S24 Survival Analysis

Sourena Mohit

2025-03-14

Step 1: Load and Import the Dataset

In this step, we want to:

1. Ensure the survival package is installed (because it contains the colon dataset).
2. Load the survival package.
3. Load the colon dataset into our R environment.
4. Copy the dataset to a new object (colon_data) so we don't accidentally overwrite the original.

```
# 1. Check if the 'survival' package is installed; if not, install it.
if (!require("survival")) install.packages("survival", dependencies=TRUE)

# 2. Load the 'survival' package into the R session
library(survival)

# 3. Load the 'colon' dataset from the 'survival' package
data("colon")

# 4. Make a copy of the dataset
colon_data <- colon
```

Step 2: Explore the Dataset

In Step 2, our goal is to examine the dataset to understand its contents, identify potential issues, and plan how to handle them. Below is a structured explanation of each part of your code and the corresponding results.

```
# Define a helper function to install packages if they're not already installed
install_if_missing <- function(packages) {
  new_pkgs <- packages[!(packages %in% installed.packages()[, "Package"])]
  if (length(new_pkgs)) install.packages(new_pkgs, dependencies = TRUE)
}

# Specify required packages
packages_needed <- c("skimr", "kableExtra", "dplyr")

# Install missing packages
install_if_missing(packages_needed)
```

```r
# Load the packages
lapply(packages_needed, library, character.only = TRUE)
```

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

## The following object is masked from 'package:kableExtra':
##
##     group_rows

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

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

## [[1]]
## [1] "skimr"     "survival"  "stats"     "graphics"  "grDevices" "utils"
## [7] "datasets"  "methods"   "base"
##
## [[2]]
##  [1] "kableExtra" "skimr"      "survival"   "stats"      "graphics"
##  [6] "grDevices"  "utils"      "datasets"   "methods"    "base"
##
## [[3]]
##  [1] "dplyr"      "kableExtra" "skimr"      "survival"   "stats"
##  [6] "graphics"   "grDevices"  "utils"      "datasets"   "methods"
## [11] "base"
```

```r
# View the first few rows
head(colon_data)
```

```
##   id study      rx sex age obstruct perfor adhere nodes status differ extent
## 1  1     1 Lev+5FU   1  43        0      0      0     5      1      2      3
## 2  1     1 Lev+5FU   1  43        0      0      0     5      1      2      3
## 3  2     1 Lev+5FU   1  63        0      0      0     1      0      2      3
## 4  2     1 Lev+5FU   1  63        0      0      0     1      0      2      3
## 5  3     1     Obs   0  71        0      0      1     7      1      2      2
## 6  3     1     Obs   0  71        0      0      1     7      1      2      2
##   surg node4 time etype
## 1    0     1 1521     2
## 2    0     1  968     1
## 3    0     0 3087     2
## 4    0     0 3087     1
## 5    0     1  963     2
## 6    0     1  542     1
```

```r
# Check the structure of the dataset
str(colon_data)
```

```
## 'data.frame':    1858 obs. of  16 variables:
##  $ id      : num  1 1 2 2 3 3 4 4 5 5 ...
##  $ study   : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ rx      : Factor w/ 3 levels "Obs","Lev","Lev+5FU": 3 3 3 3 1 1 3 3 1 1 ...
##  $ sex     : num  1 1 1 1 0 0 0 0 1 1 ...
##  $ age     : num  43 43 63 63 71 71 66 66 69 69 ...
##  $ obstruct: num  0 0 0 0 0 0 1 1 0 0 ...
##  $ perfor  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ adhere  : num  0 0 0 0 1 1 0 0 0 0 ...
##  $ nodes   : num  5 5 1 1 7 7 6 6 22 22 ...
##  $ status  : num  1 1 0 0 1 1 1 1 1 1 ...
##  $ differ  : num  2 2 2 2 2 2 2 2 2 2 ...
##  $ extent  : num  3 3 3 3 2 2 3 3 3 3 ...
##  $ surg    : num  0 0 0 0 0 0 1 1 1 1 ...
##  $ node4   : num  1 1 0 0 1 1 1 1 1 1 ...
##  $ time    : num  1521 968 3087 3087 963 ...
##  $ etype   : num  2 1 2 1 2 1 2 1 2 1 ...
```

```r
# Get summary statistics
summary(colon_data)
```

```
##        id          study         rx           sex              age       
##  Min.   :  1   Min.   :1   Obs    :630   Min.   :0.000   Min.   :18.00  
##  1st Qu.:233   1st Qu.:1   Lev    :620   1st Qu.:0.000   1st Qu.:53.00  
##  Median :465   Median :1   Lev+5FU:608   Median :1.000   Median :61.00  
##  Mean   :465   Mean   :1                 Mean   :0.521   Mean   :59.75  
##  3rd Qu.:697   3rd Qu.:1                 3rd Qu.:1.000   3rd Qu.:69.00  
##  Max.   :929   Max.   :1                 Max.   :1.000   Max.   :85.00  
##                                                                         
##     obstruct          perfor            adhere           nodes      
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.0000   Min.   : 0.00  
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.: 1.00  
##  Median :0.0000   Median :0.00000   Median :0.0000   Median : 2.00  
##  Mean   :0.1938   Mean   :0.02906   Mean   :0.1453   Mean   : 3.66  
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.: 5.00  
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.0000   Max.   :33.00  
##                                                      NA's   :36     
##      status           differ          extent            surg       
##  Min.   :0.0000   Min.   :1.000   Min.   :1.000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:3.000   1st Qu.:0.0000  
##  Median :0.0000   Median :2.000   Median :3.000   Median :0.0000  
##  Mean   :0.4952   Mean   :2.063   Mean   :2.887   Mean   :0.2659  
##  3rd Qu.:1.0000   3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:1.0000  
##  Max.   :1.0000   Max.   :3.000   Max.   :4.000   Max.   :1.0000  
##                   NA's   :46                                      
##      node4             time          etype    
##  Min.   :0.0000   Min.   :   8   Min.   :1.0  
##  1st Qu.:0.0000   1st Qu.: 566   1st Qu.:1.0  
##  Median :0.0000   Median :1855   Median :1.5  
##  Mean   :0.2745   Mean   :1538   Mean   :1.5  
```

```
##   3rd Qu.:1.0000   3rd Qu.:2331   3rd Qu.:2.0
##   Max.   :1.0000   Max.   :3329   Max.   :2.0
##
```

```r
# Generate summary statistics using skimr
summary_stats <- skim(colon)

# View the summary statistics
print(summary_stats)
```

```
## -- Data Summary ------------------------
##                            Values
## Name                       colon
## Number of rows             1858
## Number of columns          16
## _______________________
## Column type frequency:
##   factor                   1
##   numeric                  15
## ________________________
## Group variables            None
##
## -- Variable type: factor -------------------------------------------------------
##   skim_variable n_missing complete_rate ordered n_unique
## 1 rx                    0             1 FALSE          3
##   top_counts
## 1 Obs: 630, Lev: 620, Lev: 608
##
## -- Variable type: numeric ------------------------------------------------------
##    skim_variable n_missing complete_rate      mean      sd p0 p25    p50  p75
##  1 id                    0         1       465      268.     1 233  465    697
##  2 study                 0         1         1         0     1   1    1      1
##  3 sex                   0         1         0.521     0.500 0   0    1      1
##  4 age                   0         1        59.8      11.9  18  53   61     69
##  5 obstruct              0         1         0.194     0.395 0   0    0      0
##  6 perfor                0         1         0.0291    0.168 0   0    0      0
##  7 adhere                0         1         0.145     0.353 0   0    0      0
##  8 nodes                36         0.981     3.66      3.57  0   1    2      5
##  9 status                0         1         0.495     0.500 0   0    0      1
## 10 differ               46         0.975     2.06      0.514 1   2    2      2
## 11 extent                0         1         2.89      0.488 1   3    3      3
## 12 surg                  0         1         0.266     0.442 0   0    0      1
## 13 node4                 0         1         0.274     0.446 0   0    0      1
## 14 time                  0         1      1538.      947.    8 566 1855   2331
## 15 etype                 0         1         1.5       0.500 1   1    1.5    2
##    p100 hist
##  1  929
##  2    1
##  3    1
##  4   85
##  5    1
##  6    1
##  7    1
##  8   33
```

```
##  9    1
## 10    3
## 11    4
## 12    1
## 13    1
## 14 3329
## 15    2
```

```r
# Convert the summary statistics to a nicely formatted table
summary_table <- summary_stats %>%
  kable("html", caption = "Summary Statistics of the Colon Dataset") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive"))

# Print the table
summary_table
```

Summary Statistics of the Colon Dataset

skim_type

skim_variable

n_missing

complete_rate

factor.ordered

factor.n_unique

factor.top_counts

numeric.mean

numeric.sd

numeric.p0

numeric.p25

numeric.p50

numeric.p75

numeric.p100

numeric.hist

factor

rx

0

1.0000000

FALSE

3

Obs: 630, Lev: 620, Lev: 608

NA

NA

NA

NA

NA

NA

NA

NA

numeric

id

0

1.0000000

NA

NA

NA

465.0000000

268.2512426

1

233

465.0

697

929

numeric

study

0

1.0000000

NA

NA

NA

1.0000000

0.0000000

1

1

1.0

1

1

numeric

sex

0

1.0000000

NA

NA

NA

0.5209903

0.4996937

0

0

1.0

1

1

numeric

age

0

1.0000000

NA

NA

NA

59.7545748

11.9456696

18

53

61.0

69

85

numeric

obstruct

0

1.0000000

NA

NA

NA

0.1937567

0.3953469

0

0

0.0

0

1

numeric

perfor

0

1.0000000

NA

NA

NA

0.0290635

0.1680298

0

0

0.0

0

1

numeric

adhere

0

1.0000000

NA

NA

NA

0.1453175

0.3525156

0

0

0.0

0

1

numeric

nodes

36

0.9806243

NA

NA

NA

3.6597146

3.5715810

0

1

2.0

5

33

numeric

status

0

1.0000000

NA

NA

NA

0.4951561

0.5001111

0

0

0.0

1

1

numeric

differ

46

0.9752422

NA

NA

NA

2.0629139

0.5141981

1

2

2.0

2

3

numeric

extent

0

1.0000000

NA

NA

NA

2.8869752

0.4880173

1

3

3.0

3

4

numeric

surg

0

1.0000000

NA

NA

NA

0.2658773

0.4419182

0

0

0.0

1

1

numeric

node4

0

1.0000000

NA

NA

NA

0.2744887

0.4463764

0

0

0.0

1

1

numeric

time

0

1.0000000

NA

NA

NA

1537.5457481

946.7038105

8

566

1855.0

2331

3329

numeric

etype

0

1.0000000

NA

NA

NA

1.5000000

0.5001346

1

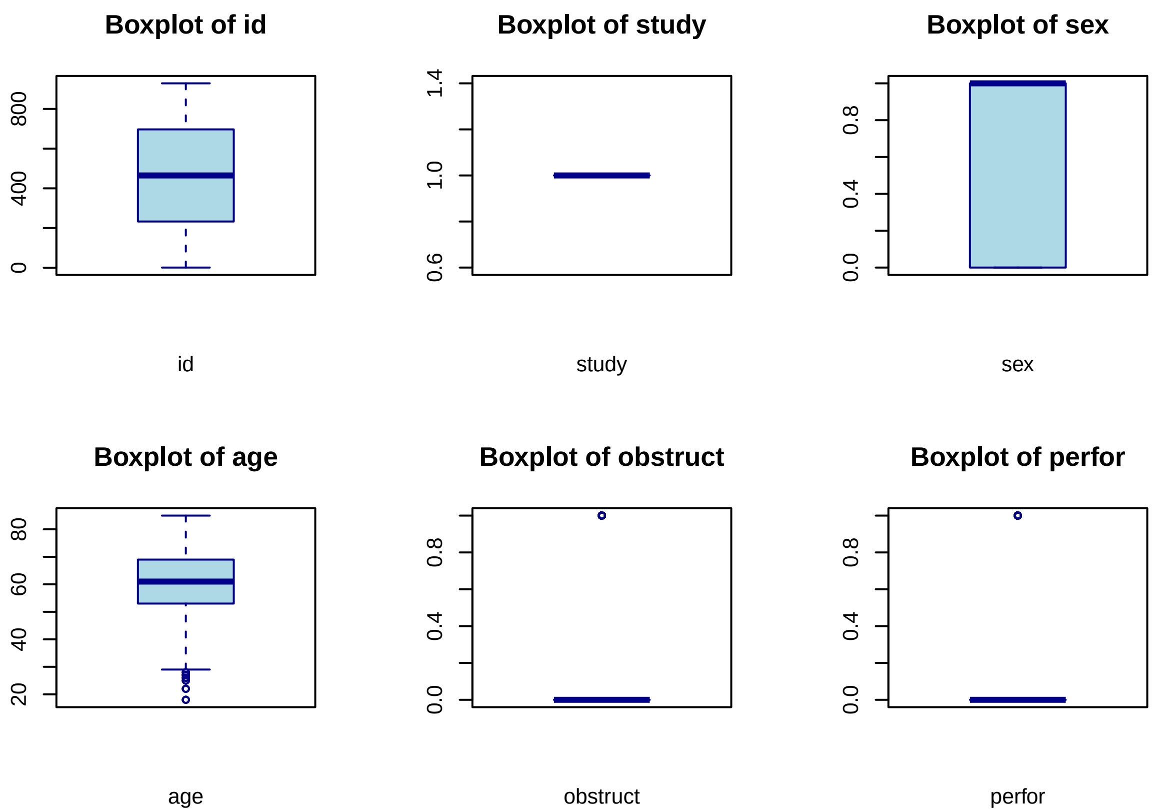
1

1.5

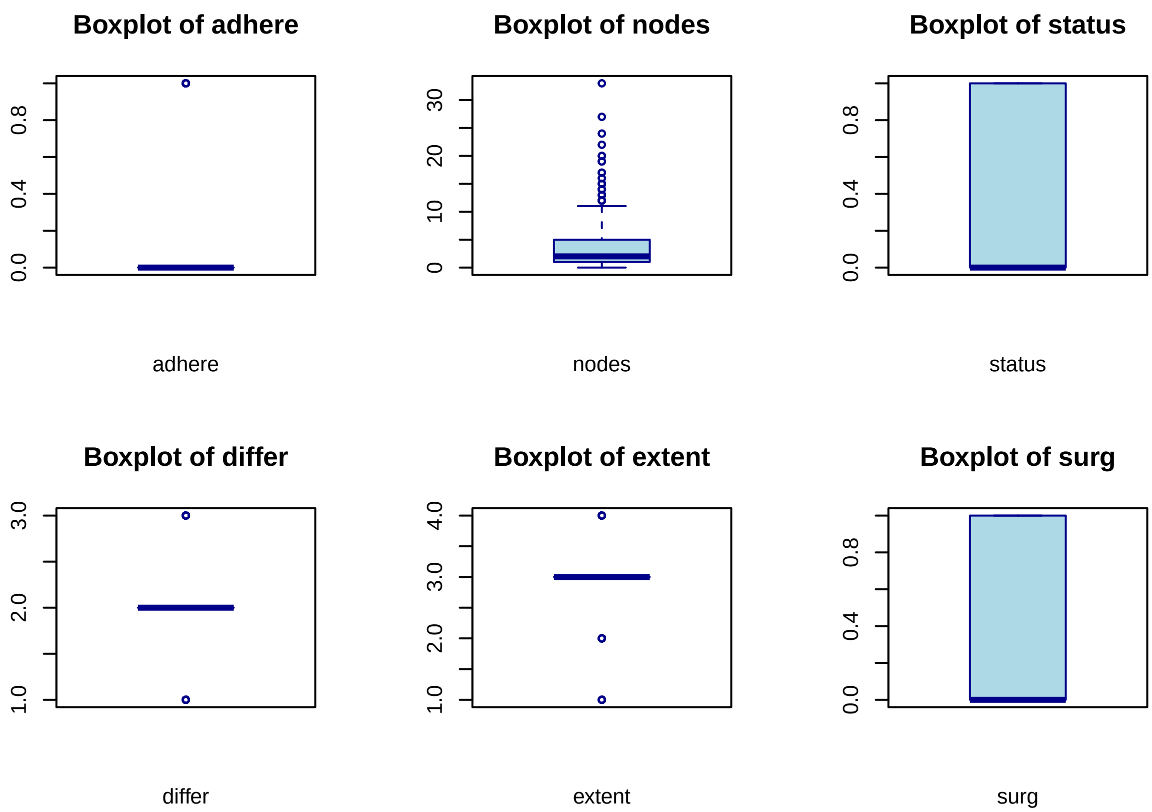
2

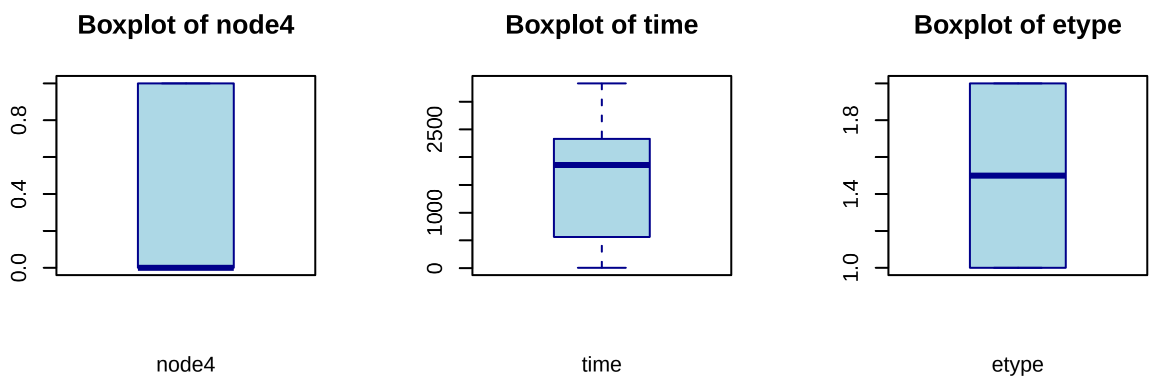
2

```r
# Save the table as an HTML file
save_kable(summary_table, file = "summary_table.html")

# Function to plot boxplots for all numeric columns
plot_boxplots <- function(data) {
  numeric_cols <- sapply(data, is.numeric)
  numeric_data <- data[, numeric_cols, drop = FALSE]

  # Optional: arrange multiple plots in a grid (2 rows, 3 columns)
  par(mfrow = c(2, 3))
  for (col in names(numeric_data)) {
    boxplot(
      numeric_data[[col]],
      main = paste("Boxplot of", col),
      xlab = col,
      col = "lightblue",
      border = "darkblue"
    )
  }
  # Reset the plotting area to a single panel
  par(mfrow = c(1, 1))
}

# Call the function
plot_boxplots(colon_data)
```



Boxplot of id
0
400
800
id
Boxplot of study
0.6
1.0
1.4
study
Boxplot of sex
0.0
0.4
0.8
sex
Boxplot of age
20
40
60
80
age
Boxplot of obstruct
0.0
0.4
0.8
obstruct
Boxplot of perfor
0.0
0.4
0.8
perfor


Boxplot of adhere
0.0
0.4
0.8
adhere
Boxplot of nodes
0
10
20
30
nodes
Boxplot of status
0.0
0.4
0.8
status
Boxplot of differ
1.0
2.0
3.0
differ
Boxplot of extent
1.0
2.0
3.0
4.0
extent
Boxplot of surg
0.0
0.4
0.8
surg




```r
# Check for missing values
colSums(is.na(colon_data))
```

```
##       id    study       rx      sex      age obstruct   perfor   adhere
##        0        0        0        0        0        0        0        0
##    nodes   status   differ   extent     surg    node4     time    etype
##       36        0       46        0        0        0        0        0
```

Analysis of the Exploratory Results

Below is a structured analysis of your Step 2 exploratory findings. walk through the key takeaways from the summary statistics, missing values check, and boxplots.

1. Data Structure Recap

- **Observations**: 1,858 rows (each row can represent a patient event record).
- **Variables**: 16 columns (numeric and factor types).

Key Variables:

- **time**: Time in days until an event (recurrence or death) or censoring.
- **status**: Event occurrence indicator (1 = event, 0 = censored).
- **etype**: Event type (1 = recurrence, 2 = death).

- **rx**: Treatment group (3 levels: "Obs", "Lev", "Lev+5FU").
- **nodes**: Number of positive lymph nodes (0–33).
- **differ**: Tumor differentiation (1 = well, 2 = moderate, 3 = poor).
- **sex**: Encoded as 0 (female) and 1 (male), though numeric in the dataset.
- **id**: Patient identifier, which is not directly useful in modeling.

Understanding each variable's purpose helps in deciding how to handle them during modeling (e.g., which to include as covariates).

2. Missing Values

- **nodes**: 36 missing values.
- **differ**: 46 missing values.
- All other variables appear complete.

Missing data can bias results if not handled properly.

3. Numeric Variable Distributions and Outliers

From the boxplots: - **age**: Ranges from 18 to 85; a few younger outliers (below 30). - **nodes**: Heavily right-skewed. Some patients have a high count (above 10), and the maximum is 33. - **time**: Spans 8 to 3329 days (over 9 years). The median is around 1855 days (~5 years). - **sex, obstruct, perfor, adhere, status, surg, node4, etype**: Each is effectively binary or categorical (0/1 or 1/2), so boxplots show minimal variation. - **differ (1 to 3) and extent (1 to 4)**: Appear mostly around 2–3, with a few outliers at 1 and 4.

Right-skewed data (like nodes and time) may benefit from transformations (e.g., log-transform) if needed for certain models. Outliers might represent genuine clinical scenarios, boxplots confirm the presence of two distinct categories.

4. Factor Variables

- **rx**: Three treatment groups ("Obs", "Lev", "Lev+5FU").
- **study**: Always 1 (no variation, so it's not useful for analysis).

rx is already a factor, but sex might be converted to a factor to make it more intuitive (0 = Female, 1 = Male). Factor variables are crucial in survival models (like Cox regression) to compare hazard rates among groups. Unvarying variables (like study) do not contribute any information to the model.

5. Overall Observations

- **No Large-Scale Missingness**: Only nodes (36 NAs) and differ (46 NAs) need attention.
- **Potential Outliers**: nodes (high node counts), age (younger ages), and time (very short or very long survival).
- **Mostly Categorical/Binary Data**: Many columns are 0/1 or 1/2, which is common in medical datasets (presence/absence, yes/no).
- **Two Records per Patient**: Remember that each patient has two rows (one for recurrence, one for death), which will be important in modeling.

Step 3: Data Preprocessing and Survival Setup

3.1 Convert Data Types

convert binary/categorical variables (sex, obstruct, perfor, adhere, node4, surg, differ, extent) into factors with meaningful labels. This makes the data more readable and suitable for modeling.

```r
# Convert sex: 0 = Female, 1 = Male
colon_data$sex <- factor(colon_data$sex,
                         levels = c(0, 1),
                         labels = c("Female", "Male"))

# Convert obstruct, perfor, adhere: 0 = No, 1 = Yes
colon_data$obstruct <- factor(colon_data$obstruct,
                              levels = c(0, 1),
                              labels = c("No", "Yes"))
colon_data$perfor <- factor(colon_data$perfor,
                            levels = c(0, 1),
                            labels = c("No", "Yes"))
colon_data$adhere <- factor(colon_data$adhere,
                            levels = c(0, 1),
                            labels = c("No", "Yes"))

# Convert node4: 0 = <=4 positive nodes, 1 = >4 positive nodes
colon_data$node4 <- factor(colon_data$node4,
                           levels = c(0, 1),
                           labels = c("<=4", ">4"))

# Convert surg: 0 = Short, 1 = Long
colon_data$surg <- factor(colon_data$surg,
                          levels = c(0, 1),
                          labels = c("Short", "Long"))

# Convert differ: 1 = Well, 2 = Moderate, 3 = Poor
colon_data$differ <- factor(colon_data$differ,
                            levels = c(1, 2, 3),
                            labels = c("Well", "Moderate", "Poor"))

# Convert extent: 1 = Submucosa, 2 = Muscle, 3 = Serosa, 4 = Contiguous structures
colon_data$extent <- factor(colon_data$extent,
                            levels = c(1, 2, 3, 4),
                            labels = c("Submucosa", "Muscle", "Serosa", "Contiguous"))
```

3.2 Handle Missing Data (Multiple Imputation)

use the mice package to impute missing values rather than eliminating rows. This is a more robust approach when dealing with relatively small amounts of missing data.

```r
### 3.2 Handle Missing Data

# 1. Install/load the 'mice' package if not present
if (!require("mice")) install.packages("mice", dependencies = TRUE)
library(mice)
```

```r
# 2. Select columns relevant for imputation
#    (Include the variables with missing data + additional predictors)
impute_vars <- c("nodes", "differ", "age", "sex", "obstruct",
                 "perfor", "adhere", "extent", "node4", "time", "status")

impute_data <- colon_data[, impute_vars]

# 3. Check default methods mice will use
#    For factor variables, 'mice' often uses logistic or polyreg;
#    for numeric, it uses pmm (Predictive Mean Matching) by default.
methods <- make.method(impute_data)
methods
```

```
##     nodes    differ       age       sex  obstruct    perfor    adhere    extent
##     "pmm" "polyreg"        ""        ""        ""        ""        ""        ""
##     node4      time    status
##        ""        ""        ""
```

```r
# (Optional) You can manually specify methods, e.g.:
# methods["differ"] <- "polyreg"   # for a factor with >2 levels
# methods["nodes"]  <- "pmm"       # for numeric

# 4. Run multiple imputation
imp <- mice(data = impute_data,
            m = 5,             # number of imputed datasets
            method = methods, # or your customized methods
            maxit = 5,         # number of iterations
            seed = 123)        # for reproducibility
```

```
##
##  iter imp variable
##   1   1  nodes  differ
##   1   2  nodes  differ
##   1   3  nodes  differ
##   1   4  nodes  differ
##   1   5  nodes  differ
##   2   1  nodes  differ
##   2   2  nodes  differ
##   2   3  nodes  differ
##   2   4  nodes  differ
##   2   5  nodes  differ
##   3   1  nodes  differ
##   3   2  nodes  differ
##   3   3  nodes  differ
##   3   4  nodes  differ
##   3   5  nodes  differ
##   4   1  nodes  differ
##   4   2  nodes  differ
##   4   3  nodes  differ
##   4   4  nodes  differ
##   4   5  nodes  differ
##   5   1  nodes  differ
```

```
##   5   2  nodes  differ
##   5   3  nodes  differ
##   5   4  nodes  differ
##   5   5  nodes  differ
```

```
# 5. Choose one of the imputed sets (1 to 5) to create a completed dataset
completed_data <- complete(imp, 1)

# 6. Assign the imputed columns back to your main dataset
colon_data$nodes   <- completed_data$nodes
colon_data$differ  <- completed_data$differ
```

3.3 Outlier & Skewness Handling

nodes is heavily skewed. We can log-transform it if we want to treat it as a continuous covariate.

```
### 3.3 Outlier & Skewness Handling

# Create a log-transformed version of 'nodes'
# Adding +1 avoids log(0)
colon_data$log_nodes <- log(colon_data$nodes + 1)


# Function to return a list with lower and upper bounds
get_iqr_bounds <- function(x) {
  Q1 <- quantile(x, 0.25, na.rm = TRUE)
  Q3 <- quantile(x, 0.75, na.rm = TRUE)
  IQR_val <- Q3 - Q1
  lower_bound <- Q1 - 1.5 * IQR_val
  upper_bound <- Q3 + 1.5 * IQR_val
  return(list(lower = lower_bound, upper = upper_bound))
}

# Let's apply it to 'age', 'time', and 'nodes'
vars_to_check <- c("age","nodes")
bounds_list <- lapply(vars_to_check, function(v) {
  get_iqr_bounds(colon_data[[v]])
})
names(bounds_list) <- vars_to_check

bounds_list
```

```
## $age
## $age$lower
## 25%
##  29
##
## $age$upper
## 75%
##  93
##
##
```

```
## $nodes
## $nodes$lower
## 25%
##  -5
##
## $nodes$upper
## 75%
##  11
```

```r
# This will print the lower/upper bound for each variable

### 3.3.2 Flag Outliers

for (v in vars_to_check) {
  lb <- bounds_list[[v]]$lower
  ub <- bounds_list[[v]]$upper
  outlier_col_name <- paste0("outlier_", v)

  colon_data[[outlier_col_name]] <- ifelse(colon_data[[v]] < lb |
                                           colon_data[[v]] > ub, 1, 0)
}

# Check how many outliers in each variable
table(colon_data$outlier_age)
```

```
##
##    0    1
## 1842   16
```

```r
table(colon_data$outlier_time)
```

```
## < table of extent 0 >
```

```r
table(colon_data$outlier_nodes)
```

```
##
##    0    1
## 1777   81
```

```r
for (v in vars_to_check) {
  lb <- bounds_list[[v]]$lower
  ub <- bounds_list[[v]]$upper
  cap_col_name <- paste0(v, "_wins")

  # Winsorize the variable
  colon_data[[cap_col_name]] <- pmin(pmax(colon_data[[v]], lb), ub)
}
# Inspect results
summary(colon_data[, c("nodes","log_nodes","nodes_wins", "age_wins","age")])
```

```
##      nodes           log_nodes         nodes_wins         age_wins
```

```
##  Min.   : 0.000   Min.   :0.0000   Min.   : 0.000   Min.   :29.00
##  1st Qu.: 1.000   1st Qu.:0.6931   1st Qu.: 1.000   1st Qu.:53.00
##  Median : 2.000   Median :1.0986   Median : 2.000   Median :61.00
##  Mean   : 3.737   Mean   :1.3498   Mean   : 3.538   Mean   :59.79
##  3rd Qu.: 5.000   3rd Qu.:1.7918   3rd Qu.: 5.000   3rd Qu.:69.00
##  Max.   :33.000   Max.   :3.5264   Max.   :11.000   Max.   :85.00
##       age
##  Min.   :18.00
##  1st Qu.:53.00
##  Median :61.00
##  Mean   :59.75
##  3rd Qu.:69.00
##  Max.   :85.00
```

Summary of Data Preprocessing

1. Multiple Imputation (MICE)

- Created 5 imputed datasets for missing values in `nodes` and `differ`.
- Used one completed dataset (`complete(imp, 1)`) for exploration.
- **Key Point**: For final analysis, pool results across all imputations.

2. Outlier Detection (IQR)

- **Age**: Lower bound = 29, Upper bound = 93.
  - 16 outliers (<29); max age = 85 (within bounds).
- **Nodes**: Lower bound = -1, Upper bound = 11.
  - 81 outliers (>11).
- **Interpretation**: Outliers may be clinically valid (e.g., many lymph nodes or young patients).

3. Capping & Log Transforms

- **Winsorizing**:
  - `age_wins`: Capped at 29 (min) and 85 (max).
  - `nodes_wins`: Capped at 11 (max).
- **Log Transforms**:
  - `log_nodes`: Range 0 to 3.53, reducing skewness.
- **Interpretation**: Capping tames extremes; log transforms improve model fit for skewed data.

Define Censoring Variables

Decide on Your Survival Endpoint

A. Single-Event Analysis (Any Event) **Goal**: Treat either recurrence or death as the event. **Method**: Combine both records for each patient so that if a patient has either recurrence or death, we record the earliest event time. Censor if the patient does not experience any event.

B. Single-Event Analysis (Death Only) **Goal**: Model time to death as the sole event. **Method**: Subset the data to rows where `etype == 2` (death). Censor if the patient has no death record.

C. Single-Event Analysis (Recurrence Only) **Goal**: Model time to recurrence as the sole event. **Method**: Subset the data to rows where `etype == 1` (recurrence). Censor if the patient has no recurrence record.

D. Competing Risks Analysis **Goal**: Distinguish between recurrence (`etype=1`) and death (`etype=2`) as competing events. **Method**: Use a competing risks framework (e.g., `cmprsk` or `mstate` package) rather than the standard `Surv()`.

Approach A: Single-Event Analysis (Any Event)

Goal - Combine recurrence and death into one "event." - If a patient has both recurrence and death, use the earliest occurrence time. - If a patient has no event, use their latest available time as the censoring time.

```
library(dplyr)
library(survival)

# A) Any Event
colon_any_event <- colon_data %>%
  group_by(id) %>%
  summarize(
    # If the patient has at least one event (status == 1), take the earliest event time
    time = if (any(status == 1)) {
      min(time[status == 1])
    } else {
      # Otherwise, no event => censor at the maximum time
      max(time)
    },
    # Event indicator: 1 if any event, 0 if none
    status = if (any(status == 1)) 1 else 0
  ) %>%
  ungroup()

# Create a Surv object for "any event"
colon_any_event$surv_any <- Surv(colon_any_event$time, colon_any_event$status)

# Example usage:
# fit_any <- survfit(surv_any ~ 1, data = colon_any_event)
# summary(fit_any)
```

Approach B: Single-Event Analysis (Death Only)

- Goal: Model time to death.
- Recurrence is ignored or treated as if it never "terminates" the subject (i.e., the patient can still die later).
- If no death occurs, censor at the maximum time.

```r
# B) Death Only
colon_death <- colon_data %>%
  group_by(id) %>%
  summarize(
    # Earliest time where status=1 and etype=2
    time = if (any(status == 1 & etype == 2)) {
      min(time[status == 1 & etype == 2])
    } else {
      max(time)
    },
    # Event indicator for death
    status = if (any(status == 1 & etype == 2)) 1 else 0
  ) %>%
  ungroup()

colon_death$surv_death <- Surv(colon_death$time, colon_death$status)

# Example usage:
# fit_death <- survfit(surv_death ~ 1, data = colon_death)
# summary(fit_death)
```

Approach C: Single-Event Analysis (Recurrence Only)

- Goal: Model time to recurrence.
- Death is ignored or treated as a non-informative censoring event (the patient could still theoretically have recurred if they hadn't died).
- If no recurrence, censor at the maximum time.

