R code for Survival Lab #1

Spring 2025

Contents

1 Background

Data Description:

This dataset contains information on 90 male patients diagnosed with laryngeal cancer between 1970 and 1978 at a Dutch hospital. The follow-up period extends until January 1, 1993, and the primary outcome is the time from initial treatment to death. Also recorded are the patient's age at diagnosis and stage of patient's cancer at enrollment. The four stages are based on the T.N.M. (primary tumor (T), nodal involvement (N) and distant metastasis (M) grading) classification:

- Stage I: T1 N0 M0
- Stage II: T2 N0 M0
- Stage III: T3 N0 M0 and Tx N1 M0 (where x = 1, 2, 3)
- Stage IV: all other combinations

Data Dictionary

Variable	Description	Values
futime	Time to death (in years)	Continuous
died	Mortality status	0 = censored, 1 = death
stage	Disease stage	I, II, III, IV (TNM-based)
age	Age at diagnosis	

Table 1: Data Dictionary

This R code in this document demonstrates various analyses for the larynx_M dataset in the context of Survival Lab #1.

2 Data

Load all the R packages needed for Survival Lab #1

```
packages_to_load <- c("haven", "survival", "gtsummary", "survminer", "biostat3")
lapply(packages_to_load, library, character.only = TRUE)
```

Load the dataset and set the stage as a factor variable with four levels:

```
# Read in the dataset
larynx_M <- read_dta("larynx_M.dta")
# Quick look at the data, Make sure variable names match what you expect
head(larynx_M)
summary(larynx_M)

# Convert stage to an ordered factor: 1 (Stage I) to 4 (Stage IV)
larynx_M$stage <- factor(larynx_M$stage, levels = 1:4,
                         labels = c("I", "II", "III", "IV"))

# Frequency table of cancer stages
table(larynx_M$stage)
```

In R, survival data is declared using the Surv() function from the survival package. The time variable is specified first (here, *futime*), followed by the event indicator (here, *died*). The function:

```r
# 'futime' is time to event; 'died' is event indicator (1=death, 0=censored)
Surv(time = larynx_M$futime, event = larynx_M$died)
```

tells R that each value of *futime* represents either the time to event (if *died* == 1) or a censoring time (if *died* == 0). Just like *stset* in Stata, this creates a survival object that you can use in Kaplan-Meier estimation, Cox models, and other survival analysis procedures in R.

3 Exploring Stage Effects Using Kaplan-Meier Curves

Kaplan-Meier Curves by Stage (R version of sts graph, by(stage))

```r
# Create survival object
larynx_surv <- Surv(time = larynx_M$futime, event = larynx_M$died)
# Fit Kaplan-Meier curves by stage
fit_km <- survfit(larynx_surv ~ larynx_M$stage)
```

Equivalent Alternative Syntax

Use formula + data argument:

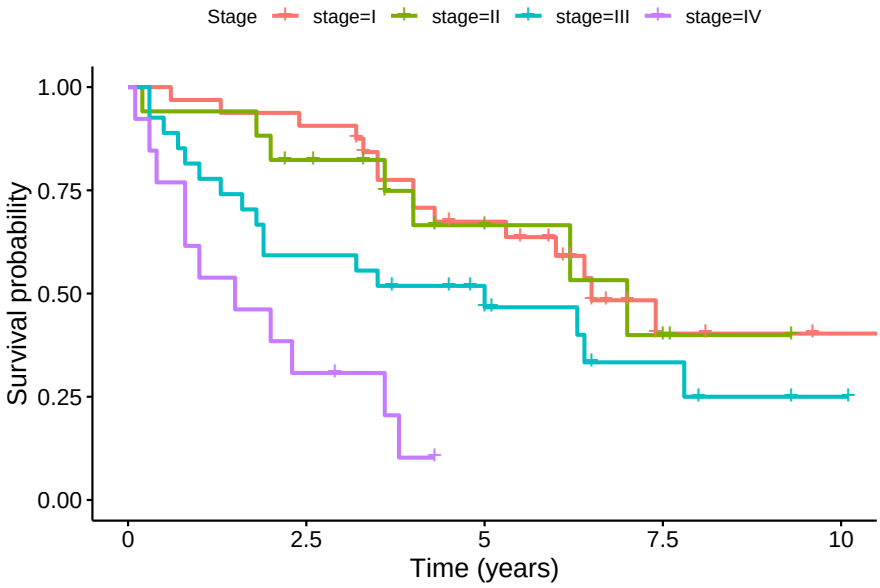
```r
fit_km <- survfit(Surv(futime, died) ~ stage, data = larynx_M)
```

