

SurvdigitizeR: R Package to Automate the Digitization of Published Kaplan-Meier Curves

Jasper Zhang

The Hospital for Sick Children and the University of Toronto

Oct 26, 2023



Outline

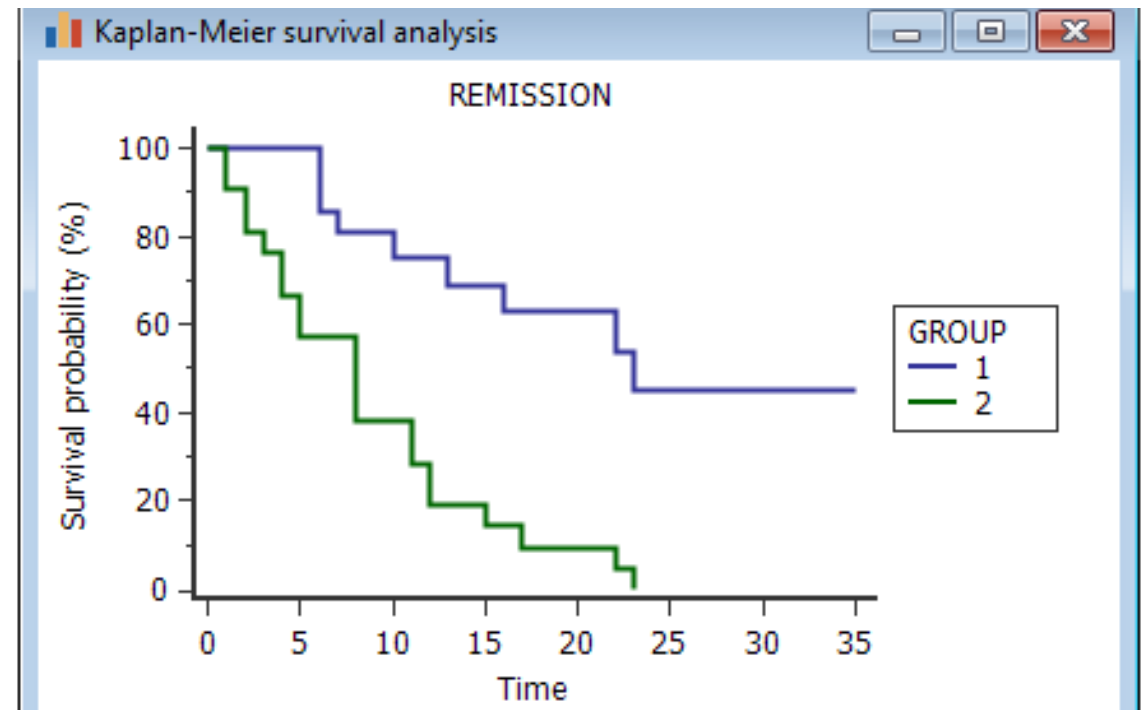
- Introduction
- The R-based Survival Curve Digitizer
- Live Demonstration
- Applications and Discussions
- Conclusion and Q&A

The Role of Survival Analysis in Economic Evaluations and Meta-analyses

- Economic evaluations and meta-analyses often rely on survival probabilities from Kaplan-Meier (KM) curves.
- Health technology assessments use statistical methods like decision analytic modelling, which often use time-to-event data.
- Time-to-event data allow researchers to estimate survival or incidence probabilities, supporting evidence-based healthcare decision-making.
- It is challenging to obtain individual-level patient data (IPD), and researchers often work on summarized statistics.

What is a Kaplan-Meier Curve?

- Graphical representation of survival data
- Estimates event probabilities over time while accounting for censored observations
- Widely used in medical research for survival analysis and comparing treatment groups



Challenges in Manual IPD Reconstruction

- Current methods to reconstruct patient-level data involve **manually digitizing KM curves** and extracting "at risk" tables.
- Manual extraction of probabilities from KM curves is time-consuming, expensive, and prone to error, difficult to reproduce.
- Manual digitization requires technical training, leading to increased time commitment.

Manual Digitization

Step 1: Open Source Image File

Step 2: Set Axes

Step 3: Select Curve

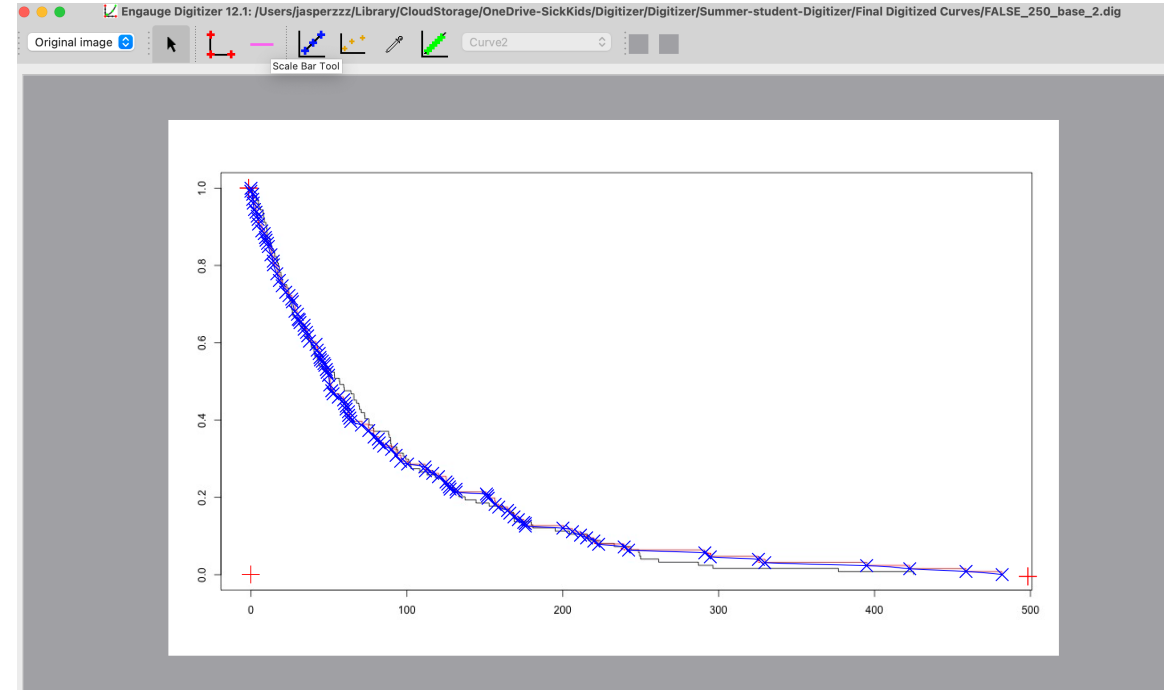
Step 4: Digitize Curve

Use the 'Digitize' or 'Segment Fill' tool to trace over the survival curve.