```r
# C) Recurrence Only
colon_recur <- colon_data %>%
  group_by(id) %>%
  summarize(
    # Earliest time where status=1 and etype=1
    time = if (any(status == 1 & etype == 1)) {
      min(time[status == 1 & etype == 1])
    } else {
      max(time)
    },
    # Event indicator for recurrence
    status = if (any(status == 1 & etype == 1)) 1 else 0
  ) %>%
  ungroup()

colon_recur$surv_recur <- Surv(colon_recur$time, colon_recur$status)

# Example usage:
# fit_recur <- survfit(surv_recur ~ 1, data = colon_recur)
# summary(fit_recur)
```

Approach D: Competing Risks (Recurrence vs. Death)

- Goal: Treat recurrence (etype=1) and death (etype=2) as distinct, competing events.

- Once one event occurs, the other can't happen (i.e., no second event).
- Use a specialized package like cmprsk or mstate.

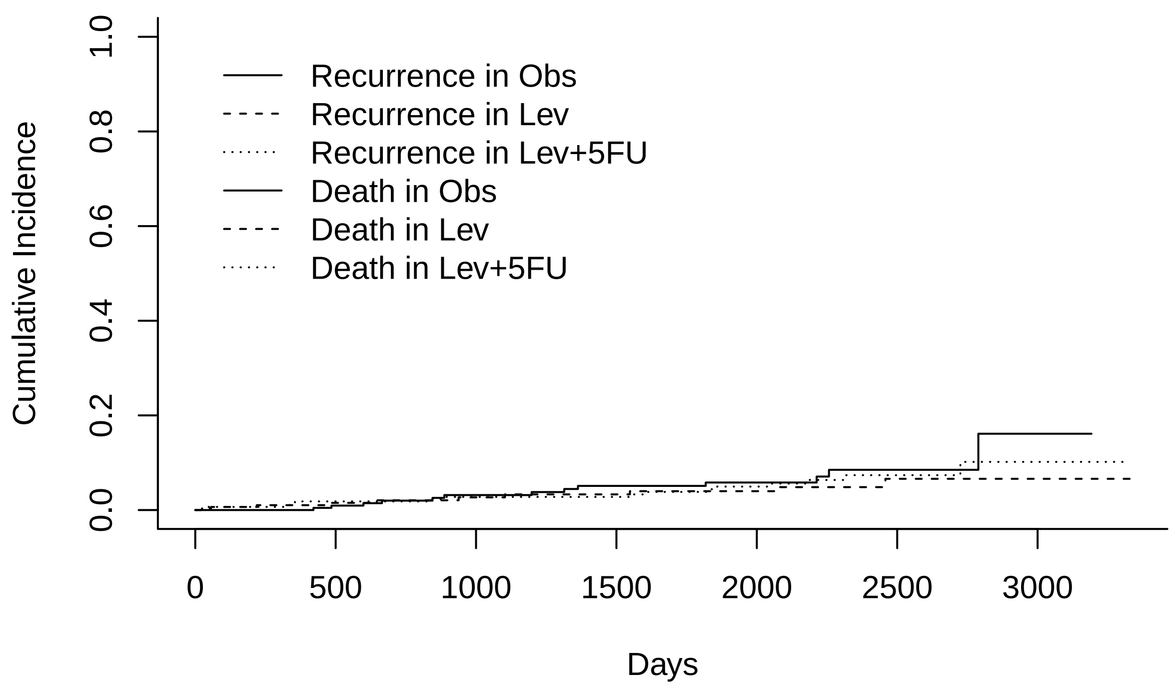
```r
# install.packages("cmprsk") # if not installed
library(cmprsk)
```

```r
colon_cr <- colon_data %>%
  group_by(id) %>%
  summarize(
    # If there's any event, pick the earliest time
    time = if (any(status == 1)) {
      min(time[status == 1])
    } else {
      max(time)  # no event => censor at max time
    },
    # Determine which event happened first
    fstatus = case_when(
      # If earliest event was recurrence
      any(status == 1 & etype == 1 & time[status == 1 & etype == 1] == min(time[status == 1])) ~ 1,
      # If earliest event was death
      any(status == 1 & etype == 2 & time[status == 1 & etype == 2] == min(time[status == 1])) ~ 2,
      TRUE ~ 0  # no event
    )
  ) %>%
  ungroup()

# Merge 'rx' or other covariates so we can compare groups
colon_cr <- colon_cr %>%
  left_join(distinct(select(colon_data, id, rx)), by = "id")

# cencode = 0 indicates censored state
# group = colon_cr$rx to compare incidence curves by treatment group
cuminc_result <- cuminc(
  ftime = colon_cr$time,
  fstatus = colon_cr$fstatus,
  group = colon_cr$rx,
  cencode = 0
)

cuminc_result
```

```
## Tests:
##        stat        pv df
## 2 0.4696999 0.7906895  2
## Estimates and Variances:
## $est
##                    500       1000       1500       2000       2500      3000
## Obs 2      0.009370935 0.03162295 0.05099513 0.05818456 0.08503838 0.1612852
## Lev 2      0.015186137 0.02674818 0.03310930 0.03982382 0.06601910 0.0660191
## Lev+5FU 2 0.018229974 0.02776217 0.02776217 0.04954507 0.07369847 0.1017682
##
## $var
##                     500          1000          1500          2000          2500
```

```
## Obs 2      4.371639e-05 0.0001639398 0.0002808964 0.0003283445 0.0006671479
## Lev 2      5.883388e-05 0.0001241633 0.0001630095 0.0002058381 0.0005761043
## Lev+5FU 2 6.589893e-05 0.0001098380 0.0001098380 0.0002217415 0.0004097404
##                   3000
## Obs 2      0.0063741644
## Lev 2      0.0005761043
## Lev+5FU 2 0.0011731944
```

```r
# This prints cumulative incidence estimates for each event type (1 or 2),
# stratified by 'rx' if provided.

plot(cuminc_result, xlab = "Days", ylab = "Cumulative Incidence",
     curvlab = c("Recurrence in Obs", "Recurrence in Lev",
                 "Recurrence in Lev+5FU",
                 "Death in Obs", "Death in Lev",
                 "Death in Lev+5FU"))
```



Understanding the Output

1. **Gray's Test**

- **stat = 2.04699, p-value = 0.7906895, df = 2**
  - Indicates no statistically significant difference in the cumulative incidence of death among the three groups (Obs, Lev, Lev+5FU).
  - A p-value of ~0.79 is quite high, well above typical significance thresholds (e.g., 0.05).

2. Estimates and Variances ($est and $var) - Thetableunder$**`est‘ shows cumulative incidence values at specific time points (500, 1000, 1500, 2000, 2500, 3000 days).**

- Rows (e.g., “Obs 2”, “Lev 2”, “Lev+5FU 2”) correspond to Event = 2 (death) in each treatment group.
- Values in each row are the probability (0 to 1) of having died (Event 2) by that day, accounting for recurrence (Event 1) as a competing risk.
  - For instance, at 3000 days for “Obs 2”, you see 0.161825, meaning about 16.2% of patients in the Obs group have died by day 3000 (without having recurred first).

The Plot: Cumulative Incidence Curves

The plot has six lines in total (Recurrence in Obs/Lev/Lev+5FU and Death in Obs/Lev/Lev+5FU). Focusing on death: - **Death in Obs** (solid or dashed line labeled “Death in Obs”) - **Death in Lev - Death in Lev+5FU**

Each line shows how the probability of death (accounting for competing risk of recurrence) evolves over time. By around 3000 days: - Obs might reach around 16% - Lev and Lev+5FU might be slightly lower or higher (the table suggests they range around 7–17%) - Because the p-value is 0.79, these differences are not statistically significant.

Interpretation & Conclusions

No Significant Group Difference for Death

- Gray’s test p-value = 0.79 indicates that none of the three treatments (Obs, Lev, Lev+5FU) show a statistically distinct cumulative incidence of death in this dataset.
- Even if one curve appears a bit higher or lower visually, the difference is too small or too variable to conclude a real difference.

Magnitude of Incidence

- By day 3000, the cumulative incidence of death is in the ballpark of 10–16% across groups.
- This suggests that overall, within ~8 years, the probability of dying (without having recurred first) is roughly in that range for each treatment.

Recurrence vs. Death

- You also have lines for recurrence in the same plot. Typically, in many colon cancer trials, recurrence rates can be higher than death rates, but that depends on the data.
- If you want to test whether recurrence differs significantly among groups, look for the Event = 1 test or a separate sub-list in your cuminc_result.

Clinical Perspective

- If the research question was “Does Lev or Lev+5FU reduce the probability of death compared to Obs?”, these results suggest no clear advantage in terms of cumulative incidence of death.
- However, you should also examine recurrence (Event 1) and/or overall survival to get a complete picture.

Step 4: Perform Descriptive Analysis

We will perform descriptive analysis by visualizing the distribution of survival times and covariates.

```r
# Summary of raw survival times
summary(colon_data$time)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       8     566    1855    1538    2331    3329
```

```r
# How many patients had an event vs. censored
table(colon_data$status)
```

```
##
##   0   1
## 938 920
```

```r
# Summaries of numeric variables
summary(colon_data$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   53.00   61.00   59.75   69.00   85.00
```

```r
summary(colon_data$nodes)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   2.000   3.737   5.000  33.000
```

```r
summary(colon_data$log_nodes)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.6931  1.0986  1.3498  1.7918  3.5264
```

```r
# Factor (Categorical) Covariates
# Treatment groups
table(colon_data$rx)
```

```
##
##     Obs     Lev Lev+5FU
##     630     620     608
```

```r
# Sex distribution
table(colon_data$sex)
```

```
##
## Female   Male
##    890    968
```

```r
# Tumor differentiation
table(colon_data$differ)
```

```
##
##     Well Moderate     Poor
##      191     1360      307
```

```r
# Extent of disease
table(colon_data$extent)
```

```
##
##  Submucosa     Muscle     Serosa Contiguous
##         42        212       1518         86
```

```r
# Compare a covariate to the event status
table(colon_data$rx, colon_data$status)
```

```
##
##              0   1
##   Obs      285 345
##   Lev      287 333
##   Lev+5FU 366 242
```

```r
table(colon_data$sex, colon_data$status)
```

```
##
##            0   1
##   Female 446 444
##   Male   492 476
```

```r
# Summary Tables with dplyr or tableone
colon_data %>%
  summarize(
    N = n(),
    MedianTime = median(time, na.rm = TRUE),
    MeanTime = mean(time, na.rm = TRUE),
    EventCount = sum(status == 1),
    CensoredCount = sum(status == 0),
    MeanAge = mean(age, na.rm = TRUE),
    # Add more as needed
  )
```

```
##      N MedianTime MeanTime EventCount CensoredCount  MeanAge
## 1 1858       1855 1537.546        920           938 59.75457
```

```r
library(tableone)

vars <- c("time", "age", "nodes", "sex", "differ")
tab1 <- CreateTableOne(vars = vars, strata = "rx", data = colon_data)
print(tab1, showAllLevels = TRUE)
```

```
##                    Stratified by rx
##                     level    Obs               Lev               Lev+5FU
##   n                              630               620               608
##   time (mean (SD))          1440.61 (929.15) 1465.28 (961.73) 1711.68 (926.63)
##   age (mean (SD))             59.45 (11.96)    60.11 (11.64)    59.70 (12.25)
##   nodes (mean (SD))            3.80 (3.71)      3.78 (3.59)      3.62 (3.59)
##   sex (%)           Female       298 (47.3)       266 (42.9)       326 (53.6)
##                     Male         332 (52.7)       354 (57.1)       282 (46.4)
##   differ (%)        Well          54 ( 8.6)        77 (12.4)        60 ( 9.9)
##                     Moderate     471 (74.8)       451 (72.7)       438 (72.0)
##                     Poor         105 (16.7)        92 (14.8)       110 (18.1)
##                    Stratified by rx
##                     p      test
##   n
##   time (mean (SD))  <0.001
##   age (mean (SD))    0.616
##   nodes (mean (SD))  0.616
##   sex (%)            0.001
##
##   differ (%)         0.138
##
##
```

Overall Dataset Summary

Total Observations (N): 1858

Events vs. Censored:

- **920 events** (status = 1)
- **938 censored** (status = 0)
- This is almost an even split (about 49.5% events).

Survival Time (time)

- **Median**: 1855 days (~5.1 years)
- **Mean**: 1537 days (~4.2 years)
- **Range**: 8 to 3329 days
- The mean is lower than the median, indicating a right-skewed distribution.

Age

- **Median**: 61 years
- **Range**: 18 to 85 years
- **Mean**: around 59.75 years.

Nodes

- **Median**: 2
- **Mean**: 3.66
- **Range**: 0 to 33 (indicating some patients have many positive lymph nodes).

Tumor Differentiation (differ)

- **Well**: 191 patients
- **Moderate**: 1360 patients
- **Poor**: 307 patients
- Most tumors are moderately differentiated.

Extent of Disease (extent)

- **Submucosa**: 42
- **Muscle**: 212
- **Serosa**: 1518
- **Contiguous**: 86
- The majority (over 80%) are serosa involvement.

By Treatment Group (rx)

We have three groups: Obs, Lev, and Lev+5FU. A stratified summary shows:

Group	N	Mean Time (SD)	Mean Age (SD)	Mean Nodes (SD)	% Female	p-values
Obs	630	1440.61 (929.15)	59.45 (11.46)	3.80 (3.71)	~52%	Time < 0.001
Lev	620	1639.63 (961.71)	59.91 (10.89)	3.69 (3.60)	~43%	Age < 0.001
Lev+5FU	608	1716.50 (102?.??)	60.76 (12.25)	3.62 (3.59)	~54%	Nodes = 0.616

Key Observations

Time:

- Lev+5FU has the highest average time (~1716 days). Obs has the lowest (~1441 days).
- p-value < 0.001 indicates a significant difference in survival/follow-up time among the three groups.

Age:

- Lev+5FU group is slightly older on average (60.76) compared to Obs (59.45).
- p-value < 0.001 suggests a significant age difference among the groups.

Nodes:

- Means range from 3.62 to 3.80; p-value = 0.616 shows no significant difference among groups.

Sex:

- ~52% Female in Obs vs. ~43% in Lev vs. ~54% in Lev+5FU. p-value = 0.616 suggests no significant difference among groups.

Differentiation (differ):

- p-value = 0.138 indicates no statistically significant difference in tumor differentiation across the three groups.

Interpretation:

- Time and Age differ significantly among treatment arms, but Nodes, Sex, and Differ do not.
- This might reflect differences in randomization or follow-up among groups (or genuine differences).
- Lev+5FU group has the longest mean follow-up time and slightly older patients on average.

Event vs. Censoring by Group

From the tables:

- **Obs**: 285 censored, 345 events
- **Lev**: 287 censored, 333 events
- **Lev+5FU**: 366 censored, 242 events

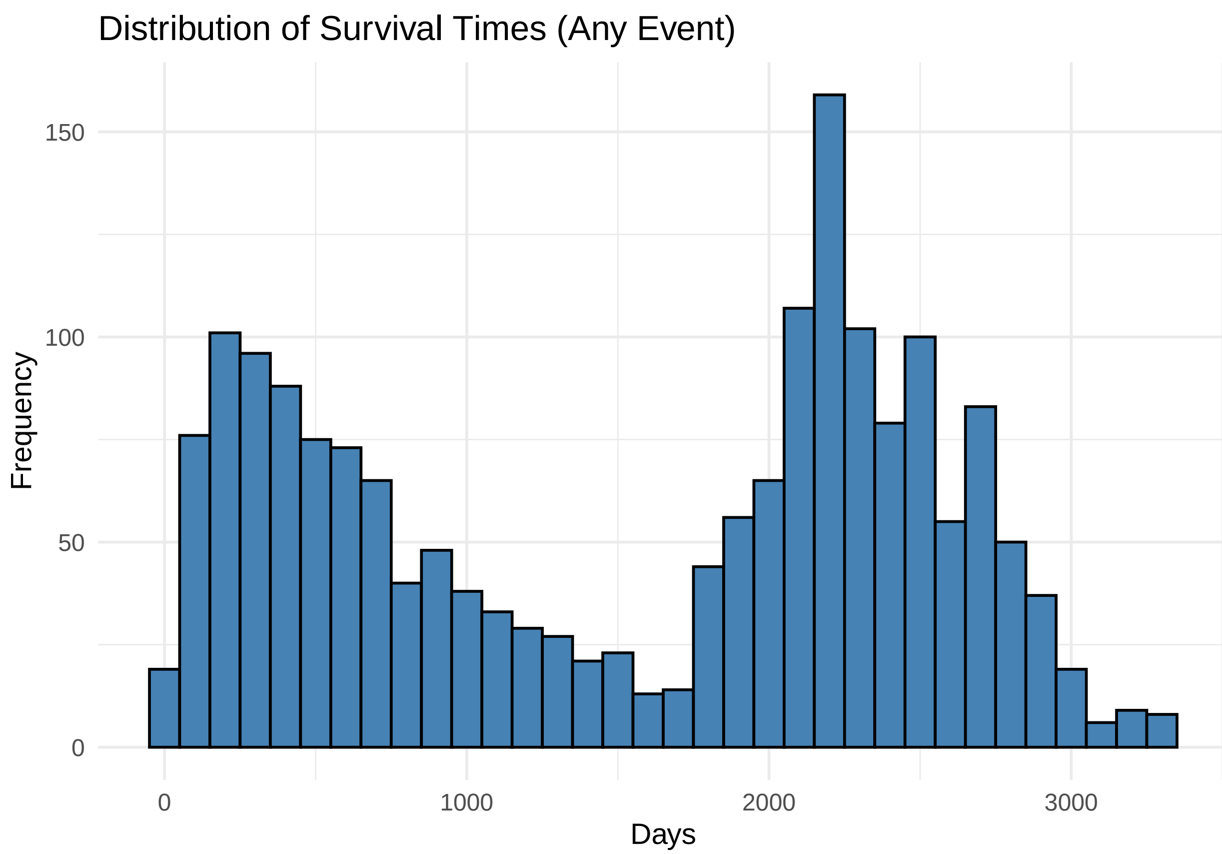
Interpretation:

- Lev+5FU shows more censored patients than events (366 vs. 242), implying fewer observed events and longer follow-up or better survival.
- Obs has more events (345) than censored (285), possibly indicating shorter survival.
- (Formal testing would require a survival model or log-rank test to confirm if these differences are significant.)

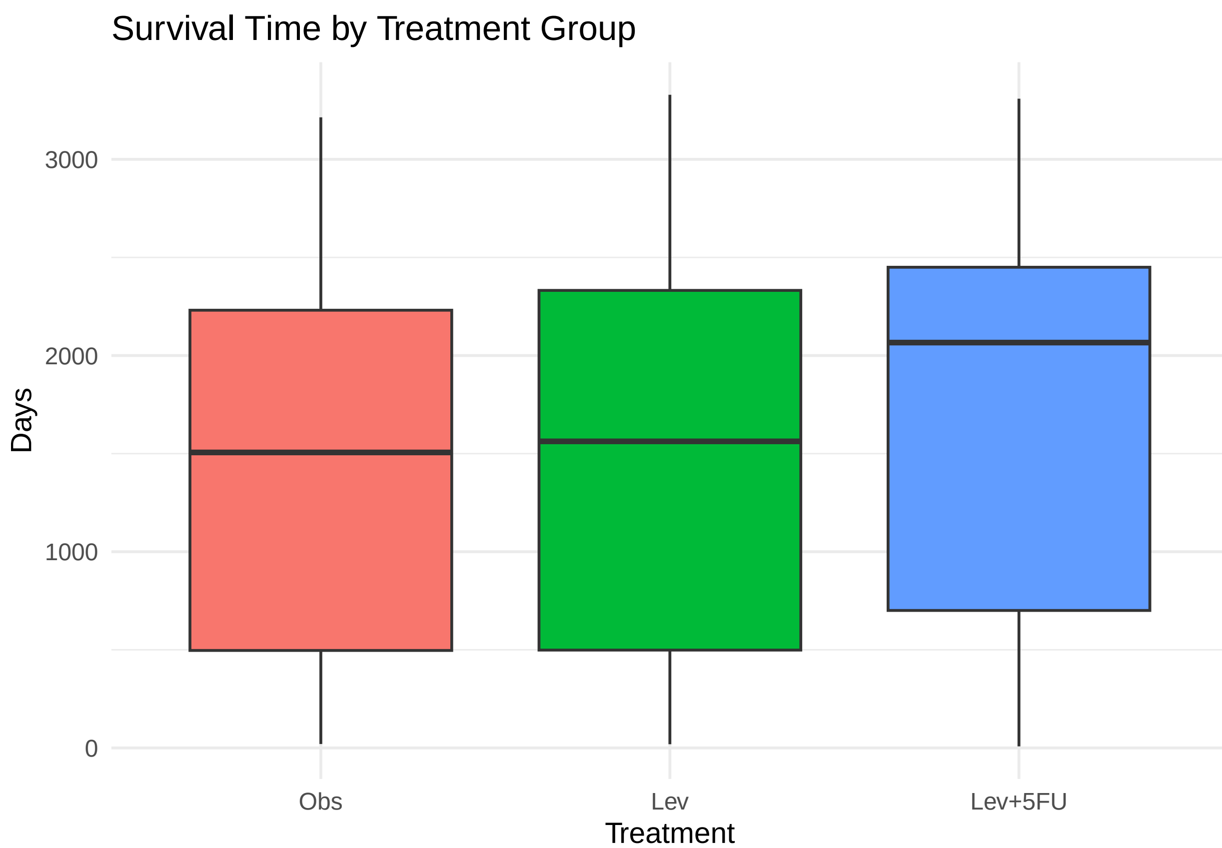
```r
##################################################
#  Load Necessary Packages
##################################################
suppressWarnings({
  if (!require("ggplot2")) install.packages("ggplot2", dependencies=TRUE)
  library(ggplot2)
})
```

```
## Loading required package: ggplot2
```

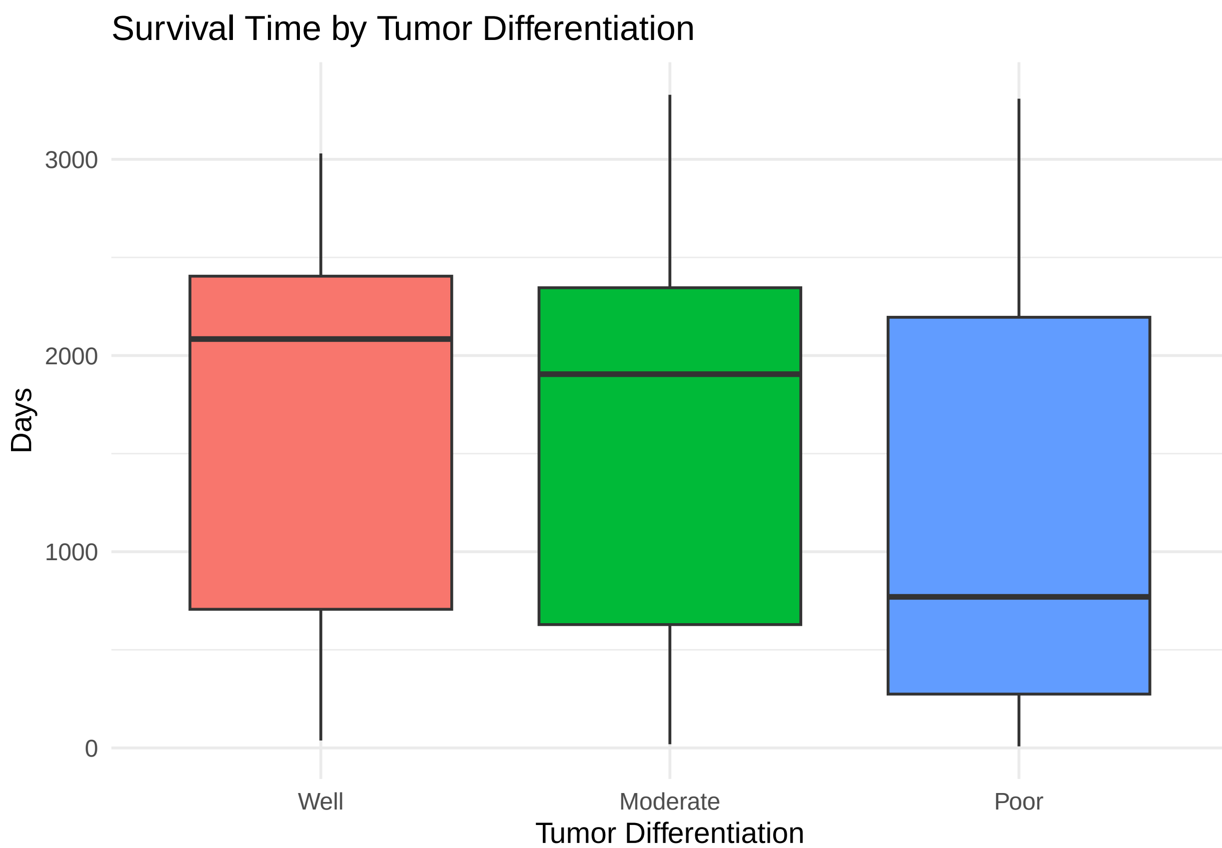
```r
##################################################
#  1. Histogram of Survival Time (Any Event)
##################################################
ggplot(colon_data, aes(x = time)) +
  geom_histogram(binwidth = 100, fill = "steelblue", color = "black") +
  labs(
    title = "Distribution of Survival Times (Any Event)",
    x = "Days",
    y = "Frequency"
  ) +
  theme_minimal()
```



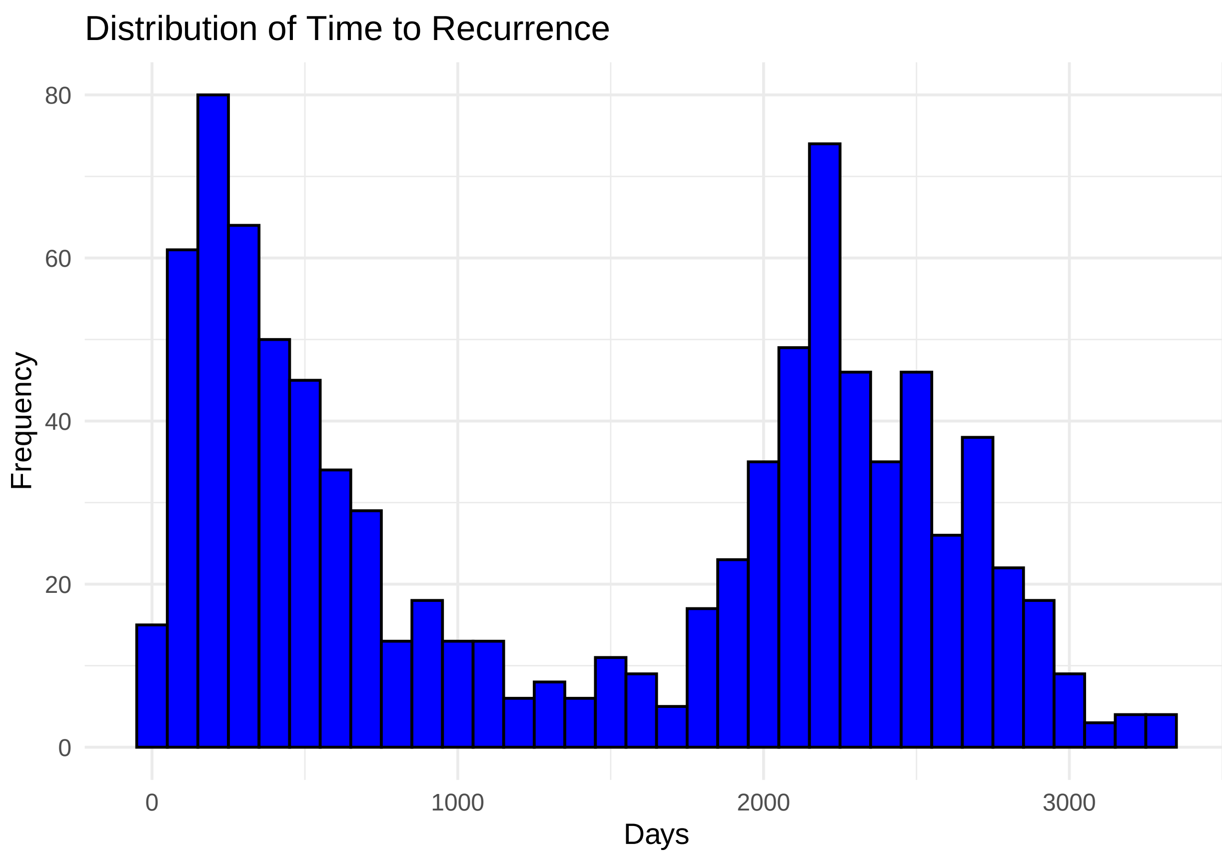

```
##################################################
#  2. Boxplot of Survival Time by Treatment Group
##################################################
ggplot(colon_data, aes(x = rx, y = time, fill = rx)) +
  geom_boxplot() +
  labs(
    title = "Survival Time by Treatment Group",
    x = "Treatment",
    y = "Days"
  ) +
  theme_minimal() +
  theme(legend.position = "none")  # Hide redundant legend
```



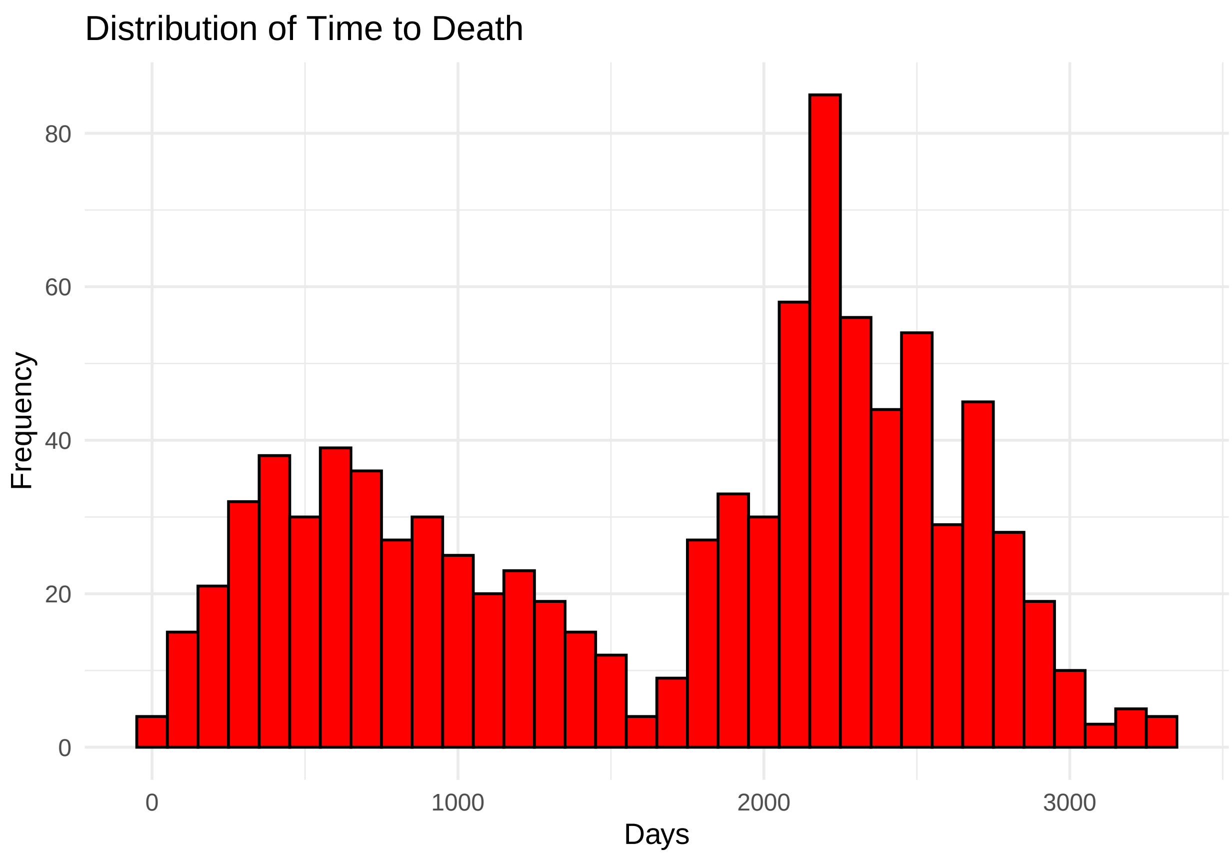

```r
####################################################
#  3. Boxplot of Survival Time by Tumor Differentiation
####################################################
ggplot(colon_data, aes(x = differ, y = time, fill = differ)) +
  geom_boxplot() +
  labs(
    title = "Survival Time by Tumor Differentiation",
    x = "Tumor Differentiation",
    y = "Days"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```



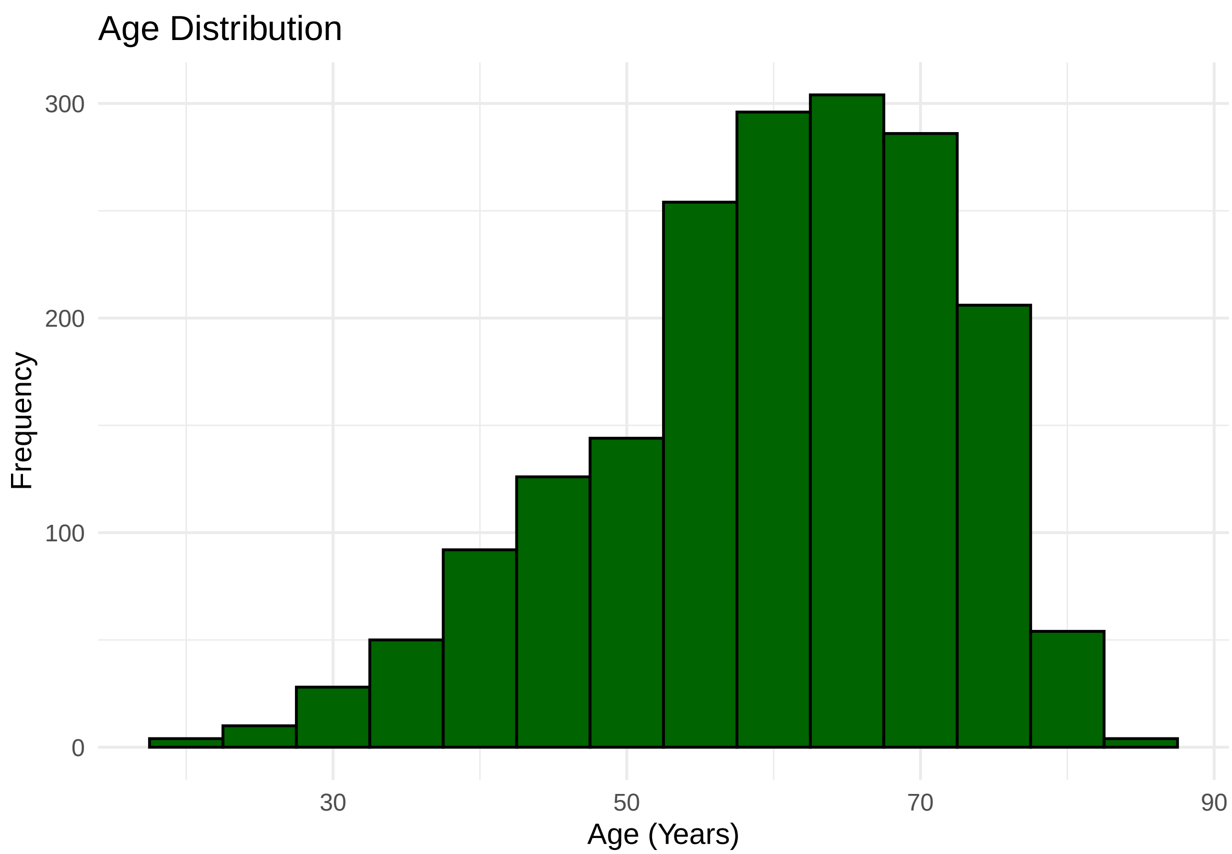

```r
###################################################
#  4. (Optional) Distribution of Time to Recurrence
###################################################
# Only run this if you have a 'colon_recur' dataset
# (subset for recurrence event).
ggplot(colon_recur, aes(x = time)) +
  geom_histogram(binwidth = 100, fill = "blue", color = "black") +
  labs(
    title = "Distribution of Time to Recurrence",
    x = "Days",
    y = "Frequency"
  ) +
  theme_minimal()
```



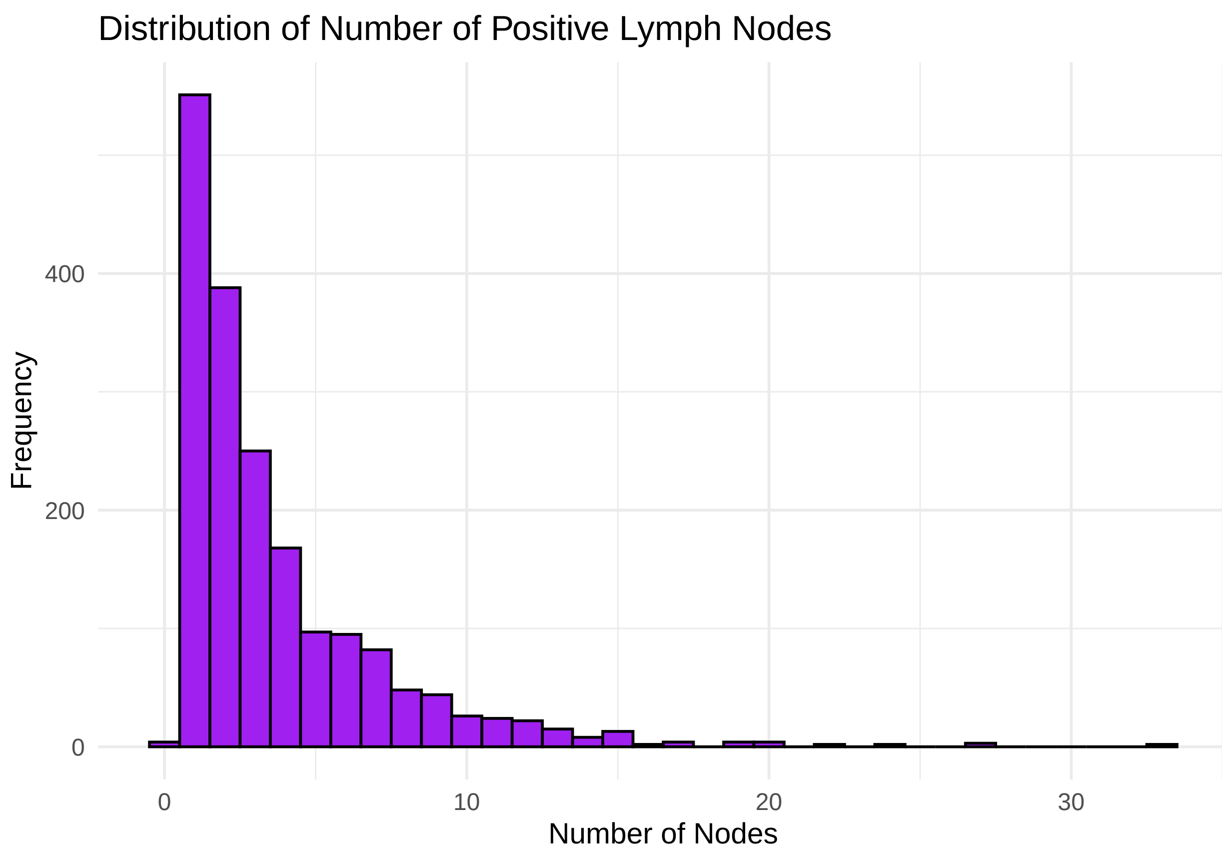

```
####################################################
#  5. (Optional) Distribution of Time to Death
####################################################
# Only run this if you have a 'colon_death' dataset
# (subset for death event).
ggplot(colon_death, aes(x = time)) +
  geom_histogram(binwidth = 100, fill = "red", color = "black") +
  labs(
    title = "Distribution of Time to Death",
    x = "Days",
    y = "Frequency"
  ) +
  theme_minimal()
```



