Untitled

Connor Train

2024-08-26

```
library(readr)
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)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(tidyr)
library(conflicted)
```

Question 1

```
# Load the cleaned data
data <- read.csv("updated_2023-2024_Team_Alphabetical.csv")
```

```r
# Calculate differential metrics with the new column names
data <- data %>%
  mutate(
    Diff_eFG_Pct = Off_eFG_Pct - Def_eFG_Pct,
    Diff_TOV_Pct = Off_TOV_Pct - Def_TOV_Pct,
    Diff_ORB_Pct = Off_ORB_Pct - Def_DRB_Pct,
    Diff_FTR = Off_FTR - Def_FTR
  )
```

```r
# Calculate the correlation between differential metrics and wins
correlation_eFG <- cor(data$Diff_eFG_Pct, data$W)
correlation_TOV <- cor(data$Diff_TOV_Pct, data$W)
correlation_ORB <- cor(data$Diff_ORB_Pct, data$W)
correlation_FTR <- cor(data$Diff_FTR, data$W)

cat("Correlation between Differential eFG_Pct and Wins: ", correlation_eFG, "\n")
```

```
## Correlation between Differential eFG_Pct and Wins:  0.8897218
```

```r
cat("Correlation between Differential TOV_Pct and Wins: ", correlation_TOV, "\n")
```

```
## Correlation between Differential TOV_Pct and Wins:  -0.3905277
```

```r
cat("Correlation between Differential ORB_Pct and Wins: ", correlation_ORB, "\n")
```

```
## Correlation between Differential ORB_Pct and Wins:  -0.1217998
```

```r
cat("Correlation between Differential FTR and Wins: ", correlation_FTR, "\n")
```

```
## Correlation between Differential FTR and Wins:  0.3905331
```

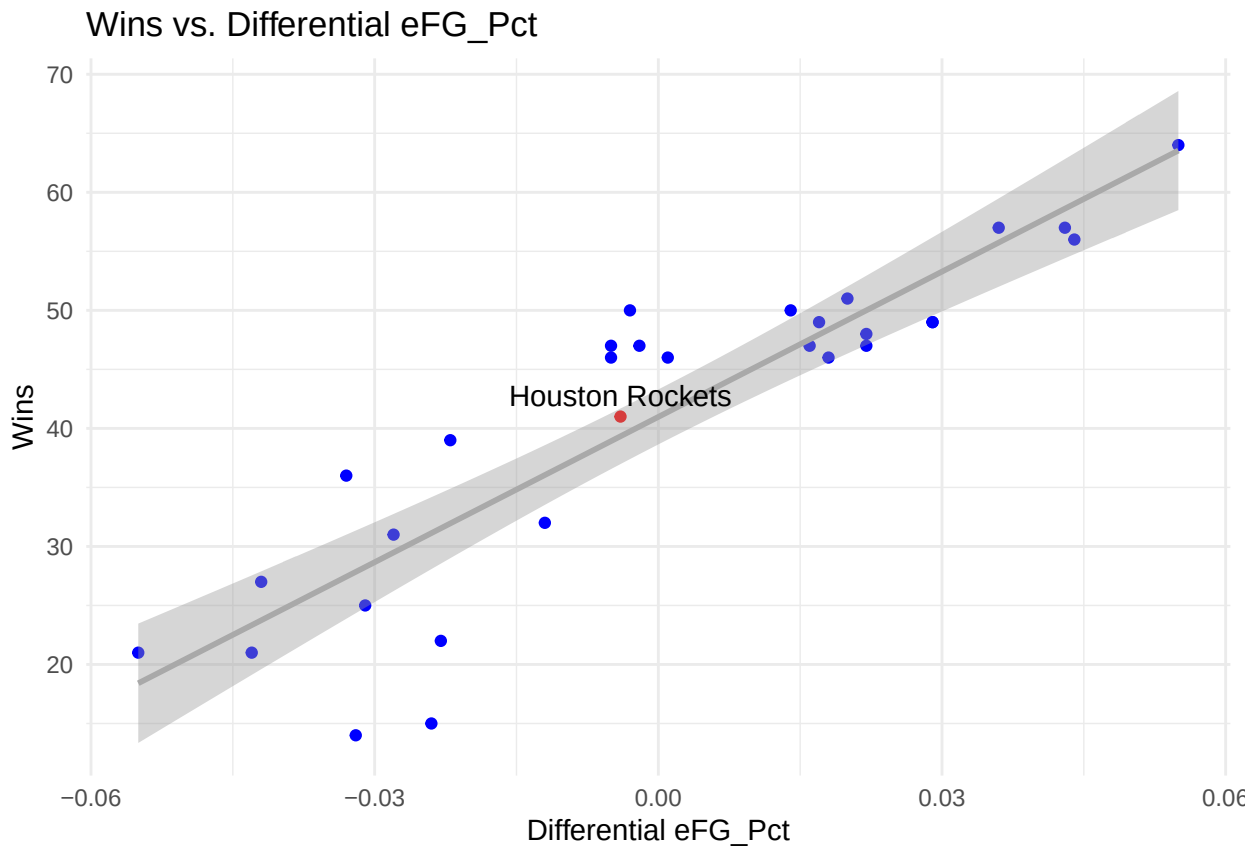
```r
# Rank teams based on differential metrics and their Offensive Rating (ORtg)
data <- data %>%
  mutate(
    eFG_Rank = rank(-Diff_eFG_Pct),  # Negative for descending order (higher eFG_Pct is better)
    TOV_Rank = rank(Diff_TOV_Pct),   # Positive for ascending order (lower TOV_Pct is better)
    ORB_Rank = rank(-Diff_ORB_Pct),  # Negative for descending order (higher ORB_Pct is better)
    FTR_Rank = rank(-Diff_FTR),      # Negative for descending order (higher FTR is better)
    ORtg_Rank = rank(-ORtg)          # Negative for descending order (higher ORtg is better)
  )

# Rockets' rankings within the league and compare to other teams
data %>%
  dplyr::filter(Team == "Houston Rockets" |
                Team %in% c("Dallas Mavericks", "Memphis Grizzlies",
                            "San Antonio Spurs", "New Orleans Pelicans")) %>%
  select(Team, eFG_Rank, TOV_Rank, ORB_Rank, FTR_Rank, ORtg_Rank)
```

```
##                   Team eFG_Rank TOV_Rank ORB_Rank FTR_Rank ORtg_Rank
## 1     Dallas Mavericks       13      7.5       21     14.0       9.5
```

```
## 2      Houston Rockets       17      5.5       13     22.0      20.0
## 3    Memphis Grizzlies       28     13.0       14     20.0      30.0
## 4 New Orleans Pelicans       11      4.0       18      8.5      11.0
## 5    San Antonio Spurs       22     27.0       24     16.0      26.0
```

