Survival Analysis
Week 10

PH 700A, Spring 2025

Rick Calvo

Table of contents

1 Week 10 Survival Analysis

1.1 Session Overview

- Packages
- Survival & Longitudinal Analysis
  - Background
  - Purpose
  - Data Requirements
  - Types
- Kaplan-Meier Curves
- Cox Proportional Hazards Models

1.2 Packages

`descr` - Quick crosstabs

`survival` - Part of base installation. Primary package for survival analysis

`survminer` - Supplemental package for survival curve plotting

1.3 Basic Commands

From `descr`:

- `CrossTable(x,y,data)`

From `survival`:

- `Surv(time,event)`
- `survfit(formula, data)`
- `coxph()`
- `cox.zph()`

From `survminer`

- `ggsurvplot()`
- `ggadjustedcurves()`
- `ggcoxzph()`

2 Survival and Longitudinal Data Analysis

2.1 Background

Useful when study participants are *surveilled* over time.

- Time-to-event analysis for prospective designs
- Major strength in accounting for timing effects across observations
- Allows for analysis of events that are singular or recurrent
- Can address changes in risk factor status over time

2.2 Recall from Logistic Regression

Assumptions:

- Participants can only experience one event category
- Participants have approximately the same amount of *follow-up time*
- All variables must be (approximately) normally distributed
- Risk factor and outcome relationships are monotonic

What can you do if these assumptions cannot be satisfied?

2.3 Common Survival Methods

Name	Use Case
Kaplan-Meier Estimation	Bivariate analysis of singular events
Cox Proportional Hazards	Multivariable analysis of singular events
Fine & Gray Competing Risks	Primary vs alternative events over time
Shared Frailty	Clustered data
Andersen-Gill Proportional Intensity	Recurrent events
Generalized Estimating Equations	Longitudinal w/ correlated events

…among others…

2.4 Strengths of Survival Analysis

- Addresses differences in participants contributing different amounts of time to the study
- Follow-up time offers a "window" to observe an event
- Addresses end-of-follow-up "non-events" through censoring
- No assumptions made on status at end of follow-up; only that they did or did not see the event
- Expands upon *incidence density* (instead of the *incidence rate*)

2.5 Study Schema

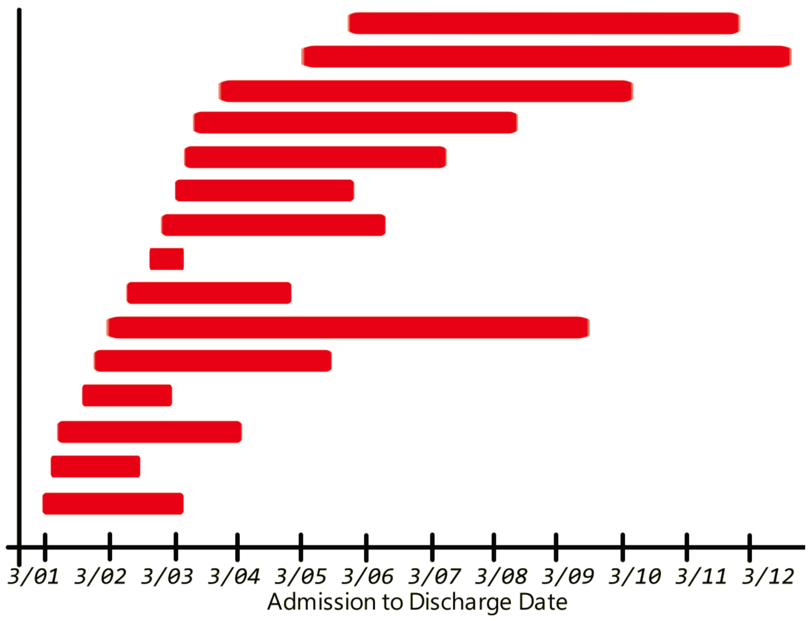


Figure 1: An example of timing differences affecting a sample.

2.6 Survival Function

In binary logistic regression, the *logistic function* was simplified as $1 = p + q$, where p is the probability of the event, and q is the probability of no event.

