Exploratory Data Analysis

James Martinez

March 12, 2019

Introduction

This report is a continuation of the All-NBA Team Capstone project, which utilizes historical NBA statistics from 1937 to 2012 (https://www.kaggle.com/open-source-sports/mens-professional-basketball) to predict All-NBA Teams. It will cover the exploratory data analysis and statistical analysis of the cleaned *players.joined* data frame, which was exported as *players_clean.csv*.

The data cleaning report can be found in *Data Wrangling.Rmd* in my capstone project repository (https://github.com/martij222/capstone-project).

Importing the Cleaned Data

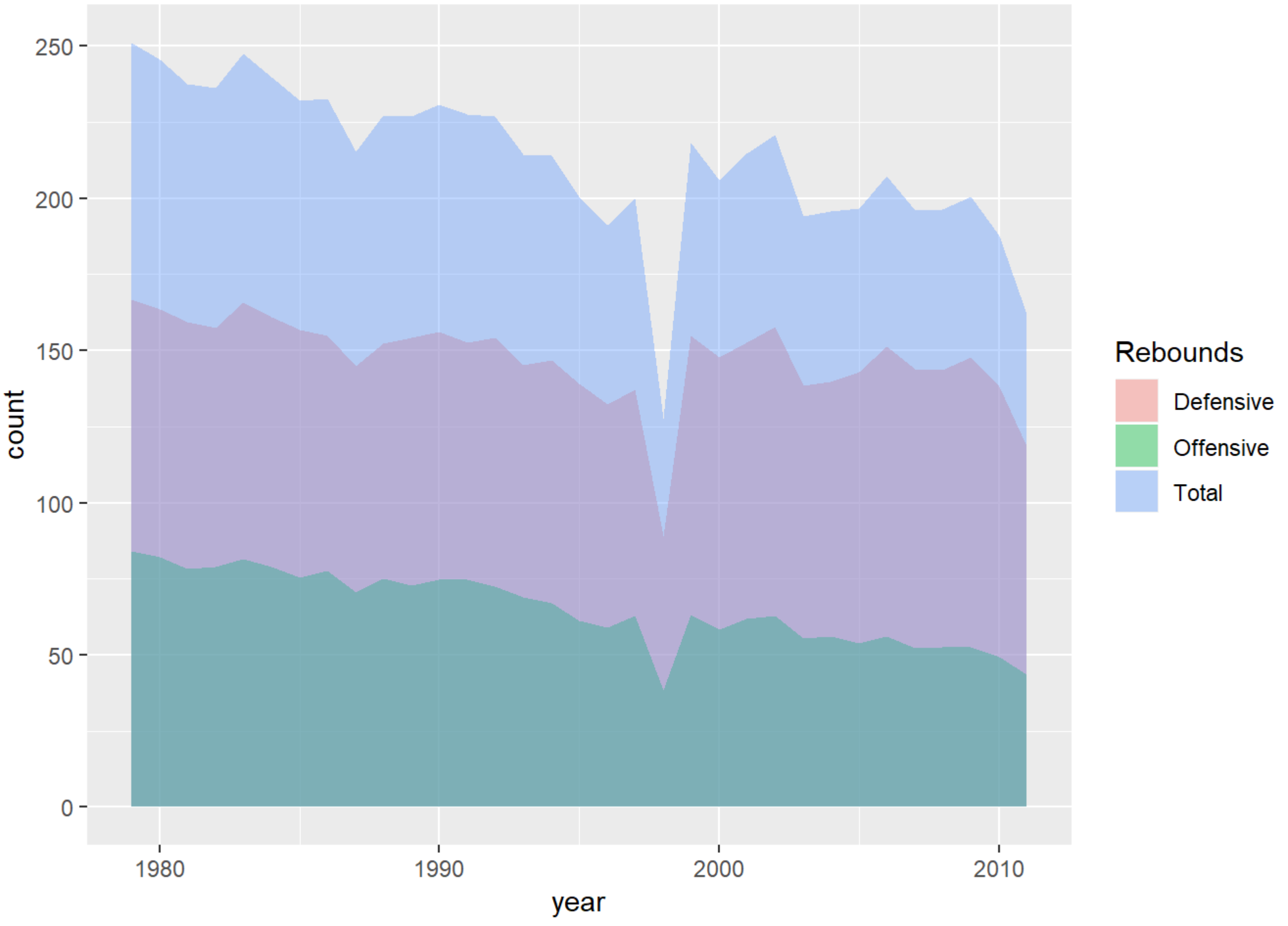
```
# Store clean data as "players"
players <- as_tibble(read.csv("players_clean.csv"))
players
```

```
## # A tibble: 14,577 x 31
##    playerID  year tmID     GP minutes points oRebounds dRebounds rebounds
##    <fct>    <int> <fct> <int>   <int>  <int>     <int>     <int>    <int>
##  1 abdulka~  1979 LAL      82    3143   2034       190       696      886
##  2 aberntо~  1979 GSW      67    1222    362        62       129      191
##  3 adamsal~  1979 PHO      75    2168   1118       158       451      609
##  4 archina~  1979 BOS      80    2864   1131        59       138      197
##  5 awtrede~  1979 CHI      26     560     86        29        86      115
##  6 bailegu~  1979 WSB      20     180     38         6        22       28
##  7 baileja~  1979 SEA      67     726    312        71       126      197
##  8 ballagr~  1979 WSB      82    2438   1277       240       398      638
##  9 bantomi~  1979 IND      77    2330    908       192       264      456
## 10 barnema~  1979 SDC      20     287     64        34        43       77
## # ... with 14,567 more rows, and 22 more variables: assists <int>,
## #   steals <int>, blocks <int>, turnovers <int>, PF <int>,
## #   fgAttempted <int>, fgMade <int>, ftAttempted <int>, ftMade <int>,
## #   threeAttempted <int>, threeMade <int>, center <int>, forward <int>,
## #   guard <int>, award <fct>, allDefFirstTeam <int>,
## #   allDefSecondTeam <int>, allNBAFirstTeam <int>, allNBASecondTeam <int>,
## #   MVP <int>, defPOTY <int>, allstar <int>
```

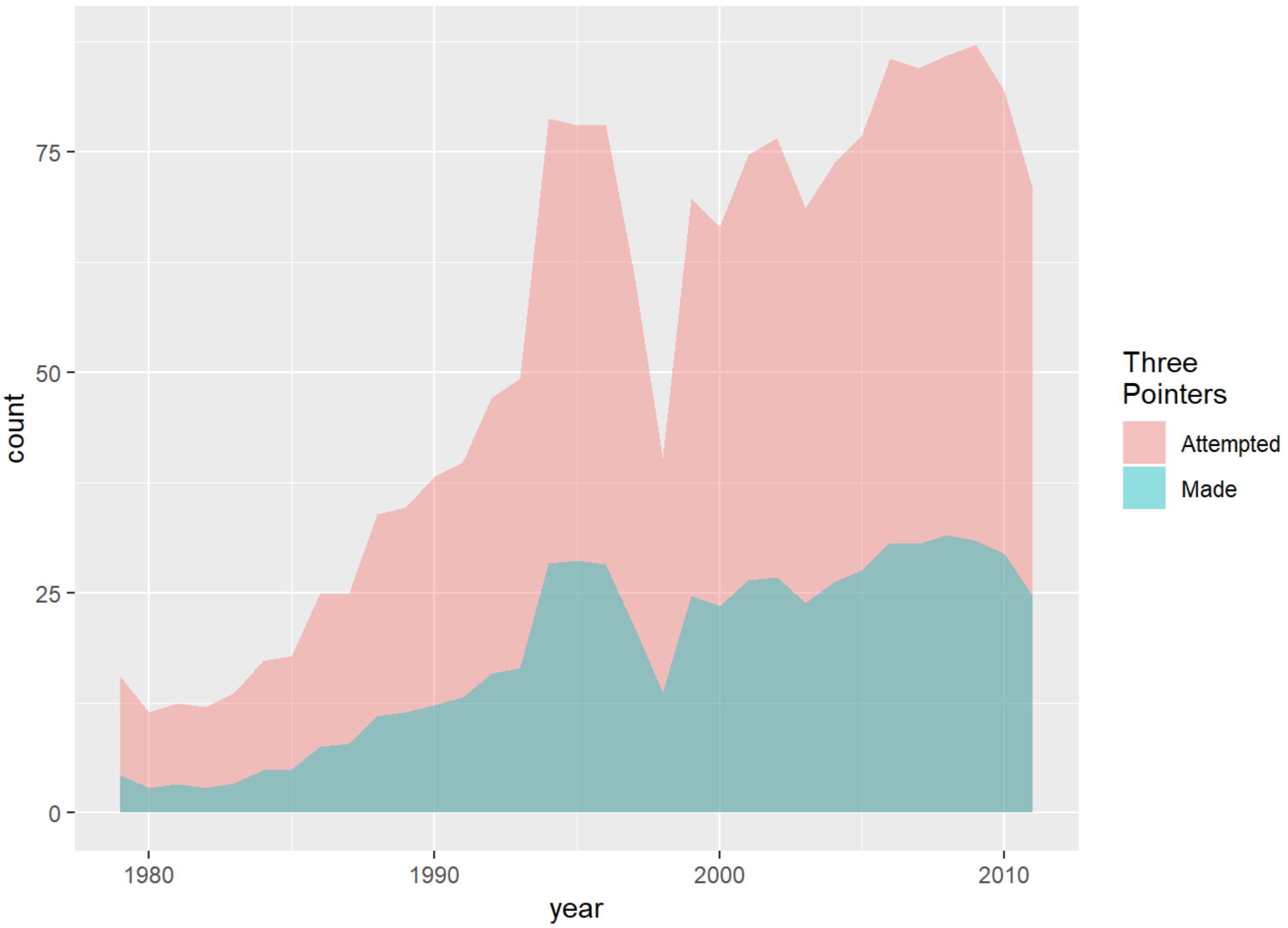
Exploratory Data Analysis

We can get a feel of each variable by plotting the means of each season. Rebounds, 3-pointers, field goals, and free throws are plotted together as fills to get an idea of the proportions.

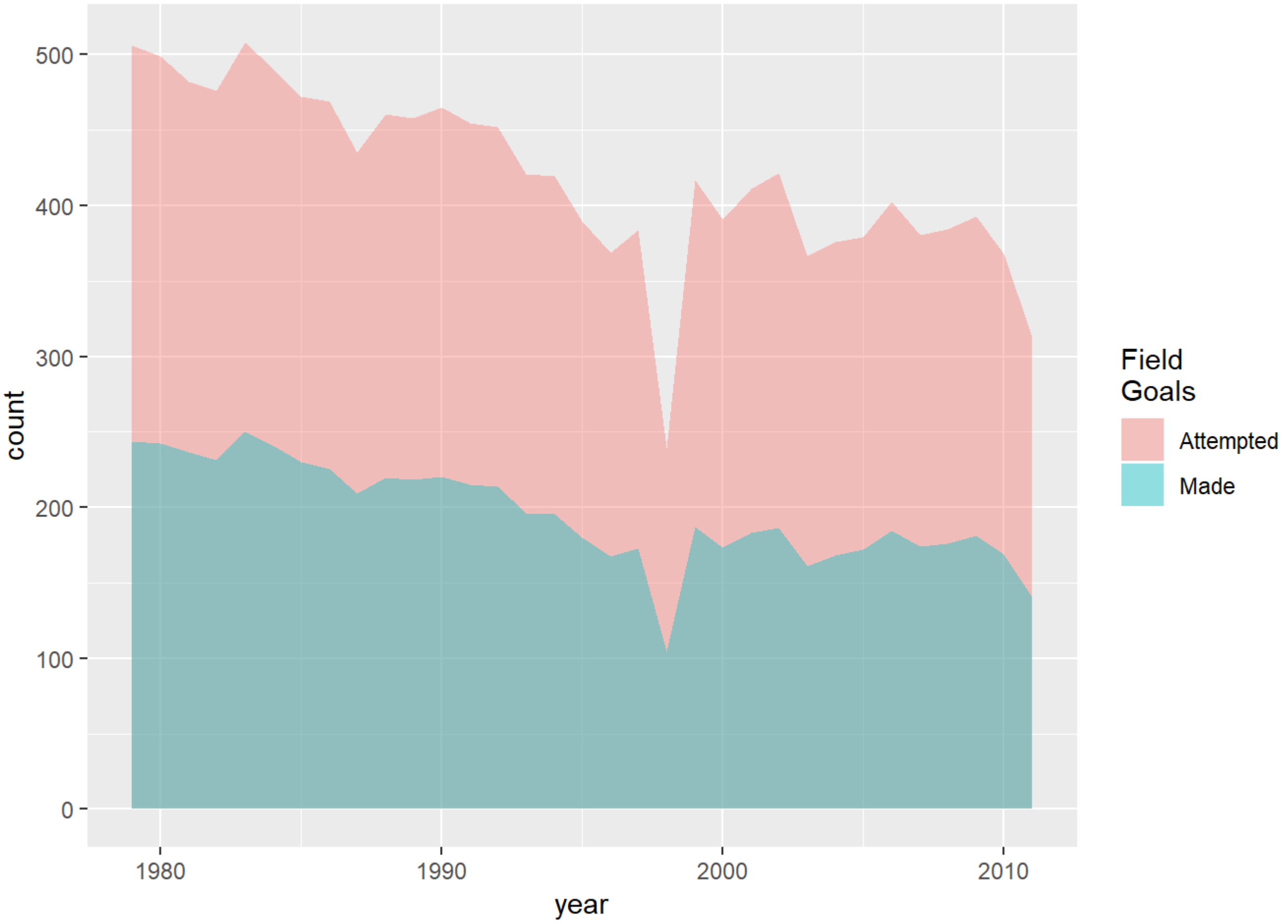
```r
# Calculate the means of each game statistic
playersmean <- players %>%
  select(c(playerID, year, minutes, points, oRebounds, dRebounds, rebounds, assists, steals, blo
cks, turnovers, PF, fgAttempted, fgMade, ftAttempted, ftMade, threeAttempted, threeMade)) %>%
  group_by(year) %>%
  summarize_if(is.numeric, mean)

# Create a plot for rebounds
playersmean %>%
  select(year, contains("rebound")) %>%
  gather(reboundType, count, -year) %>%
  ggplot(aes(x = year, y = count, fill = reboundType)) +
    geom_ribbon(aes(ymin = 0, ymax = count), alpha = 0.4) +
    scale_fill_discrete(name = "Rebounds",
                        labels = c("Defensive", "Offensive", "Total"))
```



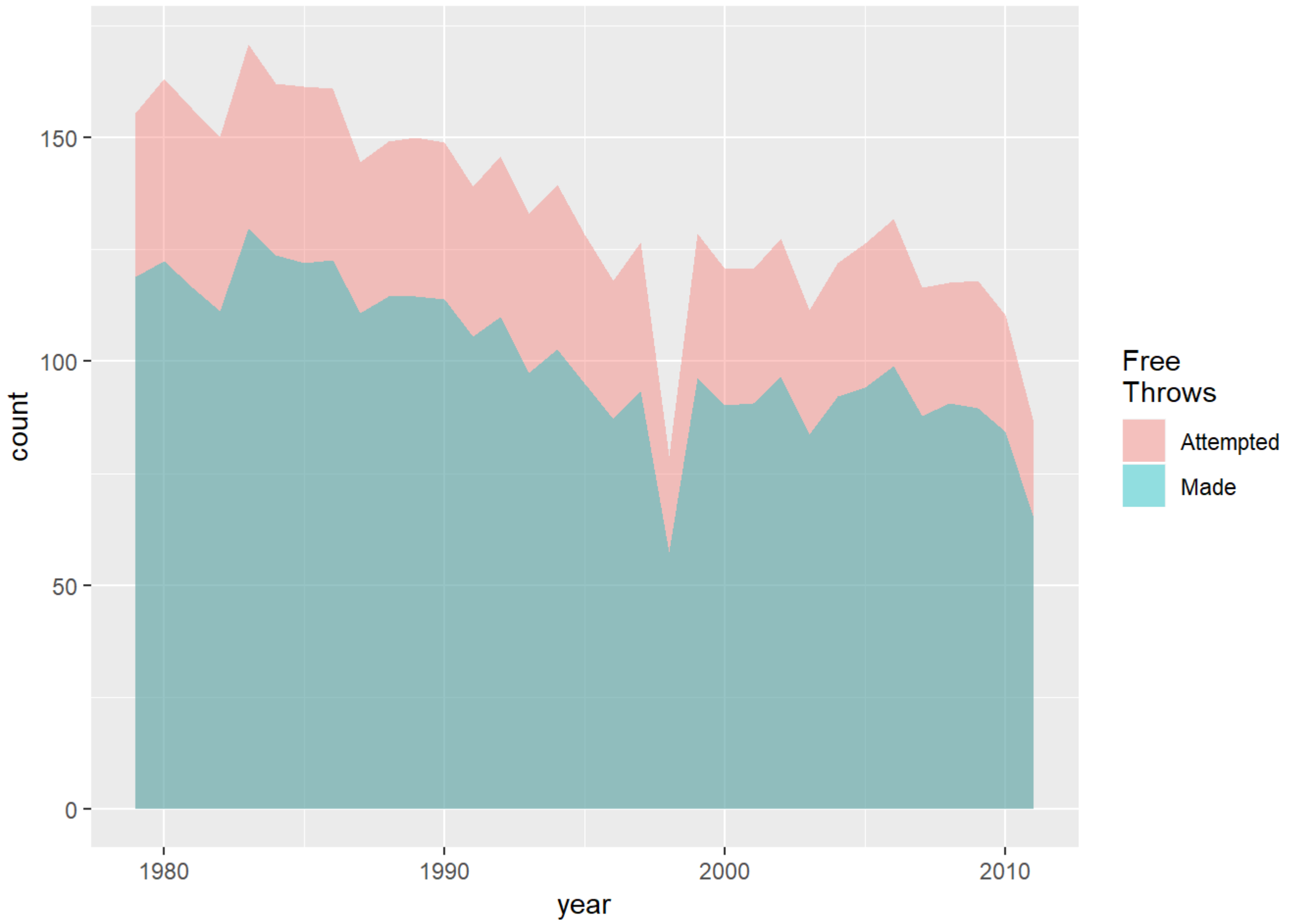
```r
# Create a plot for 3-pointers
playersmean %>%
  select(year, contains("three")) %>%
  gather(shot, count, -year) %>%
  ggplot(aes(x = year, y = count, fill = shot)) +
    geom_ribbon(aes(ymin = 0, ymax = count), alpha = 0.4) +
    scale_fill_discrete(name = "Three\nPointers",
                        labels = c("Attempted", "Made"))
```



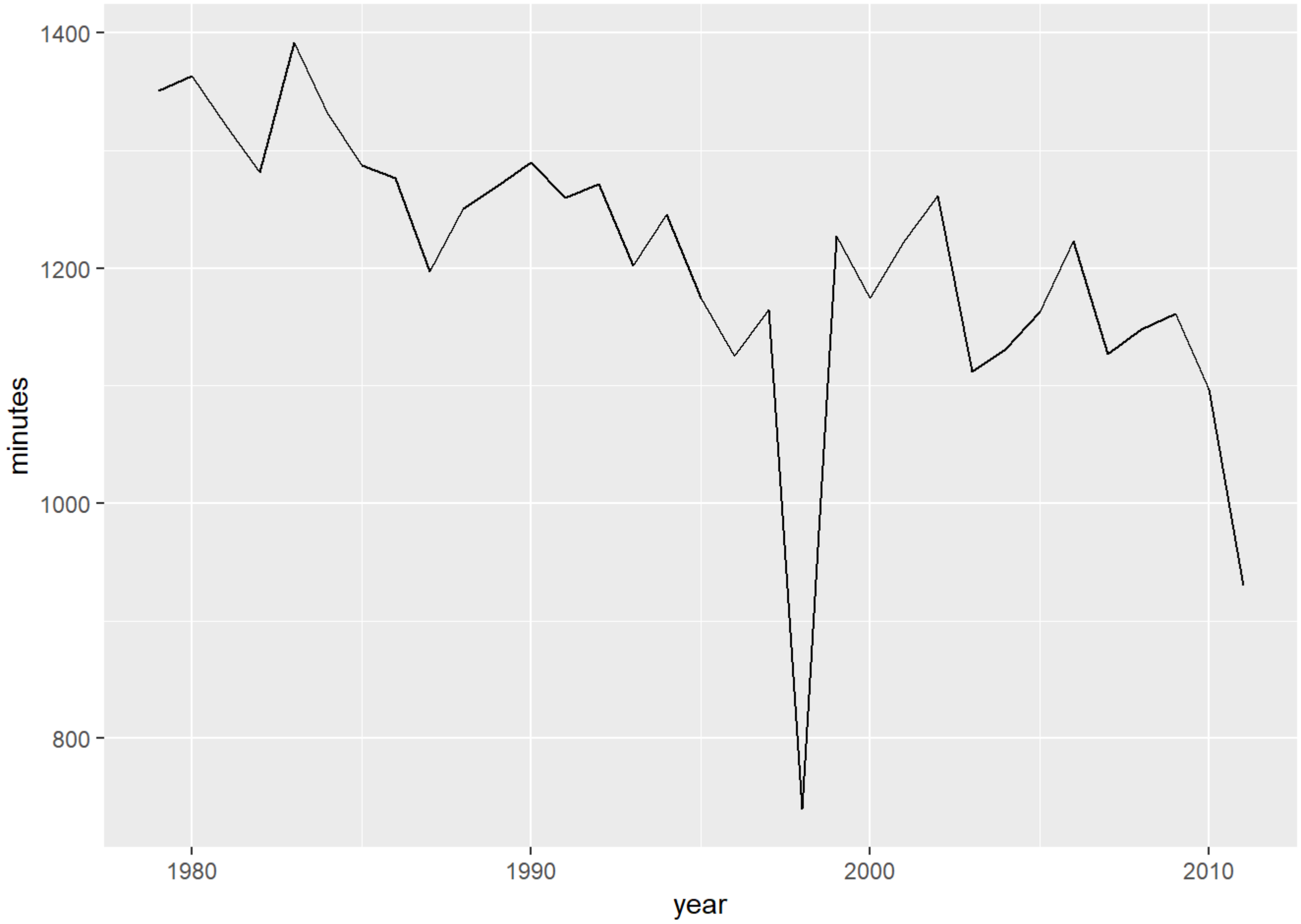
```
# Create a plot for field goals
playersmean %>%
  select(year, contains("fg")) %>%
  gather(fieldGoal, count, -year) %>%
  ggplot(aes(x = year, y = count, fill = fieldGoal)) +
    geom_ribbon(aes(ymin = 0, ymax = count), alpha = 0.4) +
    scale_fill_discrete(name = "Field\nGoals",
                        labels = c("Attempted", "Made"))
```

