Assessment_1

Luka.C, Harris.P, Jason.S, Rounak.A

09/09/2021

Question 1

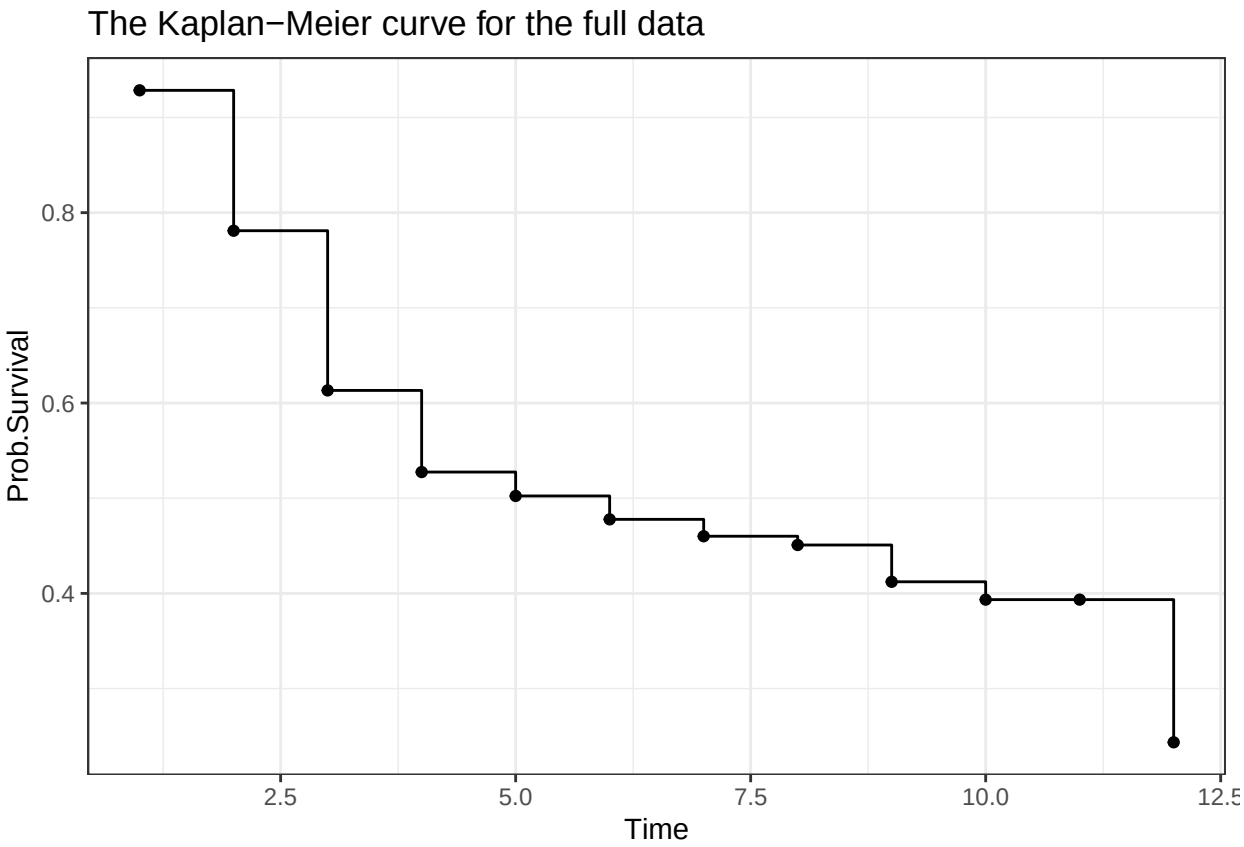
Kaplan Meier

```
#t = at a particular time period
#n = # of the individuals that are still surviving at t (i.e customer still using our services)
#d = # of event at t (i.e customer left)
KM_estimate <- function (time, event)
{
    sorted_time <- sort(time)
    event <- event[order(time)] # event being ordered
    ni <- length(time):1
    ni <- ni[!duplicated(sorted_time)]
    di <- tapply(event, sorted_time, sum)
    ti <- unique(sorted_time)
    si <- (ni - di)/ni
    cum_survivial_i <- cumprod(si)
    cum_risk_i <- 1 - cum_survivial_i
    results <- cbind(time = ti, n_risk = ni, n_events = di, condsurv = si,
        survival = cum_survivial_i, risk = cum_risk_i)
    dimnames(results)[1] <- list(NULL)
    results[, ]
}
```

Plot full data

```
time <-  churn_dat$months_active
event <- churn_dat$churned
result <- KM_estimate(time, event)
result <- as.data.frame(result)
```

```
ggplot(result, aes(x=time, y = survival))+
  geom_point()+
  geom_step ()+
  theme_bw()+
  ggtitle("The Kaplan-Meier curve for the full data") +
  labs(x= 'Time',
         y="Prob.Survival")
```

