

AN INVESTIGATION OF UMPIRE PERFORMANCE USING PITCHF/X DATA VIA
LONGITUDINAL ANALYSIS

by

CHRISTOPHER JUAREZ

B.S., Kansas State University, 2010

A REPORT

submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE

Department of Statistics
College of Arts and Sciences

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2012

Approved by:

Major Professor
Abigail Jager, Ph.D

Copyright

CHRISTOPHER JUAREZ

2012

Abstract

Baseball has long provided statisticians a playground for analysis. In this report we discuss the history of Major League Baseball (MLB) umpires, MLB data collection, and the use of technology in sports officiating. We use PITCHf/x data to answer 3 questions. 1) Has the proportion of incorrect calls made by a major league umpire decreased over time? 2) Does the proportion of incorrect calls differ for umpires hired prior to the implementation of technology in evaluating umpire performance from those hired after? 3) Does the rate of change in the proportion of incorrect calls differ for umpires hired prior to the implementation of technology in evaluating umpire performance from those hired after?

PITCHf/x is a publicly available database which gathers characteristics for every pitch thrown in one of the 30 MLB parks. In 2002, MLB began to use camera technology in umpire evaluations; prior to 2007, the data were not publicly available. Data were collected at the pitch level and the proportion of incorrect calls was calculated for each umpire for the first third, second third, and last third of each of the seasons for 2008-2011. We collected data from retrosheet.org, which provides game summary information. We also determined the year of each umpire's MLB debut to differentiate pre- and post-technology hired umpires for our analysis.

We answered our questions of interest using longitudinal data analysis, using a random coefficients model. We investigated the choice of covariance structure for our random coefficients model using Akaike's Information Criterion and the Bayesian Information Criterion. Further, we compared our random coefficients model to a fixed slopes model and a general linear model.

Table of Contents

List of Figures	vi
List of Tables	vii
List of Models	viii
Acknowledgements	ix
Dedication	x
Chapter 1 - Introduction	1
History of Data Collection in Baseball	1
History of Umpires in Baseball	2
Recent New Developments of Technology in Sports	4
QuesTec	5
PITCHf/x	6
Previous Research	8
Questions of Interest	9
Chapter 2 - Data Collection	11
Pitch Level Data	11
Game Level Data	16
Umpire Level Data	17
Discussion	18
Process of Data Collection	18
Statistical Software	18
2008 Data	19
2009, 2010, 2011 Data	20
Chapter 3 - The Model	22
The Random Coefficients Model	22
WITHIN-UNIT VARIATION	26
AMONG UNIT-VARIATION	26
Choice of covariance structure	26
Chapter 4 - The Analysis	29

Choosing the Best Random Effects Model.....	29
Analysis of Model 2.....	31
Diagnostics Check	31
Interpreting the SAS output	33
Answering Our Questions of Interest.....	37
Random Coefficients Model vs. Fixed Slopes Model	38
Fixed Slopes Model vs. General Linear Model	40
Chapter 5 - Conclusion	42
Issues and Areas for Future Research.....	42
Appendix A – R Code.....	45
Appendix B – SAS Code	60
Appendix C – SAS Output.....	70

List of Figures

Figure 1.1 Television screenshot of baseball game with Pitch Trax, which utilizes PITCHf/x technology to show viewers where the ball crosses home plate (Fast March 2011)	7
Figure 1.2 Strikezone report for left handed pitches to Derek Jeter (Lefkowitz 2009)	8
Figure 2.1 A pictorial representation of the typical strike zone (retrieved from Strike Zone en.JPG from Wikipedia).	13
Figure 2.2 Spaghetti plot of incorrect calls by time period separated by umpires MLB debut era (Pre- and Post-technology implementation in evaluation).....	21
Figure 4.1 Residuals vs. Predicted Values for Model 2.....	32
Figure 4.2 Normal Probability Plot for Model 2.....	32
Figure 4.3 Observed (black) and estimated (blue) proportion of incorrect calls for Adrian Johnson and average proportion of incorrect calls for post-tech umpires (red, dotted) plotted	36
Figure 4.4 Plot of average estimates for Post-tech (red, solid) and Post-tech (blue, dashed) umpires.....	36

List of Tables

Table 2.1 Variables used to assess location of the ball as it crosses home plate (*Indicates a variable we created)	14
Table 2.2 Variables to assess the umpire's call (*Indicates a variable we created).....	15
Table 2.3 Time periods for which the proportion of incorrect calls is summarized.....	16
Table 2.4 Variables collected from www.retrosheet.org	16
Table 2.5 Information on the birth year and first occurrence as MLB umpire (* Indicates a variable we created)	17
Table 2.6 Contingency table of era by generation frequencies.....	18
Table 4.1 AIC and BIC for each model	30
Table 4.2 Covariance Parameter Estimates and Solution for Fixed Effects for Model 2	33
Table 4.3 A portion of the Solutions for Random Effects for Model 2	34
Table 4.4 Estimated and observed proportions of incorrect calls for umpire Adrian Johnson.....	35
Table 4.5 Fit statistics for random coefficients model and mixed effects model	39
Table 4.6 Fit Statistics for the general linear model	40

List of Models

Model I.....	22
Model II	23
Model III.....	24
Model IV.....	24
Model V	25
Model VI.....	25
Model VII.....	34
Model VIII	38
Model IX.....	38
Model X	40

Acknowledgements

I would like to express the deepest appreciation to my major professor, Dr. Abigail Jager, who has shown me patience and understanding. She allowed me the pleasure of being her first student as a major professor and I think we made a great team!

I would like to sincerely thank Dr. Leigh Murray for seeing something in me long before I did. At first I was afraid of her, but she took me on when I was an undergraduate and taught me how to have confidence in the knowledge that I possess.

My thanks go to Dr. Boyer for sticking around after retirement to serve on my committee.

Without the McNair Scholars Program, I never would have put graduate school in my future, so I must give them a huge thanks. Lora Boyer, Jon Tvitte, and the rest of the McNair staff are possibly the best resources I have available on the K-State campus. I wish their program the best of luck.

My fellow graduate students, especially Natalya Makaraova, Garth Highland, Indu Seetharaman, Seth Demel, and Carlie Shannon, who helped me through this process. I don't think it's possible to make it though without helping one another, and I'm glad they were there to help me.

My parents, John and Christine Juarez, constantly tell me how proud they are of my achievements. I think I turned out pretty great which reflects on them. Thanks Mom and Dad!

A special thanks to my friends, and anyone I forgot. They kept me sane, inspired me to push further, helped me along the way, or just listened to my rants. Everyone needs people like that in their life.

My appreciation goes to the faculty in the Department of Statistics. They have provided me a wonderful home these past 2 years, making it even harder to leave.

Lastly, I want to give a hug and thank you to Pam Schierer, Teresa Zerbe, and Angie Ladner. These fabulous ladies provided me a place to run for a quick laugh, snack, or answer to a question. I don't know how much work I kept them from doing, but they never seemed to mind.

Dedication

I dedicate this to me! Now go run and be free.



Chapter 1 - Introduction

"Statistics are the lifeblood of baseball. In no other sport are so many available and studied so assiduously by participants and fans. Much of the game's appeal, as a conversation piece, lies in the opportunity the fan gets to back up opinions and arguments with convincing figures, and it is entirely possible that more American boys have mastered long division by dealing with batting averages than in any other way." - Leonard Koppett in *A Thinking Man's Guide to Baseball* (1967)

History of Data Collection in Baseball

Baseball has long provided a playground for statisticians. Books such as *The Book: Playing the Percentages in Baseball*, websites like Baseball Prospectus and The Hardball Times, and even societies, like the Society of American Baseball Research, have been created by baseball statisticians and enthusiasts (Palmer 2007). Statisticians have even given the use of statistics in baseball a special name—sabermetrics. Sabermetrics is defined by Bill James as “the search for objective knowledge about baseball” (Grabiner). This term is based on the acronym SABR for the Society of American Baseball Research.

The Society of American Baseball Research, founded by L. Robert Davids in August of 1971 in Cooperstown, New York, began as an organization of baseball historians, statisticians and researchers. To date SABR has over 6,000 members worldwide and holds an annual national meeting where members can present research and meet former major league players (sabr.org).

Initially, data for each game was collected as a game log. Game logs are a collection of information for each game played including home/away, score by inning, batting line-up, day/night game, and more retrieved from www.retrosheet.org. While game logs can be found for games dating back to 1871, play-by-play datasets have only been available since 1984. Play-by-play data records information on every pitch of a game. Retrosheet.org was founded in order to collect as much play-by-play information as possible. Pete Palmer, a major contributor to sabermetrics, recalls that when he began baseball analysis work in the 1960s there were no play-by-play datasets of any kind. The Elias Sports Bureau was commissioned to produce

computerized play-by-play data for 1969 and 1970 by Eldon and Harlan Mills. This data was used for the Bureau's Player Win Average calculations, a new method devised to evaluate the probability of winning a game based on the number of players on base and the number of outs for the current batter. While impressive at this time point, the gathering of play-by-play data was not done extensively until 1984 (Palmer 2007).

Today, there are a number of resources with play-by-play data available to anyone with an internet connection. This availability has been attributed to Bill James (Palmer 2007). James was having trouble gathering data for his annual publication *Baseball Abstracts* in the late 1970s. James encouraged his readers to gather and share data by scoring games at ballparks or from radio or television broadcasts. James and his readers then created what is known as Project Scoresheet in 1984.

The collection of play-by-play data for Project Scoresheet was managed and continued by Gary Gillette in 1990. Gillette continued this work with his Baseball Workshop in 1996 and currently with 24-7 Baseball (Palmer 2007).

Dave Smith, a significant contributor to the collection of play-by-play data with Project Scoresheet, created Retrosheet in 1989 (Palmer 2007). Retrosheet collected pre-1984 games and currently has play-by-play information on nearly every game dating back to 1960. This resource is available to anyone at www.retrosheet.org. Retrosheet received game data from 1984-1990 from Gillette after Project Scoresheet ended. Gillette also made available raw game statistics for games from 1991-1998. From these Retrosheet created the game logs. STATS Inc. provided play-by-play data to Smith for years 1991-1992.

Altogether, hundreds of people have created a great deal of data available to analyze. And analyze they do!

History of Umpires in Baseball

Prior to 1858, three officials were commonly used for each game—one umpire chosen by each team and a neutral party to decide the split decisions. A single umpire was sanctioned in 1858. This umpire was sometimes a spectator or even a player that was chosen by the home team and consented upon by the visiting team captain (www.sdabu.com). When the first league was created in 1871, umpires in baseball were volunteers. The visiting team would submit 5 names, and the home team would pick an umpire from the list. It was not until 7 years

later, in 1878 that umpires began receiving payment by the National League of \$5 per game. Twenty years later, William Hulbert, the National League president formed the first umpiring staff. This staff included 20 men, from which the six teams chose their umpires. In 1882, the American Association created an umpiring staff that was hired, paid, and assigned to games by the league. These individuals were paid \$140 per month and received an additional \$3 for every day they were on the road (Gassko 2007).

The perception of the umpire has changed over time as well. In the 1870s the volunteers were often distinguished in appearance, wearing a top hat, coat, and cane. When organized umpiring began in 1882, umpires were given uniforms. It seems as though this is when the perception of the umpire changed from respectable to villainous (www.sdabu.com). Umpires were faced with frequent changes in rules, as well as abuse by players and fans. It was common for umpires to be spiked, kicked, cursed and spat at by players. Fans would throw a variety of objects while yelling profanities at the umpires. This created the need for police escorts. The leagues did little to combat this behavior because it helped boost ticket sales. It wasn't until 1903, with the urging of Byron Bancroft Johnson, that the umpire began to transition back to being a respected individual (www.sdabu.com).

While respectable, the job initially did not pay very well. In the 1900s umpires made anywhere from \$1,500 to \$2,000 per year, roughly equivalent to \$34,000 to \$46,000 today. However they also had to pay for their own transportation, lodging, and uniform expenses. The top salary in 1910 was \$3,000, nearly \$69,000 today's dollars. By 1940, men could earn between \$5,000 up to \$12,000 annually with an additional \$2,500 for umpiring the World Series. This is equivalent to \$82,000 to \$197,000 annually with a \$41,000 bonus in today's dollars. Additionally, umpires were still not fully compensated for travel, clothing, or gear expenses. In 1940, they received an allowance of \$750, which only covered half of their travel expenses (www.sdabu.com). Today, a member of the union of Major League Umpires Association, formed in 1968, makes \$100,000 to \$300,000 per year depending on their experience (Gassko 2007).

As noted earlier, umpires were commonly just spectators of the sport prior to the creation of the umpiring staff. When the umpiring staff was created by the American Association and National League, no formal schooling was required and most training was done by umpiring for minor league games (www.sdabu.com). In 1935, the first umpiring school was created by

George Barr. Forty-five years later, it had become impossible to become a major league umpire without attending a school first (Gassko 2007). Today, professional umpires are required to graduate from either The Jim Evans Umpire Academy or The Harry Wendelstedt Umpire School.

With such requirements and high pay, one would expect stringent evaluations of umpires. In order to do this, the Major League Umpires Association uses baseball game data and statistics. While it is known that game statistics are used in this evaluation, the process and extent to which they are used is closely guarded and has not been published (Adair 2003).

Recent New Developments of Technology in Sports

As technology has evolved, so has the incorporation of technology in sports. The Hawk-Eye system was introduced by the International Tennis Federation in 2005. Hawk-Eye utilizes a system of cameras that maps the path of the ball and point of contact on the court to determine if the line judge made the correct call (Vilines 2010). The International Football Association (FIFA) is currently researching goal-line technology (retrieved from www.soccerway.com). FIFA has tested three different goal-line technologies in three matches and will decide if the technology will be used by July 2, 2012 (retrieved from www.soccerway.com). This sensor technology signals to the umpire when the ball has crossed the goal-line yielding a point for the scoring team (McGrath 2010). With the advancement in the ability to store data, massive datasets are being created in many sports, including baseball. With the ability to take clear pictures, cameras can be used throughout sports stadiums to aide in, and reduce error in, data collection.

Players are able to gather instant feedback on swinging patterns through the use of video analysis. According to John Dever (Lavin 2001), players and coaches can view video of at-bats during a game. This allows players to make adjustments quickly and reduce errors. Furthermore, teams can create databases containing over 400 hours of archived video. This allows players and coaches to not only observe how well they are doing, but observe an opponent's habits as well (Lavin 2001). Having this information can give a batter an advantage if he can predict the type of pitch he might receive in key situations.

Furthermore, Lavin discusses how datasets have been created by scouting officials in Major League Baseball (MLB). Teams are able to research various data and tendencies of opponents. He discusses a unit sold by Recreational Technologies of Olathe, KS, that allows

someone observing a game to tap a screen for each pitch thrown and determine where it crossed home plate and if contact was made. The observer can also attach a radar gun to this system and record the speed of the pitch.

QuesTec

Between 2002 and 2008, QuesTec was utilized to evaluate the performance of umpires in MLB (Adair 2003; Kalk 2009). The system used four cameras placed throughout the stadium and a computer operator. The operator would calibrate the system before each game providing a center point from which to track each pitch. During a game, the QuesTec operator would watch the game on a small screen, placing a line at the top of the belt and the hollow of the back of the knee of each batter for each pitch thrown (Karegeannes 2004). After the game, the operator produced a data CD which contained a pitch table, pitch locations, accuracy chart, and consistency chart which were given to MLB and the umpire the next day (Karegeannes 2004).

QuesTec was a valuable tool in objectively rating home plate umpires and holding them accountable. The Umpire Information System (UIS) program was created to evaluate umpires using the QuesTec system (Adair 2003). With the defined strike zone for each batter, the QuesTec operator sets up a computer program that calls balls and strikes for each pitch and reviews each pitch. He identifies bad tracks—where the computed track is clearly not in accord with other information. On a CD for the plate umpire, the computed ball tracks and umpire ball-strike calls for each pitch are assigned a letter C (for correct) if the umpire and UIS are in agreement, A (for acceptable) if the calls disagree but a change of two inches in the computed trajectory would bring the calls in agreement, and N (for not acceptable) if the calls disagree by greater than two inches (Adair 2003).

According to Hugo Lindgren (2003) of *The New York Times*, “If an umpire’s calls disagree with the computer’s more than 10 percent of the time, his performance will be considered substandard and possibly held against him in future promotion considerations and when lucrative post-season assignments are made.”

One noted weakness of QuesTec is the system’s consistent inability to correctly call certain types of pitches, mostly sliders and curves. Another disadvantage of the system was that the operator was sometimes unable to see the batter because of a coach or bench player blocking

the camera. Finally, shadows often made it difficult to clearly see a pitch using the system (Karegeannes 2004).

From what I can tell, this data was not released publicly and the contract between QuesTec and MLB ended in 2008 with the implementation of PITCHf/x (Kalk 2009).

PITCHf/x

PITCHf/x was created and has been maintained by Sportvision since the MLB playoffs in 2006. The system currently utilizes three cameras in every MLB stadium and collects information on the speed, location, and trajectory of each pitch. This information is collected in real time and is then entered into a database. From this database broadcasters and sports enthusiasts can see the results seconds after the pitch has occurred.

Some of the variables available include pitcher, umpire, batter, stadium, pitch count for each pitch, total number of appearances at bat for each batter, the umpire's decision (ball, strike), as well as the PITCHf/x coordinates recorded from the cameras. The system also classifies the type of pitch as fastball, curveball, etc. based on starting and ending speed, location, and trajectory (Nathan 2010; Garik16 2011).

PITCHf/x technology has created a new way for spectators to watch baseball. Sports enthusiasts who watch a televised baseball game will see a strike zone with a grid, as well as pitching placement and speed for every pitch. This can be seen in Figure 1.1.



Figure 1.1 Television screenshot of baseball game with Pitch Trax, which utilizes PITCHf/x technology to show viewers where the ball crosses home plate (Fast March 2011)

It is with this data that fans can participate in data analysis. Many websites have been created that allow a person to select certain game characteristics and then create a scatter plot of the strike zone, for that game. One popular site is www.brooksbaseball.com, created by Dan Brooks. This site allows the user to select the data, game and pitcher. For the selected game, you can select the plots you want to see, such as balls and strikes, batter stance, or pitch type. One can even select a specific batter or inning from the game. A series of tables and plots are then provided. For example, if we were interested in how left-handed pitchers throw a ball to Derek Jeter, we can obtain the dataset and 6 various plots in order to visually identify trends (Lefkowitz 2009). One such plot is the Strikezone Report, given in Figure 1.2. In this plot we can see the location of various types of pitches and whether they were called strikes or balls. Called strikes and balls are differentiated by the color of the symbol, red for called strikes and green for called balls. Therefore, any green symbols within the strike zone box and any red symbols outside the strike zone box represent a disagreement of calls between PITCHf/x and the home plate umpire. One can conclude from this plot that left-handed pitchers rarely pitch below the strike zone to the catcher's right.

