DataAnalysisFinal

Landon Armstrong

2023-12-08

## Issue Description - NFL Penalties History

Learning more about the history of NFL penalties, affects they’ve had, and creating top 10 categories, since that’s a popular thing in sports. Want to be able to accomplish these things by accessing years of data and how we can create visual displays of information that’s been collected.

## Questions

Two specific questions I would like to attempt to answer. 1. Top 10 Leading type of penalties 2. Top 10 Leading Teams in penalties

## Data Source

<https://www.kaggle.com/datasets/mattop/nfl-penalties-data-2009-2022-season/data> Made use of the penalties.csv and players.csv datasets in my project

Youtube video: <https://www.youtube.com/watch?v=BaIaoLCi4lU&ab_channel=LandonArmstrong>

Explore the nuances of each NFL season, dissecting penalty types, frequency, and the teams and players most frequently penalized. Uncover trends, anomalies, and strategic shifts that have shaped the league’s dynamic landscape over the years.

## Documentation

There is no direct link to the data documentation. Similar information can be found on other sites, but could not find the exact data of this.

## Description of the Data

## Setup

#Use the tools in R such as str() and summary() to describe the original dataset you imported.  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.4 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(tidymodels)

## ── Attaching packages ────────────────────────────────────── tidymodels 1.1.1 ──  
## ✔ broom 1.0.5 ✔ rsample 1.2.0  
## ✔ dials 1.2.0 ✔ tune 1.1.2  
## ✔ infer 1.0.5 ✔ workflows 1.1.3  
## ✔ modeldata 1.2.0 ✔ workflowsets 1.0.1  
## ✔ parsnip 1.1.1 ✔ yardstick 1.2.0  
## ✔ recipes 1.0.8   
## ── Conflicts ───────────────────────────────────────── tidymodels\_conflicts() ──  
## ✖ scales::discard() masks purrr::discard()  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ recipes::fixed() masks stringr::fixed()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ yardstick::spec() masks readr::spec()  
## ✖ recipes::step() masks stats::step()  
## • Dig deeper into tidy modeling with R at https://www.tmwr.org

library(readr)  
library(caret)

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following objects are masked from 'package:yardstick':  
##   
## precision, recall, sensitivity, specificity  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

library(rsample)  
library(dplyr)  
  
penalties <- read.csv("penalties.csv")  
players <- read.csv("players.csv")

## Description Part 1

str(penalties)

## 'data.frame': 17136 obs. of 12 variables:  
## $ Name : chr "Chop Block" "Clipping" "Defensive 12 On-field" "Defensive Delay of Game" ...  
## $ Count : int 0 0 1 1 5 16 4 0 8 0 ...  
## $ Yards : int 0 0 5 5 25 80 57 0 40 0 ...  
## $ Declined : int 0 0 0 0 1 2 0 0 0 0 ...  
## $ Offsetting: int 0 0 0 0 0 0 1 0 0 0 ...  
## $ Home\_Count: int 0 0 0 1 1 10 1 0 4 0 ...  
## $ Away\_Count: int 0 0 1 0 4 6 3 0 4 0 ...  
## $ Off\_Count : int 0 0 0 0 0 0 0 0 7 0 ...  
## $ Def\_Count : int 0 0 0 1 5 16 4 0 0 0 ...  
## $ ST\_Count : int 0 0 1 0 0 0 0 0 1 0 ...  
## $ Week : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Year : int 2009 2009 2009 2009 2009 2009 2009 2009 2009 2009 ...

str(players)

## 'data.frame': 43407 obs. of 11 variables:  
## $ Name : chr "D.Bell" "D.Stewart" "T.Polamalu" "R.Incognito" ...  
## $ Team : chr "Buffalo" "Tennessee" "Pittsburgh" "St. Louis" ...  
## $ Pos : chr "T" "T" "SS" "C" ...  
## $ Count : int 3 3 2 3 3 1 2 2 2 2 ...  
## $ Yards : int 15 13 30 35 20 5 17 15 20 10 ...  
## $ Declined : int 1 0 0 0 0 0 0 0 0 0 ...  
## $ Offsetting: int 0 0 1 0 0 1 0 0 0 0 ...  
## $ Pre.Snap : int 3 3 0 1 2 1 0 1 1 2 ...  
## $ Penalties : chr "Illegal Formation (2), False Start (1), Offensive Holding (1)" "Illegal Formation (2), False Start (1)" "Unnecessary Roughness (1), Defensive Pass Interference (1), Face Mask (15 Yards) (1)" "Unnecessary Roughness (2), False Start (1)" ...  
## $ Week : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ Year : int 2009 2009 2009 2009 2009 2009 2009 2009 2009 2009 ...