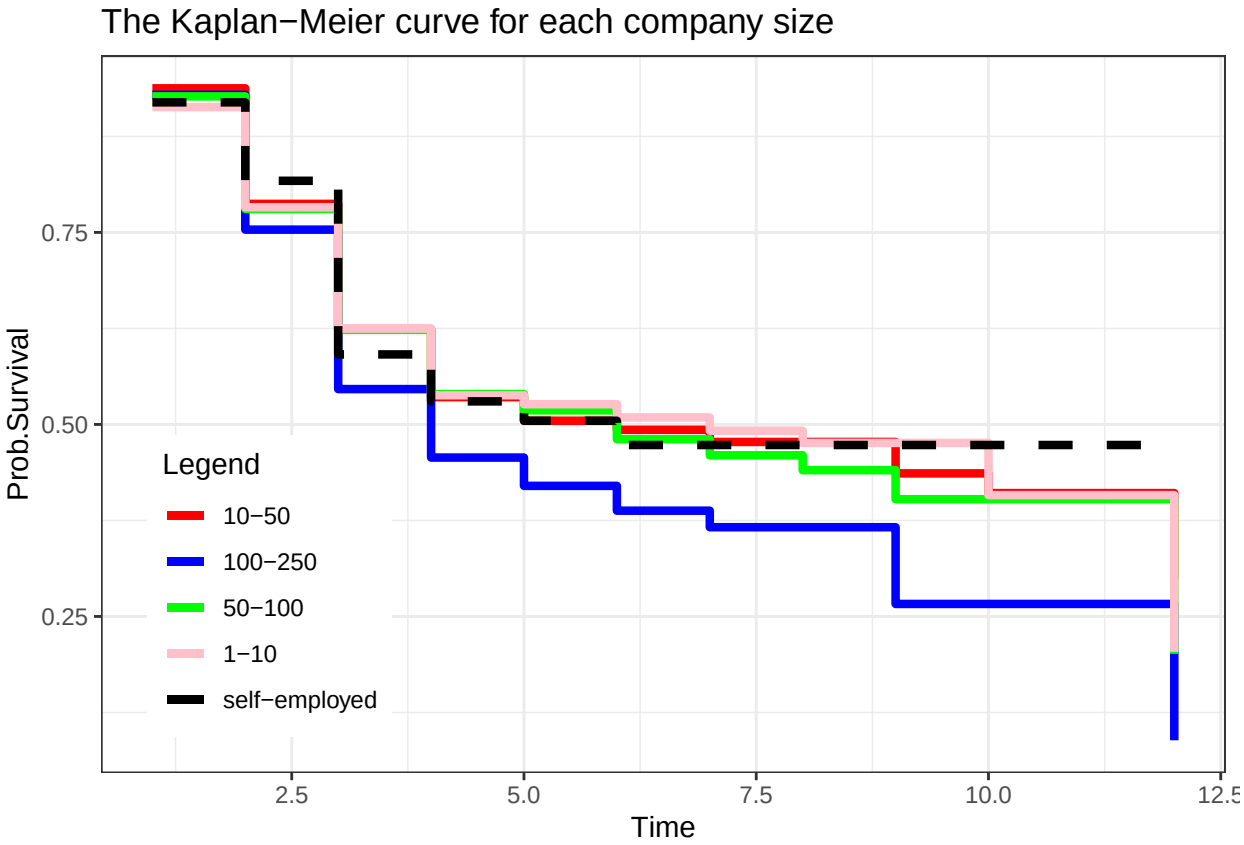


Plot for each individual company size

```
df_10to50 <-  churn_dat %>% filter(company_size == "10-50")
s1_time <-  df_10to50$months_active
s1_event <- df_10to50$churned
s1_result <- KM_estimate(s1_time, s1_event)
s1_result <- as.data.frame(s1_result)
#100-250
df_100to250 <- churn_dat %>% filter(company_size ==  "100-250")
s2_time <-  df_100to250$months_active
s2_event <- df_100to250$churned
s2_result <- KM_estimate(s2_time, s2_event)
s2_result <- as.data.frame(s2_result)
#"50-100"
df_50to100 <- churn_dat %>% filter(company_size ==  "50-100")
s3_time <-  df_50to100$months_active
s3_event <- df_50to100$churned
s3_result <- KM_estimate(s3_time, s3_event)
s3_result <- as.data.frame(s3_result)
#"1-10"
df_1to10 <- churn_dat %>% filter(company_size ==  "1-10")
s4_time <-  df_1to10$months_active
s4_event <- df_1to10$churned
s4_result <- KM_estimate(s4_time, s4_event)
```

```r
s4_result <- as.data.frame(s4_result)
# "self-employed"
df_self_employed <- churn_dat %>% filter(company_size ==  "self-employed")
s5_time <-  df_self_employed$months_active
s5_event <- df_self_employed$churned
s5_result <- KM_estimate(s5_time, s5_event)
s5_result <- as.data.frame(s5_result)
```

```r
colors <- c("10-50"="red", "100-250" = "blue","50-100" = "green","1-10" = "pink","self-employed" = "bla
ggplot()+
    geom_step(data =s1_result,
              aes(x=time, y = survival,
              color = "10-50"),
              size = 1.5)+
    geom_step(data =s2_result,
              aes(x=time, y = survival,
              color = "100-250"),
              size = 1.5)+
    geom_step(data =s3_result,
              aes(x=time, y = survival,
              color = "50-100"),
              size = 1.5)+
    geom_step(data =s4_result,
              aes(x=time, y = survival,
              color = "1-10"),
              size = 1.5)+
    geom_step(data =s5_result,
              aes(x=time, y = survival,
              color = "self-employed"),
              size = 1.5,
              linetype = 2)+
    labs(x= 'Time',
         y="Prob.Survival",
         color = 'Legend')+
        scale_color_manual(values = colors)+
    theme_bw()+
    theme(legend.position = c(.15,.26))+
    ggtitle("The Kaplan-Meier curve for each company size")
```


The Kaplan−Meier curve for each company size
Prob.Survival
0.75
0.50
0.25
Legend
10−50
100−250
50−100
1−10
self−employed
2.5
5.0
7.5
10.0
12.5
Time

Interpretation :

The Kaplan-Meier curves for each respective company size are overall pretty similar in shape. They all exhibit the same early drop offs in survival probability followed by more stability in the back two thirds of the time period.

The somewhat outlier of the group however is the company size of 100 to 250 clients. Its curve drops lower, a lot earlier than the rest of the curves, this indicates it is losing customers quicker than the others and, with the curve finishing lowest out of the five, also illustrates that it has a lower rate of keeping customers long term and hence a higher customer churn rate. Another interesting feature of the graph is the self-employed data. The graph for the self-employed companies shows the graph running off from about time 6 indicating that no more clients had churned from that point onwards to the point of censoring. However, this company size also had the lowest number of observations being only 62 which could explain the disparity in customer churn to the other companies.

Although size of data definitely plays a role in determining the shape and pattern of the graphs, a more logical reason to explain the graphs could be that companies with larger numbers of clients may struggle to attain the same depth and quality of business-client relations as their low client number counterparts, hence resulting in higher customer churn rates.

Question 2

Function to find median

```
near_median <- function(fit){
  if (length(fit$n) > 1) {
    stop("This only works for a single survival curve!")
  }
  index <- which.min(abs(fit$surv - 0.5))
  return(fit$time[index])
}
average_median <- function(fit) {
if (length(fit$n) > 1) {
stop("This only works for a single survival curve!")
}
suppressWarnings(lower_ind <- which.min(log(fit$surv - 0.5)))
suppressWarnings(upper_ind <- which.min(log(0.5 - fit$surv)))
return((fit$time[lower_ind] + fit$time[upper_ind])/2)
}
```

Filter data

```
filter_df <- function(size){
df <- churn_dat %>% filter(company_size == size)
time <-  df$months_active
event <- df$churned
fit <- surv_fit(Surv(time,event)~1, data = df)
return(fit)
}
```

```
#here estimate median based on sizes
fit <- filter_df("10-50")
fit %>% tidy()
```

```
## # A tibble: 12 x 8
##     time n.risk n.event n.censor estimate std.error conf.high conf.low
##    <dbl>  <dbl>   <dbl>    <dbl>    <dbl>     <dbl>     <dbl>    <dbl>
##  1     1    673      42       14    0.938   0.00994     0.956    0.919
##  2     2    617      99       35    0.787   0.0202      0.819    0.757
##  3     3    483     100       59    0.624   0.0308      0.663    0.588
##  4     4    324      46       69    0.536   0.0382      0.577    0.497
##  5     5    209      12       69    0.505   0.0418      0.548    0.465
##  6     6    128       3        0    0.493   0.0440      0.537    0.452
##  7     7    125       4       55    0.477   0.0469      0.523    0.435
##  8     8     66       0       31    0.477   0.0469      0.523    0.435
##  9     9     35       3       15    0.436   0.0699      0.500    0.380
## 10    10     17       1        0    0.411   0.0925      0.492    0.343
## 11    11     16       0        5    0.411   0.0925      0.492    0.343
## 12    12     11       3        8    0.299   0.207       0.448    0.199
```

```
s1_median <- average_median(fit)
```

The median time is where the survival probability is equal to 0.5. Although there is no exact time where this occurs we know that the median exists between times 5 and 6. As the survival probabilities of the two times are near equally far from 0.5 (which are 0.505 and 0.493 respectively).

