

# How to Construct a Quantitative Analysis Table with the open-cbgm

Sum of PERC1	Column Labels																
Row Labels	A	P72	P74	P78	01	02	03	04	018	020	025	044	049	056	0142	0251	0316
A		83.7%	100.0%	70.0%	90.2%	93.4%	96.4%	91.7%	86.9%	91.5%	88.3%	88.5%	88.4%	87.4%	88.8%	88.2%	84.2%
P72	83.7%		83.3%	68.4%	76.1%	81.3%	85.2%	78.4%	77.2%	80.4%	73.1%	77.8%	77.7%	77.7%	78.6%	93.8%	73.3%
P74	100.0%	83.3%		0.0%	83.3%	83.3%	83.3%	100.0%	100.0%	83.3%	100.0%	83.3%	100.0%	100.0%	100.0%	0.0%	0.0%
P78	70.0%	68.4%	0.0%		68.4%	70.0%	70.0%	65.0%	70.0%	70.0%	80.0%	60.0%	70.0%	70.0%	70.0%	70.0%	0.0%
01	90.2%	76.1%	83.3%	68.4%		84.7%	88.4%	84.7%	81.3%	85.0%	81.4%	87.8%	81.3%	81.9%	82.7%	94.1%	78.9%
02	93.4%	81.3%	83.3%	70.0%	84.7%		92.7%	88.7%	85.2%	87.2%	86.0%	86.8%	85.6%	85.2%	86.5%	88.2%	86.1%
03	96.4%	85.2%	83.3%	70.0%	88.4%	92.7%		90.4%	86.7%	90.8%	87.3%	91.0%	87.2%	87.2%	88.6%	88.2%	81.6%
04	91.7%	78.4%	100.0%	65.0%	84.7%	88.7%	90.4%		85.0%	87.2%	86.5%	84.6%	85.6%	83.3%	84.3%	88.2%	73.0%
018	86.9%	77.2%	100.0%	70.0%	81.3%	85.2%	86.7%	85.0%		90.5%	92.2%	84.4%	96.5%	92.0%	92.3%	100.0%	65.8%
020	91.5%	80.4%	83.3%	70.0%	85.0%	87.2%	90.8%	87.2%	90.5%		92.2%	87.5%	93.4%	90.5%	91.3%	100.0%	78.9%
025	88.3%	73.1%	100.0%	80.0%	81.4%	86.0%	87.3%	86.5%	92.2%	92.2%		83.8%	97.1%	94.2%	94.2%	100.0%	71.1%
044	88.5%	77.8%	83.3%	60.0%	87.8%	86.8%	91.0%	84.6%	84.4%	87.5%	83.8%		84.8%	83.9%	84.3%	87.5%	80.6%
049	88.4%	77.7%	100.0%	70.0%	81.3%	85.6%	87.2%	85.6%	96.5%	93.4%	97.1%	84.8%		90.9%	91.8%	100.0%	65.8%
056	87.4%	77.7%	100.0%	70.0%	81.9%	85.2%	87.2%	83.3%	92.0%	90.5%	94.2%	83.9%	90.9%		99.5%	100.0%	73.7%
0142	88.8%	78.6%	100.0%	70.0%	82.7%	86.5%	88.6%	84.3%	92.3%	91.3%	94.2%	84.3%	91.8%	99.5%		100.0%	73.7%
0251	88.2%	93.8%	0.0%	70.0%	94.1%	88.2%	88.2%	88.2%	100.0%	100.0%	100.0%	87.5%	100.0%	100.0%	100.0%		0.0%
0316	84.2%	73.3%	0.0%	0.0%	78.9%	86.1%	81.6%	73.0%	65.8%	78.9%	71.1%	80.6%	65.8%	73.7%	73.7%	0.0%	

a more-or-less painless procedure

Nicholas Lammé  
2023

## The Problem

There is no obvious way to generate a quantitative analysis table from the open-cbgm, developed by Joey McCollum,<sup>1</sup> except for manually comparing witnesses and recording the data one witness at a time. Even if you're only working with a handful of witnesses, this can still be a very time-consuming task. This seems like an obvious function that the software should be able to perform because the CBGM is already comparing all witnesses to all others at all places of variation.

To do this by hand, you would loop through your witnesses using the **compare\_witnesses** function, and record their pregenealogical relationship to every other witnesses in a table. For example, `./compare_witnesses cache.db 01` will return the following. This is the first 7 closes witnesses to 01 for Jude.

