ReliaGrowR: Modeling and Plotting Functions for Reliability Growth Analysis (RGA)

Keywords: Reliability Growth Analysis, ReliaGrowR, R package, reliability engineering, life data analysis

## Summary & Conclusions

ReliaGrowR is an open-source project for Reliability Growth Analysis (RGA), providing classic models for analyzing and visualizing reliability growth data. The project includes functionality for various reliability growth models, such as the Duane Model, Crow-AMSAA Model, Piecewise NHPP Model, and Piecewise NHPP with Change Point Detection. ReliaGrowR is lightweight, easy to use, and extensible, allowing users to contribute to the project or suggest improvements. The project is available on the Comprehensive R Archive Network (CRAN) and has been verified through example analyses and unit tests to ensure reliability. ReliaGrowR is the only project for RGA currently available on CRAN, making it a valuable resource for reliability engineers and researchers.

## Introduction

Reliability Growth Analysis (RGA) is an important part of reliability engineering, focused on improving system reliability during development and testing. By analyzing failure data, RGA helps engineers identify trends, estimate reliability parameters, and track improvements over time. These insights are essential for assessing the impact of design changes, guiding engineering decisions, and managing risks effectively.

ReliaGrowR [1] is an open-source software (OSS) and non-commercial project for RGA. It provides a set of classic models for analyzing reliability growth data and visualizing results. The project is currently experimental and under active development, with plans to expand functionality and add advanced models in future releases. As an OSS project, ReliaGrowR promotes reproducibility and transparency by allowing users to review the source code, verify calculations, and contribute to development. In addition, open access lowers barriers to entry and promotes equity by making RGA tools available to engineers regardless of resources.

ReliaGrowR is built on the R programming language [2], which is widely used for statistical computing and data analysis. By leveraging R, the project benefits from a large user community and a rich ecosystem of packages that can be integrated with the software. ReliaGrowR is the only package on the Comprehensive R Archive Network (CRAN) dedicated to RGA. It fills a gap not addressed by other reliability engineering packages, such as WeibullR [3] for life data analysis and WeibullR.alt [4] for accelerated life testing. While other open-source projects (e.g., reliability [5]) offer limited RGA support, none currently provide a broad range of models. In contrast, ReliaGrowR implements core reliability growth models, including the Duane model [6], Crow-AMSAA model [7], Piecewise NHPP model [8], and Piecewise NHPP with Change Point Detection [9]. These models are essential for understanding how reliability improves—or degrades—over time as design changes are made to a product or system.

## Design

ReliaGrowR is implemented as an R project, providing modeling and plotting tools for analyzing reliability growth over time. The design emphasizes simplicity and ease of use, with most features available through a small set of core functions. The main functions include rga for fitting reliability growth models and duane for the Duane model. Additional utility functions support plotting, printing results, data conversion, and goodness-of-fit testing.

The project is built mostly from base R functions, with minimal reliance on external packages. This approach keeps the software lightweight and easy to install. The main dependency is the segmented package [10], which supports regression modeling with break or change points. This package provides the core functionality for the Piecewise NHPP model, with or without change-point detection.

ReliaGrowR also suggests a few optional packages for enhanced functionality. These include ellmer [11] for interacting with Large Language Models (LLMs), knitr [12] and rmarkdown [13] for creating dynamic documents, plumber[14] for Application Program Interface (API) development, and testthat [15] for unit testing. However, none of these packages are required for the core functions of the project.

## Usage

### Package

ReliaGrowR is available as a package on CRAN. To install R, follow the instructions provided on the CRAN website for a particular operating system. Once R is installed, install the ReliaGrowR package from CRAN using the following command:

install.packages("ReliaGrowR")

To use the ReliaGrowR package, load the package into the current session with the following command:

library(ReliaGrowR)

### API

For web-based applications and non-R users, the project includes an Application Programming Interface (API) that provides access to its core functionality. The API is implemented using the plumber package, which exposes R functions as endpoints that can be called from other programming languages or web applications. Available endpoints include rga, duane, plot.rga, plot.duane, and rdt. These endpoints allow users to fit models, generate plots, and calculate reliability demonstration plans.

