
Theses & Dissertations

Graduate Studies

Spring 5-6-2023

Inter-Rater Reliability of Statistics Based on Reconstructed Individual Patient Data from Published Kaplan-Meier Curves

Megan E. Smith
University of Nebraska Medical Center

Tell us how you used this information in this [short survey](#).

Follow this and additional works at: <https://digitalcommons.unmc.edu/etd>



Part of the [Biostatistics Commons](#)

Recommended Citation

Smith, Megan E., "Inter-Rater Reliability of Statistics Based on Reconstructed Individual Patient Data from Published Kaplan-Meier Curves" (2023). *Theses & Dissertations*. 737.

<https://digitalcommons.unmc.edu/etd/737>

This Thesis is brought to you for free and open access by the Graduate Studies at DigitalCommons@UNMC. It has been accepted for inclusion in Theses & Dissertations by an authorized administrator of DigitalCommons@UNMC. For more information, please contact digitalcommons@unmc.edu.

INTER-RATER RELIABILITY OF STATISTICS BASED ON
RECONSTRUCTED INDIVIDUAL PATIENT DATA
FROM PUBLISHED KAPLAN-MEIER CURVES

by

Megan Smith

A THESIS

Presented to the Faculty of
the University of Nebraska Graduate College
in Partial Fulfillment of the Requirements
for the Degree of Master of Science

Biostatistics Graduate Program

Under the Supervision of Professor Lynette Smith

University of Nebraska Medical Center
Omaha, Nebraska

April 2023

Advisory Committee:

Christopher S. Wichman, Ph.D.

Yeongjin Gwon, Ph.D.

ACKNOWLEDGEMENTS

I am deeply grateful to all those who helped me during my academic career at the University of Nebraska Medical Center. In particular, I'd like to offer my greatest thanks to my thesis advisor, Dr. Lynette Smith, for her valuable efforts and guidance during the pursuit of my thesis, as well as the rest of my academic journey at UNMC. I'd also like to thank my committee members, Dr. Christopher Wichman and Dr. Yeongjin Gwon for their advice and feedback throughout the process.

In addition, I would like to thank my parents, Allen and Mary Jo, for their support and encouragement of my academic pursuits throughout my life. My most special thanks to my husband, Vince, for his tremendous support in enduring this process with me.

Finally, this thesis is dedicated to my beloved uncle, Kerry James Smith (10/04/1957 – 11/29/2001).

ABSTRACT

INTER-RATER RELIABILITY OF STATISTICS BASED ON RECONSTRUCTED INDIVIDUAL PATIENT DATA FROM PUBLISHED KAPLAN-MEIER CURVES

Megan Smith, M.S.

University of Nebraska Medical Center, 2023

Advisor: Lynette Smith, Ph.D.

Introduction: Time-to-event outcomes include two elements: an indicator variable for whether the event has taken place, and the length of time from some origin point to the occurrence of the event of interest. Due to the complexity of these data, secondary analysis methods, such as indirect comparisons and meta-analysis, are easier to perform when individual-level patient data (IPD) is available.

Objectives: In 2021, an R package `IPDfromKM` was published, which contains an algorithm for reconstructing IPD from a Kaplan-Meier graph. The current research aimed to investigate the reproducibility of the `IPDfromKM` algorithm.

Methods: Three statisticians (MS, LS, CW) from the University of Nebraska Medical Center Department of Biostatistics independently generated reconstructed IPD for a sample of published Kaplan-Meier curves from peer-reviewed research journals. A sample of survival metrics were collected from the reconstructed IPD datasets using the IPDfromKM package, and then compared for inter-rater reliability with the intraclass correlation coefficient (ICC).

Results: Eleven Kaplan-Meier curves from five recently published journal articles were selected. The absolute agreement for survival time estimates was calculated to have an ICC of 0.967 (95% CI, 0.946, 0.981), demonstrating an excellent level of agreement. Agreement for survival probability estimates was also excellent, with an ICC of 0.983 (95% CI, 0.973, 0.99).

Conclusions: The high level of inter-rater reliability of the reconstructed IPD datasets showed that the IPDfromKM algorithm provides a reproducible reconstruction of the actual survival data.

TABLE OF CONTENTS

Inter-Rater Reliability of Statistics Based on Reconstructed Individual Patient Data from Published Kaplan-Meier Curves.....	i
ACKNOWLEDGEMENTS.....	ii
ABSTRACT	iii
TABLE OF CONTENTS	v
LIST OF TABLES.....	ix
LIST OF FIGURES.....	x
LIST OF ABBREVIATIONS	xi
CHAPTER 1: INTRODUCTION	1
1.1 Survival Analysis: Time-to-Event Outcomes	1
1.2 Motivation for Reconstructing Individual-Level Patient Data	3
1.3 R Package IPDfromKM.....	6
1.4 Objectives.....	7
CHAPTER 2: METHODS.....	9
2.1 Selection of Statistical Programmers	9
2.2 Literature Search for Kaplan-Meier Curves	9
2.3 Reconstruction of Individual Patient Data	10

Figure 1. Flowchart of methods for assessing agreement of reconstructed statistics from the IPDfromKM algorithm	10
Graph Digitizing	11
R Package IPDfromKM.....	12
Table 1. Arguments Passed to preprocess() Function in the R Package	
IPDfromKM.....	13
Table 2. Arguments Passed to getIPD() Function in the R Package	
IPDfromKM.....	14
2.4 Methods for Inter-Rater Agreement of Survival Statistics from Reconstructed IPD Datasets.....	14
Survival Estimates from IPDfromKM Package.....	14
Intraclass Correlation Coefficient.....	15
Table 3. Interpretation of the Intraclass Correlation Coefficient	17
Exploratory Analyses.....	17
Sample Size and Power Analysis	18
CHAPTER 3: RESULTS.....	21
3.1 Literature Search for Published Kaplan-Meier Graphs	21
Table 4. Characteristics of Included Studies and Planned Benchmarks for Analysis of Agreement.....	22
3.2 Characteristics of Reconstructed IPD Datasets	22

Figure 2. Example Output From the <i>survreport()</i> Function in R Package	
IPDfromKM. Reconstructed survival function and cumulative risk function are shown in the top two panels, and survival estimates in the bottom panels.....	23
Table 5. Median Survival Reported and Calculated for Included Studies.....	24
Table 6. Survival Probabilities Reported and Calculated for Included Studies	25
3.3 Intraclass Correlation Coefficient.....	26
Inter-Rater Reliability	26
Table 7. Inter-rater Agreement of IPDfromKM Algorithm	27
Individual Curve Analysis	28
Table 8. Per Curve Analysis for Inter-rater Reliability	28
Secondary Outcomes.....	29
CHAPTER 4: DISCUSSION.....	33
4.1 Findings	33
4.2 Limitations.....	34
4.3 Implications for Further Research.....	36
BIBLIOGRAPHY	37
APPENDIX A: ORIGINAL PUBLISHED KAPLAN-MEIER GRAPHS	41
Figure A.1 Published Kaplan-Meier Plot for Sample Curves 1 and 2.....	41

Figure A.2 Published Kaplan-Meier Plot for Sample Curves 3 and 4.....	42
Figure A.3 Published Kaplan-Meier Plot for Sample Curve 5	43
Figure A.4 Published Kaplan-Meier Plot for Sample Curves 6, 7, 8, and 9....	44
Figure A.5 Published Kaplan-Meier Plot for Sample Curves 10 and 11.....	45
APPENDIX B: RAW DATA FOR AGREEMENT ANALYSIS.....	46
Table B.1. Raw Data for Agreement Analysis	46
APPENDIX C: R CODE.....	48

LIST OF TABLES

Table 1. Arguments Passed to preprocess() Function in the R Package IPDfromKM.....	13
Table 2. Arguments Passed to getIPD() Function in the R Package IPDfromKM	14
Table 3. Interpretation of the Intraclass Correlation Coefficient.....	17
Table 4. Characteristics of Included Studies and Planned Benchmarks for Analysis of Agreement.....	22
Table 5. Median Survival Reported and Calculated for Included Studies	24
Table 6. Survival Probabilities Reported and Calculated for Included Studies.	25
Table 7. Inter-rater Agreement of IPDfromKM Algorithm.....	27
Table 8. Per Curve Analysis for Inter-rater Reliability	28
Table B.1. Raw Data for Agreement Analysis.....	46

LIST OF FIGURES

Figure 1. Flowchart of methods for assessing agreement of reconstructed statistics from the IPDfromKM algorithm.....	10
Figure 2. Example Output From the <i>survreport()</i> Function in R Package IPDfromKM. Reconstructed survival function and cumulative risk function are shown in the top two panels, and survival estimates in the bottom panels.	23
Figure 3. Bland-Altman plot for agreement of reported and reconstructed survival probabilities.....	31
Figure 4. Bland-Altman plot for agreement of reported and reconstructed median survival times	32
Figure A.1 Published Kaplan-Meier Plot for Sample Curves 1 and 2	41
Figure A.2 Published Kaplan-Meier Plot for Sample Curves 3 and 4	42
Figure A.3 Published Kaplan-Meier Plot for Sample Curve 5.....	43
Figure A.4 Published Kaplan-Meier Plot for Sample Curves 6, 7, 8, and 9	44
Figure A.5 Published Kaplan-Meier Plot for Sample Curves 10 and 11	45

LIST OF ABBREVIATIONS

CI	Confidence interval
DFS	Disease-free survival
HR	Hazard ratio
ICC	Intraclass correlation coefficient
IPD	Individual patient data
KM	Kaplan-Meier
LOO	Leave-one-out
OR	Odds ratio
OS	Overall survival
PFS	Progression-free survival
RFS	Recurrence-free survival
RR	Relative risk

CHAPTER 1: INTRODUCTION

1.1 Survival Analysis: Time-to-Event Outcomes

Time-to-event survival analysis is a statistical technique used to analyze the time it takes for an event of interest to occur, such as death, disease, or failure. Survival analyses are so-called because a common application is in the evaluation of duration of survival, where the event of interest is death. Other examples where time-to-event outcomes may be utilized for evaluating a new intervention include: time to conception in fertility treatment, time free of seizures in epilepsy, and length of hospital stay.

Time-to-event outcomes include two elements: an indicator variable for whether the event has taken place, and the length of time from some origin point (i.e. randomization) to the occurrence of the event of interest. In survival analysis, the event is not necessarily observed for all subjects, and those participants are subsequently right-censored. Depending on the scenario, observations can be left-, right-, or interval-censored. Right-censoring is used in survival analysis when it is known that the event (e.g. death) will occur at some point in the future, but did not occur while the subject was being observed. Left-censoring is when an event is known to have occurred before a certain point in

time, but is not known exactly when, and interval-censoring is used when an event is known to have occurred within a given interval, but the exact time is not known. Censored observations contribute the time piece of the outcome variable, while also recognizing that the event of interest did not occur during the period of observation. For example, in a clinical trial of cancer patients, some patients may die during the observed study period, which are counted as events, and other patients will die outside the scope of the study, at some point in the future; these latter observations would be right-censored.

The goal of survival analysis is to estimate the survival function, which describes the probability of the event not occurring up to a given time, and to identify factors that may influence the timing of the event. These methods are commonly used in cancer trials, where new treatments are targeted at significantly increasing patients' survival prognosis. (1) Clinical trials for new drugs seeking FDA approval are often designed to demonstrate superiority over the current standard of care, in terms of overall survival (OS), progression-free survival (PFS), or another time-to-event outcome. OS is considered the gold standard in many oncology clinical trials, due to being objective, patient-centered, and easy to measure.

When the trial is based on a time-to-event endpoint, a Kaplan-Meier (KM) curve is the typical statistical tool used to report and interpret the results of these

analyses.(2) A Kaplan-Meier curve is a graphical representation of the survival function, which shows the probability of survival over time. Kaplan-Meier curves are non-parametric, meaning there are no assumptions made about the underlying distribution of the data, and they are able to model data that are censored, truncated, or have missing values. The curve is constructed by calculating the proportion of individuals who survive without experiencing the event of interest at each time point. These proportions are then plotted against time, resulting in a stepwise function that shows the cumulative survival probability over time. Kaplan-Meier curves are commonly used to estimate survival metrics such as median survival time, the percentage of patients who survive beyond a certain time point, and the hazard ratio, which measures the relative risk of an event occurring in one group compared to another.

1.2 Motivation for Reconstructing Individual-Level Patient

Data

For many incurable diseases, and especially in oncology, there are frequently multiple drugs developed for the exact same or similar indications. With so many different treatments being developed in a relatively short period of time, there is a need to compare efficacies to determine the place in therapy for new pharmaceutical agents.(3) Additionally, newly developed agents have more

evidence in the form of single-arm phase-II studies than randomized phase-III studies, highlighting the need for the capability to compare multiple groups of survival data across trials.

Situations frequently arise where researchers want to compare outcomes across multiple trials, and the methods are fairly straightforward for outcomes measured as binary or count data. The fact that time-to-event outcomes are comprised of two elements makes the comparison of survival outcomes across multiple datasets more complicated. Odds ratios (ORs) or relative risks (RRs) are used to measure dichotomous outcomes, but they only account for the number of events and not time of occurrence. If survival outcomes were to be measured as count data only, then trials of different length and maturity of follow-up may be compared, resulting in an unreliable estimate.⁽⁴⁾ On the other hand, calculating ORs or RRs using survival data only at certain timepoints can make the results more interpretable, but at risk of information loss and potentially misleading estimates. Thus, these types of measures are less appropriate for analyzing time-to-event outcomes.

Meta-analysis is a statistical technique used to combine data from multiple trials to produce a summary estimate. Methods for pooling outcomes that are dichotomous or continuous are fairly straightforward, but methods for combining time-to-event data are more complicated. One common approach

analyzes the hazard ratios (HRs) of individual trials and then combines them into a pooled summary estimate.(5) However, summary HRs are heavily dependent on the assumption of proportional hazards, which has been shown to be especially challenging when combining studies from vastly different follow-up durations.(6) Another drawback of using this method is that it is not applicable to single-arm trials.

In situations like these, where aggregate data measures are not appropriate or insufficient for statistical pooling in meta-analysis, individual-level patient data (IPD) is the gold standard. (7) In many cases, authors of primary research articles will allow access to de-identified IPD in order to enable other researchers to conduct further analyses. However, access to IPD is not always granted, especially in cancer trials. Serious privacy concerns exist when sample sizes and event counts are small. In some cases, mere knowledge of a patient's involvement in a trial could allow someone to distinguish private clinical attributes even in a de-identified data set. (8) These privacy concerns have prevented a large number of clinical trial authors from sharing IPD survival data, which restricts the ability to conduct further analyses and impedes medical knowledge.

In recent years, algorithms that reconstruct individual patient-level data from the graphs of Kaplan-Meier curves have been published and subsequently

improved upon.⁽⁹⁾ The process of reconstructing IPD from Kaplan-Meier survival data involves using statistical methods to estimate the underlying distribution of survival times and the censoring mechanism. These estimates can then be used to simulate individual-level data that is consistent with the observed survival data.

1.3 R Package **IPDfromKM**

In 2021, a new algorithm for reconstructing IPD, **IPDfromKM**, was published by statisticians, Liu et al, at the MD Anderson Cancer Center in Houston.⁽¹⁰⁾ The authors published **IPDfromKM**, a package for the R programming language that contains an algorithm for reconstructing IPD from a Kaplan-Meier graph.⁽¹¹⁾ This allows researchers to obtain synthesized individual patient-level data (IPD) from Kaplan-Meier survival curves, which can then be used for secondary analysis of the survival data (i.e. meta-analysis). The algorithm accepts most Kaplan-Meier graphs for reconstruction of individual patient-level data because the program only requires an image file of the Kaplan-Meier graph, along with the input of the total number of enrolled patients. Notably, this method can also work for other time-to-event endpoints that have been represented with a KM graph.

