Colon Cancer Survival Analysis

Calder Glass, Kotaro Ito, Robin Zhan

2025-06-01

Introduction

We explore the *colon* dataset found in the R "survival" package, containing data observed from a clinical trial where stage B/C colon cancer patients receive adjuvant chemotherapy. 929 independent patients–484 men and 445 women–were randomly assigned between two treatments and a control group: Levamisole, Levamisole and Fluorouracil, and control(denoted as `Lev`, `Lev+5FU`, and `Obs` in the dataset). "Levamisole is a low-toxicity compound that was originally used to treat worm infestations in animals", while "5-FU is a moderately toxic chemotherapy agent" used to treat cancer.[1]

Patients were then observed until one of two events occurred: recurrence or death (denoted as "1" and "2" in its respective order under column `etype`). The time of occurrence, in days, was then recorded to later investigate and determine whether or not different treatments were effective in keeping the patients alive. Each patient in the dataset, identified by their `id`, has two rows for both recurrence and death. The status column indicates whether or not the event occurred or not ("0" indicates no and "1" indicates yes). If a patient has been recorded for 3000 days for both recurrence and death and the status remains 0 for both, it signifies that they did not experience any event for 3000 days and dropped out of the study for unknown reasons. Figure 1 and 2 below represents the Kaplan-Meier Survival Curve after splitting the dataset by `etype` (recurrence and death). The convex shape of Figure 1 conveys that many recurrences occur early on while the curve for death events show that deaths in patients are gradual and consistent.

Figure 1: KP for Recurrence Events

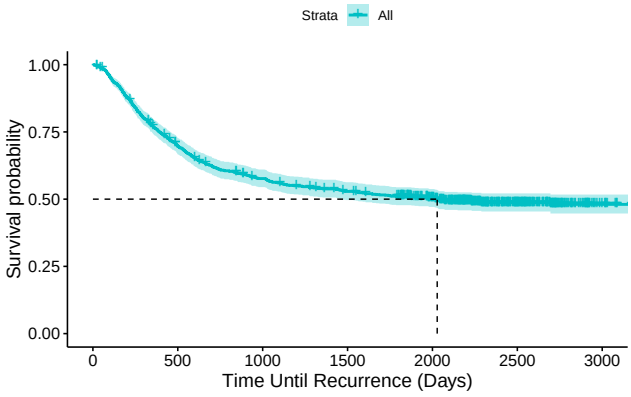
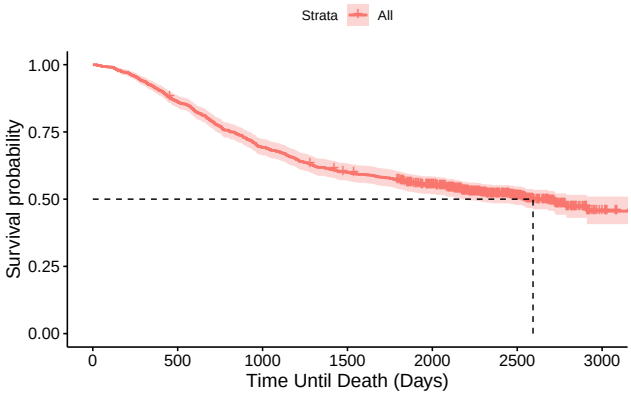


Figure 2: KP for Death Events



The dataset contains the `id`, `age`(in years), `sex` ("1" indicates man and "0" indicates woman), `rx` (the treatment type or control), `obstruct` ("1" indicates a colon obstructed by a tumor and "0" indicates no obstruction), `perfor` ("1" indicates a perforated colon and "0" indicates no perforation), `adhere` ("1" indicates cancer adhering to other organs and "0" indicates no adherence), `nodes` (the number of lymph nodes with colon cancer), `time` (time until event occurrence or censoring), `status` (whether or not the event occurred or not), `differ` ("3" indicates quickly growing cancer, "2" indicates moderate growth, and "1" indicates slowly

[1] colon: Chemotherapy for Stage B/C colon cancer. Colon Cancer. Retrieved May 1st, 2025, from https://rdrr.io/cran/survival/man/colon.html

growing and less likely to spread), `extent` (describes the spread of the tumor and ranges from 1-4, where "1" indicates that the tumor is limited to the inner lining of the colon and "4" indicates invasion of tumor to nearby organs and tissues), `surg` ("1" indicates a long time between initial surgery and registering to the study while "0" indicates a short time), `node4` ("1" indicates a patient has more than four positive lymph nodes and "0" indicates four or less), and `etype` (recurrence or death event) of each patient.[1]

Taking all of the covariates we listed above into consideration, our aim is to determine whether or not a specific treatment has a significant effect on the survival of the patient. Our secondary objective is to assess which covariate(s) have a significant effect on the hazard risk. In the course of the analysis, we omit observations with N/A values, reducing our final dataset to 888 independent patients. A five percent significance level (0.05) will be used to balance the risk of false positives to detect meaningful effects.

