + ITEM16 + ITEM20
    DEP =~ ITEM5 + ITEM10 + ITEM11 + ITEM15 + ITEM22
    ACC =~ ITEM4 + ITEM7  + ITEM9  + ITEM12 + ITEM17 + ITEM18 + ITEM19 + ITEM21

    # Covariances between latent variables
    EMO ~~ DEP
    DEP ~~ ACC
    EMO ~~ ACC
"

# Estimate the model.
model_ex_1_fit_ml <- cfa(model_ex_1, data = data_ex_1, estimator = "ML")

# Visualize the model.
semPaths(model_ex_1_fit_ml, what = "paths", sizeMan = 3)
```



```r
# Model summary for the `ML` estimator.
summary(model_ex_1_fit_ml)
```

```
## lavaan 0.6-12 ended normally after 46 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        47
##
##   Number of observations                           372
##
## Model Test User Model:
##
##   Test statistic                               695.719
##   Degrees of freedom                               206
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   EMO =~
##     ITEM1             1.000
##     ITEM2             0.887    0.061   14.621    0.000
##     ITEM3             1.021    0.068   15.085    0.000
##     ITEM6             0.764    0.064   12.013    0.000
##     ITEM8             1.143    0.066   17.299    0.000
##     ITEM13            1.017    0.065   15.544    0.000
##     ITEM14            0.848    0.069   12.251    0.000
##     ITEM16            0.715    0.058   12.410    0.000
##     ITEM20            0.753    0.056   13.410    0.000
##   DEP =~
##     ITEM5             1.000
##     ITEM10            1.142    0.127    8.986    0.000
##     ITEM11            1.353    0.142    9.511    0.000
##     ITEM15            0.905    0.109    8.318    0.000
##     ITEM22            0.768    0.121    6.361    0.000
##   ACC =~
##     ITEM4             1.000
##     ITEM7             0.970    0.150    6.482    0.000
##     ITEM9             1.780    0.254    7.007    0.000
##     ITEM12            1.499    0.221    6.769    0.000
##     ITEM17            1.348    0.181    7.463    0.000
##     ITEM18            1.918    0.262    7.329    0.000
##     ITEM19            1.716    0.238    7.205    0.000
##     ITEM21            1.356    0.218    6.219    0.000
##
```

```
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   EMO ~~
##     DEP               0.701    0.099    7.061    0.000
##   DEP ~~
##     ACC              -0.172    0.035   -4.850    0.000
##   EMO ~~
##     ACC              -0.192    0.042   -4.537    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .ITEM1             1.128    0.095   11.861    0.000
##    .ITEM2             1.105    0.090   12.214    0.000
##    .ITEM3             1.301    0.108   12.031    0.000
##    .ITEM6             1.553    0.121   12.888    0.000
##    .ITEM8             0.852    0.081   10.553    0.000
##    .ITEM13            1.142    0.097   11.821    0.000
##    .ITEM14            1.804    0.140   12.844    0.000
##    .ITEM16            1.235    0.096   12.812    0.000
##    .ITEM20            1.075    0.085   12.585    0.000
##    .ITEM5             1.503    0.125   12.026    0.000
##    .ITEM10            1.169    0.107   10.901    0.000
##    .ITEM11            1.044    0.112    9.330    0.000
##    .ITEM15            1.106    0.093   11.838    0.000
##    .ITEM22            2.076    0.160   12.958    0.000
##    .ITEM4             0.802    0.062   12.901    0.000
##    .ITEM7             0.523    0.042   12.572    0.000
##    .ITEM9             1.117    0.093   11.952    0.000
##    .ITEM12            0.987    0.080   12.287    0.000
##    .ITEM17            0.375    0.035   10.739    0.000
##    .ITEM18            0.909    0.081   11.224    0.000
##    .ITEM19            0.844    0.073   11.557    0.000
##    .ITEM21            1.245    0.098   12.764    0.000
##     EMO               1.625    0.190    8.551    0.000
##     DEP               0.705    0.132    5.321    0.000
##     ACC               0.193    0.048    4.047    0.000
```

e. Re-estimate the model, but now use the Satorra-Bentler estimator to estimate the *MFTS*. How does the scaling factors relate to the unscaled $\chi^2$ value?

Now we are going to re-estimate the model using Maximum Likelihood with robust standard errors (SE) and a Satorra-Bentler scaled test statistic (i.e., $\chi^2$). The estimator we are interested in is called `MLM` in `lavaan`.

```r
# Re-estimate the model.
model_ex_1_fit_mlm <- cfa(model_ex_1, data = data_ex_1, estimator = "MLM")

# Model summary.
summary(model_ex_1_fit_mlm)
```

```
## lavaan 0.6-12 ended normally after 46 iterations
##
##   Estimator                                         ML
```

```
##   Optimization method                           NLMINB
##   Number of model parameters                        47
##
##   Number of observations                           372
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               695.719     567.753
##   Degrees of freedom                               206         206
##   P-value (Chi-square)                           0.000       0.000
##   Scaling correction factor                                  1.225
##     Satorra-Bentler correction
##
## Parameter Estimates:
##
##   Standard errors                           Robust.sem
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   EMO =~
##     ITEM1             1.000
##     ITEM2             0.887    0.040   22.391    0.000
##     ITEM3             1.021    0.053   19.310    0.000
##     ITEM6             0.764    0.070   10.974    0.000
##     ITEM8             1.143    0.059   19.366    0.000
##     ITEM13            1.017    0.062   16.340    0.000
##     ITEM14            0.848    0.058   14.584    0.000
##     ITEM16            0.715    0.066   10.826    0.000
##     ITEM20            0.753    0.061   12.303    0.000
##   DEP =~
##     ITEM5             1.000
##     ITEM10            1.142    0.152    7.509    0.000
##     ITEM11            1.353    0.162    8.368    0.000
##     ITEM15            0.905    0.123    7.366    0.000
##     ITEM22            0.768    0.122    6.284    0.000
##   ACC =~
##     ITEM4             1.000
##     ITEM7             0.970    0.128    7.563    0.000
##     ITEM9             1.780    0.322    5.529    0.000
##     ITEM12            1.499    0.241    6.232    0.000
##     ITEM17            1.348    0.200    6.757    0.000
##     ITEM18            1.918    0.298    6.435    0.000
##     ITEM19            1.716    0.287    5.978    0.000
##     ITEM21            1.356    0.227    5.984    0.000
##
## Covariances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   EMO ~~
##     DEP               0.701    0.106    6.608    0.000
```

