ETC5242Assignment

Sahinya Akila

9/4/2021

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## Remove the line break in the file name!
churn_dat <- read_csv("https://raw.githubusercontent.com/square/pysurvival/master/pysurvival/datasets/cl
```

```
## Rows: 2000 Columns: 14

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): product_travel_expense, product_payroll, product_accounting, compan...
## dbl (9): product_data_storage, csat_score, articles_viewed, smartphone_notif...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
churn_dat <- churn_dat %>% filter(months_active > 0) %>% select(c(months_active, churned, company_size))
```

```
km_model <- function(time, event){
  dataset <- data_frame(time, event)

  km_data <- dataset %>%
    group_by(time, event) %>%
    summarise(died = n()) %>%
    ungroup() %>%
    mutate(risk = nrow(dataset) - accumulate(died, `+`) + died) %>%
    filter(event == 1) %>%
```

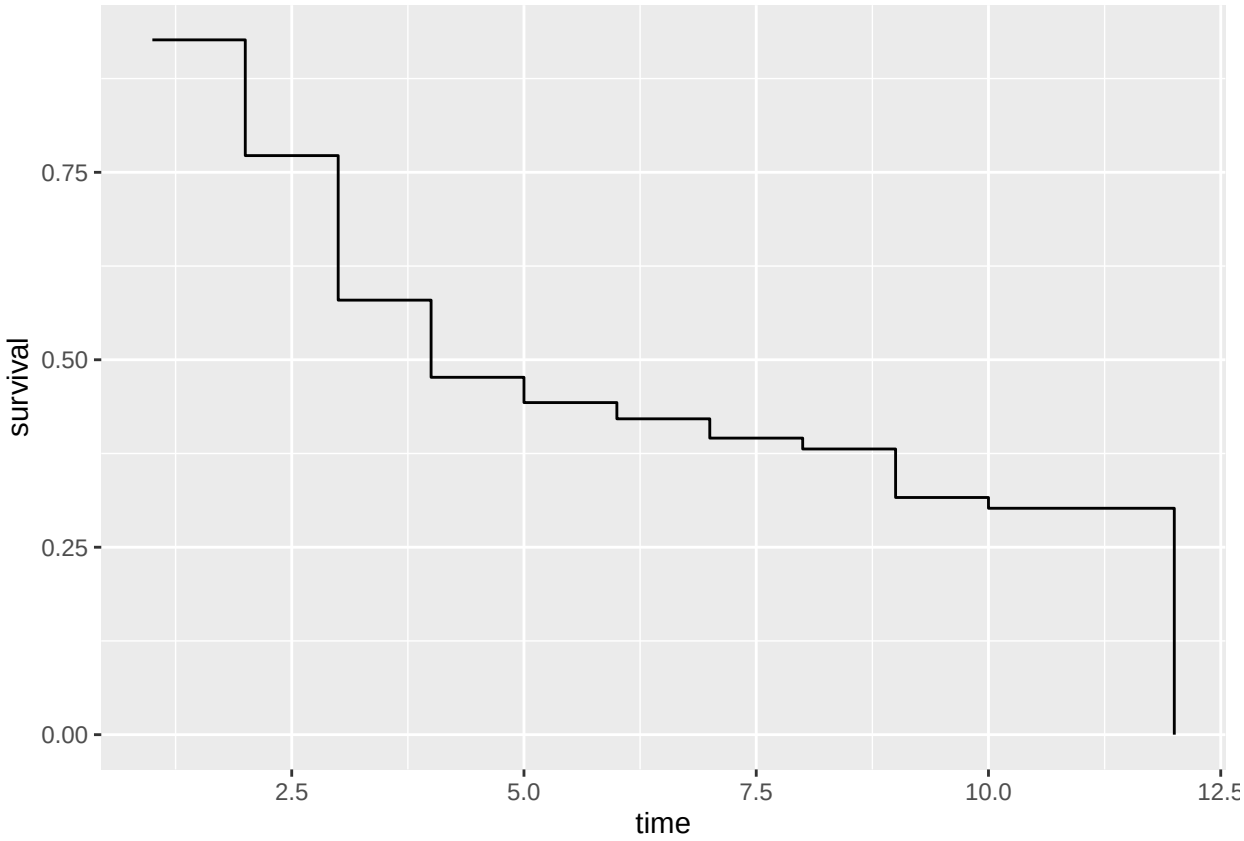
```
    mutate(probability = 1 - died/risk,
           survival = accumulate(probability, `*`))
  return(km_data %>% select(time, survival))
}