Model Fitting

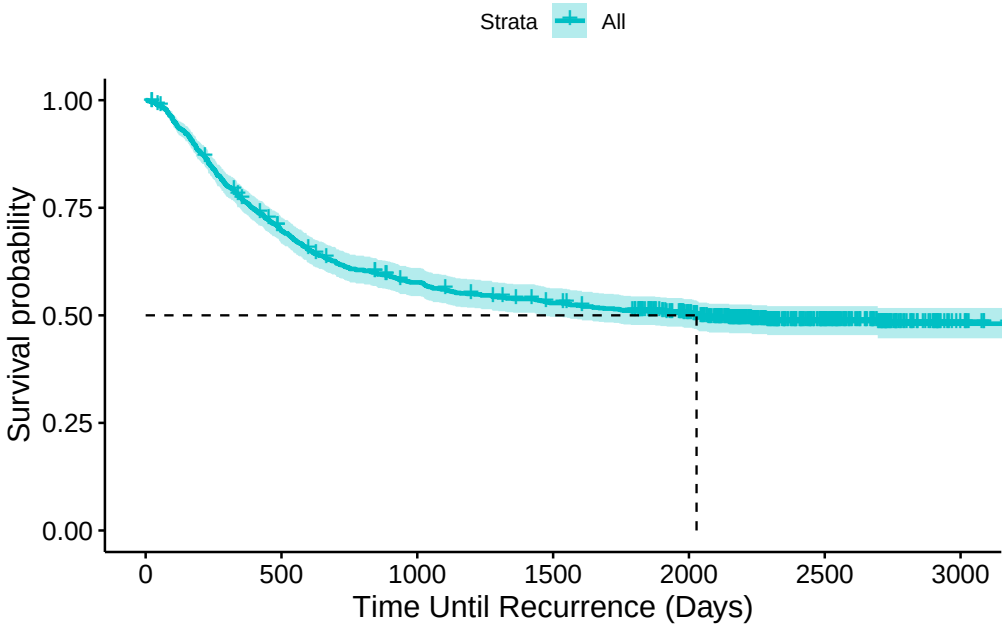
With the clinical context established and the relevant covariates explained, we begin to evaluate the effects of treatment and other factors on the patient. Given that our dataset includes two types of events — recurrence of cancer and death — we begin by modeling these outcomes separately using marginal Cox proportional hazards models. This allows us to estimate the hazard associated with each covariate for each event type independently and by fitting separate Cox models for recurrence and death, we can assess whether specific treatments or patient characteristics are associated with an increased or decreased risk for each type of event.

Marginal Model: Recurrence

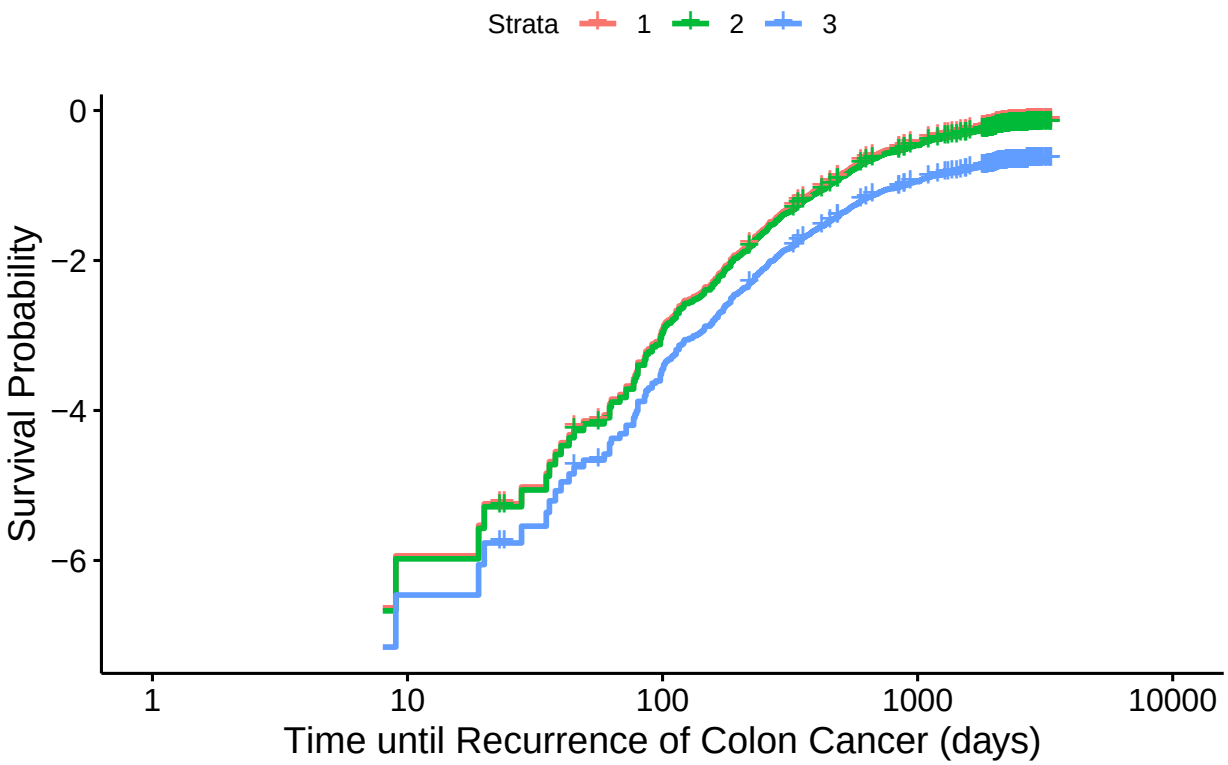
Kaplan-Meier Estimate for Recurrence

```
# data for recurrence event
recurrence_data <- colon[colon$etype == 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,
           xlab = "Time Until Recurrence (Days)",
           palette = "#00BFC4", # Customize the color
           title = "Figure 1: KP for Recurrence Events",
           surv.median.line = 'hv',
           break.time.by=500)
```

Figure 1: KP for Recurrence Events




```
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 of Survival Functions for Different Treatments",
           xlab = "Time until Recurrence of Colon Cancer (days)",
           ylab = "Survival Probability")
```



Exploring Covariate

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 of best fit 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 does not 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:
# Construct a list of covariates to put into the models:
recurrence_covariates = c("obstruct", "adhere", "nodes", "node4", "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 the AIC function

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

```
aic_lvl1 = map_dbl(recurrence_models, AIC) |>
  sort()

aic_lvl1
```

