Final_Project_Markdown

Kameron V

2024-04-18

Dataset Initialization

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
data = read.csv("~/Downloads/NBA Stats.csv", sep = ";", header = FALSE)

colnames(data) = data[1,]
data = data[-1,]
head(data,10)
```

```
##    Rk                   Player  Pos Age  Tm  G GS   MP  FG  FGA   FG%  3P 3PA
## 2   1         Precious Achiuwa PF-C  24 TOT 59 18 23.1 3.5  6.9   0.5 0.4 1.5
## 3   1         Precious Achiuwa    C  24 TOR 25  0 17.5 3.1  6.8 0.459 0.5 1.9
## 4   1         Precious Achiuwa   PF  24 NYK 34 18 27.1 3.7  7.1 0.529 0.3 1.2
## 5   2              Bam Adebayo    C  26 MIA 56 56 34.5 7.6 14.9 0.511 0.1 0.3
## 6   3             Ochai Agbaji   SG  23 TOT 66 17 19.6 2.1  5.2 0.409 0.8 2.7
## 7   3             Ochai Agbaji   SG  23 UTA 51 10 19.7 2.1  4.9 0.426 0.9 2.8
## 8   3             Ochai Agbaji   SG  23 TOR 15  7 19.5 2.3  6.2 0.366 0.6 2.6
## 9   4             Santi Aldama   PF  23 MEM 54 28 25.8   4  9.3 0.431 1.6 4.9
## 10  5 Nickeil Alexander-Walker   SG  25 MIN 66 18 23.4 2.8  6.3 0.438 1.5 3.9
## 11  6            Grayson Allen   SG  28 PHO 61 61 33.7 4.4  8.7 0.504 2.7 5.7
##      3P%  2P  2PA   2P%  eFG%  FT FTA   FT% ORB DRB  TRB AST STL BLK TOV  PF
## 2   0.27 3.1  5.4 0.564 0.529   1 1.6 0.621 2.7   4  6.7 1.4 0.7   1 1.2 1.9
## 3  0.277 2.6  4.9 0.528 0.497   1 1.7 0.571   2 3.4  5.4 1.8 0.6 0.5 1.2 1.6
## 4  0.262 3.4  5.8 0.586 0.552   1 1.6  0.66 3.2 4.5  7.7 1.1 0.8 1.3 1.2 2.2
## 5  0.188 7.6 14.6 0.517 0.513 4.6   6 0.754 2.3 8.1 10.4 3.9 1.1   1 2.3 2.4
## 6  0.309 1.3  2.4 0.522 0.491 0.4 0.5 0.727 0.7 1.7  2.5 0.9 0.5 0.5 0.7 1.4
## 7  0.331 1.2  2.1 0.551  0.52 0.3 0.4  0.75 0.7 1.8  2.5 0.9 0.5 0.6 0.7 1.3
## 8  0.231 1.7  3.6 0.463 0.414 0.6 0.9 0.692 0.9 1.5  2.4 0.9 0.5 0.4 0.9 1.9
## 9   0.34 2.4  4.5 0.529 0.519 0.9 1.4 0.613 1.2 4.4  5.6 2.2 0.7 0.8 1.2 1.4
## 10 0.384 1.3  2.4 0.525 0.556 0.6 0.8 0.792 0.5 1.6    2 2.6 0.8 0.5   1 1.8
## 11 0.476 1.7    3 0.557 0.659 1.7 1.9 0.889 0.6 3.2  3.8   3 0.9 0.6 1.3 2.2
##     PTS
## 2   8.4
## 3   7.7
## 4   8.8
## 5  19.9
## 6   5.5
## 7   5.4
## 8   5.7
## 9  10.5
## 10  7.7
## 11 13.2
```

##Remove Duplicate Data

```
duplicates_idx = duplicated(data$Player)
data_new = data[!duplicates_idx,]
head(data_new,10)
```

