Software Tool Article

Open Access CrossMark

# SiPy 0.7.0 – R-based ANOVA and survival analyses

## Abstract

Statistics in Python (SiPy) is a data analysis tool built using Python and integrates analysis from R; yet, aiming to reduce the learning curve need to learn either Python or R. Recently, SiPy version 0.6.0 had been released but is lacking in ANOVA and survival analyses. Here, we extend SiPy version 0.6.0 to version 0.7.0 (codenamed Keropok), released on 05 December 2025, by integrating 8 ANOVA-based methods and 15 survival analysis methods from R.

**Keywords:** SiPy version 0.6.0, data, experimental designs, medical research, data analysts

Volume 9 Issue 1 - 2026

Wira Bin Ambel,[1] Maurice HT Ling[2,3]

[1]School of Health & Life Sciences, Teesside University, UK
[2]Newcastle Australia Institute of Higher Education, University of Newcastle, Australia
[3]HOHY PTE LTD, Singapore

**Correspondence:** Maurice HT Ling, Newcastle Australia Institute of Higher Education, University of Newcastle, Australia

**Received:** December 12, 2025 | **Published:** January 08, 2026

## Introduction

Statistical analysis of clinical and experimental data frequently requires the use of models that can partition, compare, and explain sources of variability across groups, covariates, and time.[1,2] Analysis of variance (ANOVA) and its extensions remain foundational tools for evaluating mean differences under varying experimental designs.[3] Similarly, survival analysis methods are central to the analysis of time-to-event data in medical research.[4,5]

While both Python and R platforms are frequent choices among data analysts,[6] it is generally accepted that R is stronger in analytics, especially statistical analysis methods, compared to Python.[7] SiPy[8] is a lightweight statistical interface written in Python, and has been demonstrated as a potential platform for incorporating R methods while reducing the learning curve needed to learn R. In this paper, we integrated R-based ANOVA and survival analysis methods into SiPy 0.6.0;[8] thereby, presenting SiPy 0.7.0 (codenamed as Keropok) released on 05 December 2025 and illustrate its application through a simulated clinical dataset designed to resemble a multi-centre randomized trial.

### Simulated data for case study

We generate a set of data using a Python data generation script (file name = survival_dataset_generator.py in sipy/data folder) comprising of 1000 patients across 20 centres, randomized 1:1 to Drug or Control arms, to resemble a multi-centre clinical trial. The ages of these patients were normally distributed and averaged at 60 years old with standard deviation of 10 years old, with 50% of each gender. Three stages of disease (I, II, III) was randomized at 40% to 40% to 20%. Baseline biomarker was normally distributed at 50 units with a standard deviation of 10 units. Baseline quality-of-life score was normally distributed at 70 units with a standard deviation of 12 units.

The simulated data incorporated treatment effects at 3 and 6 months. In the Drug arm, mean biomarker reductions of 5.0 and 8.0 units at 3 and 6 months respectively, with quality-of-life improvement of 4 units at 6 months. In the Control arm, mean biomarker reductions of 1.0 and 0.5 units at 3 and 6 months respectively, with quality-of-life improvement of 1.0 unit at 6 months. Time-to-event data was generated using a Weibull distribution.

The resulting simulated data is stored as a comma-delimited file in sipy/data folder as survival_dataset.csv and can be read into SiPy as sdata variable using the following command: read csv sdata from data/survival_dataset.csv.

### ANOVA-based analyses

The following ANOVA-based methods from R are available in SiPy 0.7.0 (sample ANOVA analyses are shown in Figure 1):

I. ANOVA. For 1-way ANOVA; for example, to evaluate the raw treatment effect at 6 months between drug and control arms (such as in Satre et al.[9]) with Tukey as posthoc test (example command: ranova anova data=sdata y=biomarker_6m x=arm posthoc=tukey) as shown in Figure 1A where the results only the means between the drug and control arms are significant (p-value < 2e-16). This can be extended to 2-way ANOVA with arm and gender as factors (example command: ranova anova data=sdata y=biomarker_6m x=arm,sex posthoc=tukey) or 3-way ANOVA with arm, gender and centre as factors (example command: ranova anova data=sdata y=biomarker_6m x=arm,sex,center posthoc=tukey) or N-way ANOVA.