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

# Build 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 the AIC function

recurrence_models2 = map(recurrence_covariates2, \(v)
                         coxph(as.formula(paste("Surv(time, status) ~ rx +
                                                node4 + ", v)),
                               data = recurrence_data)) |>
  set_names(recurrence_covariates2)

aic_lvl2 = map_dbl(recurrence_models2, AIC) |>
  sort()

aic_lvl2
```

```
##   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:
# Construct a 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 the AIC function

recurrence_models3 = map(recurrence_covariates3, \(v)
                         coxph(as.formula(paste(
                           "Surv(time, status) ~ rx + node4 + extent + ", v)),
                           data = recurrence_data)) |>
  set_names(recurrence_covariates3)

aic_lvl3 = map_dbl(recurrence_models3, AIC) |>
  sort()

aic_lvl3
```

```
##     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 the AIC function

recurrence_models4 = map(recurrence_covariates4, \(v)
                         coxph(as.formula(paste("Surv(time, status) ~ rx +
                                                node4 + extent + surg + ", v)),
                               data = recurrence_data)) |>
  set_names(recurrence_covariates4)

aic_lvl4 = map_dbl(recurrence_models4, AIC) |>
  sort()

aic_lvl4
```

```
##   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 the AIC function

recurrence_models5 = map(recurrence_covariates5, \(v)
                         coxph(as.formula(paste("Surv(time, status) ~ rx +
                                                node4 + extent + surg +
                                                differ + ", v)),
                               data = recurrence_data)) |>
  set_names(recurrence_covariates5)

aic_lvl5 = map_dbl(recurrence_models5, AIC) |>
  sort()

aic_lvl5
```

```
## 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 + node4 + extent + surg +
                                                differ + obstruct + ", v)),
                               data = recurrence_data)) |>
  set_names(recurrence_covariates6)

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.

Full Coxph Model for Recurrence

```
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 continguous 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.

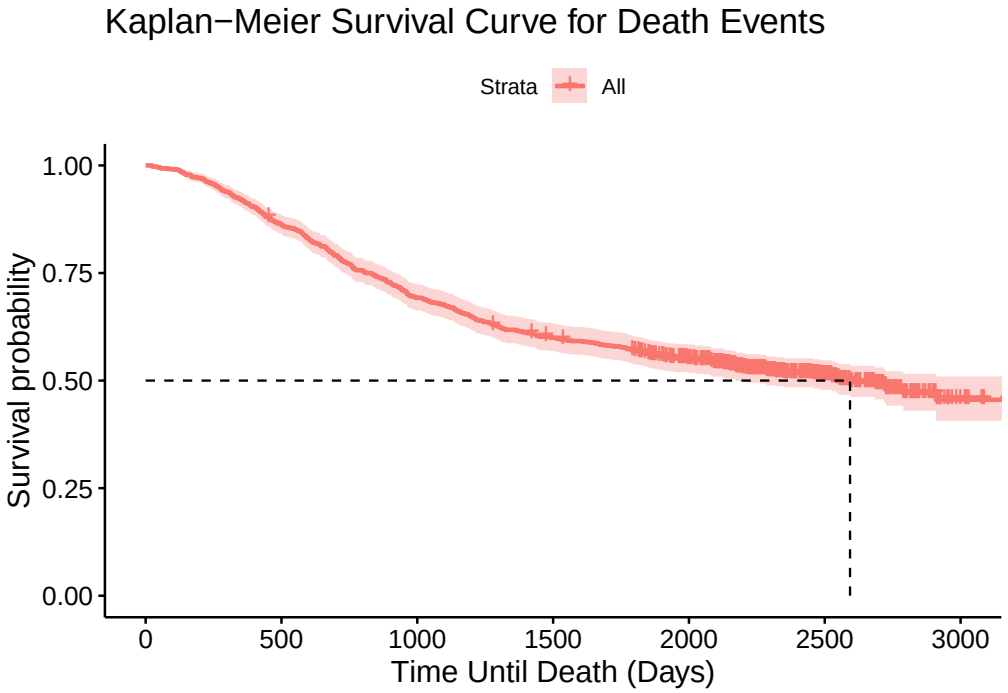
Patients whose colons were obstructed by a tumor had a 23.39% higher hazard rate compared to those who were obstruction-free.

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 is the best fit for the recurrence data.

Marginal Model: Death

Kaplan-Meier Estimate for Death

```
# Data set for death event
# Filter the data for death event
colon_death <- colon_clean[colon_clean$etype == 2, ]

# Fit the Kaplan-Meier model for the death event
km_fit_death <- survfit(Surv(time, status) ~ 1, data = colon_death)

# Plot the Kaplan-Meier curve for death events
ggsurvplot(km_fit_death, data = colon_death,
           title = "Kaplan-Meier Survival Curve for Death Events",
           xlab = "Time Until Death (Days)",
           surv.median.line = 'hv',
           break.time.by = 500)
```



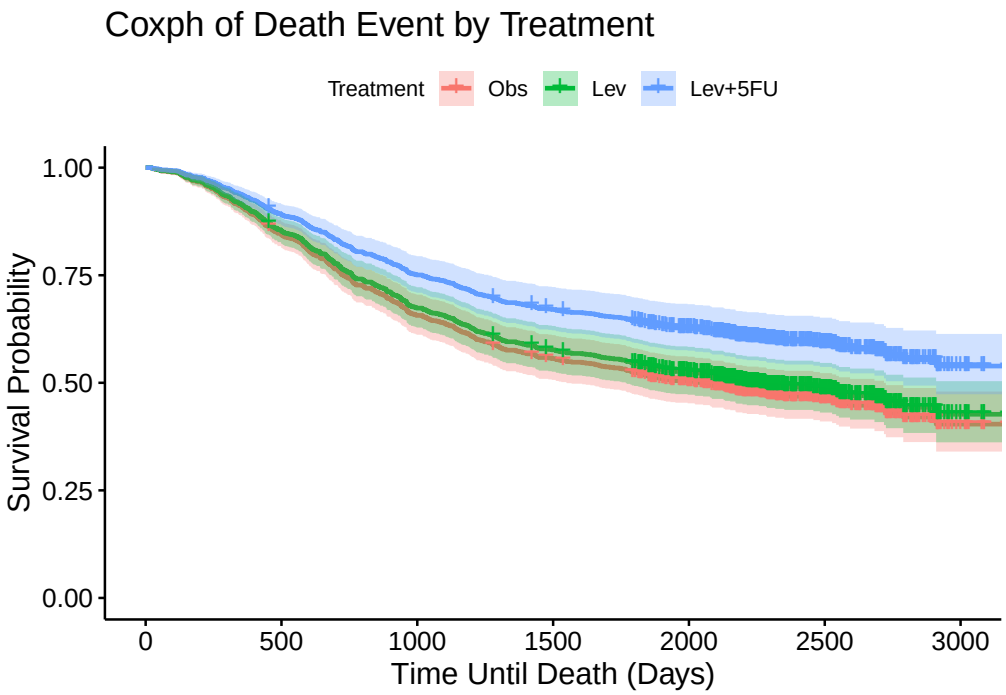
```r
# Median survival time
Death_med <- surv_median(km_fit_death)
print(Death_med)
```