```
##    Rk                   Player  Pos Age  Tm  G GS   MP   FG  FGA   FG%  3P 3PA
## 2   1         Precious Achiuwa PF-C  24 TOT 59 18 23.1  3.5  6.9   0.5 0.4 1.5
## 5   2              Bam Adebayo    C  26 MIA 56 56 34.5  7.6 14.9 0.511 0.1 0.3
## 6   3             Ochai Agbaji   SG  23 TOT 66 17 19.6  2.1  5.2 0.409 0.8 2.7
## 9   4             Santi Aldama   PF  23 MEM 54 28 25.8    4  9.3 0.431 1.6 4.9
## 10  5 Nickeil Alexander-Walker   SG  25 MIN 66 18 23.4  2.8  6.3 0.438 1.5 3.9
## 11  6            Grayson Allen   SG  28 PHO 61 61 33.7  4.4  8.7 0.504 2.7 5.7
## 12  7            Jarrett Allen    C  25 CLE 61 61 31.7  6.6 10.5 0.626   0 0.1
## 13  8            Jose Alvarado   PG  25 NOP 45  0 17.2  2.5    6 0.417 1.3 3.5
## 14  9            Kyle Anderson   PF  30 MIN 63  7 22.4  2.5  5.3 0.471 0.1 0.6
## 15 10    Giannis Antetokounmpo   PF  29 MIL 64 64   35 11.4 18.6 0.616 0.5 1.8
##      3P%   2P  2PA   2P%  eFG%  FT  FTA   FT% ORB DRB  TRB AST STL BLK TOV  PF
## 2   0.27  3.1  5.4 0.564 0.529   1  1.6 0.621 2.7   4  6.7 1.4 0.7   1 1.2 1.9
## 5  0.188  7.6 14.6 0.517 0.513 4.6    6 0.754 2.3 8.1 10.4 3.9 1.1   1 2.3 2.4
## 6  0.309  1.3  2.4 0.522 0.491 0.4  0.5 0.727 0.7 1.7  2.5 0.9 0.5 0.5 0.7 1.4
## 9   0.34  2.4  4.5 0.529 0.519 0.9  1.4 0.613 1.2 4.4  5.6 2.2 0.7 0.8 1.2 1.4
## 10 0.384  1.3  2.4 0.525 0.556 0.6  0.8 0.792 0.5 1.6    2 2.6 0.8 0.5   1 1.8
## 11 0.476  1.7    3 0.557 0.659 1.7  1.9 0.889 0.6 3.2  3.8   3 0.9 0.6 1.3 2.2
## 12     0  6.6 10.4 0.631 0.626 3.1  4.2 0.733 3.4 7.3 10.7 2.7 0.7 1.2 1.6 2.1
## 13 0.369  1.2  2.5 0.482 0.524 0.6  0.9  0.69 0.4 1.8  2.1 1.9 1.1 0.3 0.8 1.5
## 14 0.211  2.3  4.7 0.505 0.483 1.2  1.7   0.7 0.8 2.8  3.5 4.1 0.8 0.5 1.3 1.5
## 15 0.292 10.9 16.8  0.65  0.63 7.4 11.2 0.665 2.5 8.7 11.2 6.4 1.2   1 3.4 2.9
##     PTS
## 2   8.4
## 5  19.9
## 6   5.5
## 9  10.5
## 10  7.7
## 11 13.2
## 12 16.2
## 13    7
## 14  6.3
## 15 30.8
```

```
summary(data_new)
```

```
##       Rk               Player              Pos                Age           
##  Length:556         Length:556         Length:556         Length:556        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##       Tm                 G                  GS                 MP           
##  Length:556         Length:556         Length:556         Length:556        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##       FG                FGA                FG%                 3P           
##  Length:556         Length:556         Length:556         Length:556        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##      3PA                3P%                 2P                2PA           
##  Length:556         Length:556         Length:556         Length:556        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##      2P%                eFG%                FT                FTA           
##  Length:556         Length:556         Length:556         Length:556        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##      FT%                ORB                DRB                TRB           
##  Length:556         Length:556         Length:556         Length:556        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##      AST                STL                BLK                TOV           
##  Length:556         Length:556         Length:556         Length:556        
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##       PF                PTS           
##  Length:556         Length:556        
##  Class :character   Class :character  
##  Mode  :character   Mode  :character  
```

Age/Pts scored

