IPUMS
Working Papers

Does 1 + 2 = 8?
Automating QA/QC for Tabular Data

Tracy Kugler†
University of Minnesota

Tsu Zhu
University of Minnesota

Finn Roberts
Max Plank Institute

CJ Adams
Australian National University

December 2025

Working Paper No. 2025-04
DOI: https://doi.org/10.18128/IPUMS2025-04

†Address correspondence to Tracy Kugler: takugler@umn.edu. This research was supported by grants from the National Science Foundation (BCS 2121891) and the National Institute for Child Health and Human Development (P2C HD041023).

Does 1 + 2 = 8? Automating QA/QC for Tabular Data

Tracy Kugler[1], Tsu Zhu[2], Finn Roberts[3], CJ Adams[4]

Abstract

The IPUMS International Historical Geographic Information System (IHGIS) assembles published data tables from population and agricultural censuses around the world and makes them available in a consistent, analysis-ready format. Source materials for IHGIS include print documents, which must be scanned and digitized using optical character recognition (OCR). Unlike characters in words, which can be checked against a dictionary, OCR software has no internal mechanism for validating recognition of numerical digits. We have developed an R package that fills this gap by checking the internal consistency of values within data tables. Using structured metadata describing the contents of rows and columns, the tools identify logical relationships between counts and totals. For example, counts for the sub-units of a geographic unit should sum to the count for their parent unit, and urban plus rural should equal the total. The tools then identify violations of the expected relationships and, by leveraging multiple relationships in which each cell participates, can often pinpoint cells where digits have been incorrectly recognized, infer the correct values, and make automated corrections. In cases where the tools cannot automatically pinpoint and correct errors, they produce output visually highlighting potentially incorrect cells to facilitate manual review. The best results are achieved for tables with at least one relationship in both rows and columns, where the software identifies and corrects 95% of error cells on average. These tools enable IHGIS to efficiently produce high-quality digital data for use in a wide variety of population-related research.

Keywords: OCR, data quality, census, tabular data, automation

[1] Dr. Tracy Kugler is a Principal Research Scientist and Product Manager of IHGIS at the Institute for Social Research and Data Innovation (ISRDI) at the University of Minnesota. She can be reached at takugler@umn.edu.
[2] Tsu Zhu is a Data Analyst for IHGIS and co-developer of the R Package detailed in this article. She can be reached at zhu00996@umn.edu.
[3] Finn Roberts is a former Senior Data Analyst for IHGIS and co-developer of the R Package. He is currently a Technical Assistant for MoveApps at the Max Plank Institute for Animal Behavior in Konstanz Germany and can be reached at froberts@ag.mpg.de.
[4] CJ Adams is an alumnus undergraduate research assistant for IHGIS and contributed to the initial version of the R Package. He recently received a Master of Diplomacy from the Australian National University, College of Asia and the Pacific and can be reached at C.jacob.adams@gmail.com.

Introduction

The IPUMS International Historical Geographic Information System (IHGIS) project has developed an R package to check the internal consistency of values in data tables, such as tables published in census results documents. IHGIS provides standardized, analysis-ready data sourced from diverse population and housing and agricultural census documents published by national statistical offices. For documents only available in print, optical character recognition (OCR) software is utilized to convert the tables to machine-readable, editable data, but this process may misidentify some characters in the input document, resulting in incorrect data values in the output file. Unlike characters in words that can be checked against a dictionary, OCR software has no internal mechanism for validating recognition of numerical digits. The R package fills this gap by identifying and correcting cell-level errors introduced during OCR processing. It uses structured metadata in tables to identify sum and total relationships, which are often present in both the rows and the columns. These relationships are then leveraged to identify inconsistencies between calculated sums across cells and the values given in a corresponding total row or column. Intersecting relationships across the rows and columns help pinpoint the locations of cells with incorrect values and enable the tools to infer and automatically update cells with corrected values. Simulation runs with tables containing known errors have shown that in a fairly typical case where tables have at least one relationship for both rows and columns and 1% of the cells contain errors, the package can automatically identify and correct nearly 95% of errors.

The flow of this paper follows the workflow for processing IHGIS data. Part 1 describes the preliminary steps needed to transform source documents into the standard CSV structure taken as input by the R package. These steps include using OCR software to digitize scanned documents and applying a custom markup framework and conversion software to construct the standard CSV files. In Part 2 we highlight salient features of the standard CSV file structure, including consistent row and column structures, an expanded geographic hierarchy, and meaningful blanks. In Part 3 we describe the automated QA/QC process in R and the algorithms used to identify and validate relationships, pinpoint errors, and update cells to their inferred corrected value. Part 3 also discusses scenarios where errors cannot be automatically corrected. We conclude with detailed performance metrics in Part 4.

1. Preliminary Stages

The QA/QC tools are embedded in a more extensive IHGIS data processing pipeline. Preliminary steps to convert scans of print documents into the standard CSV structure used as input for the QA/QC R package include conducting OCR and using a custom markup framework to enable transformation into the standard CSV structure.

1.1. OCR

IHGIS data tables sourced from print documents are first digitized using ABBYY FineReader[5] OCR software to convert scanned image PDFs to Excel worksheets. ABBYY has an interactive user interface that allows adjustment of row and column separators to ensure that table structure is maintained (Figure 1).

Figure 1. ABBYY FineReader interface

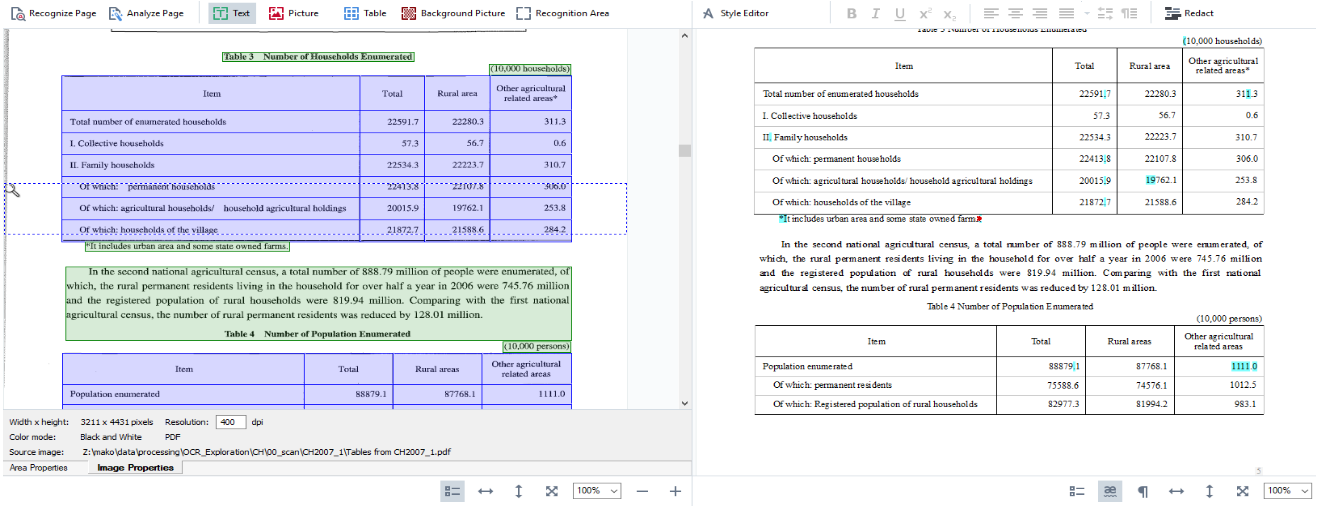


Table 3 Number of Households Enumerated

(10,000 households)

Item	Total	Rural area	Other agricultural related areas*
Total number of enumerated households	22591.7	22280.3	311.3
I. Collective households	57.3	56.7	0.6
II. Family households	22534.3	22223.7	310.7
Of which: permanent households	22413.8	22107.8	306.0
Of which: agricultural households/ household agricultural holdings	20015.9	19762.1	253.8
Of which: households of the village	21872.7	21588.6	284.2

*It includes urban area and some state owned farms.

In the second national agricultural census, a total number of 888.79 million of people were enumerated, of which, the rural permanent residents living in the household for over half a year in 2006 were 745.76 million and the registered population of rural households were 819.94 million. Comparing with the first national agricultural census, the number of rural permanent residents was reduced by 128.01 million.

Table 4 Number of Population Enumerated

(10,000 persons)

Item	Total	Rural areas	Other agricultural related areas
Population enumerated	88879.1	87768.1	1111.0

Width x height: 3211 x 4431 pixels Resolution: 400 dpi
Color mode: Black and White PDF
Source image: Z:\mako\data\processing\OCR_Exploration\CH\00_scan\CH2007_1\Tables from CH2007_1.pdf
Area Properties Image Properties

(10,000 households)

Item	Total	Rural area	Other agricultural related areas*
Total number of enumerated households	22591.7	22280.3	311.3
I. Collective households	57.3	56.7	0.6
II. Family households	22534.3	22223.7	310.7
Of which: permanent households	22413.8	22107.8	306.0
Of which: agricultural households/ household agricultural holdings	20015.9	19762.1	253.8
Of which: households of the village	21872.7	21588.6	284.2

*It includes urban area and some state owned farm.

In the second national agricultural census, a total number of 888.79 million of people were enumerated, of which, the rural permanent residents living in the household for over half a year in 2006 were 745.76 million and the registered population of rural households were 819.94 million. Comparing with the first national agricultural census, the number of rural permanent residents was reduced by 128.01 million.

Table 4 Number of Population Enumerated

(10,000 persons)

Item	Total	Rural areas	Other agricultural related areas
Population enumerated	88879.1	87768.1	1111.0
Of which: permanent residents	75588.6	74576.1	1012.5
Of which: Registered population of rural households	82977.3	81994.2	983.1

OCR for text uses dictionaries to improve the recognition of uncertain characters. For example, if the software were uncertain about the second character in the word “number”/”namber,” it could determine that it should be a ‘u’ because “number” is in the dictionary while “namber” is not. For numeric values, there is no universal dictionary that can be used to check for recognition of incorrect strings of digits. This gap in OCR technology is particularly problematic for historical print documents that often have poor image quality. Conditions like physical damage, very fine print, smudges, and other issues make it difficult for OCR technology to properly recognize numbers automatically. Digits may also be mistaken for

[5] https://pdf.abbyy.com/

letters or punctuation. For example, a 1 may be recognized as an ‘i’ or ‘!’, further complicating matters. Manual review of OCR output is infeasible at scale and would inevitably miss errors, necessitating automated quality checks.

1.2. Markup

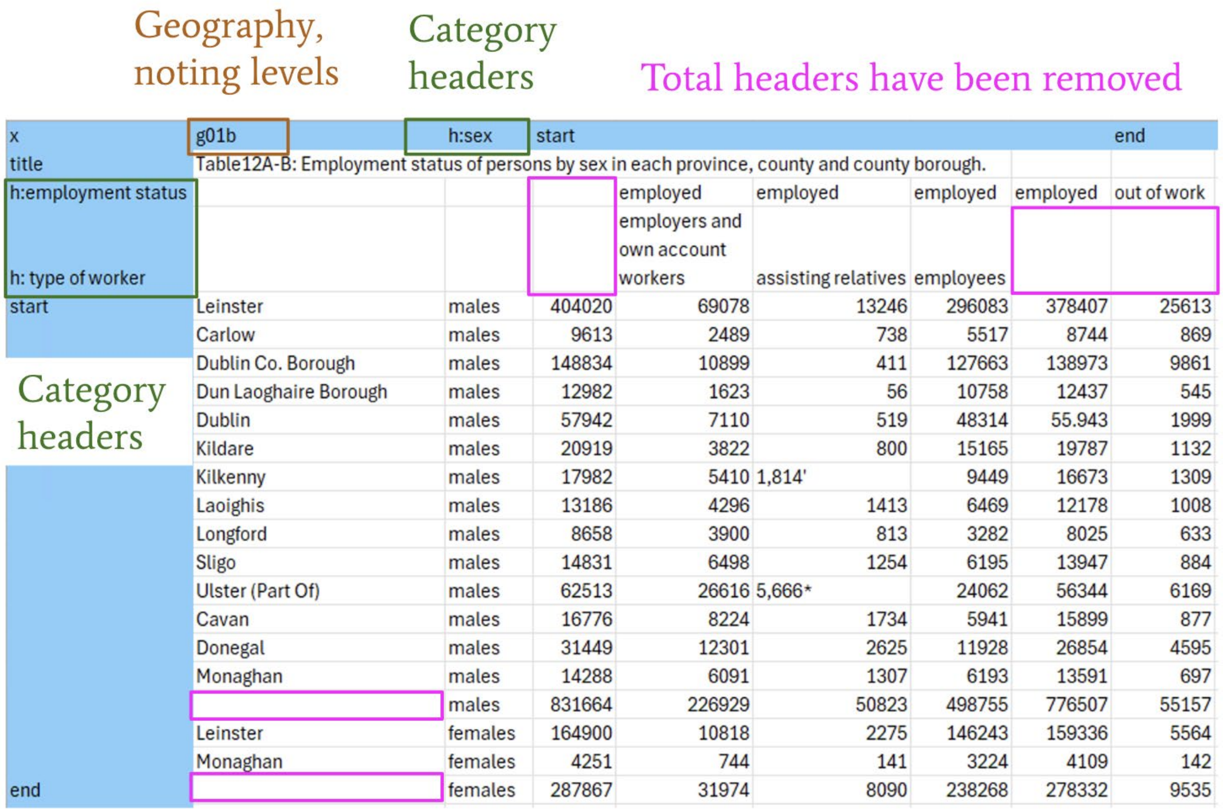
Following OCR, we apply a markup framework to indicate the structure of the source tables and facilitate automated conversion into standardized CSV files. The manual markup process identifies the location of key metadata elements within the table using standardized tags (Figure 2). Keywords are entered into the blue areas around the perimeter of the table to tag the locations of category headers, geographic unit names, and the beginning and end of the data. The structural tags are used to restructure the marked up table and build a standard CSV file where each column and row contains full metadata information.

