Survival Analysis: Hard Drive Reliability Sample *

Cesar Conejo Villalobos *Data Scientist*

The branch of statistics that study the expected duration of time for an event to occur is called survival analysis. The number of events can be one or more. This project reviews nonparametric methods like Kaplan-Meier, Nelson-Aalen, and Cox proportional hazards model. These techniques are applied to the Hard Drive data sets of Backblaze. This application of survival analysis is called *failure-time analysis*. In this way, the goal is to find the survival probabilities of the hard disks using the data collected by Backblaze in 2019. With the raw data, we create new variables for applying survival models. The major package used for this exercise is survival. For the number of files, it also uses data.table package.

Keywords: Survival Analysis, Kaplan-Meier, Nelson-Aalen, Proporcional Hazard models, Cox

Nonparametric models

For this project, we follow the content of S. Klugman (2008) about the estimation for modified data and the most common occurrences in actuarial work. In this case, we need to deal with the following scenario:

1. Truncated data (left truncated): An observation is (left) truncated at d if when it is below d it is not recorded, but when it is above d it is recorded at its observed value.
2. Censored data (right censored): An observation is (right) censored at u if when it is above u it is recorded as being equal to u, but when it is below u it is recorded at its observed value.

In the case of censored data, we can use the Kaplan-Meier product-limit estimator for producing a nonparametric estimate of the survival function $S(t)$. It is defined as:

Kaplan-Meier

$$S_n(t) = \begin{cases} 1 & 0 \leq t < y_1 \\ \prod_{i=1}^{j-1}\left(\frac{r_i - s_i}{r_i}\right) & y_{j-1} \leq t < y_j,\ j = 2, ..., k \\ \prod_{i=1}^{k}\left(\frac{r_i - s_i}{r_i}\right) \text{ or } 0 & t \geq y_k \end{cases}$$

Where:

- $y_1 < y_2 < \cdots < y_k$ the k unique values that appear in the sample.
- s_i: Number of times the uncensored observation y_i appears in the sample.
- r_i: Is the *risk set* at the i-th ordered observation y_i. It comprises the data that is under observation at that age. Include all the fails and censored observations.

*Template taken from (http://github.com/svmiller). **Corresponding author**: svmille@clemson.edu.

Because of the relationship $S(t) = e^{-H(t)}$, the hazard function may be obtained by the inverse transformation of the Kaplan-Meier estimate: $\hat{H}(t) = log(\hat{S}(t))$.

On the other hand, an alternative to the KM estimator is a modification of the Nelson-Aalen estimate of the cumulative hazard rate function

Nelson Aalen

$$\hat{H}(x) = \begin{cases} 0 & 0 \leq t < y_1 \\ \sum_{i=1}^{j-1}(\frac{s_i}{r_i}) & y_{j-1} \leq t < y_j,\ j = 2, ..., k \\ \sum_{i=1}^{k}(\frac{s_i}{r_i}) & t \geq y_k \end{cases}$$

Taking $\hat{S}(t) = e^{-\hat{H}(t)}$. Finally, The Cox proportional hazards (Cox PH) model fits survival data with associated values z to a hazard function of the form:

Proportional hazards models

$$\begin{aligned} h(x|z) &= h_0(x)c(\beta_1 z_1 + \cdots + \beta_p z_p) \\ &= h_0(x)c(\beta^T z) \end{aligned}$$

where

- $c(y)$ is any function that takes positives values. Usually, the exponential function is used $c(y) = e^y$.
- $z = (z_1, ..., z_p)^T$ is a column vector of the z values called *covariates*
- $\beta = (\beta_1, ..., \beta_p)^T$ is a column vector of coefficients.

In this case, our goal is to estimate the value of $h_0(t)$ (called baseline hazard rate function) and the vector of coefficients β. If the estimate of the baseline survival function $S_0(t)$ is provided, then the estimate of the survival function for an individual with covariates z_j may be obtained with the following relationship:

$$\hat{S}(t|z) = \hat{S}_0(t)^{exp(\beta^T z)}$$

Data Preparation

The first that we need to do, is to meet the database of Backblaze. This company recollects a daily file in *csv* that contains the following columns:

- Date: Date of file.
- Serial Number: Assigned serial number of the drive. We use it as ID.
- Model: Assigned model number by the manufacturer.
- Capacity: Drive capacity in bytes.
- Failure: Contains two states: 0, if the drive is ok, 1 if this is the last day the drive was operational before failing.
- Smart Stats: Several columns of data of statistics reported by the drive.

We can see a description from the Smart stats in the wikipedia page. We only show variables considered as crucial for predicting drive failure. Also, we show the smart variable 9 that displays the count of hours in power-on state. This variable lets us calculating the age and study time for the survival models.

```
library("tidyverse")
library("XML")
library("rvest")

smart_parsed <- read_html("https://en.wikipedia.org/wiki/S.M.A.R.T.",
                          encoding = "UTF-8")
tables <- html_table(smart_parsed, fill = TRUE)

# Extract S.M.A.R.T table.
smart_table_code <- tables[[3]]

# Take four columns. Description is extensive
smart_table_code <- smart_table_code[,1:4]

# Change columns names
colnames(smart_table_code) <- c("ID","Attribute","Ideal","Crucial")

# Substract the first three digits as ID
smart_table_code$ID <- str_remove(substr(smart_table_code$ID,
                                         1,
                                         nchar(smart_table_code$ID) - 4),
                                  "^0+")

# Leave only crucial variables and variable Power-On Hours.
detail_code <- smart_table_code[smart_table_code$Crucial != ""
                                | smart_table_code$ID == "9", 1:3]
```

```r
# Delete [:digit:] pattern
detail_code$Attribute <- str_replace_all(detail_code$Attribute,
                                         "[^[A-Za-z()]]",
                                         " ") %>%
                             str_replace_all(.,"[ ]+", " ")


# Show variables
knitr::kable(detail_code, row.names = F, caption = "SMART Variables")
```

Table 1: SMART Variables

ID	Attribute	Ideal
5	Reallocated Sectors Count	Low
9	Power On Hours	
10	Spin Retry Count	Low
184	End to End error IOEDC	Low
187	Reported Uncorrectable Errors	Low
188	Command Timeout	Low
196	Reallocation Event Count	Low
197	Current Pending Sector Count	Low
198	(Offline) Uncorrectable Sector Count	Low
201	Soft Read Error Rate orTA Counter Detected	Low

However, because of the number of `NA` values, we only use the following smart variables:

- smart_9_raw
- smart_5_normalized
- smart_10_normalized
- smart_197_normalized
- smart_198_normalized

Due to the fact we need to group the data by serial_number, we create the following variables:

- First entry: Min(Smart_9_row)

- Last Entry: Max(Smart_9_row)

These variable are significant because we can use it for defining the variables used in the survival model:

- age: Time in days the hard disk has actually been used.