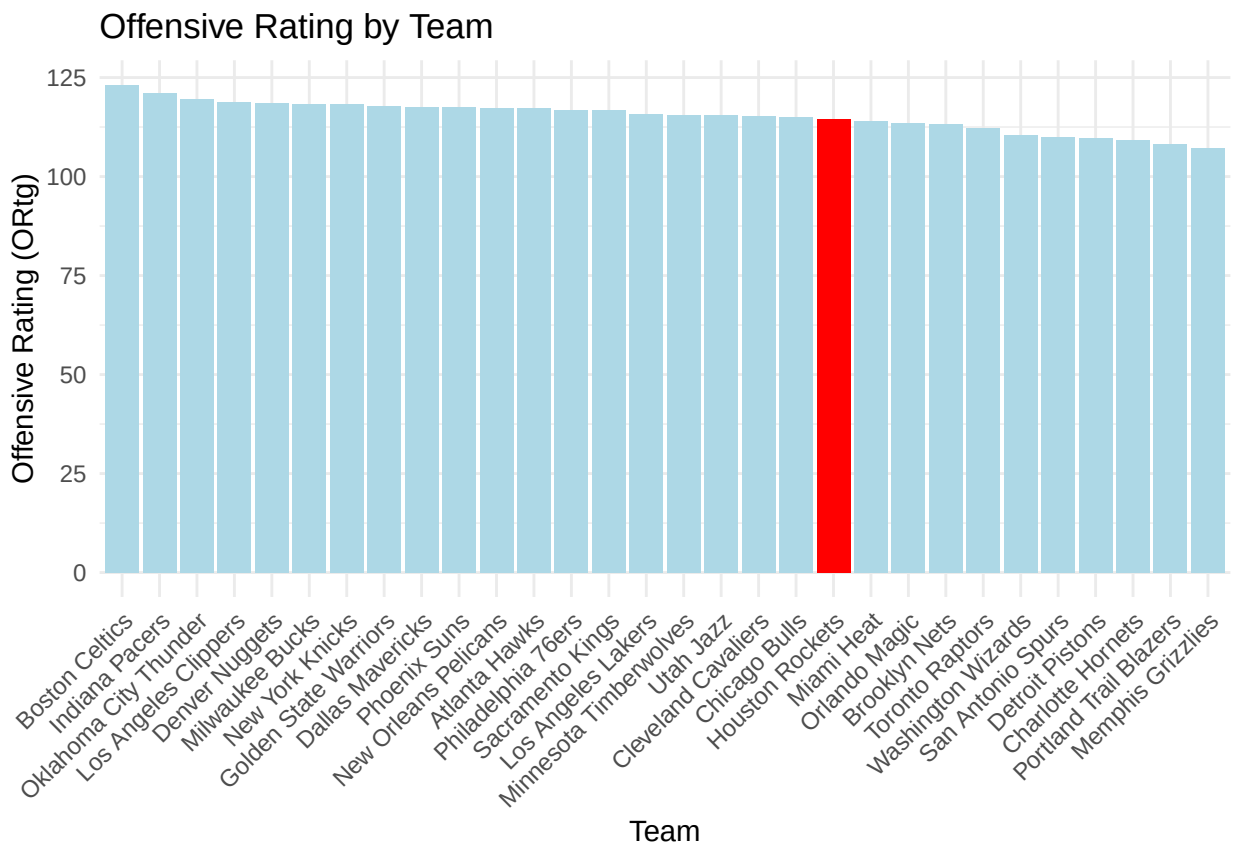
```r
# Scatter plot of Wins vs. Differential eFG_Pct with label for Houston Rockets
ggplot(data, aes(x = Diff_eFG_Pct, y = W)) +
  geom_point(color = ifelse(data$Team == "Houston Rockets", "red", "blue")) +
  geom_smooth(method = "lm", color = "darkgray") +  # Add a linear trendline
  geom_text(data = subset(data, Team == "Houston Rockets"),
            aes(label = Team), vjust = -0.5, hjust = 0.5, color = "black", size = 4) +  # Label only th
  labs(title = "Wins vs. Differential eFG_Pct",
       x = "Differential eFG_Pct",
       y = "Wins") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

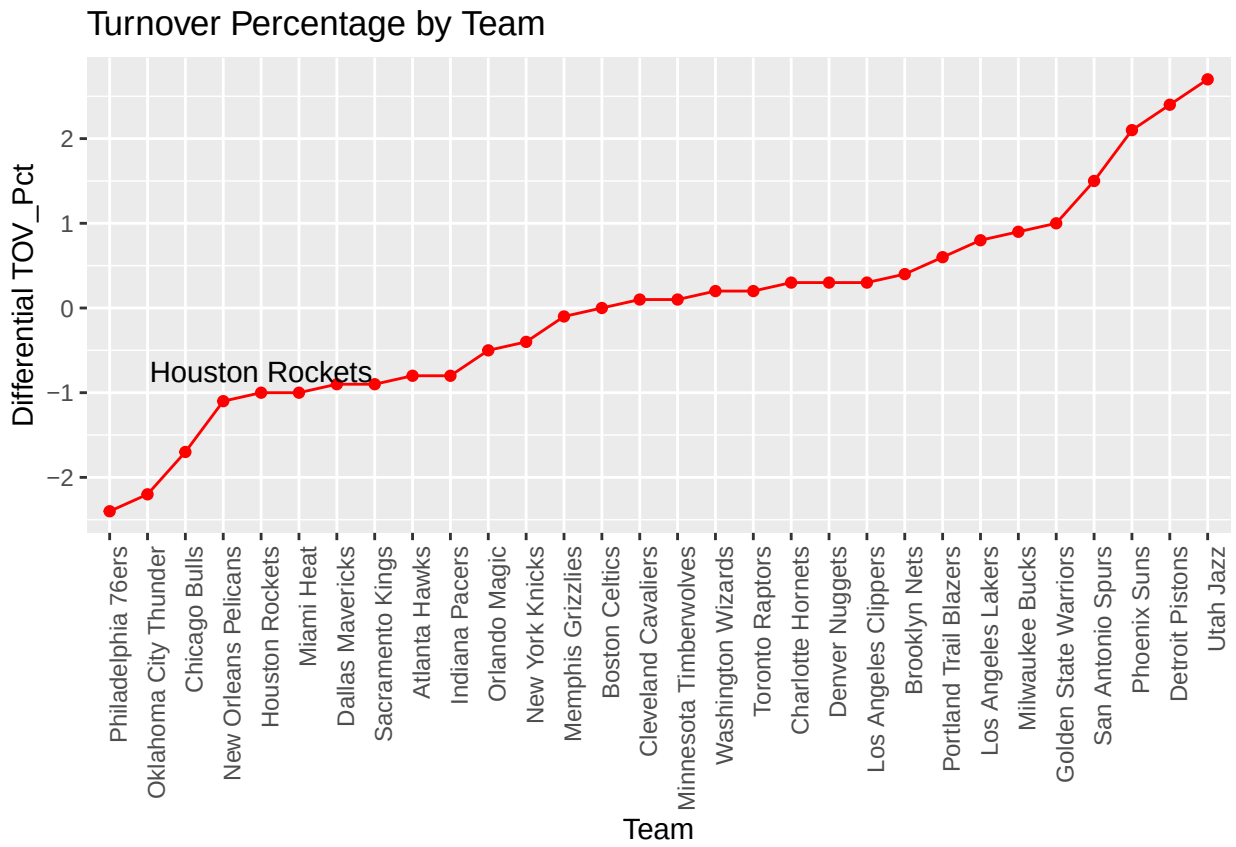



