SESUG Paper 245-2019

Should I Wear Pants? And Where Should I Travel in the Portuguese Expanse? Automating Business Rules and Decision Rules Through Reusable Decision Table Data Structures that Leverage SAS Arrays

Troy Martin Hughes
Louise Hadden, Abt Associates Inc.

ABSTRACT

Decision tables operationalize one or more contingencies and the respective actions that should be taken when contingencies are true. Decision tables capture conditional logic in dynamic control tables rather than hardcoded programs, facilitating maintenance and modification of the business rules and decision rules they contain—without the necessity to modify the underlying code (that interprets and operationalizes the decision tables). This text introduces a flexible, data-driven SAS® macro that ingests decision tables—maintained as comma-separated values (CSV) files—into SAS to dynamically write conditional logic statements that can subsequently be applied to SAS data sets. This metaprogramming technique relies on SAS temporary arrays that can accommodate limitless contingency groups and contingencies of any content. To illustrate the extreme adaptability and reusability of the software solution, several decision tables are demonstrated, including those that separately answer the questions *Should I wear pants* and *Where should I travel in the Portuguese expanse?* The DECISION_TABLE SAS macro is included and is adapted from the author's text: *SAS® Data-Driven Development: From Abstract Design to Dynamic Functionality*. [1]

INTRODUCTION

The International Organization for Standardization (ISO) defines a *decision table* as a "table of all contingencies that are to be considered in the description of a problem together with the action to be taken." [2] For example, a coworker may wash his hands after using the restroom *only* when the restroom has other occupants, but when he is alone, he chooses not to wash his hands because he's a dirty boy. In this example, the filthy coworker has only one decision point with two contingencies—the mutually exclusive presence or absence of coworkers. This single decision point (i.e., independent variable) drives the dichotomous outcome (i.e., dependent variable), the decision of whether to wash or not to wash one's hands. The ISO defines a *decision outcome* as the "result of a decision (which therefore determines the control flow alternative taken)." [2] The examples in this text demonstrate one or more mutually exclusive decision points that prescribe a single outcome (e.g., hand washing); however, more complex decision tables could yield multiple decision outcomes (e.g., hand washing and post-micturition hair combing).

Decision outcomes are driven by decision rules, which are the conditional logic statements that prescribe some predetermined outcome based on dynamic inputs. ISO defines a *decision rule* as a "combination of conditions (also known as causes) and actions (also known as effects) that produce a specific outcome in decision table testing and cause-effect graphing." [3] In a concrete design paradigm, decision rules are hardcoded, whereas data-driven software design extracts decision rules from code and maintains them instead within external data structures—control tables that can be modified by stakeholders (including nontechnical ones) when decision rules need to be altered.

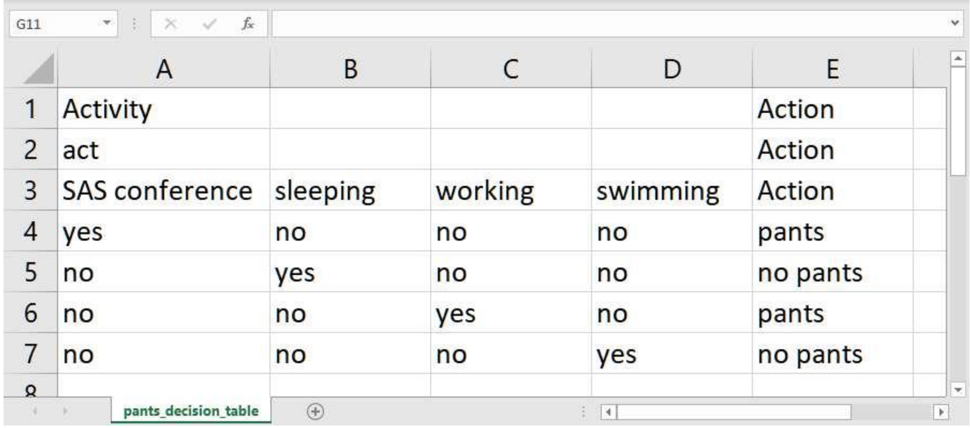
The DECISION_TABLE macro demonstrated in this text can flexibly accommodate decision tables of varying size (including both the number of decision points and the respective contingencies thereof) and content, and its reusability is highlighted by the diverse examples in this text. The flexibility of this solution is delivered through temporary SAS arrays (that invoke the _TEMPORARY_ option), which allow values to be assigned dynamically to the arrays that are created. Although decision tables may be insufficient for complex, algorithmic logic and statistical models, they are useful in simpler cases.

Decision tables are especially preferred in environments in which decision rules must be malleable and in which individual stakeholders might maintain disparate decision rules. Thus, rather than maintaining multiple versions of code (containing hardcoded conditional logic or other data models), stakeholders can maintain one program that is fed different decision tables to effect diverse, dynamic outcomes. This data-driven design approach also facilitates modification of decision tables by nontechnical personnel. For example, a business analyst with no expertise in SAS software development can maintain a decision table within an Excel spreadsheet and can alter and apply this data model whenever necessary—as well as best decide when and where to wear pants!

SHOULD I WEAR PANTS?

In some cases, yes. At the moment while writing this text, no. This simple example introduces the concepts of decision tables, decision rules, and decision outcomes by demonstrating some circumstances for which pants probably should be worn. It is not intended to demonstrate *all* circumstances for which pants should be worn.

The Pants decision table (pants_decision_table.xlsx) is demonstrated in Figure 1.



