Heart Failure - Survival Analysis

Le Tan Dang Khoa

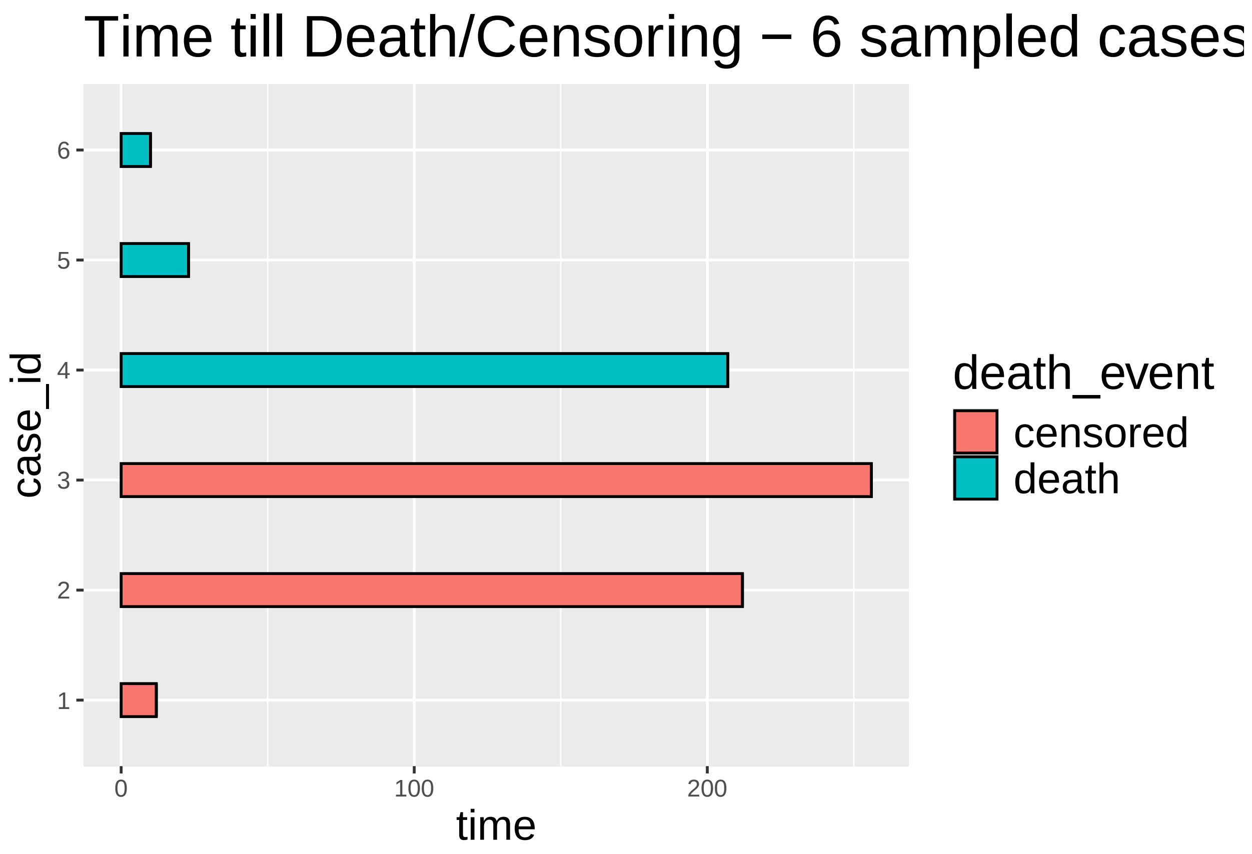
17 July, 2024

Contents

1 Data description

- Age: Age of patient at time of operation
- Operation_year: Patient's year of operation (year - 19XX)
- Nb_pos_detected: Number of positive axillary nodes detected
- Surv: Survival status (class attribute) 1 = the patient survived 5 years or longer 2 = the patient died within 5 year

```
library(tidyverse)
library(survival)
library(survminer)
options(repr.plot.width = 16, repr.plot.height = 7)
# Read data
all_df <- read_csv('heart_failure_clinical_records_dataset.csv', col_type = cols())
plot_censoring <- all_df %>% group_by(DEATH_EVENT) %>% sample_n(3) %>% ungroup() %>% select(time, DEATH_EV
# Sample 3 + 3 points from death/censored groups, and plot them for comparison
plot_censoring <- all_df %>% group_by(DEATH_EVENT) %>% sample_n(3) %>% ungroup() %>% select(time, DEATH_EV
# Data Transformation for Plotting
plot_censoring %>%
    mutate(
        time_start = 0,
        case_id = factor(c(1:nrow(plot_censoring))),
        death_event = factor(ifelse(DEATH_EVENT == 1, "death", "censored"))
    )%>%
# Reshape Data for Plotting
  pivot_longer(
      cols = c(time, time_start),
      names_to = "source",
      values_to = "time"
  ) %>%
ggplot(aes(x = time, y = case_id, group = factor(case_id))) +
    geom_bar(stat = "Identity", aes(fill = death_event), colour = "black", width = 0.3) +
    ggtitle("Time till Death/Censoring - 6 sampled cases from dataset") +
    theme(plot.title = element_text(size = 22),
          legend.title = element_text(size = 18),
          legend.text = element_text(size = 16),
          axis.title = element_text(size = 16))
```



The above plot shows 6 patients, 3 of whom **died** and 3 of whom were **censored** (dropped out of the study before death could be observed).

The fact that some patients were censored does not mean that their data is completely missing. We still know that they **did not die, at least till the time they were censored!**