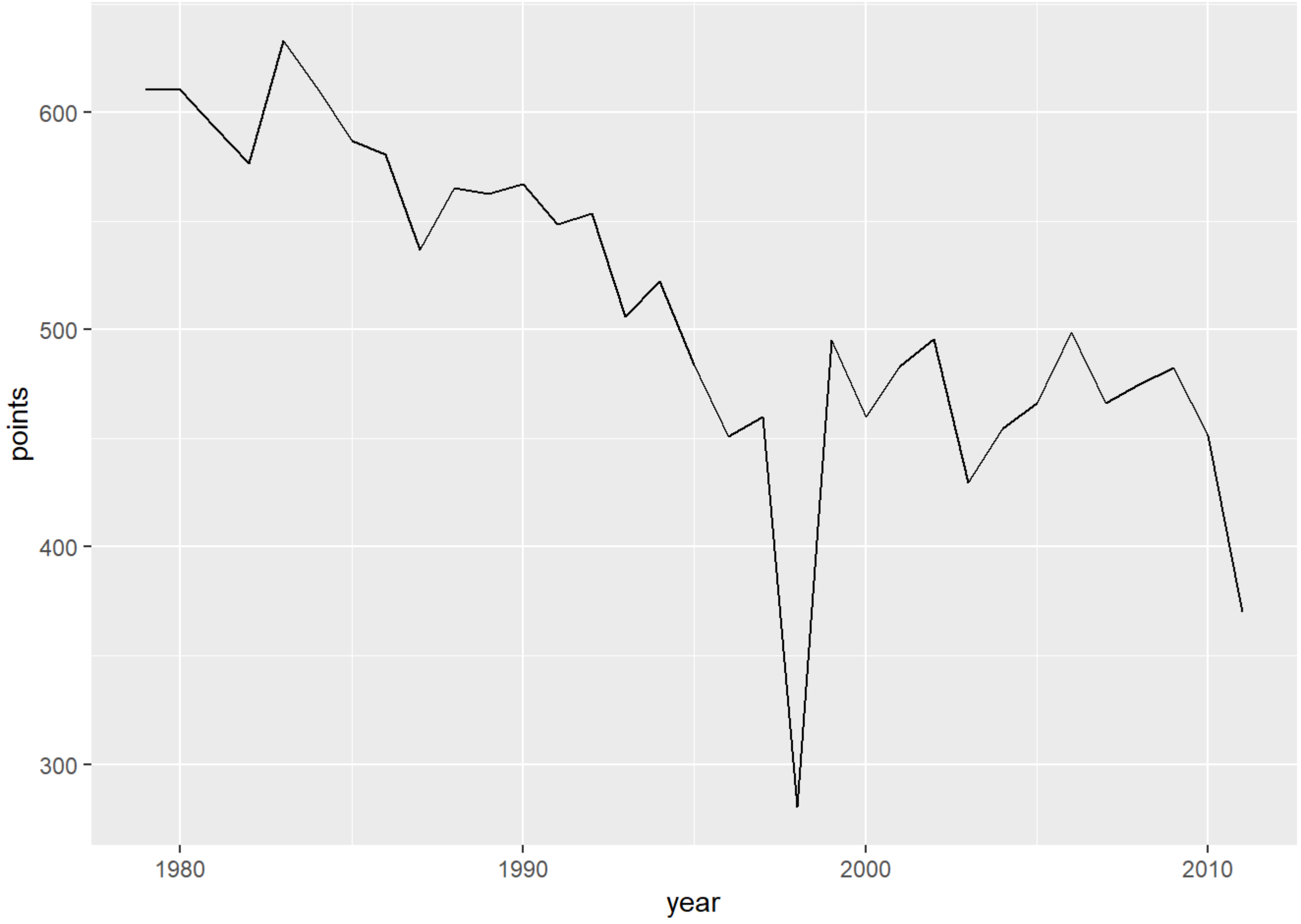


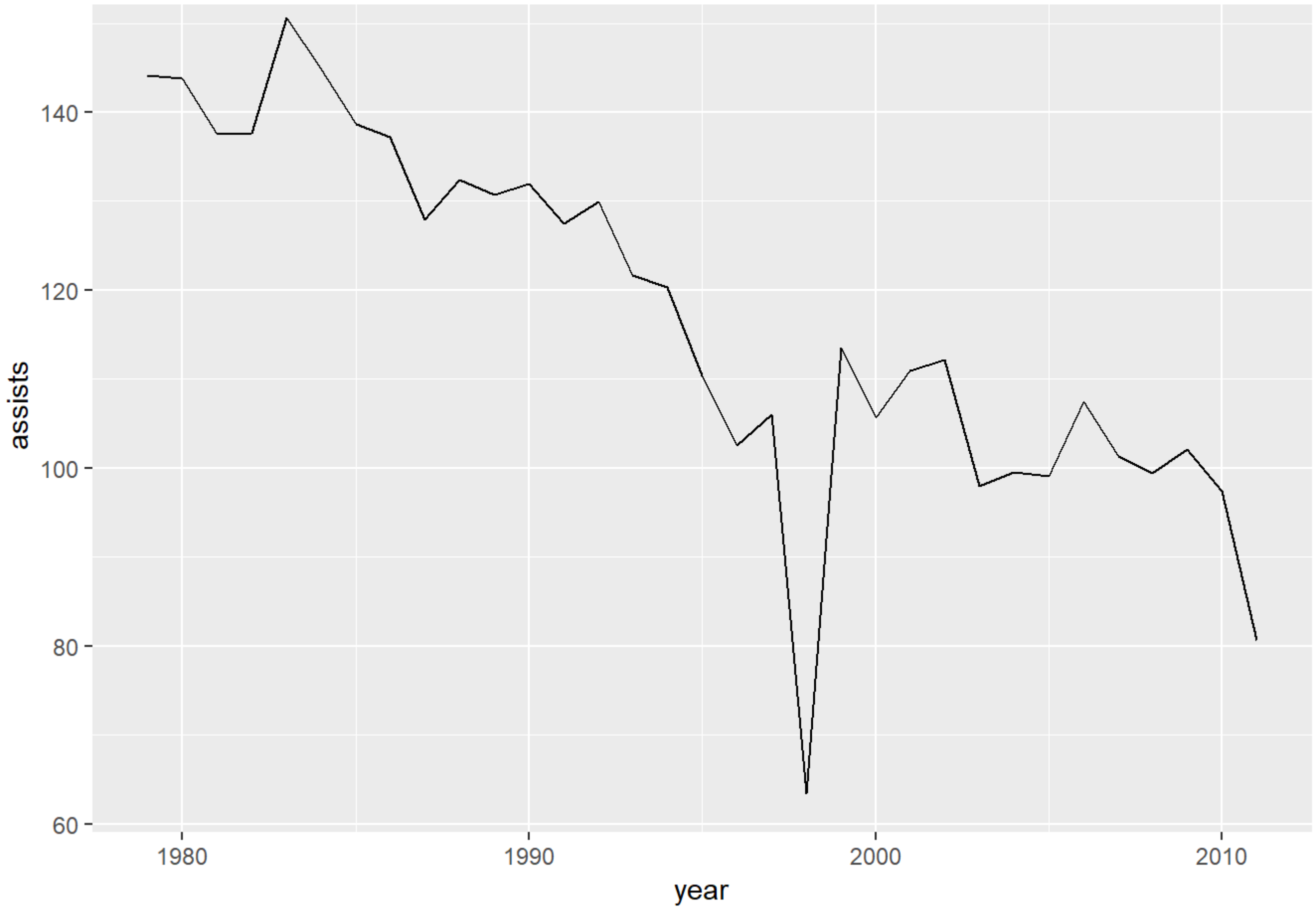
```
# Create a plot for free throws
playersmean %>%
  select(year, contains("ft")) %>%
  gather(freeThrow, count, -year) %>%
  ggplot(aes(x = year, y = count, fill = freeThrow)) +
    geom_ribbon(aes(ymin = 0, ymax = count), alpha = 0.4) +
    scale_fill_discrete(name = "Free\nThrows",
                        labels = c("Attempted", "Made"))
```

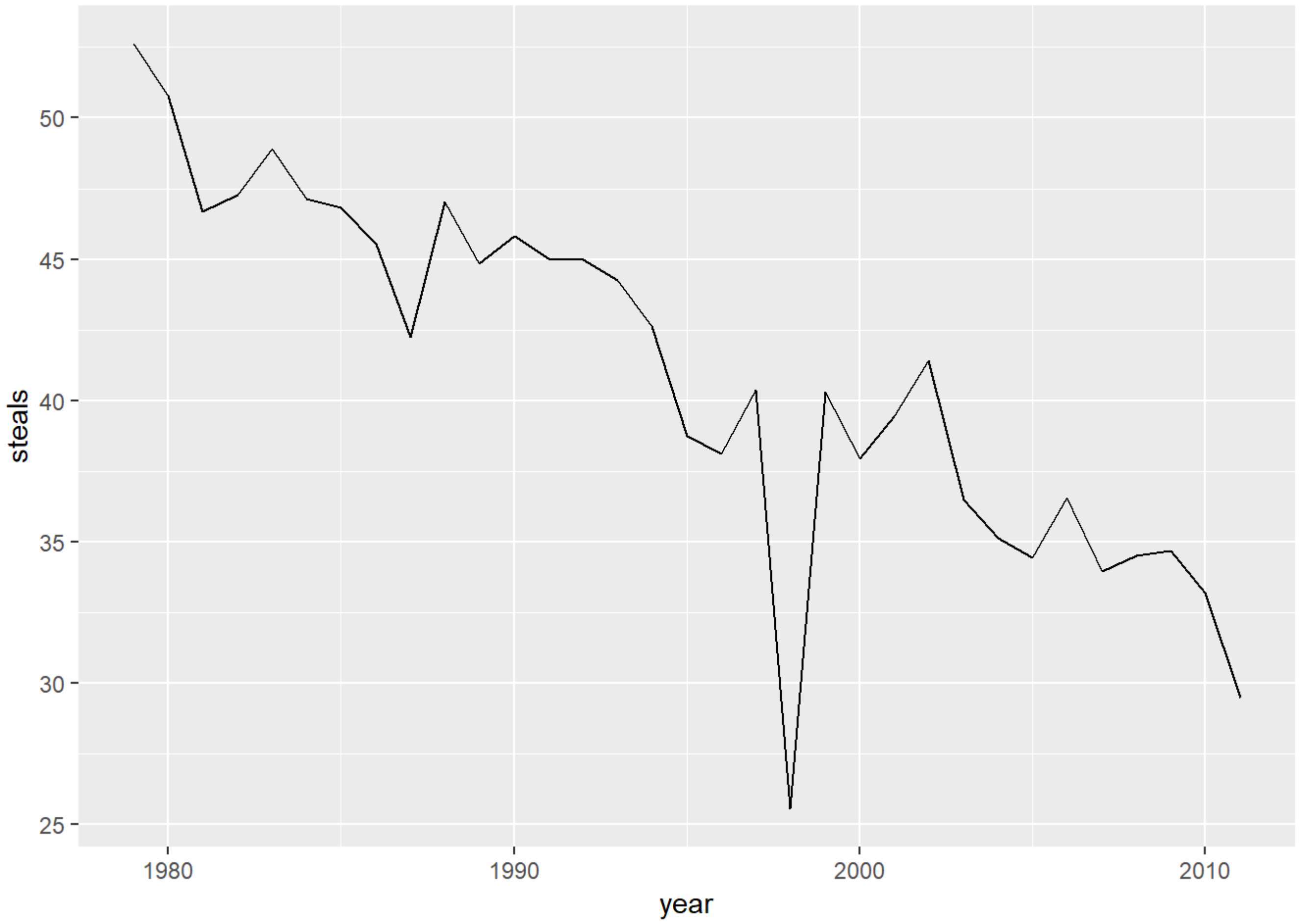


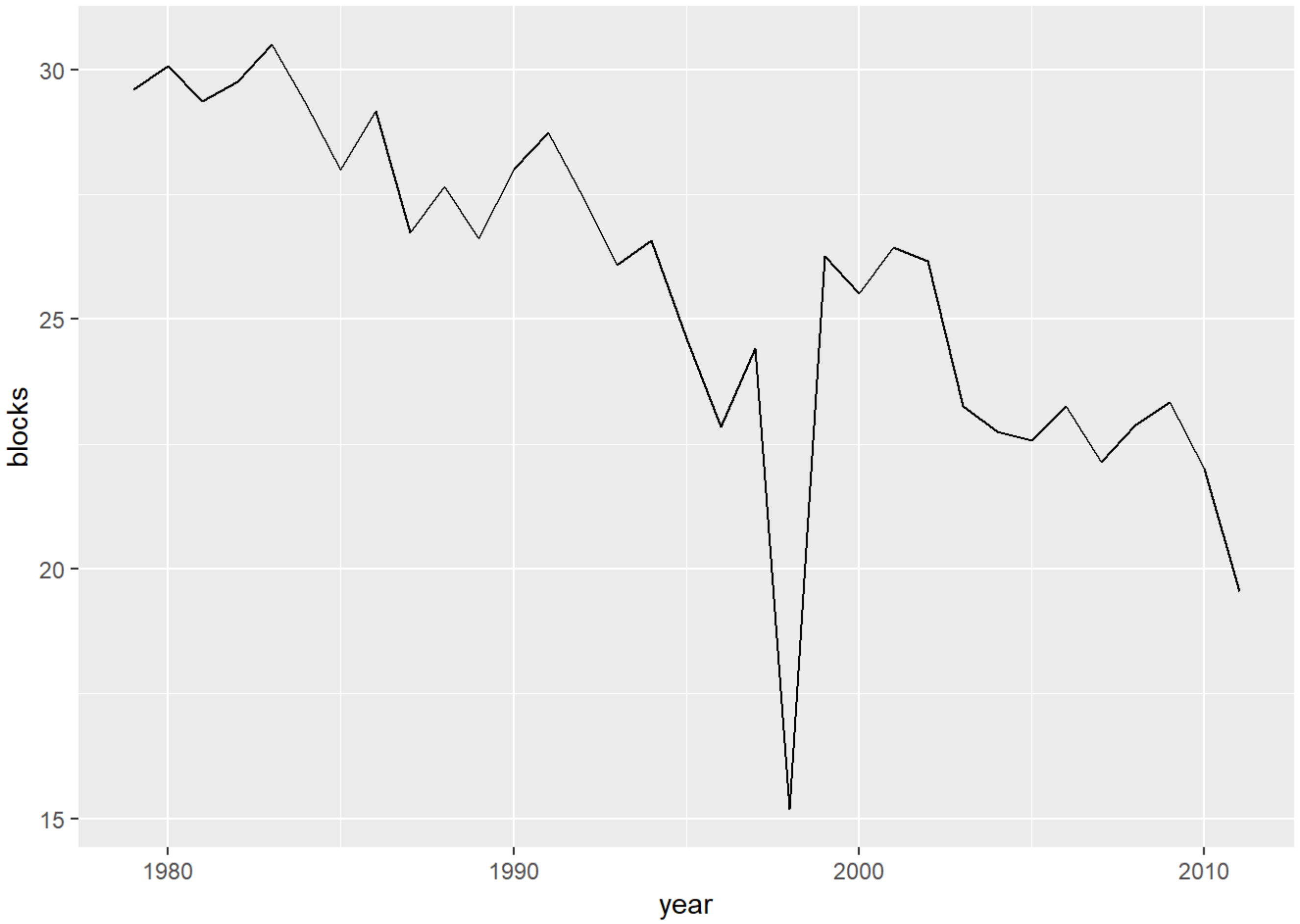
```
# Create line plots for everything else
for(i in names(playersmean)){
  if (!grepl("year|rebound|three|fg|ft", tolower(i))){
    plt <- playersmean %>%
      ggplot(aes_string(x = "year", y = i)) + geom_line()
    print(plt)
  }
}
```