```
##   DEP ~~
##     ACC              -0.172    0.036   -4.777    0.000
##   EMO ~~
##     ACC              -0.192    0.040   -4.796    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .ITEM1             1.128    0.093   12.177    0.000
##    .ITEM2             1.105    0.088   12.506    0.000
##    .ITEM3             1.301    0.106   12.317    0.000
##    .ITEM6             1.553    0.134   11.550    0.000
##    .ITEM8             0.852    0.082   10.450    0.000
##    .ITEM13            1.142    0.124    9.173    0.000
##    .ITEM14            1.804    0.142   12.730    0.000
##    .ITEM16            1.235    0.110   11.278    0.000
##    .ITEM20            1.075    0.137    7.860    0.000
##    .ITEM5             1.503    0.179    8.381    0.000
##    .ITEM10            1.169    0.147    7.959    0.000
##    .ITEM11            1.044    0.141    7.398    0.000
##    .ITEM15            1.106    0.153    7.220    0.000
##    .ITEM22            2.076    0.184   11.266    0.000
##    .ITEM4             0.802    0.113    7.124    0.000
##    .ITEM7             0.523    0.075    7.010    0.000
##    .ITEM9             1.117    0.149    7.487    0.000
##    .ITEM12            0.987    0.126    7.852    0.000
##    .ITEM17            0.375    0.056    6.635    0.000
##    .ITEM18            0.909    0.143    6.376    0.000
##    .ITEM19            0.844    0.111    7.622    0.000
##    .ITEM21            1.245    0.133    9.338    0.000
##     EMO               1.625    0.148   11.004    0.000
##     DEP               0.705    0.158    4.452    0.000
##     ACC               0.193    0.050    3.839    0.000
```

The Satorra-Bentler method adjusts the $\chi^2$ downwards because due to non-normality it is otherwise overestimated. In other words, it takes into account kurtosis.

To obtain roughly the same *MFTS* (i.e., $\chi^2$) we observed when we used `estimator = "ML"` we need Satorra-Bentler MFTS $\times$ scaling correction factor

f. Evaluate the fit of the model estimated in (e).

We observe a $\chi^2 = 567.753$ with $DF = 206$ and a $p$-value $< 0.001$. Hypothesis that model exactly reproduces data must be rejected.

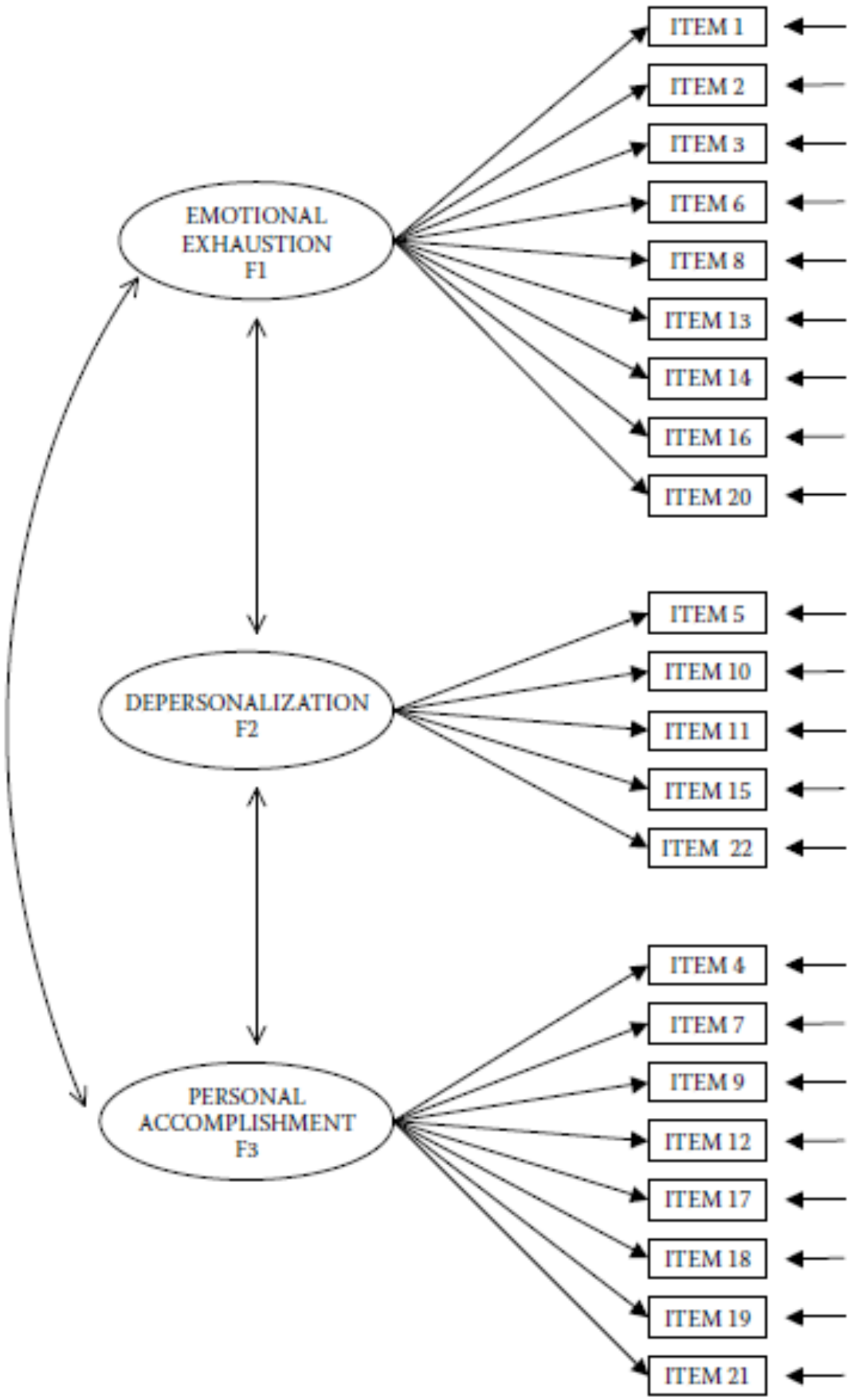



Figure 1: Hypothesized CFA model of factorial structure for the *Maslach Burnout Inventory* (MBI).

## Exercise 2

a. Import the dataset `bdihk2c2.csv` that is available in the folder for this practical on Canvas.

Set the working directory to the location where your data file has been downloaded and load the data.

