

An Examination of Timeout Value, Strategy, and Momentum in NCAA Division 1
Men's Basketball

A Senior Project Report
Presented to
The Faculty

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Science
in
Applied Mathematics
Yale University

Luke Benz

April 2019

Acknowledgements

I would like to thank my advisor, Dr. Xiaofei Wang, for her guidance and mentorship on this project. This project would not be possible without her constant feedback and suggestions that vastly improved the overall quality of this work. I would also like to thank Dr. Joshua Warren, who helped inspire some of the work presented in this thesis through discussions regarding a final project for his Fall 2018 “Bayesian Statistics” course. Finally, I owe a tremendous thank you to past mentors of mine: Dr. Elizabeth Claus at the Yale School of Public Health and Drs. Antonio Possolo and Andrew Rukhin at the National Institute of Standards and Technology. I could not be where I am today without their support.

Table of Contents

Introduction	1
1.1 Review of NCAA Timeout Rules	3
Chapter 2: Data	7
2.1 Additional Data Cleaning Notes	8
2.2 Imputation of Missing Pointsreads	9
Chapter 3: Timeouts As Local Augmentation in Win Probability Prediction	11
3.1 Win Probability Model Framework	11
3.2 Model Building	12
3.3 Model Assesment	14
3.4 Implications on Timeout Value and After Timeout Momentum	16
Chapter 4: After Timeout Momentum	21
4.1 How To Measure Momentum	21
4.2 Examining the After Timeout Effect	22
4.3 Points Above Expectation After Timeouts	30
Conclusion	41
5.1 Discussion of Results	41
5.2 Limitations	42
5.3 Future Work	43
Appendix A: Code	45
Appendix B: Code for Graphics and Tables	63
References	81

List of Tables

3.1	Win Probability Model Structure	12
4.1	Pre and Post Non-Media Timeout Summary Statistics	24
4.2	Pre and Post Media Timeout Summary Statistics	24
4.3	Summary Table of Post Timeout Improvement	38
4.4	Summary Table of Points Above Expectation	39

List of Figures

1.1	Histogram of Timeout Occurences	4
1.2	Histogram of Non-Media Timeout Occurences	5
2.1	Comparison of Imputed Point Spreads and Vegas Point Spreads . . .	10
3.1	Win Probability Model 0 Coefficient Estimates and 95 Percent Confidence Intervals	14
3.2	Win Probability Model 1 Coefficient Estimates 95 Percent Confidence Intervals	15
3.3	Model Log Loss on Test Set	16
3.4	Model Log Loss and Misclassification Rate on Test Set Over Time . .	17
3.5	Localized Improvement in Win Probability Prediction	19
4.1	Boxplots for Pre and Post (Non Media) Timeout Net Score Differentials	23
4.2	Boxplots for Pre and Post (Media) Timeout Net Score Differentials .	25
4.3	Histograms of Improvement in Net Score Differential Following Non-Media Timeouts	26
4.4	Histograms of Improvement in Net Score Differential Following Non-Media Timeouts	27
4.5	Post-Timeout Improvement Following Non-Media Timeouts Over the Course of the Game	28
4.6	Post-Timeout Improvement Following Media Timeouts Over the Course of the Game	29
4.7	Mixed Effects Model Summaries	31
4.8	Boxplots for Points Above Expectation after Timeouts	32
4.9	Boxplots for Points Above Expectation after Timeouts Following Negative Scoring Runs	33
4.10	Points Above Expectation Throughout the Course of a Game (Non-Media Timeouts)	34
4.11	Points Above Expectation Throughout the Course of a Game (Media Timeouts)	35
4.12	Value of Taking a Timeout	37

Abstract

Fans watching a basketball game often have a belief that they can sense when one team has “momentum”. Coaches seem to take timeouts when their team is on a negative scoring run, feeling pressure to stop an opponent’s quick flurry of scoring. This thesis seeks to examine how timeouts are used in NCAA Division 1 men’s basketball and whether there is any truth to the notion that timeouts stop opponent momentum by decreasing the rate of opponent scoring or swinging the rate of scoring in favor of the timeout-calling team. Additionally, this work attempts to quantify the value of a taking a timeout throughout the course of the game. The Introduction provides further background into the problem of interest, while Chapter 2 discusses the data used in this project. The utility of timeout indicators in providing local prediction augmentation for in-game win probability models is discussed in Chapter 3. Chapter 4 tackles the primary question of interest, examining after-timeout momentum explicitly. General discussion of findings as well as limitations of the work and plans for future research are presented in the Conclusion.

Introduction

The sport of basketball is often described as “a game of runs”. A typical basketball game will often have several stretches when one team scores the majority of points in a short portion of the game, only to be followed by the other team answering with a scoring run of its own. While the National Basketball Association (NBA) has the least inherent randomness among the four major North American professional sports (Lopez et al., 2018), unpredictability pervades National Collegiate Athletics Association (NCAA) Division 1 men’s basketball. Variance in game outcomes is the reason that March Madness, the NCAA men’s basketball championship tournament, attracts millions of viewers from across the country, many of whom have little to no interest in sports the rest of the year.

There are several potential reasons why college basketball is less predictable than the NBA. To begin with, there are 353 schools sponsoring Division 1 men’s basketball teams, compared to only 30 NBA teams, meaning the spread of talent is more uneven. Playoffs in college basketball, both in the form of conference postseason tournaments and March Madness, utilize a single elimination format. In the NBA, each playoff round is a best of seven series, making upsets significantly more unlikely. In addition to higher variance in game outcomes at the season level, there exists more inherent randomness at the in-game level as well. College basketball games are shorter in length (40 minutes vs. 48 minute games in the NBA), and the skill of the average college player is much worse than that of the average NBA player. While at the end of the day, the hoop is still 10 feet tall, these fundamental differences between college basketball and basketball at the professional level seem like likely reasons for the increased variance in outcomes at the college level.

One way that coaches try to swing a game in favor of their team is through the use of timeouts. Teams often take timeouts when their opponent is on a scoring run, with the objective of stopping the opponent’s scoring run. The majority of coaches likely think that by stopping the flow of game play, they are somehow able to disrupt their opponent and effectively reset the momentum of the game. Do timeouts really influence “momentum”, or is what coaches, players, and fans perceive as momentum really just the product of an inherently random game?

The study of momentum and streaks in the sport of basketball is not new by any means. Perhaps the most well known examination of momentum in basketball is that of the hot hand theory, the idea that a player who has made several shots in a row has “found his groove” and is more likely to make future shots. The hot hand is something that many players, as well as spectators of the game, seem to believe in. In

the first, and most famous paper to examine the hot hand theory, (Gilovich, Vallone, & Tversky, 1985) found no evidence for this existence of player shooting momentum. More recent papers, such as (Bocskocsky, Ezekowitz, & Stein, 2014) and (Miller & Sanjurjo, 2018) have proposed new ways to measure the hot hand effect, and suggest that there is some truth to the hot hand theory. Nevertheless, views on the existence of the hot hand, and on the existence of momentum in sports more generally, remain somewhat divided with no clear consensus on how momentum should be measured or even what the term momentum refers to.

Previous research regarding the potential relationship between timeouts and momentum exists, but to date, no large scale study of the potential relationship between timeouts and momentum exists about college basketball. (Mace, Lalli, Shea, & Nevin, 1992) analyzed a sample of 14 college basketball games, and measured the ratio of reinforcers (points and turnovers forced) to adversities (missed shots, turnovers conceded, and fouls), examining game event outcomes on possessions immediately following an adversity. They concluded that ratio of positive response to adversity for the opposing team compared to the team calling the timeout dropped on average from 2.63 in a 3 minute interval proceeding the timeout to 1.11 in a 3 minute interval immediately following the timeout. Another study (Roane, Kelly, Trosclair, & Hauer, 2004) replicated the study of (Mace et al., 1992) on a sample of 6 NCAA women's basketball games and obtained similar findings that timeouts on average yielded a better ratio of positive response to adversity for the timeout-calling team. (Permutt, 2011) examined a much larger sample of over 3000 NBA games and concluded that the team calling a timeout received a benefit of about 0.2 points from calling the timeout. This analysis was very limited in scope however, and only looked at timeouts taken in the first half following a 6-0 scoring run. Moreover, no consideration was given to other potentially important covariates, such as the time of game during which the timeout was taken, the relative team strengths, and the current score differential at the point of taking a timeout. A 2008 Wall Street Journal article (Weinbach, 2008) examined how NBA teams performed in the two possessions immediately following a timeout and showed that different teams had different rates of success following timeout situations, though made no comment on the overall value of taking a timeout.

Clearly there is a lot of room to improve upon the existing literature regarding the potential relationship between timeouts and momentum in college basketball. Moreover, there are several differences between NBA and NCAA basketball timeout rules that render analyses of post-timeout momentum in the two different leagues effectively distinct. To begin with, NCAA basketball is played in two halves as opposed to the four quarters of an NBA game, and college teams have fewer timeouts at their disposal than do NBA teams. There also exists significant differences in the structure of media timeouts (breaks in play where the televised broadcast cuts to commercial). Lastly, unlike NBA rules, NCAA rules do not allow teams to advance the ball up the court following a timeout. A full review of NCAA timeout rules is presented in the following section.

The purpose of this paper is to provide an in-depth look at the timeout in NCAA Division 1 men's college basketball. Attempts will be made to answer the question of whether timeouts do indeed stop opponent momentum, and to quantify the value of

the timeout over the course of the game. A large portion of this problem is defining in mathematical terms what is meant by momentum. Chapter 2 details the data used in this project. Chapter 3 explores timeout indicators as local augmentations in win probability predictions. Chapter 4 examines the relationship between pre- and post-timeout scoring runs in order to test for the existence of timeouts' momentum-stopping capability, as well as quantifying the value of the timeout over the course of the game. Discussion of results and remarks on limitations of the work are presented in the Conclusion.

1.1 Review of NCAA Timeout Rules

Throughout the paper, references will be made to specific NCAA timeout terms and rules, so to get the reader up to speed, a summary of NCAA basketball rule 5-14.10, governing timeout usage, is presented below (NCAA, 2018).

1. Each team has 4 timeouts over the course of the game. 3 timeouts last 30 seconds each and 1 timeout lasts 60 seconds.
2. One additional 30 second timeout is granted to each team at the start of the second half, though teams can begin the second half with no more than 3 30 second timeouts. Thus, a team does not gain back an additional 30 second timeout if it did not use one in the first half. For this reason, the act of taking a 30 second timeout right before halftime (when no prior timeouts have been called) is colloquially referred to as a “use it or lose it” timeout.
3. There is a media timeout at the first dead-ball stoppage of play after every multiple of 4 minutes. That is, the first stoppage of play under 16 minutes, 12 minutes, 8 minutes, and 4 minutes respectively cues an official media timeout. This paper uses the terms media timeout and TV timeout interchangeably.
4. Barring some technical details which can be read in (NCAA, 2018), the only exception to rule 3 is when a team takes a timeout within 30 seconds of where a media timeout could occur. That is, if a team takes a timeout with between 12:00-12:30 remaining on the game clock, the existing timeout replaces the normal media timeout that would occur at the first dead ball under 12 minutes.
5. Each team is given one additional 30 second timeout per overtime period, and carries all remaining timeouts from the end of second half/overtime period into the next overtime period.

While data used in this thesis make a distinction whether a particular timeout was called by a coach (non-media timeout) or was a television timeout (media timeout), no distinction is made between non-media timeouts that are 30 seconds or 60 seconds in length. As such, analyses in this work treat all non-media timeouts as identical despite the fact there may be subtle differences between the value and effect possessed by the longer timeouts.

Histograms are presented below showing the distributions of the frequency when timeouts occur over the course of the game. Figure 1.1 includes both media and non-media timeouts, while Figure 1.2 is limited to non-media timeouts only. The

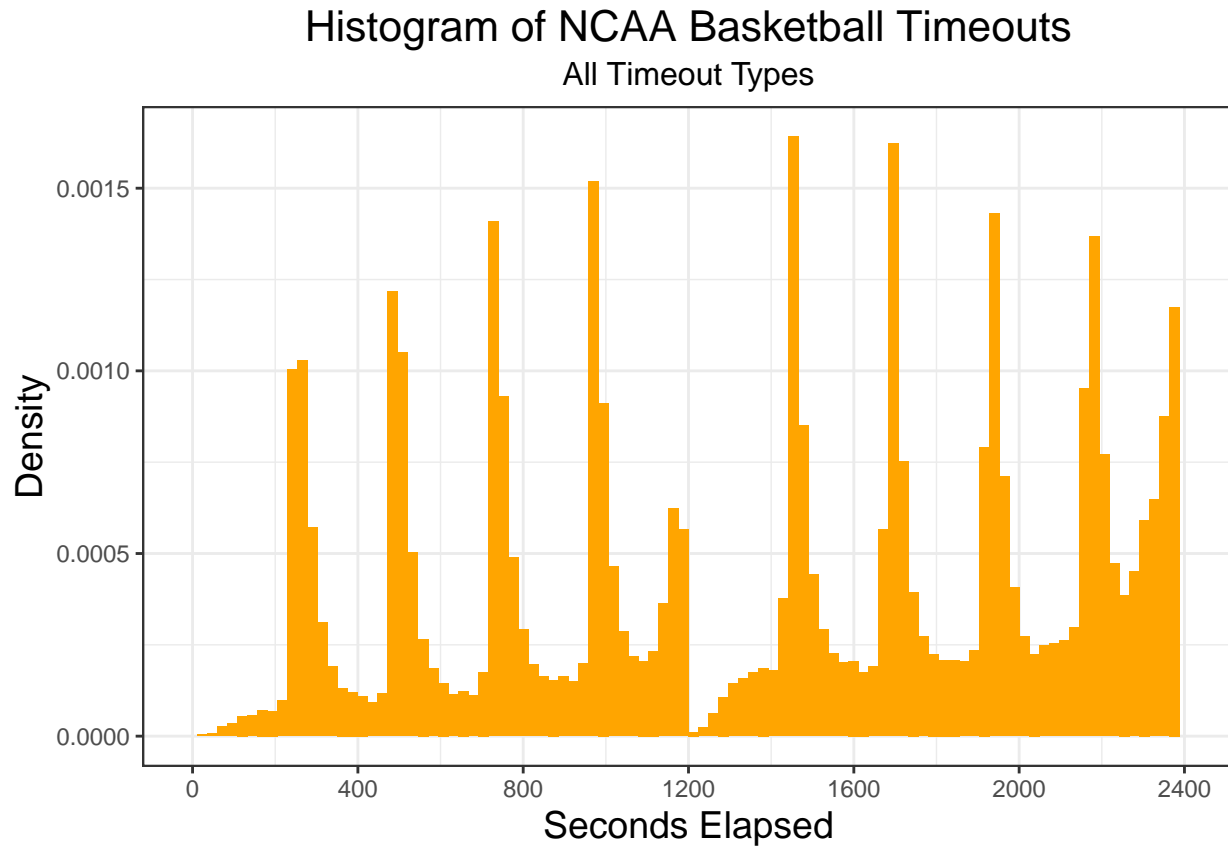


Figure 1.1: Histogram of Timeout Occurences

vertical lines in Figure 1.2 denote the 4 minute intervals which designate the first possible instances of media timeouts, as outlined in bullet point 3 in the above list.

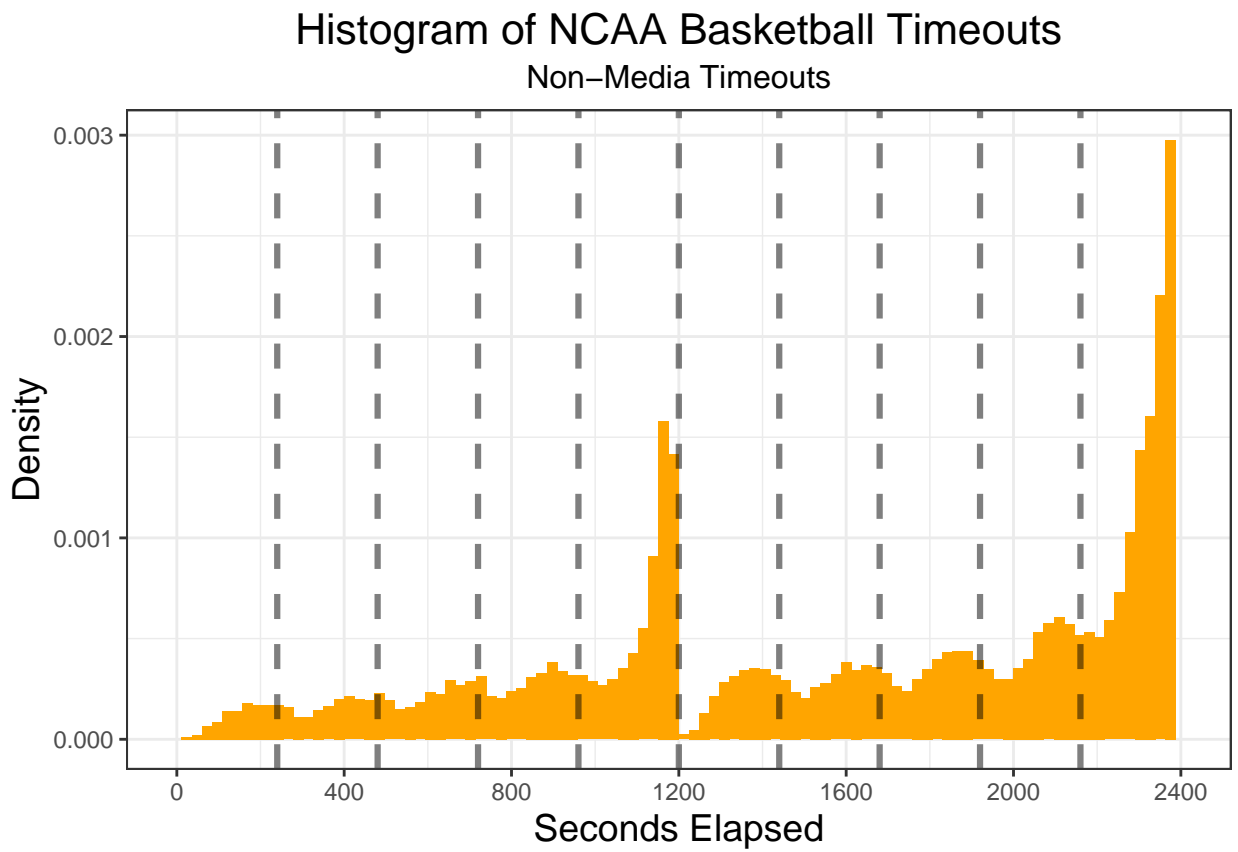


Figure 1.2: Histogram of Non-Media Timeout Occurences

Chapter 2

Data

The data used in this project are 10,409 complete NCAA basketball play-by-play logs from the 2016-17 and 2017-18 seasons, scraped from ESPN.com using the `ncaahoopR` package (Benz, 2019). Each play-by-play log consists of over 20 variables with information about each play of the game, including the time at which the play took place, each team's score as a result of the play, and a text description of what took place during the play. Before any further analysis, the following variables from the play-by-play logs were kept/created:

- `game_id`: The unique ESPN identifier for the game in question.
- `date`: The date on which the game was played.
- `team`: The home team as listed on ESPN.
- `opponent`: The away team as listed on ESPN.
- `play_id`: A unique play identifier associated with each row in the game log.
- `half`: The half during which each play took place
- `secs_remaining`: The number of seconds remaining in the game, as would be shown on a scoreboard.
- `secs_elapsed`: `2400 - secs_remaining`.
- `secs_remaining_absolute`: The number of seconds remaining to be played, including any additional overtimes.
- `team_score`: Number of points scored by `team`.
- `opp_score`: Number of points score by `opponent`.
- `score_diff`: `team_score - opp_score`
- `timeout_ind`:
 - 1 if `team` took a timeout in previous 60 seconds of game time and `opponent` did not.
 - -1 if `opponent` took a timeout in previous 60 seconds of game time and `team` did not.
 - 0 if both teams took a timeout in previous 60 seconds of game time or neither team took a timeout in previous 60 seconds of game time.
- `timeout_diff`: The difference in the number of timeouts `team` has remaining compared to `opponent`.

- **avored_by**: The number of points **team** was favored by over **opponent**. When available on ESPN, this is the Las Vegas pre-game point spread. When missing, this value is imputed as described in Section 1.2.
- **description**: Text description of game event.
- **win**: An indicator variable whether **team** beat **opponent**.

In cases where games are played at neutral sites, **team** is given to the second team listed on ESPN, where the home team would normally be listed, and **opponent** is assigned to the team listed first. Unfortunately, **ncaahoopR** does not yet make available the information whether or not a particular game was played at a neutral site. Games were not duplicated with **team** and **opponent** switched in order to get data from both teams perspectives, as doing so would no longer allow for the treatment of games as independent of one other. Moreover, including duplicate game logs with switched team perspectives can artificially decrease variance of coefficient estimates in any model trained on the data, leading to overly confident model fitting (Wang, 2015).

The fact that **team** is home in the majority of cases is not problematic for two reasons. To begin with, home court advantage, which has been shown to be between 3-4 points (Pomeroy, 2018), is already factored into the **avored_by** variable. This is the case for both Las Vegas point spreads provided by ESPN and those that were imputed, as described in Section 1.2. Furthermore, when looking at the relationship between timeouts and momentum in Chapter 4, variables are flipped (not duplicated) when necessary so that everything can be viewed from the perspective of the timeout-taking team. More detail on this is provided in Chapter 4.