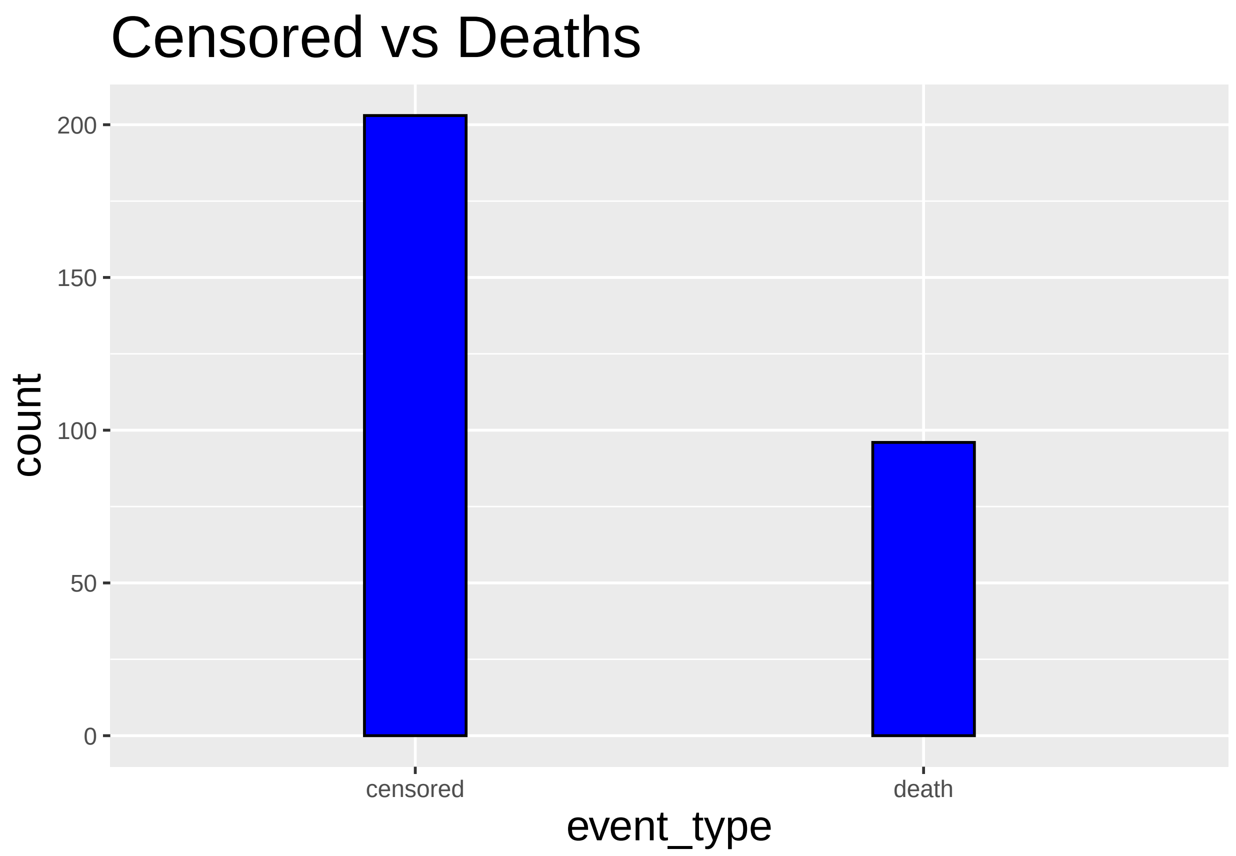
The challenge is to incorporate both types of outcome (actual event vs censoring) as well as the time taken for the observation, in order to come up with a statistical distribution of the time-to-event. This distribution is useful at the population-level, and will almost never be valid at the individual level.

2 Survival Analysis vs Classification

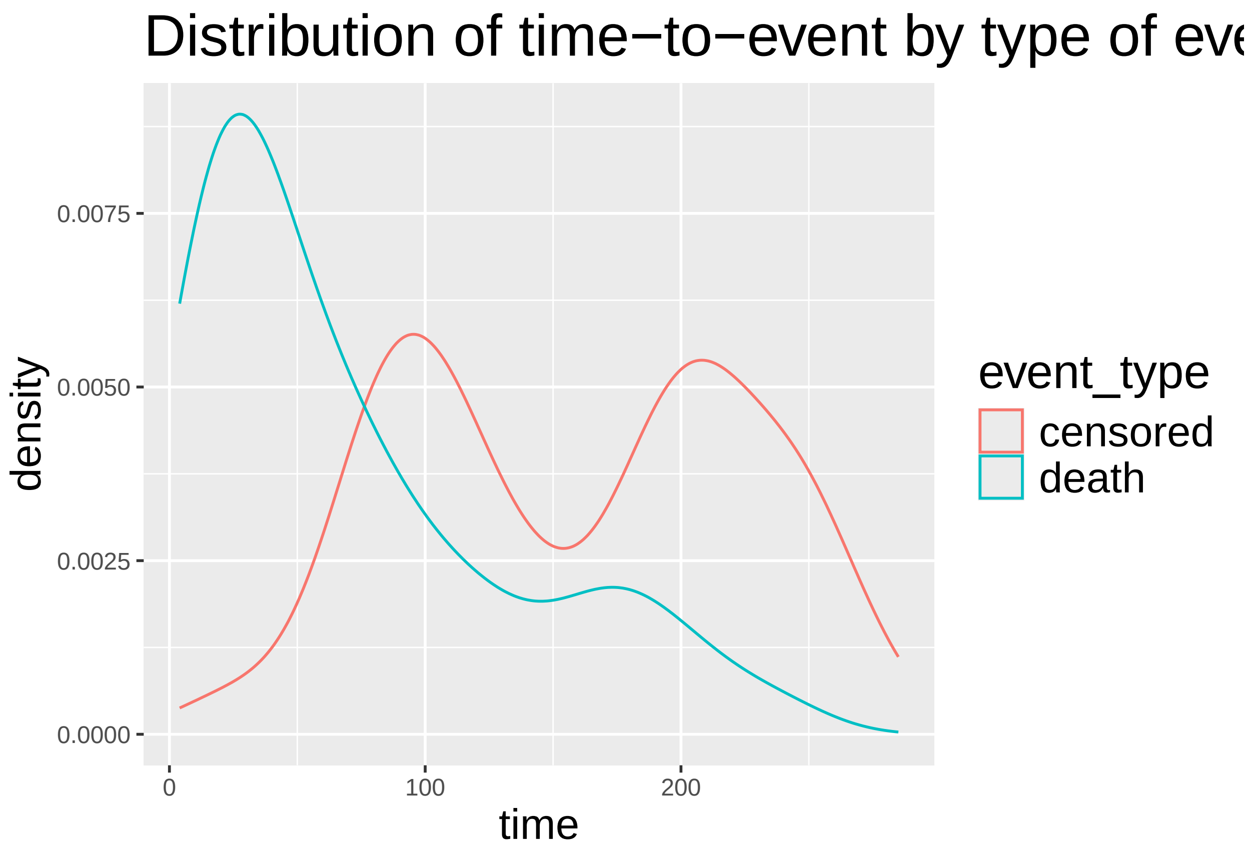
One must be careful to distinguish between survival analysis and classification models. For instance, **a common fallacy is to treat the time variable of a clinical data-set as a feature for a classification model.** If you are asked to predict how long a new cohort of patients will survive, you won't have the time variable since that is unknown!

