SCORE Project: NCAA Men's Basketball Statistical Analysis

CDTs Hayden Coffman and Sean O'Hara

2024-05-01

Learning Goals

1. Students will be able to determine the strength of the correlation between offensive rebounds, blocks, turnovers in winning percentage.
2. Students will be able to generate a chart to visualize the relationship between different variables that can influence win percentage.
3. Students will become confident in running basic linear regression in R, building a linear model in R, and generating plots using ggplot in RStudio to apply these concepts to different data sets.
4. Students will validate their linear regression models using the four assumptions of linear regression.

Introduction

Can we determine a positive correlation between an NCAA Men's basketball team's statistics to determine their win percentage?

This module aims at teaching the student how to successfully analyze a season of college basketball statistics by using linear regression to determine a team's win percentage using a number of different variables. The student will learn how to plot in R using GGplot2 and build a linear model using the $lm()$ function in R.

Data

The data comes from sports-reference.com and contains all statistics from the 2022-23 Mens College Basketball season. The data contains 363 NCAA Division I teams that competed in the 2022-23 season.

Variables contained in this dataset include wins, losses, conference wins and losses, points scored for and against, minutes played, field goals and field goal percentage, 3 point baskets and percentage, offensive and total rebounds, assists, steals, blocks, turnovers, and personal fouls.

Data Exploration

To begin, we will load the necessary R libraries and load the data set into R from its existing CSV file. Next we will look at the first 10 rows of our NCAA Basketball data set.

```
# Load the necessary R libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- 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
```

```r
library(readr)
library(knitr)

# Read in the CSV File
NCAA_Basketball_Stats <- read_csv("ncaa_basketball_stats.xls.csv", show_col_types = FALSE)

# View the first 10 rows of the data frame as a table
# We will only select the variables that we care about for this exercise
NCAA_Basketball_Stats %>%
  select(School, win_loss_percent, orb, tov, blk) %>%
  head(10) %>%
  kable()
```

School	win_loss_percent	orb	tov	blk
Abilene Christian	0.43	307	379	71
Air Force	0.44	213	391	127
Akron	0.67	335	370	87
Alabama	0.84	484	512	189
Alabama A&M	0.46	334	498	133
Alabama State	0.26	387	414	110
Albany (NY)	0.26	317	414	51
Alcorn State	0.56	384	404	80
American	0.53	259	448	126
Appalachian State	0.50	288	352	146

Data Exploration

Next, we will produce summary statistics for the first 10 rows of our NCAA Basketball data set.

```r
# Produce a table of summary statistics for the first 10 rows
NCAA_Basketball_Stats %>%
  select(School, win_loss_percent, orb, tov, blk) %>%
  summary() %>%
  head(10) %>%
  kable()
```

School	win_loss_percent	orb	tov	blk
Length:363	Min. :0.0900	Min. :194.0	Min. :281.0	Min. : 37