Genealogical comparisons for W1 = 01 (180 extant passages):

W2	DIR	NR	PASS	EQ		W1>W2	W1<W2	NOREL	UNCL	EXPL	COST
A	>	1	180	161	( 89.444%)	0	16	0	3	177	17.000
NA28	>	1	180	161	( 89.444%)	1	14	0	4	175	15.000
81	>	2	179	159	( 88.827%)	1	13	2	4	172	14.000
326	>	3	178	157	( 88.202%)	4	13	2	2	170	14.000
2805	>	3	180	157	( 87.222%)	6	9	3	5	166	10.000
436	>	4	180	156	( 86.667%)	6	13	1	4	169	14.000
03	>	5	177	155	( 87.571%)	2	15	1	4	170	16.000

Likewise, there are other tables that can be constructed like a find relatives table, but must also be done by hand. Because there is no way currently for the open-cbgm (or any implementation of the CBGM that I am aware of) to do this, the problem can be solved by writing a script to do the work for us. I freely admit that I am no programmer. The solution here is offered in the hope that 1) it will be useful to others trying to do the same, and 2) someone will genuine skill in programming will come along and improve on the concept.

## My Solution

Since the manual procedure involves looping through elements (witnesses) in an array (the total witness set), I realized that the computer could do it for me. I wrote a

---

1. <https://github.com/jjmccollum/open-cbgm-standalone>. The GitHub contains very detailed instructions for using the software. There is also Joey's illustrated crash course, [https://www.academia.edu/43490548/The\\_CBGM\\_An\\_Illustrated\\_Crash\\_Course\\_Supplement\\_to\\_The\\_open\\_cbgm\\_Library\\_Design\\_and\\_Demonstration\\_](https://www.academia.edu/43490548/The_CBGM_An_Illustrated_Crash_Course_Supplement_to_The_open_cbgm_Library_Design_and_Demonstration_).

script in bash for that purpose.<sup>2</sup> Again, the script is kind of primitive and does not generate the QA table all at once. Further steps must be taken to construct the final table. The script does most of the heavy lifting, however.

```
$ qaTableStarter.sh /Users/nicholaslamme/Public/open-cbgm-standalone/build/bin/qaTableStarter.sh
#!/bin/bash

A=(A P72 P74 P78 01 02 03 04 018 020 025 044 049 056 0142 0251 0316)

B=(A P72 P74 P78 01 02 03 04 018 020 025 044 049 056 0142 0251 0316)

command=./compare_witnesses

database=cacheECM.db

for item1 in "${A[@]"; do
    for item2 in "${B[@]"; do
        if [[ $item1 == "$item2" ]]; then
            continue
            break
        fi
        $command -f csv -o /Users/nicholaslamme/Public/open-cbgm-standalone/build/bin/
qaTable/$item1-$item2.csv $database $item1 $item2
    done
done

cd /Users/nicholaslamme/Public/open-cbgm-standalone/build/bin/qaTable/

for f in *.csv
do
    sed -e "s/^/${f},/" $f >> combined.csv
done
```

The script relies on two identical arrays through which it loops comparing all witnesses against all others with the open-cbgm. It creates a file for each comparison and when it finishes, the script writes the file name to the first column of each .csv file and combines all the files into one master file. The .csv files are comma-delimited.

The user needs to replace the "database" variable with their own database and

---

2. You can download the script here: <https://github.com/dopeyduck/qa-table-starter>

supply their own witnesses to the arrays. Also, the user will need to specify their own directory on their system for the files. Lastly, the script must be run from inside the same directory as the open-cbgm.

The following is the process I use to create the QA table with this script.

