Homework 1

P8108 - Survival Analysis

William Donovan

```r
library(tidyverse)
library(survival)
library(survminer)
library(ggsurvfit)
```

Question 1

```r
# Load in Q1 data
q1_df = read_csv("data/Q1data_extracted.csv")
```

a. The MLE $\hat{\lambda}$ for an Exponential distribution is given by:

$$\hat{\lambda} = \frac{d}{\sum_i t_i} = \frac{\text{The number of events}}{\text{Person-time: total number of time units observed on all individuals}}$$

Using R to calculate:

```r
# Calculate number of events and person-time for relapse
d_relapse = sum(pull(q1_df, Relapse))
sum_time_relapse = sum(pull(q1_df, Relapse_Time))

# Calculate relapse MLE
mle_relapse = d_relapse/sum_time_relapse

# Calculate number of events and person-time for death
d_death = sum(pull(q1_df, Death))
sum_time_death = sum(pull(q1_df, Death_Time))

# Calculate death MLE
mle_death = d_death/sum_time_death
```

`mle_relapse` $\hat{\lambda}_{relapse} = 0.032$

`mle_death` $\hat{\lambda}_{death} = 0.013$

The maximum likelihood estimator $\hat{\lambda}$ is an estimator for the hazard rate parameter, λ, which is constant in an exponential distribution. The estimated hazard rate of relapse $\hat{\lambda}_{relapse}$ is 0.032 events per month of person-time. The estimated hazard rate of death $\hat{\lambda}_{death}$ is 0.013 events per month of person-time.

b. We can use the MLE to calculate the quantities below.

   i. **Mean**
   The expectation, or mean, of the exponential distribution is $\frac{1}{\lambda}$.

$$\mu_{relapse} = 1/0.032 = 31.25$$
$$\mu_{death} = 1/0.013 = 74.333$$

ii. **Median**
The median of an exponential distribution is given by $\tau = \frac{-log(0.5)}{\lambda}$.

$$\tau_{relapse} = \frac{-log(0.5)}{0.032} = 21.488$$

$$\tau_{death} = \frac{-log(0.5)}{0.013} = 51.524$$

iii. **1 & 2 Year Relapse-Free & Survival Probabilities**
These are calculated using the survival functions $S_R(t)$ and $S_D(t)$. Under the exponential distribution $S(t) = e^{-\lambda t}$.

$$S_R(12) = e^{-0.032(12)} = 0.679$$
$$S_R(24) = e^{-0.032(24)} = 0.461$$
$$S_D(12) = e^{-0.013(12)} = 0.851$$
$$S_D(24) = e^{-0.013(24)} = 0.724$$

iv. **1 & 2 Year Relapse and Death Probabilities**
This is easily calculated from the survival function since $F(t) = 1 - S(t)$.

$$F_R(12) = 1 - S_R(12) = 0.321$$
$$F_R(24) = 1 - S_R(24) = 0.539$$
$$F_D(12) = 1 - S_D(12) = 0.149$$
$$F_D(24) = 1 - S_D(24) = 0.276$$

v. **Probability of Staying Relapse-Free 2 Years Given 1 Year Relapse-Free**
This is a conditional probability denoted as $S_R(24|12)$ and is easily calculated using $S_R(24|12) = S_R(24)/S_R(12)$ since it is certain $S_R(12|24) = 1$. This simplification is shown below.

$$S_R(24|12) = \frac{S_R(24 \cap 12)}{S_R(12)} = \frac{S_R(12|24)S_R(24)}{S_R(12)} = \frac{S_R(24)}{S_R(12)} = \frac{0.461}{0.679} = 0.679$$

As expected, $S_R(24|12) = S_R(12)$ since the hazard rate λ of an exponential distribution is constant.

vi. **Median (Using Non-Parametric Methods)**
If an exponential distribution is not assumed, the median time-to-event can be calculated using a Kaplan-Meier estimate. However, in this case, only the median time-to-relapse can be calculated. The median time-to-event is given by the smallest t where $\hat{S}(t) \leq 0.5$. For deaths, the KM survival estimator $\hat{S}(t)$ never reaches 0.5, since 7 of 10 observations are censored, and can therefore not be estimated. For relapse, $\hat{S}(t)$ drops to 0.5 at 27 months, so the median time-to-relapse is calculated to be 27 months. This can be confirmed with R:

```r
km_q1 = survfit(
  Surv(Relapse_Time, Relapse) ~ 1,
  data = q1_df)

surv_median(km_q1)
```

```
##   strata median lower upper
## 1    All     27    12    NA
```

