Second Assignment

Lúa Arconada & Alejandro Macías

2024-05-09

1. Consider a survival function with constant hazard $h(t) = 0.07$ when $0 \leq t \leq 5$, and $h(t) = 0.4$ for $t > 5$ (this is known as piecewise constant hazard).

Plot the hazard function and the corresponding survival function for $0 < t < 10$.

To plot the hazard function, we simply have to define it in R:

```
h = function(t){
  0.07 * ((0 <= t) & (t <= 5)) + 0.4 * (5 < t)
}
```

$$h(t) = \begin{cases} 0.07, & 0 \leq t \leq 5 \\ 0.4, & 5 < t \end{cases}$$

Obtaining the survival function from the hazard function is easy through the use of the following expression

$$S(t) = e^{-H(t)},$$

where $H(t) = \int_0^t h\,(u)du$.

First, we can easily compute $H(t)$:

$$H(t) = \begin{cases} 0.07t, & 0 \leq t \leq 5 \\ 0.4t, & 5 < t \end{cases}$$

```
H = function(t){
  t*h(t)
}
```

Finally, we compute $S(t)$:

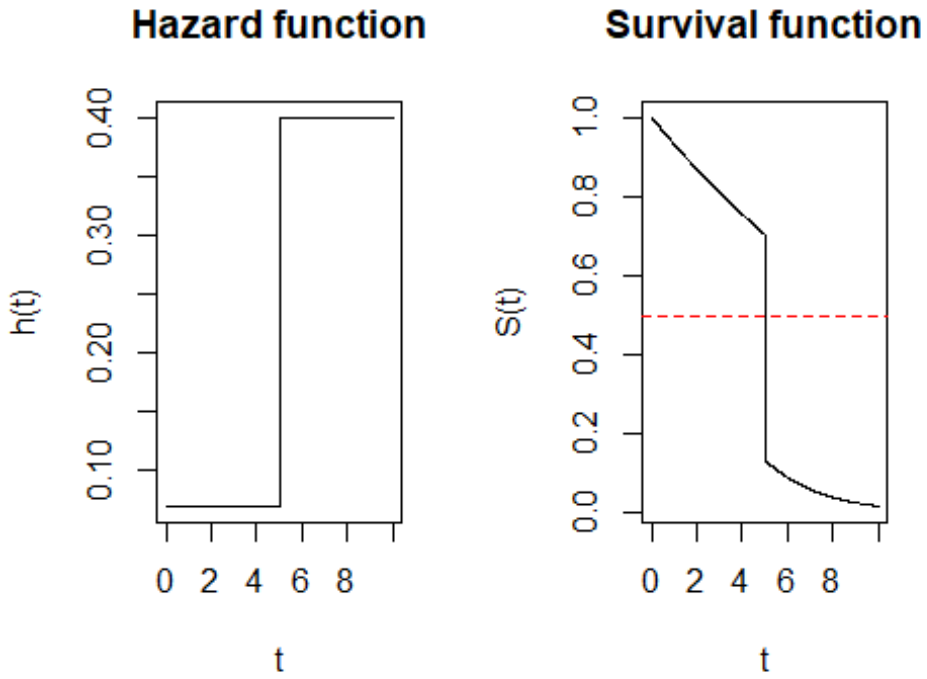
$$S(t) = \begin{cases} e^{-0.07t}, & 0 \leq t \leq 5 \\ e^{-0.4t}, & 5 < t \end{cases}$$

```
S = function(t){
  exp(-H(t))
}
```

We plot both:

```
par(mfrow=c(1,2))
t = seq(0,10,by=0.01)
```

```r
plot(t, h(t), type='l', main = 'Hazard function')
plot(t,S(t),type='l', main = 'Survival function')
abline(h=0.5, col="red", lty=2)
```



The red dotted line in the survival function's plot indicates where the function would meet the value 0.5, which will be handy for the next section of the exercise.

What is the median survival time?

Wa have to solve for t the following equation:

$$S(t) = 0.5$$

```r
equation <- function(t, c) {
  S(t) - 0.5
}

uniroot(equation, interval = c(0, 10))$root
```

```
## [1] 4.999973
```

As we can see, the median survival time is approximately 5 units of time.

2. Suppose that we assume that the time-to-event is a Rayleigh distribution with density function:

$$f(y) = (\lambda_0 + \lambda_1 y)e^{-\lambda_0 y - \frac{1}{2}\lambda_1 y^2}, \quad y > 0$$

Calculate the survival, hazard, and cumulative hazard functions.

To obtain the survival function starting from the density function, the following expression can be used:

$$S(t) = 1 - F(t) = \int_t^{+\infty} f\,(y)dy$$

Obtaining the survival function allows for the computation of the cumulative hazard function through the following expression:

$$H(t) = -\log\big(S(t)\big)$$

and using either of the previously obtained functions the hazard function can be derived:

$$h(t) = -\frac{d\log\big(S(t)\big)}{dt} = \frac{dH(t)}{dt}$$

For the case at hand, we simply have to realise that the expression of the density is easily integrable in order to obtain the survival function:

$$S(t) = \int_t^{+\infty} f\,(y)dy = \int_t^{+\infty} (\lambda_0 + \lambda_1 y)\, e^{-\lambda_0 y - \frac{1}{2}\lambda_1 y^2} dy = \Big[-e^{-\lambda_0 y - \frac{1}{2}\lambda_1 y^2}\Big]_t^{+\infty} = e^{-\lambda_0 t - \frac{1}{2}\lambda_1 t^2}$$

Next, we obtain cumulative hazard function:

$$H(t) = -\log\left(e^{-\lambda_0 t - \frac{1}{2}\lambda_1 t^2}\right) = \lambda_0 t + \frac{1}{2}\lambda_1 t^2$$

Finally, we compute the hazard function.

$$h(t) = \frac{d}{dt}\left(\lambda_0 t + \frac{1}{2}\lambda_1 t^2\right) = \lambda_0 + \lambda_1 t$$

Note that, in all of the cases above, λ_0 and λ_1 have been assumed non-negative in order for any of the steps taken to make sense.

3. The file `Henning.txt` contains data from a study of criminal recidivism by Henning and Frueh (1996), who followed 194 inmates released from a medium-security prison for a maximum of three years from the day of their release; during the period of study, 106 of the released prisoners were rearrested.

First of all, we load the dataset and take a look at the first few observations to see its structure.

```
# Read the data from the "Henning.txt" file, assuming it's a tabular data
file with headers
data <- read.table("Henning.txt", header = TRUE)
```

```r
# Display the first few rows of the data to inspect its structure
head(data)
```

```
##   id     months censor personal property       cage
## 1  1 0.06570842      0        1        1  -1.675198
## 2  2 0.13141684      0        0        1 -10.482864
## 3  3 0.22997947      0        1        1  -4.426738
## 4  4 0.29568789      0        0        1 -11.328860
## 5  5 0.29568789      0        1        1  -7.164589
## 6  6 0.32854209      0        1        0  -2.868901
```

Compute and plot the Kaplan-Meier estimate of the survival function for all of the data.

The provided code employs the Kaplan-Meier estimator, a fundamental tool in survival analysis for estimating the probability that an individual survives beyond a certain time point, called the survival function ($S(t)$). This method is particularly suited for censored data, where observations may not have experienced the event of interest by the end of the study period or are incomplete due to other reasons such as loss to follow-up. In essence, it allows us to estimate the survival probabilities over time, even when some observations are censored.

Within the code, the survival times of individuals (`data$months`) and their corresponding event indicators (`data$censor`) are utilized to construct a survival object (`Surv()` function). This object encapsulates the necessary information about the time to event and the event status of each observation. It serves as the foundation for subsequent survival analysis.

