Imperial College London – Zambart

Workshop on *Analysing and modelling epidemic data*

Practical 2 Part 1: Estimating parameters from data.

Ruth McCabe and Dr Pablo N Perez-Guzman

Background

When a new infectious disease emerges, one of the biggest concerns faced by policymakers is whether hospitals will have the capacity to treat all patients requiring care. We can use mathematical models to estimate this, but we need to be able to parameterise them using relevant data.

We consider a situation in which there is a new disease and patients are being admitted to hospital. You are tasked with estimating the number of hospital beds that will be needed over time to care for infected patients. In order to do this, we need to understand the amount of time that patients are ill for.

We will be doing our analysis in R studio. Code is provided in boxes for each question.

Objectives

- Estimate the mean time to recovery and death of patients.
- Understand the impact on estimates if using truncated data.
- Correct estimates derived from truncated data

Example 1: Estimating the mean time to recovery

A doctor from the local hospital has recorded the number of days that it has taken the first 100 patients to recover from the disease:

1, 12, 6, 7, 10, 9, 21, 9, 8, 7, 3, 9, 2, 8, 25, 12, 30, 3, 3, 6,

9, 8, 4, 13, 7, 6, 4, 13, 37, 6, 4, 78, 6, 12, 10, 5, 21, 7, 5, 15,

7, 4, 23, 13, 7, 19, 8, 2, 5, 4, 1, 22, 3, 22, 3, 59, 3, 11, 20, 8,

4, 5, 16, 2, 23, 4, 2, 17, 3, 3, 16, 5, 2, 10, 4, 9, 2, 5, 9, 1,

2, 7, 12, 8, 8, 15, 8, 8, 5, 4, 7, 4, 4, 10, 16, 12, 4, 11, 11, 10

On average, how long does it take patients to recover?

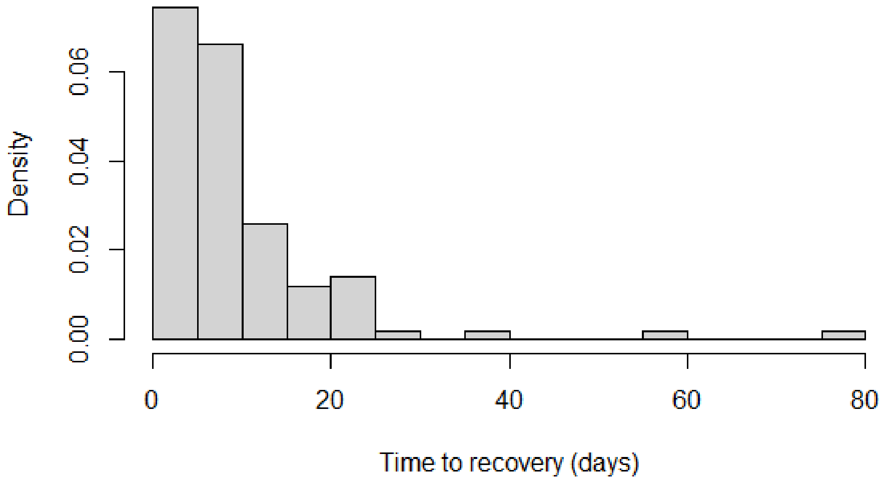
1. Read the data into R using the following code:

```
recovery_data <- c(1, 12, 6, 7, 10, 9, 21, 9, 8, 7, 3, 9, 2, 8, 25, 12, 30, 3, 3, 6, 9,

                 8, 4, 13, 7, 6, 4, 13, 37, 6, 4, 78, 6, 12, 10, 5, 21, 7, 5, 15, 7, 4,

                 23, 13, 7, 19, 8, 2, 5, 4, 1, 22, 3, 22, 3, 59, 3, 11, 20, 8, 4, 5, 16,

                 2, 23, 4, 2, 17, 3, 3, 16, 5, 2, 10, 4, 9, 2, 5, 9, 1, 2, 7, 12, 8, 8,

                 15, 8, 8, 5, 4, 7, 4, 4, 10, 16, 12, 4, 11, 11, 10)
```

2. Use a histogram to look at the distribution of the data. What continuous probability distribution best describes them? (Eg. Normal, Exponential, Gamma, …)

```
hist(recovery_data, freq = FALSE,

   breaks = 20, bty = "n", xlab = "Time to recovery",main="")
```

Solution: The data are not normally distributed. Because of the small sample size, it is difficult to definitively assign a distribution to these data. However, an exponential distribution is appropriate.