2.1 Additional Data Cleaning Notes

Some additional clarification is required on the relationship between the **secs_remaining**, **secs_remaining_absolute**, and **secs_elapsed** variables. **secs_remaining** is a direct conversion of the time remaining in regulation to seconds. The beginning of every game, which is 40 minutes in length, has **secs_remaining** = 2400, while halftime sees **secs_remaining** set to 1200, and so on. Should the game go to overtime, where each overtime period adds an additional 5 minutes, **secs_remaining** is reset to 300 seconds and counts down the time remaining in each respective overtime period. No knowledge of any future overtime periods is included in the coding of this variable, as future overtime(s) is not something a player or coach would know about until the end of regulation. Knowledge of future overtime periods is, however, factored into the **secs_remaining_absolute** variable. In games where no overtime is played, **secs_remaining** and **secs_remaining_absolute** are identical. In games with overtime, **secs_remaining_absolute** is a shifted version of **secs_remaining** such that the end of the game is given value 0 and the start of the game is given the value of the total number of seconds in the game between regulation and overtime combined. For games with a single five minute overtime period, **secs_remaining_absolute** would begin with values 2700 seconds (2400 seconds remaining in regulation + 300 seconds of overtime). Lastly, **secs_elapsed** is

simply an inverted version of `secs_remaining` given by the formula `secs_elapsed = 2400 - secs_remaining`. The advantage of this encoding is that the start of the game is given value 0. Throughout this paper, `secs_remaining` will typically be used in models and calculations, as expressing things terms of time remaining in the game is how coaches and players think. `secs_elapsed` will be used in plots in order to have more intuitive, easier to understand plots. This will always be explained explicit throughout the text, but the similarity of these variables can easily confuse a reader if not careful.

In addition to 10,409 game data set spanning the 2016-17 and 2017-18 seasons, an additional 3,625 games from the 2018-19 season (through February 3rd, 2019) are used as a test set in Chapter 2. Unless otherwise noted, any reference to training set refers to the 10,409 game data set spanning the 2016-17 and 2017-18 seasons and any reference to test set refers to the 3,625 game set from the 2018-19 season.

2.2 Imputation of Missing Pointsreads

Of the 10,409 games in the primary set used in this analysis, 7,369 had a pre game point spread from Las Vegas listed on ESPN. When games on ESPN do not have listed point spreads, the `ncaahoopR` package (Benz, 2019) makes an attempt to impute the missing value. The methodology for such an imputation is outlined in (Benz, 2018b), which is a variation of the Massey method (Massey, 1997) for score differential prediction in sports. To briefly summarize, in a game between team i and team j , the score differential S_{ij} (team i score - team j score) is modeled as follows:

$$\begin{aligned} S_{ij} &= \beta_i - \beta_j + \rho_l + \epsilon_{ij} \\ \epsilon_{ij} &\sim N(0, \sigma^2) \end{aligned}$$

where β_i and β_j are team-strength parameters for teams i and j respectively, and ρ_l is a location parameter denoting whether team i is playing the game in question at home, away, or at a neutral site. The team strength parameter β_i can be interpreted as the number of points team i is relative to the average college basketball team on a neutral floor. Estimates of team strength and location parameters are found by fitting a linear regression model to the matrix of score differential, with covariates team, opponent, and location, as outlined in (Massey, 1997). Previous games are given weight inversely proportional to the recency of the game according to the weighting function outlined in (Benz, 2018b).

When no pre-game line is available for a game on ESPN, `ncaahoopR` imputes it with \hat{S}_{ij} using the model outlined above. The model from (Benz, 2018b) only considers games between Division 1 teams, so games where a Division 1 team plays a non-Division 1 opponent are unable to be imputed by this method. Of the 3040 games without an available point spread in the primary set for the purposes of this project, 2012 were successfully imputed, and 1028 remained missing. Of the 3,625 games in the 2018-19 test set, 499 pre game point spreads were missing, of which 146 were successfully imputed, leaving 353 still missing.

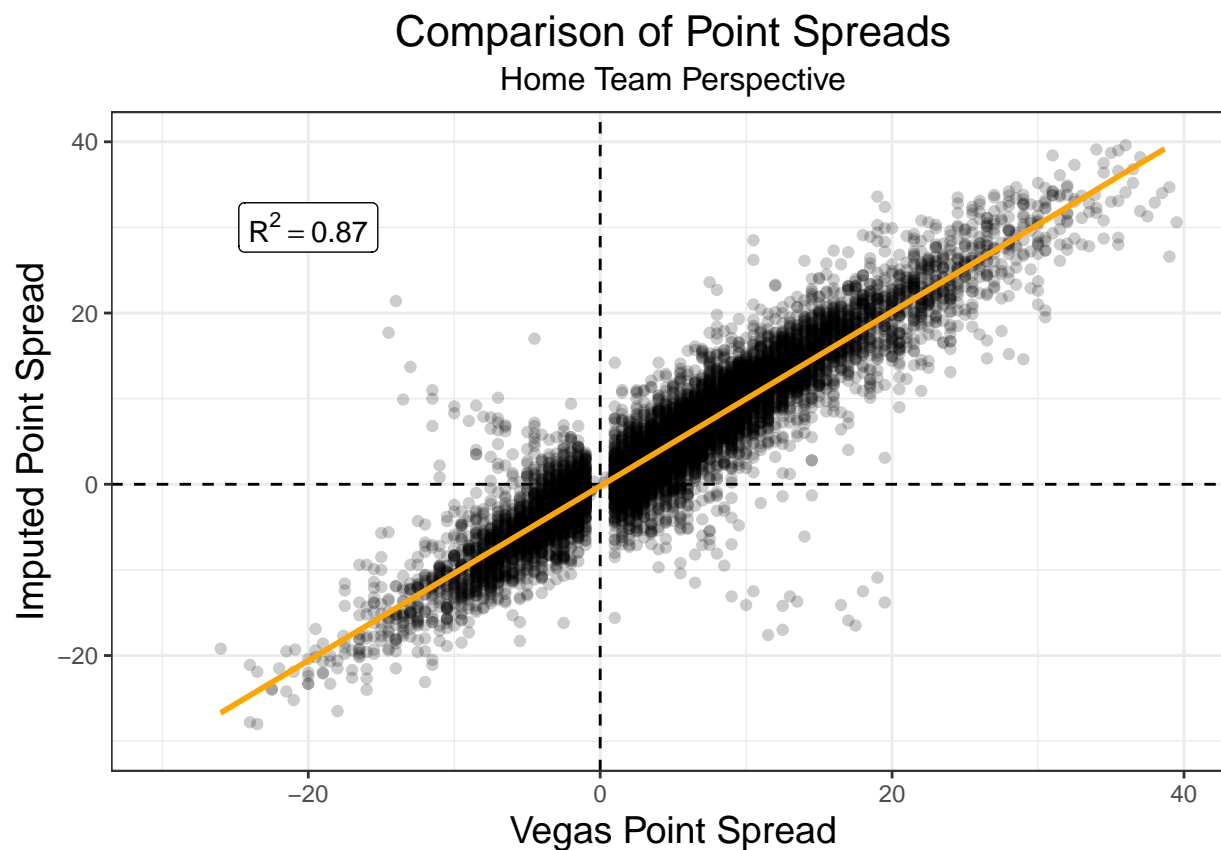


Figure 2.1: Comparison of Imputed Point Spreads and Vegas Point Spreads

To demonstrate that this method effectively models the behavior of Las Vegas point spreads, Figure 2.1 shows a comparison between Vegas point spreads available for all training/test set games and point spreads imputed for those games. In general, this model offers a good proxy ($R^2 = 0.87$) for imputing point spreads in the few cases they are unavailable, especially given that many Vegas point spreads are generated in a similar model based fashion.

Chapter 3

Timeouts As Local Augmentation in Win Probability Prediction

3.1 Win Probability Model Framework

A win probability model estimates team i 's chances of beating team j throughout the course of a game. That is, the estimated odds that team i wins the game are updated after each play of the game in question. More formally, let p_{ikt} denote the probability that team i wins game k with time t remaining in the game. Let Y_{ik} be an indicator whether team i won the game k . The following model is assumed:

$$\begin{aligned} Y_{ik}|p_{ikt} &\sim \text{Bernoulli}(p_{ikt}) \\ \text{logit}(p_{ikt}) &= \mathbf{X}_{ikt}^T \boldsymbol{\beta}_t + \epsilon_{ikt} \\ \epsilon_{ikt} &\sim N(0, \sigma^2) \end{aligned}$$

In the above model, observations are of the form $(\mathbf{X}_{ikt}, Y_{ik})$, where \mathbf{X}_{ikt} denotes a vector of covariates of interest with time t remaining in game k and, Y_{ik} is an indicator whether team i won ($Y_{ik} = 1$) or lost ($Y_{ik} = 0$) the game in question. \mathbf{X}_{ikt} occur at discrete, non-regular time points t , as observations are only available after the occurrence of certain game events (such as a made basket, foul, turnover, or timeout) which don't adhere to a regular time schedule.

$\boldsymbol{\beta}_t$ represents the vector of coefficients for the covariates of interest with time t remaining. One challenge in fitting the above model is that the covariates of interest, such as score differential and pre-game point spread, have non-linear dependencies on the amount of time remaining in the game. While time remaining is not explicitly a covariate in any version of this model, coefficient estimates for covariates of interest are obtained at various values of time remaining t . The manner in which such estimates are obtained is discussed in Section 3.2.

To combat covariates' non-linear dependence on time, various versions of this type of college basketball win probability model have broken the game into discrete time chunks, over which the coefficients are assumed to be constant (Benz, 2018a; Burke, 2009; Torvik, 2017). However, numerous problems can arise when one categorizes or makes discrete a continuous variable (Bennette & Vickers, 2012; Lopez, 2018). To get

around this problem, several versions of this model were fit over discrete intervals which overlapped by 90%. LOESS smoothing was then applied to the resulting coefficient estimates to obtain smooth coefficient functions over time. Coefficient estimates represent the increase in the log odds of winning a game while holding all of the other covariates in the model constant.

3.2 Model Building

One way to begin to quantify the importance of timeouts is to look whether including variables related to timeouts in versions of the win probability model framework presented in Section 3.1 can improve win probability prediction. Several models were built under the framework of Section 3.1 using the following covariates and interval structure.

Table 3.1: Win Probability Model Structure

Model	Variables Included	Time Interval Structure ($t - \Delta, t]$
0	<ul style="list-style-type: none"> • Score Differential • Pre Game Point Spread 	<ul style="list-style-type: none"> • $\Delta = 100, 600 \leq t \leq 2400$ • $\Delta = 50, 100 \leq t \leq 600$ • $\Delta = 10, 10 \leq t \leq 100$ • $\Delta = 2, 1 \leq t \leq 10$ • $\Delta = 1, 0 \leq t \leq 1$
1	<ul style="list-style-type: none"> • Score Differential • Pre Game Point Spread • Timeout Indicator • Timeout Differential 	<ul style="list-style-type: none"> • $\Delta = 100, 600 \leq t \leq 2400$ • $\Delta = 50, 100 \leq t \leq 600$ • $\Delta = 10, 10 \leq t \leq 100$ • $\Delta = 2, 1 \leq t \leq 10$ • $\Delta = 1, 0 \leq t \leq 1$
2	<ul style="list-style-type: none"> • Score Differential • Pre Game Point Spread • Timeout Indicator • Timeout Differential 	<ul style="list-style-type: none"> • $\Delta = 60, 600 \leq t \leq 2400$ • $\Delta = 30, 100 \leq t \leq 600$ • $\Delta = 10, 20 \leq t \leq 100$ • $\Delta = 2, 1 \leq t \leq 20$ • $\Delta = 1, 0 \leq t \leq 1$

Model	Variables Included	Time Interval Structure ($t - \Delta, t]$
3	<ul style="list-style-type: none"> • Score Differential • Pre Game Point Spread • Timeout Indicator • Timeout Differential 	<ul style="list-style-type: none"> • $\Delta = 30, 300 \leq t \leq 2400$ • $\Delta = 15, 60 \leq t \leq 300$ • $\Delta = 3, 1 \leq t \leq 60$ • $\Delta = 1, 0 \leq t \leq 1$
4	<ul style="list-style-type: none"> • Score Differential • Pre Game Point Spread • Timeout Indicator 	<ul style="list-style-type: none"> • $\Delta = 100, 600 \leq t \leq 2400$ • $\Delta = 50, 100 \leq t \leq 600$ • $\Delta = 10, 10 \leq t \leq 100$ • $\Delta = 2, 1 \leq t \leq 10$ • $\Delta = 1, 0 \leq t \leq 1$

In Table 3.1, the time interval structure refers to the discrete time chunks over which coefficient estimates are obtained. For each interval of the form $(t - \Delta, t]$, the model from section 3.1 is fit on all data from the 10,409 game training set such that $t - \Delta \leq \text{time_remaining} \leq t$. Consecutive intervals overlap one another by 90%. As the time remaining, t approaches 0, the rate at which model coefficients change with time increases. As such, Δ is decreased in order to properly capture the manner in which these coefficients vary as a function of time.

Model 0 contains no timeout related variables, and as such, it will be referred to as the null model. It serves as a baseline to test whether timeout indicators and related timeout covariates can improve win probability prediction. Differing values of Δ in Models 1, 2, and 3 serve to test differing interval lengths over which to estimate $\hat{\beta}$. Smaller values of Δ reflect the assumption that β changes faster with time, and thus serve to give more granular estimates of $\hat{\beta}$. The time structure used to fit Model 3 is the most granular, followed by Model 2 and Model 1, respectively. Models 0 and 4 are fit with the same time interval structure as Model 1 using a different set of covariates.

When fitting Model 0 for example, the 10,409 game training set was filtered to include game data with between 2300-2400 seconds remaining. The logistic regression model in Section 3.1 was then fit on the filtered data to obtain coefficient estimates for that interval. This step was then repeated for the next 100 second interval, 2290-2390 seconds remaining, which overlapped the previous interval by 90%. The process is continued for all 100 second intervals until $t = 600$ seconds, at which point the interval length, Δ , drops to 50 seconds, and the process carries on in this fashion according to the specifications given in Table 3.1.

Figure 3.1 and Figure 3.2 show smoothed coefficient estimates over the course of a game for Model 0 and Model 1 respectively, with 95% confidence intervals for these estimates shown in light blue. Coefficient plots for Models 2 and 3 are nearly entirely

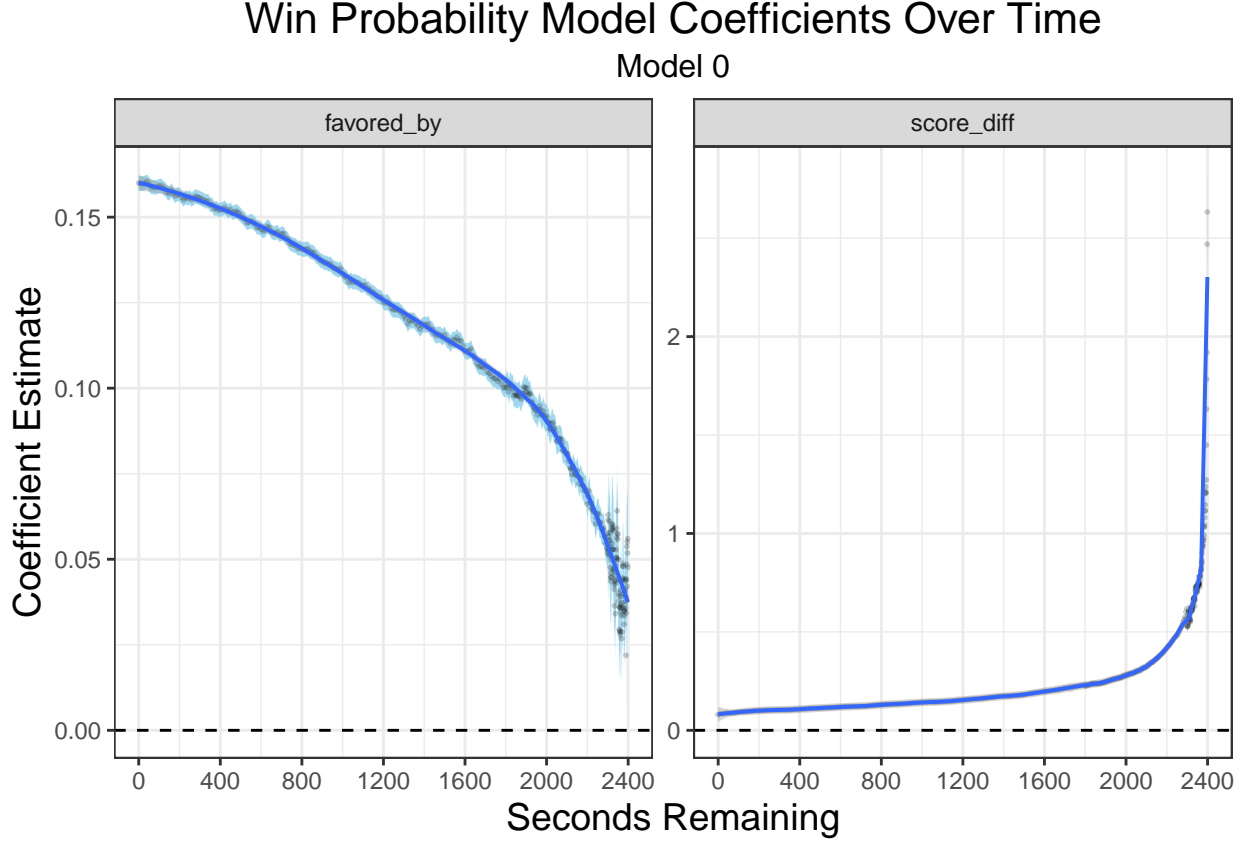


Figure 3.1: Win Probability Model 0 Coefficient Estimates and 95 Percent Confidence Intervals

identical to that of Model 1, and thus are not displayed so as not to present repetitive information. The coefficient plot for Model 4 is exactly the same as Model 1 except that it does not include the timeout differential variable in the model.

3.3 Model Assessment

In order to compare the various models and see if variables relating to timeouts can augment win probability prediction, each of the models built were predicted on the the 3,625 game test set from the first half of the 2018-19 season. Each model was fit using LOESS span parameter $\in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. Models were compared on the basis of log-loss.

