20211010_first_check_in

Jinglun Li; Fuyu Guo; Xinhao Li

10/10/2021

```
library(tidyverse)
library(survival)
library(ggpubr)
library(survminer)
```

```
#############################
# load data
dta <- read.csv("heart_f.csv")

# select variables we will use
dta <- dplyr::select(dta,
                     "age", "sex", "anaemia",
                     "diabetes", "ejection_fraction", "smoking",
                     "platelets", "serum_creatinine", "serum_sodium",
                     "time", "DEATH_EVENT")

# rename the variables to make our work easier
names(dta) <- c("age", "sex", "anemia",
                     "dbt", "ef", "smoking",
                     "plat", "ser_crt", "ser_na",
                     "time", "death")

# check sample size
dim(dta)
```

```
## [1] 299  11
```

```
# check if there is any missing value in variables
complete.cases(dta) %>% all()
```

```
## [1] TRUE
```

```
# no missing value



########################################################
########################################################
# check exposure distribution
summary(dta$ser_crt)
```

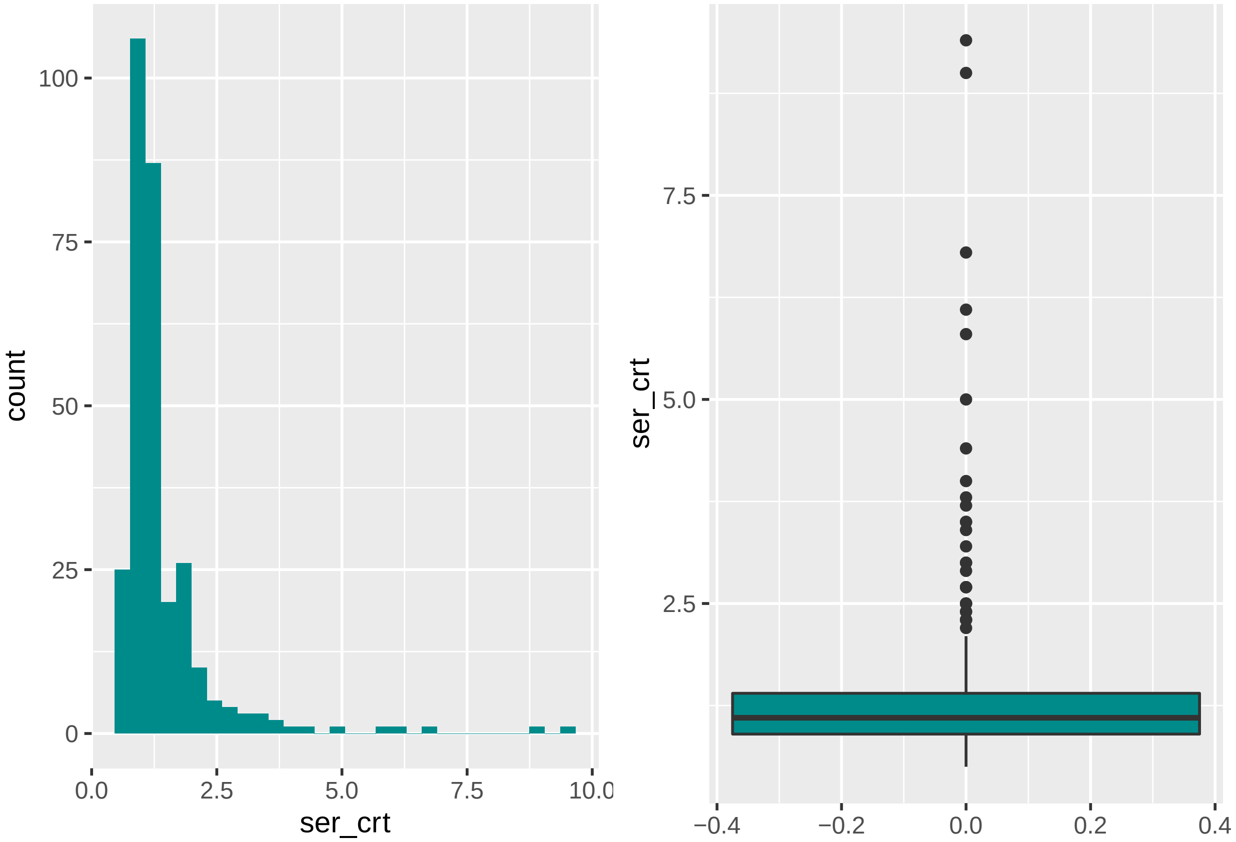
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.500   0.900   1.100   1.394   1.400   9.400
```

