Analytics - NBA

Setup and Data

Part 1 – Data Cleaning

In this section, you’re going to work to answer questions using data from both team and player stats. All provided stats are on the game level.

Question 1

QUESTION: What was the Warriors’ Team offensive and defensive eFG% in the 2015-16 regular season? Remember that this is in the data as the 2015 season.

```
# extract warriors offensive stats for 2015 regular season
# compute eFG using (FG2M + 0.5*FG3M)/FGA per game
team_data_transform_11 <- team_data %>%
                         filter(season == 2015 & gametype == 2 & off_team == "GSW") %>%
                         mutate(efg = (fg2made + 0.5*fg3made)/fgattempted)

# compute eFG% offensive for the season
off_efg <- round(mean(team_data_transform_11$efg)*100,1)

# extract warriors offensive stats for 2015 regular season
# compute eFG using (FG2M + 0.5*FG3M)/FGA per game
team_data_transform_12 <- team_data %>%
                         filter(season == 2015 & gametype == 2 & def_team == "GSW") %>%
                         mutate(efg = (fg2made + 0.5*fg3made)/fgattempted)

# compute eFG% defensive for the season
def_efg <- round(mean(team_data_transform_12$efg)*100,1)
```

ANSWER 1:

Offensive: 41.3% eFG
Defensive: 39.2% eFG

Question 2

QUESTION: What percent of the time does the team with the higher eFG% in a given game win that game? Use games from the 2014-2023 regular seasons. If the two teams have an exactly equal eFG%, remove that game from the calculation.

```
# extract team data for regular season 2014-2023
# compute eFG using (FG2M + 0.5*FG3M)/FGA for offensive team of each game
team_data_transform2 <- team_data %>%
                         filter(season >= 2014 & season <= 2023 & gametype == 2) %>%
                         mutate(efg = (fg2made + 0.5*fg3made)/fgattempted)

# group games by game id
# select teams for each game and determine winner
team_data_transform2 <- team_data_transform2 %>%
                        group_by(nbagameid) %>%
                        summarize(
                          teamA = first(off_team),
                          teamB = last(off_team),
                          efgA = first(efg),
                          efgB = last(efg),
                          pointsA = first(points),
                          pointsB = last(points),
                          winner = if_else(first(points) > last(points), first(off_team), last(off_team))
                        )

# filter out games with equal FG%
team_data_transform2 <- team_data_transform2 %>%
                        filter(efgA != efgB)

# determine if winner of each game had higher fg%
team_data_transform2 <- team_data_transform2  %>%
                        mutate(higherFG_won = ifelse((efgA>efgB & winner == teamA)|(efgB > efgA & winner == teamB), 1, 0))

# percentage of games won by team with higher fg%
higher_efg_won_perc <- round(mean(team_data_transform2$higherFG_won)*100,1)
```

ANSWER 2:

73.0%

Question 3

QUESTION: What percent of the time does the team with more offensive rebounds in a given game win that game? Use games from the 2014-2023 regular seasons. If the two teams have an exactly equal number of offensive rebounds, remove that game from the calculation.

```
# extract team data for regular season 2014-2023
# group games by game id
# select teams for each game and determine winner
# filter out games with equal offensive rebounds
team_data_transform2 <- team_data %>%
                        filter(season >= 2014 & season <= 2023 & gametype == 2) %>%
                        group_by(nbagameid) %>%
                        summarize(
                          teamA = first(off_team),
                          teamB = last(off_team),
                          off_rebA = first(reboffensive),
                          off_rebB = last(reboffensive),
                          pointsA = first(points),
                          pointsB = last(points),
                          winner = if_else(first(points) > last(points), first(off_team), last(off_team))
                        ) %>%
                        filter(off_rebA != off_rebB)

# determine if winner of each game had higher fg%
team_data_transform2 <- team_data_transform2  %>%
                        mutate(higher_off_reb_won = ifelse((off_rebA>off_rebB & winner == teamA)|(off_rebB > off_rebA & winn
er == teamB), 1, 0))

# percentage of games won by team with higher fg%
higher_off_reb_won_perc <- round(mean(team_data_transform2$higher_off_reb_won)*100,1)
```

ANSWER 3:

46.2%

Question 4

QUESTION: Do you have any theories as to why the answer to question 3 is lower than the answer to question 2? Try to be clear and concise with your answer.