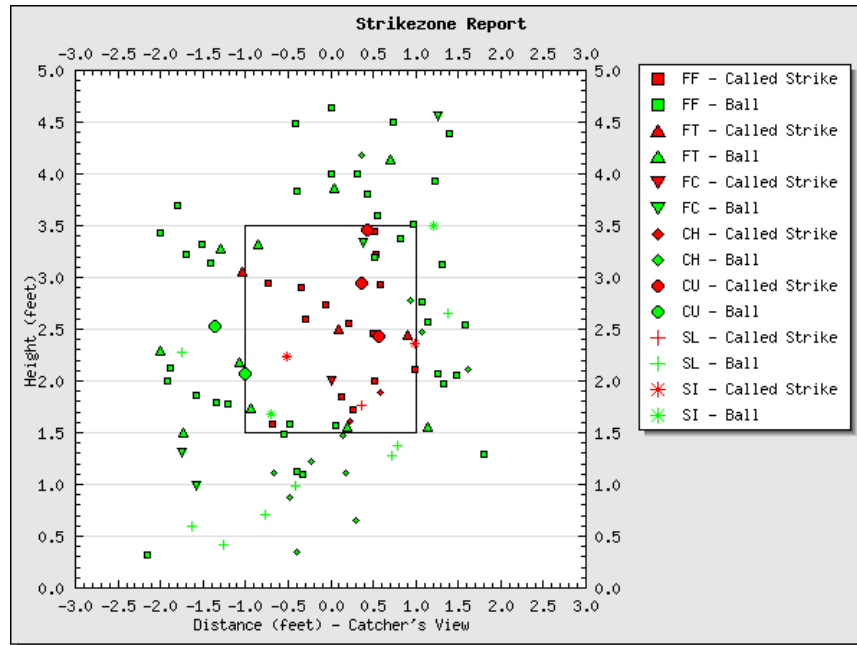


Figure 1.2 Strikezone report for left handed pitches to Derek Jeter (Lefkowitz 2009)

Previous Research

“With advances in technology and mass media coverage of every game, never before has the ball/strike call been more scrutinized and analyzed” (Schlegel 2010). Now that the data is publicly available, baseball enthusiasts are happy to use it.

Josh Kalk (2009), a self-proclaimed physics and math geek, published an article in The Hardball Times listing John Walsh, also of The Hardball Times, as helping pioneer research of the strike zone. Walsh is credited with defining an actual strike zone for left- and right-handed batters (Kalk 2009).

Continuing on with analysis of the strike zone, Kalk investigated strike zones for one umpire, Angel Hernandez, by drawing the strike zone based on a rectangle with the least amount of errors. That is, the called strikes outside the box and called balls inside the box are minimized in Kalk’s diagrams. Kalk also raises the question of the top and bottom of the strike zone and suggests averaging each at-bat for a hitter and using those numbers for the top and bottom of the strike zone for all his at-bats. Further, he suggests normalizing all the data to a league average height. He claims that these corrections would fix any problems with the top and bottom of the strike zone. He credits Walsh as being the first to do this in Walsh’s 2007 The Hardball Times article The Eye of the Umpire in which he discusses the top and bottom of the strike zone.

More work on the strike zone has been conducted by Mike Fast (2011a,b,c) of Baseball Prospectus. Fast (2011a) cites Jonathan Hale's research on umpire strike zones and breaks down the results by umpire and pitcher. He further (2011b) researched how the strike zone might change based on a variety of situations: inning, pitcher's age/experience, pitcher control, home/away team, etc. Fast also notes Walsh and J-Doug Mathewson's research on how an umpire's strike zone changes based on the ball count and other factors. Fast notes that Jonathan Hale, Dave Allen, John Walsh, and J-Doug Mathewson have all observed that the strike zone is bigger in ball-strike counts that favor the hitter and smaller in counts that favor the pitcher (2011a). Fast discusses that a strong correlation exists between the typical pitch location and the horizontal shift in a batter's strike zone. The Catcher Target Theory, a theory that umpires adapt the strike zone based on the location of the catcher's glove, is also discussed in Fast's (2011a) article "The Real Strike Zone."

Three months later, Fast (2011c) followed up with "The Real Strike Zone, Part 2." This article discusses the top and bottom strike zone problems discussed previously. In this article, Fast raises the question of the utility and accuracy of a zone evaluation system based on the unreliable top and bottom of the strike zone obtained from PITCHf/x. In other words, how can a system that does not follow the rulebook be used to evaluate if an umpire follows the rulebook?

Parsons, et al. (2011) have published a study looking at the effect of race of the pitcher and umpire on the percentage of strikes called and found umpires to be slightly racially biased. Obviously, a publication such as this would raise controversy. One criticism is how Parsons, et al. (2011) defined race. According to the study, race is classified based on where a player is born, and "all remaining unclassified players and umpires are classified by visual inspection of pictures found in Internet Searches" (Parsons et al. 2011). Kalk (2009) notes a blogger, Phil Birnbaum, who questions Parsons' results because the number of African American and Latino umpires is small.

Questions of Interest

While the previous research was done fairly recently, some baseball statistics enthusiasts have begun to answer a variety of questions that could not be answered without the recent

additional technology. However, no research clearly answers our questions of interest. We focus on the following:

- 1) Have umpires decreased the proportion of incorrect calls they make since the introduction of PITCHf/x?
- 2) Does the proportion of incorrect calls differ for umpires hired prior to the implementation of technology from those hired after?
- 3) Does the proportion of incorrect calls change at a different rate for umpires hired prior to the implementation of technology from those hired after?

Using PITCHf/x data gathered from Joe Lefkowitz, a computer science major at Stevens Institute of Technology, data gathered from www.retrosheet.org, and additional data gathered on the umpires from *Major League Umpires' Performance, 2007-2010: A Comprehensive Statistical Review*, We will create a dataset which would then allow us to use longitudinal data analysis to answer these questions.

Chapter 2 - Data Collection

Our original aim was to evaluate umpire performance based on:

- 1) The number of years an umpire has been with MLB by placing umpires into one of two groups—pre-technology or post-technology, based on the year of their first major league game. A pre-technology era umpire would be hired prior to 2002, the inaugural year of the Umpire Information System (UIS) being used in umpire evaluation.
- 2) The umpire's age. Placing umpires into one of 3 categories—Baby boomers, Generation X, and Generation Y, for umpires born prior to 1964, between 1965 and 1980, and after 1980 respectively.
- 3) Where the umpire was born. Following the method of Parsons et al. (2011), we aimed to see if a racial differences exist in umpire performance.
- 4) If the umpire was calling pitches during a day or night game.
- 5) The type of stadium the umpire was calling a game in. This would be classified as domed or outdoor.
- 6) The size of the crowd for the game an umpire was calling.
- 7) The proportion of left-handed pitches in a game.
- 8) The proportion of left-handed batters in a game.
- 9) The type of pitch the umpire is calling.

We aimed to collect data from 2008-2011 for analysis. In order to do this, we needed to collect data on three different levels with each coming from multiple data sources. First, we needed to collect data at the pitch level. For each individual pitch, the PITCHf/x system generated a ball or called-strike, based on the coordinates of the ball as it crosses home plate. This would then be compared to the call made by the umpire. We also needed the handedness (left or right) of the pitcher and batter for each pitch. Second, we needed to gather data at the game level. This included indoor/outdoor, day/night, etc. Lastly, we needed to identify umpire information—birth year, birth location, and MLB umpiring debut year.

Pitch Level Data

The first set of data that we needed is data regarding the call of each particular pitch. Not only did we need the call made by the umpire himself, we needed to be able to identify if the call

made by the umpire is correct or incorrect. This brought us to our first major set of data, the PITCHf/x data.

PITCHf/x data is publicly available from www.mlb.com, although in a form that can be difficult to use and manage. For those who are not comfortable navigating the site, there are various generous people who have gathered the data and present it in various formats (e.g. www.brooksbaseball.com, www.pitchfx.texasleaguers.com). PITCHf/x data is available for dates starting in 2007; however, various aspects of data collection were changed between the 2007 and 2008 season, including points of measurement, and many of the first games of 2007 were not available.

We used Joe Litkowitz's website www.joelitkowitz.com to collect the PITCHf/x data. Litkowitz's site provides a point-and-click interface that allows one to gather a set of data from his PITCHf/x database by selecting game conditions. For example, suppose we are interested in evaluating how a left-handed pitcher performs when the bases are loaded and there are two outs in away games. We can select these game features and Litkowitz will provide us with a .csv file, and a variety of graphs.

Initially, we planned to download four data files, one for each year in our analysis. Unfortunately, the datasets created by Litkokwitz's site will only list the first 15,000 pitches for each query (there are about 700,000 pitches thrown each season). This required creating multiple search queries, without repeating information gathered previously. We thought that the easiest way to do this would be to obtain one .csv file for each umpire for each year. Since there are roughly 100 umpires, this would provide us a total of 400 separate files.

Unfortunately, this system only worked in for years 2009-2011. Litkowitz does not provide the umpires for games in the 2008 season. Instead, we chose the data based on home team and visiting team with the intention of linking an umpire to each game later. Luckily, the 2009, 2010, and 2011 season came accompanied with the home plate umpire's name. We obtained the umpire information for 2008 and linked it back to the PITCHf/x data; we describe this process later. The final dataset is a combination of 308 .csv files containing the recorded call of the official, pitch location coordinates from PITCHf/x, speed of the pitch, and a calculation of the type of pitch for every pitch thrown from 2008 through 2011.

To appraise the call made by the umpire, we needed to define the strike zone. Officially, a strike is to be called if any part of the ball falls within the strike zone (www.mlb.com). The

strike zone is in the shape of home plate and extends from the hollow of the knee to the top of the shoulder. As stated earlier in the Previous Research section of Chapter 1, many umpires deviate from this strike zone and the top and bottom of the strike zone set by the PITCHf/x operator model the typical vertical strike zone of umpires rather than the official definition. An excellent 3D illustration of this is provided in Figure 2.1. The standard baseball has a diameter of 3 inches, so we added 1.5” to each side of the regulation 17” wide home plate to account for the system measuring the center of the ball. This sets the strike zone sides at 10” to the left and right of the center of home plate. In order to assess the location of the pitch, we used the variables listed in Table 2.1 of the PITCHf/x dataset. Table 2.2 includes the variables used to assess the umpire’s call.



Figure 2.1 A pictorial representation of the typical strike zone (retrieved from Strike Zone en.JPG from Wikipedia).

Table 2.1 Variables used to assess location of the ball as it crosses home plate (*Indicates a variable we created)

<i>Variable Name</i>	<i>Description</i>	<i>Possible values</i>
Sz_top	The distance in feet from the ground to the top of the strike zone. Via video, the operator sets a line at the batter's belt to which the system adds 4 inches to the top for each pitch. This allows the strike zone to vary not only by player, but by pitch as well.	Positive real numbers carried to 2 decimal places
Sz_bot	The distance in feet to the ground to the rulebook defined bottom of the strike zone. The operator places a line at the hollow of the knee.	Positive real numbers carried to 2 decimal places
Px	The left/right distance, in feet, of the pitch from the middle of the plate as it crossed home plate. The PITCHf/x coordinate system is oriented to the catcher's/umpires perspective, with distance to the right being positive and the distance to the left being negative	Negative and Positive real numbers carried to 2 decimal places
Pz	The vertical distance, in feet of the pitch from the ground as it crossed home plate.	Positive real numbers carried to 2 decimal places
Sz_left*	The left most point a ball shall be ruled a strike as defined by www.mlb.com. This value corresponds to the left side of home plate which is 17 inches wide. We add 1.5" to take into account, that the PITCHf/x system measures the center of the ball and a ball is 3"	.8333 feet
Sz_right*	The right most point a ball shall be ruled a strike as defined by www.mlb.com. This value corresponds to the right most side of home plate. Again, we add 1.5" to take into account that we are measuring the center of the ball.	-.8333 feet

Table 2.2 Variables to assess the umpire's call (*Indicates a variable we created)

<i>Variable Name</i>	<i>Description</i>	<i>Value assigned</i>
Home Umpire	The first and last name of the umpire. Note: PITCHf/x did not give this value in 2008.	
Result.Type	This is a dummy variable given to the corresponding Pitch.Result. This call is based on play and the home plate umpire's ruling. (If Result.Type = X, the pitch was omitted from further analysis)	If a pitch resulted in play: X If the pitch was called a strike: S If the pitch is called a ball: B
Ump.call*	This is a simplified binary response indicating if the umpire's call is a ball or strike	0 if a strike 1 if a ball
Ball.call*	This is a simplified binary response indicating if, based on the Px and Pz coordinates, the ball is a strike or a ball	0 if a strike 1 if a ball
Error.call*	This is a binary variable indicating if Ump.Call and Ball.Call agree. If Ump.Call \neq Ball.Call, then an error has been made (This variable is created after all in-play pitches are removed).	0 if no error is made 1 if an error is made

Using the above variables, we can measure the proportion of errors each umpire makes over a given period of time. Initially, we were interested in measuring the proportion of incorrect calls during a given game and evaluating the covariates mentioned at the beginning of Chapter 2. After further consideration, we feel that this makes little sense; measurable change is unlikely to be seen on a game-by-game basis and we wanted to avoid modeling noise. Therefore, we decided to measure the proportion of incorrect calls within the periods defined in Table 2.3. We selected these intervals to have approximately the same number of games in each time period. We code the proportion of incorrect calls `prop_incorrect` in the R code in Appendix A.

Table 2.3 Time periods for which the proportion of incorrect calls is summarized

<i>Year</i> 2008	Period 1: April 1—May 31	Period 2: June 1—July 31	Period 3: August 1—September 30
2009	Period 4: April 5—May 31	Period 5: June 1—July 31	Period 6: August 1—October 6
2010	Period 7: April 4—May 31	Period 8: June 1—July 31	Period 9: August 1—October 3
2011	Period 10: March 31—May 31	Period 11: June 1—July 31	Period 12: August 1—September 28

Game Level Data

Much of the non-umpire information that we wanted to collect came from two sources. First, the PITCHf/x database we created using Litkowitz’s website gave pitcher handedness, batter handedness, and pitch type. The remaining game level data were gathered from www.retrosheet.org. Most importantly, for linking the retrosheet data to the PITCHf/x data, we needed the Park ID, home plate umpire’s name, and the home/visiting team’s game number. This information is described in Table 2.4.

Table 2.4 Variables collected from www.retrosheet.org

<i>Variable</i>	<i>Description</i>	<i>Values</i>
Day/Night	An indicator of whether the game is considered a day or night game	Day/Night
Park ID	A unique 5 digit identification for each baseball park.	Eg. Kauffman Stadium’s Park ID is KAN06
Attendance	An estimate of the number of people in attendance of the game	Positive integer values
Home Plate Umpire Name	The first and last name of the home plate umpire. This is the value we will use for the 2008 data, where we are missing the home plate umpire.	
Home/Visiting Game #	A numeric value indicating how many previous games the home (visiting) team has played.	Positive integer values

In order to correctly match the data from www.retrosheet.org to our PITCHf/x data, we utilize the merge function in R. For years 2009 — 2011, we matched the datasets on umpire, month, day, and year. For the 2008 dataset, we matched on month, day, year, and game number. This was done to insure that the proper umpire is placed with the proper game. By merging on game number, we eliminated the possibility of assigning the wrong umpire the wrong game on a double-header day.

Umpire Level Data

In order to make comparisons based on the umpire’s age or experience we gathered information from Andrew Goldblatt’s *Major League Umpires’ Performance, 2007-2010: A Comprehensive Statistical Review*. We created indicator variables for the generation in which the umpire was born and the era of technology in which the umpire made his MLB debut. This information is found in Table 2.5.

Table 2.5 Information on the birth year and first occurrence as MLB umpire (* Indicates a variable we created)

<i>Variable</i>	<i>Description</i>	<i>Values</i>	
Birth.Year	The year the umpire was born. We used this to create the Gen variable.		
Gen*	An indicator if umpire is born in the Baby Boomer era, the Generation X era, or the Generation Y era	Gen	Count
		Baby-Boom	39
		Gen X	50
		Gen Y	6
First Occurrence as MLB umpire	The year the umpire was first called to umpire a MLB game.		
Tech*	An indicator if the umpire was first called to umpire prior to the implementation of technology in his evaluation	Tech	Count
		Pre	19
		Post	76

Discussion

As previously stated, initially we were interested in looking at a game-by-game analysis; so we collected game varying covariates. However, we feel that a more realistic analysis is achievable by looking at a period-by-period analysis for a few reasons. First, we do not believe an umpire receives evaluation after each game. Second, if an analysis were done at the game-by-game level, we feel we would be modeling noise rather than a specific trend. Therefore, after collecting this information, we abandoned our game-by-game analysis and the associated covariates.

Furthermore, after reading much of the criticism of the Parsons et al. (2011) article and considering the methods they used to determine race, we believe there is not an adequate amount of diversity in race of umpires to confidently make inference. For this reason, we chose not to look at umpire race as a factor.

Lastly, as shown in Table 2.6, we do not have the necessary number of observations in each generation by era cell, to make sound inference. Therefore, we dismissed the generation in which the umpire was born as a predictor and used technology era of first MLB debut because it is more relevant to our questions of interest.

Table 2.6 Contingency table of era by generation frequencies

	<i>Baby Boomers</i>	<i>Generation X</i>	<i>Generation Y</i>	<i>Row Total</i>
<i>Post-Technology</i>	0	20	4	24
<i>Pre-Technology</i>	41	29	0	70
<i>Column Total</i>	41	49	4	94

Process of Data Collection

Statistical Software

In order to clean the data and develop our final dataset we used R 2.14.7. Following is an outline of the procedure we used to create the dataset in R. The complete R script can be found in Appendix A.

2008 Data

First, we gathered every pitch in the 2008 season by selecting one home team and fifteen away teams in the PITCHf/x tool query options at www.joelefkowitz.com. This created a .csv file with less than the maximum of 15,000 pitches. This process was repeated for the remaining fifteen away teams. This yielded two .csv files for each of the 30 teams for a total of 60 separate files.

Next, because we originally planned to do a game-by-game analysis we wanted to separate each game. We created the first R script which reads in all 60 .csv files, and then separated them into the individual games. Then R exports 2,366 .csv files (one for each game).

Third, we created a loop to read in each of the 2,366 .csv files one at a time. We removed any observations in which a called-strike or ball was not given in the Pitch.Result, as well as any observations where $sz_top = 0$; this is clearly an operator error. For each pitch, we assessed the location of the ball as it crossed home plate. If the pitch had a $-0.8333 \leq Px \leq 0.8333$, and a $sz_top \leq Pz \leq sz_bot$, it is classified as a strike under the variable ball.call (refer to Table 2.1 for definitions). If it fails to meet either of these, it is coded as a ball. The ump.call variable was created to compare the umpires call to ball.call.

To define variables for the 12 time points defined in Table 2.3, we created a group variable to distinguish between the first third, middle third, or last third of a season. This will be used in a third R script, described later. We also take a count of incorrect calls and a count of balls or called-strikes for each game.