```
sd(dta$ser_crt)
```

```
## [1] 1.03451
```

```
p1 <- ggplot(dta) +
        geom_histogram(aes(x = ser_crt), fill = "DarkCyan")
p2 <- ggplot(dta) +
        geom_boxplot(aes(y = ser_crt), fill = "DarkCyan")
ggarrange(p1, p2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggsave("20211001_exposure_distribution.png", width = 150, height = 80,
       units = "mm")


# calculate the normal level proportion
dta$ser_crt_group <- if_else(dta$ser_crt <= 1.5, "normal", "abnormal")
dta$ser_crt_group %>% table
```

```
## .
## abnormal   normal
##       67      232
```

```
dta$ser_crt_group %>% table %>% prop.table()
```

```
## .
##  abnormal    normal
## 0.2240803 0.7759197
```

```r
#######################################################
#######################################################
# check total person-time
sum(dta$time)
```

```
## [1] 38948
```

```r
summary(dta$time)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     4.0    73.0   115.0   130.3   203.0   285.0
```

```r
sd(dta$time)
```

```
## [1] 77.61421
```

```r
# check time to the event among patients who died finally
summary(dta$time[dta$death == 1])
```

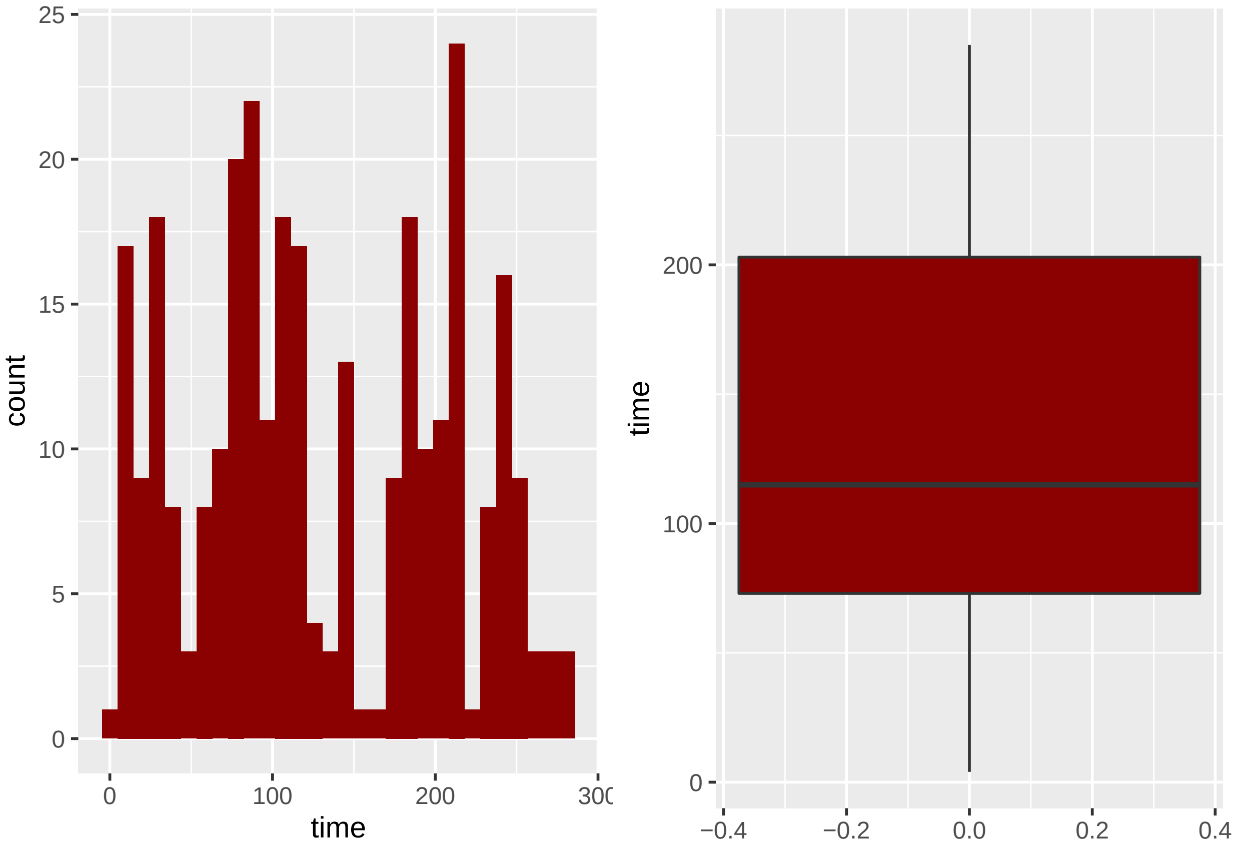
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4.00   25.50   44.50   70.89  102.25  241.00
```

```r
sd(dta$time[dta$death == 1])
```

```
## [1] 62.37828
```

```r
# make plots
p1 <- ggplot(dta) +
        geom_histogram(aes(x = time), fill = "DarkRed")
p2 <- ggplot(dta) +
        geom_boxplot(aes(y = time), fill = "DarkRed")
ggarrange(p1, p2)
```

