Regression2

2025-01-14

```r
# Load necessary libraries
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```r
library(dplyr)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

```r
library(stats)
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.3
```

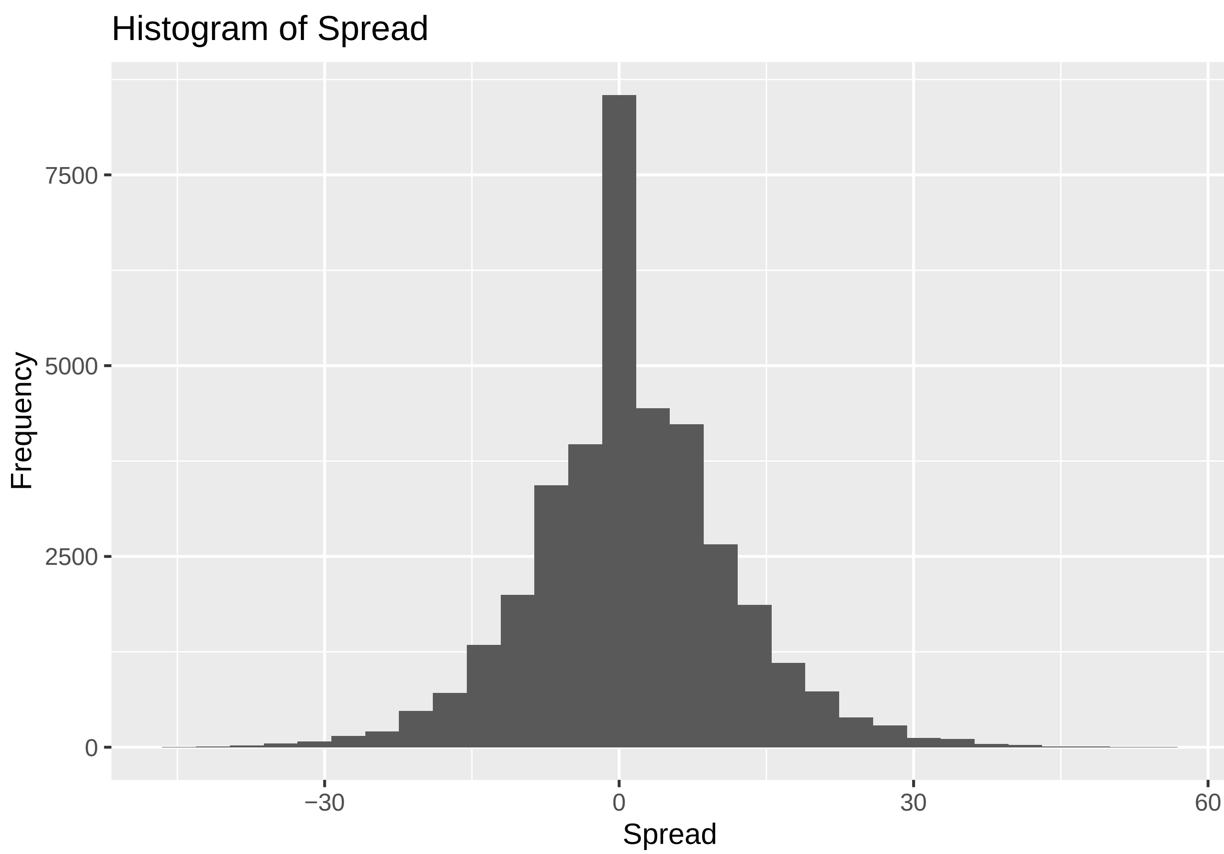
```r
library(glmnet)
getwd()
```

