02443 Stochastic Simulation - Project 1

Group 23:
Gunn Persdóttir Jacobsen - s175799
Julie Clausen - s180733
Søren Skjernaa - s223316

15. June 2023

Contents

0 Note

Group formation:
We tried to find a group on the discussion forum and by writing our contact information on all blackboards in the exercise rooms, but did not get any response (likely due to the fact that we work in R, as most other groups seem to use Python). To ensure we finished the project on time we started solving it and wrote the report anyways.

Concerning the report:
For all testing purposes we use a significance level of $\alpha = 0.05$. All R-code used to produced the results in this report is viewable in the appendix.

1 Part 1

In this part, we work with a discrete-time model including a random variable denoted by X_t, where t represents time in months. These random variables can take on values from 1 to 5. Additionally, our model assumes that X_t has the Markov property, such that the future values of X_t must be conditionally independent of the past values given the present value.

Given a specific probability matrix P where p_{ij} is the probability of transitioning from stage i to stage j, we are interested in determining the survival distribution and answering questions about the development of cancer stages in women where the stages are:

1. Breast tumor removed after surgery.

2. Local reappearance of cancer.

3. Distant reappearance of cancer.

4. Both local and distance reapparance of cancer.

5. Death.

The observed probability matrix P is given by

$$P = \begin{pmatrix} 0.9915 & 0.005 & 0.0025 & 0 & 0.001 \\ 0 & 0.986 & 0.005 & 0.004 & 0.005 \\ 0 & 0 & 0.992 & 0.003 & 0.005 \\ 0 & 0 & 0 & 0.991 & 0.009 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

1.1 Task 1

We simulate the lifetime distribution of 1000 women starting in state 1 and iterate until all women have reached state 5. We simulate using the event-by-event principle, so for each woman we sample a new stage X_{t+1} given the probabilities of transitioning from stage X_t starting from stage $X_0 = 1$ until death. Thus for each woman we get a Markov Chain $(X_0, X_1, \ldots, X_{n_i})$ where n_i is the month woman i enter stage 5 (death).

By plotting the lifetime distribution as a histogram, we are able to get a more visual look into how many months elapse before the women reach state 5 (death).

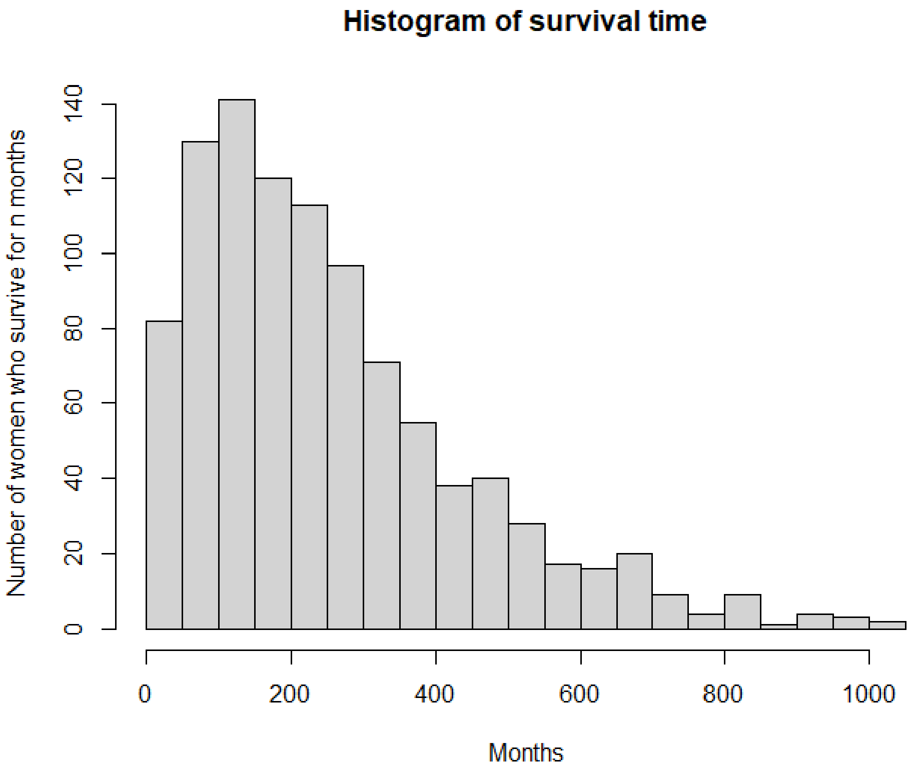

Figure 1: Histogram of survival times

In Figure 1 the histogram peaks at 100-150 in survival time while it flattens out around survival time 600 to 1100. This indicates that there is a significant proportion of women that die within the first 100-150 months (approximately 8-13 years) after surgery. However, the histogram also suggests that there is an additional part of women who survives beyond this period, implying that in some cases we have long-term survival.

The flattening of the histogram around survival time 600-1100 means that a small group of women survives beyond until this time frame.

In state 1 the woman has had their breast tumor removed, when going into state 2, the cancer has recurred locally. From our simulation we observe the fraction of women for whom the cancer eventually reappears locally, at any point before reaching the death state (state 5) as 59.3%.

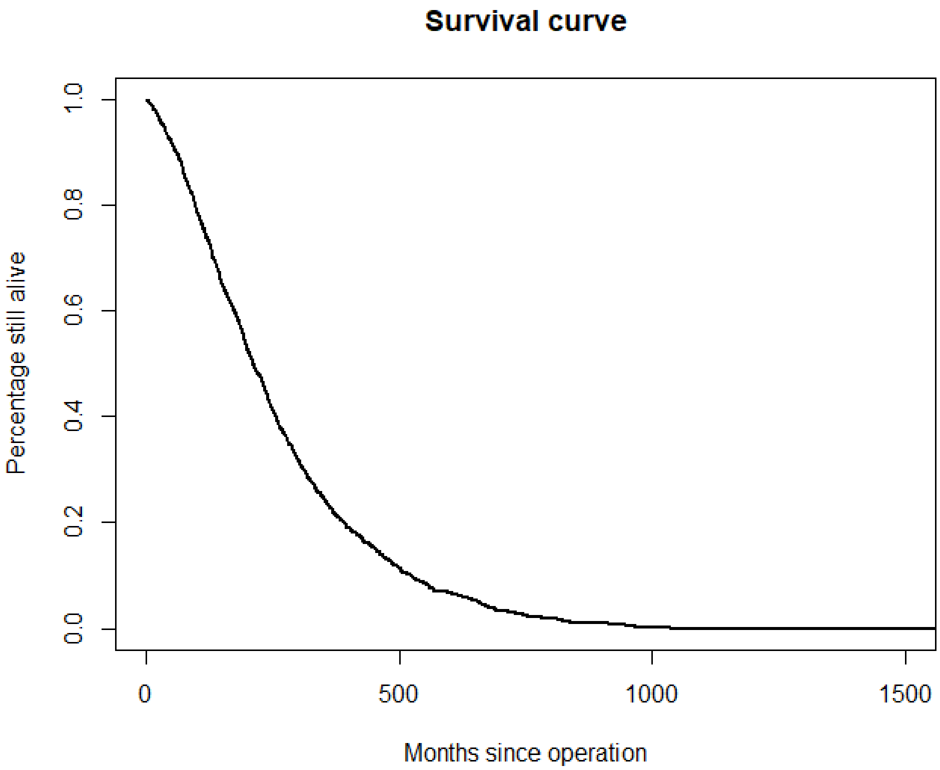

Figure 2: Survival curve estimated from the simulation.

In Figure 2 we see the survival curve representing the months after the operation and the percentage of the women surviving. Around 500 months the percentage of surviving flattens and thereby the percentage for survival is crucial.

1.2 Task 2

Based on the simulation, we find the empirical distribution of X_{120}. This is compared to the theoretical distribution $p_t = p_0(P^t)$ where the initial distribution $p_0 = (1, 0, 0, 0, 0)$.

	State 1	State 2	State 3	State 4	State 5
Empirical	0.334	0.171	0.161	0.072	0.262
Theoretical	0.359	0.159	0.166	0.068	0.248

Table 1: Empirical and theoretical distribution of the stages at time $t = 120$.

