ETC5242Assignment

Sahinya Akila(29201128) , Xinyi Cui(29645530), Pranali Angne(32355068), Janice Hsin Hsu(32195109)

9/4/2021

Contents


```
library(tidyverse)
library(survival)
library(survminer)
library(kableExtra)
library(knitr)
library(ggplot2)
```

```r
# Reading the data
churn_dat <- read_csv("https://raw.githubusercontent.com/square/pysurvival/master/pysurvival/datasets/cl

# Filtering data
churn_dat <- churn_dat %>%
  filter(months_active > 0) %>%
  select(c(company_size, months_active, churned)) %>%
  na.omit()
```

1 Question 1

1.1 Write the function

```r
# Kaplan Meier Function
km_model <- function(time, event){
  dataset <- data_frame(time, event)
  dataset1 <- dataset %>%
    group_by(time, event) %>%
    summarise(total_count = n()) %>%
    ungroup() %>%
    pivot_wider(names_from = event,
                values_from = total_count,
                values_fill = 0,
                names_prefix = "status")

result <- data_frame(result = double())
temp_val <- nrow(dataset)
survival_val <- 1
for (i in 1:nrow(dataset1)){
  survival_val <- survival_val * (1 - dataset1$status1[i]/temp_val)
  result <- rbind(result, survival_val)
  temp_val <- temp_val - (dataset1$status0[i] + dataset1$status1[i])
}

dataset1 <- cbind(dataset1, result)
names(dataset1) <- c("time", "status0", "status1", "survival")


  return(dataset1 %>% select(time, survival))
}
```

1.2 The Kaplan-Meier curve for the full data

```r
km_survive <- km_model(churn_dat$months_active, churn_dat$churned)

km_survive %>%
  ggplot(aes(time, survival)) +
  geom_step() +
```

```
  theme_linedraw() +
  theme(panel.background = element_rect(fill = "linen"))
```

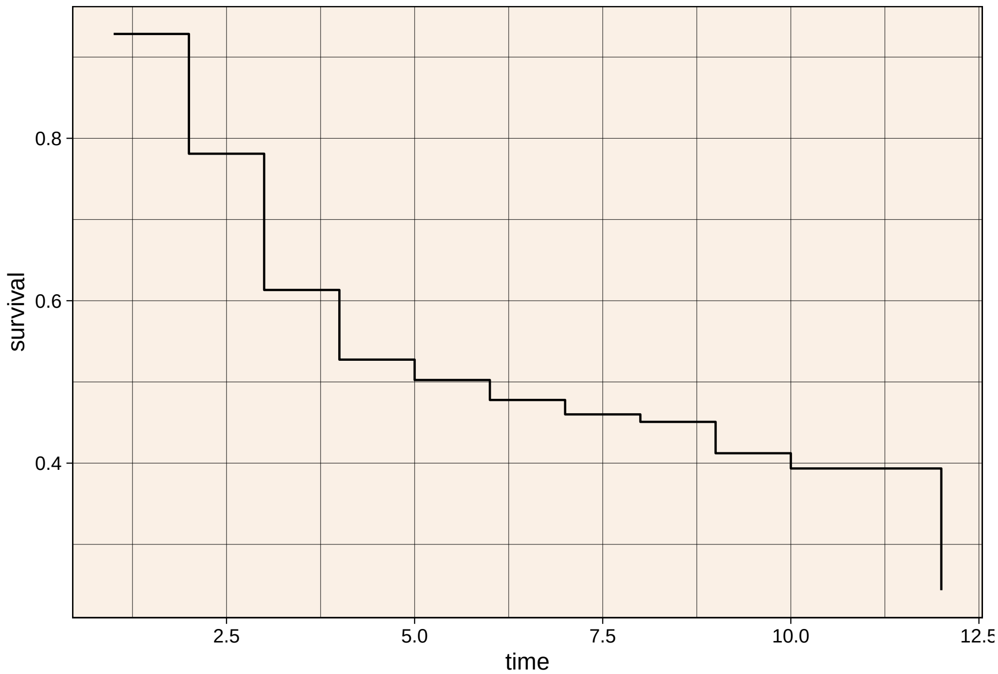


Figure 1: Kaplan Meier Curve for the Survival Data

- The Kaplan Meier curve for the full data shows that the customers churned to 50% by month 5, and the probability slow down until month 10 and then decrease at the end of month 12.

