Lab 8 BIS 505b

Maria Ciarleglio

4/26/2021

Goal of Lab 8

In **Lab 8**, we will analyze a **survival** or **time-to-event endpoint**. We will begin by discussing how to **(1)** work with dates. Next, we **(2)** summarize the survival experience using the Kaplan-Meier estimate of the survival function and **(3)** compare survival curves using a log-rank test. Finally, we will **(4)** model the time-to-event outcome using a Cox Proportional hazards model and **(5)** estimate adjusted survival probabilities from the Cox PH model. The Bonus Material presents an additional option for plotting survival curves using the `survminer` package.

Analysis Data Set

In this lab, we will analyze data from the Worcester Heart Attack Study whose main goal was to describe factors associated with trends over time in survival following hospital admission for acute myocardial infarction (MI). This data set is contained in `whas.csv` and is imported as the data frame `whas` in code chunk 3 above. The **Data Key** is provided below. In this lab, our endpoint of interest is survival time (in days) following hospitalization for the acute MI (time variable, `lenfol`, to be created). Death was observed for the subjects with `fstat = 1` on the date recorded in `fdate`.

Variable Name	Definition
`id`	Subject ID
`age`	Age (years)
`sex`	Sex
	0 = Male
	1 = Female

Variable Name	Definition
`hr`	Heart rate at admission (bpm)
`bmi`	Body mass index (kg/m2)
`cvd`	History of cardiovascular disease (CVD)
	0 = No
	1 = Yes
`admitdate`	Hospital admission date
`disdate`	Hospital discharge date
`fdate`	Date of last follow-up
`fstat`	Vital status at last follow-up
	0 = Alive
	1 = Dead

- **Creating New Variables and Factor Variables:**

We begin by creating categorical variables for the following quantitative variables:

- Age (`agegrp`), <60, 60-74, ≥ 75
- Heart rate (`hrgrp`), <85 and ≥ 85
- BMI category (`bmigrp`), <25: Underweight and normal weight; ≥ 25: Overweight and obese

```
# Creating age groups
whas$agegrp[whas$age < 60] <- 1
whas$agegrp[whas$age >= 60 & whas$age < 75] <- 2
whas$agegrp[whas$age >= 75] <- 3

# Creating heart rate groups
whas$hrgrp <- ifelse(whas$hr >= 85, 1, 0)

# Creating BMI groups
whas$bmigrp <- ifelse(whas$bmi >= 25, 1, 0)
```

Next, we create factor variable versions of the **categorical variables** in the data frame (`sex` , `cvd` , `agegrp` , `hrgrp` and `bmigrp`). The **event/censoring indicator** in our survival analysis (`fstat`) does not need to be converted to a factor. As always, the **first level** specified in the `factor()` function is the **reference level** of the factor.

```
# Creating factor variables in whas using mutate() function in "dplyr" package
whas <- mutate(whas,
               sex_factor = factor(sex,
                                   levels = 0:1,
                                   labels = c("Male", "Female")),
               cvd_factor = factor(cvd,
                                   levels = 0:1,
                                   labels = c("No", "Yes")),
               agegrp_factor = factor(agegrp,
                                      levels = 1:3,
                                      labels = c("<60", "60-74", ">=75")),
               hrgrp_factor = factor(hrgrp,
                                     levels = 0:1,
                                     labels = c("<85", ">=85")),
               bmigrp_factor = factor(bmigrp,
                                      levels = 0:1,
                                      labels = c("Underweight/Normal weight",
                                                 "Overweight/Obese")))
```

Working with Dates

In **survival analysis**, the outcome of interest is **time until an event occurs**. Survival endpoints consist of **(1)** a *time component*, t_i (the time from a clearly defined start point until the event of interest or right-censoring occurs) and **(2)** an *event indicator*, δ_i, that equals `1` if the event is observed and equals `0` if the subject is right-censored.

Calculating **survival time** often requires working with **date variables**. There are three dates in the `whas` data frame, `admitdate`, `disdate`, and `fdate`. We will use these variables to calculate length of hospital stay (`los`) and length of follow-up (`lenfol`).

- Length of hospital stay (`los`) is equal to the length of time between hospital admission (`admitdate`) and discharge (`disdate`).
- Length of follow-up (`lenfol`) is equal to the length of time between hospital admission (`admitdate`) and last follow-up (`fdate`). For those who died during follow-up (`fstat = 1`), `fdate` is equal to the date of death; for those who did not die during follow-up (i.e., right censored with `fstat = 0`), `fdate` is equal to the last time the patient was known to be alive.

Notice that the three date variables are imported into **R** as *character variables*:

```
class(whas$admitdate)
```

```
## [1] "character"
```

```
class(whas$disdate)
```

```
## [1] "character"
```

```
class(whas$fdate)
```

```
## [1] "character"
```

The `lubridate` package contains several functions that make it easier to work with dates in **R**. We can convert character variables to date variables, extract the day, month, or year from a date, and calculate intervals of time between two dates on different time scales.

Date format	**`lubridate` function**
Year Month Day	`ymd()`
Day Month Year	`dmy()`
Month Day Year	`mdy()`
Extract month	`month()`
Extract day of month	`day()`
Extract year	`year()`
Days between two dates	`as.duration(startdate %--% enddate)/ddays(1)`
Months between two dates	`as.duration(startdate %--% enddate)/dmonths(1)`
Years between two dates	`as.duration(startdate %--% enddate)/dyears(1)`

All four lines of code below will convert the inputted dates (character strings) to an **R** date. To verify that an object is of class "Date", use either the `str()` function or the `class()` function.

```
ymd("2021/4/26")
```

```
## [1] "2021-04-26"
```

```
ymd("2021 Apr 26")
```

```
## [1] "2021-04-26"
```

```
ymd("21 April 26")
```

```
## [1] "2021-04-26"
```

```
datetry <- ymd("2021-04-26")
class(datetry)
```

```
## [1] "Date"
```

In the `whas` data, we will overwrite `admitdate`, `disdate`, and `fdate` with the date versions of these variables. Since all three variables are formatted the same way (day/month/year), we will use the `dmy()` function to create the date variables:

```
# Using mdy() function in "lubridate" package
whas$admitdate <- mdy(whas$admitdate)
whas$disdate <- mdy(whas$disdate)
whas$fdate <- mdy(whas$fdate)
```

Exercise: Are the three date variables now recognized as dates in **R**?

▶ Answer:

Using the date variables, we can calculate the **interval of time** between hospital admission and discharge (`los`) and the **interval of time** between hospital admission and last follow-up (`lenfol`). The length of stay will be recorded in days, while the length of follow-up will be recorded in years. The length of follow-up (time from hospital admission for acute MI to death or right censoring) is the main time variable of interest in this study.

The `as.duration(startdate %--% enddate)` function in the `lubridate` package calculates the interval of time between `startdate` and `enddate` . Dividing this value by `ddays(1)` then returns the interval of time in *days*. Specifying `dmonths(1)` or `dyears(1)` instead will return the interval of time in *months* and *years*, respectively.

```
# Length of stay (days)
whas$los <- as.duration(whas$admitdate %--% whas$disdate)/ddays(1)

# Length of follow-up (years)
whas$lenfol <- as.duration(whas$admitdate %--% whas$fdate)/dyears(1)
```

Exercise: Calculate your age in days. **Note:** The `today()` function returns today's date.