In Table 1 we see that the empirical probabilities and theoretical probabilities at $t = 120$ are very close. This implies that the simulation results are consistent with the expected distribution based on the theoretical calculations.

We examine the similarity between the two distributions by further testing and performing a χ^2-test. The χ^2-test assesses the goodness of fit between the observed counts (empirical) and the expected counts (theoretical) in each state. The obtained χ^2 test-statistic returns a p-value of 0.427, indicating that we cannot reject the null hypothesis that the empirical distribution and theoretical distribution are significantly different. Thereby supporting the conclusion that the simulation results are consistent with the expected distribution.

1.3 Task 3

We evaluate the simulated lifetime distribution by comparing it to the theoretical distribution. The theoretical lifetime distribution is known as a discrete phase-type distribution, which can be described by its probability mass function and mean.

The probability mass function of the lifetime follows the formula $P(T = t) = \pi(P_s)^t p_s$, with π denoting the distribution over the states from 1 to 4 at $t = 0$. P_s is a 4×4 submatrix of P, obtained by removing the last row and column, and P_s represents the transition probabilities between states 1 to 4, excluding state 5. Furthermore, we have p_s indicating the probability of dying from states $1, 2, 3, 4$.

To determine if the simulated lifetimes follow this distribution, we can compare the empirical lifetime distribution obtained from the simulation with the theoretical distribution described above.

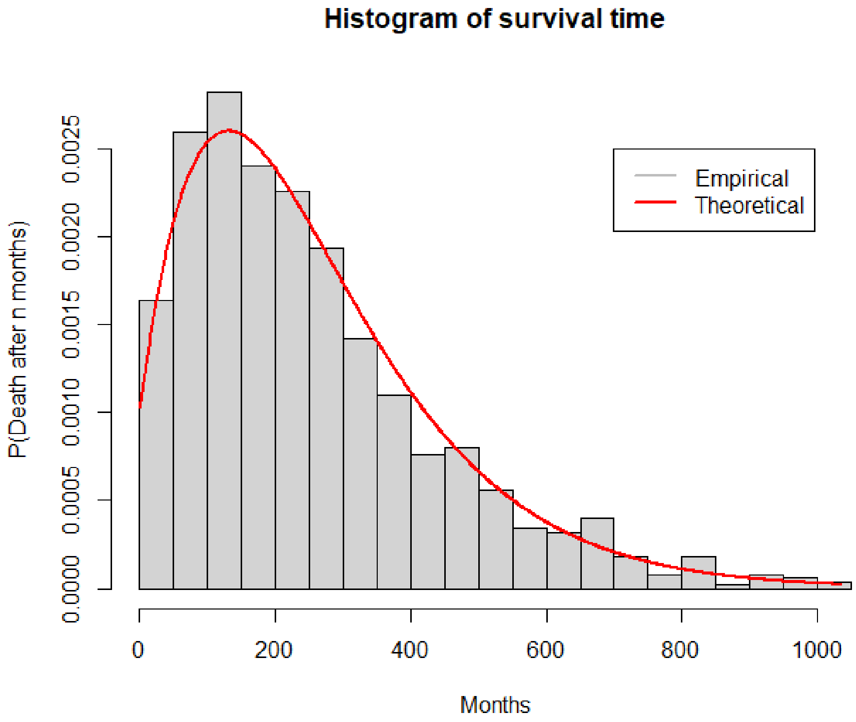

Figure 3: Histogram of the probability of dying after t months, overlaid with the theoretical $P(T = t)$.

In Figure 3 the plot combines the empirical survival time distribution obtained from the simulation with the theoretical probability-mass function (pdf) of the lifetime distribution. We see that the simulated lifetimes are consistent with the expected behavior according to the discrete phase-type distribution.

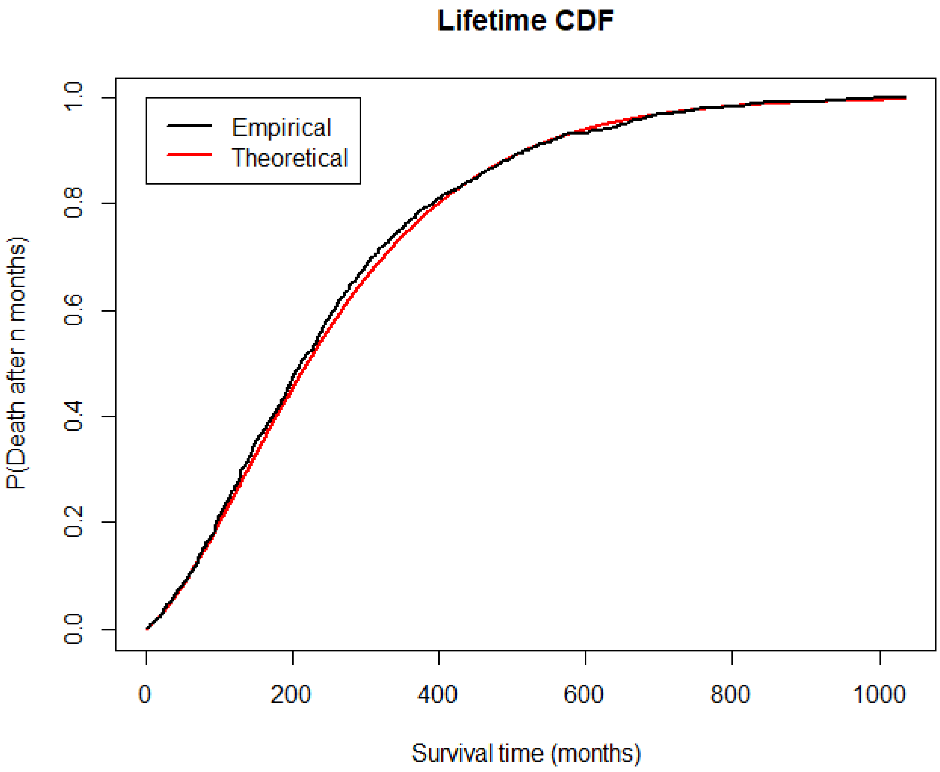

Figure 4: Empirical and theoretical CDF for the survival time.

In Figure 4 the cumulative distribution function (CDF) for the theoretical lifetime distribution and the empirical CDF obtained from the simulation results is plotted against each other. The red line represents the theoretical CDF of the lifetime distribution, while the black line represents the empirical CDF based on the simulated survival times. The figure shows that the simulated survival times align with the expected cumulative distribution according to the theoretical distribution.

Additionally, we run a Kolmogorov-Smirnov test to compare the empirical and theoretical lifetime distributions, the result of the adjusted D_n-value is 0.845. We compare this with the 95% quantile in the corresponding distribution which is 1.358. Since the adjusted D_n-value of 0.845 is lower than the 95% quantile of 1.358, we cannot reject the hypothesis that the simulation follows the theoretical distribution.

1.4 Task 4

We estimate the expected lifetime of a woman who survives the first 12 months following surgery but experiences a recurrence of breast cancer either locally or distantly (or both), within that time frame by applying rejection sampling. We simulate the women and retain the simulations that satisfy the given criteria. By repeating this process until we obtain 1000 acceptable simulations, we can estimate the expected lifetime.

