Final Project - NBA Exploratory Analysis

Clajerson Gimena

6-8-2025

Load Packages

```
library(RSQLite)
library(DBI)
library(RMariaDB)
library(dplyr, warn.conflicts = FALSE)
library(ggplot2)
library(bit64)
library(tidyr)
library(car)
library(gridExtra)
library(grid)
library(patchwork)
```

Establishing a Connection

```
con <- DBI::dbConnect(RSQLite::SQLite(), dbname = "nba.sqlite")
```

Exploratory Data Analysis

Question 1: How many players went UCLA

```
dbGetQuery(con, "SELECT COUNT(*) AS ucla_alumni
                 FROM common_player_info
                 WHERE school LIKE 'UCLA'
           ")
```

```
  ucla_alumni
1          62
```

Question 2: What is the Average Draft Combine Statistics Overtime

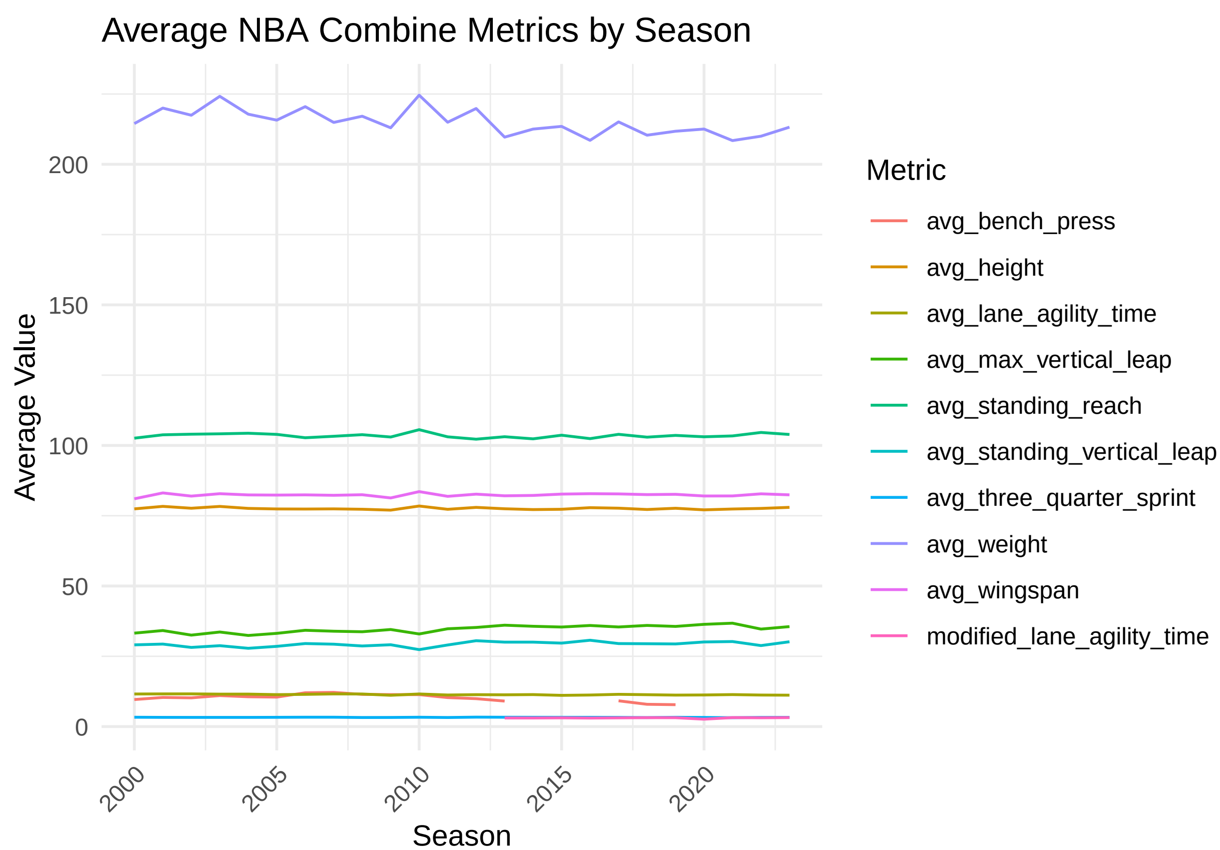
```
combine_stats_avg <- dbGetQuery(con, "SELECT season,
                                             AVG(height_wo_shoes) AS avg_height,
                                             AVG(weight) AS avg_weight,
                                             AVG(wingspan) AS avg_wingspan,
                                             AVG(standing_reach) AS avg_standing_reach,
                                             AVG(standing_vertical_leap)
                                             AS avg_standing_vertical_leap,
                                             AVG(max_vertical_leap) AS avg_max_vertical_leap,
                                             AVG(lane_agility_time) AS avg_lane_agility_time,
                                             AVG(modified_lane_agility_time)
                                             AS modified_lane_agility_time,
                                             AVG(three_quarter_sprint) AS avg_three_quarter_sprint,
                                             AVG(bench_press) AS avg_bench_press
                                      FROM draft_combine_stats
                                      GROUP BY season

                                 ")

head(combine_stats_avg)
```

```
  season avg_height avg_weight avg_wingspan avg_standing_reach
1   2000   77.43846   214.4846     81.02308           102.5923
2   2001   78.33013   220.0000     83.11538           103.7872
3   2002   77.66159   217.4512     81.98476           104.0000
4   2003   78.30769   224.1795     82.83654           104.1314
5   2004   77.60443   217.8228     82.39873           104.3481
6   2005   77.40625   215.7037     82.34375           103.9375
  avg_standing_vertical_leap avg_max_vertical_leap avg_lane_agility_time
1                   29.07627              33.24167              11.59328
2                   29.35065              34.16883              11.62592
3                   28.15753              32.53425              11.63781
4                   28.77778              33.63380              11.54600
5                   27.84507              32.41549              11.55648
6                   28.54167              33.16667              11.36141
  modified_lane_agility_time avg_three_quarter_sprint avg_bench_press
1                         NA                 3.323793        9.612903
2                         NA                 3.281299       10.363636
3                         NA                 3.270417       10.220779
4                         NA                 3.262535       11.026667
5                         NA                 3.269155       10.608696
6                         NA                 3.292083       10.493333
```

```
combine_stats_avg$season <- as.integer(combine_stats_avg$season)
```

```
# Pivot to long format
combine_stats_long <- combine_stats_avg %>%
  pivot_longer(
    cols = -season,
    names_to = "metric",
    values_to = "average"
  )

combine_stats_long$season <- as.integer(combine_stats_long$season)
```

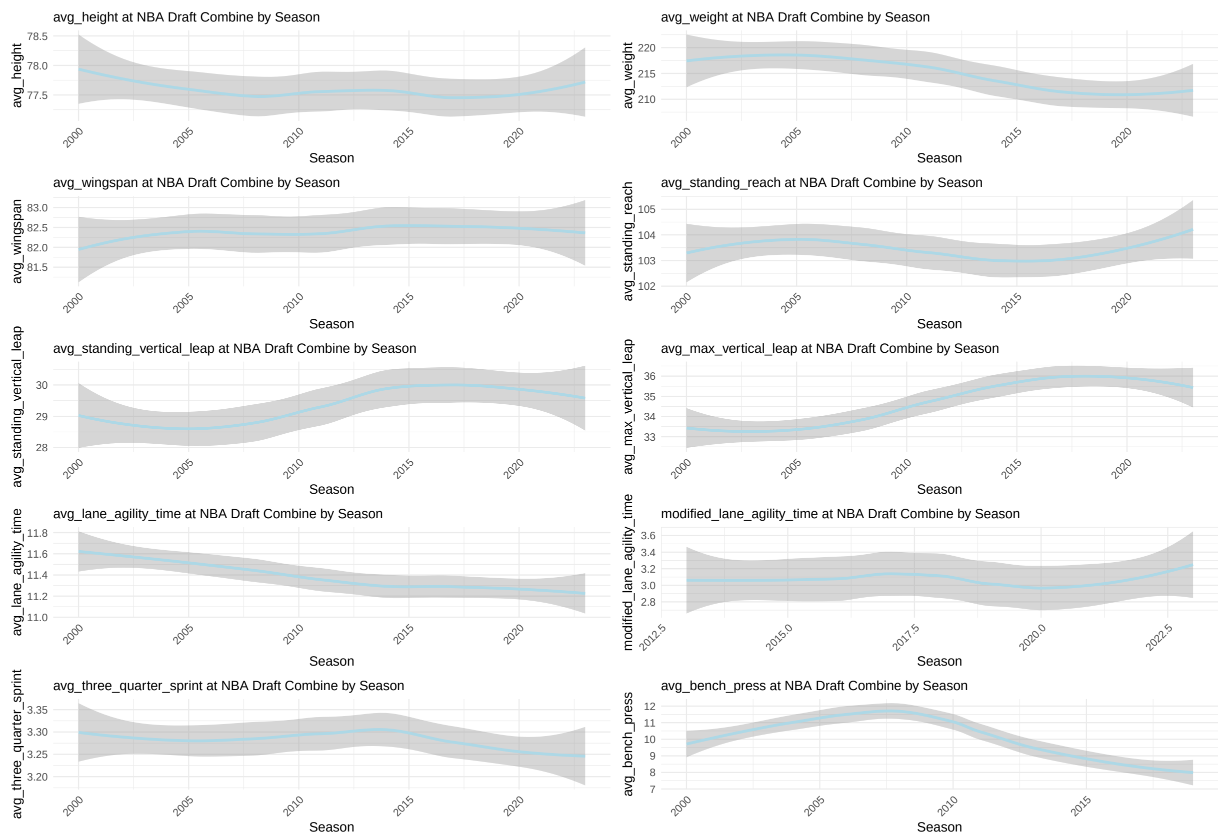
```r
# Plot all metrics in one line chart with color and legend
ggplot(combine_stats_long, aes(x = season, y = average, color = metric)) +
  geom_line(size = .5) +
  # geom_point(size = .25) +
  theme_minimal() +
  labs(
    title = "Average NBA Combine Metrics by Season",
    x = "Season",
    y = "Average Value",
    color = "Metric"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Analysis Part 1 Here we can see all the average values for the combine statistics of all the players from 2000 up until now. Although each line tends to be relatively straight, let's anayze these metrics in their own plots.