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

```
## Warning: ‘data_frame()‘ was deprecated in tibble 1.1.0.
## Please use ‘tibble()‘ instead.
```

```
## ‘summarise()‘ has grouped output by ’time’. You can override using the ‘.groups‘ argument.
```

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

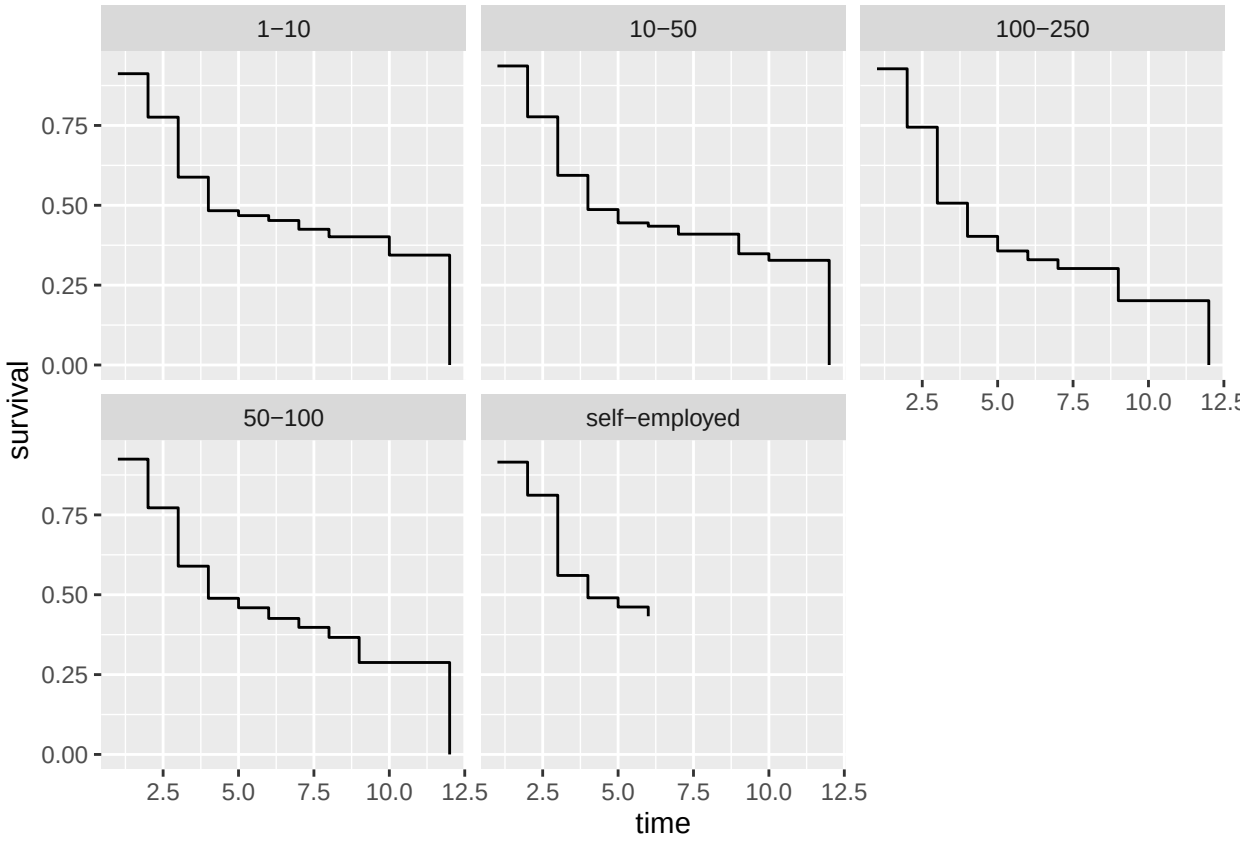



```
company_km_model <- data.frame(time = double(), survival = double(), company_size = character())
for(size in unique(churn_dat$company_size)){
  filtered <- churn_dat %>% filter(company_size == size)
  final_model <- km_model(filtered$months_active, filtered$churned) %>% mutate(company_size = size)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