II. Kruskal-Wallis Test is a non-parametric equivalent of 1-way ANOVA.[10] For example, if the data cannot be assumed to be normally distributed, to evaluate the raw treatment effect at 6 months between drug and control arms with Dunn as posthoc test (example command: ranova kruskal data=sdata y=biomarker_6m x=arm posthoc=dunn).

III. Friedman Test is a non-parametric equivalent of repeated measures ANOVA.[11] For example, evaluating various stress reduction techniques using the same set of test subjects and assuming no interaction between the techniques.

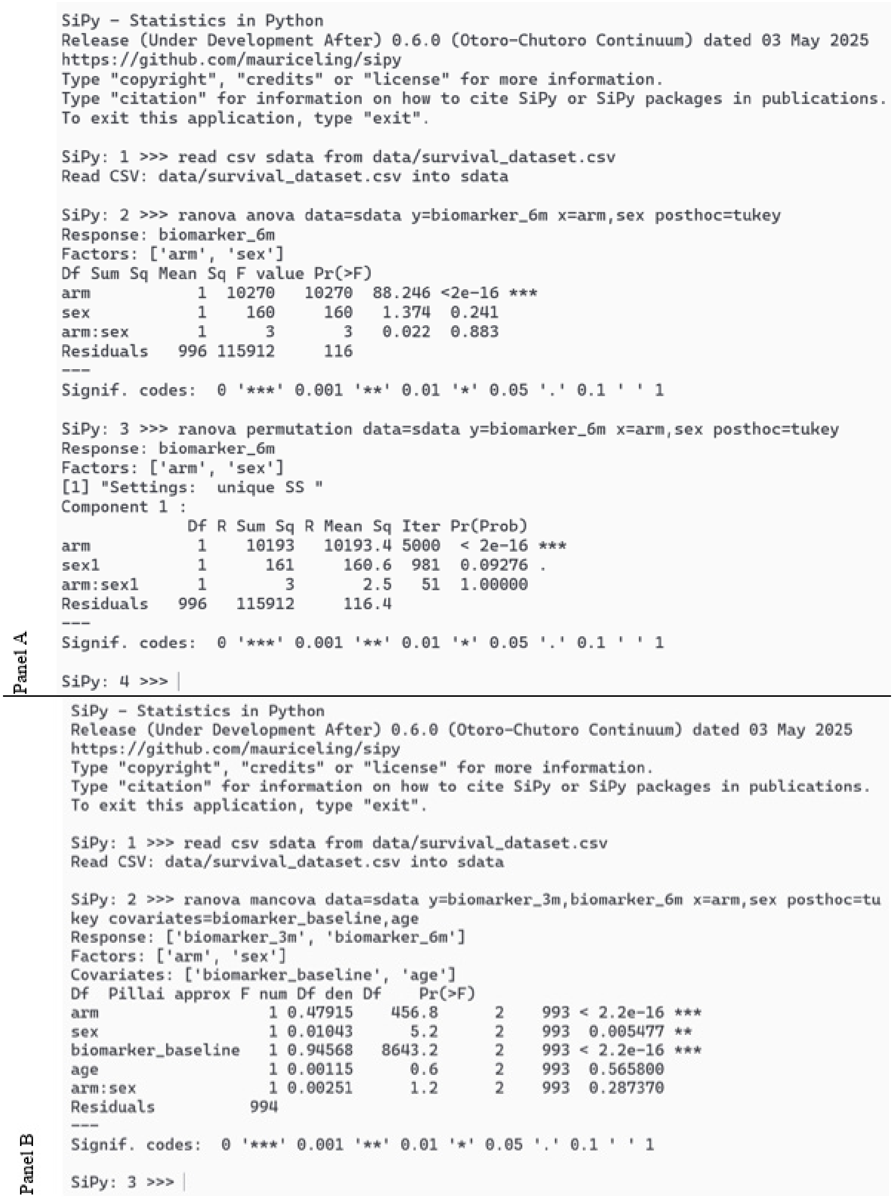
IV. Welch Test is used when the assumption of equal variances cannot be assumed in 1-way ANOVA.[12] For example, if the variances of biomarker levels at 6 months between drug and control arms cannot be assumed to be equal, to evaluate the raw treatment effect at 6 months between drug and control arms with Games-Howell as posthoc (example command: ranova welch data=sdata y=biomarker_6m x=arm posthoc=games-howell).