1400
1200
1000
800
minutes
1980
1990
2000
2010
year

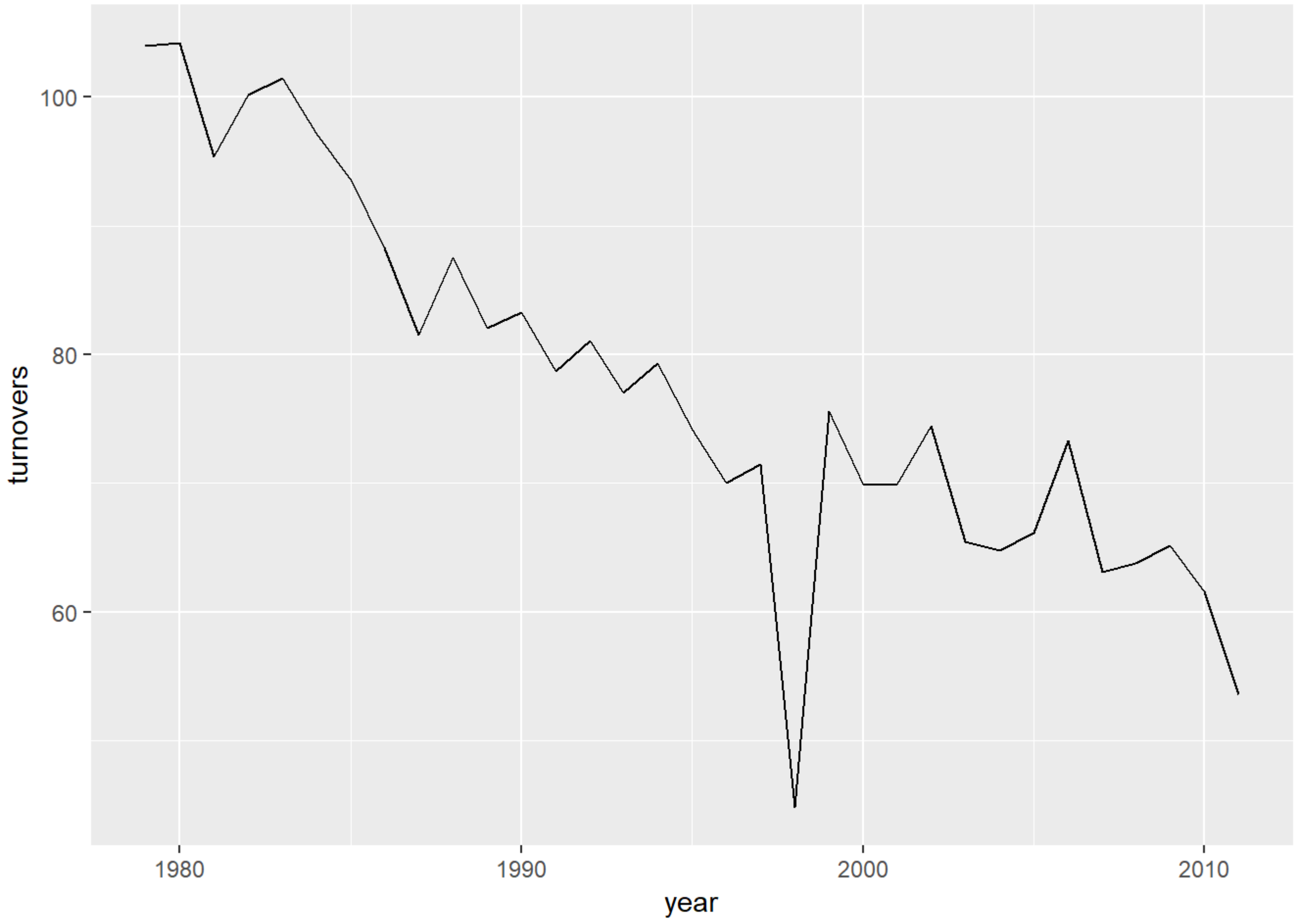

600
500
400
300
points
1980
1990
2000
2010
year


assists
140
120
100
80
60
1980
1990
2000
2010
year

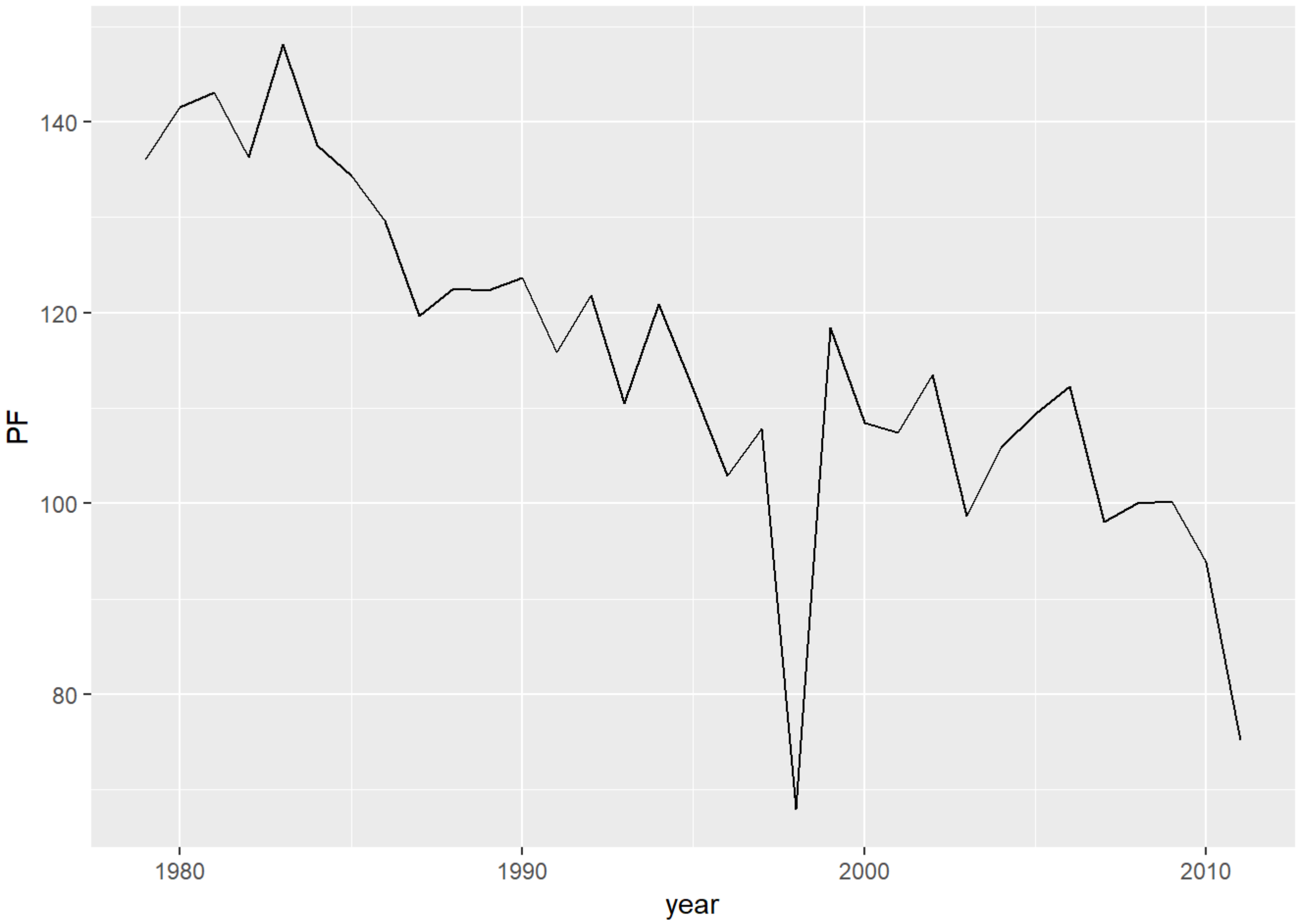

steals
50
45
40
35
30
25
1980
1990
2000
2010
year



30
25
20
15
blocks
1980
1990
2000
2010
year



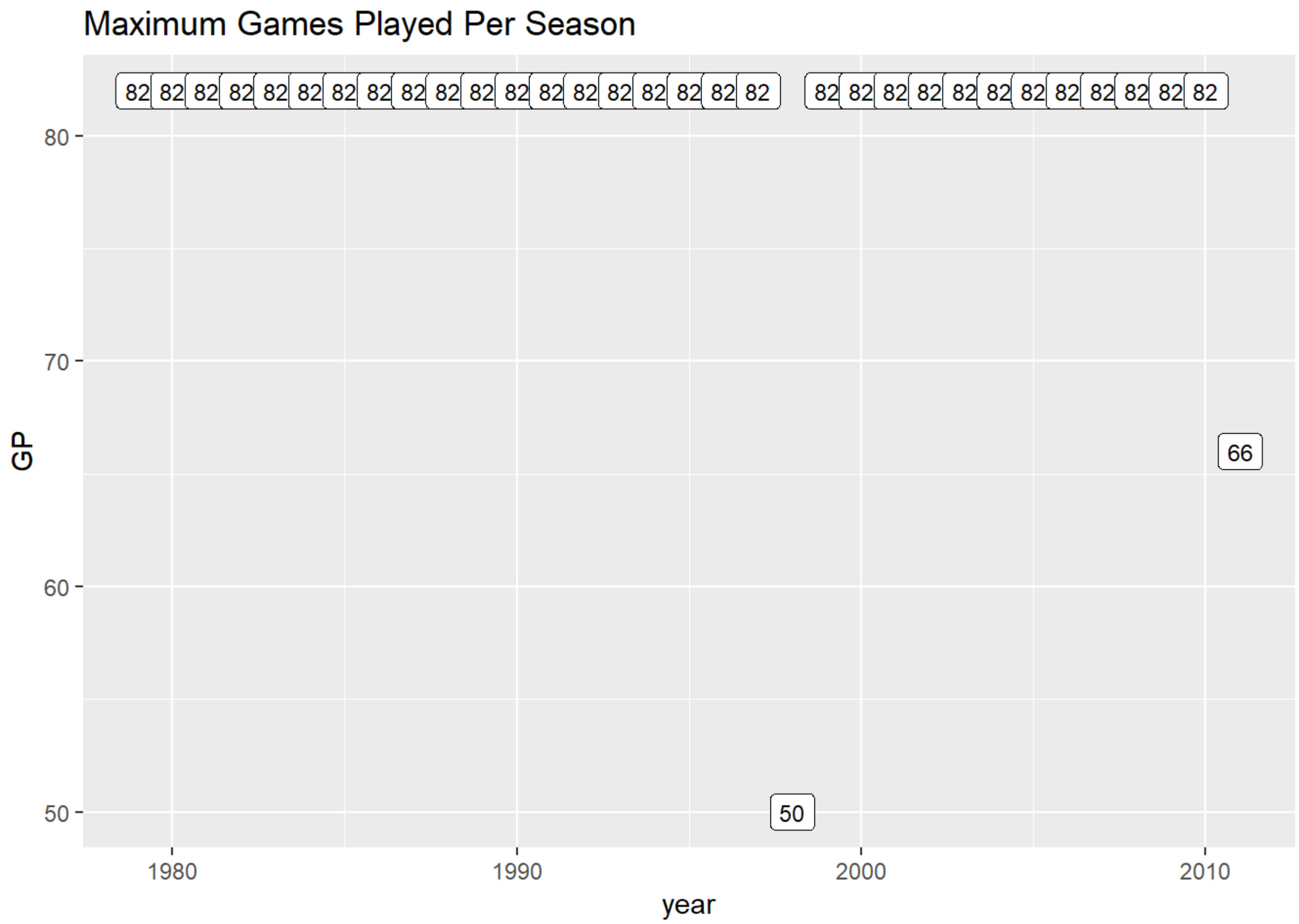
100
80
60
turnovers
1980
1990
2000
2010
year



Looking at the plots above, we notice a few things:

1. 3-point shooting became an increasingly important part of the game over the years.
2. Despite the 3-point shooting, the points scored per game decreased overall.
3. Most importantly, **there's a noticeable dive in every statistic in 1998 and 2012**.

We can see why this is by plotting the maximum games played per season:



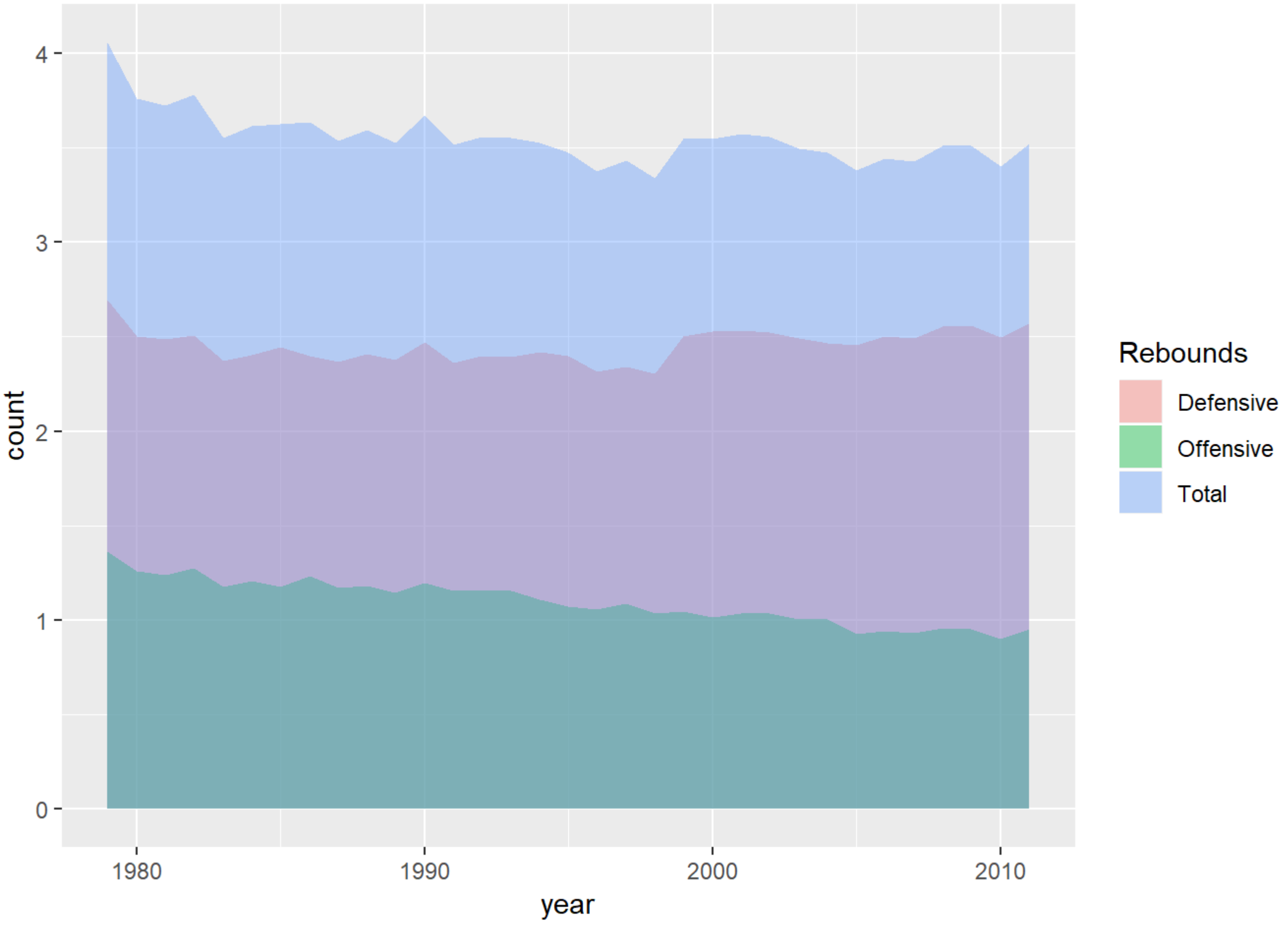
In the history of the NBA, there have been a total of four lockouts (https://en.wikipedia.org/wiki/NBA_lockout). On two of these occassions (1995 and 1996), players and owners were able to reach an agreement before the start of the regular season. However, the two most recent lockouts actually extended into what would have been the beginning of the regular seasons, forcing shortened seasons of 50 games per team in 1998-1999 (https://en.wikipedia.org/wiki/1998%E2%80%9399_NBA_lockout) and 66 games per team in 2011-2012 (https://en.wikipedia.org/wiki/2011_NBA_lockout). The 1998-1999 lockout even resulted in the cancellation of the season’s All-Star game.

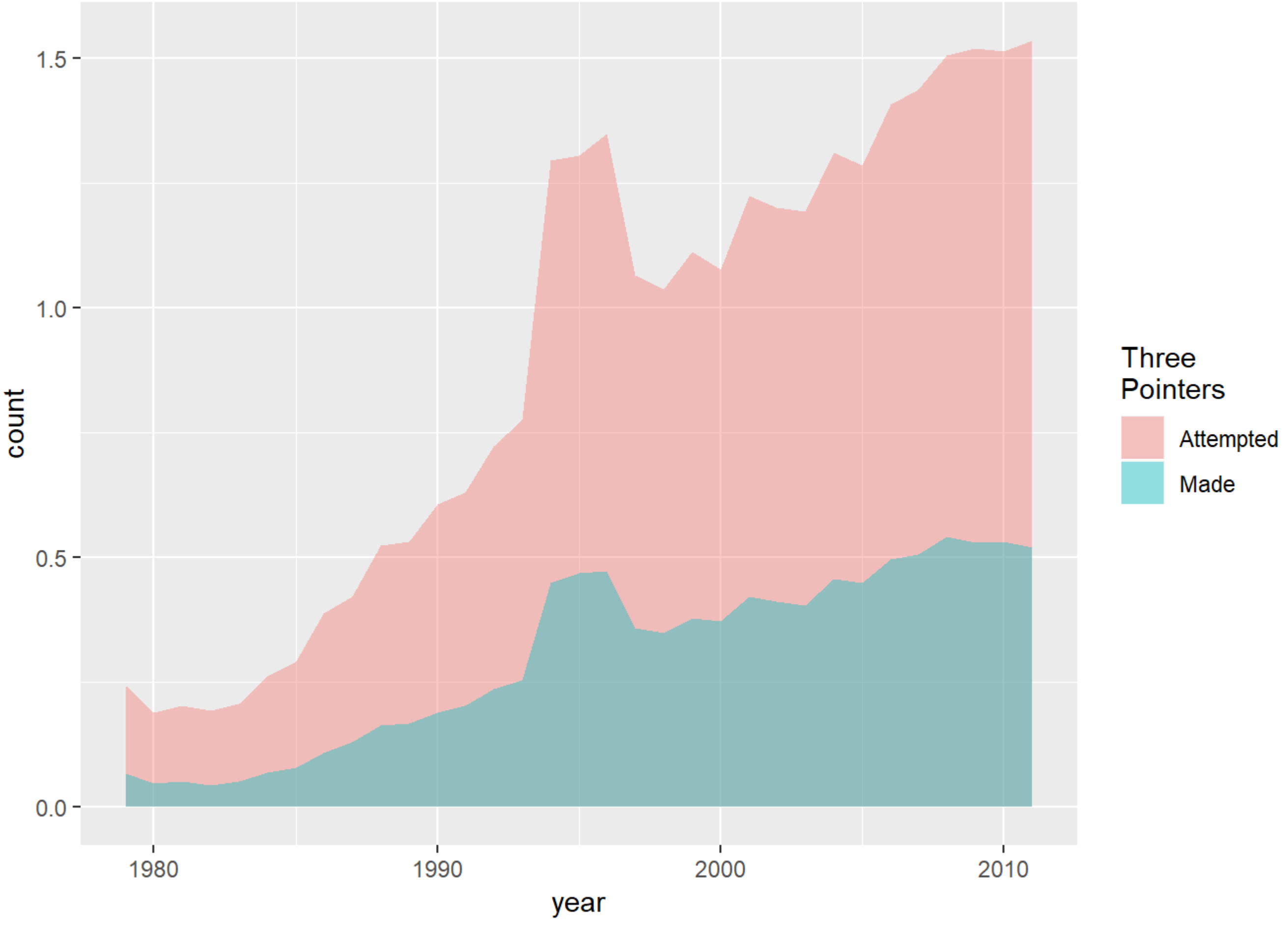
To avoid the dives in our plots due to the shortened seasons, we can add some features to the *players* data that consider the tracked stats on a per-game basis.

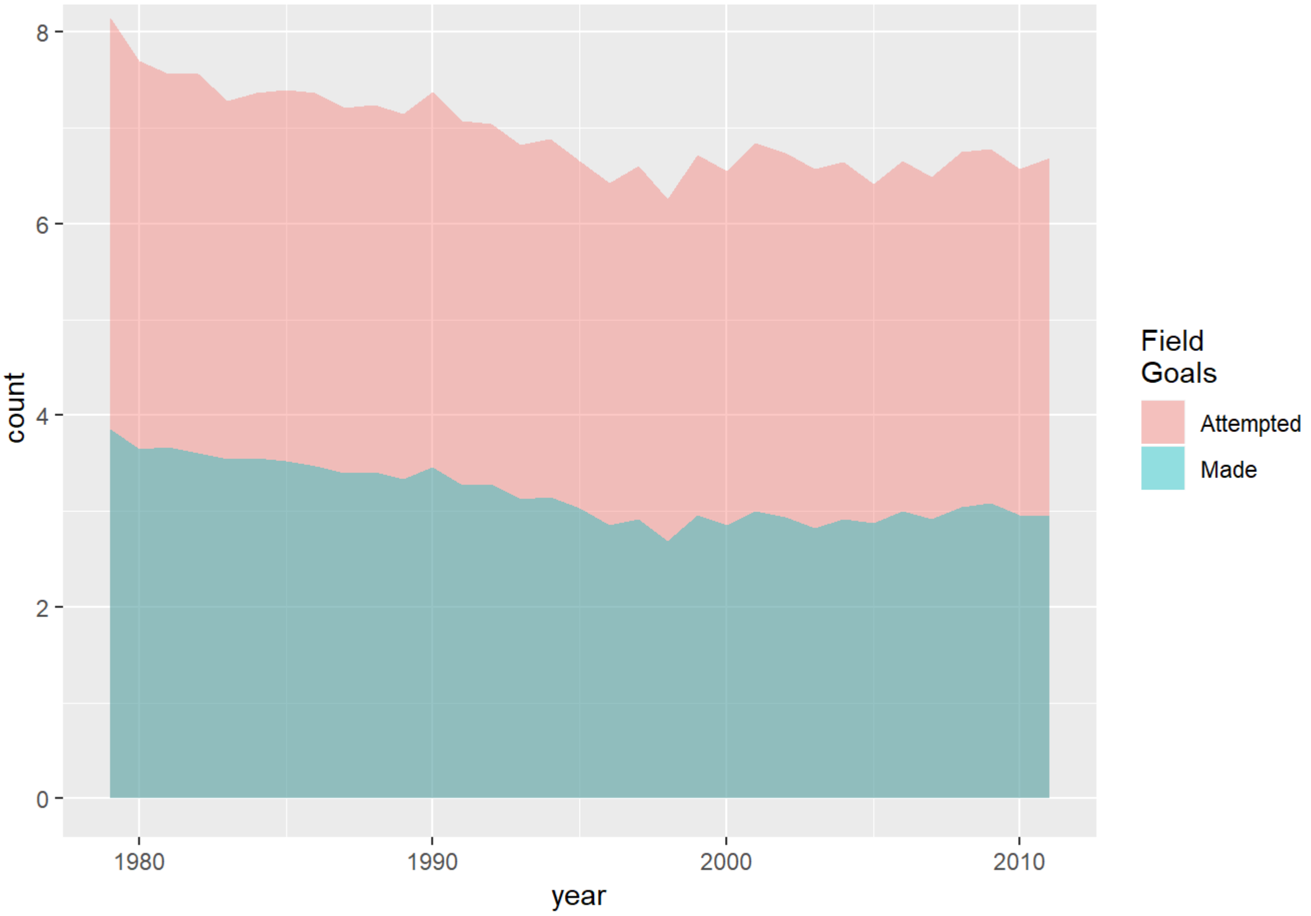
Feature Engineering

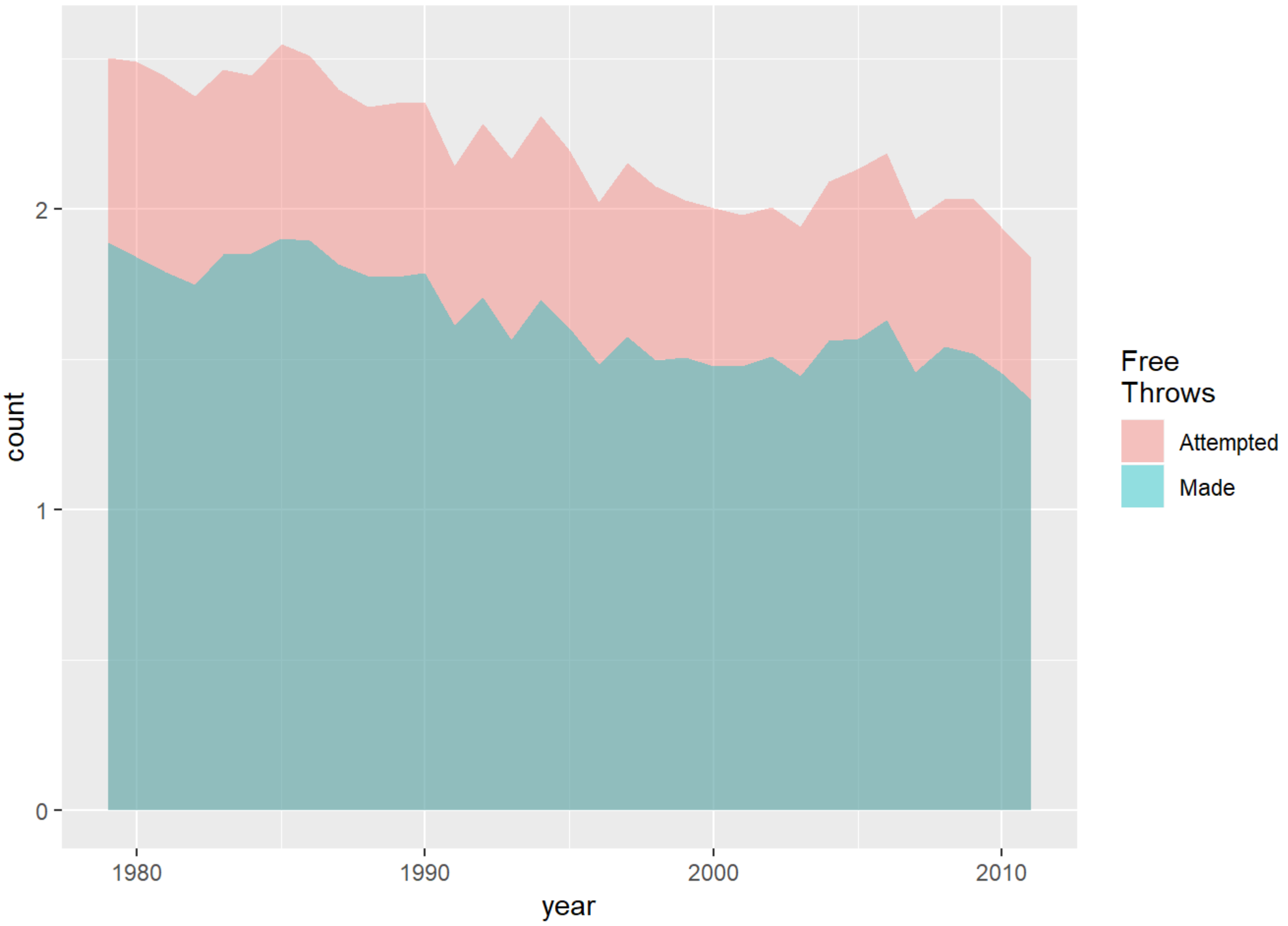
Rather than omit the seasons with fewer games, we can add the per-game stats and re-run the code from above to generate the same plots without the dives.

