

Workshop Description

Longitudinal models are a broad class of approaches for understanding change and stability over time. There are many different approaches for understanding repeated measures data, but individual researchers are often exposed to only one framework (at best) and may be unaware of the full range of options for longitudinal modeling. In this workshop, we will navigate these issues, focusing on the multi-faceted decision process needed for model-selection among various popular modeling frameworks, specific modeling decisions that are required for appropriate analysis and inference, and steps for best connecting theories of change with model implementation. Where they arise, we will also compare the relative strengths of various modeling frameworks, including mixed effect (multilevel and generalized additive) and structural equation (latent curve and latent change score) models for fitting the various model options. The workshop will have two parts, one focused on developing a theoretical understanding of the models and their assumptions, and the second focused on gaining practical experience with fitting the models in R (with mentions of additional options available in other programs).

Workshop Planned Schedule

Day 1

9:00 – 10:30: Introduction to Longitudinal Modeling (Goals, Approaches, Data)
10:30 – 10:45: Break
10:45 – 12:00: Approaches to Incorporating Time
12:00 – 13:30: Lunch
13:30 – 14:45: Determining the Optimal Shape of Change Over Time
14:45 – 15:00: Break
15:00 – 16:30: Hack-a-thon Part 1
16:30 – 17:00: Q&A and Wrap-up

Day 2:

9:00 – 10:30: Covariates & Distal Outcomes
10:30 – 10:45: Break
10:45 – 12:00: Nested Data, Advanced Applications
12:00 – 13:30: Lunch
13:30 – 14:00: Q&A and Wrap-up
14:00 – 16:30: Hack-a-thon Part 2

Workshop Preparation

The following are the packages we will be using in this workshop. Please ensure they are installed properly on your local machine. If you would like to copy and paste the code below, you can do so from <https://e-m-mccormick.github.io/static/longitudinal-primer/01-introduction.html>:

```

packages <- c("utils", "tidyverse", "downloadthis",      # packages for data management
             "foreign", "MplusAutomation",             # packages for writing data
             "sjPlot", "broom", "kableExtra",          # packages for generating tables
             "nlme", "lme4", "lmerTest", "stats",      # packages for MLMs
             "mgcv", "gamm4", "itsadug",              # packages for GAMMs
             "lavaan",                                # packages for SEMs
             "ggplot2", "semPlot", "ggeffects",       # packages for visualization
             "interactions")                          # packages for probing interaction


if (length(setdiff(packages, rownames(installed.packages()))) > 0) {
  install.packages(setdiff(packages,
                          rownames(installed.packages())),
                  repos = "http://cran.us.r-project.org")
}
invisible(lapply(packages,
                 library,
                 character.only = TRUE))

```

I encourage everyone to bring their own data – ideally in a clean and ready-to-use format – so they can see how these models play outside the curated datasets that often go with workshops etc. However, I have also tried my best to ensure that our example datasets look as much like real data as possible! The code companion comes with several datasets that can be freely downloaded, and I encourage you to use them whenever they are helpful – some models might not be possible with your own data, so they are handy to have available. You can download the example datasets and read their descriptions from here: <https://e-m-mccormick.github.io/static/longitudinal-primer/07-datasets.html>

Datasets

Use the button below to download all datasets used in this primer. Note that these datasets have been either **simulated or synthesized** and therefore **are not to be used to test substantive research hypotheses**.

 [Download Codebook Datasets](#)

Descriptions

Executive Function

The `executive.function` dataset is a single-cohort longitudinal design, consisting of 342 adolescents, ages 11.74 – 15.33, assessed annually across 4 waves. The repeated measures of interest are DLPFC activation during an executive function task (`dlpfc*`) and behavioral scores on that task (`ef*`). Additional variables include time-invariant covariates, self-identified sex (`sex`) and assigned treatment group (`tx`), as well as age at observation (`age*`). A subset of this dataset is used as the `single.cohort` data in Chapter 3.

Instructions for Downloading R, RStudio, and R Packages

Downloading R

One of the nice things about R is that it is free to download and use. Just go to <https://www.r-project.org/> and click the “download R” link. Then run the installer.

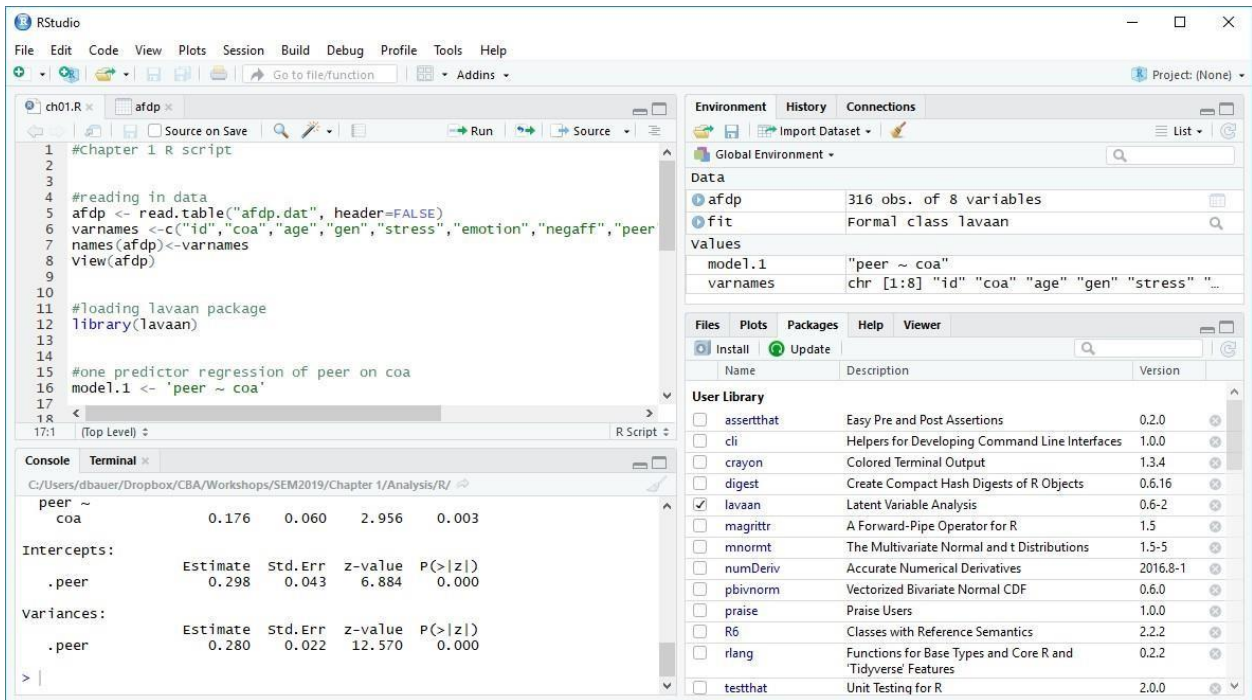
Downloading and Using the RStudio Interface