## The Procedure

*Step 1: Modify the script with your information and run it.*

From the command line (I'm in a Mac), run the following command:

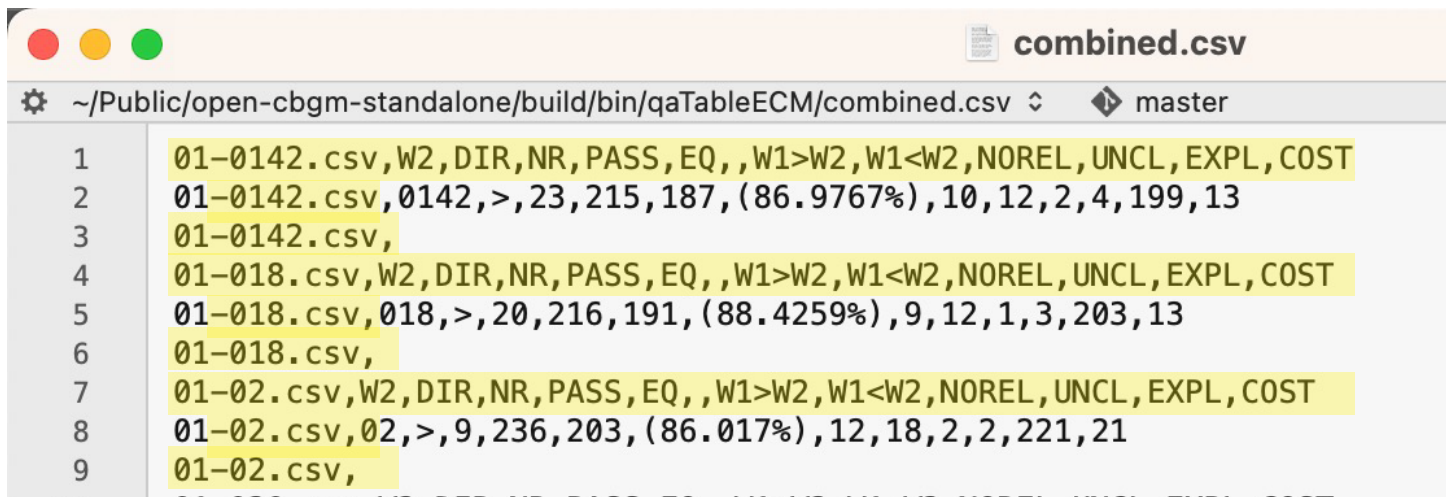
```
sh qaTableStarter.sh
```

*Step 2: Locate the "combined.csv" file in the directory you specified in the script.*



Notice the naming convention of each file: W1-W2.csv (witness1-witness2.csv). The W1 is the one being compared to all others. This convention was chosen to avoid the script overwriting the same files every time it is looped through the arrays. There's probably a better way to achieve what I'm after, but this is how I was able to do it. This file name is written to each file as the first column entry. We will need to edit out the "-W2.csv" in the next steps.

Step 3: Open the "combined.csv" file in a text editor (like BBEdit or Visual Studio Code).



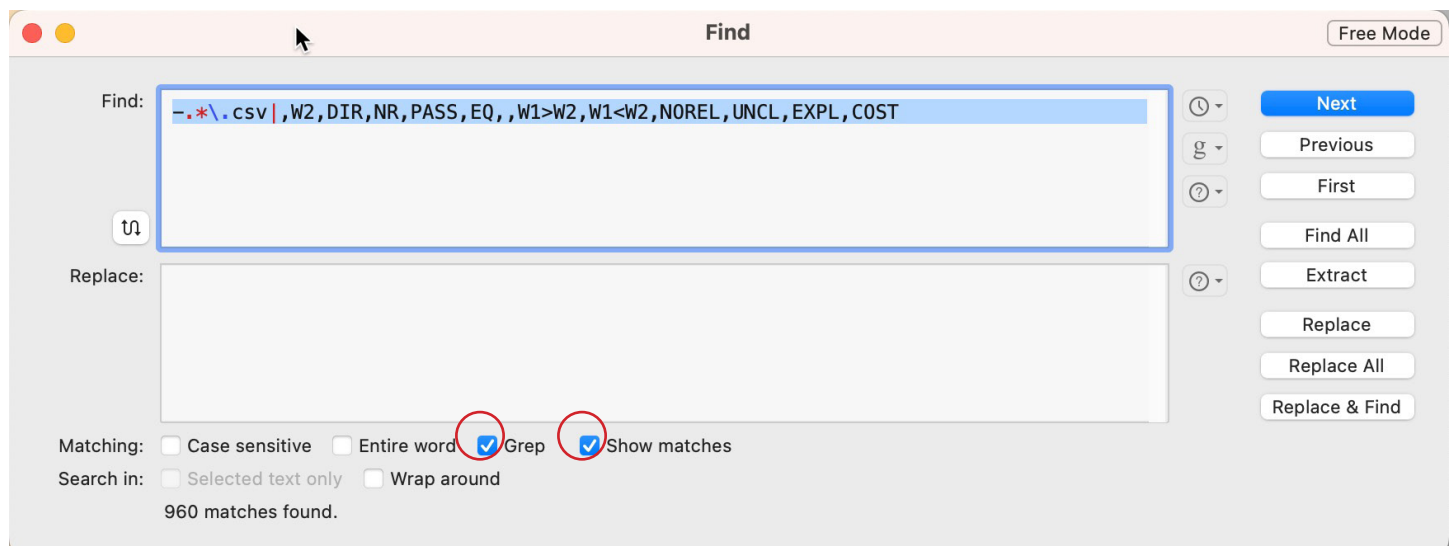
What we are trying to do is clean up the file by eliminating unnecessary information that I wasn't sure how to strip out with the script. This step only requires two steps.