1.3 The Kaplan-Meier curve for each company_size

```
company_km_model <- data.frame(time = double(),
                               survival = double(),
                               company_size = character())
for(size in unique(churn_dat$company_size)){
  filtered <- churn_dat %>% filter(company_size == size)
  final_model <- km_model(filtered$months_active,
                          filtered$churned) %>%
    mutate(company_size = size)
  company_km_model <- rbind(company_km_model, final_model)
}

company_km_model %>%
  ggplot(aes(time, survival)) +
  geom_step() +
  facet_wrap(~company_size) +
  theme(plot.background = element_rect(fill = "white")) +
  theme(panel.background = element_rect(fill = "#e3ebbc",
```

```
                                    colour = "black",
                                    size = 0.5, linetype = "solid"))
```

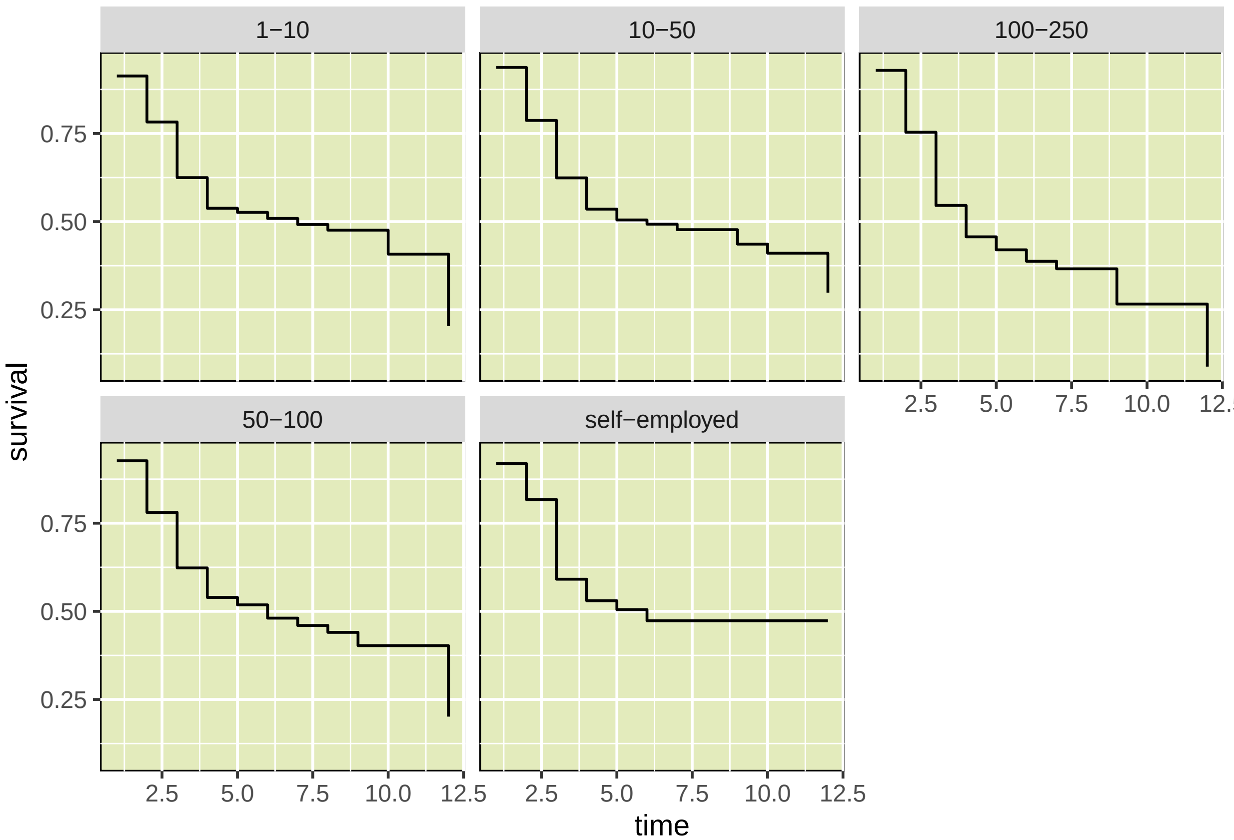


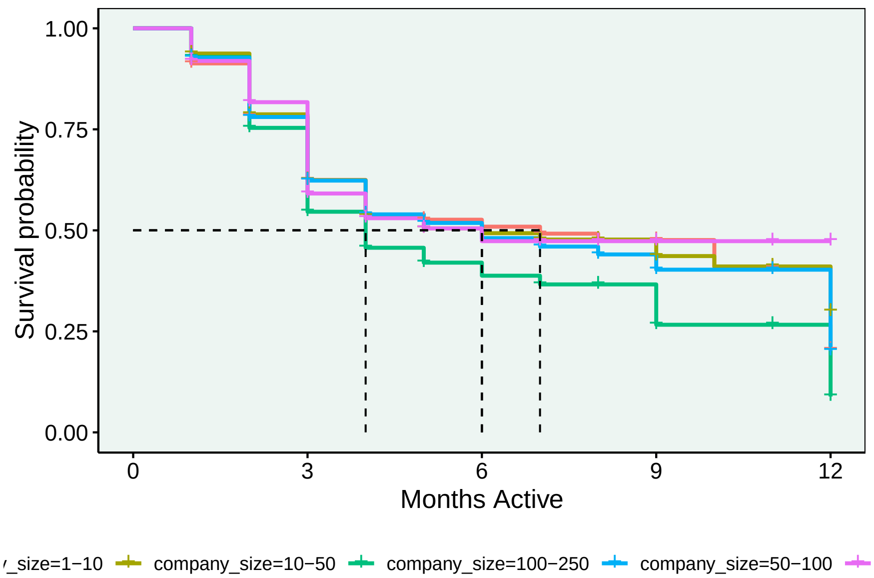
Figure 2: The Kaplan-Meier curve for each company_size

- The Kaplan-Meier curve for the full data and company size of 50-100 are very similar.
- For all of the company regardless of their sizes, they all have a rapid churned decrease for the first 4 months, and come with a flatter churned from month 6 to month 8. Next, a drop at the end of the time at month 12 except for self-employed company.
- For self-employed company, the survival probability stays around 50% in month 6. As there are no customers churned after month 6, there is a flat line shown in the graph.

2 Question 2

2.1 Compute the Kaplan-Meir curve and use this to estimate the median churn time.

```
fit <- survfit(Surv(months_active, churned) ~ company_size,
                 data = churn_dat)

g2 <- ggsurvplot(fit,
                surv.median.line = "hv",
                conf.int = FALSE)

g2$plot +
  labs(x = "Months Active") +
  theme(legend.position="bottom") +
  theme(plot.background = element_rect(fill = "white")) +