V. Permutation Test can be used when normality cannot be assumed but N-way ANOVA is needed;[13] hence, cannot be addressed by Kruskal-Wallis Test or Welch Test. For example, to evaluate the raw treatment effect at 6 months between drug and control arms in various centres, the command will be ranova permutation data=sdata y=biomarker_6m x=arm,center; as shown in Figure 1A which also shows that only the means between the drug and control arms are significant (p-value < 2e-16).

VI. ANCOVA, which is ANOVA with one or more continuous variables (known as covariates). As ANCOVA mathematically partitions the variance of the dependent variable into variances of the independent covariates and variances of independent factors, it is commonly used to account for baseline differences measured as covariates.[14] For example, the raw treatment effect at 6 months between drug and control arms must take into account of the baseline biomarker levels. Hence, baseline biomarker level is used as a covariate (example command: ranova ancova data=sdata y=biomarker_6m x=arm covariates=biomarker_baseline posthoc=tukey). Similar to N-way ANOVA, there can be multiple covariates (example command: ranova ancova data=sdata y=biomarker_6m x=arm,sex covariates=biomarker_baseline,age posthoc=tukey).

VII. MANOVA is ANOVA with more than one dependent variables. For example, we can compare the mean biomarker levels at 3 months and 6 months simultaneously between drug versus control arm (example command: ranova manova data=sdata y=biomarker_3m,biomarker_6m x=arm posthoc=tukey). Similar to N-way ANOVA, there can be multiple factors (example command: ranova manova data=sdata y=biomarker_3m,biomarker_6m x=arm,sex posthoc=tukey).

VIII. MANCOVA is then a covariates extension to MANOVA; much like ANCOVA to ANOVA. As such, MANCOVA can be used to control for baseline in MANOVA.[15] For example, to compare the mean biomarker levels at 3 months and 6 months simultaneously between drug versus control arm and gender while controlling for baseline biomarker levels and age, the command will be ranova mancova data=sdata y=biomarker_3m,biomarker_6m x=arm,sex posthoc=tukey covariates=biomarker_baseline,age; as shown in Figure 1B showing that arm, sex, and biomarker_baseline are significant (p-value < 0.05).



```
SiPy - Statistics in Python
Release (Under Development After) 0.6.0 (Otoro-Chutoro Continuum) dated 03 May 2025
https://github.com/mauriceling/sipy
Type "copyright", "credits" or "license" for more information.
Type "citation" for information on how to cite SiPy or SiPy packages in publications.
To exit this application, type "exit".

SiPy: 1 >>> read csv sdata from data/survival_dataset.csv
Read CSV: data/survival_dataset.csv into sdata

SiPy: 2 >>> ranova anova data=sdata y=biomarker_6m x=arm,sex posthoc=tukey
Response: biomarker_6m
Factors: ['arm', 'sex']
Df Sum Sq Mean Sq F value Pr(>F)
arm           1  10270   10270  88.246 <2e-16 ***
sex           1    160     160   1.374  0.241
arm:sex       1      3       3   0.022  0.883
Residuals   996 115912     116
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

SiPy: 3 >>> ranova permutation data=sdata y=biomarker_6m x=arm,sex posthoc=tukey
Response: biomarker_6m
Factors: ['arm', 'sex']
[1] "Settings:  unique SS "
Component 1 :
             Df R Sum Sq R Mean Sq Iter Pr(Prob)
arm           1    10193   10193.4 5000  < 2e-16 ***
sex1          1      161     160.6  981  0.09276 .
arm:sex1      1        3       2.5   51  1.00000
Residuals   996   115912     116.4
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

SiPy: 4 >>> 
```

