Predicting College Basketball Adjusted Defensive Efficiency

Memphis Lau

2023-03-13

Introduction

Those engaged in any sport know the classic saying: "Defense wins championships". This is especially the case in college men's basketball. With March Madness beginning, and everyone scrambling to fill out their brackets and pick a winner, this project aims to explore trends in defensive efficiency in the sport of college basketball.

In simple terms, defense is a team's ability to stop the opposing team from scoring the basketball. Defensive efficiency refers to how many points a team allows per 100 possessions. Ken Pomeroy, an infamous name in college basketball statistics, popularized the measurement of adjusted defensive efficiency, which is a team's defensive efficiency multiplied by the national defensive efficiency and divided by the opposing team's offensive efficiency. In other words, this is the measure of a team's defensive efficiency versus an average opponent. These numbers can be found on kenpom.com.

Historically, in regards to success in the March Madness tournament, adjusted defensive efficiency (or ADJDE) has been a very predictive and important statistic. Kenpom.com has tracked data on every college basketball team since 2002. Every team that has won the national championship has been top 22 in adjusted defensive efficiency that season. The average ranking in ADJDE for these championship-winning teams is 9.1 (https://www.collegebasketballtimes.com/post/men-s-ncaa-tournament-breakdown-by-the-numbers). In the last 10 years, 40 of the 40 Final Four teams have been inside the top 50 in ADJDE, with 17 being in the top 10 (https://www.covers.com/ncaab/march-madness/trends). The main point is that this statistic of adjusted defensive efficiency has major implications in postseason success and can help guide who you pick in your bracket.

Without using kenpom.com, finding a team's adjusted defensive efficiency can be quite challenging. We would need to know the nationwide average, as well as factor in opposing teams' offensive efficiency. **In this project, I aim to find a linear model to predict a college basketball team's adjusted defensive efficiency using a linear model of easily accessible team statistics**.

From Kaggle, I have a dataset containing team statistics from every men's college basketball team in the last 10 seasons. Along with their adjusted defensive efficiency, my dataset also contains a team's offensive and defensive statistics regarding rebounds, shot percentage, turnovers, etc. While irrelevant to my project, my dataset also contains the team's postseason results in a given season. In this project, I will first explore the dataset and pick variables that I want to add to my model. Then, I will create the model using my training set of data, run the model on a test set, check accuracy and model diagnostics, and try to improve the model.

Data Description

```
data <- read.csv('alldataclean.csv')
head(data,3)
```

```
##      TEAM CONF  G  W ADJOE ADJDE BARTHAG EFG_O EFG_D  TOR TORD  ORB  DRB  FTR
## 1 Gonzaga  WCC 32 28 120.3  89.9  0.9662  58.7  43.1 15.7 16.3 29.1 23.4 30.6
## 2 Houston Amer 38 32 116.5  88.5  0.9595  53.1  43.3 16.9 21.7 37.7 28.0 29.0
## 3  Kansas  B12 40 34 119.8  91.3  0.9580  53.8  45.8 17.3 18.1 32.9 28.6 32.3
##   FTRD X2P_O X2P_D X3P_O X3P_D ADJ_T  WAB POSTSEASON SEED YEAR
## 1 22.7  60.4  41.8  37.0  30.5  72.6  6.7        S16    1 2022
## 2 34.7  54.5  43.4  33.8  28.8  63.7  6.2         E8    5 2022
## 3 27.7  53.6  46.4  36.1  29.8  69.1 10.4  Champions    1 2022
```

```
colnames(data)
```

```
##  [1] "TEAM"       "CONF"       "G"          "W"          "ADJOE"
##  [6] "ADJDE"      "BARTHAG"    "EFG_O"      "EFG_D"      "TOR"
## [11] "TORD"       "ORB"        "DRB"        "FTR"        "FTRD"
## [16] "X2P_O"      "X2P_D"      "X3P_O"      "X3P_D"      "ADJ_T"
## [21] "WAB"        "POSTSEASON" "SEED"       "YEAR"
```

```
nrow(data)
```

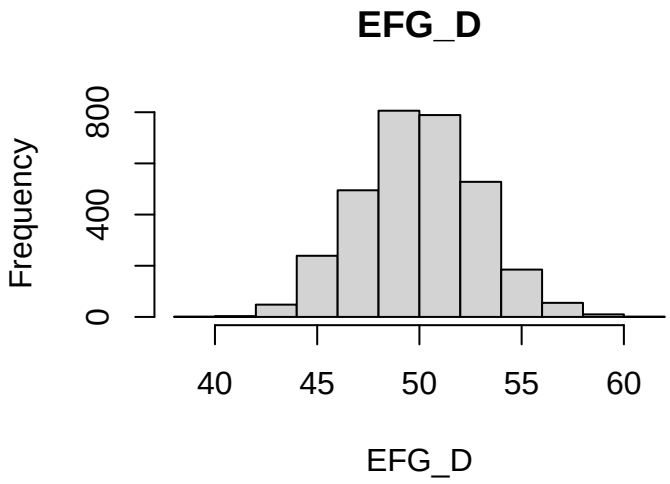
```
## [1] 3160
```

Our data has 3160 rows of 24 different variables. We pick "ADJDE" to be our response variable. When picking predictor variables, we want columns with a D (defense) in the name. For our initial model, we will choose "EFG_D", "TORD", "DRB", "FTRD", and "ADJ_T".