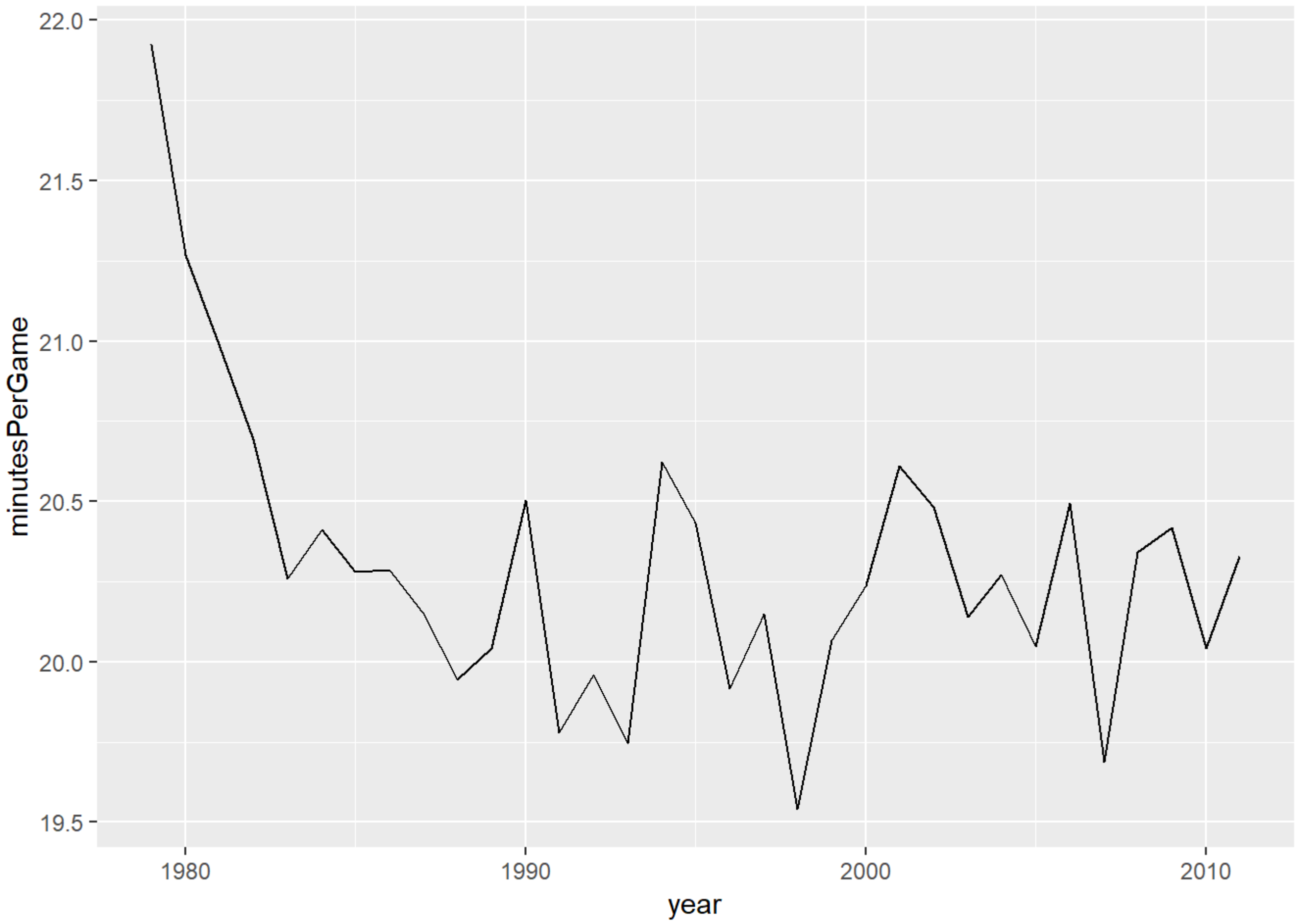
```
# Create variables for per-game statistics
players <- players %>%
  mutate(
    minutesPerGame = minutes / GP,
    pointsPerGame = points / GP,
    assistsPerGame = assists / GP,
    oReboundsPerGame = oRebounds / GP,
    dReboundsPerGame = dRebounds / GP,
    reboundsPerGame = rebounds / GP,
    stealsPerGame = steals / GP,
    blocksPerGame = blocks / GP,
    turnoversPerGame = turnovers / GP,
    fgAttemptedPerGame = fgAttempted / GP,
    fgMadePerGame = fgMade/ GP,
    ftAttemptedPerGame = ftAttempted / GP,
    ftMadePerGame = ftMade / GP,
    threeAttemptedPerGame = threeAttempted / GP,
    threeMadePerGame = threeMade / GP)
```