Get Median survival

```r
# Print a summary of the fitted Kaplan-Meier model
# Includes number of events, median survival, etc.
fit_km
```

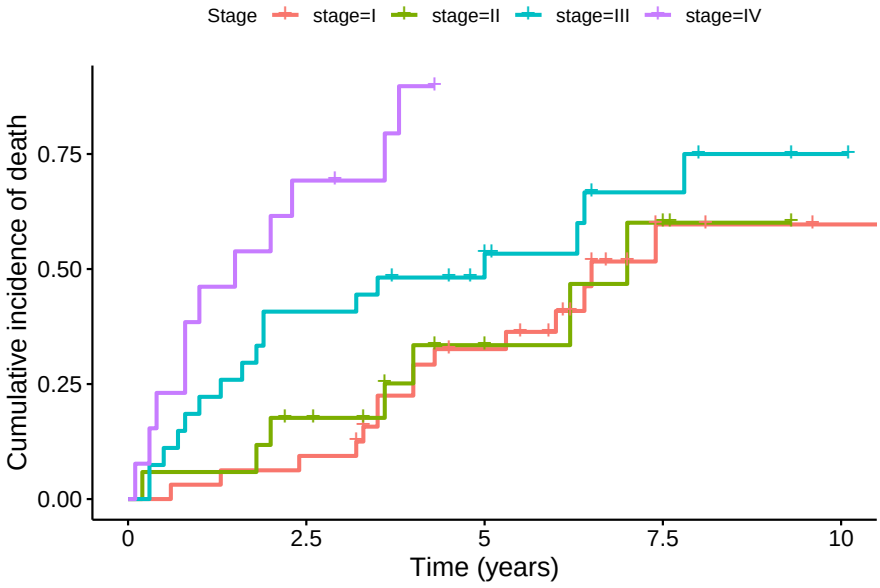
Plot KM curves by stage

```r
# Create a basic Kaplan-Meier plot
ggsurvplot(fit_km,
           data = larynx_M,
           risk.table = FALSE,
           conf.int = FALSE,
           pval = FALSE,
           xlab = "Time (years)",
           ylab = "Survival probability",
           legend.title = "Stage")
```