  theme(panel.background = element_rect(fill = "#edf5f2",
                                        colour = "black",
                                        size = 0.5, linetype = "solid"))
```




```
median_function <- function(fit){
  index <- which.min(abs(fit$surv - 0.5))
  median <- fit$time[index]
  return(median)
}
```

```
for (size in unique(churn_dat$company_size)){
  temp_data <- churn_dat %>% filter(company_size == size)
  name <- size
  assign(name, survfit(Surv(months_active, churned) ~ company_size, data = temp_data))
}
```

```
company_median <- data_frame(company_size = unique(churn_dat$company_size),
                             median = c(NA, NA, NA, NA, NA))

for (i in 1:length(company_median$company_size)){
  company_median$median[i] <- median_function(get(company_median$company_size[i]))
}
```

Table 1: Medians for different company sizes

company_size	median
10-50	**5**
100-250	**4**
50-100	**5**
1-10	**7**
self-employed	**5**

```
company_median %>%
  knitr::kable(caption = "Medians for different company sizes") %>%
  kable_styling(c("hover", "striped")) %>%
  column_spec(1:2, bold = T) %>%
  row_spec(1:5, color = "black", background = "#e9e6f0")
```

2.2 Use a non-parametric bootstrap to construct 90% confidence intervals for the median of each company size

```
bootstrapmedian <- function(df_median, df){
  bootstrap <- tibble(experiment = rep(1:1000, each = nrow(df)),
                        ind = sample(1:nrow(df), size = nrow(df)*1000, replace = TRUE),
                        timestar = df$months_active[ind],
                        churnstar = df$churned[ind])

  bias <- bootstrap %>%
    group_by(experiment) %>%
    summarise(delta = median_function(df_median) -
              median_function(survfit(Surv(timestar,
                                            churnstar) ~ experiment))) %>%
    na.omit()

  ci <- median_function(df_median) +
    quantile(bias$delta, c(0.05, 0.95))

  return(ci)
}
```

