

1. The following issues were identified in the variable appraisal modules

- a. The stratified variable appraisal module does not include the sum of the sample. Moreover, both the stratified variable appraisal module and the unrestricted variable appraisal modules do not include the file name.

(Fixed) Added Sum to all output windows

Std. Err. Total: 682,478,157,833
Point Estimate: 684,205,112,809
Total Sum: 3,452,094,415.92
Confidence Levels
0% 90% 95%

(Fixed) Added file output to all saved files

Lower: 0
Upper: 0
Precision Amount: 0
Precision Percent: 0.00%
t-Value: 1.533206274059
Text File: /home/cbtek/dev/bin/TeamCBTek/debug/ssrnOutput.txt

- b. There is still an issue with the precision being reported as extremely large when the sample results are zero for all items. The issue actually occurs for a range of different sample sizes though not in all cases. It may be that a check for zero against the point estimate is not catching the fact that the point estimate is zero due to the use of floating point. If you decide to set the precision to zero when the point estimate is very small (rather than zero exactly), then please note this in the file explaining the differences from RAT-STATS.

(Unverified Fix) We have done our best to attempt to reproduce this issue but to no avail. We did however notice that point-estimate was being compared to zero and changed the code to this:

```
summary.precisionPercent80 = 0.;  
summary.precisionPercent90 = 0.;  
summary.precisionPercent95 = 0.;  
  
if (m_summaryPointEstimate(index) > 0)  
{  
    summary.precisionPercent80 = (m_summaryPrecisionAmount80(index) / m_summaryPointEstimate(index));  
    summary.precisionPercent90 = (m_summaryPrecisionAmount90(index) / m_summaryPointEstimate(index));  
    summary.precisionPercent95 = (m_summaryPrecisionAmount95(index) / m_summaryPointEstimate(index));  
}
```

This is similar to how the VBA code handles this situation. Set the precision to 0 first and then only change it if point-estimate is greater than zero. We are hoping this solves the issue for whatever dataset was used.

2. The following issues were identified with the simple random sample module

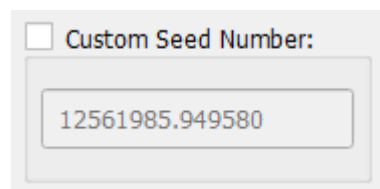
- a. Unable to pull sample when the sample size equals the number of records in the frame.

(Fixed) Changed Highnumber to be inclusive in validator:

```
m_conditionLogger.addError(((high - low) + 1) < (order + spare),  
    "The sampling frame is less than the total number of records")
```

- b. The seed is not always being output correctly. For example, a seed of 1100 was output as 11 (though it appears that the seed input by the user was actually used).

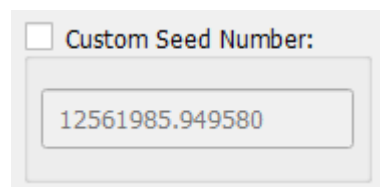
(Fixed) Seed box was converted to text box. Whatever value that appears in the seed box will be identical to what appears in the output.



A screenshot of a web form with a checkbox labeled "Custom Seed Number:". Below the checkbox is a text input field containing the value "12561985.949580".

- c. When user enters seed, the number is changed (perhaps because floating point does not allow for exact output up to 12 decimal places). As a point of clarification, the issue with the previous code was that the software changed the seed input by the user without alerting the user. The software can place reasonable restrictions on the seed as long as the user is aware of them, the differences are documented, and the software can handle seeds output by RAT-STATS.

(Fixed) Seed box was converted to text box. Whatever the user enters will not be automatically converted/changed.



A screenshot of a web form with a checkbox labeled "Custom Seed Number:". Below the checkbox is a text input field containing the value "12561985.949580".

Below are a few additional differences from RAT-STATS that do **not** constitute errors given the specification documents and the referenced RAT-STATS User Manual and Companion Manual:

1. When importing data as .xls or .txt zeros are shown as blank cell. The import still works which is why this isn't considered an error.

(Fixed) This was caused by the “Random Numbers Seed Label” issue above and has been subsequently fixed. A function for removing trailing zeros was removing them from the left side of the decimal causing all the zeros to disappear from the table and also causing 11000 to appear as 11 in the issue above.

Preview Table:

A	
1	0.0000000001
2	-0.32
3	-3,452,100,000
4	-3,450
5	-0.0000000001
6	-1,200
7	-14.4
8	-920
9	0
10	0
11	0
12	0

2. There is no clear indicator of program focus when the software is on the output window. This is not considered an error because the full contents of the output are available in the output file and the program focus is clear otherwise.

(Note) Focus will be on the “Execute” button after program output is shown. Pressing <TAB> will highlight the next button after “Execute”

3. Microsoft narrator reads the contents of the number boxes as zero even after they have been changed. This is not considered an error because the team provided a work-around through the use of alternate screen reading software that is able to read the numbers after they have changed.

(Fixed) All number boxes have been converted over to text-boxes and Microsoft narrator now reads the numbers correctly.