PCA Analysis on NBA Data

Mu-Chia Lee, Rebeka Róth

2024-02-05

Contents

Introduction

In this analysis, we will perform a Principal Component Analysis (PCA) on a dataset containing NBA player statistics. The dataset contains various player statistics specifically related to play by play, such as bad passes, lost ball, and offensive fouls. The goal of this analysis is to identify the most important components that explain the variance in the dataset and to visualize the relationships between the variables and the observations.

Data Preparation

```
# Read the dataset
player_season_stats <- read_csv("../../project/nba_data/Player Play By Play.csv")

# Filter for the past 3 seasons and remove players with less than 1000 minutes played
player_season_stats <- player_season_stats %>%
  filter(season <= 2023 & season > 2020 & mp > 1000)
```

```r
# Convert seas_id and player_id to strings and drop birth_year
player_season_stats <- player_season_stats %>%
  filter(season <= 2023 & season > 2020 & mp > 1000) %>%
  mutate(seas_id = as.character(seas_id),
         player_id = as.character(player_id),
         birth_year = NULL) %>%  # This removes the birth_year column
  mutate(across(where(is.numeric), ~replace_na(., 0)))


# View the dimensions and the head of the modified dataset
dim(player_season_stats)
```

```
## [1] 832  27
```

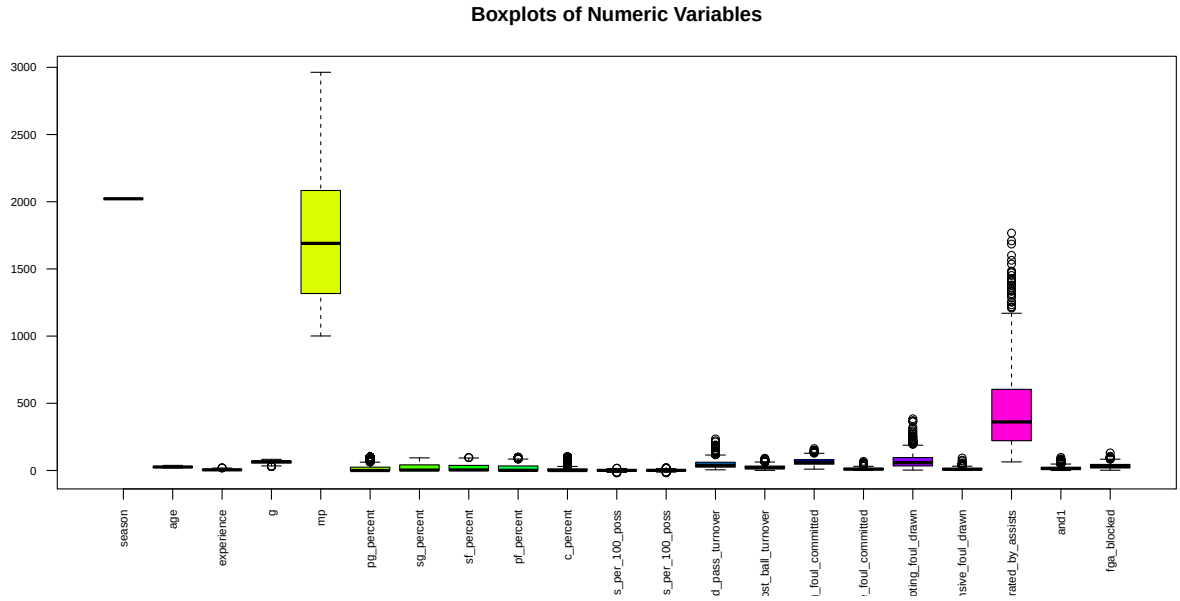
```r
head(player_season_stats)
```

```
## # A tibble: 6 x 27
##   seas_id season player_id player pos     age experience lg    tm        g    mp
##   <chr>    <dbl> <chr>     <chr>  <chr> <dbl>      <dbl> <chr> <chr> <dbl> <dbl>
## 1 30467     2023 5027      AJ Gr~ SF       19          1 NBA   ATL      72  1401
## 2 30462     2023 4219      Aaron~ PF       27          9 NBA   DEN      68  2055
## 3 30464     2023 4805      Aaron~ SF       23          3 NBA   IND      73  1816
## 4 30465     2023 4900      Aaron~ SG       24          2 NBA   OKC      70  1297
## 5 30468     2023 3734      Al Ho~ C        36         16 NBA   BOS      63  1922
## 6 30469     2023 3982      Alec ~ SG       31         12 NBA   DET      51  1122
## # i 16 more variables: pg_percent <dbl>, sg_percent <dbl>, sf_percent <dbl>,
## #   pf_percent <dbl>, c_percent <dbl>, on_court_plus_minus_per_100_poss <dbl>,
## #   net_plus_minus_per_100_poss <dbl>, bad_pass_turnover <dbl>,
## #   lost_ball_turnover <dbl>, shooting_foul_committed <dbl>,
## #   offensive_foul_committed <dbl>, shooting_foul_drawn <dbl>,
## #   offensive_foul_drawn <dbl>, points_generated_by_assists <dbl>, and1 <dbl>,
## #   fga_blocked <dbl>
```

