Key Considerations When Understanding Differences in Statistical Methodology Implementations Across Programming Languages – An Introduction to the CAMIS Project

Min-Hua Jen, Brian Varney, Kyle Lee, Benjamin Arancibia, Mia Qi, Lyn Taylor, Christina Fillmore, Joseph Rickert, Mike Stackhouse, Michael Rimler

phuse.global

Contents

Introduction

Traditionally, highly regulated industries (such as the pharmaceutical industry) have limited themselves to the use of commercially available software. When taking such an approach, the responsibility for the validation and testing of the product was often delegated to the software development company themselves, to ensure the software performs in line with its documentation to produce accurate, reliable and reproducible results. However, one downside of this approach is that new methods and functionality can be slow to be adopted, which limits new method implementation and tools that can bring in efficiencies. With the increase in popularity of data science, the rate at which community-led tools and methods are being developed in open-source software is rapid.

The availability of advanced analytic capabilities has led to increased desire for clinical data scientists in regulated industries to have access and approval to use open-source software (Rimler et al. 2022). The use of open-source software is now widely accepted (FDA, 2015); however, the increased variety of tools has resulted in an overlap of capabilities. This has raised challenging questions of traditional approaches to clinical analyses – particularly in situations where the overlap yields different results. One example of this challenge encompasses discrepancies which have been discovered in statistical analysis results between different programming languages, even when working within qualified statistical computing environments. Subtle differences exist between the fundamental approaches and assumptions implemented within each language, yielding differences in results which are correct and consistent with their respective documentation. The fact that these differences exist may cause unease for sponsor companies when submitting to a regulatory agency as it is uncertain if the agency will view these differences as problematic. By understanding the source of any discrepancies, one can reinstate that confidence.

This white paper aims to empower clinical data scientists to make informed choices on the implementation of statistical analyses when multiple languages yield different results. Our objective is not to prescribe what that choice should be, but rather provide guidance on the types of questions clinical data scientists should ask, to identify the fundamental sources of discrepant results. Our objective is also to invite the wider community to contribute to an open-source repository designed for Comparing Analysis Method Implementations in Software *(CAMIS: https://github.com/PSIAIMS/CAMIS).*

Background

As clinical data analytics evolves within the pharmaceutical industry, a large and noteworthy contingent of people and organisations have explored the use of various computational technologies as an effort to reimagine how to tell the story about the data collected during a clinical trial. These technologies, whether available commercially or as open source, offer new potential in the ability of a sponsor company to discover new medicines and demonstrate that they can be safely and effectively administered to patients for a given indication. We are seeing applications of machine learning and artificial intelligence being built into exploratory analyses as well as automation of conventional reporting pipelines. We are witnessing a desired transformation of how we deliver clinical insights from flat data files with rows/columns and compiled PDF reports into dynamic visualisation platforms which facilitate a reviewer to explore the trial database in a three-dimensional way. It has also been noted that the tools other industries most commonly use for these 'new' ways of data engineering, data analytics and data reporting are often built on programming languages not historically used within the pharmaceutical industry. We are therefore experiencing a dramatic shift away from dependence on a small set of commercially available software solutions towards embracing many languages to build and use the best-fit tools to extract the most knowledge from clinical data.

The overdependence on a single solution from one programming language brought to light an element of our data analytics outputs that had previously been overlooked. Within the clinical reporting pipeline (transforming patient-level clinical trial data from collection to submission), the industry has predominantly relied on comparing results to an independently generated second set of results as the primary form of quality control (QC). In the early years, comparisons were made on paper and thoroughly verified by a human that the number in the table matched the number independently derived by a second programmer. As technology progressed, electronic comparisons of the output data presented in a table reduced the risk of human error caused by the validator missing a discrepancy. The theory is that if two people put the same inputs through two independently developed processes and achieve the same outcome, then the outcome must be right. It's not a perfect system and it can produce false positives, but efficiencies were gained and quality improved. However, up until recently, the QC process has nearly always been implemented with the same programming language being used for the generation of results ('on production') and for independent QC. The shift in the industry to explore other languages has now raised questions such as "What if the numbers don't match? Which is correct?"