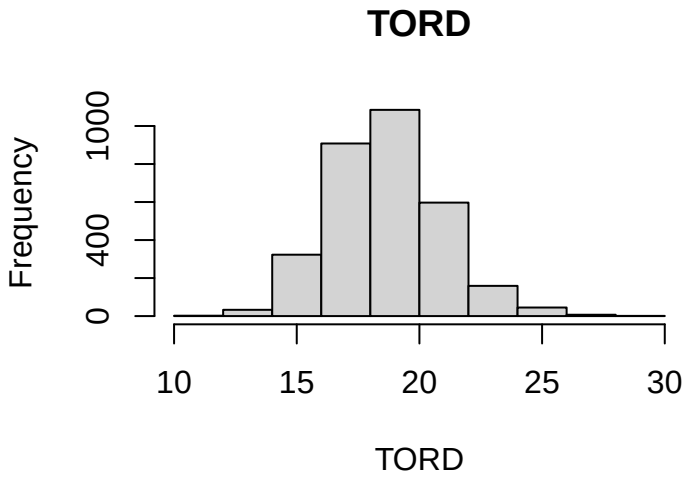
- EFD_D : Effective Field Goal Percentage Allowed. This percentage is calculated by a formula that weighs three point shots and free throws differently than two point shots.
- TORD : Turnover Percentage Committed. This is the rate at which a team causes the opposing team to turn the ball over.
- DRB: Offensive Rebound Rate Allowed. This measures a team's ability to grab rebounds after the opposing team misses.
- FTRD : Free Throw Rate Allowed. This is how often a team's opponent shoots free throws against them.
- ADJ_T : Adjusted Tempo. This is an estimate of the amount of possessions per 40 minutes a team would have against an average team.

Distribtion of the Variables

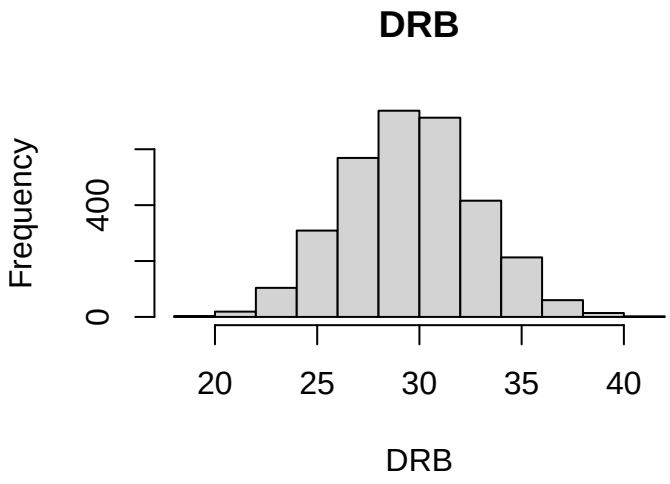
```
hist(data$EFG_D, main = "EFG_D", xlab = 'EFG_D')
```



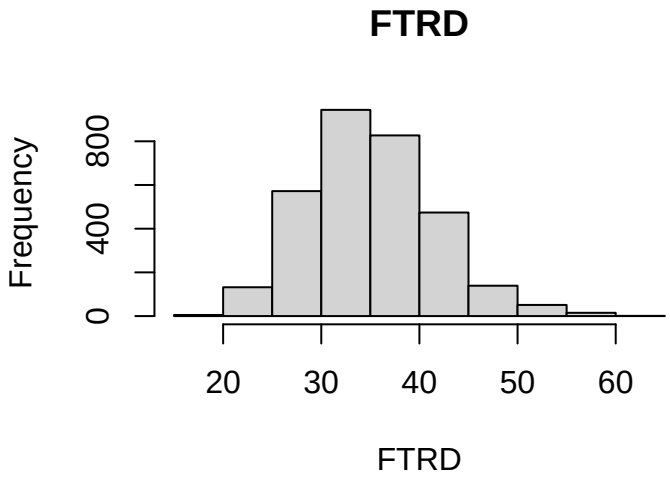

```
hist(data$TORD, main = "TORD", xlab = 'TORD')
```



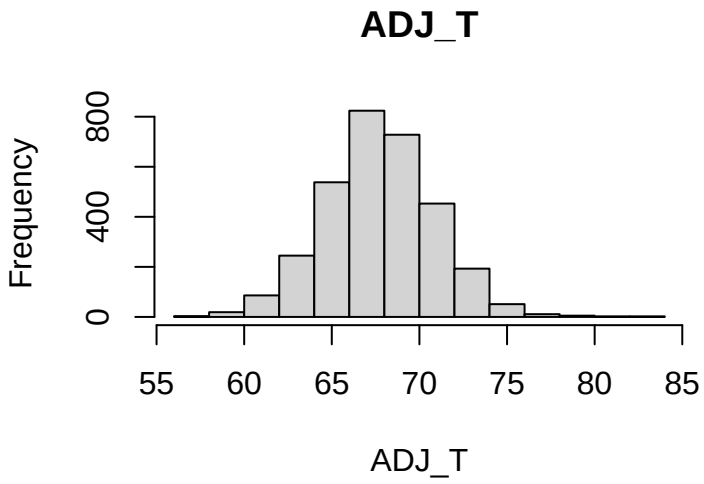

```
hist(data$DRB, main = "DRB", xlab = 'DRB')
```




```
hist(data$FTRD, main = "FTRD", xlab = 'FTRD')
```