To get started, let's look at the total number of deaths vs censored cases in our data, as well as the difference in the distribution of time for each event type.

```r
options(repr.plot.width = 10, repr.plot.height = 4)

all_df %>%
    mutate(event_type = factor(ifelse(DEATH_EVENT == 1, "death", "censored"))) %>%
    group_by(event_type) %>%
    tally(name = "count") %>%
    ggplot(aes(x = event_type, y = count)) +
    geom_bar(stat = "Identity", fill = "blue", width = 0.2, colour = "black") +
    ggtitle("Censored vs Deaths") +
    theme(plot.title = element_text(size = 22),
          axis.title = element_text(size = 16))
```



```r
# Distribution of time-to-event by event type
options(repr.plot.width = 12, repr.plot.height = 5)

all_df %>%
    mutate(
        event_type = factor(ifelse(DEATH_EVENT == 1, "death", "censored"))
    ) %>%
    select(event_type, time) %>%
    ggplot(aes(x = time, colour = event_type)) +
    geom_density() +
    ggtitle("Distribution of time-to-event by type of event (Censored vs Deaths)") +
    theme(plot.title = element_text(size = 22),
          axis.title = element_text(size = 16),
          legend.title = element_text(size = 18),
          legend.text = element_text(size = 16))
```



We observe that the time-to-event is distributed very differently for patients who eventually end up dying vs those who end up dropping out of the study.

3 Some Terminology

3.1 The Survival Function

The Survival Function, S(t), is the probability that the time-to-event, depicted by random variable T, is > t.

$S(t) = P(T > t)$

3.2 Hazard Function

The Hazard Function, $\lambda(t)$, is the event rate in the next infinitesimal time duration Δt, given that there was survival till now.

$\lambda(t) = lim_{\Delta t \to 0} \frac{P(t \leqslant T < t + \Delta t)}{\Delta t . S(t)} = -\frac{S'(t)}{S(t))}$

Although the Survival Function is the end goal, it is often easier to model using the Hazard Function.

It's important to note that the Hazard Function is not a Probability. It can be greater than 1.

4 Methodology - Prediction vs Inference

Medical data-sets are often composed of hundreds of individual patients as opposed to tens of thousands of cases typically found in ML problem statements. This makes data splitting hard to do without losing statistical power. Additionally, the end goal is not prediction but rather understanding the effect size of a particular feature.

5 Modelling Approaches

5.1 Kaplan-Meier Estimator

The Kaplan-Meier estimator is a test statistic that gives us an approximation of the true survival function of a population, the approximation getting better with increasing sample size. This estimator can robustly handle censoring, and can be derived from the Hazard Function using Maximum Likelihood Estimation.

This estimator can be used for simple comparison of survival rates between groups (For instance, survival rates between smokers and non-smokers).

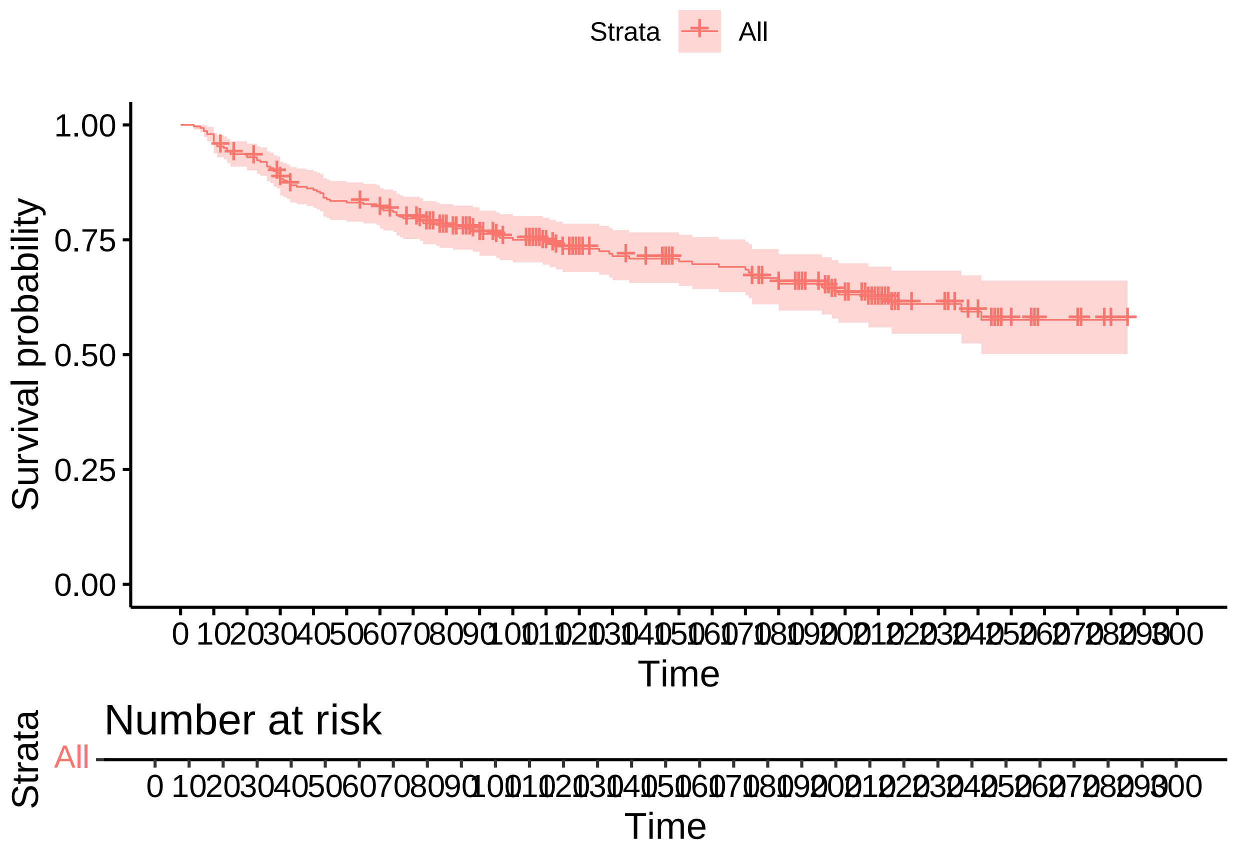
```r
# Kaplan-meier analysis
km_model <- survfit(Surv(time, DEATH_EVENT) ~ 1, data = all_df)
summary(km_model, times = seq(from = 0, to = 290, by = 30))
```

```
## Call: survfit(formula = Surv(time, DEATH_EVENT) ~ 1, data = all_df)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     0    299       0    1.000  0.0000        1.000        1.000
##    30    264      35    0.882  0.0187        0.846        0.920
##    60    239      19    0.817  0.0225        0.774        0.863
##    90    189      15    0.763  0.0250        0.715        0.813
##   120    145       7    0.730  0.0268        0.680        0.785
##   150    118       5    0.703  0.0285        0.649        0.761
##   180    106       8    0.654  0.0313        0.596        0.719
##   210     62       4    0.622  0.0337        0.559        0.692
##   240     34       2    0.594  0.0378        0.524        0.673
##   270      6       1    0.576  0.0407        0.501        0.661
```

