Assignment 5: win probability and evaluating Zoë's NBA hypothesis

Due Friday, November 21 at 11:59 PM

Last year, my sister-in-law, Zoë, mentioned that she does not enjoy watching basketball very much, because "all that matters is the last two minutes of the game." She felt that the win probability for most of the game, until the last couple of minutes, was not a strong predictor of the final winner. In this homework, we will use NBA data to build a win probability model and evaluate Zoë's claim.

Submit your assignment to Gradescope as a PDF file generated from your R Markdown. Make sure your PDF includes all code, outputs, and written answers to questions. All code and outputs used to answer the written questions must be included in the PDF for full credit.

Load the NBA data

Run the code below to load NBA play-by-play data from the 2023–24 season using the `hoopR` package. The code below also creates a column that indicates whether the home team won or lost (`home_win`) and a column with the score differential (`score_diff`).

```
library(hoopR) # package to get NBA data
# Load NBA play-by-play data for the 2023-24 season
nba_pbp <- load_nba_pbp(seasons = 2024)
# Get the final score and winner for each game
final_nba_score <- nba_pbp %>%
  group_by(game_date, game_id) %>%
  summarize(final_home_score = max(home_score),
            final_away_score = max(away_score)) %>%
  mutate(home_win =
           as.numeric(final_home_score > final_away_score))
# Join final_nba_score to nba_pbp
nba_df <- nba_pbp %>%
  left_join(final_nba_score, by = c("game_date", "game_id")) %>%
  mutate(is_home_possession = team_id == home_team_id,
         score_diff = home_score - away_score)
# Display first few rows of nba_df
nba_df %>%
  head(5)
```

```
## # A tibble: 5 x 67
##   game_play_number         id sequence_number type_id type_text text  away_score
##              <int>      <dbl>           <int>   <int> <chr>     <chr>      <int>
## 1                1    4.02e 9               4     615 Jumpball  Dani~          0
## 2                2    4.02e 9               7     110 Driving ~ Jrue~          0
## 3                3    4.02e 9               9     132 Step Bac~ Luka~          0
## 4                4    4.02e10              10     155 Defensiv~ Jays~          0
## 5                5    4.02e10              11      45 Personal~ Luka~          0
## # i 60 more variables: home_score <int>, period_number <int>,
## #   period_display_value <chr>, clock_display_value <chr>, scoring_play <lgl>,
## #   score_value <int>, team_id <int>, athlete_id_1 <int>, athlete_id_2 <int>,
## #   athlete_id_3 <int>, wallclock <chr>, shooting_play <lgl>,
```

```
## #   coordinate_x_raw <dbl>, coordinate_y_raw <dbl>, game_id <int>,
## #   season <int>, season_type <int>, home_team_id <int>, home_team_name <chr>,
## #   home_team_mascot <chr>, home_team_abbrev <chr>, ...
```

Create an NBA win probability model

1 [8 points]. Using the `nba_df` dataset, build a win probability model to predict whether the home team wins (based on the `home_win` column). Your model should use the following predictors:

- `score_diff`: the score differential, calculated as home score minus away score
- `start_game_seconds_remaining`: the number of seconds remaining in the game

Once you've built the model, create a new column in your dataset called `wp` that contains the predicted win probabilities from your model.

```r
# Ensure start_game_seconds_remaining exists; if not, create it
if(!"start_game_seconds_remaining" %in% names(nba_df)){
  to_seconds <- function(clock){
    parts <- as.numeric(unlist(strsplit(clock, ":")))
    if(length(parts) != 2 || any(is.na(parts))) return(0)
    60*parts[1] + parts[2]
  }

  nba_df <- nba_df %>%
    mutate(clock_sec = sapply(clock_display_value, to_seconds), seconds_elapsed = case_when(
              period_number <= 4 ~ (period_number - 1)*720 + (720 - clock_sec),
              TRUE ~ 4*720 + (period_number - 5)*300 + (300 - clock_sec)
            ),
            start_game_seconds_remaining = pmax(2880 - seconds_elapsed, 0)) %>%
    select(-clock_sec, -seconds_elapsed)
}

# win probability model
wp_model <- glm(home_win ~ score_diff + start_game_seconds_remaining,
                data = nba_df,
                family = binomial)

# add predicted wp column
nba_df <- nba_df %>%
  mutate(wp = predict(wp_model, newdata = ., type = "response"))

summary(wp_model)
```