Therefore, for size 10-50, the average median is 5.5

```
#here estimate median based on sizes
fit <- filter_df("100-250")
fit %>% tidy()
```

```
## # A tibble: 11 x 8
##     time n.risk n.event n.censor estimate std.error conf.high conf.low
##    <dbl>  <dbl>   <dbl>    <dbl>    <dbl>     <dbl>     <dbl>    <dbl>
##  1     1    240      17        6   0.929     0.0178     0.962   0.897
##  2     2    217      41        9   0.754     0.0373     0.811   0.700
##  3     3    167      46       23   0.546     0.0606     0.615   0.485
##  4     4     98      16       20   0.457     0.0752     0.529   0.394
##  5     5     62       5       18   0.420     0.0841     0.495   0.356
##  6     6     39       3        0   0.388     0.0960     0.468   0.321
##  7     7     36       2       12   0.366     0.104      0.449   0.299
##  8     8     22       0       11   0.366     0.104      0.449   0.299
##  9     9     11       3        2   0.266     0.212      0.403   0.176
## 10    11      6       0        3   0.266     0.212      0.403   0.176
## 11    12      3       2        1   0.0888    0.844      0.464   0.0170
```

```
s2_median <- near_median(fit)
```

The median time is where the survival probability is equal to 0.5. Although there is no exact time where this occurs we know that the median exists between times 5 and 6. As the survival probabilities of the two times are NOT nearly equal to 0.5 (which are 0.54 and 0.45 respectively), we can use the average median of 5.5 as the most suitable measure of the median.

Therefore, for size 100-250, the near median is 4

```
#here estimate median based on sizes
fit <- filter_df("50-100")
fit %>% tidy()
```

```
## # A tibble: 11 x 8
##     time n.risk n.event n.censor estimate std.error conf.high conf.low
##    <dbl>  <dbl>   <dbl>    <dbl>    <dbl>     <dbl>     <dbl>    <dbl>
##  1     1    672      49       22    0.927    0.0108     0.947   0.908
##  2     2    601      95       25    0.781    0.0207     0.813   0.749
##  3     3    481      97       71    0.623    0.0309     0.662   0.587
##  4     4    313      42       67    0.540    0.0381     0.581   0.501
##  5     5    204       8       72    0.518    0.0406     0.561   0.479
##  6     6    124       9        0    0.481    0.0478     0.528   0.438
##  7     7    115       5       39    0.460    0.0517     0.509   0.415
##  8     8     71       3       33    0.440    0.0574     0.493   0.394
##  9     9     35       3       21    0.403    0.0773     0.469   0.346
## 10    11     11       0        7    0.403    0.0773     0.469   0.346
## 11    12      4       2        2    0.201    0.506      0.543   0.0747
```

```
s3_median <- average_median(fit)
```

The median time is where the survival probability is equal to 0.5. Although there is no exact time where this occurs we know that the median exists between times 5 and 6. As the survival probabilities of the two times are nearly equal to 0.5 (which are 0.51 and 0.48 respectively)

Therefore, for size 50-100, the near median is 5.5

```
#here estimate median based on sizes
fit <- filter_df("1-10")
fit %>% tidy()
```

```
## # A tibble: 12 x 8
##     time n.risk n.event n.censor estimate std.error conf.high conf.low
##    <dbl>  <dbl>   <dbl>    <dbl>    <dbl>     <dbl>     <dbl>    <dbl>
##  1     1    311      27        4    0.913    0.0175     0.945   0.882
##  2     2    280      40       12    0.783    0.0300     0.830   0.738
##  3     3    228      46       38    0.625    0.0448     0.682   0.572
##  4     4    144      20       32    0.538    0.0559     0.600   0.482
```

```
##  5     5     92       2       29    0.526    0.0581     0.590   0.470
##  6     6     61       2        0    0.509    0.0627     0.576   0.450
##  7     7     59       2       26    0.492    0.0672     0.561   0.431
##  8     8     31       1       13    0.476    0.0748     0.551   0.411
##  9     9     17       0       10    0.476    0.0748     0.551   0.411
## 10    10      7       1        0    0.408    0.171      0.571   0.292
## 11    11      6       0        4    0.408    0.171      0.571   0.292
## 12    12      2       1        1    0.204    0.728      0.849   0.0490
```

```
s4_median <- average_median(fit)
```

The median time is where the survival probability is equal to 0.5. Although there is no exact time where this occurs we know that the median exists between times 5 and 6. As the survival probabilities of the two times are nearly equal to 0.5 (which are .50 and .49 respectively)

Therefore, for size 1-10, the near median is 6.5

```
#here estimate median based on sizes
fit <- filter_df("self-employed")
fit %>% tidy()
```

```
## # A tibble: 11 x 8
##     time n.risk n.event n.censor estimate std.error conf.high conf.low
##    <dbl>  <dbl>   <dbl>    <dbl>    <dbl>     <dbl>     <dbl>    <dbl>
##  1     1     62       5        3    0.919    0.0376     0.990    0.854
##  2     2     54       6        1    0.817    0.0611     0.921    0.725
##  3     3     47      13        5    0.591    0.109      0.732    0.478
##  4     4     29       3        5    0.530    0.126      0.678    0.414
##  5     5     21       1        4    0.505    0.135      0.658    0.387
##  6     6     16       1        0    0.473    0.150      0.635    0.353
##  7     7     15       0        5    0.473    0.150      0.635    0.353
##  8     8     10       0        3    0.473    0.150      0.635    0.353
##  9     9      7       0        4    0.473    0.150      0.635    0.353
## 10    11      3       0        2    0.473    0.150      0.635    0.353
## 11    12      1       0        1    0.473    0.150      0.635    0.353
```

```
s5_median <- average_median(fit)
```

The median time is where the survival probability is equal to 0.5. Although there is no exact time where this occurs we know that the median exists between times 5 and 6. As the survival probabilities of the two times are nearly equal to 0.5 (which are .50 and .47 respectively)

Therefore, for size "self-employed", the near median is 5.5

Part 2

Since the previously defined function could only work for a single survival curve, therefore, this function is made.

The following function was reference from surv_median from the survminer package. Nonetheless, its the median survival with upper and lower confidence limits for the median at 95% confidence levels. So, I changed it to a way that it can compute the median at 90% CI instead.

```
median_at_90_percent <- function (fit, combine = FALSE)
{
    .median <- function(fit) {
        if (!is.null(fit$strata) | is.matrix(fit$surv)) {
            .table <- as.data.frame(summary(fit)$table)
        }
        else {
            .table <- t(as.data.frame(summary(fit)$table)) %>%
                as.data.frame()
            rownames(.table) <- "All"
        }
        .table$strata <- rownames(.table)
        .table <- .table %>% dplyr::select_(.dots = c("strata",
            "median", "`0.9LCL`", "`0.9UCL`"))
        colnames(.table) <- c("strata", "median",
            "lower", "upper")
        rownames(.table) <- NULL
        .table
    }
   .median(fit)
}
```

Function for plotting histogram

```
#create a function for plotting
plot_boot_data <- function(experiments, size, s_median){
  fit <- survfit(Surv(time_star, event_star) ~ experiment, data = experiments,conf.int= 0.9)
  #get the median of surv
  surv_med <- median_at_90_percent(fit)
  surv_med <- data.frame(surv_med)
  med <-  surv_med$median
  med <- data.frame(med)
  #get the upper CI
  upper <-  mean(surv_med$upper, na=T)
  #get the lower CI
```

```
  lower <- mean(surv_med$lower, na=T)
  ggplot(med , aes(x = med, fill= med)) +
    geom_histogram(binwidth = .8)+
    geom_vline(xintercept = upper, colour="blue",linetype="dashed")+
    geom_vline(xintercept = lower, colour="blue",linetype="dashed")+
    geom_vline(xintercept = s_median, colour="black")+
    ggtitle(  paste("The estimate of the median for", size))+
      labs(x= 'Median',
         y="Count")+
    theme_bw()

}
```