▶ Answer:

Research Questions

We are interested in determining which factors are associated with **survival after hospitalization for acute MI**. We will explore the impact of age, sex, heart rate, BMI, and history of CVD on time to death.

Estimating the Survival Function

The **survival function**, $P(T > t)$, describes the survival experience in a population over time and reports the probability of being event-free (i.e., surviving past) some specified time t. The most widely used estimator of the survival function is the **Kaplan-Meier estimator**, also known as the **product-limit estimator**, $\hat{S}(t)$.

To obtain estimates of the **Kaplan-Meier estimator** in **R**, we use the `survfit()` function in the `survival` package. The `survfit()` function requires that that we specify the *survival endpoint* (t_i, δ_i) using the `Surv()` function. The `Surv()` function produces the appropriate structure for censored survival endpoint. In the `whas` data, `lenfol` is the **survival time variable** and `fstat` is the **event indicator**.

The `survfit()` output also includes the estimated standard error of $S(t)$ and pointwise **confidence intervals** for $S(t)$. There are two commonly-reported confidence intervals for $S(t)$:

1. **Linear** or **symmetric** ("plain") **CI**, $\hat{S}(t) \pm z_{1-\frac{\alpha}{2}} \widehat{\mathrm{SE}}\left[\hat{S}(t_{(j)})\right]$ for $t_{(j)} \leq t < t_{(j+1)}$
2. **Log-log CI**, which creates a CI for $\log\left(-\log S(t)\right) = (c_L, c_U)$ that is transformed to give a CI for $S(t)$, $\left(\exp\left(-e^{c_U}\right), \exp\left(-e^{c_L}\right)\right)$

Both are valid confidence intervals. However, the log-log confidence interval is more commonly used since it will never give a CI endpoint that is outside of the range of [0, 1]. [*Remember, $S(t)$ is a probability*.]

`survfit()` **Function Arguments**	**Option Definition**
`formula=`	`Surv(time, status) ~ group_variable` (*Note:* use `~ 1` for an overall survival curve)
`data=`	Data frame containing sample data
`conf.type=`	Type of confidence interval produced: No CI (`=none`), symmetric or linear CI (`=plain`), and log-log CI (`=log`) (default)
`conf.int=`	Confidence level of 2-sided CI for the survival curve(s), `=0.95` (default)

The syntax below returns the Kaplan-Meier estimator for the full `whas` sample in the object `km` .

```
# Overall KM survival probabilities
km <- survfit(Surv(lenfol, fstat) ~ 1, data = whas)
```

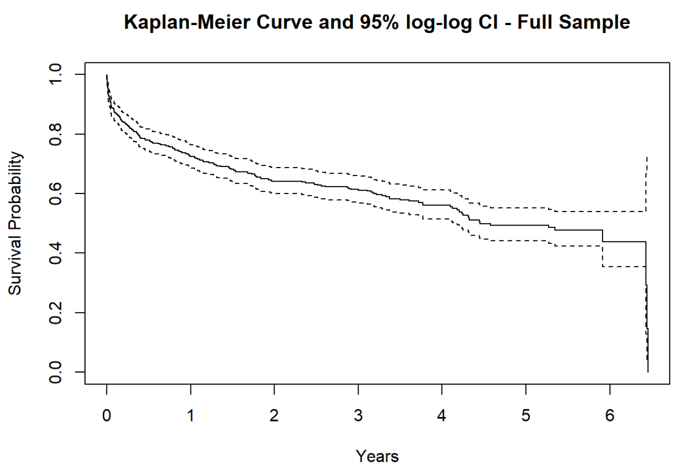
We can print the **table of Kaplan-Meier probabilities** using `summary(km)` , **plot the Kaplan-Meier survival curve(s)** using `plot(km)` , and print the **median survival time**, **25th** and **75th percentiles of survival time** using `quantile(km)` . *Note:* When printing the Kaplan-Meier probabilities using `summary(km)` , the printed tables may be long since there will be one row for each unique event time.

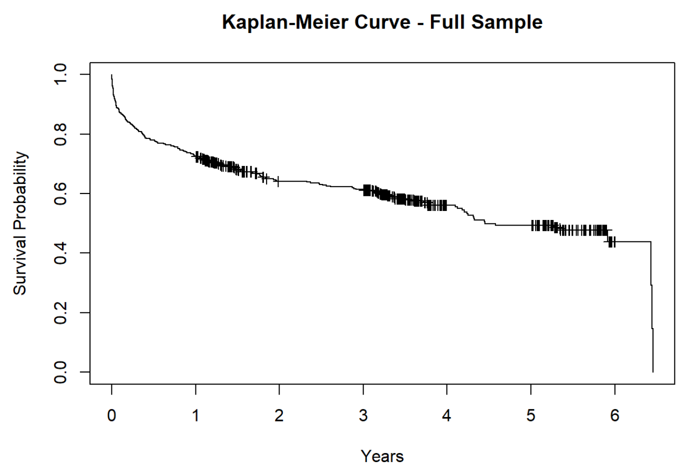
Using `survfit()` **Object**	**Output**
`summary(km)`	At each unique event time: Kaplan-Meier survival probability $\hat{S}(t)$, n at risk, n events, estimated standard error of $\hat{S}(t)$, and 95% CI for $S(t)$
`plot(km, ...)`	Plot of Kaplan-Meier curve. Some commonly used options for `plot()` below:
`conf.int=`	Display CI in plot; `=TRUE` (default for 1 curve), `=FALSE` (default for 2 curves)
`mark.time=`	Display censoring times on survival curve; `=FALSE` (default)
`col=`	Line color (default `=1` (i.e., `"black"`))
`lty=`	Line type (default `=1`)
`lwd=`	Line width (default `=1`)
`xmax=`	Maximum x-axis plot coordinate (time)

Using `survfit()` Object	Output
`quantile(km)`	25th, 50th, and 75th percentiles of survival time and 95% CIs
`quantile(km)$quantile`	25th, 50th, and 75th percentiles of survival time only

```
# KM survival probability table
# summary(km)   # Note: Many rows in this KM probability table. Not printed, here
```

```
# Basic overall KM survival curve plot
plot(km, xlab = "Years", ylab = "Survival Probability")
title("Kaplan-Meier Curve and 95% log-log CI - Full Sample")

# Suppress CIs, display censoring times on KM curve (vertical ticks)
plot(km, xlab = "Years", ylab = "Survival Probability",
     conf.int = FALSE,
     mark.time = TRUE)
title("Kaplan-Meier Curve - Full Sample")
```





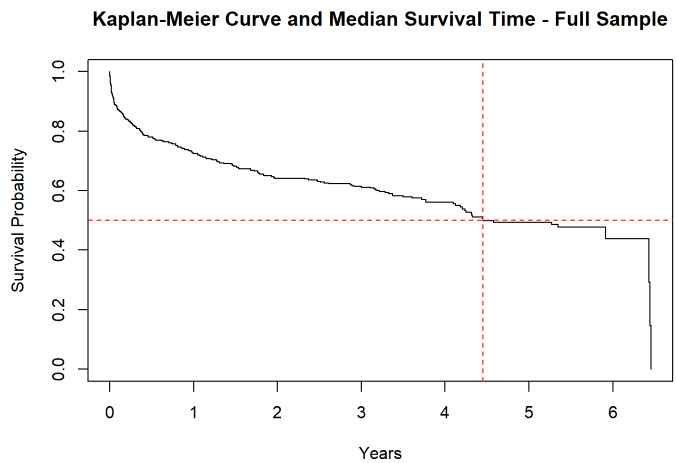
- We see that the survival curve shows a rapid decline within the first year and then displays a steady decline thereafter before ending at zero.