Formally, let \hat{p}_{ikt_j} denote the predicted win probability for team i in game k at the j^{th} time stamp, t_j , and y_{ik} be an indicator whether team i won game k . The average log loss is then given by

$$-\frac{1}{N} \sum_{k=1}^N \frac{1}{n_k} \sum_{j=1}^{n_k} y_{ik} \log(\hat{p}_{ikt_j}) + (1 - y_{ik}) \log(1 - \hat{p}_{ikt_j})$$

where $N = 3,272$ denotes the number of games in the test set with available or

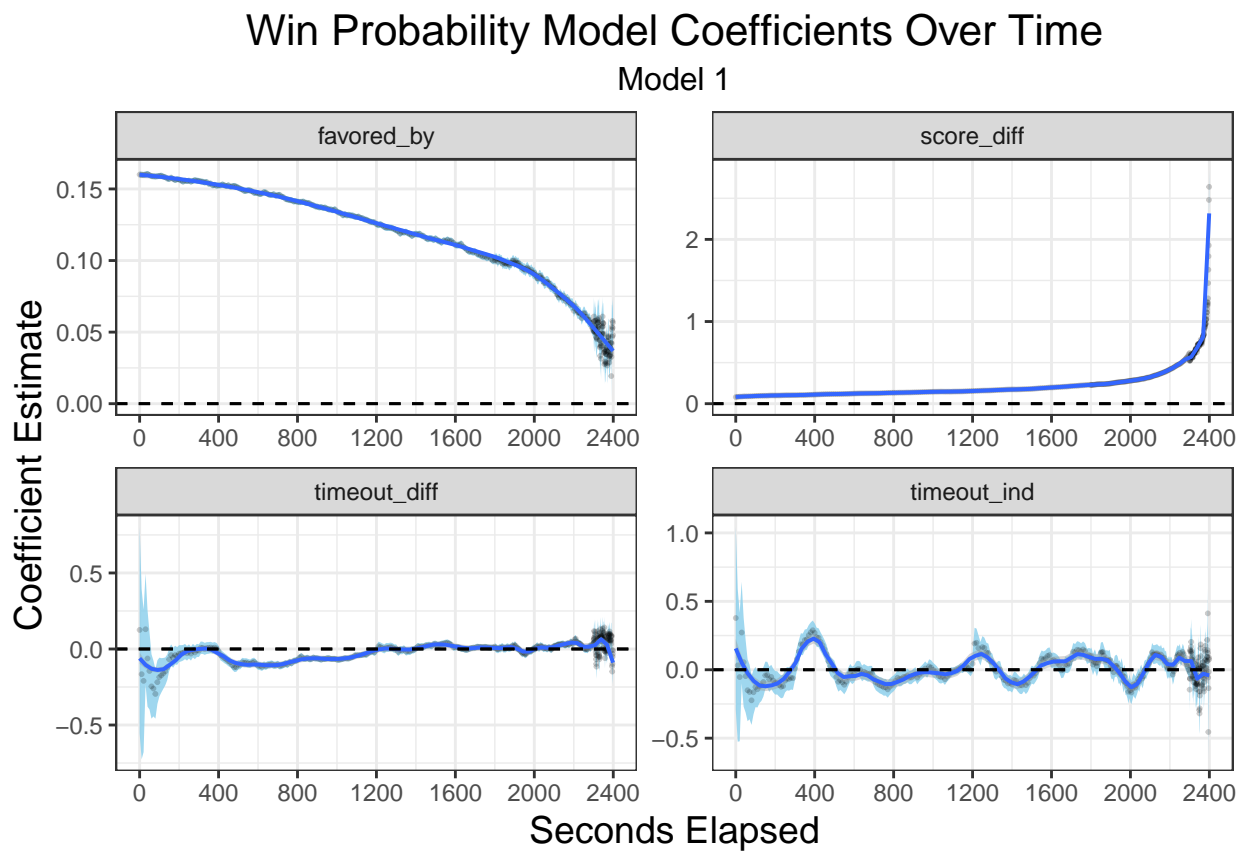


Figure 3.2: Win Probability Model 1 Coefficient Estimates 95 Percent Confidence Intervals

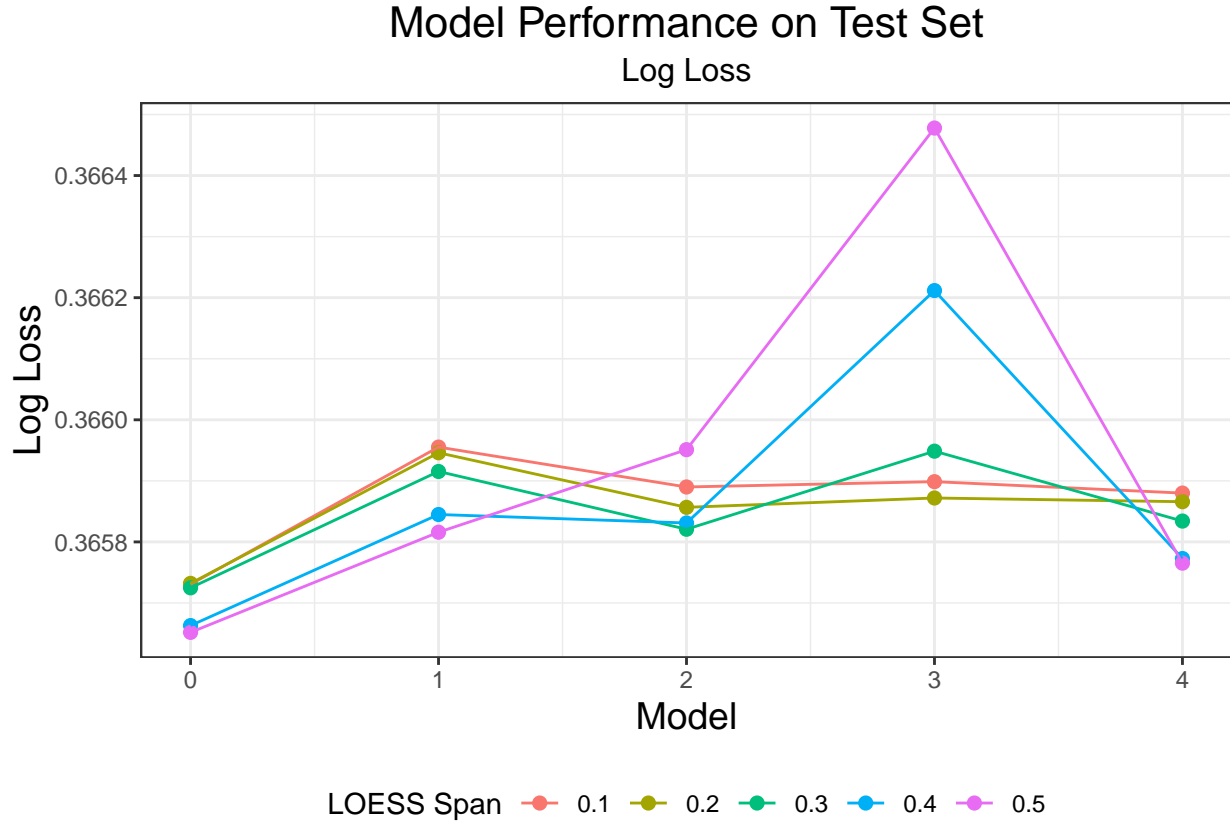


Figure 3.3: Model Log Loss on Test Set

imputed point spread, and n_k denotes the number of play time stamps available for game k .

Figure 3.3 shows the the log loss for each of the models with various LOESS span parameters. Overall the best null model was Model 0 with LOESS span = 0.5 and the best non-null model was the span = 0.5 version of Model 4. In addition to 90% overlapping intervals, 50% and 75% percent overlapping intervals were examined, but decreasing the intervals only decreased the predictive accuracy of any model.

3.4 Implications on Timeout Value and After Timeout Momentum

Estimates of \hat{p}_{ikt} become more accurate over the course of the game, as the current score differential becomes more and more important in determining which teams win the game the less time there is remaining. For example, all other factors being equal, a team holding a 5 point lead with 30 seconds left in the game is much more likely to win than a team holding a 5 point lead with 30 minutes left in the game. Thus, it is of interest to compare how the best models performed during specific portions of game time.

While the null model (Model 0) performed best globally, one might wonder if

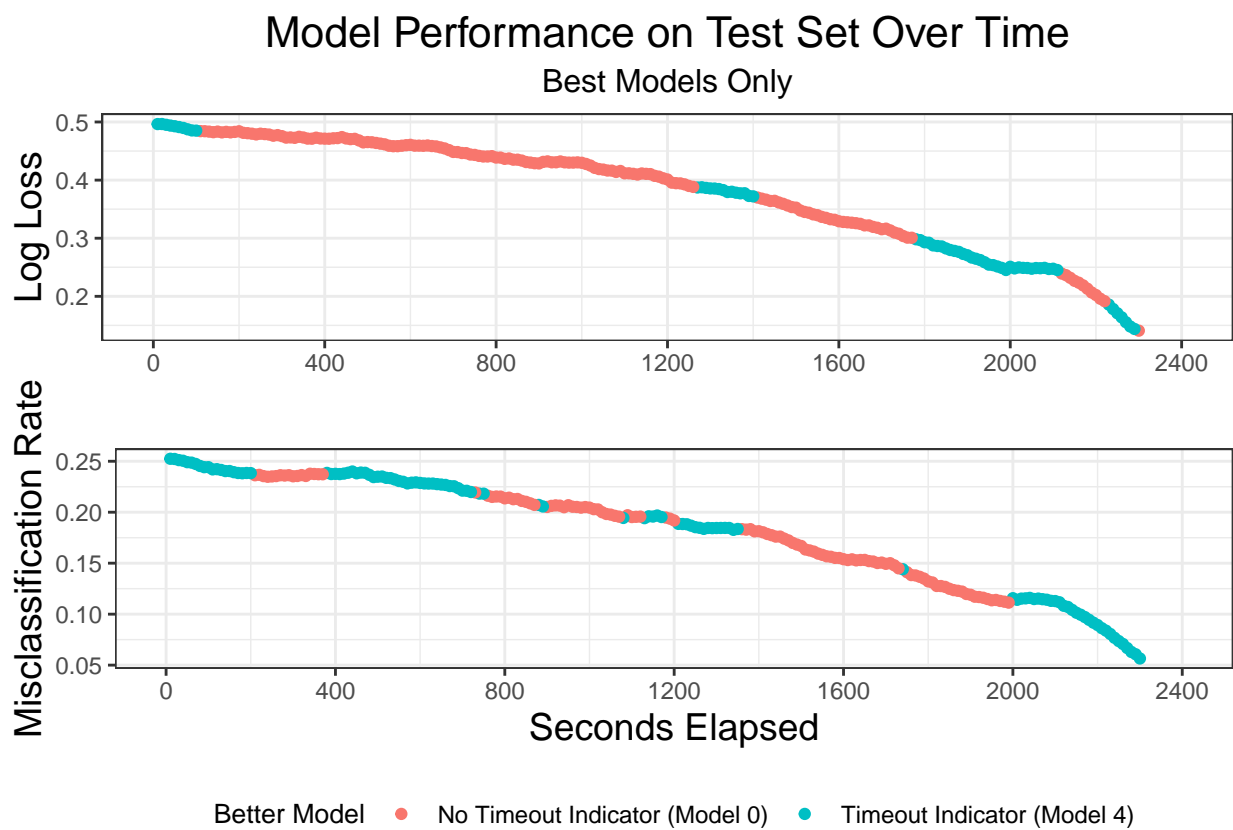


Figure 3.4: Model Log Loss and Misclassification Rate on Test Set Over Time

including the timeout indicator (`timeout_ind`), as Model 4 does, might improve win probability prediction locally. To test this question, each game was broken up into 100 second intervals of the form $(t - 100, t]$ with 90% overlap, and the log loss was calculated for that interval. For example, the log loss for each of the two best models was computed for win probability estimates made with 2300-2400 seconds remaining in the game, then for estimates made with 2290-2390 seconds remaining in the game, and so on down until the interval 0-100 seconds remaining in the game. Note that while these intervals may seem similar to those used to fit the various models in Section 3.1, they are unrelated. Rather, the 100 second rolling intervals serve to ensure that each portion of game time has enough observations to adequately estimate log loss during that particular portion of the game.

The minimum log loss between these two models is shown as a function of time in Figure 3.4. In order to give a more intuitive sense of how the best models compare, Figure 3.4 also presents an evaluation of the models on the basis of misclassification rate over time. Misclassification rate is calculated by rounding each \hat{p}_{ikt} to either 0 or 1 and computing the rate with which those rounded values disagree with the true y_{ik} . Interestingly, including an indicator of whether a team or its opponent took a timeout in the previous minute of game play can lead to improved win probability, especially at the beginning and end of each half.

As Figure 3.5 shows, the regions where including the `timeout_ind` variable improves prediction on the basis of misclassification rate nicely align with regions where the coefficient estimate differs significantly from 0. It's worth noting that both of the best models performed very similarly on the basis of log loss and misclassification rate. In fact, when plotting the log loss curves for both models on top of one another, the two are nearly indistinguishable from one another. Nevertheless, the regions where one model outperforms the other are continuous chunks of time that are not insignificant in length.

One must be very careful with the direction of causality, however, and not think that taking a timeout at certain times of the game necessarily makes a team more or less likely to win the game. In the beginning of the game, the sign of the `timeout_ind` is negative. This is not to suggest that taking a timeout early in the game reduces a team's chances of winning the game. In all likelihood, a team must be getting heavily outscored to start the game in order to be in a situation where a timeout is taken that early in the game. They are less likely to win because they are getting outscored, not because they decided to take the timeout. Nevertheless, there are some very interesting associations between the smoothed `timeout_ind` coefficient function and win probability. The fact that the coefficient changes from positive to negative in a significant manner several times through the course of a game suggests that certain timeouts in certain portions of the game are more strongly associated with winning than other times of the game.

In all likelihood, the reason that the null model, with no timeout indicator variable, performs better globally is that including it yields over-fitting at certain portions of the game. Nevertheless, the fact that including `timeout_ind` yields local augmentation in prediction accuracy is strongly suggestive that timeouts need to be studied in a localized fashion rather than a global manner. In examining the effect of taking a

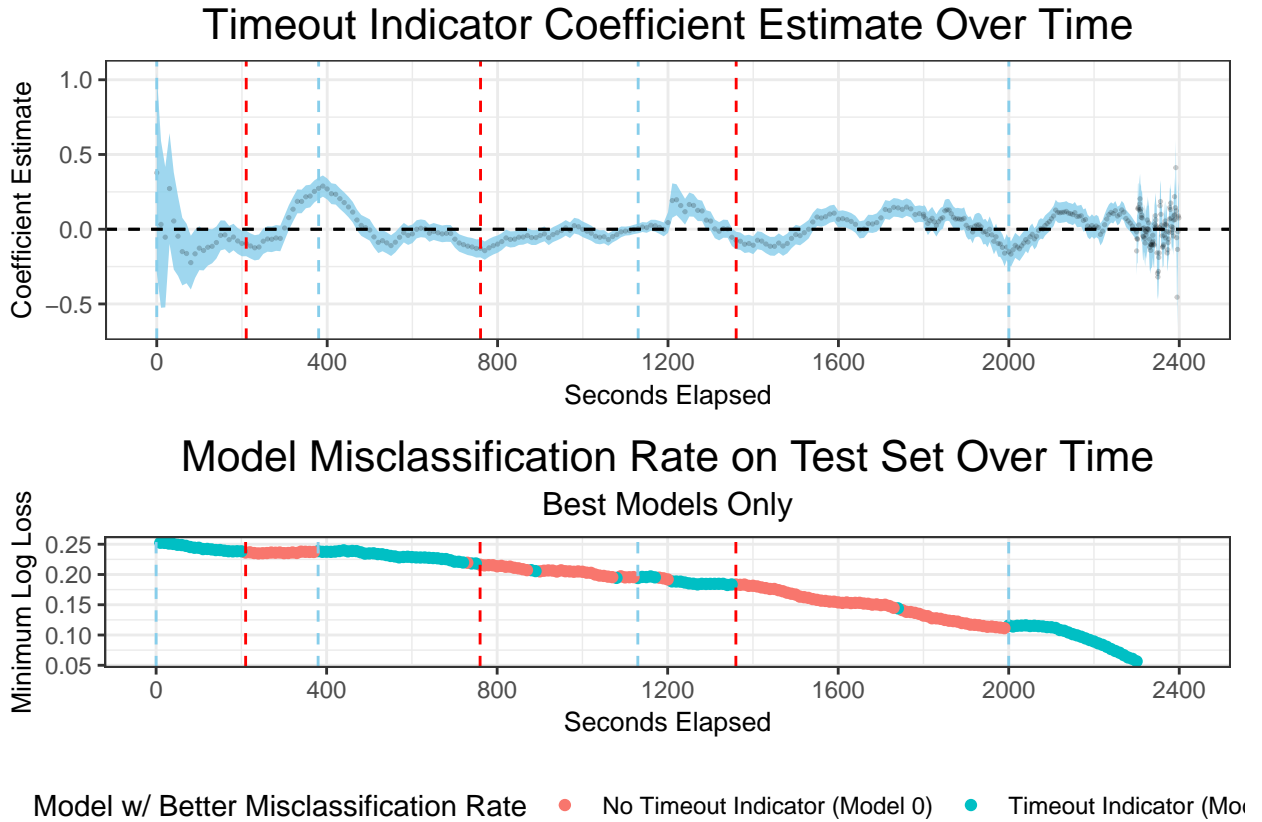


Figure 3.5: Localized Improvement in Win Probability Prediction

timeout on win probability predictions, it's clear that if any after timeout effect is to exist, it is one that exists in the immediate aftermath of taking a timeout and not necessarily for the rest of the game. In the study of after timeout momentum and a quest to quantify the value of taking a timeout in Chapter 4, everything will be studied in a local manner.

Chapter 4

After Timeout Momentum

4.1 How To Measure Momentum

The stage has now been set to study after timeout momentum. As mentioned in the Introduction, a large part of the problem with studying something like momentum is trying to figure out what exactly is meant by the term, and how to go about trying to quantify it. While there is likely no right way or no best way to measure momentum, one way to tackle the problem is to look at scoring runs going into and coming out of timeouts. Specifically, I will look at the net score differential for the timeout calling team in 60, 120, 180, 240 and 300 second intervals before and after the timeout. The purpose of examining intervals of differing length is to see how long any after-timeout effect may last should it exist. Intervals beyond the length of 300 seconds were not examined because it's extremely likely that there would be a formal stoppage of play, either in the form of a media timeout or another called timeout, in the 300 seconds following a timeout.

Formally, let S_{tiB} denote the the number of points scored an i -second interval **before** a timeout is called for the the team calling the timeout. Similarly, let S_{oiB} denote the number of points scored in the same i -second interval for non-timeout calling team. The net score differential for the timeout calling team during the i second pre-timeout interval, N_{tiB} is defined as

$$N_{tiB} = S_{tiB} - S_{oiB}$$

Everything above can be defined analogously for the i second interval **after** the time out. Then, the net score differential for the timeout calling team during the i -second post-timeout interval is given by

$$N_{tiA} = S_{tiA} - S_{oiA}$$

A first pass quantity of interest when measuring momentum is then the difference in the net score differential for the timeout calling team after the timeout versus before the timeout. We will denote this quantity.

$$\Delta_{ti} = N_{tiA} - N_{tiB}$$

In order to illustrate all of these quantities, consider the following game situation between Yale and Harvard.

- Harvard takes a timeout.
- In the 180-second interval prior to taking the timeout, Yale outscores Harvard 12-5.
- In the 180-second interval following the timeout, Harvard outscores Yale 10-7.

In this situation, Harvard is the timeout taking team and Yale is the opponent, as they were not the team which took the timeout. This scenario yields:

- $t = \text{Harvard}$, $o = \text{Yale}$, $i = 180$
- $S_{tiB} = 5$ (the number of points Harvard, the timeout taking team, scored in the 180 seconds **before** the timeout).
- $S_{oiB} = 12$ (the number of points Yale scored in the 180 seconds **before** the timeout).
- $N_{tiB} = -7$ (the net score differential for Harvard, the timeout taking team, during the 180 seconds **before** the timeout).
- $S_{tiA} = 10$ (the number of points Harvard, the timeout taking team, scored in the 180 seconds **after** the timeout).
- $S_{oiA} = 7$ (the number of points Yale scored in the 180 seconds **after** the timeout).
- $N_{tiA} = 3$ (the net score differential for Harvard, the timeout taking team, during the 180 seconds **after** the timeout).
- $\Delta_{ti} = 10$ (the difference in the net score differential for Harvard, the timeout taking team, during the 180 seconds **after** the timeout compared to the 180 seconds **before** the timeout).

4.2 Examining the After Timeout Effect

Not surprisingly, teams call timeouts when local scoring runs are not in their favor. On average, the net score differential in 60 second intervals immediately preceding timeouts was $\bar{N}_{t60} = -1.48$. Similar net score differentials hold for longer length intervals preceding timeouts. Full summary statistics for called timeouts (non-media timeouts) are presented below in Table 4.1. In Figure 4.1, box plots comparing the distributions of pre and post timeout net score differential (N_{tiB} and N_{tiA}).

With the exception if $i = 60$ seconds, which is skewed left, all other intervals have symmetric post-timeout distributions. This is not entirely surprising, as 60 seconds yields time for roughly one possession per team. In most cases, the team calling the timeout will have possession of the ball following the timeout, and will be ensure of at least one possession within the interval, something their opponent may not have the luxury of. For all interval lengths, the average post-timeout net score differential is not only an improvement over the average pre-timeout net score differential, but is actually positive. Remarkably, even \bar{N}_{t300} , the average post-timeout net score differential in the five minute interval following the timeout is positive ($\bar{N}_{t300} = 0.3$). This is perhaps somewhat surprising because in a five minute interval, there is a high

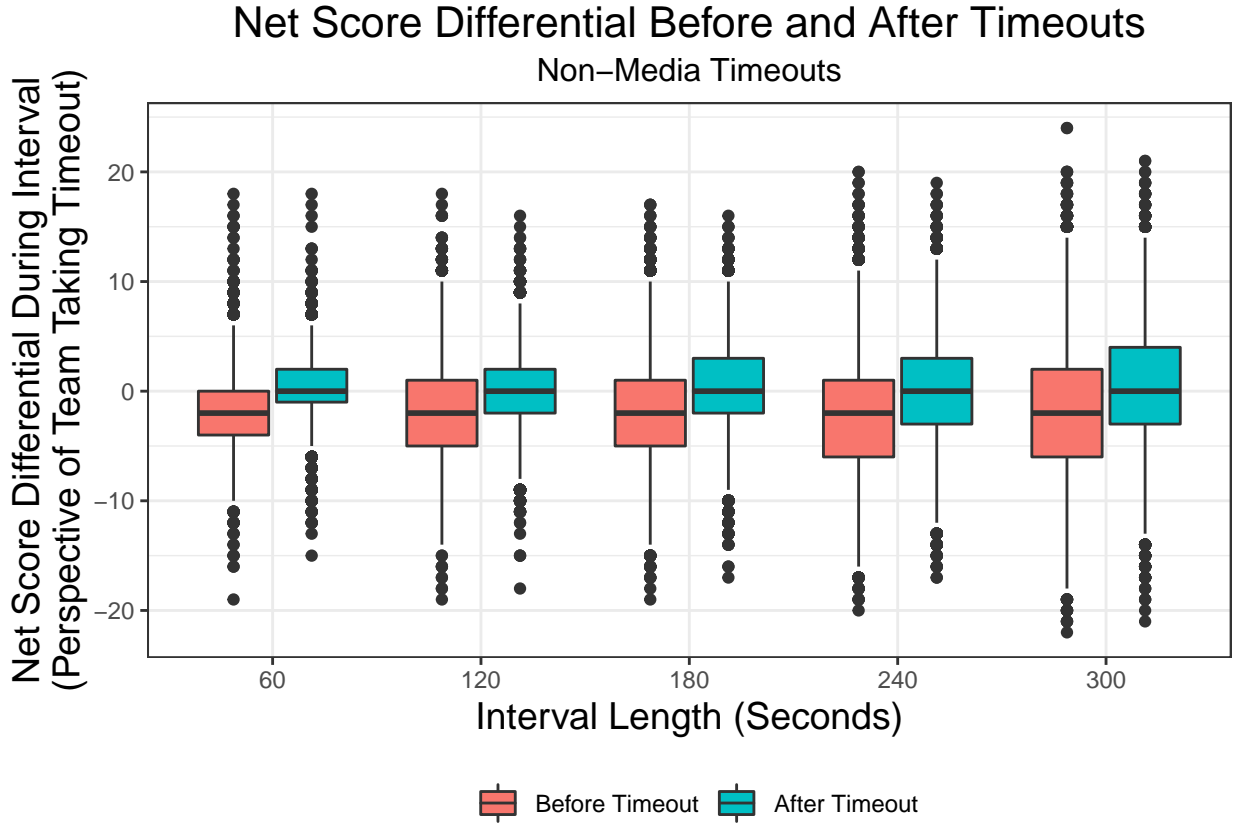


Figure 4.1: Boxplots for Pre and Post (Non Media) Timeout Net Score Differentials

likelihood of experiencing some form of timeout, whether a timeout by the opposing team in response to the original timeout or just a media timeout occurring at its normal 4 minute interval.