	A	B	C	D	E
1	Activity				Action
2	act				Action
3	SAS conference	sleeping	working	swimming	Action
4	yes	no	no	no	pants
5	no	yes	no	no	no pants
6	no	no	yes	no	pants
7	no	no	no	yes	no pants

Figure 1. Pants Decision Table

The spreadsheet should be exported to a comma-separated values (CSV) file (pants_decision.table.csv):

```
Activity,,,,Action
act,,,,Action
SAS conference,sleeping,working,swimming,Action
yes,no,no,no,pants
no,yes,no,no,no pants
no,no,yes,no,pants
no,no,no,yes,no pants
```

Decision tables can be represented myriad ways, with the spreadsheet in Figure 1 demonstrating only one method. However, as the decision table also represents a user-defined data structure, it is worthwhile to define some data rules for this data structure usage. This data structure template definition increases the ease with which decision table instances can be implemented in the future with unrelated content:

- The decision table is stored in a tabular format (e.g., Excel, CSV).
- Each row must have the same number of columns.
- Each column must have the same number of observations.
- The first row contains labels that can be optionally applied, with each label appearing in the column of its first option; the rightmost column must be ACTION.
- The second row contains variable names for each decision point, with each name appearing in the column of its first option; the rightmost column must be ACTION.
- The third row contains a list of contingencies (i.e., case-insensitive values) that correspond to the decision points/variables above; the rightmost column must be ACTION.
- The fourth and subsequent rows contain case-insensitive values of YES or NO, depicting whether the above option is active or inactive for a specific rule. As each decision point contains mutually

exclusive contingencies, only one YES contingency is allowed per decision points with the remaining contingencies indicating NO.

- Data types for all cells are character.
- All column widths have a maximum number of 32 characters (which can be increased in the ARRAY statements).
- A total number of 50 columns (across all variables and their respective options) can be processed (which can be increased in the ARRAY statements).

Note that the rules speak only to the data structure and its format; they explicitly do not allude to how the data are interpreted by SAS or the content (i.e., knowledge domain) of the data. This ensures that end users or subject matter experts (SMEs) who may be maintaining the decision table are focused on entering accurate data in the correct format, whereas developers are focused on understanding the structure so that they can implement an appropriate solution to interpret and operationalize the decision rules (i.e., control data) within the data structure.

The decision table in Figure 1 can be distilled to the following decision rules:

- If I am at a SAS conference, I should wear pants.
- If I am sleeping, I should not wear pants.
- If I am working, I should wear pants.
- If I am swimming, I should not wear pants.

As mentioned previously, more complex decision tables might include multiple outcomes that can be achieved based on underlying decision rules.

The Pants decision table is ingested into a SAS data set by invoking the DECISION_TABLE macro, included in the subsequent section. Note that a user-specified file location (&LOC) must be selected in which the decision_table.sas program file and all decision tables should be saved:

```
%let loc=d:\sas\;      * USER MUST CHANGE LOCATION *;
%include "&loc.decision_table.sas";
%decision_table(csv=&loc.pants_decision_table.csv);
```

When the Pants decision table is ingested, DECISION_TABLE dynamically creates the &DECISIONRULES global macro variable (in which indentation has been added to improve readability):

```
if upcase(act)="SAS CONFERENCE" and upcase(act)^="SLEEPING" and
   upcase(act)^="WORKING" and upcase(act)^="SWIMMING" then action="pants";
else if upcase(act)^="SAS CONFERENCE" and upcase(act)="SLEEPING" and
   upcase(act)^="WORKING" and upcase(act)^="SWIMMING" then action="no pants";
else if upcase(act)^="SAS CONFERENCE" and upcase(act)^="SLEEPING" and
   upcase(act)="WORKING" and upcase(act)^="SWIMMING" then action="pants";
else if upcase(act)^="SAS CONFERENCE" and upcase(act)^="SLEEPING"
   and upcase(act)^="WORKING" and upcase(act)="SWIMMING" then action="no pants";
```

This conditional logic can be subsequently applied to a data set by executing the &DECISIONRULES macro variable. For example, the Activities data set contains three observations—one that results in a “pants” action, one in a “no pants” action (ooh la la!), and one that results in an undefined action:

```
data activities;
   length act $32;
   label act='Activity';
   act='sleeping';
   output;
   act='working';
   output;
   act='laughing';
   output;
run;
```

When a DATA step invokes the &DECISION_RULES global macro variable, the Action variable is created based on the decision rules contained in the referenced decision table:

```
data activities_rules;
   set activities;
   length action $32;
   &decisionrules;
run;
```

Table 1 demonstrates the resultant Activities_rules data set that includes the Action variable.

Activity	**Action**
sleeping	no pants
working	pants
laughing	

Table 1. Activities_rules Data Set with Decision Rules Applied

Note that as “laughing” is not defined in the decision table, it produces no associated outcome.

DECISION_TABLE MACRO

The DECISION_TABLE macro should be saved in a user-specified folder as decision_table.sas:

```
%macro decision_table(csv= /* path and CSV file name */);
data _null_;
   infile "&csv" truncover end=eof;
   length line $10000 tot 8 i 8 rule $30000;
   format line $10000.;
   input line & $;
   array contlabel[50] $32 _temporary_;
   array contlist[50] $32 _temporary_;
   array vallist[50] $32 _temporary_;
   retain tot;
   retain rule '';
   * get contingency labels;
   if _n_=1 then do;
      do tot=1 to countw(line,',','m');
         contlabel[tot]=scan(line,tot,',','m');
         end;
      tot=tot-1;
      end;
   * get contingency groups/variables;
   if _n_=2 then do;
      do i=1 to tot;
         contlist[i]=scan(line,i,',','m');
         if missing(contlist[i]) then contlist[i]=contlist[i-1];
         end;
      end;
   * get contingency values;
   if _n_=3 then do;
      do i=1 to tot;
         vallist[i]=upcase(scan(line,i,',','m'));
         end;
      end;
   * get decision rules;
   if _n_>3 then do;
      do i=1 to tot;
         if i=tot then do;
            rule=strip(rule) || ' then action="' ||
               strip(scan(line,i,',')) || '";';
            end;
         else do;
            if i=1 then rule=strip(rule) || ifc(_n_=4,' if',' else if');
            else rule=strip(rule) || ' and ';
            rule=strip(rule) || ' upcase(' || strip(contlist[i]) || ')' ||
                  ifc(upcase(scan(line,i,','))='YES','=','^=') || '"' ||
                  strip(vallist[i]) || '"';
            end;
         end;
      end;
   if eof then call symputx('decisionrules',strip(rule),'g');
run;
%mend;
```

