Article from

The Modeling Platform

April 2020

SOCIETY OF ACTUARIES
MODELING SECTION

Tidy Data Formats and Cloud Storage, Part 1

Tidy Data

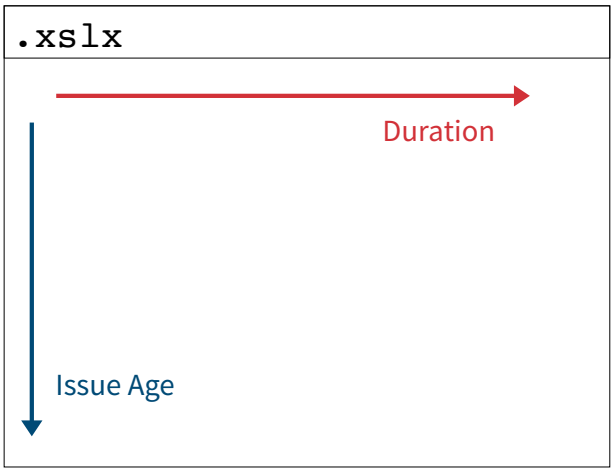
By Matthew Caseres

As of this writing, there are 3,010 Excel files in the mortality table repository. These files are in a format that does not play well with the databases R and Python. I converted all 2015 VBT and 2017 CSO tables (about 150 Excel files) into a format that can be easily joined to an experience study engine for A/E analysis. In this article, we discuss why we would want to reformat mortality tables and demonstrate how to reformat hundreds of select and ultimate tables quickly using R.

CURRENT MORTALITY TABLE FORMATS

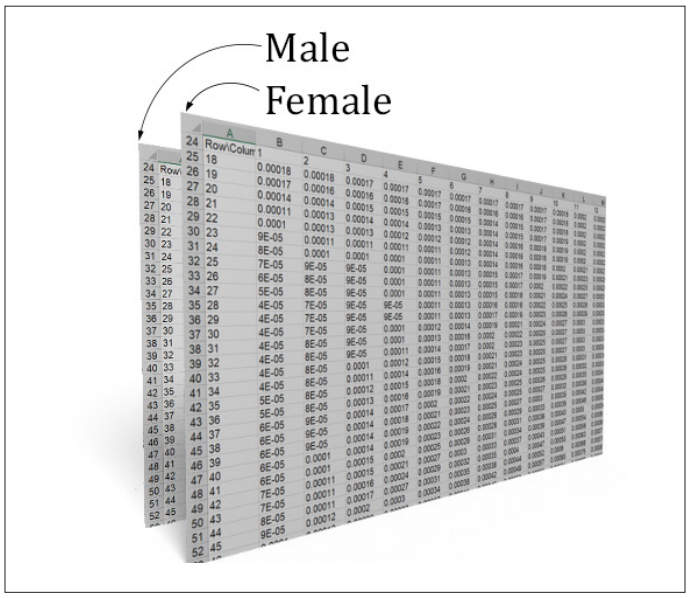
Let us consider a situation where the existing mortality tables are stored in Excel. For each demographic split, we have a single Excel file that contains the select and ultimate tables. In the select table worksheets, rows represent issue ages and columns represent durations. The ultimate table is stored as a single column with mortality rates for each attained age, as shown in Figure 1.

Figure 1
Structure of Mortality Table



Some mortality table sets have many demographics, which leads to many separate files, as shown in Figure 2. The 2017 CSO tables with preferred structure have 10 Excel files associated with them.

Figure 2
Structure of Mortality Table Set With Multiple Demographics



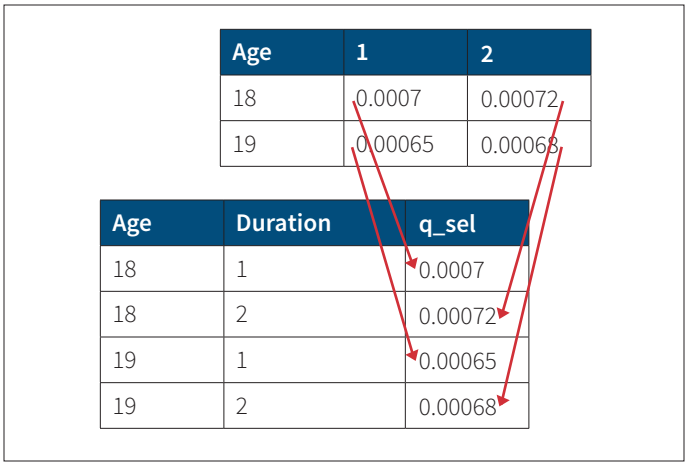
My view is that we should stack all the tables on top of each other and add new columns to serve as identifiers. In this way, we can include an arbitrary number of tables within a single rectangle of data that has columns to identify things like the mortality basis. This is one of the steps in making the dataset "tidy."

Copyright © 2020 Society of Actuaries. All rights reserved.