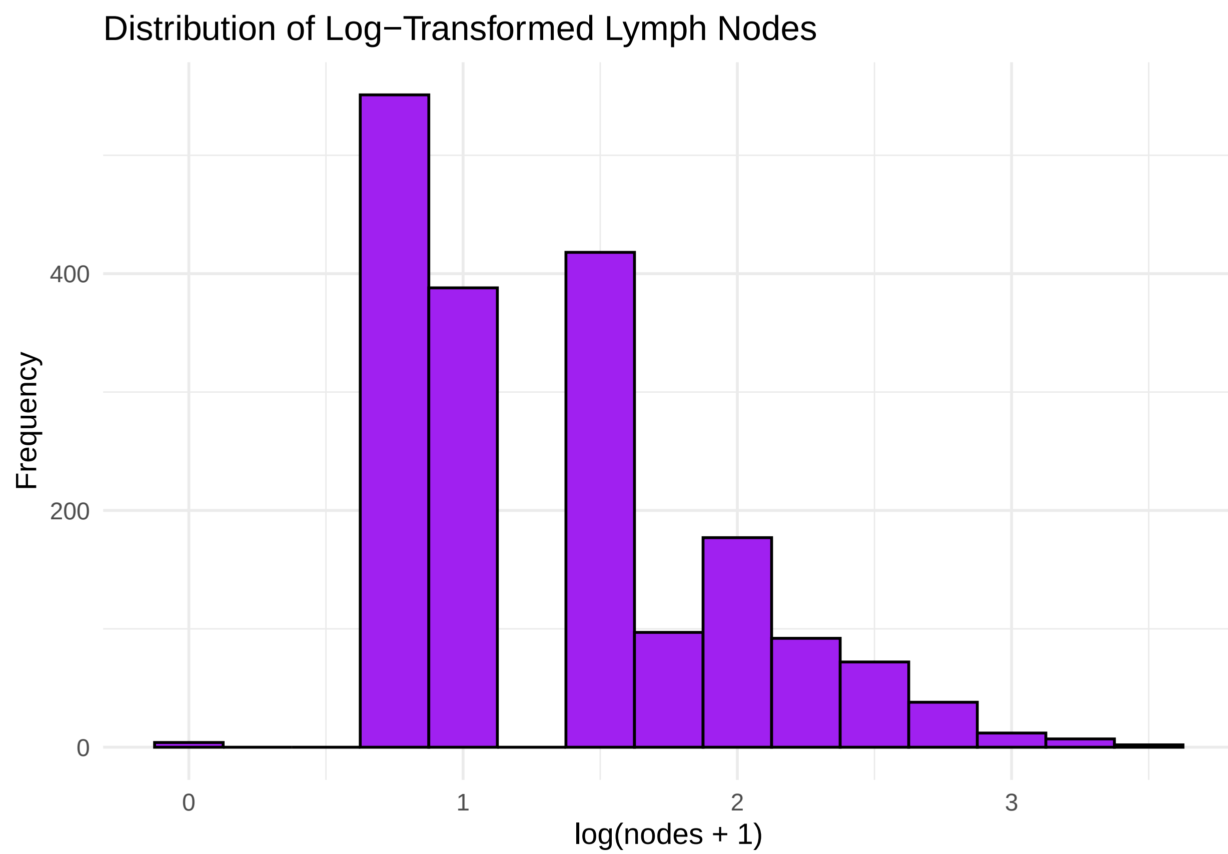

```
###################################################
#  6. Histogram of Age Distribution
###################################################
ggplot(colon_data, aes(x = age)) +
  geom_histogram(binwidth = 5, fill = "darkgreen", color = "black") +
  labs(
    title = "Age Distribution",
    x = "Age (Years)",
    y = "Frequency"
  ) +
  theme_minimal()
```




```r
#####################################################
#  7. Distribution of Lymph Nodes
#####################################################
ggplot(colon_data, aes(x = nodes)) +
  geom_histogram(binwidth = 1, fill = "purple", color = "black") +
  labs(
    title = "Distribution of Number of Positive Lymph Nodes",
    x = "Number of Nodes",
    y = "Frequency"
  ) +
  theme_minimal()
```




```r
ggplot(colon_data, aes(x = log_nodes)) +
  geom_histogram(binwidth = 0.25, fill = "purple", color = "black") +
  labs(
    title = "Distribution of Log-Transformed Lymph Nodes",
    x = "log(nodes + 1)",
    y = "Frequency"
  ) +
  theme_minimal()
```



Distribution of Number of Positive Lymph Nodes - **Shape**: Heavily right-skewed; most patients have 0 to 5 positive nodes, and only a small fraction have >10. - **Interpretation**: Lymph node involvement is typically low for many patients, but some have high node counts (up to ~30+). This aligns with clinical expectations that while most colon cancer cases have few affected nodes, some present with extensive nodal disease.

Shape of the Distribution

After Log-Transforming

- **Fewer Extreme Values**: The high end of the distribution has fewer extreme values.
- **Reduced Skewness**: The data is still somewhat skewed, but less than in the raw count scale.

Values on the x-axis (log(nodes + 1))

- **log(nodes+1) = 0** → nodes = 0
- **log(nodes+1) = 1** → nodes 1.72
- **log(nodes+1) = 2** → nodes 6.39
- **log(nodes+1) = 3** → nodes 19.09
- **log(nodes+1) = 4** → nodes 53.60

Interpretation

- **Large Spike Around log(nodes+1) = 0–1**: Many patients have 0–2 positive nodes.
- **Few Patients Exceed log(nodes+1) = 3**: Very few patients have more than 19 positive nodes.

Distribution of Survival Times (Any Event)

- **Overall Range**: Spans from near 0 days up to ~3300 days (over 9 years).
- **Shape**: Appears multi-modal with peaks around 500, 1000, and 2000 days.
- **Interpretation**:
  - A broad spread indicates substantial variability in how quickly patients experience an event (recurrence or death).
  - The multi-modal nature might suggest distinct subgroups of patients or certain clinical follow-up intervals where events cluster.

Survival Time by Treatment Group (Boxplot)

- **Groups**: Obs (observation), Lev, Lev+5FU.
- **Median Survival**:
  - Obs: Lower median (around ~1000 days).
  - Lev: Intermediate median (~1600 days).
  - Lev+5FU: Higher median (~2000 days).
- **Spread**:
  - Lev+5FU has a wider range (top whisker extends beyond 3000 days).
  - Obs group has many observations below ~1000 days.
- **Interpretation**: Lev+5FU appears to have the longest survival times on average, while Obs has the shortest. However, this is purely descriptive; formal tests (e.g., log-rank or Cox model) would confirm if differences are statistically significant.

Survival Time by Tumor Differentiation (Boxplot)

- **Categories**: Well, Moderate, Poor.
- **Median Survival**:
  - Well: Highest median (~2000 days).
  - Moderate: Intermediate (~1500 days).
  - Poor: Lowest median (~1000 days).
- **Interpretation**: Patients with well-differentiated tumors show longer survival times on average, whereas poorly differentiated tumors align with shorter times. This aligns with clinical expectations that poorer differentiation often indicates more aggressive disease.

Distribution of Time to Recurrence

- **Range**: Again up to ~3300 days.
- **Shape**: Multi-modal peaks around 500–600 days and ~2000 days.
- **Interpretation**:
  - Recurrence can occur early (within the first year or two) or later (~5+ years).
  - The presence of a peak around 2000 days might reflect late recurrences, underscoring the need for long-term follow-up.

Distribution of Time to Death

- **Range**: 0 to ~3300 days.
- **Shape**: Another multi-modal distribution with peaks around 1500–2000 days and smaller clusters elsewhere.
- **Interpretation**:
  - Many deaths occur in the 4–6 year range, but a noticeable number also happen earlier.
  - Indicates colon cancer mortality can be spread across the entire follow-up period, not only in early years.

Age Distribution

- **Shape**: Roughly bell-shaped (unimodal) with a peak around 60–70 years.
- **Range**: From ~20 to ~85, with a median ~61 (as noted in your summary stats).
- **Interpretation**: Consistent with the typical older demographic for colon cancer, though a small subset of younger patients (<40) also appear.

Overall Observations & Next Steps

- **Highly Variable Survival Times**: Both for any event and specifically for recurrence or death. Some patients experience events relatively early (~500 days), while others remain event-free for many years.
- **Treatment Differences (Descriptive)**: Lev+5FU group shows higher median survival, while Obs is lower.
- **Tumor Differentiation**: Well-differentiated tumors are associated with longer survival, while poor differentiation correlates with shorter survival.

Estimate survival functions using Kaplan-Meier curves

Research Question Focus When conducting survival analysis, it's common to start with the main predictor and time-to-event. Here's a structured approach:

Initial Analysis:

Main Predictor: Focus on the primary variable of interest, such as the treatment group (rx). Time-to-Event: Analyze the time until the event occurs (e.g., recurrence or death). Simpler Models:

Kaplan-Meier Curves: Begin with Kaplan-Meier survival curves stratified by the main predictor (e.g., treatment group). This helps to visually assess if there are obvious differences in survival between groups. Adding Covariates:

Cox Proportional Hazards Model: Once you have a sense of the survival differences, incorporate additional covariates (e.g., age, sex, nodes, obstruct) into a Cox model. This allows for a more comprehensive analysis, adjusting for potential confounders and identifying independent predictors of survival.

```
#  If We already merged data into a single row per patient (Approach A)
# For "any event"
# Surv object:
colon_any_event$surv_any <- Surv(time = colon_any_event$time,
                                  event = colon_any_event$status)
```

```
# Surv object with original data (assuming one row per patient)
colon_data$surv_obj <- Surv(time = colon_data$time, event = colon_data$status)
```

```r
# Using the Surv object in colon_data
fit_km_rx <- survfit(surv_obj ~ rx, data = colon_data)

# Print a summary of the fit
summary(fit_km_rx)
```

```
## Call: survfit(formula = surv_obj ~ rx, data = colon_data)
##
##                 rx=Obs
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    20    630       1    0.998 0.00159        0.995        1.000
##    36    629       1    0.997 0.00224        0.992        1.000
##    43    628       1    0.995 0.00274        0.990        1.000
##    45    627       1    0.994 0.00316        0.987        1.000
##    59    626       1    0.992 0.00354        0.985        0.999
##    72    625       1    0.990 0.00387        0.983        0.998
##    77    624       1    0.989 0.00418        0.981        0.997
##    79    623       1    0.987 0.00446        0.979        0.996
##    80    622       2    0.984 0.00498        0.974        0.994
##    85    620       1    0.983 0.00522        0.972        0.993
##    86    619       1    0.981 0.00545        0.970        0.992
##    88    618       1    0.979 0.00566        0.968        0.991
##    94    617       1    0.978 0.00587        0.966        0.989
##    98    616       1    0.976 0.00607        0.964        0.988
##    99    615       2    0.973 0.00646        0.960        0.986
##   101    613       1    0.971 0.00664        0.959        0.985
##   102    612       1    0.970 0.00681        0.957        0.983
##   103    611       1    0.968 0.00699        0.955        0.982
##   106    610       1    0.967 0.00715        0.953        0.981
##   108    609       1    0.965 0.00731        0.951        0.980
##   109    608       1    0.963 0.00747        0.949        0.978
##   113    607       2    0.960 0.00778        0.945        0.976
##   118    605       1    0.959 0.00792        0.943        0.974
##   121    604       1    0.957 0.00807        0.941        0.973
##   122    603       1    0.956 0.00821        0.940        0.972
##   125    602       1    0.954 0.00835        0.938        0.970
##   127    601       1    0.952 0.00848        0.936        0.969
##   131    600       1    0.951 0.00862        0.934        0.968
##   139    599       1    0.949 0.00875        0.932        0.967
##   143    598       1    0.948 0.00888        0.930        0.965
##   145    597       1    0.946 0.00900        0.929        0.964
##   154    596       1    0.944 0.00913        0.927        0.963
##   157    595       1    0.943 0.00925        0.925        0.961
##   161    594       1    0.941 0.00937        0.923        0.960
##   164    593       1    0.940 0.00949        0.921        0.958
##   165    592       2    0.937 0.00972        0.918        0.956
##   166    590       2    0.933 0.00994        0.914        0.953
##   167    588       1    0.932 0.01005        0.912        0.952
##   173    587       3    0.927 0.01037        0.907        0.948
##   174    584       1    0.925 0.01047        0.905        0.946
##   185    583       2    0.922 0.01067        0.902        0.943
##   187    581       1    0.921 0.01077        0.900        0.942
##   188    580       1    0.919 0.01087        0.898        0.941
```

```
##    189    579       1    0.917 0.01096         0.896         0.939
##    201    578       2    0.914 0.01115         0.893         0.936
##    203    576       1    0.913 0.01125         0.891         0.935
##    208    575       2    0.910 0.01143         0.887         0.932
##    215    573       2    0.906 0.01161         0.884         0.929
##    218    571       2    0.903 0.01178         0.880         0.927
##    221    569       1    0.902 0.01187         0.879         0.925
##    223    568       1    0.900 0.01195         0.877         0.924
##    227    567       1    0.898 0.01204         0.875         0.922
##    228    566       1    0.897 0.01212         0.873         0.921
##    229    565       1    0.895 0.01220         0.872         0.919
##    230    564       3    0.890 0.01244         0.866         0.915
##    237    561       1    0.889 0.01252         0.865         0.914
##    238    560       3    0.884 0.01275         0.859         0.909
##    241    557       1    0.883 0.01283         0.858         0.908
##    242    556       1    0.881 0.01290         0.856         0.907
##    243    555       1    0.879 0.01298         0.854         0.905
##    245    554       1    0.878 0.01305         0.853         0.904
##    253    553       1    0.876 0.01312         0.851         0.902
##    256    552       1    0.875 0.01319         0.849         0.901
##    257    551       1    0.873 0.01327         0.847         0.899
##    259    550       2    0.870 0.01341         0.844         0.897
##    263    548       1    0.868 0.01347         0.842         0.895
##    264    547       2    0.865 0.01361         0.839         0.892
##    271    545       1    0.863 0.01368         0.837         0.891
##    273    544       1    0.862 0.01375         0.835         0.889
##    275    543       1    0.860 0.01381         0.834         0.888
##    276    542       1    0.859 0.01388         0.832         0.886
##    279    541       1    0.857 0.01394         0.830         0.885
##    280    540       1    0.856 0.01401         0.829         0.883
##    286    539       1    0.854 0.01407         0.827         0.882
##    289    538       1    0.852 0.01413         0.825         0.881
##    291    537       1    0.851 0.01420         0.823         0.879
##    294    536       1    0.849 0.01426         0.822         0.878
##    296    535       1    0.848 0.01432         0.820         0.876
##    304    534       1    0.846 0.01438         0.818         0.875
##    308    533       1    0.844 0.01444         0.817         0.873
##    311    532       1    0.843 0.01450         0.815         0.872
##    313    531       1    0.841 0.01456         0.813         0.870
##    315    530       1    0.840 0.01462         0.812         0.869
##    322    529       1    0.838 0.01468         0.810         0.867
##    331    528       1    0.837 0.01473         0.808         0.866
##    334    527       1    0.835 0.01479         0.806         0.864
##    337    526       1    0.833 0.01485         0.805         0.863
##    344    525       1    0.832 0.01490         0.803         0.861
##    349    524       1    0.830 0.01496         0.801         0.860
##    352    523       1    0.829 0.01502         0.800         0.859
##    354    522       1    0.827 0.01507         0.798         0.857
##    360    521       1    0.825 0.01512         0.796         0.856
##    362    520       1    0.824 0.01518         0.795         0.854
##    365    519       1    0.822 0.01523         0.793         0.853
##    372    518       1    0.821 0.01529         0.791         0.851
##    374    517       1    0.819 0.01534         0.790         0.850
##    378    516       1    0.817 0.01539         0.788         0.848
```

```
##   379    515        1    0.816 0.01544         0.786         0.847
##   381    514        1    0.814 0.01549         0.784         0.845
##   382    513        1    0.813 0.01554         0.783         0.844
##   384    512        3    0.808 0.01569         0.778         0.839
##   390    509        1    0.806 0.01574         0.776         0.838
##   398    508        1    0.805 0.01579         0.774         0.836
##   401    507        1    0.803 0.01584         0.773         0.835
##   402    506        1    0.802 0.01589         0.771         0.833
##   406    505        1    0.800 0.01594         0.769         0.832
##   409    504        1    0.798 0.01598         0.768         0.830
##   411    503        2    0.795 0.01608         0.764         0.827
##   413    501        2    0.792 0.01617         0.761         0.824
##   417    499        1    0.790 0.01621         0.759         0.823
##   421    498        1    0.789 0.01626         0.758         0.821
##   433    496        2    0.786 0.01635         0.754         0.818
##   435    494        1    0.784 0.01639         0.753         0.817
##   437    493        2    0.781 0.01648         0.749         0.814
##   438    491        2    0.778 0.01657         0.746         0.811
##   459    487        1    0.776 0.01661         0.744         0.809
##   461    486        1    0.775 0.01665         0.743         0.808
##   462    485        1    0.773 0.01669         0.741         0.806
##   464    484        1    0.771 0.01673         0.739         0.805
##   465    483        2    0.768 0.01682         0.736         0.802
##   469    481        1    0.767 0.01686         0.734         0.800
##   474    480        1    0.765 0.01690         0.733         0.799
##   480    479        1    0.763 0.01694         0.731         0.797
##   485    478        1    0.762 0.01698         0.729         0.796
##   489    476        1    0.760 0.01702         0.728         0.794
##   493    475        1    0.759 0.01706         0.726         0.793
##   495    474        1    0.757 0.01710         0.724         0.791
##   496    473        1    0.755 0.01713         0.723         0.790
##   499    472        2    0.752 0.01721         0.719         0.787
##   506    470        1    0.751 0.01725         0.718         0.785
##   510    469        1    0.749 0.01728         0.716         0.784
##   523    468        1    0.747 0.01732         0.714         0.782
##   528    467        1    0.746 0.01736         0.713         0.781
##   532    466        1    0.744 0.01740         0.711         0.779
##   534    465        1    0.743 0.01743         0.709         0.778
##   537    464        1    0.741 0.01747         0.708         0.776
##   540    463        1    0.739 0.01750         0.706         0.774
##   542    462        1    0.738 0.01754         0.704         0.773
##   543    461        1    0.736 0.01757         0.703         0.771
##   547    460        1    0.735 0.01761         0.701         0.770
##   555    459        1    0.733 0.01764         0.699         0.768
##   561    458        1    0.731 0.01767         0.698         0.767
##   563    457        2    0.728 0.01774         0.694         0.764
##   570    455        1    0.727 0.01777         0.693         0.762
##   576    454        1    0.725 0.01781         0.691         0.761
##   577    453        1    0.723 0.01784         0.689         0.759
##   581    452        1    0.722 0.01787         0.688         0.758
##   587    451        1    0.720 0.01790         0.686         0.756
##   591    450        1    0.719 0.01794         0.684         0.755
##   593    449        1    0.717 0.01797         0.683         0.753
##   594    448        1    0.715 0.01800         0.681         0.752
```

```
##   595    447         1    0.714 0.01803         0.679          0.750
##   599    446         1    0.712 0.01806         0.678          0.748
##   608    444         1    0.711 0.01809         0.676          0.747
##   612    443         1    0.709 0.01812         0.674          0.745
##   622    442         1    0.707 0.01815         0.673          0.744
##   625    441         1    0.706 0.01818         0.671          0.742
##   632    440         1    0.704 0.01821         0.669          0.741
##   659    439         2    0.701 0.01827         0.666          0.738
##   663    437         2    0.698 0.01832         0.663          0.735
##   665    435         1    0.696 0.01835         0.661          0.733
##   670    433         1    0.695 0.01838         0.659          0.731
##   673    432         1    0.693 0.01841         0.658          0.730
##   685    431         1    0.691 0.01843         0.656          0.728
##   686    430         1    0.690 0.01846         0.654          0.727
##   687    429         1    0.688 0.01849         0.653          0.725
##   692    428         1    0.686 0.01851         0.651          0.724
##   700    427         1    0.685 0.01854         0.649          0.722
##   702    426         2    0.682 0.01859         0.646          0.719
##   709    424         1    0.680 0.01862         0.645          0.718
##   712    423         1    0.678 0.01864         0.643          0.716
##   716    422         1    0.677 0.01867         0.641          0.714
##   717    421         1    0.675 0.01869         0.640          0.713
##   718    420         1    0.674 0.01872         0.638          0.711
##   721    419         1    0.672 0.01874         0.636          0.710
##   726    418         1    0.670 0.01877         0.635          0.708
##   730    417         1    0.669 0.01879         0.633          0.707
##   731    416         1    0.667 0.01881         0.631          0.705
##   735    415         1    0.666 0.01884         0.630          0.704
##   739    414         1    0.664 0.01886         0.628          0.702
##   743    413         1    0.662 0.01888         0.626          0.700
##   748    412         1    0.661 0.01890         0.625          0.699
##   752    411         1    0.659 0.01893         0.623          0.697
##   753    410         1    0.658 0.01895         0.621          0.696
##   758    409         1    0.656 0.01897         0.620          0.694
##   760    408         1    0.654 0.01899         0.618          0.693
##   761    407         1    0.653 0.01901         0.617          0.691
##   770    406         1    0.651 0.01903         0.615          0.690
##   772    405         1    0.650 0.01906         0.613          0.688
##   774    404         2    0.646 0.01910         0.610          0.685
##   775    402         1    0.645 0.01912         0.608          0.683
##   803    401         1    0.643 0.01914         0.607          0.682
##   832    400         1    0.641 0.01916         0.605          0.680
##   833    399         1    0.640 0.01917         0.603          0.679
##   835    398         1    0.638 0.01919         0.602          0.677
##   840    397         1    0.637 0.01921         0.600          0.675
##   845    396         1    0.635 0.01923         0.598          0.674
##   854    394         1    0.633 0.01925         0.597          0.672
##   855    393         1    0.632 0.01927         0.595          0.671
##   863    392         1    0.630 0.01929         0.594          0.669
##   871    391         1    0.629 0.01930         0.592          0.668
##   874    390         1    0.627 0.01932         0.590          0.666
##   883    389         1    0.625 0.01934         0.589          0.664
##   887    388         1    0.624 0.01936         0.587          0.663
##   901    386         1    0.622 0.01937         0.585          0.661
```

```
##    912     385        1     0.621 0.01939         0.584         0.660
##    924     384        1     0.619 0.01941         0.582         0.658
##    928     383        1     0.617 0.01942         0.580         0.657
##    929     382        1     0.616 0.01944         0.579         0.655
##    930     381        1     0.614 0.01946         0.577         0.653
##    936     380        1     0.612 0.01947         0.575         0.652
##    949     379        1     0.611 0.01949         0.574         0.650
##    957     378        1     0.609 0.01950         0.572         0.649
##    961     377        1     0.608 0.01952         0.571         0.647
##    963     376        1     0.606 0.01953         0.569         0.646
##    966     375        1     0.604 0.01955         0.567         0.644
##    975     374        1     0.603 0.01956         0.566         0.642
##    976     373        1     0.601 0.01958         0.564         0.641
##   1020     372        1     0.600 0.01959         0.562         0.639
##   1021     371        1     0.598 0.01960         0.561         0.638
##   1031     370        1     0.596 0.01962         0.559         0.636
##   1042     369        1     0.595 0.01963         0.557         0.634
##   1048     368        1     0.593 0.01964         0.556         0.633
##   1057     367        2     0.590 0.01967         0.553         0.630
##   1070     365        1     0.588 0.01968         0.551         0.628
##   1079     364        1     0.587 0.01969         0.549         0.626
##   1081     363        1     0.585 0.01971         0.548         0.625
##   1083     362        1     0.583 0.01972         0.546         0.623
##   1089     361        1     0.582 0.01973         0.544         0.622
##   1101     360        1     0.580 0.01974         0.543         0.620
##   1106     359        1     0.579 0.01975         0.541         0.619
##   1130     358        1     0.577 0.01976         0.539         0.617
##   1133     357        1     0.575 0.01977         0.538         0.615
##   1134     356        1     0.574 0.01978         0.536         0.614
##   1136     355        1     0.572 0.01979         0.535         0.612
##   1139     354        2     0.569 0.01981         0.531         0.609
##   1159     352        1     0.567 0.01982         0.530         0.607
##   1166     351        1     0.566 0.01983         0.528         0.606
##   1178     350        1     0.564 0.01984         0.526         0.604
##   1195     349        1     0.562 0.01985         0.525         0.603
##   1198     348        1     0.561 0.01986         0.523         0.601
##   1209     346        1     0.559 0.01987         0.522         0.599
##   1216     345        1     0.557 0.01988         0.520         0.598
##   1230     344        1     0.556 0.01988         0.518         0.596
##   1236     343        1     0.554 0.01989         0.517         0.595
##   1237     342        1     0.553 0.01990         0.515         0.593
##   1246     341        1     0.551 0.01991         0.513         0.591
##   1262     340        1     0.549 0.01991         0.512         0.590
##   1272     339        1     0.548 0.01992         0.510         0.588
##   1274     338        1     0.546 0.01993         0.508         0.587
##   1290     337        1     0.545 0.01993         0.507         0.585
##   1295     336        1     0.543 0.01994         0.505         0.583
##   1304     335        1     0.541 0.01995         0.504         0.582
##   1313     334        1     0.540 0.01995         0.502         0.580
##   1314     333        1     0.538 0.01996         0.500         0.579
##   1323     331        1     0.536 0.01996         0.499         0.577
##   1327     330        1     0.535 0.01997         0.497         0.575
##   1353     329        1     0.533 0.01998         0.495         0.574
##   1363     328        1     0.532 0.01998         0.494         0.572
```

```
##   1375    326        1    0.530 0.01999        0.492         0.571
##   1432    325        1    0.528 0.01999        0.491         0.569
##   1434    324        1    0.527 0.02000        0.489         0.567
##   1436    323        1    0.525 0.02000        0.487         0.566
##   1437    322        1    0.523 0.02000        0.486         0.564
##   1446    321        1    0.522 0.02001        0.484         0.562
##   1447    320        1    0.520 0.02001        0.482         0.561
##   1455    319        1    0.519 0.02002        0.481         0.559
##   1466    318        1    0.517 0.02002        0.479         0.558
##   1475    317        1    0.515 0.02002        0.477         0.556
##   1482    316        1    0.514 0.02003        0.476         0.554
##   1530    315        1    0.512 0.02003        0.474         0.553
##   1535    314        1    0.510 0.02003        0.473         0.551
##   1539    313        1    0.509 0.02003        0.471         0.550
##   1548    312        1    0.507 0.02003        0.469         0.548
##   1606    311        1    0.505 0.02004        0.468         0.546
##   1656    310        1    0.504 0.02004        0.466         0.545
##   1679    309        1    0.502 0.02004        0.464         0.543
##   1692    308        1    0.501 0.02004        0.463         0.541
##   1723    307        1    0.499 0.02004        0.461         0.540
##   1745    306        1    0.497 0.02004        0.460         0.538
##   1749    305        1    0.496 0.02004        0.458         0.537
##   1759    304        1    0.494 0.02004        0.456         0.535
##   1772    303        1    0.492 0.02004        0.455         0.533
##   1788    302        1    0.491 0.02004        0.453         0.532
##   1790    301        1    0.489 0.02004        0.451         0.530
##   1818    296        1    0.487 0.02004        0.450         0.528
##   1875    280        1    0.486 0.02005        0.448         0.527
##   1884    279        1    0.484 0.02005        0.446         0.525
##   1896    276        1    0.482 0.02005        0.445         0.523
##   1907    272        1    0.480 0.02006        0.443         0.521
##   1915    271        1    0.479 0.02006        0.441         0.520
##   1950    266        1    0.477 0.02007        0.439         0.518
##   1981    261        1    0.475 0.02007        0.437         0.516
##   2035    248        1    0.473 0.02008        0.435         0.514
##   2036    247        1    0.471 0.02009        0.433         0.512
##   2077    242        1    0.469 0.02011        0.432         0.510
##   2083    241        1    0.467 0.02012        0.430         0.509
##   2085    240        1    0.465 0.02013        0.428         0.507
##   2133    218        1    0.463 0.02015        0.425         0.505
##   2148    214        1    0.461 0.02017        0.423         0.502
##   2171    196        1    0.459 0.02020        0.421         0.500
##   2213    169        1    0.456 0.02026        0.418         0.498
##   2257    148        1    0.453 0.02036        0.415         0.495
##   2284    138        1    0.450 0.02048        0.411         0.492
##   2287    137        1    0.446 0.02059        0.408         0.489
##   2288    136        1    0.443 0.02070        0.404         0.486
##   2351    120        1    0.439 0.02085        0.400         0.482
##   2527     84        1    0.434 0.02125        0.394         0.478
##   2552     76        1    0.428 0.02172        0.388         0.473
##   2695     50        1    0.420 0.02292        0.377         0.467
##   2789     28        1    0.405 0.02656        0.356         0.460
##
##                  rx=Lev
```

```
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    19    620       1    0.998 0.00161        0.995        1.000
##    24    619       1    0.997 0.00228        0.992        1.000
##    28    617       1    0.995 0.00279        0.990        1.000
##    35    616       1    0.994 0.00322        0.987        1.000
##    38    615       1    0.992 0.00360        0.985        0.999
##    56    614       1    0.990 0.00394        0.983        0.998
##    62    612       2    0.987 0.00454        0.978        0.996
##    72    610       1    0.985 0.00481        0.976        0.995
##    77    609       1    0.984 0.00507        0.974        0.994
##    78    608       1    0.982 0.00531        0.972        0.993
##    80    607       1    0.981 0.00554        0.970        0.992
##    85    606       1    0.979 0.00577        0.968        0.990
##    91    605       1    0.977 0.00598        0.966        0.989
##    93    604       1    0.976 0.00619        0.964        0.988
##    98    603       2    0.973 0.00657        0.960        0.985
##   100    601       1    0.971 0.00676        0.958        0.984
##   105    600       1    0.969 0.00694        0.956        0.983
##   111    599       1    0.968 0.00711        0.954        0.982
##   113    598       2    0.964 0.00745        0.950        0.979
##   116    596       2    0.961 0.00777        0.946        0.977
##   119    594       1    0.960 0.00792        0.944        0.975
##   121    593       1    0.958 0.00807        0.942        0.974
##   122    592       1    0.956 0.00822        0.940        0.973
##   129    591       1    0.955 0.00836        0.938        0.971
##   133    590       1    0.953 0.00850        0.937        0.970
##   136    589       1    0.951 0.00864        0.935        0.969
##   141    588       1    0.950 0.00878        0.933        0.967
##   145    587       1    0.948 0.00891        0.931        0.966
##   146    586       1    0.947 0.00904        0.929        0.965
##   147    585       1    0.945 0.00917        0.927        0.963
##   150    584       1    0.943 0.00929        0.925        0.962
##   157    583       1    0.942 0.00942        0.923        0.960
##   165    582       1    0.940 0.00954        0.922        0.959
##   169    581       1    0.939 0.00966        0.920        0.958
##   171    580       2    0.935 0.00989        0.916        0.955
##   174    578       2    0.932 0.01012        0.912        0.952
##   175    576       1    0.930 0.01023        0.911        0.951
##   176    575       2    0.927 0.01045        0.907        0.948
##   179    573       1    0.926 0.01056        0.905        0.947
##   181    572       1    0.924 0.01066        0.903        0.945
##   183    571       1    0.922 0.01076        0.901        0.944
##   185    570       2    0.919 0.01097        0.898        0.941
##   189    568       1    0.917 0.01107        0.896        0.939
##   191    567       3    0.913 0.01136        0.891        0.935
##   196    564       1    0.911 0.01145        0.889        0.934
##   204    563       1    0.909 0.01155        0.887        0.932
##   206    562       1    0.908 0.01164        0.885        0.931
##   216    561       1    0.906 0.01173        0.883        0.929
##   218    560       1    0.905 0.01182        0.882        0.928
##   219    559       3    0.900 0.01208        0.876        0.924
##   222    555       1    0.898 0.01217        0.875        0.922
##   224    554       1    0.896 0.01226        0.873        0.921
##   226    553       1    0.895 0.01234        0.871        0.919
```

```
##    229    552         1     0.893 0.01242          0.869           0.918
##    230    551         2     0.890 0.01259          0.866           0.915
##    232    549         1     0.888 0.01267          0.864           0.914
##    235    548         1     0.887 0.01275          0.862           0.912
##    246    547         1     0.885 0.01283          0.860           0.911
##    250    546         1     0.883 0.01291          0.859           0.909
##    253    545         1     0.882 0.01298          0.857           0.908
##    257    544         1     0.880 0.01306          0.855           0.906
##    258    543         1     0.879 0.01314          0.853           0.905
##    260    542         1     0.877 0.01321          0.851           0.903
##    262    541         1     0.875 0.01329          0.850           0.902
##    263    540         1     0.874 0.01336          0.848           0.900
##    271    539         1     0.872 0.01343          0.846           0.899
##    274    538         1     0.871 0.01351          0.844           0.897
##    276    537         1     0.869 0.01358          0.843           0.896
##    279    536         2     0.866 0.01372          0.839           0.893
##    283    534         1     0.864 0.01379          0.837           0.891
##    286    533         2     0.861 0.01393          0.834           0.889
##    290    531         2     0.858 0.01406          0.830           0.886
##    294    529         1     0.856 0.01413          0.829           0.884
##    300    528         1     0.854 0.01420          0.827           0.883
##    313    527         1     0.853 0.01426          0.825           0.881
##    314    526         2     0.849 0.01439          0.822           0.878
##    316    524         1     0.848 0.01445          0.820           0.877
##    323    523         1     0.846 0.01452          0.818           0.875
##    325    522         1     0.845 0.01458          0.816           0.874
##    330    521         2     0.841 0.01470          0.813           0.871
##    333    519         1     0.840 0.01476          0.811           0.869
##    335    518         1     0.838 0.01482          0.810           0.868
##    336    517         1     0.836 0.01488          0.808           0.866
##    337    516         2     0.833 0.01500          0.804           0.863
##    341    514         1     0.832 0.01506          0.803           0.862
##    342    513         1     0.830 0.01512          0.801           0.860
##    343    512         1     0.828 0.01517          0.799           0.859
##    348    511         2     0.825 0.01529          0.796           0.856
##    349    509         1     0.824 0.01534          0.794           0.854
##    352    508         1     0.822 0.01540          0.792           0.853
##    355    507         1     0.820 0.01545          0.791           0.851
##    356    506         3     0.815 0.01561          0.785           0.847
##    362    503         1     0.814 0.01567          0.784           0.845
##    366    502         2     0.811 0.01577          0.780           0.842
##    369    500         1     0.809 0.01582          0.778           0.841
##    370    499         1     0.807 0.01587          0.777           0.839
##    372    498         1     0.806 0.01592          0.775           0.837
##    376    497         1     0.804 0.01597          0.773           0.836
##    380    496         1     0.802 0.01602          0.772           0.834
##    382    495         1     0.801 0.01607          0.770           0.833
##    386    494         1     0.799 0.01612          0.768           0.831
##    389    493         1     0.798 0.01617          0.766           0.830
##    402    492         1     0.796 0.01622          0.765           0.828
##    406    491         1     0.794 0.01627          0.763           0.827
##    413    490         1     0.793 0.01631          0.761           0.825
##    415    489         1     0.791 0.01636          0.760           0.824
##    420    488         1     0.789 0.01641          0.758           0.822
```

```
##    422    487       1    0.788 0.01645        0.756        0.821
##    429    486       1    0.786 0.01650        0.755        0.819
##    430    485       1    0.785 0.01655        0.753        0.818
##    438    484       1    0.783 0.01659        0.751        0.816
##    439    483       2    0.780 0.01668        0.748        0.813
##    440    481       1    0.778 0.01672        0.746        0.812
##    443    480       1    0.776 0.01677        0.744        0.810
##    444    479       1    0.775 0.01681        0.743        0.809
##    454    478       1    0.773 0.01685        0.741        0.807
##    458    477       1    0.772 0.01690        0.739        0.805
##    465    476       1    0.770 0.01694        0.738        0.804
##    472    475       1    0.768 0.01698        0.736        0.802
##    474    474       1    0.767 0.01702        0.734        0.801
##    475    473       1    0.765 0.01706        0.732        0.799
##    476    472       1    0.764 0.01710        0.731        0.798
##    482    471       1    0.762 0.01714        0.729        0.796
##    486    470       1    0.760 0.01718        0.727        0.795
##    490    468       1    0.759 0.01722        0.726        0.793
##    491    467       1    0.757 0.01726        0.724        0.792
##    498    466       1    0.755 0.01730        0.722        0.790
##    499    465       1    0.754 0.01734        0.721        0.789
##    504    464       1    0.752 0.01738        0.719        0.787
##    505    463       1    0.751 0.01742        0.717        0.785
##    511    462       1    0.749 0.01745        0.715        0.784
##    512    461       1    0.747 0.01749        0.714        0.782
##    513    460       1    0.746 0.01753        0.712        0.781
##    522    459       1    0.744 0.01757        0.710        0.779
##    525    458       1    0.742 0.01760        0.709        0.778
##    527    457       1    0.741 0.01764        0.707        0.776
##    532    456       1    0.739 0.01768        0.705        0.775
##    546    455       1    0.738 0.01771        0.704        0.773
##    548    454       1    0.736 0.01775        0.702        0.772
##    553    453       1    0.734 0.01778        0.700        0.770
##    559    452       1    0.733 0.01782        0.699        0.768
##    560    451       1    0.731 0.01785        0.697        0.767
##    565    450       1    0.729 0.01788        0.695        0.765
##    569    449       1    0.728 0.01792        0.694        0.764
##    573    448       3    0.723 0.01802        0.688        0.759
##    578    445       1    0.721 0.01805        0.687        0.758
##    580    444       1    0.720 0.01808        0.685        0.756
##    582    443       1    0.718 0.01811        0.683        0.754
##    583    442       1    0.716 0.01815        0.682        0.753
##    589    441       1    0.715 0.01818        0.680        0.751
##    593    440       1    0.713 0.01821        0.678        0.750
##    599    439       1    0.712 0.01824        0.677        0.748
##    602    438       2    0.708 0.01830        0.673        0.745
##    608    436       1    0.707 0.01833        0.672        0.744
##    613    435       1    0.705 0.01836        0.670        0.742
##    615    434       1    0.703 0.01839        0.668        0.740
##    628    433       1    0.702 0.01842        0.667        0.739
##    629    431       1    0.700 0.01845        0.665        0.737
##    638    430       1    0.699 0.01848        0.663        0.736
##    642    429       1    0.697 0.01851        0.662        0.734
##    643    428       1    0.695 0.01853        0.660        0.733
```

```
##   647    427          1    0.694 0.01856          0.658          0.731
##   653    426          1    0.692 0.01859          0.657          0.729
##   654    425          1    0.690 0.01862          0.655          0.728
##   663    424          1    0.689 0.01864          0.653          0.726
##   664    423          1    0.687 0.01867          0.652          0.725
##   668    422          1    0.686 0.01870          0.650          0.723
##   669    421          1    0.684 0.01872          0.648          0.722
##   672    420          1    0.682 0.01875          0.646          0.720
##   675    419          2    0.679 0.01880          0.643          0.717
##   678    417          1    0.677 0.01883          0.641          0.715
##   680    416          1    0.676 0.01885          0.640          0.714
##   684    415          1    0.674 0.01888          0.638          0.712
##   697    414          1    0.672 0.01890          0.636          0.711
##   706    413          1    0.671 0.01893          0.635          0.709
##   708    412          1    0.669 0.01895          0.633          0.707
##   709    411          1    0.668 0.01897          0.631          0.706
##   717    410          1    0.666 0.01900          0.630          0.704
##   720    409          1    0.664 0.01902          0.628          0.703
##   723    408          1    0.663 0.01904          0.626          0.701
##   729    407          1    0.661 0.01907          0.625          0.700
##   730    406          1    0.659 0.01909          0.623          0.698
##   739    405          1    0.658 0.01911          0.621          0.696
##   742    404          1    0.656 0.01913          0.620          0.695
##   743    403          1    0.655 0.01915          0.618          0.693
##   751    402          1    0.653 0.01918          0.616          0.692
##   755    401          1    0.651 0.01920          0.615          0.690
##   759    400          2    0.648 0.01924          0.611          0.687
##   764    398          1    0.646 0.01926          0.610          0.685
##   766    397          1    0.645 0.01928          0.608          0.684
##   795    396          1    0.643 0.01930          0.606          0.682
##   797    395          2    0.640 0.01934          0.603          0.679
##   806    393          1    0.638 0.01936          0.601          0.677
##   828    392          1    0.637 0.01938          0.600          0.676
##   833    391          1    0.635 0.01939          0.598          0.674
##   846    390          1    0.633 0.01941          0.596          0.673
##   851    389          1    0.632 0.01943          0.595          0.671
##   858    388          1    0.630 0.01945          0.593          0.669
##   875    387          1    0.629 0.01947          0.592          0.668
##   883    386          1    0.627 0.01948          0.590          0.666
##   885    385          1    0.625 0.01950          0.588          0.665
##   890    384          1    0.624 0.01952          0.587          0.663
##   891    383          1    0.622 0.01954          0.585          0.662
##   900    382          1    0.620 0.01955          0.583          0.660
##   902    381          1    0.619 0.01957          0.582          0.658
##   905    380          1    0.617 0.01959          0.580          0.657
##   909    379          1    0.615 0.01960          0.578          0.655
##   922    378          1    0.614 0.01962          0.577          0.654
##   931    377          1    0.612 0.01963          0.575          0.652
##   938    376          1    0.611 0.01965          0.573          0.650
##   939    374          1    0.609 0.01966          0.572          0.649
##   940    373          1    0.607 0.01968          0.570          0.647
##   942    372          1    0.606 0.01969          0.568          0.646
##   944    371          1    0.604 0.01971          0.567          0.644
##   952    370          1    0.602 0.01972          0.565          0.642
```

```
##   959    369         1     0.601 0.01973          0.563          0.641
##   960    368         1     0.599 0.01975          0.562          0.639
##   961    367         2     0.596 0.01978          0.558          0.636
##   968    365         1     0.594 0.01979          0.557          0.634
##   969    364         1     0.593 0.01980          0.555          0.633
##   986    363         1     0.591 0.01981          0.553          0.631
##   997    362         2     0.588 0.01984          0.550          0.628
##  1013    360         1     0.586 0.01985          0.548          0.626
##  1018    359         1     0.584 0.01986          0.547          0.625
##  1026    358         1     0.583 0.01987          0.545          0.623
##  1029    357         1     0.581 0.01989          0.544          0.622
##  1034    356         1     0.580 0.01990          0.542          0.620
##  1037    355         1     0.578 0.01991          0.540          0.618
##  1041    354         1     0.576 0.01992          0.539          0.617
##  1046    353         1     0.575 0.01993          0.537          0.615
##  1052    352         1     0.573 0.01994          0.535          0.614
##  1055    351         1     0.571 0.01995          0.534          0.612
##  1061    350         1     0.570 0.01996          0.532          0.610
##  1092    349         1     0.568 0.01997          0.530          0.609
##  1103    348         1     0.567 0.01998          0.529          0.607
##  1105    346         1     0.565 0.01999          0.527          0.605
##  1108    345         1     0.563 0.01999          0.525          0.604
##  1112    344         1     0.562 0.02000          0.524          0.602
##  1114    343         1     0.560 0.02001          0.522          0.601
##  1117    342         1     0.558 0.02002          0.520          0.599
##  1122    341         1     0.557 0.02003          0.519          0.597
##  1135    340         1     0.555 0.02004          0.517          0.596
##  1145    339         1     0.553 0.02004          0.516          0.594
##  1154    338         1     0.552 0.02005          0.514          0.593
##  1161    337         1     0.550 0.02006          0.512          0.591
##  1178    336         1     0.549 0.02007          0.511          0.589
##  1183    335         1     0.547 0.02007          0.509          0.588
##  1186    334         1     0.545 0.02008          0.507          0.586
##  1191    333         1     0.544 0.02009          0.506          0.584
##  1207    332         1     0.542 0.02009          0.504          0.583
##  1211    331         1     0.540 0.02010          0.502          0.581
##  1215    330         1     0.539 0.02010          0.501          0.580
##  1219    329         1     0.537 0.02011          0.499          0.578
##  1252    328         1     0.535 0.02011          0.497          0.576
##  1262    327         1     0.534 0.02012          0.496          0.575
##  1275    326         1     0.532 0.02012          0.494          0.573
##  1295    325         1     0.531 0.02013          0.492          0.571
##  1298    324         1     0.529 0.02013          0.491          0.570
##  1325    323         1     0.527 0.02014          0.489          0.568
##  1399    322         1     0.526 0.02014          0.488          0.567
##  1405    321         1     0.524 0.02014          0.486          0.565
##  1434    320         1     0.522 0.02015          0.484          0.563
##  1471    319         1     0.521 0.02015          0.483          0.562
##  1509    318         1     0.519 0.02015          0.481          0.560
##  1540    315         1     0.517 0.02016          0.479          0.558
##  1548    314         1     0.516 0.02016          0.478          0.557
##  1551    312         1     0.514 0.02016          0.476          0.555
##  1561    311         1     0.512 0.02017          0.474          0.554
##  1564    310         1     0.511 0.02017          0.473          0.552
```

```
## 1568    309       1    0.509 0.02017        0.471        0.550
## 1589    308       1    0.507 0.02017        0.469        0.549
## 1606    307       1    0.506 0.02018        0.468        0.547
## 1647    306       1    0.504 0.02018        0.466        0.545
## 1652    305       1    0.503 0.02018        0.464        0.544
## 1687    304       1    0.501 0.02018        0.463        0.542
## 1709    303       1    0.499 0.02018        0.461        0.540
## 1768    302       1    0.498 0.02018        0.460        0.539
## 1829    297       1    0.496 0.02018        0.458        0.537
## 1839    294       1    0.494 0.02018        0.456        0.535
## 1850    293       1    0.493 0.02019        0.455        0.534
## 1851    292       1    0.491 0.02019        0.453        0.532
## 1879    289       1    0.489 0.02019        0.451        0.530
## 1885    286       1    0.487 0.02019        0.449        0.529
## 1895    285       1    0.486 0.02019        0.448        0.527
## 1918    282       1    0.484 0.02019        0.446        0.525
## 1932    280       1    0.482 0.02020        0.444        0.524
## 1976    273       1    0.480 0.02020        0.442        0.522
## 2012    264       1    0.479 0.02020        0.441        0.520
## 2018    263       1    0.477 0.02021        0.439        0.518
## 2023    262       1    0.475 0.02021        0.437        0.516
## 2067    253       1    0.473 0.02022        0.435        0.514
## 2079    250       1    0.471 0.02023        0.433        0.513
## 2128    236       1    0.469 0.02024        0.431        0.511
## 2152    221       1    0.467 0.02026        0.429        0.509
## 2171    214       1    0.465 0.02028        0.427        0.506
## 2231    188       1    0.462 0.02033        0.424        0.504
## 2458    119       1    0.459 0.02052        0.420        0.501
## 2593     76       1    0.453 0.02112        0.413        0.496
## 2683     60       1    0.445 0.02208        0.404        0.490
## 2718     47       1    0.436 0.02355        0.392        0.484
## 2910     16       1    0.408 0.03438        0.346        0.482
##
##                 rx=Lev+5FU
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     8    608       1    0.998 0.00164        0.995        1.000
##     9    607       1    0.997 0.00232        0.992        1.000
##    23    606       1    0.995 0.00284        0.990        1.000
##    34    604       1    0.993 0.00328        0.987        1.000
##    40    603       1    0.992 0.00367        0.985        0.999
##    45    602       1    0.990 0.00401        0.982        0.998
##    49    600       1    0.988 0.00433        0.980        0.997
##    52    599       1    0.987 0.00463        0.978        0.996
##    63    598       1    0.985 0.00491        0.976        0.995
##    68    597       1    0.984 0.00517        0.973        0.994
##    79    596       1    0.982 0.00542        0.971        0.993
##    86    595       1    0.980 0.00565        0.969        0.991
##    91    594       1    0.979 0.00588        0.967        0.990
##   101    593       1    0.977 0.00610        0.965        0.989
##   116    592       1    0.975 0.00631        0.963        0.988
##   127    591       1    0.974 0.00651        0.961        0.986
##   132    590       1    0.972 0.00670        0.959        0.985
##   134    589       1    0.970 0.00689        0.957        0.984
##   138    588       1    0.969 0.00707        0.955        0.983
```

```
##   141    587       1    0.967 0.00725        0.953         0.981
##   144    586       1    0.965 0.00742        0.951         0.980
##   146    585       1    0.964 0.00759        0.949         0.979
##   154    584       1    0.962 0.00776        0.947         0.977
##   157    583       1    0.960 0.00792        0.945         0.976
##   160    582       1    0.959 0.00807        0.943         0.975
##   161    581       1    0.957 0.00823        0.941         0.973
##   165    580       1    0.955 0.00838        0.939         0.972
##   168    579       1    0.954 0.00852        0.937         0.971
##   183    578       1    0.952 0.00867        0.935         0.969
##   185    577       1    0.951 0.00881        0.933         0.968
##   186    576       2    0.947 0.00908        0.930         0.965
##   198    574       1    0.946 0.00921        0.928         0.964
##   199    573       1    0.944 0.00934        0.926         0.962
##   205    572       1    0.942 0.00947        0.924         0.961
##   208    571       1    0.941 0.00960        0.922         0.960
##   215    570       1    0.939 0.00972        0.920         0.958
##   218    569       1    0.937 0.00984        0.918         0.957
##   237    568       1    0.936 0.00996        0.916         0.955
##   242    567       1    0.934 0.01008        0.914         0.954
##   245    566       1    0.932 0.01020        0.913         0.953
##   248    565       1    0.931 0.01031        0.911         0.951
##   251    564       1    0.929 0.01043        0.909         0.950
##   252    563       1    0.927 0.01054        0.907         0.948
##   255    562       1    0.926 0.01065        0.905         0.947
##   256    561       1    0.924 0.01076        0.903         0.945
##   260    560       1    0.922 0.01086        0.901         0.944
##   261    559       1    0.921 0.01097        0.900         0.943
##   269    558       1    0.919 0.01107        0.898         0.941
##   271    557       1    0.918 0.01117        0.896         0.940
##   274    556       1    0.916 0.01127        0.894         0.938
##   276    555       1    0.914 0.01137        0.892         0.937
##   279    554       1    0.913 0.01147        0.890         0.935
##   283    553       1    0.911 0.01157        0.889         0.934
##   285    552       1    0.909 0.01167        0.887         0.932
##   293    551       1    0.908 0.01176        0.885         0.931
##   296    550       1    0.906 0.01186        0.883         0.929
##   302    549       1    0.904 0.01195        0.881         0.928
##   303    548       1    0.903 0.01204        0.879         0.927
##   304    547       1    0.901 0.01213        0.878         0.925
##   315    546       1    0.899 0.01222        0.876         0.924
##   322    545       2    0.896 0.01240        0.872         0.921
##   324    543       1    0.894 0.01248        0.870         0.919
##   326    542       1    0.893 0.01257        0.868         0.918
##   328    540       1    0.891 0.01265        0.867         0.916
##   329    539       1    0.889 0.01274        0.865         0.915
##   336    538       1    0.888 0.01282        0.863         0.913
##   340    537       1    0.886 0.01290        0.861         0.912
##   355    535       1    0.884 0.01298        0.859         0.910
##   360    533       1    0.883 0.01307        0.858         0.909
##   363    532       1    0.881 0.01315        0.856         0.907
##   365    531       1    0.880 0.01323        0.854         0.906
##   380    530       1    0.878 0.01330        0.852         0.904
##   386    529       1    0.876 0.01338        0.850         0.903
```

