Survival Analysis for Algal Growth: A Handout for Korey & Maria

Marc Los Huertos

November 19, 2025

Contents

1 Goal and Justification

The purpose of this handout is to compare the time it takes for algal cultures under three different treatments to reach a specific growth milestone (an absorbance threshold). Survival analysis is used because it correctly handles cultures that never reach the threshold (right-censoring) and provides robust, interpretable comparisons of growth rates over time.

2 Explanation for Korey and Maria

Survival analysis focuses on time-to-event data. In our case, the "event" is reaching a chosen optical density (OD) threshold. This approach allows us to include both cultures that reach the threshold and those that do not, without bias.

Survival Term	Algae Equivalent	Statistical Role
Event	Reaching OD threshold (e.g., OD = 0.6)	Defines the failure point we are trackin
Time	Hours until OD $\geq$ threshold	The variable we are modeling
Censoring	Not reaching threshold by experiment end	Allows us to use incomplete data
Hazard Ratio	Relative rate of reaching threshold	HR $>$ 1 means faster growth

3 Packages and Data Setup

We first load the required packages. These provide tools for data manipulation (tidyverse), survival analysis (survival), visualization (survminer), and tidying model output (broom).

```
library(tidyverse)

## -- Attaching core tidyverse packages ------------------------
tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts()
--
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>)
to force all conflicts to become errors

library(survival)
library(survminer)
```

```
## Loading required package:  ggpubr
##
## Attaching package:  'survminer'
##
## The following object is masked from 'package:survival':
##
##     myeloma
```

```
library(broom)
```

3.1 Data Simulation

We simulate growth data for three treatments. Each treatment has slightly different average growth rates and variability. The function `make_growth` generates noisy logistic growth curves, some of which never reach the threshold (to mimic censoring).

```
set.seed(2025)
n_rep <- 10
times <- seq(0, 72, by = 8)

make_growth <- function(n, treatment_label, mu_time_to_mid = 30, sd_time = 6, maxOD =
  tibble(sample = paste0(treatment_label, "_", seq_len(n))) %>%
    rowwise() %>%
    mutate(
      t_mid = rnorm(1, mu_time_to_mid, sd_time) |> pmax(6),
      slope = (maxOD) / (t_mid + 0.1),
      never = runif(1) < prop_no_reach
    ) %>%
    ungroup() %>%
    expand_grid(time = times) %>%
    rowwise() %>%
    mutate(
      mu = plogis((time - rnorm(1, mu_time_to_mid, sd_time))/7) * maxOD,
      absorbance = mu + rnorm(1, 0, 0.03)
    ) %>%
    ungroup() %>%
    mutate(treatment = treatment_label)
}
```