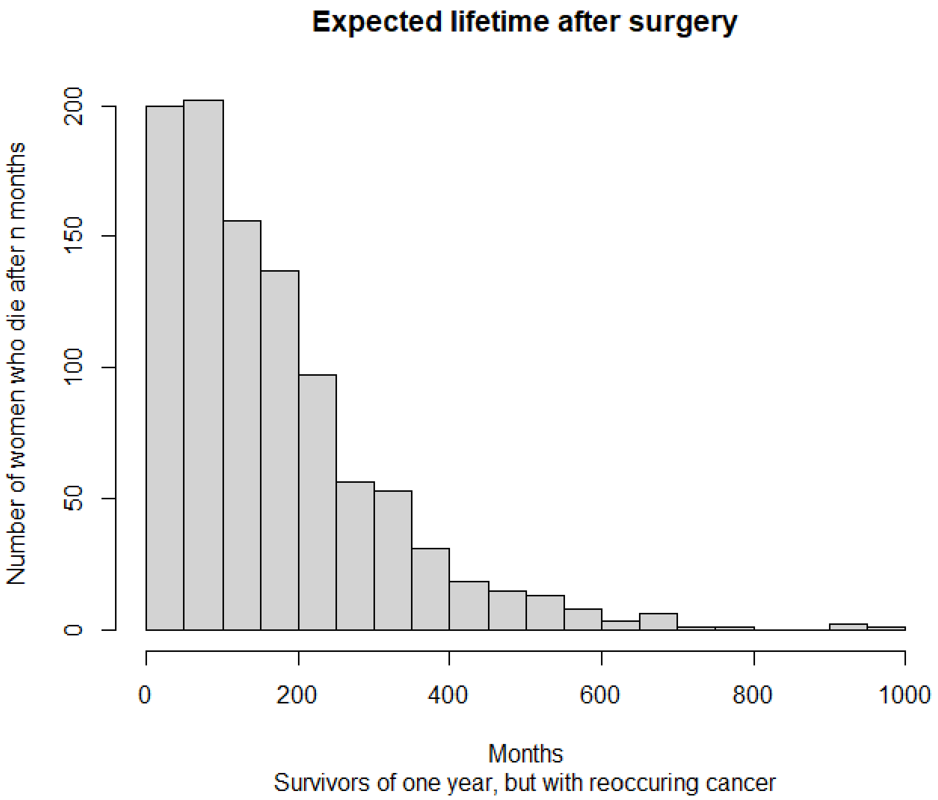

Figure 5: Histogram of survival times

Figure 5 shows the histogram for the expected lifetime distribution of women who reach state 2, 3 or 4 within the first 12 month, but not stage 5 (death). In the histogram, we see that from 600 to 1000 the histogram is very flat indicating that there is a relatively low probability of survival until that time frame for the simulated women who meet the inclusion criteria. This suggests that the recurrence of breast cancer within the first 12 months has a significant impact on the expected lifetime after surgery. We also see that there is a large drop off in the number of women who survive for more than 100 months.

Additionally, we calculate the expected lifetime, along with the corresponding confidence intervals, for the simulated women who meet the inclusion criteria. The estimated mean expected lifetime is 166.367 months, and the estimated variance is 19454.67 months. We note that the variance is quite large. It is pulled up, by the 14 women who manage to survive for more than 600 months (50 years) after surgery. Using a t-distribution with 999 degrees of freedom and a 95% confidence level, the 95% confidence interval for the expected lifetime is approximately (157.712, 175.022) months.

1.5 Task 5

We now estimate the fraction of women who die within 350 months of their surgery. We do this by running 100 simulations, where we simulate 200 women in each run. Based on the 100 simulations we calculate both a crude Monte Carlo estimate of the fraction and an estimate using control variates to reduce the estimator's variance.

The control variates estimation is based on the variables $Z_i = X_i + c(Y_i - \mu_Y)$, where

- X_i is the observed fraction of women who died in the first 350 months in simulation i
- Y_i is the mean survival time of all women in simulation i.
- c is defined as $c = -\frac{\text{Cov}(X_i, Y_i)}{\text{Var}(Y_i)}$, where the covariance and variance are observed sample covariance and sample variance.
- μ_Y is the mean survival time, which is known and given by the formula $\text{E}(T) = \pi(I - P_s)^{-1}\mathbf{1} \approx 262.37$.

Note that $\text{E}(Z_i) = \text{E}(X_i)$ and since the mean survival time and the fraction of deaths within 350 months are correlated $\text{E}(Z_i)$ is an estimator of the fraction of deaths within 350 months of surgery with reduced variance.

Performing the simulations we get the following results

	Estimate	Variance	Lower 95% CI	Upper 95% CI
Crude Estimator	0.735	0.0007	0.733	0.738
Control variates Estimator	0.736	0.0003	0.734	0.737

The 95 % confidence interval was constructed using quantiles from the t-distribution. We observe a reduction in the estimator's variance of 59%.

1.6 Task 6

The discrete time Markov-chain model comes with a set of assumptions for our cancer case. Some of these are:

1. Stage transitions happens can only happen with one month intervals between them. Thus if a woman goes from stage 1 to 2 and then to 4, within the same month we could not model this. Instead our model would need to change from 1 to 4. A solution to this problem could be to use Continuous-Time Markov Chains (CTMC) as described in the next part.

2. The model is restricted to only five stages and the transition probabilities are the same for each woman and stay fixed over time. Thus, other health factors are not considered and neither is the possibility of external factors which may change the probabilities over time. A fix to this could be to include more states, specifying the circumstances of the women in each state.

3. In the model, it is possible to go from state 2 to state 3, meaning the local metastasis disappear and the distant metastasis appears. This scenario seems strange in real life (although it could be possible). Furthermore, patients can never be cured from reappearing cancer, or hop back in stages. This could easily be changed by changing P.

4. The probability of changing state at each month does not depend on time already spent in the current state. Thus there is no difference in your probability of death from state 4, depending on if you have been in stage 4 for one month or 100 months. To change this, we would need something other than a Markov-chain.

2 Part 2

As discussed in the last section of part 1, we addressed certain realistic concerns regarding the assumptions made during the simulation of the discrete event time model. In this next part we will try update the simulation to a CTMC model, and thereby eliminate some of the assumptions that were made in the previous part.
A CTMC is characterized by a transition-rate matrix, where the off-diagonal element q_{ij} represents the rate at which the CTMC moves from state i to state j when it is currently in state i. Furthermore, the sojourn time spent in state i is exponentially distributed with rate $-q_{ii}$.

2.1 Task 7

We are given the following transition-rate matrix:

$$Q = \begin{pmatrix} 0.0085 & 0.00 & 50.002 & 50 & 0.001 \\ 0 & 0.014 & 0.005 & 0.004 & 0.005 \\ 0 & 0 & 0.008 & 0.003 & 0.005 \\ 0 & 0 & 0 & 0.009 & 0.009 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

We use this to simulate the cancer stage of $n = 1000$ women after surgery. To simulate the life of one woman starting from state one until death, we proceed as follows: First, we sample a time from the exponential distribution with rate $-q_{ii}$ (i being the current state) to determine when she transitions to another state. Next, we sample a new state based on their respective probabilities, with the probability of moving from state i to state j being $-q_{ij}/q_{ii}$. This process continues until we sample state 5 (death). The resulting information is stored in a state matrix of size 1000×5, where each row represents a woman, and each column represents the time when she entered the corresponding state. The lifetime distribution after surgery (time when state 5 is entered) is summarized in the histogram below.

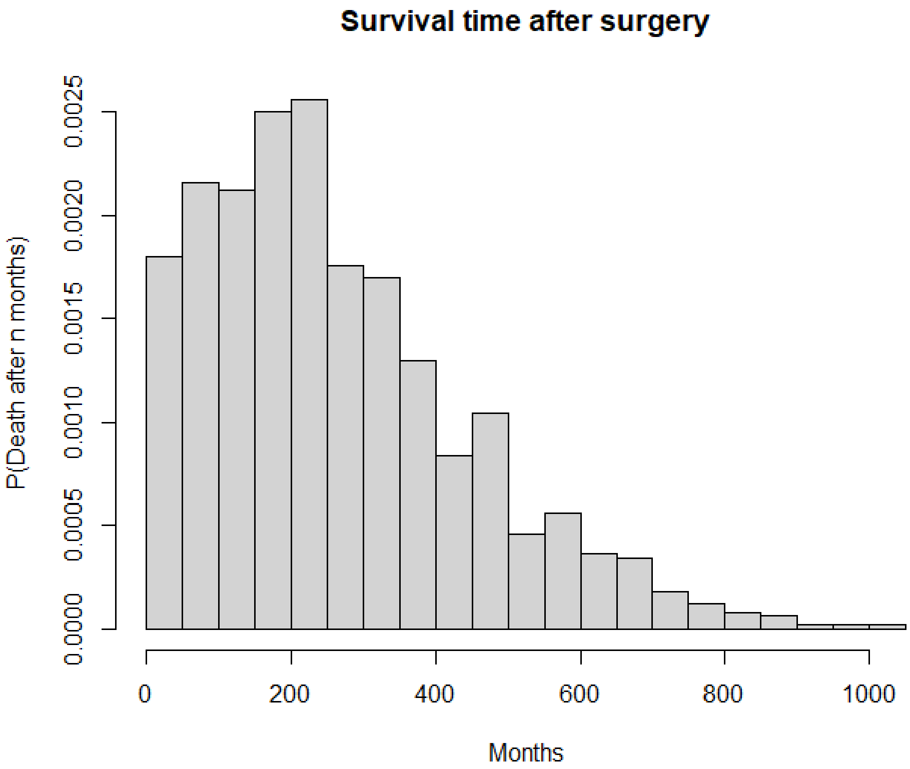

Figure 6: Histogram of survival times.