Percentiles of survival time, such as **median survival time** are often presented in survival analysis. The **median survival time** is estimated as the smallest survival time for which the survivor function $\hat{S}(t) \leq 0.5$. If a survival curve does not reach a probability of 0.5, then the median survival time is not calculated.

```
# 25th, median, and 75th percentiles of survival time
quantile(km)$quantile
```

```
##        25        50        75
## 0.8104038 4.4544832 6.4421629
```

```
# Overall KM survival curve plot with median survival time lines
plot(km, xlab = "Years", ylab = "Survival Probability",
     conf.int = FALSE)

abline(h = 0.5, col = "red", lty = 2)    # horizontal line at S^(t)=0.5
abline(v = quantile(km)$quantile[2], col = "red", lty = 2)  # vertical line at t^50

title("Kaplan-Meier Curve and Median Survival Time - Full Sample")
```



- In the full `whas` sample, the estimated **median survival time** $\hat{t}_{50}$ is equal to 4.5 years. That is, half of patients are expected to survive 4.5 years after hospitalization for acute MI
- The **25th percentile of survival time** tells us that 75% of individuals are expected to survive at least $\hat{t}_{25}$ = 0.8 year
- The **75th percentile of survival time** tells us that 25% of individuals are expected to survive at least $\hat{t}_{75}$ = 6.4 years.

Comparing Survival Functions

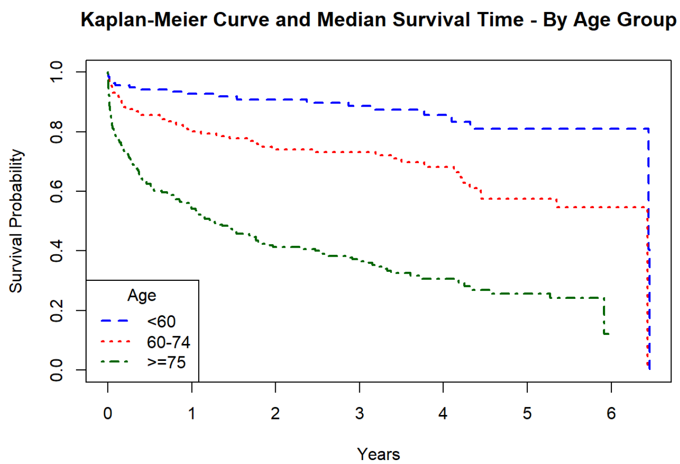
Graphically

We can compare the survival experiences in **two or more key subgroups** by plotting the estimated survival curves for each group in one plot area. To estimate the Kaplan-Meier survival probabilities within each subgroup, use the `survfit()` function and specify the grouping variable after the `~` . The syntax below returns the Kaplan-Meier estimator for each of the three age groups (`agegrp_factor`) in the object `km.age` .

```
# KM survival probabilities by age group
km.age <- survfit(Surv(lenfol, fstat) ~ agegrp_factor, data = whas)
```

Just as we did with one survival curve, we can **print the estimated KM survival probabilities for each group** using `summary(km.age)` , **plot the Kaplan-Meier curves** using `plot(km.age)` , and print the **quartiles of survival time** for each group using `quantile(km.age)` .

```
# Plot KM survival curves by age group
plot(km.age, xlab = "Years", ylab = "Survival Probability",
     col = c("blue", "red", "darkgreen"), lty = 2:4, lwd = 2)

# Useful legend positions: "bottomleft" "bottomright" "topleft" and "topright"
legend("bottomleft", title = "Age",            # legend position and title
       legend = levels(whas$agegrp_factor),    # legend group labels
       col = c("blue", "red", "darkgreen"), lty = 2:4, lwd = 2)

title("Kaplan-Meier Curve and Median Survival Time - By Age Group")
```



- The survival probabilities are highest in the youngest age group and lowest in the oldest age group.

```
# 25th, median, and 75th percentiles of survival time by age group
quantile(km.age)$quantile
```

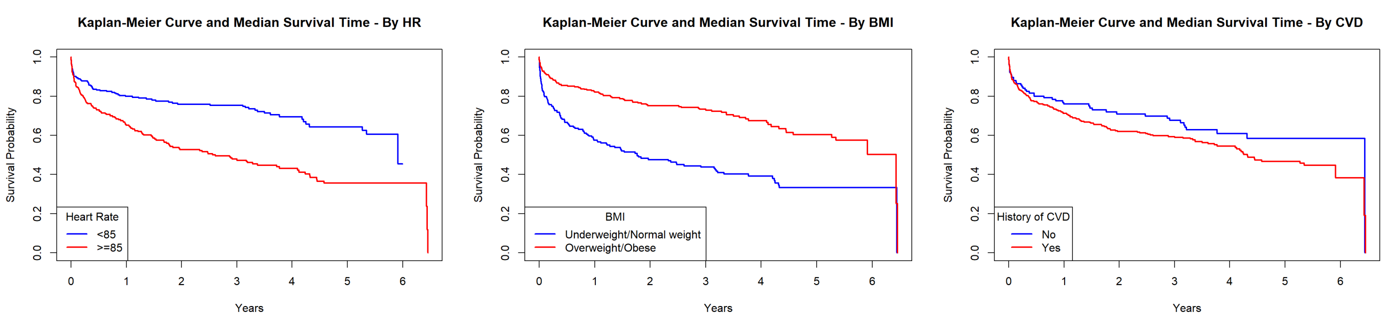
```
##                              25       50       75
## agegrp_factor=<60   6.4421629 6.442163 6.455852
## agegrp_factor=60-74 1.9548255 6.433949 6.433949
## agegrp_factor=>=75  0.1670089 1.221081 5.273101
```

- The median survival time in those 75+ years of age is only 1.22 years, while the median survival times in the 60-74 age group and the <60 age group are 6.43 and 6.44 and years, respectively.

