The Impact of Play Clock Timing on NFL Play Outcomes

Dominic Hoar-Weiler, Nathan Yao, and Justin Fung

January 6, 2025

Contents

1 Introduction

One moment etched in the hearts of Minnesota Vikings fans is the 2018 NFL playoff game against the New Orleans Saints. With just 5 seconds left on the play clock, Case Keenum launches a high, spiraling pass to Stefon Diggs, who streaks down the right sideline. The crowd erupts as Diggs makes the catch and sprints into the end zone, completing one of the most iconic touchdowns in NFL history. This unforgettable play, now famously known as the Minneapolis Miracle, highlights the critical role of the play clock. In football, the play clock isn't just a countdown, it's a pressure cooker that influences decision making, timing, and ultimately, the outcome of a play.

The focus of this research is to examine how the play clock at the snap, along with other variables, impacts play success. By investigating this relationship, we aim to develop a predictive model that forecasts play outcomes based on the timing of the snap.

2 Exploratory Analysis:

2.1 Background and Variables

We analyze multiple datasets, including player statistics, play-by-play data, and game metadata, sourced from the 2024–2025 NFL season as part of the Big Data Bowl competition.

The data that was used for most of analysis came from the "plays.csv" data set, A number of exploratory variables were recorded:
- yardsToGo:Distance needed for a first down (numeric)
- gameClock:Time on clock of play (MM:SS)
- offenseFormation:Formation used by possession team (text)
- playClockAtSnap:What the play clock value was at time of snap (numeric)
- passResult:Dropback outcome of the play (C: Complete pass, I: Incomplete pass , S: Quarterback sack, IN: Intercepted pass, R: Scramble, text)

- yardsGained:Net yards gained by the offense, including penalty yardage (numeric)

and many more.

2.2 Some EDA with Play Clock at Snap

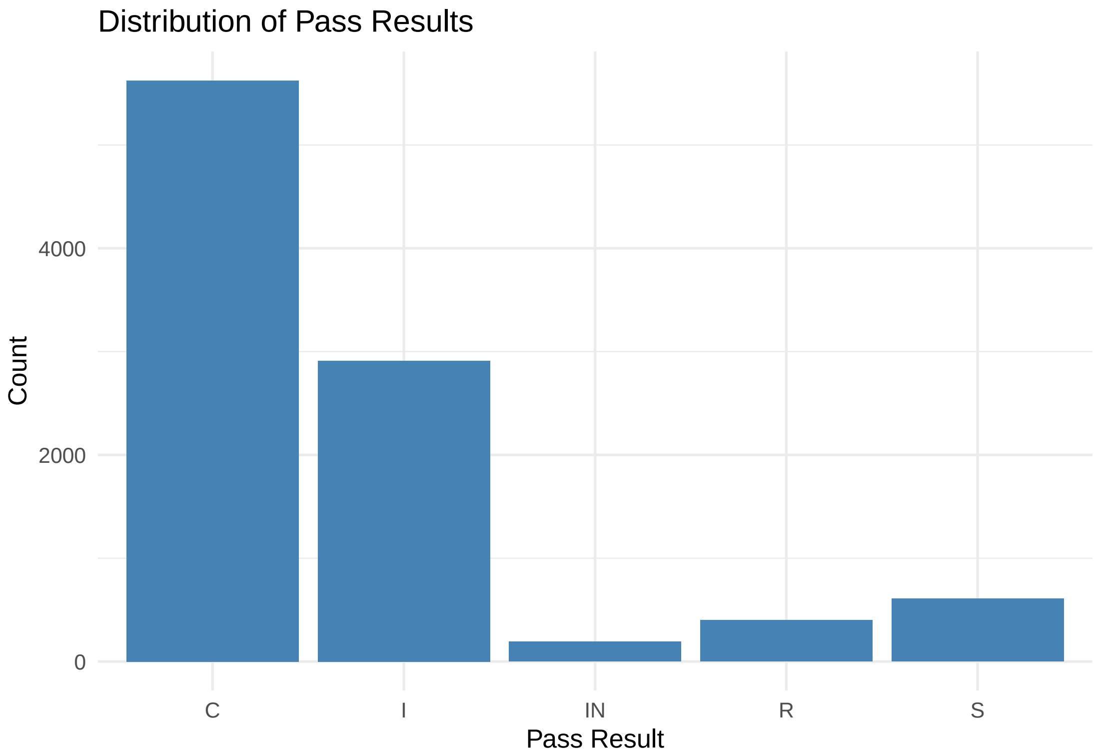
2.2.1 Distributions of Important Variables