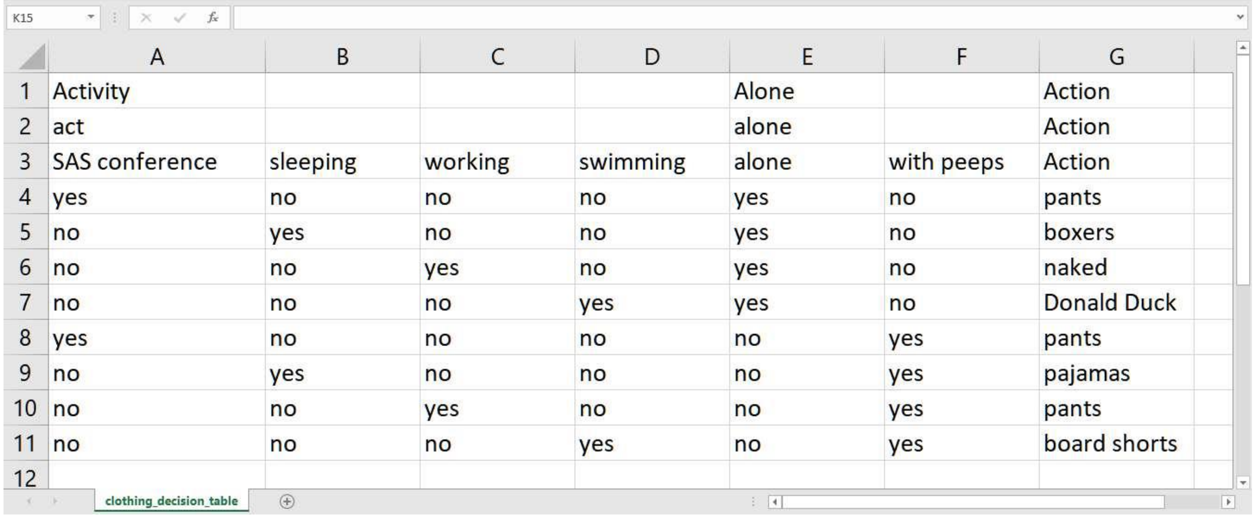
The DECISION_TABLE macro relies on temporary arrays (i.e., Contlabel, Contlist, Vallist) to hold the contingency group (i.e., label or category), variable, and values. Each array can hold 50 elements, although this number can be increased by modifying the ARRAY statements. The Pants decision table is simplistic in that it contains only one decision point (Activity), thus it also contains only one variable (Act); there is always a one-to-one correlation between the number of decision points and variables, as they must appear in the same columns within the decision table spreadsheet. The Pants decision table contains four contingencies, each of which is mutually exclusive. Thus, in this simplified data model, one cannot be both swimming and at a SAS conference (despite the significant swimming, hot tubbing, and skinny dipping that occurs at SAS conferences).

The Rule variable is incrementally built (and retained) as the DECISION_TABLE macro iterates across each observation within the decision table. When the end-of-file (EOF) marker is reached, the &DECISIONRULES global macro variable is initialized to the value of Rule, which can be used in subsequent DATA steps to execute dynamically generated conditional logic statements. The UPCASE functions throughout DECISION_TABLE ensure that all values within the decision table are case-insensitive when evaluated in these logic statements.

Decision table outcomes can be duplicated across observations, thus as Figure 1 demonstrates, two pathways lead to a “pants” outcome and two pathways lead to a “no pants” outcome. However, the arrangement of all contingency values must be unique for each observation. For example, Figure 1 demonstrates a YES-NO-NO-NO pattern that results in a “pants” outcome, thus no other row can contain the identical YES-NO-NO-NO pattern.

WHAT PANTS SHOULD I WEAR?

Adding a second decision point to the decision table effectively adds a second independent variable that can be altered to add variability to outcomes. For example, the decision to wear or not to wear pants is likely determined not only by one’s activity but also by one’s company—those are, the observers of the pants or pantsless activities. Moreover, greater granularity in resultant actions might be achieved by expanding possible outcomes beyond the pants-no pants dichotomy. Figure 2 refactors the Pants decision table into the more generic Clothing decision table (clothing_decision_table.xlsx).



	A	B	C	D	E	F	G
1	Activity				Alone		Action
2	act				alone		Action
3	SAS conference	sleeping	working	swimming	alone	with peeps	Action
4	yes	no	no	no	yes	no	pants
5	no	yes	no	no	yes	no	boxers
6	no	no	yes	no	yes	no	naked
7	no	no	no	yes	yes	no	Donald Duck
8	yes	no	no	no	no	yes	pants
9	no	yes	no	no	no	yes	pajamas
10	no	no	yes	no	no	yes	pants
11	no	no	no	yes	no	yes	board shorts
12							

Figure 2. Updated Clothing Decision Table

The Clothing decision table now has two contingency groups representing the intersection of the independent variables *What activity am I performing* and *Am I alone or not*? Thus, it is now possible to be alone or with peeps at a SAS conference, although "pants" are recommended in both cases—you know, because it's a conference. However, the Alone variable is more discriminating for other activities. For example, pants are recommended when working with others but only a shirt (aka *Donald Ducking* it) might be recommended when working alone (presumably because a Skype session only exposes the top half of your business to coworkers). Just don't stand up in the middle of that remote meeting to go grab a cup of coffee or a snack!

