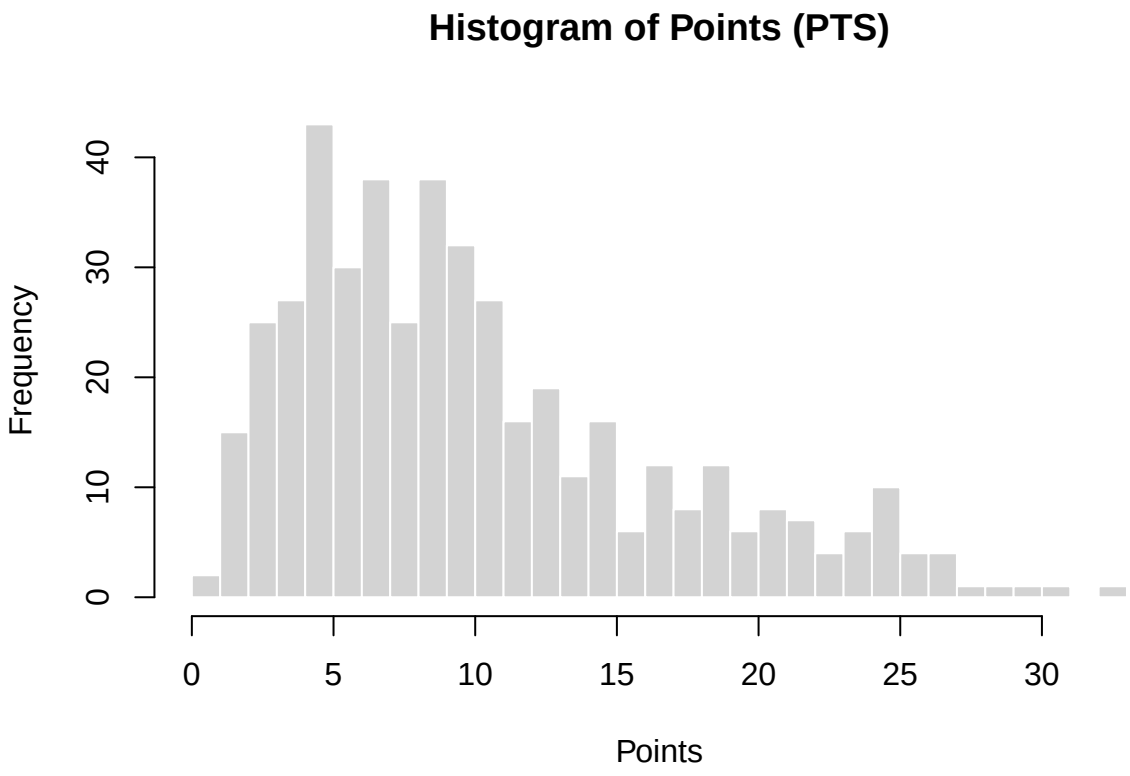
Regression Modeling

2025-12-05

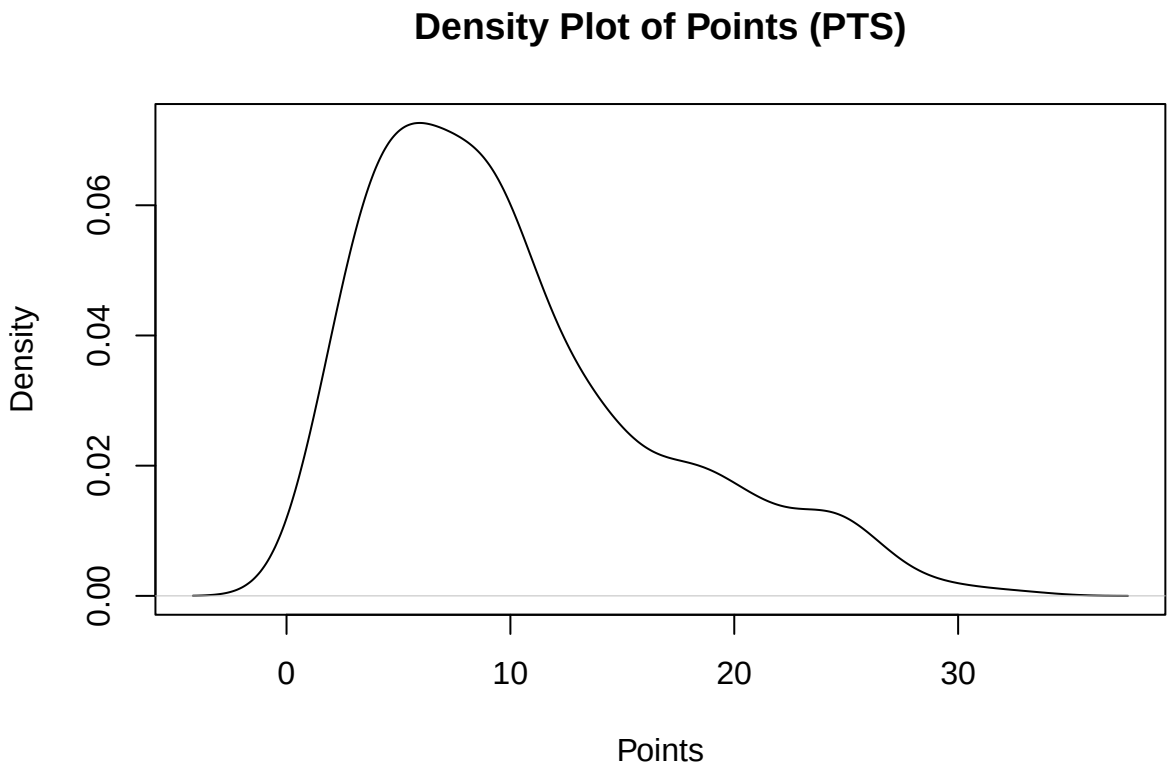
We will continue our analysis of the 2024 NBA player statistics by looking to implement both Linear and logistic Regression models to predict player's points per game (PTS) as well as the overall ranking of the players respectively based on other statistics. We will use our insights from our correlation analysis in helping choose appropriate parameters within these models.

Before fitting a linear regression model to predict PTS, it's useful to look at how the