R Notebook

Alex Baker, James Kitch, Jackson Smith, Matthew Sheridan

Introduction:

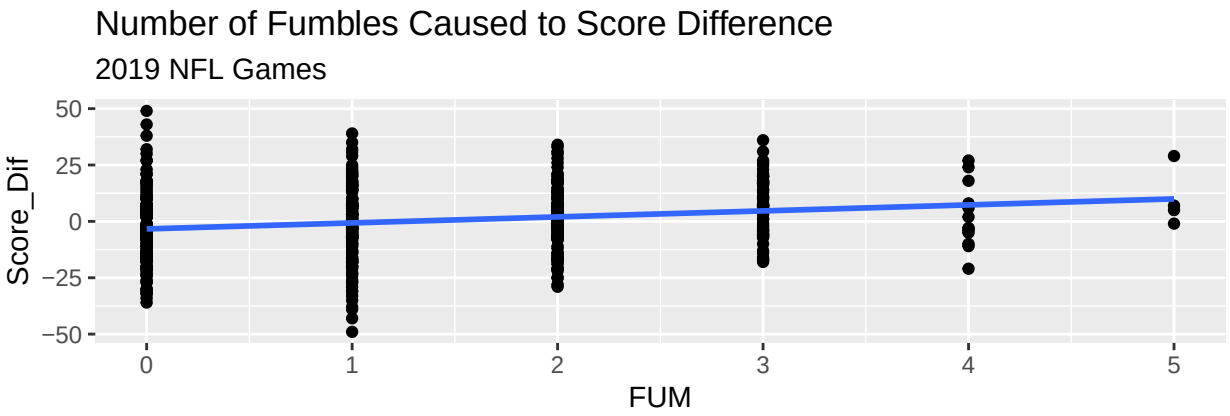
Our data comes from the nflfastR package, which contains accumulated play-by-play data from 1999 through 2021, with additional predictors beginning in 2006. We hope to create an efficient, streamlined model capable of predicting the score differential of an NFL game. Furthermore, we also want to be able to use what we gain from our models to help make insightful analyses and predictions in other scenarios. Some predictors will impact only one of these models while others will prove to be far more significant than widely recognized: it all depends on which refining method produces the most successful and significant model.

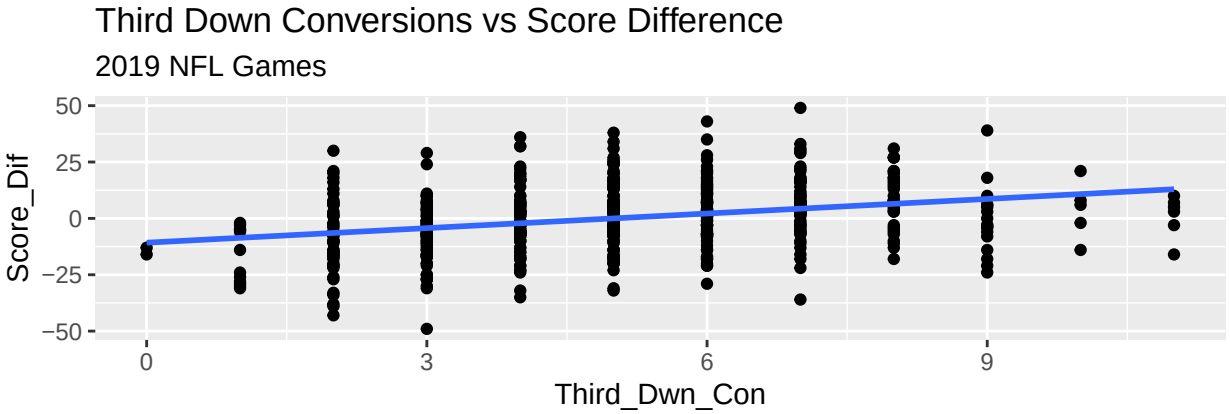
While we will have some variables with relatively clear-cut relationships with score differential, our analysis will also focus on how more "strategic" variables relate to success. More turnovers in a game and more total yards should correlate strongly with score differential, but we are interested in variables that have a more unclear relationship with the final outcome. Are missed extra point attempts indicative of the team as a whole having a bad day? What about third down conversion rate? And quarterback hits? These are the kinds of interactions and trends we hope to expose in our model. By using metrics that are not generally used to predict game outcomes, we hope to find insight into predicting wins that are not conventionally expected.