The Kaplan-Meier estimator is then applied through the `survfit()` function, which computes the survival probabilities at distinct time points where events occur or censoring events are observed. This estimation method accounts for the presence of censored data, adjusting the survival probabilities accordingly. It works in the following way:

1. At each event time t_i, count the number of individuals who have not experienced the event before t_i (individuals who are still at risk).

2. Compute the probability of survival at t_i as the ratio of the number of individuals who have not experienced the event by time t_i to the number of individuals at risk just before t_i.

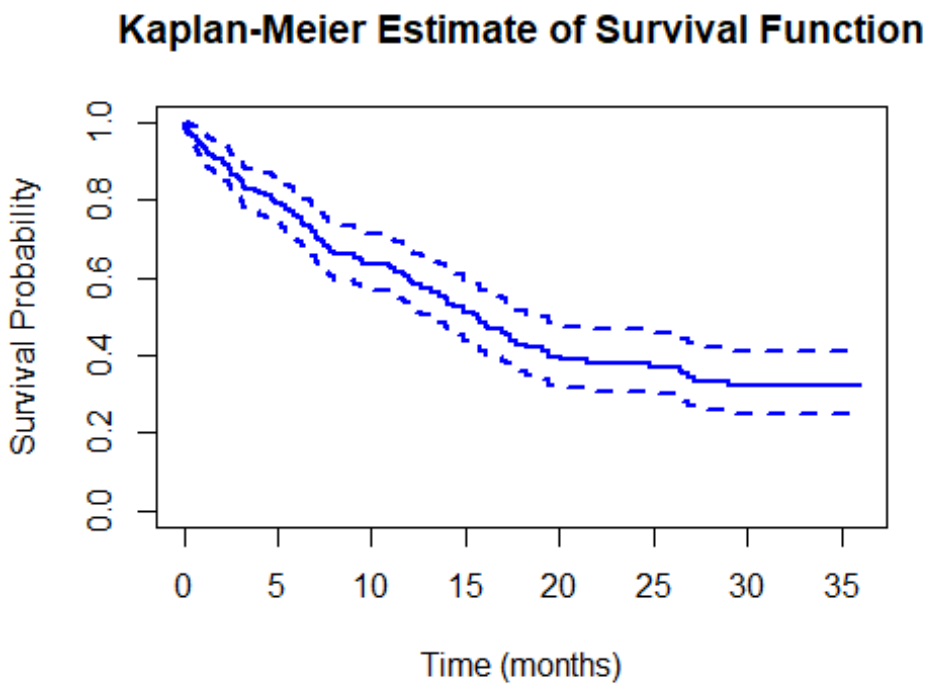
3. Multiply the previous survival probability by the computed survival probability to obtain the Kaplan-Meier estimate of the survival function.

By following these steps, you can manually compute the Kaplan-Meier estimate of the survival function for your dataset without relying on specific statistical software like R. However, performing these calculations manually can be time-consuming and prone to errors, especially for large datasets. Using statistical software such as R or

Python with libraries like `lifelines` can automate these computations and provide more efficient and accurate results.

Finally, the resulting survival curve is visualized using the `plot()` function. This graphical representation provides a clear depiction of how survival probabilities evolve over time in our study population of inmates.

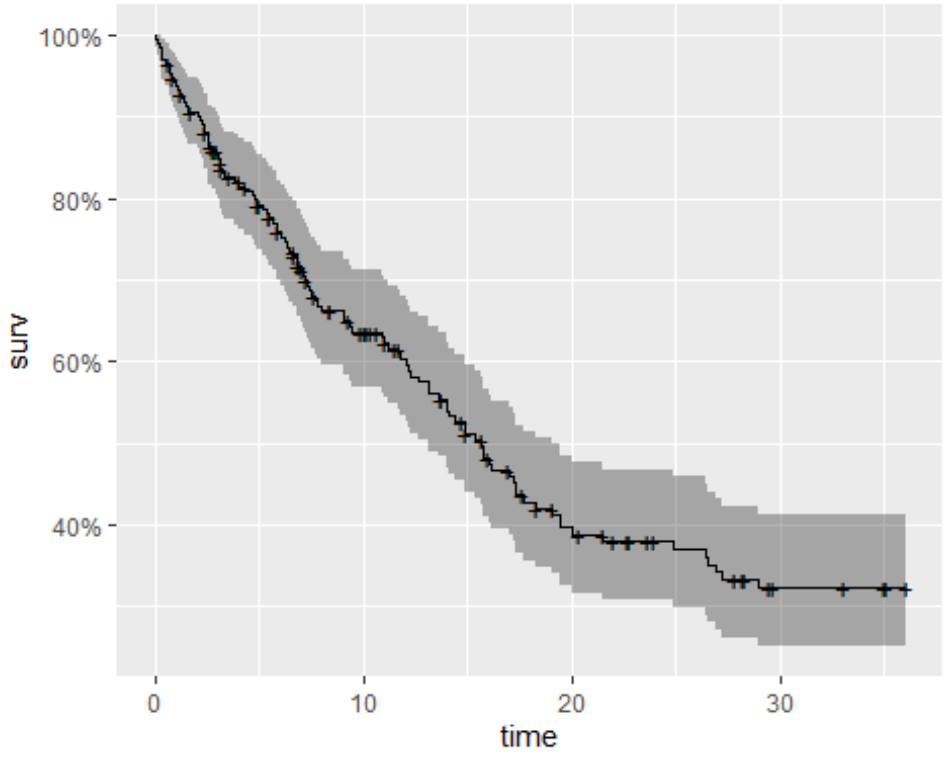
```r
# Load the survival package for survival analysis functions
library(survival)

# Create a survival object using the Surv() function
# The Surv() function constructs a survival object with time-to-event data and censoring indicators
# Here, data$months represents survival times and data$censor represents event indicators (coded inversely)
surv_obj <- Surv(time = data$months, event = 1 - data$censor)

# Fit the Kaplan-Meier estimator using the survfit() function
# The formula ~ 1 indicates that we're estimating the survival function without grouping or stratification
# The survfit() function calculates the Kaplan-Meier survival estimates based on the survival object
km_fit <- survfit(surv_obj ~ 1)

# Plot the Kaplan-Meier survival curve
plot(
  km_fit,
  main = "Kaplan-Meier Estimate of Survival Function",  # Title of the plot
  xlab = "Time (months)",  # Label for the x-axis indicating time
  ylab = "Survival Probability",  # Label for the y-axis indicating survival probability
  col = "blue",  # Color of the survival curve (can be specified as a character string or numeric code)
  lty = 1,  # Line type of the survival curve (1 = solid line)
  lwd = 2  # Line width of the survival curve
)
```



We can plot in another, more automated way using the `autoplot()` function. When used with a survfit object like `km_fit`, it automatically selects an appropriate visualization method based on the object type and displays our survival curve. It provides a quick and easy way to generate plots with default settings, saving us from specifying plot parameters manually.

We can see some differences in the appearance. For example, the y-axis changes from displaying the survival probabilities as a number between 0 and 1 to a percentage.

```
# Load the ggplot2 package for data visualization
library(ggplot2)

# Load the ggfortify package for enhanced visualization of survival
analysis results
library(ggfortify)

# Generate an automatic plot of the Kaplan-Meier survival curve using
autoplot() function
# The autoplot() function automatically selects an appropriate
visualization method based on the input object
# In this case, it plots the Kaplan-Meier survival curve from the survfit
object (km_fit)
autoplot(km_fit)
```



Compute and plot separate survival curves for those with and without a record of crime against persons; test for differences between the two survival functions.

