Problem: Multiple .csv files of data were generated by the software for a commonly used lab test. Due to the nature of the test, "blank" or "compliance" data must first be gathered without a sample. This compliance data is then eventually used to correct the "compression" data gathered from actually testing the sample.

The analysis preparation process involves copying and pasting data from multiple files into one Excel sheet, and then utilizing the VLOOKUP() function in Excel to locate Load values in the Compliance data which correspond to Load values in the Compression data, and then return the Compliance Crosshead, as shown in Fig. 1 below. This process is necessary for every data set needing to be analyzed and involves an unnecessary amount of effort. The VLOOKUP() function and input cannot be reasonably standardized due to differing sizes of data sets, and the function's dependence on how the analyst chooses to arrange the different sets.

Solution: A script and web application was written in R and R Shiny, respectively, that utilized basic parsing principles (primarily via the *findInterval()* function) to extract and compile the useful data, and then produce the desired calculations on the data set.

Example Usage: The user first uploads the Compliance .csv file via the uppermost button, then the Compression .csv file via the button directly below, as shown in Fig. 2. The user then clicks "Correct crosshead data" to generate a preview of the resulting output, as shown in Fig. 3. Once satisfied, the user then can download the results as a new .csv via the "Download…" button.

Multiple Compression data sets can be corrected using the same Compliance file, which is held in memory until changed; the user needs only to upload the new Compression .csv file and "Correct crosshead data" again.

Benefit: A .csv file is produced as output. The .csv file, shown as Fig. 4, represents the important data and calculations for all samples in a consolidated format. The resulting output eliminated the need for repetitive copying and pasting by lab staff, significantly reducing both the time spent and risk of errors in data preparation for lab reports.

Figure 1: Previous standard method of preparing data for analysis: Compliance and Compression data have been merged into the same Excel worksheet via copy-paste. The "CompCross" column utilizes the VLOOKUP() function (encircled); the output of the function is then used to calculate the "CorrCross" column.
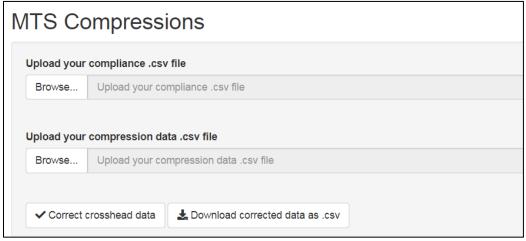

Figure 2: User interface of application. User uploads two .csv files, a Compliance data file and Compression data file. Once both files are uploaded, the user presses the "Correct crosshead data" button.

MTS Compressions

Upload your compliance .csv file

| Browse... | compliance_edit.csv |

Upload complete

Upload your compression data .csv file

| Browse... | compression1_edit.csv |

Upload complete

✓ Correct crosshead data   ⬇ Download corrected data as .csv

Show 100 ▾ entries   Search: [   ]

	Load ⇕	Crosshead ⇕	CompCross ⇕	CorrectedCross ⇕
1	-0.166	19.9803	0.3386	19.6417
2	0.476	19.9134	-0.0039	19.9173
3	12.987	19.5433	-0.0945	19.6378
4	14.98	19.1575	-0.1063	19.2638

Figure 3: After the "Correct…" button is pressed, a preview of the data is shown below the user interface. When satisfied, the user is able to "Download corrected data as .csv".

	A	B	C	D	E
1		Load	Crosshead	CompCross	CorrectedCross
2	1	-0.166	19.9803	0.3386	19.6417
3	2	0.476	19.9134	-0.0039	19.9173
4	3	12.987	19.5433	-0.0945	19.6378
5	4	14.98	19.1575	-0.1063	19.2638
6	5	12.5	18.8425	-0.0906	18.9331
7	6	12.768	18.5236	-0.0945	18.6181
8	7	14.098	18.189	-0.1024	18.2914
9	8	14.813	17.8465	-0.1024	17.9489
10	9	14.708	17.5158	-0.1024	17.6182
11	10	15.382	17.1772	-0.1063	17.2835
12	11	16.533	16.8347	-0.1102	16.9449
13	12	16.626	16.4961	-0.1102	16.6063
14	13	17.465	16.1614	-0.1142	16.2756
15	14	18.129	15.8228	-0.1181	15.9409
16	15	18.939	15.4921	-0.122	15.6141
17	16	19.607	15.1536	-0.1181	15.2717
18	17	20.472	14.8228	-0.1181	14.9409
19	18	21.416	14.4843	-0.1181	14.6024
20	19	22.332	14.1575	-0.1181	14.2756
21	20	23.332	13.8189	-0.1339	13.9528
22	21	24.69	13.4882	-0.1417	13.6299
23	22	26.127	13.1457	-0.1457	13.2914

data-2019-06-10   ⊕

Figure 4: Output .csv in Excel which allowed readable record-keeping. Data includes primary calculation of interest, the "CorrectedCross" column.