The **IPDfromKM** package for the R programming language uses an iterative algorithm and a non-parametric KM estimation method to recover individual event times from survival curves, and also includes functions for further data extraction, estimation, and visualization. The package implements a two-stage modified iterative K-M estimation algorithm (modified-iKM) approach proposed by Guyot et al in 2012 (9) with some modifications. (10) Additional information may also be included to increase the accuracy of the estimation, such as numbers of patients at risk at given timepoints, total number of patients, and total number of events. The additional functions for survival analysis can be used to calculate summary statistics such as median survival times and hazard ratios as applicable.

1.4 Objectives

The **IPDfromKM** algorithm is a relatively new algorithm, being published in 2021, but the enhanced accessibility of the R package has led to the quick adoption of this method in many areas of research.(3) The main objective of this research was to describe the level of replicability and agreement of statistics estimated from the **IPDfromKM** method among multiple individuals implementing the algorithm. Reconstructed survival statistics, including survival probabilities and survival times, were estimated from datasets

reconstructed by a sample of programmers, and then compared for level of agreement with the intraclass correlation coefficient (ICC). Reconstructed survival metrics were also compared for agreement between software platforms, and similarity to reported values from the original publications.

Agreement was assessed by the intraclass correlation coefficient (ICC), which quantifies the reliability of multiple raters measuring subjects similarly on a quantitative scale. (12) The ICC ranges from 0 to 1, with 0 being no agreement and 1 meaning perfect agreement. The aim of the current research is to demonstrate that when implemented by different individuals, the **IPDfromKM** algorithm results in good to excellent inter-rater reliability of reconstructed survival estimates, that is, ICC values greater than 0.75.

Additional motivation was for validation and testing the performance of **IPDfromKM** based on real Kaplan-Meier curves published in peer-reviewed journals. The implications of this research are related to the relevance and applicability of the **IPDfromKM** algorithm; this may be a useful tool for reconstructing individual patient survival data if the results are independent of the programmer conducting analysis.

CHAPTER 2: METHODS

2.1 Selection of Statistical Programmers

The main objective of this research was to test the level of agreement of statistics estimated by the **IPDfromKM** algorithm when implemented by different individuals. To accomplish this, it was decided that three individuals would perform the reconstruction of KM survival data, which would subsequently be evaluated for level of agreement. These three individuals were assumed to be a representative sample from the larger population of statisticians, therefore enhancing the generalizability of the results. Two statisticians (LS and CW) from the Department of Biostatistics at University of Nebraska Medical Center agreed to participate in this research, and the author (MS) acted as the third programmer.

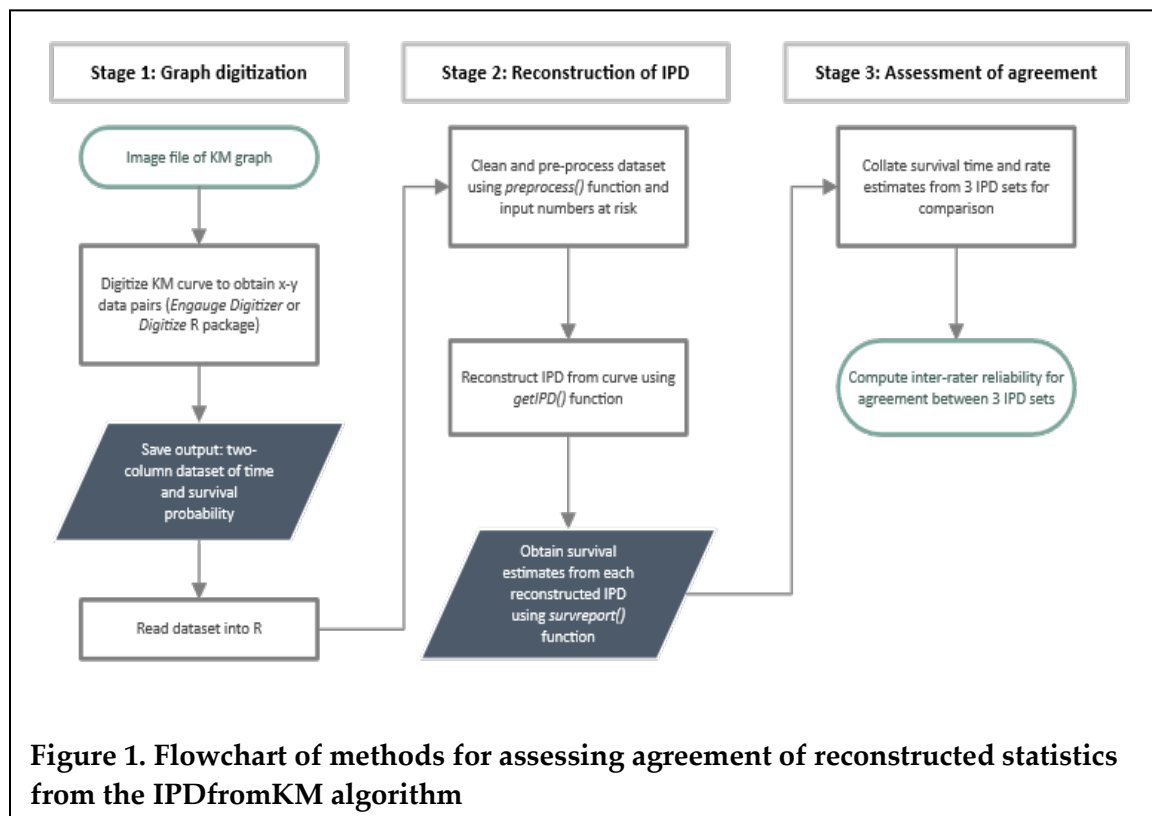
2.2 Literature Search for Kaplan-Meier Curves

Samples of published Kaplan-Meier curves were identified by searching PubMed for the terms “kaplan-meier” “survival” “cancer” and “treatment”. Due to the overwhelming number of primary research articles that presented results with a Kaplan-Meier plot, a sample of articles were selected from recent years of publication. Articles were then further screened for relevance. A variety of graph

designs were selected, including plots with single/multiple curves, small and large sample sizes, censoring marks (with/without), and 95% confidence bands (with/without).

2.3 Reconstruction of Individual Patient Data

Determination of inter-rater reliability of the **IPDfromKM** algorithm was implemented in a multi-stage approach, as shown in the flowchart in Figure 1. In Stage 1, a KM curve was digitized using a graph digitizing program. In Stage 2, reconstructed IPD was obtained from the **IPDfromKM** package, and in Stage 3, the ICC was computed from reconstructed estimates.



Graph Digitizing

For each of the included Kaplan-Meier curves, a high-quality image file was obtained. Each graph was digitalized and converted into x-y data pairs using two methods. Two programmers (MS and LS) used the **Engauge Digitizer** software program (Version 12.1, <https://github.com/markummitchell/engauge-digitizer>), while the third programmer (CW) utilized the R package **Digitize** (<https://github.com/tpoisot/digitize>). (13, 14)

Both digitizing programs use similar methods to convert graphical features into x-y data pairs. Details and specifications for each program can be found at the respective websites. Briefly, the user first imports the image file of the graph into the program, and is then prompted to select with the cursor the locations of the x-axis maximum, y-axis maximum, and the origin on the graph. After these three reference points are input, the user selects points along the survival curve with the cursor. Each programmer was directed to select points as close as possible to the location of the steps in the function. The selected points are then output as a dataset containing x-y data pairs corresponding to points on the Kaplan-Meier curve.

The exported dataset was saved as a .csv file for input. The R package **IPDfromKM** was then used to reconstruct IPD on the basis of the input x-y data

pairs, along with either the number of patients at each specified timepoint if available, or the total number of patients in the trial.

R Package IPDfromKM

After digitizing the Kaplan-Meier survival curve, the csv file dataset of x-y pairs was passed to the *preprocess()* function in the **IPDfromKM** package. The *preprocess()* function takes several arguments for input; these are listed in Table 1.

The *preprocess()* function prepares the raw data for reconstruction by cleaning the data, adjusting for monotonicity, and performing step control. This function outputs a class object that can be used directly in the **IPDfromKM** package. For this analysis, vectors containing the number of patients at risk at specified timepoints were also input into the function, if available. This decision was made in an effort to improve the accuracy of IPD reconstruction. However, it should be noted that this information is not available for every KM curve; in some cases only the total number of patients in the trial is provided.

The *preprocess()* function returns a class object containing the preprocessed data that can be further used in the algorithm. The class object from the previous step can then be passed directly to the *getIPD()* function in the **IPDfromKM** package. The arguments for the *getIPD()* function are listed below in Table 2.

Table 1. Arguments Passed to preprocess() Function in the R Package IPDfromKM

Argument	Description
dat	Two-column dataset extracted from a KM curve, with the first column being survival time, and the second survival probability.
trisk	If applicable, a vector containing risk time points (i.e. points at which number of patients at risk is reported. Often found under the x-axis of the KM curve.
nrisk	If applicable, a vector containing the number of patients at risk. Often found under the x-axis of the KM curve.
maxy	Scale of the y-axis. Set to 1 when probabilities are plotted using decimal numbers (e.g. 0.8). Set to 100 when probabilities are reported in percentages (e.g. 80%)
totalpts	the initial number of patients, with a default value of NULL. However, when both trisk and nrisk are NULL, this number is required for the estimation.
Abbreviations: KM; Kaplan-Meier	

Table 2. Arguments Passed to `getIPD()` Function in the R Package `IPDfromKM`

Argument	Description
<code>prep</code>	Class object returned from <code>preprocess()</code> function
<code>armID</code>	If applicable. Label used as group indicator when multiple KM curves are plotted on one graph
<code>tot.events</code>	Total number of events. May not be available for all KM curves.

Abbreviations: KM; Kaplan-Meier

Once the IPD has been reconstructed with the `getIPD()` function, additional functions can be called to further analyze the data, including `summary()`, `plot()`, and `survreport()`.

2.4 Methods for Inter-Rater Agreement of Survival

Statistics from Reconstructed IPD Datasets

Survival Estimates from `IPDfromKM` Package

At the time of curve selection, each Kaplan-Meier curve included in this analysis was visually inspected, and benchmark metrics were identified for comparison. The `IPDfromKM` package contains additional functions for supplementary survival analysis on the reconstructed IPD that can be used to

calculate summary statistics such as median survival times, survival probability, and hazard ratios, as applicable. In order to compare the reconstructed IPD datasets from the three programmers, the pre-specified benchmark metrics were collected from each of the reconstructed data sets using the *survreport()* function in the **IPDfromKM** package. Standardized metrics were estimated for both survival probabilities and survival times from the reconstructed datasets. The collated probability estimates and time estimates were then compared for level of agreement using the intraclass correlation coefficient (ICC).

Intraclass Correlation Coefficient

The intraclass correlation coefficient (ICC) is an index of reliability that is commonly used to determine the reproducibility of a quantitative measurement.⁽¹²⁾ Depending on the situation and how reliability is being assessed, the ICC can be used several ways.⁽¹⁵⁾ Inter-rater reliability is determined when two or more raters are measuring the same group of subjects; this method is the primary objective of this analysis. Test-retest reliability (repeated measures of the same subject) and intra-rater reliability (multiple scores from the same raters) can also use the ICC for evaluation.

In the primary analysis, the two-way random-effects model was used for measuring inter-rater reliability. In this model, both subjects and raters are

considered as random effects, which is appropriate for this research question because each subject is measured by the same randomly selected set of k -raters with similar characteristics. An additional random-effect term was used where multiple estimates were estimated from the same curve IPD. This model can be represented by the following equation:

$$Y_{ijk} = \beta_0 + \eta_i + \theta_j + \varepsilon_{ijk}$$

where Y_{ijk} is the k th measurement of subject i from curve j , η_i is the random effect of subject i , θ_j is the random effect of curve j , and ε_{ijk} is the unaccounted for error. The variances of these random effects are estimated by the random-effects model, and are also used to calculate the ICC. The reliability between pairs of observations is calculated by the ratio of true variance over true variance plus error variance. In other words, it represents the between-pair variance as a proportion of the total variance of the observations. The ICC is commonly written using the Greek letter ρ , and can be computed with the following formula:

$$\rho = \frac{\sigma_{\eta}^2}{\sigma_{\eta}^2 + \sigma_{\theta}^2 + \sigma_{\varepsilon}^2}$$

where $\sigma_{\eta}^2, \sigma_{\theta}^2, \sigma_{\varepsilon}^2$ are the respective variances as estimated by the random-effects inter-rater reliability model.

The ICC can take a value from zero to one, with zero meaning no agreement and 1 implying absolute agreement. A suggested interpretation is that

index values below 0.5 indicate poor reliability, between 0.5 and 0.75 moderate reliability, between 0.75 and 0.9 good reliability, and above 0.9 indicates excellent reliability (15) (Table 3). A set of $k = 3$ raters was used, and ICC was calculated for absolute agreement of single measurements. All statistical analyses were conducted in R programming language. (Appendix C)

Table 3. Interpretation of the Intraclass Correlation Coefficient

Value	Interpretation
< 0.5	Poor agreement
0.5 – 0.75	Moderate agreement
0.75 – 0.9	Good agreement
> 0.9	Excellent agreement

Exploratory Analyses

Several exploratory subgroup analyses were pre-specified to further investigate the IPDfromKM method. When available, reported survival estimate values from the original publications were compared to the reconstructed estimates. These reported values were also subdivided into survival time and survival probability estimates and observed for differences in level of agreement.

An additional analysis was conducted to compare the raw reconstructed IPD for each curve that is returned from the *getIPD()* function in the **IPDfromKM** package.

A set of simple sensitivity analyses were also conducted to explore the influence of each individual rater in the level of agreement. For each comparison, the dataset for one of the reviewers was omitted, and evaluation of agreement was performed on the remaining datasets. The purpose of this was to investigate trends in individual implementation of the IPDfromKM algorithm. When the programmers MS and LS are omitted from analysis, evaluations of agreement account for inter-rater and inter-platform differences, due to the differences in graph digitizing software. When only CW is omitted from analysis, agreement is affected only by inter-rater effects, because the digitizing platforms are the same. If all raters can apply the algorithm equally well, we would expect to not see much fluctuation in levels of agreement when raters are iteratively omitted from the calculation. The objective for the sensitivity analysis is to investigate the resilience of the agreement estimate.

Sample Size and Power Analysis

The intraclass correlation coefficient measures the reliability of repeated measurements of continuous outcomes. The Power Analysis & Sample Size

software (PASS 2019) was used to determine required sample size.(16) The expected sample size needed was calculated based on 3 raters and an estimated intraclass correlation of approximately 0.8. Using $k = 3$ raters for repeated measurements, a two-sided 95% confidence interval with a width of 0.198 requires a sample size of $n = 37$ when the estimated intraclass correlation is 0.800. Assuming a drop-out rate of 15% for missing (i.e. not calculable) estimates, the target sample size for this analysis was $n = 44$ estimates among the eleven KM survival curves. The data was analyzed using a two-level two-way mixed-effects ANOVA model. (17)

2.5 Bland-Altman Plots for Comparison of Agreement

Bland-Altman plots are a way to visually investigate inter-rater agreement. Also known as Tukey Mean-Difference plots, Bland-Altman plots are a graphical tool that compare two different measurement techniques or devices.(18) This type of plot is created by plotting the mean of two measurements on the x -axis, and the difference between the two on the y -axis. Creating a Bland-Altman plot gives a clear illustration of rater measurements so that the level of agreement between the two methods can be compared, and any systematic differences between them can be identified.

For the purposes of this analysis, Bland-Altman plots were constructed for comparison of reported median survival and reported survival rate estimates with the same rater who was the primary author of this research (MS).

CHAPTER 3: RESULTS

3.1 Literature Search for Published Kaplan-Meier Graphs

Eleven curves from five recently published journal articles were selected for comparison of inter-rater agreement. (19-23) A variety of graphical designs were represented, including plots for both phase II single-arm trials and phase III randomized trials, censoring marks (with or without), and 95% confidence bands (with or without). Original high-resolution image files were obtained for each included KM curve and can be found in the appendix (Appendix A).