The Kaplan-Meier model provides a table of the cumulative survival probability by time. Such "life tables" are commonly used by life insurance companies in planning life insurance products.

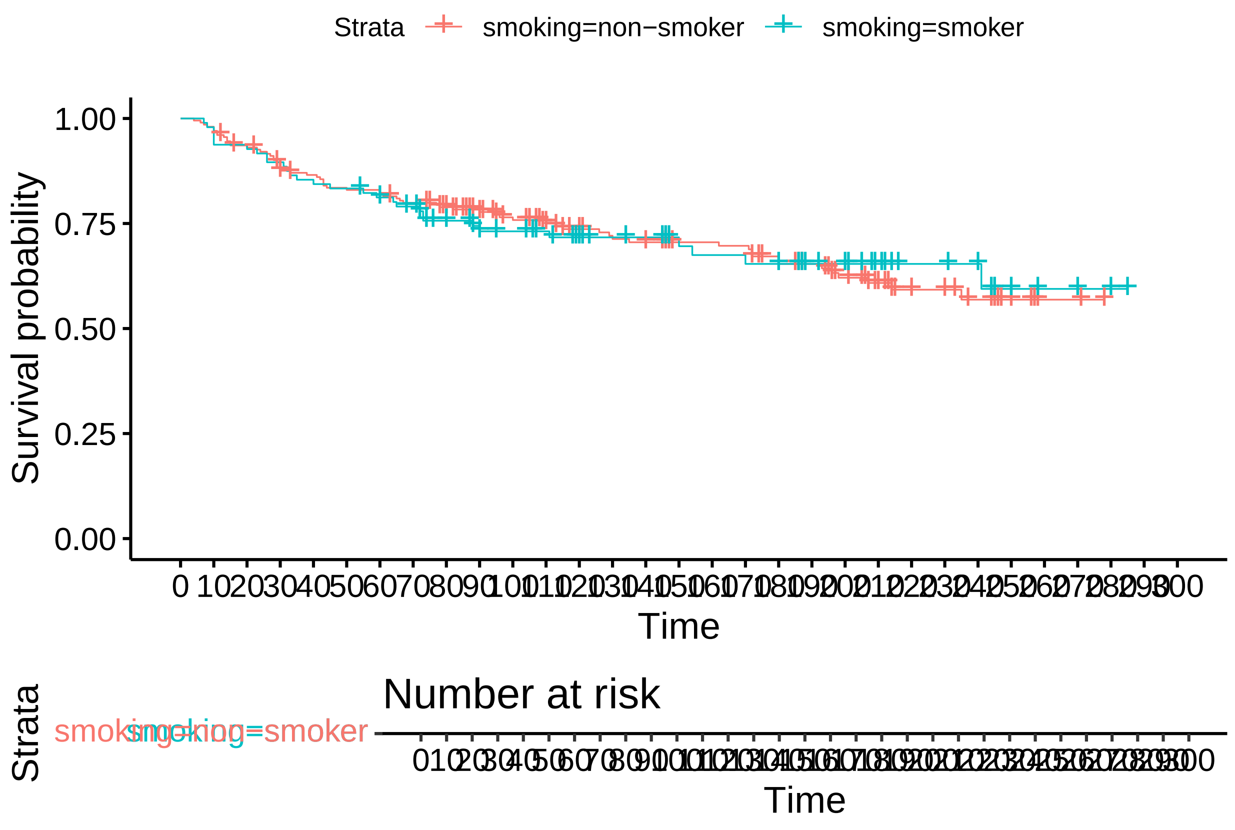
Kaplan-Meier plots can be graphically depicted as well.

```r
# Plot Kaplan-Meier plot
options(repr.plot.width = 18, repr.plot.height = 8)

ggsurvplot(km_model, data = all_df, risk.table = TRUE,
           break.time.by = 10, size = 0.3, tables.height = 0.15)
```



- The Kaplan-Meier plot approaches the true survival curve of the population, as sample size increases.
- On the plot, a censoring event is denoted by the "+" tick marks.
  A Kaplan-Meier plot can also be used to analyze impact of categorical features on survival. For example, is there a decrease in survival for smokers vs non-smokers?

```r
# Kaplan-Meier curve based on presence/absence of smoking
km_model <- all_df %>%
    mutate(
        smoking = factor(ifelse(smoking == 0, "non-smoker", "smoker"))
    ) %>%
    survfit(Surv(time, DEATH_EVENT) ~ smoking, data = .)

ggsurvplot(km_model, data = all_df, risk.table = TRUE,
           break.time.by = 10, size = 0.3, tables.height = 0.20)
```



Surprisingly, the above plot shows us that non-smokers tend to have a higher probability of survival initially but a lower survival for longer time horizons, compared to smokers!