## Description Part 2

summary(penalties)

## Name Count Yards Declined   
## Length:17136 Min. : 0.000 Min. : 0.00 Min. : 0.0000   
## Class :character 1st Qu.: 0.000 1st Qu.: 0.00 1st Qu.: 0.0000   
## Mode :character Median : 0.000 Median : 0.00 Median : 0.0000   
## Mean : 2.643 Mean : 22.46 Mean : 0.3634   
## 3rd Qu.: 2.000 3rd Qu.: 15.00 3rd Qu.: 0.0000   
## Max. :74.000 Max. :712.00 Max. :17.0000   
## Offsetting Home\_Count Away\_Count Off\_Count   
## Min. :0.00000 Min. : 0.000 Min. : 0.000 Min. : 0.000   
## 1st Qu.:0.00000 1st Qu.: 0.000 1st Qu.: 0.000 1st Qu.: 0.000   
## Median :0.00000 Median : 0.000 Median : 0.000 Median : 0.000   
## Mean :0.09109 Mean : 1.284 Mean : 1.358 Mean : 1.223   
## 3rd Qu.:0.00000 3rd Qu.: 1.000 3rd Qu.: 1.000 3rd Qu.: 0.000   
## Max. :8.00000 Max. :45.000 Max. :38.000 Max. :59.000   
## Def\_Count ST\_Count Week Year   
## Min. : 0.0000 Min. : 0.0000 Min. : 1 Min. :2009   
## 1st Qu.: 0.0000 1st Qu.: 0.0000 1st Qu.: 5 1st Qu.:2012   
## Median : 0.0000 Median : 0.0000 Median : 9 Median :2016   
## Mean : 0.9926 Mean : 0.4269 Mean : 9 Mean :2016   
## 3rd Qu.: 0.0000 3rd Qu.: 0.0000 3rd Qu.:13 3rd Qu.:2019   
## Max. :29.0000 Max. :21.0000 Max. :17 Max. :2022

summary(players)

## Name Team Pos Count   
## Length:43407 Length:43407 Length:43407 Min. :0.0000   
## Class :character Class :character Class :character 1st Qu.:1.0000   
## Mode :character Mode :character Mode :character Median :1.0000   
## Mean :0.9894   
## 3rd Qu.:1.0000   
## Max. :6.0000   
## Yards Declined Offsetting Pre.Snap   
## Min. : 0.000 Min. :0.0000 Min. :0.00000 Min. :0.0000   
## 1st Qu.: 5.000 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.0000   
## Median : 5.000 Median :0.0000 Median :0.00000 Median :0.0000   
## Mean : 8.559 Mean :0.1303 Mean :0.03405 Mean :0.3714   
## 3rd Qu.: 10.000 3rd Qu.:0.0000 3rd Qu.:0.00000 3rd Qu.:1.0000   
## Max. :106.000 Max. :3.0000 Max. :2.00000 Max. :4.0000   
## Penalties Week Year   
## Length:43407 Min. : 1.000 Min. :2009   
## Class :character 1st Qu.: 4.000 1st Qu.:2012   
## Mode :character Median : 9.000 Median :2015   
## Mean : 8.847 Mean :2015   
## 3rd Qu.:13.000 3rd Qu.:2019   
## Max. :17.000 Max. :2022