Next, manipulation of the game id variable in the PITCHf/x data was needed in order to compare it to game information in the Retrosheet database. At this point, the retrosheet data was brought in. Again, the date was modified allowing us to match information from the retrosheet dataset to the PITCHf/x dataset, specifically the home plate umpire.

Now, for each game the dataset contains in a single line the game ID information, the counts described above as well as the game log information gathered from retrosheet.

Finally, in our third R script we read in the datafile containing 2,366 lines and split the data based on the time periods. We calculated the proportion of incorrect calls for each time period for each umpire. We exported this data to another .csv file, which will be used in SAS.

2009, 2010, 2011 Data

As stated, for the remaining years, the home plate umpire is recorded in the PITCHf/x datasets. Because of this, we modified our code for years 2009, 2010, and 2011.

First, we gathered every pitch seen by a certain umpire in the respective season by selecting that umpire in the PITCHf/x tool query options at www.joelefkowitz.com. This created a .csv file with less than the maximum of 15,000 pitches. This process is repeated for each of the 83 umpires. This yielded 83 .csv files for a single year for a total of 249 .csv files for the 2009, 2010, and 2011 seasons.

Next, we created a loop to read in one .csv file at a time. We removed any observations in which a called-strike or ball was not given in the Pitch.Result, as well as any observations where $sz_top = 0$. The location of the ball as it crossed home plate was assessed as in 2008 and was classified accordingly. The process continued in a similar fashion to 2008 with the exception of matching the PITCHf/x data and retrosheet data. For those years, we still matched on date, but we also matched on umpire.

In summary, we started with approximately 2,800,000 pitches and extracted the 9,363 games from 2008—2011. We took that information and classified it into the 12 time periods (3 time periods per year for 4 years) for each umpire giving a total of 928 datalines in our final dataset. Graphically, this data is seen in Figure 2.2. In this plot, we have side-by-side spaghetti plots where we separated the data into the pre- and post-technology era umpires. Each black line represents the observed proportion of incorrect calls for a single umpire coming from his respective technology era. Each line connects a maximum of 12 points; each point represents the proportion of incorrect calls in each time period.

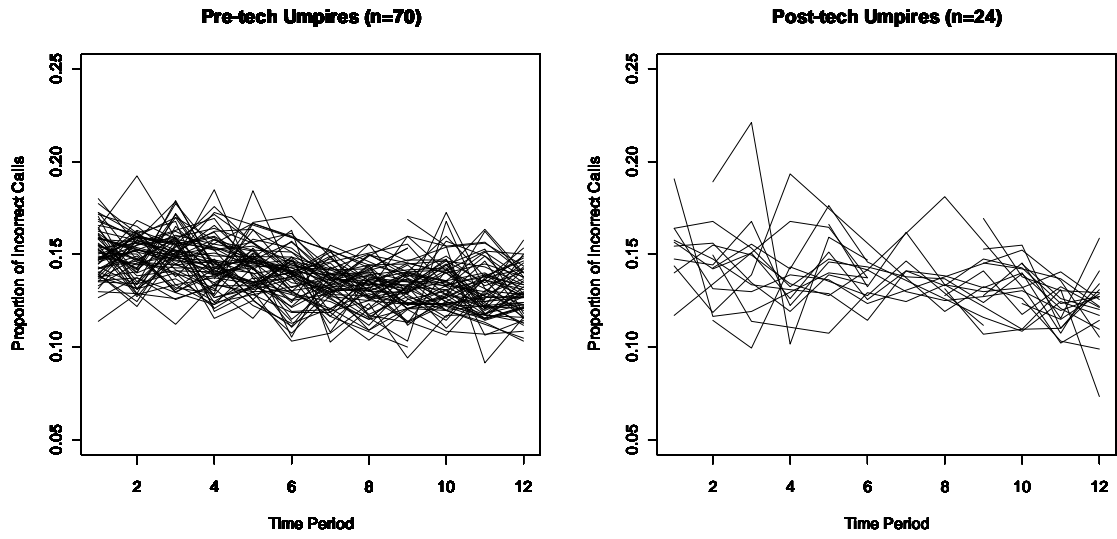


Figure 2.2 Spaghetti plot of incorrect calls by time period separated by umpires MLB debut era (Pre- and Post-technology implementation in evaluation)

Chapter 3 - The Model

To evaluate home plate umpire performance over time we use longitudinal data analysis. We will utilize a random coefficients model.

The Random Coefficients Model

The random coefficients model allows us to assume a model with umpire-specific trajectories. That is, each umpire has his own intercept and own slope that determine his own inherent trend. We also have two sources of variation: 1) within-umpire variation—variation due to random error and 2) among-umpire variation—inherent trajectories are “high” or “low” with different steepness over time across umpires, suggesting that the regression parameters vary across umpires.

We will develop this model in two stages, the individual model and the population model. The individual model is a model unique to a specific umpire of interest. The population model is a model that describes the linear trend of the average umpire. First, we describe the model at the level of the i^{th} umpire. Here each model has the form of a regression model unique to the i^{th} umpire. The model for umpire i , $i = 1, \dots, m$, is

Model I

$$Y_{ij} = \beta_{0i} + \beta_{1i}t_{ij} + e_{ij}, \quad j = 1, \dots, n_i$$

where Y_{ij} is the proportion of incorrect calls made by umpire i at time period j , t_{ij} is the j^{th} time period for umpire i . n_i is the number of time points for the i^{th} umpire (the maximum value of n_i is 12). In this model β_{0i} is the umpire-specific intercept for umpire i , β_{1i} is the umpire-specific slope for umpire i , and e_{ij} is the within-umpire random deviation with mean 0 that represents the deviation introduced solely by sources within an umpire. Because the proportion of incorrect calls made by umpires across all time periods has a mean of 0.1383 and is centered away from zero with a standard deviation of 0.01613, we feel that we can fit a model that assumes Y_{ij} to be normal.

The regression parameter vector for each umpire is

$$\boldsymbol{\beta}_i = \begin{pmatrix} \beta_{0i} \\ \beta_{1i} \end{pmatrix}.$$

We can write Model I more concisely by defining

$$\mathbf{Y}_i = (Y_1 \quad Y_2 \quad \cdots \quad Y_{n_i})',$$

$$\mathbf{e}_i = (e_1 \quad e_2 \quad \cdots \quad e_{n_i})'$$

and

$$\mathbf{Z}_i = \begin{pmatrix} 1 & t_{i1} \\ 1 & t_{i2} \\ \vdots & \vdots \\ 1 & t_{in_i} \end{pmatrix}.$$

Model I can now be written as

Model II

$$\mathbf{Y}_i = \mathbf{Z}_i \boldsymbol{\beta}_i + \mathbf{e}_i, \quad i = 1, \dots, m.$$

Model II only tells part of the story; it describes what happens at the level of an individual umpire, and includes explicit mention (through e_{ij}) of within-umpire variation. However, it does not model among-umpire variation. Visual inspection leads us to recognize that inherent trends differ across umpires. This is illustrated in Figure 2.2, where each individual trajectory differs slightly in slope and/or intercept.

In order to consider the population model, we treat each observed umpire as arising from a hypothetical population of all professional umpires. We will allow each umpire in the population to have his own intercept and slope describing the change in proportion of incorrect calls over time. We may think of this population of slopes and intercepts as a population of random vectors $\boldsymbol{\beta}_i$, one for each umpire. This defines a unique random vector for each umpire distinguishing his trajectory.

This way of thinking suggests a model for the population as follows. Define

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}, \tag{3.1}$$

where $\boldsymbol{\beta}$ is the mean vector of the population of all $\boldsymbol{\beta}_i$ with β_0 and β_1 representing the mean values of intercept and slope respectively. Then we can write

$$\boldsymbol{\beta}_i = \boldsymbol{\beta} + \mathbf{b}_i \quad (3.2)$$

where

$$\mathbf{b}_i = \begin{pmatrix} b_{0i} \\ b_{1i} \end{pmatrix}$$

Here, \mathbf{b}_i is a vector of random effects describing how the intercept and slope for the i^{th} umpire deviate from the mean values. Thus, (3.2) is regression-type model. The vectors \mathbf{b}_i are assumed to have mean $\mathbf{0}$ and some covariance matrix that describes the nature of the variation—how intercepts and slopes vary among umpires.

Thus, while the individual umpire model summarizes how things vary within an umpire, this model characterizes the variation among umpires, representing the population through average intercepts and slopes. Combining Model I and equation (3.2) together gives a complete description of what we believe about each umpire and the population of umpires acknowledging the two sources of variation.

We can substitute equation (3.2) into Model I to obtain

Model III

$$Y_{ij} = (\beta_0 + b_{0i}) + (\beta_1 + b_{1i})t_{ij} + e_{ij}.$$

This shows what we are assuming: each umpire has an intercept and slope that varies about the “typical” or mean intercept and slope. Model III models the response for the i^{th} umpire at the j^{th} observation.

To assess our questions of interest, we need to consider additional information for our longitudinal model, namely the era in which each official made his MLB umpiring debut. We refer to these as pre-tech and post-tech for umpires hired prior to the year 2002 and after 2002, respectively, (2002 being the year technology was used in umpire evaluation). When we include technology era, we essentially have two β_0 's and two β_1 's, i.e. different intercept and slope parameters for each tech group. That is, we have the following two models:

Model IV

$$Y_{ij} = (\beta_{0,Pre} + b_{0i}) + (\beta_{1,Pre} + b_{1i})t_{ij} + e_{ij},$$

to model the proportion of incorrect calls made by the i^{th} umpire at the j^{th} observation when the umpire made his MLB debut prior to 2002, and

Model V

$$Y_{ij} = (\beta_{0,Post} + b_{0i}) + (\beta_{1,Post} + b_{1i})t_{ij} + e_{ij},$$

to model the proportion of incorrect calls made by the i^{th} umpire at the j^{th} observation when the umpire made his MLB debut during or after 2002. We can write these two scalar models in matrix form as follows:

Model VI

$$Y_i = Z_i A_i \beta + Z_i b_i + e_i$$

with

$$Y_i = (Y_{i1} \ Y_{i2} \ \dots \ Y_{in_i})'$$

$$\beta = \begin{pmatrix} \beta_{0,Post} \\ \beta_{1,Post} \\ \beta_{0,Pre} \\ \beta_{1,Pre} \end{pmatrix}$$

$$b_i = \begin{pmatrix} b_{0i} \\ b_{1i} \end{pmatrix}$$

$$Z_i = \begin{pmatrix} 1 & t_{i1} \\ 1 & t_{i2} \\ \vdots & \vdots \\ 1 & t_{in_i} \end{pmatrix}$$

and

$$A_i = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

if the umpire is hired during the post-tech era or

$$A_i = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

if the umpire is hired during the pre-tech era.

To obtain Model V from Model VI for a post-tech hired umpire:

$$\begin{pmatrix} Y_{i1} \\ Y_{i2} \\ \vdots \\ Y_{in_i} \end{pmatrix} = \begin{pmatrix} 1 & t_{i1} \\ 1 & t_{i2} \\ \vdots & \vdots \\ 1 & t_{in_i} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \beta_{0,Post} \\ \beta_{1,Post} \\ \beta_{0,Pre} \\ \beta_{1,Pre} \end{pmatrix} + \begin{pmatrix} 1 & t_{i1} \\ 1 & t_{i2} \\ \vdots & \vdots \\ 1 & t_{in_i} \end{pmatrix} \begin{pmatrix} b_{0i} \\ b_{1i} \end{pmatrix} + \begin{pmatrix} e_{i1} \\ e_{i2} \\ \vdots \\ e_{in_i} \end{pmatrix}.$$

After matrix calculations,

$$\begin{aligned} Y_{ij} &= 1 * 1 * \beta_{0,Post} + t_{ij} * 1 * 1 * \beta_{1,Post} + 1 * b_{0i} + t_{ij} b_{1i} + e_{ij} \\ &= \beta_{0,Post} + \beta_{1,Post} t_{ij} + b_{0i} + b_{1i} t_{ij} + e_{ij}. \end{aligned}$$

After re-arranging our paramters we have

$$Y_{ij} = (\beta_{0,Post} + b_{0i}) + (\beta_{1,Post} + b_{1i})t_{ij} + e_{ij}$$

WITHIN-UNIT VARIATION

In Model II and Model VI the within-unit random vector \mathbf{e}_i has mean zero and represents the deviations introduced solely by sources within an umpire. This includes measurement error, random fluctuations, or both. We make the standard assumption that \mathbf{e}_i and \mathbf{b}_i are independent.

To characterize within-umpire variation and correlation due to within-umpire sources, we specify a covariance structure model for $var(\mathbf{e}_i)$. In general, write $\mathbf{R}_i = var(\mathbf{e}_i)$, where \mathbf{R}_i is an $(n_i \times n_i)$ covariance matrix, a maximum of (12×12) . We will investigate various structures for \mathbf{R}_i later in this chapter.

For a given response for a single umpire i at time point t_{ij} , if we assume a normal distribution reasonably represents the population of possible responses from this umpire at this time, we would then assume that each e_{ij} is normally distributed as well. This implies that $\mathbf{e}_i \sim \mathcal{N}_{n_i}(\mathbf{0}, \mathbf{R}_i)$.

AMONG UNIT-VARIATION

In the population model, Model VI, the random effects \mathbf{b}_i represent variation among umpires. $Var(\mathbf{b}_i)$ characterizes this variation.

Intercepts and slopes for umpires may tend to be large or small together. For example, umpires with steeper negative slopes may tend to “start out” with higher proportions of incorrect calls for the first time period. Alternatively, small intercepts may tend to happen with small slopes. In either case, this suggests that $var(\mathbf{b}_i)$ is not a diagonal matrix. Rather, we expect there to be some correlation between intercepts and slopes. Formally, we assume that $var(\mathbf{b}_i) = \mathbf{D}$ for some unstructured covariance matrix \mathbf{D} . We assume that the populations of intercepts and slopes are normally distributed; $\mathbf{b}_i \sim \mathcal{N}_2(\mathbf{0}, \mathbf{D})$.

Choice of covariance structure

One of our main goals is to identify if the proportion of incorrect calls made by umpires hired before technology was implemented in performance evaluations is significantly different from the proportion of incorrect calls made by the umpires hired after implementation. Because

of this, we believe it is reasonable to consider fitting a model in which the covariance matrices $var(\mathbf{e}_i) = \mathbf{R}_i$ and $var(\mathbf{b}_i) = \mathbf{D}$ are different for each group (pre-technology, post-technology).

While considering \mathbf{R}_i and \mathbf{D} being different for each group, we further consider the structure of \mathbf{R}_i . It seems reasonable to assume that the proportion of incorrect calls will be more highly correlated for time periods closer together. With our time points being the first, second, and last third of a single season, it seems reasonable to assume that the highest correlation of proportion of incorrect calls would exist within one or two time periods. That is, an umpire makes fairly similar calls within a year's time. Further, it seems reasonable to assume that the correlation between proportions of incorrect calls after two time periods would be very small and can be modeled as having no correlation. With these assumptions, we hypothesize that a model where \mathbf{R}_i has a two-dependent structure with different σ_{pre}^2 and σ_{post}^2 and different \mathbf{D}_{pre} and \mathbf{D}_{post} is reasonable.

We will consider this model as well as the models in which the correlational structure of the covariance matrix \mathbf{R}_i is diagonal, autoregressive of order 1, one-dependent, and compound symmetric. A diagonal structure would be appropriate if the proportion of incorrect calls across time periods were not correlated. If the proportion of incorrect calls were measured in time periods that were equally spaced and followed an exponential decrease in correlation over time, an autoregressive structure would be appropriate. A one-dependent structure would imply that the proportion of incorrect calls at time points t_{ij} and t_{ik} were correlated only when t_{ij} is the adjacent time point to t_{ik} and all other proportions of incorrect calls measured at non-adjacent time points were uncorrelated to those measured at t_{ij} . A compound symmetric structure would suggest that the proportion of incorrect calls for all time points are equally correlated; this is the least likely to be the case for our data.

We will investigate four cases for each correlational structure listed above. The first case is when the covariance matrix \mathbf{R}_i is the same for each group (pre-technology, post-technology) and the covariance matrix $\text{var}(\mathbf{b}_i) = \mathbf{D}$ is also the same for each group. The second case is when the covariance matrix \mathbf{R}_i is different for each group and the covariance matrix $\text{var}(\mathbf{b}_i) = \mathbf{D}$ is the same for each group. The third case is when the covariance matrix \mathbf{R}_i is the same for each group and the covariance matrix $\text{var}(\mathbf{b}_i) = \mathbf{D}$ is different for each group. Finally, we consider the case when the covariance matrix \mathbf{R}_i is different for each group and the covariance matrix $\text{var}(\mathbf{b}_i) = \mathbf{D}$ is also different for each group.

We will also investigate the need for a random slope and intercept model. We compare the model described above to a more parsimonious linear mixed model in which the slope is fixed and an even more simple, general linear model for longitudinal data where both the slope and intercept are fixed.

Chapter 4 - The Analysis

Using the MIXED procedure in SAS 9.2 we coded and analyzed our models. A description of how we utilized MIXED to estimate the hypothesized random coefficients model—a model having separate two-dependent R_i matrices, and different unstructured D matrices for pre- and post-tech umpires, is given. Other models are estimated with only slight modifications. We then provide a detailed analysis of the model we deem most appropriate. Lastly, we investigate and compare results obtained from our best random coefficients model with a model in which we assume the slope is fixed and with a general linear model.

We used the following SAS code to fit our random coefficients model using maximum likelihood and the two-dependent covariance structure for R_i . Note that the SAS variable `coded` represents the time periods described in Table 2.3.

```
PROC MIXED DATA=masters METHOD=ml;
  CLASS umpire tech;
  MODEL prop_incorrect = tech tech*coded/NOINT SOLUTION CHISQ
DDFM=satterth;
  RANDOM intercept coded/TYPE=un SUBJECT=umpire GROUP=tech;* G GCORR V
VCORR;
  REPEATED / TYPE=toep(3) SUBJECT=umpire GROUP=tech;* R RCORR;
RUN;
```

In the `RANDOM` statement we specify a random coefficients model by including both the intercept and slope (`coded`). If the R_i matrix is in a form other than diagonal, a `REPEATED` statement is used. In the `RANDOM` statement, `TYPE=un`, specifies an unstructured matrix for D . In the `REPEATED` statement, `TYPE=toep(3)`, specifies a two-dependent correlation structure for R_i . The `SUBJECT` option identifies the experimental unit. The `GROUP` option is used to communicate to SAS that we want different covariance parameters for each technology era.

Choosing the Best Random Effects Model

We use Akaike's Information Criterion and the Bayesian Information Criterion to aid in model selection. The AIC and BIC are provided in Table 4.1 for each model with each covariance structure described in Chapter 3.