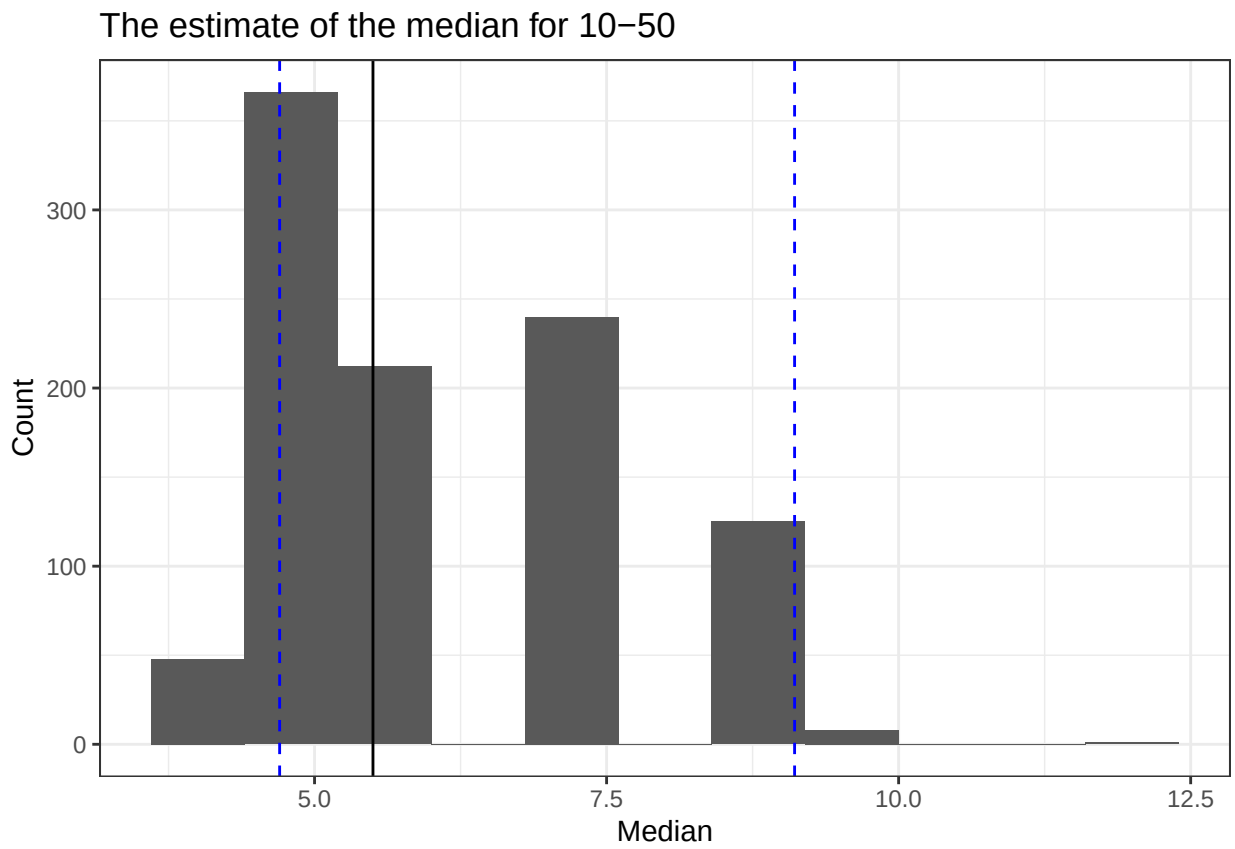
Bootstrap

```
#create a function of  the dataframe by sizes
boot <- function(size,n_sims){
#1. filter data into a particular size
df <- churn_dat %>% filter(company_size == size)
n <-  nrow(df)
#2. run the bootstrap
experiments <-  tibble(experiment = rep(1:n_sims, each = n),
                       index = sample(1:n, size = n * n_sims, replace = TRUE),
                       time_star = df$months_active[index],
                       event_star = df$churned[index])
return(experiments)
}
```

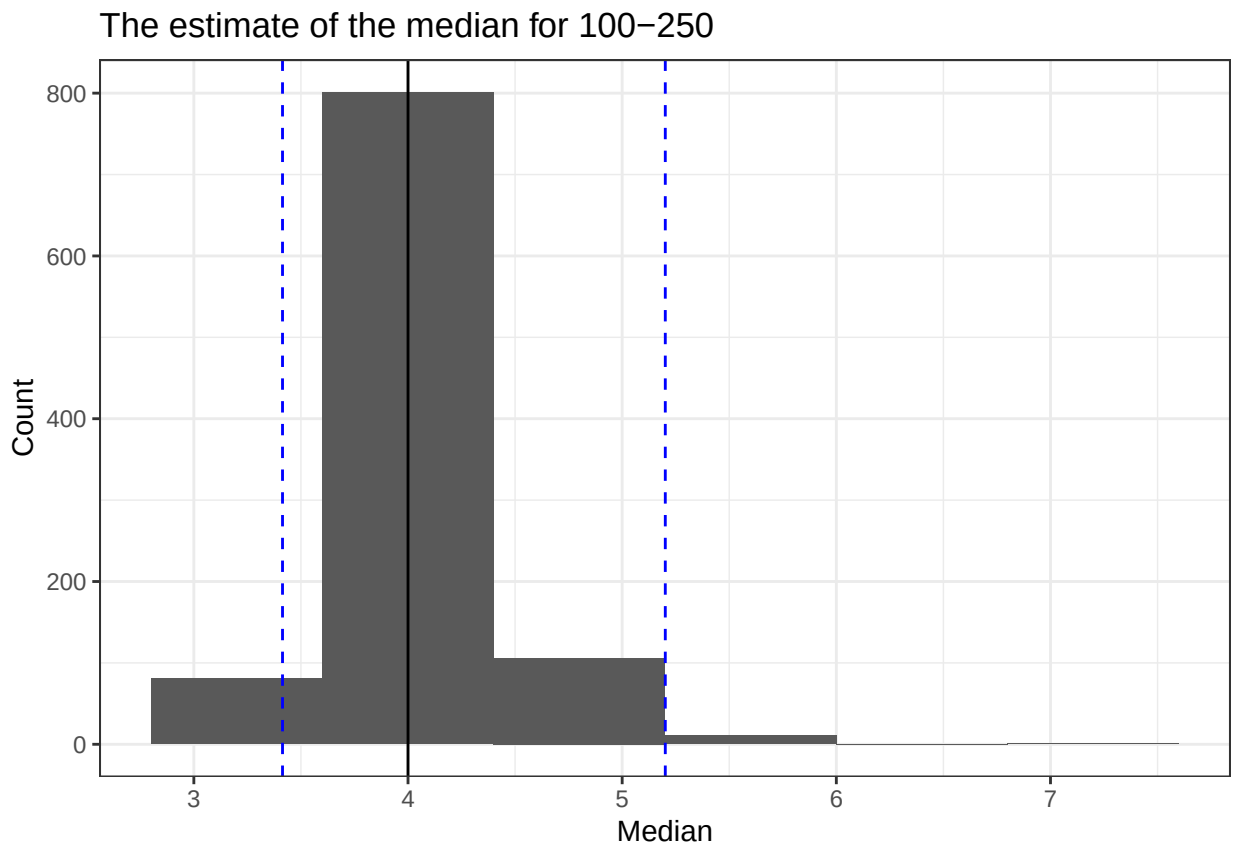
Histograms with confidence intervals for median

The graphs shown for each company size show the bootstrapped median data accompanied by the 90% confidence intervals for the said median in blue dashed lines, and the median found from the original churn data in a solid black line.