There were a variety of possible variables that along with playClockAtSnap that would be important in our analysis. This section explores distributions of important variables and the relationships between them. Below are the different variables and their distributions that we looked at in our analysis:

passResult (Categorical): We have filtered out the NA values which indicate non-pass plays, as we are only trying to explore passing plays.

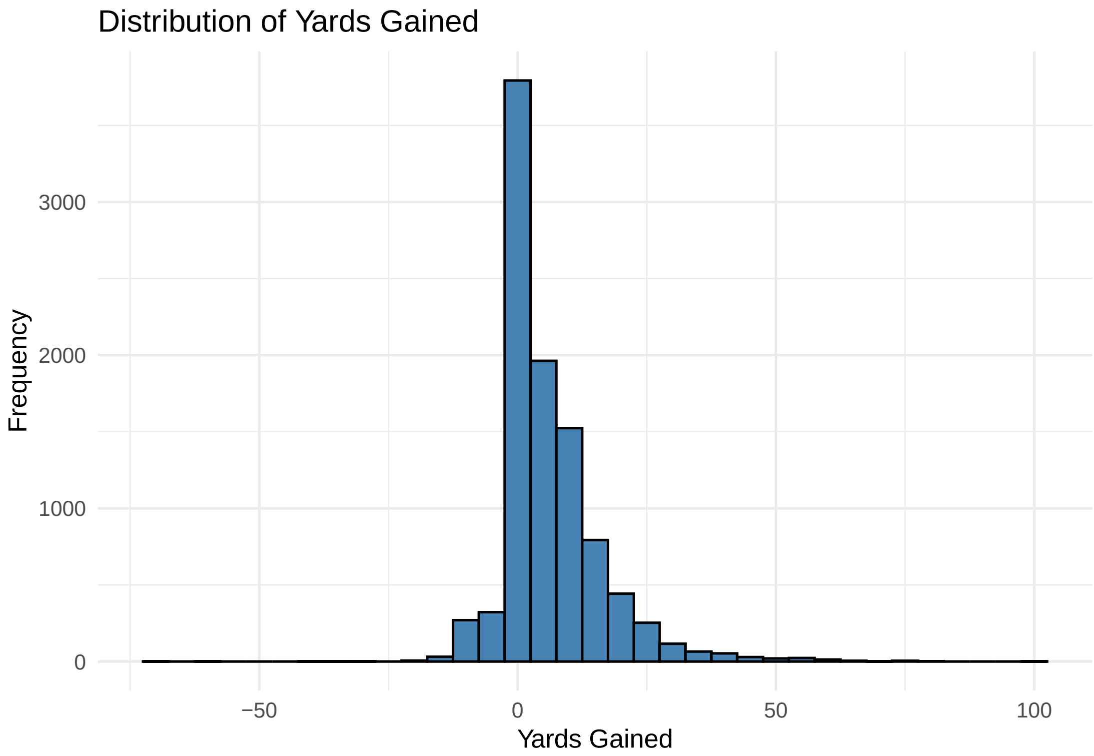
```
plays_filtered <- plays_raw_data %>%
  filter(!is.na(passResult))

ggplot(plays_filtered, aes(x = passResult)) +
  geom_bar(fill = "steelblue") +
  labs(
    title = "Distribution of Pass Results",
    x = "Pass Result",
    y = "Count"
  ) +
  theme_minimal()
```