The function grwr\_api starts a local server to host the API, which can be accessed through standard HTTP requests. The server is launched with the following command:

grwr\_api()

Once the server is running, users can access the API endpoints directly or use the package’s built-in web interface. The interface allows users to upload data, select models, and generate plots without writing any code. The API is still under development and has not yet been deployed to a public ser

### The Duane Model

The Duane Model provides a simple and graphical way to observe and analyze whether failure rates are improving as changes are made to a product or system. The Duane Model is a log-log plot of the cumulative Mean Time Between Failures (MTBF) vs cumulative time.

The slope of the line on the plot indicates the rate of reliability growth:

* A positive slope means that the system is improving (reliability is growing, the failure rate is decreasing).
* A zero slope means there is no change in reliability (the system is stable).
* A negative slope indicates that reliability is worsening (the failure rate is increasing).

To use the Duane Model in ReliaGrowR, use the duane\_plot function. This function takes a a vector of failure times and a vector of failure counts, and generates a log-log plot of cumulative MTBF vs cumulative time.

First, set up some dummy cumulative time and failure data:

times <- c(100, 200, 300, 400, 500)  
failures <- c(1, 2, 1, 3, 2)

Next, use the duane\_plot function to create the plot:

fit <- duane(times, failures)  
plot(fit, main = "Duane Model Plot", xlab = "Cumulative Time", ylab = "Cumulative MTBF")



The plot shows the cumulative MTBF on the y-axis and cumulative time on the x-axis, with a fitted line indicating the reliability growth trend. The duane\_plot function returns a duane object with the model results that can be further customized or saved. To view the model results, print the duane object using the print function:

print(fit)

## Duane Analysis Result  
## ----------------------  
## Linear model (log-log scale): log(MTBF) ~ log(Time)  
##   
## Coefficients:  
## Estimate Std. Error  
## (Intercept) 3.6144974 0.35199619  
## log\_cum\_times 0.2013244 0.05624037  
##   
## Log-likelihood: 4.78  
## AIC: -3.55, BIC: -4.72

The duane object contains the model type, parameters, standard error, and goodness-of-fit statistics log-likelihood, AIC, and BIC.

### The Crow-AMSAA Model

The Army Materiel Systems Analysis Activity Model by Crow (Crow-AMSAA) takes failure behavior as a Non-Homogeneous Poisson Process (NHPP) governed by a power law, making the model particularly effective for systems undergoing reliability growth due to continuous improvements.

The Crow-AMSAA model is a statistical model that characterizes the relationship between the cumulative number of failures and cumulative time as the following

where denotes the cumulative number of failures by time , is a scaling parameter, and is the shape parameter.

Similar to the Duane model, the shape parameter indicates whether the system’s reliability is improving or deteriorating.

* If > 1, then the system is improving (reliability is growing, the failure rate is decreasing).
* If = 1, then there is no change in reliability (the system is stable).
* If > 1, then reliability is worsening (the failure rate is increasing).

To use the Crow-AMSAA Model in ReliaGrowR, use the rga function. This function takes a vector of failure times and a vector of failure counts, and generates a plot of cumulative MTBF vs cumulative time with the fitted model.

First, set up some dummy cumulative time and failure data:

times <- c(100, 200, 300, 400, 500)  
failures <- c(1, 2, 1, 3, 2)

Then use the rga function to fit the model and the rga\_plot function to plot the results:

result <- rga(times, failures)  
plot(result, main = "Crow-AMSAA Model Plot", xlab = "Cumulative Time", ylab = "Cumulative Failures")



The rga\_plot function generates a plot showing the cumulative MTBF on the y-axis and cumulative time on the x-axis, with a fitted curve indicating the reliability growth trend. The rga function returns an rga object containing the fitted model parameters. To view the model results, print the rga object:

print(result)

## Reliability Growth Analysis (RGA)  
## ---------------------------------  
## Model Type: Crow-AMSAA   
##   
## Parameters (per segment):  
## Beta: 0.7987 (SE = 0.0562)  
## Lambda: 0.0269  
##   
## Goodness of Fit:  
## Log-likelihood: 4.78  
## AIC: -3.55  
## BIC: -4.72

Similarly, the rga object contains the model type, parameters, standard error, and goodness-of-fit statistics log-likelihood, AIC, and BIC.