3. What is the mean recovery time?

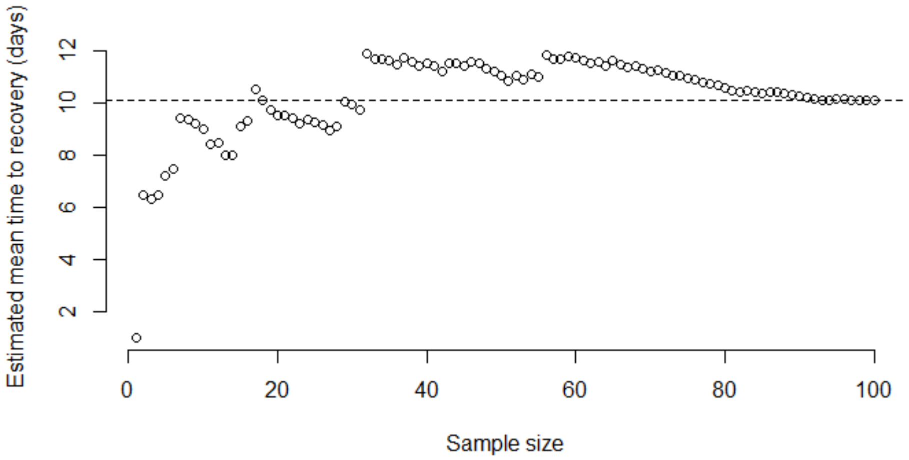
```
mean(recovery_data)
```

Solution: 10.13 days.

4. Consider how the mean recovery time changes as the sample size increases. Do you notice any convergence?

```
n_sample <- 1:n

estimator_over_n <- cumsum(recovery_data) / n_sample

plot(estimator_over_n,

    xlab = "Sample size",

    ylab = "Estimated mean time to recovery",

    bty = "n")

abline(h = mean(recovery_data), lty = 2)
```

Solution: As we increase the sample size to include all of the recovery times above, we see that the estimated mean converges to the overall mean when we have approximately 90 observations. As the sample size increases, the variability in the estimate decreases and thus we can have more confidence our estimate.



Example 2: Estimating the mean times to recovery or death

In Example 1, all of the initial patients recovered. Unfortunately, this is extremely unlikely to be the case in reality when faced with an emerging infectious disease. There are competing risks of whether an individual will recover or die.

We are now going to flash forward to the end of the first wave of this outbreak. Colleagues in a neighbouring state have analysed all of the patients admitted to their hospitals in order to estimate the mean time to recovery or to death (the outcome). We don't get to see their data, but we can see that they modelled them using an exponential distribution.

Let us consider an example for the time to recovery, noting that the procedure is identical for the time to death.

Let $x_1, x_2, x_3, \dots.., x_n$ denote the n observed times for each patient to recover.

Recalling that if a random variable X has an exponential distribution, then $f(X;\lambda) = \lambda \exp(-\lambda x)$is the probability density function.

The likelihood of the observed data is:

$$L = \prod_{i=1}^{n} f(x_i;\lambda) = \prod_{i=1}^{n} \lambda \exp(-\lambda x_i) = \lambda^n \exp\left(-\lambda \sum_{i=1}^{n} x_i \right)$$

The log-likelihood is:

$$l = n \log(\lambda) - \lambda \sum_{i=1}^{n} x_i$$

To find the Maximum Likelihood Estimator (MLE), we differentiate the log-likelihood, set this equal to zero and solve for λ:

$$\frac{\partial l}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} x_i$$

$$\frac{\partial l}{\partial \lambda} = 0 \rightarrow \widehat{\lambda} = \frac{n}{\sum_{i=1}^{n} x_i}$$

By following this procedure, the researchers estimated that:

$$\widehat{\lambda_r} = \frac{1}{14}$$

$$\widehat{\lambda_d} = \frac{1}{7}$$

This means that, on average, patients who recover do so in an average of 14 days, and patients who die do so in average of 7 days.