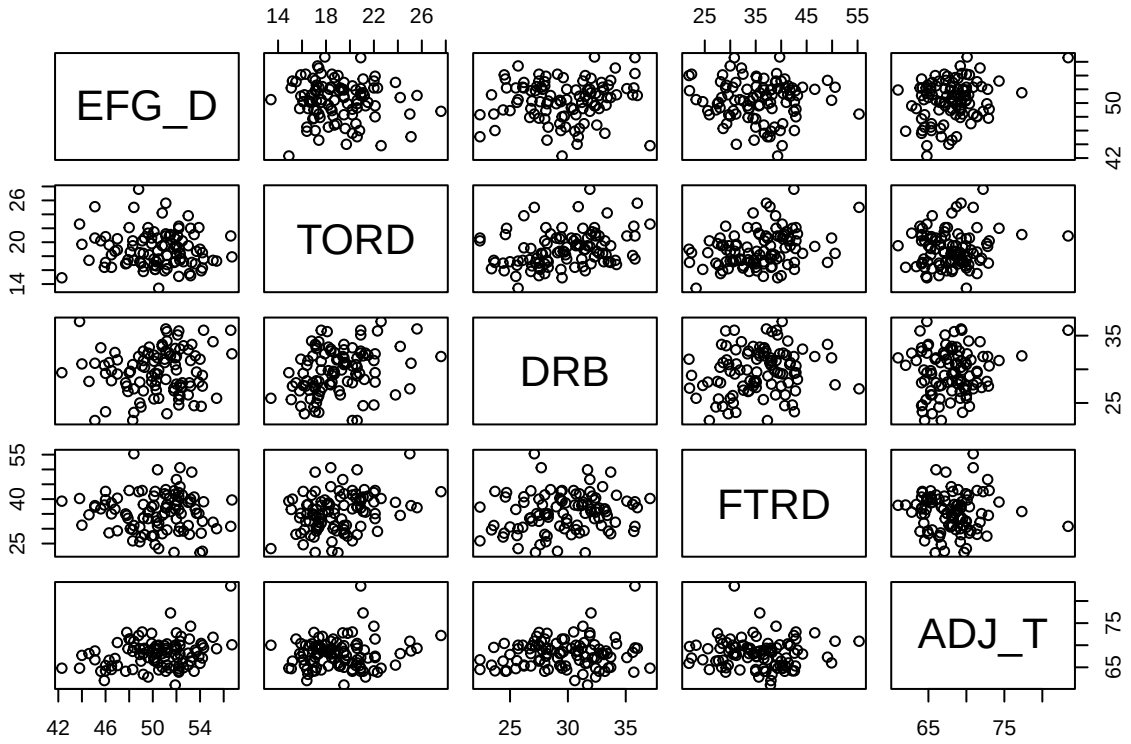

```
hist(data$ADJ_T, main = "ADJ_T", xlab = 'ADJ_T')
```



All of our chosen predictor variables are relatively normally distributed.

Relationships Among Variables

```
# We look at a sample of 100 random teams since looking at 3160 datapoints is too much.
n <- 100
set.seed(1)
sample_idx <- sample(nrow(data), size = n, replace = FALSE)
data_sample <- data[sample_idx, ]
pairs(data_sample[, c('EFG_D','TORD', 'DRB', 'FTRD', 'ADJ_T')])
```




```
cor(data_sample[,c('EFG_D', 'TORD', 'DRB', 'FTRD', 'ADJ_T')])
```

```
##             EFG_D        TORD        DRB        FTRD       ADJ_T
## EFG_D  1.00000000 -0.07530907 0.09757293 -0.04812982  0.28291112
## TORD  -0.07530907  1.00000000 0.30246507  0.32021456  0.08431561
## DRB    0.09757293  0.30246507 1.00000000  0.17183085  0.11817416
## FTRD  -0.04812982  0.32021456 0.17183085  1.00000000 -0.03450480
## ADJ_T  0.28291112  0.08431561 0.11817416 -0.03450480  1.00000000
```

We see that none of the predictor variables are strongly linearly correlated with one another. This is a good sign, as to avoid multicollinearity when we create our model.

Creating our Initial Model

```
#Splitting Data into Train and Test Set
train_split <- nrow(data) * 0.8
train_index <- sample(nrow(data), size = train_split, replace = FALSE)
train <- data[train_index, ]
test <- data[-train_index, ]
```

```
#Creating model
model <- lm(ADJDE ~ EFG_D + TORD + DRB + FTRD + ADJ_T, data = train)
model
```

```
##
## Call:
## lm(formula = ADJDE ~ EFG_D + TORD + DRB + FTRD + ADJ_T, data = train)
##
## Coefficients:
## (Intercept)        EFG_D         TORD          DRB         FTRD        ADJ_T
##    19.18176      1.57759     -1.09226      0.58987      0.18306      0.02144
```

The formula our model gives us is:

$ADJDE = 19.18176 + 1.57759EFG\_D$ - $1.09226TORD + 0.58987DRB + 0.18306FTRD + 0.02144ADJ\_T$.