Aim to capture all important points accurately, especially where significant changes occur.

Step 5: If multiple curves exist, repeat Steps 3-4.

Step 6: Export the Digitized Data



A New Approach: Automating KM Curve Digitization



The aim is to develop an efficient and accurate algorithm to automate the extraction of survival probabilities from KM curves.



The performance of the algorithm will be evaluated through a simulation study and validation on real-world published KM plots.



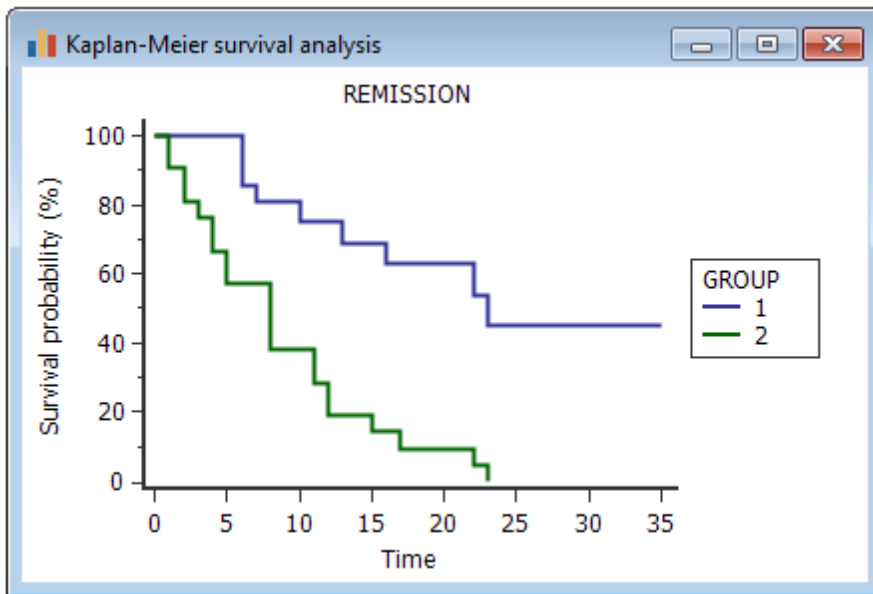
This approach aims to reduce user input, increase accuracy, and ensure reliability.



The goal is to provide an open-source R package and a Shiny App for easy accessibility, thereby promoting reproducibility and consistency in results.

The R-based Survival Curve Digitizer

- An algorithm that **automates** the digitization of KM curves from image files
 - Minimize user input and errors
 - Improve reproducibility



id	time	St	curve	
1		0	100	1
2	0.25531915		100	1
3	0.31914894	99.4520548		1
4	0.38297872	99.4520548		1
5	0.44680851	99.1780822		1
6	0.57446809	99.1780822		1
7	0.63829787	98.630137		1
8	0.63829787	98.630137		1
9	0.70212766	97.5342466		1
10	0.70212766	97.5342466		1

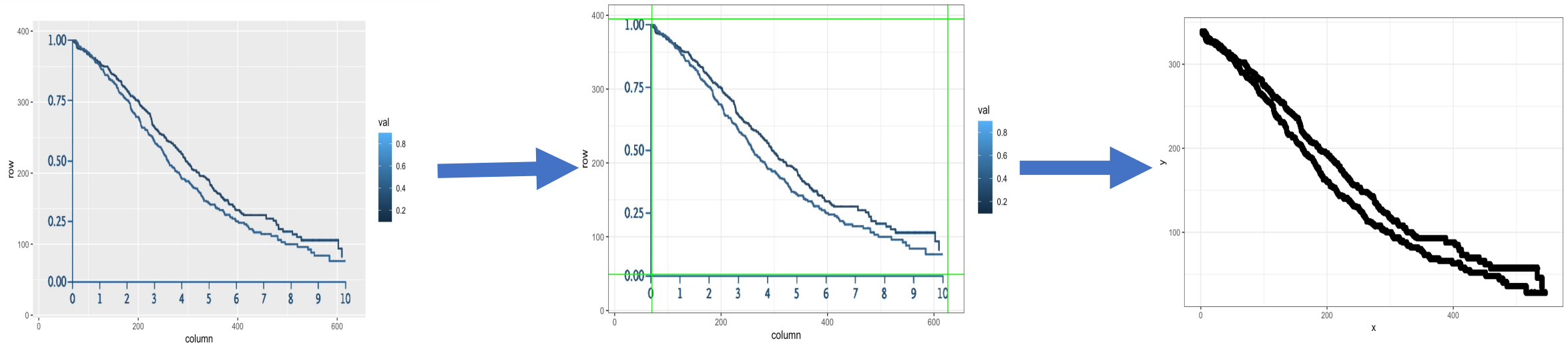
Methods Overview

- HSL scale image processing
 - Simplifies the process of identifying and separating colours in KM curve images
 - Enhances the accuracy and robustness of the digitization algorithm
- Optical character recognition (OCR) for axis location and labels
 - Detect the location of words and numbers as references.
- K-medoids clustering for separating multiple curves
 - K-medoids is a clustering algorithm that uses medoids instead of centroids to minimize the sum of dissimilarities between data points, making it more robust to noise and outliers than K-means.