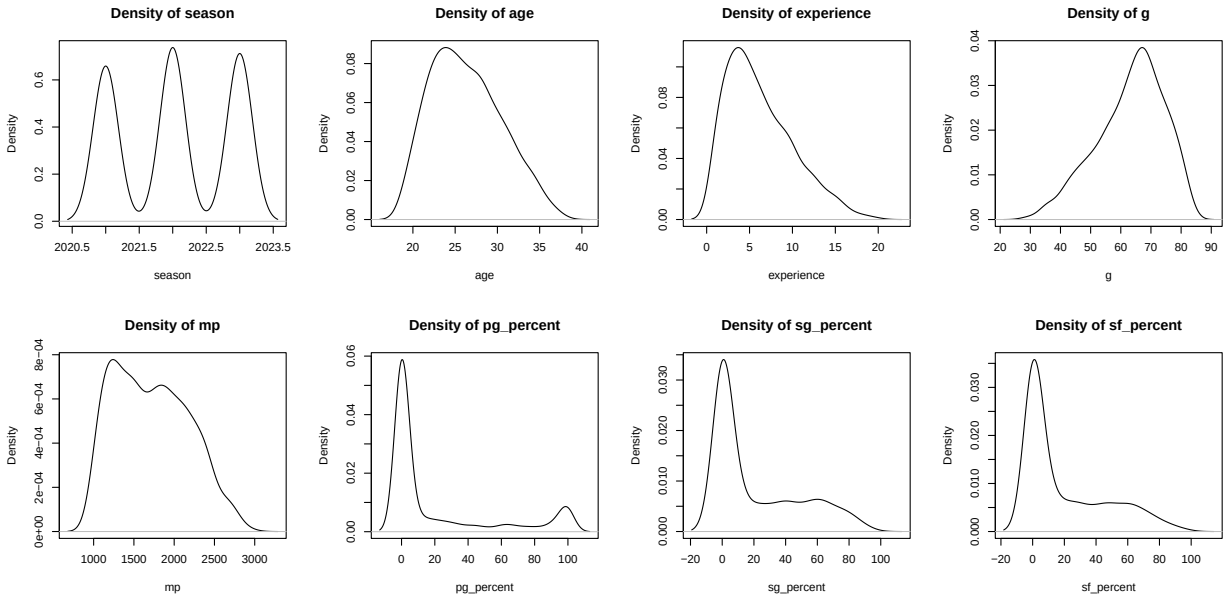
Exploratory Data Analysis

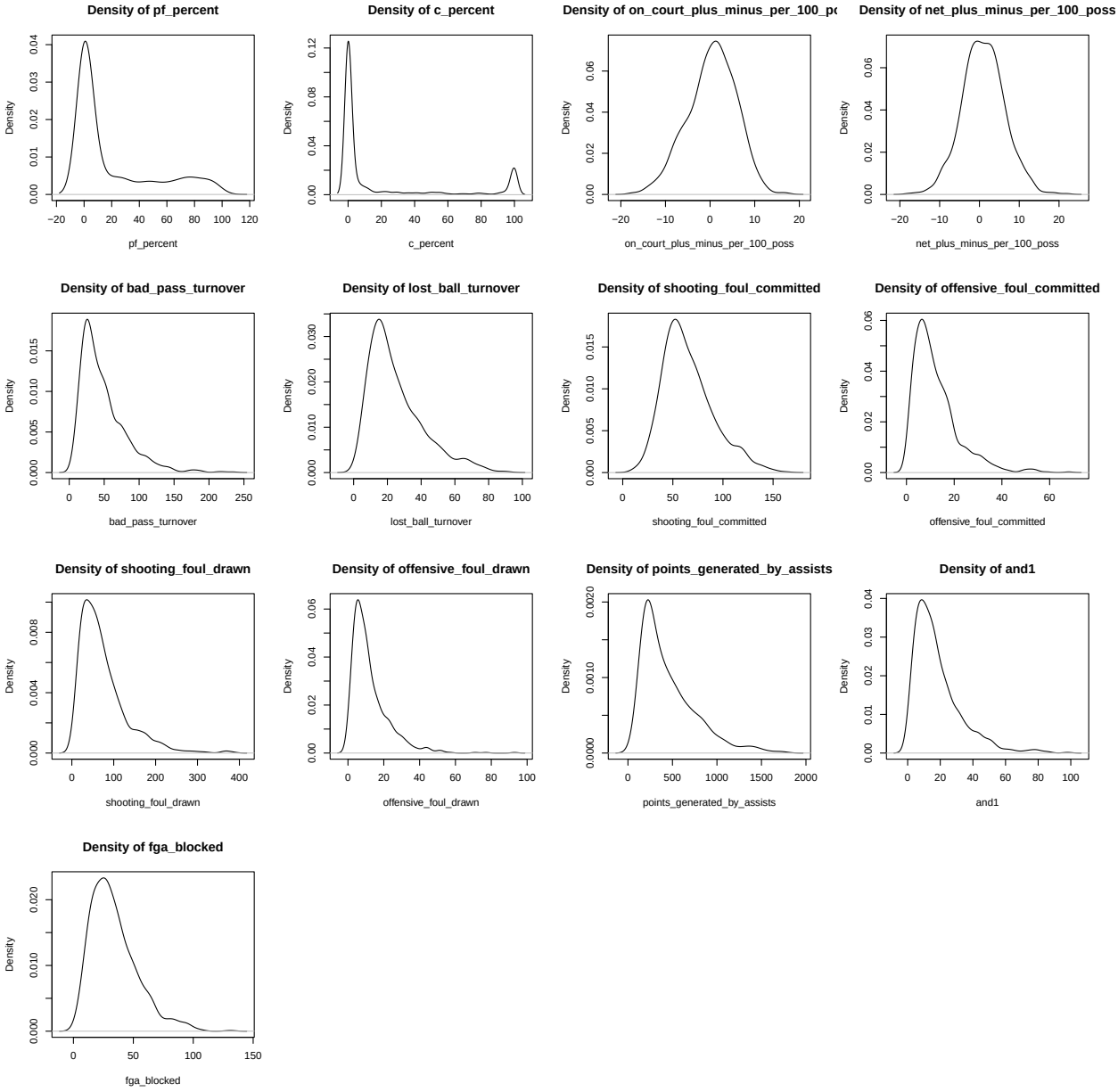
Univariate Plots

```r
# Selecting numeric columns for plotting
numeric_cols <- select_if(player_season_stats, is.numeric)

# Creating boxplots
boxplot(numeric_cols,
        main = "Boxplots of Numeric Variables",
        las = 2, # Rotate axis labels
        cex.axis = 0.7,
        col = rainbow(ncol(numeric_cols)))
```




```
# Creating density plots for each numeric variable
par(mfrow = c(2, 4)) # Adjust the grid size based on the number of numeric variables
for(col_name in names(numeric_cols)) {
  plot(density(na.omit(numeric_cols[[col_name]])),
       main = paste("Density of", col_name),
       xlab = col_name)
}
```





Exploratory Data Analysis Findings

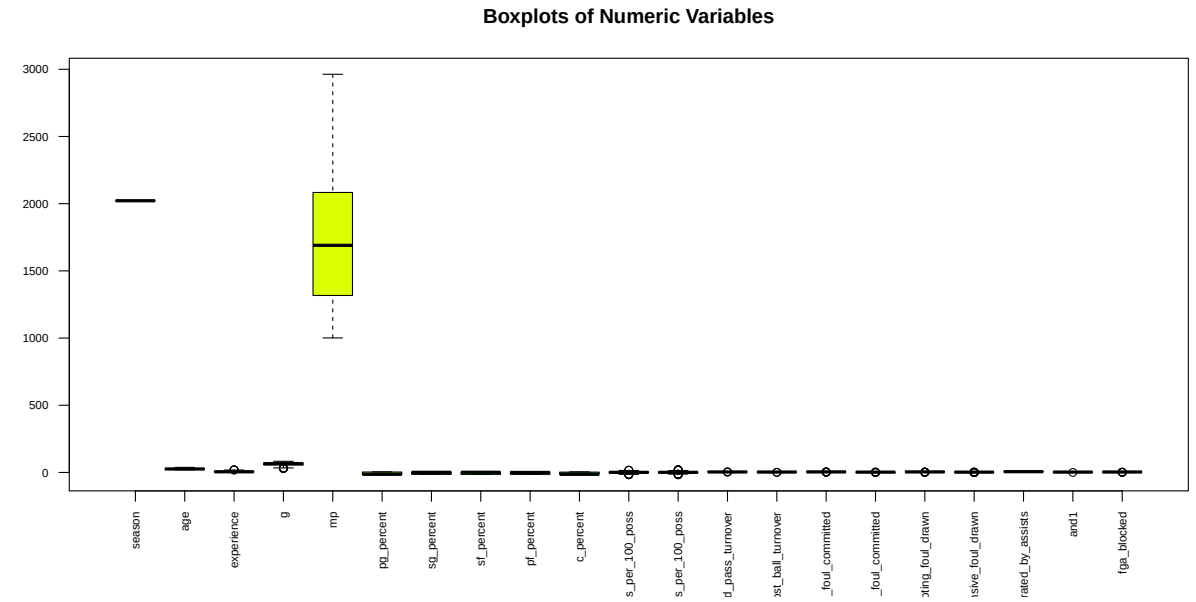
Univariate Distribution Observations

- **Season**: Peaks in the distribution correspond to individual seasons.
- **Age & Experience**: Both age and experience distributions are right-skewed, indicating a younger and less experienced player cohort.
- **Game Metrics**: Skewed distributions with long tails for various metrics reflect the diversity of player performance.
- **Turnovers**: Right-skewed distributions suggest turnovers are infrequent for most players, but high for some.
- **Positional Play Percentages**: Most players have low percentages, indicating specialization in fewer positions.
- **Foul-related Metrics**: Right-skewed foul distributions suggest most players commit few fouls.

- **Performance Metrics**: Plus-minus metrics show outliers, indicating exceptional performances.
- **Assists**: The distribution is right-skewed, with most players generating fewer points through assists.
- **Drawn Fouls**: Densities are concentrated near zero, indicating most players draw few fouls.
- **Blocked Attempts**: The majority of players have low block counts, with the distribution being right-skewed.
- **And-1 Plays**: Such plays are rare events for most players.