Table 4.1 AIC and BIC for each model

Model #	R_i		$var(\mathbf{b}_i) = \mathbf{D}$ Unstructured	AIC	BIC
0	Two-Dependent	Different	Different	-5366.2	-5327.6
1	Diagonal	Same	Same	-5327.2	-5309.2
2		Different	Same	-5373.1	-5352.5
3		Same	Different	-5326	-5300.6
4		Different	Different	-5373	-5347.3
5	1 st order autoregressive	Same	Same	-5325.4	-5304.9
6		Different	Same	-5370	-5344.3
7		Same	Different	-5324.3	-5296
8		Different	Different	-5369.9	-5339
9	One-dependent	Same	Same	-5325.5	-5304.9
10		Different	Same	-5370.1	-5344.3
11		Same	Different	-5324.4	-5296
12		Different	Different	-5370	-5339.1
13	Two-dependent	Same	Same	-5325.9	-5302.8
14		Different	Same	-5368.7	-5337.8
15		Same	Different	-5325	-5294.1
16		Different	Different	-5366.2	-5327.6
17	Compound Symmetric	Same	Same	-5325.2	-5304.6
18		Different	Same	-5373.3	-5350.1
19		Same	Different	INFINITE LIKELIHOOD	
20		Different	Different	DID NOT CONVERGE	

Note: Different is used when the correlation matrix R_i or D for pre-tech umpires does not equal R_i or D for post-tech umpires.

Based on the AIC and BIC, we see that our initial hypothesized model is not the recommended model. It seems that a two-dependent correlation structure with different R_i and different D may be overcomplicated and thus undesirable. This is because smaller AIC and BIC values are more desirable and we consider a decrease of 2 points in the AIC to indicate substantial improvement to model fit (Burnham & Anderson 2002).

Models with the same R_i for both tech groups yield noticeably higher AIC and BIC values than those where R_i is different regardless of D and the structure of R_i . Within each correlation structure for R_i , the AIC are nearly the same when R_i are assumed to be equal for pre- and post-tech era umpires, only differing at most by 1.2 units; the BIC values are very similar as well,

differing by at most 8.9 units. Models with differing \mathbf{D} show no significant improvement in the AIC and BIC values when we hold the \mathbf{R}_i constant. Thus, it seems reasonable to use a model where the variation of the random components is assumed not to differ based on the umpire's debut era. The autoregressive, one-dependent, and two-dependent structures perform similarly; however, none perform significantly better than the diagonal structure to warrant using a more complicated structure. Based on the BIC, a diagonal \mathbf{R}_i that differs for pre- and post-tech era umpires and a common unstructured \mathbf{D} is most recommended. This is in agreement with Davidian's (2005, p. 328) comment that the correlation structure that is considered extensively and almost exclusively in much of literature has a diagonal structure for \mathbf{R}_i . For these reasons we continue our analysis with model 2 where we assume the off-diagonal elements of \mathbf{R}_i are zero.

Analysis of Model 2

Diagnostics Check

As with any model estimation, we need to check that our model assumptions are adequately met. Figure 4.1 is a plot of residuals versus predicted values. There does not appear to be any significant pattern. Figure 4.2 is a normal probability plot of the residuals. We see that our data are symmetrically distributed with possibly heavy tails. This would suggest a transformation of our response. However, when the response was log and square root transformed no improvements in the residual plots were found.

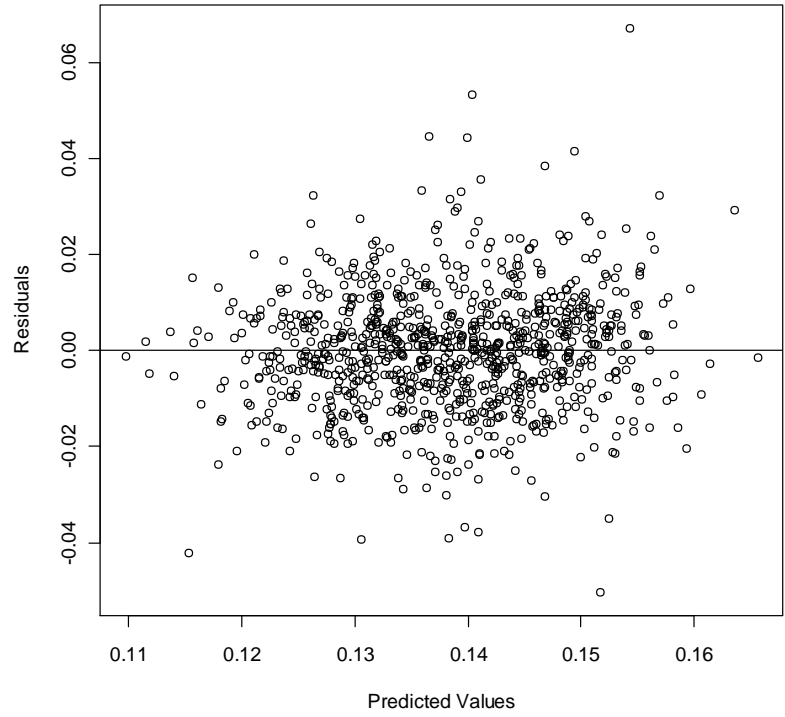


Figure 4.1 Residuals vs. Predicted Values for Model 2

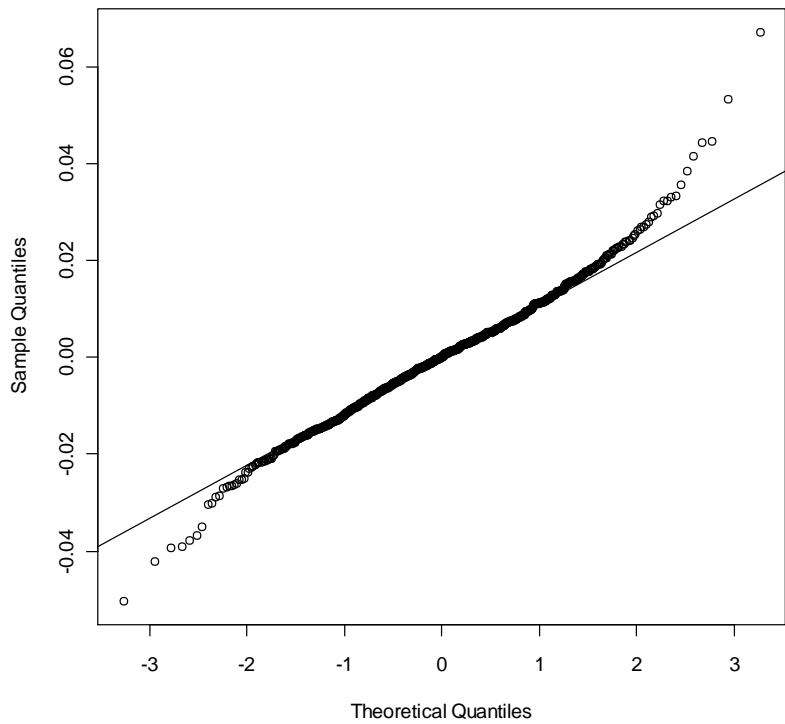


Figure 4.2 Normal Probability Plot for Model 2

Interpreting the SAS output

The parameters estimated by SAS are in Table 4.2.

Table 4.2 Covariance Parameter Estimates and Solution for Fixed Effects for Model 2

<i>Cov Parm</i>	<i>Subject</i>	<i>Group</i>	<i>Estimate</i>	<i>Standard Error</i>	<i>Z Value</i>	<i>Pr > Z </i>	<i>Alpha</i>	<i>Lower</i>	<i>Upper</i>
$\widehat{var}(b_0)$	Umpire		0.000043	0.000015	2.85	0.0022	0.05	0.000024	0.000098
$\widehat{cov}(b_0, b_1)$	Umpire		6.302E-8	0
$\widehat{var}(b_1)$	Umpire		4.28E-57
Residual	Umpire	tech	0.000303	0.000035	8.63	<.0001	0.05	0.000245	0.000386
Residual	Umpire	Post- tec							
Residual	Umpire	tech	0.000133	7.188E-6	18.57	<.0001	0.05	0.000120	0.000149
Residual	Umpire	Pre- tec							

<i>Effect</i>	<i>Tech</i>	<i>Estimate</i>	<i>Standard Error</i>	<i>DF</i>	<i>t Value</i>	<i>Pr > t </i>	<i>Lower</i>	<i>Upper</i>
tech	Post-tec	0.1537	0.003572	150	43.03	<.0001	0.1467	0.1608
tech	Pre-tec	0.1526	0.001173	83.1	130.11	<.0001	0.1503	0.1550
Coded*tech	Post-tec	-0.00260	0.000410	178	-6.34	<.0001	-0.003	-0.0018
Coded*tech	Pre-tec	-0.00210	0.000122	705	-17.23	<.0001	-0.002	-0.0019

The covariance parameter estimates are:

$$\widehat{var}(b_i) = \widehat{D} = \begin{bmatrix} 0.000043 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\hat{\sigma}_{Post-tec}^2 = 0.000303$$

and

$$\hat{\sigma}_{Pre-tec}^2 = 0.000133.$$

The estimated fixed effects $\widehat{\beta}$ are:

$$\widehat{\beta} = \begin{pmatrix} \hat{\beta}_{0,Post-tec} \\ \hat{\beta}_{1,Post-tec} \\ \hat{\beta}_{0,Pre-tec} \\ \hat{\beta}_{1,Pre-tec} \end{pmatrix} = \begin{pmatrix} 0.1537 \\ -0.00260 \\ 0.1526 \\ -0.00210 \end{pmatrix}.$$

Note that the estimates for D_{12} and D_{22} are going to zero. The numerical algorithms used to calculate these estimates and standard errors are failing, thus SAS sets them to zero. We also obtained predictions of the random effects for the individual umpires. These estimates are best

linear unbiased predictors. These predictors are given in Table 4.3 for 3 umpires. Again we note the issues with the estimation of the standard error for the random slope coefficients.

Table 4.3 A portion of the Solutions for Random Effects for Model 2

<i>Effect</i>	<i>Umpire</i>	<i>Estimate</i>	<i>Std Err Pred</i>	<i>DF</i>	<i>t Value</i>	<i>Pr > t </i>
Intercept	Adrian Johnson	-0.00014	0.004147	928	-0.03	0.9734
Coded	Adrian Johnson	-1.86E-7	0	924	-Infy	<.0001
Intercept	Bob Davidson	0.007661	0.003028	928	2.53	0.0116
Coded	Bob Davidson	0.000011	0	924	Infy	<.0001
Intercept	Alfonso Marquez	0.004631	0.003234	928	1.43	0.1525
Coded	Alfonso Marquez	7.057E-6	0	924	Infy	<.0001

We can now estimate the proportion of incorrect calls for each umpire using the random coefficients model. We will use Adrian Johnson, a post-technology hired umpire, for an example. The components of our estimated model are:

Model VII

$$\mathbf{Z}_{Adrian\ Johnson} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ \vdots & \vdots \\ 1 & 12 \end{pmatrix}$$

$$\hat{\boldsymbol{\beta}} = \begin{pmatrix} \hat{\beta}_{0,Post-tech} \\ \hat{\beta}_{1,Post-tech} \\ \hat{\beta}_{0,Pre-tech} \\ \hat{\beta}_{1,Pre-tech} \end{pmatrix} = \begin{pmatrix} 0.1537 \\ -0.00260 \\ 0.1526 \\ -0.00210 \end{pmatrix}$$

$$\mathbf{A}_{Adrian\ Johnson} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\hat{\mathbf{b}}_{Adrian\ Johnson} = \begin{pmatrix} \hat{b}_{0,Adrian\ Johnson} \\ \hat{b}_{1,Adrian\ Johnson} \end{pmatrix} = \begin{pmatrix} -0.00014 \\ 0 \end{pmatrix}$$

Yielding an estimate of $\hat{\mathbf{Y}}_{Adrian\ Johnson}$ which we can compare to the actual $\mathbf{Y}_{Adrian\ Johnson}$.

Table 4.4 Estimated and observed proportions of incorrect calls for umpire Adrian Johnson

Time Period j	$\hat{Y}_{Adrian Johnson,j}$	$Y_{Adrian Johnson,j}$
1	0.1510	0.1477
2	0.1484	0.1442
3	0.1458	0.1680
4	0.1432	0.1222
5	0.1406	0.1457
6	0.1380	0.1431
7	0.1354	0.1382
8	0.1328	0.1370
9	0.1302	0.1238
10	0.1376	0.1395
11	0.1250	0.1206
12	0.1224	0.1095

Visually, we can see these values in Figure 4.3. The black line represents the observed proportion of incorrect calls made by Adrian Johnson at each time point. The blue, solid line represents the linear estimates from Model VII. The red, dotted line is the estimate for the average post-tech umpire from the random coefficients model above.

Figure 4.4 is a plot of the average estimated proportion of incorrect calls for pre- and post-tech umpires represented by a blue, dashed line and a solid, red line, respectively. Note these estimates are obtained from the $\hat{\beta}$ vector in Model VII. Later, we will test if the slope of each line is significantly different from zero as well as perform a test of parallelism.

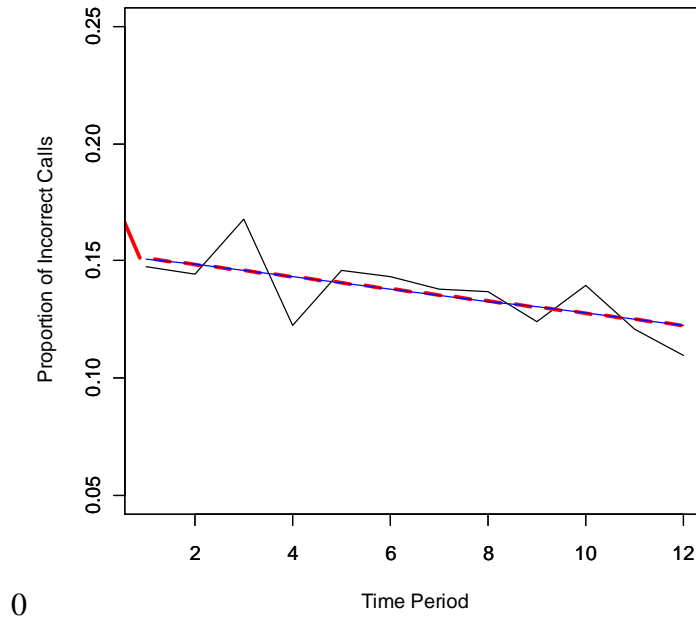


Figure 4.3 Observed (black) and estimated (blue) proportion of incorrect calls for Adrian Johnson and average proportion of incorrect calls for post-tech umpires (red, dotted) plotted

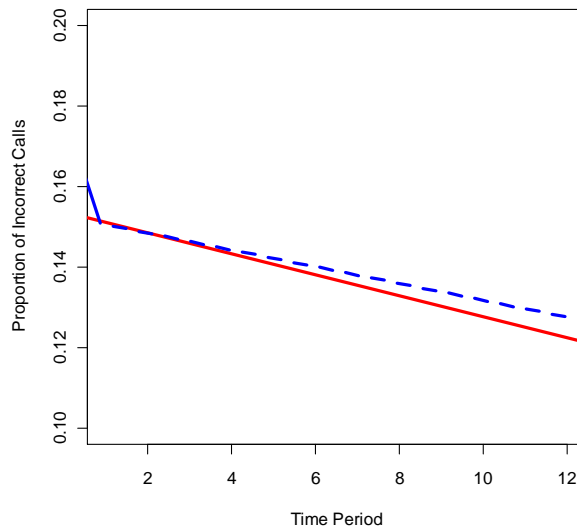


Figure 4.4 Plot of average estimates for Post-tech (red, solid) and Post-tech (blue, dashed) umpires

Answering Our Questions of Interest

First, we evaluate whether the change in the proportion of incorrect calls made by post-technology umpires significantly decreased. We do the same for pre-technology umpires.

Formally, we are testing

$$H_0: \beta_{1,Post-tech} \geq 0$$

$$H_A: \beta_{1,Post-tech} < 0$$

and

$$H_0: \beta_{1,Pre-tech} \geq 0$$

$$H_A: \beta_{1,Pre-tech} < 0$$

Our estimates of $\beta_{1,Post-tech}$ and $\beta_{1,Pre-tech}$ (and their respective standard errors) are -0.0026 (0.00041) and -0.0021 (0.000122). At a Type I error rate of 0.05, with a p-value for both < 0.0001 , we reject the null hypothesis in both cases (from Table 4.2). That is, there is a statistically significant decrease in the proportion of incorrect calls made by umpires from the beginning of the 2008 to the end of the 2011 seasons. Similar results are obtained regardless of the chosen covariance structures for \mathbf{D} and \mathbf{R}_i .

Second, we evaluate whether the change in the proportion of incorrect calls made by pre- and post-tech umpires is different over time. That is, are the average slopes seen in Figure 4.4 different? We perform a hypothesis test for difference in mean slopes. Formally, we are testing

$$H_0: \beta_{1,Post-tech} - \beta_{1,Pre-tech} = 0$$

$$H_A: \beta_{1,Post-tech} - \beta_{1,Pre-tech} \neq 0.$$

With a p-value of 0.2469, we fail to reject H_0 and conclude that it is possible that there is no difference in mean slopes. Similar results are obtained regardless of the chosen covariance structures for \mathbf{D} and \mathbf{R}_i . We conclude that we have no evidence the change in the proportion of incorrect calls is different for Pre-tech and Post-tech hired umpires.

Third, we aim to determine whether the proportion of incorrect calls differs significantly depends on the debut era of the umpire. To answer this question, we performed a Wald test of hypothesis for an overall difference in average proportion of incorrect calls for pre- and post-tech hired umpires. Formally, we are testing

$$H_0: \begin{pmatrix} \beta_{0,Post} - \beta_{0,Pre} \\ \beta_{1,Post} - \beta_{1,Pre} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$H_A: \begin{pmatrix} \beta_{0,Post} - \beta_{0,Pre} \\ \beta_{1,Post} - \beta_{1,Pre} \end{pmatrix} \neq \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

This test can be done in SAS using the following `CONTRAST` statement:

```
contrast 'overall tech diff' tech 1 -1, Coded*tech 1 -1/chisq;
```

With a Wald test-statistic of 2.68 and corresponding p-value of 0.2615 we conclude we do not have evidence that the average proportion of incorrect calls are different for each technology group. Again, regardless of the covariance structures for \mathbf{R}_i and \mathbf{D} we used, the conclusion comparing pre- and post-tech umpires remains the same.

Random Coefficients Model vs. Fixed Slopes Model

We now discuss whether we needed to include a random intercept or a random slope in the model. In our previous models, we included random effects for both intercept and slopes and thus call it a random coefficients model. Alternatively, we could treat one of these as fixed. Investigation into the \mathbf{D} matrix used in Model VII shows us that D_{22} , the variance of the b_{1i} 's, is virtually negligible relative to the size of the mean slope. According to Davidian (2005), this can create computational difficulties in the numerical algorithms we use to implement fitting a random coefficients model.

Alternatively, we considered a model with a random intercept and fixed slope.