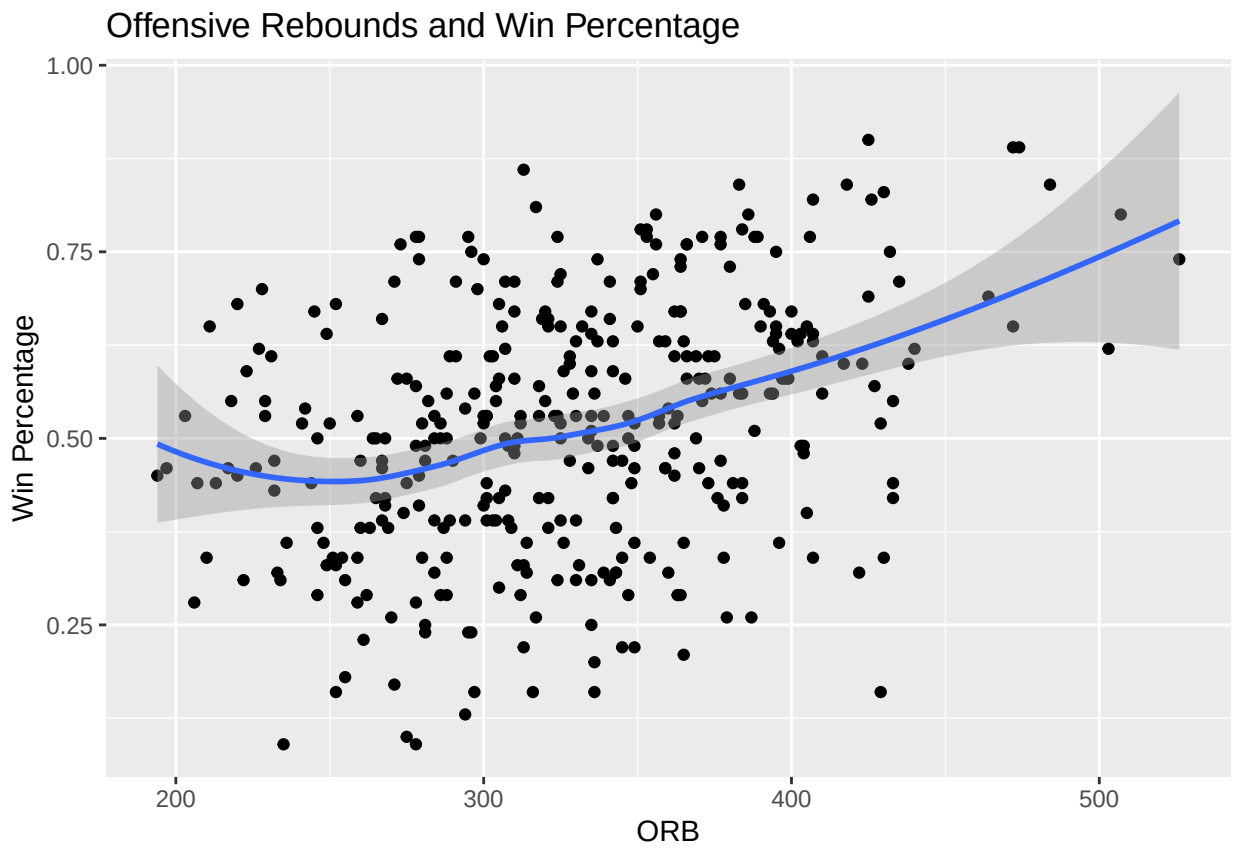
School	win_loss_percent	orb	tov	blk
Class :character	1st Qu.:0.3900	1st Qu.:284.0	1st Qu.:379.0	1st Qu.: 83
Mode :character	Median :0.5200	Median :326.0	Median :406.0	Median :101
NA	Mean :0.5151	Mean :328.0	Mean :409.7	Mean :106
NA	3rd Qu.:0.6350	3rd Qu.:367.5	3rd Qu.:437.0	3rd Qu.:125
NA	Max. :0.9000	Max. :526.0	Max. :573.0	Max. :242

Activities

Part A: Plot Offensive Rebounds and Win Percentage using the "ggPlot2" library in R. Do you notice any trends?

```
NCAA_Basketball_Stats %>%
  ggplot(aes(x = orb, y = win_loss_percent)) +
    geom_point() +
    labs(title = "Offensive Rebounds and Win Percentage",
       x = "ORB",
       y = "Win Percentage") +
  geom_smooth()
```

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



You should notice that there is an increase in Win Percentage as a team increases their number of Offensive Rebounds.

Part B: Create a linear model using the lm function in R Studio to fit a linear model to our data frame. We will use this function to carry out linear regression. This will help us analyze the relationship between offensive rebounds and winning percentage.

In this first example, we will apply the linear regression model below to our data. Let Y_i be a random variable for the number of wins team i gets in a season. B_0, the intercept, represents the base number of wins for any team, and B_1 is the coefficient for how many more wins a team gets for each offensive rebound. Let ϵ be the missing variables that can cause variation in the response variable, winning percentage.

$$Y_i = \beta_0 + \beta_1 \text{orb}_i + \epsilon_i$$

```
# Create a linear model to predict Win/Loss Percentage using Offensive Rebounds and produce a summary
linfit <- lm(win_loss_percent ~ orb, data = NCAA_Basketball_Stats)
summary(linfit)
```

```
##
## Call:
## lm(formula = win_loss_percent ~ orb, data = NCAA_Basketball_Stats)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45551 -0.10033  0.00854  0.10758  0.35990
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.1887052  0.0453268   4.163 3.93e-05 ***
## orb         0.0009949  0.0001359   7.320 1.62e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1558 on 361 degrees of freedom
## Multiple R-squared:  0.1293, Adjusted R-squared:  0.1268
## F-statistic: 53.59 on 1 and 361 DF,  p-value: 1.621e-12
```