Step 4: Clean the file by executing find/replace searches using general regular expressions.

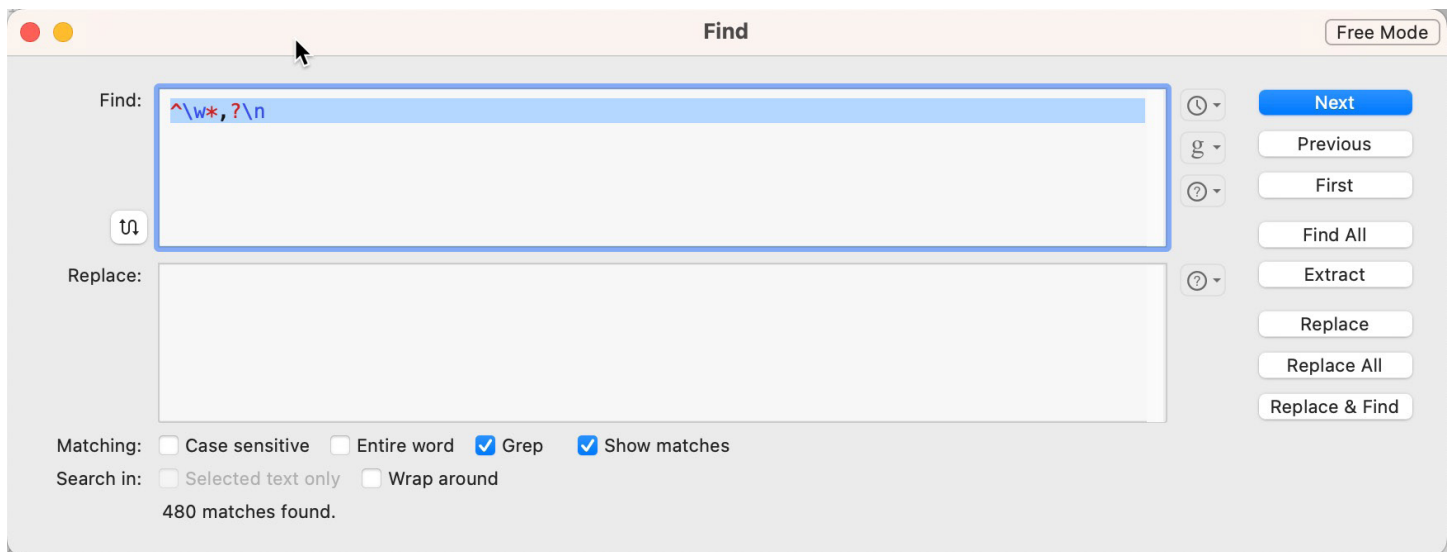
I run the following searches in order and replace with nothing.

`-.*\.csv|,W2,DIR,NR,PASS,EQ,,W1>W2,W1<W2,NOREL,UNCL,EXPL,COST`

`^\w*,?\n`



```
01-0142.csv,W2,DIR,NR,PASS,EQ,,W1>W2,W1<W2,NOREL,UNCL,EXPL,COST
01-0142.csv,0142,>,23,215,187,(86.9767%),10,12,2,4,199,13
01-0142.csv,
```



```
01
01,0142,>,23,215,187,(86.9767%),10,12,2,4,199,13
01,
01
```