One interesting observation readily seen in both Figure 4.1 and Table 4.1 is that for each interval, $\text{Var}(N_{tiA}) < \text{Var}(N_{tiB})$. That is, the variance in the net scoring differentials following a timeout is always less than the variance in the net score differentials in an equal length interval prior to the timeout.

A natural question one might wonder is whether similar results hold after media timeouts, timeouts that are not called by teams. Figure 4.2 and Table 4.2 present box plots and a pre- and post-timeout net score differential mean and variance table for media timeouts. Because media timeouts are not called by either team, the notion of N_{tiA} and N_{tiB} , which were defined in the context of a timeout-calling team, does not make sense. Instead, for media timeouts, everything is treated from the perspective of the home team. That is, in the context of a television timeout, N_{hiA} and N_{hiB} refer to the net score differentials before and after the commercial break from the home team's point of view. For the purposes of comparing N_{hiA} and N_{hiB} before and after a given timeout, this doesn't yield any problems. However, one must be careful when comparing these similar quantities between media timeouts and non-media timeouts. Recall that home teams have on average a 3-4 point home court advantage (Pomeroy,

Table 4.1: Pre and Post Non-Media Timeout Summary Statistics

Interval Length (Seconds)	Pre-Timeout Mean Net Score Differential	Post-Timeout Mean Net Score Differential
60	-1.48 (± 2.91)	0.26 (± 2.34)
120	-1.92 (± 3.86)	0.29 (± 3.21)
180	-2.05 (± 4.51)	0.29 (± 3.93)
240	-2.06 (± 5.06)	0.31 (± 4.59)
300	-2.00 (± 5.55)	0.30 (± 5.17)

Table 4.2: Pre and Post Media Timeout Summary Statistics

Interval Length (Seconds)	Pre-Timeout Mean Net Score Differential	Post-Timeout Mean Net Score Differential
60	0.15 (± 2.13)	0.15 (± 2.26)
120	0.31 (± 3.10)	0.34 (± 3.21)
180	0.51 (± 3.82)	0.51 (± 3.95)
240	0.69 (± 4.42)	0.71 (± 4.53)
300	0.89 (± 4.96)	0.88 (± 5.08)

2018). Prorating such an advantage on a per 60 second basis, to outscore an equal strength opponent by 0.08-0.10 points per minute. Thus, it is not entirely surprising to see for example, that $\bar{N}_{h60B} = 0.15$. What is more interesting is that $\bar{N}_{h60A} = 0.15$ as well. Hence, the pre- and post- media timeout net score differentials in one minute intervals following TV timeouts are identical. Similar results hold for all five of the interval lengths examined, with the full table of \bar{N}_{hiA} , \bar{N}_{hiB} , $\text{Var}(N_{hiA})$, and $\text{Var}(N_{hiB})$ given in Table 4.2.

To this point, only distributions of the N_{tiB} and N_{tiA} have been compared, along with their counterparts for non-media timeouts, N_{hiB} and N_{hiA} . In order to begin to quantify the value of a timeout, as least in its ability to offer a swing in short term net score differential, we return to the idea of Δ_{ti} , the “difference of differences” (i.e. the difference in the short term net score differentials before and after a timeout). In the same way that N_{tiB} and N_{tiA} had counterparts N_{hiB} and N_{hiA} for media timeouts, Δ_{ti} has counterpart Δ_{hi} for media timeouts. In fact, one can extend the notion of Δ_{ti} so that the pre- and post- timeout intervals need not be of equal length. That is, let

$$\Delta_{tij} = N_{tjA} - N_{tiB}$$

denote the difference of differences for pre-timeout interval i and post-timeout interval j . Of course, this quantity has equivalent counterpart for non-media timeouts, Δ_{hij} . Histograms of Δ_{tij} and Δ_{hij} are presented in Figure 4.3 and Figure 4.4 respectively. Faceted sub plots reference the before and after timeout intervals, with negative intervals corresponding to pre-timeout intervals and positive intervals corresponding to post-timeout intervals. Table 4.3 provides a summary table of $\bar{\Delta}_{tij}$ and $\bar{\Delta}_{hij}$.

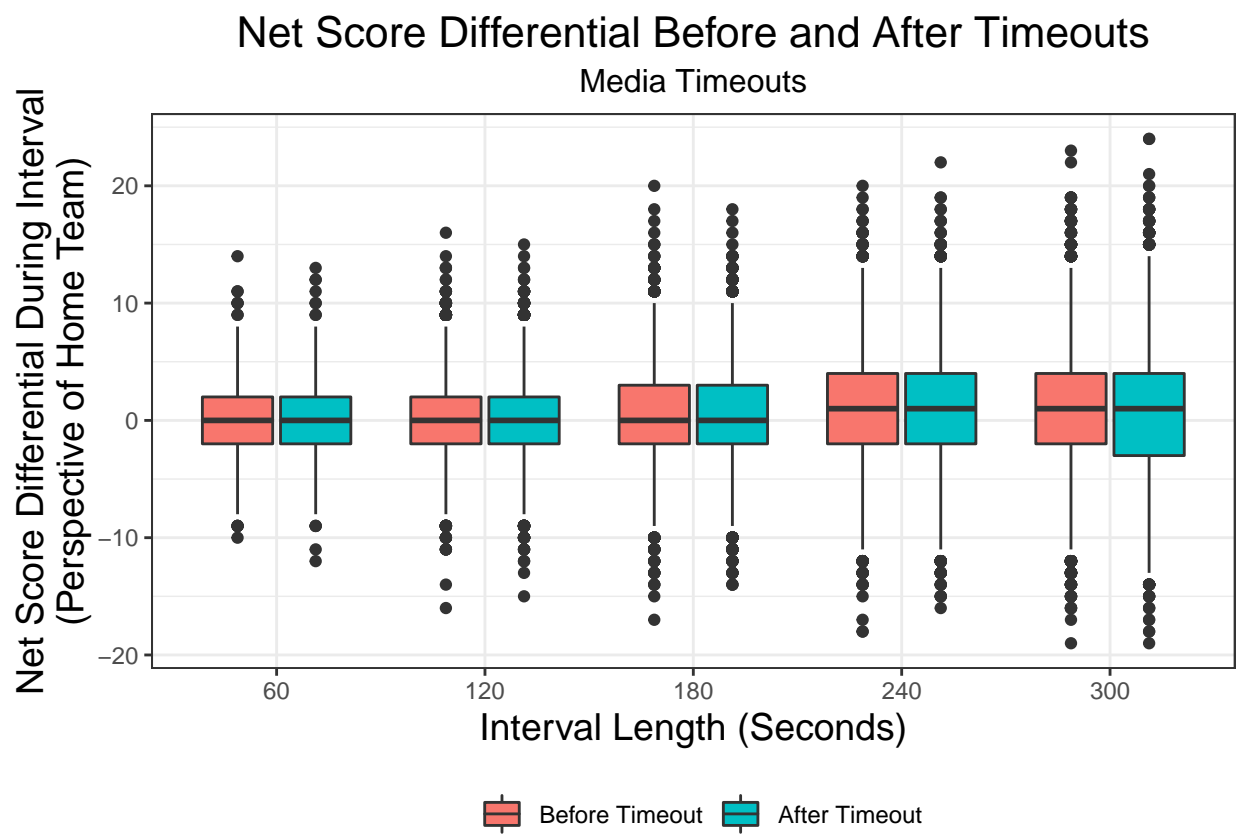


Figure 4.2: Boxplots for Pre and Post (Media) Timeout Net Score Differentials

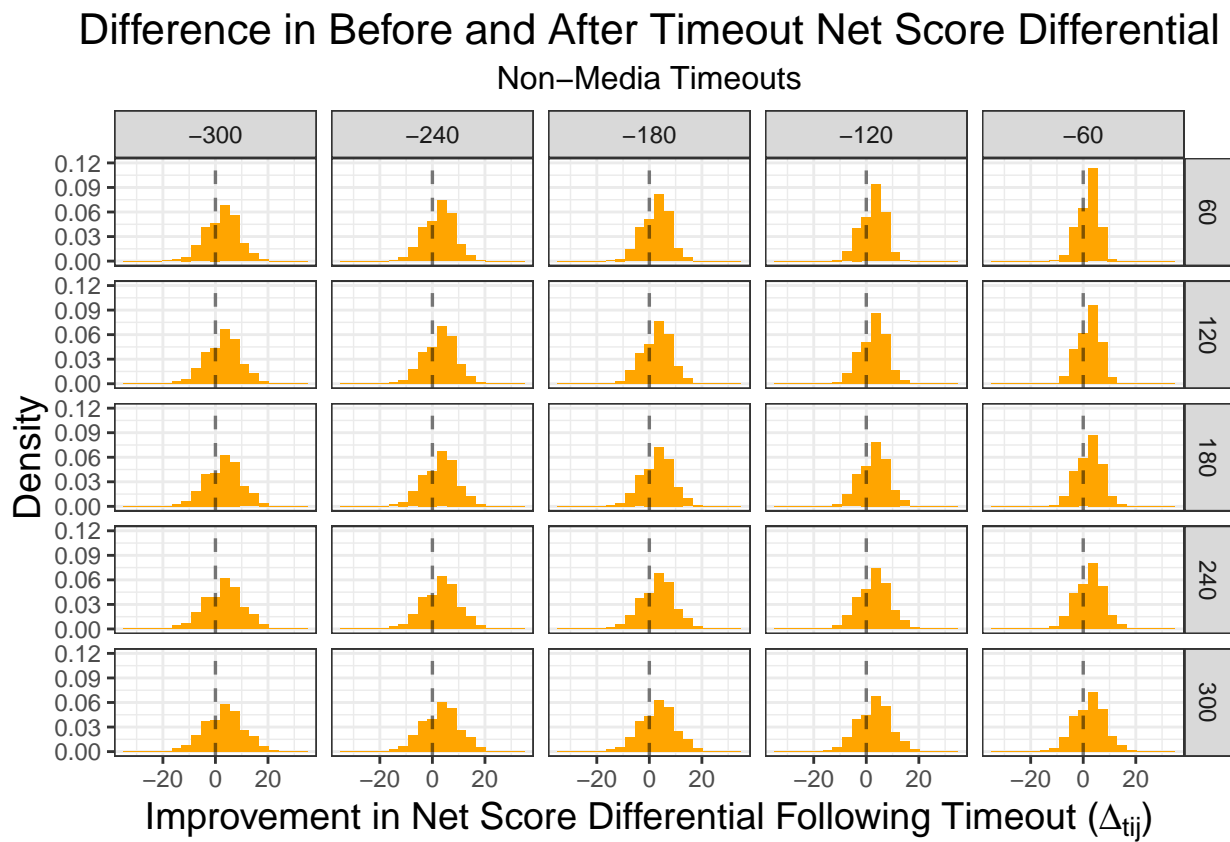


Figure 4.3: Histograms of Improvement in Net Score Differential Following Non-Media Timeouts

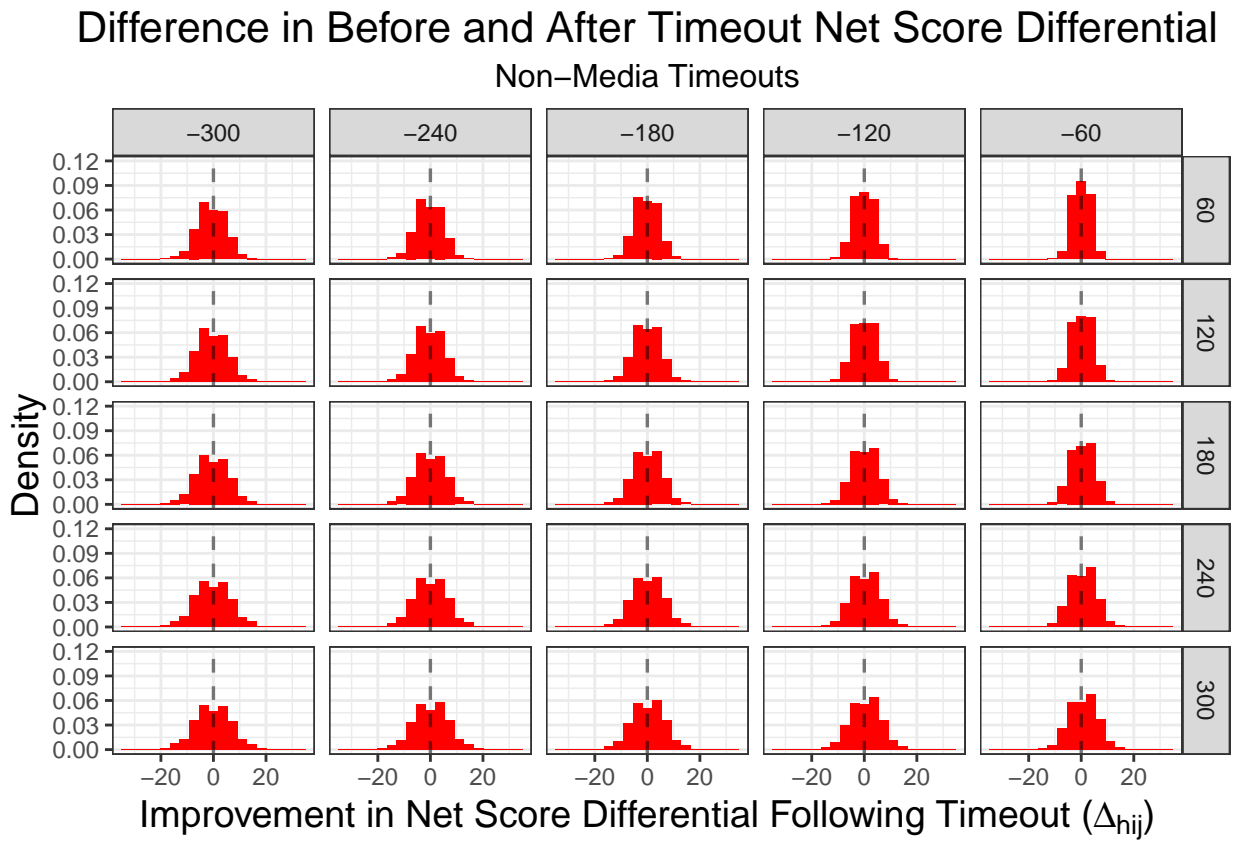


Figure 4.4: Histograms of Improvement in Net Score Differential Following Non-Media Timeouts

Improvement in Net Score Differential in Interval Following Timeout Over Course of Game

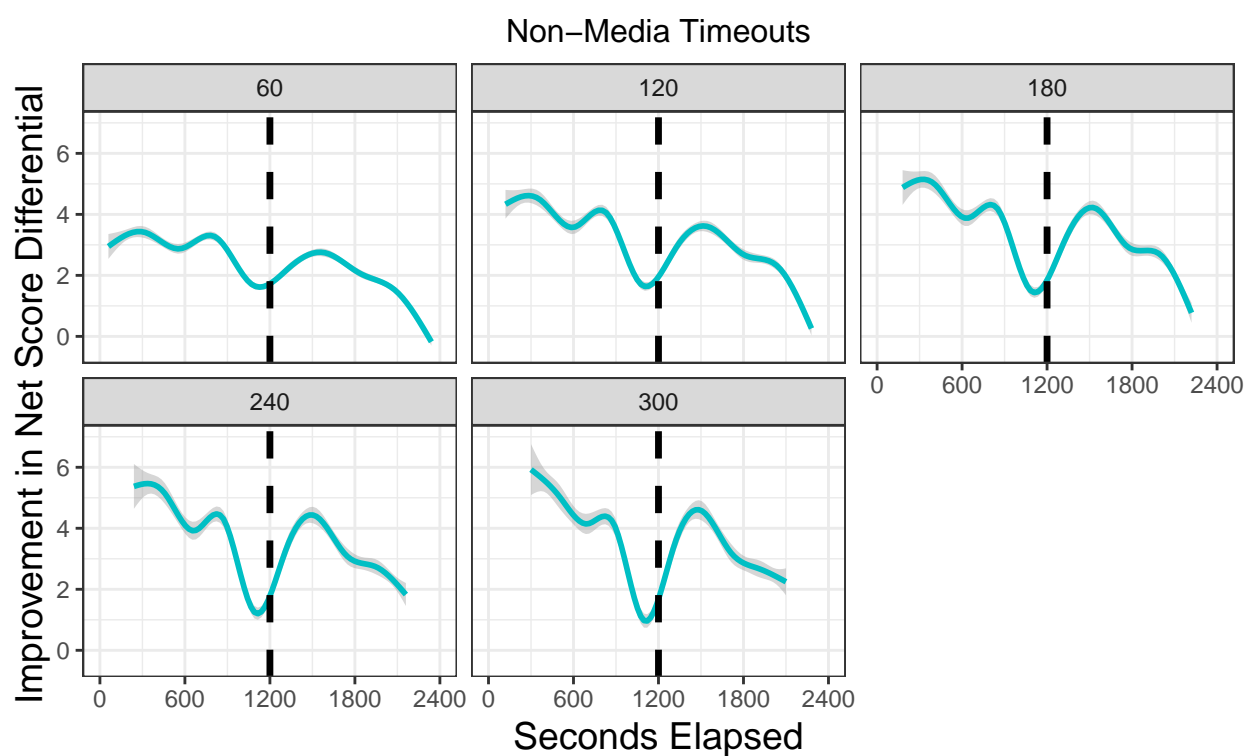


Figure 4.5: Post-Timeout Improvement Following Non-Media Timeouts Over the Course of the Game

Improvement in Net Score Differential in Interval Following Timeout Over Course of Game

Media Timeouts

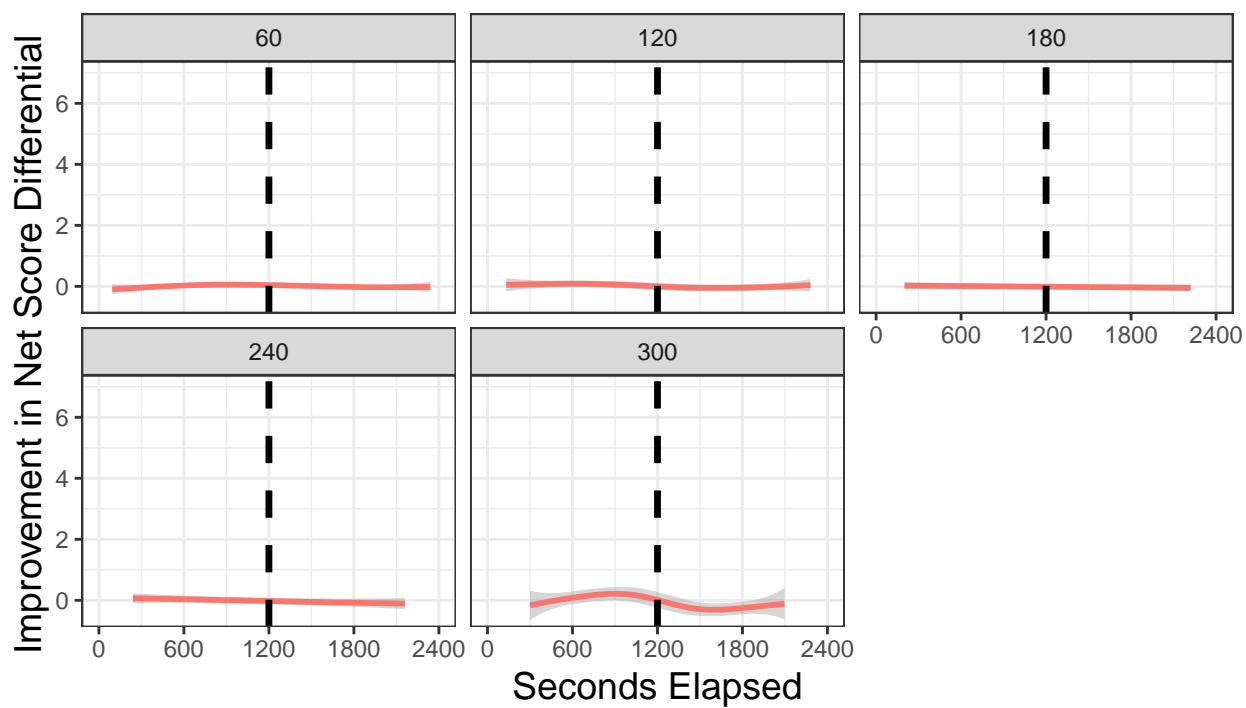


Figure 4.6: Post-Timeout Improvement Following Media Timeouts Over the Course of the Game

Figure 4.5 and Figure 4.6 show the result of fitting Generalized Additive Models (GAM) to get a smoothed plot on Δ_{ti} and Δ_{hi} respectively as a function of time. The individual values of Δ_{ti} and Δ_{hi} are not shown due to the fact that there are too many points to glean any value from adding them to the plots. These plots can be interpreted as smooth estimates of the improvement a timeout yields in short term net score differential in the ensuing post-timeout interval. To be clear, this is not yet a derivation of the value of a timeout. After all, we have seen that timeouts are called on adverse scoring runs, and even without taking a timeout, one might expect some form of simple regression to the mean. To combat the bias of when timeouts are most often called, and to adjust for the presence potentially confounding variables in quantifying the value of a timeout, Section 4.3 creates a framework entitled Points Above Expectation.