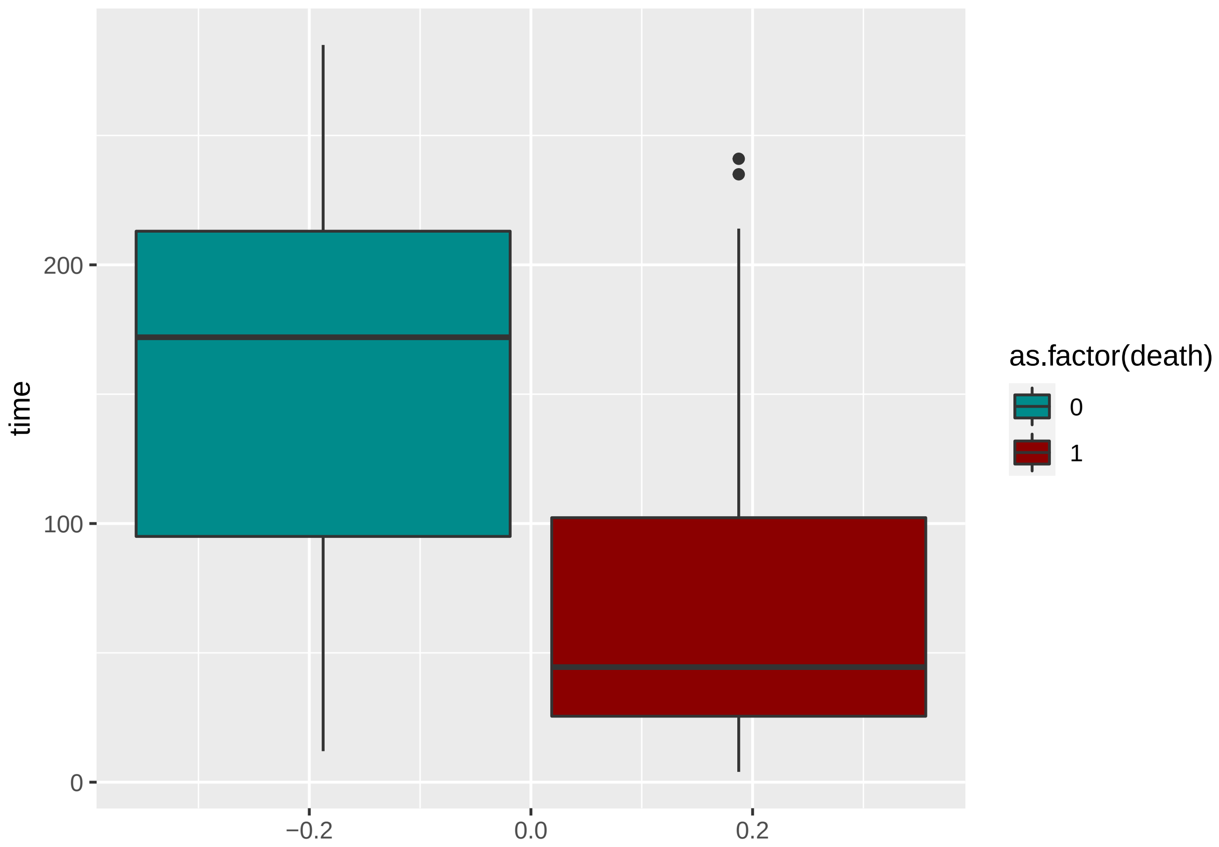
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggsave("20211001_death_distribution.png", width = 150, height = 80,
       units = "mm")