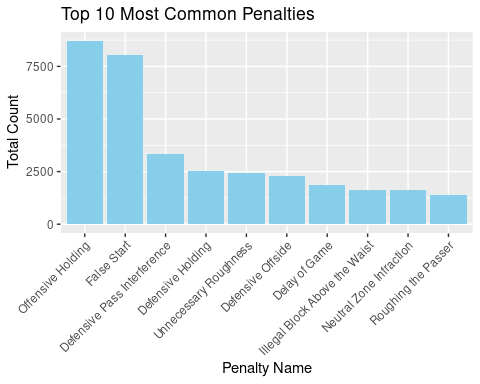
## Top 10 Most Common Penalties

#Count the number of penalties for each unique penalty  
penalty\_counts <- penalties %>%  
 group\_by(Name) %>%  
 summarize(Total\_Count = sum(Count))  
  
# Arrange in descending order to find the most common penalties  
penalty\_counts <- penalty\_counts %>%  
 arrange(desc(Total\_Count))  
  
#Table of our top 10  
head(penalty\_counts, 10)

## # A tibble: 10 × 2  
## Name Total\_Count  
## <chr> <int>  
## 1 Offensive Holding 8716  
## 2 False Start 8032  
## 3 Defensive Pass Interference 3357  
## 4 Defensive Holding 2550  
## 5 Unnecessary Roughness 2441  
## 6 Defensive Offside 2267  
## 7 Delay of Game 1866  
## 8 Illegal Block Above the Waist 1637  
## 9 Neutral Zone Infraction 1623  
## 10 Roughing the Passer 1401

## Plot

ggplot(penalty\_counts[1:10, ], aes(x = reorder(Name, -Total\_Count), y = Total\_Count)) +  
 geom\_bar(stat = "identity", fill = "skyblue") +  
 labs(title = "Top 10 Most Common Penalties",  
 x = "Penalty Name",  
 y = "Total Count") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



## Compare Top 10 Most Common Penalties (2009 vs 2022)

# Filter data for 2009 and 2022  
penalties\_data\_2009\_2022 <- penalties %>%  
 filter(Year %in% c(2009, 2022))  
  
# Group by year and penalty name, and sum the counts  
penalty\_counts\_by\_year <- penalties\_data\_2009\_2022 %>%  
 group\_by(Year, Name) %>%  
 summarize(Total\_Count = sum(Count))

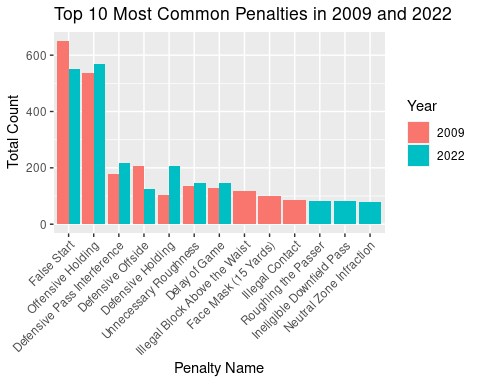
## `summarise()` has grouped output by 'Year'. You can override using the  
## `.groups` argument.

## Further arrange the data

# Arrange in descending order within each year  
penalty\_counts\_by\_year <- penalty\_counts\_by\_year %>%  
 arrange(Year, desc(Total\_Count))  
  
# Display the most common penalties for 2009 and 2022  
top\_penalties\_2009\_2022 <- penalty\_counts\_by\_year %>%  
 group\_by(Year) %>%  
 slice\_head(n = 10)

## Plot

# Create a side-by-side bar plot for visualization  
ggplot(top\_penalties\_2009\_2022, aes(x = reorder(Name, -Total\_Count), y = Total\_Count, fill = as.factor(Year))) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(title = "Top 10 Most Common Penalties in 2009 and 2022",  
 x = "Penalty Name",  
 y = "Total Count",  
 fill = "Year") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))