```
library(ggplot2)
library(dplyr)

# Check for duplicate rows based on the "Player" column
duplicates_idx = duplicated(data_new$Player)

# Filter out duplicate rows
data_new = data_new[!duplicates_idx,]

# Define columns to convert to numeric
numeric_columns = c("Age", "G", "GS", "MP", "FG", "FGA", "FG%", "3P", "3PA", "3P%",
                     "2P", "2PA", "2P%", "eFG%", "FT", "FTA", "FT%", "ORB", "DRB",
                     "TRB", "AST", "STL", "BLK", "TOV", "PF", "PTS")

# Convert columns to numeric
data_new[numeric_columns] = lapply(data_new[numeric_columns], as.numeric)



library(dplyr)
# Only include players age 23-35
filtered_data <- data_new[data_new$Age >= 23 & data_new$Age <= 35, ]

# Group by Age & Player then calculate avg pts for each player at each age
average_points_by_age_player <- filtered_data %>%
  group_by(Age, Player) %>%
  summarise(Avg_Points = mean(PTS))
```

```
## `summarise()` has grouped output by 'Age'. You can override using the `.groups`
## argument.
```

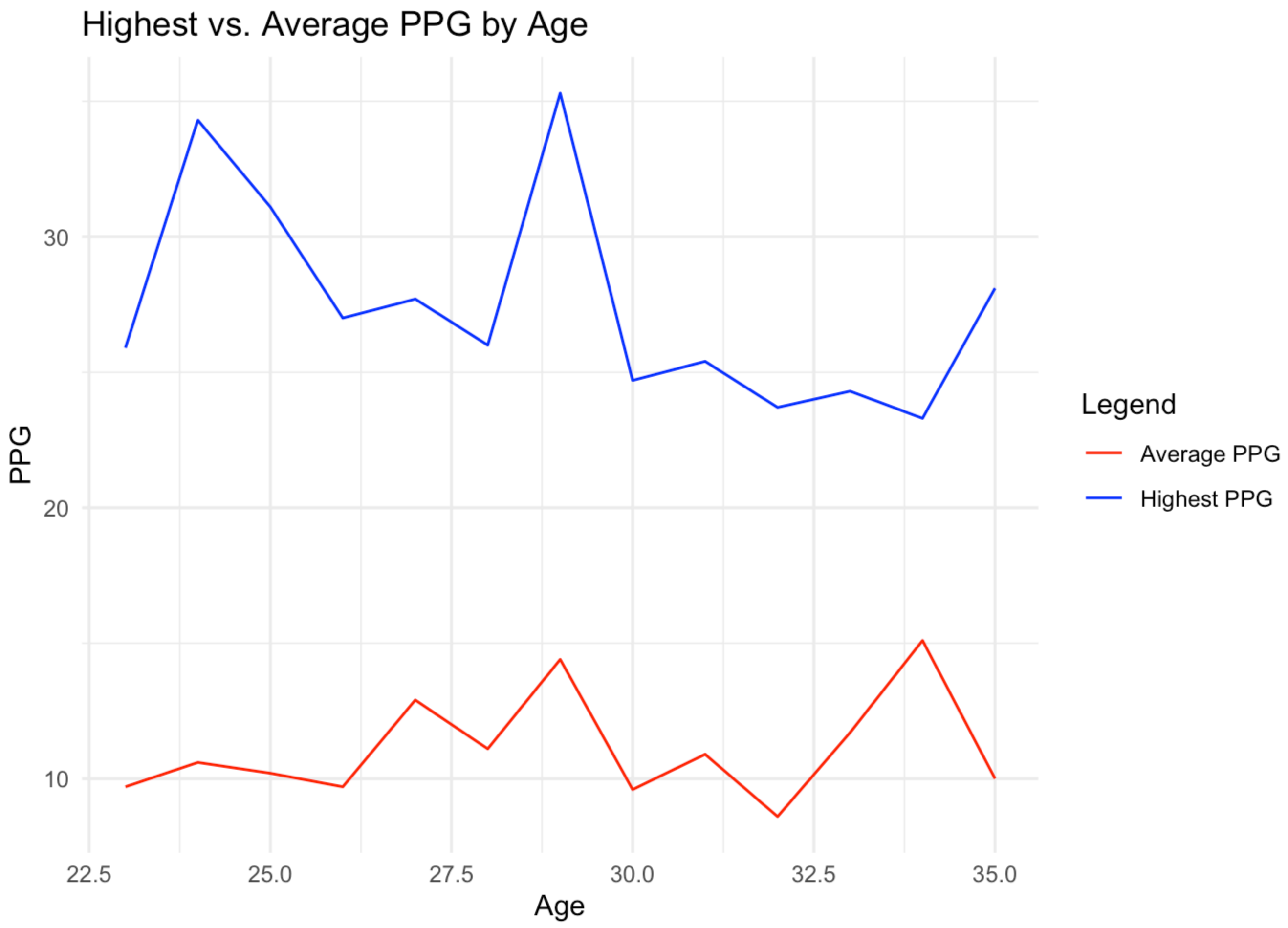
```
# Find the player with the highest avg pts for each age
top_players_by_age <- average_points_by_age_player %>%
  group_by(Age) %>%
  filter(Avg_Points == max(Avg_Points))