```
## [1] "/Users/brynhaden/Desktop/538"
```

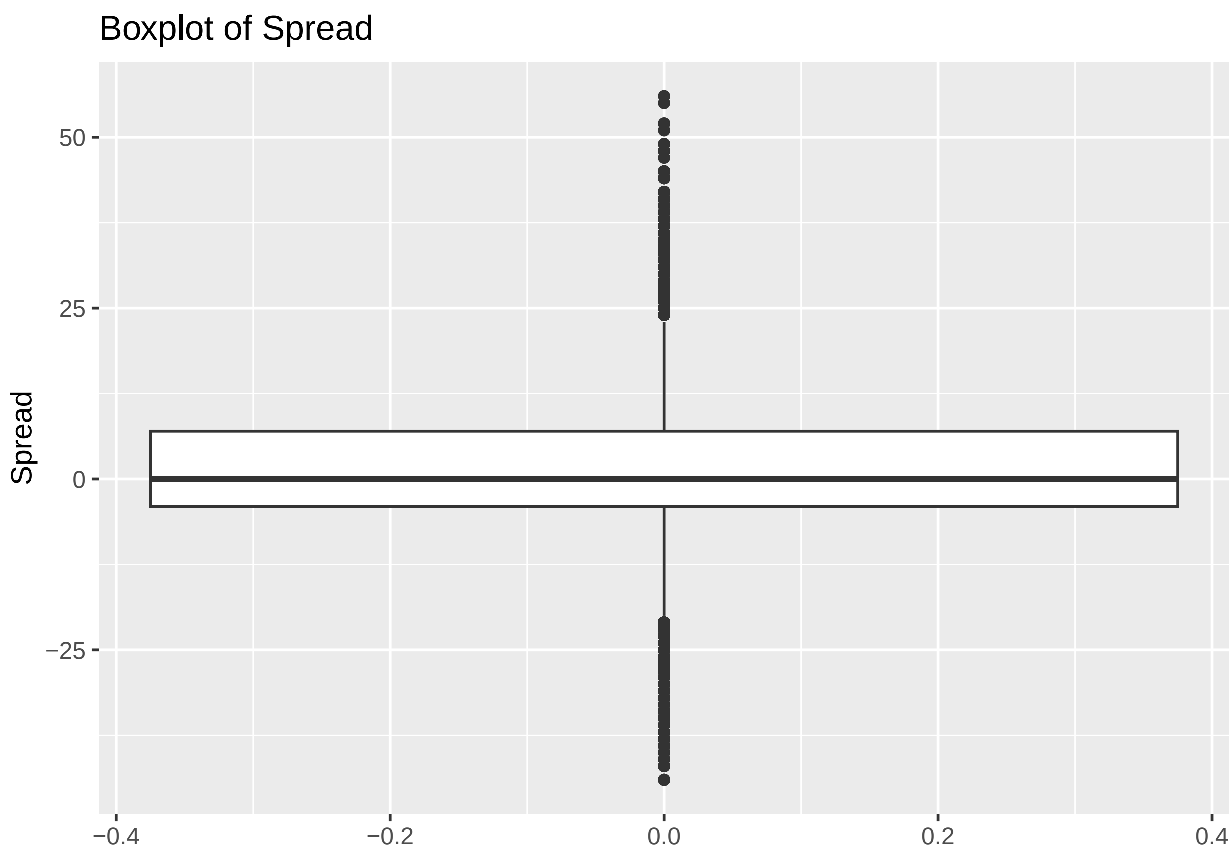
```r
setwd("~/Desktop/538")
df <- read.csv("kicking.csv")
View(df)

# Step 1: Create the Spread Variable
df <- df %>%
  mutate(spread = home_score_pre - visiting_score_pre)

# Step 2: Create a Histogram of the Spread Variable
ggplot(df, aes(x = spread)) +
  geom_histogram(bins = 30) +
  ggtitle('Histogram of Spread') +
  xlab('Spread') +
  ylab('Frequency')
```




```
# Just to be safe I am creating a Boxplot of the Spread Variable too
ggplot(df, aes(y = spread)) +
  geom_boxplot() +
  ggtitle('Boxplot of Spread') +
  ylab('Spread')
```




```r
# Step 3: Fit the Initial Linear Regression Model
model <- lm(spread ~ home_team + away_team + quarter + yardline, data = df)

# Extract and display coefficients and p-values
coefficients_table <- summary(model)$coefficients[, c("Estimate", "Pr(>|t|)")]
print(coefficients_table)
```

