

# PSTAT175 - Final Project - Part 1 - Recurrence

Zifeng(Robin) Zhan, Calder Glass, Kotaro Ito

2025-04-29

We used a significance level of  $\alpha = 0.05$  as the critical value.

```
data("colon")
```

```
## Warning in data("colon"): data set 'colon' not found
```

```
View(colon)
```

```
library(survival)
library(survminer)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## Warning: package 'purrr' was built under R version 4.4.3
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## v forcats   1.0.0      v stringr   1.5.1
## v lubridate 1.9.4      v tibble   3.2.1
## v purrr     1.0.4      v tidyr    1.3.1
## v readr     2.1.5
```

```
## -- 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
```

```
data("colon")
```

```
## Warning in data("colon"): data set 'colon' not found
```

```
View(colon)
```

```
sum(is.na(colon))
```

```
## [1] 82
```

```
# There are 82 missing observations - accounts for 4.4% of the data, safe enough to remove those observations
```

```
colon = tibble(colon)
```

```
colon <- colon %>%  
  drop_na()
```

```
# data for recurrence event
```

```
recurrence_data <- colon[colon$type == 1, ]
```

```
# survival object
```

```
surv_object_recurrence <- Surv(time = recurrence_data$time,  
                                event = recurrence_data$status)
```

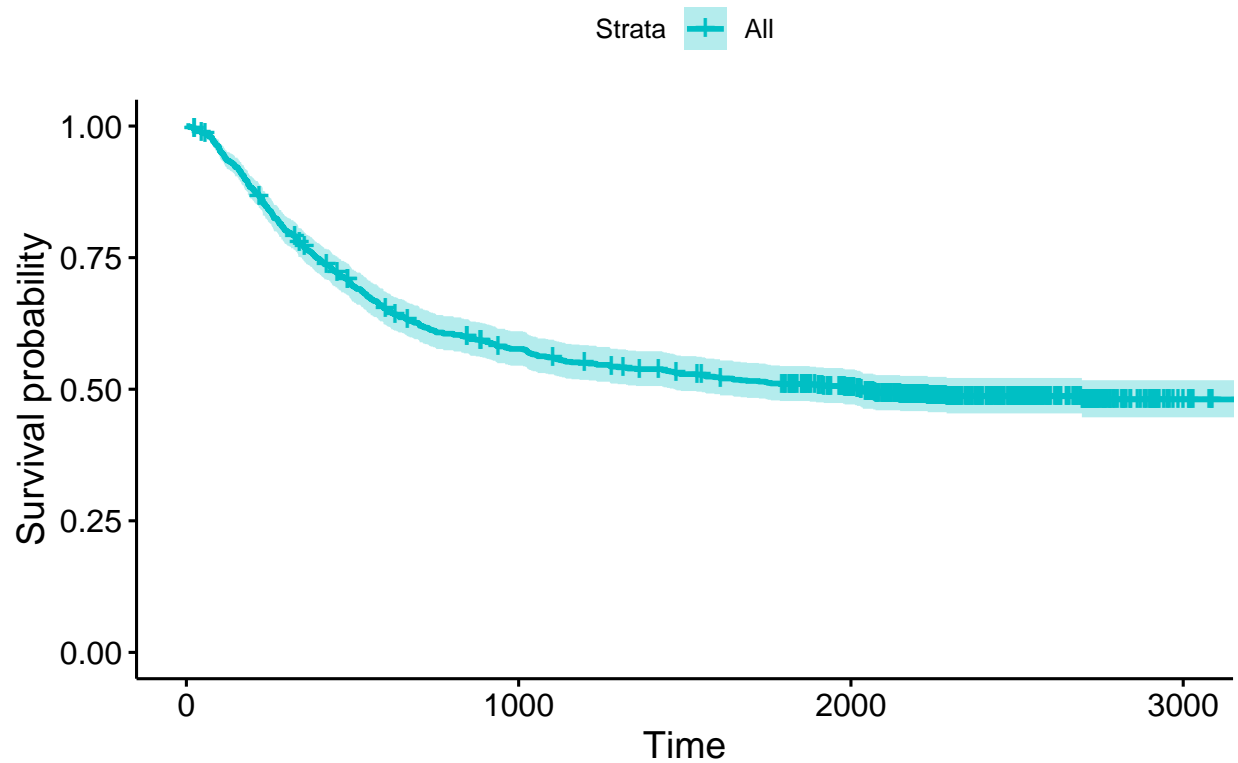
```
# Fit the Kaplan-Meier model for recurrence events
```

```
km_fit_recurrence <- survfit(surv_object_recurrence ~ 1)
```

```
# Plot the Kaplan-Meier survival curve for recurrence events with a title
```

```
ggsurvplot(km_fit_recurrence,  
            data = recurrence_data,  
            palette = "#00BFC4", # Customize the color  
            title = "Kaplan-Meier Survival Curve for Recurrence Events")
```