```
# For example.
setwd("/Users/mihai/Downloads")

# Load data.
data_ex_2 <- read.csv("bdihk2c2.csv")

# Inspect the data.
View(data_ex_2)
```

Quickly list the variables and their names.

```
# List the variables.
str(data_ex_2)
```

```
## 'data.frame':    486 obs. of  23 variables:
##  $ linkvar: int  102 107 139 165 166 171 179 180 189 190 ...
##  $ gender : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ age    : int  14 14 14 14 14 14 14 14 14 14 ...
##  $ BDI2_1 : int  1 1 2 0 1 0 2 2 1 0 ...
##  $ BDI2_2 : int  0 0 0 0 0 0 2 1 0 0 ...
##  $ BDI2_3 : int  2 2 2 0 0 0 3 1 0 2 ...
##  $ BDI2_4 : int  0 0 0 0 0 0 1 1 0 0 ...
##  $ BDI2_5 : int  1 0 0 0 0 0 3 0 0 0 ...
##  $ BDI2_6 : int  0 0 1 0 0 0 2 0 1 0 ...
##  $ BDI2_7 : int  1 0 1 0 0 0 2 2 1 0 ...
##  $ BDI2_8 : int  3 0 1 0 0 0 1 0 1 0 ...
##  $ BDI2_9 : int  0 0 1 0 1 0 2 2 1 0 ...
##  $ BDI2_10: int  1 0 2 0 0 0 1 2 1 3 ...
##  $ BDI2_11: int  1 0 1 0 1 0 1 2 1 0 ...
##  $ BDI2_12: int  3 0 0 0 0 0 2 3 0 0 ...
##  $ BDI2_13: int  1 0 1 0 1 0 3 1 0 0 ...
##  $ BDI2_14: int  2 1 2 0 0 0 2 1 1 0 ...
##  $ BDI2_15: int  1 0 0 0 0 0 3 2 1 0 ...
##  $ BDI2_16: int  2 1 2 1 0 0 2 1 2 2 ...
##  $ BDI2_17: int  1 0 0 0 0 0 2 1 1 0 ...
##  $ BDI2_18: int  0 1 1 0 0 0 1 3 1 0 ...
##  $ BDI2_19: int  0 1 0 0 0 0 2 0 1 2 ...
##  $ BDI2_20: int  1 1 2 0 0 0 1 2 1 0 ...
```

b. Inspect the *skewness* and *kurtosis* of `BDI2_1` to `BDI2_20` using the `psych` package. Do you see indications of severe deviations from normality?

We can again use the `describe` function in the `psych` package.

```
# Describe the data using `psych`.
describe(data_ex_2)
```

```
##         vars   n    mean     sd median trimmed    mad min  max range  skew
## linkvar    1 486 1226.80 629.81 1189.5 1236.54 743.52 101 2275  2174 -0.05
```

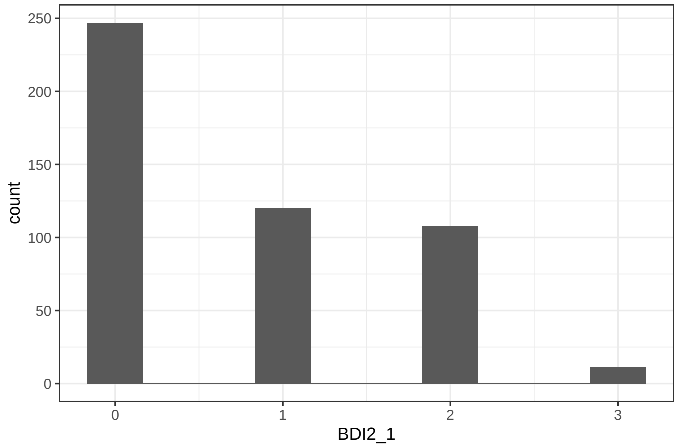
```
## gender     2 486    1.51   0.50    2.0    1.52   0.00   1    2     1 -0.05
## age        3 486   15.74   1.18   15.0   15.67   1.48  14   18     4  0.42
## BDI2_1     4 486    0.76   0.87    0.0    0.67   0.00   0    3     3  0.69
## BDI2_2     5 486    0.49   0.70    0.0    0.37   0.00   0    3     3  1.43
## BDI2_3     6 486    0.83   0.88    1.0    0.76   1.48   0    3     3  0.51
## BDI2_4     7 486    0.49   0.70    0.0    0.37   0.00   0    3     3  1.46
## BDI2_5     8 486    0.48   0.76    0.0    0.32   0.00   0    3     3  1.69
## BDI2_6     9 486    0.62   0.96    0.0    0.41   0.00   0    3     3  1.44
## BDI2_7    10 486    0.54   0.81    0.0    0.37   0.00   0    3     3  1.47
## BDI2_8    11 486    0.50   0.77    0.0    0.34   0.00   0    3     3  1.63
## BDI2_9    12 486    0.26   0.54    0.0    0.14   0.00   0    3     3  2.17
## BDI2_10   13 486    0.45   0.88    0.0    0.23   0.00   0    3     3  1.92
## BDI2_11   14 486    0.90   0.82    1.0    0.82   1.48   0    3     3  0.63
## BDI2_12   15 486    0.53   0.72    0.0    0.40   0.00   0    3     3  1.26
## BDI2_13   16 486    0.59   0.72    0.0    0.47   0.00   0    3     3  1.12
## BDI2_14   17 486    0.52   0.77    0.0    0.38   0.00   0    3     3  1.29
## BDI2_15   18 486    0.77   0.80    1.0    0.68   1.48   0    3     3  0.79
## BDI2_16   19 486    0.99   0.77    1.0    0.96   1.48   0    3     3  0.34
## BDI2_17   20 486    0.64   0.75    0.0    0.53   0.00   0    3     3  0.88
## BDI2_18   21 486    0.62   0.82    0.0    0.47   0.00   0    3     3  1.27
## BDI2_19   22 486    0.93   0.83    1.0    0.86   1.48   0    3     3  0.56
## BDI2_20   23 486    0.93   0.71    1.0    0.89   0.00   0    3     3  0.47
##         kurtosis    se
## linkvar    -1.16 28.57
## gender     -2.00  0.02
## age        -0.79  0.05
## BDI2_1     -0.83  0.04
## BDI2_2      1.82  0.03
## BDI2_3     -1.10  0.04
## BDI2_4      2.07  0.03
## BDI2_5      2.48  0.03
## BDI2_6      0.88  0.04
## BDI2_7      1.39  0.04
## BDI2_8      2.26  0.03
## BDI2_9      4.47  0.02
## BDI2_10     2.50  0.04
## BDI2_11    -0.20  0.04
## BDI2_12     1.15  0.03
## BDI2_13     0.97  0.03
## BDI2_14     0.69  0.04
## BDI2_15    -0.04  0.04
## BDI2_16    -0.46  0.03
## BDI2_17    -0.12  0.03
## BDI2_18     0.97  0.04
## BDI2_19    -0.38  0.04
## BDI2_20     0.15  0.03
```

It looks like we see some signs of non-normality.

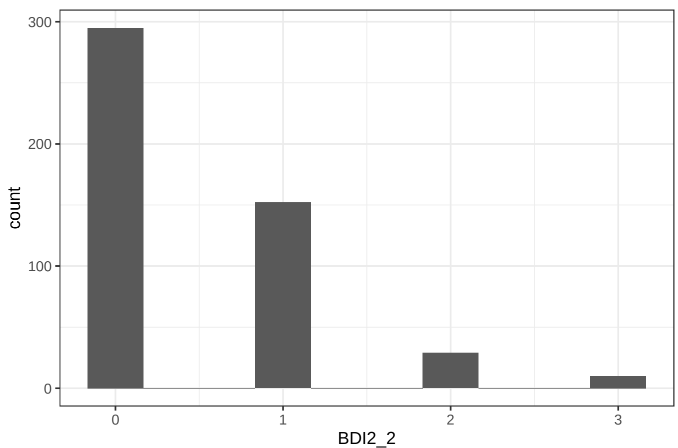
c. Develop histograms (using the `ggplot2` package) for the variables `BDI2_1` and `BDI2_20`. What do you learn from the inspection of these histograms?

- *Tip: When working with `R` you will often encounter parts that you just don't know how to implement, so don't be ashamed to Google things (e.g., "how to create and histogram using `ggplot2` in `R`").*

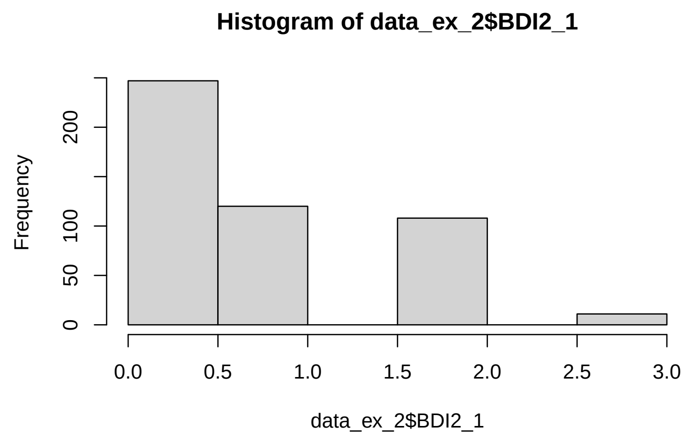
```
# Histogram for variable `BDI2_1`.
ggplot(data = data_ex_2) +
    geom_histogram(mapping = aes(BDI2_1), bins = 10) +
    theme_bw()
```



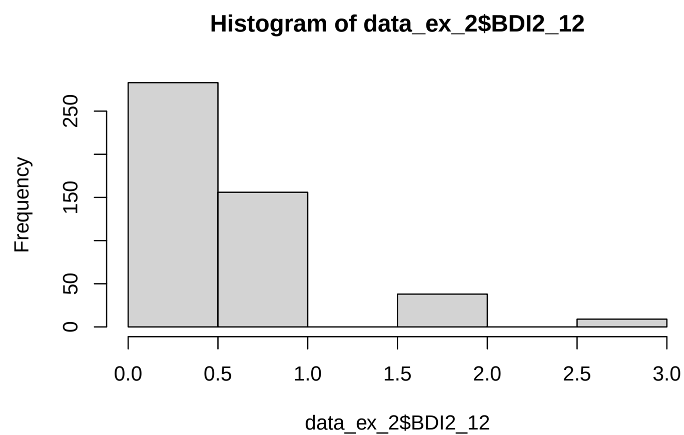