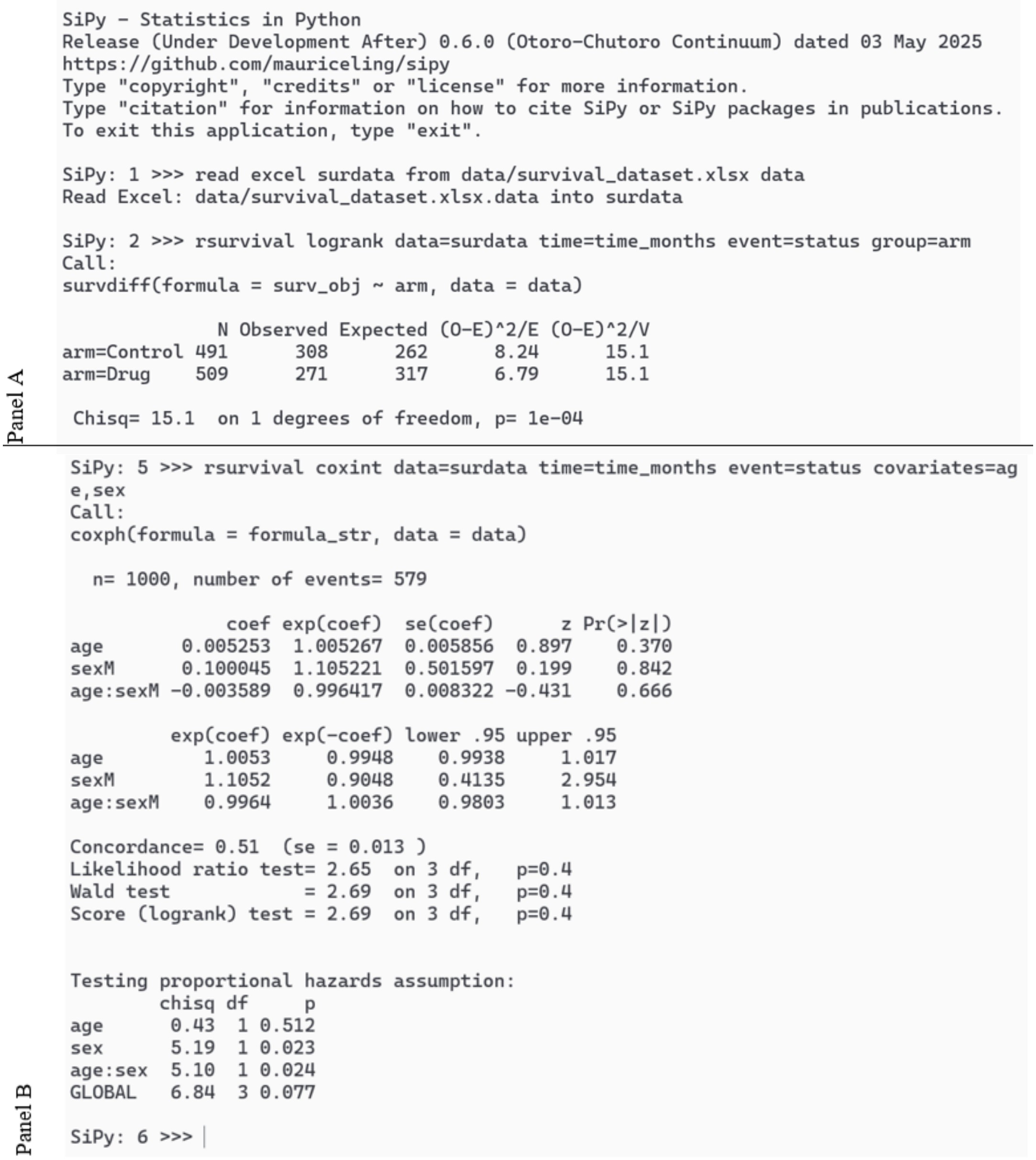
```
SiPy - Statistics in Python
Release (Under Development After) 0.6.0 (Otoro-Chutoro Continuum) dated 03 May 2025
https://github.com/mauriceling/sipy
Type "copyright", "credits" or "license" for more information.
Type "citation" for information on how to cite SiPy or SiPy packages in publications.
To exit this application, type "exit".

SiPy: 1 >>> read csv sdata from data/survival_dataset.csv
Read CSV: data/survival_dataset.csv into sdata

SiPy: 2 >>> ranova mancova data=sdata y=biomarker_3m,biomarker_6m x=arm,sex posthoc=tu
key covariates=biomarker_baseline,age
Response: ['biomarker_3m', 'biomarker_6m']
Factors: ['arm', 'sex']
Covariates: ['biomarker_baseline', 'age']
Df  Pillai approx F num Df den Df    Pr(>F)
arm                  1 0.47915    456.8      2    993 < 2.2e-16 ***
sex                  1 0.01043      5.2      2    993  0.005477 **
biomarker_baseline   1 0.94568   8643.2      2    993 < 2.2e-16 ***
age                  1 0.00115      0.6      2    993  0.565800
arm:sex              1 0.00251      1.2      2    993  0.287370
Residuals          994
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

SiPy: 3 >>> 
```