```
##                                        Estimate     Pr(>|t|)
## (Intercept)                       -0.1467707015 7.421768e-01
## home_teamAtlanta Falcons           3.3153462600 2.571834e-15
## home_teamBaltimore Ravens          4.0513259582 2.783177e-21
## home_teamBuffalo Bills             0.8216035348 5.514046e-02
## home_teamCarolina Panthers         1.2007129117 4.802833e-03
## home_teamChicago Bears             1.0826154804 1.087406e-02
## home_teamCincinnati Bengals        1.8368493157 1.455515e-05
## home_teamCleveland Browns         -0.2039829380 6.378471e-01
## home_teamDallas Cowboys            1.3575577486 1.125494e-03
## home_teamDenver Broncos            1.4676812283 5.219834e-04
## home_teamDetroit Lions            -0.3575447194 3.952222e-01
## home_teamGreen Bay Packers         4.3057994426 7.796176e-25
## home_teamHouston Texans            0.0006363789 9.988003e-01
## home_teamIndianapolis Colts        2.5555807527 1.455059e-09
## home_teamJacksonville Jaguars     -0.4104981651 3.432528e-01
## home_teamKansas City Chiefs        2.4408586996 9.369693e-09
## home_teamLas Vegas Raiders        -1.2760912000 2.257609e-03
## home_teamLos Angeles Chargers      3.7028944616 1.784196e-18
```

```
## home_teamLos Angeles Rams         -0.1648383984 6.923566e-01
## home_teamMiami Dolphins           -0.6364412297 1.369593e-01
## home_teamMinnesota Vikings         2.0470688048 1.383208e-06
## home_teamNew England Patriots      5.0514722100 6.037601e-33
## home_teamNew Orleans Saints        1.9504290512 1.892020e-06
## home_teamNew York Giants           0.4793384456 2.536390e-01
## home_teamNew York Jets             1.2181191803 4.388199e-03
## home_teamPhiladelphia Eagles       2.8806211019 9.993343e-12
## home_teamPittsburgh Steelers       3.7922070029 1.043125e-18
## home_teamSan Francisco 49ers       0.1441967748 7.324988e-01
## home_teamSeattle Seahawks          2.6658106557 2.595157e-10
## home_teamTampa Bay Buccaneers     -0.5848700987 1.678282e-01
## home_teamTennessee Titans         -0.3386457041 4.235102e-01
## home_teamWashington Football Team -0.6695593945 1.161001e-01
## away_teamAtlanta Falcons          -1.1647513002 5.932296e-03
## away_teamBaltimore Ravens         -1.9594019685 4.973033e-06
## away_teamBuffalo Bills            -0.5424944856 2.060707e-01
## away_teamCarolina Panthers        -2.2780757491 6.623338e-08
## away_teamChicago Bears            -0.5756351379 1.775073e-01
## away_teamCincinnati Bengals       -1.3435553853 1.732558e-03
## away_teamCleveland Browns          0.4930226658 2.503026e-01
## away_teamDallas Cowboys           -1.5231639107 3.523363e-04
## away_teamDenver Broncos           -1.5893336075 1.766630e-04
## away_teamDetroit Lions             0.6539494557 1.237408e-01
## away_teamGreen Bay Packers        -2.4271612962 6.721818e-09
## away_teamHouston Texans            0.4460923450 2.952691e-01
## away_teamIndianapolis Colts       -1.3622538714 1.170796e-03
## away_teamJacksonville Jaguars      1.6752676623 9.210827e-05
## away_teamKansas City Chiefs       -1.3353029051 1.647318e-03
## away_teamLas Vegas Raiders         1.1625368643 6.363651e-03
## away_teamLos Angeles Chargers     -1.9475226654 4.108997e-06
## away_teamLos Angeles Rams          0.0778199119 8.537825e-01
## away_teamMiami Dolphins           -0.1941760010 6.505887e-01
## away_teamMinnesota Vikings        -1.2238643035 3.914646e-03
## away_teamNew England Patriots     -5.2192020353 6.668133e-35
## away_teamNew Orleans Saints       -2.2923913372 3.662590e-08
## away_teamNew York Giants          -1.0420390637 1.361436e-02
## away_teamNew York Jets             0.5284080878 2.203669e-01
## away_teamPhiladelphia Eagles      -1.1033823417 8.516234e-03
## away_teamPittsburgh Steelers      -2.8379081652 5.683650e-11
## away_teamSan Francisco 49ers       0.4026791998 3.387076e-01
## away_teamSeattle Seahawks         -0.7541441894 7.394891e-02
## away_teamTampa Bay Buccaneers      0.0576870778 8.905376e-01
## away_teamTennessee Titans          0.2323420249 5.867627e-01
## away_teamWashington Football Team -0.2630409005 5.373263e-01
## quarter                            0.4311610285 1.220363e-19
## yardline                          -0.0143358468 4.402449e-03
```