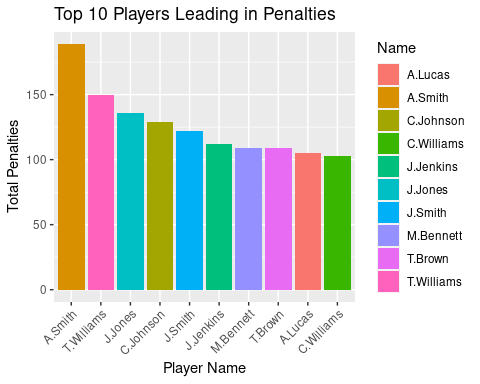


## Top 10 Players Leading in Penalties

# Preprocess the data  
penalties\_data\_processed <- players %>%  
 mutate(Penalty\_List = strsplit(as.character(Penalties), ", ")) %>%  
 unnest(Penalty\_List)  
  
# Group by player and calculate total penalties  
top\_players <- penalties\_data\_processed %>%  
 group\_by(Name) %>%  
 summarize(Total\_Penalties = n()) %>%  
 arrange(desc(Total\_Penalties)) %>%  
 head(10)

## Plot

# Create a bar plot for the top 10 players  
ggplot(top\_players, aes(x = reorder(Name, -Total\_Penalties), y = Total\_Penalties, fill = Name)) +  
 geom\_bar(stat = "identity") +  
 labs(title = "Top 10 Players Leading in Penalties",  
 x = "Player Name",  
 y = "Total Penalties") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))

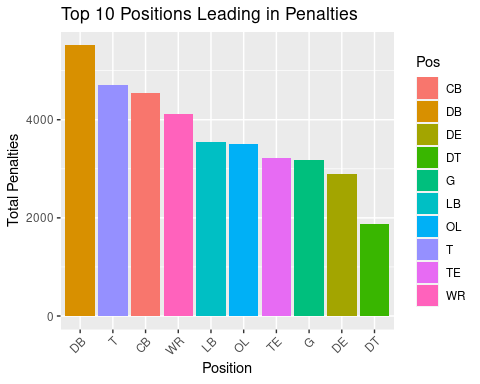


## Top 10 Positions Leading in Penalties

# Group by position and calculate total penalties  
top\_positions <- penalties\_data\_processed %>%  
 group\_by(Pos) %>%  
 summarize(Total\_Penalties = n()) %>%  
 arrange(desc(Total\_Penalties)) %>%  
 head(10)

## Plot

# Create a bar plot for the top 10 positions  
ggplot(top\_positions, aes(x = reorder(Pos, -Total\_Penalties), y = Total\_Penalties, fill = Pos)) +  
 geom\_bar(stat = "identity") +  
 labs(title = "Top 10 Positions Leading in Penalties",  
 x = "Position",  
 y = "Total Penalties") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



## Top 10 Teams Leading in Penalties

# Group by team and calculate total penalties  
top\_teams <- penalties\_data\_processed %>%  
 group\_by(Team) %>%  
 summarize(Total\_Penalties = n()) %>%  
 arrange(desc(Total\_Penalties)) %>%  
 head(10)

## Plot

# Create a bar plot for the top 10 teams  
ggplot(top\_teams, aes(x = reorder(Team, -Total\_Penalties), y = Total\_Penalties, fill = Team)) +  
 geom\_bar(stat = "identity") +  
 labs(title = "Top 10 Teams Leading in Penalties",  
 x = "Team",  
 y = "Total Penalties") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_y\_continuous(breaks = seq(0, max(top\_teams$Total\_Penalties), by = 100))

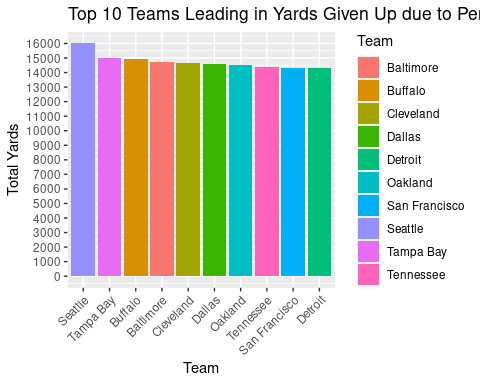


## Top 10 Teams Leading in Yards Given Up due to Penalties

# Preprocess the data to separate individual penalties  
penalties\_data\_processed <- players %>%  
 mutate(Penalty\_List = strsplit(as.character(Penalties), ", ")) %>%  
 unnest(Penalty\_List)  
  