- study_time: Period of time that hard drive is working on during the observation time.

$$\text{Study time} = \frac{\text{Last Entry} - \text{First Entry}}{24}$$

Finally, for the other values, we take the mean of all the normalized observations during 2019.

```r
#Load libraries
library("data.table")

# Read Multiple .csv files. 365 files with daily data of HDD.
# Only choose the columns:
# date
# serial_number
# model
# capacity_bytes
# failure
# smart_9_raw
# smart_5_normalized
# smart_10_normalized
# smart_197_normalized
# smart_198_normalized

file_names <- list.files("data/drive_stats_2019",
                        pattern="*.csv",
                        full.names=TRUE)

data <- rbindlist(lapply(file_names, function(x)
                                    fread(input = x,
                                          header = TRUE,
                                          stringsAsFactors = FALSE,
                                          select = c("date", "serial_number",
                                                     "model", "capacity_bytes",
                                                     "failure", "smart_9_raw",
                                                     "smart_5_normalized",
                                                     "smart_10_normalized",
                                                     "smart_197_normalized",
                                                     "smart_198_normalized")
                                           )
                         )
                  )

#Modify data. Simplify capacity bytes and HDD models
data[, c("capacity_bytes", "model") := list(round(capacity_bytes/10e11),
                                            ifelse(grepl("^ST",model),
                                                   'Seagate',
                                                   str_extract(model, "^[^\\s]+")))]

# Group of data using data table commands

max_hour_smart_9_raw <- as.integer(max(data$smart_9_raw[!is.na(data$smart_9_raw)])
                                   + 1)

data_group <- data[, list(TB = max(capacity_bytes),
```

```
                          count_obs = .N,
                          min_date = min(date),
                          max_date = max(date),
                          min_Hours = min(smart_9_raw),
                          max_Hours = max(smart_9_raw),
                          count_fail = sum(failure),
                          fail       = max(failure),
                          first_date_fail = min(ifelse(failure == 1,
                                                       date,
                                                       "2020-01-01")),
                          first_hour_fail = min(ifelse(failure == 1,
                                                       smart_9_raw,
                                                       max_hour_smart_9_raw)),
                          mean_reallocated   = mean(smart_5_normalized),
                          mean_spin_retry    = mean(smart_10_normalized),
                          mean_current_pend  = mean(smart_197_normalized),
                          mean_uncorrectable = mean(smart_198_normalized)
                          ),
            by =.(serial_number, model)]

# Creation variables for survival models
# age: Count of hours of first power on measure in days
# study_time: Count of days between the first measure and the last measure
#              or measure of fail

data_group <- data_group %>%
  mutate(age = floor(min_Hours/24),
         study_time = ifelse(fail == 1,
                             floor((first_hour_fail - min_Hours)/24),
                             floor((max_Hours - min_Hours)/24)) + 1
  )

# save aggregated data
write.csv(x =  data_group,
          file = "output/data/data_group_2019.csv")
```

Exploratory Analysis

```r
# Load aggregate data
data_group <- fread(input = "output/data/data_group_2019.csv",
                    header = TRUE,
                    stringsAsFactors = FALSE,
                    select = c("serial_number",
                               "model",
                               "TB",
                               "fail",
                               "first_date_fail",
                               "mean_reallocated",
                               "mean_spin_retry",
                               "mean_current_pend",
                               "mean_uncorrectable",
                               "age",
                               "study_time")
                    )
```

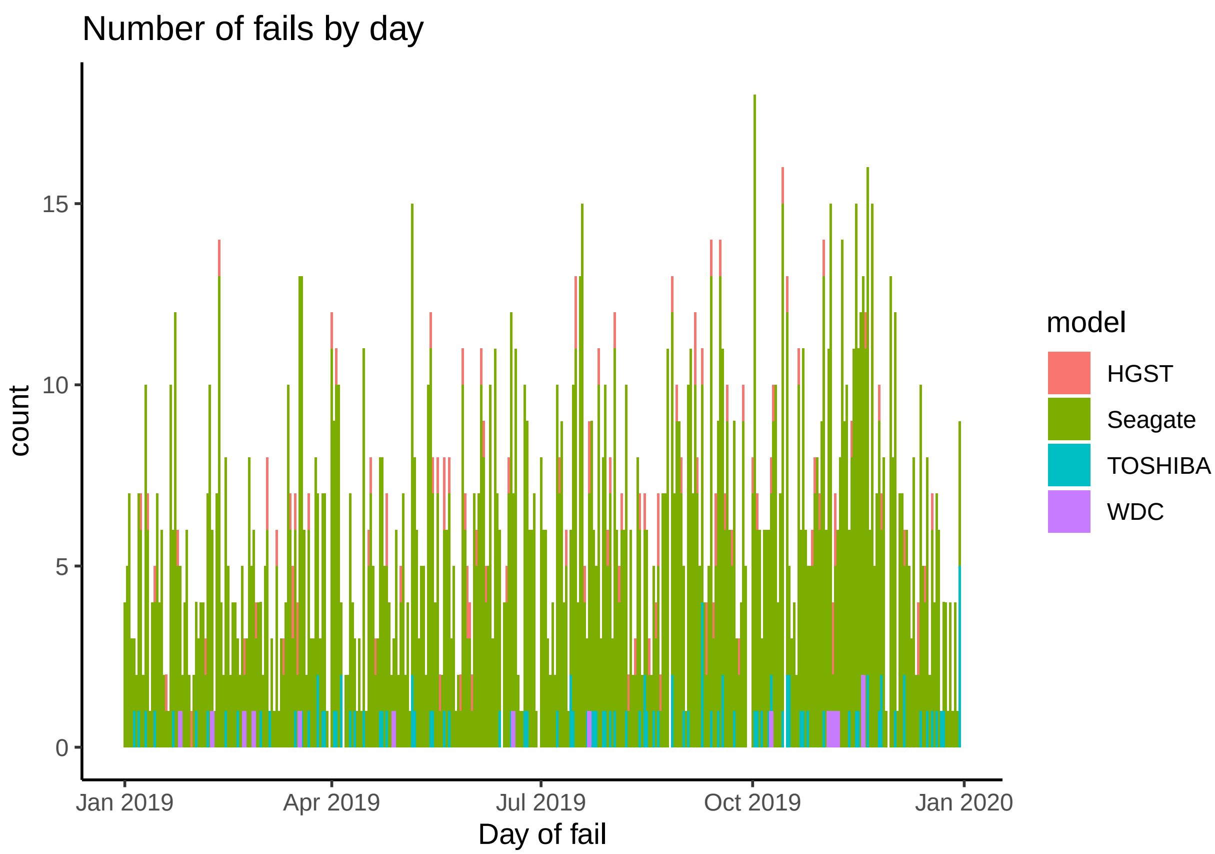
```r
# Modify data
data_group <-  data_group %>%
               mutate(model           = as.factor(model),
                      serial_number   = as.factor(serial_number),
                      first_date_fail = as.Date(first_date_fail)
                     ) %>%
               filter(age >= 0,
                     mean_reallocated >= 0) %>%
               droplevels()

# Create Variable: Median Age:
ageMed <- summary(data_group$age)["Median"]

data_group$age_Med <- data_group$age - ageMed
```

```r
# Distribution data
summary(data_group)
```

