```
#install and load packages, set options
install.packages("tidyverse")
install.packages("ggrepel")
install.packages("ggimage")
install.packages("nflfastR")
library(tidyverse)
library(ggrepel)
library(ggimage)
library(nflfastR)
options(scipen = 9999)

##data preparation
#load and bind data
pbp2018 = readRDS(gzcon(url(
  "https://github.com/guga31bb/nflfastR-data/blob/master/data/
play_by_play_2018.rds?raw=true")))
pbp2019 = readRDS(gzcon(url(
  "https://github.com/guga31bb/nflfastR-data/blob/master/data/
play_by_play_2019.rds?raw=true")))
pbpfull = bind_rows(pbp2018,pbp2019)
rm(pbp2018, pbp2019)

#create winteam and poswins variable - sees who wins | sees if team in
possession wins
pbpfull = pbpfull %>% mutate(winteam = ifelse(result > 0, home_team,
                                              ifelse(result == 0, "tie",
                                                     ifelse(result < 0,
away_team, "NA"))))

pbpfull = pbpfull %>% mutate(poswins = ifelse(winteam ==
posteam,"PosWins","PosLoses")) %>%
  mutate(poswins = fct_relevel(poswins, "PosLoses"))

#creating the spread variable relative to the team in possession
pbpfull = pbpfull %>% mutate(posspread = ifelse(posteam == home_team,
spread_line, -1*spread_line))

#converting qtr, down, and poswins variables to factors
cols = c("qtr","down","poswins")
pbpfull = pbpfull %>% mutate_at(cols,as_factor)

#dropping NA's from “yardline_100”, “game_seconds_remaining”, “down”,
“posspread”, and “score_differential”
pbpfull = pbpfull %>%
  drop_na(yardline_100) %>%
  drop_na(game_seconds_remaining) %>%
  drop_na(down) %>%
  drop_na(posspread) %>%
  drop_na(score_differential)

#filter out overtime and tie games
pbpfull = pbpfull %>%
  filter(qtr != 5, result != 0)
```

```
##model creation
#create logistic regression, predictions
mod1 = glm(poswins ~ game_seconds_remaining + score_differential +
yardline_100 + down + posspread, data = pbpfull, family = "binomial")

predictions_log = predict(mod1, type = "response")
pbpfull = pbpfull %>% mutate(problog = predictions_log) %>%
  mutate(prob_home_log = ifelse(posteam == home_team, problog , 1-problog))

##model visualization

#aesthetics
vertical.lines = c(900, 1800, 2700, 3600)
pbpfull %>% filter(game_id == "2018_17_MIA_BUF") %>%
  ggplot(aes(x = game_seconds_remaining,y = prob_home_log)) +
  geom_rect(aes(xmin=0, xmax=3600, ymin=0.5, ymax=1), fill = "#00338D",
alpha = .5) +
  geom_rect(aes(xmin=0, xmax=3600, ymin=0, ymax=0.5), fill = "#008E97",
alpha = .5) +
  geom_line(size = 1, color = 'white') +
  theme_bw() +
  scale_x_reverse(breaks=seq(0,3600,by=450)) +
  ylim(0,1) +
  xlab("Game Time Remaining (seconds)") +
  ylab("Home Team Win Probability") + geom_vline(xintercept =
vertical.lines, color = "orange") +
  annotate("label", x = 3450, y = .95, label = paste0('Bills')) +
  annotate("label", x = 3350, y = .05, label = paste0('Dolphins'))
```