```
##   strata median lower upper
## 1    All   2593  2174    NA
```

The Kaplan-Meier curve declines slowly and almost linearly over the 3000 days follow-up and the median survival time is 2593 days. At the beginning of the study before one year (365 days), the survival probability is roughly above 90%, which indicate that patients in the study begin with a near-perfect chance of remaining alive. Also, the numerous tick marks in the late tail indicate that many individuals were censored alive at the later stage of the study, which is common because it is hard to follow-up for a long period.

Proportional Hazard Model for Death

```r
# Coxph model with rx(treatment) as covariate
cox_death <- coxph(Surv(time, status) ~ rx, data = colon_death)
# Create fit for different treatment
fit_death <- survfit(cox_death, newdata = data.frame(rx = c("Obs", "Lev",
                                                            "Lev+5FU")))

# Plot fit for coxph
ggsurvplot(fit_death, data = colon_death, conf.int = TRUE,
           ylab = "Survival Probability",
           xlab = "Time Until Death (Days)",
           title = "Coxph of Death Event by Treatment",
           legend.title = "Treatment",
           legend.lab = levels(colon_death$rx),
           break.time.by = 500)
```



```r
# median survival time
cox_med <- surv_median(fit_death)
print(cox_med)
```

```
##   strata median lower upper
## 1      1   2052  1550  2718
## 2      2   2257  1767    NA
## 3      3     NA  2789    NA
```

This plot show the survival curves for three treatment after fitting a Cox model with only treatment (`rx`) as the covariate. `Lev+5FU` (blue) curve locate above the other two curves, which might indicate that Levamisole+5-FU (`Lev+5FU`) can increase patients' survival rate. And the its survival probability decrease from 1 and end above 0.5, show that most of patient survive after the study.

`Lev` (green) and `obs` (red) lines do not show much difference. The median survival time for `obs` is 2052 days and for `Lev` is 2257 days greater than `obs`. 95% confidence interval of median survival time for `obs` is (1550, 2718) and the lower bound for confidence interval for `Lev` is 1767. Since the two confidence interval is overlap, there is not statistically significant different between the median survival time of `obs` and `Lev`.

```r
# Summary of PH model
summary(cox_death)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ rx, data = colon_death)
##
##   n= 888, number of events= 430
##
##               coef exp(coef) se(coef)      z Pr(>|z|)
## rxLev     -0.06269   0.93923  0.11319 -0.554  0.57967
## rxLev+5FU -0.38280   0.68195  0.12110 -3.161  0.00157 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## rxLev        0.9392      1.065    0.7524    1.1725
## rxLev+5FU    0.6819      1.466    0.5379    0.8646
##
## Concordance= 0.537  (se = 0.013 )
## Likelihood ratio test= 11.41  on 2 df,   p=0.003
## Wald test            = 10.9  on 2 df,   p=0.004
## Score (logrank) test = 11.02  on 2 df,   p=0.004
```

Since the p-value of likelihood ratio test is 0.003 less than $\alpha = 0.05$, there is sufficient evidence to conclude that `rx` has significant impact on the survival time of patients.

Since hazard ratio for `Lev` is 0.9392, patients on `Lev` has 6.08% lower hazard rate than observation. Also the 95% confidence interval for `Lev` is (0.7524, 1.1725) including 1. Thus, Levamisole does not have significant impact on the survival probability.

The hazard ratio for `Lev+5FU` is 0.6819, `Lev+5FU` has 32% lower hazard rate than `Observation`. Also, 95 confidence interval (0.5379, 0.8646) does not include 1. Treatment `Lev+5FU` significantly lower the hazard rate and increase the survival probability of patient.

Exploring Covariate

AIC

```
# 1st Covariate
# List of Covariate to test
uni_vars <- c("obstruct", "adhere", "nodes", "node4", "differ",
              "extent", "surg", "perfor", "age", "sex")

## 2.  Build one model per variable
uni_models <- map(uni_vars, \(v)
  coxph(as.formula(paste("Surv(time, status) ~ rx + ", v)),
        data = colon_death)
) |> set_names(uni_vars)

## 3.  Grab AIC
aic_tbl <- map_dbl(uni_models, AIC) |>
           sort() |>
           round(2)

# 1st = node4
aic_tbl
```

```
##    node4    nodes   extent   differ   adhere obstruct     surg      age
##  5446.81  5460.49  5507.84  5519.59  5525.62  5526.58  5527.03  5529.74
##   perfor      sex
##  5530.06  5530.38
```

Since `node4` has smallest AIC, `node4` will be first covariate.

```
# 2nd Covariate
uni_vars2 <- c("obstruct", "adhere", "differ",
              "extent", "surg", "perfor", "age", "sex")

uni_models2 <- map(uni_vars2, \(v)
  coxph(as.formula(paste("Surv(time, status) ~ rx + node4 + ", v)),
        data = colon_death)
) |> set_names(uni_vars2)

aic_tbl2 <- map_dbl(uni_models2, AIC) |>
           sort() |>
           round(2)

# 2nd = extent
aic_tbl2
```

```
##   extent   differ obstruct   adhere     surg      age   perfor      sex
##  5432.64  5443.44  5443.53  5444.36  5444.46  5445.52  5448.34  5448.79
```

`extent` has smallest AIC and will be second covariate.