The Clothing spreadsheet should be exported to a CSV file (clothing_decision_table.csv):

```
Activity,,,,Alone,,Action
act,,,,alone,,Action
SAS conference,sleeping,working,swimming,alone,with peeps,Action
yes,no,no,no,yes,no,pants
no,yes,no,no,yes,no,boxers
no,no,yes,no,yes,no,naked
no,no,no,yes,yes,no,Donald Duck
yes,no,no,no,no,yes,pants
no,yes,no,no,no,yes,pajamas
no,no,yes,no,no,yes,pants
no,no,no,yes,no,yes,board shorts
```

With no modification to the DECISION_TABLE macro (and thus promoting code stability and integrity, hallmarks of software quality), the following code can be run to build the &DECISIONRULES global macro variable:

```
%let loc=d:\sas\;     * USER MUST CHANGE LOCATION *;
%include "&loc.decision_table.sas";
%decision_table(csv=&loc.clothing_decision_table.csv);
```

The &DECISIONRULES macro variable is initialized to the following code (now displayed without indentation or carriage returns):

```
if upcase(act)="SAS CONFERENCE" and upcase(act)^="SLEEPING" and
upcase(act)^="WORKING" and upcase(act)^="SWIMMING" and upcase(alone)="ALONE" and
upcase(alone)^="WITH PEEPS" then action="pants"; else if upcase(act)^="SAS
CONFERENCE" and upcase(act)="SLEEPING" and upcase(act)^="WORKING" and
upcase(act)^="SWIMMING" and upcase(alone)="ALONE" and upcase(alone)^="WITH PEEPS"
then action="boxers"; else if upcase(act)^="SAS CONFERENCE" and
upcase(act)^="SLEEPING" and upcase(act)="WORKING" and upcase(act)^="SWIMMING" and
upcase(alone)="ALONE" and upcase(alone)^="WITH PEEPS" then action="naked"; else if
upcase(act)^="SAS CONFERENCE" and upcase(act)^="SLEEPING" and
upcase(act)^="WORKING" and upcase(act)="SWIMMING" and upcase(alone)="ALONE" and
upcase(alone)^="WITH PEEPS" then action="Donald Duck"; else if upcase(act)="SAS
CONFERENCE" and upcase(act)^="SLEEPING" and upcase(act)^="WORKING" and
upcase(act)^="SWIMMING" and upcase(alone)^="ALONE" and upcase(alone)="WITH PEEPS"
then action="pants"; else if upcase(act)^="SAS CONFERENCE" and
upcase(act)="SLEEPING" and upcase(act)^="WORKING" and upcase(act)^="SWIMMING" and
upcase(alone)^="ALONE" and upcase(alone)="WITH PEEPS" then action="pajamas"; else
if upcase(act)^="SAS CONFERENCE" and upcase(act)^="SLEEPING" and
upcase(act)="WORKING" and upcase(act)^="SWIMMING" and upcase(alone)^="ALONE" and
upcase(alone)="WITH PEEPS" then action="pants"; else if upcase(act)^="SAS
CONFERENCE" and upcase(act)^="SLEEPING" and upcase(act)^="WORKING" and
upcase(act)="SWIMMING" and upcase(alone)^="ALONE" and upcase(alone)="WITH PEEPS"
then action="board shorts";
```

As &DECISIONRULES now relies on both the Act and Alone variables, these two variables would need to be found in the data set to which these rules would be applied (not demonstrated). Note that with a relatively simple decision table, the &DECISIONRULES macro variable is already 1,602 characters in length:

```
%put %length(&decisionrules);
```

Were the decision table complexity to continue to increase, including the number of decision points and/or contingency values, the length of &DECISIONRULES would continue to increase—and at some point could surpass the 30,000 character limit established in the LENGTH statement within the DECISION_TABLE macro. A more robust version of the macro (not demonstrated) could evaluate macro variable length to ensure it was less than 30,000 characters, and perform exception handling in the event that the length exceeded this threshold.

The genius and power of DECISION_TABLE is its ability to flex to incorporate different or additional contingency groups and values without any modification to the underlying code. This data-driven design ensures that unrelated decision rules can be operationalized by leveraging the identical decision table data structure. By utilizing this template (and the data rules prescribed on Page 2), SAS practitioners can reuse not only the DECISION_TABLE macro but also the underlying data structure on which it relies. This reusability—of both the macro and the decision table data structure—is demonstrated in the following section in which unrelated decisions rules are applied to unrelated data.

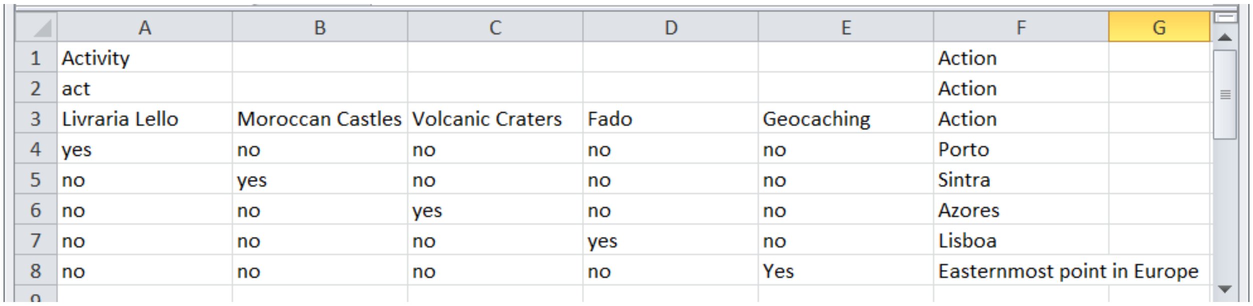
WHERE SHOULD I TRAVEL IN THE PORTUGUESE EXPANSE?