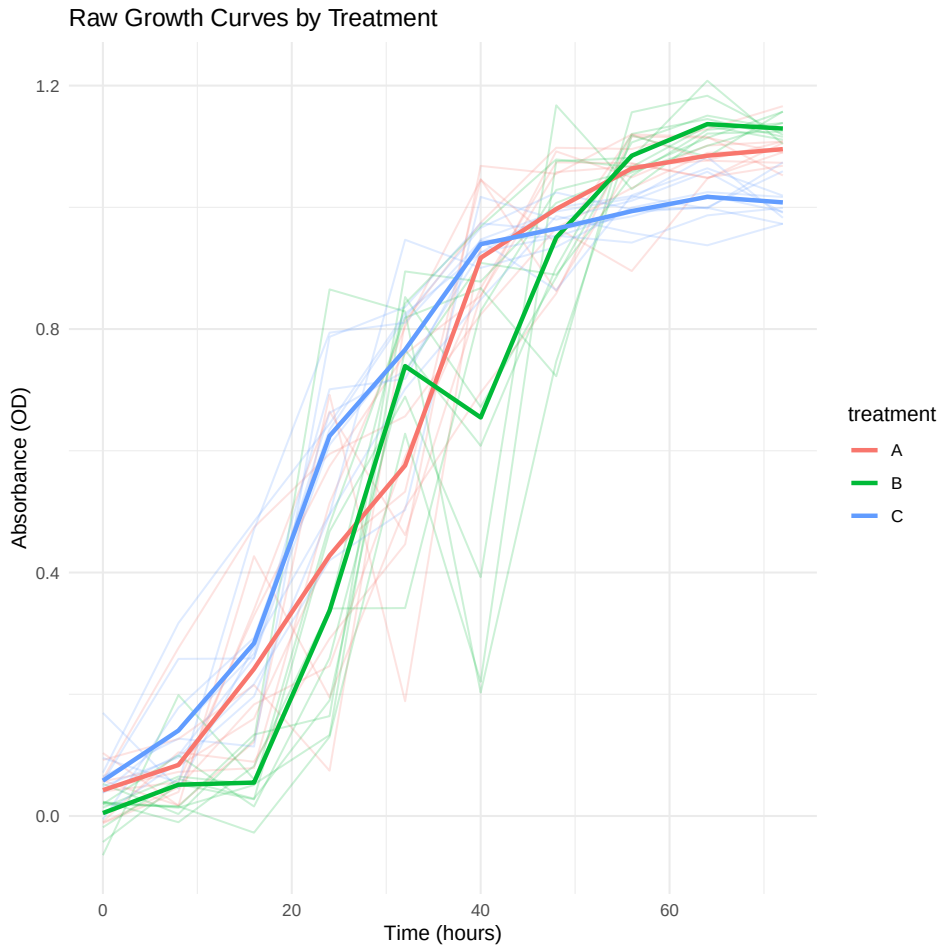
```r
df_A <- make_growth(n_rep, "A", mu_time_to_mid = 28, sd_time = 6, maxOD = 1.1, prop_n
df_B <- make_growth(n_rep, "B", mu_time_to_mid = 34, sd_time = 7, maxOD = 1.15, prop_
df_C <- make_growth(n_rep, "C", mu_time_to_mid = 22, sd_time = 5, maxOD = 1.0, prop_n

df_raw <- bind_rows(df_A, df_B, df_C) %>%
  mutate(treatment = factor(treatment))
```

4 Visualization and Event Definition

4.1 Raw Growth Curves

We plot individual growth curves and treatment means. This shows variability and average trends.

```r
df_means <- df_raw %>%
  group_by(treatment, time) %>%
  summarise(mean_abs = mean(absorbance), .groups = "drop")

p_raw <- ggplot() +
  geom_line(data = df_raw,
            aes(time, absorbance, group = sample, color = treatment),
            alpha = 0.2, show.legend = FALSE) +
  geom_line(data = df_means,
            aes(time, mean_abs, color = treatment),
            linewidth = 1.1) +
  labs(x = "Time (hours)",
       y = "Absorbance (OD)",
       title = "Raw Growth Curves by Treatment") +
  theme_minimal()
p_raw
```