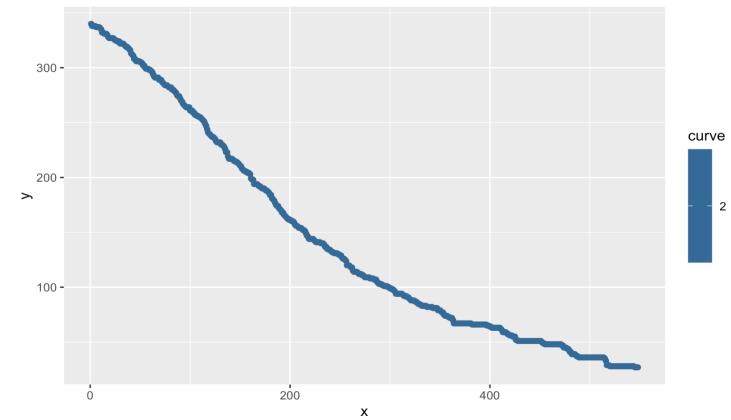
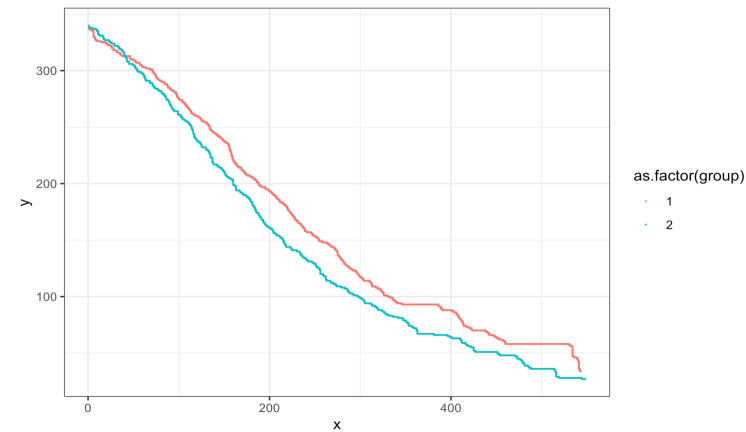
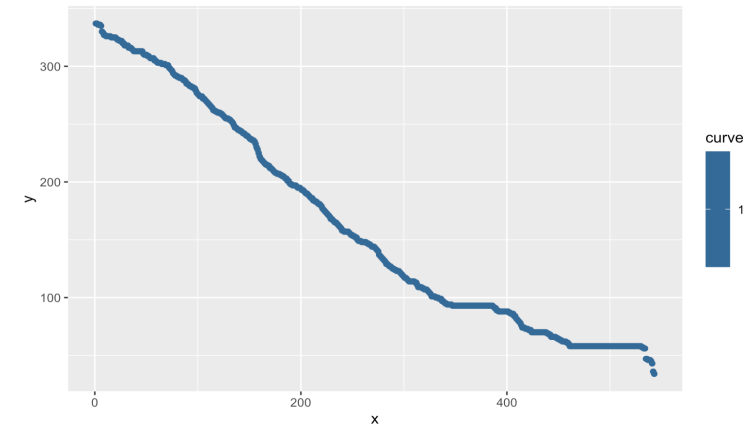
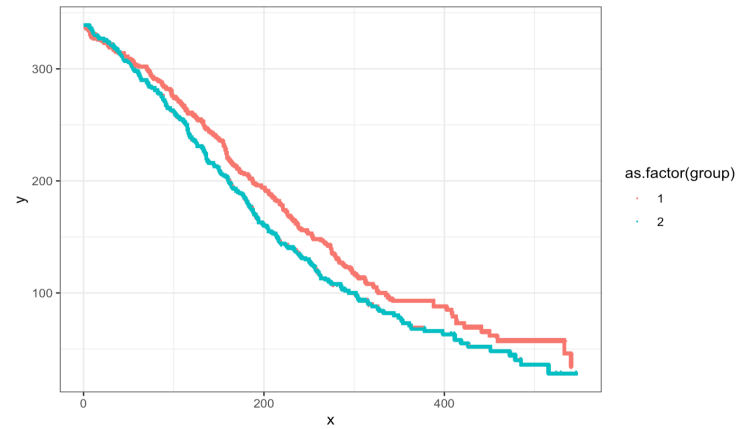
Read Image and Identify Axis

- Specify inputs: number of curves, y and x-axis increments, Remove legends and clear the plot.
- Read .jpeg or .png images to Hue, Saturation, Lightness (HSL) scale.
- Detect the Axis and Reset pixel locations and Clean the figure