```r
# Bar chart of Offensive Rating (ORtg) by Team with Rockets highlighted
ggplot(data, aes(x = reorder(Team, -ORtg), y = ORtg, fill = Team == "Houston Rockets")) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("TRUE" = "red", "FALSE" = "lightblue")) +
  labs(title = "Offensive Rating by Team",
       x = "Team",
       y = "Offensive Rating (ORtg)") +
```

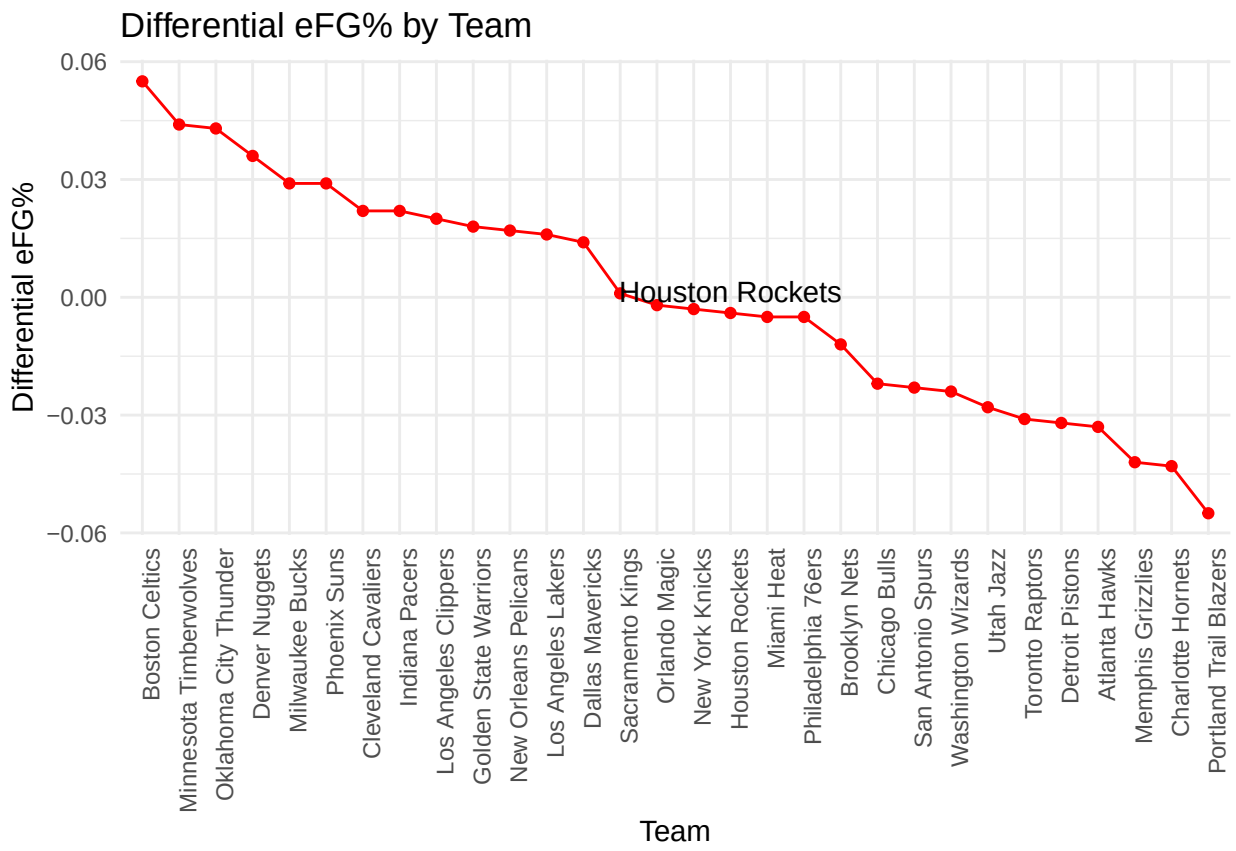
```
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        legend.position = "none")
```