```
# 3rd Covariate
uni_vars3 <- c("obstruct", "adhere", "differ",
               "surg", "perfor", "age", "sex")

uni_models3 <- map(uni_vars3, \(v)
  coxph(as.formula(paste("Surv(time, status) ~ rx + node4 + extent + ", v)),
        data = colon_death)
) |> set_names(uni_vars3)

aic_tbl3 <- map_dbl(uni_models3, AIC) |>
           sort() |>
           round(2)

# 3rd  = surg
aic_tbl3
```

```
##     surg   differ obstruct      age   adhere   perfor      sex
##  5429.86  5430.15  5431.04  5431.17  5431.66  5434.44  5434.61
```

surg with smallest AIC will be third covariate.

```
# 4th Covariate
uni_vars4 <- c("obstruct", "adhere", "differ",
               "perfor", "age", "sex")

uni_models4 <- map(uni_vars4, \(v)
  coxph(as.formula(paste("Surv(time, status) ~ rx + node4 + extent + surg + ", v)),
        data = colon_death)
) |> set_names(uni_vars4)

aic_tbl4 <- map_dbl(uni_models4, AIC) |>
           sort() |>
           round(2)

# 4th = differ
aic_tbl4
```

```
##   differ obstruct      age   adhere   perfor      sex
##  5427.51  5428.30  5428.62  5429.07  5431.68  5431.85
```

Differ has smallest AIC and will be fourth covariate.

```
# 5th Covariate
uni_vars5 <- c("obstruct", "adhere",
               "perfor", "age", "sex")

uni_models5 <- map(uni_vars5, \(v)
  coxph(as.formula(paste("Surv(time, status) ~ rx + node4 + extent + surg +
                         differ + ", v)),
        data = colon_death)
) |> set_names(uni_vars5)
```

```r
aic_tbl5 <- map_dbl(uni_models5, AIC) |>
            sort() |>
            round(2)

aic_tbl5
```

```
## obstruct      age   adhere   perfor      sex
##  5425.77  5426.61  5427.38  5429.36  5429.50
```

We selected extra covariates by forward AIC while always keeping treatment (`rx`) in the model. Adding `node4`, `extent`, and `surg` each cut AIC by > 2 points, and `differ` lowered it by another 2.4; `obstruct` reduced AIC by < 2. Because 2 points is the standard threshold for a meaningful gain, we stopped at `rx + node4 + extent + surg + differ`. This captures nearly all improvement in fit without adding unnecessary parameters.

Full Coxph Model for Death

```r
full_death <- coxph(Surv(time, status) ~ node4 + extent +
                      surg + differ + rx, data = colon_death)
summary(full_death)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ node4 + extent + surg +
##     differ + rx, data = colon_death)
##
##   n= 888, number of events= 430
##
##                     coef exp(coef) se(coef)      z Pr(>|z|)
## node41           0.90682   2.47645  0.10076  9.000  < 2e-16 ***
## extent2          0.58647   1.79764  0.60331  0.972  0.33100
## extent3          1.10438   3.01735  0.58214  1.897  0.05781 .
## extent4          1.56152   4.76605  0.61527  2.538  0.01115 *
## surglong         0.23256   1.26182  0.10625  2.189  0.02862 *
## differmoderate -0.08838   0.91541  0.16812 -0.526  0.59910
## differpoor       0.23602   1.26620  0.19489  1.211  0.22587
## rxLev           -0.04548   0.95554  0.11429 -0.398  0.69066
## rxLev+5FU       -0.37257   0.68896  0.12185 -3.058  0.00223 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                 exp(coef) exp(-coef) lower .95 upper .95
## node41             2.4764     0.4038    2.0326    3.0172
## extent2            1.7976     0.5563    0.5510    5.8646
## extent3            3.0173     0.3314    0.9641    9.4437
## extent4            4.7661     0.2098    1.4271   15.9176
## surglong           1.2618     0.7925    1.0246    1.5540
## differmoderate     0.9154     1.0924    0.6584    1.2727
## differpoor         1.2662     0.7898    0.8642    1.8552
## rxLev              0.9555     1.0465    0.7638    1.1954
## rxLev+5FU          0.6890     1.4515    0.5426    0.8748
```

```
##
## Concordance= 0.662  (se = 0.013 )
## Likelihood ratio test= 126.3  on 9 df,   p=<2e-16
## Wald test            = 129.1  on 9 df,   p=<2e-16
## Score (logrank) test = 139.3  on 9 df,   p=<2e-16
```

After adjusting for the four strongest prognostic factors—`node4`, `extent`, `surg`, and `differ`—the overall likelihood-ratio test is highly significant ($p < 2Ö10^{-16}$), confirming that the set of covariates is statistically significant to explain variation in survival model. From the summary of the cox proptional model, we can observe the following effect of treatment and prognostic covariate:

Treatment effect:
The combination therapy `Levamisole+5-FU` has statistically significant survival benefit, reducing the hazard of death by approximately 31% with (HR = 0.689, 95% CI 0.54˘0.88). `Levamisole` alone does not show significant benefit because the 95% CI (0.7638 – 1.1954) include 1.

Prognostic covariates:
`node4`: having more than 4 positive lymph nodes has hazard ratio of 2.4764 and significantly increase the hazard risk by 147% compared to less than 4 lymph nodes (95% CI 2.03˘3.02).
`extent`: Contiguous structures of local spread (`extent` = 4) raises the hazard by 377% compared to to submucosa of local spread (`extent`=1) (HR = 4.77, 95 % CI 1.96–15.9).
`surg`: Long time from surgery to registration (`surg` = 1) also raise the hazard rate by 26.18% compared to shorter time (`surg`= 0) (HR = 1.26, 95% CI 1.02˘1.55).

Counting Process Model

The results from the marginal Cox proportional hazards models revealed that the combination treatment of Levamisole and 5-Fluorouracil (`Lev+5FU`) was significantly associated with reduced hazard for both recurrence and death. Additionally, time from initial surgery(`surg`), level of tumor spread(`extent`), and whether or not patients had more than four positive lymph nodes(`node4`) were important covariates associated with recurrence and death.

The data is looking at the time between recurrences of colon cancer and death. As a result, it cannot be treated as independent intervals like the gap model because there is a relationship between the recurrence of colon cancer and death from colon cancer. Thus, the counting process model is the best model to use.