From these numbers alone, we see that the allowed field goal percentage (EFG_D) has the largest effect in determining adjusted defensive efficiency since it has the highest regression coefficient. To put it into context, for every 1 percentage point in opponent effective field goal percentage, adjusted defensive efficiency is expected to increase by 1.578 percent, holding the other variables constant. We now look at the summary table of this model to see t-tests and more.

```
summary(model)
```

```
##
## Call:
## lm(formula = ADJDE ~ EFG_D + TORD + DRB + FTRD + ADJ_T, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6684 -2.0742  0.1726  2.1513 12.4605
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.18176    1.67513  11.451   <2e-16 ***
## EFG_D        1.57759    0.02187  72.131   <2e-16 ***
## TORD        -1.09226    0.02974 -36.723   <2e-16 ***
## DRB          0.58987    0.02039  28.927   <2e-16 ***
## FTRD         0.18306    0.01033  17.727   <2e-16 ***
## ADJ_T        0.02144    0.02014   1.065    0.287
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.026 on 2522 degrees of freedom
## Multiple R-squared:  0.7789, Adjusted R-squared:  0.7784
## F-statistic:  1777 on 5 and 2522 DF,  p-value: < 2.2e-16
```

Our initial model has a R-squared value of 0.779. This means that ~78% of the variance in ADJDE is explained by our chosen variables. We see that the t-test is statistically significant (< 0.05) in all predictor variables except adjusted tempo.

```
anova(model)
```

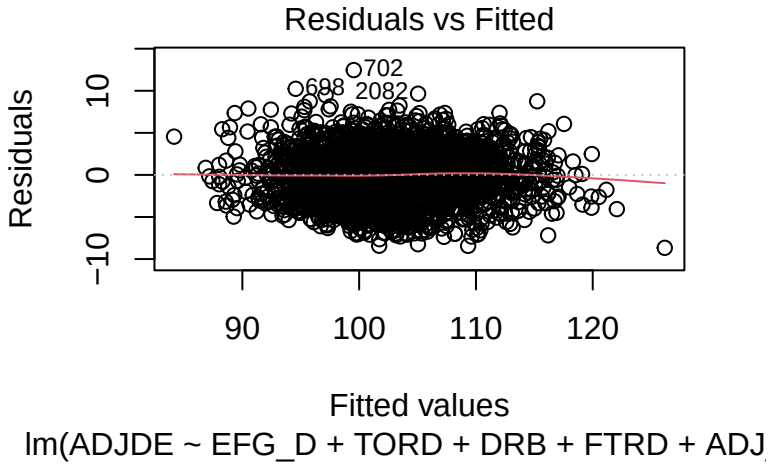
```
## Analysis of Variance Table
##
## Response: ADJDE
##             Df Sum Sq Mean Sq   F value Pr(>F)
```

```
## EFG_D        1  62656   62656 6843.5503 <2e-16 ***
## TORD         1   5175    5175  565.2195 <2e-16 ***
## DRB          1  10615   10615 1159.4006 <2e-16 ***
## FTRD         1   2875    2875  314.0399 <2e-16 ***
## ADJ_T        1     10      10    1.1332 0.2872
## Residuals 2522  23090       9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F-tests tell us the same story. All predictor variables are statistically significant except for adjusted tempo (ADJ_T).

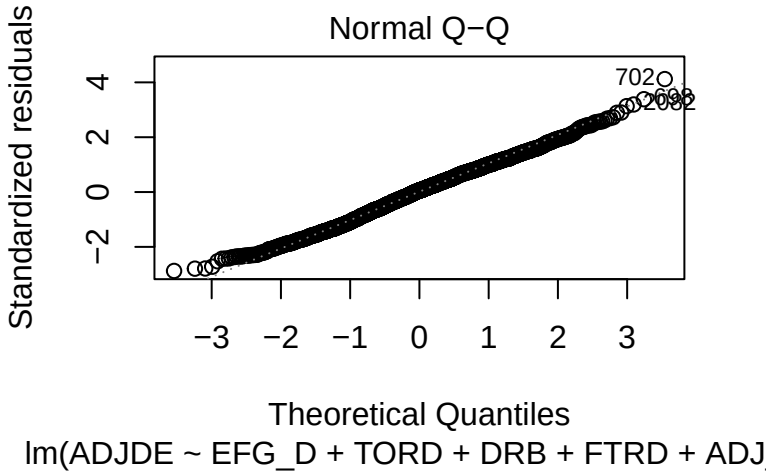
Now, we will look at diagnostic plots for this first model.