```
# Line graph of Turnover Percentage (TOV_Pct) by Team
ggplot(data, aes(x = reorder(Team, TOV_Rank), y = Diff_TOV_Pct)) +
  geom_line(group = 1, color = "red") +
  geom_point(color = "red") +
  geom_text(data = subset(data, Team == "Houston Rockets"),
            aes(label = Team), vjust = -0.5, hjust = 0.5, color = "black", size = 4) +  # Label only th
  labs(title = "Turnover Percentage by Team",
       x = "Team",
       y = "Differential TOV_Pct") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```r
# Line graph of Differential eFG% (Diff_eFG_Pct) by Team
ggplot(data, aes(x = reorder(Team, eFG_Rank), y = Diff_eFG_Pct)) +
  geom_line(group = 1, color = "red") +
  geom_point(color = "red") +
  geom_text(data = subset(data, Team == "Houston Rockets"),
            aes(label = Team), vjust = -0.5, hjust = 0.5, color = "black", size = 4) +  # Label only th
  labs(title = "Differential eFG% by Team",
       x = "Team",
       y = "Differential eFG%") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Question 2

```r
# Perform the multiple linear regression
model <- lm(W ~ Diff_eFG_Pct + Diff_TOV_Pct + Diff_ORB_Pct + Diff_FTR, data = data)

# View the summary of the regression model
summary(model)
```

```
##
## Call:
## lm(formula = W ~ Diff_eFG_Pct + Diff_TOV_Pct + Diff_ORB_Pct +
##     Diff_FTR, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.8698  -1.9872   0.6042   2.0988   8.2726
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   89.1459    16.0869   5.542 9.24e-06 ***
## Diff_eFG_Pct 381.8220    29.2329  13.061 1.14e-12 ***
## Diff_TOV_Pct  -3.3459     0.6539  -5.117 2.76e-05 ***
## Diff_ORB_Pct   0.9343     0.3118   2.996  0.00609 **
## Diff_FTR     106.0491    30.5605   3.470  0.00190 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.146 on 25 degrees of freedom
## Multiple R-squared:  0.9176, Adjusted R-squared:  0.9044
## F-statistic: 69.61 on 4 and 25 DF,  p-value: 3.502e-13
```

```
# Filter Rockets
rockets_data <- data %>% dplyr::filter(Team == "Houston Rockets")

# Predicted wins for the Rockets
predicted_wins <- predict(model, newdata = rockets_data)
predicted_wins
```

```
##        1
## 41.28876
```

```
# Compare predicted wins to actual wins for the Rockets
actual_wins <- rockets_data$W
cat("Predicted Wins: ", round(predicted_wins, 2), "\n")
```

```
## Predicted Wins:  41.29
```