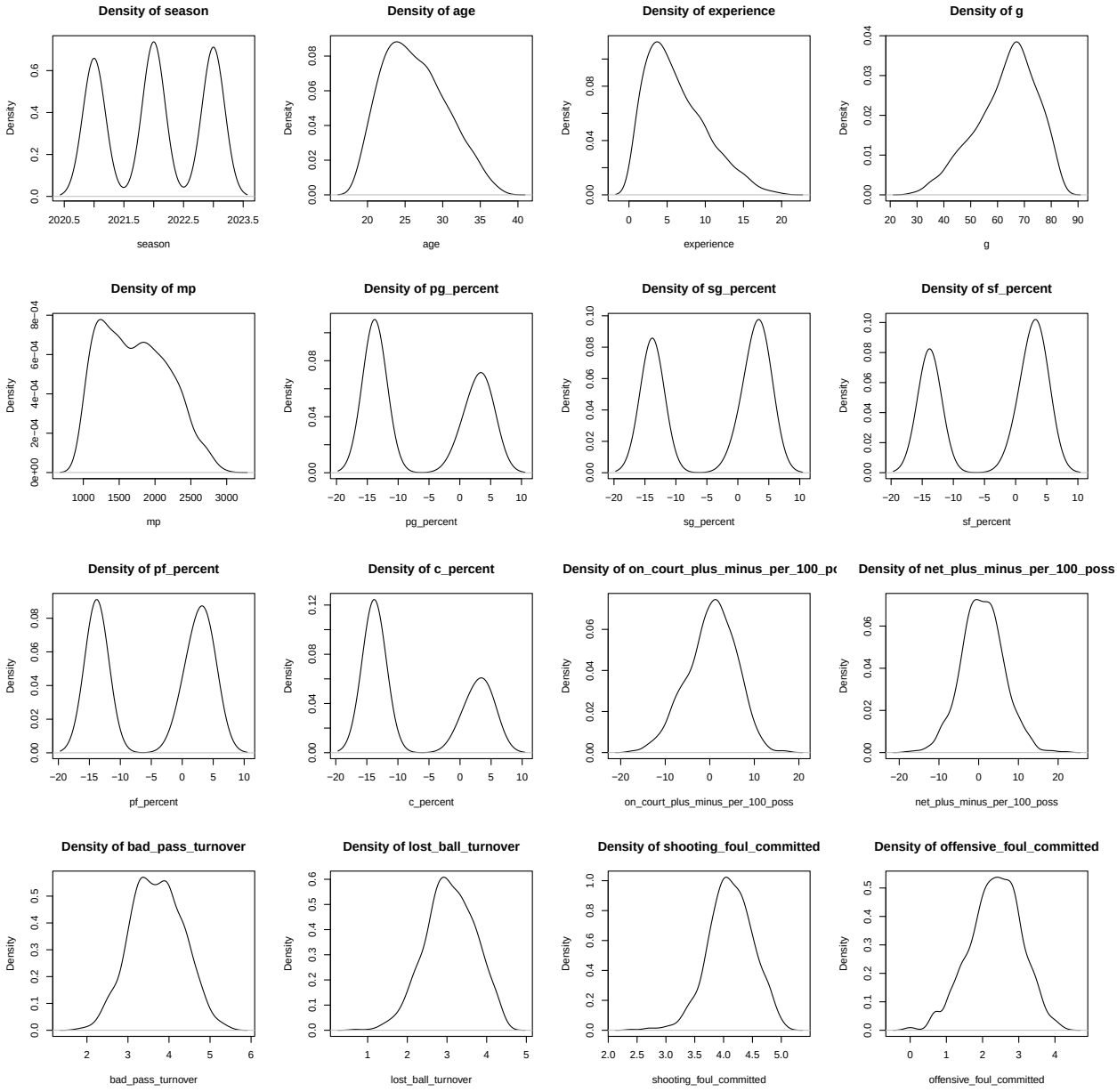
```
# Log transformation for cumulative fields with a shift for zero values
player_season_stats <- player_season_stats %>%
  mutate(across(c(bad_pass_turnover, lost_ball_turnover, shooting_foul_committed,
                  offensive_foul_committed, shooting_foul_drawn, offensive_foul_drawn,
                  points_generated_by_assists, and1, fga_blocked),
                ~log1p(.)))

# Log transformation for percentages (assuming no zero values, if there are, consider adding a small co
player_season_stats <- player_season_stats %>%
  mutate(across(c(pg_percent, sg_percent, sf_percent, pf_percent, c_percent),
                ~log(. + 1e-6)))  # Adding a small constant to avoid log(0)
```

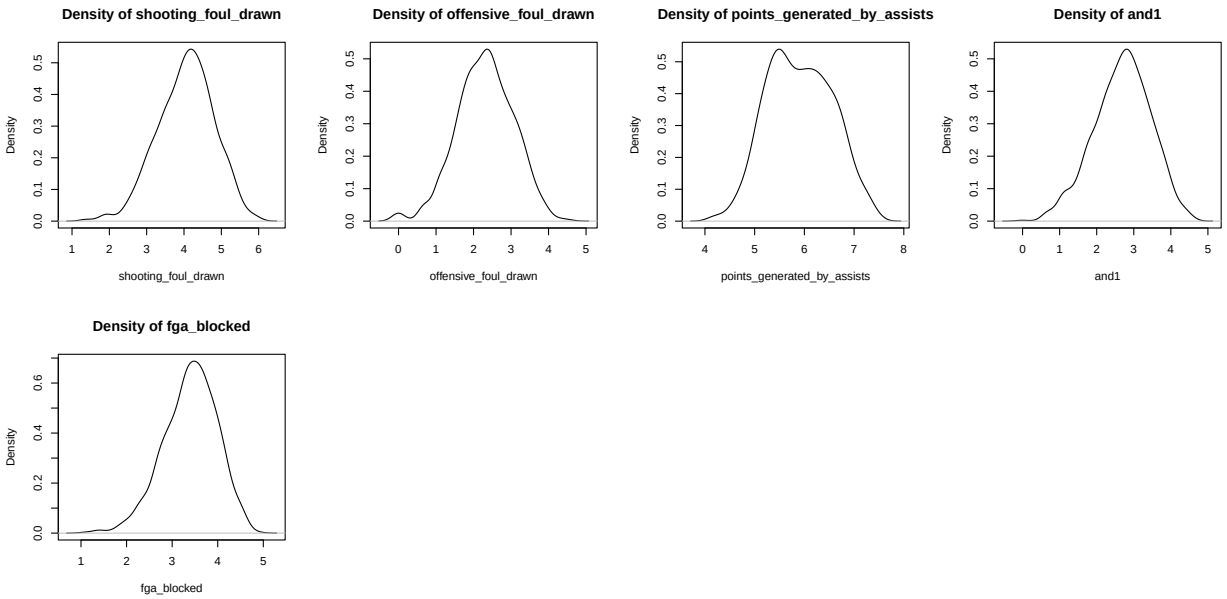
```
# Selecting numeric columns for plotting
numeric_cols <- select_if(player_season_stats, is.numeric)

# Creating boxplots
boxplot(numeric_cols,
        main = "Boxplots of Numeric Variables",
        las = 2, # Rotate axis labels
        cex.axis = 0.7,
        col = rainbow(ncol(numeric_cols)))
```




```
# Creating density plots for each numeric variable
par(mfrow = c(2, 4)) # Adjust the grid size based on the number of numeric variables
for(col_name in names(numeric_cols)) {
  plot(density(na.omit(numeric_cols[[col_name]])),
```

```
        main = paste("Density of", col_name),
        xlab = col_name)
}
```





Post-transformation

Log transformation successfully moderated the skewness across most variables, resulting in more symmetric, bell-shaped distributions indicative of normality. This adjustment is particularly evident in `bad_pass_turnover`, `lost_ball_turnover`, `shooting_foul_committed`, and `offensive_foul_committed`, where the long tails have been considerably reduced. Nevertheless, the change in `on_court_plus_minus_per_100_poss` and `net_plus_minus_per_100_poss` was less marked, suggesting that these variables may require alternative approaches to normalization.