Question 2

```r
# Load in Q2 data
q2_df = read_csv("data/Q2data_extracted.csv")
```

a. Kaplan-Meier Survival Estimate

t_j	d_j	c_j	r_j	$\lambda_j=(d_j/r_j)$	$\hat{S}(t_j)=\prod_j(1-\lambda_j)$
2	1	0	17	$\frac{1}{17}$	$1.000(1-\frac{1}{17})=0.941$
3	1	0	16	$\frac{1}{16}$	$0.941(1-\frac{1}{16})=0.882$
4	1	0	15	$\frac{1}{15}$	$0.882(1-\frac{1}{15})=0.824$
12	1	0	14	$\frac{1}{14}$	$0.824(1-\frac{1}{14})=0.765$
22	1	0	13	$\frac{1}{13}$	$0.765(1-\frac{1}{13})=0.706$
48	1	0	12	$\frac{1}{12}$	$0.706(1-\frac{1}{12})=0.647$
51	0	1	11	$\frac{0}{11}$	$0.647(1-\frac{0}{11})=0.647$
56	0	1	10	$\frac{0}{10}$	$0.647(1-\frac{0}{10})=0.647$
80	2	0	9	$\frac{2}{9}$	$0.647(1-\frac{2}{9})=0.503$
90	1	0	7	$\frac{1}{7}$	$0.503(1-\frac{1}{7})=0.431$
94	0	1	6	$\frac{0}{6}$	$0.431(1-\frac{0}{6})=0.431$
160	1	0	5	$\frac{1}{5}$	$0.431(1-\frac{1}{5})=0.345$
161	1	0	4	$\frac{1}{4}$	$0.345(1-\frac{1}{4})=0.259$
180	1	1	3	$\frac{1}{3}$	$0.259(1-\frac{1}{3})=0.173$
238	1	0	1	1	$0.173(1-1)=0$

b.
```r
# Log-log CI
km_loglog = survfit2(
  Surv(Value, Binary) ~ 1,
  data = q2_df,
  conf.type = "log-log")

summary(km_loglog)
```

```
## Call: survfit(formula = Surv(Value, Binary) ~ 1, data = q2_df, conf.type = "log-log")
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     2     17       1    0.941  0.0571       0.6502        0.991
##     3     16       1    0.882  0.0781       0.6060        0.969
##     4     15       1    0.824  0.0925       0.5471        0.939
##    12     14       1    0.765  0.1029       0.4883        0.904
##    22     13       1    0.706  0.1105       0.4315        0.866
##    48     12       1    0.647  0.1159       0.3771        0.823
##    80      9       2    0.503  0.1272       0.2436        0.716
##    90      7       1    0.431  0.1277       0.1870        0.656
##   160      5       1    0.345  0.1280       0.1216        0.584
##   161      4       1    0.259  0.1217       0.0691        0.505
##   180      3       1    0.173  0.1074       0.0296        0.416
```

```
##   238       1       1    0.000     NaN           NA           NA
```