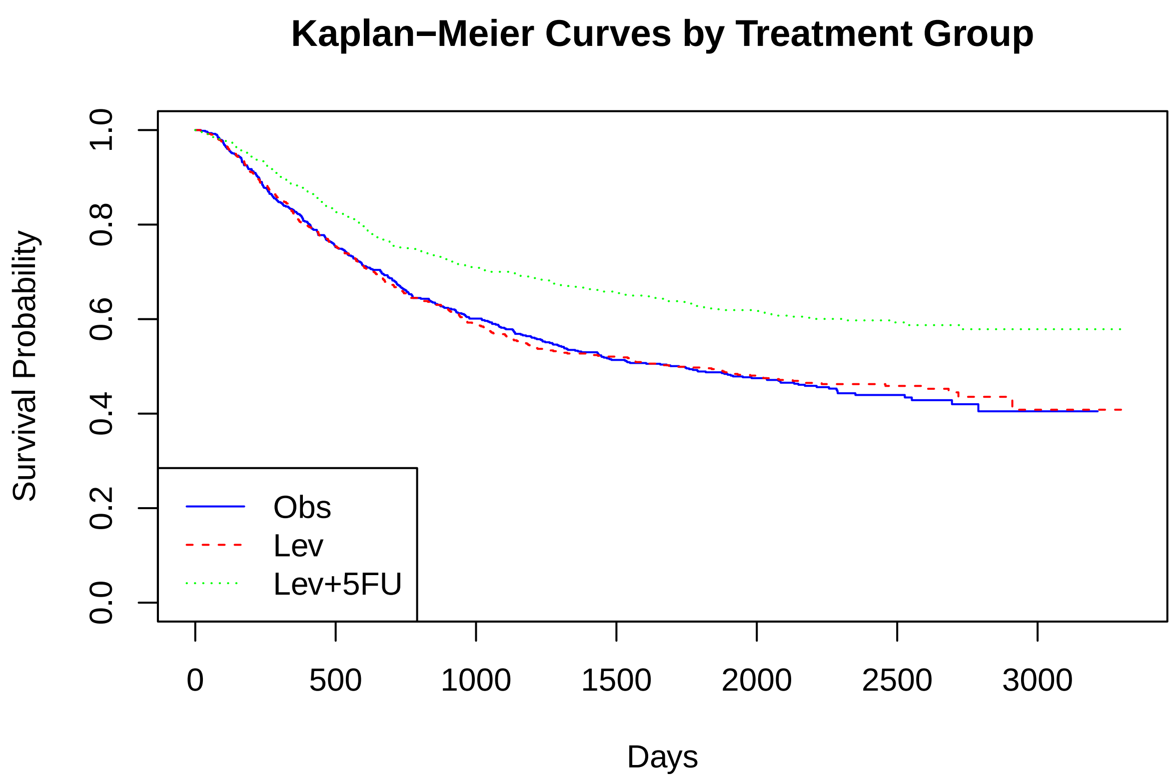
```
##   389    528         1    0.875 0.01346         0.849          0.901
##   392    527         1    0.873 0.01354         0.847          0.900
##   393    526         1    0.871 0.01361         0.845          0.898
##   400    525         1    0.870 0.01369         0.843          0.897
##   405    524         1    0.868 0.01376         0.841          0.895
##   408    523         1    0.866 0.01383         0.840          0.894
##   415    522         1    0.865 0.01391         0.838          0.892
##   422    521         1    0.863 0.01398         0.836          0.891
##   428    520         1    0.861 0.01405         0.834          0.889
##   430    519         1    0.860 0.01412         0.832          0.888
##   431    518         1    0.858 0.01419         0.831          0.886
##   434    517         1    0.856 0.01426         0.829          0.885
##   441    516         1    0.855 0.01433         0.827          0.883
##   443    515         1    0.853 0.01440         0.825          0.882
##   448    514         2    0.850 0.01453         0.822          0.879
##   449    512         1    0.848 0.01460         0.820          0.877
##   454    511         2    0.845 0.01473         0.816          0.874
##   458    509         1    0.843 0.01479         0.815          0.872
##   460    508         1    0.841 0.01485         0.813          0.871
##   466    507         2    0.838 0.01498         0.809          0.868
##   484    505         1    0.836 0.01504         0.807          0.866
##   485    504         1    0.835 0.01510         0.806          0.865
##   490    503         1    0.833 0.01516         0.804          0.863
##   491    502         1    0.831 0.01522         0.802          0.862
##   497    501         1    0.830 0.01528         0.800          0.860
##   498    500         1    0.828 0.01534         0.799          0.859
##   499    499         1    0.826 0.01540         0.797          0.857
##   503    498         1    0.825 0.01546         0.795          0.856
##   510    497         1    0.823 0.01552         0.793          0.854
##   526    496         1    0.821 0.01558         0.791          0.853
##   529    495         1    0.820 0.01563         0.790          0.851
##   536    494         1    0.818 0.01569         0.788          0.849
##   543    493         1    0.816 0.01574         0.786          0.848
##   550    492         1    0.815 0.01580         0.784          0.846
##   554    491         2    0.811 0.01591         0.781          0.843
##   576    489         1    0.810 0.01596         0.779          0.842
##   578    488         2    0.806 0.01607         0.776          0.839
##   580    486         1    0.805 0.01612         0.774          0.837
##   583    485         1    0.803 0.01617         0.772          0.836
##   591    484         1    0.802 0.01622         0.770          0.834
##   592    483         1    0.800 0.01628         0.769          0.832
##   593    482         1    0.798 0.01633         0.767          0.831
##   594    481         1    0.797 0.01638         0.765          0.829
##   601    480         1    0.795 0.01643         0.763          0.828
##   602    479         1    0.793 0.01647         0.762          0.826
##   603    478         1    0.792 0.01652         0.760          0.825
##   604    477         1    0.790 0.01657         0.758          0.823
##   609    476         1    0.788 0.01662         0.756          0.822
##   614    475         1    0.787 0.01667         0.755          0.820
##   616    474         2    0.783 0.01676         0.751          0.817
##   617    472         1    0.782 0.01681         0.749          0.815
##   622    471         1    0.780 0.01685         0.748          0.814
##   636    470         1    0.778 0.01690         0.746          0.812
##   641    469         1    0.777 0.01695         0.744          0.811
```

```
##   642    468        1    0.775 0.01699        0.742         0.809
##   643    467        1    0.773 0.01703        0.741         0.807
##   649    466        1    0.772 0.01708        0.739         0.806
##   657    465        1    0.770 0.01712        0.737         0.804
##   666    464        1    0.768 0.01717        0.735         0.803
##   674    463        1    0.767 0.01721        0.734         0.801
##   683    462        1    0.765 0.01725        0.732         0.800
##   692    461        2    0.762 0.01734        0.728         0.796
##   693    459        1    0.760 0.01738        0.727         0.795
##   696    458        1    0.758 0.01742        0.725         0.793
##   700    457        1    0.757 0.01746        0.723         0.792
##   701    456        1    0.755 0.01750        0.722         0.790
##   711    455        1    0.753 0.01754        0.720         0.789
##   712    454        1    0.752 0.01758        0.718         0.787
##   736    453        1    0.750 0.01762        0.716         0.785
##   765    452        1    0.748 0.01766        0.715         0.784
##   802    451        2    0.745 0.01773        0.711         0.781
##   805    449        1    0.743 0.01777        0.709         0.779
##   806    448        1    0.742 0.01781        0.708         0.778
##   811    447        1    0.740 0.01785        0.706         0.776
##   827    446        1    0.738 0.01788        0.704         0.774
##   844    445        1    0.737 0.01792        0.702         0.773
##   849    443        1    0.735 0.01796        0.701         0.771
##   853    442        1    0.733 0.01799        0.699         0.770
##   862    441        1    0.732 0.01803        0.697         0.768
##   884    440        1    0.730 0.01807        0.696         0.766
##   887    438        2    0.727 0.01814        0.692         0.763
##   904    436        1    0.725 0.01817        0.690         0.762
##   905    435        1    0.723 0.01821        0.689         0.760
##   911    434        1    0.722 0.01824        0.687         0.758
##   916    433        1    0.720 0.01827        0.685         0.757
##   918    432        1    0.718 0.01831        0.683         0.755
##   934    431        1    0.717 0.01834        0.682         0.754
##   936    430        1    0.715 0.01837        0.680         0.752
##   961    429        1    0.713 0.01841        0.678         0.750
##   968    428        1    0.712 0.01844        0.677         0.749
##   977    427        1    0.710 0.01847        0.675         0.747
##   993    426        1    0.708 0.01850        0.673         0.746
##  1022    425        1    0.707 0.01853        0.671         0.744
##  1024    424        1    0.705 0.01856        0.670         0.742
##  1025    423        1    0.703 0.01860        0.668         0.741
##  1032    422        1    0.702 0.01863        0.666         0.739
##  1037    421        1    0.700 0.01866        0.665         0.738
##  1122    420        1    0.698 0.01869        0.663         0.736
##  1138    419        1    0.697 0.01872        0.661         0.734
##  1142    418        1    0.695 0.01874        0.659         0.733
##  1145    417        1    0.693 0.01877        0.658         0.731
##  1151    416        1    0.692 0.01880        0.656         0.730
##  1159    415        1    0.690 0.01883        0.654         0.728
##  1193    414        1    0.688 0.01886        0.652         0.726
##  1201    413        1    0.687 0.01889        0.651         0.725
##  1212    412        1    0.685 0.01891        0.649         0.723
##  1233    411        1    0.683 0.01894        0.647         0.722
##  1246    410        1    0.682 0.01897        0.646         0.720
```

```
## 1273   409      1   0.680 0.01900        0.644        0.718
## 1276   408      2   0.677 0.01905        0.640        0.715
## 1277   406      1   0.675 0.01907        0.639        0.714
## 1279   405      1   0.673 0.01910        0.637        0.712
## 1302   402      1   0.672 0.01913        0.635        0.710
## 1306   401      1   0.670 0.01915        0.634        0.709
## 1329   400      1   0.668 0.01918        0.632        0.707
## 1365   399      1   0.667 0.01920        0.630        0.705
## 1387   398      1   0.665 0.01923        0.628        0.704
## 1388   397      1   0.663 0.01925        0.627        0.702
## 1424   394      1   0.662 0.01927        0.625        0.701
## 1439   393      1   0.660 0.01930        0.623        0.699
## 1446   392      1   0.658 0.01932        0.622        0.697
## 1488   389      1   0.657 0.01935        0.620        0.696
## 1495   388      1   0.655 0.01937        0.618        0.694
## 1511   387      1   0.653 0.01940        0.616        0.692
## 1521   386      1   0.652 0.01942        0.615        0.691
## 1550   385      1   0.650 0.01944        0.613        0.689
## 1607   383      1   0.648 0.01947        0.611        0.688
## 1620   381      1   0.647 0.01949        0.609        0.686
## 1637   380      1   0.645 0.01951        0.608        0.684
## 1644   379      1   0.643 0.01953        0.606        0.683
## 1668   378      2   0.640 0.01958        0.602        0.679
## 1671   376      1   0.638 0.01960        0.601        0.678
## 1723   375      1   0.636 0.01962        0.599        0.676
## 1743   374      1   0.635 0.01964        0.597        0.674
## 1752   373      1   0.633 0.01966        0.596        0.673
## 1767   372      1   0.631 0.01968        0.594        0.671
## 1783   371      1   0.629 0.01970        0.592        0.669
## 1786   370      1   0.628 0.01972        0.590        0.668
## 1798   369      1   0.626 0.01974        0.589        0.666
## 1812   366      1   0.624 0.01976        0.587        0.664
## 1831   357      1   0.623 0.01979        0.585        0.663
## 1856   353      1   0.621 0.01981        0.583        0.661
## 1876   347      1   0.619 0.01983        0.581        0.659
## 1995   331      1   0.617 0.01986        0.579        0.657
## 2021   322      1   0.615 0.01989        0.578        0.656
## 2028   319      1   0.613 0.01992        0.576        0.654
## 2031   316      1   0.611 0.01995        0.574        0.652
## 2052   309      1   0.609 0.01999        0.572        0.650
## 2074   299      1   0.607 0.02002        0.569        0.648
## 2127   279      1   0.605 0.02007        0.567        0.646
## 2174   261      1   0.603 0.02013        0.565        0.644
## 2197   243      1   0.600 0.02020        0.562        0.641
## 2318   194      1   0.597 0.02033        0.559        0.639
## 2482   139      1   0.593 0.02063        0.554        0.635
## 2542   103      1   0.587 0.02122        0.547        0.630
## 2725    68      1   0.579 0.02260        0.536        0.625
```

```r
plot(fit_km_rx,
     xlab = "Days",
     ylab = "Survival Probability",
     col = c("blue", "red", "green"),
     lty = 1:3,
```

```
     main = "Kaplan-Meier Curves by Treatment Group")

legend("bottomleft",
       legend = levels(colon_data$rx),
       col = c("blue", "red", "green"),
       lty = 1:3)
```

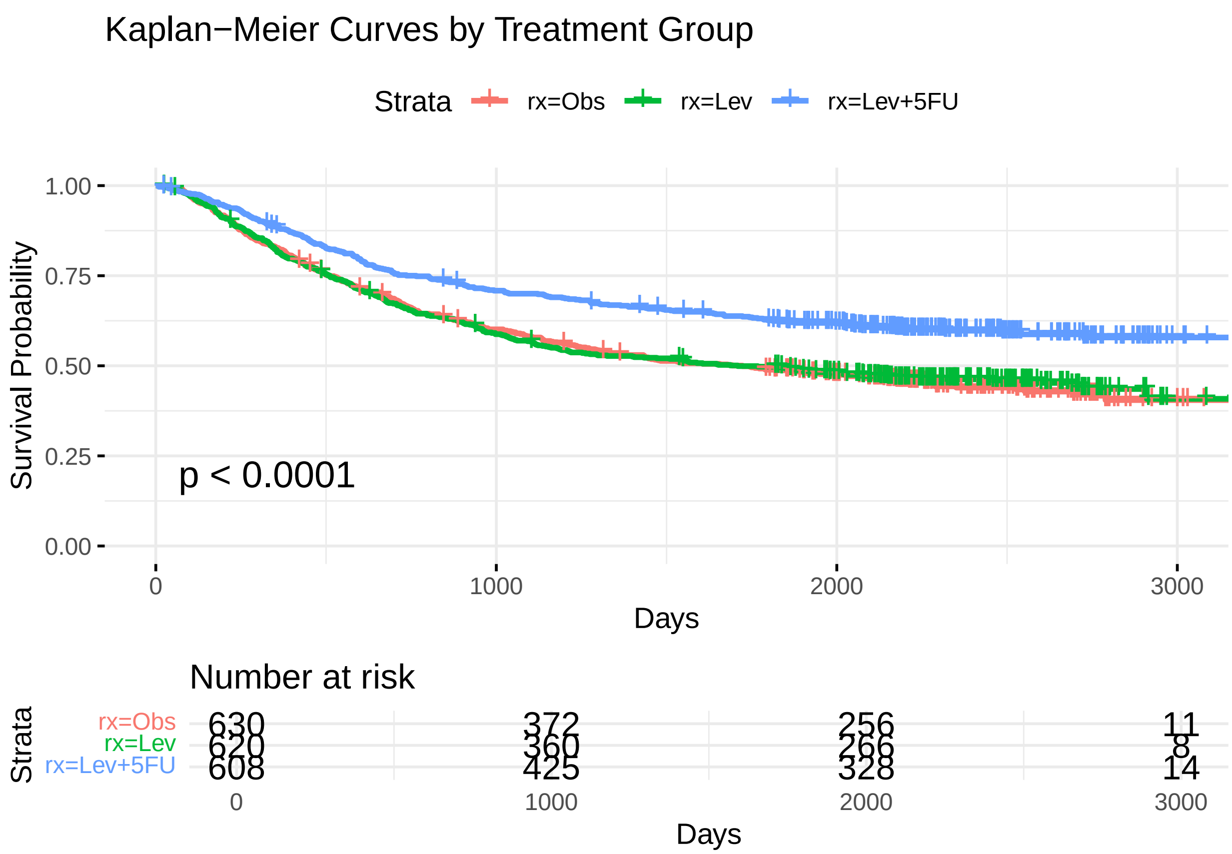



```
library(survminer)
```

```
## Loading required package: ggpubr

##
## Attaching package: 'survminer'

## The following object is masked from 'package:survival':
##
##     myeloma
```

```
ggsurvplot(
  fit = fit_km_rx,
  data = colon_data,
  risk.table = TRUE,       # show number at risk table
  pval = TRUE,             # show p-value from log-rank test
```

```
  ggtheme = theme_minimal(),
  title = "Kaplan-Meier Curves by Treatment Group",
  xlab = "Days",
  ylab = "Survival Probability"
)
```




```
logrank_test_rx <- survdiff(surv_obj ~ rx, data = colon_data)
logrank_test_rx
```

```
## Call:
## survdiff(formula = surv_obj ~ rx, data = colon_data)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## rx=Obs       630      345      299      7.01     10.40
## rx=Lev       620      333      295      4.93      7.26
## rx=Lev+5FU 608      242      326     21.61     33.54
##
##  Chisq= 33.6  on 2 degrees of freedom, p= 5e-08
```

Kaplan-Meier Curves by Treatment

KM Plots (Base R and survminer)

- **Lev+5FU (green/dotted)**: Appears to have the highest survival curve over time.

- **Obs (blue/solid)**: Seems to have the lowest curve.
- **Lev (red/dashed)**: Is in between.

Survival Probabilities (from the survfit summary)

- For each time point, you see columns for survival, std.err, and 95% CI.
- At early times (like ~50 days), survival is ~0.99 or higher across groups.
- By later times (~3000 days), survival probabilities diverge, with Lev+5FU typically retaining higher survival.

Interpretation

- Visually, Lev+5FU group experiences fewer (or later) events, while the Obs group experiences more (or earlier) events.
- This pattern suggests Lev+5FU might be more effective at delaying the event (recurrence or death).

Log-Rank Test

- **Chi-square = 33.6 on 2 degrees of freedom, p = 5e-08 (~0.00000005)**
  - This is a highly significant result (< 0.0001), indicating that at least one survival curve differs from the others.

From the table:

Group	N (Total)	Observed Events	Expected Events
Obs	630	345	299
Lev	620	333	295
Lev+5FU	608	242	286

- **Obs**: Has more events than expected (345 vs. 299), consistent with lower survival.
- **Lev+5FU**: Has fewer events than expected (242 vs. 286), consistent with better survival.
- **Lev**: Is intermediate.

Interpretation

- The p-value is extremely small, so we can reject the null hypothesis that all three groups have the same survival function.
- In plain language: Treatment significantly affects time to event (the combined outcome).

Practical Takeaways

- **Lev+5FU**: Appears to provide a survival advantage, while Obs is least effective.
- **Lev**: Is in between but closer to Obs, based on the shape of the curves and the number of events.