For each curve, benchmark timepoints and percentiles were chosen for the primary outcome of comparison of overall inter-rater reliability (Table 3).

Median survival estimates were reported for all curves except one; curve 5 data did not reach maturity for calculation of median survival. Median survival estimates can be found in Table 4. All studies except one (curves 10 and 11) also reported survival probability estimates for at least one benchmark timepoint, shown in Table 5. Curve 5 is excluded from Table 4, and Curves 10 and 11 are excluded from Table 5 due to no information to report.

Table 4. Characteristics of Included Studies and Planned Benchmarks for Analysis of Agreement

ID	Sample Size	Maximum Survival on KM Plot, months	Benchmark Months for Survival Comparison	Benchmark Percentiles for Comparison
1 (19)	22	60	6, 12, 24, 36, 48	85, 75, 60, 50
2 (19)	22	60	12, 18, 24, 30, 48	85, 75, 60, 50
3 (23)	55	18	3, 6, 9, 12, 15	75, 60, 50, 40, 30
4 (23)	55	18	3, 6, 9, 12, 15	85, 75, 60, 50
5 (20)	59	30	3, 6, 9, 12, 18, 24	85, 75, 65, 55
6 (21)	233	72	12, 24, 36, 48, 60, 66	85, 75, 60, 50
7 (21)	237	72	12, 24, 36, 48, 60, 66	75, 60, 50, 40, 30
8 (21)	339	72	12, 24, 36, 48, 60, 66	85, 75, 60, 50
9 (21)	343	72	12, 24, 36, 48, 60, 66	85, 75, 60, 50, 40
10 (22)	41	36	6, 12, 18, 24, 30	75, 60, 50, 40, 30
11 (22)	41	54	6, 12, 18, 24, 36, 48	85, 75, 60, 50

Abbreviations: KM Kaplan-Meier

3.2 Characteristics of Reconstructed IPD Datasets

Each of the three programmers independently implemented the IPDfromKM algorithm using the graphical digitizing program of their choice to

obtain reconstructed IPD datasets. The pre-specified benchmark survival estimates were then generated for each reconstructed dataset using the *survreport()* function in the **IPDfromKM** package. Survival estimates reported in the original publication for each study are presented in Table 4 and Table 5, along with the respective estimates from each reconstructed IPD dataset as well. An example of the output from the *survreport()* function in the **IPDfromKM** package is shown below in Figure 1.

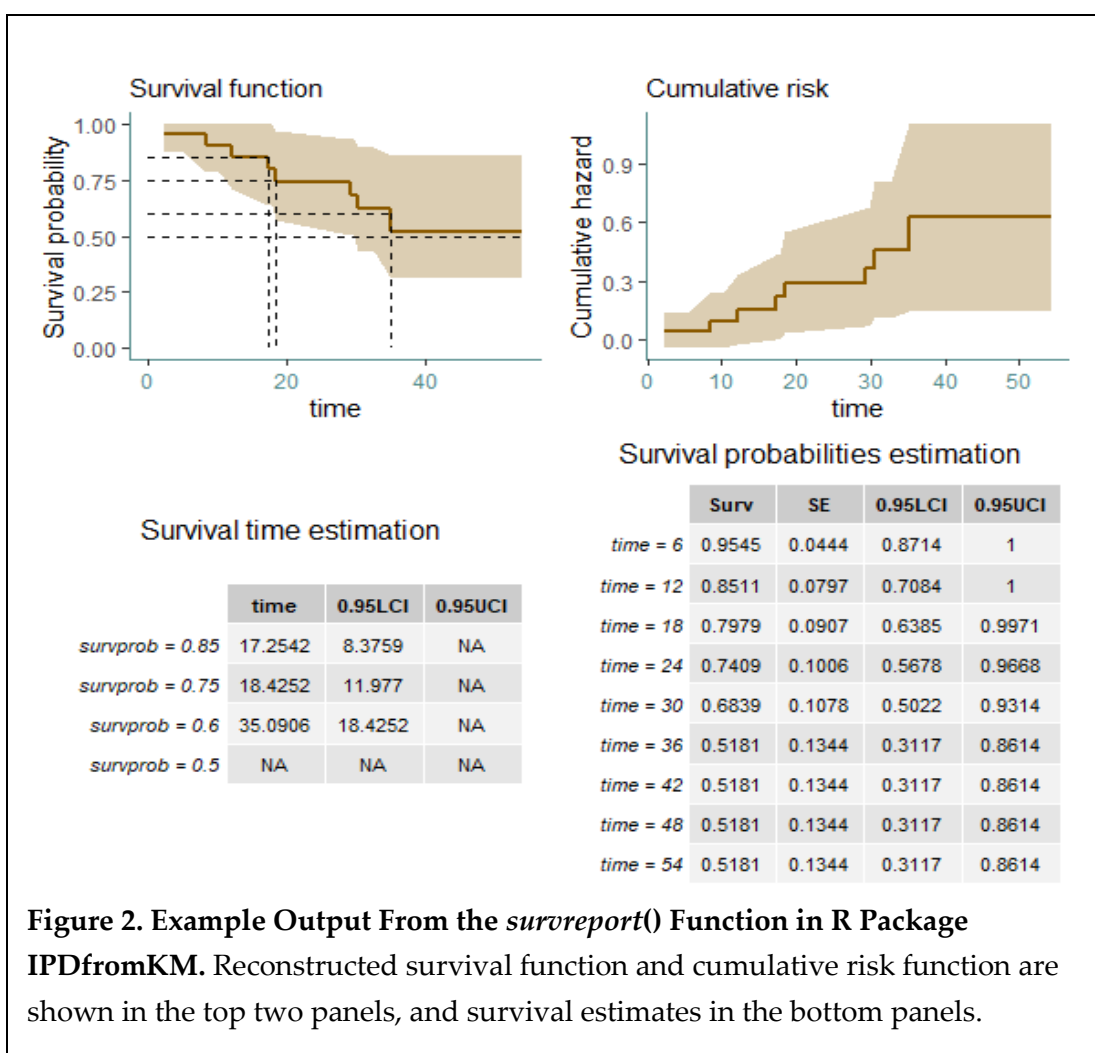


Table 5. Median Survival Reported and Calculated for Included Studies

ID	Outcome	Reported Median Survival, months	Reconstructed Median Survival, months (MS)	Reconstructed Median Survival, months (LS)	Reconstructed Median Survival, months (CW)
1	OS	35.1	NC	NC	54.17
2	RFS	34	34.05	34.05	54.34
3	PFS	4.07	4.07	4.07	4.05
4	OS	13.73	13.75	13.45	13.73
6	DFS	65.8	65.86	66.48	65.82
7	DFS	21.9	21.94	21.34	21.26
8	DFS	65.8	65.87	66.11	NC
9	DFS	28.1	28.04	29.44	36.43
10	PFS	15.23	15.22	15.48	15.52
11	OS	27.57	27.71	28.13	27.56

Abbreviations: DFS, disease-free survival; NC, not calculable; NR, not reported; OS, overall survival; PFS, progression-free survival; RFS, recurrence-free survival

Table 6. Survival Probabilities Reported and Calculated for Included Studies

ID	Outcome	Time, months	Reported Survival Probability, %	Reconstr. Survival Probability, % (MS)	Reconstr. Survival Probability, % (LS)	Reconstr. Survival Probability, % (CW)
1	OS	12	85.1	85.11	89.16	85.11
		24	74.5	74.09	77.62	74.09
		36	49.7	51.81	NC	56.99
2	RFS	12	100.0	100.00	100.00	100.00
		24	52.4	52.66	56.82	51.69
		36	43.6	42.13	NC	51.69
3	PFS	12	17.3	17.04	19.78	16.92
4	OS	12	54.1	54.37	55.58	54.37
5	PFS	12	72.5	72.50	73.75	72.50
6	DFS	24	90	89.97	90.39	89.99
		36	84	83.41	83.79	83.43
		48	70	69.24	69.74	69.74
7	DFS	24	46	46.40	46.43	46.37
		36	34	34.25	33.93	34.01
		48	29	28.80	28.51	28.64
8	DFS	24	90	90.40	90.07	92.04
		36	85	84.40	84.72	88.64
		48	73	72.75	73.06	83.59
9	DFS	24	55	54.46	54.64	59.73
		36	44	43.99	44.12	50.38
		48	38	37.70	37.49	44.09

Abbreviations: DFS, disease-free survival; NR, not reported; OS, overall survival; PFS, progression-free survival; RFS, recurrence-free survival

3.3 Intraclass Correlation Coefficient

Inter-Rater Reliability

For the primary outcome of assessing inter-rater reliability, the values for the three reconstructed IPD datasets were assessed using the intraclass correlation coefficient (ICC). Standardized metrics were estimated for both survival probabilities and survival times from the reconstructed datasets, and then compared for level of agreement using the intraclass correlation coefficient (ICC). The raw data for all primary analysis measurements can be found in the Appendix (Table B.1).

The absolute agreement for survival time estimates with $k = 3$ raters on an actual sample of $n = 44$ estimates from 11 KM curves was calculated to have an ICC of 0.966 (95% CI: 0.926, 0.989), demonstrating an excellent level of agreement. With a sample of $n = 53$ estimates from 11 KM survival curves, the agreement for survival probability estimates had an ICC of 0.933 (95% CI: 0.860, 0.979). Primary and secondary outcomes are reported in Table 6. The primary outcomes of level of agreement for survival probability and survival time estimates were both greater than 0.9, and indicate an excellent level of agreement.

Table 7. Inter-rater Agreement of IPD from KM Algorithm

Agreement Measure	Sample Size	ICC	95% CI
Survival Times	44	0.966	(0.926, 0.989)
Survival Probabilities	53	0.933	(0.860, 0.979)
Reported Values			
Survival Times	8	0.865	(0.672, 0.967)
Survival Probabilities	19	0.993	(0.986, 0.997)
Sensitivity Analysis			
Survival Times			
No MS	44	0.941	(0.870, 0.981)
No LS	44	0.941	(0.869, 0.981)
No CW	47	0.957	(0.904, 0.986)
Survival Probabilities			
No MS	53	0.904	(0.791, 0.969)
No LS	56	0.890	(0.760, 0.965)
No CW	58	0.902	(0.788, 0.969)
Abbreviations: CI, Confidence interval; ICC Intraclass correlation coefficient			

Individual Curve Analysis

An additional analysis was conducted to compare the raw reconstructed IPD times for each curve and assess the ICC. Results are shown in Table 8. The per-curve analysis of the raw reconstructed time data shows generally good agreement among raters. Curves 3, 4, 6, 7, 8, 9, and 10 all have ICC values very

Table 8. Per Curve Analysis for Inter-rater Reliability

Curve ID	Sample Size	ICC	95% CI
1	22	0.886	(0.782, 0.947)
2	22	0.853	(0.730, 0.931)
3	55	0.990	(0.984, 0.994)
4	55	0.998	(0.997, 0.999)
5	59	0.897	(0.734, 0.952)
6	233	0.998	(0.997, 0.998)
7	237	0.992	(0.989, 0.994)
8	339	0.998	(0.998, 0.998)
9	343	0.999	(0.997, 0.999)
10	41	0.987	(0.977, 0.993)
11	41	0.684	(0.344, 0.845)

Abbreviations: CI, Confidence interval; ICC Intraclass correlation coefficient

close to 1, demonstrating the excellent agreement between individual programmers implementing the IPDfromKM method. Other curves show good to excellent inter-rater reliability, with confidence intervals ranging from above 0.7 to greater than 0.9. Notably, Curve 11 showed only moderate agreement, with a point estimate of 0.684 and 95% confidence interval (0.344, 0.845). The algorithm produced generally consistent estimates, but curves with smaller sample size may be more affected by unreliability.

Secondary Outcomes

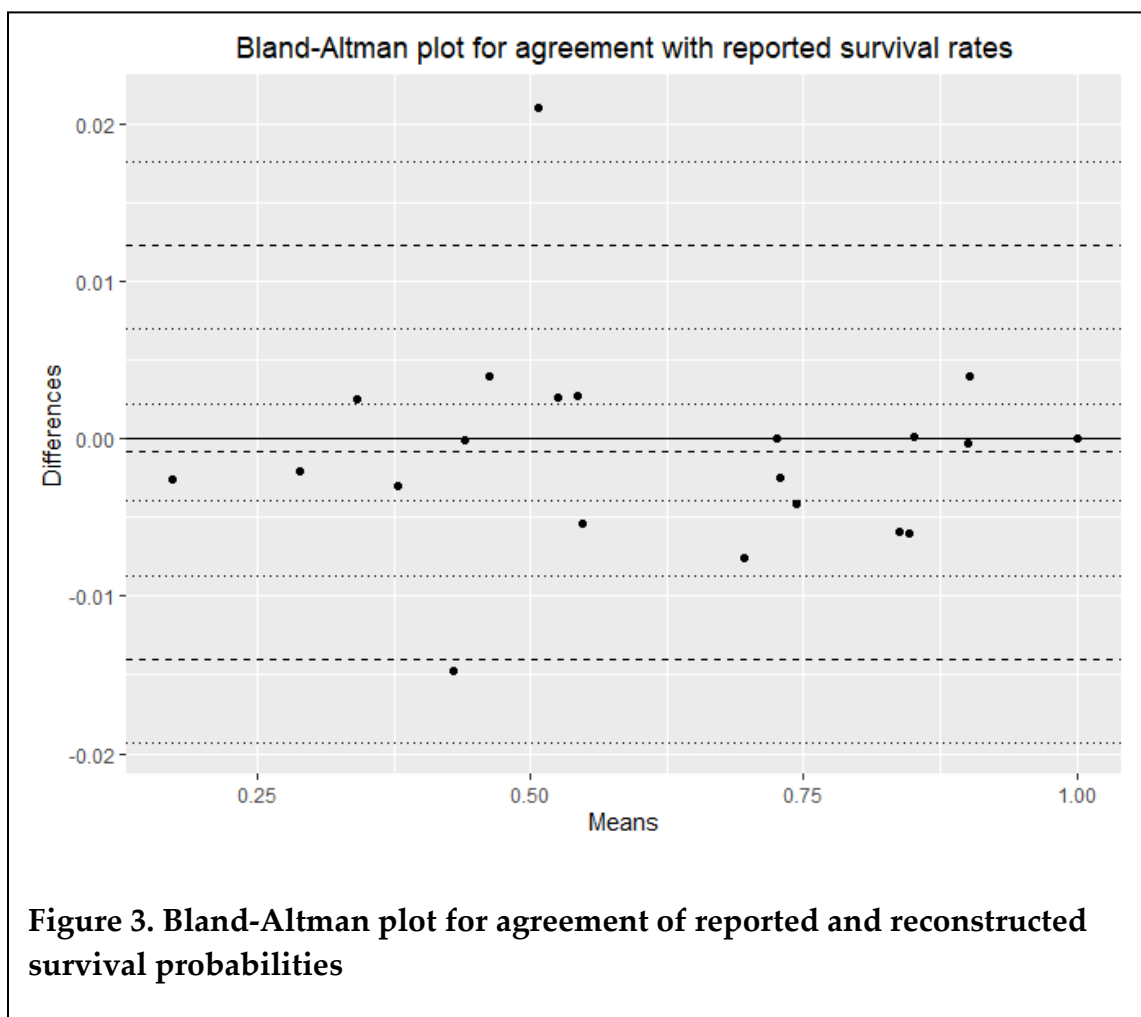
A secondary analysis was conducted on the subset of survival estimates that were reported in the original research studies compared with the estimates from the three reconstructed IPD datasets. Again, survival estimates were divided into time estimates and probability estimates, and ICC was calculated for each group. For this secondary analysis, the difference in levels of agreement between time and probability estimates was more drastic. Survival probability estimates had a calculated ICC value of 0.993 (95% CI: 0.986, 0.997), while the agreement level for median survival time estimates was represented by an ICC of only 0.865. With a 95% confidence interval (CI) ranging from 0.672 to 0.967, the level of agreement between reported and reconstructed median survival times

was indicated as moderate to excellent agreement, demonstrating a higher level of uncertainty in this measure.

