ST227: R in Life Insurance

Kaplan-Meier Estimation

Dr. Viet Dang

11/03/2022

Kaplan - Meier estimation

- Let us re-create the following example from the lecture slide.

```
survData <- data.frame(
  time = c(10,13,18,19,23,30,36,38,54,56,59,75,93,97,104,107,107,107),
  observed = c(T,F,T,F,T,T,F,F,T,T,T,T,F,T,F,F)
)
head(survData)
```

```
##   time observed
## 1   10     TRUE
## 2   13    FALSE
## 3   18    FALSE
## 4   19     TRUE
## 5   23    FALSE
## 6   30     TRUE
```

- If you had data stored in an external excel spreadsheet, consider using `readxl::read_excel` to import it.

```
survData <- read_excel(file.choose())
```

- Above: `head` displays the first few rows of the data frame - this avoids cluttering the display.

Kaplan - Meier estimation

- Assumptions: all uncensored death times are **distinct**.
- Steps:
  1. Calculate the number of individuals at risk at each time.
  2. Filter out i.e. remove the right censored individuals.
  3. Calculate step-wise survival probability and Kaplan Meier estimate.
- Step 1: the number of individuals at risk is all remaining observed units (inclusive). This means the remaining number of rows.

```
survData$atRisk <- nrow(survData):1
head(survData)
```

```
##   time observed atRisk
## 1   10     TRUE     18
## 2   13    FALSE     17
## 3   18    FALSE     16
## 4   19     TRUE     15
## 5   23    FALSE     14
## 6   30     TRUE     13
```

Subsetting of a vector

- For step 2: we will need **subsetting** techniques.
- Subsetting means selecting a portion (i.e. a subset) of the data. It is critical to data analysis.
- Each language has slightly different subsetting syntax. R being built for data analysis has very mature subsetting mechanics.
- Example:

```
x <- c(1,4,3,7,8,2,4)
x[1]
```

```
## [1] 1
```

```
x[c(1,2,4)]
```

```
## [1] 1 4 7
```

- In the second line of code, we selected the first component. In the third line, the first, second and fourth component.

Subsetting a vector

- In the previous example, we directly specified the coordinates of the desired subset. This is called *numerical subsetting*.
- Another important technique is *logical* and *Boolean subsetting*.
- We need a logical vector of the same length as the parent vector. It will select all elements corresponding to a `TRUE`.


```
x <- c(1,5,3,7,8,2,4) #same x as above
selection <- c(T,F,T,F,T,F,T) #select every other elements
x[selection]
```

```
## [1] 1 3 8 4
```

- You can chain operations together for very succinct and self-explanatory codes, e.g:


```
x[x>3] #need x such that x > 3
```

```
## [1] 5 7 8 4
```

Subsetting of a data frame

- A data frame is a two dimensional structure. You might want to subset by either rows or columns.
- We use R's built-in data set `mtcars`.

```
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

- Below are a few examples.

```
mtcars[5,2] #fifth row, second column
mtcars[c(1,2,3),c(1,2)] #rows 1 to 3, columns 1 to 2
mtcars[c(1,2), ] #rows 1,2 and all columns
mtcars[ ,c(1,2)] #all rows and columns 1 to 2
```

Kaplan - Meier estimation

- Back to the main problem. We will need to:
  1. negate the observed vector. This achieves an indicator of fully observed records.
  2. use this Boolean vector to subset all the rows corresponding to fully observed records.


```
survData2 <- survData[survData$observed, ]
head(survData2)
```

```
##    time observed atRisk
## 1    10     TRUE     18
## 4    19     TRUE     15
## 6    30     TRUE     13
## 7    36     TRUE     12
## 11   49     TRUE      9
## 12   75     TRUE      7
```

Kaplan - Meier estimation

- We can now fill in the details:

```
survData2$death <- 1 #or rep(1,time=nrow(survData2))
survData2$survProb <- (survData2$atRisk - survData2$death)/survData2$atRisk
survData2$KP <- cumprod(survData2$survProb)
#a more succint syntax:
# with(survData2,{ (atRisk-Death)/atRisk })
survData2
```

```
##    time observed atRisk death  survProb        KP
## 1    10     TRUE     18     1 0.9444444 0.9444444
## 4    19     TRUE     15     1 0.9333333 0.8814815
## 6    30     TRUE     13     1 0.9230769 0.8136752
## 7    36     TRUE     12     1 0.9166667 0.7458689
## 11   39     TRUE      8     1 0.8750000 0.6526353
## 12   75     TRUE      7     1 0.8571429 0.5594017
## 13   93     TRUE      6     1 0.8333333 0.4661681
## 14   97     TRUE      5     1 0.8000000 0.3729345
## 16  107     TRUE      3     1 0.6666667 0.2486230
```