install.packages("dplyr")

library(dplyr)

library(tidyr)

**#Clean data--change form**

*##change player\_id into number*

df$player\_id <- as.integer(factor(df$player\_id, levels = unique(df$player\_id)))

*##delate useless list*

df <- df[, -c(2, 3, 4,5,6,7,10)]

*##put time together*

df <- df %>%

unite(time, year, month, day, hour, sep = "-")

*##put same players information together*

df <- df %>%

group\_by(event\_timestamp) %>%

pivot\_wider(names\_from = event\_name, values\_from = event\_value, names\_prefix = "event\_")

*##rename list*

df <- df %>%

rename(warning=event\_health\_warning,

deaths = event\_number\_of\_deaths,

times = event\_total\_seconds\_in\_level,

inputs=event\_total\_number\_of\_inputs,

world=event\_world\_number,

level=event\_level\_number,

health\_pickup=`event\_number\_of\_health\_pick-ups\_collected`,

health\_recovered=event\_amount\_of\_health\_recovered,

fh\_recovered=event\_amount\_of\_health\_recovered\_when\_on\_final\_health,

fh\_time=event\_number\_of\_time\_player\_entered\_final\_health,

fh\_time\_spend=event\_total\_seconds\_spend\_on\_final\_health,

fh\_inputs=event\_total\_number\_of\_inputs\_on\_final\_health,

score=event\_score\_collected,

collected=event\_total\_number\_of\_giant\_collectibles\_collected,

collected\_badge=event\_collectible\_badge\_obtained,

time\_badge=event\_time\_badge\_obtained,

gameover=event\_game\_over,

health\_pickup\_spawned=`event\_number\_of\_health\_pick-ups\_spawned`)

**#Clean data--delete data**

df$gameover[is.na(df$gameover)] <- 0

df$collected\_badge[is.na(df$collected\_badge)] <- 0

df$time\_badge[is.na(df$time\_badge)] <- 0

df <- df[!is.na(df$warning), ]

df$times[df$times >= 2000] <- NA

df<-df[df$player\_id!=358,]

colSums(is.na(df))

**#Add Varabies**

*## degree*

convert <- function(x, y) {

return((x-1)\*4 + y)

}

df$degree <- mapply(convert, df$world, df$level)

*##Churn Rate*

*###status*

df <- df %>%

group\_by(player\_id) %>%

mutate(status = ifelse(degree == max(degree), 1, 0))

*###churn*

df <- df %>%

group\_by(player\_id) %>%

mutate(max\_date\_for\_status\_1 = max(ifelse(status == 1, event\_timestamp, as.Date(NA)), na.rm = TRUE)) %>%

ungroup() %>%

mutate(churn = ifelse(status == 1 & event\_timestamp == max\_date\_for\_status\_1, 1, 0)) %>%

select(-max\_date\_for\_status\_1)

*###natural-churn*

df$natural\_churn <- ifelse(df$degree == 28 & df$churn == 1, 2, df$churn)

*##dup\_flag*

df$dup\_flag <- ifelse(duplicated(df[c("player\_id", "degree")]) | duplicated(df[c("player\_id", "degree")], fromLast = TRUE), 1, 0)

*##dup\_count*

df <- df %>%

arrange(time) %>%

group\_by(player\_id, degree) %>%

mutate(dup\_count = row\_number() - 1) %>%

ungroup()

*##total time*

df <- df %>%

group\_by(player\_id) %>%

mutate(total\_time = sum(times))

**#Analysis**

*##cox*

install.packages("survival")

library(survival)

surv\_obj <- with(df, Surv(degree, churn))

cox\_model<-coxph(surv\_obj~deaths+times+inputs+warning+total\_time+dup\_count+gameover, data = df)

summary(cox\_model)

*##cox hierarchy*

df$natural\_churn <- as.factor(df$natural\_churn)

surv\_obj1 <- Surv(time = df$degree, event = df$natural\_churn)

model1<-coxph(surv\_obj~deaths+times+inputs+warning+total\_time+dup\_count+gameover + strata(natural\_churn), data = df)

summary(model1)

***##crr***

install.packages("cmprsk")

library(cmprsk)

cov.mat<-cbind(df$inputs,df$deaths,df$dup\_count, df$warning, df$total\_time, df$times)

fit <- crr(ftime = df$degree, fstatus = df$natural\_churn, cov.mat)

fit1 <- crr(ftime = df$degree, fstatus = df$natural\_churn, cov1 = df[, c("inputs", "deaths","collected","collected\_badge","time\_badge","dup\_count","warning","total\_time","times","gameover")],

failcode=1,cencode=0)

fit\_na1 <- crr(ftime = df$degree, fstatus = df$natural\_churn, cov1 = df[, c("inputs", "deaths","collected","collected\_badge","time\_badge","dup\_count","warning","total\_time","times","gameover")],

failcode=2,cencode=0)

Fit2 <- crr(ftime = df$degree, fstatus = df$natural\_churn, cov1 = df[, c("collected\_badge","time\_badge","dup\_count","total\_time","warning","gameover")],

failcode=1,cencode=0)

summary(fit\_na1)

summary(fit2)

***##warning***

library(stats)

model\_deaths <- lm(deaths ~ warning, data = df)

model\_times <- lm(times ~ warning, data = df)

model\_inputs <- lm(inputs ~ warning, data = df)

model\_score <- lm(score ~ warning, data = df)

model\_dup\_count <- lm(dup\_count ~ warning, data = df)

model\_gameover <- glm(gameover~ warning, data = df, family = binomial)

summary(model)

**#Save data**

write.csv(df, file = "zool.csv", row.names = FALSE)