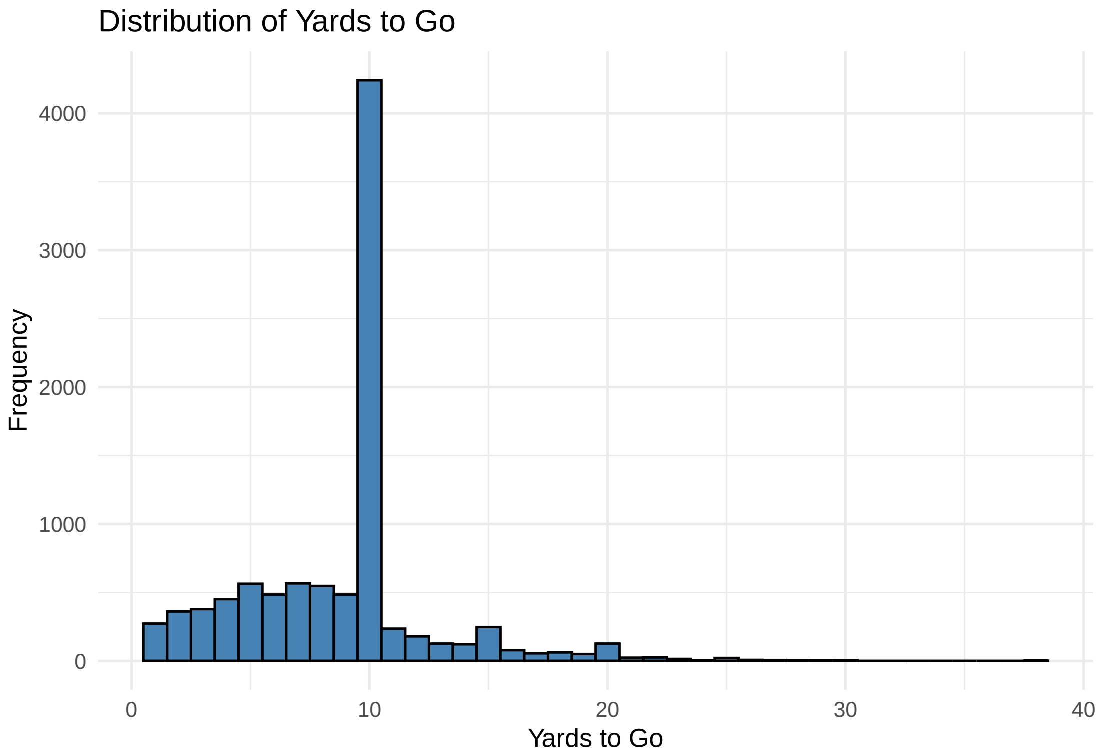
yardsGained (Quantitative):

```
ggplot(plays_filtered, aes(x = yardsGained)) +
  geom_histogram(binwidth = 5, fill = "steelblue", color = "black") +
  labs(
    title = "Distribution of Yards Gained",
    x = "Yards Gained",
    y = "Frequency"
  ) +
  theme_minimal()
```



yardsToGo (Quantitative):

```
ggplot(plays_filtered, aes(x = yardsToGo)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +
  labs(
    title = "Distribution of Yards to Go",
    x = "Yards to Go",
    y = "Frequency"
  ) +
  theme_minimal()
```



