```
library(dplyr)
library(tidyverse)
library(survival)
library(survminer)
options(scipen = 999)

master = read_csv("Master.csv")

batters = read_csv("Batting.csv")

head(master)

# Input and select desired data from the Master.csv
master =
  master %>%
  select(playerID, nameFirst, nameLast, nameGiven, weight, height, bats, throws, debut, finalGame)

master =
  master %>%
  mutate(finalGame = substr(finalGame, 0, 4)) %>%
  mutate(debut = substr(debut, 0, 4))

head(master)


# Input and select desired data from the Batting.csv
batters =
  batters %>% group_by(playerID) %>%
  summarise(LastYear = max(yearID), totalSeasons = sum(!is.na(playerID)))

head(batters)


# join the master and batters to create data
data = master %>% left_join(batters)

head(data)

# trim out bad datapoints (NA values) and if the player throws with both hands
data = data %>%
  filter(!is.na(weight), !is.na(height), !is.na(bats), !is.na(throws), !is.na(debut), !is.na(finalGame),
!is.na(LastYear), throws != 'S')

# add a boolean (one or zero) whether the players last year is 2016, meaning they are alive, alive being a 0
final = data %>%
  mutate(isIn = ifelse(LastYear > 2015, 0, 1))

head(final)

# seperate for inspection purposes
playersIn = final %>%
  filter(isIn == 0)

playersOut = final %>%
  filter(isIn == 1)

longestPlayers = final %>%
  filter(totalSeasons > 20)

longestPlayers

# build the object to start our model based on the players total seasons and whether or not they are "alive"
object = Surv(final$totalSeasons, final$isIn)
object

# build out kaplan meier object
km = survfit(object~1)
summary(km)

plot(km, conf.int = TRUE)


survfit(object ~ 1, data = final)

# Show the survival of players based on batting position
survfit(object ~ bats, data = final) %>%
  ggsurvplot(palette = "Set2",
             risk.table = TRUE,
             xlab = "Total Seasons",
             legend.labs = c("Both", "Left", "Right"),
             legend.title = "Key: ",
             pval = TRUE,
             risk.table.y.test = FALSE)

survdiff(object ~ bats, data = final)
```

```
# Show the survival of players based on batting position
survfit(object ~ throws, data = final) %>%
  ggsurvplot(palette = "Set2",
             risk.table = TRUE,
             xlab = "Total Seasons",
             legend.labs = c("Left", "Right"),
             legend.title = "Key: ",
             pval = TRUE,
             risk.table.y.test = FALSE)

survdiff(object ~ throws, data = final)

# Show the survival of players based on batting position
survfit(object ~ throws + bats, data = final) %>%
  ggsurvplot(palette = "Set2",
             risk.table = TRUE,
             xlab = "Total Seasons",
             legend.labs = c("Throws:Left, Bats: Both", "Throws:Left, Bats: Left", "Throws:Left, Bats: Right",
"Throws:Right, Bats: Both", "Throws:Right, Bats: Left", "Throws:Right, Bats: Right"),
             legend.title = "Key: ",
             pval = TRUE,
             risk.table.y.test = FALSE)

survdiff(object ~ throws + bats, data = final)


#first model - note P-Value for height is the highest
cox1 = coxph(object~ weight + height + bats + throws, data = final)
summary(cox1)
plot(survfit(cox1), conf.int = TRUE)


# trim out height

# second model - note P-Values for batsL and throwsS are both high
cox2 = coxph(object~ weight + bats + throws, data = final)
summary(cox2)
plot(survfit(cox2), conf.int = TRUE)

# remove weight
cox3 = coxph(object~ bats + throws, data = final)
summary(cox3)
plot(survfit(cox3), conf.int = TRUE)


# try running some predictions
predictions = predict(cox1, newdata = final, type = "expected") -
  final$totalSeasons

summary(predictions)

# this is probably wrong
hist(predictions,
     xlab = "Total Seasons",
     ylab = "Number of predictions")

# Test the proportional hazard assumptions
cox.zph(cox1)

# check null and alternative hyp in lesson 36
plot(cox.zph(cox1))

ggcoxzph(cox.zph(cox1))
```