```
plot(model,1)
```



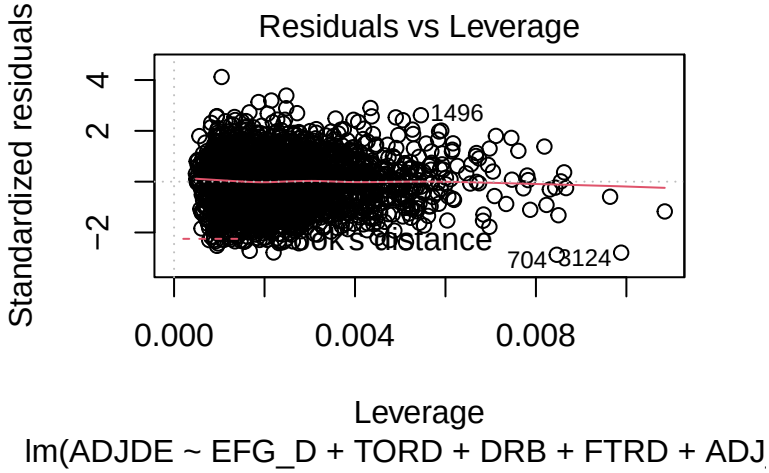
The residuals vs fitted values plot looks good. The red line is relatively straight and centered around 0. This means the relationship is linear, and the average of the errors is 0.

```
plot(model, 2)
```



The QQ-plot is also good. The points follow a straight line, suggesting that the errors are normally distributed.

```
plot(model, 5)
```



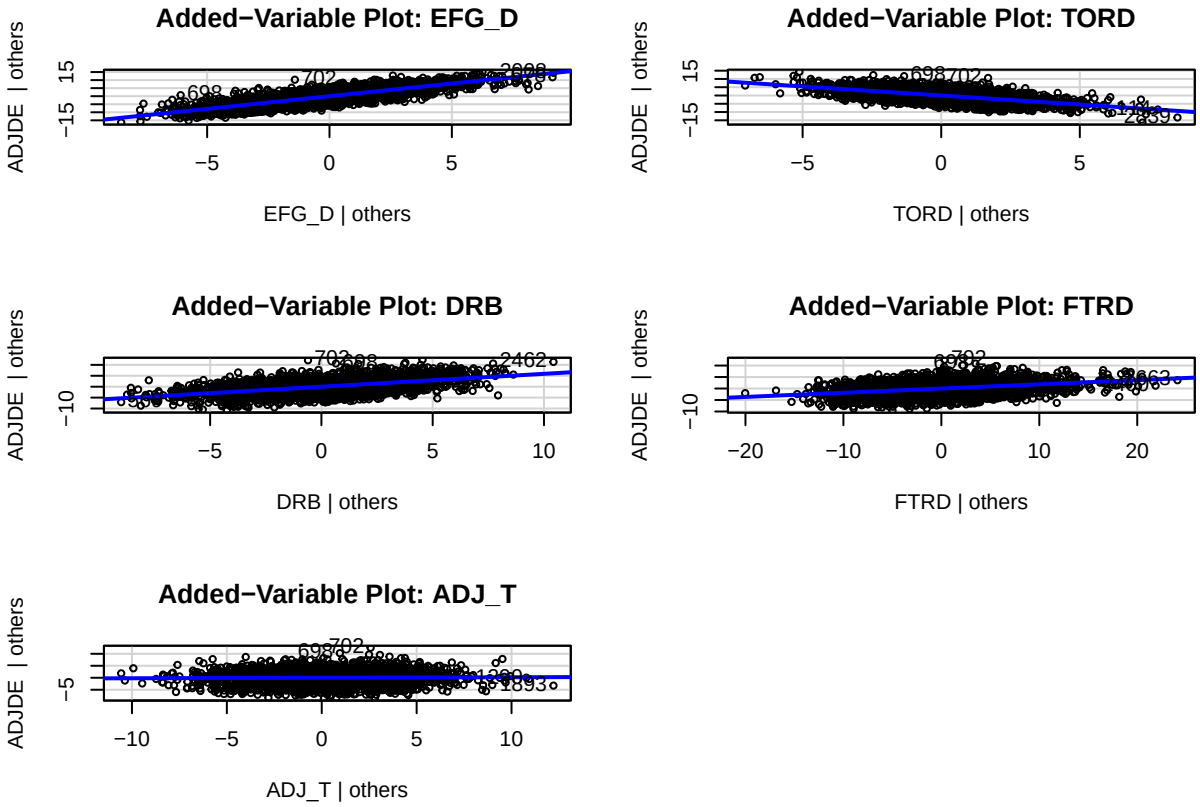
Though it is difficult to see, there are relatively few amount of points with standardized residuals above 2 or below -2. To see if there are any high leverage points, we calculate the threshold with the formula 2 * (p + 1) / n = (2 * 6) / 2528 = 0.0047. We see that there is actually a high amount of leverage points.

Now, we look at added variable plots.

```
par(mfrow=c(3,2))
library(car)
```

```
## Loading required package: carData
```

```
avPlot(model, variable = 'EFG_D', ask = FALSE)
avPlot(model, variable = 'TORD', ask = FALSE)
avPlot(model, variable = 'DRB', ask = FALSE)
avPlot(model, variable = 'FTRD', ask = FALSE)
avPlot(model, variable = 'ADJ_T', ask = FALSE)
```



These added variable plots reinforce the idea that we should consider a model without the adjusted tempo variable, since that plot is flat and not showing any trend. This means that adjusted tempo does not have much influence on the defensive efficiency of a team.

Second Model: Reduced Model

```
# Creating reduced model
model2 <- lm(ADJDE ~ EFG_D + TORD + DRB + FTRD, data = train)
summary(model2)
```