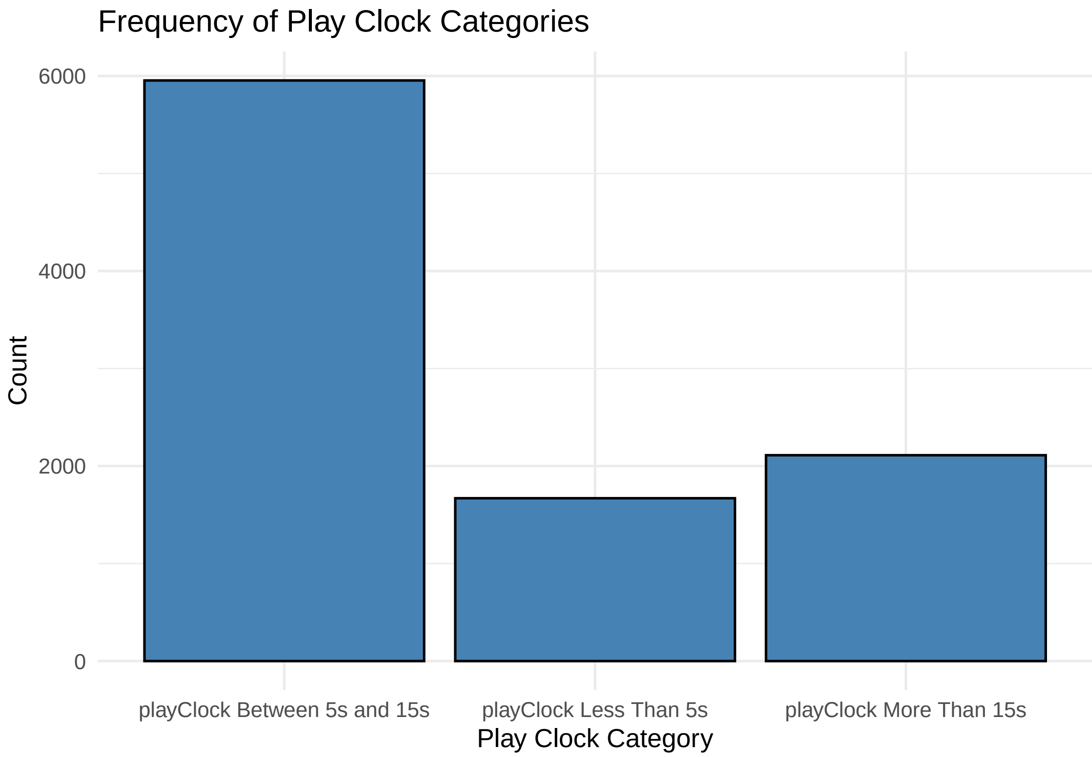
playClockAtSnap (Quantitative): For the play clock at snap, the quantitative variables was split up into 3 buckets: High (>=15 seconds), Medium (5–15 seconds), and Low (<5 seconds). These thresholds were chosen to reflect typical game scenarios: early snaps, moderate decision-making time, and rushed snaps. In this case, we created a new variable called playClockCategory which is a categorical variable that has counts of plays in each of the 3 categories. Below is the distribution:

```
plays_raw_data <- plays_raw_data %>%
  mutate(playClockCategory =
           case_when(playClockAtSnap >= 15 ~ "playClock More Than 15s",
                     playClockAtSnap >= 5 &
                     playClockAtSnap < 15 ~ "playClock Between 5s and 15s",
                     playClockAtSnap < 5 ~ "playClock Less Than 5s",
```

```
    )
  )

plays_filtered <- plays_raw_data %>%
  filter(!is.na(playClockCategory), !is.na(passResult))

ggplot(plays_filtered, aes(x = playClockCategory)) +
  geom_bar(fill = "steelblue", color = "black") +
  labs(
    title = "Frequency of Play Clock Categories",
    x = "Play Clock Category",
    y = "Count"
  ) +
  theme_minimal()
```