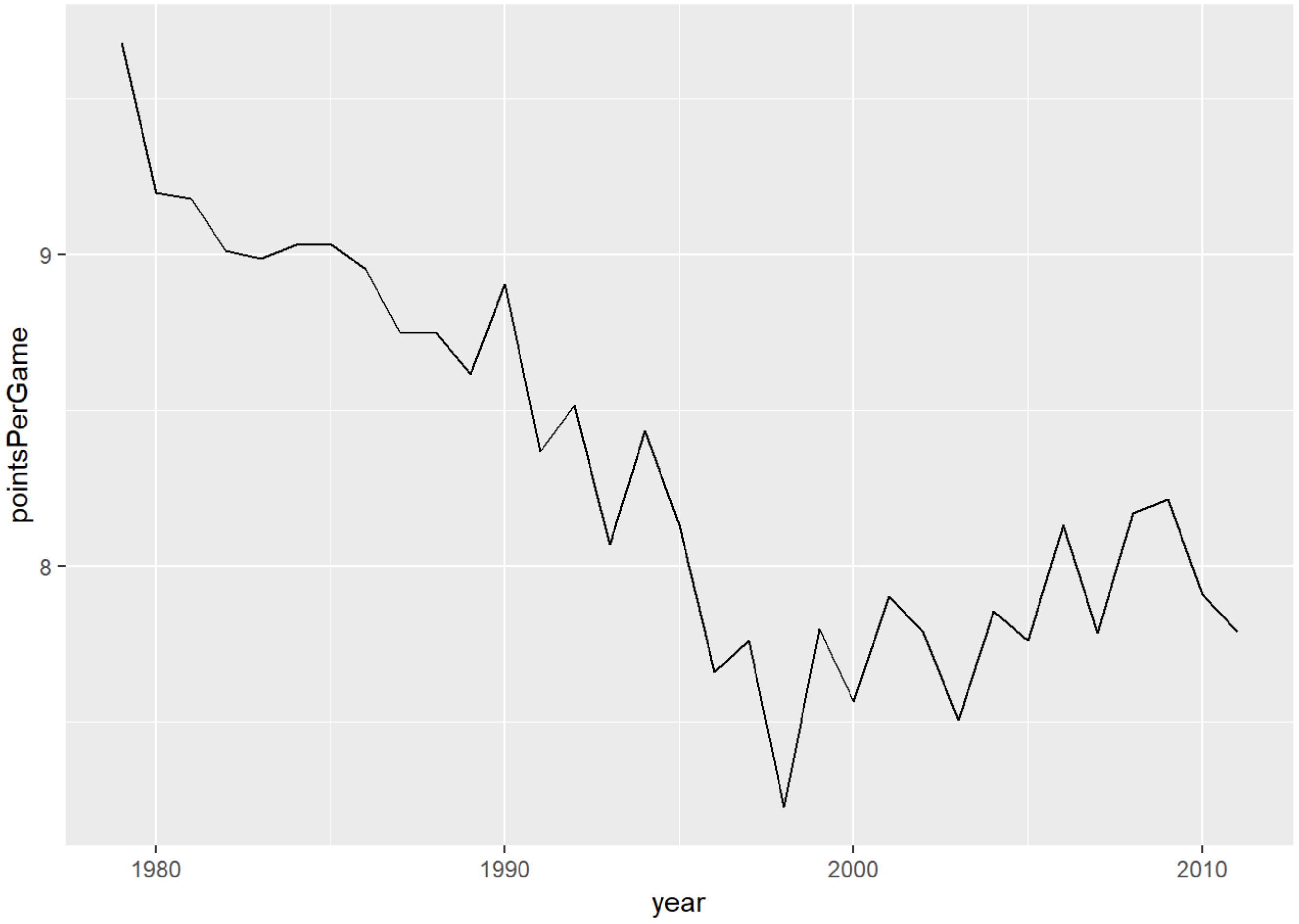

count
4
3
2
1
0
1980
1990
2000
2010
year
Rebounds
Defensive
Offensive
Total

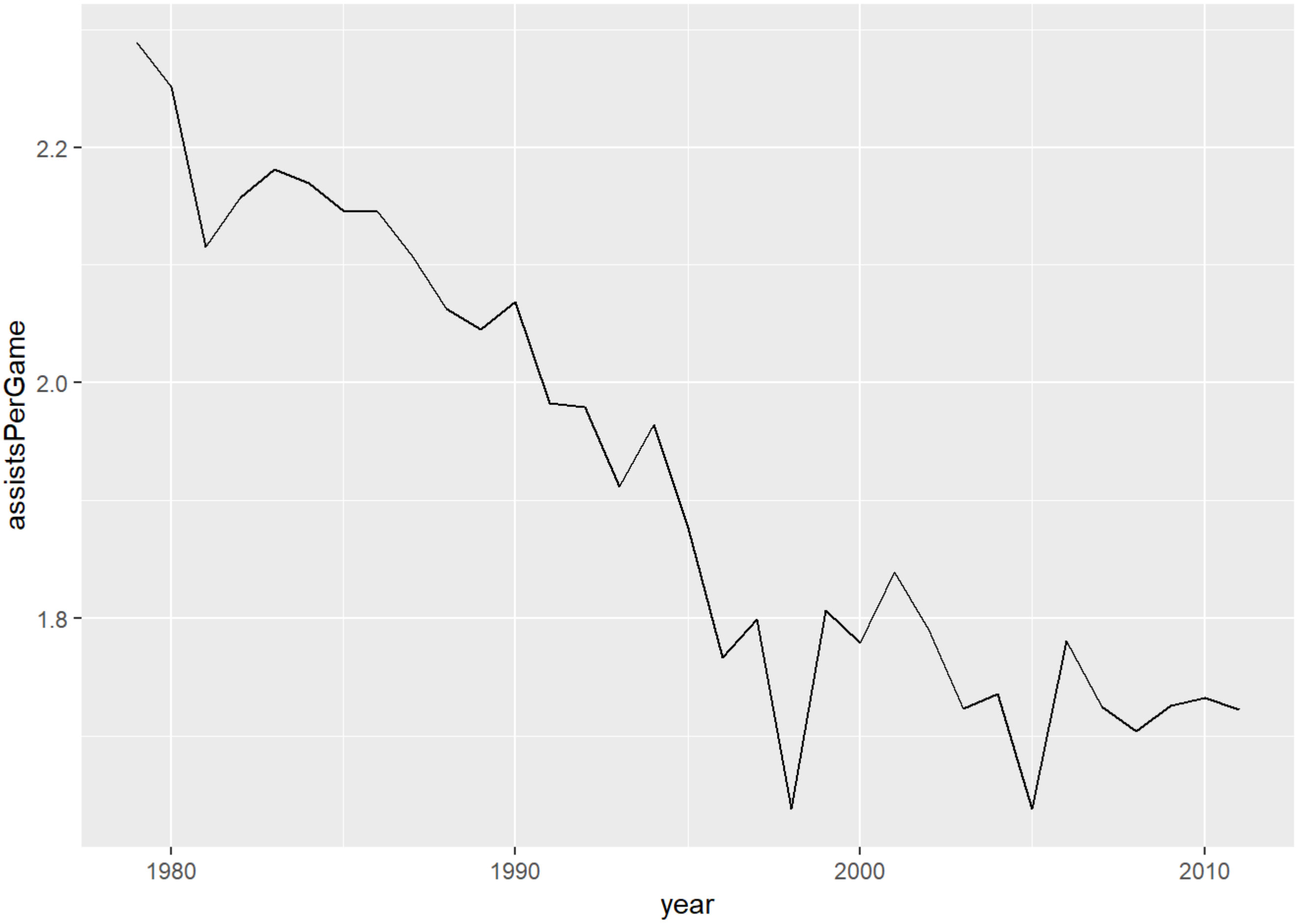

count
1.5
1.0
0.5
0.0
1980
1990
2000
2010
year
Three
Pointers
Attempted
Made

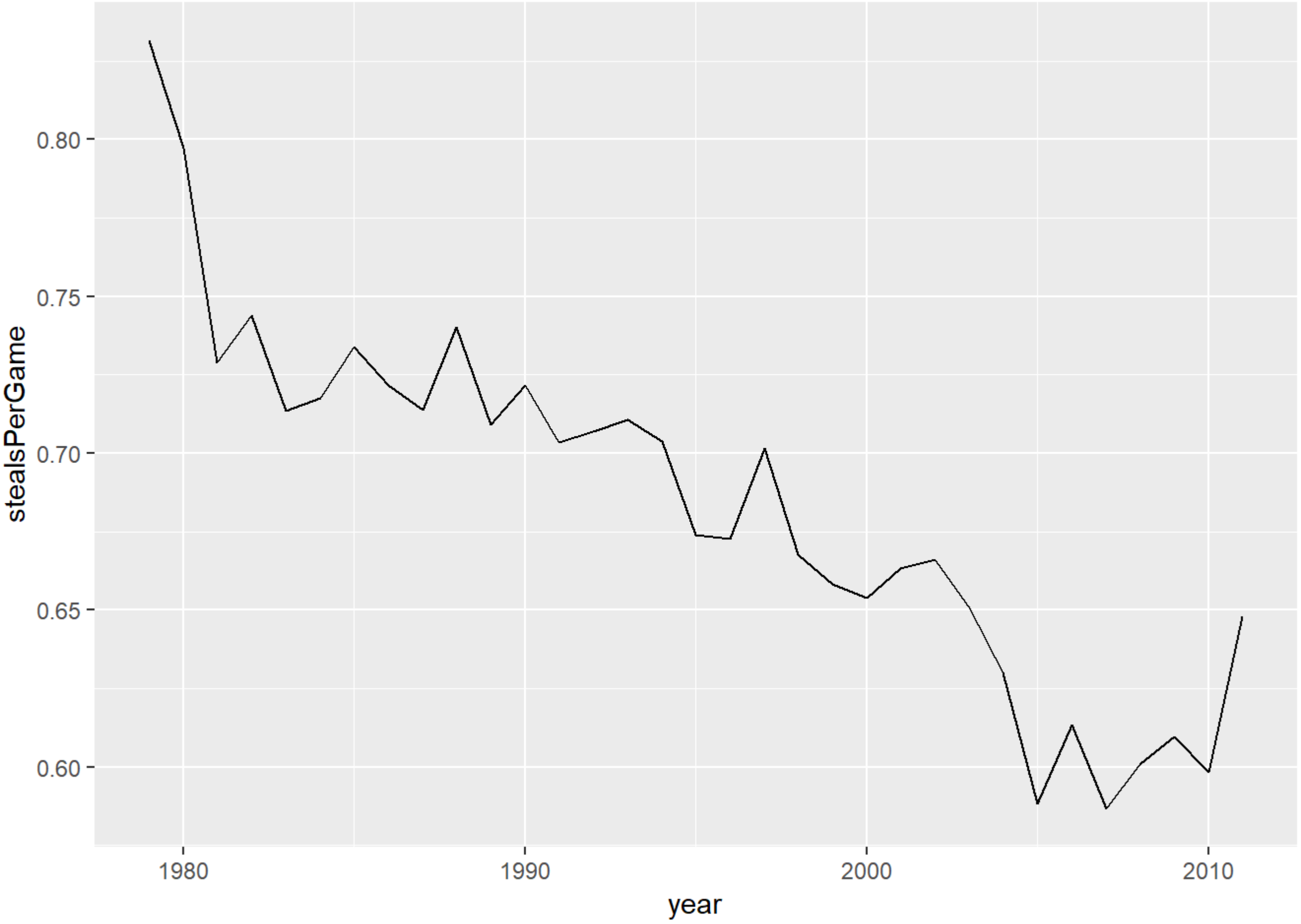

count
8
6
4
2
0
1980
1990
2000
2010
year
Field Goals
Attempted
Made

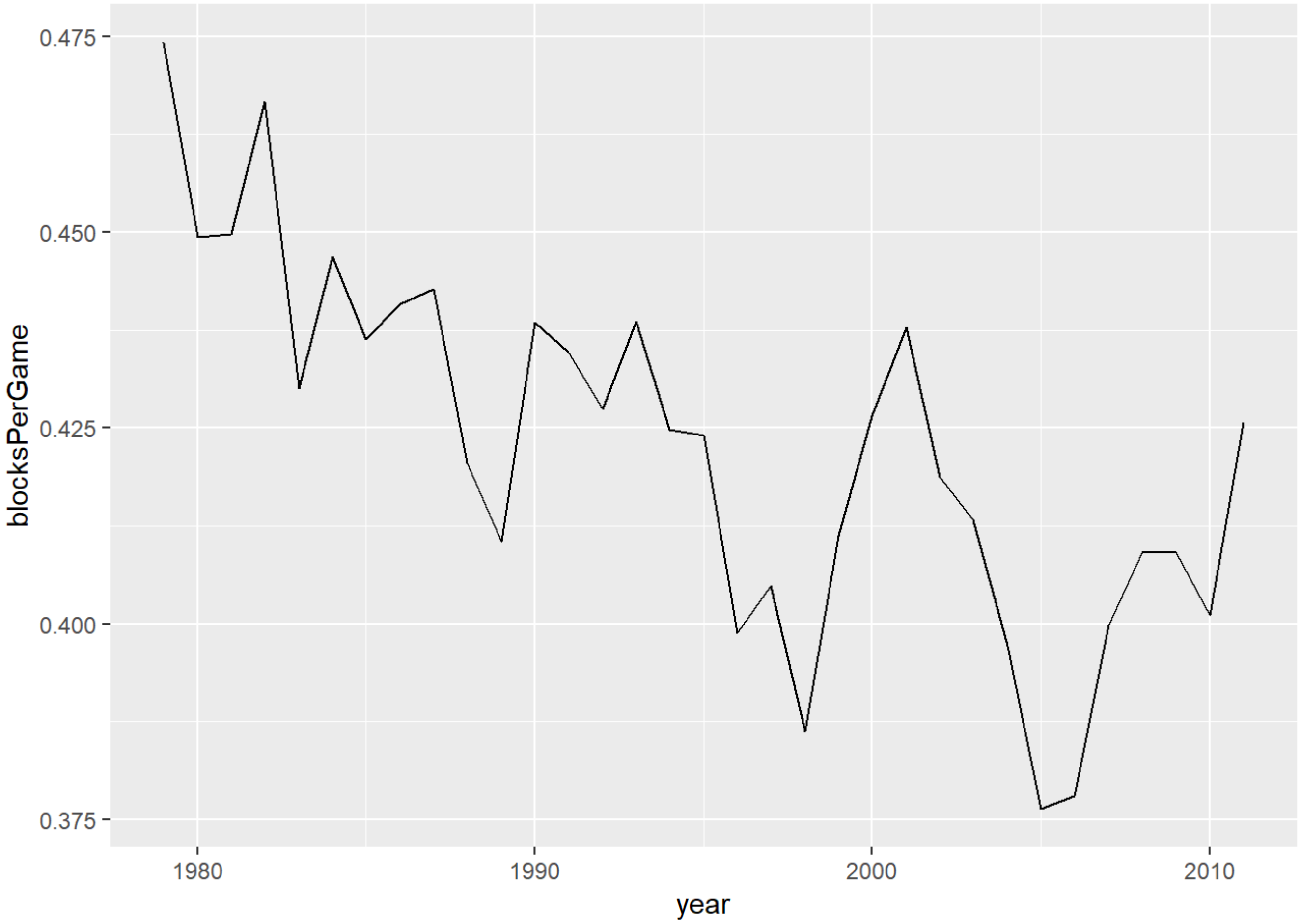

count
2
1
0
1980
1990
2000
2010
year
Free Throws
Attempted
Made

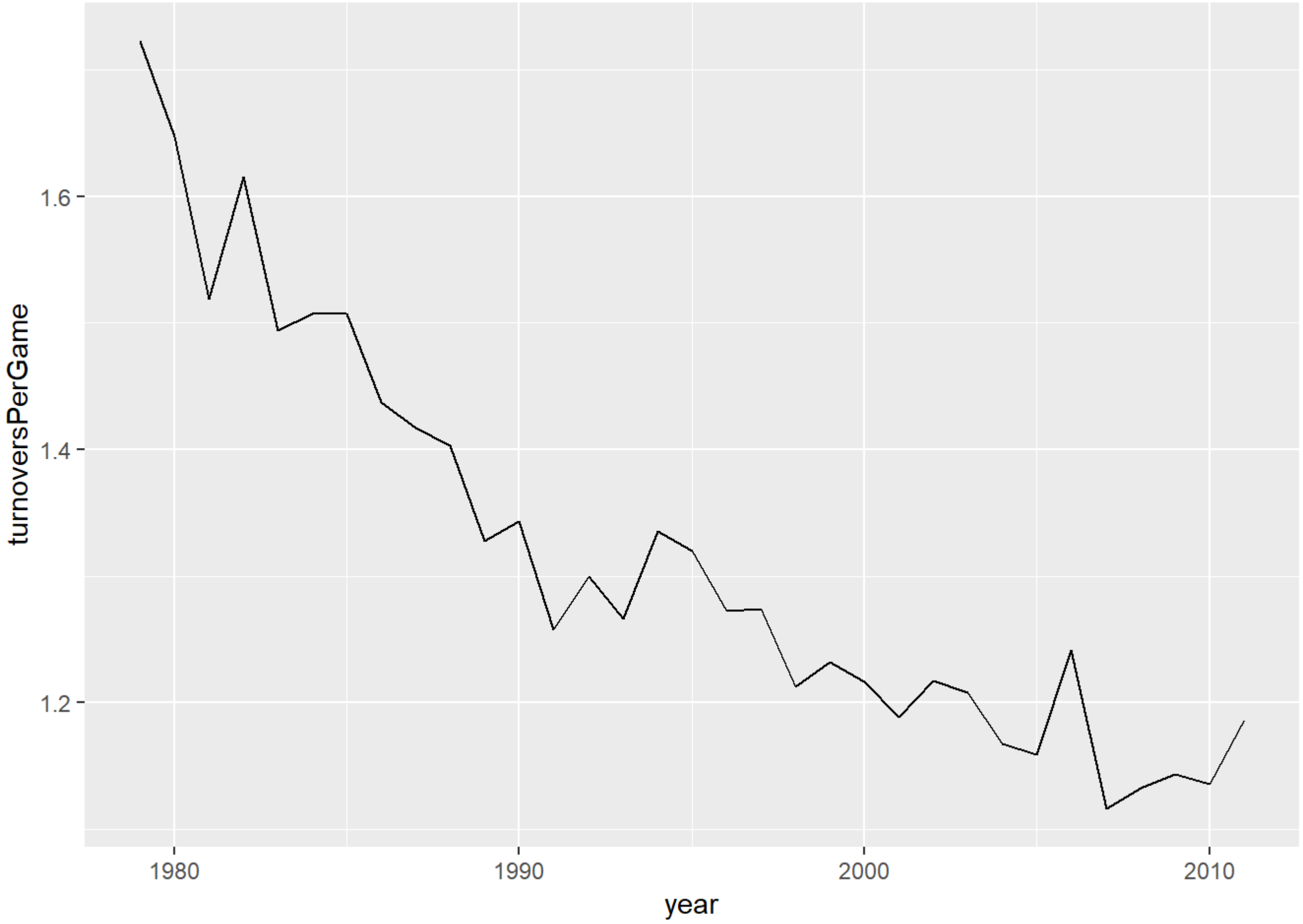

minutesPerGame
22.0
21.5
21.0
20.5
20.0
19.5
1980
1990
2000
2010
year


pointsPerGame
9
8
1980
1990
2000
2010
year


assistsPerGame
2.2
2.0
1.8
1980
1990
2000
2010
year


stealsPerGame
0.80
0.75
0.70
0.65
0.60
1980
1990
2000
2010
year

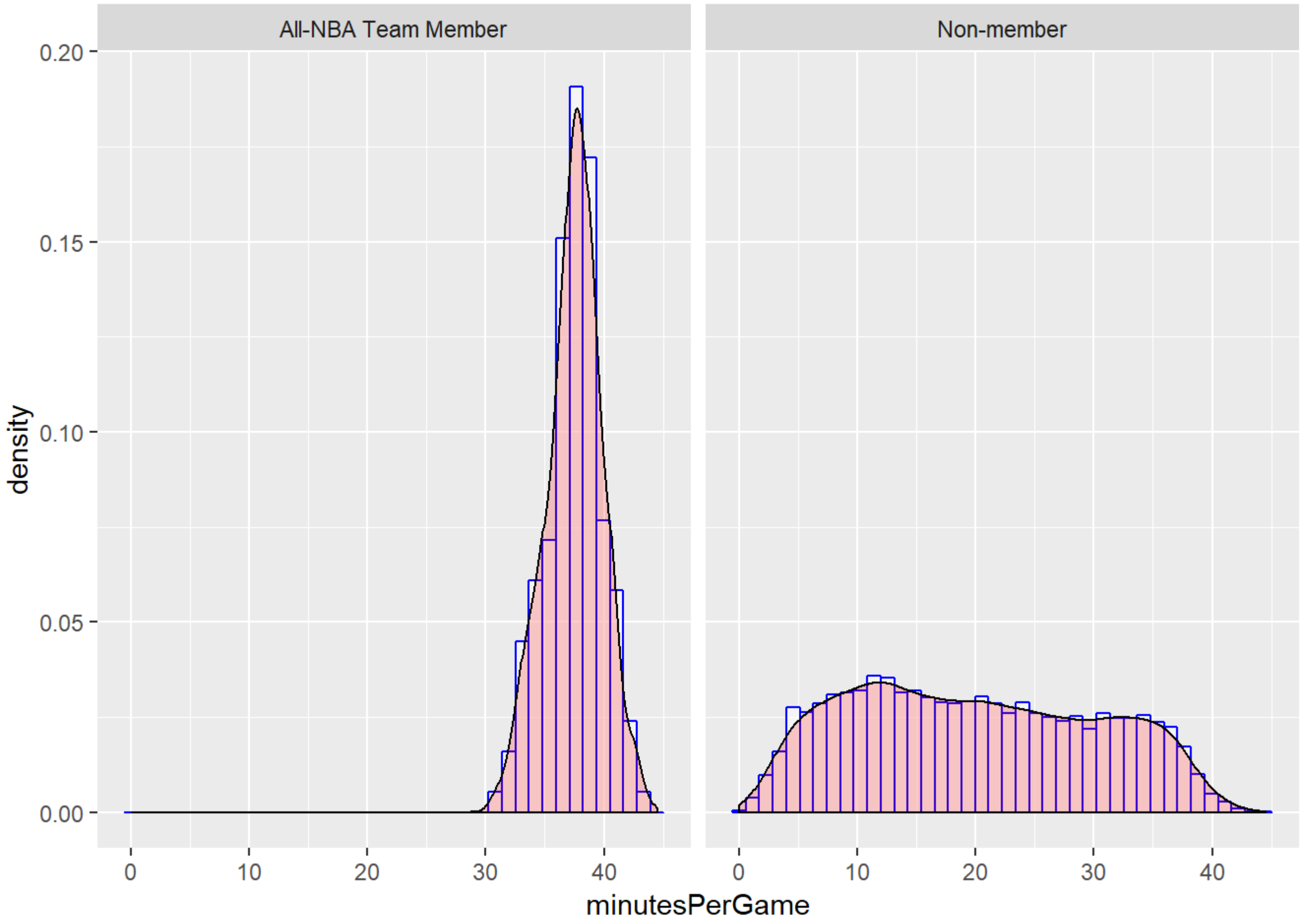

0.475
0.450
0.425
0.400
0.375
blocksPerGame
1980
1990
2000
2010
year

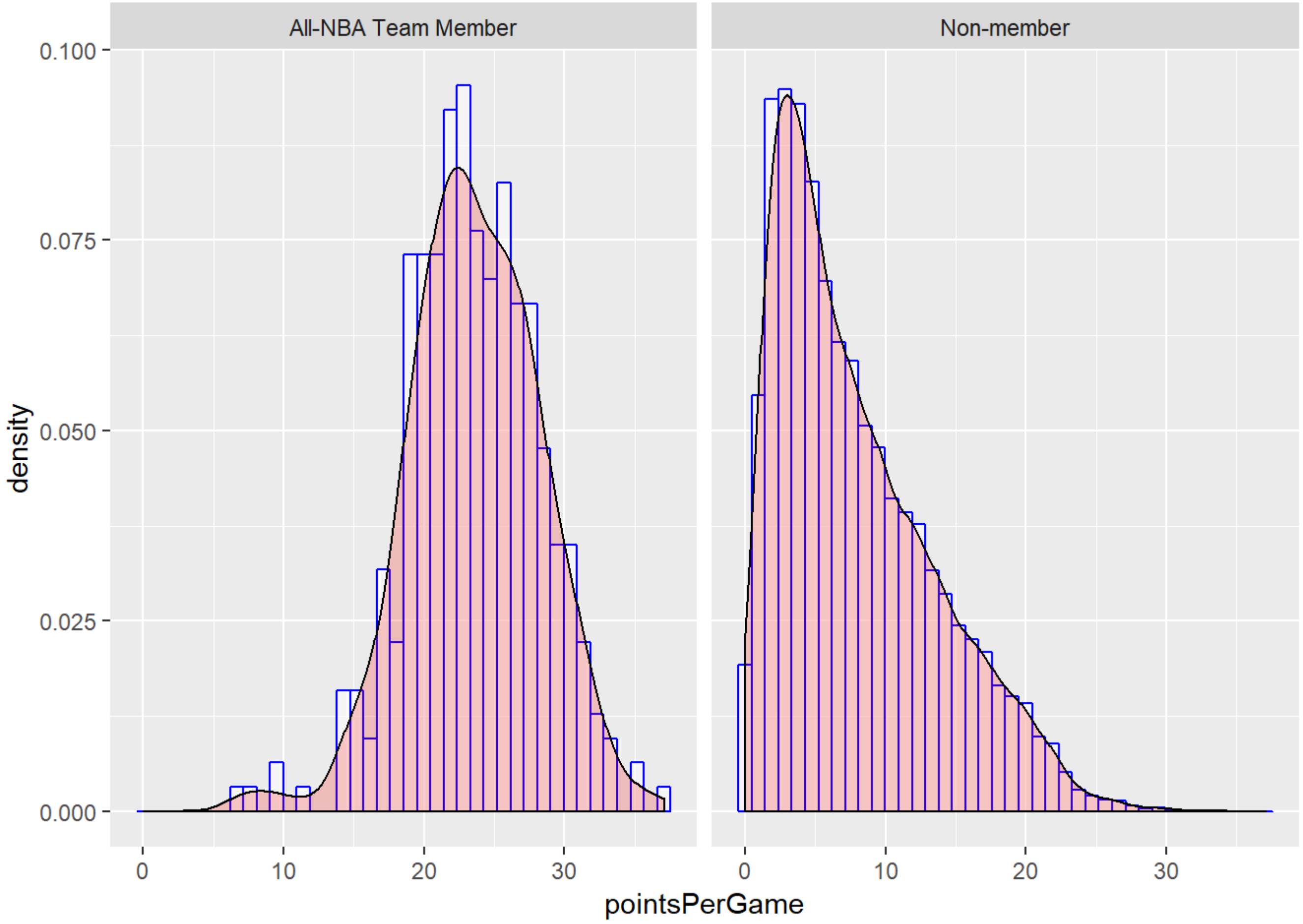

1.6
1.4
1.2
turnoversPerGame
1980
1990
2000
2010
year

Now we can see things a bit more clearly. It looks like the average game seemed to run at a slower pace in general, as most tracked stats saw downward trends - particularly points and turnovers per game. If we assume that most teams prioritized running plays in half court, rather than focusing on transition plays, it sheds some light on the increasing significance of the 3-point shot. Of course, since these are league averages, we could also interpret the overall decrease in tracked statistics as the game becoming much more team-oriented.

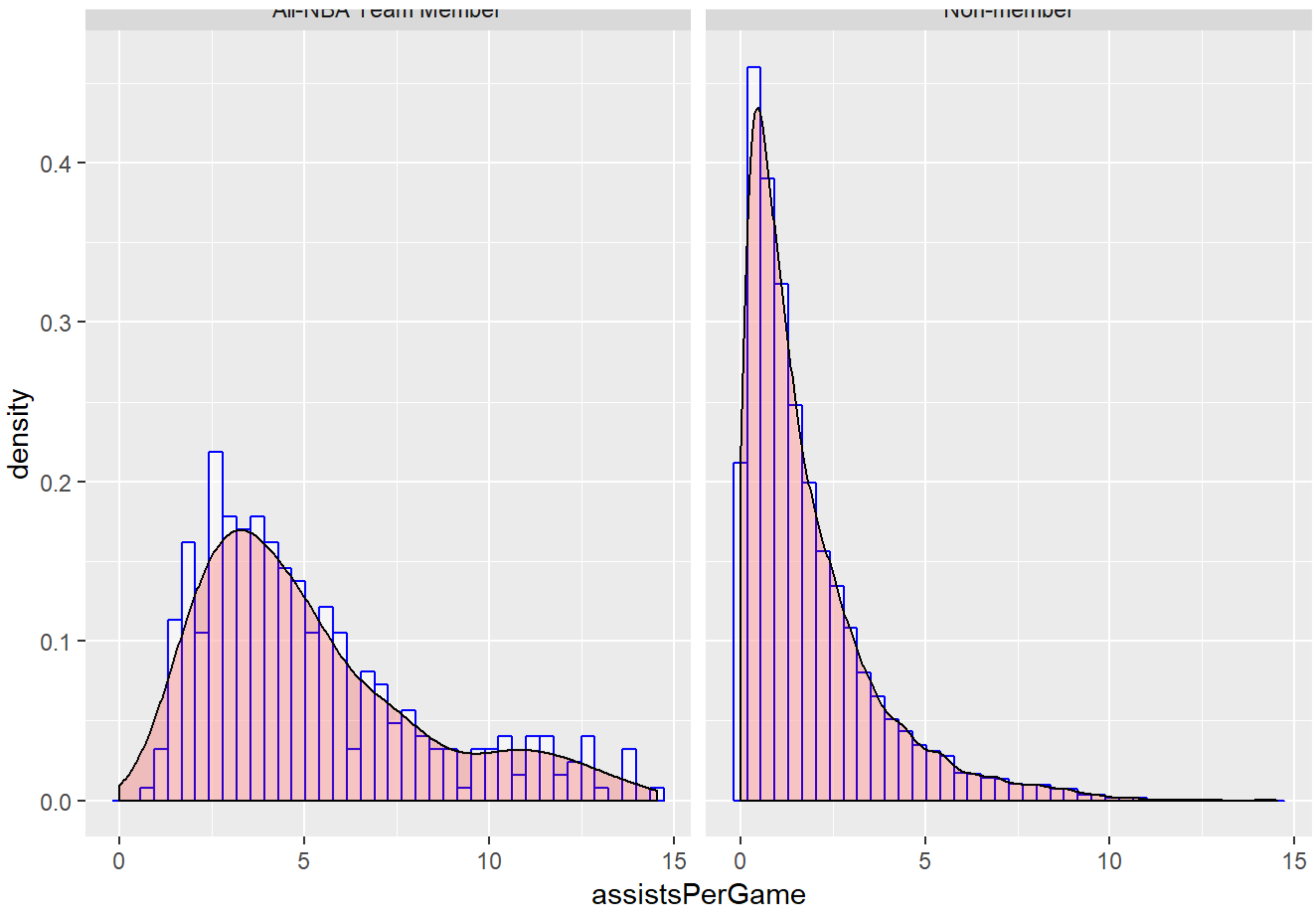
We are also interested in exploring the differences between players who made the All-NBA teams and those who did not. We can plot similar data separated by All-NBA team membership and create histograms to observe the stat distributions directly. Before we can do that, however, we need to create a single column that indicates membership of either team (as opposed to our current indicator variables).

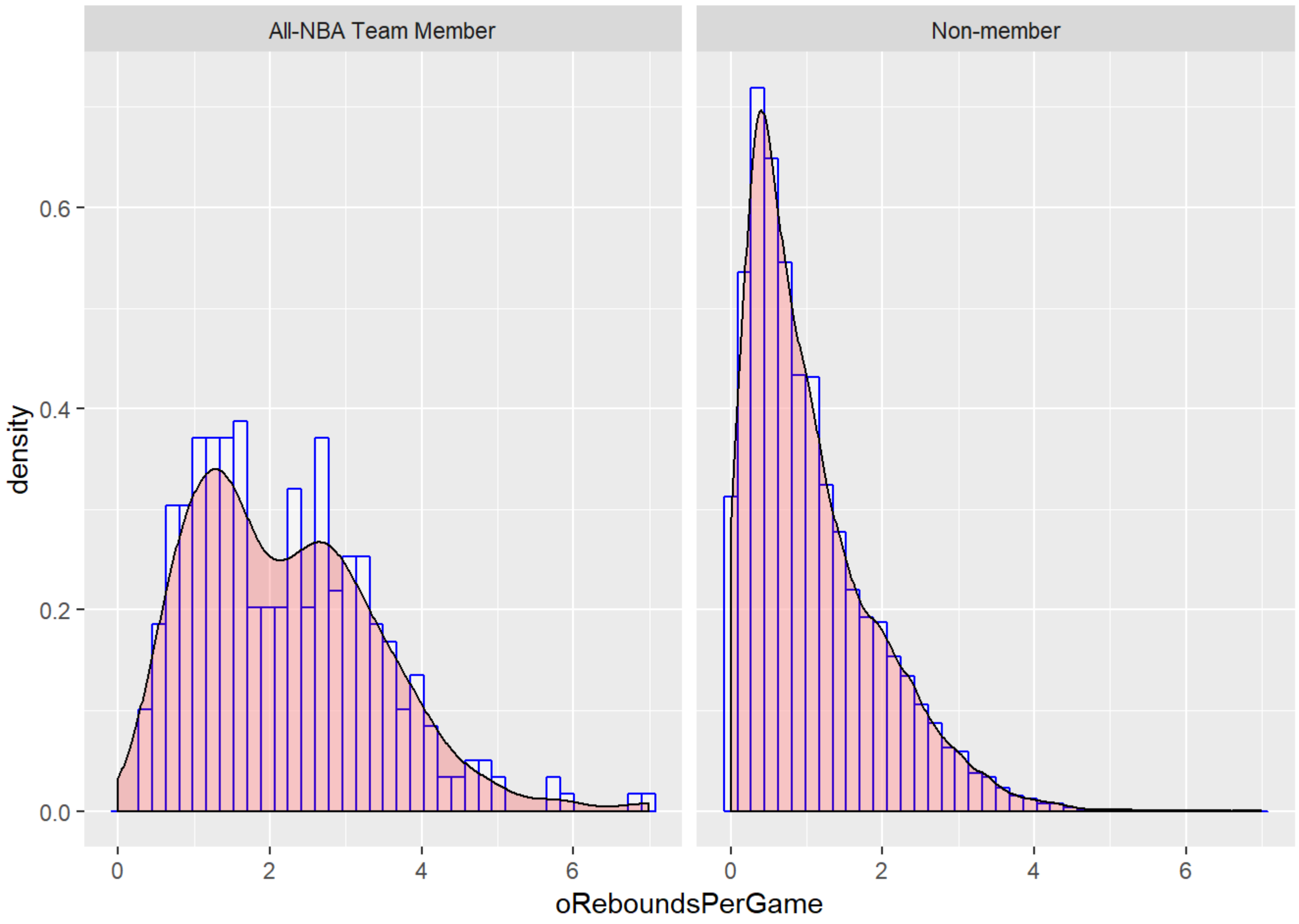
```
# A separate variable is created just for plotting
p <- players %>%
  mutate(
    distinction = case_when(
      grepl("All-NBA", award) ~ "All-NBA Team Member",
      TRUE ~ "Non-member")
  ) %>%
  select(contains("PerGame"), GP, PF, distinction) # Select certain columns (less plotting)

# Create histograms for each feature
for (i in names(p)) {
  if(i != "distinction"){
    plt <- p %>%
      ggplot(aes_string(i)) +
        geom_histogram(aes(y = ..density..), color = "blue", fill = "white", alpha = 0.5, bins =
40) +
        geom_density(alpha = 0.2, fill = "red") +
        facet_wrap(distinction ~ .)
      print(plt)
  }
}
```

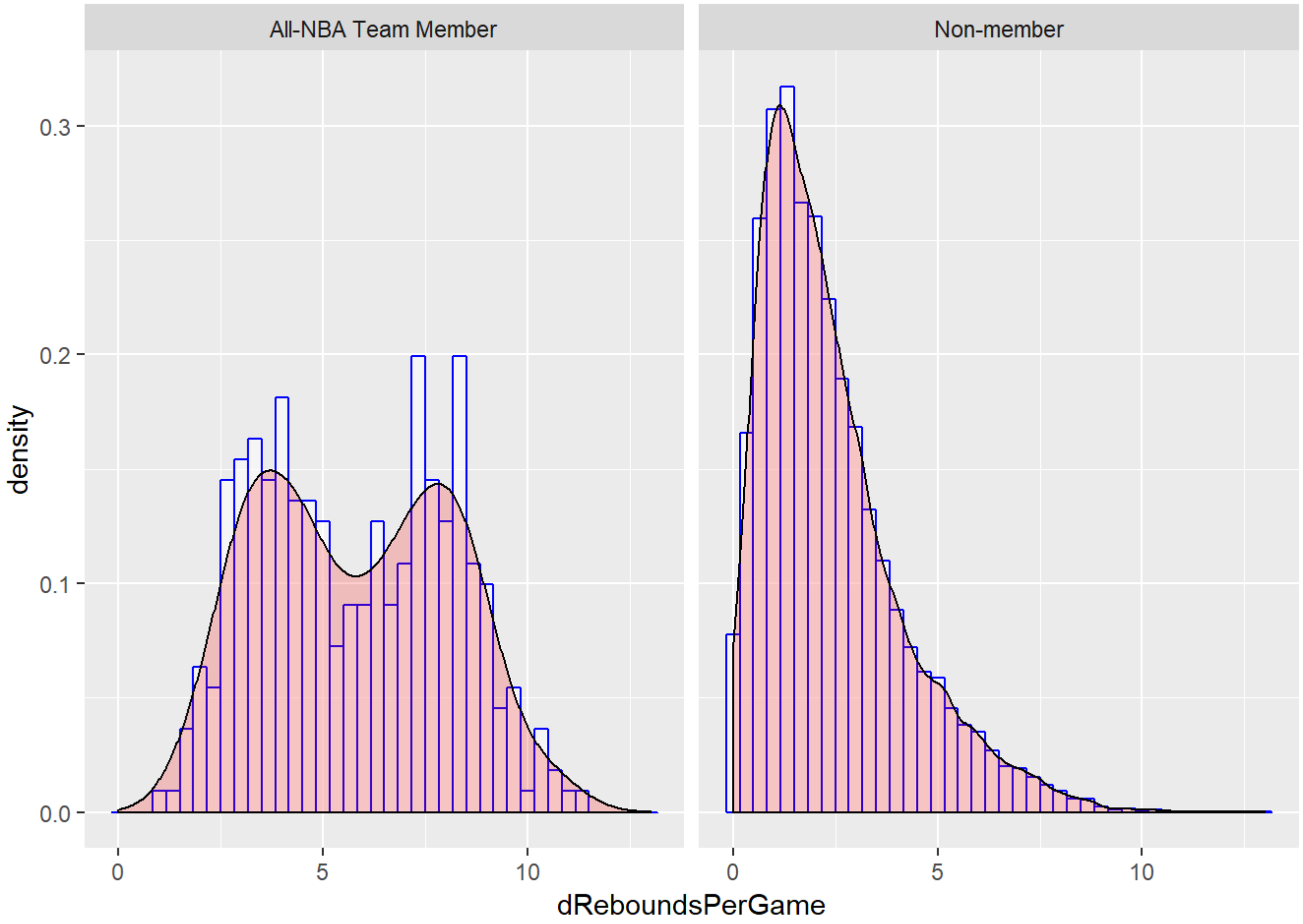

All-NBA Team Member
Non-member
density
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
0
10
20
30
40
minutesPerGame