Portugal is a sublime vacation destination, with sights and experiences for every type of traveler. Making a decision as to where to spend your time is a pleasantly difficult endeavor. Enter the decision table! Decision tables allow one to identify and rank the delicious, perhaps licentious choices involved, and formalize the choices involved in booking travel. The decision of where to travel in the Portuguese expanse provided the perfect opportunity to test DECISION_TABLE's vaunted ability to incorporate new contingency groups and values.

The first task is to define desired activities – in the case of vacation planning, what do you want to do? Is there a particular museum, restaurant, or scenic view that you want to take in? Record those choices in your planning spreadsheet.

IN THE BEGINNING

For my initial attempt, I limited my subordinate actions to Livraria Lello (the divine bookstore that J.K. Rowling spent time in while writing Harry Potter and the Sorcerer's Stone), Morroccan Castles (a la PARQUE E PALÁCIO NACIONAL DA PENA), volcanic craters, Fado (a cross between folk music and opera), and geocaching. As you might guess, each of these actions is most easily accessed in specific destinations in Portugal.



	A	B	C	D	E	F	G
1	Activity					Action	
2	act					Action	
3	Livraria Lello	Moroccan Castles	Volcanic Craters	Fado	Geocaching	Action	
4	yes	no	no	no	no	Porto	
5	no	yes	no	no	no	Sintra	
6	no	no	yes	no	no	Azores	
7	no	no	no	yes	no	Lisboa	
8	no	no	no	no	Yes	Easternmost point in Europe	

Figure 3. Initial Portuguese Expanse Decision Table

The PortugueseExpants spreadsheet should be exported to a CSV file (portugueseexpants0.csv):

```
Activity,,,,,Action
act ,,,,,Action
Livraria Lello,Moroccan Castles,Volcanic Craters,Fado,Geocaching,Action
yes,no,no,no,no,Porto
no,yes,no,no,no,Sintra
```

```
no,no,yes,no,no,Azores
no,no,no,yes,no,Lisboa
no,no,no,no,Yes,Easternmost point in Europe
```

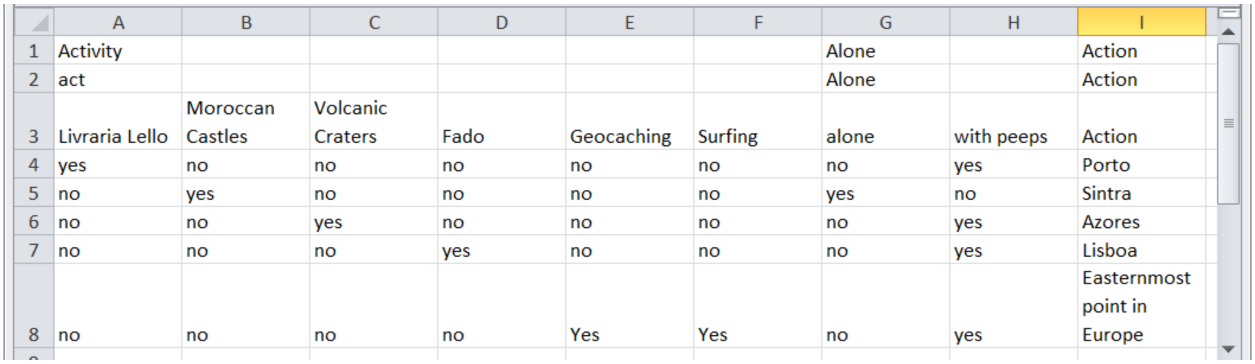
The entire decision-making process takes the CSV file above into memory and constructs a giant macro variable to contain the entire decision-making process. A data set containing each of the subordinate actions in rows is constructed, the macro is invoked, and decisions are magically made. At the last minute I added surfing to the activities_rules0 data set shown in Table 2, but since it was not in the list of subordinate actions, no decision is returned, and indeed, since surfing can be done at two of the decisions / destinations, it was out of scope for the structure of the macro.

Action	**Decision**
Livraria Lello	Porto
Moroccan Castles	Sintra
Volcanic Craters	Azores
Fado	Lisboa
Geocaching	Easternmost point in Europe
Surfing	

Table 2. Activities_rules0 Data Set with Decision Rules Applied

SECONDS

Vacation plans are rarely made in a vacuum, just as the decision to wear pants (or not). An important aspect of a choice of destination is whether or not you will be accompanied (and whether or not you'll be wearing pants at the destination). The next step was to add a second decision point to both the input data and the decision file construction process. Surfing sounded pretty good to me, so I added that as a desired action, and added a second decision point, alone or with peeps.