response variable is distributed. This helps confirm that the values make sense, there are no extreme outliers, and that PTS behaves in a way that works well for linear modeling.

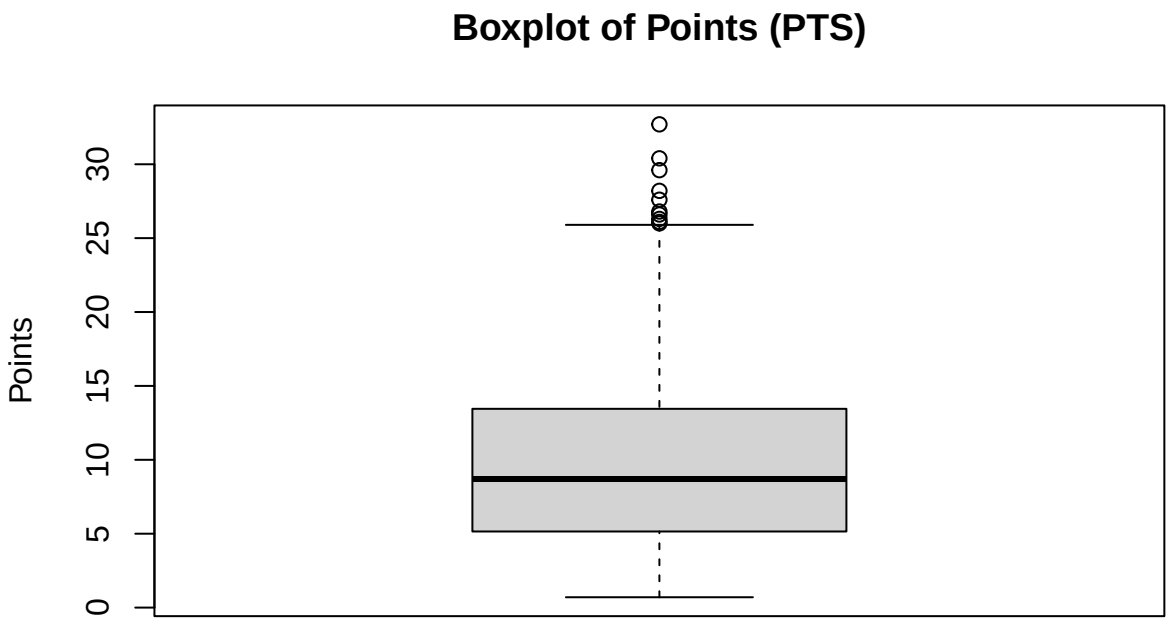
```
hist(
  clean_data$PTS,
  breaks = 30,
  main = "Histogram of Points (PTS)",
  xlab = "Points",
  col = "lightgray",
  border = "white"
)
```



```r
plot(
  density(clean_data$PTS, na.rm = TRUE),
  main = "Density Plot of Points (PTS)",
  xlab = "Points"
)
```




```r
boxplot(
  clean_data$PTS,
  main = "Boxplot of Points (PTS)",
  ylab = "Points"
)
```