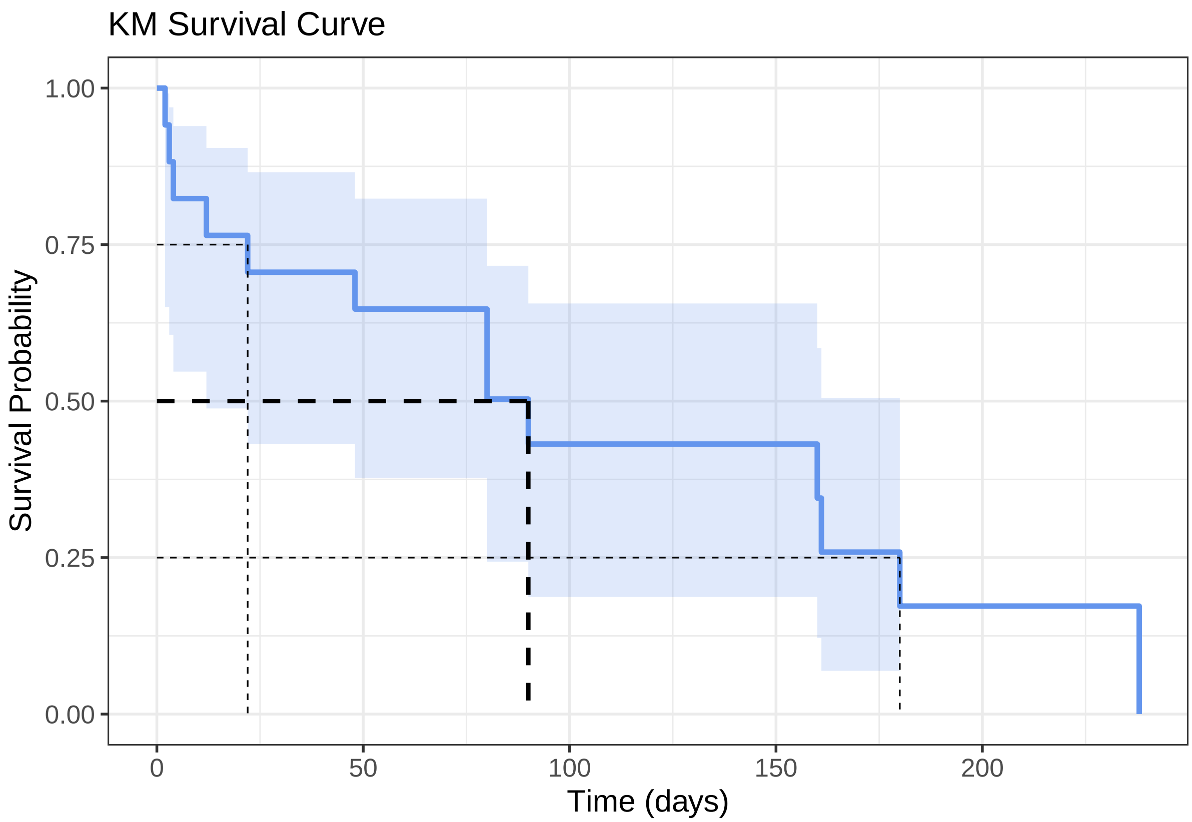
```r
# Linear CI
km_linear = survfit2(
  Surv(Value, Binary) ~ 1,
  data = q2_df,
  conf.type = "plain")

summary(km_linear)
```

```
## Call: survfit(formula = Surv(Value, Binary) ~ 1, data = q2_df, conf.type = "plain")
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     2     17       1    0.941  0.0571       0.8293        1.000
##     3     16       1    0.882  0.0781       0.7292        1.000
##     4     15       1    0.824  0.0925       0.6423        1.000
##    12     14       1    0.765  0.1029       0.5631        0.966
##    22     13       1    0.706  0.1105       0.4893        0.922
##    48     12       1    0.647  0.1159       0.4199        0.874
##    80      9       2    0.503  0.1272       0.2541        0.752
##    90      7       1    0.431  0.1277       0.1811        0.682
##   160      5       1    0.345  0.1280       0.0942        0.596
##   161      4       1    0.259  0.1217       0.0204        0.497
##   180      3       1    0.173  0.1074       0.0000        0.383
##   238      1       1    0.000     NaN          NaN          NaN
```

The "log-log" approach to calculating the 95% confidence intervals is done in order to keep the interval within the [0, 1] bounds of probability. The "linear" approach however is a simple $\hat{S}(t) \pm z_{1-\alpha/2}(SE)$ which can often lead to confidence intervals out of the [0, 1] interval. This does indeed happen with the above Linear CI calculation but the shown interval is truncated at 0.000 and 1.000 by the `survfit2()` function. Using the linear CI calculation, the upper 95% CI at $t_j = 2$ is:

$$\hat{S}(t) \pm z_{1-\alpha/2}(SE) = 0.941 + 1.96(0.0571) = 1.053$$

c.
```r
km_loglog |>
  ggsurvfit(color = "cornflowerblue", linewidth = 1) +
  add_confidence_interval(fill = "cornflowerblue") +
  add_quantile(y_value = 0.25, linewidth = 0.3) +
  add_quantile(y_value = 0.5, linewidth = 0.8) +
  add_quantile(y_value = 0.75, linewidth = 0.3) +
  labs(x = "Time (days)",
       title = "KM Survival Curve")
```



d.
```
km_loglog |>
  quantile(probs = c(0.25, 0.5, 0.75), conf.int = FALSE)
```

```
##  25  50  75
##  22  90 180
```