### The Piecewise NHPP Model

The Piecewise NHPP model is an extension of the standard NHPP model that includes different segments or phases of time that follow separate failure distributions. This model is particularly useful when a system experiences changes in failure behavior over different development phases, such as the initial, interim and final phases of a development process.

For a Piecewise NHPP model, the cumulative number of failures is modeled as a piecewise function, where each segment has its own parameters. Formally, for time t within phase i, the cumulative number of failures is given by:

where is the cumulative number of failures at time , is the scaling parameter for phase , is the shape parameter for phase , and is an indicator function that is 1 if is in phase and 0 otherwise. The parameters and are estimated from the failure data.

To use the Piecewise NHPP model in ReliaGrowR, first, set up some cumulative time and failure data and specify a breakpoint:

times <- c(25, 55, 97, 146, 201, 268, 341, 423, 513, 609, 710, 820, 940, 1072, 1217)  
failures <- c(1, 1, 2, 4, 4, 1, 1, 2, 1, 4, 1, 1, 3, 3, 4)  
breaks <- 500

Then use the rga function with model type “Piecewise NHPP model” to fit the model and the rga\_plot function to plot the results:

result <- rga(times, failures, model\_type = "Piecewise NHPP", breaks = breaks)  
plot(result, main = "Piecewise NHPP Model Plot", xlab = "Cumulative Time", ylab = "Cumulative Failures")



To view the model results, print the rga object using the print function:

print(result)

## Reliability Growth Analysis (RGA)  
## ---------------------------------  
## Model Type: Piecewise NHPP   
##   
## Breakpoints (original scale):  
## 500   
##   
## Parameters (per segment):  
## Betas: 0.8182, 0.3902  
## Std. Errors (Betas): 0.0465, 0.0324  
## Lambdas: 0.0642, 0.9362  
##   
## Goodness of Fit:  
## Log-likelihood: 17.32  
## AIC: -24.64  
## BIC: -21.10

Now the rga object contains among other information, the parameters for each phase.

### The Piecewise NHPP with Change Point Detection

The Piecewise NHPP with Change Point Detection is an advanced model to identify changes in failure behavior and model system reliability. This method builds on the Piecewise NHPP model by introducing the concept of change points, which represent the time when the underlying failure behavior changes. Detection of change points involves statistical techniques that analyze failure data to automatically identify when the behavior changes, allowing for a more precise segmentation of the model into different distributions.

To use the Piecewise NHPP with Change Point Detection in ReliaGrowR, use the rga function with the model type set to “Piecewise NHPP” and breaks set to NULL. The function will automatically detect change points based on the provided failure data. First, set up some cumulative time and failure data:

times <- c(25, 55, 97, 146, 201, 268, 341, 423, 513, 609, 710, 820, 940, 1072, 1217)  
failures <- c(1, 1, 2, 4, 4, 1, 1, 2, 1, 4, 1, 1, 3, 3, 4)

Then use the rga function with model type “Piecewise NHPP model” to fit the model and the rga\_plot function to plot the results:

result <- rga(times, failures, model\_type = "Piecewise NHPP")  
plot(result, main = "Piecewise NHPP with Change Point Detection Plot", xlab = "Cumulative Time", ylab = "Cumulative Failures")



Print the rga object using the print function to view the model results:

print(result)

## Reliability Growth Analysis (RGA)  
## ---------------------------------  
## Model Type: Piecewise NHPP   
##   
## Breakpoints (original scale):  
## 523.9797   
##   
## Parameters (per segment):  
## Betas: 0.8182, 0.3902  
## Std. Errors (Betas): 0.0465, 0.0324  
## Lambdas: 0.0642, 0.9362  
##   
## Goodness of Fit:  
## Log-likelihood: 17.32  
## AIC: -24.64  
## BIC: -21.10

## Goodness-of-Fit Plots

The project includes wrapper functions for generating goodness-of-fit plots, which help assess how well a model matches the observed data. Two key functions are qqplot.rga for Q-Q plots and ppplot.rga for P-P plots. Both functions take an rga object as input.

times <- c(50, 100, 150, 200, 300, 400, 600, 800, 1000)  
failures <- c(1, 1, 2, 2, 3, 4, 5, 6, 8)  
result <- rga(times, failures)  
qqplot.rga(result)



The qqplot.rga function produces a Q-Q plot that compares the quantiles of the observed data with those of the fitted model. A good fit is indicated when the points closely follow the 45-degree reference line.