# make a plot by death = 0

ggplot(dta) +
        geom_boxplot(aes(y = time, group = death, fill = as.factor(death))) +
        scale_fill_manual(values = c("DarkCyan", "DarkRed"))
```



```
ggsave("20211001_death_distribution_by_death.png", width = 150, height = 80,
       units = "mm")



# calculate the proportion of death in 30 days
dta$death_30 <- NA
dta$death_30[dta$death == 1& dta$time <= 30] <- "Yes"
dta$death_30[(dta$death == 1& dta$time > 30) |
               (dta$death == 0& dta$time > 30)] <- "No"
dta$death_30[dta$death == 0& dta$time <= 30] <- "Censored"
table(dta$death_30)
```

```
##
## Censored       No      Yes
##        5      259       35
```

```
table(dta$death_30) %>% prop.table()
```

```
##
##   Censored         No        Yes
## 0.01672241 0.86622074 0.11705686
```

```
##############################################
#############################################
# Check Characteristics of the 299 patient by serum creatinine level
#  Table 1 #
# check continuous variables
```

```
re1 <- dta %>% group_by(ser_crt_group) %>%
        summarise(avg_crt = mean(ser_crt),
                  sd_crt = sd(ser_crt),
                  avg_time = mean(time),
                  sd_time = sd(time),
                  avg_age = mean(age),
                  sd_age = sd(age),
                  avg_plat = mean(plat),
                  sd_plat = sd(plat),
                  avg_na = mean(ser_na),
                  sd_na = sd(ser_na)) %>%
        t()


re1 <- re1[, c(2,1)]
re1
```

```
##               [,1]        [,2]
## ser_crt_group "normal"    "abnormal"
## avg_crt       "1.025991"  "2.667761"
## sd_crt        "0.2071469" "1.5996475"
## avg_time      "136.3362"  "109.2239"
## sd_time       "75.80768"  "80.66162"
## avg_age       "59.63937"  "64.97015"
## sd_age        "11.49359"  "12.41330"
## avg_plat      "265270.8"  "256734.7"
## sd_plat       " 96762.64" "101796.45"
## avg_na        "137.3233"  "134.2090"
## sd_na         "3.609004"  "5.889223"
```

```
# t-test for continuous variables
lapply(c("ser_crt", "time", "age", "plat", "ser_na"),
       function (x){
               dta$x <- dta[,x]
               t.test(x~ser_crt_group, data = dta)$p.value %>% return()
       })
```