Using R, we can see the 25th percentile (22 days), the median or 50th percentile (90 days), and the 75th percentile (180 days). This is shown above on the plot by the horizontal dashed lines.

e. The cumulative hazard can be calculated from the KM survival estimate using the relationship

$$\hat{\Lambda}_{KM}(t) = -\log\left(\hat{S}_{KM}(t)\right)$$

```
surv_summary(km_loglog) |>
  mutate(
    km_cumhaz = -log(surv)) |>
  select(time, km_cumhaz)
```

```
##    time  km_cumhaz
## 1     2 0.06062462
## 2     3 0.12516314
## 3     4 0.19415601
## 4    12 0.26826399
## 5    22 0.34830669
## 6    48 0.43531807
## 7    51 0.43531807
## 8    56 0.43531807
## 9    80 0.68663250
## 10   90 0.84078318
```

```
## 11    94 0.84078318
## 12   160 1.06392673
## 13   161 1.35160880
## 14   180 1.75707391
## 15   238        Inf
```

f. The Nelson-Aalen cumulative hazard estimate can be calculated using

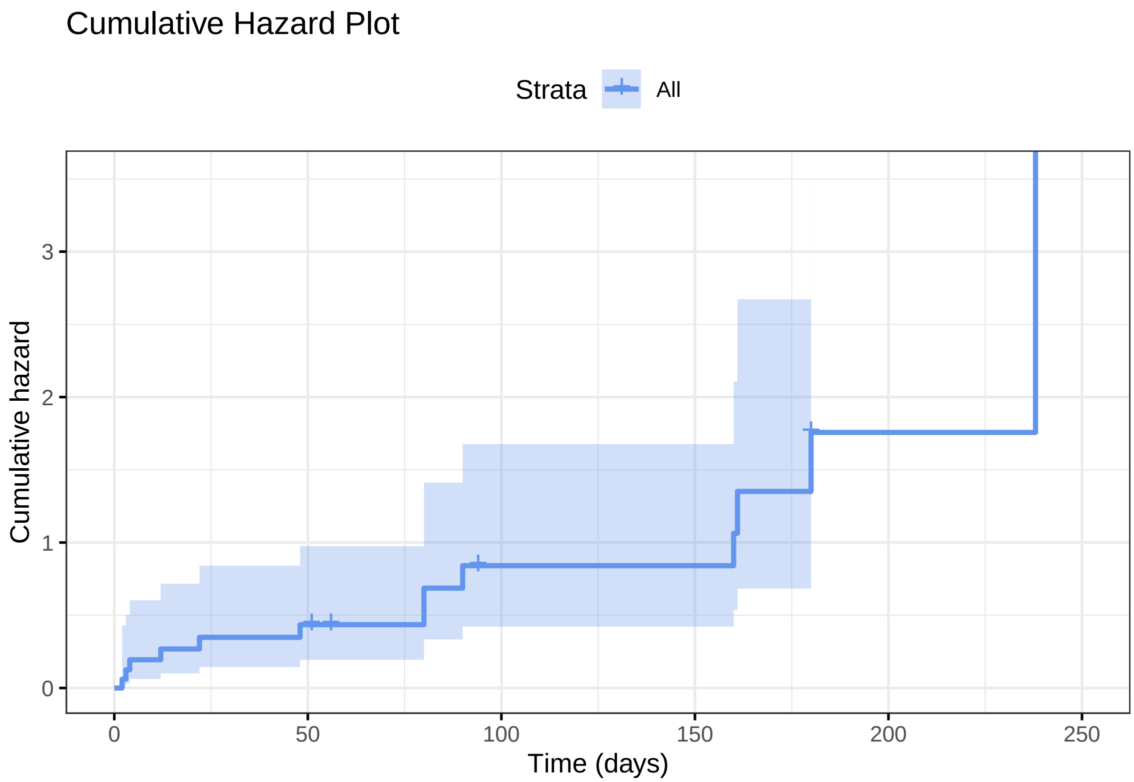
$$\hat{\Lambda}_{NA}(t) = \sum_{t_j \leq t} d_j/r_j$$

```r
na_df = surv_summary(km_loglog) |>
  mutate(
    hazard = n.event / n.risk,
    na_cumhaz = cumsum(hazard)) |>
  select(time, na_cumhaz)

na_df
```

```
##    time  na_cumhaz
## 1     2 0.05882353
## 2     3 0.12132353
## 3     4 0.18799020
## 4    12 0.25941877
## 5    22 0.33634184
## 6    48 0.41967518
## 7    51 0.41967518
## 8    56 0.41967518
## 9    80 0.64189740
## 10   90 0.78475454
## 11   94 0.78475454
## 12  160 0.98475454
## 13  161 1.23475454
## 14  180 1.56808788
## 15  238 2.56808788
```