```r
generate_plots <- function(data) {
  column_names <- colnames(data)[-1]

  data$season <- as.integer(data$season)
  plot_list <- list()

  # Code to execute
  for (col in column_names) {
```

```
    plot <- ggplot(data, aes(x = season, y = .data[[col]])) +
           geom_smooth(color = "lightblue", size = .5) +
           theme_minimal(base_size = 4.5) +
           labs(
             title = paste(col, "at NBA Draft Combine by Season"),
             x = "Season",
             y = col
           ) +
           theme(axis.text.x = element_text(angle = 45, hjust = 1))

    plot_list[[col]] <- plot
  }
  return(plot_list)
}

# Generate the plots
plot_list <- generate_plots(combine_stats_avg)
Reduce(`+`, plot_list) + plot_layout(ncol = 2)
```



Analysis Part 2 When looking at the individual plots for the metrics, some metrics have a apparent change over time. These most noticable include…

- The average weight of players has went down
- The average max vertical leap of players went up
- The average lane agility time of players went down
- The average three quarter sprint time of players went down
- The average amount of benchpress reps of players went down

This shows how each draft class of NBA players are becoming more athletic. Players are jumping higher, running faster, and becoming more leaner.

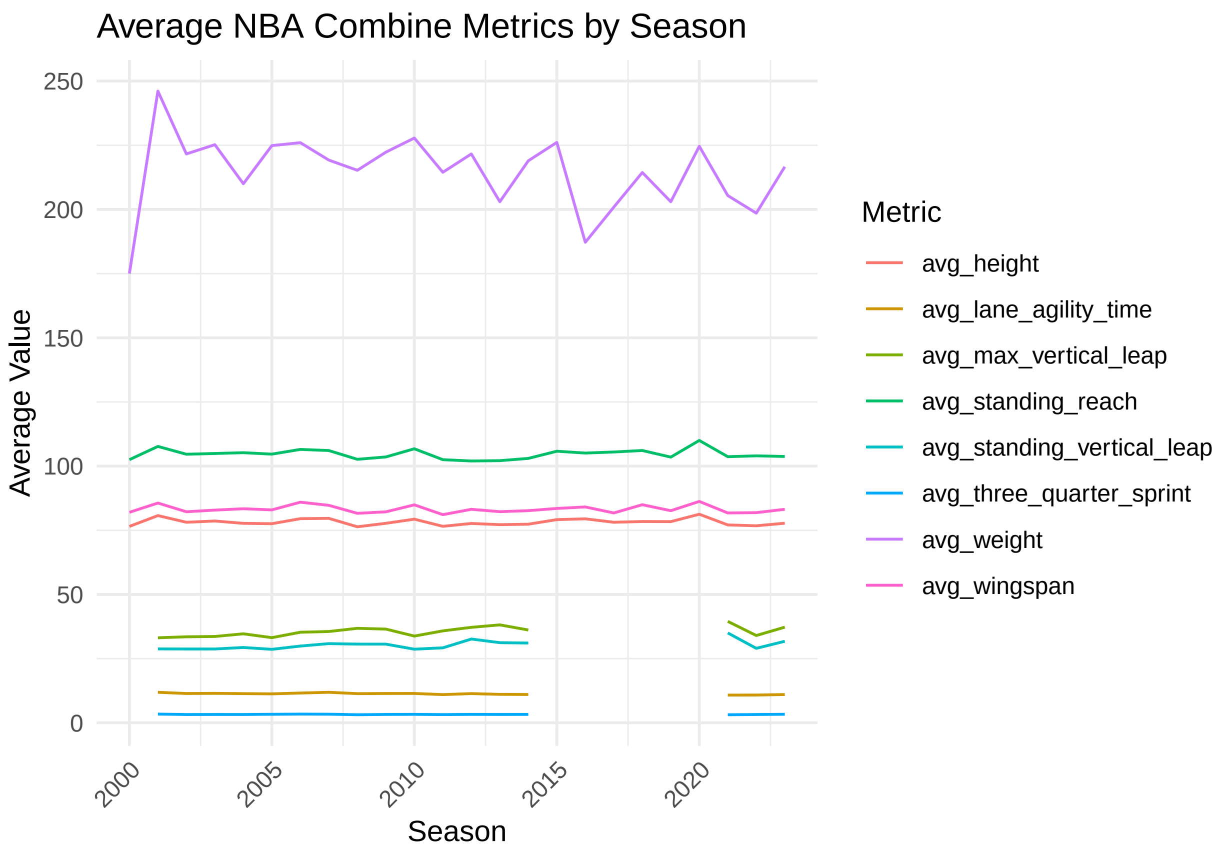
Question 3: What is the Average Draft Combine Statistics for the Top 10 Overall Picks Each Year

```
combine_stats_avg_top_10 <- dbGetQuery(con, "WITH draftStatsMetrics AS (
                                        SELECT dcs.season, dcs.player_id, dcs.player_name,
                                               dh.overall_pick, dcs.position,
                                               dcs.height_wo_shoes, dcs.weight,
                                               dcs.wingspan, dcs.standing_reach,
                                               dcs.standing_vertical_leap,
                                               dcs.max_vertical_leap,
                                               dcs.lane_agility_time,
                                               dcs.three_quarter_sprint
                                        FROM draft_combine_stats AS dcs
                                        INNER JOIN draft_history AS dh
                                        ON dcs.player_id = dh.person_id
                                     )

                                  SELECT season,
                                         AVG(height_wo_shoes) AS avg_height,
                                         AVG(weight) AS avg_weight,
                                         AVG(wingspan) AS avg_wingspan,
                                         AVG(standing_reach) AS avg_standing_reach,
                                         AVG(standing_vertical_leap)
                                          AS avg_standing_vertical_leap,
                                         AVG(max_vertical_leap)
                                          AS avg_max_vertical_leap,
                                         AVG(lane_agility_time)
                                          AS avg_lane_agility_time,
                                         AVG(three_quarter_sprint)
                                          AS avg_three_quarter_sprint
                                  FROM draftStatsMetrics
                                  WHERE overall_pick <= 10
                                  GROUP BY season
                                  ")

head(combine_stats_avg_top_10)
```

```
  season avg_height avg_weight avg_wingspan avg_standing_reach
1   2000   76.50000   175.0000     82.00000           102.5000
2   2001   80.72222   246.1111     85.61111           107.6667
3   2002   78.12500   221.6250     82.21875           104.6250
4   2003   78.61111   225.2222     82.86111           104.9167
5   2004   77.69444   210.0000     83.38889           105.2222
6   2005   77.55556   224.8667     82.94444           104.6667
  avg_standing_vertical_leap avg_max_vertical_leap avg_lane_agility_time
1                         NA                    NA                    NA
2                   28.77778              33.11111              11.88111
3                   28.75000              33.50000              11.40375
4                   28.75000              33.62500              11.46500
5                   29.33333              34.66667              11.36000
6                   28.61111              33.16667              11.26444
  avg_three_quarter_sprint
1                       NA
2                 3.377778
```

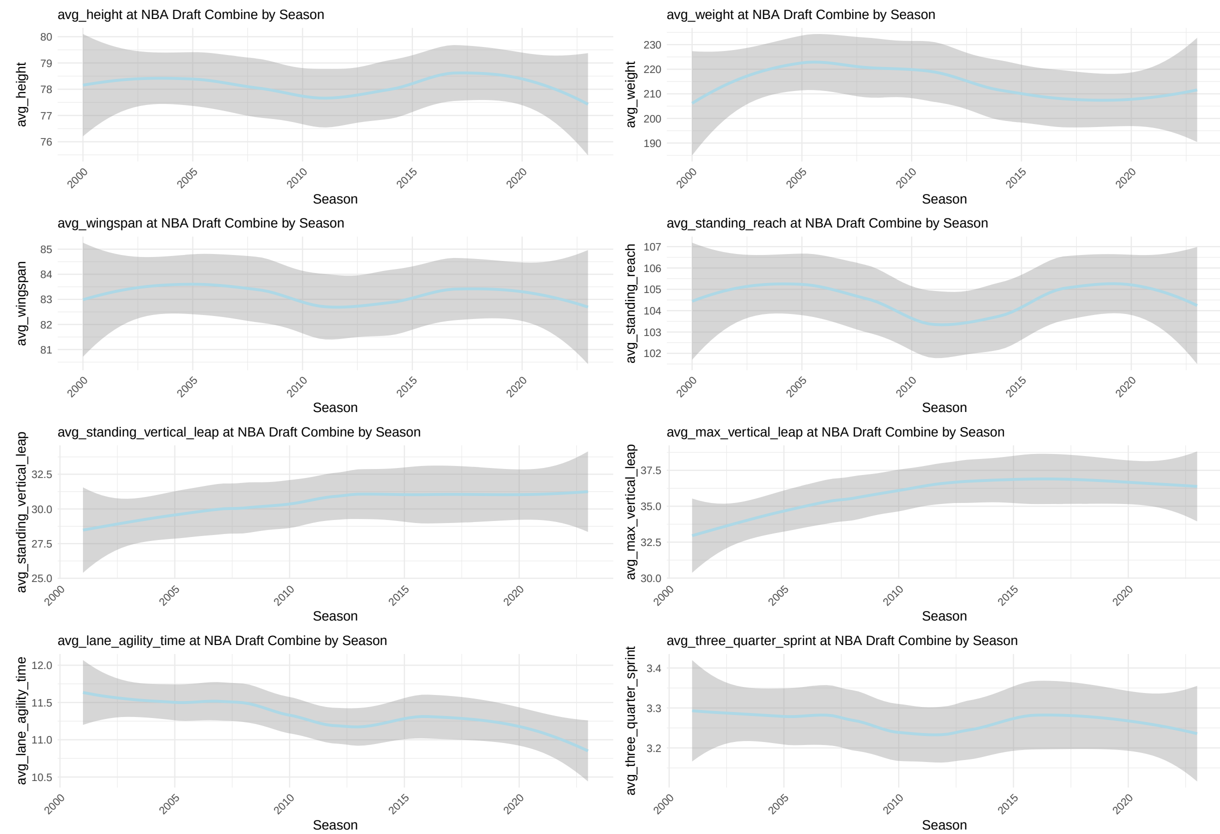
```
3                  3.216250
4                  3.230000
5                  3.226667
6                  3.303333
```