ANSWER 4:

Effective FG% is a better measure of a teams offensive efficiency than offensive rebounds. higher eFG% implies that the team is more efficient at converting its shot attempts into points while, higher offensive rebounds implies higher second chance points but not necessarily converted opportunities.

Question 5

QUESTION: Look at players who played at least 25% of their possible games in a season and scored at least 25 points per game played. Of those player-seasons, what percent of games were they available for on average? Use games from the 2014-2023 regular seasons.

For example:

- Ja Morant does not count in the 2023-24 season, as he played just 9 out of 82 games this year, even though he scored 25.1 points per game.
- Chet Holmgren does not count in the 2023-24 season, as he played all 82 games this year but scored 16.5 points per game.
- LeBron James does count in the 2023-24 season, as he played 71 games and scored 25.7 points per game.

```
data_transformed <- player_data %>%
                  # select key columns
                  select(season, gametype,nbapersonid,player_name, starter, missed, points) %>%
                  # filter by 2014-2023 regular season and player not availble due to injury
                  filter (season >= 2014 & season <= 2023 & gametype == 2 & missed == 0) %>%
                  # group by season and player identifiers
                  group_by(season, nbapersonid, player_name) %>%
                  # create games_played, percent_played and points per game played variables
                  summarise(games_played = sum(starter), percent_played = sum(starter)/82*100, ppg = round(mean(points[start
er == 1]),1),.groups = "drop") %>%
                  # exclude rows with percent_played less than 25% and ppg < 25
                  filter(percent_played >= 25 & ppg >= 25) %>%
                  arrange(player_name)

player_season_avg_availability <- round(mean(data_transformed$percent_played),1)
```

ANSWER 5:

80.2% of games

Question 6

QUESTION: What % of playoff series are won by the team with home court advantage? Give your answer by round. Use playoffs series from the 2014-**2022** seasons. Remember that the 2023 playoffs took place during the 2022 season (i.e. 2022-23 season).

```
team_data %>%
select(season, gametype, nbagameid, off_team, def_team, off_home, off_win, points) %>%
filter(season >= 2014 & season <= 2022 & gametype == 4) %>%
group_by(season, nbagameid) %>%
summarise(home_team = off_team[off_home == 1], away_team = off_team[off_home == 0], home_points = points[off_home == 1], awa
y_points = points[off_home == 0], game_winner = off_team[off_win == 1]) %>%
arrange(nbagameid) %>%
mutate(team_pair = map2_chr(home_team, away_team, ~ paste(sort(c(.x, .y)), collapse = "-"))) %>%
  group_by(season,team_pair) %>%
  summarise(no_games = n(), first_game_id = first(nbagameid), home_advantage = first(home_team), series_winner = last(game_w
inner)) %>%
  arrange(first_game_id) %>%
  group_by(season) %>%
  mutate(
    row_num = row_number(),
    round = case_when(
      row_num <= 8 ~ "First Round",
      row_num <= 12 ~ "Conference Semis",
      row_num <= 14 ~ "Conference Finals",
      TRUE ~ "NBA Finals"
    )
  ) %>%
  select(-first_game_id, -row_num) %>%
  group_by(round) %>%
  summarise(total_games = n(), home_adv_wins = sum(home_advantage == series_winner), home_adv_win_prop = home_adv_wins/total
_games)
```

```
## # A tibble: 4 × 4
##   round             total_games home_adv_wins home_adv_win_prop
##   <chr>                   <int>         <int>             <dbl>
## 1 Conference Finals          18            10             0.556
## 2 Conference Semis           36            23             0.639
## 3 First Round                72            61             0.847
## 4 NBA Finals                  9             7             0.778
```