The histogram shows a right-skewed distribution, which is completely expected for NBA scoring data. Most players score modest amounts, while a smaller group scores very high this is natural and not a problem for linear regression. There is no severe skew, no long tail, and no shape that suggests the need for a transformation.

The density curve reinforces this pattern. It is unimodal, smooth, and shows only mild right-skewness. Nothing in the distribution suggests instability or irregularity.

The boxplot shows a handful of high-scoring players that appear as "upper outliers," but these represent real star players, not data errors. This is normal for sports stats and does not violate linear regression assumptions.

The distribution of PTS shows mild right-skewness and a small number of high-scoring players, which is expected in NBA data. These patterns do not violate linear regression assumptions, and the data appears appropriate for modeling without transformation. As such we can continue with implementation of our model.

We will begin with choosing variables we want to include in the model.

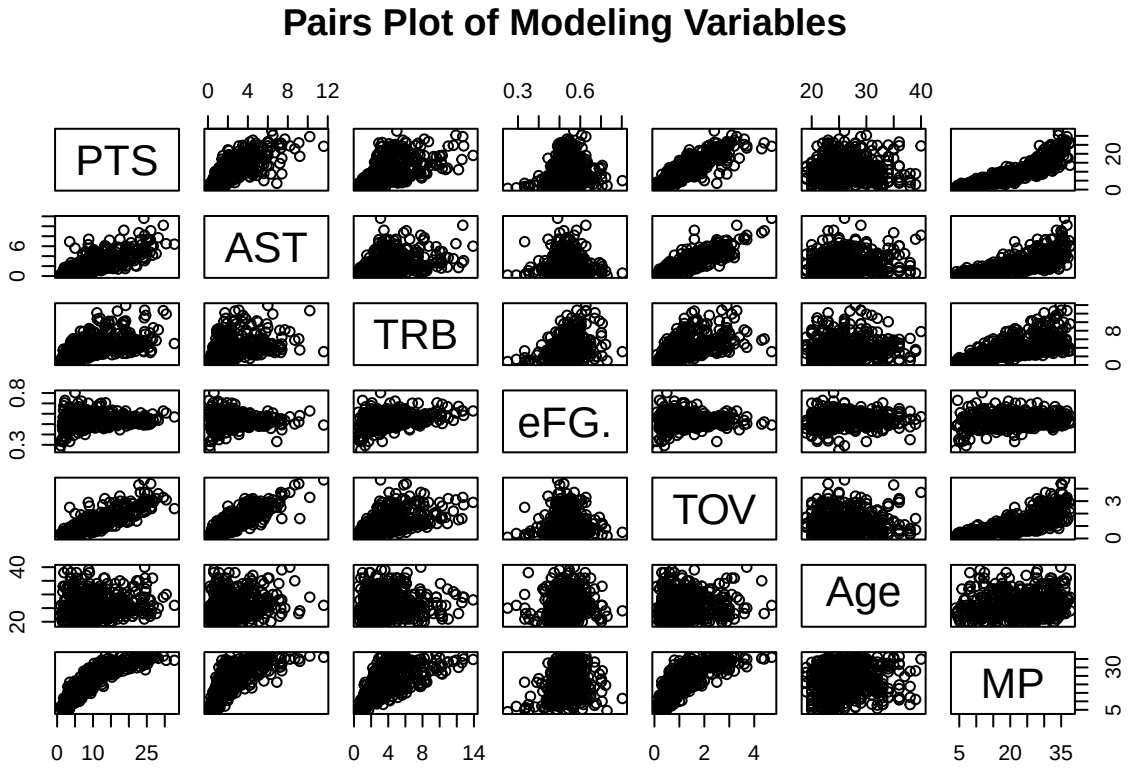
```
model_vars <- c("PTS", "AST", "TRB", "eFG.", "TOV", "Age", "MP")
stats_model <- clean_data[, model_vars, drop = FALSE]
```

