SENG 438 - Lab Report #5

Group 11:
Ryan Khryss Obiar (30151002)
Armin Sandhu (30143482)
Parsa Kargari (30143368)
Jayden Mikulcik (30139637)

Repo Link:
https://github.com/seng438-winter-2024/seng438-a5-arminsandhu

1 Introduction

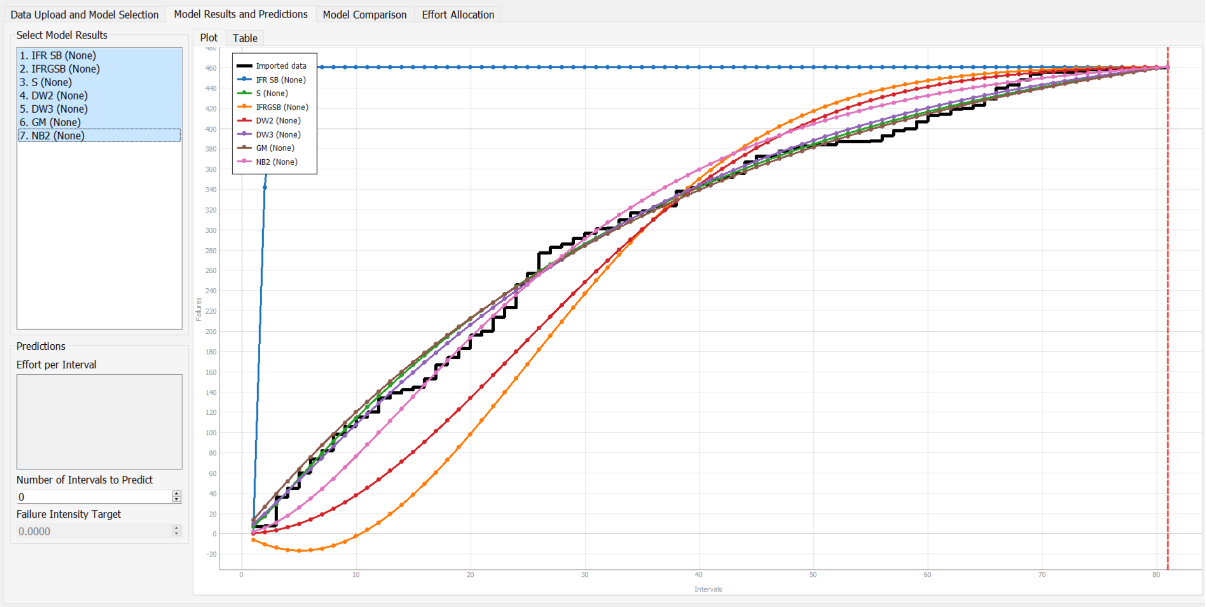
For this assignment our group is tasked to dive into both Reliability growth testing and Reliability assessment using Reliability Demonstration Chart (RDC), both of which are essential practices to understand in order to have a proper grasp on the Software testing that gets applied in industry. The first part of the lab is focussed on Reliability growth testing in which we will be using START (our chosen tool) to measure the failure rate MTTF and the reliability of the system under test through analyzing the test data provided. The end goal of this half of the lab is to gain an understanding of reliability growth testing and why it is useful. The latter part of this lab focuses on assessment using a reliability demonstration chart - which is an efficient way of checking whether the target failure rate or MTTF is met or not. It will give us the chance to familiarize ourselves with RDC and analyze the testing for a given MTTF of the system under test through plotting the test data. Overall this assignment will be used as a great introduction to the assessment of failure data.

2 Assessment Using Reliability Growth Testing

Dataset used:
Failure_Data_Set > Failure_Count > DATASET3.DAT
Hazard functions Available:

1. IFR Salvia & Bollinger
2. IFR generalized Salvia & Bollinger
3. S Distribution
4. Discrete Weibull (Order 2)
5. Discrete Weibull(Type III)
6. Geometric
7. Negative Binomial (Order 2)
8. Truncated Logistic

MVF with all:



Select Model Results

1. IFR SB (None)
2. IFRGSB (None)
3. S (None)
4. DW2 (None)
5. DW3 (None)
6. GM (None)
7. NB2 (None)

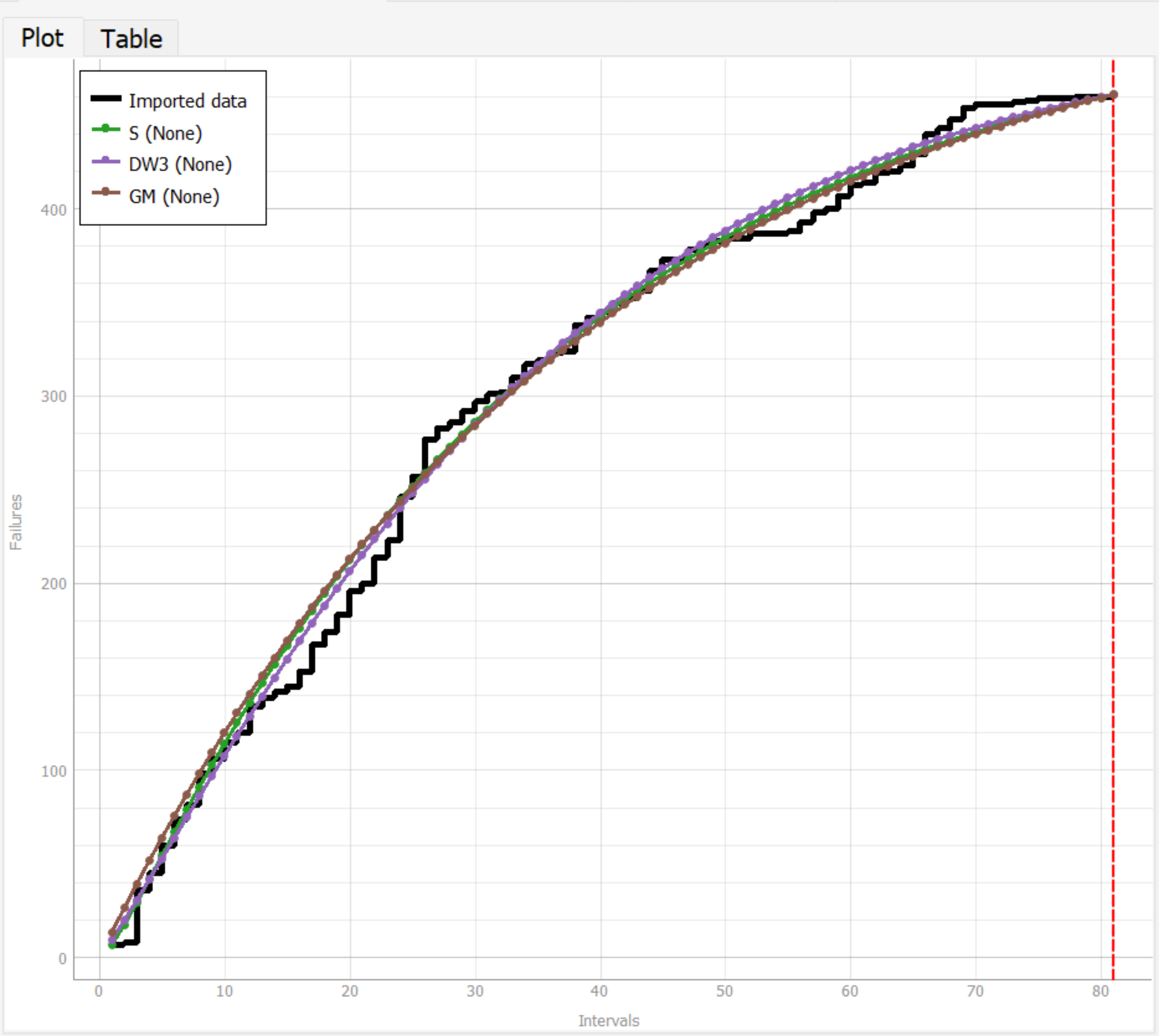
Predictions

Effort per Interval

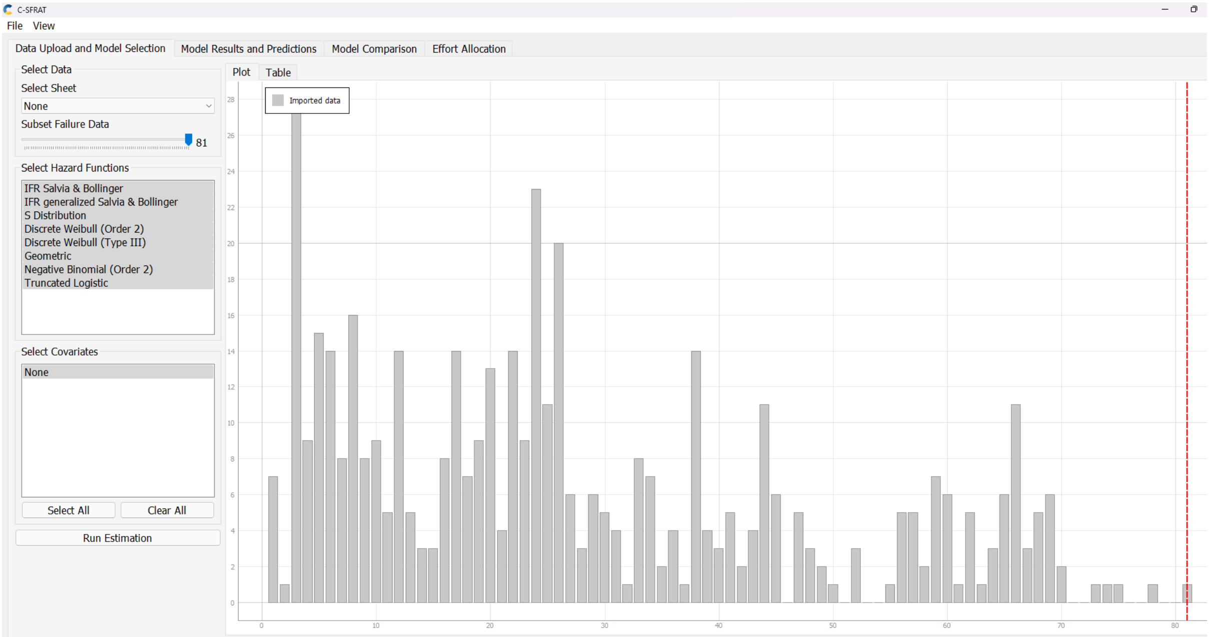
Number of Intervals to Predict: 0

Failure Intensity Target: 0.0000

MVF (Minimum Viable Feature) best functions:



Intensity Graph:



Select Data

Select Sheet: None

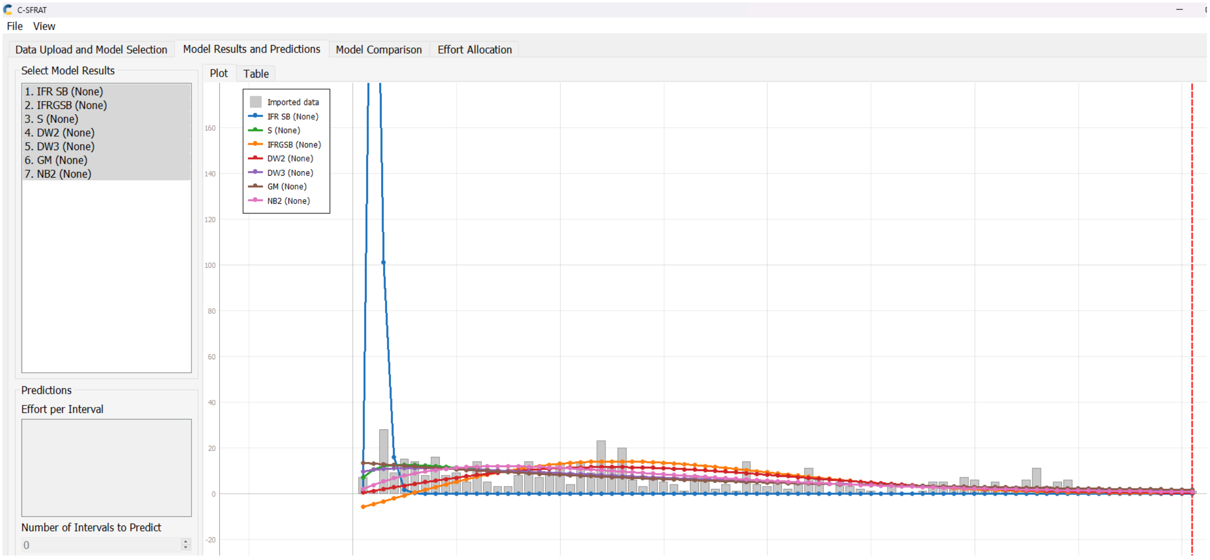
Subset Failure Data: 81

Select Hazard Functions

- IFR Salvia & Bollinger
- IFR generalized Salvia & Bollinger
- S Distribution
- Discrete Weibull (Order 2)
- Discrete Weibull (Type III)
- Geometric
- Negative Binomial (Order 2)
- Truncated Logistic