The *survival function* is represented as $S(t) = 1 - F(t)$

Where:

- $S(t)$ is the probability of **s**urvival (i.e. not experiencing the event) at a specific time t
- $F(t)$ is the probability of **f**ailure (i.e. experiencing the event) at a specific time t

2.7 Framework

Start of Study

	Event	Neg.
Exposed	0	500
Unexposed	0	500

Remaining = 1000

Exited = 0

$S(t = 0) = 1$

Middle of Study (T = 15 days)

	Event	Neg.
Exposed	70	330
Unexposed	50	380

Remaining = 830

Exited Non-events = 170

$S(t = 15) = 0.86$

2.8

Last Day of Study (T = 29)

	Event	Neg.
Exposed	190	100
Unexposed	110	150

Remaining = 550

Exited Non-events = 450

$S(t = 29) = 0.45$

End of Study (T = 30)

	Event	Neg.
Exposed	190	0
Unexposed	110	0

Remaining = 300

Exited Non-events = 700

$S(t = 30) = 0$

We have no true knowledge of non-events. We only know that we didn't see an event while we were looking.

2.9 Assumptions

- All participants start at $t = 0$ with $S(t = 0) = 1$
- $S(t)$ can only go down as t goes up, starting at 100%.
- All variables are *Independent and Identically Distributed* (IID)
- Variables should have the same probability distribution and no intercorrelation
- Missing values are *Missing Completely At Random*
- The probability of censoring is not related to the event
- Nobody "survives" to infinity time $S(t = \infty) = 0$

2.10 Data Requirements

- A binary `outcome` coded `0` vs. `1`
- `1` refers to your event of interest
- `0` would refer to censored non-events
- Covariates of interest
- At least one categorical independent variable
- Timing of the event

patientid	age	sex	var1	cat	time	event	...
A_12345	25	M	243	A	7	1	...
B_29828	25	M	125	A	2	0	...
C_22244	49	F	284	B	10	0	...
D_55457	49	F	96	B	11	1	...
E_00056	18	F	101	A	6	0	...
F_23492	18	F	192	B	6	1	...
G_25221	62	M	204	B	3	0	...
H_51100	62	M	222	A	4	1	...

`time` is calculated from a start to an end (i.e. admission to discharge)

3 Kaplan-Meier Estimation

3.1 Background

- The most basic type of survival analysis
- Graphical evaluation of times-to-event
- Assesses differences in event rates by a single categorical independent variable
- Lots of data checks and preliminary diagnostics need to be performed before analysis

3.2 Primary Commands

To get started, R requires you to set up a survival object containing just time to event and event status.

```r
library(survival)

km0 <- Surv(time = timingvar, event = eventVar)

km.fit0 <- survfit(km0 ~ 1, data = df)
```

`Surv(timingvar, eventVar)` creates the survival object called, `km0`

`survfit(km0 ~ 1, data = df)` fits the survival object against anything to the right of the `~`

With no variables and only the constant `1` on the right side of the formula, the total sample survival will be fit.

3.3 Evaluating Independent Variables

You can include pre-set objects or specify the entire equation manually.

```r
km.fit1 <- survfit(formula = km0 ~ var1, data = df)

km.fit1 <- survfit(formula = Surv(timingvar, eventVar) ~ var1, data = df)
```

Here, `var1` is evaluated against the survival object.

`var1` *should* be categorical for reasonable graphics generation.

3.4 Visualization Commands

`ggsurvplot` draws the overall survival curve from the fitted `survfit`

```r
km.plot1 <- ggsurvplot(km.fit1, data = df,
                       risk.table = TRUE,
                       risk.table.y.text = FALSE,
                       risk.table.y.text.col = TRUE,
                       pval = TRUE,
                       conf.int = TRUE,
                       xlim = c(LBOUND, UBOUND),
                       break.time.by = NUMBER,
                       ggtheme = theme_minimal()
            )
```

`km.plot1` stores the survival curves by `var1` in the `km.fit1` object.

`risk.table = TRUE`, `risk.table.y.text = FALSE`, and `risk.table.y.text.col = TRUE` are options to show and modify aspects of the *lifetable* under the graph.

`pval = TRUE` and `conf.int = TRUE` are options to add the p-value and 95% confidence intervals to the graph.

`xlim = c(LBOUND, UBOUND)` allows you to specify the starting and ending times shown on the X-axis. Change `LBOUND` and `UBOUND` based on the overall plot range.

