Table of Contents

[Overview 1](#_Toc23325917)

[R Libraries 1](#_Toc23325918)

[Functions 1](#_Toc23325919)

[1. getKeyValues 1](#_Toc23325920)

[2. convertScores 2](#_Toc23325921)

[3. dfToUpper 2](#_Toc23325922)

[Inputs 2](#_Toc23325923)

[Score Sheet 2](#_Toc23325924)

[Sheet 1: Unadjusted 3](#_Toc23325925)

[Sheet 2: Adjusted 3](#_Toc23325926)

[Chemical Input Data 4](#_Toc23325927)

[Sheets 1 & 2: Unadjusted 4](#_Toc23325928)

[Output 4](#_Toc23325929)

[Running within RStudio 4](#_Toc23325930)

[Running from Command Line 5](#_Toc23325931)

# Overview

This script takes two inputs 1) scoreSheet and 2) inputData to produce unadjusted and adjusted scores for chemicals based on mapping between raw input values and the scoreSheet. It leverages R’s ability to parse character strings into logical/boolean expressions.

# R Libraries

1. readxl
   1. Used to load in .xlsx files
2. dplyr
   1. Used for data wrangling and piping
3. stringr
   1. Used for parsing and manipulating character strings

# Functions

## getKeyValues

* 1. Description: Parses the semi-colon separated key and value pairs from the Score Sheet input file
  2. Returns: A list which contains key and value pairs for each input variable parsed from the Score Sheet input file.
  3. Note: Will only work if the Score Sheet input file is present and formatted correctly

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| rubric | Either the adjusted or unadjusted sheet from the Score Sheet input .xlsx file |

## convertScores

* 1. Description: Loops through each key-value pair for a given input variable, parses the character strings into boolean expressions, and converts input variable values into their scale score.
  2. Returns: A modified version of the input df parameter with variable values for a chemical that have been converted to their scale scores using the keyValues and inputs parameters.

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| df | A dataframe to input the final converted output scores |
| chem | The chemical whose values are being converted |
| inputs | A list of variables being converted |
| keyValues | A list of key-value pairs from getKeyValues() |

## dfToUpper

* 1. Description: A helper method that converts all character types to uppercase. This helps ensure case sensitivity is not an issue for user input files
  2. Returns: A modified dataframe with all character types as uppercase values

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| df | A dataframe (typically the Chemical Input Data) |

# Inputs

## Score Sheet

**Variable Input:** scoreSheet

**Input Format:** path to the .xlsx file from the current directory

**Example file:** tableScoring.xlsx

**Description:** This Excel input is the scoring logic behind how the input data should be converted/recoded and how a resulting unadjusted and adjusted score should be calculated. This document can be customized to include additional scoring variables and multiple calculation parameters.

**Note:**

* Ensure proper units for data input values
* Ensure varName corresponds to input data variable names
* Ensure levels and scores are semi-colon separated in their corresponding order using proper R boolean syntax
* Ensure fx values are applied appropriately for adjusting scores
* Ensure it is an .xlsx file with an “adjusted” and “unadjusted” sheet
* Missing values are ignored for score calculations. This includes both inputData varName values and scoreSheet “levels” and “scores” variables

### Sheet 1: Unadjusted

|  |  |
| --- | --- |
| **Variable** | **Definition/Use** |
| Table | Numeric value of the table each input variable is grouped by |
| Description | Full text of score prompt/name |
| varName | Short-hand form of score prompt/name   * MUST correspond to the input data variable names |
| levels | List of the individual levels for scoring varName in a way that R can parse into logical statements   * MUST have the form “x OPERATOR value” * MUST use R logical/boolean expression syntax * Multiple levels MUST be semi-colon separated |
| scores | Ordered list of corresponding values for each level that the varName input value should be converted to |
| units | The units each input value should be converted to before the scoring process |

### Sheet 2: Adjusted

|  |  |
| --- | --- |
| **Variable** | **Definition/Use** |
| Table | Numeric value of the table each input variable is grouped by |
| tblNum | Clerical variable for grouping varName entries within their corresponding Table |
| fx | Which function should be used to calculate the adjustment. The **default is to sum** the values together   * sum = sum converted values * max# = find the max of these grouped variables with the same max# label * mn# = Minnesota/Non-Minnesota pair with the same mn# label. Take the MN value. If MN is empty, find the max of this pair |
| Description | Full text of score prompt/name |
| varName | Short-hand form of score prompt/name |
| levels | List of the individual levels for scoring varName in a way that R can parse into logical statements   * MUST have the form “x OPERATOR value” * MUST use R logical/boolean expression syntax * Multiple levels MUST be semi-colon separated |
| scores | Ordered list of corresponding values for each level that the varName input value should be converted to |

## Chemical Input Data

**Variable Input:** inputData

**Input Format:** path to the .xlsx file from the current directory

**Example file:** toyScoreData.xlsx

**Description:** This Excel input is the input data that must be formatted properly so that individual chemicals to be scored are in a “chemical” column, one per row. Have all variables to be scores formatted as columns with corresponding chemical row values. See the example “toyScoreData.xlsx” for clarification.

**Note:**

* Ensure your score data is a .xlsx file with an “adjusted” and “unadjusted” sheet
* Ensure variable values match the expected values from the “levels” variable from the scoreSheet input file’s sheets.
* All inputs are converted to uppercase to remove case-sensitivity concerns

### Sheets 1 & 2: Unadjusted

|  |  |  |  |
| --- | --- | --- | --- |
| **chemical** | **var1** | **var2** | **var3** |
| chemical1 | chem1-var1 value | chem1-var2 value | chem1-var3 value |
| chemical2 | chem2-var1 value | chem2-var2 value | chem2-var3 value |

**Note:**

* Ensure variable column names correspond to varName listed in the scoreSheet input file’s “unadjusted” sheet

# Output

**Example file:** ExposureTableScoresOutput.csv

**Description:** The output is a .csv file formatted like the inputData input file with the chemical data. Every row represents a different chemical within the chemical column. Each additional column represents the individual varName and score each input variable was converted/recoded to using the scoreSheet variable. The final “Table#Score” and Adjusted/Unadjusted Score columns represent the final scores for each chemical based on their varName input values by Table group and overall.

# Running within RStudio

* Change the input file paths for “scoreSheet” and “inputData” to the correct relative file path in your working directory

# Running from Command Line

**Command:** $ RScript “scriptName.R” pathToScoreSheetFile pathToInputDataFile

**Example:** $ RScript “scoringScript.R” input/scoreSheet.xlsx input/inputData.xlsx

**Note:**

* Ensure the “scoreSheet” input file path is listed first, followed by the “inputData” input file path
* Pass the parameters without single or double quotes as the relative path to the files from the working file directory