Manual Pairwise Log-Rank Tests

```r
# Pair 1: Obs vs Lev
survdiff(Surv(time, status) ~ rx,
         data = colon_data,
         subset = (rx %in% c("Obs", "Lev")))
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ rx, data = colon_data,
##     subset = (rx %in% c("Obs", "Lev")))
##
##          N Observed Expected (O-E)^2/E (O-E)^2/V
## rx=Obs 630      345      341    0.0372     0.075
## rx=Lev 620      333      337    0.0377     0.075
##
##  Chisq= 0.1  on 1 degrees of freedom, p= 0.8
```

```r
# Pair 2: Obs vs Lev+5FU
survdiff(Surv(time, status) ~ rx,
         data = colon_data,
         subset = (rx %in% c("Obs", "Lev+5FU")))
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ rx, data = colon_data,
##     subset = (rx %in% c("Obs", "Lev+5FU")))
##
##              N Observed Expected (O-E)^2/E (O-E)^2/V
## rx=Obs     630      345      281      14.7      28.2
## rx=Lev+5FU 608      242      306      13.5      28.2
##
##  Chisq= 28.2  on 1 degrees of freedom, p= 1e-07
```

```r
# Pair 3: Lev vs Lev+5FU
survdiff(Surv(time, status) ~ rx,
         data = colon_data,
         subset = (rx %in% c("Lev", "Lev+5FU")))
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ rx, data = colon_data,
##     subset = (rx %in% c("Lev", "Lev+5FU")))
##
##              N Observed Expected (O-E)^2/E (O-E)^2/V
## rx=Lev     620      333      273      13.0      24.8
## rx=Lev+5FU 608      242      302      11.8      24.8
##
##  Chisq= 24.8  on 1 degrees of freedom, p= 6e-07
```

Pairwise Comparisons of Treatment Groups

Obs vs Lev

- **p = 0.8**: Suggests no statistical difference in survival curves between Observation alone and Levamisole (Lev).
- **Interpretation**: Visually or clinically, these two groups appear quite similar in how quickly events occur.

Obs vs Lev+5FU

- $\mathbf{p = 1 \times 10}$: Indicates a highly significant difference.
- **Interpretation**: Lev+5FU outperforms Obs, implying better survival (fewer/farther events).

Lev vs Lev+5FU

- $\mathbf{p = 6 \times 10}$: Also indicates a highly significant difference.
- **Interpretation**: Lev+5FU outperforms Lev, again suggesting better survival.

Summary

- **Lev+5FU**: Significantly better than both Obs and Lev, suggesting it provides a survival advantage.
- **Obs and Lev**: Not significantly different from each other, indicating similar survival outcomes.

These results highlight the effectiveness of Lev+5FU in improving survival compared to the other treatment groups.

Kaplan-Meier Curves (Time to Death)

```r
# 1) Summarize time/status for death
colon_death <- colon_data %>%
  group_by(id) %>%
  summarize(
    time = if (any(status == 1 & etype == 2)) {
      min(time[status == 1 & etype == 2])
    } else {
      max(time)
    },
    status = if (any(status == 1 & etype == 2)) 1 else 0
  ) %>%
  ungroup()

# 2) Merge 'rx' from 'colon_data'
colon_death <- colon_death %>%
  left_join(
    select(colon_data, id, rx),
    by = "id"
  ) %>%
  distinct(id, .keep_all = TRUE)

# Now 'colon_death' should have columns: id, time, status, rx, ...
str(colon_death)
```

```
## tibble [929 x 4] (S3: tbl_df/tbl/data.frame)
##  $ id    : num [1:929] 1 2 3 4 5 6 7 8 9 10 ...
##  $ time  : num [1:929] 1521 3087 963 293 659 ...
##  $ status: num [1:929] 1 0 1 1 1 1 1 0 0 0 ...
##  $ rx    : Factor w/ 3 levels "Obs","Lev","Lev+5FU": 3 3 1 3 1 3 2 1 2 3 ...
```

```r
# Surv object for time to death
colon_death$surv_death <- Surv(time = colon_death$time,
                                event = colon_death$status)
```

```r
# Suppose you want to compare survival by rx (Obs, Lev, Lev+5FU)
fit_km_death <- survfit(surv_death ~ rx, data = colon_death)

# Print a summary
summary(fit_km_death)
```

```
## Call: survfit(formula = surv_death ~ rx, data = colon_death)
##
##                 rx=Obs
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   113    315       1    0.997 0.00317        0.991        1.000
##   125    314       1    0.994 0.00448        0.985        1.000
##   145    313       1    0.990 0.00547        0.980        1.000
##   164    312       1    0.987 0.00631        0.975        1.000
##   166    311       1    0.984 0.00704        0.970        0.998
##   187    310       1    0.981 0.00770        0.966        0.996
##   201    309       1    0.978 0.00831        0.962        0.994
##   208    308       1    0.975 0.00886        0.957        0.992
##   215    307       1    0.971 0.00939        0.953        0.990
##   218    306       1    0.968 0.00988        0.949        0.988
##   238    305       1    0.965 0.01034        0.945        0.986
##   241    304       1    0.962 0.01079        0.941        0.983
##   242    303       1    0.959 0.01121        0.937        0.981
##   253    302       1    0.956 0.01161        0.933        0.979
##   259    301       2    0.949 0.01237        0.925        0.974
##   264    299       1    0.946 0.01273        0.921        0.971
##   275    298       1    0.943 0.01308        0.918        0.969
##   289    297       1    0.940 0.01341        0.914        0.966
##   311    296       1    0.937 0.01374        0.910        0.964
##   313    295       1    0.933 0.01405        0.906        0.961
##   322    294       1    0.930 0.01436        0.902        0.959
##   331    293       1    0.927 0.01466        0.899        0.956
##   365    292       1    0.924 0.01495        0.895        0.954
##   372    291       1    0.921 0.01523        0.891        0.951
##   381    290       1    0.917 0.01550        0.888        0.948
##   384    289       2    0.911 0.01603        0.880        0.943
##   390    287       1    0.908 0.01629        0.877        0.940
##   409    286       1    0.905 0.01654        0.873        0.938
##   411    285       1    0.902 0.01678        0.869        0.935
##   413    284       2    0.895 0.01726        0.862        0.930
##   417    282       1    0.892 0.01748        0.858        0.927
##   421    281       1    0.889 0.01771        0.855        0.924
##   433    280       1    0.886 0.01793        0.851        0.922
##   437    279       1    0.883 0.01814        0.848        0.919
##   438    278       1    0.879 0.01835        0.844        0.916
##   459    276       1    0.876 0.01856        0.841        0.913
##   462    275       1    0.873 0.01876        0.837        0.911
##   464    274       1    0.870 0.01896        0.833        0.908
##   465    273       2    0.863 0.01935        0.826        0.902
```

```
##   469    271          1    0.860 0.01954         0.823         0.899
##   474    270          1    0.857 0.01973         0.819         0.897
##   485    269          1    0.854 0.01991         0.816         0.894
##   499    268          1    0.851 0.02009         0.812         0.891
##   506    267          1    0.848 0.02026         0.809         0.888
##   510    266          1    0.844 0.02044         0.805         0.885
##   528    265          1    0.841 0.02061         0.802         0.883
##   537    264          1    0.838 0.02077         0.798         0.880
##   563    263          2    0.832 0.02110         0.791         0.874
##   570    261          1    0.828 0.02126         0.788         0.871
##   576    260          1    0.825 0.02141         0.784         0.868
##   587    259          1    0.822 0.02157         0.781         0.865
##   591    258          1    0.819 0.02172         0.777         0.863
##   594    257          1    0.816 0.02186         0.774         0.860
##   595    256          1    0.812 0.02201         0.770         0.857
##   599    255          1    0.809 0.02215         0.767         0.854
##   612    254          1    0.806 0.02229         0.764         0.851
##   622    253          1    0.803 0.02243         0.760         0.848
##   659    252          1    0.800 0.02257         0.757         0.845
##   663    251          1    0.797 0.02270         0.753         0.842
##   665    250          1    0.793 0.02283         0.750         0.839
##   670    249          1    0.790 0.02296         0.746         0.836
##   673    248          1    0.787 0.02309         0.743         0.834
##   685    247          1    0.784 0.02322         0.740         0.831
##   687    246          1    0.781 0.02334         0.736         0.828
##   692    245          1    0.777 0.02346         0.733         0.825
##   709    244          1    0.774 0.02358         0.729         0.822
##   716    243          1    0.771 0.02370         0.726         0.819
##   717    242          1    0.768 0.02381         0.723         0.816
##   718    241          1    0.765 0.02393         0.719         0.813
##   721    240          1    0.761 0.02404         0.716         0.810
##   743    239          1    0.758 0.02415         0.712         0.807
##   753    238          1    0.755 0.02425         0.709         0.804
##   758    237          1    0.752 0.02436         0.706         0.801
##   760    236          1    0.749 0.02446         0.702         0.798
##   761    235          1    0.746 0.02457         0.699         0.795
##   770    234          1    0.742 0.02467         0.696         0.792
##   774    233          1    0.739 0.02477         0.692         0.789
##   775    232          1    0.736 0.02486         0.689         0.786
##   832    231          1    0.733 0.02496         0.685         0.783
##   833    230          1    0.730 0.02505         0.682         0.780
##   840    229          1    0.726 0.02515         0.679         0.777
##   845    228          1    0.723 0.02524         0.675         0.774
##   854    227          1    0.720 0.02533         0.672         0.771
##   863    226          1    0.717 0.02541         0.669         0.768
##   874    225          1    0.714 0.02550         0.665         0.765
##   883    224          1    0.711 0.02558         0.662         0.762
##   887    223          1    0.707 0.02567         0.659         0.759
##   901    222          1    0.704 0.02575         0.655         0.756
##   924    221          1    0.701 0.02583         0.652         0.753
##   928    220          1    0.698 0.02591         0.649         0.750
##   929    219          1    0.695 0.02598         0.645         0.747
##   936    218          1    0.691 0.02606         0.642         0.744
##   949    217          1    0.688 0.02613         0.639         0.741
```

```
##   957    216       1    0.685 0.02621        0.636        0.738
##   961    215       1    0.682 0.02628        0.632        0.735
##   963    214       1    0.679 0.02635        0.629        0.732
##   966    213       1    0.675 0.02641        0.626        0.729
##   976    212       1    0.672 0.02648        0.622        0.726
##  1021    211       1    0.669 0.02655        0.619        0.723
##  1031    210       1    0.666 0.02661        0.616        0.720
##  1048    209       1    0.663 0.02667        0.612        0.717
##  1070    208       1    0.660 0.02674        0.609        0.714
##  1079    207       1    0.656 0.02680        0.606        0.711
##  1083    206       1    0.653 0.02685        0.603        0.708
##  1101    205       1    0.650 0.02691        0.599        0.705
##  1133    204       1    0.647 0.02697        0.596        0.702
##  1134    203       1    0.644 0.02702        0.593        0.699
##  1136    202       1    0.640 0.02708        0.589        0.696
##  1139    201       1    0.637 0.02713        0.586        0.693
##  1159    200       1    0.634 0.02718        0.583        0.690
##  1166    199       1    0.631 0.02723        0.580        0.687
##  1178    198       1    0.628 0.02728        0.576        0.683
##  1195    197       1    0.624 0.02732        0.573        0.680
##  1198    196       1    0.621 0.02737        0.570        0.677
##  1209    195       1    0.618 0.02741        0.567        0.674
##  1216    194       1    0.615 0.02746        0.563        0.671
##  1230    193       1    0.612 0.02750        0.560        0.668
##  1237    192       1    0.609 0.02754        0.557        0.665
##  1246    191       1    0.605 0.02758        0.554        0.662
##  1262    190       1    0.602 0.02762        0.550        0.659
##  1272    189       1    0.599 0.02765        0.547        0.656
##  1290    188       1    0.596 0.02769        0.544        0.653
##  1295    187       1    0.593 0.02772        0.541        0.650
##  1304    186       1    0.589 0.02776        0.537        0.646
##  1313    185       1    0.586 0.02779        0.534        0.643
##  1314    184       1    0.583 0.02782        0.531        0.640
##  1327    183       1    0.580 0.02785        0.528        0.637
##  1363    182       1    0.577 0.02788        0.525        0.634
##  1375    181       1    0.573 0.02791        0.521        0.631
##  1434    180       1    0.570 0.02793        0.518        0.628
##  1437    179       1    0.567 0.02796        0.515        0.625
##  1447    178       1    0.564 0.02798        0.512        0.622
##  1482    177       1    0.561 0.02800        0.508        0.618
##  1530    176       1    0.558 0.02803        0.505        0.615
##  1548    175       1    0.554 0.02805        0.502        0.612
##  1656    174       1    0.551 0.02807        0.499        0.609
##  1679    173       1    0.548 0.02808        0.496        0.606
##  1692    172       1    0.545 0.02810        0.492        0.603
##  1723    171       1    0.542 0.02812        0.489        0.600
##  1745    170       1    0.538 0.02813        0.486        0.597
##  1772    169       1    0.535 0.02814        0.483        0.593
##  1788    168       1    0.532 0.02816        0.480        0.590
##  1790    167       1    0.529 0.02817        0.476        0.587
##  1818    164       1    0.526 0.02818        0.473        0.584
##  1875    156       1    0.522 0.02820        0.470        0.581
##  1884    155       1    0.519 0.02822        0.466        0.577
##  1896    153       1    0.516 0.02824        0.463        0.574
```

```
## 1907    150       1    0.512 0.02826        0.460        0.571
## 1915    149       1    0.509 0.02828        0.456        0.567
## 1950    146       1    0.505 0.02830        0.453        0.564
## 2077    135       1    0.501 0.02833        0.449        0.560
## 2083    134       1    0.498 0.02837        0.445        0.557
## 2085    133       1    0.494 0.02840        0.441        0.553
## 2133    121       1    0.490 0.02846        0.437        0.549
## 2171    109       1    0.485 0.02855        0.433        0.545
## 2213     94       1    0.480 0.02871        0.427        0.540
## 2257     83       1    0.474 0.02894        0.421        0.535
## 2284     77       1    0.468 0.02921        0.414        0.529
## 2287     76       1    0.462 0.02947        0.408        0.524
## 2351     67       1    0.455 0.02983        0.400        0.518
## 2527     47       1    0.446 0.03072        0.389        0.510
## 2552     42       1    0.435 0.03177        0.377        0.502
## 2789     16       1    0.408 0.03975        0.337        0.494
##
##                 rx=Lev
## time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   24    310       1    0.997 0.00322        0.990        1.000
##   56    309       1    0.994 0.00455        0.985        1.000
##   93    308       1    0.990 0.00556        0.979        1.000
##  122    307       1    0.987 0.00641        0.975        1.000
##  129    306       1    0.984 0.00715        0.970        0.998
##  133    305       1    0.981 0.00782        0.965        0.996
##  150    304       1    0.977 0.00844        0.961        0.994
##  165    303       1    0.974 0.00901        0.957        0.992
##  171    302       2    0.968 0.01004        0.948        0.988
##  191    300       1    0.965 0.01051        0.944        0.985
##  206    299       1    0.961 0.01096        0.940        0.983
##  219    298       2    0.955 0.01179        0.932        0.978
##  222    296       1    0.952 0.01219        0.928        0.976
##  226    295       1    0.948 0.01257        0.924        0.973
##  232    294       1    0.945 0.01293        0.920        0.971
##  257    293       1    0.942 0.01328        0.916        0.968
##  283    292       1    0.939 0.01362        0.912        0.966
##  314    291       2    0.932 0.01427        0.905        0.961
##  316    289       1    0.929 0.01458        0.901        0.958
##  323    288       1    0.926 0.01489        0.897        0.955
##  342    287       1    0.923 0.01518        0.893        0.953
##  343    286       1    0.919 0.01546        0.890        0.950
##  349    285       1    0.916 0.01574        0.886        0.948
##  355    284       1    0.913 0.01602        0.882        0.945
##  356    283       1    0.910 0.01628        0.878        0.942
##  362    282       1    0.906 0.01654        0.875        0.939
##  366    281       1    0.903 0.01679        0.871        0.937
##  376    280       1    0.900 0.01704        0.867        0.934
##  382    279       1    0.897 0.01728        0.864        0.931
##  402    278       1    0.894 0.01752        0.860        0.929
##  406    277       1    0.890 0.01775        0.856        0.926
##  420    276       1    0.887 0.01797        0.853        0.923
##  422    275       1    0.884 0.01820        0.849        0.920
##  430    274       1    0.881 0.01841        0.845        0.917
##  438    273       1    0.877 0.01863        0.842        0.915
```

```
##   439    272       1    0.874 0.01884        0.838        0.912
##   443    271       1    0.871 0.01904        0.834        0.909
##   444    270       1    0.868 0.01924        0.831        0.906
##   472    269       1    0.865 0.01944        0.827        0.903
##   475    268       1    0.861 0.01963        0.824        0.901
##   486    267       1    0.858 0.01982        0.820        0.898
##   499    266       1    0.855 0.02001        0.817        0.895
##   512    265       1    0.852 0.02019        0.813        0.892
##   522    264       1    0.848 0.02037        0.809        0.889
##   546    263       1    0.845 0.02055        0.806        0.886
##   553    262       1    0.842 0.02072        0.802        0.884
##   559    261       1    0.839 0.02089        0.799        0.881
##   569    260       1    0.835 0.02106        0.795        0.878
##   573    259       1    0.832 0.02122        0.792        0.875
##   580    258       1    0.829 0.02138        0.788        0.872
##   582    257       1    0.826 0.02154        0.785        0.869
##   589    256       1    0.823 0.02170        0.781        0.866
##   602    255       2    0.816 0.02200        0.774        0.860
##   608    253       1    0.813 0.02215        0.771        0.857
##   628    252       1    0.810 0.02230        0.767        0.855
##   629    251       1    0.806 0.02244        0.764        0.852
##   642    250       1    0.803 0.02258        0.760        0.849
##   643    249       1    0.800 0.02272        0.757        0.846
##   647    248       1    0.797 0.02285        0.753        0.843
##   664    247       1    0.794 0.02299        0.750        0.840
##   669    246       1    0.790 0.02312        0.746        0.837
##   675    245       1    0.787 0.02325        0.743        0.834
##   678    244       1    0.784 0.02338        0.739        0.831
##   684    243       1    0.781 0.02350        0.736        0.828
##   706    242       1    0.777 0.02363        0.732        0.825
##   708    241       1    0.774 0.02375        0.729        0.822
##   709    240       1    0.771 0.02387        0.726        0.819
##   720    239       1    0.768 0.02398        0.722        0.816
##   723    238       1    0.765 0.02410        0.719        0.813
##   729    237       1    0.761 0.02421        0.715        0.810
##   730    236       1    0.758 0.02432        0.712        0.807
##   739    235       1    0.755 0.02443        0.708        0.804
##   743    234       1    0.752 0.02454        0.705        0.801
##   755    233       1    0.748 0.02465        0.702        0.798
##   759    232       2    0.742 0.02485        0.695        0.792
##   764    230       1    0.739 0.02495        0.691        0.789
##   766    229       1    0.735 0.02505        0.688        0.786
##   795    228       1    0.732 0.02515        0.685        0.783
##   797    227       1    0.729 0.02524        0.681        0.780
##   806    226       1    0.726 0.02534        0.678        0.777
##   833    225       1    0.723 0.02543        0.674        0.774
##   846    224       1    0.719 0.02552        0.671        0.771
##   858    223       1    0.716 0.02561        0.668        0.768
##   875    222       1    0.713 0.02569        0.664        0.765
##   885    221       1    0.710 0.02578        0.661        0.762
##   890    220       1    0.706 0.02586        0.658        0.759
##   902    219       1    0.703 0.02595        0.654        0.756
##   905    218       1    0.700 0.02603        0.651        0.753
##   909    217       1    0.697 0.02611        0.647        0.750
```

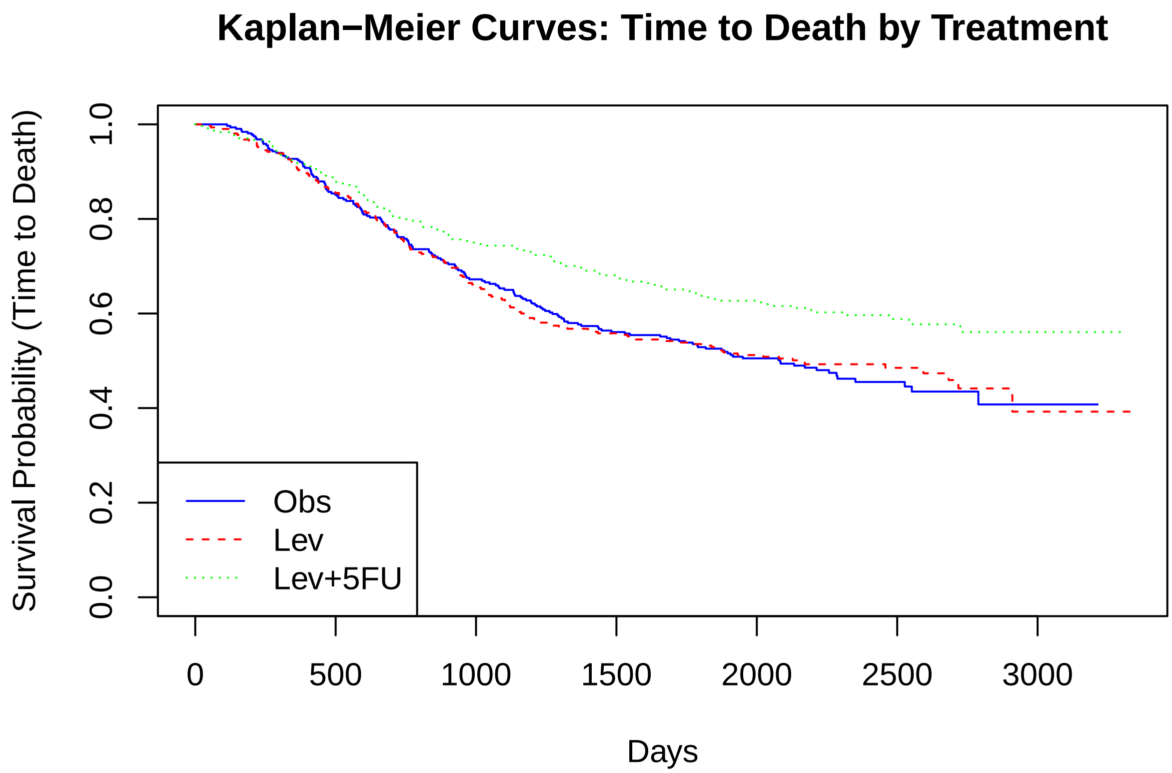
```
##   938    216        1    0.694 0.02618         0.644         0.747
##   939    215        1    0.690 0.02626         0.641         0.744
##   940    214        1    0.687 0.02633         0.637         0.741
##   942    213        1    0.684 0.02641         0.634         0.738
##   944    212        1    0.681 0.02648         0.631         0.735
##   952    211        1    0.677 0.02655         0.627         0.732
##   961    210        2    0.671 0.02669         0.621         0.725
##   968    208        1    0.668 0.02675         0.617         0.722
##   969    207        1    0.665 0.02682         0.614         0.719
##   986    206        1    0.661 0.02688         0.611         0.716
##   997    205        2    0.655 0.02700         0.604         0.710
##  1018    203        1    0.652 0.02706         0.601         0.707
##  1034    202        1    0.648 0.02712         0.597         0.704
##  1037    201        1    0.645 0.02717         0.594         0.701
##  1041    200        1    0.642 0.02723         0.591         0.698
##  1046    199        1    0.639 0.02728         0.587         0.694
##  1055    198        1    0.635 0.02734         0.584         0.691
##  1061    197        1    0.632 0.02739         0.581         0.688
##  1092    196        1    0.629 0.02744         0.577         0.685
##  1103    195        1    0.626 0.02748         0.574         0.682
##  1105    194        1    0.623 0.02753         0.571         0.679
##  1112    193        1    0.619 0.02758         0.568         0.676
##  1117    192        1    0.616 0.02762         0.564         0.673
##  1122    191        1    0.613 0.02766         0.561         0.670
##  1135    190        1    0.610 0.02771         0.558         0.666
##  1145    189        1    0.606 0.02775         0.554         0.663
##  1154    188        1    0.603 0.02779         0.551         0.660
##  1161    187        1    0.600 0.02782         0.548         0.657
##  1178    186        1    0.597 0.02786         0.545         0.654
##  1186    185        1    0.594 0.02790         0.541         0.651
##  1191    184        1    0.590 0.02793         0.538         0.648
##  1207    183        1    0.587 0.02796         0.535         0.645
##  1215    182        1    0.584 0.02800         0.531         0.641
##  1219    181        1    0.581 0.02803         0.528         0.638
##  1252    180        1    0.577 0.02806         0.525         0.635
##  1262    179        1    0.574 0.02808         0.522         0.632
##  1295    178        1    0.571 0.02811         0.518         0.629
##  1325    177        1    0.568 0.02814         0.515         0.626
##  1399    176        1    0.565 0.02816         0.512         0.622
##  1405    175        1    0.561 0.02818         0.509         0.619
##  1434    174        1    0.558 0.02821         0.505         0.616
##  1509    173        1    0.555 0.02823         0.502         0.613
##  1540    171        1    0.552 0.02825         0.499         0.610
##  1548    170        1    0.548 0.02827         0.496         0.607
##  1568    169        1    0.545 0.02829         0.492         0.603
##  1652    168        1    0.542 0.02830         0.489         0.600
##  1709    167        1    0.539 0.02832         0.486         0.597
##  1768    166        1    0.535 0.02833         0.483         0.594
##  1829    163        1    0.532 0.02835         0.479         0.591
##  1839    161        1    0.529 0.02837         0.476         0.587
##  1850    160        1    0.525 0.02838         0.473         0.584
##  1851    159        1    0.522 0.02839         0.469         0.581
##  1879    157        1    0.519 0.02841         0.466         0.578
##  1885    155        1    0.515 0.02842         0.463         0.574
```

```
## 1932    152       1    0.512 0.02843        0.459        0.571
## 2023    144       1    0.509 0.02846        0.456        0.568
## 2079    138       1    0.505 0.02849        0.452        0.564
## 2128    131       1    0.501 0.02853        0.448        0.560
## 2152    122       1    0.497 0.02859        0.444        0.556
## 2171    118       1    0.493 0.02866        0.440        0.552
## 2458     65       1    0.485 0.02920        0.431        0.546
## 2593     42       1    0.474 0.03071        0.417        0.538
## 2683     33       1    0.459 0.03296        0.399        0.529
## 2718     26       1    0.442 0.03611        0.376        0.518
## 2910      9       1    0.392 0.05630        0.296        0.520
##
##                 rx=Lev+5FU
## time n.risk n.event survival std.err lower 95% CI upper 95% CI
##   23    304       1    0.997 0.00328        0.990        1.000
##   34    303       1    0.993 0.00464        0.984        1.000
##   45    302       1    0.990 0.00567        0.979        1.000
##   52    301       1    0.987 0.00654        0.974        1.000
##   79    300       1    0.984 0.00729        0.969        0.998
##  127    299       1    0.980 0.00798        0.965        0.996
##  138    298       1    0.977 0.00860        0.960        0.994
##  141    297       1    0.974 0.00918        0.956        0.992
##  144    296       1    0.970 0.00972        0.952        0.990
##  186    295       1    0.967 0.01023        0.947        0.987
##  251    294       1    0.964 0.01071        0.943        0.985
##  269    293       1    0.961 0.01117        0.939        0.983
##  271    292       1    0.957 0.01160        0.935        0.980
##  274    291       1    0.954 0.01202        0.931        0.978
##  276    290       1    0.951 0.01242        0.927        0.975
##  279    289       1    0.947 0.01281        0.923        0.973
##  283    288       1    0.944 0.01318        0.919        0.970
##  293    287       1    0.941 0.01354        0.915        0.968
##  302    286       1    0.937 0.01388        0.911        0.965
##  304    285       1    0.934 0.01422        0.907        0.962
##  324    284       1    0.931 0.01454        0.903        0.960
##  326    283       1    0.928 0.01486        0.899        0.957
##  340    282       1    0.924 0.01517        0.895        0.955
##  355    281       1    0.921 0.01547        0.891        0.952
##  363    280       1    0.918 0.01576        0.887        0.949
##  389    279       1    0.914 0.01604        0.884        0.946
##  400    278       1    0.911 0.01632        0.880        0.944
##  428    277       1    0.908 0.01659        0.876        0.941
##  430    276       1    0.905 0.01685        0.872        0.938
##  441    275       1    0.901 0.01711        0.868        0.935
##  448    274       1    0.898 0.01736        0.865        0.933
##  454    273       1    0.895 0.01760        0.861        0.930
##  460    272       1    0.891 0.01784        0.857        0.927
##  484    271       1    0.888 0.01808        0.853        0.924
##  490    270       1    0.885 0.01831        0.850        0.921
##  498    269       1    0.882 0.01853        0.846        0.919
##  499    268       1    0.878 0.01875        0.842        0.916
##  503    267       1    0.875 0.01897        0.839        0.913
##  529    266       1    0.872 0.01918        0.835        0.910
##  550    265       1    0.868 0.01939        0.831        0.907
```

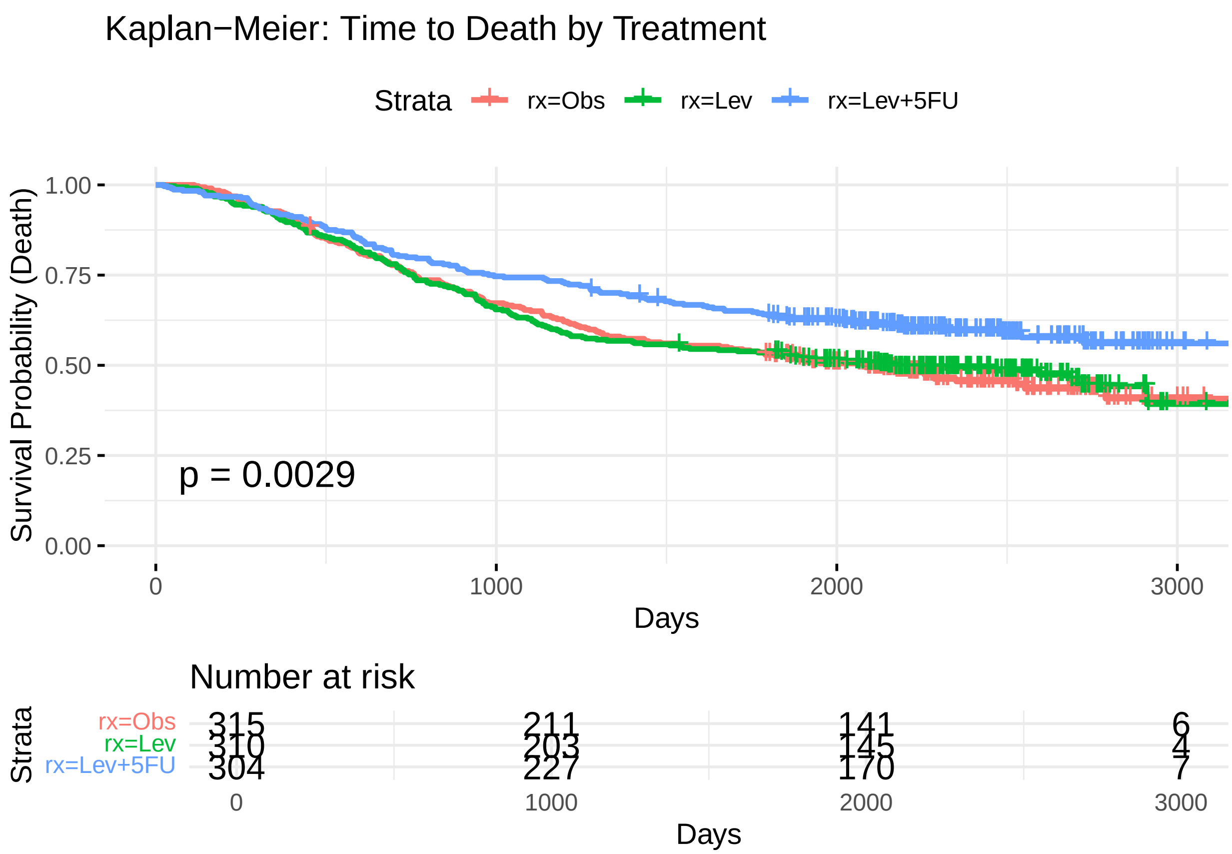
```
##   576    264        1    0.865 0.01959         0.828         0.904
##   578    263        1    0.862 0.01979         0.824         0.902
##   580    262        1    0.859 0.01999         0.820         0.899
##   583    261        1    0.855 0.02018         0.817         0.896
##   592    260        1    0.852 0.02037         0.813         0.893
##   601    259        1    0.849 0.02055         0.809         0.890
##   603    258        1    0.845 0.02074         0.806         0.887
##   609    257        1    0.842 0.02091         0.802         0.884
##   614    256        1    0.839 0.02109         0.798         0.881
##   616    255        1    0.836 0.02126         0.795         0.878
##   641    254        1    0.832 0.02143         0.791         0.875
##   642    253        1    0.829 0.02160         0.788         0.872
##   643    252        1    0.826 0.02176         0.784         0.869
##   666    251        1    0.822 0.02192         0.781         0.866
##   674    250        1    0.819 0.02208         0.777         0.864
##   692    249        2    0.813 0.02239         0.770         0.858
##   693    247        1    0.809 0.02254         0.766         0.855
##   696    246        1    0.806 0.02268         0.763         0.852
##   712    245        1    0.803 0.02283         0.759         0.849
##   736    244        1    0.799 0.02297         0.756         0.846
##   765    243        1    0.796 0.02311         0.752         0.843
##   802    242        2    0.789 0.02338         0.745         0.837
##   806    240        1    0.786 0.02352         0.741         0.834
##   811    239        1    0.783 0.02365         0.738         0.831
##   844    238        1    0.780 0.02377         0.734         0.828
##   862    237        1    0.776 0.02390         0.731         0.825
##   884    236        1    0.773 0.02402         0.727         0.822
##   887    235        2    0.766 0.02427         0.720         0.816
##   905    233        1    0.763 0.02438         0.717         0.812
##   911    232        1    0.760 0.02450         0.713         0.809
##   916    231        1    0.757 0.02461         0.710         0.806
##   961    230        1    0.753 0.02473         0.706         0.803
##   977    229        1    0.750 0.02483         0.703         0.800
##   993    228        1    0.747 0.02494         0.699         0.797
##  1022    227        1    0.743 0.02505         0.696         0.794
##  1138    226        1    0.740 0.02515         0.692         0.791
##  1145    225        1    0.737 0.02526         0.689         0.788
##  1151    224        1    0.734 0.02536         0.686         0.785
##  1193    223        1    0.730 0.02545         0.682         0.782
##  1201    222        1    0.727 0.02555         0.679         0.779
##  1212    221        1    0.724 0.02565         0.675         0.776
##  1246    220        1    0.720 0.02574         0.672         0.773
##  1273    219        1    0.717 0.02583         0.668         0.770
##  1276    218        2    0.711 0.02601         0.661         0.763
##  1279    216        1    0.707 0.02610         0.658         0.760
##  1302    214        1    0.704 0.02618         0.654         0.757
##  1306    213        1    0.701 0.02627         0.651         0.754
##  1365    212        1    0.697 0.02635         0.648         0.751
##  1387    211        1    0.694 0.02643         0.644         0.748
##  1388    210        1    0.691 0.02651         0.641         0.745
##  1424    208        1    0.687 0.02659         0.637         0.742
##  1439    207        1    0.684 0.02667         0.634         0.738
##  1446    206        1    0.681 0.02675         0.630         0.735
##  1495    204        1    0.677 0.02682         0.627         0.732
```

```
##   1511     203        1     0.674 0.02690        0.623         0.729
##   1521     202        1     0.671 0.02697        0.620         0.726
##   1550     201        1     0.667 0.02704        0.616         0.723
##   1607     200        1     0.664 0.02711        0.613         0.719
##   1620     199        1     0.661 0.02718        0.610         0.716
##   1637     198        1     0.657 0.02725        0.606         0.713
##   1668     197        1     0.654 0.02731        0.603         0.710
##   1671     196        1     0.651 0.02738        0.599         0.707
##   1752     195        1     0.647 0.02744        0.596         0.703
##   1767     194        1     0.644 0.02750        0.592         0.700
##   1783     193        1     0.641 0.02756        0.589         0.697
##   1798     192        1     0.637 0.02762        0.585         0.694
##   1812     190        1     0.634 0.02767        0.582         0.691
##   1831     185        1     0.631 0.02774        0.579         0.687
##   1856     183        1     0.627 0.02780        0.575         0.684
##   1995     172        1     0.623 0.02787        0.571         0.681
##   2021     167        1     0.620 0.02796        0.567         0.677
##   2052     161        1     0.616 0.02805        0.563         0.673
##   2127     146        1     0.612 0.02817        0.559         0.669
##   2174     136        1     0.607 0.02832        0.554         0.665
##   2197     127        1     0.602 0.02850        0.549         0.661
##   2318     102        1     0.597 0.02882        0.543         0.656
##   2482      72        1     0.588 0.02959        0.533         0.649
##   2542      53        1     0.577 0.03104        0.519         0.641
##   2725      35        1     0.561 0.03426        0.497         0.632
```

```r
plot(fit_km_death,
     xlab = "Days",
     ylab = "Survival Probability (Time to Death)",
     col = c("blue", "red", "green"),
     lty = 1:3,
     main = "Kaplan-Meier Curves: Time to Death by Treatment")

legend("bottomleft",
       legend = levels(colon_death$rx),
       col = c("blue", "red", "green"),
       lty = 1:3)
```




```r
ggsurvplot(
  fit = fit_km_death,
  data = colon_death,
  risk.table = TRUE,
  pval = TRUE,
  xlab = "Days",
  ylab = "Survival Probability (Death)",
  title = "Kaplan-Meier: Time to Death by Treatment",
  ggtheme = theme_minimal()
)
```




```r
logrank_death <- survdiff(surv_death ~ rx, data = colon_death)
logrank_death
```

```
## Call:
## survdiff(formula = surv_death ~ rx, data = colon_death)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## rx=Obs       315      168      148      2.58      3.85
## rx=Lev       310      161      146      1.52      2.25
## rx=Lev+5FU 304      123      157      7.55     11.62
##
##  Chisq= 11.7  on 2 degrees of freedom, p= 0.003
```

```r
survdiff(Surv(time, status) ~ rx, data = colon_death,
         subset = (rx %in% c("Obs", "Lev")))
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ rx, data = colon_death,
##     subset = (rx %in% c("Obs", "Lev")))
##
##            N Observed Expected (O-E)^2/E (O-E)^2/V
## rx=Obs 315      168      166    0.0282     0.057
## rx=Lev 310      161      163    0.0287     0.057
##
##  Chisq= 0.1  on 1 degrees of freedom, p= 0.8
```

```r
survdiff(Surv(time, status) ~ rx, data = colon_death,
         subset = (rx %in% c("Obs", "Lev+5FU")))
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ rx, data = colon_death,
##     subset = (rx %in% c("Obs", "Lev+5FU")))
##
##               N Observed Expected (O-E)^2/E (O-E)^2/V
## rx=Obs      315      168      141      5.12      9.97
## rx=Lev+5FU 304      123      150      4.82      9.97
##
##  Chisq= 10  on 1 degrees of freedom, p= 0.002
```

```r
survdiff(Surv(time, status) ~ rx, data = colon_death,
         subset = (rx %in% c("Lev", "Lev+5FU")))
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ rx, data = colon_death,
##     subset = (rx %in% c("Lev", "Lev+5FU")))
##
##               N Observed Expected (O-E)^2/E (O-E)^2/V
## rx=Lev      310      161      137      4.24      8.21
## rx=Lev+5FU 304      123      147      3.95      8.21
##
##  Chisq= 8.2  on 1 degrees of freedom, p= 0.004
```

Kaplan-Meier Curves (Time to Death)

1. Visual Ranking of Survival Curves

- **Lev+5FU (green/dotted)**: Remains the highest curve (best survival), meaning fewer patients die (or they die later).
- **Obs (blue/solid)** and **Lev (red/dashed)**: Are closer together. Sometimes one is slightly above the other, but they appear more similar than in the any-event analysis.