Select Covariates: None

Intensity Graph for all models:



Select Model Results

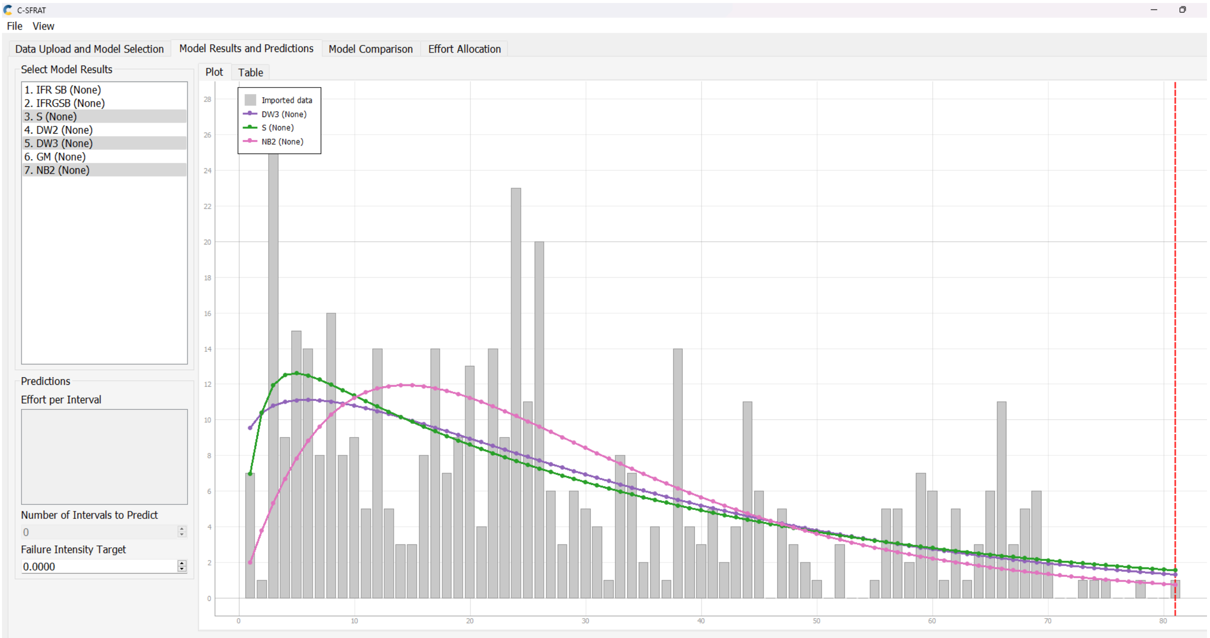
1. IFR SB (None)
2. IFRGSB (None)
3. S (None)
4. DW2 (None)
5. DW3 (None)
6. GM (None)
7. NB2 (None)

Predictions

Effort per Interval

Number of Intervals to Predict: 0

Best fit:



Select Model Results

1. IFR SB (None)
2. IFRGSB (None)
3. S (None)
4. DW2 (None)
5. DW3 (None)
6. GM (None)
7. NB2 (None)

Predictions

Effort per Interval

Number of Intervals to Predict: 0

Failure Intensity Target: 0.0000

Result of model Comparison:

Though we had depicted 3 models, for our analysis and comparison we decided to select the 2 best models, being the S distribution (S) and the Discrete Weibull 3 (DW3). When looking at these two models for the MVF Charts, we can conclude the following:

- Both models are generally good for modelling the cumulative failures throughout each interval.
- The S distribution model closely follows the trend of the actual failure data
- The DW3 also closely follows the trend of the actual failure data

This shows that both models made reliable predictions based on the input failure data and correctly showcase the trend of the reliability of the SUT. Again, Looking at the trends of the imported data and the 2 chosen models, they all show that the curve of the MVF plot flattens out at the top, insinuating that the number of failures during the intervals stabilises, indicating reliability in the system.