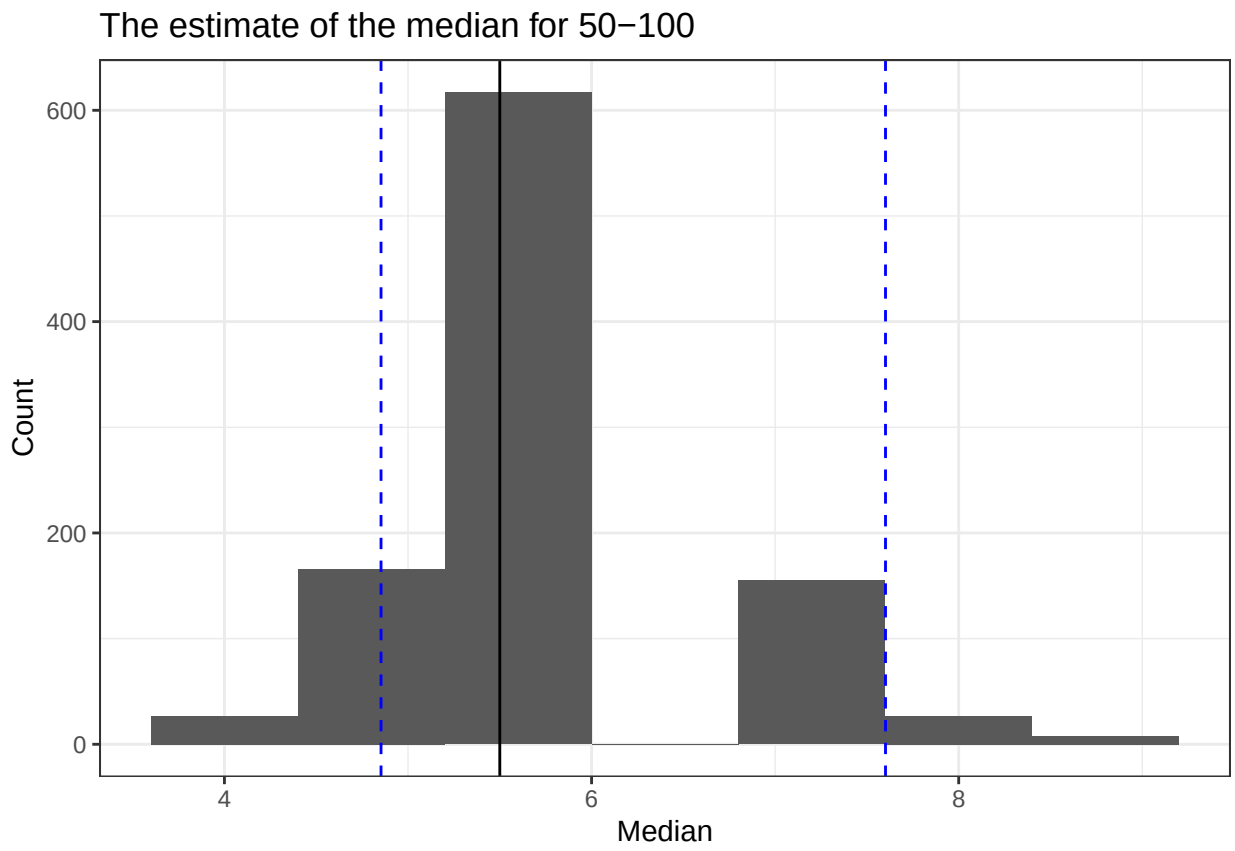
```
set.seed(999)
#"10-50"
df_10to50 <- boot("10-50",1000)
plot_boot_data(df_10to50, "10-50",s1_median)
```



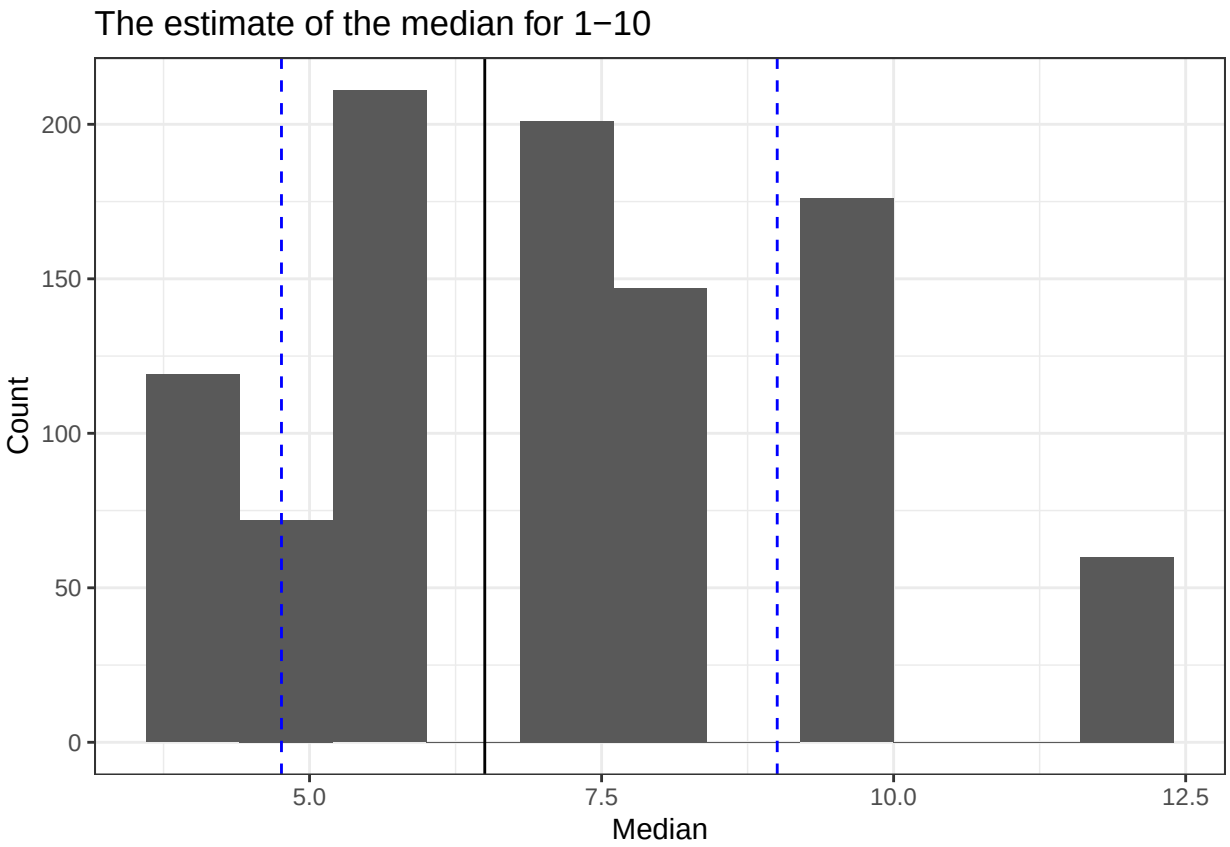

```
#"100-250"
df_100to250 <- boot("100-250",1000)
plot_boot_data(df_100to250,"100-250",s2_median)
```