```r
# Extract and display R-Squared and Adjusted R-Squared
r_squared <- summary(model)$r.squared
adjusted_r_squared <- summary(model)$adj.r.squared
cat("R-Squared:", r_squared, "\n")
```

```
## R-Squared: 0.04939167
```

```r
cat("Adjusted R-Squared:", adjusted_r_squared, "\n")
```

```
## Adjusted R-Squared: 0.04770315
```

```r
# Step 4: Fit the Interaction Model
model_interaction <- lm(spread ~ home_team * away_team + quarter + yardline, data = df)

# Extract and display R-Squared and Adjusted R-Squared for the interaction model
r_squared_interaction <- summary(model_interaction)$r.squared
adjusted_r_squared_interaction <- summary(model_interaction)$adj.r.squared
cat("Interaction Model R-Squared:", r_squared_interaction, "\n")
```

```
## Interaction Model R-Squared: 0.1831133
```

```r
cat("Interaction Model Adjusted R-Squared:", adjusted_r_squared_interaction, "\n")
```

```
## Interaction Model Adjusted R-Squared: 0.1600044
```

```r
# Step 5: Split Data and Evaluate Models
set.seed(42)
trainIndex <- createDataPartition(df$spread, p = 0.8, list = FALSE)

# Split the data
train_df <- df[trainIndex, ]
test_df <- df[-trainIndex, ]

# Ensure all levels are present in the training and test data
train_df <- train_df %>%
  mutate(home_team = factor(home_team, levels = unique(df$home_team)),
         away_team = factor(away_team, levels = unique(df$away_team)),
         quarter = factor(quarter, levels = unique(df$quarter)),
         yardline = factor(yardline, levels = unique(df$yardline)))

test_df <- test_df %>%
  mutate(home_team = factor(home_team, levels = levels(train_df$home_team)),
         away_team = factor(away_team, levels = levels(train_df$away_team)),
         quarter = factor(quarter, levels = levels(train_df$quarter)),
         yardline = factor(yardline, levels = levels(train_df$yardline)))

# Remove rows in test_df with levels not present in train_df
test_df <- test_df %>%
  filter(home_team %in% levels(train_df$home_team),
         away_team %in% levels(train_df$away_team),
         quarter %in% levels(train_df$quarter),
         yardline %in% levels(train_df$yardline))

# Without Interaction Model
model_train <- lm(spread ~ home_team + away_team + quarter + yardline, data = train_df)

# With Interaction Model
model_train_interaction <- lm(spread ~ home_team * away_team + quarter + yardline, data = train_df)
```

```r
# Predict on the test data
predictions <- predict(model_train, newdata = test_df)
predictions_interaction <- predict(model_train_interaction, newdata = test_df)
```

```
## Warning in predict.lm(model_train_interaction, newdata = test_df): prediction
## from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```r
# Calculate RMSE and MAD only if there are no NA values
if (sum(is.na(predictions)) == 0 && sum(is.na(predictions_interaction)) == 0) {
  rmse_without_interaction <- sqrt(mean((test_df$spread - predictions)^2))
  mad_without_interaction <- mean(abs(test_df$spread - predictions))

  rmse_with_interaction <- sqrt(mean((test_df$spread - predictions_interaction)^2))
  mad_with_interaction <- mean(abs(test_df$spread - predictions_interaction))

