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 probabilities that the hard disk works well for 1 year using the data collected in 2019. The 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 ocurrences 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 (rigth 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 bleow u it is recorded at its observed value.

In 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 who are 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 othe 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 like ID.
- Model: Assigned model number by the manufacturer.
- Capacity: Drive capacity in bytes.
- Failure: Constains two states: 0, if the drive is ok, 1 if this is the last day the drive was operational before failing.
- Smart Stats: 80 columns of data of statitics 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 a drive failure. Also, we show the smart variable 9 that display the count of hours in power-on state. This variable let us calculating the age and study time for the survival models.

```r
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 column. Description is extensive
smart_table_code <- smart_table_code[,1:4]

# Change column 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+")

# Let 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
# 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[45]	Low
197	Current Pending Sector Count[45]	Low
198	(Offline) Uncorrectable Sector Count[45]	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)

They let us to measure the period of time that hard drive is working on. So, using these two variables, we have the study time in days:

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

Finally, we take the mean of the normalized values in the aggregation.

```r
#Load libraries
library("data.table")
library("tidyverse")
library("survival")
library("survminer")
library("KMsurv")

# Read Multiple .csv files. 365 files with daily data of HDD.
# Only choose the columns:
```

```r
# 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

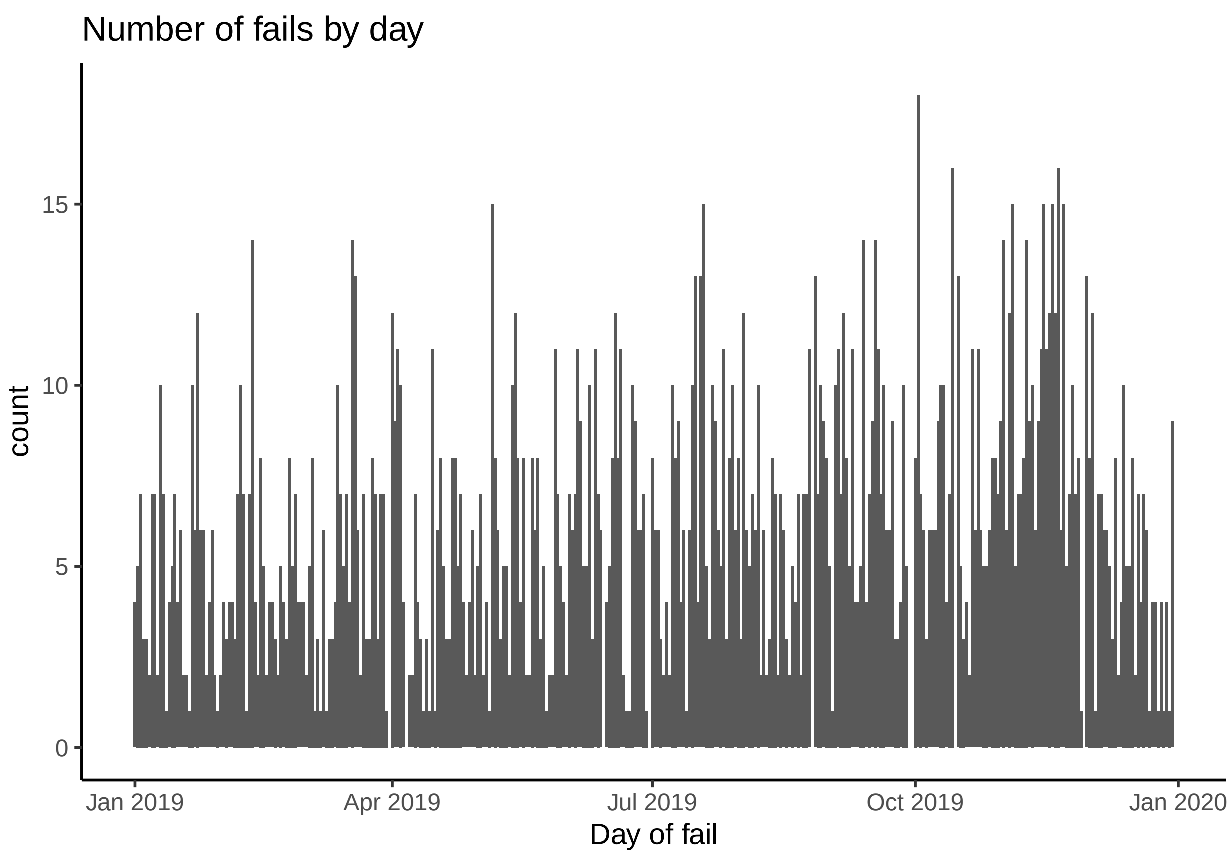
```
# Load aggregate data
data_group <- fread(input = "output/data/data_group_2019.csv",
                    header = TRUE,
                    stringsAsFactors = FALSE)
```

```
# Modify data
data_group <-  data_group %>%
                mutate(model = as.factor(model),
                       serial_number = as.factor(serial_number),
                       min_date = as.Date(min_date),
                       max_date = as.Date(max_date),
                       first_date_fail = as.Date(first_date_fail)
                      ) %>%
```