```
# Histogram for variable `BDI2_2`.
ggplot(data = data_ex_2) +
    geom_histogram(mapping = aes(BDI2_2), bins = 10) +
    theme_bw()
```




```
# Or you can also use the built-in function `hist` in `R` for this.
# Histogram for `BDI2_1` using `hist`
hist(data_ex_2$BDI2_1)
```




```
# Histogram for `BDI2_2` using `hist`
hist(data_ex_2$BDI2_12)
```




d. Estimate the model in Figure 2, but with the following additional constraints and model estimation specifications:
   1. Use `BDI2_3`, `BDI2_12`, and `BDI2_16` as marker variables.
   2. Constrain the variances of `F1`, `F2`, and `F3` to be equal.
   3. Fix the variance of `F4` to 1.
   4. Define the observed variables as ordered categorical variables.
   5. Use as estimator the *Mean and Variance Adjusted Weighted Least Squares* estimator (WLSMV).
   6. Evaluate the fit of this model.

*Note:* variables miss a `C` in the labeling, so `CBD` in picture is `BD` in the dataset.

First, we specify the model syntax.

```
# Model syntax.
model_ex_2 <- "
    # Measurement model.
    F1 =~ NA * BDI2_1 + BDI2_2 + 1 * BDI2_3 + BDI2_5 + BDI2_6 + BDI2_7 + BDI2_8 + BDI2_9 + BDI2_10 + BDI2_14
    F2 =~ NA * BDI2_4 + BDI2_11 + 1 * BDI2_12 + BDI2_13 + BDI2_17 + BDI2_19
    F3 =~ NA * BDI2_15 + 1 * BDI2_16 + BDI2_18 + BDI2_20

