Statistical packages - report 3

Urszula Grochocińska, Marcin Mazurkiewicz

February 3, 2019

Contents

1 Introduction

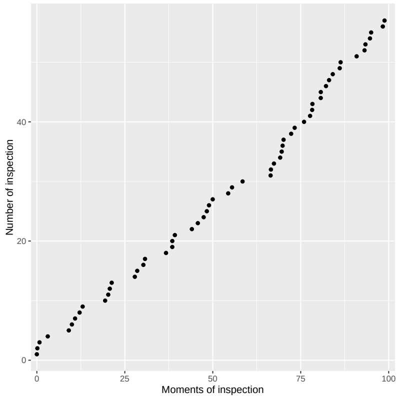
This report concerns periodic inspections and interval censored data. The first task is to prepare a generator for periodically inspected lightbulb. Then there would be presented a short analysis basing on this generator. Finally there are two approaches shown: first uses the naive estimator (without censored data) and the second one with censord data.

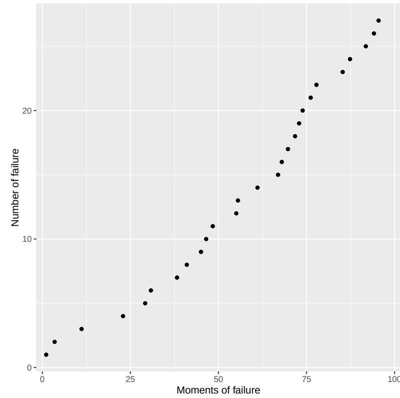
1.1 Task 1 – creating generator

We assume, that the life times of lightbulbs is a vector of iid random variables from exponential distribution with failure rate λ. The times between inspections is a vector of iid random variables from exponential distribution with rate v. Change of a failed lightbulb can occur only at the moment of inspection. The generator should take 3 parameters: two are mentioned above and additional T_0 which is the time horizon. The output of the generator is:

- interval censored lifetimes of lightbulbs,
- moments of inspection,
- moments of light failures.

```
>   v <- .7
>   lambda <- 0.5
>   T_0 <- 100
>   inspection_times <- c(0, sort(runif(n = rpois(1, lambda = T_0 * v), min = 0, max = T_0)))
>   lightbulb <- 1
>   new_bulb_moments <- c(0)
>   lightbulbTime <- rexp(1, rate = lambda)
>   lightbulbLifetimes <- c()
>   for (i in 1:(length(inspection_times) - 1)){
+     if ( inspection_times[i + 1] - new_bulb_moments[lightbulb] > lightbulbTime){
+       new_bulb_moments <- append(new_bulb_moments, inspection_times[i + 1])
+       lightbulb <- lightbulb + 1
+       lightbulbLifetimes <- append(lightbulbLifetimes, lightbulbTime)
```






```
+       lightbulbTime <- rexp(1, rate = lambda)
+     }
+   }
>   df_inspection_times <- data.frame(number = 1:length(inspection_times), moments = inspection_times)
```

1.2 Task 2 – generator analysis

This section presents the answers for the question about generator – in our case the parameters of the generator are $v = 0.7 \quad \lambda = 0.5 \quad T_0 = 100$ and we made 100 Monte Carlo iterations (when it was needed):

1. What is the percentage of time when there is no light?
   Answer: The share of time without light is at level 49.5%.
2. What is the average number of lightbulb replacements?
   Answer: The average number of replacements is 29.75.
3. What is the average time without light?
   Answer: The average time without light is 1.47.

```
>   momentsOfFailure <- lightbulbLifetimes + new_bulb_moments[1:length(new_bulb_moments) - 1]
>   percentage_of_time_withoutlight <- sum(new_bulb_moments[2:length(new_bulb_moments)] - momentsOfFailu
>   N <- 100
>   v <- .7
>   lambda <- 0.5
>   T_0 <- 100
>   avg_no_of_replacements <- 0
>   avg_time_without_light <- 0
>   for (j in 1:N){
+       inspection_times <- c(0, sort(runif(n = rpois(1, lambda = T_0 * v), min = 0, max = T_0)))
+
+       lightbulb <- 1
+       new_bulb_moments <- c(0)
+       lightbulbTime <- rexp(1, rate = lambda)
+       lightbulbLifetimes <- c()
+       for (i in 1:(length(inspection_times) - 1)){
+           if ( inspection_times[i + 1] - new_bulb_moments[lightbulb] > lightbulbTime){
+               new_bulb_moments <- append(new_bulb_moments, inspection_times[i + 1])
+               lightbulb <- lightbulb + 1
```

```
+               lightbulbLifetimes <- append(lightbulbLifetimes, lightbulbTime)
+               lightbulbTime <- rexp(1, rate = lambda)
+           }
+       }
+       momentsOfFailure <- lightbulbLifetimes + new_bulb_moments[1:length(new_bulb_moments) - 1]
+       avg_time_without_light <- avg_time_without_light
+                                 + mean(new_bulb_moments[2:length(new_bulb_moments)]
+                                 - momentsOfFailure)
+       avg_no_of_replacements <- avg_no_of_replacements + lightbulb
+   }
>   avg_no_of_replacements <- avg_no_of_replacements / N
>   avg_time_without_light <- avg_time_without_light / N
```

1.3 Task 3 – naive estimator

This approach assumes that failures occured during inspections. We estimate the failure rate using an average of right sides of intervals

how the mean of such estimator depends on true failure rate and inspection rate, how the variance of such estimator depends on true failure rate and inspection rate, how the bias of such estimator depends on true failure rate and inspection rate, how the mean square error of such estimator depends on true failure rate and inspection rate.

```
>   v <- .7
>   lambda <- 0.5
>   T_0 <- 100
>   inspection_times <- c(0, sort(runif(n = rpois(1, lambda = T_0 * v), min = 0, max = T_0)))
>   lightbulb <- 1
>   new_bulb_moments <- c(0)
>   lightbulbTime <- rexp(1, rate = lambda)
>   lightbulbLifetimes <- c()
>   for (i in 1:(length(inspection_times) - 1)){
+       if ( inspection_times[i + 1] - new_bulb_moments[lightbulb] > lightbulbTime){
+           new_bulb_moments <- append(new_bulb_moments, inspection_times[i + 1])
+           lightbulb <- lightbulb + 1
+           lightbulbLifetimes <- append(lightbulbLifetimes, lightbulbTime)
+           lightbulbTime <- rexp(1, rate = lambda)
+       }
+   }
>   naive_lightbulb_lifetime <- new_bulb_moments[2:length(new_bulb_moments)]
>                               - new_bulb_moments[1:length(new_bulb_moments) - 1]

 [1]   0.000000  -3.518197 -18.665587 -20.078466 -21.056650 -26.795721
 [7] -27.401696 -29.390893 -31.089993 -31.305725 -32.983759 -35.946149
[13] -38.911737 -43.948623 -48.319488 -52.374975 -54.513284 -55.884514
[19] -58.415268 -60.090423 -62.959315 -68.784215 -77.512856 -80.204929
[25] -81.523693 -83.573478 -86.246395 -87.362737 -89.644510 -90.467140

>   mean_naive <- mean(naive_lightbulb_lifetime)
>   variance_naive <- var(naive_lightbulb_lifetime)
```