  # Display RMSE and MAD
  results_table <- data.frame(
    Model = c("Without Interaction", "With Interaction"),
    RMSE = c(rmse_without_interaction, rmse_with_interaction),
    MAD = c(mad_without_interaction, mad_with_interaction)
  )
  print(results_table)
} else {
  print("There are NA values in the predictions.")
}
```

```
##                 Model     RMSE      MAD
## 1 Without Interaction 9.829160 7.349435
## 2    With Interaction 9.346047 7.129127
```

```r
# Subset the data for only field goals
field_goals_df <- df %>%
  filter(play_type == "Field Goal")

# Filter kickers with more than 100 field goals
field_goals_df <- field_goals_df %>%
  group_by(kicker_name) %>%
  filter(n() > 100) %>%
  ungroup()

# Check the number of observations
if (nrow(field_goals_df) != 14052) {
  stop("The number of observations is not 14,052 after filtering.")
}

# Ensure the spread variable is included
field_goals_df <- field_goals_df %>%
  mutate(spread = home_score_pre - visiting_score_pre)

# Fit a logistic regression model
logistic_model <- glm(scored ~ yardline + quarter + kicker_name + spread,
                      data = field_goals_df,
```

```r
                       family = binomial)

# Display coefficients and p-values
coefficients_table_logistic <- summary(logistic_model)$coefficients[, c("Estimate", "Pr(>|z|)")]
print(coefficients_table_logistic)
```

```
##                               Estimate      Pr(>|z|)
## (Intercept)               4.493252e+00 7.563903e-131
## yardline                 -1.140623e-01 6.573491e-278
## quarter                   3.016088e-02  2.099537e-01
## kicker_nameB.Cundiff     -8.544758e-01  5.602106e-04
## kicker_nameB.McManus     -2.793965e-01  2.994869e-01
## kicker_nameB.Walsh       -6.116980e-02  8.206974e-01
## kicker_nameC.Barth       -1.438467e-01  5.819694e-01
## kicker_nameC.Boswell      1.033958e-01  7.443060e-01
## kicker_nameC.Catanzaro   -2.676394e-01  3.620957e-01
## kicker_nameC.Parkey      -4.037003e-01  1.871049e-01
## kicker_nameC.Santos      -4.802242e-01  9.614530e-02
## kicker_nameC.Sturgis     -3.713322e-01  1.842565e-01
## kicker_nameD.Akers       -4.118093e-01  6.380830e-02
## kicker_nameD.Bailey       2.099743e-01  4.076081e-01
## kicker_nameD.Carpenter   -8.813673e-02  7.118360e-01
## kicker_nameD.Defense     -1.874542e+01  9.108622e-01
## kicker_nameD.Hopkins     -6.719339e-02  8.237347e-01
## kicker_nameG.Gano        -7.769174e-02  7.498605e-01
## kicker_nameG.Hartley     -6.734298e-01  3.625355e-02
## kicker_nameG.Zuerlein    -1.717773e-03  9.944259e-01
## kicker_nameH.Butker       3.092648e-01  4.158806e-01
## kicker_nameJ.Brown        2.784272e-02  9.041270e-01
## kicker_nameJ.Carney      -3.744693e-01  2.218952e-01
## kicker_nameJ.Elam        -3.884159e-01  1.602102e-01
## kicker_nameJ.Feely       -2.513110e-01  2.865366e-01
## kicker_nameJ.Hanson      -9.846057e-04  9.968666e-01
## kicker_nameJ.Kasay       -8.600615e-02  7.367911e-01
## kicker_nameJ.Lambo        2.453023e-01  4.539476e-01
## kicker_nameJ.Myers        9.244333e-02  7.584206e-01
## kicker_nameJ.Nedney      -5.633498e-03  9.853555e-01
## kicker_nameJ.Reed        -6.009706e-01  2.089215e-02
## kicker_nameJ.Scobee      -3.658232e-01  1.033846e-01
## kicker_nameJ.Tucker       8.491973e-01  2.271681e-03
## kicker_nameJ.Wilkins     -3.781134e-01  1.998276e-01
## kicker_nameK.Brown       -6.174116e-01  1.821911e-02
## kicker_nameK.Forbath      2.554693e-01  4.310787e-01
## kicker_nameL.Tynes       -6.550409e-01  8.284922e-03
## kicker_nameM.Bryant       5.260091e-02  8.121796e-01
## kicker_nameM.Crosby      -3.027348e-01  1.541723e-01
## kicker_nameM.Nugent      -5.253062e-01  1.814561e-02
## kicker_nameM.Prater      -4.082761e-02  8.537580e-01
## kicker_nameM.Stover      -2.387765e-01  4.212284e-01
## kicker_nameN.Folk        -3.935573e-01  7.918627e-02
## kicker_nameN.Kaeding     -1.491391e-01  5.902584e-01
## kicker_nameN.Novak       -3.585118e-01  1.538954e-01
## kicker_nameN.Nullified   -1.938398e+01  9.155388e-01
```