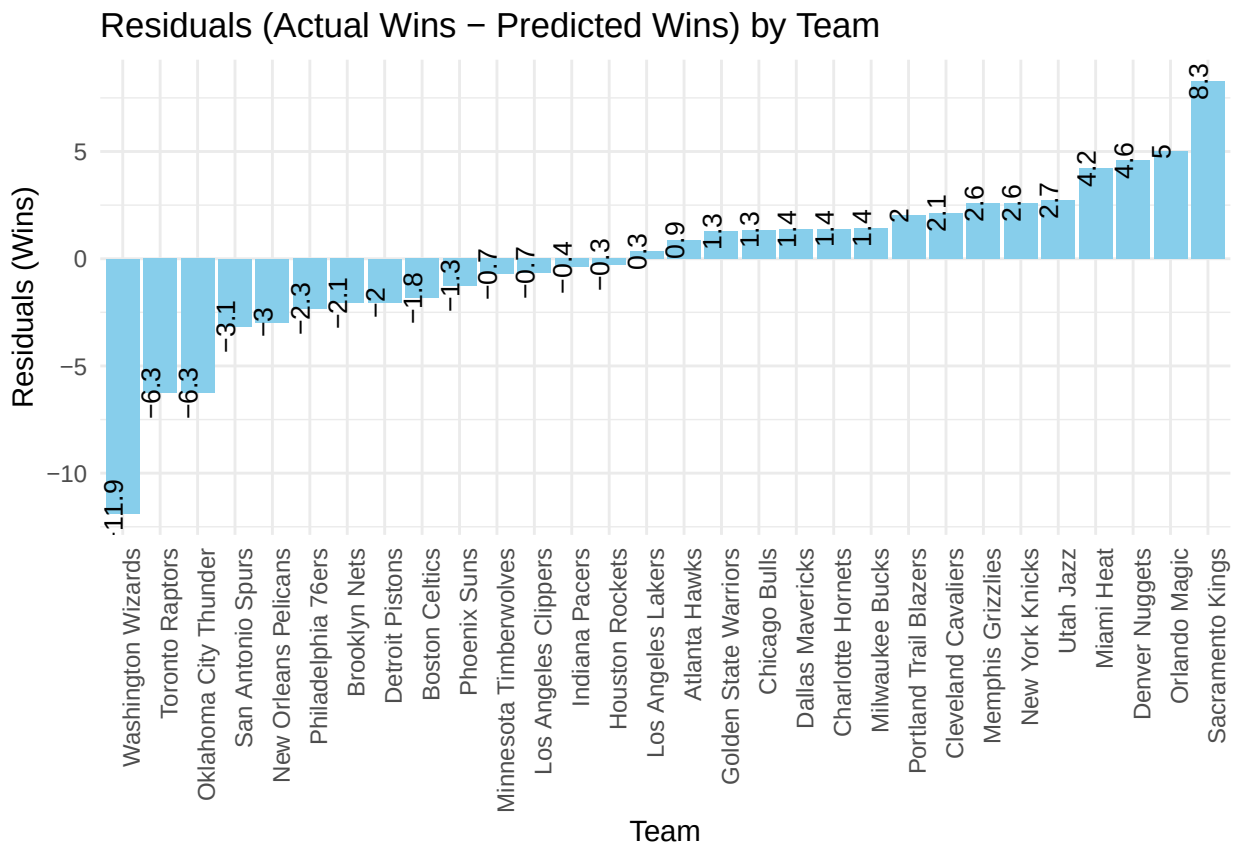
```
cat("Actual Wins: ", actual_wins, "\n")
```

```
## Actual Wins:  41
```

```
# Calculate residuals for all teams
data <- data %>%
  mutate(Predicted_Wins = predict(model, newdata = data),
         Residuals = W - Predicted_Wins)

# View residuals for the Rockets
rockets_residual <- data %>% dplyr::filter(Team == "Houston Rockets") %>%
  select(Team, W, Predicted_Wins, Residuals)

rockets_residual
```

```
##              Team  W Predicted_Wins  Residuals
## 1 Houston Rockets 41       41.28876 -0.2887603
```

```
# Plot residuals for all teams
ggplot(data, aes(x = reorder(Team, Residuals), y = Residuals)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  geom_text(aes(label = round(Residuals, 1)), vjust = 0, hjust = 0.5, angle = 90, color = "black", size
  labs(title = "Residuals (Actual Wins - Predicted Wins) by Team",
       x = "Team",
       y = "Residuals (Wins)") +
   theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Question 3