An example of this question is comparing rounding rules between SAS® and R. It is now becoming well understood that the default rounding rules (implemented in the respective language's **round()** function) are different, but only when the number being rounded is equidistant between the two possible results. The **round()** function in SAS will round the number 'away from zero', meaning 12.5 rounds to the integer 13. The **round()** function in Base R will round the number 'to even', meaning 12.5 rounds to the integer 12. SAS also has the **round()** function, which rounds to even, and the **janitor** package in R contains a function that rounds away from zero. In this use case, SAS produces an accurate result from its **round()** function, based on its documentation, as does R. Both are accurate based on what they say they do, but they produce different results.

Clinical data scientists must then decide how they round this result. Do they round in line with the R default or with the SAS default and why? To answer this question, clinical data scientists need to understand the rationale behind the round-to-even rule and the round-away-from-zero rule, and even other rounding rules that may exist. To our knowledge, this 'how do I round' question had never been asked with respect to clinical trial reporting before the difference between R and SAS default rounding was discovered. It is up to the clinical data scientists to determine and justify the 'correct answer'. For example, with the appropriate number of significant digits, the difference between these results may be inconsequential to interpretation when presenting data on

a table. However, rounding to even is intended to avoid biasing results away from zero, and if this is a risk within an analysis, it should be carefully considered the potential better option.

Why should clinical data scientists care? Why does it matter? One reason is because they want to tell the most accurate story of their data, but more importantly, in the highly regulated pharmaceutical industry, a third-party reviewer will be assessing the integrity of the data. If the reviewer attempts to reproduce the same results and chooses a different language, the clinical data scientists must be able to explain why the results may differ, otherwise the integrity of the entire data package may be questioned. By understanding the implications of choosing a statistical modelling implementation in language A versus language B, the clinical data scientists can communicate the rationale of the choice, based on sound statistical reasoning, and instil confidence in the regulatory body reviewing the submitted data. Moreover, when should clinical data scientists start considering any of the issues mentioned above? They could consider them when planning the study, when planning the draft statistical analysis plans (SAPs) and when performing analysis.

It should be noted that in what follows, it is assumed that statistical packages and routines perform their methods in a manner consistent with their documentation. The question at hand is not whether the procedures are accurate or reliable, but rather in what ways similar implementations across languages differ. Hence, we are not concerned here with another major area of discussion within the industry – the validation of packages and software.

Analysis Framework

With the number of statistical packages and macros currently in use, clinical data scientists face a considerable number of questions to answer, for example which package should I use? What degree of precision is required? Are the analyses reproducible? Here, we propose a framework to help guide clinical data scientists in their thinking in determining the most appropriate statistical analyses for their applications to answer these questions. This framework is not to be taken as a determinative solution. Rather, it should help start the conversation between clinical data scientists and other relevant parties.

Step 1: Define the research question

The research question should be clearly specified in the protocol and in the SAP. Hence the first step is for clinical data scientists to familiarise themselves with the research questions. If the clinical data scientists do not keep this in mind, the remaining steps will turn into a cycle that will remain unguided in the overall purpose of the study in question.

Step 2: Define the statistical design to examine the research questions

The statistical designs are often informed by the protocols and procedures set in an experiment. Considerations such as estimands, randomisation based on inclusion/exclusion criteria, and stratified analyses help determine the model-making process.

Step 3: Look at the technical aspects

The clinical data scientists should perform a basic literature search to determine how to implement the statistical models as specified. A variety of packages may perform the same analyses but use other kinds of optimisation routines in the backend. Here is a starting list of questions to consider:

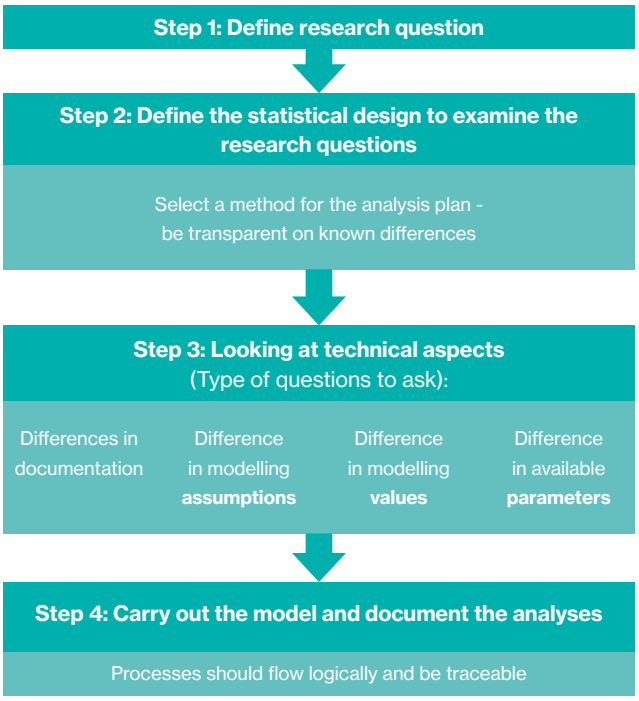
- Which model (and corresponding summary statistics) should be used to answer the research question based on the study design?
- What are the key factors of the model, e.g. coefficient and variance estimation procedures, missing data strategy?
- Can the analysis be carried out in R, or SAS, or some other platform available to the organisation?
- If the analysis can be implemented in both R and SAS, what are the differences in documentation of the key factors?

These questions should be answered in context of the statistical design and be consistent with the organisation's previous practices in similar trials where applicable.

Step 4: Carry out the model and document the analyses

After considering all the technical considerations, the clinical data scientists should carry out the analysis as planned and document accordingly to enable reproducibility of the results. Details such as the R/SAS version, and the packages and macros used, should be recorded.

A four-step algorithm to aid choice of statistical software and methods



Framework Application

We will now consider an application of the proposed framework in the context of survival analysis using the data of 500 subjects from the Worcester Heart Attack Study.1 This class of statistical model has produced the richest set of numerical differences amongst the use cases CAMIS has investigated (Jen & Qi, 2021, 2022). The main goal of this study was to describe factors associated with trends over time, in the incidence and survival rates following hospital admission for acute myocardial infarction.

Step 1: Define the research question

The objective in this use case is to evaluate the efficacy of receiving artery fibrillation treatment in comparison with not receiving treatment, in terms of survival time for patients who have had a heart attack.

Step 2: Define the statistical design to examine the research question

This is a longitudinal study of residents of the Worcester, Mass, metropolitan area hospitalised with acute myocardial infarction (AMI) in seven one-year periods between 1986 and 1997 whose information about prehospital delay was available. Subjects were followed up for death status. Follow-up time for all participants begins at the time of hospital admission after AMI and ends with death or loss to follow-up.

Since the survival status and survival time are both collected, survival analysis is the most appropriate statistical method for answering the research question. For the survival time, the Kaplan-Meier (KM) method (Kaplan and Meier, 1958) will be used to generate KM curves, and the medians, quantiles and percentages of patients' event-free estimates in every three-month intervals for each arm will be summarised. In the possible presence of a substantial number of small cells, the hazard ratio (HR) for treatment effect and its 95% confidence interval (CI) will be estimated using the unstratified Cox proportional hazards (PH) semiparametric model (Cox 1972) and an unstratified log-rank test () will be performed. The p-value from the log-rank test will be used to determine the impact of artery fibrillation on survival time after heart attack.

Step 3: Look at the technical aspects

As a statistician, there is a common understanding that the Cox proportional hazards model is widely available in platforms such as R, SAS and Python. One may ask whether both SAS and R implement the Cox proportional hazards and, from there, check the documentation for how certain statistical methods – confidence interval computation, optimisation routines, etc. – are handled. Clinical data scientists might consider the following initial set of questions:

- **What are the key factors of the Cox proportional hazards model, the KM method and the log-rank test?**
  Cox model: A semiparametric model that assumes proportional hazards.
  Coefficient estimation: The coefficient is estimated by maximising the partial likelihood and computed by iterative algorithms, e.g. the Newton-Raphson method. The partial likelihood function is impacted by a tie-handling approach.
  Variance estimation: The model-based variance estimate is obtained by inverting the information matrix.
  Missing data: Observations with a missing value for the failure time, the censoring variable or the explanatory variables are not used in the analysis. No missing data in the example dataset.
  KM method: The KM method calculates the probability of occurrence of an event at a certain point of time (Breheny, P.,). The estimate of the standard error is computed using Greenwood's formula. The confidence intervals for the quantile survival time and landmark estimate are based on a g-transformed confidence interval for the survival time.
  Log-rank test: A non-parametric test using rank statistics.
- **Can the analysis be carried out in R, SAS or some other platform available to the organisation?**
  After looking at SAS 9.4, we observe that we can model the Cox PH model using SAS PHREG, and the KM curves and log-rank test can be performed using SAS LIFETEST, while in R, we can use coxph, survfit in the survival package *(survminer.pdf (r-project.org))* for the Cox model and KM estimations (Therneau, 2022; Kassambara and Kosinski, 2021).
- **Are there different methods for handling ties in the data?**
  Both SAS and R, according to their documentation, handle Efron's method, Breslow's method and the exact method. By default, in SAS, the method is Breslow. In R, the default is Efron. There is a reference that indicates that both use the same paper (Breslow, 1974; Efron, 1977).
- **Are there differences in the optimisation routines used to compute the maximum likelihood estimates for the treatment (with or without receiving artery fibrillation treatment)?**
  In SAS, the routine used is a modified Newton-Raphson method (Heinze and Schemper, 2001) while in R, the standard Newton-Raphson method is used with half-stepping.

Step 4: Carry out the model and document the analyses

Once an initial inventory of the technical considerations has been developed, the clinical data scientists should carry out the analyses as intended. Below, we indicate for this example that the staff carried out the analyses in both R and SAS for exploratory purposes. Typically, they might implement this in one statistical programming language.

	Results		SAS Results	
	AFB=No	AFB=Yes	AFB=No	AFB=Yes
Event - n	168	47	168	47
Censored - n	254	31	254	31
Time to event (years)				
25th percentile (95% CI)	0.94 (0.55, 1.47)	0.26 (0.05, 1.11)	0.94 (0.51, 1.45)	0.26 (0.05, 0.90)
Median (95% CI)	5.91 (4.32, NE)	2.37 (1.27, 4.24)	5.91 (4.32, NE)	2.37 (1.15, 3.77)
75th percentile (95% CI)	6.44 (6.44, NE)	6.43 (4.24, NE)	6.44 (6.44, NE)	6.43 (4.24, NE)
Min, Max	(0.0, 6.5)	(0.0, 6.4)	(0.0, 6.5)	(0.0, 6.4)
Landmark estimates				
1 - year event-free rate (95% CI)	0.739 (0.699, 0.782)	0.641 (0.543 0.757)	0.739 (0.695, 0.779)	0.641 (0.524, 0.736)
3 - year event-free rate (95% CI)	0.642 (0.595, 0.691)	0.455 (0.351, 0.589)	0.642 (0.591, 0.687)	0.455 (0.335, 0.567)
5 - year event-free rate (95% CI)	0.530 (0.472, 0.595)	0.315 (0.211, 0.470)	0.530 (0.467, 0.589)	0.315 (0.195, 0.442)
p-value		0.001		0.001
Hazard ratio (95% CI)		0.583 (0.421, 0.806)		0.584 (0.422, 0.808)

We note that there are small differences (highlighted in blue) which may lead us to consider asking several further questions to justify the clinical data scientists' choice of platform. For example, how is the upper bound of the confidence intervals estimated? It is expected that the clinical data scientists will cycle between Step 3 and Step 4 to account for any discrepancies which may or may not be addressed by the statistical analysis plan such as the optimisation routine or the default methods, which are often not specified during the planning stage of the analysis. After further consideration, according to the software documentations on handling ties, we learn that the values presented are different because of the following:

- **The default method for handling ties between the two platforms is different.** The survival package in R uses Efron's method of handling ties while SAS uses Breslow's method. In fact, both options are available in R and SAS, so by simply changing the default method, we would expect an identical HR and CI (Franklin D.). From the argument for *coxph*, there are three possible choices for handling tied event times. The Breslow approximation is the easiest to program and hence became the first option coded for almost all computer routines. It then ended up as the default option when other options were added to "maintain backwards compatibility". The Efron option is more accurate if there are many ties, and it is the default option in R. In practice, the number of ties is usually small, in which case all the methods are statistically indistinguishable. All three methods are asymptotically equivalent, which is an ideal property for constructing the relevant estimators (Hertz-Picciotto, I. and Rockhill, B., 1997).
- **The default method for confidence intervals of the KM estimates is different.** R uses 'log', and SAS uses 'log-log'. R and SAS both offer the same types of confidence intervals but must be specified. 'log-log' prevents the problem of having confidence intervals of >1 or <0, which might happen if using 'log' transformation. However, both R and SAS will clip the interval at [0, 1] and report a bound >1 as 1 and <0 as 0.