ANSWER 6:

Round 1: XX.X%
Round 2: XX.X%
Conference Finals: XX.X%
Finals: XX.X%

Question 7

QUESTION: Among teams that had at least a +5.0 net rating in the regular season, what percent of them made the second round of the playoffs the **following** year? Among those teams, what percent of their top 5 total minutes played players (regular season) in the +5.0 net rating season played in that 2nd round playoffs series? Use the 2014-2021 regular seasons to determine the +5 teams and the 2015-2022 seasons of playoffs data.

For example, the Thunder had a better than +5 net rating in the 2023 season. If we make the 2nd round of the playoffs **next** season (2024-25), we would qualify for this question. Our top 5 minutes played players this season were Shai Gilgeous-Alexander, Chet Holmgren, Luguentz Dort, Jalen Williams, and Josh Giddey. If three of them play in a hypothetical 2nd round series next season, it would count as 3/5 for this question.

Hint: The definition for net rating is in the data dictionary.

ANSWER 7:

Percent of +5.0 net rating teams making the 2nd round next year: XX.X%
Percent of top 5 minutes played players who played in those 2nd round series: XX.X%

Part 2 – Playoffs Series Modeling

For this part, you will work to fit a model that predicts the winner and the number of games in a playoffs series between any given two teams.

This is an intentionally open ended question, and there are multiple approaches you could take. Here are a few notes and specifications:

1. Your final output must include the probability of each team winning the series. For example: “Team A has a 30% chance to win and team B has a 70% chance.” instead of “Team B will win.” You must also predict the number of games in the series. This can be probabilistic or a point estimate.

2. You may use any data provided in this project, but please do not bring in any external sources of data.

3. You can only use data available prior to the start of the series. For example, you can’t use a team’s stats from the 2016-17 season to predict a playoffs series from the 2015-16 season.

4. The best models are explainable and lead to actionable insights around team and roster construction. We’re more interested in your thought process and critical thinking than we are in specific modeling techniques. Using smart features is more important than using fancy mathematical machinery.