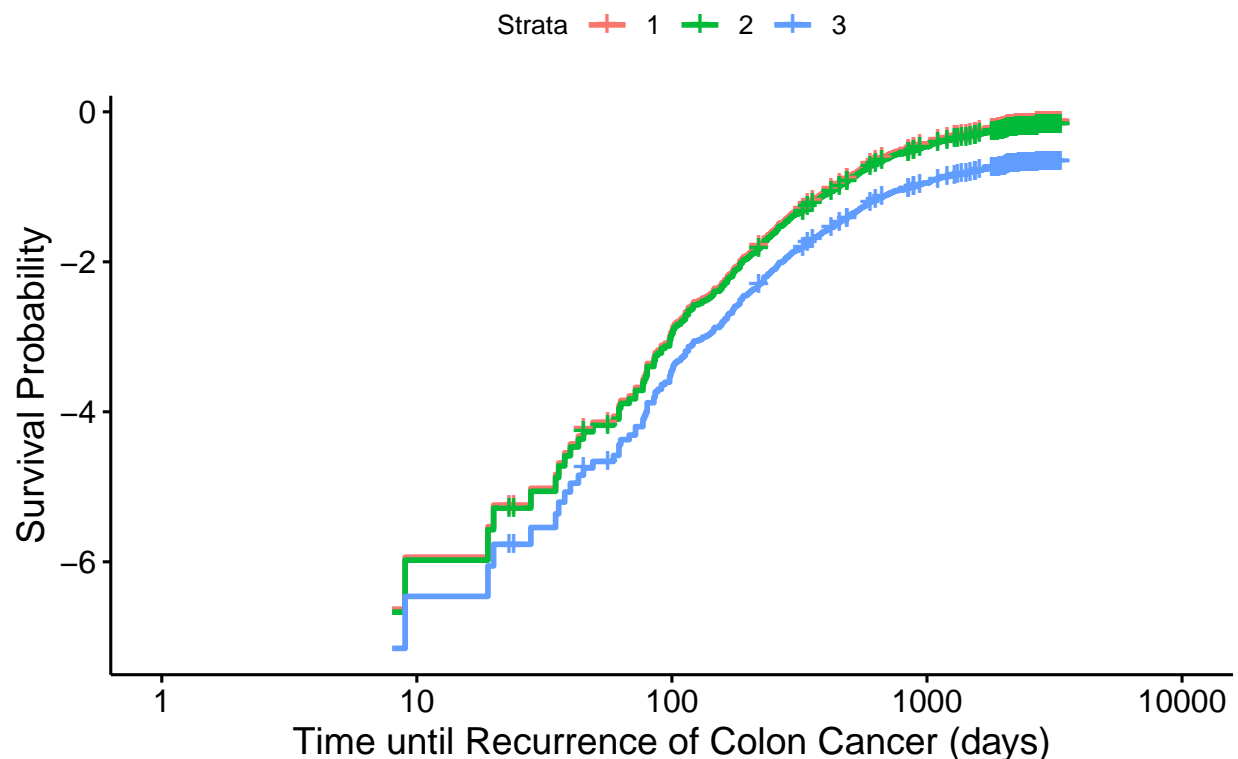
## Kaplan–Meier Survival Curve for Recurrence Events



Then we checked if the Cox Proportional Hazards assumption was violated or not in terms of the treatment covariate.

```
recurrence_obj = coxph(Surv(time, status) ~ rx, data = recurrence_data)
recurrence_fit = survfit(recurrence_obj, newdata = data.frame(rx= c("Obs", "Lev", "Lev+5FU")))
ggsurvplot(recurrence_fit, data = recurrence_data, fun = "cloglog", conf.int = FALSE, title = "Comparison")
```

## Comparison of Survival Functions for Different Treatments



The survival curves are fairly parallel, so it's reasonable to assume that the Cox Proportional Hazards Assumption for the recurrence portion of the data is not violated by the treatment covariate.

Covariates that are either binaries or have very few levels, 3 to 4 maximum, are factored so that the effect of each level of the covariates can be summarized.

```
# Factor some of the covariates first in order to do the cox ph tests:
```

```
colon <- colon %>%  
  mutate_at(c("sex", "obstruct", "perfor", "adhere", "differ", "etype", "node4", "surg", "extent"), fac
```

AIC:

Now that the treatment covariate has been confirmed to not violate the Cox Proportional Hazards Assumption and the recurrence subset of colon cancer data has been cleaned, the next step is to find the model with the most significant covariates under the AIC criterion.