Equivalent to: sts graph, by(stage) failure

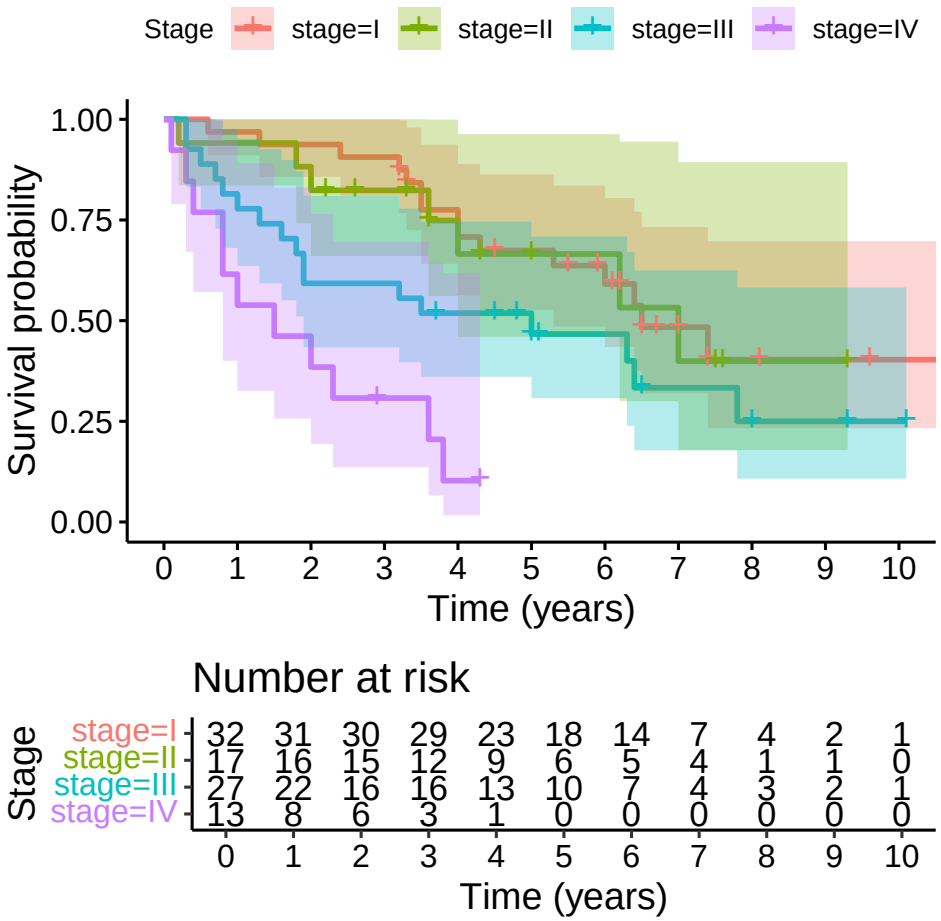
```
# Plot cumulative incidence (1 - survival)
ggsurvplot(fit_km, data = larynx_M,
           fun = "event",          # plot failure probability
           xlab = "Time (years)", ylab = "Cumulative incidence of death",
           legend.title = "Stage")
```



Combine all options: CI, censoring, and risk table.

```
# Full-featured KM plot with confidence intervals, censoring ticks, and risk table
ggsurvplot(fit_km, data = larynx_M,
           conf.int = TRUE,        # show confidence intervals
```

```
          censor = TRUE,          # show censoring marks
          break.time.by = 1,
          risk.table = TRUE,      # display number at risk table
          risk.table.y.text.col = TRUE,
          risk.table.height = 0.3,     # allocate 30% of total figure height to the risk table
          xlab = "Time (years)",
          ylab = "Survival probability",
          legend.title = "Stage")
```



The *surv.median.line* option adds visual reference lines to show median survival times on the Kaplan-Meier plot.

Question 3.1: Based on the Kaplan-Meier's what is your impression of the influence of the stages on death? Does it appear that the effect of 1 unit change in stage is the same across the range of values?

Question 3.2: Calculate and display the survival probabilities at years 1, 2, and 5 for each stage using the default method and the log-log method to match the results obtained from Stata

Use the default option

```
fit.KM.default <- survfit(Surv(futime, died) ~ stage, data = larynx_M)

fit.KM.default %>%
  tbl_survfit(times = c(1,2,5), label="**Stage**")
```

Use the log-log option to match STATA output

```
fit.KM.Stata <- survfit(Surv(futime, died) ~ stage, data = larynx_M, conf.type="log-log")

fit.KM.Stata %>%
  tbl_survfit(times = c(1,2,5), label="**Stage**")
```

Question 3.3: Obtain median survival or other percentiles

```
# Use the default option
fit.KM.default %>%
  tbl_survfit(probs = 0.5, label="**Stage**")

# Use the log-log option
fit.KM.Stata %>%
  tbl_survfit(probs = 0.5, label="**Stage**")
```

4 The Cox Model

Fit Cox proportional hazards models with various tie-handling methods and report regression coefficients along with confidence intervals

The default approach for handling ties used by the "coxph" function is Efron's method, which offers higher accuracy when there is a large number of ties. To obtain results consistent with STATA output, apply the Breslow method by specifying ties="breslow"; this method may be less accurate but is somewhat faster to compute.