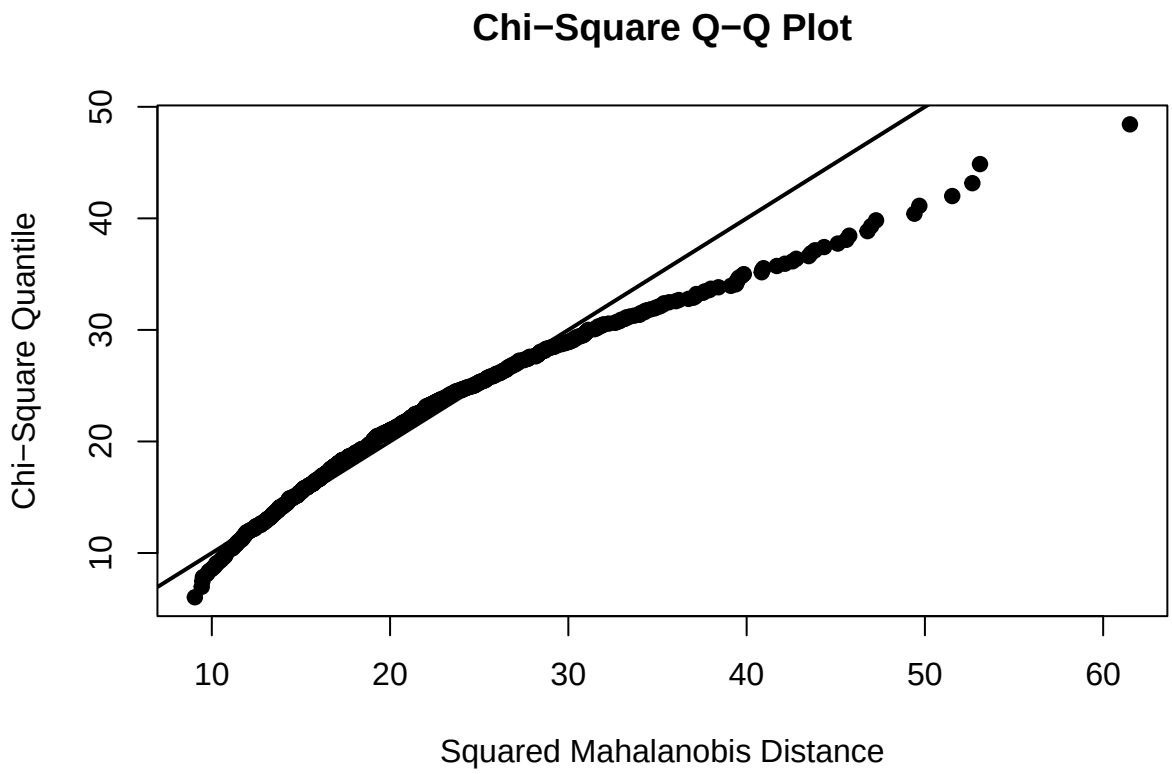
Conclusion

The applied log transformation is largely effective, moving the distributions closer to normality. This enhances the dataset's compatibility with parametric methods, which typically assume normally distributed variables. Further exploration may include additional normality testing and alternative data transformations as needed.

Multivariate Normality Check

To assess multivariate normality, a chi-square quantile plot is employed, which compares the distribution of the dataset with a chi-square distribution. Deviations from the chi-square line would suggest a departure from multivariate normality.

```
# Computing the chi-square quantile plot
# Select only numeric columns from the dataset
numeric_data <- player_season_stats %>%
  select_if(is.numeric)

# Now run the multivariate normality test using the 'numeric_data' dataframe
mvnData <- mvn(numeric_data, mvnTest = "hz", univariatePlot = "qq", multivariatePlot = "qq")
```



Chi-Square Q-Q Plot Analysis

The Chi-Square Q-Q plot is a graphical representation to assess the multivariate normality of our NBA dataset. Points lying along the reference line suggest a distribution is consistent with multivariate normality.

Observations:

- The plot indicates that many of the data points follow the reference line closely at the beginning, suggesting a good fit to the normal distribution for a majority of the data.
- However, as we move towards the tail (higher Mahalanobis distances), we observe a noticeable deviation from the line, with points splaying upwards. This pattern indicates that there are outliers present in the dataset or variables that exhibit heavy tails compared to a normal distribution.

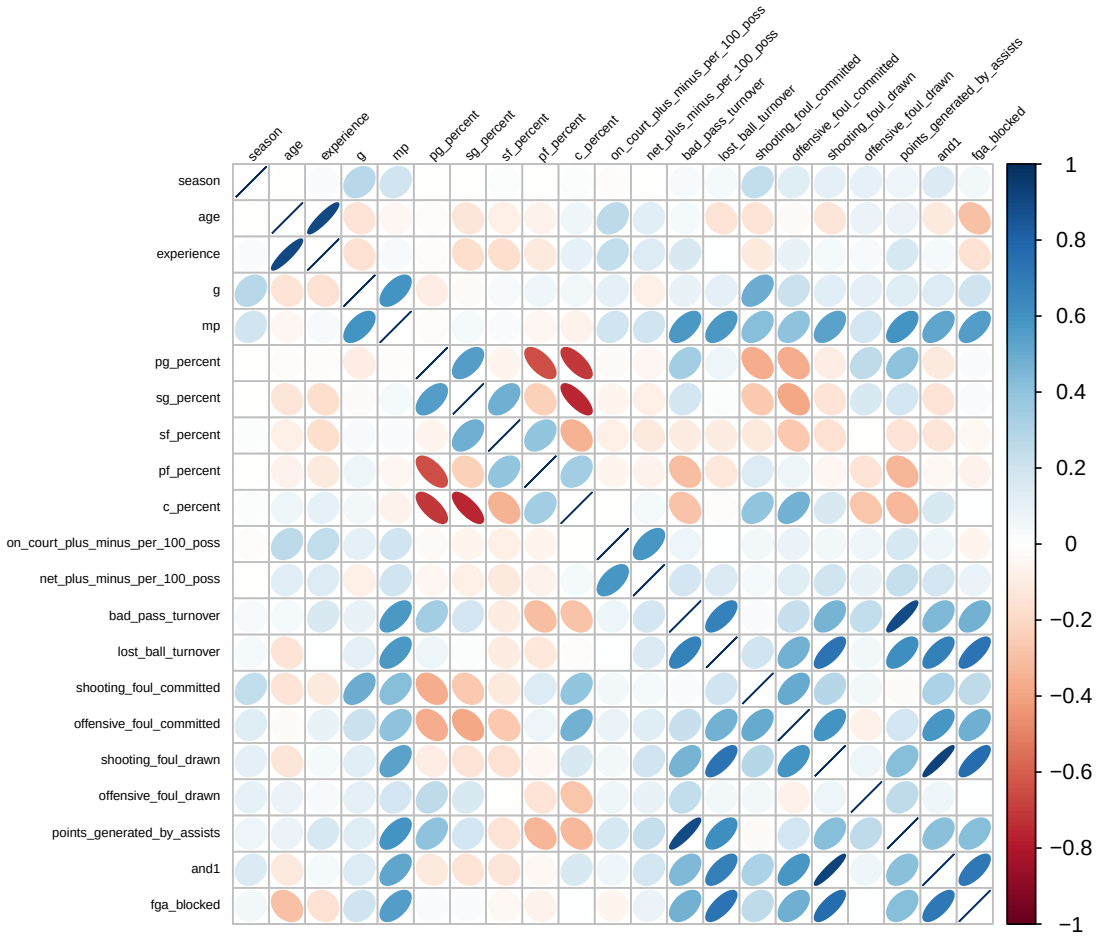
Implications for PCA:

- While PCA does not strictly require multivariate normality, significant deviations from normality can affect the interpretation of the principal components.
- The presence of outliers or heavy-tailed distributions may influence the calculation of principal components, potentially overstating the importance of outlier-related variance.

Correlation Analysis

A correlation matrix is a crucial step in understanding the inter-relationships between variables before proceeding with PCA. It helps in identifying variables that are highly correlated, which might be combined into a single principal component.

```
# Compute and visualize the correlation matrix

# Set larger plotting area
options(repr.plot.width=15, repr.plot.height=15)

# Compute the correlation matrix
cor_matrix <- cor(numeric_data, use = "complete.obs")

# Visualize the correlation matrix with adjustments
corrplot(cor_matrix, method = "ellipse", tl.cex = 0.5, tl.col = "black", tl.srt = 45)
```




```
# Reset to default plotting area (if needed)
options(repr.plot.width=7, repr.plot.height=7)
```

Correlation Matrix Analysis

The correlation matrix provides a visual representation of the relationship between pairs of variables in the NBA player statistics dataset.