We can similarly **graphically compare the survival experiences** in those with heart rate < 85 vs. ≥ 85, in those who are underweight or normal weight vs. those who are overweight or obese, and in those with and without history of cardiovascular disease. Notice that at the beginning of this lab, we dichotomized or categorized the quantitative variables `age`, `hr` and `bmi`. We did this so that we could generate Kaplan-Meier curves at this stage. Although we will most likely enter these variables into any future models in their original *quantitative* form, we needed to *categorize* these quantitative variables to create Kaplan-Meier curves.

```
# KM survival probabilities by heart rate group
km.hr <- survfit(Surv(lenfol, fstat) ~ hrgrp_factor, data = whas)

# KM survival probabilities by BMI group
km.bmi <- survfit(Surv(lenfol, fstat) ~ bmigrp_factor, data = whas)

# KM survival probabilities by CVD group
km.cvd <- survfit(Surv(lenfol, fstat) ~ cvd_factor, data = whas)
```

```
# Plot KM survival curves by heart rate group
plot(km.hr, xlab = "Years", ylab = "Survival Probability",
     col = c("blue", "red"), lwd = 2)
legend("bottomleft", title = "Heart Rate",
       legend = levels(whas$hrgrp_factor),
       col = c("blue", "red"), lwd = 2)
title("Kaplan-Meier Curve and Median Survival Time - By HR")

# Plot KM survival curves by BMI group
plot(km.bmi, xlab = "Years", ylab = "Survival Probability",
     col = c("blue", "red"), lwd = 2)
legend("bottomleft", title = "BMI",
       legend = levels(whas$bmigrp_factor),
       col = c("blue", "red"), lwd = 2)
title("Kaplan-Meier Curve and Median Survival Time - By BMI")

# Plot KM survival curves by CVD history
plot(km.cvd, xlab = "Years", ylab = "Survival Probability",
     col = c("blue", "red"), lwd = 2)
legend("bottomleft", title = "History of CVD",
       legend = levels(whas$cvd_factor),
       col = c("blue", "red"), lwd = 2)
title("Kaplan-Meier Curve and Median Survival Time - By CVD")
```



Based on the Kaplan-Meier plots,…

- Those with *lower heart rate* (< 85) have a better survival experience than those with higher heart rate ($\geq$ 85).
- Those who are *overweight or obese* have a better survival experience than those who are underweight or normal weight. While this seems counter-intuitive, perhaps a lower BMI is indicative of frailty in this population. Or this effect might be confounded by other unexplained patient characteristics.
- Those *without a history of CVD* have a better survival experience than those with a history of CVD.

Exercise: Plot the Kaplan-Meier survival curves for males and females (`sex_factor`). Comment on how the survival experience differs in males and females.

▶ Answer:

Log-Rank Test

The **log-rank test** is the most widely used method of comparing two survival curves and can be extended to the comparison of three or more curves. Under H_0, the distribution of survival times is identical in the g groups being compared. The log-rank statistic is based on the summed observed minus expected number of events for a given group and its variance estimate.

- H_0: $S_1(t) = S_2(t) = \ldots = S_g(t)$ for all times t vs.
- H_1: H_0 is false for some value of time t

When $g = 2$, we can state H_1 as: $S_1(t) \neq S_2(t)$ for some value of time t.

The log-rank test statistic, X^2, is compared to a **chi-square distribution** with $g - 1$ degrees of freedom. For example, when comparing two groups, the log-rank test statistic is compared to a chi-square distribution with 1 degree of freedom.

The log-rank test is carried out using the `survdiff()` function in **R**.

```
# Log-rank test comparing age groups
logrank.age <- survdiff(Surv(lenfol, fstat) ~ agegrp_factor, data = whas)
logrank.age
```

```
## Call:
## survdiff(formula = Surv(lenfol, fstat) ~ agegrp_factor, data = whas)
##
##                        N Observed Expected (O-E)^2/E (O-E)^2/V
## agegrp_factor=<60    138       20     72.0     37.52     58.04
## agegrp_factor=60-74 146       49     68.8      5.68      8.42
## agegrp_factor=>=75  216      146     74.3     69.23    108.33
##
##  Chisq= 116  on 2 degrees of freedom, p= <0.0000000000000002
```

- The log-rank test statistic, X^2 = 115.7 is compared to a chi-square distribution with 2 degrees of freedom (p-value <.001). Thus, we have evidence to reject H_0 and conclude that the survival experience is not identical in the three age groups in this population.

Note: When comparing more than 2 groups, rejection of H_0 does not indicate which groups are significantly different (similar to what we encountered in ANOVA). We can perform **post-hoc pairwise comparisons** with a **Bonferroni adjustment** for multiplicity to maintain the overall desired type I error level α. The `survminer` package contains the `pairwise_survdiff()` function that calculates pairwise log-rank comparisons between group levels with corrections for multiple testing.

```
# Bonferroni-adjusted pairwise Log-rank tests
pairs.age <- pairwise_survdiff(Surv(lenfol, fstat) ~ agegrp_factor,
                               data = whas, p.adjust.method = "bonferroni")
pairs.age
```

```
##
##  Pairwise comparisons using Log-Rank test
##
## data:  whas and agegrp_factor
##
##       <60                 60-74
## 60-74 0.00033             -
## >=75  < 0.0000000000000002 0.00000000016
##
## P value adjustment method: bonferroni
```

- The **Bonferroni-adjusted p-values** indicate that all three age groups are significantly different.
- <60 year-olds vs. 60-74 year-olds: Bonferroni-adjusted p-value <.001
- <60 year-olds vs. 75+ year-olds: Bonferroni-adjusted p-value <.001
- 60-74 year-olds vs. 75+ year-olds: Bonferroni-adjusted p-value <.001

Exercise: Is there a significant difference in the survival curves in males and females?

▶ Answer:

Cox Proportional Hazards Model

The **Cox proportional hazards (PH) model** models the hazard $h(t; x)$ as a function of the covariates x. This model describes the hazard at time t for an individual with covariates x. In the Cox model, the hazard at time t is the product of the **baseline hazard function** $h_0(t)$ and the exponentiated linear predictor $\exp(\beta\, x)$, giving $h(t; x) = h_0(t)\, \exp(\beta\, x)$. Using the (natural) log link gives the formulation of the Cox proportional hazards regression model that is linear in the coefficients:

$$\log(h(t; x)) = \log(h_0(t)) + \beta_1\, x_1 + \beta_2\, x_2 + \ldots + \beta_k\, x_k$$

An important property of the Cox model is that the baseline hazard, $h_0(t)$, is an unspecified function. That is, no assumption is made about the form or shape of the baseline hazard. This makes the Cox model a flexible model since we do not have to specify the distribution of the survival times. You will notice that the output from the Cox PH model does not estimate an intercept term.

- The estimated slope b_j is equal to the estimated **log-hazard ratio** associated with a 1-unit increase in x_j controlling for or holding all other predictors constant. We must **exponentiate** the slope to find the estimated **hazard ratio** (i.e., $\hat{HR} = e^{b_j}$). The Cox PH model assumes that the hazard ratio is constant over time (i.e., the hazard for one individual is proportional to the hazard for any other individual and that hazard ratio is independent of time).

A **hypothesis test** of the slope parameter $H_0 : \beta_j = 0$ vs. $H_1 : \beta_j \neq 0$ is performed using a **Wald test**, $z = \frac{b_j}{s_{b_j}}$, which is compared to a **standard Normal distribution**. Under H_0, $\beta_j = 0$, there is no association between x_j and the hazard of the event. When the $\log(HR) = 0$, the $HR = e^0 = 1$.