4.3 Points Above Expectation After Timeouts

A natural question to ponder when examining after timeout momentum is “what is gained by taking a timeout”. That is, how much better does a team do *because* of taking a timeout instead of letting the flow of play continue. To answer this question, and to account for the fact that coaches are more likely to call timeouts when their team is on a negative scoring run, this section outlines a framework for modeling the number of points a team would expect to score in a certain time interval given a particular game situation, were the run of play to continue as normal. Taking the difference between the number of points a team scored following a timeout and the number of points they were expected to score given the particular game situation allows one to quantify the value of a timeout in terms of Points Above Expectation (PAE).

More formally, let S_{ijkB} and S_{ijkA} denote the net score differential for a team in i -second intervals before and after game state j (time remaining, score differential, and pre-game point spread) of game k . This is a simple extension of S_{tiB} and S_{tiA} from the previous section to allow for all game states, not just timeout. The following mixed-effects model is considered.

$$\begin{aligned} S_{ijkA} &= \mathbf{X}_{ijk}^T \boldsymbol{\beta} + \gamma_k + \epsilon_{ijk} \\ \gamma_k &\sim N(0, \tau^2) \\ \epsilon_{ijk} &\sim N(0, \sigma^2) \end{aligned}$$

\mathbf{X}_{ijk} is a vector of covariates encoding information about the particular game state. Specifically, the following covariates are considered:

- S_{ijkB} : The team’s net score differential in the i second interval preceding game state j of game k .
- **favored_by**: The pre-game point spread from the team’s perspective.
- **score_diff**: The current score differential from the team’s perspective.
- **secs_remaining**: The amount of time remaining in the game.

γ_k is a game-level random effect for game k . It is included in the model to deal with the fact that observations can come from the same game and non truly independent, thus violating the traditional least squares regression framework. ϵ_{ijk} represents traditional random error. A version of the mixed-effects model was fit for each of the five time intervals, $i \in \{60, 120, 180, 240, 300\}$. The training set for these models removed all game states where a timeout took place (both media and non-media) in order to adequately capture expected behavior without stoppage of play. Predicting these models on game states that are timeouts, both media timeouts and called timeouts, then allows for an estimate of how much better or worse a team performed as a result of the stoppage in play. Specifically points above expectation is given by the residual

$$\hat{S}_{ijkA} - S_{ijkA}$$

It's worth noting that unlike the win probability models in Chapter 3, the vector of coefficients, β does not vary greatly with time. Perhaps these coefficients would change slightly in end of game scenarios, when stoppages in play are more frequent. Time remaining is taken into account in the model, and any possible (but small) non-linear dependence on time for the remaining coefficients is accepted as a possible limitation of the points above expectation framework. Another possible limitation of the PAE model framework is the fact that timeouts can happen after game state j but before the end of the i -second interval. That is, the model doesn't adequately account for the fact that a particular game state may not be a timeout but may imminently followed by one.

	60 Sec		120 Sec		180 Sec		240 Sec		300 Sec	
<i>Predictors</i>	<i>Estimates</i>	<i>p</i>	<i>Estimates</i>	<i>p</i>	<i>Estimates</i>	<i>p</i>	<i>Estimates</i>	<i>p</i>	<i>Estimates</i>	<i>p</i>
(Intercept)	0.171	<0.001	0.376	<0.001	0.604	<0.001	0.843	<0.001	1.092	<0.001
avored_by	0.065	<0.001	0.136	<0.001	0.209	<0.001	0.285	<0.001	0.363	<0.001
score_diff	-0.072	<0.001	-0.154	<0.001	-0.240	<0.001	-0.331	<0.001	-0.426	<0.001
secs_remaining	-0.000	<0.001	-0.000	<0.001	-0.001	<0.001	-0.001	<0.001	-0.001	<0.001
score_run_pre	0.024	<0.001	0.027	<0.001	0.034	<0.001	0.038	<0.001	0.039	<0.001
Random Effects										
σ^2	5.10		9.06		12.35		14.95		16.83	
τ^2	0.52	game_id	2.30	game_id	5.53	game_id	10.43	game_id	17.14	game_id
Observations	1610978		1523730		1438629		1354127		1269404	
Marginal R ² / Conditional R ²	0.079 / 0.165		0.159 / 0.329		0.224 / 0.464		0.276 / 0.573		0.318 / 0.662	

Figure 4.7: Mixed Effects Model Summaries

Model summaries for the mixed-effects models are shown in Figure 4.7. Note that for each model, the `secs_remaining` coefficient estimate is both negative and statistically significant, but in some cases is smaller in magnitude than -0.001, likely because the units of time are in seconds. Interestingly, the `score_diff` coefficient also has negative estimate (and significantly so) for each of the five intervals. Most likely, this is simply due to the presence of other variables capturing similar information,

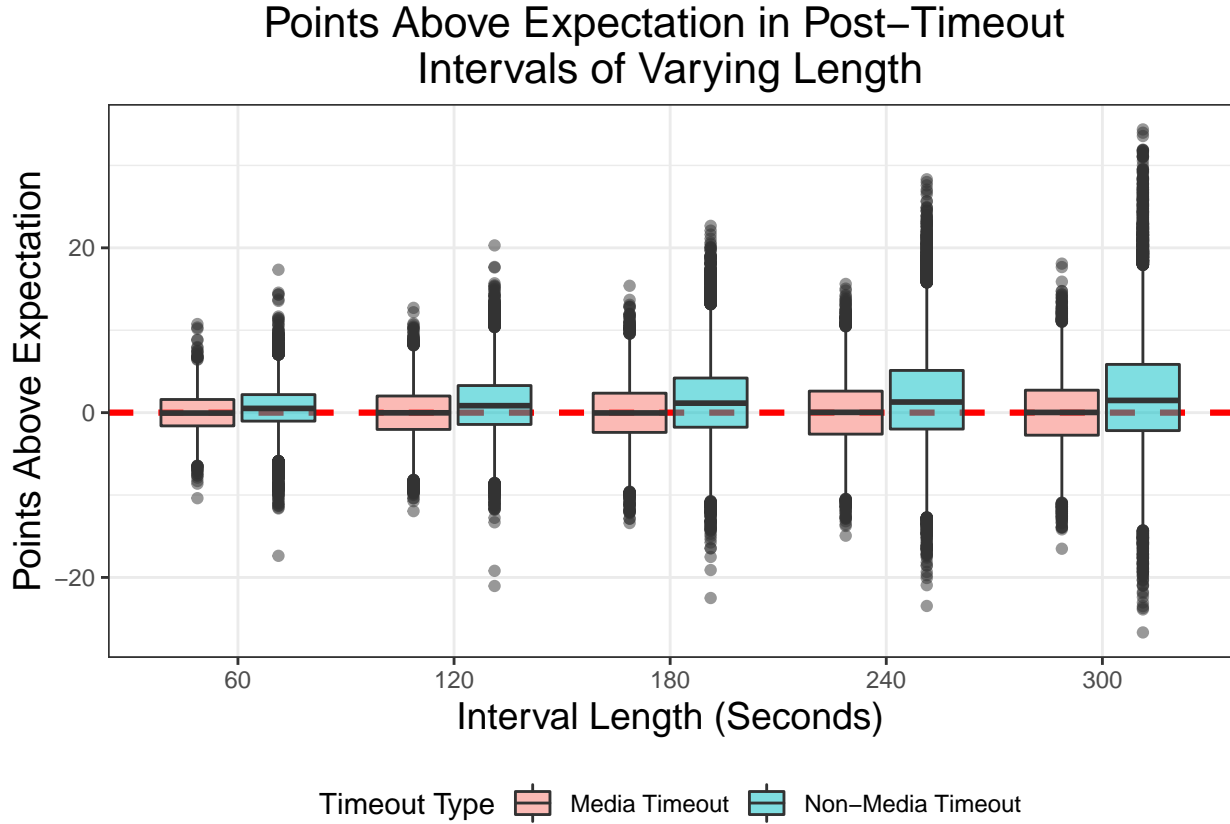


Figure 4.8: Boxplots for Points Above Expectation after Timeouts

such as `score_run_pre` (S_{ijkB}) and `avored_by`. Not surprisingly, we see that S_{ijkB} is positively associated with S_{ijkA} .

Predicting each of the models on timeout game states allows for the computation of points score relative to expectation following timeout. Distributions of PAE after media and non-media timeouts are shown in Figure 4.8. For each of the five intervals, non-media timeouts have median PAE > 0 . That is, on average, teams calling a timeout perform better in up to five minute intervals following the the timeout than would be expected were no timeout to be called from an equivalent game state. Notice, however, that this does not hold for media timeouts, suggesting that even when accounting for the fact that teams are more likely to take timeouts when on negative scoring runs, there appears to be more benefit in timeouts taken compared to media timeouts. As Table 4.4 shows, the average benefit of a timeout is between 0.5-0.6 points per minute in the first minute after the timeout. It's value on a per minute basis decays in the following minutes, but it is still worth an average of between 1.5-2.2 points above expectation five minutes after the timeout was taken.

Given that media timeouts occur at fixed times, it's natural to wonder how teams perform out of media timeout following negative scoring runs. Figure 4.9 shows box plots of points above expectation following timeouts where the net score differential entering the timeout is < 0 . It's clear that even following negative scoring runs, called timeouts yield more benefit than media timeouts.

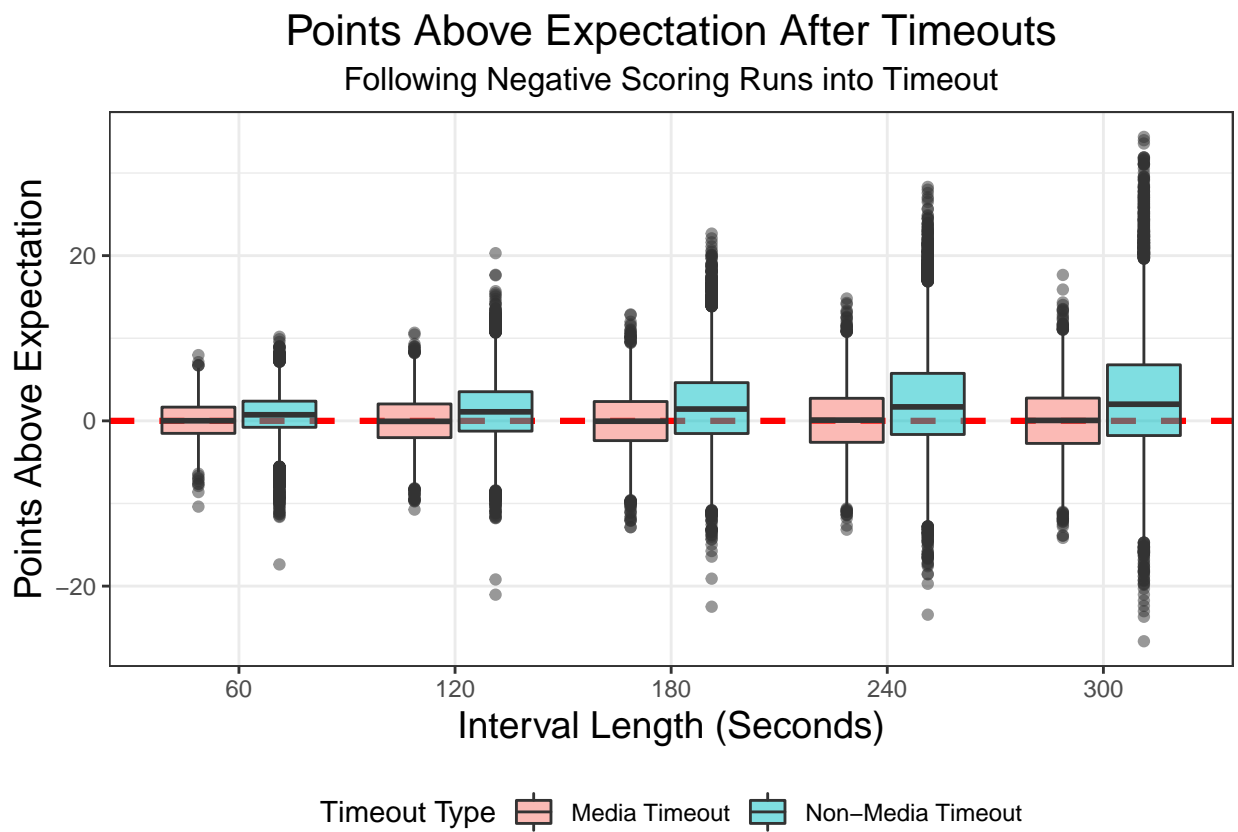


Figure 4.9: Boxplots for Points Above Expectation after Timeouts Following Negative Scoring Runs

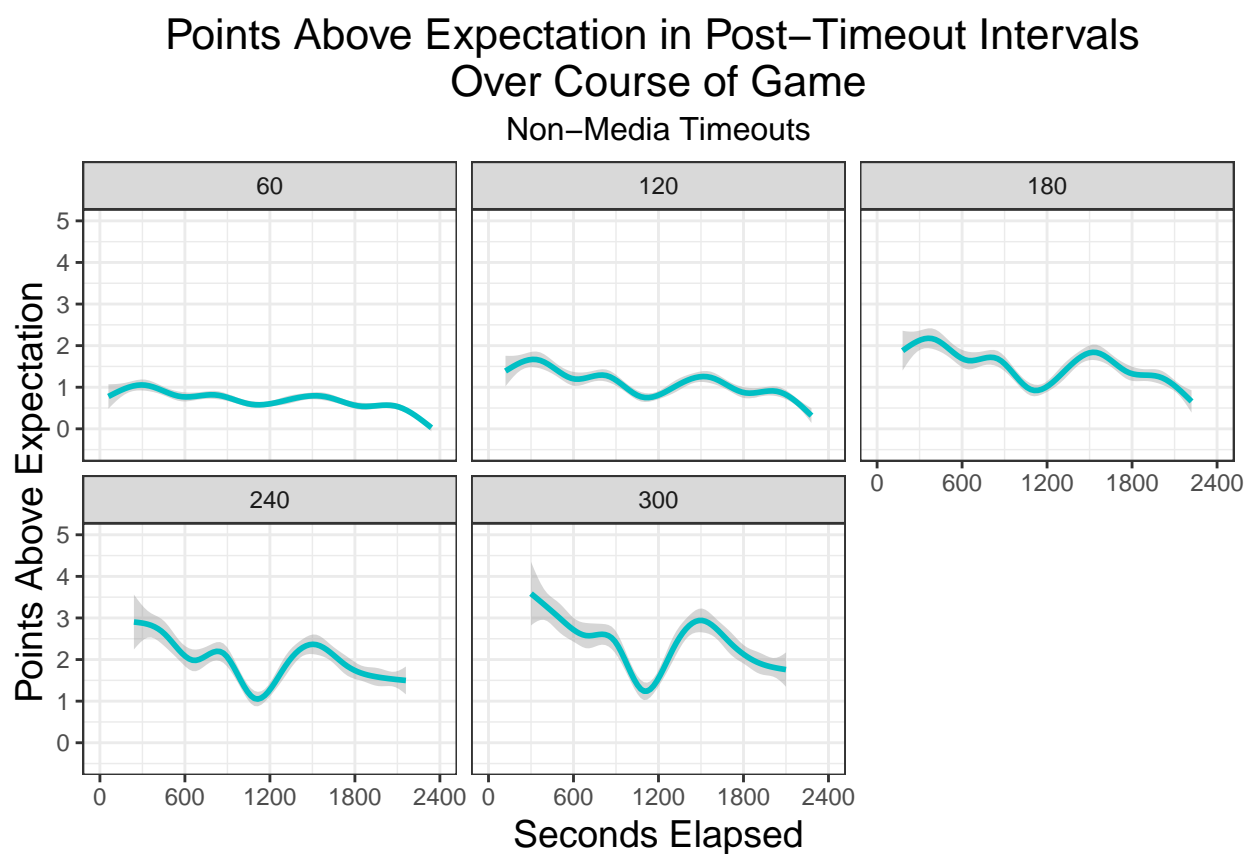


Figure 4.10: Points Above Expectation Throughout the Course of a Game (Non-Media Timeouts)

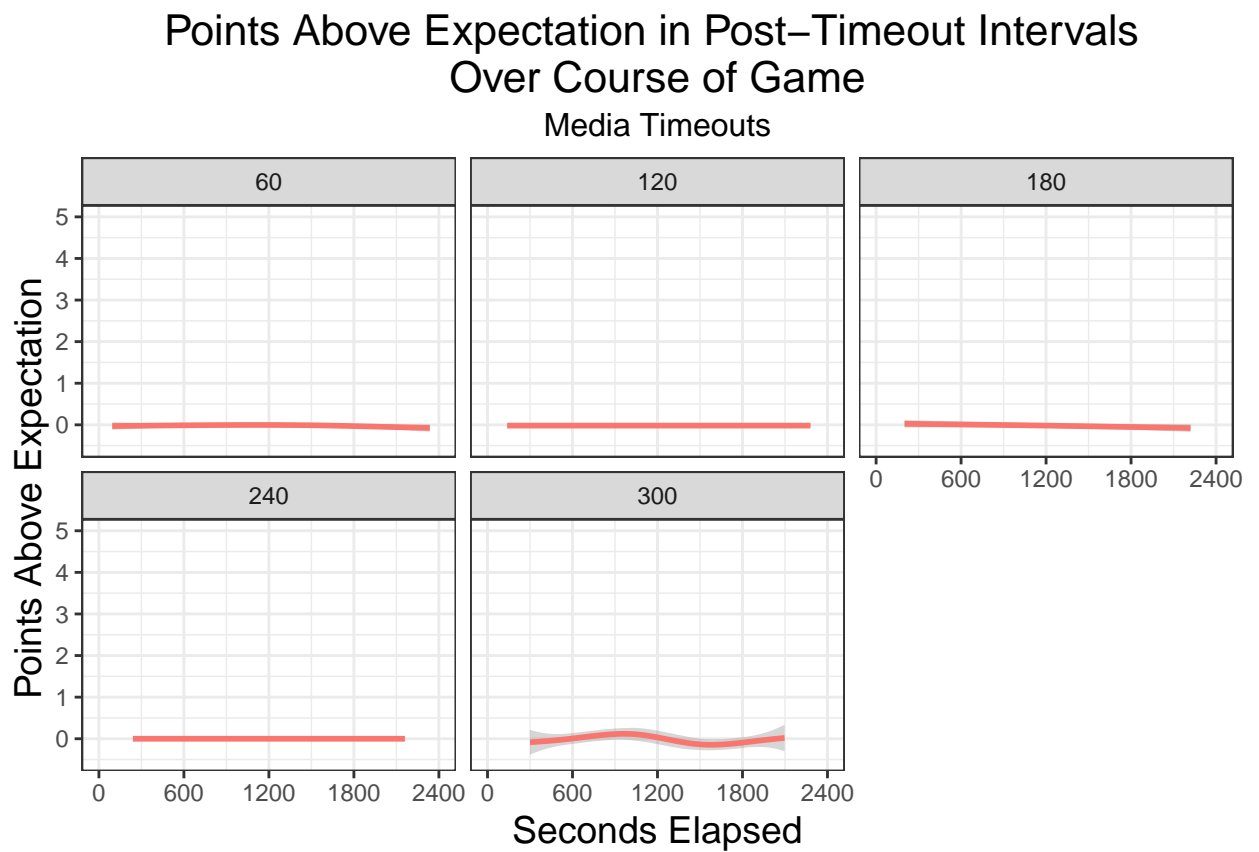


Figure 4.11: Points Above Expectation Throughout the Course of a Game (Media Timeouts)

Figure 4.10 and Figure 4.11 show the result of applying GAM smoothing on PAE after non-media and media timeouts, respectively, over the course of the game. It seems that teams tend to out perform expectation after timeouts by larger margins earlier in the game than later in the game. This, and the fact the the results of Table 4.4 more closely align with late game timeout PAE, are likely due to the fact that more timeouts are taken later in the game than in the beginning of the game, and that there are more stoppages of continuous game play towards the end of games. These results do not mean to suggest that timeouts at the beginning of games necessarily more valuable than late game timeouts, but rather that on average, teams tend to receive a larger points above expectation benefit from early timeouts. Each additional point difference in `score_diff` becomes increasingly more valuable over the course of the game, as shown in Figure 3.1, and this fact must be taken into account when trying to account for the value of a timeout.