`break.time.by = NUMBER` allows you to specify the tick-mark intervals on the x-axis. Change `NUMBER` to the interval of time you want to use.

`ggtheme = theme_minimal()` is a `ggplot` function allowing you to set themes. The minimal theme is being specified here but there are many others.

The y-axis can be rescaled manually for better view of the curves.

```r
km.plot1$plot <- km.plot1$plot + ylim(c(0.6,1.0))
```

The `0.6` and `1.0` refer to the percent of remaining non-events on the y-axis.

3.5 Statistical Test for Difference in Survival

```r
library(survival)

survdiff(Surv(timingvar, event)) ~ var1, rho = 0)

survdiff(Surv(timingvar, event)) ~ var1, rho = 1)
```

`rho = 0` performs the *log-rank test* by `var1`.

`rho = 1` performs the *Peto & Peto test* by `var1`.

The Peto test gives more weight to earlier events; this is when the remaining sample should be largest. It is useful when the event rate is low or with small samples.

The log-rank test gives equal weight to all events. This is the default method.

3.6 Example

```r
library(descr)

CrossTable(df$ambulance, df$admitted, prop.chisq = FALSE, chisq = TRUE)
```

```
   Cell Contents
|-------------------------|
|                       N |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

=====================================
                df$admitted
df$ambulance        0       1   Total
-------------------------------------
0                  37      52      89
                0.416   0.584   0.401
                0.514   0.347        
                0.167   0.234        
-------------------------------------
1                  35      98     133
                0.263   0.737   0.599
                0.486   0.653        
                0.158   0.441        
-------------------------------------
Total              72     150     222
                0.324   0.676        
=====================================

Statistics for All Table Factors

Pearson's Chi-squared test 
------------------------------------------------------------
Chi^2 = 5.663998      d.f. = 1      p = 0.0173 

Pearson's Chi-squared test with Yates' continuity correction 
------------------------------------------------------------
Chi^2 = 4.989155      d.f. = 1      p = 0.0255 
```

Crosstab shows a statsig relationship between arrival via ambulance and getting admitted.

3.7 Assessing Follow-up Time

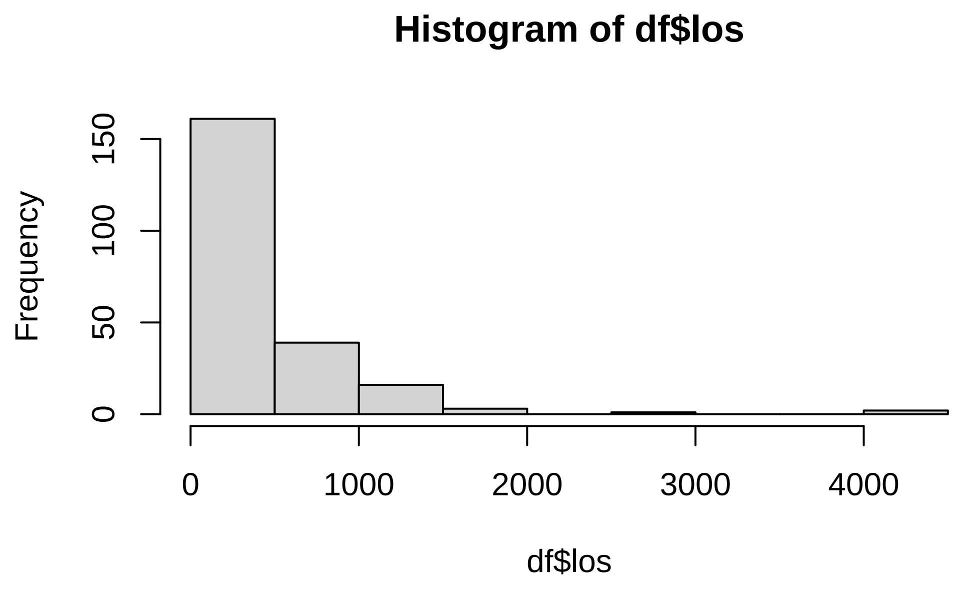
```
library(survival)
library(explore)

df %>% describe(admitted)
```

```
variable = admitted
type     = factor
na       = 0 of 222 (0%)
unique   = 2
 0       = 72 (32.4%)
 1       = 150 (67.6%)
```

```
df$los <- as.numeric(df$los)

df %>% describe(los)
```

```
variable = los
type     = double
na       = 0 of 222 (0%)
unique   = 203
min|max  = 4 | 4 460
q05|q95  = 114.55 | 1 237.05
q25|q75  = 253.25 | 526.0875
median   = 350.5
mean     = 485.7803
```

```
hist(df$los)
```