Observations from the Correlation Plot:

- There are several variables with strong positive correlations, indicated by the dark blue ellipses, such as between `bad_pass_turnover` and `lost_ball_turnover`. This suggests that players who tend to make bad passes also tend to lose the ball more frequently.
- Some variables exhibit strong negative correlations, as shown by the dark red ellipses. For example, `on_court_plus_minus_per_100_poss` and `net_plus_minus_per_100_poss` are negatively correlated with turnover-related metrics, indicating that higher turnovers are associated with a less positive impact on the court.
- The positional play percentages (like `pg_percent`, `sg_percent`, etc.) show some moderate positive correlations with each other, which could suggest that players who spend more time in one position might also cover other positions to some extent.
- Interestingly, there are some variables with very little to no correlation, such as `age` and `shooting_foul_committed`. This lack of relationship could indicate that age is not a significant factor in the tendency to commit shooting fouls.

PCA Suitability:

- The presence of both strong positive and negative correlations suggests that PCA could be a suitable method for dimensionality reduction for this dataset.
- Variables with high correlations can potentially be combined into principal components, which would simplify the dataset while retaining most of the variability.
- The lack of correlation between some variables should be noted as it means that PCA may not reduce dimensionality in these areas without loss of information.

Concluding Remarks:

- The correlation analysis has uncovered actionable insights into the interplay between various player statistics.
- The suitability of PCA is reaffirmed by the identified correlations, and it stands to reason that PCA can help in summarizing the data into principal components that capture the essence of the players' performance metrics.

Principal Component Analysis (PCA)

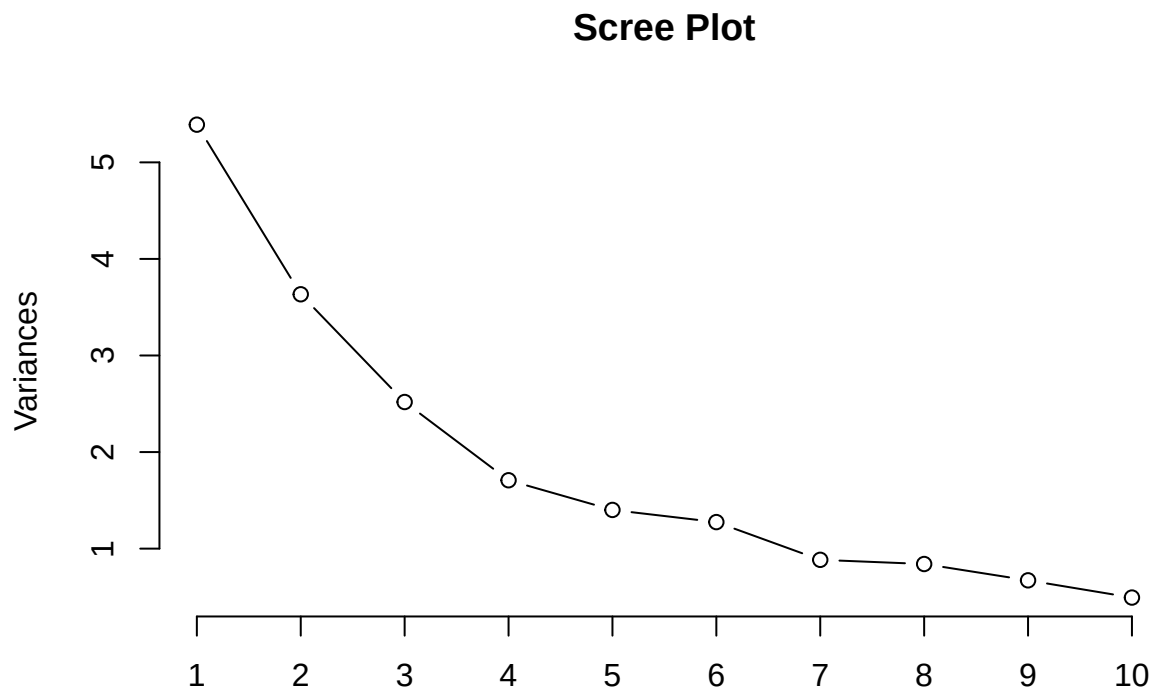
PCA is performed on the standardized variables using the correlation matrix to identify the principal components that explain the most variance within the dataset.

```
# Perform PCA using the correlation matrix of the standardized variables
pca_result <- prcomp(numeric_data, scale = TRUE)

# Summary of PCA results showing the importance of each principal component
summary(pca_result)
```

```
## Importance of components:
```

```
##                            PC1    PC2    PC3     PC4    PC5    PC6    PC7
## Standard deviation     2.3217 1.9062 1.5869 1.30691 1.1835 1.1291 0.9403
## Proportion of Variance 0.2567 0.1730 0.1199 0.08133 0.0667 0.0607 0.0421
## Cumulative Proportion  0.2567 0.4297 0.5496 0.63097 0.6977 0.7584 0.8005
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.91693 0.81954 0.70242 0.60631 0.55335 0.52519 0.49754
## Proportion of Variance 0.04004 0.03198 0.02349 0.01751 0.01458 0.01313 0.01179
## Cumulative Proportion  0.84051 0.87250 0.89599 0.91350 0.92808 0.94121 0.95300
##                           PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     0.47020 0.44689 0.43149 0.39671 0.30804 0.26099 0.24421
## Proportion of Variance 0.01053 0.00951 0.00887 0.00749 0.00452 0.00324 0.00284
## Cumulative Proportion  0.96353 0.97304 0.98190 0.98940 0.99392 0.99716 1.00000
```

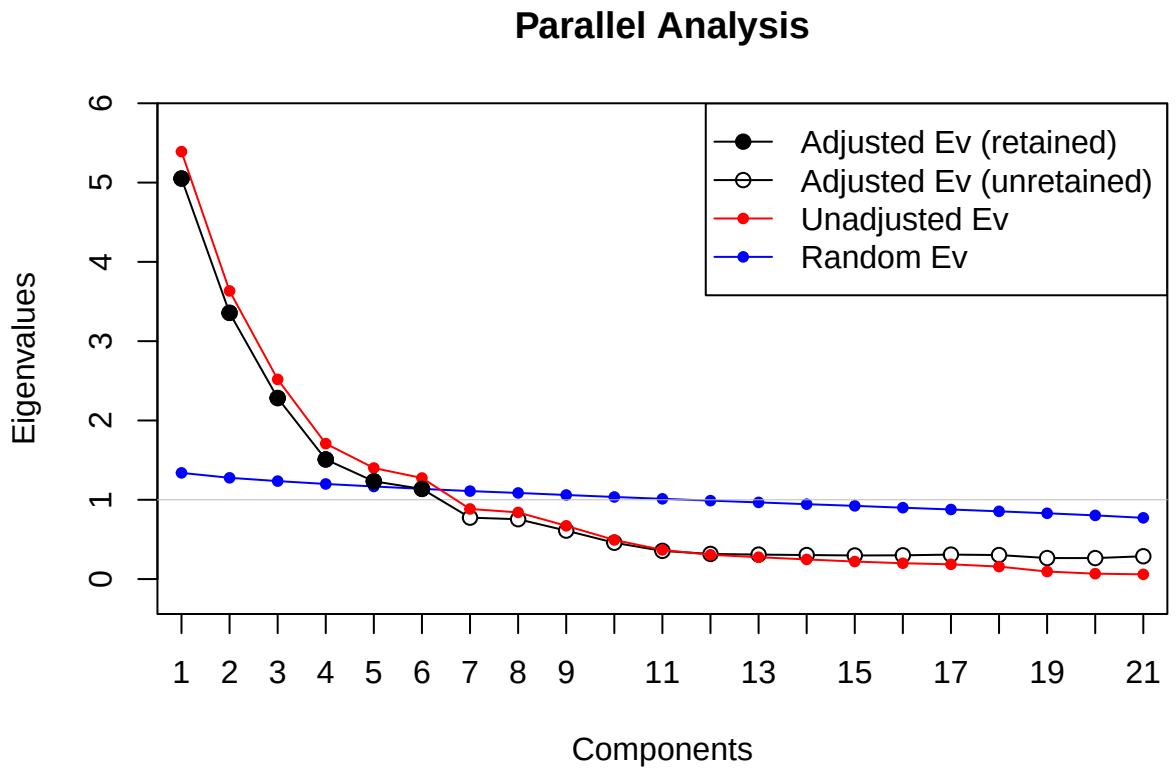
```r
# Scree plot to visualize the variance explained by each principal component
plot(pca_result, type = "lines", main = "Scree Plot")
```




```r
# Perform parallel analysis to help decide the number of principal components to retain
paran(numeric_data, iterations = 1000, centile = 95, status = TRUE, graph = TRUE)
```

```
##
## Using eigendecomposition of correlation matrix.
## Computing: 10%  20%  30%  40%  50%  60%  70%  80%  90%  100%
##
##
## Results of Horn's Parallel Analysis for component retention
```

```
## 1000 iterations, using the 95 centile estimate
##
## ----------------------------------------------------
## Component   Adjusted    Unadjusted    Estimated
##             Eigenvalue  Eigenvalue    Bias
## ----------------------------------------------------
## 1           5.050995    5.390313      0.339317
## 2           3.356873    3.633788      0.276915
## 3           2.282744    2.518253      0.235508
## 4           1.508792    1.708011      0.199218
## 5           1.232912    1.400716      0.167803
## 6           1.136953    1.274760      0.137807
## ----------------------------------------------------
##
## Adjusted eigenvalues > 1 indicate dimensions to retain.
## (6 components retained)
```