```
# Load the dataset
player_data <- read_csv("2023-2024 NBA Player Stats - Regular.csv")
```

```
## Rows: 735 Columns: 30
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): Player, Pos, Tm
## dbl (27): Rk, Age, G, GS, MP, FG, FGA, FG_Pct, 3P, 3PA, 3P_Pct, 2P, 2PA, 2P_...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Remove the 'Rk' (Rank) column as it's not needed for classification
player_data <- player_data %>% select(-Rk)

# Handle any missing values
player_data <- na.omit(player_data)
```

```
# Select relevant columns for player classification
player_data <- player_data %>%
  select(Player, Pos, Tm, PTS, AST, TRB, STL, BLK, eFG_Pct)
```

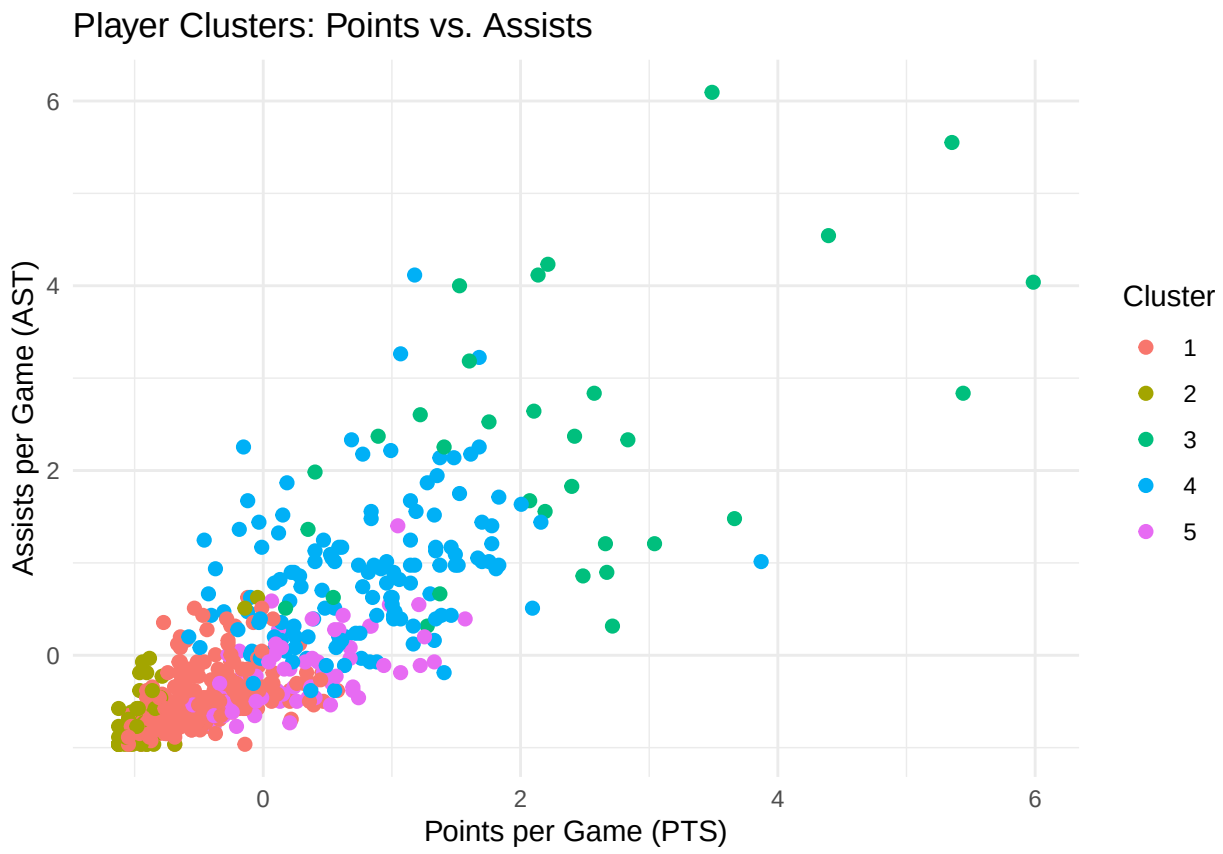
```r
# Aggregate the player stats by summing or averaging where appropriate
player_data_aggregated <- player_data %>%
  group_by(Player) %>%
  summarize(
    PTS = sum(PTS),
    AST = sum(AST),
    TRB = sum(TRB),
    STL = sum(STL),
    BLK = sum(BLK),
    eFG_Pct = mean(eFG_Pct)
  )
```