Remove rows with missing values only for modelling

```
stats_model <- stats_model |> drop_na()
```

Exploratory Check of Predictors:

```r
pairs(stats_model, main = "Pairs Plot of Modeling Variables")
```



Before fitting the regression model, we take a quick look at the relationships between PTS and potential predictors. This helps confirm that the variables chosen are relevant and not too strongly correlated with each other. For example we would not want to include "FGA" as one of our variables as shown in our correlation analysis pts~fga had a correlation value of 0.99 making them too strongly correlated for the sake of our analysis as we would like to explore the predictive power of other variables rather then just creating the most accurate model.

As we already know if we include FGA in our model it will make other variables insignificant we will look to create two seperate models. One without FGA to look explore the contribution of other statistics in predicting number of points. And another with FGA to create the most accurate model.

We will begin with the model excluding FGA. This model predicts points per game using a combination of shooting volume (FGA), playmaking (AST), rebounding (TRB), shooting efficiency (eFG.), turnovers (TOV), and age. These variables were chosen to avoid multicollinearity and to reflect different aspects of player performance.

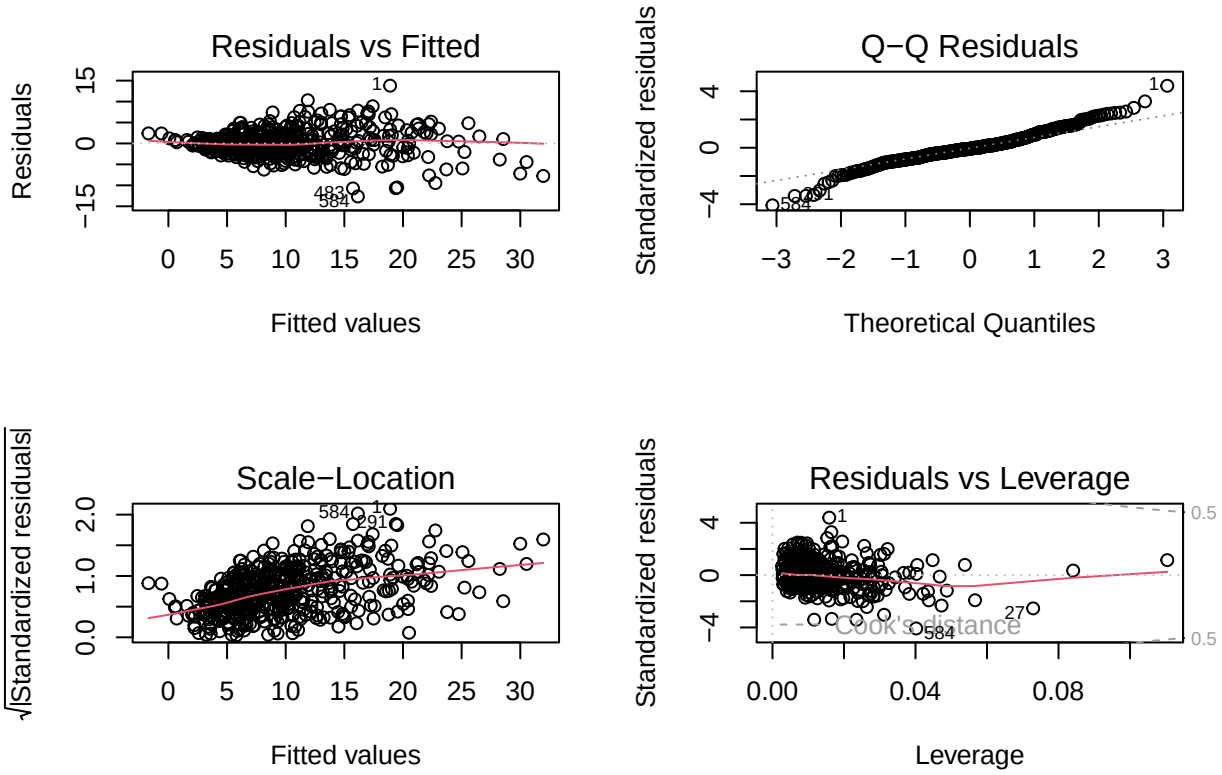
```r
linear_model_1 <- lm(PTS ~ AST + TRB + eFG. + TOV + Age, data = stats_model)
summary(linear_model_1)
```

```
##
## Call:
## lm(formula = PTS ~ AST + TRB + eFG. + TOV + Age, data = stats_model)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -12.6543  -1.7210  -0.2434   1.5287  13.7962
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.98182    1.53232  -4.556 6.71e-06 ***
## AST          0.37809    0.16846   2.244 0.025292 *
## TRB          0.30345    0.08250   3.678 0.000263 ***
## eFG.        13.28761    2.52456   5.263 2.19e-07 ***
## TOV          5.52321    0.42108  13.117  < 2e-16 ***
## Age          0.04355    0.03593   1.212 0.226106
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.169 on 450 degrees of freedom
## Multiple R-squared:  0.7643, Adjusted R-squared:  0.7616
## F-statistic: 291.8 on 5 and 450 DF,  p-value: < 2.2e-16
```