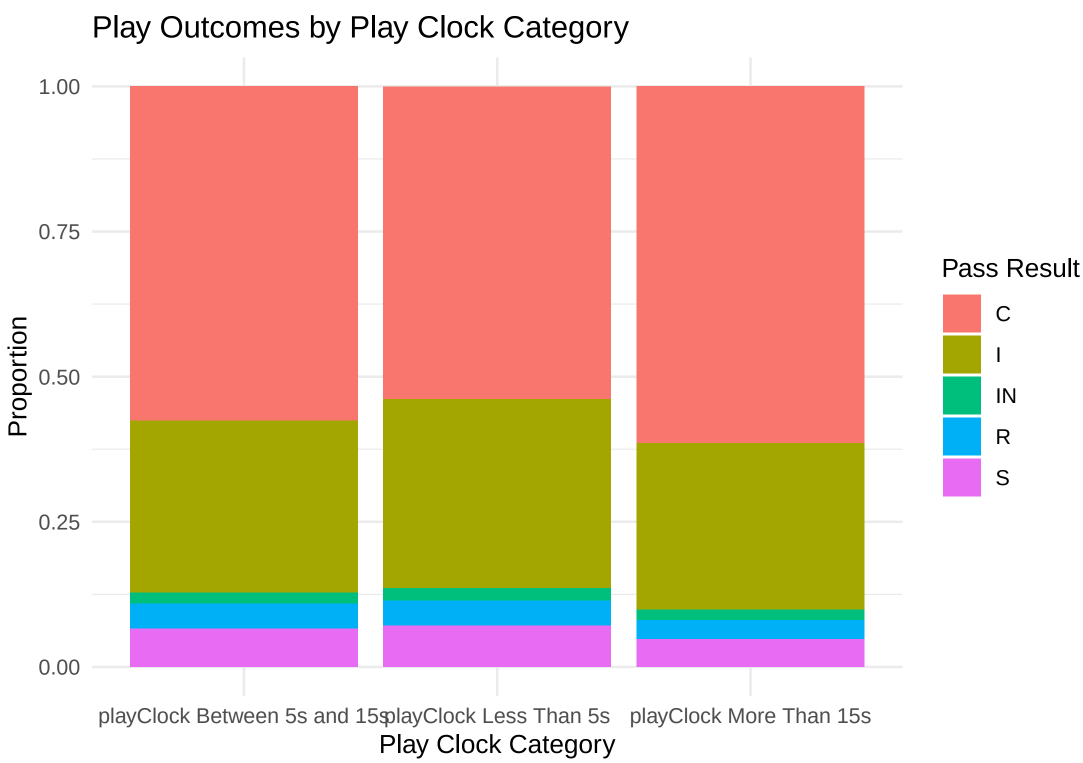


3 Will add a summary of these tomorrow and still need to edit everything beneath this line.

3.0.1 Relationships Between Variables

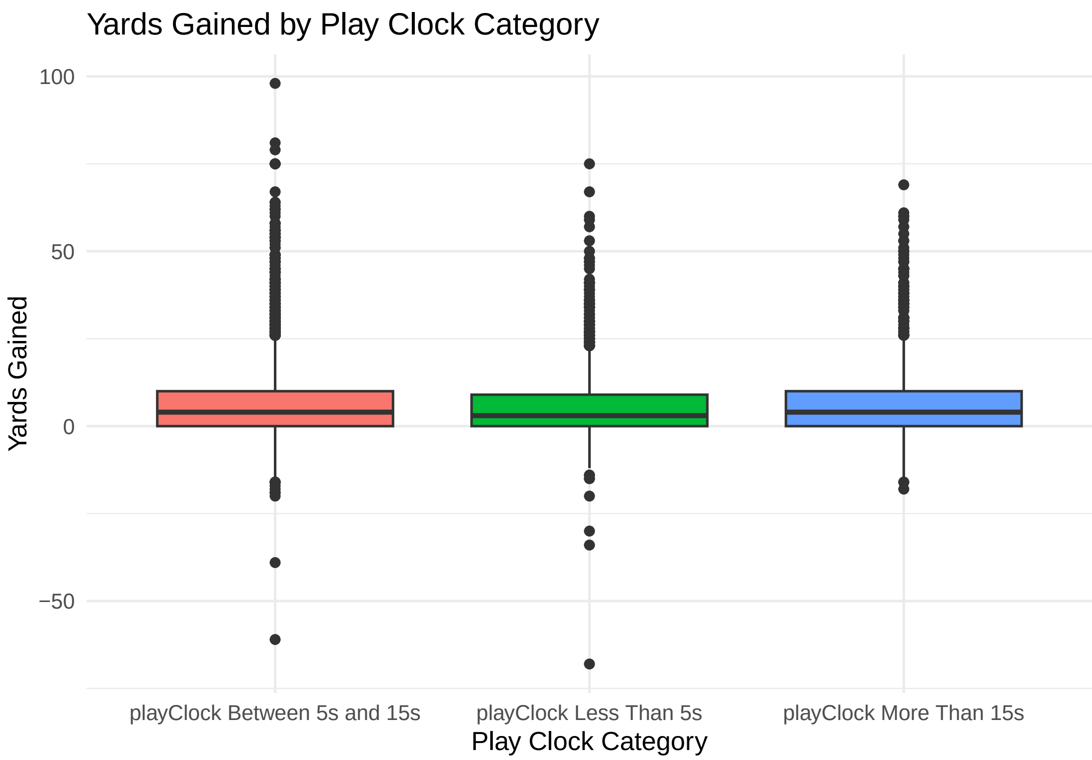
This is the most interesting find in my data. I categorized the playClockAtSnap into high, medium, and low. I then plotted the results of the different buckets to see the results. We can see that the lower the play clock, might have a higher likelihood of rushed plays and false starts. I was thinking that it could be correlated with poor play outcomes due to rushed decision making or how ready the defense is. On the other hand, I theorized that the higher play clock could be correlated with ready offenses while also potential catching the defense off guard. We can see that when the playClock is less than or equal to 5s, the results of the play are more spread out than than when the playClock is higher indicating that results vary more.

```
ggplot(plays_filtered, aes(x = playClockCategory, fill = passResult)) +
  geom_bar(position = "fill") +
  labs(
    title = "Play Outcomes by Play Clock Category",
    x = "Play Clock Category",
    y = "Proportion",
    fill = "Pass Result"
  ) +
  theme_minimal()
```