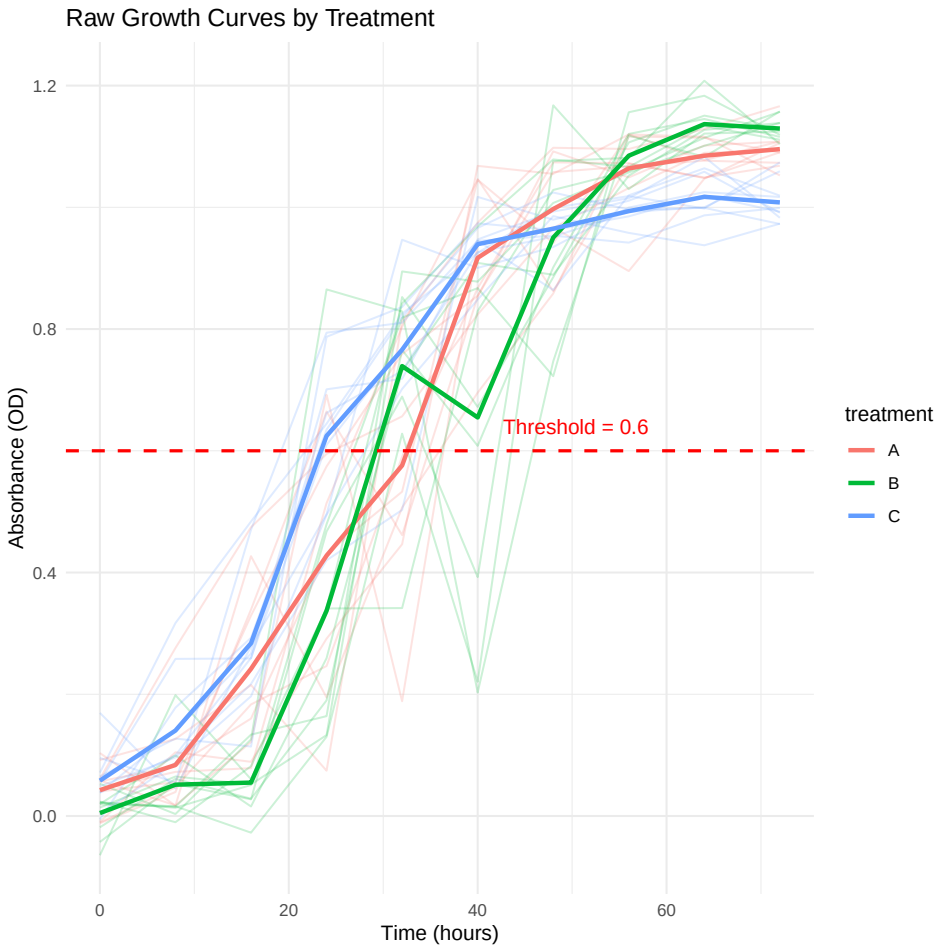


4.2 Defining the Event

We define the event as the first time a culture reaches OD ≥ 0.6. If a culture never reaches this threshold, it is censored at its last observation time.

```
threshold <- 0.6

time_to_event <- df_raw %>%
  group_by(sample, treatment) %>%
  arrange(time) %>%
  summarise(
```

```
    event_time = {
      hit_rows <- which(absorbance >= threshold)
      if(length(hit_rows) == 0) NA_real_ else time[min(hit_rows)]
    },
    last_time = max(time),
    .groups = "drop"
  ) %>%
  mutate(
    status = if_else(is.na(event_time), 0L, 1L),
    time = if_else(is.na(event_time), last_time, event_time)
  )

cat("Censoring Status:\n")

## Censoring Status:

time_to_event %>% count(treatment, status)

## # A tibble: 3 x 3
##   treatment status     n
##   <fct>      <int> <int>
## 1 A              1    10
## 2 B              1    10
## 3 C              1    10

p_raw + geom_hline(yintercept = threshold, linetype = "dashed", color = "red", linewi
  annotate("text", x = max(df_raw$time)*0.7, y = threshold + 0.04, label = paste0("Th
```