```r
# Standardize the data
player_data_scaled <- player_data_aggregated %>%
  mutate(across(c(PTS, AST, TRB, STL, BLK, eFG_Pct), scale))
```

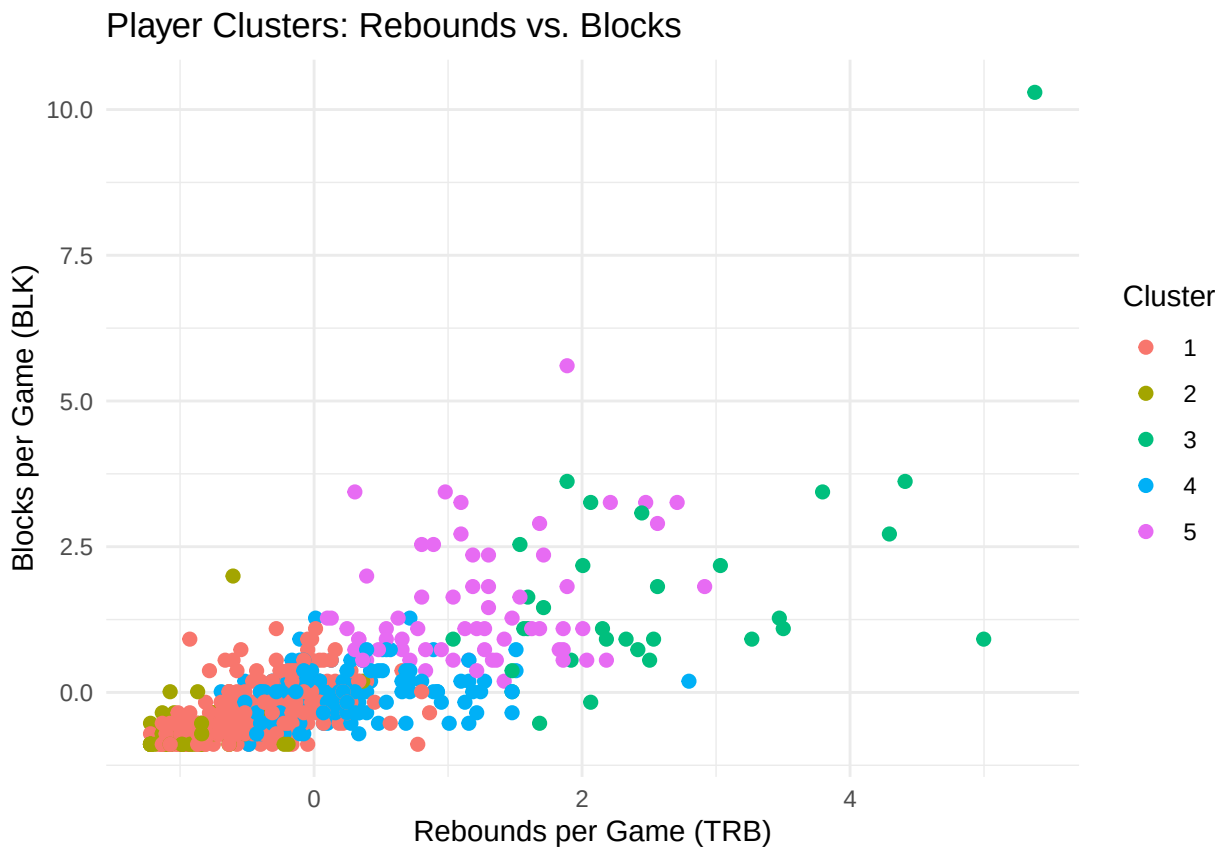
```r
# Set a seed for reproducibility
set.seed(42)

# Perform K-Means clustering with 5 clusters
kmeans_result <- kmeans(player_data_scaled[, c('PTS', 'AST', 'TRB', 'STL', 'BLK', 'eFG_Pct')], centers =