Going back to the life insurance example, insurance companies can use such simple models to design business rules for life insurance products. For a smoker, their insurance premiums will be adjusted upwards to counteract the lowered survival probability. # Cox Proportional Hazards Model

```
# Change columns into factors and scale columns to enable better model fit
all_df <- all_df %>%
    mutate(
        anaemia = factor(ifelse(anaemia == 1, "anaemic", "non-anaemic"), levels = c("non-anaemic", "anaemi
        diabetes = factor(ifelse(diabetes == 1, "diabetic", "non-diabetic"), levels = c("non-diabetic", "d
        high_blood_pressure = factor(ifelse(high_blood_pressure == 1, "high-bp", "non-high-bp"), levels = 
        sex = factor(ifelse(sex == 0, "female", "male"), levels = c("female", "male")),
        smoking = factor(ifelse(smoking == 0, "non-smoker", "smoker"), levels = c("non-smoker", "smoker"))
        platelets = platelets/1e4,
        creatinine_phosphokinase = creatinine_phosphokinase/1e3
    )
```

```
all_df %>% head
```

```
## # A tibble: 6 x 13
##     age anaemia     creatinine_phosphokinase diabetes     ejection_fraction
##   <dbl> <fct>                          <dbl> <fct>                    <dbl>
## 1    75 non-anaemic                    0.582 non-diabetic                20
## 2    55 non-anaemic                    7.86  non-diabetic                38
## 3    65 non-anaemic                    0.146 non-diabetic                20
## 4    50 anaemic                        0.111 non-diabetic                20
## 5    65 anaemic                        0.16  diabetic                    20
## 6    90 anaemic                        0.047 non-diabetic                40
## # i 8 more variables: high_blood_pressure <fct>, platelets <dbl>,
## #   serum_creatinine <dbl>, serum_sodium <dbl>, sex <fct>, smoking <fct>,
## #   time <dbl>, DEATH_EVENT <dbl>
```

```
# Cox proportional hazard model
cox_model <- coxph(Surv(time, DEATH_EVENT) ~ age + anaemia + creatinine_phosphokinase + diabetes + ejectio
                   high_blood_pressure + platelets + smoking + sex,
                   data = all_df)
summary(cox_model)
```

```
## Call:
## coxph(formula = Surv(time, DEATH_EVENT) ~ age + anaemia + creatinine_phosphokinase +
##     diabetes + ejection_fraction + high_blood_pressure + platelets +
##     smoking + sex, data = all_df)
##
##   n= 299, number of events= 96
##
##                                  coef exp(coef)  se(coef)      z Pr(>|z|)
## age                          0.048866  1.050079  0.009154  5.338 9.39e-08 ***
## anaemiaanaemic               0.395116  1.484556  0.210633  1.876   0.0607 .
## creatinine_phosphokinase     0.167006  1.181761  0.100425  1.663   0.0963 .
## diabetesdiabetic             0.070908  1.073483  0.215029  0.330   0.7416
## ejection_fraction           -0.053933  0.947496  0.011173 -4.827 1.39e-06 ***
## high_blood_pressurehigh-bp   0.482632  1.620334  0.214730  2.248   0.0246 *
## platelets                   -0.009633  0.990413  0.011328 -0.850   0.3951
## smokingsmoker                0.051413  1.052757  0.249994  0.206   0.8371
## sexmale                     -0.173410  0.840793  0.250258 -0.693   0.4884
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                             exp(coef) exp(-coef) lower .95 upper .95
## age                            1.0501     0.9523    1.0314    1.0691
## anaemiaanaemic                 1.4846     0.6736    0.9824    2.2433
## creatinine_phosphokinase       1.1818     0.8462    0.9706    1.4388
## diabetesdiabetic               1.0735     0.9315    0.7043    1.6362
## ejection_fraction              0.9475     1.0554    0.9270    0.9685
## high_blood_pressurehigh-bp     1.6203     0.6172    1.0637    2.4682
## platelets                      0.9904     1.0097    0.9687    1.0126
## smokingsmoker                  1.0528     0.9499    0.6450    1.7184
## sexmale                        0.8408     1.1894    0.5148    1.3731
##
## Concordance= 0.706  (se = 0.029 )
## Likelihood ratio test= 59.3  on 9 df,   p=2e-09
## Wald test            = 54.53  on 9 df,   p=1e-08
## Score (logrank) test = 56.35  on 9 df,   p=7e-09
```

Calling summmary on an object in R gives a lot of useful information. For the coxph object, there's quite a bit of an infodump.

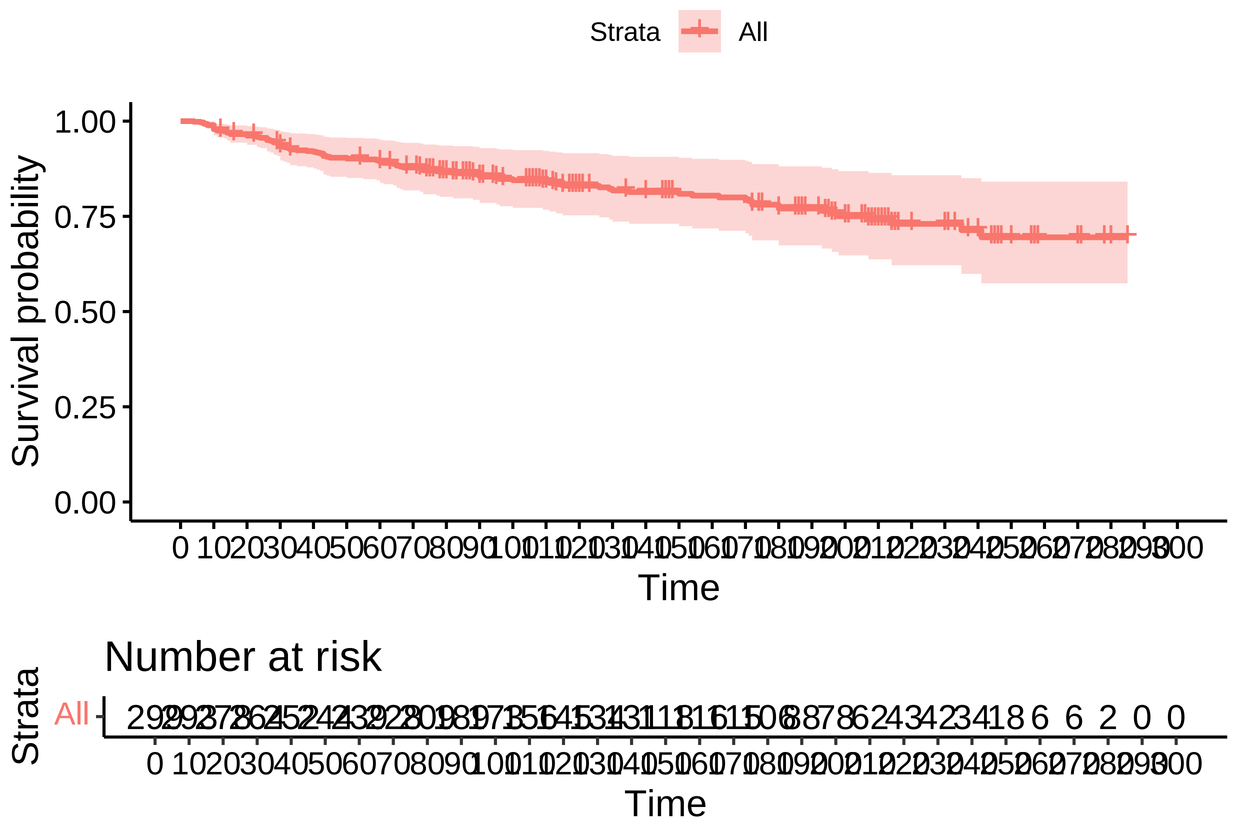
1. The data under "Call" is the exact formula used, and a summary of the data. n = 299 patients were present in the data and 96 of them encountered the event. The remaining were censored.
2. The next table is a summary of the covariate, and the computed coefficients. Since, in the cox proportional hazard model, the covariate is of the form ecoefficient * x, R also provides the exp(coeff) values.
3. The next table provides exp(coeff) and exp(-coeff) values. This is useful because it helps us look at the impact of 1 unit increase or 1 unit decrease in x. For instance, for the covariate creatinine_phosphokinase, increase