`coxph()` **Function Arguments**	**Option Definition**
`formula=`	`Surv(time, status) ~ predictor_variable1 + predictor_variable2`
`data=`	Data frame containing sample data

Simple Cox PH Model

We begin by fitting an **unadjusted Cox PH model** using age group (`agegrp_factor`) as the only predictor. We would like to determine if there is an association between age group and hazard of death after hospitalization for acute MI. Since age group <60 is the reference group, we will compare the hazard of death in 60-74 year-olds vs. those <60 and we will compare the hazard of death in those 75+ vs. those <60. *Note:* Since we have `age` recorded as a quantitative variable in the `whas` data frame, we could also create a model using quantitative `age` as the independent variable. However, let's begin by constructing a model using `agegrp_factor` as an exercise in how to interpret estimated coefficients associated with a categorical variable in the Cox PH model.

The `contrasts()` function returns the dummy variable coding that **R** uses to represent a factor variable. `agegrp_factor` is made up of two dummy variables (z_1 and z_2). z_1 equals `1` in 60-74 year-olds and z_2 equals `1` in 75+ year-olds. Individuals <60 years old (the reference category) have both z_1 and z_2 equal to `0` .

```
contrasts(whas$agegrp_factor)
```

```
##       60-74 >=75
## <60       0    0
## 60-74     1    0
## >=75      0    1
```

To describe the association between hazard of death and **age group** (`agegrp_factor`), fit the Cox PH model, $\log(h(t; x)) = \log(h_0(t)) + \beta_1\,\text{Age}_{60-74} + \beta_2\,\text{Age}_{75+}$. The output of the `coxph()` function is usually saved as an object (`cox.agegrp` , below) and the `summary()` function is applied to that object (`summary(cox.agegrp)`) to output detailed results.

```
# Cox PH model of age group
cox.agegrp <- coxph(Surv(lenfol, fstat) ~ agegrp_factor, data = whas)
summary(cox.agegrp)
```

```
## Call:
## coxph(formula = Surv(lenfol, fstat) ~ agegrp_factor, data = whas)
##
##   n= 500, number of events= 215
##
##                      coef exp(coef) se(coef)     z             Pr(>|z|)
## agegrp_factor60-74 1.0288    2.7976   0.2759 3.729             0.000192
## agegrp_factor>=75  2.0723    7.9432   0.2516 8.236 < 0.0000000000000002
##
##                    exp(coef) exp(-coef) lower .95 upper .95
## agegrp_factor60-74     2.798     0.3574     1.629     4.804
## agegrp_factor>=75      7.943     0.1259     4.851    13.007
##
## Concordance= 0.694  (se = 0.016 )
## Likelihood ratio test= 117.7  on 2 df,   p=<0.0000000000000002
## Wald test            = 91.69  on 2 df,   p=<0.0000000000000002
## Score (logrank) test = 115.7  on 2 df,   p=<0.0000000000000002
```

We can extract the **model coefficients** (b_1, b_2) using the `coef()` function and the **confidence intervals** of the model parameters (β_1, β_2) using the `confint.default()` function. Remember that we must exponentiate b_j to give an estimate of the hazard ratio. Similarly, we must exponentiate the confidence interval for β_j to give a confidence interval for the hazard ratio, e^{β_j}.

```
# Slope coefficient = logHR, exponentiated slope coefficient = HR and 95% CI for HR
round(cbind(bj=coef(cox.agegrp), HR=exp(coef(cox.agegrp)), exp(confint.default(cox.agegrp))), 5)
```

```
##                         bj      HR   2.5 %   97.5 %
## agegrp_factor60-74 1.02877 2.79762 1.62913  4.80420
## agegrp_factor>=75  2.07231 7.94318 4.85074 13.00712
```

- The **fitted model** is given by the equation, $\log(\hat{h}(t;x)) = \log(\hat{h}_0(t)) +$ 1.029 Age$_{60-74}$ $+$ 2.072 Age$_{75+}$
- The **estimated slope** of Age$_{60-74}$, $b_1 =$ 1.029 is equal to the *log-hazard ratio* of death in those *60-74* vs. those *<60* (ref). The exponentiated slope e^{b_1} gives the estimated **hazard ratio**, $\hat{HR} = e^{b_1} =$ 2.8 [95% CI (1.63, 4.8)]. In this study, 60-74 year olds had 2.8 times the hazard of death compared to those <60 years old.
- The **estimated slope** of Age$_{75+}$, $b_2 =$ 2.072 is equal to the *log-hazard ratio* of death in those *75+* vs. those *<60* (ref). The exponentiated slope e^{b_2} gives the estimated **hazard ratio**, $\hat{HR} = e^{b_2} =$ 7.94 [95% CI (4.85, 13.01)]. In this study, 75+ year olds had 7.94 times the hazard of death compared to those <60 years old.
- A **significance test of the slope** ($H_0 : \beta_1 = 0$ vs. $\beta_1 \neq 0$) reports a z-statistic z = 3.73, which is compared to a standard Normal distribution. We have evidence to reject H_0 and conclude that the hazard of death is significantly different in those 60-74 vs. those <60 (p-value <.001).
- A **significance test of the slope** ($H_0 : \beta_2 = 0$ vs. $\beta_2 \neq 0$) reports a z-statistic z = 8.24, which is compared to a standard Normal distribution. We have evidence to reject H_0 and conclude that the hazard of death is significantly different in those 75+ vs. those <60 (p-value <.001).

A **Likelihood Ratio Test** can be used to simultaneously test the significance of a group or set of parameters when fitting a Cox PH regression model. For example, to test the overall significance of our 3-level **age group** variable, we would test: $H_0 : \beta_1 = \beta_2 = 0$ vs. $H_1 : \beta_1, \beta_2$ not both 0. Here, we are comparing two **nested models**,

- **Full model**: $\log(h(t;x)) = \log(h_0(t)) + \beta_1 \text{ Age}_{60-74} + \beta_2 \text{ Age}_{75+}$
- **Reduced model** (i.e., model under H_0, without `agegrp_factor`): $\log(h(t;x)) = \log(h_0(t))$

The **likelihood ratio test statistic** compares the likelihood of the full and reduced models, $\text{G} = -2\log\text{-likelihood}(R) - (-2\log\text{-likelihood}(F))$. The test statistic is compared to an Chi-square distribution with *degrees of freedom* equal to the number of parameters tested under H_0, χ^2_{df}.

The `Anova()` function in the `car` package applied to a model object (e.g., `cox.agegrp`) returns individual likelihood ratio tests for each variable in the model. In the case of `cox.agegrp` , `agegrp_factor` is the only variable in the model; however, this technique can also be used in multiple Cox PH models.

```
# LRT using Anova() function in the "car" package
Anova(cox.agegrp)
```

```
## Analysis of Deviance Table
##  Cox model: response is Surv(lenfol, fstat)
## Terms added sequentially (first to last)
##
##                 loglik  Chisq Df            Pr(>|Chi|)
## NULL           -1227.3
## agegrp_factor -1168.5 117.67  2 < 0.00000000000000022
```

- Based on the output above, the **likelihood ratio test** of `agegrp_factor` ($H_0 : \beta_1 = \beta_2 = 0$ vs. $H_1 : \beta_1$, β_2 not both 0) has a test statistic G = 117.7, which is compared to an Chi-square distribution with 2 degrees of freedom. The overall effect of age group is statistically significant in this Cox PH model (p-value <.001). We have evidence to reject H_0 and conclude that at least one of β_1 or β_2 is not equal to 0.