PCA Results Interpretation

The PCA analysis provides insights into the data's underlying structure by identifying the principal components that explain the most variance.

Scree Plot Observations:

- The scree plot suggests an elbow around the fourth component, implying that the first four components are the most significant in terms of variance explanation.
- While the first two components show a steeper drop, the third and fourth components continue to contribute meaningfully to the variance.

Parallel Analysis Observations:

- The parallel analysis indicates that the actual data eigenvalues remain above the random eigenvalues until somewhere between the sixth and seventh components, suggesting that up to six components may be statistically significant.

Eigenvalues and Variance Explained:

- The first component (PC1) explains a substantial 25.67% of the variance, and subsequent components (PC2 through PC4) continue to add non-negligible amounts, leading to a cumulative variance of over 63% explained by the first four components.
- The eigenvalues greater than 1 criterion supports retaining up to the seventh component. However, based on the scree plot and parallel analysis, a more conservative approach might involve focusing on the first four to six components.

Decision on Components to Retain:

- Considering the scree plot and parallel analysis, retaining the first four principal components would capture the most significant variance while still providing a manageable number of components for interpretation.
- If a broader variance capture is desired, extending to six components is justified by the parallel analysis, which would encompass the majority of significant variance as indicated by the eigenvalues compared to the random data.

Conclusion:

- The PCA has effectively distilled the NBA player statistics dataset into a smaller set of components that still capture a significant portion of the information.
- This dimensionality reduction facilitates a clearer understanding of the data's structure and can aid in further analysis, such as predictive modeling or exploratory visualization.

Examination and Interpretation of Principal Component Loadings

Now that we have identified the number of principal components to retain, we will examine the loadings for each of these components to interpret their significance.

```
# Extract the loadings for the principal components
loadings <- pca_result$rotation[, 1:6]  # Assuming we retain six components based on previous analysis

# View the loadings matrix
print(loadings)
```

```
##                                          PC1          PC2         PC3
## season                           -0.0833164340  0.033657124 -0.05268130
## age                               0.0449332274 -0.016338741  0.52588058
## experience                       -0.0256105521 -0.006979832  0.53323299
## g                                -0.1509482279  0.076327908 -0.17447225
## mp                               -0.3387974811 -0.036643824 -0.05106887
## pg_percent                       -0.0005964565 -0.454010412  0.00980973
## sg_percent                        0.0413032773 -0.393972780 -0.20839409
## sf_percent                        0.0890344079 -0.093938292 -0.27533297
## pf_percent                        0.0639174508  0.295170057 -0.15254018
## c_percent                        -0.0349845441  0.462142580  0.11531747
## on_court_plus_minus_per_100_poss -0.0688659999 -0.007820209  0.31347614
## net_plus_minus_per_100_poss      -0.1171598691 -0.008304034  0.27122193
## bad_pass_turnover                -0.3057227970 -0.251699279  0.08392872
## lost_ball_turnover               -0.3622936404 -0.060999873 -0.04746654
## shooting_foul_committed          -0.1843901549  0.262926973 -0.11269197
## offensive_foul_committed         -0.2826921064  0.258095026  0.04473434
## shooting_foul_drawn              -0.3717150042  0.065072535 -0.02996037
## offensive_foul_drawn             -0.0700909624 -0.180816570  0.04475262
## points_generated_by_assists      -0.2954115609 -0.274084851  0.12556656
## and1                             -0.3628419079  0.073959902 -0.02730562
## fga_blocked                      -0.3446127092 -0.006954921 -0.17930347
##                                          PC4          PC5          PC6
## season                           -0.31323878  0.272829894  0.198165909
## age                              -0.15036720 -0.082951474  0.377269150
## experience                       -0.06253848 -0.079473835  0.393449411
## g                                -0.49712583  0.261456330  0.012508841
## mp                               -0.31568215 -0.008424539  0.042076786
## pg_percent                        0.08354946  0.236506329 -0.053362311
## sg_percent                       -0.13607182 -0.180398732  0.047601311
## sf_percent                       -0.28261308 -0.539660290  0.224352353
## pf_percent                       -0.17736641 -0.450315112  0.160512075
## c_percent                         0.10953281  0.088173538 -0.005168268
## on_court_plus_minus_per_100_poss -0.31018954 -0.202320294 -0.489422123
## net_plus_minus_per_100_poss      -0.12668669 -0.265843796 -0.564267010
## bad_pass_turnover                 0.03716614 -0.026288559  0.096273857
## lost_ball_turnover                0.17672001 -0.118987010  0.067744647
## shooting_foul_committed          -0.29103313  0.238231916 -0.040159462
## offensive_foul_committed          0.06154732  0.052265104  0.036283589
## shooting_foul_drawn               0.17865678 -0.123478452  0.037009858
## offensive_foul_drawn             -0.25708220  0.166519191  0.011291231
## points_generated_by_assists      -0.01229044  0.006270515  0.035036006
## and1                              0.14213884 -0.111874693  0.051924211
## fga_blocked                       0.16738461 -0.110424791  0.017065874
```

```r
# Interpretation of each component based on the loadings
# Usually, loadings with an absolute value > 0.3 are considered significant
interpret_loadings <- function(loadings_matrix, cutoff = 0.3) {
  interpretations <- apply(loadings_matrix, 2, function(component) {
    significant_loadings <- which(abs(component) > cutoff)
    names(significant_loadings) <- names(component)[significant_loadings]
    return(significant_loadings)
  })
  return(interpretations)
```

```
}

# Get interpretations for each component
component_interpretations <- interpret_loadings(loadings)
print(component_interpretations)
```

```
## $PC1
##                   mp   bad_pass_turnover  lost_ball_turnover shooting_foul_drawn
##                    5                   13                  14                  17
##                 and1         fga_blocked
##                   20                   21
##
## $PC2
## pg_percent sg_percent  c_percent
##          6          7         10
##
## $PC3
##                               age                        experience
##                                 2                                 3
## on_court_plus_minus_per_100_poss
##                                11
##
## $PC4
##                             season                                 g
##                                  1                                 4
##                                mp on_court_plus_minus_per_100_poss
##                                 5                                11
##
## $PC5
## sf_percent pf_percent
##          8          9
##
## $PC6
##                               age                        experience
##                                 2                                 3
## on_court_plus_minus_per_100_poss      net_plus_minus_per_100_poss
##                                11                                12
```

Principal Component Loadings Interpretation

The loadings of the principal components provide insights into which variables contribute most to each component in the PCA.

PC1 Loadings:

- PC1 characterizes players with significant offensive engagement, reflected by loadings on `bad_pass_turnover`, `lost_ball_turnover`, `shooting_foul_drawn`, `and1`, and `fga_blocked`. This factor distinguishes high-usage players—typically ball handlers and slashers—who actively create plays, often resulting in turnovers and drawing fouls while also being prone to having their shots blocked. It highlights a profile of players who are central to offensive action and decision-making on the court.