We can use this information to simulate plausible times to recovery and death and then consider the differences in these two distributions.

1. Sample 1000 observations from each of these distributions.

```
data <- data.frame(
  recovery = round(rexp(1000, 1 / 14)),
  death = round(rexp(1000, 1 / 7)))
```

2. Using the formulae above, what are the MLEs of your two samples (time to recovery and time to death)?

```
length(data$recovery)/sum(data$recovery)
length(data$death)/sum(data$death)
```

Solution: MLE recovery: 0.070; MLE death: 0.150.

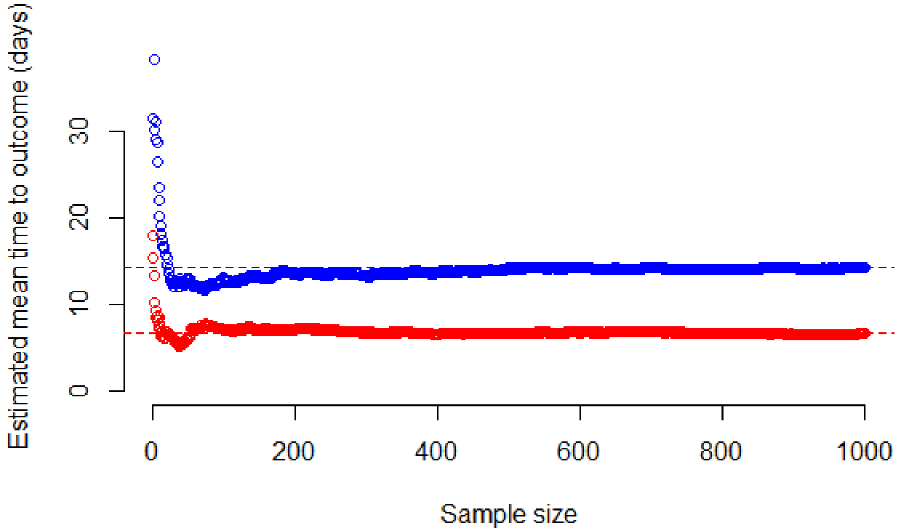
3. What does this translate to in terms of the mean number of days until the outcome? Verify your calculation by using the mean() function.

```
1/(length(data$recovery)/sum(data$recovery))
1/(length(data$death)/sum(data$death))
mean(data$recovery)
mean(data$death)
```

Solution: Based on the MLEs, we estimate an average of 14.26 days to recovery and 6.69 days to death. This is extremely close to the observed values of 14 and 7 days. We get the same values when using the mean() function on our directly rather than having to calculate the MLE explicilty as above.

4. Consider how the mean time to recovery and mean time to death change as the sample size increases. Do you notice any convergence?

Solution: As in Example 1, we observe convergence to the mean outcome time as the sample size increases.



Example 3: Analysing truncated data

During an outbreak, data involving the time to an event are often subject to survival bias. For example, in the first two weeks of an outbreak, you only know what has happened in those two weeks and not what will happen to patients in the future. It is possible that a patient alive on day 14 could die on day 15.

INSERT DIAGRAM TO ILLUSTRATE THIS

Now imagine that your colleagues from Example 2 want to estimate the time to each outcome at the second week of the outbreak. We only know the outcomes of patients up to this time, and so we will remove everyone else from our analysis in this question. We want to investigate how this would change our estimates of the time to outcome.