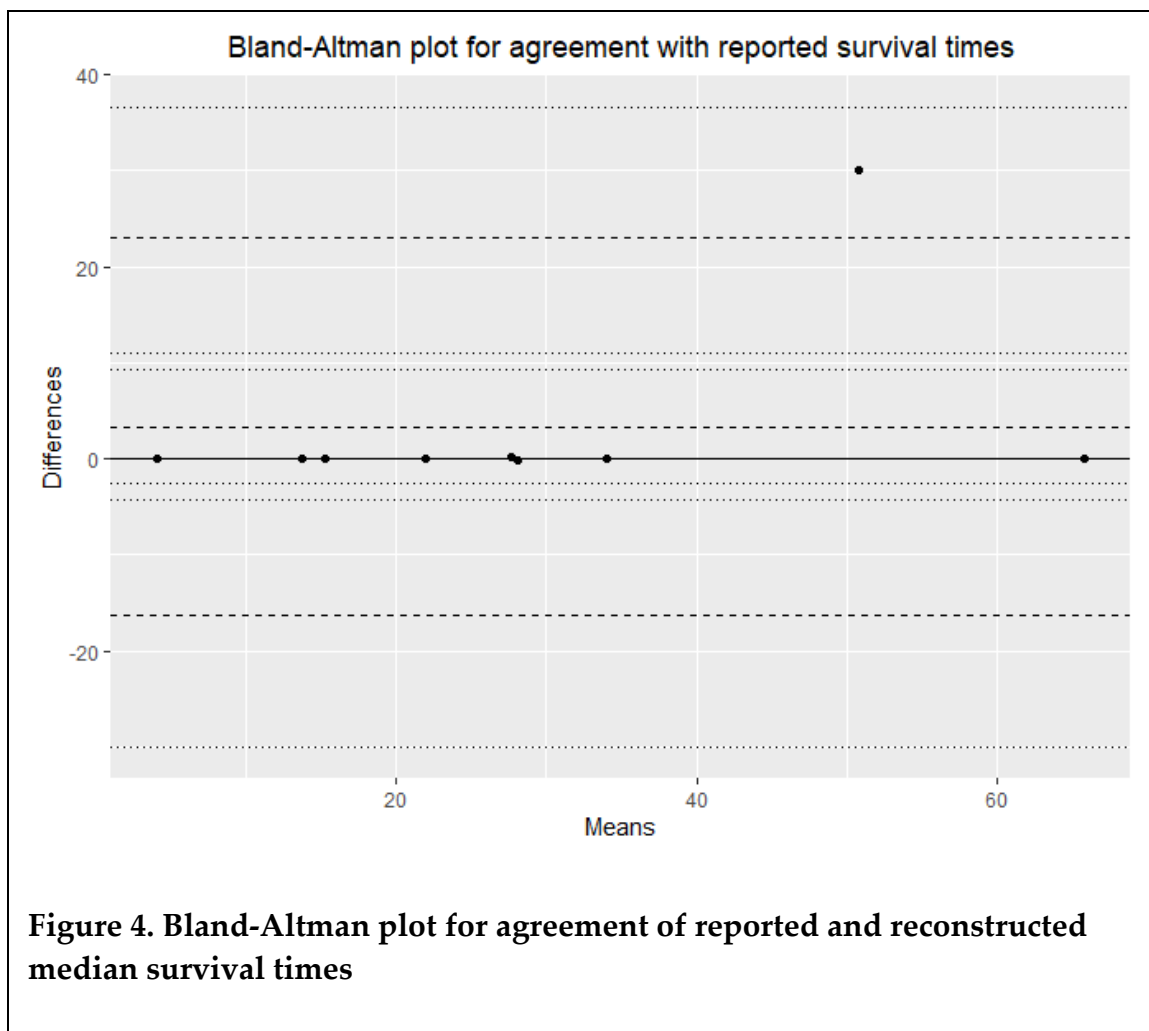
A set of simple sensitivity analyses were also conducted to explore the influence of each individual rater in the level of agreement. For each comparison, the dataset for one of the reviewers was omitted, and evaluation of agreement was performed on the remaining datasets. The results for this analysis generally showed that the omission of each individual rater does not drastically change the agreement level of the estimates. All of the leave-one-out sensitivity analyses maintained an excellent level of agreement, with ICC values greater than 0.9. Overall, there was no evidence found that any one of the individuals drastically affected the level of inter-rater reliability of the estimates.

3.4 Bland-Altman Plots

The Bland-Altman plots are shown below in Figures 3 and 4. The estimated survival probabilities (shown in Figure 3) presented good visual agreement, with points scattered about the zero line with no clear pattern. A couple points have a large difference compared to reported values, but the plot generally shows good level of agreement between estimates with no clear discernable systematic biases.



Similarly, the Bland-Altman plot representing the reported median survival time estimates (shown in Figure 4 below) showed generally good agreement. There was a single data point that had an extreme value for difference compared to mean, but all other points were very close to the zero line with no clear discernable patterns. There appear to be no unidentified sources of systematic bias within the data.



CHAPTER 4: DISCUSSION

4.1 Findings

The IPDfromKM algorithm is a relatively new algorithm, published in 2021, that has enabled researchers to independently reconstruct IPD from published Kaplan-Meier graphs, which can subsequently be used for secondary analysis of survival data. The main objective of this research was to investigate the level of replicability of the IPDfromKM method among multiple individuals implementing the algorithm, assessed for agreement with the intraclass correlation coefficient (ICC). The aim of the current research was to demonstrate that, the IPDfromKM algorithm results in good to excellent inter-rater reliability of reconstructed survival estimates when implemented by different individuals.

The high level of inter-rater reliability of the reconstructed IPD datasets showed that the IPDfromKM algorithm provides a reliable reconstruction of the actual survival data. All but one of the calculated ICC values were greater than 0.9, indicating an excellent level of agreement. The inter-rater reliability for reconstructed median survival times compared to the actual reported values was the lowest, with an ICC of 0.865 (95% CI, 0.672, 0.967). The wide range of the 95%

confidence interval (CI) is indicative of the small sample size for this specific comparison, resulting in greater uncertainty in the reliability statistic.

Many of the individual curves resulted in ICC values very close to 1, demonstrating the excellent agreement between individual programmers implementing the IPDfromKM method. Other curves show good to excellent inter-rater reliability, with confidence intervals ranging from above 0.7 to greater than 0.9. Curves with lower ICC values tended to be from trials with smaller number of subjects, suggesting that small sample size curves analyzed by this algorithm may be at higher risk of unreliable estimates.

Additionally, the Bland-Altman plots show generally good agreement between reported values and reconstructed values. A few points have relatively large differences from reported values, but the plots generally show good level of agreement between survival time and rate estimates with no clear discernable systematic biases.

4.2 Limitations

There were several limitations with the implementation of the IPDfromKM algorithm. Firstly, the algorithm does not accept input for censoring marks. Rather, censored observations survival times are estimated according to the modified i-KM method, introducing some level of uncertainty. Also,

reconstructed IPD from this method does not contain any information about covariates from the original dataset, unless they are graphically represented on the Kaplan-Meier graph. For example, in order to obtain patient-level information on gender, we would require a KM graph with separate survival curves for male and female subjects.

Additionally, limitations related to graph processing were found during this research. A variety of graph types were selected for this analysis in order to test a real representative sample of published KM curves. Some curves incorporate censoring marks, which are additional symbols along the curve marking where censored observations occur. Kaplan-Meier graphs with a remarkably high number of censoring marks were sometimes challenging during the graphical processing step in discerning the location of the individual steps on the survival curve. Likewise, trials with an exceptionally large sample size could pose the same challenges.

Lastly, it was found that in this sample of eleven curves, the reported median survival time estimates were associated with the lowest levels of inter-rater reliability. Incidentally, a large proportion of the survival curves included in this analysis had tails that ended near the median survival line. This suggests that survival curves that end close to the median survival may be quite inaccurately estimated.

4.3 Implications for Further Research

Overall, the **IPDfromKM** algorithm and package was found to be a powerful tool for synthesizing IPD from published Kaplan-Meier survival curves. The algorithm accepts most Kaplan-Meier graphs because the program only requires an image file of the KM survival curve and the total number of enrolled patients. The straightforwardness of the approach, combined with the broad eligibility of most published KM survival curves, poses significant potential for reconstruction of many time-to-event datasets. Notably, this method can work for any time-to-event endpoint that is represented with a KM graph. This algorithm is especially promising for implementation in secondary analysis of survival data, such as meta-analyses or indirect comparisons for decision-making in clinical practice.

BIBLIOGRAPHY

1. Delgado A, Guddati AK. Clinical endpoints in oncology - a primer. *Am J Cancer Res.* 2021;11(4):1121-31. Epub 20210415. PubMed PMID: 33948349; PubMed Central PMCID: PMC8085844.
2. Batson S, Greenall G, Hudson P. Review of the Reporting of Survival Analyses within Randomised Controlled Trials and the Implications for Meta-Analysis. *PLOS ONE.* 2016;11(5):e0154870. doi: 10.1371/journal.pone.0154870.
3. Messori A, Damuzzo V, Rivano M, Cancanelli L, Di Spazio L, Ossato A, et al. Application of the IPDfromKM-Shiny Method to Compare the Efficacy of Novel Treatments Aimed at the Same Disease Condition: A Report of 14 Analyses. *Cancers.* 2023;15(6):1633. PubMed PMID: doi:10.3390/cancers15061633.
4. Tierney JF, Stewart LA, Ghersi D, Burdett S, Sydes MR. Practical methods for incorporating summary time-to-event data into meta-analysis. *Trials.* 2007;8(1):16. doi: 10.1186/1745-6215-8-16.
5. Hebert AE, Kreaden US, Yankovsky A, Guo D, Li Y, Lee S-H, et al. Methodology to standardize heterogeneous statistical data presentations for combining time-to-event oncologic outcomes. *PLOS ONE.* 2022;17(2):e0263661. doi: 10.1371/journal.pone.0263661.
6. Rulli E, Ghilotti F, Biagioli E, Porcu L, Marabese M, D'Incalci M, et al. Assessment of proportional hazard assumption in aggregate data: a systematic review on statistical methodology in clinical trials using time-to-event endpoint. *British Journal of Cancer.* 2018;119(12):1456-63. doi: 10.1038/s41416-018-0302-8.

7. Stewart LA, Tierney JF. To IPD or not to IPD? Advantages and disadvantages of systematic reviews using individual patient data. *Eval Health Prof.* 2002;25(1):76-97. doi: 10.1177/0163278702025001006. PubMed PMID: 11868447.
8. Bonomi L, Jiang X, Ohno-Machado L. Protecting patient privacy in survival analyses. *Journal of the American Medical Informatics Association.* 2019;27(3):366-75. doi: 10.1093/jamia/ocz195.
9. Guyot P, Ades AE, Ouwens MJNM, Welton NJ. Enhanced secondary analysis of survival data: reconstructing the data from published Kaplan-Meier survival curves. *BMC Medical Research Methodology.* 2012;12(1):9. doi: 10.1186/1471-2288-12-9.
10. Liu N, Zhou Y, Lee JJ. IPDfromKM: reconstruct individual patient data from published Kaplan-Meier survival curves. *BMC Medical Research Methodology.* 2021;21(1):111. doi: 10.1186/s12874-021-01308-8.
11. Liu N, Lee JJ, Zhou Y. IPDfromKM: Map Digitized Survival Curves Back to Individual Patient Data. 0.1.10 ed. CRAN.R-project.org2020.
12. Shrout PE, Fleiss JL. Intraclass correlations: uses in assessing rater reliability. *Psychol Bull.* 1979;86(2):420-8. doi: 10.1037//0033-2909.86.2.420. PubMed PMID: 18839484.
13. Poisot T. The digitize package: extracting numerical data from scatterplots. *The R Journal.* 2011;3(1):25-6.
14. Mitchell M, Muftakhidinov B, Winchen T. Engauge Digitizer Software. 12.1 ed2023.

15. Koo TK, Li MY. A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research. *Journal of Chiropractic Medicine*. 2016;15(2):155-63. doi: <https://doi.org/10.1016/j.jcm.2016.02.012>.
16. NCSS L. PASS 2019 Power Analysis and Sample Size Software. 19.0.6 ed. Kaysville, Utah, USA: NCSS, LLC; 2019.
17. Bonett DG. Sample size requirements for estimating intraclass correlations with desired precision. *Statistics in Medicine*. 2002;21(9):1331-5. doi: <https://doi.org/10.1002/sim.1108>.
18. Bland JM, Altman DG. Measuring agreement in method comparison studies. *Stat Methods Med Res*. 1999;8(2):135-60. doi: 10.1177/096228029900800204. PubMed PMID: 10501650.
19. Acuna-Villaorduna A, Shankar V, Wysota M, Jirgal A, Kabarriti R, Bellemare S, et al. Induction Chemotherapy With FOLFIRINOX Followed by Chemoradiation With Gemcitabine in Patients With Borderline-Resectable Pancreatic Ductal Adenocarcinoma. *Cancer Control*. 2022;29:1-9. doi: 10.1177/10732748221134411. PubMed PMID: 36221952.
20. Tanaka H, Tanzawa S, Misumi T, Makiguchi T, Inaba M, Honda T, et al. A phase II study of S-1 and cisplatin with concurrent thoracic radiotherapy followed by durvalumab for unresectable, locally advanced non-small-cell lung cancer in Japan (SAMURAI study): primary analysis. *Therapeutic Advances in Medical Oncology*. 2022;14:1-14. doi: 10.1177/17588359221142786.
21. Herbst RS, Wu Y-L, John T, Grohe C, Majem M, Wang J, et al. Adjuvant Osimertinib for Resected EGFR-Mutated Stage IB-III A Non-Small-Cell Lung Cancer: Updated Results From the Phase III Randomized ADAURA Trial. *Journal of Clinical Oncology*. 2023;41(10):1830-40. doi: 10.1200/JCO.22.02186.

22. Shi Y, Xu H, Raynor WY, Ding J, Lin L, Zhou C, et al. Efficacy and Failure Patterns of Early SBRT to the Primary Tumor in Advanced EGFR-Mutation-Positive Lung Cancer with EGFR-TKI Treatment: A Prospective, Single Arm, Phase II Study. *Life*. 2022;12(12):1954. PubMed PMID: doi:10.3390/life12121954.

23. Takeda M, Shimokawa M, Nakamura A, Nosaki K, Watanabe Y, Kato T, et al. A phase II study (WJOG12819L) to assess the efficacy of osimertinib in patients with *EGFR* mutation-positive NSCLC in whom systemic disease (T790M-negative) progressed after treatment with first- or second-generation EGFR TKIs and platinum-based chemotherapy. *Lung Cancer*. 2023;177:44-50. doi: 10.1016/j.lungcan.2023.01.011.