	A	B	C	D	E	F	G	H	I
1	Activity						Alone		Action
2	act						Alone		Action
3	Livraria Lello	Moroccan Castles	Volcanic Craters	Fado	Geocaching	Surfing	alone	with peeps	Action
4	yes	no	no	no	no	no	no	yes	Porto
5	no	yes	no	no	no	no	yes	no	Sintra
6	no	no	yes	no	no	no	no	yes	Azores
7	no	no	no	yes	no	no	no	yes	Lisboa
8	no	no	no	no	Yes	Yes	no	yes	Easternmost point in Europe

Figure 4. Second Portuguese Expanse Decision Table

The PortugueseExpants1 spreadsheet should be exported to a CSV file (portugueseexpants1.csv):

```
Activity,,,,,,Alone,,Action
act ,,,,,,Alone,,Action
Livraria Lello,Moroccan Castles,Volcanic Craters,
Fado,Geocaching,Surfing,alone,with peeps,Action
yes,no,no,no,no,no,no,yes,Porto
no,yes,no,no,no,no,yes,no,Sintra
no,no,yes,no,no,no,no,yes,Azores
no,no,no,yes,no,no,no,yes,Lisboa
no,no,no,no,Yes,Yes,no,yes,Easternmost point in Europe
```

The macro variable resolves to this:

```
SYMBOLGEN:  Macro variable DECISIONRULES resolves to if upcase(act)="LIVRARIA
 LELLO" and upcase(act)^="MOROCCAN CASTLES" and
 upcase(act)^="VOLCANIC CRATERS" and upcase(act)^="FADO" and
 upcase(act)^="GEOCACHING" and upcase(act)^="SURFING" and
 upcase(Alone)^="ALONE" and upcase(Alone)="WITH PEEPS" then
 action="Porto"; else if upcase(act)^="LIVRARIA LELLO" and
 upcase(act)="MOROCCAN CASTLES" and upcase(act)^="VOLCANIC CRATERS"
 and upcase(act)^="FADO" and upcase(act)^="GEOCACHING" and
 upcase(act)^="SURFING" and upcase(Alone)="ALONE" and
 upcase(Alone)^="WITH PEEPS" then action="Sintra"; else if
 upcase(act)^="LIVRARIA LELLO" and upcase(act)^="MOROCCAN CASTLES"
 and upcase(act)="VOLCANIC CRATERS" and upcase(act)^="FADO" and
 upcase(act)^="GEOCACHING" and upcase(act)^="SURFING" and
 upcase(Alone)^="ALONE" and upcase(Alone)="WITH PEEPS" then
 action="Azores"; else if upcase(act)^="LIVRARIA LELLO" and
 upcase(act)^="MOROCCAN CASTLES" and upcase(act)^="VOLCANIC
 CRATERS" and upcase(act)="FADO" and upcase(act)^="GEOCACHING" and
 upcase(act)^="SURFING" and upcase(Alone)^="ALONE" and
 upcase(Alone)="WITH PEEPS" then action="Lisboa"; else if
 upcase(act)^="LIVRARIA LELLO" and upcase(act)^="MOROCCAN CASTLES"
 and upcase(act)^="VOLCANIC CRATERS" and upcase(act)^="FADO" and
 upcase(act)="GEOCACHING" and upcase(act)="SURFING" and
 upcase(Alone)^="ALONE" and upcase(Alone)="WITH PEEPS" then
 action="Easternmost point in Europe";
```

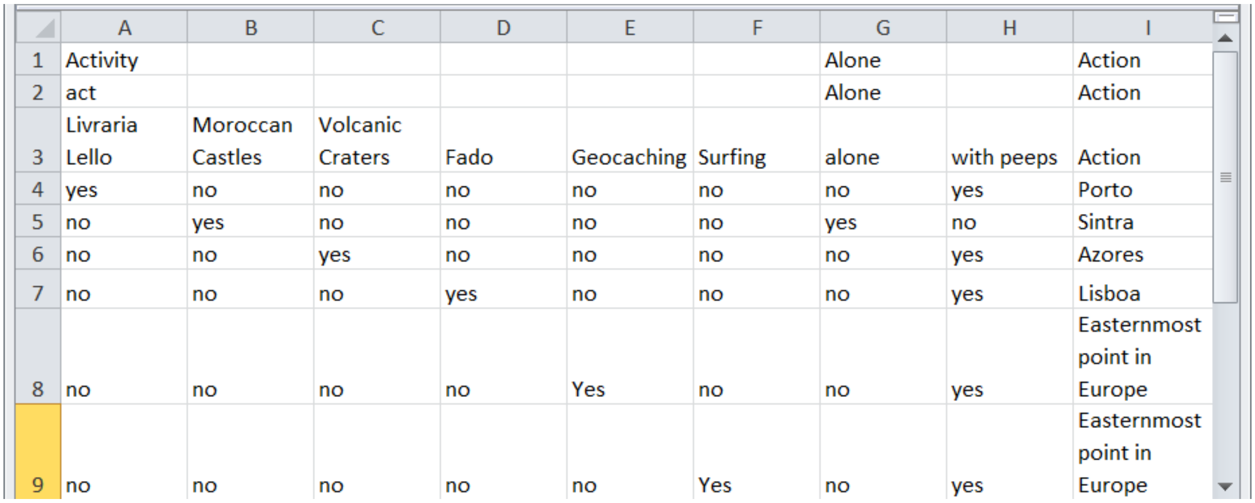
Activity	**Alone or With Peeps**	**Decision**
Livraria Lello	With Peeps	Porto
Moroccan Castles	Alone	Sintra
Volcanic Craters	With Peeps	Azores
Fado	With Peeps	Lisboa
Geocaching	With Peeps	☹
Surfing	Alone	☹

Table 3. Activities_rules1 Data Set with Decision Rules Applied

I didn't follow the mutually exclusive rules of the DECISION_TREE macro. This time, surfing still doesn't get a decision AND geocaching doesn't get a decision, either – so going to the easternmost point in Europe definitely loses out, which is a shame. The macro gets that you can't do two activities at the same time in the same place with peeps and alone – until we tell it so.

THIRDS

For the third round, I corrected the user input error and added a decision line for both activities in the easternmost point in Europe with similar characteristics. The single change allows all decisions to resolve, and is depicted in Figure 5.