Take note of the p-values of the variables, the adjusted R squared value, and the p value of the model itself. B_0 and B_1 had p values $3.93e-05$ and $1.62e-12$ respectively. You can locate these p values under the $Pr(>|t|)$ column in the Coefficients table. These are both extremely small. In simple terms, p values represent the probability that these results occurred by chance alone. In this situation, the p values tell us that for the intercept, it is likely that teams will reach a certain number of wins, and more offensive rebounds equates to more wins in a season.

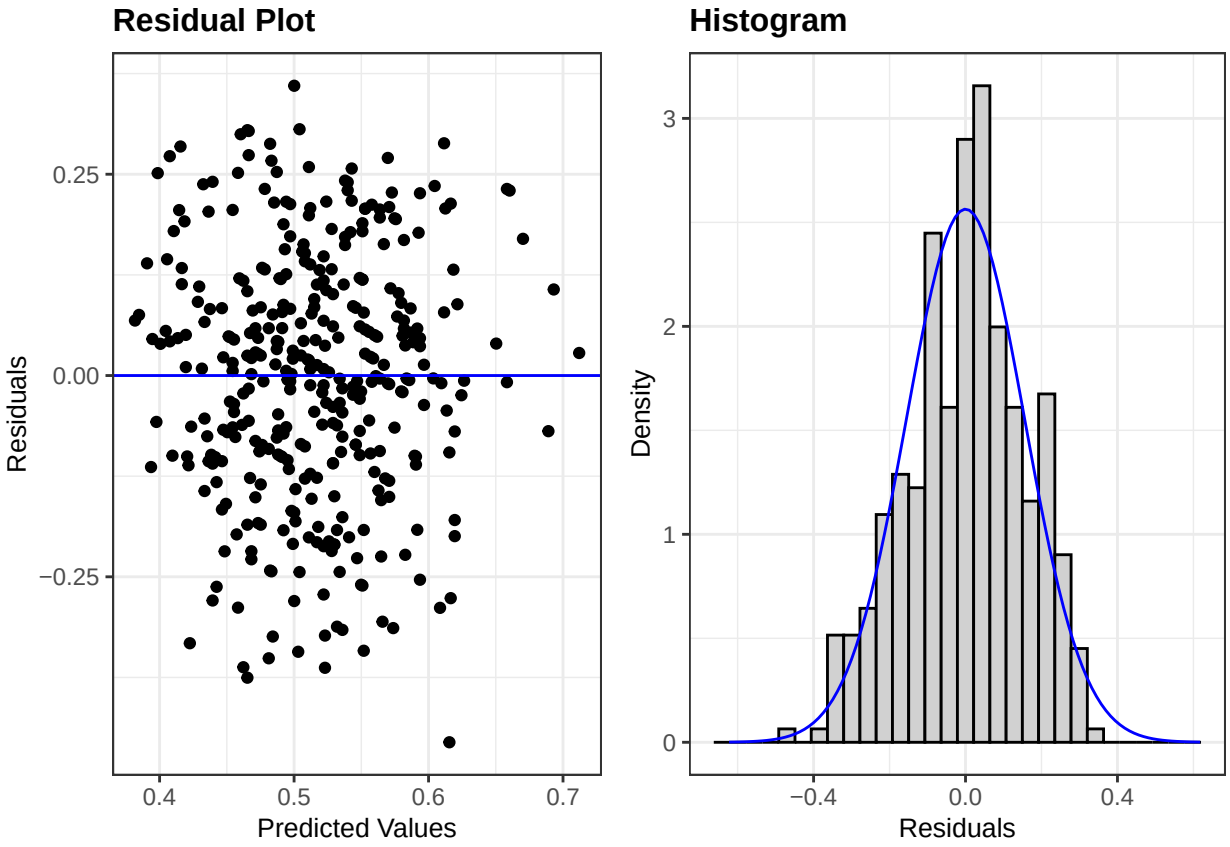
Moving to the final block of text, we can find the Adjusted R-squared value of 0.1268. This tells us the proportion of the variance in the response variable, win percentage, accounted for by the model. Using our intuition, 12.68 percent is not much. Ideally, we would want over 80 percent. However, the p value of the model is $1.621e-12$. This is significant, so, moving forward, we should look to add more variables to our linear regression to increase the Adjusted R squared value. The reference in Method B also gives insight into our linear regression results.

