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 aspect of reliability engineering, aimed at improving system reliability during development and testing. By analyzing failure data, RGA helps engineers identify trends, estimate reliability parameters, and visualize reliability improvement over time. This insight is crucial for assessing the impact of design changes, guiding engineering decisions, and managing risks effectively.

ReliaGrowR [1] is an RGA open-source software (OSS) project for non-commercial use. The project provides a suite of classic models for analyzing reliability growth data and visualizing results. The project is currently experimental and under active development, with new functionality and advanced models planned for future development releases. As an OSS project, ReliaGrowR enables reproducibility and transparency, allowing users to view the source code, verify calculations, and contribute to the project’s development. In addition, OSS lowers barriers to entry and promotes equity by providing access to reliability engineers regardless of resources.

ReliaGrowR is built on the widely adopted R programming language [2], which is known for its extensive statistical and data analysis capabilities. By leveraging R, the project benefits from a large and active community of users and developers, as well as a rich ecosystem of packages and libraries that can be integrated with the project. ReliaGrowR is the only package dedicated to RGA currently available on the Comprehensive R Archive Network (CRAN). The project fills the gap for RGA functionality from other reliability engineering projects, 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 functionality, none currently support a broad range of RGA models. In contrast, ReliaGrowR supports 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 changes are made to a product or system.

## Design

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

The project is built largely from base functions, with limited dependencies on external packages. This approach ensures that the project remains lightweight and easy to install. The project’s primary dependency is the segmented package [10], which is used for regression modeling with break or change points. This package provides the underlying functionality for the Piecewise NHPP model with or without change point detection.

ReliaGrowR also suggests a few additional packages for improved functionality. These packages include ellmer [11] for chatting with Large Language Models (LLMs), knitr [12] and rmarkdown [13] for creating dynamic documents, and testthat [14] for unit testing. However, these packages are not required for the core functionality 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)

### Application Programming Interface

For web-based applications and non-R users, the project includes an Application Programming Interface (API) for accessing the core functionality. The API is implemented using the plumber [14] package, which allows R functions to be exposed as a set of endpoints that can be called from other programming languages or web applications. The API endpoints include rga, duane, plot.rga, plot.duane, and rdt. These endpoints provide access to the core functionality of the project, allowing users to fit models, generate plots, and calculate reliability demonstration plans.

The function grwr\_api spins up a local server to host the API that can be accessed via HTTP requests. The server can be started with the following command:

grwr\_api()

Once the server is running, users can access the API endpoints using HTTP requests or use the web interface provided by the package. The web interface allows users to input data, select models, and generate plots without needing to write any code. The API is in developement and is not deployed to a public server at this time.

### 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).

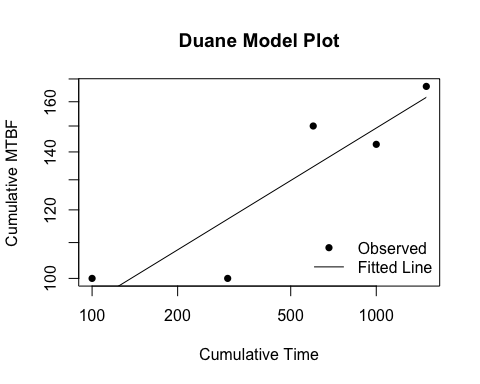
To use the Duane Model in ReliaGrowR, use the duane\_plot function. This function takes a a vector of failure times and a vector of failure counts, and generates a log-log plot of cumulative MTBF vs cumulative time.

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).

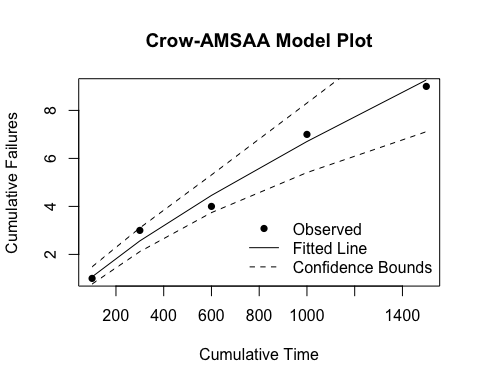
To use the Crow-AMSAA Model in ReliaGrowR, use the rga function. This function takes a vector of failure times and a vector of failure counts, and generates a plot of cumulative MTBF vs cumulative time with the fitted model.

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:

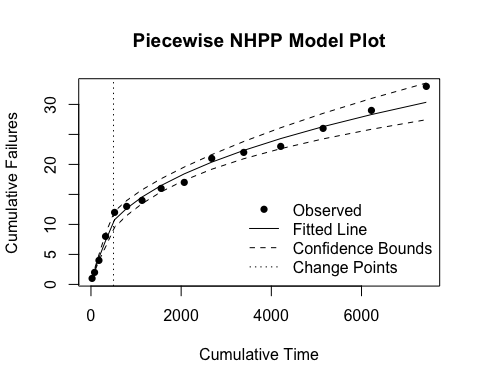
where is the cumulative number of failures at time , is the scaling parameter for phase , is the shape parameter for phase , and is an indicator function that is 1 if is in phase and 0 otherwise. The parameters and are estimated from the failure data.