Figure 4.12 estimates the value of taking a timeout by multiplying the smoothed 300-second points above expectation curves by the smoothed score differential (`score_diff`) coefficient function from Figure 3.1. The resulting values reflect a team's increase in the odds of winning a game by taking a timeout and experiencing the average 300-second boost in points above expectation. In formal terms, let $\hat{\beta}_t$ denote the smoothed score differential (`score_diff`) coefficient estimate at time t , and let ξ_t denote the smoothed average 300-second PAE at time t . Figure 4.12 simply presents a plot of the function

$$e^{\hat{\beta}_t \times \xi_t}$$

This is a decent proxy for timeout value because it is the points above expectation in the five minute interval following a timeout multiplied by the value of increasing `score_diff` by 1 point in the team's favor at that particular time of game. Because we are looking at the PAE curve for the 300 second interval before/after a timeout, we are unable to obtain estimates of relative value in the first or last five minutes of a game. 300 seconds was chosen because it provided the most stable estimates towards the end of the game, where smaller length intervals presented extreme noise. It's likely that the value of a timeout would only increase towards the end of the game. Nevertheless, this method illuminates that timeouts offer diminishing returns as the first half of the game wears on, with timeouts taken right before halftime, often "use it or lose it" timeouts, being the least valuable. This suggests that if a coach has the desire to use a timeout in the first half, they should take a greedy approach and call a timeout sooner rather than later. Perhaps not surprisingly, second half timeouts are worth more than first half timeouts, with timeouts becoming seemingly more valuable throughout the course of the second half.



Figure 4.12: Value of Taking a Timeout

Table 4.3: Summary Table of Post Timeout Improvement

Pre- Timeout Interval (Seconds)	Post- Timeout Interval (Seconds)	$\bar{\Delta}_{tij}$ (Non-Media Timeouts)	$\bar{\Delta}_{tij}$ (Media Timeouts)
60	60	1.92 (± 3.67)	0.00 (± 3.08)
60	120	2.13 (± 4.25)	0.19 (± 3.83)
60	180	2.23 (± 4.77)	0.35 (± 4.44)
60	240	2.31 (± 5.27)	0.54 (± 4.97)
60	300	2.36 (± 5.76)	0.72 (± 5.47)
120	60	2.42 (± 4.48)	-0.16 (± 3.80)
120	120	2.67 (± 4.95)	0.02 (± 4.40)
120	180	2.80 (± 5.39)	0.19 (± 4.94)
120	240	2.89 (± 5.82)	0.37 (± 5.41)
120	300	2.95 (± 6.26)	0.55 (± 5.87)
180	60	2.56 (± 5.07)	-0.36 (± 4.38)
180	120	2.83 (± 5.48)	-0.18 (± 4.91)
180	180	2.97 (± 5.87)	-0.01 (± 5.40)
180	240	3.07 (± 6.27)	0.17 (± 5.83)
180	300	3.13 (± 6.68)	0.34 (± 6.24)
240	60	2.57 (± 5.58)	-0.53 (± 4.90)
240	120	2.85 (± 5.96)	-0.35 (± 5.37)
240	180	3.01 (± 6.32)	-0.18 (± 5.82)
240	240	3.12 (± 6.67)	-0.01 (± 6.21)
240	300	3.18 (± 7.06)	0.16 (± 6.60)
300	60	2.52 (± 6.05)	-0.74 (± 5.39)
300	120	2.80 (± 6.40)	-0.57 (± 5.82)
300	180	2.96 (± 6.74)	-0.40 (± 6.24)
300	240	3.07 (± 7.06)	-0.23 (± 6.61)
300	300	3.14 (± 7.42)	-0.07 (± 6.96)

Table 4.4: Summary Table of Points Above Expectation

Post-Timeout Interval Length (Seconds)	Mean PAE (Non-Media Timeouts)	Median PAE (Non-Media Timeouts)	Mean PAE (Media Timeouts)	Median PAE (Media Timeouts)
60	0.59 (± 2.47)	0.515	-0.02 (± 2.17)	-0.051
120	0.98 (± 3.60)	0.841	-0.02 (± 2.97)	-0.022
180	1.37 (± 4.67)	1.154	-0.02 (± 3.53)	-0.038
240	1.79 (± 5.71)	1.288	0.00 (± 3.87)	0.035
300	2.18 (± 6.78)	1.484	-0.03 (± 4.09)	0.025

Conclusion

5.1 Discussion of Results

There are many results in this thesis worth briefly summarizing and commenting on. To begin with, it's quite clear that on average, taking a timeout improves net score differential in intervals immediately proceeding the timeout when compared to equal length intervals heading into the timeout. Taking a timeout also acts as a form of variance reduction, decreasing the variance in the distribution of net score differential after the timeout compared to before the timeout. Thus, a timeout really is a "stabilizer", offering teams a sort of way to reset the game, at least from a local standpoint. Interestingly, these results don't hold for any stoppage of play, but rather only timeouts called by one of the two teams. That is, there seems to be some kind of fundamental difference, psychological or otherwise, between called timeouts vs. media timeouts. This also suggests that coaches are inherently rational in how they utilize their timeouts. They call timeouts that benefit their own team more than their opponent, thus providing their own team with an advantage. Perhaps a large part of the reason that media timeouts have seemingly little impact on local scoring runs, at least in an average case analysis as is considered in this thesis, is because of they are relatively fixed in time (i.e. media timeouts occur at times that doesn't favor either team on average).

Even when accounting for the fact the teams tend to take timeouts when on adverse local scoring runs by building a points above expectation framework, we still find big differences in media timeouts and non-media timeouts. If asked to report a single figure for the value of a timeout, this paper would concluded that timeouts are worth between 1.5-2.2 points over a five minute interval following the timeout. A better quantification of the the value of a timeout looks at the number of points above expectation a timeout contributes on average over varying intervals throughout the course of a game, as is done in Figure 4.10. When taking into account in the fact that a one point change in score differential increases in value the less time there is remaining in a game, we arrive at an estimate of the value of a timeout over the course of the entire game. Timeouts diminish in value over the course of the first half, with "use it or lose it" timeouts immediately before halftime representing the least valuable timeouts. This is likely because coaches take these timeouts not because they sense that a timeout will provide their team with more of an advantage than it will provide their opponent, but rather because the timeout will go to waste should it not be utilized. These timeouts still generate positive value, but are only worth about

33-50% of the value of timeouts taken earlier in the half. In the second half of games, timeouts are more valuable than in the first half, with their value increasing rapidly at the end of games. With respect to coaching strategy, these results suggest that coaches should be willing to use timeouts sooner in the first half, especially given the fact that the recoup one timeout following halftime. While timeouts right before halftime are the least valuable, they still offer positive points above expectation compared to not taking a timeout. Thus, coaches should never enter halftime having not taken a timeout in the first half.

The results of this thesis contribute a lot to the limited literature on timeouts' ability to stop opponent momentum. These results confirm the findings of (Mace et al., 1992) and (Roane et al., 2004) that timeouts in college basketball do indeed seem to stop opponent momentum. In estimating the value of a timeout, this work suggests a much larger value than the estimate 0.2 reported by (Permutt, 2011) for the NBA. (Permutt, 2011) scope was significantly more limited than this work, restricting itself to examining intervals following timeouts after 6-0 runs in the first half. NBA teams have many more timeouts per game, which might contribute to the lower number reported by (Permutt, 2011). This thesis is the first known work to date to comment on how the value of timeouts evolves throughout the course of the game. While the results of previous papers should not be discounted, this work offers a sample size significantly larger than any work to date, and thus, possess the unique ability to answer several questions with more clarity and detail than other previous works.

5.2 Limitations

While this work does a lot to advance the existing literature on the topic of timeouts and momentum in college basketball, it is not without limitations. To begin with, when assessing the value of timeouts, this work can not adequately comment on the worth of timeouts in end of game situations. Moreover, it only attempts to assess the value of timeouts in terms of points above expectation, although coaches might take timeouts when maximizing points above expectation is not the primary objective. Taking a timeout allows coaches to substitute players, and a coach may decide to utilize a timeout in order to get an optimal offensive, or particularly, defensive lineup near the end of the game. Teams may take a timeout to avoid getting trapped into a turnover or a five-second violation should they be unable to inbound the ball. Coaches often use timeouts in end of game situations to ensure their players are on the same page regarding fouling the opponent to stop the clock. While trying to maximizing net score differential usually corresponds with maximizing win probability, this is not always the case. Fouling when up by three points to prevent an opponent from attempting a game-tying three pointer, for example, is a decision trying to maximize a team's chances of winning rather than maximizing localized net score differential. As such, measuring the value of timeouts on the basis of points above expectation fails to account for other ways in which timeouts can provide value. As mentioned in Chapter 4, another limitation of this analysis is it's failure to consider downstream stoppages of play. There is almost certainly differences in the value of timeouts right

before a media timeout or opponent timeout when compared to a longer stretch of uninterrupted play. Finally, `ncaahoopR` does not currently keep track of which team is in possession of the ball. While possession of the ball is not too valuable for majority of the game, it is extremely valuable in end of game situations. As such, any future exploration of end of game timeout value must incorporate whether or not the team taking the timeout has possession of the ball.

Looking at differences in net score differential before and after timeouts is by no means the only way to measure momentum. An initial approach to question sought to treat it as a change point detection problem, examining where change points occurred in game score differentials and seeing if there was any association between when change points occurred and when timeouts were taken. This idea hit a snag when the assumption of independent observations was not met. Nevertheless, this idea seems like a natural way to phrase the problem and could lead to better ways to measure momentum, perhaps with something like a Hidden Markov Model, or treating the game in a more stochastic manner.

5.3 Future Work

Future work hopes to examine timeout usage in the context of team/coach evaluation. That is, in theory, this line of work could be extended to examine which teams perform best out of timeouts and which coaches call timeouts at the most theoretically optimal times of game. Improvements for quantifying timeout value in end of game situations is another logical source of future exploration. Lastly, a natural extension of this work will account for timeout sequencing, sequences of consecutive timeouts called in close proximity to one another, in order to explore how such sequences not only change the value of timeouts, but also whether there is a strategic advantage in quickly countering an opponent's called timeout.

Appendix A

Code

All code and data is available in GitHub repository at <https://github.com/lbenz730/Senior-Thesis>. The `ncaahoopR` package can be found on GitHub at <https://github.com/lbenz730/ncaahoopR>. Additionally, all code used in this project is included in the appendices. Code used for scraping and cleaning data, as well as building and evaluating various models has been included in the first appendix. Code for generating all figures and tables in the report is included in the second appendix.

Scraping Data

```
### Scrape.R
### Scrapes Data for 2016-17 and 2017-18 Seasons
### Also Scrapes 2018-19 season data as test set (Run on 2/3/2019)

library(ncaahoopR)

### Load Games IDs to Scrape
load("espn_game_ids.rda")
espn_game_ids <-
  espn_game_ids[!espn_game_ids %in% gsub("[^0-9]", "",
                                          dir("pbp_data/"))]
n <- length(espn_game_ids)

for(i in 1:n) {
  cat("Game", i, "of", n, "\n")
  fails <- 0
  while(fails >= 0 & fails <= 5) {
    x <- try(get_pbp_game(espn_game_ids[i]))
    if(class(x) != "try-error") {
      write.csv(x, paste0("pbp_data/game_id_",
                          espn_game_ids[i], ".csv"),
```

```

        row.names = F)
    fails <- -1
  }else{
    fails <- fails + 1
  }
}
}

### Scrape 2018-19 Data
game_ids <- c()
i <- 0
for(team in ids$team) {
  i <- i + 1
  cat("Scraping Team", i, "of", "353\n")
  game_ids <- c(
    game_ids,
    get_schedule(team) %>%
      filter(date < Sys.Date()) %>%
      pull(game_id))
}

game_ids <- unique(game_ids)
n <- length(game_ids)
for(i in 1:n) {
  cat("Scraping Game", i, "of", n, "\n")
  x <- get_pbp_game(game_ids[i])
  write.csv(x, paste0("test_pbp_data/game_id_",
                     game_ids[i], ".csv"),
           row.names = F)
}

```

Cleaning Data

```

### Clean.R
### Prepares data for modeling purposes
### Notes:
###   In creating the training set, I get cleaned versions from
###   both team's perspectives. The purpose of this is two-fold.
###   Win probability should be symmetrical--that is,
###   if we switched which team's perspective we are predicting
###   the model from, all that should change is estimates of win
###   probability (p) go to (1-p). Moreover, the default rendering
##   of ncaahoopR and the underlying ESPN play by play game logs

```

```

### is to examine everything from the perspective of the home
### team. While home court advantage is implicit in the
### favored_by column, we need to be careful to ensure that
### we are not conflating important effects with
### simply being the home team.

library(dplyr)
library(readr)

### Names of all ESPN files
file_names <- unlist(sapply(dir("pbp_data",
                                full.names = T), dir,
                                full.names = T))

n <- length(file_names)

for(i in 1:n) {
  if(i %% 100 == 0) {
    cat("Cleaning File #:", i, "of", n, "\n")
  }
  x <- read_csv(file_names[i], col_types = cols())

  ### Check to make sure pbp log is valid
  bad_file <-
    any(abs(x$score_diff[2:nrow(x)] -
           x$score_diff[1:(nrow(x) - 1)]) > 3)

  if(!bad_file) {
    ### Clean data set from home team's perspective
    team <-
      mutate(x, "win" = ifelse(score_diff[nrow(x)] > 0, 1, 0),
             "timeout_ind" = home_timeout_ind - away_timeout_ind,
             "timeout_diff" = home_time_out_remaining -
               away_time_out_remaining,
             "timeout_to_use" = home_time_out_remaining > 0) %>%
      select(game_id, date, home, away, play_id, half,
             secs_remaining, secs_remaining_absolute,
             home_score, away_score, score_diff,
             timeout_ind, timeout_diff, timeout_to_use,
             home_favored_by, win, description) %>%
      rename("team" = home,
             "opponent" = away,
             "team_score" = home_score,
             "opp_score" = away_score,
             "favored_by" = home_favored_by)
  }
}

```

```

    if(!exists("train")) {
      train <- team
    }else{
      train <- bind_rows(train, team)
    }
  }
}

### Remove Games with Bad Timeout Parsing
bad_ids <- c()
for(i in 1:n) {
  if(i %% 100 == 0) {
    cat("Cleaning File #:", i, "of", n, "\n")
  }
  x <- read_csv(file_names[i], col_types = cols())

  if(any(x$home_time_out_remaining < 0 |
        x$away_time_out_remaining < 0 |
        x$home_time_out_remaining > 6 |
        x$away_time_out_remaining > 6)) {
    bad_ids <- c(bad_ids, x$game_id[1])
  }
}

train <- filter(train, !game_id %in% bad_ids)

### Fix Issue with Timeout Indicator being flipped
game_ids <- unique(train$game_id)
j <- 0
for(i in 1:length(game_ids)) {
  print(i)
  x <- filter(train, game_id == game_ids[i])
  y <- filter(x, timeout_ind == 1,
              description == paste(x$opponent[1], " Timeout"))
  if(nrow(y) > 0) {
    x$timeout_diff <- x$timeout_diff * -1
    x$timeout_ind <- x$timeout_ind * -1
    train$timeout_diff[train$game_id == game_ids[i]] <- x$timeout_diff
    train$timeout_ind[train$game_id == game_ids[i]] <- x$timeout_ind
    j <- j + 1
  }
}

### Save Training Data

```

```
save("train", file = "cleaned_training_data.rda")
```

Fit Win Probability Models

```
### fit_model.R
### Obtains coefficient estimates for smoothed win probability model
library(dplyr)
load("cleaned_training_data.rda")

mnum <- 1 ### Set to 0, 1, 2, or 3

### Set Rolling Intervals
### Model 1 (90% Overlap):
###   - 600 to 2400 seconds remaining, 100 second intervals
###   - 100 to 600 seconds remaining, 50 second intervals
###   - 10 to 100 seconds remaining, 5 second intervals
###   - 0 to 10 second remainings, 2 second intervals
###   - 0 to 1 second remaining
###   Model 0 and 4 includes same time/stamp breakdown

if(mnum == 0 | mnum == 1) {
  max_sec <- c(seq(2400, 600, -10), seq(600, 100, -5),
               seq(100, 10, -1), seq(10, 2, -1), 1)
  min_sec <- c(seq(2300, 500, -10), seq(550, 50, -5),
               seq(95, 5, -1), seq(8, 0, -1), 0)
  n <- length(max_sec)
}

### Model 2 (90% Overlap):
###   - 600 to 2400 seconds remaining, 60 second intervals
###   - 100 to 600 seconds remaining, 30 second intervals
###   - 20 to 100 seconds remaining, 10 second intervals
###   - 0 to 20 second remainings, 2 second intervals
###   - 0 to 1 second remaining

if(mnum == 2) {
  max_sec <- c(seq(2400, 600, -6), seq(600, 100, -3),
               seq(100, 20, -1), seq(20, 2, -1), 1)
  min_sec <- c(seq(2340, 540, -6), seq(570, 70, -3),
               seq(90, 10, -1), seq(18, 0, -1), 0)
  n <- length(max_sec)
}
```


Predict Win Probability Models on Test Set

```

### model_cv.R
### Predicts fitted win probability models on test set data
### Cross Validates over different values of LOESS span parameters

library(readr)
library(dplyr)

### Fits Test win probability on 2018-19 season games
files <- unlist(sapply(dir("test_pbp_data",
                          full.names = T), dir,
                          full.names = T))
n <- length(files)

### Define Logit Function
logit <- function(x) {
  tmp <- exp(x)
  case_when(
    tmp == Inf ~ 1,
    tmp == -Inf ~ 0,
    T ~ tmp/(1 + tmp)
  )
}

### Set Model Number
model_n <- 0
coeffs <- read_csv(paste0("model_coefficients/model_",
                          model_n, "_coeffs.csv"),
                  col_types = cols())

### Set Coefficients to achieve deterministic
### relationship at max_time = 0
coeffs$estimate[coeffs$max_time <= 2 &
                coeffs$coefficient == "favored_by"] <- 0

### Cross Validate over span
spans <- seq(0.1, 0.5, 0.1)
for(j in 1:length(spans)) {
  ### Fit Loess Models to get smooth functions
  ### of coefficient estimate over time
  score_diff_smooth <-
    loess(estimate ~ max_time,
          data = filter(coeffs, coefficient == "score_diff"),

```



```

### Capture Game Determinism
index <- x$secs_remaining == 0
sc_diff[index] <- 20
fb[index] <- predict(favored_by_smooth, newdata = 1)
to_diff[index] <- predict(timeout_diff_smooth, newdata = 1)
to_ind[index] <- predict(timeout_ind_smooth, newdata = 1)

### Compute log odds of winning
log_odds <-
  sc_diff * x$score_diff +
  fb * x$home_favored_by +
  to_ind * (x$home_timeout_ind -
            x$away_timeout_ind) * switch +
  to_diff * (x$home_time_out_remaining -
            x$away_time_out_remaining) * switch

x <- mutate(x,
  "win_prob" = logit(log_odds),
  "win" = ifelse(score_diff[nrow(x)] > 0, 1, 0),
  "favored_by" = home_favored_by,
  "timeout_ind" = home_timeout_ind -
    away_timeout_ind,
  "timeout_diff" = home_time_out_remaining -
    away_time_out_remaining,
  "model" = model_n,
  "span" = "mixed") %>%
  select(game_id, date, win, win_prob,
    secs_remaining, score_diff, favored_by,
    timeout_ind, timeout_diff, model, span)

if(any(is.na(x$win_prob))) {
  break
}

### Save Results
if(i == 1) {
  test <- x
}else{
  test <- bind_rows(test, x)
}
}
}

write.csv(test, paste0("testcv_results/model_",
  model_n, "span_", spans[j], ".csv"),

```

```

        row.names = F)
}

```

Compute Model MSE, Log-Loss, and Misclassification Rate

```

### eval.R
### Computes MSE, Misclassification Rate
### and Log-Loss for different models

library(dplyr)
library(readr)

### Loss Functions
log_loss <- function(x, y) {
  eps <- 1e-16
  x[x < eps] <- eps
  x[x > (1 - eps)] <- 1 - eps
  return(-mean(y * log(x) + (1-y) * log(1-x)))
}

mse <- function(x, y) {
  return(mean((x-y)^2))
}

misclass_rate <- function(x, y) {
  return(mean(round(x) != y))
}

### Recreate ggplot2 colors
### Copied from https://stackoverflow.com/questions/8197559/
### emulate-ggplot2-default-color-palette
gg_color_hue <- function(n) {
  hues = seq(15, 375, length = n + 1)
  hcl(h = hues, l = 65, c = 100)[1:n]
}

files <- dir("testcv_results", full.names = T)

### Entire Game Evaluation
for(i in 1:length(files)) {
  cat(i, "of", length(files), "\n")
  test <- read_csv(files[i], col_types = cols())
  model <- substr(files[i], 22, 22)
}

```

```

span <- ifelse(model == "5", "multi-span",
              substr(files[i], 28, 30))
model <- ifelse(model == "5", "4", model)
df <- tibble(
  "model" = model,
  "span" = span,
  "log_loss" = log_loss(test$win_prob, test$win),
  "mse" = mse(test$win_prob, test$win),
  "misclass_rate" = misclass_rate(test$win_prob,
                                  test$win)
)

if(i == 1) {
  results <- df
}else{
  results <- bind_rows(results, df)
}
}

max_times <- seq(100, 2400, 10)
min_times <- seq(0, 2300, 10)
### Entire Game Evaluation
for(i in 1:length(files)) {
  cat(i, "of", length(files), "\n")
  test <- read_csv(files[i], col_types = cols())

  model <- substr(files[i], 22, 22)
  span <- ifelse(model == "5", "multi-span",
                substr(files[i], 28, 30))
  model <- ifelse(model == "5", "4", model)

  for(j in 1:(length(min_times))) {
    x <- filter(test,
                secs_remaining >= min_times[j],
                secs_remaining <= max_times[j+1])
    df <- tibble(
      "model" = model,
      "span" = span,
      "log_loss" = log_loss(x$win_prob, x$win),
      "mse" = mse(x$win_prob, x$win),
      "misclass_rate" = misclass_rate(x$win_prob,
                                      x$win),
      "min_time" = min_times[j],
      "max_time" = max_times[j]
    )
  }
}