```r
# Pivot to long format
combine_stats_long_top_10 <- combine_stats_avg_top_10 %>%
  pivot_longer(
    cols = -season,
    names_to = "metric",
    values_to = "average"
  )

combine_stats_long_top_10$season <- as.integer(combine_stats_long_top_10$season)

# Plot all metrics in one line chart with color and legend
ggplot(combine_stats_long_top_10, aes(x = season, y = average, color = metric)) +
  geom_line(size = .5) +
  # geom_point(size = .25) +
  theme_minimal() +
  labs(
    title = "Average NBA Combine Metrics by Season",
    x = "Season",
    y = "Average Value",
    color = "Metric"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Analysis Part 1 Again, here is a plot of all the average combine metrics for the Top 10 Players overtime. These lines are not indicative to how sensitive these metrics are. Let's analyze the individual metric plots again.

```
plot_list <- generate_plots(combine_stats_avg_top_10)
Reduce(`+`, plot_list) + plot_layout(ncol = 2)
```



Analysis Part 2 From these plots we can see that the Top 10 Players in the draft class has become athletic in terms of...

- Higher Average Standing Vertical Leap
- Higher Average Max Vertical Leap
- Lower Average Lane Agility Time

Question 4: Do players of certain positions commit more fouls?

```
position_fouls <- dbGetQuery(con, "WITH fouls AS (
                                      SELECT game_id, eventnum, eventmsgtype,
                                             homedescription, player1_id, player1_name,
                                             player1_team_id
                                      FROM play_by_play
                                      WHERE homedescription LIKE '%Foul%'
                                      UNION ALL
                                      SELECT game_id, eventnum, eventmsgtype,
                                             visitordescription, player2_id, player2_name,
                                             player2_team_id
                                      FROM play_by_play
                                      WHERE visitordescription LIKE '%Foul%'
                                   ),

                                   position AS (
                                      SELECT person_id, display_first_last, position
                                      FROM common_player_info
                                   ),

                                   season AS (
                                      SELECT SUBSTRING(season_id, 2) AS current_season,
                                      game_id, season_type
                                      FROM game
                                   ),

                                   foulCounts AS (
                                     SELECT current_season, position, season_type,
                                     COUNT(*) AS fouls
                                     FROM fouls AS f
                                     INNER JOIN position AS p
                                     ON f.player1_id = p.person_id
                                     INNER JOIN season AS s
                                     ON f.game_id = s.game_id
                                     WHERE season_type = 'Regular Season'
                                       AND position != ''
                                     GROUP BY current_season, position
                                   ),

                                   positionGroup AS (
                                     SELECT *
                                     FROM foulCounts
                                   ),

                                   foulSum AS (
                                     SELECT current_season, SUM(fouls) AS fouls, position
                                     FROM positionGroup
                                     GROUP BY current_season, position
                                     ORDER BY current_season, position
                                   ),

                                   foulTotal AS (
                                     SELECT *,
```

```
                                            SUM(fouls) OVER (
                                               PARTITION BY current_season
                                            ) AS fouls_total
                                     FROM foulSum
                                   )

                                   SELECT *,
                                          ((1.0 * fouls) / fouls_total) * 100 AS position_pct
                                   FROM foulTotal
           ")

head(position_fouls)
```

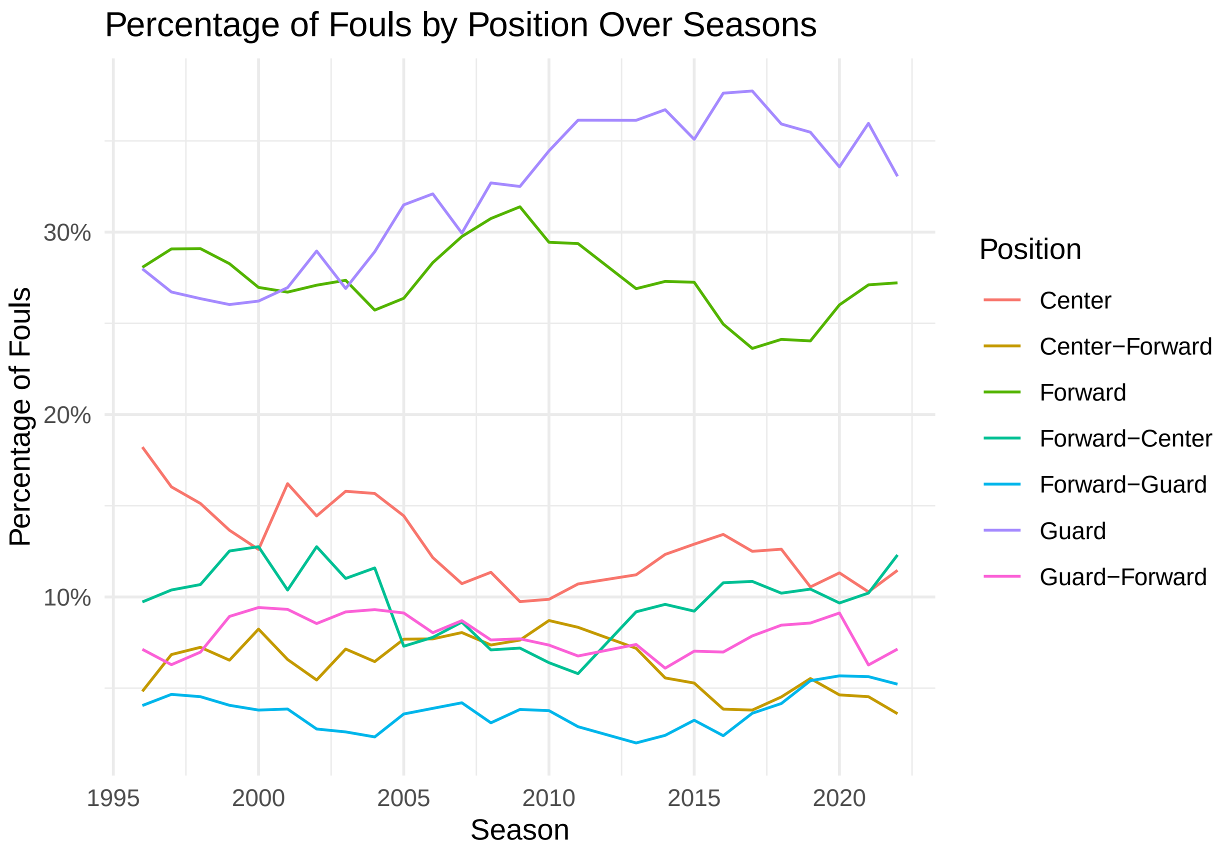
```
  current_season fouls       position fouls_total position_pct
1           1996  3506         Center       19248    18.214879
2           1996   929 Center-Forward       19248     4.826475
3           1996  5403        Forward       19248    28.070449
4           1996  1872 Forward-Center       19248     9.725686
5           1996   779  Forward-Guard       19248     4.047174
6           1996  5387          Guard       19248    27.987323
```

```
position_fouls <- position_fouls %>%
  mutate(current_season = as.integer(current_season))
```

Data Cleaning

```
ggplot(position_fouls, aes(x = current_season, y = position_pct, color = position)) +
  geom_line(size = .5) +
  # geom_point(size = .25) +
  scale_y_continuous(labels = scales::percent_format(scale = 1)) +
  labs(
    title = "Percentage of Fouls by Position Over Seasons",
    x = "Season",
    y = "Percentage of Fouls",
    color = "Position"
  ) +
  theme_minimal()
```

Visualization



Analysis Part 1 Here we can see how each position in the NBA contributes to the total percentage of fouls per year overtime. We can see that although players at the Guard position didn't account for the highest percentage of fouls initially, we can see that over time their contribution to the number of fouls increases drastically. We can see the opposite relationship when analyzing players playing at the center position. They first account for ~18% of fouls initially but has starkly decreased to around ~12% in recent years. We can see these trends better if we generalize these positions to the positions that these players favor (Center, Guard, Forward).