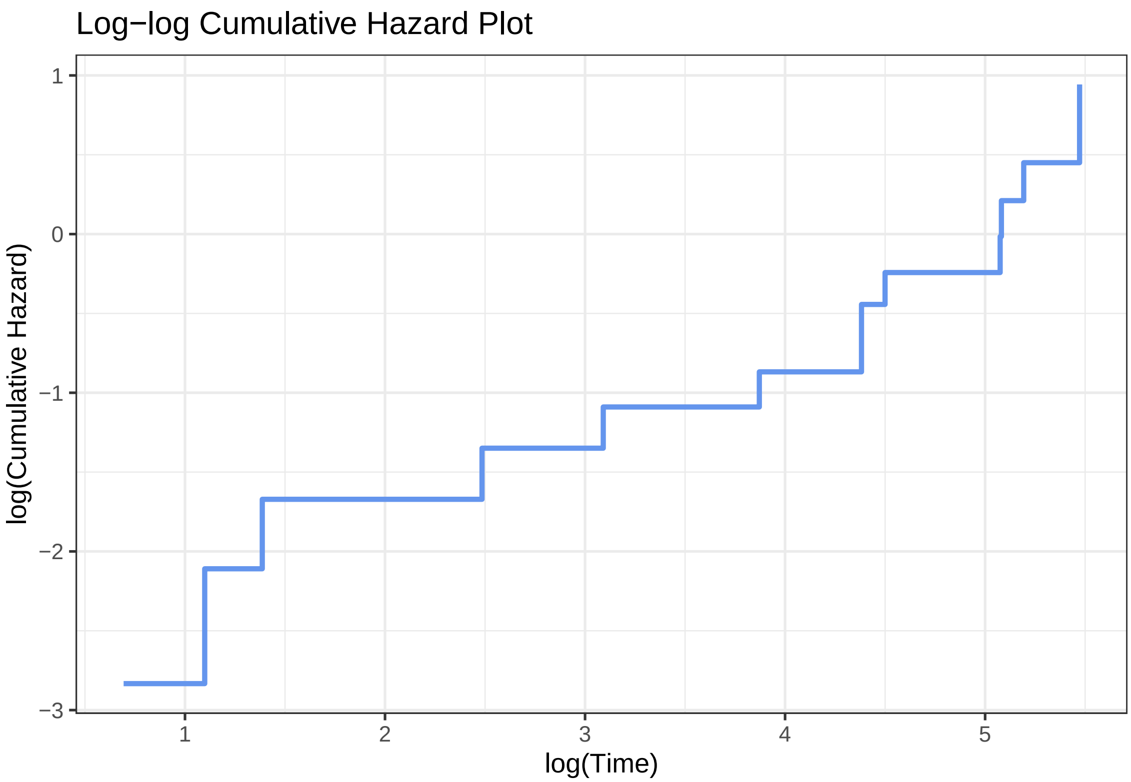
g. Cumulative Hazard plots

i.
```r
km_loglog |>
  ggsurvplot(data = q2_df,
             fun = "cumhaz",
             title = "Cumulative Hazard Plot",
             xlab = "Time (days)",
             palette = "cornflowerblue",
             ggtheme = theme_bw())
```



The cumulative hazard seems to be slightly concave up, which would indicate a slight increase in hazard over time. A Weibull distribution may fit better than an exponential distribution since hazard appears to not be constant.

ii.
```
surv_summary(km_loglog) |>
  mutate(
    hazard = n.event / n.risk,
    log_cumhaz = log(cumsum(hazard)),
    log_time = log(time)) |>
  ggplot(aes(x = log_time, y = log_cumhaz)) +
  geom_step(color = "cornflowerblue",
            linewidth = 1) +
  theme_bw() +
  labs(title = "Log-log Cumulative Hazard Plot",
       x = "log(Time)",
       y = "log(Cumulative Hazard)")
```