by 1 unit (that is, by 1000 since we have scaled the actual values) will result in hazard 18% higher (1 - 1.18). Decrease by 1 unit will result in a hazard 16% lower (1 - 0.84).
4. The model has a Concordance of 0.706. This is similar to an AUC score.

6 Using the Cox Proportional Hazards model

Now that the model has been fit, we can use it to plot the cumulative survival probability of a population.

```r
# Plot the survival for a population with mean value of covariates
ggsurvplot(survfit(cox_model), data = all_df, risk.table = TRUE, break.time.by = 10)
```



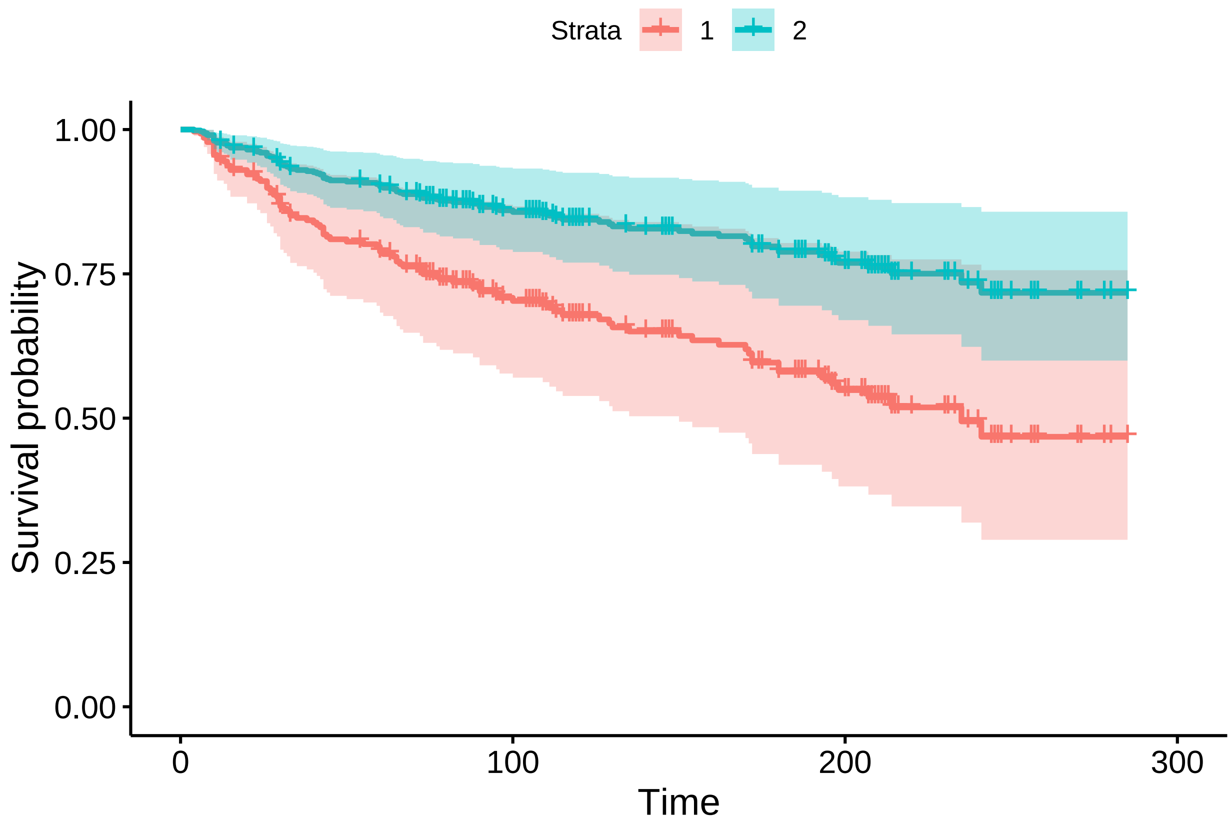
We can compare the survival of 2 groups of populations as well.

```r
# A data-set with 2 rows. 1 row per factor level. Numerical covariates are set to median value.
compare_smoking_median_age <- tibble(
    age = rep(median(all_df$age), 2),
    anaemia = factor(c("anaemic", "non-anaemic"), levels = levels(all_df$anaemia)),
    creatinine_phosphokinase = rep(median(all_df$creatinine_phosphokinase), 2),
    diabetes = factor(c("diabetic", "non-diabetic"), levels = levels(all_df$diabetes)),
    ejection_fraction = rep(median(all_df$ejection_fraction), 2),
    high_blood_pressure = factor(c("high-bp", "non-high-bp"), levels = levels(all_df$high_blood_pressure))
    platelets = rep(median(all_df$platelets), 2),
    smoking = factor(c("smoker", "non-smoker"), levels = levels(all_df$smoking)),
    sex = factor(c("male", "female"), levels = levels(all_df$sex)),
)

compare_smoking_median_age
```

```
## # A tibble: 2 x 9
##     age anaemia     creatinine_phosphokinase diabetes     ejection_fraction
##   <dbl> <fct>                          <dbl> <fct>                    <dbl>
## 1    60 anaemic                         0.25 diabetic                    38
## 2    60 non-anaemic                     0.25 non-diabetic                38
## # i 4 more variables: high_blood_pressure <fct>, platelets <dbl>,
## #   smoking <fct>, sex <fct>
```

```
ggsurvplot(survfit(cox_model, data = compare_smoking_median_age, newdata = compare_smoking_median_age), co
```

```
## Warning: `gather_()` was deprecated in tidyr 1.2.0.
## i Please use `gather()` instead.
## i The deprecated feature was likely used in the survminer package.
##   Please report the issue at <https://github.com/kassambara/survminer/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Things don't look so good for a man who is anaemic, diabetic, has high blood pressure, and is a smoker, compared to a woman who doesn't have any of those conditions/habits. ## Testing the Proportional Hazards assumption As with all models, it's important to test if their assumptions are met.

The big assumption of a proportional hazard model is that each covariate can only push up or push down the baseline hazard proportionally. That is, the coefficient for each covariate is not time-varying.

This can be tested using the scaled schoenfeld residual test

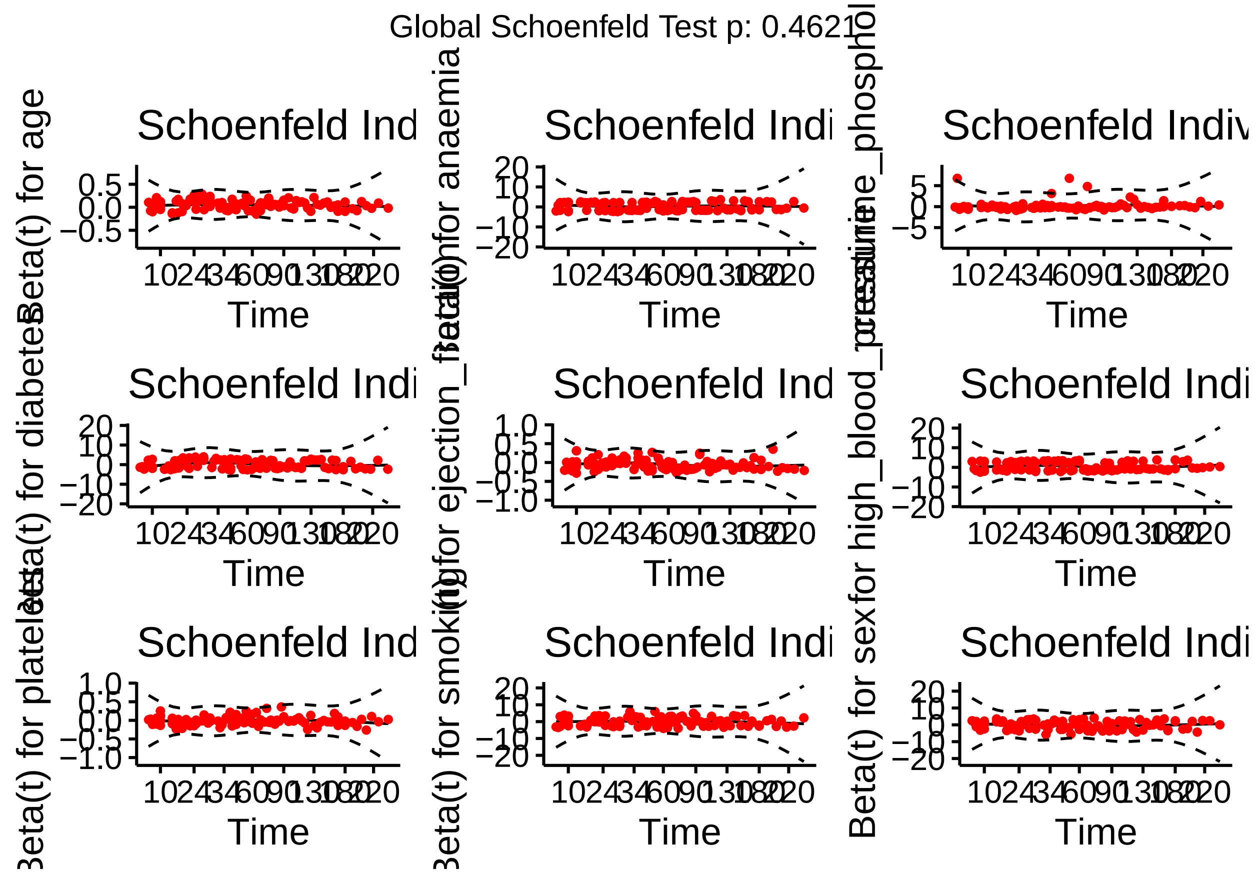
```r
cox.zph(cox_model)
```