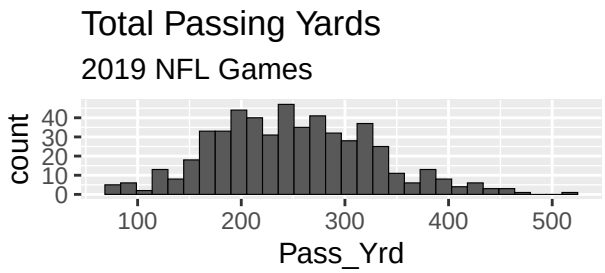
Data Cleaning, EDA:

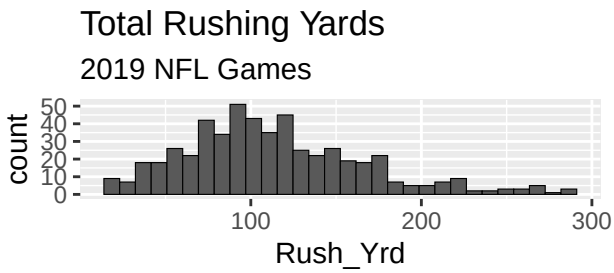
Below, we graphed the distributions for a number of different potential predictors, as well as their relationship with the response variable of interest. Most predictors are slightly right-skewed, as they have a positive support and have a slight "bell" shape for the most common values but are stretched out by the few exceptional games where a team throws for 500 yards or rushes for 250 yards. However, it is mild right-skewness as it is not to the extent that log-transforming them would make the distributions more symmetric. For example, if we log-transform `Pass_Yrd` we actually find that the the resulting distribution is left-skewed, suggesting that the log transformation was actually too strong!

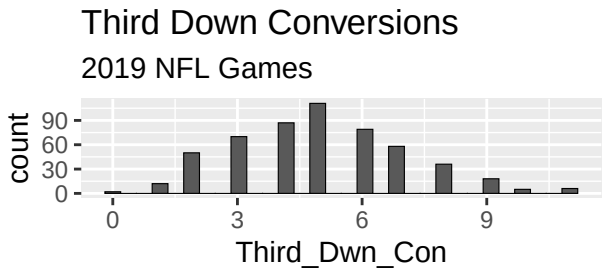
Furthermore, we can do a preliminary investigation of the relationships between some of the predictors and the outcome variable of score differential in 2019. Notably, we can see that the number of fumbles caused by a team is positively associated with score differential and that it is a relatively linear trend. The number of third down conversions is also positively associated with score differential, but this seems more complicated than a "simple" linear trend. Teams with a very large number (> 8) of third down conversions in a game appear to have a more negative score differential than teans with 5-7 third down conversions. This suggests that in future models investigating the effects of quadratic or higher order polynomial terms could be reasonable.

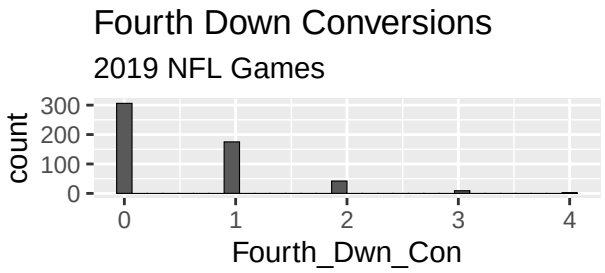

Number of Fumbles Caused to Score Difference
2019 NFL Games
Score_Dif
50
25
0
−25
−50
0
1
2
3
4
5
FUM