	A	B	C	D	E	F	G	H	I
1	Activity						Alone		Action
2	act						Alone		Action
3	Livraria Lello	Moroccan Castles	Volcanic Craters	Fado	Geocaching	Surfing	alone	with peeps	Action
4	yes	no	no	no	no	no	no	yes	Porto
5	no	yes	no	no	no	no	yes	no	Sintra
6	no	no	yes	no	no	no	no	yes	Azores
7	no	no	no	yes	no	no	no	yes	Lisboa
8	no	no	no	no	Yes	no	no	yes	Easternmost point in Europe
9	no	no	no	no	no	Yes	no	yes	Easternmost point in Europe

Figure 5. Third Portuguese Expanse Decision Table

The PortugueseExpants2 spreadsheet should be exported to a CSV file (portugueseexpants2.csv):

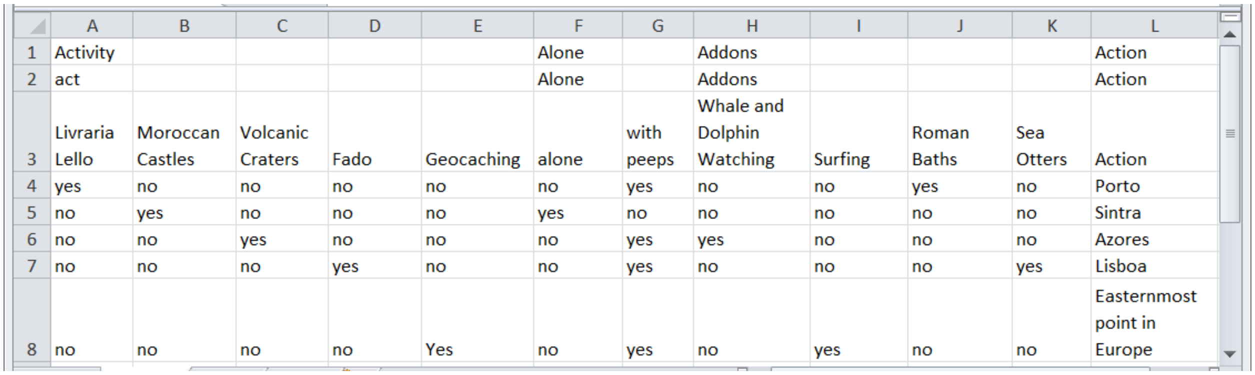
```
Activity,,,,,,Alone,,Action
act ,,,,,,Alone,,Action
Livraria Lello,Moroccan Castles,Volcanic Craters,
Fado,Geocaching,Surfing,alone,with peeps,Action
yes,no,no,no,no,no,no,yes,Porto
no,yes,no,no,no,no,yes,no,Sintra
no,no,yes,no,no,no,no,yes,Azores
no,no,no,yes,no,no,no,yes,Lisboa
no,no,no,no,Yes,no,no,yes,Easternmost point in Europe
no,no,no,no,no,Yes,no,yes,Easternmost point in Europe
```

Choices have been made for all of my activities, alone or with peeps. The fact that I can accomplish TWO goals in the same destination is pretty tempting.

Activity	**Alone or With Peeps**	**Decision**
Livraria Lello	With Peeps	Porto
Moroccan Castles	Alone	Sintra
Volcanic Craters	With Peeps	Azores
Fado	With Peeps	Lisboa
Geocaching	With Peeps	Easternmost point in Europe
Surfing	With Peeps	Easternmost point in Europe

Table 4. Activities_rules2 Data Set with Decision Rules Applied

For the fourth and final round, I decided I really couldn't live without more than one activity in most destinations, so I added a third decision point to the program. The change follows the rule of mutual exclusivity for both activities in the easternmost point in Europe with similar characteristics. The single change allows all decisions to resolve, with the original five lines for the five destinations.



	A	B	C	D	E	F	G	H	I	J	K	L
1	Activity					Alone		Addons				Action
2	act					Alone		Addons				Action
3	Livraria Lello	Moroccan Castles	Volcanic Craters	Fado	Geocaching	alone	with peeps	Whale and Dolphin Watching	Surfing	Roman Baths	Sea Otters	Action
4	yes	no	no	no	no	no	yes	no	no	yes	no	Porto
5	no	yes	no	no	no	yes	no	no	no	no	no	Sintra
6	no	no	yes	no	no	no	yes	yes	no	no	no	Azores
7	no	no	no	yes	no	no	yes	no	no	no	yes	Lisboa
8	no	no	no	no	Yes	no	yes	no	yes	no	no	Easternmost point in Europe

Figure 6. Fourth Portuguese Expanse Decision Table

The PortugueseExpants4 spreadsheet should be exported to a CSV file (portugueseexpants4.csv):

```
Activity,,,,,Alone,,Addons,,,,Action
act ,,,,,Alone,,Addons,,,,Action
Livraria Lello,Moroccan Castles,Volcanic Craters,Fado,Geocaching,alone,with
peeps,Whale and Dolphin Watching,Surfing,Roman Baths,Sea Otters,Action
yes,no,no,no,no,no,yes,no,no,yes,no,Porto
no,yes,no,no,no,yes,no,no,no,no,no,Sintra
no,no,yes,no,no,no,yes,yes,no,no,no,Azores
no,no,no,yes,no,no,yes,no,no,no,yes,Lisboa
no,no,no,no,Yes,no,yes,no,yes,no,no,Easternmost point in Europe
```