```
## [[1]]
## [1] 5.15281e-12
##
## [[2]]
## [1] 0.01574423
##
## [[3]]
## [1] 0.002168052
##
## [[4]]
## [1] 0.5423805
##
## [[5]]
## [1] 9.389263e-05
```

```
###########################################################
# categorical variables
```

```r
# rename the categories
dta$ser_crt_group <- factor(dta$ser_crt_group , levels = c("normal", "abnormal"))
dta$sex <- if_else(dta$sex == 1, "male", "female")
dta$ef[dta$ef <= 30] <- "<=30"
dta$ef[dta$ef > 30 & dta$ef < 45] <- "41-44"
dta$ef[dta$ef >= 45] <- ">=45"


# check the proportion
lapply(c("death", "death_30", "sex", "smoking", "anemia", "dbt",
         "ef"),
       function(x){
        s1 <- table(dta[,x], dta$ser_crt_group)
        s2 <- table(dta[,x], dta$ser_crt_group) %>% as.matrix()
        s2 <- cbind(s2[,1]/sum(dta$ser_crt_group == "normal"),  s2[,1]/sum(dta$ser_crt_group == "abnorma
        s3 <- table(dta[,x], dta$ser_crt_group) %>% chisq.test()
        list(s1, s2, s3) %>% return()
         })
```

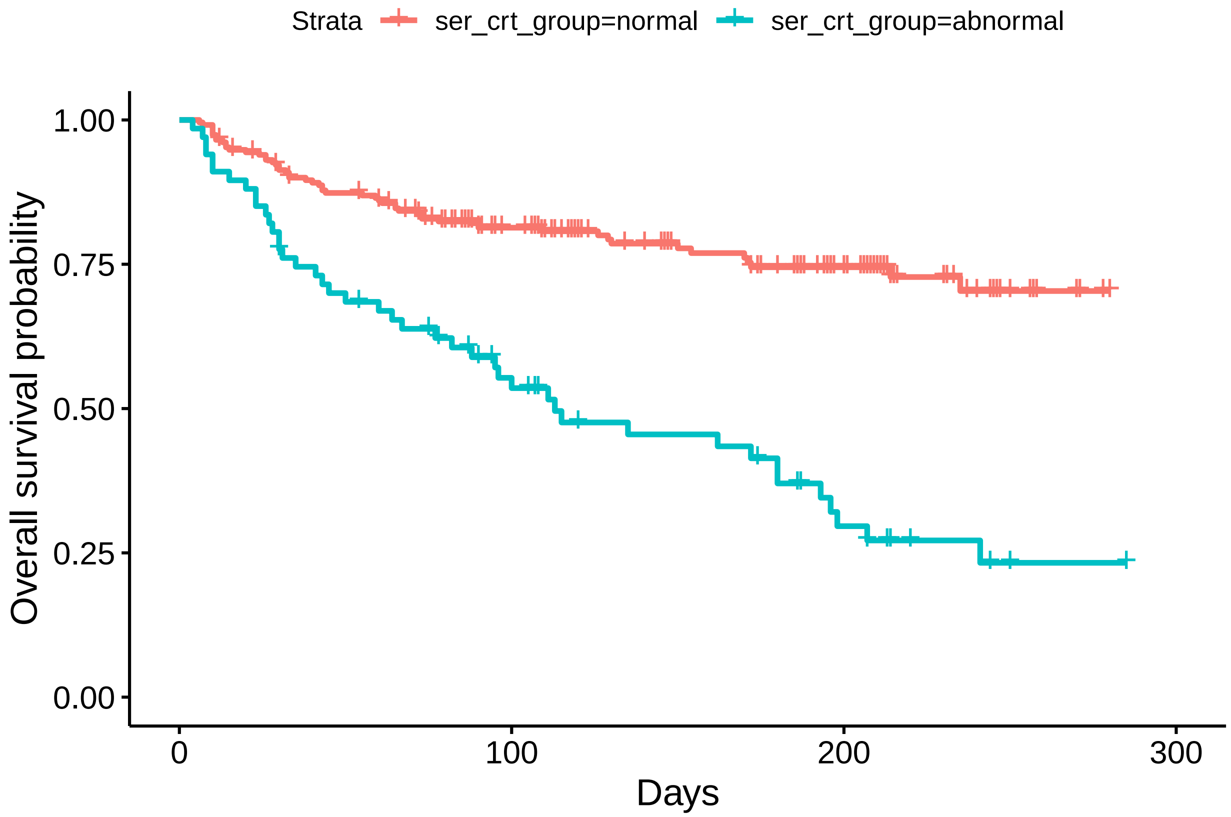
```
## Warning in chisq.test(.): Chi-squared approximation may be incorrect
```