Now, looking at the two models for the intensity graphs, we can conclude the following:

- The imported data (failure data) depicted a right skewed (positively skewed) graph, indicating that the amount of failures per interval decreases as time goes on.
- The DW3 also depicts a right skewed graph, highlighting the highest point of the intensity of the imported data, and accurately depicting the decrease in failures per interval
- The S distribution model also depicts a right skewed graph, again highlighting the highest points of the intensity of the imported data and following the decreasing trend on failures per interval across time. Looking at this graph, we could infer that it follows the imported data trend just slightly better than the DW3 Model.

Result of range analysis:

Looking at the range tests we as a group cumulatively decided that the entire part of the data was valid for the our range test as all the data appeared to be within a reasonable range and no outliers were detected in the dataset we chose which was Failure-Count/DATASET3.DAT. After deciding the range and completing an analysis of the range test the end of the test appeared to taper off into a more linear line. Overall in the range test we decided that the entire range was valid and the data spikes were able to taper off successfully.

Target Failure Rate:

Using Dataset 3 from Failure counts, we are given three columns, the two major columns will be the time intervals, as well as the cumulative failures. There are 81 instances in the dataset, with the cumulative amount of failures at the end is 461. The first 30 are through the VALIDATION phase, the next 10 are through FIELD trials, then the last few instances are during system OPERATION. Per instance, we are expecting (461/81) around 5.7 failures per instance.

Advantages and Disadvantages of Reliability Growth Analysis:

In essence, through using reliability growth analysis these are some pros and cons we believe encapsulate this method

Advantages:

- Identify problem areas in a software
    - Reliability growth models could be very handy as they help organizations find issues in the system which will allow them to concentrate on fixing them to make the whole system more reliable.
- Quantification of Reliability of a System
    - This method lets us directly test and quantify the reliability of a system, not only letting an organization know more about a software's health, but to future enhance that software to ensure it is as sound as possible. It takes in failure data and directly correlates it to the reliability of the software, a good measure on the robustness of the software.

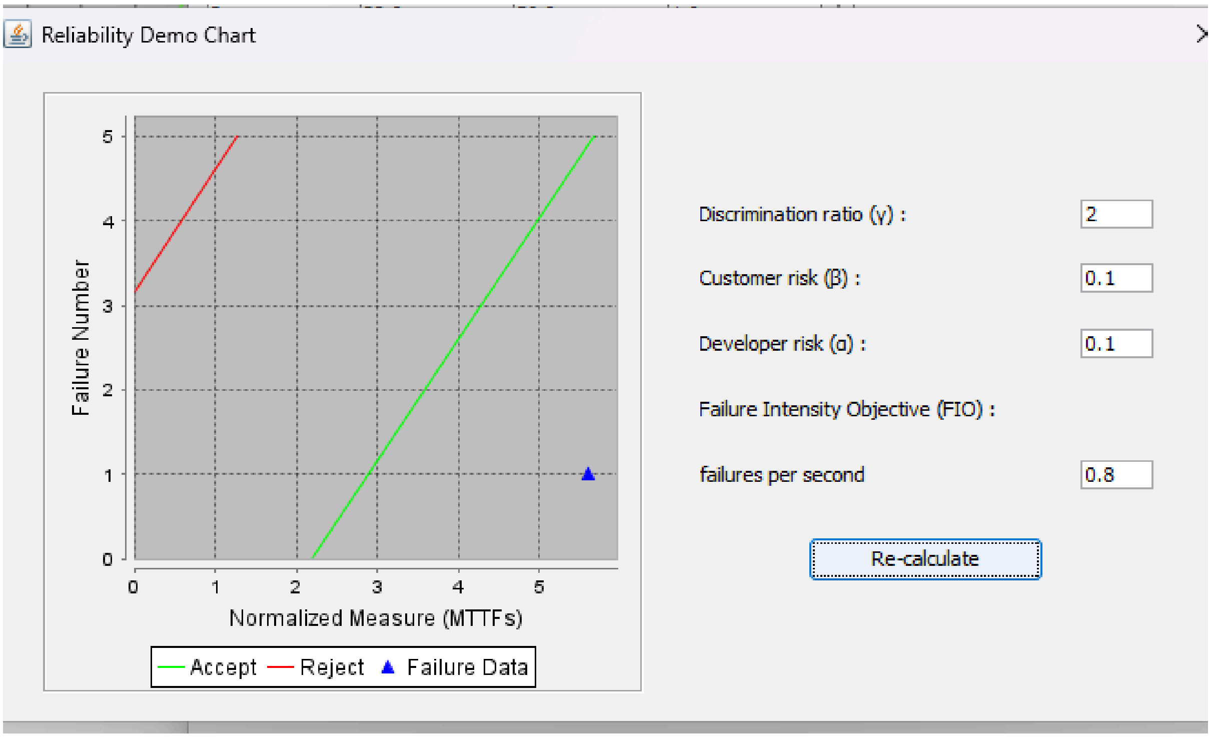
Disadvantages:

- Outputs a predictive analysis of the reliability of a SUT
    - This method does have its perks, but one thing to consider is that these quantifications are only predictions and these results may be different than what could be seen (due to external factors regarding the software system)
- Need of data
    - This method requires readily available data on hand about the failures of the system. This data could be hard to get and therefore would create difficulties in assessing the reliability of a system.

3 Assessment Using Reliability Demonstration Chart

3 plots for MTTFmin, twice and half of it for your test data

Double MTTF:



Discrimination ratio (γ) : 2

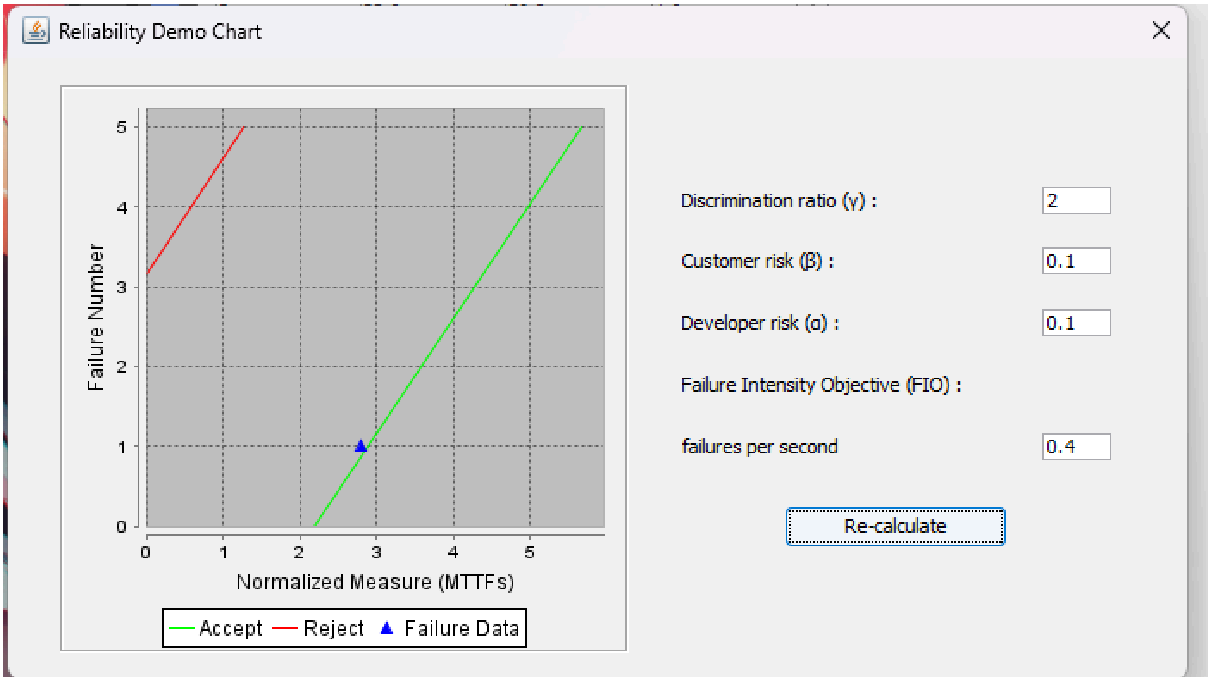
Customer risk (β) : 0.1

Developer risk (α) : 0.1

Failure Intensity Objective (FIO) :

failures per second 0.8

MTTF:



Discrimination ratio (γ) : 2

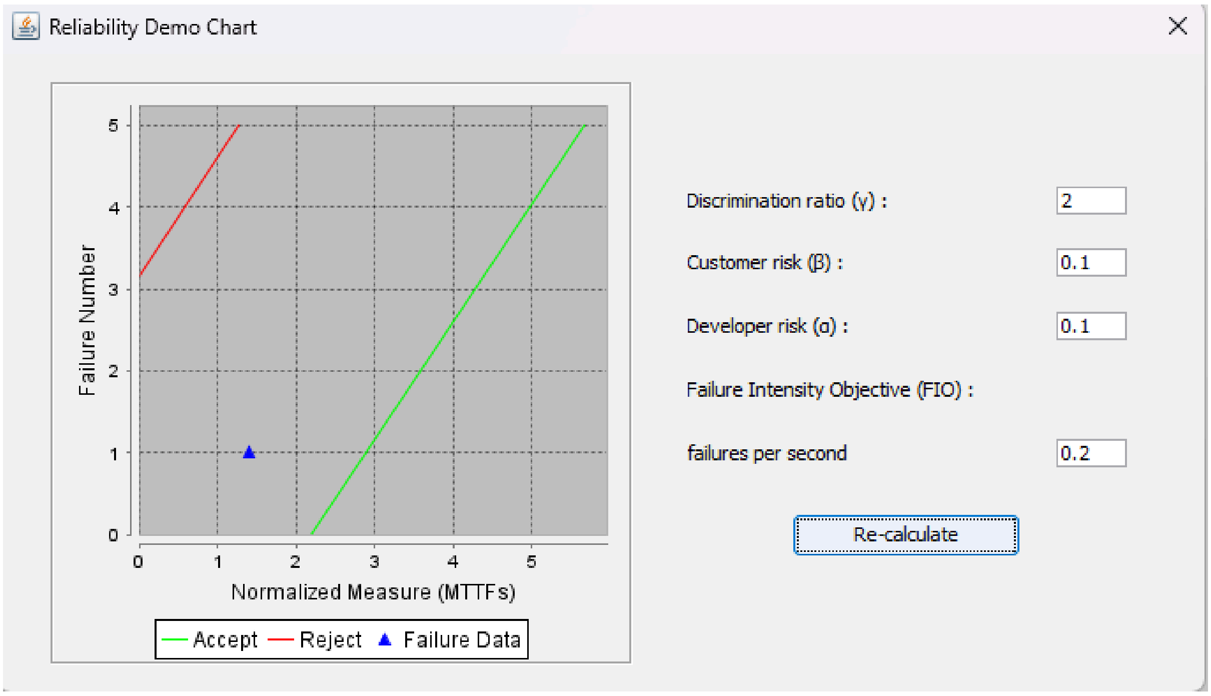
Customer risk (β) : 0.1

Developer risk (α) : 0.1

Failure Intensity Objective (FIO) :

failures per second 0.4

Half MTTF:



Discrimination ratio (γ) : 2

Customer risk (β) : 0.1

Developer risk (α) : 0.1

Failure Intensity Objective (FIO) :

failures per second 0.2

Evaluation and Justification of MTTFmin:

With a value of 0.4 as our failures per second, the blue triangle, failure data, falls within the accepted and neutral regions of the RDC. Since half of these failures per second rate lies within the neutral area, and the double of this value falls within the accepted area, it indicates a favourable trend towards meeting reliability.

Advantages and Disadvantages of RDC:

Again, through using RDC and getting familiar with this method, these are some pros and cons we believe encapsulate it:

Advantages:

- Visual representation
    - Using this method could help users easily understand if failure data is within ranges of acceptable, neutral, or rejected. It is easy to comprehend and thus make analysis of a system more efficient.
- Easy to use to help organizations and developers in deciding how sound a product is.
    - Can be easily used and accessed (on both Mac and Windows) to help determine whether or not a product should be released or if a product would need more support and development needed on it.

Disadvantages:

- Shows the demonstration with the given data
    - If you're trying to predict in the future, it might be different as it's only showing visualizations of what's given, even if it's wrong this data has to be accurate (similar to Reliability Growth Analysis)

- Use of Complex formulas
    - One using RDC may not completely understand why the results are as they are due to the complex statistical formulas used. This can be unclear when trying to understand results.