PC2 Loadings:

- Principal component two (PC2) has high loadings for position-related percentages such as `pg_percent`, `sg_percent`, and `c_percent`. It seems to capture elements related to the players' positions on the court, possibly indicating a 'positional play style' factor.

PC3 Loadings:

- PC3 is significantly influenced by `age`, `experience`, and `on_court_plus_minus_per_100_poss`, which might be reflecting an 'experience and impact' factor, combining the maturity and on-court effectiveness of players.

PC4 Loadings:

- The fourth principal component (PC4) includes `season`, `g` (likely games played), and `mp` (minutes played), along with `on_court_plus_minus_per_100_poss`. This component may represent a 'seasonal activity and performance' factor, relating to how active players were in a season and their overall contribution to games.

PC5 Loadings:

- PC5 is predominantly associated with `sf_percent` and `pf_percent`, which could indicate a 'forward positions' factor, representing the roles of small forwards and power forwards.

PC6 Loadings:

- Lastly, PC6 ties back to `age`, `experience`, and the plus-minus metrics again (`on_court_plus_minus_per_100_poss` and `net_plus_minus_per_100_poss`). This suggests an 'age and effectiveness' factor where older, more experienced players' contributions to the team's performance are captured.

Conclusion:

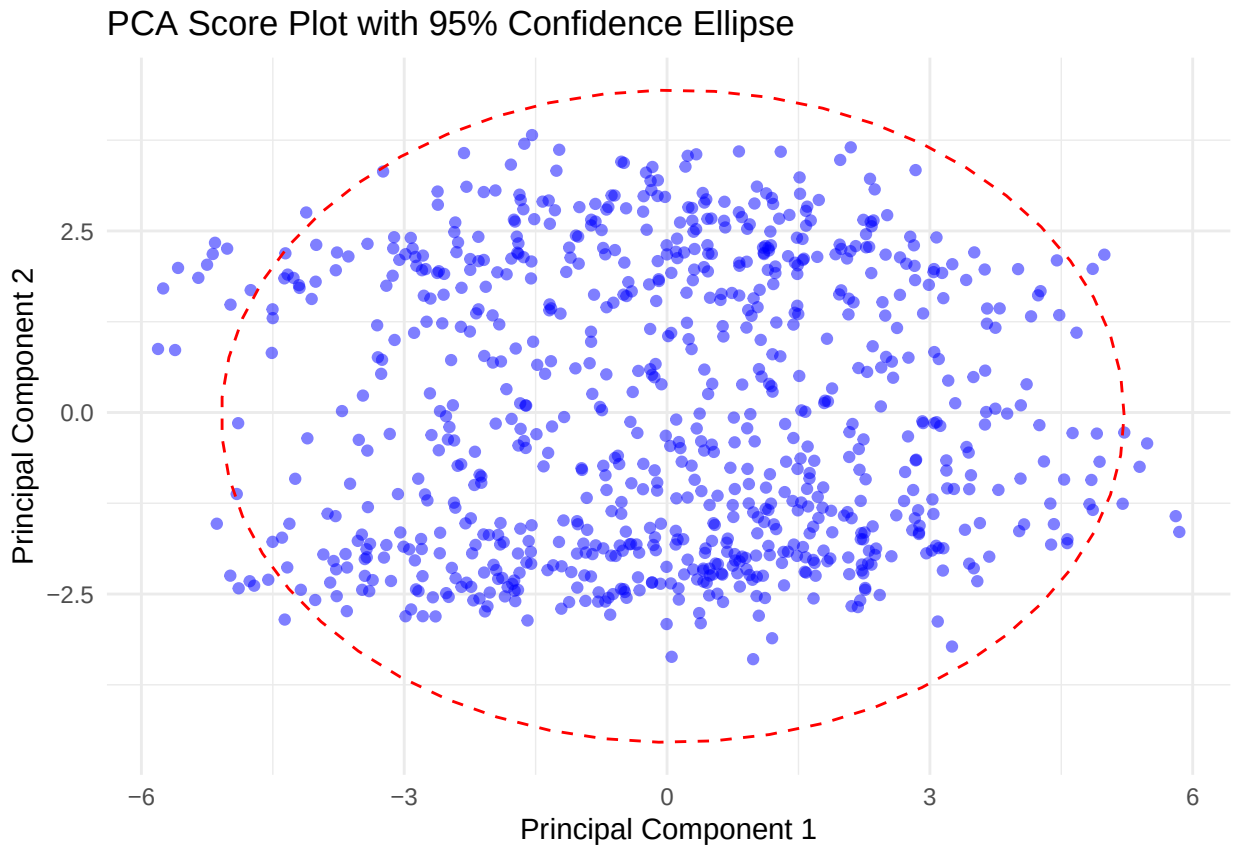
- The principal components extracted seem to encapsulate various aspects of player performance, from in-game actions to positions and roles, as well as experience levels and their impact on the game.
- These components can serve as new variables for further analysis, where they may be used to cluster players into different profiles or predict outcomes.

Score Plot of Principal Component Scores

Creating a score plot allows us to visualize the data in the reduced-dimensional space formed by the principal components. It can help identify any trends or groupings among the observations.

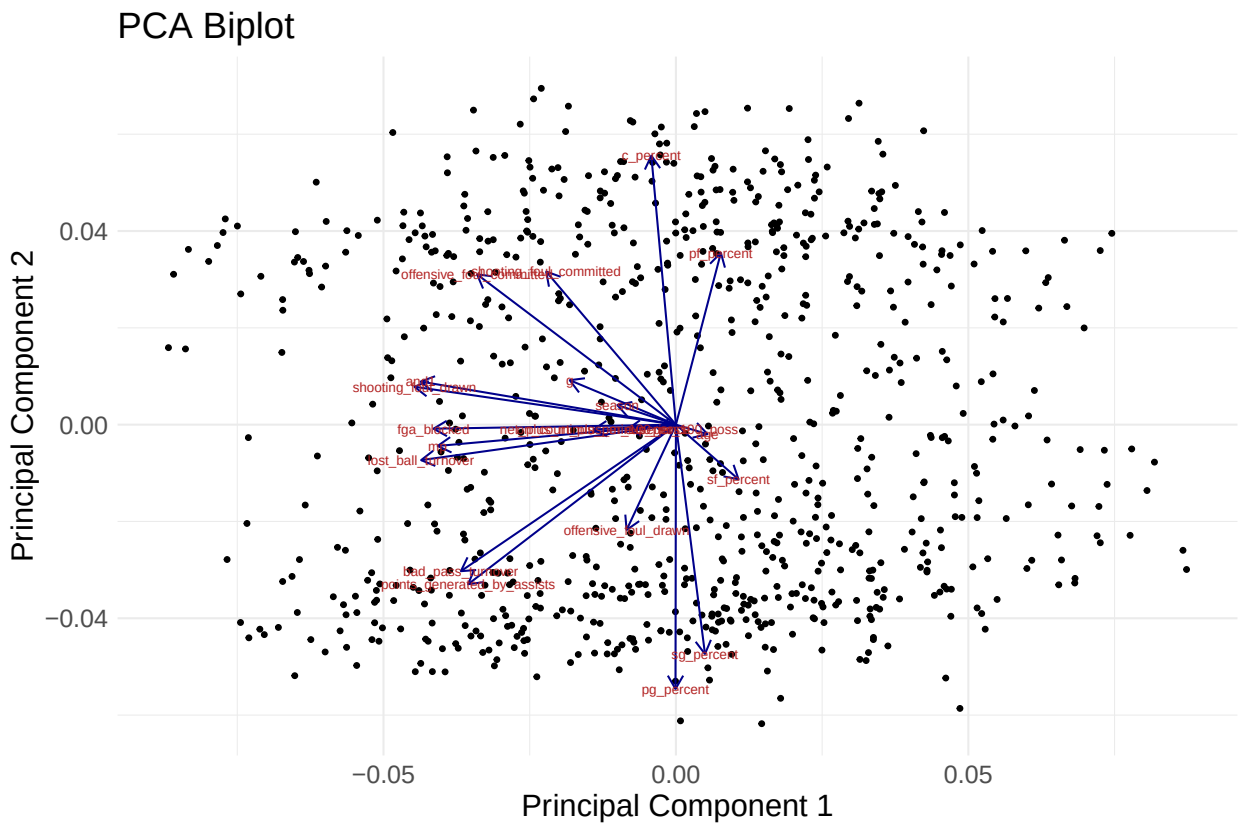
```
library(ggplot2)
library(ggforce) # for geom_ellipse

# Convert PCA scores to a data frame
scores_df <- as.data.frame(pca_result$x)

# Creating the score plot for the first two principal components
```

```r
ggplot(scores_df, aes(x = PC1, y = PC2)) +
  geom_point(color = 'blue', alpha = 0.5) +  # Points
  stat_ellipse(type = "t", level = 0.95, linetype = "dashed", color = "red", size = 0.5) +  # 95% Confi
  theme_minimal() +
  labs(title = "PCA Score Plot with 95% Confidence Ellipse",
       x = "Principal Component 1",
       y = "Principal Component 2")
```