Model VIII

$$Y_{ij} = \beta_{0i} + \beta_1 t_{ij} + e_{ij}$$

$$\beta_{0i} = \beta_0 + b_{0i}$$

This implies Model IX for the entire vector \mathbf{Y}_i is

Model IX

$$\mathbf{Y}_i = \mathbf{Z}_i \mathbf{A}_i \boldsymbol{\beta}_i + \mathbf{1} b_{0i} + \mathbf{e}_i$$

where $\mathbf{1}$ is an $(n_i \times 1)$ vector of 1's and $\mathbf{Z}_i \mathbf{A}_i$ is our design matrix for umpire i

$$\mathbf{Z}_i = \begin{pmatrix} 1 & t_{i1} \\ 1 & t_{i2} \\ \vdots & \vdots \\ 1 & t_{in_i} \end{pmatrix}$$

$$\mathbf{A}_i = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \text{ if Post - tech}$$

$$\mathbf{A}_i = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ if Pre - tech}$$

$$\boldsymbol{\beta}_i = \begin{pmatrix} \beta_{0,Post-tech} \\ \beta_{1,Post-tech} \\ \beta_{0,Pre-tech} \\ \beta_{1,Pre-tech} \end{pmatrix}$$

To informally test whether the slopes are fixed we can consider Model VI as the “full” random coefficients model and Model IX as the “reduced” fixed slopes model. In the reduced model

$$var(\mathbf{b}_i) = D_{11}.$$

We will assume in both the full and reduced model we can continue to use

$$var(\mathbf{e}_i) = \mathbf{R}_i = \sigma_{Pre-tech}^2 \mathbf{I}_{n_i}$$

$$var(\mathbf{e}_i) = \mathbf{R}_i = \sigma_{Post-tech}^2 \mathbf{I}_{n_i}$$

for pre- and post-technology hired umpires ,respectively. Again, we used AIC and BIC to decide which model to use. These are provided in Table 4.5. We see that both the AIC and BIC values are smaller for the mixed effects model where we assume slopes fixed. According to Davidian (2005, p. 387), formally testing this with a likelihood ratio test is difficult and is often not done by practioners.

Table 4.5 Fit statistics for random coefficients model and mixed effects model

	<i>-2 Log Likelihood</i>	<i>AIC</i>	<i>BIC</i>
Random Coefficients Model	-5389.1	-5373.1	-5352.5
Mixed effects model, random intercept, fixed slope	-5389	-5375	-5357

Again, regardless of the model chosen, we find no significant difference in slopes of pre- and post-technology umpires. Furthermore, we find no significant overall technology difference.

Fixed Slopes Model vs. General Linear Model

Now that we have decided a fixed slopes model is more appropriate than the random coefficients model, we look to see if treating the intercept as random is truly beneficial to our analysis. Again, we will test this informally with the AIC and BIC values; see Table 4.6.

We define our “full” model as Model IX and our “reduced” model in Model X.

Model X

$$Y_i = \mathbf{Z}_i \mathbf{A}_i \boldsymbol{\beta}_i + \mathbf{e}_i$$

where $\mathbf{Z}_i \mathbf{A}_i = \mathbf{X}_i$ is our design matrix for umpire i

$$\mathbf{Z}_i = \begin{pmatrix} 1 & t_{i1} \\ 1 & t_{i2} \\ \vdots & \vdots \\ 1 & t_{in_i} \end{pmatrix}$$

$$\mathbf{A}_i = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \text{ if } \textit{Post - tech}$$

$$\mathbf{A}_i = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \text{ if } \textit{Pre - tech}$$

$$\boldsymbol{\beta}_i = \begin{pmatrix} \beta_{0, \textit{Post-tech}} \\ \beta_{1, \textit{Post-tech}} \\ \beta_{0, \textit{Pre-tech}} \\ \beta_{1, \textit{Pre-tech}} \end{pmatrix}.$$

Here, we have no \mathbf{D} matrix, and our

$$\Sigma_i = \textit{var}(\mathbf{e}_i) = \mathbf{R}_i = \sigma_{\textit{Pre-tech}}^2 \mathbf{I}_{n_i}$$

$$\Sigma_i = \textit{var}(\mathbf{e}_i) = \mathbf{R}_i = \sigma_{\textit{Post-tech}}^2 \mathbf{I}_{n_i}$$

for pre- and post-technology umpires respectively.

Table 4.6 Fit Statistics for the general linear model

	<i>-2 Log Likelihood</i>	<i>AIC</i>	<i>BIC</i>
General Linear Model	-5266.5	-5254.5	-5239.0
Mixed effects model, random intercept, fixed slope	-5389	-5375	-5357

Comparing the AIC and BIC, it is clear that treating the intercepts as random is beneficial. However, had we naively chosen the general linear model, our conclusions about umpires would have remained the same. Note that the equation fit by the general linear model yields the same solid, red and dashed, blue lines in Figure 4.4.

Chapter 5 - Conclusion

In conclusion, all models we evaluated yielded the same answers to our questions of interest.

- 1) We can say that the proportion of incorrect calls has statistically significantly decreased from the beginning of the 2008 season to the end of the 2011 season.
- 2) We cannot say that pre- and post-technology umpires had significantly different proportions of incorrect calls.
- 3) We cannot say that pre- and post-technology umpires have a significantly different rate of change in the proportion of incorrect calls made.

In terms of model building, when evaluating umpires, it is beneficial to treat their intercepts as random. Furthermore, in our study, we found that we can treat the slope as fixed. While modeling variability is important to a statistician, we find that the random coefficients model is robust in terms of which covariance structure we used to model within-umpire variability.

Issues and Areas for Future Research

While we did our best to use PITCHf/x data to evaluate umpire performance, we had to make a very large assumption—a pitch outside the strike zone at the front of the plate (where PITCHf/x measures the location) will remain outside the strike zone. Josh Kalk (2009) gives a great example of where making this assumption would lead to a discrepancy in the umpire's call and the PITCHf/x systems call if the ball were to curve into the strike zone at the back of the plate. While this is not necessarily an area of research for a statistician, collaboration with a physicist may lead to an algorithm to handle this.

Initially, we set out to use the PITCHf/x data to see if the umpires' perceived strike zone is changing over time to look more like the rulebook defined strike zone. However, after reading literature, we found that this is not possible. The PITCHf/x operator defines the top of the strike zone in a fashion that describes the industry standard top of the strike zone. This calls into question how the MLB is using PITCHf/x to evaluate their umpires.

In our evaluation, we fit straight-line models. After looking at the data, more complex models, such as mixture models or spline fitting, may be more appropriate. A nonparametric approach to answering questions about umpire performance is also a suggested area of future research.

With the data we collected, one could consider analyzing day/night and indoor/outdoor games separately. Furthermore, investigation into the other covariates we obtained is still of interest. We chose to divide our data into thirds of a season; it may be appropriate to look at different time periods.

There are a great number of questions that can be answered now that public pitch-by-pitch data is available. One can investigate pitcher, batter, and umpire performance for different aspects of the game. We would find it interesting to see how a rookie pitcher's/batter's performance changes over time in high stress situations.

Bibliography

- Adair, R. K. (2003). Cameras and computers, or umpires?. *Baseball Research Journal*. 32. (2003): 22.
- Burnham, A and Anderson, D (1998). Model selection and inference. New York, New York: Springer-Verlag New York Inc..
- Davidian, M. (2005). *Applied Longitudinal Data Analysis*. Unpublished Lecture Notes, Department of Statistics, North Carolina State University, Raleigh, North Carolina. Available at: <http://www.stat.ncsu.edu/people/davidian/courses/st732/>
- Fast, M (2011a, February 16). The real strike zone. *Spinning Yarn*. Retrieved September 9, 2011, from Baseball Prospectus: <http://www.baseballprospectus.com/article.php?articleid=12965>
- Fast, M. (2011b, March 2). How accurate is PitchTrax?. *Spinning Yarn*. Retrieved May 30, 2012, from Baseball Prospectus: <http://www.baseballprospectus.com/article.php?articleid=13109>
- Fast, M. (2011c, June 1). The real strike zone, part 2. *Spinning Yarn*. Retrieved September 12, 2011, from Baseball Prospectus: <http://www.baseballprospectus.com/article.php?articleid=14098>
- Garik16. (2011, March 31) A Pitch F/x primer: some tips on run values, pitch classifications, and heat maps." Retrieved February 6, 2012, from Beyond the Boxscore: www.beyondtheboxscore.com/2011/3/31/2068855/pitch-fx-primer
- Gassko, D. (2007, February 1). The outside corner. Retrieved January 24, 2012, from The Hardball Times: <http://www.hardballtimes.com/main/printarticle/the-outside-corner/>
- Goldblatt, A. (2011). Major league umpires' performance, 2007-2010: a comprehensive statistical review. Jefferson, North Carolina, and London: McFarland & Company, Inc.
- Grabiner, D. The Sabermetric manifesto. Retrieved May 30, 2012, from Sean Lahman: <http://seanlahman.com/baseball-archive/sabermetrics/sabermetric-manifesto/>
- Kalk, J. (2009, January 20). That was a strike?. Retrieved September 12, 2011, from The Hardball Times: <http://www.hardballtimes.com/main/printarticle/that-was-a-strike/>
- Karegeannes, J. (2004, August 10). Confessions of a Questec operator: how the system works, how it can be improved. Retrieved January 24, 2012, from Baseball Prospectus: <http://www.baseballprospectus.com/article.php?articleid=3326>
- Kovach, M. (2011, June 2). A very non-statistical look at umpires and strike zones. Retrieved September 12, 2011, from The Hardball Times: http://www.hardballtimes.com/main/blog_article/a-very-non-statistical-look-at-umpires-and-strike-zones/

- Lavin, J. (2001, May 18). Technology and baseball. Retrieved January 23, 2012, from Computoregde Magazine: <http://joelavin.com/baseball.html>
- Leftkowitz, J. (2009). *Joe Leftkowitz's Pitch F/x tool*. [Datafile]. Available from: <http://www.joeleftkowitz.com/pitch.php>
- Lindgren, H. (2003, December 14). The foolproof umpire. *The New York Times*. Retrieved May 29, 2012, from QuesTec: http://www.questec.com/q2001/spfe_nytimes_1203.htm
- McGrath, L. (2010, April 19). Goal-line technology: crossing too many lines for FIFA? Retrieved May 29, 2012, from GeekWeek: <http://www.geekweek.com/2010/04/goal-line-technology-cross-too-many-lines-for-fifa-1.html>
- Nathan, Alan M. (2010, March 8). MLB PITCHf/x data. *MLB Extended Gameday Pitch Logs: A Tutorial*. Retrieved May 30, 2012, from: <http://webusers.npl.illinois.edu/~a-nathan/pob/tracking.htm>
- Palmer, P. (2007) Foreward. In T. M. Tango, M. G. Lichtman, A. E. Dolphin, *The Book: Playing the Percentages in Baseball*. (p.p. 12-13). Washington, D.C.: Potomac Books, Inc.
- Parsons, C.A., Sulaeman, J., Yates, M.C., and Hamermesh, D.S. (2011). "Strike Three: Discrimination, Incentives, and Evaluation." *American Economic Review*, 101, 1410—1435.
- Schlegel, J. (2010, July 20). Game changers: the electronic strike zone. Retrieved February 1, 2012, from MLB.com: http://mlb.mlb.com/news/print.jsp?ymd=20100719&content_id=12406800
- Vilines, A. (2010, September 28). The history of HawkEye: tennis' instant replay. Retrieved May 29, 2012, from GEM Tennis: <http://www.gemtennis.com/2010/09/28/the-history-of-hawkeye-tennis-instant-replay/>
- Walsh, J. (2007, July 25). The eye of the umpire. Retrieved May 29, 2012, from The Hardball Times: <http://www.hardballtimes.com/main/printarticle/the-eye-of-the-umpire/>

Appendix A – R Code

2008

Reading in the PITCHf/x data for 2008, matching to retrosheet, defining a strike zone

```
#####
## 2008 DATA ###
#####

##### The following will create a dataset to be read into SAS
##### We will call from PITCHf/x and RETROSHEET
##### We will output to FROM R TO SAS
```

```

##### This code is set up to be used on a PC

#####
###          ###
### Reading in PITCHf/x Data  ###
###          ###
#####

#On a Mac use:
#setwd("/Volumes/Masters/2008/Games Matched full/")

#On a PC use:
setwd("E://2008//Games Matched full//")

#Setting up the left and right strike zones.  These are constant for all batters
sz_left<- -0.8333
sz_right<- 0.8333

a<-list.files()
a

data2008<-c()
final.data.set<-c()
full.count.matrix<-c()
for (x in a) {
  setwd("E://2008//Games Matched full//")
  #setwd("/Volumes/Masters/2008/Games Matched full/")

  # Use the following line if you are only looking at one file
  #x<-"Matched 1 .csv"
  u<-read.csv(x, header = T, na.strings=T)
  u$dataset = x

  # Creating a dataset where we only have Balls and Called Strikes
  # And removing any observations where the top of the strike zone is 0.
  strikes_and_balls_2<-u[u[,"Pitch.Result"]=="Ball" | u[,"Pitch.Result"]=="Called Strike",]
  strikes_and_balls<-strikes_and_balls_2[strikes_and_balls_2[,"sz_top"]!=0,]

  # Setting up the Strike Zone for the left and the right, based on the size of home plate.
  #sz_left<-rep(-.8333,length(t(strikes_and_balls)))
  #sz_right<-rep(.8333,length(t(strikes_and_balls)))
  #strikes_and_balls<-cbind(strikes_and_balls,sz_left)
  #strikes_and_balls<-cbind(strikes_and_balls,sz_right)

  # Coding the PITCHf/x systems calls as 1s (balls) and 0s (strikes)

  #Set all values of ball.call equal to 0 (this will represent a strike)
  ball.call<-rep(0,dim(strikes_and_balls)[1])

  #Set ball.call equal to 1 if any of the following occur
  # 1) If the pitch is left of the sz_left
  # 2) If the pitch is right of the sz_right
  # 3) If the pitch is above the sz_top
  # 4) If the pitch is below the sz_bottom

  for (i in 1:length(ball.call)) {
    if (strikes_and_balls[i,"px"] <sz_left) ball.call[i]=1 else
      if (strikes_and_balls[i,"px"] > sz_right) ball.call[i]=1 else
        if (strikes_and_balls[i,"pz"] < strikes_and_balls[i,"sz_bot"]) ball.call[i]=1
    else
      if (strikes_and_balls[i,"pz"] > strikes_and_balls[i,"sz_top"])
  ball.call[i]=1 else ball.call[i]=0
  }

  # Coding the Umpires call as a 1s and 0s
  ump.call<-rep(99999,dim(strikes_and_balls)[1])

```



```

#Will be a zero if called a strike, 1 if called ball
for (i in 1:length(ump.call)) {
  if (strikes_and_balls[i,"Pitch.Result"]=="Ball") ump.call[i]=1 else ump.call[i]=0
}

unique(ball.call) #Verifying we have changed all ball.calls to either 1s or 0s
unique(ump.call) #Verifying we have changed all ump.calls to either 1s or 0s

#Attaching ball.call and ump.call to our dataset of information containing only balls and
strikes
strikes_and_balls<-cbind(strikes_and_balls, ball.call)
strikes_and_balls<-cbind(strikes_and_balls, ump.call)

## Coding if the Umpires call agrees with the PITCHf/x systems call.
#1s (umpire does not agree with system) 0s (umpire agrees with system).
error.call<-rep(999999,dim(strikes_and_balls)[1]) ###error.call will be a 1 if the umpire makes
an error in his call, a 0 if umpire agrees with PITCHf/x
unique(error.call)
for (i in 1:length(error.call)) {
  if (strikes_and_balls[i,"ump.call"] != strikes_and_balls[i,"ball.call"]) error.call[i]=1
  else error.call[i]=0
}

unique(error.call) #verifying we have at least one error.

error.call<-as.data.frame(error.call)
strikes_and_balls<-cbind(strikes_and_balls, error.call)

strikes_and_balls<-strikes_and_balls[,-
c(6,7,8,9,17,19,20,21,22,23,24,25,26,27,28,32,33,34,35,36,37,38,39,40)]
dim(strikes_and_balls)

#####
###                               ###
### Separating the games into     ###
### MAR APR MAY JUN JULY         ###
### AUG SEP OCT                  ###
###                               ###
#####

group1<-strikes_and_balls[strikes_and_balls$month <= 5, ] #MAR APR MAY
group2<-strikes_and_balls[strikes_and_balls$month <= 7 & strikes_and_balls$month >5, ] #JUN JULY
group3<-strikes_and_balls[strikes_and_balls$month > 7, ] #AUG SEP OCT

#####
###                               ###
### Separating the information     ###
### for each umpire for each game ###
###                               ###
#####

# Create vectors for the count of incorrect calls per game, left handedness, and fast pitches
count.incorrect.calls<-rep(0,length(unique(strikes_and_balls$gid)))
left.handed.pitcher.count<-rep(0,length(unique(strikes_and_balls$gid)))
left.handed.batter.count<-rep(0,length(unique(strikes_and_balls$gid)))
fastballs<-rep(0,length(unique(strikes_and_balls$gid)))
total.per.game<-rep(0,length(unique(strikes_and_balls$gid)))

# Create a vector to place the name of the umpire per game in
ump.count<-as.data.frame(rep(0,length(unique(strikes_and_balls$gid))))

count.matrix<-c()
#strikes_and_balls$pitch_type2<-strikes_and_balls$pitch_type