The final .csv file should look like this.

```
01,0142,>,23,215,187,(86.9767%),10,12,2,4,199,13
01,018,>,20,216,191,(88.4259%),9,12,1,3,203,13
01,02,>,9,236,203,(86.017%),12,18,2,2,221,21
01,020,>,13,220,199,(90.4545%),4,14,1,2,213,16
01,025,>,28,96,86,(89.5833%),2,5,1,2,91,5
01,0251,=,18,18,(100%),0,0,0,0,18,
01,03,>,6,236,206,(87.2881%),9,18,1,2,224,21
01,0316,<,44,36,(81.8182%),6,1,0,1,37,
01,04,<,215,186,(86.5116%),13,12,2,3,198,
```

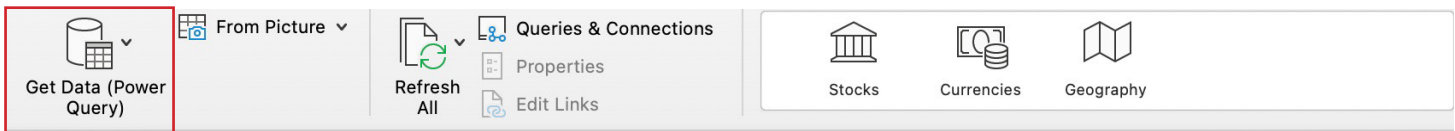
Save the file.

Each entry starts with W1, followed by the witness with which it is compared (W2), its direction (DIR), rank (NR), passages extant (PASS), passages at which W1 and W2 are equal (EQ), the percent agreement (PERC1), and other genealogical information provided by the open-cbgm. We will only need three columns for the QA table, W1, W2, and PERC1. We will also have to name these manually in Excel.

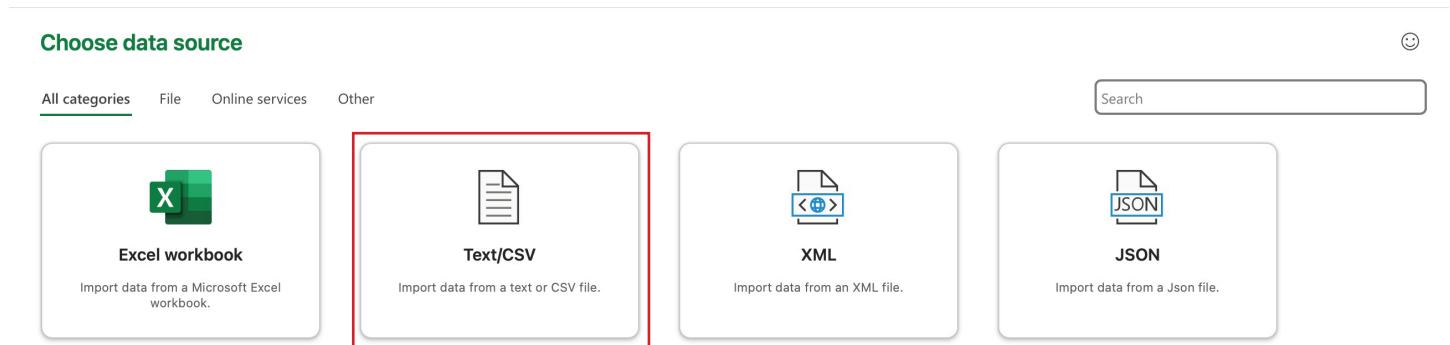
*Step 5: Open a new Excel spreadsheet.*



Step 6: In the Data tab, select "Get Data" (this may vary from Mac to Windows).



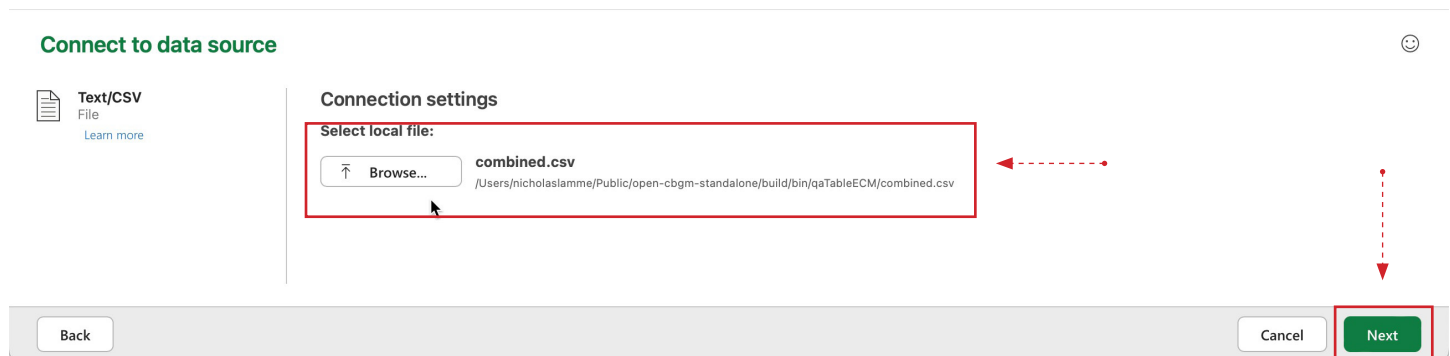
Step 7: Choose Text/CSV file option.