We also investigated the relationship that the play block at snap might have with the yards gained that play. However, as we can see in the below visualization, there is no clear correlation between the two.

```
ggplot(plays_filtered, aes(x = playClockCategory, y = yardsGained,
                           fill = playClockCategory)) +
  geom_boxplot() +
  labs(
    title = "Yards Gained by Play Clock Category",
    x = "Play Clock Category",
    y = "Yards Gained"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```



And we can see here, the playClock is indeed statistically significant with a value of 0.0002549 < 0.05. The next step would be to find a model to potentially predict results.

```
contingency_table <- table(plays_filtered$playClockCategory,
                           plays_filtered$passResult)

chi_sq_test <- chisq.test(contingency_table)
print(chi_sq_test)
```

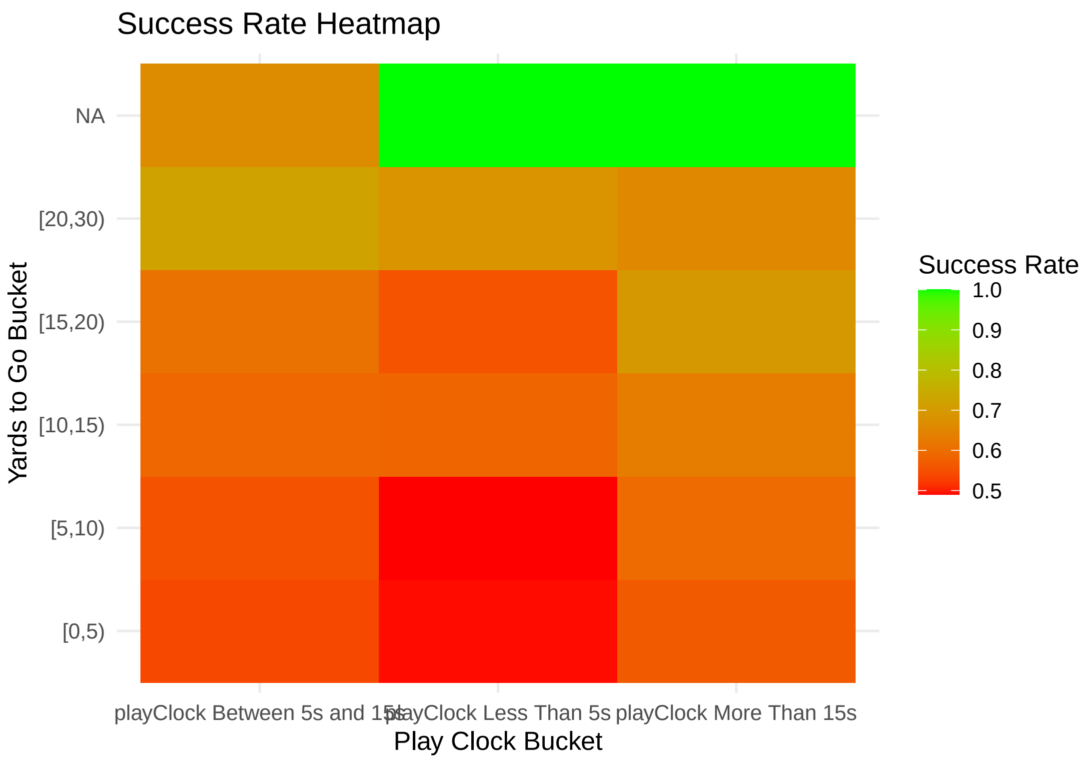
```
##
##  Pearson's Chi-squared test
##
## data:  contingency_table
## X-squared = 29.539, df = 8, p-value = 0.0002549
```

Further exploration of how the play clock affects other variables yields the following heat map. This heat map illustrates the play success rate based on the play clock and yards to go during that play.