```

```

fastball<-rep(999, length(strikes_and_balls$pitch_type))

for (i in 1:length(strikes_and_balls$pitch_type)){
  if (strikes_and_balls$pitch_type[i]=="FC") {fastball[i]=1} else
    if (strikes_and_balls$pitch_type[i]=="FF") {fastball[i]=1} else
      if (strikes_and_balls$pitch_type[i]=="FT") {fastball[i]=1} else
        if (strikes_and_balls$pitch_type[i]=="FS") {fastball[i]=1} else
          if (strikes_and_balls$pitch_type[i]=="FA") {fastball[i]=1} else
            fastball[i]=0
        }
    }
strikes_and_balls<-cbind(strikes_and_balls,fastball)
total.fastball<-c()
for (i in 1:length(unique(strikes_and_balls$gid))){
  total.incorrect.per.game<-
sum(strikes_and_balls$error.call[strikes_and_balls["error.call"]==1 &
  strikes_and_balls["gid"]==unique(strikes_and_balls$gid)[i]])
  total.left.handed.pitcher<-
summary(strikes_and_balls[strikes_and_balls$gid==unique(strikes_and_balls$gid)[i],"Pitcher.Handedness"]) [1]
  total.left.handed.batter<-
summary(strikes_and_balls[strikes_and_balls$gid==unique(strikes_and_balls$gid)[i],"Batter.Handedness"]) [1]
  total.fastball<-
sum(strikes_and_balls$fastball[strikes_and_balls$gid==unique(strikes_and_balls$gid)[i]])
  total.per.game<-
length(strikes_and_balls$gid[strikes_and_balls$gid==unique(strikes_and_balls$gid[i])])

  left.handed.pitcher.count[i]<-total.left.handed.pitcher
  left.handed.batter.count[i]<-total.left.handed.batter
  count.incorrect.calls[i]<-total.incorrect.per.game
  fastballs[i]<-total.fastball
  ump.count[i]<-strikes_and_balls$Umpire[1]
}

count.incorrect.calls
left.handed.pitcher.count
left.handed.batter.count
fastballs
total.per.game
ump.count<-ump.count[,1]
count.title<-as.data.frame(unique(strikes_and_balls$gid))
individual.count.matrix<-cbind(count.incorrect.calls,left.handed.pitcher.count,
left.handed.batter.count,fastballs,total.per.game,ump.count,count.title)
colnames(individual.count.matrix)<-c("Count of Incorrect Calls", "Count of Left Pitchers",
"Count of Left Batters","Count of Fastballs","Total Count", "Umpire", "Game ID")
count.matrix<-rbind(count.matrix, individual.count.matrix)

game<-count.matrix["Game ID"] #Getting only the gid vectors

game<-as.character(game) #Turn it into a string

splitgame<-strsplit(game, "_") # splitting it into multiple strings
dataset<-as.data.frame(splitgame) # saving it as a matrix of strings
dataset<-t(dataset)

row.names(dataset)<-NULL
colnames(dataset)<-c("id", "year", "month", "day", "visit", "home", "meetings")
#<Verify that this is the order in which things are seen

dataset[,"home"]<-toupper(substring(dataset[,"home"],1,3))
dataset[,"visit"]<-toupper(substring(dataset[,"visit"],1,3))

dataset<-as.data.frame(dataset)
dataset$meetings<-as.numeric(dataset$meetings)
count.matrix <-cbind(count.matrix,dataset[,2:7])
dim(count.matrix)
count.matrix<-as.data.frame(count.matrix)

full.count.matrix<-rbind(full.count.matrix,count.matrix)

```

```

#####
###                               ###
### Reading in the Retrosheet data ###
###                               ###
#####

setwd("E://Retrosheet data")
#setwd("/Volumes/Masters/Retrosheet data") # <----CHANGE WHEN ON MAC
a<-list.files()
a

retro <- c()
#for (x in a) {
  x<-"2008 Retrosheet data.csv"
  u<-read.csv(x, header = T, na.strings=T)
  u$dataset = x
  retro<- rbind(retro, u)
  #}

unique(retro$dataset)

#On MAC use:                               <<<<<<<NEED TO CHANGE ON CAMPUS
#setwd("/Volumes/Masters/")

#On PC use:
setwd("E://")

listing<-read.csv("Retrosheet titles.csv", header=F)
colnames(retro)<-c(as.character(t(listing)), "dataset")

#let's get rid of the stuff we don't want
retro<-retro[,-c(5,8,10,11,14:16,20:77,80:161)]

if (retro[,2]==0) {retro[retro[,2]==0,2]=1}

## We need to change the date column into multiple columns such that we
## can match based on day, month, then year, then whether it is the first
## second or third game of a double or triple header.

#####
###                               ###
### Breaking up the date variable  ###
### for the RETRO data            ###
###                               ###
#####

date<-retro$Date #Getting only the Date vector

date<-as.character(date) #Turn it into a string

splitdate<-strsplit(date, "/") # splitting it into multiple strings
dataset<-as.data.frame(splitdate) # saving it as a matrix of strings
dataset<-t(dataset)

row.names(dataset)<-NULL
colnames(dataset)<-c("month", "day", "year")

retro<-cbind(retro,dataset[,1:3])
dim(retro)

#####
###                               ###
### Merging the 2 datasets         ###
### to make one awesome set       ###
###                               ###
#####

#to get only the games umpired by Adrian Johnson

```

```

mrg_2<-merge(count.matrix, retro, by.x=c("month", "home", "day", "meetings","year"),
  by.y=c("month", "Home Team", "day", "Number of Games", "year"))
mrg<-merge(count.matrix, retro, by.x=c("month", "home", "day"), by.y=c("month", "Home Team",
  "day"))

final.data.set<-rbind(final.data.set,mrg)
#data2008<-rbind(data2008, strikes_and_balls)
}

#write.csv(data2008,file="/Volumes/Masters/data2008.csv")
#write.csv(final.data.set,file="/Volumes/Masters/proportionsbygame.csv")

write.csv(final.data.set,file="E://From R to SAS 2//From R to SAS 2008 breaking into chunks part
1.csv")

```

Grouping the 2008 data into time periods for each umpire

```

##### 2008 Separated in chunks part 2

#####
## 2008 DATA ###
#####

##### The following will create a dataset to be read into SAS
##### We will call from PITCHf/x and RETROSHEET
##### We will separate 2008 into groups of months

##### This code is set up to be used on a PC

#####
###          ###
### Reading in the 1st chunk  ###
###          ###
#####

#On a Mac use:
#setwd("/Volumes/Masters/From R to SAS 2/")

#On a PC use:
setwd("E://From R to SAS 2//")

data2008<-read.csv("From R to SAS 2008 breaking into chunks part 1.csv", header=T)

group1<-data2008[data2008[,"month"]<=5,]      ## This gives months March, April, and May
group2<-data2008[data2008[,"month"]==6 | data2008[,"month"]==7,] #|
data2008[data2008[,"month"]==7,]      ## This gives months June, July
group3<-data2008[data2008[,"month"]>=8,]      ## This gives months August, September, October

prop.matrix.group1<-as.data.frame(t(as.data.frame(rep(0,6))))
row.names(prop.matrix.group1)<-NULL
colnames(prop.matrix.group1)<-c("Umpire", "Prop_Incorrect", "Prop_Left_Pitch",
  "Prop_Left_Batters", "Prop_Fastballs", "Total_Games")
prop.matrix.group1
for (i in 1: length(unique(group1[,"Home.plate.umpire.name"]))) {
  count_incorrect_calls<-
  sum(group1[group1[,"Home.plate.umpire.name"]==unique(group1[,"Home.plate.umpire.name"])[i], "Coun
t.of.Incorrect.Calls"])
  count_left_pitch<-
  sum(group1[group1[,"Home.plate.umpire.name"]==unique(group1[,"Home.plate.umpire.name"])[i], "Coun
t.of.Left.Pitchers"])
  count_left_batters<-
  sum(group1[group1[,"Home.plate.umpire.name"]==unique(group1[,"Home.plate.umpire.name"])[i], "Coun
t.of.Left.Batters"])
  count_fastballs<-
  sum(group1[group1[,"Home.plate.umpire.name"]==unique(group1[,"Home.plate.umpire.name"])[i], "Coun
t.of.Fastballs"])
  total_count<-
  sum(group1[group1[,"Home.plate.umpire.name"]==unique(group1[,"Home.plate.umpire.name"])[i], "Tota
l.Count"])
}

```

```

total_num_games<-
length(group1[group1["Home.plate.umpire.name"]==unique(group1["Home.plate.umpire.name"])[i],"H
ome.plate.umpire.name"])

prop_incorrect<-count_incorrect_calls/total_count
prop_left_pitch<-count_left_pitch/total_count
prop_left_batters<-count_left_batters/total_count
prop_fastballs<-count_fastballs/total_count

ump.prop<-c(as.character(unique(group1["Home.plate.umpire.name"])[i]), prop_incorrect,
prop_left_pitch, prop_left_batters, prop_fastballs, total_num_games)

ump.prop<-as.data.frame(t(as.data.frame(ump.prop)))
colnames(ump.prop)<-c("Umpire", "Prop_Incorrect", "Prop_Left_Pitch", "Prop_Left_Batters",
"Prop_Fastballs", "Total_Games")
row.names(ump.prop)<-NULL
prop.matrix.group1<-rbind(prop.matrix.group1, ump.prop)
}
prop.matrix.group1<-prop.matrix.group1[-1,]
prop.matrix.group1

Early_Mid_late<-rep("Early", dim(prop.matrix.group1)[1])

prop.matrix.group1<-cbind(prop.matrix.group1,Early_Mid_late)
prop.matrix.group1

prop.matrix.group2<-as.data.frame(t(as.data.frame(rep(0,6))))
row.names(prop.matrix.group2)<-NULL
colnames(prop.matrix.group2)<-c("Umpire", "Prop_Incorrect", "Prop_Left_Pitch",
"Prop_Left_Batters", "Prop_Fastballs", "Total_Games")
prop.matrix.group2
for (i in 1: length(unique(group2["Home.plate.umpire.name"]))) {
count_incorrect_calls<-
sum(group2[group2["Home.plate.umpire.name"]==unique(group2["Home.plate.umpire.name"])[i],"Coun
t.of.Incorrect.Calls"])
count_left_pitch<-
sum(group2[group2["Home.plate.umpire.name"]==unique(group2["Home.plate.umpire.name"])[i],"Coun
t.of.Left.Pitchers"])
count_left_batters<-
sum(group2[group2["Home.plate.umpire.name"]==unique(group2["Home.plate.umpire.name"])[i],"Coun
t.of.Left.Batters"])
count_fastballs<-
sum(group2[group2["Home.plate.umpire.name"]==unique(group2["Home.plate.umpire.name"])[i],"Coun
t.of.Fastballs"])
total_count<-
sum(group2[group2["Home.plate.umpire.name"]==unique(group2["Home.plate.umpire.name"])[i],"Tota
l.Count"])
total_num_games<-
length(group2[group2["Home.plate.umpire.name"]==unique(group2["Home.plate.umpire.name"])[i],"H
ome.plate.umpire.name"])

prop_incorrect<-count_incorrect_calls/total_count
prop_left_pitch<-count_left_pitch/total_count
prop_left_batters<-count_left_batters/total_count
prop_fastballs<-count_fastballs/total_count

ump.prop<-c(as.character(unique(group2["Home.plate.umpire.name"])[i]), prop_incorrect,
prop_left_pitch, prop_left_batters, prop_fastballs, total_num_games)

ump.prop<-as.data.frame(t(as.data.frame(ump.prop)))
colnames(ump.prop)<-c("Umpire", "Prop_Incorrect", "Prop_Left_Pitch", "Prop_Left_Batters",
"Prop_Fastballs", "Total_Games")
row.names(ump.prop)<-NULL
prop.matrix.group2<-rbind(prop.matrix.group2, ump.prop)
}
prop.matrix.group2<-prop.matrix.group2[-1,]
prop.matrix.group2

Early_Mid_late<-rep("Mid", dim(prop.matrix.group2)[1])

```

```

prop.matrix.group2<-cbind(prop.matrix.group2,Early_Mid_late)
prop.matrix.group2

prop.matrix.group3<-as.data.frame(t(as.data.frame(rep(0,6))))
row.names(prop.matrix.group3)<-NULL
colnames(prop.matrix.group3)<-c("Umpire", "Prop_Incorrect", "Prop_Left_Pitch",
  "Prop_Left_Batters", "Prop_Fastballs", "Total_Games")
prop.matrix.group3
for (i in 1: length(unique(group3["Home.plate.umpire.name"]))) {
  count_incorrect_calls<-
  sum(group3[group3["Home.plate.umpire.name"]==unique(group3["Home.plate.umpire.name"])[i],"Count.of.Incorrect.Calls"])
  count_left_pitch<-
  sum(group3[group3["Home.plate.umpire.name"]==unique(group3["Home.plate.umpire.name"])[i],"Count.of.Left.Pitchers"])
  count_left_batters<-
  sum(group3[group3["Home.plate.umpire.name"]==unique(group3["Home.plate.umpire.name"])[i],"Count.of.Left.Batters"])
  count_fastballs<-
  sum(group3[group3["Home.plate.umpire.name"]==unique(group3["Home.plate.umpire.name"])[i],"Count.of.Fastballs"])
  total_count<-
  sum(group3[group3["Home.plate.umpire.name"]==unique(group3["Home.plate.umpire.name"])[i],"Total.Count"])
  total_num_games<-
  length(group3[group3["Home.plate.umpire.name"]==unique(group3["Home.plate.umpire.name"])[i],"Home.plate.umpire.name"])

  prop_incorrect<-count_incorrect_calls/total_count
  prop_left_pitch<-count_left_pitch/total_count
  prop_left_batters<-count_left_batters/total_count
  prop_fastballs<-count_fastballs/total_count

  ump.prop<-c(as.character(unique(group3["Home.plate.umpire.name"])[i]), prop_incorrect,
  prop_left_pitch, prop_left_batters, prop_fastballs, total_num_games)

  ump.prop<-as.data.frame(t(as.data.frame(ump.prop)))
  colnames(ump.prop)<-c("Umpire", "Prop_Incorrect", "Prop_Left_Pitch", "Prop_Left_Batters",
  "Prop_Fastballs", "Total_Games")
  row.names(ump.prop)<-NULL
  prop.matrix.group3<-rbind(prop.matrix.group3, ump.prop)
}
prop.matrix.group3<-prop.matrix.group3[-1,]
prop.matrix.group3

Early_Mid_late<-rep("Late", dim(prop.matrix.group3)[1])

prop.matrix.group3<-cbind(prop.matrix.group3,Early_Mid_late)
prop.matrix.group3

prop.matrix.2008<-rbind(prop.matrix.group1, prop.matrix.group2, prop.matrix.group3)
prop.matrix.2008

#####
## Attaching Umpire Information ##
#####

umpire<-read.csv("E://Umpire list.csv", header=T)

merged_2008<-merge(prop.matrix.2008, umpire, by.x="Umpire", by.y="Umpire.Name")
merged_2008<-merged_2008[,c(1:7,12,13)]
merged_2008

write.csv(merged_2008, file="E://From R to SAS 2//Full Year data//2008 broken into 3rds.csv")

```

2009 (2010 and 2011 follow in the same fashion)

Reading in the PITCHf/x data for 2009, matching to retrosheet, defining a strike zone

```
#####
## 2009 DATA ###
#####

##### The following will create a dataset to be read into SAS
##### We will call from PITCHf/x and RETROSHEET
##### We will output to FROM R TO SAS

##### This code is set up to be used on a PC

#####
###          ###
### Reading in PITCHf/x Data  ###
###          ###
#####

#On a Mac use:
#setwd("/Volumes/Masters/Datafiles/2009/CSV files/")

#On a PC use:
setwd("E://Datafiles//2009//CSV files//")

#Setting up the left and right strike zones.  These are constant for all batters
sz_left<- -0.8333
sz_right<- 0.8333

a<-list.files()
a

data2009<-c()
final.data.set<-c()
full.count.matrix<-c()
for (x in a) {
  setwd("E://Datafiles//2009//CSV files//")
  #setwd("/Volumes/Masters/Datafiles/2009/CSV files/")

  # Use the following line if you are only looking at one file
  #x<-"Adrian Johnson 2009.csv"
  u<-read.csv(x, header = T, na.strings=T)
  u$dataset = x

  # Creating a dataset where we only have Balls and Called Strikes
  # And removing any observations where the top of the strike zone is 0.
  strikes_and_balls_2<-u[u[,"Pitch.Result"]=="Ball" | u[,"Pitch.Result"]=="Called Strike",]
  strikes_and_balls<-strikes_and_balls_2[strikes_and_balls_2[,"sz_top"]!=0,]

  # Setting up the Strike Zone for the left and the right, based on the size of home plate.
  #sz_left<-rep(-.8333,length(t(strikes_and_balls)))
  #sz_right<-rep(.8333,length(t(strikes_and_balls)))
  #strikes_and_balls<-cbind(strikes_and_balls,sz_left)
  #strikes_and_balls<-cbind(strikes_and_balls,sz_right)

  # Coding the PITCHf/x systems calls as 1s (balls) and 0s (strikes)

  #Set all values of ball.call equal to 0 (this will represent a strike)
  ball.call<-rep(0,dim(strikes_and_balls)[1])

  #Set ball.call equal to 1 if any of the following occur
  # 1) If the pitch is left of the sz_left
  # 2) If the pitch is right of the sz_right
  # 3) If the pitch is above the sz_top
  # 4) If the pitch is below the sz_bottom

  for (i in 1:length(ball.call)) {
    if (strikes_and_balls[i,"px"] <sz_left) ball.call[i]=1 else
      if (strikes_and_balls[i,"px"] > sz_right) ball.call[i]=1 else
        if (strikes_and_balls[i,"pz"] < strikes_and_balls[i,"sz_bot"]) ball.call[i]=1
  }
}
```

```

else
    if (strikes_and_balls[i,"pz"] > strikes_and_balls[i,"sz_top"])
ball.call[i]=1 else ball.call[i]=0
}

# Coding the Umpires call as a 1s and 0s
ump.call<-rep(99999,dim(strikes_and_balls)[1])

#Will be a zero if called a strike, 1 if called ball
for (i in 1:length(ump.call)) {
  if (strikes_and_balls[i,"Pitch.Result"]=="Ball") ump.call[i]=1 else ump.call[i]=0
}

unique(ball.call) #Verifying we have changed all ball.calls to either 1s or 0s
unique(ump.call) #Verifying we have changed all ump.calls to either 1s or 0s

#Attaching ball.call and ump.call to our dataset of information containing only balls and strikes
strikes_and_balls<-cbind(strikes_and_balls, ball.call)
strikes_and_balls<-cbind(strikes_and_balls, ump.call)

## Coding if the Umpires call agrees with the PITCHf/x systems call.
#1s (umpire does not agree with system) 0s (umpire agrees with system).
error.call<-rep(999999,dim(strikes_and_balls)[1]) ###error.call will be a 1 if the umpire makes
an error in his call, a 0 if umpire agrees with PITCHf/x
unique(error.call)
for (i in 1:length(error.call)) {
  if (strikes_and_balls[i,"ump.call"] != strikes_and_balls[i,"ball.call"]) error.call[i]=1 else
  error.call[i]=0
}

unique(error.call) #verifying we have at least one error.

error.call<-as.data.frame(error.call)
strikes_and_balls<-cbind(strikes_and_balls, error.call)

strikes_and_balls<-strikes_and_balls[,-
c(6,7,8,9,17,19,20,21,22,23,24,25,26,27,28,32,33,34,35,36,37,38,39,40)]
dim(strikes_and_balls)

#####
###                               ###
### Separating the information    ###
### for each umpire for each game ###
###                               ###
#####

# Create vectors for the count of incorrect calls per game, left handedness, and fast pitches
count.incorrect.calls<-rep(0,length(unique(strikes_and_balls$gid)))
left.handed.pitcher.count<-rep(0,length(unique(strikes_and_balls$gid)))
left.handed.batter.count<-rep(0,length(unique(strikes_and_balls$gid)))
fastballs<-rep(0,length(unique(strikes_and_balls$gid)))
total.per.game<-rep(0,length(unique(strikes_and_balls$gid)))

# Create a vector to place the name of the umpire per game in
ump.count<-as.data.frame(rep(0,length(unique(strikes_and_balls$gid))))

count.matrix<-c()
#strikes_and_balls$pitch_type2<-strikes_and_balls$pitch_type
fastball<-rep(999, length(strikes_and_balls$pitch_type))

for (i in 1:length(strikes_and_balls$pitch_type)){
  if (strikes_and_balls$pitch_type[i]=="FC") {fastball[i]=1} else
  if (strikes_and_balls$pitch_type[i]=="FF") {fastball[i]=1} else