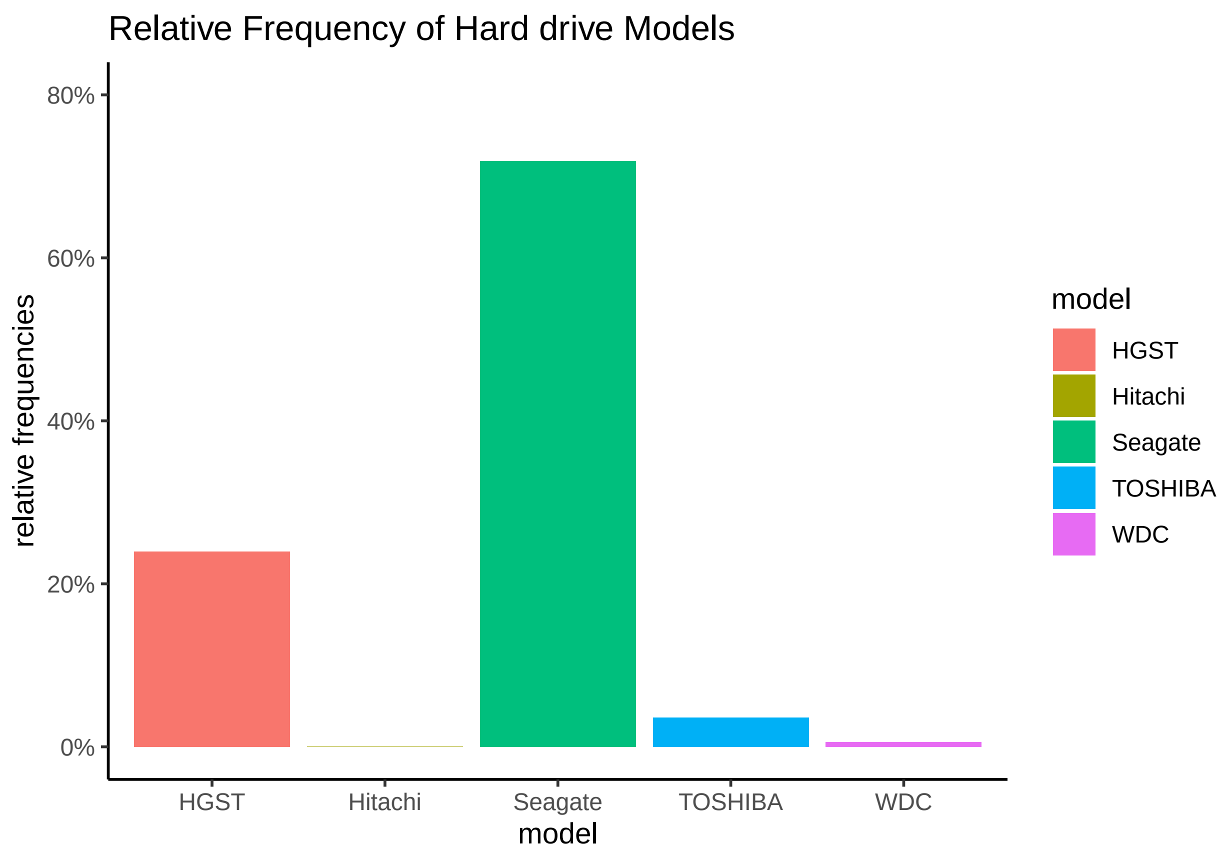
```r
                select(-V1) %>%
                filter(age >= 0,
                       mean_reallocated >= 0)

# Distribution data
summary(data_group)
```

```
##    serial_number          model              TB          count_obs
##  175PP3HDT:     1   DELLBOSS:     0   Min.   : 0.0   Min.   :  1
##  175PP3I4T:     1   HGST    :31459   1st Qu.: 4.0   1st Qu.:302
##  175PP3I5T:     1   Hitachi :    16   Median : 8.0   Median :364
##  175PP3I6T:     1   Seagate :94474   Mean   : 8.4   Mean   :298
##  175PP3I8T:     1   TOSHIBA : 4755   3rd Qu.:12.0   3rd Qu.:365
##  175PP3I9T:     1   WDC     :   744   Max.   :16.0   Max.   :365
##  (Other)  :131442
##     min_date             max_date            min_Hours       max_Hours
##  Min.   :2019-01-01   Min.   :2019-01-01   Min.   :    0   Min.   :    0
##  1st Qu.:2019-01-01   1st Qu.:2019-12-31   1st Qu.: 1230   1st Qu.: 9956
##  Median :2019-01-01   Median :2019-12-31   Median :11935   Median :20654
##  Mean   :2019-02-11   Mean   :2019-12-06   Mean   :14012   Mean   :21128
##  3rd Qu.:2019-01-01   3rd Qu.:2019-12-31   3rd Qu.:23624   3rd Qu.:32037
##  Max.   :2019-12-31   Max.   :2019-12-31   Max.   :70465   Max.   :70770
##
##    count_fail          fail        first_date_fail      first_hour_fail
##  Min.   :0.00   Min.   :0.00   Min.   :2019-01-01   Min.   :   47
##  1st Qu.:0.00   1st Qu.:0.00   1st Qu.:2020-01-01   1st Qu.:70771
##  Median :0.00   Median :0.00   Median :2020-01-01   Median :70771
##  Mean   :0.02   Mean   :0.02   Mean   :2019-12-29   Mean   :69875
##  3rd Qu.:0.00   3rd Qu.:0.00   3rd Qu.:2020-01-01   3rd Qu.:70771
##  Max.   :2.00   Max.   :1.00   Max.   :2020-01-01   Max.   :70771
##
##  mean_reallocated mean_spin_retry mean_current_pend mean_uncorrectable
##  Min.   : 31      Min.   : 75      Min.   : 87       Min.   : 87
##  1st Qu.:100      1st Qu.:100      1st Qu.:100       1st Qu.:100
##  Median :100      Median :100      Median :100       Median :100
##  Mean   :101      Mean   :101      Mean   :101       Mean   :101
##  3rd Qu.:100      3rd Qu.:100      3rd Qu.:100       3rd Qu.:100
##  Max.   :252      Max.   :252      Max.   :252       Max.   :252
##
##       age         study_time
##  Min.   :   0   Min.   :  1
##  1st Qu.:  51   1st Qu.:301
##  Median : 497   Median :364
##  Mean   : 583   Mean   :297
##  3rd Qu.: 984   3rd Qu.:364
##  Max.   :2936   Max.   :579
##
```