APPENDIX A: ORIGINAL PUBLISHED KAPLAN-MEIER GRAPHS

Figure A.1 Published Kaplan-Meier Plot for Sample Curves 1 and 2

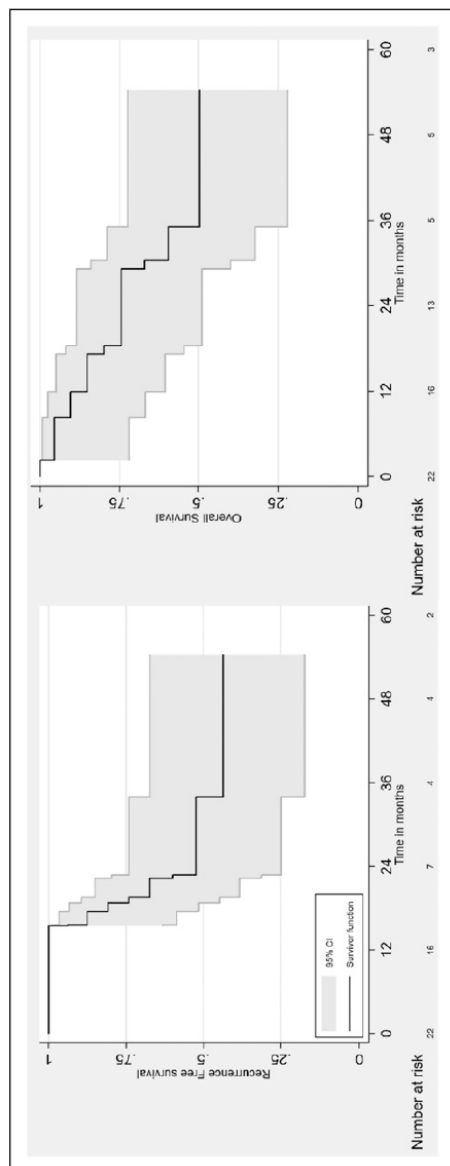


Figure 3. Recurrence Free-Survival and Overall Survival of patients with BR-PDAC treated with neoadjuvant FOLFIRINOX followed by Gemcitabine/RT (n = 22)

Figure A.2 Published Kaplan-Meier Plot for Sample Curves 3 and 4

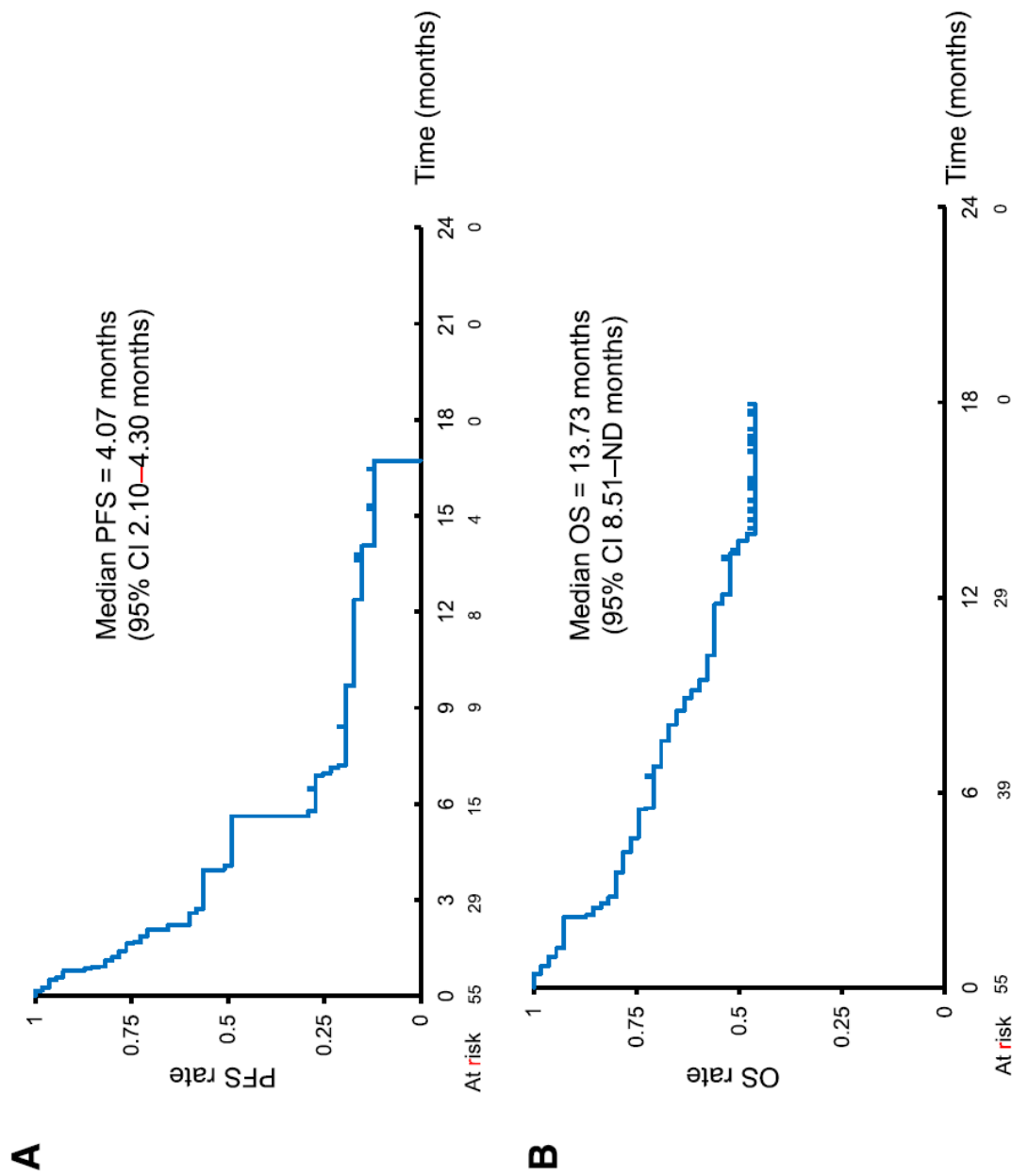


Fig. 3. Survival. Kaplan-Meier plots of progression-free survival (PFS; A) and overall survival (OS; B). CI, confidence interval; ND, not defined.

Figure A.3 Published Kaplan-Meier Plot for Sample Curve 5

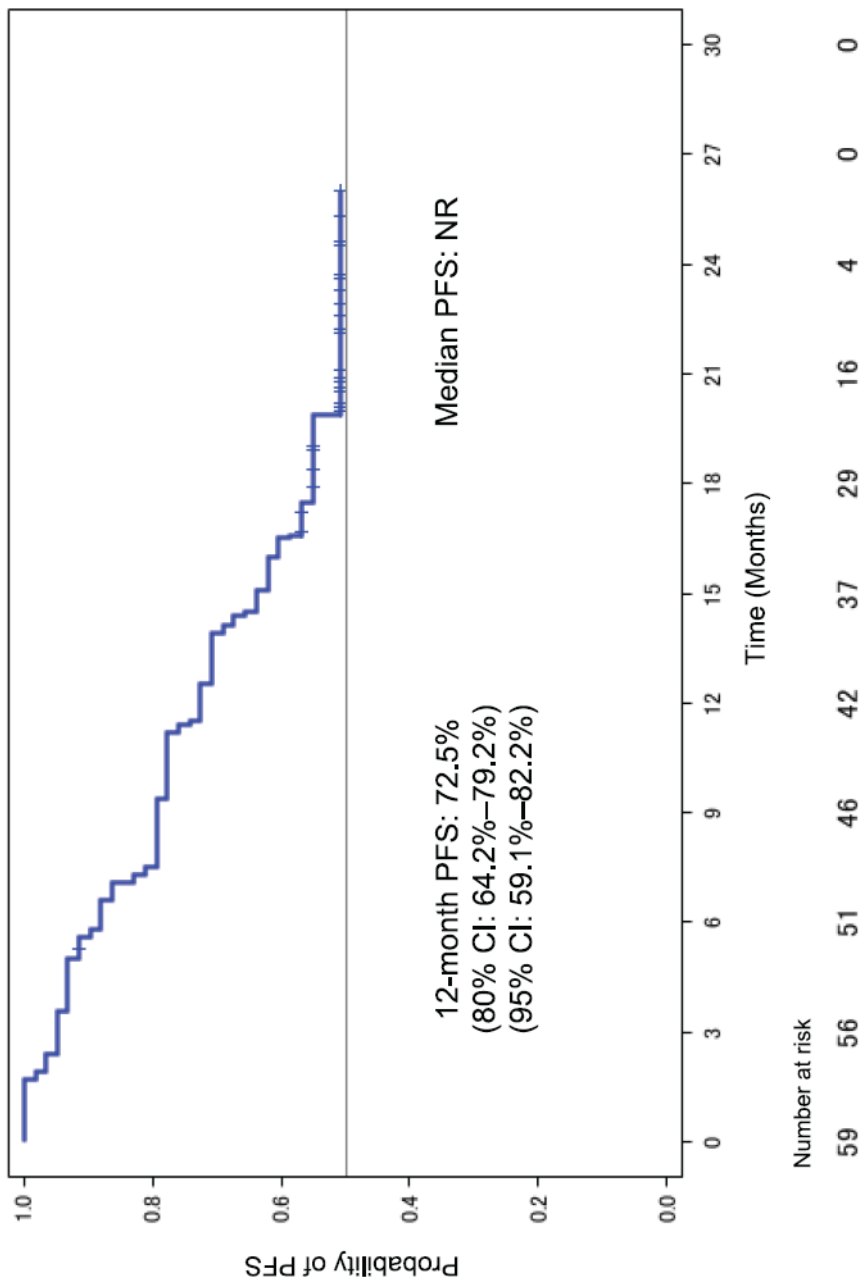


Figure 2. Kaplan-Meier curve for PFS from the first registration in the intention-to-treat population. CI, confidence interval; NR, not reached; PFS, progression-free survival.

Figure A.4 Published Kaplan-Meier Plot for Sample Curves 6, 7, 8, and 9

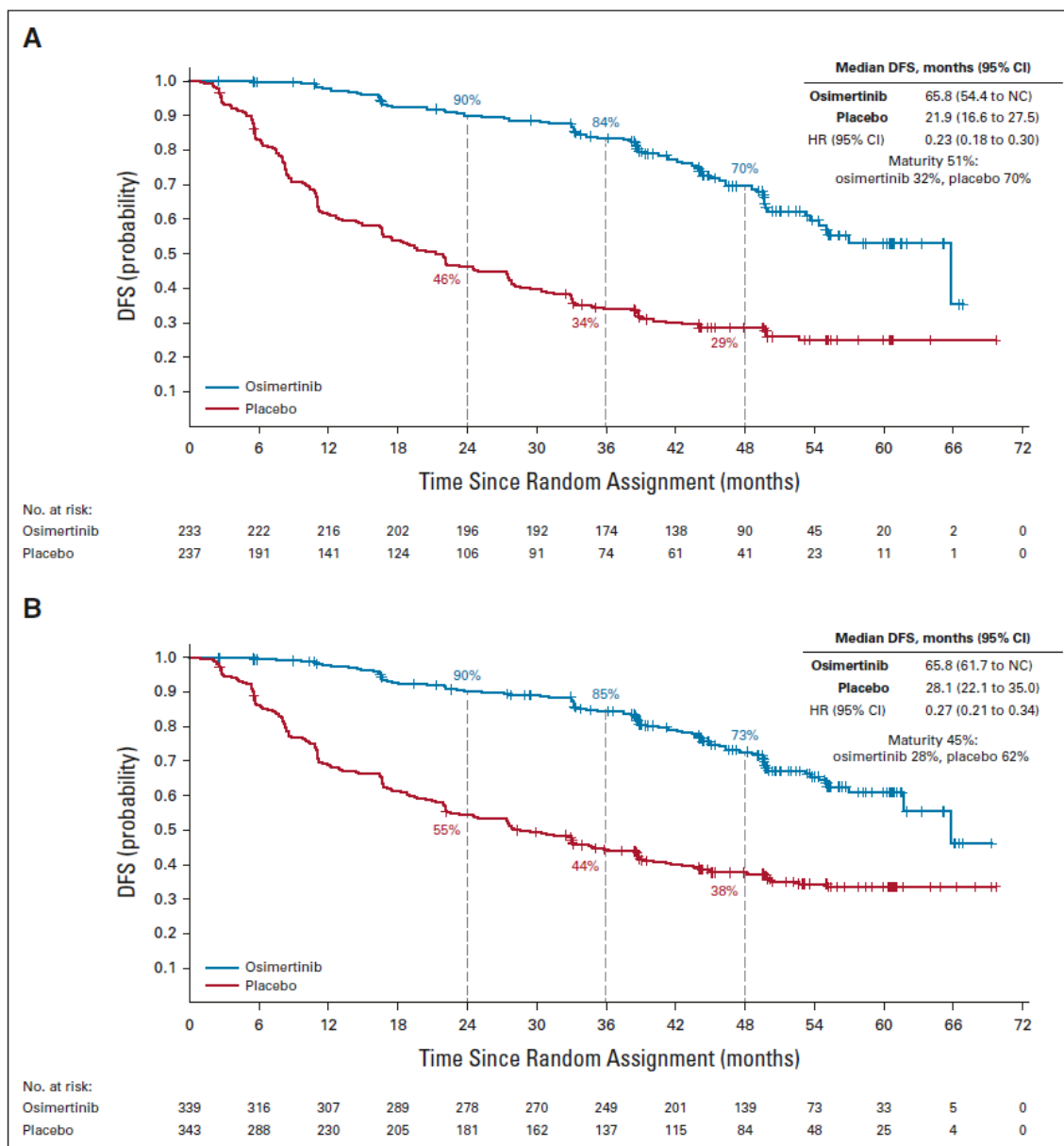


FIG 1. DFS per investigator assessment. Kaplan-Meier estimates of duration of (A) DFS in patients with stage II-IIIa disease and (B) in the overall population (stage IB-IIIa) by seventh edition staging per the protocol (full analysis set). Tick marks indicate censored data. An HR < 1 favors osimertinib. DFS, disease-free survival; HR, hazard ratio; NC, not calculated.

Herbst, Wu (21)

Figure A.5 Published Kaplan-Meier Plot for Sample Curves 10 and 11

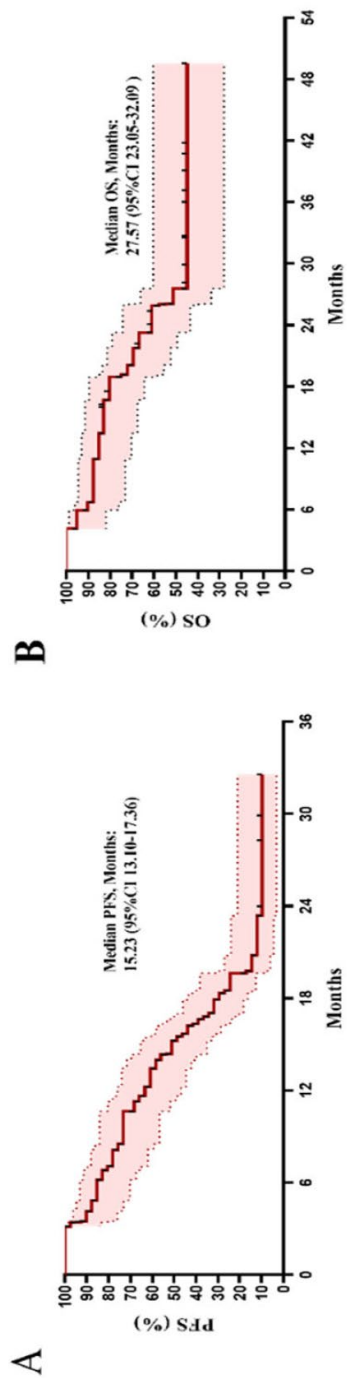


Figure 2. Survival analysis of patients in the whole group. (A) Median PFS; (B) median OS.