5. Include, as part of your answer:

- A brief written overview of how your model works, targeted towards a decision maker in the front office without a strong statistical background.
- What you view as the strengths and weaknesses of your model.
- How you’d address the weaknesses if you had more time and/or more data.
- Apply your model to the 2024 NBA playoffs (2023 season) and create a high quality visual (a table, a plot, or a plotly) showing the 16 teams’ (that made the first round) chances of advancing to each round.

```
regular_season <- team_data %>%
  filter(gametype == 2) %>%
  select(season,nbagameid,off_team,def_team,off_home,off_win,fg2made,fg3made,ftmade,assists,turnovers,possessions,points)

regular_season <- regular_season %>%
  inner_join(regular_season, by = c("def_team" = "off_team", "nbagameid" = "nbagameid", "season" = "season"), suffix = c("_off","_def")) %>%
  select(-off_home_off, -def_team_def, -off_home_def, -off_win_def) %>%
  arrange(nbagameid)

regular_season <- regular_season %>%
  group_by(season,off_team) %>%
  summarise(wins = sum(off_win_off),
            off_fg2made = round(mean(fg2made_off),1),
            off_fg3made = round(mean(fg3made_off),1),
            off_ftmade = round(mean(ftmade_off),1),
            off_assists = round(mean(assists_off),1),
            off_turnovers = round(mean(turnovers_off),1),
            off_points = round(mean(points_off),1),
            off_possessions = round(mean(possessions_off),1),
            def_fg2made = round(mean(fg2made_def),1),
            def_fg3made = round(mean(fg3made_def),1),
            def_ftmade = round(mean(ftmade_def),1),
            def_assists = round(mean(assists_def),1),
            def_turnovers = round(mean(turnovers_def),1),
            def_points = round(mean(points_def),1),
            def_possessions = round(mean(possessions_def),1)) %>%
  arrange(season,off_team)
```

```
playoff_data  <- team_data %>%
  filter(gametype == 4, season >= 2014) %>%
  group_by(season, nbagameid) %>%
  summarise(home_team = off_team[off_home == 1], away_team = off_team[off_home == 0], home_points = points[off_home == 1], a
way_points = points[off_home == 0], game_winner = off_team[off_win == 1]) %>%
  arrange(nbagameid) %>%
  mutate(team_pair = map2_chr(home_team, away_team, ~ paste(sort(c(.x, .y)), collapse = "-"))) %>%
    group_by(season,team_pair) %>%
    summarise(no_games = n(), first_game_id = first(nbagameid), home_advantage = first(home_team), series_winner = last(game
_winner)) %>%
    arrange(first_game_id) %>%
    group_by(season) %>%
    mutate(
      row_num = row_number(),
      round = case_when(
        row_num <= 8 ~ "First Round",
        row_num <= 12 ~ "Conference Semis",
        row_num <= 14 ~ "Conference Finals",
        TRUE ~ "NBA Finals"
      )
    ) %>%
  select(-first_game_id, -row_num) %>%
  separate(team_pair,into = c("team1","team2"), sep = "-") %>%
  pivot_longer(cols = c(team1,team2),names_to = "teams", values_to = "team") %>%
  select(-teams)
```

```
teams <- regular_season %>% filter(season == "2014") %>% pull(off_team) %>% unique()

df <- tibble(
  season = as.double(as.character(factor(rep(2014:2023, each = 30)))),
  off_team = as.character(rep(teams,10)),
  fir_round = rep(0,300),
  fir_round_games = rep(0,300),
  fir_round_w = rep(0,300),
  conf_semi = rep(0,300),
  conf_semi_games = rep(0,300),
  conf_semi_w = rep(0,300),
  conf_fin = rep(0,300),
  conf_fin_games = rep(0,300),
  conf_fin_w = rep(0,300),
  nba_fin = rep(0,300),
  nba_fin_games = rep(0,300),
  nba_fin_w = rep(0,300)
)

df <- df %>%
  inner_join(regular_season,by = c("off_team","season")) %>%
  arrange(season,off_team) %>%
  left_join(playoff_data,by = c("season","off_team" = "team")) %>%
  select(season,off_team,round,series_winner,everything()) %>%
  mutate(
    fir_round = if_else(round == "First Round", 1, 0),
    fir_round = replace_na(fir_round,0),
    fir_round_games = if_else(round == "First Round", no_games, 0),
    fir_round_games = replace_na(fir_round_games,0),
    fir_round_w = if_else(round == "First Round" & off_team == series_winner, 1, 0),
    fir_round_w = replace_na(fir_round_w,0),
    conf_semi = if_else(round == "Conference Semis", 1, 0),
    conf_semi = replace_na(conf_semi,0),
    conf_semi_games = if_else(round == "Conference Semis", no_games, 0),
    conf_semi_games = replace_na(conf_semi_games,0),