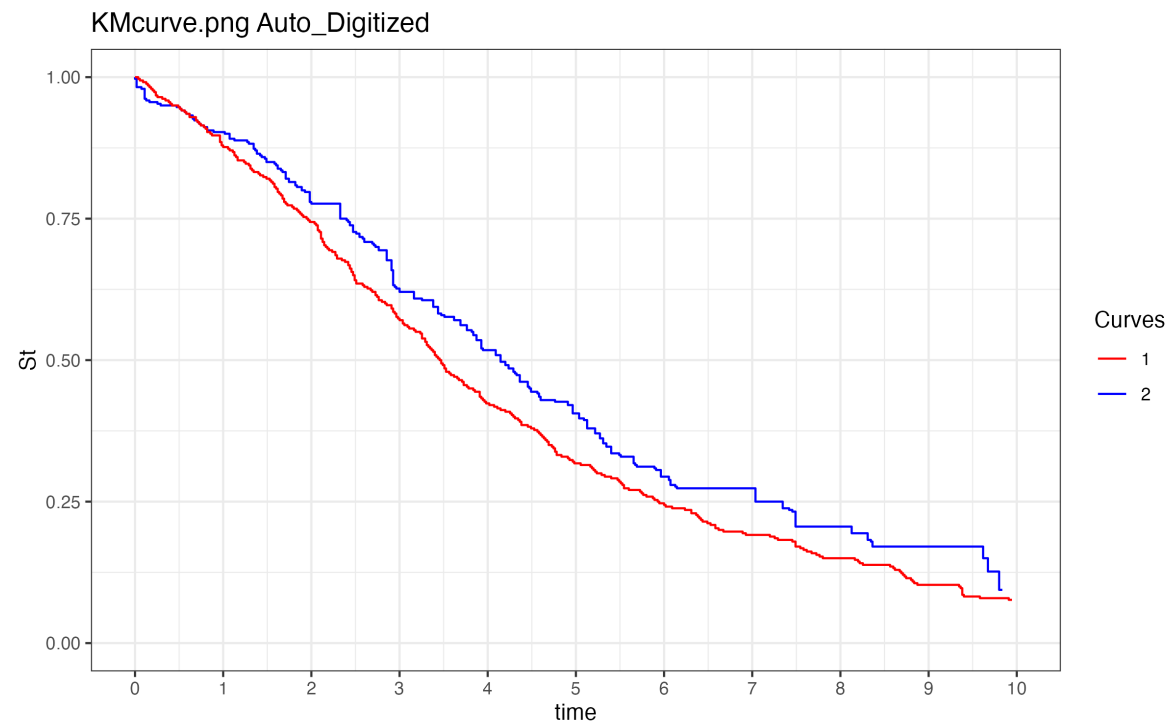
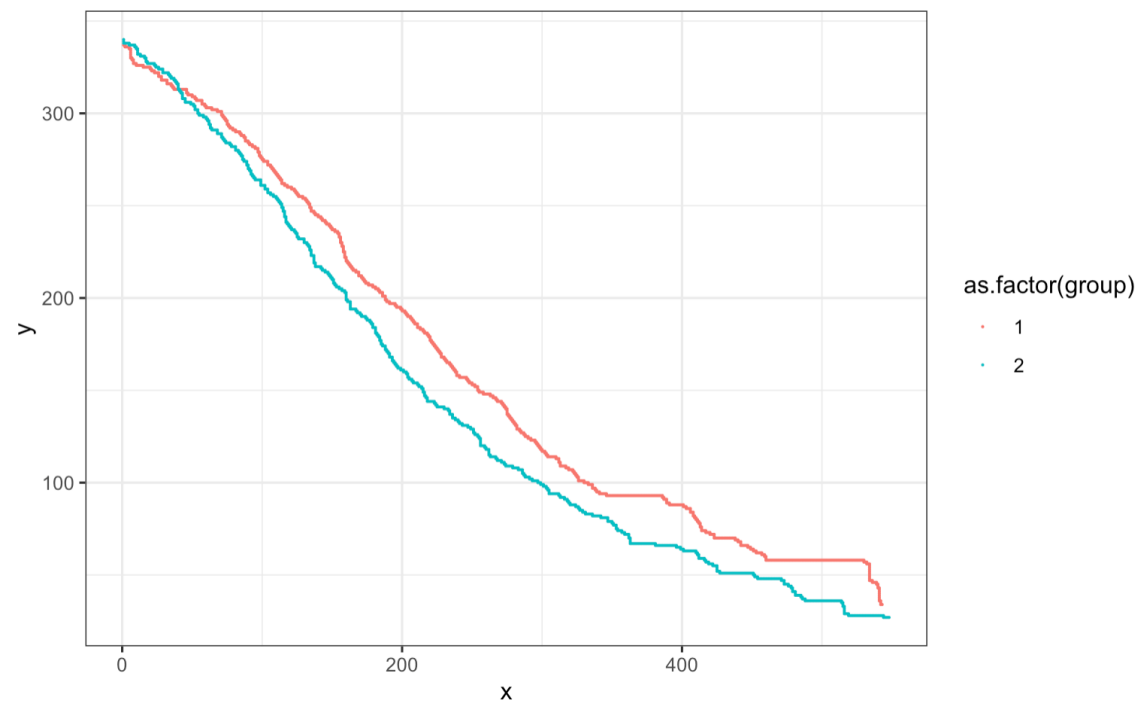
Reading in KM curves