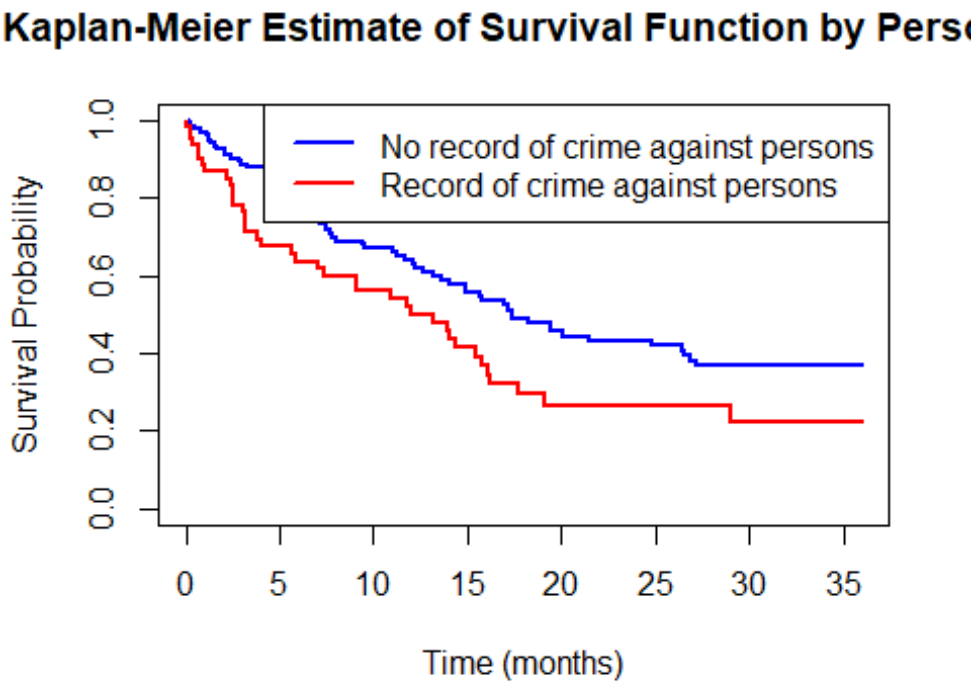
We are going to follow the same process as in the previous subwording of the exercise, we are still utilizing the `survival` package in R to compute Kaplan-Meier estimates and plot survival curves. However, we are know computing the Kaplan-Meier estimates for each group defined by the variable `personal`, which indicates whether individuals have a record of crime against persons or not. This introduces the concept of stratification, where survival curves are generated separately for each level of the grouping variable.

Since we are stratifying by the `personal` variable, the `plot()` function is used to generate separate survival curves for each group and we specify different colors for each group to differentiate them in the plot. Moreover, we add a legend to the plot to provide information about the groups represented by the different colors in the plot, helping to interpret the survival curves more easily.

This second plot compares visually survival curves between the different groups, which provides insights into how the presence or absence of a record of crime against persons may influence survival probabilities over time.

```
# Create a survival object
surv_obj <- Surv(time = data$months, event = 1 - data$censor)

# Compute Kaplan-Meier estimates for each group defined by 'personal'
personal = data$personal
```

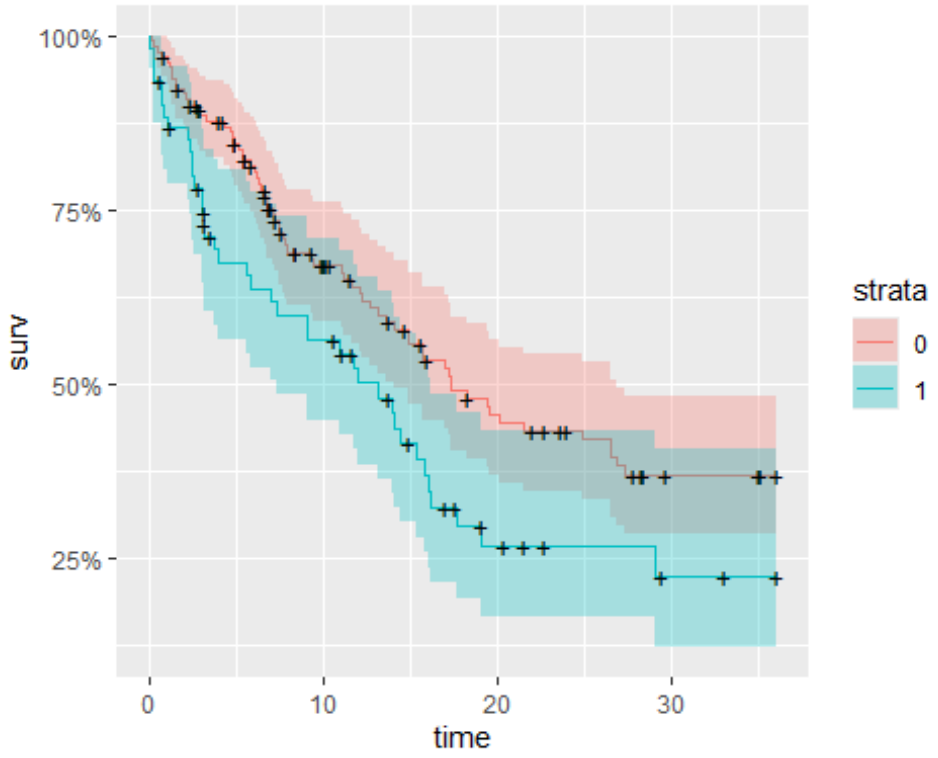
```r
km_fit_personal <- survfit(surv_obj ~ personal)

# Plot Kaplan-Meier estimates for each group
plot(
  km_fit_personal,  # Kaplan-Meier survival estimates
  col = c("blue", "red"),  # Color of the survival curves for each group
  lty = 1,  # Line type of the survival curves (1 = solid line)
  lwd = 2,  # Line width of the survival curves
  main = "Kaplan-Meier Estimate of Survival Function by Personal",  # 
Title of the plot
  xlab = "Time (months)",  # Label for the x-axis indicating time
  ylab = "Survival Probability"  # Label for the y-axis indicating 
survival probability
)

# Add legend
legend(
  "topright",  # Position of the legend
  legend = c("No record of crime against persons", "Record of crime 
against persons"),  # Labels for the groups
  col = c("blue", "red"),  # Colors corresponding to each group
  lty = 1,  # Line type of the legend
  lwd = 2  # Line width of the legend
)
```



We can see that the blue line, which corresponds with the inmates with no record of crime against persons, is always above the one corresponding to the inmates that do

have record of crime against persons. This may lead us to think that the first group has more chances of survival (staying out of prison) than the second group. However, we cannot be sure, we are going to carry out a test to see it; but before we take a look at another plot of this curves.

```
# Create an autoplot of the Kaplan-Meier survival estimates for each
group defined by 'personal'
# This function automatically generates a plot using ggplot2, which is
more flexible and customizable than the base R plot function
autoplot(km_fit_personal)
```



However, in this new plot, we can see that the confidence intervals of both curves overlap, and the curve representing the group with a lower survival probability (inmates with a record of crime against persons) occasionally crosses over the curve representing the group with a higher survival probability. This makes us doubt our previous belief, as it suggests that there may not be a significant difference in survival between the two groups. For those who do have a record of crime against persons, survival chances go down to below 25%, while those who do not have said record see their survival chances drop to around 27%.

In statistical terms, the overlapping confidence intervals leads us to believe that the difference in survival probabilities between the two groups is not statistically significant at certain points in time. Additionally, the occasional crossing of the survival curves suggests that the difference in survival probabilities between the two groups may not be consistent over time.