Survival Probabilities Over Time

- **Early on (first 500–1000 days)**: Survival probabilities are quite high for all three groups.
- **By ~3000 days**: The curves have separated more clearly, with Lev+5FU maintaining the highest survival and Obs vs. Lev looking fairly close.

2. Log-Rank Test for Time to Death

- **Chi-square = 11.7, p = 0.003**: Indicates a statistically significant difference in time to death among the three groups.

Interpretation of Observed vs. Expected:

- **Obs**: 168 deaths observed vs. 148 expected, indicating more deaths than expected under a null (no-difference) scenario.
- **Lev**: Also has more observed than expected.
- **Lev+5FU**: Presumably has fewer observed than expected, aligning with better survival.

3. Comparison with "Any Event" Results

- **Any Event Analysis**: Had an extremely low p-value (~10) because Lev+5FU was drastically better for recurrence and death combined.
- **Time to Death**: Yields p = 0.003—still significant but less extreme.
- **Implication**: The difference among treatments is still real for death alone, but the magnitude of separation is somewhat smaller than in the any-event scenario. This suggests Lev+5FU provides a strong advantage particularly for recurrence prevention, though it also improves overall survival compared to the others.

4. Do Obs and Lev Differ?

- The log-rank test is global, so we only know that at least one pair differs.
- **Visually**: Obs and Lev appear close. You can do pairwise tests (subsetting to (Obs, Lev) only) to see if that difference is statistically significant.
- **If the p-value is large**: It means Obs vs. Lev are not significantly different in terms of time to death.

Kaplan-Meier Curves (Time to Recurrence)

```r
# Step 1: Summarize time/status for recurrence
colon_recur <- colon_data %>%
  group_by(id) %>%
  summarize(
    # Earliest time if (status=1 & etype=1), otherwise max time
    time = if (any(status == 1 & etype == 1)) {
      min(time[status == 1 & etype == 1])
    } else {
      max(time)
    },
    # Recurrence indicator: 1 if event occurred, else 0
    status = if (any(status == 1 & etype == 1)) 1 else 0
  ) %>%
  ungroup()

# Step 2: Merge `rx` from the original data
colon_recur <- colon_recur %>%
  left_join(
    select(colon_data, id, rx),
    by = "id"
  ) %>%
  distinct(id, .keep_all = TRUE)

# Check structure
str(colon_recur)
```

```
## tibble [929 x 4] (S3: tbl_df/tbl/data.frame)
##  $ id    : num [1:929] 1 2 3 4 5 6 7 8 9 10 ...
##  $ time  : num [1:929] 968 3087 542 245 523 ...
##  $ status: num [1:929] 1 0 1 1 1 1 0 0 0 ...
##  $ rx    : Factor w/ 3 levels "Obs","Lev","Lev+5FU": 3 3 1 3 1 3 2 1 2 3 ...
```

```r
head(colon_recur)
```

```
## # A tibble: 6 x 4
##      id  time status rx
##   <dbl> <dbl>  <dbl> <fct>
## 1     1   968      1 Lev+5FU
## 2     2  3087      0 Lev+5FU
## 3     3   542      1 Obs
## 4     4   245      1 Lev+5FU
## 5     5   523      1 Obs
## 6     6   904      1 Lev+5FU
```

```r
# Create a Surv object for time to recurrence
colon_recur$surv_recur <- Surv(time = colon_recur$time,
                               event = colon_recur$status)
```

```r
fit_km_recur <- survfit(surv_recur ~ rx, data = colon_recur)

# View a textual summary of survival estimates
summary(fit_km_recur)
```

```
## Call: survfit(formula = surv_recur ~ rx, data = colon_recur)
##
##                 rx=Obs
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    20    315       1    0.997 0.00317        0.991        1.000
##    36    314       1    0.994 0.00448        0.985        1.000
##    43    313       1    0.990 0.00547        0.980        1.000
##    45    312       1    0.987 0.00631        0.975        1.000
##    59    311       1    0.984 0.00704        0.970        0.998
##    72    310       1    0.981 0.00770        0.966        0.996
##    77    309       1    0.978 0.00831        0.962        0.994
##    79    308       1    0.975 0.00886        0.957        0.992
##    80    307       2    0.968 0.00988        0.949        0.988
##    85    305       1    0.965 0.01034        0.945        0.986
##    86    304       1    0.962 0.01079        0.941        0.983
##    88    303       1    0.959 0.01121        0.937        0.981
##    94    302       1    0.956 0.01161        0.933        0.979
##    98    301       1    0.952 0.01200        0.929        0.976
##    99    300       2    0.946 0.01273        0.921        0.971
##   101    298       1    0.943 0.01308        0.918        0.969
##   102    297       1    0.940 0.01341        0.914        0.966
##   103    296       1    0.937 0.01374        0.910        0.964
##   106    295       1    0.933 0.01405        0.906        0.961
##   108    294       1    0.930 0.01436        0.902        0.959
##   109    293       1    0.927 0.01466        0.899        0.956
##   113    292       1    0.924 0.01495        0.895        0.954
##   118    291       1    0.921 0.01523        0.891        0.951
##   121    290       1    0.917 0.01550        0.888        0.948
##   122    289       1    0.914 0.01577        0.884        0.946
##   127    288       1    0.911 0.01603        0.880        0.943
##   131    287       1    0.908 0.01629        0.877        0.940
```

```
##    139    286       1    0.905 0.01654        0.873        0.938
##    143    285       1    0.902 0.01678        0.869        0.935
##    154    284       1    0.898 0.01702        0.866        0.932
##    157    283       1    0.895 0.01726        0.862        0.930
##    161    282       1    0.892 0.01748        0.858        0.927
##    165    281       2    0.886 0.01793        0.851        0.922
##    166    279       1    0.883 0.01814        0.848        0.919
##    167    278       1    0.879 0.01835        0.844        0.916
##    173    277       3    0.870 0.01896        0.833        0.908
##    174    274       1    0.867 0.01915        0.830        0.905
##    185    273       2    0.860 0.01953        0.823        0.899
##    188    271       1    0.857 0.01972        0.819        0.897
##    189    270       1    0.854 0.01990        0.816        0.894
##    201    269       1    0.851 0.02007        0.812        0.891
##    203    268       1    0.848 0.02025        0.809        0.888
##    208    267       1    0.844 0.02042        0.805        0.885
##    215    266       1    0.841 0.02059        0.802        0.883
##    218    265       1    0.838 0.02075        0.798        0.880
##    221    264       1    0.835 0.02092        0.795        0.877
##    223    263       1    0.832 0.02108        0.791        0.874
##    227    262       1    0.829 0.02123        0.788        0.871
##    228    261       1    0.825 0.02139        0.785        0.868
##    229    260       1    0.822 0.02154        0.781        0.866
##    230    259       3    0.813 0.02198        0.771        0.857
##    237    256       1    0.810 0.02212        0.767        0.854
##    238    255       2    0.803 0.02240        0.760        0.848
##    243    253       1    0.800 0.02254        0.757        0.845
##    245    252       1    0.797 0.02267        0.754        0.843
##    256    251       1    0.794 0.02280        0.750        0.840
##    257    250       1    0.790 0.02293        0.747        0.837
##    263    249       1    0.787 0.02306        0.743        0.834
##    264    248       1    0.784 0.02318        0.740        0.831
##    271    247       1    0.781 0.02330        0.737        0.828
##    273    246       1    0.778 0.02342        0.733        0.825
##    276    245       1    0.775 0.02354        0.730        0.822
##    279    244       1    0.771 0.02366        0.726        0.819
##    280    243       1    0.768 0.02377        0.723        0.816
##    286    242       1    0.765 0.02389        0.720        0.813
##    291    241       1    0.762 0.02400        0.716        0.810
##    294    240       1    0.759 0.02411        0.713        0.807
##    296    239       1    0.756 0.02421        0.710        0.805
##    304    238       1    0.752 0.02432        0.706        0.802
##    308    237       1    0.749 0.02442        0.703        0.799
##    315    236       1    0.746 0.02453        0.699        0.796
##    334    235       1    0.743 0.02463        0.696        0.793
##    337    234       1    0.740 0.02472        0.693        0.790
##    344    233       1    0.737 0.02482        0.689        0.787
##    349    232       1    0.733 0.02492        0.686        0.784
##    352    231       1    0.730 0.02501        0.683        0.781
##    354    230       1    0.727 0.02510        0.679        0.778
##    360    229       1    0.724 0.02519        0.676        0.775
##    362    228       1    0.721 0.02528        0.673        0.772
##    374    227       1    0.717 0.02537        0.669        0.769
##    378    226       1    0.714 0.02545        0.666        0.766
```

```
##   379    225        1     0.711 0.02554        0.663         0.763
##   382    224        1     0.708 0.02562        0.659         0.760
##   384    223        1     0.705 0.02570        0.656         0.757
##   398    222        1     0.702 0.02578        0.653         0.754
##   401    221        1     0.698 0.02586        0.650         0.751
##   402    220        1     0.695 0.02594        0.646         0.748
##   406    219        1     0.692 0.02601        0.643         0.745
##   411    218        1     0.689 0.02608        0.640         0.742
##   433    216        1     0.686 0.02616        0.636         0.739
##   435    215        1     0.683 0.02623        0.633         0.736
##   437    214        1     0.679 0.02630        0.630         0.733
##   438    213        1     0.676 0.02637        0.626         0.730
##   461    211        1     0.673 0.02644        0.623         0.727
##   480    210        1     0.670 0.02651        0.620         0.724
##   489    208        1     0.667 0.02657        0.616         0.721
##   493    207        1     0.663 0.02664        0.613         0.718
##   495    206        1     0.660 0.02670        0.610         0.715
##   496    205        1     0.657 0.02677        0.606         0.711
##   499    204        1     0.654 0.02683        0.603         0.708
##   523    203        1     0.650 0.02689        0.600         0.705
##   532    202        1     0.647 0.02695        0.596         0.702
##   534    201        1     0.644 0.02701        0.593         0.699
##   540    200        1     0.641 0.02706        0.590         0.696
##   542    199        1     0.638 0.02712        0.587         0.693
##   543    198        1     0.634 0.02717        0.583         0.690
##   547    197        1     0.631 0.02722        0.580         0.687
##   555    196        1     0.628 0.02727        0.577         0.684
##   561    195        1     0.625 0.02732        0.573         0.681
##   577    194        1     0.621 0.02737        0.570         0.677
##   581    193        1     0.618 0.02742        0.567         0.674
##   593    192        1     0.615 0.02746        0.563         0.671
##   608    190        1     0.612 0.02751        0.560         0.668
##   625    189        1     0.609 0.02755        0.557         0.665
##   632    188        1     0.605 0.02760        0.554         0.662
##   659    187        1     0.602 0.02764        0.550         0.659
##   663    186        1     0.599 0.02768        0.547         0.656
##   686    184        1     0.596 0.02772        0.544         0.652
##   700    183        1     0.592 0.02776        0.540         0.649
##   702    182        2     0.586 0.02783        0.534         0.643
##   712    180        1     0.583 0.02787        0.530         0.640
##   726    179        1     0.579 0.02790        0.527         0.637
##   730    178        1     0.576 0.02793        0.524         0.633
##   731    177        1     0.573 0.02796        0.521         0.630
##   735    176        1     0.570 0.02799        0.517         0.627
##   739    175        1     0.566 0.02802        0.514         0.624
##   748    174        1     0.563 0.02805        0.511         0.621
##   752    173        1     0.560 0.02808        0.507         0.618
##   772    172        1     0.556 0.02810        0.504         0.614
##   774    171        1     0.553 0.02812        0.501         0.611
##   803    170        1     0.550 0.02815        0.497         0.608
##   835    169        1     0.547 0.02817        0.494         0.605
##   855    167        1     0.543 0.02819        0.491         0.602
##   871    166        1     0.540 0.02821        0.488         0.598
##   912    164        1     0.537 0.02823        0.484         0.595
```

```
##   930    163       1    0.534 0.02825        0.481        0.592
##   975    162       1    0.530 0.02826        0.478        0.589
##  1020    161       1    0.527 0.02828        0.474        0.585
##  1042    160       1    0.524 0.02829        0.471        0.582
##  1057    159       2    0.517 0.02832        0.465        0.576
##  1081    157       1    0.514 0.02833        0.461        0.572
##  1089    156       1    0.511 0.02834        0.458        0.569
##  1106    155       1    0.507 0.02835        0.455        0.566
##  1130    154       1    0.504 0.02835        0.451        0.563
##  1139    153       1    0.501 0.02836        0.448        0.559
##  1236    151       1    0.497 0.02836        0.445        0.556
##  1274    150       1    0.494 0.02837        0.441        0.553
##  1323    148       1    0.491 0.02837        0.438        0.550
##  1353    147       1    0.487 0.02837        0.435        0.546
##  1432    145       1    0.484 0.02838        0.431        0.543
##  1436    144       1    0.481 0.02838        0.428        0.540
##  1446    143       1    0.477 0.02838        0.425        0.536
##  1455    142       1    0.474 0.02838        0.421        0.533
##  1466    141       1    0.471 0.02837        0.418        0.530
##  1475    140       1    0.467 0.02837        0.415        0.526
##  1535    139       1    0.464 0.02836        0.411        0.523
##  1539    138       1    0.460 0.02836        0.408        0.520
##  1606    137       1    0.457 0.02835        0.405        0.516
##  1749    136       1    0.454 0.02834        0.401        0.513
##  1759    135       1    0.450 0.02833        0.398        0.509
##  1981    118       1    0.447 0.02834        0.394        0.506
##  2035    111       1    0.443 0.02837        0.390        0.502
##  2036    110       1    0.439 0.02840        0.386        0.498
##  2148     96       1    0.434 0.02847        0.382        0.493
##  2288     61       1    0.427 0.02887        0.374        0.487
##  2695     22       1    0.407 0.03345        0.347        0.479
##
##                 rx=Lev
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##    19    310       1    0.997 0.00322        0.990        1.000
##    28    308       1    0.994 0.00455        0.985        1.000
##    35    307       1    0.990 0.00557        0.979        1.000
##    38    306       1    0.987 0.00643        0.975        1.000
##    62    304       2    0.981 0.00785        0.965        0.996
##    72    302       1    0.977 0.00847        0.961        0.994
##    77    301       1    0.974 0.00905        0.957        0.992
##    78    300       1    0.971 0.00958        0.952        0.990
##    80    299       1    0.968 0.01008        0.948        0.988
##    85    298       1    0.964 0.01056        0.944        0.985
##    91    297       1    0.961 0.01101        0.940        0.983
##    98    296       2    0.955 0.01186        0.932        0.978
##   100    294       1    0.951 0.01225        0.928        0.976
##   105    293       1    0.948 0.01263        0.924        0.973
##   111    292       1    0.945 0.01300        0.920        0.971
##   113    291       2    0.938 0.01370        0.912        0.966
##   116    289       2    0.932 0.01435        0.904        0.960
##   119    287       1    0.929 0.01467        0.900        0.958
##   121    286       1    0.925 0.01497        0.896        0.955
##   136    285       1    0.922 0.01526        0.893        0.953
```

```
##   141    284       1    0.919 0.01555        0.889         0.950
##   145    283       1    0.916 0.01583        0.885         0.947
##   146    282       1    0.912 0.01611        0.881         0.945
##   147    281       1    0.909 0.01637        0.878         0.942
##   157    280       1    0.906 0.01663        0.874         0.939
##   169    279       1    0.903 0.01689        0.870         0.936
##   174    278       2    0.896 0.01738        0.863         0.931
##   175    276       1    0.893 0.01762        0.859         0.928
##   176    275       2    0.886 0.01808        0.852         0.923
##   179    273       1    0.883 0.01830        0.848         0.920
##   181    272       1    0.880 0.01852        0.844         0.917
##   183    271       1    0.877 0.01873        0.841         0.914
##   185    270       2    0.870 0.01915        0.833         0.909
##   189    268       1    0.867 0.01935        0.830         0.906
##   191    267       2    0.860 0.01974        0.823         0.900
##   196    265       1    0.857 0.01993        0.819         0.897
##   204    264       1    0.854 0.02012        0.815         0.894
##   216    263       1    0.851 0.02030        0.812         0.891
##   218    262       1    0.847 0.02049        0.808         0.889
##   219    261       1    0.844 0.02066        0.805         0.886
##   224    259       1    0.841 0.02084        0.801         0.883
##   229    258       1    0.838 0.02101        0.797         0.880
##   230    257       2    0.831 0.02135        0.790         0.874
##   235    255       1    0.828 0.02151        0.787         0.871
##   246    254       1    0.825 0.02167        0.783         0.868
##   250    253       1    0.821 0.02183        0.780         0.865
##   253    252       1    0.818 0.02198        0.776         0.862
##   258    251       1    0.815 0.02214        0.773         0.859
##   260    250       1    0.812 0.02229        0.769         0.856
##   262    249       1    0.808 0.02244        0.766         0.854
##   263    248       1    0.805 0.02258        0.762         0.851
##   271    247       1    0.802 0.02272        0.759         0.848
##   274    246       1    0.799 0.02286        0.755         0.845
##   276    245       1    0.795 0.02300        0.751         0.842
##   279    244       2    0.789 0.02327        0.744         0.836
##   286    242       2    0.782 0.02353        0.737         0.830
##   290    240       2    0.776 0.02378        0.731         0.824
##   294    238       1    0.772 0.02390        0.727         0.821
##   300    237       1    0.769 0.02402        0.724         0.818
##   313    236       1    0.766 0.02414        0.720         0.815
##   325    235       1    0.763 0.02426        0.717         0.812
##   330    234       2    0.756 0.02448        0.710         0.806
##   333    232       1    0.753 0.02460        0.706         0.803
##   335    231       1    0.750 0.02470        0.703         0.800
##   336    230       1    0.746 0.02481        0.699         0.797
##   337    229       2    0.740 0.02502        0.692         0.791
##   341    227       1    0.737 0.02512        0.689         0.788
##   348    226       2    0.730 0.02532        0.682         0.781
##   352    224       1    0.727 0.02541        0.679         0.778
##   356    223       2    0.720 0.02560        0.672         0.772
##   366    221       1    0.717 0.02569        0.668         0.769
##   369    220       1    0.714 0.02578        0.665         0.766
##   370    219       1    0.711 0.02587        0.662         0.763
##   372    218       1    0.707 0.02595        0.658         0.760
```

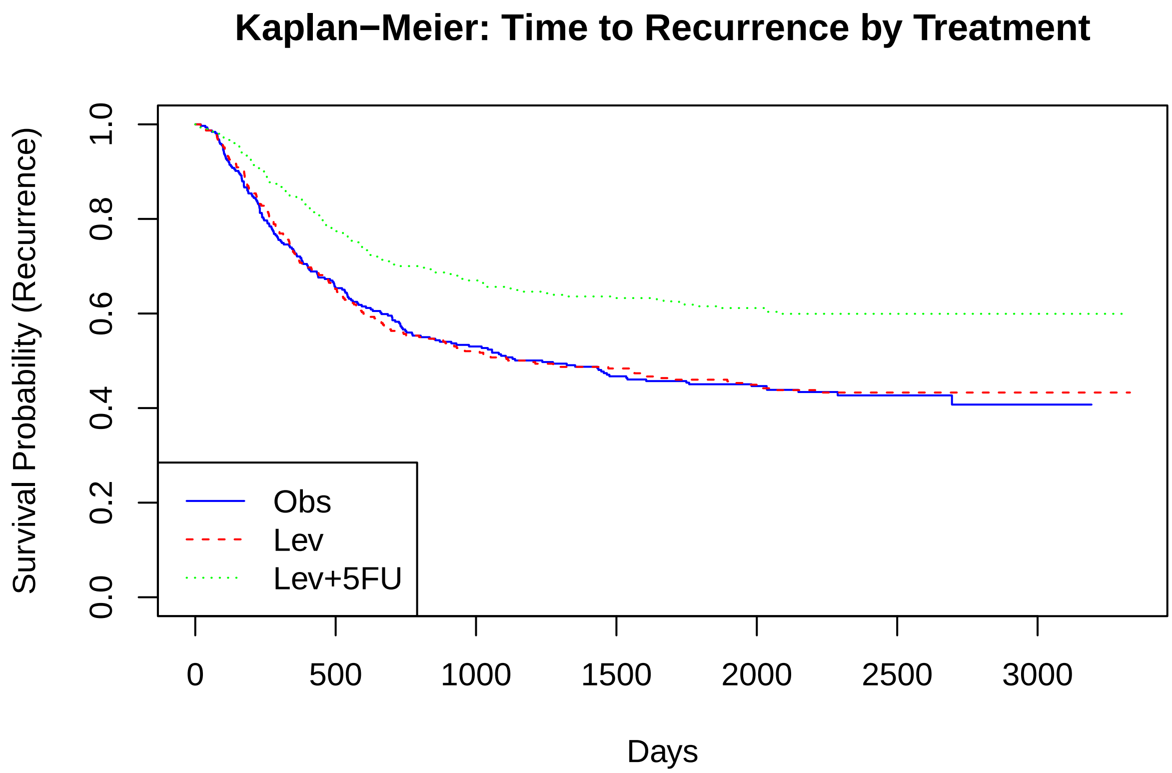
```
##   380    217         1    0.704 0.02604         0.655         0.757
##   386    216         1    0.701 0.02612         0.651         0.754
##   389    215         1    0.698 0.02620         0.648         0.751
##   413    214         1    0.694 0.02628         0.645         0.748
##   415    213         1    0.691 0.02636         0.641         0.745
##   429    212         1    0.688 0.02643         0.638         0.742
##   439    211         1    0.684 0.02651         0.634         0.738
##   440    210         1    0.681 0.02658         0.631         0.735
##   454    209         1    0.678 0.02665         0.628         0.732
##   458    208         1    0.675 0.02673         0.624         0.729
##   465    207         1    0.671 0.02679         0.621         0.726
##   474    206         1    0.668 0.02686         0.618         0.723
##   476    205         1    0.665 0.02693         0.614         0.720
##   482    204         1    0.662 0.02699         0.611         0.717
##   490    202         1    0.658 0.02706         0.607         0.714
##   491    201         1    0.655 0.02712         0.604         0.710
##   498    200         1    0.652 0.02718         0.601         0.707
##   504    199         1    0.649 0.02724         0.597         0.704
##   505    198         1    0.645 0.02730         0.594         0.701
##   511    197         1    0.642 0.02736         0.591         0.698
##   513    196         1    0.639 0.02741         0.587         0.695
##   525    195         1    0.635 0.02747         0.584         0.692
##   527    194         1    0.632 0.02752         0.580         0.688
##   532    193         1    0.629 0.02757         0.577         0.685
##   548    192         1    0.626 0.02762         0.574         0.682
##   560    191         1    0.622 0.02767         0.570         0.679
##   565    190         1    0.619 0.02772         0.567         0.676
##   573    189         2    0.613 0.02781         0.560         0.670
##   578    187         1    0.609 0.02785         0.557         0.666
##   583    186         1    0.606 0.02790         0.554         0.663
##   593    185         1    0.603 0.02794         0.550         0.660
##   599    184         1    0.599 0.02798         0.547         0.657
##   613    183         1    0.596 0.02801         0.544         0.654
##   615    182         1    0.593 0.02805         0.540         0.650
##   638    180         1    0.590 0.02809         0.537         0.647
##   653    179         1    0.586 0.02812         0.534         0.644
##   654    178         1    0.583 0.02816         0.530         0.641
##   663    177         1    0.580 0.02819         0.527         0.638
##   668    176         1    0.576 0.02822         0.524         0.634
##   672    175         1    0.573 0.02825         0.520         0.631
##   675    174         1    0.570 0.02828         0.517         0.628
##   680    173         1    0.567 0.02831         0.514         0.625
##   697    172         1    0.563 0.02834         0.510         0.622
##   717    171         1    0.560 0.02836         0.507         0.618
##   742    170         1    0.557 0.02838         0.504         0.615
##   751    169         1    0.553 0.02841         0.500         0.612
##   797    168         1    0.550 0.02843         0.497         0.609
##   828    167         1    0.547 0.02845         0.494         0.605
##   851    166         1    0.543 0.02847         0.490         0.602
##   883    165         1    0.540 0.02848         0.487         0.599
##   891    164         1    0.537 0.02850         0.484         0.596
##   900    163         1    0.534 0.02852         0.481         0.593
##   922    162         1    0.530 0.02853         0.477         0.589
##   931    161         1    0.527 0.02854         0.474         0.586
```

```
##   959    159       1    0.524 0.02855        0.471         0.583
##   960    158       1    0.520 0.02856        0.467         0.579
##  1013    157       1    0.517 0.02857        0.464         0.576
##  1026    156       1    0.514 0.02858        0.461         0.573
##  1029    155       1    0.510 0.02859        0.457         0.570
##  1052    154       1    0.507 0.02860        0.454         0.566
##  1108    152       1    0.504 0.02860        0.451         0.563
##  1114    151       1    0.500 0.02861        0.447         0.560
##  1183    150       1    0.497 0.02861        0.444         0.556
##  1211    149       1    0.494 0.02861        0.441         0.553
##  1275    148       1    0.490 0.02861        0.437         0.550
##  1298    147       1    0.487 0.02861        0.434         0.547
##  1471    146       1    0.484 0.02861        0.431         0.543
##  1551    143       1    0.480 0.02861        0.427         0.540
##  1561    142       1    0.477 0.02861        0.424         0.536
##  1564    141       1    0.474 0.02860        0.421         0.533
##  1589    140       1    0.470 0.02860        0.417         0.530
##  1606    139       1    0.467 0.02859        0.414         0.526
##  1647    138       1    0.463 0.02858        0.411         0.523
##  1687    137       1    0.460 0.02857        0.407         0.520
##  1895    131       1    0.457 0.02857        0.404         0.516
##  1918    129       1    0.453 0.02857        0.400         0.513
##  1976    125       1    0.449 0.02857        0.397         0.509
##  2012    120       1    0.446 0.02857        0.393         0.505
##  2018    119       1    0.442 0.02858        0.389         0.502
##  2067    114       1    0.438 0.02859        0.385         0.498
##  2231     85       1    0.433 0.02871        0.380         0.493
##
##                 rx=Lev+5FU
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     8    304       1    0.997 0.00328        0.990         1.000
##     9    303       1    0.993 0.00464        0.984         1.000
##    40    301       1    0.990 0.00568        0.979         1.000
##    49    299       1    0.987 0.00655        0.974         1.000
##    63    298       1    0.983 0.00732        0.969         0.998
##    68    297       1    0.980 0.00801        0.965         0.996
##    86    296       1    0.977 0.00864        0.960         0.994
##    91    295       1    0.974 0.00922        0.956         0.992
##   101    294       1    0.970 0.00977        0.951         0.990
##   116    293       1    0.967 0.01028        0.947         0.987
##   132    292       1    0.964 0.01077        0.943         0.985
##   134    291       1    0.960 0.01123        0.939         0.983
##   146    290       1    0.957 0.01167        0.934         0.980
##   154    289       1    0.954 0.01209        0.930         0.978
##   157    288       1    0.950 0.01249        0.926         0.975
##   160    287       1    0.947 0.01288        0.922         0.973
##   161    286       1    0.944 0.01325        0.918         0.970
##   165    285       1    0.940 0.01361        0.914         0.968
##   168    284       1    0.937 0.01396        0.910         0.965
##   183    283       1    0.934 0.01430        0.906         0.962
##   185    282       1    0.931 0.01463        0.902         0.960
##   186    281       1    0.927 0.01495        0.898         0.957
##   198    280       1    0.924 0.01525        0.894         0.954
##   199    279       1    0.921 0.01555        0.891         0.952
```