print(top_players_by_age)
```

```
## # A tibble: 13 × 3
## # Groups:   Age [13]
##      Age Player                  Avg_Points
##    <dbl> <chr>                        <dbl>
##  1    23 Tyrese Maxey                  25.9
##  2    24 Luka Don?i?                   34.3
##  3    25 Shai Gilgeous-Alexander       31.1
##  4    26 De'Aaron Fox                  27  
##  5    27 Donovan Mitchell              27.7
##  6    28 Nikola Joki?                  26  
##  7    29 Joel Embiid                   35.3
##  8    30 Anthony Davis                 24.7
##  9    31 Kyrie Irving                  25.4
## 10    32 Kawhi Leonard                 23.7
## 11    33 Damian Lillard                24.3
## 12    34 DeMar DeRozan                 23.3
## 13    35 Kevin Durant                  28.1
```

```
library(dplyr)

# Filter the data to include only players aged 23 to 35 and who played at least 10 minut
es
filtered_data <- data_new[data_new$Age >= 23 & data_new$Age <= 35 & data_new$MP >= 10, ]

# Calculate the highest player's points for each age
highest_points_by_age <- filtered_data %>%
  group_by(Age) %>%
  summarise(Highest_Points = max(PTS))

# Calculate the average points for each age (considering only players who played at leas
t 10 minutes)
average_points_by_age <- filtered_data %>%
  group_by(Age) %>%
  summarise(Avg_Points = round(mean(PTS), 1))

# Merge the two datasets for comparison
comparison <- merge(highest_points_by_age, average_points_by_age, by = "Age")

# Print the comparison
print(comparison)
```

```
##    Age Highest_Points Avg_Points
## 1   23           25.9        9.7
## 2   24           34.3       10.6
## 3   25           31.1       10.2
## 4   26           27.0        9.7
## 5   27           27.7       12.9
## 6   28           26.0       11.1
## 7   29           35.3       14.4
## 8   30           24.7        9.6
## 9   31           25.4       10.9
## 10  32           23.7        8.6
## 11  33           24.3       11.7
## 12  34           23.3       15.1
## 13  35           28.1       10.0
```

```r
library(ggplot2)

# Assuming you have dataframes highest_ppg_by_age and average_points_by_age with Age and PPG columns

# Create line plot for highest PPG by age
ggplot() +
  geom_line(data = highest_points_by_age, aes(x = Age, y = Highest_Points, color = "Highest PPG")) +
  geom_line(data = average_points_by_age, aes(x = Age, y = Avg_Points, color = "Average PPG")) +
  labs(title = "Highest vs. Average PPG by Age",
       x = "Age",
       y = "PPG",
       color = "Legend") +
  scale_color_manual(values = c("red", "blue")) +
  theme_minimal()
```



##TS%