Depending on the plot, we have drawn to opposite conclusions, so we cannot be sure if the Kaplan-Meier survival curves confidently conclude that one group has a consistently higher or lower chance of survival (staying out of prison) compared to the other group or the contrary hypothesis, they have the same chance of survival between the two groups.

Further statistical analysis, such as log-rank tests or Cox proportional hazards models, is necessary to assess the significance of any observed differences in survival between the two groups. We are going to perform a log-rank test.

The log-rank test is a statistical hypothesis test used to compare the survival distributions of two or more groups. It is commonly employed in survival analysis to determine whether there are significant differences in survival probabilities between groups over time, which is exactly what we wanna see since we have contradictory intuitions.

The test begins with survival data and a binary indicator variable representing group membership (`person`). From this data, the test evaluates the null hypothesis that there is no difference in the survival distributions between the groups. The alternative hypothesis posits that there is a difference in survival distributions.

For each group, the test calculates the observed number of events and the expected number of events under the null hypothesis. The expected number of events is calculated based on the assumption that the survival distributions of the groups are the same.

The log-rank test statistic is then computed based on the difference between the observed and expected numbers of events for each group. It essentially measures how much the observed survival experience differs from what would be expected if the null hypothesis were true.

```r
# Perform the log-rank test
logrank_test <- survdiff(surv_obj ~ data$personal)

# Display the test results
logrank_test
```

```
## Call:
## survdiff(formula = surv_obj ~ data$personal)
##
##                     N Observed Expected (O-E)^2/E (O-E)^2/V
## data$personal=0 133       67     77.8      1.50       5.7
## data$personal=1  61       39     28.2      4.14       5.7
##
##  Chisq= 5.7  on 1 degrees of freedom, p= 0.02
```

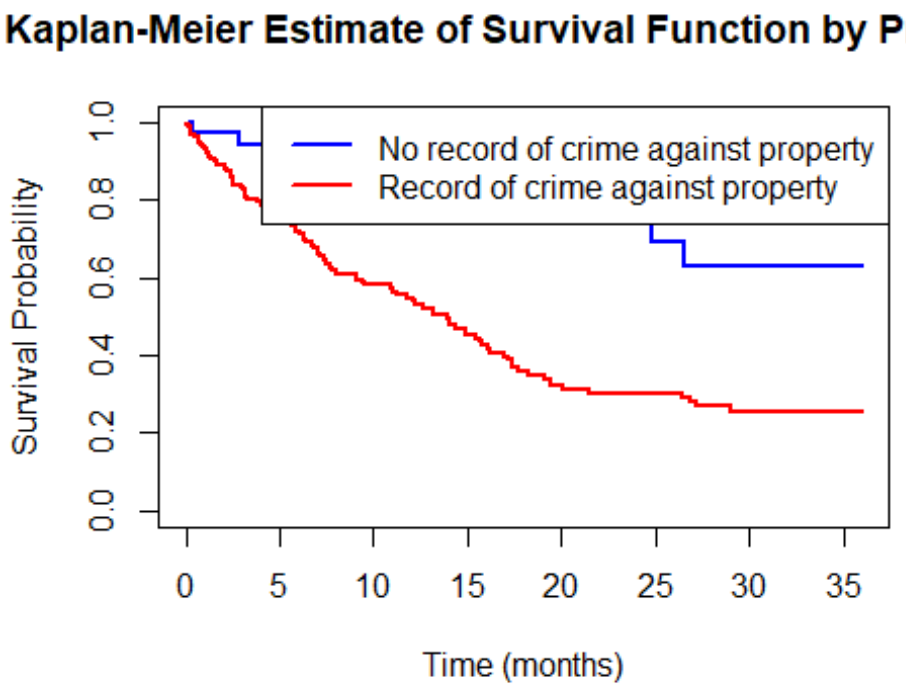
The computed p-value is 0.02, which indicates that there is statistically significant evidence against the null hypothesis (that the survival curves for the two groups are the same) at the conventional significance level of $\alpha = 0.05$. Therefore, we can

conclude that there are differences between the survival functions of individuals with and without a record of crime against persons (our first intuition).

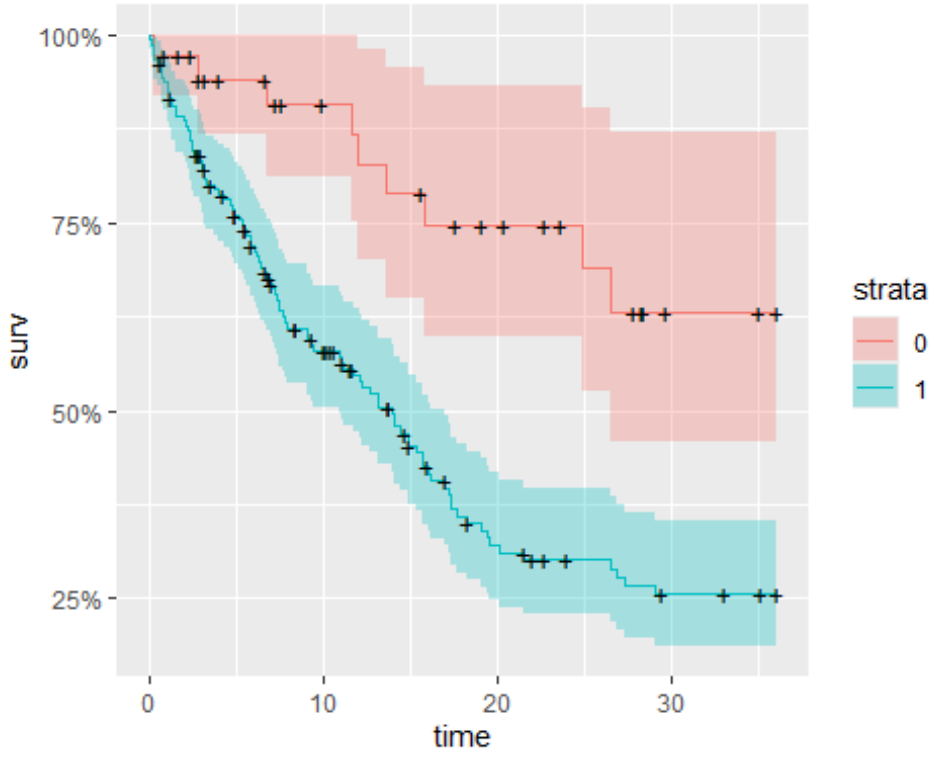
Compute and plot separate survival curves for those with and without a record of crime against property; test for differences between the two survival functions.

We are going to repeat the same process as in the previous subwording of the exercise, but now the groups are determined by the `property` variables, instead of the `personal` variable.

```
# Define the property variable
property = data$property

# Fit the Kaplan-Meier estimator for each group defined by 'property'
km_fit_property <- survfit(surv_obj ~ property)

# Plot Kaplan-Meier estimates for each group
plot(
  km_fit_property,  # Kaplan-Meier survival estimates
  col = c("blue", "red"),  # Colors for each group
  lty = 1,  # Line type of the survival curves (1 = solid line)
  lwd = 2,  # Line width of the survival curves
  main = "Kaplan-Meier Estimate of Survival Function by Property",  #
Title of the plot
  xlab = "Time (months)",  # Label for the x-axis indicating time
  ylab = "Survival Probability"  # Label for the y-axis indicating
survival probability
)

# Add legend
legend(
  "topright",  # Position of the legend
  legend = c("No record of crime against property", "Record of crime
against property"),  # Labels for the groups
  col = c("blue", "red"),  # Colors corresponding to each group
  lty = 1,  # Line type of the legend
  lwd = 2  # Line width of the legend
)
```




```r
# Create an autoplot of the Kaplan-Meier survival estimates for each group defined by 'property'
# This function automatically generates a plot using ggplot2, which is more flexible and customizable than the base R plot function
autoplot(km_fit_property)
```



This time, both plots lead us to believe the same thing: the survival chances are different depending on the group. Inmates with no record of crime against property have a much higher chance of surviving (staying out of prison) than those who do have a record of crime against property. In those who do not have a record, the chance only goes down to around 62%, while, on the other hand, for those who do have a record, their chance goes down to 25%.