# Group by team and calculate total yards given up  
top\_teams\_yards <- penalties\_data\_processed %>%  
 group\_by(Team) %>%  
 summarize(Total\_Yards = sum(Yards)) %>%  
 arrange(desc(Total\_Yards)) %>%  
 head(10)

## Plot

# Create a bar plot for the top 10 teams in yards given up  
ggplot(top\_teams\_yards, aes(x = reorder(Team, -Total\_Yards), y = Total\_Yards, fill = Team)) +  
 geom\_bar(stat = "identity") +  
 labs(title = "Top 10 Teams Leading in Yards Given Up due to Penalties",  
 x = "Team",  
 y = "Total Yards") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_y\_continuous(breaks = seq(0, max(top\_teams\_yards$Total\_Yards), by = 1000))



## Penalty Count Prediction

## Split the data

Put 70% of the data in training. Set the strata to Count

# Set the seed for reproducibility  
set.seed(123)  
  
# Create the split indices based on the Count variable  
penalties\_split <- initial\_split(players, prop = 0.7, strata = Count)  
  
# Extract the training and testing sets  
penalties\_training <- penalties\_split %>%  
 training()  
penalties\_test <- penalties\_split %>%  
 testing()

## Verify the Split

Examine the distributions of penalties in the training and test dataframes

## Solution

nrow(penalties\_training)

## [1] 30384

nrow(penalties\_test)

## [1] 13023

summary(penalties\_training)

## Name Team Pos Count   
## Length:30384 Length:30384 Length:30384 Min. :0.0000   
## Class :character Class :character Class :character 1st Qu.:1.0000   
## Mode :character Mode :character Mode :character Median :1.0000   
## Mean :0.9887   
## 3rd Qu.:1.0000   
## Max. :6.0000   
## Yards Declined Offsetting Pre.Snap   
## Min. : 0.000 Min. :0.0000 Min. :0.00000 Min. :0.0000   
## 1st Qu.: 5.000 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.0000   
## Median : 5.000 Median :0.0000 Median :0.00000 Median :0.0000   
## Mean : 8.569 Mean :0.1302 Mean :0.03485 Mean :0.3685   
## 3rd Qu.:10.000 3rd Qu.:0.0000 3rd Qu.:0.00000 3rd Qu.:1.0000   
## Max. :91.000 Max. :3.0000 Max. :2.00000 Max. :4.0000   
## Penalties Week Year   
## Length:30384 Min. : 1.000 Min. :2009   
## Class :character 1st Qu.: 4.000 1st Qu.:2012   
## Mode :character Median : 9.000 Median :2015   
## Mean : 8.835 Mean :2015   
## 3rd Qu.:13.000 3rd Qu.:2019   
## Max. :17.000 Max. :2022

summary(penalties\_test)

## Name Team Pos Count   
## Length:13023 Length:13023 Length:13023 Min. :0.0000   
## Class :character Class :character Class :character 1st Qu.:1.0000   
## Mode :character Mode :character Mode :character Median :1.0000   
## Mean :0.9911   
## 3rd Qu.:1.0000   
## Max. :5.0000   
## Yards Declined Offsetting Pre.Snap   
## Min. : 0.000 Min. :0.0000 Min. :0.00000 Min. :0.000   
## 1st Qu.: 5.000 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.000   
## Median : 6.000 Median :0.0000 Median :0.00000 Median :0.000   
## Mean : 8.536 Mean :0.1305 Mean :0.03217 Mean :0.378   
## 3rd Qu.: 10.000 3rd Qu.:0.0000 3rd Qu.:0.00000 3rd Qu.:1.000   
## Max. :106.000 Max. :2.0000 Max. :2.00000 Max. :4.000   
## Penalties Week Year   
## Length:13023 Min. : 1.000 Min. :2009   
## Class :character 1st Qu.: 4.000 1st Qu.:2012   
## Mode :character Median : 9.000 Median :2016   
## Mean : 8.875 Mean :2015   
## 3rd Qu.:13.000 3rd Qu.:2019   
## Max. :17.000 Max. :2022

## Create a linear regression model

Call it linear\_model

## Solution

# Initialize a linear regression object, linear\_model  
linear\_model <- linear\_reg() %>%   
 # Set the model engine  
 set\_engine("lm") %>%   
 # Set the model mode  
 set\_mode("regression")

## Train

Fit a model, lm\_fit, to predict penalties using yards as the only predictor. Use the training data only.