```
## ‘summarise()‘ has grouped output by ’time’. You can override using the ‘.groups‘ argument.
```

```
## `summarise()` has grouped output by 'time'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'time'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'time'. You can override using the `.groups` argument.
## `summarise()` has grouped output by 'time'. You can override using the `.groups` argument.
```

```
company_km_model %>%
  ggplot(aes(time, survival)) +
  geom_step() +
  facet_wrap(~company_size)
```



Q2

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

```
library(survival)
fit <- survfit(Surv(months_active, churned) ~ 1, data = churn_dat)
event_times <- fit$time
kaplan_meier <- fit$surv
```

```
median(Surv(churn_dat$months_active, churn_dat$churned))
```

```
## $quantile
## 50
##  6
```

```
##
## $lower
## 50
##  5
##
## $upper
## 50
##  7
```

```
sd(Surv(churn_dat$months_active, churn_dat$churned))
```

```
## [1] 2.432778
```

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

```
summary(fit)
```

```
## Call: survfit(formula = Surv(months_active, churned) ~ 1, data = churn_dat)
##
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1   1958     140    0.928 0.00582        0.917        0.940
##     2   1769     281    0.781 0.00944        0.763        0.800
##     3   1406     302    0.613 0.01132        0.591        0.636
##     4    908     127    0.527 0.01203        0.504        0.552
##     5    588      28    0.502 0.01235        0.479        0.527
##     6    368      18    0.478 0.01304        0.453        0.504
##     7    350      13    0.460 0.01345        0.434        0.487
##     8    200       4    0.451 0.01395        0.424        0.479
##     9    105       9    0.412 0.01773        0.379        0.448
##    10     44       2    0.393 0.02130        0.354        0.438
##    12     21       8    0.244 0.04373        0.171        0.346
```

```
y = rnorm(nrow(churn_dat), mean = mean(churn_dat$months_active), sd = sd(churn_dat$months_active))
bootstrap <- tibble(experiment = rep(1:10000, each = 100),
ind = sample(1:100, size = 100*10000, replace = TRUE),
ystar = y[ind])
bias <- bootstrap %>%
group_by(experiment) %>%
summarise(delta = median(y) - median(ystar))
median(y) + quantile(bias$delta, c(0.05, 0.95))
```

```
##       5%      95%
## 3.281716 4.437346
```

```
companysize <- c("self-employed", "1-10", "10-50", "50-100", "100-250")
comp_boot <- function(companysize) {
  comp_data <- churn_dat %>%
    filter(company_size == companysize)
y = rnorm(nrow(comp_data), mean = mean(comp_data$months_active), sd = sd(comp_data$months_active))
bootstrap <- tibble(experiment = rep(1:10000, each = 100),
ind = sample(1:100, size = 100*10000, replace = TRUE),
```

```
ystar = y[ind])
bias <- bootstrap %>%
group_by(experiment) %>%
summarise(delta = median(y) - median(ystar))
median(y) + quantile(bias$delta, c(0.05, 0.95), na.rm = TRUE)