```
##
## Call:
## lm(formula = ADJDE ~ EFG_D + TORD + DRB + FTRD, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6660 -2.0655  0.1598  2.1503 12.5158
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 20.39954    1.22371   16.67   <2e-16 ***
## EFG_D        1.58286    0.02131   74.29   <2e-16 ***
## TORD        -1.09328    0.02973  -36.78   <2e-16 ***
## DRB          0.58959    0.02039   28.92   <2e-16 ***
## FTRD         0.18300    0.01033   17.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.026 on 2523 degrees of freedom
## Multiple R-squared:  0.7788, Adjusted R-squared:  0.7784
## F-statistic:  2220 on 4 and 2523 DF,  p-value: < 2.2e-16
```

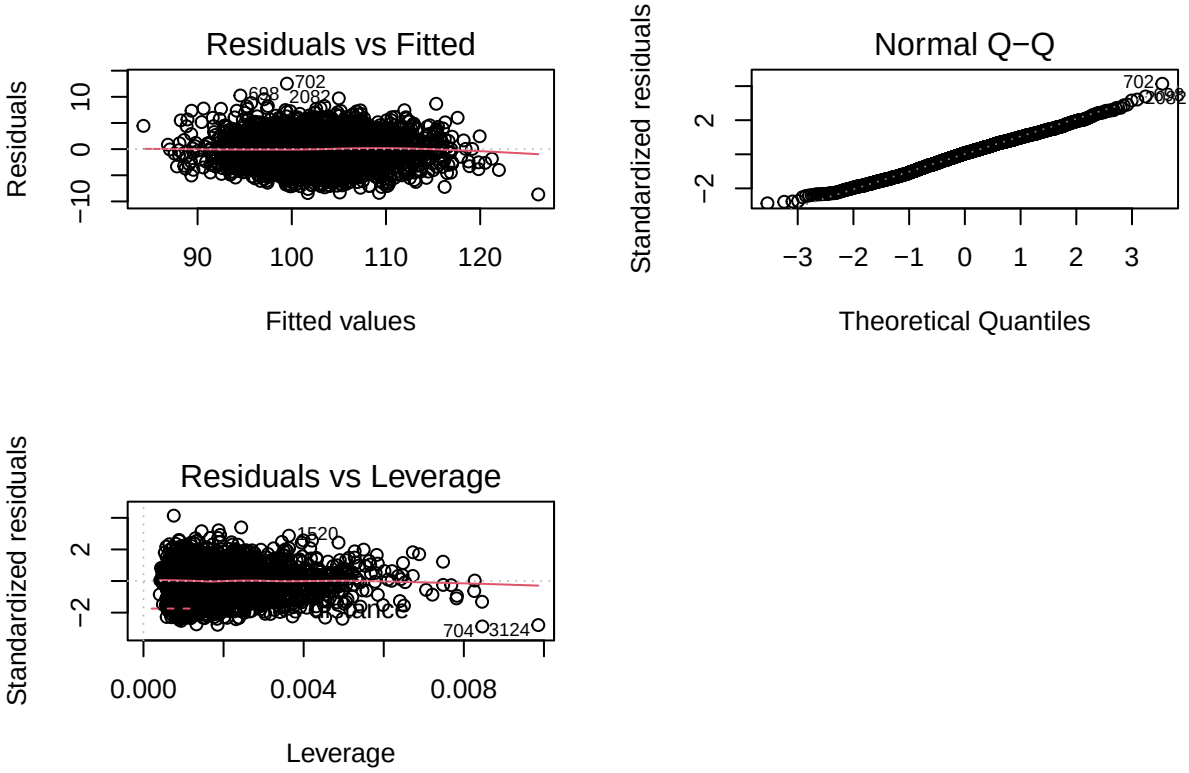
In this model, all of the predictor variables are significant.

```
#anova test for our two models
anova(model2, model)
```

```
## Analysis of Variance Table
##
## Model 1: ADJDE ~ EFG_D + TORD + DRB + FTRD
## Model 2: ADJDE ~ EFG_D + TORD + DRB + FTRD + ADJ_T
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1   2523 23101
## 2   2522 23090  1    10.375 1.1332 0.2872
```

The p-value of this anova test is not statistically significant (above 0.05). Thus, we do not have sufficient evidence to conclude that the full model is better than our reduced model. We can conclude this model without the ADJ_T variable is better.

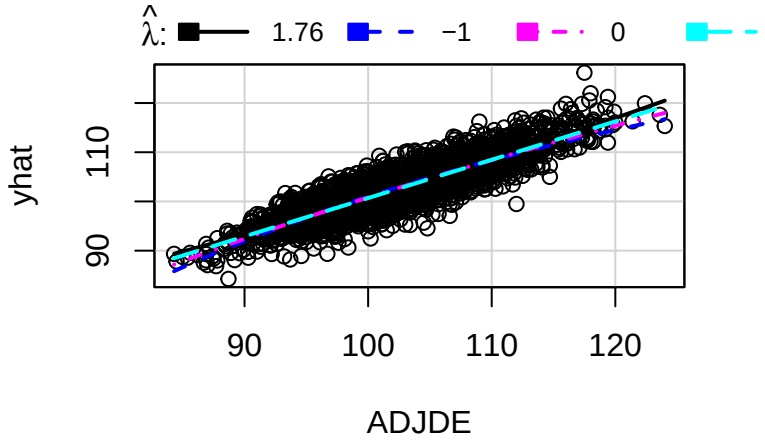
```
par(mfrow=c(2,2))
plot(model2, 1)
plot(model2, 2)
plot(model2, 5)
```



Our diagnostic plots suggest our model is valid (for the same reasons as explained in the first model). However, the model only has an R-squared value of 0.77. In hopes to improve this, we will try a transformation.

Third Model: Transformation