    # Structural model.
```

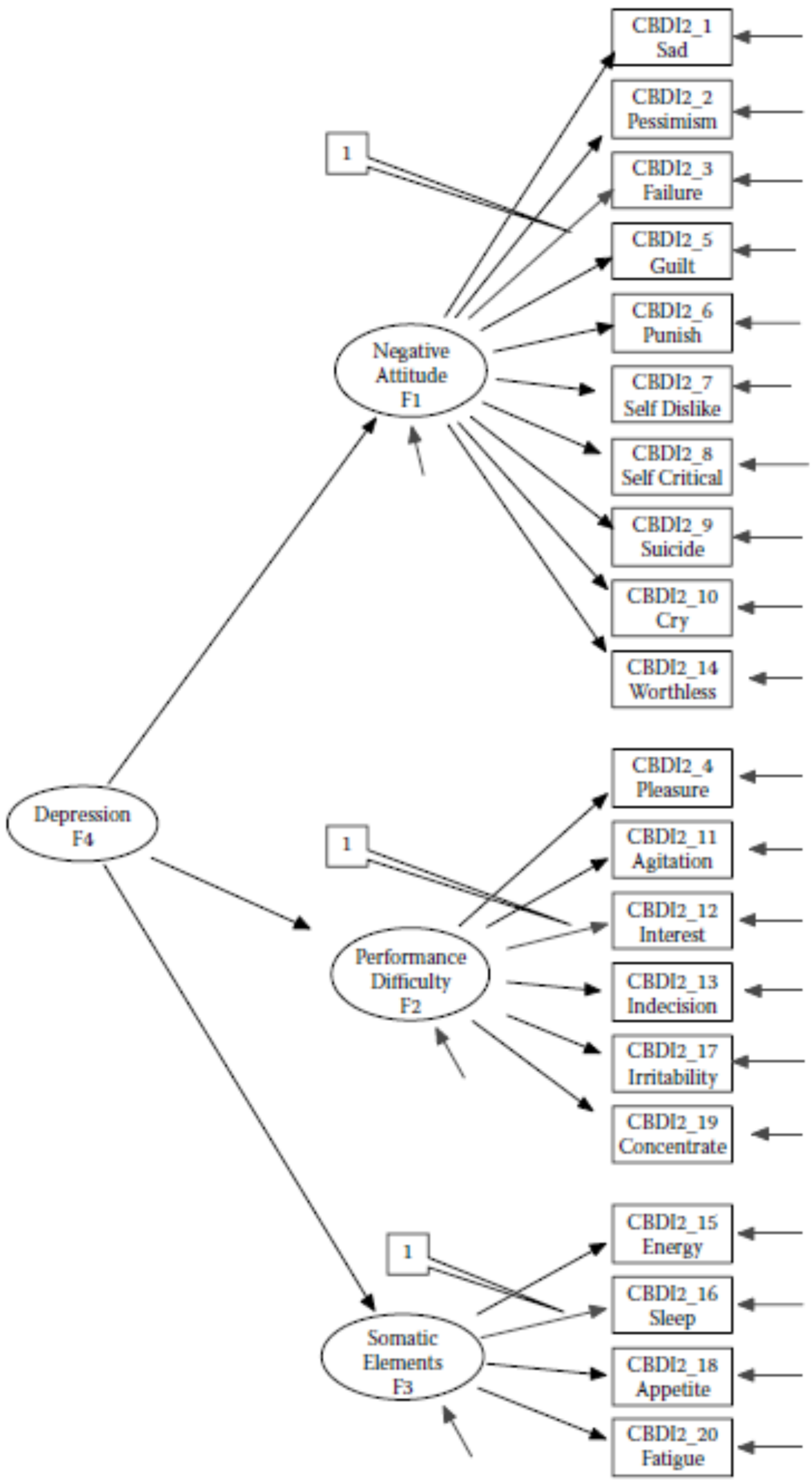


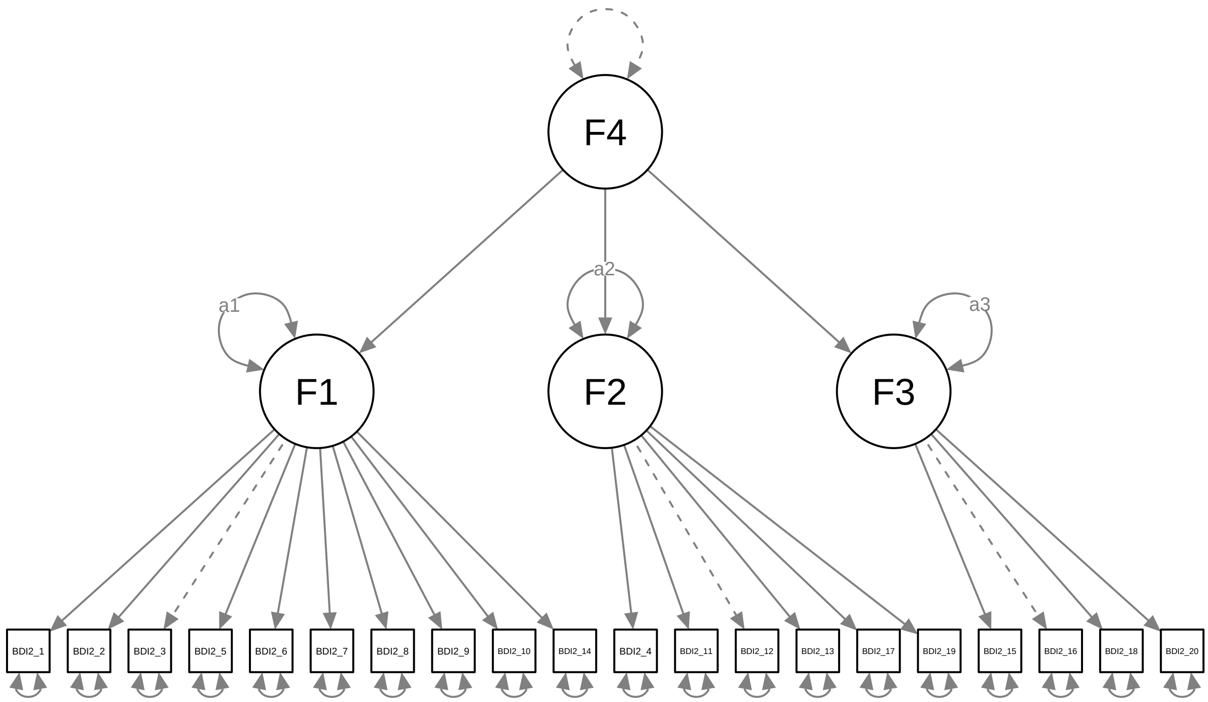

Figure 2: Hypothesized second-order model of factorial structure for the Chinese version of the *Beck Depression Inventory II*.

```
    F4 =~ NA * F1 + F2 + F3

    # Constrain the variance of the structural factor to 1.
    F4 ~~ 1 * F4

    # Add labels for the variances of the latent variables.
    F1 ~~ a1 * F1
    F2 ~~ a2 * F2
    F3 ~~ a3 * F3

    # Constrain the variances of the latent variables to be equal.
    a1 == a2
    a1 == a3
    a2 == a3
"
```

*What other shorter syntax would allow us to constrain the variances of the latent variables?*

Now we can estimate the model using the `ML` approach.