However, even though we are quite sure of this assumption, we are going to carry out a log-rank test nevertheless.

```
# Perform the log-rank test
logrank_test <- survdiff(surv_obj ~ data$property)

# Display the test results
logrank_test
```

```
## Call:
## survdiff(formula = surv_obj ~ data$property)
##
##                     N Observed Expected (O-E)^2/E (O-E)^2/V
## data$property=0  36        9     24.7      9.97      13.1
## data$property=1 158       97     81.3      3.02      13.1
##
##  Chisq= 13.1  on 1 degrees of freedom, p= 3e-04
```

The very small p-value indicates that there is strong evidence (stronger than in the previous test, which makes sense since the difference between the curves is more

visually clear) against the null hypothesis (that the survival curves for the two groups are the same) at any reasonable significance level. Therefore, we can conclude that there are statistically significant differences between the survival functions of individuals with and without a record of crime against property.

Fit a Cox regression of time to re-arrest on the covariates personal. property, and cage.

A Cox regression model is a statistical technique used for analyzing the relationship between the survival time of subjects and one or more predictor variables. The Cox regression model assumes that the hazard (the probability of an event occurring at a particular time given that it has not occurred before) for any individual is the product of a baseline hazard function and a set of covariates (predictor variables) raised to unknown coefficients. Importantly, the Cox model does not assume a specific distribution for the survival times, making it a semiparametric model.

We fit a Cox regression model using the `coxph` function. This model examines the relationship between survival time and several explanatory variables, `personal`, `property` and 'cage.

```
# Fit a Cox regression model
cox_model <- coxph(Surv(months, censor) ~ personal + property + cage,
data = data)

# Display the summary of the model
cox_model
```

```
## Call:
## coxph(formula = Surv(months, censor) ~ personal + property +
##     cage, data = data)
##
##               coef exp(coef) se(coef)      z        p
## personal  0.34006   1.40504  0.25678  1.324 0.185382
## property -0.06842   0.93387  0.25255 -0.271 0.786457
## cage      0.04384   1.04481  0.01236  3.548 0.000389
##
## Likelihood ratio test=14.74  on 3 df, p=0.002049
## n= 194, number of events= 88
```

(i) Determine by a Wald test whether each estimated coefficient is statistically significant.

We compute the Wald test within the context of our Cox regression model, and we display the p-values associated with the tests to determine if each coefficient is statistically significant or not.

```
# Compute the Wald test for coefficients in a Cox proportional hazards
model
wald_test <- summary(cox_model)

# Extract the p-values for the coefficients
wald_test$coefficients[, "Pr(>|z|)"]
```

```
##     personal     property         cage
## 0.1853823985 0.7864570430 0.0003885532
```

The p-value associated with the coefficient for the variable `personal` is approximately 0.1854. This suggests that the coefficient for `personal` is not statistically significant at the conventional significance level of $\alpha = 0.05$, since we cannot reject the null hypothesis that it is not statistically significant (equal to 0).

Similarly, the p-value associated with the coefficient for the variable `property` is approximately 0.7865. This indicates that the coefficient for `property` is also not statistically significant at the conventional significance level of $\alpha = 0.05$.

However, the p-value associated with the coefficient for the variable `cage` is approximately 0.0004. This suggests that the coefficient for `cage` is statistically significant at the conventional significance level of $\alpha = 0.05$. Therefore, the centered age at the rime of release (`cage`) appears to have a significant effect on the risk of re-arrest in this model.

(ii) Interpret each of the estimate Cox-regression coefficients.

The model estimates the hazard ratios, which represent the proportional change in the hazard for one unit change in the predictor variable. A hazard ratio greater than 1 indicates an increased risk of the event, while a hazard ratio less than 1 indicates a decreased risk. The significance of the coefficients and hazard ratios is assessed using statistical tests.

We extract the coefficients of the Cox regression model.

```
# Display the coefficients of the Cox proportional hazards model
cox_model$coefficients
```

```
##    personal    property        cage
##  0.34006438 -0.06842002  0.04383763
```

These are the raw coefficients estimated by the Cox regression model. They represent the change in the log hazard ratio associated with a one-unit change in the predictor variable, holding other variables constant. For example, a coefficient of 0.34006 for `personal` means that individuals with a record of crime against persons have a log hazard ratio `0.34006` units higher than those without such a record.

```
# Display the exponential of the coefficients of the Cox proportional
hazards model
exp(cox_model$coefficients)
```

```
##  personal  property      cage
## 1.4050381 0.9338682 1.0448127
```

These are the exponential transformations of the coefficients, also known as hazard ratios. They represent the multiplicative effect of a one-unit change in the predictor variable on the hazard of the event of interest. For example, a hazard ratio of `$1.405$`

`forpersonal` means that individuals with a record of crime against persons have a 1.405 times higher hazard of re-arrest compared to those without such a record.

We interpret all the coefficients.

1. `Personal`: The coefficient for the variable `personal` is 0.34006. This means that individuals with a record of crime against persons have an $exp(0.34006) = 1.405$ times higher hazard of re-arrest compared to those without such a record, holding other variables constant. However, remember that the coefficient is not statistically significant (p-value $= 0.1854$).

2. `Property`: The coefficient for the variable `property` is -0.06842. This means that individuals with a record of crime against property have an $exp(-0.06842) = 0.934$ times lower hazard of re-arrest compared to those without such a record, holding other variables constant. Similar to `personal`, the coefficient is not statistically significant (p-value $= 0.7865$).

3. `Cage`: The coefficient for the variable `cage` is 0.04384. This means that for every one-unit increase in the centered age at the time of release (`cage`), the hazard of re-arrest increases by $exp(0.04384) = 1.045$ times, holding other variables constant. The coefficient is statistically significant (p-value $= 0.0004$), suggesting that age has a significant effect on the hazard of re-arrest.

These interpretations provide insights into how each variable affects the hazard of re-arrest in the Cox regression model, considering their coefficient values and statistical significance.

4. Given a hazard function $h(t) = c$, where $c > 0$, derive the survival and the density function. Calculate the median failure time for $c = 5$.

Survival Function $S(t)$

The survival function represents the probability that a subject survives beyond time t. We have seen that it can be computed using the following expression:

$$S(t) = \exp\big(-H(t)\big) = \exp\left(-\int_0^t h\,(u)\,du\right),$$

where $H(t)$ is the cumulative hazard function or the integrated hazard function and we have used the following definition in the second equality:

$$H(t) = \int_0^t h\,(u)du.$$

Now, using our data $h(t) = c$, the integral becomes:

$$\int_0^t h\,(u)du = \int_0^t c\ du = ct$$

So, the survival function becomes:

$$S(t) = e^{-ct}$$

Density Function $f(t)$

The density function represents the probability density of failure occurring at time t. We use the following equation to compute it:

$$h(t) = \frac{f(t)}{S(t)} \Rightarrow f(t) = h(t)S(t),$$

where, as we know and computed, $h(t) = c$ and $S(t) = e^{-ct}$. We can solve for $f(t)$ to obtain our density function $f(t)$:

$$f(t) = ce^{-ct}$$

Median Failure Time

The median failure time represents the point in time at which half of the subjects in a survival analysis have experienced the event of interest (such as death, failure, etc.), while the other half have not.