The “Addons” decision point allows more options per choice or destination, as long the rule of mutual exclusivity is followed: for example, you can’t be both alone and with peeps in the same destination. The code to run the modified decision tree now includes an additional decision point (addons), but no modifications (to the underlying code) were required to run DECISION_TREE. The perfect combinations of geocaching and surfing; Livraria Lello and roman baths; Fado and sea otters, and volcanic craters and whale and dolphin watching have been achieved.

```
%let loc=g:\pants\;     * USER MUST CHANGE LOCATION *;
%include "&loc.decision_table.sas";
%decision_table(csv=&loc.PortugueseExpants4.csv);

data activities4;
   length act alone addons $32;
   label act='Activity' alone='Alone or With Peeps' addons='Additional Activities';
   act='Livraria Lello'; alone='With Peeps'; addons='Roman Baths';
   output;
   act='Moroccan Castles'; alone='Alone'; addons='';
   output;
   act='Volcanic Craters'; alone='With Peeps'; addons='Whale and Dolphin Watching';
   output;
   act='Fado'; alone='With Peeps'; addons='Sea Otters';
   output;
   act='Geocaching'; alone='With Peeps'; addons='Surfing';
   output;
run;

data activities_rules4;
   length act alone action $32;
   set activities4;
   label action='Decision';
   &decisionrules;
run;
```

The macro variable has increased in length due to the additional decision points, and resolves to this:

```
SYMBOLGEN:  Macro variable DECISIONRULES resolves to:
```

```
if upcase(act)="LIVRARIA LELLO" and upcase(act)^="MOROCCAN CASTLES" and
upcase(act)^="VOLCANIC CRATERS" and upcase(act)^="FADO" and
upcase(act)^="GEOCACHING" and upcase(Alone)^="ALONE" and
upcase(Alone)="WITH PEEPS" and upcase(Addons)^="WHALE AND DOLPHIN
WATCHING" and upcase(Addons)^="SURFING" and upcase(Addons)="ROMAN
BATHS" and upcase(Addons)^="SEA OTTERS" then action="Porto"; else
if upcase(act)^="LIVRARIA LELLO" and upcase(act)="MOROCCAN
CASTLES" and upcase(act)^="VOLCANIC CRATERS" and
upcase(act)^="FADO" and upcase(act)^="GEOCACHING" and
upcase(Alone)="ALONE" and upcase(Alone)^="WITH PEEPS" and
upcase(Addons)^="WHALE AND DOLPHIN WATCHING" and
upcase(Addons)^="SURFING" and upcase(Addons)^="ROMAN BATHS" and
upcase(Addons)^="SEA OTTERS" then action="Sintra"; else if
upcase(act)^="LIVRARIA LELLO" and upcase(act)^="MOROCCAN CASTLES"
and upcase(act)="VOLCANIC CRATERS" and upcase(act)^="FADO" and
upcase(act)^="GEOCACHING" and upcase(Alone)^="ALONE" and
upcase(Alone)="WITH PEEPS" and upcase(Addons)="WHALE AND DOLPHIN
WATCHING" and upcase(Addons)^="SURFING" and upcase(Addons)^="ROMAN
BATHS" and upcase(Addons)^="SEA OTTERS" then action="Azores"; else
if upcase(act)^="LIVRARIA LELLO" and upcase(act)^="MOROCCAN
CASTLES" and upcase(act)^="VOLCANIC CRATERS" and
upcase(act)="FADO" and upcase(act)^="GEOCACHING" and
upcase(Alone)^="ALONE" and upcase(Alone)="WITH PEEPS" and
upcase(Addons)^="WHALE AND DOLPHIN WATCHING" and
upcase(Addons)^="SURFING" and upcase(Addons)^="ROMAN BATHS" and
upcase(Addons)="SEA OTTERS" then action="Lisboa"; else if
upcase(act)^="LIVRARIA LELLO" and upcase(act)^="MOROCCAN CASTLES"
and upcase(act)^="VOLCANIC CRATERS" and upcase(act)^="FADO" and
upcase(act)="GEOCACHING" and upcase(Alone)^="ALONE" and
upcase(Alone)="WITH PEEPS" and upcase(Addons)^="WHALE AND DOLPHIN
WATCHING" and upcase(Addons)="SURFING" and upcase(Addons)^="ROMAN
BATHS" and upcase(Addons)^="SEA OTTERS" then action="Easternmost
point in Europe";
```

Activity	**Alone or With Peeps**	**Decision**	**Additional Activities**
Livraria Lello	With Peeps	Porto	Roman Baths
Moroccan Castles	Alone	Sintra	
Volcanic Craters	With Peeps	Azores	Whale and Dolphin Watching
Fado	With Peeps	Lisboa	Sea Otters
Geocaching	With Peeps	Easternmost point in Europe	Surfing

Table 5. Activities_rules4 Data Set with Decision Rules Applied

CONCLUSION

Whether deciding to wear pants or where to travel in the Portuguese expanse, decision tables provide a method to capture decision rules or business rules, including their dynamic outcomes or actions. Although conditional logic and other hardcoded methods exist to achieve identical functionality, decision tables represent a malleable, data-driven design solution that facilitates the reusability of not only decision table data structures but also the underlying code that interprets these control data. This text introduced

DECISION_TABLE, a flexible SAS macro that can interpret and operationalize decision tables of varying size and content.

REFERENCES

[1] T. M. Hughes, SAS(r) Data-Driven Development: From Abstract Design to Dynamic Functionality, San Diego, California: CreateSpace, 2019.

[2] Information Processing--Specification of single-hit decision tables, vol. ISO 5806, Geneva, Switzerland: International Organization for Standardization (ISO), 1984.

[3] Software and systems engineering--Software testing--Part 4: Test techniques., Vols. ISO/IEC/IEEE 29119-4, Geneva: International Organization for Standardization and International Electrotechnical Commission, 2015.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Troy Martin Hughes
E-mail: troymartinhughes@gmail.com

Name: Louise S. Hadden
E-mail: louise_hadden@abtasssoc.com