Color Clustering and Lines Isolation



Map Pixel location back to Time and Probability Scales

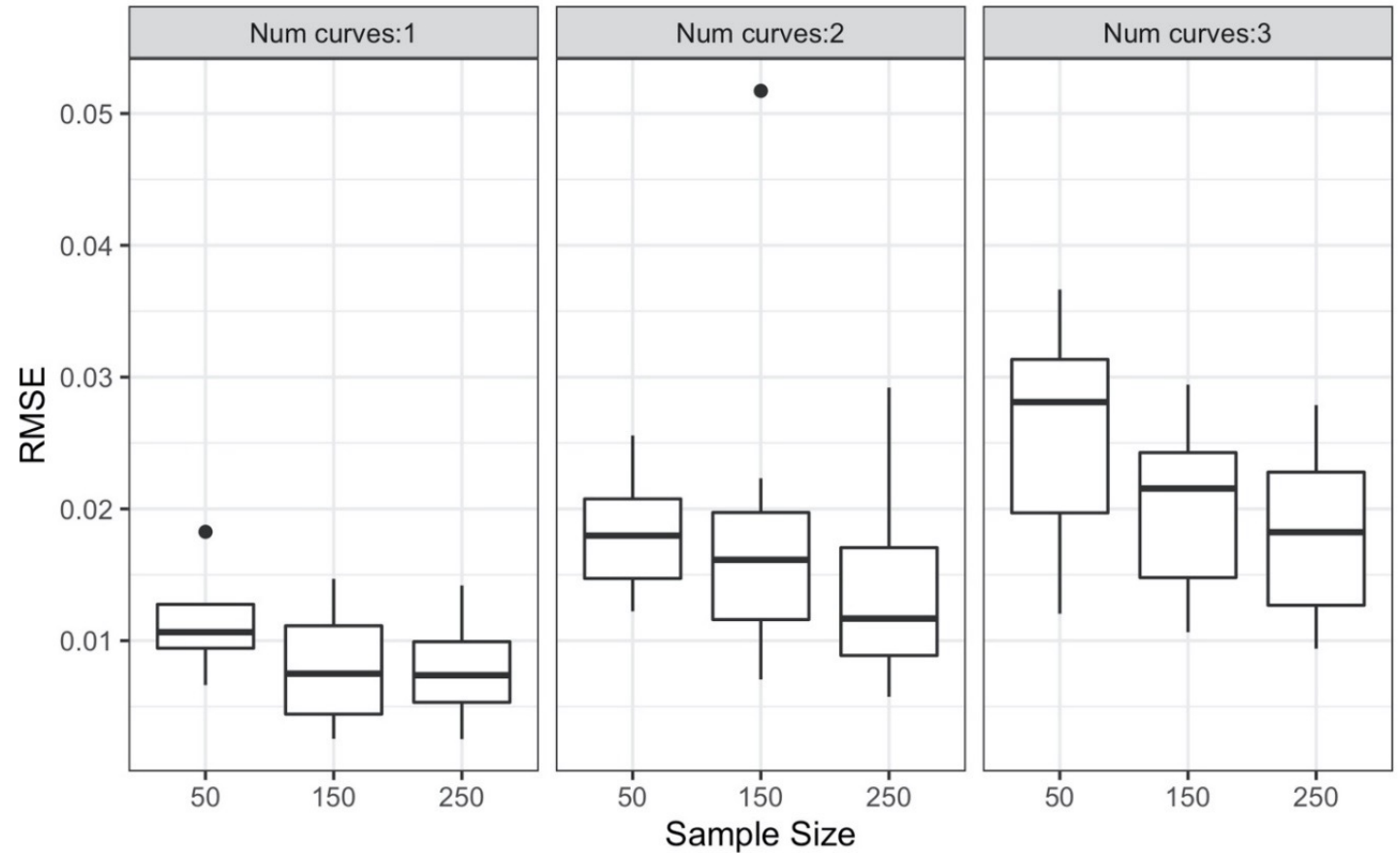


Performance Validation: Simulation, Real-World

- **Simulations:** Conducted simulations by generating 36 survival data sets from an exponential distribution and random censoring. Plotted KM curves using base R graphics and ggplot style.
 - comparison of source data and digitized KM curves using the Root Mean Squared Error (RMSE) for both auto-digitized and manually digitized curves against the actual values.
- **Real-World Scenario:** Evaluated the real-world application by comparing auto-digitized and manually digitized curves, completed by well-trained researchers, using 8 real-world survival KM curves digitized with Engauge.
 - Bland-Altman Analysis: Assessed the **agreement** between manual and automated digitization using Bland-Altman plots as a measure of consistency.