```
## kicker_nameN.Rackers    -1.642357e-01  5.049405e-01
## kicker_nameO.Mare       -6.939895e-01  4.975649e-03
## kicker_nameO.Offense    -4.406103e+00  1.283895e-52
## kicker_nameP.Dawson     -8.516912e-02  6.974804e-01
## kicker_nameR.Bironas     4.005512e-03  9.869670e-01
## kicker_nameR.Bullock    -7.528326e-02  7.893902e-01
## kicker_nameR.Gould       1.169594e-01  6.043778e-01
## kicker_nameR.Lindell    -4.638521e-01  5.043213e-02
## kicker_nameR.Longwell   -2.064971e-02  9.399198e-01
## kicker_nameR.Succop     -2.572878e-01  2.693386e-01
## kicker_nameS.Gostkowski -1.276979e-01  5.612677e-01
## kicker_nameS.Graham     -2.048627e-01  4.134084e-01
## kicker_nameS.Hauschka    1.293589e-01  5.980029e-01
## kicker_nameS.Janikowski -1.657580e-01  4.239367e-01
## kicker_nameS.Suisham    -4.281831e-01  7.726610e-02
## kicker_nameW.Lutz        3.345980e-01  3.185878e-01
## spread                  -2.903243e-04  9.187716e-01
```

```r
# Predict probabilities
predicted_probabilities <- predict(logistic_model, type = "response")

# Convert probabilities to binary outcomes (0 or 1)
predicted_outcomes <- ifelse(predicted_probabilities > 0.5, 1, 0)

# Create a confusion matrix
confusion_matrix <- table(Predicted = predicted_outcomes, Actual = field_goals_df$scored)

# Print the confusion matrix
print(confusion_matrix)
```

```
##          Actual
## Predicted     0     1
##         0   592   100
##         1  1816 11544
```

```r
# Create an empty column for predictions
field_goals_df$predicted <- NA

# Create a fold column
set.seed(42)
field_goals_df$fold <- sample(1:10, nrow(field_goals_df), replace = TRUE)

# Perform 10-Fold Cross Validation
for (i in 1:10) {
  # Fit the model on training data (fold != i)
  train_data <- field_goals_df %>% filter(fold != i)
  test_data <- field_goals_df %>% filter(fold == i)

  logistic_model_cv <- glm(scored ~ yardline + quarter + kicker_name + spread,
                           data = train_data,
                           family = binomial)

  # Predict on test data (fold == i)
```

```r
  test_data$predicted <- predict(logistic_model_cv, newdata = test_data, type = "response")

  # Convert probabilities to binary outcomes (0 or 1)
  test_data$predicted <- ifelse(test_data$predicted > 0.5, 1, 0)

  # Save predictions
  field_goals_df$predicted[field_goals_df$fold == i] <- test_data$predicted
}

# Ensure there are no NA values in the predictions
sum(is.na(field_goals_df$predicted))  # Should be 0
```

```
## [1] 0
```

```r
# Create a confusion matrix for cross-validation predictions
confusion_matrix_cv <- table(Predicted = field_goals_df$predicted, Actual = field_goals_df$scored)

# Print the confusion matrix
print(confusion_matrix_cv)
```

```
##          Actual
## Predicted     0     1
##         0   591   101
##         1  1817 11543
```