Another free program, RStudio, provides a convenient interface for using R. The installer for RStudio can be found at

<https://www.rstudio.com/products/rstudio/download/>

Select the RStudio Desktop / Open Source License for download and then run the installer. The

default layout for RStudio is shown below



The top left window shows an R script file. Saving R script in a file enables you to share code and re-run analyses quickly. You can highlight individual lines or blocks of lines and click “Run” to submit them to R and obtain the results.

The bottom left window shows the output and also displays a command prompt (the >). When you want to execute a function quickly and don’t care to save it to an R script, you can simply type it at the command prompt.

The top right window shows the objects currently in working memory. Here we have two data objects, **afdp** (the data file) and **fit** (containing model results). We also have a model syntax object called **model.1** and a vector of variable labels called **varnames**.

iv

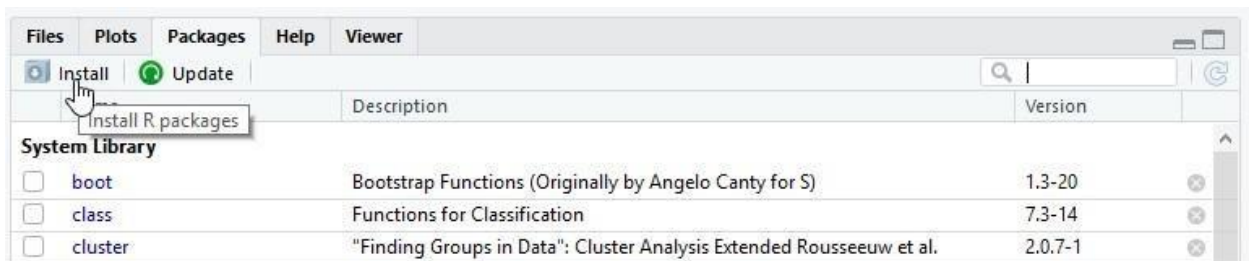
Finally, the bottom right window is useful for a variety of purposes, including viewing and installing packages, displaying plots, etc.

You can rearrange or resize these windows however you like.

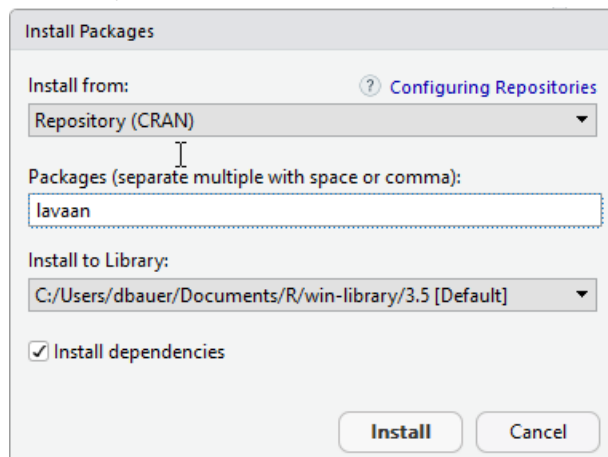
Installing and Using Lavaan and other R Packages

Although R comes with some built in functionality, much of what you can do with R comes through packages contributed by the scientific community. For this workshop, we will be primarily using the **lavaan** (**LA**tent **VA**riable **AN**alysis) package developed by Yves Rosseel from Ghent University. Although **lavaan** is still considered to be in beta-testing (i.e., experimental, meaning there is no guarantee everything will work as it should), it is widely used and considered to generate accurate results.

There are a couple ways you can download this and other packages in R. Using RStudio, the easiest way to install a package is to click the **packages** tab (in the lower right window under the default configuration) and then click the **install** button, as shown



This will bring up the Install Packages dialogue box shown below. Just type "lavaan" on the Packages line, as shown, then click "Install".



Alternatively, you can type the following at the command prompt:

```
> install.packages("lavaan", dependencies = TRUE)
```

Regardless of how you install **lavaan**, to actually use the package, you will need to also run the line

```
> library(lavaan)
```

v

The same steps are used for installing any other package in R (with the exception that some R packages are not provided through the CRAN repository accessed with these commands, in which case they must be downloaded directly from the developer).

Additional packages used in various demonstrations presented in these notes include **plyr**, **semTools**, and **psych**.

If you have questions about **lavaan**, you may wish to consult the discussion group. Go to <https://groups.google.com/d/forum/lavaan/> and join the group. Then you can email questions to lavaan@googlegroups.com.