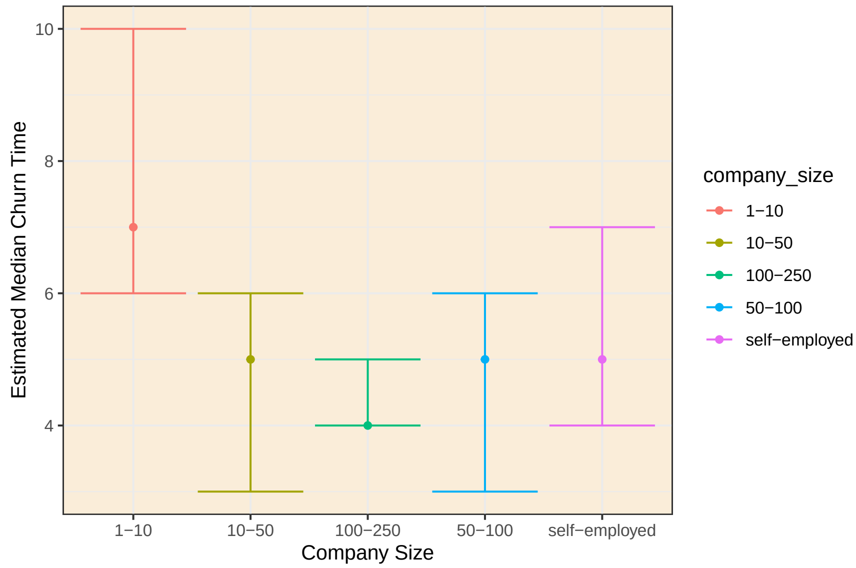
```
company_median_ci <- data_frame(company_size = unique(churn_dat$company_size),
                                median = c(NA, NA, NA, NA, NA),
                                lci = c(NA, NA, NA, NA, NA),
                                uci = c(NA, NA, NA, NA, NA))

for (i in 1:length(company_median_ci$company_size)){
  ci <- bootstrapmedian(get(company_median_ci$company_size[i]),
                        churn_dat %>%
                          filter(company_size == company_median_ci$company_size[i]))
  company_median_ci$median[i] <- median_function(get(company_median_ci$company_size[i]))
         company_median_ci$lci[i] <- ci[1]
         company_median_ci$uci[i] = ci[2]
```

company_size	median	lci	uci
10-50	**5**	**3**	**6**
100-250	**4**	**4**	**5**
50-100	**5**	**3**	**6**
1-10	**7**	**6**	**10**
self-employed	**5**	**4**	**7**

```
}
company_median_ci %>%
  kbl() %>%
  kable_styling(c("hover", "striped")) %>%
  column_spec(1:4, bold = T) %>%
  row_spec(1:5, color = "black", background = "#e6f0ed")
```

2.3 Make a plot that shows that estimate of the median and the corresponding confidence interval on the same axes

```
ggplot(company_median_ci,
       aes(x = company_size,
           y = median,
           color = company_size)) +
  geom_errorbar(aes(ymax = uci, ymin = lci)) +
  geom_point() +
  theme_bw() +
  labs(x = "Company Size",
       y = "Estimated Median Churn Time",
       caption = "Estimated mean under 90% CI") +
  theme(plot.background = element_rect(fill = "white")) +
  theme(panel.background = element_rect(fill = "#faedd9",
                                         colour = "black",
                                         size = 0.5, linetype = "solid"))
```



The table above demonstrates the median churn time estimated for different company size. - Company size of 1-10 have the highest estimated median of 7 months. - Company size of 100-250 have the lowest estimated median of 4 months. - The rest of the company sizes have the same estimated median of 5 months.