```
heatmap_data <- plays_filtered |>
  mutate(yardsToGoBucket = cut(yardsToGo, breaks = c(0, 5, 10, 15, 20, 30),
                               right = FALSE)) |>
  group_by(playClockCategory, yardsToGoBucket) |>
  summarize(success_rate =
              mean(passResult == "C", na.rm = TRUE), .groups = "drop")

ggplot(heatmap_data, aes(x = playClockCategory, y = yardsToGoBucket,
```

```
                         fill = success_rate)) +
  geom_tile() +
  labs(
    title = "Success Rate Heatmap",
    x = "Play Clock Bucket",
    y = "Yards to Go Bucket",
    fill = "Success Rate"
  ) +
  theme_minimal() +
  scale_fill_gradient(low = "red", high = "green")
```



4 Modeling

Here for the predictive model I used a multinomial regression model because the prediction outcome is nominal with multiple categories which is perfect for this kind of model. The code block below makes the model and puts the predicted probabilities of the model back into the filtered plays dataframe as its own column.

```r
library(nnet)

plays_filtered <- plays_filtered %>%
  mutate(
    playClockCategory = factor(playClockCategory),
    passResult = factor(passResult)
  ) |>
  filter(!is.na(passResult), !is.na(playClockAtSnap))

multinom_model <- multinom(passResult ~ playClockAtSnap, data = plays_filtered)
```

```
## # weights:  15 (8 variable)
## initial  value 15667.878078
## iter  10 value 10546.466417
## iter  20 value 10303.646910
## final  value 10303.627223
## converged
```

```r
summary(multinom_model)
```

```
## Call:
```

```
## multinom(formula = passResult ~ playClockAtSnap, data = plays_filtered)
##
## Coefficients:
##    (Intercept) playClockAtSnap
## I   -0.5919611    -0.006360061
## IN  -3.2412784    -0.013173404
## R   -2.4540657    -0.018526994
## S   -1.9070770    -0.031906983
##
## Std. Errors:
##    (Intercept) playClockAtSnap
## I   0.04296130     0.003490795
## IN  0.13745234     0.011541819
## R   0.09674842     0.008267758
## S   0.07957039     0.007104210
##
## Residual Deviance: 20607.25
## AIC: 20623.25
```