```
# Preparing the split
colon$start = 0
colon$stop = colon$time

# Builds the start and stop covariates for the subset of the colon cancer data where only the recurrenc
recurrence_data = recurrence_data %>%
  mutate(start = 0,
         stop = time)
# Established the subset of the colon cancer data where only the deaths from colon cancer are noted

death_data = colon[colon$etype == 2,]

# Modifying the death subset to join the recurrence subset by id. The mutation is so the end time for t

death_data = death_data %>%
  left_join(recurrence_data %>% select(id, recurrence_time = time), by = "id") %>%
  mutate(start = recurrence_time,
         stop = time)
```

```
# The two subsets are then combined by the id covariate from earlier. They are then arranged in order o

colon_counting = bind_rows(recurrence_data, death_data) %>%
  arrange(id) %>%
  select(-recurrence_time) %>%
  mutate(start = if_else(start == stop, 0, start))
```

Similar to the AIC process for the marginal model for the recurrence data, the study covariate will not be included. However, the *id* covariate will be included, but as a cluster in order to compress multiple *id* rows that have the same *id* into one subject/*id*.

```
# Level 1:
# list of covariates to put into the models
colon_covariates = c("obstruct", "adhere", "nodes", "node4", "differ", "extent", "surg", "perfor", "sex

# 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

colon_models = map(colon_covariates, \(v) coxph(as.formula(paste("Surv(start, stop, status) ~ rx + clus

colon_aic_lvl1 = map_dbl(colon_models, AIC) |>
  sort()

colon_aic_lvl1
```

```
##    node4    nodes   extent   differ   adhere     surg obstruct   perfor
## 12078.67 12096.23 12181.91 12226.09 12229.57 12231.17 12234.77 12239.16
##      sex      age
## 12240.89 12241.37
```

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
colon_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

colon_models2 = map(colon_covariates2, \(v) coxph(as.formula(paste("Surv(start, stop, status) ~ rx + clu

colon_aic_lvl2 = map_dbl(colon_models2, AIC) |>
  sort()

colon_aic_lvl2
```

```
##   extent     surg   adhere obstruct   differ   perfor      age      sex
## 12034.65 12068.37 12069.54 12071.37 12073.02 12077.78 12080.24 12080.31
```

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:
# Construct a list of covariates to put into the models:
colon_covariates3 = c("obstruct", "adhere", "differ", "surg", "perfor", "sex", "age")

# Build a model per covariate by pasting the given covariate into the formula.

# The set_names function helps to clarify up which AIC value corresponds to which model when performing

colon_models3 = map(colon_covariates3, \(v) coxph(as.formula(paste("Surv(start, stop, status) ~ rx + cl

colon_aic_lvl3 = map_dbl(colon_models3, AIC) |>
  sort()

colon_aic_lvl3
```

```
##     surg   adhere obstruct   differ   perfor      age      sex
## 12022.98 12029.86 12031.04 12031.85 12035.00 12036.10 12036.46
```

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:
# Construct a list of covariates to put into the models:
colon_covariates4 = c("obstruct", "adhere", "differ",  "perfor", "sex", "age")

# Build a model per covariate by pasting the given covariate into the formula.

# The set_names function helps to clarify up which AIC value corresponds to which model when performing

colon_models4 = map(colon_covariates4, \(v) coxph(as.formula(paste("Surv(start, stop, status) ~ rx + cl

colon_aic_lvl4 = map_dbl(colon_models4, AIC) |>
  sort()

colon_aic_lvl4
```

```
##   adhere obstruct   differ   perfor      age      sex
## 12018.49 12019.92 12020.32 12023.43 12024.58 12024.71
```

The model with the `adhere` covariate, the binary variable of whether the cancer had adhered to other organs, had the lowest AIC.

Therefore, forward selection proceeds with the above covariate.

```
# Level 5:
# Construct a list of covariates to put into the models:
colon_covariates5 = c("obstruct", "differ",  "perfor", "sex", "age")

# Build a model per covariate by pasting the given covariate into the formula.

# The set_names function helps to clarify up which AIC value corresponds to which model when performing

colon_models5 = map(colon_covariates5, \(v) coxph(as.formula(paste("Surv(start, stop, status) ~ rx + clu

colon_aic_lvl5 = map_dbl(colon_models5, AIC) |>
  sort()

colon_aic_lvl5
```

```
## obstruct   differ   perfor      sex      age
## 12015.16 12016.94 12019.68 12020.25 12020.35
```

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:
# Construct a list of covariates to put into the models:
colon_covariates6 = c("differ",  "perfor", "sex", "age")

# Build a model per covariate by pasting the given covariate into the formula.

# The set_names function helps to clarify up which AIC value corresponds to which model when performing

colon_models6 = map(colon_covariates6, \(v) coxph(as.formula(paste("Surv(start, stop, status) ~ rx + clu

colon_aic_lvl6 = map_dbl(colon_models6, AIC) |>
  sort()

colon_aic_lvl6
```

```
##   differ      age   perfor      sex
## 12013.16 12016.74 12016.74 12017.06
```

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 7:
# Construct a list of covariates to put into the models:
colon_covariates7 = c("perfor", "sex", "age")

# Build a model per covariate by pasting the given covariate into the formula.

# The set_names function helps to clarify up which AIC value corresponds to which model when performing
```

```
colon_models7 = map(colon_covariates7, \(v) coxph(as.formula(paste("Surv(start, stop, status) ~ rx + cl

colon_aic_lvl7 = map_dbl(colon_models7, AIC) |>
  sort()

colon_aic_lvl7
```

```
##   perfor      age      sex
## 12014.80 12014.83 12014.97
```