1. Truncate the data generated in Example 2 to observations less than 14 days.

```
truncated_recov <- data$recovery[which(data$recovery <= 14)]

truncated_death <- data$death[which(data$death <= 14)]
```

2. How many observations are there, what is the MLE and what is the mean of:
   a. Time to recovery?

```
length(truncated_recov)

length(truncated_recov)/sum(truncated_recov)

mean(truncated_recov)
```

Solution: Only 633 patients have a recovery time less than 14 days meaning that we have lost around 1/3 of our cohort. The MLE is 0.168. The mean recovery time is 6 days, which is substantially less than the 14 days that we know is the true time to recovery.

b. Time to death?

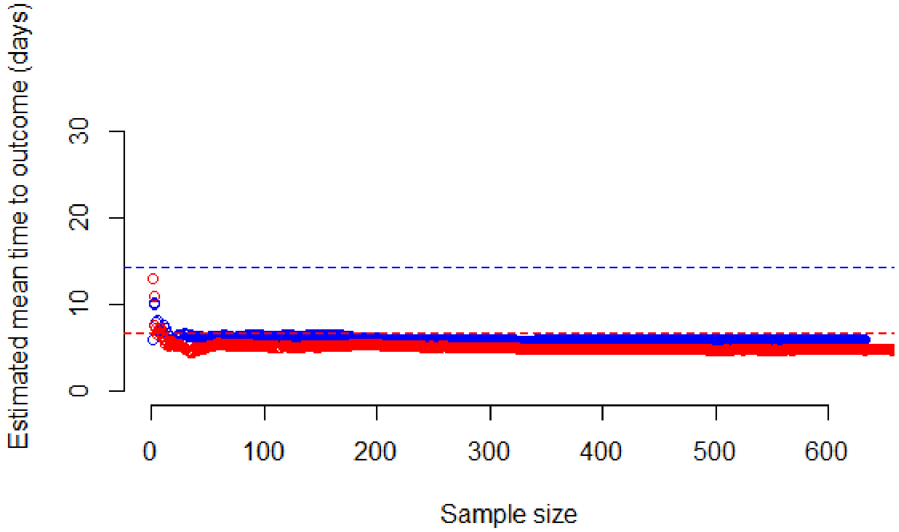
```
length(truncated_death)

length(truncated_death)/sum(truncated_death)

mean(truncated_death)
```

Solution: 885 patients have a time to death of less than 14 days. This is less than within our cohort of patients who recover, as we know that the time to death is shorter than the time to recovery. The MLE is 0.212. The mean time to death within the truncated sample is 4.7 days, compared to the true time to death of 7 days.

3. Consider how the mean time to recovery and mean time to death change as the sample size increases. Do you notice any convergence?

Solution: The samples converge to the sample mean, but as we established above this is substantially different from what we know the true outcome times to be. If we based our estimates on these data alone, we would estimate a much shorter outcome time than is the case. This would impact on the results of our mathematical model that we are looking to parameterise with this information.

Truncation is a common problem in epidemiological data and it is important to be aware of this when analysing event data.



Example 4: Adjusting for truncated data

We have seen that analysing data early on in the outbreak will be subject to bias, in that only events which happen up until that time are included in the analysis. As demonstrated above, this could substantially lower our estimates of the outcome time and thus give us a false picture of the dynamics of the disease.

We can adjust our estimation procedure to account for the fact that our data are truncated.

If a random variable X_T is distributed according to a truncated exponential distribution, then $f(X_T;\lambda) = \frac{\lambda\exp(-\lambda x_T)}{1-\exp(\lambda\beta)}$

where β is the truncation time.

We can follow the steps as above to calculate the MLE, but this involves more complicated mathematics. We shall instead deploy a built-in function in R, uniroot(), to estimate the MLE under this different likelihood function.