3 Question 3

3.1 Choose company size of 50-100

```
q3_company <- churn_dat %>%
  filter(company_size == "50-100")

q3_fit <- survfit(Surv(months_active, churned) ~1,
                  data = q3_company)
```

3.2 Use a nonparametric bootstrap to re-sample the data and construct 90% confidence intervals for the survival curve at each time.

```
bootstrap_time <- tibble(experiment = rep(1:1000, each =672),
                        ind = sample(1:672,
                                     size = 672*1000,
                                     replace = TRUE),
                                    months_active = q3_company$months_active[ind],
                                    churned = q3_company$churned[ind])
bias_time <- bootstrap_time %>%
  group_by(experiment) %>%
  summarise(delta = q3_fit$surv - survfit(Surv(months_active, churned) ~1)$surv)

lower <- q3_fit$surv + quantile(bias_time$delta, 0.05)
upper <- q3_fit$surv + quantile(bias_time$delta, 0.95)
```

Table 2: 90survival curve at each time for company size 50-100

Month	Probability	Lower Confidence Interval	Upper Confidence Interval
1	0.9270833	0.8774578	0.9784106
2	0.7805394	0.7309138	0.8318667
3	0.6231333	0.5735077	0.6744606
4	0.5395180	0.4898924	0.5908453
5	0.5183604	0.4687348	0.5696877
6	0.4807375	0.4311119	0.5320648
7	0.4598358	0.4102103	0.5111632
8	0.4404062	0.3907806	0.4917335
9	0.4026571	0.3530315	0.4539844
10	0.4026571	0.3530315	0.4539844
11	0.2013285	0.1517030	0.2526558

```
Month <- c(1:11)

time_50_100_CIs <- data.frame(Month, q3_fit$surv, lower, upper) %>%
  rename("Probability" = q3_fit.surv,
         "Lower Confidence Interval" = lower,
         "Upper Confidence Interval" = upper)

kable(time_50_100_CIs,
      caption = "90% confidence intervals for the
      survival curve at each time for company size 50-100") %>%
  kable_styling(c("hover", "striped"))%>%
  row_spec(1:11, color = "black", background = "#e6f0ed")
```

3.3 Compute simultaneous coverage for the entire survival function.

4 Question 4

4.1 Write a function to compute the log-rank test statistic for two populations.

```
q4_comp <- churn_dat %>%
 mutate(comp_hyp = case_when(company_size == "50-100" ~ 1,
                             company_size== "100-250" ~ 2,
                             TRUE ~ 0))

q4_comp <- q4_comp %>%
  filter(comp_hyp == 1| comp_hyp == 2)
```

```
survdiff(Surv(months_active, churned) ~ comp_hyp,
         data=q4_comp)
```

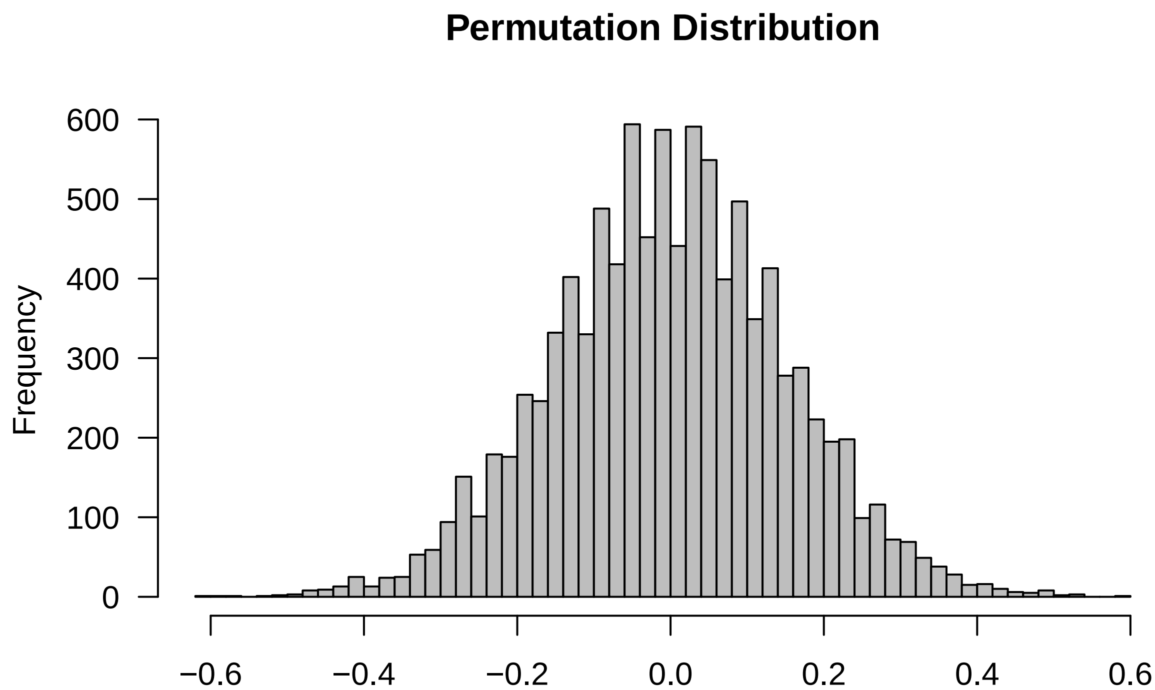
```
## Call:
## survdiff(formula = Surv(months_active, churned) ~ comp_hyp, data = q4_comp)
```

```
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## comp_hyp=1 672      313      332      1.14      5.26
## comp_hyp=2 240      135      116      3.27      5.26
##
##  Chisq= 5.3  on 1 degrees of freedom, p= 0.02
```

```r
treatment <- q4_comp$churned
outcome <- q4_comp$months_active

#Difference in means
original <- diff(tapply(outcome, treatment, mean))
mean(outcome[treatment==1])-mean(outcome[treatment==0])
```

```
## [1] -1.896937
```

```r
#Permutation test
permutation.test <- function(treatment, outcome, n){
  distribution=c()
  result=0
  for(i in 1:n){
    distribution[i]=diff(by(outcome,
                            sample(treatment, length(treatment), FALSE),
                            mean))
  }
  result=sum(abs(distribution) >= abs(original))/(n)
  return(list(result, distribution))
}

test1 <- permutation.test(treatment, outcome, 10000)
hist(test1[[2]], breaks=50, col='grey',
     main="Permutation Distribution",
     las=1, xlab='')
abline(v=original, lwd=3, col="red")
```




```r
test1[[1]]
```

```
## [1] 0
```

```r
#Compare to t-test
t.test(outcome~treatment)
```

```
##
##  Welch Two Sample t-test
##
## data:  outcome by treatment
## t = 13.702, df = 842.56, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.625195 2.168678
## sample estimates:
## mean in group 0 mean in group 1
##        4.823276        2.926339
```