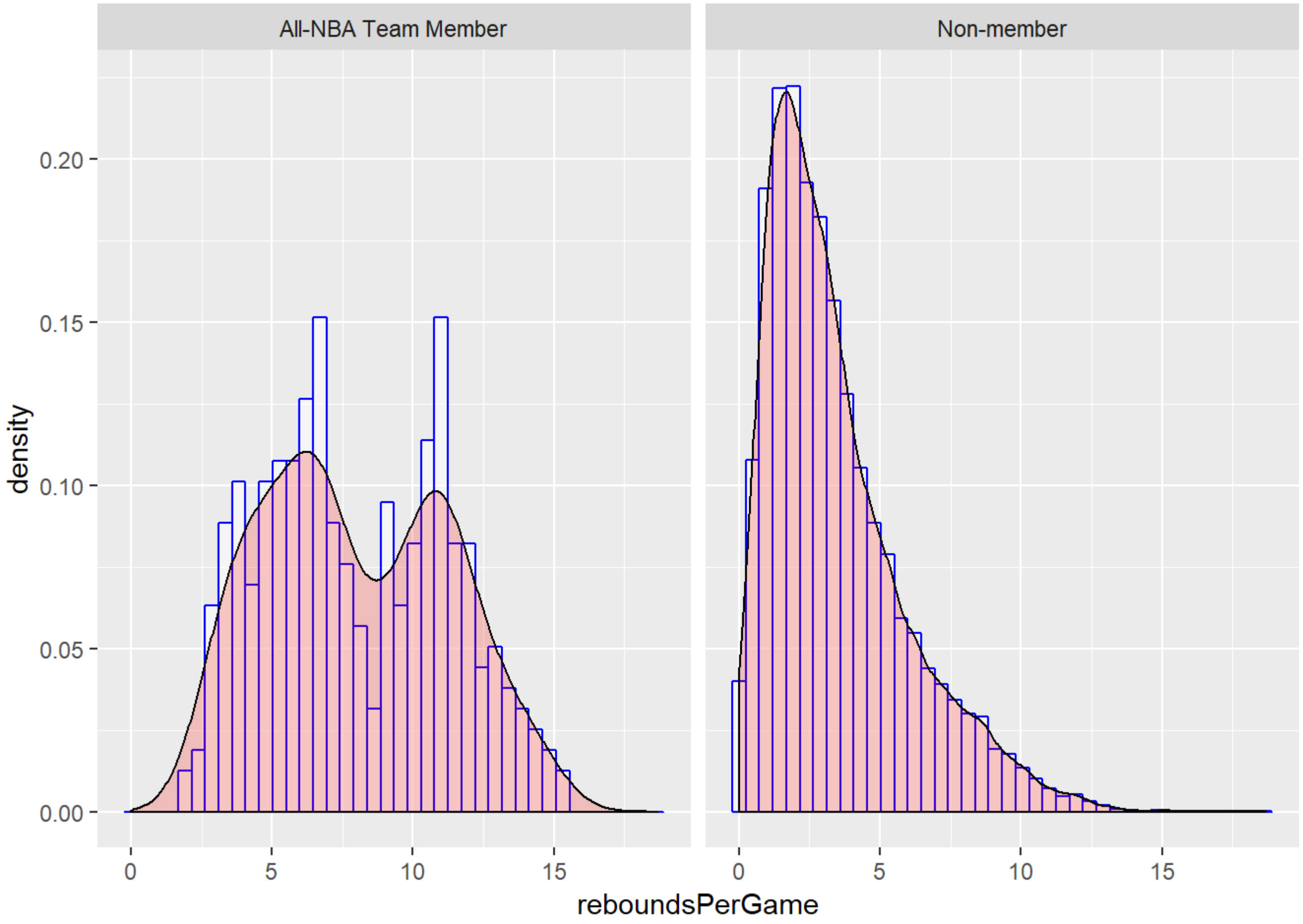

All-NBA Team Member
Non-member
density
0.100
0.075
0.050
0.025
0.000
0
10
20
30
0
10
20
30
pointsPerGame

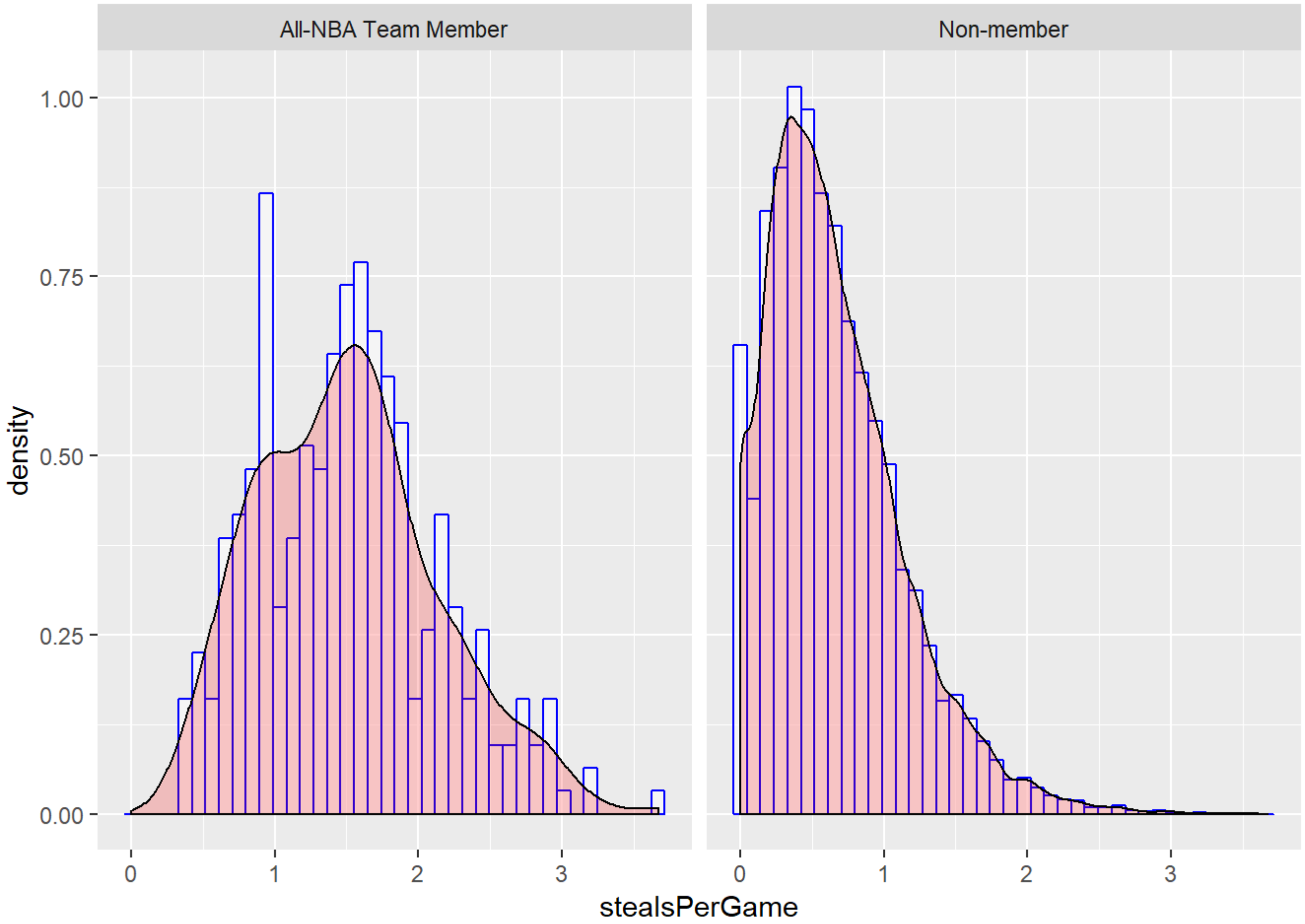
All-NBA Team Member
Non-member

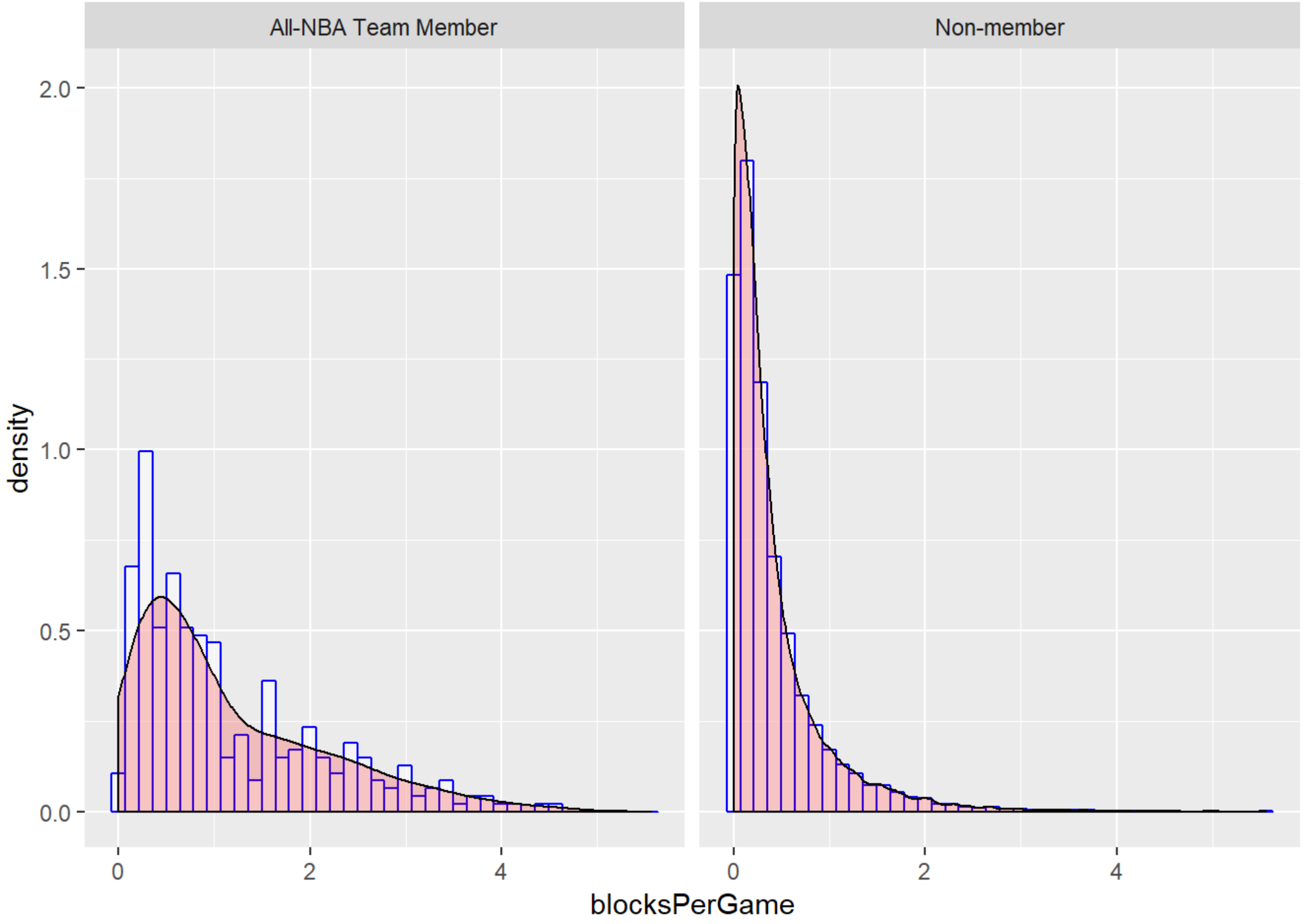

All-NBA Team Member
Non-member
density
0.4
0.3
0.2
0.1
0.0
0
5
10
15
0
5
10
15
assistsPerGame

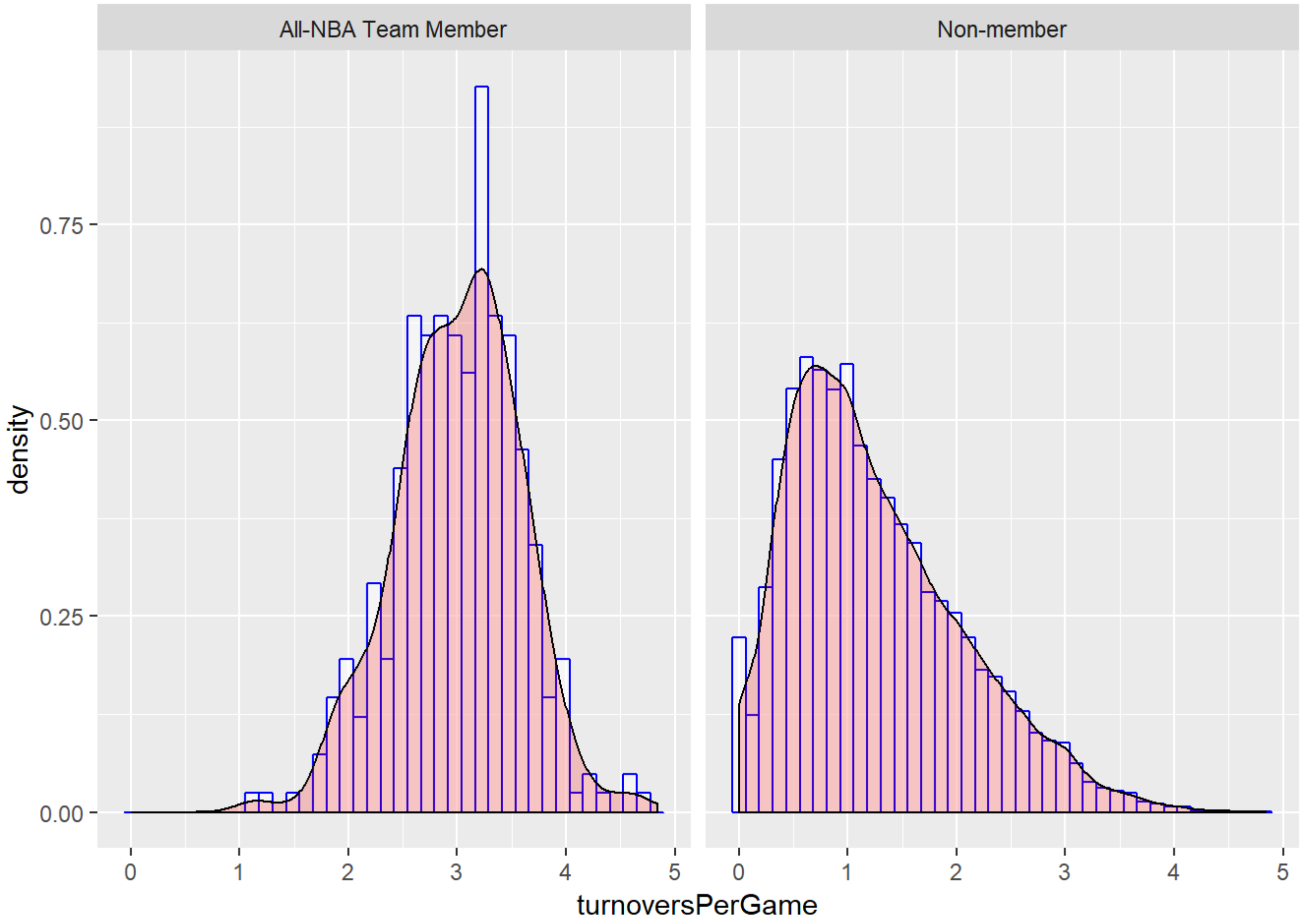

All-NBA Team Member
Non-member
density
0.6
0.4
0.2
0.0
0
2
4
6
0
2
4
6
oReboundsPerGame

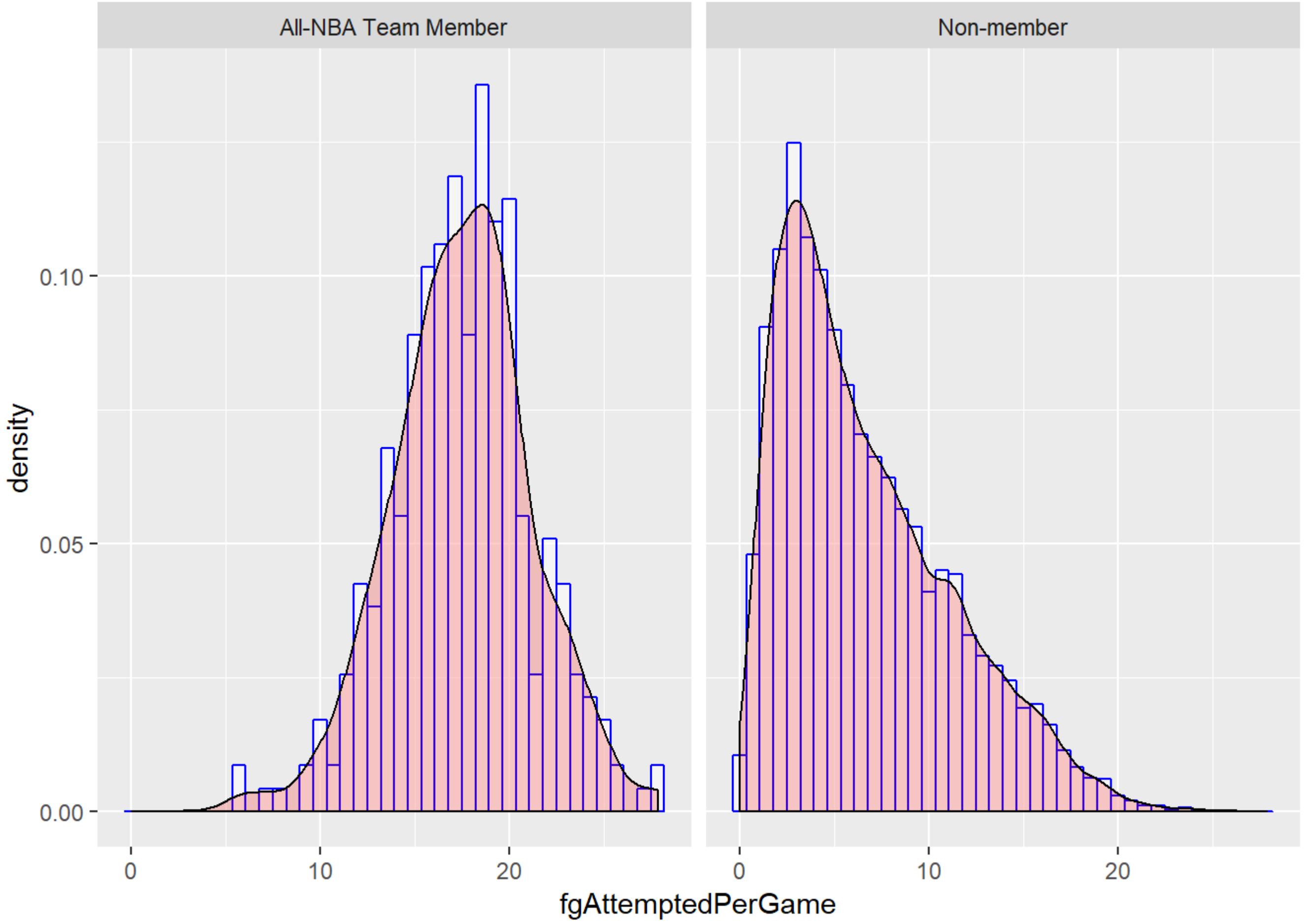

All-NBA Team Member
Non-member
density
0.3
0.2
0.1
0.0
0
5
10
0
5
10
dReboundsPerGame