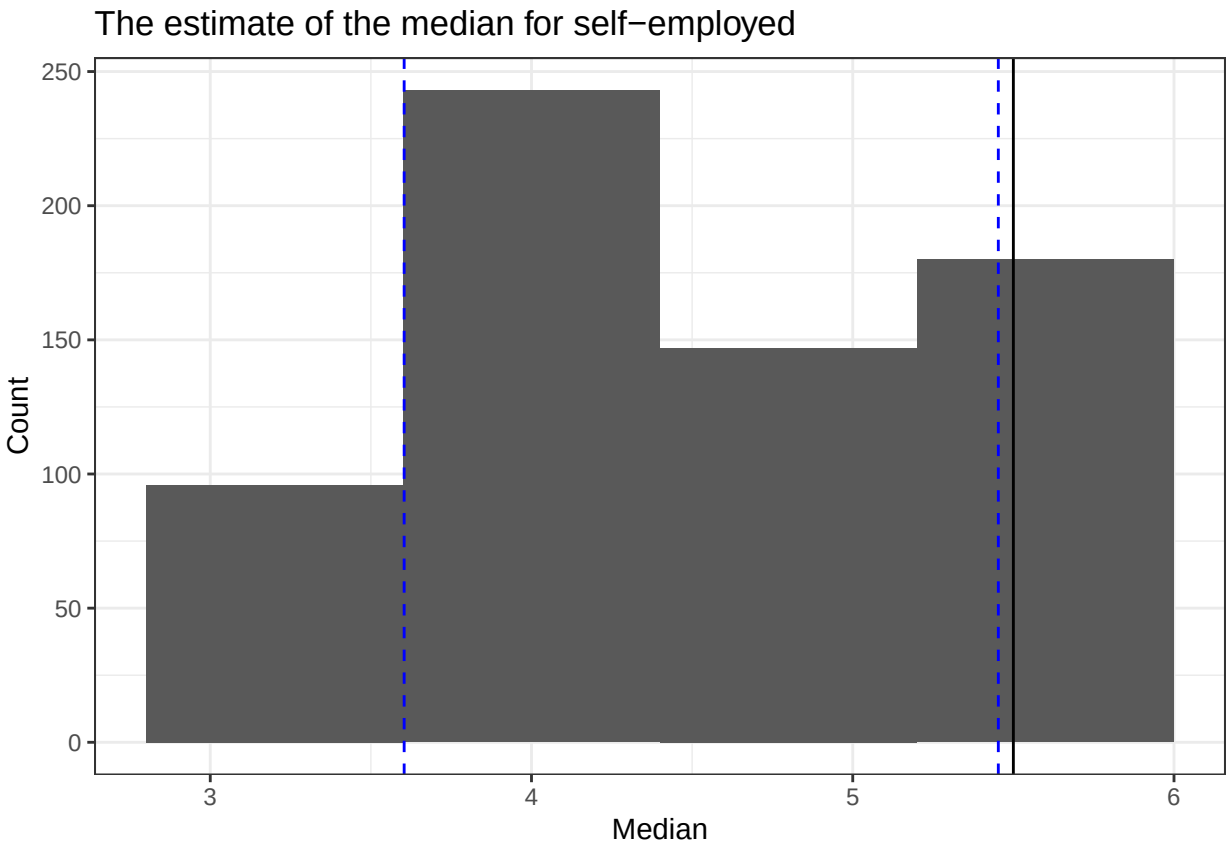

```
#"50-100"
df_50to100 <- boot("50-100",1000)
plot_boot_data(df_50to100,"50-100",s3_median)
```




```
#"1-10"
df_1to10 <- boot("1-10",1000)
plot_boot_data(df_1to10,"1-10",s4_median)
```




```
#"self-employed"
df_self_employed <- boot("self-employed",1000)
plot_boot_data(df_self_employed,"self-employed",s5_median)
```


The estimate of the median for self−employed
250
200
150
100
50
0
Count
3
4
5
6
Median

Overall Comment on Median Churn Time:

Overall, the median churn times across all five company sizes doesn't change a whole lot. For the three biggest sizes being; 10-50, 50-100, and 100-250, the median churn time was 5.5 months for all of them. This indicates that on average, they lose half of their customers within the first 5.5 months of gaining their business.

It would seem fitting that smaller client size businesses would have a greater capability to hold on to customers and hence have a larger median churn time. This rings true for the median of 6.5 for size 1-10, but does not hold for the self-employed companies who obtained a median churn time of 5. This was the only company size who's median was better calculated by the near-median method and due to the significant lack of observations compared to its larger size counterparts, could have a median that perhaps poorly represents the true population.

Although it may have the smallest median churn time, the self-employed company data actually sustained a steady level of survival probability from the median time onwards a lot better than the other sizes. Suggesting that although it may lose the same amount of customers (proportionately) a lot quicker than the other company sizes, it overall holds on to them a lot better in the long run.

Question 3

Filter data

```
churn_dat_50_100 <- churn_dat %>% filter(company_size == "50-100")

fit_hat <- survfit(Surv(months_active, churned) ~ 1, data = churn_dat_50_100) %>% tidy()
```

Bootstarp

```
set.seed(1888)
n = nrow(churn_dat_50_100)
n_sims = 10000

experiments = tibble(experiment = rep(1:n_sims, each = n),
                     index = sample(1:n, size = n * n_sims, replace = TRUE),
                     time_star = churn_dat_50_100$months_active[index],
                     event_star = churn_dat_50_100$churned[index])

bias <- experiments %>%
  group_by(experiment) %>%
  summarise(fortify(fit <- survfit(Surv(time_star, event_star) ~ 1)))
```