Another integral feature of markup files are the blank cells used to indicate row and column totals (highlighted in pink in Figure 2). The blank cells play a vital role in interpreting relationships in the standard CSV file.

1.2.a. Geographic dictionaries

Tables often contain data for multiple levels in a hierarchy of sub-national units, where the sum across groups of child units should match the total for their parents. We construct a geographic dictionary (“geog dictionary”) for each census dataset to capture the nested parent-child relationships among all units present in the tables. Geog dictionaries are typically based on a table that contains the finest geographic level present in all tables in the dataset. The software that converts markup files to standard CSVs uses the dataset’s geog dictionary to determine the parent, grandparent, etc. for each subnational unit in each table and fully specify hierarchical geographic relationships in the output.

Figure 2. Marked up table from 1971 Ireland Population and Housing Census. (Note: Rows have been truncated to show key markup features.)



x	g01b	h:sex	start					end
title	Table12A-B: Employment status of persons by sex in each province, county and county borough.							
h:employment status				employed	employed	employed	employed	out of work
h: type of worker				employers and own account workers	assisting relatives	employees		
start	Leinster	males	404020	69078	13246	296083	378407	25613
	Carlow	males	9613	2489	738	5517	8744	869
	Dublin Co. Borough	males	148834	10899	411	127663	138973	9861
	Dun Laoghaire Borough	males	12982	1623	56	10758	12437	545
	Dublin	males	57942	7110	519	48314	55.943	1999
	Kildare	males	20919	3822	800	15165	19787	1132
	Kilkenny	males	17982	5410	1,814'	9449	16673	1309
	Laoighis	males	13186	4296	1413	6469	12178	1008
	Longford	males	8658	3900	813	3282	8025	633
	Sligo	males	14831	6498	1254	6195	13947	884
	Ulster (Part Of)	males	62513	26616	5,666*	24062	56344	6169
	Cavan	males	16776	8224	1734	5941	15899	877
	Donegal	males	31449	12301	2625	11928	26854	4595
	Monaghan	males	14288	6091	1307	6193	13591	697
		males	831664	226929	50823	498755	776507	55157
	Leinster	females	164900	10818	2275	146243	159336	5564
	Monaghan	females	4251	744	141	3224	4109	142
end		females	287867	31974	8090	238268	278332	9535

2. Standard CSV Files

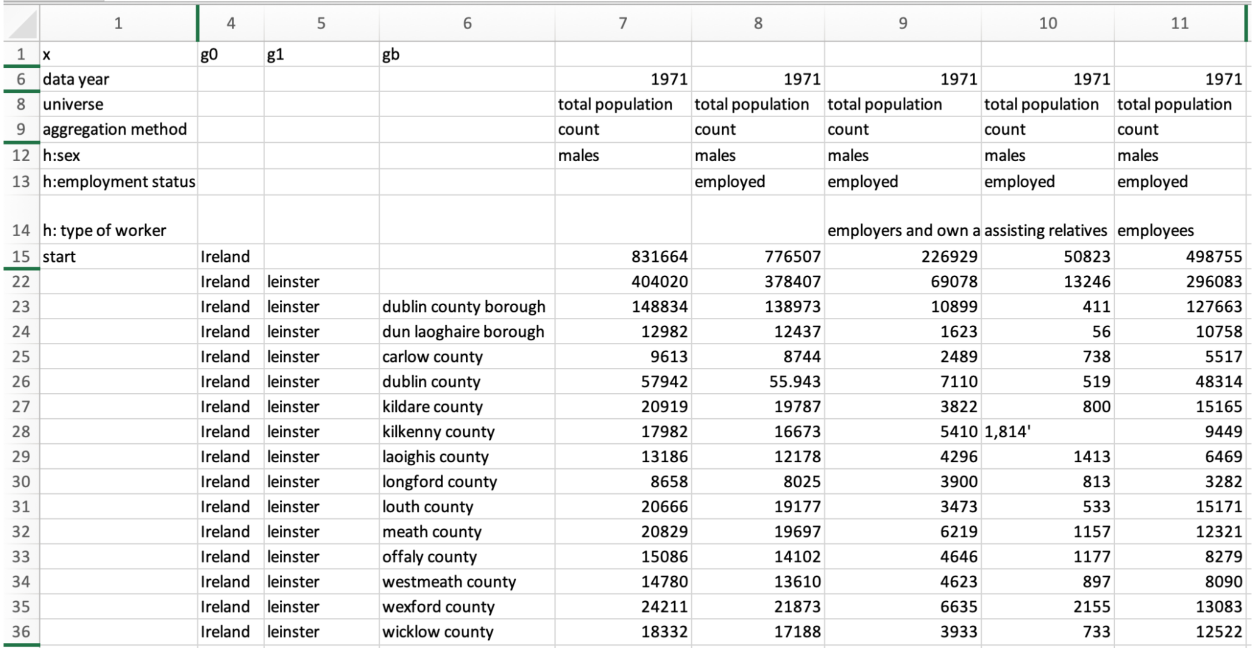
Internally developed Python software tools are used to convert from markup files to standard CSV format. The tools first verify the markup by checking for any missing elements or internal inconsistencies. Once workbooks have been fully verified, they are converted to standard CSVs, which involves (1) unstacking rows with non-geographic headers so each row in the standard CSV represents a geographic unit and all other variables are in columns, and (2) expanding hierarchical geographic unit relationships.

2.1. Unstacking

Tables are restructured so each row represents a geographic unit and each column is a measured variable (Figure 3). If rows in the source table represent categories rather than geographic units, they are unstacked into additional columns. In the example table, each geographic unit appeared on two rows in the source table, once for males and once for females.

In the standard CSV file, each geographic unit has just one row with a set of columns for males and a second set of columns for females (out of frame to the right). The blank row and column header cells representing totals are retained from the markup and play an important role in identifying relationships for QA/QC. Any data values that were mis-recognized during OCR are still present in the standard CSV file.

Figure 3. Fully unstacked standard CSV file from the marked up 1971 Ireland Population and Housing Census table. (Note: Rows and columns have been truncated so that important standard CSV features are visible.)

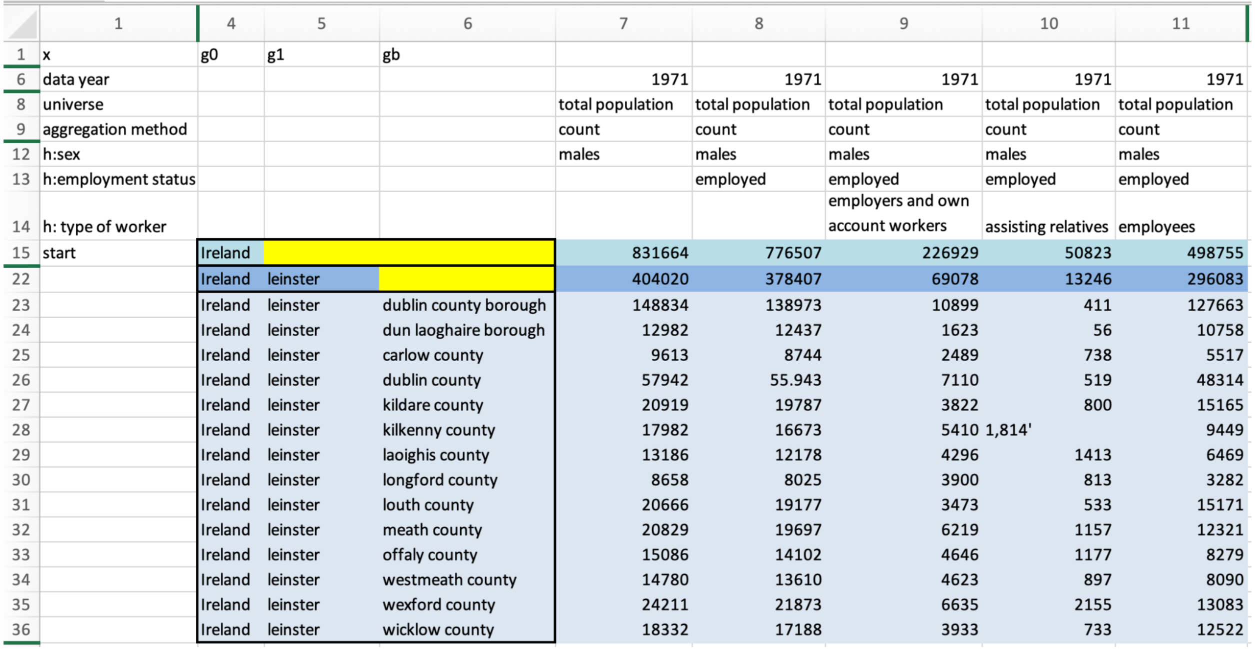


	1	4	5	6	7	8	9	10	11
1	x	g0	g1	gb					
6	data year				1971	1971	1971	1971	1971
8	universe				total population	total population	total population	total population	total population
9	aggregation method				count	count	count	count	count
12	h:sex				males	males	males	males	males
13	h:employment status					employed	employed	employed	employed
14	h: type of worker						employers and own a	assisting relatives	employees
15	start	Ireland			831664	776507	226929	50823	498755
22		Ireland	leinster		404020	378407	69078	13246	296083
23		Ireland	leinster	dublin county borough	148834	138973	10899	411	127663
24		Ireland	leinster	dun laoghaire borough	12982	12437	1623	56	10758
25		Ireland	leinster	carlow county	9613	8744	2489	738	5517
26		Ireland	leinster	dublin county	57942	55.943	7110	519	48314
27		Ireland	leinster	kildare county	20919	19787	3822	800	15165
28		Ireland	leinster	kilkenny county	17982	16673	5410	1,814'	9449
29		Ireland	leinster	laoighis county	13186	12178	4296	1413	6469
30		Ireland	leinster	longford county	8658	8025	3900	813	3282
31		Ireland	leinster	louth county	20666	19177	3473	533	15171
32		Ireland	leinster	meath county	20829	19697	6219	1157	12321
33		Ireland	leinster	offaly county	15086	14102	4646	1177	8279
34		Ireland	leinster	westmeath county	14780	13610	4623	897	8090
35		Ireland	leinster	wexford county	24211	21873	6635	2155	13083
36		Ireland	leinster	wicklow county	18332	17188	3933	733	12522

2.2. Expanding Geographic Hierarchy

The single column tagged "g01b" in the markup has been expanded to three distinct columns labeled "g0", "g1," and "gb" in the standard CSV file, based on the parent-child relationships identified in the geog dictionary. Parent columns are to the left of child columns, and parent unit names are repeated on the rows for each of their children. In our example, the province of Leinster (g1) has fourteen county-level children (gb), highlighted in blue (Figure 4). (Additional provinces are not shown in the illustration.) The first row for each province has a blank in column "gb," indicating that these rows are the *total* counts for the province. We expect each province total to equal the sum of its child county values. We also expect that the province totals all add up to the national total (row 15), which has blank cells in both columns "g1" and "gb.

Figure 4. Highlighted geog relationships from expanded geographic hierarchy. Rows with gb labels are interpreted as the children of their parent g1 unit, and therefore should add up to the g1 total row. Rows for g1 are interpreted as children of g0, and therefore should add up to the g0 national-level total row. (Rows have been truncated for simplicity.)