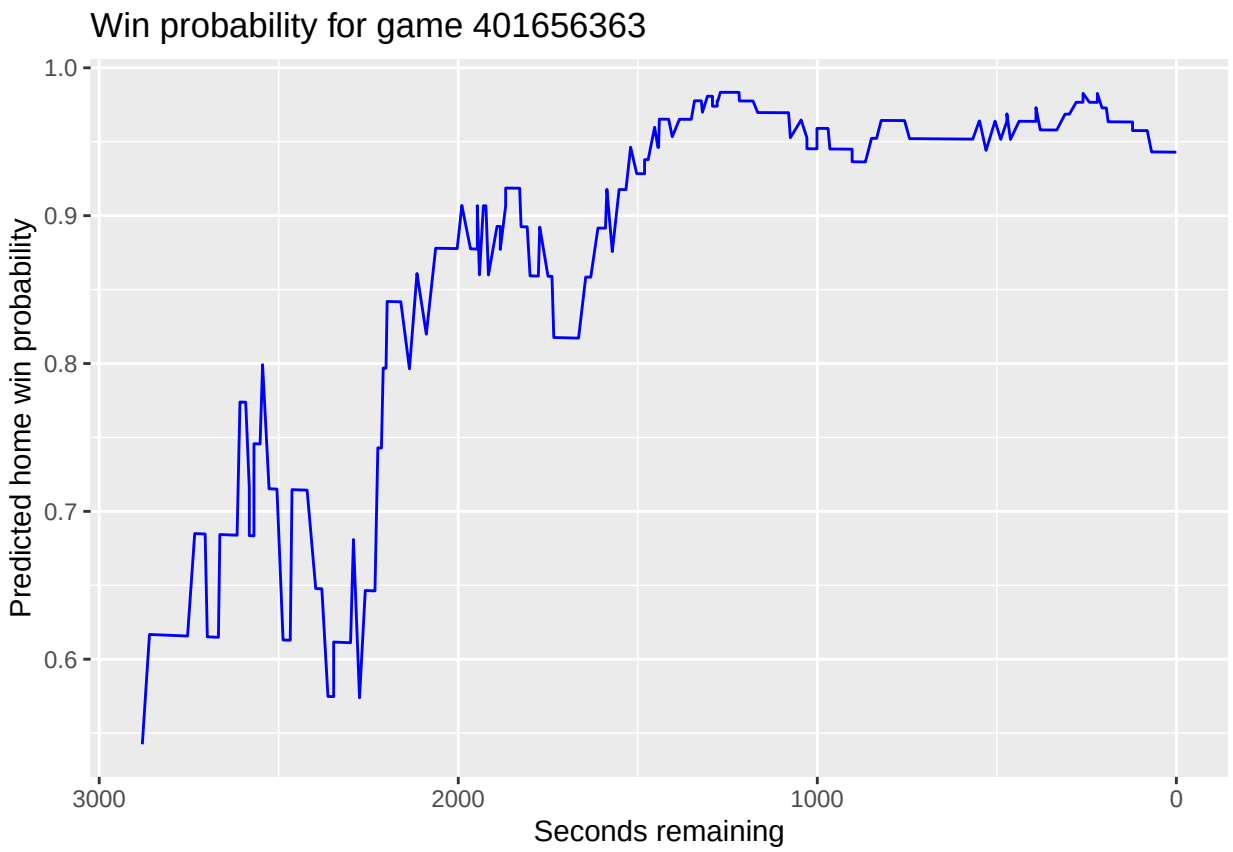
```
##
## Call:
## glm(formula = home_win ~ score_diff + start_game_seconds_remaining,
##     family = binomial, data = nba_df)
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  4.881e-02  6.418e-03   7.606 2.83e-14 ***
## score_diff                   1.531e-01  4.284e-04 357.438  < 2e-16 ***
## start_game_seconds_remaining 4.216e-05  3.685e-06  11.438  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 845667  on 614446  degrees of freedom
## Residual deviance: 611056  on 614444  degrees of freedom
## AIC: 611062
##
## Number of Fisher Scoring iterations: 5
```