4 Comparison of Results

The key differences of the two techniques used, are discussed in the next section.

As for our results and testing:
In our first part, we found that the “S distribution” and “Discrete Weibull” were to be the best fits for modeling the failures. In our second part, instead of docusing on the failures over time, we focused on whether the system met different reliability targets set by us. We plotted RDC plots for MTTFmin, twice MTTFmin, and half MTTFmin which all indicated different risk acceptance levels (accepted, and neutral).
The Reliability Growth Testing method that we used, showed that it was more sensitive to variations of the data, as the intensity graph showed fluctuations. On the other hand. The RDC did not measure sensitivity, but it assessed the measure due to predeterminced values and conditions.
In our part 2 of RDC, we noticed how decreasing the failures per second, would also decrease the MTTFs, pushing the failure data towards the failure data.

Furthermore, RDC provided us with a clear, binary outcome that is more straightforward in making decisions when compared to our Reliability Growth Testing method in part 1. Overall, both parts showed valuable results and information to us, with Part 1 providing a more dynamic, retailed look at reliability growth and Part 2 providing us with a more straightforward and static information. We also noticed how RDC may serve as a “validation step” before deployment or decisions, as it is a binary decision of pass/fail, whereas, the growth testing results showed us how it can be used for ongoing system reliability improvements.

5 Discussion on Similarity and Differences of the Two Techniques

In discussing the similarities and differences between Reliability Growth Testing and Assessment Using Reliability Demonstration Chart (RDC), it's crucial to consider their distinct objectives, methodologies, and applications within reliability assessment. Reliability Growth Testing aims to iteratively assess and improve the reliability of a system over time through continuous testing. This approach is good for dynamic systems undergoing ongoing development. In contrast, RDC focuses on demonstrating whether a system meets predefined reliability targets within a limited testing period. It assesses the system's reliability based on collected failure data at specific time points. Methodologically, Reliability Growth Testing involves iterative testing processes, where failure data is continuously collected, analyzed, and utilized to refine reliability estimates using things like statistical models (Geometric, Laplace, etc.). On the other hand, RDC utilizes graphical representations, such as charts, to visualize the relationship between failure data and predefined reliability targets. It assesses whether the observed failure data aligns with the expected reliability levels. While both techniques share the common goal of assessing and enhancing system reliability, they differ significantly in their methodologies and objectives. Understanding these differences is essential for selecting the most appropriate approach based on project requirements and constraints, ensuring effective reliability assessment and improvement strategies are implemented.

6 How the team work/effort was divided and managed

For this lab specifically, we decided to work on the whole lab together to ensure that our understanding of failure data assessment was sound and that everyone was able to get some hands-on experience with the tools used in this lab. We had issues running the tools and executing them with the data provided to us on the Macbook computers in the group so working together in this manner really helped everyone in the group understand each part of this lab. The environment was one in which we could ask questions, and especially one that ensured we were all on the same page which we believe was paramount to our understanding of Reliability Growth Testing and assessment using a Reliability Demonstration Chart.Initial steps involved a few members understanding how to change the input files while the others tried the two different RDC methods to use which helped cut down in time allowing for efficiency to thrive in this assignment.

7 Difficulties encountered, challenges overcome, and lessons learned

During our time completing this lab, as mentioned above a significant challenge that rose was that not all team members could execute the given tools with the data provided, ie. those on Macbook computers. This forced us to rethink our usual divide-and-conquer strategy. Instead, we opted for a collaborative approach, where everyone pitched in to find a solution. ANother difficulty was our unfamiliarity with the software. We had to invest extra time to understand its ins and outs fully. This involved delving into tutorials, documentation, and collaborating with others to grasp its functionalities effectively. However, through teamwork and perseverance, we managed to overcome these hurdles. This experience emphasized the importance of adaptability and teamwork in navigating challenges. It also strengthened our understanding of the analysis process and fostered closer bonds within the team.

8 Comments/feedback on the lab itself

Overall feedback on this lab was good as we got to use and understand and used another few tools in this industry. This coupled with the critical thinking it took to come up with each method we used meant to our group a lab that is essential to complete in order to round out our skills as software engineers. Though we went through a few difficulties completing the lab on our macbooks, the general consensus of our group was that the lab was well designed from the content we had to implement.