```
# Estimate the model using the `ML` approach.
model_ex_2_fit_ml <- cfa(model_ex_2, data = data_ex_2)

# Visualize the model.
semPaths(model_ex_2_fit_ml, what = "paths", sizeMan = 3)
```




```
# Model summary.
summary(model_ex_2_fit_ml)
```

```
## lavaan 0.6-12 ended normally after 45 iterations
##
##   Estimator                                         ML
```

```
##   Optimization method                           NLMINB
##   Number of model parameters                        43
##   Number of equality constraints                     3
##   Row rank of the constraints matrix                 2
##
##   Number of observations                           486
##
## Model Test User Model:
##
##   Test statistic                               394.871
##   Degrees of freedom                               169
##   P-value (Chi-square)                           0.000
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   F1 =~
##     BDI2_1            1.260    0.108   11.692    0.000
##     BDI2_2            1.026    0.087   11.793    0.000
##     BDI2_3            1.000
##     BDI2_5            0.880    0.088   10.006    0.000
##     BDI2_6            0.976    0.108    9.044    0.000
##     BDI2_7            1.231    0.102   12.094    0.000
##     BDI2_8            0.985    0.092   10.760    0.000
##     BDI2_9            0.600    0.062    9.643    0.000
##     BDI2_10           0.770    0.097    7.946    0.000
##     BDI2_14           1.193    0.098   12.226    0.000
##   F2 =~
##     BDI2_4            0.811    0.064   12.685    0.000
##     BDI2_11           1.080    0.075   14.387    0.000
##     BDI2_12           1.000
##     BDI2_13           0.943    0.066   14.334    0.000
##     BDI2_17           0.969    0.069   14.108    0.000
##     BDI2_19           0.949    0.076   12.450    0.000
##   F3 =~
##     BDI2_15           1.574    0.138   11.395    0.000
##     BDI2_16           1.000
##     BDI2_18           0.836    0.114    7.349    0.000
##     BDI2_20           1.307    0.118   11.088    0.000
##   F4 =~
##     F1                0.439    0.036   12.093    0.000
##     F2                0.493    0.030   16.256    0.000
##     F3                0.351    0.032   10.847    0.000
##
## Variances:
##                    Estimate  Std.Err  z-value  P(>|z|)
```

```
##     F4               1.000
##    .F1        (a1)   0.031    0.004    7.090    0.000
##    .F2        (a2)   0.031    0.004    7.090    0.000
##    .F3        (a3)   0.031    0.004    7.090    0.000
##    .BDI2_1           0.406    0.029   13.999    0.000
##    .BDI2_2           0.256    0.018   13.918    0.000
##    .BDI2_3           0.514    0.035   14.819    0.000
##    .BDI2_5           0.401    0.027   14.807    0.000
##    .BDI2_6           0.701    0.047   15.040    0.000
##    .BDI2_7           0.316    0.023   13.635    0.000
##    .BDI2_8           0.374    0.026   14.538    0.000
##    .BDI2_9           0.213    0.014   14.906    0.000
##    .BDI2_10          0.646    0.042   15.218    0.000
##    .BDI2_14          0.276    0.020   13.486    0.000
##    .BDI2_4           0.308    0.021   14.351    0.000
##    .BDI2_11          0.355    0.026   13.671    0.000
##    .BDI2_12          0.254    0.019   13.332    0.000
##    .BDI2_13          0.274    0.020   13.698    0.000
##    .BDI2_17          0.307    0.022   13.809    0.000
##    .BDI2_19          0.447    0.031   14.421    0.000
##    .BDI2_15          0.262    0.024   10.882    0.000
##    .BDI2_16          0.441    0.030   14.586    0.000
##    .BDI2_18          0.565    0.038   15.009    0.000
##    .BDI2_20          0.244    0.020   12.171    0.000
##
## Constraints:
##                                                |Slack|
##     a1 - (a2)                                    0.000
##     a1 - (a3)                                    0.000
##     a2 - (a3)                                    0.000
```

Now we can specify which variables should be treated as ordered variables and use the `WLSMV` estimator. First, let's store the names of those variables in a vector for convenience.

```r
# Store the variable names.
ordinal_variables_ex_2 <- c(
    "BDI2_1", "BDI2_2", "BDI2_3", "BDI2_4",
    "BDI2_5", "BDI2_6", "BDI2_7", "BDI2_8",
    "BDI2_9", "BDI2_10", "BDI2_11", "BDI2_12",
    "BDI2_13", "BDI2_14", "BDI2_15", "BDI2_16",
    "BDI2_17", "BDI2_18", "BDI2_19", "BDI2_20"
)

# Print the variable names.
print(ordinal_variables_ex_2)
```

```
##  [1] "BDI2_1"  "BDI2_2"  "BDI2_3"  "BDI2_4"  "BDI2_5"  "BDI2_6"  "BDI2_7"
##  [8] "BDI2_8"  "BDI2_9"  "BDI2_10" "BDI2_11" "BDI2_12" "BDI2_13" "BDI2_14"
## [15] "BDI2_15" "BDI2_16" "BDI2_17" "BDI2_18" "BDI2_19" "BDI2_20"
```

Estimate the model using the `WLSMV` estimator and use the variables names stored in `ordinal_variables_ex_2` to indicate to `lavaan` which variables should be treated as ordinal.

```r
# Estimate the model using the `WLSMV` estimator.
model_ex_2_fit_wlsmv <- cfa(
    model_ex_2,
    data = data_ex_2,
    ordered = ordinal_variables_ex_2,
    estimator = "WLSMV"
)

# Model summary.
summary(model_ex_2_fit_wlsmv)
```

```
## lavaan 0.6-12 ended normally after 35 iterations
##
##   Estimator                                       DWLS
##   Optimization method                           NLMINB
##   Number of model parameters                        83
##   Number of equality constraints                     3
##   Row rank of the constraints matrix                 2
##
##   Number of observations                           486
##
## Model Test User Model:
##                                               Standard      Robust
##   Test Statistic                               226.433     338.720
##   Degrees of freedom                               169         169
##   P-value (Chi-square)                           0.002       0.000
##   Scaling correction factor                                  0.788
##   Shift parameter                                           51.254
##     simple second-order correction
##
## Parameter Estimates:
##
##   Standard errors                           Robust.sem
##   Information                                 Expected
##   Information saturated (h1) model        Unstructured
##
## Latent Variables:
##                    Estimate  Std.Err  z-value  P(>|z|)
##   F1 =~
##     BDI2_1            1.174    0.063   18.558    0.000
##     BDI2_2            1.179    0.064   18.371    0.000
##     BDI2_3            1.000
##     BDI2_5            0.976    0.070   13.866    0.000
##     BDI2_6            0.874    0.068   12.779    0.000
##     BDI2_7            1.234    0.068   18.080    0.000
##     BDI2_8            1.035    0.064   16.163    0.000
##     BDI2_9            0.959    0.075   12.723    0.000
##     BDI2_10           0.831    0.079   10.485    0.000
##     BDI2_14           1.233    0.068   18.040    0.000
##   F2 =~
##     BDI2_4            0.881    0.042   20.903    0.000
```