The linear regression model without FGA explains a substantial portion of scoring variation (Adjusted $R^2 = 0.76$). Assists, rebounds, shooting efficiency, and turnovers are all significant positive predictors of points per game, reflecting that players with higher offensive involvement and efficiency tend to score more. Age was not statistically significant, which matches our earlier correlation findings showing that age is only weakly related to performance variables.

```r
#The diagnostic plots help us check whether the assumptions of linear regression are met, including lin
par(mfrow=c(2,2))
plot(linear_model_1)
```



```r
par(mfrow=c(1,1))
```

The residuals vs fitted plot shows no major patterns, suggesting that the linearity assumption is reasonably met. The Q–Q plot indicates that residuals are approximately normally distributed, with only mild deviations at the extremes. The scale–location plot shows relatively constant variance, suggesting that the homoscedasticity assumption is not violated. The residuals vs leverage plot identifies a few influential cases, which is expected for NBA data, but none appear to unduly distort the model.

The diagnostic plots indicate that the assumptions of linear regression are reasonably satisfied. The residuals show no major nonlinear patterns, residual variance appears relatively constant, and residuals are approximately normally distributed. A few influential observations appear in the leverage plot, which is expected due to the presence of high-usage superstar players, but they do not distort the model. Overall, the model is statistically sound and provides meaningful insights into the predictors of scoring when shot volume is intentionally excluded.

Within our data set we include data on the overall rankings of players throughout the 2024-2025 Season. We will first look to see if we can accurately predict the numerical ranking of players using a linear regression model.

We begin by creating a 80/20 Split to train and test the accuracy of a linear model.

```r
# Create training indices
clean_data <- clean_data |> drop_na()

train_idx <- sample(1:nrow(clean_data), size = 0.8 * nrow(clean_data))

train_data <- clean_data[train_idx, ]
test_data  <- clean_data[-train_idx, ]
```

```r
rk_lm_model <- lm(Rk ~ PTS + AST + TRB + STL + BLK + eFG. + MP + TOV,
                  data = train_data)

summary(rk_lm_model)
```

```
##
## Call:
## lm(formula = Rk ~ PTS + AST + TRB + STL + BLK + eFG. + MP + TOV,
##     data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -95.267 -26.597   1.381  26.412 110.010
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  628.7906    19.8873  31.618  < 2e-16 ***
## PTS          -11.9335     0.8551 -13.956  < 2e-16 ***
## AST           12.3877     2.6927   4.601 5.92e-06 ***
## TRB           -0.8460     1.6235  -0.521  0.60262
## STL            3.4149     8.1623   0.418  0.67593
## BLK           17.0800     8.4023   2.033  0.04284 *
## eFG.        -150.5187    37.9285  -3.968 8.80e-05 ***
## MP            -8.8593     0.6077 -14.579  < 2e-16 ***
## TOV          -20.5552     7.0682  -2.908  0.00387 **
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.38 on 346 degrees of freedom
## Multiple R-squared:  0.9309, Adjusted R-squared:  0.9293
## F-statistic: 582.3 on 8 and 346 DF,  p-value: < 2.2e-16
```

```
train_pred_lm <- predict(rk_lm_model, newdata = train_data)
test_pred_lm  <- predict(rk_lm_model, newdata = test_data)

# Compute RMSE
rmse_train <- sqrt(mean((train_data$Rk - train_pred_lm)^2))
rmse_test  <- sqrt(mean((test_data$Rk - test_pred_lm)^2))

rmse_train
```

```
## [1] 38.87785
```

```
rmse_test
```

```
## [1] 40.26133
```

As shown our RMSE for our training set is 38.3585 and for our testing set is 41.57316. As these values are very close together, this tells us that our model is not over fitting and is able to generalize to new unseen data.