- The outcome of the analysis shows that the p value is extremely small which is statistically significant. It points out the strong evidence against the null hypothesis. Therefore, the churn rate is significantly different between these two company sizes.
- Additionally, for the permutation distribution, we can figure out from the graph above that it is normally distributed with the mean of 0.

Table 3: Estimated mean and the median of the churn time for each company size with Weibull distribution

company_size	median	mean
10-50	5.69	6.79
100-250	4.7	5.45
50-100	5.56	6.61
1-10	5.74	7
self-employed	6.23	7.92

5 Question 5

5.1 fit a Weibull distribution to the survival data to estimate the mean and the median of the churn time for each company size

```
function_fit <- function(dat){
  fit <- survreg(Surv(months_active, churned) ~ 1,
                 data = dat,
                 dist = "weibull" )
  rweibull_shape <- 1 / fit$scale ## Approximately 3
  rweibull_scale <- exp(coef(fit)) ## approximately 7

median <- rweibull_scale*log(2)^(1/rweibull_shape)
mean <- rweibull_scale*gamma(1+(1/rweibull_shape))

return(c(median, mean))

}
```

```
weibull <- data_frame(company_size = character(),
                      median = double(),
                      mean = double())
for (size in unique(churn_dat$company_size)){
  temp_data <- churn_dat %>% filter(company_size == size)
  return_values <- function_fit(temp_data)
  weibull <- rbind(weibull,
                   c(size, round(return_values[1], 2),
                     round(return_values[2],2)))
}

names(weibull) <- c("company_size", "median", "mean")

kable(weibull,
caption = "Estimated mean and the median of
      the churn time for each company size with Weibull distribution") %>%
  kable_styling(c("hover", "striped"))%>%
  row_spec(1:5, color = "black", background = "#e6f0ed")
```

- The mean and median estimates derived from the weibull distribution are different from the estimates obtained using the Kaplan-Meier model. There is a increase by 0.5 to 1 in the values produced in this model, whereas the values given by the Kaplan-Meier model are lower.

- Therefore, parametric estimators are to be given more consideration when compared to non-parametric, Kaplan eier estimations. Also, we find that the reduction in efficiency of the Kaplan–Meier survival estimate becomes negligible quickly as the number of parameters in the parametric model increases.