```
## [[1]]
## [[1]][[1]]
##
##     normal abnormal
##   0    179       24
##   1     53       43
##
## [[1]][[2]]
##        [,1]      [,2]
## 0 0.7715517 2.6716418
## 1 0.2284483 0.7910448
##
## [[1]][[3]]
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  .
## X-squared = 38.872, df = 1, p-value = 4.525e-10
##
##
##
## [[2]]
## [[2]][[1]]
##
##             normal abnormal
##   Censored      4        1
##   No          208       51
##   Yes          20       15
##
## [[2]][[2]]
##                  [,1]       [,2]
## Censored 0.01724138 0.05970149
## No       0.89655172 3.10447761
## Yes      0.08620690 0.29850746
##
## [[2]][[3]]
##
##  Pearson's Chi-squared test
##
## data:  .
## X-squared = 9.534, df = 2, p-value = 0.008506
##
##
##
## [[3]]
## [[3]][[1]]
##
##          normal abnormal
##   female     83       22
##   male      149       45
##
## [[3]][[2]]
##             [,1]     [,2]
## female 0.3577586 1.238806
## male   0.6422414 2.223881
##
## [[3]][[3]]
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  .
## X-squared = 0.089291, df = 1, p-value = 0.7651
##
##
##
## [[4]]
## [[4]][[1]]
##
##     normal abnormal
##   0    154       49
##   1     78       18
##
## [[4]][[2]]
##        [,1]     [,2]
## 0 0.6637931 2.298507
## 1 0.3362069 1.164179
##
## [[4]][[3]]
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  .
## X-squared = 0.8004, df = 1, p-value = 0.371
##
##
##
## [[5]]
## [[5]][[1]]
##
##     normal abnormal
##   0    130       40
##   1    102       27
##
## [[5]][[2]]
##        [,1]     [,2]
## 0 0.5603448 1.940299
## 1 0.4396552 1.522388
##
## [[5]][[3]]
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  .
## X-squared = 0.1551, df = 1, p-value = 0.6937
##
##
##
## [[6]]
## [[6]][[1]]
##
##     normal abnormal
##   0    134       40
##   1     98       27
##
## [[6]][[2]]
##        [,1]     [,2]
## 0 0.5775862 2.000000
## 1 0.4224138 1.462687
##
## [[6]][[3]]
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  .
## X-squared = 0.020568, df = 1, p-value = 0.886
##
##
##
## [[7]]
## [[7]][[1]]
##
##         normal abnormal
##   <=30      61       32
##   >=45      69       11
##   41-44    102       24
##
## [[7]][[2]]
##            [,1]      [,2]
## <=30  0.2629310 0.9104478
## >=45  0.2974138 1.0298507
## 41-44 0.4396552 1.5223881
```

```
##
## [[7]][[3]]
##
##  Pearson's Chi-squared test
##
## data:  .
## X-squared = 11.971, df = 2, p-value = 0.002515
```

```
##################################################################
#############################################################
# survival plots for crude association

ggsurvplot(
    fit = survfit(Surv(time, death) ~ ser_crt_group, data = dta,),
    xlab = "Days",
    ylab = "Overall survival probability")
```



```
ggsave("20211001_survival_plots_crude.png", width = 150, height = 80,
       units = "mm")
```