For the AIC tests, the covariates *study* and *id* are not included. The *id* covariate is the same as the observation number, it doesn't have contextual significance to the event of relapse or death from colon cancer. The *study* covariate is not included as all of the subjects are from the same study.

```
# Level 1:
```

```
# list of covariates to put into the models
```

```
recurrence_covariates = c("obstruct", "adhere", "nodes", "node4", "differ", "extent", "surg", "perfor",
```

```
# building a model per covariate by pasting the given covariate into the formula
```

```
# the set_names function helps to clear up which AIC value corresponds to which model when performing t
```

```
recurrence_models = map(recurrence_covariates, \(v) coxph(as.formula(paste("Surv(time, status) ~ rx + ", v)))

aic_lv11 = map_dbl(recurrence_models, AIC) |>
  sort()

aic_lv11
```

```
##      node4      nodes      extent      differ      adhere      surg obstruct      perfor
## 5666.763 5677.508 5710.227 5728.937 5732.669 5733.290 5733.868 5735.037
##      sex      age
## 5735.202 5735.275
```

The model with the *node4* covariate, the binary variable for whether the patient had more than 4 positive lymph nodes, had the lowest AIC.

Since the *nodes* covariate and *node4* covariate are closely related, the *nodes* covariate will be skipped.

Therefore, forward selection proceeds with the above covariate.

```
# Level 2:
# list of covariates to put into the models
recurrence_covariates2 = c("obstruct", "adhere", "differ", "extent", "surg", "perfor", "sex", "age")

# building a model per covariate by pasting the given covariate into the formula

# the set_names function helps to clear up which AIC value corresponds to which model when performing t

recurrence_models2 = map(recurrence_covariates2, \(v) coxph(as.formula(paste("Surv(time, status) ~ rx + ", v)))

aic_lv12 = map_dbl(recurrence_models2, AIC) |>
  sort()

aic_lv12
```

```
##      extent obstruct      differ      surg      adhere      perfor      sex      age
## 5647.130 5663.826 5664.174 5664.489 5664.654 5666.367 5667.227 5668.279
```

The model with the *extent* covariate, the description of the local spread of the tumor, had the lowest AIC.

Therefore, forward selection proceeds with the above covariate.

```
# Level 3:
# list of covariates to put into the models
recurrence_covariates3 = c("obstruct", "adhere", "differ", "surg", "perfor", "sex", "age")

# building a model per covariate by pasting the given covariate into the formula

# the set_names function helps to clear up which AIC value corresponds to which model when performing t

recurrence_models3 = map(recurrence_covariates3, \(v) coxph(as.formula(paste("Surv(time, status) ~ rx + ", v)))

aic_lv13 = map_dbl(recurrence_models3, AIC) |>
  sort()

aic_lv13
```

```
##      surg      differ obstruct      adhere      perfor      sex      age
## 5644.216 5646.022 5646.036 5646.736 5647.565 5647.708 5648.738
```

The model with the *surg* covariate, the time from initial surgery to registration in the study, had the lowest AIC.

Therefore, forward selection proceeds with the above covariate.

```
# Level 4:
# list of covariates to put into the models
recurrence_covariates4 = c("obstruct", "adhere", "differ", "perfor", "sex", "age")

# building a model per covariate by pasting the given covariate into the formula

# the set_names function helps to clear up which AIC value corresponds to which model when performing t

recurrence_models4 = map(recurrence_covariates4, \(v) coxph(as.formula(paste("Surv(time, status) ~ rx +

aic_lv14 = map_dbl(recurrence_models4, AIC) |>
  sort()

aic_lv14
```

```
##      differ obstruct      adhere      perfor      sex      age
## 5643.089 5643.385 5643.903 5644.596 5644.625 5645.735
```

The model with the *differ* covariate, the description of the removed cancer cells from the colon, had the lowest AIC.

Therefore, forward selection proceeds with the above covariate.

```
# Level 5:
# list of covariates to put into the models
recurrence_covariates5 = c("adhere", "obstruct", "perfor", "sex", "age")

# building a model per covariate by pasting the given covariate into the formula

# the set_names function helps to clear up which AIC value corresponds to which model when performing t

recurrence_models5 = map(recurrence_covariates5, \(v) coxph(as.formula(paste("Surv(time, status) ~ rx +

aic_lv15 = map_dbl(recurrence_models5, AIC) |>
  sort()

aic_lv15
```

```
## obstruct      adhere      sex      perfor      age
## 5641.966 5643.228 5643.275 5643.552 5644.507
```

The model with the *obstruct* covariate, the binary variable for whether the cancer had adhered to other organs, had the lowest AIC.