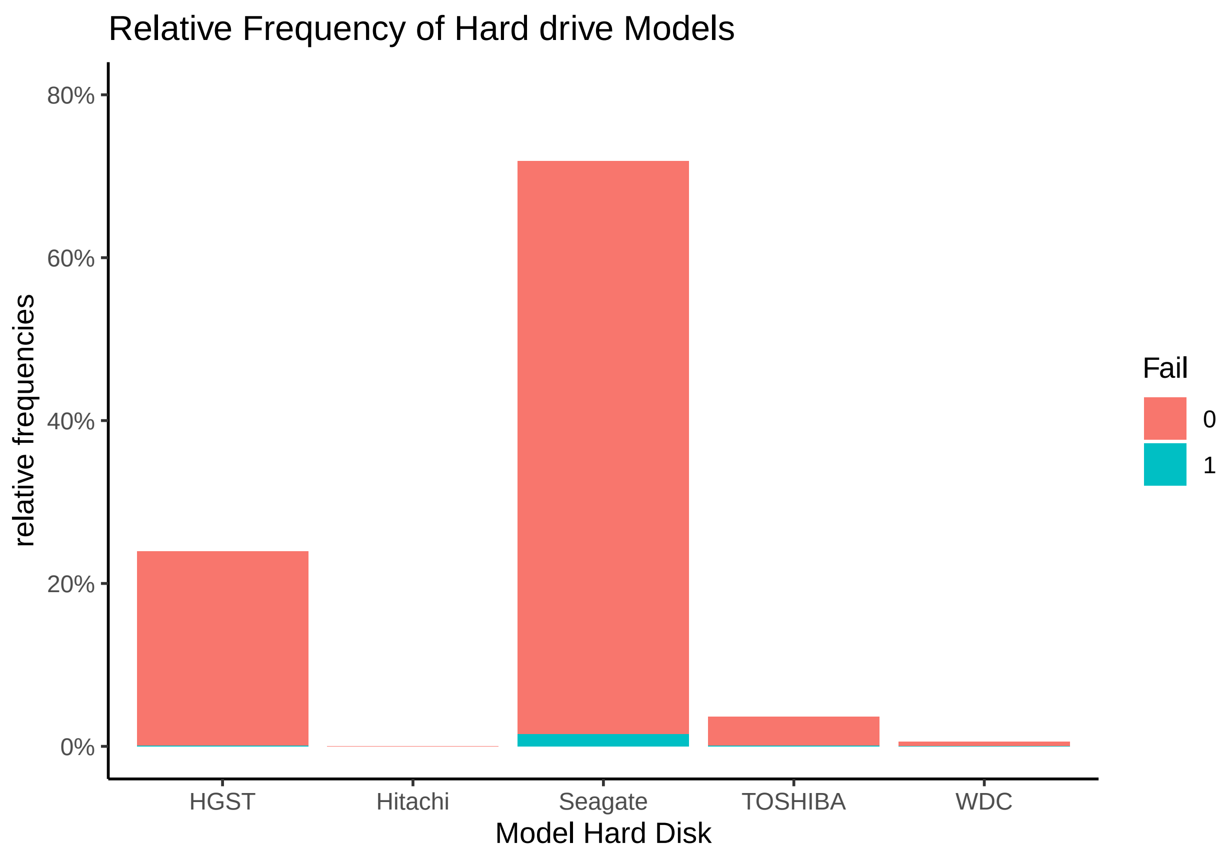
```
##    serial_number        model              TB              fail      
##  175PP3HDT:     1   HGST   :31459   Min.   : 0.0   Min.   :0.00  
##  175PP3I4T:     1   Hitachi:   16   1st Qu.: 4.0   1st Qu.:0.00  
##  175PP3I5T:     1   Seagate:94474   Median : 8.0   Median :0.00  
##  175PP3I6T:     1   TOSHIBA: 4755   Mean   : 8.4   Mean   :0.02  
##  175PP3I8T:     1   WDC    :  744   3rd Qu.:12.0   3rd Qu.:0.00  
##  175PP3I9T:     1                   Max.   :16.0   Max.   :1.00  
##  (Other)  :131442                                                
##  first_date_fail      mean_reallocated mean_spin_retry mean_current_pend
##  Min.   :2019-01-01   Min.   : 31      Min.   : 75     Min.   : 87      
##  1st Qu.:2020-01-01   1st Qu.:100      1st Qu.:100     1st Qu.:100      
##  Median :2020-01-01   Median :100      Median :100     Median :100      
##  Mean   :2019-12-29   Mean   :101      Mean   :101     Mean   :101      
##  3rd Qu.:2020-01-01   3rd Qu.:100      3rd Qu.:100     3rd Qu.:100      
##  Max.   :2020-01-01   Max.   :252      Max.   :252     Max.   :252      
##                                                                         
##  mean_uncorrectable      age          study_time     age_Med      
##  Min.   : 87        Min.   :   0   Min.   :  1   Min.   :-497  
##  1st Qu.:100        1st Qu.:  51   1st Qu.:301   1st Qu.:-446  
##  Median :100        Median : 497   Median :364   Median :   0  
##  Mean   :101        Mean   : 583   Mean   :297   Mean   :  86  
##  3rd Qu.:100        3rd Qu.: 984   3rd Qu.:364   3rd Qu.: 487  
##  Max.   :252        Max.   :2936   Max.   :579   Max.   :2439  
## 
```

```r
# Detail age/study_time by fail

 ggplot(data = data_group) +
  geom_point(aes(x = age, y = study_time,
                 shape = as.factor(fail), color = as.factor(fail) )) +
  labs(title = "Distribution Study time and fails",
       fill =  "Fail",
       x = "Age",
       y = "Time")
```

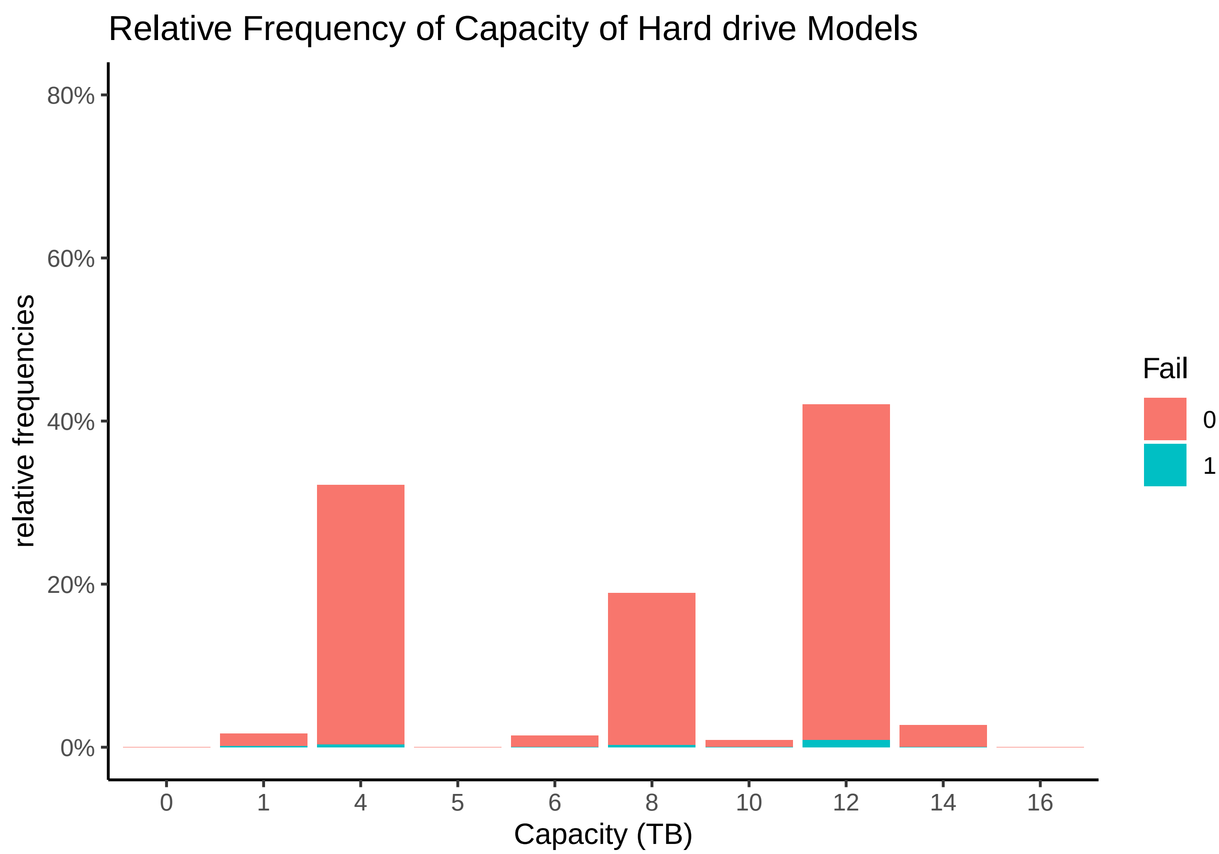

```r
# Number of fails by day

fails <- data_group %>% filter(fail == 1)

ggplot(data = fails, aes(x = first_date_fail, fill = model)) +
  geom_bar() +
  labs(title = "Number of fails by day") +
  xlab("Day of fail") +
  theme_classic()
```