Step 8: Select the combined.csv file and then Next.

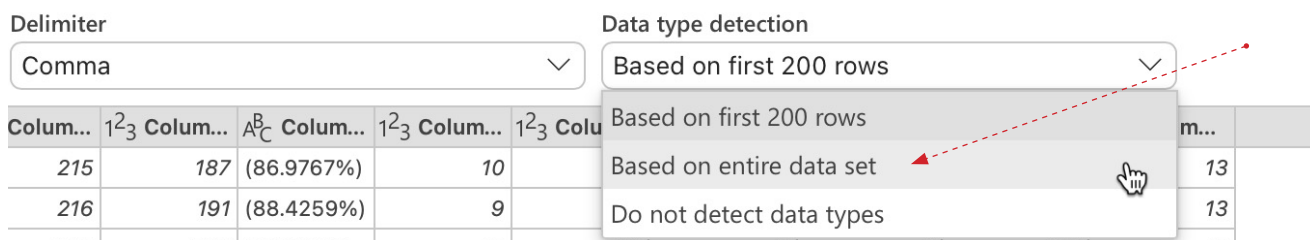
Step 9: Choose the data source.

Step 10: Load the file and continue.



Step 11: Specify data type.

In the Preview Data Field dialogue box, make sure to select the "Based on entire data set" option from the "Data type detection" drop-down menu. Then click "Load".



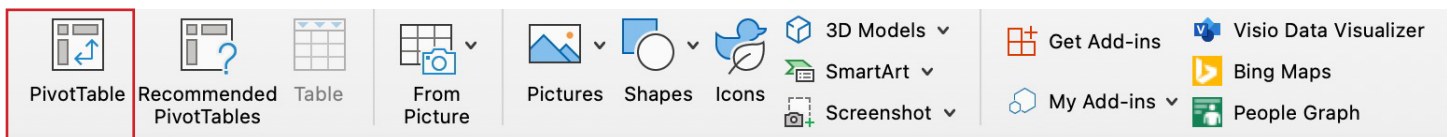
*Step 12: Rename the headers to match the open-cbgm tables.*

W1	W2	DIR	NR	PASS	EQ	PERC1	W1>W2	W1<W2	NOREL	UNCL	EXPL	COST
01	0142	>	23	215	187	(86.9767%)	10	12	2	4	199	13
01	018	>	20	216	191	(88.4259%)	9	12	1	3	203	13