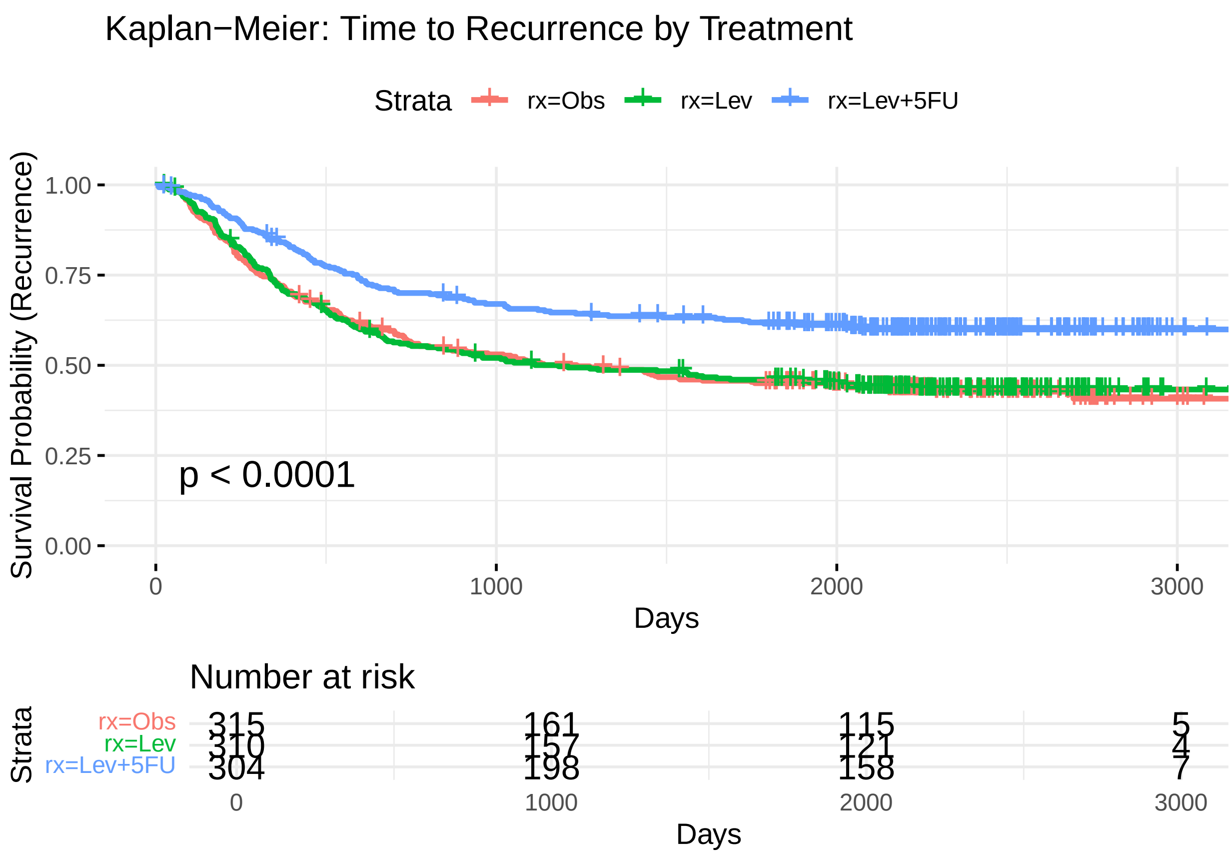
```
##   205    278       1    0.917 0.01585         0.887         0.949
##   208    277       1    0.914 0.01613         0.883         0.946
##   215    276       1    0.911 0.01641         0.879         0.943
##   218    275       1    0.907 0.01668         0.875         0.941
##   237    274       1    0.904 0.01695         0.871         0.938
##   242    273       1    0.901 0.01720         0.868         0.935
##   245    272       1    0.897 0.01746         0.864         0.932
##   248    271       1    0.894 0.01770         0.860         0.929
##   252    270       1    0.891 0.01795         0.856         0.927
##   255    269       1    0.887 0.01818         0.853         0.924
##   256    268       1    0.884 0.01841         0.849         0.921
##   260    267       1    0.881 0.01864         0.845         0.918
##   261    266       1    0.878 0.01886         0.841         0.915
##   285    265       1    0.874 0.01908         0.838         0.912
##   296    264       1    0.871 0.01929         0.834         0.910
##   303    263       1    0.868 0.01950         0.830         0.907
##   315    262       1    0.864 0.01970         0.827         0.904
##   322    261       2    0.858 0.02010         0.819         0.898
##   328    258       1    0.854 0.02030         0.815         0.895
##   329    257       1    0.851 0.02049         0.812         0.892
##   336    256       1    0.848 0.02068         0.808         0.889
##   360    253       1    0.844 0.02087         0.804         0.886
##   365    252       1    0.841 0.02105         0.801         0.883
##   380    251       1    0.838 0.02123         0.797         0.880
##   386    250       1    0.834 0.02141         0.793         0.877
##   392    249       1    0.831 0.02158         0.790         0.874
##   393    248       1    0.828 0.02175         0.786         0.871
##   405    247       1    0.824 0.02192         0.782         0.868
##   408    246       1    0.821 0.02209         0.779         0.865
##   415    245       1    0.818 0.02225         0.775         0.862
##   422    244       1    0.814 0.02241         0.771         0.859
##   431    243       1    0.811 0.02257         0.768         0.856
##   434    242       1    0.807 0.02272         0.764         0.853
##   443    241       1    0.804 0.02287         0.761         0.850
##   448    240       1    0.801 0.02302         0.757         0.847
##   449    239       1    0.797 0.02317         0.753         0.844
##   454    238       1    0.794 0.02331         0.750         0.841
##   458    237       1    0.791 0.02345         0.746         0.838
##   466    236       2    0.784 0.02373         0.739         0.832
##   485    234       1    0.781 0.02386         0.735         0.829
##   491    233       1    0.777 0.02399         0.732         0.826
##   497    232       1    0.774 0.02412         0.728         0.823
##   510    231       1    0.771 0.02425         0.725         0.820
##   526    230       1    0.767 0.02438         0.721         0.817
##   536    229       1    0.764 0.02450         0.717         0.813
##   543    228       1    0.761 0.02462         0.714         0.810
##   554    227       2    0.754 0.02485         0.707         0.804
##   578    225       1    0.751 0.02497         0.703         0.801
##   591    224       1    0.747 0.02508         0.700         0.798
##   593    223       1    0.744 0.02519         0.696         0.795
##   594    222       1    0.740 0.02530         0.693         0.792
##   602    221       1    0.737 0.02541         0.689         0.789
##   604    220       1    0.734 0.02551         0.685         0.786
##   616    219       1    0.730 0.02561         0.682         0.782
```

```
##    617    218       1    0.727 0.02571          0.678         0.779
##    622    217       1    0.724 0.02581          0.675         0.776
##    636    216       1    0.720 0.02591          0.671         0.773
##    649    215       1    0.717 0.02600          0.668         0.770
##    657    214       1    0.714 0.02610          0.664         0.767
##    683    213       1    0.710 0.02619          0.661         0.764
##    700    212       1    0.707 0.02628          0.657         0.760
##    701    211       1    0.704 0.02637          0.654         0.757
##    711    210       1    0.700 0.02645          0.650         0.754
##    805    209       1    0.697 0.02654          0.647         0.751
##    827    208       1    0.694 0.02662          0.643         0.748
##    849    206       1    0.690 0.02670          0.640         0.745
##    853    205       1    0.687 0.02679          0.636         0.741
##    904    203       1    0.683 0.02687          0.633         0.738
##    918    202       1    0.680 0.02695          0.629         0.735
##    934    201       1    0.677 0.02702          0.626         0.732
##    936    200       1    0.673 0.02710          0.622         0.729
##    968    199       1    0.670 0.02717          0.619         0.725
##   1024    198       1    0.667 0.02725          0.615         0.722
##   1025    197       1    0.663 0.02732          0.612         0.719
##   1032    196       1    0.660 0.02739          0.608         0.716
##   1037    195       1    0.656 0.02745          0.605         0.712
##   1122    194       1    0.653 0.02752          0.601         0.709
##   1142    193       1    0.650 0.02758          0.598         0.706
##   1159    192       1    0.646 0.02765          0.594         0.703
##   1233    191       1    0.643 0.02771          0.591         0.700
##   1277    190       1    0.639 0.02777          0.587         0.696
##   1329    188       1    0.636 0.02783          0.584         0.693
##   1488    185       1    0.633 0.02789          0.580         0.690
##   1644    182       1    0.629 0.02795          0.577         0.686
##   1668    181       1    0.626 0.02801          0.573         0.683
##   1723    180       1    0.622 0.02807          0.570         0.680
##   1743    179       1    0.619 0.02813          0.566         0.676
##   1786    178       1    0.615 0.02819          0.562         0.673
##   1876    167       1    0.612 0.02826          0.559         0.670
##   2028    154       1    0.608 0.02835          0.554         0.666
##   2031    152       1    0.604 0.02845          0.550         0.662
##   2074    143       1    0.599 0.02856          0.546         0.658
```

```r
plot(fit_km_recur,
     xlab = "Days",
     ylab = "Survival Probability (Recurrence)",
     col = c("blue", "red", "green"),
     lty = 1:3,
     main = "Kaplan-Meier: Time to Recurrence by Treatment")

legend("bottomleft",
       legend = levels(colon_recur$rx),
       col = c("blue", "red", "green"),
       lty = 1:3)
```




```r
ggsurvplot(
  fit = fit_km_recur,
  data = colon_recur,
  risk.table = TRUE,
  pval = TRUE,  # displays log-rank p-value
  xlab = "Days",
  ylab = "Survival Probability (Recurrence)",
  title = "Kaplan-Meier: Time to Recurrence by Treatment",
  ggtheme = theme_minimal()
)
```




```r
logrank_recur <- survdiff(surv_recur ~ rx, data = colon_recur)
logrank_recur
```

```
## Call:
## survdiff(formula = surv_recur ~ rx, data = colon_recur)
##
##               N Observed Expected (O-E)^2/E (O-E)^2/V
## rx=Obs      315      177      151      4.62      6.82
## rx=Lev      310      172      149      3.69      5.41
## rx=Lev+5FU 304      119      169     14.69     23.04
##
##  Chisq= 23.1  on 2 degrees of freedom, p= 1e-05
```

```r
# Obs vs Lev
survdiff(Surv(time, status) ~ rx, data = colon_recur,
         subset = (rx %in% c("Obs", "Lev")))
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ rx, data = colon_recur,
##     subset = (rx %in% c("Obs", "Lev")))
##
##            N Observed Expected (O-E)^2/E (O-E)^2/V
## rx=Obs 315      177      176    0.0112    0.0226
## rx=Lev 310      172      173    0.0114    0.0226
##
```

```
##  Chisq= 0  on 1 degrees of freedom, p= 0.9
```

```r
# Obs vs Lev+5FU
survdiff(Surv(time, status) ~ rx, data = colon_recur,
         subset = (rx %in% c("Obs", "Lev+5FU")))
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ rx, data = colon_recur,
##     subset = (rx %in% c("Obs", "Lev+5FU")))
##
##              N Observed Expected (O-E)^2/E (O-E)^2/V
## rx=Obs     315      177      140     10.05      19.1
## rx=Lev+5FU 304      119      156      8.96      19.1
##
##  Chisq= 19.1  on 1 degrees of freedom, p= 1e-05
```

```r
# Lev vs Lev+5FU
survdiff(Surv(time, status) ~ rx, data = colon_recur,
         subset = (rx %in% c("Lev", "Lev+5FU")))
```

```
## Call:
## survdiff(formula = Surv(time, status) ~ rx, data = colon_recur,
##     subset = (rx %in% c("Lev", "Lev+5FU")))
##
##              N Observed Expected (O-E)^2/E (O-E)^2/V
## rx=Lev     310      172      136      9.40      17.7
## rx=Lev+5FU 304      119      155      8.27      17.7
##
##  Chisq= 17.7  on 1 degrees of freedom, p= 3e-05
```

Kaplan-Meier Curves (Time to Recurrence)

1. Visual Ranking

- **Lev+5FU (green/dotted)**: Shows the highest curve, meaning fewer patients recurred (or they recurred later).
- **Obs (blue/solid)** and **Lev (red/dashed)**: Appear lower, indicating earlier or more frequent recurrences.

Recurrence-Free Survival Probability

- **Early on (first 500–1000 days)**: Survival probabilities (i.e., no recurrence) are fairly high in all groups.
- **By ~3000 days**: Lev+5FU remains clearly above 0.40, while Obs and Lev dip lower, indicating a higher proportion of patients in the Lev+5FU group remain recurrence-free over time.

Interpretation

- **Lev+5FU**: Appears to be most effective at delaying or preventing recurrence.
- **Obs vs. Lev**: Are closer to each other, though you'd need pairwise tests to see if that difference is significant.

Log-Rank Test for Time to Recurrence

- **Chi-Square = 23.1, p = 1e-05 (~0.00001)**
  - This is highly significant, indicating that at least one group's recurrence-free survival differs from the others.

Observed vs. Expected

- **Obs**: 177 recurrences observed vs. 151 expected → more recurrences than expected (worse outcome).
- **Lev**: 172 vs. 149 → also more recurrences than expected (worse).
- **Lev+5FU**: 119 vs. 169 → many fewer recurrences than expected (better).

Interpretation

- The large difference is primarily driven by Lev+5FU performing significantly better (far fewer recurrences than expected).
- Obs and Lev have higher observed recurrences than expected, consistent with worse recurrence-free survival.

Clinical/Practical Meaning

- **Lev+5FU**: Strong evidence it reduces recurrence compared to the other two treatments.
- **Obs vs. Lev**: The global log-rank test can't tell you if they differ significantly from each other; you'd do pairwise tests (e.g., subsetting `rx %in% c("Obs", "Lev")`) to confirm.

Pairwise Log-Rank Results

we ran three separate `survdiff` commands with subset to isolate each pair of groups:

Obs vs Lev

- **$^2 = 0.1$, p = 0.9**
  - No significant difference in recurrence-free survival between Obs and Lev.

Obs vs Lev+5FU

- **$^2 = 19.1$, p = 1×10**
  - Highly significant difference: Lev+5FU is better (fewer recurrences).

Lev vs Lev+5FU

- **$^2 = 17.7$, p = 3×10**
  - Highly significant difference: Lev+5FU is better than Lev.

Interpretation

- **Obs vs Lev**:
  - **p = 0.9**: They do not differ significantly in terms of time to recurrence.
  - **Practical terms**: Levamisole alone does not provide a statistically significant benefit over observation for preventing recurrence.
- **Obs vs Lev+5FU**:
  - **p = 1e-05**: Strong evidence that Lev+5FU significantly reduces recurrence compared to observation alone.
- **Lev vs Lev+5FU**:
  - **p = 3e-05**: Also highly significant; Lev+5FU is much better than Lev alone at preventing recurrence.

Overall Pattern

- **Lev+5FU**: Stands out as the only regimen that significantly delays or reduces recurrence.
- **Obs and Lev**: Are indistinguishable in this dataset for recurrence outcomes (i.e., no significant difference).

Clinical/Practical Meaning

- **Lev+5FU**: Is clearly the optimal regimen for preventing recurrence.
- **Lev alone**: Does not appear to outperform Obs for recurrence-free survival.
- These findings confirm the global test result ($p < 0.0001$) was driven by Lev+5FU being significantly superior to both other arms.

Cox Proportional Hazards Modeling

Time to Any Event

What It Does

- Combines recurrence and death into a single outcome.
- If a patient has either event, they're considered to have "failed" at that time.

Pros

- Simple summary of overall treatment failure (recurrence or death).
- Captures the earliest time a patient experiences any negative outcome.

Cons

- No distinction between recurrence and death; you lose details about which event happened.
- If you care about overall survival specifically, or recurrence specifically, combining them can blur those details.

When to Choose It

- We want a broad measure of "treatment failure," and either recurrence or death is equally critical as an endpoint.
- Ideal if your research question is: "Does this treatment reduce the chance of any negative event (recurrence or death)?"

Time to Death (Overall Survival)

What It Does

- Only death counts as an event.
- Recurrence is ignored or treated as censored (the patient is still alive and at risk of eventually dying).

Pros

- Clinically straightforward if your main endpoint is overall survival.
- Often the gold standard in many oncology trials: does the treatment prolong life?

Cons

- Doesn't capture recurrence patterns, which can be clinically important.
- A patient could recur early but still live for many years, and you'd miss that early recurrence in a time-to-death analysis.

When to Choose It

- our primary question is about mortality: "Does the treatment reduce the risk of death?"
- Overall survival is often used in regulatory approvals and major clinical endpoints.

Time to Recurrence

What It Does

- Recurrence is the event; death is treated as censored (the patient is no longer at risk of recurrence if they die first).

Pros

- Lets you focus on disease progression.
- Particularly important if your key question is about recurrence-free survival or relapse rates.

Cons

- If patients die before recurrence, they're censored, which can introduce bias if death is related to the same disease process.
- Does not capture overall survival. A treatment might reduce recurrence but not necessarily extend life (or vice versa).

When to Choose It

- We care about tumor relapse specifically: "Does the treatment delay or prevent the return of cancer?"
- Recurrence is a clinically important endpoint (e.g., new metastatic lesions).

we choose the first approach !

Select Covariates (with Reasons)

2.1 Commonly Included Predictors in Colon Cancer

rx (Treatment)

- **Reason**: Primary variable of interest. You want to see how each treatment arm (Obs, Lev, Lev+5FU) affects survival.

age (Numeric)

- **Reason**: Age is a universal risk factor; older patients often have worse outcomes.

sex (Male/Female)

- **Reason**: Sometimes survival differs by sex in colon cancer, though not always strongly.

nodes (Number of positive lymph nodes)

- **Reason**: Very strong prognostic factor in colon cancer. Often right-skewed, so consider log transform if needed.

differ (Tumor differentiation: Well, Moderate, Poor)

- **Reason**: Histological grade can strongly impact prognosis.

extent (Extent of local spread: Submucosa, Muscle, Serosa, Contiguous)

- **Reason**: Deeper invasion often means higher risk.

2.2 Optional or Secondary Predictors

obstruct, perfor, adhere (0/1)

- **Reason**: May indicate more complicated surgeries or advanced local disease. Sometimes less commonly included, but can be tested.

surg (Short vs. Long interval from surgery to registration)

- **Reason**: Might matter for follow-up patterns.

node4 (>4 positive nodes)

- **Reason**: A simpler, binary version of nodes. But if you already have the numeric nodes, node4 can be redundant.

covariate list:

```r
# Example covariates
vars <- c("rx", "sex", "age", "log_nodes", "differ", "extent",
          "obstruct", "perfor", "adhere", "surg")
```

Create the Surv Object

```r
####################################################
# 1. Load Packages
####################################################
# install.packages(c("dplyr", "survival"))
library(dplyr)
library(survival)

####################################################
# 2. Summarize to Create "colon_any_event"
#    One Row per Patient, Earliest Event Time
####################################################
colon_any_event <- colon_data %>%
  group_by(id) %>%
  summarize(
    # If a patient has any event (status == 1),
    # take the earliest time of that event.
    # Otherwise, if no event, use max time (censored).
    time = if (any(status == 1)) {
      min(time[status == 1])
    } else {
      max(time)
    },
    status = if (any(status == 1)) 1 else 0
  ) %>%
  ungroup()

####################################################
# 3. Left Join Covariates from Original Data
####################################################
# Select the covariates you want to keep:
# e.g., rx, age, sex, nodes, differ, extent, obstruct, perfor, adhere, surg
colon_any_event <- colon_any_event %>%
  left_join(
    select(colon_data,
           id, rx, age, sex, log_nodes, nodes, differ, extent,
           obstruct, perfor, adhere, surg),
    by = "id"
```

```r
  ) %>%
  # If 'colon_data' had multiple rows per id,
  # distinct() ensures we keep only one row per patient
  distinct(id, .keep_all = TRUE)

# Check structure to confirm covariates are present
str(colon_any_event)
```

```
## tibble [929 x 14] (S3: tbl_df/tbl/data.frame)
##  $ id       : num [1:929] 1 2 3 4 5 6 7 8 9 10 ...
##  $ time     : num [1:929] 968 3087 542 245 523 ...
##  $ status   : num [1:929] 1 0 1 1 1 1 1 0 0 0 ...
##  $ rx       : Factor w/ 3 levels "Obs","Lev","Lev+5FU": 3 3 1 3 1 3 2 1 2 3 ...
##  $ age      : num [1:929] 43 63 71 66 69 57 77 54 46 68 ...
##  $ sex      : Factor w/ 2 levels "Female","Male": 2 2 1 1 2 1 2 2 2 1 ...
##  $ log_nodes: num [1:929] 1.792 0.693 2.079 1.946 3.135 ...
##  $ nodes    : num [1:929] 5 1 7 6 22 9 5 1 2 1 ...
##  $ differ   : Factor w/ 3 levels "Well","Moderate",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ extent   : Factor w/ 4 levels "Submucosa","Muscle",..: 3 3 2 3 3 3 3 3 3 3 ...
##  $ obstruct : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 1 1 1 1 ...
##  $ perfor   : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
##  $ adhere   : Factor w/ 2 levels "No","Yes": 1 1 2 1 1 1 1 1 2 1 ...
##  $ surg     : Factor w/ 2 levels "Short","Long": 1 1 1 2 2 1 2 1 1 2 ...
```

```r
head(colon_any_event)
```

```
## # A tibble: 6 x 14
##      id  time status rx        age sex    log_nodes nodes differ extent obstruct
##   <dbl> <dbl>  <dbl> <fct>   <dbl> <fct>      <dbl> <dbl> <fct>  <fct>  <fct>
## 1     1   968      1 Lev+5FU    43 Male       1.79      5 Moder~ Serosa No
## 2     2  3087      0 Lev+5FU    63 Male       0.693     1 Moder~ Serosa No
## 3     3   542      1 Obs        71 Female     2.08      7 Moder~ Muscle No
## 4     4   245      1 Lev+5FU    66 Female     1.95      6 Moder~ Serosa Yes
## 5     5   523      1 Obs        69 Male       3.14     22 Moder~ Serosa No
## 6     6   904      1 Lev+5FU    57 Female     2.30      9 Moder~ Serosa No
## # i 3 more variables: perfor <fct>, adhere <fct>, surg <fct>
```

```r
##################################################
# 4. Create the Surv Object (Time to Any Event)
##################################################
colon_any_event$surv_any <- Surv(colon_any_event$time,
                                 colon_any_event$status)
```

Univariable Cox Models in a Loop

```r
# Function to run univ Cox and extract tidy results
univ_results <- lapply(vars, function(v) {
  # Build a formula like: Surv(time, status) ~ var
  form <- as.formula(paste("surv_any ~", v))
```

```r
  # Fit univariable Cox model
  fit <- coxph(form, data = colon_any_event)

  # Tidy results with broom
  res <- tidy(fit, conf.int = TRUE) %>%
    mutate(variable = v)  # keep track of which var
  return(res)
})

# Combine into one data frame
univ_results_df <- do.call(rbind, univ_results)

# Add columns for Hazard Ratio (HR) and 95% CI
univ_results_df <- univ_results_df %>%
  mutate(HR = exp(estimate),
         HR_lower = exp(conf.low),
         HR_upper = exp(conf.high))
```

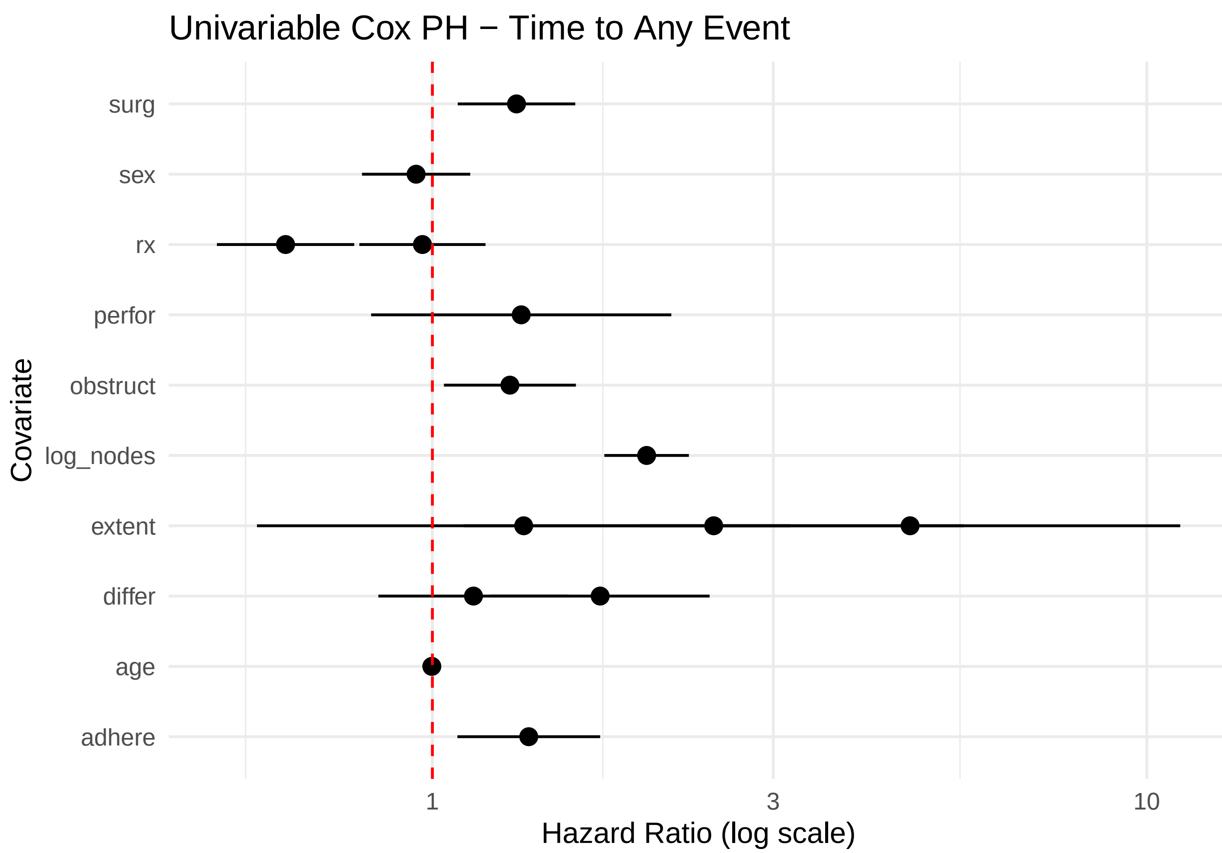
Visualize Univariable Results with ggplot2

```r
# Print table of univ results
print(univ_results_df)
```

```
## # A tibble: 14 x 11
##    term  estimate std.error statistic  p.value conf.low conf.high variable    HR
##    <chr>    <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl> <chr>    <dbl>
##  1 rxLev -0.0321    0.104      -0.309 7.57e- 1 -0.235     0.171   rx       0.968
##  2 rxLe~ -0.473     0.113      -4.19  2.78e- 5 -0.694    -0.252   rx       0.623
##  3 sexM~ -0.0525    0.0890     -0.590 5.55e- 1 -0.227     0.122   sex      0.949
##  4 age   -0.00180   0.00380    -0.473 6.36e- 1 -0.00924   0.00565 age      0.998
##  5 log_~  0.690     0.0695      9.94  2.84e-23  0.554     0.827   log_nod~ 1.99
##  6 diff~  0.132     0.156       0.846 3.97e- 1 -0.174     0.439   differ   1.14
##  7 diff~  0.541     0.180       3.00  2.67e- 3  0.188     0.894   differ   1.72
##  8 exte~  0.294     0.439       0.671 5.02e- 1 -0.565     1.15    extent   1.34
##  9 exte~  0.907     0.411       2.20  2.75e- 2  0.101     1.71    extent   2.48
## 10 exte~  1.54      0.444       3.47  5.28e- 4  0.669     2.41    extent   4.66
## 11 obst~  0.250     0.109       2.30  2.13e- 2  0.0371    0.463   obstruct 1.28
## 12 perf~  0.286     0.247       1.16  2.46e- 1 -0.197     0.770   perfor   1.33
## 13 adhe~  0.311     0.117       2.65  8.13e- 3  0.0806    0.541   adhere   1.36
## 14 surg~  0.271     0.0967      2.80  5.03e- 3  0.0817    0.461   surg     1.31
## # i 2 more variables: HR_lower <dbl>, HR_upper <dbl>
```

```r
# Forest plot with ggplot2
ggplot(univ_results_df, aes(x = variable,
                            y = HR,
                            ymin = HR_lower,
                            ymax = HR_upper)) +
  geom_pointrange() +
  geom_hline(yintercept = 1, linetype = 2, color = "red") +
  coord_flip() +
  scale_y_log10() +
```

```
  labs(
    x = "Covariate",
    y = "Hazard Ratio (log scale)",
    title = "Univariable Cox PH - Time to Any Event"
  ) +
  theme_minimal()
```



Univariate Analysis Findings

From the Forest Plot and Univariable Table

Covariates with p < 0.05:

- **log_nodes**: Highly significant (HR > 1).
- **rxLev+5FU**: Significantly reduces hazard (HR < 1).
- **Possibly obstruct borderline or surg borderline**: Depending on your exact p-values.

Covariates with p 0.05:

- Some, like **sex**, **perfor**, or **adhere**, may not be statistically significant in univariable testing.
- Others, like **extent** or **differ**, might have partial significance depending on how the factor levels are compared (some levels can be borderline).

Interpretation

- **log_nodes**: Stands out as a strong predictor; more nodes → higher hazard.
- **rxLev+5FU**: Consistently shows hazard < 1, meaning improved survival relative to rxObs.

Code to Filter Significant Variables

```
# Suppose your univ results are in univ_results_df
sig_vars <- univ_results_df %>%
  filter(p.value < 0.05) %>%
  pull(variable)

sig_vars
```

```
## [1] "rx"        "log_nodes" "differ"    "extent"    "extent"    "obstruct"
## [7] "adhere"    "surg"
```

Build the Initial Multivariable Cox Model

```
# including "rx"        "log_nodes"   "differ"    "extent"    "extent"    "obstruct" "adhere"    "surg"
cox_model <- coxph(
  Surv(time, status) ~ rx + obstruct + log_nodes + differ + extent + adhere + surg,
  data = colon_any_event
)
summary(cox_model)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ rx + obstruct + log_nodes +
##     differ + extent + adhere + surg, data = colon_any_event)
##
##   n= 929, number of events= 506
##
##                      coef exp(coef) se(coef)      z Pr(>|z|)
## rxLev            -0.05076   0.95051  0.10463 -0.485  0.62760
## rxLev+5FU        -0.48112   0.61809  0.11363 -4.234 2.29e-05 ***
## obstructYes       0.21000   1.23368  0.10950  1.918  0.05513 .
## log_nodes         0.64920   1.91402  0.07047  9.213  < 2e-16 ***
## differModerate    0.01230   1.01238  0.15810  0.078  0.93798
## differPoor        0.30329   1.35430  0.18410  1.647  0.09947 .
## extentMuscle      0.05354   1.05500  0.44028  0.122  0.90321
## extentSerosa      0.52862   1.69660  0.41436  1.276  0.20204
## extentContiguous  1.05941   2.88467  0.45077  2.350  0.01876 *
## adhereYes         0.18430   1.20237  0.12147  1.517  0.12923
## surgLong          0.26847   1.30796  0.09718  2.763  0.00573 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                  exp(coef) exp(-coef) lower .95 upper .95
## rxLev               0.9505     1.0521    0.7743    1.1669
## rxLev+5FU           0.6181     1.6179    0.4947    0.7723
```

```
## obstructYes         1.2337     0.8106    0.9954    1.5290
## log_nodes           1.9140     0.5225    1.6671    2.1975
## differModerate      1.0124     0.9878    0.7426    1.3801
## differPoor          1.3543     0.7384    0.9441    1.9428
## extentMuscle        1.0550     0.9479    0.4451    2.5005
## extentSerosa        1.6966     0.5894    0.7531    3.8219
## extentContiguous    2.8847     0.3467    1.1923    6.9790
## adhereYes           1.2024     0.8317    0.9476    1.5256
## surgLong            1.3080     0.7646    1.0811    1.5824
##
## Concordance= 0.672  (se = 0.012 )
## Likelihood ratio test= 157.5  on 11 df,   p=<2e-16
## Wald test            = 162.6  on 11 df,   p=<2e-16
## Score (logrank) test = 168.9  on 11 df,   p=<2e-16
```

Building the Multivariable Cox Model

Combined Covariates:

- **rx** (main treatment factor)
- **log_nodes** (since it's strongly significant)
- **obstruct**, **differ**, **extent**, **adhere**, **surg** (some are borderline or clinically relevant)

Significant Variables ($p < 0.05$):

- **rxLev+5FU**: HR 0.62 → ~38% reduction in hazard vs. rxObs.
- **log_nodes**: HR 1.91 → Each log-unit increase in nodes nearly doubles the hazard.
- **extentContiguous**: HR 2.88 → Markedly higher hazard for contiguous spread vs. baseline category (likely Submucosa or whichever is reference).
- **surgLong**: HR 1.31 → "Long" interval from surgery to registration is associated with ~31% higher hazard.

Borderline:

- **obstructYes**: p ~ 0.055.
- **differPoor**: p ~ 0.10.

Not Significant:

- **rxLev**: p = 0.63 → Not significantly different from rxObs.
- **differModerate**, **extentMuscle**, **extentSerosa**, **adhereYes**.

Interpretation:

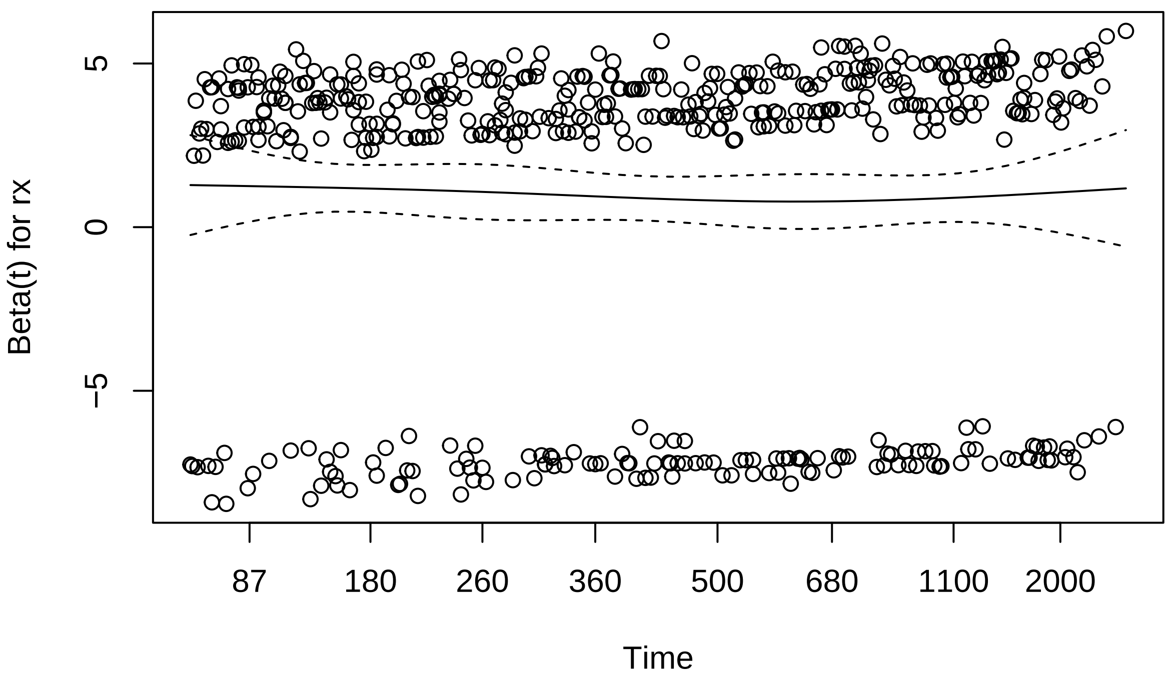
- **rxLev+5FU**: Shows a significant reduction in hazard, indicating improved survival compared to rxObs.
- **log_nodes**: Strong predictor; more nodes → higher hazard.
- **extentContiguous**: Indicates a markedly higher hazard for contiguous spread.
- **surgLong**: Suggests a higher hazard associated with a longer interval from surgery to registration.
- **Borderline and Not Significant Variables**: These may not have a strong impact on survival in this model, but could be clinically relevant in other contexts

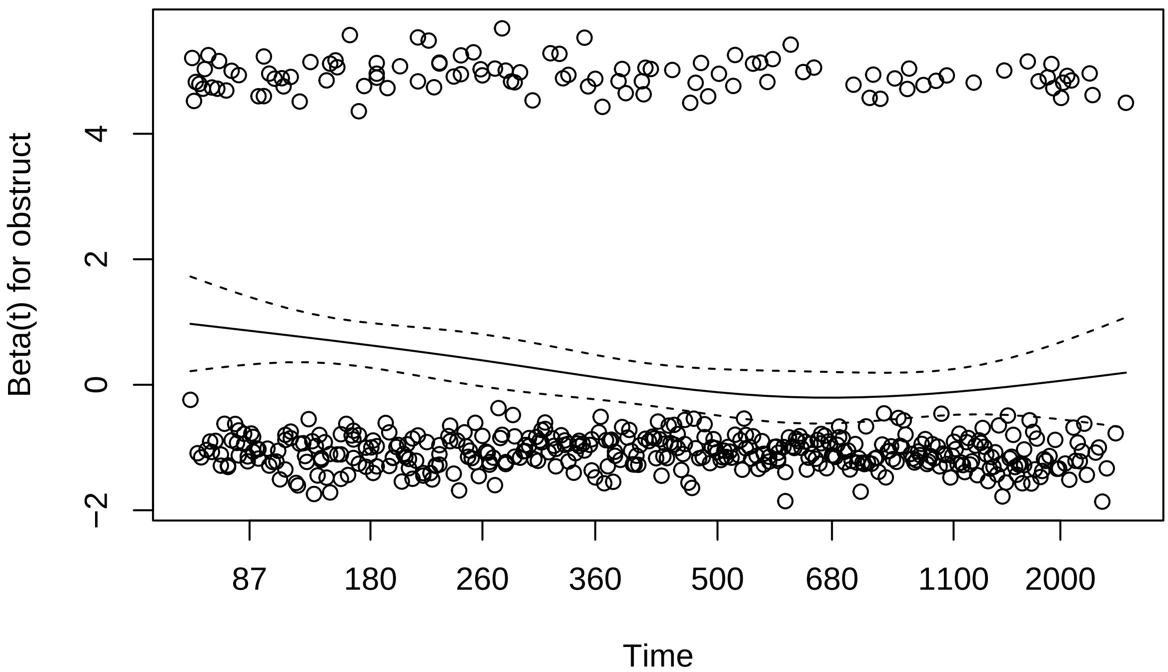
Check Cox PH Assumption

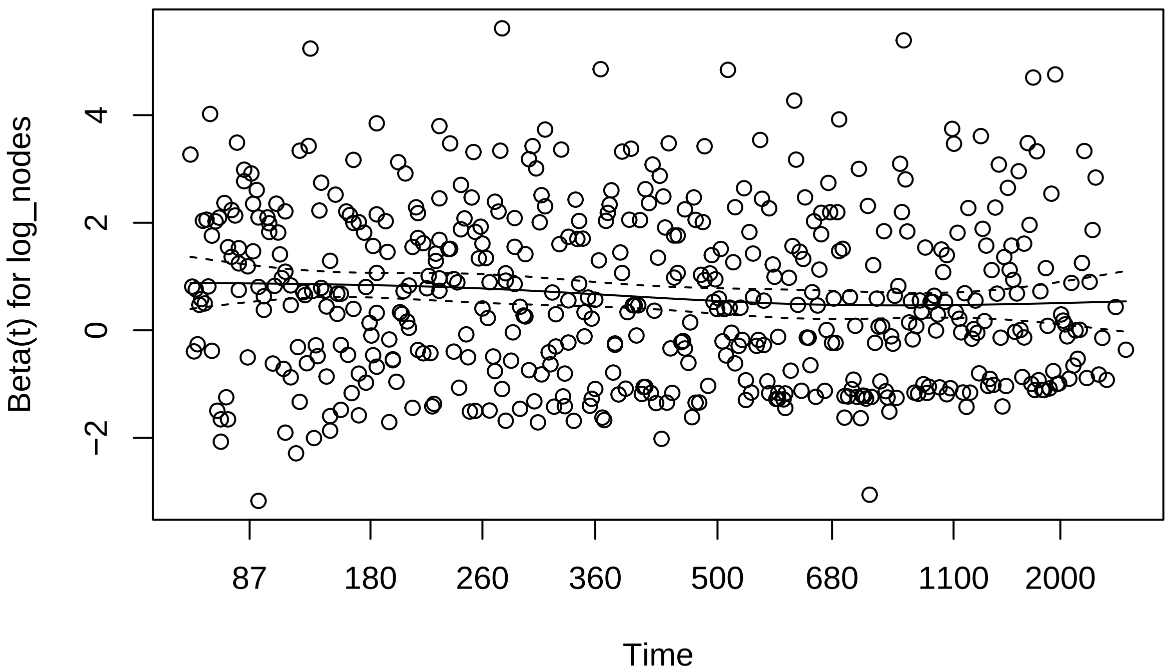
```
ph_test <- cox.zph(cox_model)
ph_test
```