```
##     BDI2_11           0.936    0.040   23.162    0.000
##     BDI2_12           1.000
##     BDI2_13           0.928    0.042   21.899    0.000
##     BDI2_17           0.933    0.044   21.230    0.000
##     BDI2_19           0.805    0.044   18.133    0.000
##   F3 =~
##     BDI2_15           1.492    0.109   13.677    0.000
##     BDI2_16           1.000
##     BDI2_18           0.805    0.104    7.752    0.000
##     BDI2_20           1.400    0.110   12.776    0.000
##   F4 =~
##     F1                0.606    0.033   18.464    0.000
##     F2                0.770    0.027   28.794    0.000
##     F3                0.503    0.040   12.657    0.000
##
## Intercepts:
##                    Estimate  Std.Err  z-value  P(>|z|)
##    .BDI2_1            0.000
##    .BDI2_2            0.000
##    .BDI2_3            0.000
##    .BDI2_5            0.000
##    .BDI2_6            0.000
##    .BDI2_7            0.000
##    .BDI2_8            0.000
##    .BDI2_9            0.000
##    .BDI2_10           0.000
##    .BDI2_14           0.000
##    .BDI2_4            0.000
##    .BDI2_11           0.000
##    .BDI2_12           0.000
##    .BDI2_13           0.000
##    .BDI2_17           0.000
##    .BDI2_19           0.000
##    .BDI2_15           0.000
##    .BDI2_16           0.000
##    .BDI2_18           0.000
##    .BDI2_20           0.000
##    .F1                0.000
##    .F2                0.000
##    .F3                0.000
##     F4                0.000
##
## Thresholds:
##                    Estimate  Std.Err  z-value  P(>|z|)
##     BDI2_1|t1         0.021    0.057    0.363    0.717
##     BDI2_1|t2         0.691    0.062   11.118    0.000
##     BDI2_1|t3         2.002    0.126   15.937    0.000
##     BDI2_2|t1         0.271    0.058    4.707    0.000
##     BDI2_2|t2         1.403    0.083   16.952    0.000
##     BDI2_2|t3         2.042    0.130   15.712    0.000
##     BDI2_3|t1        -0.088    0.057   -1.541    0.123
```

```
##     BDI2_3|t2         0.602    0.061    9.895    0.000
##     BDI2_3|t3         2.085    0.135   15.449    0.000
##     BDI2_5|t1         0.369    0.058    6.329    0.000
##     BDI2_5|t2         1.350    0.080   16.789    0.000
##     BDI2_5|t3         1.786    0.106   16.858    0.000
##     BDI2_6|t1         0.336    0.058    5.789    0.000
##     BDI2_6|t2         1.001    0.069   14.594    0.000
##     BDI2_6|t3         1.337    0.080   16.743    0.000
##     BDI2_7|t1         0.325    0.058    5.609    0.000
##     BDI2_7|t2         1.148    0.073   15.752    0.000
##     BDI2_7|t3         1.761    0.104   16.927    0.000
##     BDI2_8|t1         0.331    0.058    5.699    0.000
##     BDI2_8|t2         1.325    0.079   16.695    0.000
##     BDI2_8|t3         1.761    0.104   16.927    0.000
##     BDI2_9|t1         0.807    0.064   12.571    0.000
##     BDI2_9|t2         1.715    0.101   17.036    0.000
##     BDI2_9|t3         2.642    0.239   11.047    0.000
##     BDI2_10|t1        0.639    0.061   10.421    0.000
##     BDI2_10|t2        1.168    0.074   15.885    0.000
##     BDI2_10|t3        1.461    0.086   17.079    0.000
##     BDI2_14|t1        0.331    0.058    5.699    0.000
##     BDI2_14|t2        1.099    0.071   15.407    0.000
##     BDI2_14|t3        2.085    0.135   15.449    0.000
##     BDI2_4|t1         0.261    0.058    4.526    0.000
##     BDI2_4|t2         1.446    0.085   17.051    0.000
##     BDI2_4|t3         2.002    0.126   15.937    0.000
##     BDI2_11|t1       -0.380    0.058   -6.509    0.000
##     BDI2_11|t2        0.807    0.064   12.571    0.000
##     BDI2_11|t3        1.737    0.102   16.986    0.000
##     BDI2_12|t1        0.208    0.057    3.623    0.000
##     BDI2_12|t2        1.301    0.078   16.595    0.000
##     BDI2_12|t3        2.085    0.135   15.449    0.000
##     BDI2_13|t1        0.067    0.057    1.178    0.239
##     BDI2_13|t2        1.301    0.078   16.595    0.000
##     BDI2_13|t3        2.042    0.130   15.712    0.000
##     BDI2_17|t1        0.041    0.057    0.725    0.468
##     BDI2_17|t2        1.071    0.071   15.192    0.000
##     BDI2_17|t3        2.246    0.157   14.341    0.000
##     BDI2_19|t1       -0.397    0.059   -6.778    0.000
##     BDI2_19|t2        0.737    0.063   11.722    0.000
##     BDI2_19|t3        1.737    0.102   16.986    0.000
##     BDI2_15|t1       -0.176    0.057   -3.080    0.002
##     BDI2_15|t2        0.943    0.067   14.047    0.000
##     BDI2_15|t3        1.868    0.113   16.572    0.000
##     BDI2_16|t1       -0.596    0.061   -9.807    0.000
##     BDI2_16|t2        0.697    0.062   11.205    0.000
##     BDI2_16|t3        1.965    0.122   16.131    0.000
##     BDI2_18|t1        0.145    0.057    2.537    0.011
##     BDI2_18|t2        1.118    0.072   15.547    0.000
##     BDI2_18|t3        1.715    0.101   17.036    0.000
##     BDI2_20|t1       -0.620    0.061  -10.158    0.000
```