To find the median failure time, we need to solve for t in the survival function equation when $S(t) = 0.5$, using that $c = 5$.

$$e^{-5t} = 0.5$$

Taking the natural logarithm of both sides:

$$-5t = \ln(0.5)$$

$$t = -\frac{\ln(0.5)}{5}$$

$$t \approx 0.1386$$

So, the median failure time for $c = 5$ is approximately 0.1386 units of time.

5. Consider and explain briefly the case of recurrent events, and show an example by using R.

Recurrent events occur when an individual can experience the same type of event multiple times over the course of the study, for example, non-fatal events. These events are not independent, as the occurrence of one event may affect the likelihood or timing of subsequent events. Recurrent events are common in medical research and can include events such as hospitalizations, disease exacerbation, or re-occurrences of a specific medical condition.

When analyzing recurrent events, the simplest way to analyze a recurrent event data is to focus on time to the first occurrence, reducing the problem to that of a univariate event time. However, it is proven to be very inefficient. Specialized statistical methods

are required to properly account for the correlation between events within the same individual and to appropriately handle censoring. Some commonly used methods for analyzing recurrent events include:

1. Counting Process Models: These models treat each event as a separate observation and model the hazard rate as a function of time-varying covariates.
2. Prentice, Williams, and Peterson Model: This model extends the Cox proportional hazards model to handle recurrent events by modeling the cumulative hazard of the k-th event.
3. Shared Frailty Models: These models introduce random effects at the individual level to account for unobserved heterogeneity in the baseline hazard.
4. Multi-State Models: These models represent the different states an individual can transition between over time and can accommodate recurrent events as well as competing risks.

The choice of method depends on the specific research question, study design, and assumptions about the underlying data generating process. Properly accounting for recurrent events is crucial for obtaining valid and reliable estimates of event occurrence and for making accurate predictions about future events.

Here is a brief example.

```r
# Load the survival package
library(survival)

# Simulate recurrent event data
set.seed(123)
n <- 1000  # Number of individuals
time <- matrix(rexp(5 * n), nrow = n, ncol = 5)  # Generate event times
for 5 events
status <- matrix(sample(0:1, 5 * n, replace = TRUE), nrow = n, ncol = 5)
# Generate event status (0=censored, 1=event)
group <- sample(1:2, n, replace = TRUE)  # Generate group indicator
variable

# Combine event times and statuses into long format
time_long <- as.vector(time)
status_long <- as.vector(status)
group_long <- rep(group, each = 5)

# Create a data frame with simulated data
data <- data.frame(
  time = time_long,
  status = status_long,
  group = group_long
)
```

```r
# Fit a Cox proportional hazards model for recurrent events
fit <- coxph(Surv(time, status) ~ group, data = data)

# Display summary of the fitted model
summary(fit)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ group, data = data)
##
##   n= 5000, number of events= 2441
##
##            coef exp(coef) se(coef)      z Pr(>|z|)
## group -0.01265   0.98743  0.04051 -0.312    0.755
##
##       exp(coef) exp(-coef) lower .95 upper .95
## group    0.9874      1.013    0.9121     1.069
##
## Concordance= 0.495  (se = 0.006 )
## Likelihood ratio test= 0.1  on 1 df,   p=0.8
## Wald test            = 0.1  on 1 df,   p=0.8
## Score (logrank) test = 0.1  on 1 df,   p=0.8
```

In the fitted Cox proportional hazards model, the coefficient for the group variable is -0.01265. This coefficient represents the log hazard ratio comparing the two groups. Since the coefficient is close to zero and not statistically significant (p-value $= 0.755$), we do not have evidence to suggest a difference in the hazard of the event between the two groups.

The hazard ratio associated with the group variable is 0.9874. This hazard ratio indicates that individuals in the second group (compared to the first/reference group) have a hazard of experiencing the event that is 0.9874 times the hazard of individuals in the reference group. However, since the hazard ratio is close to 1, it suggests no practical difference in the hazard of the event between the two groups.

The concordance statistic of 0.495 indicates that the model's ability to discriminate between individuals who experience the event at different times is poor.

The likelihood ratio test, Wald test, and Score (log-rank) test all have p-values greater than 0.05, indicating that the group variable is not statistically significant in predicting the event occurrence.

Overall, based on this analysis, there is no evidence to suggest a significant difference in the hazard of the event between the two groups.

6. Develop a brief example of Survival Analysis from a Bayesian point of view using the rstanarm and dynsurv packages.

```r
library(dynsurv)
library(survival)
library(ggfortify)
```

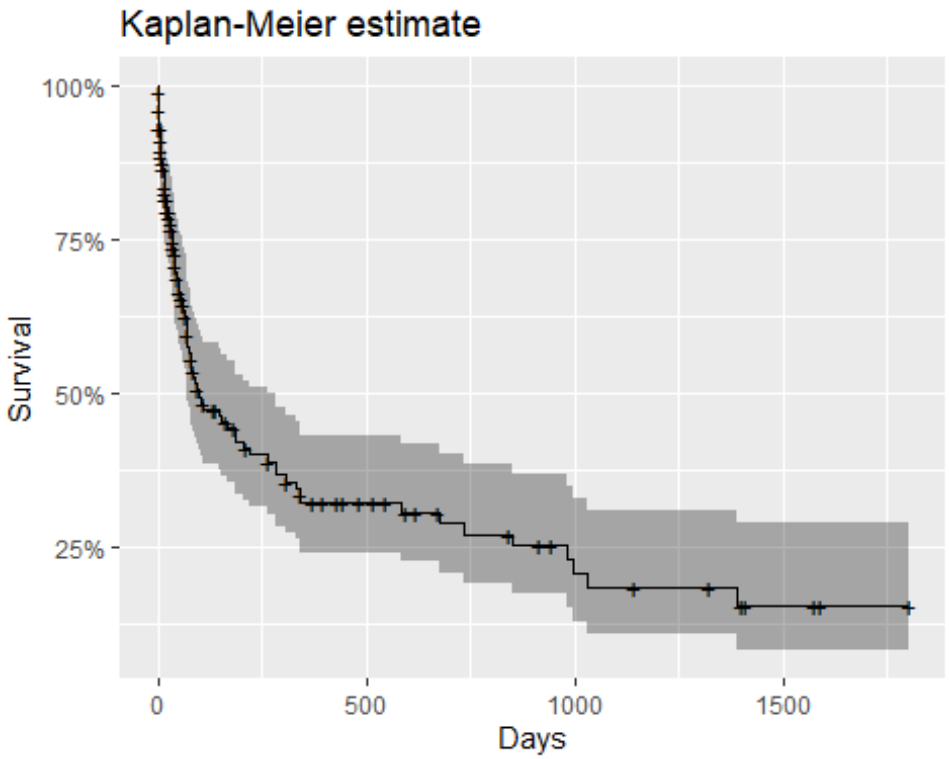
The examples developed for this exercise will use the `heart` dataset from the `survival` library. The 172 patients in this dataset are those within the waiting list for the Stanford heart transplant program. It contains information in `start-stop` format about the age of the patients (-48 years), the year of acceptance into the program (in years after 1 Nov 1967), whether they have had a prior bypass surgery or whether they have received a transplant. The event of interest in this case is death.

```
data(heart, package="survival")
heart$surgery = factor(heart$surgery)
```