```
#TS% = (PTS) / 2*(FGA + (0.44*FTA))

# Top 5 PPG players
sorted_data = data_new[order(-data_new$PTS), ]
top_players = sorted_data[1:5, ]

# Calculate the formula for the top 5 players
TS_val = ((top_players$PTS / (2 * (top_players$FGA + (0.44 * top_players$FTA)))) * 100)

# TS%
for(i in 1:5) {
  cat(top_players[i, "Player"], ": ", round(TS_val[i],1),"%", "\n")
}
```

```
## Joel Embiid :  64.3 %
## Luka Don?i? :  62.1 %
## Shai Gilgeous-Alexander :  64.5 %
## Giannis Antetokounmpo :  65.5 %
## Kevin Durant :  63.1 %
```

#Does Position Affect TS%?

```
# Only PG and C positions
pg_players = data_new[data_new$Pos == "PG", ]
c_players = data_new[data_new$Pos == "C", ]

# Calculate the TS%
ts_percent_pg = (pg_players$PTS) / (2 * (pg_players$FGA + (0.44 * pg_players$FTA))) * 10
0
ts_percent_c = (c_players$PTS) / (2 * (c_players$FGA + (0.44 * c_players$FTA))) * 100

# Calculate the avg TS%
avg_ts_percent_pg = mean(ts_percent_pg, na.rm = TRUE)
avg_ts_percent_c = mean(ts_percent_c, na.rm = TRUE)

cat("Average TS% for PG players:", avg_ts_percent_pg, "\n")
```

```
## Average TS% for PG players: 51.98816
```

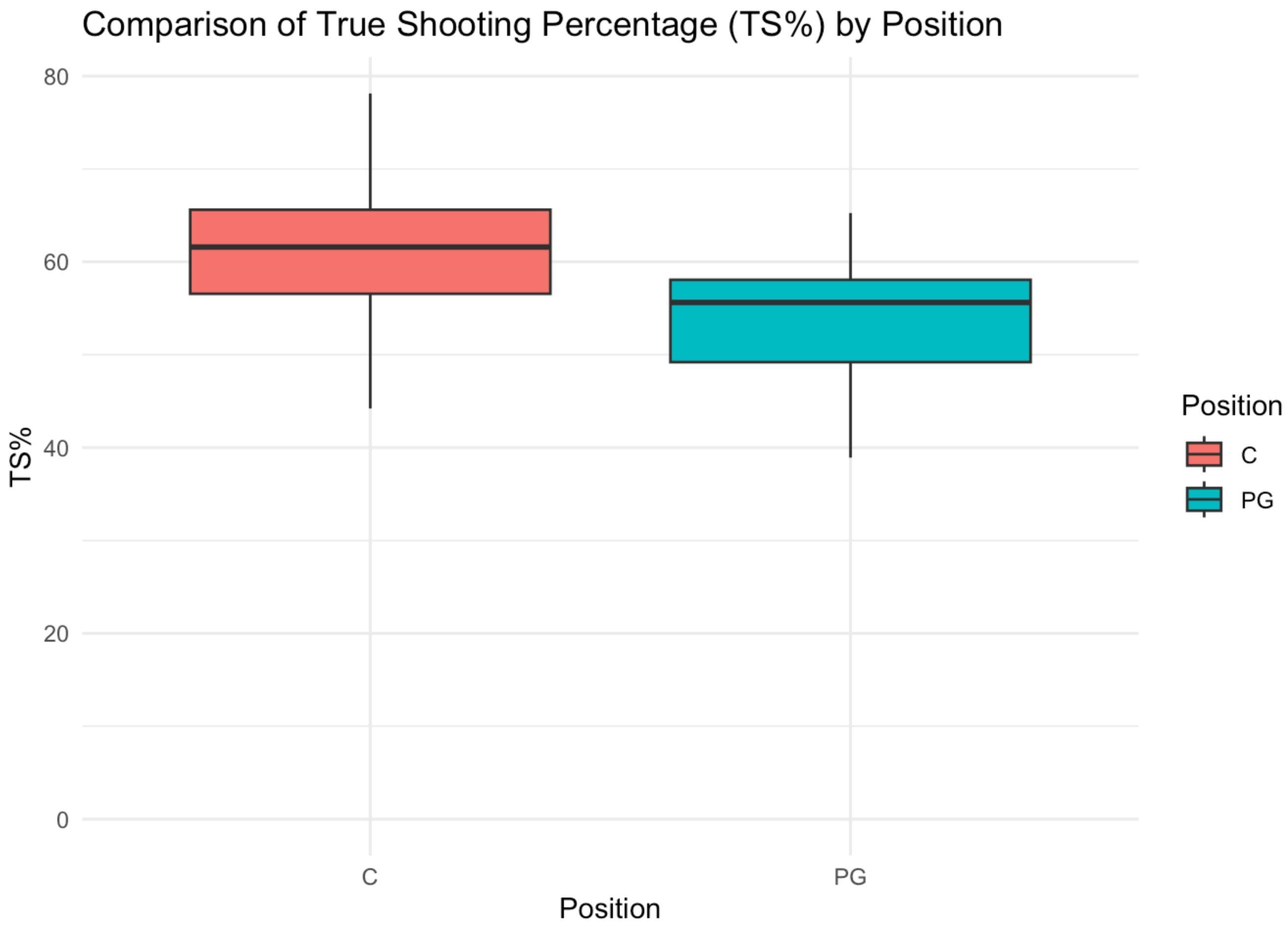
```
cat("Average TS% for C players:", avg_ts_percent_c, "\n")
```