2 [4 points]. Plot the win probability vs seconds remaining in the game for *one game* in the dataset. Look up information about that game. Do your win probability values seem to make sense?

```
example_game_id <- nba_df$game_id[1]
one_game <- nba_df %>%
  filter(game_id == example_game_id)

ggplot(one_game, aes(x = start_game_seconds_remaining, y = wp)) +
  geom_line(color = "blue") +
  scale_x_reverse() +
  labs(title = paste("Win probability for game", example_game_id),
       x = "Seconds remaining",
       y = "Predicted home win probability")
```



The win probability makes sense as game events that are in favor of the home team increase the probability and nudge it towards 1. Events are reflect better performance and win probability for the away team, would nudge the graph towards 0 as seen in dips as the seconds remaining(x-axis) decreases. These events and changes in the win probability match the specific game recap.

3 [4 points]. What percentage of the time does the team that the win probability model predicts will win at halftime actually go on to win the game? *Hint*: filter the dataset so that `start_game_seconds_remaining`

is equal to $24 * 60 = 1440$ and calculate the proportion of games that the team predicted to win based on the win probability model ends up winning the game.

```
halftime_preds <- nba_df %>%
  filter(!is.na(wp)) %>%
  group_by(game_id) %>%
  filter(abs(start_game_seconds_remaining - 1440) ==
           min(abs(start_game_seconds_remaining - 1440))) %>%
  slice(1) %>% ungroup()

halftime_accuracy <- mean((halftime_preds$wp > 0.5) == (halftime_preds$home_win == 1))

halftime_accuracy
```

```
## [1] 0.7424242
```

4 [4 points]. What percentage of the time does the team that the win probability model predicts will win at the 2 minute mark, win the game? *Hint*: filter the dataset so that `start_game_seconds_remaining` is equal to $2 * 60 = 120$ and calculate the proportion of games that the team predicted to win based on the win probability model ends up winning the game.

```
two_min_preds <- nba_df %>%
  filter(!is.na(wp)) %>%
  group_by(game_id) %>%
  filter(abs(start_game_seconds_remaining - 120) ==
           min(abs(start_game_seconds_remaining - 120))) %>%
  slice(1) %>% ungroup()

two_min_accuracy <- mean((two_min_preds$wp > 0.5) == (two_min_preds$home_win == 1))

two_min_accuracy
```

```
## [1] 0.9219697
```

Comparison to the NFL

Now we will compare our NBA results to the NFL in order to evaluate Zoë's hypothesis. Run the following code to load in NFL data from last season. The column `home_wp` gives the win probability prediction for the *home team*.

```
library(nflreadr) # library with NFL data
nfl_pbp <- nflreadr::load_pbp(seasons = 2024)
# add column with whether home team won
nfl_df <- nfl_pbp %>%
  mutate(home_win = ifelse(result > 0, 1, 0)) %>%
  filter(season_type == "REG")
```

5 [3 points]. What percentage of the time does the team that the win probability model predicts will win at the half, win the game? *Hint*: filter the dataset so that `game_seconds_remaining` is equal to $30 * 60 = 1800$ and calculate the proportion of games that the team predicted to win based on the win probability model ends up winning the game.

```
nfl_halftime <- nfl_df %>%
  filter(!is.na(home_wp)) %>%
  group_by(game_id) %>%
  filter(abs(game_seconds_remaining - 1800) ==
           min(abs(game_seconds_remaining - 1800))) %>%
```

```
  slice(1) %>% ungroup()

nfl_halftime_accuracy <- mean((nfl_halftime$home_wp > 0.5) == (nfl_halftime$home_win == 1))

nfl_halftime_accuracy
```

```
## [1] 0.7434944
```