1. Run the function titled truncation_adjusted_MLE() using the code below.

```
truncation_adjusted_MLE <- function(data, trunc_cut_off = 14){

  dL <- function(lambda, n = length(data), sum_obs = sum(data)){

  n / lambda - sum_obs - ((n*trunc_cut_off*exp(-lambda*trunc_cut_off))/(1 - exp(-
lambda*trunc_cut_off)))

}

  f_zero <- function(lambda){

    dL(lambda, n = length(data),sum_obs = sum(data))

  }

   ML_sol <- uniroot(f_zero, interval = c(1e-6, 1e6))

    return(list("MLE" = ML_sol$root,

              "average_outcome" = 1/ML_sol$root))

}
```

2. Use the function to work out the MLE and average outcome time for recovery and death, and compare this to what we observed in Example 2.
    a. Time to recovery

Solution: The MLE is 0.066 which gives a mean outcome time of 15 days.

In Example 2, the MLE was 0.070 with a mean outcome time of 14 days. Therefore, we have mostly recaptured the information lost from truncating the data to 14 days, but have slightly overestimated what we know the true value to be observed at the end of the first wave.

    b. Time to death

Solution: The MLE is 0.129 which gives a mean outcome time of 7.8 days.

In Example 2, the MLE was 0.150 with a mean outcome time of 7 days. As above, we have slightly overestimated but are much closer to the true value than we were under the truncated data without adjustment.

Extension: Biased estimates under different truncation times

This is an optional extra for this practical for those who have time.

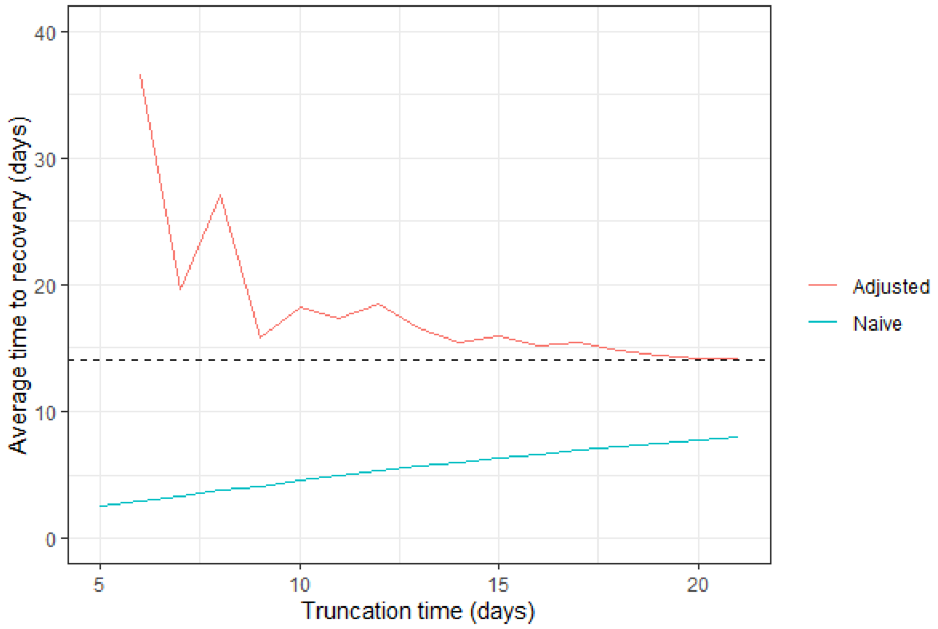
Using what we have learned so far in this practical, can you analyse the effect of different truncation times on the estimates of the mean outcome times?

Solution: Use the code to estimate the naïve and adjusted outcome times for different truncation times.