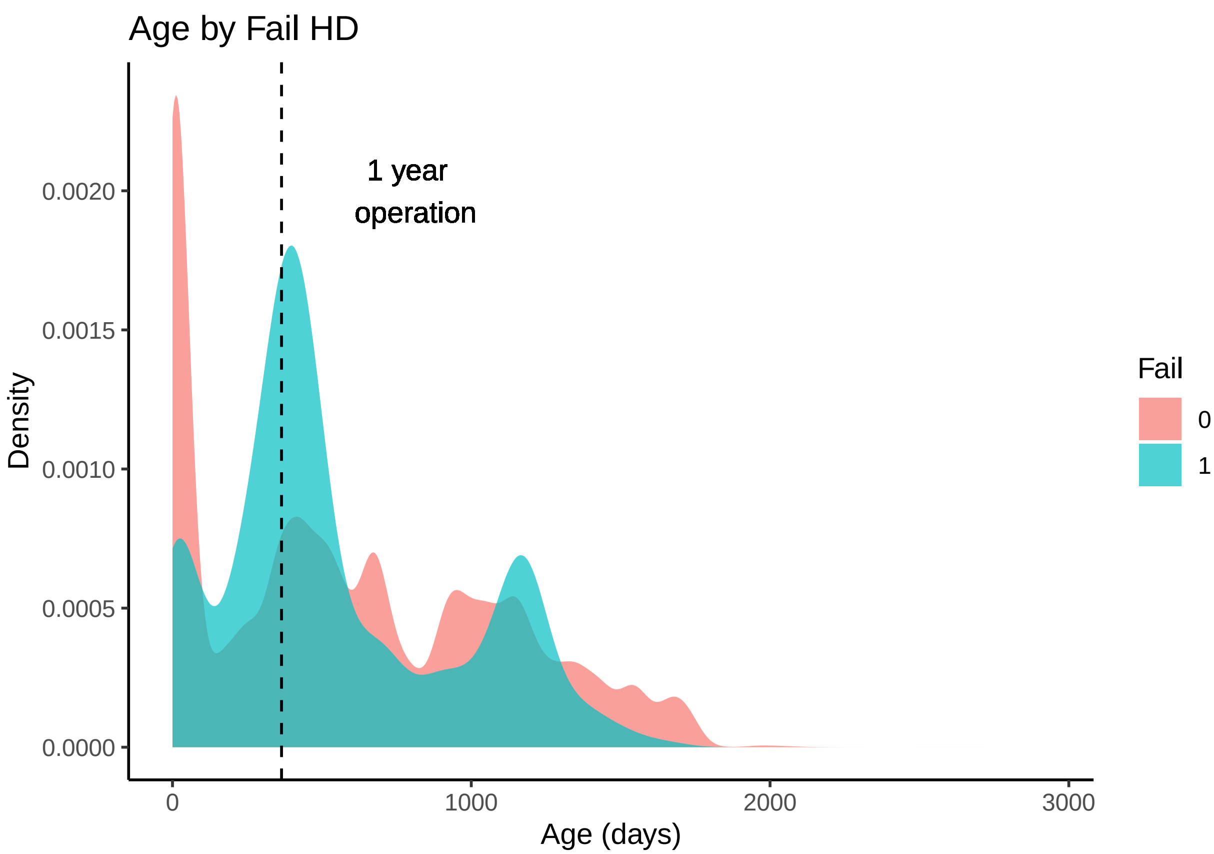

```
# Models of Hard disk and fails

ggplot(data = data_group, aes(x = model)) +
  geom_bar(aes(y = (..count..)/sum(..count..), fill = as.factor(fail))) +
  scale_y_continuous(labels=scales::percent, limits = c(0,0.8)) +
  ylab("relative frequencies") +
  labs(title = "Relative Frequency of Hard drive Models",
       fill =  "Fail",
       x = "Model Hard Disk") +
  theme_classic()
```



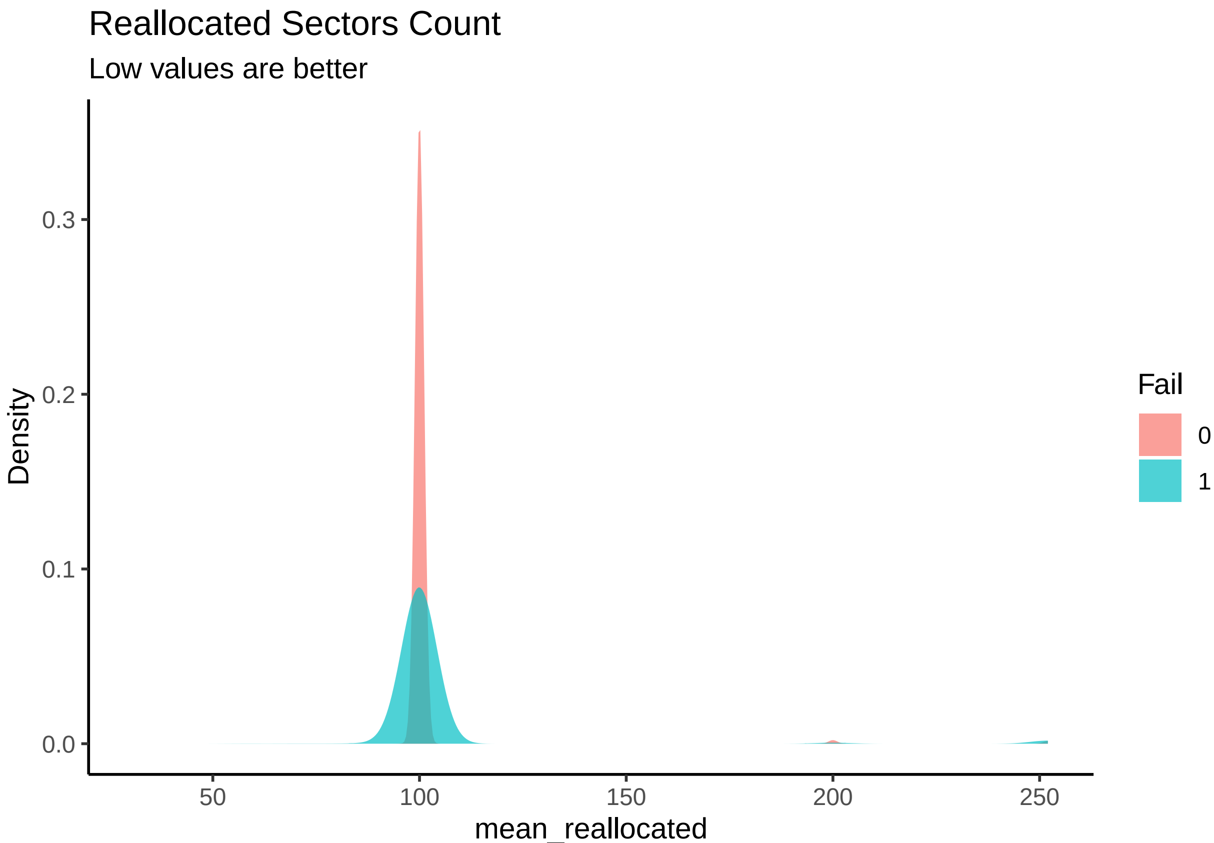

```r
# Capacity of Hard disk and fails

ggplot(data = data_group, aes(x = as.factor(TB))) +
  geom_bar(aes(y = (..count..)/sum(..count..), fill = as.factor(fail))) +
  scale_y_continuous(labels=scales::percent, limits = c(0,0.8)) +
  ylab("relative frequencies") +
  labs(title = "Relative Frequency of Capacity of Hard drive Models",
       fill =  "Fail",
       x = "Capacity (TB)") +
  theme_classic()
```