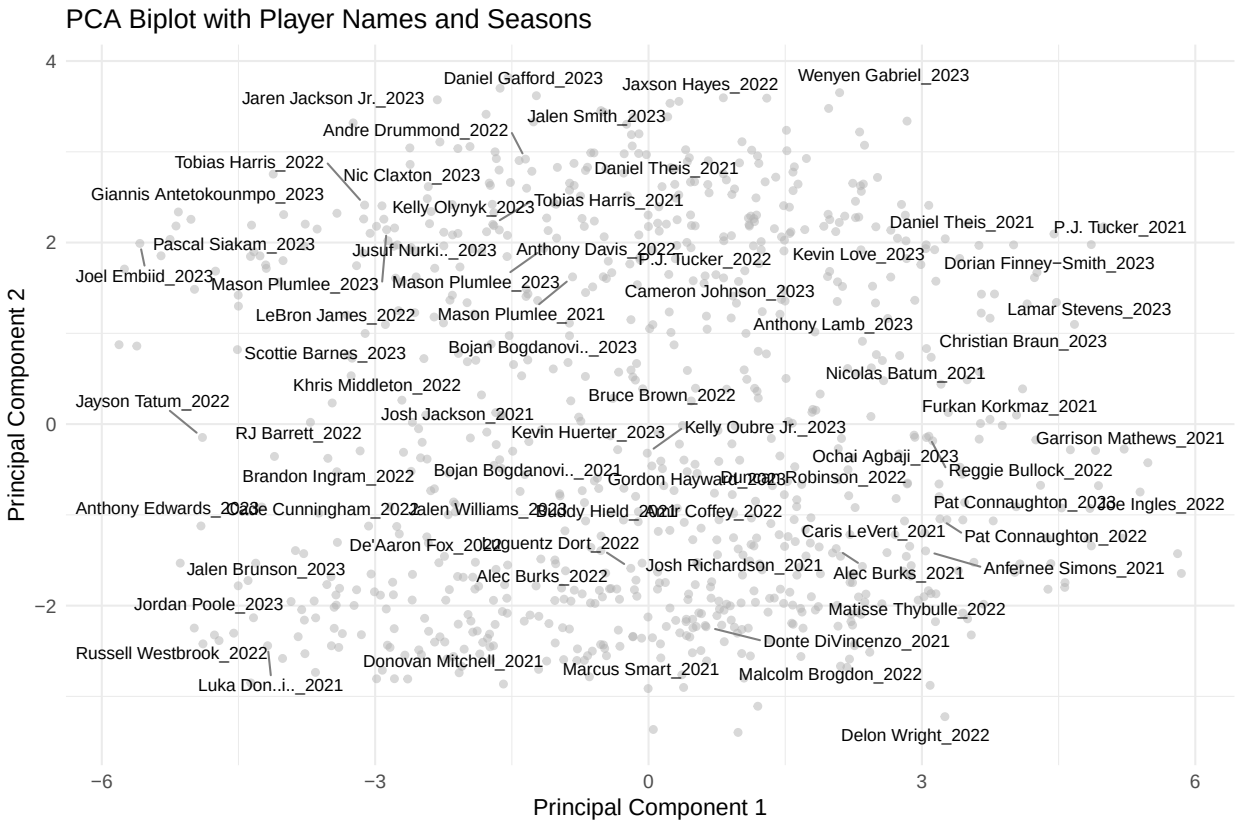

```r
library(ggfortify)
library(ggplot2)

# Define color choices for arrows and labels
arrow_color <- 'darkblue'  # Choose a color for arrows
label_color <- 'firebrick'  # Choose a color for labels

# Create the biplot with adjusted colors and sizes
biplot <- autoplot(pca_result, data = scores_df, label = FALSE,  # Turn off individual point labels
                   loadings = TRUE, loadings.label = TRUE, loadings.label.size = 2.5,
                   loadings.colour = arrow_color, loadings.label.colour = label_color,
                   size = 1) +  # Adjust the size of points
  labs(title = "PCA Biplot",
       x = "Principal Component 1",
       y = "Principal Component 2") +
  theme_minimal() +
  theme(legend.position = "none",  # Remove the legend
```

```
        text = element_text(size = 16))  # Adjust text size

# Display the plot
print(biplot)
```




```
library(ggfortify)
library(ggplot2)
library(ggrepel)
library(dplyr)

# Create a combined identifier for player and season
player_season_stats$identifier <- paste(player_season_stats$player, player_season_stats$season, sep = "

# Bind the identifier column to the PCA scores data frame
scores_df <- bind_cols(player_season_stats$identifier, pca_result$x)
colnames(scores_df)[1] <- "identifier"  # Make sure the identifier column is properly named

# Define the number of points to display
set.seed(123)  # Set a random seed for reproducibility
n_display <- 100  # Number of points to display

# Randomly sample identifiers to display labels
selected_labels <- sample(scores_df$identifier, n_display, replace = FALSE)

# Filter the scores_df to only include the selected labels
selected_scores_df <- scores_df %>%
```

```
  filter(identifier %in% selected_labels)

# Create the biplot with randomly selected player names and seasons
biplot <- ggplot() +
  geom_point(data = scores_df, aes_string(x = "PC1", y = "PC2"), color = 'grey70', alpha = 0.5) + # All
  geom_text_repel(data = selected_scores_df, aes_string(label = "identifier", x = "PC1", y = "PC2"),
                  size = 3, box.padding = 0.35, point.padding = 0.5,
                  segment.color = 'grey50') +
  labs(title = "PCA Biplot with Player Names and Seasons", x = "Principal Component 1", y = "Principal C
  theme_minimal() +
  theme(legend.position = "none", text = element_text(size = 12))

# Display the plot
print(biplot)
```



Conclusion

The application of Principal Component Analysis (PCA) on the NBA player statistics dataset has been illuminating. The analysis successfully reduced the complex, multi-dimensional space into a lower-dimensional representation that captured significant aspects of player performance. In the context of the PC1 and PC2 plot, the PCA method effectively distinguished players based on their roles and playing styles.

Star guards and primary ball handlers, exemplified by players like Luka Dončić and Russell Westbrook, clustered towards the lower-left corner of the plot. This area characterizes players with high offensive engagement and ball-handling responsibilities. Conversely, dominant big men such as Joel Embiid and

Giannis Antetokounmpo occupied the upper-left corner, representing their significant presence in the interior game and scoring efficiency.

The 3-and-D specialists known for their defensive prowess and perimeter shooting, like P.J. Tucker and Dorian Finney-Smith, were found in the upper-right corner, indicating a different set of skills impacting the game through defense and efficient shooting.

The PCA provided a powerful means to distill the essence of the data, preserving the relationships and patterns that are most informative of the players' roles and contributions on the court. It underscored the potential of using advanced statistical techniques to capture the intricacies of sports performance, offering a snapshot that correlates well with the recognized playing styles and positions in basketball.

In summary, the PCA on this NBA dataset was not only a technical success in terms of data reduction but also delivered meaningful insights into the nature of the game, reaffirming the value of PCA in sports analytics.