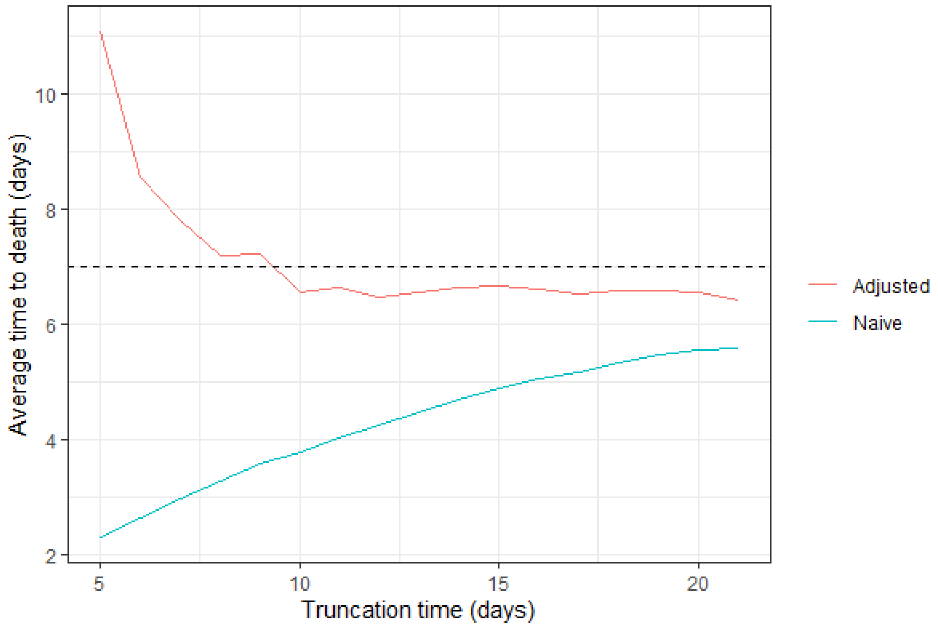
```
truncation_adjusted_MLE_extension <- function(df = data, trunc_cut_off){

   # truncate original data

  data_recov_trunc <- df$recovery[which(df$recovery <= trunc_cut_off)]

  data_death_trunc <- df$death[which(df$death <= trunc_cut_off)]

  ## naive MLE and outcome time

  ##recovery

  MLE_naive_recov <- length(data_recov_trunc)/sum(data_recov_trunc)

  average_outcome_naive_recov <- 1/MLE_naive_recov

  #death

  MLE_naive_death <- length(data_death_trunc)/sum(data_death_trunc)

  average_outcome_naive_death <- 1/MLE_naive_death

  ### adjusted for truncation

  ## recovery

  dL_recov <- function(lambda, n = length(data_recov_trunc), sum_obs =
sum(data_recov_trunc)){

    n / lambda - sum_obs - ((n*trunc_cut_off*exp(-lambda*trunc_cut_off))/(1 - exp(-
lambda*trunc_cut_off)))

  }

  f_zero_recov <- function(lambda){

    dL_recov(lambda, n = length(data_recov_trunc),sum_obs = sum(data_recov_trunc))

  }

  ML_sol_recov <- uniroot(f_zero_recov, interval = c(1e-6, 1e6))

  ## death

  dL_death <- function(lambda, n = length(data_death_trunc), sum_obs =
sum(data_death_trunc)){

    n / lambda - sum_obs - ((n*trunc_cut_off*exp(-lambda*trunc_cut_off))/(1 - exp(-
lambda*trunc_cut_off)))

  }

   f_zero_death <- function(lambda){

    dL_death(lambda, n = length(data_death_trunc),sum_obs = sum(data_death_trunc))

  }

  ML_sol_death <- uniroot(f_zero_death, interval = c(1e-6, 1e6))

  return(c("trunc_cut_off" = trunc_cut_off,

    "MLE_naive_recov" = MLE_naive_recov
```

```
trunc_times <- seq(5,21,1)

sens_analysis <- c()

for(i in trunc_times){

  output <- truncation_adjusted_MLE_extension(df = data, trunc_cut_off = i)

  sens_analysis <- rbind(sens_analysis,output)

}

sens_analysis <- data.frame(sens_analysis)
```

As we increase the truncation time from 5 days to 21 days, the naïve estimate of average time to recovery slowly increases but doesn't quite converge to the true value. However, the adjusted estimate (derived from the adjusted likelihood) does for higher truncation values.



A similar trend is observed for the naïve estimate of time to death, but in this instance the adjusted estimate hasn't quite converged to the observed value by 21 days (although it is close).


Average time to death (days)
10
8
6
4
2
5
10
15
20
Truncation time (days)
Adjusted
Naive