**Figure 1** Screenshots of Sample ANOVA-Based Analyses. Panel A shows 2-way ANOVA, and Permutation test. Panel B shows MANCOVA.

**Citation:** Ambel WB, Ling MHT. SiPy 0.7.0 – R-based ANOVA and survival analyses. *Open Access J Sci.* 2026;9(1):1–5.
DOI: 10.15406/oajs.2026.09.00280

## Survival-based analyses

The following Survival-Based Analyses methods from R are available in SiPy 0.7.0 (sample survival analyses are shown in Figure 2):



```
SiPy - Statistics in Python
Release (Under Development After) 0.6.0 (Otoro-Chutoro Continuum) dated 03 May 2025
https://github.com/mauriceling/sipy
Type "copyright", "credits" or "license" for more information.
Type "citation" for information on how to cite SiPy or SiPy packages in publications.
To exit this application, type "exit".

SiPy: 1 >>> read excel surdata from data/survival_dataset.xlsx data
Read Excel: data/survival_dataset.xlsx.data into surdata

SiPy: 2 >>> rsurvival logrank data=surdata time=time_months event=status group=arm
Call:
survdiff(formula = surv_obj ~ arm, data = data)

               N Observed Expected (O-E)^2/E (O-E)^2/V
arm=Control 491      308      262      8.24      15.1
arm=Drug    509      271      317      6.79      15.1

 Chisq= 15.1  on 1 degrees of freedom, p= 1e-04
```

```
SiPy: 5 >>> rsurvival coxint data=surdata time=time_months event=status covariates=ag
e,sex
Call:
coxph(formula = formula_str, data = data)

  n= 1000, number of events= 579

              coef exp(coef)  se(coef)      z Pr(>|z|)
age       0.005253  1.005267  0.005856  0.897    0.370
sexM      0.100045  1.105221  0.501597  0.199    0.842
age:sexM -0.003589  0.996417  0.008322 -0.431    0.666

         exp(coef) exp(-coef) lower .95 upper .95
age         1.0053     0.9948    0.9938     1.017
sexM        1.1052     0.9048    0.4135     2.954
age:sexM    0.9964     1.0036    0.9803     1.013

Concordance= 0.51  (se = 0.013 )
Likelihood ratio test= 2.65  on 3 df,   p=0.4
Wald test            = 2.69  on 3 df,   p=0.4
Score (logrank) test = 2.69  on 3 df,   p=0.4


Testing proportional hazards assumption:
        chisq df     p
age      0.43  1 0.512
sex      5.19  1 0.023
age:sex  5.10  1 0.024
GLOBAL   6.84  3 0.077

SiPy: 6 >>> |
```