```
##                           chisq df     p
## age                      0.7713  1 0.380
## anaemia                  0.2858  1 0.593
## creatinine_phosphokinase 0.7315  1 0.392
## diabetes                 0.1268  1 0.722
## ejection_fraction        5.6415  1 0.018
## high_blood_pressure      0.0895  1 0.765
## platelets                0.1005  1 0.751
## smoking                  0.4923  1 0.483
## sex                      0.3095  1 0.578
## GLOBAL                   8.7353  9 0.462
```

The test reports the chi-square test statistic for all covariates + for the global model. We observe that:

1. The global test is not significant at the 5% level
2. Only the ejection_fraction is significant at the 5% level We should plot the Schoenfeld residuals in order to form a decision:

```r
options(repr.plot.width = 18, repr.plot.height = 12)
ggcoxzph(cox.zph(cox_model))
```



Based on the plots, it doesn't look like any of the covariates have time-varying residuals. The proportional hazard assumption holds.

7 Updated Survival Prediction for Censored Patients

Can we update our model's prediction, given that we know censored patients survived till time T? It turns out we can, using a simple update rule :

$$\begin{aligned} P(T > t \mid T > s) &= \frac{P(T > t \text{ and } T > s)}{P(T > s)} \\ &= \frac{P(T > t)}{P(T > s)} \\ &= \frac{S(t)}{S(s)} \end{aligned}$$

That is, all we need to do is scale the new survival predictions, based on the last known survival predictions.