WHAT IS TIDY DATA?

Tidy data is a concept that was introduced in a paper by Hadley Wickham. Put simply, each column represents a variable and each row represents an observation. The mortality table Excel files are not tidy because the mortality variable belongs to many columns instead of a single column. Figure 3 demonstrates how one could reformat these tables into a tidy format. See that there is now a single column representing the mortality rate variable.

Figure 3
Tidying Process



There are functions to perform this operation in R, Python, SQL and Power Query.

WHY TIDY DATA?

A single table can be created from multiple files, and tidy data also allows for join operations.

> The tidy data format allows for join operations.

Fewer Tables

There are 28 separate Excel files for the 2015 VBT ALB Relative Risk tables. Figure 4 shows what the Excel files for the 2015 VBT ALB Relative Risk tables look like after being converted to a single data frame in a tidy format.

Instead of having a separate block of data for separate genders, we include an additional column that distinguishes the gender. Instead of adding a new table for each demographic split included in the table structure, we add a new column. This allows for a single table to be made from the original 28 separate blocks of data.

We don't have to put the data in a tidy format to condense the Excel files into a single dataset. We could add more columns to the dataset and stack the data without moving the durations into their own column. However, this would make it more difficult to do lookups. The tidy data format allows for join operations instead of forcing us into the INDEX-MATCH-MATCH Excel pattern.

Join Operations

Suppose we are building a database application that analyzes company mortality experience. Figure 5 shows what a section of that data might look like.

Figure 4
Tidy Select Mortality Format

gender	tobacco	relative_risk	issue_age	duration	q_sel
Female	Non-Smoker	RR50	18	1	0.00018
Female	Non-Smoker	RR50	18	2	0.00018
Female	Non-Smoker	RR50	18	3	0.00017
Female	Non-Smoker	RR50	18	4	0.00017
Female	Non-Smoker	RR50	18	5	0.00017
Female	Non-Smoker	RR50	18	6	0.00017
Female	Non-Smoker	RR50	18	7	0.00017
Female	Non-Smoker	RR50	18	8	0.00017
Female	Non-Smoker	RR50	18	9	0.00017
Female	Non-Smoker	RR50	18	10	0.00018
Female	Non-Smoker	RR50	18	11	0.00020

Copyright © 2020 Society of Actuaries. All rights reserved.

Figure 5
Example Experience Database Format

Gender	Duration	Issue Age	Exposures	Death
F	12	30	1	0
F	13	30	1	0
F	14	30	1	0
F	15	30	1	1

To do an A/E analysis, we must add a new column representing the expected number of deaths. This information would be filled in using mortality tables. An INDEX-MATCH-MATCH looking up the mortality rate for issue age 30 and duration 15 from the non-tidy Excel tables looks like this.

```
"=INDEX(table,MATCH(30,issue_age),
MATCH(15,duration))"
```

Because we have different tables for each demographic, the actual formula would involve some conditional logic that changes the mortality table on the basis of the demographic. If you have 28 tables to choose from, this becomes unworkable in Excel without using VBA.

Changing to a tidy data format will help, because each column specifies a demographic variable and we can create a unique lookup identifier for each combination. In Excel, we concatenate the demographic columns to create an identifier column in our tidy mortality table and in our mortality experience.

```
"gender&risk&tobacco&issue_age&duration"
```

These new columns can be used as keys in a VLOOKUP to assign the appropriate rates to our experience intervals. In R/Python/SQL, this sort of operation is common and is said to be a "join on multiple keys." Here is how we could attach the 2017 CSO Preferred Structure to our experience data using R.

```
left_join(experience, CSO2017_preferred_
structure, by = c(gender, risk, tobacco,
issue_age, duration))
```

DATA CONVERSION PROCESS

The approximately 200 Excel files associated with the 1983 GAM, 2001 CSO, 2015 VBT, 2017 CSO, AM92 and UP1984 tables have been converted into a tidy format. We used R and interacted with Excel using the package "readxl."

We placed the files for a mortality basis in a folder and iterated over them, allowing us to convert the 28 files associated with the 2015 VBT Relative Risk ALB tables much quicker than converting one at a time. Identification columns were created from the cell containing the description of the table for each Excel file. Separate tables are created for each Excel file, which are then stacked on top of each other to create a single table.

The scripts that perform the conversions are in the subfolders of data-raw in the repository for the code. The converted data can be downloaded in a CSV format from OneDrive.

WHAT'S NEXT?

The next article will be on creating a single data table from the select and ultimate tables for real-world use cases. This format is available on OneDrive as the "combined" format. We include a demonstration of Google BigQuery and use it to store an alternative format in which there is a single table that represents what was once 130 Excel files. ■

Matthew Caseres, ASA, is working on starting an actuarial nonprofit. He can be reached at *matthewcaseres@outlook.com* and on GitHub at *github.com/ActuarialAnalyst.*

Copyright © 2020 Society of Actuaries. All rights reserved.