3.8 Creating Survival Object

```r
los_admit <- Surv(time = df$los, event = as.numeric(df$admitted))

head(los_admit, 100)
```

```
  [1]  263.78333   345.00000   418.00000+  219.00000   103.36667   188.00000
  [7]    4.00000+  108.00000+  244.00000+  466.00000   114.00000+  244.00000
 [13]  350.00000+ 1376.00000+  319.00000+  205.00000+  167.93333    97.00000
 [19]  323.00000   330.00000   607.00000+  268.00000   273.00000   187.00000
 [25]  177.00000   341.00000   474.00000   520.00000   460.83333   454.00000+
 [31]  696.00000   412.00000   108.18333   586.00000  1566.00000   353.00000+
 [37]  426.00000+  333.00000   253.00000   486.00000   810.00000+  337.00000
 [43]  286.00000   333.00000   156.00000+  398.00000+  416.00000  1240.00000+
 [49]  346.00000   864.00000   133.68333   337.00000   254.00000   337.66667
 [55]  203.00000+  317.00000   539.00000+  338.00000+  300.00000+  405.00000
 [61]  648.00000   580.00000+  299.00000+  517.00000   293.00000   180.25000
 [67]  153.60000   302.95000   523.00000+  401.76667   207.00000+  518.00000
 [73]   89.16667   654.00000   362.00000   364.00000  1125.00000+  423.00000+
 [79]  386.00000   421.00000+  345.00000+  108.00000+  304.00000+  331.00000
 [85]  248.00000   381.00000+  797.00000  1291.81667   296.00000+  527.11667
 [91]  224.73333   188.00000+  225.86667   261.00000+  542.00000   533.00000+
 [97]  306.00000   569.00000+  219.00000   158.00000
```

The survival object uses `los` for the timing variable (right censored) and `admitted` as the event.

Important note: the survival object created by `Surv()` requires `time =` to be a numeric variable, but the event can be any type. HOWEVER, some diagnostic procedures require it to be numeric, so we add `as.numeric()` when we specify `event =`.

The `head()` command allows you to visually inspect the distribution of events over time.

- Observations with a "+" are censored at the time indicated
- Observations showing only the time were events at the time indicated
- No missing values to address; no patterns in censoring or events are apparent

3.9 Fitting Survival Curves

```r
km1 <- survfit(los_admit ~ 1, data = df)

summary(km1, times = c(1, 24, 48, 72, 168, 350, 500, 1000, 2000))
```

```
Call: survfit(formula = los_admit ~ 1, data = df)

 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1    222       0   1.0000 0.00000       1.0000        1.000
   24    221       0   1.0000 0.00000       1.0000        1.000
   48    221       0   1.0000 0.00000       1.0000        1.000
   72    220       1   0.9955 0.00451       0.9867        1.000
  168    199      17   0.9177 0.01860       0.8819        0.955
  350    112      65   0.5957 0.03451       0.5318        0.667
  500     61      36   0.3845 0.03618       0.3198        0.462
 1000     22      21   0.2248 0.03456       0.1663        0.304
 2000      3      10   0.0901 0.03345       0.0435        0.187
```

Performing the `summary()` command on a fit survival object outputs a lifetable at specified timepoints.

`km1` holds the survival results for the full sample

3.10 Visualizing the KM Curve

Using `library(survminer)`, we can quickly generate Kaplan-Meier curves with our fit results.

