**R-Studio BASICS**

The R-Studio interface:

1. “Code or Script Editor”: This is where you will write all your code so that it can be saved and easily reproduced. This is called an R script of .R file type.
2. “R Console”: This is where code is executed and where output will appear including error messages.
3. “Workspace”: This is where variables, objects, or data-frames loaded into R can be viewed.
4. “Plots and files”: This is where plots will be viewable and where you can see new or existing files on your computer.



R Coding Language Properties:

* Syntax **is** case sensitive
* **Code line length:** no greater than 80 characters/columns (90 in newest version)
* **Variable names (objects):** name variables thoughtfully in a descriptive manner
* **Add comments (#**)**:** Use to annotate commands (green text).
* **End of line or code block:** R will automatically recognize the end of a line or code block. If you want to signal a line break within a given line use **;**

Uploading data:

* **Data types easily read by R:** Convert data file to **.csv** tab delimited files. When converting from SPSS leave the variable names (1st row) in the data file.
* **Create an R Project:** Make a separate folder for each lab and assignment where all files are located including data. This makes it so you do not have to use a breakable file-path to input data into R.
* **Data storage and organization:** Organize data, R-scripts, and other related files carefully in a project folder.

**Installing and loading “packages”:** Packages contain functions in them. To refer to a specific function within a package it is written as, dplyr::mutate

\*Text highlighted in yellow is code you change depending on the package needed

Install using:

install.packages(“tidyverse”)

Load packages using:

library(tidyverse)

**MplusAutomation:** MplusAutomation is a package developed for R which can be used to command Mplus to generate input files, estimate batches of models, and then extract key information from outputs. This is sometimes called a “wrapper” software, as two programs are interacting (i.e., we tell R what we want and it will command Mplus to execute the task).

Package “rhdf5”: We will need this package to read .gh5 files, or the plot information generated by Mplus. This package needs to be installed in a special way using the following code:

if (!requireNamespace("BiocManager", quietly = TRUE))

install.packages("BiocManager")

BiocManager::install("rhdf5")

library(rhdf5)