5 Survival Analysis and Modeling

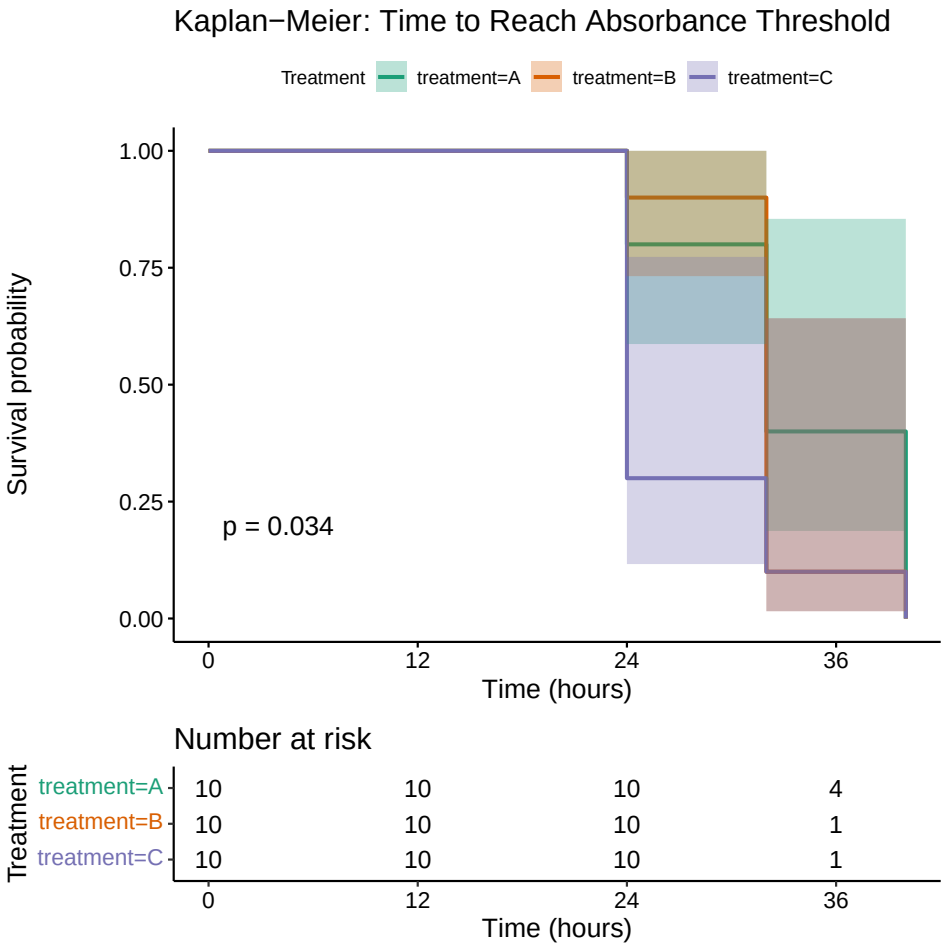
5.1 Kaplan-Meier Curves

Kaplan-Meier curves estimate the probability of not yet reaching the threshold over time. Each step down represents cultures reaching the event.

```
km_fit <- survfit(Surv(time, status) ~ treatment, data = time_to_event)
summary(km_fit)

## Call: survfit(formula = Surv(time, status) ~ treatment, data = time_to_event)
```

```
##
##                 treatment=A
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    24     10       2      0.8   0.126        0.587        1.000
##    32      8       4      0.4   0.155        0.187        0.855
##    40      4       4      0.0     NaN           NA           NA
##
##                 treatment=B
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    24     10       1      0.9  0.0949       0.7320        1.000
##    32      9       8      0.1  0.0949       0.0156        0.642
##    40      1       1      0.0     NaN           NA           NA
##
##                 treatment=C
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    24     10       7      0.3  0.1449       0.1164        0.773
##    32      3       2      0.1  0.0949       0.0156        0.642
##    40      1       1      0.0     NaN           NA           NA
```

```
ggsurvplot(km_fit, data = time_to_event, risk.table = TRUE, pval = TRUE,
           conf.int = TRUE, palette = "Dark2",
           title = "Kaplan-Meier: Time to Reach Absorbance Threshold",
           xlab = "Time (hours)", legend.title = "Treatment",
           break.time.by = 12)
```



5.2 Log-Rank Test

The log-rank test compares survival curves across treatments. Null hypothesis: no difference between groups.

```
logrank <- survdiff(Surv(time, status) ~ treatment, data = time_to_event)
logrank

## Call:
## survdiff(formula = Surv(time, status) ~ treatment, data = time_to_event)
##
```

```
##              N Observed Expected (O-E)^2/E (O-E)^2/V
## treatment=A 10       10    12.93    0.6653     3.318
## treatment=B 10       10    10.63    0.0377     0.153
## treatment=C 10       10     6.43    1.9774     6.069
##
##  Chisq= 6.7  on 2 degrees of freedom, p= 0.03

p_val <- 1 - pchisq(logrank$chisq, df = length(logrank$n) - 1)
cat("Omnibus log-rank p-value:", signif(p_val, 3), "\n")

## Omnibus log-rank p-value: 0.0345
```

5.3 Cox Proportional Hazards Model

The Cox model estimates hazard ratios (HR). HR > 1 means faster reaching of the threshold compared to the reference group. Assumption: proportional hazards (HR constant over time).