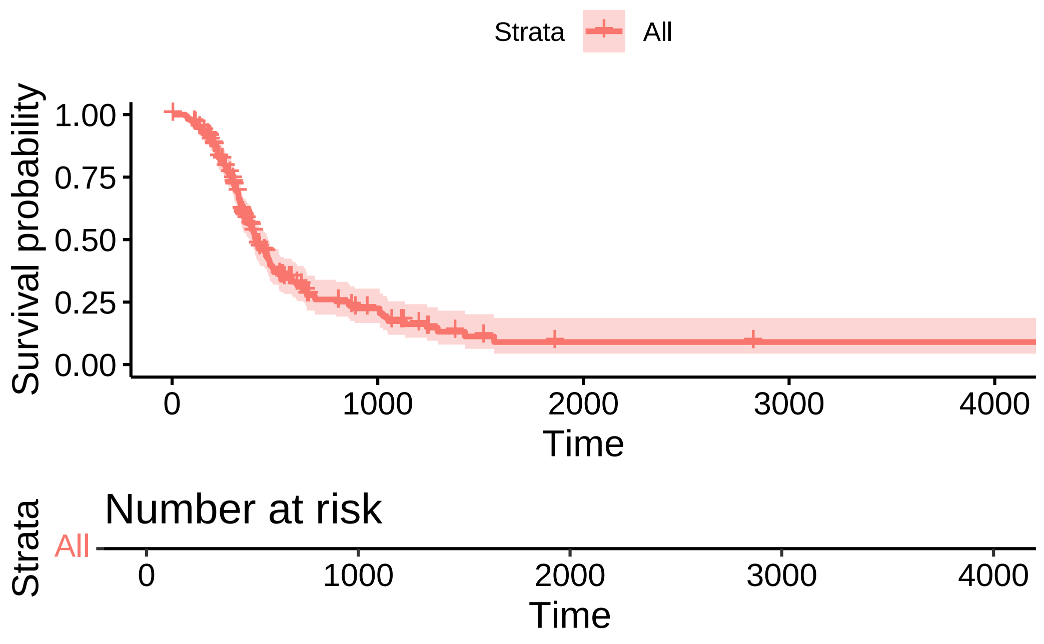
The command is `ggsurvplot()` and provides several options to customize results.

```r
library(survminer)
```

```
Loading required package: ggpubr
```

```
Attaching package: 'survminer'


The following object is masked from 'package:survival':

    myeloma
```

```
# Total Sample
kmplot1 <- ggsurvplot(fit = km1, data = df, risk.table = TRUE)
kmplot1
```



3.11 Comparing Survival by Groups

```
km2 <- survfit(los_admit ~ ambulance, data = df)

summary(km2, times = c(1, 24, 48, 72, 168, 350, 500, 1000, 2000))
```

```
Call: survfit(formula = los_admit ~ ambulance, data = df)

                ambulance=0
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1     89       0    1.000  0.0000       1.0000        1.000
   24     88       0    1.000  0.0000       1.0000        1.000
   48     88       0    1.000  0.0000       1.0000        1.000
   72     87       1    0.989  0.0113       0.9667        1.000
```

```
  168     78       4    0.942  0.0252       0.8937        0.993
  350     48      18    0.707  0.0517       0.6128        0.816
  500     20      20    0.374  0.0615       0.2711        0.516
 1000      4       8    0.178  0.0580       0.0942        0.338
 2000      1       1    0.134  0.0582       0.0570        0.314

                ambulance=1 
 time n.risk n.event survival std.err lower 95% CI upper 95% CI
    1    133       0    1.000  0.0000       1.0000        1.000
   24    133       0    1.000  0.0000       1.0000        1.000
   48    133       0    1.000  0.0000       1.0000        1.000
   72    133       0    1.000  0.0000       1.0000        1.000
  168    121      13    0.902  0.0258       0.8532        0.954
  350     64      47    0.529  0.0445       0.4490        0.624
  500     41      16    0.386  0.0447       0.3072        0.484
 1000     18      13    0.244  0.0425       0.1738        0.344
 2000      2       9    0.084  0.0381       0.0346        0.204
```

Because our independent variable has two categories, `summary()` will generate a lifetable for each one.