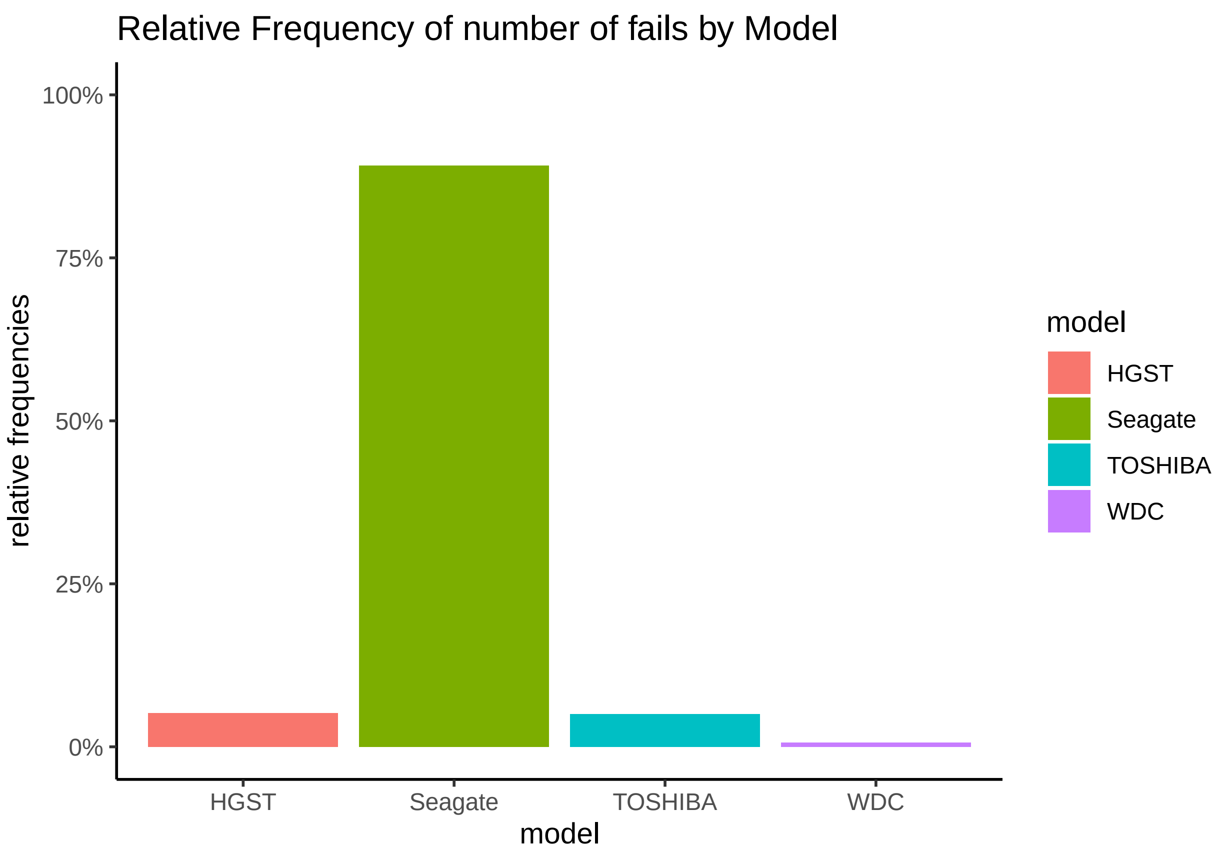
```r
# Number of fails by day

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

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