In the Kaplan-Meier survival curves, we saw that survival experience was best in younger patients and worst in older patients. The estimated hazard ratios from the Cox PH model also showed this same association. Next, let's fit a Cox PH model using **quantitative** `age` . We should expect to see an estimated hazard ratio >1, indicating that older age is associated with a greater hazard of death.

```
# Cox PH model of age (quantitative)
cox.age <- coxph(Surv(lenfol, fstat) ~ age, data = whas)
summary(cox.age)
```

```
## Call:
## coxph(formula = Surv(lenfol, fstat) ~ age, data = whas)
##
##   n= 500, number of events= 215
##
##         coef exp(coef) se(coef)     z            Pr(>|z|)
## age 0.066339  1.068589 0.006079 10.91 <0.0000000000000002
##
##     exp(coef) exp(-coef) lower .95 upper .95
## age     1.069     0.9358     1.056     1.081
##
## Concordance= 0.731  (se = 0.018 )
## Likelihood ratio test= 142.1  on 1 df,   p=<0.0000000000000002
## Wald test            = 119.1  on 1 df,   p=<0.0000000000000002
## Score (logrank) test = 126.6  on 1 df,   p=<0.0000000000000002
```

- The **fitted model** is given by the equation, $\log(\hat{h}(t;x)) = \log(\hat{h}_0(t)) +$ 0.066 Age
- The **estimated slope** of Age, $b_1 =$ 0.066 is equal to the *log-hazard ratio* of death associated with a 1-year increase in age. A 1-year increase in age increases the hazard of death by 7%; $\hat{HR} = e^{b_1} =$ 1.069 [95% CI (1.056, 1.081)].
- A **significance test of the slope** ($H_0 : \beta_1 = 0$ vs. $\beta_1 \neq 0$) reports a z-statistic z = 10.91, which is compared to a standard Normal distribution. We have evidence to reject H_0 and conclude that the hazard of death is significantly associated with age of the patient at time of hospitalization (p-value <.001).

Exercise: Based on a fitted Cox PH model, is there a significant difference in the hazard of death in those with and without a history of CVD?

▶ Answer:

Multiple Cox PH Model

We can extend the Cox PH model to include additional predictors. Below, we consider a model that contains `age`, `sex_factor`, `hr` and `bmi`.

```
# Multiple Cox PH model 1
cox.mult1 <- coxph(Surv(lenfol, fstat) ~ age + sex_factor + hr + bmi, data = whas)
summary(cox.mult1)
```

```
## Call:
## coxph(formula = Surv(lenfol, fstat) ~ age + sex_factor + hr +
##     bmi, data = whas)
##
##   n= 500, number of events= 215
##
##                        coef exp(coef)  se(coef)      z             Pr(>|z|)
## age                0.059826  1.061652  0.006628  9.026 < 0.0000000000000002
## sex_factorFemale -0.149060  0.861517  0.141593 -1.053              0.29246
## hr                 0.012277  1.012353  0.002751  4.464           0.00000806
## bmi               -0.043035  0.957878  0.015635 -2.753              0.00591
##
##                  exp(coef) exp(-coef) lower .95 upper .95
## age                 1.0617     0.9419    1.0479    1.0755
## sex_factorFemale    0.8615     1.1607    0.6527    1.1371
## hr                  1.0124     0.9878    1.0069    1.0178
## bmi                 0.9579     1.0440    0.9290    0.9877
##
## Concordance= 0.751  (se = 0.017 )
## Likelihood ratio test= 168.8  on 4 df,   p=<0.0000000000000002
## Wald test            = 142.8  on 4 df,   p=<0.0000000000000002
## Score (logrank) test = 156  on 4 df,   p=<0.0000000000000002
```

- The effect of sex is not statistically significant in a model that contains age, heart rate, and bmi (p-value = 0.292). Thus, we will remove this predictor from the model and re-run the regression model.

```
# Multiple Cox PH model 2
cox.mult2 <- coxph(Surv(lenfol, fstat) ~ age + hr + bmi, data = whas)
summary(cox.mult2)
```

```
## Call:
## coxph(formula = Surv(lenfol, fstat) ~ age + hr + bmi, data = whas)
##
##   n= 500, number of events= 215
##
##          coef exp(coef)  se(coef)      z             Pr(>|z|)
## age  0.058633  1.060386  0.006544  8.960 < 0.0000000000000002
## hr   0.012083  1.012156  0.002766  4.368            0.0000125
## bmi -0.041684  0.959173  0.015437 -2.700              0.00693
##
##     exp(coef) exp(-coef) lower .95 upper .95
## age    1.0604     0.9431    1.0469    1.0741
## hr     1.0122     0.9880    1.0067    1.0177
## bmi    0.9592     1.0426    0.9306    0.9886
##
## Concordance= 0.749  (se = 0.017 )
## Likelihood ratio test= 167.7  on 3 df,   p=<0.0000000000000002
## Wald test            = 141.2  on 3 df,   p=<0.0000000000000002
## Score (logrank) test = 154.6  on 3 df,   p=<0.0000000000000002
```

- The **fitted model** is given by the equation, $\log(\hat{h}(t;x)) = \log(\hat{h}_0(t)) +$ 0.059 Age $+$ 0.012 HR $-$ 0.042 BMI

- Wald tests of the individual slopes show that there is evidence to reject $H_0 : \beta_j = 0$ for all parameters.

Controlling for all of the other variables in the model…

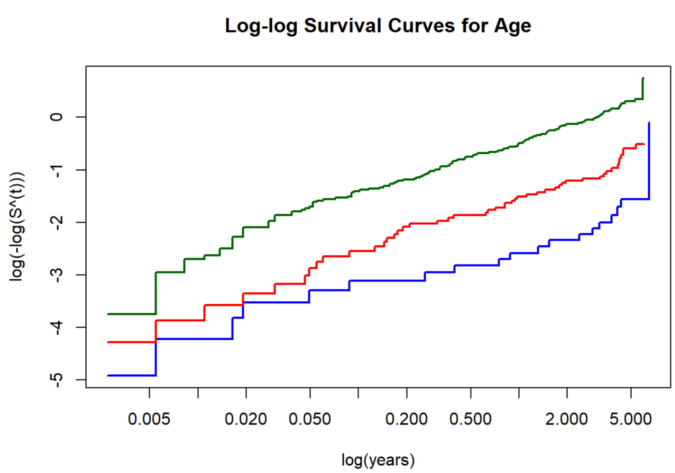
- As **age** increases, the hazard of death following hospitalization for acute MI increases. A 1-year increase in age increases the hazard of death by 6%; adjusted $\hat{HR} = e^{b_1} =$ 1.06 [95% CI (1.047, 1.074)].
- As **heart rate** increases, the hazard of death following hospitalization for acute MI increases. A 1-BPM increase in heart rate increases the hazard of death by 1%, adjusted $\hat{HR} = e^{b_2} =$ 1.012 [95% CI (1.007, 1.018)].
- As **BMI** increases, the hazard of death following hospitalization for acute MI decreases. A 1-unit increase in BMI decreases the hazard of death by 4%, adjusted $\hat{HR} = e^{b_3} =$ 0.959 [95% CI (0.931, 0.989)].