Part C: Validate the use of linear regression using the "ggResidpanel" R library.

- A residual is the difference between the observed and the predicted value.

```
# Load the 'ggResidpanel' library
library(ggResidpanel)

# Create the Residual and Histogram Plots using the 'linfit' model data from part B
resid_panel(linfit, plots = c('resid', 'hist'))
```



Part D: Use the Four Assumptions (L.I.N.E.) to Validate the Linear Regression Model that we created.

- Linearity of Residuals
- Independence of Residuals
- Normal Distribution of Residuals
- Equal Variance of Residuals

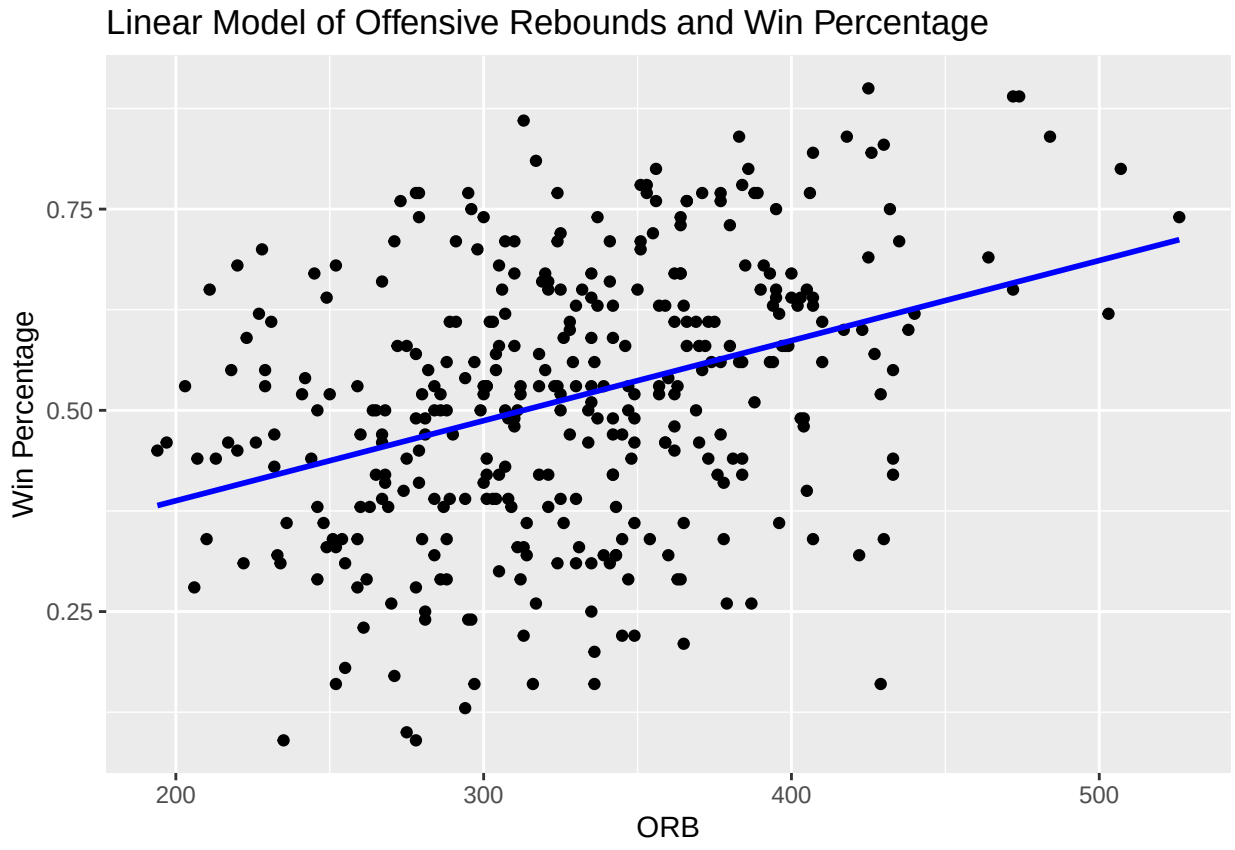
In this example, the scatterplot satisfies both the linearity and equal variance of the residuals. It satisfies linearity because the residuals are not forming any curves or patterns, and it satisfies equal variance because there are no dots fanning out in a triangular fashion. Finally, the histogram verifies normality. The residuals form a normal distribution, and they are not skewed in any direction. This data set is cross-sectional, meaning we gather the data for each team at one time at the end of the season, instead of over a period of time. In this case, we assume the independence assumption is met.