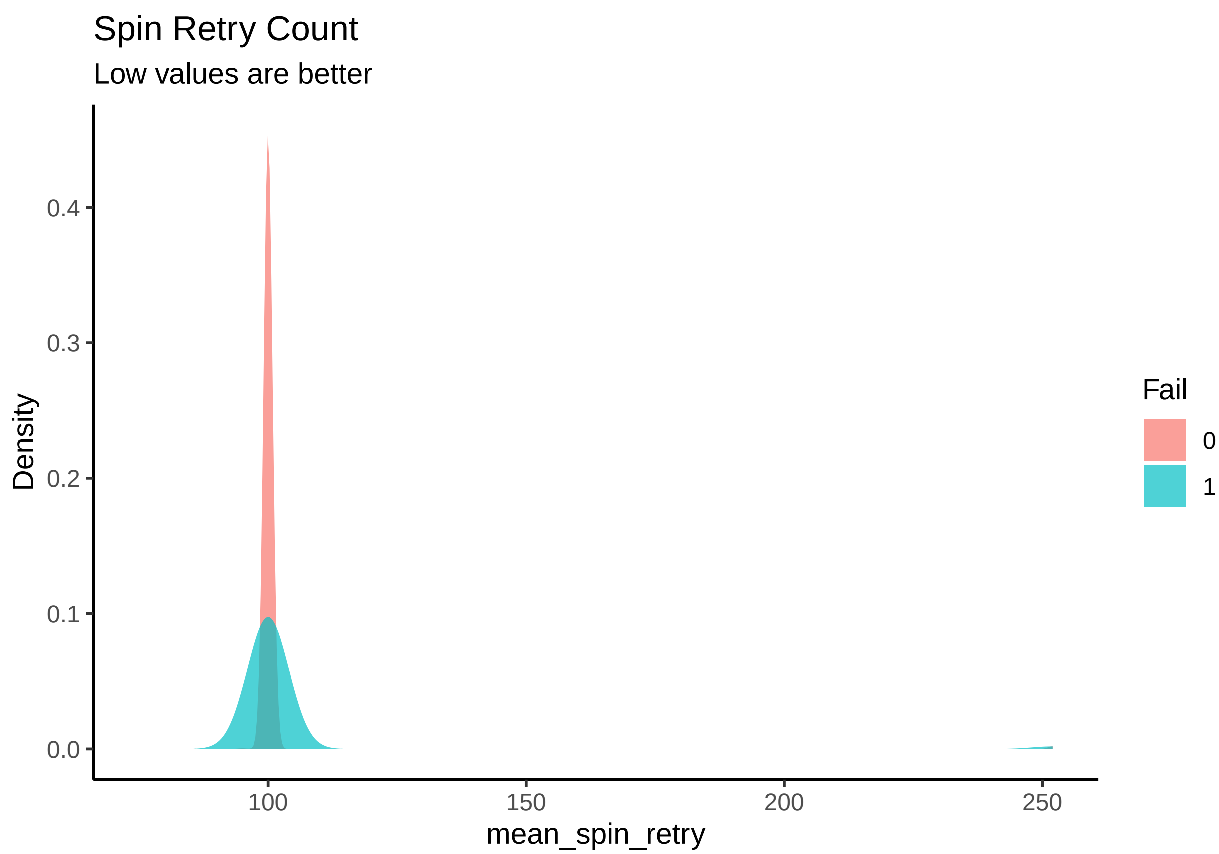



```
# Distribution of Age and fails

ggplot(data = data_group, (aes(x = age, fill = as.factor(fail)))) +
  geom_density(alpha = 0.7, color = NA) +
  labs(
    y = "Density",
    x = "Age (days)",
    fill = "Fail",
    title = "Age by Fail HD"  ) +
  theme_classic() +
  geom_vline(xintercept = 365, linetype = "dashed") +
  geom_text(x = 800, y = 0.002, label= "1 year \n operation")
```




```
# mean_reallocated

  ggplot(data = data_group, (aes(x = mean_reallocated, fill = as.factor(fail)))) +
  geom_density(alpha = 0.7, color = NA) +
  labs(
    y = "Density",
    fill = "Fail",
    title = "Reallocated Sectors Count",
    subtitle = "Low values are better"
  ) +
    theme_classic()
```


Reallocated Sectors Count
Low values are better
Density
0.3
0.2
0.1
0.0
50
100
150
200
250
mean_reallocated
Fail
0
1



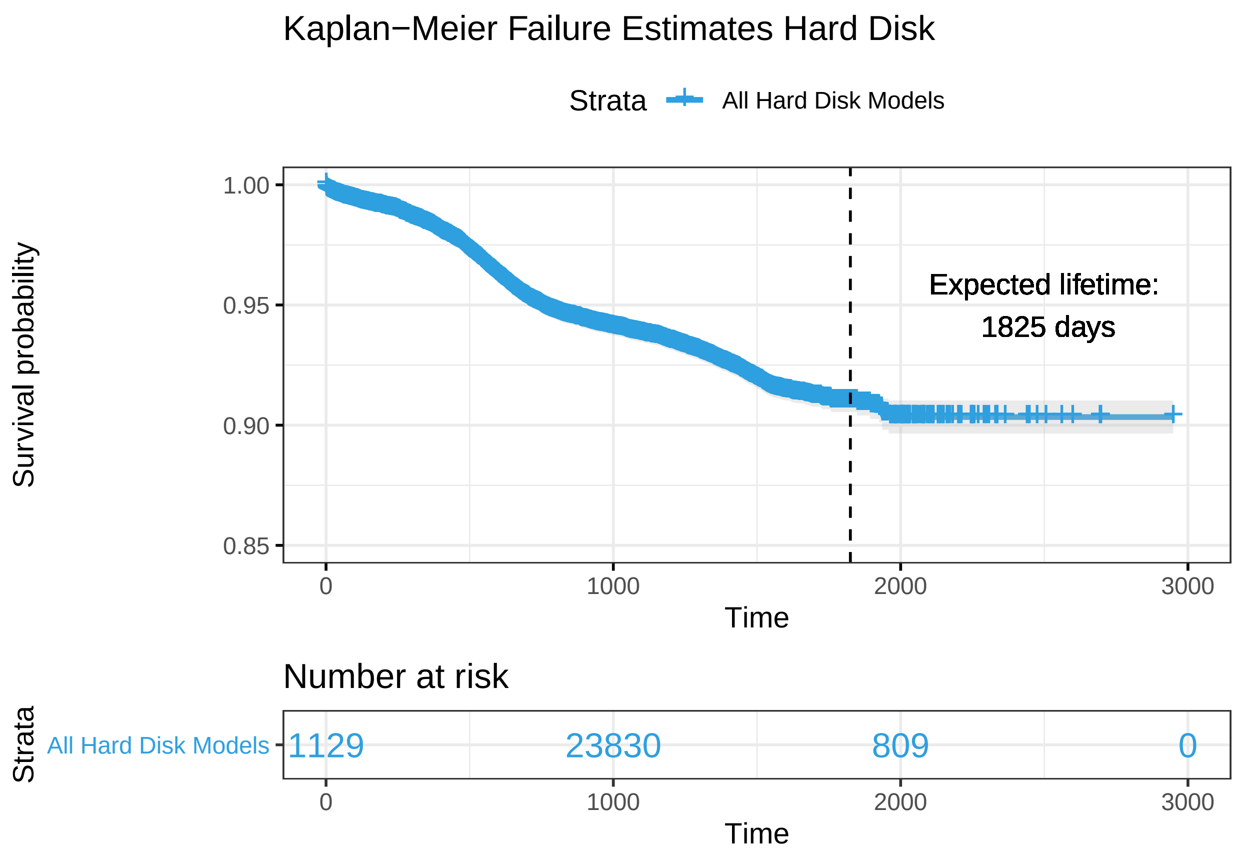
Survival models

```
# Load libraries
library("survival")
library("survminer")
library("KMsurv")

# attach data
attach(data_group)

# 1) Kaplan-Meier Global probababilities

# 1.1) Survival function
surv_object_HDD <- Surv(age, age + study_time, fail)

km_survival_HDD <- survfit(surv_object_HDD ~ 1)

# Global option / Description of results
print(km_survival_HDD)
```