dynsurv

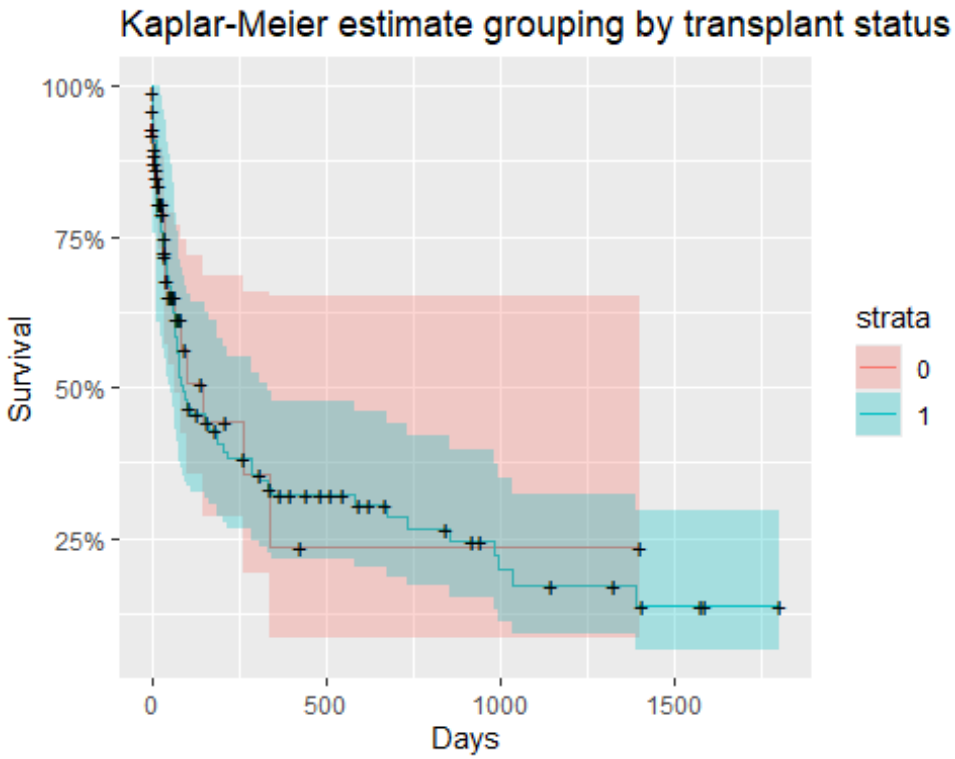
First, in order to first visualise the dataset, we can compute the Kaplan-Meier estimate of the survival function.

```
km0 = survfit(Surv(start, stop, event)~1, data=heart)
autoplot(km0, main="Kaplan-Meier estimate", xlab="Days", ylab="Survival")
```

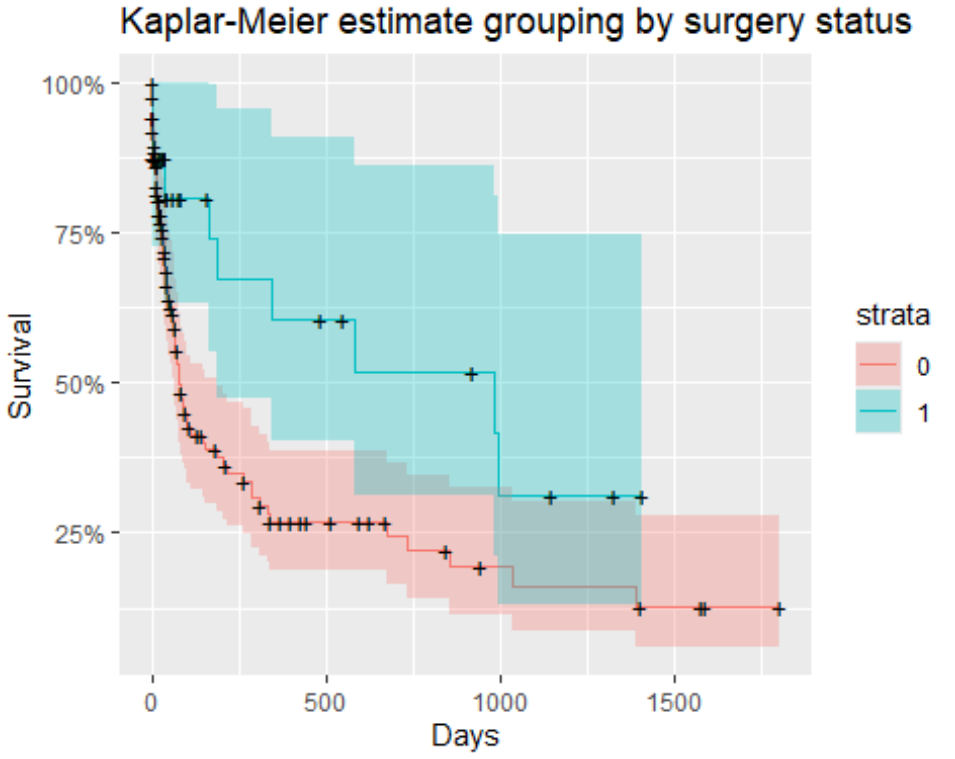


We can also check the survival functions for different combinations of patients who have had a transplant or surgery.

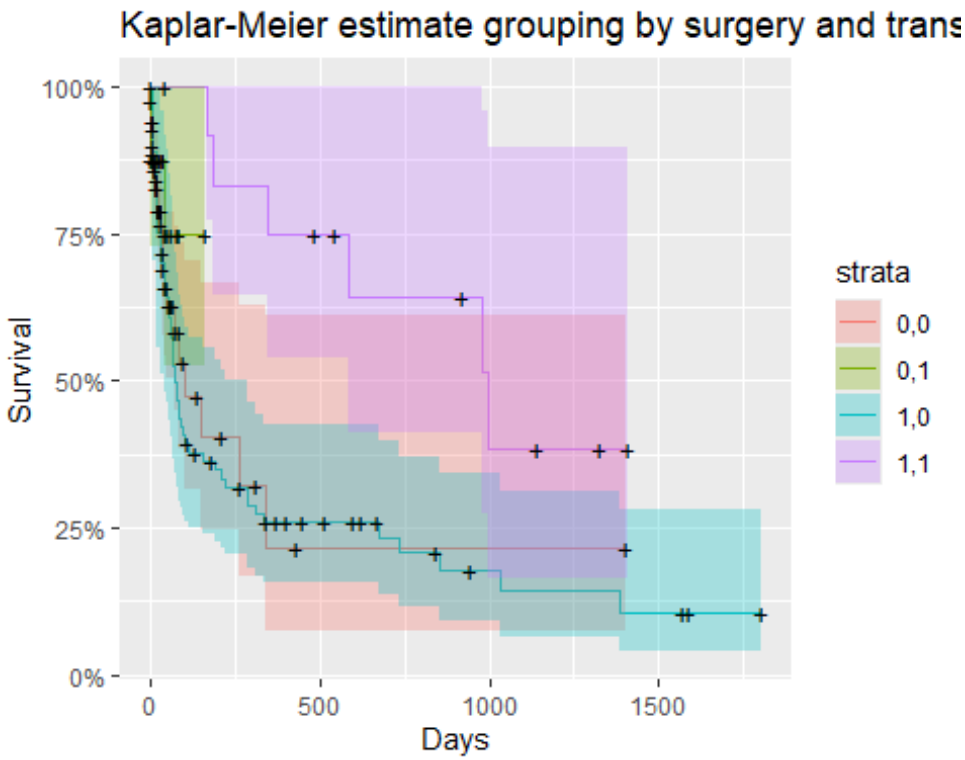
```
km1 = survfit(Surv(start, stop, event)~transplant, data=heart)
autoplot(km1, main="Kaplar-Meier estimate grouping by transplant status",
xlab="Days", ylab="Survival")
```




```
km2 = survfit(Surv(start, stop, event)~surgery, data=heart)
autoplot(km2, main="Kaplar-Meier estimate grouping by surgery status",
xlab="Days", ylab="Survival")
```



```
km3 = survfit(Surv(start, stop, event)~transplant+surgery, data=heart)
autoplot(km3, main="Kaplar-Meier estimate grouping by surgery and
transplant", xlab="Days",
         ylab="Survival")
```