Similarly, a P-P plot can be created with:

ppplot.rga(result)



The ppplot.rga function compares the cumulative probabilities of the observed data with those of the fitted model. As with the Q-Q plot, a good fit is indicated when the points closely follow the 45-degree line.

## Data Conversion

Reliability data can be recorded in different formats, such as exact failure times, right-censored times, or failure times grouped into intervals. The weibull\_to\_rga utility function converts data from the format used in the WeibullR package into the format required for RGA analysis in ReliaGrowR. It accepts a vector of failure times and optional vectors of right-censored (suspension) times and interval-censored times, then returns a data frame with cumulative times and failure counts suitable for use with the rga function.

For example, consider the following dataset with failure times, right-censored times, and interval-censored times:

failures <- c(100, 200, 200, 400)  
suspensions <- c(250, 350, 450)  
interval\_starts <- c(150, 300)  
interval\_ends <- c(180, 320)

The data can be converted using weibull\_to\_rga:

result <- weibull\_to\_rga(failures, suspensions, interval\_starts, interval\_ends)  
print(result)

## CumulativeTime Failures  
## 1 100 1  
## 2 265 1  
## 3 465 2  
## 6 1225 1  
## 8 1975 1

The resulting data frame contains cumulative times and failure counts, which can then be passed directly to the rga function for reliability growth analysis.

## Reliability Demonstration Plan

ReliaGrowR provides a utility function, rdt, for calculating the required test time or sample size for a Reliability Demonstration Test (RDT). The calculation is based on the specified target reliability, mission time, and confidence level.

To use rdt, supply the target reliability, mission time, confidence level, and either the test time or sample size. For example, the following code calculates the required test time for a target reliability of 0.9 over a 100-hour mission with 95% confidence and a sample size of 10:

test\_plan <- rdt(target = 0.9, mission\_time = 100, conf\_level = 0.95, beta = 1, n = 10)  
print(test\_plan)

## Reliability Demonstration Test (RDT) Plan  
## -----------------------------------------  
## Distribution: Exponential   
## Weibull Shape Parameter (Beta): 1   
## Target Reliability: 0.9   
## Mission Time: 100   
## Input Sample Size (n): 10   
## Required Test Time (T): 284.33

The rdt function returns the required test time to achieve the specified reliability demonstration.

## Testing

ReliaGrowR was verified through example analyses and unit tests to ensure the package performs as expected. Unit tests follow best practices for package development and use the testthat framework. They cover core functionality such as model fitting, plotting, and change-point detection, and run automatically during development to ensure reliability and correctness. All examples and tests are included in the package source code, allowing users to review and run them.

The package has also been tested on different operating systems and R versions to confirm compatibility and performance. Test results are documented on CRAN. Full documentation and working examples are available on the project website, where users can also contribute or report issues.

## Extensions

ReliaGrowR is designed with extensibility in mind, allowing users to add custom models or new features as needed. Its accessibility and ease of use also make it well-suited for education, helping students and professionals learn RGA concepts through practical examples and hands-on experimentation [15]. While the package includes basic plotting functions, users can enhance visualizations or create interactive applications by integrating other libraries [16], [17].

The package is currently experimental and under active development, with new features and improvements added regularly. Planned updates include support for advanced models such as Bayesian reliability growth, as well as additional goodness-of-fit tests. Work is also underway to improve the API and web interface for easier access to the core functionality.