```
## Average TS% for C players: 60.1759
```

#Boxplot

```
library(ggplot2)

# df
position_data = data.frame(
  Position = c(rep("PG", length(ts_percent_pg)), rep("C", length(ts_percent_c))),
  TS_Percent = c(ts_percent_pg, ts_percent_c)
)

# box plot
ggplot(position_data, aes(x = Position, y = TS_Percent, fill = Position)) +
  geom_boxplot(outlier.shape = NA) +
  labs(
    title = "Comparison of True Shooting Percentage (TS%) by Position",
    x = "Position",
    y = "TS%"
  ) +
  theme_minimal()
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



#Best PGs

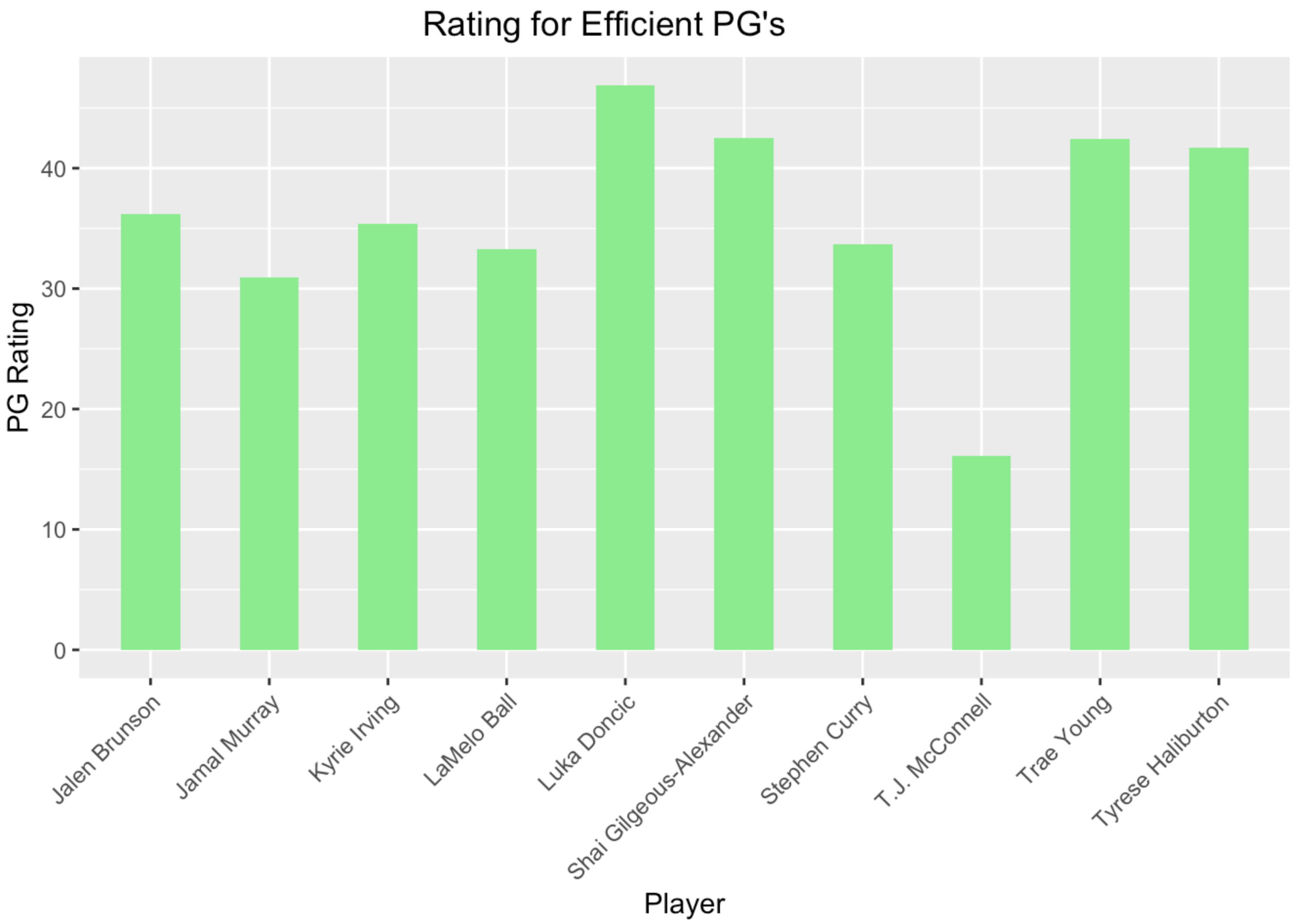
```
# Top 10 PGs in Player Efficiency Rating (PER)
selected_players = c("Shai Gilgeous-Alexander", "Luka Don?i?", "Jalen Brunson", "Tyrese
Haliburton",
                      "Kyrie Irving", "T.J. McConnell", "LaMelo Ball", "Jamal Murray",
                      "Stephen Curry", "Trae Young")

selected_data = data_new[data_new$Player %in% selected_players, ]

# Replace Don?i?" with Doncic
selected_data$Player = gsub("Luka Don\\?i\\?", "Luka Doncic", selected_data$Player)
print(selected_data)
```

```
##      Rk                   Player Pos Age  Tm  G GS   MP   FG  FGA   FG%  3P  3PA
## 31   22             LaMelo Ball  PG  22 CHO 22 22 32.3  8.3 19.2 0.433 3.2  9.0
## 105  74           Jalen Brunson  PG  27 NYK 61 61 35.2  9.6 20.3 0.475 2.6  6.6
## 148 111           Stephen Curry  PG  35 GSW 59 59 32.8  8.9 19.7 0.449 4.9 12.1
## 164 125             Luka Doncic  PG  24 DAL 58 58 37.4 11.7 23.7 0.494 3.9 10.3
## 223 168 Shai Gilgeous-Alexander  PG  25 OKC 65 65 34.4 11.0 20.2 0.544 1.3  3.5
## 249 190       