# Add the cluster assignment to the data
player_data_scaled$Cluster <- kmeans_result$cluster
```

```r
# Summarize the clusters to understand the player types
cluster_summary <- player_data_scaled %>%
  group_by(Cluster) %>%
  summarize(
    Avg_PTS = mean(PTS),
    Avg_AST = mean(AST),
    Avg_TRB = mean(TRB),
    Avg_STL = mean(STL),
    Avg_BLK = mean(BLK),
    Avg_eFG_Pct = mean(eFG_Pct)
  )
```

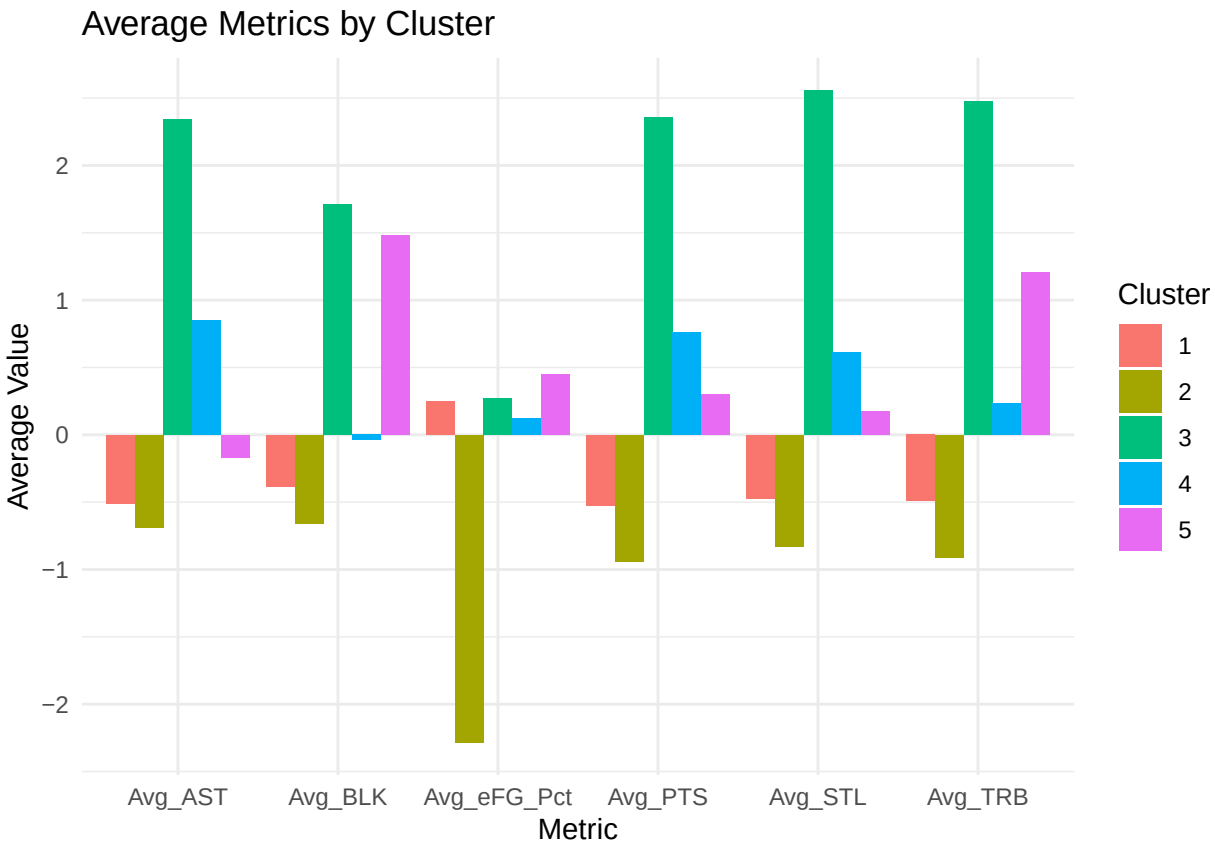
```r
# Scatter plot of Points per Game (PTS) vs. Assists per Game (AST) colored by Cluster
ggplot(player_data_scaled, aes(x = PTS, y = AST, color = as.factor(Cluster))) +
  geom_point(size = 2) +
  labs(title = "Player Clusters: Points vs. Assists",
       x = "Points per Game (PTS)",
       y = "Assists per Game (AST)",
       color = "Cluster") +
  theme_minimal()
```




```
# Scatter plot of Rebounds per Game (TRB) vs. Blocks per Game (BLK) colored by Cluster
ggplot(player_data_scaled, aes(x = TRB, y = BLK, color = as.factor(Cluster))) +
  geom_point(size = 2) +
  labs(title = "Player Clusters: Rebounds vs. Blocks",
       x = "Rebounds per Game (TRB)",
       y = "Blocks per Game (BLK)",
       color = "Cluster") +
  theme_minimal()
```




```r
# Calculate the average metrics for each cluster for a bar plot
cluster_avg <- cluster_summary %>%
  pivot_longer(cols = -Cluster, names_to = "Metric", values_to = "Value")

# Bar plot of average metrics by cluster
ggplot(cluster_avg, aes(x = Metric, y = Value, fill = as.factor(Cluster))) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Metrics by Cluster",
       x = "Metric",
       y = "Average Value",
       fill = "Cluster") +
  theme_minimal()
```


Average Metrics by Cluster
Average Value
2
1
0
−1
−2
Avg_AST
Avg_BLK
Avg_eFG_Pct
Avg_PTS
Avg_STL
Avg_TRB
Metric
Cluster
1
2
3
4
5