```



```

        if (strikes_and_balls$pitch_type[i]=="FT") {fastball[i]=1} else
          if (strikes_and_balls$pitch_type[i]=="FS") {fastball[i]=1} else
            if (strikes_and_balls$pitch_type[i]=="FA") {fastball[i]=1} else
fastball[i]=0
    }
strikes_and_balls<-cbind(strikes_and_balls,fastball)
total.fastball<-c()
for (i in 1:length(unique(strikes_and_balls$gid))){
    total.incorrect.per.game<-
sum(strikes_and_balls$error.call[strikes_and_balls[, "error.call"]==1 &
    strikes_and_balls[, "gid"]==unique(strikes_and_balls$gid) [i]])
    total.left.handed.pitcher<-
summary(strikes_and_balls[strikes_and_balls$gid==unique(strikes_and_balls$gid) [i], "Pitcher.Hande
dness"]) [1]
    total.left.handed.batter<-
summary(strikes_and_balls[strikes_and_balls$gid==unique(strikes_and_balls$gid) [i], "Batter.Hande
dness"]) [1]
    total.fastball<-
sum(strikes_and_balls$fastball[strikes_and_balls$gid==unique(strikes_and_balls$gid) [i]])
    total.count<-
length(strikes_and_balls$gid[strikes_and_balls$gid==unique(strikes_and_balls$gid) [i]])

    left.handed.pitcher.count[i]<-total.left.handed.pitcher
    left.handed.batter.count[i]<-total.left.handed.batter
    count.incorrect.calls[i]<-total.incorrect.per.game
    fastballs[i]<-total.fastball
    ump.count[i]<-strikes_and_balls$Umpire[1]
    total.per.game[i]<-total.count
}

count.incorrect.calls
left.handed.pitcher.count
left.handed.batter.count
fastballs
total.per.game
ump.count<-ump.count[,1]
count.title<-as.data.frame(unique(strikes_and_balls$gid))
individual.count.matrix<-cbind(count.incorrect.calls,left.handed.pitcher.count,
left.handed.batter.count,fastballs,total.per.game,ump.count,count.title)
colnames(individual.count.matrix)<-c("Count of Incorrect Calls", "Count of Left Pitchers",
"Count of Left Batters","Count of Fastballs","Total Count", "Umpire", "Game ID")
count.matrix<-rbind(count.matrix, individual.count.matrix)

game<-count.matrix[, "Game ID"] #Getting only the gid vectors

game<-as.character(game) #Turn it into a string

splitgame<-strsplit(game, "_") # splitting it into multiple strings
dataset<-as.data.frame(splitgame) # saving it as a matrix of strings
dataset<-t(dataset)

row.names(dataset)<-NULL
colnames(dataset)<-c("id", "year", "month", "day", "visit", "home", "meetings")
#<Verify that this is the order in which things are seen

dataset[, "home"]<-toupper(substring(dataset[, "home"],1,3))
dataset[, "visit"]<-toupper(substring(dataset[, "visit"],1,3))

dataset<-as.data.frame(dataset)
dataset$meetings<-as.numeric(dataset$meetings)
count.matrix <-cbind(count.matrix,dataset[,2:7])
dim(count.matrix)
count.matrix<-as.data.frame(count.matrix)

full.count.matrix<-rbind(full.count.matrix,count.matrix)

#####
###                               ###
### Reading in the Retrosheet data ###

```

```

###                                     ###
#####

setwd("E://Retrosheet data")
#setwd("/Volumes/Masters/Retrosheet data") # <----CHANGE WHEN ON MAC
a<-list.files()
a

retro <- c()
#for (x in a) {
  x<-"2009 Retrosheet data.csv"
  u<-read.csv(x, header = T, na.strings=T)
  u$dataset = x
  retro<- rbind(retro, u)
  #}

unique(retro$dataset)

#On MAC use:                               <<<<<<<NEED TO CHANGE ON CAMPUS
#setwd("/Volumes/Masters/")

#On PC use:
setwd("E://")

listing<-read.csv("Retrosheet titles.csv", header=F)
colnames(retro)<-c(as.character(t(listing)), "dataset")

#let's get rid of the stuff we don't want
retro<-retro[,-c(5,8,10,11,14:16,20:77,80:161)]

if (retro[,2]==0) {retro[retro[,2]==0,2]=1}

## We need to change the date column into multiple columns such that we
## can match based on day, month, then year, then whether it is the first
## second or third game of a double or triple header.

#####
###                                     ###
###   Breaking up the date variable   ###
###         for the RETRO data         ###
###                                     ###
#####

date<-retro$Date #Getting only the Date vector

date<-as.character(date) #Turn it into a string

splitdate<-strsplit(date, "/") # splitting it into multiple strings
dataset<-as.data.frame(splitdate) # saving it as a matrix of strings
dataset<-t(dataset)

row.names(dataset)<-NULL
colnames(dataset)<-c("month", "day", "year")

retro<-cbind(retro,dataset[,1:3])
dim(retro)

#####
###                                     ###
###   Merging the 2 datasets           ###
###   to make one awesome set         ###
###                                     ###
#####

#to get only the games umpired by Adrian Johnson

mrg_2<-merge(count.matrix, retro, by.x=c("month", "home", "day", "meetings","year"),
  by.y=c("month", "Home Team", "day", "Number of Games", "year"))
mrg<-merge(count.matrix, retro, by.x=c("month", "home", "day"), by.y=c("month", "Home Team",
  "day"))

```

```

final.data.set<-rbind(final.data.set,mrg)
  #data2009<-rbind(data2009, strikes_and_balls)
  }

#write.csv(data2009,file="/Volumes/Masters/data2009.csv")
#write.csv(final.data.set,file="/Volumes/Masters/proportionsbygame.csv")

write.csv(final.data.set,file="E://From R to SAS 2//From R to SAS 2009 breaking into chunks part
1.csv")

```

Grouping the 2009 data into time periods for each umpire

```

##### 2009 Separated in chunks part 2

#####
## 2009 DATA ##
#####

##### The following will create a dataset to be read into SAS
##### We will call from PITCHf/x and RETROSHEET
##### We will separate 2009 into groups of months

##### This code is set up to be used on a PC

#####
###                               ###
### Reading in the 1st chunk  ###
###                               ###
#####

#On a Mac use:
#setwd("/Volumes/Masters/From R to SAS 2/")

#On a PC use:
setwd("E://From R to SAS 2//")

data2009<-read.csv("From R to SAS 2009 breaking into chunks part 1.csv", header=T)

group1<-data2009[data2009[,"month"]<=5,]      ## This gives months March, April, and May
group2<-data2009[data2009[,"month"]==6 | data2009[,"month"]==7,] #|
  data2009[data2008[,"month"]==7,]  ## This gives months June, July
group3<-data2009[data2009[,"month"]>=8,]    ## This gives months August, September, October

prop.matrix.group1<-as.data.frame(t(as.data.frame(rep(0,6))))
row.names(prop.matrix.group1)<-NULL
colnames(prop.matrix.group1)<-c("Umpire", "Prop_Incorrect", "Prop_Left_Pitch",
  "Prop_Left_Batters", "Prop_Fastballs", "Total_Games")
prop.matrix.group1
for (i in 1:length(unique(group1[,"Home.plate.umpire.name"]))) {

  count_incorrect_calls<-
  sum(group1[group1[,"Home.plate.umpire.name"]==unique(group1[,"Home.plate.umpire.name"])[i], "Coun
t.of.Incorrect.Calls"])
  count_left_pitch<-
  sum(group1[group1[,"Home.plate.umpire.name"]==unique(group1[,"Home.plate.umpire.name"])[i], "Coun
t.of.Left.Pitchers"])
  count_left_batters<-
  sum(group1[group1[,"Home.plate.umpire.name"]==unique(group1[,"Home.plate.umpire.name"])[i], "Coun
t.of.Left.Batters"])
  count_fastballs<-
  sum(group1[group1[,"Home.plate.umpire.name"]==unique(group1[,"Home.plate.umpire.name"])[i], "Coun
t.of.Fastballs"])
  total_count<-
  sum(group1[group1[,"Home.plate.umpire.name"]==unique(group1[,"Home.plate.umpire.name"])[i], "Tota
l.Count"])
  total_num_games<-
  length(group1[group1[,"Home.plate.umpire.name"]==unique(group1[,"Home.plate.umpire.name"])[i], "H
ome.plate.umpire.name"])

```

```

prop_incorrect<-count_incorrect_calls/total_count
prop_left_pitch<-count_left_pitch/total_count
prop_left_batters<-count_left_batters/total_count
prop_fastballs<-count_fastballs/total_count

ump.prop<-c(as.character(unique(group1["Home.plate.umpire.name"])[i]), prop_incorrect,
prop_left_pitch, prop_left_batters, prop_fastballs, total_num_games)

ump.prop<-as.data.frame(t(as.data.frame(ump.prop)))
colnames(ump.prop)<-c("Umpire", "Prop_Incorrect", "Prop_Left_Pitch", "Prop_Left_Batters",
"Prop_Fastballs", "Total_Games")
row.names(ump.prop)<-NULL
prop.matrix.group1<-rbind(prop.matrix.group1, ump.prop)
}
prop.matrix.group1<-prop.matrix.group1[-1,]
prop.matrix.group1

Early_Mid_late<-rep("Early", dim(prop.matrix.group1)[1])

prop.matrix.group1<-cbind(prop.matrix.group1,Early_Mid_late)
prop.matrix.group1

prop.matrix.group2<-as.data.frame(t(as.data.frame(rep(0,6))))
row.names(prop.matrix.group2)<-NULL
colnames(prop.matrix.group2)<-c("Umpire", "Prop_Incorrect", "Prop_Left_Pitch",
"Prop_Left_Batters", "Prop_Fastballs", "Total_Games")
prop.matrix.group2
for (i in 1:length(unique(group2["Home.plate.umpire.name"]))) {

count_incorrect_calls<-
sum(group2[group2["Home.plate.umpire.name"]==unique(group2["Home.plate.umpire.name"])[i], "Count.of.Incorrect.Calls"])
count_left_pitch<-
sum(group2[group2["Home.plate.umpire.name"]==unique(group2["Home.plate.umpire.name"])[i], "Count.of.Left.Pitchers"])
count_left_batters<-
sum(group2[group2["Home.plate.umpire.name"]==unique(group2["Home.plate.umpire.name"])[i], "Count.of.Left.Batters"])
count_fastballs<-
sum(group2[group2["Home.plate.umpire.name"]==unique(group2["Home.plate.umpire.name"])[i], "Count.of.Fastballs"])
total_count<-
sum(group2[group2["Home.plate.umpire.name"]==unique(group2["Home.plate.umpire.name"])[i], "Total.Count"])
total_num_games<-
length(group2[group2["Home.plate.umpire.name"]==unique(group2["Home.plate.umpire.name"])[i], "Home.plate.umpire.name"])

prop_incorrect<-count_incorrect_calls/total_count
prop_left_pitch<-count_left_pitch/total_count
prop_left_batters<-count_left_batters/total_count
prop_fastballs<-count_fastballs/total_count

ump.prop<-c(as.character(unique(group2["Home.plate.umpire.name"])[i]), prop_incorrect,
prop_left_pitch, prop_left_batters, prop_fastballs, total_num_games)

ump.prop<-as.data.frame(t(as.data.frame(ump.prop)))
colnames(ump.prop)<-c("Umpire", "Prop_Incorrect", "Prop_Left_Pitch", "Prop_Left_Batters",
"Prop_Fastballs", "Total_Games")
row.names(ump.prop)<-NULL
prop.matrix.group2<-rbind(prop.matrix.group2, ump.prop)
}
prop.matrix.group2<-prop.matrix.group2[-1,]
prop.matrix.group2

Early_Mid_late<-rep("Mid", dim(prop.matrix.group2)[1])

prop.matrix.group2<-cbind(prop.matrix.group2,Early_Mid_late)
prop.matrix.group2

```

```

prop.matrix.group3<-as.data.frame(t(as.data.frame(rep(0,6))))
row.names(prop.matrix.group3)<-NULL
colnames(prop.matrix.group3)<-c("Umpire", "Prop_Incorrect", "Prop_Left_Pitch",
"Prop_Left_Batters", "Prop_Fastballs", "Total_Games")
prop.matrix.group3
for (i in 1:length(unique(group3[,"Home.plate.umpire.name"]))) {

count_incorrect_calls<-
sum(group3[group3[,"Home.plate.umpire.name"]==unique(group3[,"Home.plate.umpire.name"])[i], "Coun
t.of.Incorrect.Calls"])
count_left_pitch<-
sum(group3[group3[,"Home.plate.umpire.name"]==unique(group3[,"Home.plate.umpire.name"])[i], "Coun
t.of.Left.Pitchers"])
count_left_batters<-
sum(group3[group3[,"Home.plate.umpire.name"]==unique(group3[,"Home.plate.umpire.name"])[i], "Coun
t.of.Left.Batters"])
count_fastballs<-
sum(group3[group3[,"Home.plate.umpire.name"]==unique(group3[,"Home.plate.umpire.name"])[i], "Coun
t.of.Fastballs"])
total_count<-
sum(group3[group3[,"Home.plate.umpire.name"]==unique(group3[,"Home.plate.umpire.name"])[i], "Tota
l.Count"])
total_num_games<-
length(group3[group3[,"Home.plate.umpire.name"]==unique(group3[,"Home.plate.umpire.name"])[i], "H
ome.plate.umpire.name"])

prop_incorrect<-count_incorrect_calls/total_count
prop_left_pitch<-count_left_pitch/total_count
prop_left_batters<-count_left_batters/total_count
prop_fastballs<-count_fastballs/total_count

ump.prop<-c(as.character(unique(group3[,"Home.plate.umpire.name"])[i]), prop_incorrect,
prop_left_pitch, prop_left_batters, prop_fastballs, total_num_games)

ump.prop<-as.data.frame(t(as.data.frame(ump.prop)))
colnames(ump.prop)<-c("Umpire", "Prop_Incorrect", "Prop_Left_Pitch", "Prop_Left_Batters",
"Prop_Fastballs", "Total_Games")
row.names(ump.prop)<-NULL
prop.matrix.group3<-rbind(prop.matrix.group3, ump.prop)
}
prop.matrix.group3<-prop.matrix.group3[-1,]
prop.matrix.group3

Early_Mid_late<-rep("Late", dim(prop.matrix.group3)[1])

prop.matrix.group3<-cbind(prop.matrix.group3,Early_Mid_late)
prop.matrix.group3

prop.matrix.2009<-rbind(prop.matrix.group1, prop.matrix.group2, prop.matrix.group3)
prop.matrix.2009

#####
## Attaching Umpire Information ##
#####

umpire<-read.csv("E://Umpire list.csv", header=T)

merged_2009<-merge(prop.matrix.2009, umpire, by.x="Umpire", by.y="Umpire.Name")
merged_2009<-merged_2009[,c(1:7,12,13)]
merged_2009

write.csv(merged_2009, file="E://From R to SAS 2//Full Year data//2009 broken into 3rds.csv")

```

Appendix B – SAS Code

Importing and creating a dataset for exporting

```
options nodate pageno=1 formdlim="~";

PROC IMPORT OUT= WORK.masters_chunked
  DATAFILE="E:\From R to SAS 2\Full Year data\Adapted\All in 1
file\Chunked data.xlsx"
  DBMS=EXCEL REPLACE;
  Range="Chunked data$";
  GETNAMES=YES;
  MIXED=NO;
  SCANTEXT=YES;
  USEDATE=YES;
  SCANTIME=YES;
run;

data masters;
  set masters_chunked;
  if MLB_Debut < 2002 then tech = 'Pre-tech';
  if MLB_Debut >= 2002 then tech = 'Post-tech';

  if Birthdate <=1964 then gen = 'Babyboom'; else
  if 1965<= Birthdate <= 1980 then gen = 'Gen_X'; else
  if Birthdate >1980 then gen = 'Gen_Y';

  coded1=coded;

run;

proc print data=masters;
run;

proc export data=masters
outfile="E:\Chunked data to excel.csv"
dbms=CSV
replace;
run;

proc sort data=masters;
by tech umpire coded;
run;

ods html file="E:/SAS OUTPUT CHUNKED DATA.html";
```

Code for our random coefficients models

```
* MODEL 0;
* Ri = TOEP(3) with variance sigma^2 different in both techs;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix different for both techs;
* Specified in the RANDOM statement;
```

```

title 'RANDOM COEFFICIENT MODEL WITH TWO-DEPENDENT WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE DIFFERENT FOR EACH TECH';
title3 'DIFFERENT D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
  class Umpire tech gen;
  model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
  random intercept coded/type=un subject=umpire group=tech;* g gcorr v
vcorr;
  repeated / type=toep(3) subject=umpire group=tech;* r rcorr;
  estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
  contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;

run;

*****;
* Models with Ri=Diagonal, D=UNSTRUCTURED ;
* SAME-SAME, DIFF-SAME, SAME-DIFF, DIFF-DIFF ;
*****;

* MODEL 1;
* Ri = diagonal with constant variance sigma^2 same in both techs;
* No REPEATED statement necessary to fit this Ri (default);
* D = (2x2) unstructured matrix same for both techs;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH DIAGONAL WITHIN-UMPIRE';
title2 'COVARIANCE MATRIX WITH CONSTANT VARIANCE SAME FOR EACH TECH';
title3 'SAME D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
  class Umpire tech coded1;
  model Prop_Incorrect = tech tech*coded/noint solution chisq;
  random intercept coded/type=un subject=umpire;* g gcorr v vcorr;
  estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
  contrast 'overall tech diff' tech 1 -1, Coded*tech 1 -1/chisq;

run;

* MODEL 2;
* Fit the same model but with separate diagonal Ri matrix for;
* each tech. Thus there are 2 separate variances sigma^2_(Pre and Post);
* D still = (2x2) unstructured matrix same for both techs;
* Specified in the RANDOM statement;
title 'RANDOM COEFFICIENT MODEL WITH DIAGONAL WITHIN-UMPIRE';
title2 'COVARIANCE MATRIX WITH SEPARATE CONSTANT VARIANCE FOR EACH TECH';
title3 'SAME D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml covtest cl;
  class Umpire tech coded1;
  model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
  repeated / group=tech subject=umpire r rcorr;
  random intercept coded/type=un subject=umpire g gcorr v vcorr;
  estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
  contrast 'overall tech diff' tech 1 -1, Coded*tech 1 -1/chisq;

run;

* MODEL 3;
** Ri = diagonal with constant variance sigma^2 same in both techs;