**Figure 2** Screenshots of Example Survival Analyses. Panel A shows Log-rank test. Panel B shows Cox proportional hazard model with interaction between the covariates.

I. Accelerated Failure Time model (AFT) is commonly utilised to estimate the effect of covariates on the survival times of the patients under a chosen parametric distribution.[16] For instance, to assess how treatment arm, age, sex, disease stage, baseline biomarker level and baseline quality of life influence survival time under a Weibull distribution, the command will be rsurvival aft data=surdata time=time_months event=status covariates=arm,age,sex,stage,biomarker_baseline,qol_baseline dist=Weibull.

II. Competing Risks Regression (Fine-Gray Model) evaluates the effect of covariates on the incidence of an event of interest in the presence of various competing risks.[17] For instance, the command used to determine how treatment arm, age and sex affect failure from cause 1 while accounting for other causes (example command: rsurvival competing data=surdata time=time_months event=status cause=cause group=arm covariates=age,sex).

III. Cox Interaction Model (Cox-Int) is an extension of the Cox model by including interaction effects among covariates.[18] This is useful to assess whether the effect of a covariate depends on another. For example, to test the effect of age and sex at different stages (example command: rsurvival coxint data=surdata time=time_months event=status covariates=age,sex,stage) as shown in Figure 2B.

**Citation:** Ambel WB, Ling MHT. SiPy 0.7.0 – R-based ANOVA and survival analyses. *Open Access J Sci.* 2026;9(1):1‒5. DOI: 10.15406/oajs.2026.09.00280

IV. Cox Proportional Hazards Model (Cox) is utilised to evaluate the effects of covariates on hazard rate while assuming proportional hazards over time.[18] For example, to determine whether survival is influenced by treatment age, sex, and disease stage (example command: rsurvival cox data=surdata time=time_months event=status covariates=age,sex,stage).

V. Exponential-AFT is a survival method by assuming a constant hazard rate over time under an exponential distribution[19] (example command: rsurvival expaft data=surdata time=time_months event=status covariates=arm,age,sex,stage)

VI. Frailty-Cox Model accounts for random variables shared within clusters such as study groups[20] (example command: rsurvival frailtycox data=surdata time=time_months event=status group=center covariates=age,sex,stage).

VII. Interval-Censored Model evaluates interval-censored models with various parametric distribution methods without any covariate adjustments.[21] For example, the command used to analyse the survival time for different groups will be rsurvival intcens data=surdata time1=time1 time2=time2 event=event group=group.

VIII. Kaplan-Meier (KM) method is used to estimate and visualise unadjusted survival times between different groups.[22] For example, the command for the comparison of survival distributions between drug and control arms without covariate adjustment (example command: rsurvival km data=surdata time=time_months event=status group=arm). This produces the Kaplan-Meier curves and median survival times for each arm.

IX. Left-Truncated Cox Model (LT-Cox) is a survival model used to account for data due to delayed study entry where participants are only observed after the onset of the disease of interest[23] (example command: rsurvival ltcox data=surdata entry=entry time=time event=event covariates=group,age,sex).

X. Log-rank Test is a non-parametric test used to compare survival curves between different groups.[24] An example would be to test if survival distributions differ between drug and control arms without considering any covariates (example command: rsurvival logrank data=surdata time=time_months event=status group=arm), as shown in Figure 2A showing that the survival distributions between drug and control arms is significant (p-value = 0.04).