3.12 Statistical Tests

```
library(survival)

# log-rank test from library(survival)
survdiff(los_admit ~ ambulance, data = df, rho = 0)
```

```
Call:
survdiff(formula = los_admit ~ ambulance, data = df, rho = 0)

              N Observed Expected (O-E)^2/E (O-E)^2/V
ambulance=0  89       52       54    0.0734     0.117
ambulance=1 133       98       96    0.0413     0.117

 Chisq= 0.1  on 1 degrees of freedom, p= 0.7 
```

```
# peto test from library(survival)
survdiff(los_admit ~ ambulance, data = df, rho = 1)
```

```
Call:
survdiff(formula = los_admit ~ ambulance, data = df, rho = 1)

              N Observed Expected (O-E)^2/E (O-E)^2/V
ambulance=0  89     32.1     35.3     0.275     0.618
```

```
ambulance=1 133     61.0     57.9     0.168     0.618

 Chisq= 0.6  on 1 degrees of freedom, p= 0.4
```

Neither the *log-rank test*, nor the *Peto and Peto test* show any statistical difference in admission rates by arrival type.

4 Cox Proportional Hazards Regression

4.1 Background

- The multivariable analogue to Kaplan-Meier analysis
- Models a binary event over a time t
- Interested in associations w/ risk factors $x_1, x_2, ..., x_n$
- Outputs *beta estimates* and *hazard ratios* for each covariate, and can be interpreted like relative risks

4.2 Cumulative Hazard Function

This is the foundation of estimation with *Cox Proportional Hazards Regression.*

The "hazards" is the instantaneous rate of an event given that no event happened up to t.

$H(t) = -ln(S(t))$

Skipping the calculus, *hazard rate* is simply the *negative natural logarithm of survival rate.*

The *hazard ratio* is the ratio of these *hazard rates* between groups and represents the *relative risk.*

The hazard function $h(t)$ is defined as:

$h(t) = h_0(t) * e^{b_1 x_1 + ... + b_n x_n}$

4.3 Proportional Hazards Assumption

To estimate an *internally valid* effect a risk factor has on an event (aka *outcome*), we assume:

- The probability distribution is *memoryless*
- All participants are affected in approximately the same way by the risk factor
- The risk factor applies a constant amount of impact on risk for the event at all points in time
- Time does not affect manifestation the event

If all are safely assumed, survival analysis can proceed.

4.4 Commands

```
library(survival)

cox.m1 <- coxph(formula = Surv(time = timevar, event = outcomevar) ~ var1 + var2 + ... + varn, data

phcheck.m1 <- cox.zph(cox.m1)

summary(cox.m1)
```

An object called `cox.m1` contains the results of the fitted Cox model, where:

- `coxph()` is the primary command that performs the calculations
- `formula =` prefaces the listing of variables:
- `Surv(time = timevar, event = outcomevar)` is the survival object containing the primary outcome (`outcomevar`) and the timing variable (`timevar`)
- `~` separates the dependent variable portion from the independent variable portion of the formula
- `var1 + var2 + ... + varn` is the listing of independent variables to be included in the model
- `data = df` tells R what data frame to use and where to find the variables
- `cox.zph()` tests the proportional hazards assumption using the `cox.m1` object that is created in the modeling step
- `summary()` shows the results of the `cox.m1` object

4.5 Cox Model Output

```
cox.m1 <- coxph(formula = Surv(time = los, event = as.numeric(admitted)) ~ ambulance, data = df)

summary(cox.m1)
```

```
Call:
coxph(formula = Surv(time = los, event = as.numeric(admitted)) ~
    ambulance, data = df)

  n= 222, number of events= 150

              coef exp(coef) se(coef)     z Pr(>|z|)
ambulance1 0.05937   1.06116  0.17300 0.343    0.731

           exp(coef) exp(-coef) lower .95 upper .95
ambulance1     1.061     0.9424     0.756     1.489
```

```
Concordance= 0.521  (se = 0.022 )
Likelihood ratio test= 0.12  on 1 df,   p=0.7
Wald test            = 0.12  on 1 df,   p=0.7
Score (logrank) test = 0.12  on 1 df,   p=0.7
```

Statistical result is the same as the KM result because both are only accounting for the same single covariate `ambulance`.

The top table from `summary(cox.m1)` can be read as follows:

- `coef`: the beta coefficient for `ambulance` = 1 that is 0.05937
- `exp(coef)`: the hazard ratio for `ambulance` which is $e^{0.05937}$ or HR = 1.06116.
- `se(coef)`: the standard error of the beta coefficient
- `z`: the z-value from the test for difference of `ambulance` on a chi-square distribution
- `Pr(>|z|)`: the p-value for the independent variable(s)

The middle table lists the covariates in the model, the hazard ratio [`exp(coef)`], the inverse hazard ratio [`exp(-coef)`], and the lower and upper bounds of the 95% confidence interval around the hazard ratio.