Mean	Standard deviation	95% CI
263.63	182.93	(252.27, 274.98)

The mean of the survival time is about 263.6 months which corresponds to approximately 22 years. Using the observations from the simulation we estimate that for 61.3 % of the women experience cancer reappearing distantly after 30.5 months (state 3 or 4).

2.2 Task 8

It is known that the theoretical cumulative distribution function (CDF) of a lifetime is given by

$$F_T(t) = 1 - \mathbf{p}_0 \exp\left(\mathbf{Q}_s t\right)\mathbf{1}$$

where $\mathbf{p}_0$ is the initial probability for states 1 to 4, $\mathbf{1}$ is a four-dimensional vector of ones and $\mathbf{Q}_s$ is a the sub-matrix of Q with the last row and column removed. The matrix exponential is calculated as $\exp\left(\mathbf{Q}_s t\right) = \sum_{i=1}^{\infty} \frac{(\mathbf{Q}_s t)^i}{i!}$. Because we know that every woman starts in state 1, we define $\mathbf{p}_0 = (1, 0, 0, 0)$.

Using this, we plot the theoretical CDF together with the empirical CDF from the simulation. We observe a close alignment between the empirical and the theoretical distribution. The

Kolmogorov-Smirnov test, in the case where all parameters are known, yields an adjusted test statistic of 0.933 (with the 95% quantile being 1.358). Thus we do not reject the hypothesis that the simulated data follows the theoretical lifetime distribution.

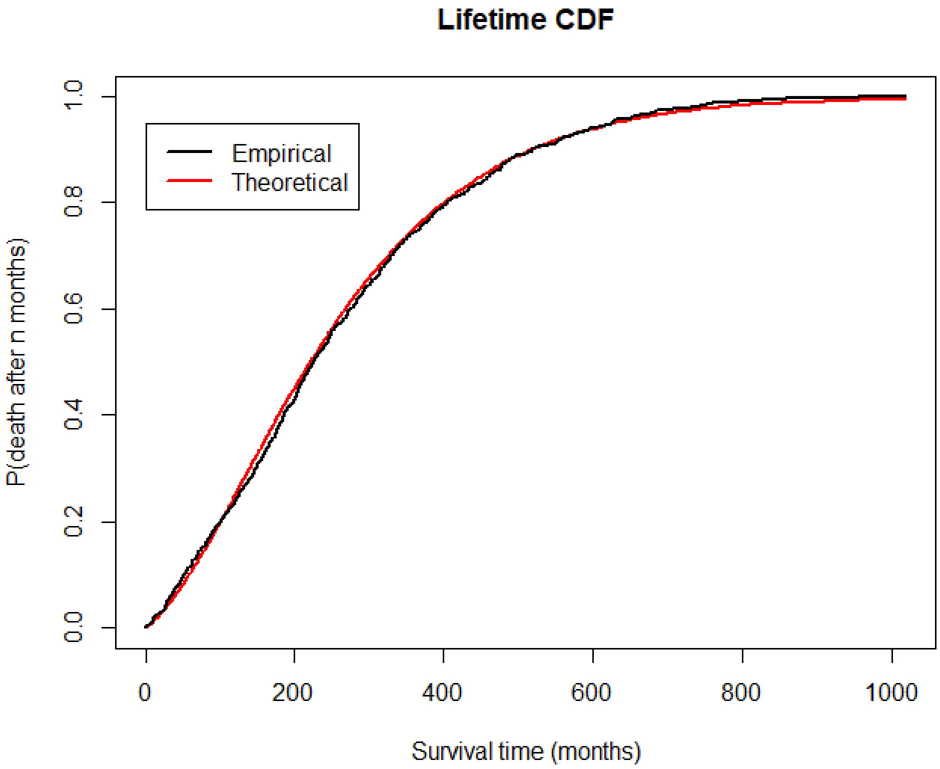

Figure 7: Empirical and theoretical CDF for the survival time.

2.3 Task 9

After applying a specific preventive treatment, we have been given the resulting transmission-rate matrix:

$$Q^* = \begin{pmatrix} -0.00475 & 0.0025 & 0.00125 & 0 & 0.001 \\ 0 & -0.007 & 0 & 0.002 & 0.005 \\ 0 & 0 & -0.008 & 0.003 & 0.005 \\ 0 & 0 & 0 & -0.009 & 0.009 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

To evaluate the effects of this new treatment, we simulate the lives of 1000 women using the updated transition matrix Q^*, following the same approach as in task 7. The duration of each woman's life is recorded in the 5th column of the state matrix, allowing us to determine the number of women who have passed away at a given time. The plot below presents two distinct curves representing the Kaplan-Meier estimates of the survival function given by

$$\widehat{S}(t) = \frac{n - d(t)}{n}$$

where $d(t)$ is the number of women who have died at time t. The black curve presents the people who have received the old treatment and the red curve the people that have received the new treatment.

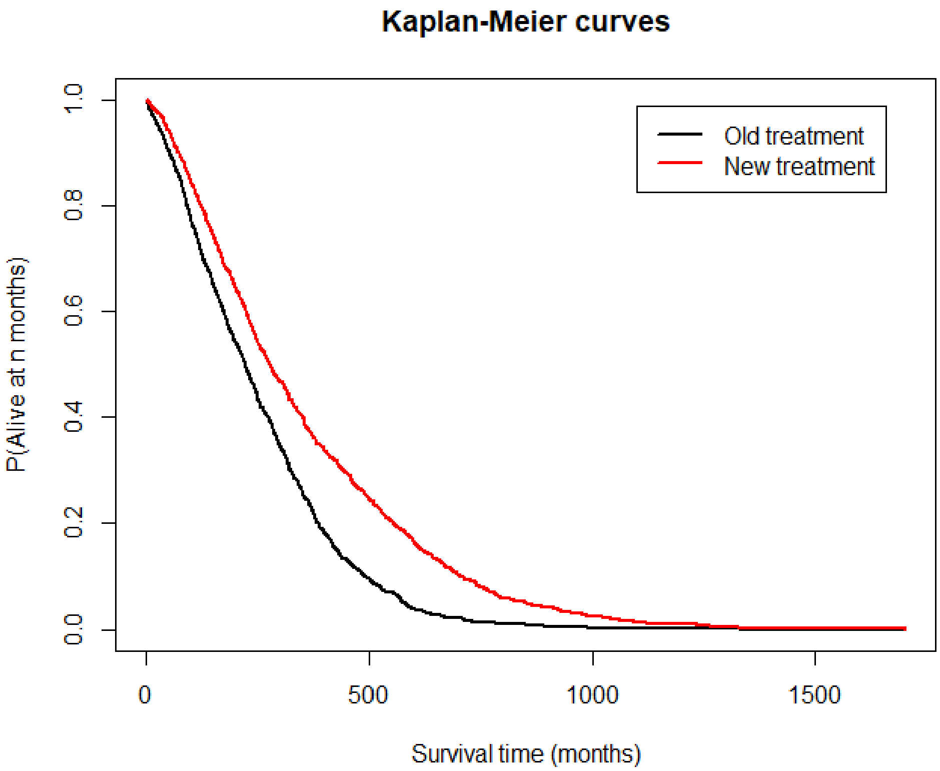

Figure 8: Estimated survival functions $\widehat{S}(t)$ for women being given the new and the old treatment respectively.