APPENDIX B: RAW DATA FOR AGREEMENT ANALYSIS

Table B.1. Raw Data for Agreement Analysis

curve.ID	index	MS	LS	CW
1	1	17.2542	17.1007	17.0741
2	1	18.4252	29.2136	18.3367
3	1	35.0906	35.0774	30.3607
4	1	NA	NA	54.1683
5	1	0.9545	1.0000	0.9545
6	1	0.8511	0.8916	0.8511
7	1	0.7409	0.7762	0.7409
8	1	0.5181	NA	0.5699
9	1	0.5181	NA	0.5699
10	2	17.6162	18.8277	17.5820
11	2	18.7539	19.6426	18.8730
12	2	22.8699	22.8163	22.8074
13	2	34.0539	34.0529	54.3443
14	2	5 1.0000	1.0000	1.0000
15	2	6 0.8125	0.8705	0.8077
16	2	7 0.5266	0.5682	0.5169
17	2	8 0.5266	0.5682	0.5169
18	2	9 0.4213	NA	0.5169
19	3	1 1.6509	1.6805	1.6358
20	3	2 2.5917	2.5864	2.5475
21	3	3 4.0715	4.0700	4.0492
22	3	4 5.7648	5.7768	5.5777
23	3	5 5.7648	5.7768	5.5777
24	3	6 0.5662	0.5795	0.5662
25	3	7 0.2734	0.2797	0.2734
26	3	8 0.1917	0.1978	0.1904
27	3	9 0.1704	0.1978	0.1692
28	3	10 0.1193	0.1385	NA
29	4	1 2.4631	2.5844	2.4344
30	4	2 4.6212	5.4788	4.5777
31	4	3 9.3083	9.4587	9.6714
32	4	4 13.7512	13.4450	13.7332
33	4	5 0.8000	0.8178	0.8000
34	4	6 0.7091	0.7248	0.7091
35	4	7 0.6182	0.6319	0.6182
36	4	8 0.5437	0.5558	0.5437
37	4	9 0.4687	0.4600	0.4687
38	5	1 7.3070	7.2920	7.1004
39	5	2 11.4133	11.4988	11.3974
40	5	3 14.5425	14.4936	14.5415
41	5	4 17.4421	19.9538	NA
42	5	5 0.9492	0.9655	0.9492
43	5	6 0.8804	0.8956	0.8804
44	5	7 0.7941	0.8078	0.7941
45	5	8 0.7250	0.7375	0.7250
46	5	9 0.6387	0.6497	0.6387
47	5	10 0.5477	0.5577	0.5593
48	6	1 33.6776	33.7967	33.2830
49	6	2 43.7540	44.1608	43.8792
50	6	3 53.7419	53.8629	53.3887
51	6	4 65.8643	66.4775	65.8189

52	6	5	0.9730	0.9819	0.9775
53	6	6	0.8997	0.9039	0.8999
54	6	7	0.8341	0.8379	0.8343
55	6	8	0.6924	0.6974	0.6974
56	6	9	0.5224	0.5332	0.5301
57	6	10	0.3483	0.5332	0.3446
58	7	1	8.4120	8.3426	8.2868
59	7	2	13.7778	13.2612	12.7019
60	7	3	21.9443	21.3389	21.2604
61	7	4	29.4480	29.3978	28.1208
62	7	5	42.1254	43.9569	41.0943
63	7	6	0.6202	0.6190	0.6125
64	7	7	0.4640	0.4643	0.4637
65	7	8	0.3425	0.3393	0.3401
66	7	9	0.2880	0.2851	0.2864
67	7	10	0.2465	0.2458	NA
68	7	11	0.2465	0.2458	NA
69	8	1	34.2711	34.5819	42.6087
70	8	2	44.9420	44.6272	57.5652
71	8	3	61.6872	61.8947	NA
72	8	4	65.8701	66.1132	NA
73	8	5	0.9811	0.9810	0.9873
74	8	6	0.9040	0.9007	0.9204
75	8	7	0.8440	0.8472	0.8864
76	8	8	0.7275	0.7306	0.8359
77	8	9	0.6112	0.6097	0.7193
78	8	10	0.4630	0.5628	0.7193
79	9	1	6.6937	7.5016	9.0435
80	9	2	10.8581	10.8915	13.8261
81	9	3	18.9201	19.5042	24.0000
82	9	4	28.0398	29.4394	36.4348
83	9	5	41.6046	43.5782	54.3478
84	9	6	0.6920	0.6914	0.7718
85	9	7	0.5446	0.5464	0.5973
86	9	8	0.4399	0.4412	0.5038
87	9	9	0.3770	0.3749	0.4409
88	9	10	0.3394	0.3360	0.3667
89	9	11	0.3394	0.3360	0.3667
90	10	1	8.5867	10.6294	8.5346
91	10	2	13.3960	13.9556	13.3815
92	10	3	15.2196	15.4785	15.5239
93	10	4	16.3318	16.6206	16.3317
94	10	5	17.9449	18.3037	17.8771
95	10	6	0.8780	0.8780	0.8537
96	10	7	0.6341	0.6585	0.6341
97	10	8	0.2927	0.3171	0.2927
98	10	9	0.1220	0.1220	0.0976
99	10	10	0.1220	0.1220	0.0976
100	11	1	13.4601	15.9866	13.3981
101	11	2	19.2598	20.0735	19.1547
102	11	3	25.9616	26.0534	25.8283
103	11	4	27.7074	28.1269	27.5604
104	11	5	0.9024	1.0000	0.9024
105	11	6	0.8537	0.8780	0.8537
106	11	7	0.8049	0.8049	0.8049
107	11	8	0.6098	0.6585	0.6098
108	11	9	0.4390	0.4634	NA
109	11	10	0.4390	0.4634	NA

APPENDIX C: R CODE

```

#install.packages("IPDfromKM")
library(IPDfromKM)

#####
#Curve 1
#####
#

data1 <- read.csv('Curve 1_axis.csv')
times1 <- c(0,12,24,36,48,60)
# Vector for nrisk - counts of subjects
counts1 <- c(22,16,13,5,5,3)

# Call IPDfromKM package - preprocess function
pre_IPD1 <- IPDfromKM::preprocess(data1, trisk=times1, nrisk=counts1,
                                  totalpts = 22, maxy=1)

# IPDfromKM package, getIPD fn - input preprocessed data
IPD_1 <- IPDfromKM::getIPD(pre_IPD1)
#print(IPD_1)
#summary(IPD_1)
#plot(IPD_1)

# survreport fn -
report1 <- IPDfromKM::survreport(ipd1 = IPD_1$IPD,arms=1,interval = 6,
                                 s=c(0.85,0.75,0.6,0.5),showplots = TRUE)

surv.1 <- c(report1$arm1$survtime$time,
            report1$arm1$survprob$Surv[c(1,2,4,6,8)])
surv.1
#
#
#####
#Curve 2
#####
#

data2 <- read.csv('Curve 2_axis edited.csv')
times2 <- c(0,12,24,36,48,60)
# Vector for nrisk - counts of subjects
counts2 <- c(22,16,7,4,4,2)

# Call IPDfromKM package - preprocess function
pre_IPD2 <- IPDfromKM::preprocess(data2, trisk=times2, nrisk=counts2,
                                  totalpts = 22, maxy=1)

# IPDfromKM package, getIPD fn - input preprocessed data
IPD_2 <- IPDfromKM::getIPD(pre_IPD2)
#summary(IPD_2)

```

```

#plot(IPD_2)

# survreport fn -
report2 <- IPDfromKM::survreport(ipd1 = IPD_2$IPD,arms=1,interval = 6,
                                s=c(0.85,0.75,0.6,0.5),showplots = TRUE)

surv.2 <- c(report2$arm1$survtime$time,
            report2$arm1$survprob$Surv[c(2:5,8)])
surv.2
#
#
#####
#Curve 3
#####
#

data3 <- read.csv('Curve 3_.csv')
times3 <- c(0,3,6,9,12,15)
# Vector for nrisk - counts of subjects
counts3 <- c(55,29,15,9,8,4)

# Call IPDfromKM package - preprocess function
pre_IPD3 <- IPDfromKM::preprocess(data3, trisk=times3, nrisk=counts3,
                                  totalpts = 55, maxy=1)

# IPDfromKM package, getIPD fn - input preprocessed data
IPD_3 <- IPDfromKM::getIPD(pre_IPD3)
#summary(IPD_3)
#plot(IPD_3)

# survreport fn -
report3 <- IPDfromKM::survreport(ipd1 = IPD_3$IPD,arms=1,interval = 3,
                                s=c(.75,.6,.5,.4,.3),showplots = TRUE)

surv.3 <- c(report3$arm1$survtime$time,
            report3$arm1$survprob$Surv[c(1:5)])
surv.3
#
#
#####
#Curve 4
#####
#

data4 <- read.csv('Curve 4_.csv')
times4 <- c(0,6,12,18,24)
# Vector for nrisk - counts of subjects
counts4 <- c(55,39,29,0,0)

# Call IPDfromKM package - preprocess function
pre_IPD4 <- IPDfromKM::preprocess(data4, trisk=times4, nrisk=counts4,
                                  totalpts = 55, maxy=1)

# IPDfromKM package, getIPD fn - input preprocessed data
IPD_4 <- IPDfromKM::getIPD(pre_IPD4)

```

```

#summary(IPD_4)
#plot(IPD_4)

# survreport fn -
report4 <- IPDfromKM::survreport(ipd1 = IPD_4$IPD,arms=1,interval = 3,
                                s=c(0.85,0.75,0.6,0.5),showplots = TRUE)

surv.4 <- c(report4$arm1$survtime$time,
            report4$arm1$survprob$Surv[c(1:5)])
surv.4
#
#
#####
#Curve 5
#####
#

data5 <- read.csv('Curve 5_axis edited.csv')
times5 <- c(0,3,6,9,12,15,18,21,24,27)
# Vector for nrisk - counts of subjects
counts5 <- c(59,56,51,46,42,37,29,16,4,0)

# Call IPDfromKM package - preprocess function
pre_IPD5 <- IPDfromKM::preprocess(data5, trisk=times5, nrisk=counts5,
                                totalpts = 59, maxy=1)

# IPDfromKM package, getIPD fn - input preprocessed data
IPD_5 <- IPDfromKM::getIPD(pre_IPD5)
#summary(IPD_5)
#plot(IPD_5)

# survreport fn -
report5 <- IPDfromKM::survreport(ipd1 = IPD_5$IPD,arms=1,interval = 3,
                                s=c(0.85,0.75,0.65,0.55),showplots = TRUE)

surv.5 <- c(report5$arm1$survtime$time,
            report5$arm1$survprob$Surv[c(1:6)])
surv.5
#
#
#####
#Curve 6
#####
#

data6 <- read.csv('Curve 6_.csv')
times6 <- c(0,6,12,18,24,30,36,42,48,54,60,66,72)
# Vector for nrisk - counts of subjects
counts6 <- c(233,222,216,202,196,192,174,138,90,45,20,2,0)

# Call IPDfromKM package - preprocess function
pre_IPD6 <- IPDfromKM::preprocess(data6, trisk=times6, nrisk=counts6,
                                totalpts = 233, maxy=1)

# IPDfromKM package, getIPD fn - input preprocessed data

```

```

IPD_6 <- IPDfromKM::getIPD(pre_IPD6)
#summary(IPD_6)
#plot(IPD_6)

# survreport fn -
report6 <- IPDfromKM::survreport(ipd1 = IPD_6$IPD,arms=1,interval = 6,
                                s=c(0.85,0.75,0.6,0.5),showplots = TRUE)

surv.6 <- c(report6$arm1$survtime$time,
            report6$arm1$survprob$Surv[c(2,4,6,8,10,11)])
surv.6
#
#
#####
#Curve 7
#####
#

data7 <- read.csv('Curve 7_.csv')
# Vector for nrisk - counts of subjects
counts7 <- c(237,191,141,124,106,91,74,61,41,23,11,1,0)

# Call IPDfromKM package - preprocess function
pre_IPD7 <- IPDfromKM::preprocess(data7, trisk=times6, nrisk=counts7,
                                totalpts = 237, maxy=1)

# IPDfromKM package, getIPD fn - input preprocessed data
IPD_7 <- IPDfromKM::getIPD(pre_IPD7)
#summary(IPD_7)
#plot(IPD_7)

# survreport fn -
report7 <- IPDfromKM::survreport(ipd1 = IPD_7$IPD,arms=1,interval = 6,
                                s=c(0.75,0.6,0.5,.4,.3),showplots = TRUE)

surv.7 <- c(report7$arm1$survtime$time,
            report7$arm1$survprob$Surv[c(2,4,6,8,10,11)])
surv.7
#
#
#####
#Curve 8
#####
#

data8 <- read.csv('Curve 8_.csv')
# Vector for nrisk - counts of subjects
counts8 <- c(339,316,307,289,278,270,249,201,139,73,33,5,0)

# Call IPDfromKM package - preprocess function
pre_IPD8 <- IPDfromKM::preprocess(data8, trisk=times6, nrisk=counts8,
                                totalpts = 339, maxy=1)

# IPDfromKM package, getIPD fn - input preprocessed data
IPD_8 <- IPDfromKM::getIPD(pre_IPD8)

```

```

#summary(IPD_8)
#plot(IPD_8)

# survreport fn -
report8 <- IPDfromKM::survreport(ipd1 = IPD_8$IPD,arms=1,interval = 6,
                                s=c(.85,.75,.6,.5),showplots = TRUE)

surv.8 <- c(report8$arm1$survtime$time,
            report8$arm1$survprob$Surv[c(2,4,6,8,10,11)])
surv.8
#
#####
#Curve 9
#####
#

data9 <- read.csv('Curve_9_.csv')
# Vector for nrisk - counts of subjects
counts9 <- c(343,288,230,205,181,162,137,115,84,48,25,4,0)

# Call IPDfromKM package - preprocess function
pre_IPD9 <- IPDfromKM::preprocess(data9, trisk=times6, nrisk=counts9,
                                  totalpts = 343, maxy=1)

# IPDfromKM package, getIPD fn - input preprocessed data
IPD_9 <- IPDfromKM::getIPD(pre_IPD9)
#summary(IPD_9)
#plot(IPD_9)

# survreport fn -
report9 <- IPDfromKM::survreport(ipd1 = IPD_9$IPD,arms=1,interval = 6,
                                s=c(.85,.75,.6,.5,.4),showplots = TRUE)

surv.9 <- c(report9$arm1$survtime$time,
            report9$arm1$survprob$Surv[c(2,4,6,8,10,11)])
surv.9
#
#
#####
#Curve 10
#####
#

data10 <- read.csv('Curve_10.csv')
# No number at risk provided on graph

# Call IPDfromKM package - preprocess function
pre_IPD10 <- IPDfromKM::preprocess(data10, totalpts = 41, maxy=100)

# IPDfromKM package, getIPD fn - input preprocessed data
IPD_10 <- IPDfromKM::getIPD(pre_IPD10)
##summary(IPD_10)
#plot(IPD_10)

# survreport fn -

```



```

report10 <- IPDfromKM::survreport(ipd1 = IPD_10$IPD,arms=1,interval = 6,
                                s=c(.75,.6,.5,.4,.3),showplots = TRUE)

surv.10 <- c(report10$arm1$survtime$time,
             report10$arm1$survprob$Surv[c(1:5)])

surv.10
#
#
#####
#Curve 11
#####
#
data11 <- read.csv('Curve 11.csv')
# No number at risk provided on graph
# Call IPDfromKM package - preprocess function
pre_IPD11 <- IPDfromKM::preprocess(data11, totalpts = 41, maxy=100)

# IPDfromKM package, getIPD fn - input preprocessed data
IPD_11 <- IPDfromKM::getIPD(pre_IPD11)
#summary(IPD_11)
#plot(IPD_11)
# survreport fn -
report11 <- IPDfromKM::survreport(ipd1 = IPD_11$IPD,arms=1,interval = 6,
                                s=c(.85,.75,.6,.5),showplots = TRUE)
surv.11 <- c(report11$arm1$survtime$time,
             report11$arm1$survprob$Surv[c(1:4,6,8)])

surv.11
#
#
#####
#
##### Lynette Smith
#####
#Curve 1
#####
#

data1.LS <- read.csv('Curve 1 LS.csv')
pre_IPD1.LS <- IPDfromKM::preprocess(data1.LS, trisk=times1, nrisk=counts1,
                                    totalpts = 22, maxy=1)
IPD_1.LS <- IPDfromKM::getIPD(pre_IPD1.LS)
report1.LS <- IPDfromKM::survreport(ipd1 = IPD_1.LS$IPD,arms=1,interval = 6,
                                   s=c(0.85,0.75,0.6,0.5),showplots = T)
surv.1.LS <- c(report1.LS$arm1$survtime$time,
              report1.LS$arm1$survprob$Surv[c(1,2,4,6,8)])

surv.1.LS
#
#
#####
#Curve 2
#####
#

data2.LS <- read.csv('Curve 2 LS.csv')
pre_IPD2.LS <- IPDfromKM::preprocess(data2.LS, trisk=times2, nrisk=counts2,

