Continouous Time Markov (CTMC) with R

In this exercise, you will use R language to develop simple Continuous Time Markov Chain (CTMC) dependability models.

Exercise 1) Please install R and RStudio as follows

Installation Guide for R and RStudio on Windows

Introduction

This document provides a step-by-step guide on how to install R and RStudio on Windows operating systems. R is a free software environment for statistical computing and graphics, and RStudio is a popular integrated development environment (IDE) for R.

Installing R

Step 1: Download R

1. **Navigate to the CRAN Website**
   - Open your web browser and go to the Comprehensive R Archive Network (CRAN) at The Comprehensive R Archive Network .
2. **Download R for Windows**
   - Click on the link titled “Download R for Windows”.
   - In the "base" subdirectory, click on “Download R x.x.x for Windows” (where x.x.x represents the latest version number).

Step 2: Install R

1. **Run the Installer**
   - Locate the downloaded file and double-click on it to run the installer.
2. **Follow Installation Instructions**
   - Proceed through the installation wizard. The default settings are generally appropriate for most users.
   - Select the components to install (you can typically leave these at their default settings).
   - Choose the installation directory (or use the default location).
   - Select the Start Menu folder (or use the default setting).
   - Choose additional tasks (such as creating a desktop icon or editing the system PATH) based on your preferences.

Installing RStudio

Step 1: Download RStudio

1. **Navigate to the RStudio Download Page**
   - Open your web browser and visit the RStudio download page at Posit .
2. **Download RStudio Desktop**
   - Under “RStudio Desktop,” find the Free version.
   - Click on the link for the Windows version to download the installer.

Step 2: Install RStudio

1. **Run the RStudio Installer**
   - Once the download is complete, locate and run the installer file.
2. **Follow Installation Instructions**

- Proceed through the RStudio installation wizard.
- Choose the installation directory (or use the default location).
- Complete the installation process.

Verifying the Installation

1. **Open R and RStudio**
   - After installation, locate R and RStudio in your Start Menu or Desktop and open them to ensure they are correctly installed.
2. **Check for Errors**
   - When opening R and RStudio, look for any error messages. If there are none, the installation was likely successful.

Exercise 2): Run the following example.

Problem: We have a PC with one disk. The MTTF of the disk is 3 years. The recovery of the disk takes, on average, 6 hours.

Open RStudio and run the following script

```r
rm(list = ls())
# Install and load the 'expm' package if not already installed
if (!require(expm)) install.packages("expm")
library(expm)


# Define the transition rates
mttf <- 3 * 365 * 24 # MTTF in hours
mttr <- 6 # MTTR in hours

lambda <- 1 / mttf # Failure rate
mu <- 1 / mttr # Repair rate

cat("lambda: ", lambda, "\n")
cat("mu:", mu, "\n")

# Create the rate matrix

#Q = | q00, q01|
#    | q10, q11|

# Adjusted Generator Matrix for CTMC
# In R, the c() function is used to create vectors.
# The letter c stands for "combine"
# In R, the matrix() function is used to create matrices.
# A matrix is a two-dimensional, rectangular layout of data elements,
# arranged in rows and columns. Each element in a matrix can be accessed
# using its row and column indices.
# byrow = TRUE means that the elements in the data vector are filled into
# the matrix row by row.

Q <- matrix(c(
  -lambda, lambda,
  mu, -(mu)
), byrow = TRUE, nrow = 2)

# Compute the steady-state probabilities
Q_t <- t(Q)                                # Transpose of Q
```

```
39 Q_t[1,] <- rep(1, ncol(Q))                 # Replace first row with 1s for the sum constraint
40 steady_state <- solve(Q_t, c(1,0))       # Solve for steady-state probabilities
41
42 # Steady state availability
43 availability <- sum(steady_state[1])     # Sum of probabilities of Operational and Degraded states
44 # NOTE. In models with more states you can sum more states, e.g., sum(steady_state[1:2]) will sum states 1 and 2
45 # Output the results
46 print(paste("Steady State Availability:", availability))
```

Deliverables:

- From the `Transition matrix` (line 32), provide the Markov Chain Model.
- A brief report summarizing the steady state availability of the system.

Exercise 3): Calculate the steady state availability of a RAID 1 system

RAID (Redundant Array of Independent Disks) is a data storage virtualization technology that combines multiple physical disk drive components into one or more logical units for data redundancy, performance improvement, or both. There are many RAID modes, but in this exercise, you will concentrate only on RAID 1. RAID 1 is all about redundancy through mirroring. It involves copying (mirroring) the same data on two or more disks. This setup is simple:

- **Redundancy**: Provides 100% redundancy as the same data is written on two or more drives.
- **Performance**: Read operations can be performed faster as data can be read from two disks simultaneously. Write operations, however, have no speed benefit.
- **Capacity**: The effective storage capacity is only half of the total disk capacity because the same data is stored on each disk.
- **Fault Tolerance**: Can withstand a single drive failure without data loss.

You, as a **reliability engineer**, should perform the following tasks:

1. Develop a CTMC model for RAID 1 steady-state system availability in R. Assume an MTTF of 3 years for each disk and an MTTR of 6 hours for disk replacement.
2. Provide the steady-state system availability of the system.

Deliverables:

- R script containing the CTMC model and calculations.
- A brief report summarizing the findings, including graphs or tables generated from the analysis.