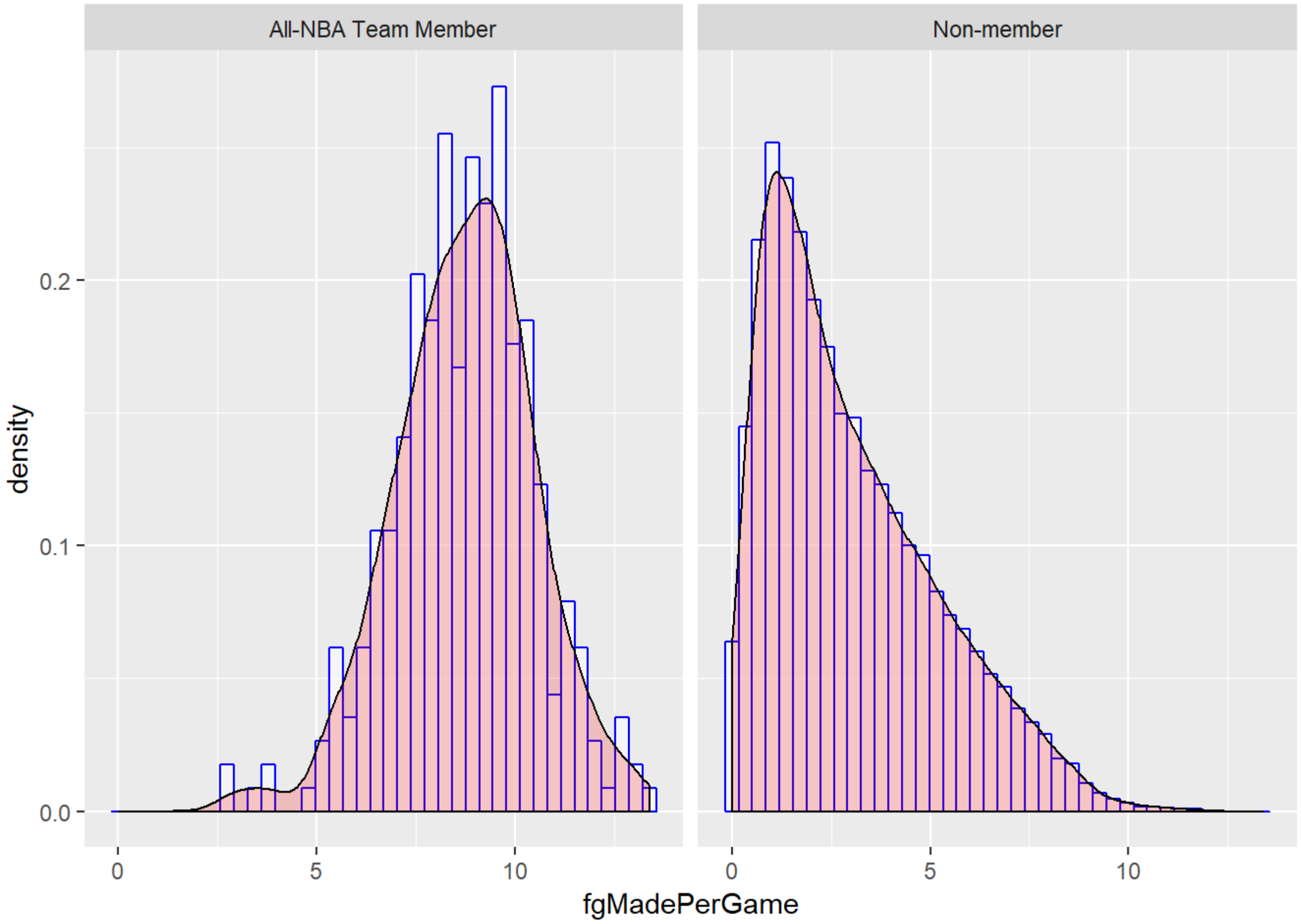

All-NBA Team Member
Non-member
density
0.20
0.15
0.10
0.05
0.00
0
5
10
15
0
5
10
15
reboundsPerGame

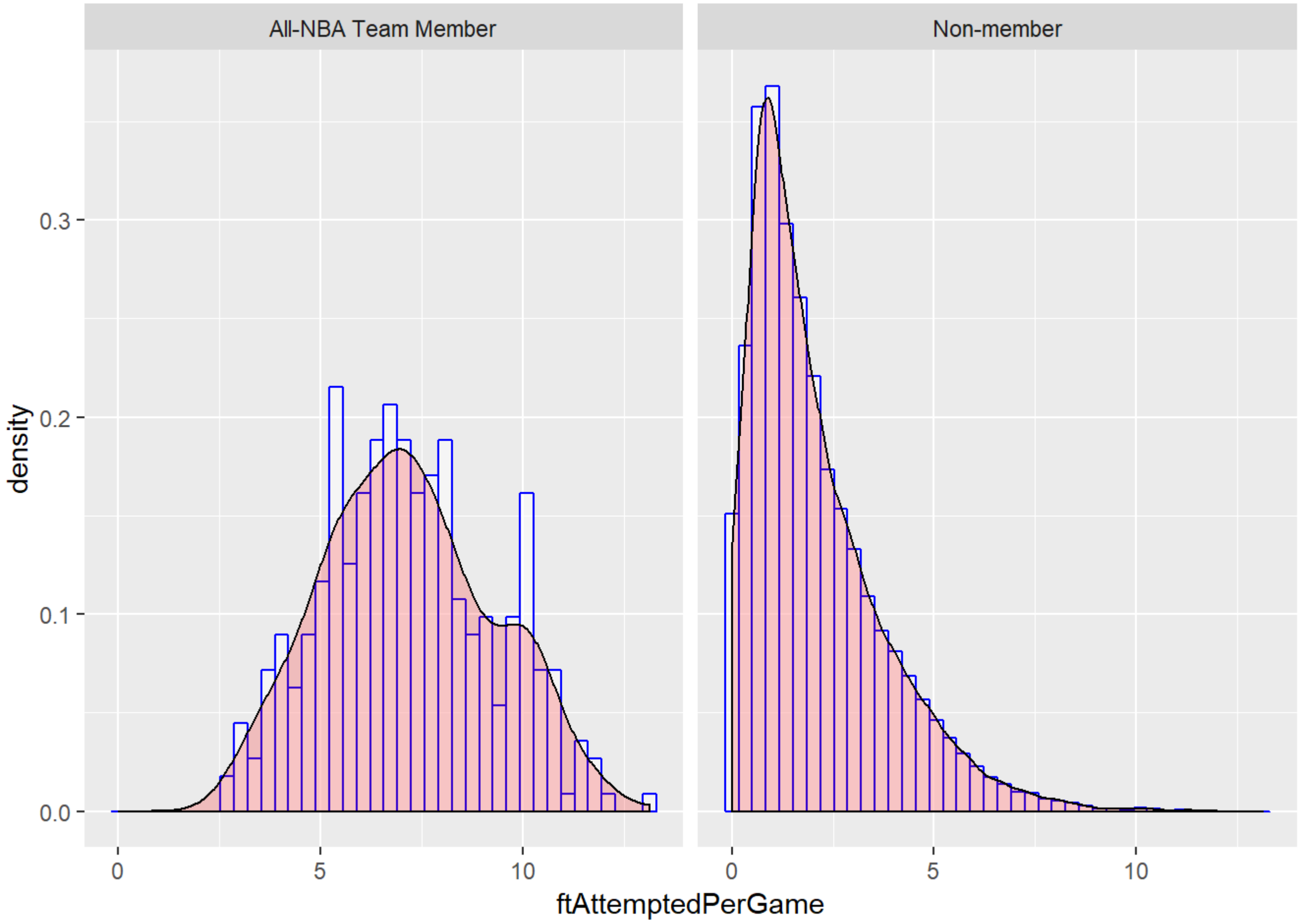

All-NBA Team Member
Non-member
density
1.00
0.75
0.50
0.25
0.00
0
1
2
3
stealsPerGame

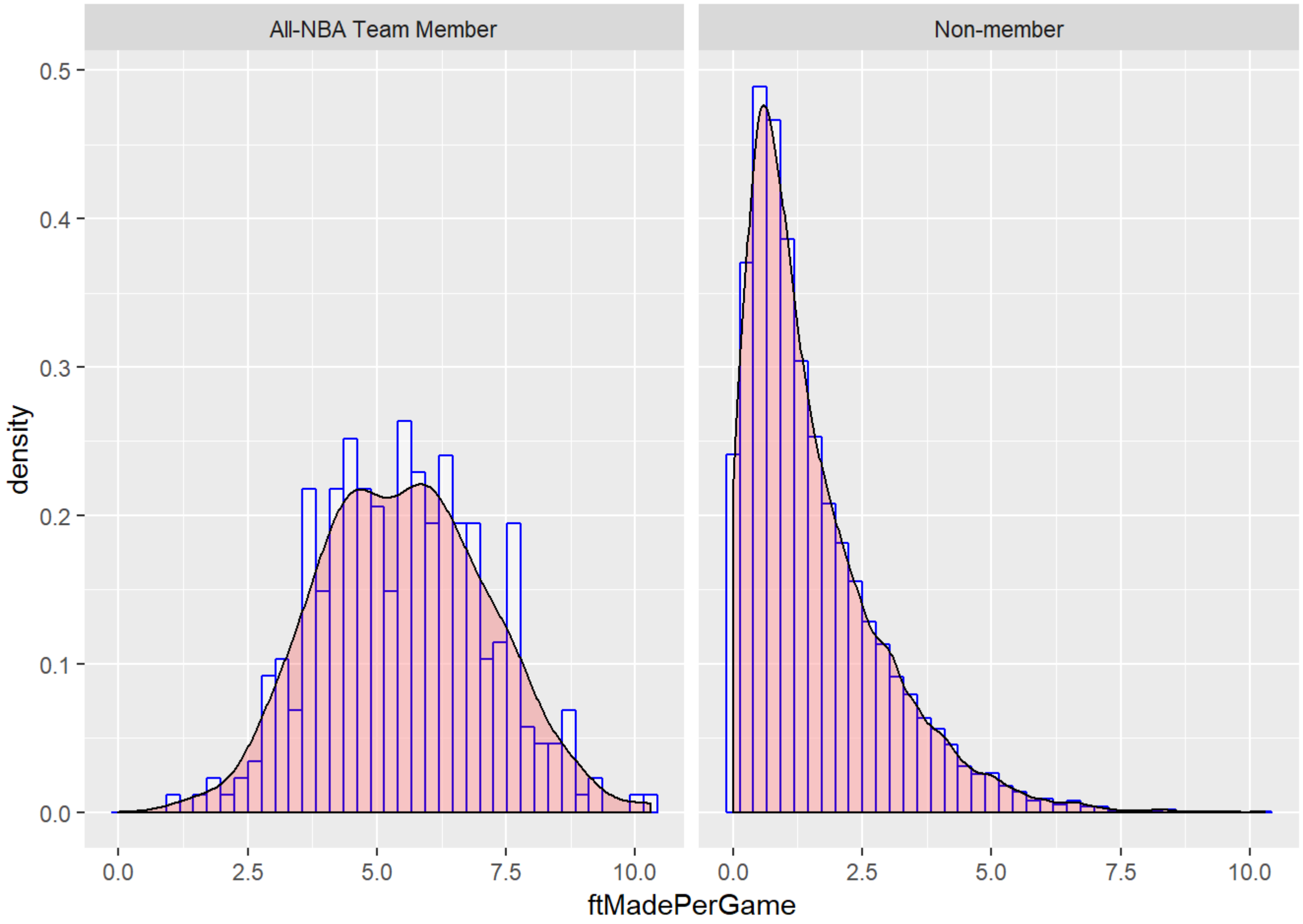

All-NBA Team Member
Non-member
density
2.0
1.5
1.0
0.5
0.0
0
2
4
blocksPerGame

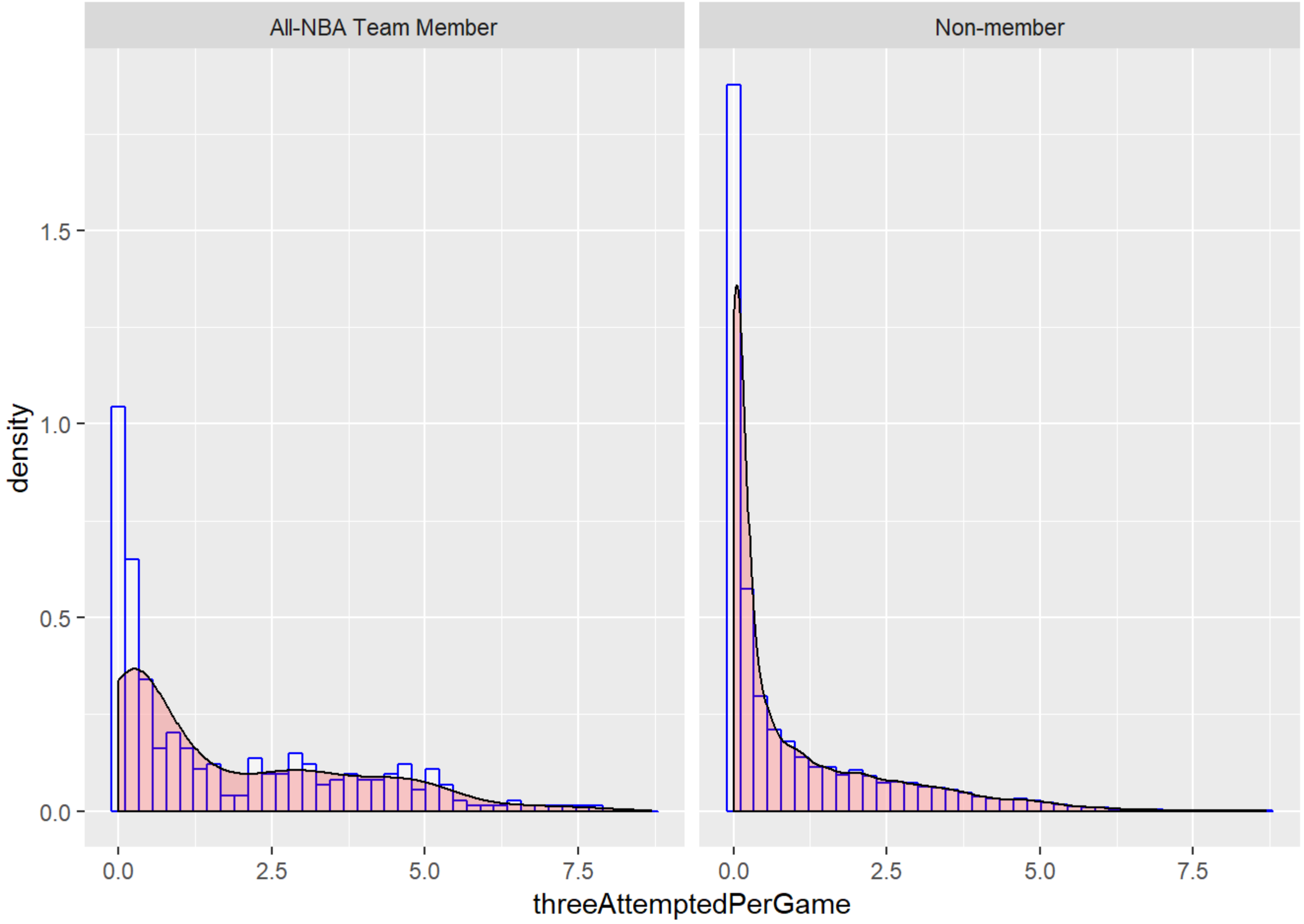

All-NBA Team Member
Non-member
density
0.75
0.50
0.25
0.00
0
1
2
3
4
5
turnoversPerGame

