cmstatr: An R Package for Statistical Analysis of Composite Material Data

3/4/2020

Summary

Strength data for composite materials used in aerospace applications, such as carbon fiber and fiberglass reinforced composites, are normally analyzed using statistical methods because of the inherenet variabilities in the constituent materials and in the processing. The design standards for civil aviation require that the probability of structural failure due to these variabilities to be minimized, and to do so, the designer must use what are called "Design Values" for each material in stress analyses and ensure they exceed the actual stresses experienced by those materials in service. Design Values are determined such that, with 95% confidence, they are either the 99% or 90% one-sided lower confidence bound of the material strength, depending on the type of structure. These one-sided tolerance bounds are referred to as A-Basis and B-Basis values, respectively. The statistical methods for calculating these basis values are outlined in Composite Materials Handbook, Volume 1, Revision G, or CMH-17-1G in short (CMH-17 2012). The use of these methods is widely accepted by industry and civil aviation regulators.

Currently, many users use MS Excel spreadsheets to perform these analyses. The MS Excel spreadsheets typically used, such as `STAT-17` (Materials Sciences Corporation 2008), `ASAP` (Raju and Tomblin 2008) and `CMH17-STATS` (Witchita State University and Keshavanarayana 2012), use password-protected `VBA` macros to perform the computations. As such, the code cannot be audited by the user. `cmstatr` is an R package that addresses this issue by implementing the same statistical analysis techniques found in CMH-17-1G in an open-source environment.

Statement of Need

The purpose of `cmstatr` is to:

- Provide a consistent user interface for computing A- and B-Basis values and performing the related diagnostic tests in the `R` programming environment
- Allow auditing of the code used to compute A- and B-Basis values
- Enable users to automate computation workflows or to perform simulation studies

Implementation Goals

`cmstatr` aims give a consistent interface for the user. Most functions are written to work with the `tidyverse` (Wickham et al. 2019) and most functions have similar argument lists. The intent is to make the package easy to learn and use.

The implementation of `cmstatr` also aims to avoid the use of look-up tables that are prevailent in calculation spreadsheets and minimize the use of approximations. While this decision leads to increased computation time, the typically small data sets (tens to hundreds of observations) associated with composite material test data, and the speed of modern computers make this practical for interactive programming.

Example Usage

Normally, to use `cmstatr` the user will load `cmstatr` itself as well as the `tidyverse` package.

```r
library(cmstatr)
library(tidyverse)
```

`cmstatr` contains some example data sets, which can be used to demonstrate the features of the package. One of those data sets — `carbon.fabric.2` — will be used in the following example. This data set contains results from several mechanical tests of a typical composite material, and contains the typical measurements obtained from a test lab. In the following examples, results from tension testing in the warp fiber direction (`WT`) per ASTM D3039 will be used. Part of this data set is shown below.

```r
carbon.fabric.2 %>%
  filter(test == "WT") %>%
  head(10)
```

```
##    test condition batch thickness nplies strength modulus failure_mode
## 1    WT       CTD     A     0.112     14  142.817   9.285          LAT
## 2    WT       CTD     A     0.113     14  135.901   9.133          LAT
## 3    WT       CTD     A     0.113     14  132.511   9.253          LAT
## 4    WT       CTD     A     0.112     14  135.586   9.150          LAB
## 5    WT       CTD     A     0.113     14  125.145   9.270          LAB
## 6    WT       CTD     A     0.113     14  135.203   9.189          LGM
## 7    WT       CTD     A     0.113     14  128.547   9.088          LAB
## 8    WT       CTD     B     0.113     14  127.709   9.199          LGM
## 9    WT       CTD     B     0.113     14  127.074   9.058          LGM
## 10   WT       CTD     B     0.114     14  126.879   9.306          LGM
```

One common task is to calculate B-Basis values. The single-point basis functions perform the following tests: - the maximum normed residual test for outliers within a batch (CMH-17 2012), - the Anderson-Darling k-Sample test to check if batches are drawn from the same (unspecified) distribution (Scholz and Stephens 1987), - the maximum normed residual test for outliers within the data, and - the Anderson-Darling test for a particular probability distribution (Lawless 1982).

Assuming that the data from the warp tension (WT) tested at elevated-temperature/wet condition (ETW) follows a normal distribution, then this can be done using the function for a normal distrisbution:

```r
carbon.fabric.2 %>%
  filter(test == "WT") %>%
  filter(condition == "ETW") %>%
  basis_normal(strength, batch)
```

```
## Warning: `anderson_darling_normal` failed: Anderson-Darling test rejects
## hypothesis that data is drawn from a normal distribution
```

```
##
## Call:
## basis_normal(data = ., x = strength, batch = batch)
##
## Distribution:  Normal    ( n =  18 )
## The following diagnostic tests failed:
##     `anderson_darling_normal`
## B-Basis:   ( p =  0.9 , conf =  0.95 )
## 122.9315
```

All of the various basis functions perform diagnostic tests for each of the statistical tests mentioned above. If any of the diagnostic tests failed, a warning is shown and the test failure is also recorded in the returned object (and shown in that object's `print` method). In the example above, the output shows that the Anderson-Darling test for normality (Lawless 1982) rejects the hypothesis that the data is drawn from a normal distribution.

Two non-parametric basis calculations, based on (Guenther 1969) and (Vangel 1994) are also implemented in `cmstatr`. These functions perform the same diagnostic tests, but omits the Anderson-Darling test for a particular distribution.