Based on the presented plot, it is appears that the treatment has an effect indicated by the slightly higher estimate of survival time.

2.4 Task 10 (optional)

We did not complete Task 10 before having to hand in.

2.5 Task 11

By going from the discrete to the continuous time model, we have overcome the limitation of assuming only one event occurring per month. This means that in the continuous time model, a woman can e.g. transition from state 1 to state 2 and from state 2 to state 3 in a time span of less than one month, which was not feasible in the discrete time model.

On the other hand, the continuous model relies on the assumption that the duration in a state is sampled from an exponential distribution. This distribution is characterized by its memoryless property, implying that the probability of a woman remaining in a state for an additional 12 months, given that she has already been in that state for 36 months, is the same as if she has only been in that state for 1 month. This assumption, most likely, does not accurately reflect real-life scenarios as we would expect a cancer disease to worsen over time.

We could extend the model by letting the sojourn times be Erlang distributed, i.e. given $k \in \mathbf{N}$ the sojourn time for state i follows the Erlang$(k, -q_{ii})$-distribution. The probability of transitioning from state i to state j could remain $-q_{ij}/q_{ii}$. Note however, that if the sojourn times are Erlang distributed, the Markov property is lost as the Erlang distribution is not memoryless. Instead we would obtain a semi-Markov chain, where the probability of transitioning to a new state, depends on the time already spent in the current state.

3 Part 3

Finally, we show how the matrix Q of transitions-rates for the Continuous-Time Markov Chain (CTMC) may be estimated, assuming we have a data set of screenings for the women. In our scenario, we assume the screenings were conducted every 48 months, starting from month 0 and following the patients until death. Thus, our data set consists of time series $X_1, X_2, \dots, X_n$ for the n women, where $X_i = (X_{i,0}, X_{i,1}, \dots, X_{i,n_i})$ and $X_{i,j}$ is the state of woman i at $j \cdot 48$ month.

3.1 Task 12

In order to create the time series for estimating Q, we simulate $n = 1000$ women using the previously described simulation and Q from Task 7. When simulating the women, we record which state they were in at the times $t = 0, 48, 96, \dots, n_i \cdot 48$ until death, to construct $X_i = (X_{i,0}, X_{i,1}, \dots, X_{i,n_i})$.

The time series for the first 20 simulated women is seen below.

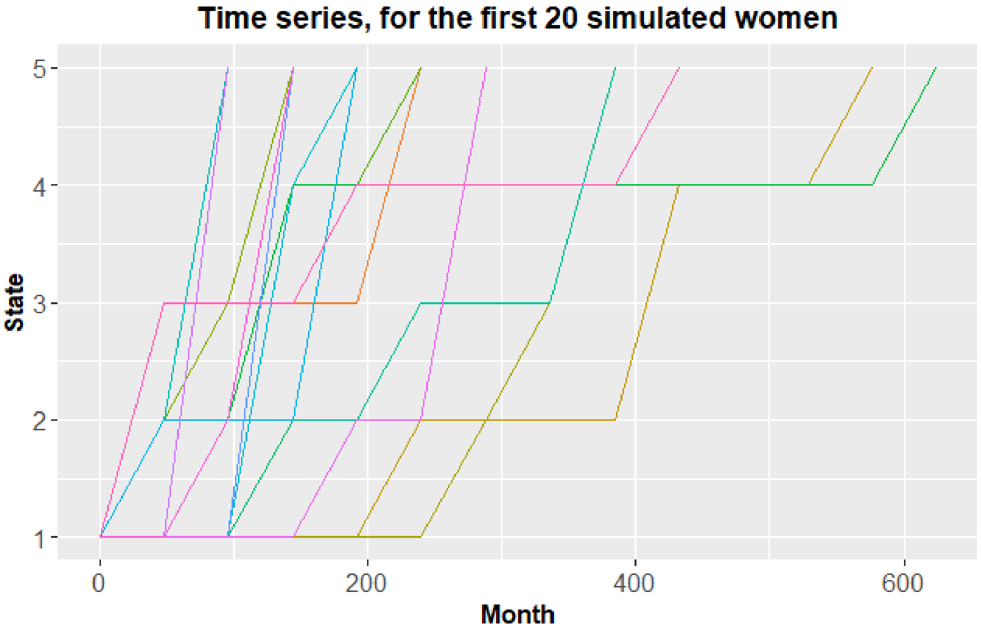

Figure 9: Time series for the first 20 simulated women.

As the point of Part 3 is to estimate Q, from the simulated time series, we will not go into detail with survival curves for this simulation.

3.2 Task 13

Assuming we only have access to the simulated time series from Task 12, we now estimate Q using the Monte Carlo Expectation Maximization Algorithm. The steps of the algorithm are as follows for the k'th iteration:

0. Initialize Q_0 as the initial guess of Q. We have used

$$Q_0 = \begin{pmatrix} -0.01 & 0.0025 & 0.0025 & 0.0025 & 0.0025 \\ 0 & -0.01 & 0.0034 & 0.0033 & 0.0033 \\ 0 & 0 & -0.01 & 0.005 & 0.005 \\ 0 & 0 & 0 & -0.01 & 0.01 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

   Our choice of Q_0 assumes equal mean sojourn time in each state $i = 1, 2, 3, 4$ and equal transitioning probability from state i to all states $i + 1, \ldots, 5$. Thus we do not use any of our previous knowledge about the cancer case.

1. Using Q_k, simulate complete trajectories for each woman respecting the observed time series.

   - For women i, we simulate each of the 48 months periods until death. Thus, simulating a period from month $j \cdot 48$ to month $(j + 1) \cdot 48$, we simulate using Q_k starting in state $X_{i,j}$. The simulation is only accepted if it ends in state $X_{i,j+1}$. If not we simulate the period again.
   - During simulation we record the sojourn time of each state the simulation passes through and add it to an overall record in the variable $S_k = (S_{k,1}, S_{k,2}, S_{k,3}, S_{k,4}, S_{k,5})$ if the simulation is accepted. Furthermore, we record the transitions between states and add it to an overall record of the number of transitions $N_{k,i,j}$ from state i to state j, if the simulation is accepted.
   - Unless the end state is 5 (death) we simulate until 48 months have passed, and cap the sojourn time of the last state so that the sojourn times sum to 48, over the 48 months period.
   - If the end state is 5 (death), we run the simulation for a maximum of 48 months until state 5 occurs.

2. Using the observed total number of transitions $N_{k,i,j}$ from state i to state j for all women in simulation k and the total sojourn time in each state $S_k = (S_{k,1}, S_{k,2}, S_{k,3}, S_{k,4}, S_{k,5})$ for all women in simulation k we calculate Q_{k+1} as:

   - The off-diagonals of Q_{k+1} are

$$q_{ij} = \frac{N_{k,i,j}}{S_{k,i}}, \qquad \text{for } i \neq j \text{ and } i = 1, 2, 3, 4.$$

   - The elements of row 5 are $q_{5j} = 0$ for $j = 1, 2, 3, 4, 5$.
   - The diagonals q_{ii} of Q_{k+1} are the negative row-sum of their respective rows.

3. If $\|Q_k - Q_{k+1}\|_\infty < 10^{-3}$ we say that the algorithm have converged with $\widetilde{Q} = Q_{k+1}$. Else we go to step 1.

Using the described algorithm, we reach convergence in three iterations. Our estimated Q matrix is

$$\widetilde{Q} = \begin{pmatrix} -0.0085 & 0.005 & 0.0024 & 0 & 0.0010 \\ 0 & -0.0136 & 0.0052 & 0.0038 & 0.0046 \\ 0 & 0 & -0.0086 & 0.0032 & 0.0053 \\ 0 & 0 & 0 & -0.009 & 0.009 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

We see that the Monte-Carlo Expectation Maximization algorithm efficiently produced an estimation of Q that is close to the original. Changing the convergence threshold to 10^{-4} the algorithm did not converge within 200 iterations. Thus, with the data available we cannot improve the estimation by much.