```
##     BDI2_20|t2        0.927    0.067   13.887    0.000
##     BDI2_20|t3        2.002    0.126   15.937    0.000
##
## Variances:
##                   Estimate  Std.Err  z-value  P(>|z|)
##     F4               1.000
##    .F1        (a1)   0.067    0.009    7.609    0.000
##    .F2        (a2)   0.067    0.009    7.609    0.000
##    .F3        (a3)   0.067    0.009    7.609    0.000
##    .BDI2_1           0.401
##    .BDI2_2           0.396
##    .BDI2_3           0.566
##    .BDI2_5           0.586
##    .BDI2_6           0.668
##    .BDI2_7           0.338
##    .BDI2_8           0.535
##    .BDI2_9           0.600
##    .BDI2_10          0.700
##    .BDI2_14          0.339
##    .BDI2_4           0.488
##    .BDI2_11          0.422
##    .BDI2_12          0.341
##    .BDI2_13          0.432
##    .BDI2_17          0.426
##    .BDI2_19          0.573
##    .BDI2_15          0.290
##    .BDI2_16          0.681
##    .BDI2_18          0.793
##    .BDI2_20          0.374
##
## Scales y*:
##                   Estimate  Std.Err  z-value  P(>|z|)
##     BDI2_1           1.000
##     BDI2_2           1.000
##     BDI2_3           1.000
##     BDI2_5           1.000
##     BDI2_6           1.000
##     BDI2_7           1.000
##     BDI2_8           1.000
##     BDI2_9           1.000
##     BDI2_10          1.000
##     BDI2_14          1.000
##     BDI2_4           1.000
##     BDI2_11          1.000
##     BDI2_12          1.000
##     BDI2_13          1.000
##     BDI2_17          1.000
##     BDI2_19          1.000
##     BDI2_15          1.000
##     BDI2_16          1.000
##     BDI2_18          1.000
```

```
##     BDI2_20           1.000
##
## Constraints:
##                                              |Slack|
##     a1 - (a2)                                  0.000
##     a1 - (a3)                                  0.000
##     a2 - (a3)                                  0.000
```

The hypothesis that model exactly reproduces data must be rejected.

*As you may have noticed, for some of the fit indices we also have a* robust *version. Check out this question for more information: https://stats.stackexchange.com/q/241896/116619.*

### Note on using the `WLSMV` estimator in `lavaan`

As we discussed, and also saw in the documentation of `lavaan`, when the data are continuous, the default estimator is the *Maximum Likelihood* approach (i.e., `ML`). The `ML` estimator hinges on the assumption that our data are multivariate normally distributed. For cases when the assumption of normality is violated, `lavaan` provides *robust* variants of the `ML` estimator. One such estimator is, for instance, the `MLM` estimator that uses the *Satorra-Bentler* scaled test statistic (Satorra & Bentler, 2001). In essence, these *robust* estimators differ in how the standard errors for the parameter estimates (i.e., used to calculate the $p$-values) and the $\chi^2$ test statistic are determined—i.e., in such a way that they are robust to violations of the normality assumption.

However, data are not always continuous. The `lavaan` package also provides specific estimators for such scenarios. For example, we can use the `WLS` estimator (i.e., based on the *Weighted Least Squares* approach) for categorical endogenous variables. Similarly, the `WLS` estimator also has *robust* variants, e.g., `WLSMV` (i.e., *Mean and Variance Adjusted Weighted Least Squares*), which uses the *Diagonally Weighted Least Squares* (i.e., `DWLS`) estimator, but "the full weight matrix to compute robust standard errors, and a mean- and variance-adjusted test statistic" (i.e., see this page in the `lavaan` documentation).

We can specify which variables should be treated as ordinal via the `ordered` argument in `lavaan`. Note that whenever we use the `ordered` argument, the estimator is automatically set to `WLSMV`. Variables specified as *ordinal* will be treated using a *threshold* structure. This boils down to assuming that a particular item has an underlying normal distribution (i.e., continuous Gaussian), but its distribution was discretized in our sample (i.e., split) at particular points (i.e., see Figure 3).

In this case, `lavaan` will use the threshold model to create a corresponding normally distributed latent variable for each ordinal item. This latent variable is then used in the measurement model (i.e., what we specified in the `lavaan` syntax) instead of our observed item. Therefore, the additional threshold parameters estimated for each variable specified as ordinal (e.g., `BDI2_1|t1`, `BDI2_1|t2`, and `BDI2_1|t3` for item `BDI2_1` in the example above) do not change the degrees of freedom of our model. Finally, not all endogenous variables need to be ordinal when the `WLSMV` estimator is used. We can have a mix of ordinal and continuous variables. However, for those that we indicated as ordinal via the `ordered` argument, `lavaan` will use the procedure presented above.

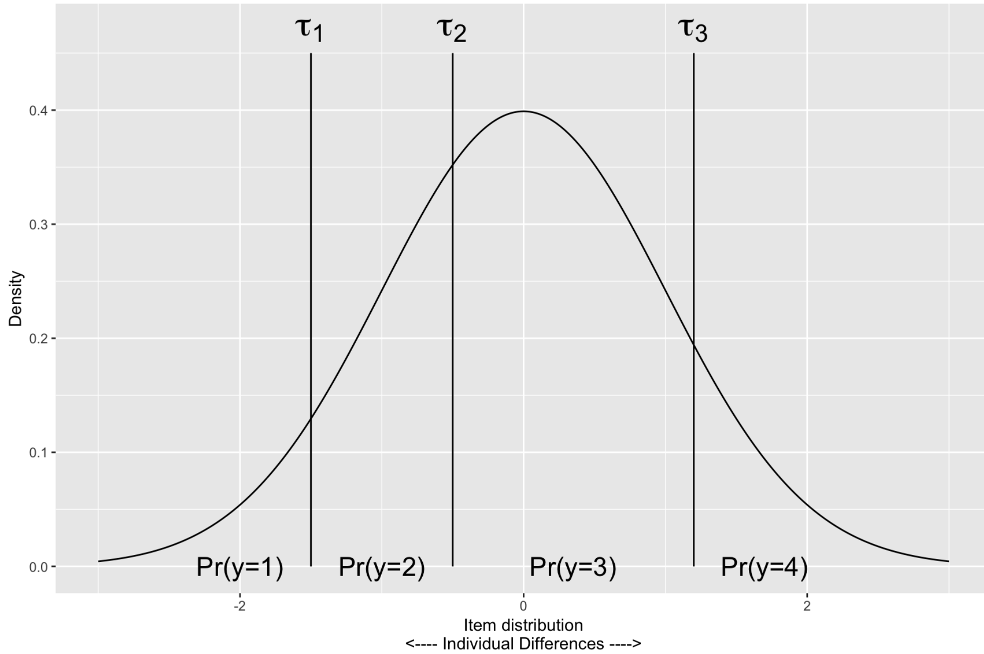



Figure 3: Graded threshold model with four answering anchors.

## References

Satorra, A., & Bentler, P. M. (2001). A scaled difference chi-square test statistic for moment structure analysis. *Psychometrika*, *66*(4), 507–514. https://doi.org/10.1007/BF02296192