Bayesian Psychometric
Measurement Using 'Stan'

# Diagnostic Modeling With R and Stan

StanCon 2023

**Instructor**: W. Jake Thompson, Ph.D.
wjakethompson@ku.edu
https://wjakethompson.com

## Course Description

Diagnostic classification models (DCMs; also called cognitive diagnostic models [CDMs]) are a powerful tool for analyzing assessment data (e.g., dichotomous or polytomously scored items) to understand human cognition and behavior. These models provide actionable feedback by reporting specific skills a respondent has mastered, or which traits are present. DCMs are intended to be multidimensional, allowing many attributes to be measured on a single assessment. For example, an educational assessment may assess student proficiency on several standards. In this workshop, we will cover how to easily estimate and evaluate DCMs with Stan and the R package measr. The measr package includes convenience functions for creating Stan scripts, estimating models, and extract relevant pieces of model output. The workshop will include hands-on examples of defining a DCM, estimating the model, and evaluating the fit (e.g., test- and item-level fit, classification accuracy and consistency, etc.).

### Goals:

- Get familiar with the basics of DCMs
- Understand how to use the R package measr to estimate DCMs with Stan
- Learn how to evaluate DCMs using functions built into measr

### Helpful pre-requisites:

- Familiarity with R. The focus of the tutorial will be using R to estimate and evaluate DCMs, and all exercises are written in R.

**About the instructor**: Jake earned his Ph.D. in Educational Psychology and Research from the University of Kansas. He is currently the Assistant Director of Psychometrics at Accessible Teaching, Learning, and Assessment Systems (ATLAS). In that role, Jake is the Principal Investigator for a grant funded by the Institute for Education Sciences to develop new software for estimating and evaluating diagnostic models. His research interests include the applications of diagnostic models, Bayesian methods for model evaluation, R development, and data visualization.

## Course Outline

- Introduction to Diagnostic Models
    - Conceptual foundations
    - Statistical foundations
- Estimating Diagnostic models
    - Diagnostic models with Stan
    - Diagnostic models with R and measr
    - Extracting model components
    - Defining prior distributions

- Evaluating Diagnostic models
  - Absolute model- and item-level fit
  - Model comparisons with loo
  - Reliability

All materials are available on the tutorial's website: https://stancon2023.measr.info.

## Installation Instructions

To prepare for the workshop, please install the following:

- A recent version of R (≥ 4.2.0), which is available for free at https://cran.r-project.org/
- (Recommended) A recent version of RStudio Desktop (≥ 2023.06.0-421), available for free at https://posit.co/download/rstudio-desktop/
- The R packages we will use, which you can install by connecting to the internet, opening RStudio, and running the following code in the console:

```
install.packages(c("measr", "fs"), repos = "https://cran.rstudio.com/",
                 dependencies = TRUE)
```

### Optional

By default, measr using rstan as a backend for estimating models. However, the version of rstan on CRAN is outdated. Therefore, I recommend using an updated version of rstan from the stan-dev/r-packages repository, or using the cmdstanr interface to CmdStan.

#### Updated rstan

You can download an updated version of rstan by running the following code:

```
install.packages(
  c("StanHeaders", "rstan"),
  repos = c("https://mc-stan.org/r-packages/", getOption("repos"))
)
```

To verify that rstan is installed correctly, run the following code. If it works, then things are set up properly! For additional help installing rstan, see the RStan Getting Started help page.

```
example(stan_model, package = "rstan", run.dontrun = TRUE)
```

#### cmdstanr

For cmdstanr, I recommend installing the development version directly from Github:

```
install.packages("remotes")
remotes::install_github("stan-dev/cmdstanr")
```

After installing cmdstanr, you will also need to install CmdStan, which can be by running:

```
cmdstanr::install_cmdstan()
```

To ensure that cmdstanr and CmdStan are installed properly, try running the following code. If it works, then you are good to go! For help with install cmdstanr, see the Getting Started with CmdStanR vignette.

```
file <- file.path(cmdstan_path(), "examples", "bernoulli", "bernoulli.stan")
mod <- cmdstan_model(file)
```