```r
# Models of HDD (total)

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




```r
# Fails by models of HDD (total)

ggplot(data = fails, aes(model)) +
  geom_bar(aes(y = (..count..)/sum(..count..), fill = model)) +
  scale_y_continuous(labels=scales::percent, limits = c(0,1)) +
  ylab("relative frequencies") +
  labs(title = "Relative Frequency of number of fails by Model") +
  theme_classic()
```



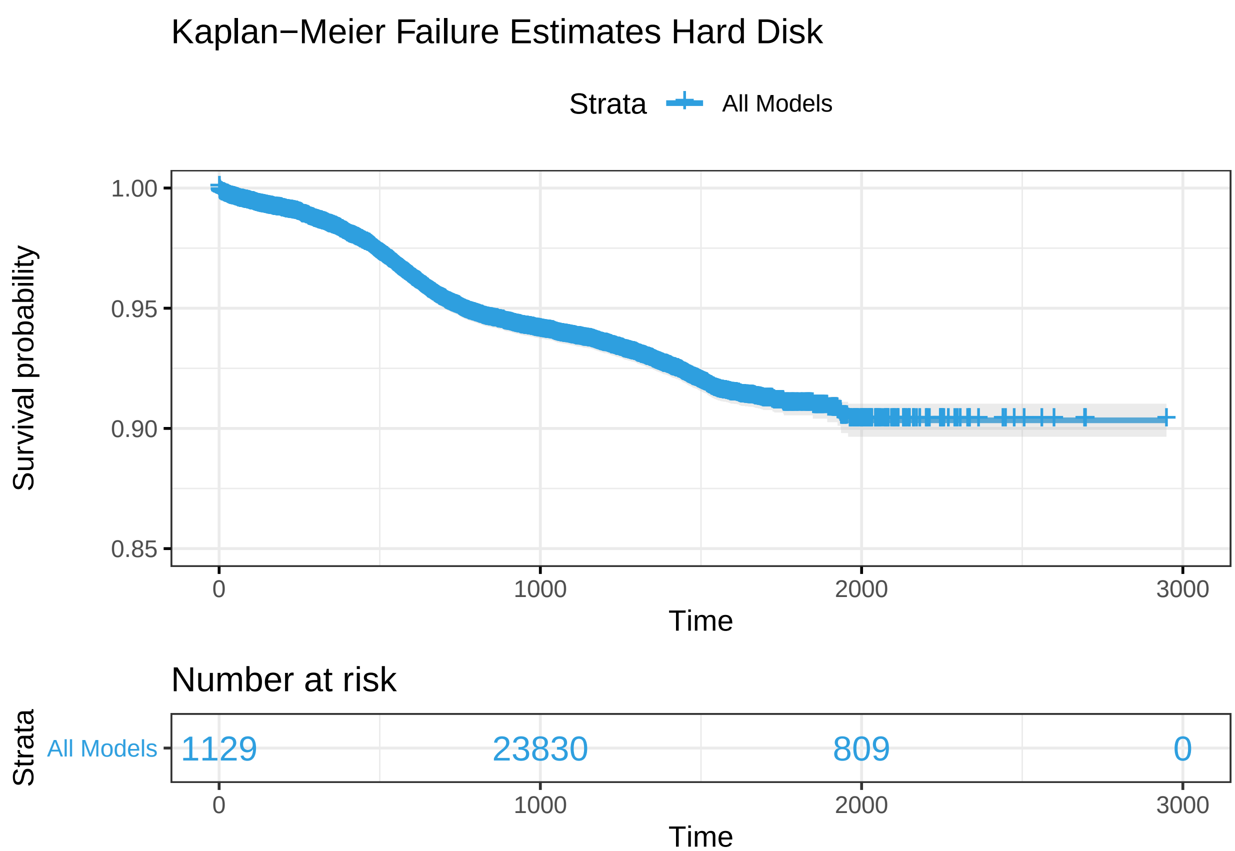
Survival models

```r
# 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
print(km_survival_HDD)
```