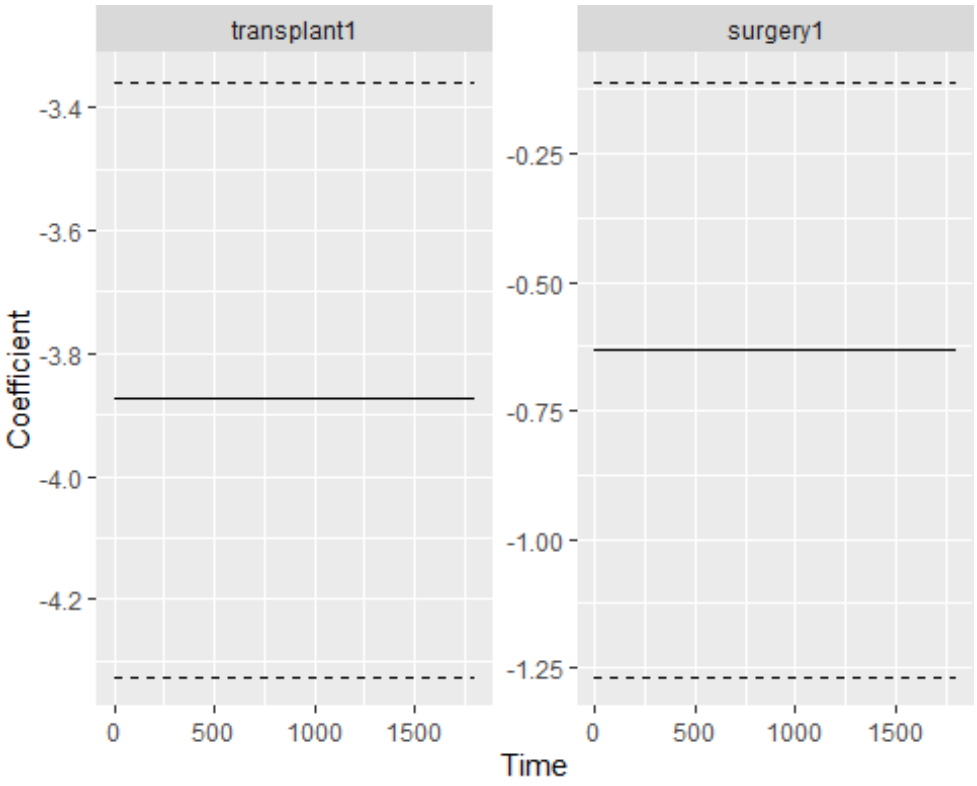
Next, we will fit Cox a regression using a Bayesian approach. The dynsurv package does so by implementing the fit within a Gibbs sampling framework. This package includes options to fit Cox regression models with time-independent, time-varying or dynamic covariate coefficients. As in any Bayesian method, the priors of the Bayesian Cox model have to be specified each time it is fit. More specifically, priors for the baseline hazard and the coefficients have to be specified. Some parameters for the Gibbs sampling algorithm also have to be fixed, such as the number of total iterations, the number of burning iterations or the amount of thinning.

We can start by fitting a Cox regression model with time-independent coefficients, where a Gamam prior is used for the baseline hazard and a Normal prior is used for the coefficients.

```
fit0 = bayesCox(Surv(start, stop, type="interval2")~transplant+surgery,
                data=heart,
                model="TimeIndep",
                base.prior = list(type = "Gamma", shape = 0.1, rate =
0.1),
                coef.prior = list(type = "Normal", mean = 0, sd = 1),
                gibbs = list(iter = 100, burn = 20, thin = 1, verbose =
FALSE))
```

We can visualize the (constant) coefficients with their 95% confidence intervals.

```
plotCoef(coef(fit0, level=0.95))
```

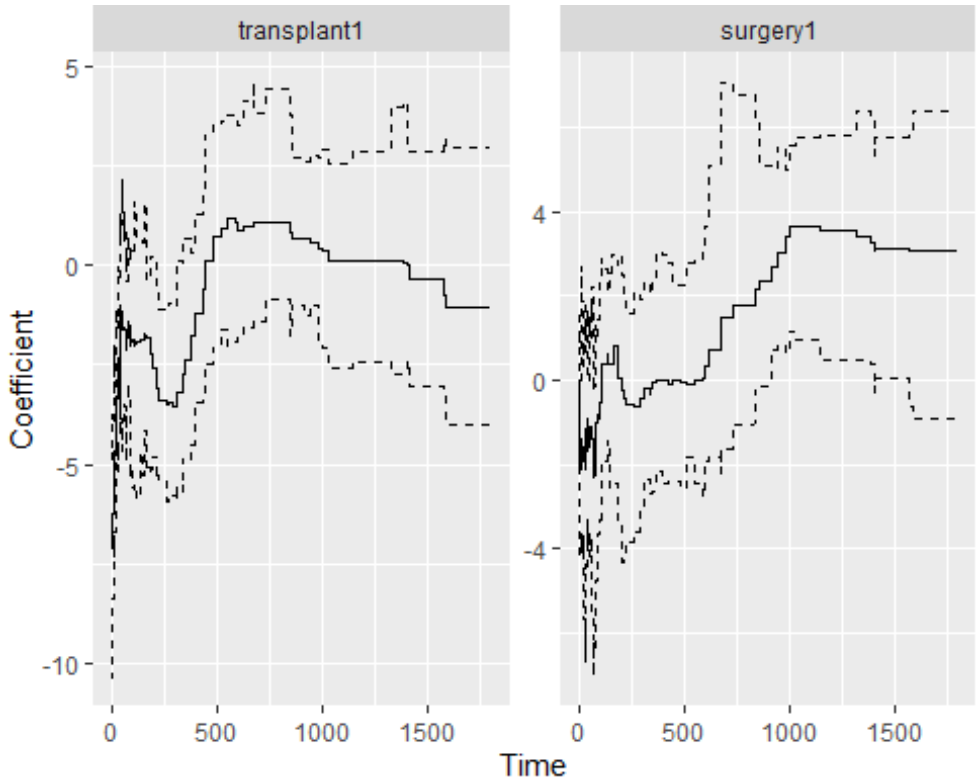


The coefficient corresponding to `transplant` takes a value of -4.087. This indicates that there is a reduction in hazard after having a transplant of about $1-e^{-4.087} \% $. Furthermore, the 95% confidence interval for this variable of $(-4.515, -3.472)$ suggests that it is significant at the 0.05 significance level.

On the other hand, the coefficient for `surgery` is -0.478, indicating a reduction of about $1 - e^{-0.477} \approx 37.9\%$ after undergoing surgery. However, its confidence interval of $(-1.154, 0.276)$ suggest that this variable is not statistically significant at the 0.05 significance level.

Since the surgery and transplant status of a patient can change along the time of observation, it also makes sense to consider time-varying coefficients.

```
fit1 = bayesCox(Surv(start, stop, type="interval2")~transplant+surgery,
                data=heart,
                model="TimeVary",
                base.prior = list(type = "Gamma", shape = 0.1, rate =
0.1),
                coef.prior = list(type = "AR1", sd = 1),
                gibbs = list(iter = 100, burn = 20, thin = 1, verbose =
F))

plotCoef(coef(fit1, level=0.95))
```



We observe quite a different landscape to the one seen in the time-independent Cox regression's coefficients. In any case, the coefficient for `surgery` stays insignificant, whereas the coefficient for `transplant` starts being significant at a value similar to the one obtained earlier, but loses significance as time advances.