XI. Nonparametric Interval-Censored Model (NPMLE) evaluates survival interval-censored data without any distribution method.[25] For example, to assess the nonparametric survival functions by treatment group (example command: rsurvival intnp data=surdata time1=time1 time2=time2 event=event group=group).

XII. Parametric Accelerated Failure Time Model (Interval-AFT) assesses the effects of covariates on the time of event of interests with no proportionality assumptions under a chosen parametric distribution such as log-logistic.[26] As such, to determine how group, age and sex affect the survival time under a log-logistic distribution (example command: rsurvival int-aft data=surdata time1=time1 time2=time2 event=event covariates=group,age,sex).

XIII. Parametric Interval-Censored Model (Interval-Parametric) assumes a specific survival distribution such as Weibull to analyse interval-censored data[27] (example command: rsurvival intpar data=surdata time1=time1 time2=time2 event=event covariates=group,age,sex dist=Weibull).

XIV. Semiparametric Interval-Censored Model (Interval-sp) combines nonparametric estimation with time-dependent covariates[28] (example command: rsurvival intsp data=surdata time1=time1 time2=time2 event=event covariates=group,age,sex).

XV. Time-Dependent Cox Model (TD-Cox) is used to analyse time-varying covariates over time[29] (example command: rsurvival tdcox data=surdata time=time_months event=status group=arm covariates=treatment_td,age,sex).

## Concluding remarks

We extend SiPy 0.6.0[8] to SiPy 0.7.0 (codenamed as Keropok), which was released on 05 December 2025, by incorporating R-based ANOVA and survival analysis methods into a consistent Python interface. This design could potentially reduce the steep learning curve for users while improving the accessibility of sophisticated R statistical tools.[30] Future work could focus on extending the capability of SiPy by incorporating other statistical tools such as mixed-effects models to analyse longitudinal data.[31] Continuous updates will also be performed to ensure latest methods and bug fixes are available through SiPy.

### Data availability

Source codes SiPy can be found at https://github.com/mauriceling/sipy while documentation can be found at https://github.com/mauriceling/sipy/wiki. The release page for SiPy 0.7.0 (codenamed as Keropok) can be found at https://bit.ly/SiPy-070.

## Acknowledgments

None.

## Conflicts of interest

The authors declare no conflict of interest.

## References

1. Fitzmaurice GM, Ravichandran C. A Primer in Longitudinal Data Analysis. *Circulation*. 2008;118(19):2005–2010.

2. Murphy JI, Weaver NE, Hendricks AE. Accessible analysis of longitudinal data with linear mixed effects models. *Disease Models & Mechanisms*. 2022;15(5):dmm048025.

3. Fisher RA. The Correlation between Relatives on the Supposition of Mendelian Inheritance. *Transactions of the Royal Society of Edinburgh*. 2012;52(2):399–433.

4. Kaplan EL, Meier P. Nonparametric Estimation from Incomplete Observations. *Journal of the American Statistical Association*. 1958;53(282):457–481.

5. Rich JT, Neely JG, Paniello RC, et al. A practical guide to understanding Kaplan-Meier curves. *Otolaryngology--Head and Neck Surgery*. 2010;143(3):331–336.

6. Brown TR. An Introduction to R and Python for Data Analysis: A Side-By-Side Approach (Chapman and Hall/CRC, Boca Raton), 1st edn. 2023.

7. Colliau T, Rogers G, Hughes Z, et al. MatLab vs. Python vs. R. *Journal of Data Science*. 2017;15(3):355–372.

8. Tan NT, Mugundhan M, Liu T, et al. SiPy - Bringing Python and R to the End-User in a Plugin-Extensible System. *Medicon Medical Sciences*. 2025;8(6):32–41.

9. Satre DD, Leibowitz A, Sterling SA, et al. A randomized clinical trial of Motivational Interviewing to reduce alcohol and drug use among patients with depression. *Journal of Consulting and Clinical Psychology*. 2016;84(7):571–579.