```

```

    )

    if(i == 1 & j == 1) {
      time_results <- df
    }else{
      time_results <- bind_rows(time_results, df)
    }
  }
}

write.csv(results, "testcv_results/results.csv",
          row.names = F)
write.csv(time_results, "testcv_results/time_results.csv",
          row.names = F)

```

Computing Net Scoring Runs

```

### score_runs.R
### Computes net score differential
### Into/out of every point of all games
### For intervals of varying length

library(dplyr)

load("cleaned_training_data.rda")
train <- mutate(train, secs_elapsed = 2400 - secs_remaining)
game_ids <- unique(train$game_id)

### Score Run Function
score_run <- function(j, interval) {
  if(interval > 0) {
    if(x$secs_remaining[j] < interval) {
      return(NA)
    }
    scores <- x$score_diff[x$secs_elapsed >=
                          x$secs_elapsed[j] &
                          x$secs_elapsed <=
                          (x$secs_elapsed[j] + interval)]
  }else{
    if(x$secs_elapsed[j] < abs(interval)) {
      return(NA)
    }
    scores <- x$score_diff[x$secs_elapsed <=

```

```

        x$secs_elapsed[j] &
        x$secs_elapsed >=
        (x$secs_elapsed[j] + interval))
    }
    return(scores[length(scores)] - scores[1])
}

intervals <- c(seq(-300, -60, 60), seq(60, 300, 60))
n <- length(game_ids)
for(i in 1:n) {
  if(i %% 100 == 0) {
    cat(i, "of", n, "\n")
  }
  ### Remove Duplicate Time Events
  x <- filter(train, game_id == game_ids[i]) %>%
    filter(!duplicated(secs_elapsed))
  nx <- nrow(x)
  if(nx > 0) {
    for(k in 1:length(intervals)) {
      if(k == 1) {
        df <- mutate(x,
                      "score_run" =
                        sapply(1:nx,
                              score_run,
                              interval = intervals[k]),
                      "run_interval" = intervals[k])
      }else{
        df <- df %>% bind_rows(
          mutate(x,
                 "score_run" =
                   sapply(1:nx, score_run,
                         interval = intervals[k]),
                 "run_interval" = intervals[k]))
      }
    }
    if(i == 1) {
      train_runs <- df
    }else{
      train_runs <- bind_rows(train_runs, df)
    }
  }
}

```

```
train_runs <- filter(train_runs, !is.na(score_run))
save("train_runs", file = "train_runs.rda")
```

Points Above Expectation Framework

```
### pts_above_exp.R
### Computes after timeout
### Points above expectation

library(dplyr)
library(lme4)
load("train_runs.rda")

# Stripwhite function
stripwhite <- function(x) gsub("\\s*$", "", gsub("^\\s*", "", x))

### All Pre/Post Pairs
x <- filter(train_runs, run_interval < 0) %>%
  mutate("abs_interval" = abs(run_interval))
y <- filter(train_runs, run_interval > 0) %>%
  select(game_id, play_id, score_run, run_interval)

### All Pairs
pre_post_pairs_all <- inner_join(
  x, y,
  by = c("game_id", "play_id", "abs_interval" = "run_interval"),
  suffix = c("_pre", "_post"))

pre_post_pairs_all <- rename(
  pre_post_pairs_all, "run_interval_pre" = run_interval,
  "run_interval_post" = abs_interval)

no_timeout_pairs <- filter(pre_post_pairs_all,
                           !grepl("Timeout", description))
timeout_pairs <- filter(pre_post_pairs_all,
                        grepl("Timeout", description))

### Data Cleaning
timeout_pairs$timeout_taker <-
  stripwhite(gsub(" Timeout", "", timeout_pairs$description))
timeout_pairs <- mutate(
  timeout_pairs,
  "score_run_pre" = ifelse(opponent == timeout_taker,
```

```

        -score_run_pre, score_run_pre),
"score_run_post" = ifelse(opponent == timeout_taker,
        -score_run_post, score_run_post),
"favored_by" = ifelse(opponent == timeout_taker,
        -favored_by, favored_by))

### Predicted Post Interval Run
lmer.60 <-
  lmer(score_run_post ~ favored_by + score_diff +
        secs_elapsed + score_run_pre +
        (1|game_id),
        data = filter(no_timeout_pairs,
                        run_interval_post == 60))
lmer.120 <-
  lmer(score_run_post ~ favored_by + score_diff +
        secs_elapsed + score_run_pre +
        (1|game_id),
        data = filter(no_timeout_pairs,
                        run_interval_post == 120))
lmer.180 <-
  lmer(score_run_post ~ favored_by + score_diff +
        secs_elapsed + score_run_pre +
        (1|game_id),
        data = filter(no_timeout_pairs,
                        run_interval_post == 180))
lmer.240 <-
  lmer(score_run_post ~ favored_by + score_diff +
        secs_elapsed + score_run_pre +
        (1|game_id),
        data = filter(no_timeout_pairs,
                        run_interval_post == 240))
lmer.300 <-
  lmer(score_run_post ~ favored_by + score_diff +
        secs_elapsed + score_run_pre +
        (1|game_id),
        data = filter(no_timeout_pairs, run_interval_post == 300))

### Predict Post Time Out Runs
timeout_pairs$pred_score_run_post <- NA
timeout_pairs$pred_score_run_post[timeout_pairs$
                                run_interval_post == 60] <-
  predict(lmer.60,
          newdata = timeout_pairs[timeout_pairs$

```

```

                                run_interval_post == 60,],
    allow.new.levels = T)
timeout_pairs$pred_score_run_post[timeout_pairs$
                                run_interval_post == 120] <-
    predict(lmer.120,
            newdata = timeout_pairs[timeout_pairs$
                                    run_interval_post == 120,],
            allow.new.levels = T)
timeout_pairs$pred_score_run_post[timeout_pairs$
                                run_interval_post == 180] <-
    predict(lmer.180,
            newdata = timeout_pairs[timeout_pairs$
                                    run_interval_post == 180,],
            allow.new.levels = T)
timeout_pairs$pred_score_run_post[timeout_pairs$
                                run_interval_post == 240] <-
    predict(lmer.240,
            newdata = timeout_pairs[timeout_pairs$
                                    run_interval_post == 240,],
            allow.new.levels = T)
timeout_pairs$pred_score_run_post[timeout_pairs$
                                run_interval_post == 300] <-
    predict(lmer.300,
            newdata = timeout_pairs[timeout_pairs$
                                    run_interval_post == 300,],
            allow.new.levels = T)

### Pts Above Expectation
timeout_pairs <- mutate(
  timeout_pairs,
  "pts_above_exp" = score_run_post - pred_score_run_post,
  "timeout_type" = ifelse(timeout_taker == "Official TV",
                          "Media Timeout", "Non-Media Timeout"))

```

Imputation Evaluation

```

### imputation_evaluation.R
### Allows for evaluation of imputed
### point spreads on 2018-19 test set

library(ncaahoopR)
x <- read.csv("2019_Final.csv", as.is = T)

```



```

files <- unlist(sapply(dir("test_pbp_data",
                          full.names = T), dir,
                          full.names = T))

n <- length(files)
vegas_lines <- c()
imputed_lines <- c()

for(i in 1:n) {
  if(i %% 100 == 0) {
    print(i)
  }
  y <- read.csv(files[i], as.is = T)
  home <- dict$NCAA[dict$ESPN == y$home[1] |
                    dict$ESPN_PBP == y$home[1]]
  away <- dict$NCAA[dict$ESPN == y$away[1] |
                    dict$ESPN_PBP == y$away[1]]
  if(length(home) == 0) {
    home <- NA
  }
  if(length(away) == 0) {
    away <- NA
  }
  date_line <- y$date[1]
  pred_line <- filter(x, team == home,
                      opponent == away,
                      date == date_line) %>%
    pull(pred_score_diff)
  true_line <- y$home_favored_by[1]

  if(length(c(pred_line, true_line)) == 2) {
    if(!is.na(true_line) & !is.na(pred_line)) {
      if(true_line != pred_line) {
        vegas_lines <-
          c(vegas_lines, true_line)
        imputed_lines <-
          c(imputed_lines, pred_line)
      }
    }
  }
}

files <- unlist(sapply(dir("pbp_data", full.names = T),
                      dir, full.names = T))
n <- length(files)

```

```

for(i in 1:n) {
  if(i %% 100 == 0) {
    print(i)
  }
  y <- read.csv(files[i], as.is = T)
  home <- dict$NCAA[dict$ESPN == y$home[1] |
                    dict$ESPN_PBP == y$home[1]]
  away <- dict$NCAA[dict$ESPN == y$away[1] |
                    dict$ESPN_PBP == y$away[1]]
  if(length(home) == 0) {
    home <- NA
  }
  if(length(away) == 0) {
    away <- NA
  }
  date_line <- y$date[1]
  pred_line <- filter(x, team == home,
                     opponent == away,
                     date == date_line) %>%
    pull(pred_score_diff)
  true_line <- y$home_favored_by[1]

  if(length(c(pred_line, true_line)) == 2) {
    if(!is.na(true_line) & !is.na(pred_line)) {
      if(true_line != pred_line) {
        vegas_lines <- c(vegas_lines,
                        true_line)
        imputed_lines <- c(imputed_lines,
                          pred_line)
      }
    }
  }
}

df <- data.frame("vegas_line" = vegas_lines,
                 "imputed_line" = imputed_lines)
write.csv(df, "line_imputation.csv",
          row.names = F)

```

Appendix B

Code for Graphics and Tables

Introduction

```
library(dplyr)
library(ggplot2)
load("cleaned_training_data.rda")
train <- mutate(train, secs_elapsed = 2400 - secs_remaining)

### Figure 1.1
timeouts <- filter(train, grepl("Timeout", description))
ggplot(timeouts, aes(x = secs_elapsed, y = ..density..)) +
  geom_histogram(bins = 100, fill = "orange") +
  theme_bw() +
  theme(plot.title = element_text(size = 16, hjust = 0.5),
        plot.subtitle = element_text(size = 12, hjust = 0.5),
        axis.title = element_text(size = 14)) +
  labs(x = "Seconds Elapsed",
       y = "Density",
       title = "Histogram of NCAA Basketball Timeouts",
       subtitle = "All Timeout Types") +
  scale_x_continuous(breaks = seq(0, 2400, 400),
                    limits = c(0, 2400))

### Figure 1.2
ggplot(filter(timeouts, description != "Official TV Timeout"),
       aes(x = secs_elapsed, y = ..density..)) +
  geom_histogram(bins = 100, fill = "orange") +
  theme_bw() +
  theme(plot.title = element_text(size = 16, hjust = 0.5),
        plot.subtitle = element_text(size = 12, hjust = 0.5),
        axis.title = element_text(size = 14)) +
  labs(x = "Seconds Elapsed",
```

```

    y = "Density",
    title = "Histogram of NCAA Basketball Timeouts",
    subtitle = "Non-Media Timeouts") +
  geom_vline(xintercept = seq(240, 2260, 240),
             alpha = 0.5, lty = 2, size = 1.1) +
  scale_x_continuous(breaks = seq(0, 2400, 400),
                     limits = c(0, 2400))

```

Chapter 2

```

### Figure 2.1
library(ggplot2)
lines <- read.csv("line_imputation.csv",
                  as.is = T)
ggplot(lines, aes(x = vegas_line, y = imputed_line)) +
  geom_vline(xintercept = 0, lty = 2) +
  geom_hline(yintercept = 0, lty = 2) +
  geom_point(alpha = 0.2) +
  geom_smooth(method = "lm", color = "orange") +
  theme_bw() +
  theme(plot.title = element_text(size = 16,
                                   hjust = 0.5),
        plot.subtitle = element_text(size = 12,
                                       hjust = 0.5),
        axis.title = element_text(size = 14)) +
  labs(x = "Vegas Point Spread",
       y = "Imputed Point Spread",
       title = "Comparison of Point Spreads",
       subtitle = "Home Team Perspective") +
  xlim(c(-30, 40)) +
  ylim(c(-30, 40)) +
  annotate(geom = "label", x = -20, y = 30,
         label = expression(R^2 == 0.87))

```

Chapter 3

```

### Figure 3.1
library(dplyr)
library(ggplot2)
gg_color_hue <- function(n) {
  hues = seq(15, 375, length = n + 1)
  hcl(h = hues, l = 65, c = 100)[1:n]
}

```

```

}

z <- read.csv("model_coefficients/model_0_coeffs.csv",
              as.is = T)
ggplot(z, aes(x = 2400 - max_time,
              y = estimate,
              group = coefficient)) +
  facet_wrap(~coefficient, ncol = 2, scales = "free") +
  geom_ribbon(aes(ymax = estimate + 2 * std_error,
                ymin = estimate - 2 * std_error),
            alpha = 0.8, fill = "skyblue") +
  geom_point(size = 0.35, alpha = 0.2) +
  geom_smooth(data = filter(z, coefficient == "score_diff"),
             method = "loess", formula = "y~x",
             span = 0.1, size = 0.7) +
  geom_smooth(data = filter(z, coefficient != "score_diff"),
             method = "loess",
             formula = "y~x",
             span = 0.4,
             size = 0.7) +
  geom_hline(yintercept = 0, lty = 2) +
  theme_bw() +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 16,
                                   hjust = 0.5),
        plot.subtitle = element_text(size = 12,
                                      hjust = 0.5),
        axis.title = element_text(size = 14)) +
  labs(x = "Seconds Remaining",
       y = "Coefficient Estimate",
       title = "Win Probability Model",
       subtitle = "Coefficients Over Time",
       subtitle = "Model 0") +
  scale_x_continuous(breaks = seq(0, 2400, 400),
                    limits = c(0, 2400))

### Figure 3.2
z <- read.csv("model_coefficients/model_1_coeffs.csv",
              as.is = T)
ggplot(z, aes(x = 2400 - max_time,
              y = estimate,
              group = coefficient)) +
  facet_wrap(~coefficient, ncol = 2, scales = "free") +

```

```

geom_ribbon(aes(ymax = estimate + 2 * std_error,
               ymin = estimate - 2 * std_error),
           alpha = 0.8, fill = "skyblue") +
geom_point(size = 0.35, alpha = 0.2) +
geom_smooth(data = filter(z, coefficient == "favored_by"),
            method = "loess", formula = "y~x",
            span = 0.4, size = 0.7) +
geom_smooth(data = filter(z, coefficient != "favored_by"),
            method = "loess", formula = "y~x",
            span = 0.1, size = 0.7) +
geom_hline(yintercept = 0, lty = 2) +
geom_hline(yintercept = 0, lty = 2) +
theme_bw() +
theme(legend.position = "bottom",
      plot.title = element_text(size = 16,
                                hjust = 0.5),
      plot.subtitle = element_text(size = 12,
                                   hjust = 0.5),
      axis.title = element_text(size = 14)) +
labs(x = "Seconds Elapsed",
     y = "Coefficient Estimate",
     title = "Win Probability Model Coefficients Over Time",
     subtitle = "Model 1") +
scale_x_continuous(breaks = seq(0, 2400, 400),
                  limits = c(0, 2400))

### Figure 3.3
results <- read.csv("testcv_results/results.csv",
                  as.is = T)
ggplot(results, aes(x = model, y = log_loss)) +
  geom_point(aes(color = as.character(span)),
            size = 2) +
  geom_line(aes(color = as.character(span))) +
  theme_bw() +
  labs(x = "Model",
       y = "Log Loss",
       title = "Model Performance on Test Set",
       subtitle = "Log Loss",
       color = "LOESS Span") +
  theme(plot.title = element_text(size = 16,
                                  hjust = 0.5),
        plot.subtitle = element_text(size = 12,
                                      hjust = 0.5),

```

```

axis.title = element_text(size = 14),
legend.position = "bottom")

### Figure 3.4
time_results <- read.csv("testcv_results/time_results.csv",
                        as.is = T)
p1 <- mutate(time_results, "model_id" = paste(model, span)) %>%
  filter(model_id %in% c("0 0.5", "4 0.5"),
         !is.na(log_loss)) %>%
  group_by(max_time) %>%
  summarise("log_loss" = min(log_loss)) %>%
  inner_join(time_results, by = c("max_time",
                                "log_loss")) %>%

  ggplot(aes(x = 2400 - max_time,
             y = log_loss)) +
  geom_point(aes(col = as.character(model))) +
  theme_bw() +
  labs(x = "",
       y = "Log Loss",
       title = "Model Performance on
Test Set Over Time",
       subtitle = "Best Models Only") +
  theme(plot.title = element_text(size = 16,
                                  hjust = 0.5),
        plot.subtitle = element_text(size = 12,
                                      hjust = 0.5),
        axis.title = element_text(size = 14),
        legend.position = "none") +
  scale_color_manual(values = gg_color_hue(2)) +
  scale_x_continuous(breaks = seq(0, 2400, 400),
                    limits = c(0, 2400))

time_results2 <- mutate(time_results, "model_id" =
                        paste(model, span)) %>%
  filter(model_id %in% c("0 0.5", "4 0.5"),
         !is.na(misclass_rate))
p2 <- group_by(time_results2, max_time) %>%
  summarise("misclass_rate" = min(misclass_rate)) %>%
  inner_join(time_results2, by = c("max_time",
                                "misclass_rate")) %>%
  filter(!duplicated(max_time)) %>%
  ggplot(aes(x = 2400 - max_time,

```

```

        y = misclass_rate)) +
geom_point(aes(col = as.character(model))) +
theme_bw() +
labs(x = "Seconds Elapsed",
     y = "Misclassification Rate",
     color = "Better Model") +
theme(plot.title = element_text(size = 16,
                                hjust = 0.5),
      plot.subtitle = element_text(size = 12,
                                    hjust = 0.5),
      axis.title = element_text(size = 14),
      legend.position = "bottom",
      legend.title = element_text(size = 10)) +
scale_color_manual(values = gg_color_hue(2),
                   labels = c("No Timeout Indicator
                              (Model 0)",
                              "Timeout Indicator
                              (Model 4)")) +
scale_x_continuous(breaks = seq(0, 2400, 400),
                   limits = c(0, 2400))
gridExtra::grid.arrange(p1, p2)

### Figure 3.5
z <- read.csv("model_coefficients/model_1_coeffs.csv",
              as.is = T) %>%
  filter(coefficient == "timeout_ind")
p1 <- ggplot(z, aes(x = 2400 - max_time,
                   y = estimate,
                   group = coefficient)) +
  geom_ribbon(aes(ymax = estimate + 2 * std_error,
                ymin = estimate - 2 * std_error),
            alpha = 0.8, fill = "skyblue") +
  geom_point(alpha = 0.2, size = 0.25) +
  geom_hline(yintercept = 0, lty = 2) +
  geom_hline(yintercept = 0, lty = 2) +
  theme_bw() +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 16,
                                    hjust = 0.5),
        plot.subtitle = element_text(size = 12,
                                       hjust = 0.5),
        axis.title = element_text(size = 10)) +
  labs(x = "Seconds Elapsed",
       y = "Coefficient Estimate",

```



```

    title = "Timeout Indicator
    Coefficient Estimate Over Time") +
scale_x_continuous(breaks = seq(0, 2400, 400),
                    limits = c(0, 2400)) +
geom_vline(xintercept =
            2400 - c(2400, 2020, 1270, 400),
            lty = 2, col = "skyblue") +
geom_vline(xintercept = 2400 - c(2190, 1640, 1040),
            col = "red", lty = 2)
p2 <- group_by(time_results2, max_time) %>%
  summarise("misclass_rate" =
            min(misclass_rate)) %>%
  inner_join(time_results2,
             by = c("max_time", "misclass_rate")) %>%
  filter(!duplicated(max_time)) %>%
  ggplot(aes(x = 2400 - max_time,
             y = misclass_rate)) +
  geom_point(aes(col = as.character(model))) +
  theme_bw() +
  labs(x = "Seconds Elapsed",
       y = "Minimum Log Loss",
       title = "Model Misclassification
       Rate on Test Set Over Time",
       subtitle = "Best Models Only",
       color = "Model w/ Better Misclassification Rate") +
  theme(plot.title = element_text(size = 16,
                                   hjust = 0.5),
        plot.subtitle = element_text(size = 12,
                                       hjust = 0.5),
        axis.title = element_text(size = 10),
        legend.position = "bottom") +
  scale_color_manual(values = gg_color_hue(2),
                     labels =
                       c("No Timeout Indicator (Model 0)",
                         "Timeout Indicator (Model 4)")) +
  scale_x_continuous(breaks = seq(0, 2400, 400),
                     limits = c(0, 2400)) +
  geom_vline(xintercept = 2400 - c(2400, 2020, 1270, 400),
             lty = 2, col = "skyblue") +
  geom_vline(xintercept = 2400 - c(2190, 1640, 1040),
             col = "red", lty = 2)
gridExtra::grid.arrange(p1, p2)