```
# Predict the new survival function for censored patients

# Given the coxph model, calculate the survival probability of each censored patient at the time they got
censored_patients <- all_df %>%
    filter(DEATH_EVENT == 0) %>%
    mutate(
        last_survival = exp(-1 * predict(cox_model, newdata = ., type = "expected")),
        join_col = 1,
        patient_id = seq(1, nrow(.), 1)
    )

# Since we know that the patients were alive at the time they got censored, use that information to update
# the survival probability for each patient (calculated in the previous code chunk)
censored_patients %<>%
    inner_join(
        tibble(
            time_pred = seq(1, 300, 1),
            join_col = 1
        ),
        by = "join_col"
    ) %>%
    rename(
        censored_time = time,
        time = time_pred) %>%
    mutate(
        original_survival = exp(-1 * predict(cox_model, newdata = ., type = "expected")),
        updated_survival = case_when(
            time <= censored_time ~ 1,
            TRUE ~ original_survival/last_survival
        )
    )
```

```
## Warning in inner_join(., tibble(time_pred = seq(1, 300, 1), join_col = 1), : Detected an unexpected man
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 1 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

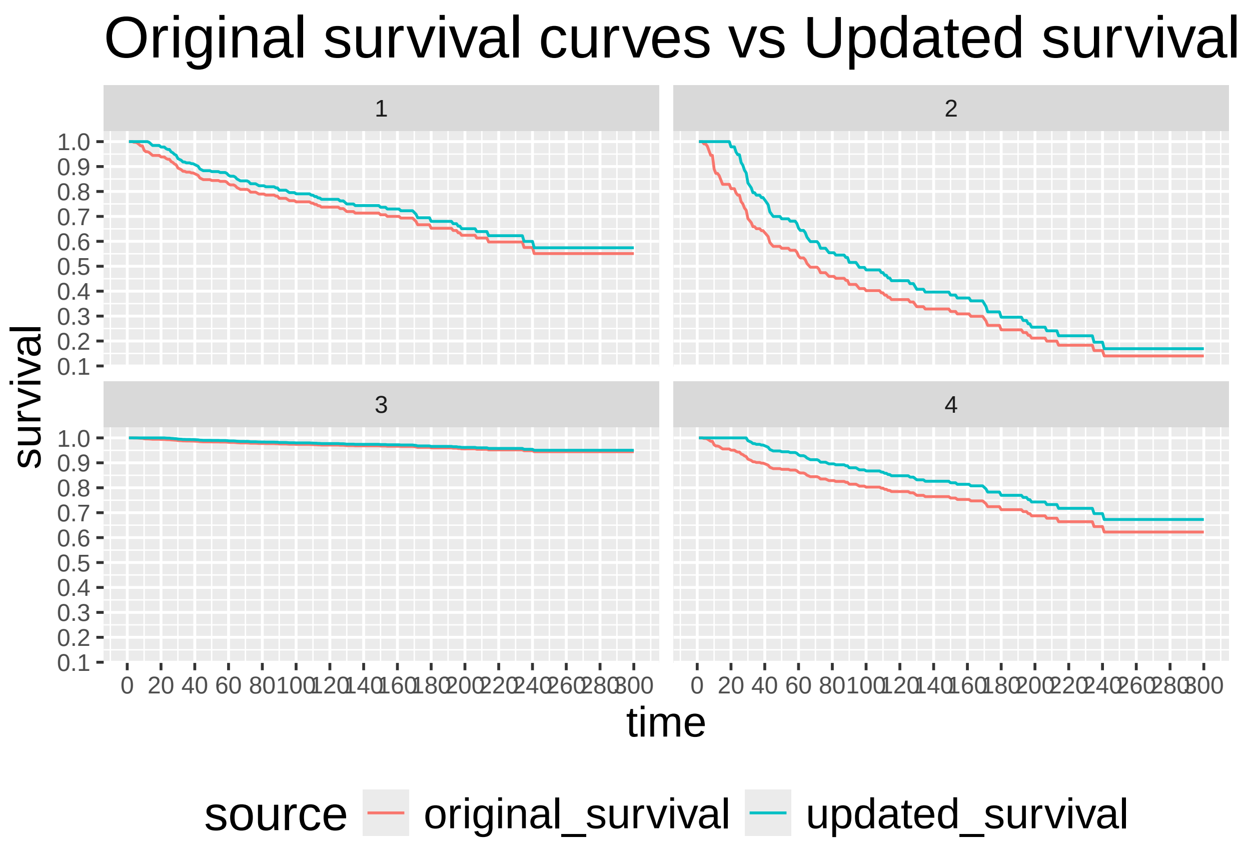
```
censored_patients %>%
    select(patient_id, censored_time, time, original_survival, updated_survival) %>%
    head(15)
```

```
## # A tibble: 15 x 5
##    patient_id censored_time  time original_survival updated_survival
##         <dbl>         <dbl> <dbl>             <dbl>            <dbl>
##  1          1            12     1             1                1    
##  2          1            12     2             1                1    
##  3          1            12     3             1                1    
##  4          1            12     4             0.997            1    
##  5          1            12     5             0.997            1    
##  6          1            12     6             0.994            1    
##  7          1            12     7             0.989            1    
##  8          1            12     8             0.983            1    
##  9          1            12     9             0.983            1    
## 10          1            12    10             0.965            1    
## 11          1            12    11             0.959            1    
## 12          1            12    12             0.959            1    
## 13          1            12    13             0.956            0.997
## 14          1            12    14             0.951            0.991
## 15          1            12    15             0.945            0.984
```

To summarize the above table:

1. For censored patient 1, the original_survival column gives the survival probability by time.
2. Since we know that the patient survived at least till time = 12, we can update our predicted probabilities based on this information
3. For any time <= 12, the updated survival probability will be 1 (since we know they did not die)
4. For time > 12, the updated survival probabilities are computed as per column updated_survival We visualize this updated survival probabilities for the first 4 patients below

```r
censored_patients %>%
    select(patient_id, time, original_survival, updated_survival) %>%
    filter(patient_id < 5) %>%
    pivot_longer(cols = original_survival:updated_survival,
                 names_to = "source",
                 values_to = "survival") %>%
    ggplot(aes(x = time, y = survival, colour = source)) +
    geom_line() +
    facet_wrap(vars(patient_id)) +
    ggtitle("Original survival curves vs Updated survival curves - 4 patients") +
    scale_y_continuous(breaks = seq(0, 1, 0.1)) +
    scale_x_continuous(breaks = seq(0, 300, 20)) +
    theme(plot.title = element_text(size = 22),
          axis.title = element_text(size = 16),
          legend.title = element_text(size = 18),
          legend.text = element_text(size = 16),
          legend.position = "bottom")
```



From the plotted survival curves (as well as from the formula), we can see that new information about the survival of the patient can only offset the survival curve, not change the shape.