90% Confidence Intervals for each time point

```
dist_t1 <- bias %>% filter(time == 1)
dist_t1 <- tibble(est_star = dist_t1$surv,
```

```
                  est_hat = rep(fit_hat$estimate[1]),
                  delta_star = est_hat - est_star)
conf_t1 <- tibble(lower = fit_hat$estimate[1] + quantile(dist_t1$delta_star, 0.05),
                  upper = fit_hat$estimate[1] + quantile(dist_t1$delta_star, 0.95))

dist_t2 <- bias %>% filter(time == 2)
dist_t2 <- tibble(est_star = dist_t2$surv,
                  est_hat = rep(fit_hat$estimate[2]),
                  delta_star = est_hat - est_star)
conf_t2 <- tibble(lower = fit_hat$estimate[2] + quantile(dist_t2$delta_star, 0.05),
                  upper = fit_hat$estimate[2] + quantile(dist_t2$delta_star, 0.95))

dist_t3 <- bias %>% filter(time == 3)
dist_t3 <- tibble(est_star = dist_t3$surv,
                  est_hat = rep(fit_hat$estimate[3]),
                  delta_star = est_hat - est_star)
conf_t3 <- tibble(lower = fit_hat$estimate[3] + quantile(dist_t3$delta_star, 0.05),
                  upper = fit_hat$estimate[3] + quantile(dist_t3$delta_star, 0.95))

dist_t4 <- bias %>% filter(time == 4)
dist_t4 <- tibble(est_star = dist_t4$surv,
                  est_hat = rep(fit_hat$estimate[4]),
                  delta_star = est_hat - est_star)
conf_t4 <- tibble(lower = fit_hat$estimate[4] + quantile(dist_t4$delta_star, 0.05),
                  upper = fit_hat$estimate[4] + quantile(dist_t4$delta_star, 0.95))

dist_t5 <- bias %>% filter(time == 5)
dist_t5 <- tibble(est_star = dist_t5$surv,
                  est_hat = rep(fit_hat$estimate[5]),
                  delta_star = est_hat - est_star)
conf_t5 <- tibble(lower = fit_hat$estimate[5] + quantile(dist_t5$delta_star, 0.05),
                  upper = fit_hat$estimate[5] + quantile(dist_t5$delta_star, 0.95))

dist_t6 <- bias %>% filter(time == 6)
dist_t6 <- tibble(est_star = dist_t6$surv,
                  est_hat = rep(fit_hat$estimate[6]),
                  delta_star = est_hat - est_star)
conf_t6 <- tibble(lower = fit_hat$estimate[6] + quantile(dist_t6$delta_star, 0.05),
                  upper = fit_hat$estimate[6] + quantile(dist_t6$delta_star, 0.95))

dist_t7 <- bias %>% filter(time == 7)
dist_t7 <- tibble(est_star = dist_t7$surv,
                  est_hat = rep(fit_hat$estimate[7]),
                  delta_star = est_hat - est_star)
conf_t7 <- tibble(lower = fit_hat$estimate[7] + quantile(dist_t7$delta_star, 0.05),
                  upper = fit_hat$estimate[7] + quantile(dist_t7$delta_star, 0.95))

dist_t8 <- bias %>% filter(time == 8)
dist_t8 <- tibble(est_star = dist_t8$surv,
                  est_hat = rep(fit_hat$estimate[8]),
                  delta_star = est_hat - est_star)
conf_t8 <- tibble(lower = fit_hat$estimate[8] + quantile(dist_t8$delta_star, 0.05),
                  upper = fit_hat$estimate[8] + quantile(dist_t8$delta_star, 0.95))
```

```
dist_t9 <- bias %>% filter(time == 9)
dist_t9 <- tibble(est_star = dist_t9$surv,
                  est_hat = rep(fit_hat$estimate[9]),
                  delta_star = est_hat - est_star)
conf_t9 <- tibble(lower = fit_hat$estimate[9] + quantile(dist_t9$delta_star, 0.05),
                  upper = fit_hat$estimate[9] + quantile(dist_t9$delta_star, 0.95))

dist_t11 <- bias %>% filter(time == 11)
dist_t11 <- tibble(est_star = dist_t11$surv,
                  est_hat = rep(fit_hat$estimate[10]),
                  delta_star = est_hat - est_star)
conf_t11 <- tibble(lower = fit_hat$estimate[10] + quantile(dist_t11$delta_star, 0.05),
                   upper = fit_hat$estimate[10] + quantile(dist_t11$delta_star, 0.95))

dist_t12 <- bias %>% filter(time == 12)
dist_t12 <- tibble(est_star = dist_t12$surv,
                  est_hat = rep(fit_hat$estimate[11]),
                  delta_star = est_hat - est_star)
conf_t12 <- tibble(lower = if(fit_hat$estimate[11] + quantile(dist_t12$delta_star, 0.05) > 0){
  print(fit_hat$estimate[11] + quantile(dist_t12$delta_star, 0.05))
} else {
  0
},

                   upper = fit_hat$estimate[11] + quantile(dist_t12$delta_star, 0.95))
```

Table of confidence intervals

t	lower	upper
1	0.9107143	0.9434524
2	0.7543467	0.8078102
3	0.5921235	0.6552333
4	0.5062992	0.5736682
5	0.4844446	0.5533244
6	0.4437245	0.5192588
7	0.4214538	0.4999542
8	0.3993802	0.4826334
9	0.3534004	0.4547156
11	0.3533915	0.4547169
12	0.0000000	0.4026571

Comparing the confidence intervals to the confidence intervals for the median in questioin 2, we can see that the value of 0.5 could possibly land at a time value of either 5 or 6 which is consistant with the results of the confidence interval for the median in question 2.

Coverage of Confidence Intervals, the probability that the true survival function lies inside all of the CI bounds for all values of t

```r
coverage <- mean(dist_t1$est_star > conf_t1$lower[1] & dist_t1$est_star < conf_t1$upper[1] &
       dist_t2$est_star > conf_t2$lower[1] & dist_t2$est_star < conf_t2$upper[1] &
       dist_t3$est_star > conf_t3$lower[1] & dist_t3$est_star < conf_t3$upper[1] &
       dist_t4$est_star > conf_t4$lower[1] & dist_t4$est_star < conf_t4$upper[1] &
       dist_t5$est_star > conf_t5$lower[1] & dist_t5$est_star < conf_t5$upper[1] &
       dist_t6$est_star > conf_t6$lower[1] & dist_t6$est_star < conf_t6$upper[1] &
       dist_t7$est_star > conf_t7$lower[1] & dist_t7$est_star < conf_t7$upper[1] &
       dist_t8$est_star > conf_t8$lower[1] & dist_t8$est_star < conf_t8$upper[1] &
       dist_t9$est_star > conf_t9$lower[1] & dist_t9$est_star < conf_t9$upper[1] &
       dist_t11$est_star > conf_t11$lower[1] & dist_t11$est_star < conf_t11$upper[1] &
       dist_t12$est_star > conf_t12$lower[1] & dist_t12$est_star < conf_t12$upper[1])
print(coverage)
```

```
## [1] 0.4674
```

The coverage or in other words the probability that the true survival function is contained within these confidence intervals for each value of time for a company size of 50-100 is 0.4674. This is because we have computed a 90% confidence interval for each value of t, that is the probability that the true parameter for that t value falls within that interval. The more intervals you have the lower the probability that the true survival functions is contained purely within those bounds.