Using a $\log \hat{\Lambda}(t)$ vs $\log(t)$ plot can give an idea about what distribution could fit the data. The log cumulative hazard vs. log(time) plot appears roughly linear, meaning a Weibull or exponential distribution could be a good fit. However, for an exponential distribution to fit, the slope should be equal to 1 since hazard is constant. Since the slope seems to be slightly less than 1, an exponential distribution may not fit. However, with the small sample size, it is difficult to be certain, and more observations would help clarify the trend.

h. Using the Nelson-Aalen cumulative hazard, the Fleming-Harrington estimator can be calculated through the relationship

$$\hat{S}_{FH}(t) = \exp\left(-\hat{\Lambda}_{NA}(t)\right)$$

```
# Data-frame for time and Fleming-Harrington survival estimate
fh_df = na_df |>
  mutate(fh_surv = exp(-na_cumhaz)) |>
  select(time, fh_surv)

# Data-frame for time and Kaplan-Meier survival estimate
km_df = surv_summary(km_loglog) |>
  mutate(km_surv = surv) |>
  select(time, km_surv)

# Joining data-frames for comparison
left_join(fh_df, km_df, by = "time")
```

```
##    time    fh_surv   km_surv
## 1     2 0.94287314 0.9411765
## 2     3 0.88574735 0.8823529
## 3     4 0.82862283 0.8235294
## 4    12 0.77149988 0.7647059
## 5    22 0.71437886 0.7058824
```

```
## 6     48 0.65726028 0.6470588
## 7     51 0.65726028 0.6470588
## 8     56 0.65726028 0.6470588
## 9     80 0.52629289 0.5032680
## 10    90 0.45623167 0.4313725
## 11    94 0.45623167 0.4313725
## 12   160 0.37353090 0.3450980
## 13   161 0.29090616 0.2588235
## 14   180 0.20844337 0.1725490
## 15   238 0.07668203 0.0000000
```

The Fleming-Harrington estimator closely follows the Kaplan-Meier estimator, however remains slightly higher and notably does not drop to 0.

Question 3

a. The Actuarial Lifetable is constructed manually in R below. The Actuarial Estimate is shown under `act_surv` and is calculated using the function:

$$\hat{S}(t_j) = \prod_{\ell \leq j} \left(1 - \frac{d_\ell}{r'_\ell}\right)$$

```
lifetable_df = q2_df |>
  mutate(
    interval = cut(Value,
                   breaks = seq(0, max(Value) + 30, by = 30),
                   right = FALSE)) |>
  group_by(interval) |>
  summarise(
    d_j = sum(Binary == 1),
    c_j = sum(Binary == 0)) |>
  mutate(
    r_j = nrow(q2_df) - lag(cumsum(d_j + c_j), default = 0),
    rprime_j = r_j - (c_j / 2),
    qhat_j = d_j / rprime_j,
    act_surv = cumprod(1 - qhat_j))

lifetable_df
```

```
## # A tibble: 7 x 7
##   interval   d_j   c_j   r_j rprime_j qhat_j act_surv
##   <fct>    <int> <int> <int>    <dbl>  <dbl>    <dbl>
## 1 [0,30)       5     0    17     17   0.294     0.706
## 2 [30,60)      1     2    12     11   0.0909    0.642
## 3 [60,90)      2     0     9      9   0.222     0.499
## 4 [90,120)     1     1     7      6.5 0.154     0.422
## 5 [150,180)    2     0     5      5   0.4       0.253
## 6 [180,210)    1     1     3      2.5 0.4       0.152
## 7 [210,240)    1     0     1      1   1         0
```

b. The hazard function at the midpoint of each time interval can be calculated using:

$$\hat{\lambda}(t_{mj}) = \frac{d_j}{b_j(r'_j - d_j/2)}$$

where b_j is the interval width (30).