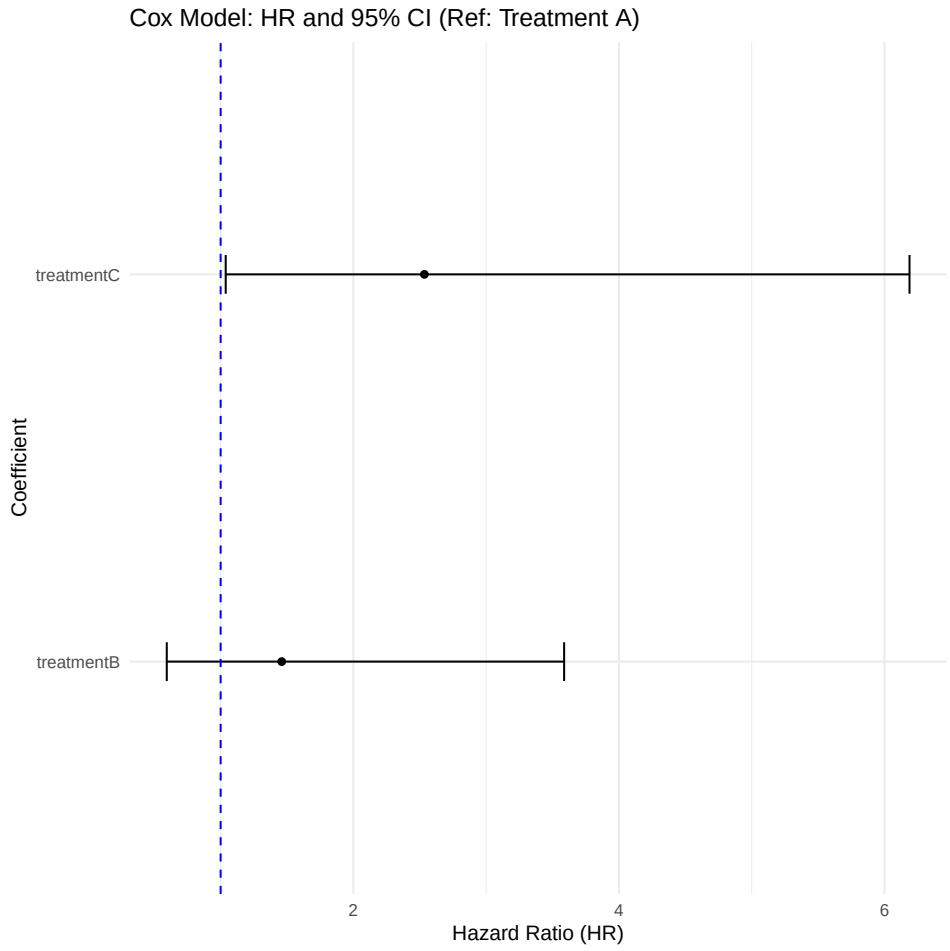
```
time_to_event <- time_to_event %>% mutate(treatment = relevel(treatment, ref = "A"))
cox1 <- coxph(Surv(time, status) ~ treatment, data = time_to_event)
summary(cox1)

## Call:
## coxph(formula = Surv(time, status) ~ treatment, data = time_to_event)
##
##   n= 30, number of events= 30
##
##              coef exp(coef) se(coef)     z Pr(>|z|)
## treatmentB 0.3785    1.4601   0.4586 0.825   0.4092
## treatmentC 0.9301    2.5347   0.4554 2.042   0.0411 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##            exp(coef) exp(-coef) lower .95 upper .95
## treatmentB     1.460     0.6849    0.5943     3.587
## treatmentC     2.535     0.3945    1.0382     6.189
##
## Concordance= 0.722  (se = 0.085 )
## Likelihood ratio test= 4.04  on 2 df,   p=0.1
```

```
## Wald test            = 4.23  on 2 df,   p=0.1
## Score (logrank) test = 4.45  on 2 df,   p=0.1
```

5.4 Hazard Ratio Forest Plot

We visualize HR estimates and confidence intervals. If CI does not cross 1, the effect is statistically significant.

```
hr <- tidy(cox1, exponentiate = TRUE, conf.int = TRUE)

ggplot(hr, aes(x = term, y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high), width = 0.1) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "blue") +
  coord_flip() +
  labs(y = "Hazard Ratio (HR)", x = "Coefficient", title = "Cox Model: HR and 95% CI
  theme_minimal()
```



6 Assumptions Check

6.1 Proportional Hazards Assumption

The Cox model assumes hazard ratios are constant over time. We test this with Schoenfeld residuals. A non-significant p-value ($p > 0.05$) indicates the assumption holds.

```
ph_test <- cox.zph(cox1)
ph_test
```

```
##           chisq df    p
```

```
## treatment  3.61  2 0.16
## GLOBAL     3.61  2 0.16

plot(ph)

## Error in plot(ph):  object 'ph' not found
```