Checking the Cox PH Assumption

An important assumption of the Cox PH model is that the **hazards are proportional over time** (i.e., our hazard ratios are not a function of time). If the proportional hazards assumption holds, then the log cumulative hazard curves (commonly known as **log-log survival curves**) over levels of a covariate plotted against $\log(t)$ will be **parallel**.

In practice, the estimated Kaplan-Meier survival curves $\hat{S}(t)$ for levels of a categorical (or categorized) covariate are transformed to give $\log(-\log(\hat{S}(t)))$ and the curves are plotted vs. $\log(t)$. The `plot()` function has a built-in option for requesting this plot, so we do not have to manually perform the log-log transformation. For example, using the `survfit()` object that we created when estimating the Kaplan-Meier survival probabilities for the three age group categories (`km.age`), we can create a plot of the log-log survival curves vs. the log of time for the three age groups using the `fun = "cloglog"` option in the `plot()` function. Crossing curves or extreme lack of parallelism suggests that the proportional hazards assumption may not be valid for that variable.

```
# Assessing PH assumption for age
plot(km.age, fun = "cloglog",
     xlab = "log(years)", ylab = "log(-log(S^(t)))",
     col = c("blue", "red", "darkgreen"), lwd = 2)
title("Log-log Survival Curves for Age")
```



- The log-log survival curves appear fairly parallel and do not suggest a violation of the proportional hazards assumption for the age variable. This means that we can include and interpret the effect of age in the Cox proportional hazards model.

Cox Adjusted Survival Curves

When a Cox PH model is used to fit survival data, we can plot **adjusted survival curves**, $\hat{S}(t; x)$, that adjust for explanatory variables used as predictors in the model. The estimated baseline survival function $\hat{S}_0(t)$ is estimated by **R**.

$$\hat{S}(t; x) = \left[\hat{S}_0(t)\right]^{\exp(b_1\, x_1 + b_2\, x_2 + \ldots + b_k\, x_k)}$$

The adjusted survival curves are estimated at specific covariate values. For *categorical predictors*, the choice of values of x_j is clear, however, for *quantitative predictors*, the choice can be arbitrary. Often, the value of a *quantitative predictor* is set equal to the **overall sample mean of the variable**. For example, our model `cox.mult2` contains three quantitative predictors, `age`, `hr` and `bmi`.

```
meanage <- mean(whas$age, na.rm = TRUE)
meanhr <- mean(whas$hr, na.rm = TRUE)
meanbmi <- mean(whas$bmi, na.rm = TRUE)

c(ageval = meanage, hrval = meanhr, bmival = meanbmi)
```

```
##   ageval    hrval   bmival 
## 69.84600 87.01800 26.61378
```

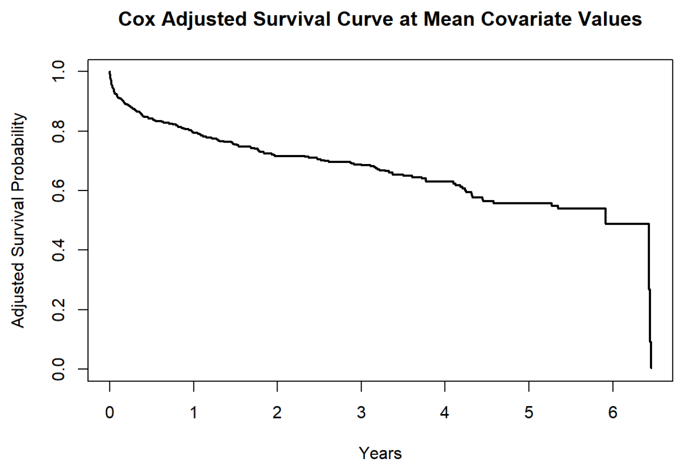
We'll estimate the adjusted survival curve for an individual who is 69.85 years old, with a heart rate of 87.02 BPM, and a BMI of 26.61. As in previous models, we must specify the values of x used in the prediction (data frame `pred.x2`, below). This data frame is then used in the `newdata=` argument of the `survfit()` function to output the adjusted survival probabilities.

```
# Adjusted S(t) estimated at mean values of age, hr and bmi
pred.x2 <- data.frame(age = mean(whas$age, na.rm = TRUE),
                      hr = mean(whas$hr, na.rm = TRUE),
                      bmi = mean(whas$bmi, na.rm = TRUE))

# Adjusted survival probabilities
Shat2 <- survfit(cox.mult2, newdata = pred.x2, data = whas)

# Adjusted median survival time
quantile(Shat2)$quantile
```

```
##       25       50       75 
## 1.538672 5.913758 6.442163
```

- The adjusted median survival time for this individual is 5.91 years.

```
# Plot of adjusted survival curve at fixed values of x
plot(Shat2, xlab = "Years", ylab = "Adjusted Survival Probability",
     conf.int = FALSE, lwd = 2)
title("Cox Adjusted Survival Curve at Mean Covariate Values")
```



- The adjusted survival curve looks similar to the overall Kaplan-Meier survival function. However, an advantage of the adjusted survival function is that we can predict the survival probabilities assuming different covariate values.

Suppose our Cox PH model contained a categorical covariate (e.g., `agegrp_factor`). The following syntax produces adjusted survival curves for those <60, 60-74, and 75+ when heart rate equals 80 and 100 and BMI equals its mean value in the sample. The `expand.grid()` function is useful for creating a data frame from all combinations of input vectors (i.e., all combinations of the levels of `agegrp_factor` and heart rates 80 and 100):

```
# Multiple Cox PH model 3
cox.mult3 <- coxph(Surv(lenfol, fstat) ~ agegrp_factor + hr + bmi, data = whas)

# Adjusted S(t) estimated at levels of agegrp_factor, at hr of 80 and 100 and at mean bmi
pred.x3 <- data.frame(expand.grid(agegrp_factor = levels(whas$agegrp_factor),
                                  hr = c(80, 100)),
                      bmi = mean(whas$bmi, na.rm = TRUE))
pred.x3
```

