Punt

Sam Hughes

1/8/2022

This file details my adventure in converting the punt into a resulting expected point metric. The expected points were calculated by me in my first down document, so all I needed was to figure out the yardline of where the punt play ended. Unfortunately, this was not as easy as I anticipated because simple play by play data actually does not give this information. Instead, I formulated this using a function that checked the next play yardline, whether the punting play ended in a score, and whether it resulted in a turnover. Once I did this, I could make a model that predicts the resulting expected points of a punt.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.0.5
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.0.5
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.0.3
```

```
## Loading required package: lattice
```

```
# load in data from csv
df_punting_data = read.csv("punt_data.csv")[ , 2:3]

### Make model that predicts the resulting expected points after a punt

# Split data
```

```r
set.seed(123)
training_samples_punt = createDataPartition(df_punting_data$resulting_ep, p = 0.8, list = FALSE)
punt_train = df_punting_data[training_samples_punt, ]
punt_test = df_punting_data[-training_samples_punt, ]

# aggregate test data for graph later
punt_test_agg = punt_test %>%
  group_by(start) %>%
  summarise("mean" = mean(resulting_ep), "count" = length(resulting_ep))

# I know from the plot in practicing_punting_data that the equation is likely a quadratic or cubic
punt_train$start_squared = punt_train$start^2
punt_train$start_cubed = punt_train$start^3

summary(lm(resulting_ep ~ ., data = punt_train))
```

```
##
## Call:
## lm(formula = resulting_ep ~ ., data = punt_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.5637 -0.2959  0.0225  0.3146  8.1610
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.631e+00  4.433e-01  -5.935 2.99e-09 ***
## start          1.230e-01  2.136e-02   5.758 8.64e-09 ***
## start_squared -1.934e-03  3.320e-04  -5.826 5.75e-09 ***
## start_cubed    6.787e-06  1.672e-06   4.060 4.92e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9474 on 21400 degrees of freedom
## Multiple R-squared:  0.2807, Adjusted R-squared:  0.2806
## F-statistic:  2784 on 3 and 21400 DF,  p-value: < 2.2e-16
```

```r
anova(lm(resulting_ep ~ ., data = punt_train))
```

```
## Analysis of Variance Table
##
## Response: resulting_ep
##                  Df  Sum Sq Mean Sq  F value    Pr(>F)
## start             1  7077.1  7077.1 7883.936 < 2.2e-16 ***
## start_squared     1   404.5   404.5  450.652 < 2.2e-16 ***
## start_cubed       1    14.8    14.8   16.486 4.918e-05 ***
## Residuals     21400 19209.9     0.9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Everything is significant
```

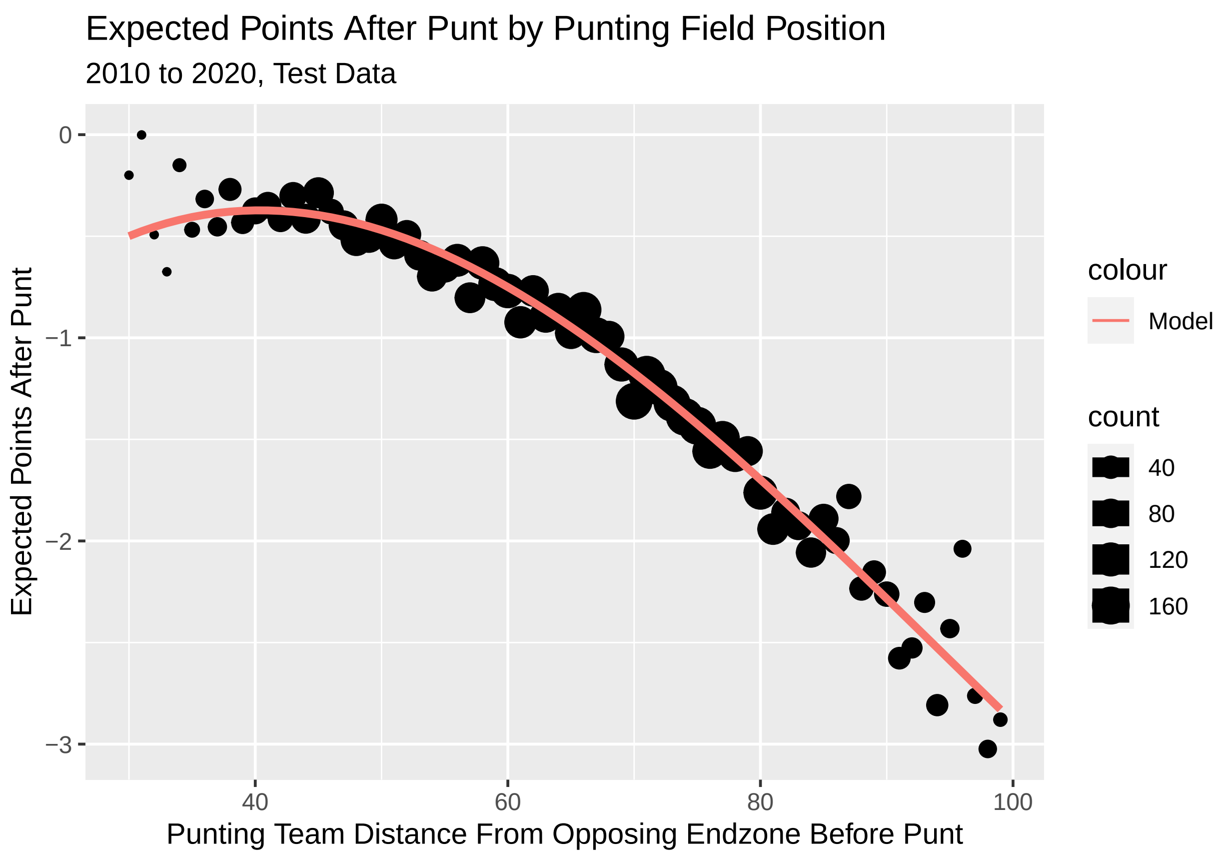
```r
punt_model = lm(resulting_ep ~ ., data = punt_train)

punt_equation = function(yardline) {
  punt_model$coefficients[1] + punt_model$coefficients[2] * yardline + punt_model$coefficients[3] * yard
}

punt_equation(90)
```

```
## (Intercept)
##   -2.283713
```

```r
ggplot(data = punt_test_agg,
       mapping = aes(x = start,
                     y = mean))+
  ggtitle("Expected Points After Punt by Punting Field Position", subtitle = "2010 to 2020, Test Data")+
  xlab("Punting Team Distance From Opposing Endzone Before Punt")+
  ylab("Expected Points After Punt")+
  geom_point(aes(size = count))+
  geom_line(aes(y = punt_equation(start),
                size = 3, colour = "Model"))
```



The fit is fantastic on this model, especially considering that this is the test data, so I do not feel as if it is necessary to make any other models.