Tyrese Haliburton  PG  23 IND 54 53 32.3  7.5 15.5 0.482 2.9  7.7
## 298 233            Kyrie Irving  SG  31 DAL 45 45 34.2  9.6 19.5 0.493 3.0  7.1
## 403 321          T.J. McConnell  PG  31 IND 56  4 17.8  4.1  7.4 0.550 0.2  0.5
## 450 357            Jamal Murray  PG  26 DEN 51 51 32.0  8.0 16.6 0.479 2.4  5.7
## 696 553              Trae Young  PG  25 ATL 51 51 36.5  8.2 19.3 0.426 3.3  8.9
##       3P%  2P  2PA   2P%  eFG%  FT FTA   FT% ORB DRB TRB  AST STL BLK TOV  PF
## 31  0.355 5.1 10.3 0.500 0.515 4.1 4.7 0.865 1.3 3.8 5.1  8.0 1.8 0.2 3.8 3.6
## 105 0.400 7.0 13.7 0.512 0.540 5.3 6.3 0.844 0.6 3.0 3.6  6.5 0.9 0.2 2.5 1.9
## 148 0.407 3.9  7.6 0.517 0.574 4.3 4.7 0.920 0.5 3.8 4.4  4.9 0.8 0.4 3.0 1.7
## 164 0.378 7.8 13.3 0.583 0.576 7.0 8.9 0.788 0.8 8.2 9.0  9.8 1.4 0.6 4.0 2.1
## 223 0.376 9.6 16.6 0.580 0.577 7.8 8.9 0.879 0.9 4.8 5.6  6.4 2.1 0.9 2.2 2.5
## 249 0.370 4.6  7.8 0.593 0.574 2.9 3.4 0.857 0.6 3.3 3.9 11.4 1.0 0.7 2.4 1.0
## 298 0.417 6.7 12.4 0.536 0.568 3.2 3.6 0.906 0.7 4.3 5.0  5.2 1.2 0.5 1.7 1.8
## 403 0.333 3.9  6.9 0.566 0.561 0.8 1.0 0.782 0.5 2.2 2.7  5.4 1.1 0.1 1.4 1.0
## 450 0.416 5.6 10.9 0.512 0.551 2.6 3.1 0.841 0.7 3.4 4.1  6.6 0.9 0.6 2.1 1.8
## 696 0.371 4.9 10.3 0.473 0.512 6.6 7.8 0.856 0.4 2.3 2.7 10.8 1.4 0.2 4.3 2.1
##      PTS
## 31  23.9
## 105 27.2
## 148 26.9
## 164 34.3
## 223 31.1
## 249 20.7
## 298 25.4
## 403  9.1
## 450 20.9
## 696 26.4
```

```
# Calculate TS_val
selected_data$TS_val = (selected_data$PTS / (2 * (selected_data$FGA + (0.44 * selected_data$FTA)))) * 100

selected_data$points_TS = ifelse(selected_data$TS_val < 57, -1, ifelse(selected_data$TS_val > 57, 1, 0))