	1	4	5	6	7	8	9	10	11
1	x	g0	g1	gb					
6	data year				1971	1971	1971	1971	1971
8	universe				total population	total population	total population	total population	total population
9	aggregation method				count	count	count	count	count
12	h:sex				males	males	males	males	males
13	h:employment status					employed	employed	employed	employed
14	h: type of worker						employers and own account workers	assisting relatives	employees
15	start	Ireland			831664	776507	226929	50823	498755
22		Ireland	leinster		404020	378407	69078	13246	296083
23		Ireland	leinster	dublin county borough	148834	138973	10899	411	127663
24		Ireland	leinster	dun laoghaire borough	12982	12437	1623	56	10758
25		Ireland	leinster	carlow county	9613	8744	2489	738	5517
26		Ireland	leinster	dublin county	57942	55.943	7110	519	48314
27		Ireland	leinster	kildare county	20919	19787	3822	800	15165
28		Ireland	leinster	kilkenny county	17982	16673	5410	1,814'	9449
29		Ireland	leinster	laoighis county	13186	12178	4296	1413	6469
30		Ireland	leinster	longford county	8658	8025	3900	813	3282
31		Ireland	leinster	louth county	20666	19177	3473	533	15171
32		Ireland	leinster	meath county	20829	19697	6219	1157	12321
33		Ireland	leinster	offaly county	15086	14102	4646	1177	8279
34		Ireland	leinster	westmeath county	14780	13610	4623	897	8090
35		Ireland	leinster	wexford county	24211	21873	6635	2155	13083
36		Ireland	leinster	wicklow county	18332	17188	3933	733	12522

2.3. Additional metadata details

Standard CSV files also contain additional metadata describing the universe, data year, and aggregation method for each column, which may be relevant for QA/QC.

Universe

The universe describes the scope of what is counted or measured in each column. In our example table, the universe for the entire table is "total population." Some tables contain multiple universes. For example, one group of columns may count households by drinking water source and another group of columns may count individual people by drinking water source. In such cases, relationships will be built separately for each universe.

Data year

The data in each column represent a particular data year. (In our example, all data pertain to 1971 and the data year row is not shown.) Tables including historical data or projections will have different data years for different columns, and relationships will be built separately for each data year.

Aggregation method

Aggregation method indicates how the data from individual enumeration forms were summarized to generate the values in each column. Tables often include data that are not simple counts, such as percents, differences, or growth rates. The QA/QC tools do not currently support validation of non-count cells, and columns with an aggregation method other than count are removed from the validation process by default.

3. Automated QA/QC

The QA/QC R package takes standard CSV files as input and attempts to identify and update erroneous data values. The QA/QC process consists of four stages, corresponding to functions in the R package. First, a standard CSV file is read and the structured metadata are used to identify row (geographic) and column (h-tag) relationships (`load_std_csv()`). Then simple data cleaning is performed and relationships are validated by checking whether sums match given totals as expected (`validate_std_csv()`). Next, the package attempts to correct invalid relationships by leveraging intersecting relationships to pinpoint error cells and infer correct values (`update_std_csv()`). Finally, an updated version of the table is exported, highlighting any remaining invalid relationships for manual review (`write_std_csv()`).

3.1. Identifying Relationships

3.1.a. Loading a Standard CSV File

The package first reads in a standard CSV file using the `load_std_csv()` function. The output of this function is an `ihgis_std_csv` object that stores the raw contents of the input CSV. The function also parses geographic unit labels, h-tag labels, blank cells, and other metadata to identify relationships. For our example table from Ireland 1971, the package identifies 4 column relationships among the sex, employment status, and type of worker dimensions and 5 row relationships among the g0, g1, and gb geog levels (Figure 5).

Figure 5. An `ihgis_std_csv` object with identified data dimensions, metadata, and relationships.

```
Unprocessed <ihgis_std_csv> for file IE1971pop_3-1_Table12A-B.csv
  Data dimensions: [37, 12]
  h-tag Relationships: 4
    • Levels: h:sex, h:employment status, h: type of worker
  geog Relationships: 5
    • Levels: g0, g1, gb
```

The `$relate` field of the `ihgis_std_csv` object contains information about the column and row relationships. Printing the `$relate` field displays the indices or header labels of the columns or rows participating in each relationship (Figure 6). Pipes (‘|’) are used to concatenate labels across levels and the initial ‘.’ on the right side of the equations is shorthand for the labels on the left side. For example, column 7 (males) should equal the sum of columns 8 (males | employed) and 12 (males | out of work), and row 48 (Ireland | Ulster (part of)) should equal the sum of rows 49, 50, and 51, the three counties in Ulster (Cavan, Donegal, and Monaghan). The columns or rows that represent the total value (recognized due to a blank cell) are referred to as parents, while the columns or rows on the right side are referred to as children. At this stage, just the structure of the relationships has been identified; the values have not yet been validated.

Figure 6. An `ihgis_std_csv` relate field with all relationships identified.

```
--- Column relationships ---
• 7 = 8 + 12
• 8 = 9 + 10 + 11
• 13 = 17 + 18
• 17 = 14 + 15 + 16
--- Column relationships ---
• males = .|employed + .|out of work
• males|employed = .|employers and own account workers + .|assisting relatives +
.|employees
• females = .|employed + .|out of work
• females|employed = .|employers and own account workers + .|assisting relatives +
.|employees

--- Row relationships ---
• 15 = 16 + 22 + 37 + 48
• 16 = 17 + 18 + 19 + 20 + 21
• 22 = 23 + 24 + 25 + 26 + 27 + 28 + 29 + 30 + 31 + 32 + 33 + 34 + 35 + 36
• 37 = 38 + 39 + 40 + 41 + 42 + 43 + 44 + 45 + 46 + 47
• 48 = 49 + 50 + 51
--- Row relationships ---
• Ireland = .|connacht + .|leinster + .|munster + .|ulster (part of)
• Ireland|connacht = .|galway county + .|leitrim county + .|mayo county +
.|roscommon county +
  .|sligo county
```

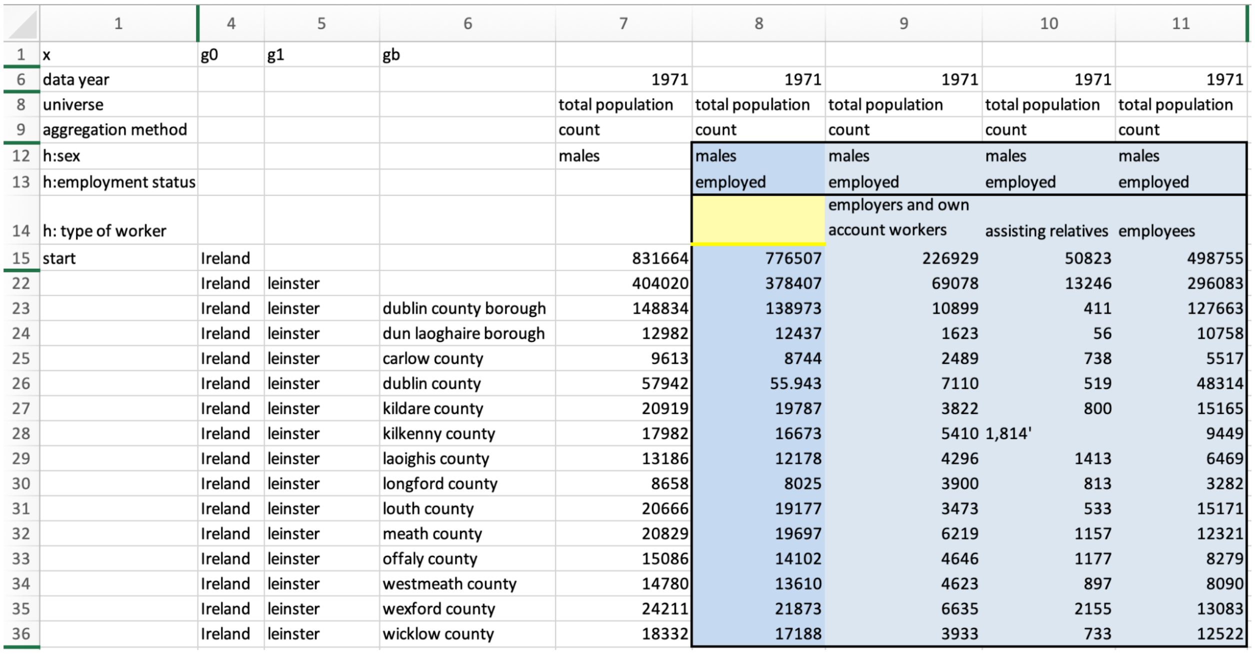
```
• Ireland|leinster = .|dublin county borough + .|dun laoghaire borough + .|carlow
county +
  .|dublin county + .|kildare county + .|kilkenny county + .|laoighis county +
.|longford county
  + .|louth county + .|meath county + .|offaly county + .|westmeath county +
.|wexford county +
  .|wicklow county
• Ireland|munster = .|cork county borough + .|limerick county borough + .|waterford
county
  borough + .|clare county + .|cork county + .|kerry county + .|limerick county +
.|tipperary
  county, north riding + .|tipperary county, south riding + .|waterford county
• Ireland|ulster (part of) = .|cavan county + .|donegal county + .|monaghan county
```

3.1.b. Meaningful Blanks in Categorical Labels

To better understand how the package interprets relationships, it is important to revisit the meaningful blanks in a standard CSV file. As noted, blanks indicate parents totals in relationships, but the structure differs between geog and h-tag relationships. Geographic relationships entail a strict parent-child nested hierarchy, and the parent columns are always to the left of child columns and repeated on the rows for each of their children.

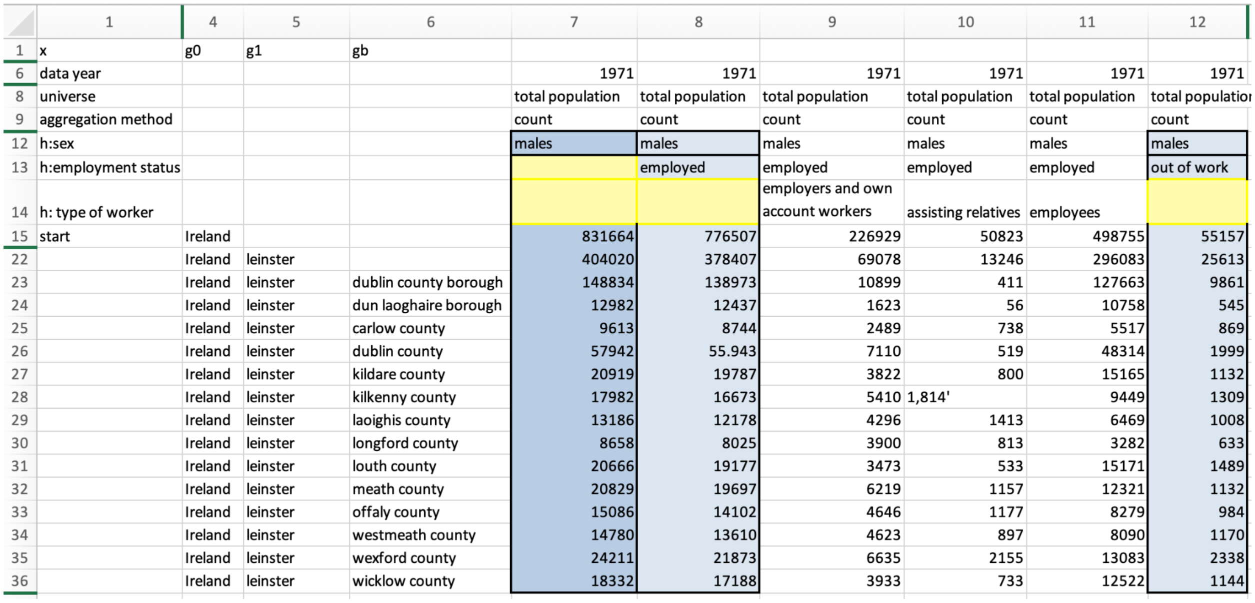
Identifying categorical (“h-tag”) relationships among the columns is somewhat complicated by the lack of a strict parent-child hierarchy. The first colum of our example standard CSV contains three rows having tags of the form “h:{dimension}”, for the dimensions sex, employment status and type of worker (Figures 7 and 8). Similar to the geographic metadata, blank cells among the category labels on each of these rows indicate columns representing a total over that row’s dimension. However, unlike the geographic relationships, the order of the header rows does not necessarily follow a clear hierarchical relationship. Children of a given parent are identified as columns that have the same non-blank values in all other header rows. For example, column 8 has “males” in the h:sex row, “employed” in the h:employment status row, and a blank in the h:type of worker row, indicating that it is a total column. Its children are columns 9, 10, and 11, which are also “males” and “employed” and have categories of type of worker specified (Figure 7). This is relationship is seen in the `$relate` field of the standard CSV object as `males|employed = .|employers and own account workers + .|assisting relatives + .|employees`. Similarly, column 7 has “males” in the h:sex row and blanks for both h:employment status and h:type of worker. Its children are columns 8 and 11, which are also “males” and have employment status categories “employed” and “out of work” (Figure 8).