```

```

                                totalpts = 22, maxy=1)
IPD_2.LS <- IPDfromKM::getIPD(pre_IPD2.LS)

report2.LS <- IPDfromKM::survreport(ipd1 = IPD_2.LS$IPD,arms=1,interval = 6,
                                s=c(0.85,0.75,0.6,0.5),showplots = TRUE)
surv.2.LS <- c(report2.LS$arm1$survtime$time,
              report2.LS$arm1$survprob$Surv[c(2:5,8)])
surv.2.LS
#
#
#####
#Curve 3
#####
#

data3.LS <- read.csv('Curve 3 LS.csv')
pre_IPD3.LS <- IPDfromKM::preprocess(data3.LS, trisk=times3, nrisk=counts3,
                                totalpts = 55, maxy=1)
IPD_3.LS <- IPDfromKM::getIPD(pre_IPD3.LS)

report3.LS <- IPDfromKM::survreport(ipd1 = IPD_3.LS$IPD,arms=1,interval = 3,
                                s=c(.75,.6,.5,.4,.3),showplots = TRUE)
surv.3.LS <- c(report3.LS$arm1$survtime$time,
              report3.LS$arm1$survprob$Surv[c(1:5)])
surv.3.LS
#
#
#####
#Curve 4
#####
#

data4.LS <- read.csv('Curve 4 LS.csv')
pre_IPD4.LS <- IPDfromKM::preprocess(data4.LS, trisk=times4, nrisk=counts4,
                                totalpts = 55, maxy=1)
IPD_4.LS <- IPDfromKM::getIPD(pre_IPD4.LS)

report4.LS <- IPDfromKM::survreport(ipd1 = IPD_4.LS$IPD,arms=1,interval = 3,
                                s=c(0.85,0.75,0.6,0.5),showplots = TRUE)
surv.4.LS <- c(report4.LS$arm1$survtime$time,
              report4.LS$arm1$survprob$Surv[c(1:5)])
surv.4.LS
#
#
#####
#Curve 5
#####
#

data5.LS <- read.csv('Curve 5 LS.csv')
pre_IPD5.LS <- IPDfromKM::preprocess(data5.LS, trisk=times5, nrisk=counts5,
                                totalpts = 59, maxy=1)
IPD_5.LS <- IPDfromKM::getIPD(pre_IPD5.LS)

report5.LS <- IPDfromKM::survreport(ipd1 = IPD_5.LS$IPD,arms=1,interval = 3,

```

```

                                s=c(0.85,0.75,0.65,0.55),showplots = TRUE)
surv.5.LS <- c(report5.LS$arm1$survtime$time,
              report5.LS$arm1$survprob$Surv[c(1:6)])
surv.5.LS
#
#
#####
#Curve 6
#####
#

data6.LS <- read.csv('Curve 6 LS.csv')
pre_IPD6.LS <- IPDfromKM::preprocess(data6.LS, trisk=times6, nrisk=counts6,
                                   totalpts = 233, maxy=1)
IPD_6.LS <- IPDfromKM::getIPD(pre_IPD6.LS)

report6.LS <- IPDfromKM::survreport(ipd1 = IPD_6.LS$IPD,arms=1,interval = 6,
                                   s=c(0.85,0.75,0.6,0.5),showplots = TRUE)
surv.6.LS <- c(report6.LS$arm1$survtime$time,
              report6.LS$arm1$survprob$Surv[c(2,4,6,8,10,11)])
surv.6.LS
#
#
#####
#Curve 7
#####
#

data7.LS <- read.csv('Curve 7 LS.csv')
pre_IPD7.LS <- IPDfromKM::preprocess(data7.LS, trisk=times6, nrisk=counts7,
                                   totalpts = 237, maxy=1)
IPD_7.LS <- IPDfromKM::getIPD(pre_IPD7.LS)

report7.LS <- IPDfromKM::survreport(ipd1 = IPD_7.LS$IPD,arms=1,interval = 6,
                                   s=c(0.75,0.6,0.5,.4,.3),showplots = TRUE)
surv.7.LS <- c(report7.LS$arm1$survtime$time,
              report7.LS$arm1$survprob$Surv[c(2,4,6,8,10,11)])
surv.7.LS
#
#
#
#
#####
#Curve 8
#####
#

data8.LS <- read.csv('Curve 8 LS.csv')
pre_IPD8.LS <- IPDfromKM::preprocess(data8.LS, trisk=times6, nrisk=counts8,
                                   totalpts = 339, maxy=1)
IPD_8.LS <- IPDfromKM::getIPD(pre_IPD8.LS)

report8.LS <- IPDfromKM::survreport(ipd1 = IPD_8.LS$IPD,arms=1,interval = 6,
                                   s=c(.85,.75,.6,.5),showplots = TRUE)
surv.8.LS <- c(report8.LS$arm1$survtime$time,

```

```

report8.LS$arm1$survprob$Surv[c(2,4,6,8,10,11)])
surv.8.LS
#
#
#####
#Curve 9
#####
#

data9.LS <- read.csv('Curve 9 LS.csv')
pre_IPD9.LS <- IPDfromKM::preprocess(data9.LS, trisk=times6, nrisk=counts9,
                                     totalpts = 343, maxy=1)
IPD_9.LS <- IPDfromKM::getIPD(pre_IPD9.LS)

report9.LS <- IPDfromKM::survreport(ipd1 = IPD_9.LS$IPD,arms=1,interval = 6,
                                   s=c(.85,.75,.6,.5,.4),showplots = TRUE)
surv.9.LS <- c(report9.LS$arm1$survtime$time,
               report9.LS$arm1$survprob$Surv[c(2,4,6,8,10,11)])
surv.9.LS
#
#
#####
#Curve 10
#####
#

data10.LS <- read.csv('Curve 10 LS.csv')
pre_IPD10.LS <- IPDfromKM::preprocess(data10.LS, totalpts = 41, maxy=100)
IPD_10.LS <- IPDfromKM::getIPD(pre_IPD10.LS)

report10.LS <- IPDfromKM::survreport(ipd1 = IPD_10.LS$IPD,arms=1,interval = 6,
                                    s=c(.75,.6,.5,.4,.3),showplots = TRUE)
surv.10.LS <- c(report10.LS$arm1$survtime$time,
                report10.LS$arm1$survprob$Surv[c(1:5)])
surv.10.LS
#
#
#####
#Curve 11
#####
#

data11.LS <- read.csv('Curve 11 LS.csv')
pre_IPD11.LS <- IPDfromKM::preprocess(data11.LS, totalpts = 41, maxy=100)
IPD_11.LS <- IPDfromKM::getIPD(pre_IPD11.LS)

report11.LS <- IPDfromKM::survreport(ipd1 = IPD_11.LS$IPD,arms=1,interval = 6,
                                    s=c(.85,.75,.6,.5),showplots = TRUE)

surv.11.LS <- c(report11.LS$arm1$survtime$time,
                report11.LS$arm1$survprob$Surv[c(1:4,6,8)])
surv.11.LS
#
#

```

```
#####
#
##### Chris Wichman
#####
#Curve 1
#####
#
data1.CW <- read.csv('Curve01 CSW data.csv')
pre_IPD1.CW <- IPDfromKM::preprocess(data1.CW, trisk=times1, nrisk=counts1,
                                     totalpts = 22, maxy=1)
IPD_1.CW <- IPDfromKM::getIPD(pre_IPD1.CW)

report1.CW <- IPDfromKM::survreport(ipd1 = IPD_1.CW$IPD,arms=1,interval = 6,
                                   s=c(0.85,0.75,0.6,0.5),showplots = T)
surv.1.CW <- c(report1.CW$arm1$survtime$time,
               report1.CW$arm1$survprob$Surv[c(1,2,4,6,8)])
surv.1.CW
#
#
#####
#Curve 2
#####
#
data2.CW <- read.csv('Curve02 CSW data.csv')
pre_IPD2.CW <- IPDfromKM::preprocess(data2.CW, trisk=times2, nrisk=counts2,
                                     totalpts = 22, maxy=1)
IPD_2.CW <- IPDfromKM::getIPD(pre_IPD2.CW)

report2.CW <- IPDfromKM::survreport(ipd1 = IPD_2.CW$IPD,arms=1,interval = 6,
                                   s=c(0.85,0.75,0.6,0.5),showplots = TRUE)
surv.2.CW <- c(report2.CW$arm1$survtime$time,
               report2.CW$arm1$survprob$Surv[c(2:5,8)])
surv.2.CW
#
#
#####
#Curve 3
#####
#
data3.CW <- read.csv('Curve03 CSW data.csv')
pre_IPD3.CW <- IPDfromKM::preprocess(data3.CW, trisk=times3, nrisk=counts3,
                                     totalpts = 55, maxy=1)
IPD_3.CW <- IPDfromKM::getIPD(pre_IPD3.CW)

report3.CW <- IPDfromKM::survreport(ipd1 = IPD_3.CW$IPD,arms=1,interval = 3,
                                   s=c(.75,.6,.5,.4,.3),showplots = TRUE)
surv.3.CW <- c(report3.CW$arm1$survtime$time,
               report3.CW$arm1$survprob$Surv[c(1:5)])
surv.3.CW
#
#
#####
#Curve 4
```

```
#####
#

data4.CW <- read.csv('Curve04 CSW data.csv')
pre_IPD4.CW <- IPDfromKM::preprocess(data4.CW, trisk=times4, nrisk=counts4,
                                     totalpts = 55, maxy=1)
IPD_4.CW <- IPDfromKM::getIPD(pre_IPD4.CW)

report4.CW <- IPDfromKM::survreport(ipd1 = IPD_4.CW$IPD,arms=1,interval = 3,
                                   s=c(0.85,0.75,0.6,0.5),showplots = TRUE)
surv.4.CW <- c(report4.CW$arm1$survtime$time,
               report4.CW$arm1$survprob$Surv[c(1:5)])
surv.4.CW
#
#
#####
#Curve 5
#####
#

data5.CW <- read.csv('Curve05 CSW data.csv')
pre_IPD5.CW <- IPDfromKM::preprocess(data5.CW, trisk=times5, nrisk=counts5,
                                     totalpts = 59, maxy=1)
IPD_5.CW <- IPDfromKM::getIPD(pre_IPD5.CW)

report5.CW <- IPDfromKM::survreport(ipd1 = IPD_5.CW$IPD,arms=1,interval = 3,
                                   s=c(0.85,0.75,0.65,0.55),showplots = TRUE)
surv.5.CW <- c(report5.CW$arm1$survtime$time,
               report5.CW$arm1$survprob$Surv[c(1:6)])
surv.5.CW
#
#
#####
#Curve 6
#####
#

data6.CW <- read.csv('Curve06 CSW osimertinib data.csv')
pre_IPD6.CW <- IPDfromKM::preprocess(data6.CW, trisk=times6, nrisk=counts6,
                                     totalpts = 233, maxy=1)
IPD_6.CW <- IPDfromKM::getIPD(pre_IPD6.CW)

report6.CW <- IPDfromKM::survreport(ipd1 = IPD_6.CW$IPD,arms=1,interval = 6,
                                   s=c(0.85,0.75,0.6,0.5),showplots = TRUE)
surv.6.CW <- c(report6.CW$arm1$survtime$time,
               report6.CW$arm1$survprob$Surv[c(2,4,6,8,10,11)])
surv.6.CW
#
#
#####
#Curve 7
#####
#

data7.CW <- read.csv('Curve07 CSW placebo data.csv')
```

```

pre_IPD7.CW <- IPDfromKM::preprocess(data7.CW, trisk=times6, nrisk=counts7,
                                     totalpts = 237, maxy=1)
IPD_7.CW <- IPDfromKM::getIPD(pre_IPD7.CW)

report7.CW <- IPDfromKM::survreport(ipd1 = IPD_7.CW$IPD,arms=1,interval = 6,
                                   s=c(0.75,0.6,0.5,.4,.3),showplots = TRUE)
surv.7.CW <- c(report7.CW$arm1$survtime$time,
               report7.CW$arm1$survprob$Surv[c(2,4,6,8,10,11)])

surv.7.CW
#
#
#####
#Curve 8
#####
#

data8.CW <- read.csv('Curve08 CSW data.csv')
pre_IPD8.CW <- IPDfromKM::preprocess(data8.CW, trisk=times6, nrisk=counts8,
                                     totalpts = 339, maxy=1)
IPD_8.CW <- IPDfromKM::getIPD(pre_IPD8.CW)
plot(IPD_8.CW)
report8.CW <- IPDfromKM::survreport(ipd1 = IPD_8.CW$IPD,arms=1,interval = 6,
                                   s=c(.85,.75,.6,.5),showplots = TRUE)
surv.8.CW <- c(report8.CW$arm1$survtime$time,
               report8.CW$arm1$survprob$Surv[c(2,4,6,8,10,11)])

surv.8.CW
#
#
#####
#Curve 9
#####
#

data9.CW <- read.csv('Curve09 CSW data.csv')
pre_IPD9.CW <- IPDfromKM::preprocess(data9.CW, trisk=times6, nrisk=counts9,
                                     totalpts = 343, maxy=1)
IPD_9.CW <- IPDfromKM::getIPD(pre_IPD9.CW)

report9.CW <- IPDfromKM::survreport(ipd1 = IPD_9.CW$IPD,arms=1,interval = 6,
                                   s=c(.85,.75,.6,.5,.4),showplots = TRUE)
surv.9.CW <- c(report9.CW$arm1$survtime$time,
               report9.CW$arm1$survprob$Surv[c(2,4,6,8,10,11)])

surv.9.CW
#
#
#####
#Curve 10
#####
#

data10.CW <- read.csv('Curve10 CSW data.csv')
pre_IPD10.CW <- IPDfromKM::preprocess(data10.CW, totalpts = 41, maxy=100)
IPD_10.CW <- IPDfromKM::getIPD(pre_IPD10.CW)

report10.CW <- IPDfromKM::survreport(ipd1 = IPD_10.CW$IPD,arms=1,interval = 6,

```

```

                                s=c(.75,.6,.5,.4,.3),showplots = TRUE)
surv.10.CW <- c(report10.CW$arm1$survtime$time,
               report10.CW$arm1$survprob$Surv[c(1:5)])
surv.10.CW
#
#
#####
#Curve 11
#####
#

data11.CW <- read.csv('Curve11 CSW data.csv')
pre_IPD11.CW <- IPDfromKM::preprocess(data11.CW, totalpts = 41, maxy=100)
IPD_11.CW <- IPDfromKM::getIPD(pre_IPD11.CW)

report11.CW <- IPDfromKM::survreport(ipd1 = IPD_11.CW$IPD,arms=1,interval = 6,
                                   s=c(.85,.75,.6,.5),showplots = TRUE)

surv.11.CW <- c(report11.CW$arm1$survtime$time,
               report11.CW$arm1$survprob$Surv[c(1:4,6,8)])
surv.11.CW
#
#
#####
#
##### ICC
#####
#
# dataframe for all raw reconstructed values

df.all <- data.frame(
  curve.ID = c(rep(1,length(surv.1)), rep(2,length(surv.2)),
              rep(3,length(surv.3)), rep(4,length(surv.4)),
              rep(5,length(surv.5)), rep(6,length(surv.6)),
              rep(7,length(surv.7)), rep(8,length(surv.8)),
              rep(9,length(surv.9)), rep(10,length(surv.10)),
              rep(11,length(surv.11))),
  index = c(1:length(surv.1), 1:length(surv.2), 1:length(surv.3),
            1:length(surv.4), 1:length(surv.5), 1:length(surv.6),
            1:length(surv.7), 1:length(surv.8), 1:length(surv.9),
            1:length(surv.10), 1:length(surv.11)),
  MS = c(surv.1, surv.2, surv.3, surv.4, surv.5, surv.6, surv.7,
        surv.8, surv.9, surv.10, surv.11),
  LS = c(surv.1.LS, surv.2.LS, surv.3.LS, surv.4.LS, surv.5.LS, surv.6.LS,
        surv.7.LS, surv.8.LS, surv.9.LS, surv.10.LS, surv.11.LS),
  CW = c(surv.1.CW, surv.2.CW, surv.3.CW, surv.4.CW, surv.5.CW, surv.6.CW,
        surv.7.CW, surv.8.CW, surv.9.CW, surv.10.CW, surv.11.CW)
)
#####
#
##### Intraclass Correlation Coefficient
#####
#install.packages("irr")
require(irr)