6 [3 points]. What percentage of the time does the team that the win probability model predicts will win at the 2 minute mark, win the game?

```
nfl_two_min <- nfl_df %>%
  filter(!is.na(home_wp)) %>%
  group_by(game_id) %>%
  filter(abs(game_seconds_remaining - 120) ==
           min(abs(game_seconds_remaining - 120))) %>%
  slice(1) %>% ungroup()

nfl_two_min_accuracy <- mean((nfl_two_min$home_wp > 0.5) == (nfl_two_min$home_win == 1))

nfl_two_min_accuracy
```

```
## [1] 0.8921933
```

7 [4 points]. Do you agree or disagree with Zoë's hypothesis? Explain why or why not. How else could you analyze her claim?

I agree with Zoë's hypothesis to an extent as her conclusion is true. Our calculations show that the win probability model at halftime is only moderately predictive — the team favored at halftime does not win nearly all the time. But at the two-minute mark, the predicted favorite wins a very high percentage of games. This supports Zoë's theory that the last few minutes contain much more decisive information. However, the halftime predictions are still better than random chance, meaning earlier parts of the game do matter. So the truth is a combination as last two minutes matter a lot, but the rest of the game is still quite informative.

Extra credit [3 points]. Evaluate Zoë's claim using WNBA data (i.e., repeating steps 1-4). Does her hypothesis hold up in the WNBA? *Hint:* you can get WNBA play-by-play data using the `wehoop` library in R.

```
library(wehoop)
wnba <- load_wnba_pbp(seasons = 2024)

# convert "MM:SS" clock to seconds
to_seconds <- function(clock){
  parts <- suppressWarnings(as.numeric(unlist(strsplit(clock, ":"))))
  if(length(parts) != 2 || any(is.na(parts))) return(0)
  60*parts[1] + parts[2]
}

# Add start_game_seconds_remaining (WNBA = 10-min quarters)
wnba_df <- wnba %>%
  mutate(
    clock_sec = sapply(clock_display_value, to_seconds),
    seconds_elapsed = case_when(
      period_number <= 4 ~ (period_number - 1)*600 + (600 - clock_sec),    # regulation
      TRUE ~ 4*600 + (period_number - 5)*300 + (300 - clock_sec)           # OT = 5 minutes
    ),
    start_game_seconds_remaining = pmax(2400 - seconds_elapsed, 0),
```

```
    score_diff = home_score - away_score
  )

# Final outcome
final_wnba <- wnba_df %>%
  group_by(game_id, game_date) %>%
  summarize(final_home = max(home_score),
            final_away = max(away_score),
            .groups = "drop") %>%
  mutate(home_win = as.numeric(final_home > final_away))

# Join
wnba_df <- wnba_df %>%
  left_join(final_wnba, by = c("game_id", "game_date"))

# Model
wnba_model <- glm(home_win ~ score_diff + start_game_seconds_remaining,
                  data = wnba_df,
                  family = binomial)

wnba_df <- wnba_df %>%
  mutate(wp = predict(wnba_model, newdata = ., type = "response"))


wnba_halftime <- wnba_df %>%
  group_by(game_id) %>%
  filter(abs(start_game_seconds_remaining - 1200) ==
           min(abs(start_game_seconds_remaining - 1200))) %>%
  slice(1) %>%
  ungroup()

mean((wnba_halftime$wp > 0.5) == (wnba_halftime$home_win == 1))
```

```
## [1] 0.7333333
```

```
wnba_two <- wnba_df %>%
  group_by(game_id) %>%
  filter(abs(start_game_seconds_remaining - 120) ==
           min(abs(start_game_seconds_remaining - 120))) %>%
  slice(1) %>%
  ungroup()

mean((wnba_two$wp > 0.5) == (wnba_two$home_win == 1))
```

```
## [1] 0.9208333
```