Figure 7. Highlighted h-tag relationship: total employed males = employed males who are employers and own account workers + employed males assisting relatives + employed males that are employees.



	1	4	5	6	7	8	9	10	11
1	x	g0	g1	gb					
6	data year				1971	1971	1971	1971	1971
8	universe				total population	total population	total population	total population	total population
9	aggregation method				count	count	count	count	count
12	h:sex				males	males	males	males	males
13	h:employment status					employed	employed	employed	employed
14	h: type of worker						employers and own account workers	assisting relatives	employees
15	start	Ireland			831664	776507	226929	50823	498755
22		Ireland	leinster		404020	378407	69078	13246	296083
23		Ireland	leinster	dublin county borough	148834	138973	10899	411	127663
24		Ireland	leinster	dun laoghaire borough	12982	12437	1623	56	10758
25		Ireland	leinster	carlow county	9613	8744	2489	738	5517
26		Ireland	leinster	dublin county	57942	55.943	7110	519	48314
27		Ireland	leinster	kildare county	20919	19787	3822	800	15165
28		Ireland	leinster	kilkenny county	17982	16673	5410	1,814'	9449
29		Ireland	leinster	laoighis county	13186	12178	4296	1413	6469
30		Ireland	leinster	longford county	8658	8025	3900	813	3282
31		Ireland	leinster	louth county	20666	19177	3473	533	15171
32		Ireland	leinster	meath county	20829	19697	6219	1157	12321
33		Ireland	leinster	offaly county	15086	14102	4646	1177	8279
34		Ireland	leinster	westmeath county	14780	13610	4623	897	8090
35		Ireland	leinster	wexford county	24211	21873	6635	2155	13083
36		Ireland	leinster	wicklow county	18332	17188	3933	733	12522

Figure 8. Highlighted h-tag relationship: total males = employed males + males out of work.



	1	4	5	6	7	8	9	10	11	12
1	x	g0	g1	gb						
6	data year				1971	1971	1971	1971	1971	1971
8	universe				total population	total population	total population	total population	total population	total populatio
9	aggregation method				count	count	count	count	count	count
12	h:sex				males	males	males	males	males	males
13	h:employment status					employed	employed	employed	employed	out of work
14	h: type of worker						employers and own account workers	assisting relatives	employees	
15	start	Ireland			831664	776507	226929	50823	498755	55157
22		Ireland	leinster		404020	378407	69078	13246	296083	25613
23		Ireland	leinster	dublin county borough	148834	138973	10899	411	127663	9861
24		Ireland	leinster	dun laoghaire borough	12982	12437	1623	56	10758	545
25		Ireland	leinster	carlow county	9613	8744	2489	738	5517	869
26		Ireland	leinster	dublin county	57942	55.943	7110	519	48314	1999
27		Ireland	leinster	kildare county	20919	19787	3822	800	15165	1132
28		Ireland	leinster	kilkenny county	17982	16673	5410	1,814'	9449	1309
29		Ireland	leinster	laoighis county	13186	12178	4296	1413	6469	1008
30		Ireland	leinster	longford county	8658	8025	3900	813	3282	633
31		Ireland	leinster	louth county	20666	19177	3473	533	15171	1489
32		Ireland	leinster	meath county	20829	19697	6219	1157	12321	1132
33		Ireland	leinster	offaly county	15086	14102	4646	1177	8279	984
34		Ireland	leinster	westmeath county	14780	13610	4623	897	8090	1170
35		Ireland	leinster	wexford county	24211	21873	6635	2155	13083	2338
36		Ireland	leinster	wicklow county	18332	17188	3933	733	12522	1144

These two relationships also highlight another important feature of standard CSV files: rows and columns may participate in multiple relationships. Total employed males in column 8 is both a child *and* a parent; it is both an element that adds up to the total across all males as well as a total value itself. Data values in these columns can also participate in geographic relationships, representing intersecting relationships.

In the Ireland table example, the parent category labels appear on rows above the child categories, but this is not always the case. For example, a table might include categories for male/female in one row, urban/rural on the next row, and marital status on a third row. A column may then have a blank in the top row, “urban” in the middle row, and “married” in the third row. This column is a total across married people in urban areas and should equal the sum of “male | urban | married” plus “female | urban | married.” Additionally, if universe or data year vary across the columns in a table, columns are grouped into subsets with matching universe and data year before identifying relationships.

3.2. Cleaning

Due to the lack of internal validation, OCR commonly mis-recognizes numbers as letters or punctuation. Prior to validating relationships, the function `validate_std_csv()` performs basic cleaning to convert specific commonly-substituted letters and symbols to numeric digits. For example, upper- or lower-case letter “O” is converted to zero, and upper- and lower-case “I”, lower-case “L”, and punctuation like “]”, “!”, and “|” are converted to the digit “1”. Other punctuation marks that do not have a clear corresponding digit are removed in an attempt to produce meaningful numeric values. For example, the value 1,814’ seen in the previous figures (row 28, column 10) is cleaned by removing the apostrophe. The locations of cleaned cells are stored in a `$cleaned` field in the ‘ihgis_std_csv’ object (Figure 9). Cells that contain remaining non-numeric characters after cleaning are treated as having a value of 0 for the purpose of validating relationships.

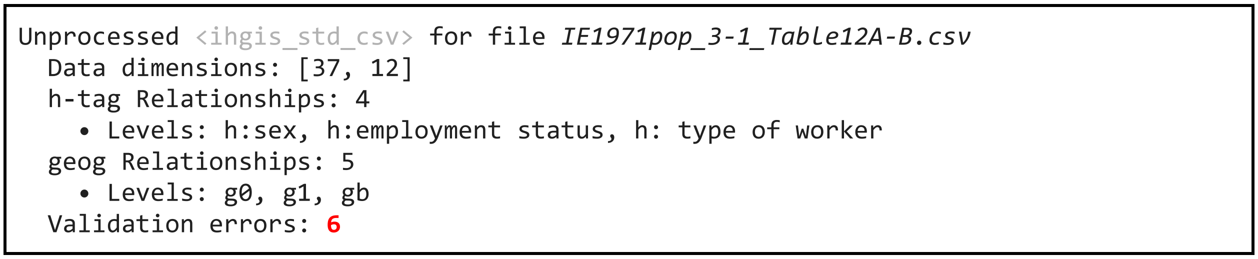
Figure 9. Values that were stripped of punctuation and stored as cleaned values from standard CSV.

```
$cleaned
# A tibble: 1 × 3
   rows  cols  diff
  <int> <int> <dbl>
1    28    10  1814
```

3.3. Validating Relationships

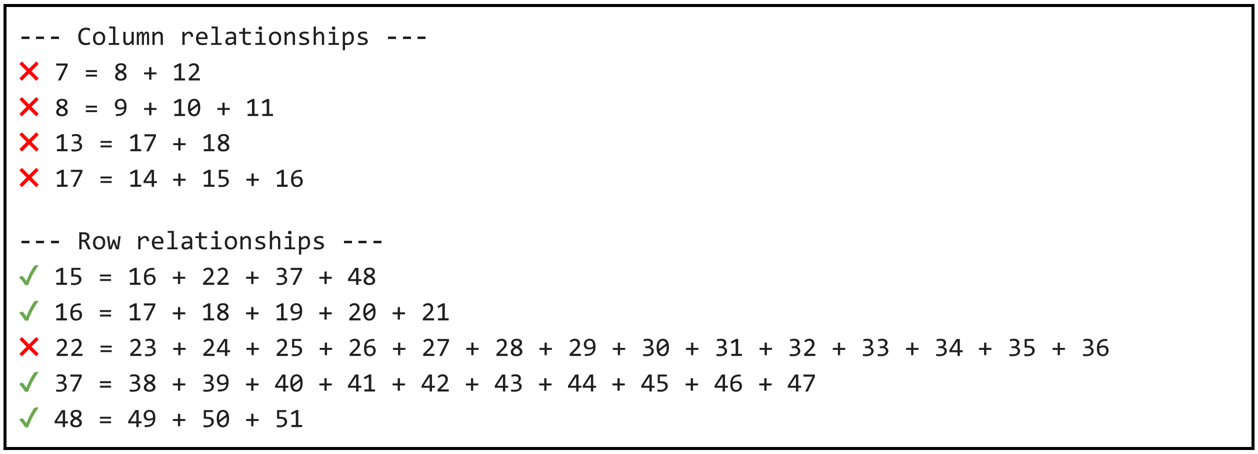
After cleaning, `validate_std_csv()` calculates the sums of child cells in each identified relationship and checks the sum against the total reported in the relationship’s parent cell. Instances where the child sum does not match the parent total are recorded in the `ihgis_std_csv` object. Printing the object now reports the number of validation errors (Figure 10). Printing the `$relate` field of the object indicates which relationships include instances of validation errors by marking them with a red ‘X’ (Figure 11). In our example, all the column relationships and one of the row relationships contain errors. This means that in each of the column relationships, there is at least one row that doesn’t add up as expected, and in relationship with row 22 as the parent, at least one column doesn’t add up. At this point, the exact cells within these relationships that contain incorrect digits are not known; the parent and/or any child cells could contain errors.

Figure 10. A validated `ihgis_std_csv` object with added information about possible data errors.



```
Unprocessed <ihgis_std_csv> for file IE1971pop_3-1_Table12A-B.csv
  Data dimensions: [37, 12]
  h-tag Relationships: 4
    • Levels: h:sex, h:employment status, h: type of worker
  geog Relationships: 5
    • Levels: g0, g1, gb
  Validation errors: 6
```

Figure 11. An `ihgis_std_csv` relate field with all relationships identified.



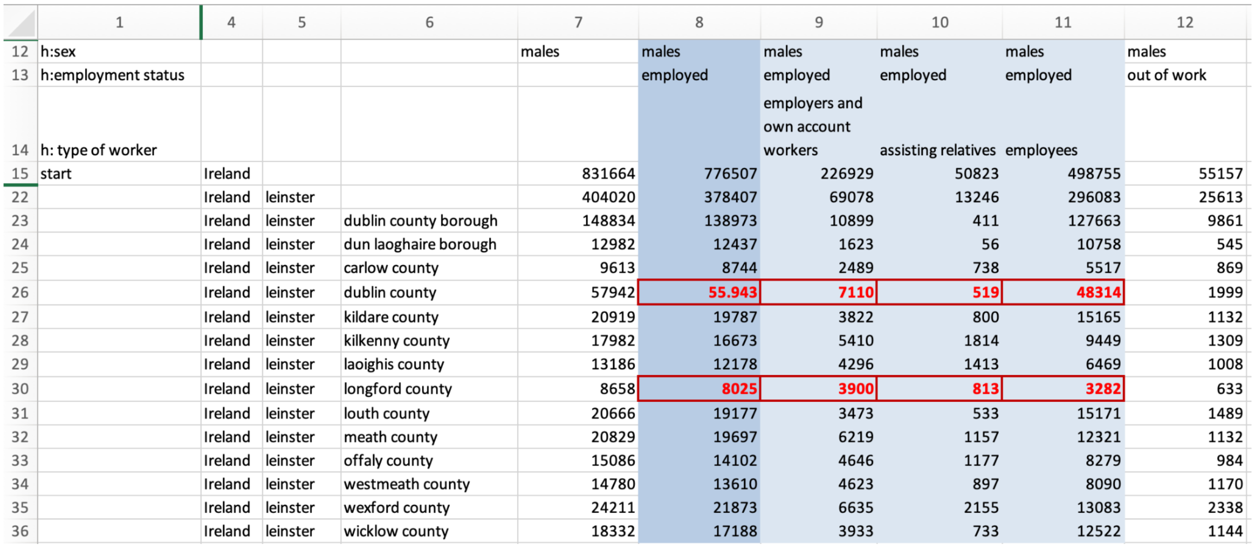
```
--- Column relationships ---
✗ 7 = 8 + 12
✗ 8 = 9 + 10 + 11
✗ 13 = 17 + 18
✗ 17 = 14 + 15 + 16

--- Row relationships ---
✓ 15 = 16 + 22 + 37 + 48
✓ 16 = 17 + 18 + 19 + 20 + 21
✗ 22 = 23 + 24 + 25 + 26 + 27 + 28 + 29 + 30 + 31 + 32 + 33 + 34 + 35 + 36
✓ 37 = 38 + 39 + 40 + 41 + 42 + 43 + 44 + 45 + 46 + 47
✓ 48 = 49 + 50 + 51
```

3.4. Pinpointing Error Cells

We can start to see the power of intersecting relationships by looking at the invalid relationships identified in a section of our example table (Figures 12a-c). For this example, we will focus on the columns for males and the rows for Leinster and its child counties. For the column relationship with parent column 8 (‘males : employed’ equals the sum of the three types of worker), rows 26 and 30 do not add up as expected (Figure 12a). Similarly, for the column relationship ‘males’ = ‘males | employed’ + ‘males | out of work’, row 26 is invalid (Figure 12b).