## Solution

lm\_fit <- linear\_model %>%   
 fit(Count ~ Yards,  
 data = penalties\_training)

## Examine the Model

Print lm\_fit to view model information and use the tidy() function.

## Solution

lm\_fit

## parsnip model object  
##   
##   
## Call:  
## stats::lm(formula = Count ~ Yards, data = data)  
##   
## Coefficients:  
## (Intercept) Yards   
## 0.60334 0.04498

tidy(lm\_fit)

## # A tibble: 2 × 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 0.603 0.00359 168. 0  
## 2 Yards 0.0450 0.000324 139. 0

## Predictions

Create penalties\_predictions using the predict() function on lm\_fit

## Solution

penalties\_predictions <- predict(lm\_fit,  
 new\_data = penalties\_test)  
head(penalties\_predictions)

## # A tibble: 6 × 1  
## .pred  
## <dbl>  
## 1 1.28  
## 2 1.19  
## 3 1.50  
## 4 1.37  
## 5 1.28  
## 6 1.50

str(penalties\_predictions)

## tibble [13,023 × 1] (S3: tbl\_df/tbl/data.frame)  
## $ .pred: num [1:13023] 1.28 1.19 1.5 1.37 1.28 ...

## Compare

Use summary() to compare the distributions of the actual and predicted values

summary(penalties\_predictions)

## .pred   
## Min. :0.6033   
## 1st Qu.:0.8282   
## Median :0.8732   
## Mean :0.9873   
## 3rd Qu.:1.0531   
## Max. :5.3709

summary(penalties\_test$Count)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 1.0000 1.0000 0.9911 1.0000 5.0000

## Build results

Create a dataframe: penalties\_results

## Solution

penalties\_results <- penalties\_test %>%   
 select(Count,Yards) %>%   
 bind\_cols(penalties\_predictions)  
  
str(penalties\_results)

## 'data.frame': 13023 obs. of 3 variables:  
## $ Count: int 3 3 3 2 2 2 2 2 0 1 ...  
## $ Yards: int 15 13 20 17 15 20 15 16 0 5 ...  
## $ .pred: num 1.28 1.19 1.5 1.37 1.28 ...

## Evaluate the Model

Use the rmse() and rsq() functions

## Solution

penalties\_results %>%   
 rmse(Count, .pred)

## # A tibble: 1 × 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 rmse standard 0.397

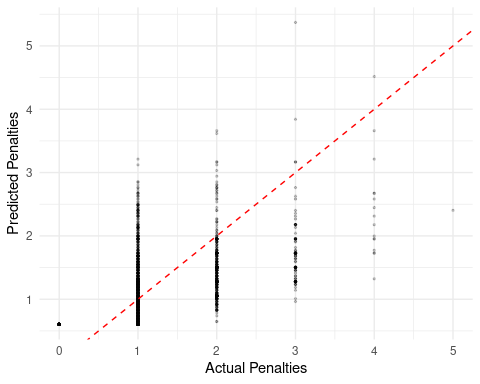
penalties\_results %>%   
 rsq(Count, .pred)

## # A tibble: 1 × 3  
## .metric .estimator .estimate  
## <chr> <chr> <dbl>  
## 1 rsq standard 0.390

## Create an R squared plot of model performance

## Solution

ggplot(penalties\_test, aes(x = Count, y = penalties\_predictions$.pred)) +  
 geom\_point(alpha = 0.2, size = 0.5) +  
 geom\_abline(intercept = 0, slope = 1, color = "red", linetype = "dashed") +  
 labs(x = 'Actual Penalties', y = 'Predicted Penalties') +  
 theme\_minimal() +  
 theme(plot.title = element\_text(hjust = 0.5))



## THE END