None of the models at this level have an AIC lower than the previous model, so the counting process model after AIC criterion has the following covariates, besides treatment and id: `node4`, `extent`, `surg`, `adhere`, `obstruct`, 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(start, stop, status) ~ rx + cluster(id) + node4 + extent + surg + adhere + obstruct +
```

```
## Call:
## coxph(formula = Surv(start, stop, status) ~ rx + node4 + extent +
##     surg + adhere + obstruct + differ, data = recurrence_data,
##     cluster = id)
##
##   n= 888, number of events= 446
##
##                coef exp(coef)  se(coef) robust se      z Pr(>|z|)
## rxLev     -0.009694  0.990353  0.110379  0.112620 -0.086 0.931405
## rxLev+5FU -0.491428  0.611752  0.121668  0.125261 -3.923 8.74e-05 ***
## node4      0.837207  2.309906  0.099433  0.103383  8.098 5.58e-16 ***
## extent     0.479950  1.615993  0.119031  0.126425  3.796 0.000147 ***
## surg       0.225565  1.253031  0.103859  0.106033  2.127 0.033394 *
## adhere     0.178029  1.194860  0.127011  0.127578  1.395 0.162879
## obstruct   0.213281  1.237733  0.117668  0.126294  1.689 0.091264 .
## differ     0.167086  1.181855  0.097400  0.105629  1.582 0.113690
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##           exp(coef) exp(-coef) lower .95 upper .95
## rxLev        0.9904     1.0097    0.7942     1.235
## rxLev+5FU    0.6118     1.6346    0.4786     0.782
## node4        2.3099     0.4329    1.8862     2.829
## extent       1.6160     0.6188    1.2613     2.070
## surg         1.2530     0.7981    1.0179     1.542
## adhere       1.1949     0.8369    0.9305     1.534
## obstruct     1.2377     0.8079    0.9663     1.585
## differ       1.1819     0.8461    0.9608     1.454
##
## Concordance= 0.661  (se = 0.013 )
## Likelihood ratio test= 128.2  on 8 df,   p=<2e-16
## Wald test            = 125  on 8 df,   p=<2e-16
## Score (logrank) test = 134.6  on 8 df,   p=<2e-16,   Robust = 109.1  p=<2e-16
```

```
##
##   (Note: the likelihood ratio and score tests assume independence of
##      observations within a cluster, the Wald and robust score tests do not).
```

From the likelihood ratio test, the p-value is less than $2e-16$, which is much less than $\alpha = 0.05$.

The hazard rate for patients who took the treatment with just Levamisole is 2.208% less hazardous than taking no treatment at all. Those who took Fluoracil in addition to Levamisole benefited with a hazard ratio of 0.6062, 39.38% 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 3.3764 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 3.86% lower hazard rate than patients whose cancer cells were "well differentiated", while those with "poorly differentiated" cells had 28.41% higher hazard rate compared to same base group.

Patients whose colons were obstructed by a tumor had a 23.37% 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 either recurrence or death from 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 is the best fit for the data as a whole.

Proptional Hazards Assumption

To verify that the Cox models we use are valid, we assessed the proportional-hazards (PH) assumption. Our checking strategy combined (i) graphical inspection and (ii) formal statistical testing.

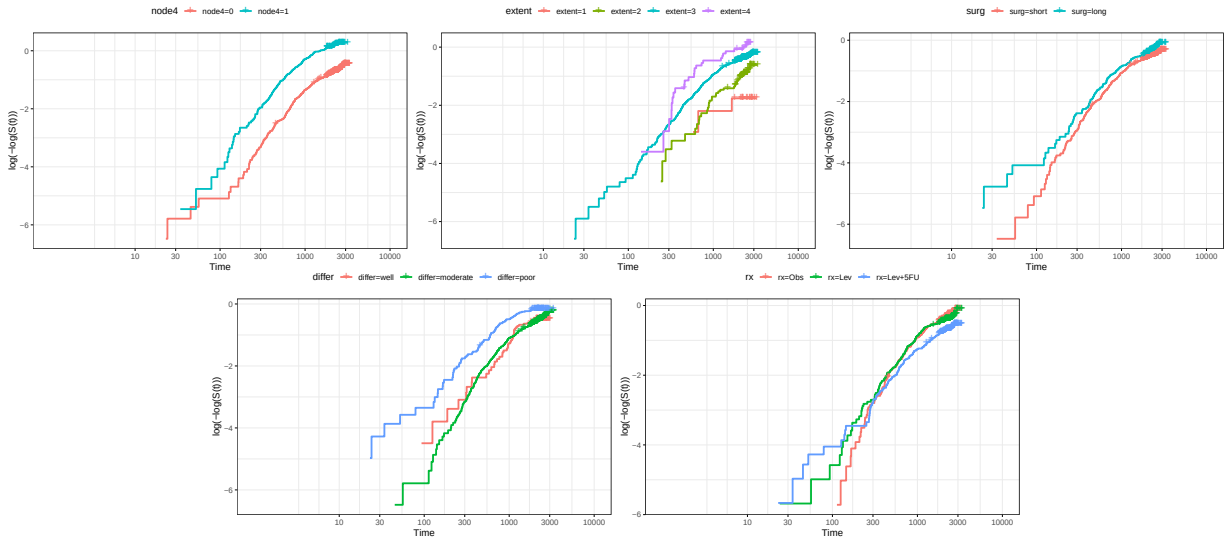
Recurrence Event

Death Event

Log-log survival plots

Under the proportional-hazard (PH) assumption the category-specific curves should be roughly straight and parallel; systematic divergence, covergence, or corssing suggests non-PH behavior.