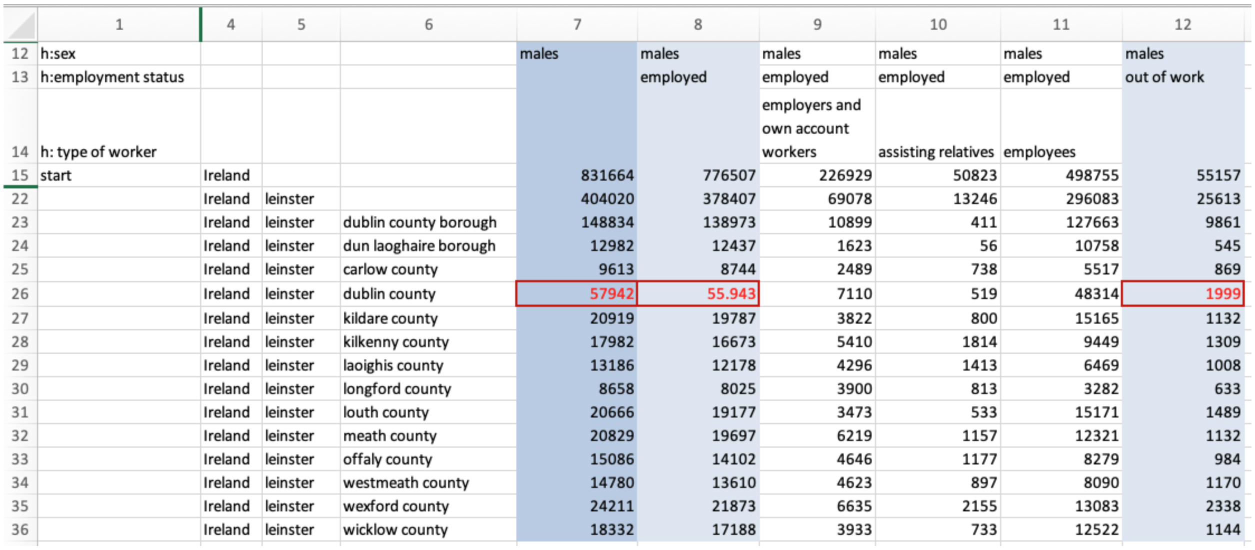
Finally, for the row relationship with row 22 as the parent (Leinster and its gb children), columns 8 and 10 don't add up (Figure 12c). Looking at each of these relationships independently, it is not clear which cell or cells contain errors. But when viewed together, it becomes clear that the cells that are part of multiple invalid relationships are the likely locations of OCR errors.

Figure 12a. Identified inconsistent sums for parent column 8 ('males : employed' equals the sum of the three types of employment) on rows 26 and 30. Red cells are identified as being involved in an inconsistent relationship. Blue shading indicates relationships identified by the package, with parents in darker blue and children in lighter blue.



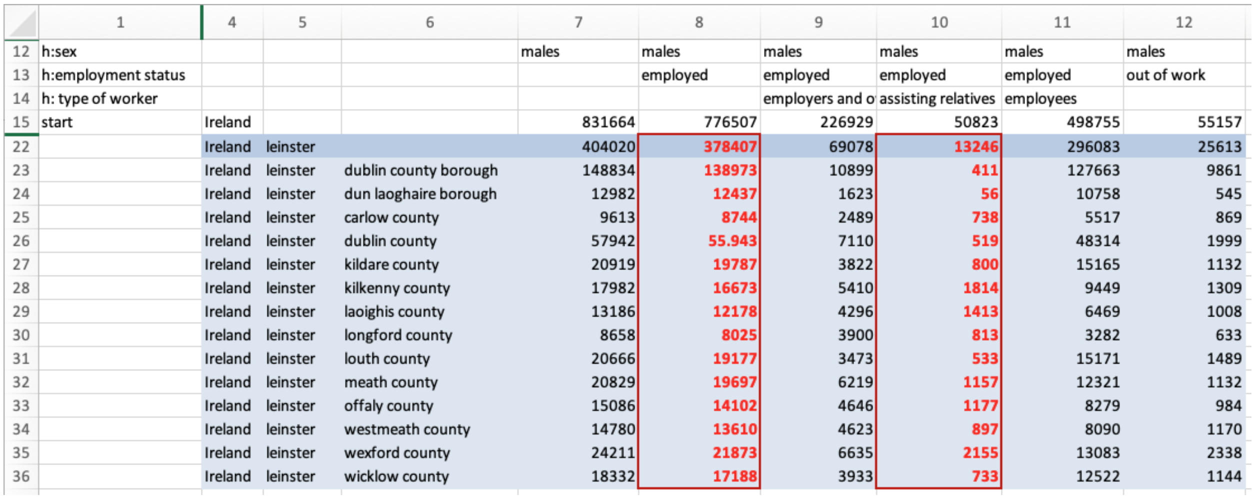
	1	4	5	6	7	8	9	10	11	12
12	h:sex				males	males	males	males	males	males
13	h:employment status					employed	employed	employed	employed	out of work
14	h: type of worker						employers and own account workers	assisting relatives	employees	
15	start	Ireland			831664	776507	226929	50823	498755	55157
22		Ireland	leinster		404020	378407	69078	13246	296083	25613
23		Ireland	leinster	dublin county borough	148834	138973	10899	411	127663	9861
24		Ireland	leinster	dun laoghaire borough	12982	12437	1623	56	10758	545
25		Ireland	leinster	carlow county	9613	8744	2489	738	5517	869
26		Ireland	leinster	dublin county	57942	55.943	7110	519	48314	1999
27		Ireland	leinster	kildare county	20919	19787	3822	800	15165	1132
28		Ireland	leinster	kilkenny county	17982	16673	5410	1814	9449	1309
29		Ireland	leinster	laoighis county	13186	12178	4296	1413	6469	1008
30		Ireland	leinster	longford county	8658	8025	3900	813	3282	633
31		Ireland	leinster	louth county	20666	19177	3473	533	15171	1489
32		Ireland	leinster	meath county	20829	19697	6219	1157	12321	1132
33		Ireland	leinster	offaly county	15086	14102	4646	1177	8279	984
34		Ireland	leinster	westmeath county	14780	13610	4623	897	8090	1170
35		Ireland	leinster	wexford county	24211	21873	6635	2155	13083	2338
36		Ireland	leinster	wicklow county	18332	17188	3933	733	12522	1144

Figure 12b. Identified inconsistent sums for parent column 7 ('males' = 'males : employed' + 'males : out of work') on row 26.



	1	4	5	6	7	8	9	10	11	12
12	h:sex				males	males	males	males	males	males
13	h:employment status					employed	employed	employed	employed	out of work
14	h: type of worker						employers and own account workers	assisting relatives	employees	
15	start	Ireland			831664	776507	226929	50823	498755	55157
22		Ireland	leinster		404020	378407	69078	13246	296083	25613
23		Ireland	leinster	dublin county borough	148834	138973	10899	411	127663	9861
24		Ireland	leinster	dun laoghaire borough	12982	12437	1623	56	10758	545
25		Ireland	leinster	carlow county	9613	8744	2489	738	5517	869
26		Ireland	leinster	dublin county	57942	55.943	7110	519	48314	1999
27		Ireland	leinster	kildare county	20919	19787	3822	800	15165	1132
28		Ireland	leinster	kilkenny county	17982	16673	5410	1814	9449	1309
29		Ireland	leinster	laoighis county	13186	12178	4296	1413	6469	1008
30		Ireland	leinster	longford county	8658	8025	3900	813	3282	633
31		Ireland	leinster	louth county	20666	19177	3473	533	15171	1489
32		Ireland	leinster	meath county	20829	19697	6219	1157	12321	1132
33		Ireland	leinster	offaly county	15086	14102	4646	1177	8279	984
34		Ireland	leinster	westmeath county	14780	13610	4623	897	8090	1170
35		Ireland	leinster	wexford county	24211	21873	6635	2155	13083	2338
36		Ireland	leinster	wicklow county	18332	17188	3933	733	12522	1144

Figure 12c. Identified inconsistent sums for row 22 as the parent (Leinster and its gb children) where columns 8 and 10 don't add up

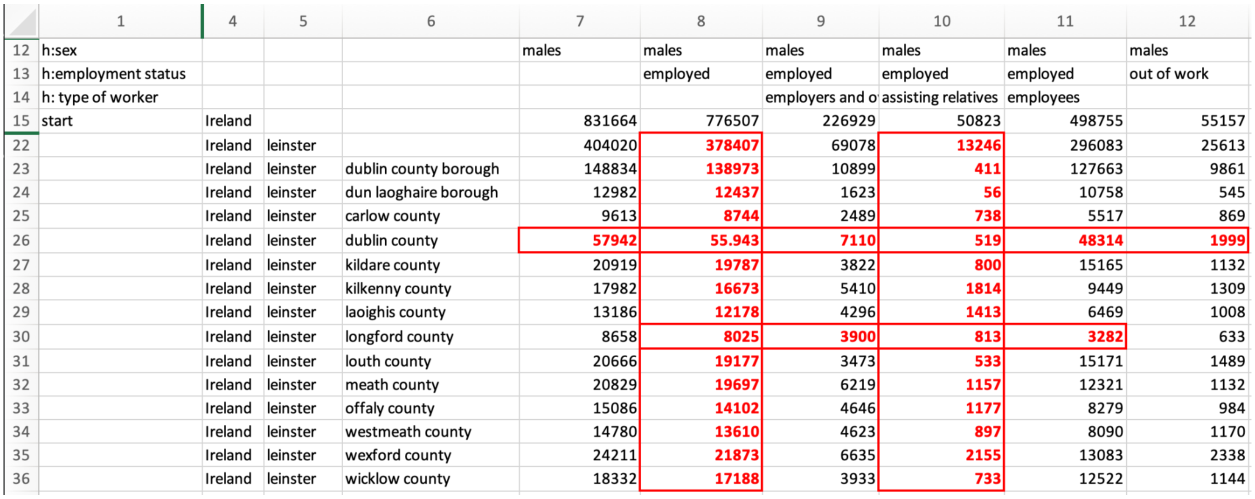


	1	4	5	6	7	8	9	10	11	12
12	h:sex				males	males	males	males	males	males
13	h:employment status					employed	employed	employed	employed	out of work
14	h: type of worker						employers and o	assisting relatives	employees	
15	start	Ireland			831664	776507	226929	50823	498755	55157
22		Ireland	leinster		404020	378407	69078	13246	296083	25613
23		Ireland	leinster	dublin county borough	148834	138973	10899	411	127663	9861
24		Ireland	leinster	dun laoghaire borough	12982	12437	1623	56	10758	545
25		Ireland	leinster	carlow county	9613	8744	2489	738	5517	869
26		Ireland	leinster	dublin county	57942	55.943	7110	519	48314	1999
27		Ireland	leinster	kildare county	20919	19787	3822	800	15165	1132
28		Ireland	leinster	kilkenny county	17982	16673	5410	1814	9449	1309
29		Ireland	leinster	laoighis county	13186	12178	4296	1413	6469	1008
30		Ireland	leinster	longford county	8658	8025	3900	813	3282	633
31		Ireland	leinster	louth county	20666	19177	3473	533	15171	1489
32		Ireland	leinster	meath county	20829	19697	6219	1157	12321	1132
33		Ireland	leinster	offaly county	15086	14102	4646	1177	8279	984
34		Ireland	leinster	westmeath county	14780	13610	4623	897	8090	1170
35		Ireland	leinster	wexford county	24211	21873	6635	2155	13083	2338
36		Ireland	leinster	wicklow county	18332	17188	3933	733	12522	1144

The `validate_std_csv()` function leverages intersecting relationships to narrow in on a set of candidate error cells. The initial set of candidate error cells includes all cells that participate in invalid relationships (highlighted in red in Figures 12a-c). Based on the idea that an incorrect cell is likely to invalidate every relationship it participates in, the function drops cells that participate in valid relationships from the set of candidate error cells. The remaining candidate error cells are then subject to updates based on inferred values.