Part E: Overlay the linear model onto the data using the geom_smooth() function in "ggplot2".

```
# Create the new graph with the linear model
linfit <- ggplot(NCAA_Basketball_Stats, aes(x = orb, y = win_loss_percent)) +
  geom_point() +
  scale_x_continuous() +
  scale_y_continuous() +
  geom_smooth(method = "lm", se = FALSE, color = 'blue') +
  labs(title = "Linear Model of Offensive Rebounds and Win Percentage",
       x = "ORB",
       y = "Win Percentage")

# Render the graph
linfit
```

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



You should notice the increase in win percentage as the number of offensive rebounds increases.

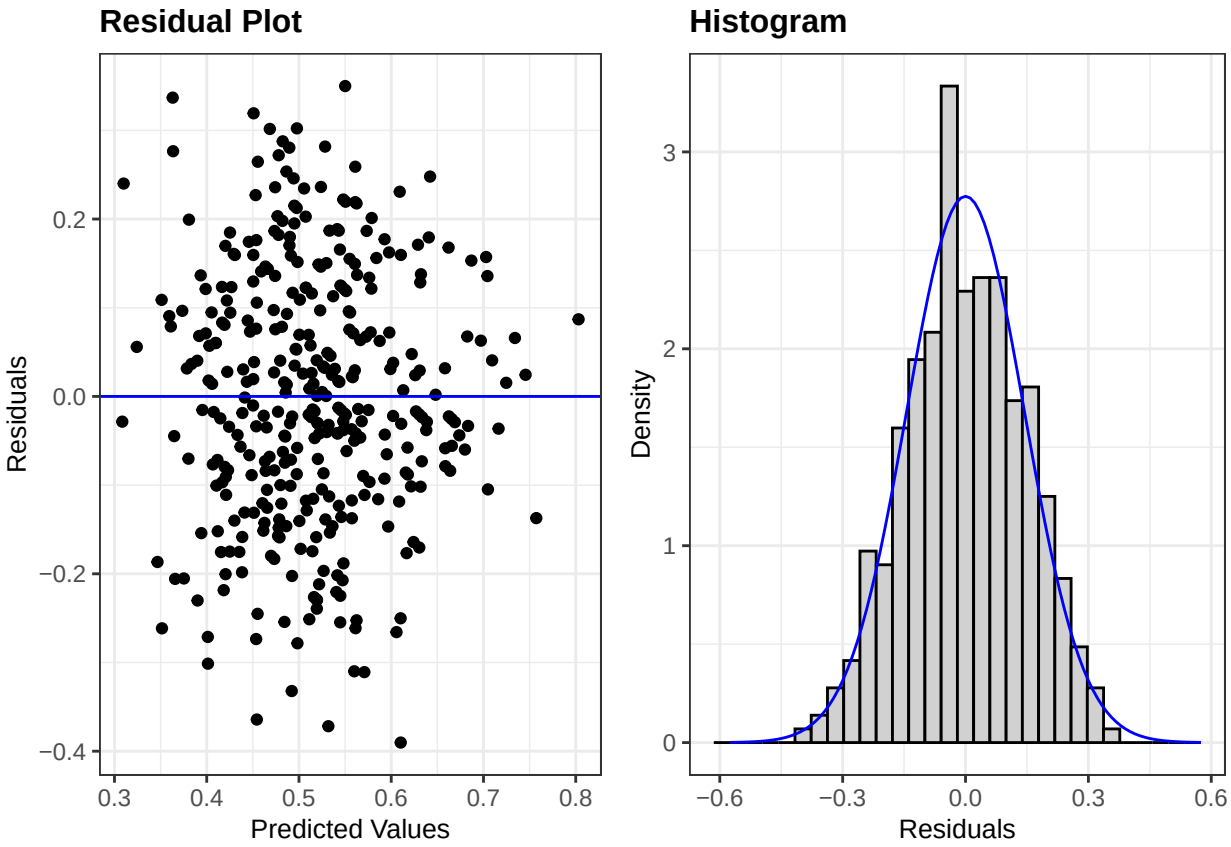
Part F: Add more variables to the linear model to predict winning percentage. How did the adjusted R-squared value change after adjusting for more variables? Are these additional variables and the model still significant? Is the linear model still applicable? What do you notice about the coefficients for turnovers and offensive rebounds?