4 Appendix

4.1 Part 1 - R code

```r
# Preamble ######################################################################

## File Description #############################################################
#
#   Soren Skjernaa - s223316
#   15/06-2023
#
#   Stochastic Simulation
#   Project 1
#   Part 1
#
#   Notes:
#       ...
#
################################################################################

## Clean up #####################################################################
rm(list = ls())
if(!is.null(dev.list())) dev.off()

## Library ######################################################################
library(matrixcalc)



# _ #############################################################################
# Task 1 ########################################################################

# Parameters
P <- matrix(c(0.9915, 0.005, 0.0025, 0, 0.001,
             0 , 0.986, 0.005, 0.004, 0.005,
             0, 0, 0.992, 0.003, 0.005,
             0, 0, 0, 0.991, 0.009,
             0, 0, 0, 0, 1), nrow = 5, byrow = TRUE)
n <- 1000        # Number of women simulated
m <- dim(P)[1]   # Number of states



# Simulation function
sim1 <- function(n){

    # Storage of result
    survival_time <- numeric(n)
    state_matrix <- matrix(5, nrow = n, ncol = 10^4)

    # Simulation
    for (i in 1:n){

        state <- 1     # All women start in state 1
        time <- 0      # Time starts at 0
        alive <- TRUE  # All women are alive at start

        while (alive){

            # Store current state
            state_matrix[i, time] <- state

            # Change state
            state <- sample(1:m, size = 1, prob = P[state,])

            # Check for death
            if (state == 5){
                alive <- FALSE
            }

            time <- time + 1
        }

        # Store survival time and death state
        survival_time[i] <- time
        state_matrix[i, time] <- state
    }

    # Empirical probabilities at each time step
    prob_matrix <- matrix(0, 10^4, m)
    for (i in 1:10^4){
        for (j in 1:m){
            prob_matrix[i,j] <- sum(state_matrix[,i] == j) / n
        }
    }

    # Return results
    result <- list("survival_time" = survival_time,
                  "state_matrix" = state_matrix,
                  "prob_matrix" = prob_matrix)
    return(result)

}

```

```r

# Performs simulation
sim <- sim1(n)
survival_time <- sim$survival_time
state_matrix <- sim$state_matrix
emp_prob_matrix <- sim$prob_matrix


# Plot of survival time
hist(survival_time)


# Count women who enter state 2
state_2_count <- 0
for (i in 1:n){
    state_2_count <- state_2_count + as.integer(2 %in% state_matrix[i,])
}
state_2_prob <- state_2_count / n
state_2_prob


# Survival curve
plot(1:10^4, 1- emp_prob_matrix[,5], xlim = c(0, 1500),
     type = "l", lwd = 2,
     main = "Survival curve", xlab = "Months since operation",
     ylab = "Percentage still alive")


# Task 2 #######################################################################

# Theoretical probability distribution at time t = 120
t <- 120
p0 <- c(1, 0, 0, 0, 0)
theor_prob_120 <- p0 %*% matrix.power(P, t)

# Comparison of results
rbind("Empirical" = emp_prob_matrix[120,],
      "Theoretical" = theor_prob_120)

# Chi Square test
state_120_count <- emp_prob_matrix[120,] * n
chisq.test(state_120_count, p = theor_prob_120)


# Task 3 #######################################################################

# Define the constants used
```

```r
Pi <- c(1, 0, 0, 0)
Ps <- P[1:4, 1:4]
ps <- P[1:4, 5]

# Probability distribution of lifetime T
lifetime_pdf <- function(t){

    p <- Pi %*% matrix.power(Ps, t) %*% ps
    return(p)
}


# Plot of survival time against lifetime pdf
hist(survival_time, prob = TRUE, breaks = 20)
max_lifetime <- max(survival_time)
x = seq(1, max_lifetime, 1)
y <- numeric(length(x))
for (i in 1:length(x)){
    y[i] <- lifetime_pdf(x[i])
}
lines(x, y, lwd = 2, col = "red")
legend(500, 0.0018, legend = c("Empirical", "Theoretical"),
      col=c("gray", "red"), lty=1, lwd = 2)


# Plot of theoretical lifetime CDF and empirical
CDF_theoretical <- cumsum(y)
CDF_empirical <- emp_prob_matrix[1:max_lifetime, 5]
plot(x, CDF_theoretical, type = "l", lwd = 2, col = "red",
     main = "Lifetime CDF", xlab = "Survival time (months)", ylab = "P(death)")
lines(x, CDF_empirical, lwd = 2)
legend(0, 1, legend = c("Empirical", "Theoretical"),
      col=c("black", "red"), lty=1, lwd = 2)

# Perform Kolmogorov Smirnov test
Dn <- max(abs(CDF_empirical - CDF_theoretical))
adj_DN <- (sqrt(max_lifetime) + 0.12 + 0.11 / sqrt(max_lifetime)) * Dn
adj_DN # 95% quantile is 1.358

# Task 4 ----------------------------------------------------------------------

# Parameters
n <- 1000      # Number of simulated patients fulfilling criteria

# Simulate women who survives 12 month, but with reappearing cancer
sim2 <- function(n){

   # Storage of result
   survival_time <- numeric(n)

   # Counter of women who meet inclusion criteria
   counter <- 1

   # Simulation
   while (counter != n){

       state <- 1                  # All women start in state 1
       time <- 0                   # Time starts at 0
       alive <- TRUE               # All women are alive at start
       state_first_12 <- numeric(12) # States first 12 months
       state_first_12[1] = 1

       while (alive){

           time <- time + 1

           # Change state
           state <- sample(1:m, size = 1, prob = P[state,])

           if (time <= 12){
               state_first_12[time] <- state
           }

           # Check for death
           if (state == 5){
               alive <- FALSE
           }

       }

       inclusion_criteria_1 <- !(5 %in% state_first_12)
       inclusion_criteria_2 <- any(c(2, 3, 4) %in% state_first_12)

       if (inclusion_criteria_1 && inclusion_criteria_2){

           survival_time[counter] <- time
           counter <- counter + 1

       }
   }

   return(survival_time)
}

```

```
# Simulate survival time of 1000 women who fulfill the criteria
sim <- sim2(n)

# Plot Histogram
hist(sim,
     main = "Expected lifetime after surgery",
     sub = "Survivors of one year, but with reoccuring cancer",
     xlab = "Months", ylab = "P(Death after n months)")

# Confidence intervals
mu <- mean(sim)
S2 <- var(sim)
t <- qt(0.975, df = n-1)
mu
S2
c(mu - sqrt(S2 / n) * t, mu + sqrt(S2 / n) * t)


# Task 5 #######################################################################

# Parameters
k <- 350    # Find fraction of women who die within k months
l <- 100    # Number of simulations
n <- 200    # Number of women to simulate in each simulations


# Vectors to store X (parameter to estimate) and Y (control variate)
deaths_within_350 <- numeric(l)
mean_survival_time <- numeric(l)


# Simulation loop
for (i in 1:l){

    result <- sim1(n)
    survival_time <- result$survival_time
    deaths_within_350[i] <- sum(survival_time <= k) / n
    mean_survival_time[i] <- mean(survival_time)
}


# Mean lifetime
theor_lifetime_mean <- Pi %*% solve(diag(4) - Ps) %*% rep(1, 4)


# Calculate fraction of deaths within k months using control variates
c <- - cov(deaths_within_350, mean_survival_time) / var(mean_survival_time)
```

```
Z <- deaths_within_350 + c * (mean_survival_time - theor_lifetime_mean)



# Confidence interval without using control variates
mu <- mean(deaths_within_350)
S2 <- var(deaths_within_350)
t <- qt(0.975, df = l-1)
CL <- mu - sqrt(S2 / l)
CU <- mu + sqrt(S2 / l)
temp1 <- c("Mean" = mu, "Var" = S2, "CL" = CL, "CU" = CU)



# Confidence interval using control variates
mu <- mean(Z)
S2 <- var(Z)
t <- qt(0.975, df = l-1)
CL <- mu - sqrt(S2 / l)
CU <- mu + sqrt(S2 / l)
temp2 <- c("Mean" = mu, "Var" = S2, "CL" = CL, "CU" = CU)


# Present results and reduction in variance
rbind("No CV" = temp1, "With CV" = temp2)
1 - temp2["Var"] / temp1["Var"]
```