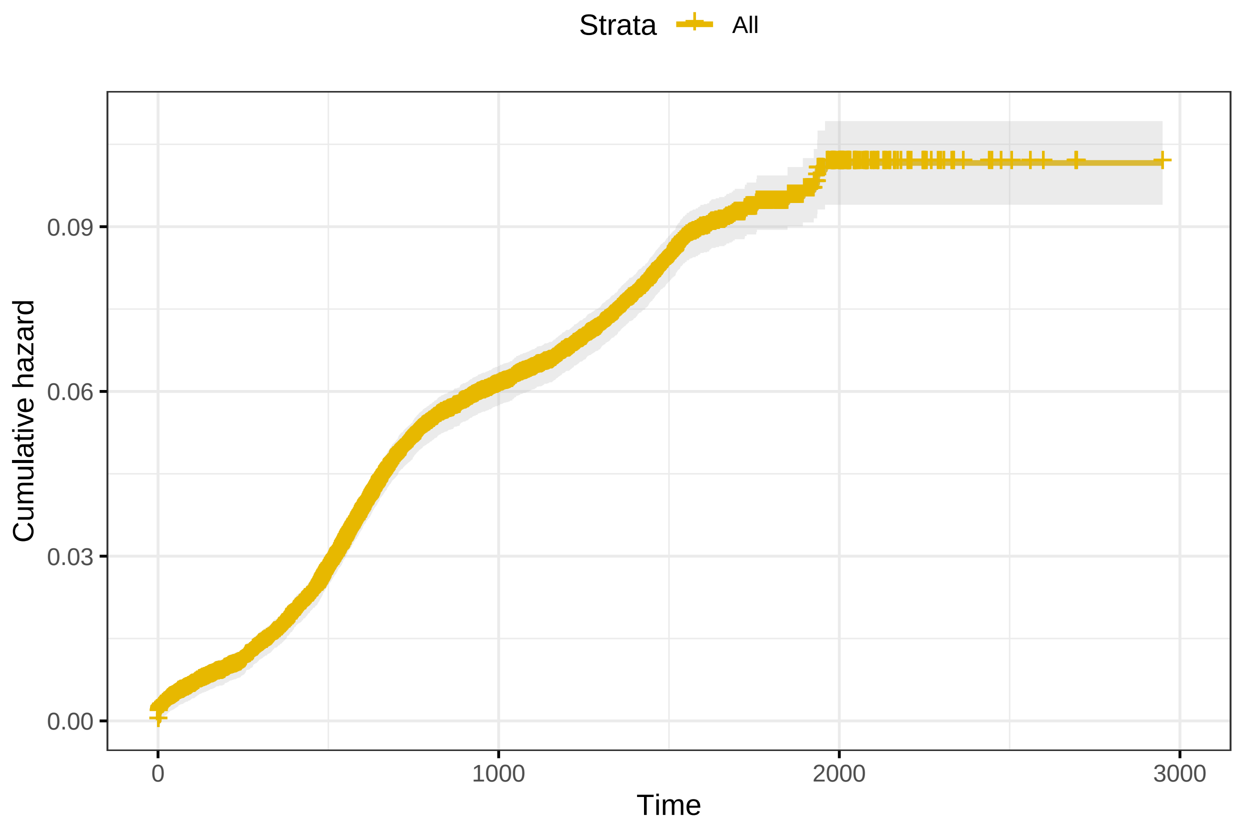
```
## Call: survfit(formula = surv_object_HDD ~ 1)
##
```

```
## records   n.max n.start  events  median 0.95LCL 0.95UCL
##  131448   36656    1129    2211      NA      NA      NA
```

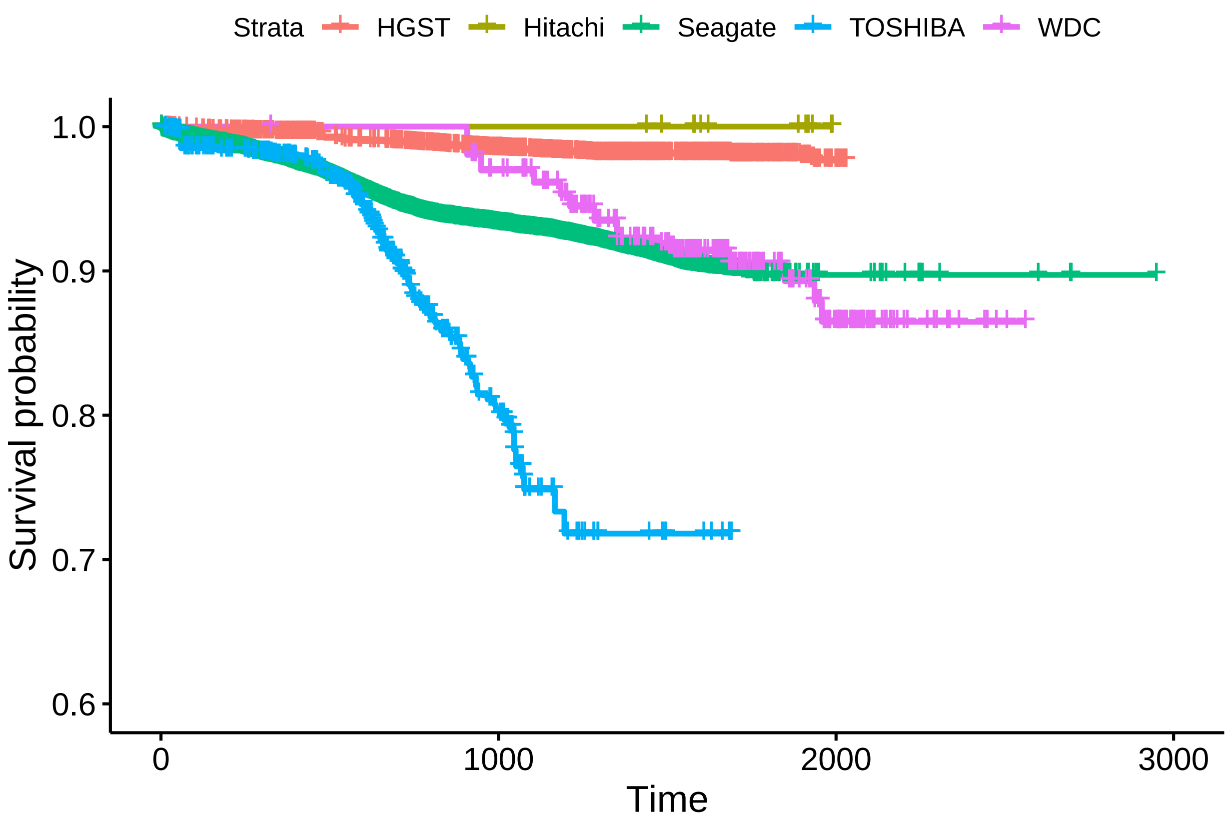
```
ggsurv <- ggsurvplot(
  km_survival_HDD,
  data       = data_group,
  ylim       = c(0.85,1),
  palette    = "#2E9FDF",            # custom color palettes
  conf.int   = TRUE,                 # Add confidence interval
  risk.table = TRUE,                 # Add risk table
  risk.table.col    = "strata",      # Risk table color by groups
  legend.lab        = "All Hard Disk Models", # Change legend labels
  risk.table.height = 0.25,          # Useful to change when you have multiple groups
  ggtheme           = theme_bw(),     # Change ggplot2 theme
  title             = "Kaplan-Meier Failure Estimates Hard Disk"
)

# Drawing a vertical line at Expected lifetime

ggsurv$plot <- ggsurv$plot +
geom_vline(xintercept = 1825, linetype = "dashed") +
  geom_text(x = 2500, y = 0.95, label="Expected lifetime:\n 1825 days")

print(ggsurv)
```




```
# 1.2) cumulative hazard

ggsurvplot(km_survival_HDD,
           data      = data_group,
           conf.int = TRUE,
           ggtheme  = theme_bw(),        # Change ggplot2 theme
           palette  = "#E7B800",
           fun      = "cumhaz")
```