Generalize to Centers, Guards, and Forwards Only

```
position_fouls <- dbGetQuery(con, "WITH fouls AS (
                                     SELECT game_id, eventnum, eventmsgtype, homedescription,
                                            player1_id, player1_name, player1_team_id
                                     FROM play_by_play
                                     WHERE homedescription LIKE '%Foul%'
                                     UNION ALL
                                     SELECT game_id, eventnum, eventmsgtype, visitordescription,
                                            player2_id, player2_name, player2_team_id
                                     FROM play_by_play
                                     WHERE visitordescription LIKE '%Foul%'
                                  ),

                                  position AS (
                                     SELECT person_id, display_first_last, position
                                     FROM common_player_info
                                  ),
```

```
                                        season AS (
                                           SELECT SUBSTRING(season_id, 2) AS current_season,
                                           game_id, season_type
                                           FROM game
                                        ),

                                        foulCounts AS (
                                          SELECT current_season, position, season_type,
                                          COUNT(*) AS fouls
                                          FROM fouls AS f
                                          INNER JOIN position AS p
                                          ON f.player1_id = p.person_id
                                          INNER JOIN season AS s
                                          ON f.game_id = s.game_id
                                          WHERE season_type = 'Regular Season' AND position != ''
                                          GROUP BY current_season, position
                                        ),

                                        positionGroup AS (
                                          SELECT *,
                                                CASE
                                                   WHEN position LIKE 'Guard%' THEN 'Guard'
                                                   WHEN position LIKE 'Forward%' THEN 'Forward'
                                                   WHEN position LIKE 'Center%' THEN 'Center'
                                                END AS position_group
                                          FROM foulCounts
                                        ),

                                        foulSum AS (
                                          SELECT current_season, SUM(fouls) AS fouls,
                                          position_group
                                          FROM positionGroup
                                          GROUP BY current_season, position_group
                                          ORDER BY current_season, position_group
                                        ),

                                        foulTotal AS (
                                          SELECT *,
                                                 SUM(fouls) OVER (
                                                    PARTITION BY current_season
                                                 ) AS fouls_total
                                          FROM foulSum
                                        )

                                        SELECT *,
                                               ((1.0 * fouls) / fouls_total) * 100 AS position_pct
                                        FROM foulTotal
            ")

head(position_fouls)
```

```
  current_season fouls position_group fouls_total position_pct
1           1996  4435         Center       19248     23.04135
```

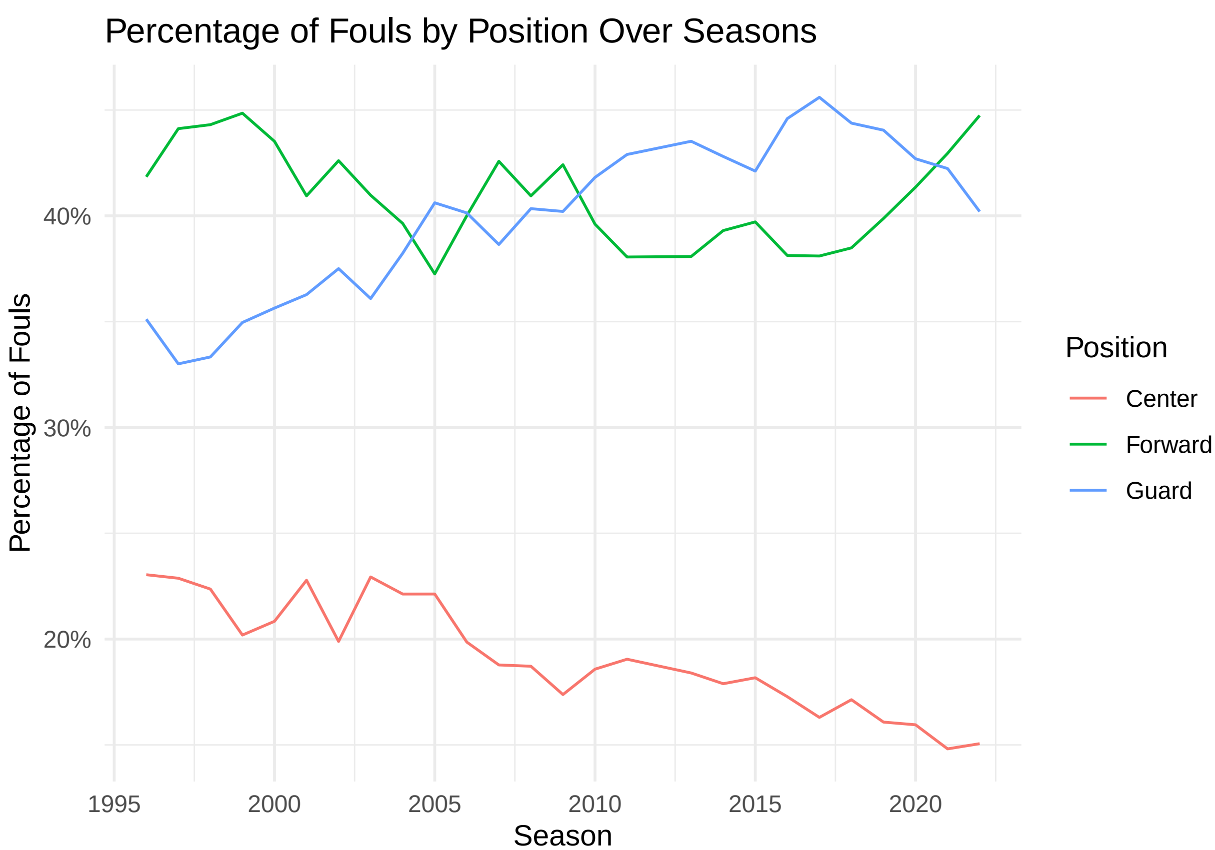
```
2           1996  8054        Forward       19248      41.84331
3           1996  6759          Guard       19248      35.11534
4           1997  4370         Center       19103      22.87599
5           1997  8428        Forward       19103      44.11872
6           1997  6305          Guard       19103      33.00529
```

```r
position_fouls <- position_fouls %>%
  mutate(current_season = as.integer(current_season))
```

Data Cleaning

```r
ggplot(position_fouls, aes(x = current_season, y = position_pct, color = position_group)) +
  geom_line(size = .5) +
  # geom_point(size = .25) +
  scale_y_continuous(labels = scales::percent_format(scale = 1)) +
  labs(
    title = "Percentage of Fouls by Position Over Seasons",
    x = "Season",
    y = "Percentage of Fouls",
    color = "Position"
  ) +
  theme_minimal()
```

Visualization



Analysis Part 2 In this graph we generalized the different types of positions to the main 3 (Center, Forward, and Guard). Here we can clearly see players playing at the Center position have contributed much less to the number of fouls while guards grown to account for the most interchangeable with the forward positon in recent years.

Question 5: Do certain schools have a higher chance of producing players of a certain position?

```r
school_position <- dbGetQuery(con, "WITH playerPosition AS (
                                      SELECT *,
                                             CASE
                                               WHEN position LIKE '%Guard%' THEN 'Guard'
                                               WHEN position LIKE '%Forward%' AND position
                                                NOT LIKE '%Guard%' THEN 'Forward'
                                               WHEN position LIKE '%Center%' AND position
                                                NOT LIKE '%Forward%' THEN 'Center'
                                               ELSE 'Other'
                                             END AS position_group
                                      FROM common_player_info AS cpi
                                      LEFT JOIN draft_history AS dh
                                      ON cpi.person_id = dh.person_id
                                      ),

                                   positionCount AS (
                                      SELECT school, position_group,
                                             COUNT(*) AS n_position,
                                             SUM(COUNT(*)) OVER (
                                             PARTITION BY school
                                             ) AS school_total_draft
                                      FROM playerPosition
                                      WHERE school NOT NULL AND school != ''
                                        AND school NOT LIKE '% %' AND position != ''
                                      GROUP BY school, position_group
                                      ORDER BY n_position, position_group
                                   )

                                   SELECT school, position_group AS position,
                                          n_position, school_total_draft,
                                          ROUND(1.0 * n_position / school_total_draft, 3)
                                            AS pct_school
                                   FROM positionCount
                                   ORDER BY school_total_draft DESC


                             ")
head(school_position)
```

SQL Query

```
    school position n_position school_total_draft pct_school
1 Kentucky   Center         12                 89      0.135
2 Kentucky  Forward         31                 89      0.348
3 Kentucky    Guard         46                 89      0.517
4     UCLA   Center         11                 66      0.167
5     UCLA  Forward         26                 66      0.394
6     UCLA    Guard         29                 66      0.439
```