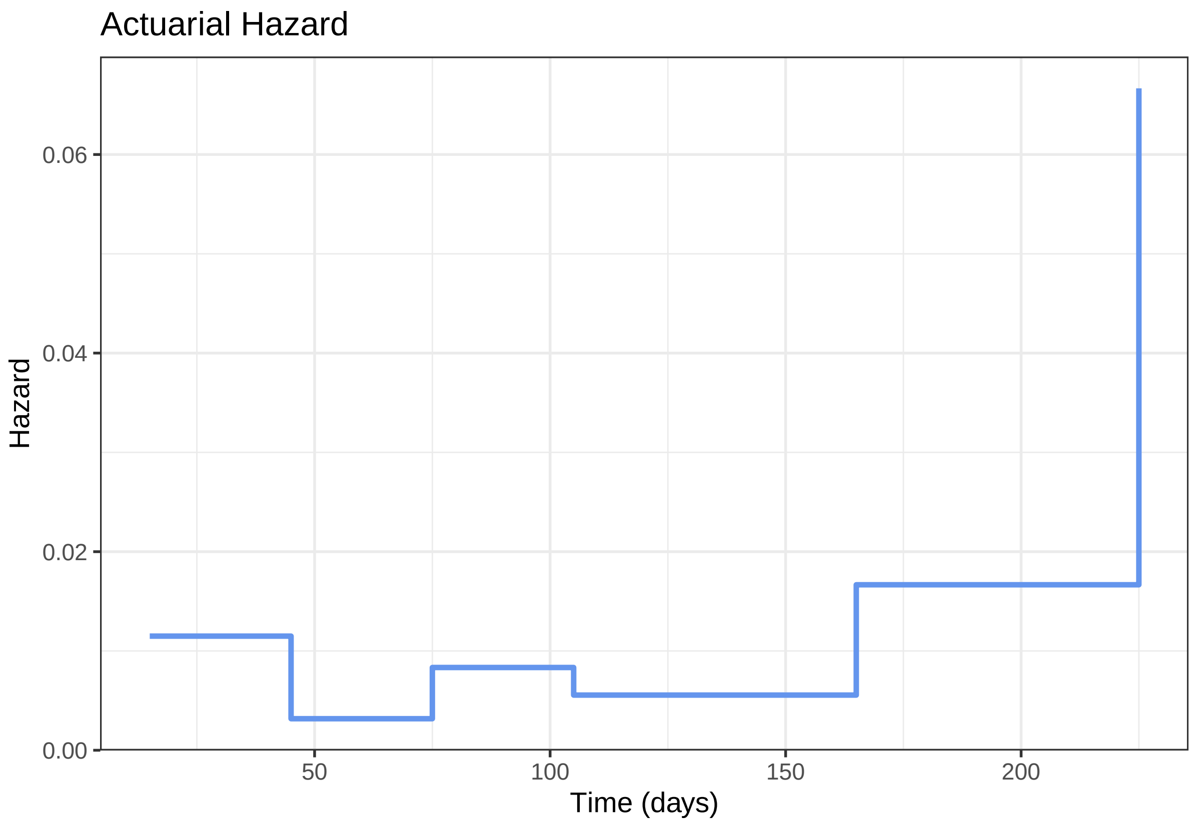
Adding hazard to the lifetable:

```
lifetable_df = lifetable_df |>
  mutate(
    hazard = d_j / (30 * (rprime_j - (d_j/2))))

lifetable_df
```

```
## # A tibble: 7 x 8
##   interval    d_j   c_j   r_j rprime_j qhat_j act_surv  hazard
##   <fct>     <int> <int> <int>    <dbl>  <dbl>    <dbl>   <dbl>
## 1 [0,30)        5     0    17     17    0.294     0.706 0.0115
## 2 [30,60)       1     2    12     11    0.0909    0.642 0.00317
## 3 [60,90)       2     0     9      9    0.222     0.499 0.00833
## 4 [90,120)      1     1     7      6.5  0.154     0.422 0.00556
## 5 [150,180)     2     0     5      5    0.4       0.253 0.0167
## 6 [180,210)     1     1     3      2.5  0.4       0.152 0.0167
## 7 [210,240)     1     0     1      1    1         0     0.0667
```

Plotting hazard over time:

```
lifetable_df |>
  mutate(
    lower = as.numeric(sub("\\[(\\d+),.*", "\\1", interval)),
    upper = as.numeric(sub(".*,([0-9]+)\\)", "\\1", interval)),
    midpoint = (lower + upper) / 2,) |>
  ggplot(aes(x = midpoint, y = hazard)) +
  geom_step(color = "cornflowerblue", linewidth = 1) +
  theme_bw() +
  labs(
    title = "Actuarial Hazard",
    x = "Time (days)",
    y = "Hazard")
```



c. From the plot above, hazard seems to increase with time. It is relatively stable at first, but starts to increase in the last few intervals. Since an exponential model relies on hazard remaining constant, an exponential model would not be appropriate.