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

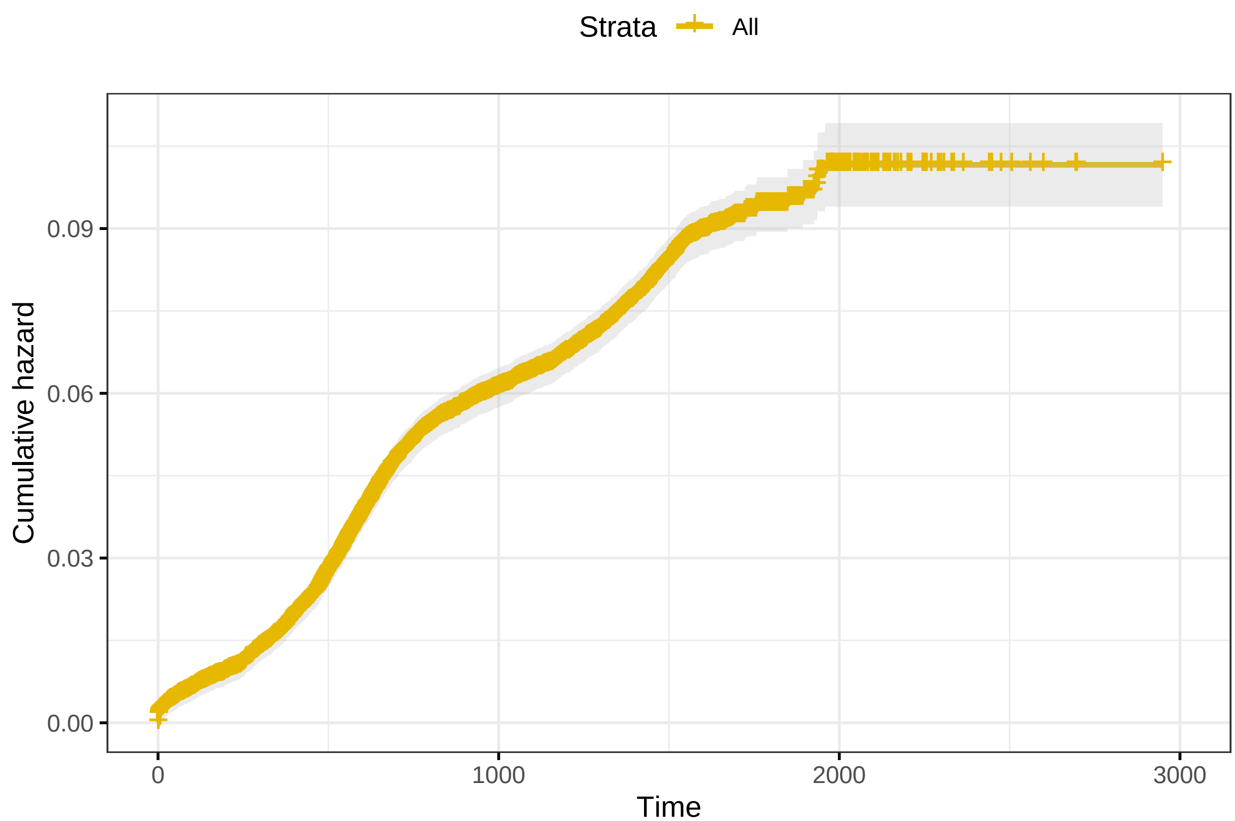
```
## records   n.max n.start  events  median 0.95LCL 0.95UCL
##  131448   36656    1129    2211      NA      NA      NA
```

```
# Graph
ggsurvplot(
  km_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              = "Kaplan-Meier Failure Estimates Hard Disk"
)
```