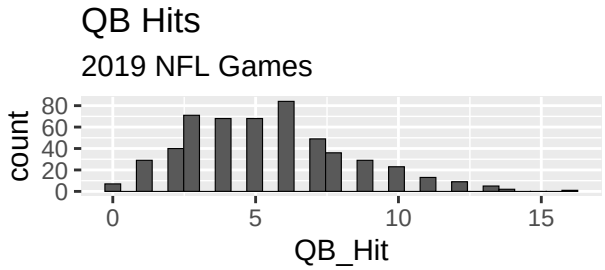

Third Down Conversions vs Score Difference
2019 NFL Games
Score_Dif
50
25
0
−25
−50
0
3
6
9
Third_Dwn_Con

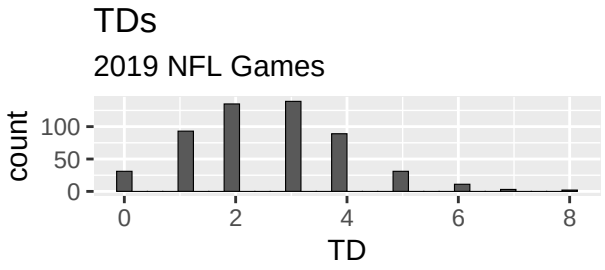












Initial Modeling:

The code for initializing the models has been hidden for space constraints. The linear model was using all predictors, the stepwise model stepped from an intercept only model to all predictors, and the randomforest has not been tuned (YET) and uses all predictors, as well as the default parameters. We can see that the RMSE's are pretty close together for all the models, but the stepwise linear model seems to outperform all.

LMER Modeling

```
data_2017_19_scale <- data_2017_19
data_2017_19_scale[,6:7] <- data_2017_19_scale[,6:7] %>% sapply(scale, scale=F)


lm1 <- lm(Score_Dif ~. -Team_Name -Game_id -year, data=data_2017_19_scale)

lmer_all <- lmer(Score_Dif ~ Home + TFL + QB_Hit + Pass_Yrd + Rush_Yrd + TD + FG_attempt + XP_missed + I
```

```
## Warning in model.matrix.default(fixedform, fr, contrasts): the response appeared
## on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(fixedform, fr, contrasts): problem with term 15
## in model.matrix: no columns are assigned
```

```
lmer_all_slope <- lmer(Score_Dif ~ Home + TFL + QB_Hit + Pass_Yrd + Rush_Yrd + TD + FG_attempt + XP_miss
```

```
## Warning in model.matrix.default(fixedform, fr, contrasts): the response appeared
## on the right-hand side and was dropped
```

```
## Warning in model.matrix.default(fixedform, fr, contrasts): problem with term 15
## in model.matrix: no columns are assigned
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.562117 (tol = 0.002, component 1)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unider
##  - Rescale variables?
```

```
lmer_all_step <- lmer(Score_Dif ~ TD + FG_attempt + INT + Rush_Yrd + QB_Hit + FUM +
    Third_Dwn_Con + FG_missed + Fourth_Dwn_Con + Home + (1+Rush_Yrd|Team_Name),
    data=data_2017_19_scale)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model failed to conver
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unider
##  - Rescale variables?
```

```
AIC(lmer_all, lmer_all_slope, lmer_all_step, step1, lm1)
```

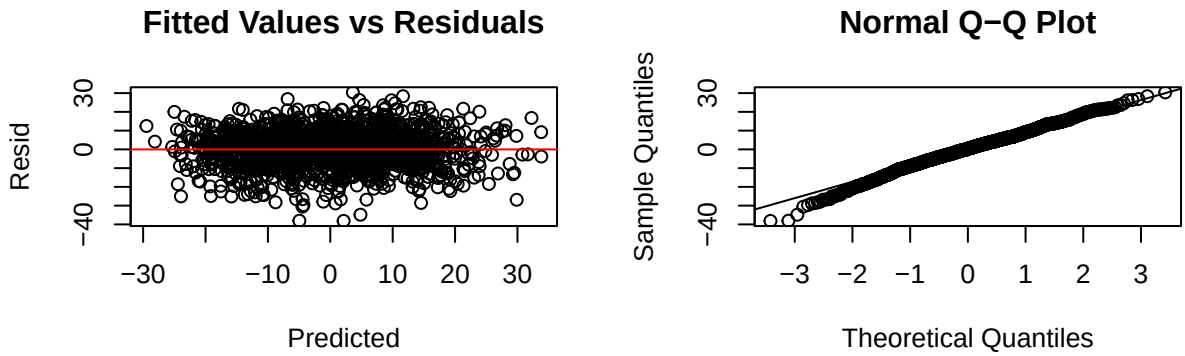
```
##                df      AIC
## lmer_all       17 11906.45
## lmer_all_slope 19 11908.01
## lmer_all_step  15 11891.03
## step1          43 11855.26
## lm1            16 11911.35
```