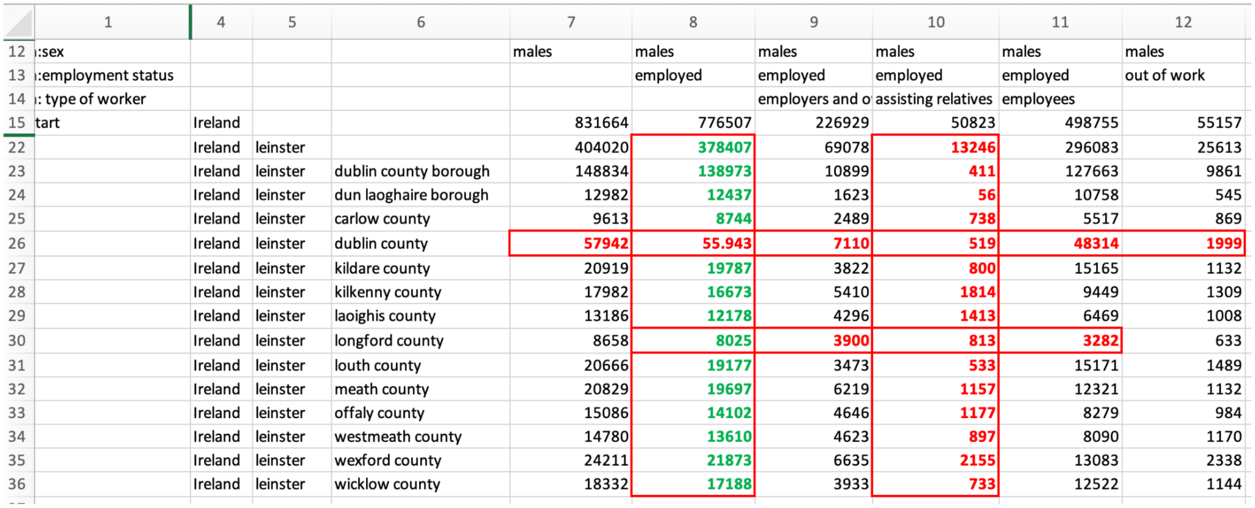
In our example, all cells in columns 8 and 10, all cells in row 26 (Dublin County), and cells in columns 8 through 11 on row 30 (Longford County) are initially considered candidate error cells because they participate in invalid relationships (Figure 13a). We can see how this set is narrowed down by progressively dropping cells that participate in valid relationships for each relationship in turn. The column relationship 7 = 8 + 12, is valid for all rows except Dublin (row 26), so cells in column 8 on all other rows are dropped from the candidate error set (green text in Figure 13b). Similarly, the column relationship 8 = 9 + 10 + 11 is valid for all rows except 26 and 30, so most cells in column 10 are dropped (Figure 13c). Finally, the row relationship for Leinster and its child counties is valid for columns 7, 9, 11, and 12, so the remaining candidate error cells in those columns are dropped (Figure 13d). After considering all three relationships, we are left with three candidate error cells in this section, those still marked in red text in Figure 13d [(26, 8), (26, 10), and (30, 10)].

Figure 13a. All cells participating in invalid relationships are initially considered candidate error cells.



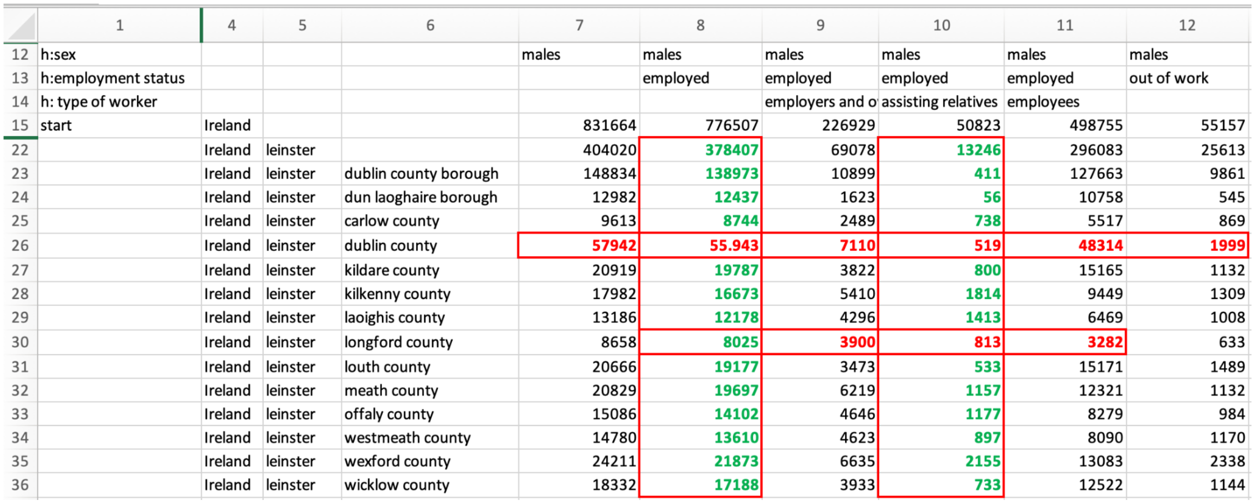
	1	4	5	6	7	8	9	10	11	12
12	h:sex				males	males	males	males	males	males
13	h:employment status					employed	employed	employed	employed	out of work
14	h: type of worker						employers and o	assisting relatives	employees	
15	start	Ireland			831664	776507	226929	50823	498755	55157
22		Ireland	leinster		404020	378407	69078	13246	296083	25613
23		Ireland	leinster	dublin county borough	148834	138973	10899	411	127663	9861
24		Ireland	leinster	dun laoghaire borough	12982	12437	1623	56	10758	545
25		Ireland	leinster	carlow county	9613	8744	2489	738	5517	869
26		Ireland	leinster	dublin county	57942	55.943	7110	519	48314	1999
27		Ireland	leinster	kildare county	20919	19787	3822	800	15165	1132
28		Ireland	leinster	kilkenny county	17982	16673	5410	1814	9449	1309
29		Ireland	leinster	laoighis county	13186	12178	4296	1413	6469	1008
30		Ireland	leinster	longford county	8658	8025	3900	813	3282	633
31		Ireland	leinster	louth county	20666	19177	3473	533	15171	1489
32		Ireland	leinster	meath county	20829	19697	6219	1157	12321	1132
33		Ireland	leinster	offaly county	15086	14102	4646	1177	8279	984
34		Ireland	leinster	westmeath county	14780	13610	4623	897	8090	1170
35		Ireland	leinster	wexford county	24211	21873	6635	2155	13083	2338
36		Ireland	leinster	wicklow county	18332	17188	3933	733	12522	1144

Figure 13b. Column relationship 7 = 8 + 12 is valid for all rows except Dublin (row 26), so all other cells in column 8 are dropped from the set of candidate error cells.



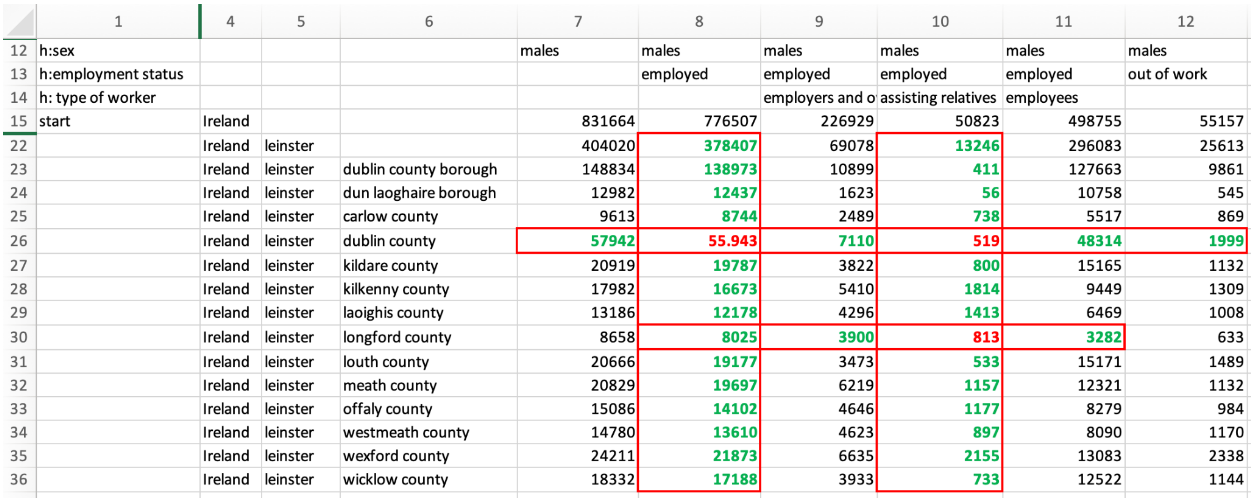
	1	4	5	6	7	8	9	10	11	12
12	:sex				males	males	males	males	males	males
13	:employment status					employed	employed	employed	employed	out of work
14	: type of worker						employers and o	assisting relatives	employees	
15	tart	Ireland			831664	776507	226929	50823	498755	55157
22		Ireland	leinster		404020	378407	69078	13246	296083	25613
23		Ireland	leinster	dublin county borough	148834	138973	10899	411	127663	9861
24		Ireland	leinster	dun laoghaire borough	12982	12437	1623	56	10758	545
25		Ireland	leinster	carlow county	9613	8744	2489	738	5517	869
26		Ireland	leinster	dublin county	57942	55.943	7110	519	48314	1999
27		Ireland	leinster	kildare county	20919	19787	3822	800	15165	1132
28		Ireland	leinster	kilkenny county	17982	16673	5410	1814	9449	1309
29		Ireland	leinster	laoighis county	13186	12178	4296	1413	6469	1008
30		Ireland	leinster	longford county	8658	8025	3900	813	3282	633
31		Ireland	leinster	louth county	20666	19177	3473	533	15171	1489
32		Ireland	leinster	meath county	20829	19697	6219	1157	12321	1132
33		Ireland	leinster	offaly county	15086	14102	4646	1177	8279	984
34		Ireland	leinster	westmeath county	14780	13610	4623	897	8090	1170
35		Ireland	leinster	wexford county	24211	21873	6635	2155	13083	2338
36		Ireland	leinster	wicklow county	18332	17188	3933	733	12522	1144

Figure 13c. Column relationship 8 = 9 + 10 + 11 is valid for all rows except 26 and 30, so most cells in column 10 are dropped.



	1	4	5	6	7	8	9	10	11	12
12	h:sex				males	males	males	males	males	males
13	h:employment status					employed	employed	employed	employed	out of work
14	h: type of worker						employers and o	assisting relatives	employees	
15	start	Ireland			831664	776507	226929	50823	498755	55157
22		Ireland	leinster		404020	378407	69078	13246	296083	25613
23		Ireland	leinster	dublin county borough	148834	138973	10899	411	127663	9861
24		Ireland	leinster	dun laoghaire borough	12982	12437	1623	56	10758	545
25		Ireland	leinster	carlow county	9613	8744	2489	738	5517	869
26		Ireland	leinster	dublin county	57942	55.943	7110	519	48314	1999
27		Ireland	leinster	kildare county	20919	19787	3822	800	15165	1132
28		Ireland	leinster	kilkenny county	17982	16673	5410	1814	9449	1309
29		Ireland	leinster	laoighis county	13186	12178	4296	1413	6469	1008
30		Ireland	leinster	longford county	8658	8025	3900	813	3282	633
31		Ireland	leinster	louth county	20666	19177	3473	533	15171	1489
32		Ireland	leinster	meath county	20829	19697	6219	1157	12321	1132
33		Ireland	leinster	offaly county	15086	14102	4646	1177	8279	984
34		Ireland	leinster	westmeath county	14780	13610	4623	897	8090	1170
35		Ireland	leinster	wexford county	24211	21873	6635	2155	13083	2338
36		Ireland	leinster	wicklow county	18332	17188	3933	733	12522	1144

Figure 13d. Row relationship for Leinster and its child counties is valid for columns 7, 9, 11, and 12, so the remaining candidate error cells in those columns are dropped. After considering all three relationships, we are left with three candidate error cells in this section (marked in red text).



	1	4	5	6	7	8	9	10	11	12
12	h:sex				males	males	males	males	males	males
13	h:employment status					employed	employed	employed	employed	out of work
14	h: type of worker						employers and o	assisting relatives	employees	
15	start	Ireland			831664	776507	226929	50823	498755	55157
22		Ireland	leinster		404020	378407	69078	13246	296083	25613
23		Ireland	leinster	dublin county borough	148834	138973	10899	411	127663	9861
24		Ireland	leinster	dun laoghaire borough	12982	12437	1623	56	10758	545
25		Ireland	leinster	carlow county	9613	8744	2489	738	5517	869
26		Ireland	leinster	dublin county	57942	55.943	7110	519	48314	1999
27		Ireland	leinster	kildare county	20919	19787	3822	800	15165	1132
28		Ireland	leinster	kilkenny county	17982	16673	5410	1814	9449	1309
29		Ireland	leinster	laoighis county	13186	12178	4296	1413	6469	1008
30		Ireland	leinster	longford county	8658	8025	3900	813	3282	633
31		Ireland	leinster	louth county	20666	19177	3473	533	15171	1489
32		Ireland	leinster	meath county	20829	19697	6219	1157	12321	1132
33		Ireland	leinster	offaly county	15086	14102	4646	1177	8279	984
34		Ireland	leinster	westmeath county	14780	13610	4623	897	8090	1170
35		Ireland	leinster	wexford county	24211	21873	6635	2155	13083	2338
36		Ireland	leinster	wicklow county	18332	17188	3933	733	12522	1144

3.5. Updating Values

After candidate error cells have been narrowed down, remaining candidate error cells are stored in the `$validation` field of the `ihgis_std_csv` object (Figure 14a). The `diff` column in this table is the difference between the sum of the children compared to their parent's total. Each set of child cells and their associated parent cell are identified by a unique hash value in the `relate_id` column. The `$validation` field is referenced by the function `update_std_csv()`, which uses an iterative algorithm to infer correct values and make automatic updates.