4.2 Part 2 - R code

```
# Preamble #####################################################################

## File Description ############################################################
#
#   S0ren Skjernaa - s223316
#   15/06-2023
#
#   Stochastic Simulation
#   Project 1
#   Part 2
#
#   Notes:
#       ...
#
###############################################################################

## Clean up ####################################################################
```

```
rm(list = ls())
if(!is.null(dev.list())) dev.off()

## Library ######################################################################
library(matrixcalc)
library(expm)
library(survival)


# _ #############################################################################
# Task 7 ########################################################################

# Parameters
n <- 1000         # Number of women simulated
Q1 <- matrix(c(-0.0085, 0.005, 0.0025, 0, 0.001,
              0, -0.014, 0.005, 0.004, 0.005,
              0, 0, -0.008, 0.003, 0.005,
              0, 0, 0, -0.009, 0.009,
              0, 0, 0, 0, 0), nrow = 5, byrow = TRUE)



# Simulation function
sim1 <- function(n, Q){

    # Number of states
    m <- dim(Q)[1]

    # Storage of result
    state_matrix <- matrix(NA, nrow = n, ncol = 5)
    state_matrix[,1] <- 0

    # Simulation
    for (i in 1:n){

        state <- 1    # All women start in state 1
        time <- 0     # Time starts at 0
        alive <- TRUE # All women are alive at start

        while (alive){

            # Sample sojourn time
            sojourn_time <- rexp(1, rate = - Q[state, state])

            # Calculate time since surgery
            time <- time + sojourn_time

            # Sample state shift
            p <- - Q[state, 1:m] / Q[state, state]
            p[p == -1] <- 0
            state <- sample(1:m, size = 1, prob = p)

            # Record state shift
            state_matrix[i, state] <- time

            # Check for death
            if (state == 5){
                alive <- FALSE
            }
        }
    }

    return(state_matrix)
}


# Performs simulation
sim <- sim1(n, Q1)
emp_lifetime <- sim[, 5]


# Histogram of survival time
hist(emp_lifetime, prob = TRUE,
     main = "Survival time after surgery",
     xlab = "Months", ylab = "P(Death after n months)")

# Confidence intervals
mu <- mean(emp_lifetime)
S2 <- var(emp_lifetime)
t <- qt(0.975, df = n-1)
mu
sqrt(S2)
c(mu - sqrt(S2 / n) * t, mu + sqrt(S2 / n) * t)


# Proportion of women with cancer reappearing distantly after 30.5 months
sum(sim[,3] > 30.5 | sim[,4] > 30.5, na.rm = TRUE) / n


# Task 8 #######################################################################

# Define the sub matrix of Q1
Q1s <- Q1[1:4, 1:4]

```

```r
# Define the initial probability p0
p0 <- c(1, 0, 0, 0, 0)

# Define the theoretical CDF of lifetime
FT <- function(t){

    result <- 1 - p0[1:4] %*% expm(Q1s * t) %*% rep(1, 4)
    return(result)
}


# Calculate the empirical CDF
x <- sort(emp_lifetime)
CDF_empirical <- cumsum(rep(1/n, n))


# Plot of theoretical lifetime CDF and empirical
CDF_theoretical <- numeric(length(x))
for (i in 1:length(x)){
    CDF_theoretical[i] <- FT(x[i])
}
plot(x, CDF_theoretical, type = "l", lwd = 2, col = "red",
     main = "Lifetime CDF", xlab = "Survival time (months)",
     ylab = "P(death after n months)")
lines(x, CDF_empirical, lwd = 2)
legend(0, 0.99, legend = c("Empirical", "Theoretical"),
       col=c("black", "red"), lty=1, lwd = 2)

# Perform Kolmogorov Smirnov test
Dn <- max(abs(CDF_empirical - CDF_theoretical))
adj_DN <- (sqrt(n) + 0.12 + 0.11 / sqrt(n)) * Dn
adj_DN # 95% quantile is 1.358


# Task 9 #######################################################################


# Parameters
Q2 <- matrix(c(-sum(0.0025, 0.00125, 0, 0.001), 0.0025, 0.00125, 0, 0.001,
              0, - sum(0, 0, 0.002, 0.005), 0, 0.002, 0.005,
              0, 0, -sum(0, 0, 0.003, 0.005), 0.003, 0.005,
              0, 0, 0, -sum(0, 0, 0, 0.009), 0.009,
              0, 0, 0, 0, 0),
            nrow = 5, byrow = TRUE)

# Kaplan-Meier estimate
S <- function(t, lifetime){

    # Number of women
    N <- length(lifetime)

    # Number of dead women at time t
    d <- sum(lifetime <= t)

    # Survival fraction
    survival_fraction <- (N - d) / N
    return(survival_fraction)
}

# Simulate lifetimes with and without treatment
temp <- sim1(n, Q1)
emp_lifetime1 <- temp[,5]
temp <- sim1(n, Q2)
emp_lifetime2 <- temp[,5]

# Estimate the two Kaplan-Meier curves and plot them
max_lifetime <- max(emp_lifetime1, emp_lifetime2)
x <- seq(0, max_lifetime + 1, 1)

y1 <- numeric(length(x))
for (i in 1:length(x)){
    y1[i] <- S(x[i], emp_lifetime1)
}

y2 <- numeric(length(x))
for (i in 1:length(x)){
    y2[i] <- S(x[i], emp_lifetime2)
}

plot(x, y1, type = "l", lwd = 2, col = "black",
     main = "Kaplan-Meier curves", xlab = "Survival time (months)",
     ylab = "P(Alive at n months)")
lines(x, y2, lwd = 2, col = "red")
legend(800, 0.99, legend = c("Old treatment", "New treatment"),
       col=c("black", "red"), lty=1, lwd = 2)


# Task 10 ######################################################################

# Pool the two populations and sort based on lifetime
test_data <- data.frame("Group" = c(rep(1, n), rep(2, n)),
                      "lifetime" = c(emp_lifetime1, emp_lifetime2))
test_data <- test_data[order(test_data[,2], decreasing = FALSE),]

```

```r
# Add N_j
N_j <- (2 * n - 1):0
test_data["N_j"] <- N_j

# Add N_ij
N_1j <- numeric(2 * n)
N_1j[1] <- n
N_2j <- numeric(2 * n)
N_2j[1] <- n

if (test_data[1, 1] == 1){

    N_1j[1] <- n - 1
    N_2j[1] <- n

} else{

    N_1j[1] <- n
    N_2j[1] <- n - 1
}

for (j in 2:(2 * n)){

    # Group event happens in at time j
    group <- test_data[j, 1]

    # Downsize groups
    if (group == 1){

        N_1j[j] <- N_1j[j - 1] - 1
        N_2j[j] <- N_2j[j - 1]

    } else{

        N_1j[j] <- N_1j[j - 1]
        N_2j[j] <- N_2j[j - 1] - 1
    }
}

test_data["N_1j"] <- N_1j
test_data["N_2j"] <- N_2j

# Add O_ij and O_j
O_1j <- rep(n, 2 * n) - N_1j
O_2j <- rep(n, 2 * n) - N_2j
O_j <- O_1j + O_2j

```

```r
test_data["O_1j"] <- O_1j
test_data["O_2j"] <- O_2j
test_data["O_j"] <- O_j

# Calculate E_ij
E_1j <- O_j * (N_1j / N_j)
E_2j <- O_j * (N_2j / N_j)

test_data["E_1j"] <- E_1j
test_data["E_2j"] <- E_2j


# Calculate V_ij
V_1j <- E_1j * ((N_j - O_j) / N_j) * ((N_j - N_1j) / (N_j - 1))
V_2j <- E_2j * ((N_j - O_j) / N_j) * ((N_j - N_2j) / (N_j - 1))

test_data["V_1j"] <- V_1j
test_data["V_2j"] <- V_2j

# Calculate test-statistic
Z_1 <- sum(O_1j[1:1000] - E_1j[1:1000]) / sqrt(sum(V_1j[1:1000]))
Z_2 <- sum(O_2j[1:1000] - E_2j[1:1000]) / sqrt(sum(V_2j[1:1000]))

# Calculate p-values
p_1 <- pnorm(Z_1, lower.tail = FALSE)
p_2 <- pnorm(Z_2, lower.tail = FALSE)

# Summarize results
c("Z1" = Z_1, "p1" = p_1, "Z2" = Z_2, "p2" = p_2)
```