```

```

* No REPEATED statement necessary to fit this Ri (default);
* D = (2x2) unstructured matrix different for both techs;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH DIAGONAL WITHIN-UMPIRE';
title2 'COVARIANCE MATRIX WITH SAME CONSTANT VARIANCE FOR EACH TECH';
title3 'DIFFERENT D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml covtest cl;
  class Umpire tech coded1;
  model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
  *repeated / group=tech subject=umpire;
  random intercept coded/type=un subject=umpire group=tech;* g gcorr v
vcorr;
  estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
  contrast 'overall tech diff' tech 1 -1, Coded*tech 1 -1/chisq;
run;

* MODEL 4;
* Fit the same model but with separate diagonal Ri matrix for;
* each tech. Thus there are 2 separate variances sigma^2_(Pre and Post);
* D still = (2x2) unstructured matrix different for both techs;
* Specified in the RANDOM statement;
title 'RANDOM COEFFICIENT MODEL WITH DIAGONAL WITHIN-UMPIRE';
title2 'COVARIANCE MATRIX WITH SEPARATE CONSTANT VARIANCE FOR EACH TECH';
title3 'SAME D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml covtest cl;
  class Umpire tech coded1;
  model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
  repeated / group=tech subject=umpire;
  random intercept coded/type=un subject=umpire group=tech;* g gcorr v
vcorr;
  estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
  contrast 'overall tech diff' tech 1 -1, Coded*tech 1 -1/chisq;
run;

*****;
* Models with Ri=AR(1), D=UNSTRUCTURED          ;
* SAME-SAME, DIFF-SAME, SAME-DIFF, DIFF-DIFF    ;
*****;

* MODEL 5;
* Ri is AR(1) with the same variance and rho value for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix same for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH AR(1) WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE SAME FOR EACH TECH';
title3 'SAME D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
  class Umpire tech gen;
  model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
  random intercept coded/type=un subject=umpire;* g gcorr v vcorr;
  repeated / type=ar(1) subject=umpire;* r rcorr;

```



```

        estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
        contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

* MODEL 6;
* Ri is AR(1) with the different variance and rho value for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix same for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH AR(1) WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE DIFFERENT FOR EACH TECH';
title3 'SAME D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
    class Umpire tech gen;
    model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
    random intercept coded/type=un subject=umpire;* g gcorr v vcorr;
    repeated / type=ar(1) subject=umpire group=tech;* r rcorr;
    estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
    contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

* MODEL 7;
* Ri is AR(1) with the same variance and rho value for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix different for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH AR(1) WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE SAME FOR EACH TECH';
title3 'DIFFERENT D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
    class Umpire tech gen;
    model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
    random intercept coded/type=un subject=umpire group=tech;* g gcorr v
vcorr;
    repeated / type=ar(1) subject=umpire;* r rcorr;
    estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
    contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

* MODEL 8;
* Ri is AR(1) with the different variance and rho value for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix different for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH AR(1) WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE DIFFERENT FOR EACH TECH';
title3 'DIFFERENT D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
    class Umpire tech gen;
    model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;

```

```

    random intercept coded/type=un subject=umpire group=tech g gcorr v
vcorr;
    repeated / type=ar(1) subject=umpire group=tech;* r rcorr;
    estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
    contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

*****;
* Models with Ri=TOEP(2), D=UNSTRUCTURED      ;
* SAME-SAME, DIFF-SAME, SAME-DIFF, DIFF-DIFF ;
*****;

* MODEL 9;
* Ri is TOEP(2) with the same variance for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix same for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH COMMON ONE-DEPENDENT WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE SAME FOR EACH TECH';
title3 'SAME D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
    class Umpire tech gen;
    model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
    random intercept coded/type=un subject=umpire;* g gcorr v vcorr;
    repeated / type=toep(2) subject=umpire;* r rcorr;
    estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
    contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

* MODEL 10;
* Ri is TOEP(2) with the Different variance for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix same for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH COMMON ONE-DEPENDENT WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE DIFFERENT FOR EACH TECH';
title3 'SAME D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
    class Umpire tech gen;
    model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
    random intercept coded/type=un subject=umpire;* g gcorr v vcorr;
    repeated / type=toep(2) subject=umpire group=tech;* r rcorr;
    estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
    contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

* MODEL 11;
* Ri is TOEP(2) with the same variance for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix different for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH COMMON ONE-DEPENDENT WITHIN-UMPIRE';

```

```

title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE SAME FOR EACH TECH';
title3 'DIFFERENT D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
  class Umpire tech gen;
  model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
  random intercept coded/type=un subject=umpire group=tech;* g gcorr v
vcorr;
  repeated / type=toep(2) subject=umpire;* r rcorr;
  estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
  contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

* MODEL 12;
* Ri is TOEP(2) with the different variance for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix different for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH COMMON ONE-DEPENDENT WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE DIFFERENT FOR EACH TECH';
title3 'DIFFERENT D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
  class Umpire tech gen;
  model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
  random intercept coded/type=un subject=umpire group=tech;* g gcorr v
vcorr;
  repeated / type=toep(2) subject=umpire group=tech;* r rcorr;
  estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
  contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

*****;
* Models with Ri=TOEP(3), D=UNSTRUCTURED ;
* SAME-SAME, DIFF-SAME, SAME-DIFF, DIFF-DIFF ;
*****;

* MODEL 13;
* Ri is TOEP(3) with the same variance for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix same for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH COMMON TWO-DEPENDENT WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE SAME FOR EACH TECH';
title3 'SAME D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
  class Umpire tech gen;
  model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
  random intercept coded/type=un subject=umpire;* g gcorr v vcorr;
  repeated / type=toep(3) subject=umpire;* r rcorr;
  estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
  contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

```

```

* MODEL 14;
* Ri is TOEP(3) with the Different variance for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix same for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH COMMON TWO-DEPENDENT WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE DIFFERENT FOR EACH TECH';
title3 'SAME D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
  class Umpire tech gen;
  model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
  random intercept coded/type=un subject=umpire;* g gcorr v vcorr;
  repeated / type=toep(3) subject=umpire group=tech;* r rcorr;
  estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
  contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

* MODEL 15;
* Ri is TOEP(3) with the same variance for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix different for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH COMMON TWO-DEPENDENT WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE SAME FOR EACH TECH';
title3 'DIFFERENT D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
  class Umpire tech gen;
  model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
  random intercept coded/type=un subject=umpire group=tech;* g gcorr v
vcorr;
  repeated / type=toep(3) subject=umpire;* r rcorr;
  estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
  contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

* MODEL 16;
* Ri is TOEP(3) with the different variance for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix different for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH COMMON TWO-DEPENDENT WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE DIFFERENT FOR EACH TECH';
title3 'DIFFERENT D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
  class Umpire tech gen;
  model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
  random intercept coded/type=un subject=umpire group=tech;* g gcorr v
vcorr;
  repeated / type=toep(3) subject=umpire group=tech;* r rcorr;
  estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
  contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;

```

```

run;

*****;
* Models with Ri=CS, D=UNSTRUCTURED          ;
* SAME-SAME, DIFF-SAME, SAME-DIFF, DIFF-DIFF ;
*****;

* MODEL 17;
* Ri is CS with the same variance for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix same for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH COMMON COMPOUND SYMMETRIC WITHIN-
UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE SAME FOR EACH TECH';
title3 'SAME D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
    class Umpire tech gen;
    model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
    random intercept coded/type=un subject=umpire;* g gcorr v vcorr;
    repeated / type=cs subject=umpire;* r rcorr;
    estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
    contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

* MODEL 18;
* Ri is CS with the Different variance for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix same for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH COMMON CS WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE DIFFERENT FOR EACH TECH';
title3 'SAME D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
    class Umpire tech gen;
    model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
    random intercept coded/type=un subject=umpire g gcorr v vcorr;
    repeated / type=cs subject=umpire group=tech r rcorr;
    estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
    contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

* MODEL 19;
* Ri is CS with the same variance for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix different for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH COMMON CS WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE SAME FOR EACH TECH';
title3 'DIFFERENT D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
    class Umpire tech gen;

```

```

        model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
        random intercept coded/type=un subject=umpire group=tech;* g gcorr v
vcorr;
        repeated / type=cs subject=umpire;* r rcorr;
        estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
        contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

* MODEL 20;
* Ri is CS with the different variance for each tech;
* Specified in the REPEATED statement;
* D = (2x2) unstructured matrix different for both tech;
* Specified in the RANDOM statement;

title 'RANDOM COEFFICIENT MODEL WITH COMMON CS WITHIN-UMPIRE';
title2 'CORRELATION MATRIX WITH CONSTANT VARIANCE DIFFERENT FOR EACH TECH';
title3 'DIFFERENT D MATRIX FOR BOTH TECHS';
proc mixed data=masters method=ml;
        class Umpire tech gen;
        model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
        random intercept coded/type=un subject=umpire group=tech;* g gcorr v
vcorr;
        repeated / type=cs subject=umpire group=tech;* r rcorr;
        estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
        contrast 'overall tech diff' tech 1 -1, tech*coded 1 -1/chisq;
run;

```

Code for our fixed-slopes model

```

*MODEL 21;
*Fit the mixed model but with separate diagonal Ri matrix for;
*each tech. Thus there are 2 separate variances sigma^2_(Pre and Post);
*D = (1x1) matrix same for both techs;
*Specified in the RANDOM statement;
title 'MIXED EFFECTS MODEL WITH DIAGONAL WITHIN-UMPIRE';
title2 'COVARIANCE MATRIX WITH SEPARATE CONSTANT VARIANCE FOR EACH TECH';
title3 'SAME D SCALAR FOR BOTH TECHS, INTERCEPTS RANDOM, SLOPES FIXED';
proc mixed data=masters method=ml covtest cl;
        class Umpire tech coded1;
        model Prop_Incorrect = tech tech*coded/noint solution chisq
ddfm=satterth;
        repeated / group=tech subject=umpire r rcorr;
        random intercept/type=un subject=umpire g gcorr v vcorr;
        estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;
        contrast 'overall tech diff' tech 1 -1, coded*tech 1 -1/chisq;
run;

```

Code for our general linear model

```

*MODEL 22;
*Fit the general linear model but with separate diagonal Ri matrix for;
*each tech. Thus there are 2 separate variances sigma^2_(Pre and Post);
title 'GENERAL LINEAR MODEL WITH DIAGONAL WITHIN-UMPIRE';
title2 'COVARIANCE MATRIX WITH SEPARATE CONSTANT VARIANCE FOR EACH TECH';

```

```
title3 'SAME D SCALAR FOR BOTH TECHS, INTERCEPTS FIXED, SLOPES FIXED';  
proc mixed data=masters method=ml covtest cl;  
  class Umpire tech coded1;  
  model Prop_Incorrect = tech tech*coded/noint solution chisq  
ddfm=satterth;  
  repeated / group=tech subject=umpire r rcorr;  
  estimate 'diff in mean slope' tech 0 0 tech*coded 1 -1;  
  contrast 'overall tech diff' tech 1 -1, coded*tech 1 -1/chisq;  
run;  
ods html close;
```

Appendix C – SAS Output

RANDOM COEFFICIENT MODEL WITH TWO-DEPENDENT WITHIN-UMPIRE
CORRELATION MATRIX WITH CONSTANT VARIANCE DIFFERENT FOR EACH TECH
DIFFERENT D MATRIX FOR BOTH TECHS

5

The Mixed Procedure

Dimensions

Covariance Parameters	12
Columns in X	4
Columns in Z Per Subject	4
Subjects	97
Max Obs Per Subject	12

Number of Observations

Number of Observations Read	928
Number of Observations Used	928
Number of Observations Not Used	0

Iteration History

Iteration	Evaluations	-2 Log Like	Criterion
0	1	-5239.80764757	
1	4	-5392.54824588	0.00138619
2	2	-5395.36449328	0.00024857
3	1	-5396.19788788	0.00001174
4	1	-5396.24129828	0.00000006
5	1	-5396.24151814	0.00000000

Convergence criteria met.

Covariance Parameter Estimates

Cov Parm	Subject	Group	Estimate
UN(1,1)	Umpire	tech Post-tec	0.000132
UN(2,1)	Umpire	tech Post-tec	-7.93E-6
UN(2,2)	Umpire	tech Post-tec	5.47E-7
UN(1,1)	Umpire	tech Pre-tech	0.000040
UN(2,1)	Umpire	tech Pre-tech	4.91E-7
UN(2,2)	Umpire	tech Pre-tech	2.92E-23
Variance	Umpire	tech Post-tec	0.000281
TOEP(2)	Umpire	tech Post-tec	-0.00004
TOEP(3)	Umpire	tech Post-tec	-0.00006
Variance	Umpire	tech Pre-tech	0.000134
TOEP(2)	Umpire	tech Pre-tech	4.58E-6
TOEP(3)	Umpire	tech Pre-tech	-1.6E-6

Fit Statistics

-2 Log Likelihood	-5396.2
AIC (smaller is better)	-5366.2
AICC (smaller is better)	-5365.7
BIC (smaller is better)	-5327.6

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
10	156.43	<.0001

Solution for Fixed Effects

Effect	tech	Estimate	Standard Error	DF	t Value	Pr > t
tech	Post-tec	0.1534	0.003725	11.1	41.18	<.0001
tech	Pre-tech	0.1526	0.001173	70.8	130.08	<.0001
Coded*tech	Post-tec	-0.00250	0.000374	8.86	-6.69	<.0001
Coded*tech	Pre-tech	-0.00210	0.000125	161	-16.87	<.0001

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	Chi-Square	F Value	Pr > ChiSq	Pr > F
tech	2	18.1	18616.5	9308.25	<.0001	<.0001
Coded*tech	2	15.2	329.56	164.78	<.0001	<.0001

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > t
diff in mean slope	-0.00040	0.000394	10.9	-1.01	0.3339

Contrasts

Label	Num DF	Den DF	Chi-Square	F Value	Pr > ChiSq	Pr > F
overall tech diff	2	18.3	2.95	1.48	0.2285	0.2545

MIXED EFFECTS MODEL WITH DIAGONAL WITHIN-UMPIRE
 COVARIANCE MATRIX WITH SEPARATE CONSTANT VARIANCE FOR EACH TECH
 SAME D SCALAR FOR BOTH TECHS, INTERCEPTS RANDOM, SLOPES FIXED

109

The Mixed Procedure

Dimensions

Covariance Parameters	3
Columns in X	4
Columns in Z Per Subject	1
Subjects	97
Max Obs Per Subject	12

Number of Observations

Number of Observations Read	928
Number of Observations Used	928
Number of Observations Not Used	0

Iteration History

Iteration	Evaluations	-2 Log Like	Criterion
0	1	-5239.80764757	
1	2	-5389.04065330	0.00000168
2	1	-5389.04666851	0.00000000

Convergence criteria met.

Covariance Parameter Estimates

Cov Parm	Subject	Group	Estimate	Standard Error	Z Value	Pr > Z	Alpha	Lower
UN(1,1)	Umpire		0.000043	8.782E-6	4.94	<.0001	0.05	0.000030
Residual	Umpire	tech Post-tec	0.000303	0.000035	8.64	<.0001	0.05	0.000244
Residual	Umpire	tech Pre-tech	0.000133	7.188E-6	18.57	<.0001	0.05	0.000120

Covariance Parameter Estimates

Cov Parm	Subject	Group	Upper
UN(1,1)	Umpire		0.000067
Residual	Umpire	tech Post-tec	0.000385
Residual	Umpire	tech Pre-tech	0.000149

Fit Statistics

-2 Log Likelihood	-5389.0
AIC (smaller is better)	-5375.0
AICC (smaller is better)	-5374.9
BIC (smaller is better)	-5357.0

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
2	149.24	<.0001

Solution for Fixed Effects

Effect	tech	Estimate	Standard Error	DF	t Value	Pr > t
tech	Post-tec	0.1537	0.003580	205	42.95	<.0001
tech	Pre-tech	0.1526	0.001178	238	129.54	<.0001
Coded*tech	Post-tec	-0.00260	0.000410	179	-6.34	<.0001
Coded*tech	Pre-tech	-0.00210	0.000122	707	-17.23	<.0001

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	Chi-Square	F Value	Pr > ChiSq	Pr > F
tech	2	220	18627.1	9313.56	<.0001	<.0001
Coded*tech	2	284	336.87	168.43	<.0001	<.0001

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > t
diff in mean slope	-0.00050	0.000428	211	-1.16	0.2469

Contrasts

Label	Num DF	Den DF	Chi-Square	F Value	Pr > ChiSq	Pr > F
overall tech diff	2	221	2.69	1.35	0.2603	0.2625

GENERAL LINEAR MODEL WITH DIAGONAL WITHIN-UMPIRE
 COVARIANCE MATRIX WITH SEPARATE CONSTANT VARIANCE FOR EACH TECH
 SAME D SCALAR FOR BOTH TECHS, INTERCEPTS FIXED, SLOPES FIXED

114

The Mixed Procedure

Dimensions

Covariance Parameters	2
Columns in X	4
Columns in Z	0
Subjects	97

Max Obs Per Subject 12

Number of Observations

Number of Observations Read 928
Number of Observations Used 928
Number of Observations Not Used 0

Iteration History

Iteration Evaluations -2 Log Like Criterion
0 1 -5239.80764757
1 1 -5266.47004159 0.00000000

Convergence criteria met.

Covariance Parameter Estimates

Cov Parm Subject Group Estimate Standard Error Z Pr > Z Alpha Lower
Residual Umpire tech Post-tec 0.000326 0.000036 9.08 <.0001 0.05 0.000266
Residual Umpire tech Pre-tec 0.000181 9.259E-6 19.53 <.0001 0.05 0.000164

Covariance Parameter Estimates

Cov Parm Subject Group Upper
Residual Umpire tech Post-tec 0.000410
Residual Umpire tech Pre-tec 0.000200

Fit Statistics

-2 Log Likelihood -5266.5
AIC (smaller is better) -5254.5
AICC (smaller is better) -5254.4
BIC (smaller is better) -5239.0

Null Model Likelihood Ratio Test

DF Chi-Square Pr > ChiSq
1 26.66 <.0001

Solution for Fixed Effects

Effect tech Estimate Standard Error DF t Value Pr > |t|

tech	Post-tec	0.1526	0.003138	165	48.64	<.0001
tech	Pre-tech	0.1523	0.001018	763	149.60	<.0001
Coded*tech	Post-tec	-0.00247	0.000396	165	-6.24	<.0001
Coded*tech	Pre-tech	-0.00209	0.000140	763	-14.90	<.0001

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	Chi-Square	F Value	Pr > ChiSq	Pr > F
tech	2	270	24747.6	12373.8	<.0001	<.0001
Coded*tech	2	270	260.88	130.44	<.0001	<.0001

The Mixed Procedure

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > t
diff in mean slope	-0.00038	0.000420	208	-0.91	0.3664

Contrasts

Label	Num DF	Den DF	Chi-Square	F Value	Pr > ChiSq	Pr > F
overall tech diff	2	208	3.39	1.69	0.1839	0.1864