10. Kruskal WH, Wallis WA. Use of Ranks in One-Criterion Variance Analysis. *Journal of the American Statistical Association*. 1952;47(260):583–621.

11. Friedman M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J Am Stat Assoc.* 1937;32(200):675–701.

12. Welch BL. On the comparison of severl mean values: An alternative approach. *Biometrika.* 1951;38(3–4):330–336.

13. Phipson B, Smyth GK. Permutation P-values should never be zero: calculating exact P-values when permutations are randomly drawn. *Statistical Applications in Genetics and Molecular Biology*. 2010;9:Article 39.

14. Van Breukelen GJP. ANCOVA versus change from baseline had more power in randomized studies and more bias in nonrandomized studies. *Journal of Clinical Epidemiology*. 2006;59(9):920–925.

15. Mokesh Rayalu G, Ravisankar J, Mythili GY. MANCOVA for one way classification with homogeneity of regression coefficient vectors. *IOP Conference Series: Materials Science and Engineering*. 2017;263:042134.

16. Ramchandani R, Finkelstein DM, Schoenfeld DA. Estimation for an accelerated failure time model with intermediate states as auxiliary information. *Lifetime Data Analysis*. 2020;26(1):1–20.

17. Dignam JJ, Zhang Q, Kocherginsky M. The Use and Interpretation of Competing Risks Regression Models. *Clinical Cancer Research*. 2012;18(8):2301–2308.

18. Deo SV, Deo V, Sundaram V. Survival analysis - Part 2: Cox proportional hazards model. *Indian Journal of Thoracic and Cardiovascular Surgery*. 2021;37(2):229–233.

19. Mwirigi N. Application of Exponential Distribution in Modeling of State Holding Time in HIV/AIDS Transition Dynamics. *Open Journal of Modelling and Simulation*. 2024;12(04):159–183.

20. Garibotti G, Smith KR, Kerber RA, et al. Longevity and Correlated Frailty in Multigenerational Families. *The Journals of Gerontology Series A: Biological Sciences and Medical Sciences*. 2006;61(12):1253–1261.

21. Sparling YH. Parametric survival models for interval-censored data with time-dependent covariates. *Biostatistics*. 2006;7(4):599–614.

22. Rich JT, Neely JG, Paniello RC, et al. A practical guide to understanding Kaplan-Meier curves. *Otolaryngology–Head and Neck Surgery.* 2010;143(3):331–336.

23. Rennert L, Xie SX. Cox regression model under dependent truncation. *Biometrics*. 2022;78(2):460–473.

24. Gregg ME, Datta S, Lorenz D. A log rank test for clustered data with informative within-cluster group size. Statistics in *Medicine*. 2018;37(27):4071–4082.

25. Zhang Z, Sun J. Interval censoring. *Statistical Methods in Medical Research*. 2010;19(1):53–70.

26. Hu M, Shi X, Gong Z, et al. Collaborative Inference for Accelerated Failure Time Model Using Clinical Center-Level Summary Statistics. *Statistics in Medicine*. 2025;44(23–24):e70279.

27. Lindsey JC, Ryan LM. Methods for interval-censored data. *Statistics in Medicine*. 1998;17(2):219–238.

28. Gu Y, Zeng D, Lin DY. Semiparametric Regression Analysis of Interval-Censored Multi-State Data with An Absorbing State. *Journal of the American Statistical Association*. 2025;120(552):1–21.

29. Zhang Z, Reinikainen J, Adeleke KA, et al. Time-varying covariates and coefficients in Cox regression models. *Annals of Translational Medicine*. 2018;6(7):121–121.

30. Hoffman AM, Wright C. Ten simple rules for teaching an introduction to R. *PLOS Computational Biology*. 2024;20(5):e1012018.

31. Tomiko Yamada Da Silveira L, Carvalho Ferreira J, Maria Patino C. Mixed-effects model: a useful statistical tool for longitudinal and cluster studies. *Jornal Brasileiro de Pneumologia*. 2023;49(2):e20230137.