    conf_semi_w = if_else(round == "Conference Semis" & off_team == series_winner, 1, 0),
    conf_semi_w = replace_na(conf_semi_w,0),
    conf_fin = if_else(round == "Conference Finals" , 1, 0),
    conf_fin = replace_na(conf_fin,0),
    conf_fin_games = if_else(round == "Conference Finals" , no_games, 0),
    conf_fin_games = replace_na(conf_fin_games,0),
    conf_fin_w = if_else(round == "Conference Finals" & off_team == series_winner, 1, 0),
    conf_fin_w = replace_na(conf_fin_w,0),
    nba_fin = if_else(round == "NBA Finals", 1, 0),
    nba_fin = replace_na(nba_fin,0),
    nba_fin_games = if_else(round == "NBA Finals", no_games, 0),
    nba_fin_games = replace_na(nba_fin_games,0),
    nba_fin_w = if_else(round == "NBA Finals" & off_team == series_winner, 1, 0),
    nba_fin_w = replace_na(nba_fin_w,0)
  ) %>%
  group_by(season,off_team) %>%
  summarise(fir_round = sum(fir_round),
            fir_round_games = sum(fir_round_games),
            fir_round_win = sum(fir_round_w),
            conf_semi = sum(conf_semi),
            conf_semi_games = sum(conf_semi_games),
            conf_semi_win = sum(conf_semi_w),
            conf_final = sum(conf_fin),
            conf_final_games = sum(conf_fin_games),
            conf_final_win = sum(conf_fin_w),
            nba_final = sum(nba_fin),
            nba_final_games = sum(nba_fin_games),
            nba_final_win = sum(nba_fin_w),
            off_fg2made = first(off_fg2made),
            off_fg3made = first(off_fg3made),
            off_ftmade = first(off_ftmade),
            off_assists = first(off_assists),
            off_turnovers = first(off_turnovers),
            off_points = first(off_points),
            off_possessions = first(off_possessions),
            def_fg2made = first(def_fg2made),
            def_fg3made = first(def_fg3made),
            def_ftmade = first(def_ftmade),
            def_assists = first(def_assists),
            def_turnovers = first(def_turnovers),
            def_points = first(def_points),
            def_possessions = first(def_possessions))

df
```

```
## # A tibble: 300 × 28
## # Groups:   season [10]
##    season off_team fir_round fir_round_games fir_round_win conf_semi
##     <dbl> <chr>        <dbl>           <dbl>         <dbl>     <dbl>
##  1   2014 ATL              1               6             1         1
##  2   2014 BKN              1               6             0         0
##  3   2014 BOS              1               4             0         0
##  4   2014 CHA              0               0             0         0
##  5   2014 CHI              1               6             1         1
##  6   2014 CLE              1               4             1         1
##  7   2014 DAL              1               5             0         0
##  8   2014 DEN              0               0             0         0
##  9   2014 DET              0               0             0         0
## 10   2014 GSW              1               4             1         1
## # ℹ 290 more rows
## # ℹ 22 more variables: conf_semi_games <dbl>, conf_semi_win <dbl>,
## #   conf_final <dbl>, conf_final_games <dbl>, conf_final_win <dbl>,
## #   nba_final <dbl>, nba_final_games <dbl>, nba_final_win <dbl>,
## #   off_fg2made <dbl>, off_fg3made <dbl>, off_ftmade <dbl>, off_assists <dbl>,
## #   off_turnovers <dbl>, off_points <dbl>, off_possessions <dbl>,
## #   def_fg2made <dbl>, def_fg3made <dbl>, def_ftmade <dbl>, …
```

```
set.seed(111)

season_round_win <- function(cur_season){
  train_data <- df %>% filter(season < cur_season)
  test_data <- df %>% filter(season == cur_season)

  if(nrow(train_data) > 0){

    model <- glm(fir_round_win ~ off_fg2made + off_fg3made + off_ftmade + off_assists + off_turnovers + off_points + off_pos
sessions +
                   def_fg2made + def_fg3made + def_ftmade + def_assists + def_turnovers + def_points + def_possessions,data
= train_data, family = "binomial")

    model_games <- lm(fir_round_games ~ off_fg2made + off_fg3made + off_ftmade + off_assists + off_turnovers + off_points +
off_possessions +
                   def_fg2made + def_fg3made + def_ftmade + def_assists + def_turnovers + def_points + def_possessions,data
= train_data)

    predictions <- round(predict(model, test_data, type = "response"),1)
    predictions_games <- round(predict(model_games, test_data))