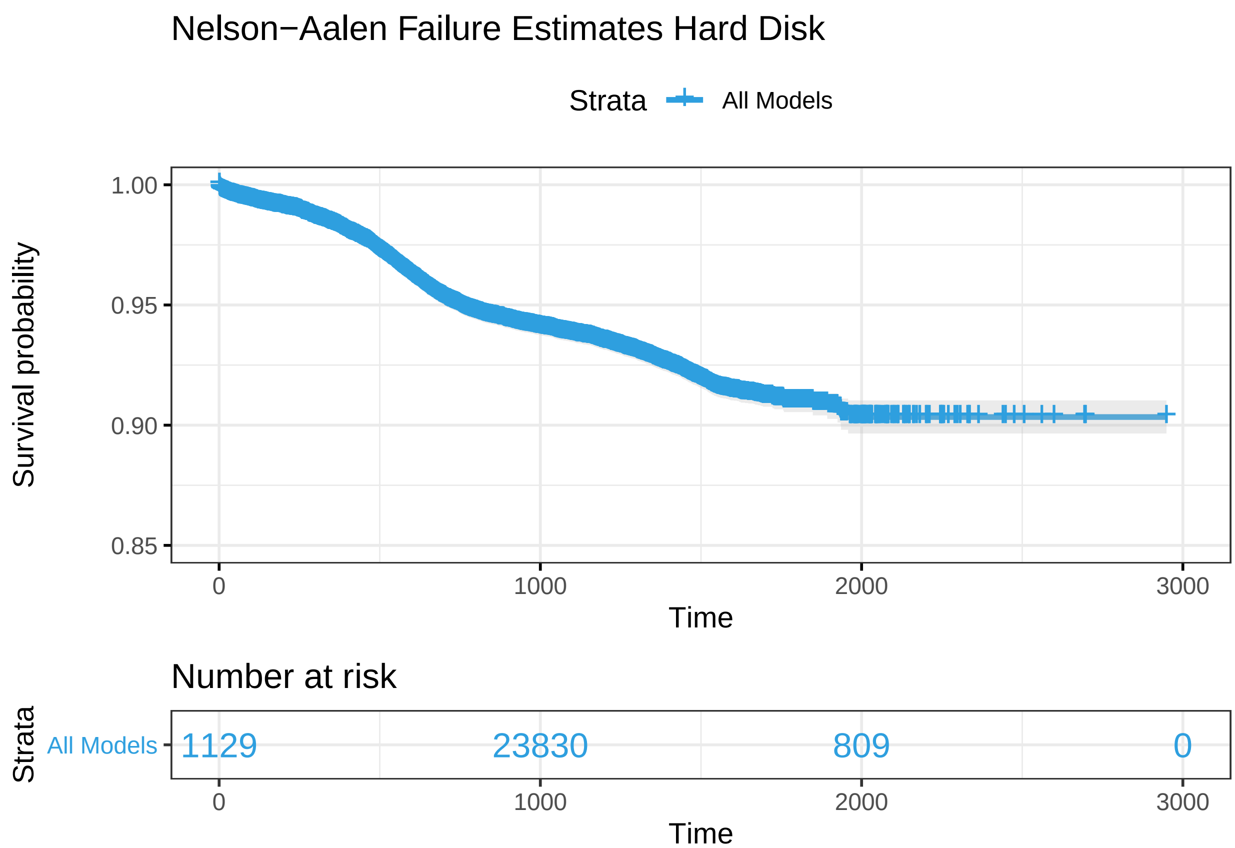

```r
# 2) Kaplan-Meier non-parametric analysis by model
km_survival_model <- survfit(surv_object_HDD ~ model)

ggsurvplot(km_survival_model,
           data = data_group,
           ylim = c(0.6, 1),
           legend.lab = c("HGST", "Hitachi", "Seagate","TOSHIBA", "WDC")
           )
```




```r
# 3) Nelson-Aalen non-parametric analysis
na_survival_HDD <- survfit(coxph(surv_object_HDD ~ 1), type = "aalen")
print(na_survival_HDD)
```

```
## Call: survfit(formula = coxph(surv_object_HDD ~ 1), type = "aalen")
##
## records   n.max n.start  events  median 0.95LCL 0.95UCL
##  131448   36656    1129    2211      NA      NA      NA
```

```r
ggsurvplot(
  na_survival_HDD,
  data        = data_group,
  ylim        = c(0.85,1),
  size        = 1,                  # change line size
  palette     = "#2E9FDF",          # custom color palettes
  conf.int    = TRUE,               # Add confidence interval
  risk.table = TRUE,                # Add risk table
  risk.table.col     = "strata",    # Risk table color by groups
  legend.lab         = "All Models", # Change legend labels
  risk.table.height = 0.25,         # Useful to change when you have multiple groups
  ggtheme            = theme_bw(),   # Change ggplot2 theme
  title              = "Nelson-Aalen Failure Estimates Hard Disk"
  )
```




```r
# 4) PH COX Models

# 4.1) Univariate Cox regression

# 4.1.1) Validation and Diagnostics PH assumptions

res_cox_Age <- coxph(surv_object_HDD ~ age_Med , data = data_group)

test_cox_Age <- cox.zph(res_cox_Age)

test_cox_Age
```

```
##          chisq df    p
## age_Med  1.48  1 0.22
## GLOBAL   1.48  1 0.22
```

```r
# Conclusion: High p-value
# Test is not statistically significant for the age Med covariate.
# we can assume the proportional hazards.

# 4.1.2) Summary of results:

summary(res_cox_Age)
```

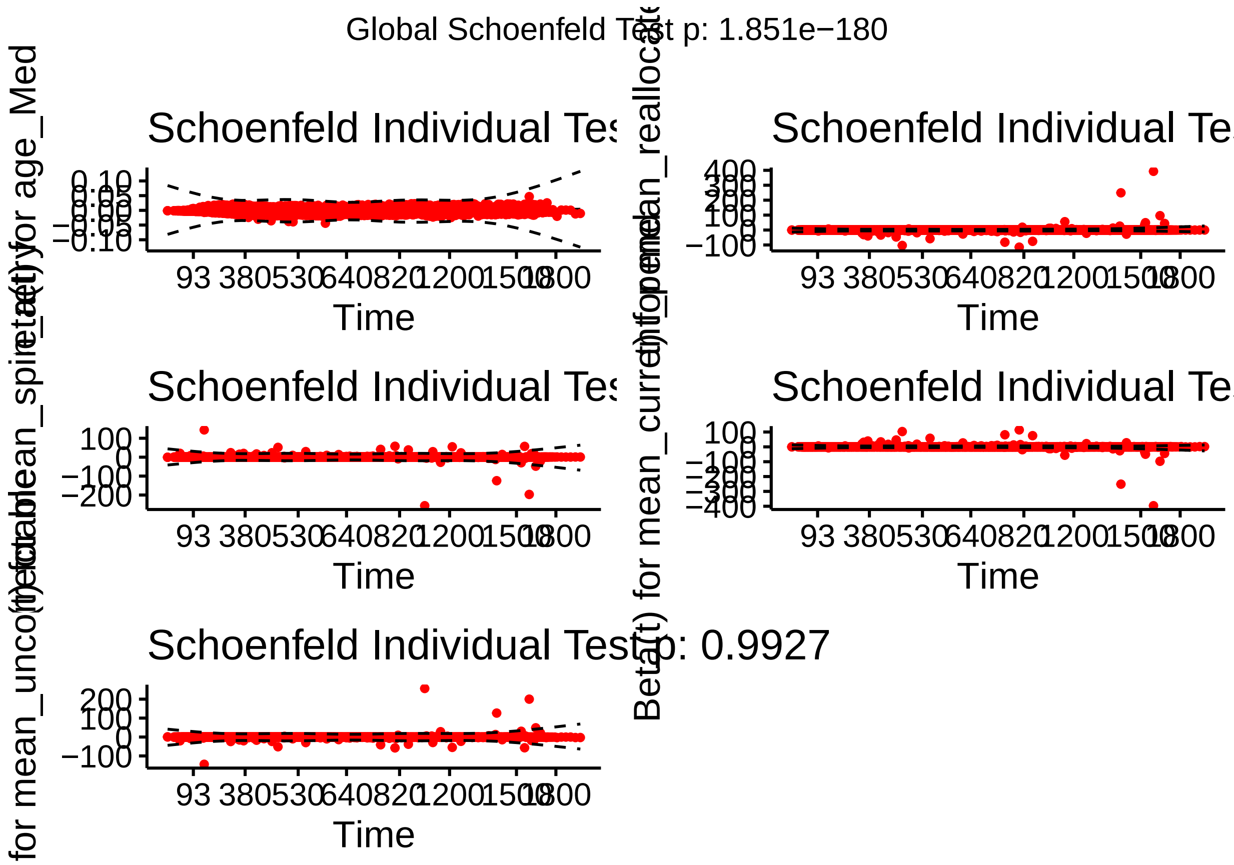
```
## Call:
## coxph(formula = surv_object_HDD ~ age_Med, data = data_group)
##
##   n= 131448, number of events= 2211
##
##               coef exp(coef)  se(coef)     z    Pr(>|z|)
## age_Med -0.001235  0.998766  0.000221 -5.59 0.000000023 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##         exp(coef) exp(-coef) lower .95 upper .95
## age_Med     0.999          1     0.998     0.999
##
## Concordance= 0.544  (se = 0.006 )
## Likelihood ratio test= 31.4  on 1 df,   p=0.00000002
## Wald test            = 31.2  on 1 df,   p=0.00000002
## Score (logrank) test = 31.3  on 1 df,   p=0.00000002
```

```r
# 4.2) Multivariate Cox regression

# 4.2.1) Validation and Diagnostics PH assumptions

# a) Proportional hazards assumption

res_cox_Mult <- coxph(surv_object_HDD ~ age_Med +
                                        mean_reallocated +
                                        mean_spin_retry +
                                        mean_current_pend +
                                        mean_uncorrectable,
                      data = data_group)


test_cox_mult <- cox.zph(res_cox_Mult)

test_cox_mult
```

```
##                          chisq df                  p
## age_Med              2.4553131  1               0.12
## mean_reallocated     1.4467856  1               0.23
## mean_spin_retry      0.0000030  1               1.00
## mean_current_pend    0.0097448  1               0.92
## mean_uncorrectable   0.0000827  1               0.99
## GLOBAL             845.2764059  5 <0.0000000000000002
```

```r
# Conclusion: Test is not statistically significant for each of the covariates,
# Global test is not statistically significant.
# Therefore, we can assume the proportional hazards.
```

```r
# b) Graphical diagnostic

# Scaled Schoenfeld residuals vs ime

ggcoxzph(test_cox_mult)
```