The update logic first identifies all cells in the `$validation` table that have a consistent integer `diff` value across all of their relationships. (Non-integer values are not handled because the values are all assumed to be counts.) In these cases, updating the cell by that `diff` amount will resolve all relationship discrepancies for that cell. For our example, cell [row 30, column 10] has a `diff` value of 30 for both the relationships it participates in (Figure 14b). The algorithm therefore adds 30 to this cell, changing its value from 813 to 843. It then revalidates the relationships based on any updated values.

It should also be noted that the update logic also drops rows from the `$validate` table that share the same `relate_id.` These are cases where multiple members of a set of parent and child cells have been flagged as candidate error cells. The `diff` values from a relationship containing multiple errors are unreliable. By dropping these rows from the `$validate` table, the algorithm can then use more reliable `diff` values from the other relationships the cells participate in to inform inferences.

In our example, after updating cell [row 30, column 10], the geographic relationship between Leinster and its child counties in column 10 becomes valid. This means that cell [row 26, column 10] is no longer considered a potential error cell because it participates in this valid relationship. The `$validation` table after revalidation therefore includes only the rows for cell [row 26, column 8]. Since its `diff` value (55,887.06) is not an integer, it is not automatically updated and is left as a validation error for manual inspection and correction. If any cells with consistent integer `diff` values remained at this point, they would be updated, and another validation iteration would be conducted.

Figure 14a. The '$validate' table for our example table section. Each row represents a combination of a candidate error cell and a relationship. Columns provide row and column indices, a parent/child flag, the difference between the sum of the children and the parent value, an indicator for the type of relationship, and a unique relationship identifier (all cells in each specific set of parent and children will have the same identifier).

```
   rows  cols parent      diff   meta_type  relate_id
  <int> <dbl> <lgl>      <dbl>    <chr>      <chr>
1    26     8 FALSE   55887.06    htag      17e8075322cdba5f20af5373a9267a25
2    26     8 TRUE    55887.06    htag      ebee37bfa5ef2bd4696e862cea9ea415
3    26     8 FALSE   55887.06    geog      1804a97b27c9aceca0deb885a9e0aa47
4    26    10 FALSE  -55887.06    htag      ebee37bfa5ef2bd4696e862cea9ea415
5    26    10 FALSE         30    geog      2dfe7a618d60fa36541d9bf090c5a186
6    30    10 FALSE         30    htag      54dfc907fe84fba5fa41291558f1f63a
7    30    10 FALSE         30    geog      2dfe7a618d60fa36541d9bf090c5a186
```

Figure 14b. Same difference values for data cell [row 30, column 10] in different participating relationships.

```
  rows  cols parent  diff meta_type relate_id
  <int> <dbl> <lgl>  <dbl> <chr>     <chr>
1    30    10 FALSE     30 htag      54dfc907fe84fba5fa41291558f1f63a
2    30    10 FALSE     30 geog      2dfe7a618d60fa36541d9bf090c5a186
```

After running `update_std_csv()`, the `ihgis_std_csv` object contains the current status of the relationships along with a listing of cells that were automatically updated (Figure 15). We see that the set of updated cells contains the cleaned data value that was stripped of punctuation marks [row 28, column 10], as well as the cell that was successfully updated [row 30, column 10]. Situations where the tools are not able to make automatic updates are described in the following section. The tools are designed to be conservative; instead of trying to update ambiguous cells, they are left to the user to manually investigate and correct.

Figure 15. Resulting relationship statuses and list of updated cells after calling `update_std_csv()`.

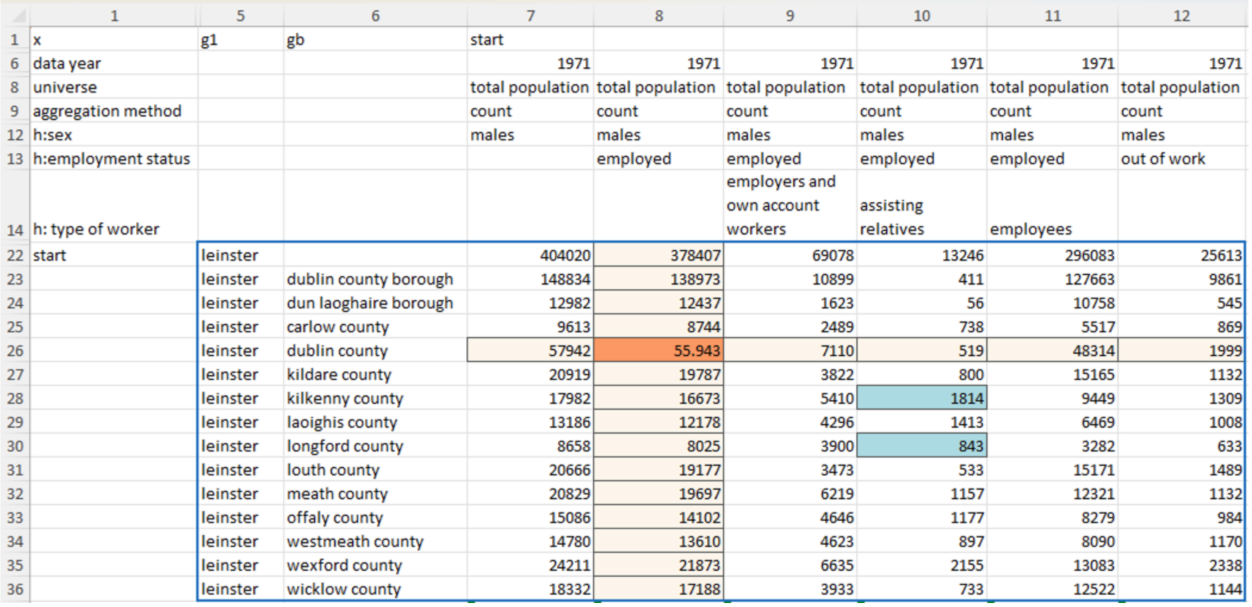
```
--- Column relationships ---
✗ 7 = 8 + 12
✗ 8 = 9 + 10 + 11
✗ 13 = 17 + 18
✓ 17 = 14 + 15 + 16

--- Row relationships ---
✓ 15 = 16 + 22 + 37 + 48
✓ 16 = 17 + 18 + 19 + 20 + 21
✗ 22 = 23 + 24 + 25 + 26 + 27 + 28 + 29 + 30 + 31 + 32 + 33 +
  34 + 35 + 36
✓ 37 = 38 + 39 + 40 + 41 + 42 + 43 + 44 + 45 + 46 + 47
✓ 48 = 49 + 50 + 51

--- Updated Cells ---
   rows  cols   diff
2    28    10   1814
3    30    10     30
```

3.5.1 Visual cues for manual review

To facilitate the manual review and corrections, the R package includes a feature to write out an updated standard file in .xlsx format with additional formatting. Cells highlighted in dark orange are remaining candidate error cells, cells highlighted in light orange are involved in a relationship with a candidate error cell, but are not themselves candidate error cells, and cells highlighted in blue have been automatically updated (Figure 16).

In our example table section, two updated cells are highlighted in blue, the cleaned cell [row 28, column 10] and the corrected value [row 30, column 10]. The non-integer cell remains as a candidate error cell and is highlighted in dark orange. The light orange cells provide a visual cue to indicate the relationships this cell participates in. When multiple candidate error cells remain, the pattern of dark orange and light orange cells can help to diagnose the root of the problem.

Figure 16. An updated standard CSV file with highlighting for remaining errors (orange) and updated values (blue).



	1	5	6	7	8	9	10	11	12
1	x	g1	gb	start					
6	data year			1971	1971	1971	1971	1971	1971
8	universe			total population	total population	total population	total population	total population	total population
9	aggregation method			count	count	count	count	count	count
12	h:sex			males	males	males	males	males	males
13	h:employment status				employed	employed	employed	employed	out of work
14	h: type of worker					employers and own account workers	assisting relatives	employees	
22	start	leinster		404020	378407	69078	13246	296083	25613
23		leinster	dublin county borough	148834	138973	10899	411	127663	9861
24		leinster	dun laoghaire borough	12982	12437	1623	56	10758	545
25		leinster	carlow county	9613	8744	2489	738	5517	869
26		leinster	dublin county	57942	55.943	7110	519	48314	1999
27		leinster	kildare county	20919	19787	3822	800	15165	1132
28		leinster	kilkenny county	17982	16673	5410	1814	9449	1309
29		leinster	laoighis county	13186	12178	4296	1413	6469	1008
30		leinster	longford county	8658	8025	3900	843	3282	633
31		leinster	louth county	20666	19177	3473	533	15171	1489
32		leinster	meath county	20829	19697	6219	1157	12321	1132
33		leinster	offaly county	15086	14102	4646	1177	8279	984
34		leinster	westmeath county	14780	13610	4623	897	8090	1170
35		leinster	wexford county	24211	21873	6635	2155	13083	2338
36		leinster	wicklow county	18332	17188	3933	733	12522	1144

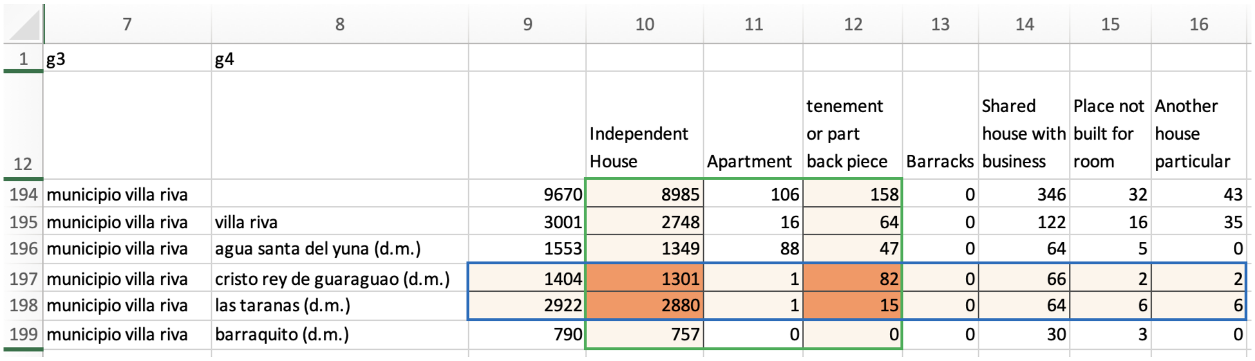
3.6. When Automated Updates Are Not Possible

In some instances, the QA/QC tools do not have sufficient information or are otherwise unable to make automatic updates to error cells, as we have seen with the decimal value in our Ireland 1971 example. Other situations in which automated updates are not possible include non-unique solutions, errors that cancel each other out, and cases where inferred relationships do not behave as expected.

3.6.1. Non-unique solutions

When intersecting relationships contain multiple errors, it is possible that modifying distinct sets of candidate error cells could make the relationships valid. For example, Figure 17 has two intersecting relationships: the sum of columns 10 through 16 should equal the total in column 9 (blue outline), and the sum of rows 195 through 199 should equal the total in row 194 (green outline). In row 197, the child columns sum to 1,454, which is 50 greater than the given total of 1404. Similarly, in row 198, the sum is 2974, which is 50 greater than 2,922. Likewise, the row relationships in columns 10 and 12 are both off by 50 (sum 9,035 versus total 8,985 in column 10 and sum 208 versus total 158 in column 12). The tools are able to identify the four cells highlighted in orange at the intersection of these relationships as the likely locations of OCR errors. Updating one cell in each intersecting row and column by 50 would make all the relationships valid. But it is not clear whether the updates should be made to cells [row 197, column 10] and [row 198, column 12] (upper left/lower right) or to cells [row 197, column 12] and [row 198, column 10] (lower left/upper right). The tools cannot resolve these two possible solutions, so they are left to the user to review against the original source document.