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

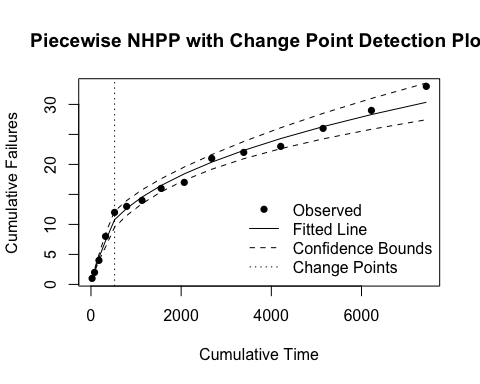
The Piecewise NHPP with Change Point Detection is an advanced model to identify changes in failure behavior and model system reliability. This method builds on the Piecewise NHPP model by introducing the concept of change points, which represent the time when the underlying failure behavior changes. 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:

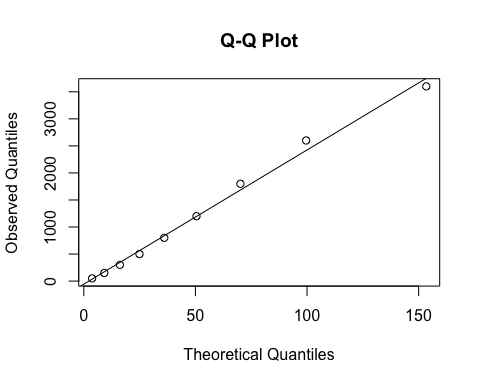
print(result)

## 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 to generate goodness-of-fit plots for the fitted models. These plots help assess how well the model fits the data. To generate goodness-of-fit plots, use the qqplot.rga and ppplot.rga functions for Q-Q and P-P plots, respectively. These functions take an rga object as input. For example, to create a Q-Q plot for a Crow-AMSAA model:

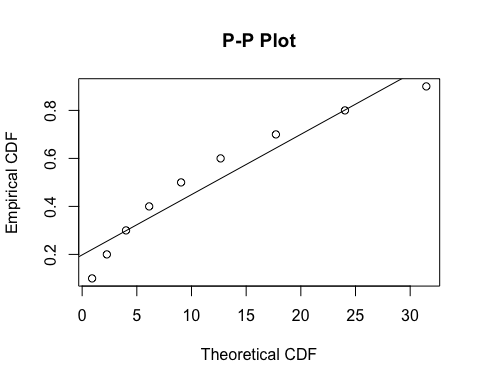
times <- c(50, 100, 150, 200, 300, 400, 600, 800, 1000)  
failures <- c(1, 1, 2, 2, 3, 4, 5, 6, 8)  
result <- rga(times, failures)  
qqplot.rga(result)



The qqplot.rga function generates a Q-Q plot comparing the quantiles of the observed data to the quantiles of the fitted model. A good fit is indicated by points closely following the 45-degree reference line.

Similarly, to create a P-P plot:

ppplot.rga(result)



The ppplot.rga function generates a P-P plot comparing the cumulative probabilities of the observed data to the cumulative probabilities of the fitted model. Again, a good fit is indicated by points closely following the 45-degree reference line.

## Data Conversion

Reliability data can be represented in different formats, such as exact failure times, right censored times, or failure times over specified intervals. The weibull\_to\_rga utility function converts data from the format used in the WeibullR package to the format required for RGA analysis in ReliaGrowR. This function takes a vector of failure times, and optional vectors of right censored or suspension times, and interval censored times, and returns a data frame with cumulative time and failure counts suitable for use with the rga function.

Consider the following example data that includes 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)

Use the weibull\_to\_rga function to convert the data:

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 time and failure counts that can be used with the rga function for RGA analysis.

## Reliability Demonstration Plan

ReliaGrowR includes a utility function rdt to calculate the required test time or sample size for a Reliability Demonstration Test (RDT) based on specified target reliability, mission time, and confidence level.

To use the rdt function, specify the target reliability, mission time, confidence level, and either the test time or sample size. For example, to calculate the required test time for a target reliability of 0.9 over a mission time of 100 hours with a confidence level of 0.95 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 that the package performs as expected. The unit tests follow best practices for package development and use the testthat package for testing. The tests cover the core functionality, including model fitting, plotting, and change point detection. These tests run automatically during package development to ensure reliability and correctness. All examples and unit tests are included in the package source code and are available for users to review and run. ReliaGrowR was also tested on different operating systems and R versions to ensure compatibility and performance. The results of these tests are documented on CRAN. Full documentation and working examples are available on the project website, where users can also contribute to or report issues with the package.

## Extensions

ReliaGrowR is designed with extensibility in mind, enabling users to incorporate custom models or additional features as needed. The accessibility and ease of use make the package particularly well-suited for educational purposes, allowing students and professionals to explore RGA concepts through practical examples and hands-on experimentation [15]. While the package includes basic plotting capabilities, users can easily enhance visualizations or develop interactive applications by integrating other libraries [16], [17].

ReliaGrowR is currently experimental and under active development, with new features and improvements added on a regular basis. Planned enhancements include support for advanced models, such as Bayesian reliability growth models, and additional goodness-of-fit tests. The project also aims to improve the API and web interface for easier access to the core functionality.

As an open-source project, contributions from the community are welcome. Users can submit pull requests to the public repository, report issues, or propose enhancements. This collaborative approach ensures that the package continues to evolve in response to 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>.