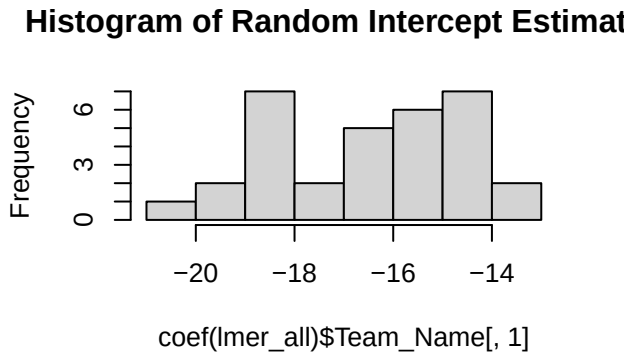
Next, we fit a Linear Mixed-Effects model with random intercepts by team to see if that had any impact on our predictions. Given that observations can be grouped by team and there are 32 teams, a mixed-effects model was a natural next step in our modeling approach. We fit two different types of mixed-effects models: one with all available predictors, and one with the predictors chosen by sequential variable selection. We found that the inclusion of a random intercept did not improve prediction substantially, with the random intercept term only accounting for $\approx 5\%$ of the total variance in the model. Most coefficient estimates were very similar to the standard OLS run previously. Some of this may be due to the fact that there are a relatively large number of observations per team. Although there are 32 teams, each team has 48 - 60 "measurements" or games in the dataset and thus the benefit of linear mixed-effects modeling in shrinking outlier intercepts towards the mean is less pronounced.

Nonetheless, a random intercept model with only the predictors returned by the stepwise variable selection process as outlined previously did increase the impact of the random intercept and decreased the AIC. We also fit a linear mixed-effects model with a random intercept by team and random slope for the relationship of total rushing yards with final score differential. In theory, if some teams were primarily a "passing" team, total rushing yards might have less of an outcome on final score differential than if their primary game strategy relied on the run. However, we find that this random slope model actually worsens the model AIC suggesting that this is not a particularly important trend to model.

The fitted values vs residuals plot and Normal Q-Q plots are both reasonable for this random-intercept mixed-effects model, demonstrating that constant variance and normality of residuals are acceptable assumptions for this model. However, there is some concern with the distribution of fitted random intercept estimates–the histogram is neither normal-shaped or symmetric. This could potentially be fixed with **Blank but it is probably not an issue at the end of the day**.

```
par(mfrow=c(2, 2))
plot(resid(lmer_all)~predict(lmer_all), main="Fitted Values vs Residuals", xlab="Predicted", ylab="Resi
abline(h=0, col = "red")
#normality of residuals
qqnorm(resid(lmer_all))
qqline(resid(lmer_all))
#check normality of random effects - non-normal
hist(coef(lmer_all)$Team_Name[,1], main="Histogram of Random Intercept Estimates")
```





Conclusion:

National Football League data provides a ripe opportunity for statistical analysis, especially when able to access play-by-play level data using the `nflfastR` package. While most variables have a slight right skew to them and only take on positive values, this did not present tremendous challenges for our models. Going forward, we're interested to continue investigating the impact of variables that provide a clear relationship towards "success" in a game–such as total passing yards or total rushing yards–as well as more "strategic" variables such as the number of third and fourth down conversions. Given the violations of symmetric

distributions in many of the predictors and the potential for non-linear relationships to arise, examining the performance and interpretations of Random Forest models will also be a focus in our analysis.