Question 4

Filtering data

```r
churn_dat_50_100 <- churn_dat %>% filter(company_size == "50-100")
churn_dat_50_100 <- tibble(months_active = churn_dat_50_100$months_active,
                           churned = churn_dat_50_100$churned)

churn_dat_100_250 <- churn_dat %>% filter(company_size == "100-250")
churn_dat_100_250 <- tibble(months_active = churn_dat_100_250$months_active,
                           churned = churn_dat_100_250$churned)
```

Creating a matrix to smaple from

```r
x <- Surv(churn_dat_50_100$months_active, churn_dat_50_100$churned)
y <- Surv(churn_dat_100_250$months_active, churn_dat_100_250$churned)

z <-  c(x, y)
```

Function for computing the test-statistic

```r
z_stat <- function(var1, var2){
```

```
e = fit_x$n * (var1$d + var2$d)/(fit_x$n + fit_y$n)

v = e*((var1$n + var2$n - var1$d - var2$d)/(var1$n + var2$n))* (var2$n/(var1$n + var2$n - 1))

z = ((sum(var1$d - e))/(sqrt(sum(v))))

return(z)
}
```

Loop to repeat permutation samples from Z

```
set.seed(1888)
n = 10000
z_stat_matrix <- matrix(0, ncol = n)
for(i in 1:n)
  {
nx <- length(x)
ny <- length(y)

z_star <- sample(z, replace = FALSE)

x_star <- z_star[1:nx]
y_star <- z_star[(nx + 1):(nx + ny)]

fit_x1 <- survfit(x_star ~ 1) %>% tidy()
fit_x <- tibble(n = fit_x1$n.risk,
                d = fit_x1$n.event)
fit_y1 <- survfit(y_star ~ 1) %>% tidy()
fit_y <- tibble(n = fit_y1$n.risk,
                d = fit_y1$n.event)

s_stats = z_stat(fit_x, fit_y)

z_stat_matrix[,i] <- s_stats
}
```

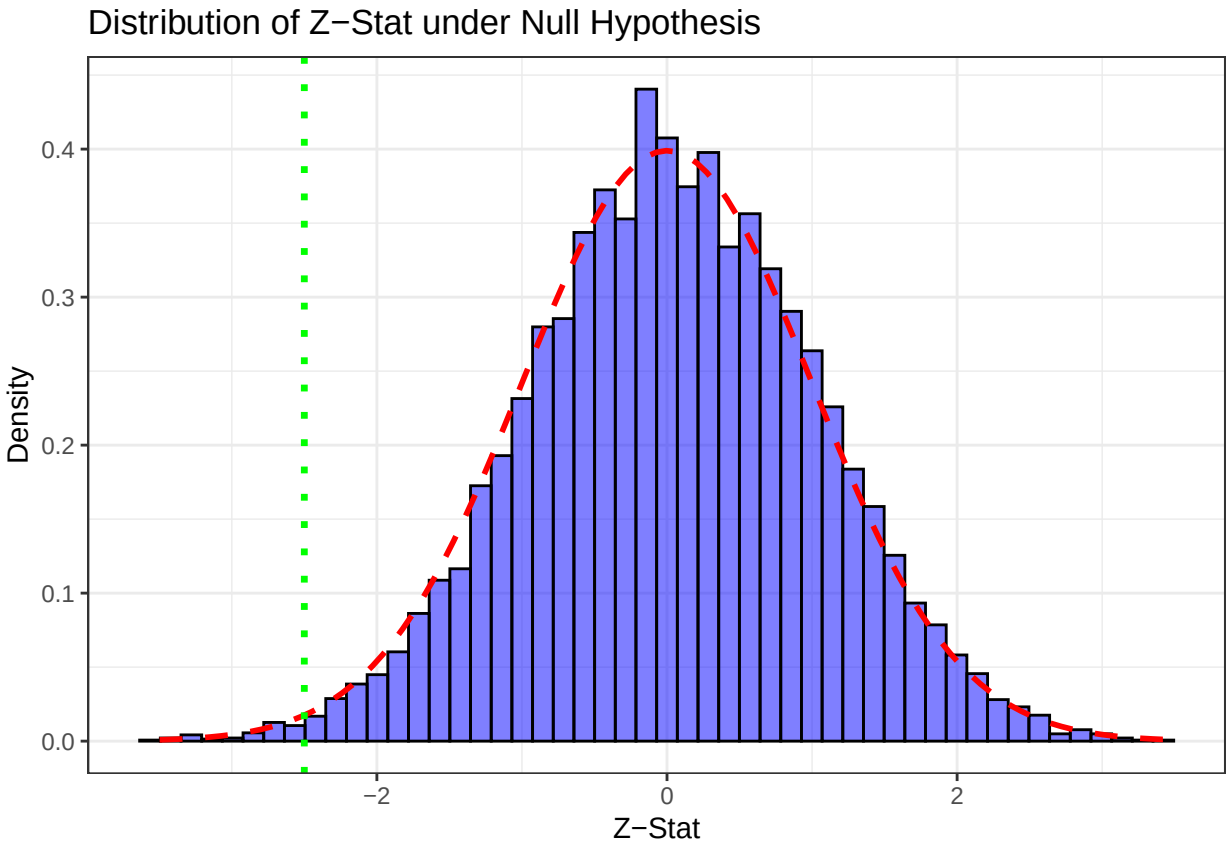
```
x_obs <- survfit(x ~ 1) %>% tidy()
x_obs <- tibble(n = x_obs$n.risk,
                d = x_obs$n.event)
y_obs <- survfit(y ~ 1) %>% tidy()
y_obs <- tibble(n = y_obs$n.risk,
                d = y_obs$n.event)
obs_test_stat <- z_stat(x_obs, y_obs)
```

Graph of results

As can be seen from the graph below whem the distribution of the t-stat is scaled to a PDF is approaches a N(0, 1) distribution which is shown by the red dashed line, in addition our observed t-stat falls quite far to the left at a value of -2.5 . This information alone leads us to believe that the survival curves of company size 50-100 and 100-250 are different.

```
z_plot <- data.frame(data = t(z_stat_matrix))

z_plot %>% ggplot(aes(x = data)) + geom_histogram(aes(y = after_stat(density)),
                                                  bins = 50,
                                                  colour = "black",
                                                  fill = "blue",
                                                  alpha = 0.5) +
  stat_function(fun = dnorm, colour = "red", linetype = "dashed", size = 1) +
  ggtitle("Distribution of Z-Stat under Null Hypothesis") +
  xlab("Z-Stat") + ylab("Density") +
  geom_vline(xintercept =  obs_test_stat, linetype = "dotted", colour = "green", size = 1.2) +
  theme_bw()
```



Permutation confidence interval

The permutation 95% confidence interval finds the the values for which 2.5% of the data is below and 2.5% of the data is above.

```
perm_conf_int <- tibble(lower = quantile(z_stat_matrix, 0.025),
                        upper = quantile(z_stat_matrix, 0.975))
kable(perm_conf_int)
```

lower	upper
-1.90813	1.959091

CLT Confidence Interval

The Central limit theorem confidence interval states that for a normal distribution with a given mean and standard deviation, that 95% of the values lie between 1.96 time the standard deviation above and below the mean.

```
clt_mean <- mean(z_stat_matrix)
clt_sd <- sd(z_stat_matrix)
clt_conf_int <- tibble(lower = qnorm(0.025, mean = clt_mean, sd = clt_sd),
                       upper = qnorm(0.975, mean = clt_mean, sd = clt_sd))
kable(clt_conf_int)
```

lower	upper
-1.894334	1.9777

As can be seen there is slight differences in the clt confidence interval and the permutation confidence interval which is expected. This is due to the fact that with a large enough number of samples under the null hypotheis, that being the survival curves are the same, thus come from the same set of data the distribution of the t-stat approaches a normal distribution with a mean of 0 and standard deviation of 1. Our observed test statistic between company sizes of 50-100 and 100-250 was -2.5 which is outside both the permutation and clt confidence interval. We can treat these confidence intervals as t-stats for two tailed hypothesis test at a 5% significance level. As our observed test statistic was -2.5 we can conclude that we have sufficient evidence to reject the null hypothesis in favour of the alternative hypothesis. Where the null is that they have the same survival curves and the alternative being they have different survival curves.