The diagnostic test can be run directly using `cmstatr` as well. For example, the failed diagnostic test above can be run as follows:

```
carbon.fabric.2 %>%
  filter(test == "WT") %>%
  filter(condition == "ETW") %>%
  anderson_darling_normal(strength)
```

```
##
## Call:
## anderson_darling_normal(data = ., x = strength)
##
## Distribution:  Normal ( n =  18 )
## Test statistic: A =  0.9381665
## Significance:  0.01103075  (assuming unknown parameters)
## Conclusion: Sample is not drawn from a Normal distribution (alpha =  0.05 )
```

If the failure of a diagnostic test is decided to be acceptable, the test result can be overridden to hide the warning in the basis function output:

```
carbon.fabric.2 %>%
  filter(test == "WT") %>%
  filter(condition == "ETW") %>%
  basis_normal(strength, batch, override = c("anderson_darling_normal"))
```

```
##
## Call:
## basis_normal(data = ., x = strength, batch = batch, override = c("anderson_darling_normal"))
##
## Distribution:  Normal    ( n =  18 )
## The following diagnostic tests were overridden:
##     `anderson_darling_normal`
## B-Basis:   ( p =  0.9 , conf =  0.95 )
## 122.9315
```

`cmstatr` also provides functions for calculating basis values from data pooled across multiple testing environments, as recommended by (CMH-17 2012). These functions calculate a global variance from the data in all tested environmental conditions, then they apply the global variance to the individual mean values of each condition to find the corresponding basis values.

Another common statistical analysis provided by `cmstatr` is for determining whether a sample is drawn from an existing population. This is often used to determine if test data from a second manufacturing site supports the basis values determined from test data generated at the first manufacturing source. The statistical test often recommended for this application generates limits to the mean and minimum individual value (Vangel 2002) that the new sample must exceed. `cmstatr` provides functions for computing the limits based on this test. This statistical test has higher power than some other tests that are available.

For example:

```
carbon.fabric.2 %>%
  filter(test == "WT") %>%
  filter(condition == "RTD")  %>%
  equiv_mean_extremum(strength, n_sample = 8, alpha = 0.05)
```

```
##
```

```
## Call:
## equiv_mean_extremum(df_qual = ., data_qual = strength, n_sample = 8,
##     alpha = 0.05)
##
## For alpha = 0.05 and n = 8
## ( k1 = 2.700045 and k2 = 0.6789966 )
##                    Min Individual      Sample Mean
##      Thresholds:         121.4921         135.0655
```

Validation and Comparison With Existing Tools

Where possible, `cmstatr` has been verified against the examples given in the articles in which the statistical methods were published. Unit tests have been written so that this verification is re-checked routinely to prevent unintended regressions. Agreement between `cmstatr` and the examples in the original articles is within the expected numeric accuracy.

`cmstatr` has also been verified against existing software, such as `STAT-17` (Materials Sciences Corporation 2008), `ASAP` (Raju and Tomblin 2008) and `CMH17-STATS` (Witchita State University and Keshavanarayana 2012) using several example data sets. Agreement between `cmstatr` and the other softwares is generally good, but some results differ slightly, likely due to approximations used in the software. Comparison between `cmstatr` and the other softwares is performed within various unit tests to guard against future regressions.

The tests are automatically run each time a change is made to the code of `cmstatr` using a continuous integration service. Additionally, `CRAN` runs `R CMD check` on each package routinely.

Reproducibility

It is envisioned that many users of `cmstatr` will use it within an R Notebook or a Jupyter Notebook. It is further envisioned that this notebook will be directly converted into the statistical analysis report. If this is done, the reader of the statistical report will be able to verify all of the detailed steps used in the statistical analysis.

Acknowledgement

The author would like to thank Mr. Billy Cheng for his contributions to `cmstatr` and this paper.

References

CMH-17. 2012. "Composites Materials Handbook, Volume 1. Polymer Matrix Composites Guideline for Characterization of Structural Materials." CMH-17-1G. SAE International.

Guenther, William. 1969. *Determination of Sample Size for Distribution-Free Tolerance Limits. Statistical Research Report, Institute of Mathematics, University of Oslo.* 1.

Lawless, J. F. 1982. *Statistical Models and Methods for Lifetime Data.* John Wiley & Sons.

Materials Sciences Corporation. 2008. "CMH-17 Statistical Analysis for B-Basis and a-Basis Values." STAT-17 Rev 5. Materials Sciences Corporation.

Raju, K.S., and J.S. Tomblin. 2008. "Agate Statistical Analysis Program." ASAP-2008. Wichita State University.

Scholz, F.W., and M.A Stephens. 1987. "K-Sample Anderson-Darling Tests." *Journal of the American*

Statistical Association 82 (399): 918–24.

Vangel, Mark. 1994. "One-Sided Nonparametric Tolerance Limits" 23: 1137–54. doi:10.1080/03610919408813222.

———. 2002. "Lot Acceptance and Compliance Testing Using the Sample Mean and an Extremum." *Technometrics* 44 (3): 242–49. doi:10.1198/004017002188618428.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the tidyverse." *Journal of Open Source Software* 4 (43): 1686. doi:10.21105/joss.01686.

Witchita State University, and Suresh Keshavanarayana. 2012. "CMH17-Stats V2011 1.1." Witchita State University.