    test_data <- test_data %>% mutate( prob_fir_round_win = predictions, pred_fir_round_games = predictions_games)


    model <- glm(conf_semi_win ~ fir_round_win + off_fg2made + off_fg3made + off_ftmade + off_assists + off_turnovers + off_
points + off_possessions +
                   def_fg2made + def_fg3made + def_ftmade + def_assists + def_turnovers + def_points + def_possessions,data
= train_data, family = "binomial")

    model_games <- lm(conf_semi_games ~ fir_round_games +off_fg2made + off_fg3made + off_ftmade + off_assists + off_turnover
s + off_points + off_possessions +
                   def_fg2made + def_fg3made + def_ftmade + def_assists + def_turnovers + def_points + def_possessions,data
= train_data)

    predictions <- round(predict(model, test_data, type = "response"),1)
    predictions_games <- round(predict(model_games, test_data))
    test_data <- test_data %>% mutate( prob_conf_semi_win = predictions, pred_conf_semi_games = predictions_games)


    model <- glm(conf_final_win ~ conf_semi_win + fir_round_win + off_fg2made + off_fg3made + off_ftmade + off_assists + off
_turnovers + off_points + off_possessions +
                   def_fg2made + def_fg3made + def_ftmade + def_assists + def_turnovers + def_points + def_possessions,data
= train_data, family = "binomial")

    model_games <- lm(conf_final_games ~ conf_semi_games + fir_round_games + off_fg2made + off_fg3made + off_ftmade + off_as
sists + off_turnovers + off_points + off_possessions +
                   def_fg2made + def_fg3made + def_ftmade + def_assists + def_turnovers + def_points + def_possessions,data
= train_data)
```

```
    predictions <- round(predict(model, test_data, type = "response"),1)
    predictions_games <- round(predict(model_games, test_data))
    test_data <- test_data %>% mutate( prob_conf_fin_win = predictions, pred_conf_fin_games = predictions_games)


    model <- glm(nba_final_win ~ conf_final_win + conf_semi_win + fir_round_win + off_fg2made + off_fg3made + off_ftmade + o
ff_assists + off_turnovers + off_points + off_possessions +
                   def_fg2made + def_fg3made + def_ftmade + def_assists + def_turnovers + def_points + def_possessions,data
= train_data, family = "binomial")

    model_games <- lm(nba_final_games ~ conf_final_games + conf_semi_games + fir_round_games + off_fg2made + off_fg3made + o
ff_ftmade + off_assists + off_turnovers + off_points + off_possessions +
                   def_fg2made + def_fg3made + def_ftmade + def_assists + def_turnovers + def_points + def_possessions,
                 data = train_data)

    predictions <- round(predict(model, test_data, type = "response"),1)
    predictions_games <- round(predict(model_games, test_data))
    test_data <- test_data %>% mutate( prob_nba_fin_win = predictions, pred_nba_fin_games = predictions_games)




    return(test_data %>% select(season,off_team,prob_fir_round_win,pred_fir_round_games,
                                prob_conf_semi_win,pred_conf_semi_games,
                                prob_conf_fin_win,pred_conf_fin_games,
                                prob_nba_fin_win,pred_nba_fin_games))
  }
}
```

```
results <- unique(df$season) %>% map_dfr(season_round_win)
results
```

```
## # A tibble: 270 × 10
## # Groups:   season [9]
##    season off_team prob_fir_round_win pred_fir_round_games prob_conf_semi_win
##     <dbl> <chr>                 <dbl>                <dbl>              <dbl>
##  1   2015 ATL                       0                    7                0.3
##  2   2015 BKN                       0                    2                1  
##  3   2015 BOS                       1                    4                0  
##  4   2015 CHA                       1                    2                0  
##  5   2015 CHI                       1                    0                0  
##  6   2015 CLE                       1                    3                0.9
##  7   2015 DAL                       1                    1                0  
##  8   2015 DEN                       1                    0                0  
##  9   2015 DET                       0                    1                0  
## 10   2015 GSW                       1                    8                1  
## # ℹ 260 more rows
## # ℹ 5 more variables: pred_conf_semi_games <dbl>, prob_conf_fin_win <dbl>,
## #   pred_conf_fin_games <dbl>, prob_nba_fin_win <dbl>, pred_nba_fin_games <dbl>
```