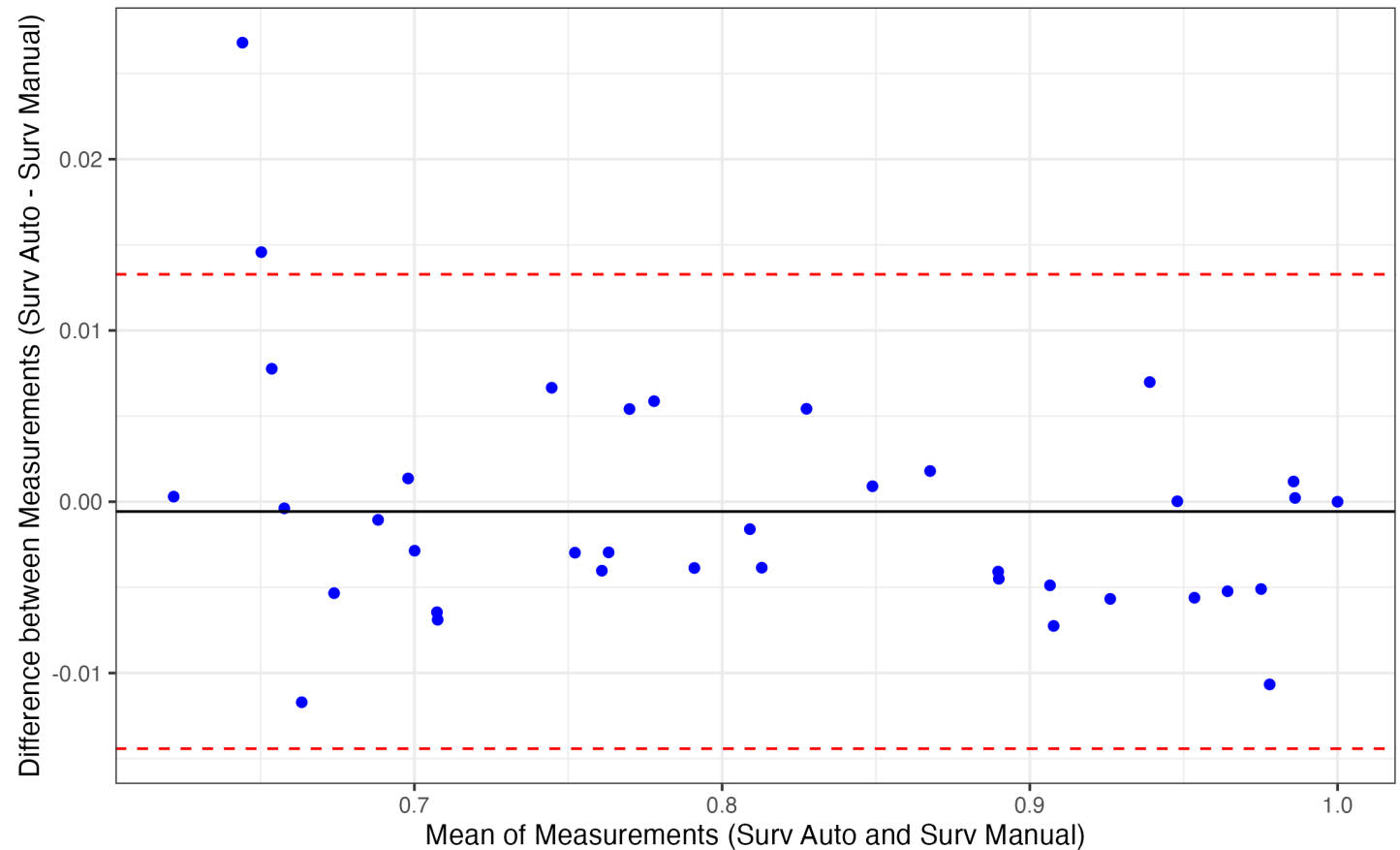
Simulation Result

- RMSE Boxplot stratified by sample size and number of curves



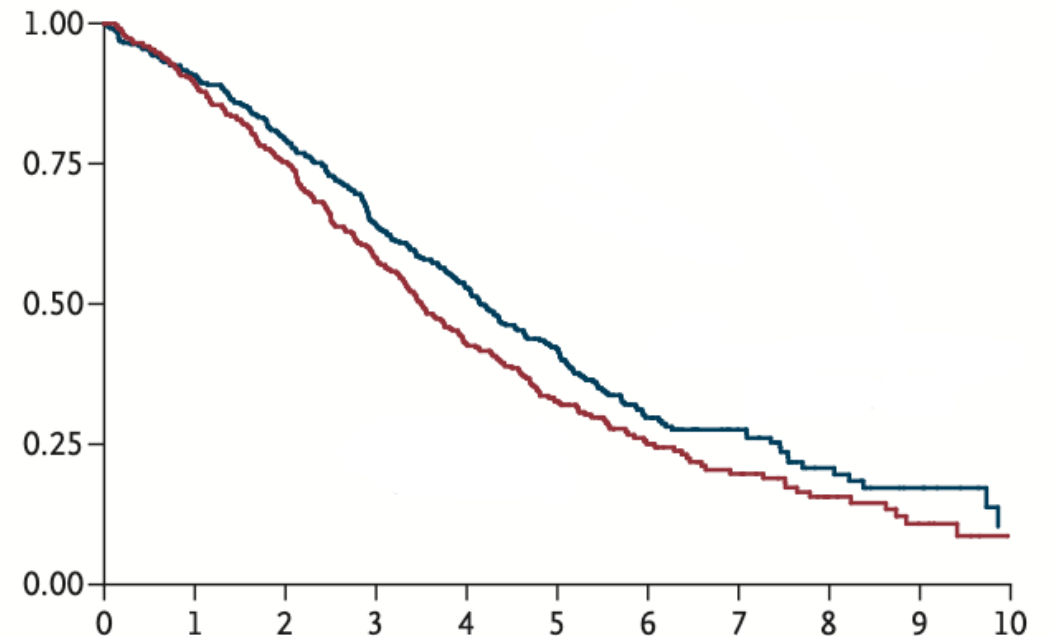
Comparison on agreement

- Performance evaluation: Bland-Altman analysis, also known as Bland-Altman plot or Limits of Agreement (LoA) plot



Live Demonstration

- **Number of curves [2]:** Specifies the number of survival curves to digitize in the image.
- No indicators for censoring
- **X-axis range [0-10 by step size 1]:** Defines the range and step size of the x-axis.
- **Y-axis range[0-1 by step size 0.25]:** Defines the range and step size of the y-axis.
- **Y-axis vertical [TRUE]:** This boolean argument determines the orientation of the Y-axis labels.



Step 1. Install the Package

```
# Install from GitHub  
devtools::install_github("Pechli-Lab/SurvdigitizeR")
```

```
# load the Library  
library(SurvdigitizeR)
```

Step 2. Run survival_digitize function

```
# Run 'survival_digitize' to digitize KM curve, output stored in 'out'
out <-survival_digitize(

  # Specify image file path for KM curve
  img_path = here::here("vignettes","KMcurve.png"),

  # Specify number of curves in image
  num_curves = 2,

  # Specify if the KM curve has censoring (False in this case)
  censoring = F,

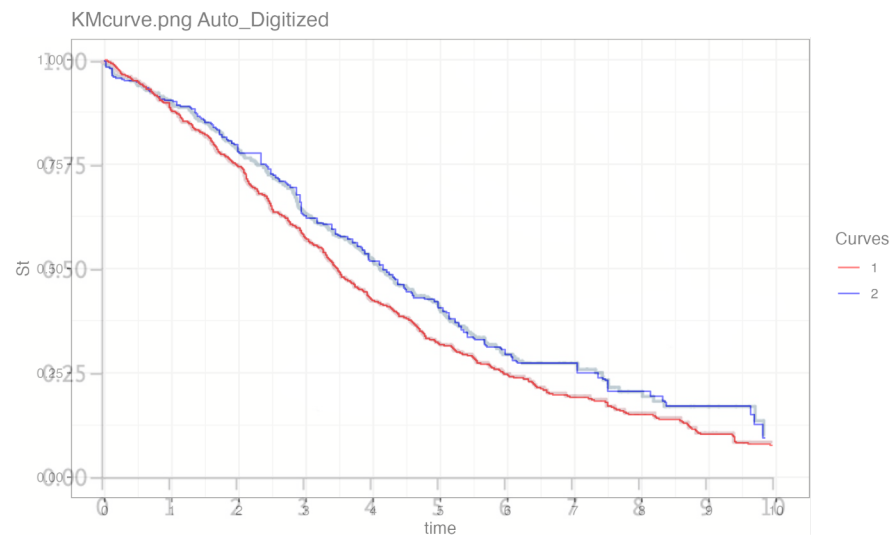
  # Define start, end, increment of X-axis
  x_start = 0, x_end = 10, x_increment = 1,

  # Define start, end, increment of Y-axis
  y_start = 0, y_increment = 0.25, y_end = 1,

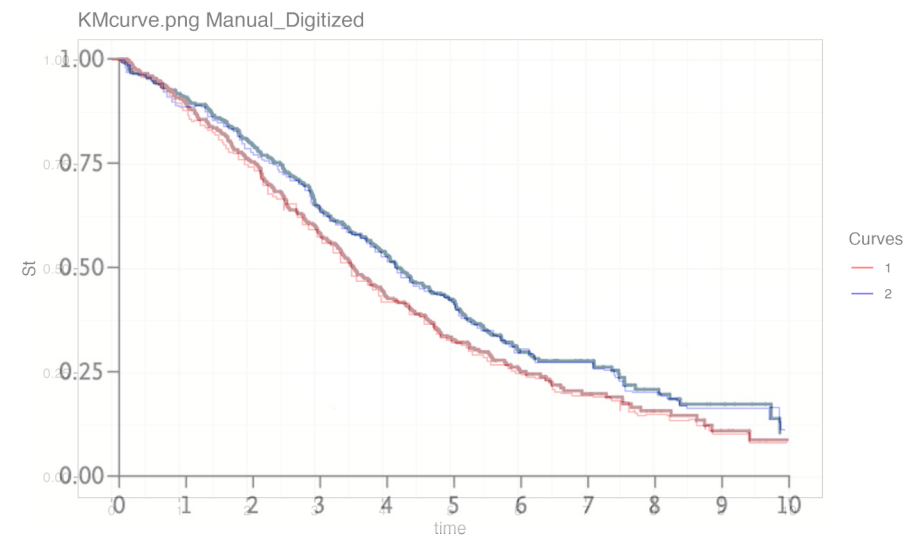
  # Specify orientation of Y-axis labels (True for vertical)
  y_text_vertical = T
)
```