```
inverseResponsePlot(model2, key = TRUE)
```




```
##      lambda      RSS
## 1  1.755519 17933.10
## 2 -1.000000 18691.72
## 3  0.000000 18241.55
## 4  1.000000 17990.24
```

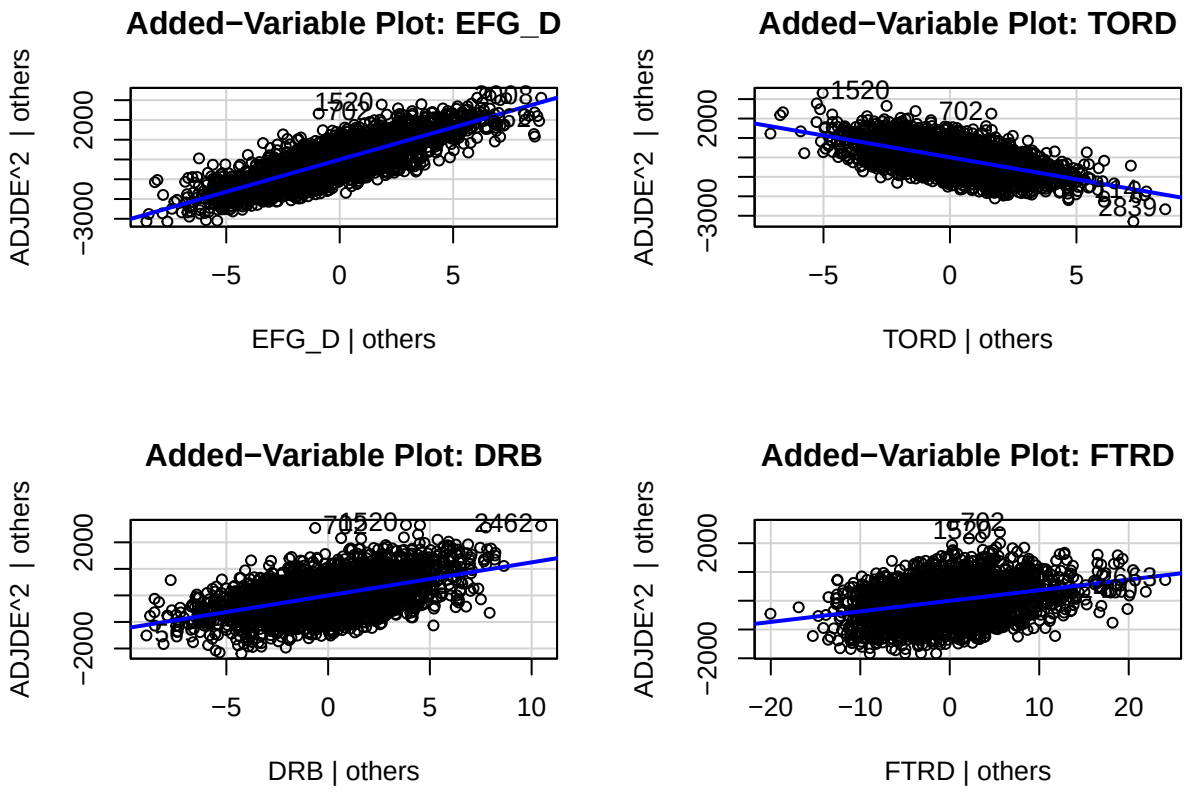
The Box-cox transformation method tells us that a better estimation for our model would come if we squared our Y variable.

```
model3 <- lm(ADJDE**2 ~ EFG_D + TORD + DRB + FTRD, data = train)
summary(model3)
```