```
for (v in c("node4", "extent", "surg", "differ", "rx")) {
  fit  <- survfit(as.formula(paste("Surv(time,status) ~", v)), data = colon_death)
  print(ggsurvplot(fit, data = colon_death, fun = "cloglog",
                   legend.title = v, ggtheme = theme_bw()))
}
```



- *node4*: Two curves seems parallel so *node4* satisfy the assumption.
- *extent*: At the early stage the blue curve cross the purple curve and the green curve cross the red curve. After that, the purple, blue, and green curves are relatively parallel but the red curve seems to deviate from them. Thus, the assumption is concerning.
- *surg*: Two curve never cross and remain parallel over time. So, *surg* satisfy the assumption.
- *differ*: The green curve cross the other two curves four times over time. And the gap between each curve become smaller over time. Thus, *differ* violate the assumption.
- *rx*: Three curve cross each other at the early stage but with few observations. And the green and red curve are closed to each other but the gap between each curve is relatively parallel. It is hard to make conclusion by looking at the log-log plot so we will use the statistical testing to test assumption.

Cox ZPH

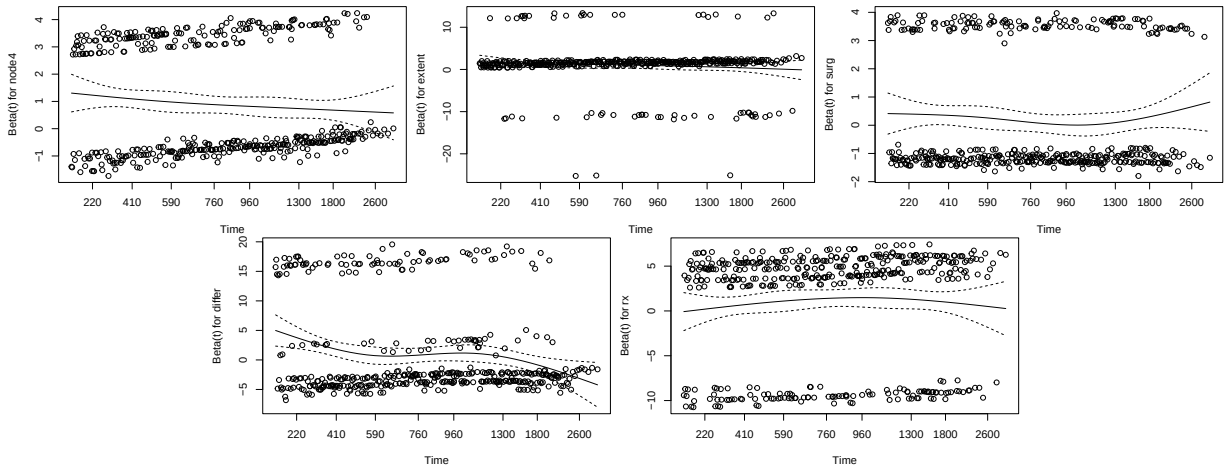
```
ph <- cox.zph(full_death)
print(ph)
```

```
##          chisq df       p
## node4   5.4346  1 0.01974
## extent  6.5521  3 0.08763
## surg    0.0205  1 0.88616
## differ 15.8513  2 0.00036
## rx      2.5680  2 0.27693
## GLOBAL 27.5832  9 0.00112
```

- *node4*: Since p-value is 0.01974 less than $\alpha = 0.05$, *node4* violate the assumption. This conclusion is different from the conclusion made from log-log plot.
- *extent*: Since p-value is 0.08763 greater than $\alpha = 0.05$, it does not violate the assumption. Although there is no sufficient evidence of non-PH, it is concerning about the assumption.
- *surg*: p-value is large so it does not violate the assumption.

- *differ*: Since p-value is 0.00036 smaller than $\alpha = 0.05$, it violate the assumption.
- *rx*: p-value is large so it does not violate the assumption.

```
plot(ph)
```



- *node4*
  - Smoothed $\beta(t)$ line is almost flat except for a slight downward bend late in follow-up.
  - **Decision**: keep node4 in the model; add a mild log-time interaction (node4 $\times$ log t)) so its tiny time-trend is absorbed while preserving a clinically useful hazard-ratio.
- *differ*
  - Smoothed $\beta(t)$ line declines steadily and have a clear monotone drift.
  - **Decision**: stratify on differ ('strata(differ)'). This gives each grade its own baseline hazard, removes the PH requirement, and avoids reporting a misleading, time-dependent HR for grade—acceptable because grade is a control variable, not a primary target of inference.

```
full_death2 <- coxph(Surv(time, status) ~ node4 + extent +
                       surg + strata(differ) + rx, data = colon_death)
# summary(full_death2)
cox.zph(full_death2)
```

```
##           chisq df     p
## node4   3.77162  1 0.052
## extent  5.89794  3 0.117
## surg    0.00474  1 0.945
## rx      2.46820  2 0.291
## GLOBAL 12.18168  7 0.095
```

```
fit_final <- coxph(Surv(time, status) ~ extent + surg + node4 + (node4) +
                     strata(differ) + rx, data = colon_death,
                   tt = function(x,t,...) {x*log(t)})

cox.zph(fit_final)              # should now give non-significant p-values
```

```
##           chisq df     p
## extent  5.89794  3 0.117
## surg    0.00474  1 0.945
## node4   3.77162  1 0.052
## rx      2.46820  2 0.291
## GLOBAL 12.18168  7 0.095
```

References

1. colon: Chemotherapy for Stage B/C colon cancer. Colon Cancer. Retrieved May 1st, 2025, from https://rdrr.io/cran/survival/man/colon.html

2. CG Moertel, TR Fleming, JS MacDonald, DG Haller, JA Laurie, CM Tangen, JS Ungerleider, WA Emerson, DC Tormey, JH Glick, MH Veeder and JA Maillard, Fluorouracil plus Levamisole as an effective adjuvant therapy after resection of stage II colon carcinoma: a final report. Annals of Internal Med, 122:321-326, 1991.