points_scoring = selected_data$PTS
points_assists = 2 * selected_data$AST
points_turnovers = -2 * selected_data$TOV
points_FT = ifelse(selected_data$`FT%` > 0.85, 1, -1)
points_3P = ifelse(selected_data$`3P%` > 0.35, 1, -1)
total_points = points_scoring + points_assists + points_turnovers + points_FT + points_3P + selected_data$points_TS

player_points = data.frame(Player = selected_data$Player, PG_Rating = total_points)

player_points = player_points[order(-player_points$PG_Rating), ]

print(player_points)
```

```
##                     Player PG_Rating
## 4              Luka Doncic      46.9
## 5  Shai Gilgeous-Alexander      42.5
## 10             Trae Young      42.4
## 6        Tyrese Haliburton      41.7
## 2            Jalen Brunson      36.2
## 7             Kyrie Irving      35.4
## 3            Stephen Curry      33.7
## 1              LaMelo Ball      33.3
## 9             Jamal Murray      30.9
## 8           T.J. McConnell      16.1
```

```
library(ggplot2)

# Assuming you already have the dataframe player_points with the PG_Rating column

# Create bar plot
ggplot(player_points, aes(x = Player, y = PG_Rating)) +
  geom_bar(stat = "identity", fill = "lightgreen", width = 0.5) +
  labs(title = "                                    Rating for Efficient PG's",
       x = "Player",
       y = "PG Rating") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better readability
```


Rating for Efficient PG's
PG Rating
0
10
20
30
40
Jalen Brunson
Jamal Murray
Kyrie Irving
LaMelo Ball
Luka Doncic
Shai Gilgeous-Alexander
Stephen Curry
T.J. McConnell
Trae Young
Tyrese Haliburton
Player