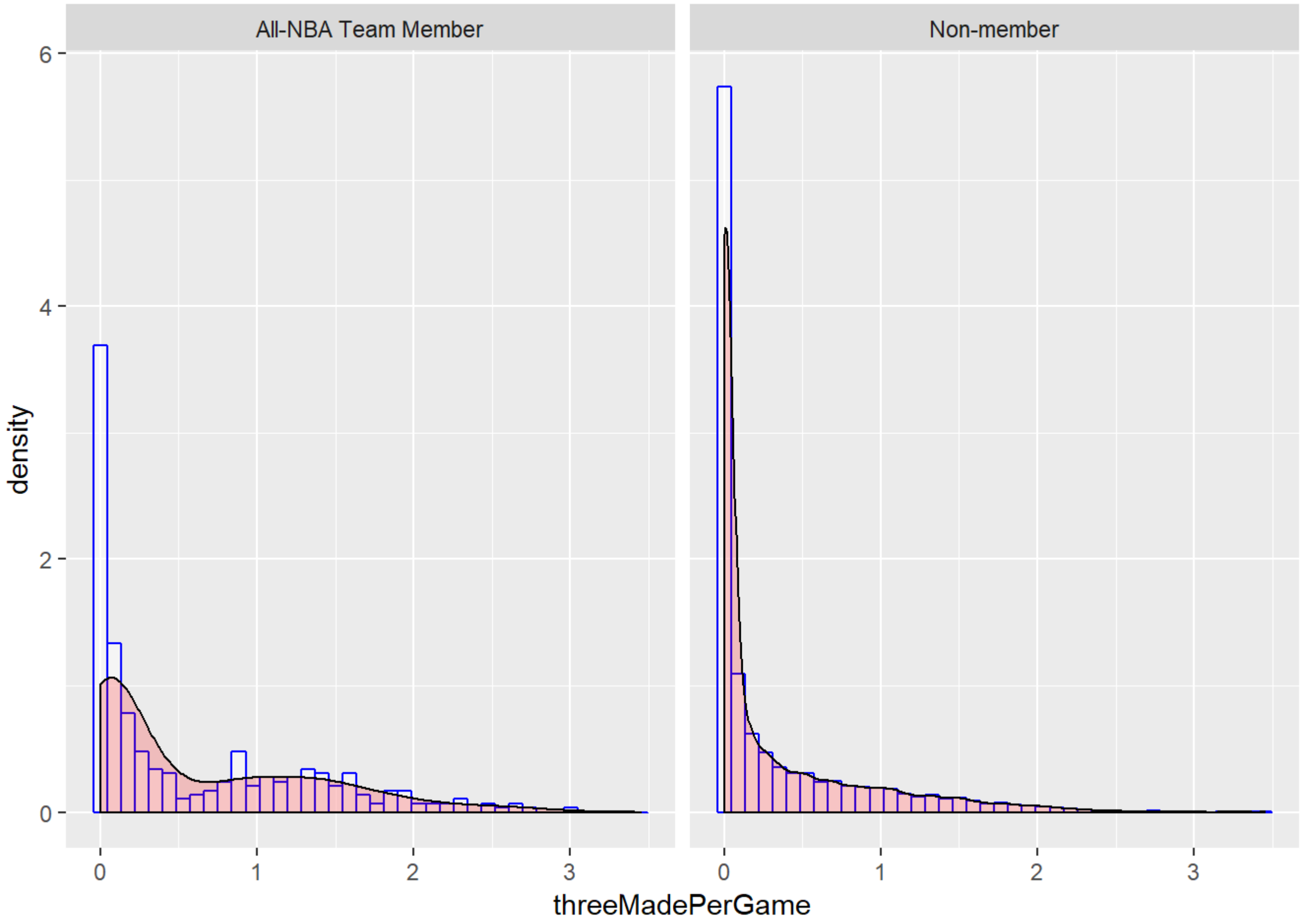

All-NBA Team Member
Non-member
density
0.10
0.05
0.00
0
10
20
fgAttemptedPerGame

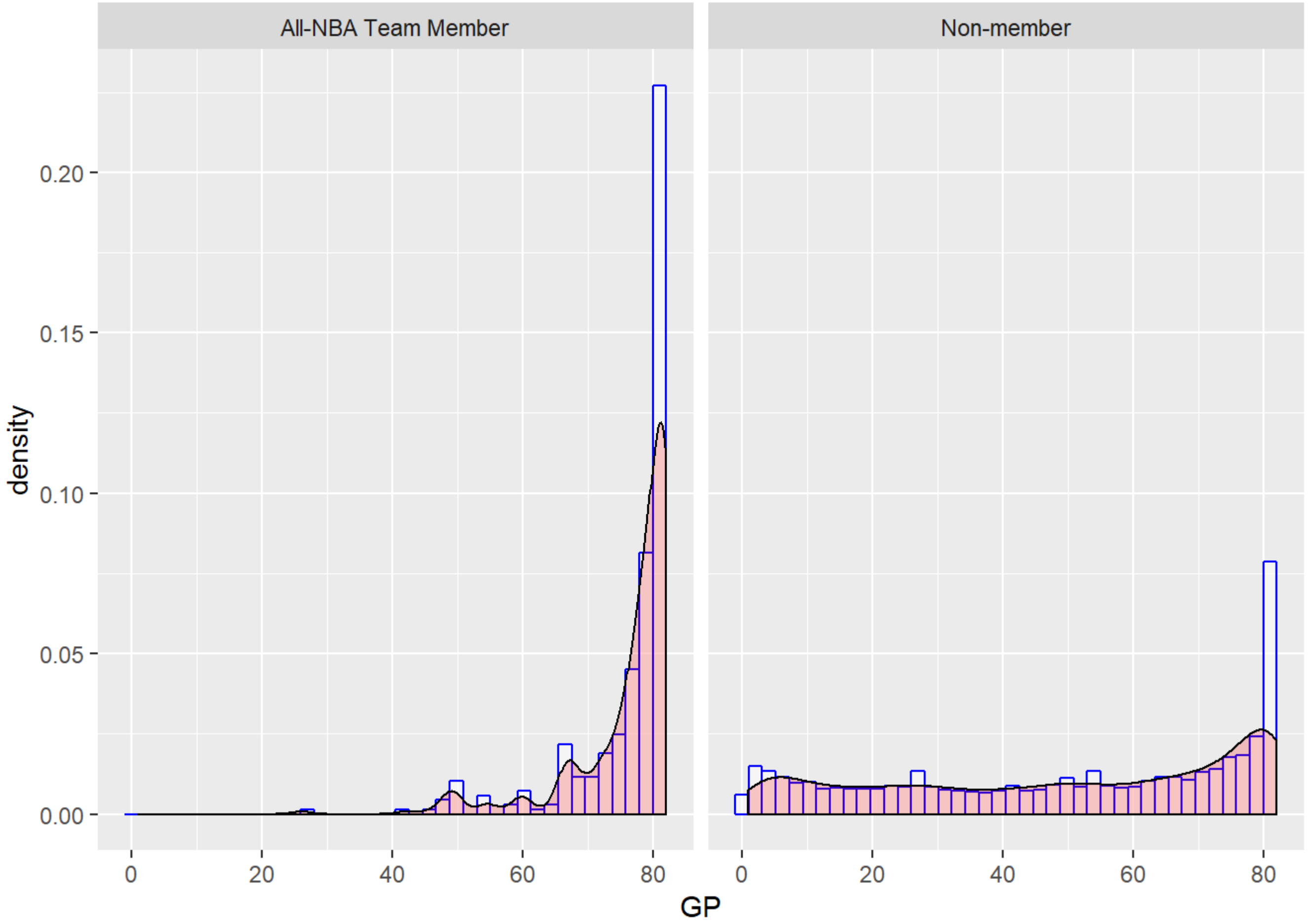

All-NBA Team Member
Non-member
density
0.2
0.1
0.0
0
5
10
0
5
10
fgMadePerGame

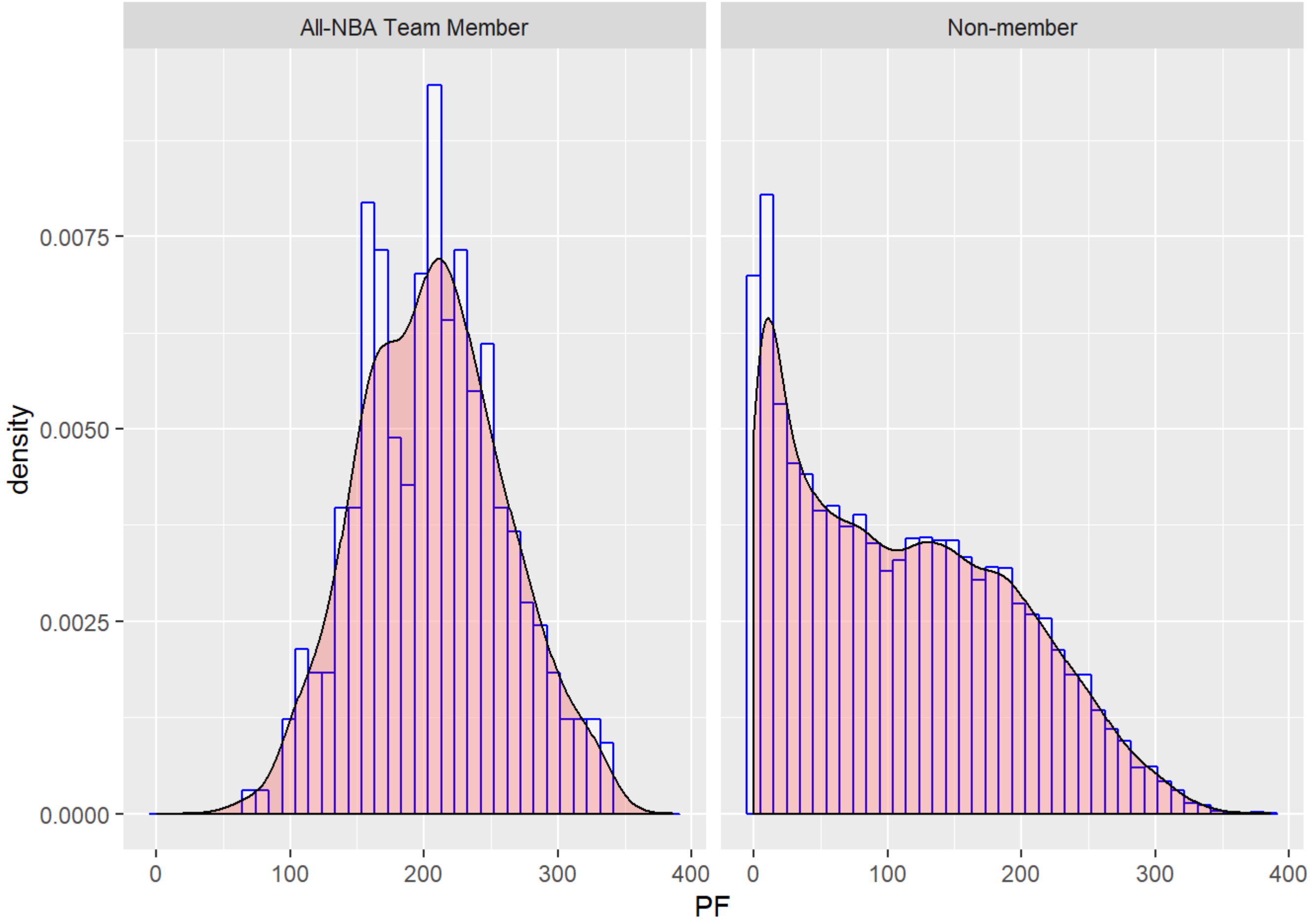

All-NBA Team Member
Non-member
density
0.3
0.2
0.1
0.0
0
5
10
0
5
10
ftAttemptedPerGame


All-NBA Team Member
Non-member
density
0.0
0.1
0.2
0.3
0.4
0.5
0.0
2.5
5.0
7.5
10.0
ftMadePerGame

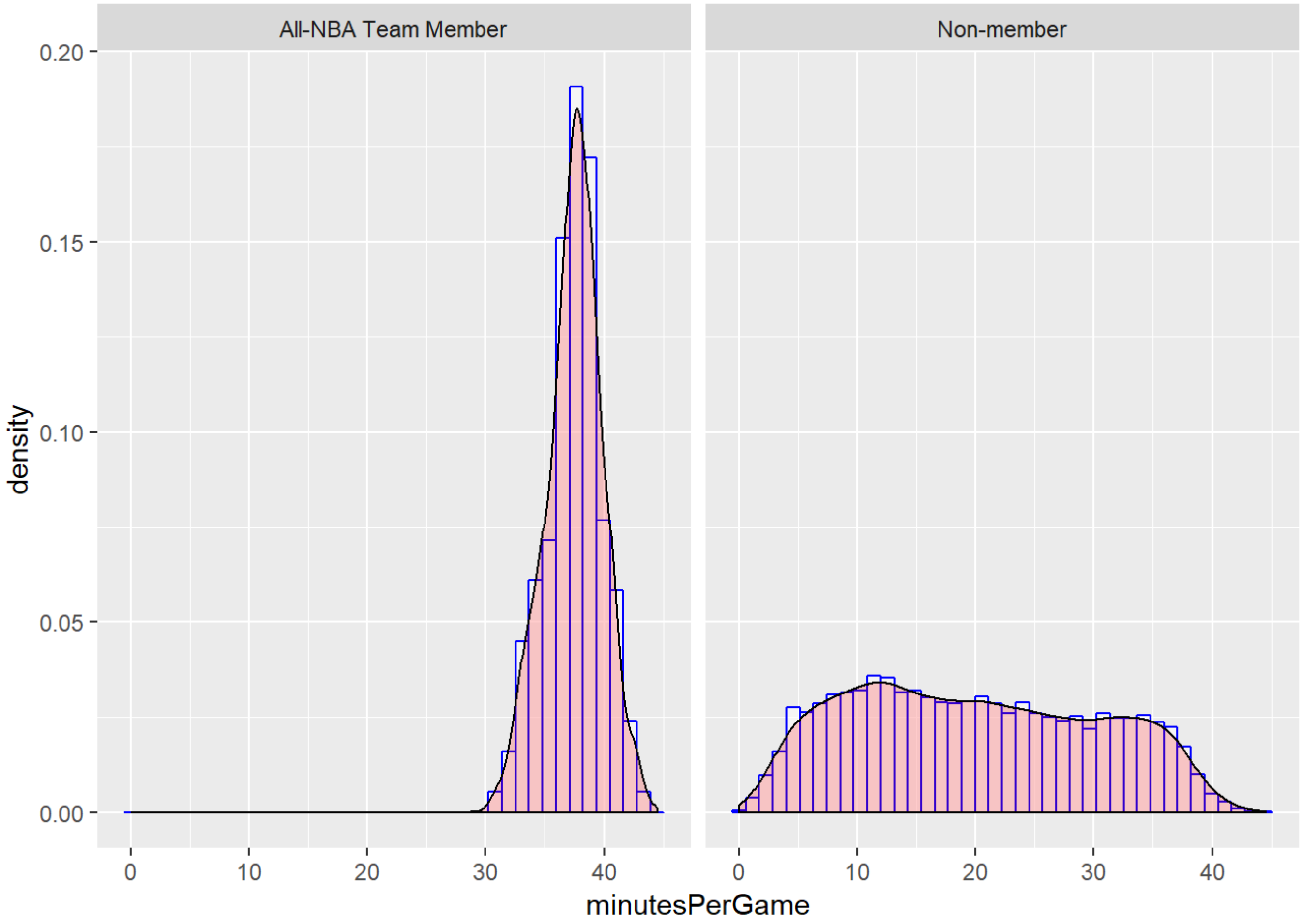

All-NBA Team Member
Non-member
density
0.0
0.5
1.0
1.5
0.0
2.5
5.0
7.5
threeAttemptedPerGame

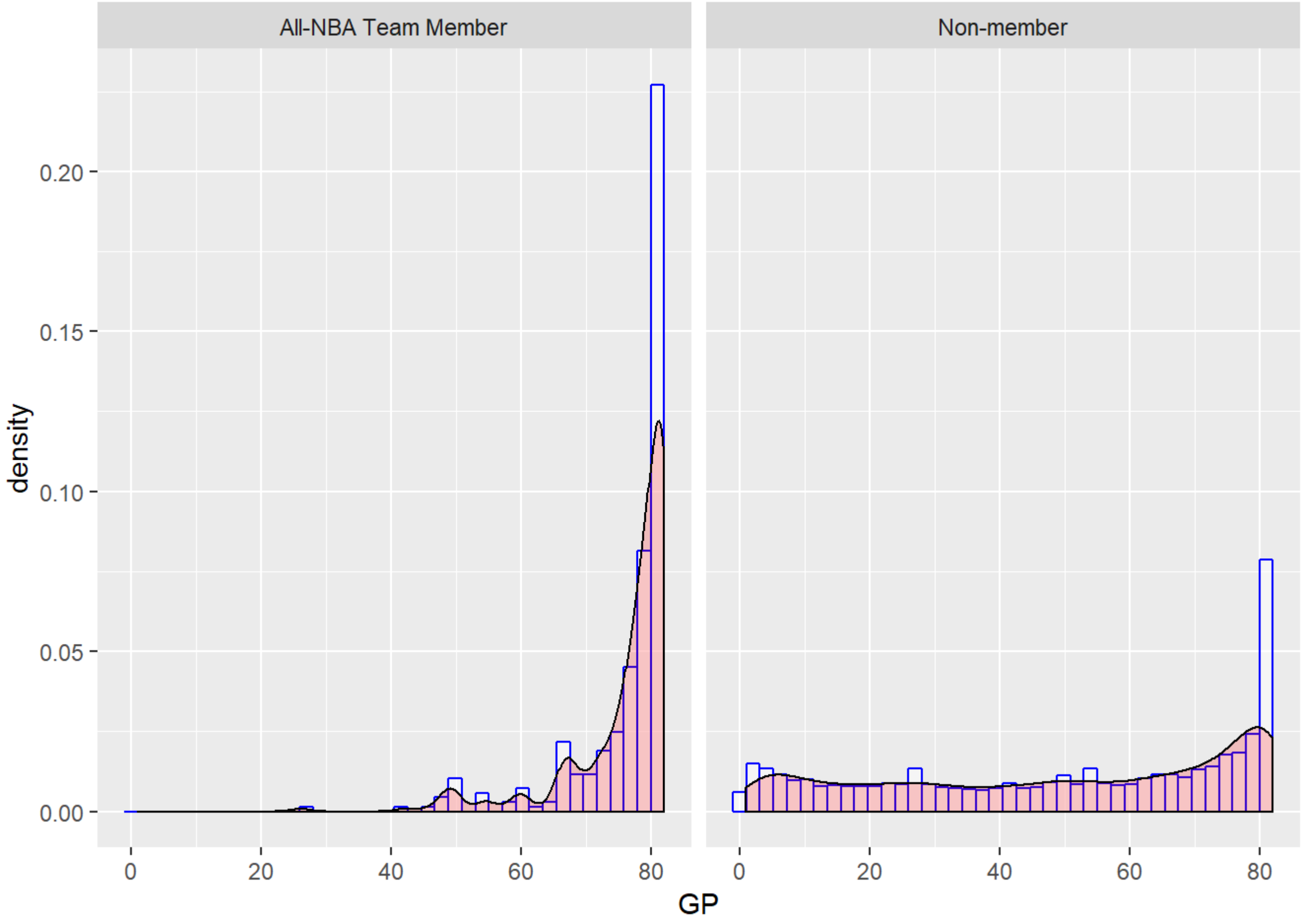

All-NBA Team Member
Non-member
density
6
4
2
0
0
1
2
3
0
1
2
3
threeMadePerGame


All-NBA Team Member
Non-member
density
0.20
0.15
0.10
0.05
0.00
0
20
40
60
80
0
20
40
60
80
GP



The code above overlays the density histograms with their kernel density estimates (basically smoothing out the histogram) for All-NBA team members versus non-members. As expected, players who made the All-NBA team generally performed better in every game, in regards to per-game points, assists, rebounds, etc. It also looks like they had more personal fouls and turnovers, but this can be attributed to the fact that they simply had **much more playtime**.


All-NBA Team Member
Non-member
density
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
minutesPerGame


All-NBA Team Member
Non-member
density
0.20
0.15
0.10
0.05
0.00
0
20
40
60
80
GP

We can see that All-NBA team members not only play for most of the game's duration, but they also play for most of the games *in the season*. This is probably a decent indicator of player health during the regular season, which is definitely a contributing factor to making the All-NBA team. We can add a couple features to account for this.

```
# Create a feature to show games played and health
players <- players %>%
  group_by(year) %>%
  mutate(GPRatio = GP / max(GP)) %>%
  ungroup() %>%
  mutate(healthy = case_when(
    GPRatio >= 0.7 ~ as.integer(1), # median of GPRatio
    TRUE ~ as.integer(0)
  ))
```

Some extra features are added to compare a player's performance to the his team and to the league. Note that, in order to avoid NaN entries, league and team offensive/defensive rebounds utilize `case_when()` to calculate the proportion of rebounds tallied for non-zero values, or set the value to zero otherwise. This is mostly to account for players who have very few games played.

```
# Add some stats to compare by season
players <- players %>%
  group_by(year) %>%
  mutate(
    lgPoints = points / sum(points),
    lgAssists = assists / sum(assists),
    lgRebounds = rebounds / sum(rebounds),
    lgDRebounds = case_when(
      dRebounds != 0 ~ dRebounds / sum(dRebounds),
      TRUE ~ 0),
    lgORebounds = case_when(
      oRebounds != 0 ~ oRebounds / sum(oRebounds),
      TRUE ~ 0)) %>%
  ungroup()

# Add some stats to compare by team
players <- players %>%
  group_by(year, tmID) %>%
  mutate(
    tmPoints = points / sum(points),
    tmAssists = assists / sum(assists),
    tmRebounds = rebounds / sum(rebounds),
    tmDRebounds = case_when(
      dRebounds != 0 ~ dRebounds / sum(dRebounds),
      TRUE ~ 0),
    tmORebounds = case_when(
      oRebounds != 0 ~ oRebounds / sum(oRebounds),
      TRUE ~ 0)) %>%
  ungroup()
```

The stats generated in the code above are calculated in separate groupings, and are ungrouped afterwards to add more player-specific features (see basketball-reference.com (https://www.basketball-reference.com/about/glossary.html#mp)).

```
# More individual player stats
players <- players %>%
  mutate(
    ftPct = ftMade / ftAttempted,
    fgPct = fgMade / fgAttempted,
    efgPct = (fgMade + 0.5 * threeMade) / fgAttempted,
    astTovRatio = case_when(
      turnovers != 0 ~ assists / turnovers,
      TRUE ~ 0),
    dReboundPct = dRebounds / rebounds,
    oReboundPct = oRebounds / rebounds,
    totalGameScore = points + 0.4 * (fgMade + threeMade) - 0.7 * fgAttempted - 0.4 * (ftAttempted - ftMade) + 0.7 * oRebounds + 0.3 * dRebounds + steals + 0.7 * assists + 0.7 * blocks - 0.4 * PF - turnovers,
    avgGameScore = totalGameScore / GP)
```

Summary of new features added:

- Per-game statistics (points, rebounds, assists, etc.)
- League/Team points/assists/rebounds
- Field Goal and Free Throw Percentage (FG%/FT%)
- Effective Field Goal Percentage (eFG%)
- Assist-to-Turnover Ratio
- Total and Average Game Score
- Games played ratio
- Indicator for health

```
glimpse(players)
```

```
## Observations: 14,577
## Variables: 66
## $ playerID              <fct> abdulka01, abernto01, adamsal01, archina...
## $ year                  <int> 1979, 1979, 1979, 1979, 1979, 1979, 1979...
## $ tmID                  <fct> LAL, GSW, PHO, BOS, CHI, WSB, SEA, WSB, ...
## $ GP                    <int> 82, 67, 75, 80, 26, 20, 67, 82, 77, 20, ...
## $ minutes               <int> 3143, 1222, 2168, 2864, 560, 180, 726, 2...
## $ points                <int> 2034, 362, 1118, 1131, 86, 38, 312, 1277...
## $ oRebounds             <int> 190, 62, 158, 59, 29, 6, 71, 240, 192, 3...
## $ dRebounds             <int> 696, 129, 451, 138, 86, 22, 126, 398, 26...
## $ rebounds              <int> 886, 191, 609, 197, 115, 28, 197, 638, 4...
## $ assists               <int> 371, 87, 322, 671, 40, 26, 28, 159, 279,...
## $ steals                <int> 81, 35, 108, 106, 12, 7, 21, 90, 85, 5, ...
## $ blocks                <int> 280, 12, 55, 10, 15, 4, 54, 36, 49, 12, ...
## $ turnovers             <int> 297, 39, 218, 242, 27, 11, 79, 133, 189,...
## $ PF                    <int> 216, 118, 237, 218, 66, 18, 116, 197, 26...
## $ fgAttempted           <int> 1383, 318, 875, 794, 60, 35, 271, 1101, ...
## $ fgMade                <int> 835, 153, 465, 383, 27, 16, 122, 545, 38...
## $ ftAttempted           <int> 476, 82, 236, 435, 50, 13, 101, 227, 209...
## $ ftMade                <int> 364, 56, 188, 361, 32, 5, 68, 171, 139, ...
## $ threeAttempted        <int> 1, 1, 2, 18, 0, 1, 0, 47, 3, 0, 221, 0, ...
## $ threeMade             <int> 0, 0, 0, 4, 0, 1, 0, 16, 1, 0, 73, 0, 0,...
## $ center                <int> 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0...
## $ forward               <int> 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0...
## $ guard                 <int> 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1...
## $ award                 <fct> All-Defensive First Team, All-NBA First ...
## $ allDefFirstTeam       <int> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ allDefSecondTeam      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ allNBAFirstTeam       <int> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ allNBASecondTeam      <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ MVP                   <int> 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ defPOTY               <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ allstar               <int> 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ minutesPerGame        <dbl> 38.329268, 18.238806, 28.906667, 35.8000...
## $ pointsPerGame         <dbl> 24.804878, 5.402985, 14.906667, 14.13750...
## $ assistsPerGame        <dbl> 4.5243902, 1.2985075, 4.2933333, 8.38750...
## $ oReboundsPerGame      <dbl> 2.3170732, 0.9253731, 2.1066667, 0.73750...
## $ dReboundsPerGame      <dbl> 8.4878049, 1.9253731, 6.0133333, 1.72500...
## $ reboundsPerGame       <dbl> 10.8048780, 2.8507463, 8.1200000, 2.4625...
## $ stealsPerGame         <dbl> 0.9878049, 0.5223881, 1.4400000, 1.32500...
## $ blocksPerGame         <dbl> 3.41463415, 0.17910448, 0.73333333, 0.12...
## $ turnoversPerGame      <dbl> 3.6219512, 0.5820896, 2.9066667, 3.02500...
## $ fgAttemptedPerGame    <dbl> 16.865854, 4.746269, 11.666667, 9.925000...
## $ fgMadePerGame         <dbl> 10.182927, 2.283582, 6.200000, 4.787500,...
## $ ftAttemptedPerGame    <dbl> 5.804878, 1.223881, 3.146667, 5.437500, ...
## $ ftMadePerGame         <dbl> 4.4390244, 0.8358209, 2.5066667, 4.51250...
## $ threeAttemptedPerGame <dbl> 0.01219512, 0.01492537, 0.02666667, 0.22...
## $ threeMadePerGame      <dbl> 0.00000000, 0.00000000, 0.00000000, 0.05...
## $ GPRatio               <dbl> 1.00000000, 0.81707317, 0.91463415, 0.97...
## $ healthy               <int> 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0...
## $ lgPoints              <dbl> 1.031236e-02, 1.835337e-03, 5.668250e-03...
## $ lgAssists             <dbl> 7.967015e-03, 1.868276e-03, 6.914768e-03...
## $ lgRebounds            <dbl> 0.0109295010, 0.0023561340, 0.0075124900...
## $ lgDRebounds           <dbl> 1.291352e-02, 2.393454e-03, 8.367813e-03...
## $ lgORebounds           <dbl> 0.0069935218, 0.0022820966, 0.0058156655...
## $ tmPoints              <dbl> 0.215511761, 0.042623337, 0.122668422, 0...
## $ tmAssists             <dbl> 0.1537505180, 0.0428994083, 0.1410424880...
## $ tmRebounds            <dbl> 0.237025147, 0.053173719, 0.172570133, 0...
## $ tmDRebounds           <dbl> 0.262641509, 0.052933935, 0.183482506, 0...
## $ tmORebounds           <dbl> 0.174632353, 0.053679654, 0.147525677, 0...
## $ ftPct                 <dbl> 0.7647059, 0.6829268, 0.7966102, 0.82988...
## $ fgPct                 <dbl> 0.6037599, 0.4811321, 0.5314286, 0.48236...
## $ efgPct                <dbl> 0.6037599, 0.4811321, 0.5314286, 0.48488...
## $ astTovRatio           <dbl> 1.2491582, 2.2307692, 1.4770642, 2.77272...
## $ dReboundPct           <dbl> 0.7855530, 0.6753927, 0.7405583, 0.70050...
## $ oReboundPct           <dbl> 0.2144470, 0.3246073, 0.2594417, 0.29949...
## $ totalGameScore        <dbl> 1850.2, 290.4, 977.3, 1036.6, 90.8, 37.7...
## $ avgGameScore          <dbl> 22.563415, 4.334328, 13.030667, 12.95750...
```

Future work to be done