```

Chapter 4

```

### Figure 4.1
load("train_runs.rda")
library(forcats)
library(dplyr)
library(ggplot2)
### Data Cleaning
timeouts <- filter(train_runs,
                    grepl("Timeout", description))

# Stripwhite function
stripwhite <- function(x) gsub("\\s*$", "", gsub("^\\s*", "", x))
timeouts$timeout_taker <-
  stripwhite(gsub(" Timeout", "", timeouts$description))
timeouts <- mutate(timeouts,
                   score_run = ifelse(opponent == timeout_taker,
                                       -score_run, score_run),
                   favored_by = ifelse(opponent == timeout_taker,
                                       -favored_by, favored_by))

non_tv <- filter(timeouts,
                 !grepl("Official TV Timeout", description))
tv <- filter(timeouts,
             grepl("Official TV Timeout", description))

# ggcolors
gg_color_hue <- function(n) {
  hues = seq(15, 375, length = n + 1)
  hcl(h = hues, l = 65, c = 100)[1:n]
}

ggplot(non_tv, aes(x = as.factor(abs(run_interval)),
                  y = score_run)) +
  geom_boxplot(aes(fill = run_interval > 0)) +
  theme_bw() +
  theme(plot.title = element_text(size = 16,
                                   hjust = 0.5),
        plot.subtitle = element_text(size = 12,
                                       hjust = 0.5),
        axis.title = element_text(size = 14),
        legend.position = "bottom") +
  labs(x = "Interval Length (Seconds)",
       y = "Net Score Differential During Interval\n(Perspective of Team Taking Timeout)",

```

```

    fill = "",
    title = "Net Score Differential
Before and After Timeouts",
    subtitle = "Non-Media Timeouts") +
scale_fill_manual(values = gg_color_hue(2),
                  labels = c("Before Timeout", "After Timeout"))

### Table 4.1
library(knitr)
library(kableExtra)
y <- group_by(non_tv, run_interval) %>%
  summarise("mean" = mean(score_run),
            "sd" = sd(score_run)) %>%
  ungroup()
y <- rbind(
  y[1:5,] %>%
    mutate("run_interval" = abs(run_interval),
           "to_relativity" = "Before Timeout"),
  y[6:10,] %>%
    mutate("to_relativity" = "After Timeout"))

y <- cbind(round(y[1:5, 1:3], 2),
           round(y[10:6, 2:3], 2))[5:1, c(1, 2, 4, 3, 5)]
mutate(y, "before" = paste0(sprintf("%.2f", y$mean),
                             " ($\\pm$",
                             sprintf("%.2f", y$sd), ")"),
        "after" = paste0(sprintf("%.2f", y$mean.1),
                          " ($\\pm$",
                          sprintf("%.2f", y$sd.1), ")")) %>%
select(run_interval, before, after) %>%
rename("Interval Length (Seconds)" = run_interval,
       "Pre-Timeout Mean Net Score Differential
" = before,
       "Post-Timeout Mean Net Score Differential"
= after) %>%
kable(caption = "\\label{tab:4.1} Pre and
Post Non-Media Timeout Summary Statistics",
      "latex",
      booktabs = T,
      row.names = F, align = "ccccc",
      escape = F) %>%
column_spec(1:3, width = "3cm") %>%
column_spec(2:3, width = "6cm")

```

```

### Figure 4.2
ggplot(tv, aes(x = as.factor(abs(run_interval)),
               y = score_run)) +
  geom_boxplot(aes(fill = run_interval > 0)) +
  theme_bw() +
  theme(plot.title = element_text(size = 16,
                                   hjust = 0.5),
        plot.subtitle = element_text(size = 12,
                                       hjust = 0.5),
        axis.title = element_text(size = 14),
        legend.position = "bottom") +
  labs(x = "Interval Length (Seconds)",
       y = "Net Score Differential During Interval
\n(Perspective of Home Team)",
       fill = "",
       title = "Net Score Differential Before
and After Timeouts",
       subtitle = "Media Timeouts") +
  scale_fill_manual(values = gg_color_hue(2),
                    labels = c("Before Timeout", "After Timeout"))

### Table 4.2
y <- group_by(tv, run_interval) %>%
  summarise("mean" = mean(score_run),
            "sd" = sd(score_run)) %>%
  ungroup()
y <- rbind(
  y[1:5,] %>%
    mutate("run_interval" = abs(run_interval),
           "to_relativity" = "Before Timeout"),
  y[6:10,] %>%
    mutate("to_relativity" = "After Timeout"))

y <- cbind(round(y[1:5, 1:3], 2),
           round(y[10:6, 2:3], 2))[5:1, c(1, 2, 4, 3, 5)]
mutate(y, "before" = paste0(sprintf("%.2f", y$mean),
                           " ($\\pm$",
                           sprintf("%.2f", y$sd), ")"),
       "after" = paste0(sprintf("%.2f", y$mean.1),
                        " ($\\pm$", sprintf("%.2f", y$sd.1), ")")) %>%
  select(run_interval, before, after) %>%
  rename("Interval Length (Seconds)" = run_interval,
        "Pre-Timeout Mean Net Score Differential
" = before,

```

```

      "Post-Timeout Mean Net Score Differential
      " = after) %>%
kable(caption = "\\label{tab:4.2} Pre and Post
      Media Timeout Summary Statistics",
      "latex",
      booktabs = T,
      row.names = F, align = "ccccc",
      escape = F) %>%
column_spec(1, width = "3cm") %>%
column_spec(2:3, width = "6cm")

### Figure 4.3
pre_post_pairs <- inner_join(
  non_tv,
  select(non_tv, game_id, play_id, score_run, run_interval),
  by = c("game_id", "play_id"),
  suffix = c("_pre", "_post")) %>%
filter(run_interval_pre < 0, run_interval_post > 0)

ggplot(pre_post_pairs,
  aes(x = score_run_post - score_run_pre,
      y = ..density..)) +
facet_grid(run_interval_post ~ run_interval_pre) +
geom_histogram(bins = 20, fill = "orange") +
theme_bw() +
theme(plot.title = element_text(size = 16,
                                hjust = 0.5),
      plot.subtitle = element_text(size = 12,
                                   hjust = 0.5),
      axis.title = element_text(size = 14),
      legend.position = "bottom") +
labs(x = expression(
  paste("Improvement in Net Score Differential
        Following Timeout (", Delta[tij], ")"),
  sep = ""))
),
y = "Density",
color = "Post Timeout - Pre Timeout Delta",
title = "Difference in Before and After
Timeout Net Score Differential",
subtitle = "Non-Media Timeouts") +
geom_vline(xintercept = 0, lty = 2,
  size = 0.6, alpha = 0.5) +
scale_x_continuous(limits = c(-35, 35),

```

```

        breaks = seq(-20, 20, 20)) +
scale_y_continuous(limits = c(0, 0.12),
        breaks = seq(0, 0.12, 0.03))

### Figure 4.4
pre_post_pairs_tv <- inner_join(
  tv,
  select(tv, game_id, play_id, score_run, run_interval),
  by = c("game_id", "play_id"),
  suffix = c("_pre", "_post")) %>%
  filter(run_interval_pre < 0, run_interval_post > 0)

ggplot(pre_post_pairs_tv,
  aes(x = score_run_post - score_run_pre,
      y = ..density..)) +
  facet_grid(run_interval_post ~ run_interval_pre) +
  geom_histogram(bins = 20, fill = "red") +
  theme_bw() +
  theme(plot.title = element_text(size = 16,
                                   hjust = 0.5),
        plot.subtitle = element_text(size = 12,
                                       hjust = 0.5),
        axis.title = element_text(size = 14),
        legend.position = "bottom") +
  labs(x = expression(
    paste("Improvement in Net Score
          Differential Following Timeout (", Delta[hij], ")",
          sep = ""))
),
  y = "Density",
  color = "Post Timeout - Pre Timeout Delta",
  title = "Difference in Before and
  After Timeout Net Score Differential",
  subtitle = "Non-Media Timeouts") +
  geom_vline(xintercept = 0, lty = 2,
    size = 0.6, alpha = 0.5) +
  scale_x_continuous(limits = c(-35, 35),
    breaks = seq(-20, 20, 20)) +
  scale_y_continuous(limits = c(0, 0.12),
    breaks = seq(0, 0.12, 0.03))

### Table 4.3
cbind(group_by(pre_post_pairs,
  run_interval_pre,

```

```

        run_interval_post) %>%
  summarise(
    "non_media_delta" =
      round(mean(score_run_post - score_run_pre), 2),
    "sd" = round(sd(score_run_post - score_run_pre), 2)) %>%
  ungroup() %>%
  mutate(run_interval_pre =
    -run_interval_pre,
    "non_media_delta" =
      paste0(non_media_delta, " ($\\pm$", sd, ")")) %>%
  arrange(run_interval_pre) %>%
  select(-sd),
  group_by(pre_post_pairs_tv,
    run_interval_pre,
    run_interval_post) %>%
  summarise(
    "media_delta" = round(mean(score_run_post - score_run_pre), 2),
    "sd" = round(sd(score_run_post - score_run_pre), 2)) %>%
  ungroup() %>%
  mutate(run_interval_pre =
    -run_interval_pre,
    "media_delta" =
      paste0(media_delta, " ($\\pm$", sd, ")")) %>%
  arrange(run_interval_pre) %>%
  select(media_delta) %>%
  rename("Pre-Timeout Interval
    (Seconds)" = run_interval_pre,
    "Post-Timeout Interval
    (Seconds)" = run_interval_post,
    "$$\\bar\\Delta_{tij}$$
    (Non-Media Timeouts)" = non_media_delta,
    "$$\\bar\\Delta_{tij}$$
    (Media Timeouts)" = media_delta) %>%
  kable(caption = "\\label{tab:4.3} Summary
    Table of Post Timeout Improvement",
    "latex",
    escape = F,
    booktabs = T,
    row.names = F, align = "ccccc") %>%
  column_spec(1:2, width = "2cm") %>%
  column_spec(3:4, width = "4cm")

```

Figure 4.5

ggplot(filter(

```

pre_post_pairs,
-run_interval_pre == run_interval_post),
aes(x = secs_elapsed,
     y = score_run_post - score_run_pre)) +
facet_wrap(~run_interval_post) +
geom_smooth(col = gg_color_hue(2)[2]) +
geom_vline(xintercept = 1200, lty = 2, size = 1.2) +
theme_bw() +
theme(plot.title = element_text(size = 16, hjust = 0.5),
      plot.subtitle = element_text(size = 12, hjust = 0.5),
      axis.title = element_text(size = 14)) +
labs(x = "Seconds Elapsed",
     y = "Improvement in
Net Score Differential",
     fill = "",
     title = "Improvement in Net Score
Differential in Interval Following
Timeout \nOver Course of Game",
     subtitle = "Non-Media Timeouts") +
scale_x_continuous(breaks = seq(0, 2400, 600),
                   limits = c(0, 2400)) +
coord_cartesian(ylim=c(-0.5, 7))

```

Figure 4.6

```

ggplot(filter(pre_post_pairs_tv,
-run_interval_pre == run_interval_post),
aes(x = secs_elapsed,
     y = score_run_post - score_run_pre)) +
facet_wrap(~run_interval_post) +
geom_smooth(col = gg_color_hue(2)[1]) +
geom_vline(xintercept = 1200, lty = 2, size = 1.2) +
theme_bw() +
theme(plot.title = element_text(size = 16, hjust = 0.5),
      plot.subtitle = element_text(size = 12, hjust = 0.5),
      axis.title = element_text(size = 14)) +
labs(x = "Seconds Elapsed",
     y = "Improvement in Net Score Differential",
     fill = "",
     title = "Improvement in Net Score
Differential in Interval Following
Timeout \nOver Course of Game",
     subtitle = "Media Timeouts") +
scale_x_continuous(breaks = seq(0, 2400, 600),
                   limits = c(0, 2400)) +

```



```

coord_cartesian(ylim=c(-0.5, 7))

### Figure 4.7
library(sjPlot)
tab_model(lmer.60, lmer.120, lmer.180,
          lmer.240, lmer.300,
          dv.labels = c("60 Sec", "120 Sec",
                        "180 Sec", "240 Sec", "300 Sec"),
          show.ci = F)

### Figure 4.8
### Points above expectation computed
### in pts_above_exp.R
timeout_pairs <- mutate(
  timeout_pairs,
  "pts_above_exp" = score_run_post - pred_score_run_post,
  "timeout_type" =
    ifelse(timeout_taker == "Official TV",
           "Media Timeout", "Non-Media Timeout"))
ggplot(timeout_pairs,
       aes(x = as.factor(run_interval_post),
           y = pts_above_exp)) +
  geom_hline(yintercept = 0, lty = 2, col = "red", size = 1.1) +
  geom_boxplot(aes(fill = timeout_type), alpha = 0.5) +
  theme_bw() +
  labs(x = "Interval Length (Seconds)",
       y = "Points Above Expectation",
       fill = "Timeout Type",
       title = "Points Above Expectation in
Post-Timeout Intervals of Varying Length") +
  theme(plot.title = element_text(size = 16, hjust = 0.5),
        plot.subtitle = element_text(size = 12, hjust = 0.5),
        axis.title = element_text(size = 14),
        legend.position = "bottom")

### Figure 4.9
ggplot(filter(timeout_pairs, score_run_pre < 0),
       aes(x = as.factor(run_interval_post), y = pts_above_exp)) +
  geom_hline(yintercept = 0, lty = 2, col = "red", size = 1.1) +
  geom_boxplot(aes(fill = timeout_type), alpha = 0.5) +
  theme_bw() +
  labs(x = "Interval Length (Seconds)",

```

```

    y = "Points Above Expectation",
    fill = "Timeout Type",
    title = "Points Above Expectation After Timeouts",
    subtitle = "Following Negative Scoring Runs into Timeout") +
  theme(plot.title = element_text(size = 16, hjust = 0.5),
        plot.subtitle = element_text(size = 12, hjust = 0.5),
        axis.title = element_text(size = 14),
        legend.position = "bottom")

### Table 4.4
library(kableExtra)
group_by(filter(timeout_pairs,
                timeout_taker != "Official TV"),
  run_interval_post) %>%
summarise("mean_pae_ntv" =
  round(mean(pts_above_exp, na.rm = T), 3),
  "median_pae_ntv" =
  round(median(pts_above_exp, na.rm = T), 3),
  "sd_ntv" =
  round(sd(pts_above_exp, na.rm = T), 2)) %>%
inner_join(group_by(
  filter(timeout_pairs,
    timeout_taker == "Official TV"),
  run_interval_post) %>%
  summarise("mean_pae_tv" =
    round(mean(pts_above_exp, na.rm = T), 3),
    "median_pae_tv" =
    round(median(pts_above_exp, na.rm = T), 3),
    "sd_tv" =
    round(sd(pts_above_exp, na.rm = T), 2))) %>%
mutate("mean_pae_ntv" =
  paste0(mean_pae_ntv, " (\\pm", sd_ntv, ")"),
  "mean_pae_tv" =
  paste0(mean_pae_tv, " (\\pm", sd_tv, ")")) %>%
select(-sd_ntv, -sd_tv) %>%
rename("Post-Timeout Interval
Length (Seconds)" = run_interval_post,
  "Mean PAE (Non-Media Timeouts)" = mean_pae_ntv,
  "Median PAE (Non-Media
Timeouts)" = median_pae_ntv,
  "Mean PAE (Media Timeouts)" = mean_pae_tv,
  "Median PAE (Media
Timeouts)" = median_pae_tv) %>%
kable(caption = "\\label{tab:4.4}")

```

```

Summary Table of Points Above Expectation",
"latex",
escape = F,
booktabs = T,
row.names = F, align = "ccccc") %>%
column_spec(1:5, width = "3cm")

```

Figure 4.10

```

ggplot(filter(timeout_pairs, timeout_taker != "Official TV"),
  aes(x = secs_elapsed, y = pts_above_exp)) +
geom_smooth(col = gg_color_hue(2)[2]) +
facet_wrap(~run_interval_post) +
theme_bw() +
labs(x = "Seconds Elapsed",
  y = "Points Above Expectation",
  title = "Points Above Expectation in
Post-Timeout Intervals Over Course of Game",
  subtitle = "Non-Media Timeouts") +
theme(plot.title = element_text(size = 16, hjust = 0.5),
  plot.subtitle = element_text(size = 12, hjust = 0.5),
  axis.title = element_text(size = 14),
  legend.position = "bottom") +
scale_x_continuous(breaks = seq(0, 2400, 600),
  limits = c(0, 2400)) +
coord_cartesian(ylim=c(-0.5, 5))

```

Figure 4.11

```

ggplot(filter(timeout_pairs, timeout_taker == "Official TV"),
  aes(x = secs_elapsed, y = pts_above_exp)) +
geom_smooth(col = gg_color_hue(2)[1]) +
facet_wrap(~run_interval_post) +
theme_bw() +
labs(x = "Seconds Elapsed",
  y = "Points Above Expectation",
  title = "Points Above Expectation in
Post-Timeout Intervals Over Course of Game",
  subtitle = "Media Timeouts") +
theme(plot.title = element_text(size = 16, hjust = 0.5),
  plot.subtitle = element_text(size = 12, hjust = 0.5),
  axis.title = element_text(size = 14),
  legend.position = "bottom") +
scale_x_continuous(breaks = seq(0, 2400, 600),
  limits = c(0, 2400)) +
coord_cartesian(ylim=c(-0.5, 5))

```

```

### Figure 4.12
coeffs <- read.csv("model_coefficients/model_0_coeffs.csv",
                  as.is = T)
score_diff_smooth <-
  loess(estimate ~ max_time,
        data = filter(coeffs,
                      coefficient == "score_diff"),
        span = 0.5)

s.300 <- loess(pts_above_exp ~ secs_elapsed,
              data = filter(timeout_pairs,
                            timeout_taker != "Official TV",
                            run_interval_post == 300),
              span = 0.175)
sec <- seq(0, 2400, 1)

sd_smooth <- predict(score_diff_smooth,
                    newdata = 2400 - sec)
smooth300 <- predict(s.300, newdata = sec)
value300 <- (sd_smooth * smooth300)

df <- data.frame("time" = sec,
                 "value" = exp(value300))

ggplot(df, aes(x = time, y = value)) +
  geom_line(col = gg_color_hue(2)[2], size = 2) +
  scale_x_continuous(breaks = seq(0, 2400, 400),
                    limits = c(0, 2400)) +
  scale_y_continuous(breaks = seq(0, 1, 0.2),
                    limits = c(0, 1)) +
  theme_bw() +
  labs(x = "Seconds Elapsed",
       y = "Relative Value",
       title = "Relative Value of Taking a Timeout") +
  theme(plot.title = element_text(size = 16, hjust = 0.5),
        plot.subtitle = element_text(size = 12, hjust = 0.5),
        axis.title = element_text(size = 14))

```

References

- Bennette, C., & Vickers, A. (2012). Against quantiles: Categorization of continuous variables in epidemiologic research, and its discontents. *BMC Medical Research Methodology*, 12–21.
- Benz, L. (2018a). A Bayesian Examination of Win Probability and Timeout Usage in NCAA Men's Basketball. https://github.com/lbenz730/BIS_567_Final_Project/blob/master/Documents/Fall%202018/BIS%20567/Final%20Project/Benz_BIS_567_Final_Poster.pdf.
- Benz, L. (2018b). NCAA hoops model methodology. http://rpubs.com/lbenz730/ncaa_hoops_methodology.
- Benz, L. (2019). ncaahoopR: An R package for working with NCAA basketball play-by-play data. <https://github.com/lbenz730/ncaahoopR>.
- Bocskocsky, A., Ezekowitz, J., & Stein, C. (2014). The hot hand: A new approach to an old "fallacy". *MIT Sloan Sports Analytics Conference*.
- Burke, B. (2009). Modeling Win Probability for a College Basketball Game: A Guest Post from Brian Burke. <https://bit.ly/1FJIjd4>.
- Gilovich, T., Vallone, R., & Tversky, A. (1985). The hot hand in basketball: On the misperception of random sequences. *Cognitive Psychology*, 17(3), 295–314.
- Lopez, M. (2018). On the risks of categorizing a continuous variable (with an application to baseball data). <https://statsbylopez.netlify.com/post/on-the-risks-of-categorizing-a-continuous-variable/>.
- Lopez, M., Matthews, G., & Baumer, B. (2018). How often does the best team win? A unified approach to understanding randomness in North American Sport. *The Annals of Applied Statistics*, 12(4), 2483–2516.
- Mace, F. C., Lalli, J., Shea, M., & Nevin, J. (1992). Behavioral momentum in college basketball. *Journal of Applied Behavior Analysis*, 25(3).
- Massey, K. (1997). Statistical models applied to the rating of sports teams. *Honors Thesis, Bluefield College*.
- Miller, J., & Sanjurjo, A. (2018). Suprised by the gambler's and hot hand fallacies? A truth in the law of small numbers. *Econometrica*, 86(6), 2019–2047.

- NCAA. (2018). 2018-19 NCAA Division I Men's Basketball Regular Season Media Timeout Format (Rule 5-14.10).
- Permutt, S. (2011). The efficacy of momentum-stopping timeouts on short-term performance in the national basketball association. *Bryn Mawr College Senior Thesis*.
- Pomeroy, K. (2018). Mining point spread data: Home court advantage. <https://kenpom.com/blog/mining-point-spread-data-home-court-advantage/>.
- Roane, H., Kelly, M., Trosclair, N., & Hauer, L. (2004). Behavioral momentum in sports: A partial replication in women's basketball. *Journal of Applied Behavior Analysis*, 37(3), 385–390.
- Torvik, B. (2017). How I Built a (Crappy) Basketball Win Probability Model. <http://adamcwisports.blogspot.com/2017/07/how-i-built-crappy-basketball-win.html>.
- Wang, X. (2015). Learning statistics with R, from the ground up. http://xiaofei-wang.com/research/conf/JSM2015/2015JSM_talk.pdf.
- Weinbach, J. (2008). The art of the timeout. *Wall Street Journal*.