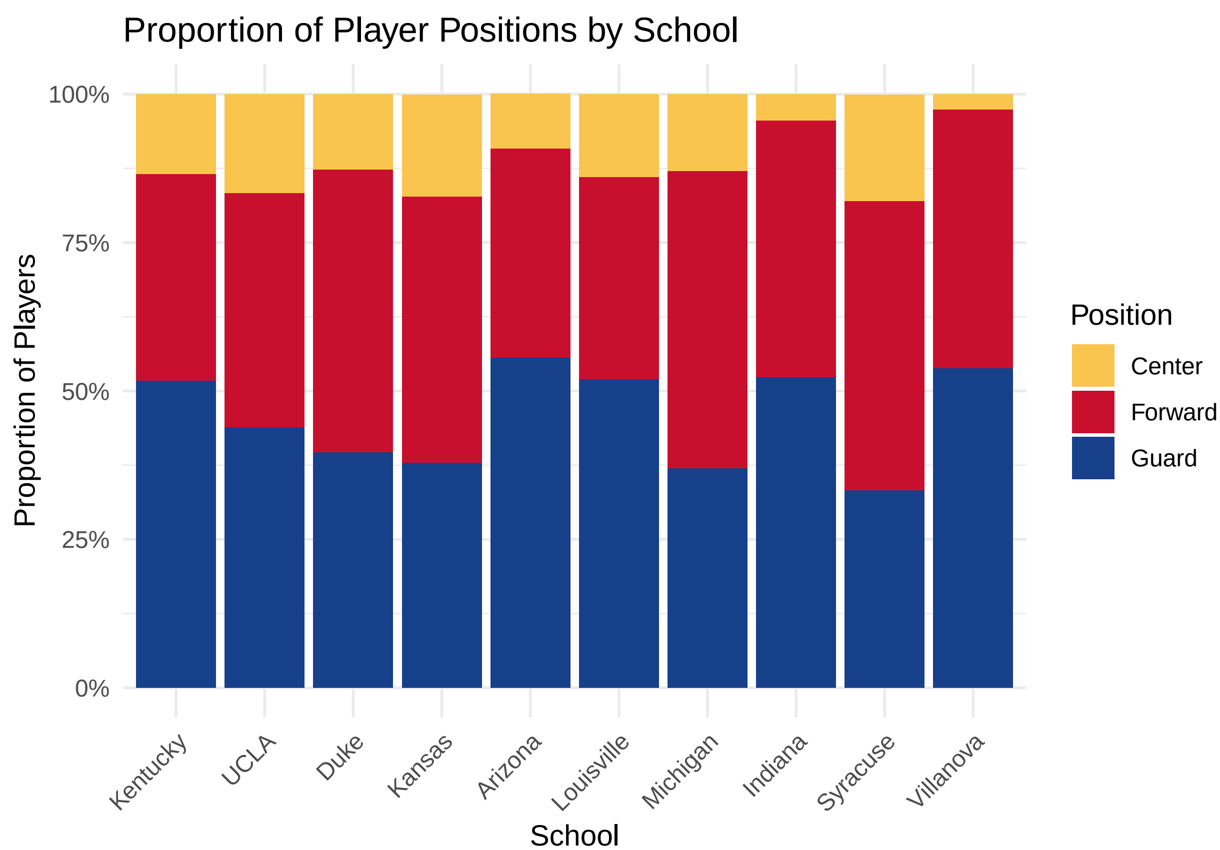
```r
top_schools <- school_position %>%
```

```r
  group_by(school) %>%
  summarise(total = sum(n_position)) %>%
  top_n(10, total) %>%
  pull(school)
```

Top 10 Schools

```r
ggplot(
  school_position %>% filter(school %in% top_schools),
  aes(x = reorder(school, -school_total_draft), y = pct_school, fill = position)
) +
  geom_bar(stat = "identity") +
  labs(
    title = "Proportion of Player Positions by School",
    x = "School", y = "Proportion of Players",
    fill = "Position"
  ) +
  scale_fill_manual(values = c("Guard" = "#17408B", "Forward" = "#c8102e", "Center" = "#FAC54E")) +
  scale_y_continuous(labels = scales::percent_format()) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Visualization: Stacked Bar Chart

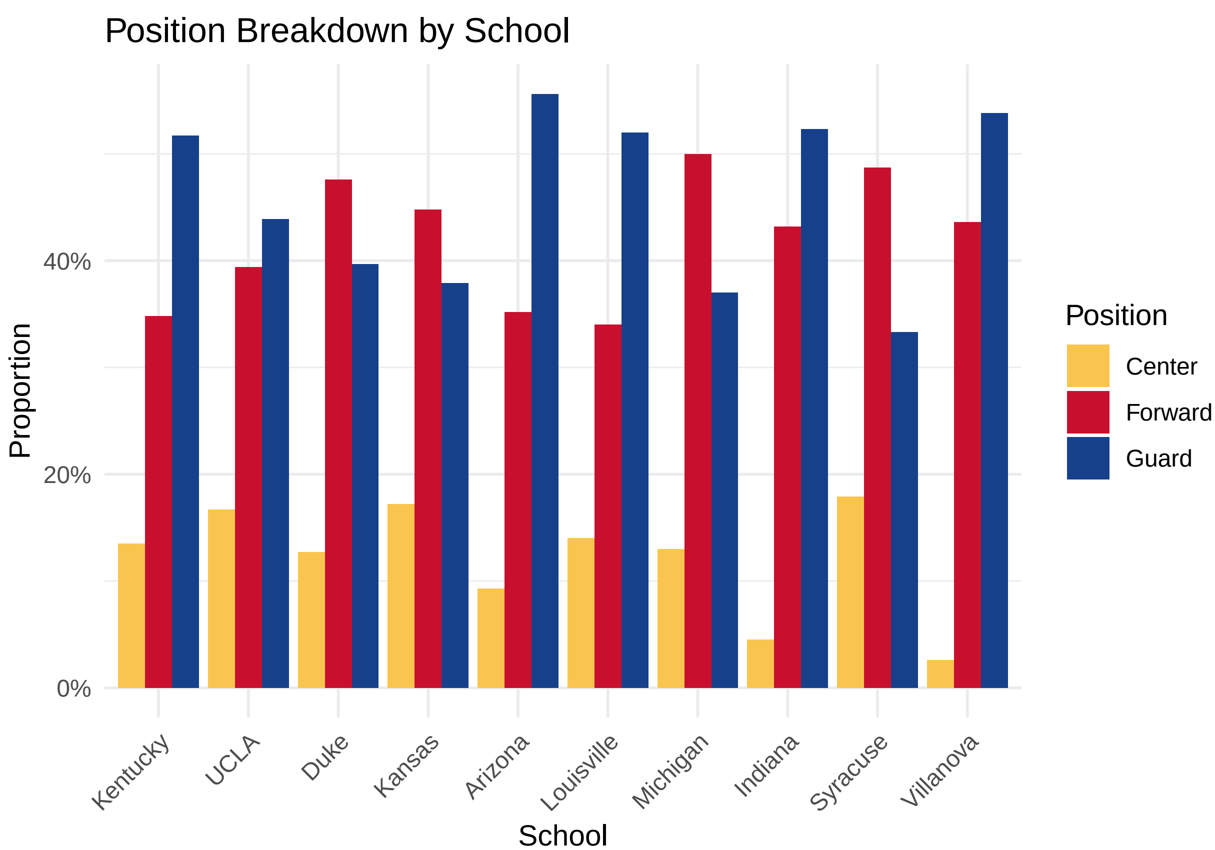


Analysis Part 1 Here we can see the Top 10 schools where players come from when entering the NBA draft. We can see that schools like Arizona, Indiana, Villanova, Louisville, Kentucky output more than 50%

of their athletes as Guards to the draft.

```
ggplot(
  school_position %>% filter(school %in% top_schools),
  aes(x = reorder(school, -school_total_draft), y = pct_school, fill = position)
) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Position Breakdown by School",
    x = "School", y = "Proportion",
    fill = "Position"
  ) +
  scale_fill_manual(values = c("Guard" = "#17408B", "Forward" = "#c8102e", "Center" = "#FAC54E")) +
  scale_y_continuous(labels = scales::percent_format()) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Visualization: Grouped Bar Charts



Analysis Part 2 Here we can see better how the Top 10 schools compare to their players' positions when entering the draft. Arizona outputs the most Guards, Syracuse outputs the most Centers, adn Michigan outputs the mose Forwards.

Question 6: Is there a relationship between the characteristics of a player and the position(s) that they play?

```r
position_combine <- dbGetQuery(con, "WITH playerPosition AS (
                                     SELECT cpi.person_id,
                                            cpi.display_first_last, dcs.position,
                                            CASE
                                              WHEN dcs.position LIKE 'PG%'
                                               THEN 'Point Guard'
                                              WHEN dcs.position LIKE 'PF%'
                                               THEN 'Power Forward'
                                              WHEN dcs.position LIKE 'SG%'
                                               THEN 'Shooting Guard'
                                              When dcs.position LIKE 'SF%'
                                               THEN 'Small Forward'
                                              WHEN dcs.position LIKE 'C%'
                                               THEN 'Center'
                                              ELSE 'Other'
                                            END AS position_group,
                                           dcs.height_wo_shoes, dcs.weight,
                                           dcs.wingspan, dcs.standing_reach,
                                           dcs.standing_vertical_leap, dcs.max_vertical_leap,
                                           dcs.lane_agility_time,
                                           dcs.three_quarter_sprint
                                     FROM common_player_info AS cpi
                                     RIGHT JOIN draft_combine_stats AS dcs
                                     ON cpi.person_id = dcs.player_id
                                     WHERE dcs.position NOT NULL AND dcs.position !=''
                                     )

                               SELECT *
                               FROM playerPosition

                            ")

head(position_combine)
```