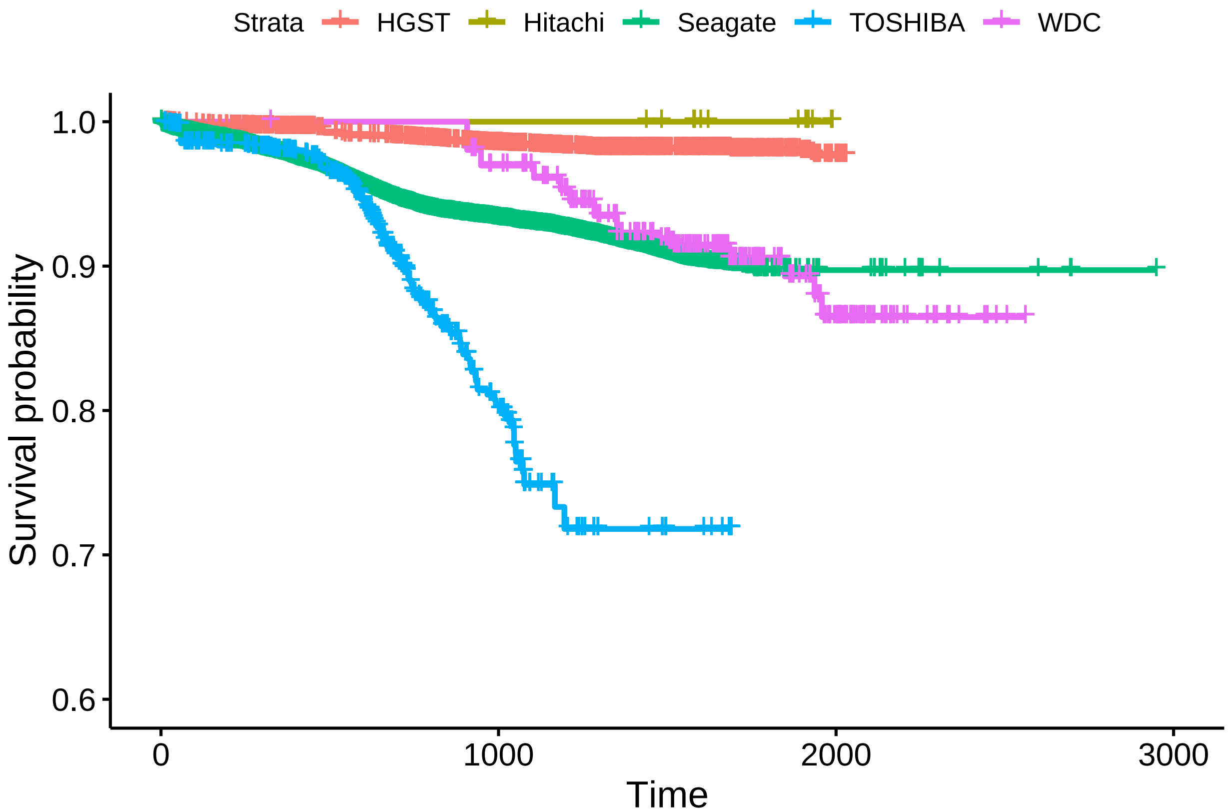



```
# 1.2) cumulative hazard

ggsurvplot(km_survival_HDD,
           data      = data_group,
```