We constructed a framework which may prove useful for clinical data scientists when considering the technical aspects of implementing the desired analyses. Clinical data scientists may cycle between Steps 3 and 4 to provide a more exhaustive approach. We illustrate an example using both SAS and R to provide a frame of reference, though typically clinical data scientists might implement the analysis in one language, paying careful attention to the available documentation and methods being applied.

Call to action

This white paper outlines a framework for addressing discrepancies between statistical languages, and there is no doubt that this is a cumbersome and complex task – particularly for an organisation to approach alone. Therefore, the PHUSE initiative CAMIS (Comparing Analysis Method Implementations in Software), in collaboration with PSI, is inviting all organisations, working groups and individuals to collaborate to an open-source repository to store and share similarities and differences found when implementing analysis methods in different software. Although work to date has been focused on SAS and R, the framework intends to quickly extend to allow contributions using other open-source software such as python and Julia. The repository aims to be the single go-to source that the community access when they find differences running analysis methods in software.

As more findings are uncovered and stored in this accessible repository, it will naturally improve the quality of data analysis produced as an industry and reduce duplication of efforts. To facilitate this work, PHUSE/PSI have initiated a *https://github.com/PSIAIMS/CAMIS* repository. Contribution is welcome from all, and contact can be made through *workinggroups@phuse.global* or the GitHub repository directly.

References

Rimler, M. S., Rickert, J., Jen, M-H., & Stackhouse, M. (2022). Understanding differences in statistical methodology implementations across programming languages. BioPharm_fall2022FINAL.pdf (higherlogicdownload.s3.amazonaws.com)

U.S. Food and Drug Administration. (2015, May 6). Statistical Software Clarifying Statement. Food and Drug Administration. Retrieved August 12, 2022, from *https://www.fda.gov/media/109552/download*

Kaplan, E. L., & Meier, P. (1958). Nonparametric estimation from incomplete observations. Journal of the American Statistical Association, 53(282), 457¬–481.

Cox, D. R. (1972). Regression models and life-tables. J R Stat Soc Ser B., 34(2), 187–220.

Breheny, P. (2015, September 10). Inference for the Kaplan-Meier Estimator. The University of Iowa. *https://myweb.uiowa.edu/pbreheny/7210/f15/notes/9-10.pdf*

Breslow, N. E. (1974). Covariance analysis of censored survival data. Biometrics, 30(1), 89–99.

Efron, B. (1977). The efficiency of Cox's likelihood function for censored data. Journal of the American Statistical Association, 72(359), 557–565.

Hertz-Picciotto, I., & Rockhill, B. (1997). Validity and efficiency of approximation methods for tied survival times in Cox regression. Biometrics, 53(3), 1151–1156.

Franklin, D. "Our Survival Confidence Intervals are not the Same!" PharmaSUG 2014, Paper SP10. *https://www.pharmasug.org/proceedings/2014/SP/PharmaSUG-2014-SP10.pdf*

Heinze, G., & Schemper, M. (2001). A solution to the problem of monotone likelihood in Cox regression. Biometrics, 57(1), 114–119.

Therneau, T. M. (2022, August 9). Survival analysis [R package survival version 3.4-0]. The Comprehensive R Archive Network. Retrieved August 12, 2022, from *https://cran.r-project.org/package=survival*

Kassambara, A., & Kosinski, M. (2021, March 9). Drawing survival curves using 'ggplot2' [R package survminer version 0.4.9]. The Comprehensive R Archive Network. Retrieved August 12, 2022, from *https://cran.r-project.org/web/packages/survminer/ index.html*

Jen, M-H., & Qi, M. (2021). Survival analysis in R vs. SAS [Conference presentation]. PHUSE Americas Autumn SDE. Retrieved August 12, 2022.

Jen, M-H., & Qi, M. (2022). CSRMLW project summary with focus on survival analysis in R vs. SAS [Conference presentation]. PSI Conference.