Therefore, forward selection proceeds with the above covariate.

```

# Level 6:
# list of covariates to put into the models
recurrence_covariates6 = c("adhere", "perfor", "sex", "age")

# building a model per covariate by pasting the given covariate into the formula

# the set_names function helps to clear up which AIC value corresponds to which model when performing t

recurrence_models6 = map(recurrence_covariates6, \(v) coxph(as.formula(paste("Surv(time, status) ~ rx +", v)))

aic_lvl6 = map_dbl(recurrence_models6, AIC) |>
  sort()

aic_lvl6

```

```

##   adhere      sex   perfor      age
## 5642.073 5642.383 5642.879 5643.608

```

None of the AICs shown above are less than the previous model, so the chosen model has the following covariates: *obstruct*, *surg*, *extent*, *node4*, and *differ*.

Next, the model was summarized in order to conclude relationships between the different levels of covariates, such as the treatment covariate and the differentiation covariate.

```
summary(coxph(Surv(time, status) ~ rx + node4 + extent + surg + differ + obstruct, data = recurrence_data))
```

```

## Call:
## coxph(formula = Surv(time, status) ~ rx + node4 + extent + surg +
##       differ + obstruct, data = recurrence_data)
##
##      n= 888, number of events= 446
##
##              coef exp(coef) se(coef)      z Pr(>|z|)
## rxLev      -0.00597   0.99405  0.11039 -0.054   0.9569
## rxLev+5FU -0.49184   0.61150  0.12168 -4.042 5.3e-05 ***
## node4       0.83270   2.29952  0.09942  8.375 < 2e-16 ***
## extent      0.49861   1.64643  0.11946  4.174 3.0e-05 ***
## surg        0.22771   1.25572  0.10393  2.191 0.0284 *
## differ      0.18001   1.19723  0.09706  1.855 0.0637 .
## obstruct    0.21228   1.23649  0.11773  1.803 0.0714 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##              exp(coef) exp(-coef) lower .95 upper .95
## rxLev          0.9940      1.0060    0.8006  1.2342
## rxLev+5FU      0.6115      1.6353    0.4818  0.7762
## node4          2.2995      0.4349    1.8924  2.7943
## extent         1.6464      0.6074    1.3027  2.0808
## surg           1.2557      0.7964    1.0243  1.5394
## differ         1.1972      0.8353    0.9898  1.4481
## obstruct       1.2365      0.8087    0.9817  1.5574
##
## Concordance= 0.661 (se = 0.013 )

```

```
## Likelihood ratio test= 126.3 on 7 df, p=<2e-16
## Wald test           = 127.1 on 7 df, p=<2e-16
## Score (logrank) test = 132.7 on 7 df, p=<2e-16
```

From the likelihood ratio test, the p-value is less than  $2e - 16$ , which is much less than the critical value/significance level of 0.05.

The hazard rate for patients who took the treatment with just Levamisole is 1.163% less hazardous than taking no treatment at all. Those who took Fluoracil in addition to Levamisole benefited with a hazard ratio of 0.6065, 39.35% less hazardous than no treatment at all.

Patients who had more than 4 positive lymph nodes had over double the hazard rate of those who didn't.

As the spread of the tumor developed from muscles to contiguous structures, the hazard ratio to those who only had submucosa development increased to as high as 3.64 times as likely to suffer a recurrence of colon cancer.

Patients with a long time from their initial surgery to registration in the study had a 25% greater hazard rate than those with a shorter time interval.

Patients whose removed cancer cells were "moderately differentiated" had a 3.24% lower hazard rate than patients whose cancer cells were "well differentiated", while those with "poorly differentiated" cells had 31.25% higher hazard rate compared to same base group.

Patients whose colons were obstructed by a tumor had a 23.39% higher hazard rate compared to those who were obstruction-free.

With the following exceptions of the treatment level that included Fluoracil and Levamisole, the node level of patients who had more than 4 positive lymph nodes, and the long time interval level between initial surgery to registering for the study, all of the other covariates' levels had 95% confidence intervals which contained the baseline 1. This suggests that the most significant levels of covariates in their effect on the hazard rate of the recurrence of colon cancer are Levamisole + Fluoracil as a treatment, over 4 positive lymph nodes, spread of cancer to the contiguous structures, and a long time between initial surgery to registration for the study.

Counting Process Model:

It is important to investigate the effect of the covariates in regards to the counting process model.