The RMSE values being around ~40 indicates to us that on overage the model is able to predict the numerical ranking of players of within 40 places from their actual rank. While this may seem like a lot, this is to be expected as the numerical ranking of players is non-linear and many players have very similar stats. As such small differences in ranking may come from subjective or external factors that can not be explained purely through raw data. The results suggest that the model captures broad patterns in how player statistics relate to ranking, but we are limited by the complexity of external factors and their influences on ranking as well as the limitations in linear regression. While our model is not precise in rank prediction, it is useful in the understanding of general trends.

We now will look to create a multinomial Log-linear model to see if we can accurately determine if a player is an Elite/Above Average/Role(Average) or Low(below average) player.

```
clean_data$RankTier <- cut(
  clean_data$Rk,
  breaks = c(0, 50, 150, 300, 444),
  labels = c("Low", "Role", "Above Average", "Elite")
)
clean_data$RankTier <- relevel(clean_data$RankTier, ref = "Low")
```

```
n <- nrow(clean_data)

# Creating train/test data
train_idx <- sample(1:n, size = 0.8 * n)
train_data <- clean_data[train_idx, ]
test_data  <- clean_data[-train_idx, ]
```

```r
# Fitting model
multi_mod <- multinom(
  RankTier ~ PTS + MP + AST + TRB + STL + BLK + eFG. + TOV + Age,
  data = train_data
)
```

```
## # weights:  44 (30 variable)
## initial  value 428.364958
## iter  10 value 241.859974
## iter  20 value 141.398011
## iter  30 value 19.088812
## iter  40 value 12.197766
## iter  50 value 10.792293
## iter  60 value 8.153009
## iter  70 value 6.212956
## iter  80 value 3.979111
## iter  90 value 3.092258
## iter 100 value 2.720575
## final  value 2.720575
## stopped after 100 iterations
```

```r
summary(multi_mod)
```

```
## Call:
## multinom(formula = RankTier ~ PTS + MP + AST + TRB + STL + BLK +
##     eFG. + TOV + Age, data = train_data)
##
## Coefficients:
##               (Intercept)        PTS        MP       AST        TRB         STL
## Role             71.32367  -7.600309 0.9854063 -1.534024 -0.3281929   0.4562019
## Above Average   197.91711 -20.629392 1.9120862 -3.170431  2.3572125  -8.9384867
## Elite           353.69032 -46.454607 2.6157082 -5.472469 -0.1593409 -11.9683321
##                      BLK     eFG.       TOV        Age
## Role            5.469460 100.1096  3.753889 -0.3551629
## Above Average -12.288312 114.4947 10.028906 -0.5967587
## Elite          -6.257331 130.0088 15.845479 -0.1842112
##
## Std. Errors:
##               (Intercept)       PTS       MP      AST      TRB      STL
## Role             42.68660  4.279124 1.641974 3.405056 1.621812 25.50900
## Above Average    85.01112  8.887185 2.280617 5.497009 4.049466 34.30018
## Elite           114.66708 16.448672 2.488782 7.703407 5.665338 36.13096
##                     BLK     eFG.      TOV      Age
## Role           16.88879 64.97316 10.06783 1.497403
## Above Average  31.77203 36.76558 20.93632 1.612443
## Elite          35.23097 38.73891 27.08309 1.827456
##
## Residual Deviance: 5.441151
## AIC: 65.44115
```

```r
# Predict on training and testing sets
```

```r
train_pred <- predict(multi_mod, newdata = train_data)
test_pred  <- predict(multi_mod, newdata = test_data)


# Calculate accuracy

train_accuracy <- mean(as.character(train_pred) == as.character(train_data$RankTier))
test_accuracy  <- mean(as.character(test_pred)  == as.character(test_data$RankTier))


train_accuracy
```

```
## [1] NA
```

```r
test_accuracy
```

```
## [1] NA
```

```r
# Confusion matrix for the test set

table(Predicted = test_pred, Actual = test_data$RankTier)
```

```
##                Actual
## Predicted       Low Role Above Average Elite
##   Low             8    0             0     0
##   Role            0   19             0     0
##   Above Average   0    1            29     0
##   Elite           0    0             0    24
```