As an open-source project, ReliaGrowR welcomes contributions from the community. Users can submit pull requests, report issues, or suggest enhancements through the public repository. This collaborative approach ensures that the package evolves in step with user needs and advances in reliability engineering.

## References

1. Placeholder, “ReliaGrowR: Reliability Growth Analysis”, R package version 0.1, 2024, <doi:10.32614/CRAN.package.ReliaGrowR>, <https://cran.r-project.org/package=ReliaGrowR>.
2. R Core Team, “R: A Language and Environment for Statistical Computing”, R Foundation for Statistical Computing, Vienna, Austria, 2024, <https://www.R-project.org/>.
3. D. Silkworth, J. Symynck, “WeibullR: Weibull Analysis for Reliability Engineering”, R package version 1.2.1, 2022, <https://CRAN.R-project.org/package=WeibullR>.
4. D. Silkworth, “WeibullR.ALT: Accelerated Life Testing Using ‘WeibullR’”, R package version 0.7.2, 2022, <https://CRAN.R-project.org/package=WeibullR.ALT>.
5. M. Reid, “Reliability – a Python library for reliability engineering”, Version 0.8.2, 2022, <https://doi.org/10.5281/ZENODO.3938000>.
6. J. T. Duane, “Learning Curve Approach to Reliability Monitoring”, IEEE Transactions on Aerospace, vol. 2, no. 2, pp. 563-566, April 1964, doi: 10.1109/TA.1964.4319640.
7. L.H. Crow, “Reliability analysis for complex repairable systems.”, Reliability and biometry: Statistical analysis of lifelength, pp. 379-410, 1974.
8. H. Guo, A. Mettas, G. Sarakakis, P. Niu, “Piecewise NHPP models with maximum likelihood estimation for repairable systems”, 2010 Proceedings - Annual Reliability and Maintainability Symposium (RAMS), San Jose, CA, USA, 2010, pp. 1-7, <doi:10.1109/RAMS.2010.5448029>.
9. S. Fasola, V.M.R Muggeo, K. Kuchenhoff, “A heuristic, iterative algorithm for change-point detection in abrupt change models”, Computational Statistics, 33, pp. 997-1015, 2018.
10. V.M.V Muggeo, “segmented: an R Package to Fit Regression Models with Broken-Line Relationships”, R News, 8/1, pp. 20-25, <https://cran.r-project.org/doc/Rnews/>.
11. H. Wickham, J. Cheng, A. Jacobs, G. Aden-Buie, B. Schloerke, “ellmer: Chat with Large Language Models”, R package version 0.3.2, 2025, <https://CRAN.R-project.org/package=ellmer>.
12. Y. Xie, “knitr: A General-Purpose Package for Dynamic Report Generation in R”, R package version 1.49, 2024, <https://yihui.org/knitr/>.
13. J. Allaire, Y. Xie, C. Dervieux, J. McPherson, J. Luraschi, K. Ushey, A. Atkins, H. Wickham, J. Cheng, W. Chang, R. Iannone, (2024). “rmarkdown: Dynamic Documents for R”, R package version 2.29, 2024,  
    <https://github.com/rstudio/rmarkdown>.
14. A. B. Schloerke, “plumber: An API Generator for R”. R package version 1.3.0, 2025, <https://CRAN.R-project.org/package=plumber>.
15. H. Wickham, “testthat: Get Started with Testing”, The R Journal, 3, pp. 5-10, 2011,  
    <https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf>.
16. Placeholder, “ReliaLearnR: An Interactive Introduction to Reliability Analysis”, R package version 0.1.1, 2023, <doi:10.32614/CRAN.package.ReliaLearnR>, <https://CRAN.R-project.org/package=ReliaLearnR>.
17. Placeholder, “ReliaPlotR: Interactive Reliability Plots, R package version 0.2.0, 2023, <doi:10.32614/CRAN.package.ReliaPlotR>, <https://CRAN.R-project.org/package=ReliaPlotR>.
18. Placeholder, “ReliaShinR: A Shiny App for Reliability Analysis”, R package version 0.2.0, 2023, <doi:10.32614/CRAN.package.ReliaShiny> <https://CRAN.R-project.org/package=ReliaShiny>.