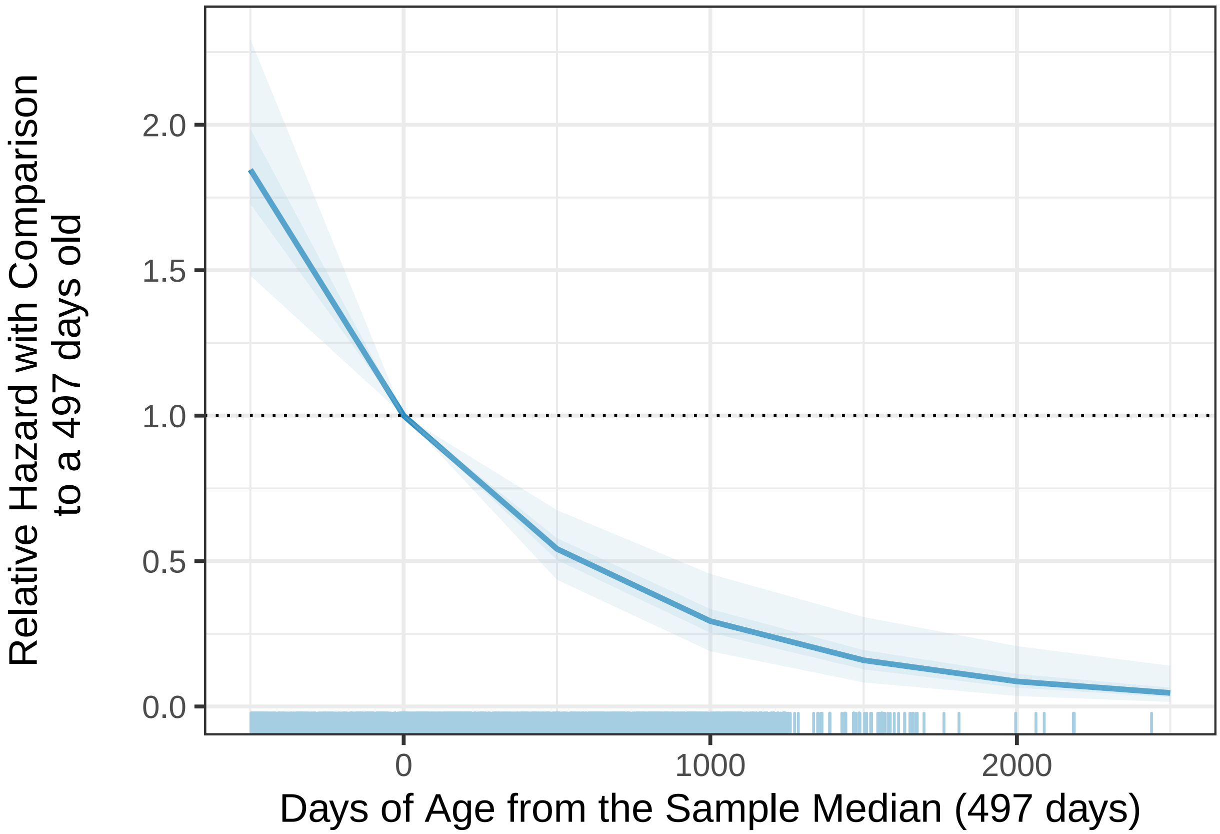



```r
# 4.2.2) Summary of results:

summary(res_cox_Mult)
```

```
## Call:
## coxph(formula = surv_object_HDD ~ age_Med + mean_reallocated +
##     mean_spin_retry + mean_current_pend + mean_uncorrectable,
##     data = data_group)
##
##   n= 131448, number of events= 2211
##
##                          coef  exp(coef)  se(coef)     z     Pr(>|z|)
## age_Med              -0.001322  0.998679  0.000221 -5.97 0.0000000023 ***
## mean_reallocated     -0.048595  0.952567  0.034233 -1.42         0.16
## mean_spin_retry       0.497011  1.643801  0.113621  4.37 0.0000121836 ***
## mean_current_pend     0.055529  1.057100  0.034357  1.62         0.11
```

```
## mean_uncorrectable -0.490172  0.612521  0.113671 -4.31 0.0000161627 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                    exp(coef) exp(-coef) lower .95 upper .95
## age_Med                0.999      1.001     0.998     0.999
## mean_reallocated       0.953      1.050     0.891     1.019
## mean_spin_retry        1.644      0.608     1.316     2.054
## mean_current_pend      1.057      0.946     0.988     1.131
## mean_uncorrectable     0.613      1.633     0.490     0.765
##
## Concordance= 0.571  (se = 0.006 )
## Likelihood ratio test= 144  on 5 df,   p=<0.0000000000000002
## Wald test            = 220  on 5 df,   p=<0.0000000000000002
## Score (logrank) test = 271  on 5 df,   p=<0.0000000000000002
```

```r
# 5) Simulated relative hazards of days on survival time for Hard Disk

library(simPH)

Sim1 <- coxsimLinear(res_cox_Age, b = "age_Med", Xj = seq(-500, 2500, by = 500))

simGG(Sim1,
      xlab = "Days of Age from the Sample Median (497 days)",
      ylab = "Relative Hazard with Comparison\n to a 497 days old\n")
```




```
detach(data_group)
```

References

S. Klugman, H. Panjer, G. Willmont. 2008. *Loss Models: From data to decisions*. Wiley.