SQL Query

```
  person_id display_first_last position position_group height_wo_shoes weight
1   1630173   Precious Achiuwa       PF  Power Forward           79.50    234
2    203112         Quincy Acy       PF  Power Forward           78.50  223.8
3    203500       Steven Adams        C         Center           82.75  254.5
4   1630534       Ochai Agbaji       SG Shooting Guard           76.50 214.40
5   1630534       Ochai Agbaji       SG Shooting Guard           76.50 216.80
6    200772       Maurice Ager    SG-PG Shooting Guard           75.25    203
  wingspan standing_reach standing_vertical_leap max_vertical_leap
1    84.75          108.5                     NA                NA
2    86.75          106.5                   32.0              37.0
3    88.50          109.5                   28.5              33.0
4    82.00          103.5                   32.0              41.5
5    82.25          104.0                   32.0              39.0
6    79.75          101.5                   29.5              35.0
  lane_agility_time three_quarter_sprint
```

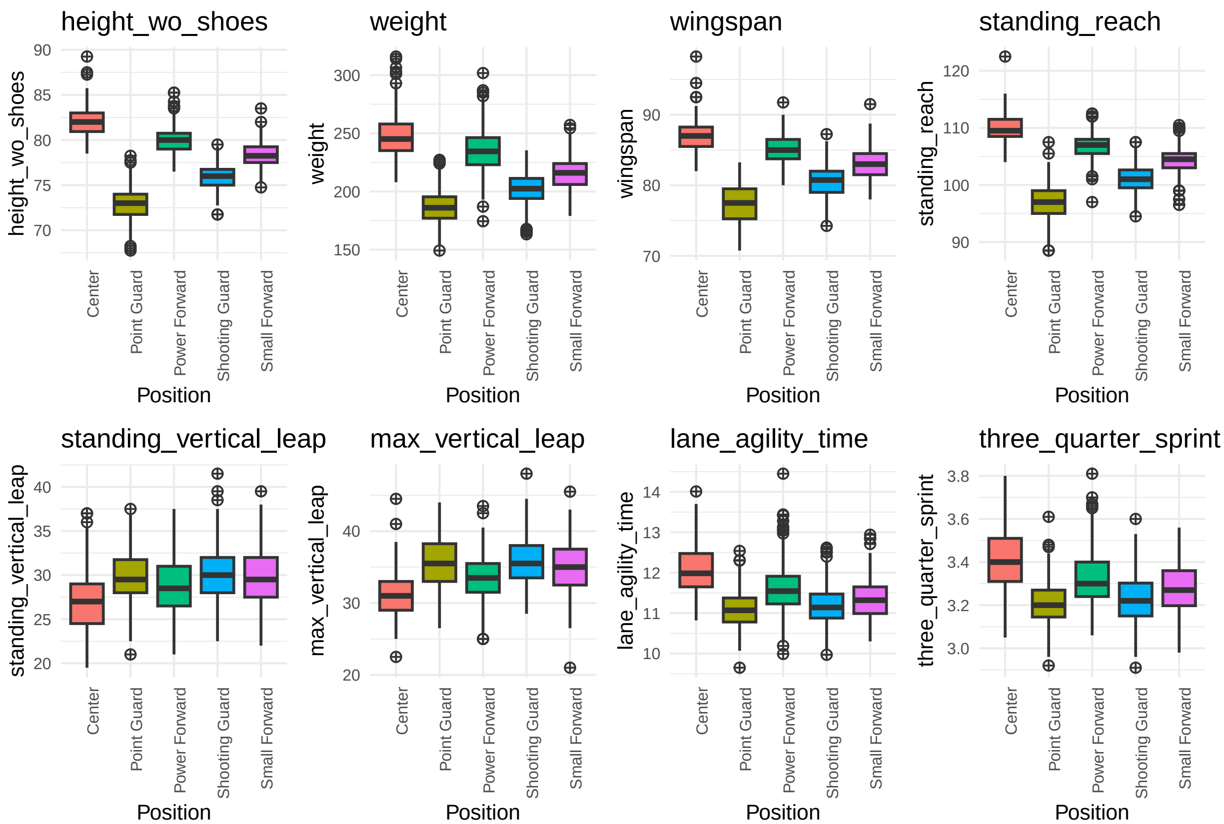
```
1               NA                    NA
2            10.48                  3.28
3            11.85                  3.40
4            10.88                  3.13
5            10.77                  3.13
6            11.73                  3.22
```

```
position_combine_cleaned <- position_combine %>%
  drop_na(height_wo_shoes, weight, wingspan, standing_reach, standing_vertical_leap,
          max_vertical_leap, lane_agility_time, three_quarter_sprint) %>%
    mutate(across(c(height_wo_shoes, weight, wingspan, standing_reach,
                    standing_vertical_leap, max_vertical_leap,
                    lane_agility_time, three_quarter_sprint),
                  ~ as.numeric(.)))

nrow(position_combine_cleaned)
```

Visualization: Box-Whisker Plots

```
[1] 1375
```

```
make_boxplots <- function(data, position_col, variables) {

  plot_list <- list()

  for (var in variables) {
    p <- ggplot(data, aes_string(x = position_col, y = var, fill = position_col)) +
      geom_boxplot(outlier.shape = 10) +
      labs(title = var,
           x = "Position",
           y = var) +
      theme_minimal(base_size = 8) +  # slightly smaller text for better layout
      theme(axis.text.x = element_text(angle = 90, hjust = 1),
            legend.position = "none")

    plot_list[[var]] <- p
  }

  # Combine plots using patchwork
  combined_plot <- wrap_plots(plot_list, ncol = 4)
  print(combined_plot)
}

vars_to_plot <- c("height_wo_shoes", "weight", "wingspan",
                  "standing_reach", "standing_vertical_leap",
                  "max_vertical_leap", "lane_agility_time", "three_quarter_sprint")

make_boxplots(position_combine_cleaned, position_col = "position_group", variables = vars_to_plot)
```



Analysis Here we can see the metrics that divide the different playing positions. Select positions have many metrics that indicate their ideal position. Centers tend to be the tallest, heaviest, possess the largest wingspan, tallest standing reach, lowest standing vertical leap, lowest max vertical leap, slowest lane agility time, and the slowest three quarter sprint. Guards on the other hand tend to be the shortest, weight the least, have the smallest wingspan, have the smallest standing_reach, have the highest max vertical leap, have the fastest lane agility time, and the fastest three quarter sprint.

This graphs support the skill set needed by each playing position. Centers typically are expected to play more within the paint, grabbing rebounds by utilizing their big frame to their advantage. Point Guards use their smaller frame to speed past opponents and look for opportunities to execute plays for themselves and their team.

Question 7: Which Team has the Best Regualar Season Winning Percentage

```
best_teams <- dbGetQuery(con, "WITH combinedTeam AS (
                     SELECT SUBSTRING(season_id, 2) AS season,
                     team_name_home AS team_name, wl_home,
                        CASE
                          WHEN wl_home = 'W' THEN 1
                          ELSE 0
                        END AS win_count,
                        CASE
                          WHEN wl_home = 'L' THEN 1
                          ELSE 0
                        END AS loss_count,
                        season_type
                     FROM game
                     UNION ALL
                     SELECT SUBSTRING(season_id, 2) AS season,
                     team_name_away AS team_name, wl_away,
                        CASE
                          WHEN wl_away = 'W' THEN 1
                          ELSE 0
                        END AS win_count,
                        CASE
                          WHEN wl_away = 'L' THEN 1
                          ELSE 0
                        END AS loss_count,
                        season_type
                     FROM game
                ),

                teamRecords AS (
                    SELECT season, team_name, SUM(win_count) AS season_wins,
                    SUM(loss_count) AS season_losses
                    FROM combinedTeam
                    WHERE season_type = 'Regular Season'
                    GROUP BY season, team_name
                )

                SELECT *, (1.0 * season_wins ) / (season_wins + season_losses) AS win_pct
                FROM teamRecords
                ORDER BY win_pct DESC
                LIMIT 10
           ")

best_teams
```

