Template Method - Nested Iterators

Adam Garber

Norwegian University of Science and Technology - A Course in `MplusAutomation`

May 31, 2021

Data source:

Change starting location to folder `23-template-method`

```
source("rep_functions.R")

change_here(glue("{project_location}/23-template-method"))

here()
```

```
## [1] "/Users/agarber/github/NTNU-workshop/23-template-method"
```

Load packages

```
library(MplusAutomation)
library(relimp)
library(tidyverse)
library(here)
library(janitor)
library(gt)
```

---

0. Write the Mplus `template.txt` file

- This is a special type of Mplus input file that includes the [[/init]] section at the top
- This section of code provides the instructions for doing "iterations" or "loops" to generate multiple input files
- Make sure to **UPDATE** the file-path in the template file so that the input files are generated in the correct location.

---

1. Write Mplus input files

```
createModels(here("PYDI_enumeration_template1.txt"))
```

```
## writing file: C1_PYDI_LCA.inp
## writing file: C2_PYDI_LCA.inp
## writing file: C3_PYDI_LCA.inp
## writing file: C4_PYDI_LCA.inp
## writing file: C5_PYDI_LCA.inp
## writing file: C6_PYDI_LCA.inp
## writing file: C7_PYDI_LCA.inp
## writing file: C8_PYDI_LCA.inp
```

---

2. Run models

- `recursive = TRUE` tells R to run all models within a parent folder. The recursive option is useful because when generating large batches of input files we can use the template file to create a nested set of sub-folder to organize models by type.

```
runModels(here("mplus_files"), recursive = TRUE)
```

---

3. Read models

```
output_enum <- readModels(here("mplus_files"), quiet = TRUE)
```

```
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
## <simpleError in startLine:endLine: NA/NaN argument>
```

---

4. Extract fit information from output files

- Any information in the ouptut or .gh5 files can be extracted and organized.
- This includes the summary statistics which are necessary to look at for choosing the number of classes.
- The models can be sorted based on a give statistic, such as the BIC.

```
showSummaryTable(output_enum, keepCols=c("Title","Parameters", "LL", "BIC","aBIC","BLRT_PValue"), sortBy
```

---

5. Mplus Object lists

- Click on the Mplus object in your R environment. This is an object including nested lists.
- This code tells R to look inside the object `output_enum` and extract the probabilities from the `.gh5` file associated with the 3-class model

```
conditional_probs <- as.data.frame(output_enum[["C3_PYDI_LCA.out"]]
                            [["gh5"]]
                            [["means_and_variances_data"]]
                            [["estimated_probs"]]
                            [["values"]]
                            [seq(2, 14, 2),]) # seq("from","to","by")

conditional_probs
```

```
##          V1        V2        V3
## 1 0.8205463 0.1732870 0.2748898
## 2 0.9877123 0.5810680 0.8469923
## 3 0.9621444 0.5305504 0.6980789
## 4 1.0000000 0.6431175 0.8178859
## 5 0.9918640 0.1466496 0.6845955
## 6 0.9936640 0.3763532 0.9552785
## 7 0.9949189 0.2167879 0.8959467
```