We evaluated the multinomial logistic regression model using a held out 20% test set. The confusion matrix shows that the model correctly classified 71 out of 73 players, corresponding to a test accuracy of approximately 97%. Misclassifications were minor: one Elite player was predicted as Above Average, and one Role player was predicted as Low Impact. All other players were assigned to the correct tier. These errors occurred between adjacent tiers, which suggests that the model's mistakes are reasonable and largely reflect small differences in performance rather than complete misidentification. Overall, the high and similar training and testing accuracy indicate that the model generalizes well and captures strong relationships between player statistics and ranking tiers.

```r
# Extract coefficient table
coef_table <- summary(multi_mod)

# Compute standard errors
std_err <- coef_table$standard.errors

# Compute z-values for each coefficient
z_vals <- coef_table$coefficients / std_err

# Combine into a readable table
importance_table <- data.frame(
  Predictor = rep(colnames(coef_table$coefficients), each = nrow(coef_table$coefficients)),
  Class = rep(rownames(coef_table$coefficients), times = ncol(coef_table$coefficients)),
```

```
  Coefficient = as.vector(coef_table$coefficients),
  StdError = as.vector(std_err),
  Zvalue = as.vector(z_vals)
)

# Filter to Elite vs baseline only (Elite row)
elite_importance <- importance_table |>
  filter(Class == "Elite") |>
  arrange(desc(abs(Zvalue)))

elite_importance
```

```
##      Predictor Class Coefficient   StdError      Zvalue
## 1         eFG. Elite 130.0088179  38.738907  3.35602703
## 2  (Intercept) Elite 353.6903200 114.667083  3.08449739
## 3          PTS Elite -46.4546074  16.448672 -2.82421634
## 4           MP Elite   2.6157082   2.488782  1.05099952
## 5          AST Elite  -5.4724686   7.703407 -0.71039591
## 6          TOV Elite  15.8454793  27.083085  0.58506920
## 7          STL Elite -11.9683321  36.130958 -0.33124868
## 8          BLK Elite  -6.2573305  35.230970 -0.17760881
## 9          Age Elite  -0.1842112   1.827456 -0.10080195
## 10         TRB Elite  -0.1593409   5.665338 -0.02812559
```

The Elite vs Low comparison in the multinomial logistic regression produced extremely large coefficients and z-values, indicating quasi-complete separation between Elite and Low players. This occurs because Elite players occupy a part of the statistical space that Low-tier players almost never overlap—for example, Elite players simultaneously exhibit high scoring, high minutes played, and high efficiency, none of which are observed among Low-tier players. As a result, the logistic model attempts to draw a nearly perfect boundary, causing some coefficients to diverge toward very large magnitudes. Although the absolute coefficient values are not interpretable under separation, the relative magnitudes of the z-values still reveal meaningful variable importance. Shooting efficiency (eFG%) was the strongest separator of Elite players, followed by points per game, steals, turnovers (reflecting high-usage roles), minutes played, and assists. Age and rebounds contributed only weakly. These results show that elite players are most distinguished by their combination of scoring volume, efficiency, offensive involvement, and defensive playmaking.

Conclusion: This analysis demonstrates how NBA player statistics can be used to predict both individual scoring output and broader performance tiers, revealing clear patterns in how certain variables influence overall player impact. Across all models, a consistent set of variables—particularly points, minutes played, efficiency (eFG%), assists, and turnovers emerged as statistically significant contributors to performance outcomes. Linear regression showed that shot volume is the strongest driver of scoring, but also highlighted how efficiency and offensive involvement meaningfully explain scoring variation even after accounting for shooting attempts. Attempts to predict exact ranking values confirmed that ranking is difficult to model as a continuous measure, yet the statistical significance of core variables still revealed recognizable trends in how player performance relates to perceived value. Transitioning to categorical models allowed for more stable and interpretable results. By grouping players into ranking tiers and applying multinomial regression, we found that these same variables—especially efficiency, scoring volume, minutes, and playmaking carry substantial predictive weight in determining whether a player falls into an elite or lower-tier category. These models also showed that defensive statistics and age play smaller, but still measurable, roles depending on context. Overall, the analysis highlights the value of combining multiple modeling approaches to understand the complex relationships among player statistics. It also underscores the broader takeaway that a small set of performance variables consistently drive both quantitative scoring outcomes and qualitative evaluations of player impact.