}
a2 = map(companysize, comp_boot)
a2
```

```
## [[1]]
##  5% 95%
##  NA  NA
##
## [[2]]
##       5%      95%
## 3.276229 4.733812
##
## [[3]]
##       5%      95%
## 3.043262 4.068507
##
## [[4]]
##       5%      95%
## 3.863457 4.560358
##
## [[5]]
##       5%      95%
## 3.350368 4.565762
```

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

```
fit1 <- survfit(Surv(months_active, churned) ~ company_size, data = churn_dat)
event_times <- fit$time
kaplan_meier <- fit$surv

summary(fit1)
```

```
## Call: survfit(formula = Surv(months_active, churned) ~ company_size,
##     data = churn_dat)
##
##                 company_size=1-10
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1    311      27    0.913  0.0160        0.882        0.945
##     2    280      40    0.783  0.0235        0.738        0.830
##     3    228      46    0.625  0.0280        0.572        0.682
##     4    144      20    0.538  0.0301        0.482        0.600
##     5     92       2    0.526  0.0306        0.470        0.590
##     6     61       2    0.509  0.0319        0.450        0.576
##     7     59       2    0.492  0.0331        0.431        0.561
##     8     31       1    0.476  0.0356        0.411        0.551
```

```
##    10      7       1    0.408  0.0700        0.292        0.571
##    12      2       1    0.204  0.1484        0.049        0.849
##
##                company_size=10-50
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1    673      42    0.938 0.00932        0.919        0.956
##     2    617      99    0.787 0.01591        0.757        0.819
##     3    483     100    0.624 0.01923        0.588        0.663
##     4    324      46    0.536 0.02046        0.497        0.577
##     5    209      12    0.505 0.02113        0.465        0.548
##     6    128       3    0.493 0.02171        0.452        0.537
##     7    125       4    0.477 0.02240        0.435        0.523
##     9     35       3    0.436 0.03048        0.380        0.500
##    10     17       1    0.411 0.03799        0.343        0.492
##    12     11       3    0.299 0.06168        0.199        0.448
##
##                company_size=100-250
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1    240      17   0.9292  0.0166        0.897        0.962
##     2    217      41   0.7536  0.0281        0.700        0.811
##     3    167      46   0.5460  0.0331        0.485        0.615
##     4     98      16   0.4569  0.0344        0.394        0.529
##     5     62       5   0.4200  0.0353        0.356        0.495
##     6     39       3   0.3877  0.0372        0.321        0.468
##     7     36       2   0.3662  0.0381        0.299        0.449
##     9     11       3   0.2663  0.0565        0.176        0.403
##    12      3       2   0.0888  0.0749        0.017        0.464
##
##                company_size=50-100
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1    672      49    0.927  0.0100       0.9076        0.947
##     2    601      95    0.781  0.0162       0.7495        0.813
##     3    481      97    0.623  0.0193       0.5865        0.662
##     4    313      42    0.540  0.0205       0.5007        0.581
##     5    204       8    0.518  0.0211       0.4787        0.561
##     6    124       9    0.481  0.0230       0.4378        0.528
##     7    115       5    0.460  0.0238       0.4155        0.509
##     8     71       3    0.440  0.0253       0.3935        0.493
##     9     35       3    0.403  0.0311       0.3460        0.469
##    12      4       2    0.201  0.1019       0.0747        0.543
##
##                company_size=self-employed
##  time n.risk n.event survival std.err lower 95% CI upper 95% CI
##     1     62       5    0.919  0.0346        0.854        0.990
##     2     54       6    0.817  0.0499        0.725        0.921
##     3     47      13    0.591  0.0644        0.478        0.732
##     4     29       3    0.530  0.0667        0.414        0.678
##     5     21       1    0.505  0.0681        0.387        0.658
##     6     16       1    0.473  0.0708        0.353        0.635
```

```
library(survminer)
```

```
## Loading required package: ggpubr
```

```
library(ggplot2)
ggsurvplot(fit1, conf.int = TRUE, pval = TRUE)
```