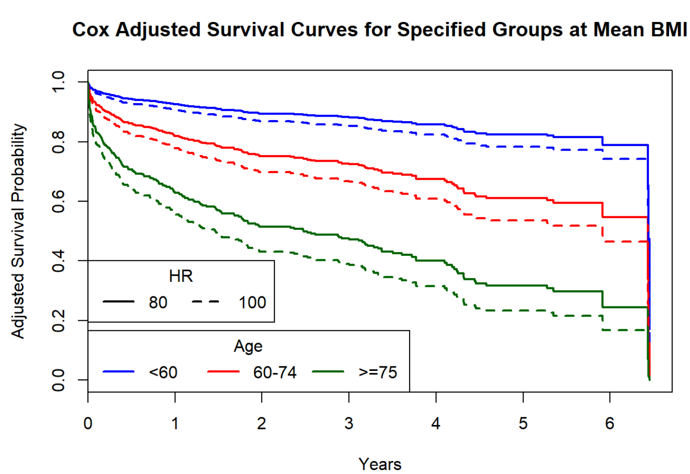
```
##   agegrp_factor  hr      bmi
## 1           <60  80 26.61378
## 2         60-74  80 26.61378
## 3          >=75  80 26.61378
## 4           <60 100 26.61378
## 5         60-74 100 26.61378
## 6          >=75 100 26.61378
```

```
# Adjusted survival probabilities
Shat3 <- survfit(cox.mult3, newdata = pred.x3, data = whas)

# Adjusted median survival times
cbind(pred.x3, quantile(Shat3)$quantile)
```

```
##   agegrp_factor  hr       bmi        25        50        75
## 1           <60  80 26.61378 6.4339493 6.442163 6.455852
## 2         60-74  80 26.61378 2.3244353 6.433949 6.442163
## 3          >=75  80 26.61378 0.3203285 2.477755 5.913758
## 4           <60 100 26.61378 5.9137577 6.442163 6.455852
## 5         60-74 100 26.61378 1.3114305 5.913758 6.442163
## 6          >=75 100 26.61378 0.1752225 1.464750 4.446270
```

- The adjusted median survival times for the three age groups, <60, 60-74, and 75+ when heart rate is fixed at 80 BPM and BMI is fixed at 26.61 are equal to 6.44, 6.43, and 2.48 years, respectively. When heart rate is assumed to equal 100 BPM and BMI is fixed at its mean value, the adjusted median survival times for the three age groups are equal to 6.44, 5.91, and 1.46 years, respectively.

```
# Plot of adjusted survival curve at fixed values of x
plot(Shat3, xlab = "Years", ylab = "Adjusted Survival Probability",
     col = rep(c("blue", "red", "darkgreen"), 2), lwd = 2,
     lty = c(rep(1,3), rep(2,3)),
     xaxs= "S")   # option to remove buffer space between y-axis and t=0

legend("bottomleft", title = "Age",
       legend = levels(whas$agegrp_factor),
       col = c("blue", "red", "darkgreen"), lwd = 2,
       horiz = TRUE)

legend(x = 0, y = 0.4,   # top-left coordinate of legend box
       title = "HR",
       legend = c("80", "100"),
       lty = c(1, 2), lwd = 2,
       horiz = TRUE)

title("Cox Adjusted Survival Curves for Specified Groups at Mean BMI")
```

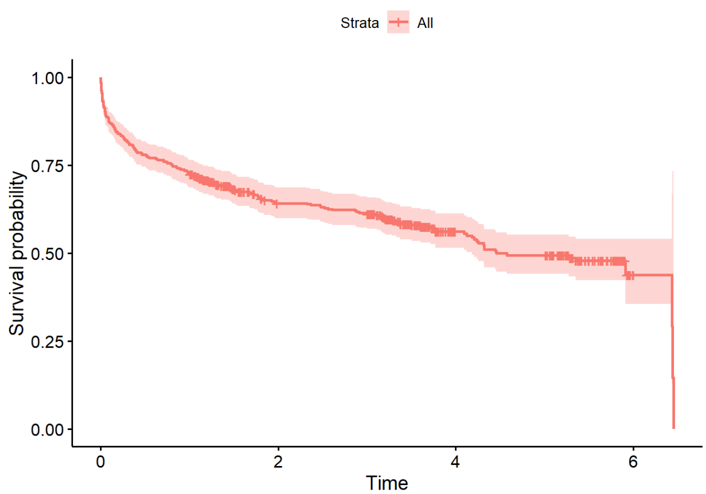


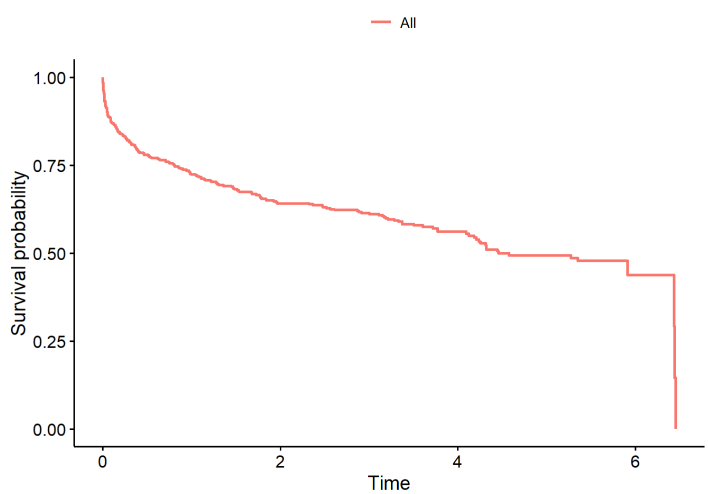
- The adjusted survival curves look similar to our Kaplan-Meier curves plotted earlier. We do see the same trend observed in the estimated hazard ratios. That is, as age increases, the survival probabilities decrease, and as heart rate increases, survival probabilities decrease.

Bonus Material: Survival Plotting with `survminer` Package

The `survminer` package contains additional functions that produce beautiful, high-quality survival analysis visualizations. The `ggsurvplot()` function plots Kaplan-Meier survival curves using `survfit()` objects. The overall Kaplan-Meier survival curve for the full sample is below. I noticed that the default Kaplan-Meier plot produced did not show the KM curve going to zero (which it does since the largest “time” is an event/death). To remedy this, we can specify the x-axis limits using the `xlim=` argument to be sure that the full range of follow-up is displayed in the figure.

```
# Overall KM survival probabilities
km <- survfit(Surv(lenfol, fstat) ~ 1, data = whas)

# "Basic" overall KM survival curve plot
ggsurvplot(km, data = whas, xlim = c(0, max(whas$lenfol)))

# Suppress CI, censor ticks, legend title
ggsurvplot(km, data = whas, xlim = c(0, max(whas$lenfol)),
           conf.int = FALSE,
           censor.shape = "",
           legend.title = "")
```





The Kaplan-Meier survival curves by sex are shown below, with *some* additional options that are available in `ggsurvplot()`.

```
# KM survival curves by sex
km.sex <- survfit(Surv(lenfol, fstat) ~ sex_factor, data = whas)

# Plot KM survival curves by sex
ggsurvplot(km.sex, data = whas,
           xlim = c(0, max(whas$lenfol)), # x-axis range
           size = 1,                      # line width
           censor.shape = "",             # suppress censor ticks
           palette = c("blue", "red"), # colors
           conf.int = TRUE,               # display CIs for S(t)
           risk.table = TRUE,             # show risk table (number at risk over time)
           risk.table.col = "strata",     # risk table color by groups
           legend.labs = levels(whas$sex_factor), # change legend labels
           legend.title = "",             # suppress legend title
           pval = TRUE,                   # show log-rank p-value
           xlab = "Time in years",        # x-axis label
           break.time.by = 1,             # x-axis time intervals (1-year)
           ggtheme = theme_bw(),          # customize plot and risk table with a ggplot() theme
           risk.table.y.text.col = TRUE)  # color risk table text annotations
```