$$Y_i = \beta_0 + \beta_1 \text{orb}_i + \beta_2 \text{tov}_i + \beta_3 \text{blk}_i + \epsilon_i$$

```
# Create a new linear model with the additional variables: Turnovers and Blocks
linfit <- lm(win_loss_percent ~ orb + tov + blk, data = NCAA_Basketball_Stats)
summary(linfit)
```

```
##
## Call:
## lm(formula = win_loss_percent ~ orb + tov + blk, data = NCAA_Basketball_Stats)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39028 -0.09838 -0.01275  0.10850  0.34993
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.3919850  0.0707477   5.541 5.84e-08 ***
## orb          0.0008969  0.0001423   6.304 8.52e-10 ***
## tov         -0.0008390  0.0001751  -4.792 2.43e-06 ***
## blk          0.0016285  0.0002525   6.450 3.63e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1444 on 359 degrees of freedom
## Multiple R-squared:  0.2566, Adjusted R-squared:  0.2504
## F-statistic:  41.3 on 3 and 359 DF,  p-value: < 2.2e-16
```

```
# Create the new residual panel of the histogram and residual plots
resid_panel(linfit, plots = c('resid', 'hist'))
```



Analysis:

Using the same validity conditions as Part C, the residuals meet all of the assumptions for linear regression. There is linearity because there are no curves in scatterplot. The independence is cross sectional. For normality, the histogram has normal distribution, and for equal variance, the scatterplot has no fanning or triangle shape. In addition, after accounting for turnovers and blocks, our adjusted R squared value has doubled to 0.2566. This is a better fit of the data, but still not great. Now look at the coefficients for offensive rebounds and turnovers. They are nearly exactly opposite. This makes sense because an offensive rebound gives your team a new possession and a turnovers costs you a possession. All of the coefficients and the model are significant because they have low p values. In addition to offensive rebounds, accounting for turnovers and blocks improves the effectiveness of the linear regression model.

Conclusion

1. In conclusion, students should be comfortable creating basic plots using ggplot. In addition, they will be able to create, interpret, validate, and graph linear models. This allows the students to identify linear trends between variables.

2. Students were able to find a correlation between offensive rebounds and win percentage. Also, they were able to build a better model that accounts for offensive rebounds, blocks, and turnovers that improves the Adjusted R Squared value of the model.

3. Students will be able to identify the strength of their models utilizing the R squared value. They will also be able to understand and interpret p values.

4. For future projects, students can utilize logistic regression to simulate a binary response variable. For example, students can use offensive rebounds to determine if an NCAA basketball team will win their conference.

Methods and More Instructional Content

For more instructional content on topics that were covered in this module, see the links below.

1. Linear Regression https://www.khanacademy.org/math/statistics-probability/describing-relationships-quantitative-data/introduction-to-trend-lines/a/linear-regression-review. This provides a basic overview of linear regression, how to fit a line to data using equations, and understanding linear regression mathematically.
2. Analyzing Linear Regression in R https://libguides.princeton.edu/c.php?g=1315411&p=9671574#s-lg-box-wrapper-36293217. This discusses creating and interpreting a linear model in R.
3. Creating Graphics Using GGplot https://r4ds.had.co.nz/data-visualisation.html. Build a strong understanding of ggplot2 and how to graph data.
4. Validating Linear Models https://blog.uwgb.edu/bansalg/statistics-data-analytics/linear-regression/what-are-the-four-assumptions-of-linear-regression/. Understand when linear regression is applicable when analyzing a data set and building a model.