4.3 Part 3 - R code

```r
# Preamble ######################################################################

## File Description ############################################################
#
#   Soren Skjernaa - s223316
#   15/06-2023
#
#   Stochastic Simulation
#   Project 1
#   Part 3
#
#   Notes:
#       ...
#
```

```r
################################################################################

## Clean up ####################################################################
rm(list = ls())
if(!is.null(dev.list())) dev.off()

## Library #####################################################################
library(matrixcalc)
library(expm)
library(ggplot2)

# _ ############################################################################
# Task 12 ######################################################################

# Parameters
n <- 1000        # Number of women simulated
Q <- matrix(c(-0.0085, 0.005, 0.0025, 0, 0.001,
             0, -0.014, 0.005, 0.004, 0.005,
             0, 0, -0.008, 0.003, 0.005,
             0, 0, 0, -0.009, 0.009,
             0, 0, 0, 0, 0), nrow = 5, byrow = TRUE)
m <- dim(Q)[1]


# Simulation function
sim1 <- function(n, Q){

    # Number of states
    m <- dim(Q)[1]

    # Storage of result
    state_matrix <- matrix(NA, nrow = n, ncol = 5)
    state_matrix[,1] <- 0

    state_ts <- matrix(5, n, ceiling(150 * 12 / 48) + 1)
    state_ts[,1] <- 1

    # Simulation
    for (i in 1:n){

        state <- 1    # All women start in state 1
        time <- 0     # Time starts at 0
        alive <- TRUE # All women are alive at start

        while (alive){

            # Sample sojourn time
            sojourn_time <- rexp(1, rate = - Q[state, state])

            # Calculate time since surgery
            time <- time + sojourn_time

            # Record state time series
            start_record <- min(which(state_ts[i,] == 5))
            end_record <- ceiling(time / 48)

            if (start_record <= end_record){
                for (j in start_record:end_record){
                    state_ts[i, j] <- state
                }
            }

            # Sample state shift
            p <- - Q[state, 1:m] / Q[state, state]
            p[p == -1] <- 0
            state <- sample(1:m, size = 1, prob = p)

            # Record state shift
            state_matrix[i, state] <- time

            # Check for death
            if (state == 5){
                alive <- FALSE
            }
        }

    }

    results <- list("state_matrix" = state_matrix,
                    "state_ts" = state_ts)
    return(results)
}


# Performs simulation
sim <- sim1(n, Q1)
ts <- sim$state_ts

# Storing the time series in a data frame for plotting
df_ts <- data.frame("Woman" = numeric(0),
                    "Month" = numeric(0),
                    "State" = numeric(0))
for (i in 1:dim(ts)[1]){

    n_obs <- min(which(ts[i,] == 5))

    for (j in 1:n_obs){

        temp <- c("Woman" = i, "Month" = (j - 1) * 48, "State" = ts[i, j])
        df_ts <- rbind(df_ts, temp)
    }
}
colnames(df_ts) <- c("Woman", "Month", "State")

# Plot the time series
ggplot(df_ts[1:155,], aes(x = Month, y = State, group = as.factor(Woman),
                  col = as.factor(Woman))) +
    geom_line() +
    labs(title = "Plot of simulated time series",
         x = "Month",
         col = "State") +
    theme(plot.title = element_text(hjust=0.5, size=14, face="bold"),
          axis.text = element_text(size=12),
          axis.title = element_text(size=12, face="bold"),
          legend.position = "none")


# Task 13 ######################################################################


## Simulate 48 months ##########################################################
sim2_step48 <- function(initial, end, Q, sojourns, transitions){

    converged = FALSE
    while (!converged){

        # Initialize parameters for simulation
        time_left <- 48
        alive <- TRUE
        state <- initial
        temp_sojourns <- sojourns
        temp_trans <- transitions

        # Main loop
        while (time_left > 0 && alive && state <= end){

            # Sample sojourn time and add it to time spent in current state
            sojourn_time <- rexp(1, rate = - Q[state, state])
            temp_sojourns[state] <- temp_sojourns[state] +
                                      min(sojourn_time, time_left)

            # Calculate time left until next medical visit
            time_left <- time_left - sojourn_time
                time_left

            # If there is still time left sample a new state to shift to
            if (time_left > 0){

                # Sample state shift and record it
                p <- - Q[state, 1:m] / Q[state, state]
                p[p == -1] <- 0
                new_state <- sample(1:m, size = 1, prob = p)
                temp_trans[state, new_state] <- temp_trans[state, new_state] + 1
                state <- new_state
            }

            # Check for death
            if (state == 5){
                alive <- FALSE
            }
        }

        # Check if simulation ends in right step
        if (end != 5){

            if (time_left <= 0 && state == end){

                converged <- TRUE
                sojourns <- temp_sojourns
                transitions <- temp_trans
            }
        } else{

            if (state == end){

                converged <- TRUE
                sojourns <- temp_sojourns
                transitions <- temp_trans
            }
        }
    }


    result <- list("sojourns" = sojourns, "transitions" = transitions)
    return(result)
}


```

```
## Simulate for all women ######################################################
sim2 <- function(n, Q, ts){

    # Number of states
    m <- dim(Q)[1]

    # Storage of result
    sojourns <- numeric(m)
    transitions <- matrix(0, m, m)


    # Loop over the women
    for (i in 1:n){

        # Loop over each 48 month period
        obs <- min(which(ts[i,] == 5))
        for (j in 1:(obs - 1)){

            # Start and end state of the 48 month period
            initial <- ts[i, j]
            end <- ts[i, j + 1]

            # Simulation of a 48 month period
            temp <- sim2_step48(initial, end, Q, sojourns, transitions)

            # Results
            sojourns <- temp$sojourns
            transitions <- temp$transitions
        }
    }

    result <- list("sojourns" = sojourns, "transitions" = transitions)
    return(result)
}



## Estimate Q ##################################################################

# Initialize original guess at matrix Q
Q_old <- matrix(c(-0.01, 0.0025, 0.0025, 0.0025, 0.0025,
                  0, -0.01, 0.0034, 0.0033, 0.0033,
                  0, 0, -0.01, 0.005, 0.005,
                  0, 0, 0, -0.01, 0.01,
                  0, 0, 0, 0, 0),
                nrow = 5, byrow = TRUE)

```

```
# Check if initialized matrix is allowable
sum(Q_old[1,])
sum(Q_old[2,])
sum(Q_old[3,])
sum(Q_old[4,])


# Perform the Monte Carlo Expectation Maximization
threshold <- 10^(-3)
converged <- FALSE
counter <- 1

while (!converged){

    # Print iteration number
    print(counter)

    # Step 1: Simulate possible trajectories for al ts
    sim <- sim2(n, Q, ts)
    sojourns <- sim$sojourns
    transitions <- sim$transitions

    # Step 2 and 3: Find N_ij, S_i and the new matrix Q_k+1

    # Initialize new Q matrix
    Q_new <- matrix(0, 5, 5)

    # Calculate new Q off-diagonals
    for (i in 1:(m - 1)){
        for (j in 1:m){
            if (i != j){

                N_ij <- transitions[i, j]
                S_i <- sojourns[i]
                Q_new[i, j] <- N_ij / S_i
            }
        }
    }

    # Calculate Q diagonal
    for (i in 1:(m - 1)){

        Q_new[i, i] <- - sum(Q_new[i,])
    }

    # Compare old and new Q matrix
    converged <- norm(Q_old - Q_new, type = "i") < threshold

    # Update Q
    Q_old <- Q_new

    # Increment counter
    counter <- counter + 1
}

# Results
Q
round(Q_new, 4)
```