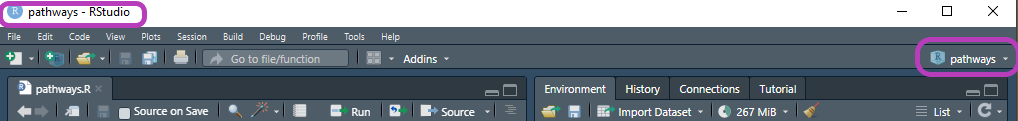
How to determine model pathways

# Requirements

* The `pathways` code works off of Amos output; therefore, the model needs to be run in Amos first. You will need to know the path and filename of the Amos output.
* R and RStudio need to be installed, as well as the `**tidyverse**` and `**glue**` packages.

# Open R project

Double-click on **pathways.Rproj** to open RStudio and the R project needed to determine pathways. Once RStudio is open, confirm that you are in the right project (upper left-hand corner should say “pathways - RStudio” and upper right-hand corner should say “pathways”).



If “**pathways.R**” script is not showing in upper left quarter of RStudio, open it by going to the Files pane (right bottom quarter of RStudio) and click on “pathways.R”.

# Set values

## Path location

* Line 13 sets where to find the folder where structural equation modelling (“SEM”) for the BC Public Service is done. PATH\_root will likely not change, so you probably don’t need to change this value. (If you do change it, use the //SFP.IDIR.BCGOV/… referencing so that it works regardless of which LAN letter a user has set, typically “W”, but you never know.)
* Line 17 sets the **YEAR** folder that you need. Change this accordingly. It will also be used in the name of the final pathways csv.
* Line 18 sets the rest of the path, including the file name of the Amos Output that should be already run. Change the **FILE\_sem** setting as needed. It will be joined to PATH\_root and YEAR so you will need to include any sub-folders to get there, as well as the actual Amos Output filename. Make sure this setting starts with “/” to add it properly after YEAR in the path used.

## Management & Roof drivers

You need to set the short-hand names of the management and roof drivers. You don’t need to set building block drivers as those are just everything else.

* Line 22 sets the management drivers. Change the **dr\_mgmt** setting if needed (unlikely but possible).
* Line 23 sets the roof drivers. Change the **dr\_roof** setting if needed (also unlikely).

Note that these settings must be within a **c()** if there is more than one such driver, and in quotation marks (e.g., dr\_mgmt <- c(“Exec”, “Sup”) )

## Negative paths

It is possible to get a statistically significant but negative estimate in the model. Because that is confusing to interpret (e.g., a negative impact of Teamwork on Empowerment means that as Teamwork scores improve, Empowerment scores drop), we typically exclude these connections.

* Line 28 sets whether to remove or keep any negative estimates. Set **remove\_neg\_estimates** to TRUE if you want to exclude any negative estimates (most likely scenario). If for some reason you decide to keep negative estimates, set **remove\_neg\_estimates** to FALSE (you should have a good reason to do so).

Note that TRUE or FALSE settings must be in capital letters and NOT in quotation marks.

## Path strength minimums

Path strengths can be Minimally Strong, Moderately Strong, Strong or Very Strong. Typically, we set these as Minimally Strong is 0.099 or less, Moderately Strong is 0.100 through 0.199, Strong is 0.200 through 0.299, and Very Strong is 0.300 or greater.

* Lines 32 through 34 set these minimums. Minimally Strong is automatically set as anything less than the minimum setting for Moderately Strong. It is highly unlikely that these settings will ever change.

# Run code

Once you have reviewed and changed any settings (lines 13 through 34), you can run the entire script with Ctrl+A (to select everything) and Ctrl+Enter (to run selection). The final output is a csv saved to the R project folder and is titled **“pathways\_YEAR.csv”** (where YEAR is what was set above).

If you want to run through the code bit by bit, make sure you run the `load packages` section first (lines 7 & 8). Then run the `SET VALUES` section (lines 11 through 34).

* The `load model` section (lines 37 though 50) reads in the Amos output and identifies various parts.
* The `pull connections` section (lines 51 though 96) pulls the “Standardized Regression Weights:” section, and formats so that path connections and estimates are lined up. It also determines whether the line relates to a driver -> driver connection (type “driver”) or is a driver -> question (type “question”) connection. For driver types, strength is determined (using the path strength minimums set above) and connection is set as an underscore-separated text field. This is called **std\_reg\_weights** and is used in code below. Then, the code checks for any negative estimates (and warns if there are any), and then removes any negative estimates if remove\_neg\_estimates was set to TRUE.
* The `path stats` section (lines 97 through 148) gets the driver order (bottom/ management to top/roof). As an FYI, it also flags if any sets of drivers are at the same level (saved as **multis**). Then path stats are calculated: for each driver, the block (i.e., Foundation, Building, Roof), number of connections from and to other drivers, number of underlying questions, and order is noted and called **path\_stats**.
* The `create all paths` section is the rest of the code. It has sub-sections:
  1. create ‘base’ to pull from
  2. run for Exec (z=1) – start paths beginning with Exec
  3. run through remaining non-roof drivers (z=2, …) – iterate over all other paths by running through remaining non-roof drivers
  4. create full path variable and number of drivers – variable listing all drivers in a path (e.g., Exec/Vision/Commit)
  5. calculate absolute path strengths – all weights are multiplied together (other than the last weight that connects to roof driver)
  6. remove duplicates – I think duplicates are made because of the multis and the way the code iterates
  7. determine relative path strengths and rank – group paths by number of drivers involved and then calculate the average path strength and standard deviation for each group; then set as “stronger” or “weaker” of each path to its group’s average and one standard deviation above its group’s average; also, determined each path’s rank, based on its strength
  8. format and save – reorder columns, drop a couple temporary variables, sort by path rank and save as “**pathways\_YEAR.csv**” in R project folder; the number of total paths and total paths per roof driver are printed to screen