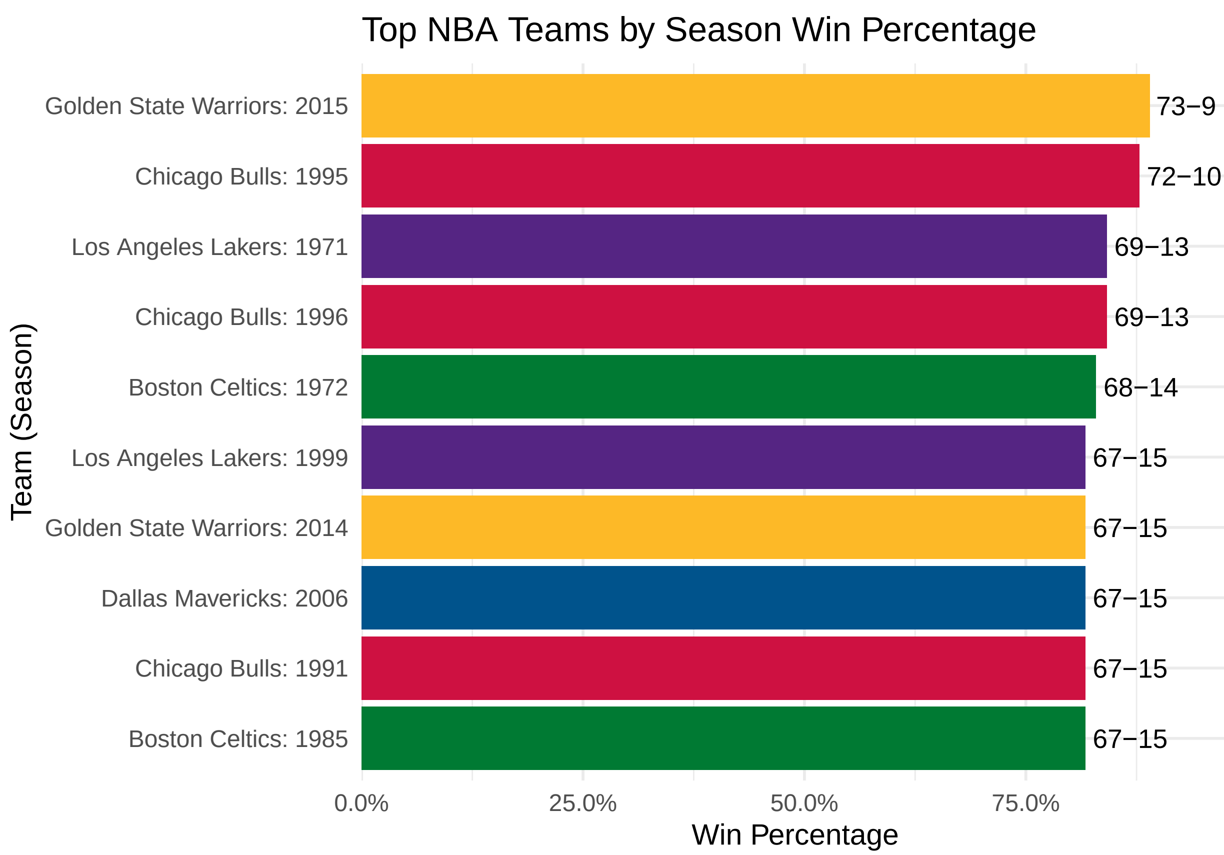
SQL Query

```
   season             team_name season_wins season_losses   win_pct
1    2015 Golden State Warriors          73             9 0.8902439
2    1995         Chicago Bulls          72            10 0.8780488
3    1971    Los Angeles Lakers          69            13 0.8414634
4    1996         Chicago Bulls          69            13 0.8414634
5    1972        Boston Celtics          68            14 0.8292683
```

```
6    1985        Boston Celtics          67            15 0.8170732
7    1991         Chicago Bulls          67            15 0.8170732
8    1999    Los Angeles Lakers          67            15 0.8170732
9    2006      Dallas Mavericks          67            15 0.8170732
10   2014 Golden State Warriors          67            15 0.8170732
```

```r
# Order team names by win percentage
best_teams$team_label <- paste(best_teams$team_name, best_teams$season, sep = ": ")
best_teams$record <- paste0(best_teams$season_wins, "-", best_teams$season_losses)

team_colors <- c(
  "Golden State Warriors" = "#FDB927",
  "Chicago Bulls" = "#CE1141",
  "Los Angeles Lakers" = "#552583",
  "Boston Celtics" = "#007A33",
  "Dallas Mavericks" = "#00538C"
)



# Plot
ggplot(best_teams, aes(x = reorder(team_label, win_pct), y = win_pct, fill = team_name)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = record),
            hjust = -0.1,
            color = "black",
            size = 3.5) +
  coord_flip() +
  scale_fill_manual(values = team_colors) +
  labs(title = "Top NBA Teams by Season Win Percentage",
       x = "Team (Season)",
       y = "Win Percentage") +
  scale_y_continuous(labels = scales::percent_format(accuracy = 0.1), expand = expansion(mult = c(0, 0.
  theme_minimal() +
  theme(legend.position = "none")
```

Visualization: Bar Chart



Analysis Here we can see the top 10 teams with the best regular season winning percentage of all time. The best team of all time is the 2015 Golden State Warriors.

Question 8: Which Teams has the Best Regualar Season Winning Percentage Over a Four Year Period?

```
best_team_4_years <- dbGetQuery(con, "WITH combinedTeam AS (
                     SELECT SUBSTRING(season_id, 2) AS season,
                     team_name_home AS team_name, wl_home,
                        CASE
                          WHEN wl_home = 'W' THEN 1
                          ELSE 0
                        END AS win_count,
                        CASE
                          WHEN wl_home = 'L' THEN 1
                          ELSE 0
                        END AS loss_count,
                        season_type
                     FROM game
                     UNION ALL
                     SELECT SUBSTRING(season_id, 2) AS season,
                     team_name_away AS team_name, wl_away,
                        CASE
                          WHEN wl_away = 'W' THEN 1
                          ELSE 0
                        END AS win_count,
                        CASE
                          WHEN wl_away = 'L' THEN 1
                          ELSE 0
                        END AS loss_count,
                        season_type
                     FROM game
                ),

                teamRecords AS (
                    SELECT season, team_name, SUM(win_count) AS season_wins,
                    SUM(loss_count) AS season_losses
                    FROM combinedTeam
                    WHERE season_type = 'Regular Season'
                    GROUP BY season, team_name
                ),

                fourYearPeriod AS(
                    SELECT *, (1.0 * season_wins ) / (season_wins + season_losses) AS win_pct,
                           SUM(season_wins) OVER (
                              PARTITION BY team_name
                              ORDER BY season ASC, team_name ASC
                              ROWS BETWEEN 3 PRECEDING AND CURRENT ROW
                           ) AS season_wins_4_year,
                           SUM(season_losses) OVER (
                              PARTITION BY team_name
                              ORDER BY season ASC, team_name ASC
                              ROWS BETWEEN 3 PRECEDING AND CURRENT ROW
                           ) AS season_losses_4_year,
                          COUNT(*) OVER (
                              PARTITION BY team_name
                              ORDER BY season ASC, team_name ASC
```

```
                              ROWS BETWEEN 3 PRECEDING AND CURRENT ROW
                           ) AS count_years
                    FROM teamRecords
                )

                SELECT season, team_name, season_wins, season_losses,
                       win_pct, season_wins_4_year, season_losses_4_year,
                       (1.0 * season_wins_4_year) / (season_wins_4_year + season_losses_4_year)
                          AS win_pct_4_year,
                       count_years
                FROM fourYearPeriod
                WHERE count_years = 4
                ORDER BY win_pct_4_year DESC
                LIMIT 10

           ")

best_team_4_years
```

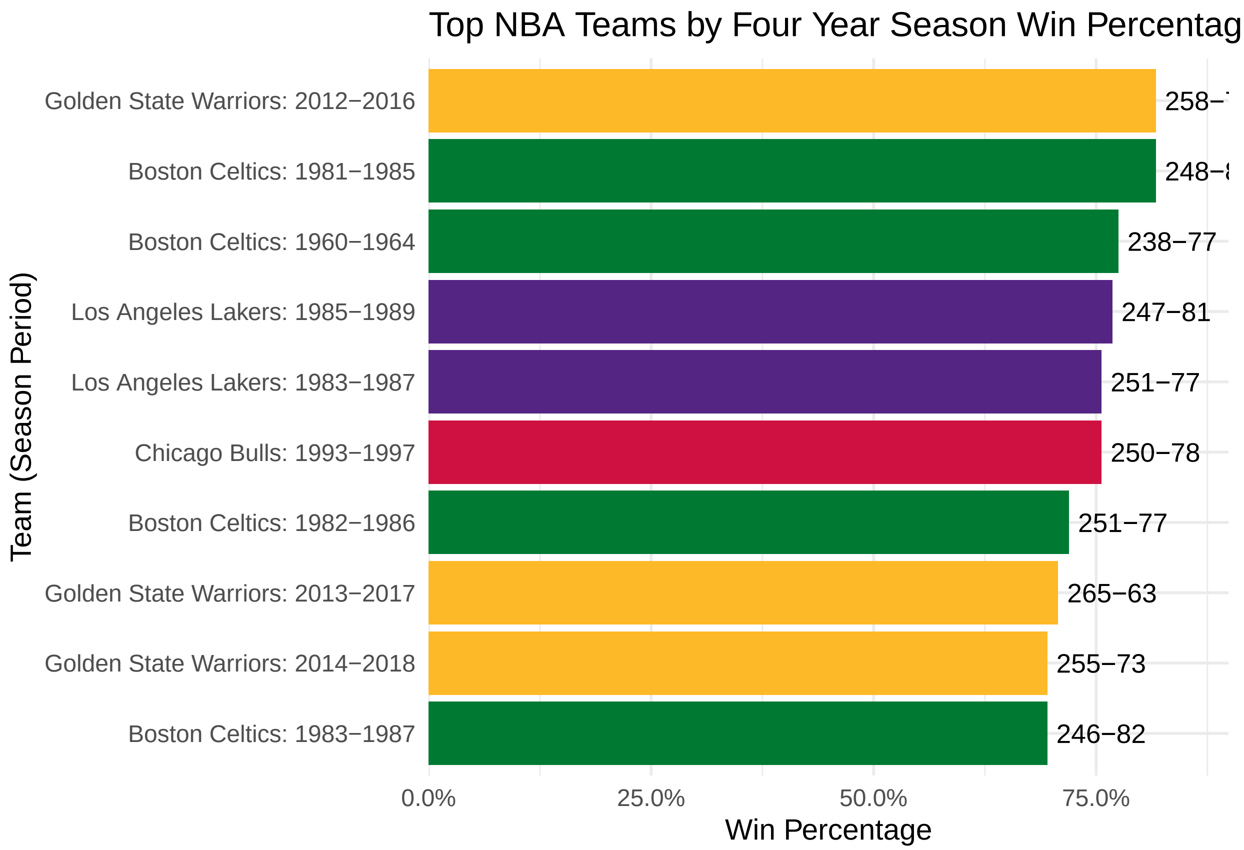
```
   season             team_name season_wins season_losses   win_pct
1    2017 Golden State Warriors          58            24 0.7073171
2    2016 Golden State Warriors          67            15 0.8170732
3    2018 Golden State Warriors          57            25 0.6951220
4    1986        Boston Celtics          59            23 0.7195122
5    1987    Los Angeles Lakers          62            20 0.7560976
6    1997         Chicago Bulls          62            20 0.7560976
7    1985        Boston Celtics          67            15 0.8170732
8    1964        Boston Celtics          62            18 0.7750000
9    1989    Los Angeles Lakers          63            19 0.7682927
10   1987        Boston Celtics          57            25 0.6951220
   season_wins_4_year season_losses_4_year win_pct_4_year count_years
1                 265                   63      0.8079268           4
2                 258                   70      0.7865854           4
3                 255                   73      0.7774390           4
4                 251                   77      0.7652439           4
5                 251                   77      0.7652439           4
6                 250                   78      0.7621951           4
7                 248                   80      0.7560976           4
8                 238                   77      0.7555556           4
9                 247                   81      0.7530488           4
10                246                   82      0.7500000           4
```

```r
# First, ensure season column is integer
best_team_4_years <- best_team_4_years %>%
  mutate(season = as.integer(season))

# Now transform season into "start-end" format
best_team_4_years <- best_team_4_years %>%
  mutate(season_window = paste0(season - 4, "-", season))

best_team_4_years
```