*Step 10: Format the PERC1 column.*

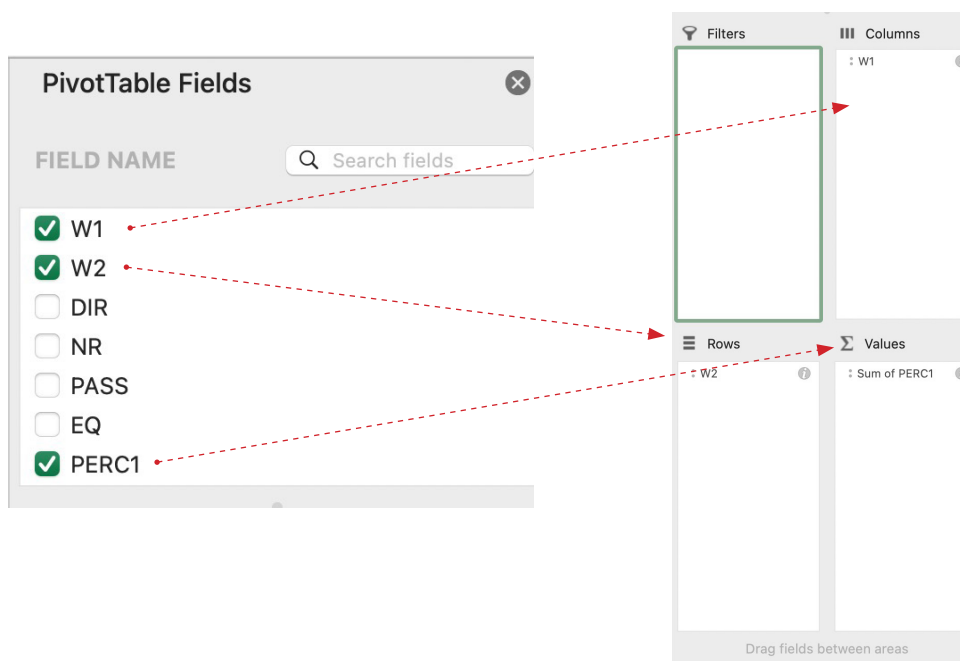
When the table opens, you will notice that the open-cbgm includes percents in parenthesis. Excel interprets these as negative numbers. To fix this, type -1 into any cell outside the table. Then select the cell and copy it (CMD/CTL-C). Next, highlight all the data in the PERC1 column, and change the format to "Number" under the Home tab in the ribbon. Then right-click, navigate to "Special Paste," and then to "Paste Special...". Under "Operations," select "Multiply" and click OK. Then change the style to "Percentage" on the Ribbon and adjust the decimal place as desired.

*Step 11: Under the "Insert" tab, select "Pivot Table," and accept the default values.*



*Step 12: Arrange the information.*

Drag W1 to the columns box, W2 to the Rows box, and PERC1 to the Values box.





### Step 13: Format the final table.

For the final format, you will have to convert the decimals to percentages and center the information in the cells. I also right-justify the first column so it's closer to the rest of the text. Excel will place this in a new tab in the workbook. The percentages can always be expanded to different decimal places.

Here is the final table of our 17 witness sample.

QA Table W2																		
W1		A	P72	P74	P78	01	02	03	04	018	020	025	044	049	0142	0251	0316	Grand Total
A			82%	100%	78%	91%	93%	95%	92%	94%	97%	96%	88%	94%	92%	100%	84%	92%
P72	82%			80%	70%	77%	80%	81%	79%	80%	81%	73%	76%	81%	81%	83%	55%	77%
P74	100%	80%			100%	100%	83%	83%	80%	100%	100%	100%	83%	100%	80%	100%	100%	93%
P78	78%	70%	100%			74%	74%	74%	73%	78%	78%	100%	65%	78%	74%	82%	100%	80%
01	91%	77%	100%	74%			86%	87%	87%	88%	90%	90%	86%	89%	87%	100%	82%	88%
02	93%	80%	83%	74%	86%			91%	89%	90%	92%	92%	84%	91%	89%	100%	82%	88%
03	95%	81%	83%	74%	87%	91%			89%	91%	94%	94%	88%	91%	90%	94%	80%	88%
04	92%	79%	80%	73%	87%	89%	89%			93%	93%	96%	85%	93%	90%	94%	73%	87%
018	94%	80%	100%	78%	88%	90%	91%	93%			95%	94%	87%	96%	93%	94%	84%	90%
020	97%	81%	100%	78%	90%	92%	94%	93%	95%			97%	89%	96%	93%	94%	86%	92%
025	96%	73%	100%	100%	90%	92%	94%	96%	94%	97%			86%	96%	93%	100%	88%	93%
044	88%	76%	83%	65%	86%	84%	88%	85%	87%	89%	86%			87%	86%	72%	75%	83%
049	94%	81%	100%	78%	89%	91%	91%	93%	96%	96%	96%	87%			92%	94%	84%	91%
0142	92%	81%	80%	74%	87%	89%	90%	90%	93%	93%	93%	86%	92%			88%	85%	87%
0251	100%	83%	100%	82%	100%	100%	94%	94%	94%	94%	94%	100%	72%	94%	88%		100%	93%
0316	84%	55%	100%	100%	82%	82%	80%	73%	84%	86%	88%	75%	84%	85%	100%			84%
Grand Total	92%	77%	93%	80%	88%	88%	88%	87%	90%	92%	93%	83%	91%	87%	93%	84%		88%