The bottom listing includes:

- `Concordance`: Harrell's concordance statistic is a measure of model validity based on sensitivity and specificity
- `Likelihood ratio test`, `Wald test`, and `Score (logrank) test` are three statistical tests for the overall model fitness. These test a null hypothesis that the calculated coefficients for each covariate included are zero.

4.6 Proportional Hazards Assumption Testing

The command `cox.zph()` performed on the Cox model results will assess the proportional hazards assumption.

```r
library(survival)

phchk.m1 <- cox.zph(cox.m1)
phchk.m1
```

```
          chisq df    p
ambulance  1.93  1 0.17
GLOBAL     1.93  1 0.17
```

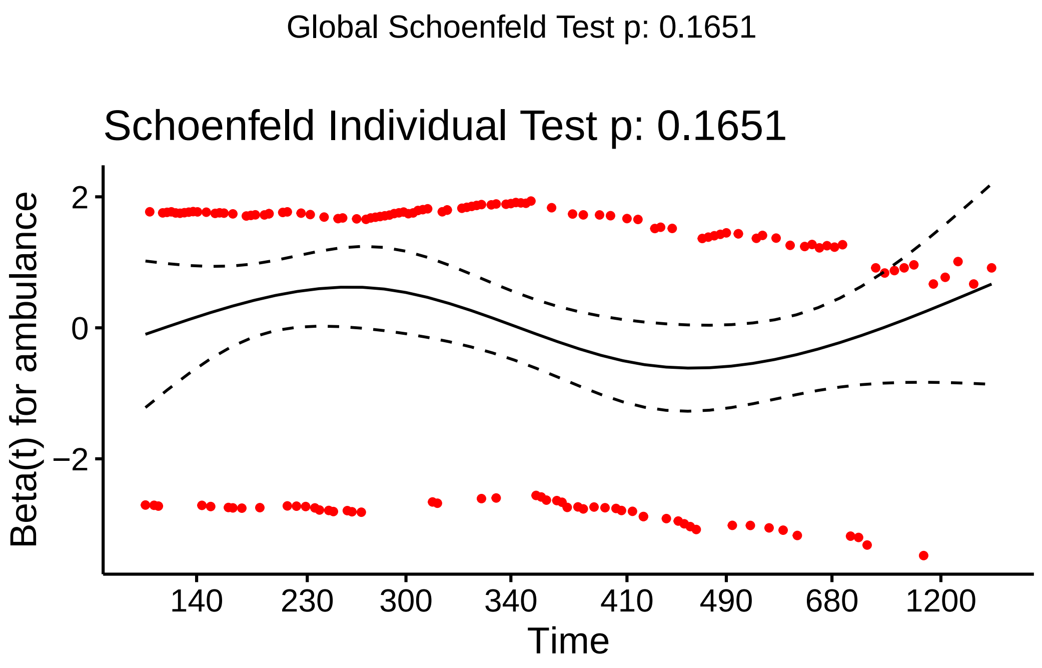
The p-value for test is > 0.050 so it appears that the proportional hazards assumption is still valid for this bivariate model.

Characteristic	HR	95% CI	p-value
ambulance			
0	—	—	
1	1.06	0.76, 1.49	0.7

Abbreviations: CI = Confidence Interval, HR = Hazard Ratio

4.7 Proportional Hazards Assumption With Schoenfeld Residuals

```r
library(survminer)
ggcoxzph(phchk.m1)
```



The *Schoenfeld Individual Test* is a test of the residuals' independence versus time.