Transform Data

```
   season            team_name season_wins season_losses   win_pct
1    2017 Golden State Warriors          58            24 0.7073171
2    2016 Golden State Warriors          67            15 0.8170732
3    2018 Golden State Warriors          57            25 0.6951220
4    1986        Boston Celtics          59            23 0.7195122
5    1987    Los Angeles Lakers          62            20 0.7560976
6    1997         Chicago Bulls          62            20 0.7560976
7    1985        Boston Celtics          67            15 0.8170732
8    1964        Boston Celtics          62            18 0.7750000
9    1989    Los Angeles Lakers          63            19 0.7682927
10   1987        Boston Celtics          57            25 0.6951220
   season_wins_4_year season_losses_4_year win_pct_4_year count_years
1                 265                   63      0.8079268           4
2                 258                   70      0.7865854           4
3                 255                   73      0.7774390           4
4                 251                   77      0.7652439           4
5                 251                   77      0.7652439           4
6                 250                   78      0.7621951           4
7                 248                   80      0.7560976           4
8                 238                   77      0.7555556           4
9                 247                   81      0.7530488           4
10                246                   82      0.7500000           4
   season_window
1      2013-2017
2      2012-2016
3      2014-2018
4      1982-1986
5      1983-1987
6      1993-1997
7      1981-1985
8      1960-1964
9      1985-1989
10     1983-1987
```

```r
# Order team names by win percentage
best_team_4_years$team_label <- paste(best_team_4_years$team_name, best_team_4_years$season_window, sep
best_team_4_years$record <- paste0(best_team_4_years$season_wins_4_year, "-", best_team_4_years$season_

# Plot
ggplot(best_team_4_years, aes(x = reorder(team_label, win_pct), y = win_pct, fill = team_name)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = record),
            hjust = -0.1,
            color = "black",
            size = 3.5) +
  coord_flip() +
  scale_fill_manual(values = team_colors) +
  labs(title = "Top NBA Teams by Four Year Season Win Percentage",
       x = "Team (Season Period)",
       y = "Win Percentage") +
  scale_y_continuous(labels = scales::percent_format(accuracy = 0.1), expand = expansion(mult = c(0, 0.
  theme_minimal() +
  theme(legend.position = "none")
```



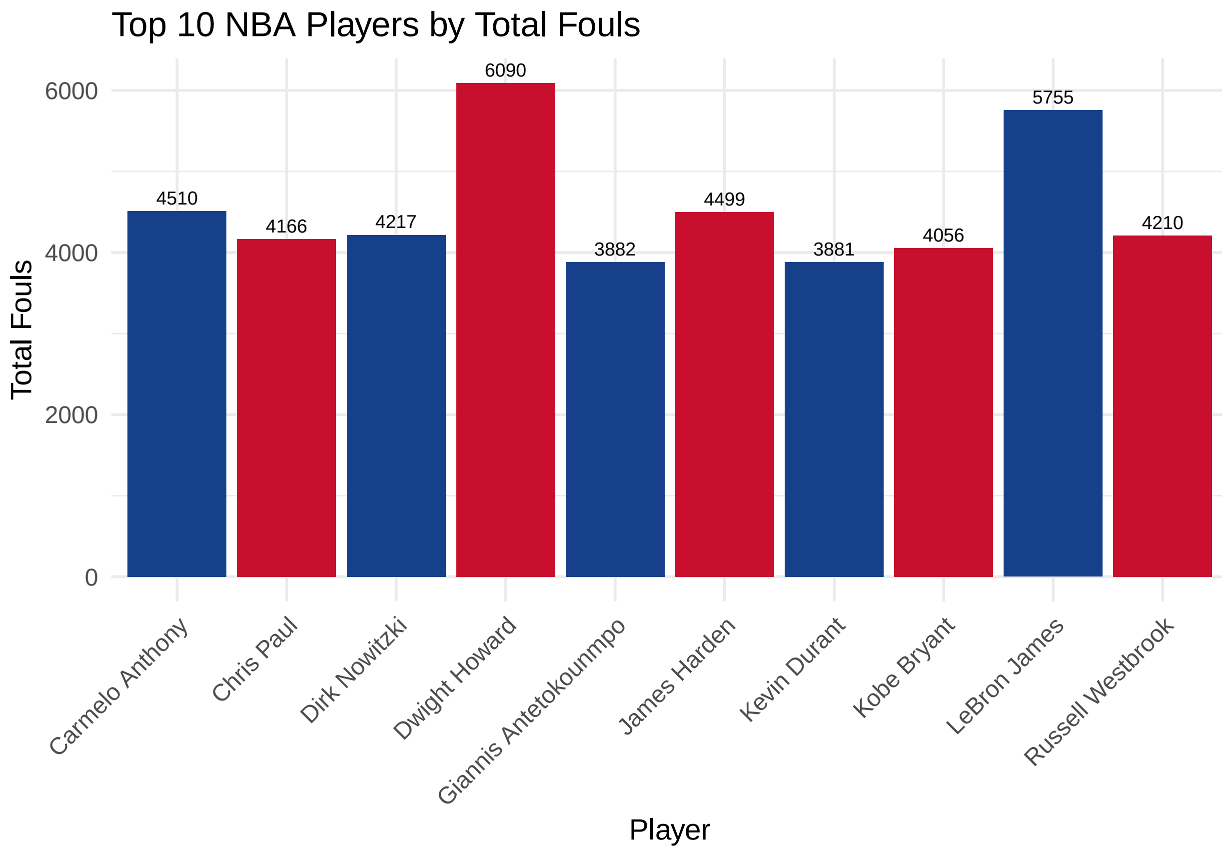
Analysis Here we can see the top 10 teams by a four year period winning percentage of all time. The Golden State Warriors claim the highest winning percentage with a 258-70 record.

Question 9: What Player has the Most Fouls of All Time

```r
playerFouls <- dbGetQuery(con, "WITH fouls AS (
                                  SELECT game_id, eventnum, eventmsgtype,
                                         homedescription, player1_id, player1_name,
                                         player1_team_id
                                  FROM play_by_play
                                  WHERE homedescription LIKE '%Foul%'
                                  UNION ALL
                                  SELECT game_id, eventnum, eventmsgtype,
                                         visitordescription, player2_id, player2_name,
                                         player2_team_id
                                  FROM play_by_play
                                  WHERE visitordescription LIKE '%Foul%'
                              )

                              SELECT player1_id AS player_id, player1_name AS player_name, \
                                     COUNT(*) AS foul_count
                              FROM fouls
                              WHERE player_id != '0'
                              GROUP BY player1_id, player1_name
                              ORDER BY COUNT(*) DESC
                              LIMIT 10
           ")

playerFouls
```

```
   player_id          player_name foul_count
1       2730         Dwight Howard       6090
2       2544          LeBron James       5755
3       2546       Carmelo Anthony       4510
4     201935          James Harden       4499
5       1717         Dirk Nowitzki       4217
6     201566     Russell Westbrook       4210
7     101108            Chris Paul       4166
8        977           Kobe Bryant       4056
9     203507 Giannis Antetokounmpo       3882
10    201142          Kevin Durant       3881
```

```r
# Create alternating color vector
alternating_colors <- rep(c("#17408B", "#c8102e"), length.out = nrow(playerFouls))


# Create the bar chart
ggplot(playerFouls, aes(x = player_name, y = foul_count, fill = player_name)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = foul_count), vjust = -0.5, size = 2.5) +
  scale_fill_manual(values = alternating_colors) +
  labs(title = "Top 10 NBA Players by Total Fouls",
       x = "Player",
       y = "Total Fouls") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none")
```



Analysis Here we can see the player that has committed the most fouls of all time is Dwight Howard.