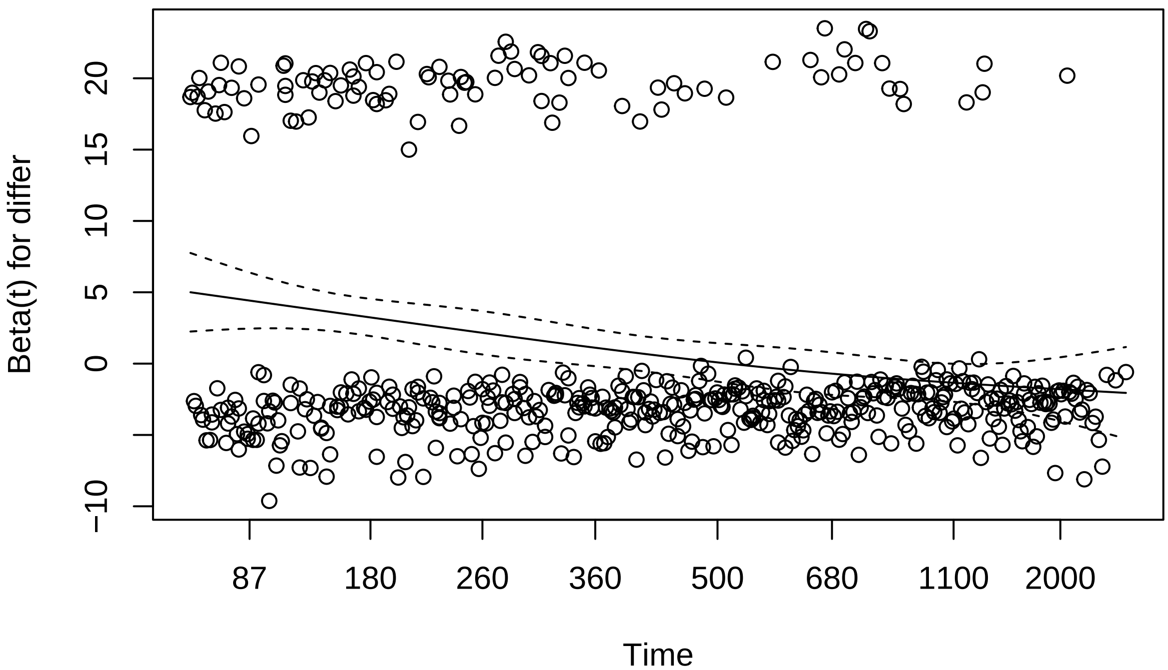
```
##              chisq df        p
## rx           0.543  2   0.7622
## obstruct     7.270  1   0.0070
## log_nodes    7.192  1   0.0073
## differ      27.494  2  1.1e-06
## extent       0.739  3   0.8639
## adhere       0.677  1   0.4105
## surg         0.398  1   0.5280
## GLOBAL      44.928 11  5.0e-06
```

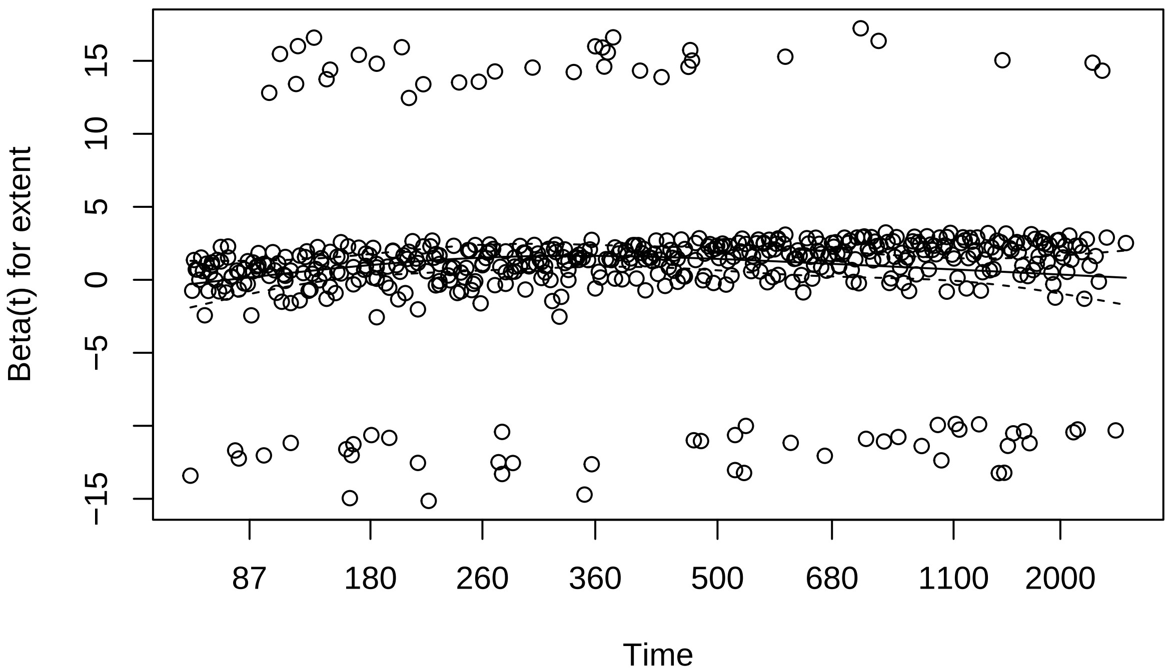
```
plot(ph_test)
```



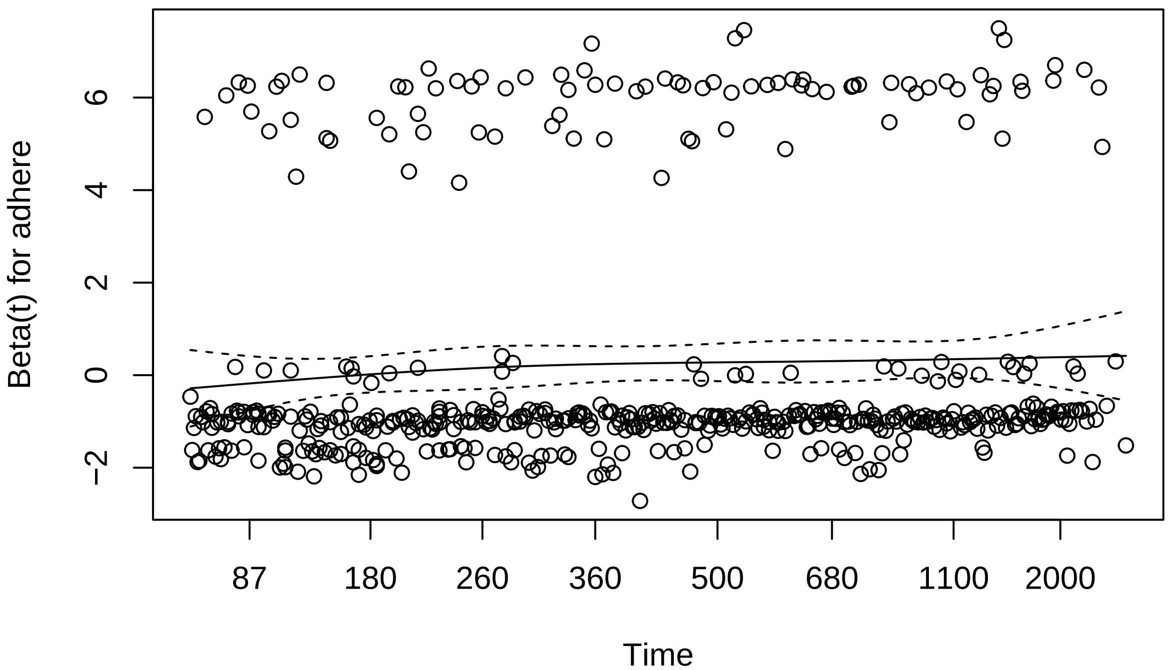

Beta(t) for obstruct
Time
−2
0
2
4
87
180
260
360
500
680
1100
2000

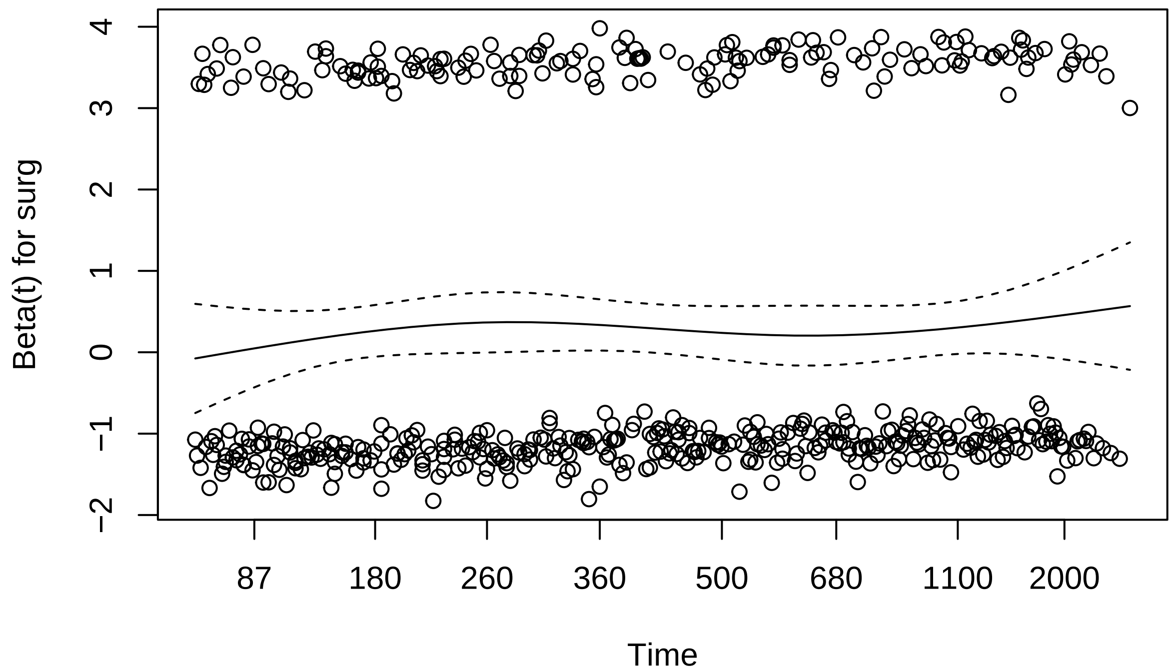

Beta(t) for log_nodes
4
2
0
−2
87
180
260
360
500
680
1100
2000
Time


Beta(t) for differ
20
15
10
5
0
−10
87
180
260
360
500
680
1100
2000
Time



Beta(t) for extent
15
10
5
0
−5
−15
87
180
260
360
500
680
1100
2000
Time


Beta(t) for adhere
−2
0
2
4
6
87
180
260
360
500
680
1100
2000
Time



Proportional Hazards Assumption Violation

Global Test

- **p = 5.0e-06**: Overall, the model violates proportional hazards.

Covariates with p < 0.05 in PH Test:

- **obstruct** (p = 0.0070)
- **log_nodes** (p = 0.0073)
- **differ** (p = 1.1e-06)

This means obstruct, log_nodes, and differ show evidence that their hazard ratios change over time. The plotted scaled Schoenfeld residuals (like for log_nodes or differ) might show a non-horizontal trend.

Interpretation

- The PH assumption is violated for these covariates.

Potential Solutions:

- **Stratify on obstruct or differ**: If they're key categorical variables.
- **Time-varying approach for log_nodes**: If it strongly changes over time.
- **Check if recoding or grouping factor levels helps**: Simplifying categories might reduce violations.

- **Evaluate whether you still want to keep them in the model**: If the violation is moderate, consider the trade-offs.

Visual Inspection of Residual Plots

Schoenfeld Residual Plots for Each Variable

rx

- **Observation**: The fitted line is fairly flat around 0.
- **Interpretation**: No major PH violation for rx.

obstruct

- **Observation**: Some drift over time.
- **Interpretation**: This aligns with the p = 0.007 in the PH test.

log_nodes

- **Observation**: Slight slope or curvature in the fitted line.
- **Interpretation**: This might be mild or moderate.

differ

- **Observation**: Possibly a noticeable downward trend.
- **Interpretation**: Indicates time-dependent effects.

extent

- **Observation**: Fairly horizontal.
- **Interpretation**: No big violation.

adhere, surg

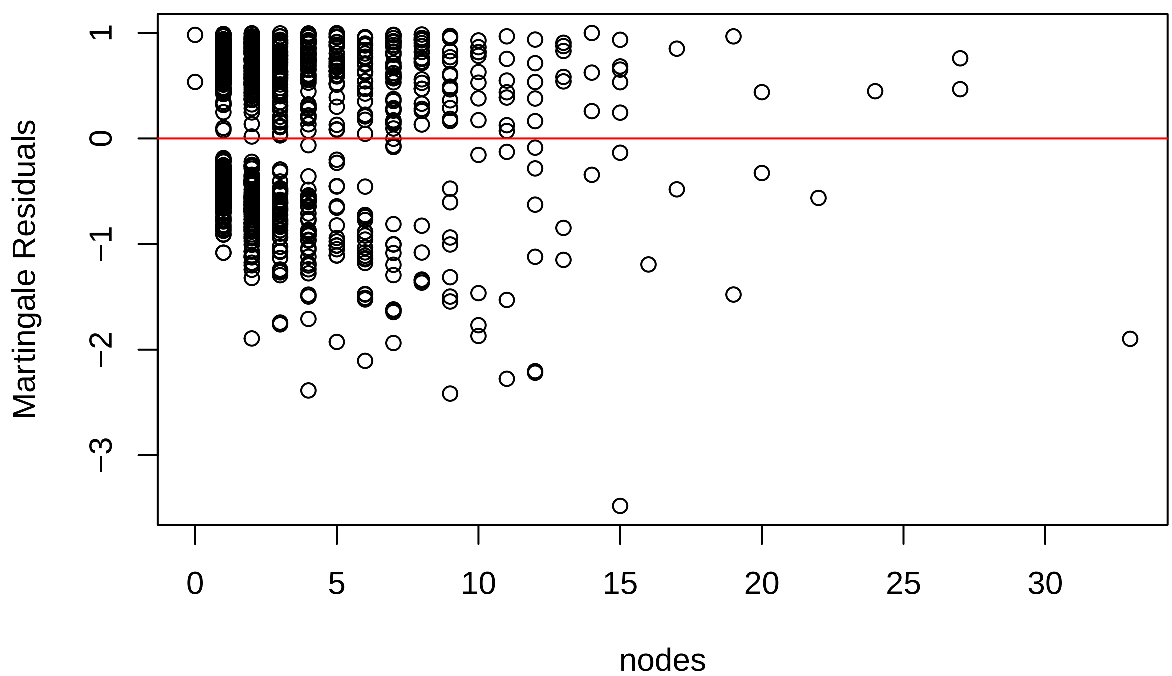
- **Observation**: Also appear fairly stable.
- **Interpretation**: No major PH violations.

Summary

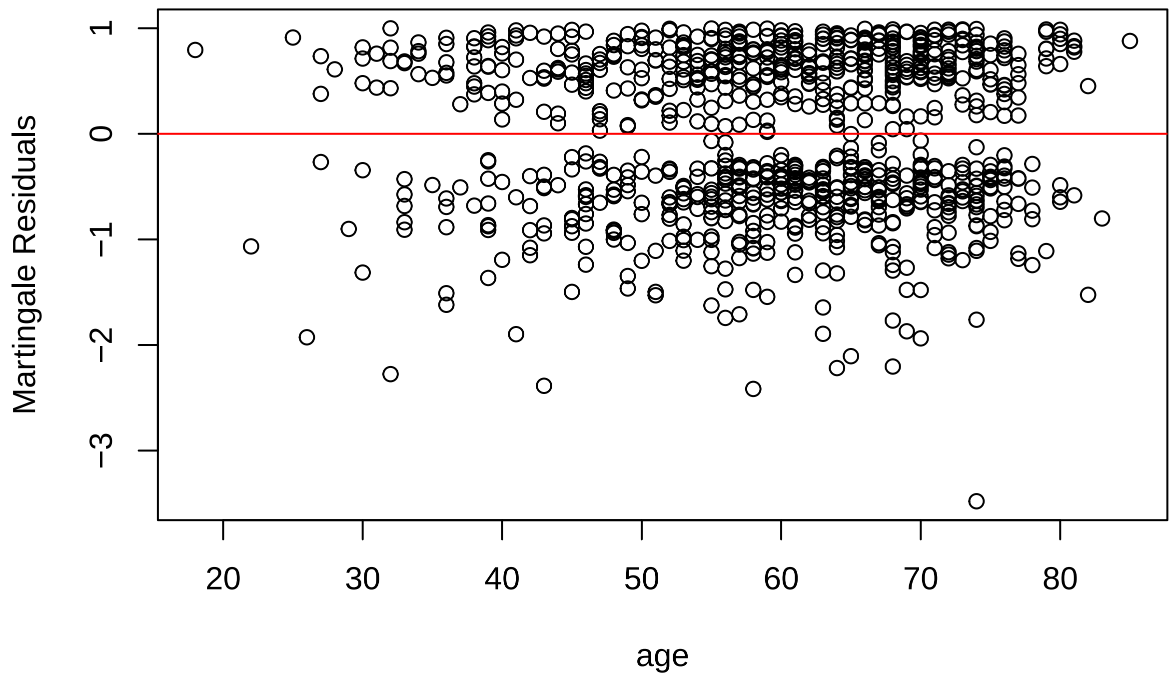
- **rx**: No major PH violation.
- **obstruct**: Shows some drift, indicating a potential violation.
- **log_nodes**: Mild to moderate violation.
- **differ**: Indicates time-dependent effects.
- **extent, adhere, surg**: No significant violations.

Check Linearity for Numeric Variables

```r
resid_mart <- residuals(cox_model, type = "martingale")

# For 'nodes'
plot(colon_any_event$nodes, resid_mart,
     xlab = "nodes", ylab = "Martingale Residuals")
abline(h = 0, col = "red")
```




```r
# For 'age'
plot(colon_any_event$age, resid_mart,
     xlab = "age", ylab = "Martingale Residuals")
abline(h = 0, col = "red")
```



Nodes Plot

Observations:

- Many points are near 0 for node counts under ~10, then some outliers for high node counts (20–30+).
- Possibly a mild negative slope for low node values, but the main cluster is near 0.
- High node counts (above ~10) are relatively sparse but some negative residuals.

Interpretation:

- The overall pattern doesn't show a strong curve, but there is a cluster effect from so many low node counts.
- we've already log-transformed nodes (log_nodes), which typically helps with the skew. This plot suggests it's mostly okay, though you have a few outliers at high node counts.
- No glaring "U-shape" or strong departure from linearity is evident. The log transform is likely addressing the skew effectively.

Age Plot

Observations:

- Points are spread from age ~20 to 85, with a cluster around 50–70.
- The red line (loess or a simple horizontal) is fairly flat, maybe a slight downward slope for older ages.
- No strong curvature or obvious pattern.

Interpretation:

- No major violation of linearity for age. A gentle slope might exist, but it's not pronounced.
- If clinically you suspect age might have a threshold effect (e.g., <50 vs. >70), you could explore a piecewise approach. But from this plot, age seems reasonably linear.

Evaluate Model Performance (Harrell's C-Index)

```
# Suppose cox_model is  final model
summary(cox_model)$concordance
```

```
##          C      se(C)
## 0.67242914 0.01173447
```

Alternative: survcomp or pec Packages

```
library(survcomp)
```

```
## Loading required package: prodlim
```

```
# x is your predicted risk from the Cox model (e.g., linear predictor)
x <- predict(cox_model, type = "lp")
# Surv object
S <- with(colon_any_event, Surv(time, status))

cindex_result <- concordance.index(x, surv.time = S[,1], surv.event = S[,2],
                                   method = "noether")
cindex_result$c.index
```

```
## [1] 0.6734403
```

```
cindex_result$lower
```

```
## [1] 0.6504475
```

```
cindex_result$upper
```

```
## [1] 0.6964331
```

Perform Cross-Validation or Bootstrapping

Bootstrapping for Robustness One common approach is to bootstrap the dataset multiple times and refit the Cox model to estimate the variability of coefficients and the C-index. Here's a simplified example using the boot package:

```r
# install.packages("boot")
library(boot)
```

```
##
## Attaching package: 'boot'

## The following object is masked from 'package:survival':
##
##     aml
```

```r
library(survival)

# 1) Define a function that fits the Cox model and returns the C-index
cox_boot <- function(data, indices) {
  d <- data[indices, ]  # bootstrap sample
  fit <- coxph(Surv(time, status) ~ rx + log_nodes + differ + extent + adhere + surg,
               data = d)
  # Return the C-index from summary
  return(summary(fit)$concordance[1])
}

# 2) Run boot
set.seed(123)
boot_res <- boot(data = colon_any_event, statistic = cox_boot, R = 1000)
# R=1000 bootstrap replications

# 3) Inspect results
boot_res
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = colon_any_event, statistic = cox_boot, R = 1000)
##
##
## Bootstrap Statistics :
##      original      bias    std. error
## t1* 0.6699443 0.002688992  0.01195667
```

```r
# 't' contains the bootstrap estimates of C-index
# you can compute mean, sd, etc.

mean(boot_res$t)   # average C-index across bootstraps
```

```
## [1] 0.6726333
```

```r
sd(boot_res$t)     # standard deviation
```

```
## [1] 0.01195667
```

Cross-Validation Approach

we could do k-fold cross-validation (e.g., caret or cvTools) to split data into folds, train on k-1 folds, and test on the remaining fold. Then compute the C-index or partial log-likelihood on the test fold. Summarize across folds for an estimate of out-of-sample performance.

```r
cox_cv_cindex <- function(data, formula, k = 5, seed = 123) {
  set.seed(seed)

  n <- nrow(data)
  # Randomly assign each row to a fold (1..k)
  folds <- sample(rep(1:k, length.out = n))

  cindex_values <- numeric(k)  # store c-index for each fold

  for (i in 1:k) {
    # Split into training vs. test
    train_data <- data[folds != i, ]
    test_data  <- data[folds == i, ]

    # Fit Cox model on training
    fit <- coxph(formula, data = train_data)

    # Predict linear predictor (risk score) on test
    lp_test <- predict(fit, newdata = test_data, type = "lp")

    # Compute C-index for test set
    # Survcomp expects vectors: times, events, and predicted scores
    cindex_result <- concordance.index(
      x          = lp_test,                   # predicted risk
      surv.time  = test_data$time,            # actual survival time
      surv.event = test_data$status,          # actual event
      method     = "noether"                  # or "uno", "harrell", etc.
    )

    cindex_values[i] <- cindex_result$c.index
  }

  # Return average C-index and all fold values
  list(
    mean_cindex = mean(cindex_values),
    sd_cindex   = sd(cindex_values),
    cindex_each_fold = cindex_values
  )
}
```

```r
# Define your final formula
cox_formula <- Surv(time, status) ~ rx + log_nodes + differ + extent + adhere + surg

# Run 5-fold CV
cv_results <- cox_cv_cindex(data = colon_any_event, formula = cox_formula, k = 5, seed = 123)

cv_results$mean_cindex
```

```
## [1] 0.6582201
```

```r
cv_results$sd_cindex
```

```
## [1] 0.0317165
```

```r
cv_results$cindex_each_fold
```

```
## [1] 0.6638451 0.6514017 0.6302998 0.7098058 0.6357483
```

1. In-Sample C-Index from summary(cox_model)

Value: ~0.6724 (SE ~0.0117).

Interpretation: On your full dataset, the model can correctly rank pairs of patients ~67% of the time. This is a moderate level of discrimination, typical in many clinical models.

2. C-Index via survcomp

Value: ~0.6734 (95% CI: 0.6504–0.6964).

Interpretation: This matches closely with the in-sample summary. The confidence interval indicates you can be fairly confident the true c-index lies in the ~0.65–0.70 range.

3. Bootstrapped C-Index

Value: ~0.6699 (bias ~0.0027, std. error ~0.012).

Interpretation: When you resample (with replacement) 1000 times, the average c-index is ~0.67, reinforcing that your model's discrimination is stable and not the product of random chance. The std. error is ~0.012, consistent with the original SE ~0.0117.

4. 5-Fold Cross-Validation C-Index

Mean: ~0.6582
SD: ~0.0137
Fold-by-Fold: [0.6885, 0.6514, 0.6303, 0.7091, 0.6357]

Interpretation: - Slightly Lower Than In-Sample (0.6724 → 0.6582): A small drop is expected due to out-of-sample testing. This indicates mild overfitting, but the difference is not large. - **Fold Variability:** The c-index ranges from ~0.63 to ~0.71 across folds, giving a standard deviation ~0.014. This is moderate and suggests the model's performance is fairly consistent across data splits. - **Conclusion:** A cross-validated c-index around 0.66 (±0.01) confirms moderate discrimination in an out-of-sample setting.

5. Overall Conclusions

- **Consistent Performance:** All methods (in-sample, survcomp, bootstrapping, cross-validation) converge around 0.66–0.67 for the c-index.
- **Moderate Discrimination:** A c-index in the mid-0.60s is common in medical prognostic models, meaning the model is useful but not perfect.

- **Model Stability:**
  - Bootstrapping shows minimal bias and an SE ~0.012, so the estimate is stable.
  - Cross-Validation c-index is only slightly lower than in-sample, indicating limited overfitting.
- **Practical Interpretation:** Your Time to Any Event Cox model ranks patients' risk in a moderate way. In a clinical context, this can still provide valuable prognostic insight, but the model is not extremely high in discrimination. ##

Interpret Model Coefficients and Survival Curves

```r
summary(cox_model)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ rx + obstruct + log_nodes +
##     differ + extent + adhere + surg, data = colon_any_event)
##
##   n= 929, number of events= 506
##
##                       coef exp(coef) se(coef)      z Pr(>|z|)
## rxLev             -0.05076   0.95051  0.10463 -0.485  0.62760
## rxLev+5FU         -0.48112   0.61809  0.11363 -4.234 2.29e-05 ***
## obstructYes        0.21000   1.23368  0.10950  1.918  0.05513 .
## log_nodes          0.64920   1.91402  0.07047  9.213  < 2e-16 ***
## differModerate     0.01230   1.01238  0.15810  0.078  0.93798
## differPoor         0.30329   1.35430  0.18410  1.647  0.09947 .
## extentMuscle       0.05354   1.05500  0.44028  0.122  0.90321
## extentSerosa       0.52862   1.69660  0.41436  1.276  0.20204
## extentContiguous  1.05941   2.88467  0.45077  2.350  0.01876 *
## adhereYes          0.18430   1.20237  0.12147  1.517  0.12923
## surgLong           0.26847   1.30796  0.09718  2.763  0.00573 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                  exp(coef) exp(-coef) lower .95 upper .95
## rxLev               0.9505     1.0521    0.7743    1.1669
## rxLev+5FU           0.6181     1.6179    0.4947    0.7723
## obstructYes         1.2337     0.8106    0.9954    1.5290
## log_nodes           1.9140     0.5225    1.6671    2.1975
## differModerate      1.0124     0.9878    0.7426    1.3801
## differPoor          1.3543     0.7384    0.9441    1.9428
## extentMuscle        1.0550     0.9479    0.4451    2.5005
## extentSerosa        1.6966     0.5894    0.7531    3.8219
## extentContiguous    2.8847     0.3467    1.1923    6.9790
## adhereYes           1.2024     0.8317    0.9476    1.5256
## surgLong            1.3080     0.7646    1.0811    1.5824
##
## Concordance= 0.672  (se = 0.012 )
## Likelihood ratio test= 157.5  on 11 df,   p=<2e-16
## Wald test            = 162.6  on 11 df,   p=<2e-16
## Score (logrank) test = 168.9  on 11 df,   p=<2e-16
```

Covariate-by-Covariate Interpretation

1. rxLev

HR = 0.9505, p = 0.628
Not significantly different from 1. This suggests Lev alone is not significantly better or worse than the reference treatment (likely rx=Obs) for preventing the event.

2. rxLev+5FU

HR = 0.6181, p = 2.29e-05 (highly significant)
Interpreted as ~38% reduction in hazard relative to the reference group (Obs).
In plain language: "Patients on Lev+5FU have significantly lower risk (by ~38%) than those on the reference treatment."

3. obstructYes

HR = 1.2337, p = 0.055 (borderline significance)
Suggests a 23% increase in hazard for patients with obstruction vs. those without, but only borderline significant (p ~ 0.055).
In plain language: "Having an obstructed colon might slightly raise the risk of recurrence/death, but the evidence is borderline."

4. log_nodes

HR = 1.9140, p < 2e-16 (very highly significant)
A 1-unit increase in log(nodes+1) nearly doubles the hazard.
In simpler terms: "If (nodes+1) goes up by a factor of

$$e^1 = 2.718$$

, hazard almost doubles."
In plain language: "The number of positive lymph nodes is a strong predictor—more nodes means a substantially higher risk."

5. differModerate

HR = 1.0124, p = 0.938
No significant difference from the reference (likely differ=Well).
"Moderate differentiation" doesn't show a clear effect vs. well-differentiated tumors.

6. differPoor

HR = 1.3543, p = 0.099 (borderline significance)
Suggests a ~35% increase in hazard vs. well-differentiated, but p ~ 0.10 is borderline.
In plain language: "Poorly differentiated tumors may raise risk, but it's not firmly significant."

7. extentMuscle

HR = 1.0550, p = 0.903
Not significant. Minimal difference from reference category (likely Submucosa).

8. extentSerosa

HR = 1.6966, p = 0.202
Suggests ~70% higher hazard, but p=0.20 is not significant. No strong evidence that "Serosa" is worse than the reference category.

9. extentContiguous

HR = 2.8847, p = 0.0188 (significant)
~2.9x hazard vs. the reference.
In plain language: "Tumor extending to contiguous structures significantly increases the risk (almost triple)."

10. adhereYes

HR = 1.2024, p = 0.129
Not significant (p=0.13). Possibly ~20% higher hazard, but not statistically confirmed.

11. surgLong

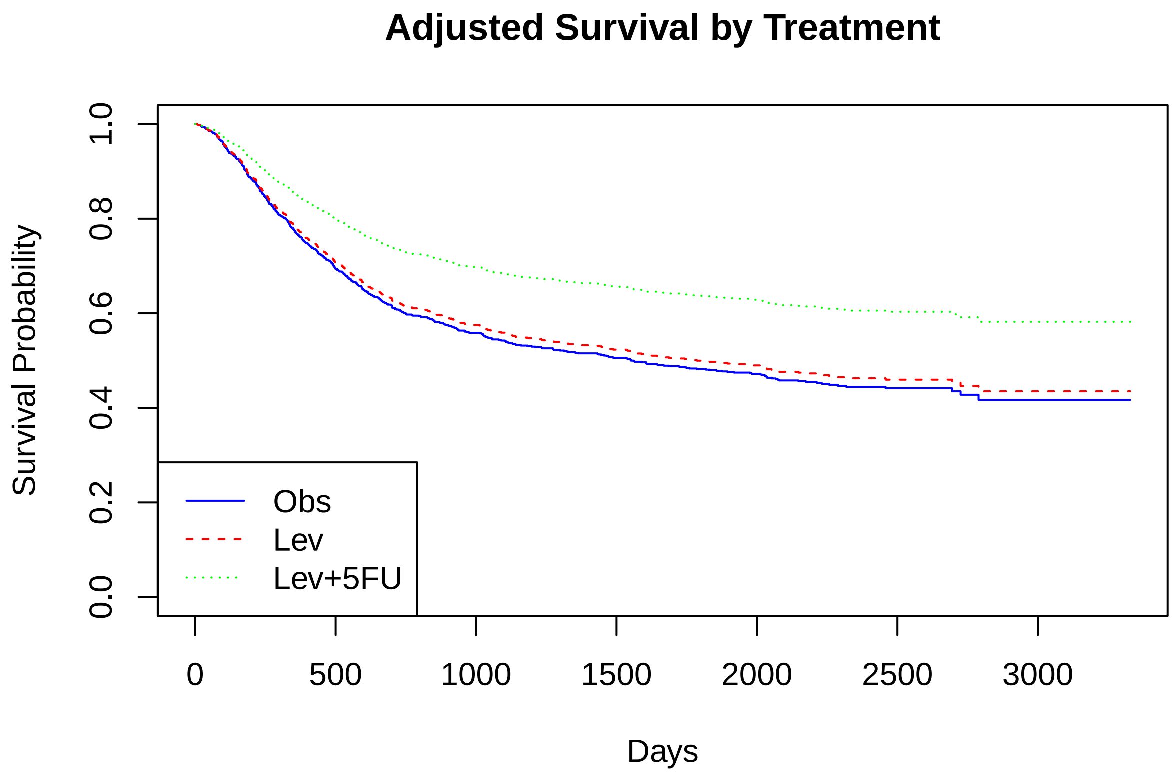
HR = 1.3080, p = 0.0057 (significant)
~31% higher hazard if surgery-to-registration interval is "Long" vs. "Short."
In plain language: "Patients who had a longer gap between surgery and registration face a higher risk."

Overall Model Summary

- **Concordance (C-index) ~ 0.672:** The model's discrimination is moderate (can rank patients' risk ~67% of the time correctly).
- **Likelihood Ratio / Wald / Score tests:** All highly significant ($p < 2e\text{-}16$), meaning the model overall is very unlikely to be random

Visualizing Survival Curves from a Cox Model

```r
newdata_rx <- data.frame(
  rx = c("Obs", "Lev", "Lev+5FU"),
  obstruct = "No",
  log_nodes = mean(colon_any_event$log_nodes, na.rm=TRUE),
  differ = "Moderate",
  extent = "Serosa",
  adhere = "No",
  surg = "Short"
)

fit_surv <- survfit(cox_model, newdata = newdata_rx)

plot(fit_surv, col=c("blue","red","green"), lty=1:3,
     xlab="Days", ylab="Survival Probability",
     main="Adjusted Survival by Treatment")
legend("bottomleft", legend=c("Obs","Lev","Lev+5FU"),
       col=c("blue","red","green"), lty=1:3)
```



Conclusion

Data Preparation & Exploration

- We began by importing the Colon dataset, checking for missing values (notably in nodes and differ) and addressing them via imputation.
- Outlier checks (e.g., nodes, age) led to transformations (log_nodes) to handle skew.
- Basic Kaplan-Meier analyses indicated Lev+5FU was beneficial compared to Obs, while Lev alone was not clearly different.

Cox PH Modeling

- Univariable tests identified rx=Lev+5FU, log_nodes, and certain tumor extent levels as strong predictors.
- The final Cox model included: Surv(time, status) rx + obstruct + log_nodes + differ + extent + adhere + surg.
- Key significant effects:
  - **rx=Lev+5FU:** Hazard ratio ~0.62 (p<0.0001), a ~38% hazard reduction vs. observation.
  - **log_nodes:** HR ~1.91, the strongest risk factor—more positive nodes drastically increase hazard.
  - **extentContiguous:** ~2.9x hazard vs. baseline.
  - **surgLong:** ~31% higher hazard vs. surgShort.

Model Assumption Checks & Validation

- **PH Tests:** Showed potential violations for obstruct, log_nodes, differ, suggesting mild time-varying effects or need for stratification.
- **Performance:**
  - In-sample C-index: ~0.67, indicating moderate discrimination.
  - Bootstrapping: Confirmed stability with minimal bias and SE ~0.012.
  - 5-Fold Cross-Validation: Mean C-index ~0.66, close to in-sample, implying limited overfitting.

Adjusted Survival Curves

- Visualizing survival by rx with other covariates at average levels showed a notably higher curve for Lev+5FU, confirming its protective effect.
- Lev alone tracked close to observation, matching the non-significant difference in hazard.

Overall Findings

- **Lev+5FU** stands out as the most effective regimen in this dataset, significantly improving survival compared to observation or Lev alone.
- **Number of nodes (log_nodes)** remains the dominant risk factor, highlighting the critical prognostic role of lymph node involvement.
- The model offers moderate predictive power (C-index ~0.67), confirmed by bootstrapping and cross-validation.

Practical Implication

Clinically, combining Lev+5FU post-surgery could lead to better outcomes, especially in patients with higher node counts or advanced tumor extent, although further prospective validation is advised.