Result

Auto Digitization



Manual Digitization



Save to table

```
# Write 'out' (digitized data) to a CSV file
write.csv(
  # Object containing digitized data
  out,
  # Specify file location
  here::here("vignettes", "digitized_data",
  paste0("out_put_", img_name,
  "_Auto_Digitized.csv")),
  # Exclude row names in the file
  row.names = FALSE
)
```

	A	B	C	D	
1	id	time	St	curve	
2	1	0	1	1	
3	2	0	1	1	
4	3	0.01818182	0.99705882	1	
5	4	0.05454545	0.99705882	1	
6	5	0.07272727	0.99411765	1	
7	6	0.09090909	0.99411765	1	
8	7	0.10909091	0.97647059	1	
9	8	0.10909091	0.97647059	1	
10	9	0.12727273	0.97352941	1	
11	10	0.14545455	0.97352941	1	
12	11	0.16363636	0.97058824	1	
13	12	0.18181818	0.97058824	1	
14	13	0.2	0.96764706	1	
15	14	0.27272727	0.96764706	1	
16	15	0.29090909	0.96176471	1	
17	16	0.38181818	0.96176471	1	
18	17	0.4	0.95294118	1	
19	18	0.47272727	0.95294118	1	

Vignettes page

SurvdigitizeR 0.0.0.9000

Reference

Articles ▾

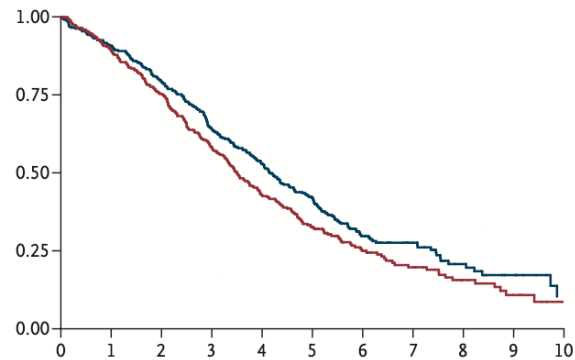
Example-digitization

Jasper Zhang

2023-06-09

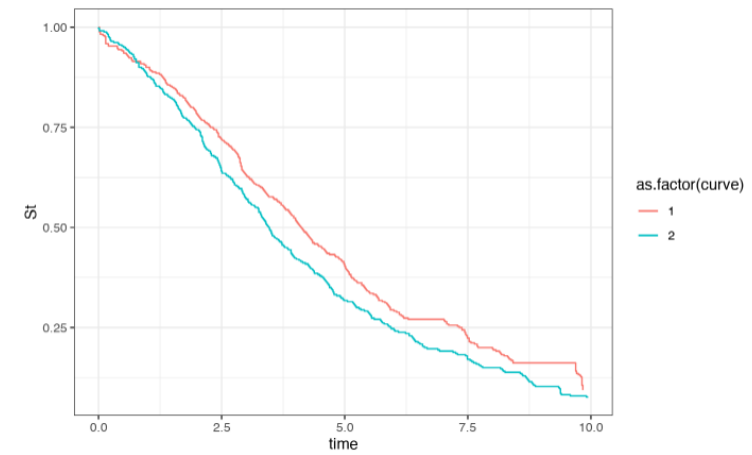
```
library(SurvdigitizeR)
library(here)
library(ggplot2)
library(jpeg)
library(dplyr)
```