Figure 17. Standard CSV file with remaining errors that have non-unique solutions.



	7	8	9	10	11	12	13	14	15	16
1	g3	g4								
12				Independent House	Apartment	tenement or part back piece	Barracks	Shared house with business	Place not built for room	Another house particular
194	municipio villa riva		9670	8985	106	158	0	346	32	43
195	municipio villa riva	villa riva	3001	2748	16	64	0	122	16	35
196	municipio villa riva	agua santa del yuna (d.m.)	1553	1349	88	47	0	64	5	0
197	municipio villa riva	cristo rey de guaraguao (d.m.)	1404	1301	1	82	0	66	2	2
198	municipio villa riva	las taranas (d.m.)	2922	2880	1	15	0	64	6	6
199	municipio villa riva	barraquito (d.m.)	790	757	0	0	0	30	3	0

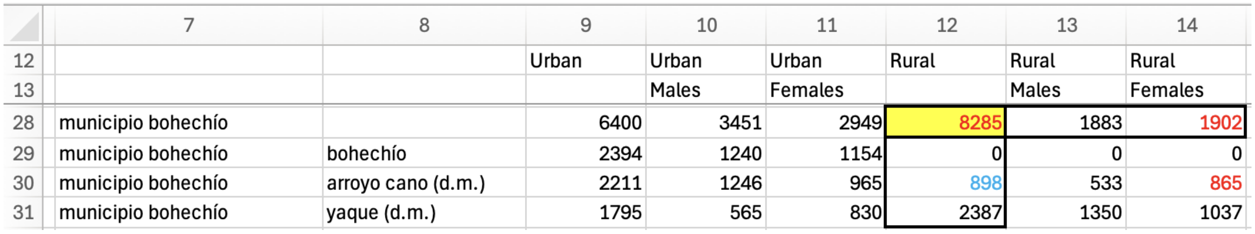
3.6.2. Canceled errors

In some cases, multiple errors may cancel each other out, making relationships appear valid even when they contain errors. The standard CSV shown in Figure 18a contains errors in [row 28, column 14] with the incorrect value 1902, [row 30, column 14] with the incorrect value of 865, and [row 28, column 12] with the incorrect value 8285. These values differ from the source document, which has the values 1402, 365, and 3285 in these locations (Figure 18b). In column 14, the parent total value 1902 and one of its child values, 865, are both off by 500. These matching errors in the parent and child values make the relationship appear valid.

The fact that this relationship appears valid complicates the other relationships these cells participate in. Naturally, those other relationships will not add up correctly due to these errors. But because these cells participate in a relationship that appears valid, they are dropped from consideration as error cells.

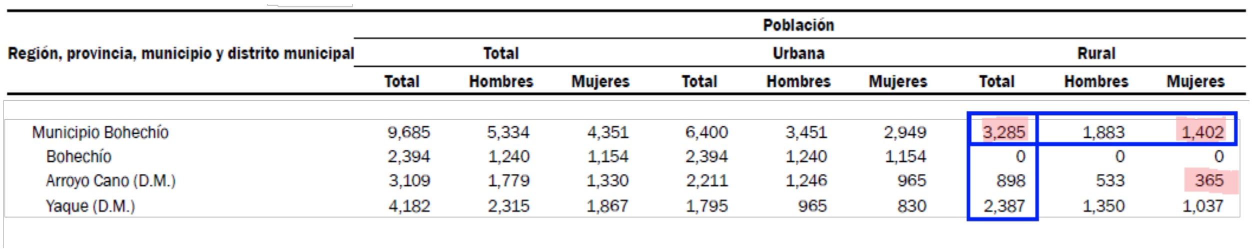
In this example cell [row 30, column 12], which had the correct value of 898, ends up incorrectly updated to 1398, which is 500 more than its original value (Figure 18c). This cell participates in the rural males + rural females = rural relationship with the error cell [row 30, column 14]. It also participates in a geog relationship with another error cell [row 28, column 12]. The `$validate` table includes [row 28, column 12], but does not include [row 30, column 14] because it participates in the seemingly valid relationship in column 14. Because [row 28, column 12] and [row 30, column 12] are in the same set of parent-child cells, the `diff` values from that relationship are dropped from the `$validate` table as unreliable. This leaves [row 30, column 12] with just one `diff` value, the 500 from the invalid relationship on row 30. This cell is then updated, apparently correcting the relationship on row 30. Cell [row 28, column 12] is left as an unresolved candidate error cell due to conflicting `diff` values between its row and column relationships (Figure 18b). Reviewing the highlighted cells against the source document should help the user realize that the wrong cell was updated and identify the correct values.

Figure 18a. Input Standard CSV file with errors.



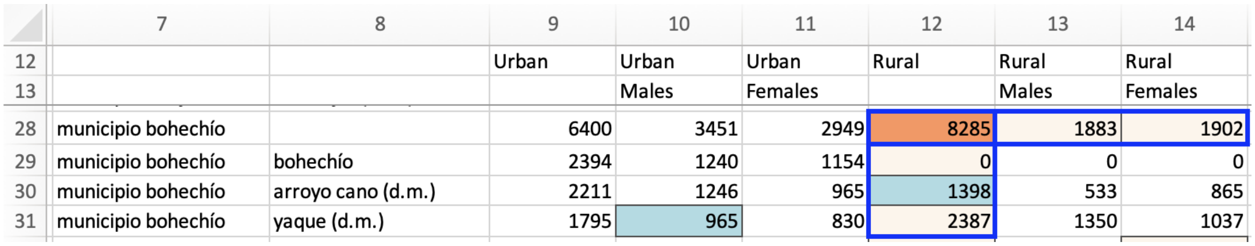
	7	8	9	10	11	12	13	14
12			Urban	Urban	Urban	Rural	Rural	Rural
13				Males	Females		Males	Females
28	municipio bohechío		6400	3451	2949	8285	1883	1902
29	municipio bohechío	bohechío	2394	1240	1154	0	0	0
30	municipio bohechío	arroyo cano (d.m.)	2211	1246	965	898	533	865
31	municipio bohechío	yaque (d.m.)	1795	565	830	2387	1350	1037

Figure 18b. Source document with correct values.



Región, provincia, municipio y distrito municipal	Población								
	Total			Urbana			Rural		
	Total	Hombres	Mujeres	Total	Hombres	Mujeres	Total	Hombres	Mujeres
Municipio Bohechío	9,685	5,334	4,351	6,400	3,451	2,949	3,285	1,883	1,402
Bohechío	2,394	1,240	1,154	2,394	1,240	1,154	0	0	0
Arroyo Cano (D.M.)	3,109	1,779	1,330	2,211	1,246	965	898	533	365
Yaque (D.M.)	4,182	2,315	1,867	1,795	965	830	2,387	1,350	1,037

Figure 18c. Updated Standard CSV file with remaining errors that cancel each other out.



	7	8	9	10	11	12	13	14
12			Urban	Urban	Urban	Rural	Rural	Rural
13				Males	Females		Males	Females
28	municipio bohechío		6400	3451	2949	8285	1883	1902
29	municipio bohechío	bohechío	2394	1240	1154	0	0	0
30	municipio bohechío	arroyo cano (d.m.)	2211	1246	965	1398	533	865
31	municipio bohechío	yaque (d.m.)	1795	965	830	2387	1350	1037

3.6.3. Structural issues

In some cases every cell in a given relationship is flagged as an error. This usually results from the following scenarios:

(1) Files that contain only geographic or h-tag relationships, but not both. Without intersecting relationships, the tools are not able to eliminate cells participating in valid relationships from consideration as potential errors leaving all the cells in an invalid relationship under consideration as potential error cells.

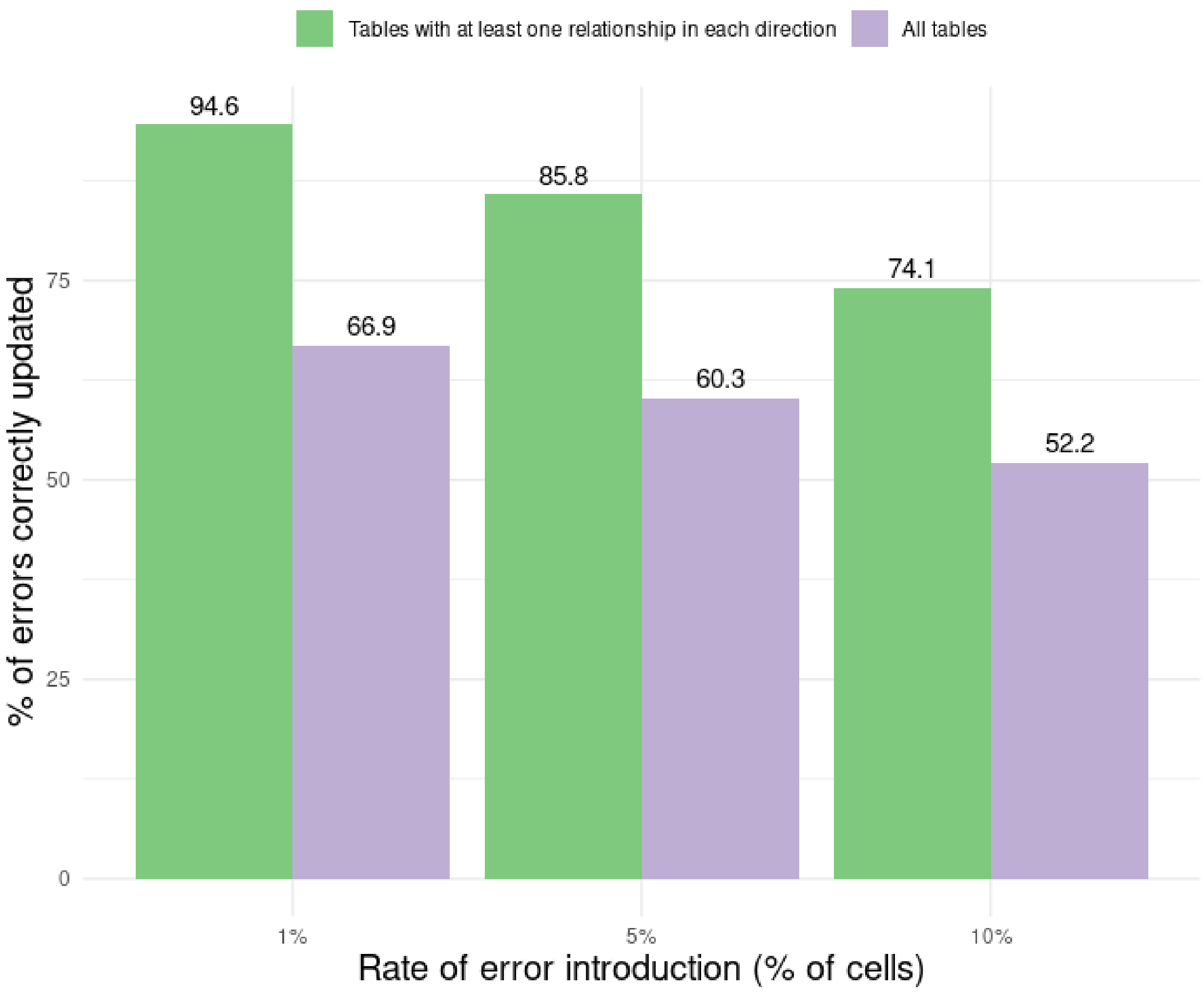
(2) Incorrect markup leading to improperly structured metadata in the standard CSV input file. Markup errors, such as blanking out non-total cells or failing to separate sets of categories into different h-tags, can cause the tools to infer relationships incorrectly. If the tools misidentify a relationship, then it is not surprising that validating that “relationship” would produce nothing but errors. In these cases, the markup stage needs to be revisited to properly identify the relationships.

(3) The structure of the table violates assumptions made in identifying relationships. For example, in a table of population by languages spoken, people may be double-counted if they reported speaking more than one language. Due to this double-counting, the sum across languages may be greater than the total number of people, making the relationship appear invalid. In these cases, no corrections are needed.

4. Performance Metrics

To measure the performance of the automated QA/QC tools, we ran simulations on standard CSV files in which errors were randomly introduced into a known percentage of cells. As expected, the tools performed better on tables with more relationships. Performance was particularly sensitive to the presence or absence of both h-tag and geog relationships. In a fairly typical case where tables have at least one h-tag and one geog relationship and 1% of cells have errors, the tools were able to automatically identify and correct nearly 95% of errors (Figure 19). Performance drops off as more errors are introduced, but even with 10% of cells having errors, tables with at least one relationship in each direction have a 75% success rate of correctly updating the values.

Figure 19. Success rate of correctly updating errors relative to the rate of error introduction.


🟩 Tables with at least one relationship in each direction
🟪 All tables
% of errors correctly updated
Rate of error introduction (% of cells)