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



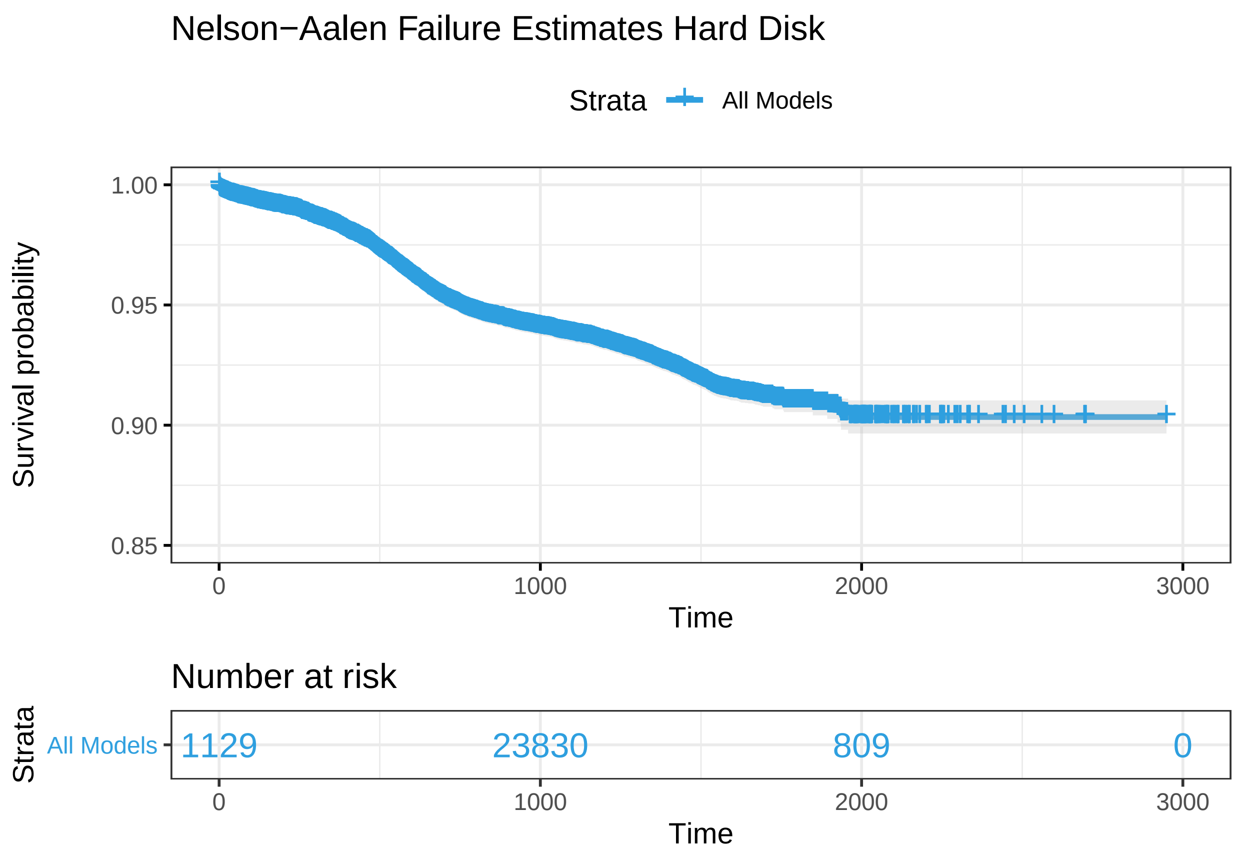

```r
# 2) Kaplan-Meier non-parametric analysis by model
km_survival_model <- survfit(surv_object_HDD ~ model)

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) Univariate Compute the Cox model
res_cox_hdd <- coxph(surv_object_HDD ~ model, data = data_group)
res_cox_hdd
```

```
## Call:
## coxph(formula = surv_object_HDD ~ model, data = data_group)
##
##                    coef exp(coef)  se(coef)  z             p
## modelHGST      -2.08420   0.12441   0.29334 -7 0.000000000001
## modelHitachi   -9.84690   0.00005 441.21496  0          0.98
## modelSeagate   -0.64657   0.52384   0.28150 -2          0.02
## modelTOSHIBA    0.01058   1.01064   0.29800  0          0.97
## modelWDC             NA        NA   0.00000 NA            NA
##
## Likelihood ratio test=369  on 4 df, p=<0.0000000000000002
## n= 131448, number of events= 2211
```

```r
detach(data_group)
```

References

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