```



```

# Subgroups
# Survival time estimates
times.MS <- c(report1$arm1$survtime$time, report2$arm1$survtime$time,
              report3$arm1$survtime$time, report4$arm1$survtime$time,
              report5$arm1$survtime$time, report6$arm1$survtime$time,
              report7$arm1$survtime$time, report8$arm1$survtime$time,
              report9$arm1$survtime$time, report10$arm1$survtime$time,
              report11$arm1$survtime$time)

times.LS <- c(report1.LS$arm1$survtime$time, report2.LS$arm1$survtime$time,
              report3.LS$arm1$survtime$time, report4.LS$arm1$survtime$time,
              report5.LS$arm1$survtime$time, report6.LS$arm1$survtime$time,
              report7.LS$arm1$survtime$time, report8.LS$arm1$survtime$time,
              report9.LS$arm1$survtime$time, report10.LS$arm1$survtime$time,
              report11.LS$arm1$survtime$time)

times.CW <- c(report1.CW$arm1$survtime$time, report2.CW$arm1$survtime$time,
              report3.CW$arm1$survtime$time, report4.CW$arm1$survtime$time,
              report5.CW$arm1$survtime$time, report6.CW$arm1$survtime$time,
              report7.CW$arm1$survtime$time, report8.CW$arm1$survtime$time,
              report9.CW$arm1$survtime$time, report10.CW$arm1$survtime$time,
              report11.CW$arm1$survtime$time)

# Dataframe for ALL survival time estimates
all.times <- data.frame(times.MS,times.LS,times.CW)
sum(is.na(all.times))

#####
#
##### Collate data
#####
#install.packages("irr")
#install.packages("lme4")
#install.packages("misty")
install.packages("psychometric")
require(irr)
require(lme4)
require(misty)
require(psychometric)

# Subgroups
# Survival time estimates
times.MS <- c(report1$arm1$survtime$time, report2$arm1$survtime$time,
              report3$arm1$survtime$time, report4$arm1$survtime$time,
              report5$arm1$survtime$time, report6$arm1$survtime$time,
              report7$arm1$survtime$time, report8$arm1$survtime$time,
              report9$arm1$survtime$time, report10$arm1$survtime$time,
              report11$arm1$survtime$time)

times.LS <- c(report1.LS$arm1$survtime$time, report2.LS$arm1$survtime$time,
              report3.LS$arm1$survtime$time, report4.LS$arm1$survtime$time,
              report5.LS$arm1$survtime$time, report6.LS$arm1$survtime$time,
              report7.LS$arm1$survtime$time, report8.LS$arm1$survtime$time,
              report9.LS$arm1$survtime$time, report10.LS$arm1$survtime$time,

```

```

report11.LS$arm1$survtime$time)

times.CW <- c(report1.CW$arm1$survtime$time, report2.CW$arm1$survtime$time,
report3.CW$arm1$survtime$time, report4.CW$arm1$survtime$time,
report5.CW$arm1$survtime$time, report6.CW$arm1$survtime$time,
report7.CW$arm1$survtime$time, report8.CW$arm1$survtime$time,
report9.CW$arm1$survtime$time, report10.CW$arm1$survtime$time,
report11.CW$arm1$survtime$time)

ID.times <- c(rep(1,length(report1$arm1$survtime$time)),
rep(2,length(report2$arm1$survtime$time)),
rep(3,length(report3$arm1$survtime$time)),
rep(4,length(report4$arm1$survtime$time)),
rep(5,length(report5$arm1$survtime$time)),
rep(6,length(report6$arm1$survtime$time)),
rep(7,length(report7$arm1$survtime$time)),
rep(8,length(report8$arm1$survtime$time)),
rep(9,length(report9$arm1$survtime$time)),
rep(10,length(report10$arm1$survtime$time)),
rep(11,length(report11$arm1$survtime$time)))

# Dataframe for ALL survival time estimates
all.times <- data.frame(times.MS,times.LS,times.CW)

# Survival probability estimates

prob.MS <- c(report1$arm1$survprob$Surv[c(1,2,4,6,8)],
report2$arm1$survprob$Surv[c(2:5,8)],
report3$arm1$survprob$Surv[c(1:5)],
report4$arm1$survprob$Surv[c(1:5)],
report5$arm1$survprob$Surv[c(1:6)],
report6$arm1$survprob$Surv[c(2,4,6,8,10,11)],
report7$arm1$survprob$Surv[c(2,4,6,8,10,11)],
report8$arm1$survprob$Surv[c(2,4,6,8,10,11)],
report9$arm1$survprob$Surv[c(2,4,6,8,10,11)],
report10$arm1$survprob$Surv[c(1:5)],
report11$arm1$survprob$Surv[c(1:4,6,8)])

prob.LS <- c(report1.LS$arm1$survprob$Surv[c(1,2,4,6,8)],
report2.LS$arm1$survprob$Surv[c(2:5,8)],
report3.LS$arm1$survprob$Surv[c(1:5)],
report4.LS$arm1$survprob$Surv[c(1:5)],
report5.LS$arm1$survprob$Surv[c(1:6)],
report6.LS$arm1$survprob$Surv[c(2,4,6,8,10,11)],
report7.LS$arm1$survprob$Surv[c(2,4,6,8,10,11)],
report8.LS$arm1$survprob$Surv[c(2,4,6,8,10,11)],
report9.LS$arm1$survprob$Surv[c(2,4,6,8,10,11)],
report10.LS$arm1$survprob$Surv[c(1:5)],
report11.LS$arm1$survprob$Surv[c(1:4,6,8)])

prob.CW <- c(report1.CW$arm1$survprob$Surv[c(1,2,4,6,8)],
report2.CW$arm1$survprob$Surv[c(2:5,8)],
report3.CW$arm1$survprob$Surv[c(1:5)],
report4.CW$arm1$survprob$Surv[c(1:5)],
report5.CW$arm1$survprob$Surv[c(1:6)],

```

```

report6.CW$arm1$urvprob$Surv[c(2,4,6,8,10,11)],
report7.CW$arm1$urvprob$Surv[c(2,4,6,8,10,11)],
report8.CW$arm1$urvprob$Surv[c(2,4,6,8,10,11)],
report9.CW$arm1$urvprob$Surv[c(2,4,6,8,10,11)],
report10.CW$arm1$urvprob$Surv[c(1:5)],
report11.CW$arm1$urvprob$Surv[c(1:4,6,8)])

ID.prob <- c(rep(1,5), rep(2,5), rep(3,5), rep(4,5), rep(5,6), rep(6,6),
            rep(7,6), rep(8,6), rep(9,6), rep(10,5), rep(11,6))

#####
##### NEW ICC
#####

m.icc <- data.frame(times = c(times.MS,times.LS,times.CW),
                   rater = c(rep(1,length(times.MS)),
                              rep(2,length(times.LS)),
                              rep(3, length(times.CW))),
                   curveID = rep(ID.times,3))
m.icc.prob <- data.frame(rate = c(prob.MS, prob.LS, prob.CW),
                        rater = c(rep(1,length(prob.MS)),
                                   rep(2,length(prob.LS)),
                                   rep(3, length(prob.CW))),
                        curveID = rep(ID.prob,3))

# NEW ICC
ICC2.lme(times, grp = curveID, data = m.icc)
ICC2.CI(times, iv =curveID, data = m.icc)

# NEW ICC - probabilities
ICC2.lme(rate, grp = curveID, data = m.icc.prob)
ICC2.CI(rate, iv =curveID, data = m.icc.prob)

##### Reported benchmarks - ALL
#####

# Median survival times

rep.times.MS <- c(report1$arm1$urvtime$time[4],
                 report2$arm1$urvtime$time[4],
                 report3$arm1$urvtime$time[3], report4$arm1$urvtime$time[4],
                 report6$arm1$urvtime$time[4], report7$arm1$urvtime$time[3],
                 report8$arm1$urvtime$time[4], report9$arm1$urvtime$time[4],
                 report10$arm1$urvtime$time[3], report11$arm1$urvtime$time[4])

rep.times.LS <- c(report1.LS$arm1$urvtime$time[4],
                 report2.LS$arm1$urvtime$time[4],
                 report3.LS$arm1$urvtime$time[3],
                 report4.LS$arm1$urvtime$time[4],
                 report6.LS$arm1$urvtime$time[4],
                 report7.LS$arm1$urvtime$time[3],

```

```

        report8.LS$arm1$survtime$time[4],
report9.LS$arm1$survtime$time[4],
        report10.LS$arm1$survtime$time[3],
report11.LS$arm1$survtime$time[4])

rep.times.CW <- c(report1.CW$arm1$survtime$time[4],
report2.CW$arm1$survtime$time[4],
        report3.CW$arm1$survtime$time[3],
report4.CW$arm1$survtime$time[4],
        report6.CW$arm1$survtime$time[4],
report7.CW$arm1$survtime$time[3],
        report8.CW$arm1$survtime$time[4],
report9.CW$arm1$survtime$time[4],
        report10.CW$arm1$survtime$time[3],
report11.CW$arm1$survtime$time[4])

# probabilities

rep.prob.MS <- c(report1$arm1$survprob$Surv[c(2,4,6)],
        report2$arm1$survprob$Surv[c(2,4,6)],
        report3$arm1$survprob$Surv[4],
        report4$arm1$survprob$Surv[4],
        report5$arm1$survprob$Surv[4],
        report6$arm1$survprob$Surv[c(4,6,8)],
        report7$arm1$survprob$Surv[c(4,6,8)],
        report8$arm1$survprob$Surv[c(4,6,8)],
        report9$arm1$survprob$Surv[c(4,6,8)])

rep.prob.LS <- c(report1.LS$arm1$survprob$Surv[c(2,4,6)],
        report2.LS$arm1$survprob$Surv[c(2,4,6)],
        report3.LS$arm1$survprob$Surv[4],
        report4.LS$arm1$survprob$Surv[4],
        report5.LS$arm1$survprob$Surv[4],
        report6.LS$arm1$survprob$Surv[c(4,6,8)],
        report7.LS$arm1$survprob$Surv[c(4,6,8)],
        report8.LS$arm1$survprob$Surv[c(4,6,8)],
        report9.LS$arm1$survprob$Surv[c(4,6,8)])

rep.prob.CW <- c(report1.CW$arm1$survprob$Surv[c(2,4,6)],
        report2.CW$arm1$survprob$Surv[c(2,4,6)],
        report3.CW$arm1$survprob$Surv[4],
        report4.CW$arm1$survprob$Surv[4],
        report5.CW$arm1$survprob$Surv[4],
        report6.CW$arm1$survprob$Surv[c(4,6,8)],
        report7.CW$arm1$survprob$Surv[c(4,6,8)],
        report8.CW$arm1$survprob$Surv[c(4,6,8)],
        report9.CW$arm1$survprob$Surv[c(4,6,8)])

times.Reported <- c(35.1, 34, 4.07, 13.73, 35.8, 21.9, 65.8, 28.1, 15.23,
27.57)
prob.Reported <- c(.851, .745, .497, 1, .524, .436, .173, .541, .725, .90,
        .84, .70, .46, .34, .29, .90, .85, .73, .55, .44, .38)

rep.times.ICC <- data.frame(times.Reported, rep.times.MS, rep.times.LS,
        rep.times.CW)

```

```

rep.prob.ICC <- data.frame(prob.Reported, rep.prob.MS, rep.prob.LS,
rep.prob.CW)

# ICC - REPORTED times
icc(rep.times.ICC,model = 'twoway', type = 'agreement', unit = 'single')
# ICC - REPORTED probabilities
icc(rep.prob.ICC,model = 'twoway', type = 'agreement', unit = 'single')

#####
#####
# Sensitivity analysis

# Times agreement
# no MS
ICC2.CI(times, iv = curveID, data = m.icc[m.icc$rater !=1,])
# no LS
ICC2.CI(times, iv = curveID, data = m.icc[m.icc$rater !=2,])
# no CW
ICC2.CI(times, iv = curveID, data = m.icc[m.icc$rater !=3,])

# reported times #####
# no MS
ICC2.CI(rate, iv = curveID, data = m.icc.prob[m.icc.prob$rater !=1,])
# no LS
ICC2.CI(rate, iv = curveID, data = m.icc.prob[m.icc.prob$rater !=2,])
# no CW
ICC2.CI(rate, iv = curveID, data = m.icc.prob[m.icc.prob$rater !=3,])

# Probs agreement
# no MS
no.MS.prob <- data.frame(prob.LS,prob.CW)
icc(no.MS.prob,model = 't', type = 'a', unit = 's')
# no LS
no.LS.prob <- data.frame(prob.MS,prob.CW)
icc(no.LS.prob,model = 't', type = 'a', unit = 's')
# no CW
no.CW.prob <- data.frame(prob.MS,prob.LS)
icc(no.CW.prob,model = 't', type = 'a', unit = 's')

#####
# Compare IPD

#####          Per curve ICC #####

#curve 1
icc(data.frame(IPD_1$IPD$time, IPD_1.LS$IPD$time, IPD_1.CW$IPD$time),
      model = 't', type = 'a', unit = 's')
#curve 2
icc(data.frame(IPD_2$IPD$time, IPD_2.LS$IPD$time, IPD_2.CW$IPD$time),
      model = 't', type = 'a', unit = 's')
#curve 3
icc(data.frame(IPD_3$IPD$time, IPD_3.LS$IPD$time, IPD_3.CW$IPD$time),
      model = 't', type = 'a', unit = 's')
#curve 4

```

```

icc(data.frame(IPD_4$IPD$time, IPD_4.LS$IPD$time, IPD_4.CW$IPD$time),
     model = 't', type = 'a', unit = 's')
#curve 5
icc(data.frame(IPD_5$IPD$time, IPD_5.LS$IPD$time, IPD_5.CW$IPD$time),
     model = 't', type = 'a', unit = 's')
#curve 6
icc(data.frame(IPD_6$IPD$time, IPD_6.LS$IPD$time, IPD_6.CW$IPD$time),
     model = 't', type = 'a', unit = 's')
#curve 7
icc(data.frame(IPD_7$IPD$time, IPD_7.LS$IPD$time, IPD_7.CW$IPD$time),
     model = 't', type = 'a', unit = 's')
#curve 8
icc(data.frame(IPD_8$IPD$time, IPD_8.LS$IPD$time, IPD_8.CW$IPD$time),
     model = 't', type = 'a', unit = 's')
#curve 9
icc(data.frame(IPD_9$IPD$time, IPD_9.LS$IPD$time, IPD_9.CW$IPD$time),
     model = 't', type = 'a', unit = 's')
#curve 10
icc(data.frame(IPD_10$IPD$time, IPD_10.LS$IPD$time, IPD_10.CW$IPD$time),
     model = 't', type = 'a', unit = 's')
#curve 11
icc(data.frame(IPD_11$IPD$time, IPD_11.LS$IPD$time, IPD_11.CW$IPD$time),
     model = 't', type = 'a', unit = 's')

#####
#####      Bland-Altman      #####
#####

install.packages("blandr")
install.packages("ggplot2")
require(blandr)
require(ggplot2)

ba.plot.times <- blandr.draw(rep.times.MS, times.Reported, ciShading = FALSE)
ba.plot.time + labs(title = 'Bland-Altman plot for agreement with reported
survival times')
plot.new()
ba.plot.prob <- blandr.draw(rep.prob.MS, prob.Reported, ciShading = FALSE)
ba.plot.prob + labs(title = 'Bland-Altman plot for agreement with reported
survival rates')

```