Image to be digitized

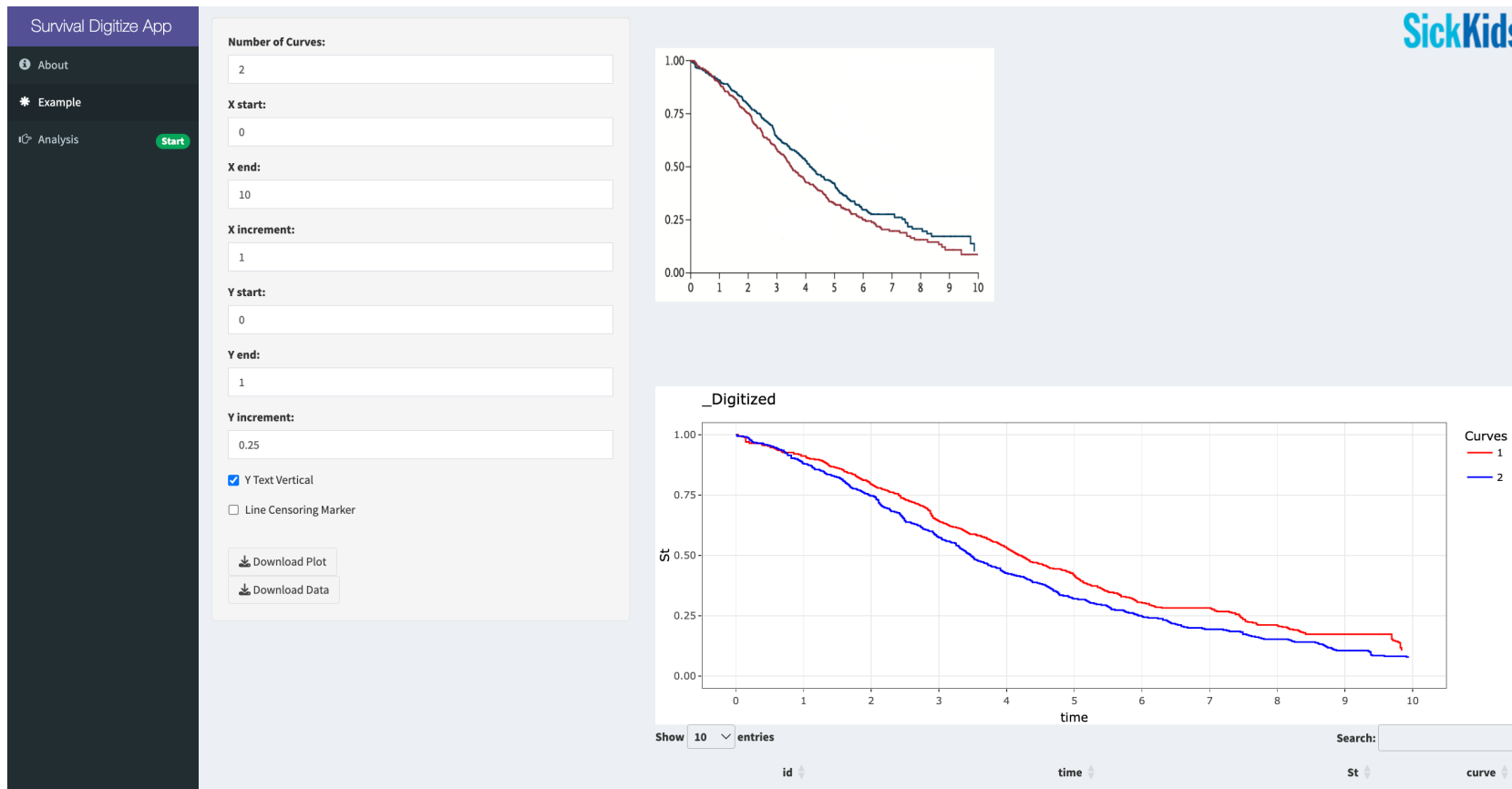


```
out1 <- survival_digitize(img_path = here::here("vignettes", "KMcurve.png"),
                          bg_lightness = 0.3, attempt_OCR = T, word_sensitivity = 30, num_curves = 2, censor
ing = F,
                          x_start = 0, x_end = 10, x_increment = 1, y_start = 0, y_increment = 0.25, y_end =
1, y_text_vertical = T)

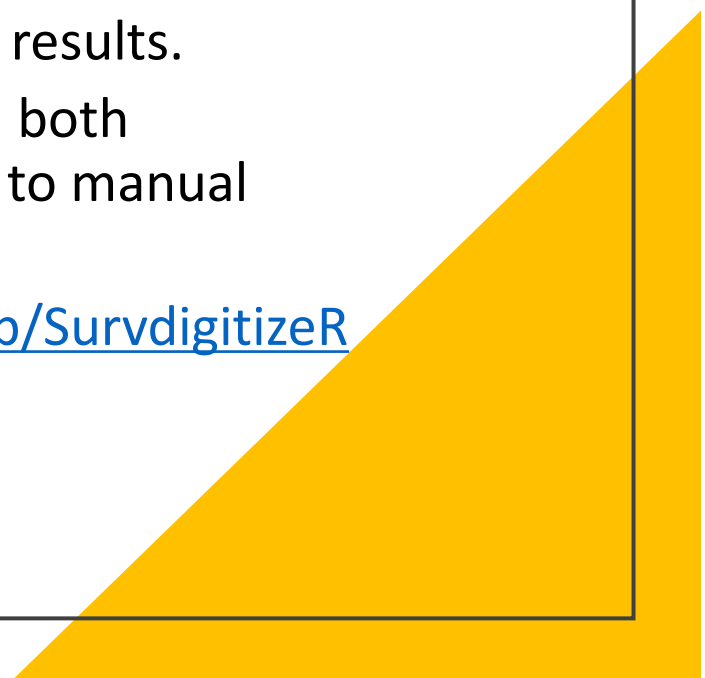
out1 %>%
  ggplot(aes(x = time, y = St, color = as.factor(curve), group = curve)) +
  geom_step() + theme_bw()
```



R Shiny App



Conclusions

- **Streamlines digitization with minimal input:** The algorithm simplifies the digitization process, requiring minimal user interaction for efficient results.
 - **Accurate in various scenarios:** Successfully digitizes KM curves in both simulated and real-world situations, showing accuracy comparable to manual digitization methods.
 - Open-source R package on GitHub: <https://github.com/Pechli-Lab/SurvdigitizeR>
 - R Shiny App: <https://pechlilab.shinyapps.io/Shiny-KMcurve/>
- 
- A large yellow triangle is positioned in the bottom right corner of the slide, pointing towards the top right.

Acknowledgements

This study was financially supported by an unrestricted grant by the Canadian Agency for Drugs and Technologies in Health (CADTH)

Collaborators: Juan David Rios, Tilemanchos Pechlivanoglou, Alan Yang, Qiyue, Zhang Dimitrios Deris, Ian Cromwell, Petros Pechlivanoglou

The poster for this project won the **first prize** in the virtual poster presentation competition in the student conference (CSSC 2023) of 2023 Statistical Society of Canada Annual Meeting in May 2023.



Questions

References

Guyot, P., Ades, A., Ouwens, M. J., & Welton, N. J. (2012). Enhanced secondary analysis of survival data: Reconstructing the data from published Kaplan-Meier survival curves. *BMC Medical Research Methodology*, 12(1), 9. <https://doi.org/10.1186/1471-2288-12-9>

Gallacher, D., Kimani, P., & Stallard, N. (2021). Extrapolating Parametric Survival Models in Health Technology Assessment: A Simulation Study. *Medical Decision Making*, 41(1), 37–50. <https://doi.org/10.1177/0272989X20973201>

Ooms, J. (2023). *tesseract: Open Source OCR Engine*. <https://docs.ropensci.org/tesseract/> (website) <https://github.com/ropensci/tesseract> (devel)

Mitchell, M., Muftakhidinov, B., Winchen, T., Wilms, A., Schaik, B. V., Badshah400, Mo-Gul, Badger, T. G., Jędrzejewski-Szmek, Z., Kensington, & Kylesower. (2020). *Engauge Digitizer Software* (v12.2.1). Zenodo. <https://doi.org/10.5281/ZENODO.3941227>