```
##
## Call:
## lm(formula = ADJDE^2 ~ EFG_D + TORD + DRB + FTRD, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1777.33  -426.26    23.82   434.53  2620.53
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6419.176    252.271  -25.45   <2e-16 ***
## EFG_D         326.100      4.392   74.24   <2e-16 ***
## TORD         -225.780      6.129  -36.84   <2e-16 ***
## DRB           124.592      4.203   29.64   <2e-16 ***
## FTRD           36.842      2.129   17.31   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 623.8 on 2523 degrees of freedom
## Multiple R-squared:  0.7795, Adjusted R-squared:  0.7791
## F-statistic:  2229 on 4 and 2523 DF,  p-value: < 2.2e-16
```

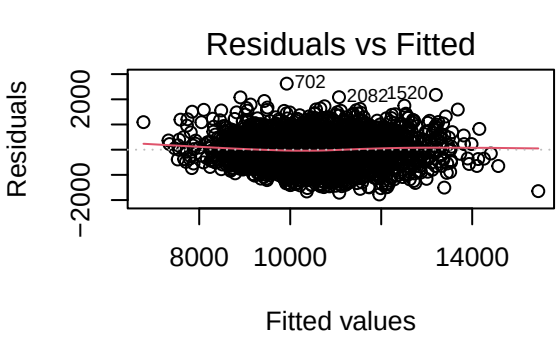
The r-squared value is very very slightly higher.

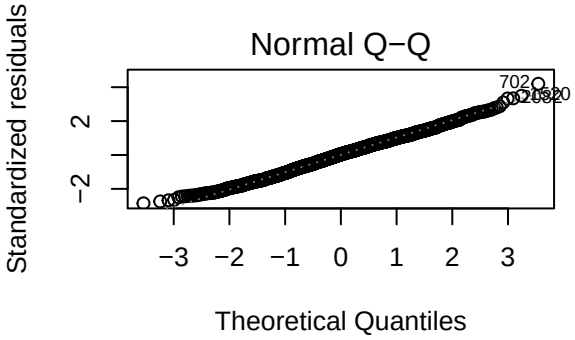
```
par(mfrow=c(2,2))
avPlot(model3, variable = 'EFG_D', ask = FALSE)
avPlot(model3, variable = 'TORD', ask = FALSE)
avPlot(model3, variable = 'DRB', ask = FALSE)
avPlot(model3, variable = 'FTRD', ask = FALSE)
```

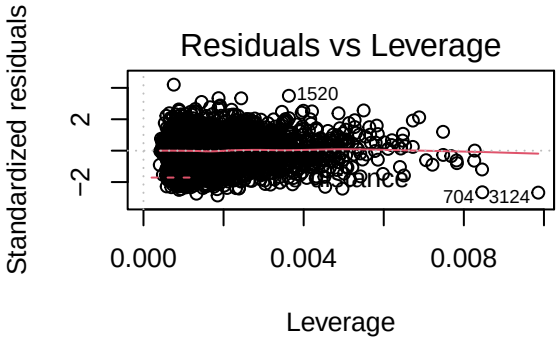


The added variable plots are strong with this transformed, reduced model. Each of the predictor variables have a linear relationship with the squared value of adjusted defensive efficiency, given the other predictor variables.

```
par(mfrow=c(2,2))
plot(model3, 1)
plot(model3, 2)
plot(model3, 5)
```







All the diagnostic plots also suggest this is a good model. The residuals vs fitted line is straight and centered at 0. The QQ-plot is linear, meaning the errors are normally distributed. The residuals vs leverage graph shows that majority of the points do not have high leverage.

Lastly, we check for multicollinearity with the vifs.

```
vif(model3)
```

```
##    EFG_D     TORD      DRB     FTRD
## 1.028944 1.206898 1.188867 1.216453
```

All the predictor variables have vifs under 5, so there is no issue with multicollinearity here.

Checking against Test Set

```
differences <- 0
for (x in 1:nrow(test)) {
  actual <- test[x, 'ADJDE']
  predicted <- sqrt(-6419.18 + 326.1*test[x,'EFG_D'] - 225.78*test[x, 'TORD'] + 124.59*test[x, 'DRB'] +
  difference <- (actual - predicted)^2
  differences <- differences + difference
}
```

```
differences
```

```
## [1] 6452.806
```

This is the residual sum of squares of the test set. Considering there are 632 observations in the test dataset, a RSS of 6400 is not bad, especially knowing our model only has an R-squared value of 0.78.

Discussion

In summary, I wanted to find a linear model that could predict a college basketball team's adjusted defensive efficiency. In the introduction, I explained why this statistic is so important to a team's success. My dataset had every single Division I college basketball team in the last 10 years, along with their basic team statistics. First, I created a model with the predictor variables: opponents' field goal percentage (EFG_D), opponents' turnover rate (TORD), defensive rebound rate (DRB), opponents' free throw rate (FTRD), and the team's adjusted tempo (ADJ_T). Running diagnostic tests and looking at summary tables told me that the adjusted tempo predictor variable was not statistically significant to predicting adjusted defensive efficiency.

I then created a reduced model, removing the ADJ_T variable. This model had good diagnostic plots, but still only had an R-squared value of 0.777. In hopes to create a better model, I performed a Box-cox transformation. I found that squaring my response variable ADJDE would be best. The model only very slightly improved. Testing our model on our test set, we see that the RSS is not too large.

The limitations of my analysis come from the limitations to the variables I have in my dataset. While statistics like opponent turnover rate and opponent free throw rate are interesting and valuable, this model would've benefited from measures like number of steals and blocks per game. Opponent strength could also be a important factor, as our response variable, adjusted defensive efficiency, measures a team's defensive efficiency against an average Division I team. In addition, another slight limitation I had was that there were too many data points in my dataset for my visualizations to be clear. I could not make out any clear patterns and relationships between my variables. At the same time, I felt it was important to have every team represented in my model. In the future, my model can be improved with access to more defensive statistics, as well as maybe more transformations to my predictor variables based on patterns I could see.

Sources

- **Dataset**: https://www.kaggle.com/datasets/alecbensman/college-basketball-march-madness-data
- https://www.covers.com/ncaab/march-madness/trends
- https://www.collegebasketballtimes.com/post/men-s-ncaa-tournament-breakdown-by-the-numbers