Note that even after specifying ties="breslow", the results might still slightly differ from those provided by STATA, as different optimization algorithms are employed.

```
fit.cox.default <- coxph(Surv(futime, died) ~ stage, data = larynx_M)
summary(fit.cox.default)

fit.cox.default %>%
  tbl_regression(exp = TRUE)

fit.cox.Stata <- coxph(Surv(futime, died) ~ stage, data = larynx_M, ties = "breslow")
summary(fit.cox.Stata)

fit.cox.Stata %>%
  tbl_regression(exp = TRUE)
```

Question 4.1: Is stage a statistically significant predictor? Which stage is at highest risk of death? Which are second and third?

Question 4.2: Obtain the hazard ratio of stage II compared with Stage I.

Question 4.3.: Obtain the hazard ratio of stage III compared with Stage II.

> Use lincom() from *biostat3* package: You can calculate the hazard ratio between any two groups using the lincom() function from the biostat3 package. This function performs contrast tests to compare the model's coefficients directly without needing to re-fit the model.

```
## Compare Stage III to Stage II in the Cox model using lincom()
# - This computes: log(HR_stageIII) - log(HR_stageII)
# - eform = TRUE returns the hazard ratio (HR) instead of the log-HR

# library(biostat3)  # Uncomment if not already loaded
lincom(fit.cox.default, c("stageIII - stageII"), eform = TRUE)
```

Question 4.4.: Obtain the hazard ratio of stage IV compared with Stage III.

Question 4.5: Does it appear that the effect of 1 unit change in stage is the same across the range of values?

Question 4.6: Do your answers above agree with the Kaplan-Meier graphs?

Question 4.7: Implement a trend test for stage. Is there evidence of a linear trend?

The code below fits a Cox proportional hazards model using the ordered stage as a predictor to evaluate the linear relationship between the ordered stage and survival.

```
# Fit a Cox proportional hazards model treating 'stage' as an ordered variable
# - This tests for a linear trend in hazard across the stage levels
# - Assumes that higher stages are associated with increasing hazard
# - 'ordered(stage)' automatically treats the factor as numeric ranks (1 < 2 < 3 < 4)

fit.trend <- coxph(Surv(futime, died) ~ ordered(stage), data = larynx_M)

# View model summary
# - Check the coefficient and p-value for the ordered stage term
# - A significant positive coefficient suggests a linear increase in hazard with stage
summary(fit.trend)
```

5 Changing the Reference Group

Question 5.1: Fit a new model for stage using 4 as the baseline group. How does it compare to the previous model with 1 as the baseline? Is stage a stronger predictor?

Change Reference Group to Stage IV and Refit

```
# Relevel the stage factor so that Stage III is the reference level
larynx_M$stage <- relevel(larynx_M$stage, ref = "IV")

# Refit the Cox model to directly compare Stage I vs Stage IV
cox_stage_ref4 <- coxph(Surv(futime, died) ~ stage, data = larynx_M)

# View the summary to see the HR for Stage I vs Stage IV
summary(cox_stage_ref4)
```

Question 5.2: Obtain the hazard ratio of stage III compared with Stage II. How does it compare to the one from Question 4.3?

Question 5.3: Obtain the hazard ratio of stage IV compared with Stage III. How does it compare to the one from Question 4.4?

Question 5.4:
Overall, what is similar and different between the model fits with two different reference groups?

6 Continuous Predictors

Question 6.1: What is the effect of age on survival after adjusting for stage?

```
# Fit Cox proportional hazards model with age and stage as covariates
# - Age is continuous
# - Stage is categorical (as a factor)
fit.age.stage <- coxph(Surv(futime, died) ~ age + stage, data = larynx_M)

# View model summary
summary(fit.age.stage)
```

Question 6.2: Obtain the hazard ratio of a 10 years increase in age. Has the significance level of the age effect become stronger?

```
# Estimate the HR for a 10-year increase in age using lincom()
# - This is equivalent to multiplying the coefficient for age by 10
# - eform = TRUE gives the HR and 95% CI
lincom(fit.age.stage, c("10 * age"), eform = TRUE)
```

Question 6.3: Obtain the hazard ratio of a 10 years decrease in age. lincom -10*age, hr

Question 6.4: Verify that the hazard ratio and limits of its confidence interval in Question 6.3 is the reciprocal (one divided by the value) of the corresponding values in Question 6.2.