```r
predictions <- predict(multinom_model, type = "probs")


nrow(plays_filtered)
```

```
## [1] 9735
```

```r
nrow(predictions)
```

```
## [1] 9735
```

```r
plays_filtered <- cbind(plays_filtered, predictions)
```

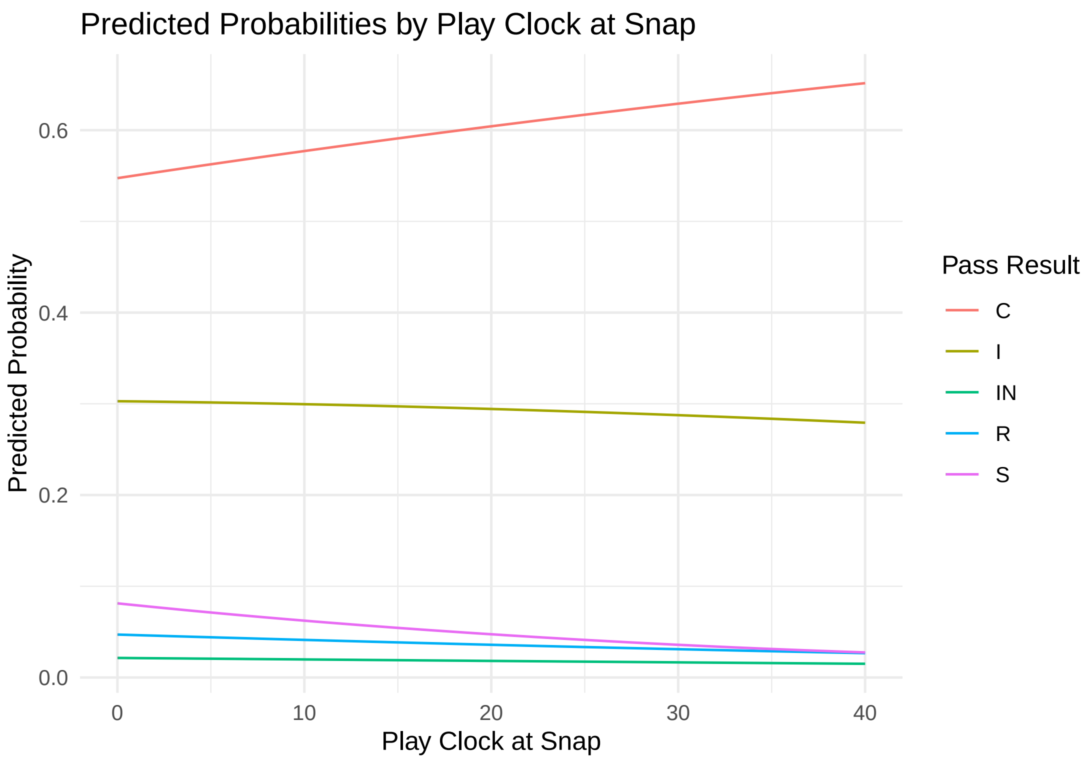
Here I plotted the predicted probabilities to visualize the model. As we can see, as the play clock at snap increases, the probability of completing a pass also increases from the prediction of the model. This is reinforcing what we saw before on the exploratory data histogram of more completions when the play clock is higher. We can also see that the probabilities of the other categories are decreasing which is also parallel to what we saw before we fitted the model. Next we should discuss that this predictor may not be the best predictor and model for pass results due to the low explanation of variance from the model.

```r
predicted_probs_df <- plays_filtered %>%
  select(playClockAtSnap, passResult, C, I, IN, R, S) %>%
  pivot_longer(
    cols = C:S,
    names_to = "Predicted_Class",
    values_to = "Probability"
  )

ggplot(predicted_probs_df, aes(x = playClockAtSnap, y = Probability, color =
                                 Predicted_Class)) +
  geom_line(stat = "smooth", method = "loess") +
  labs(
    title = "Predicted Probabilities by Play Clock at Snap",
    x = "Play Clock at Snap",
    y = "Predicted Probability",
    color = "Pass Result"
```

```
  ) +
  theme_minimal()
```

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



5 Prediction and Model Evaluation

6 Discussion