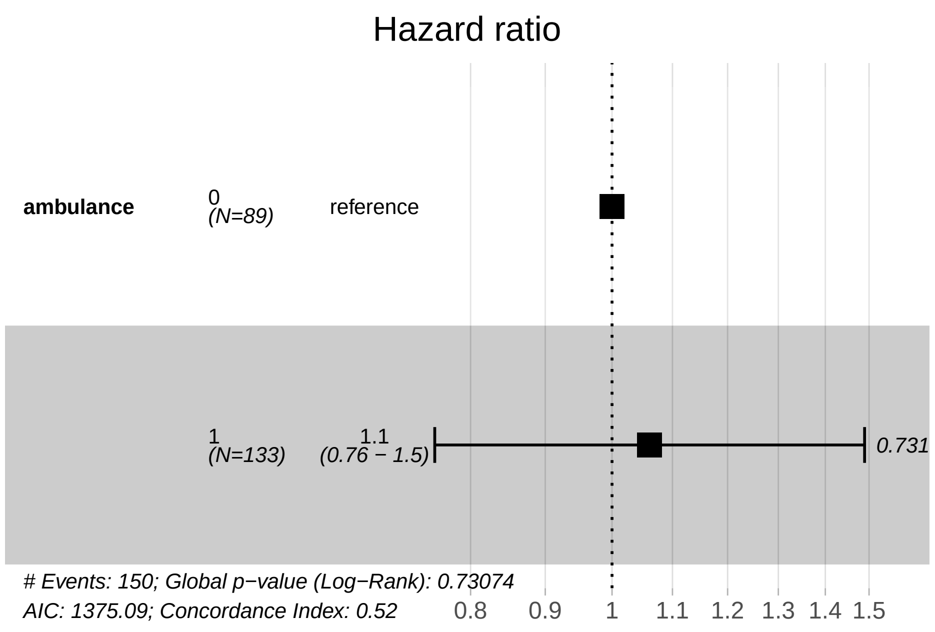
Residuals for each event and non-event as they leave the study should be approximately equally spread out and roughly symmetric.

The top set of red dots are *events* at each time. The bottom set are for *non-events*.

4.8 Visualization

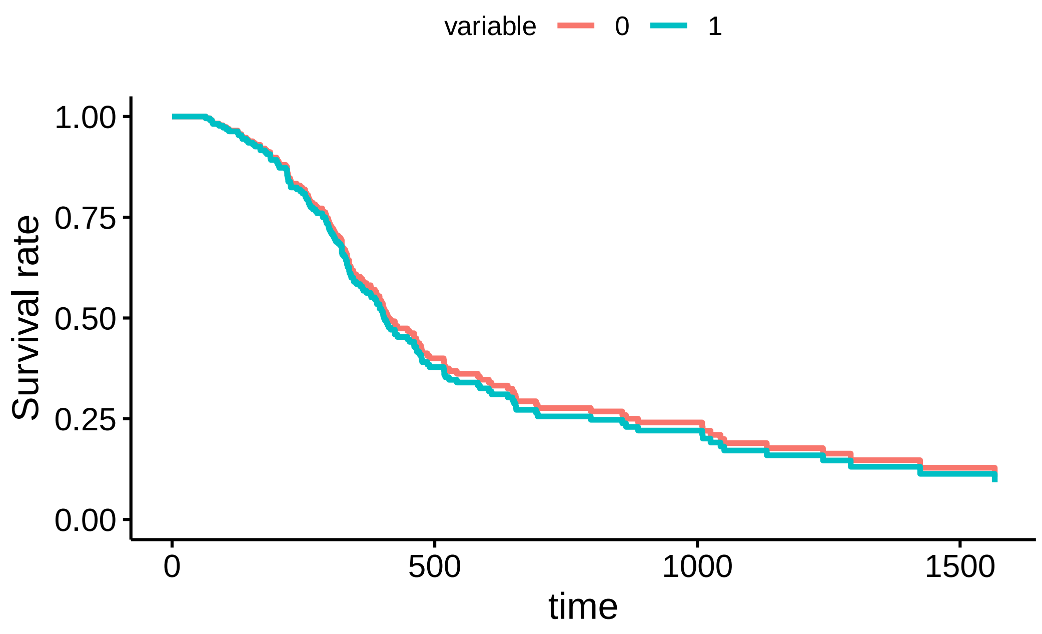
```r
library(gtsummary)

cox.m1 %>% tbl_regression(exp = TRUE)
```

```r
ggforest(cox.m1, data = df)
```



4.9 Survival Curves

```r
library(survminer)
ggadjustedcurves(cox.m1, variable = "ambulance", data = df)
```


variable
Survival rate
time